

Beatriz Rubio Serrano

Algorithms for curve design and accurate computations with totally positive matrices

Director/es
Mainar Maza, Esmeralda

<http://zaguan.unizar.es/collection/Tesis>



Universidad
Zaragoza

Tesis Doctoral

ALGORITHMS FOR CURVE DESIGN AND
ACCURATE COMPUTATIONS WITH TOTALLY
POSITIVE MATRICES

Autor

Beatriz Rubio Serrano

Director/es

Mainar Maza, Esmeralda

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

2021

**ALGORITHMS FOR CURVE DESIGN AND
ACCURATE COMPUTATIONS WITH
TOTALLY POSITIVE MATRICES**

BEATRIZ RUBIO SERRANO

Supervisor: Prof. Dr. ESMERALDA MAINAR MAZA

DOCTORAL THESIS IN MATHEMATICS
Department of Applied Mathematics-IUMA
University of Zaragoza
May, 2021

This doctoral thesis is presented as a compendium of the following publications:

ARTICLE 1

- [73] E. Mainar, J.M. Peña, B. Rubio, Evaluation and subdivision algorithms for general classes of totally positive rational bases, *Computer Aided Geometric Design* **81** (2020).
<https://doi.org/10.1016/j.cagd.2020.101900>

ARTICLE 2

- [74] E. Mainar, J.M. Peña, B. Rubio, Accurate bidiagonal decomposition of collocation matrices of weighted φ -transformed systems, *Numerical Linear Algebra Appl.* **e2295** (2020).
<https://doi.org/10.1002/nla.2295>

ARTICLE 3

- [39] R. Gonzalez, E. Mainar, E. Paluzo, B. Rubio, Neural-Network-Based Curve Fitting Using Totally Positive Rational Bases, *Mathematics* **8**, 2197 (2020).
<https://doi.org/10.3390/math8122197>

ARTICLE 4

- [75] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with Wronskian matrices, *Calcolo* **58**, 1 (2021).
<https://doi.org/10.1007/s10092-020-00392-4>

ARTICLE 5

- [76] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with collocations and Wronskian matrices of Jacoby polynomials, *Journal of Scientific Computing* (2021), 87, 7.
<https://doi.org/10.1007/s10915-021-01500-4>

The work developed in this thesis is part of the projects:

- E41-20R: Análisis Numérico, Optimización y Aplicaciones. Gobierno de Aragón.
- E41-17R: Análisis Numérico, Optimización y Aplicaciones. Gobierno de Aragón.
- PGC2018-096321-B-I00: Análisis de la representación de curvas y superficies, cálculos precisos con matrices estructuradas y aplicaciones.
- MTM2015-65433-P: Métodos Numéricos en la aproximación de curvas y superficies, matrices positivas y aplicaciones.

In addition, the FPI grant: BES-2016-078789 of the Spanish Ministry of Science and Innovation sponsored the author.

Disseminations of results

Published articles

- [72] E. Mainar, J.M. Peña, B. Rubio. Accurate least squares fitting with a general class of shape preserving bases, *Monografías Matemáticas García de Galdeano* ISBN 978-84-1340-039-6, 183–182 (2019).
- [25] L. Diaz, B. Rubio, J.A. Albajez, J.A. Yagüe, E. Mainar, M. Torralba, Trajectory Definition with High Relative Accuracy (HRA) by Parametric Representation of Curves in Nano-Positioning Systems, *Micromachines* **10**, 597 (2019).
<https://doi.org/10.3390/mi10090597>
- [73] E. Mainar, J.M. Peña, B. Rubio, Evaluation and subdivision algorithms for general classes of totally positive rational bases, *Computer Aided Geometric Design* **81** (2020).
<https://doi.org/10.1016/j.cagd.2020.101900>
- [74] E. Mainar, J.M. Peña, B. Rubio, Accurate bidiagonal decomposition of collocation matrices of weighted φ -transformed systems, *Numerical Linear Algebra Appl.* **e2295** (2020).
<https://doi.org/10.1002/nla.2295>
- [39] R. Gonzalez, E. Mainar, E. Paluzo, B. Rubio, Neural-Network-Based Curve Fitting Using Totally Positive Rational Bases, *Mathematics* **8**, 2197 (2020).
<https://doi.org/10.3390/math8122197>
- [75] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with Wronskian matrices, *Calcolo* **58**, 1 (2021).
<https://doi.org/10.1007/s10092-020-00392-4>
- [76] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with collocations and Wronskian matrices of Jacoby polynomials, *Journal of Scientific Computing* **87**, 77 (2021).
<https://doi.org/10.1007/s10915-021-01500-4>

Under review articles

- [77] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with Wronskian matrices of Bessel and Laguerre polynomials.
- [76] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with Wronskian matrices of Bernstein and related bases.

- [79] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with Wronskian matrices of geometric and Poisson bases.

Conference contributions

- E. Mainar, J.M. Peña, B. Rubio. Oral presentation. “Weighted φ -transformed system” in the 9th Conference on Curves and Surface, Arcachon, Francia, 2018.
- E. Mainar, J.M. Peña, B. Rubio. Oral presentation. “Shape preserving properties of general class of bases and accurate computation” in the Fifteenth International Conference Zaragoza-Pau, Jaca, 2018.
- E. Mainar, J.M. Peña, B. Rubio. Oral presentation. “Shape Preserving Properties and Algorithms for Weighted φ -Transformed Systems” in CGTA2019 Conference on Geometry: Theory and Applications, Innsbruck, Austria, 2019.
- E. Mainar, J.M. Peña, B. Rubio. Oral presentation. “Weighted φ -transformed systems” in International Geometry Summit, School of Computer Science, Simon Fraser University, Vancouver, Canada, 2019.
- E. Mainar, J.M. Peña, B. Rubio. Oral presentation. “Algorithms for general classes of rational bases” in the Workshop on CAGD and Robotics, Universidad CEU Cardenal Herrera, Valencia, 2020.
- E. Mainar, J.M. Peña, B. Rubio. Oral presentation. “Total positivity and Weighted φ -Transformed systems” in Seminar of recent results on CAGD, Institute of Applied Geometry, Johannes Kepler University, Linz, Austria, 2020.

Research stay at Johannes Kepler University

Position: Visiting Scholar in the Institute of Applied Geometry at the Johannes Kepler University.

Period: From March 3rd to June 2nd 2020 (3 months).

Supervisor: Prof. Dr. Bert Jüttler.

Purpose of the work: Isogeometric analysis.

Agradecimientos

Ha llegado el mejor momento de un trabajo, el de agradecer a todas las personas que lo han hecho posible.

En primer lugar, quiero expresar mi más profundo agradecimiento a mi directora de tesis, la profesora Esmeralda Mainar, por su dedicación, perseverancia y coraje; por su guía y minucioso trabajo, que ha sido esencial en la realización de esta tesis. Por encarnar la disciplina, el rigor y la excelencia, y ser ejemplo vivo de que no son incompatibles con el entusiasmo, la alegría, la compasión, el amor y el cuidado de los que nos rodean.

Mención especial merece también el profesor Juan Manuel Peña por compartir conmigo su sabiduría, saber hacer, tiempo y esfuerzo en todo el trabajo que hemos realizado durante estos años. Gracias por su comprensión, apoyo y generosidad. Estaré eternamente agradecida por haberme permitido formar parte de su grupo de investigación.

No podría dejar de expresar mi más sincero agradecimiento a los profesores Luis Rández y Pedro J. Miana por el apoyo absoluto y la confianza que siempre me han demostrado.

Me gustaría también manifestar mi gratitud al Instituto Univesitario de Matemáticas y Aplicaciones (IUMA), al Departamento de Matemática Aplicada y, en especial, a mis compañeros de la EINA por sus ánimos y por el agradable ambiente de trabajo; por hacerme sentir como una más desde mi llegada y prestarme todo su apoyo.

Gracias al profesor Bert Jüttler por su hospitalidad en Linz y por compartir conmigo su conocimiento. Quisiera dar las gracias también a cada miembro del Instituto de Geometría Aplicada de la Universidad Johannes Kepler por hacerme sentir como en casa. Un especial recuerdo a Monika y a Somma.

A la profesora Lucía Hilario y al grupo de Investigación de Automatización Industrial y Robótica de la Universidad CEU Cardenal Herrera: gracias por los buenos momentos vividos.

La investigación tal y como la conocemos no existiría sin las instituciones públicas que la financian y mantienen. Este trabajo ha podido ser desarrollado gracias al Ministerio de Ciencia e Innovación y a la Agencia Estatal de Investigación (MTM2015-65433-P y PGC2018- 096321-B-I00), al Gobierno de Aragón (E41-17R) y al Fondo Europeo de Desarrollo Regional (FEDER) *Construyendo Europa desde Aragón*.

Nada habría sido posible de no haber tenido el amor incondicional de mi madre y de mi padre. Nada habría sido posible sin su lucha incansable para sacar la vida adelante y superarse cada día. Nada habría sido posible si no me hubieran enseñado a soñar con metas que parecían imposibles.

Gracias a Cotton por llenarnos de alegría y felicidad.

Gracias a Rosa y a Aniceto por su cariño y por ser un gran apoyo.

Gracias a mi tío Pedro, a primo Raúl, a Violeta y a Raquel, que siempre han creído en mi y han estado ahí para brindarme una palabra de aliento en el momento preciso. Gracias a todos los que, de una manera u otra, me han ayudado y aliviado en el camino, haciendo posible que hoy esté aquí. Gracias por llevarme de la mano. Gracias a todos de corazón.

A Álex le estoy más agradecida de lo que puedo expresar con palabras: por darme la energía cuando la mía se estaba agotando, por tener un oído atento a mis preocupaciones, por su fe en mí, por ser mi equilibrio y mi mejor mitad, pero sobre todo por su amor. Me has hecho comprender que el amor es para los que entienden el significado de la palabra «cuidar». La vida es muy fácil a tu lado.

Beatriz Rubio Serrano

Zaragoza, Mayo, 2021

Cada momento de búsqueda es un momento de encuentro — dijo el muchacho a su corazón—. Mientras busqué mi tesoro, todos mis días fueron luminosos, porque yo sabía que cada momento formaba parte del sueño de encontrar. Mientras busqué este tesoro mío, descubrí por el camino cosas que jamás habría soñado encontrar si no hubiese tenido el valor de intentar cosas imposibles para los pastores. El Alquimista, Paulo Coelho

Acknowledgment

The best time of a work has arrived, the time to thank all the people that have made it possible.

First of all, I would like to express my deepest thanks to my thesis supervisor, Professor Esmeralda Mainar, for her dedication, perseverance and courage; for her guidance and meticulous work, which has been essential for the realization of this thesis. Because she embodies discipline, rigour and excellence, being a living example that demonstrates that these qualities are not incompatible with those of compassion, love and care for the ones around us.

Professor Juan Manuel Peña also deserves a special mention for sharing with me his knowledge, know-how, time and effort in all the work we have done along these years. Thanks for his understanding, support and generosity. I will be eternally grateful to him for having allowed me to be part of his research group.

I could not fail to express my most sincere thanks to professors Luis Rández and Pedro J. Miana for the absolute support and trust they have always shown me.

I would like to manifest my gratitude to the University Institute of Mathematics and its Applications, to the Department of Applied Mathematics, and specially to my colleagues of the EINA for their encouragement and for the pleasant work environment; for making me feel like one of them since my arrival and for bringing me all their support.

Thanks to Professor Bert Jüttler for his hospitality in Linz and for sharing with me his knowledge. I would also like to thank each member of the Institute of Applied Geometry at Johannes Kepler University for making me feel at home, with a special mention to Monika and Somma.

To Professor Lucía Hilario and the Industrial Automation and Robotics Research Group of The Cardenal Herrera University: thanks for good times that we have enjoyed

Research as we know it would not exist without the public institutions that finance and support it. This thesis has been developed thanks to the Ministry of Science and Innovation and the State Research Agency (MTM2015-65433-P and PGC2018- 096321-B-I00), the Government of Aragon (E41-17R), and the European Development Fund Regional (ERDF)) *Construyendo Europa desde Aragón*.

Nothing would have been possible without the unconditional love of my parents. Nothing would have been possible without their tireless struggle to move forward and become better every day. Nothing would have been possible if they hadn't taught me to dream of goals that at first seemed to be impossible.

Thanks to Cotton for bringing us joy and happiness.

Thanks to Rosa and Aniceto for their love and for being a great support.

Thanks to my uncle Pedro, my cousin Raúl, Violeta y Raquel, who were always there to give me a word of encouragement at the right moment. Thanks to all those who, in one way or another, have helped me and relieved me along the way, making it possible for me to be where I am today. Thanks for taking me by the hand. Thanks to all of them from my heart.

I am more grateful to Alex than I can put into words: for giving me energy when mine was running low, for having an attentive ear to my worries, for his faith in me, for being my balance and my best

half; and above all for his love. You have made me realized that love is for those who understand the meaning of the word “care”. Life is really easy by your side.

Beatriz Rubio Serrano

Zaragoza, May, 2021

“Every second of the search is an encounter with God,” the boy told his heart. “When I have been truly searching for my treasure, every day has been luminous, because I’ve known that every hour was a part of the dream that I would find it. When I have been truly searching for my treasure, I’ve discovered things along the way that I never would have seen had I not had the courage to try things that seemed impossible for a shepherd to achieve.” El Alquimista, Paulo Coelho

Dedicada a mi madre, a Esme y a Álex

Resumen

Esta tesis doctoral se enmarca dentro de la teoría de la Positividad Total. Las matrices totalmente positivas han aparecido en aplicaciones de campos tan diversos como Teoría de Aproximación, Biología, Economía, Combinatoria, Estadística, Ecuaciones Diferenciales, Mecánica, Diseño Geométrico Asistido por Ordenador o Álgebra Lineal Numérica (veáse [58], [33], [2], [37], [27], [93], [23]). En esta tesis nos centraremos en dos de los campos que están relacionados con las matrices totalmente positivas.

Por un lado, esta tesis se inscribe dentro del campo del Diseño Geométrico Asistido por Ordenador (CAGD) y, en particular, del estudio de la representación de curvas por medio de polígonos de control. La importancia de las matrices totalmente positivas en CAGD proviene del hecho de que los sistemas normalizados totalmente positivos, cuyas matrices de colocación son estocásticas y totalmente positivas, proporcionan representaciones que preservan la forma [9, 92]. La base de los polinomios de Bernstein es la base polinómica más utilizada en CAGD (veáse [28], [31]). La matriz de colocación de la base de Bernstein es estocástica y totalmente positiva. De hecho, la base de Bernstein es la B-base normalizada de su espacio generado y tiene las óptimas propiedades de preservación de forma [9]. Las curvas definidas paramétricamente mediante esta base, llamadas curvas de Bézier, son de gran interés en CAGD, ya que proporcionan la representación de curvas polinómicas con óptimas propiedades de preservación de forma. El algoritmo de de Casteljau es un algoritmo de corte de esquinas con la propiedad de evaluar la curva Bézier a partir de su polígono de control. Otra importante propiedad de este algoritmo es la propiedad de subdivisión. Uno de los objetivos de esta tesis es encontrar nuevos sistemas de funciones, no necesariamente polinómicos, que generen curvas con propiedades geométricas y con algoritmos similares a la curvas de Bézier, y que expandan el rango de aplicaciones de las curvas Bézier pudiendo así alcanzar formas más complejas. Los principales resultados que obtenemos son:

- Dado un sistema inicial, un conjunto de pesos y una función positiva φ , definimos un nuevo sistema de funciones llamado sistema φ -transformado ponderado (*weighted φ -transformed system*).
- Mostramos que los sistemas φ -transformados ponderados incluyen importantes bases útiles en CAGD y en Estadística, como la base de Poisson, la base de Bernstein de grado negativo o las bases racionales.
- Probamos que los sistemas φ -transformados ponderados heredan del sistema inicial las propiedades de preservación de forma.
- Las curvas definidas mediante bases de Bernstein y B-spline racionales se han convertido en una herramienta estándar en CAGD, dado que permiten la representación exacta de numerosas secciones cónicas, de esferas y de cilindros. En esta tesis, obtenemos como un caso de sistema φ -transformado ponderado una clase general de bases racionales totalmente positivas que

pertencen a espacios racionales que mezclan polinomios algebraicos, trigonométricos e hiperbólicos. Además, presentamos algoritmos de evaluación y subdivisión para las curvas paramétricas generadas por estas bases.

- Obtenemos como un caso de las bases racionales anteriormente propuestas una clase particular de bases racionales totalmente positivas que satisfacen relaciones de recurrencia y generan nuevos espacios racionales anidados. Para estas bases, presentamos algoritmos de evaluación y subdivisión.
- Aplicando técnicas de Inteligencia Artificial, presentamos una red neuronal de una capa oculta basada en la clase general de bases racionales propuesta. Con esta red neuronal abordamos el problema de encontrar las curvas racionales que mejor se ajustan a un conjunto dado de puntos. En este proceso de aproximación, la base racional es un hiperparámetro y se puede cambiar sustituyendo los factores lineales por funciones polinómicas, trigonométricas o hiperbólicas, pudiendo así alcanzar formas más complejas y ampliando de esta manera el rango potencial de aplicaciones de esta red neuronal.

Por otro lado, nuestra investigación también se centra en el campo del Álgebra Lineal Numérica, concretamente en el diseño y análisis de algoritmos adaptados a la estructura de las matrices totalmente positivas que permitan resolver con alta precisión relativa problemas algebraicos asociados a estas matrices. Hoy en día, muchos de los problemas que surgen en Física, Ingeniería, Química, Biomedicina o CAGD, entre otros, requieren de métodos numéricos con los que resolver sistemas de ecuaciones lineales o con los que hallar los valores propios o valores singulares de las matrices asociadas al modelo matemático. Estos métodos numéricos son objeto de una intensa investigación debido a que en las aplicaciones actuales aparecen continuamente nuevas clases de matrices estructuradas (entre las que se encuentran las matrices totalmente positivas) con las que surge la necesidad de desarrollar algoritmos específicos más eficientes y/o precisos que los existentes.

La obtención de la factorización bidiagonal de las matrices totalmente positivas en términos de los multiplicadores de la eliminación de Neville (véase [34]) ha sido un punto muy importante a la hora de obtener algoritmos precisos con los que realizar cálculos algebraicos con un error mucho menor que el de los algoritmos convencionales y con el mismo coste computacional ([23], [62]). Hasta ahora, esto se ha logrado con algunas subclases relevantes de matrices totalmente positivas con aplicaciones en CAGD (cf. [83, 85, 17, 15, 71]), Finanzas (cf. [18]) o Combinatoria (cf. [16]). En esta tesis presentamos los resultados que garantizan las buenas propiedades computacionales de los sistemas φ -transformados ponderados. También, mostramos ejemplos de matrices wronskianas para las que se pueden realizar diferentes cálculos algebraicos con alta precisión relativa. Las matrices wronskianas surgen frecuentemente en diferentes aplicaciones. Por ejemplo, en los problemas de interpolación de Hermite, y en particular en los problemas de interpolación de Taylor. Sin embargo, hasta ahora, no hay ejemplos de cálculos precisos para matrices que involucren derivadas de las funciones de las bases. Los principales resultados que obtenemos son:

- A partir de la factorización bidiagonal de las matrices de colocación de un sistema inicial, diseñamos un algoritmo preciso para construir la factorización bidiagonal de la matriz de colocación del correspondiente sistema φ -transformado ponderado. Este algoritmo lo utilizamos para realizar con precisión diferentes cálculos algebraicos. Debido a las buenas propiedades geométricas de las curvas generadas por los sistemas φ -transformados ponderados, este algoritmo puede ser útil en problemas de interpolación y aproximación.

- Presentamos algoritmos precisos con los que calcular la factorización bidiagonal de la matriz wronskiana de la base de los monomios y la factorización bidiagonal de la matriz wronskiana de la base de los polinomios exponenciales. También se muestra que estos algoritmos se pueden utilizar para realizar con precisión algunos cálculos algebraicos asociados a estas matrices wronskianas, como el cálculo de sus inversas, sus valores propios o sus valores singulares y las soluciones de algunos sistemas lineales.
- Obtenemos un método preciso para construir la factorización bidiagonal de las matrices de colocación y wronskianas de los polinomios de Jacobi, y lo utilizamos para calcular con alta precisión relativa sus valores propios, valores singulares, matrices inversas y la solución de algunos sistemas lineales asociados a estas matrices. Consideramos también los casos particulares de las matrices de colocación y wronskianas de los polinomios de Legendre, polinomios de Gegenbauer, polinomios de Chebyshev de primer y segundo tipo, y los polinomios racionales de Jacobi.
- Diseñamos un método para obtener la factorización bidiagonal de la matriz wronskiana de los polinomios de Bessel y la factorización bidiagonal de la matriz wronskiana de los polinomios de Laguerre. Utilizamos este método para calcular con alta precisión relativa sus valores singulares, sus matrices inversas, así como la solución de algunos sistemas lineales.
- Proporcionamos un algoritmo con el que obtener la factorización bidiagonal de las matrices wronskianas de la base de los polinomios de Bernstein y la factorización bidiagonal de otras bases relacionadas, como la base de Bernstein de grado negativo o la base binomial negativa. También mostramos que este algoritmo puede usarse para realizar con alta precisión relativa algunos cálculos algebraicos con estas matrices wronskianas, como el cálculo de sus inversas, sus valores propios o sus valores singulares y las soluciones de algunos sistemas lineales relacionados.
- Diseñamos algoritmos con los que obtener la factorización bidiagonal de la matriz wronskiana de la base geométrica y la factorización bidiagonal de la matriz wronskiana de la base de Poisson. Estos algoritmos los utilizamos para calcular con precisión diferentes cálculos algebraicos.
- La complejidad de todos los algoritmos propuestos para resolver los problemas algebraicos mencionados es comparable a la de los algoritmos LAPACK tradicionales, los cuales, como ilustraremos, no ofrecen tal precisión.

Esta memoria es una tesis doctoral por compendio de publicaciones y está estructurada en cinco partes. La primera parte está compuesta por un lado, por la Introducción, y por otro lado, por los resultados auxiliares y las herramientas que vamos a emplear en el desarrollo del trabajo. En la segunda parte, presentamos los artículos [73], [74], [39], [75] y [76] que pertenecen al compendio de publicaciones de esta tesis. En la tercera parte, justificamos la unidad temática de las publicaciones mencionadas. También incluimos sus principales resultados. En la cuarta parte, presentamos los últimos resultados que hemos obtenido y que no están incluidos en los artículos que pertenecen al compendio de publicaciones de esta tesis. Finalmente, en la quinta parte, se describen las conclusiones y el posible trabajo futuro que puede continuar desarrollándose como resultado de la investigación de esta tesis. El código de los algoritmos y de los experimentos numéricos se puede encontrar y descargar en la siguiente dirección web: <https://github.com/BeatrizRubio>.

Abstract

This doctoral thesis is framed within the theory of Total Positivity. Totally positive matrices have appeared in applications from fields as diverse as Approximation Theory, Biology, Economics, Combinatorics, Statistics, Differential Equations, Mechanics, Computer Aided Geometric Design or Linear Numerical Algebra (see [58], [33], [2], [37], [27], [93], [23]). In this thesis, we will focus on two of the fields that are related to totally positive matrices.

On the one hand, this thesis falls within the field of Computer-Aided Geometric Design (CAGD) and, in particular, the study of the representation of curves by means of control polygons. The importance of totally positive matrices comes from the fact that the normalized totally positive systems, whose collocation matrices are totally positive, provide shape preserving representations [9, 92]. The Bernstein basis of polynomials is the most used polynomial basis in CAGD (see [28], [31]). The collocation matrix of the Bernstein basis is stochastic and totally positive. In fact, the Bernstein basis is the normalized B-basis of its generated space and has the optimal shape preserving properties [9]. The curves defined parametrically by means of this basis, called Bézier curves, are of great interest in CAGD since they provide the representation of polynomial curves with optimal shape preserving properties. The de Casteljau algorithm is a corner cutting algorithm with the property of evaluating the Bézier curve from its control polygon. Another important property of this algorithm is the subdivision property. One of the objectives of this thesis is to find new systems of functions, not necessarily polynomials, that generate curves with geometric properties and with algorithms similar to the Bézier curves and that expand the range of applications of the Bézier curves, thus being able to reach more complex shapes. The main results we obtain are:

- Given an initial system, a set of weights and, a positive function φ , we define a new system of functions called weighted φ -transformed system.
- We show that weighted φ -transformed systems include important bases useful in CAGD and Statistics, such as Poisson basis, Bernstein basis of negative degree or rational bases.
- We prove that weighted φ -transformed system inherits from the initial system its shape preserving properties.
- Curves defined by Bernstein bases and rational B-splines have become a standard tool in CAGD since they allow the exact representation of numerous conic, sphere, and cylinder sections. In this thesis, we obtain as an example of a weighted φ -transformed system a general class of totally positive rational bases that belong to rational spaces that mix algebraic, trigonometric and hyperbolic polynomials. In addition, we present evaluation and subdivision algorithms for the parametric curves generated by these bases.

- We obtain as an example of the general class of rational bases previously proposed a particular class of totally positive rational bases that satisfy recurrence relations and generate new nested rational spaces. For these bases, we present evaluation and subdivision algorithms.
- Applying Artificial Intelligence techniques, we present a one-hidden-layer neural network based on the proposed general class of rational bases. With this neural network, we tackle the problem of finding the rational curves that best fit a given set of data points. In this process of approximation, the rational basis is a hyperparameter and can be changed by substituting the linear factors for polynomial, trigonometric or hyperbolic functions, thus being able to reach more difficult shapes and expanding in this way the potential range of applications of this neural network.

On the other hand, our research also focuses on the field of Numerical Linear Algebra, specifically on the design and analysis of algorithms adapted to the structure of totally positive matrices that allow us to solve with high relative accuracy algebraic problems associated with these matrices. Nowadays, many of the problems that arise in Physics, Engineering, Chemistry, Biomedicine or CAGD, among others, require numerical methods with which to solve systems of linear equations or with which to find the eigenvalues or singular values of the associated matrices to the mathematical model. These numerical methods are the object of intense research because in current applications new classes of structured matrices are continually appearing (among we can find totally positive matrices) with which the need arises to develop more efficient and/or accurate specific algorithms than the existing ones.

Obtaining the bidiagonal factorization of the totally positive matrices in terms of the multipliers of the Neville elimination (see [34]) has been a very important point in obtaining accurate algorithms with which to perform algebraic calculations with these matrices with a much smaller error than that of conventional algorithms and with the same computational cost ([23], [62]). Up to now, this has been achieved with some relevant subclasses of TP matrices with applications to CAGD (cf. [83, 85, 17, 15, 71]), to Finance (cf. [18]) or to Combinatorics (cf. [16]). In this thesis, we present the results that guarantee the good computational properties of the weighted φ -transformed systems. Also, we show examples of Wronskian matrices for which different algebraic computations can be performed with high relative accuracy. Wronskian matrices frequently arise in different applications, for instance, in Hermite interpolation problems, and in particular in Taylor interpolation problems. However, so far, there are no examples of accurate computations for matrices involving derivatives of the basis functions. The main results we obtain are:

- From the bidiagonal factorization of the collocation matrix of an initial system we design an accurate algorithm to construct the bidiagonal factorization of the collocation matrix of the corresponding weighted φ -transformed system. We use this algorithm to perform accurately different algebraic computations. Due to the good geometric properties of the curves generated by the weighted φ -transformed systems, this algorithm can be useful in interpolation and approximation problems.
- We present accurate algorithms with which to calculate the bidiagonal factorization of the Wronskian matrix of the monomial basis of polynomials and the bidiagonal factorization of the Wronskian matrix of the basis of exponential polynomials. It is also shown that these algorithms can be used to perform accurately some algebraic computations associated with these Wronskian matrices, such as the computation of their inverses, their eigenvalues or their singular values, and the solutions of some linear systems.

- We obtain an accurate method to construct the bidiagonal factorization of the collocation and Wronskian matrices of the Jacobi polynomials, and we use it to compute with high relative accuracy their eigenvalues, singular values, inverse matrices and, the solution of some linear systems associated with these matrices. We also consider the particular cases of collocation and Wronskian matrices of Legendre polynomials, Gegenbauer polynomials, Chebyshev polynomials of the first and second type, and Jacobi rational polynomials.
- We design a method to obtain the bidiagonal factorization of the Wronskian matrix of the Bessel polynomials and the bidiagonal factorization of the Wronskian matrix of the Laguerre polynomials. We use this method to compute with high relative accuracy their singular values, inverse matrices, as well as the solution of some linear systems.
- We provide an algorithm to obtain the bidiagonal factorization of the Wronskian matrices of Bernstein polynomials and the bidiagonal factorization of other related bases, such as the negative degree Bernstein basis or the negative binomial basis. We also show that this algorithm can be used to perform with high relative accuracy some algebraic computations with these Wronskian matrices, such as the computation of their inverses, their eigenvalues or their singular values, and the solutions of some related linear systems.
- We design accurate algorithms with which to obtain the bidiagonal factorization of the Wronskian matrix of the geometric basis and the bidiagonal factorization of the Wronskian matrix of the Poisson basis. We use these algorithms to compute accurately different algebraic computations.
- The complexity of all the proposed algorithms for solving the mentioned algebraic problems is comparable to that of the traditional LAPACK algorithms, which, as we will illustrate, deliver no such accuracy.

This work is a doctoral thesis by compendium of publications and is structured in five parts. The first part is composed on the one hand, by the Introduction, and on the other hand, by the Background with the auxiliary results and the tools that we are going to use in the development of the work. In the second part, we present the articles [73], [74], [39], [75] and [76] which belong to the compendium of publications of this thesis. In the third part, we justify the thematic unit of the mentioned publications. We also include their main results. In the fourth part, we present the latest results that we have obtained and that are not included in the articles that belong to the compendium of publications of this thesis. Finally, in the fifth part, the conclusions and the possible future work that might continue to be developed as a result of the research of this thesis are described. The code of the algorithms and of the numerical experiments can be found and downloaded at the following website: <https://github.com/BeatrizRubio>.

Contents

Part I: INTRODUCTION AND BACKGROUND	3
1 Introducción	5
2 Introduction	15
3 Background	25
3.1 Error analysis and high relative accuracy	25
3.2 Totally positive matrices, Neville elimination and computations with high relative accuracy	26
3.3 Normalized totally positive bases and corner cutting algorithms	33
Part II: PRESENTATION OF THE PUBLICATIONS	41
Article 1	43
Article 2	59
Article 3	77
Article 4	99
Article 5	117
Part III: JUSTIFICATION OF THE THEMATIC UNIT AND SUMMARY OF THE PUBLICATIONS	149
4 Total positivity and Weighted φ-transformed systems: shape preserving properties and accurate computations	151
4.1 Introduction	151
4.2 Notations and auxiliary results	153
4.3 Weighted φ -transformed systems	155
4.4 A general class of rational bases	157
4.5 A particular class of rational bases satisfying recurrence relations	160
4.6 Bidiagonal factorization of rational bases with high relative accuracy	161
4.7 Interesting examples of rational bases	162
4.8 Accurate computations with collocation matrices of rational bases	165
4.9 Curve Fitting with Neural Networks using a general class of rational bases	167
4.10 Accurate least squares fitting with a general class of rational bases	171

5	Total positivity and accurate computations with Wronskian matrices of monomial, exponential and Jacobi polynomials	173
5.1	Introduction	173
5.2	Notations and previous results	174
5.3	Total Positivity and factorization of Wronskian matrices of monomial basis	176
5.4	Total positivity and factorization of Wronskian matrices of exponential polynomials	177
5.5	Total positivity and factorizations of collocation matrices of Jacobi polynomials	179
5.6	Total positivity and factorizations of Wronskian matrices of Jacobi polynomials	181
5.7	Collocation and Wronskian matrices of well known orthogonal bases	183
Part IV: PRESENTATION OF THE LATEST OBTAINED RESULTS		187
6	Total positivity and accurate computations with Wronskian matrices of Bessel and Laguerre polynomials	189
6.1	Introduction	189
6.2	Notations and previous results	190
6.3	Total positivity and factorization of Wronskian matrices of Bessel polynomials	191
6.4	Total positivity and factorization of Wronskian matrices of Laguerre polynomials	194
6.5	Numerical experiments	196
7	Total positivity and accurate computations with Wronskian matrices of Bernstein and related bases	201
7.1	Introduction	201
7.2	Notations and preliminary results	202
7.3	Bidiagonal decomposition of the collocation matrix of a general class of functions	205
7.4	Accurate computations with Wronskian matrices of a general class of functions including Bernstein polynomials	206
7.5	Numerical experiments	216
8	Total positivity and accurate computations with Wronskian matrices of geometric and Poisson bases	219
8.1	Introduction	219
8.2	Notations and preliminary results	220
8.3	Bidiagonal factorization of Wronskian matrices of geometric bases	221
8.4	Bidiagonal factorization of Wronskian matrices of Poisson bases	224
8.5	Numerical experiments	229
Part V: CONCLUSIONS AND FUTURE WORK		233
Conclusiones y trabajo futuro		235
Conclusions and future work		239
Appendix		243
Bibliography		245

Part I

INTRODUCTION AND BACKGROUND

1

Introducción

Esta tesis doctoral se enmarca dentro de la teoría de la Positividad Total. La teoría de la Positividad Total es una materia interdisciplinar que tiene sus orígenes en la década de 1930 a partir del trabajo de F.R. Gantmacher y M.G. Kreinn en relación con vibraciones de sistemas mecánicos. Independientemente, I.J. Schoenberg también desarrolló esta teoría con respecto a la propiedad de disminución de la variación de las matrices. En la década de 1960, S. Karlin publicó varios artículos sobre positividad total, los cuales se refieren principalmente a los núcleos totalmente positivos pero también tratan la versión discreta de las matrices totalmente positivas (véase el artículo [2] de T. Ando donde se presenta una lista muy completa de resultados sobre las matrices totalmente positivas hasta 1986). En los últimos años, varios investigadores de la Universidad de Zaragoza (J. Carnicer, J. Delgado, M. Gasca, E. Mainar, J.M Peña) también han profundizado en el estudio de las matrices totalmente positivas en varias disciplinas. Las matrices totalmente positivas han aparecido en aplicaciones de campos tan diversos como, Teoría de Aproximación, Biología, Economía, Combinatoria, Estadística, Ecuaciones Diferenciales, Mecánica, Diseño Geométrico Asistido por Ordenador o Álgebra Lineal Numérica (ver [58], [33], [2], [37], [27], [93], [23]). En esta tesis nos centraremos en dos de los campos que están relacionados con las matrices totalmente positivas. Por un lado, en Diseño Geométrico Asistido por Ordenador donde la importancia de las matrices totalmente positivas proviene del hecho de que los sistemas normalizados totalmente positivos, cuyas matrices de colocación son totalmente positivas, proporcionan representaciones que preservan la forma [9, 92]. Por otro lado, en Álgebra Lineal Numérica, concretamente en la búsqueda de métodos numéricos adaptados a la estructura de las matrices totalmente positivas con los que podamos realizar cálculos algebraicos con alta precisión relativa.

El Diseño Geométrico Asistido por Ordenador (CAGD) es una disciplina que se ocupa de los métodos matemáticos y computacionales para la descripción de objetos geométricos que surgen en áreas que van desde sistemas de Diseño Asistido por Ordenador (CAD) y sistemas de Fabricación Asistida por Ordenador (CAM) hasta Robótica y Visualización Científica. La representación matemática de curvas y superficies en términos de fórmulas simples no siempre es la más apropiada para su tratamiento con el ordenador. A veces se requiere que los parámetros que intervienen en la definición de estas tengan un significado geométrico. Es frecuente que dichos parámetros correspondan a puntos del espacio que pueden interpretarse en términos de propiedades geométricas de las curvas y superficies representadas. En el caso de las curvas, es frecuente utilizar representaciones paramétricas de la forma

$$\gamma(t) := \sum_{i=0}^n P_i u_i(t), \quad t \in I,$$

donde (u_0, \dots, u_n) es un sistema de funciones linealmente independientes definidas en un intervalo $I =$

$[a, b]$ y los puntos P_0, \dots, P_n se denominan *puntos de control*. El polígono $P_0 \cdots P_n$ cuyos vértices son los puntos de control recibe el nombre de *polígono de control*. Un primer requisito que facilita el tratamiento de las curvas consiste en exigir que las funciones del sistema sean no negativas $u_i(t) \geq 0$, para todo $t \in I$. Decimos que un sistema (u_0, \dots, u_n) está *normalizado* si verifica $\sum_{i=0}^n u_i(t) = 1$, lo que implica que las constantes pertenecen al espacio \mathcal{U} generado por u_0, \dots, u_n . El sistema es *totalmente positivo* (TP) si sus matrices de colocación en la sucesión ordenada de nodos $t_0 < \dots < t_n$ de I

$$M \begin{pmatrix} u_0, \dots, u_n \\ t_0, \dots, t_n \end{pmatrix} := (u_j(t_i))_{i,j=0,\dots,n},$$

son *matrices totalmente positivas*, es decir, todos sus menores son no negativos. Si el sistema es totalmente positivo y normalizado (NTP), entonces la curva γ hereda ciertas propiedades geométricas de su polígono de control y, en consecuencia, imita su forma (véase [9, 92]). Debido a lo cual decimos que las bases NTP proporcionan representaciones que preservan la forma.

La propiedad de la envolvente convexa establece que una curva $\gamma(t) = \sum_{i=0}^n P_i u_i(t)$ siempre se encuentra en la envolvente convexa de su polígono de control $P_0 \cdots P_n$. Es bien conocido que la propiedad de envolvente convexa se cumple si y sólo si el sistema (u_0, \dots, u_n) está normalizado y formado por funciones no negativas. De modo que las bases NTP tienen la propiedad de envolvente convexa. Las bases NTP tienen otra propiedad geométrica interesante que es muy conveniente para los objetivos del diseño y se denomina propiedad de interpolación en los extremos: los puntos inicial y final de la curva y los puntos inicial y final (respectivamente) del polígono de control coinciden. Las propiedades de preservación de forma de las bases NTP provienen de la propiedad de disminución de la variación de sus matrices de colocación. Debido a la propiedad de disminución de la variación de las matrices TP, la monotonicidad o convexidad del polígono de control son heredadas por la curva, y la longitud, la variación angular y el número de inflexiones de la curva están respectivamente delimitadas por las del polígono de control (véase [10], [44]).

La *B-base normalizada* de un espacio dado es una base NTP tal que la matriz del cambio de base de cualquier base NTP con respecto a la B-base normalizada es TP y estocástica. Esta propiedad implica que el polígono de control de una curva con respecto a la B-base normalizada se puede obtener mediante un algoritmo de corte de esquinas a partir del polígono de control de la curva con respecto a cualquier otra base NTP. De este modo, el polígono de control con respecto a la B-base normalizada tiene una forma más próxima a la curva que el polígono de control con respecto a cualquier otra base NTP. Además, la longitud del polígono de control con respecto a la B-base normalizada se encuentra entre la longitud de la curva y la longitud de su polígono de control con respecto a cualquier otra base NTP. Se cumplen propiedades similares para otras propiedades geométricas como la variación angular o el número de inflexiones (véase [92], [10], [9]). Según el razonamiento anterior, una B-base normalizada tiene las propiedades óptimas de preservación de forma entre todas las bases NTP del espacio.

Todos los espacios de funciones de dimensión finita que admiten una base NTP tienen una única B-base normalizada con óptimas propiedades de preservación de forma (véase [9] y Capítulo 4 de [92]). Las B-bases normalizadas juegan un papel relevante en el diseño interactivo de curvas. Uno de los objetivos de esta tesis es encontrar un procedimiento general para obtener nuevos sistemas de funciones con propiedades de preservación de forma o con óptimas propiedades de preservación de forma.

El espacio de los polinomios de grado menor o igual que n definidos en el intervalo $[a, b]$ tiene bases NTP. La base de Bernstein definida por

$$B_i^n(t) := \binom{n}{i} \left(\frac{t-a}{b-a} \right)^{n-i} \left(\frac{b-t}{b-a} \right)^i, \quad i = 0, \dots, n,$$

es la B-base normalizada de este espacio. Las curvas definidas paramétricamente mediante la base de Bernstein, llamadas *curvas de Bézier*, son de gran interés en CAGD, ya que proporcionan la representación de curvas polinómicas con óptimas propiedades de preservación de forma. La teoría matemática sobre las curvas de Bézier surgió en la década de 1960. Las curvas de Bézier fueron desarrolladas independientemente por P. de Casteljaou en Citroën y por P. Bézier en Renault. Alrededor de 1970, A.R. Forrest descubrió la conexión entre las curvas de Bézier y la base de los polinomios de Bernstein.

La base de los polinomios de Bernstein puede obtenerse mediante fórmulas de recurrencia que permiten deducir el algoritmo de de Casteljaou para la evaluación de curvas. El algoritmo de de Casteljaou es un algoritmo de corte de esquinas con la propiedad de evaluar la curva de Bézier en un parámetro t_0 (en su dominio de parámetros) a partir de su polígono de control $P_0 \cdots P_n$, y se puede formular de la siguiente forma:

```

Input:  $P_0, P_1, \dots, P_n; t_0$ 
for j :=0 to n
     $P_j^n := P_j$ 
end j
for i:=n-1 to 0 step -1
    for j :=0 to i
         $P_j^i := (1 - t_0)P_j^{i+1} + t_0P_{j+1}^{i+1}$ 
    end j
end i
Output:  $P_0^0$ 

```

La curva Bézier evaluada en t_0 es el punto P_0^0 obtenido al final del algoritmo, es decir, $\gamma(t_0) = P_0^0$. De esta forma el algoritmo de de Casteljaou evalúa la curva Bézier en t_0 y puede utilizarse para calcular los diferentes puntos de la curva (véase Figura 1.1).

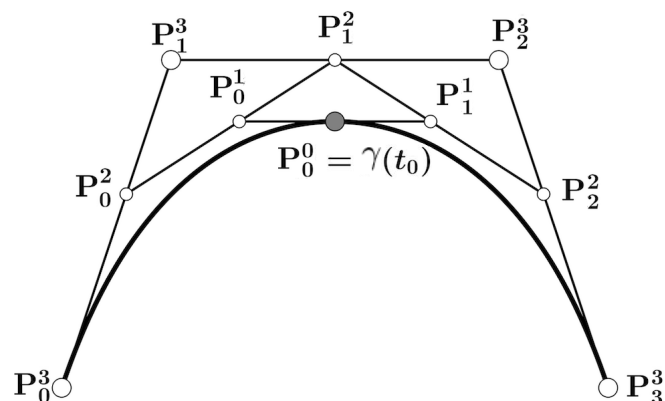


Figure 1.1: Algoritmo de de Casteljaou para la evaluación en $t_0 = 1/2$ de una curva Bézier cúbica.

Otra propiedad del algoritmo de de Casteljaou se puede descifrar a partir de la Figura 1.1: la propiedad de subdivisión. Cuando utilizamos el algoritmo de de Casteljaou para calcular el punto $\gamma(t_0) = P_0^0$, con t_0 en $(0, 1)$, los puntos $P_0^n, P_0^{n-1}, \dots, P_0^0$ forman el polígono de control de la curva $\gamma(t)$ (t en $[0, t_0]$) con

respecto a la base de Bernstein en $[0, t_0]$ (subdivisión izquierda) y los puntos $P_0^0, P_1^1, \dots, P_n^n$ forman el polígono de control de la curva $\gamma(t)$ ($t \in [t_0, 1]$) con respecto a la base de Bernstein en $[t_0, 1]$ (subdivisión derecha). Esta propiedad da lugar a una forma alternativa eficiente de dibujar la curva Bézier. En lugar de calcular los puntos mediante el algoritmo de de Casteljau usando el mismo polígono de control P_0, P_1, \dots, P_n , podemos calcular el punto de la curva correspondiente a t_0 y utilizar los dos polígonos de control correspondientes al subintervalo izquierdo y derecho. Estos polígonos de control juntos aproximan mejor a la curva Bézier que el polígono de control inicial. Podemos repetir el proceso y obtener polígonos de control que convergen a la curva Bézier (véase [13], [38]).

A pesar de su gran sencillez, el algoritmo de de Casteljau es uno de los algoritmos de corte de esquinas más importantes en CAGD. En [70] se demostró que las B-bases normalizadas son las únicas bases que dan lugar a un algoritmo de tipo de Casteljau con la propiedad de subdivisión. Uno de los objetivos de esta tesis es obtener algoritmos de corte de esquinas con las buenas propiedades del algoritmo de de Casteljau para la evaluación y subdivisión de las curvas definidas por las B-bases normalizadas que se proponen.

Dado un sistema de funciones (u_0, \dots, u_n) definido en I y valores positivos d_0, \dots, d_n tal que $\sum_{k=0}^n d_k u_k(t) \neq 0$, para todo $t \in I$, el sistema (r_0, \dots, r_n) definido por

$$r_i(t) := \frac{d_i u_i(t)}{\sum_{k=0}^n d_k u_k(t)}, \quad i = 0, \dots, n, \quad (1.1)$$

satisface $\sum_{i=0}^n r_i(t) = 1$, $\forall t \in I$, y genera un nuevo espacio de funciones. Las curvas definidas por la base racional de Bernstein y las B-spline racionales (NURBS) se han convertido en una herramienta estándar en CAGD, ya que permiten la representación exacta de secciones cónicas, esferas y cilindros. Es bien conocido que las bases obtenidas racionalizando las bases de Bernstein son también las B-bases normalizadas de los espacios generados por funciones racionales. Estos espacios están formados por funciones polinómicas racionales donde el denominador es un polinomio dado.

El primer tema contemplado en esta tesis trata sobre positividad total, CAGD y resolución de problemas algebraicos con alta precisión relativa. En esta tesis, introducimos el concepto de sistema φ -transformado ponderado, que incluye una amplia clase de representaciones útiles en Estadística y CAGD. Dado un sistema de funciones, un conjunto de pesos y una función positiva φ , decimos que $(\tilde{u}_0, \dots, \tilde{u}_n)$ es un sistema φ -transformado ponderado (*weighted φ -transformed system*) a partir de (u_0, \dots, u_n) si

$$\tilde{u}_i(t) := d_i \varphi(t) u_i(t), \quad t \in I, \quad i = 0, \dots, n. \quad (1.2)$$

Por un lado, probamos que este sistema hereda del sistema inicial las propiedades de ser TP, así como la propiedad de ser B-base. Las bases (r_0, \dots, r_n) dadas en (1.1) obtenidas al racionalizar un sistema de funciones, no necesariamente polinómicas, pueden considerarse como casos particulares de sistemas φ -transformados ponderados con $\varphi(t) = \sum_{k=0}^n d_k u_k(t)$, $t \in I$. De estos resultados se puede deducir que si el sistema inicial (u_0, \dots, u_n) es TP, entonces la base racional (r_0, \dots, r_n) es NTP. Además, si el sistema inicial (u_0, \dots, u_n) es una B-base, entonces la base racional (r_0, \dots, r_n) es la B-base normalizada de su espacio generado. Por ejemplo, se prueba que la base racional B-spline (NURBS) es la B-base normalizada de su espacio generado y, por lo tanto, presenta propiedades óptimas de preservación de forma.

En la actualidad, el diseño de curvas mediante funciones trigonométricas e hiperbólicas es cada vez más frecuente. Estas curvas son cada vez más importantes, ya que permiten la representación de cónicas, cilindros, superficies de revolución y catenarias, entre otros. En [95], se han analizado curvas racionales obtenidas a partir de la B-base normalizada de los espacios $\langle 1, \cos t, \sin t, \dots, \cos mt, \sin mt \rangle$ y $\langle 1, \cosh t, \sinh t, \dots, \cosh mt, \sinh mt \rangle$. La aplicabilidad de las citadas bases se ha ilustrado obteniendo los

polígonos de control de curvas y superficies racionales trigonométricas e hiperbólicas. La interpolación con bases racionales trigonométricas con buenas propiedades de preservación de forma es muy importante para la visualización de datos científicos y se ha aplicado en otros campos como, por ejemplo, Ingeniería, Biología, Química, Medicina o Ciencias Sociales (véase [4] y sus referencias bibliográficas).

Como se ha mencionado, las bases racionales pueden ser generadas por los sistemas φ -transformados ponderados. En [70] se analizaron generalizaciones de la base de Bernstein, obtenidas al sustituir los factores lineales por funciones polinómicas, trigonométricas o hiperbólicas. En dicho trabajo también se proporcionan las condiciones que caracterizan a estos sistemas como B-base de su espacio generado. Además, se propone un algoritmo de corte de esquinas que satisface propiedades importantes como la propiedad de evaluación, la propiedad de subdivisión y la convergencia a la curva de los polígonos de control obtenidos. En esta tesis, racionalizamos estos sistemas y, teniendo en cuenta los resultados de [70], construimos un algoritmo de corte de esquinas asociado a esta clase general de bases racionales que satisfacen las dos propiedades mencionadas del algoritmo de Casteljaou: propiedades de evaluación y subdivisión. La propiedad de subdivisión izquierda (respectivamente, derecha) significa que dado $t_0 \in [a, b]$, el algoritmo transforma el polígono de control de la curva paramétrica $\gamma(t)$ con respecto a la clase general de la base racional en $[a, b]$ en el polígono de control con respecto a la clase general de la base racional en $[a, t_0]$ (respectivamente, $[t_0, b]$). Los resultados obtenidos se pueden consultar en el Capítulo 4 o con más detalle en el artículo [73], que se presenta en la página 43. También, se puede encontrar y descargar una aplicación de Matlab en la que se ha implementado el algoritmo evaluación y subdivisión descrito en la siguiente dirección web: <https://github.com/CAGD2020/General>. Este aplicación puede ser muy útil cuando se quiere comparar las curvas generadas por las distintas bases racionales anteriormente propuestas.

En [49], [54] y [97] se analizaron espacios anidados de funciones polinómicas racionales obtenidos al multiplicar sucesivamente el denominador por factores lineales. En particular, en [97] los autores consideran la base racional de Bernstein usando una elección particular de los pesos por la que satisfacen relaciones de recurrencia. Debido a las propiedades de estos pesos, el denominador correspondiente de la base racional de Bernstein tiene la forma $L_1(t) \cdot \dots \cdot L_n(t)$, donde los factores lineales $L_i(t) = a_i(1-t) + b_it$, $i = 1, \dots, n$, están definidos por los coeficientes reales a_i y b_i , $(a_i, b_i) \neq (0, 0)$. La permutación de todos los factores lineales define $n!$ diferentes algoritmos de tipo de Casteljaou para la evaluación de las correspondientes curvas racionales de Bézier. En dicho trabajo también se plantea la necesidad de investigar nuevos espacios anidados generados por bases de funciones que satisfacen relaciones de recurrencia y que nos permiten la definición de algoritmos de evaluación con las buenas propiedades del algoritmo de de Casteljaou. En esta tesis, teniendo en cuenta las generalizaciones de la base de Bernstein obtenidas en [70], mostramos que los resultados de [97] pueden extenderse a nuevos espacios anidados de funciones racionales no polinómicas derivando fórmulas de recurrencia para los pesos y para las funciones de la base de estos espacios. Estas relaciones de recurrencia proporcionan algoritmos de evaluación y subdivisión para curvas racionales paramétricas dadas en términos de estas bases racionales. Las curvas generadas por estas bases racionales heredan propiedades geométricas y algoritmos de las curvas de Bézier racionales tradicionales, por lo que pueden considerarse herramientas de modelado en sistemas CAD y CAM. Los resultados obtenidos se pueden consultar en el Capítulo 4 o con más detalle en el artículo [73], que se presenta en la página 43. También se puede encontrar y descargar una aplicación de Matlab en la que se ha implementado el algoritmo evaluación y subdivisión descrito en la siguiente dirección web <https://github.com/CAGD2020/Particular>. Este aplicación puede ser muy útil cuando se quiere comparar las curvas generadas por las distintas bases racionales anteriormente propuestas.

Por otro lado, en esta tesis también demostramos que los sistemas φ -transformados ponderados

$(\tilde{u}_0, \dots, \tilde{u}_n)$ dados en (1.2) heredan del sistema inicial su comportamiento al realizar operaciones algebraicas con sus matrices de colocación. Un algoritmo se puede calcular con alta precisión relativa (HRA) si solamente usa productos, cocientes, sumas de números con el mismo signo, restas de números con signo opuesto o restas de datos iniciales (cf. [23]). Obtener un algoritmo que pueda realizarse con HRA es un objetivo muy deseable, ya que implica que los errores relativos cometidos en los cálculos son del mismo orden que la precisión del procesador empleado e independientes del condicionamiento del problema. Obtener algoritmos que puedan realizarse con HRA es una labor muy complicada. Sin embargo, en los últimos años, se han conseguido algunos avances en el campo del Álgebra lineal numérica para ciertas matrices TP. Las factorizaciones bidiagonales han jugado un papel crucial para obtener algoritmos con HRA para matrices TP. La parametrización de las matrices TP con las que se deducen los algoritmos HRA es proporcionada por sus factorizaciones bidiagonales, que a su vez están estrechamente relacionadas con un procedimiento de eliminación conocido como eliminación de Neville. La eliminación de Neville es un procedimiento que hace ceros en una columna de una matriz añadiendo a cada fila un múltiplo apropiado de la fila anterior y que fue ya utilizado en algunos de los primeros artículos sobre matrices TP. Sin embargo, en trabajos posteriores como [34] y [37], se desarrolló un mejor conocimiento de las propiedades de la eliminación de Neville, lo que permitió mejorar muchos de los resultados previos sobre matrices TP. Algunos de estos últimos resultados muestran que la factorización bidiagonal de una matriz TP y STP no singular puede ser caracterizada en términos de los multiplicadores y los pivotes diagonales de la eliminación de Neville (este resultado se deduce de los teoremas 4.1 y 4.2 de y p. 116 de [37]).

En [62] se demostró que, dada la factorización bidiagonal en términos de los multiplicadores y los pivotes diagonales de la eliminación de Neville con HRA de una matriz A TP no singular, podemos utilizar los algoritmos presentados en [62, 63] para calcular con HRA sus valores propios y singulares, la matriz A^{-1} , o las soluciones de sistemas de ecuaciones lineales $Ax = b$ tales que las componentes del vector b tienen signos alternos. Hasta ahora, esto se ha logrado con algunas subclases relevantes de matrices TP que presentan aplicaciones en CAGD (cf. [83, 85, 17, 15, 71]), Finanzas (cf. [18]) o Combinatoria (cf. [16]). En esta tesis extendemos el análisis de algunos de los trabajos citados a un contexto mucho más general y demostramos que las citadas operaciones algebraicas con las matrices de colocación de los sistemas φ -transformados ponderados pueden realizarse con HRA, si la factorización en producto de bidiagonales de la correspondiente matriz de colocación del sistema inicial puede obtenerse con HRA y la evaluación de la función φ no requiere restas de valores con el mismo signo, distintos a los parámetros iniciales. Los ejemplos numéricos ilustrarán que la solución de los citados sistemas lineales y el cálculo de valores propios y valores singulares de las matrices de colocación consideradas se pueden resolver con precisión incluso cuando las condiciones anteriores no se cumplen. En particular, los resultados obtenidos se pueden aplicar para realizar interpolación con alta precisión. Los resultados obtenidos se pueden consultar en el Capítulo 4 o con más detalle en el artículo [74], que se presenta en la página 59. Además, el código Matlab con factorizaciones bidiagonales de distintos sistemas φ -transformados ponderados se puede encontrar y descargar en la siguiente dirección web <https://github.com/NLA2020>.

El segundo tema considerado en esta tesis trata sobre positividad total, ajuste de curvas (*curve fitting*) y redes neuronales artificiales (*neural networks*). El problema de obtener una curva que se ajuste a un conjunto dado de puntos es uno de los desafíos más importantes de CAGD, y se ha vuelto prevalente en varios dominios aplicados e industriales, como los sistemas CAD y CAM, Gráficos y Animación por Ordenador, Diseño Robótico, Medicina, entre otros muchos. Algunos trabajos abordaron este problema utilizando las curvas de Bézier ([84], [72], [68]) y, aunque obtuvieron buenos resultados, este enfoque polinómico es algo limitado, ya que no puede describir adecuadamente algunas formas (como las cóni-

cas). Una extensión interesante al respecto la dan las bases racionales. Otro objetivo de esta tesis es abordar esta problemática con la clase general de bases racionales propuestas.

Es bien conocido que los pesos de las bases racionales pueden usarse como parámetros con los que dar forma a las curvas racionales generadas por estas bases. Sin embargo, el efecto de cambiar un peso de la base racional es diferente al de mover un punto de control de la curva. En consecuencia, el control interactivo de la forma de las curvas racionales mediante el ajuste de pesos no es una tarea sencilla, lo que hace que no sea fácil diseñar algoritmos con los que obtener los pesos apropiados (véase [28], Capítulo 13). Algunos trabajos recientes han demostrado que la aplicación de técnicas de Inteligencia Artificial (IA) puede lograr resultados notables con respecto a este problema. Para afrontar esta cuestión, en [56], se aplicó un algoritmo bioinspirado mediante el uso de curvas racionales de Bézier. Además, en [89] y [98], se aplicaron algoritmos evolutivos a curvas racionales B-spline. Con el fin de abordar este problema utilizando la clase general de bases racionales propuestas, hemos colaborado con un grupo de investigación del Departamento de Matemática Aplicada de la Universidad de Sevilla cuya investigación se centra en Redes Neuronales y Topología.

El algoritmo AdamMax es un algoritmo reciente cuya finalidad es cambiar de manera iterada los distintos parámetros del modelo definido en búsqueda de mínimos locales mediante un proceso estocástico de optimización (véase [59]). Como novedad, en esta tesis, definimos una red neuronal de una capa oculta con la que abordamos el problema de encontrar la curva que mejor ajuste a un conjunto de puntos dado, utilizando la clase general de bases racionales con óptimas propiedades de preservación de forma propuestas en esta tesis. La red neuronal es entrenada a través de un proceso de aprendizaje estocástico reciente, el algoritmo AdaMax, y tiene la finalidad de encontrar los pesos y los puntos de control adecuados de la curva de ajuste. En este proceso de aproximación, la base racional es un hiperparámetro y se puede cambiar sustituyendo los factores lineales por funciones polinómicas, trigonométricas o hiperbólicas, pudiendo así alcanzar formas más difíciles y ampliando de esta manera el rango potencial de aplicaciones de esta red neuronal. Aplicamos la red neuronal a diferentes conjuntos de puntos y mostramos que las curvas de ajuste generadas con este método alcanzan una aproximación satisfactoria. Los resultados obtenidos se pueden consultar en el Capítulo 4 o con más detalle en el artículo [39], que se presenta en la página 77. Además, el código de la experimentación se puede encontrar y descargar en la siguiente dirección web <https://github.com/Mathematics2020>.

El tercer y último tema examinado en esta tesis trata sobre positividad total y resolución con HRA de problemas algebraicos con matrices wronskianas. Dada una base (u_0, \dots, u_n) de un espacio de funciones n -veces continuamente diferenciables, definidas en un intervalo real y $x \in I$, la *matriz wronskiana* en x está definida por

$$W(u_0, \dots, u_n)(x) := (u_{j-1}^{(i-1)}(x))_{i,j=1,\dots,n+1}.$$

En [51] se muestran aplicaciones de las matrices wronskianas del sistema (v_0, \dots, v_n) de funciones definidas por $v_i(t) = t^i e^{\lambda_k t}$, $k = 1, \dots, s$, $i = 0, \dots, m_{k-1}$. En particular, se muestra cómo las matrices wronskianas de esta secuencia de funciones aparecen de forma natural en campos como la teoría espectral de las matrices, la controlabilidad y la teoría de la estabilidad de Lyapunov.

Como se ha mencionado anteriormente, un objetivo importante en matemáticas computacionales es encontrar algoritmos con HRA con los que realizar cálculos matriciales. Además hemos visto que entre las subclases de las matrices TP para las que se ha obtenido la factorización bidiagonal con HRA (cf. [15], [17], [82], [86]) hay muchos ejemplos de matrices de colocación $(u_{j-1}(t_i))_{1 \leq i, j \leq n+1}$ de sistemas (u_0, \dots, u_n) de funciones definidas en un subconjunto real I ($t_1 < t_2 < \dots < t_{n+1}$ en I). Sin embargo, hasta ahora, no hay ejemplos de cálculos precisos con matrices que involucren derivadas de las funciones de la base. Un objetivo adicional de esta tesis es obtener algoritmos con los que realizar cálculos algebraicos precisos con matrices wronskianas que tienen aplicaciones en CAGD y que también pueden surgir en

problemas de interpolación de Hermite, en particular en problemas de interpolación de Taylor.

En esta tesis también demostramos que la matriz wronskiana de la base de los monomios $(1, x, \dots, x^n)$ es TP en $x > 0$ y mostramos que su factorización bidiagonal se puede realizar con HRA. Adicionalmente, probamos que la matriz wronskiana de la base de $(e^{\lambda_0 x}, e^{\lambda_1 x}, \dots, e^{\lambda_n x})$ polinomios exponenciales es STP si $0 < \lambda_0 < \lambda_1 < \dots < \lambda_n$ para todos los $x \in \mathbb{R}$ y proporcionamos su factorización bidiagonal. El cálculo con HRA de esta factorización debería requerir la evaluación con HRA de las funciones exponenciales involucradas. Aunque esto no se puede garantizar, los experimentos numéricos muestran una precisión similar a la obtenida para la base de los monomios. También se prueba que los algoritmos mencionados pueden usarse para realizar con precisión cálculos algebraicos con las matrices wronskianas de la base de los monomios y la base de los polinomios exponenciales. Los ejemplos numéricos muestran que se pueden realizar con alta precisión el cálculo de sus inversas, sus valores propios o singulares y las soluciones de sistemas lineales $Wx = b$ tales que las componentes del vector b tienen signo alterno. Los resultados obtenidos se pueden ver en el Capítulo 5 o con más detalle en el artículo [75], que se presenta en la página 99. Además, el código con la experimentación numérica se puede encontrar y descargar en la siguiente dirección web: <https://github.com/Calcolo2020>.

Los polinomios de Jacobi $J_n^{(\alpha, \beta)}(x)$ forman una clase de polinomios ortogonales clásicos que incluye muchas familias importantes de polinomios ortogonales, como los polinomios de Legendre y Chebyshev. De hecho, los polinomios de Jacobi son ortogonales con respecto al peso $(1-x)^\alpha(1+x)^\beta$ en el intervalo $[-1, 1]$ y presentan múltiples aplicaciones. Por ejemplo, en la Teoría de Aproximación, en la Cuadratura Gaussiana para calcular numéricamente integrales, en Ecuaciones Diferenciales o en Física Aplicada (cf. [5], [64]). En esta tesis, probamos la positividad total estricta de las matrices de colocación de los polinomios de Jacobi en $(1, \infty)$, así como la positividad total de sus matrices wronskianas. También obtenemos un método preciso para construir la factorización bidiagonal de dichas matrices y lo usamos con los algoritmos presentados en [63] para calcular con HRA sus inversas, sus valores propios, sus valores singulares y las soluciones de algunos sistemas lineales. Además, consideramos los casos particulares de las matrices de colocación y wronskianas de los polinomios de Legendre, los polinomios de Gegenbauer, los polinomios de Chebyshev de primer y segundo tipo y los polinomios racionales de Jacobi. Asimismo, mostramos resultados numéricos que confirman la alta precisión en dichos cálculos algebraicos. Los resultados obtenidos se pueden consultar en el Capítulo 5 o con más detalle en el artículo [76], que se presenta en la página 117. Además, el código con la experimentación numérica se puede encontrar y descargar en la siguiente dirección web: <https://github.com/JSC2021>.

Después de una parametrización adecuada de las matrices, se ha logrado el objetivo de encontrar algoritmos con HRA con los que se pueden realizar cálculos algebraicos con las matrices de colocación de los polinomios de Bessel (véase aplicaciones en [47] y en sus referencias bibliográficas) y las matrices de colocación de los polinomios de Laguerre generalizados. En ambos casos, se demostró que las matrices de colocación son TP y se obtuvo su factorización bidiagonal con HRA (véase [21], [20]). En esta tesis, demostramos que la matriz wronskiana de los polinomios de Bessel es TP para todo $x > 0$ y también probamos que su factorización bidiagonal se puede calcular con HRA. Aunque confirmamos que la matriz wronskiana de los polinomios de Laguerre generalizados no es TP, obtenemos una factorización bidiagonal de esta matriz wronskiana y la usamos para definir algoritmos con los que podemos resolver con HRA diferentes problemas algebraicos. Mostramos también ejemplos numéricos que ilustran la gran precisión de los métodos mencionados en los cálculos algebraicos que se pueden realizar con las matrices wronskianas de los polinomios de Bessel y los polinomios de Laguerre generalizados, como el cálculo de sus inversas, sus valores singulares y las soluciones de algunos sistemas lineales. Los resultados obtenidos se han incluido en el Capítulo 6.

En [82] se demostró que se pueden realizar con HRA diferentes cálculos algebraicos con las matri-

ces de colocación de la base de Bernstein. En esta tesis, vamos a tratar sus matrices wronskianas. En primer lugar, definimos una clase general de bases que incluyen la base de Bernstein y otras bases relacionadas, como la base de Bernstein de grado negativo (ver [41]) o la base binomial negativa. Además, caracterizamos cuando las matrices wronskianas de estas bases generales son TP. Una primera dificultad encontrada para obtener algoritmos precisos con los que realizar cálculos algebraicos con las matrices wronskianas de la base de Bernstein, la base de Bernstein de grado negativo o la base binomial negativa proviene del hecho de que estas matrices nunca son TP. Sin embargo, a pesar de que no son TP, hemos obtenido una factorización bidiagonal de estas matrices wronskianas y la hemos utilizado para derivar algoritmos para calcular con HRA sus valores propios y singulares, sus inversas y la solución de algunos sistemas lineales. Además, mostramos resultados numéricos que confirman la alta precisión en dichos cálculos algebraicos. Los resultados obtenidos se pueden consultar en el Capítulo 7.

La distribución geométrica tiene aplicaciones en modelos poblacionales y econométricos, y la distribución de Poisson es popular por modelar el número de veces que ocurre un evento en un intervalo de tiempo o espacio. Asociadas a estas distribuciones, se pueden definir las bases correspondientes. La base de Poisson también juega un papel útil en CAGD (véase [42] y [90]). En esta tesis, mostramos que las matrices wronskianas de la base geométrica y las matrices wronskianas de la base de Poisson no son TP. Sin embargo, las relacionamos con otras matrices TP, de modo que sus factorizaciones bidiagonales asociadas se pueden utilizar para proporcionar algoritmos precisos con los que calcular sus valores propios o valores singulares, sus inversas o la solución de algunos sistemas lineales. Asimismo, mostramos experimentos numéricos que confirman los resultados teóricos obtenidos. Los citados resultados y los experimentos numéricos realizados se pueden consultar en el Capítulo 8.

La complejidad de todos los algoritmos propuestos para resolver los problemas algebraicos mencionados es comparable a la de los algoritmos LAPACK tradicionales, los cuales, como ilustraremos, no ofrecen tal precisión.

Esta memoria se estructura en cinco partes de la siguiente forma. La primera parte está compuesta por esta Introducción y por el Capítulo 3. En el Capítulo 3, presentamos notaciones matriciales y conceptos básicos relacionados con la teoría de la positividad total, CAGD y HRA. También se incluyen los resultados auxiliares y las herramientas que vamos a emplear en el desarrollo del trabajo. En la segunda parte, presentamos en las páginas 43, 59, 77, 99 y 117 los artículos [73], [74], [39], [75] y [76] que pertenecen al compendio de las publicaciones de esta tesis. En la tercera parte, justificamos la unidad temática de las publicaciones y presentamos los principales resultados obtenidos en estos artículos. Más específicamente, en el Capítulo 4 definimos los sistemas φ -transformados ponderados e incluimos los resultados que garantizan sus buenas propiedades geométricas y computacionales. En particular, mostramos la factorización bidiagonal de las matrices de colocación de los sistemas φ -transformados ponderados. Además, presentamos algoritmos de evaluación y subdivisión para una clase general de bases racionales, las cuales pueden considerarse como un caso particular de los sistemas φ -transformados ponderados, y proporcionamos su factorización bidiagonal con HRA. Al final de este capítulo presentamos un método de aprendizaje para encontrar la curva que mejor se ajuste a un conjunto de puntos dado, utilizando técnicas de Inteligencia Artificial con la clase general de bases racionales propuesta. En el Capítulo 5, proporcionamos algoritmos con los que calcular la factorización bidiagonal de las matrices wronskianas de la base de los monomios y la factorización bidiagonal de la base de los polinomios exponenciales. También, mostramos que estos algoritmos pueden usarse para realizar con precisión algunos cálculos algebraicos con estas matrices wronskianas. Además, obtenemos un método preciso para construir la factorización bidiagonal de las matrices de colocación y wronskianas de los polinomios de Jacobi y de sus polinomios relacionados. Usamos este método para calcular con HRA sus valores propios y valores singulares, inversas y la solución de algunos sistemas lineales. En la

cuarta parte, presentamos los últimos resultados obtenidos, los cuales no están incluidos en los artículos que pertenecen a el compendio de publicaciones. En particular, en el Capítulo 6, proporcionamos un método para obtener la factorización bidiagonal de las matrices wronskianas de los polinomios de Bessel y la factorización bidiagonal de los polinomios de Laguerre. Este método se puede utilizar para calcular con HRA algunos cálculos algebraicos con estas matrices wronskianas. En el Capítulo 7, proporcionamos una factorización bidiagonal de las matrices wronskianas de la base de los polinomios de Bernstein y factorizaciones bidiagonales de otras bases relacionadas, como la base de Bernstein de grado negativo o la base binomial negativa. También mostramos que estas factorizaciones pueden usarse para realizar con HRA algunos cálculos algebraicos con estas matrices wronskianas. En el Capítulo 8, proporcionamos una factorización bidiagonal de las matrices wronskianas de la base geométrica y de la base de Poisson. Además, mostramos que con estas factorizaciones podemos realizar con HRA diferentes cálculos algebraicos con estas matrices wronskianas. Finalmente, en la quinta parte, se describen las conclusiones y el posible trabajo futuro que puede continuar desarrollándose como resultado de la investigación de esta tesis.

2

Introduction

This doctoral thesis is framed within the theory of Total Positivity. The theory of Total Positivity is an interdisciplinary area that has its origins in the 1930s from the work of F.R. Gantmacher and M.G. Krein in connection with vibrations of mechanical systems. Independently, I.J. Schoenberg also developed this theory in regard to the variation diminishing property of matrices. In the 1960s, S. Karlin published several papers on total positivity, which mostly concerns totally positive kernels but also treats the discrete version of totally positive matrices (see the paper [2] by T. Ando, which presents a very complete list of results on total positivity matrices until 1986). In recent years, several researchers from the University of Zaragoza (J. Carnicer, J. Delgado, M. Gasca, E. Mainar, J.M Peña) have also delved into the study of totally positive matrices in several disciplines. Totally positive matrices present important applications in many fields, such as Approximation Theory, Biology, Economics, Combinatorics, Statistics, Differential Equations, Mechanics, Computer-Aided Geometric Design or Numerical Linear Algebra (see [58], [33], [2], [37], [27], [93], [23]). In this thesis, we will focus on two fields related to totally positive matrices. On the one hand, on applications in Computer-Aided Geometric Design where the importance of totally positive matrices comes from the fact that the normalized totally positive systems, whose collocation matrices are totally positive, provide shape preserving representations [9, 92]. On the other hand, on applications related to the search of numerical methods adapted to the structure of the totally positive matrices in order to perform with high relative accuracy algebraic computations with these matrices.

Computer-Aided Geometric Design (CAGD) is a discipline that deals with mathematical and computational methods for the description of geometric objects that arise in areas ranging from Computer-Aided Design (CAD) systems and Computer-Aided Manufacturing (CAM) systems to Robotics and Scientific Visualization. The mathematical representation of curves and surfaces in terms of simple formulas is not always the most appropriate for their treatment with the computer. Sometimes it is required that the parameters involved in the definition have a geometric meaning. It is common that these parameters correspond to points in the space that can be interpreted in terms of geometric properties of the curves and surfaces represented. In the case of curves, it is usual to use parametric representations of the form

$$\gamma(t) := \sum_{i=0}^n P_i u_i(t), \quad t \in I,$$

where (u_0, \dots, u_n) is a system of linearly independent functions defined on $I = [a, b]$ and the points P_0, \dots, P_n are called *control points*. The polygon $P_0 \cdots P_n$ whose vertices are the control points is called *control polygon*.

A first requirement for the treatment of curves is that the functions of the system are nonnega-

tive $u_i(t) \geq 0$, for all $t \in I$. We say that a system of functions (u_0, \dots, u_n) is *normalized* if it satisfies $\sum_{i=0}^n u_i(t) = 1$. This implies that the constants belong to the space \mathcal{U} generated by u_0, \dots, u_n . The system is *totally positive* (TP) if its collocation matrices in the ordered sequence of nodes $t_0 < \dots < t_n$ in I

$$M \begin{pmatrix} u_0, \dots, u_n \\ t_0, \dots, t_n \end{pmatrix} := (u_j(t_i))_{i,j=0,\dots,n},$$

are *totally positive matrices*, that is, all of their minors are nonnegative. If the system is totally positive and normalized (NTP), the curve γ inherits certain geometric properties from its control polygon and, consequently, mimics its shape (see [9],[92]). Thus, NTP bases provide shape preserving representations.

The convex hull property states that a curve $\gamma(t) = \sum_{i=0}^n P_i u_i(t)$ always lies in the convex hull of its control polygon $P_0 \cdots P_n$. It is well known that the convex hull property holds if and only if the system (u_0, \dots, u_n) is normalized and formed by nonnegative functions. Therefore, NTP bases have the convex hull property. NTP bases have another interesting geometric property that is very convenient for the design purposes, which is called endpoint interpolation property: the initial and final endpoints of the curve and the initial and final endpoints (respectively) of the control polygon concur. Shape preserving properties of NTP bases come from the variation diminishing property of their collocation matrices. Due to the variation diminishing property of TP matrices, the monotonicity or convexity of the control polygon are inherited by the curve, and the length, angular variation and number of inflections of the curve are respectively bounded by those of the control polygon (see [10], [44]).

The *normalized B-basis* of a given space is an NTP basis such that the matrix of change of basis of any NTP basis with respect to the normalized B-basis is TP and stochastic. This property implies that the control polygon of a curve with respect to the normalized B-basis can be obtained by a corner cutting algorithm from its control polygon with respect to any other NTP basis. Thus, the control polygon with respect to the normalized B-basis is closer in shape to the curve than the control polygon with respect to any other NTP basis. Furthermore, the length of the control polygon with respect to the normalized B-basis lies between the length of the curve and the length of its control polygon with respect to any other NTP basis. Similar properties hold for other geometric properties such as angular variation or number of inflections (see [92], [10], [9]). By the previous reasoning, a normalized B-basis has the optimal shape preserving properties among all NTP bases of the space.

All finite dimensional spaces of functions generated by a NTP basis have a unique normalized B-basis with optimal shape preserving properties (see [9] and Chapter 4 of [92]). The normalized B-bases play a relevant role in the interactive design of curves. One of the objectives of this thesis is to find a general procedure for generating new systems of functions with shape preserving properties or optimal shape preserving properties.

The space of polynomials of degree less than or equal to n defined on the interval $[a, b]$ has NTP bases. The Bernstein basis defined by

$$B_i^n(t) := \binom{n}{i} \left(\frac{t-a}{b-a} \right)^{n-i} \left(\frac{b-t}{b-a} \right)^i, \quad i = 0, \dots, n,$$

is the normalized B-basis of this space. Curves parametrically defined by the Bernstein basis, called *Bézier curves*, are of great interest in CAGD since they provide the representation of polynomial curves with optimal shape preserving properties. The mathematical theory regarding Bézier curves arose in the 1960s. Bézier curves were independently developed by P. de Casteljaou at Citroën and by P. Bézier at Renault. Around 1970, A.R. Forrest discovered the connection between the Bézier curves and the Bernstein polynomial basis.

The Bernstein polynomial basis can be obtained by means of recurrence relations that allow us to deduce the de Casteljau algorithm for the evaluation of curves. The de Casteljau algorithm is a corner cutting algorithm with the property of evaluating the Bézier curve in a parameter t_0 (in its parameter domain) from its control polygon $P_0 \cdots P_n$, and can be formulated as follows:

```

Input:  $P_0, P_1, \dots, P_n; t_0$ 
for j :=0 to n
     $P_j^1 := P_j$ 
end j
for i:=n-1 to 0 step -1
    for j :=0 to i
         $P_j^i := (1 - t_0)P_j^{i+1} + t_0P_{j+1}^{i+1}$ 
    end j
end i
Output:  $P_0^0$ 

```

The Bézier curve evaluated in t_0 is the point P_0^0 obtained at the end of the algorithm; that is $\gamma(t_0) = P_0^0$. Thus, the de Casteljau algorithm evaluates the Bézier curve in t_0 and this algorithm can be used to compute many points of the curve and then draw the curve (see Figure 2.1).

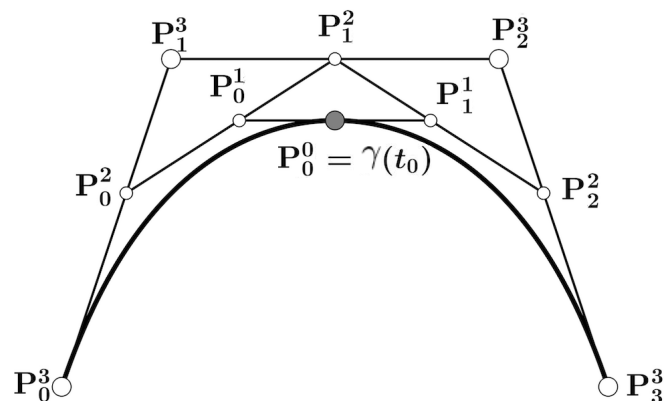


Figure 2.1: De Casteljau's algorithm for the evaluation in $t_0 = 1/2$ of a cubic Bézier curve.

Another property of the de Casteljau algorithm can be guessed from Figure 2.1: the subdivision property. When we use the de Casteljau algorithm to compute the point $\gamma(t_0) = P_0^0$, for t_0 on $(0, 1)$, the points $P_0^n, P_0^{n-1}, \dots, P_0^0$ form the control polygon of the curve $\gamma(t)$ ($t \in [0, t_0]$) with respect to the Bernstein basis on $[0, t_0]$ (left subdivision) and the points $P_0^0, P_1^1, \dots, P_n^n$ form the control polygon of the curve $\gamma(t)$ ($t \in [t_0, 1]$) with respect to the Bernstein basis on $[t_0, 1]$ (right subdivision). This property gives rise to an efficient alternative way to draw the Bézier curve. Instead of computing many points by the de Casteljau algorithm using the same control polygon P_0, P_1, \dots, P_n , we can compute the point of the curve corresponding to t_0 and keep the two control polygons corresponding to the left and right subintervals. These control polygons together are a better approximation to the Bézier curve than the primitive control polygon. We can repeat the process and get a sequence of control polygons that

converge to the Bézier curve (see [13], [38]).

Despite its simplicity, the de Casteljau algorithm is one of the fundamental corner cutting algorithms in CAGD. In [70] it was proved that the normalized B-bases are the only bases that give rise to a Casteljau-type algorithm with the subdivision property. One of the objectives of this thesis is to obtain corner cutting algorithms with the good properties of the de Casteljau algorithm for the evaluation and subdivision of curves defined by the normalized B-bases that are proposed.

Given a system (u_0, \dots, u_n) of functions defined on I and positive values d_0, \dots, d_n such that $\sum_{k=0}^n d_k u_k(t) \neq 0$, for all $t \in I$, the system (r_0, \dots, r_n) defined by

$$r_i(t) := \frac{d_i u_i(t)}{\sum_{k=0}^n d_k u_k(t)}, \quad i = 0, \dots, n, \quad (2.1)$$

satisfies $\sum_{i=0}^n r_i(t) = 1$, $\forall t \in I$, and generates a new space of rational functions. Curves defined by rational Bernstein and rational B-spline bases (NURBS) have become a standard tool in CAGD since they allow the exact representation of conic sections, spheres and cylinders. It is well known that the bases obtained by rationalizing Bernstein bases are also the normalized B-bases of the generated spaces of rational functions. These spaces are made up of rational polynomial functions where the denominator is a given polynomial.

The first topic surveyed in this thesis deals with total positivity, CAGD and solving algebraic problems with high relative accuracy. In this thesis, we introduce the concept of weighted φ -transformed system, which includes a very large class of useful representations in Statistics and CAGD. Given a system (u_0, \dots, u_n) of functions, a set of weights d_0, \dots, d_n and a positive function φ , we say that $(\tilde{u}_0, \dots, \tilde{u}_n)$ is a *weighted φ -transformed system* from (u_0, \dots, u_n) if

$$\tilde{u}_i(t) := d_i \varphi(t) u_i(t), \quad t \in I, \quad i = 0, \dots, n. \quad (2.2)$$

On the one hand, it is proved that this system inherits from the initial system the properties of being TP, as well as the property of being B-basis. The bases (r_0, \dots, r_n) given in (2.1) obtained by rationalizing a system of functions, not necessarily polynomials, can be considered as particular cases of weighted φ -transformed systems with $\varphi(t) = \sum_{k=0}^n d_k u_k(t)$, $t \in I$. From these results, it can be deduced that if the initial system (u_0, \dots, u_n) is TP then the rational basis (r_0, \dots, r_n) is NTP. Furthermore, if the initial system (u_0, \dots, u_n) is a B-basis then the rational basis (r_0, \dots, r_n) is the normalized B-basis of its generated space. For example, it is shown that the rational B-spline basis (NURBS) is the normalized B-basis of its generated space and, therefore, has the optimal shape preserving properties.

Trigonometric and hyperbolic curves are getting considerable importance since they provide the opportunity to construct conics, cylinders, surfaces of revolution and catenaries, among others. In [95], through the normalized B-basis of the spaces $\langle 1, \cos t, \sin t, \dots, \cos mt, \sin mt \rangle$ and $\langle 1, \cosh t, \sinh t, \dots, \cosh mt, \sinh mt \rangle$, rational bases are generated. The applicability of the mentioned bases has been illustrated by obtaining the control polygon of the rational trigonometric and hyperbolic curves, and multivariate surfaces. Shape preserving rational trigonometric and hyperbolic interpolation is very important in Scientific Data Visualization and has been applied to other fields such as Engineering, Biology, Chemistry, Medical and Social Sciences (see [4] and the references therein).

As mentioned before, rational bases can be generated by weighted φ -transformed systems. In [70] generalizations of the Bernstein basis, obtained by substituting the linear factors for polynomial, trigonometric or hyperbolic functions, were analyzed. In that work, the conditions characterizing these systems as a B-basis of its generated space are also provided. Furthermore, a corner cutting algorithm satisfying important properties such as evaluation property, subdivision property and convergence to the curve of the resulting control polygons is proposed. In this thesis, we rationalize these systems and, supported by

the results of [70], we construct a corner cutting algorithm associated to these general class of rational bases satisfying the two mentioned properties of the de Casteljau algorithm: evaluation and subdivision properties. Left (respectively, right) subdivision property means that given $t_0 \in [a, b]$, the algorithm transforms the control polygon of the parametric curve $\gamma(t)$ with respect to the general class of rational basis on $[a, b]$ into the control polygon with respect to the general class of rational basis on $[a, t_0]$ (respectively, $[t_0, b]$). The obtained results can be seen in Chapter 4 or, in more detail, in the article [73], which is presented on page 43. Additionally, a Matlab Application with the implementation of the obtained evaluation and subdivision algorithms can be found and downloaded at the following website: <https://github.com/CAGD2020/General>. This application can be very useful when we want to compare the curves generated by the different rational bases previously proposed.

In [49], [54] and [97] nested spaces of rational polynomial functions obtained by successively multiplying the denominator by linear factors are analyzed. In particular, in [97] the authors consider the rational Bernstein basis using a particular choice of the weights satisfying recurrence relations. Due to the properties of these weights, the denominator of the corresponding rational Bernstein basis has the form $L_1(t) \cdot \dots \cdot L_n(t)$, where the linear factors $L_i(t) = a_i(1-t) + b_it$, $i = 1, \dots, n$, are defined by the real coefficients a_i and b_i , $(a_i, b_i) \neq (0, 0)$. The permutation of all the linear factors defines $n!$ different de Casteljau-type algorithms for the evaluation of the corresponding rational Bézier curves. In that work, it is also suggested the need to investigate new spaces generated by bases of functions, satisfying recurrence relations and allowing us the definition of evaluation algorithms with the good properties of the de Casteljau algorithm. In this thesis, taking into account the generalizations of the Bernstein basis presented in [70], we show that the results from [97] can be extended to new nested spaces of non-polynomial rational functions deriving recurrence formulas for the weights and basis functions of these spaces. These recurrence relations provide evaluation and subdivision algorithms for parametric rational curves given in terms of the considered rational bases. Curves generated by these rational bases inherit geometric properties and algorithms of the traditional rational Bézier curves and so, they can be considered as modeling tools in CAD/CAM systems. The obtained results can be seen in Chapter 4 or, in more detail, in the article [73], which is presented on page 43. Additionally, a Matlab Application with the implementation of the obtained evaluation and subdivision algorithms can be found and downloaded at the following website: <https://github.com/CAGD2020/Particular>. This application can be very useful when we want to compare the curves generated by the different rational bases previously proposed.

On the other hand, we show that weighted ϕ -transformed systems $(\tilde{u}_0, \dots, \tilde{u}_n)$ given in (2.2) inherit from the initial system their behavior when performing algebraic computations with its collocation matrices. An algorithm can be computed with high relative accuracy (HRA) when it only uses products, quotients, sums of numbers of the same sign, subtractions of numbers of opposite sign or subtraction of initial data (cf. [23]). Performing an algorithm with HRA is a very desirable goal. HRA implies that the relative errors of the computations are of the order of the machine precision, independently of the size of the condition number. This goal is difficult to assure, although in recent years there have been some advances, in particular in the field of Numerical Linear Algebra. Bidiagonal factorizations have played a crucial role to derive algorithms with HRA for TP matrices. The parametrization of TP matrices leading to HRA algorithms is provided by their bidiagonal factorizations, which are in turn closely related to an elimination procedure known as Neville elimination. Neville elimination is a procedure to make zeros in a column of a matrix by adding to each row an appropriate multiple of the previous one and it had been already used in some of the first papers on TP matrices. However, in later papers such as [34] and [37], a better knowledge of the properties of Neville elimination was developed allowing to improve many previous results on those matrices. Some of these latest results show that the bidiagonal factorization

of a nonsingular TP and STP matrix can be characterized in terms of the multipliers and the diagonal pivots of the Neville elimination (it follows from theorems 4.1 and 4.2 of and p. 116 of [37]).

In [62] it was shown that given the bidiagonal factorization in terms of the multipliers and the diagonal pivots of the Neville elimination of a nonsingular TP matrix A with HRA, we can perform the algorithms presented in [62, 63] to compute with HRA its eigenvalues and singular values, the matrix A^{-1} and even the solution of $Ax = b$ for vectors b with alternating signs. Up to now, this has been achieved with some relevant subclasses of TP matrices with applications to CAGD (cf. [83, 85, 17, 15, 71]), to Finance (cf. [18]) or to Combinatorics (cf. [16]). In this thesis, we extend the analysis of some of those works to a more general framework and assure that the algebraic computations mentioned above can be performed with HRA for the collocation matrices of weighted φ -transformed systems, assuming that the bidiagonal factorization of the corresponding collocation matrix of the initial system can be obtained with HRA and that the evaluation of φ does not require subtractions up to initial data. The numerical examples will illustrate that the solution of mentioned linear systems and the computation of eigenvalues and singular values with the considered collocation matrices can be solved accurately even when the above conditions do not hold. In particular, the results can be applied to perform interpolation with high precision. The obtained results can be seen in Chapter 4 or, in more detail, in the article [74], which is presented on page 59. Additionally, the code for the experimentation can be found and downloaded at the following website: <https://github.com/NLA2020>.

The second topic considered in this thesis deals with total positivity, curve fitting and neural networks. The problem of obtaining a curve that fits a given set of points is one of the most important challenges of CAGD, and it has become prevalent in various applied and industrial domains, such as CAD/CAM systems, Computer Graphics and Animation, Robotic Design, Medical, among many others. Some works addressed this problem using Bézier curves ([84], [72],[68]) and, although they obtained good results, this polynomial approach is somewhat limited since it cannot adequately describe some shapes (such as conics). An interesting extension in this regard is given by the rational bases. Another objective of this thesis is to approach this problem with the proposed general class of rational bases.

It is well known that the weights of the rational bases can be used as parameters with which to shape the rational curves generated by these bases. However, the effect of changing a weight in a rational basis is different from that of moving a control point of the curve. Thus, the interactive control of the shape of the rational curves by adjusting weights is not an easy task, which makes it difficult to design algorithms with which to obtain the appropriate weights (see [27], Chapter 13). Some recent works have shown that the application of Artificial Intelligence (AI) techniques can achieve remarkable results regarding this problem. To address this question, in [56], a bioinspired algorithm was applied using rational Bézier curves. Furthermore, in [89] and [98], evolutionary algorithms were applied to rational B-spline curves. To tackle this issue by using the general class of rational bases proposed in this thesis, we have collaborated with a research group from the Department of Applied Mathematics of the University of Seville whose research focuses on Neural Networks and Topology.

The AdamMax algorithm is a recent algorithm whose purpose is to iteratively change the different parameters of the defined model in search of local minima using a stochastic optimization process (see [59]). As a novelty, in this thesis, we define a neural network of a hidden layer with which we approach the problem of finding the curve that best fits a given set of points, using the general class of rational bases with optimal shape preservation properties proposed in this thesis. The neural network is trained through a recent stochastic learning process, the AdaMax algorithm, and is intended to find the suitable weights and control points of the fitting curve. In this approximation process, the rational basis is a hyperparameter and can be changed by substituting the linear factors for polynomial, trigonometric or hyperbolic functions, thus being able to reach more difficult shapes and thus expanding the potential

range of applications of this neural network. We apply the neural network to different sets of data points and show that the fitting curves generated with this method reach a satisfactory approximation. The obtained results can be seen in Chapter 4 or, in more detail, in the article [39], which is presented on page 77. Additionally, the code for the experimentation can be found and downloaded at the following website: <https://github.com/Mathematics2020>.

The third and last topic surveyed in this thesis deals with total positivity and resolution with HRA of algebraic problems with Wronskian matrices. For a given basis (u_0, \dots, u_n) of a space of n -times continuously differentiable functions, defined on a real interval I and $x \in I$, the *Wronskian matrix* at x is defined by

$$W(u_0, \dots, u_n)(x) := (u_{j-1}^{(i-1)}(x))_{i,j=1, \dots, n+1}.$$

In [51] applications of the Wronskian matrices of the system (v_0, \dots, v_n) of functions defined by $v_i(t) = t^i e^{\lambda_i t}$, $k = 1, \dots, s$, $i = 0, \dots, m_{k-1}$, are shown. In particular, it is shown how the Wronskian matrices of this sequence of functions appear naturally in fields such as the Spectral Matrix Theory, Controllability and Lyapunov's Stability Theory.

As mentioned before, an important goal in computational mathematics is to find algorithms with HRA to perform matrix computations. Furthermore, we have seen that among the subclasses of the TP matrices for which the bidiagonal factorization has been obtained with HRA (cf. [15], [17], [82], [86]) there are many examples of collocation matrices $(u_{j-1}(t_i))_{1 \leq i, j \leq n+1}$ of systems (u_0, \dots, u_n) of functions defined on a real subset I ($t_1 < t_2 < \dots < t_{n+1}$ in I). However, so far, there are no examples of accurate computations with matrices involving derivatives of the basis functions. An additional objective of this thesis is to obtain algorithms to perform accurately algebraic computations with Wronskian matrices that have applications in CAGD and that can also arise in Hermite interpolation problems, in particular in Taylor interpolation problems.

In this thesis, we also prove that the Wronskian matrix of the monomial basis $(1, x, \dots, x^n)$ of polynomials is TP at $x > 0$ and show that its bidiagonal factorization can be performed with HRA. Additionally, we prove that the Wronskian matrix of the basis $(e^{\lambda_0 x}, e^{\lambda_1 x}, \dots, e^{\lambda_n x})$ of exponential polynomials is STP if $0 < \lambda_0 < \lambda_1 < \dots < \lambda_n$ for all $x \in \mathbb{R}$ and we provide its bidiagonal factorization as well. The computation with HRA of this factorization should require the evaluation with HRA of the involved exponential function. Although this cannot be guaranteed, numerical experiments show an accuracy similar to the obtained for the basis of monomials. It is also proved that the aforementioned algorithms can be used to perform accurate algebraic computations with these Wronskian matrices. The numerical experiments show that the computations of their inverses, their eigenvalues or singular values, and the solutions of linear systems $Wx = b$, for vectors b with alternating signs, can be performed with high accuracy. The obtained results can be seen in Chapter 5 or, in more detail, in the article [75], which is presented on page 99. Also, the code with the numerical experimentation can be found and downloaded at the following website: <https://github.com/Calcolo2021>.

Jacobi polynomials $J_n^{(\alpha, \beta)}(x)$ form a class of classical orthogonal polynomials that includes many important families of orthogonal polynomials, such as Legendre and Chebyshev polynomials. In fact, Jacobi polynomials are orthogonal with respect to the weight $(1-x)^\alpha(1+x)^\beta$ on the interval $[-1, 1]$ and have multiple applications. For instance, in Approximation Theory, in Gaussian Quadrature to numerically compute integrals, in Differential Equations or in Applied Physics (cf. [5], [64]). In this thesis, we prove the strict total positivity of the collocation matrices of Jacobi polynomials on $(1, \infty)$ as well as the total positivity of their Wronskian matrices. We also obtain an accurate method to construct the bidiagonal factorization of these matrices and we use it with the algorithms presented in [63] to compute with HRA their inverses, their eigenvalues, their singular values and the solutions of some linear systems. Moreover, we consider the collocation and Wronskian matrices of Legendre polynomials, Gegenbauer

polynomials, Chebyshev polynomials of the first and second kind, and rational Jacobi polynomials. In addition, we show numerical experiments that confirm the high accuracy in algebraic computations. The obtained results can be seen in Chapter 5 or, in more detail, in the article [76], which is presented on page 117. Also, the code with the numerical experimentation can be found and downloaded at the following website: <https://github.com/JSC2021>.

After a suitable parametrization of the matrices, the goal of finding algorithms with HRA for matrix calculations has been achieved for the collocation matrices of Bessel polynomials (see applications in [47] and references in there) and for the collocation matrices of generalized Laguerre polynomials. In both cases, the collocation matrices are TP and a bidiagonal factorization with HRA was obtained for them (see [21], [20]). In this thesis, we prove that the Wronskian matrix of Bessel polynomials is TP for all $x > 0$ and also show that its bidiagonal factorization can be computed with HRA. Although we confirm that the Wronskian matrix of generalized Laguerre polynomials is not TP, we obtain a bidiagonal factorization of this Wronskian matrix and use it to derive algorithms to compute with HRA many algebraic problems. We show as well numerical examples that illustrate the great accuracy of the mentioned methods in the algebraic calculations that can be performed with the Wronskian matrices of the considered bases, such as the calculation of their inverses, their singular values and the solutions of some linear systems. The obtained results can be seen in Chapter 6.

In [82] it was shown that many algebraic computations with the collocation matrices of the Bernstein basis can be performed with HRA. In this thesis, we are going to treat its Wronskian matrices. Firstly, we define a general class of bases that include the Bernstein basis and other related bases, such as the Bernstein basis of negative degree (see [41]) or the negative binomial basis. Moreover, we characterize when the Wronskian matrices of these general bases are TP. The first difficulty found to obtain accurate algorithms with which to perform algebraic computations with the Wronskian matrices of the Bernstein basis, the Bernstein basis of negative degree, or the negative binomial basis comes from the fact that these matrices are never TP. However, in spite that they are not TP, we have obtained a bidiagonal factorization of these Wronskian matrices and we have used it to derive algorithms to compute with HRA their eigenvalues or singular values, their inverses and the solution of some linear systems. The obtained results can be seen in Chapter 7.

The geometric distribution has applications in population and econometric models and the Poisson distribution is popular for modeling the number of times an event occurs in an interval of time or space. Associated to these distributions, the corresponding bases can be defined. The Poisson basis also plays a useful role in CAGD (see [42] and [90]). In this thesis, we show that the Wronskian matrix of the geometric basis and the Wronskian matrix of the Poisson basis are not TP. However, we relate them with other TP matrices so that their associated bidiagonal factorizations can be used to provide accurate algorithms with which to compute their eigenvalues or singular values, inverses and the solutions of some linear systems. In addition, we show numerical experiments that confirm this accuracy. The obtained results can be seen in Chapter 8.

The complexity of all the proposed algorithms for solving the mentioned algebraic problems is comparable to that of the traditional LAPACK algorithms, which, as we will illustrate, deliver no such accuracy.

This work is structured in five parts in the following way. The first part is composed by this Introduction and Chapter 3. In Chapter 3, we present matrix notations and basic concepts related to the theory of total positivity, CAGD and HRA. The auxiliary results and the tools that we are going to use in the development of the work are also included. In the second part, we present on pages 43, 59, 77, 99 and 117 the articles [73], [74], [39], [75] and [76] which belong to the compendium of publications of this thesis. The third part is composed by Chapter 4 and Chapter 5. The purpose of these chapters is to

justify the thematic unit of the publications and present the main obtained results in these articles. More specifically, in Chapter 4, we define the weighted φ -transformed systems and include the results that guarantee their good geometric and computational properties. In particular, we show the bidiagonal factorization of the collocation matrices of the weighted φ -transformed systems. Furthermore, we present evaluation and subdivision algorithms for a general class of rational bases, which can be considered as a particular case of weighted φ -transformed systems, and we provide their bidiagonal factorization with HRA. At the end of this chapter, we present a learning method to find the curve that best fits a given set of points by using Artificial Intelligence techniques with the proposed general class of rational bases. In Chapter 5, we provide algorithms for computing the bidiagonal factorization of the Wronskian matrices of the monomial basis of polynomials and the bidiagonal factorization of the basis of exponential polynomials. We also show that these algorithms can be used to perform accurately some algebraic computations with these Wronskian matrices. Moreover, we obtain an accurate method to construct the bidiagonal factorization of collocation and Wronskian matrices of Jacobi polynomials and their related polynomials. We use this method to compute with HRA their eigenvalues and singular values, inverses and the solution of some linear systems. In the fourth part, we present the latest obtained results, which are not included in the articles which belong to the compendium of publications. In particular, in Chapter 6, we provide a method to obtain the bidiagonal factorization of the Wronskian matrices of the Bessel polynomials and the bidiagonal factorization of the Laguerre polynomials. This method can be used to compute with HRA some algebraic computations with these Wronskians matrices. In Chapter 7, we provide a bidiagonal factorization of the Wronskian matrices of Bernstein basis of polynomials and bidiagonal factorizations of other related bases, such as the Bernstein basis of negative degree or the negative binomial basis. We also show that these factorizations can be used to perform with HRA some algebraic computations with these Wronskian matrices. At last, in Chapter 8, we provide a bidiagonal factorization of the Wronskian matrices of geometric basis and a bidiagonal factorization of Poisson basis. Besides, we show that these factorizations can be used to perform accurately some algebraic computations with these Wronskian matrices. Finally, in the fifth part, the conclusions and the possible future work that might continue to be developed as a result of the research of this thesis are described.

3

Background

ABOUT THIS CHAPTER

This chapter has an introductory nature and its goal is to include some basic concepts and notations. We introduce the concept of high relative accuracy and totally positive and strictly totally positive matrices, which will be key tools for our work. We recall the Neville elimination process that will be used for factorizing totally positive (strictly totally positive) matrices as product of simpler totally positive (strictly totally positive) matrices. We also explain the corner cutting algorithms, which are profoundly associated with stochastic and totally positive matrices. Finally, we present normalized totally positive bases and their shape preserving properties for the parametric representation of curves. We also introduce B-bases, which have optimal shape preserving properties.

3.1 Error analysis and high relative accuracy

In the study of numerical methods we have to take into account a very important task: the error analysis. In order to carry out the error analysis of an algorithm, we have to set some assumptions about the accuracy of the basic arithmetic operations. These assumptions are mainly embodied in the following model:

$$fl(x \odot y) = (x \odot y)(1 + \delta), \quad |\delta| < u, \quad \odot \in \{+, -, *, /\}, \quad (3.1)$$

where $fl(x \odot y)$ means the result of the operation \odot . The quantity u is called unit roundoff and is the maximum possible relative error consequence of the rounding.

In general, if we consider that our computed solution is the exact solution of a perturbed problem, the backward error measures the distance between the perturbed problem and the initial problem. In contrast, the forward error measures the distance between the computed and the exact solution.

Let us recall that the condition number of a non singular matrix A is

$$\kappa_k(A) = \|A\|_k \|A^{-1}\|_k, \quad (3.2)$$

where usually, $k \in \{1, 2, \infty\}$ and $\|\cdot\|$ is the corresponding matrix norm. Moreover, as $\kappa_k(A) \geq \|A\|_k \|A^{-1}\|_k \geq 1$ thus, a matrix is said well-conditioned when $\kappa_k(A) \rightarrow 1$ and ill-conditioned when $\kappa_k(A) \gg 1$.

The conditioning of the problem measures the effect on the solution of data perturbations. The backward and forward errors are related through the conditioning of the problem by this relation:

$$\text{forward error} \leq \text{condition number} \times \text{backward error}, \quad (3.3)$$

which allows us to obtain a forward error bound through the backward error.

However, in some problems it is possible to find a parametrization of the data and an algorithm leading to small forward error bounds in spite of a bad conditioning with its initial parametrization. The desired goal is to guarantee *high relative accuracy* (HRA). We say that we have performed an algorithm with HRA if the following formula holds:

$$\text{relative forward error} \leq Ku, \quad (3.4)$$

for some constant K , where u is the unit roundoff. It is not always possible to guarantee HRA for a given problem. An example of a simple problem for which a HRA algorithm cannot be found is provided by the sum of three real numbers $x + y + z$ (see [23]). As we shall recall in this chapter, for some structured classes of matrices, HRA algorithms can be found.

A sufficient condition to assure the HRA of an algorithm is the non inaccurate cancellation (NIC) condition and it is satisfied if it only uses additions of numbers of the same sign, multiplications, divisions and subtractions (additions of numbers of different sign) of the initial data (cf. [23]). Performing an algorithm with HRA is a very desirable goal. HRA implies that the relative errors of the computations are of the order of the machine precision, independently of the size of the condition number. This goal is difficult to assure although in recent years there have been some advances, in particular in the field of Numerical Linear Algebra. Up to now, computations with HRA are guaranteed only for a few classes of TP matrices. Previously, a reparametrization of the matrices is needed. Bidiagonal factorizations have played a crucial role to derive algorithms with HRA for TP matrices.

3.2 Totally positive matrices, Neville elimination and computations with high relative accuracy

Let us first recall some basic concepts.

Definition 3.1. A matrix $A = (a_{i,j})_{1 \leq i,j \leq n}$ is *stochastic* if

$$\sum_{j=1}^n a_{i,j} = 1, \quad i = 1, \dots, n.$$

Definition 3.2. A matrix is *totally positive* (TP) if all its minors are nonnegative and it is *strictly totally positive* (STP) if all its minors are positive (see [2]).

It is very easy to check that the bidiagonal matrices

$$A := \begin{pmatrix} a_{1,1} & & & & & & \\ a_{2,1} & a_{2,2} & & & & & \\ & \ddots & \ddots & & & & \\ & & \ddots & \ddots & & & \\ & & & \ddots & \ddots & & \\ & & & & a_{n,n-1} & a_{n,n} & \end{pmatrix}$$

with nonnegative entries are TP. Also, in p.212 of [2], it was proved that the Vandermonde matrix for the nodes $\{x_i\}_{1 \leq i \leq n}$

$$VDM(x_1, \dots, x_n) := \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix}$$

is STP when the nodes satisfy $0 < x_1 < x_2 < \dots < x_n$. Moreover, under these conditions,

$$\det VDM(x_1, \dots, x_n) = \prod_{i < j} (x_j - x_i) > 0.$$

TP matrices are, in particular, sign-regular matrices (matrices whose minors have the same sign). Corollary 5.4 of [2] characterizes the class of sign-regular matrices by the following variation diminishing property.

Proposition 3.1. Let $\alpha = (\alpha_0, \dots, \alpha_m) \in \mathbb{R}^{m+1}$ and $A \in \mathbb{R}^{(n+1) \times (m+1)}$. Then A is sign-regular if and only if

$$V(A\alpha) \leq V(\alpha),$$

where $V(\alpha)$ is the number of strict changes of sign of the components of α .

The following result corresponds to Theorem 3.2 of [2] and proves that the product of TP matrices is also TP. In the sequel we are going to use this property.

Proposition 3.2. Let $A \in \mathbb{R}^{(n+1) \times (m+1)}$ and $B \in \mathbb{R}^{(m+1) \times (p+1)}$. If A and B are TP matrices then the matrix $C := AB$ is TP. Moreover, if A and B are STP matrices then the matrix $C := AB$ is STP.

From Theorem 3.1 of [2], the following property of TP and STP matrices can be proved.

Proposition 3.3. Let $A \in \mathbb{R}^{(n+1) \times (m+1)}$ be a STP matrix and $B \in \mathbb{R}^{(m+1) \times (p+1)}$ a bidiagonal TP matrix. Then the matrix $C := AB$ is STP.

Neville elimination is a particularly important procedure when studying TP matrices. This process can be used to factorize an stochastic and TP matrix and express it as the product of bidiagonal stochastic and TP matrices. As we are going to see in this chapter, this kind of factorization is important in the field of Computer Aided Geometric Design (CAGD), in particular regarding corner cutting algorithms. Moreover, using this factorization, many algebraic problems with STP matrices can be solved with high relative accuracy (HRA).

Let us now recall some basic matrix notations and results on Neville elimination. Our notation follows the notation used in [34, 36].

Given $n \in \mathbb{N}$ and $k \in \{1, \dots, n\}$, let $Q_{k,n}$ be the set of increasing sequences of k positive integers less than or equal to n . If $\alpha, \beta \in Q_{k,n}$, we denote by $A[\alpha|\beta]$ the $k \times k$ submatrix of A containing rows of places α and columns of places β . Besides, $A[\alpha] := A[\alpha|\alpha]$.

Neville elimination (see [34, 36, 37]) is an alternative procedure to Gaussian elimination to make zeros in a column of a matrix by adding to a given row an appropriate multiple of the previous one. For a given nonsingular matrix $A = (a_{i,j})_{1 \leq i,j \leq n}$, this elimination procedure consists of at most $n - 1$ successive major steps, resulting in a sequence of matrices as follows:

$$A^{(1)} := A \rightarrow A^{(2)} \rightarrow \dots \rightarrow A^{(n)} = U.$$

For $1 \leq k \leq n - 1$, the matrix $A^{(k+1)}$ is obtained from $A^{(k)}$ by

$$a_{i,j}^{(k+1)} := \begin{cases} a_{i,j}^{(k)}, & \text{if } 1 \leq i \leq k, \\ a_{i,j}^{(k)} - \frac{a_{i,k}^{(k)}}{a_{i-1,k}^{(k)}} a_{i-1,j}^{(k)}, & \text{if } k+1 \leq i, j \leq n \text{ and } a_{i-1,k}^{(k)} \neq 0, \\ a_{i,j}^{(k)}, & \text{if } k+1 \leq i \leq n \text{ and } a_{i-1,k}^{(k)} = 0, \end{cases}$$

is called a *Cauchy-Vandermonde matrix* for the nodes $\{x_i\}_{1 \leq i \leq n}$ and the poles $\{d_j\}_{1 \leq j \leq l}$. Observe that for $l = 0$ the matrix A is a classical Vandermonde matrix and for $l = n$ the matrix A is a classical Cauchy matrix. Let us also observe that the Cauchy-Vandermonde matrix A is the coefficient matrix of the linear system associated with the following interpolation problem in the basis

$$\mathcal{B} = \{v_i(x)_{1 \leq i \leq n}\} = \left\{ \frac{1}{x-d_1}, \frac{1}{x-d_2}, \dots, \frac{1}{x-d_l}, 1, x, x^2, \dots, x^{n-l-1} \right\}.$$

Given the interpolation nodes x_i , $i = 1, \dots, n$, and the interpolation data b_i , $i = 1, \dots, n$, we find the function

$$f(x) = \sum_{k=1}^n c_k v_k(x)$$

(a rational function with prescribed poles) such that $f(x_i) = b_i$ for $i = 1, \dots, n$.

In [88] it was proved that the Cauchy-Vandermonde matrices are STP when the nodes $\{x_i\}_{1 \leq i \leq n}$ and the poles $\{d_j\}_{1 \leq j \leq l \leq n}$ satisfy $0 < x_1 < x_2 < \dots < x_n$ and $0 < -d_1 < -d_2 < \dots < -d_l$.

For the particular case $n = 4$ and $l = 2$, the Cauchy-Vandermonde matrix (3.8) is

$$A = \begin{pmatrix} \frac{1}{x_1-d_1} & \frac{1}{x_1-d_2} & 1 & x_1 \\ \frac{1}{x_2-d_1} & \frac{1}{x_2-d_2} & 1 & x_2 \\ \frac{1}{x_3-d_1} & \frac{1}{x_3-d_2} & 1 & x_3 \\ \frac{1}{x_4-d_1} & \frac{1}{x_4-d_2} & 1 & x_4 \end{pmatrix}. \quad (3.9)$$

Let us suppose that the nodes satisfy $0 < x_1 < x_2 < x_3 < x_4$, the poles satisfy $0 < -d_1 < -d_2$ and, consequently, the matrix A is STP. Therefore, the Cauchy-Vandermonde matrix (3.9) can be factorized, by (3.7), as follows

$$A = F_3 F_2 F_1 D G_1 G_2 G_3.$$

Using the results of [34], we are going to obtain the three lower triangular bidiagonal matrices: F_1, F_2, F_3 . For the computation of

$$F_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ m_{2,1} & 1 & 0 & 0 \\ 0 & m_{3,2} & 1 & 0 \\ 0 & 0 & m_{4,3} & 1 \end{pmatrix},$$

we have to obtain the multipliers $m_{2,1}, m_{3,2}$ and $m_{4,3}$ of the Neville elimination of A . Using (3.5) and (3.6), we get

$$m_{2,1} = \frac{p_{2,1}}{p_{1,1}} = \frac{a_{2,1}^{(1)}}{a_{1,1}^{(1)}} = \frac{1}{\frac{x_2-d_1}{x_1-d_1}} = \frac{x_1-d_1}{x_2-d_1},$$

$$m_{3,2} = \frac{p_{3,2}}{p_{2,2}} = \frac{\frac{\det A[2,3|1,2]}{\det A[2|1]}}{\frac{\det A[1,2|1,2]}{\det A[1|1]}} = \frac{(x_3-x_2)(x_2-d_1)(x_1-d_2)}{(x_3-d_1)(x_3-d_2)(x_2-x_1)},$$

$$m_{4,3} = \frac{p_{4,3}}{p_{3,3}} = \frac{\frac{\det A[2,3,4|1,2,3]}{\det A[2,3|1,2]}}{\frac{\det A[1,2,3|1,2,3]}{\det A[1,2|1,2]}} = \frac{(x_4-x_2)(x_4-x_3)(x_3-d_1)(x_3-d_2)}{(x_4-d_1)(x_4-d_2)(x_3-x_1)(x_3-x_2)}.$$

Observe that we can guarantee that the multipliers are positive since $0 < -d_1$. For the computation of

$$F_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & m_{3,1} & 1 & 0 \\ 0 & 0 & m_{4,2} & 1 \end{pmatrix}$$

we have to obtain the multipliers $m_{3,1}$ and $m_{4,2}$ of the Neville elimination of A . Using (3.5) and (3.6), we get

$$m_{3,1} = \frac{p_{3,1}}{p_{2,1}} = \frac{a_{3,1}^{(1)}}{a_{2,1}^{(1)}} = \frac{\frac{1}{x_3 - d_1}}{\frac{1}{x_2 - d_1}} = \frac{x_2 - d_1}{x_3 - d_1},$$

$$m_{4,2} = \frac{p_{4,2}}{p_{3,2}} = \frac{\frac{\det A[3,4|1,2]}{\det A[3|1]}}{\frac{\det A[2,3|1,2]}{\det A[2|1]}} = \frac{(x_4 - x_3)(x_3 - d_1)(x_2 - d_2)}{(x_4 - d_1)(x_4 - d_2)(x_3 - x_2)}.$$

Observe that we can guarantee that the multipliers are positive since $0 < -d_1 < -d_2$ and $0 < x_1 < x_2 < x_3 < x_4$.

Finally, for the computation of

$$F_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & m_{4,1} & 1 \end{pmatrix}$$

we have to compute the multiplier $m_{4,1}$ of the Neville elimination of A . Using (3.5) and (3.6), we get

$$m_{4,1} = \frac{p_{4,1}}{p_{3,1}} = \frac{a_{4,1}^{(1)}}{a_{3,1}^{(1)}} = \frac{\frac{1}{x_4 - d_1}}{\frac{1}{x_3 - d_1}} = \frac{x_3 - d_1}{x_4 - d_1}.$$

Observe that we can guarantee that the multipliers are positive since $0 < -d_1 < -d_2$ and $0 < x_1 < x_2 < x_3 < x_4$.

By considering the matrix

$$A^T = \begin{pmatrix} \frac{1}{x_1 - d_1} & \frac{1}{x_2 - d_1} & \frac{1}{x_3 - d_1} & \frac{1}{x_4 - d_1} \\ \frac{1}{x_1 - d_2} & \frac{1}{x_2 - d_2} & \frac{1}{x_3 - d_2} & \frac{1}{x_4 - d_2} \\ 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \end{pmatrix},$$

we can obtain the upper triangular bidiagonal matrices: G_1^T, G_2^T, G_3^T .

For the computation of

$$G_1^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \hat{m}_{2,1} & 1 & 0 & 0 \\ 0 & \hat{m}_{3,2} & 1 & 0 \\ 0 & 0 & \hat{m}_{4,3} & 1 \end{pmatrix},$$

we have to compute the multipliers $\hat{m}_{2,1}$, $\hat{m}_{3,2}$ and $\hat{m}_{4,3}$ of the Neville elimination of A^T . Using (3.5) and (3.6) we get

$$\hat{m}_{2,1} = \frac{\hat{p}_{2,1}}{\hat{p}_{1,1}} = \frac{\frac{1}{x_1 - d_2}}{\frac{1}{x_1 - d_1}} = \frac{x_1 - d_1}{x_1 - d_2},$$

$$\hat{m}_{3,2} = \frac{\hat{p}_{3,2}}{\hat{p}_{2,2}} = \frac{\frac{\det A^T[2,3|1,2]}{\det A^T[2|1]}}{\frac{\det A^T[1,2|1,2]}{\det A^T[1|1]}} = \frac{(x_2 - d_1)(x_1 - d_2)}{(d_1 - d_2)},$$

$$\hat{m}_{4,3} = \frac{\hat{p}_{4,3}}{\hat{p}_{3,3}} = \frac{\frac{\det A^T[2,3,4|1,2,3]}{\det A^T[2,3|1,2]}}{\frac{\det A^T[1,2,3|1,2,3]}{\det A^T[1,2|1,2]}} = x_3 - d_1.$$

For the computation of

$$G_2^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \hat{m}_{3,1} & 1 & 0 \\ 0 & 0 & \hat{m}_{4,2} & 1 \end{pmatrix},$$

we have to compute the multipliers $\hat{m}_{3,1}$ and $\hat{m}_{4,2}$ of the Neville elimination of A^T . Using (3.5) and (3.6), we get

$$\hat{m}_{3,1} = \frac{\hat{p}_{3,1}}{\hat{p}_{2,1}} = \frac{1}{\frac{1}{x_1 - d_2}} = x_1 - d_2,$$

$$\hat{m}_{4,2} = \frac{\hat{p}_{4,2}}{\hat{p}_{3,2}} = \frac{\frac{\det A^T[3,4|1,2]}{\det A^T[3|1]}}{\frac{\det A^T[2,3|1,2]}{\det A^T[2|1]}} = x_2 - d_2.$$

For the computation of

$$G_3^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \hat{m}_{4,1} & 1 \end{pmatrix},$$

we have to compute the multiplier $\hat{m}_{4,1}$ of the Neville elimination of A^T . Using (3.5) and (3.6), we get

$$\hat{m}_{4,1} = \frac{\hat{p}_{4,1}}{\hat{p}_{3,1}} = \frac{x_1}{1} = x_1.$$

Finally, we are going to obtain the diagonal matrix D , whose diagonal entries are the diagonal pivots of the Neville elimination of A ,

$$D = \begin{pmatrix} p_{1,1} & 0 & 0 & 0 \\ 0 & p_{2,2} & 0 & 0 \\ 0 & 0 & p_{3,3} & 0 \\ 0 & 0 & 0 & p_{4,4} \end{pmatrix},$$

$$p_{1,1} = \frac{1}{x_1 - d_1}, \quad p_{2,2} = \frac{(d_1 - d_2)(x_2 - x_1)}{(x_2 - d_1)(x_1 - d_2)(x_2 - d_2)}, \quad p_{3,3} = \frac{(x_3 - x_1)(x_3 - x_2)}{(x_3 - d_1)(x_3 - d_2)},$$

$$p_{4,4} = \frac{\det A[1, 2, 3, 4 | 1, 2, 3, 4]}{\det[1, 2, 3][1, 2, 3]} = \frac{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)}{(x_4 - d_1)(x_4 - d_2)}.$$

Finally, let us observe that we can guarantee that all the multipliers of the Neville elimination of A^T are positive since $0 < -d_1 < -d_2$ and $0 < x_1 < x_2 < x_3 < x_4$. Furthermore, all multipliers of the Neville elimination of A and A^T and pivots of the Neville elimination of A can be computed with HRA.

3.3 Normalized totally positive bases and corner cutting algorithms

Given a basis (u_0, \dots, u_n) of functions defined on $[a, b]$ and $P_0, \dots, P_n \in \mathbb{R}^k$, we can define a parametric curve as

$$\gamma(t) := \sum_{i=0}^n P_i u_i(t), \quad t \in [a, b].$$

The polygon $P_0 \cdots P_n$ formed by the ordered sequence of points $P_i \in \mathbb{R}^k$, $i = 0, \dots, n$, is called the *control polygon of γ* and the points P_i , $i = 0, \dots, n$, are named *control points of γ* with respect to (u_0, \dots, u_n) (see Figure 3.1).

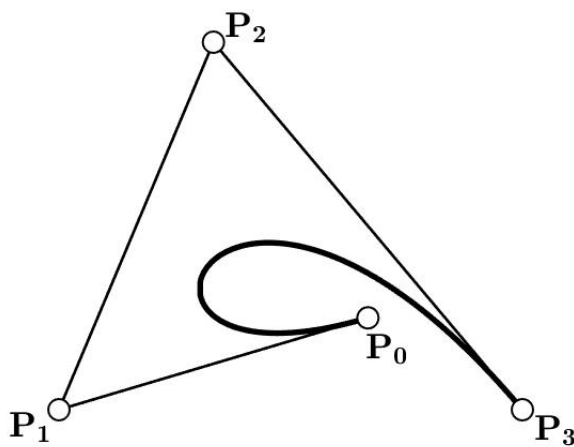


Figure 3.1: Control points of a parametric γ .

Many algorithms for the computation of curves consist on successive alterations of their control polygons. What these algorithms have in common is that the new obtained polygons are made up of successive replacements of two adjacent control points by convex combinations between them. These algorithms are called *corner cutting algorithms* due to their geometric interpretation. A relevant example is the de Casteljaun-type algorithms for the evaluation and subdivision of Bézier curves. Let us now recall some basic aspects of corner cutting algorithms.

An *elementary corner cutting* is a transformation which maps any polygon $P_0 \cdots P_n$ into another polygon $Q_0 \cdots Q_n$ defined by one of the following ways

$$\begin{aligned}
 Q_j &= P_j, \quad j \neq i \\
 Q_i &= (1-\lambda)P_i + \lambda P_{i+1}, \quad \text{for some } i \in 0, \dots, n-1, \quad 0 \neq \lambda < 1,
 \end{aligned}
 \tag{3.10}$$

or

$$\begin{aligned}
 Q_j &= P_j, \quad j \neq i \\
 Q_i &= (1-\lambda)P_i + \lambda P_{i-1}, \quad \text{for some } i \in 1, \dots, n, \quad 0 \neq \lambda < 1.
 \end{aligned}
 \tag{3.11}$$

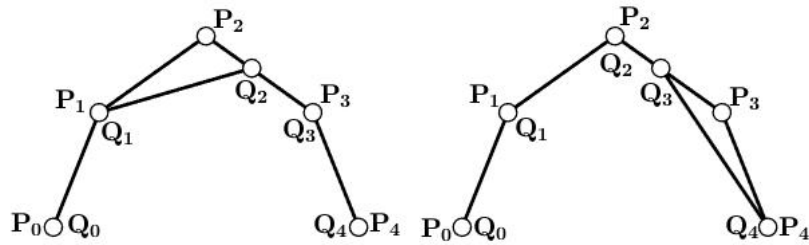


Figure 3.2: Corner cutting algorithms (3.10) and (3.11).

The matrix representation of the elemental geometric transformations (3.10) is

$$\begin{pmatrix} Q_0 \\ \vdots \\ Q_n \end{pmatrix} = U \begin{pmatrix} P_0 \\ \vdots \\ P_n \end{pmatrix},$$

where U is the nonsingular, stochastic, upper triangular and TP matrix of the form

$$U = \begin{pmatrix} 1 & 0 & & & & & \\ & 1 & 0 & & & & \\ & & \ddots & \ddots & & & \\ & & & 1-\lambda & \lambda & & \\ & & & & \ddots & \ddots & \\ & & & & & 1 & 0 \\ & & & & & & 1 & 0 \end{pmatrix}.$$

The matrix representation of the elemental geometric transformations (3.11) is

$$\begin{pmatrix} Q_0 \\ \vdots \\ Q_n \end{pmatrix} = L \begin{pmatrix} P_0 \\ \vdots \\ P_n \end{pmatrix},$$

where L is the nonsingular, stochastic, lower triangular and TP matrix of the form

$$L = \begin{pmatrix} 1 & & & & & & & & \\ 0 & 1 & & & & & & & \\ & \ddots & \ddots & & & & & & \\ & & \lambda & 1 - \lambda & & & & & \\ & & & & \ddots & \ddots & & & \\ & & & & & 0 & 1 & & \\ & & & & & 0 & & 1 & \end{pmatrix}.$$

A *corner cutting algorithm* is any composition of elementary corner cuttings (3.10) and (3.11). Therefore a corner cutting algorithm is described by means of a matrix which is nonsingular, TP and stochastic, as a product of the previous ones (see Proposition 3.3).

Conversely, a nonsingular stochastic TP matrix A can be factorized as $A = LU$, where L (respectively, U) is a nonsingular stochastic lower (respectively, upper) triangular TP matrix (see Theorem 3.5 of [2]). Furthermore, an $(m + 1) \times (m + 1)$ nonsingular lower triangular stochastic matrix L admits a (unique) factorization in terms of bidiagonal stochastic matrices as

$$L = L_{m-1} \cdots L_1 L_0,$$

where

$$L_l = \begin{pmatrix} 1 & & & & & & & & \\ 0 & 1 & & & & & & & \\ & \ddots & \ddots & & & & & & \\ & & 0 & 1 & & & & & \\ & & l_l^{(l)} & 1 - l_l^{(l)} & & & & & \\ & & & & \ddots & \ddots & & & \\ & & & & & l_{m-1}^{(l)} & 1 - l_{m-1}^{(l)} & & \end{pmatrix}, \quad 0 \leq l \leq m - 1,$$

(see Theorem 4.5 of [37]). Analogously, a nonsingular, upper triangular, stochastic matrix U can be factorized in terms of bidiagonal stochastic matrices as

$$U = U_{m-1} \cdots U_1 U_0,$$

where

$$U_l = \begin{pmatrix} u_l^{(l)} & 1 - u_l^{(l)} & & & & & & & \\ & \ddots & \ddots & & & & & & \\ & & u_{m-l}^{(l)} & 1 - u_{m-l}^{(l)} & & & & & \\ & & & 1 & 0 & & & & \\ & & & & \ddots & \ddots & & & \\ & & & & & 1 & 0 & & \\ & & & & & & & & 1 \end{pmatrix}, \quad 0 \leq l \leq m - 1.$$

These factorizations appear naturally from Neville elimination (see [34]) and have an interpretation in terms of corner cutting algorithms.

Now, let us introduce some basic concepts related to the systems used for the parametric representation of curves.

Definition 3.3. Let (u_0, \dots, u_n) be a system of functions defined on the subset $I \subseteq \mathbb{R}$. The *collocation matrix* of the system (u_0, \dots, u_n) at any sequence of points $t_1 < t_2 < \dots < t_{n+1}$ is

$$M \begin{pmatrix} u_0, \dots, u_n \\ t_1, \dots, t_{n+1} \end{pmatrix} := (u_{j-1}(t_i))_{1 \leq i, j \leq n+1}.$$

Taking into account the properties of the collocation matrices of a given system, we can give the following definitions.

Definition 3.4. A system of functions (u_0, \dots, u_n) defined on the subset $I \subseteq \mathbb{R}$ is TP if all its collocation matrices are TP. A TP system of functions on I is normalized (NTP) if it is a partition of the unity, that is,

$$\sum_{i=0}^n u_i(t) = 1, \quad t \in I.$$

Clearly, (u_0, \dots, u_n) is an NTP system of functions if and only if all its collocation matrices are stochastic and TP.

NTP bases are commonly used in CAGD due to their shape preserving properties (see [9], [92]).

The *convex hull property* is an important property for curve design. The convex hull of a polygon $P_0 \cdots P_n$ is the set defined by

$$\mathcal{H}(P_0 \cdots P_n) := \left\{ \sum_{i=0}^n \lambda_i P_i \mid \lambda_i \geq 0, \quad i = 0, \dots, n, \quad \sum_{i=0}^n \lambda_i = 1 \right\}.$$

We say that a system (u_0, \dots, u_n) has the convex hull property if any parametric curve

$$\gamma(t) = \sum_{i=0}^n P_i u_i(t), \quad t \in [a, b],$$

lies in the convex hull of $P_0 \cdots P_n$. In order to illustrate this property, we have implemented a Geogebra application that can be found in this url <https://ggbm.at/AFMCyøKB>. Clearly, the convex hull property holds if and only if the system is a partition of the unity.

In order to be able to guide the curve and put together several pieces of curves, it is desirable for the designer to have precise control over what happens at the ends of the curve. This leads to the *endpoint interpolation property*. A system (u_0, \dots, u_n) has the endpoint interpolation property if any parametric curve

$$\gamma(t) = \sum_{i=0}^n P_i u_i(t), \quad t \in [a, b],$$

satisfies $\gamma(a) = P_0$ and $\gamma(b) = P_n$, that is, the first control point always coincides with the start point of the curve and the last control point always coincides with the final point of the curve. This property is also illustrated in the Geogebra application mentioned before.

In interactive design we also want that the shape of a parametrically defined curve mimics the shape of its control polygon because then we can predict or manipulate the shape of the curve by choosing suitable control points. Due to the variation diminishing property of TP matrices, curves defined by NTP bases imitate the shape of their control polygons. In [44] it was proved that the length, number of inflections and angular variation of a parametric curve are bounded above by those of the control polygon with respect to NTP bases (see also [10]).

In fact, given a system of functions, if all curves generated from it satisfy simultaneously the convex hull, the endpoint interpolation and the variation diminishing properties then the system is NTP (see Figure 3.3).

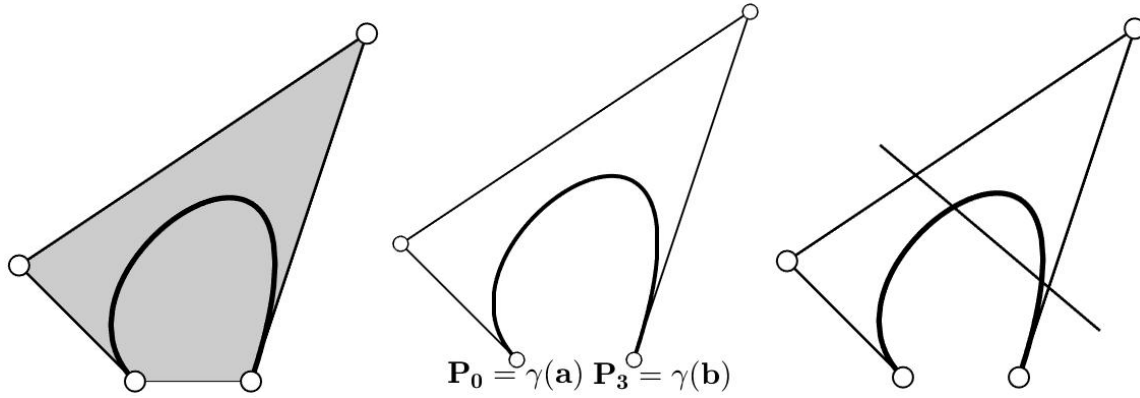


Figure 3.3: Convex hull, endpoint interpolation and variation diminishing properties.

As we are going to see in the following result, the product of TP bases and nonsingular TP matrices provides other TP bases.

Proposition 3.4. Let (u_0, \dots, u_n) be a TP basis of a space \mathcal{U} of functions defined on the subset $I \subseteq \mathbb{R}$ and let $A \in \mathbb{R}^{(n+1) \times (n+1)}$ be a nonsingular and TP matrix. The system of functions (v_0, \dots, v_n) defined on I by

$$(v_0, \dots, v_n) := (u_0, \dots, u_n)A$$

is a TP basis of \mathcal{U} . Moreover, if (u_0, \dots, u_n) is NTP and A is stochastic and TP, then (v_0, \dots, v_n) is NTP.

Proof. Taking into account that A is a nonsingular matrix and (u_0, \dots, u_n) is a basis, we deduce that the functions of (v_0, \dots, v_n) are linearly independent and thus (v_0, \dots, v_n) is a new basis of the space \mathcal{U} . Let us now consider any sequence of points $t_1 < \dots < t_{n+1}$ on I . The corresponding collocation matrix of (v_0, \dots, v_n) satisfies

$$(v_{j-1}(t_i))_{i,j=1,\dots,n+1} = (u_{j-1}(t_i))_{i,j=1,\dots,n+1}A.$$

Since (u_0, \dots, u_n) is a TP system, then the collocation matrix $(u_{j-1}(t_i))_{i,j=1,\dots,n+1}$ is TP. Let us observe that the product of TP matrices is, by Proposition 3.2, a TP matrix. Therefore, the collocation matrix $(v_{j-1}(t_i))_{i,j=1,\dots,n+1}$ is TP and thus (v_0, \dots, v_n) is a TP system. If, in addition, (u_0, \dots, u_n) is a partition of the unity,

$$\sum_{i=0}^n u_i(t) = 1, \quad t \in I,$$

and A is a stochastic matrix we can write

$$(v_0(t), \dots, v_n(t))e_{n+1} = (u_0(t), \dots, u_n(t))Ae_{n+1} = (u_0(t), \dots, u_n(t))e_{n+1} = 1, \quad t \in I,$$

where $e_{n+1} = (1, \dots, 1)^T \in \mathbb{R}^{n+1}$. So we deduce that,

$$\sum_{i=0}^n v_i(t) = 1, \quad t \in I,$$

that is, (v_0, \dots, v_n) is a partition of the unity and therefore is an NTP system. \square

One interesting goal in CAGD is to find the set of all TP bases of a space from a particular basis which generates every TP basis by means of nonsingular and TP matrices. The concept of *B-basis* arises in this context.

Definition 3.5. Let (u_0, \dots, u_n) be a TP basis of a space \mathcal{U} . Then (u_0, \dots, u_n) is a *B-basis* if for any other TP basis (v_0, \dots, v_n) of \mathcal{U} the matrix K of change of basis such that

$$(v_0, \dots, v_n) = (u_0, \dots, u_n)K$$

is TP. A B-basis (u_0, \dots, u_n) is a *normalized B-basis* of a space \mathcal{U} if the system (u_0, \dots, u_n) is a partition of the unity.

Among all NTP bases of a space, by Corollary 4.4 of [10], we can find a unique normalized B-basis. Furthermore, as we are going to see in the following result, corresponding to Proposition 3.12 of [10], the normalized B-bases can be characterized by their behavior at the ends of the definition interval.

Proposition 3.5. Let (u_0, \dots, u_n) be a totally positive basis of a space \mathcal{U} of functions defined on $I \subseteq \mathbb{R}$. Then (u_0, \dots, u_n) is a B-basis if and only if the following condition hold for all $i \neq j$

$$\inf \left\{ \frac{u_i(t)}{u_j(t)} \mid t \in I_0, \quad u_j(t) \neq 0 \right\} = 0.$$

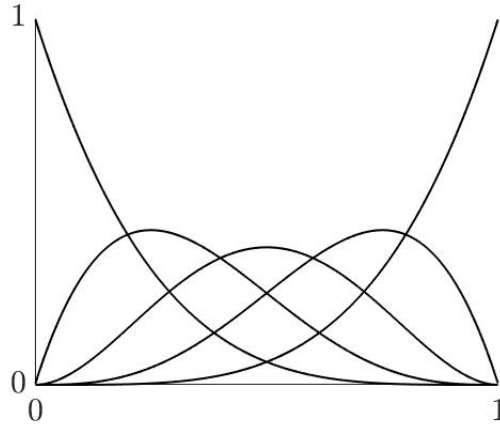
The following result guarantees the existence of B-bases and normalized B-bases when the considered spaces have TP or NTP, respectively, bases (see Remark 3.8 and Theorem 4.2 (i) of [10]).

Proposition 3.6. If a vector space of functions has a TP (respectively, NTP) basis, then it has a B-basis (respectively normalized B-basis).

Now, we establish the uniqueness of normalized B-bases, which was proved in Corollary 3.9 (iii) and Theorem 4.2(i), both of [10].

Proposition 3.7. Let (u_0, \dots, u_n) be a B-basis (respectively, normalized B-basis) of a space of functions of \mathcal{U} . A basis of \mathcal{U} is a B-basis (respectively normalized B-basis) if and only if it is of the form $(d_0 u_0, \dots, d_n u_n)$ with $d_i > 0$ (respectively, $d_i = 1$) for all $i = 0, \dots, n$.

It is well known that the Bernstein basis of degree n on a compact interval is the normalized B-basis of the corresponding space of polynomials. Moreover, the B-spline basis associated to a knot vector is the normalized B-basis of their corresponding space of spline functions.

Figure 3.4: Bernstein basis polynomials of degree 4 on $[0, 1]$.

Let (v_0, \dots, v_n) be an NTP basis of a space \mathcal{U} and (u_0, \dots, u_n) the corresponding normalized B-basis. Since (u_0, \dots, u_n) is the normalized B-basis, the matrix K of the change of basis such that

$$(v_0, \dots, v_n) = (u_0, \dots, u_n)K$$

is nonsingular, stochastic and TP. Let us suppose that γ is a parametric curve defined in terms of (v_0, \dots, v_n) . This curve can be written in matrix form as

$$\gamma(t) = \sum_{i=0}^n P_i v_i(t) = (v_0(t), \dots, v_n(t)) \begin{pmatrix} P_0 \\ \vdots \\ P_n \end{pmatrix}, \quad t \in [a, b],$$

where $P_0 \cdots P_n$ is the control polygon of γ with respect to the NTP basis (v_0, \dots, v_n) . This curve can also be written in terms of the normalized B-basis (u_0, \dots, u_n) , that is,

$$\gamma(t) = \sum_{i=0}^n Q_i u_i(t) = (u_0(t), \dots, u_n(t)) \begin{pmatrix} Q_0 \\ \vdots \\ Q_n \end{pmatrix}, \quad t \in [a, b],$$

where $Q_0 \cdots Q_n$ is the control polygon of γ with respect to the normalized B-basis. Clearly, we have

$$\begin{aligned} \gamma(t) &= (v_0(t), \dots, v_n(t)) \begin{pmatrix} P_0 \\ \vdots \\ P_n \end{pmatrix} = (u_0(t), \dots, u_n(t)) K \begin{pmatrix} P_0 \\ \vdots \\ P_n \end{pmatrix} \\ &= (u_0(t), \dots, u_n(t)) \begin{pmatrix} Q_0 \\ \vdots \\ Q_n \end{pmatrix} \end{aligned}$$

where the polygon $Q_0 \cdots Q_n$ obviously satisfies

$$\begin{pmatrix} Q_0 \\ \vdots \\ Q_n \end{pmatrix} = K \begin{pmatrix} P_0 \\ \vdots \\ P_n \end{pmatrix}.$$

Taking into account that K is a nonsingular, stochastic and TP matrix, we can deduce that the polygon $Q_0 \cdots Q_n$ can be obtained from a corner cutting algorithm starting with $P_0 \cdots P_n$. This fact guarantees that the control polygon of γ with respect to the normalized B-basis is closer to the parametric curve than the control polygon of γ with respect to the NTP basis (v_0, \dots, v_n) (see Figure (3.5)). Actually, the normalized B-basis has the optimal shape preserving properties among all NTP bases of \mathcal{U} in the sense that any parametric curve is going to imitate better the shape of its control polygon with respect to the normalized B-basis than the shape of its control polygon with respect to any other NTP basis.

In order to illustrate these facts, we have implemented a Geogebra application representing the control points of a polynomial parametric curve with respect to a four dimensional NTP polynomial basis of degree 3 on $[0, 1]$ (control polygon $P_0 \cdots P_3$) and the control points with respect to the Bernstein basis of degree 3 on $[0, 1]$ (control polygon $Q_0 \cdots Q_3$), it can be found in this url <https://ggbm.at/ACWPzxKg>. In the application the points can be moved and it can be easily observed that $Q_0 \cdots Q_3$ can be obtained from a corner cutting starting with $P_0 \cdots P_3$ and that for any choice of control points the curve mimics the shape of its control polygon with respect to the normalized B-basis.

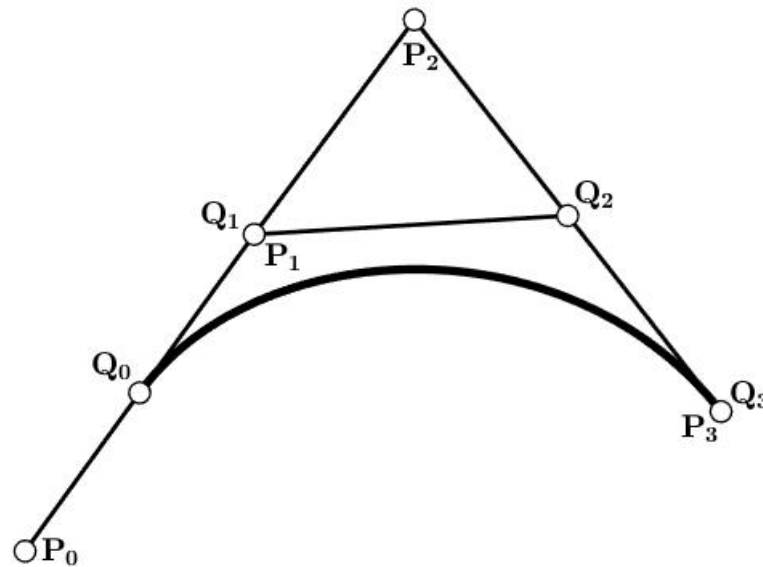


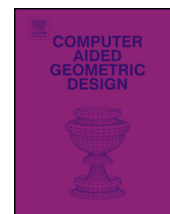
Figure 3.5: Corner cutting to obtain the control polygon $Q_0 \cdots Q_3$ of γ with respect to the normalized B-basis.

Part II

PRESENTATION OF THE PUBLICATIONS

ARTICLE 1

- [73] E. Mainar, J.M. Peña, B. Rubio, Evaluation and subdivision algorithms for general classes of totally positive rational bases, *Computer Aided Geometric Design* **81** (2020).
<https://doi.org/10.1016/j.cagd.2020.101900>



Evaluation and subdivision algorithms for general classes of totally positive rational bases^{☆,☆☆}



E. Mainar, J.M. Peña, B. Rubio^{*}

Departamento de Matemática Aplicada/IUMA, Universidad de Zaragoza, Spain

ARTICLE INFO

Article history:

Received 14 November 2019
 Received in revised form 26 May 2020
 Accepted 28 May 2020
 Available online 3 June 2020

Keywords:

Normalized totally positive basis
 Normalized B-basis
 Rational basis
 de Casteljaou algorithms
 Subdivision algorithms

ABSTRACT

Weighted φ -transformed systems include many systems of functions useful in C.A.G.D. It is proved that these systems inherit some geometric properties of a given initial system, such as shape preservation or optimal shape preservation. A general class of important rational bases can be obtained as a particular example of weighted φ -transformed systems. For these bases, evaluation and subdivision algorithms are presented. Some relevant applications are pointed out.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Given an initial system of functions and a positive function φ , weighted φ -transformed systems provide a very general procedure for generating new systems of functions useful for curve design. These systems include bases formed by Poisson functions (see Morin and Goldman, 2000), by Bernstein basis functions of negative degree (see Goldman, 1999) and important rational bases (see Šir and Jüttler, 2015; Han et al., 2016) as well as systems belonging to spaces mixing algebraic, trigonometric and hyperbolic polynomials, which are useful in many applications, for instance in Isogeometric Analysis (cf. Manni et al., 2011). We prove in this paper that weighted φ -transformed systems inherit from the initial system its nice geometric properties and we provide algorithms for a general class of rational bases.

In CAGD shape preservation is a necessary requirement for the representation of information through graphs and images. As explained in Section 2, shape preserving representations in computer-aided design are associated with normalized totally positive (NTP) bases, because these bases guarantee that the curve imitates the geometric properties of its control polygon. Among all NTP bases of a space, there exists a unique normalized B-basis, which is the basis with optimal shape preserving properties (cf. Peña, 1997; Carnicer and Peña, 1994). The Bernstein bases and the B-spline bases are the normalized B-bases of their corresponding generated spaces. We shall prove in Section 2 that the shape preserving properties associated to TP bases and B-bases are inherited by the representations associated to their weighted φ -transformed systems. In fact, the total positivity and the property of being a B-basis are preserved. Bases formed by Poisson functions, Bernstein basis functions of negative degree and rational bases can be considered as particular examples of weighted φ -transformed systems. Therefore,

[☆] Editor: Ulrich Reif.

^{☆☆} This work was partially supported through the Spanish research grant PGC2018-096321-B-I00 (MCIU/AEI), by Gobierno de Aragón (E41-17R) and Feder 2014-2020 "Construyendo Europa desde Aragón".

^{*} Corresponding author.

E-mail address: brubio@unizar.es (B. Rubio).

using the results of Section 2, one can deduce that bases formed by Bernstein basis functions of negative degree are TP, that bases formed by Poisson functions are B-bases and that rational systems obtained from B-bases are the normalized B-bases of the generated spaces of rational functions and have optimal shape preserving properties. In particular, it can be deduced that the Rational Bernstein basis and NURBS bases are NTP and the normalized B-bases of the corresponding spaces of rational functions.

In Mainar and Peña (1999), for a given space of functions that admits shape preserving representations a corner cutting algorithm, called a B-algorithm, is proposed. B-algorithms are evaluation algorithms satisfying important properties such as a subdivision property and convergence to the curve of the resulting control polygons. Supported by the results in Mainar and Peña (1999), in Section 3 evaluation and subdivision algorithms for a general class of rational bases are deduced.

In Šir and Jüttler (2015) nested spaces of rational polynomial functions of a given degree n and with a common denominator are considered. The corresponding rational Bézier curves admit up to $n!$ different de Casteljau-type algorithms. In Section 4 we show that the results from Šir and Jüttler (2015) can be extended to spaces of non polynomial rational functions deriving recurrence formulas for the weights and basis functions of these spaces. Curves generated by these weighted φ -transformed bases inherit geometric properties and algorithms of the traditional rational Bézier curves and so they can be considered as modeling tools in CAD/CAM systems.

2. Weighted φ -transformed systems and total positivity

Given a system (u_0, \dots, u_n) of linearly independent functions defined on an interval $I \subseteq \mathbb{R}$ and $P_0, \dots, P_n \in \mathbb{R}^n$ we can define a parametric curve as

$$\gamma(t) := \sum_{i=0}^n P_i u_i(t), \quad t \in I.$$

The polygon $P_0 \cdots P_n$, formed by the ordered sequence of points $P_i \in \mathbb{R}^n$, $i = 0, \dots, n$, is called the *control polygon* of γ and the points P_i , $i = 0, \dots, n$, are named *control points* of γ with respect to (u_0, \dots, u_n) .

A matrix is *totally positive* (TP) if all its minors are nonnegative and *strictly totally positive* (STP) if all the minors are positive (see Ando, 1987). A system of functions (u_0, \dots, u_n) defined on the subset $I \subseteq \mathbb{R}$ is TP if all its collocation matrices

$$(u_{j-1}(t_i))_{i,j=1,\dots,n+1}, \quad \text{with } t_1 < \dots < t_{n+1} \text{ in } I,$$

are TP. A TP system of functions on I is *normalized* (NTP) if $\sum_{i=0}^n u_i(t) = 1$, for all $t \in I$. NTP bases are commonly used in computer-aided design due to their shape preserving properties (see Carnicer and Peña, 1993; Peña, 1997).

Among all NTP bases of a space, we can find a unique normalized B-basis, which is the optimal shape preserving basis (cf. Carnicer and Peña, 1994). For instance, the Bernstein bases and the B-spline bases are the normalized B-bases of their corresponding spaces. The following characterization of a B-basis is a consequence of Corollary 3.10 and Proposition 3.11 of Carnicer and Peña (1994).

Theorem 1. Let (u_0, \dots, u_n) be a TP basis of a space \mathcal{U} . Then (u_0, \dots, u_n) is a B-basis if and only if for any other TP basis (v_0, \dots, v_n) of \mathcal{U} the change of basis matrix K with $(v_0, \dots, v_n) = (u_0, \dots, u_n)K$ is TP.

Let (u_0, \dots, u_n) be a system of functions defined on I and d_0, \dots, d_n positive real values. From Lemma A.1 of Goldman (1985) it can be proved that the system $(d_0 u_0, \dots, d_n u_n)$ is TP if and only if (u_0, \dots, u_n) is TP.

Now let us also consider a positive function $\varphi : I \rightarrow \mathbb{R}$. We say that $(\tilde{u}_0, \dots, \tilde{u}_n)$ is a *weighted φ -transformed system* from (u_0, \dots, u_n) if

$$\tilde{u}_i(t) := d_i \varphi(t) u_i(t), \quad t \in I, \quad i = 0, \dots, n. \quad (1)$$

The following result shows that a weighted φ -transformed system also inherits from the initial system the properties of being TP and being a B-basis.

Theorem 2. Let (u_0, \dots, u_n) be a system of functions defined on I and let $(\tilde{u}_0, \dots, \tilde{u}_n)$ be the weighted φ -transformed system given by (1).

- i) If (u_0, \dots, u_n) is TP, then $(\tilde{u}_0, \dots, \tilde{u}_n)$ is TP.
- ii) If (u_0, \dots, u_n) is a B-basis, then $(\tilde{u}_0, \dots, \tilde{u}_n)$ is a B-basis.

Proof. i) Let us suppose that (u_0, \dots, u_n) is TP on I . Given $t_1 < \dots < t_{n+1}$ in I , the corresponding collocation matrix satisfies

$$(\tilde{u}_{j-1}(t_i))_{1 \leq i, j \leq n+1} = \text{diag}(d_0, \dots, d_n) (u_{j-1}(t_i))_{1 \leq i, j \leq n+1} \text{diag}(\varphi(t_1), \dots, \varphi(t_{n+1})),$$

and is TP since it is the product of TP matrices (cf. Theorem 3.1 of Ando, 1987). So, $(\tilde{u}_0, \dots, \tilde{u}_n)$ is TP.

ii) Let us now assume that (u_0, \dots, u_n) is a B-basis of \mathcal{U} . Let $(\tilde{v}_0, \dots, \tilde{v}_n)$ be any TP basis of the space $\tilde{\mathcal{U}}$ generated by $(\tilde{u}_0, \dots, \tilde{u}_n)$ and $\tilde{K} \in \mathbb{R}^{(n+1) \times (n+1)}$ the change of basis matrix such that

$$(\tilde{v}_0, \dots, \tilde{v}_n) = (\tilde{u}_0, \dots, \tilde{u}_n)\tilde{K}. \tag{2}$$

Let us now see that the system (v_0, \dots, v_n) defined by

$$v_i(t) := \frac{1}{d_i\varphi(t)}\tilde{v}_i(t), \quad t \in I, \quad i = 0, \dots, n, \tag{3}$$

is a TP basis of \mathcal{U} . Clearly,

$$(v_0(t), \dots, v_n(t)) = (\tilde{v}_0(t), \dots, \tilde{v}_n(t))D_1(t), \quad t \in I, \tag{4}$$

where $D_1(t) := \frac{1}{\varphi(t)}D^{-1}$ and $D := \text{diag}(d_0, \dots, d_n)$. By (2) and (4),

$$(v_0(t), \dots, v_n(t)) = (\tilde{u}_0(t), \dots, \tilde{u}_n(t))\tilde{K}D_1(t), \quad t \in I.$$

On the other hand, observe that

$$(\tilde{u}_0(t), \dots, \tilde{u}_n(t)) = (u_0(t), \dots, u_n(t))D_2(t), \quad t \in I, \tag{5}$$

where $D_2(t) := D\varphi(t)$. Taking into account that $D_2(t)\tilde{K}D_1(t) = D\tilde{K}D^{-1}$ for all $t \in I$, we have

$$(v_0, \dots, v_n) = (u_0, \dots, u_n)K, \quad K := D\tilde{K}D^{-1}, \tag{6}$$

and we can conclude that (v_0, \dots, v_n) is a basis of \mathcal{U} . Observe that, by (3), the collocation matrix $(v_{j-1}(t_i))_{1 \leq i, j \leq n+1}$ satisfies

$$(v_{j-1}(t_i))_{1 \leq i, j \leq n+1} = \text{diag}\left(\frac{1}{\varphi(t_1)}, \dots, \frac{1}{\varphi(t_{n+1})}\right) (\tilde{v}_{j-1}(t_i))_{1 \leq i, j \leq n+1} D^{-1}$$

and it is TP since it is the product of TP matrices. This proves that (v_0, \dots, v_n) is a TP basis of \mathcal{U} . By Theorem 1, the matrix K such that

$$(v_0, \dots, v_n) = (u_0, \dots, u_n)K \tag{7}$$

is TP. Finally, by (6), $\tilde{K} = D^{-1}KD$ and it is a TP matrix because \tilde{K} is the product of TP matrices. Then the result follows from (2) and Theorem 1. \square

Let us now consider a first interesting example. Recall that the Poisson basis functions

$$b_k(t) := \frac{t^k}{k!}e^{-t}, \quad t \in [0, \infty), \quad k \in \mathbb{N},$$

are the limit as n tends to infinity of the Bernstein basis of degree n over the interval $[0, n]$, that is,

$$b_k(t) = \lim_{n \rightarrow \infty} B_k^n(t/n), \quad B_k^n(t) = \binom{n}{k}t^k(1-t)^{n-k}, \quad t \in [0, 1],$$

and they also play a useful role in CAGD (cf. Morin and Goldman, 2000). For a given $n \in \mathbb{N}$, the system (b_0, \dots, b_n) of Poisson basis functions can be considered as a weighted φ -transformed system from the monomial basis $(1, t, \dots, t^n)$ with $\varphi(t) = e^{-t}$ and $d_i = 1/i!$, $i = 0, \dots, n$. Then, using Theorem 2 and the well known fact that the monomial basis is TP and a B-basis on $[0, \infty)$ (see Section 6 of Carnicer and Peña, 1993 and Carnicer and Peña, 1994), we deduce from Theorem 2 that (b_0, \dots, b_n) is TP on $[0, \infty)$ and also a B-basis.

Poisson curves are analytic functions represented by infinite series, $F(t) = \sum_{k \geq 0} P_k b_k(t)$, $0 \leq t < R$, such that this series converges for $0 \leq t < R$. The Poisson basis functions are positive on $(0, \infty)$ and form a partition of the unity. Therefore Poisson curves are affinely invariant and lie in the convex hull of their control points (see Morin and Goldman, 2000).

Bernstein bases of negative degree are also interesting cases to which we can apply Theorem 2. For $n \in \mathbb{N}$, the Bernstein basis functions of degree $-n$ are defined by

$$B_k^{-n}(t) := \binom{-n}{k}t^k(1-t)^{-n-k},$$

$$\binom{-n}{k} = \frac{(-n)(-n-1)\dots(-n-k+1)}{k!} = (-1)^k \binom{n+k-1}{k}, \quad k = 0, 1, \dots. \tag{8}$$

From (8), one can easily check that the Bernstein basis functions of negative degree are non-negative over the interval $(-\infty, 0]$. Furthermore, they are linearly independent and form a partition of unity (cf. Goldman, 1999).

Let us observe that $(B_m^{-n}, B_{m-1}^{-n}, \dots, B_0^{-n})$ can be considered as a weighted φ -transformed system from the basis $(1, (t-1)/t, \dots, ((t-1)/t)^m)$ with $\varphi(t) = (-t)^m(1-t)^{-n-m}$ and $d_k = \binom{n-1+m-k}{m-k}$, $k = 0, \dots, m$. Taking into account that the monomial basis is TP on the interval $[0, \infty)$ and the fact that $(t-1)/t$, $t \in (-\infty, 0)$, takes increasing positive values, we can deduce that $(1, (t-1)/t, \dots, ((t-1)/t)^m)$ is TP on $(-\infty, 0)$. Consequently, using Theorem 2 (i), we conclude that the system of Bernstein basis functions of negative degree $(B_m^{-n}, B_{m-1}^{-n}, \dots, B_0^{-n})$ is also TP on its natural domain. As a special case of Theorem 2, Appendix A of Goldman (1999) proves the Descartes' Law of Signs for the Bernstein bases of negative degree which implies the variation diminishing property of the generated curves and agrees with the total positivity of the basis mentioned above. Standard recurrence relations and two term formulas for differentiation and degree elevation are also deduced in Goldman (1999).

Since any finite sequence of Bernstein basis functions of negative degree and any finite sequence of Poisson basis functions are TP, the same property can be derived for the infinite sequence of Bernstein basis functions of negative degree and the infinite sequence of Poisson basis functions on their natural domain. In this last case, since any finite sequence of Poisson basis functions is also a B-basis, a similar property can be expected for the infinite sequence of Poisson basis functions.

Recall that given a system (u_0, \dots, u_n) of functions defined on I and positive values w_0, \dots, w_n such that $\sum_{k=0}^n w_k u_k(t) \neq 0$, for all $t \in I$, the system (r_0, \dots, r_n) defined by

$$r_i(t) := \frac{w_i u_i(t)}{\sum_{k=0}^n w_k u_k(t)}, \quad i = 0, \dots, n,$$

satisfies $\sum_{i=0}^n r_i(t) = 1$, $\forall t \in I$, and generates a new space of rational functions. If (u_0, \dots, u_n) is TP then $\sum_{k=0}^n w_k u_k(t) > 0$, $\forall t \in I$, and (r_0, \dots, r_n) can be considered as a particular weighted φ -transformed system with

$$\varphi(t) := \frac{1}{\sum_{k=0}^n w_k u_k(t)}, \quad t \in I. \quad (9)$$

Now observe that, by Theorem 2, (r_0, \dots, r_n) is NTP. Furthermore, if (u_0, \dots, u_n) is a B-basis, we can also use Theorem 2 to deduce that (r_0, \dots, r_n) is the normalized B-basis of the generated space. Although this fact has been mentioned in Mainar and Peña (1999), to the best of our knowledge, no proof of this result has been provided in the literature. Rational Bernstein basis are NTP basis (see Delgado and Peña, 2013). From Theorem 2 the optimal shape preserving properties of these rational polynomial systems can be also guaranteed. Similarly, using Theorem 2, one can also deduce that NURBS bases are NTP and are the normalized B-bases of the corresponding spaces of rational spline functions.

3. A general class of rational bases

Suppose that $I \subseteq \mathbb{R}$ and $f, g : I \rightarrow \mathbb{R}$ are nonnegative continuous functions. Define the system

$$(u_0^n, \dots, u_n^n), \quad u_k^n(t) := \binom{n}{k} f^k(t) g^{n-k}(t), \quad t \in I, \quad k = 0, \dots, n. \quad (10)$$

Spaces generated by the system (10) and the associated de Casteljau-type evaluation algorithms are discussed in Disibuyuk and Goldman (2015) and Gonsor and Neamtu (1994).

The following result corresponds to Proposition 19 of Mainar and Peña (1999) and provides the conditions characterizing the system defined in (10) as a B-basis of its generated space.

Proposition 3. *The system given in (10) is a B-basis if and only if the function f/g defined on $I_0 := \{t \in I \mid g(t) \neq 0\}$ is increasing and satisfies*

$$\inf \left\{ \frac{f(t)}{g(t)} \mid t \in I_0 \right\} = 0, \quad \sup \left\{ \frac{f(t)}{g(t)} \mid t \in I_0 \right\} = +\infty. \quad (11)$$

For any positive weights w_i^n , $i = 0, \dots, n$, let us denote by $(\rho_0^n, \dots, \rho_n^n)$ the rational bases such that

$$\rho_i^n(t) := w_i^n \frac{1}{\omega^n(t)} u_i^n(t), \quad i = 0, \dots, n, \quad \omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t). \quad (12)$$

This system spans the space of rational functions with denominator $\omega^n(t)$,

$$\mathcal{R}^n := \text{span}\{\rho_i^n(t) \mid i = 0, \dots, n\} = \{u(t)/\omega^n(t) \mid u(t) \in \mathcal{U}^n\} \quad (13)$$

where \mathcal{U}^n is the space generated by the basis (10). Using Proposition 3 and Theorem 2 we deduce the following result.

Corollary 4. *The system of functions given in (12) is the normalized B-basis of the space \mathcal{R}^n in (13) if and only if the function f/g defined on $I_0 := \{t \in I \mid g(t) \neq 0\}$ is increasing and satisfies (11).*

Consequently, if the functions f and g satisfy conditions (11), the rational systems (12) are shape preserving.

Given $I = [a, b]$ and $f, g : I \rightarrow \mathbb{R}$ satisfying conditions (11), let us consider the system (u_0^n, \dots, u_n^n) defined in (10) and its generated space \mathcal{U} . For any $t_0 \in (a, b]$, such that $f(t_0) > 0$, define $I' := [a, t_0]$ and the system $(\tilde{u}_0^n, \dots, \tilde{u}_n^n)$ given by

$$\tilde{u}_i^n(t) := \binom{n}{i} \tilde{f}^i(t) \tilde{g}^{n-i}(t), \quad t \in I', \quad i = 0, \dots, n, \tag{14}$$

where

$$\tilde{f}(t) := \frac{f(t)}{f(t_0)}, \quad t \in I', \quad \tilde{g}(t) := \frac{f(t_0)g(t) - g(t_0)f(t)}{f(t_0)}, \quad t \in I'.$$

In Mainar and Peña (1999) it is shown that the system (14) is a B-basis of the space $\mathcal{U}|_{I'}$ formed by the restrictions to I' of the functions of \mathcal{U} . On the other hand, the matrix L such that

$$(u_0^n(t), \dots, u_n^n(t)) := (\tilde{u}_0^n(t), \dots, \tilde{u}_n^n(t)) L, \quad t \in I', \tag{15}$$

is nonsingular, lower triangular and TP.

Now let us consider positive weights $w_i^n, i = 0, \dots, n$. By (15) we can write

$$(u_0^n(t), \dots, u_n^n(t)) (w_0^n, \dots, w_n^n)^T = (\tilde{u}_0^n(t), \dots, \tilde{u}_n^n(t)) L (w_0^n, \dots, w_n^n)^T, \quad t \in I'. \tag{16}$$

Taking into account (16), it can be deduced that the weights $\tilde{w}_i^n, i = 0, \dots, n$, obtained by

$$(\tilde{w}_0^n, \dots, \tilde{w}_n^n)^T := L (w_0^n, \dots, w_n^n)^T \tag{17}$$

satisfy

$$\omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t) = \sum_{i=0}^n \tilde{w}_i^n \tilde{u}_i^n(t), \quad t \in I'. \tag{18}$$

Then, by Theorem 2, we can deduce that the system $(\tilde{\rho}_0^n, \dots, \tilde{\rho}_n^n)$ defined by

$$\tilde{\rho}_i^n(t) := \tilde{w}_i^n \frac{1}{\omega^n(t)} \tilde{u}_i^n(t), \quad t \in I', \quad i = 0, \dots, n, \tag{19}$$

is the normalized B-basis of the space $\mathcal{R}^n|_{I'}$ formed by the restrictions to I' of the functions of the rational space \mathcal{R}^n in (13). Clearly, by defining the diagonal matrices $D := \text{diag}\{w_0^n, \dots, w_n^n\}$ and $\tilde{D} := \text{diag}\{\tilde{w}_0^n, \dots, \tilde{w}_n^n\}$, we have

$$\begin{aligned} (\rho_0^n(t), \dots, \rho_n^n(t)) &= \frac{1}{\omega^n(t)} (u_0^n(t), \dots, u_n^n(t)) D = \frac{1}{\omega^n(t)} (\tilde{u}_0^n(t), \dots, \tilde{u}_n^n(t)) L D \\ &= \frac{1}{\omega^n(t)} (\tilde{u}_0^n(t), \dots, \tilde{u}_n^n(t)) \tilde{D} \tilde{D}^{-1} L D = (\tilde{\rho}_0^n(t), \dots, \tilde{\rho}_n^n(t)) \tilde{D}^{-1} L D, \quad t \in I'. \end{aligned}$$

Therefore, $\tilde{L} := \tilde{D}^{-1} L D$ is the change of basis matrix such that

$$(\rho_0^n(t), \dots, \rho_n^n(t)) := (\tilde{\rho}_0^n(t), \dots, \tilde{\rho}_n^n(t)) \tilde{L}, \quad t \in I'.$$

In Mainar and Peña (1999) a de Casteljau-like algorithm (called a B-algorithm) providing exact evaluation and subdivision for parametric curves $\gamma(t) := \sum_{i=0}^n P_i u_i^n(t), t \in I$ is proposed. Now, exploiting the results in Mainar and Peña (1999) and using the factorization of the matrix L in terms of bidiagonal matrices, we can obtain a bidiagonal factorization of the change of basis matrix \tilde{L} and derive this kind of algorithm for the evaluation and subdivision of the rational curve $\rho(t) := \sum_{i=0}^n P_i \rho_i^n(t), t \in I$. The computed points $P_0^0, P_1^1, \dots, P_n^n$ satisfy

$$\rho(t) := \sum_{i=0}^n P_i^i \tilde{\rho}_i^n(t), \quad t \in I',$$

and, in particular, $P_n^n = \rho(t_0)$.

Algorithm 1: Evaluation and left subdivision algorithm.

for $j:=0$ **to** n

$$d_j^0 := w_j^n, \quad P_j^0 := P_j$$

for $i:=0$ **to** $n-1$

for $j:=0$ **to** i

$$P_j^{i+1} := P_j^i$$

for $j:=i+1$ **to** n

$$d_j^{i+1} := g(t_0)d_{j-1}^i + f(t_0)d_j^i$$

$$P_j^{i+1} := g(t_0) \frac{d_{j-1}^i}{d_j^{i+1}} P_{j-1}^i + f(t_0) \frac{d_j^i}{d_j^{i+1}} P_j^i$$

Observe that if $(w_0^n u_0^n, \dots, w_n^n u_n^n)$ is normalized then $\omega^n(t) = 1$ and the algorithm reduces to the Algorithm 5.1 in Mainar and Peña (1999) for the evaluation of non rational curves defined in terms of a normalized B-basis.

Similarly, we can consider $t_0 \in [a, b)$ and $I'' = [t_0, b]$. If $g(t_0) > 0$, then the system $(\widehat{u}_0, \dots, \widehat{u}_n)$ given by

$$\widehat{u}_i(t) := \binom{n}{i} \widehat{f}^i(t) \widehat{g}^{n-i}(t), \quad t \in I'', \quad i = 0, \dots, n, \quad (20)$$

where

$$\widehat{f}(t) := \frac{g(t_0)f(t) - f(t_0)g(t)}{g(t_0)}, \quad t \in I'', \quad \widehat{g}(t) := \frac{g(t)}{g(t_0)}, \quad t \in I'', \quad (21)$$

is a B-basis of $\mathcal{U}|_{I''}$. Let U be the nonsingular, upper triangular, TP matrix such that $(u_0^n(t), \dots, u_n^n(t)) := (\widehat{u}_0^n(t), \dots, \widehat{u}_n^n(t)) U$, for all $t \in I''$. Using the previous reasoning, we can deduce that

$$\omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t) = \sum_{i=0}^n \widehat{w}_i^n \widehat{u}_i^n(t) \quad (22)$$

where the weights \widehat{w}_i^n , $i = 0, \dots, n$, satisfy

$$(\widehat{w}_0^n, \dots, \widehat{w}_n^n)^T := U (w_0^n, \dots, w_n^n)^T. \quad (23)$$

Therefore, by Theorem 2, the system $(\widehat{\rho}_0^n, \dots, \widehat{\rho}_n^n)$ defined by

$$\widehat{\rho}_i^n(t) := \widehat{w}_i^n \frac{1}{\omega^n(t)} \widehat{u}_i^n(t), \quad t \in I'', \quad i = 0, \dots, n, \quad (24)$$

is the normalized B-basis of the space $\mathcal{R}^n|_{I''}$. Supported by the results in Mainar and Peña (1999), we can deduce the following algorithm for the evaluation and right subdivision of the rational curve $\rho(t) = \sum_{i=0}^n P_i \rho_i^n(t)$, $t \in I$.

Algorithm 2: Evaluation and Right subdivision algorithm.

for $j:=0$ **to** n

$$d_j^0 := w_j^n, \quad P_j^0 := P_j$$

for $i:=0$ **to** $n-1$

for $j:=0$ **to** $n-i-1$

$$d_j^{i+1} := g(t_0)d_j^i + f(t_0)d_{j+1}^i$$

$$P_j^{i+1} := g(t_0) \frac{d_j^i}{d_j^{i+1}} P_j^i + f(t_0) \frac{d_{j+1}^i}{d_j^{i+1}} P_{j+1}^i$$

for $j:=n-i$ **to** n

$$P_j^{i+1} := P_j^i$$

In order to illustrate the results of this section, let us first consider an interesting basis of rational polynomial functions. The Lupaş q -analogues of the Bernstein functions of degree n (cf. Han et al., 2016) are the rational Bernstein functions

$$\rho_i^n(t) := \frac{a_{n,i}(t)}{w_n(t)}, \quad a_{n,i}(t) := w_i^n t^i (1-t)^{n-i}, \quad w_n(t) := \sum_{i=0}^n a_{n,i}(t) = \prod_{j=1}^n (1-t+q^{j-1}t) \quad (25)$$

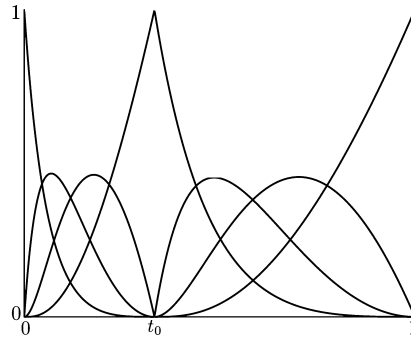


Fig. 1. Weighted Lupaş q -analogues of the Bernstein functions of degree 3 of $\mathcal{R}^n|_{I'}$ and $\mathcal{R}^n|_{I''}$ with $d_i = 1, i = 0, 1, 2, 3, q = 3$ and $t_0 = 1/3$.

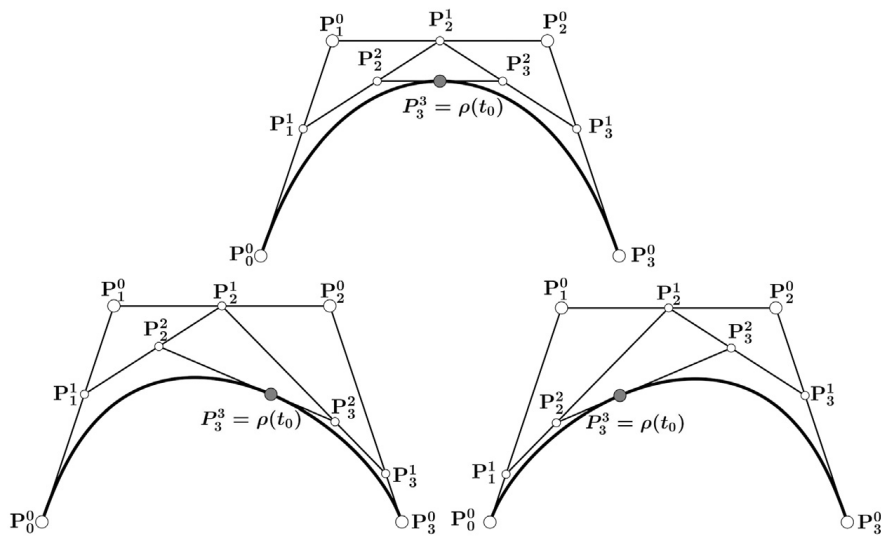


Fig. 2. de Casteljau-like algorithm to subdivide the rational curves of Lupaş q -analogues of the Bernstein functions at $t_0 = 1/4$ of degree 3 with $q = 3$. Top: Weights $d_i = [1, 1, 1, 1]$. Left bottom: Weights $d_i = [5, 1, 1, 1]$. Right bottom: Weights $d_i = [1, 1, 1, 5]$.

where $w_i^n := \binom{n}{i} q^{i(i-1)/2}$ and the q -binomial coefficient $\binom{n}{k}$, for integers $0 \leq k \leq n$, is defined by

$$\binom{n}{0} := 1, \quad \binom{n}{k} := \frac{[n][n-1] \cdots [n-k+1]}{[k]} = \frac{[n]!}{[k]![n-k]}, \quad k > 0,$$

with

$$[k]! := \begin{cases} [k][k-1] \cdots [1], & k \geq 1, \\ 1, & k = 0. \end{cases} \quad \text{and} \quad [k] := \begin{cases} (1-q^k)/(1-q), & q \neq 1, \\ k, & q = 1. \end{cases}$$

Now, by considering positive weights d_0, \dots, d_n , we can define the weighted Lupaş q -analogue of Bernstein functions of degree n as

$$r_i^n(t; q) := d_i \frac{1}{\omega^n(t)} a_{n,i}(t), \quad i = 0, \dots, n, \tag{26}$$

where $\omega^n(t) := \sum_{k=0}^n d_k a_{n,k}(t)$.

Using Proposition 3 and Theorem 2 we can deduce that the basis (26) is the normalized B-basis of its generated space \mathcal{R}^n of rational polynomials defined on $[0, 1]$. Given $t_0 \in (0, 1)$, using (19) and (24) we can obtain the bases of the normalized B-basis of $\mathcal{R}^n|_{I'}$ and $\mathcal{R}^n|_{I''}$. Fig. 1 shows these bases for $n = 3$.

In Fig. 2 we illustrate an example of the de Casteljau-like algorithm to subdivide the rational parametric curves $\gamma(t) := \sum_{i=0}^n P_i r_i^n(t; q)$ at a given parameter $t_0 \in (0, 1)$. We can also observe the effect of the weights, d_0, \dots, d_n , in the shape of corresponding curves.

Trigonometric and hyperbolic curves are attracting a lot of interest, since they provide the opportunity to construct catenaries, conics, cylinders and surfaces of revolution. In Rót (2015), by means of the normalized B-basis of the spaces $\langle 1, \cos t, \sin t, \dots, \cos mt, \sin mt \rangle$ and $\langle 1, \cosh t, \sinh t, \dots, \cosh mt, \sinh mt \rangle$, rational bases are generated. The applicability of the proposed construction is illustrated by the exact control point based representation of rational trigonometric or hyperbolic curves and multivariate surfaces.

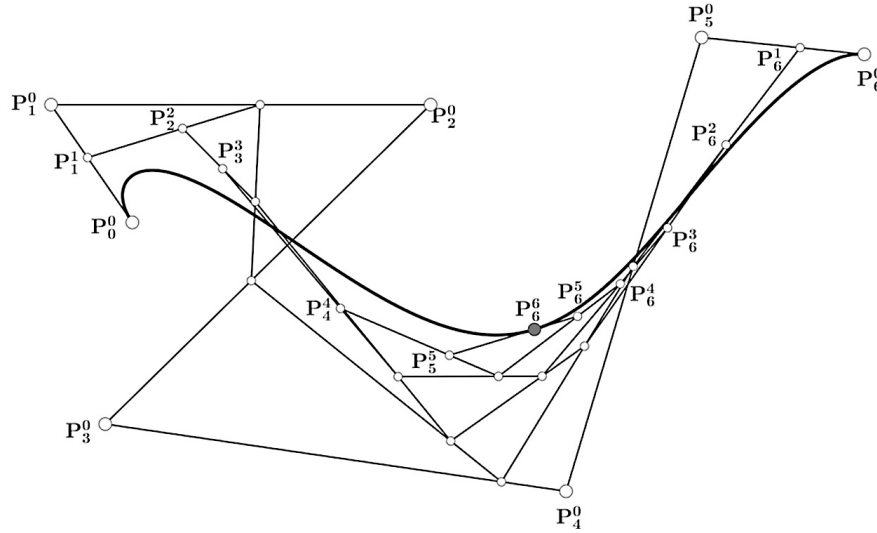


Fig. 3. de Casteljau-like algorithm to subdivide the rational trigonometric curves at $t_0 = 1/5$, $f(t) := \sin(\frac{1+t}{2})$, $g(t) := \sin(\frac{1-t}{2})$ and $w_i^n = [15, 5, 5, 5, 15, 5, 15]$.

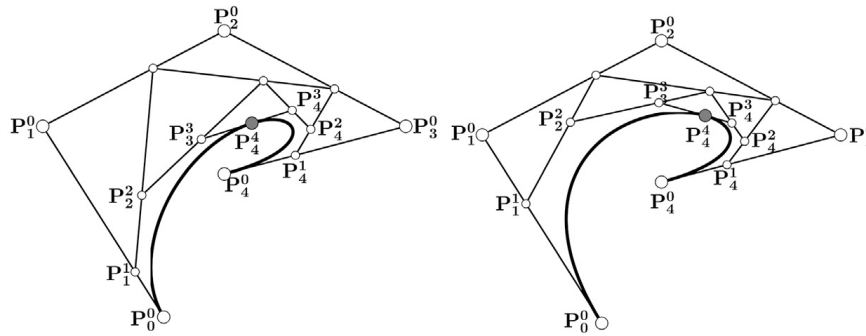


Fig. 4. de Casteljau-like algorithm to subdivide the rational hyperbolic curves at $t_0 = 1/5$ with $f(t) := \sinh(\frac{1+t}{2})$ and $g(t) := \sinh(\frac{1-t}{2})$. Left: Weights $w_i^n = [5, 1, 1, 1, 1]$, Right: Weights $w_i^n = [1, 1, 1, 1, 5]$.

In Fig. 3 we illustrate an example of the de Casteljau-like algorithm to subdivide the rational trigonometric curves $\gamma(t) := \sum_{i=0}^n P_i \rho_i^n(t)$ at a given parameter $t_0 \in (-\Delta, \Delta)$, $0 < \Delta < \pi/2$ by considering

$$f(t) := \sin\left(\frac{\Delta + t}{2}\right), \quad g(t) := \sin\left(\frac{\Delta - t}{2}\right), \quad t \in I = [-\Delta, \Delta].$$

Finally, in Fig. 4 we illustrate an example of the de Casteljau-like algorithm to subdivide the rational hyperbolic curves $\gamma(t) := \sum_{i=0}^n P_i \rho_i^n(t)$ at a given parameter $t_0 \in (-\Delta, \Delta)$, $0 < \Delta$, by considering

$$f(t) = \sinh\left(\frac{\Delta + t}{2}\right), \quad g(t) := \sinh\left(\frac{\Delta - t}{2}\right), \quad t \in I = [-\Delta, \Delta]. \tag{27}$$

We can also observe the effect of the weights, w_0^n, \dots, w_n^n , in the shape of corresponding curves.

4. A particular class of rational bases satisfying recurrence relations

Suppose that $I = [a, b]$ and $f, g : I \rightarrow \mathbb{R}$ are nonnegative continuous functions. Following the approach of Šir and Jüttler (2015), let us now consider an infinite sequence of linear factors

$$L_i(t) := a_i g(t) + b_i f(t), \quad i \in \mathbf{Z}_+, \tag{28}$$

defined by positive coefficients a_i and b_i , $(a_i, b_i) \neq (0, 0)$. For any positive integer n , let us also define

$$\omega^n(t) := L_1(t) \cdots L_n(t). \tag{29}$$

It can be checked that the function ω^n has a unique representation in terms of the basis (10),

$$\omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t), \quad u_i^n(t) := \binom{n}{i} f^i(t) g^{n-i}(t), \quad i = 0, \dots, n,$$

where

$$w_i^n = \frac{1}{\binom{n}{i}} \left(\sum_{\substack{K \cup L = \{1, \dots, n\} \\ |K| = (n-i), |L| = i}} \prod_{k \in K} a_k \prod_{l \in L} b_l \right). \tag{30}$$

Clearly, the positivity of the coefficients a_i and b_i guarantees the positivity of ω_i^n , $i = 0, \dots, n$, and $\omega^n(t)$, $\forall t \in I$.

Consider the rational basis $(\rho_0^n, \dots, \rho_n^n)$ defined in (12) and let \mathcal{R}^n be the generated space of rational functions. In this section we shall see that, for this particular choice of the weights w_i^n , the rational bases $(\tilde{\rho}_0^n, \dots, \tilde{\rho}_n^n)$ and $(\hat{\rho}_0^n, \dots, \hat{\rho}_n^n)$ defined in (19) and (24) satisfy recurrence relations defining de Casteljau-type evaluation and degree elevation algorithms.

First, let us prove that the weights \tilde{w}_i^n of (17) can be obtained recursively.

Proposition 5. For a given $n \in \mathbb{N}$, the weights \tilde{w}_i^n given in (17) satisfy

$$\tilde{w}_i^n = a_n \frac{n-i}{n} \tilde{w}_i^{n-1} + L_n(t_0) \frac{i}{n} \tilde{w}_{i-1}^{n-1}, \quad i = 0, \dots, n, \tag{31}$$

with $\tilde{w}_0^0 := 1$ and $L_n(t) = a_n g(t) + b_n f(t)$.

Proof. Observe that, by (29), $\omega^n(t) = \omega^{n-1}(t) L_n(t)$. Then, taking into account (18), we deduce that

$$\omega^n(t) = \left(\sum_{i=0}^{n-1} \tilde{w}_i^{n-1} \tilde{u}_i^{n-1}(t) \right) L_n(t). \tag{32}$$

We can easily derive that

$$L_n(t) = a_n \tilde{g}(t) + L_n(t_0) \tilde{f}(t), \tag{33}$$

therefore

$$\sum_{i=0}^n \tilde{w}_i^n \tilde{u}_i^n(t) = a_n \sum_{i=0}^{n-1} \tilde{w}_i^{n-1} \tilde{u}_i^{n-1}(t) \tilde{g}(t) + L_n(t_0) \sum_{i=0}^{n-1} \tilde{w}_i^{n-1} \tilde{u}_i^{n-1}(t) \tilde{f}(t). \tag{34}$$

Taking into account that $\frac{\binom{n-1}{i}}{\binom{n}{i}} = \frac{n-i}{n}$ and $\frac{\binom{n-1}{i}}{\binom{n}{i+1}} = \frac{i+1}{n}$, we can write

$$\begin{aligned} \sum_{i=0}^n \tilde{w}_i^n \tilde{u}_i^n(t) &= a_n \sum_{i=0}^{n-1} \frac{n-i}{n} \tilde{w}_i^{n-1} \tilde{u}_i^{n-1}(t) + L_n(t_0) \sum_{i=0}^{n-1} \frac{i+1}{n} \tilde{w}_i^{n-1} \tilde{u}_{i+1}^n(t) \\ &= a_n \sum_{i=0}^{n-1} \frac{n-i}{n} \tilde{w}_i^{n-1} \tilde{u}_i^{n-1}(t) + L_n(t_0) \sum_{i=1}^n \frac{i}{n} \tilde{w}_{i-1}^{n-1} \tilde{u}_i^n(t). \end{aligned}$$

Finally, by comparing the coefficients with respect the basis $(\tilde{u}_0^n, \dots, \tilde{u}_n^n)$, the result follows. \square

Similarly, taking into account (29), (22) and the equality $L_n(t) = L_n(t_0) \hat{g}(t) + b_n \hat{f}(t)$ we can obtain a recurrence formula satisfied by the weights \hat{w}_i^n given in (23).

Proposition 6. For a given $n \in \mathbb{N}$, the weights \hat{w}_i^n given in (23) satisfy the recurrence formula

$$\hat{w}_i^n := L_n(t_0) \frac{n-i}{n} \hat{w}_i^{n-1} + b_n \frac{i}{n} \hat{w}_{i-1}^{n-1}, \quad i = 0, \dots, n, \tag{35}$$

with $\hat{w}_0^0 := 1$ and $L_n(t) = a_n g(t) + b_n f(t)$.

The following recurrence relation proves the nested nature of the spaces $\mathcal{R}^n|_{I'}$.

Proposition 7. The system $(\tilde{\rho}_0^n, \dots, \tilde{\rho}_n^n)$ defined in (19) satisfies

$$\tilde{\rho}_i^n(t) := a_{n+1} \frac{n+1-i}{n+1} \frac{\tilde{w}_i^n}{\tilde{w}_i^{n+1}} \tilde{\rho}_i^{n+1}(t) + L_{n+1}(t_0) \frac{i+1}{n+1} \frac{\tilde{w}_i^n}{\tilde{w}_{i+1}^{n+1}} \tilde{\rho}_{i+1}^{n+1}(t). \tag{36}$$

Proof. The following equalities can easily be checked from (14), (19) and (29)

$$\tilde{\rho}_i^{n+1}(t) = \frac{n+1}{n+1-i} \frac{\tilde{g}(t)}{L_{n+1}(t)} \frac{\tilde{w}_i^{n+1}}{\tilde{w}_i^n} \tilde{\rho}_i^n(t), \quad \tilde{\rho}_{i+1}^{n+1}(t) = \frac{n+1}{i+1} \frac{\tilde{f}(t)}{L_{n+1}(t)} \frac{\tilde{w}_{i+1}^{n+1}}{\tilde{w}_i^n} \tilde{\rho}_i^n(t).$$

Then we can write

$$a_{n+1} \frac{\tilde{g}(t)}{L_{n+1}(t)} = a_{n+1} \frac{n+1-i}{n+1} \frac{\tilde{w}_i^n}{\tilde{w}_i^{n+1}} \frac{\tilde{\rho}_i^{n+1}(t)}{\tilde{\rho}_i^n(t)}, \tag{37}$$

and

$$L_{n+1}(t_0) \frac{\tilde{f}(t)}{L_{n+1}(t)} = L_{n+1}(t_0) \frac{i+1}{n+1} \frac{\tilde{w}_i^n}{\tilde{w}_{i+1}^{n+1}} \frac{\tilde{\rho}_{i+1}^{n+1}(t)}{\tilde{\rho}_i^n(t)}. \tag{38}$$

Taking into account that $L_{n+1}(t) = a_{n+1}\tilde{g}(t) + L_{n+1}(t_0)\tilde{f}(t)$ and the addition of (37) and (38), we can deduce that

$$1 = a_{n+1} \frac{n+1-i}{n+1} \frac{\tilde{w}_i^n}{\tilde{w}_i^{n+1}} \frac{\tilde{\rho}_i^{n+1}(t)}{\tilde{\rho}_i^n(t)} + L_{n+1}(t_0) \frac{i+1}{n+1} \frac{\tilde{w}_i^n}{\tilde{w}_{i+1}^{n+1}} \frac{\tilde{\rho}_{i+1}^{n+1}(t)}{\tilde{\rho}_i^n(t)}$$

and the result follows. \square

Using similar reasoning and taking into account that $L_{n+1}(t) = L_{n+1}(t_0)\hat{g}(t) + b_{n+1}\hat{f}(t)$, we can prove the following recurrence relations satisfied by the bases (24) of $\mathcal{R}^n|_{I''}$ and consequently the nested nature of these spaces.

Proposition 8. The system $(\hat{\rho}_0^n, \dots, \hat{\rho}_n^n)$ defined in (24) satisfies

$$\hat{\rho}_i^n(t) := L_{n+1}(t_0) \frac{n+1-i}{n+1} \frac{\hat{w}_i^n}{\hat{w}_i^{n+1}} \hat{\rho}_i^{n+1}(t) + b_{n+1} \frac{i+1}{n+1} \frac{\hat{w}_i^n}{\hat{w}_{i+1}^{n+1}} \hat{\rho}_{i+1}^{n+1}(t). \tag{39}$$

The following result proves recurrence relations satisfied by the functions of the system $(\tilde{\rho}_0^n, \dots, \tilde{\rho}_n^n)$ given in (19).

Proposition 9. The system $(\tilde{\rho}_0^n, \dots, \tilde{\rho}_n^n)$ given in (19) satisfies

$$\tilde{\rho}_i^n(t) := a_n \frac{\tilde{g}(t)}{L_n(t)} \tilde{\rho}_i^{n-1}(t) + L_n(t_0) \frac{\tilde{f}(t)}{L_n(t)} \tilde{\rho}_{i-1}^{n-1}(t), \quad i = 0, \dots, n. \tag{40}$$

Proof. Taking into account (32) and (33), we have

$$\begin{aligned} \sum_{i=0}^{n-1} \tilde{w}_i^n \tilde{u}_i^n(t) &= a_n \sum_{i=0}^{n-1} \tilde{w}_i^{n-1} \tilde{u}_i^{n-1}(t) \tilde{g}(t) + L_n(t_0) \sum_{i=0}^{n-1} \tilde{w}_i^{n-1} \tilde{u}_i^{n-1}(t) \tilde{f}(t) \\ &= a_n \sum_{i=0}^{n-1} \tilde{w}_i^{n-1} \tilde{u}_i^{n-1}(t) \tilde{g}(t) + L_n(t_0) \sum_{i=1}^n \tilde{w}_{i-1}^{n-1} \tilde{u}_{i-1}^{n-1}(t) \tilde{f}(t). \end{aligned} \tag{41}$$

Let us observe that the terms containing $\tilde{u}_i^{n-1}(t)\tilde{g}(t)$ or $\tilde{u}_{i-1}^{n-1}(t)\tilde{f}(t)$ are both multiples of $\tilde{u}_i^n(t)$,

$$\tilde{u}_i^{n-1}(t)\tilde{g}(t) = \binom{n-1}{i} \tilde{u}_i^n(t), \quad \tilde{u}_{i-1}^{n-1}(t)\tilde{f}(t) = \binom{n-1}{i-1} \tilde{u}_i^n(t).$$

Then, by comparing the terms in both sides in the expansion (41) with respect to the basis $\tilde{u}_0^n(t), \dots, \tilde{u}_0^n(t)$, we can write

$$\tilde{w}_i^n \frac{\tilde{u}_i^n(t)}{\omega^n(t)} = a_n \frac{\tilde{g}(t)\tilde{w}_i^{n-1}\tilde{u}_i^{n-1}(t)}{L_n(t)\omega^{n-1}(t)} + L_n(t_0) \frac{\tilde{f}(t)\tilde{w}_{i-1}^{n-1}\tilde{u}_{i-1}^{n-1}(t)}{L_n(t)\omega^{n-1}(t)}$$

and the result follows. \square

Observe that the weights w_i^n of (30) and the functions of the system $(\rho_0^n, \dots, \rho_n^n)$ can be obtained recursively by considering in (31), (36) and (40) $t_0 = b$. Moreover, for the particular choice $f(t) = t$, $g(t) = 1 - t$ and $t_0 = 1$ we obtain formula (5) in Proposition 2, formula (11) in Proposition 4 and formula (8) in Proposition 3 of Šir and Jüttler (2015).

Consider the parametric curve

$$\tilde{\gamma}(t) := \sum_{i=0}^n P_i \tilde{\rho}_i^n(t), \quad t \in [a, t_0], \tag{42}$$

where $(\tilde{\rho}_0^n, \dots, \tilde{\rho}_n^n)$ is the system defined in (19) and P_0, \dots, P_n are control points. Define the following algorithm.

Algorithm 3: de Casteljau-type algorithm for evaluation.

for $i:=0$ **to** n

$$P_i^0 := P_i$$

for $j:=1$ **to** n

for $i:=0$ **to** $n-j$

$$P_i^j(t) := a_j \frac{\tilde{g}(t)}{L_j(t)} P_i^{j-1}(t) + L_j(t_0) \frac{\tilde{f}(t)}{L_j(t)} P_{i+1}^{j-1}(t)$$

Proposition 10. *The points defined in Algorithm 3 satisfy*

$$P_i^j(t) := \sum_{k=0}^j P_{i+k} \tilde{\rho}_k^j(t), \quad t \in [a, t_0]. \tag{43}$$

In particular we have $P_0^n(t) := \tilde{\gamma}(t)$.

Proof. Let us prove (43) by induction on j . For $j = 0$, if $i = 0$, by convention $\rho_0^0(t) = 1$, and we can deduce that

$$\sum_{k=0}^0 P_0 \rho_0^0(t) = P_0 \cdot 1 = P_0 = P_0^0(t).$$

Now suppose that $P_i^{j-1}(t)$ satisfy (43). Then, by Algorithm 3, we can deduce that

$$\begin{aligned} P_i^j(t) &= a_j \frac{\tilde{g}(t)}{L_j(t)} P_i^{j-1}(t) + L_j(t_0) \frac{\tilde{f}(t)}{L_j(t)} P_{i+1}^{j-1}(t) \\ &= a_j \frac{\tilde{g}(t)}{L_j(t)} \left(\sum_{k=0}^{j-1} P_{i+k} \tilde{\rho}_k^{j-1}(t) \right) + L_j(t_0) \frac{\tilde{f}(t)}{L_j(t)} \left(\sum_{k=0}^{j-1} P_{i+k+1} \tilde{\rho}_k^{j-1}(t) \right) \\ &= \sum_{k=0}^j P_{i+k} \left(a_j \frac{\tilde{g}(t)}{L_j(t)} \tilde{\rho}_k^{j-1}(t) + L_j(t_0) \frac{\tilde{f}(t)}{L_j(t)} \tilde{\rho}_{k-1}^{j-1}(t) \right). \end{aligned}$$

Then the result follows from (40). \square

Observe that permutation of all the linear factors L_1, \dots, L_n defines $n!$ different de Casteljau-type algorithms for evaluation of these parametric rational curves (see Figs. 6 and 8).

Following a similar line of reasoning, the following recurrence for the functions of $(\hat{\rho}_0^n, \dots, \hat{\rho}_n^n)$ given in (24) can also be proved. These recurrence relations provide evaluation algorithms for parametric rational curves given in terms of these bases.

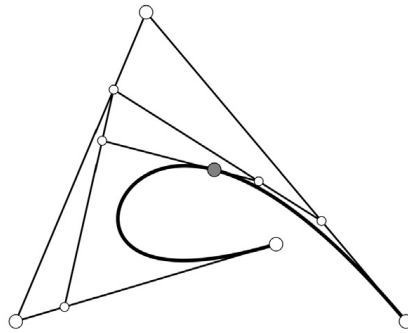


Fig. 5. Algorithm of evaluation and subdivision for curves of the Lupaş q -analogues of the Bernstein functions for $q = 3$ and $t_0 = 1/2$.

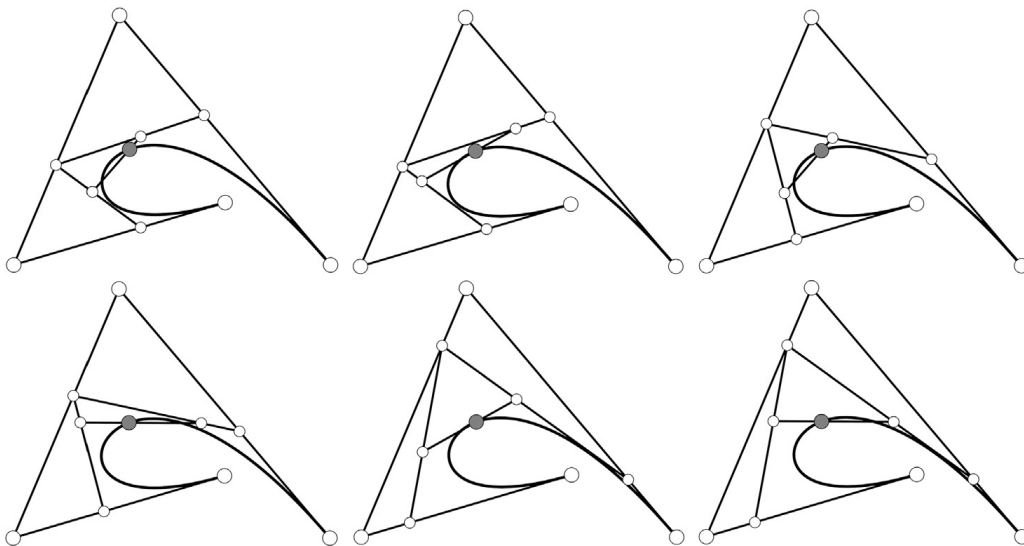


Fig. 6. Six different de Casteljau-type algorithms for the Lupaş q -analogues of the Bernstein functions of the restrictions to $[0, 1/2]$ for $q = 3$.

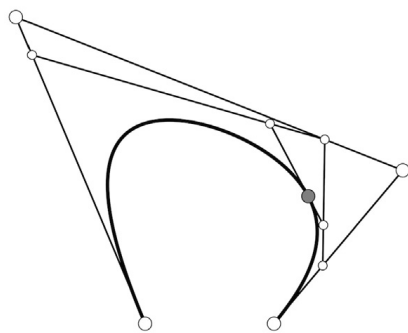


Fig. 7. Algorithm for evaluation and subdivision for hyperbolic curves with $f(t) := \sinh(\frac{1+t}{2})$, $g(t) := \sinh(\frac{1-t}{2})$, $t_0 = 1/2$ and the weights w_i^n given from (30) with $a = [5, 3, 1]$ and $b = [1, 4, 5]$.

Proposition 11. The system $(\hat{\rho}_0^n, \dots, \hat{\rho}_n^n)$ given in (24) satisfies

$$\hat{\rho}_i^n(t) := L_n(t_0) \frac{\hat{g}(t)}{L_n(t)} \hat{\rho}_i^{n-1}(t) + b_n \frac{\hat{f}(t)}{L_n(t)} \hat{\rho}_{i-1}^{n-1}(t), \quad i = 0, \dots, n. \tag{44}$$

Fig. 5 illustrates the algorithms for evaluation and subdivision for curves of the Lupaş q -analogues of the Bernstein functions defined in (25). Fig. 6 shows the six different de Casteljau-type algorithms for the Lupaş q -analogues of the Bernstein functions of the restrictions to I' defined in (14) obtained from Algorithm 3 of de Casteljau-type.

Fig. 7 illustrates the algorithm for evaluation and subdivision for hyperbolic curves. Fig. 8 shows the six different de Casteljau-type algorithms for hyperbolic functions of the restrictions to I' .

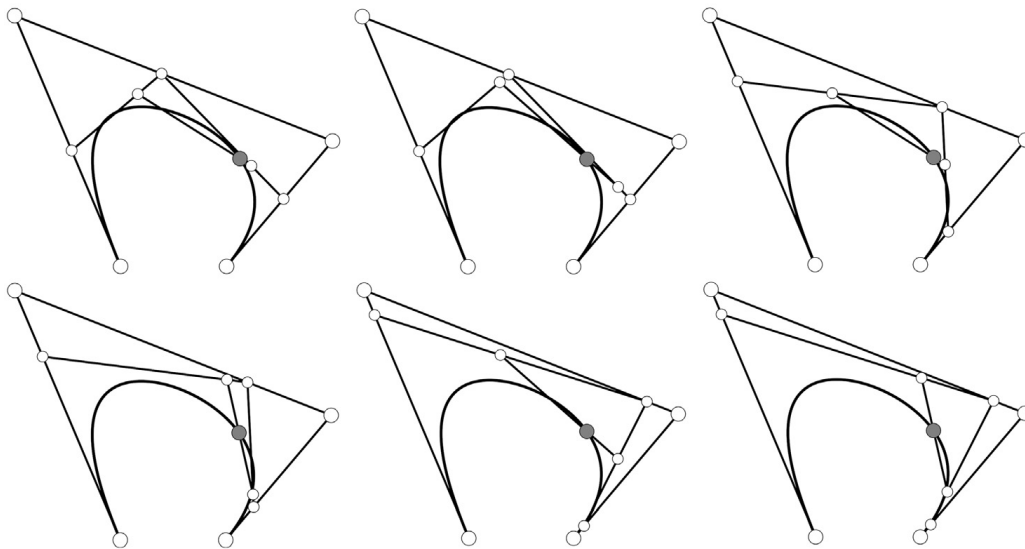


Fig. 8. Six different de Casteljau-type algorithms for hyperbolic functions of the restrictions to $[-1, 1/2]$ with $f(t) := \sinh(\frac{1+t}{2})$, $g(t) := \sinh(\frac{1-t}{2})$, $L_1(t) = 5g(t) + f(t)$, $L_2(t) = 3g(t) + 4f(t)$ and $L_3(t) = g(t) + 5f(t)$.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We thank the anonymous referees for their helpful comments and suggestions, which have improved this paper.

References

- Ando, T., 1987. Totally positive matrices. *Linear Algebra Appl.* 90, 165–219.
- Carnicer, J.M., Peña, J.M., 1993. Shape preserving representations and optimality of the Bernstein basis. *Adv. Comput. Math.* 1, 173–196.
- Carnicer, J.M., Peña, J.M., 1994. Totally positive bases for shape preserving curve design and optimality of B-splines. *Comput. Aided Geom. Des.* 11, 633–654.
- Delgado, J., Peña, J.M., 2013. Accurate computations with collocation matrices of rational bases. *Appl. Math. Comput.* 219, 4354–4364.
- Disibuyuk, C., Goldman, R., 2015. A unifying functional structure for polar forms and for functional Bernstein-Bezier curves. *J. Approx. Theory* 192, 234–249.
- Goldman, R., 1985. Polya's urn model and computer aided geometric design. *SIAM J. Algebraic Discrete Methods* 6 (1), 1–28.
- Goldman, R., 1999. The rational Bernstein bases and the multirational blossoms. *Comput. Aided Geom. Des.* 16, 701–738.
- Gonsor, D., Neamtu, M., 1994. Non-polynomial polar forms. In: *Curves and Surfaces in Geometric Design*. Chamonix Mont-Blanc, 1993. A K Peters, Wellesley, MA, pp. 193–200.
- Han, L.W., Wua, Y.S., Chua, Y., 2016. Weighted Lupaş q-Bézier curves. *J. Comput. Appl. Math.* 308, 318–329.
- Mainar, E., Peña, J.M., 1999. Corner cutting algorithms associated with optimal shape preserving representations. *Comput. Aided Geom. Des.* 16, 883–906.
- Manni, C., Pelosi, F., Sampoli, M.L., 2011. Generalized B-splines as a tool in isogeometric analysis. *Comput. Methods Appl. Mech. Eng.* 200, 867–881.
- Morin, G., Goldman, R., 2000. A subdivision scheme for Poisson curves and surfaces. *Comput. Aided Geom. Des.* 17, 813–833.
- Peña, J.M., 1997. Shape preserving representations for trigonometric polynomial curves. *Comput. Aided Geom. Des.* 14, 5–11.
- Rôt, Á., 2015. Control point based exact description of trigonometric/hyperbolic curves, surfaces and volumes. *J. Comput. Appl. Math.* 290, 74–91.
- Šir, Z., Jüttler, B., 2015. On de Casteljau-type algorithms for rational Bézier curves. *J. Comput. Appl. Math.* 288, 244–250.

ARTICLE 2

- [74] E. Mainar, J.M. Peña, B. Rubio, Accurate bidiagonal decomposition of collocation matrices of weighted φ -transformed systems, *Numerical Linear Algebra Appl.* **e2295** (2020).
<https://doi.org/10.1002/nla.2295>



Accurate bidiagonal decomposition of collocation matrices of weighted φ -transformed systems

Esmeralda Mainar¹ | Juan Manuel Peña¹ | Beatriz Rubio¹

Departamento de Matemática Aplicada/IUMA, Universidad de Zaragoza, Zaragoza, Spain

Correspondence

Beatriz Rubio, Departamento de Matemática Aplicada/IUMA, Universidad de Zaragoza, Zaragoza, Spain.
Email: brubio@unizar.es

Funding information

Feder 2014-2020, Grant/Award Number: “Construyendo Europa desde Aragón”; Gobierno de Aragón, Grant/Award Number: E41\17R; MCIU/AEI, Grant/Award Number: PGC2018-096321-B-I00

Summary

Given a system of functions, we introduce the concept of weighted φ -transformed system, which will include a very large class of useful representations in Statistics and Computer Aided Geometric Design. An accurate bidiagonal decomposition of the collocation matrices of these systems is obtained. This decomposition is used to present computational methods with high relative accuracy for solving algebraic problems with collocation matrices of weighted φ -transformed systems such as the computation of eigenvalues, singular values, and the solution of some linear systems. Numerical examples illustrate the accuracy of the performed computations.

KEYWORDS

accurate computations, bidiagonal decompositions, rational basis

1 | INTRODUCTION

In this article, we present a very general procedure for generating, from an initial system and a positive function φ , new systems of functions useful for many applications. These systems, which we call weighted φ -transformed systems, arise with relevant probability distributions. They also include important rational bases^{1,2} as well as systems belonging to spaces mixing algebraic, trigonometric, and hyperbolic polynomials, which are useful in many applications of Approximation Theory and Computer Aided Geometric Design (CAGD). The weighted φ -transformed systems inherit from the initial system its accuracy when computing with its collocation matrices.

The accurate computation with structured classes of matrices is an important issue in Numerical Linear Algebra and it is receiving increasing attention in the recent years (cf. References 3–5). For this purpose, a parametrization adapted to the structure of the considered matrices is needed. Let us recall that an algorithm can be performed with high relative accuracy (HRA) if it does not include subtractions (except of the initial data), that is, if it only includes products, divisions, sums of numbers of the same sign, and subtractions of the initial data (cf. Reference 6). Performing an algorithm with HRA is a very desirable goal because it implies that the relative errors of the computations are of the order of the machine precision, independently of the size of the condition number of the considered problem. Let us recall that a totally positive (TP) matrix has all its minors nonnegative. TP matrices arise in many applications (cf. Reference 7). It is known that, for some subclasses of TP matrices, many algebraic computations can be performed with HRA. For instance, the computation of their eigenvalues, singular values, or the solutions of linear systems $Ax = b$ such that the components of b have alternating signs (see Reference 8 and the references therein). The key tool for this purpose is provided by the algorithms of References 6 and 9 jointly with the use of a bidiagonal factorization of a nonsingular TP matrix, which can be obtained with HRA for some of those matrices. Up to now, this has been achieved with some relevant subclasses of TP matrices with applications

to CAGD (cf. References 4,10–12), to Finance (cf. Reference 13), or to Combinatorics (cf. Reference 5). In the case of CAGD, the importance of TP matrices comes from the fact that the normalized systems whose collocation matrices are TP provide shape-preserving representations.^{14,15} In Reference 12, we presented many important bases used in CAGD whose collocation matrices admit many computations with HRA.

In Section 3, we extend the analysis of Reference 12 to the more general framework of this article and assure that the algebraic computations mentioned above can be performed with HRA for the collocation matrices of weighted φ -transformed systems, assuming that the bidiagonal factorization of the corresponding collocation matrix of the initial system can be obtained with HRA and that the evaluation of φ does not require subtractions up to initial data. Our numerical examples will illustrate that the solution of linear systems and the computation of eigenvalues and singular values can be solved accurately even when the above conditions do not hold. In particular, our results can be applied to perform interpolation with high precision.

The layout of the article is as follows. Section 2 includes matrix notations and basic concepts. We recall the Neville elimination procedure, which allows us to introduce the bidiagonal factorization of a strictly TP matrix. Section 3 introduces the weighted φ -transformed systems and includes the results guaranteeing their nice computational properties. The bidiagonal factorization of the collocation matrices of the weighted φ -transformed systems is obtained. Section 4 includes many examples of weighted φ -transformed systems related to probabilistic distributions. Section 5 shows a class of rational spaces that can be generated by weighted φ -transformed systems. Curves generated by these weighted φ -transformed systems inherit geometric properties and algorithms of the traditional rational Bézier curves and so they can be considered as modeling tools in CAD/CAM systems. Finally, Section 6 includes numerical examples showing the accurate computation of eigenvalues and singular values and accurate solutions of linear systems associated with the collocation matrices of weighted φ -transformed systems.

2 | BASIC NOTATIONS AND AUXILIARY RESULTS

A matrix is *totally positive* (TP) if all its minors are nonnegative and *strictly totally positive* (STP) if they are positive.⁷

Let us now recall some basic matrix notations and results on Neville elimination. Our notation follows the notation used in References 16 and 17. Given $n \in \mathbb{N}$ and $k \in \{1, \dots, n\}$, let $Q_{k,n}$ be the set of increasing sequences of k positive integers less than or equal to n . If $\alpha, \beta \in Q_{k,n}$, we denote by $A[\alpha|\beta]$ the $k \times k$ submatrix of A containing rows of places α and columns of places β . Neville elimination^{16,17} is a procedure to make zeros in a column of a matrix by adding to a given row an appropriate multiple of the previous one. For a given nonsingular matrix $A = (a_{ij})_{1 \leq i,j \leq n}$, this elimination procedure consists of at most $n - 1$ successive major steps, resulting in the sequence of matrices:

$$A^{(1)} := A \rightarrow A^{(2)} \rightarrow \dots \rightarrow A^{(n)} = U.$$

For $1 \leq k \leq n - 1$, $A^{(k+1)} = (a_{ij}^{(k+1)})_{1 \leq i,j \leq n}$ is obtained from $A^{(k)} = (a_{ij}^{(k)})_{1 \leq i,j \leq n}$ by defining

$$\begin{cases} a_{i,k}^{(k+1)} := 0, & i = k + 1, \dots, n, \\ a_{ij}^{(k+1)} := a_{ij}^{(k)} - \frac{a_{i,k}^{(k)}}{a_{i-1,k}^{(k)}} a_{i-1,j}^{(k)} & \text{if } a_{i-1,k}^{(k)} \neq 0, \quad k + 1 \leq i, j \leq n, \end{cases}$$

so that $A^{(k+1)}$ has zeros below its main diagonal in the k first columns. Finally, U is an upper triangular matrix. The element $p_{i,j} := a_{ij}^{(j)}$, is called the (i, j) pivot of the Neville elimination of A for $1 \leq j \leq i \leq n$. The pivots $p_{i,i}$ are called diagonal pivots. The Neville elimination can be performed without row exchanges if all the pivots are nonzero and, in this case, lemma 2.6 of Reference 16 implies that $p_{i,1} = a_{i,1}$, for $1 \leq i \leq n$, and

$$p_{i,j} = \frac{\det A[i-j+1, \dots, i|1, \dots, j]}{\det A[i-j+1, \dots, i-1|1, \dots, j-1]}, \quad 1 < j \leq i \leq n. \quad (1)$$

Furthermore, the (i, j) multiplier of the Neville elimination of A is

$$m_{i,j} := \frac{a_{ij}^{(j)}}{a_{i-1,j}^{(j)}} = \frac{p_{i,j}}{p_{i-1,j}}, \quad 1 \leq j < i \leq n. \quad (2)$$

Neville elimination has been used to characterize TP and STP matrices.^{16,17} From theorem 4.1 of Reference 16 and p. 116 of Reference 17 (see also theorem 2.1 of Reference 18), a given matrix A is STP if and only if the Neville elimination of A and A^T can be performed without row exchanges, all the multipliers of the Neville elimination of A and A^T are positive, and all the diagonal pivots of the Neville elimination of A are positive.

Bidiagonal factorizations have played a crucial role to derive for TP matrices algorithms with HRA (cf. Reference 6). According to the arguments of p. 116 of Reference 17, an STP matrix $A \in \mathbb{R}^{(n+1) \times (n+1)}$ can be factorized in the form

$$A = F_n F_{n-1} \dots F_1 D G_1 \dots G_{n-1} G_n, \quad (3)$$

where F_i and G_i are the lower and upper triangular bidiagonal matrices

$$F_i = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & m_{i+1,1} & 1 & & \\ & & & \ddots & \ddots & \\ & & & & m_{n+1,n+1-i} & 1 \end{pmatrix}, \quad G_i^T = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \hat{m}_{i+1,1} & 1 & & \\ & & & \ddots & \ddots & \\ & & & & \hat{m}_{n+1,n+1-i} & 1 \end{pmatrix}, \quad (4)$$

and $D = \text{diag}(p_{1,1}, \dots, p_{n+1,n+1})$. The entries $m_{i,j}$ and $\hat{m}_{i,j}$ are the multipliers of the Neville elimination of A and A^T , respectively, and the diagonal entries $p_{i,i}$ are the diagonal pivots of the Neville elimination of A .

3 | WEIGHTED φ -TRANSFORMED SYSTEMS

Let us first introduce a key concept of this article. Let (u_0, \dots, u_n) be a system of functions defined on $I = [a, b]$, $\varphi : [a, b] \rightarrow \mathbb{R}$ a positive function, and d_0, \dots, d_n positive real values. The corresponding *weighted φ -transformed* system from (u_0, \dots, u_n) is the system $(\tilde{u}_0, \dots, \tilde{u}_n)$ of functions defined by

$$\tilde{u}_i(t) := d_i \varphi(t) u_i(t), \quad t \in [a, b], \quad i = 0, \dots, n. \quad (5)$$

Let us suppose that (u_0, \dots, u_n) is a system of functions defined on $I = [a, b]$ and $a < t_1 < \dots < t_{n+1} < b$ is a sequence of nodes such that the corresponding collocation matrix

$$A := (u_{j-1}(t_i))_{1 \leq i, j \leq n+1} \quad (6)$$

is STP. Let

$$A = F_n F_{n-1} \dots F_1 D G_1 \dots G_{n-1} G_n \quad (7)$$

be the bidiagonal factorization (3) such that F_i and G_i are the lower and upper triangular bidiagonal matrices of the form (4) and D is a diagonal matrix.

The following result proves that the collocation matrix of the corresponding weighted φ -transformed system $(\tilde{u}_0, \dots, \tilde{u}_n)$ at nodes $a < t_1 < \dots < t_{n+1} < b$

$$\tilde{A} := (\tilde{u}_{j-1}(t_i))_{1 \leq i, j \leq n+1} = (d_{j-1} \varphi(t_i) u_{j-1}(t_i))_{1 \leq i, j \leq n+1} \quad (8)$$

is also STP and obtains its bidiagonal factorization (3) from the factorization (7) of the collocation matrix A given in Equation (6).

Theorem 1. *The collocation matrix (8) is STP and it can be factorized as*

$$\tilde{A} = \tilde{F}_n \tilde{F}_{n-1} \dots \tilde{F}_1 \tilde{D} \tilde{G}_1 \dots \tilde{G}_{n-1} \tilde{G}_n, \quad (9)$$

where \tilde{F}_i and \tilde{G}_i are the lower and upper bidiagonal matrices of the form

$$\tilde{F}_i = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & r_{i+1,1} & 1 & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & r_{n+1,n+1-i} & 1 \end{pmatrix}, \quad \tilde{G}_i^T = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & \hat{r}_{i+1,1} & 1 & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & \hat{r}_{n+1,n+1-i} & 1 \end{pmatrix} \quad (10)$$

and $\tilde{D} = \text{diag}(q_{1,1}, \dots, q_{n+1,n+1})$. The entries $r_{i,j}$, $\hat{r}_{i,j}$, and $q_{i,i}$ are given by

$$\begin{aligned} r_{i,j} &= \frac{\varphi(t_i)}{\varphi(t_{i-1})} m_{i,j}, & \hat{r}_{i,j} &= \frac{d_{i-1}}{d_{i-2}} \hat{m}_{i,j}, & 1 \leq j < i \leq n+1, \\ q_{i,i} &= d_{i-1} \varphi(t_i) p_{i,i}, & & & 1 \leq i \leq n+1, \end{aligned} \quad (11)$$

where $m_{i,j}$, $\hat{m}_{i,j}$, and $p_{i,i}$ are the entries of the matrices of the bidiagonal factorization (7) of the collocation matrix A defined in Equation (6).

Proof. Observe that

$$\tilde{A} = \text{diag}(\varphi(t_1), \dots, \varphi(t_{n+1})) A \text{diag}(d_0, \dots, d_n).$$

Hence, taking into account the positivity of φ and the coefficients d_0, \dots, d_n , we deduce that $\text{diag}(\varphi(t_1), \dots, \varphi(t_{n+1}))$ and $\text{diag}(d_0, \dots, d_n)$ are nonsingular and TP matrices. Since A is STP and, by theorem 3.1 of Reference 7, the product of STP matrices by a nonsingular TP matrix is a STP matrix, we conclude that \tilde{A} is STP. In order to compute the pivots and the multipliers of the Neville elimination of \tilde{A} , we need to obtain its minors with j initial consecutive columns and j consecutive rows starting with row $i-j+1$.

Let $1 \leq j \leq i \leq n+1$. For any $1 \leq k \leq j$, each entry of the k th row of the matrix $\tilde{A}[i-j+1, \dots, i|1, \dots, j]$ has as common factor $\varphi(t_{i-j+k})$ and each entry of the k th column of the matrix $\tilde{A}[i-j+1, \dots, i|1, \dots, j]$ has as common factor d_{k-1} . Therefore we can write

$$\tilde{A}[i-j+1, \dots, i|1, \dots, j] = D_1 A[i-j+1, \dots, i|1, \dots, j] D_2,$$

where $D_1 := \text{diag}(\varphi(t_{i-j+1}), \dots, \varphi(t_i))$ and $D_2 := \text{diag}(d_0, \dots, d_{j-1})$. Using properties of determinants, we can write

$$\det \tilde{A}[i-j+1, \dots, i|1, \dots, j] = \prod_{k=1}^j (d_{k-1} \varphi(t_{i-j+k})) \det A[i-j+1, \dots, i|1, \dots, j].$$

Let us denote by $\tilde{p}_{i,j}$ the pivot obtained in the Neville elimination procedure of \tilde{A} . Taking into account the previous formula and Equation (1), we deduce that

$$\tilde{p}_{i,j} = \frac{\det \tilde{A}[i-j+1, \dots, i|1, \dots, j]}{\det \tilde{A}[i-j+1, \dots, i-1|1, \dots, j-1]} = d_{j-1} \varphi(t_i) p_{i,j}, \quad (12)$$

where $p_{i,j}$ is the pivot obtained in the Neville elimination procedure of the matrix A . Observe that, for the particular case $i=j$, we have

$$q_{i,i} = d_{i-1} \varphi(t_i) p_{i,i}, \quad 1 \leq i \leq n+1.$$

Finally, from formulas (12) and (2),

$$r_{i,j} = \frac{\tilde{p}_{i,j}}{\tilde{p}_{i-1,j}} = \frac{d_{j-1} \varphi(t_i) p_{i,j}}{d_{j-1} \varphi(t_{i-1}) p_{i-1,j}} = \frac{\varphi(t_i)}{\varphi(t_{i-1})} m_{i,j}, \quad 1 \leq j < i \leq n+1.$$

Analogously, for any $1 \leq k \leq j$, each entry of the k th row of \tilde{A}^T has as common factor d_{k-1} and each entry of the k th column of the matrix \tilde{A}^T has as common factor $\varphi(t_k)$. Then

$$\tilde{A}^T[i-j+1, \dots, i|1, \dots, j] = \hat{D}_1 A^T[i-j+1, \dots, i|1, \dots, j] \hat{D}_2,$$

where $\hat{D}_1 := \text{diag}(d_{i-j}, \dots, d_{i-1})$ and $\hat{D}_2 := \text{diag}(\varphi(t_1), \dots, \varphi(t_j))$. Using properties of determinants, we can write

$$\det \tilde{A}^T[i-j+1, \dots, i|1, \dots, j] = \prod_{k=1}^j d_{i-k} \varphi(t_k) \det A^T[i-j+1, \dots, i|1, \dots, j].$$

Taking into account the previous formula, (1), and (2) we deduce that

$$\frac{\det \tilde{A}^T[i-j+1, \dots, i|1, \dots, j]}{\det \tilde{A}^T[i-j+1, \dots, i-1|1, \dots, j-1]} = d_{i-1} \varphi(t_j) p_{i,j}.$$

Finally, taking into account Equation (2), we conclude

$$\hat{r}_{i,j} = \frac{d_{i-1}}{d_{i-2}} \hat{m}_{i,j}, \quad 1 \leq j < i \leq n+1. \quad \blacksquare$$

We say that a system of functions (u_0, \dots, u_n) defined on $I = [a, b]$ is TP (respectively, STP) if all its collocation matrices (6) in I are TP (respectively, STP). As a consequence of the previous result, we have that weighted φ -transformed systems inherit the property of being STP, as stated in the following result.

Corollary 1. *Let (u_0, \dots, u_n) be a STP system of functions defined on $I = [a, b]$. Then any weighted φ -transformed system $(\tilde{u}_0, \dots, \tilde{u}_n)$ given by Equation (5) is STP.*

Remark 1. Observe that, if the evaluation of φ can be performed by arithmetic operations and it does not require subtractions (except for the initial data), the entries of the bidiagonal factorization of Theorem 1 can be obtained from the bidiagonal factorization of Equation (7) without performing subtractions. Therefore, if the bidiagonal factorization of Equation (7) can be performed with HRA, then the bidiagonal factorization of Theorem 1 can be also performed with HRA. It is known that the bidiagonal factorization (3) of the collocation matrices associated with some important bases used in CAGD can be performed with HRA.¹² In consequence, the bidiagonal factorization of the collocation matrices of their corresponding weighted φ -transformed systems can also be performed with HRA and we can apply the algorithms presented in References 6 and 9 to perform many algebraic computations with HRA. For instance, the computation of their eigenvalues, singular values, or the solutions of some linear systems is associated with these collocation matrices.

4 | WEIGHTED φ -TRANSFORMED PROBABILITY DISTRIBUTIONS

A probability distribution is a mathematical function that provides the probabilities of occurrence of different possible results in an experiment. In this section, we are going to see some interesting bases that can be defined from probability distributions and can be considered as weighted φ -transformed systems from other bases whose collocation matrices are STP and can be factorized as in Equation (9).

The *binomial distribution* is frequently used to model the number of successes in a sample of size n . If the probability of success is $t \in [0, 1]$, $n \in \mathbb{N}$ is the number of trials, and $k \in \mathbb{N}$ is the number of successes, then the probability of getting exactly k successes in n trials is given by

$$P(k \text{ successes in } n \text{ trials}) = \binom{n}{k} t^k (1-t)^{n-k}, \quad k = 0, 1, \dots, n.$$

The binomial functions coincide with the Bernstein polynomials of degree n ,

$$B_k^n(t) := \binom{n}{k} t^k (1-t)^{n-k}, \quad t \in [0, 1], \quad k = 0, 1, \dots, n.$$

The collocation matrix of the Bernstein basis (B_0^n, \dots, B_n^n) for any sequence of parameters $0 < t_1 < \dots < t_{n+1} < 1$ is STP and its corresponding bidiagonal factorization (3) and (4) is given by

$$\begin{aligned} m_{i,j} &= \frac{(1-t_i)^{n-j+1}(1-t_{i-j})}{(1-t_{i-1})^{n-j+2}} \frac{\prod_{k=1}^{j-1} (t_i - t_{i-k})}{\prod_{k=2}^j (t_{i-1} - t_{i-k})}, \quad 1 \leq j < i \leq n+1, \\ \hat{m}_{i,j} &= \frac{n-i+2}{i-1} \frac{t_j}{(1-t_j)}, \quad 1 \leq j < i \leq n+1, \\ p_{i,i} &= \binom{n}{i-1} \frac{(1-t_i)^{n-i+1}}{\prod_{k=1}^{i-1} (1-t_k)} \prod_{k=1}^{i-1} (t_i - t_k), \quad 1 \leq i \leq n+1 \end{aligned} \quad (13)$$

(see Reference 19 or theorem 3 of Reference 12). The bidiagonal factorization of the collocation matrix of the Bernstein basis can be performed with HRA. In Reference 10, accurate computations to solve algebraic problems associated with collocation matrices of Bernstein bases are shown. Urn models extend the binomial distribution. In Reference 20, it is shown the connection between these models and CAGD and Approximation Theory, in particular with splines.

The *negative binomial distribution* is an appropriate model to treat those processes in which a certain trial is repeated until a certain number of favorable results are achieved for the first time. If the probability of failure is $t \in [0, 1]$, r is the number of failures, and k is the number of successes, then the probability of r failures up to obtain k successes (at least 1 success) is given by

$$P(r \text{ failures up to } k \text{ successes}) = \binom{k+r-1}{r} t^r (1-t)^k.$$

Let us observe that, if $n = k + r - 1$, the negative binomial basis (b_0, \dots, b_n) defined by $b_r(t) := \binom{n}{r} t^r (1-t)^{n-r+1}$, $r = 0, \dots, n$, can be considered as a weighted φ -transformed system from the Bernstein basis with $\varphi(t) = 1-t$ and $d_i = 1$ for $i = 0, \dots, n$. Then, using Corollary 1, we deduce that the negative binomial basis is STP on $(0, 1)$ and, using Theorem 1 and the bidiagonal factorization (3), (4), (13) of the collocation matrix of the Bernstein basis at $0 < t_1 < \dots < t_{n+1} < 1$, the corresponding bidiagonal factorization of the collocation matrix of the negative binomial basis is given by

$$\begin{aligned} r_{i,j} &= \frac{(1-t_i)^{n-j+2}(1-t_{i-j})}{(1-t_{i-1})^{n-j+3}} \frac{\prod_{k=1}^{j-1} (t_i - t_{i-k})}{\prod_{k=2}^j (t_{i-1} - t_{i-k})}, \quad 1 \leq j < i \leq n+1, \\ \hat{r}_{i,j} &= \frac{n-i+2}{i-1} \frac{t_j}{(1-t_j)}, \quad 1 \leq j < i \leq n+1, \\ q_{i,i} &= \binom{n}{i-1} \frac{(1-t_i)^{n-i+2}}{\prod_{k=1}^{i-1} (1-t_k)} \prod_{k=1}^{i-1} (t_i - t_k), \quad 1 \leq i \leq n+1. \end{aligned} \quad (14)$$

Let us remark that the evaluation of $\varphi(t) = 1-t$ does not include subtractions (except for the initial data). Hence, the bidiagonal factorization (3), (4), (14) of the collocation matrix of the negative binomial basis can be also performed with HRA. Section 6 will show accurate results obtained when computing their eigenvalues, singular values, or the solutions of some linear systems associated with these collocation matrices, using the bidiagonal factorization (3), (4), (14), and the algorithms presented in References 6 and 9.

The *geometric distribution* has applications in population and econometric models. If the probability of success is $t \in [0, 1]$ and k is the number of failures, then the probability of k failures up to obtain a success is given by

$$P(k \text{ failures until a success}) := (1-t)^k t.$$

For a given $n \in \mathbb{N}$, the geometric basis functions $b_k(t) := (1-t)^k t$, $k = 0, \dots, n$ can be considered as a weighted φ -transformed system from the basis $(1, 1-t, \dots, (1-t)^n)$ with $\varphi(t) = t$ and $d_i = 1$ for $i = 0, \dots, n$. The monomial basis

$(1, t, \dots, t^n)$ is STP on $(0, \infty)$ and the bidiagonal factorization (3), (4) of its collocation matrix at $0 < t_0 < \dots < t_{n+1} < 1$ is given by

$$\begin{aligned} m_{i,j} &= \frac{\prod_{k=1}^{j-1} (t_i - t_{i-k})}{\prod_{k=2}^j (t_{i-1} - t_{i-k})}, \quad \hat{m}_{i,j} = t_j, \quad 1 \leq j < i \leq n+1, \\ p_{i,i} &= \prod_{k=1}^{i-1} (t_i - t_k), \quad 1 \leq i \leq n+1 \end{aligned} \quad (15)$$

(see References 10,21 or theorem 3 of Reference 12).

Using (15) and Theorem 1, it can be deduced that the bidiagonal decomposition of the collocation matrix of the geometric basis (b_0, \dots, b_n) is given by

$$\begin{aligned} r_{i,j} &= \frac{t_i}{t_{i-1}} \frac{\prod_{k=1}^{j-1} (t_{i-k} - t_i)}{\prod_{k=2}^j (t_{i-k} - t_{i-1})}, \quad \hat{r}_{i,j} = 1 - t_j, \quad 1 \leq j < i \leq n+1, \\ q_{i,i} &= t_i \prod_{k=1}^{i-1} (t_k - t_i), \quad 1 \leq i \leq n+1. \end{aligned} \quad (16)$$

Since the evaluation of $\varphi(t) = t$ does not include subtractions, the bidiagonal factorization (3), (4), (16) of the collocation matrix of the geometric basis for any sequence of parameters $0 < t_{n+1} < t_n < \dots < t_1 < 1$ can be also performed with HRA. Section 6 will show accurate results obtained when computing their eigenvalues, singular values, or the solutions of some linear systems associated with these collocation matrices using the bidiagonal factorization (3), (4), (16), and the algorithms presented in References 6 and 9.

The *Poisson distribution* is popular for modeling the number of times an event occurs in an interval of time or space. An event can occur $k = 0, 1, 2, \dots$ times in an interval. If the average number of events in an interval, also called the rate parameter, is designated by t , then the probability of observing k events in an interval is given by

$$P(k \text{ events in interval}) = \frac{t^k}{k!} e^{-t}.$$

The Poisson basis functions $b_k(t) := \frac{t^k}{k!} e^{-t}$, $k \in \mathbb{N}$, are the limit as n tends to infinity of the Bernstein basis of degree n over the interval $[0, n]$, that is,

$$b_k(t) = \lim_{n \rightarrow \infty} B_k^n(t/n), \quad B_k^n(t) = \binom{n}{k} t^k (1-t)^{n-k}, \quad t \in [0, 1],$$

and they also play a useful role in CAGD.²² For a given $n \in \mathbb{N}$, the Poisson basis (b_0, \dots, b_n) can be considered as a weighted φ -transformed system from the monomial basis $(1, t, \dots, t^n)$ with $\varphi(t) = e^{-t}$ and $d_i = 1/i!$, $i = 0, \dots, n$. Then, using Corollary 1, we deduce that the Poisson basis is STP on $(0, \infty)$ and, taking into account (15) and Theorem 1, we deduce that the bidiagonal factorization (3), (4) of the collocation matrix of the Poisson basis at positive values $t_1 < \dots < t_{n+1}$ is given by

$$\begin{aligned} r_{i,j} &= e^{t_{i-1}-t_i} \frac{\prod_{k=1}^{j-1} (t_i - t_{i-k})}{\prod_{k=2}^j (t_{i-1} - t_{i-k})}, \quad \hat{r}_{i,j} = \frac{1}{i-1} t_j, \quad 1 \leq j < i \leq n+1, \\ q_{i,i} &= \frac{e^{-t_i}}{(i-1)!} \prod_{k=1}^{j-1} (t_i - t_k), \quad 1 \leq i \leq n+1. \end{aligned} \quad (17)$$

Let us observe that the computation with HRA of the bidiagonal decomposition (3), (4), (17) should require the evaluation with HRA of the involved exponential function. Although this cannot be guaranteed, Section 6 will show that accurate algebraic computations with the collocation matrices associated with these nonpolynomial bases can be performed.

5 | RATIONAL WEIGHTED φ -TRANSFORMED SYSTEMS

Given a system (u_0, \dots, u_n) of functions defined on I and positive values d_0, \dots, d_n such that $\sum_{k=0}^n d_k u_k(t) \neq 0$, for all $t \in I$, the system (r_0, \dots, r_n) defined by

$$r_i(t) := \frac{d_i u_i(t)}{\sum_{k=0}^n d_k u_k(t)}, \quad i = 0, \dots, n,$$

satisfies $\sum_{i=0}^n r_i(t) = 1, \forall t \in I$, and generates a new space of rational functions. If (u_0, \dots, u_n) is TP then $\sum_{k=0}^n d_k u_k(t) > 0, \forall t \in I$, and (r_0, \dots, r_n) can be considered as a particular weighted φ -transformed system with

$$\varphi(t) := \frac{1}{\sum_{k=0}^n d_k u_k(t)}, \quad t \in I. \quad (18)$$

Given $t_1 < \dots < t_{n+1}$ in I such that the corresponding collocation matrix of (u_0, \dots, u_n) is STP, by Theorem 1, we deduce that the corresponding collocation matrix of (r_0, \dots, r_n) is also STP and, by considering (18) in Theorem 1, we can obtain its bidiagonal factorization (3), (4) from the corresponding bidiagonal factorization of the collocation matrix of (u_0, \dots, u_n) . It is important to notice that, by Remark 1, this bidiagonal factorization can be frequently used to perform algebraic calculations and interpolation with HRA. The particular cases of rational Bernstein bases and rational Said-Ball bases were considered in Reference 18.

Now we shall consider nested spaces generated by a general class of rational weighted φ -transformed systems admitting degree elevation and de Casteljau-type evaluation algorithms.

Let us suppose that $I = [a, b]$ and $f, g : I \rightarrow \mathbb{R}$ are nonnegative continuous functions such that $f(t) \neq 0, g(t) \neq 0, \forall t \in (a, b)$. Let us define the system

$$(u_0^n, \dots, u_n^n), \quad u_i^n(t) := \binom{n}{i} f^i(t) g^{n-i}(t), \quad t \in [a, b], \quad i = 0, \dots, n. \quad (19)$$

Following the approach of Reference 1, for the particular case of rational Bernstein functions, let us consider linear factors $L_i(t) = a_i g(t) + b_i f(t)$ defined by positive values a_i and $b_i, i \in \mathbb{Z}_+$, and

$$\omega^n(t) := L_1(t) \cdot \dots \cdot L_n(t). \quad (20)$$

It can be easily checked that $\omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t)$ where

$$w_i^n = \frac{1}{\binom{n}{i}} \left(\sum_{\substack{K \cup L = \{1, \dots, n\} \\ |K| = (n-i), |L| = i}} \prod_{k \in K} a_k \prod_{l \in L} b_l \right). \quad (21)$$

The positivity of a_i and b_i guarantees that $\omega_i^n > 0$ and $\omega^n(t) > 0, \forall t \in (a, b)$. Let us now denote by $(\rho_0^n, \dots, \rho_n^n)$ the weighted $1/\omega^n$ -transformed system corresponding to the weights w_0^n, \dots, w_n^n given in Equation (21)

$$\rho_i^n(t) := w_i^n \frac{1}{\omega^n(t)} u_i^n(t), \quad i = 0, \dots, n. \quad (22)$$

This system spans the space of rational functions with common denominator $\omega^n(t)$,

$$\mathcal{R}^n := \text{span}\{\rho_i^n(t) | i = 0, \dots, n\} = \{u(t)/\omega^n(t) | u(t) \in \mathcal{U}^n\},$$

where \mathcal{U}^n is the space of functions generated by the basis (19).

Let us observe that proposition 2 of Reference 1 establishes the following recurrence relation satisfied by the weights (21)

$$w_i^n = a_n \frac{(n-i)}{n} w_i^{n-1} + b_n \frac{i}{n} w_{i-1}^{n-1}, \quad 0 \leq i \leq n. \tag{23}$$

On the other hand, by replacing in propositions 3 and 4 of Reference 1 the functions t and $1-t$ by $f(t)$ and $g(t)$, respectively, one can easily deduce the following relations satisfied by the functions of weighted $1/\omega^n$ -transformed systems

$$\begin{aligned} \rho_i^n(t) &= a_n \frac{g(t)}{L_n(t)} \rho_i^{n-1}(t) + b_n \frac{f(t)}{L_n(t)} \rho_{i-1}^{n-1}(t), \quad i = 0, \dots, n, \\ \rho_i^n(t) &= a_{n+1} \frac{n+1-i}{n+1} \frac{w_i^n}{w_i^{n+1}} \rho_i^{n+1}(t) + b_{n+1} \frac{i+1}{n+1} \frac{w_i^n}{w_{i+1}^{n+1}} \rho_{i+1}^{n+1}(t), \quad i = 0, \dots, n. \end{aligned}$$

These relations guarantee the nested nature of the generated spaces, that is, $\mathcal{R}^n \subset \mathcal{R}^{n+1}$, and allow the definition of degree elevation and de Casteljau-type algorithms for the evaluation of parametric curves

$$\gamma(t) = \sum_{i=0}^n P_i \rho_i^n(t), \quad t \in [a, b].$$

Theorem 2 of Reference 12 proves that, given nonnegative functions $f, g : I \rightarrow \mathbb{R}$ such that $f(t) \neq 0, g(t) \neq 0, \forall t \in (a, b)$ and f/g is a strictly increasing function, then

$$A := \left(\binom{n}{j-1} f^{j-1}(t_i) g^{n-j+1}(t_i) \right)_{1 \leq i, j \leq n+1}, \quad a < t_1 < \dots < t_{n+1} < b, \tag{24}$$

is STP.

Moreover, in theorem 3 of Reference 12, the following bidiagonal decomposition (3) of the collocation matrices (24) was deduced

$$A = F_n F_{n-1} \dots F_1 D G_1 \dots G_{n-1} G_n, \tag{25}$$

where F_i and $G_i, 1 \leq i \leq n$, are the lower and upper triangular bidiagonal matrices of the form (4) and $D = \text{diag}(p_{1,1}, \dots, p_{n+1,n+1})$. The entries $m_{i,j}, \hat{m}_{i,j}$ and $p_{i,i}$ are given by

$$\begin{aligned} m_{i,j} &= \frac{g^{n-j+1}(t_i) g(t_{i-j})}{g^{n-j+2}(t_{i-1})} \frac{\prod_{k=1}^{j-1} (f(t_i) g(t_{i-k}) - f(t_{i-k}) g(t_i))}{\prod_{k=2}^j (f(t_{i-1}) g(t_{i-k}) - f(t_{i-k}) g(t_{i-1}))}, \quad 1 \leq j < i \leq n+1, \\ \hat{m}_{i,j} &= \frac{n-i+2}{i-1} \frac{f(t_j)}{g(t_j)}, \quad 1 \leq j < i \leq n+1, \\ p_{i,i} &= \binom{n}{i-1} \frac{g^{n-i+1}(t_i)}{\prod_{k=1}^{i-1} g(t_k)} \prod_{k=1}^{i-1} (f(t_i) g(t_k) - f(t_k) g(t_i)), \quad 1 \leq i \leq n+1. \end{aligned} \tag{26}$$

Using Corollary 1, we deduce that the corresponding weighted $1/\omega^n$ -transformed systems (22) are STP on (a, b) and then are of interest in CAGD and have shape preserving properties. According to Theorem 1, the collocation matrix \tilde{A} of the weighted $1/\omega^n$ -transformed systems (22) corresponding to $a < t_1 < \dots < t_{n+1} < b$ is STP and can be factorized as

$$\tilde{A} = \tilde{F}_n \tilde{F}_{n-1} \dots \tilde{F}_1 \tilde{D} \tilde{G}_1 \dots \tilde{G}_{n-1} \tilde{G}_n, \tag{27}$$

where \tilde{F}_i and $\tilde{G}_i, 1 \leq i \leq n$, are the lower and upper triangular bidiagonal matrices of the form (10) and $\tilde{D} = \text{diag}(q_{1,1}, \dots, q_{n+1,n+1})$. The off-diagonal entries $r_{i,j}, \hat{r}_{i,j}$ of \tilde{F}_i and \tilde{G}_i , respectively, and the diagonal entries $q_{i,i}$ of \tilde{D} are

$$\begin{aligned}
 r_{i,j} &= \frac{\omega^n(t_{i-1})}{\omega^n(t_i)} m_{i,j}, \quad \hat{r}_{i,j} = \frac{w_{i-1}^n}{w_{i-2}^n} \hat{m}_{i,j}, \quad 1 \leq j < i \leq n + 1, \\
 q_{i,i} &= \frac{w_{i-1}^n}{\omega^n(t_i)} p_{i,i}, \quad 1 \leq i \leq n + 1,
 \end{aligned}
 \tag{28}$$

where ω^n and w_i^n are defined in (20) and (21), respectively, and $m_{i,j}, \hat{m}_{i,j}, p_{i,i}$ are the entries given in Equation (26).

Let us observe that by Remark 1, if the evaluation of f and g does not require subtractions (except for the initial data) and the computation of (26) can be performed with HRA, then weighted $1/\omega^n$ -transformed systems guarantee excellent computational properties since many algebraic computations associated with \tilde{A} can be performed with HRA.

Let us now consider some interesting examples that can be obtained by considering $f(t) = t, g(t) = 1 - t, t \in [0, 1]$. In this case the functions u_i^n defined in Equation (19) coincide with the Bernstein polynomials

$$u_i^n(t) = B_i^n(t) = \binom{n}{i} t^i (1 - t)^{n-i}, \quad i = 0, \dots, n.
 \tag{29}$$

For the choice $a_i = a, b_i = b, 1 \leq i \leq n$, we have $w_i^n = a^{n-i} b^i, 0 \leq i \leq n$. In this case, $\omega^n(t) = (a(1 - t) + bt)^n$ and the corresponding weighted $1/\omega^n$ -transformed systems are Bernstein polynomials composed with a rational reparametrization of degree 1 that maps the boundaries of the interval $[0, 1]$ onto itself.¹ In fact

$$\rho_i^n(t) = B_i^n \left(\frac{bt}{a(1-t) + bt} \right), \quad i = 0, \dots, n.$$

Using the bidiagonal factorization given in Equations (26) and (28), we can obtain the coefficients of the bidiagonal factorization (27) of the collocation matrices of this basis.

$$\begin{aligned}
 r_{i,j} &= \frac{(a(1 - t_{i-1}) + bt_{i-1})^n (1 - t_i)^{n-j+1} (1 - t_{i-j}) \prod_{k=1}^{j-1} (t_i - t_{i-k})}{(a(1 - t_i) + bt_i)^n (1 - t_{i-1})^{n-j+2} \prod_{k=2}^j (t_{i-1} - t_{i-k})}, \\
 \hat{r}_{i,j} &= \frac{n - i + 2b}{i - 1} \frac{t_j}{a(1 - t_j)}, \quad 1 \leq j < i \leq n + 1, \\
 q_{i,i} &= a^{n-i+1} b^{i-1} \binom{n}{i-1} \frac{(1 - t_i)^{n-i+1}}{(a(1 - t_i) + bt_i)^n} \prod_{k=1}^{i-1} \frac{(t_i - t_k)}{(1 - t_k)}, \quad 1 \leq i \leq n + 1.
 \end{aligned}$$

Let us recall that given a real number $q > 0$ and any nonnegative integer k , the q -integer $[k]$ is defined as

$$[k] := \begin{cases} (1 - q^k)/(1 - q), & q \neq 1, \\ k, & q = 1, \end{cases}$$

and the q -factorial $[k]!$ as

$$[k]! := \begin{cases} [k][k-1] \dots [1], & k \geq 1, \\ 1, & k = 0. \end{cases}$$

For integers $0 \leq k \leq n$, the q -binomial coefficient $[nk]$ is defined by

$$\begin{bmatrix} n \\ 0 \end{bmatrix} := 1, \quad \begin{bmatrix} n \\ k \end{bmatrix} := \frac{[n][n-1] \dots [n-k+1]}{[k]!} = \frac{[n]!}{[k]![n-k]!} \quad k > 0.$$

The Lupaş q -analogues of the Bernstein functions of degree n (cf. Reference 2) are the rational Bernstein functions

$$\rho_i^n(t) := \frac{a_{n,i}(t)}{w_n(t)}, \quad i = 0, \dots, n$$

with

$$a_{n,i}(t) := \binom{n}{i} q^{i(i-1)/2} t^i (1-t)^{n-i}, \quad w_n(t) := \sum_{i=0}^n a_{n,i}(t) = \prod_{j=1}^n (1-t + q^{j-1}t).$$

Clearly, this basis is a weighted $1/\omega^n$ -transformed system (22) where the weights $w_i^n = \binom{n}{i} q^{i(i-1)/2}$ can be obtained from (21) for the particular choice $a_i = 1$ and $b_i = q^{i-1}$, $i = 1, \dots, n$.¹ The bidiagonal factorization (28) of its collocation matrices coincides with the obtained in Reference 8.

Now, by considering positive weights d_0, \dots, d_n , we can define the weighted Lupaş q -analogue of Bernstein functions of degree n as

$$r_{n,i}(t; q) := \frac{d_i a_{n,i}(t)}{\sum_{k=0}^n d_k a_{n,k}(t)}, \quad i = 0, \dots, n.$$

Using the bidiagonal factorization given in Equations (26) and (28), we can obtain the coefficients of the bidiagonal factorization (27) of the collocation matrices of weighted Lupaş q -analogue of Bernstein functions as follows:

$$\begin{aligned} r_{i,j} &= \frac{\sum_{k=0}^n d_k a_{n,k}(t_{i-1}) (1-t_i)^{n-j+1} (1-t_{i-j})}{\sum_{k=0}^n d_k a_{n,k}(t_i) (1-t_{i-1})^{n-j+2}} \frac{\prod_{k=1}^{j-1} (t_i - t_{i-k})}{\prod_{k=2}^j (t_{i-1} - t_{i-k})}, \\ \hat{r}_{i,j} &= \frac{d_{i-1}}{d_{i-2}} \frac{n-i+2}{i-1} \frac{1-q^{n-i+2}}{1-q^{i-1}} q^{i-2} \frac{t_j}{1-t_j}, \quad 1 \leq j < i \leq n+1, \\ q_{i,i} &= d_{i-1} \binom{n}{i-1} q^{(i-1)(i-2)/2} \frac{(1-t_i)^{n-i+1}}{\sum_{k=0}^n d_k a_{n,k}(t_i)} \prod_{k=1}^{i-1} \frac{(t_i - t_k)}{(1-t_k)}, \quad 1 \leq i \leq n+1. \end{aligned}$$

Finally, let us observe that there are other interesting choices of functions $f(t)$ and $g(t)$ satisfying conditions of theorem 2 of Reference 12. We can consider $f(t) := t^2$ and $g(t) := 1 - t^2$, $t \in [0, 1]$. In this case, the basis (19) is the basis with optimal shape preserving properties of the space $\langle 1, t^2, \dots, t^{2n} \rangle$ of even polynomials of degree less than or equal to $2n$ on $[0, 1]$.

Another particular case can be given by considering the functions

$$f(t) = \sin^2(t/2) = \frac{1 - \cos(t)}{2}, \quad g(t) = \cos^2(t/2) = \frac{1 + \cos(t)}{2}, \quad t \in I = [0, \pi].$$

In Reference 15, it was proved that the system (19) is the basis with optimal shape preserving properties of the space of even trigonometric polynomials given by $\langle 1, \cos(t), \cos(2t), \dots, \cos(nt) \rangle$.

Now, let us consider $0 < \Delta < \pi/2$ and

$$f(t) := \sin\left(\frac{\Delta+t}{2}\right), \quad g(t) := \sin\left(\frac{\Delta-t}{2}\right), \quad t \in I = [-\Delta, \Delta].$$

For $n = 2m$, the system (19) is a basis that coincides, up to a positive scaling, with the basis with optimal shape preserving properties of the space $\langle 1, \cos(t), \sin(t), \dots, \cos(mt), \sin(mt) \rangle$ of trigonometric polynomials of degree less than or equal to m on I (see section 3 of Reference 23).

Finally, for any $\Delta > 0$, we can also consider

$$f(t) = \sinh\left(\frac{\Delta+t}{2}\right), \quad g(t) := \sinh\left(\frac{\Delta-t}{2}\right), \quad t \in I = [-\Delta, \Delta].$$

For $n = 2m$, the system (19) is a basis with shape-preserving properties of the space $\langle 1, e^t, e^{-t}, \dots, e^{mt}, e^{-mt} \rangle$ of hyperbolic polynomials of degree less than or equal to m on I .

Taking into account that curves generated by the corresponding weighted $1/\omega^n$ -transformed systems also inherit algorithms of the traditional rational Bézier curves, they can be considered as modeling tools in CAD/CAM systems. Trigonometric and hyperbolic curves are getting considerable importance since they provide the opportunity to construct

conics, cylinders and surfaces of revolution, catenary, and so on. Shape-preserving rational trigonometric interpolation is very important in scientific data visualization and has been applied in other fields such as engineering, biology, chemistry, medical, and social sciences (see Reference 24 and the references therein).

In the next section, we are going to illustrate accurate computations with collocation matrices of the considered weighted φ -transformed systems.

6 | NUMERICAL EXPERIMENTS

In Reference 6, assuming that the multipliers and diagonal pivots of the Neville elimination of a nonsingular $n \times n$ TP matrix A and its transpose are known with HRA, Koev presents algorithms for computing with HRA its eigenvalues, singular values, and the solution of linear systems of equations $Ax = c$ where the entries of the vector c have alternating signs. In Reference 9, Koev implemented these algorithms with the Matlab or Octave functions `TNSolve`, `TNEigenvalues`, and `TNSingularvalues`. The computational cost of the function `TNSolve` is $\mathcal{O}(n^2)$ elementary operations and it requires as input arguments the bidiagonal factorization (3) of the matrix A and the vector c of the linear system $Ax = c$. The computational cost of `TNEigenvalues` and `TNSingularvalues` is $\mathcal{O}(n^3)$. These functions also require as input argument the bidiagonal factorization (3) of the matrix A .²¹

We have implemented the Matlab or Octave function `TNBDA`, which takes as input arguments the bidiagonal factorization (3) of the collocation matrix at t_1, \dots, t_{n+1} of a system, positive values d_0, \dots, d_n , and $\varphi(t_1), \dots, \varphi(t_{n+1})$ for a given positive function φ . Using Equation (11), `TNBDA` computes the bidiagonal factorization (3) of the collocation matrix at t_1, \dots, t_{n+1} of the corresponding weighted φ -transformed system. We have used this bidiagonal decomposition with `TNSolve`, `TNEigenValues`, and `TNSingularValues` in order to obtain solutions for the above-mentioned algebraic problems.

Now we include some numerical experiments considering collocation matrices of weighted φ -transformed systems. Due to the ill conditioning of these matrices, traditional methods do not achieve accurate solutions when solving the mentioned algebraic problems. The numerical results show this fact and the high accuracy of the algorithms that we have presented, even when the bidiagonal factorization of A is not computed with HRA.

6.1 | Linear systems

Linear systems arise when solving interpolation problems. Hence, in this section, we shall illustrate the accuracy of the computed solutions of $Ax = c$ when using the function `TNSolve` with the bidiagonal factorization of A given by `TNBDA`. We have obtained the solution of the systems using Mathematica with a precision of 100 digits and considered this solution exact. Then we have computed with Matlab two approximations, the first one using `TNBDA` and `TNSolve` and the second one using the Matlab command `\`.

First, we have considered collocation matrices of $(n + 1)$ -dimensional negative binomial bases, geometric bases, and Poisson bases at equidistant parameters in $(0, 1)$. Table 1 illustrates the condition number of these matrices.

Relative errors solving $A_n x = c_n$ with $c_n = ((-1)^{i+1} c_i)_{1 \leq i \leq n+1}$, where c_i is a random integer value are shown in Table 2. The computed results confirm the accuracy of the proposed method that, clearly, keeps the accuracy when the dimension of the problem increases. By contrast, when n increases the condition number of the considered matrices considerably increases and that explains the bad results obtained with the Matlab command `\`.

$n + 1$	$\kappa(A_n)$	$\kappa(A_n)$	$\kappa(A_n)$
10	1.0×10^4	8.5×10^7	6.0×10^6
20	1.8×10^8	6.7×10^{16}	3.6×10^{15}
25	2.5×10^{10}	2.0×10^{21}	8.0×10^{20}
50	1.4×10^{21}	5.4×10^{43}	1.4×10^{53}

TABLE 1 Condition number of collocation matrices of negative binomial bases (left), geometric bases (middle), and Poisson bases (right)

TABLE 2 Relative errors when solving $\mathbf{A}_n x = \mathbf{c}_n$ with collocation matrices of negative binomial bases (left), geometric bases (middle), and Poisson bases (right)

$n + 1$	$\mathbf{A}_n \setminus \mathbf{c}_n$	TNBDA	$\mathbf{A}_n \setminus \mathbf{c}_n$	TNBDA	$\mathbf{A}_n \setminus \mathbf{c}_n$	TNBDA
10	3.44798×10^{-13}	4.87933×10^{-16}	1.39105×10^{-11}	1.75715×10^{-16}	1.47267×10^{-10}	1.13940×10^{-15}
20	3.65455×10^{-10}	7.82315×10^{-16}	0.00173326	5.94029×10^{-16}	0.000338972	5.15797×10^{-16}
25	4.08151×10^{-8}	8.70322×10^{-16}	0.951529	8.85806×10^{-16}	0.999085	3.73189×10^{-16}
50	1.00833	5.31708×10^{-16}	1.000000	7.22416×10^{-16}	1.000000	2.51721×10^{-15}

TABLE 3 Condition number of collocation matrices of weighted φ -transformed systems with $f(t) = t, g(t) = 1 - t$ (left) and with $f(t) = t^2, g(t) = 1 - t^2$ (right)

$n + 1$	$\kappa(\mathbf{A}_n)$	$\kappa(\mathbf{A}_n)$
10	1.7×10^5	1.4×10^3
20	1.2×10^{11}	4.7×10^6
25	1.1×10^{14}	2.8×10^8
50	6.2×10^{28}	1.9×10^{17}

TABLE 4 Condition number of collocation matrices of weighted φ -transformed systems with $f(t) = \sin^2(t/2), g(t) = \cos^2(t/2)$ (left), with $f(t) = \sin((1+t)/2), g(t) = \sin((1-t)/2)$ (middle), and with $f(t) = \sinh((1+t)/2), g(t) = \sinh((1-t)/2)$ (right)

$n + 1$	$\kappa(\mathbf{A}_n)$	$\kappa(\mathbf{A}_n)$	$\kappa(\mathbf{A}_n)$
10	1.1×10^4	3.1×10^5	9.9×10^4
20	1.3×10^9	4.2×10^{11}	4.5×10^{10}
25	5.0×10^{11}	4.9×10^{14}	3.1×10^{13}
50	8.6×10^{24}	1.2×10^{30}	5.9×10^{27}

TABLE 5 Relative errors solving $\mathbf{A}_n x = \mathbf{c}_n$ with $f(t) = t, g(t) = 1 - t$ (left) and with $f(t) = t^2, g(t) = 1 - t^2$ (right)

$n + 1$	$\mathbf{A}_n \setminus \mathbf{c}_n$	TNBDA	$\mathbf{A}_n \setminus \mathbf{c}_n$	TNBDA
10	3.3579×10^{-13}	1.1191×10^{-15}	2.4173×10^{-14}	9.1615×10^{-16}
20	1.2413×10^{-9}	6.2974×10^{-16}	5.6324×10^{-11}	2.4457×10^{-15}
25	4.1424×10^{-7}	2.0843×10^{-15}	3.0923×10^{-9}	2.0016×10^{-15}
50	1.0000	7.5480×10^{-15}	0.9998	6.4231×10^{-15}

For different values of n , we have also considered collocation matrices at equidistant parameters in the interior of the interval domain of rational weighted $1/\omega^n$ -transformed systems (22), obtained by considering factors $L_i(t) = a_i g(t) + b_i f(t)$ with $a_i = 2$ and $b_i = 5, i \in \mathbb{N}$. Tables 3 and 4 illustrate the condition number of all considered matrices.

We have considered $\mathbf{c}_n = ((-1)^{i+1} c_i)_{1 \leq i \leq n+1}$, where c_i is a nonnegative random real number. Table 5 shows the relative errors when $f(t) = t, g(t) = 1 - t$ and the relative errors when $f(t) = t^2, g(t) = 1 - t^2, t \in [0, 1]$. Let us observe that, if $f(t) = t$ and $g(t) = 1 - t$, then

$$f(t_i)g(t_j) - f(t_j)g(t_i) = t_i - t_j.$$

On the other hand, if $f(t) = t^2$ and $g(t) = 1 - t^2$, then

$$f(t_i)g(t_j) - f(t_j)g(t_i) = (t_i - t_j)(t_i + t_j).$$

In both cases, the parameters (28) of the bidiagonal factorization (27) can be obtained with HRA and then $\mathbf{A}_n x = \mathbf{c}_n$ can also be solved with HRA. The numerical experiments confirm this fact.

Finally, Table 6 shows the relative errors in the solution of $\mathbf{A}_n x = \mathbf{c}_n$ with other functions f and g . In these cases, the computation with HRA of the parameters (28) of the bidiagonal factorization (27) should require the evaluation with HRA of the involved trigonometric or hyperbolic functions. Although this cannot be guaranteed, the numerical experiments

TABLE 6 Relative errors solving $A_n x = c_n$ with $f(t) = \sin^2(t/2)$, $g(t) = \cos^2(t/2)$ (left), with $f(t) = \sin((1+t)/2)$, $g(t) = \sin((1-t)/2)$ (middle), and with $f(t) = \sin h((1+t)/2)$, $g(t) = \sin h((1-t)/2)$ (right)

$n + 1$	$A_n \setminus c_n$	TNBDA	$A_n \setminus c_n$	TNBDA	$A_n \setminus c_n$	TNBDA
10	1.54259×10^{-14}	3.21312×10^{-16}	2.6135×10^{-13}	1.8511×10^{-15}	5.1019×10^{-14}	1.9355×10^{-15}
20	2.33365×10^{-12}	5.28031×10^{-16}	3.3808×10^{-9}	1.4878×10^{-15}	2.8313×10^{-9}	2.1849×10^{-15}
25	3.55721×10^{-11}	3.10246×10^{-15}	1.8684×10^{-6}	2.3294×10^{-15}	2.0441×10^{-7}	2.9733×10^{-15}
50	0.00306142	3.25552×10^{-15}	1.0000	2.1001×10^{-14}	1.0000	8.1439×10^{-15}

$n + 1$	Eig	TNBDA	Svd	TNBDA
10	3.09244×10^{-13}	0.	1.13631×10^{-13}	7.82315×10^{-16}
20	1.98025×10^{-9}	7.45447×10^{-16}	6.52333×10^{-10}	9.63835×10^{-16}
25	4.32252×10^{-7}	1.36414×10^{-15}	2.53703×10^{-8}	2.01433×10^{-16}
50	6784.57	1.17511×10^{-15}	416.354	6.06316×10^{-16}

TABLE 7 Relative errors when computing the lowest eigenvalue (left) and the lowest singular value (right) of collocation matrices of binomial negatives basis functions

$n + 1$	Eig	TNBDA	Svd	TNBDA
10	3.31746×10^{-11}	3.50531×10^{-16}	9.946626×10^{-11}	2.99503×10^{-16}
20	0.00583381	6.47223×10^{-16}	0.00181714	1.70411×10^{-16}
25	86.706	4.24794×10^{-16}	89.422	7.48982×10^{-16}
50	3.97376×10^{23}	1.37939×10^{-15}	1.37387×10^{23}	6.33473×10^{-16}

TABLE 8 Relative errors when computing the lowest eigenvalue (left) and the lowest singular value (right) of collocation matrices of geometric bases

$n + 1$	Eig	TNBDA	Svd	TNBDA
10	1.67788×10^{-10}	2.01781×10^{-15}	1.24564×10^{-10}	5.0222×10^{-16}
20	0.000067064	4.78068×10^{-16}	0.00187398	4.00738×10^{-16}
25	34.131	6.17955×10^{-16}	14.667	8.09263×10^{-16}
50	2.1166×10^{18}	2.80909×10^{-15}	4.022298×10^{16}	8.40959×10^{-15}

TABLE 9 Relative errors when computing the lowest eigenvalue (left) and the lowest singular value (right) of collocation matrices of Poisson bases

show again that accurate algebraic computations with the collocation matrices associated with these nonpolynomial basis functions can be performed.

6.2 | Eigenvalues and singular values

We have also used the bidiagonal decomposition provided by TNBDA for computing, with the Matlab functions `TNEigenValues` and `TNSingularValues`, the eigenvalues and the singular values, respectively, of the collocation matrices of weighted φ -transformed systems considered in the previous subsection. We have also computed their approximations with the Matlab functions `eig` and `svd`, respectively. In order to determine the accuracy of the approximations, we have calculated the eigenvalues and singular values of the matrices by using Mathematica with a precision of 100 digits and computed the relative errors corresponding to the approximations, considering the eigenvalues and singular values provided by Mathematica as exact.

The approximations of the eigenvalues and singular values obtained by means of TNBDA are very accurate for all considered n , whereas the approximations of the eigenvalues and singular values obtained with the Matlab commands `eig` and `svd` are not very accurate when n increases. Since these collocation matrices are all STP, let us recall that, by theorem 6.2 of Reference 7, all their eigenvalues are positive and distinct. Tables 7 to 14 show the relative errors of the approximations to the lowest eigenvalue and the lowest singular value obtained with both methods.

TABLE 10 Relative errors when computing the lowest eigenvalue (left) and the lowest singular value (right) of collocation matrices of weighted φ -transformed bases (22) with $f(t) = t, g(t) = 1 - t$

$n + 1$	Eig	TNBDA	Svd	TNBDA
10	1.80908×10^{-13}	2.18488×10^{-16}	3.59637×10^{-12}	8.09628×10^{-16}
20	3.9324×10^{-8}	1.51895×10^{-16}	7.02917×10^{-7}	1.37421×10^{-15}
25	0.0000625168	1.06001×10^{-15}	0.000970628	1.93093×10^{-15}
50	2.042×10^6	6.71094×10^{-15}	6.97145×10^{10}	7.04631×10^{-15}

TABLE 11 Relative errors when computing the lowest eigenvalue (left) and the lowest singular value (right) of collocation matrices of weighted φ -transformed bases (22) with $f(t) = t^2, g(t) = 1 - t^2$

$n + 1$	Eig	TNBDA	Svd	TNBDA
10	1.10761×10^{-14}	1.69399×10^{-15}	2.53164×10^{-15}	0.
20	3.21721×10^{-11}	1.5083×10^{-15}	2.47006×10^{-11}	2.52138×10^{-15}
25	6.63705×10^{-10}	1.11226×10^{-15}	5.01779×10^{-10}	0.
50	0.156406	4.19403×10^{-15}	0.328617	8.40959×10^{-15}

TABLE 12 Relative errors when computing the lowest eigenvalue (left) and the lowest singular value (right) of collocation matrices of weighted φ -transformed bases (22) with $f(t) = \sin^2(t/2), g(t) = \cos^2(t/2)$

$n + 1$	Eig	TNBDA	Svd	TNBDA
10	9.86257×10^{-14}	4.26029×10^{-16}	2.50339×10^{-13}	1.65788×10^{-15}
20	3.46514×10^{-9}	1.92921×10^{-15}	9.69498×10^{-9}	1.72095×10^{-15}
25	2.34348×10^{-6}	1.25379×10^{-15}	1.25441×10^{-6}	3.10795×10^{-15}
50	30.0203	6.69068×10^{-15}	7.3221×10^6	3.34347×10^{-15}

TABLE 13 Relative errors when computing the lowest eigenvalue (left) and the lowest singular value (right) of weighted φ -transformed bases (22) with $f(t) = \sin((1+t)/2), g(t) = \sin((1-t)/2)$.

$n + 1$	Eig	TNBDA	Svd	TNBDA
10	1.44963×10^{-12}	5.72976×10^{-16}	3.64477×10^{-12}	$4,32152 \times 10^{-16}$
20	3.60789×10^{-7}	4.1266×10^{-15}	7.45664×10^{-7}	1.85645×10^{-15}
25	0.000101621	1.57645×10^{-15}	0.00653125	1.9671×10^{-15}
50	1.29153×10^7	3.9368×10^{-15}	6.15482×10^{11}	3.25117×10^{-16}

TABLE 14 Relative errors when computing the lowest eigenvalue (left) and the lowest singular value (right) of collocation matrices of weighted φ -transformed bases (22) with $f(t) = \sin h((1+t)/2), g(t) = \sin h((1-t)/2)$

$n + 1$	Eig	TNBDA	Svd	TNBDA
10	4.60233×10^{-13}	2.70486×10^{-16}	8.45483×10^{-13}	6.82722×10^{-16}
20	9.05243×10^{-8}	2.80275×10^{-15}	1.04254×10^{-7}	4.425×10^{-16}
25	0.0000577108	1.05533×10^{-15}	0.000107575	2.95282×10^{-16}
50	71,585.3	7.90656×10^{-15}	9.1323×10^9	7.45104×10^{-16}

ACKNOWLEDGEMENT

This work was partially supported through the Spanish research grant PGC2018-096321-B-I00 (MCIU/AEI), by Gobierno de Aragón (E41_17R), and by Feder 2014-2020 "Construyendo Europa desde Aragón."

ORCID

Esmeralda Mainar  <https://orcid.org/0000-0002-1101-6230>

Juan Manuel Peña  <https://orcid.org/0000-0002-1340-0666>

Beatriz Rubio  <https://orcid.org/0000-0001-9130-0794>

References

- Šir Z, Jüttler B. On de Casteljau-type algorithms for rational Bézier curves. *Comput Appl Math*. 2015;288:244–250.
- Han LW, Wua YS, Chua Y. Weighted Lupaş q-Bézier curves. *Comput Appl Math*. 2016;308:318–329.

3. Demmel J, Dumitriu I, Holtz O, Koev P. Accurate and efficient expression evaluation and linear algebra. *Acta Numer.* 2008;17:87–145.
4. Marco A, Martínez JJ. Bidiagonal decomposition of rectangular totally positive said-ball-vandermonde matrices: Error analysis, perturbation theory and applications. *Linear Algebra Appl.* 2016;495:90–107.
5. Delgado J, Peña JM. Fast and accurate algorithms for Jacobi-Stirling matrices. *Appl Math Comput.* 2014;236:253–259.
6. Koev P. Accurate computations with totally nonnegative matrices. *SIAM J Matrix Anal Appl.* 2007;29:731–751.
7. Ando T. Totally positive matrices. *Linear Algebra Appl.* 1987;90:165–219.
8. Delgado J, Peña JM. Accurate computations with Lupaş matrices. *Appl Math Comput.* 2017;303:171–177.
9. Koev P. accessed 26 Jan 2019. <http://www.math.sjsu.edu/~koev/software/TNTool.html>.
10. Marco A, Martínez JJ. Accurate computations with totally positive Bernstein-Vandermonde matrices. *Electron J Linear Algebra.* 2013;26:357–380.
11. Delgado J, Peña JM. Accurate computations with collocation matrices of q-Bernstein polynomials. *SIAM J Matrix Anal Appl.* 2015;36:880–893.
12. Mainar E, Peña JM. Accurate computations with collocation matrices of a general class of bases. *Linear Algebra Appl.* 2018;25:e2184.
13. Delgado J, Peña G, Peña JM. Accurate and fast computations with positive extended Schoenmakers-Coffey matrices. *Numerical Linear Algebra Appl.* 2016;23:1023–1031.
14. Carnicer JM, Peña JM. Shape preserving representations and optimality of the Bernstein basis. *Adv Comput Math.* 1993;1:173–196.
15. Peña JM. Shape preserving representations for trigonometric polynomial curves. *Comput Aided Geom Design.* 1997;14:5–11.
16. Gasca M, Peña JM. Total positivity and Neville elimination. *Linear Algebra Appl.* 1992;165:25–44.
17. Gasca M, Peña JM. On factorizations of totally positive matrices. In: Gasca M, Micchelli CA, editors. *Total Positivity and Its Applications.* Dordrecht, Netherlands: Kluwer Academic Publishers, 1996; p. 109–130.
18. Delgado J, Peña JM. Accurate computations with collocation matrices of rational bases. *Appl Math Comput.* 2013;219:4354–4364.
19. Marco A, Martínez JJ. Polynomial least squares fitting in the Bernstein basis. *Linear Algebra Appl.* 2010;433:1254–1264.
20. Goldman R. Urn models, approximations, and splines. *J Approx Theory.* 1988;54:1–66.
21. Koev P. Accurate eigenvalues and SVDs of totally nonnegative matrices. *SIAM J Matrix Anal Appl.* 2005;27(1):1–23.
22. Morin G, Goldman R. A subdivision scheme for Poisson curves and surfaces. *Comput Aided Geom Design.* 2000;17:813–833.
23. Sánchez-Reyes J. Harmonic rational Bézier curves, p-Bézier curves and trigonometric polynomials. *Comput Aided Geom Design.* 1998;15:909–923.
24. Bashir U, Ali J. Monotone rational trigonometric interpolation. *Int J Math Comput Sci.* 2014;8(2):295–297.

How to cite this article: Mainar E, Peña JM, Rubio B. Accurate bidiagonal decomposition of collocation matrices of weighted φ -transformed systems. *Numer Linear Algebra Appl.* 2020;27:e2295.





<https://doi.org/10.1002/nla.2295>

ARTICLE 3

- [39] R. Gonzalez, E.Mainar, E.Paluzo, B.Rubio, Neural-Network-Based Curve Fitting Using Totally Positive Rational Bases, *Mathematics* **8**, 2197 (2020).
<https://doi.org/10.3390/math8122197>

Article

Neural-Network-Based Curve Fitting Using Totally Positive Rational Bases

Rocio Gonzalez-Diaz ^{1,†,‡} , E. Mainar ^{2,†,§} , Eduardo Paluzo-Hidalgo ^{1,†,‡}  and B. Rubio ^{2,*,†,§} 

¹ Department of Applied Mathematics I, University of Sevilla, 41012 Sevilla, Spain; rogod@us.es (R.G.-D.); epaluzo@us.es (E.P.-H.)

² Department of Applied Mathematics, University Research Institute of Mathematics and Its Applications (IUMA), University of Zaragoza, 50001 Zaragoza, Spain; esmemain@unizar.es

* Correspondence: brubio@unizar.es

† The authors contributed equally to this work.

‡ The authors are partially supported by MICINN, FEDER/UE under grant PID2019-107339GB-I00.

§ The authors are partially supported through the Spanish research grant PGC2018-096321-B-I00 (MCIU/AEI), by Gobierno de Aragón (E41_17R) and by Feder 2014-2020 “Construyendo Europa desde Aragón”.

Received: 2 November 2020; Accepted: 8 December 2020; Published: 10 December 2020



Abstract: This paper proposes a method for learning the process of curve fitting through a general class of totally positive rational bases. The approximation is achieved by finding suitable weights and control points to fit the given set of data points using a neural network and a training algorithm, called AdaMax algorithm, which is a first-order gradient-based stochastic optimization. The neural network presented in this paper is novel and based on a recent generalization of rational curves which inherit geometric properties and algorithms of the traditional rational Bézier curves. The neural network has been applied to different kinds of datasets and it has been compared with the traditional least-squares method to test its performance. The obtained results show that our method can generate a satisfactory approximation.

Keywords: normalized totally positive bases; normalized B-bases; rational bases; curve fitting; neural network

1. Introduction

The problem of obtaining a curve that fits a given set of data points is one of the fundamental challenges of Computer Aided Geometric Design (CAGD), and it has become prevalent in several applied and industrial domains, such as Computer-Aided Design and Manufacturing (CAD/CAM) systems, Computer Graphics and Animation, Robotics Design, Medicine and many others. To face this issue, several families of bases of functions have been considered. There is a large body of literature on this topic and there are numerous methods to solve this issue, such as several least-squares techniques and different progressive iterative approximation methods (see [1–3] and the references therein).

Given a system (u_0, \dots, u_n) of linearly independent functions defined on an interval $I \subseteq \mathbb{R}$ and $P_0, \dots, P_n \in \mathbb{R}^k$, we can define a parametric curve as $\gamma(t) = \sum_{i=0}^n P_i u_i(t)$, $t \in I$. The polygon $P_0 \cdots P_n$, formed by the ordered sequence of points $P_i \in \mathbb{R}^k$, $i = 0, \dots, n$, is called the *control polygon* of γ and the points P_i , $i = 0, \dots, n$, are named the *control points* of γ with respect to (u_0, \dots, u_n) . A matrix is *totally positive* (TP) if all its minors are nonnegative (see [4]). A system of functions (u_0, \dots, u_n) defined on I is TP if all its collocation matrices $(u_j(t_i))_{i,j=0,\dots,n}$ with $t_0 < \dots < t_n$ in I are TP. A TP system of functions on I is *normalized* (NTP) if $\sum_{i=0}^n u_i(t) = 1$, for all $t \in I$. A basis provides shape-preserving representations if the shape of the curve imitates the shape of its control polygon. Normalized totally positive bases provide shape-preserving representations. The normalized B-basis of a given space is an

NTP basis such that the matrix of change of basis of any NTP basis with respect to the normalized B-basis is TP and stochastic. This property implies that the control polygon of a curve with respect to the normalized B-basis can be obtained by a corner cutting algorithm from its control polygon with respect to any other NTP basis. Thus, the control polygon with respect to the normalized B-basis is closer in shape to the curve than the control polygon with respect to any other NTP basis. Furthermore, the length of the control polygon with respect to the normalized B-basis lies between the length of the curve and the length of its control polygon with respect to any other NTP basis. Similar properties hold for other geometric properties such as angular variation or number of inflections (see [5–7]). So, the Normalized B-basis has the optimal shape-preserving properties among all NTP bases of the space. The Bernstein bases and the B-spline bases are the normalized B-bases of their corresponding generated spaces.

It is well known that the bases obtained by rationalizing Bernstein bases are also the normalized B-bases of the generated spaces of rational functions. These spaces are made up of rational polynomial functions where the denominator is a given polynomial. Rational Bernstein bases add adjustable weights to provide closer approximations to arbitrary shapes and have become a standard tool in CAGD since they allow the exact representation of conic sections, spheres and cylinders. In [8], the generalization of rational Bernstein bases obtained when replacing the linear polynomial factors by trigonometric or hyperbolic functions or their mixtures with polynomials were analyzed. The generated rational curves inherit geometric properties and algorithms of the traditional rational Bézier curves and so, they can be considered as modeling tools in CAD/CAM systems.

As mentioned before, the weights of rational bases can be used as shape parameters. However, it is well known that the effect of changing a weight in a rational basis is different from that of moving a control point of the curve (see Figure 1). Thus, the interactive shape control of rational curves through adjusting weights is not a straightforward task and it is not easy to design algorithms to obtain the appropriate weights (see [9], Chapter 13).

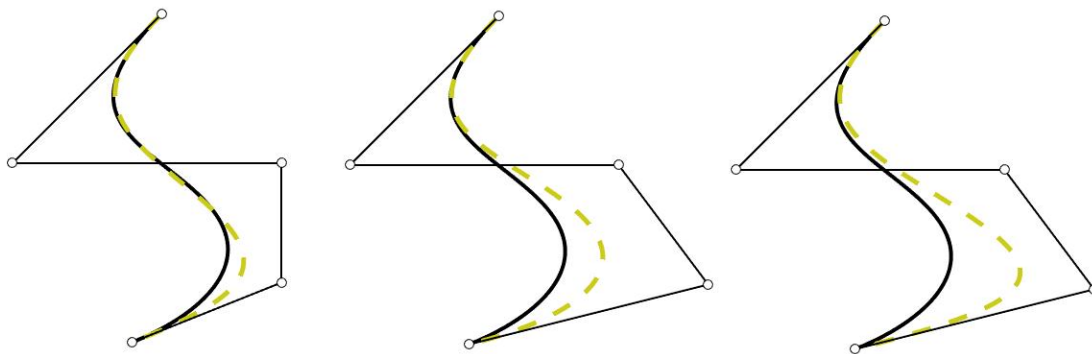


Figure 1. Initial curve (line) and curve obtained (dotted line) after changing the weights and/or control points. Left: changing the fourth weight; center: changing the fourth control point; and right: changing the fourth weight and the fourth control point.

Some recent papers have shown that the application of Artificial Intelligence (AI) techniques can achieve remarkable results regarding the problem of obtaining rational curves that fit a given set of data points. To face this issue, in [10], a bio-inspired algorithm was applied through the use of rational Bézier curves. Besides, in [11,12], evolutionary algorithms were applied to rational B-spline curves. As a novelty, in this paper, we define a one-hidden-layer neural network using the general class of rational bases with optimal shape-preserving properties proposed in [8]. In that work, the authors presented evaluation and subdivision algorithms. However, this is the first time that the problem of obtaining a rational fitting curve using these general class of totally positive rational bases is modeled by a neural network and its weights and control points optimized using a training algorithm. In this paper, we extend [8] for their application in curve fitting training the neural network with a recent stochastic learning process, the AdaMax algorithm [13], to find suitable weights and control

points. In this approximation process, the rational basis is a hyperparameter and can be changed by substituting the linear factors by polynomial, trigonometric or hyperbolic functions, thus expanding the potential range of applications to include more difficult shapes.

The layout of the paper is as follows. In Section 2, we recall several concepts regarding CAGD and we present a general class of rational bases which are normalized B-bases. Then, in Section 3, we present a one-hidden-layer neural network based on the rational bases presented in the previous section to approximate a given set of data points. This neural network is trained with an optimization algorithm to update the weights and control points used to construct a curve that approximates the given set of data points, while decreasing a loss function. In Section 4, several experiments are provided illustrating the use of the neural network with different normalized B-bases to test its performance giving an approximation of different kinds of sets of data points. Moreover, the proposed method has been compared with the traditional least-squares method. Finally, conclusions and future work are presented in Section 5.

2. Shape-Preserving and Rational Bases

Let us suppose that $I \subseteq \mathbb{R}$ and $f, g : I \rightarrow \mathbb{R}$ are nonnegative continuous functions. Then, for $n \in \mathbb{N}$, we can define the system (u_0^n, \dots, u_n^n) where:

$$u_k^n(t) = \binom{n}{k} f^k(t) g^{n-k}(t) \text{ such that } t \in I, \quad k = 0, \dots, n. \tag{1}$$

For any positive weights $w_i^n, i = 0, \dots, n$, let us define $\omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t)$ and denote by $(\rho_0^n, \dots, \rho_n^n)$ the rational basis described by

$$\rho_i^n(t) = w_i^n \frac{1}{\omega^n(t)} u_i^n(t), \quad i = 0, \dots, n, \tag{2}$$

where (u_0^n, \dots, u_n^n) is defined in (1). Clearly, this system spans a space of rational functions with denominator $\omega^n(t)$,

$$\mathcal{R}^n = \text{span}\{\rho_i^n(t) \mid i = 0, \dots, n\} = \{u(t)/\omega^n(t) \mid u(t) \in \mathcal{U}^n\}, \tag{3}$$

where \mathcal{U}^n is the space generated by (u_0^n, \dots, u_n^n) .

The following result corresponds to Corollary 4 of [8] and provides the conditions characterizing that the system given in (2) has optimal shape-preserving properties.

Proposition 1. *The system of functions given in (2) is the normalized B-basis of the space \mathcal{R}^n defined in (3) if and only if the function f/g defined on $I_0 = \{t \in I \mid g(t) \neq 0\}$ is increasing and satisfies*

$$\inf \left\{ \frac{f(t)}{g(t)} \mid t \in I_0 \right\} = 0, \quad \sup \left\{ \frac{f(t)}{g(t)} \mid t \in I_0 \right\} = +\infty. \tag{4}$$

Let us see several choices of functions f and g satisfying the conditions of Proposition 1. Let us consider the functions

$$f(t) = \frac{t-a}{b-a}, \quad g(t) = \frac{b-t}{b-a}, \quad t \in [a, b]. \tag{5}$$

It is well known that the corresponding rational basis (2), which is the rational Bernstein basis, is the normalized B-basis of its generated space (3).

We can also consider the functions

$$f(t) = t^2, \quad g(t) = 1 - t^2, \quad t \in [0, 1]. \tag{6}$$

The corresponding rational basis (2) spans the space

$$\mathcal{R}^n = \text{span}\{u(t)/\omega^n(t) \mid u(t) \in \mathcal{U}^n, \omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t)\},$$

where the system (u_0, \dots, u_n) given in (1) spans the space $\langle 1, t^2, \dots, t^{2n} \rangle$ of even polynomials defined on $[0, 1]$ of degree less than or equal to $2n$.

Trigonometric and hyperbolic bases are attracting a lot of interest, for instance in Isogeometric Analysis (cf. [14]). Let $0 < \Delta < \pi/2$. Define

$$f(t) = \sin((\Delta + t)/2) \text{ and } g(t) = \sin((\Delta - t)/2) \text{ for } t \in I = [-\Delta, \Delta]. \tag{7}$$

Let us notice that the functions f and g satisfy $f(t) > 0$ and $g(t) > 0$ for all $t \in (-\Delta, \Delta)$. Moreover, it can be checked that

$$\left(\frac{f(t)}{g(t)}\right)' = \left(\frac{\sin\left(\frac{\Delta+t}{2}\right)}{\sin\left(\frac{\Delta-t}{2}\right)}\right)' = \frac{1}{2} \frac{\sin(\Delta)}{\sin^2\left(\frac{\Delta-t}{2}\right)} > 0, \quad \forall t \in (-\Delta, \Delta). \tag{8}$$

Therefore, for any $0 < \Delta < \pi/2$, the function f/g is a strictly increasing function on $(-\Delta, \Delta)$ and f and g satisfy the conditions of Proposition 1. The corresponding rational basis (2) spans the space

$$\mathcal{R}^n = \text{span}\{u(t)/\omega^n(t) \mid u(t) \in \mathcal{U}^n, \omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t)\},$$

where, for a given $n = 2m$, the system (u_0, \dots, u_n) given in (1) is a basis that coincides, up to a positive scaling, with the normalized B-basis of the space $\langle 1, \cos t, \sin t, \dots, \cos mt, \sin mt \rangle$ of trigonometric polynomials of degree less than or equal to m on I (see Section 3 of [15]).

Finally, let $\Delta > 0$. Define

$$f(t) = \sinh((\Delta + t)/2) \text{ and } g(t) = \sinh((\Delta - t)/2) \text{ for } t \in I = [-\Delta, \Delta]. \tag{9}$$

Clearly, $f(t) > 0$ and $g(t) > 0$ for all $t \in (-\Delta, \Delta)$. Moreover, it can be checked that

$$\left(\frac{f(t)}{g(t)}\right)' = \left(\frac{\sinh\left(\frac{\Delta+t}{2}\right)}{\sinh\left(\frac{\Delta-t}{2}\right)}\right)' = \frac{1}{2} \frac{\sinh \Delta}{\sinh^2\left(\frac{\Delta-t}{2}\right)} > 0, \quad \forall t \in (-\Delta, \Delta). \tag{10}$$

Therefore, for any $\Delta > 0$, f/g is a strictly increasing function on $(-\Delta, \Delta)$ and f and g satisfy the conditions of Proposition 1. The corresponding rational basis (2) spans the space

$$\mathcal{R}^n = \text{span}\{u(t)/\omega^n(t) \mid u(t) \in \mathcal{U}^n, \omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t)\},$$

where, for $n = 2m$, the system (u_0, \dots, u_n) given in (1) spans the space $\langle 1, e^t, e^{-t}, \dots, e^{mt}, e^{-mt} \rangle$ of hyperbolic polynomials of degree less than or equal to m on I .

In Figure 2, we illustrate two examples of the rational basis (2) of degree 3. Let us observe the effect on the shape of the functions of the basis as weights change.

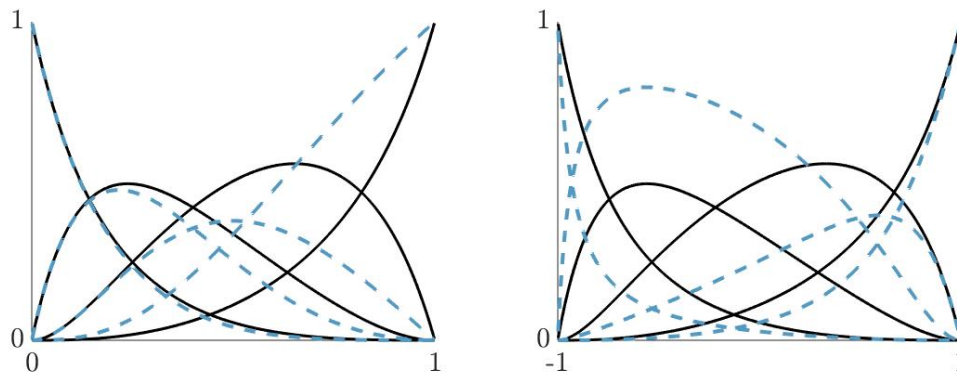


Figure 2. (Left): Rational basis (2) using $f(t) = t, g(t) = 1 - t, t \in [0, 1]$. Weights $w = [1, 2, 3, 2]$ (black line) and weights $w = [1, 2, 3, 8]$ (blue dotted line). (Right): Rational basis (2) using $f(t) = \sin((\Delta + t)/2), g(t) = \sin((\Delta - t)/2), t \in I = [-\Delta, \Delta], 0 < \Delta < \pi/2$. Weights $w = [1, 2, 3, 2]$ (black line) and weights $w = [1, 8, 3, 2]$ (blue dotted line).

Section 4 will show examples of approximations of given sets of data points using all the above mentioned normalized B-bases. Moreover, the neural network presented in the following section will be used to compute the optimal weights and control points of their corresponding fitting curves.

3. Curve Fitting with Neural Networks

It is well known that curve fitting is the process of constructing a curve, or mathematical function, that has the best fit of a given set of data points. A related topic is Regression Analysis in Machine Learning. In the literature (see [16] (Chapter 11)), a regression problem is the problem of predicting a real value for each input data. Let us consider an input space $X \subset \mathbb{R}^m$ and target values $Y \subset \mathbb{R}^k$, and a distribution over $X \times Y$, denoted by \mathcal{D} . Then, a regression problem consists of a set of labeled samples $S = \{(x_i, y_i)\}_{i \in \{0, \dots, \ell\}} \in X \times Y$ drawn according to \mathcal{D} where y_i are the target real values we want to predict. There exists a huge variety of regression algorithms, i.e., algorithms solving regression problems, such as Linear Regression, Decision Trees, Support Vector Regression, and Neural Networks, among others. The quality of the prediction of an algorithm or model depends on the difference between the target (i.e., the true value) and the predicted one, and it is measured using a loss function. Then, given a set \mathcal{H} (also called “hypothesis”) of function mappings X to Y , the aim of the regression algorithm is to use S to find $h \in \mathcal{H}$ such that the expected loss is small.

Specifically, the problem that we want to solve can be stated as follows. Suppose that f and g are functions defined on $[a, b]$ satisfying the conditions of Proposition 1. Consider a set of parameters $a \leq t_0 < \dots < t_\ell \leq b$ and a sequence of data points $s_0, \dots, s_\ell \in \mathbb{R}^k$, where each parameter t_i is associated with a data point s_i . For some $n \leq \ell$, we want to obtain a rational curve

$$c(t) = \sum_{i=0}^n \frac{w_i^n \binom{n}{i} f^i(t) g^{n-i}(t)}{\sum_{i=0}^n w_i^n \binom{n}{i} f^i(t) g^{n-i}(t)} P_i, \quad t \in [a, b], \tag{11}$$

to approximate the set of data points $s = (s_i)_{i=0}^\ell$. Therefore, the goal is to obtain the weights w_0^n, \dots, w_n^n and the control points P_0, \dots, P_n of the rational curve (11) that best fits the set of data points. In order to compute them, we have used a stochastic optimization process to train a neural network that models the rational curve $c(t)$.

The problem to be solved can be interpreted then as a regression problem where the set of labeled samples is composed of the input data, X , that is the set of parameters $a \leq t_0 < \dots \leq t_\ell \leq b$ and the target set of data points $Y = s = (s_i)_{i=0}^\ell$.

Then, the expression in (11) can be represented as a hierarchical computational graph with just one hidden layer that we will denote as $\mathcal{N}_{w,P} : \mathbb{R} \rightarrow \mathbb{R}^k$ where the computations are organized as in

Figure 3. The obtained curve $\mathcal{N}_{w,P}(t)$ is the rational curve $c(t)$ that approximates the given set of data points and we denote as the fitting curve.

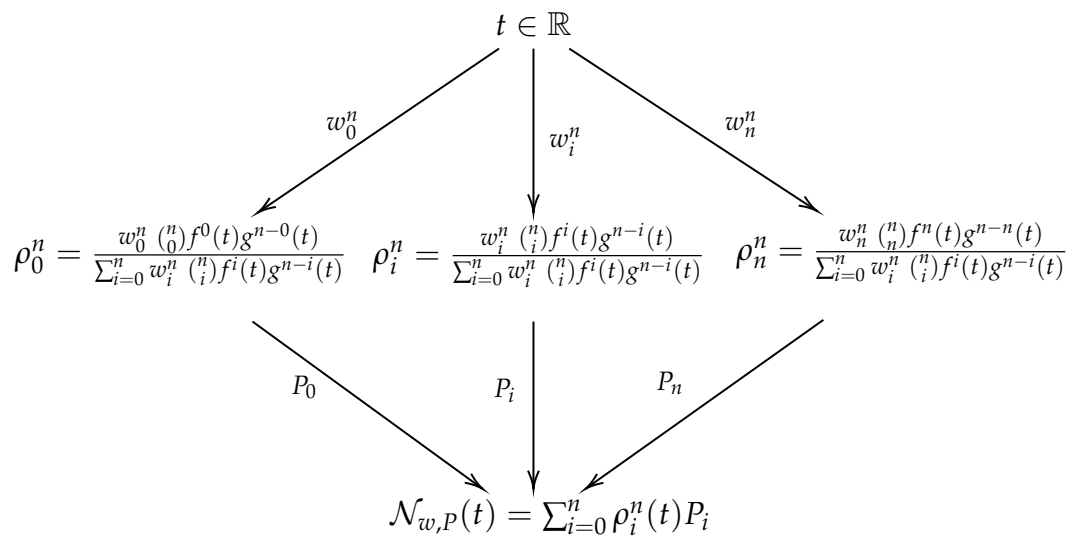


Figure 3. From top to bottom. The input layer has the parameter $t \in \mathbb{R}$ as input. The hidden layer is of width $n + 1$ and its parameters are the weights. Then, the output layer computes the approximation of the target curve and its parameters are the control points.

The key idea is to iteratively change the input weights $w = (w_i^n)_{i=0}^n$ and control points $P = (P_i)_{i=0}^\ell$ of the active curve $\mathcal{N}_{w,P}(t)$, and so it deforms towards the target shape represented by the set of data points $s = (s_i)_{i=0}^\ell$ (see Figure 4).

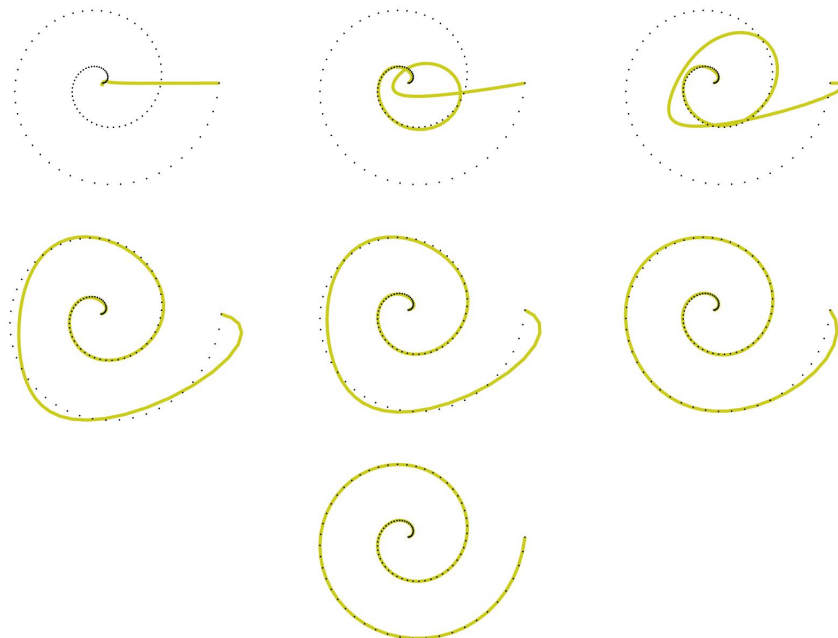


Figure 4. Evolution of the fitting curve $\mathcal{N}_{w,P}(t)$. Set of data points from the target curve (dotted) and the fitting curve (line). From top to bottom and left to right: Increment $d = 0, d = 250, d = 500, d = 1000, d = 1500, d = 2000$ and $d = 3000$.

Then, we apply an adaptive learning rate optimization algorithm to train the neural network to find the weights and control points, which can be, for example, the Adaptive Moment Estimation

(Adam) algorithm or its variant Adaptive Moment Estimation Maximum (AdaMax) algorithm based on infinity norm. These methods are used for stochastic optimization, to solve the supervised learning problem and to find the parameters where a minima is located. In this paper, we have mainly used the AdaMax variant (see Algorithm 1) because of its stability and simplicity [13]. However, the Adam method can be useful depending on the shape of the set of data points to be approximated and the choice of the loss function. The stochastic objective function, also called the loss function, measures the goodness of the fitting curve. Let us notice that there exist different loss functions such as the mean absolute error, the cross-entropy loss, the mean squared error, among others (the different loss functions implemented in Tensorflow can be consulted in [the tensorflow documentation](#)), that can be chosen depending on the problem. In our case, we have considered the mean absolute error as the loss function because of the choice of the training algorithm, given by the following expression:

$$E(w, P) = \frac{\sum_{i=0}^{\ell} |s_i - \mathcal{N}_{w,P}(t_i)|}{\ell + 1}. \quad (12)$$

The Adam and the AdaMax algorithms are stochastic gradient-based optimization algorithms and, as previously mentioned, they update the weights and the control points iteratively. The step size is a real number that measures how much the weights and the control points are updated upon each iteration. Besides, the Adam algorithm uses the first and the second moment estimate to update the weights and the control points which are updated following exponential decay rates (β_1 and β_2). Finally, as AdaMax is a variation of Adam using infinity norm, the second moment estimate has a simple recursive formula which will be denoted in Algorithm 1 as exponentially weighted infinity norm. See [13] for a detailed description of the both Adam and AdaMax algorithms.

Algorithm 1: The AdaMax algorithm [13] adapted to our context.

Result: A set of weights w and control points P .

Require: The number of iterations k or an upper bound $e \in \mathbb{R}$ for $E(w, P)$;

Require: The stepsize α ;

Require: The exponential decay rates $\beta_1, \beta_2 \in [0, 1)$;

Require: The stochastic objective function $E(w, P)$;

Require: A small constant ε for numerical stability;

Initialize: Time step $d := 0$;

Initialize: The set of weights and control points in time step $d = 0$, $w^{(0)}$ and $P^{(0)}$ randomly sampled;

Initialize: First moment vector $\gamma^{(0)} := 0$;

Initialize: Exponentially weighted infinity norm $\delta^{(0)} := 0$;

while $d < k$ or $E(w, P) > e$ **do**

$d := d + 1$ (Increment the time step);

$\gamma^{(d)} := \beta_1 \cdot \gamma^{(d-1)} + (1 - \beta_1) \cdot \nabla_{w^{(d-1)}, P^{(d-1)}} E(w^{(d-1)}, P^{(d-1)})$ (Update the biased first moment estimation);

$\delta^{(d)} := \max(\beta_2 \cdot \delta^{(d-1)}, \nabla_{w^{(d-1)}, P^{(d-1)}} E(w^{(d-1)}, P^{(d-1)}))$ (Update the exponentially weighted infinity norm);

$w^{(d)} := w^{(d-1)} - \frac{\alpha}{1 - \beta_1^d} \cdot \frac{\gamma^{(d-1)}}{\delta^{(d-1)} + \varepsilon}$ (Update the weights);

$P^{(d)} := P^{(d-1)} - \frac{\alpha}{1 - \beta_1^d} \cdot \frac{\gamma^{(d-1)}}{\delta^{(d-1)} + \varepsilon}$. (Update the control points);

end

The number of units (i.e. weights and control points) is a hyperparameter and is determined based on the complexity of the shape to be approximated. Besides, the step size, α , can be changed depending on the state of the convergence of the training procedure, for example, when the loss values

(i.e., the evaluation of the loss function) gets stuck or the update of the parameters is too big. Then, it is useful to increase or reduce, respectively, the step size according to the values of the loss function.

4. Experiments Results

In order to show the performance of the neural network $\mathcal{N}_{w,p}$, we have taken different sets of data points $s = (s_i)_{i=0}^{\ell}$. They have been chosen to reflect the variety of situations where the proposed neural network can be applied. The first set of data points belongs to a closed conic curve, the second one belongs to a transcendental curve, the third one is a curve with a twisted shape and, finally, the fourth one is a noisy set of data points from a baroque image.

In all cases, we have taken different functions f and g satisfying the conditions of Proposition 1 and allowing that the corresponding rational bases (2) have the optimal shape-preserving properties.

Remark 1. *One of the requirements for a rational basis to be the normalized B-basis of its generated space is the positivity of the weights. This makes it necessary to apply the absolute value to the weights in the weight update step of Algorithm 1. However, in the experiments shown, we opted to avoid this choice because we have observed that, although in the intermediate steps the weights could be negative, at the end of the training, the weights were positive and the convergence was faster. Nevertheless, we can add the restriction depending on the needs.*

Let us notice that we have used an uniform distribution of the parameters $t_i, i = 0, \dots, \ell$, in all the examples. This choice of parameters does not have to be the optimal but we have preferred it because of its simplicity. Besides, a normalization of the data points, $s = (s_i)_{i=0}^{\ell}$, between 0 and 1, was applied in order to facilitate the training procedure following the formula:

$$\hat{s}_i = \frac{s_i - \min S}{\max S - \min S}, \text{ for } i \in \{0, \dots, \ell\}. \quad (13)$$

The AdaMax algorithm was applied to solve the minimization problem with the mean absolute error loss function (12) with the following choice of hyperparameters: $\alpha = 0.0001, \beta_1 = 0.9, \beta_2 = 0.999$, and $\varepsilon = 10^{-7}$. Then, the number of iterations of the algorithm depends on the desired accuracy of the fitting curve to the set of data points. In this case, we have used between 3000 to 5000 iterations of the algorithm but, with more iterations and a better tuning of the parameters, the results provided here may be improved. Finally, in order to reach a better approximation, the first and the last control point of the fitting curve were fixed to be the same as the first and the last point of the set of data points, thus the obtained fitting curve is always exactly at those points. We can see in Table 1, a summary of the loss values from different fitting curves. Let us observe that the value n is the degree of the fitting curve and it depends on the complexity of the shape to be approximated. Moreover, let us notice that the proposed neural network is able to obtain a suitable accuracy with low degrees and, as a generalization of other methods, we can choose, depending on the shape of the set of data points, the basis that best fits. Note that, in CAGD, it is important to properly address the problem of curve fitting, finding a balance between accuracy and degree of the curve since high-degree curves are computationally expensive to evaluate. The AdaMax algorithm has been selected because it is computationally efficient with little memory requirements, suited for problems with large data or parameters. In Table 2, the time of execution of the Algorithm 1 using different numbers of units (i.e., weights and control points) and numbers of iterations is provided.

Table 1. Loss values of the mean absolute error (12) for different fitting curves of degree n with $f(t) = t, g(t) = 1 - t, t \in [0, 1]$ (Basis 1), $f(t) = t^2$ and $g(t) = 1 - t^2, t \in [0, 1]$, (Basis 2), $f(t) = \sin((\Delta + t)/2)$ and $g(t) = \sin((\Delta - t)/2), \Delta < \pi/2, t \in [-\Delta, \Delta]$, (Basis 3) and finally $f(t) = \sinh((\Delta + t)/2)$ and $g(t) = \sinh((\Delta - t)/2), \Delta < \pi/2, t \in [-\Delta, \Delta]$, (Basis 4). They were all trained with 4000 iterations, $\alpha = 0.0001, \beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 10^{-7}$. The process was repeated 5 times, with the loss values provided being the best values reached.

n	Basis 1	Basis 2	Basis 3	Basis 4
Circle				
3	3.3946×10^{-2}	3.7129×10^{-2}	7.0468×10^{-2}	3.6438×10^{-2}
4	2.1757×10^{-3}	1.5338×10^{-2}	3.1678×10^{-3}	2.5582×10^{-3}
5	1.7333×10^{-4}	9.2269×10^{-3}	2.8083×10^{-4}	2.2488×10^{-3}
Cycloid				
8	1.0849×10^{-3}	3.6855×10^{-4}	3.6017×10^{-4}	3.1674×10^{-4}
9	4.6163×10^{-4}	3.6855×10^{-4}	3.6017×10^{-4}	2.4914×10^{-4}
10	3.3944×10^{-4}	3.6855×10^{-4}	3.6017×10^{-4}	2.4914×10^{-4}
Archimedean spiral				
11	1.5982×10^{-3}	1.0474×10^{-2}	2.2349×10^{-2}	7.8109×10^{-4}
12	1.5982×10^{-3}	7.8916×10^{-3}	5.7801×10^{-3}	7.8109×10^{-4}
13	1.4106×10^{-3}	5.2853×10^{-3}	5.7801×10^{-3}	7.8109×10^{-4}

Table 2. Time of execution of the proposed algorithm measured in seconds for different numbers of units and iterations. The values provided are the mean of 5 repetitions with a set of data points of size 100.

$n + 1$	Number of Iterations				
	1	25	50	100	3000
5	0.1259	1.4284	2.8381	5.7259	189.5386
10	0.0989	2.0781	4.1325	10.2672	268.8726
15	0.1244	2.7142	5.3781	10.9886	347.6139
50	0.6479	8.2589	13.3576	27.4398	850.6713
100	1.1624	14.3999	32.6576	65.3298	1521.3971

The implementation (the code of the experimentation can be found in <https://github.com/Cimagroup/Curve-approximation-NN>) was developed using TensorFlow 2.0 [17] allowing developers to easily use it to build and deploy Machine Learning powered applications. All experiments were ran on a Quad-Core Intel Core i7 CPU, 2.8 GHz with 16 GB RAM. Let us see a detailed description of the experiments.

4.1. Circle Curve

Circles and conic sections play a relevant role in curve design and have been approximated in several works (see [18,19]). Let us show different approximations to the circle given by the parametric equations

$$\begin{cases} x_c(t) = \cos(t), \\ y_c(t) = \sin(t), \end{cases} \tag{14}$$

$t \in [0, 2\pi]$. First, let us see an approximation obtained performing the neural network $\mathcal{N}_{w,p}$ using polynomial functions.

We have taken the following sequence of points $s = (s_i)_{i=0}^{99}$ on the circle (14):

$$s_i = (x_c(2\pi t_i), y_c(2\pi t_i)), \quad i = 0, \dots, 99,$$

being the parameters $t_i = i/99, i = 0, \dots, 99$.

For $n = 5$, we have approximated the set s by the fitting curve $(\mathcal{N}_{w,P_x}(t), \mathcal{N}_{w,P_y}(t))$ considering the functions $f(t) = t$ and $g(t) = 1 - t$ at the vector nodes $t = (t_i)_{i=0}^{99}$ with t_i as above. After training the neural network $\mathcal{N}_{w,P}$, performing the Algorithm 1 for 3000 iterations, we have obtained the following weights:

$$w_0^5 = 1.253390550613403320, w_1^5 = 0.6930422186851501465, w_2^5 = 0.7461462020874023438, \\ w_3^5 = 0.7428795099258422852, w_4^5 = 0.7499830722808837891, w_5^5 = 1.511020541191101074,$$

and the following control points:

$$P_0 = (1, 0), P_1 = (0.9514569678202488, 2.265164366524486), \\ P_2 = (-2.041393043066488, 1.823594836649781), P_3 = (-2.573274681480492, -1.594942712567735), \\ P_4 = (0.9104298578262079, -2.558222347419337), P_5 = (1, 0).$$

We can see in Figure 5 the obtained fitting curve of degree 5 and its corresponding control polygon.

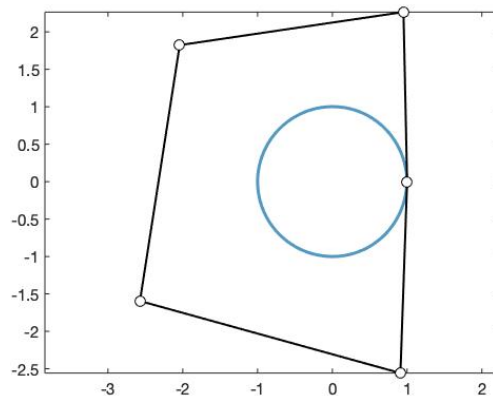


Figure 5. Fitting curve of degree 5 obtained using the functions $f(t) = t$ and $g(t) = 1 - t, t \in [0, 1]$, and its corresponding control polygon.

In order to analyze the performance of the obtained approximation to the circle, we depict some geometric approximations of the approximation error. In Figure 6, we plot the radial and curvature errors given, respectively, by

$$\epsilon_r(t) = r(t) - 1, \quad \epsilon_k = k(t) - 1,$$

where $r(t) = \sqrt{\mathcal{N}_{w,P_x}^2(t) + \mathcal{N}_{w,P_y}^2(t)}$ and $k(t) = (\mathcal{N}'_{w,P_x}(t)\mathcal{N}''_{w,P_y}(t) - \mathcal{N}''_{w,P_x}(t)\mathcal{N}'_{w,P_y}(t)) / \left(\sqrt{(\mathcal{N}'_{w,P_x}(t))^2 + (\mathcal{N}'_{w,P_y}(t))^2} \right)^3$.

We can observe that the radial error vanishes at $t = -\Delta, \Delta$ because the approximation is exact for the initial and the last points of the curve.

Let us see another approximation example of the unit circle performing the neural network $\mathcal{N}_{w,P}$ using trigonometric functions. We have the following set of data points $s = (s_i)_{i=0}^{99}$ on circle (14):

$$s_i = (x_c((t_i/\Delta + 1)\pi), y_c((t_i/\Delta + 1)\pi), \quad i = 0, \dots, 99,$$

being the parameters $t_i = -\Delta + 2\Delta i/99, i = 0, \dots, 99$, with $0 < \Delta < \pi/2$. For $n = 5$, we have approximated the set s by the fitting curve $(\mathcal{N}_{w,P_x}(t), \mathcal{N}_{w,P_y}(t))$ using the functions $f(t) = \sin((\Delta + t)/2)$ and $g(t) = \sin((\Delta - t)/2)$ at the vector nodes $t = (t_i)_{i=0}^{99}$ with t_i as above.

After training the neural network $\mathcal{N}_{w,p}$, performing Algorithm 1 for 3000 iterations, and with $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$, and $\varepsilon = 10^{-7}$, we have obtained the following weights:

$$w_0^5 = 2.193439722061157227, w_1^5 = 0.6315788030624389648, w_2^5 = 0.4818322956562042236, \\ w_3^5 = 0.4559116661548614502, w_4^5 = 0.6407876610755920410, w_5^5 = 2.386862277984619141,$$

and the following control points:

$$P_0 = (1, 0), P_1 = (1.094536581177390, 2.759880482316126), \\ P_2 = (-3.104044853918599, 2.85189370210171), P_3 = (-3.702726956127786, -2.817369554402328), \\ P_4 = (1.085284693582619, -3.000573791799708), P_5 = (1, 0).$$

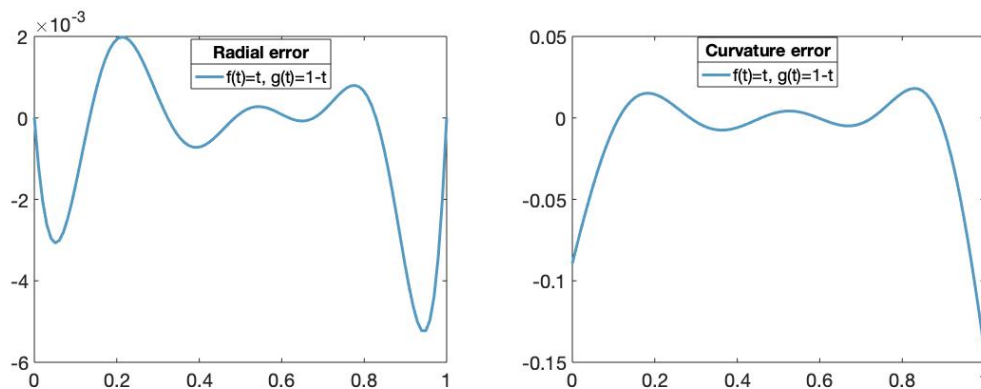


Figure 6. (Left): The x -axis represents the parameters t in $[0, 1]$ and the y -axis represents the radial error value of the fitting curve obtained using the functions $f(t) = t$ and $g(t) = 1 - t, t \in [0, 1]$. (Right): The x -axis represents the parameters t in $[0, 1]$ and the y -axis represents the curvature error value of fitting curve obtained using the functions $f(t) = t$ and $g(t) = 1 - t, t \in [0, 1]$.

We can see in Figure 7 the fitting curve of degree 5 obtained and its control polygon.

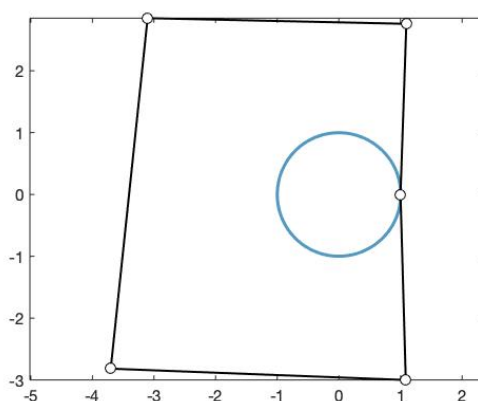


Figure 7. Fitting curve of degree 5 using $f(t) = \sin((\Delta + t)/2)$ and $g(t) = \sin((\Delta - t)/2), t \in [-\Delta, \Delta]$, and its control polygon.

Figure 8 shows its corresponding radial and curvature errors. We can observe that the radial error vanishes at $t = -\Delta, \Delta$ because the approximation is exact for the initial and the last points of the curve.

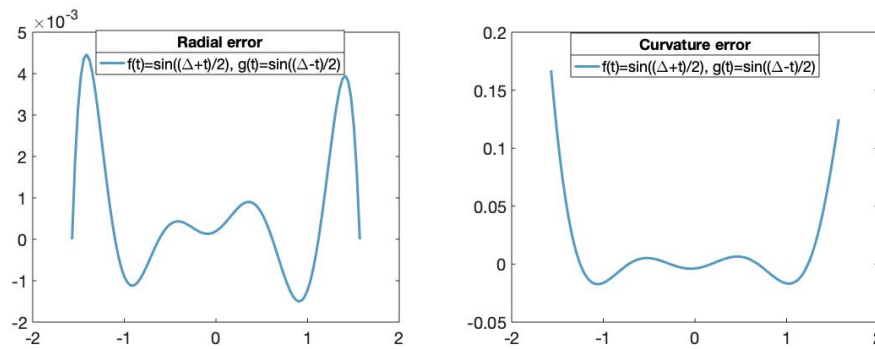


Figure 8. (Left): The x -axis represents the parameters t in $[-\Delta, \Delta]$ and the y -axis represents the radial error value of the fitting curve obtained using the trigonometric basis. (Right): The x -axis represents the parameters t in $[-\Delta, \Delta]$ and the y -axis represents the curvature error value of the fitting curves obtained using the trigonometric basis.

Finally, for different values of n , Figure 9 shows the history of the loss function given in (12) through the training process on 3000 iterations of the neural network $\mathcal{N}_{w,p}$ using the polynomial functions and the trigonometric functions. We can observe, in both cases, that the convergence to the circle of the fitting curves is faster as n increases.

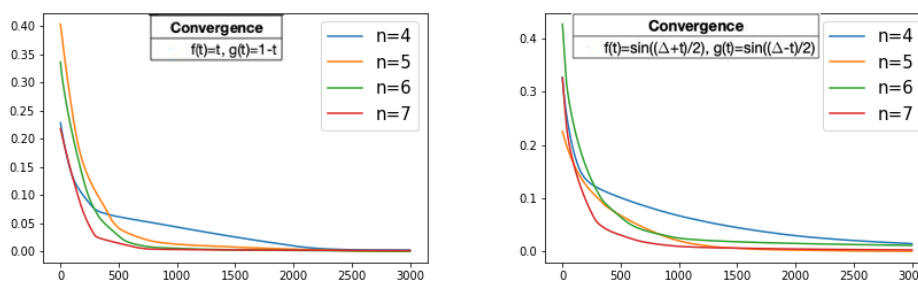


Figure 9. For different values of n , the history of the loss function, i.e., the mean absolute error values, through the training process on 3000 iterations while the fitting curves converges. (Left): Loss values of the fitting curve using $f(t) = t, g(t) = 1 - t, t \in [0, 1]$. (Right): Loss values of the fitting curve using $f(t) = \sin((\Delta + t)/2), g(t) = \sin((\Delta - t)/2), t \in [-\Delta, \Delta]$. The x -axis represents the iteration of the training algorithm and the y -axis represents the mean absolute error value.

4.2. Cycloid Curve

Cycloids are commonly used in manufacturing applications (e.g. gear tooth geometry). The cycloid is a transcendental curve and thus, it cannot be expressed by polynomials exactly. Creating a complex curve in a CAD/CAM system is not always straightforward. In [20], it is shown the need to new methods for approximating this curve. Let us show different approximations to the cycloid by several fitting curves obtained by training the neural network $\mathcal{N}_{w,p}$ using polynomial, trigonometric and hyperbolic functions f and g that satisfy the conditions of Proposition 1.

The cycloid is given by the parametric equations

$$\begin{cases} x_{cc}(t) = t - \sin t, \\ y_{cc}(t) = 1 - \cos t, \end{cases} \tag{15}$$

$t \in [0, 2\pi]$. We have taken the following sequence of points $s = (s_i)_{i=0}^{99}$ on the cycloid (15):

$$s_i = (x_{cc}(2\pi t_i), y_{cc}(2\pi t_i)), \quad i = 0, \dots, 99,$$

being the parameters $t_i = i/99, i = 0, \dots, 99$.

For $n = 10$, we have approximated the set s by two fitting curves at the vector nodes $t = (t_i)_{i=0}^{99}$ with t_i as above. One of the fitting curves is obtained using the functions $f(t) = t$ and $g(t) = 1 - t$, and the other fitting curve is obtained using the functions $f(t) = t^2$ and $g(t) = 1 - t^2$.

We can see in Figure 10, the fitting curves and their corresponding fitting error. The fitting error is given by the Euclidean norm between the approximated curve—in this case, the cycloid curve—and the obtained fitting curve. Observe that the cycloid curve is better approximated by the fitting curve obtained using the functions $f(t) = t^2$ and $g(t) = 1 - t^2$ than by the fitting curve obtained using the functions $f(t) = t$ and $g(t) = 1 - t$.

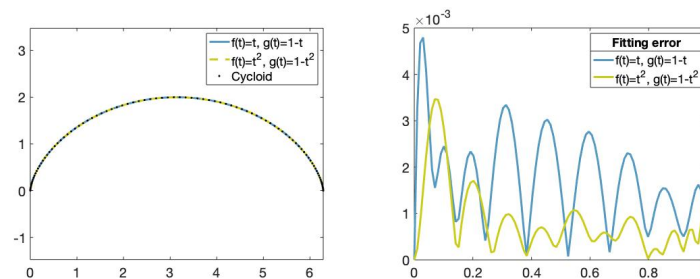


Figure 10. (Left): Set of data points on the cycloid (dotted), fitting curve obtained using the functions $f(t) = t, g(t) = 1 - t, t \in [0, 1]$ (blue) and fitting curve obtained using the functions $f(t) = t^2, g(t) = 1 - t^2, t \in [0, 1]$ (green). (Right): Fitting error comparison. The x -axis represents the parameters t in $[0, 1]$ and the y -axis represents the fitting error value of the fitting curves obtained using the functions $f(t) = t, g(t) = 1 - t, t \in [0, 1]$ (green) and the functions $f(t) = t^2, g(t) = 1 - t^2, t \in [0, 1]$ (blue).

Let us see two more approximations of the cycloid. We have taken the following sequence of points $s = (s_i)_{i=0}^{99}$ on the cycloid (15):

$$s_i = (x_{cc}((t_i/\Delta + 1)\pi), y_{cc}((t_i/\Delta + 1)\pi)), \quad i = 0, \dots, 99,$$

being the parameters $t_i = -\Delta + 2\Delta i/99, i = 0, \dots, 99$, with $0 < \Delta < \pi/2$.

For $n = 10$, we have approximated the set s by two fitting curves at the vector nodes $t = (t_i)_{i=0}^{99}$ with t_i as above. One of the fitting curves is obtained by training the neural network $\mathcal{N}_{w,p}$ using the trigonometric functions $f(t) = \sin((\Delta + t)/2)$ and $g(t) = \sin((\Delta - t)/2)$ and the other one using the hyperbolic functions $f(t) = \sinh((\Delta + t)/2)$ and $g(t) = \sinh((\Delta - t)/2)$.

We can see in Figure 11, the fitting curves and their corresponding fitting error. Let us observe that the cycloid is better approximated by the fitting curve obtained using the hyperbolic functions than the fitting curve obtained using the trigonometric functions.

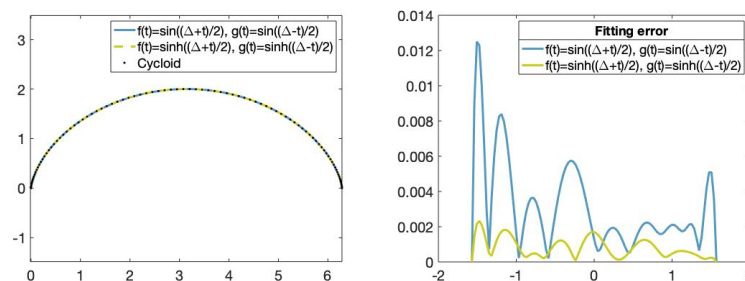


Figure 11. (Left): Set of data points on the cycloid (dotted), fitting curve obtained using the trigonometric functions $f(t) = \sin((\Delta + t)/2)$ and $g(t) = \sin((\Delta - t)/2)$ and fitting curve obtained using the hyperbolic functions $f(t) = \sinh((\Delta + t)/2)$ and $g(t) = \sinh((\Delta - t)/2)$. (Right): Fitting error comparison. The x -axis represents the parameters t in $[-\Delta, \Delta]$ and the y -axis represents the curvature error value of the fitting curve obtained using the trigonometric and hyperbolic bases.

4.3. Archimedean Spiral Curve

Finally, let us show different approximations of the Archimedean spiral given by the parametric equations

$$\begin{cases} x_{sa}(t) = t \cos(t), \\ y_{sa}(t) = t \sin(t), \end{cases} \tag{16}$$

$t \in [0, 4\pi]$. First, let us see an approximation obtained training the neural network $\mathcal{N}_{w,P}$ using polynomial functions. We have taken the following sequence of points $s = (s_i)_{i=0}^{99}$ on the Archimedean spiral (16):

$$s_i = (x_{sa}(4\pi t_i), y_{sa}(4\pi t_i)), \quad i = 0, \dots, 99,$$

being the parameters $t_i = i/99, i = 0, \dots, 99$.

For $n = 11$, we have approximated the set s by the fitting curve using the functions $f(t) = t$ and $g(t) = 1 - t$ at the vector nodes $t = (t_i)_{i=0}^{99}$ with t_i as above. Figure 12 shows the fitting curve of degree 11 and its corresponding fitting error.

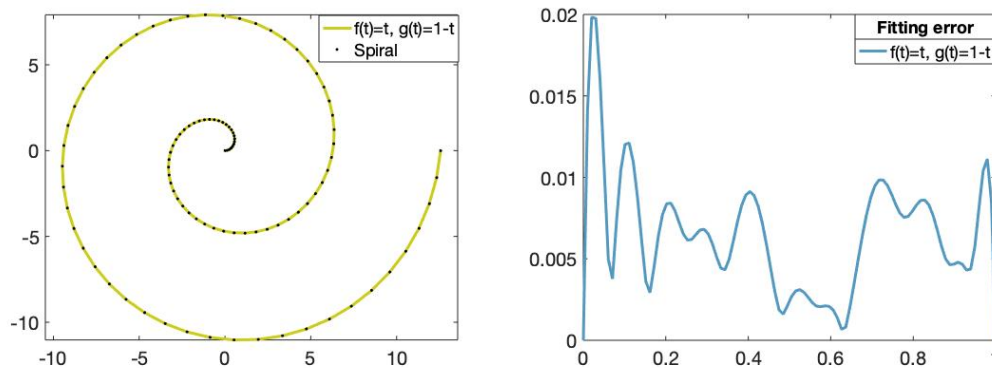


Figure 12. (Left): Set of data points on the Archimedean spiral (dotted) and the fitting curve of degree 11 obtained using the functions $f(t) = t$ and $g(t) = 1 - t, t \in [0, 1]$. (Right): The x -axis represents the parameters t in $[0, 1]$ and the y -axis represents the fitting error value of the fitting curve obtained using the functions $f(t) = t$ and $g(t) = 1 - t$.

Let us see other examples in which the Archimedean spiral is approximated by the neural network using hyperbolic functions. We have taken the following sequence of points $s = (s_i)_{i=0}^{99}$ on the Archimedean spiral (16):

$$s_i = (x_{sa}((t_i + \Delta)2\pi/\Delta), y_{sa}((t_i + \Delta)2\pi/\Delta)), \quad i = 0, \dots, 99,$$

being the parameters $t_i = -\Delta + 2\Delta i/99, i = 0, \dots, 99$. For $n = 11$, we have approximated the set s by the fitting curve using the functions $f(t) = \sinh((\Delta + t)/2)$ and $g(t) = \sinh((\Delta - t)/2)$ at the vector nodes $t = (t_i)_{i=0}^{99}$ with t_i as above.

Figure 13 shows the fitting curve of degree 11 and its corresponding fitting error.

We can see in Figure 14, the convergence of the two above fitting curves. Let us observe that the fitting curve corresponding to the hyperbolic rational basis has a faster convergence. Therefore, once again, it seems that the choice of the functions f and g , hence the choice of the rational basis, is relevant to the approximation.

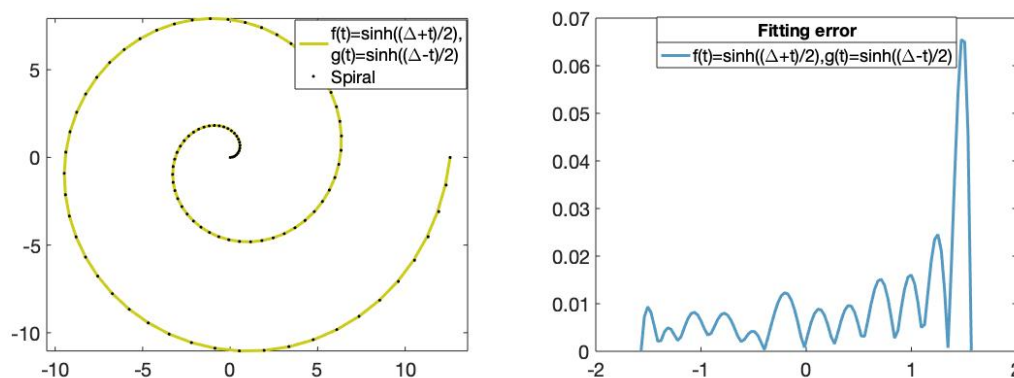


Figure 13. (Left): Set of data points on the Archimedean spiral (dotted) and the fitting curve of degree 11 with the functions $f(t) = \sinh ((\Delta + t)/2)$ and $g(t) = \sinh ((\Delta - t)/2)$. (Right): The x -axis represents the parameters t in $[-\Delta, \Delta]$ and the y -axis represents the fitting error value of the fitting curve obtained using the hyperbolic basis.

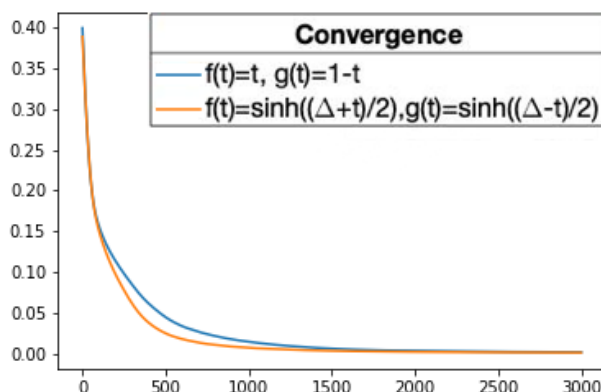


Figure 14. The history of the loss values, i.e., the mean absolute error values, through 3000 iterations of the training algorithm while the fitting curves converge is pictured. The blue line corresponds to the fitting curve of degree 11 obtained using $f(t) = t$ and $g(t) = 1 - t, t \in [0, 1]$, and the orange line corresponds to the fitting curve of degree 11 obtained using $f(t) = \sinh ((\Delta + t)/2)$ and $g(t) = \sinh ((\Delta - t)/2)$. The x -axis represents the iteration of the training algorithm and the y -axis represents the mean absolute error value. The values are the mean of 50 repetitions.

4.4. Comparison of Least-Squares Fitting and the Neural Network $\mathcal{N}_{w,p}$

In this section, the proposed neural network is compared with the least-squares method and the regularized least-squares method. It is well known that least-squares fitting is a common procedure to find the best fitting curve to a given set of data points by minimizing the sum of the squares of the data points from the curve ([9,21,22]). The least-squares method is sensitive to small perturbations in data and, in those cases, regularization methods can be applied such as the regularized least-squares method (see [23]). Therefore, two experiments have been developed. The set of data points used in the first experiment is a non-noisy parametrization of known curves, so the least-squares method has been applied. In the second experiment, we used an image to obtain a noisy set of data points and applied the regularized least-squares method.

The problem that we want to solve is stated as follows. Suppose that f and g are functions defined on $[a, b]$ satisfying the conditions of Proposition 1. Consider a set of parameters $a \leq t_0 < \dots \leq t_\ell \leq b$

and a sequence of data points $s_0, \dots, s_\ell \in \mathbb{R}^k$, where each parameter t_i is associated with a data point s_i . For some $n \leq \ell$, we want to compute a rational curve

$$c(t) = \sum_{i=0}^n \frac{w_i^n \binom{n}{i} f^i(t) g^{n-i}(t)}{\sum_{i=0}^n w_i^n \binom{n}{i} f^i(t) g^{n-i}(t)} P_i, \quad t \in [a, b], \tag{17}$$

minimizing the sum of the squares of the deviations from the set of data points $s = (s_i)_{i=0}^\ell$, that is, $\sum_{i=0}^\ell (s_i - c(t_i))^2$. In order to compute the control points $P = (P_i)_{i=0}^n$ of the fitting curve, we have to solve, in the least-squares sense, the overdetermined linear system $AP = s$, where the matrix A is

$$A = \left(\frac{w_i^n \binom{n}{i} f^i(t_j) g^{n-i}(t_j)}{\sum_{i=0}^n w_i^n \binom{n}{i} f^i(t_j) g^{n-i}(t_j)} \right)_{0 \leq i \leq n, 0 \leq j \leq \ell}.$$

In this fourth experiment, let us see two examples comparing the fitting curves obtained applying this traditional method and the fitting curves obtained by training the neural network $\mathcal{N}_{w,P}$. We have seen previously an approximation of the circle (14) obtained with the neural network $\mathcal{N}_{w,P}$ for $n = 5$ using the polynomial functions $f = t$ and $g = 1 - t, t \in [0, 1]$. Besides, we have seen an approximation of the Archimedean spiral (16) obtained with the neural network $\mathcal{N}_{w,P}$ for $n = 11$ using the polynomial functions $f = t$ and $g = 1 - t, t \in [0, 1]$. Using the same parameters and sets of data points, we solved the corresponding overdetermined linear systems $AP = s$ to obtain the control points of the curve that best fits the given data points in the least-squares sense. For this purpose, following the steps shown in Section 6.4 of [1], we have obtained the solutions of $AP = s$, firstly, by using the Matlab command `SVD` and, secondly, by using the Matlab command `mldivide`.

Since the weights are unknown and are necessary for solving the linear system $AP = s$ in the least-squares sense, these have been randomly chosen within a positive range. Let us observe that one of the advantages of the neural network $\mathcal{N}_{w,P}$ with respect to the least-squares method is that the neural network $\mathcal{N}_{w,P}$ not only finds suitable control points, but it also finds suitable weights to fit the given set of data points.

In Table 3, the execution time of the least-squares method using a different number of weights and control points is provided. We can see that, although for low-degree fitting curves, the least-squares method is faster than the method proposed in this paper, for the fitting curves of degree greater than $n = 18$, the proposed method is faster than the least-squares method.

We can see in Figures 15 and 16 that the fitting curves obtained by training the neural network are more accurate than the fitting curves obtained by applying the traditional least-squares method.

In the last experiment, we have taken 31 points from a baroque image as a set of data points (see Figure 17). We have approximated them by two fitting curves of degree 8 with hyperbolic functions $f(t) = \sin((\Delta + t)/2)$ at the vector nodes $t = (t_i)_{i=0}^{30}$, such that $t_i = -\Delta + 2\Delta(i - 1)/30, 0 < \Delta < \pi/2$. The first fitting curve was obtained by training the neural network $\mathcal{N}_{w,P}$ during 1000, reaching a final mean absolute error on the set of data points of 9×10^{-3} . Besides, this example shows the robustness of the proposed method to noisy data points. The second fitting curve was obtained with random vector weights and the regularized least-squares method by using an adaptation of the Matlab library available in [24]. This last fitting curve reached a mean absolute error of 1.95×10^{-2} . Visually, in Figure 17, it is appreciated that the fitting curve obtained by training the neural network $\mathcal{N}_{w,P}$ achieves better curvature than the regularized least-squares method (which is more evident at the beginning and at the end of the curve). Finally, we would like to highlight that we have observed in all the executed experiments a good performance of the neural network for small values of n , in spite of the complexity of the set of data points.

Table 3. For different number of weights and control points, the time of execution of the least-squares method using the Matlab commands `mldivide` and `SVD`, and the time of execution of the Algorithm 1 for 3000 iterations are provided. The values have been measured in seconds with a set of data points of size 100 and they are the mean of 5 repetitions.

$n + 1$	Least-Squares Method		Algorithm 1
	<code>mldivide</code>	<code>SVD</code>	3000 Iterations
5	63.3014	65.6015	189.5386
10	167.0402	237.0386	268.8726
15	309.5811	457.8987	347.6139
20	478.1141	774.3472	429.2819
30	961.2860	978.6830	568.5647
50	2552.4064	4097.9278	850.6713

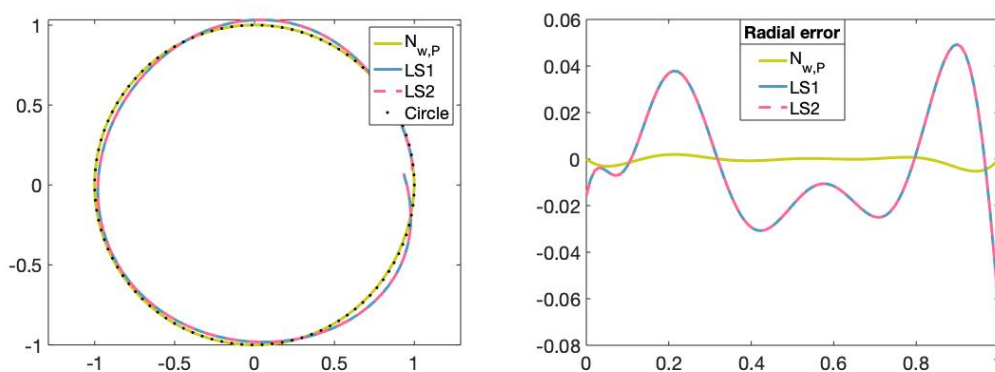


Figure 15. (Left): Set of data points on circle (dotted), fitting curve of degree 5 obtained with the neural network, fitting curve of degree 5 obtained with the least-squares method by using the Matlab command `mldivide` (LS1), fitting curve of degree 5 obtained with the least-squares method by using the Matlab command `SVD` (LS2). (Right): The x -axis represents the parameters t in $[0, 1]$ and the y -axis represents the radial error value of the fitting curves obtained with the different methods.

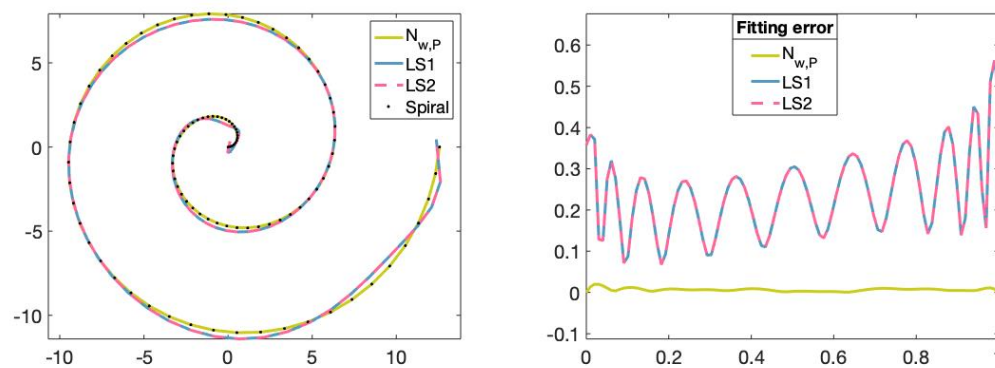


Figure 16. (Left): Set of points on the Archimedean spiral curve, fitting curve of degree 11 obtained with the neural network, fitting curve of degree 11 obtained with the least-squares method by using the Matlab command `mldivide` (LS1) and fitting curve of degree 11 obtained with the least-squares method using the Matlab command `SVD` (LS2). (Right): The x -axis represents the parameters t in $[0, 1]$ and the y -axis represents the fitting error value of the fitting curves obtained with the different methods.



Figure 17. Noisy set of data points obtained from the baroque image (points in blue), fitting curve obtained by training the proposed neural network (points in green) and fitting curve obtained using the regularized least-squares method (pink). Baroque motif image source: [Freepik.com](https://www.freepik.com).

5. Conclusions and Future Work

In this work, we have tackled the problem of finding a rational curve to fit a given set of data points. To solve this issue, we have proposed a one-hidden-layer neural network based on a general class of totally positive rational bases belonging to spaces mixing algebraic, trigonometric and hyperbolic polynomials, thus expanding the potential range of applications to include more difficult shapes. In order to obtain the weights and control points of the rational curve to fit the data points, the neural network is trained with an optimization algorithm to update the weights and control points while decreasing a loss function. The fitting curves of the numerical experiments show that for certain curves, the use of particular rational bases provides better results.

In future work, we wish to extend the neural network to obtain a suitable parametrization of the data points. The choice of the parameters can help to improve the approximation. Additionally, we plan to apply our method to curves, surfaces and high-dimensional data, and analyze, as in [25] for Bezier curves, its application to industrial software, CAD/CAM systems (such as Blender (<https://www.blender.org/>), Maya (<https://www.autodesk.es/products/maya/overview>) or Solid Edge (<https://solidedge.siemens.com/en/>)), and other real-world problems. Finally, we plan to explore the applicability of our approach to the resolution of linear differential equations (cf. [26,27]).

Author Contributions: Conceptualization, R.G.-D., E.M., E.P.-H. and B.R.; Investigation, R.G.-D., E.M., E.P.-H. and B.R.; Methodology, R.G.-D., E.M., E.P.-H. and B.R.; Software, R.G.-D., E.M., E.P.-H. and B.R.; Supervision, R.G.-D., E.M., E.P.-H. and B.R.; Validation, R.G.-D., E.M., E.P.-H. and B.R.; Writing—original draft, R.G.-D., E.M., E.P.-H. and B.R.; Writing—review and editing, R.G.-D., E.M., E.P.-H. and B.R. All authors have read and agreed to the published version of the manuscript.

Funding: MICINN, FEDER/UE under grant PID2019-107339GB-I00, PGC2018-096321-B-I00 (MCIU/AEI), Gobierno de Aragón (E41_17R) and Feder 2014–2020 “Construyendo Europa desde Aragón”.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Iglesias, A.; Gálvez, A.; Avila, A. Discrete Bézier Curve Fitting with Artificial Immune Systems. In *Intelligent Computer Graphics 2012*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 59–75. [[CrossRef](#)]
- Hoffmann, M.; Várady, L. Free-form Surfaces for scattered data by neural networks. *J. Geom. Graph.* **1998**, *2*, 1–6.
- Delgado, J.; Peña, J.M. A Comparison of Different Progressive Iteration Approximation Methods. In *Mathematical Methods for Curves and Surfaces*; Dæhlen, M., Floater, M., Lyche, T., Merrien, J.L., Mørken, K., Schumaker, L.L., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 136–152.
- Ando, T. Totally positive matrices. *Linear Algebra Appl.* **1987**, *90*, 165–219. [[CrossRef](#)]
- Peña, J. Shape preserving representations for trigonometric polynomial curves. *Comput. Aided Geom. Des.* **1997**, *14*, 5–11. [[CrossRef](#)]
- Carnicer, J.; Peña, J. Totally positive bases for shape preserving curve design and optimality of B-splines. *Comput. Aided Geom. Des.* **1994**, *11*, 633–654. [[CrossRef](#)]
- Carnicer, J.; Peña, J. Shape preserving representations and optimality of the Bernstein basis. *Adv. Comput. Math* **1993**, *1*, 173–196. [[CrossRef](#)]
- Mainar, E.; Peña, J.; Rubio, B. Evaluation and subdivision algorithms for general classes of totally positive rational bases. *Comput. Aided Geom. Des.* **2020**, *81*, 101900. [[CrossRef](#)]

9. Farin, G. *Curves and Surfaces for Computer Aided Geometric Design (Fifth Ed.): A Practical Guide*; Academic Press Professional Inc.: Cambridge, MA, USA, 2002.
10. Iglesias, A.; Gálvez, A.; Collantes, M. Global-support rational curve method for data approximation with bat algorithm. In Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, Bayonne, France, 14–17 September 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 191–205.
11. Mehnen, J.; Weinert, K. Discrete NURBS-Surface Approximation using an Evolutionary Strategy. *Reihe Comput. Intell.* **2001**, *87*. [[CrossRef](#)]
12. Van To, T.; Kositviwat, T. Using Rational B-Spline Neural Networks for Curve Approximation. In Proceedings of the 7th WSEAS International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, WI, USA, 7–9 November 2009; MMACTE'05, pp. 42–50.
13. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:cs.LG/1412.6980.
14. Manni, C.; Pelosi, F.; Lucia Sampoli, M. Generalized B-splines as a tool in isogeometric analysis. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 867–881. [[CrossRef](#)]
15. Sánchez-Reyes, J. Harmonic rational Bézier curves, p-Bézier curves and trigonometric polynomials. *Comput. Aided Geom. Des.* **1998**, *15*, 909–923. [[CrossRef](#)]
16. Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*; The MIT Press: Cambridge, MA, USA, 2012.
17. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *arXiv* **2015**, arXiv:cs.DC/1603.04467.
18. Carnicer, J.; Mainar, E.; Peña, J. Representing circles with five control points. *Comput. Aided Geom. Des.* **2003**, *20*, 501–511. [[CrossRef](#)]
19. Carnicer, J.; Mainar, E.; Peña, J. A totally positive basis for circle approximations. *Rev. Real Acad. Cienc. Exactas Fis. Nat. Ser. A Mat.* **2019**, *113*. [[CrossRef](#)]
20. Díaz Pérez, L.; Rubio Serrano, B.; Albajez García, J.; Yague Fabra, J.; Mainar Maza, E.; Torralba Gracia, M. Trajectory Definition with High Relative Accuracy (HRA) by Parametric Representation of Curves in Nano-Positioning Systems. *Micromachines* **2019**, *10*, 597. [[CrossRef](#)] [[PubMed](#)]
21. Marco, A.; Martínez, J.J. Polynomial least squares fitting in the Bernstein basis. *Linear Algebra Its Appl.* **2010**, *433*, 1254–1264. [[CrossRef](#)]
22. Mainar, E.; Peña, J.; Rubio, B. Accurate least squares fitting with a general class of shape preserving bases. In Proceedings of the Fifteenth International Conference Zaragoza-Pau on Mathematics and its Applications, Jaca, Spain, 10–12 September 2019; pp. 183–192.
23. Volkov, V.V.; Erokhin, V.I.; Kakaev, V.V.; Onufrei, A.Y. Generalizations of Tikhonov's regularized method of least squares to non-Euclidean vector norms. *Comput. Math. Math. Phys.* **2017**, *57*, 1416–1426. [[CrossRef](#)]
24. Rao, S. Regularized Least Square: Tikhonov Regularization Test for Hilbert Matrix. MATLAB Central File Exchange. 2020. Available online: <https://es.mathworks.com/matlabcentral/fileexchange/58736-regularized-least-square-tikhonov-regularization-test-for-hilbert-matrix> (accessed on 25 November 2020).
25. Fitter, H.N.; Pandey, A.B.; Patel, D.D.; Mistry, J.M. A Review on Approaches for Handling Bezier Curves in CAD for Manufacturing. *Procedia Eng.* **2014**, *97*, 1155–1166. [[CrossRef](#)]
26. Calin, I.; Öchsner, A.; Vlase, S.; Marin, M. Improved rigidity of composite circular plates through radial ribs. *Proc. Inst. Mech. Eng. Part L J. Mater. Des. Appl.* **2018**, *233*. [[CrossRef](#)]
27. Groza, G.; Pop, N. Approximate solution of multipoint boundary value problems for linear differential equations by polynomial functions. *J. Differ. Equations Appl.* **2008**, *14*, 1289–1309. [[CrossRef](#)]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

ARTICLE 4

- [75] E. Mainar, J.M. Peña, B.Rubio, Accurate computations with Wronskian matrices, *Calcolo* **58**, 1 (2021).
<https://doi.org/10.1007/s10092-020-00392-4>



Accurate computations with Wronskian matrices

E. Mainar¹ · J. M. Peña¹ · B. Rubio¹ 

Received: 16 June 2020 / Revised: 18 November 2020 / Accepted: 27 November 2020
© Istituto di Informatica e Telematica (IIT) 2021

Abstract

In this paper we provide algorithms for computing the bidiagonal decomposition of the Wronskian matrices of the monomial basis of polynomials and of the basis of exponential polynomials. It is also shown that these algorithms can be used to perform accurately some algebraic computations with these Wronskian matrices, such as the calculation of their inverses, their eigenvalues or their singular values and the solutions of some linear systems. Numerical experiments illustrate the results.

Keywords Accurate computations · Wronskian matrices · Bidiagonal decompositions

1 Introduction

The accuracy of the calculations is a desirable goal in Computational Mathematics. Let us recall that an algorithm can be performed with high relative accuracy (HRA) if it does not include subtractions of numbers having the same sign (except of the initial data if they are exact), that is, if it only includes products, divisions, additions of numbers of the same sign and subtractions of the initial data having the same sign provided that they are not affected by errors (cf. [5]). For some structured classes of matrices such algorithms have been found through an adequate parameterization of the matrix. In particular, this has been achieved for some subclasses of totally positive (TP) matrices. In [12] it was shown that, given the bidiagonal factorization of a nonsingular TP matrix A with HRA, we can compute with HRA its eigenvalues and singular values, the matrix A^{-1} and even the solution of $Ax = b$ for vectors b with alternating signs. Among the subclasses of TP matrices for which the bidiagonal factorization has been

✉ B. Rubio
brubio@unizar.com

E. Mainar
esmemain@unizar.com

J. M. Peña
jmpena@unizar.com

¹ Departamento de Matemática Aplicada/IUMA, Universidad de Zaragoza, Zaragoza, Spain

obtained with HRA (cf. [3, 4, 13, 14]), there are many examples of collocation matrices $(u_{j-1}(t_i))_{1 \leq i, j \leq n+1}$ of systems (u_0, \dots, u_n) of functions defined on a real subset I ($t_1 < t_2 < \dots < t_{n+1}$ in I). However, up to now, there are no examples of accurate computations for matrices involving derivatives of the basis functions. This paper presents some examples of Wronskian matrices for which many algebraic computations can be performed accurately. These Wronskian matrices come from applications in computer aided geometric design (CAGD) and they can also arise in Hermite interpolation problems, in particular in Taylor interpolation problems.

The paper is organized as follows. In Sect. 2, we provide basic concepts and tools. In particular we recall the Neville elimination procedure and the bidiagonal factorization of a nonsingular TP matrix. This factorization provides the adequate parameterization to derive the accurate algorithms with these matrices. Section 3 shows that the bidiagonal factorization of the Wronskian matrices of the monomial basis of polynomials can be performed with HRA. In Sect. 4 we first prove that Wronskian matrices of the basis of exponential polynomials on positive real numbers are strictly totally positive. We also provide the bidiagonal factorization of these matrices. The computation with HRA of this factorization should require the evaluation with HRA of the involved exponential functions. Although this cannot be guaranteed, numerical experiments show an accuracy similar to the obtained for the monomial basis. Finally, Sect. 5 includes numerical experiments showing the accuracy of the presented methods for the computation of all eigenvalues, all singular values, the inverses and the solution of linear systems.

2 Notations and auxiliary results

As usual, given an n -times continuously differentiable function f and x in its parameter domain, $f'(x)$ denotes the first derivative of f at x and, for any $i \leq n$, $f^{(i)}(x)$ denotes the i th derivative of f at x . Let us recall that for a given basis (u_0, \dots, u_n) of a space of n -times continuously differentiable functions, defined on a real interval I and $x \in I$, the *Wronskian matrix* at x is defined by

$$W(u_0, \dots, u_n)(x) := (u_{j-1}^{(i-1)}(x))_{i,j=1, \dots, n+1}.$$

A matrix is totally positive: TP (respectively, strictly totally positive: STP) if all its minors are nonnegative (respectively, positive). Two recent books on these matrices are [6, 16], where many applications of these matrices are presented, as well as in [1].

Neville elimination is an alternative procedure to Gaussian elimination and has been used to characterize TP and STP matrices. Given a nonsingular matrix $A = (a_{i,j})_{1 \leq i, j \leq n+1}$, Neville elimination computes a matrix sequence

$$A^{(1)} := A \rightarrow A^{(2)} \rightarrow \dots \rightarrow A^{(n+1)} = U,$$

such that, for $1 \leq k \leq n$, $A^{(k+1)} = (a_{i,j}^{(k+1)})_{1 \leq i, j \leq n+1}$ has zeros below its main diagonal in the first k columns and is computed from $A^{(k)} = (a_{i,j}^{(k)})_{1 \leq i, j \leq n+1}$ by:

$$a_{i,j}^{(k+1)} := \begin{cases} a_{i,j}^{(k)}, & \text{if } 1 \leq i \leq k, \\ a_{i,j}^{(k)} - \frac{a_{i,k}^{(k)}}{a_{i-1,k}^{(k)}} a_{i-1,j}^{(k)}, & \text{if } k+1 \leq i, j \leq n+1 \text{ and } a_{i-1,k}^{(k)} \neq 0, \\ a_{i,j}^{(k)}, & \text{if } k+1 \leq i \leq n+1 \text{ and } a_{i-1,k}^{(k)} = 0. \end{cases}$$

The element $p_{i,j} := a_{i,j}^{(j)}, 1 \leq j \leq i \leq n+1$, is called the (i, j) pivot and, in particular, $p_{i,i}$ is a diagonal pivot of the Neville elimination of A . If all the pivots are nonzero then Neville elimination can be carried out without row exchanges. In this case, by Lemma 2.6 of [7],

$$p_{i,1} = a_{i,1}, \quad 1 \leq i \leq n+1, \\ p_{i,j} = \frac{\det A[i-j+1, \dots, i|1, \dots, j]}{\det A[i-j+1, \dots, i-1|1, \dots, j-1]}, \quad 1 < j \leq i \leq n+1, \tag{1}$$

where given increasing sequences of integers α and β , $A[\alpha|\beta]$ denotes the submatrix of A containing rows of places α and columns of places β . Moreover,

$$m_{i,j} := \begin{cases} a_{i,j}^{(j)} / a_{i-1,j}^{(j)} = p_{i,j} / p_{i-1,j}, & \text{if } a_{i-1,j}^{(j)} \neq 0, \\ 0, & \text{if } a_{i-1,j}^{(j)} = 0, \end{cases}, \quad 1 \leq j < i \leq n+1, \tag{2}$$

is called the (i, j) multiplier of the Neville elimination of A .

By Theorem 4.2 and the arguments of p. 116 of [9], a nonsingular TP matrix $A = (a_{i,j})_{1 \leq i, j \leq n+1}$ admits a factorization of the form

$$A = F_n F_{n-1} \dots F_1 D G_1 \dots G_{n-1} G_n, \tag{3}$$

where F_i and G_i are the TP, lower and upper triangular bidiagonal matrices given by

$$F_i = \begin{pmatrix} 1 & & & & & & & & & \\ 0 & 1 & & & & & & & & \\ & & \ddots & & & & & & & \\ & & & \ddots & & & & & & \\ & & & & 0 & 1 & & & & \\ & & & & & m_{i+1,1} & 1 & & & \\ & & & & & & m_{i+2,2} & 1 & & \\ & & & & & & & \ddots & \ddots & \\ & & & & & & & & m_{n+1,n+1-i} & 1 \end{pmatrix}, \\ G_i^T = \begin{pmatrix} 1 & & & & & & & & & \\ 0 & 1 & & & & & & & & \\ & & \ddots & & & & & & & \\ & & & \ddots & & & & & & \\ & & & & 0 & 1 & & & & \\ & & & & & \tilde{m}_{i+1,1} & 1 & & & \\ & & & & & & \tilde{m}_{i+2,2} & 1 & & \\ & & & & & & & \ddots & \ddots & \\ & & & & & & & & \tilde{m}_{n+1,n+1-i} & 1 \end{pmatrix}, \tag{4}$$

and $D = \text{diag}(p_{1,1}, p_{2,2}, \dots, p_{n+1,n+1})$ has positive diagonal entries. If, in addition, the entries m_{ij}, \tilde{m}_{ij} satisfy

$$m_{ij} = 0 \Rightarrow m_{hj} = 0 \quad \forall h > i$$

and

$$\tilde{m}_{ij} = 0 \Rightarrow \tilde{m}_{ik} = 0 \quad \forall k > j,$$

then the decomposition (3) is unique. The diagonal entries $p_{i,i}$ of D are the diagonal pivots of the Neville elimination of A and the elements $m_{i,j}$ and $\tilde{m}_{i,j}$ are the multipliers of the Neville elimination of A and A^T , respectively. We shall denote the bidiagonal decomposition (3) of a TP matrix A by $BD(A)$ (see [11]). Given $BD(A)$, using the results in [7–9], a bidiagonal decomposition of A^{-1} can be computed as

$$A^{-1} = \tilde{G}_1 \tilde{G}_2 \cdots \tilde{G}_n D^{-1} \tilde{F}_n \cdots \tilde{F}_2 \tilde{F}_1, \quad (5)$$

where \tilde{F}_i and \tilde{G}_i , $i = 1, \dots, n$, are the lower and upper triangular bidiagonal matrices of the form of F_i and G_i , respectively, but replacing the off-diagonal entries $\{m_{i+1,1}, \dots, m_{n+1,n+1-i}\}$ and $\{\tilde{m}_{i+1,1}, \dots, \tilde{m}_{n+1,n+1-i}\}$ by $\{-m_{i+1,i}, \dots, -m_{n+1,i}\}$ and $\{-\tilde{m}_{i+1,i}, \dots, -\tilde{m}_{n+1,i}\}$ respectively. From Theorem 4.1 of [7] and p. 116 of [9], a given matrix $A = (a_{ij})_{1 \leq i, j \leq n+1}$ is STP if and only if the Neville elimination of A and A^T can be performed without row exchanges, all the multipliers of the Neville elimination of A and A^T are positive and all the diagonal pivots of the Neville elimination of A are positive.

Let us recall that a real value x is obtained with high relative accuracy (HRA) if the relative error of the computed value \tilde{x} satisfies

$$\frac{\|x - \tilde{x}\|}{\|x\|} < Ku,$$

where K is a positive constant independent of the arithmetic precision and u is the unit round-off. HRA implies that the relative errors of the computations are of the order of the machine precision. So, performing an algorithm with HRA is a very desirable goal. An algorithm can be computed with HRA when it only uses products, quotients, sums of numbers of the same sign or subtraction of initial data (cf. [5, 11]).

In [12] it was shown that if $BD(A)$, the bidiagonal factorization (3) of a nonsingular TP matrix A , is computed with HRA then we can also compute with HRA its eigenvalues and singular values, the matrix A^{-1} and even the solution of $Ax = b$ for vectors b with alternating signs.

In the following sections we shall obtain the bidiagonal factorization (3) of Wronskian matrices associated with some bases with applications in CAGD, analyzing whether it can be computed with HRA.

3 Wronskian matrices of monomial bases

The monomial basis of the space \mathbf{P}^n of polynomials of degree less than or equal to n is (m_0, \dots, m_n) with

$$m_i(x) := x^i, \quad i = 0, \dots, n. \quad (6)$$

Given $x_0 \in \mathbb{R}$, we can define a Taylor basis (n_0, \dots, n_n) of \mathbf{P}^n by

$$n_i(x) := \frac{(x - x_0)^i}{i!}, \quad i = 0, \dots, n. \quad (7)$$

It can be checked that

$$(m_0, \dots, m_n) = (n_0, \dots, n_n)W,$$

where $W := W(m_0, \dots, m_n)(x_0)$. Equivalently, we can also write

$$(n_0, \dots, n_n) = (m_0, \dots, m_n)W^{-1}.$$

In this section we are going to obtain the bidiagonal factorization (3) of W and W^{-1} and see that they can be computed with HRA. First let us prove the following auxiliary result.

Lemma 1 *Given $i, j \in \mathbb{N}$, then*

$$\frac{1}{i!}m_j^{(i)}(x) = \frac{1}{(i-1)!}m_{j-1}^{(i-1)}(x) + \frac{x}{i!}m_{j-1}^{(i)}(x), \quad x \in \mathbb{R}. \quad (8)$$

Proof Let us prove the result by induction on i . For $i = 1$ and $j \in \mathbb{N}$, taking into account that $m_j'(x) = (xm_{j-1}(x))'$, we have

$$m_j'(x) = m_{j-1}(x) + xm_{j-1}'(x), \quad x \in \mathbb{R},$$

and so formula (8) holds. Let us now suppose that (8) holds for $i > 1$ and $j \in \mathbb{N}$. Then we have

$$\begin{aligned} \frac{1}{i!}m_j^{(i+1)}(x) &= \left(\frac{1}{(i-1)!}m_{j-1}^{(i-1)}(x) + \frac{x}{i!}m_{j-1}^{(i)}(x) \right)' \\ &= \frac{i+1}{i!}m_{j-1}^{(i)}(x) + \frac{x}{i!}m_{j-1}^{(i+1)}(x), \quad x \in \mathbb{R}, \end{aligned}$$

and we can deduce that, for $j \in \mathbb{N}$,

$$\frac{1}{(i+1)!}m_j^{(i+1)}(x) = \frac{1}{i!}m_{j-1}^{(i)}(x) + \frac{x}{(i+1)!}m_{j-1}^{(i+1)}(x), \quad x \in \mathbb{R}.$$

□

For a given $x \in \mathbb{R}$, $k, n \in \mathbb{N}$ with $k \leq n$, let $U_{k,n} = (u_{ij})_{1 \leq i,j \leq n+1}$ be the upper triangular bidiagonal matrix with unit diagonal entries and such that

$$u_{i,i+1} := 0, \quad i = 1, \dots, k - 1, \quad u_{i,i+1} := x, \quad i = k, \dots, n. \tag{9}$$

In the following result we obtain an explicit expression of the entries of the product matrix $U_{1,n} \cdots U_{n,n}$.

Proposition 1 For a given $x \in \mathbb{R}$ and $n \in \mathbb{N}$, let

$$U_n := U_{1,n} \cdots U_{n,n},$$

where $U_{k,n}$, $k = 1, \dots, n$, is the upper triangular bidiagonal matrix with unit diagonal entries satisfying (9). Then $U_n = (u_{ij})_{1 \leq i,j \leq n+1}$ is an upper triangular matrix and

$$u_{ij} = \frac{1}{(i-1)!} m_{j-1}^{(i-1)}(x), \quad 1 \leq i, j \leq n+1. \tag{10}$$

Proof Clearly, U_n is an upper triangular matrix since it is the product of upper triangular bidiagonal matrices. Let us now prove (10) by induction on n . For $n = 1$,

$$U_1 = U_{1,1} = \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix}$$

and (10) clearly holds. Let us now suppose that (10) holds for $n \geq 1$. Then

$$U_{n+1} := U_{1,n+1} \cdots U_{n+1,n+1} = U_{1,n+1} \tilde{U}_{n+1},$$

where $\tilde{U}_{n+1} := U_{2,n+1} \cdots U_{n+1,n+1}$ satisfies $\tilde{U}_{n+1} = (\tilde{u}_{ij})_{1 \leq i,j \leq n+2}$ with $\tilde{u}_{i,1} = \tilde{u}_{1,i} = \delta_{1,i}$, that is, $\delta_{1,1} = 1$ and $\delta_{1,i} = 0$ for $i = 2, \dots, n+2$, and $\tilde{U}_{n+1}[2, \dots, n+2|2, \dots, n+2] = U_{1,n} \cdots U_{n,n}$. Then we have that

$$\tilde{u}_{ij} = \frac{1}{(i-2)!} m_{j-2}^{(i-2)}(x), \quad 2 \leq i, j \leq n+2.$$

Now taking into account that

$$U_{n+1} = U_{1,n+1} \tilde{U}_{n+1} = \begin{pmatrix} 1 & x & & \\ & \ddots & \ddots & \\ & & 1 & x \\ & & & 1 \end{pmatrix} \tilde{U}_{n+1},$$

and using Lemma 1, we deduce that $U_{n+1} = (u_{ij})_{1 \leq i,j \leq n+2}$ satisfies

$$\begin{aligned} u_{ij} &= \tilde{u}_{ij} + x\tilde{u}_{i+1,j} = \frac{1}{(i-2)!} m_{j-2}^{(i-2)}(x) + \frac{x}{(i-1)!} m_{j-2}^{(i-1)}(x) \\ &= \frac{1}{(i-1)!} m_{j-1}^{(i-1)}(x), \quad 1 \leq i, j \leq n+2. \end{aligned}$$

□

Let us observe that for $x > 0$ the matrices $U_{k,n}$, $k = 1, \dots, n$, are TP. Then, as a direct consequence of the previous result and taking into account that, by Theorem 3.1 of [1], the product of TP matrices is TP, we can derive the following result providing a bidiagonal factorization of the Wronskian matrix of the monomial basis (6).

Corollary 1 *Let $n \in \mathbb{N}$ and (m_0, \dots, m_n) be the monomial basis given in (6). Then for any $x \in \mathbb{R}$,*

$$W := W(m_0, \dots, m_n)(x) := \begin{pmatrix} 0! & & & \\ & 1! & & \\ & & \ddots & \\ & & & n! \end{pmatrix} U_{1,n} U_{2,n} \cdots U_{n,n}, \tag{11}$$

where $U_{k,n}$, $k = 1, \dots, n$, is the upper triangular bidiagonal matrix with unit diagonal entries satisfying (9). Moreover, if $x > 0$ then $W(m_0, \dots, m_n)(x)$ is TP.

Let us observe that (11) is the bidiagonal factorization (3) of the upper triangular, nonsingular and TP Wronskian matrix $W = W(m_0, \dots, m_n)(x)$, $x > 0$, where F_i and G_i are the TP, lower and upper triangular bidiagonal matrices in (4). Clearly $BD(W)$ can be computed with HRA and, consequently, using the bidiagonal factorization (5), W^{-1} can also be computed with HRA as stated in the following result.

Proposition 2 *Let W be the Wronskian matrix at x_0 of the monomial basis of the space of polynomials \mathbf{P}^n . Then W^{-1} can be computed with HRA.*

Furthermore, Sect. 5 will show accurate results obtained when computing the eigenvalues, singular values, the inverse and the solutions of some linear systems associated with the Wronskian matrices of monomial bases, using the bidiagonal factorization (11) and the algorithms presented in [10, 12].

Finally, in the following example, we illustrate the bidiagonal factorization (11) of the Wronskian matrix of a basis of monomials.

Example 1 For the particular case $n = 3$, the bidiagonal factorization of the Wronskian matrix of the basis (m_0, m_1, m_2, m_3) at $x \in \mathbb{R}$ is

$$W(m_0, m_1, m_2, m_3)(x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 6 \end{pmatrix} \begin{pmatrix} 1 & x & 0 & 0 \\ 0 & 1 & x & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & x & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

4 Bidiagonal factorization of the Wronskian matrix of a basis of exponential polynomials

Given $\lambda_0, \dots, \lambda_n$ and $x \in \mathbb{R}$, let us consider the basis (u_0, \dots, u_n) of exponential polynomials defined on \mathbb{R} by

$$u_i(x) := e^{\lambda_i x}, \quad i = 0, \dots, n. \quad (12)$$

The following result proves that, if $0 < \lambda_0 < \lambda_1 < \dots < \lambda_n$, the Wronskian matrix of the basis (12),

$$W(u_0, \dots, u_n)(x) = (\lambda_{j-1}^{i-1} e^{\lambda_{j-1} x})_{i,j=1, \dots, n+1}, \quad (13)$$

is STP for any $x \in \mathbb{R}$.

Theorem 1 *Let $0 < \lambda_0 < \dots < \lambda_n$ and the basis (12) of exponential polynomials. For any $x \in \mathbb{R}$, the corresponding Wronskian matrix (13) is STP and*

$$\det W(u_0, \dots, u_n)(x) = \prod_{k=0}^n e^{\lambda_k x} \prod_{0 \leq k < \ell \leq n} (\lambda_\ell - \lambda_k). \quad (14)$$

Proof The matrix $D := \text{diag}(e^{\lambda_0 x}, \dots, e^{\lambda_n x})$ is nonsingular and TP since $e^{\lambda_k x} > 0$, for all $k = 0, \dots, n$. It can be easily checked that

$$W(u_0, \dots, u_n)(x) = V_{n, \lambda_0, \dots, \lambda_n} D,$$

where $V_{n, \lambda_0, \dots, \lambda_n} := \left(\lambda_{j-1}^{i-1} \right)_{1 \leq i, j \leq n+1}$ is the $(n+1) \times (n+1)$ Vandermonde matrix corresponding to the values λ_i , $i = 0, \dots, n$. Using that $0 < \lambda_0 < \dots < \lambda_n$, we deduce that $V_{n, \lambda_0, \dots, \lambda_n}$ is STP (see [2]). Taking into account that, by Theorem 3.1 of [1], the product of a STP matrix by a nonsingular, TP matrix is a STP matrix, we conclude that $W(u_0, \dots, u_n)(x)$ is STP. Since $\det W(u_0, \dots, u_n)(x) = \det V_{n, \lambda_0, \dots, \lambda_n} \det D$ we can write

$$\det V_{n, \lambda_0, \dots, \lambda_n} = \prod_{0 \leq k < \ell \leq n} (\lambda_\ell - \lambda_k), \quad (15)$$

and deduce (14). \square

In the following result we present the bidiagonal decomposition (3) of the Wronskian matrices (13) and their inverses.

Theorem 2 *Let $0 < \lambda_0 < \dots < \lambda_n$ and the corresponding basis (12) of exponential polynomials. For a given $x \in \mathbb{R}$, $W := W(u_0, \dots, u_n)(x)$ admits a factorization of the form*

$$W = F_n F_{n-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n, \quad (16)$$

where F_i and G_i , $1 \leq i \leq n$, the lower and upper triangular bidiagonal matrices given by (4) and $D = \text{diag}(p_{1,1}, p_{2,2}, \dots, p_{n+1,n+1})$. The entries $m_{i,j}$, $\tilde{m}_{i,j}$ and $p_{i,i}$ are given by

$$m_{i,j} = \lambda_{j-1}, \quad \tilde{m}_{i,j} = e^{(\lambda_{i-1} - \lambda_{i-2})x} \prod_{k=2}^j \frac{(\lambda_{i-1} - \lambda_{i-k})}{(\lambda_{i-2} - \lambda_{i-k-1})}, \quad 1 \leq j < i \leq n + 1,$$

$$p_{i,i} = e^{\lambda_{i-1}x} \prod_{k=0}^{i-2} (\lambda_{i-1} - \lambda_k), \quad 1 \leq i \leq n + 1.$$

Proof By Theorem 1, the matrix W is STP and then the Neville elimination of W and W^T can be performed without row exchanges, leading to a factorization of type (3). The computation of the minors of W with initial consecutive columns and consecutive rows will allow us to determine the corresponding pivots $p_{i,j}$ and multipliers $m_{i,j}$.

Let $1 \leq j \leq i \leq n + 1$. The k th column of $M[i - j + 1, \dots, i | 1, \dots, j]$ has common factor $\lambda_{k-1}^{i-j} e^{\lambda_{k-1}x}$ and then

$$W[i - j + 1, \dots, i | 1, \dots, j] = V_{n, \lambda_0, \dots, \lambda_{j-1}}^T D,$$

where $D := \text{diag}(\lambda_0^{i-j} e^{\lambda_0x}, \dots, \lambda_{j-1}^{i-j} e^{\lambda_{j-1}x})$ and $V_{n, \lambda_0, \dots, \lambda_{j-1}}$ is the $j \times j$ Vandermonde matrix corresponding to parameters $\lambda_0, \dots, \lambda_{j-1}$. Using properties of determinants and (15), we can write

$$\det W[i - j + 1, \dots, i | 1, \dots, j] = \prod_{0 \leq k < \ell \leq j-1} (\lambda_\ell - \lambda_k) \prod_{k=0}^{j-1} \lambda_k^{i-j} e^{\lambda_kx}. \tag{17}$$

By (1) and (17), the pivot $p_{i,j}$ of the Neville elimination of W satisfies

$$p_{i,j} = \frac{\det W[i - j + 1, \dots, i | 1, \dots, j]}{\det W[i - j + 1, \dots, i - 1 | 1, \dots, j - 1]} = \lambda_{j-1}^{i-j} e^{\lambda_{j-1}x} \times \prod_{k=0}^{j-2} (\lambda_{j-1} - \lambda_k), \tag{18}$$

and, for the particular case $i = j$,

$$p_{i,i} = e^{\lambda_{i-1}x} \prod_{k=0}^{i-2} (\lambda_{i-1} - \lambda_k), \quad 1 \leq i \leq n + 1. \tag{19}$$

Finally, using (2) and (18), the multipliers $m_{i,j}$ can be obtained by

$$m_{i,j} = \frac{p_{i,j}}{p_{i-1,j}} = \lambda_{j-1}, \quad 1 \leq j < i \leq n + 1. \tag{20}$$

Now let us observe that each entry of the k th row of W^T has common factor $e^{\lambda_{i-j+k-1}x}$. Then we have that

$$W^T[i-j+1, \dots, i|1, \dots, j] = D_1 V_{n, \lambda_{i-j}, \dots, \lambda_{i-1}},$$

where $D_1 := \text{diag}(e^{\lambda_{i-j}x}, \dots, e^{\lambda_{i-1}x})$ and $V_{n, \lambda_{i-j}, \dots, \lambda_{i-1}}$ is the $j \times j$ Vandermonde matrix corresponding to parameters $\lambda_{i-j}, \dots, \lambda_{i-1}$. Using properties of determinants and (15), we can write

$$\det W^T[i-j+1, \dots, i|1, \dots, j] = \prod_{k=i-j}^{i-1} e^{\lambda_k x} \prod_{i-j \leq k < \ell \leq i-1} (\lambda_\ell - \lambda_k). \quad (21)$$

By (1) and (21), we deduce that

$$\begin{aligned} \tilde{p}_{i,j} &= \frac{\det W^T[i-j+1, \dots, i|1, \dots, j]}{\det W^T[i-j+1, \dots, i-1|1, \dots, j-1]} \\ &= e^{\lambda_{i-1}x} \prod_{k=i-j}^{i-2} (\lambda_{i-1} - \lambda_k). \end{aligned} \quad (22)$$

Finally, using (2) and (22), we have

$$\begin{aligned} \tilde{m}_{i,j} &= \frac{\tilde{p}_{i,j}}{\tilde{p}_{i-1,j}} = e^{(\lambda_{i-1} - \lambda_{i-2})x} \frac{\prod_{k=i-j}^{i-2} (\lambda_{i-1} - \lambda_k)}{\prod_{k=i-j-1}^{i-3} (\lambda_{i-2} - \lambda_k)} \\ &= e^{(\lambda_{i-1} - \lambda_{i-2})x} \prod_{k=2}^j \frac{(\lambda_{i-1} - \lambda_{i-k})}{(\lambda_{i-2} - \lambda_{i-k-1})}, \end{aligned} \quad (23)$$

for $1 \leq j < i \leq n+1$. □

Let us observe that the computation with HRA of the bidiagonal decomposition (16) should require the evaluation with HRA of the involved exponential function. Although this cannot be guaranteed, Sect. 5 will show accurate results obtained when computing their eigenvalues, singular values, inverses or the solutions of some linear systems associated with these Wronskian matrices of non-polynomial bases.

We finish this section illustrating the bidiagonal factorization (16) of the Wronskian matrix of a basis of exponential polynomials.

Example 2 For the particular case $n = 2$, the bidiagonal factorization of the Wronskian matrix of the basis $(e^{\lambda_0 x}, e^{\lambda_1 x}, e^{\lambda_2 x})$ at $x \in \mathbb{R}$ is

$$\begin{aligned} &W(e^{\lambda_0 x}, e^{\lambda_1 x}, e^{\lambda_2 x}) \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \lambda_0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \lambda_0 & 1 & 0 \\ 0 & \lambda_1 & 1 \end{pmatrix} \begin{pmatrix} p_{1,1} & 0 & 0 \\ 0 & p_{2,2} & 0 \\ 0 & 0 & p_{3,3} \end{pmatrix} \begin{pmatrix} 1 & e^{(\lambda_1 - \lambda_0)x} & 0 \\ 0 & 1 & e^{(\lambda_2 - \lambda_1)x} \frac{\lambda_2 - \lambda_1}{\lambda_1 - \lambda_0} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & e^{(\lambda_2 - \lambda_1)x} \\ 0 & 0 & 1 \end{pmatrix}, \end{aligned}$$

where $p_{1,1} = e^{\lambda_0 x}$, $p_{2,2} = e^{\lambda_1 x}(\lambda_1 - \lambda_0)$ and $p_{3,3} = e^{\lambda_2 x}(\lambda_2 - \lambda_0)(\lambda_2 - \lambda_1)$.

5 Numerical experiments

When the bidiagonal factorization of a nonsingular totally positive matrix is obtained with HRA, using the Matlab libraries “TNInverseExpand”, “TNEigenvalues”, “TNSingularValues” and “TNSolve”, available in [10], the computation of its inverse matrix, its eigenvalues and singular values or the solutions of some linear systems can be also performed with HRA.

We have implemented the Matlab functions “TNBDWM” and “TNBDWE” providing the bidiagonal decomposition (3) of the Wronkian matrix at x of the $(n + 1)$ -dimensional monomial and exponential basis. Now we include some numerical experiments illustrating the high accuracy obtained when using these functions and the previous libraries. Due to the ill conditioning of these matrices, traditional methods do not achieve accurate solutions when solving the mentioned algebraic problems. The numerical experiments show this fact and confirm the accuracy of the obtained results even though for some cases we cannot guarantee that the bidiagonal factorization (3) can be computed with HRA. The software with the numerical experiments will be provided by the authors upon request.

5.1 Linear systems

Let U be an $(n + 1)$ -dimensional space of n -times continuously differentiable functions defined on a real interval $I \subseteq \mathbb{R}$ and $x_0 \in I$. Given real values d_0, d_1, \dots, d_n , the corresponding Taylor interpolant in U is the function $u \in U$ such that $u^{(k)}(x_0) = d_k$, $k = 0, \dots, n$. Given a basis $\mathbf{u} = (u_0, \dots, u_n)$ of U , the Taylor interpolant can be expressed as $u(x) = \sum_{i=0}^n c_i u_i(x)$, $x \in I$, where $\mathbf{c} = (c_0, \dots, c_n)^T$ is the solution of the linear system

$$W\mathbf{c} = \mathbf{d}, \quad (24)$$

with $W = W(u_0, \dots, u_n)(x_0)$ and $\mathbf{d} = (d_0, \dots, d_n)^T$. Then we have $u(x) = \mathbf{u}(x)^T \mathbf{c}$ where $\mathbf{c} = W^{-1} \mathbf{d}$.

We have solved some linear systems (24) by considering the bases of the previous sections. We have obtained the solution of these systems using Mathematica with a precision of 100 digits and considered this solution exact. We have also computed with Matlab two approximations of this solution, the first one using “TNSolve” with the bidiagonal factorization proposed in this paper and the second one using the Matlab command `\`.

First, we have considered $x_0 = 50$ and the corresponding Wronskian matrices \mathbf{W}_n of the monomial basis $(1, x, \dots, x^n)$. Table 1 (left) illustrates the 2-norm condition number of these matrices using the Mathematica command `Norm[A,2]`. `Norm[Inverse[A],2]`. We have taken a vector $\mathbf{d}_n = ((-1)^{i+1} d_i)_{1 \leq i \leq n+1}$ where d_i is a random integer value. As we have mentioned in Sect. 3, the parameters of the bidiagonal decomposition (11) of \mathbf{W}_n can be obtained with HRA and so, the solution of $\mathbf{W}_n \mathbf{c}_n = \mathbf{d}_n$ can be performed with HRA. The numerical experiments confirm this fact and the greater accuracy of using the bidiagonal decomposition (11) (see Table 1).

Table 1 Condition number of Wronskian matrices of monomial bases at $x_0 = 50$ (left) and relative errors when solving $\mathbf{W}_n \mathbf{c}_n = \mathbf{d}_n$ with these matrices (middle and right)

$n + 1$	$\kappa_2(W_n)$	$W_n \setminus d_n$	“TNSolve”
10	1.1×10^{25}	3.8102×10^{-14}	8.8082×10^{-17}
15	4.8×10^{36}	6.6581×10^{-12}	1.7749×10^{-16}
20	3.7×10^{47}	5.0996×10^{-9}	1.1459×10^{-16}
25	8.2×10^{57}	2.7182×10^{-7}	2.8366×10^{-16}

Table 2 Condition number of Wronskian matrices of exponential bases at $x_0 = 1/2$ and $\lambda_i = i/(n + 2)$, $i = 1, \dots, n + 1$, (left) and relative errors when solving $\mathbf{W}_n \mathbf{c}_n = \mathbf{d}_n$ with these matrices (middle and right)

$n + 1$	$\kappa_2(W_n)$	$W_n \setminus \mathbf{d}_n$	“TNSolve”
10	9.6×10^7	4.0424×10^{-11}	5.4201×10^{-16}
15	2.8×10^{12}	2.7929×10^{-7}	9.3188×10^{-17}
20	8.2×10^{16}	4.7662×10^{-3}	3.8596×10^{-16}
25	2.5×10^{21}	1.4272	2.5409×10^{-15}

Now, for $x_0 = 1/2$, we have also considered Wronskian matrices \mathbf{W}_n of exponential polynomial bases with $\lambda_i = i/(n + 2)$, $i = 1, \dots, n + 1$. Table 2 (left) illustrates the 2-norm condition number of these matrices using the Mathematica command `Norm[A,2]·Norm[Inverse[A],2]`. We have also taken $\mathbf{d}_n = ((-1)^{i+1} d_i)_{1 \leq i \leq n+1}$, where d_i is a random integer value. The computation with HRA of the parameters of the bidiagonal factorization of \mathbf{W}_n cannot be guaranteed. However, these numerical experiments show again the high accuracy in the computations when using “TNSolve” with the bidiagonal factorization (16) (see Table 2).

5.2 Inverse matrix

In Section 4 of [15] the authors present the algorithm “TNInverseExpand”, which is an accurate and fast algorithm for computing the inverse of a nonsingular totally positive matrix A starting from $BD(A)$ and it has been included by P. Koev in his package TNTTool [10].

We have used the Matlab function “TNInverseExpand” with the factorization proposed in this paper in order to compute the inverse of Wronskian matrices of the bases considered in the paper. We have also computed their approximations with the Matlab function “inv”. In order to determine the accuracy of the approximations, we have calculated the inverse of these Wronskian matrices by using Mathematica with a precision of 100 digits and computed the relative errors corresponding to the approximations, considering the inverse matrix provided by Mathematica as exact.

The approximation of the inverse of the Wronskian matrices obtained by means of “TNInverseExpand” is very accurate for all considered n , providing much more accurate results than those obtained by Matlab using the command “inv”.

Table 3 Relative errors when computing the inverses of Wronskian matrices of monomial bases at $x_0 = 50$

$n + 1$	“inv”	“TNInverseExpand”
10	5.5583×10^{-14}	8.8081×10^{-17}
15	2.8550×10^{-11}	1.7749×10^{-16}
20	1.0218×10^{-9}	1.1497×10^{-16}
25	8.3974×10^{-7}	1.1944×10^{-16}

Table 4 Relative errors when computing the inverses of Wronskian matrices of exponential bases at $x_0 = 1/2$ and $\lambda_i = i/(n + 2)$, $i = 1, \dots, n + 1$

$n + 1$	“inv”	“TNInverseExpand”
10	4.0206×10^{-11}	4.0436×10^{-16}
15	2.8247×10^{-7}	3.5637×10^{-16}
20	4.8134×10^{-3}	4.0018×10^{-16}
25	1.4611	2.6557×10^{-15}

Tables 3 and 4 show the relative errors of the approximations to the inverse of the Wronskian matrices obtained with both methods.

5.3 Eigenvalues and singular values

We have also used the bidiagonal decomposition proposed in this paper with the Matlab functions “TNEigenValues” and “TNSingularValues”, to compute the eigenvalues and the singular values, respectively, of the previous Wronskian matrices. We have also computed their approximations with the Matlab functions “eig” and “svd”, respectively. In order to determine the accuracy of the approximations, we have calculated the eigenvalues and singular values of previous Wronskian matrices by using Mathematica with a precision of 100 digits and computed the relative errors corresponding to the approximations, considering the eigenvalues and singular values provided by Mathematica as exact.

Let us consider the Wronskian matrices at $x = 0.3$ of monomial bases. Table 5 (left) illustrates the 2-norm condition number of these matrices using the Mathematica command `Norm[A,2]·Norm[Inverse[A],2]`. Since these Wronskian matrices are all STP, by Theorem 6.2 of [1], all their eigenvalues are positive and distinct. Let us observe that the eigenvalues of these Wronskian matrices are $0!, \dots, n!$, so in this case the relative errors are 0 with both methods. On the other hand, the approximations of the singular values obtained by means of “TNSingularValues” are very accurate for all considered n , whereas the approximations of the singular values obtained with the Matlab command “svd” are not very accurate when n increases. Table 5 shows the relative errors of the approximations to the lowest singular value obtained with both methods.

Let us also consider Wronskian matrices of the exponential polynomial bases at $x = 1/2$ with $\lambda_i = i/(n + 2)$, $i = 1, \dots, n + 1$. The approximations of the eigenvalues and singular values obtained by means of the proposed factorization are

Table 5 Condition number of Wronskian matrices of monomial bases at $x_0 = 0.3$ (left) and relative errors when computing the lowest singular value of these matrices (middle and right)

$n + 1$	$\kappa_2(W_n)$	svd	“TNSingularValues”
10	4.5×10^5	1.5898×10^{-12}	3.9691×10^{-16}
15	1.1×10^{11}	7.2111×10^{-8}	2.6461×10^{-16}
20	1.5×10^{17}	2.4313×10^{-1}	6.6151×10^{-16}
25	7.7×10^{23}	7.4909×10^{-1}	2.6461×10^{-16}

Table 6 Relative errors when computing the lowest eigenvalue (left) and the lowest singular value (right) of Wronskian matrices of exponential bases at $x_0 = 1/2$ and $\lambda_i = i/(n + 2), i = 1, \dots, n + 1$

$n + 1$	eig	“TNEigenValues”	svd	“TNSingularValues”
10	1.8449×10^{-11}	3.1595×10^{-16}	1.7818×10^{-10}	1.5487×10^{-16}
15	1.8701×10^{-6}	7.9152×10^{-16}	3.0235×10^{-6}	1.1653×10^{-15}
20	1.1279×10^{-2}	1.1208×10^{-15}	7.0058×10^{-1}	8.6431×10^{-16}
25	1.4512×10^3	1.6727×10^{-15}	1.0646×10^2	2.4382×10^{-15}

very accurate for all considered n , whereas the approximations of the eigenvalues and singular values obtained with the Matlab commands “eig” and “svd” are not very accurate when n increases. Table 6 shows the relative errors of the approximations to the lowest eigenvalue and singular value obtained with both methods.

Acknowledgements This work was partially supported through the Spanish research grant PGC2018-096321-B-I00 (MCIU/AEI), by Gobierno de Aragón (E41_17R) and by Feder 2014–2020 “Construyendo Europa desde Aragón”.

References

1. Ando, T.: Totally positive matrices. *Linear Algebra Appl.* **90**, 165–219 (1987)
2. Carnicer, J.M., Peña, J.M.: Shape preserving representations and optimality of the Bernstein basis. *Adv. Comput. Math.* **1**, 173–196 (1993)
3. Delgado, J., Peña, J.M.: Accurate computations with collocation matrices of rational bases. *Appl. Math. Comput.* **219**, 4354–4364 (2013)
4. Delgado, J., Peña, J.M.: Accurate computations with collocation matrices of q-Bernstein polynomials. *SIAM J. Matrix Anal. Appl.* **36**, 880–893 (2015)
5. Demmel, J., Koev, P.: The accurate and efficient solution of a totally positive generalized Vandermonde linear system. *SIAM J. Matrix Anal. Appl.* **27**, 42–52 (2005)
6. Fallat, S.M., Johnson, C.R.: *Totally Nonnegative Matrices*. Princeton University Press, Princeton (2011). Princeton Series in Applied Mathematics
7. Gasca, M., Peña, J.M.: Total positivity and Neville elimination. *Linear Algebra Appl.* **165**, 25–44 (1992)
8. Gasca, M., Peña, J.M.: A matricial description of Neville elimination with applications to total positivity. *Linear Algebra Appl.* **202**, 33–53 (1994)

9. Gasca, M., Peña, J.M.: On factorizations of totally positive matrices. In: Gasca, M., Micchelli, C.A. (eds.) *Total Positivity and Its Applications*, pp. 109–130. Kluwer Academic Publishers, Dordrecht (1996)
10. Koev, P.: <http://math.mit.edu/plamen/software/TNTool.html>
11. Koev, P.: Accurate eigenvalues and SVDs of totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.* **27**, 1–23 (2005)
12. Koev, P.: Accurate computations with totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.* **29**, 731–751 (2007)
13. Marco, A., Martínez, J.J.: A fast and accurate algorithm for solving Bernstein–Vandermonde linear systems. *Linear Algebra Appl.* **422**, 616–628 (2007)
14. Marco, A., Martínez, J.J.: Accurate computations with Said–Ball–Vandermonde matrices. *Linear Algebra Appl.* **432**, 2894–2908 (2010)
15. Marco, A., Martínez, J.J.: Accurate computation of the Moore–Penrose inverse of strictly totally positive matrices. *J. Comput. Appl. Math.* **350**, 299–308 (2019)
16. Pinkus, A.: *Totally Positive Matrices*, Cambridge Tracts in Mathematics, 181. Cambridge University Press, Cambridge (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

ARTICLE 5

- [76] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with collocations and Wronskian matrices of Jacoby polynomials, *Journal of Scientific Computing* **87**, 77 (2021).



Accurate Computations with Collocation and Wronskian Matrices of Jacobi Polynomials

E. Mainar¹ · J. M. Peña¹ · B. Rubio¹

Received: 14 December 2020 / Revised: 26 March 2021 / Accepted: 17 April 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

In this paper an accurate method to construct the bidiagonal factorization of collocation and Wronskian matrices of Jacobi polynomials is obtained and used to compute with high relative accuracy their eigenvalues, singular values and inverses. The particular cases of collocation and Wronskian matrices of Legendre polynomials, Gegenbauer polynomials, Chebyshev polynomials of the first and second kind and rational Jacobi polynomials are considered. Numerical examples are included.

Keywords High relative accuracy · Bidiagonal decompositions · Jacobi polynomials · Totally positive matrices

1 Introduction

Jacobi polynomials $J_n^{(\alpha, \beta)}(x)$ (see Section 3) form a class of classical orthogonal polynomials, which includes many important families of orthogonal polynomials such as Legendre and Chebyshev polynomials (see Section 5). In fact, Jacobi polynomials are orthogonal with respect to the weight $(1-x)^\alpha(1+x)^\beta$ on the interval $[-1, 1]$ and present many useful applications. For instance, to approximation theory, to Gaussian quadrature to numerically compute integrals, to differential equations or to physical applications (cf. [2, 13]).

Let us recall that, given a system of functions (u_0, \dots, u_n) , its *collocation matrix* at points $x_1 < \dots < x_{n+1}$ is given by $(u_{j-1}(x_i))_{1 \leq i, j \leq n+1}$. This paper deals with the accurate computation when using collocation and Wronskian matrices (see Section 3) of Jacobi polynomials on $(1, \infty)$. As shown in this paper, for these matrices many algebraic computations (such as

This work was partially supported through the Spanish research grant PGC2018-096321-B-I00 (MCIU/AEI), by Gobierno de Aragón (E41_17R) and by Feder 2014-2020 “Construyendo Europa desde Aragón”.

B. Rubio
brubio@unizar.com

E. Mainar
esmemain@unizar.com

J. M. Peña
jmpena@unizar.com

¹ Departamento de Matemática Aplicada/IUMA, Universidad de Zaragoza, Zaragoza, Spain

the computation of the inverse, of all the eigenvalues and singular values, or the solutions of some linear systems) can be performed with high relative accuracy (HRA, see Section 2). Up to now, this has been obtained only for a few classes of structured matrices. Among them we can mention the collocation matrices of Bernstein polynomials [16], of Laguerre polynomials [3] and of Bessel functions [4] as well as the Wronskian matrices of the monomials and of exponential polynomials [15]. In fact, this last paper was the unique paper guaranteeing HRA for some Wronskian matrices.

Crucial facts to derive our results have been to prove the strict total positivity (see Section 2) of the collocation matrices of Jacobi polynomials on $(1, \infty)$ and the total positivity of their Wronskian matrices. Then the bidiagonal factorization with HRA has been obtained for these matrices and the algorithms presented in [12] can be used for the algebraic computations mentioned above with HRA.

As mentioned before, accurate computations with collocation matrices of other interesting bases of orthogonal polynomials, such as Laguerre polynomials or Bessel polynomials, have been already achieved (see [3] and [4]). The analysis of the domain where the corresponding collocation or Wronskian matrices, or closely related matrices, are totally positive helps to obtain their bidiagonal factorization and the solution of algebraic problems with HRA for the parameters in this domain. We shall see that for the collocation or Wronskian matrices of Jacobi bases, this domain lies outside the interval where the polynomials are orthogonal and have their zeros.

The paper is organized as follows. Section 2 presents some basic concepts and results related to the bidiagonal factorization of totally positive matrices and with HRA. In Section 3, the strict total positivity and bidiagonal factorization of the collocation matrices of Jacobi polynomials on $(1, \infty)$ are obtained. In Section 4, the total positivity and bidiagonal factorization of the corresponding Wronskian matrices are derived. Section 5 particularizes the results for some well known families of Jacobi polynomials: Legendre polynomials, Gegenbauer polynomials, Chebyshev polynomials of the first and second kind and rational Jacobi polynomials. Section 6 presents numerical examples confirming the theoretical results for the computation of eigenvalues, singular values, inverses, and the solution of linear systems with some matrices used in this paper.

2 Notations and Auxiliary Results

As usual, given an n -times continuously differentiable function f and x in its parameter domain, $f'(x)$ denotes the first derivative of f at x and, for any $i \leq n$, $f^{(i)}(x)$ denotes the i -th derivative of f at x . Let us recall that for a given basis (u_0, \dots, u_n) of a space of n -times continuously differentiable functions, defined on a real interval I and $x \in I$, the *Wronskian matrix* at x is defined by

$$W(u_0, \dots, u_n)(x) := (u_{j-1}^{(i-1)}(x))_{i,j=1,\dots,n+1}.$$

A matrix is totally positive: TP (respectively, strictly totally positive: STP) if all its minors are nonnegative (respectively, positive). Two recent books on these matrices are [6,18], where many applications of these matrices are presented, as well as in [1].

Neville elimination is an alternative procedure to Gaussian elimination and has been used to characterize TP and STP matrices. Given a nonsingular matrix $A = (a_{i,j})_{1 \leq i,j \leq n+1}$, Neville elimination computes a matricial sequence

$$A^{(1)} := A \rightarrow A^{(2)} \rightarrow \dots \rightarrow A^{(n+1)}, \quad (1)$$

$$G_i^T = \begin{pmatrix} 1 & & & & & & \\ 0 & 1 & & & & & \\ & & \ddots & \ddots & & & \\ & & & 0 & 1 & & \\ & & & \tilde{m}_{i+1,1} & 1 & & \\ & & & & \tilde{m}_{i+2,2} & 1 & \\ & & & & & \ddots & \ddots & \\ & & & & & & \tilde{m}_{n+1,n+1-i} & 1 \end{pmatrix}, \quad (8)$$

and $D = \text{diag}(p_{1,1}, p_{2,2}, \dots, p_{n+1,n+1})$ has positive diagonal entries. The diagonal entries $p_{i,i}$ of D are the diagonal pivots of the Neville elimination of A and the elements $m_{i,j}$ and $\tilde{m}_{i,j}$ are nonnegative and coincide with the multipliers of the Neville elimination of A and A^T , respectively. If, in addition, the entries m_{ij}, \tilde{m}_{ij} satisfy

$$m_{ij} = 0 \quad \Rightarrow \quad m_{hj} = 0, \quad \forall h > i$$

and

$$\tilde{m}_{ij} = 0 \quad \Rightarrow \quad \tilde{m}_{ik} = 0, \quad \forall k > j,$$

then the decomposition (7) is unique. We shall denote the bidiagonal decomposition (7) of a TP matrix A as $BD(A)$ (see [11]).

Given $BD(A)$, using the results in [7–9], a bidiagonal decomposition of A^{-1} can be computed as

$$A^{-1} = \tilde{G}_1 \tilde{G}_2 \cdots \tilde{G}_n D^{-1} \tilde{F}_n \cdots \tilde{F}_2 \tilde{F}_1, \quad (9)$$

where \tilde{F}_i and \tilde{G}_i , $i = 1, \dots, n$, are the lower and upper triangular bidiagonal matrices of the form of F_i and G_i , respectively, but replacing the off-diagonal entries $\{m_{i+1,1}, \dots, m_{n+1,n+1-i}\}$ and $\{\tilde{m}_{i+1,1}, \dots, \tilde{m}_{n+1,n+1-i}\}$ by $\{-m_{i+1,i}, \dots, -m_{n+1,i}\}$ and $\{-\tilde{m}_{i+1,i}, \dots, -\tilde{m}_{n+1,i}\}$, respectively.

Let us observe that if A is a nonsingular and TP matrix, then A^T is also nonsingular and TP. Moreover, the bidiagonal decomposition of A^T can be computed as

$$A^T = G_n^T G_{n-1}^T \cdots G_1^T D F_1^T \cdots F_{n-1}^T F_n^T, \quad (10)$$

where F_i and G_i , $i = 1, \dots, n$, are the lower and upper triangular bidiagonal matrices given in the bidiagonal factorization $BD(A)$, that is,

$$BD(A^T) = BD(A)^T.$$

Finally, let us recall that a real value x is obtained with high relative accuracy (HRA) if the relative error of the computed value \tilde{x} satisfies

$$\frac{\|x - \tilde{x}\|}{\|x\|} < Ku,$$

where K is a positive constant independent of the arithmetic precision and u is the unit round-off. HRA implies that the relative errors of the computations are of the order of the machine precision. An algorithm can be computed with HRA when it only uses products, quotients, sums of numbers of the same sign, subtractions of numbers of opposite sign or subtraction of initial data (cf. [5,10]).

In [11] it was shown that if $BD(A)$, the bidiagonal factorization (7) of a nonsingular TP matrix A , is computed with HRA then we can also compute with HRA its eigenvalues and

singular values, the matrix A^{-1} and even the solution of $Ax = b$ for vectors b with alternating signs.

3 Total Positivity and Factorizations of Collocation Matrices of Jacobi Polynomials

Given $\alpha, \beta \in \mathbb{R}$, the basis of Jacobi polynomials of the space \mathbf{P}^n of polynomials of degree less than or equal to n is $(J_0^{(\alpha,\beta)}, \dots, J_n^{(\alpha,\beta)})$ with

$$J_i^{(\alpha,\beta)}(x) := \frac{\Gamma(\alpha + i + 1)}{i! \Gamma(\alpha + \beta + i + 1)} \sum_{k=0}^i \binom{i}{k} \frac{\Gamma(\alpha + \beta + i + k + 1)}{\Gamma(\alpha + k + 1)} \left(\frac{x - 1}{2}\right)^k, \quad i = 0, \dots, n. \tag{11}$$

Let us recall that Jacobi polynomials are orthogonal on the interval $[-1, 1]$ with respect to the weight $(1 - x)^\alpha(1 + x)^\beta$.

Let us consider the lower triangular matrix $A = (a_{ij})_{1 \leq i, j \leq n+1}$ given by

$$a_{i,j} := \begin{cases} \frac{1}{(j-1)!(i-j)!} \prod_{k=j}^{i-1} (\alpha + k) \prod_{k=1}^{j-1} (\alpha + \beta + i + k - 1), & \text{if } i \geq j, \\ 0, & \text{if } i < j. \end{cases} \tag{12}$$

It can be checked that

$$(J_0^{(\alpha,\beta)}, \dots, J_n^{(\alpha,\beta)})^T = A(v_0, \dots, v_n)^T, \tag{13}$$

where (v_0, \dots, v_n) is the basis of \mathbf{P}^n such that

$$v_i(x) := \left(\frac{x - 1}{2}\right)^i, \quad i = 0, \dots, n. \tag{14}$$

The following result provides the multipliers and the diagonal pivots of the Neville elimination of the change of basis matrix A described in (12) and proves that this matrix is nonsingular and TP.

Theorem 2 *Let $A = (a_{ij})_{1 \leq i, j \leq n+1}$ be the lower triangular matrix defined in (12). Then the multipliers $m_{i,j}$ and diagonal pivots $p_{i,i}$ of the Neville elimination of A are given by*

$$m_{i,1} := \frac{\alpha + i - 1}{i - 1}, \quad m_{i,j} := \frac{\alpha + \beta + 2i - j}{\alpha + \beta + 2i - j - 2} m_{i,j-1}, \quad 1 < j < i \leq n + 1, \\ 1 < i \leq n + 1, \\ p_{i,i} := \prod_{r=1}^{i-1} \frac{(\alpha + \beta + 2i - r - 1)}{(i - r)}, \quad 1 \leq i \leq n + 1. \tag{15}$$

Moreover, for any $\alpha, \beta > -1$, A is nonsingular and TP.

Proof Let $A^{(k)} := (a_{ij}^{(k)})_{1 \leq i, j \leq n+1}$, $k = 2, \dots, n + 1$, be the matrices obtained after $k - 1$ steps of the Neville elimination of A . First, let us see by induction on k that

$$a_{i,j}^{(k)} = \frac{1}{(j - k)!(i - j)!} \prod_{r=1}^{k-1} \frac{(\alpha + \beta + 2i - r - 1)}{(i - r)}$$

$$\prod_{r=j}^{i-1} (\alpha + r) \prod_{r=1}^{j-k} (\alpha + \beta + i + r - 1), \quad (16)$$

for $1 \leq j < i \leq n + 1$. For $k = 2$, taking into account that $a_{i,j}^{(2)} = a_{i,j} - \frac{a_{i,1}}{a_{i-1,1}} a_{i-1,j}$, we have

$$\begin{aligned} a_{i,j}^{(2)} &= a_{i,j} - \frac{\alpha + i - 1}{i - 1} a_{i-1,j} \\ &= \frac{1}{(j-1)!(i-j-1)!} \prod_{r=j}^{i-1} (\alpha + r) \\ &\quad \prod_{r=1}^{j-2} (\alpha + \beta + i + r - 1) \left(\frac{\alpha + \beta + i + j - 2}{i - j} - \frac{\alpha + \beta + i - 1}{i - 1} \right) \\ &= \frac{1}{(j-1)!(i-j-1)!} \prod_{r=j}^{i-1} (\alpha + r) \\ &\quad \prod_{r=1}^{j-2} (\alpha + \beta + i + r - 1) \left(\frac{(j-1)(\alpha + \beta + 2i - 2)}{(i-j)(i-1)} \right) \\ &= \frac{1}{(j-2)!(i-j)!} \frac{(\alpha + \beta + 2i - 2)}{(i-1)} \prod_{r=j}^{i-1} (\alpha + r) \\ &\quad \prod_{r=1}^{j-2} (\alpha + \beta + i + r - 1), \quad 1 \leq j < i \leq n + 1. \end{aligned}$$

Therefore formula (16) holds for $k = 2$. Let us now suppose that (16) holds for some $k \in \{2, \dots, n\}$. Taking into account that $a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - \frac{a_{i,k}^{(k)}}{a_{i-1,k}^{(k)}} a_{i-1,j}^{(k)}$, we have

$$a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - \frac{1}{i-1} \prod_{r=1}^{k-1} \frac{(\alpha + \beta + 2i - r - 1)}{(\alpha + \beta + 2i - r - 3)} (\alpha + i - 1) a_{i-1,j}^{(k)}.$$

Then, by defining

$$C_1 := \frac{\alpha + \beta + i + j - k - 1}{i - j} - \frac{\alpha + \beta + i - 1}{i - k} = \frac{(j - k)(\alpha + \beta + 2i - k - 1)}{(i - j)(i - k)},$$

we can write

$$\begin{aligned} a_{i,j}^{(k+1)} &= \frac{1}{(j-k)!(i-j-1)!} \prod_{r=1}^{k-1} \frac{(\alpha + \beta + 2i - r - 1)}{(i - r)} \prod_{r=j}^{i-1} (\alpha + r) \\ &\quad \prod_{r=1}^{j-k-1} (\alpha + \beta + i + r - 1) C_1 \\ &= \frac{1}{(j-k-1)!(i-j)!} \prod_{r=1}^k \frac{(\alpha + \beta + 2i - r - 1)}{(i - r)} \end{aligned}$$

$$\prod_{r=j}^{i-1} (\alpha + r) \prod_{r=1}^{j-k-1} (\alpha + \beta + i + r - 1),$$

and formula (16) also holds for $k + 1$.

Now, by (4) and (16), we can easily deduce that the pivots $p_{i,j}$ of the Neville elimination of A satisfy

$$p_{i,j} = \frac{1}{(i-j)!} \prod_{r=1}^{j-1} \frac{(\alpha + \beta + 2i - r - 1)}{(i-r)} \prod_{r=j}^{i-1} (\alpha + r), \quad 1 \leq j < i \leq n + 1, \tag{17}$$

and, for the particular case $i = j$,

$$p_{i,i} = \prod_{r=1}^{i-1} \frac{(\alpha + \beta + 2i - r - 1)}{(i-r)}, \quad 1 \leq i \leq n + 1. \tag{18}$$

Let us observe that, by formula (17), the pivots of the Neville elimination of A are nonzero and so, this elimination can be performed without row exchanges. Besides, since A is lower triangular with nonzero diagonal entries, A is nonsingular and the obtained matrix U (see (3)) is diagonal and so, the Neville elimination of U^T does not perform any operation. Then, by Theorem 1, we can conclude that A is nonsingular and TP for any $\alpha, \beta > -1$.

Finally, using (6) and (18), the multipliers $m_{i,j}$ can be written as

$$m_{i,j} = \frac{(i-1-j)! (\alpha + i - 1)}{(i-j)!} \prod_{r=1}^{j-1} \frac{(\alpha + \beta + 2i - r - 1)(i-r-1)}{(\alpha + \beta + 2i - r - 3)(i-r)},$$

and we can deduce that

$$\begin{aligned} m_{i,1} &= \frac{\alpha + i - 1}{i - 1}, \quad 1 < i \leq n + 1, \\ m_{i,j} &= \frac{\alpha + \beta + 2i - j}{\alpha + \beta + 2i - j - 2} m_{i,j-1}, \quad 1 < j < i \leq n + 1. \end{aligned} \tag{19}$$

□

Corollary 1 *Let $A = (a_{ij})_{1 \leq i, j \leq n+1}$ be the lower triangular matrix defined by (12). Then, for any $\alpha, \beta > -1$, the matrix A admits a factorization of the form*

$$A = F_n F_{n-1} \cdots F_1 D, \tag{20}$$

where $F_i, i = 1, \dots, n$, is the lower triangular, bidiagonal matrix given by (8) and $D = \text{diag}(p_{1,1}, p_{2,2}, \dots, p_{n+1,n+1})$. The entries $m_{i,j}$ and $p_{i,i}$ can be obtained from (15).

Let us observe that the factorization (20) corresponds to $BD(A)$, the bidiagonal factorization (7) of A . Furthermore, for any $\alpha, \beta > -1$, $BD(A)$ can be computed with HRA, since it does not require subtractions (except of the initial data).

Remark 1 It is well known that the monomial basis $(1, t, \dots, t^n)$ of \mathbf{P}^n is STP on $(0, \infty)$. Moreover, given a sequence of positive parameters $0 < t_0 < \dots < t_n$, the bidiagonal factorization (7) of the corresponding STP collocation matrix can be described by

$$m_{i,j} = \frac{\prod_{k=1}^{j-1} (t_i - t_{i-k})}{\prod_{k=2}^j (t_{i-1} - t_{i-k})}, \quad \widehat{m}_{i,j} = t_j, \quad 1 \leq j < i \leq n + 1,$$

$$p_{i,i} = \prod_{k=1}^{i-1} (t_i - t_k), \quad 1 \leq i \leq n+1 \quad (21)$$

(see [10] or Theorem 3 of [14]). Consequently, the basis (v_0, \dots, v_n) defined in (14) is also STP on $(1, \infty)$. Furthermore, given $1 < x_1 < \dots < x_{n+1}$, by considering $t_i := (x_i - 1)/2$, $i = 1, \dots, n+1$, and using the bidiagonal factorization (21) for the collocation matrix of the monomial basis at $0 < t_1 < \dots < t_{n+1}$, it can be easily deduced that the bidiagonal decomposition (7) of the collocation matrix of (v_0, \dots, v_n) at $x_1 < \dots < x_{n+1}$ is given by:

$$\begin{aligned} m_{i,j} &= \frac{\prod_{k=1}^{j-1} (x_i - x_{i-k})}{\prod_{k=2}^j (x_{i-1} - x_{i-k})}, \quad \widehat{m}_{i,j} = (x_j - 1)/2, \quad 1 \leq j < i \leq n+1, \\ p_{i,i} &= \frac{1}{2^{i-1}} \prod_{k=1}^{i-1} (x_i - x_k), \quad 1 \leq i \leq n+1. \end{aligned} \quad (22)$$

The following result proves that, for any $\alpha, \beta > -1$, the collocation matrix of the basis (11) of Jacobi polynomials at $1 < x_1 < \dots < x_{n+1}$,

$$M_J := \left(J_{j-1}^{(\alpha, \beta)}(x_i) \right)_{1 \leq i, j \leq n+1}, \quad (23)$$

is STP.

Theorem 3 *Given $\alpha, \beta > -1$, the corresponding basis of Jacobi polynomials defined in (11) is STP on $(1, \infty)$.*

Proof Given a sequence of parameters $1 < x_1 < \dots < x_{n+1}$, by formula (13), the collocation matrix (23) of the Jacobi polynomial basis satisfies

$$M_J = M A^T, \quad (24)$$

where M is the collocation matrix at $1 < x_1 < \dots < x_{n+1}$ of the basis (v_0, \dots, v_n) defined in (14) and A is the lower triangular matrix defined by (12).

Clearly, by Remark 1, M is a STP matrix. On the other hand, by Theorem 2, given $\alpha, \beta > -1$, the lower triangular matrix A defined by (12) is nonsingular and TP. So, A^T is also a nonsingular and TP matrix. As a direct consequence of these facts and taking into account that, by Theorem 3.1 of [1], the product of a STP matrix and a nonsingular TP matrix is a STP matrix, we can conclude that the collocation matrix (23) is STP. \square

Remark 2 By Section 4 of [10], we can transpose the bidiagonal decomposition (20) of the lower triangular and TP matrix A to obtain the corresponding bidiagonal decomposition of A^T (see (10)). Clearly, since $BD(A)$ can be computed with HRA, $BD(A^T)$ can be also computed with HRA. Moreover, the collocation matrix of the basis (v_0, \dots, v_n) defined in (14) at nodes $1 < x_1 < \dots < x_{n+1}$ is STP and its corresponding bidiagonal decomposition can be obtained with HRA (see (22)). If the bidiagonal decompositions of two nonsingular, TP matrices can be computed with HRA, using Algorithm 5.1 of [11], we can also obtain with HRA the bidiagonal decomposition of the nonsingular and TP product matrix. Consequently, we can derive with HRA the bidiagonal matrices (8) of the bidiagonal factorization (7) of the collocation matrices of Jacobi polynomials and thus, we can also compute with HRA its inverse matrix, its eigenvalues and singular values as well as the solutions of some linear systems.

In Section 6, Algorithm 2 provides the bidiagonal decomposition of the collocation matrix (23) of the basis of Jacobi polynomials. Moreover, Section 6 illustrates accurate results obtained when computing algebraic problems using this algorithm and the algorithms presented in [11] and [12].

4 Total Positivity and Factorizations of Wronskian Matrices of Jacobi Polynomials

Given $x \in \mathbb{R}$, let $W(J_0^{(\alpha,\beta)}, \dots, J_n^{(\alpha,\beta)})(x)$ be the Wronskian matrix at x of the basis (11) of Jacobi polynomials. Using formula (13), it can be checked that

$$W(J_0^{(\alpha,\beta)}, \dots, J_n^{(\alpha,\beta)})(x) = W(v_0, \dots, v_n)(x)A^T, \quad (25)$$

where $W(v_0, \dots, v_n)(x)$ is the Wronskian matrix of the basis (v_0, \dots, v_n) given in (14) and A is the lower triangular matrix defined by (12).

In Corollary 1 of [15] it was proved that the Wronskian matrix at any positive real value of the monomial basis $(1, x, \dots, x^n)$ of the space of polynomials \mathbf{P}^n is TP on $(0, \infty)$. It was also shown that this Wronskian matrix and its inverse can be computed with HRA. Now we are going to extend these results to the basis (ℓ_0, \dots, ℓ_n) given by

$$\ell_i(x) = (ax + b)^i, \quad x \in \mathbb{R}, \quad i = 0, \dots, n, \quad (26)$$

where $a, b \in \mathbb{R}$ with $a > 0$. First let us prove the following auxiliary result.

Lemma 1 *The basis (ℓ_0, \dots, ℓ_n) defined in (26) satisfies*

$$\frac{1}{a^i i!} \ell_j^{(i)}(x) = \frac{1}{a^{i-1} (i-1)!} \ell_{j-1}^{(i-1)}(x) + \frac{ax+b}{a^i i!} \ell_{j-1}^{(i)}(x), \quad 1 \leq i, j \leq n. \quad (27)$$

Proof We prove the result by induction on i . Since $\ell_j(x) = (ax + b)\ell_{j-1}(x)$, we have

$$\ell_j'(x) = a\ell_{j-1}(x) + (ax + b)\ell_{j-1}'(x), \quad x \in \mathbb{R},$$

and so, formula (27) holds for $i = 1$ and $1 \leq j \leq n$. If (27) holds for $i > 1$, we can write

$$\frac{1}{a^i i!} \ell_j^{(i+1)}(x) = \frac{a(i+1)}{a^i i!} \ell_{j-1}^{(i)}(x) + \frac{ax+b}{a^i i!} \ell_{j-1}^{(i+1)}(x),$$

and deduce that

$$\frac{1}{a^{i+1} (i+1)!} \ell_j^{(i+1)}(x) = \frac{1}{a^i i!} \ell_{j-1}^{(i)}(x) + \frac{ax+b}{a^{i+1} (i+1)!} \ell_{j-1}^{(i+1)}(x).$$

□

Now, for a given $x \in \mathbb{R}$, $k, n \in \mathbb{N}$ with $k \leq n$, let $U_{k,n} = (u_{i,j}^{(k)})_{1 \leq i, j \leq n+1}$ be the upper triangular, bidiagonal matrix with unit diagonal entries, such that

$$u_{i,i+1}^{(k)} := 0, \quad i = 1, \dots, k-1, \quad u_{i,i+1}^{(k)} := ax + b, \quad i = k, \dots, n. \quad (28)$$

The following result shows that the product matrix $U_{1,n} \cdots U_{n,n}$ coincides, up to a positive scaling, with the Wronskian matrix of $(\ell_0, \ell_1, \dots, \ell_n)$ at x .

Proposition 1 For a given $x \in \mathbb{R}$ and $n \in \mathbb{N}$, let

$$U_n := U_{1,n} \cdots U_{n,n},$$

where $U_{k,n}$, $k = 1, \dots, n$, are the upper triangular, bidiagonal matrices with unit diagonal entries satisfying (28). Then $U_n = (u_{i,j})_{1 \leq i, j \leq n+1}$ is an upper triangular matrix and

$$u_{i,j} = \frac{1}{a^{i-1}(i-1)!} \ell_{j-1}^{(i-1)}(x), \quad 1 \leq i, j \leq n+1. \quad (29)$$

Proof First, let us observe that U_n is the product of upper triangular, bidiagonal matrices and so, it is an upper triangular matrix. Now, we prove (29) by induction on n . For $n = 1$,

$$U_1 = U_{1,1} = \begin{pmatrix} 1 & ax + b \\ 0 & 1 \end{pmatrix}$$

and (29) clearly holds. Let us observe that

$$U_{n+1} := U_{1,n+1} \cdots U_{n+1,n+1} = U_{1,n+1} \tilde{U}_{n+1},$$

where $\tilde{U}_{n+1} := U_{2,n+1} \cdots U_{n+1,n+1}$ satisfies $\tilde{U}_{n+1} = (\tilde{u}_{i,j})_{1 \leq i, j \leq n+2}$ with $\tilde{u}_{i,1} = \tilde{u}_{1,i} = \delta_{1,i}$, that is, $\delta_{1,1} = 1$ and $\delta_{1,i} = 0$ for $i = 2, \dots, n+2$, and $\tilde{U}_{n+1}[2, \dots, n+2][2, \dots, n+2] = U_{1,n} \cdots U_{n,n}$. Let us now suppose that (29) holds for $n \geq 1$. Then we have that

$$\tilde{u}_{i,j} = \frac{1}{a^{i-2}(i-2)!} \ell_{j-2}^{(i-2)}(x), \quad 2 \leq i, j \leq n+2.$$

Taking into account that $U_{n+1} = U_{1,n+1} \tilde{U}_{n+1}$ and using Lemma 1, we deduce that $U_{n+1} = (u_{i,j})_{1 \leq i, j \leq n+2}$ satisfies

$$\begin{aligned} u_{i,j} &= \tilde{u}_{i,j} + (ax + b)\tilde{u}_{i+1,j} = \frac{1}{a^{i-2}(i-2)!} \ell_{j-2}^{(i-2)}(x) + \frac{ax + b}{a^{i-1}(i-1)!} \ell_{j-2}^{(i-1)}(x) \\ &= \frac{1}{a^{i-1}(i-1)!} \ell_{j-1}^{(i-1)}(x), \quad 1 \leq i, j \leq n+2. \end{aligned}$$

□

As a direct consequence of the previous result, we can provide the bidiagonal factorization (7) of the Wronskian matrix of (ℓ_0, \dots, ℓ_n) .

Proposition 2 Let $n \in \mathbb{N}$ and (ℓ_0, \dots, ℓ_n) be the basis given in (26). Then, for any $x > -b/a$, the Wronskian matrix $W(\ell_0, \dots, \ell_n)(x)$ is TP and

$$W(\ell_0, \dots, \ell_n)(x) = \begin{pmatrix} 0! & & & \\ & a^1 1! & & \\ & & \ddots & \\ & & & a^n n! \end{pmatrix} U_{1,n} \cdots U_{n,n}, \quad (30)$$

where $U_{k,n}$, $k = 1, \dots, n$, are the upper triangular, bidiagonal matrices with unit diagonal entries satisfying (28).

Let us observe that the bidiagonal factorization (7) of $W(\ell_0, \dots, \ell_n)(x)$ is given by (30). Clearly, this factorization can be computed with HRA for any $x > -b/a$ and, consequently, using (9), its inverse matrix can also be computed with HRA as stated in the following result.

Proposition 3 *Let W be the Wronskian matrix at $x > -b/a$ of the basis (ℓ_0, \dots, ℓ_n) given in (26). Then W^{-1} can be computed with HRA.*

Now, using Proposition 2, we can immediately deduce the following factorization of the Wronskian matrix at $x \in \mathbb{R}$ of the basis (v_0, \dots, v_n) in (14),

$$W(v_0, \dots, v_n)(x) := \begin{pmatrix} \frac{1}{2^0} 0! & & & \\ & \frac{1}{2^1} 1! & & \\ & & \ddots & \\ & & & \frac{1}{2^n} n! \end{pmatrix} U_{1,n} \cdots U_{n,n}, \quad (31)$$

where $U_{k,n} = (u_{i,j}^{(k)})_{1 \leq i, j \leq n+1}$, $k = 1, \dots, n$, is the upper triangular, bidiagonal matrix with unit diagonal entries satisfying

$$u_{i,i+1}^{(k)} := 0, \quad i = 1, \dots, k-1, \quad u_{i,i+1}^{(k)} := (x-1)/2, \quad i = k, \dots, n. \quad (32)$$

Moreover, if $x > 1$, $W(v_0, \dots, v_n)(x)$ is a nonsingular and TP matrix. Then, taking into account (25), the fact that A^T is a nonsingular and TP matrix (see Theorem 2) and that the product of nonsingular TP matrices is a nonsingular and TP matrix (Theorem 3.1 of [1]), we deduce the following result on the total positivity of the Wronskian matrices of Jacobi polynomials.

Theorem 4 *Let $n \in \mathbb{N}$ and $(J_0^{(\alpha,\beta)}, \dots, J_n^{(\alpha,\beta)})$ be the Jacobi polynomial basis given in (11). For any $\alpha, \beta > -1$, the Wronskian matrix $W(J_0^{(\alpha,\beta)}, \dots, J_n^{(\alpha,\beta)})(x)$ at $x > 1$ is nonsingular and TP.*

Remark 3 Taking into account (10), we can obtain the bidiagonal decomposition (20) of the matrix A^T in (25). Clearly, since $BD(A)$ can be computed with HRA, $BD(A^T)$ can be also computed with HRA. On the other hand, the Wronskian matrix of the basis (v_0, \dots, v_n) defined in (14) is nonsingular and TP at any $x > 1$. Moreover, its corresponding bidiagonal decomposition (22) can be obtained with HRA. By Algorithm 5.1 of [11], if the bidiagonal decompositions of two nonsingular and TP matrices can be computed with HRA, then the bidiagonal decomposition of the product matrix can be also obtained with HRA. Consequently, the Wronskian matrix of the basis (11) of Jacobi polynomials can be computed with HRA and thus, we can compute with HRA its inverse matrix, its eigenvalues and singular values and the solutions of some linear systems.

In Section 6, Algorithm 3 provides the bidiagonal decomposition (7) of the Wronskian matrix (25) of the basis of Jacobi polynomials. Section 6 shows accurate results obtained when computing the mentioned algebraic problems using this algorithm and the algorithms presented in [11] and [12].

5 Collocation and Wronskian Matrices of well known Orthogonal Bases

In this section we are going to see that the results on properties and factorizations of collocation and Wronskian matrices of Jacobi polynomials obtained in the previous sections can be used to derive properties of collocation and Wronskian matrices of other well known orthogonal bases.

The following auxiliary results can be easily checked and will be useful to derive the bidiagonal decomposition of matrices obtained by scaling with a diagonal matrix a nonsingular and TP matrix.

Lemma 2 *Let F_i and $G_i, i = 1, \dots, n$, be the lower and upper, respectively, triangular bidiagonal matrices described in (8) and $\Delta = \text{diag}(d_1, d_2, \dots, d_{n+1})$ a nonsingular diagonal matrix. Then*

$$\Delta F_i = \widehat{F}_i \Delta \quad \text{and} \quad G_i \Delta = \Delta \widehat{G}_i, \quad i = 1, \dots, n, \tag{33}$$

where

$$\widehat{F}_i = \begin{pmatrix} 1 & & & & & \\ 0 & 1 & & & & \\ & \ddots & \ddots & & & \\ & & 0 & 1 & & \\ & & & r_{i+1,1} & 1 & \\ & & & & r_{i+2,2} & 1 \\ & & & & & \ddots & \ddots \\ & & & & & & r_{n+1, n+1-i} & 1 \end{pmatrix},$$

$$\widehat{G}_i^T = \begin{pmatrix} 1 & & & & & \\ 0 & 1 & & & & \\ & \ddots & \ddots & & & \\ & & 0 & 1 & & \\ & & & \tilde{r}_{i+1,1} & 1 & \\ & & & & \tilde{r}_{i+2,2} & 1 \\ & & & & & \ddots & \ddots \\ & & & & & & \tilde{r}_{n+1, n+1-i} & 1 \end{pmatrix}, \tag{34}$$

with

$$r_{i,j} = \frac{d_i}{d_{i-1}} m_{i,j}, \quad \tilde{r}_{i,j} = \frac{d_i}{d_{i-1}} \tilde{m}_{i,j}, \quad 1 \leq j < i \leq n + 1.$$

As a consequence, we have the following result.

Lemma 3 *Let $A = F_n F_{n-1} \dots F_1 D G_1 \dots G_{n-1} G_n$ be the bidiagonal decomposition (7) of a nonsingular and TP matrix A . Then, given a nonsingular matrix $\Delta = \text{diag}(d_1, d_2, \dots, d_{n+1})$, the bidiagonal decomposition (7) of ΔA and $A \Delta$ are given by*

$$\Delta A = \widehat{F}_n \widehat{F}_{n-1} \dots \widehat{F}_1 \widehat{D} G_1 \dots G_{n-1} G_n, \tag{35}$$

$$A \Delta = F_n F_{n-1} \dots F_1 \widehat{D} \widehat{G}_1 \dots \widehat{G}_{n-1} \widehat{G}_n, \tag{36}$$

where \widehat{F}_i and $\widehat{G}_i, i = 1, \dots, n$, are the lower and upper, respectively, triangular matrices described in (34) and $\widehat{D} = \Delta D = D \Delta$.

Let us start by considering the basis (L_0, \dots, L_n) of Legendre polynomials defined by

$$L_i(x) := J_i^{(0,0)}(x), \quad i = 0, \dots, n, \tag{37}$$

where $(J_0^{(0,0)}, \dots, J_n^{(0,0)})$ is the basis of Jacobi polynomials given in (11) with $\alpha = \beta = 0$. From Theorem 3, Remark 2, Theorem 4 and Remark 3, we can deduce the following result.

Theorem 5 *The basis (L_0, \dots, L_n) of Legendre polynomials, defined by (37), is STP on $(1, \infty)$. Given $x_1 < \dots < x_{n+1}$, with $x_1 > 1$, the bidiagonal decomposition (7) of the corresponding collocation matrix can be obtained with HRA. Moreover, for any $x > 1$, the Wronskian matrix $W(L_0, \dots, L_n)(x)$ is nonsingular and TP and its bidiagonal decomposition (7) can be obtained with HRA.*

Given $\lambda \in \mathbb{R}$, the basis of Gegenbauer polynomials of \mathbf{P}^n is (G_0, \dots, G_n) with

$$G_i^\lambda(x) := \frac{\Gamma(\lambda + 1/2)}{\Gamma(2\lambda)} \frac{\Gamma(i + 2\lambda)}{\Gamma(i + \lambda + 1/2)} J_i^{(\lambda-1/2, \lambda-1/2)}(x), \quad i = 0, \dots, n, \quad (38)$$

where $(J_0^{(\lambda-1/2, \lambda-1/2)}, \dots, J_n^{(\lambda-1/2, \lambda-1/2)})$ is the basis of Jacobi polynomials given in (11) with $\alpha = \beta = \lambda - 1/2$. By Theorem 3 and Remark 2, Lemma 3 and Remark 3, we can deduce the following result.

Theorem 6 *For any $\lambda > -1/2$, the basis (G_0, \dots, G_n) of Gegenbauer polynomials, defined by (38), is STP on $(1, \infty)$. Given $x_1 < \dots < x_{n+1}$, with $x_1 > 1$, the bidiagonal decomposition (7) of the corresponding collocation matrix can be obtained with HRA. Moreover, for any $x > 1$, the Wronskian matrix $W(G_0, \dots, G_n)(x)$ is nonsingular and TP and its bidiagonal decomposition (7) can be obtained with HRA.*

The basis (T_0, \dots, T_n) of Chebyshev polynomials of the first kind is defined by

$$T_i(x) := \frac{J_i^{(-1/2, -1/2)}(x)}{J_i^{(-1/2, -1/2)}(1)}, \quad i = 0, \dots, n, \quad (39)$$

where $(J_0^{(-1/2, -1/2)}, \dots, J_n^{(-1/2, -1/2)})$ is the basis of Jacobi polynomials given in (11) with $\alpha = \beta = -1/2$. Using again Theorem 3, Remark 2, Lemma 3 and Remark 3, we can deduce the following result.

Theorem 7 *The basis (T_0, \dots, T_n) of Chebyshev polynomials of the first kind, defined by (39), is STP on $(1, \infty)$. Given $x_1 < \dots < x_{n+1}$, with $x_1 > 1$, the bidiagonal decomposition (7) of the corresponding collocation matrix can be obtained with HRA. Moreover, for any $x > 1$, the Wronskian matrix $W(T_0, \dots, T_n)(x)$ is nonsingular and TP and its bidiagonal decomposition (7) can be obtained with HRA.*

The basis (U_0, \dots, U_n) of second kind Chebyshev polynomials is defined by

$$U_i(x) := (i + 1) \frac{J_i^{(-1/2, -1/2)}(x)}{J_i^{(1/2, 1/2)}(1)}, \quad i = 0, \dots, n, \quad (40)$$

where $(J_0^{(1/2, 1/2)}, \dots, J_n^{(1/2, 1/2)})$ is the basis of Jacobi polynomials given in (11) with $\alpha = \beta = 1/2$. Using again Theorem 3, Remark 2, Lemma 3 and Remark 3, we can deduce the following result.

Theorem 8 *The basis (U_0, \dots, U_n) of Chebyshev polynomials of second kind, defined by (40), is STP on $(1, \infty)$. Given $x_1 < \dots < x_{n+1}$, with $x_1 > 1$, the bidiagonal decomposition (7) of the corresponding collocation matrix can be obtained with HRA. Moreover, for any $x > 1$, the Wronskian matrix $W(U_0, \dots, U_n)(x)$ is nonsingular and TP and its bidiagonal decomposition (7) can be obtained with HRA.*

In [19], induced by Jacobi polynomials, a new orthogonal system of rational functions was introduced. For given $\alpha, \beta \in \mathbb{R}$, the system $(R_0^{(\alpha, \beta)}, \dots, R_n^{(\alpha, \beta)})$ of rational Jacobi functions is defined by

$$R_i^{(\alpha, \beta)}(x) := J_i^{(\alpha, \beta)}\left(\frac{x-1}{x+1}\right), \quad i = 0, \dots, n, \quad (41)$$

where $(J_0^{(\alpha, \beta)}, \dots, J_n^{(\alpha, \beta)})$ is the basis (11) of Jacobi polynomials. Using again Theorem 3, Remark 2, Lemma 3 and Remark 3, we can deduce the following result.

Theorem 9 *For any $\alpha, \beta > -1$, the basis $(R_0^{(\alpha, \beta)}, \dots, R_n^{(\alpha, \beta)})$ of rational Jacobi functions given in (41) is STP on $(-\infty, -1)$. Given $x_1 < \dots < x_{n+1}$, with $x_{n+1} < -1$, the bidiagonal decomposition (7) of the corresponding collocation matrix can be obtained with HRA.*

Similar results can be deduced by considering the rational counterparts of the basis of Legendre, Gegenbauer and the first and second kind Chebyshev polynomials.

Section 6 will show accurate results obtained when computing the eigenvalues, singular values, or the solutions of some linear systems associated with the collocation and Wronskian matrices of all the mentioned orthogonal bases, using their corresponding bidiagonal decompositions and the algorithms presented in [11] and [12].

6 Numerical Experiments

Given a nonsingular and TP matrix whose bidiagonal factorization (7) can be computed with HRA, the functions `TNEigenvalues`, `TNSingularValues`, `TNInverseExpand` and `TNSolve`, available in the library `TNTool` of [12], can be used to compute with HRA its eigenvalues, its singular values, its inverse matrix and the solution of some linear systems, respectively. The function `TNProduct` is also available in the mentioned library. If the bidiagonal decomposition (7) of two nonsingular and TP matrices A and B can be computed with HRA, `TNProduct` computes with HRA the bidiagonal decomposition (7) of AB .

Using Theorem 2, we have implemented the Matlab function `TNBDA` (see Algorithm 1) providing the bidiagonal decomposition (20) of the lower triangular matrix A given in (12), for given $\alpha, \beta > -1$ and $n \in \mathbb{N}$. Using `TNBDA` and taking into account Remark 1 and Theorem 3, we have also implemented the Matlab function `TNB DJ` (see Algorithm 2) for the computation of the bidiagonal decomposition (7) of the collocation matrix at $x = (x_i)_{i=1}^{n+1}$, with $1 < x_1 < \dots < x_{n+1}$, of the Jacobi polynomial basis corresponding to given $\alpha, \beta > -1$. Furthermore, using `TNBDA` and taking into account Proposition 2, Theorem 4 and Remark 3, we have implemented the Matlab function `TNB DWJ` (see Algorithm 3), which provides the bidiagonal decomposition (7) of the Wronskian matrix at $x > 1$ of the Jacobi polynomial basis corresponding to given $\alpha, \beta > -1$.

Moreover, using `TNB DJ` (`TNB DWJ`, respectively) and taking into account Lemma 3, we have also implemented the Matlab functions `TNB DG`, `TNB DT1` and `TNB DT2` (`TNB DWG`, `TNB DWT1`, `TNB DWT2`, respectively) for the computation of the bidiagonal decomposition (7) of the collocation matrix at x_1, \dots, x_{n+1} (of the Wronskian matrix at $x > 1$, respectively) of the bases (38) of Gegenbauer polynomials at a given $\lambda > -1/2$, the basis (39) of Chebyshev polynomials of the first kind and the basis (40) of Chebyshev polynomials of the second kind, respectively.

In order to check the accuracy of the solution of the above mentioned algebraic problems, obtained using the functions in [12] with the bidiagonal factorization (7), we have considered

Algorithm 1: Computation of the bidiagonal decomposition (20) of the matrix A in (12)

```

function  $BDA = \text{TNBDA}(\alpha, \beta, n + 1)$ 
 $BDA = \text{zeros}(n+1, n+1)$ 
for  $i := 2$  to  $n+1$ 
   $aux := \frac{\alpha+i-1}{i-1}$ 
   $BDA(i, 1) := aux$ 
  for  $j := 2$  to  $i-1$ 
     $aux := aux \cdot \frac{\alpha+\beta+2i-j}{\alpha+\beta+2i-j-2}$ 
     $BDA(i, j) := aux$ 
  end j
end i
 $BDA(1, 1) = 1$ 
for  $i := 2$  to  $n+1$ 
   $aux := 1$ 
  for  $k := 1$  to  $i-1$ 
     $aux := aux \cdot \frac{\alpha+\beta+2i-k-1}{i-k}$ 
     $BDA(i, i) := aux$ 
  end k
end i

```

Algorithm 2: Computation of the bidiagonal decomposition of the collocation matrix of Jacobi polynomials

```

function  $BDJ = \text{TNBDJ}(\alpha, \beta, x, n + 1)$ 
 $BDA = \text{TNBDA}(\alpha, \beta, n + 1)$ 
 $BDB = \text{zeros}(n + 1, n + 1)$ 
for  $i := 2$  to  $n+1$ 
   $BDB(i, 1) := 1$ 
   $aux := 1$ 
  for  $j := 2$  to  $i-1$ 
     $aux := aux \cdot \frac{x_i - x_{i-j+1}}{x_{i-1} - x_{i-j}}$ 
     $BDB(i, j) := aux$ 
  end j
end i
for  $i=1$  to  $n$ 
   $aux := (x_i - 1) / 2$ 
  for  $j := i+1$  to  $n+1$ 
     $BDB(i, j) := aux$ 
  end j
end i
 $aux := 1$ 
 $BDB(1, 1) = 1$ 
for  $i := 2$  to  $n+1$ 
   $aux := aux / 2$ 
  for  $k := 1$  to  $i-1$ 
     $aux := aux \cdot (x_i - x_k)$ 
     $BDB(i, i) := aux$ 
  end k
end i
 $BDJ = \text{TNProduct}(BDB, (BDA)^T)$ 

```

collocation matrices \mathbf{M}_n at $x = (x_i)_{i=1}^{n+1}$ satisfying $1 < x_1 < \dots < x_{n+1}$ and Wronskian matrices \mathbf{W}_n at $x = 2$ or $x = 50$, for $(n + 1)$ -dimensional Jacobi, Legendre, Gegenbauer and Chebyshev of the first and second kind polynomial bases. Additionally, we have also considered collocation matrices at sequences $x = ((x_i - 1)/(x_i + 1))_{i=1}^{n+1}$ such that $x_1 < x_2 < \dots < x_{n+1} < -1$ of their rational counterpart bases. For the considered collocation matrices, we have obtained the bidiagonal decomposition (7) by using TNBDJ, TNBDG, TNBDT1 and TNBDT2. For the considered Wronskian matrices, the factorization (7) has been obtained with the Matlab functions TNBDWJ, TNBDWG, TNBDWT1 and TNBDWT2. The software with the numerical experiments will be provided by the authors upon request.

Algorithm 3: Computation of the bidiagonal decomposition of the Wronskian matrix of Jacobi polynomials

```

function BDWJ = TNBDWJ( $\alpha, \beta, x, n + 1$ )
    BDA = TNBDA( $\alpha, \beta, n + 1$ )
    BDWB = zeros( $n + 1, n + 1$ )
    for i=1 to n+1
        for j :=i+1 to n+1
            BDWB(i, j) := ( $x - 1$ ) / 2
        end j
    end i
    BDWB(1, 1) := 1
    for i :=2 to n+1
        BDWB(i, i) := ( $i - 1$ ) · BDWB( $i - 1, i - 1$ ) / 2
    end i
    BDWJ = TNProduct(BDWB, (BDA)T)
    
```

Table 1 From left to right, condition number of collocation matrices at $x_i = 1 + i/(n + 1), i = 1, \dots, n + 1$, of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind polynomial bases

n+1	Jacobi $\kappa_2(M_n)$	Legendre $\kappa_2(M_n)$	Gegenbauer $\kappa_2(M_n)$	Chebyshev 1st kind $\kappa_2(M_n)$	Chebyshev 2nd kind $\kappa_2(M_n)$
10	4.2×10^{13}	5.3×10^{13}	1.2×10^{14}	1.4×10^{14}	1.8×10^{14}
15	5.1×10^{21}	9.2×10^{21}	2.1×10^{22}	3.2×10^{22}	2.1×10^{22}
20	8.2×10^{29}	1.9×10^{30}	4.4×10^{30}	7.7×10^{30}	4.4×10^{30}
25	1.5×10^{38}	4.4×10^{38}	1.0×10^{39}	2.0×10^{39}	1.0×10^{39}

Tables 1, 2, 3 and 4 illustrate the 2-norm condition number of the mentioned collocation and Wronskian matrices that have been obtained with the Mathematica command $\text{Norm}[A, 2] \cdot \text{Norm}[\text{Inverse}[A], 2]$. Observe that the condition number of the matrices considerably increases with their dimension. Due to this ill conditioning, traditional methods do not achieve accurate solutions when solving the mentioned algebraic problems. The following numerical results confirm this fact and illustrate the high accuracy obtained when using the functions in [12] with the bidiagonal factorizations (7) obtained in this paper.

6.1 Eigenvalues and Singular Values

Let us recall that all considered matrices are STP and so, all their eigenvalues are positive and distinct (see Theorem 6.2 of [1]). On the other hand, the eigenvalues of the mentioned Wronskian matrices are integers and so, they can be exactly determined.

We have compared the eigenvalues and singular values obtained when using the Matlab commands `eig` and `svd`, respectively, and those computed using the bidiagonal decompositions (7) in this paper and the Matlab functions `TNEigenValues` and `TNSingularValues`, respectively. In order to determine the accuracy of the approximations, we have also calculated the eigenvalues and singular values of the matrices by using Mathematica with a precision of 100 digits and computed the relative errors corresponding to the approximations, considering the eigenvalues and singular values provided by Mathematica as exact. We have computed the relative error of the approximations a of the exact eigenvalue and singular value \tilde{a} by means of the formula $e = |a - \tilde{a}|/|a|$.

Tables 5, 6, 7, 8 and 9 show the relative errors of the approximations to the lowest eigenvalue and the lowest singular value obtained with both methods. Observe that the eigenvalues

Table 2 From left to right, condition number of collocation matrices at $x = ((x_i - 1)/(x_i + 1))_{i=1}^{n+1}$ such that $x_i = -3 + i/(n + 1)$ of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind rational bases

n+1	Rational Jacobi $\kappa_2(M_n)$	Rational Legendre $\kappa_2(M_n)$	Rational Gegenbauer $\kappa_2(M_n)$	Rational Chebyshev 1st kind $\kappa_2(M_n)$	Rational Chebyshev 2nd kind $\kappa_2(M_n)$
10	1.7×10^{17}	1.5×10^{17}	3.7×10^{17}	4.1×10^{17}	3.7×10^{17}
15	1.7×10^{27}	2.2×10^{27}	5.5×10^{27}	7.6×10^{27}	5.5×10^{27}
20	2.4×10^{37}	3.9×10^{37}	1.0×10^{38}	1.6×10^{38}	1.0×10^{38}
25	4.0×10^{47}	8.2×10^{47}	2.1×10^{48}	3.8×10^{48}	2.1×10^{48}

Table 3 From left to right, condition number of Wronskian matrices at $x_0 = 2$ of the Jacobi (with $\alpha = 1$, $\beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind polynomial bases

n+1	Jacobi $\kappa_2(W_n)$	Legendre $\kappa_2(W_n)$	Gegenbauer $\kappa_2(W_n)$	Chebyshev 1st kind $\kappa_2(W_n)$	Chebyshev 2nd kind $\kappa_2(W_n)$
10	2.1×10^9	5.8×10^8	2.4×10^9	1.6×10^9	2.4×10^9
15	1.4×10^{16}	3.6×10^{15}	1.9×10^{16}	1.2×10^{16}	1.9×10^{16}
20	5.9×10^{23}	1.4×10^{23}	8.1×10^{23}	5.6×10^{23}	8.4×10^{23}
25	$8, 8 \times 10^{31}$	2.0×10^{31}	1.4×10^{32}	9.2×10^{31}	1.4×10^{32}

Table 4 From left to right, condition number of Wronskian matrices at $x_0 = 50$ of the Jacobi (with $\alpha = 1$, $\beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind polynomial bases

n+1	Jacobi $\kappa_2(W_n)$	Legendre $\kappa_2(W_n)$	Gegenbauer $\kappa_2(W_n)$	Chebyshev 1st kind $\kappa_2(W_n)$	Chebyshev 2nd kind $\kappa_2(W_n)$
10	5.8×10^{27}	1.0×10^{27}	5.6×10^{27}	2.8×10^{27}	5.6×10^{27}
15	7.1×10^{40}	1.2×10^{40}	7.7×10^{40}	4.0×10^{40}	7.7×10^{40}
20	1.5×10^{53}	2.6×10^{52}	1.9×10^{53}	1.0×10^{53}	1.9×10^{53}
25	9.4×10^{64}	1.6×10^{64}	1.3×10^{65}	7.2×10^{64}	1.3×10^{65}

and singular values obtained using the factorization (7) are very accurate for all considered n , whereas the approximations of the eigenvalues and singular values obtained with the Matlab commands `eig` and `svd` are not very accurate when n increases.

6.2 Inverse Matrix

We have also used the Matlab function `TNInverseExpand` (see Section 4 of [17]) with the factorization (7) proposed in this paper in order to compute the inverse of the considered collocation and Wronskian matrices. We have also computed their approximations with the Matlab command `inv`. In order to determine the accuracy of the approximations, we have calculated the inverse of these matrices by using Mathematica with a precision of 100 digits and computed the relative errors corresponding to the approximations, considering the inverse matrix provided by Mathematica as exact. We have computed the relative error of each approximation \tilde{A}^{-1} of the exact inverse matrix A^{-1} by means of the formula $e = \|A^{-1} - \tilde{A}^{-1}\|_2 / \|A^{-1}\|_2$.

Tables 10, 11 and 12 show the relative errors of the approximations to the inverse of the collocation and Wronskian matrices obtained with both methods. For all considered cases, the approximation of the inverse matrix obtained by means of `TNInverseExpand` and the factorization (7) is very accurate, providing much better results than those obtained by Matlab using the command `inv`.

6.3 Linear Systems

We shall illustrate the accuracy of the solutions of linear systems computed by using the bidiagonal factorization (7). We have obtained the solution of the linear systems using Math-

Table 5 From left to right, relative errors when computing the lowest eigenvalue of collocation matrices at $x_i = 1 + i/(n + 1), i = 1, \dots, n + 1$, of Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind polynomial bases

n+1	Jacobi		Legendre		Gegenbauer		Chebyshev 1st kind		Chebyshev 2nd kind	
	eig	TNEV	eig	TNEV	eig	TNEV	eig	TNEV	eig	TNEV
10	1.5×10^{-16}	5.4×10^{-16}	3.3×10^{-17}	6.9×10^{-16}	1.2×10^{-15}	4.9×10^{-16}	1.3×10^{-6}	1.0×10^{-15}	9.6×10^{-7}	3.0×10^{-16}
15	1.1×10^{-12}	2.2×10^{-16}	3.0×10^{-13}	3.5×10^{-16}	5.4×10^{-13}	2.3×10^{-15}	3.9	3.5×10^{-15}	1.3×10^{-1}	2.7×10^{-15}
20	1.2×10^{-11}	2.4×10^{-15}	7.7×10^{-12}	1.1×10^{-16}	4.3×10^{-12}	1.2×10^{-15}	5.9×10^8	5.3×10^{-15}	1.3×10^5	4.1×10^{-15}
25	1.3×10^{-10}	8.4×10^{-16}	1.9×10^{-10}	2.5×10^{-16}	1.8×10^{-10}	1.7×10^{-15}	1.5×10^9	9.9×10^{-15}	1.9×10^9	1.7×10^{-15}

Table 6 From left to right, relative errors when computing the lowest singular value of collocation matrices at $x_i = 1 + i/(n + 1), i = 1, \dots, n + 1$, of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind polynomial bases

n+1	Jacobi		Legendre		Gegenbauer		Chebyshev 1st kind		Chebyshev 2nd kind	
	svd	TNSV	svd	TNSV	svd	TNSV	svd	TNSV	svd	TNSV
10	2.1×10^{-16}	2.5×10^{-16}	5.9×10^{-17}	5.9×10^{-17}	1.6×10^{-15}	1.2×10^{-15}	1.3×10^{-6}	1.4×10^{-16}	9.5×10^{-7}	1.2×10^{-15}
15	1.9×10^{-12}	4.7×10^{-16}	3.7×10^{-13}	1.5×10^{-15}	8.2×10^{-13}	1.1×10^{-15}	4.6	3.8×10^{-15}	1.4×10^{-1}	3.4×10^{-15}
20	1.6×10^{-11}	2.5×10^{-15}	4.3×10^{-11}	1.4×10^{-15}	5.4×10^{-11}	7.8×10^{-16}	1.5×10^5	4.7×10^{-15}	1.5×10^5	1.2×10^{-15}
25	1.2×10^{-9}	7.4×10^{-16}	6.3×10^{-10}	6.9×10^{-16}	9.1×10^{-10}	1.8×10^{-15}	6.6×10^9	1.2×10^{-14}	8.5×10^9	1.5×10^{-15}

Table 7 From left to right, relative errors when computing the lowest eigenvalue of the collocation matrices at $x = ((x_i - 1)/(x_i + 1))_{i=1}^{n+1}$ such that $x_i = -3 + i/(n + 1)$ of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind rational bases

n+1	Rational Jacobi		Rational Legendre		Rational Gegenbauer		Rational Chebyshev 1st kind		Rational Chebyshev 2nd kind	
	eig	TNEV	eig	TNEV	eig	TNEV	eig	TNEV	eig	TNEV
10	5.8×10^{-17}	8.8×10^{-17}	4.9×10^{-18}	7.3×10^{-16}	8.6×10^{-17}	8.6×10^{-17}	9.6×10^{-17}	4.9×10^{-17}	8.7×10^{-17}	2.6×10^{-16}
15	8.0×10^{-14}	4.4×10^{-16}	1.3×10^{-14}	4.2×10^{-16}	3.1×10^{-14}	1.7×10^{-15}	3.5×10^{-15}	4.8×10^{-15}	1.9×10^{-14}	2.4×10^{-15}
20	1.8×10^{-12}	8.2×10^{-16}	1.7×10^{-12}	1.6×10^{-16}	2.1×10^{-12}	1.1×10^{-16}	1.2×10^{-12}	2.1×10^{-12}	2.0×10^{-12}	1.3×10^{-15}
25	1.6×10^{-11}	8.3×10^{-16}	7.4×10^{-11}	2.0×10^{-16}	3.4×10^{-12}	1.8×10^{-16}	7.1×10^{-12}	6.8×10^{-15}	4.7×10^{-12}	4.1×10^{-15}

Table 8 From left to right, relative errors when computing the lowest singular value of the collocation matrices at $x = ((x_i - 1)/(x_i + 1))_{i=1}^{n+1}$ such that $x_i = -3 + i/(n + 1)$ of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind rational bases

n+1	Rational Jacobi		Rational Legendre		Rational Gegenbauer		Rational Chebyshev 1st kind		Rational Chebyshev 2nd kind	
	svd	TNSV	svd	TNSV	svd	TNSV	svd	TNSV	svd	TNSV
10	2.8×10^{-17}	5.7×10^{-16}	9.4×10^{-17}	5.4×10^{-16}	1.8×10^{-16}	1.9×10^{-17}	2.8×10^{-17}	8.1×10^{-16}	2.0×10^{-17}	6.1×10^{-16}
15	7.0×10^{-14}	1.1×10^{-15}	9.9×10^{-15}	9.7×10^{-18}	3.2×10^{-14}	2.5×10^{-15}	4.4×10^{-15}	4.9×10^{-15}	1.4×10^{-14}	3.8×10^{-15}
20	3.9×10^{-12}	9.4×10^{-17}	2.2×10^{-12}	1.8×10^{-15}	3.3×10^{-12}	1.3×10^{-16}	1.5×10^{-12}	1.5×10^{-15}	3.0×10^{-12}	7.0×10^{-16}
25	3.9×10^{-12}	2.4×10^{-15}	2.3×10^{-11}	1.6×10^{-15}	2.5×10^{-11}	3.9×10^{-15}	2.7×10^{-11}	8.9×10^{-15}	2.2×10^{-11}	3.9×10^{-15}

Table 9 From left to right, relative errors when computing the lowest singular value of Wronskian matrices at $x_0 = 2$ of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind polynomial bases

n+1	Jacobi		Legendre		Gegenbauer		Chebyshev 1st kind		Chebyshev 2nd kind	
	svd	TNSV	svd	TNSV	svd	TNSV	svd	TNSV	svd	TNSV
10	3.5×10^{-9}	6.1×10^{-17}	1.5×10^{-9}	7.9×10^{-17}	4.2×10^{-9}	6.6×10^{-17}	6.3×10^{-9}	9.5×10^{-16}	4.5×10^{-9}	6.6×10^{-17}
15	2.0×10^{-2}	1.7×10^{-16}	6.7×10^{-3}	2.3×10^{-17}	2.6×10^{-2}	2.0×10^{-16}	2.5×10^{-2}	8.9×10^{-16}	2.7×10^{-2}	2.4×10^{-16}
20	1.6	3.2×10^{-16}	1.3	3.6×10^{-16}	2.2	6.0×10^{-17}	3.7	1.6×10^{-16}	7.8×10^{-1}	6.0×10^{-17}
25	2.1	3.8×10^{-16}	4.1	6.3×10^{-16}	2.9	1.6×10^{-16}	4.2	6.0×10^{-16}	2.9	1.4×10^{-15}

Table 10 From left to right, relative errors when computing the inverse of collocation matrices at $x_i = 1 + i/(n + 1)$, $i = 1, \dots, n + 1$, of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind polynomial bases

n+1	Jacobi		Legendre		Gegenbauer		Chebyshev 1st kind		Chebyshev 2nd kind	
	inv	TNInvExp	inv	TNInvExp	inv	TNInvExp	inv	TNInvExp	inv	TNInvExp
10	1.4×10^{-16}	3.8×10^{-16}	4.0×10^{-17}	8.2×10^{-16}	1.5×10^{-15}	5.4×10^{-16}	1.3×10^{-6}	4.8×10^{-6}	9.5×10^{-7}	1.6×10^{-15}
15	2.0×10^{-12}	7.5×10^{-16}	3.7×10^{-13}	3.6×10^{-16}	8.3×10^{-13}	1.9×10^{-15}	0.8	3.3×10^{-15}	1.6×10^{-1}	3.2×10^{-15}
20	3.3×10^{-11}	1.7×10^{-15}	4.6×10^{-11}	4.9×10^{-16}	5.8×10^{-11}	1.8×10^{-16}	1.0	4.6×10^{-15}	1.0	2.8×10^{-15}
25	1.4×10^{-9}	4.6×10^{-16}	6.3×10^{-10}	4.5×10^{-16}	9.2×10^{-10}	6.2×10^{-16}	1.0	9.9×10^{-15}	1.0	3.8×10^{-15}

Table 11 From left to right, relative errors when computing the inverse of the collocation matrices at $x = ((x_i - 1)/(x_i + 1))_{i=1}^{n+1}$ such that $x_i = -3 + i/(n + 1)$ of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind rational bases

n+1	Rational Jacobi		Rational Legendre		Rational Gegenbauer		Rational Chebyshev 1st kind		Rational Chebyshev 2nd kind	
	inv	TNInvExp	inv	TNInvExp	inv	TNInvExp	inv	TNInvExp	inv	TNInvExp
10	4.1×10^{-17}	8.5×10^{-17}	4.1×10^{-17}	1.9×10^{-16}	1.4×10^{-16}	2.9×10^{-16}	5.1×10^{-17}	3.9×10^{-16}	5.7×10^{-17}	3.0×10^{-16}
15	7.1×10^{-14}	2.2×10^{-16}	1.0×10^{-14}	3.0×10^{-16}	3.2×10^{-14}	2.5×10^{-15}	4.4×10^{-15}	3.7×10^{-15}	1.4×10^{-14}	2.6×10^{-15}
20	4.1×10^{-12}	7.0×10^{-16}	2.3×10^{-12}	3.2×10^{-16}	3.3×10^{-12}	5.3×10^{-16}	1.5×10^{-12}	2.9×10^{-15}	3.0×10^{-12}	8.2×10^{-16}
25	1.4×10^{-11}	5.9×10^{-16}	2.5×10^{-11}	1.1×10^{-15}	2.1×10^{-11}	1.0×10^{-15}	2.8×10^{-11}	7.8×10^{-15}	2.5×10^{-11}	2.4×10^{-15}

Table 12 From left to right, relative errors when computing the inverse of Wronskian matrices at $x_0 = 50$ of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind polynomial bases

n+1	Jacobi		Legendre		Gegenbauer		Chebyshev of the first kind		Chebyshev of the second kind	
	<i>inv</i>	TNInvExp	<i>inv</i>	TNInvExp	<i>inv</i>	TNInvExp	<i>inv</i>	TNInvExp	<i>inv</i>	TNInvExp
10	1.6×10^{-13}	1.7×10^{-17}	1.4×10^{-13}	1.8×10^{-16}	1.1×10^{-13}	7.5×10^{-17}	1.3×10^{-14}	2.2×10^{-16}	8.0×10^{-14}	6.0×10^{-16}
15	7.8×10^{-11}	4.5×10^{-17}	2.2×10^{-11}	2.0×10^{-16}	2.2×10^{-11}	4.1×10^{-15}	5.9×10^{-13}	4.6×10^{-15}	4.6×10^{-11}	4.4×10^{-15}
20	7.3×10^{-9}	1.4×10^{-16}	8.3×10^{-9}	3.9×10^{-16}	1.1×10^{-8}	1.2×10^{-15}	6.6×10^{-10}	3.3×10^{-15}	6.7×10^{-10}	1.9×10^{-15}
25	2.4×10^{-6}	1.0×10^{-16}	5.3×10^{-6}	5.0×10^{-16}	8.0×10^{-7}	4.6×10^{-15}	1.4×10^{-7}	8.2×10^{-15}	3.7×10^{-7}	4.7×10^{-15}

Table 13 From left to right, relative errors when solving $\mathbf{M}_n \mathbf{c}_n = \mathbf{d}_n$ with collocation matrices at $x_i = 1 + i/(n + 1), i = 1, \dots, n + 1$ of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind polynomial bases

n+1	Jacobi		Legendre		Gegenbauer		Chebyshev 1st kind		Chebyshev 2nd kind	
	$M_n \setminus \mathbf{d}_n$	TNSolve	$M_n \setminus \mathbf{d}_n$	TNSolve	$M_n \setminus \mathbf{d}_n$	TNSolve	$M_n \setminus \mathbf{d}_n$	TNSolve	$M_n \setminus \mathbf{d}_n$	TNSolve
10	1.4×10^{-16}	4.1×10^{-16}	6.6×10^{-17}	9.4×10^{-16}	1.5×10^{-15}	5.4×10^{-16}	1.3×10^{-6}	4.8×10^{-16}	9.5×10^{-7}	6.1×10^{-16}
15	2.0×10^{-12}	6.5×10^{-16}	3.7×10^{-13}	1.4×10^{-16}	8.3×10^{-13}	1.9×10^{-15}	8.2×10^{-1}	3.3×10^{-15}	1.6×10^{-1}	3.0×10^{-15}
20	3.3×10^{-11}	1.2×10^{-15}	4.6×10^{-11}	3.9×10^{-16}	5.8×10^{-11}	1.8×10^{-16}	1.0	4.6×10^{-15}	1.0	2.7×10^{-15}
25	1.4×10^{-9}	5.4×10^{-16}	6.3×10^{-10}	2.3×10^{-16}	9.2×10^{-10}	6.2×10^{-16}	1.0	9.9×10^{-15}	1.0	1.1×10^{-15}

Table 14 From left to right, relative errors when solving $\mathbf{M}_n \mathbf{c}_n = \mathbf{d}_n$ with collocation matrices at $x = ((x_i - 1)/(x_i + 1))_{i=1}^{n+1}$ such that $x_i = -3 + i/(n + 1)$ of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind rational bases

n+1	Rational Jacobi		Rational Legendre		Rational Gegenbauer		Rational Chebyshev 1st kind		Rational Chebyshev 2nd kind	
	$M_n \setminus \mathbf{d}_n$	TNSolve	$M_n \setminus \mathbf{d}_n$	TNSolve	$M_n \setminus \mathbf{d}_n$	TNSolve	$M_n \setminus \mathbf{d}_n$	TNSolve	$M_n \setminus \mathbf{d}_n$	TNSolve
10	2.9×10^{-17}	1.1×10^{-16}	7.9×10^{-17}	3.8×10^{-16}	1.2×10^{-16}	2.1×10^{-16}	5.5×10^{-17}	8.48×10^{-16}	6.3×10^{-17}	2.1×10^{-16}
15	7.1×10^{-14}	1.9×10^{-16}	1.0×10^{-14}	1.9×10^{-16}	3.2×10^{-14}	2.4×10^{-15}	4.4×10^{-15}	3.6×10^{-15}	1.4×10^{-14}	2.8×10^{-15}
20	4.1×10^{-12}	4.3×10^{-16}	2.3×10^{-12}	3.0×10^{-16}	3.3×10^{-12}	3.7×10^{-16}	1.5×10^{-12}	2.5×10^{-15}	3.4×10^{-12}	5.6×10^{-16}
25	1.4×10^{-11}	7.3×10^{-16}	2.5×10^{-11}	1.2×10^{-15}	2.8×10^{-11}	1.7×10^{-15}	2.8×10^{-11}	7.3×10^{-15}	2.5×10^{-11}	2.7×10^{-15}

Table 15 From left to right, relative errors when solving $\mathbf{W}_n \mathbf{c}_n = \mathbf{d}_n$ at $x_0 = 50$ of the Jacobi (with $\alpha = 1, \beta = 2$), Legendre, Gegenbauer (with $\lambda = 1$) and Chebyshev of the first and second kind polynomial bases

n+1	Jacobi		Legendre		Gegenbauer		Chebyshev 1st kind		Chebyshev 2nd kind	
	$W_n \setminus \mathbf{d}_n$	TNsolve	$W_n \setminus \mathbf{d}_n$	TNsolve	$W_n \setminus \mathbf{d}_n$	TNsolve	$W_n \setminus \mathbf{d}_n$	TNsolve	$W_n \setminus \mathbf{d}_n$	TNsolve
10	3.1×10^{-16}	1.1×10^{-16}	4.3×10^{-14}	2.3×10^{-16}	2.1×10^{-14}	4.5×10^{-17}	1.2×10^{-14}	2.6×10^{-16}	4.5×10^{-9}	6.6×10^{-17}
15	4.3×10^{-11}	9.5×10^{-17}	5.8×10^{-12}	2.8×10^{-16}	1.5×10^{-11}	3.2×10^{-15}	5.9×10^{-13}	4.6×10^{-15}	2.7×10^{-2}	2.4×10^{-16}
20	4.6×10^{-9}	1.5×10^{-16}	6.3×10^{-9}	6.1×10^{-16}	5.6×10^{-9}	1.3×10^{-15}	2.3×10^{-9}	2.6×10^{-15}	7.8×10^{-1}	6.0×10^{-17}
25	6.2×10^{-6}	1.4×10^{-16}	4.4×10^{-6}	4.2×10^{-16}	1.2×10^{-6}	7.8×10^{-16}	2.1×10^{-7}	4.5×10^{-15}	2.9	1.4×10^{-15}

emática with a precision of 100 digits and considered this solution exact. Then we have also computed with Matlab two approximations, the first one using the previous functions and the second one using the Matlab command `\`. We have computed the relative error of every approximation $\tilde{c} = (\tilde{c}_1, \dots, \tilde{c}_{n+1})$ of the solution c of the linear system by means of the formula $e = \|c - \tilde{c}\|_2 / \|c\|_2$.

Tables 13, 14 and 15 show the relative errors when solving the linear systems $\mathbf{M}_n c_n = \mathbf{d}_n$ and $\mathbf{W}_n c_n = \mathbf{d}_n$ where $\mathbf{d}_n = ((-1)^{i+1} d_i)_{1 \leq i \leq n+1}$ and $d_i, i = 1, \dots, n+1$, random integer values. The computed results confirm the accuracy of the proposed method that, clearly, keeps the accuracy when the dimension of the problem increases. In contrast, when n increases the condition number of the considered matrices considerably increases and that explains the bad results obtained with the Matlab command `\`.

References

1. Ando, T.: Totally positive matrices. *Linear Algebra Appl.* **90**, 165–219 (1987)
2. Beals, R., Wong, R.: *Special functions and orthogonal polynomials*. Cambridge University Press, Cambridge (2016)
3. Delgado, J., Orera, H., Peña, J.M.: Accurate computations with Laguerre matrices, *Numer Linear Algebra Appl.* **26**: e2217 (10 pp.) (2019)
4. Delgado, J., Orera, H., Peña, J.M.: Accurate algorithms for Bessel matrices. *J. Sci. Comput.* **80**, 1264–1278 (2019)
5. Demmel, J., Koev, P.: The accurate and efficient solution of a totally positive generalized Vandermonde linear system. *SIAM J. Matrix Anal. Appl.* **27**, 42–52 (2005)
6. Fallat, S.M., Johnson, C.R.: *Totally Nonnegative Matrices*, Princeton University Press, Princeton, NJ, Princeton Series in Applied Mathematics (2011)
7. Gasca, M., Peña, J.M.: Total positivity and Neville elimination. *Linear Algebra Appl.* **165**, 25–44 (1992)
8. Gasca, M., Peña, J.M.: A matricial description of Neville elimination with applications to total positivity. *Linear Algebra Appl.* **202**, 33–53 (1994)
9. Gasca, M., Peña, J.M.: On factorizations of totally positive matrices. In: Gasca, M., Micchelli, C.A. (eds.) *Total Positivity and Its Applications*, pp. 109–130. Kluwer Academic Publishers, Dordrecht, The Netherlands (1996)
10. Koev, P.: Accurate eigenvalues and SVDs of totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.* **27**, 1–23 (2005)
11. Koev, P.: Accurate computations with totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.* **29**, 731–751 (2007)
12. Koev, P.: <http://math.mit.edu/plamen/software/TNTool.html>
13. Koornwinder, T.H., Wong, R.S.C., Koekoek, R., Swarttouw, R.F.: Orthogonal Polynomials, in: F. W. J. Olver, D. M. Lozier, R. F. Boisver, C.W. Clark (Eds.), *NIST Handbook of Mathematical Functions*, Cambridge University Press, Cambridge, Chapter 18 (2010)
14. Mainar, E., Peña, J.M.: Accurate computations with collocation matrices of a general class of bases, *Numer Linear Algebra Appl.* **25**: e2184 (12 pp.) (2018)
15. Mainar, E., Peña, J.M., Rubio, B.: Accurate computations with Wronskian matrices. *Calcolo* **58**, 1 (2021)
16. Marco, A., Martínez, J.J.: Accurate computations with totally positive Bernstein-Vandermonde matrices. *Electron. J. Linear Algebra* **26**, 357–380 (2013)
17. Marco, A., Martínez, J.J.: Accurate computation of the Moore-Penrose inverse of strictly totally positive matrices. *J. Comput. Appl.* **350**, 299–308 (2019)
18. Pinkus, A.: *Totally positive matrices*, Cambridge Tracts in Mathematics, 181. Cambridge University Press, Cambridge (2010)
19. Wang, Z.-Q., Guo, B.-Y.: Jacobi rational approximation and spectral for differential equations of degenerate type. *Math. Comp.* **77**, 883–907 (2008)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Part III

JUSTIFICATION OF THE THEMATIC UNIT AND SUMMARY OF THE PUBLICATIONS

This doctoral thesis is presented as a compendium of the publications [73], [74], [39], [75] and [76]. The purpose of this part is to justify the thematic unit of the aforementioned publications. Their main results are also included.

4

Total positivity and Weighted φ -transformed systems: shape preserving properties and accurate computations

ABOUT THIS CHAPTER

The purpose of this chapter is to justify the thematic unit of the articles [73], [74] and [39], which belong to the compendium of publications of this thesis (see the articles on pages 43, 59 and 77, respectively). Their main results are also presented.

- [73] E. Mainar, J.M. Peña, B. Rubio, Evaluation and subdivision algorithms for general classes of totally positive rational bases, *Computer Aided Geometric Design* **81** (2020).
- [74] E. Mainar, J.M. Peña, B. Rubio, Accurate bidiagonal decomposition of collocation matrices of weighted φ -transformed systems, *Numerical Linear Algebra Appl.* **e2295** (2020).
- [39] R. Gonzalez, E. Mainar, E. Paluzo, B. Rubio, Neural-Network-Based Curve Fitting Using Totally Positive Rational Bases, *Mathematics* **8**, 2197 (2020).

4.1 Introduction

In this chapter we present a very general procedure for generating, from an initial system and a positive function φ , new systems of functions useful for curve design. These systems, which we call weighted φ -transformed systems, arise with relevant probability distributions. They also include important rational bases (see [97], [54]) as well as systems belonging to spaces that mix algebraic, trigonometric and hyperbolic polynomials, which are useful in many applications, for instance in Isogeometric Analysis (cf. [81]). The weighted φ -transformed systems inherit from the initial system its nice geometric properties and its accuracy when computing with its collocation matrices.

Let us recall that shape preserving representations in Computer-Aided Geometric Design (CAGD) are associated with normalized totally positive (NTP) bases, because these bases guarantee that the curve imitates the geometric properties of its control polygon. Among all NTP bases of a space, there exists a unique normalized B-basis, which is the basis with optimal shape preserving properties (cf. [92], [10]). The Bernstein bases and the B-spline bases are the normalized B-bases of their corresponding spaces. We shall prove in Section 4.3 that the shape preserving properties associated to NTP bases and normalized B-bases are inherited by the representations associated to their weighted φ -transformed systems. In fact, the total positivity and the property of being a B-basis are preserved. Therefore, using

the results of Section 4.3, one can deduce that bases formed by Poisson functions are B-bases and that rational systems obtained from B-bases are the normalized B-bases of the generated spaces of rational functions and have optimal shape preserving properties. In particular, it can be deduced that the Rational Bernstein basis and NURBS bases are NTP and the normalized B-bases of the corresponding spaces of rational functions.

In [70], for a given space of functions that admits shape preserving representations a corner cutting algorithm, called a B-algorithm, is proposed. B-algorithms are evaluation algorithms satisfying important properties such as a subdivision property and convergence to the curve of the resulting control polygons. Supported by the results in [70], in Section 4.4 evaluation and subdivision algorithms for a general class of rational bases, which can be considered as weighted φ -transformed systems, are deduced.

In [97] nested spaces of rational polynomial functions of a given degree n and with a common denominator are considered. The corresponding rational Bézier curves admit up to $n!$ different de Casteljau-type algorithms. In Section 4.5 we show that the results from [97] can be extended to spaces of non polynomial rational functions deriving recurrence formulas for the weights and basis functions of these spaces. Curves generated by these weighted φ -transformed bases inherit geometric properties and algorithms of the traditional rational Bézier curves and so they can be considered as modeling tools in CAD/CAM systems.

Let us recall that an algorithm can be performed with high relative accuracy (HRA) if it does not include subtractions (except of the initial data), that is, if it only includes products, divisions, sums of numbers of the same sign, subtractions of numbers of opposite sign or subtraction of initial data (cf. [62]). It is known that, for the collocation matrices A associated to some important bases used in CAGD, many algebraic computations can be performed with HRA. For instance, the computation of their eigenvalues, singular values or the solutions of linear systems $Ax = b$ such that the components of b have alternating signs (see [19] and the references therein). The key tool for this purpose is the use of a bidiagonal factorization of the collocation matrix, which can be obtained with HRA for those bases. This task was performed in [15] for the collocation matrices of rational Bernstein bases. In Section 4.3, we extend the analysis to our more general framework and assure that these computations can be performed with HRA for the collocation matrices of weighted φ -transformed systems, assuming that the bidiagonal factorization of the corresponding collocation matrix of the initial system can be obtained with HRA and that the evaluation of φ does not requires subtractions up to initial data. Our numerical examples in Section 4.8 illustrate that the computation of the eigenvalues and singular values and the solution of linear systems can be solved accurately even when the above conditions do not hold. The complexity of the proposed algorithms for solving the mentioned algebraic problems is comparable to that of the traditional LAPACK algorithms, which, as we shall illustrate, deliver no such accuracy.

To solve the problem of finding a rational curve to fit a given set of data points, we have proposed in Section 4.9 a one-hidden-layer neural network based on the general class of totally positive rational bases, presented in Section 4.4, which belong to spaces that mix algebraic, trigonometric and hyperbolic polynomials, thus being able to reach more difficult forms shapes and thus expanding the potential range of applications of this neural network. In order to obtain the weights and control points of the rational curve to fit the data points, the neural network is trained with an optimization algorithm to update the weights and control points while decreasing a loss function. The fitting curves of the numerical experiments show that for certain curves the use of particular rational bases provide better results.

As we have mentioned before, in Section 4.3 algorithms for the computation of the bidiagonal decomposition of square collocation matrices of a very general class of non-polynomial rational bases with interest in CAGD and Approximation Theory are provided. In Section 4.10, following the approach

of [84] for a polynomial case and taking into account the obtained results in [72] and [25], we generalize the mentioned bidiagonal factorization to the case of rectangular collocation matrices. Using their QR decompositions, we focus on the problem of least squares fitting in the spaces generated by the general class of rational bases proposed. By computing the bidiagonal decomposition of the coefficient matrix of the least squares problem, an algorithm for the computation of its QR decomposition is then applied. Finally, using the bidiagonal decomposition of the matrix factor R , a triangular system is solved.

The layout of the chapter is as follows. Section 4.2 includes matrix notations and basic concepts, such as totally positive bases and B-bases. We recall the Neville elimination procedure, which allows us to introduce the bidiagonal factorization of a strictly totally positive matrix. Section 4.3 introduces the weighted φ -transformed systems and includes the results guaranteeing the above mentioned nice geometric and computational properties. The bidiagonal factorization of the collocation matrices of the weighted φ -transformed systems is obtained. Section 4.4 shows a general class of rational spaces that can be generated by weighted φ -transformed systems. In particular, evaluation and subdivision algorithms for these rational bases are deduced. Moreover, Section 4.5 presents results that allow us to deduce shape preserving properties of particular rational functions that cover the family of bases introduced in [97]. In Section 4.6 the bidiagonal factorization of the collocation matrices of the mentioned rational bases is provided. Furthermore, Section 4.7 shows interesting examples and Section 4.8 includes numerical examples showing the accurate solution of linear systems associated to the collocation matrices of rational bases. In Section 4.9, we present a one-hidden-layer neural network based on the rational bases presented in Section 4.4 to approximate a given set of data points. This neural network is trained with an optimization algorithm to update the weights and control points used to construct a curve that approximates the given set of data points, while decreasing a loss function. Several experiments are provided illustrating the use of the neural network with different normalized B-bases to test its performance giving an approximation of different kinds of sets of data points. Finally, in Section 4.10 an accurate algorithm for curve fitting using rational bases and the least square method is presented.

4.2 Notations and auxiliary results

Let us recall that a matrix is *totally positive* (TP) if all its minors are nonnegative and *strictly totally positive* (STP) if they are positive (see [2]). A system of functions (u_0, \dots, u_n) defined on the subset $I \subseteq \mathbb{R}$ is TP if all its collocation matrices $(u_{j-1}(t_i))_{i,j=1,\dots,n+1}$, $t_1 < \dots < t_{n+1}$ in I are TP. A TP system of functions on I is *normalized* (NTP) if $\sum_{i=0}^n u_i(t) = 1$, for all $t \in I$. NTP bases are commonly used in CAGD due to their shape preserving properties (see [9], [92]).

Among all NTP bases of a space, we can find a unique normalized B-basis, which is the optimal shape preserving basis (cf. [10]). For instance, the Bernstein bases and the B-spline bases are the normalized B-bases of their corresponding spaces. The following characterization of a B-basis is a consequence of Corollary 3.10 of [10] and Proposition 3.11 of [10].

Theorem 4.1. *Let (u_0, \dots, u_n) be a TP basis of a space \mathcal{U} . Then (u_0, \dots, u_n) is a B-basis if for any other TP basis (v_0, \dots, v_n) of \mathcal{U} the matrix K of change of basis such that $(v_0, \dots, v_n) = (u_0, \dots, u_n)K$ is TP.*

Let us now recall some basic matrix notations and results on Neville elimination. Our notation follows the notation used in [34, 37]. Given $n \in \mathbb{N}$ and $k \in \{1, \dots, n\}$, let $Q_{k,n}$ be the set of increasing sequences of k positive integers less than or equal to n . If $\alpha, \beta \in Q_{k,n}$, we denote by $A[\alpha|\beta]$ the $k \times k$ submatrix of A containing rows of places α and columns of places β . Neville elimination (see [34, 37]), is a procedure to make zeros in a column of a matrix by adding to a given row an appropriate multiple of

the previous one. For a given nonsingular matrix $A = (a_{i,j})_{1 \leq i,j \leq n}$, this elimination procedure consists of at most $n - 1$ successive major steps, resulting in the sequence of matrices:

$$A^{(1)} := A \rightarrow A^{(2)} \rightarrow \cdots \rightarrow A^{(n)} = U.$$

For $1 \leq k \leq n - 1$, $A^{(k+1)} = (a_{i,j}^{(k+1)})_{1 \leq i,j \leq n}$ is obtained from $A^{(k)} = (a_{i,j}^{(k)})_{1 \leq i,j \leq n}$ by defining

$$a_{i,j}^{(k+1)} := a_{i,j}^{(k)} - (a_{i,k}^{(k)} / a_{i-1,k}^{(k)}) a_{i-1,j}^{(k)} \quad \text{if } a_{i-1,k}^{(k)} \neq 0, \quad k+1 \leq i, j \leq n,$$

so that $A^{(k+1)}$ has zeros below its main diagonal in the k first columns. Finally, U is an upper triangular matrix. The element $p_{i,j} := a_{i,j}^{(j)}$, is called the (i, j) pivot of the Neville elimination of A for $1 \leq j \leq i \leq n$. The pivots $p_{i,i}$ are called diagonal pivots. The Neville elimination can be performed without row exchanges if all the pivots are nonzero and, in this case, Lemma 2.6 of [34] implies that $p_{i,1} = a_{i,1}$, for $1 \leq i \leq n$, and

$$p_{i,j} = \frac{\det A[i-j+1, \dots, i | 1, \dots, j]}{\det A[i-j+1, \dots, i-1 | 1, \dots, j-1]}, \quad 1 < j \leq i \leq n. \quad (4.1)$$

Furthermore, the (i, j) multiplier of the Neville elimination of A is

$$m_{i,j} := a_{i,j}^{(j)} / a_{i-1,j}^{(j)} = p_{i,j} / p_{i-1,j}, \quad 1 \leq j < i \leq n. \quad (4.2)$$

Neville elimination has been used to characterize TP and STP matrices (see [34, 37]). From Theorem 4.1 of [34] and p. 116 of [37] (see also Theorem 2.1 of [15]), a given matrix A is STP if and only if the Neville elimination of A and A^T can be performed without row exchanges, all the multipliers of the Neville elimination of A and A^T are positive and all the diagonal pivots of the Neville elimination of A are positive.

According to the arguments of p.116 of [37], an STP matrix $A \in \mathbb{R}^{(n+1) \times (n+1)}$ can be factorized in the form

$$A = F_n F_{n-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n, \quad (4.3)$$

where F_i and G_i are the lower and upper triangular bidiagonal matrices

$$F_i = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & m_{i+1,1} & 1 & & \\ & & & \ddots & \ddots & \\ & & & & m_{n+1,n+1-i} & 1 \end{pmatrix}, \quad G_i^T = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \hat{m}_{i+1,1} & 1 & & \\ & & & \ddots & \ddots & \\ & & & & \hat{m}_{n+1,n+1-i} & 1 \end{pmatrix}, \quad (4.4)$$

and $D = \text{diag}(p_{1,1}, \dots, p_{n+1,n+1})$. The entries $m_{i,j}$ and $\hat{m}_{i,j}$ are the multipliers of the Neville elimination of A and A^T , respectively, and the diagonal entries $p_{i,i}$ are the diagonal pivots of the Neville elimination of A .

In [61], the bidiagonal factorization (4.3) of an $(n+1) \times (n+1)$ nonsingular and TP matrix A is represented by defining a matrix $BD(A) = (BD(A)_{i,j})_{1 \leq i,j \leq n+1}$ such that

$$BD(A)_{i,j} := \begin{cases} m_{i,j}, & \text{if } i > j, \\ p_{i,i}, & \text{if } i = j, \\ \tilde{m}_{j,i}, & \text{if } i < j. \end{cases} \quad (4.5)$$

Finally, let us recall that $x \in \mathbb{R}$ is obtained with high relative accuracy (HRA) if the relative error of the computed value \tilde{x} satisfies

$$\frac{\|x - \tilde{x}\|}{\|x\|} < Ku,$$

where K is a positive constant independent of the arithmetic precision and u is the unit round-off. HRA implies that the relative errors of the computations are of the order of the machine precision. An algorithm can be computed with HRA when it only uses products, quotients, sums of numbers of the same sign, subtractions of numbers of opposite sign or subtraction of initial data (cf. [23], [61]).

If the bidiagonal factorization (4.3) of a nonsingular TP matrix A is computed with HRA then, using the algorithms in [62], we can also compute with HRA its eigenvalues and singular values, the matrix A^{-1} and even the solution of $Ax = b$ for vectors b with alternating signs.

4.3 Weighted φ -transformed systems

Let us first introduce a key concept of [74] and [73] (see the articles on pages 43, 59). Let (u_0, \dots, u_n) be a system of functions defined on $I = [a, b]$, $\varphi : [a, b] \rightarrow \mathbb{R}$ a positive function and d_0, \dots, d_n positive real values. The corresponding *weighted φ -transformed system* from (u_0, \dots, u_n) is the system $(\tilde{u}_0, \dots, \tilde{u}_n)$ of functions defined by

$$\tilde{u}_i(t) := d_i \varphi(t) u_i(t), \quad t \in [a, b], \quad i = 0, \dots, n. \quad (4.6)$$

The following result is presented in [73], proves that a weighted φ -transformed system inherits the properties of being TP and being a B-basis.

Theorem 4.2. *Let (u_0, \dots, u_n) be a system of functions defined on $I = [a, b]$ and let $(\tilde{u}_0, \dots, \tilde{u}_n)$ be the weighted φ -transformed system given by (4.6).*

- i) *If (u_0, \dots, u_n) is TP, then $(\tilde{u}_0, \dots, \tilde{u}_n)$ is TP.*
- ii) *If (u_0, \dots, u_n) is a B-basis, then $(\tilde{u}_0, \dots, \tilde{u}_n)$ is a B-basis.*

Proof. See Theorem 2 of [73] (see the articles on pages 43). □

Let us suppose that (u_0, \dots, u_n) is a system of functions defined on $I = [a, b]$ and $a < t_1 < \dots < t_{n+1} < b$ is a sequence of nodes such that the corresponding collocation matrix

$$A := (u_{j-1}(t_i))_{1 \leq i, j \leq n+1} \quad (4.7)$$

is STP. Let

$$A = F_n F_{n-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n \quad (4.8)$$

be the bidiagonal factorization (4.3) such that F_i and G_i are the lower and upper bidiagonal matrices of the form (4.4) and D is a diagonal matrix.

The following result is presented in [74] and proves that the collocation matrix of the corresponding weighted φ -transformed system $(\tilde{u}_0, \dots, \tilde{u}_n)$ at nodes $a < t_1 < \dots < t_{n+1} < b$

$$\tilde{A} := (\tilde{u}_{j-1}(t_i))_{1 \leq i, j \leq n+1} = (d_{j-1} \varphi(t_i) u_{j-1}(t_i))_{1 \leq i, j \leq n+1} \quad (4.9)$$

is also STP and obtains its bidiagonal factorization (4.3) from the factorization (4.8) of the collocation matrix A given in (4.7).

Theorem 4.3. *The collocation matrix (4.9) is STP and it can be factorized as*

$$\tilde{A} = \tilde{F}_n \tilde{F}_{n-1} \cdots \tilde{F}_1 \tilde{D} \tilde{G}_1 \cdots \tilde{G}_{n-1} \tilde{G}_n, \quad (4.10)$$

where \tilde{F}_i and \tilde{G}_i are the lower and upper bidiagonal matrices of the form

$$\tilde{F}_i = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & r_{i+1,1} & 1 & & \\ & & & \ddots & \ddots & \\ & & & & r_{n+1,n+1-i} & 1 \end{pmatrix}, \quad \tilde{G}_i^T = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \hat{r}_{i+1,1} & 1 & & \\ & & & \ddots & \ddots & \\ & & & & \hat{r}_{n+1,n+1-i} & 1 \end{pmatrix} \quad (4.11)$$

and $\tilde{D} = \text{diag}(q_{1,1}, \dots, q_{n+1,n+1})$. The entries $r_{i,j}$, $\hat{r}_{i,j}$ and $q_{i,i}$ are given by

$$\begin{aligned} r_{i,j} &= \frac{\varphi(t_i)}{\varphi(t_{i-1})} m_{i,j}, & \hat{r}_{i,j} &= \frac{d_{i-1}}{d_{i-2}} \hat{m}_{i,j}, & 1 \leq j < i \leq n+1, \\ q_{i,i} &= d_{i-1} \varphi(t_i) p_{i,i}, & 1 \leq i \leq n+1, \end{aligned}$$

where $m_{i,j}$, $\hat{m}_{i,j}$ and $p_{i,i}$ are the entries of the matrices of the bidiagonal factorization (4.8) of the collocation matrix A defined in (4.7).

Proof. See Theorem 2 of [74] (see the article on page 59). \square

Observe that, if the evaluation of φ does not include subtractions (except for the initial data), the entries of the bidiagonal factorization of Theorem 4.3 can be obtained from the bidiagonal factorization of (4.8) without performing subtractions. Therefore, if the bidiagonal factorization of (4.8) can be performed with HRA, then the bidiagonal factorization of Theorem 4.3 can be also performed with HRA. It is known that, for the collocation matrices associated to some important bases used in CAGD, the bidiagonal factorization can be performed with HRA (see [71]). In consequence, the bidiagonal factorization of the collocation matrices of their corresponding weighted φ -transformed systems can be performed with HRA and we can apply the algorithms presented in [62] and [63] to perform many algebraic computations with HRA. For instance, the computation of their eigenvalues, singular values or the solutions of some linear systems associated to these collocation matrices.

Given a system (u_0, \dots, u_n) of functions defined on I and positive values d_0, \dots, d_n such that $\sum_{k=0}^n d_k u_k(t) \neq 0$, for all $t \in I$, the system (r_0, \dots, r_n) defined by

$$r_i(t) := \frac{d_i u_i(t)}{\sum_{k=0}^n d_k u_k(t)}, \quad i = 0, \dots, n,$$

satisfies $\sum_{i=0}^n r_i(t) = 1$, $\forall t \in I$, and generates a new space of rational functions. If (u_0, \dots, u_n) is TP then $\sum_{k=0}^n d_k u_k(t) > 0$, $\forall t \in I$, and (r_0, \dots, r_n) can be considered as a particular weighted φ -transformed system with

$$\varphi(t) := \frac{1}{\sum_{k=0}^n d_k u_k(t)}, \quad t \in I. \quad (4.12)$$

By Theorem 4.2, (r_0, \dots, r_n) is NTP. Furthermore, if (u_0, \dots, u_n) is a B-basis, we can also use Theorem 4.2 to deduce that (r_0, \dots, r_n) is the normalized B-basis of the generated space. We would like to observe that although this fact had been mentioned in [70], we have not found any proof of it in the literature. Rational Bernstein basis are NTP basis (see [15]). From Theorem 4.2 the optimal shape preserving

properties of these rational polynomial systems can be also guaranteed. Similarly, using Theorem 4.2 one can deduce that NURBS systems are NTP and the normalized B-bases of the corresponding spaces of rational spline functions.

On the other hand, given $t_1 < \dots < t_{n+1}$ in I such that the corresponding collocation matrix of (u_0, \dots, u_n) is STP, by Theorem 4.3, we deduce that the corresponding collocation matrix of (r_0, \dots, r_n) is also STP and, by considering (4.12) in Theorem 4.3, we can obtain its bidiagonal factorization (4.3) from the corresponding bidiagonal factorization of the collocation matrix of (u_0, \dots, u_n) . It is important to notice that this bidiagonal factorization can be frequently used to perform algebraic calculations and interpolation with HRA.

In Section 4 of [74] (see the article on page 59) we show many examples of weighted φ -transformed systems related to probabilistic distributions. Moreover, in Section 3 of [73] (see the article on pages 43) and Section 5 of [74] we show a class of rational spaces that can be generated by weighted φ -transformed systems. Let us see in the next section the corresponding general class of rational bases.

4.4 A general class of rational bases

This section contains important results obtained in [73] (see the article on page 43).

Let us suppose that $I \subseteq \mathbb{R}$ and $f, g : I \rightarrow \mathbb{R}$ are nonnegative continuous functions. Then, for $n \in \mathbb{N}$, we can define the system (u_0^n, \dots, u_n^n) where:

$$u_k^n(t) = \binom{n}{k} f^k(t) g^{n-k}(t) \text{ such that } t \in I, \quad k = 0, \dots, n. \quad (4.13)$$

For any positive weights $w_i^n, i = 0, \dots, n$, let us define $\omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t)$ and denote by $(\rho_0^n, \dots, \rho_n^n)$ the rational basis described by

$$\rho_i^n(t) = w_i^n \frac{1}{\omega^n(t)} u_i^n(t), \quad i = 0, \dots, n, \quad (4.14)$$

where (u_0^n, \dots, u_n^n) is defined in (4.13). Clearly, this system spans a space of rational functions with denominator $\omega^n(t)$,

$$\mathcal{R}^n = \text{span}\{\rho_i^n(t) \mid i = 0, \dots, n\} = \{u(t)/\omega^n(t) \mid u(t) \in \mathcal{U}^n\}, \quad (4.15)$$

where \mathcal{U}^n is the space generated by (u_0^n, \dots, u_n^n) .

The following result corresponds to Corollary 4 of [73] and provides the conditions characterizing that the system given in (4.14) has optimal shape preserving properties.

Proposition 4.1. *The system of functions given in (4.14) is the normalized B-basis of the space \mathcal{R}^n defined in (4.15) if and only if the function f/g defined on $I_0 = \{t \in I \mid g(t) \neq 0\}$ is increasing and satisfies*

$$\inf \left\{ \frac{f(t)}{g(t)} \mid t \in I_0 \right\} = 0, \quad \sup \left\{ \frac{f(t)}{g(t)} \mid t \in I_0 \right\} = +\infty. \quad (4.16)$$

Given $I = [a, b]$ and $f, g : I \rightarrow \mathbb{R}$ satisfying conditions of Proposition 4.1, let us consider the system (u_0^n, \dots, u_n^n) defined in (4.13) and its generated space \mathcal{U} . For any $t_0 \in (a, b]$, such that $f(t_0) > 0$, let us define $I' := [a, t_0]$ and the system $(\tilde{u}_0^n, \dots, \tilde{u}_n^n)$ given by

$$\tilde{u}_i^n(t) := \binom{n}{i} \tilde{f}^i(t) \tilde{g}^{n-i}(t), \quad t \in I', \quad i = 0, \dots, n, \quad (4.17)$$

where

$$\tilde{f}(t) := \frac{f(t)}{f(t_0)}, \quad t \in I', \quad \tilde{g}(t) := \frac{f(t_0)g(t) - g(t_0)f(t)}{f(t_0)}, \quad t \in I'.$$

In [70] it is shown that the system (4.17) is a B-basis of the space $\mathcal{U}|_{I'}$ formed by the restrictions to I' of the functions of \mathcal{U} . On the other hand, the matrix L such that

$$(u_0^n(t), \dots, u_n^n(t)) := (\tilde{u}_0^n(t), \dots, \tilde{u}_n^n(t))L, \quad t \in I', \quad (4.18)$$

is nonsingular, lower triangular and TP.

Now, let us consider positive weights w_i^n , $i = 0, \dots, n$. By (4.18) we can write

$$(u_0^n(t), \dots, u_n^n(t)) (w_0^n, \dots, w_n^n)^T = (\tilde{u}_0^n(t), \dots, \tilde{u}_n^n(t))L (w_0^n, \dots, w_n^n)^T, \quad t \in I'. \quad (4.19)$$

Taking into account (4.19), it can be deduced that the weights \tilde{w}_i^n , $i = 0, \dots, n$, obtained by

$$(\tilde{w}_0^n, \dots, \tilde{w}_n^n)^T := L (w_0^n, \dots, w_n^n)^T \quad (4.20)$$

satisfy

$$\omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t) = \sum_{i=0}^n \tilde{w}_i^n \tilde{u}_i^n(t), \quad t \in I'. \quad (4.21)$$

Then, by Theorem 4.2, we can deduce that the system $(\tilde{\rho}_0^n, \dots, \tilde{\rho}_n^n)$ defined by

$$\tilde{\rho}_i^n(t) := \tilde{w}_i^n \frac{1}{\omega^n(t)} \tilde{u}_i^n(t), \quad t \in I', \quad i = 0, \dots, n, \quad (4.22)$$

is the normalized B-basis of the space $\mathcal{R}^n|_{I'}$ formed by the restrictions to I' of the functions of the rational space \mathcal{R}^n in (4.15). Clearly, by defining the diagonal matrices $D := \text{diag}\{w_0^n, \dots, w_n^n\}$ and $\tilde{D} := \text{diag}\{\tilde{w}_0^n, \dots, \tilde{w}_n^n\}$, we have

$$\begin{aligned} (\rho_0^n(t), \dots, \rho_n^n(t)) &= \frac{1}{\omega^n(t)} (u_0^n(t), \dots, u_n^n(t))D = \frac{1}{\omega^n(t)} (\tilde{u}_0^n(t), \dots, \tilde{u}_n^n(t))LD \\ &= \frac{1}{\omega^n(t)} (\tilde{u}_0^n(t), \dots, \tilde{u}_n^n(t))\tilde{D}\tilde{D}^{-1}LD = (\tilde{\rho}_0^n(t), \dots, \tilde{\rho}_n^n(t))\tilde{D}^{-1}LD, \quad t \in I'. \end{aligned}$$

Therefore, $\tilde{L} := \tilde{D}^{-1}LD$ is the change of basis matrix such that

$$(\rho_0^n(t), \dots, \rho_n^n(t)) := (\tilde{\rho}_0^n(t), \dots, \tilde{\rho}_n^n(t))\tilde{L}, \quad t \in I'.$$

In [70] a de Casteljaun-like algorithm (called B-algorithm) providing exact evaluation and subdivision for parametric curves $\gamma(t) := \sum_{i=0}^n P_i u_i^n(t)$, $t \in I$ is proposed. Now, exploiting the results in [70] and using the factorization of the matrix L in terms of bidiagonal matrices, we can obtain the bidiagonal factorization of the change of basis matrix \tilde{L} and derive this kind of algorithm for the evaluation and subdivision of the rational curve $\rho(t) := \sum_{i=0}^n P_i \rho_i^n(t)$, $t \in I$. The computed points $P_0^0, P_1^1, \dots, P_n^n$ satisfy

$$\rho(t) := \sum_{i=0}^n P_i^i \tilde{\rho}_i^n(t), \quad t \in I',$$

and, in particular, $P_n^n = \rho(t_0)$.

Algorithm 1: Evaluation and left subdivision algorithm**for j :=0 to n**

$$d_j^0 := w_j^n, \quad P_j^0 := P_j$$

for i :=0 to n-1**for j :=0 to i**

$$P_j^{i+1} := P_j^i$$

for j:=i+1 to n

$$d_j^{i+1} := g(t_0)d_{j-1}^i + f(t_0)d_j^i$$

$$P_j^{i+1} := g(t_0)\frac{d_{j-1}^i}{d_j^{i+1}}P_{j-1}^i + f(t_0)\frac{d_j^i}{d_j^{i+1}}P_j^i$$

Let us observe that if $(w_0^n u_0^n, \dots, w_n^n u_n^n)$ is normalized then $\omega^n(t) = 1$ and the algorithm reduces to the Algorithm 5.1 in [70] for the evaluation of non rational curves defined in terms of a normalized B-basis.

Similarly, we can deduce the following algorithm for the evaluation and right subdivision of the rational curve $\rho(t) = \sum_{i=0}^n P_i \rho_i^n(t)$, $t \in I$.

Algorithm 2: Evaluation and Right subdivision algorithm**for j :=0 to n**

$$d_j^0 := w_j^n, \quad P_j^0 := P_j$$

for i :=0 to n-1**for j:=0 to n-i-1**

$$d_j^{i+1} := g(t_0)d_j^i + f(t_0)d_{j+1}^i$$

$$P_j^{i+1} := g(t_0)\frac{d_j^i}{d_j^{i+1}}P_j^i + f(t_0)\frac{d_{j+1}^i}{d_j^{i+1}}P_{j+1}^i$$

for j :=n-i to n

$$P_j^{i+1} := P_j^i$$

In Figure 4.1 we illustrate an example of the de Casteljaou-like algorithm to subdivide the rational trigonometric curves $\gamma(t) := \sum_{i=0}^n P_i \rho_i^n(t)$ at a given parameter $t_0 \in (-\Delta, \Delta)$, $0 < \Delta < \pi/2$ by considering

$$f(t) := \sin\left(\frac{\Delta+t}{2}\right), \quad g(t) := \sin\left(\frac{\Delta-t}{2}\right), \quad t \in I = [-\Delta, \Delta].$$

In order to illustrate these facts, we have implemented a Matlab application for obtaining the Casteljaou-like algorithm to subdivide rational curves at a given parameter t_0 by considering polynomial, trigonometric and hyperbolic functions f and g satisfying the conditions of Proposition 4.1. The application can be found and downloaded at the following website: <https://github.com/CAGD2020/General>.

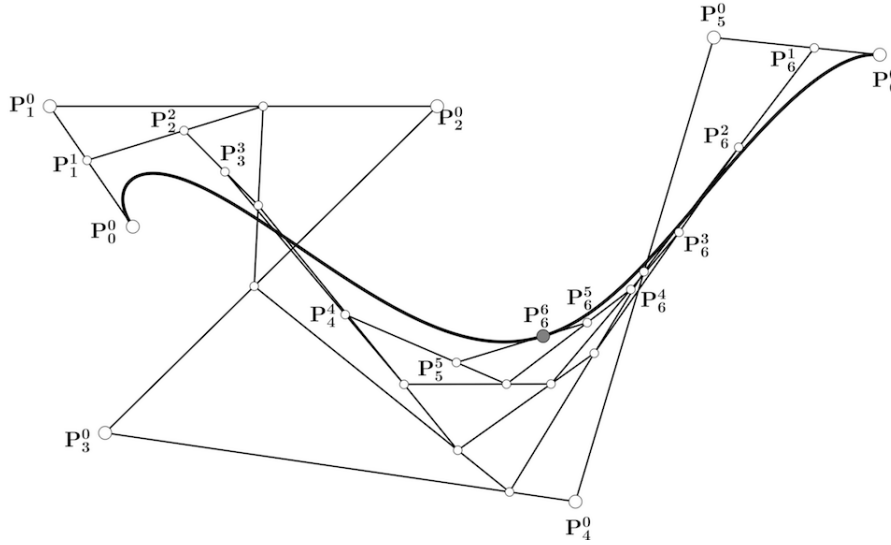


Figure 4.1: de Casteljau-like algorithm to subdivide the rational trigonometric curves at $t_0 = 1/5$, $f(t) := \sin\left(\frac{1+t}{2}\right)$, $g(t) := \sin\left(\frac{1-t}{2}\right)$ and $w_i^n = [15, 5, 5, 5, 5, 15, 5, 15]$.

4.5 A particular class of rational bases satisfying recurrence relations

This section contains important results obtained in [73] (see the article on page 43). Let us consider nested spaces generated by a particular class of rational bases admitting degree elevation and de Casteljau-type evaluation algorithms. Following the approach of [97], let us consider linear factors $L_i(t) = a_i g(t) + b_i f(t)$ defined by positive values a_i and b_i , $i \in \mathbb{Z}_+$, and

$$\widehat{\omega}^n(t) := L_1(t) \cdot \dots \cdot L_n(t). \quad (4.23)$$

It can be easily checked that $\widehat{\omega}^n(t) = \sum_{i=0}^n \widehat{w}_i^n u_i^n(t)$ where

$$\widehat{w}_i^n = \frac{1}{\binom{n}{i}} \left(\sum_{\substack{K \cup L = \{1, \dots, n\} \\ |K| = (n-i), |L| = i}} \prod_{k \in K} a_k \prod_{l \in L} b_l \right). \quad (4.24)$$

The positivity of a_i and b_i guarantees that $\widehat{w}_i^n > 0$ and $\widehat{\omega}^n(t) > 0$, $\forall t \in I$. Let us denote by $(\widehat{\rho}_0^n, \dots, \widehat{\rho}_n^n)$ the particular class of rational bases

$$\widehat{\rho}_i^n(t) := \widehat{w}_i^n \frac{1}{\widehat{\omega}^n(t)} u_i^n(t), \quad i = 0, \dots, n. \quad (4.25)$$

This basis spans the space of rational functions with denominator $\widehat{\omega}^n(t)$,

$$\widehat{\mathcal{R}}^n := \text{span}\{\widehat{\rho}_i^n(t) \mid i = 0, \dots, n\} = \{u(t)/\widehat{\omega}^n(t) \mid u(t) \in \mathcal{U}^n\},$$

where \mathcal{U}^n is the space generated by the basis (4.13).

Proposition 4.1 gives conditions that guarantee that these particular class of rational bases $(\widehat{\rho}_0^n, \dots, \widehat{\rho}_n^n)$ defined in (4.25) are of interest in CAGD and have optimal shape preserving properties. Let us observe that Proposition 2 of [97] establishes recurrence relations satisfied by the weights (4.24). Taking into

account these recurrence relations and replacing in Propositions 3 and 4 of [97] the functions t and $1-t$ by $f(t)$ and $g(t)$, respectively, one can easily deduce the following relations satisfied by these bases

$$\begin{aligned}\widehat{\rho}_i^n(t) &= a_n \frac{g(t)}{L_n(t)} \widehat{\rho}_i^{n-1}(t) + b_n \frac{f(t)}{L_n(t)} \widehat{\rho}_{i-1}^{n-1}(t), \quad i = 0, \dots, n, \\ \widehat{\rho}_i^n(t) &= a_{n+1} \frac{n+1-i}{n+1} \frac{\widehat{w}_i^n}{\widehat{w}_i^{n+1}} \widehat{\rho}_i^{n+1}(t) + b_{i+1}^{n+1} \frac{i+1}{n+1} \frac{\widehat{w}_i^n}{\widehat{w}_{i+1}^{n+1}} \widehat{\rho}_{i+1}^{n+1}(t), \quad i = 0, \dots, n.\end{aligned}$$

These relations guarantee the nested nature of the generated spaces, i.e. $\mathcal{R}^{n-1} \subset \mathcal{R}^n$, and allow the definition of degree elevation and de Casteljau-type algorithms for the evaluation of curves. In addition, Section 4 of [97] also explains a nice geometric interpretation for the influence of the initial coefficients a_i and b_i to the shape of the generated parametric curves.

In [73] we also present evaluation and subdivision algorithms for this particular class of rational bases. In order to illustrate these facts, we have implemented a Matlab application for obtaining the Casteljau-like algorithm to subdivide this particular class rational curves at a given parameter t_0 by considering polynomial, trigonometric and hyperbolic functions f and g satisfying the conditions of Proposition 4.1. The application can be found and downloaded at the following website: <https://github.com/CAGD2020/Particular>.

4.6 Bidiagonal factorization of rational bases with high relative accuracy

This section contains important results obtained in [74] (see the article on page 59).

Theorem 2 of [71] proves that, given nonnegative $f, g : I \rightarrow \mathbb{R}$ such that $f(t) \neq 0, g(t) \neq 0, \forall t \in (a, b)$ and f/g is a strictly increasing function, then

$$A := \left(\binom{n}{j-1} f^{j-1}(t_i) g^{n-j+1}(t_i) \right)_{1 \leq i, j \leq n+1}, \quad a < t_1 < \dots < t_{n+1} < b, \quad (4.26)$$

is STP. Moreover, in Theorem 3 of [71], the following bidiagonal decomposition (4.3) of the collocation matrices (4.26) was deduced

$$A = F_n F_{n-1} \dots F_1 D G_1 \dots G_{n-1} G_n, \quad (4.27)$$

where F_i and G_i , $1 \leq i \leq n$, are the lower and upper triangular bidiagonal matrices of the form (4.4) and $D = \text{diag}(p_{1,1}, \dots, p_{n+1,n+1})$. The entries $m_{i,j}$, $\widehat{m}_{i,j}$ and $p_{i,i}$ are given by

$$\begin{aligned}m_{i,j} &= \frac{g^{n-j+1}(t_i) g(t_{i-j})}{g^{n-j+2}(t_{i-1})} \frac{\prod_{k=1}^{j-1} (f(t_i) g(t_{i-k}) - f(t_{i-k}) g(t_i))}{\prod_{k=2}^j (f(t_{i-1}) g(t_{i-k}) - f(t_{i-k}) g(t_{i-1}))}, \\ \widehat{m}_{i,j} &= \frac{n-i+2}{i-1} \frac{f(t_j)}{g(t_j)}, \quad 1 \leq j < i \leq n+1, \\ p_{i,i} &= \binom{n}{i-1} \frac{g^{n-i+1}(t_i)}{\prod_{k=1}^{i-1} g(t_k)} \prod_{k=1}^{i-1} (f(t_i) g(t_k) - f(t_k) g(t_i)), \quad 1 \leq i \leq n+1.\end{aligned} \quad (4.28)$$

According to Theorem 4.3, the collocation matrix \tilde{A} of the rational basis defined in (4.14) corresponding to $a < t_1 < \dots < t_{n+1} < b$ is STP and can be factorized as

$$\tilde{A} = \tilde{F}_n \tilde{F}_{n-1} \dots \tilde{F}_1 \tilde{D} \tilde{G}_1 \dots \tilde{G}_{n-1} \tilde{G}_n, \quad (4.29)$$

where \tilde{F}_i and \tilde{G}_i , $1 \leq i \leq n$, are the lower and upper triangular bidiagonal matrices of the form (4.11) and $\tilde{D} = \text{diag}(q_{1,1}, \dots, q_{n+1,n+1})$. The off-diagonal entries $r_{i,j}$, $\hat{r}_{i,j}$ of \tilde{F}_i and \tilde{G}_i , respectively, and the diagonal entries $q_{i,i}$ of \tilde{D} are

$$\begin{aligned} r_{i,j} &= \frac{\omega^n(t_{i-1})}{\omega^n(t_i)} m_{i,j}, & \hat{r}_{i,j} &= \frac{w_{i-1}^n}{w_{i-2}^n} \hat{m}_{i,j}, & 1 \leq j < i \leq n+1, \\ q_{i,i} &= \frac{w_{i-1}^n}{\omega^n(t_i)} p_{i,i}, & 1 \leq i \leq n+1, \end{aligned} \quad (4.30)$$

where ω^n and w_i^n are defined in (4.23) and (4.24), respectively, and $m_{i,j}$, $\hat{m}_{i,j}$, $p_{i,i}$ are the entries given in (4.28).

Let us observe that if the evaluation of f and g does not include subtractions (except for the initial data) and the computation of (4.28) can be performed with HRA, then rational basis (4.14) guarantee excellent computational properties since many algebraic computations associated to \tilde{A} can be performed with HRA. Let us notice that we can obtain the same results with the particular case of rational bases defined in (4.25), where the weights satisfy (4.24). In Section 4.8 we are going to illustrate accurate computations with their corresponding collocation matrices.

Let us show some examples in the following section. Let us see several choices of functions f and g satisfying the conditions of Proposition 4.1 and allowing that the corresponding rational basis (4.14) is the normalized B-basis of its generated space. Let us also see their corresponding bidiagonal factorizations with HRA.

4.7 Interesting examples of rational bases

Let us first suppose that f is a strictly increasing function on $[a, b]$ such that $0 < f(t) < 1$ for all $t \in (a, b)$. Let us consider the function $g(t) := 1 - f(t)$. Clearly, $g(t) > 0$ on (a, b) and f/g is a strictly increasing function on (a, b) . Observe that the corresponding basis $(\rho_0^n, \dots, \rho_n^n)$ defined in (4.14) spans the space

$$\mathcal{R}^n = \text{span}\{u(t)/\omega^n(t) \mid u(t) \in \mathcal{U}^n, \omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t)\},$$

where the system (u_0, \dots, u_n) is given in (4.13).

Taking into account that

$$f(t_i)g(t_j) - f(t_j)g(t_i) = f(t_i) - f(t_j),$$

we can deduce that its corresponding collocation matrix is STP and its bidiagonal factorization (4.29) is given by

$$\begin{aligned} r_{i,j} &= \frac{\omega^n(t_{i-1})}{\omega^n(t_i)} \frac{(1-f(t_i))^{n-j+1} (1-f(t_{i-j}))}{(1-f(t_{i-1}))^{n-j+2}} \frac{\prod_{k=1}^{j-1} (f(t_i) - f(t_{i-k}))}{\prod_{k=2}^j (f(t_{i-1}) - f(t_{i-k}))}, & 1 \leq j < i \leq n+1, \\ \tilde{r}_{i,j} &= \frac{w_{i-1}^n}{w_{i-2}^n} \frac{n-i+2}{i-1} \frac{f(t_j)}{1-f(t_j)}, & 1 \leq j < i \leq n+1, \\ p_{i,i} &= \frac{w_{i-1}^n}{\omega^n(t_i)} \binom{n}{i-1} \frac{(1-f(t_i))^{n-i+1}}{\prod_{k=1}^{i-1} (1-f(t_k))} \prod_{k=1}^{i-1} (f(t_i) - f(t_k)), & 1 \leq i \leq n+1. \end{aligned} \quad (4.31)$$

We can deduce that a sufficient condition to obtain the bidiagonal decomposition (4.31) of the collocation matrix of the rational basis $(\rho_0^n, \dots, \rho_n^n)$ defined in (4.14) with HRA and the computations of its inverse, eigenvalues, singular values and the solution of some associated linear systems $Ax = b$ with HRA holds if we can compute the expressions $f(t_i) - f(t_j)$ for all $j < i$ with HRA. Let us now consider some particular examples. For the particular choice

$$f(t) := t^2, \quad g(t) := 1 - t^2, \quad t \in [0, 1],$$

the corresponding rational basis $(\rho_0^n, \dots, \rho_n^n)$ defined in (4.14) spans the space

$$\mathcal{R}^n = \text{span}\{u(t)/\omega^n(t) \mid u(t) \in \mathcal{U}^n, \omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t)\},$$

where the system (u_0, \dots, u_n) given in (4.13) spans the space $\langle 1, t^2, \dots, t^{2n} \rangle$ of even polynomials defined on $[0, 1]$ of degree less than or equal to $2n$. Moreover, in this case,

$$f(t_i) - f(t_j) = t_i^2 - t_j^2 = (t_i + t_j)(t_i - t_j)$$

and we can obtain the bidiagonal decomposition (4.31) of the collocation matrix of the rational basis $(\rho_0^n, \dots, \rho_n^n)$ defined in (4.14) with HRA because in the computation of multipliers and pivots of the Neville elimination we only perform subtractions with initial data.

Let us now consider another particular case given by the trigonometric functions

$$f(t) := \sin^2(t/2) = \frac{1 - \cos(t)}{2}, \quad g(t) := \cos^2(t/2) = \frac{1 + \cos(t)}{2}, \quad t \in [0, \pi].$$

The corresponding rational basis $(\rho_0^n, \dots, \rho_n^n)$ defined in (4.14) spans the space

$$\mathcal{R}^n = \text{span}\{u(t)/\omega^n(t) \mid u(t) \in \mathcal{U}^n, \omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t)\}, \quad (4.32)$$

where the system (u_0, \dots, u_n) given in (4.13) is the normalized B-basis of the space

$$\mathcal{U}^n = \langle 1, \cos(t), \cos(2t), \dots, \cos(nt) \rangle$$

of even trigonometric polynomials on $[0, \pi]$ (see [92]).

In this case, we can obtain the bidiagonal factorization (4.31) of the collocation matrix of the rational basis $(\rho_0^n, \dots, \rho_n^n)$ defined in (4.14) taking into account that

$$f(t_i) - f(t_j) = (\cos(t_j) - \cos(t_i))/2.$$

The computation with HRA of the pivots and multipliers of the Neville elimination is not guarantee. However, in Section 4.8 we are going to show numerical experiments that illustrate the high accuracy for solving algebraic problems.

Let us now consider a different choice of functions f and g . Let $0 < \Delta < \pi/2$ and

$$f(t) := \sin\left(\frac{\Delta+t}{2}\right), \quad g(t) := \sin\left(\frac{\Delta-t}{2}\right), \quad t \in I = [-\Delta, \Delta]. \quad (4.33)$$

Let us notice that the functions f and g clearly satisfy $f(t) > 0$ and $g(t) > 0$ for all $t \in (-\Delta, \Delta)$. Moreover, it can be checked that

$$\left(\frac{f(t)}{g(t)}\right)' = \left(\frac{\sin\left(\frac{\Delta+t}{2}\right)}{\sin\left(\frac{\Delta-t}{2}\right)}\right)' = \frac{1}{2} \frac{\sin(\Delta)}{\sin^2\left(\frac{\Delta-t}{2}\right)} > 0, \quad \forall t \in (-\Delta, \Delta).$$

Therefore, for any $0 < \Delta < \pi/2$, f/g is a strictly increasing function on $(-\Delta, \Delta)$ and thus f and g satisfy conditions of Proposition 4.1. The corresponding rational basis $(\rho_0^n, \dots, \rho_n^n)$ defined in (4.14) spans the space

$$\mathcal{R}^n = \text{span}\{u(t)/\omega^n(t) \mid u(t) \in \mathcal{U}^n, \omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t)\},$$

where, for a given $n = 2m$, the system (u_0^n, \dots, u_n^n) given in (4.13) is a basis that coincides, up to a positive scaling, with the basis with optimal shape preserving properties of the space

$$\mathcal{U}^n = \langle 1, \cos(t), \sin(t), \dots, \cos(mt), \sin(mt) \rangle$$

of trigonometric polynomials of degree less than or equal to m on I (see Section 3 of [96]).

Taking into account that

$$f(t_i)g(t_j) - f(t_j)g(t_i) = \sin(\Delta) \sin((t_i - t_j)/2),$$

we can deduce that the collocation matrix of the rational basis $(\rho_0^n, \dots, \rho_n^n)$ defined in (4.14) is STP and its bidiagonal factorization (4.29) is given by

$$\begin{aligned} r_{i,j} &= \frac{\omega^n(t_{i-1})}{\omega^n(t_i)} \frac{\sin^{n-j+1}\left(\frac{\Delta-t_i}{2}\right) \sin\left(\frac{\Delta-t_{i-j}}{2}\right)}{\sin^{n-j+2}\left(\frac{\Delta-t_{i-1}}{2}\right)} \frac{\prod_{k=1}^{j-1} \sin\left(\frac{t_i-t_{i-k}}{2}\right)}{\prod_{k=2}^j \sin\left(\frac{t_{i-1}-t_{i-k}}{2}\right)}, \quad 1 \leq j < i \leq n+1, \\ \tilde{r}_{i,j} &= \frac{w_{i-1}^n}{w_{i-2}^n} \frac{n-i+2}{i-1} \frac{\sin\left(\frac{\Delta+t_j}{2}\right)}{\sin\left(\frac{\Delta-t_j}{2}\right)}, \quad 1 \leq j < i \leq n+1, \\ q_{i,i} &= \frac{w_{i-1}^n}{\omega^n(t_i)} \binom{n}{i-1} \frac{\sin^{n-i+1}\left(\frac{\Delta-t_i}{2}\right) \sin^{i-1}(\Delta)}{\prod_{k=1}^{i-1} \sin\left(\frac{\Delta-t_k}{2}\right)} \prod_{k=1}^{i-1} \sin\left(\frac{t_i-t_k}{2}\right), \quad 1 \leq i \leq n+1. \end{aligned} \quad (4.34)$$

Although we cannot guarantee the computation of the bidiagonal computation (4.34) with HRA in Section 4.8 we are going to show numerical experiments that illustrate the high accuracy for solving algebraic problems.

Finally, we are going to consider hyperbolic functions. Let $\Delta > 0$ and

$$f(t) := \sinh\left(\frac{\Delta+t}{2}\right), \quad g(t) := \sinh\left(\frac{\Delta-t}{2}\right), \quad t \in I = [-\Delta, \Delta].$$

Clearly, $f(t) > 0$ and $g(t) > 0$ for all $t \in (-\Delta, \Delta)$. Moreover, it can be checked that

$$\left(\frac{f(t)}{g(t)}\right)' = \left(\frac{\sinh\left(\frac{\Delta+t}{2}\right)}{\sinh\left(\frac{\Delta-t}{2}\right)}\right)' = \frac{1}{2} \frac{\sinh \Delta}{\sinh^2\left(\frac{\Delta-t}{2}\right)} > 0, \quad \forall t \in (-\Delta, \Delta).$$

Therefore, for any $\Delta > 0$, f/g is a strictly increasing function on $(-\Delta, \Delta)$ and f and g satisfy conditions of Proposition 4.1. The corresponding rational basis $(\rho_0^n, \dots, \rho_n^n)$ defined in (4.14) spans the space

$$\mathcal{R}^n = \text{span}\{u(t)/\omega^n(t) \mid u(t) \in \mathcal{U}^n, \omega^n(t) = \sum_{i=0}^n w_i^n u_i^n(t)\},$$

where, for a given $n = 2m$, the system (u_0^n, \dots, u_n^n) given in (4.13) is a B-basis of the space

$$\mathcal{U}^n = \langle 1, e^t, e^{-t}, \dots, e^{mt}, e^{-mt} \rangle$$

of hyperbolic polynomials of degree less than or equal to m on I . Taking into account that

$$f(t_i)g(t_j) - f(t_j)g(t_i) = \sinh(\Delta) \sinh((t_i - t_j)/2).$$

we can deduce that the collocation matrix of the rational basis defined in (4.14) is STP and its bidiagonal factorization (4.29) is given by

$$\begin{aligned} r_{i,j} &= \frac{\omega^n(t_{i-1})}{\omega^n(t_i)} \frac{\sinh^{n-j+1}\left(\frac{\Delta-t_i}{2}\right) \sinh\left(\frac{\Delta-t_{i-j}}{2}\right)}{\sinh^{n-j+2}\left(\frac{\Delta-t_{i-1}}{2}\right)} \frac{\prod_{k=1}^{j-1} \sinh\left(\frac{t_i-t_{i-k}}{2}\right)}{\prod_{k=2}^j \sinh\left(\frac{t_{i-1}-t_{i-k}}{2}\right)}, \quad 1 \leq j < i \leq n+1, \quad (4.35) \\ \tilde{r}_{i,j} &= \frac{w_{i-1}^n}{w_{i-2}^n} \frac{n-i+2}{i-1} \frac{\sinh\left(\frac{\Delta+t_j}{2}\right)}{\sinh\left(\frac{\Delta-t_j}{2}\right)}, \quad 1 \leq j < i \leq n+1, \\ q_{i,i} &= \frac{w_{i-1}^n}{\omega^n(t_i)} \binom{n}{i-1} \frac{\sinh^{n-i+1}\left(\frac{\Delta-t_i}{2}\right) \sinh^{i-1}(\Delta)}{\prod_{k=1}^{i-1} \sinh\left(\frac{\Delta-t_k}{2}\right)} \prod_{k=1}^{i-1} \sinh\left(\frac{t_i-t_k}{2}\right), \quad 1 \leq i \leq n+1. \end{aligned}$$

Although we cannot guarantee the computation of the bidiagonal factorization given in (4.35) with HRA the numerical experiments of Section 4.8 will show high accuracy for solving the considered algebraic problems.

4.8 Accurate computations with collocation matrices of rational bases

The numerical experiments of this section use the rational bases presented in [74] and [73]. Specifically, we use the particular rational bases given in (4.14). Moreover, these numerical experiments are a part of the numerical experiments presented in [74] (see the article on page 59).

In [62], assuming that the multipliers and diagonal pivots of the Neville elimination of a nonsingular $n \times n$ TP matrix A and its transpose are known with HRA, Koev presents algorithms for computing with HRA its eigenvalues, singular values and the solution of linear systems of equations $Ax = c$ where the entries of the vector c have alternating signs. In [63] Koev implemented these algorithms with the Matlab or Octave functions `TNSolve`, `TNEigenvalues` and `TNSingularvalues`. The computational cost of the function `TNSolve` is $\mathcal{O}(n^2)$ elementary operations and it requires as input arguments the bidiagonal factorization (4.3) of the matrix A and the vector c of the linear system $Ax = c$. The computational cost of `TNEigenvalues` and `TNSingularvalues` is $\mathcal{O}(n^3)$.

Using the results in this chapter, we have implemented the Matlab function `TNBDA` for the efficient computation of the corresponding bidiagonal decomposition (4.3) of the collocation matrices at t_1, \dots, t_{n+1} of the weighted φ -transformed systems. In order to use the functions available in the library `TNTool` of [63], the implemented Matlab function give the bidiagonal decomposition (4.3) for the corresponding matrices by means of the $(n+1) \times (n+1)$ matrix $BD(\cdot)$ defined in (4.5). Observe that the computational complexity of the computation of the multipliers $m_{i,j}$, $\tilde{m}_{i,j}$ and the pivots $p_{i,i}$ of the proposed bidiagonal decomposition is $\mathcal{O}(n^2)$.

Now we include some numerical experiments considering collocation matrices of the particular rational bases given in (4.14). For different values of n we have considered collocation matrices at equidistant parameters in the interior of the interval domain of the particular rational basis (4.14), obtained by

considering $a_i = 2$ and $b_i = 5$, $i \in \mathbb{N}$. Tables 4.1 and 4.2 illustrate the 2-norm condition number of all considered matrices, computed with the Mathematica command `Norm[A, 2] · Norm[Inverse[A], 2]`. Observe that the condition number of the matrices considerably increases with their dimension. Due to the ill conditioning of these matrices, traditional methods do not achieve accurate solutions when solving the mentioned algebraic problems. The numerical results show this fact and the high accuracy of the algorithms that we have presented, even when the bidiagonal factorization of A is not computed with HRA.

n+1	$\kappa(\mathbf{A}_n)$	$\kappa(\mathbf{A}_n)$
10	1.7×10^5	1.4×10^3
20	1.2×10^{11}	4.7×10^6
25	1.1×10^{14}	2.8×10^8
50	6.2×10^{28}	1.9×10^{17}

Table 4.1: Condition number of collocation matrices of the particular rational bases with $f(t) = t$, $g(t) = 1 - t$ (left), with $f(t) = t^2$, $g(t) = 1 - t^2$ (right).

n+1	$\kappa(\mathbf{A}_n)$	$\kappa(\mathbf{A}_n)$	$\kappa(\mathbf{A}_n)$
10	1.1×10^4	3.1×10^5	9.9×10^4
20	1.3×10^9	4.2×10^{11}	4.5×10^{10}
25	5.0×10^{11}	4.9×10^{14}	3.1×10^{13}
50	8.6×10^{24}	1.2×10^{30}	5.9×10^{27}

Table 4.2: Condition number of collocation matrices of of the particular rational bases with $f(t) = \sin^2(t/2)$, $g(t) = \cos^2(t/2)$ (left), with $f(t) = \sin((1+t)/2)$, $g(t) = \sin((1-t)/2)$ (middle) with $f(t) = \sinh((1+t)/2)$, $g(t) = \sinh((1-t)/2)$ (right).

Linear systems arise when solving interpolation problems. So, in this section, we shall illustrate the accuracy of the computed solutions of $Ax = c$ when using the function `TNSolve` with the bidiagonal factorization of A given by `TNBDA`. We have obtained the solution of the systems using Mathematica with a precision of 100 digits and considered this solution exact. Then we have computed with Matlab two approximations, the first one using `TNBDA` and `TNSolve` and the second one using the Matlab command `\`. We have computed the relative error of every approximation $\tilde{c} = (\tilde{c}_1, \dots, \tilde{c}_{n+1})$ of the solution c of the linear system by means of the formula $e = \|c - \tilde{c}\|_2 / \|c\|_2$.

We have considered $\mathbf{c}_n = ((-1)^{i+1}c_i)_{1 \leq i \leq n+1}$ where c_i is a nonnegative random real number. Table 4.3 shows the relative errors when $f(t) = t$, $g(t) = 1 - t$ and the relative errors when $f(t) = t^2$, $g(t) = 1 - t^2$, $t \in [0, 1]$. As we have seen in Section 4.7, in both cases the parameters (4.30) of the bidiagonal factorization (4.29) can be obtained with HRA and then $\mathbf{A}_n x = \mathbf{c}_n$ can also be solved with HRA. The numerical experiments confirm this fact.

Finally, Table 4.4 shows the relative errors in the solution of $\mathbf{A}_n x = \mathbf{c}_n$ with other functions f and g . As we have seen in Section 4.7, in these cases the computation with HRA of the parameters (4.30) of the bidiagonal factorization (4.29) should require the evaluation with HRA of the involved trigonometric or hyperbolic functions. Although this cannot be guaranteed, the numerical experiments show again that accurate algebraic computations with the collocation matrices associated to these non-polynomial basis functions can be performed.

In [74] we show more numerical experiments with these rational bases. Moreover, we consider other collocation matrices of weighted φ -transformed systems such as the collocation matrices of $(n +$

$n+1$	$\mathbf{A}_n \setminus \mathbf{c}_n$	TNBDA	$\mathbf{A}_n \setminus \mathbf{c}_n$	TNBDA
10	3.3×10^{-13}	1.1×10^{-15}	2.4×10^{-14}	9.2×10^{-16}
20	1.2×10^{-9}	6.3×10^{-16}	5.6×10^{-11}	2.4×10^{-15}
25	4.1×10^{-7}	2.1×10^{-15}	3.1×10^{-9}	2.0×10^{-15}
50	1.0	7.5×10^{-15}	1.0	6.4×10^{-15}

Table 4.3: Relative errors solving $\mathbf{A}_n x = \mathbf{c}_n$ with $f(t) = t$, $g(t) = 1 - t$ (left), with $f(t) = t^2$, $g(t) = 1 - t^2$ (right).

$n+1$	$\mathbf{A}_n \setminus \mathbf{c}_n$	TNBDA	$\mathbf{A}_n \setminus \mathbf{c}_n$	TNBDA	$\mathbf{A}_n \setminus \mathbf{c}_n$	TNBDA
10	1.5×10^{-14}	3.2×10^{-16}	2.6×10^{-13}	1.8×10^{-15}	5.1×10^{-14}	1.9×10^{-15}
20	2.3×10^{-12}	5.3×10^{-16}	3.4×10^{-9}	1.5×10^{-15}	2.8×10^{-9}	2.2×10^{-15}
25	3.5×10^{-11}	3.1×10^{-15}	1.9×10^{-6}	2.4×10^{-15}	2.0×10^{-7}	2.9×10^{-15}
50	3.1×10^{-3}	3.2×10^{-15}	1.0	2.1×10^{-14}	1.0	8.1×10^{-15}

Table 4.4: Relative errors solving $\mathbf{A}_n x = \mathbf{c}_n$ with $f(t) = \sin^2(t/2)$, $g(t) = \cos^2(t/2)$ (left), with $f(t) = \sin((1+t)/2)$, $g(t) = \sin((1-t)/2)$ (middle) with $f(t) = \sinh((1+t)/2)$, $g(t) = \sinh((1-t)/2)$ (right).

1)-dimensional negative binomial bases, geometric bases and Poisson bases at equidistant parameters in $(0, 1)$. The code of the experimentation can be found and downloaded at the following website: <https://github.com/NLAA2020>.

4.9 Curve Fitting with Neural Networks using a general class of rational bases

This section presents the main results obtain in [39] (see the article on page 77).

The problem of obtaining an approximating curve from a given set of data points appears recurrently in several applied and industrial domains, such as CAD/CAM systems, computer graphics and animation, medicine, and many others. Although the Bernstein bases and B-spline bases are usually applied to tackle this issue, some shapes cannot yet be adequately approximated by using the polynomial scheme. In this Section we address this limitation by applying the rational bases presented in [73] which are defined in (4.14). The generalization of the rational Bernstein bases obtained when replacing the linear polynomial factors by trigonometric or hyperbolic functions or their mixtures with polynomials have been analyzed in Section 4.4. The generated rational curves inherit geometric properties and algorithms of the traditional rational Bézier curves and so, they can be considered as modeling tools in CAD/CAM systems.

However, the rational curves have an added difficulty. In general, the selection of the weights is not clear and is difficult to find algorithms for this purpose. Moreover, the effect of changing a weight is different from that of moving a control point. Thus, we propose a method for learning the process of curve fitting through the general class of rational bases defined in (4.14). In fact, the approximation is achieved by finding suitable weights and control points to fit the given set of data points using a neural network and a training algorithm, called AdaMax algorithm, which is a first-order gradient-based stochastic optimization.

Specifically, the problem that we solve can be stated as follows. Suppose that f and g are functions defined on $[a, b]$ satisfying the conditions of Proposition 4.1. Consider a set of parameters $a \leq t_0 < \dots <$

$t_\ell \leq b$ and a sequence of data points $s_0, \dots, s_\ell \in \mathbb{R}^k$, where each parameter t_i is associated with a data point s_i . For some $n \leq \ell$, we want to obtain a rational curve

$$c(t) = \sum_{i=0}^n \frac{w_i^n \binom{n}{i} f^i(t) g^{n-i}(t)}{\sum_{i=0}^n w_i^n \binom{n}{i} f^i(t) g^{n-i}(t)} P_i, \quad t \in [a, b], \quad (4.36)$$

to approximate the set of data points $s = (s_i)_{i=0}^\ell$. Therefore, the goal is to obtain the weights w_0^n, \dots, w_n^n and the control points P_0, \dots, P_n of the rational curve (4.36) that best fits the set of data points. In order to compute them, we have used a stochastic optimization process to train a neural network that models the rational curve $c(t)$.

The problem to be solved can be interpreted then as a regression problem where the set of labeled samples is composed of the input data X , that is, the set of parameters $a \leq t_0 < \dots < t_\ell \leq b$ and the target set of data points $Y = s = (s_i)_{i=0}^\ell$.

Then, the expression in (4.36) can be represented as a hierarchical computational graph with just one hidden layer that we will denote as $\mathcal{N}_{w,P} : \mathbb{R} \rightarrow \mathbb{R}^k$ where the computations are organized as in Figure 4.2. The obtained curve $\mathcal{N}_{w,P}(t)$ is the rational curve $c(t)$ that approximates the given set of data points and we denote as the fitting curve.

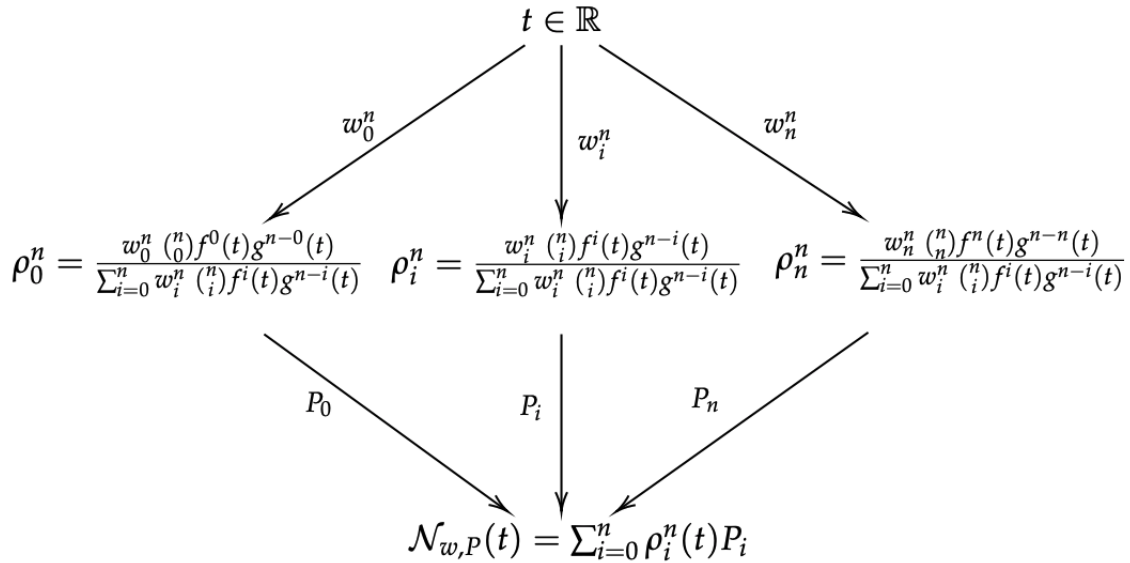


Figure 4.2: From top to bottom. The input layer has the parameter $t \in \mathbb{R}$ as input. The hidden layer is of width $n + 1$ and its parameters are the weights. Then, the output layer computes the approximation of the target curve and its parameters are the control points.

The key idea is to iteratively change the input weights $w = (w_i^n)_{i=0}^n$ and control points $P = (P_i)_{i=0}^\ell$ of the active curve $\mathcal{N}_{w,P}(t)$ and so, it deforms towards the target shape represented by the set of data points $s = (s_i)_{i=0}^\ell$ (see Figure 4 of [39] on page 3.3). Then, we apply an adaptive learning rate optimization algorithm to train the neural network to find the weights and control points, which can be, for example, the Adaptive Moment Estimation (Adam) algorithm or its variant Adaptive Moment Estimation Maximum (AdaMax) algorithm based on infinity norm. These methods are used for stochastic optimization, to solve the supervised learning problem and to find the parameters where a minima is located.

We have mainly used the AdaMax variant because of its stability and simplicity [59]. However, the Adam method can be useful depending on the shape of the set of data points to be approximated and the choice of the loss function. The stochastic objective function, also called the loss function, measures the goodness of the fitting curve. Let us notice that there exist different loss functions such as the mean absolute error, the cross entropy loss, the mean squared error, among others different loss functions implemented in Tensorflow can be consulted in the tensorflow documentation), that can be chosen depending on the problem. In our case, we have considered the mean absolute error as the loss function because of the choice of the training algorithm, given by the following expression: $E(w, P) = \sum_{i=0}^{\ell} |s_i - \mathcal{N}_{w,P}(t_i)| / (\ell + 1)$.

The Adam and the AdaMax algorithms are stochastic gradient-based optimization algorithms and, as previously mentioned, they update the weights and the control points iteratively. The step size is a real number that measures how much the weights and the control points are updated upon each iteration. Besides, the Adam algorithm uses the first and the second moment estimate to update the weights and the control points which are updated following exponential decay rates (β_1 and β_2). Finally, as AdaMax is a variation of Adam using infinity norm, the second moment estimate has a simple recursive formula which will be denoted in Algorithm 1 as exponentially weighted infinity norm. See [59] for a detailed description of the both Adam and AdaMax algorithms.

Algorithm 1: The AdaMax algorithm [59] adapted to our context.

Result: A set of weights w and control points P .

Require: The number of iterations k or an upper bound $e \in \mathbb{R}$ for $E(w, P)$;

Require: The stepsize α ;

Require: The exponential decay rates $\beta_1, \beta_2 \in [0, 1)$;

Require: The stochastic objective function $E(w, P)$;

Require: A small constant ε for numerical stability;

Initialize: Time step $d := 0$;

Initialize: The set of weights and control points in time step $d = 0$, $w^{(0)}$ and $P^{(0)}$ randomly sampled;

Initialize: First moment vector $\gamma^{(0)} := 0$;

Initialize: Exponentially weighted infinity norm $\delta^{(0)} := 0$;

while $d < k$ or $E(w, P) > e$ **do**

$d := d + 1$ (Increment the time step);

$\gamma^{(d)} := \beta_1 \cdot \gamma^{(d-1)} + (1 - \beta_1) \cdot \nabla_{w^{(d-1)}, P^{(d-1)}} E(w^{(d-1)}, P^{(d-1)})$ (Update the biased first moment estimation);

$\delta^{(d)} := \max(\beta_2 \cdot \delta^{(d-1)}, \nabla_{w^{(d-1)}, P^{(d-1)}} E(w^{(d-1)}, P^{(d-1)}))$ (Update the exponentially weighted infinity norm);

$w^{(d)} := w^{(d-1)} - \frac{\alpha}{1 - \beta_1^d} \cdot \frac{\gamma^{(d-1)}}{\delta^{(d-1)} + \varepsilon}$ (Update the weights);

$P^{(d)} := P^{(d-1)} - \frac{\alpha}{1 - \beta_1^d} \cdot \frac{\gamma^{(d-1)}}{\delta^{(d-1)} + \varepsilon}$ (Update the control points);

end

The number of units (i.e. weights and control points) is a hyperparameter and is determined based on the complexity of the shape to be approximated. Besides, the step size, α , can be changed depending on the state of the convergence of the training procedure, for example, when the loss values (i.e., the evaluation of the loss function) gets stuck or the update of the parameters is too big. Then, it is useful to increase or reduce, respectively, the step size according to the values of the loss function. In

[39] (see the article on page 3.3) we show a detailed description of the obtained neural network.

Let us see the performance of the of the neural network $\mathcal{N}_{w,P}$ with different sets of data points $s = (s_i)_{i=0}^{\ell}$ which reflect the variety of situations where the proposed neural network can be applied. The first set of data points belongs to a closed conic curve, the second one belongs to a transcendental curve and the third one is a curve with a twisted shape

In all cases, we have used the rational bases given in (4.14) taking different functions f and g satisfying the conditions of Proposition (4.1) and allowing that the corresponding rational bases (4.14) have the optimal shape preserving properties.

We can see in Table 4.5 a summary of the loss values from different fitting curves. Let us observe that the value n is the degree of the fitting curve and it depends on the complexity of the shape to be approximated. Moreover, let us notice that the proposed neural network is able to obtain a suitable accuracy with low degrees and, as a generalization of other methods, we can choose, depending on the shape of the set of data points, the basis that best fits. Note that in CAGD it is important to face properly the problem of curve fitting finding a balance between accuracy and degree of the curve since high degree curves are computationally expensive to evaluate. The AdaMax algorithm has been selected because it is a computationally efficient with little memory requirements algorithm, suited for problems with large data or parameters. In Table 4.6, the time of execution of the Algorithm 1 using different number of units (i.e., weights and control points) and number of iterations is provided.

n	Basis 1	Basis 2	Basis 3	Basis 4
Circle				
3	$3.3946 \cdot 10^{-2}$	$3.7129 \cdot 10^{-2}$	$7.0468 \cdot 10^{-2}$	$3.6438 \cdot 10^{-2}$
4	$2.1757 \cdot 10^{-3}$	$1.5338 \cdot 10^{-2}$	$3.1678 \cdot 10^{-3}$	$2.5582 \cdot 10^{-3}$
5	$1.7333 \cdot 10^{-4}$	$9.2269 \cdot 10^{-3}$	$2.8083 \cdot 10^{-4}$	$2.2488 \cdot 10^{-3}$
Cycloid				
8	$1.0849 \cdot 10^{-3}$	$3.6855 \cdot 10^{-4}$	$3.6017 \cdot 10^{-4}$	$3.1674 \cdot 10^{-4}$
9	$4.6163 \cdot 10^{-4}$	$3.6855 \cdot 10^{-4}$	$3.6017 \cdot 10^{-4}$	$2.4914 \cdot 10^{-4}$
10	$3.3944 \cdot 10^{-4}$	$3.6855 \cdot 10^{-4}$	$3.6017 \cdot 10^{-4}$	$2.4914 \cdot 10^{-4}$
Archimedean spiral				
11	$1.5982 \cdot 10^{-3}$	$1.0474 \cdot 10^{-2}$	$2.2349 \cdot 10^{-2}$	$7.8109 \cdot 10^{-4}$
12	$1.5982 \cdot 10^{-3}$	$7.8916 \cdot 10^{-3}$	$5.7801 \cdot 10^{-3}$	$7.8109 \cdot 10^{-4}$
13	$1.4106 \cdot 10^{-3}$	$5.2853 \cdot 10^{-3}$	$5.7801 \cdot 10^{-3}$	$7.8109 \cdot 10^{-4}$

Table 4.5: Loss values of the mean absolute error for different fitting curves of degree n with $f(t) = t$, $g(t) = 1 - t$, $t \in [0, 1]$ (Basis 1), $f(t) = t^2$ and $g(t) = 1 - t^2$, $t \in [0, 1]$, (Basis 2), $f(t) = \sin((\Delta + t)/2)$ and $g(t) = \sin((\Delta - t)/2)$, $\Delta < \pi/2$, $t \in [-\Delta, \Delta]$, (Basis 3) and finally $f(t) = \sinh((\Delta + t)/2)$ and $g(t) = \sinh((\Delta - t)/2)$, $\Delta < \pi/2$, $t \in [-\Delta, \Delta]$, (Basis 4). They were all trained with 4000 iterations, $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-7}$. The process was repeated 5 times, being the loss values provided the best value reached.

The implementation was developed using TensorFlow 2.0 [1] allowing developers to easily use it to build and deploy Machine Learning powered applications. The code of the experimentation can be found and downloaded at the following website:

<https://github.com/Mathematics2020>. All experiments were ran on a Quad-Core Intel Core i7 CPU, 2,8GHz with 16GB RAM. In [39] (see the article on page 3.3) we present a detailed description of these experiments. Moreover, in [39] we compare the neural network with the traditional least square method to test its performance with a noisy set of data points.

n+1	Number of iterations				
	1	25	50	100	3000
5	0.1259	1.4284	2.8381	5.7259	189.5386
10	0.0989	2.0781	4.1325	10.2672	268.8726
15	0.1244	2.7142	5.3781	10.9886	347.6139
50	0.6479	8.2589	13.3576	27.4398	850.6713
100	1.1624	14.3999	32.6576	65.3298	1521.3971

Table 4.6: Time of execution of the proposed algorithm measured in seconds for different number of units and iterations. The values provided are the mean of 5 repetitions with a set of data points of size 100.

4.10 Accurate least squares fitting with a general class of rational bases

In Section 4.3 we have obtained algorithms for the computation of the bidiagonal decomposition (4.29) of square collocation matrices of the general class of rational bases given in (4.14) (see also the article [74] on page 59).

Now, following the approach of [84] for a polynomial case and taking into account the obtained results in [72] and [25], we generalize the aforementioned bidiagonal decompositions to the case of rectangular collocation matrices.

The problem that we would like to solve is stated as follows. Suppose that f and g are functions defined on $[a, b]$ satisfying the conditions of Proposition 4.1. Consider a set of parameters $a < t_1 < \dots < t_{\ell+1} < b$ and a sequence of data points $s_1, \dots, s_{\ell+1} \in \mathbb{R}^k$, where each parameter t_i is associated with a data point s_i . For some $n \leq \ell$, we want to compute a rational curve

$$c(t) = \sum_{i=1}^{n+1} \frac{w_i^n \binom{n}{i-1} f^{i-1}(t) g^{n-i+1}(t)}{\sum_{i=1}^{n+1} w_i^n \binom{n}{i-1} f^{i-1}(t) g^{n-i+1}(t)} P_i, \quad t \in [a, b], \quad (4.37)$$

minimizing the sum of the squares of the deviations from the set of data points $s = (s_i)_{i=1}^{\ell+1}$, that is, $f = \sum_{i=1}^{\ell+1} (s_i - c(t_i))^2$. In order to compute the control points $P = (P_i)_{i=1}^{n+1}$ of the fitting curve, we have to solve, in the least square sense, the overdetermined linear system $AP = s$, where the matrix A is

$$A = \left(\frac{w_i^n \binom{n}{i-1} f^{i-1}(t_j) g^{n-i+1}(t_j)}{\sum_{i=1}^{n+1} w_i^n \binom{n}{i-1} f^{i-1}(t_j) g^{n-i+1}(t_j)} \right)_{1 \leq i \leq n+1; 1 \leq j \leq \ell+1}.$$

According to Theorem 4.3, A is STP and so has maximal rank $n + 1$. Therefore this problem has a unique solution, which is given by the solution of the linear system

$$A^T A P = A^T s.$$

Solving the previous normal equations is a worse conditioned problem than computing the solution through the QR decomposition of the coefficient matrix A , which is the usual approach. In [62] an efficient algorithm for computing the QR decomposition of an STP matrix A is presented. In [23] the Matlab or Octave library TNQR, containing an implementation of the mentioned last algorithm, is available. Assuming that the bidiagonal factorization of A is known, TNQR computes the matrix Q and the bidiagonal factorization of the matrix R with HRA. Now, following the approach of [84], we shall describe how to solve our least squares problem by means of a bidiagonal decomposition for rectangular matrices that generalizes the bidiagonal factorization described, for the square case, in the previous section and the QR decomposition provided by TNQR.

In order to compute the solution of the least squares problem, we define the $(l+1) \times (n+1)$ matrix M such that

$$\begin{aligned} M_{i,i} &:= q_{i,i}, & i = 1, \dots, n+1, \\ M_{i,j} &:= r_{i,j}, & j = 1, \dots, n+1; \quad i = j+1, \dots, l+1, \\ M_{i,j} &:= \hat{r}_{i,j}, & i = 1, \dots, n; \quad j = i+1, \dots, n+1, \end{aligned}$$

where the $r_{i,j}$, $\hat{r}_{i,j}$ and $q_{i,i}$ are obtained as in (4.30). Then, using TNQR, we can obtain the QR decomposition of A such that

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix},$$

where $Q \in \mathbb{R}^{(l+1) \times (l+1)}$ is an orthogonal matrix and $R \in \mathbb{R}^{(n+1) \times (n+1)}$ is an upper triangular matrix with positive diagonal entries. Following Section 1.3.1 in [8], the solution of the least squares problem is obtained from

$$\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = Q^T f, \quad Rc = d_1, \quad r = Q \begin{pmatrix} 0 \\ d_2 \end{pmatrix}, \quad (4.38)$$

where $d_1 \in \mathbb{R}^{n+1}$, $d_2 \in \mathbb{R}^{l-n}$ and $r = f - Ac$. The matrices Q and R have an special structure described in [35]. In particular, R is nonsingular and TP. In order to obtain the solution of the upper triangular system $Rc = d_1$, we have used the routine `TNSolve` of [62], which uses the bidiagonal decomposition of the upper triangular TP matrix R .

In order to illustrate these facts, we have implemented a Matlab application. In this application we show the performance of this algorithm with different set of data points belonging to known curves which reflect the variety of situations where the proposed algorithm can be applied. In all cases, we uses the rational bases given in (4.14) taking different functions f and g satisfying the conditions of Proposition (4.1) and allowing that the corresponding rational bases (4.14) have the optimal shape preserving properties. The application gives the corresponding fitting curve defined in (4.37) and requieres as inputs the set of data points and weights. It can be found and downloaded at the following website: <https://github.com/AppCurveFittingG>. Let us notice that we can obtain the same results with the particular case of rational bases defined in (4.25), where the weights satisfy (4.24). For this case, we have also implemented a Matlab application that can be found and downloaded at the following website: <https://github.com/AppCurveFittingP>.

For future work we would like to tackle the problem raised in Section 4.9 and Section 4.10. We wish to design a new neural network that will be trained with an optimization algorithm to update the weights while the control points are obtained with the accurate algorithm presented in Section 4.10.

5

Total positivity and accurate computations with Wronskian matrices of monomial, exponential and Jacobi polynomials

ABOUT THIS CHAPTER

The purpose of this chapter is to justify the thematic unit of the articles [75] and [76] (see on pages 99 and 117), which belong to the compendium of publications of this thesis. The main results of [75] and [76] are also presented.

[75] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with Wronskian matrices, *Calcolo* **58**, 1 (2021).

[76] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with collocations and Wronskian matrices of Jacobi polynomials, *Journal of Scientific Computing* **87**, 77 (2021).

5.1 Introduction

The accuracy of the calculations is a desirable goal in Computational Mathematics. Let us recall that an algorithm can be performed with high relative accuracy (HRA) if it does not include subtractions of numbers having the same sign (except of the initial data if they are exact), that is, if it only includes products, divisions, additions of numbers of the same sign and subtractions of the initial data having the same sign provided that they are not affected by errors (cf. [23]). For some structured classes of matrices such algorithms have been found through an adequate parameterization of the matrix. In particular, this has been achieved for some subclasses of totally positive (TP) matrices. In [62] it was shown that, given the bidiagonal factorization of a nonsingular TP matrix A with HRA, we can compute with HRA its eigenvalues and singular values, the matrix A^{-1} and even the solution of $Ax = b$ for vectors b with alternating signs. Among the subclasses of TP matrices for which the bidiagonal factorization has been obtained with HRA (cf. [15], [17], [82], [86]), there are many examples of collocation matrices $(u_{j-1}(t_i))_{1 \leq i, j \leq n+1}$ of systems (u_0, \dots, u_n) of functions defined on a real subset I ($t_1 < t_2 < \dots < t_{n+1}$ in I). However, up to now, there are no examples of accurate computations for matrices involving derivatives of the basis functions. This chapter presents some examples of Wronskian matrices for which many algebraic computations can be performed accurately. These Wronskian matrices come from applications in Computer-Aided Geometric Design (CAGD) and they can also arise in Hermite interpolation problems, in particular in Taylor interpolation problems. For example, we provide the

bidiagonal decomposition with HRA of the Wronskian matrix of monomials and an accurate bidiagonal factorization of the Wronskian matrix of exponential polynomials.

This chapter also deals with the accurate computation when using collocation and Wronskian matrices (see Section 5.5) of Jacobi polynomials on $(1, \infty)$. Crucial facts to derive our results have been to prove the strict total positivity of the collocation matrices of Jacobi polynomials on $(1, \infty)$ and the total positivity of their Wronskian matrices. Then the bidiagonal factorization with HRA has been obtained for these matrices and the algorithms presented in [63] can be used for the algebraic computations mentioned above with HRA.

The complexity of all the proposed algorithms of this chapter for solving the mentioned algebraic problems is comparable to that of the traditional LAPACK algorithms, which, as we have illustrated in [75] and [76] (see the articles on pages 99 and 117), deliver no such accuracy.

The layout of the chapter is as follows. Section 5.2 presents basic concepts and results. Section 5.3 shows that the bidiagonal factorization of the Wronskian matrices of the monomial basis of polynomials can be performed with HRA. In Section 5.4 we prove that the Wronskian matrices of the basis of exponential polynomials on positive real numbers are strictly totally positive. We also provide the bidiagonal factorization of these matrices. The computation with HRA of this factorization should require the evaluation with HRA of the involved exponential functions. Although this cannot be guaranteed, the numerical experiments presented in Section 5 of [75] (see the article on page 99) show an accuracy similar to the obtained for the monomial basis. In Section 5.5, the strict total positivity and bidiagonal factorization of the collocation matrices of Jacobi polynomials on $(1, \infty)$ are obtained. In Section 5.6, the total positivity and bidiagonal factorization of the corresponding Wronskian matrices are derived. Section 5.7 particularizes the results for some well known families of Jacobi polynomials: Legendre polynomials, Gegenbauer polynomials, Chebyshev polynomials of the first and second kind and rational Jacobi polynomials.

5.2 Notations and previous results

In this chapter we shall use the following notations. Given an n -times continuously differentiable real function f and $x \in \mathbb{R}$ in its domain, $f'(x)$ denotes the first derivative of f at x . For any $i \leq n$, $f^{(i)}(x)$ denotes the i -th derivative of f at x . Given a basis (u_0, \dots, u_n) of a space of n -times continuously differentiable functions, defined on a real interval I and $x \in I$, the *Wronskian matrix* at x is

$$W(u_0, \dots, u_n)(x) := (u_{j-1}^{(i-1)}(x))_{i,j=1, \dots, n+1}.$$

Let us recall that A matrix is totally positive (TP) if all its minors are nonnegative. Some books with many applications of TP matrices are [2, 27, 93].

Neville elimination is an alternative procedure to Gaussian elimination and has been used to characterize TP matrices. More details on this elimination method can be found in [34, 36, 37]. By Theorem 4.2 and the arguments of p.116 of [37], a nonsingular TP matrix $A = (a_{i,j})_{1 \leq i, j \leq n+1}$ admits a factorization of the form

$$A = F_n F_{n-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n, \quad (5.1)$$

where F_i and G_i are the TP, lower and upper triangular bidiagonal matrices given by

$$F_i = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & m_{i+1,1} & 1 & & \\ & & & \ddots & \ddots & \\ & & & & m_{n+1,n+1-i} & 1 \end{pmatrix}, \quad G_i^T = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \hat{m}_{i+1,1} & 1 & & \\ & & & \ddots & \ddots & \\ & & & & \hat{m}_{n+1,n+1-i} & 1 \end{pmatrix}, \quad (5.2)$$

and $D = \text{diag}(p_{1,1}, p_{2,2}, \dots, p_{n+1,n+1})$ has positive diagonal entries. If, in addition, the entries m_{ij} , \tilde{m}_{ij} satisfy

$$m_{ij} = 0 \Rightarrow m_{hj} = 0, \quad \forall h > i, \quad \text{and} \quad \tilde{m}_{ij} = 0 \Rightarrow \tilde{m}_{ik} = 0, \quad \forall k > j,$$

then the decomposition (5.1) is unique. The diagonal entries $p_{i,i}$ of D are the diagonal pivots of the Neville elimination of A and the elements $m_{i,j}$, $\tilde{m}_{i,j}$ are positive and coincide with the multipliers of the Neville elimination of A and A^T , respectively.

In [61], the bidiagonal factorization (5.1) of an $(n+1) \times (n+1)$ nonsingular and TP matrix A is represented by defining a matrix $BD(A) = (BD(A)_{i,j})_{1 \leq i,j \leq n+1}$ such that

$$BD(A)_{i,j} := \begin{cases} m_{i,j}, & \text{if } i > j, \\ p_{i,i}, & \text{if } i = j, \\ \tilde{m}_{j,i}, & \text{if } i < j. \end{cases} \quad (5.3)$$

Given $BD(A)$, using the results in [34, 36, 37], a bidiagonal decomposition of A^{-1} can be computed as

$$A^{-1} = \tilde{G}_1 \tilde{G}_2 \cdots \tilde{G}_n D^{-1} \tilde{F}_n \cdots \tilde{F}_2 \tilde{F}_1, \quad (5.4)$$

where \tilde{F}_i and \tilde{G}_i , $i = 1, \dots, n$, are the lower and upper triangular bidiagonal matrices of the form of F_i and G_i , respectively, but replacing the off-diagonal entries $\{m_{i+1,1}, \dots, m_{n+1,n+1-i}\}$ and $\{\tilde{m}_{i+1,1}, \dots, \tilde{m}_{n+1,n+1-i}\}$ by $\{-m_{i+1,1}, \dots, -m_{n+1,i}\}$ and $\{-\tilde{m}_{i+1,1}, \dots, -\tilde{m}_{n+1,i}\}$ respectively.

Let us observe that if a matrix A is nonsingular and TP, then A^T is also a nonsingular and TP matrix. Moreover, the bidiagonal decomposition of A^T can be computed as

$$A^T = G_n^T G_{n-1}^T \cdots G_1^T D F_1^T \cdots F_{n-1}^T F_n^T, \quad (5.5)$$

where F_i and G_i , $i = 1, \dots, n$, are the lower and upper triangular bidiagonal matrices in (5.1).

Finally, let us recall that $x \in \mathbb{R}$ is obtained with high relative accuracy (HRA) if the relative error of the computed value \tilde{x} satisfies

$$\frac{\|x - \tilde{x}\|}{\|x\|} < Ku,$$

where K is a positive constant independent of the arithmetic precision and u is the unit round-off. HRA implies that the relative errors of the computations are of the order of the machine precision. An algorithm can be computed with HRA when it only uses products, quotients, sums of numbers of the same sign or subtractions of exact data (cf. [23], [61]).

If the bidiagonal factorization (5.1) of a nonsingular TP matrix A is computed with HRA then, using the algorithms in [62], we can also compute with HRA its eigenvalues and singular values, the matrix A^{-1} and even the solution of $Ax = b$ for vectors b with alternating signs.

5.3 Total Positivity and factorization of Wronskian matrices of monomial basis

This section contains important results obtained in [75] (see the article on page 99). The monomial basis of the space \mathbf{P}^n of polynomials of degree less than or equal to n is (m_0, \dots, m_n) with

$$m_i(x) := x^i, \quad i = 0, \dots, n. \quad (5.6)$$

Given $x_0 \in \mathbb{R}$, we can define a Taylor basis (T_0, \dots, T_n) of \mathbf{P}^n by

$$T_i(x) := \frac{(x - x_0)^i}{i!}, \quad i = 0, \dots, n. \quad (5.7)$$

It can be checked that

$$(m_0, \dots, m_n) = (T_0, \dots, T_n)W,$$

where $W := W(m_0, \dots, m_n)(x_0)$. Equivalently, we can also write

$$(T_0, \dots, T_n) = (m_0, \dots, m_n)W^{-1}.$$

In this section we are going to obtain the bidiagonal factorization (5.1) of W and W^{-1} and see that they can be computed with HRA. First let us prove the following auxiliary result.

Lemma 5.1. *Given $i, j \in \mathbb{N}$, then*

$$\frac{1}{i!}m_j^{(i)}(x) = \frac{1}{(i-1)!}m_{j-1}^{(i-1)}(x) + \frac{x}{i!}m_{j-1}^{(i)}(x), \quad x \in \mathbb{R}. \quad (5.8)$$

Proof. See Lemma 1 of [75] (see the article on page 99). □

For a given $x \in \mathbb{R}$, $k, n \in \mathbb{N}$ with $k \leq n$, let $U_{k,n} = (u_{i,j})_{1 \leq i, j \leq n+1}$ be the upper triangular bidiagonal matrix with unit diagonal entries and such that

$$u_{i,i+1} := 0, \quad i = 1, \dots, k-1, \quad u_{i,i+1} := x, \quad i = k, \dots, n. \quad (5.9)$$

In the following result we obtain an explicit expression of the entries of the product matrix $U_{1,n} \cdots U_{n,n}$.

Proposition 5.1. *For a given $x \in \mathbb{R}$ and $n \in \mathbb{N}$, let*

$$U_n := U_{1,n} \cdots U_{n,n},$$

where $U_{k,n}$, $k = 1, \dots, n$, is the upper triangular bidiagonal matrix with unit diagonal entries satisfying (5.9). Then $U_n = (u_{i,j})_{1 \leq i, j \leq n+1}$ is an upper triangular matrix and

$$u_{i,j} = \frac{1}{(i-1)!}m_{j-1}^{(i-1)}(x), \quad 1 \leq i, j \leq n+1. \quad (5.10)$$

Proof. See Proposition 1 of [75] (see the article on page 99). □

Let us observe that for $x > 0$ the matrices $U_{k,n}$, $k = 1, \dots, n$, are TP. Then, as a direct consequence of the previous result and taking into account that, by Theorem 3.1 of [2], the product of TP matrices is TP, we can derive the following result providing a bidiagonal factorization of the Wronskian matrix of the monomial basis (5.6).

Corollary 5.1. *Let $n \in \mathbb{N}$ and (m_0, \dots, m_n) be the monomial basis given in (5.6). Then for any $x \in \mathbb{R}$,*

$$W := W(m_0, \dots, m_n)(x) := \begin{pmatrix} 0! & & & \\ & 1! & & \\ & & \ddots & \\ & & & n! \end{pmatrix} U_{1,n} U_{2,n} \cdots U_{n,n}, \quad (5.11)$$

where $U_{k,n}$, $k = 1, \dots, n$, is the upper triangular bidiagonal matrix with unit diagonal entries satisfying (5.9). Moreover, if $x > 0$ then $W(m_0, \dots, m_n)(x)$ is TP.

Let us observe that (5.11) is the bidiagonal factorization (5.1) of the upper triangular, nonsingular and TP Wronskian matrix $W = W(m_0, \dots, m_n)(x)$, $x > 0$, where F_i and G_i are the TP, lower and upper triangular bidiagonal matrices in (5.2). Clearly $BD(W)$ can be computed with HRA and, consequently, using the bidiagonal factorization (5.4), W^{-1} can also be computed with HRA as stated in the following result.

Proposition 5.2. *Let W be the Wronskian matrix at x_0 of the monomial basis of the space of polynomials \mathbf{P}^n . Then W^{-1} can be computed with HRA.*

Finally, in the following example, we illustrate the bidiagonal factorization (5.11) of the Wronskian matrix of a basis of monomials.

Example 5.1. *For the particular case $n = 3$, the bidiagonal factorization of the Wronskian matrix of the basis (m_0, m_1, m_2, m_3) at $x \in \mathbb{R}$ is*

$$W(m_0, m_1, m_2, m_3)(x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 6 \end{pmatrix} \begin{pmatrix} 1 & x & 0 & 0 \\ 0 & 1 & x & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & x & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

In Section 5 of [75] (see the article on page 99) it can be seen accurate results obtained when computing the eigenvalues, singular values, the inverse and the solutions of some linear systems associated with the Wronskian matrices of monomial bases, using the bidiagonal factorization (5.11) and the algorithms presented in [62] and [63]. The code with the numerical experimentation can be found and downloaded at the following website: <https://github.com/Calcolo2021>.

5.4 Total positivity and factorization of Wronskian matrices of exponential polynomials

This section contains important results obtained in [75] (see the article on page 99). Given $\lambda_0, \dots, \lambda_n$ and $x \in \mathbb{R}$, let us consider the basis (u_0, \dots, u_n) of exponential polynomials defined on \mathbb{R} by

$$u_i(x) := e^{\lambda_i x}, \quad i = 0, \dots, n. \quad (5.12)$$

The following result proves that, if $0 < \lambda_0 < \lambda_1 < \dots < \lambda_n$, the Wronskian matrix of the basis (5.12),

$$W(u_0, \dots, u_n)(x) = (\lambda_{j-1}^{i-1} e^{\lambda_{j-1} x})_{i,j=1, \dots, n+1}, \quad (5.13)$$

is STP for any $x \in \mathbb{R}$.

Theorem 5.1. Let $0 < \lambda_0 < \dots < \lambda_n$ and the basis (5.12) of exponential polynomials. For any $x \in \mathbb{R}$, the corresponding Wronskian matrix (5.13) is STP and

$$\det W(u_0, \dots, u_n)(x) = \prod_{k=0}^n e^{\lambda_k x} \prod_{0 \leq k < \ell \leq n} (\lambda_\ell - \lambda_k). \quad (5.14)$$

Proof. See Theorem 1 of [75] (see the article on page 99). \square

In the following result we present the bidiagonal decomposition (5.1) of the Wronskian matrices (5.13) and their inverses.

Theorem 5.2. Let $0 < \lambda_0 < \dots < \lambda_n$ and the corresponding basis (5.12) of exponential polynomials. For a given $x \in \mathbb{R}$, $W := W(u_0, \dots, u_n)(x)$ admits a factorization of the form

$$W = F_n F_{n-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n, \quad (5.15)$$

where F_i and G_i , $1 \leq i \leq n$, are the lower and upper triangular bidiagonal matrices given by (5.2) and $D = \text{diag}(p_{1,1}, p_{2,2}, \dots, p_{n+1, n+1})$. The entries $m_{i,j}$, $\tilde{m}_{i,j}$ and $p_{i,i}$ are given by

$$m_{i,j} = \lambda_{j-1}, \quad \tilde{m}_{i,j} = e^{(\lambda_{i-1} - \lambda_{i-2})x} \prod_{k=2}^j \frac{(\lambda_{i-1} - \lambda_{i-k})}{(\lambda_{i-2} - \lambda_{i-k-1})}, \quad 1 \leq j < i \leq n+1,$$

$$p_{i,i} = e^{\lambda_{i-1}x} \prod_{k=0}^{i-2} (\lambda_{i-1} - \lambda_k), \quad 1 \leq i \leq n+1.$$

Proof. See Theorem 2 of [75] (see the article on page 99). \square

Let us observe that the computation with HRA of the bidiagonal decomposition (5.15) should require the evaluation with HRA of the involved exponential function. Although this cannot be guaranteed, in Section 5 of [75] it can be seen accurate results obtained when computing their eigenvalues, singular values, inverses or the solutions of some linear systems associated with the Wronskian matrices of the bases of exponential polynomials, using the bidiagonal factorization (5.15) and the algorithms presented in [62] and [63]. The code with the numerical experimentation can be found and downloaded at the following website: <https://github.com/Calcolo2021>.

We finish this section illustrating the bidiagonal factorization (5.15) of the Wronskian matrix of a basis of exponential polynomials.

Example 5.2. For the particular case $n = 2$, the bidiagonal factorization of the Wronskian matrix of the basis $(e^{\lambda_0 x}, e^{\lambda_1 x}, e^{\lambda_2 x})$ at $x \in \mathbb{R}$ is

$$W(e^{\lambda_0 x}, e^{\lambda_1 x}, e^{\lambda_2 x}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \lambda_0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \lambda_0 & 1 & 0 \\ 0 & \lambda_1 & 1 \end{pmatrix} \begin{pmatrix} p_{1,1} & 0 & 0 \\ 0 & p_{2,2} & 0 \\ 0 & 0 & p_{3,3} \end{pmatrix} \begin{pmatrix} 1 & e^{(\lambda_1 - \lambda_0)x} & 0 \\ 0 & 1 & e^{(\lambda_2 - \lambda_1)x} \frac{\lambda_2 - \lambda_1}{\lambda_1 - \lambda_0} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & e^{(\lambda_2 - \lambda_1)x} \\ 0 & 0 & 1 \end{pmatrix},$$

where $p_{1,1} = e^{\lambda_0 x}$, $p_{2,2} = e^{\lambda_1 x}(\lambda_1 - \lambda_0)$ and $p_{3,3} = e^{\lambda_2 x}(\lambda_2 - \lambda_0)(\lambda_2 - \lambda_1)$.

The following sections contain important results obtained in [76] (see the article on page 117).

5.5 Total positivity and factorizations of collocation matrices of Jacobi polynomials

Given $\alpha, \beta \in \mathbb{R}$, the basis of Jacobi polynomials of the space \mathbf{P}^n of polynomials of degree less than or equal to n is $(J_0^{(\alpha, \beta)}, \dots, J_n^{(\alpha, \beta)})$ with

$$J_i^{(\alpha, \beta)}(x) := \frac{\Gamma(\alpha + i + 1)}{i! \Gamma(\alpha + \beta + i + 1)} \sum_{k=0}^i \binom{i}{k} \frac{\Gamma(\alpha + \beta + i + k + 1)}{\Gamma(\alpha + k + 1)} \left(\frac{x-1}{2}\right)^k, \quad i = 0, \dots, n. \quad (5.16)$$

Let us recall that Jacobi polynomials are orthogonal on the interval $[-1, 1]$ with respect to the weight $(1-x)^\alpha(1+x)^\beta$.

Let us consider the lower triangular matrix $A = (a_{ij})_{1 \leq i, j \leq n+1}$ given by

$$a_{i,j} := \begin{cases} \frac{1}{(j-1)!(i-j)!} \prod_{k=j}^{i-1} (\alpha + k) \prod_{k=1}^{j-1} (\alpha + \beta + i + k - 1), & \text{if } i \geq j, \\ 0, & \text{if } i < j. \end{cases} \quad (5.17)$$

It can be checked that

$$(J_0^{(\alpha, \beta)}, \dots, J_n^{(\alpha, \beta)})^T = A(v_0, \dots, v_n)^T, \quad (5.18)$$

where (v_0, \dots, v_n) is the basis of \mathbf{P}^n such that

$$v_i(x) := \left(\frac{x-1}{2}\right)^i, \quad i = 0, \dots, n. \quad (5.19)$$

The following result provides the multipliers and the diagonal pivots of the Neville elimination of the change of basis matrix A described in (5.17) and proves that this matrix is nonsingular and TP.

Theorem 5.3. *Let $A = (a_{ij})_{1 \leq i, j \leq n+1}$ be the lower triangular matrix defined in (5.17). Then the multipliers $m_{i,j}$ and diagonal pivots $p_{i,i}$ of the Neville elimination of A are given by*

$$m_{i,1} := \frac{\alpha + i - 1}{i - 1}, \quad m_{i,j} := \frac{\alpha + \beta + 2i - j}{\alpha + \beta + 2i - j - 2} m_{i,j-1}, \quad 1 \leq j < i \leq n+1, \quad 1 < i \leq n+1, \quad (5.20)$$

$$p_{i,i} := \prod_{r=1}^{i-1} \frac{(\alpha + \beta + 2i - r - 1)}{(i - r)}, \quad 1 \leq i \leq n+1.$$

Moreover, for any $\alpha, \beta > -1$, A is nonsingular and TP.

Proof. See Theorem 2 of [76] (see the article on page 117). □

Corollary 5.2. *Let $A = (a_{ij})_{1 \leq i, j \leq n+1}$ be the lower triangular matrix defined by (5.17). Then, for any $\alpha, \beta > -1$, the matrix A admits a factorization of the form*

$$A = F_n F_{n-1} \cdots F_1 D, \quad (5.21)$$

where F_i , $i = 1, \dots, n$, is the lower triangular, bidiagonal matrix given by (5.2) and $D = \text{diag}(p_{1,1}, p_{2,2}, \dots, p_{n+1, n+1})$. The entries $m_{i,j}$ and $p_{i,i}$ can be obtained from (5.20).

Let us observe that the factorization (5.21) corresponds to $BD(A)$, the bidiagonal factorization (5.1) of A . Furthermore, for any $\alpha, \beta > -1$, $BD(A)$ can be computed with HRA, since it does not require subtractions (except of the initial data).

Remark 5.1. It is well known that the monomial basis $(1, t, \dots, t^n)$ of \mathbf{P}^n is STP on $(0, \infty)$. Moreover, given a sequence of positive parameters $0 < t_0 < \dots < t_n$, the bidiagonal factorization (5.1) of the corresponding STP collocation matrix can be described by

$$\begin{aligned} m_{i,j} &= \frac{\prod_{k=1}^{j-1} (t_i - t_{i-k})}{\prod_{k=2}^j (t_{i-1} - t_{i-k})}, \quad \widehat{m}_{i,j} = t_j, \quad 1 \leq j < i \leq n+1, \\ p_{i,i} &= \prod_{k=1}^{i-1} (t_i - t_k), \quad 1 \leq i \leq n+1 \end{aligned} \quad (5.22)$$

(see [61] or Theorem 3 of [71]). Consequently, the basis (v_0, \dots, v_n) defined in (5.19) is also STP on $(1, \infty)$. Furthermore, given $1 < x_1 < \dots < x_{n+1}$, by considering $t_i := (x_i - 1)/2$, $i = 1, \dots, n+1$, and using the bidiagonal factorization (5.22) for the collocation matrix of the monomial basis at $0 < t_1 < \dots < t_{n+1}$, it can be easily deduced that the bidiagonal decomposition (5.1) of the collocation matrix of (v_0, \dots, v_n) at $x_1 < \dots < x_{n+1}$ is given by:

$$\begin{aligned} m_{i,j} &= \frac{\prod_{k=1}^{j-1} (x_i - x_{i-k})}{\prod_{k=2}^j (x_{i-1} - x_{i-k})}, \quad \widehat{m}_{i,j} = (x_j - 1)/2, \quad 1 \leq j < i \leq n+1, \\ p_{i,i} &= \frac{1}{2^{i-1}} \prod_{k=1}^{i-1} (x_i - x_k), \quad 1 \leq i \leq n+1. \end{aligned} \quad (5.23)$$

The following result proves that, for any $\alpha, \beta > -1$, the collocation matrix of the basis (5.16) of Jacobi polynomials at $1 < x_1 < \dots < x_{n+1}$,

$$M_J := \left(J_{j-1}^{(\alpha, \beta)}(x_i) \right)_{1 \leq i, j \leq n+1}, \quad (5.24)$$

is STP.

Theorem 5.4. Given $\alpha, \beta > -1$, the corresponding basis of Jacobi polynomials defined in (5.16) is STP on $(1, \infty)$.

Proof. See Theorem 3 of [76] (see the article on page 117). □

Remark 5.2. By Section 4 of [61], we can transpose the bidiagonal decomposition (5.21) of the lower triangular and TP matrix A to obtain the corresponding bidiagonal decomposition of A^T (see (5.5)). Clearly, since $BD(A)$ can be computed with HRA, $BD(A^T)$ can be also computed with HRA. Moreover, the collocation matrix of the basis (v_0, \dots, v_n) defined in (5.19) at nodes $1 < x_1 < \dots < x_{n+1}$ is STP and its corresponding bidiagonal decomposition can be obtained with HRA (see (5.23)). If the bidiagonal decompositions of two nonsingular, TP matrices can be computed with HRA, using Algorithm 5.1 of [62], we can also obtain with HRA the bidiagonal decomposition of the nonsingular and TP product matrix. Consequently, we can derive with HRA the bidiagonal matrices (5.2) of the bidiagonal factorization (5.1) of the collocation matrices of Jacobi polynomials and thus, we can also compute with HRA its inverse matrix, its eigenvalues and singular values as well as the solutions of some linear systems.

In Section 6 of [76] (see the article on page 117) it can be seen accurate results obtained when computing the mentioned algebraic problems with the collocation matrices of Jacobi polynomials, using the bidiagonal factorization (5.1) and the algorithms presented in [62] and [63]. The code with the numerical experimentation can be found and downloaded at the following website: <https://github.com/JSC2021>.

5.6 Total positivity and factorizations of Wronskian matrices of Jacobi polynomials

Given $x \in \mathbb{R}$, let $W(J_0^{(\alpha,\beta)}, \dots, J_n^{(\alpha,\beta)})(x)$ be the Wronskian matrix at x of the basis (5.16) of Jacobi polynomials. Using formula (5.18), it can be checked that

$$W(J_0^{(\alpha,\beta)}, \dots, J_n^{(\alpha,\beta)})(x) = W(v_0, \dots, v_n)(x)A^T, \quad (5.25)$$

where $W(v_0, \dots, v_n)(x)$ is the Wronskian matrix of the basis (v_0, \dots, v_n) given in (5.19) and A is the lower triangular matrix defined by (5.17).

In Corollary 5.1 it has been proved that the Wronskian matrix at any positive real value of the monomial basis $(1, x, \dots, x^n)$ of the space of polynomials \mathbf{P}^n is TP on $(0, \infty)$. It was also shown that this Wronskian matrix and its inverse can be computed with HRA. Now we are going to extend these results to the basis (ℓ_0, \dots, ℓ_n) given by

$$\ell_i(x) = (ax + b)^i, \quad x \in \mathbb{R}, \quad i = 0, \dots, n, \quad (5.26)$$

where $a, b \in \mathbb{R}$ with $a > 0$. First let us prove the following auxiliary result.

Lemma 5.2. *The basis (ℓ_0, \dots, ℓ_n) defined in (5.26) satisfies*

$$\frac{1}{a^i i!} \ell_j^{(i)}(x) = \frac{1}{a^{i-1} (i-1)!} \ell_{j-1}^{(i-1)}(x) + \frac{ax+b}{a^i i!} \ell_{j-1}^{(i)}(x), \quad 1 \leq i, j \leq n. \quad (5.27)$$

Proof. See Lemma 1 of [76] (see the article on page 117). □

Now, for a given $x \in \mathbb{R}$, $k, n \in \mathbb{N}$ with $k \leq n$, let $U_{k,n} = (u_{i,j}^{(k)})_{1 \leq i, j \leq n+1}$ be the upper triangular, bidiagonal matrix with unit diagonal entries, such that

$$u_{i,i+1}^{(k)} := 0, \quad i = 1, \dots, k-1, \quad u_{i,i+1}^{(k)} := ax + b, \quad i = k, \dots, n. \quad (5.28)$$

The following result shows that the product matrix $U_{1,n} \cdots U_{n,n}$ coincides, up to a positive scaling, with the Wronskian matrix of $(\ell_0, \ell_1, \dots, \ell_n)$ at x .

Proposition 5.3. *For a given $x \in \mathbb{R}$ and $n \in \mathbb{N}$, let*

$$U_n := U_{1,n} \cdots U_{n,n},$$

where $U_{k,n}$, $k = 1, \dots, n$, are the upper triangular, bidiagonal matrices with unit diagonal entries satisfying (5.28). Then $U_n = (u_{i,j})_{1 \leq i, j \leq n+1}$ is an upper triangular matrix and

$$u_{i,j} = \frac{1}{a^{i-1} (i-1)!} \ell_{j-1}^{(i-1)}(x), \quad 1 \leq i, j \leq n+1. \quad (5.29)$$

Proof. See Proposition 1 of [76] (see the article on page 99). □

As a direct consequence of the previous result, we can provide the bidiagonal factorization (5.1) of the Wronskian matrix of (ℓ_0, \dots, ℓ_n) .

Proposition 5.4. *Let $n \in \mathbb{N}$ and (ℓ_0, \dots, ℓ_n) be the basis given in (5.26). Then, for any $x > -b/a$, the Wronskian matrix $W(\ell_0, \dots, \ell_n)(x)$ is TP and*

$$W(\ell_0, \dots, \ell_n)(x) = \begin{pmatrix} 0! & & & \\ & a^1 1! & & \\ & & \ddots & \\ & & & a^n n! \end{pmatrix} U_{1,n} \cdots U_{n,n}, \quad (5.30)$$

where $U_{k,n}$, $k = 1, \dots, n$, are the upper triangular, bidiagonal matrices with unit diagonal entries satisfying (5.28).

Let us observe that the bidiagonal factorization (5.1) of $W(\ell_0, \dots, \ell_n)(x)$ is given by (5.30). Clearly, this factorization can be computed with HRA for any $x > -b/a$ and, consequently, using (5.4), its inverse matrix can also be computed with HRA as stated in the following result.

Proposition 5.5. *Let W be the Wronskian matrix at x of the basis (ℓ_0, \dots, ℓ_n) given in (5.26). Then W^{-1} can be computed with HRA.*

Now, using Proposition 5.4, we can immediately deduce the following factorization of the Wronskian matrix at $x \in \mathbb{R}$ of the basis (v_0, \dots, v_n) in (5.19),

$$W(v_0, \dots, v_n)(x) := \begin{pmatrix} \frac{1}{2^0} 0! & & & \\ & \frac{1}{2^1} 1! & & \\ & & \ddots & \\ & & & \frac{1}{2^n} n! \end{pmatrix} U_{1,n} \cdots U_{n,n}, \quad (5.31)$$

where $U_{k,n} = (u_{i,j}^{(k)})_{1 \leq i,j \leq n+1}$, $k = 1, \dots, n$, is the upper triangular, bidiagonal matrix with unit diagonal entries satisfying

$$u_{i,i+1}^{(k)} := 0, \quad i = 1, \dots, k-1, \quad u_{i,i+1}^{(k)} := (x-1)/2, \quad i = k, \dots, n. \quad (5.32)$$

Moreover, if $x > 1$, $W(v_0, \dots, v_n)(x)$ is a nonsingular and TP matrix. Then, taking into account (5.25), the fact that A^T is a nonsingular and TP matrix (see Theorem 5.3) and that the product of nonsingular TP matrices is a nonsingular and TP matrix (Theorem 3.1 of [2]), we deduce the following result on the total positivity of the Wronskian matrices of Jacobi polynomials.

Theorem 5.5. *Let $n \in \mathbb{N}$ and $(J_0^{(\alpha,\beta)}, \dots, J_n^{(\alpha,\beta)})$ be the Jacobi polynomial basis given in (5.16). For any $\alpha, \beta > -1$, the Wronskian matrix $W(J_0^{(\alpha,\beta)}, \dots, J_n^{(\alpha,\beta)})(x)$ at $x > 1$ is nonsingular and TP.*

Remark 5.3. *Taking into account (5.5), we can obtain the bidiagonal decomposition (5.21) of the matrix A^T in (5.25). Clearly, since $BD(A)$ can be computed with HRA, $BD(A^T)$ can be also computed with HRA. On the other hand, the Wronskian matrix of the basis (v_0, \dots, v_n) defined in (5.19) is nonsingular and TP at any $x > 1$. Moreover, its corresponding bidiagonal decomposition (5.23) can be obtained with HRA. By Algorithm 5.1 of [62], if the bidiagonal decompositions of two nonsingular and TP matrices can be computed with HRA, then the bidiagonal decomposition of the product matrix can be also obtained with HRA. Consequently, the Wronskian matrix of the basis (5.16) of Jacobi polynomials can be computed with HRA and thus, we can compute with HRA its inverse matrix, its eigenvalues and singular values and the solutions of some linear systems.*

Theorem 5.6. *The basis (L_0, \dots, L_n) of Legendre polynomials, defined by (5.37), is STP on $(1, \infty)$. Given $x_1 < \dots < x_{n+1}$, with $x_1 > 1$, the bidiagonal decomposition (5.1) of the corresponding collocation matrix can be obtained with HRA. Moreover, for any $x > 1$, the Wronskian matrix $W(L_0, \dots, L_n)(x)$ is nonsingular and TP and its bidiagonal decomposition (5.1) can be obtained with HRA.*

Given $\lambda \in \mathbb{R}$, the basis of Gegenbauer polynomials of \mathbf{P}^n is (G_0, \dots, G_n) with

$$G_i^\lambda(x) := \frac{\Gamma(\lambda + 1/2)}{\Gamma(2\lambda)} \frac{\Gamma(i + 2\lambda)}{\Gamma(i + \lambda + 1/2)} J_i^{(\lambda - 1/2, \lambda - 1/2)}(x), \quad i = 0, \dots, n, \quad (5.38)$$

where $(J_0^{(\lambda - 1/2, \lambda - 1/2)}, \dots, J_n^{(\lambda - 1/2, \lambda - 1/2)})$ is the basis of Jacobi polynomials given in (5.16) with $\alpha = \beta = \lambda - 1/2$. By Theorem 5.4 and Remark 5.2, Lemma 5.4 and Remark 5.3, we can deduce the following result.

Theorem 5.7. *For any $\lambda > -1/2$, the basis (G_0, \dots, G_n) of Gegenbauer polynomials, defined by (5.38), is STP on $(1, \infty)$. Given $x_1 < \dots < x_{n+1}$, with $x_1 > 1$, the bidiagonal decomposition (5.1) of the corresponding collocation matrix can be obtained with HRA. Moreover, for any $x > 1$, the Wronskian matrix $W(G_0, \dots, G_n)(x)$ is nonsingular and TP and its bidiagonal decomposition (5.1) can be obtained with HRA.*

The basis (T_0, \dots, T_n) of Chebyshev polynomials of the first kind is defined by

$$T_i(x) := \frac{J_i^{(-1/2, -1/2)}(x)}{J_i^{(-1/2, -1/2)}(1)}, \quad i = 0, \dots, n, \quad (5.39)$$

where $(J_0^{(-1/2, -1/2)}, \dots, J_n^{(-1/2, -1/2)})$ is the basis of Jacobi polynomials given in (5.16) with $\alpha = \beta = -1/2$. Using again Theorem 5.4, Remark 5.2, Lemma 5.4 and Remark 5.3, we can deduce the following result.

Theorem 5.8. *The basis (T_0, \dots, T_n) of Chebyshev polynomials of the first kind, defined by (5.39), is STP on $(1, \infty)$. Given $x_1 < \dots < x_{n+1}$, with $x_1 > 1$, the bidiagonal decomposition (5.1) of the corresponding collocation matrix can be obtained with HRA. Moreover, for any $x > 1$, the Wronskian matrix $W(T_0, \dots, T_n)(x)$ is nonsingular and TP and its bidiagonal decomposition (5.1) can be obtained with HRA.*

The basis (U_0, \dots, U_n) of second kind Chebyshev polynomials is defined by

$$U_i(x) := (i + 1) \frac{J_i^{(-1/2, -1/2)}(x)}{J_i^{(1/2, 1/2)}(1)}, \quad i = 0, \dots, n, \quad (5.40)$$

where $(J_0^{(1/2, 1/2)}, \dots, J_n^{(1/2, 1/2)})$ is the basis of Jacobi polynomials given in (5.16) with $\alpha = \beta = 1/2$. Using again Theorem 5.4, Remark 5.2, Lemma 5.4 and Remark 5.3, we can deduce the following result.

Theorem 5.9. *The basis (U_0, \dots, U_n) of Chebyshev polynomials of second kind, defined by (5.40), is STP on $(1, \infty)$. Given $x_1 < \dots < x_{n+1}$, with $x_1 > 1$, the bidiagonal decomposition (5.1) of the corresponding collocation matrix can be obtained with HRA. Moreover, for any $x > 1$, the Wronskian matrix $W(U_0, \dots, U_n)(x)$ is nonsingular and TP and its bidiagonal decomposition (5.1) can be obtained with HRA.*

In [100], induced by Jacobi polynomials, a new orthogonal system of rational functions was introduced. For given $\alpha, \beta \in \mathbb{R}$, the system $(R_0^{(\alpha, \beta)}, \dots, R_n^{(\alpha, \beta)})$ of rational Jacobi functions is defined by

$$R_i^{(\alpha, \beta)}(x) := J_i^{(\alpha, \beta)}\left(\frac{x-1}{x+1}\right), \quad i = 0, \dots, n, \quad (5.41)$$

where $(J_0^{(\alpha, \beta)}, \dots, J_n^{(\alpha, \beta)})$ is the basis (5.16) of Jacobi polynomials. Using again Theorem 5.4, Remark 5.2, Lemma 5.4 and Remark 5.3, we can deduce the following result.

Theorem 5.10. *For any $\alpha, \beta > -1$, the basis $(R_0^{(\alpha, \beta)}, \dots, R_n^{(\alpha, \beta)})$ of rational Jacobi functions given in (5.41) is STP on $(-\infty, -1)$. Given $x_1 < \dots < x_{n+1}$, with $x_{n+1} < -1$, the bidiagonal decomposition (5.1) of the corresponding collocation matrix can be obtained with HRA.*

Similar results can be deduced by considering the rational counterparts of the basis of Legendre, Gegenbauer and the first and second kind Chebyshev polynomials.

In Section 6 of [76] (see the article on page 117) it can be seen accurate results obtained when computing the eigenvalues, singular values, or the solutions of some linear systems associated with the collocation and Wronskian matrices of all the mentioned orthogonal bases, using their corresponding bidiagonal decompositions and the algorithms presented in [62] and [63]. The code with the numerical experimentation can be found and downloaded at the following website: <https://github.com/JSC2021>.

Part IV

PRESENTATION OF THE LATEST OBTAINED RESULTS

In this part, we present the latest obtained results, which are not included in the articles that belong to the compendium of publications of this thesis.

6

Total positivity and accurate computations with Wronskian matrices of Bessel and Laguerre polynomials

ABOUT THIS CHAPTER

The purpose of this chapter is to present some of the latest results that we have obtained, which are not included in the articles that belong to the compendium of publications of this thesis. It should be noted that, in this chapter, we have taken into account some of the results shown in the article [75] (see on page 99).

6.1 Introduction

An important goal in computational mathematics is finding algorithms with high relative accuracy (HRA) for matrix calculations such as obtaining their eigenvalues, singular values or inverses. After an adequate parametrization of the matrices, this goal has been achieved for the collocation matrices of some important systems of functions. This was obtained for the collocation matrices of Bessel polynomials (see applications in [21], [47] and references in there) and for the collocation matrices of generalized Laguerre polynomials (see [20]). In both cases, the collocation matrices are totally positive (see Section 6.2) and a bidiagonal factorization with HRA was obtained for them. This bidiagonal factorization is the start step to apply the algorithms with HRA of [61, 62, 63].

Algorithms with HRA for the Wronskian matrices of monomials have been obtained in [75] (see in Chapter 4 or, in more detail, in the article on page 43). In this chapter, we obtain the bidiagonal factorization with HRA for the Wronskian matrices of Bessel polynomials as well as for the Wronskian matrices of generalized Laguerre polynomials, which can be used to calculate with HRA their singular values or inverses. The complexity of the proposed algorithms for solving the mentioned algebraic problems is comparable to that of the traditional LAPACK algorithms, which, as we shall illustrate, deliver no such accuracy.

The layout of the chapter is as follows. Section 6.2 presents basic concepts and results. Section 6.3 proves the total positivity of the Wronskian matrices of Bessel polynomials defined on positive real numbers and shows that the mentioned algebraic calculations can be performed with HRA. Section 6.4 deals with the corresponding results for the Wronskian matrices of generalized Laguerre polynomials. Section 6.5 includes numerical examples illustrating the great accuracy of the presented methods for the computation of all singular values, the inverses of the matrices and the solution of some linear systems.

Finally, let us recall that $x \in \mathbb{R}$ is obtained with high relative accuracy (HRA) if the relative error of the computed value \tilde{x} satisfies

$$\frac{\|x - \tilde{x}\|}{\|x\|} < Ku,$$

where K is a positive constant independent of the arithmetic precision and u is the unit round-off. HRA implies that the relative errors of the computations are of the order of the machine precision. An algorithm can be computed with HRA when it only uses products, quotients, sums of numbers of the same sign or subtractions of exact data (cf. [23], [61]).

If the bidiagonal factorization (6.1) of a nonsingular TP matrix A is computed with HRA then, using the algorithms in [62], we can also compute with HRA its eigenvalues and singular values, the matrix A^{-1} and even the solution of $Ax = b$ for vectors b with alternating signs. In the following sections we shall obtain the bidiagonal factorization (6.1) of Wronskian matrices associated to Bessel and Laguerre polynomials, analyzing whether it can be computed with HRA.

6.3 Total positivity and factorization of Wronskian matrices of Bessel polynomials

Let us denote by \mathbf{P}^n the space of polynomials of degree less than or equal to n and (p_0, \dots, p_n) the monomial basis of \mathbf{P}^n such that

$$p_i(x) := x^i, \quad i = 0, \dots, n. \quad (6.5)$$

The following result restates Corollary 1 of [75] providing the bidiagonal factorization (6.1) of the Wronskian matrix $W(p_0, \dots, p_n)(x)$, $x \in \mathbb{R}$.

Proposition 6.1. *Let (p_0, \dots, p_n) be the monomial basis given in (6.5). For any $x \in \mathbb{R}$, the Wronskian matrix $W(p_0, \dots, p_n)(x)$ is nonsingular and can be factorized as follows,*

$$W(p_0, \dots, p_n)(x) = DG_{1,n} \cdots G_{n-1,n-1} G_{n,n}, \quad (6.6)$$

where $D = \text{diag}\{0!, 1!, \dots, n!\}$ and $G_{i,n}$, $i = 1, \dots, n$, are the upper triangular bidiagonal matrix in (6.2) with

$$\tilde{m}_{k,k-i} = x, \quad i+1 \leq k \leq n+1. \quad (6.7)$$

Moreover, if $x > 0$ then $W(p_0, \dots, p_n)(x)$ is nonsingular and TP, its bidiagonal decomposition (6.1) is given by (6.6) and (6.7) and it can be computed with HRA.

Let us recall that the Bessel basis of \mathbf{P}^n is the polynomial system (B_0, \dots, B_n) with

$$B_i(x) := \sum_{k=0}^i \frac{(i+k)!}{2^k (i-k)! k!} x^k, \quad i = 0, \dots, n. \quad (6.8)$$

In [21], the total positivity of the matrix of change of basis between the Bessel polynomial basis (B_0, \dots, B_n) and the monomials (p_0, \dots, p_n) is proved. As a consequence, accurate computations when considering collocation matrices $(B_{j-1}(x_{j-1}))_{1 \leq i, j \leq n+1}$ with $(0 <) x_0 < x_1 < \dots < x_n$ are derived.

Now, let $W(B_0, \dots, B_n)(x)$ be the Wronskian matrix at $x \in \mathbb{R}$ of the basis (6.8) of Bessel polynomials. The following result extends the results in [21] to $W(B_0, \dots, B_n)(x)$ at $x > 0$ and establishes the total positivity of this Wronskian matrix.

Theorem 6.1. *Let (B_0, \dots, B_n) be the Bessel polynomial basis defined in (6.8). For any $x > 0$, the Wronskian matrix $W := W(B_0, \dots, B_n)(x)$ is nonsingular TP and can be computed with HRA. Furthermore, the computation of all the singular values, the inverse of W , as well as the solution of the linear systems $Wx = b$, where $b = (b_0, \dots, b_n)^T$ has alternating signs, can be performed with HRA.*

Proof. It can be checked that

$$(B_0, \dots, B_n)^T = A(p_0, \dots, p_n)^T, \quad (6.9)$$

where (p_0, \dots, p_n) is the monomial basis given in (6.5) and the change of basis matrix $A = (a_{ij})_{1 \leq i, j \leq n+1}$ is lower triangular and satisfies

$$a_{i,j} := \frac{(i+j-2)!}{2^{j-1}(i-j)!(j-1)!}, \quad i \geq j. \quad (6.10)$$

Using formula (6.9), it can be checked that

$$W(B_0, \dots, B_n)(x) = W(p_0, \dots, p_n)(x)A^T, \quad (6.11)$$

where $W(p_0, \dots, p_n)(x)$ is the Wronskian matrix of the monomial basis (p_0, \dots, p_n) and A is the lower triangular matrix described by (6.10).

By Proposition 6.1, $W(p_0, \dots, p_n)(x)$, $x > 0$, is nonsingular and TP and its bidiagonal factorization (6.1) can be computed with HRA. Furthermore, by Theorem 3 of [21], A is a nonsingular TP matrix and admits a factorization of the form

$$A = F_n F_{n-1} \cdots F_1 D, \quad (6.12)$$

where F_i , $i = 1, \dots, n$, are the lower triangular bidiagonal matrices described in (6.2) and $D = \text{diag}(p_{1,1}, \dots, p_{n+1,n+1})$. The entries $m_{i,j}$, $1 \leq j < i \leq n+1$, and $p_{i,i}$, $1 \leq i \leq n+1$, are given by

$$m_{i,j} = \frac{(2i-2)(2i-3)}{(2i-j-1)(2i-j-2)}, \quad p_{i,i} = (2i-3)!!, \quad (6.13)$$

with the following double factorial notation for a positive integer k ,

$$k!! := \prod_{j=0}^{\lfloor (k-1)/2 \rfloor} (k-2j),$$

where $\lfloor (k-1)/2 \rfloor$ is the greatest integer less than or equal to $(k-1)/2$. Clearly, $m_{i,j}$, and $p_{i,i}$ are positive and can be obtained with HRA. The bidiagonal factorization (6.1) of A^T is given by $A^T = DF_1^T \cdots F_{n-1}^T F_n^T$.

On the other hand, $W(B_0, \dots, B_n)(x)$, $x > 0$, is nonsingular and TP since, by (6.11), it can be expressed as the product of two nonsingular TP matrices (see Theorem 3.1 of [2]).

Using Algorithm 5.1 of [62], if the bidiagonal decomposition (6.1) of two nonsingular TP matrices is provided with HRA, then the corresponding bidiagonal decomposition (6.1) of the product is computed with HRA. Consequently, the bidiagonal decomposition (6.1) of $W = W(B_0, \dots, B_n)(x)$, $x > 0$, can be computed with HRA. This fact guarantees that algebraic problems such as the computation of all the singular values, the inverse matrix of W , and the solution of the linear systems $Wx = b$, where $b = (b_0, \dots, b_n)^T$ has alternating signs, can be performed with HRA (see Section 3 of [23]). \square

Let us recall that the basis of reverse Bessel polynomials in \mathbf{P}^n is (R_0, \dots, R_n) with

$$R_i(x) := \sum_{k=0}^i \frac{(i+k)!}{2^k(i-k)!k!} x^{i-k}, \quad i = 0, \dots, n. \quad (6.14)$$

Let us observe that this basis is obtained when reversing the order of the coefficients of the Bessel polynomials $B_i(x)$, $i = 0, \dots, n$, in (6.8).

In [21] it is proved that the matrix of change of basis between the reverse Bessel polynomials (R_0, \dots, R_n) and the monomials (p_0, \dots, p_n) is TP. Therefore, accurate computations with collocation matrices $(R_{j-1}(x_{j-1}))_{1 \leq i, j \leq n+1}$ where $(0 <)x_0 < x_1 < \dots < x_n$ are provided.

Given $x \in \mathbb{R}$, $W(R_0, \dots, R_n)(x)$ denotes the Wronskian matrix at x of the basis (6.14) of reverse Bessel polynomials. The following result extends the results in [21] to $W(R_0, \dots, R_n)(x)$ at $x > 0$ and establishes the total positivity of this Wronskian matrix.

Theorem 6.2. *Let (R_0, \dots, R_n) be the reverse Bessel polynomial basis given in (6.14). For any $x > 0$, the Wronskian matrix $W_R := W(R_0, \dots, R_n)(x)$ is nonsingular TP and can be computed with HRA. Furthermore, the computation of all the singular values, the inverse of W_R , as well as the solution of the linear systems $W_R x = b$, where $b = (b_0, \dots, b_n)^T$ has alternating signs, can be performed with HRA.*

Proof. It can be checked that

$$(R_0, \dots, R_n)^T = C(p_0, \dots, p_n)^T, \quad (6.15)$$

where (p_0, \dots, p_n) is the monomial basis given in (6.5) and $C = (c_{ij})_{1 \leq i, j \leq n+1}$ is the lower triangular change of basis matrix such that

$$c_{i,j} = \frac{(2i-j-1)!}{2^{i-j}(j-1)!(i-j)!}, \quad i \geq j. \quad (6.16)$$

By formula (6.15) we can write

$$W(R_0, \dots, R_n)(x) = W(p_0, \dots, p_n)(x)C^T, \quad (6.17)$$

where $W(p_0, \dots, p_n)(x)$ is the Wronskian matrix of (p_0, \dots, p_n) at x and C is the lower triangular matrix described by (6.16).

Let us recall that, by Proposition 6.1, $W(p_0, \dots, p_n)(x)$, $x > 0$, is nonsingular and TP and its bidiagonal factorization (6.1) can be computed with HRA. On the other hand, by Theorem 5 of [21], the matrix C is nonsingular and TP and admits a factorization

$$C = F_n F_{n-1} \cdots F_1 D, \quad (6.18)$$

where F_i , $i = 1, \dots, n$, are the lower triangular bidiagonal matrices described in (6.2) and $D = \text{diag}(p_{1,1}, \dots, p_{n+1,n+1})$. The entries $m_{i,j}$ and $p_{i,i}$ are given by

$$m_{i,j} = 2i - 2j - 1, \quad 1 \leq j < i \leq n+1, \quad p_{i,i} = 1, \quad 1 \leq i \leq n+1, \quad (6.19)$$

and, clearly, can be obtained with HRA. The bidiagonal factorization (6.1) of C^T is given by $C^T = DF_1^T \cdots F_{n-1}^T F_n^T$.

Since $W(R_0, \dots, R_n)(x)$, $x > 0$, is the product of two nonsingular TP matrices, by (6.17), we deduce that it is nonsingular and TP (see Theorem 3.1 of [2]).

Using Algorithm 5.1 of [62], if the bidiagonal decomposition (6.1) of two nonsingular TP matrices is provided with HRA, then the corresponding bidiagonal decomposition (6.1) of the product is computed with HRA. Consequently, the bidiagonal decomposition (6.1) of $W(R_0, \dots, R_n)(x)$, $x > 0$, can be computed with HRA and so, their inverse matrix, their singular values and the solutions of the mentioned linear systems (see Section 3 of [23]). \square

Section 6.5 shows accurate results obtained when solving the mentioned algebraic problems using the bidiagonal factorization (6.1) and the algorithms presented in [62] and [63].

6.4 Total positivity and factorization of Wronskian matrices of Laguerre polynomials

Given $\alpha > -1$, the generalized Laguerre basis of \mathbf{P}^n is the polynomial system $(L_0^{(\alpha)}, \dots, L_n^{(\alpha)})$ described by

$$L_i^{(\alpha)}(x) := \sum_{k=0}^i (-1)^k \binom{i+\alpha}{i-k} \frac{x^k}{k!}, \quad i = 0, \dots, n. \quad (6.20)$$

It is well known that this polynomial basis is orthogonal on the interval $[0, \infty)$ with respect to the weight function $x^\alpha e^{-x}$.

In [20] it is proved that the matrix of change of basis between the generalized Laguerre basis (6.20) and the monomial basis is TP. Then, accurate computations when considering collocation matrices $\left(L_{j-1}^{(\alpha)}(x_{j-1})\right)_{1 \leq i, j \leq n+1}$ with $(0 >)x_0 > x_1 > \dots > x_n$ are provided.

The following result analyzes the total positivity of Laguerre Wronskian matrices and provides a factorization that allows to solve with HRA some algebraic problems.

Theorem 6.3. *Let $(L_0^{(\alpha)}, \dots, L_n^{(\alpha)})$ be the Laguerre basis defined in (6.20) and J the diagonal matrix $J := \text{diag}((-1)^{i-1})_{1 \leq i \leq n+1}$. Then, for any $x < 0$, the matrix*

$$W_J := JW(L_0^{(\alpha)}, \dots, L_n^{(\alpha)})(x) \quad (6.21)$$

is a nonsingular TP matrix and its bidiagonal decomposition (6.1) can be computed with HRA. Furthermore, the computation of all the singular values, the inverse of W_J , as well as the solution of the linear systems $W_J x = b$, where $b = (b_0, \dots, b_n)^T$ has alternating signs, can be performed with HRA.

Proof. In Theorem 2 of [20] it is shown that the matrix A of the change of basis between the generalized Laguerre basis (6.20) and the monomial basis (6.5) such that

$$(L_0^{(\alpha)}, \dots, L_n^{(\alpha)}) = (p_0, \dots, p_n)A, \quad (6.22)$$

satisfies

$$A = JS_\alpha^{-1} P_U S_0^{-1} S_\alpha, \quad (6.23)$$

where $S_\alpha := \text{diag}((\alpha + i - 1)^{i-1})_{1 \leq i \leq n+1}$ and $P_U \in \mathbb{R}^n$ is an upper triangular Pascal matrix, that is, the $(n+1) \times (n+1)$ upper triangular matrix with $\binom{j-1}{i-1}$ as (i, j) -entry for $j \geq i$.

Let (ℓ_0, \dots, ℓ_n) such that $\ell_i(x) = (-x)^i$, $i = 0, \dots, n$. Since $(p_0, \dots, p_n) = (\ell_0, \dots, \ell_n)J$, taking into account identities (6.22) and (6.23), we can write

$$(L_0^{(\alpha)}, \dots, L_n^{(\alpha)}) = (\ell_0, \dots, \ell_n) S_\alpha^{-1} P_U S_0^{-1} S_\alpha. \quad (6.24)$$

Let us observe that the upper triangular Pascal matrix P_U is nonsingular and TP (see [27]) and so are the positive diagonal matrices S_α^{-1} , S_0^{-1} and S_α . Then, we can deduce that $S_\alpha^{-1} P_U S_0^{-1} S_\alpha$ is also nonsingular and TP since it is a product of nonsingular and TP matrices.

On the other hand, since $\ell_j^{(i)}(x) = (-1)^i p_j^{(i)}(-x)$, $0 \leq i \leq j \leq n$, the following matrix equality can be easily deduced

$$JW(\ell_0, \dots, \ell_n)(x) = W(p_0, \dots, p_n)(-x), \quad x \in \mathbb{R}. \quad (6.25)$$

Then, using equality (6.25), we can deduce that the scaled Wronskian matrix $JW(L_0^{(\alpha)}, \dots, L_n^{(\alpha)})(x)$, $x \in \mathbb{R}$, satisfies

$$\begin{aligned} JW(L_0^{(\alpha)}, \dots, L_n^{(\alpha)})(x) &= JW(\ell_0, \dots, \ell_n)(x)S_\alpha^{-1}P_US_0^{-1}S_\alpha \\ &= W(p_0, \dots, p_n)(-x)S_\alpha^{-1}P_US_0^{-1}S_\alpha. \end{aligned}$$

Moreover, from Proposition 6.1, $W(p_0, \dots, p_n)(-x)$, $x < 0$, is nonsingular TP and so is the matrix $JW(L_0^{(\alpha)}, \dots, L_n^{(\alpha)})(x)$, since it is the product of nonsingular TP matrices.

The bidiagonal factorization (6.1) of $JW(\ell_0, \dots, \ell_n)(x) = W(p_0, \dots, p_n)(-x)$, $x < 0$, is described by (6.6) and (6.7). Clearly, it can be computed with HRA. On the other hand, in Theorem 2 of [20] it is shown that the bidiagonal factorization (6.1) of the matrix $S_\alpha^{-1}P_US_0^{-1}S_\alpha$ is

$$S_\alpha^{-1}P_US_0^{-1}S_\alpha = S_0^{-1}G_1 \cdots G_n,$$

where G_k , $k = 1, \dots, n$, is the bidiagonal upper triangular matrix with unit diagonal whose $(i, i+1)$ entry is

$$\tilde{m}_{i,i-k} = \frac{i+\alpha}{i}, \quad k < i.$$

Again, this factorization can be computed with HRA. Finally, following Section 5.2 of [62], the bidiagonal factorization (6.1) of $JW(L_0^{(\alpha)}, \dots, L_n^{(\alpha)})(x)$, $x < 0$, can be computed with HRA using the subtraction-free Algorithm 5.1 in [62], and the bidiagonal factorizations (6.1) of $JW(\ell_0, \dots, \ell_n)(x)$ and $S_\alpha^{-1}P_US_0^{-1}S_\alpha$, which can be provided with HRA.

This fact guarantees that algebraic problems such that the computation of all the singular values, the inverse matrix of W_J , and the solution of the linear systems $W_Jx = b$, where $b = (b_0, \dots, b_n)^T$ has alternating signs, can be performed with HRA (see Section 3 of the [23]). \square

As a consequence of the above theorem we can deduce the following result.

Corollary 6.1. *Let $W := W(L_0^{(\alpha)}, \dots, L_n^{(\alpha)})(x)$ be the wronskian matrix of the Laguerre basis $(L_0^{(\alpha)}, \dots, L_n^{(\alpha)})$ defined in (6.20). Then, for any $x < 0$, the bidiagonal factorization (6.1) of W can be computed with HRA. Moreover, the computation of all its singular values, the inverse of W , as well as the solution of the linear systems $Wx = b$, where the elements of $b = (b_0, \dots, b_n)^T$ have the same sign, can be performed with HRA.*

Proof. Let $J := \text{diag}((-1)^{i-1})_{1 \leq i \leq n+1}$. By Theorem 6.3, the bidiagonal decomposition (6.1) of $W_J := JW$ can be computed with HRA. By multiplying this factorization by $J = J^{-1}$, we can derive with HRA the corresponding bidiagonal factorization of W .

On the other hand, let us observe that, since J is a unitary matrix, the singular values of W coincide with those of W_J and then, from Theorem 6.3, their computation for $x < 0$ can be performed with HRA. Similarly, taking into account that

$$W^{-1} = W_J^{-1}J,$$

Theorem 6.3 also guarantees the accurate computation of W^{-1} . Finally, if we have a linear system of equations $Wx = b$, where the elements of $b = (b_0, \dots, b_n)^T$ have the same sign, from Theorem 6.3, we will be able to solve with HRA the equivalent system $JWx = Jb$, since Jb has alternating signs. \square

Let us now consider the polynomial basis $(\tilde{L}_0^{(\alpha)}, \dots, \tilde{L}_n^{(\alpha)})$ obtained by changing the variable in the Laguerre basis as follows:

$$\tilde{L}_i^{(\alpha)}(x) := L_i^{(\alpha)}(-x) = \sum_{k=0}^i \binom{i+\alpha}{i-k} \frac{x^k}{k!}, \quad i = 0, \dots, n. \quad (6.26)$$

As in Theorem 6.3, using the results in this chapter, the analysis of the total positivity of the Wronskian matrix $W(\tilde{L}_0^{(\alpha)}, \dots, \tilde{L}_n^{(\alpha)})(x)$, $x \in \mathbb{R}$, can also be performed.

Theorem 6.4. *Let $(\tilde{L}_0^{(\alpha)}, \dots, \tilde{L}_n^{(\alpha)})$ be the polynomial basis defined in (6.26). Then, for any $x > 0$, the Wronskian matrix $W := W(\tilde{L}_0^{(\alpha)}, \dots, \tilde{L}_n^{(\alpha)})(x)$, $x > 0$, is nonsingular and TP and its bidiagonal decomposition (6.1) can be computed with HRA. Furthermore, the computation of all the singular values, the inverse of W , as well as the solution of the linear systems $Wx = b$, where $b = (b_0, \dots, b_n)^T$ has alternating signs, can be performed with HRA.*

Proof. It can be easily checked that the matrix A of change of basis between the basis (6.26) and the monomial basis (6.5), such that $(\tilde{L}_0^{(\alpha)}, \dots, \tilde{L}_n^{(\alpha)}) = (p_0, \dots, p_n)A$, satisfies

$$A = S_\alpha^{-1} P_U S_0^{-1} S_\alpha,$$

where $S_\alpha := \text{diag}((\alpha + i - 1)^{i-1})_{1 \leq i \leq n+1}$ and P_U is the $(n+1) \times (n+1)$ upper triangular Pascal matrix. Consequently, $W(\tilde{L}_0^{(\alpha)}, \dots, \tilde{L}_n^{(\alpha)})(x)$ satisfies

$$W(\tilde{L}_0^{(\alpha)}, \dots, \tilde{L}_n^{(\alpha)})(x) = W(p_0, \dots, p_n)(x) S_\alpha^{-1} P_U S_0^{-1} S_\alpha, \quad x \in \mathbb{R},$$

where $S_\alpha^{-1} P_U S_0^{-1} S_\alpha$ is nonsingular and TP because it is a product of nonsingular and TP matrices. As in the proof of Theorem 6.3, from Proposition 6.1 and taking into account that the product of nonsingular and TP matrices is nonsingular and TP, it can be deduced that W is nonsingular TP and its bidiagonal factorization (6.1) can be provided with HRA, which guarantees that the mentioned algebraic problems can be performed with HRA. \square

Section 6.5 shows accurate results obtained when solving the mentioned algebraic problems using the bidiagonal factorization (6.1) and the algorithms presented in [62] and [63].

6.5 Numerical experiments

Given a nonsingular and TP matrix whose bidiagonal factorization (6.1) can be computed with HRA, the functions

TNSingularValues, TNInverseExpand and TNSolve, available in the library TNTTool of [63], can be used to compute with HRA its singular values, its inverse matrix and the solution of some linear systems, respectively. The function TNProduct is also available in the mentioned library and computes with HRA the bidiagonal decomposition (6.1) of AB when the bidiagonal decomposition (6.1) of two nonsingular and TP matrices A and B is provided. The computational cost of the aforementioned functions is $O(n^2)$ elementary operations for TNSolve and $O(n^3)$ for the other functions.

Using the results in this chapter, we have implemented Matlab functions for the efficient computation of the bidiagonal decomposition (6.1) of the wronkian matrix at $x \in \mathbb{R}$ of Bessel polynomial bases, reverse Bessel polynomial bases, generalized Laguerre bases and the polynomial bases defined in (6.26).

We have considered $(n+1) \times (n+1)$ Wronskian matrices \mathbf{W}_n at $x = 0.3$, $x = 2$, $x = -5$ and $x = 50$. Table 6.1 illustrates the 2-norm condition number of the mentioned Wronskian matrices computed with the Mathematica command `Norm[A, 2] · Norm[Inverse[A], 2]`. Observe that the condition number of the matrices considerably increases with their dimension. Due to this ill conditioning, traditional

Table 6.1: From left to right, condition number of Wronskian matrices at $x_0 = 2$ and $x_0 = 50$ of the Bessel polynomial bases, condition number of Wronskian matrices at $x_0 = 0.3$ and $x_0 = 50$ of the reverse Bessel polynomial bases, condition number of the Wronskian matrices of generalized Laguerre polynomials (W of Corollary 6.1) at $x_0 = -5$ (with $\alpha = 2$) and, finally, condition number of Wronskian matrices at $x_0 = 2$ of the polynomial basis defined in (6.26) (with $\alpha = 0$).

$n+1$	$\kappa_2(\mathbf{W}_n)$	$\kappa_2(\mathbf{W}_n)$	$\kappa_2(\mathbf{W}_n)$	$\kappa_2(\mathbf{W}_n)$	$\kappa_2(\mathbf{W}_n)$	$\kappa_2(\mathbf{W}_n)$
10	4.2×10^{14}	3.4×10^{31}	2.5×10^8	2.5×10^{25}	7.3×10^8	1.1×10^6
15	3.4×10^{26}	3.5×10^{49}	1.7×10^{15}	1.2×10^{37}	1.0×10^{12}	9.6×10^8
20	3.5×10^{39}	4.7×10^{67}	1.1×10^{23}	1.6×10^{49}	1.1×10^{15}	8.7×10^{11}
25	2.6×10^{54}	6.4×10^{86}	2.8×10^{32}	8.6×10^{54}	1.0×10^{18}	8.0×10^{14}

methods do not achieve accurate solutions when solving the mentioned algebraic problems. The following numerical results confirm this fact and illustrate the high accuracy obtained when using the functions in [63] with the bidiagonal factorizations (6.1) in this chapter.

We have compared the singular values obtained when using the Matlab command `svd` and those computed using the bidiagonal decompositions (6.1) in this chapter and the Matlab function `TNSingularValues`. In order to determine the accuracy of the approximations, we have also calculated the singular values of the matrices by using Mathematica with a precision of 100 digits and computed the relative errors corresponding to the approximations, considering the singular values provided by Mathematica as exact. We have computed the relative error of the approximations a of the exact eigenvalue and singular value \tilde{a} by means of the formula $e = |a - \tilde{a}|/|a|$.

Tables 6.2 and 6.3 show the relative errors of the approximations to the lowest singular value obtained with both methods for the Bessel and the Laguerre case, respectively. Observe that the singular values obtained using the factorization (6.1) are very accurate for all considered n , whereas the approximations of the singular values obtained with the Matlab commands `svd` are not very accurate when n increases.

Table 6.2: Relative errors when computing the lowest singular value of Wronskian matrices of Bessel polynomial bases at $x_0 = 2$ and Wronskian matrices of reverse Bessel polynomial bases at $x_0 = 0.3$.

$n+1$	<code>svd</code>	TNSV	<code>svd</code>	TNSV
10	3.0×10^{-4}	2.1×10^{-16}	1.4×10^{-9}	3.9×10^{-15}
15	7.6×10^{-1}	5.7×10^{-16}	5.3×10^{-3}	2.4×10^{-15}
20	7.9	3.9×10^{-16}	3.7×10^{-1}	6.8×10^{-15}
25	8.0	1.6×10^{-16}	9.6×10^{-1}	5.9×10^{-15}

We have also used the function `TNInverseExpand` (see Section 4 of [87]) with the factorization (6.1) proposed in this chapter in order to compute the inverse of the considered Wronskian matrices. We have also computed their approximations with the command function `inv`. In order to determine the accuracy of the approximations, we have calculated the inverse of these matrices by using Mathematica with a precision of 100 digits and computed the relative errors corresponding to the approximations, considering the inverse matrix provided by Mathematica as exact. We have computed the relative error of each approximation \tilde{A}^{-1} of the exact inverse matrix A^{-1} by means of the formula $e = \|A^{-1} - \tilde{A}^{-1}\|_2 / \|A^{-1}\|_2$.

Tables 6.4 and 6.5 show the relative errors of the approximation to the inverse of the considered

Table 6.3: Relative errors when computing the lowest singular value of Wronskian matrices of generalized Laguerre polynomials (W of Corollary 6.1) at $x_0 = -5$ (with $\alpha = 2$) and Wronskian matrices of the polynomial bases defined in (6.26) at $x_0 = 2$ (with $\alpha = 0$).

n+1	svd	TNSV	svd	TNSV
10	3.6×10^{-11}	2.2×10^{-16}	1.6×10^{-13}	8.3×10^{-16}
15	1.6×10^{-9}	1.2×10^{-15}	5.6×10^{-12}	1.3×10^{-17}
20	2.4×10^{-9}	4.7×10^{-15}	1.3×10^{-9}	3.0×10^{-15}
25	4.8×10^{-6}	2.6×10^{-15}	1.1×10^{-7}	1.4×10^{-15}

matrices obtained with both methods. For all considered cases, the approximation of the inverse matrix obtained by means of `TNInverseExpand` and the factorization (6.1) is very accurate, providing much better results than those obtained by using the command `inv`.

Table 6.4: Relative errors when computing the inverse of the Wronskian matrices of Bessel polynomial bases at $x_0 = 50$ and the inverse of the Wronskian matrices of reverse Bessel polynomial bases at $x_0 = 50$.

n+1	inv	TNIE	inv	TNIE
10	2.0×10^{-14}	1.8×10^{-16}	6.1×10^{-15}	5.2×10^{-17}
15	3.7×10^{-12}	1.1×10^{-16}	6.6×10^{-11}	1.8×10^{-16}
20	3.5×10^{-9}	4.8×10^{-17}	1.0×10^{-7}	4.6×10^{-16}
25	2.4×10^{-6}	2.4×10^{-16}	5.0×10^{-5}	3.0×10^{-16}

Table 6.5: Relative errors when computing the inverse of the Wronskian matrices of generalized Laguerre polynomials (W of Corollary 6.1) at $x_0 = -5$ (with $\alpha = 2$) and Wronskian matrices of the polynomial bases defined in (6.26) at $x_0 = 2$ (with $\alpha = 0$).

n+1	inv	TNIE	inv	TNIE
10	7.4×10^{-14}	1.8×10^{-16}	1.9×10^{-14}	5.7×10^{-17}
15	2.7×10^{-11}	2.1×10^{-16}	8.8×10^{-13}	2.9×10^{-16}
20	4.7×10^{-10}	4.8×10^{-15}	4.3×10^{-11}	3.6×10^{-15}
25	1.5×10^{-8}	1.6×10^{-15}	4.1×10^{-10}	1.6×10^{-15}

Finally, we shall illustrate the accuracy of the solutions of linear systems computed by using the bidiagonal factorization (6.1) with the function `TNSolve`. We have obtained the solution of the linear systems using Mathematica with a precision of 100 digits and considered this solution exact. Then we have also computed with Matlab two approximations, the first one using the previous functions and the second one using the Matlab command `\`. We have computed the relative error of every approximation $\tilde{c} = (\tilde{c}_1, \dots, \tilde{c}_{n+1})$ of the solution c of the linear system by means of the formula $e = \|c - \tilde{c}\|_2 / \|c\|_2$.

Tables 6.6 and 6.7 show the relative errors when solving the linear systems $\mathbf{W}_n c_n = \mathbf{d}_n$ where $\mathbf{d}_n = ((-1)^{i+1} d_i)_{1 \leq i \leq n+1}$ or $\mathbf{d}_n = (d_i)_{1 \leq i \leq n+1}$, in the case the generalized Laguerre polynomials (W of Corollary 6.1), and $d_i, i = 1, \dots, n+1$, are random non negative integer values. The computed results confirm the accuracy of the proposed method that, clearly, keeps the accuracy when the dimension of the problem increases. In contrast, when n increases the condition number of the considered matrices considerably increases and that explains the bad results obtained with the Matlab command `\`.

Table 6.6: Relative errors when solving $\mathbf{W}_n \mathbf{c}_n = \mathbf{d}_n$ with Wronskian matrices of Bessel polynomial bases at $x_0 = 50$ and Wronskian matrices of reverse Bessel polynomial bases at $x_0 = 50$.

$n+1$	$\mathbf{W}_n \setminus \mathbf{d}_n$	TNSolve	$\mathbf{W}_n \setminus \mathbf{d}_n$	TNSolve
10	1.4×10^{-13}	2.8×10^{-17}	3.2×10^{-14}	2.8×10^{-16}
15	1.4×10^{-11}	3.5×10^{-16}	2.7×10^{-11}	1.3×10^{-16}
20	5.1×10^{-9}	3.1×10^{-16}	6.1×10^{-8}	3.7×10^{-16}
25	1.4×10^{-6}	3.4×10^{-16}	2.0×10^{-5}	2.5×10^{-16}

Table 6.7: Relative errors when solving $\mathbf{W}_n \mathbf{c}_n = \mathbf{d}_n$ with Wronskian matrices of generalized Laguerre polynomials (W of Corollary 6.1) at $x_0 = -5$ (with $\alpha = 2$) and Wronskian matrices of the polynomial bases defined in (6.26) at $x_0 = 2$ (with $\alpha = 0$).

$n+1$	$\mathbf{W}_n \setminus \mathbf{d}_n$	TNSolve	$\mathbf{W}_n \setminus \mathbf{d}_n$	TNSolve
10	6.7×10^{-14}	1.4×10^{-16}	2.4×10^{-14}	7.2×10^{-17}
15	3.1×10^{-11}	1.5×10^{-16}	6.5×10^{-13}	3.3×10^{-16}
20	1.9×10^{-10}	3.7×10^{-15}	1.6×10^{-11}	2.6×10^{-15}
25	1.3×10^{-8}	1.5×10^{-15}	1.8×10^{-10}	6.6×10^{-15}

7

Total positivity and accurate computations with Wronskian matrices of Bernstein and related bases

ABOUT THIS CHAPTER

The purpose of this chapter is to present some of the latest results that we have obtained, which are not included in the articles that belong to the compendium of publications of this thesis. It should be noted that, in this chapter, we have taken into account some of the results shown in the article [75] (see on page 99).

7.1 Introduction

The Bernstein basis of polynomials is the polynomial basis most used in Computer-Aided Geometric Design (CAGD) (see [28], [31]). In fact, the Bernstein basis has optimal shape preserving [9] and stability [30] properties. In [82] it was shown that many algebraic computations with the collocation matrices of the Bernstein basis can be performed with High Relative Accuracy (HRA). In fact, these matrices are totally positive (TP) and so they possess a bidiagonal factorization. The bidiagonal factorization of a nonsingular TP matrix A is the start point to compute with HRA its eigenvalues and singular values, the matrix A^{-1} and even the solution of $Ax = b$ for vectors b with alternating signs. In this chapter we consider the Wronskian matrices of Bernstein polynomials and other related bases, including interesting bases such as the Bernstein basis of negative degree (see [41]) or the negative binomial basis. These Wronskian matrices come from applications in CAGD and statistics and they can also arise in Hermite interpolation problems, in particular in Taylor interpolation problems. A first difficulty found to obtain HRA algorithms with these matrices comes from the fact that the Wronskian matrices of the Bernstein basis is never TP, as can be seen in Corollary 7.1, where we characterize when the Wronskian matrices of these general bases are TP. However, in spite that they fail being TP, we have obtained a bidiagonal factorization of these Wronskian matrices and we have used it to derive algorithms to compute with HRA their eigenvalues and singular values, their inverses and the solution of some linear systems. The complexity of the proposed algorithms for solving the mentioned algebraic problems is comparable to that of the traditional LAPACK algorithms, which, as we shall illustrate, deliver no such accuracy.

We now describe the layout of the chapter. Section 7.2 presents basic definitions and results that will be used in the chapter. Section 7.3 provides the bidiagonal decomposition of the collocation matrix of the general class of functions related with the Bernstein basis. Section 7.4 deals with the accurate computations with the corresponding Wronskian matrices. We obtain a bidiagonal factorization of

then the decomposition (7.1) is unique. The diagonal entries $p_{i,i}$ of D are the diagonal pivots of the Neville elimination of A and the elements $m_{i,j}$, $\tilde{m}_{i,j}$ are nonnegative and coincide with the multipliers of the Neville elimination of A and A^T , respectively.

In [61], the bidiagonal factorization (6.1) of an $(n+1) \times (n+1)$ nonsingular and TP matrix A is represented by defining a matrix $BD(A) = (BD(A)_{i,j})_{1 \leq i,j \leq n+1}$ such that

$$BD(A)_{i,j} := \begin{cases} m_{i,j}, & \text{if } i > j, \\ p_{i,i}, & \text{if } i = j, \\ \tilde{m}_{j,i}, & \text{if } i < j. \end{cases} \quad (7.3)$$

Remark 7.1. Observe that, by Theorem 4.3 of [37], the positivity of all multipliers and diagonal pivots in Theorem 1 implies that A is STP.

The following result can be easily checked and will be useful in next sections.

Lemma 7.1. Let d_1, \dots, d_{n+1} be real values and A an $(n+1) \times (n+1)$ TP matrix whose bidiagonal factorization (7.1) is

$$A = F_n F_{n-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n.$$

Then, the bidiagonal factorization (7.1) of $\tilde{A} := A\Delta$ with $\Delta = \text{diag}(d_1, \dots, d_{n+1})$ is

$$\tilde{A} = F_n F_{n-1} \cdots F_1 \tilde{D} \tilde{G}_1 \cdots \tilde{G}_{n-1} \tilde{G}_n,$$

where $\tilde{D} = \text{diag}(d_1 p_{1,1}, d_2 p_{2,2}, \dots, d_{n+1} p_{n+1,n+1})$ and \tilde{G}_i , $i = 1, \dots, n$, are the upper bidiagonal matrices described in (7.2) whose off diagonal entries are

$$\tilde{r}_{i,j} = \frac{d_i}{d_{i-1}} \tilde{m}_{i,j}, \quad 1 \leq j < i \leq n+1.$$

Let us recall that $x \in \mathbb{R}$ is obtained with high relative accuracy (HRA) if the relative error of the computed value \tilde{x} satisfies

$$\frac{\|x - \tilde{x}\|}{\|x\|} < Ku,$$

where K is a positive constant independent of the arithmetic precision and u is the unit round-off. HRA implies that the relative errors of the computations are of the order of the machine precision. An algorithm can be computed with HRA when it only uses products, quotients, sums of numbers of the same sign, subtractions of numbers of opposite sign or subtraction of initial (cf. [23], [61]).

If the bidiagonal factorization (7.1) of a nonsingular TP matrix A is computed with HRA then, using the algorithms in [62], we can also compute with HRA its eigenvalues and singular values, the matrix A^{-1} and even the solution of $Ax = b$ for vectors b with alternating signs.

In [71], we can find algorithms for computing the bidiagonal decomposition (7.1) of the collocation matrices of a general class of bases (u_0^n, \dots, u_n^n) with

$$u_i^n(x) := \binom{n}{i} f^i(x) g^{n-i}(x), \quad x \in [a, b], \quad i = 0, \dots, n,$$

where $f, g : I \rightarrow \mathbb{R}$ are functions such that $f(x) \neq 0$, $g(x) \neq 0$ for all $x \in (a, b)$ and f/g is strictly increasing. These bases are of interest in Computer Aided Geometric Design and also in Approximation Theory. In particular, Theorem 2 of [71] proves that these collocation matrices,

$$M_{n+1, x_1, \dots, x_{n+1}} := \left(\binom{n}{j-1} f^{j-1}(x_i) g^{n-j+1}(x_i) \right)_{1 \leq i, j \leq n+1},$$

are STP at $x_1 < \dots < x_{n+1}$ on (a, b) . Moreover, Theorem 3 of [71] deduces their bidiagonal factorization (7.1). Using this factorization, in [71] accurate computations with collocation matrices of bases with algebraic, trigonometric, or hyperbolic polynomials are illustrated.

It can be easily checked that, following the proof of Theorem 3 of [71], the bidiagonal factorization (7.1) of systems $(u_0^\alpha, \dots, u_n^\alpha)$ with $\alpha \in \mathbb{R}$ and

$$u_i^\alpha(x) := f^i(x)g^{\alpha-i}(x), \quad i = 0, \dots, n, \quad (7.4)$$

can be obtained. The following result describes this factorization.

Theorem 7.2. *The collocation matrix $M_{n+1, x_1, \dots, x_{n+1}}$ of the basis (7.4) at $x_1 < \dots < x_{n+1}$ in its domain admits the following factorization*

$$M_{n+1, x_1, \dots, x_{n+1}} = F_n F_{n-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n,$$

where the entries $m_{i,j}$, $\tilde{m}_{i,j}$ and $p_{i,i}$ of F_i and G_i , $i = 1, \dots, n$, and D are given by

$$\begin{aligned} m_{i,j} &= \frac{g^{\alpha-j+1}(x_i)g(x_{i-j})}{g^{\alpha-j+2}(x_{i-1})} \frac{\prod_{k=1}^{j-1} (f(x_i)g(x_{i-k}) - f(x_{i-k})g(x_i))}{\prod_{k=2}^j (f(x_{i-1})g(x_{i-k}) - f(x_{i-k})g(x_{i-1}))}, \\ \tilde{m}_{i,j} &= \frac{f(x_j)}{g(x_j)}, \quad 1 \leq j < i \leq n+1, \\ p_{i,i} &= \frac{g^{\alpha-i+1}(x_i)}{\prod_{k=1}^{i-1} g(x_k)} \prod_{k=1}^{i-1} (f(x_i)g(x_k) - f(x_k)g(x_i)), \quad 1 \leq i \leq n+1. \end{aligned}$$

Let us denote by \mathbf{P}^n the space of polynomials of degree less than or equal to n and (p_0, \dots, p_n) the monomial basis of \mathbf{P}^n such that

$$p_i(x) := x^i, \quad i = 0, \dots, n. \quad (7.5)$$

The following result restates Corollary 1 of [75] (see the article on page 99), providing the bidiagonal factorization (7.1) of the Wronskian matrix $W(p_0, \dots, p_n)(x)$, $x \in \mathbb{R}$.

Proposition 7.1. *Let (p_0, \dots, p_n) be the monomial basis given in (7.5). For any $x \in \mathbb{R}$, the Wronskian matrix $W(p_0, \dots, p_n)(x)$ is nonsingular and can be factorized as follows,*

$$W(p_0, \dots, p_n)(x) = D G_{1,n} \cdots G_{n-1, n-1} G_{n,n}, \quad (7.6)$$

where $D = \text{diag}\{0!, 1!, \dots, n!\}$ and $G_{i,n}$, $i = 1, \dots, n$, are the upper triangular bidiagonal matrices in (7.2) with

$$\tilde{m}_{k, k-i} = x, \quad i+1 \leq k \leq n+1. \quad (7.7)$$

Moreover, if $x > 0$ then $W(p_0, \dots, p_n)(x)$ is nonsingular and TP, its bidiagonal decomposition (7.1) is given by (7.6) and (7.7) and it can be computed with HRA.

In [75] (see the article on page 99), using this result, accurate computations with Wronskian matrices of monomial bases are achieved.

In the following sections we shall obtain the bidiagonal factorization (7.1) of collocation and Wronskian matrices associated to a general class of functions that includes, as particular cases, polynomial Bernstein bases, negative binomial bases or Bernstein bases of negative degree. For all considered cases, we are going to achieve algebraic computations with HRA.

7.3 Bidiagonal decomposition of the collocation matrix of a general class of functions

Let us consider the system of functions $(f_0^\alpha, \dots, f_n^\alpha)$, $\alpha \in \mathbb{R}$, defined by

$$f_i^\alpha(x) := x^i(1-x)^{\alpha-i}, \quad i = 0, \dots, n, \quad (7.8)$$

on their natural domain. Let us observe that the Bernstein basis of the space \mathbf{P}^n , given by (B_0^n, \dots, B_n^n) and

$$B_i^n(x) := \binom{n}{i} x^i (1-x)^{n-i}, \quad i = 0, \dots, n, \quad (7.9)$$

is $(c_0 f_0^n, \dots, c_n f_n^n)$ with $c_i = \binom{n}{i}$, $i = 0, \dots, n$. Moreover, there are other interesting bases which can be obtained by scaling the systems (7.8). For example, if $\alpha = -n$ and $c_i = \binom{-n}{i} = (-1)^i \binom{n+i-1}{i}$, $i = 0, \dots, n$, then $(c_0 f_0^{-n}, \dots, c_n f_n^{-n})$ is the Bernstein basis of negative degree $(B_0^{-n}, \dots, B_n^{-n})$ with

$$B_i^{-n}(x) := \binom{-n}{i} x^i (1-x)^{-n-i} = \binom{n+i-1}{i} (-x)^i (1-x)^{-n-i}, \quad i = 0, \dots, n, \quad (7.10)$$

(cf. [41]). On the other hand, if $\alpha = n+1$ and $c_i = \binom{n}{i}$, $i = 0, \dots, n$, then $(c_0 f_0^{n+1}, \dots, c_n f_n^{n+1})$ is the negative binomial basis $(b_0^{n+1}, \dots, b_n^{n+1})$ with

$$b_i^{n+1}(x) := \binom{n}{i} x^i (1-x)^{n-i+1}, \quad i = 0, \dots, n. \quad (7.11)$$

Using Theorem 7.2, with $f(x) = x$ and $g(x) = 1-x$, it can be checked that the collocation matrix

$$M_{n+1, x_1, \dots, x_{n+1}} := \left(x_i^{j-1} (1-x_i)^{\alpha-j+1} \right)_{1 \leq i, j \leq n+1},$$

admits the following factorization

$$M_{n+1, x_1, \dots, x_{n+1}} = F_n F_{n-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n, \quad (7.12)$$

where F_i and G_i , $i = 1, \dots, n$, are the lower and upper triangular bidiagonal matrices described in (7.2) and $D = \text{diag}(p_{1,1}, \dots, p_{n+1, n+1})$. The entries $m_{i,j}$, $\tilde{m}_{i,j}$ and $p_{i,i}$ are given by

$$m_{i,j} = \frac{(1-x_i)^{\alpha-j+1} (1-x_{i-j}) \prod_{k=1}^{j-1} (x_i - x_{i-k})}{(1-x_{i-1})^{\alpha-j+2} \prod_{k=2}^j (x_{i-1} - x_{i-k})}, \quad 1 \leq j < i \leq n+1,$$

$$\tilde{m}_{i,j} = \frac{x_j}{1-x_j}, \quad 1 \leq j < i \leq n+1, \quad p_{i,i} = (1-x_i)^{\alpha-i+1} \prod_{k=1}^{i-1} \frac{x_i - x_k}{1-x_k}, \quad 1 \leq i \leq n+1.$$

Then it can be easily deduced that $M_{n+1, x_1, \dots, x_{n+1}}$ is STP for any $\alpha \in \mathbb{R}$ and any sequence of parameters such that $0 < x_1 < \dots < x_n < 1$.

Using Lemma 7.1 and the decomposition (7.12) of the collocation matrix of $(f_0^\alpha, \dots, f_n^\alpha)$, the bidiagonal factorization (7.1) of the collocation matrices of any system $(c_0 f_0^\alpha, \dots, c_n f_n^\alpha)$, $c_i \in \mathbb{R}$, $i = 0, \dots, n$, can be obtained.

In particular, the bidiagonal factorization (7.1) of the collocation matrices of Bernstein polynomial bases and negative binomial bases can be deduced. By means of this factorization, accurate computations with these matrices have been already achieved (see [71], [74], [82] and the references in there).

Furthermore, we can also deduce that the collocation matrix of the Bernstein basis of degree $-n$ satisfies

$$\left(B_{j-1}^{-n}(x_i) \right)_{1 \leq i, j \leq n+1} = F_n F_{n-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n,$$

and the entries $m_{i,j}$, $\tilde{m}_{i,j}$ and $p_{i,i}$ of F_i , G_i , $i = 1, \dots, n$, and D , respectively, are given by

$$\begin{aligned} m_{i,j} &= \frac{(1-x_i)^{-n-j+1}(1-x_{i-j})}{(1-x_{i-1})^{-n-j+2}} \frac{\prod_{k=1}^{j-1} (x_i - x_{i-k})}{\prod_{k=2}^j (x_{i-1} - x_{i-k})}, \quad 1 \leq j < i \leq n+1, \\ \tilde{m}_{i,j} &= -\frac{n+i-2}{i-1} \frac{x_j}{1-x_j}, \quad 1 \leq j < i \leq n+1, \\ p_{i,i} &= (-1)^{i-1} \binom{n+i-2}{i-1} (1-x_i)^{-n-i+1} \prod_{k=1}^{i-1} \frac{x_i - x_k}{1-x_k}, \quad 1 \leq i \leq n+1. \end{aligned} \tag{7.13}$$

Analyzing the sign of the entries in (7.13), we can deduce that the collocation matrix of the Bernstein basis of negative degree defined in (7.10) is TP for $x_{n+1} < \cdots < x_1 < 0$.

7.4 Accurate computations with Wronskian matrices of a general class of functions including Bernstein polynomials

In the following results we analyze the total positivity of the Wronskian matrices of the systems $(f_0^\alpha, \dots, f_n^\alpha)$, $\alpha \in \mathbb{R}$, with

$$f_i^\alpha(x) := x^i(1-x)^{\alpha-i}, \quad x < 1, \quad i = 0, \dots, n, \tag{7.14}$$

through their bidiagonal decomposition (7.1). First, we prove some auxiliary results.

Lemma 7.2. *For given $\alpha, t \in \mathbb{R}$ and $n \in \mathbb{N}$, let $L_{k,n} = (l_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, be the $(n+1) \times (n+1)$, lower triangular bidiagonal matrix with unit diagonal entries, such that*

$$l_{i,i-1}^{(k,n)} = 0, \quad i = 2, \dots, k, \quad l_{i,i-1}^{(k,n)} = (\alpha + 2 - i)t, \quad i = k+1, \dots, n+1.$$

Then, $L_n := L_{n,n} \cdots L_{1,n}$, is a lower triangular matrix and

$$L_n = (l_{i,j}^{(n)})_{1 \leq i, j \leq n+1}, \quad l_{i,j}^{(n)} = \frac{(i-1)!}{(j-1)!} \binom{\alpha+1-j}{\alpha+1-i} t^{i-j}, \quad 1 \leq j \leq i \leq n+1. \tag{7.15}$$

Proof. Clearly, L_n is a lower triangular matrix since it is the product of lower triangular bidiagonal matrices. Let us prove (7.15) by induction on n . For $n = 1$,

$$L_1 = L_{1,1} = \begin{pmatrix} 1 & \\ \alpha t & 1 \end{pmatrix},$$

and (7.15) clearly holds. Now, let us suppose that (7.15) holds for $n \geq 1$ and consider the $(n+2) \times (n+2)$ product matrix

$$L_{n+1} := L_{n+1,n+1} L_{n,n+1} \cdots L_{1,n+1}.$$

It can be checked that $\tilde{L}_{n+1} := L_{n+1,n+1} \cdots L_{2,n+1}$ satisfies $\tilde{L}_{n+1} = (\tilde{l}_{i,j}^{(n+1)})_{1 \leq i, j \leq n+2}$, with

$$\tilde{l}_{i,1}^{(n+1)} = \delta_{i,1}, \quad \tilde{l}_{1,i}^{(n+1)} = \delta_{1,i}, \quad i = 1, \dots, n+2, \quad \tilde{L}_{n+1}[2, \dots, n+2] = L_{n,n} \cdots L_{1,n}.$$

Therefore, since $\tilde{L}_{n+1}[2, \dots, n+2]$ satisfies the induction hypothesis, we can deduce from (7.15) that the entries of \tilde{L}_{n+1} satisfy the following equalities

$$\tilde{l}_{i,j}^{(n+1)} := \frac{(i-2)!}{(j-2)!} \binom{\alpha+2-j}{\alpha+2-i} t^{i-j}, \quad 2 \leq j \leq i \leq n+2. \quad (7.16)$$

Moreover, we can write

$$L_{n+1} = \tilde{L}_{n+1} L_{1,n+1} = \tilde{L}_{n+1} \begin{pmatrix} 1 & & & & \\ \alpha t & 1 & & & \\ & \ddots & \ddots & & \\ & & & \ddots & \\ & & & & (\alpha-n)t & 1 \end{pmatrix}. \quad (7.17)$$

Now, taking into account equalities (7.16), (7.17) and the fact that

$$\binom{\alpha+2-j}{\alpha+2-i} + \frac{\alpha+2-j}{j-1} \binom{\alpha+1-j}{\alpha+2-i} = \frac{i-1}{j-1} \binom{\alpha+2-j}{\alpha+2-i},$$

we deduce that $L_{n+1} = (l_{i,j}^{(n+1)})_{1 \leq i,j \leq n+2}$ satisfies

$$\begin{aligned} l_{i,j}^{(n+1)} &= \tilde{l}_{i,j}^{(n+1)} + \tilde{l}_{i,j+1}^{(n+1)} (\alpha+2-j)t \\ &= \frac{(i-2)!}{(j-2)!} \binom{\alpha+2-j}{\alpha+2-i} t^{i-j} + \frac{(i-2)!}{(j-1)!} \binom{\alpha+1-j}{\alpha+2-i} (\alpha+2-j)t^{i-j} \\ &= \frac{(i-2)!}{(j-2)!} \left(\binom{\alpha+2-j}{\alpha+2-i} + \frac{\alpha+2-j}{j-1} \binom{\alpha+1-j}{\alpha+2-i} \right) t^{i-j} = \frac{(i-1)!}{(j-1)!} \binom{\alpha+2-j}{\alpha+2-i} t^{i-j}, \end{aligned}$$

for $1 \leq j \leq i \leq n+2$. Consequently, (7.15) holds for all $n \in \mathbb{N}$. \square

Lemma 7.3. For a given $t \in \mathbb{R}$ and $n \in \mathbb{N}$, let $U_{k,n} = (u_{i,j}^{(k,n)})_{1 \leq j,i \leq n+1}$, $k = 1, \dots, n$, be the $(n+1) \times (n+1)$, upper triangular bidiagonal matrix with unit diagonal entries, such that

$$u_{i-1,i}^{(k,n)} := 0, \quad i = 2, \dots, k, \quad u_{i-1,i}^{(k,n)} := t, \quad i = k+1, \dots, n+1.$$

Then, $U_n := U_{1,n} \cdots U_{n,n}$, is an upper triangular matrix and

$$U_n = (u_{i,j}^{(n)})_{1 \leq i,j \leq n+1}, \quad u_{i,j}^{(n)} = \binom{j-1}{i-1} t^{j-i}, \quad 1 \leq i \leq j \leq n+1. \quad (7.18)$$

Proof. Clearly, U_n is an upper triangular matrix since it is the product of upper triangular bidiagonal matrices. Taking into account Proposition 7.1, we can deduce that

$$U_n = \left((i-1)! (p_{j-1}(t))^{(i-1)} \right)_{i,j=1, \dots, n+1},$$

where $p_j(t) := t^j$, $j = 0, \dots, n$. Finally, taking into account that

$$i! (p_j(t))^{(i)} = \binom{j}{i} t^{j-i}, \quad 0 \leq i \leq j \leq n,$$

equalities (7.18) are immediately obtained. \square

Now, using Lemma 7.2 and Lemma 7.3, we can derive the bidiagonal decomposition (7.1) of the Wronskian matrix of a system (7.14).

Theorem 7.3. *Let $n \in \mathbb{N}$, $\alpha \in \mathbb{R}$ and $(f_0^\alpha, \dots, f_n^\alpha)$ the system defined in (7.14). The Wronskian matrix $W := W(f_0^\alpha, \dots, f_n^\alpha)(x)$ admits a factorization of the form*

$$W = L_{n,n}L_{n-1,n} \cdots L_{1,n}DU_{1,n} \cdots U_{n-1,n}U_{n,n}, \quad (7.19)$$

where $L_{k,n} = (l_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, are the lower triangular bidiagonal matrices with unit diagonal entries, such that

$$l_{i,i-1}^{(k,n)} = 0, \quad i = 2, \dots, k, \quad l_{i,i-1}^{(k,n)} = (\alpha + 2 - i) \frac{-1}{1-x}, \quad i = k+1, \dots, n+1, \quad (7.20)$$

$U_{k,n} = (u_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, are the upper triangular bidiagonal matrices with unit diagonal entries, such that

$$u_{i-1,i}^{(k,n)} := 0, \quad i = 2, \dots, k, \quad u_{i-1,i}^{(k,n)} = \frac{x}{1-x}, \quad i = k+1, \dots, n+1, \quad (7.21)$$

and D is the diagonal matrix $D = \text{diag}(d_1, \dots, d_{n+1})$ with

$$d_i = (i-1)!(1-x)^{\alpha+2-2i}, \quad i = 1, \dots, n+1. \quad (7.22)$$

Proof. Let us observe that, by considering $t = -1/(1-x)$ in Lemma 7.2, we deduce that $L_n := L_{n,n}L_{n-1,n} \cdots L_{1,n}$ is a lower triangular matrix satisfying

$$L_n = (l_{i,j}^{(n)})_{1 \leq j, i \leq n+1}, \quad l_{i,j}^{(n)} = \frac{(i-1)!}{(j-1)!} \binom{\alpha+1-j}{i-j} \left(\frac{-1}{1-x} \right)^{i-j}, \quad 1 \leq j \leq i \leq n+1. \quad (7.23)$$

On the other hand, using Lemma 7.3 with $t = x/(1-x)$, we conclude that $U_n := U_{1,n} \cdots U_{n,n}$ is an upper triangular matrix with

$$U_n = (u_{i,j}^{(n)})_{1 \leq i, j \leq n+1}, \quad u_{i,j}^{(n)} = \binom{j-1}{i-1} \left(\frac{x}{1-x} \right)^{j-i}, \quad 1 \leq i \leq j \leq n+1. \quad (7.24)$$

In order to prove the result, taking into account (7.19), (7.22), (7.23) and (7.24), we have to check that

$$(f_{j-1}^\alpha)^{(i-1)}(x) = (i-1)! \left(\sum_{k=1}^{\min\{i,j\}} (-1)^{i-k} \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} x^{j-k} \right) (1-x)^{\alpha+2-i-j}, \quad (7.25)$$

for $1 \leq i, j \leq n+1$. Let us prove (7.25) by induction on i . Let $i = 1$, then

$$\sum_{k=1}^1 (-1)^{1-k} \binom{\alpha+1-k}{1-k} \binom{j-1}{k-1} x^{j-k} (1-x)^{\alpha-1-j+2} = x^{j-1} (1-x)^{\alpha+1-j} = f_{j-1}^\alpha(x),$$

for $j = 1, \dots, n+1$, and (7.25) follows. Now, let us assume that (7.25) holds for $i \geq 1$. Then, for any

$i < j \leq n + 1$, we have

$$\begin{aligned}
(f_{j-1}^\alpha)^{(i)}(x) &= \left((f_{j-1}^\alpha)^{(i-1)} \right)' \\
&= \left((i-1)! \sum_{k=1}^i (-1)^{i-k} \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} x^{j-k} (1-x)^{\alpha+2-i-j} \right)' \\
&= (i-1)! (1-x)^{\alpha+1-i-j} \left(\sum_{k=1}^i (-1)^{i-k+1} (\alpha+2-i-j) \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} x^{j-k} \right. \\
&\quad \left. + (1-x) \sum_{k=1}^i (-1)^{i-k} (j-k) \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} x^{j-k-1} \right) \\
&= (i-1)! (1-x)^{\alpha+1-i-j} \left(\sum_{k=1}^i (-1)^{i-k+1} (\alpha+2-i-j) \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} x^{j-k} \right. \\
&\quad \left. + \sum_{k=2}^{i+1} (-1)^{i-k+1} (j-k+1) \binom{\alpha+2-k}{i-k+1} \binom{j-1}{k-2} x^{j-k} \right. \\
&\quad \left. + \sum_{k=1}^i (-1)^{i-k+1} (j-k) \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} x^{j-k} \right) \\
&= (i-1)! (1-x)^{\alpha+1-i-j} \left(\sum_{k=1}^{i+1} (-1)^{i-k+1} c_k x^{j-k} \right),
\end{aligned}$$

where

$$\begin{aligned}
c_1 &= (\alpha+1-i) \binom{\alpha}{i-1} \binom{j-1}{0} = i \binom{\alpha}{i}, \\
c_k &= (\alpha+2-i-k) \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} + (j-k+1) \binom{\alpha+2-k}{i-k+1} \binom{j-1}{k-2} \\
&= i \binom{\alpha+1-k}{i+1-k} \binom{j-1}{k-1}, \quad k=2, \dots, i \\
c_{i+1} &= (j-i) \binom{\alpha+1-i}{0} \binom{j-1}{i-1} = i \binom{j-1}{i}.
\end{aligned}$$

Then, we can write

$$(f_{j-1}^\alpha)^{(i)}(x) = i! \left(\sum_{k=1}^{i+1} (-1)^{i-k+1} \binom{\alpha+1-k}{i+1-k} \binom{j-1}{k-1} x^{j-k} \right) (1-x)^{\alpha+1-i-j}, \quad j = i, \dots, n+1,$$

and check that (7.25) holds for $j = i, \dots, n+1$. Now, for $1 \leq j < i$, we can follow a similar reasoning,

$$\begin{aligned}
(f_{j-1}^\alpha)^{(i)}(x) &= \left((f_{j-1}^\alpha)^{(i-1)}(x) \right)' \\
&= \left((i-1)! \sum_{k=1}^j (-1)^{i-k} \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} x^{j-k} (1-x)^{\alpha+2-i-j} \right)' \\
&= (i-1)! (1-x)^{\alpha+1-i-j} \left(\sum_{k=1}^j (-1)^{i-k+1} (\alpha+2-i-j) \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} x^{j-k} \right. \\
&\quad \left. + (1-x) \sum_{k=1}^{j-1} (-1)^{i-k} (j-k) \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} x^{j-k-1} \right) \\
&= (i-1)! (1-x)^{\alpha+1-i-j} \left(\sum_{k=1}^j (-1)^{i-k+1} (\alpha+2-i-j) \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} x^{j-k} \right. \\
&\quad \left. + \sum_{k=2}^j (-1)^{i-k+1} (j-k+1) \binom{\alpha+2-k}{i-k+1} \binom{j-1}{k-2} x^{j-k} \right. \\
&\quad \left. + \sum_{k=1}^j (-1)^{i-k+1} (j-k) \binom{\alpha+1-k}{i-k} \binom{j-1}{k-1} x^{j-k} \right) \\
&= (i-1)! (1-x)^{\alpha+1-i-j} \left(\sum_{k=1}^j (-1)^{i-k+1} c_k x^{j-k} \right),
\end{aligned}$$

and, again, $c_k = i \binom{\alpha+1-k}{i+1-k} \binom{j-1}{k-1}$, $k = 1, \dots, j$. Then, we can write

$$(f_{j-1}^\alpha)^{(i)}(x) = i! \left(\sum_{k=1}^j (-1)^{i-k+1} \binom{\alpha+1-k}{i+1-k} \binom{j-1}{k-1} x^{j-k} \right) (1-x)^{\alpha+1-i-j}, \quad j = i, \dots, n+1,$$

and (7.25) also follows for $j = 1, \dots, i-1$. □

Theorem 7.1 and the analysis of the sign of the entries (7.20), (7.21) and (7.22) provides the following characterization of the total positivity $W(f_0^\alpha, \dots, f_n^\alpha)(x)$.

Corollary 7.1. *Given $\alpha \in \mathbb{R}$, let $(f_0^\alpha, \dots, f_n^\alpha)$ be the system defined in (7.14). The Wronskian matrix $W(f_0^\alpha, \dots, f_n^\alpha)(x)$ is TP if and only if $\alpha \leq 1-n$ and $0 < x < 1$.*

Example 7.1. *Let us illustrate with some examples the bidiagonal factorization (7.19), described by (7.20), (7.21) and (7.22), of the Wronskian matrix of $(f_0^\alpha, \dots, f_n^\alpha)$. For the particular case $n = 2$ and $\alpha = n$, the Wronskian matrix of the system $((1-x)^2, x(1-x), x^2)$ can be decomposed as follows*

$$\begin{aligned}
W(f_0^2, f_1^2, f_2^2)(x) &= \\
&\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{-1}{1-x} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \frac{-2}{1-x} & 1 & 0 \\ 0 & \frac{-1}{1-x} & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} 1 & \frac{x}{1-x} & 0 \\ 0 & 1 & \frac{x}{1-x} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{x}{1-x} \\ 0 & 0 & 1 \end{pmatrix},
\end{aligned}$$

where $d_1 = (1-x)^2$, $d_2 = 1$ and $d_3 = 2/(1-x)^2$, and it is not a TP matrix for any $x \in \mathbb{R}$.

For the particular case $n = 2$ and $\alpha = -n$, the Wronskian matrix of the system $(1/(1-x)^2, x/(1-x)^3, x^2/(1-x)^4)$ can be decomposed as follows

$$W(f_0^2, f_1^2, f_2^2)(x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{3}{1-x} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{1-x} & 1 & 0 \\ 0 & \frac{3}{1-x} & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} 1 & \frac{x}{1-x} & 0 \\ 0 & 1 & \frac{x}{1-x} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{x}{1-x} \\ 0 & 0 & 1 \end{pmatrix},$$

where $d_1 = 1/(1-x)^2$, $d_2 = 1/(1-x)^4$ and $d_3 = 2/(1-x)^6$, and, clearly, is TP for $x \in (0, 1)$.

For the particular case $n = 2$ and $\alpha = -5/2$, the Wronskian matrix of the system $(1/(1-x)^{5/2}, x/(1-x)^{7/2}, x^2/(1-x)^{9/2})$ can be decomposed as follows

$$W(f_0^2, f_1^2, f_2^2)(x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{7}{2(1-x)} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \frac{5}{2(1-x)} & 1 & 0 \\ 0 & \frac{7}{2(1-x)} & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} 1 & \frac{x}{1-x} & 0 \\ 0 & 1 & \frac{x}{1-x} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{x}{1-x} \\ 0 & 0 & 1 \end{pmatrix},$$

where $d_1 = 1/(1-x)^{5/2}$, $d_2 = 1/(1-x)^{9/2}$ and $d_3 = 2/(1-x)^{13/2}$, and, clearly, is TP for $x \in (0, 1)$.

Now, using Lema 7.1 and taking into account that the bidiagonal decomposition of the Wronskian matrix of the polynomial basis (f_0^n, \dots, f_n^n) , provided by Theorem 7.3, can be extended for all $x \neq 1$, we can derive the bidiagonal factorization of the Wronskian matrix of the Bernstein basis (7.9) using that

$$W(B_0^n, \dots, B_n^n)(x) = W(f_0^n, \dots, f_n^n)(x)\Delta, \quad \Delta := \text{diag} \left(\binom{n}{i-1} \right)_{1 \leq i \leq n+1}.$$

Theorem 7.4. Let $n \in \mathbb{N}$ and (B_0^n, \dots, B_n^n) the Bernstein basis of \mathbf{P}^n defined in (7.9). For a given $x \in \mathbb{R}$, $x \neq 1$, the Wronskian matrix $W := W(B_0^n, \dots, B_n^n)(x)$ admits a factorization of the form

$$W = L_{n,n}L_{n-1,n} \cdots L_{1,n}DU_{1,n} \cdots U_{n-1,n}U_{n,n}, \quad (7.26)$$

where $L_{k,n} = (l_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, are the lower triangular bidiagonal matrices, with unit diagonal entries, such that

$$l_{i,i-1}^{(k,n)} = 0, \quad i = 2, \dots, k, \quad l_{i,i-1}^{(k,n)} = (n+2-i) \frac{-1}{1-x}, \quad i = k+1, \dots, n+1, \quad (7.27)$$

$U_{k,n} = (u_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, are the upper triangular bidiagonal matrices, with unit diagonal entries, such that

$$u_{i-1,i}^{(k,n)} := 0, \quad i = 2, \dots, k, \quad u_{i-1,i}^{(k,n)} = \left(\frac{n+2-i}{i-1} \right) \frac{x}{1-x}, \quad i = k+1, \dots, n+1, \quad (7.28)$$

and D is the diagonal matrix $D = \text{diag}(d_1, \dots, d_{n+1})$ with

$$d_i = \binom{n}{i-1} (i-1)! (1-x)^{n+2-2i}, \quad i = 1, \dots, n+1. \quad (7.29)$$

Example 7.2. Let us illustrate the bidiagonal factorization (7.26), described by (7.27), (7.28) and (7.29), of the Wronskian matrix of the Bernstein polynomial basis. For the particular case $n = 2$,

the Wronskian matrix of $((1-x)^2, 2(1-x)x, x^2)$ can be decomposed as follows

$$W(B_0^2, B_1^2, B_2^2)(x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{-1}{1-x} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \frac{-2}{1-x} & 1 & 0 \\ 0 & \frac{-1}{1-x} & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} 1 & \frac{2x}{1-x} & 0 \\ 0 & 1 & \frac{x}{2(1-x)} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{x}{2(1-x)} \\ 0 & 0 & 1 \end{pmatrix},$$

where $d_1 = (1-x)^2$, $d_2 = 2$ and $d_3 = 2/(1-x)^2$.

Let us observe, that from Theorem 7.4, it can be deduced that the bidiagonal factorization (7.1) of the $(n+1) \times (n+1)$ dimensional Wronskian matrix W of the Bernstein basis of \mathbf{P}^n can be represented by means of the $(n+1) \times (n+1)$ matrix $BD(W) = (BD(W)_{i,j})_{1 \leq i, j \leq n+1}$ such that

$$BD(W)_{i,j} := \begin{cases} (n+2-i) \frac{-1}{1-x}, & \text{if } i > j, \\ \binom{n}{i-1} (i-1)! (1-x)^{n+2-2i}, & \text{if } i = j, \\ \binom{n+2-j}{j-1} \frac{x}{1-x}, & \text{if } i < j. \end{cases} \quad (7.30)$$

Let us observe that, analyzing the sign of the entries of (7.30), we can deduce that the Wronskian matrix of the Bernstein basis of \mathbf{P}^n is not TP for any $x \in \mathbb{R}$. However, the following result shows that the solution of several algebraic problems related to these matrices can be obtained with HRA using the bidiagonal decomposition (7.26).

Corollary 7.2. *Let $W := W(B_0^n, \dots, B_n^n)(x)$ be the Wronskian matrix of the Bernstein basis defined in (7.9) and J the diagonal matrix $J := \text{diag}((-1)^{i-1})_{1 \leq i \leq n+1}$. Then, for any $x < 0$,*

$$W_J := JWJ$$

is an STP matrix and its bidiagonal factorization (7.1) can be computed with HRA. Consequently, the computation of the eigenvalues, singular values of W , the matrix W^{-1} , as well as the solution $c = (c_1, \dots, c_{n+1})^T$ of linear systems $Wc = b$, where the entries of $b = (b_1, \dots, b_{n+1})^T$ have the same sign, can be performed with HRA.

Proof. Using Theorem 7.4 and that J^2 is the identity matrix, by (7.26) we can write

$$W_J = (JL_{n,n}J) \cdots (JL_{1,n}J)(JDJ)(JU_{1,n}J) \cdots (JU_{n,n}J), \quad (7.31)$$

which gives its bidiagonal factorization (7.31). Now, it can be easily checked that the multipliers and diagonal pivots of the bidiagonal factorization (7.1) of W_J are positive if

$$\frac{1}{1-x} > 0, \quad \frac{-x}{1-x} > 0, \quad 1-x > 0.$$

Therefore, by Remark 7.1, W_J is STP and its bidiagonal decomposition (7.31) can be computed with HRA for any $x < 0$. This fact guarantees the computation with HRA of the eigenvalues and singular values of W_J , the inverse matrix W_J^{-1} and the solution of the linear systems $W_J c = d$, where $d = (d_1, \dots, d_{n+1})^T$ has alternating signs (see Section 3 of [23]).

Let us observe that, since J is a unitary matrix, the eigenvalues and singular values of W coincide with those of W_J and therefore, using the bidiagonal decomposition (7.31) of W_J , their computation for $x < 0$ can be performed with HRA.

For the accurate computation of W^{-1} , we can take into account that

$$W^{-1} = JW_J^{-1}J. \quad (7.32)$$

Since, for $x < 0$, $W_J^{-1} = (\tilde{w}_{i,j})_{1 \leq i, j \leq +1}$ can be computed with HRA and, by (7.32), the inverse of the Wronskian matrix W satisfies $W^{-1} = ((-1)^{i+j} \tilde{w}_{i,j})_{1 \leq i, j \leq +1}$, we can also accurately compute W^{-1} by means of a suitable change of sign of the accurate computed entries of W_J^{-1} .

Finally, if we have a linear system of equations $Wc = b$, where the elements of $b = (b_1 \dots, b_{n+1})^T$ have the same sign, we can compute with HRA the solution $d \in \mathbb{R}^{n+1}$ of $W_J d = Jb$ and, consequently, the solution $c \in \mathbb{R}^{n+1}$ of the initial system since $c = Jd$. \square

Now, the following result describes the bidiagonal factorization (7.1) of Bernstein bases of negative degree (7.10). This decomposition can be easily deduced from Theorem 7.3 and Lemma 7.1, taking into account that

$$W(B_0^{-n}, \dots, B_n^{-n})(x) = W(f_0^{-n}, \dots, f_n^{-n})(x) \Delta, \quad \Delta := \text{diag} \left((-1)^i \binom{n+i-1}{i-1} \right)_{1 \leq i \leq n+1}. \quad (7.33)$$

Theorem 7.5. *Let $n \in \mathbb{N}$ and $(B_0^{-n}, \dots, B_n^{-n})$ the Bernstein basis of degree $-n$, defined in (7.10). For a given $x \in \mathbb{R}$, $x \neq 1$, the Wronskian matrix $W_{-n} := W(B_0^{-n}, \dots, B_n^{-n})(x)$ admits a factorization of the form*

$$W_{-n} = L_{n,n} L_{n-1,n} \cdots L_{1,n} D U_{1,n} \cdots U_{n-1,n} U_{n,n}, \quad (7.34)$$

where $L_{k,n} = (l_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, are the lower triangular bidiagonal matrices, with unit diagonal entries, such that

$$l_{i,i-1}^{(k,n)} = 0, \quad i = 2, \dots, k, \quad l_{i,i-1}^{(k,n)} = (n+i-2) \frac{1}{1-x}, \quad i = k+1, \dots, n+1, \quad (7.35)$$

$U_{k,n} = (u_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, are the upper triangular bidiagonal matrices, with unit diagonal entries, such that

$$u_{i-1,i}^{(k,n)} := 0, \quad i = 2, \dots, k, \quad u_{i-1,i}^{(k,n)} = - \left(\frac{n+i-2}{i-1} \right) \frac{x}{1-x}, \quad i = k+1, \dots, n+1, \quad (7.36)$$

and D is the diagonal matrix $D = \text{diag}(d_1, \dots, d_{n+1})$ with

$$d_i = (-1)^{i-1} \frac{(n+i-2)!}{(n-1)!} (1-x)^{-n+2-2i}, \quad i = 1, \dots, n+1. \quad (7.37)$$

Example 7.3. *Let us illustrate the bidiagonal factorization (7.34), described by (7.35), (7.36) and (7.37), of the Wronskian matrix of Bernstein bases of negative degree. For the particular case $n = -2$, the Wronskian matrix of $(1/(1-x)^2, -2x/(1-x)^3, 3x^2/(1-x)^4)$ can be decomposed as follows*

$$W(B_0^{-2}, B_1^{-2}, B_2^{-2})(x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{3}{1-x} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{1-x} & 1 & 0 \\ 0 & \frac{3}{1-x} & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} 1 & \frac{-2x}{1-x} & 0 \\ 0 & 1 & \frac{-3x}{2(1-x)} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{-3x}{2(1-x)} \\ 0 & 0 & 1 \end{pmatrix},$$

where $d_1 = 1/(1-x)^2$, $d_2 = -2/(1-x)^4$ and $d_3 = 3!/(1-x)^6$.

Now, from Theorem 7.5, the bidiagonal factorization (7.1) of the $(n+1) \times (n+1)$ dimensional Wronskian matrix W of the Bernstein basis of degree $-n$ can be represented by means of the $(n+1) \times (n+1)$ matrix $BD(W) = (BD(W)_{i,j})_{1 \leq i,j \leq n+1}$ such that

$$BD(W)_{i,j} := \begin{cases} (n+i-2) \frac{1}{1-x}, & \text{if } i > j, \\ (-1)^{i-1} \binom{n+i-2}{i-1} (i-1)! (1-x)^{-n+2-2i}, & \text{if } i = j, \\ -\binom{n+j-2}{j-1} \frac{x}{1-x}, & \text{if } i < j. \end{cases} \quad (7.38)$$

Using Theorem 7.5, it can be deduced that $W(B_0^{-n}, \dots, B_n^{-n})(x)$ is not TP at any $x \in \mathbb{R}$. Nevertheless, the following result shows that the bidiagonal decomposition (7.34) provides accurate computations with these matrices.

Corollary 7.3. *Let $W_{-n} := W(B_0^{-n}, \dots, B_n^{-n})(x)$ be the Wronskian matrix of the Bernstein basis of degree $-n$ defined in (7.10) and J the diagonal matrix $J := \text{diag}((-1)^{i-1})_{1 \leq i \leq n+1}$. Then, for $0 < x < 1$,*

$$W_{-n,J} := W_{-n}J$$

is an STP matrix and its bidiagonal factorization (7.1) can be computed with HRA. Consequently, the computation of the singular values of W_{-n} , the matrix W_{-n}^{-1} , as well as the solution $c = (c_1, \dots, c_{n+1})^T$ of linear systems $W_{-n}c = b$, where the entries of $b = (b_1, \dots, b_{n+1})^T$ have alternating signs, can be performed with HRA.

Proof. Taking into account Theorem 7.5, (7.33) and Lemma 7.1, it can be easily checked that the multipliers and diagonal pivots of the bidiagonal factorization (7.1) of $W_{-n,J}$ are positive if

$$\frac{1}{1-x} > 0, \quad \frac{x}{1-x} > 0, \quad 1-x > 0,$$

that is, if $0 < x < 1$. This fact guarantees, by Remark 7.1, that $W_{-n,J}$ is STP and the computation with HRA of its bidiagonal decomposition (7.1) and so, the computation with HRA of its eigenvalues and singular values, the inverse matrix $W_{-n,J}^{-1}$ and the solution of the linear systems $W_{-n,J}c = b$, where $b = (b_0, \dots, b_n)^T$ has alternating signs (see Section 3 of [23]).

On the other hand, since J is a unitary matrix, the singular values of $W_{-n,J}$ coincide with those of W_{-n} and so, their computation for $0 < x < 1$ can be performed with HRA. Similarly, taking into account that

$$W_{-n}^{-1} = JW_{-n,J}^{-1},$$

we can compute W_{-n}^{-1} accurately. Finally, if we have a linear system of equations $W_{-n}c = b$, where the elements of $b = (b_1, \dots, b_n)^T$ have alternating signs, we can solve with HRA the system $W_{-n,J}d = b$ and then obtain $c = Jd$. \square

Finally, using Lemma 7.1 and Theorem 7.3, we can derive the bidiagonal factorization of the Wronskian matrices of negative binomial bases (7.11), taking into account that

$$W(b_0^n, \dots, b_n^n)(x) = W(f_0^{n+1}, \dots, f_n^{n+1})(x)\Delta, \quad \Delta := \text{diag} \left(\binom{n}{i-1} \right)_{1 \leq i \leq n+1}.$$

Theorem 7.6. *Let $n \in \mathbb{N}$ and (b_0^n, \dots, b_n^n) the negative binomial basis of \mathbf{P}^n defined in (7.11). For a given $x \in \mathbb{R}$, $x \neq 1$, the Wronskian matrix $W_N := W(b_0^n, \dots, b_n^n)(x)$ admits a factorization of the form*

$$W_N = L_{n,n}L_{n-1,n} \cdots L_{1,n}DU_{1,n} \cdots U_{n-1,n}U_{n,n}, \quad (7.39)$$

where $L_{k,n} = (l_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, are the lower triangular bidiagonal matrices, with unit diagonal entries, such that

$$l_{i,i-1}^{(k,n)} = 0, \quad i = 2, \dots, k, \quad l_{i,i-1}^{(k,n)} = (n+3-i) \frac{-1}{1-x}, \quad i = k+1, \dots, n+1, \quad (7.40)$$

$U_{k,n} = (u_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, are the upper triangular bidiagonal matrices, with unit diagonal entries, such that

$$u_{i-1,i}^{(k,n)} := 0, \quad i = 2, \dots, k, \quad u_{i-1,i}^{(k,n)} = \left(\frac{n+2-i}{i-1} \right) \frac{x}{1-x}, \quad i = k+1, \dots, n+1, \quad (7.41)$$

and D is the diagonal matrix $D = \text{diag}(d_1, \dots, d_{n+1})$ with

$$d_i = \binom{n}{i-1} (i-1)! (1-x)^{n+3-2i}, \quad i = 1, \dots, n+1. \quad (7.42)$$

Example 7.4. Let us illustrate the bidiagonal factorization (7.39), described by (7.40), (7.41) and (7.42), of the Wronskian matrix of the negative binomial polynomial basis. For the particular case $n = 2$, the Wronskian matrix of $((1-x)^3, 2(1-x)^2x, (1-x)x^2)$ can be decomposed as follows

$$W(b_0^2, b_1^2, b_2^2)(x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{-2}{1-x} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \frac{-3}{1-x} & 1 & 0 \\ 0 & \frac{-2}{1-x} & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} 1 & \frac{2x}{1-x} & 0 \\ 0 & 1 & \frac{x}{2(1-x)} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{x}{2(1-x)} \\ 0 & 0 & 1 \end{pmatrix},$$

where $d_1 = (1-x)^3$, $d_2 = 2(1-x)$ and $d_3 = 2/(1-x)$.

Now, from Theorem 7.6, the bidiagonal factorization (7.1) of the $(n+1) \times (n+1)$ dimensional Wronskian matrix W of the Bernstein basis of degree $-n$ can be represented by means of the $(n+1) \times (n+1)$ matrix $BD(W) = (BD(W)_{i,j})_{1 \leq i, j \leq n+1}$ such that

$$BD(W)_{i,j} := \begin{cases} (n+3-i) \frac{-1}{1-x}, & \text{if } i > j, \\ \binom{n}{i-1} (i-1)! (1-x)^{n+3-2i}, & \text{if } i = j, \\ \left(\frac{n+2-j}{j-1} \right) \frac{x}{1-x}, & \text{if } i < j. \end{cases} \quad (7.43)$$

Taking into account Theorem 7.43, the Wronskian matrix of the negative binomial basis (7.11) is not TP at any $x \in \mathbb{R}$. However, following the reasoning in the proof of Corollary 7.2, we can guarantee that the solution of several algebraic problems related to these Wronskian matrices can be computed with HRA.

Corollary 7.4. Let $W_N := W(b_0^n, \dots, b_n^n)(x)$ be the Wronskian matrix of the negative binomial basis defined in (7.11) and J the diagonal matrix $J := \text{diag}((-1)^{i-1})_{1 \leq i \leq n+1}$. Then, for any $x < 0$,

$$W_{N,J} := JW_N J$$

is TP and its bidiagonal factorization (7.1) can be computed with HRA. Consequently, the computation of the eigenvalues, singular values of W_N , the matrix W_N^{-1} , as well as the solution of linear systems $W_N x = b$, where the entries of $b = (b_1, \dots, b_{n+1})^T$ have the same sign, can be performed with HRA.

Section 7.5 will show accurate computations with the Wronskian matrices of Bernstein bases, Bernstein bases of negative degree and negative binomial bases obtained by using the bidiagonal decomposition (7.1) and the algorithms in [62].

7.5 Numerical experiments

Given a nonsingular and TP matrix whose bidiagonal factorization (7.1) can be computed with HRA, the functions `TNEigenValues`, `TNSingularValues`, `TNInverseExpand` and `TNSolve`, available in the library `TNTool` of [63], can be used to compute with HRA its eigenvalues, singular values, its inverse matrix and the solution of some linear systems, respectively. The computational cost of the aforementioned functions is $O(n^2)$ elementary operations for `TNSolve` and $O(n^3)$ for the other functions.

Using the results in this chapter, we have implemented Matlab functions for the efficient computation of the bidiagonal decomposition (7.1) of the TP matrices $J\mathbf{W}_nJ$, where \mathbf{W}_n are $(n+1) \times (n+1)$ Wronskian matrices of Bernstein and negative binomial bases or \mathbf{W}_nJ , where \mathbf{W}_n are $(n+1) \times (n+1)$ Wronskian matrices of Bernstein bases of negative degree. In order to use the functions available in the library `TNTool` of [63], all the implemented Matlab functions give the bidiagonal decomposition (7.1) for the corresponding matrices by means of the $(n+1) \times (n+1)$ matrix $BD(\cdot)$ defined in (7.3). Observe that the computational complexity of the computation of the multipliers $m_{i,j}$, $\tilde{m}_{i,j}$ and the pivots $p_{i,i}$ of the proposed bidiagonal decompositions is $O(n^2)$.

We have considered Wronskian matrices \mathbf{W}_n at $x = -1$, $x = 1/7$, $x = -2$ and $x = -40$. Table 7.1 illustrates the 2-norm condition number of \mathbf{W}_n , computed with the Mathematica command `Norm[A, 2] · Norm[Inverse[A], 2]`. Observe that the condition number of the matrices considerably increases with their dimension. Due to this ill conditioning, traditional methods do not achieve accurate solutions when solving the mentioned algebraic problems. The following numerical results confirm this fact and illustrate the high accuracy obtained when using the functions in [63] with the bidiagonal factorizations (7.1) of the matrices $J\mathbf{W}_nJ$, or \mathbf{W}_nJ provided in this chapter.

Table 7.1: Condition number of Wronskian matrices of Bernstein bases at $x_0 = -1$, Wronskian matrices of Bernstein bases of negative degree at $x_0 = 1/7$ and Wronskian matrices of negative binomial bases at $x_0 = -2$.

$n+1$	$\kappa_2(\mathbf{W}_n)$	$\kappa_2(\mathbf{W}_n)$	$\kappa_2(\mathbf{W}_n)$
10	1.3×10^9	1.7×10^{15}	1.4×10^{11}
15	1.3×10^{16}	4.3×10^{23}	1.0×10^{18}
20	1.6×10^{21}	2.6×10^{31}	2.3×10^{23}
25	9.9×10^{25}	3.4×10^{37}	3.3×10^{26}

We have compared the eigenvalues and singular values obtained when using the Matlab commands `eig` and `svd`, respectively, and those computed using the bidiagonal decompositions (7.1) in this chapter and the Matlab functions `TNEigenValues` and `TNSingularValues`, respectively. In order to determine the accuracy of the approximations, we have also calculated the eigenvalues and singular values of the matrices by using Mathematica with a precision of 100 digits and computed the relative errors corresponding to the approximations, considering the eigenvalues and singular values provided by Mathematica as exact. We have computed the relative error of the approximations a of the exact eigenvalue and singular value \tilde{a} by means of the formula $e = |a - \tilde{a}|/|a|$.

Tables 7.2 and 7.3, show the relative errors of the approximations to the lowest eigenvalue and the lowest singular value obtained with both methods. Observe that the eigenvalues and singular values obtained using the factorization (7.1) are very accurate for all considered n , whereas the approximations of the eigenvalues and singular values obtained with the Matlab commands `eig` and `svd` are not very accurate when n increases.

Table 7.2: Relative errors when computing the lowest eigenvalue of the Wronskian matrices of Bernstein bases at $x_0 = -1$ and negative binomial bases at $x_0 = -2$.

n+1	eig	TNEV	eig	TNEV
10	3.0×10^{-10}	6.9×10^{-16}	1.9×10^{-6}	8.0×10^{-16}
15	1.9×10^{-4}	9.9×10^{-17}	3.1×10^{-7}	1.0×10^{-17}
20	2.8×10^1	4.4×10^{-16}	1.9×10^4	6.5×10^{-17}
25	8.8×10^6	4.9×10^{-16}	1.5×10^8	5.2×10^{-16}

Table 7.3: Relative errors when computing the lowest singular value of Wronskian matrices of Bernstein bases at $x_0 = -1$, Bernstein bases of negative degree at $x_0 = 1/7$ and negative binomial bases at $x_0 = -2$.

n+1	svd	TNSV	svd	TNSV	svd	TNSV
10	2.4×10^{-8}	3.6×10^{-19}	2.1×10^{-2}	1.2×10^{-15}	3.0×10^{-6}	1.2×10^{-15}
15	1.5×10^{-1}	3.0×10^{-16}	7.8×10^3	1.3×10^{-15}	5.4×10^{-7}	5.8×10^{-16}
20	3.2×10^3	5.2×10^{-16}	1.6×10^7	1.1×10^{-15}	8.8×10^2	8.6×10^{-16}
25	9.1×10^5	1.0×10^{-16}	5.6×10^{13}	4.3×10^{-15}	3.7×10^9	5.7×10^{-16}

We have also used the Matlab function `TNInverseExpand` (see Section 4 of [87]) with the bidiagonal factorization (7.1) in order to compute the inverse of Wronskian matrices of the bases considered. We have also computed their approximations with the Matlab functions `inv`. In order to determine the accuracy of the approximations, we have calculated the inverse of these matrices by using Mathematica with a precision of 100 digits and computed the relative errors corresponding to the approximations, considering the inverse matrix provided by Mathematica as exact. We have computed the relative error of each approximation \tilde{A}^{-1} of the exact inverse matrix A^{-1} by means of the formula $e = \|A^{-1} - \tilde{A}^{-1}\|_2 / \|A^{-1}\|_2$.

The approximation of the inverse of the Wronskian matrices obtained by means of `TNInverseExpand` is very accurate for all considered n , providing much more accurate results than those obtained by Matlab using the command `inv`. Table 7.4 shows the relative errors of the approximations to the inverse of the Wronskian matrices obtained with both methods.

Table 7.4: Relative errors when computing the inverse of Wronskian matrices of Bernstein bases at $x_0 = -1$, Bernstein bases of negative degree at $x_0 = 1/7$ and negative binomial bases at $x_0 = -2$.

n+1	inv	TNIE	inv	TNIE	inv	TNIE
10	6.6×10^{-11}	3.2×10^{-17}	6.8×10^{-11}	8.2×10^{-15}	1.5×10^{-8}	3.7×10^{-17}
15	1.3×10^{-6}	3.6×10^{-17}	1.2×10^{-6}	1.5×10^{-15}	2.2×10^{-2}	6.9×10^{-17}
20	3×10^{-1}	3.8×10^{-17}	6.5×10^{-2}	1.8×10^{-15}	2.3	1.8×10^{-16}
25	1.2	3.5×10^{-17}	5.7×10^{-1}	2.4×10^{-15}	1.0	1.3×10^{-16}

Finally, we shall illustrate the accuracy of the solutions of linear systems computed by using the bidiagonal factorization (7.1) with the function `TNSolve`. We have obtained the solution of the linear systems using Mathematica with a precision of 100 digits and considered this solution exact. Then we have also computed with Matlab two approximations, the first one using the previous functions and the second one using the Matlab command `\`. We have computed the relative error of every approximation $\tilde{c} = (\tilde{c}_1, \dots, \tilde{c}_{n+1})$ of the solution c of the linear system by means of the formula $e = \|c - \tilde{c}\|_2 / \|c\|_2$.

Table 7.5 shows the relative errors when solving the linear systems $\mathbf{W}_n c_n = \mathbf{d}_n$ where $\mathbf{d}_n = (d_i)_{1 \leq i \leq n+1}$

or $\mathbf{d}_n = ((-1)^{i-1}d_i)_{1 \leq i \leq n+1}$, in the case of Bernstein basis of negative degree, and $d_i, i = 1, \dots, n+1$, are random nonnegative integer values. The computed results confirm the accuracy of the proposed method that, clearly, keeps the accuracy when the dimension of the problem increases. In contrast, when n increases the condition number of the considered matrices considerably increases and that explains the bad results obtained with the Matlab command `\`.

Table 7.5: Relative errors when solving $\mathbf{W}_n \mathbf{c}_n = \mathbf{d}_n$ with Wronskian matrices of Bernstein bases at $x_0 = -1$, Bernstein bases of negative degree at $x_0 = 1/7$ and negative binomial bases at $x_0 = -2$.

n+1	$\mathbf{W}_n \setminus \mathbf{d}_n$	TNsolve	$\mathbf{W}_n \setminus \mathbf{d}_n$	TNsolve	$\mathbf{W}_n \setminus \mathbf{d}_n$	TNsolve
10	6.7×10^{-14}	1.4×10^{-16}	8.9×10^{-11}	9.2×10^{-16}	1.5×10^{-8}	4.4×10^{-17}
15	3.1×10^{-11}	1.5×10^{-16}	1.6×10^{-6}	1.6×10^{-16}	2.2×10^{-2}	5.9×10^{-17}
20	1.9×10^{-10}	3.7×10^{-15}	8.4×10^{-2}	2.0×10^{-15}	2.0	3.1×10^{-17}
25	1.3×10^{-8}	1.5×10^{-15}	7.1×10^{-1}	2.6×10^{-15}	1.0	7.7×10^{-17}

8

Total positivity and accurate computations with Wronskian matrices of geometric and Poisson bases

ABOUT THIS CHAPTER

The purpose of this chapter is to present some of the latest results that we have obtained, which are not included in the articles that belong to the compendium of publications of this thesis. It should be noted that, in this chapter, we have taken into account some of the results shown in the article [75] (see on page 99).

8.1 Introduction

The geometric distribution has applications in population and econometric models and the Poisson distribution is popular for modeling the number of times an event occurs in an interval of time or space. Associated to these distributions, the corresponding bases can be defined (see sections 8.3 and 8.4, respectively). The Poisson basis also plays a useful role in Computer-Aided Geometric Design (see [42] and [90]). Collocation matrices of both bases were analyzed in [74] (see the article on page 59), where it was proved their total positivity, and their bidiagonal factorization was obtained with high relative accurate (HRA). Starting with such bidiagonal factorization, the algorithms presented in [63] could be applied to calculate with HRA algebraic computations such as their inverses, their eigenvalues or their singular values and the solutions of some linear systems. Other examples of collocation matrices for which a bidiagonal factorization was obtained can be seen in [15],[17],[71],[82], [83].

Wronskian matrices arise in many applications. For instance, in Hermite interpolation problems, and in particular in Taylor interpolation problems. In [75] (see the article on page 99), the bidiagonal decomposition of the Wronskian matrices of the monomial basis was obtained. In this chapter, we deal with the Wronskian matrices of the two kind of bases mentioned in the previous paragraph. In contrast with their corresponding collocation matrices or with the Wronskian matrices of the monomial basis, we show that these Wronskian matrices are not totally positive. However, we relate them with other totally positive matrices, so that their associated bidiagonal factorizations can be used to provide accurate algorithms for the algebraic computations mentioned before. The complexity of the proposed algorithms for solving the mentioned algebraic problems is comparable to that of the traditional LAPACK algorithms, which, as we shall illustrate, deliver no such accuracy.

We now describe the layout of the chapter. In Section 8.2, we present basic notations and preliminary results. The bidiagonal factorization of Wronskian matrices of geometric bases is obtained in Section

Remark 8.1. *Observe that, by Theorem 4.3 of [37], the positivity of all multipliers and diagonal pivots of the Neville elimination of A in Theorem 1 implies that A is STP.*

Let us also recall that $x \in \mathbb{R}$ is obtained with high relative accuracy (HRA) if the relative error of the computed value \tilde{x} satisfies

$$\frac{\|x - \tilde{x}\|}{\|x\|} < Ku,$$

where K is a positive constant independent of the arithmetic precision and u is the unit round-off. HRA implies that the relative errors of the computations are of the order of the machine precision. It is known that an algorithm can be computed with HRA when it only uses products, quotients, sums of numbers of the same sign, subtractions of numbers of opposite sign or subtraction of initial data (cf. [23], [61]).

If the bidiagonal factorization (8.1) of a nonsingular TP matrix A is computed with HRA then, using the algorithms in [62], we can also compute with HRA its eigenvalues and singular values, the matrix A^{-1} and even the solution of $Ax = b$ for vectors b with alternating signs.

Let us denote by \mathbf{P}^n the space of polynomials of degree less than or equal to n and (p_0, \dots, p_n) the monomial basis of \mathbf{P}^n such that

$$p_i(x) := x^i, \quad i = 0, \dots, n. \quad (8.4)$$

The following result restates Corollary 1 of [75] (see the article on page 99), providing the bidiagonal factorization (8.1) of the Wronskian matrix $W(p_0, \dots, p_n)(x)$, $x \in \mathbb{R}$.

Proposition 8.1. *Let (p_0, \dots, p_n) be the monomial basis given in (8.4). For any $x \in \mathbb{R}$, the Wronskian matrix $W(p_0, \dots, p_n)(x)$ is nonsingular and can be factorized as follows,*

$$W(p_0, \dots, p_n)(x) = DG_{1,n} \cdots G_{n-1,n-1} G_{n,n}, \quad (8.5)$$

where $D = \text{diag}\{0!, 1!, \dots, n!\}$ and $G_{i,n}$, $i = 1, \dots, n$, are the upper triangular bidiagonal matrices in (8.2) with

$$\tilde{m}_{k,k-i} = x, \quad i+1 \leq k \leq n+1. \quad (8.6)$$

Moreover, if $x > 0$ then $W(p_0, \dots, p_n)(x)$ is nonsingular and TP, its bidiagonal decomposition (8.1) is given by (8.5) and (8.6) and it can be computed with HRA.

In [75] (see the article on page 99), using this result, accurate computations with Wronskian matrices of monomial bases are achieved.

In the following sections we shall obtain the bidiagonal factorization (8.1) of Wronskian matrices of Poisson and geometric basis functions. For all considered cases, we are going to achieve algebraic computations with HRA.

8.3 Bidiagonal factorization of Wronskian matrices of geometric bases

The geometric distribution has many applications in population and econometric models. Let us recall that the probability of k failures up to obtain a success is given by

$$P(k \text{ failures until a success}) := (1-t)^k t,$$

where the probability of success is $t \in [0, 1]$. Then, for a given $n \in \mathbb{N}$, we can define an $(n+1)$ -dimensional polynomial basis (g_0, \dots, g_n) , where

$$g_k(x) := (1-x)^k x, \quad k = 0, \dots, n. \quad (8.7)$$

In Section 4 of [74], it is proved that the collocation matrix at positive values $0 < x_1 < \dots < x_{n+1} < 1$ of (g_0, \dots, g_n) is STP. Furthermore, the bidiagonal factorization of the collocation matrix $(g_{j-1}(x_{i-1}))_{1 \leq i, j \leq n+1}$ is deduced by taking into account that each basis function $g_k(x)$ can be obtained by scaling the polynomials $(1-x)^k$ with the positive function $\varphi(x) = x$, $k = 0, \dots, n$. Using this factorization, HRA algebraic computations with collocation matrices of geometric bases (g_0, \dots, g_n) are achieved.

In this section we are going to deduce a bidiagonal decomposition of the form (8.1) of the Wronskian matrix of geometric bases (8.7). We shall see that, using this factorization and the algorithms in [62], many algebraic problems related to these matrices can be solved with HRA.

Let us start with some auxiliary results.

Lemma 8.1. *For a given $t \in \mathbb{R}$ and $n \in \mathbb{N}$, let $U_{k,n} = (u_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, be the $(n+1) \times (n+1)$, upper triangular bidiagonal matrix with unit diagonal entries, such that*

$$u_{i-1,i}^{(k,n)} := 0, \quad i = 2, \dots, k, \quad u_{i-1,i}^{(k,n)} := t, \quad i = k+1, \dots, n+1.$$

Then, $U_n := U_{1,n} \cdots U_{n,n}$, is an upper triangular matrix and

$$U_n = (u_{i,j}^{(n)})_{1 \leq i, j \leq n+1}, \quad u_{i,j}^{(n)} = \binom{j-1}{i-1} t^{j-i}, \quad 1 \leq i \leq j \leq n+1. \quad (8.8)$$

Proof. Clearly, U_n is an upper triangular matrix since it is the product of upper triangular bidiagonal matrices. Taking into account Proposition 8.1, we can deduce that

$$U_n = \left((i-1)! (p_{j-1}(t))^{(i-1)} \right)_{i,j=1, \dots, n+1},$$

where $p_j(t) := t^j$, $j = 0, \dots, n$. Finally, taking into account that

$$(p_j(t))^{(i)} = \binom{j}{i} t^{j-i}, \quad 0 \leq i \leq j \leq n,$$

equalities (8.8) are immediately obtained. \square

Theorem 8.2. *Let (g_0, \dots, g_n) be the $(n+1)$ -dimensional geometric basis defined in (8.7). The Wronskian matrix $W := W(g_0, \dots, g_n)(x)$ at a given $x \in \mathbb{R}$, $x \neq 0$, admits a factorization of the form*

$$W = L_n D_n U_{1,n} \cdots U_{n-1,n} U_{n,n}, \quad (8.9)$$

where $L_n = (l_{i,j}^{(n)})_{1 \leq j, i \leq n+1}$ is the lower triangular bidiagonal matrix with unit diagonal entries, such that

$$l_{i,i-1}^{(n)} = \frac{i-1}{x}, \quad i = 2, \dots, n+1, \quad (8.10)$$

$U_{k,n} = (u_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, are the upper triangular bidiagonal matrices with unit diagonal entries, such that

$$u_{i-1,i}^{(k,n)} = 0, \quad i = 2, \dots, k, \quad u_{i-1,i}^{(k,n)} = 1-x, \quad i = k+1, \dots, n+1, \quad (8.11)$$

and D_n is the diagonal matrix $D_n = \text{diag}(d_{1,n}, \dots, d_{n+1,n})$ with

$$d_{i,n} = (-1)^{i-1} (i-1)! x, \quad i = 1, \dots, n+1. \quad (8.12)$$

Proof. First, let us observe that $L_n D_n = (\tilde{l}_{i,j}^{(n)})_{1 \leq i, j \leq n+1}$ is the lower triangular bidiagonal matrix such that

$$\tilde{l}_{i,i}^{(n)} = (-1)^{i-1} (i-1)! x, \quad i = 1, \dots, n+1, \quad \tilde{l}_{i,i-1}^{(n)} = (-1)^i (i-1)!, \quad i = 2, \dots, n+1. \quad (8.13)$$

On the other hand, using Lemma 8.1 with $t = 1 - x$, we derive that $U_n := U_{1,n} \cdots U_{n,n}$ is the $(n+1) \times (n+1)$, upper triangular matrix described by

$$U_n = (u_{i,j}^{(n)})_{1 \leq i, j \leq n+1}, \quad u_{i,j}^{(n)} = \binom{j-1}{i-1} (1-x)^{j-i}, \quad 1 \leq i \leq j \leq n+1. \quad (8.14)$$

In order to prove the result, taking into account (8.9), (8.13) and (8.14), we have to check that

$$(g_{j-1})^{(i-1)}(x) = \begin{cases} 0, & j = 1, \dots, i-2, \\ (-1)^i (i-1)!, & j = i-1, \\ (-1)^{i-1} (i-1)! (1-i+ix), & j = i, \\ (-1)^{i-1} (i-1)! \frac{1}{j-i+1} \binom{j-1}{i-1} (1-x)^{j-i} (1-i+jx), & j > i, \end{cases} \quad (8.15)$$

for $1 \leq i, j \leq n+1$. Since

$$(g_{j-1})^{(i-1)}(x) = \left(\sum_{k=0}^{j-1} \binom{j-1}{k} (-1)^k x^{k+1} \right)^{(i-1)},$$

equalities (8.15) readily follow for $1 \leq j \leq i$ and $i = 1, \dots, n+1$. For $j > i$, (8.15) can be proved by induction on i . If $i = 1$, we clearly have $(-1)^0 (1-1)! \binom{j-1}{0} (1-x)^{j-1} jx/j = g_{j-1}(x)$, for $j = 2, \dots, n+1$. Now let us suppose that (8.15) holds for $i > 1$ and $j > i$. Then, we can write

$$\begin{aligned} (g_{j-1})^{(i)}(x) &= (-1)^{i-1} (i-1)! \frac{1}{j-i+1} \binom{j-1}{i-1} ((1-x)^{j-i} (1-i+jx))' \\ &= (-1)^i (i-1)! \binom{j-1}{i-1} (1-x)^{j-i-1} (-i+jx). \end{aligned} \quad (8.16)$$

Using (8.16), since $(i-1)! \binom{j-1}{i-1} = i! \binom{j-1}{i} \frac{1}{j-i}$, we have

$$(g_{j-1})^{(i)}(x) = (-1)^i i! \frac{1}{j-i} \binom{j-1}{i} (1-x)^{j-i-1} (-i+jx),$$

$j = i+1, \dots, n$, and consequently (8.15) follows. \square

Example 8.1. Let us illustrate the bidiagonal factorization (8.9) of the Wronskian matrix of geometric bases with a simple example. For the particular case $n = 2$, the bidiagonal factorization of the Wronskian matrix of the basis $(x, x(1-x), x(1-x)^2)$ at $x \in \mathbb{R}$ is

$$W(x, x(1-x), x(1-x)^2) = \begin{pmatrix} 1 & 0 & 0 \\ 1/x & 1 & 0 \\ 0 & 2/x & 1 \end{pmatrix} \begin{pmatrix} x & 0 & 0 \\ 0 & -x & 0 \\ 0 & 0 & 2x \end{pmatrix} \begin{pmatrix} 1 & 1-x & 0 \\ 0 & 1 & 1-x \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1-x \\ 0 & 0 & 1 \end{pmatrix}.$$

Let us observe that, from Theorem 8.2, the bidiagonal factorization (8.1) of the $(n+1) \times (n+1)$ dimensional Wronskian matrix W of the geometric basis can be represented by means of the $(n+1) \times (n+1)$ matrix $BD(W) = (BD(W)_{i,j})_{1 \leq i,j \leq n+1}$ such that

$$BD(W)_{i,j} := \begin{cases} (i-1)/x, & \text{if } i = j+1, \\ (-1)^{i-1}(i-1)!x, & \text{if } i = j, \\ 1-x, & \text{if } i < j, \\ 0, & \text{else.} \end{cases} \quad (8.17)$$

Analyzing the sign of the entries in (8.17), we can deduce from Theorem 8.1 that the Wronskian matrix of the geometric basis (8.7) is not TP for any $x \in \mathbb{R}$. However, the following result shows that, using the bidiagonal decomposition (8.9), the solution of several algebraic problems related to these matrices can be obtained with HRA .

Corollary 8.1. *Let $W := W(g_0, \dots, g_n)(x)$ be the Wronskian matrix of the geometric basis defined in (8.7) and J the diagonal matrix $J := \text{diag}((-1)^{i-1})_{1 \leq i \leq n+1}$. Then, for $x > 1$,*

$$W_J := WJ$$

is a TP matrix and its bidiagonal factorization (8.1) can be computed with HRA. Consequently, the computation of the singular values of W , the matrix W^{-1} , as well as the solution $c = (c_1 \dots, c_{n+1})^T$ of linear systems $Wc = b$, where the entries of $b = (b_1 \dots, b_{n+1})^T$ have alternating signs, can be performed with HRA.

Proof. Using (8.9) and the fact that J^2 is the identity matrix, we can write

$$W_J = L_n(D_n J)(JU_{1,n}J) \cdots (JU_{n,n}J), \quad (8.18)$$

which gives the bidiagonal factorization (8.1) of W_J . Now, it can be easily checked that if $x-1 > 0$, the bidiagonal matrices $L_n, JU_{i,n}J, i = 1, \dots, n$, as well as the diagonal matrix $D_n J$ are TP. By Theorem 1, W_J is TP for $x > 1$. This fact guarantees the computation with HRA of its bidiagonal decomposition (8.1) and so, the computation with HRA of its eigenvalues and singular values, the inverse matrix W_J^{-1} and the solution of the linear systems $W_J c = b$, where $b = (b_0, \dots, b_n)^T$ has alternating signs (see Section 3 of [23]).

On the other hand, since J is a unitary matrix, the singular values of W_J coincide with those of W and so, their computation for $x > 1$ can be performed with HRA. Similarly, taking into account that

$$W^{-1} = JW_J^{-1},$$

we can compute W^{-1} accurately. Finally, if we have a linear system of equations $Wc = b$, where the elements of $b = (b_1, \dots, b_n)^T$ have alternating signs, we can solve with HRA the system $W_J d = b$ and then obtain $c = Jd$. \square

8.4 Bidiagonal factorization of Wronskian matrices of Poisson bases

The Poisson distribution is popular for modeling the number of times an event occurs in an interval of time or space. An event can occur $k = 0, 1, 2, \dots$ times in an interval. If the average number of events in

an interval, also called the rate parameter, is designated by t , then the probability of observing k events in an interval is given by

$$P(k \text{ events in interval}) = \frac{t^k}{k!} e^{-t}.$$

The Poisson basis functions

$$P_k(x) := \frac{x^k}{k!} e^{-x}, \quad k \in \mathbb{N}, \quad (8.19)$$

are the limit as n tends to infinity of the Bernstein basis of degree n over the interval $[0, n]$, that is,

$$P_k(x) = \lim_{n \rightarrow \infty} B_k^n(x/n), \quad B_k^n(x) = \binom{n}{k} x^k (1-x)^{n-k}, \quad x \in [0, 1],$$

and they also play a useful role in CAGD (see [90]).

In Section 4 of [74], it is proved that the collocation matrix at positive values $x_1 < \dots < x_{n+1}$ of a basis of Poisson functions (P_0, \dots, P_n) is STP on $(0, \infty)$. Furthermore, the bidiagonal factorization of the collocation matrix $(P_{j-1}(x_{i-1}))_{1 \leq i, j \leq n+1}$ is deduced by taking into account that each basis function $P_k(x)$ can be obtained by scaling the polynomials $x^k/k!$ with the positive function $\varphi(x) = e^{-x}$, $k = 0, \dots, n$. Using this factorization, accurate algebraic computations with collocation matrices of Poisson bases are achieved.

In this section we are going to deduce a bidiagonal decomposition of the form (8.1) of the Wronskian matrix of Poisson bases. We shall see that, using this factorization and the algorithms in [62], many algebraic problems related to these matrices can be solved with a great accuracy.

Let us start with some auxiliary results.

Lemma 8.2. *For a given $n \in \mathbb{N}$, let $L_{k,n} = (l_{i,j}^{(k,n)})_{1 \leq i, j \leq n+1}$, $k = 1, \dots, n$, be the lower triangular bidiagonal matrix with unit diagonal entries, such that*

$$l_{i,i-1}^{(k,n)} := 0, \quad i = 2, \dots, k, \quad l_{i,i-1}^{(k,n)} := -1, \quad i = k+1, \dots, n+1.$$

Then $L_n := L_{n,n} \cdots L_{1,n}$ is a lower triangular bidiagonal matrix with

$$L_n = (l_{i,j}^{(n)})_{1 \leq i, j \leq n+1}, \quad l_{i,j} := (-1)^{i+j} \binom{i-1}{j-1}, \quad 1 \leq j \leq i \leq n+1. \quad (8.20)$$

Proof. Clearly, L is a lower triangular matrix since it is the product of lower triangular bidiagonal matrices. Now let us prove (8.20) by induction on n . For $n = 1$,

$$L_1 = L_{1,1} = \begin{pmatrix} 1 & \\ -1 & 1 \end{pmatrix}$$

and (8.20) clearly holds. Let us now suppose that (8.20) holds for $n \geq 1$. Then

$$L_{n+1} := L_{n+1,n+1} \cdots L_{1,n+1} = \tilde{L}_{n+1} L_{1,n+1}$$

where $\tilde{L}_{n+1} := L_{n+1,n+1} \cdots L_{2,n+1}$ satisfies $\tilde{L}_{n+1} = (\tilde{l}_{i,j}^{(n+1)})_{1 \leq i, j \leq n+2}$ with $\tilde{l}_{i,1} = \delta_{i,1}$, $\tilde{l}_{1,i} = \delta_{1,i}$ for $i = 1, \dots, n+2$ and $\tilde{L}_{n+1}[2, \dots, n+2] = L_{n,n} \cdots L_{1,n}$. Then we have that

$$\tilde{l}_{i,j}^{(n+1)} = (-1)^{i+j} \binom{i-2}{j-2}, \quad 2 \leq j \leq i \leq n+2.$$

Now, taking into account that

$$L_{n+1} = \tilde{L}_{n+1}L_{1,n+1} = \tilde{L}_{n+1} \begin{pmatrix} 1 & & & & \\ -1 & 1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & -1 & 1 \end{pmatrix},$$

and the fact that $\binom{i-2}{j-2} + \binom{i-2}{j-1} = \binom{i-1}{j-1}$, we deduce that $L_{n+1} = (l_{i,j}^{(n+1)})_{1 \leq i,j \leq n+2}$ satisfies

$$l_{i,j}^{(n+1)} = \tilde{l}_{i,j}^{(n+1)} - \tilde{l}_{i,j+1}^{(n+1)} = (-1)^{i+j} \binom{i-2}{j-2} - (-1)^{i+j+1} \binom{i-2}{j-1} = (-1)^{i+j} \binom{i-1}{j-1}$$

for $1 \leq j \leq i \leq n+2$. □

Lemma 8.3. For a given $t \in \mathbb{R}$ and $n \in \mathbb{N}$, let $U_{k,n} = (u_{i,j}^{(k,n)})_{1 \leq i,j \leq n+1}$, $k = 1, \dots, n$, be the $(n+1) \times (n+1)$, upper triangular bidiagonal matrix with unit diagonal entries, such that

$$u_{i-1,i}^{(k,n)} := 0, \quad i = 2, \dots, k, \quad u_{i-1,i}^{(k,n)} := \frac{t}{i-1}, \quad i = k+1, \dots, n+1.$$

Then, $U_n := U_{1,n} \cdots U_{n,n}$, is an upper triangular matrix and

$$U_n = (u_{i,j}^{(n)})_{1 \leq i,j \leq n+1}, \quad u_{i,j}^{(n)} = \frac{t^{j-i}}{(j-i)!}, \quad 1 \leq i \leq j \leq n+1.$$

Proof. First, let us observe that given $\tilde{M}_n = (\tilde{m}_{i,j})_{1 \leq i,j \leq n+1}$ and a nonsingular diagonal matrix $D_n = \text{diag}(d_1, \dots, d_{n+1})$,

$$D_n \tilde{M}_n = M_n D_n,$$

where $M_n = (m_{i,j})_{1 \leq i,j \leq n+1}$ satisfies $m_{i,j} = \tilde{m}_{i,j} d_i / d_j$, $i, j = 1, \dots, n+1$. Now, let us define $D_n := \text{diag}(d_i)_{1 \leq i \leq n+1}$, such that $d_i = (i-1)!$, $i = 1, \dots, n+1$, and let $\tilde{U}_{k,n} = (\tilde{u}_{i,j}^{(k,n)})_{1 \leq i,j \leq n+1}$, $k = 1, \dots, n$, be the $(n+1) \times (n+1)$, upper triangular bidiagonal matrix with unit diagonal entries, such that

$$\tilde{u}_{i-1,i}^{(k,n)} := 0, \quad i = 2, \dots, k, \quad \tilde{u}_{i-1,i}^{(k,n)} := t, \quad i = k+1, \dots, n+1.$$

Taking into account the fact that $d_{i-1}/d_i = 1/(i-1)$, $i = 2, \dots, n+1$, we can write

$$D_n \tilde{U}_{1,n} \cdots \tilde{U}_{n,n} = U_{1,n} \cdots U_{n,n} D_n.$$

Consequently, $U_{1,n} \cdots U_{n,n} = D_n \tilde{U}_{1,n} \cdots \tilde{U}_{n,n} D_n^{-1}$ and, applying Lemma 8.1 to $\tilde{U}_{1,n} \cdots \tilde{U}_{n,n}$,

$$U_{1,n} \cdots U_{n,n} = (u_{i,j}^{(n)})_{1 \leq i,j \leq n+1}, \quad u_{i,j}^{(n)} = \frac{(i-1)!}{(j-1)!} \binom{j-1}{i-1} t^{j-i} = \frac{t^{j-i}}{(j-i)!}, \quad 1 \leq i \leq j \leq n+1.$$

□

Now, we can deduce a bidiagonal factorization of Wronskian matrices of Poisson bases.

Theorem 8.3. Let $n \in \mathbb{N}$ and (P_0, \dots, P_n) the basis (8.19) of Poisson functions. For a given $x \in \mathbb{R}$, $W := W(P_0, \dots, P_n)(x)$ admits a factorization of the form

$$W = L_{n,n}L_{n-1,n} \cdots L_{1,n}D_nU_{1,n} \cdots U_{n-1,n}U_{n,n}, \quad (8.21)$$

where $L_{k,n} = (l_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, are the lower triangular bidiagonal matrices with unit diagonal entries, such that

$$l_{i,i-1}^{(k,n)} = 0, \quad i = 2, \dots, k, \quad l_{i,i-1}^{(k,n)} = -1, \quad i = k+1, \dots, n+1,$$

$U_{k,n} = (u_{i,j}^{(k,n)})_{1 \leq j, i \leq n+1}$, $k = 1, \dots, n$, are the upper triangular bidiagonal matrices with unit diagonal entries, such that

$$u_{i-1,i}^{(k,n)} = 0, \quad i = 2, \dots, k, \quad u_{i-1,i}^{(k,n)} = \frac{x}{i-1}, \quad i = k+1, \dots, n+1,$$

and D_n is the diagonal matrix $D_n = \text{diag}(d_1, \dots, d_{n+1})$ with $d_i = e^{-x}$, $i = 1, \dots, n+1$.

Proof. By Lemma 8.2, $L_n := L_{n,n}L_{n-1,n} \cdots L_{1,n}$ is a lower triangular matrix and satisfies

$$L_n = (l_{i,j}^{(n)})_{1 \leq i, j \leq n+1}, \quad l_{i,j}^{(n)} = (-1)^{i+j} \binom{i-1}{j-1}, \quad 1 \leq j \leq i \leq n+1.$$

On the other hand, by Lemma 8.3, $U_n := U_{1,n} \cdots U_{n-1,n}U_{n,n}$ satisfies

$$U_n = (u_{i,j}^{(n)})_{1 \leq i, j \leq n+1}, \quad u_{i,j}^{(n)} = \frac{x^{j-i}}{(j-i)!}, \quad 1 \leq i \leq j \leq n+1.$$

Now, let us see that $W = L_n D_n U_n$, that is,

$$P_{j-1}^{(i-1)}(x) = \left(\sum_{k=1}^{\min\{i,j\}} (-1)^{i+k} \binom{i-1}{k-1} \frac{x^{j-k}}{(j-k)!} \right) e^{-x}, \quad 1 \leq i, j \leq n+1. \quad (8.22)$$

We shall prove (8.22) by induction on i . For $i = 1$,

$$\sum_{k=1}^1 (-1)^{1+k} \binom{1-1}{k-1} \frac{x^{j-k}}{(j-k)!} e^{-x} = \frac{x^{j-1}}{(j-1)!} e^{-x} = P_{j-1}(x), \quad j = 1, \dots, n+1,$$

and (8.22) holds. Now, let us assume that (8.22) holds for $i \geq 1$. For any j such that $1 \leq j \leq i$, it can be checked that

$$\left(\sum_{k=1}^j (-1)^{i+k} \binom{i-1}{k-1} \frac{x^{j-k}}{(j-k)!} e^{-x} \right)' = \left(\sum_{k=1}^j (-1)^{i+k+1} c_k \frac{x^{j-k}}{(j-k)!} \right) e^{-x}$$

where

$$c_1 = \binom{i-1}{0} = 1, \quad c_k = \binom{i-1}{k-1} + \binom{i-1}{k-2} = \binom{i}{k-1}, \quad k = 2, \dots, j.$$

In a similar way it can be checked that, for any $j > i$, we have

$$\left(\sum_{k=1}^i (-1)^{i+k} \binom{i-1}{k-1} \frac{x^{j-k}}{(j-k)!} e^{-x} \right)' = \left(\sum_{k=1}^{i+1} (-1)^{i+k+1} c_k \frac{x^{j-k}}{(j-k)!} \right) e^{-x}$$

where

$$c_1 = \begin{pmatrix} i-1 \\ 0 \end{pmatrix}, \quad c_k = \begin{pmatrix} i-1 \\ k-1 \end{pmatrix} + \begin{pmatrix} i-1 \\ k-2 \end{pmatrix} = \begin{pmatrix} i \\ k-1 \end{pmatrix}, \quad k = 2, \dots, i, \quad c_{i+1} := \begin{pmatrix} i-1 \\ i-1 \end{pmatrix}.$$

Therefore,

$$P_{j-1}^{(i)}(x) = \left(\sum_{k=1}^{\min\{i+1, j\}} (-1)^{i+k+1} \binom{i}{k-1} \frac{x^{j-k}}{(j-k)!} \right) e^{-x},$$

and (8.22) holds. □

Example 8.2. Let us illustrate the bidiagonal factorization (8.21) of the Wronskian matrix of a Poisson basis. For the particular case $n = 2$, the bidiagonal factorization of the Wronskian matrix of the basis $(e^{-x}, xe^{-x}, \frac{x^2}{2}e^{-x})$ at $x \in \mathbb{R}$ is

$$W(e^{-x}, xe^{-x}, \frac{x^2}{2}e^{-x}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} e^{-x} & 0 & 0 \\ 0 & e^{-x} & 0 \\ 0 & 0 & e^{-x} \end{pmatrix} \begin{pmatrix} 1 & x & 0 \\ 0 & 1 & \frac{x}{2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{x}{2} \\ 0 & 0 & 1 \end{pmatrix}.$$

Let us observe, that from Theorem 8.3, it can be deduced that the bidiagonal factorization (8.1) of the $(n+1) \times (n+1)$ dimensional Wronskian matrix W of the Poisson basis can be represented by means of the $(n+1) \times (n+1)$ matrix $BD(W) = (BD(W)_{i,j})_{1 \leq i, j \leq n+1}$ such that

$$BD(W)_{i,j} := \begin{cases} -1, & \text{if } i > j, \\ e^{-x}, & \text{if } i = j, \\ x/(j-1), & \text{if } i < j. \end{cases} \quad (8.23)$$

Analyzing the sign of the entries in (8.23), we can deduce from Theorem 8.1 that the Wronskian matrix of the Poisson basis is not TP for any $x \in \mathbb{R}$. However, the following result shows that the solution of several algebraic problems related to these matrices can be obtained with HRA by using the bidiagonal decomposition (8.21).

Corollary 8.2. Let $W := W(P_0, \dots, P_n)(x)$ be the Wronskian matrix of the Poisson basis defined in (8.19) and J the diagonal matrix $J := \text{diag}((-1)^{i-1})_{1 \leq i \leq n+1}$. Then, for any $x < 0$,

$$W_J := JWJ$$

is an STP matrix. If, in addition, we know e^{-x} with HRA, then the bidiagonal factorization (8.1) of W_J can be computed with HRA. Consequently, the computation of the eigenvalues, singular values of W , the matrix W^{-1} , as well as the solution $c = (c_1, \dots, c_{n+1})^T$ of linear systems $Wc = b$, where the entries of $b = (b_1, \dots, b_{n+1})^T$ have the same sign, can be performed with HRA.

Proof. Using Theorem 8.3 and that J^2 is the identity matrix, by (8.21) we can write

$$W_J = (JL_{n,n}J) \cdots (JL_{1,n}J)(JDJ)(JU_{1,n}J) \cdots (JU_{n,n}J), \quad (8.24)$$

which gives its bidiagonal factorization (8.1). Now, it can be easily checked that the multipliers and diagonal pivots of the bidiagonal factorization (8.24) of W_J are positive if $x < 0$. Therefore, by Remark 8.1, W_J is STP and its bidiagonal decomposition (8.24) can be computed with HRA for any $x < 0$. This fact guarantees the computation with HRA of the eigenvalues and singular values of W_J , the inverse matrix W_J^{-1} and the solution of the linear systems $W_J c = d$, where $d = (d_1, \dots, d_{n+1})^T$ has alternating signs (see Section 3 of [23]).

Let us observe that, since J is a unitary matrix, the eigenvalues and singular values of W coincide with those of W_J and therefore, using the bidiagonal decomposition (8.24) of W_J , their computation for $x < 0$ can be performed with HRA.

For the accurate computation of W^{-1} , we can take into account that

$$W^{-1} = JW_J^{-1}J. \quad (8.25)$$

Since, for $x < 0$, $W_J^{-1} = (\tilde{w}_{i,j})_{1 \leq i, j \leq n+1}$ can be computed with HRA and, by (8.25), the inverse of the Wronskian matrix W satisfies $W^{-1} = ((-1)^{i+j} \tilde{w}_{i,j})_{1 \leq i, j \leq n+1}$, we can also accurately compute W^{-1} by means of a suitable change of sign of the accurately computed entries of W_J^{-1} .

Finally, if we have a linear system of equations $Wc = b$, where the elements of $b = (b_1, \dots, b_{n+1})^T$ have the same sign, we can compute with HRA the solution $d \in \mathbb{R}^{n+1}$ of $W_J d = Jb$ and, consequently, the solution $c \in \mathbb{R}^{n+1}$ of the initial system since $c = Jd$. □

Observe that Corollary 8.2 requires the evaluation with HRA of e^{-x} . Even when this does not hold, Section 8.5 will show that the resolution of algebraic problems with W_J can be performed through the proposed bidiagonal factorization with an accuracy independent of the conditioning or the size of the problem and so, similar to HRA. Consequently, as in the proof of Corollary 8.2, we can deduce that the computation of the eigenvalues, singular values of W , the matrix W^{-1} , as well as the solution of linear systems $Wx = b$, where the entries of $b = (b_1, \dots, b_{n+1})^T$ have the same signs, can be performed with an accuracy similar to HRA.

8.5 Numerical experiments

Given a nonsingular and TP matrix whose bidiagonal factorization (8.1) can be computed with HRA, the functions `TNEigenValues`, `TNSingularValues`, `TNInverseExpand` and `TNSolve`, available in the library `TNTool` of [63], can be used to compute with HRA its eigenvalues, singular values, its inverse matrix and the solution of some linear systems, respectively. The computational cost of the aforementioned functions is $O(n^2)$ elementary operations for `TNSolve` and $O(n^3)$ for the other functions.

Using the results in this chapter, we have implemented Matlab functions for the efficient computation of the bidiagonal decomposition (8.1) of the TP matrices $\mathbf{W}_n J$, where \mathbf{W}_n are $(n+1) \times (n+1)$ Wronskian matrices of geometric bases or $J\mathbf{W}_n J$, where \mathbf{W}_n are $(n+1) \times (n+1)$ Wronskian matrices of Poisson bases. In order to use the functions available in the library `TNTool` of [63], all the implemented Matlab functions give the bidiagonal decomposition (8.1) for the corresponding matrices by means of the $(n+1) \times (n+1)$ matrix $BD(\cdot)$ defined in (8.3). Observe that the computational complexity of the computation of the multipliers $m_{i,j}$, $\tilde{m}_{i,j}$ and the pivots $p_{i,i}$ of the proposed bidiagonal decompositions is $O(n^2)$.

We have considered Wronskian matrices \mathbf{W}_n at $x = 10$ and $x = -40$. Table 8.1 illustrates the 2-norm condition number of \mathbf{W}_n , computed with the Mathematica command `Norm[A, 2] · Norm[Inverse[A], 2]`. Observe that the condition number of the matrices considerably increases with their dimension. Due to

this ill conditioning, traditional methods do not achieve accurate solutions when solving the mentioned algebraic problems. The following numerical results confirm this fact and illustrate the high accuracy obtained when using the functions in [63] with the bidiagonal factorizations (8.1) of the matrices $\mathbf{W}_n J$, or $J\mathbf{W}_n J$ provided in this chapter. ;

Table 8.1: Condition number of Wronskian matrices of geometric bases at $x_0 = 10$ and Poisson bases at $x_0 = -40$.

$n+1$	$\kappa_2(\mathbf{W}_n)$	$\kappa_2(\mathbf{W}_n)$
5	3.1×10^6	2.8×10^{11}
10	6.0×10^{12}	1.3×10^{21}
15	7.1×10^{18}	4.1×10^{26}
20	2.2×10^{25}	5.3×10^{29}

We have compared the eigenvalues and singular values obtained when using the Matlab commands `eig` and `svd`, respectively, and those computed using the bidiagonal decompositions (8.1) in this chapter and the Matlab functions `TNEigenValues` and `TNSingularValues`, respectively. In order to determine the accuracy of the approximations, we have also calculated the eigenvalues and singular values of the matrices by using Mathematica with a precision of 100 digits and computed the relative errors corresponding to the approximations, considering the eigenvalues and singular values provided by Mathematica as exact. We have computed the relative error of the approximations a of the exact eigenvalue and singular value \tilde{a} by means of the formula $e = |a - \tilde{a}|/|a|$.

Tables 8.2 and 8.3, show the relative errors of the approximations to the lowest eigenvalue and the lowest singular value obtained with both methods. Observe that the eigenvalues and singular values obtained using the factorization (8.1) are very accurate for all considered n , whereas the approximations of the eigenvalues and singular values obtained with the Matlab commands `eig` and `svd` are not very accurate when n increases.

Table 8.2: Relative errors when computing the lowest eigenvalue of the Wronskian matrices of Poisson bases at $x_0 = -40$.

$n+1$	<code>eig</code>	TNEV
5	6.2×10^{-8}	3.1×10^{-16}
10	9.3×10^1	5.2×10^{-16}
15	4.5×10^6	2.6×10^{-16}
20	4.8×10^{11}	4.1×10^{-16}

Table 8.3: Relative errors when computing the lowest singular value of Wronskian matrices of geometric bases at $x_0 = 10$ (left) and Poisson bases at $x_0 = -40$ (right).

$n+1$	<code>svd</code>	TNSV	<code>svd</code>	TNSV
5	4.1×10^{-1}	1.9×10^{-16}	8.0×10^{-9}	3.9×10^{-17}
10	1.7×10^2	8.9×10^{-16}	2.5×10^3	6.6×10^{-16}
15	5.4×10^2	1.9×10^{-15}	4.4×10^3	4.8×10^{-16}
20	1.2×10^2	8.0×10^{-16}	1.4×10^{10}	1.3×10^{-15}

We have also used the Matlab function `TNInverseExpand` (see Section 4 of [87]) with the bidiago-

nal factorization (8.1) in order to compute the inverse of Wronskian matrices of the bases considered. We have also computed their approximations with the Matlab functions `inv`. In order to determine the accuracy of the approximations, we have calculated the inverse of these matrices by using Mathematica with a precision of 100 digits and computed the relative errors corresponding to the approximations, considering the inverse matrix provided by Mathematica as exact. We have computed the relative error of each approximation \tilde{A}^{-1} of the exact inverse matrix A^{-1} by means of the formula $e = \|A^{-1} - \tilde{A}^{-1}\|_2 / \|A^{-1}\|_2$.

The approximation of the inverse of the Wronskian matrices obtained by means of `TNInverseExpand` is very accurate for all considered n , providing much more accurate results than those obtained by Matlab using the command `inv`. Table 8.4 shows the relative errors of the approximations to the inverse of the Wronskian matrices obtained with both methods.

Table 8.4: Relative errors when computing the inverse of Wronskian matrices of geometric bases at $x_0 = 10$ (left) and Poisson bases at $x_0 = -40$ (right).

n+1	inv(W_n)	TNIE(BDGWJ_n)	inv(W_n)	TNIE(BDPJWJ_n)
5	1.8×10^{-15}	4.2×10^{-17}	6.5×10^{-11}	1.2×10^{-16}
10	1.6×10^{-13}	7.0×10^{-17}	2.3×10^{-3}	2.9×10^{-16}
15	4.7×10^{-11}	1.0×10^{-16}	1.0	4.3×10^{-16}
20	1.4×10^{-9}	1.6×10^{-16}	1.0	6.0×10^{-16}

Finally, we shall illustrate the accuracy of the solutions of linear systems computed by using the bidiagonal factorization (8.1) with the function `TNSolve`. We have obtained the solution of the linear systems using Mathematica with a precision of 100 digits and considered this solution exact. Then we have also computed with Matlab two approximations, the first one using the previous functions and the second one using the Matlab command `\`. We have computed the relative error of every approximation $\tilde{c} = (\tilde{c}_1, \dots, \tilde{c}_{n+1})$ of the solution c of the linear system by means of the formula $e = \|c - \tilde{c}\|_2 / \|c\|_2$.

Table 8.5 shows the relative errors when solving the linear systems $\mathbf{W}_n \mathbf{c}_n = \mathbf{d}_n$ where $\mathbf{d}_n = ((-1)^{i+1} d_i)_{1 \leq i \leq n+1}$, in the case of geometric bases, or $\mathbf{d}_n = (d_i)_{1 \leq i \leq n+1}$, in the case of Poisson bases, and $d_i, i = 1, \dots, n+1$, are random nonnegative integer values. The computed results confirm the accuracy of the proposed method that, clearly, keeps the accuracy when the dimension of the problem increases. In contrast, when n increases the condition number of the considered matrices considerably increases and that explains the bad results obtained with the Matlab command `\`.

Table 8.5: Relative errors when solving $\mathbf{W}_n \mathbf{c}_n = \mathbf{d}_n$ with Wronskian matrices of geometric bases at $x_0 = 10$ (left) and Poisson bases at $x_0 = -40$ (right).

n+1	W_n \ d_n	TNSolve	W_n \ d_n	TNSolve
5	1.5×10^{-15}	7.8×10^{-17}	8.2×10^{-3}	1.1×10^{-16}
10	1.8×10^{-13}	2.9×10^{-16}	2.4×10^{-3}	1.3×10^{-16}
15	4.3×10^{-11}	3.8×10^{-16}	1.0	4.2×10^{-17}
20	2.7×10^{-9}	1.1×10^{-16}	1.0	4.4×10^{-17}

Part V

CONCLUSIONS AND FUTURE WORK

Conclusiones

Esta tesis doctoral se ha enmarcado dentro de la teoría de la Positividad Total, concretamente en dos de los campos que están relacionados con las matrices totalmente positivas. Por un lado, se ha desarrollado dentro del campo del Diseño Geométrico Asistido por Ordenador. Particularmente, se han estudiado aspectos relacionados con bases totalmente positivas y se han explorado bases que heredan propiedades de preservación de forma a partir de una dada. Por otra parte, también se ha desarrollado dentro del campo del Álgebra Lineal Numérica. En concreto, se ha diseñado y analizado algoritmos adaptados a la estructura de diferentes matrices totalmente positivas que han permitido resolver con alta precisión relativa problemas algebraicos asociados a estas matrices. A continuación señalaremos las principales aportaciones de esta tesis.

- Dado un sistema de funciones inicial, un conjunto de pesos y una función positiva φ , hemos definido un nuevo sistema de funciones llamado sistema φ -transformado ponderado (*weighted φ -transformed system*), el cual incluye una amplia clase de representaciones útiles en Estadística y Diseño Geométrico Asistido por Ordenador. Se ha demostrado que estos sistemas heredan algunas propiedades geométricas de sus sistemas iniciales, como las propiedades de preservación de forma o las propiedades óptimas de preservación de forma. Hemos mostrado que se puede obtener una clase general de bases racionales importantes como un ejemplo particular de sistemas φ -transformados ponderados. Para estas bases se han presentado algoritmos de evaluación y subdivisión. Además, se han señalado algunas aplicaciones relevantes.
- Hemos obtenido la factorización bidiagonal de las matrices de colocación de los sistemas φ -transformados ponderados. Esta factorización bidiagonal se ha utilizado para obtener métodos con alta precisión relativa para la resolución de problemas algebraicos con las matrices de colocación de estos sistemas, como el cálculo de valores propios, valores singulares y la solución de algunos sistemas lineales. Los ejemplos numéricos han ilustrado la precisión de los cálculos realizados.
- Hemos abordado el problema de encontrar una curva racional que se ajuste a un conjunto dado de puntos. Para solucionar este asunto, hemos aplicado técnicas de Inteligencia Artificial y hemos propuesto una red neuronal de una capa oculta basada en curvas racionales generadas por una clase general de bases racionales pertenecientes a espacios que mezclan polinomios algebraicos, trigonométricos e hiperbólicos, pudiendo así alcanzar formas más difíciles y ampliando de esta manera el rango potencial de aplicaciones de esta red neuronal. Para obtener los pesos y puntos de control de la curva racional de ajuste, la red neuronal se entrena con un algoritmo de optimización que actualiza los pesos y los puntos de control mientras disminuye una función de pérdida. Las curvas de ajuste obtenidas en los experimentos numéricos han demostrado que para ciertos conjuntos de puntos el uso de bases racionales particulares proporciona mejores resultados.

- Hemos obtenido algoritmos precisos para calcular la factorización bidiagonal de la matriz wronskiana de la base de los monomios y la factorización bidiagonal de la matriz wronskiana de la base de polinomios exponenciales. También se ha demostrado que estos algoritmos pueden utilizarse para realizar con precisión algunos cálculos algebraicos con estas matrices wronskianas, como el cálculo de sus inversas, sus valores propios o sus valores singulares y las soluciones de algunos sistemas lineales. Experimentos numéricos han ilustrado los resultados.
- Hemos diseñado un método preciso para hallar la factorización bidiagonal de las matrices de colocación y wronskianas de los polinomios de Jacobi. Hemos utilizado el método mencionado para calcular con alta precisión relativa sus inversas, sus valores propios, sus valores singulares y las soluciones de algunos sistemas lineales. Se han considerado también los casos particulares de las matrices de colocación y wronskianas de los polinomios de Legendre, los polinomios de Gegenbauer, los polinomios de Chebyshev de primer y segundo tipo y los polinomios racionales de Jacobi. Los ejemplos numéricos han ilustrado la precisión de los cálculos realizados.
- Hemos obtenido un método para obtener la factorización bidiagonal de la matriz wronskiana de los polinomios de Bessel y la factorización bidiagonal de la matriz wronskiana de los polinomios de Laguerre. Este método puede usarse para calcular con alta precisión relativa sus valores singulares y matrices inversas, así como la solución de algunos sistemas de ecuaciones lineales. Se han incluido ejemplos numéricos que ilustran los resultados teóricos.
- Hemos diseñado algoritmos para construir la factorización bidiagonal de la matriz wronskiana de la base de los polinomios de Bernstein y la factorización bidiagonal de las matrices wronskianas de otras bases relacionadas, como la base de Bernstein de grado negativo o la base binomial negativa. También hemos demostrado que estos algoritmos pueden usarse para realizar con alta precisión relativa algunos cálculos algebraicos con estas matrices wronskianas, como el cálculo de sus inversas, sus valores propios o sus valores singulares y las soluciones de algunos sistemas lineales relacionados. Experimentos numéricos han ilustrado los resultados teóricos obtenidos.
- Hemos proporcionado algoritmos para calcular la factorización bidiagonal de la matriz wronskiana de la base geométrica y la factorización bidiagonal de la matriz wronskiana de la base de Poisson. También hemos demostrado que estos algoritmos pueden usarse para realizar con precisión algunos cálculos algebraicos con estas matrices wronskianas, como el cálculo de sus inversas, valores propios o valores singulares o las soluciones de algunos sistemas lineales relacionados. Además, hemos incluido experimentos numéricos que han ilustrados los resultados teóricos obtenidos.
- La complejidad de todos los algoritmos con los que hemos resuelto los problemas algebraicos mencionados es comparable a la de los algoritmos LAPACK tradicionales, los cuales, como hemos ilustrado, no ofrecen tal precisión.

Trabajo futuro

Las matrices de Gram aparecen en aplicaciones tan diversas como en el método de elementos finitos, en el ajuste del modelo de la estructura de covarianza, en el aprendizaje automático (*machine learning*) (véase [60],[6], [65]) y en muchos de los problemas fundamentales de interpolación y aproximación que dan lugar a interesantes cálculos de álgebra lineal relacionados. Desafortunadamente, las matrices de Gram suelen estar mal condicionadas y, por lo tanto, los cálculos mencionados pierden precisión a medida que aumenta la dimensión del problema.

Recordemos que dado un espacio de Hilbert U con un producto interno $\langle \cdot, \cdot \rangle$ y un subespacio V de dimensión $(n+1)$ generado por una base (f_0, \dots, f_n) , el cálculo de la mejor aproximación en V , con respecto a la norma definida en U de un $u \in U$ dado, es $v = \sum_{i=0}^n c_i f_i$, donde $c = (c_0, \dots, c_n)^T$ es la solución del sistema lineal $Mc = b$ y $M = (M_{i,j})_{0 \leq i, j \leq n}$ es la matriz de Gram tal que

$$M_{i,j} := \langle f_i, f_j \rangle$$

y $b = (b_i)_{0 \leq i \leq n}$ con $b_i := \langle f_i, u \rangle$.

Las matrices de Hilbert $H_n = (1/(i+j-1))_{0 \leq i, j \leq n}$ son matrices de Gram notoriamente mal condicionadas que corresponden a la base de los monomios $(1, x, \dots, x^n)$ con respecto al producto escalar: $\langle f, g \rangle = \int_0^1 f(t)g(t) dt$. En [62], se obtienen cálculos precisos con estas matrices utilizando una representación de ellas como un producto de matrices bidiagonales totalmente positivas. Hasta nuestro conocimiento, no se han analizado cálculos con alta precisión relativa (HRA) utilizando matrices de Gram de otras bases.

Gracias a la experiencia adquirida durante el desarrollo de la tesis en la obtención de la factorización de matrices totalmente positivas como producto de matrices bidiagonales totalmente positivas, nos vamos a centrar en encontrar la factorización bidiagonal de las matrices de Gram correspondientes a las bases con las que hemos trabajado en esta tesis, como las bases de Bernstein de grado positivo y negativo, la base geométrica y la base de Poisson. En este sentido, también vamos a estudiar otras bases como la base de Ball o bases que forman soluciones fundamentales de las ecuaciones diferenciales. Además, nos gustaría estudiar la aplicabilidad de las factorizaciones obtenidas en la resolución de ecuaciones diferenciales, en el método de elementos finitos y en el aprendizaje automático, entre otros.

Recientemente, las bases duales han sido intensamente estudiadas por muchos autores y se han encontrado diversas aplicaciones interesantes, especialmente en algunos problemas de aproximación relacionados con el análisis numérico y la infografía (véase [101] y sus referencias bibliográficas). Entre otras propiedades, se ha demostrado que las matrices de Gram son la matriz de cambio de base entre una base dada y su base dual. En particular, la base dual de Bernstein de grado n es la base (D_0^n, \dots, D_n^n) que satisface $\langle D_i^n, B_j^n \rangle = \delta_{i,j}$ para $i, j = 0, \dots, n$ (cf. [101, 102, 66, 67]) y consecuentemente se cumple que

$$(B_0^n, \dots, B_n^n)^T = M(D_0^n, \dots, D_n^n)^T,$$

(Lemma 1 de [69]) donde M es la matriz de Gram de la base de Bernstein de grado n (B_0^n, \dots, B_n^n) .

Siguiendo esta interesante línea de investigación, creemos que la obtención de la factorización bidiagonal de las matrices de Gram de la base de Bernstein de grado negativo o de la base de Ball podría permitir definir propiedades importantes de las correspondientes bases duales. Esto nos podría ayudar a explorar y extender las propiedades bien conocidas de la base dual de Bernstein a dichas bases duales.

Finalmente, creemos que también sería de gran interés obtener la factorización bidiagonal de las matrices de colocación, wronskianas y de Gram de las bases spline polinómicas.

Conclusions

This doctoral thesis is framed within the theory of Total Positivity, specifically in two of the fields which are related to totally positive matrices. On the one hand, it is focused on the field of Computer Aided Geometric Design. Aspects related to totally positive bases have been developed and bases that inherit shape preservation properties from a given one have been explored. On the other hand, this work has also focused on the field of Numerical Linear Algebra. Algorithms adapted to the structure of different totally positive matrices have been designed and analyzed, making it possible to solve with high relative accuracy algebraic problems associated with these matrices. Next, we will point out the main contributions of this thesis.

- Given a system of functions, a set of weights and a positive function φ , we have defined a new system of functions called weighted φ -transformed system, which includes a wide class of useful representations in Statistics and Computer-Aided Geometric Design. It has been proved that these systems inherit some geometric properties from their initial systems, such as shape preservation or optimal shape preservation. We have shown that a general class of important rational bases can be obtained as a particular example of weighted φ -transformed systems. For these bases, evaluation and subdivision algorithms have been presented. Moreover, some relevant applications have been pointed out.
- We have obtained the bidiagonal factorization of the collocation matrices of the weighted φ -transformed systems. This bidiagonal factorization has been used to obtain computational methods with high relative accuracy for solving algebraic problems with the collocation matrices of these systems such as the computation of eigenvalues, singular values and the solution of some linear systems. Numerical examples have illustrated the accuracy of the performed computations.
- We have tackled the problem of finding a rational curve to fit a given set of data points. To solve this issue, we have applied techniques of Artificial Intelligence and we have proposed a one-hidden-layer neural network based on a general class of rational bases belonging to spaces with mix algebraic, trigonometric and hyperbolic polynomials, thus being able to reach more difficult shapes and thus expanding the potential range of applications of this neural network. In order to obtain the weights and control points of the rational curve to fit the set of data points, the neural network is trained with an optimization algorithm that updates the weights and control points while decreasing a loss function. The fitting curves of the numerical experiments have shown that for certain curves the use of particular rational bases provides better results.
- We have obtained algorithms for computing the bidiagonal factorization of the Wronskian matrix of the monomial basis of polynomials and the bidiagonal factorization of the Wronskian matrix of the basis of exponential polynomials. It has been also shown that these algorithms can be used

to perform accurately some algebraic computations with these Wronskian matrices, such as the calculation of their inverses, their eigenvalues or their singular values, and the solutions of some linear systems. Numerical experiments have illustrated the results.

- We have design an accurate method to construct the bidiagonal factorization of collocation and Wronskian matrices of Jacobi polynomials. We have used the mentioned method to compute with high relative accuracy their inverses, their eigenvalues, their singular values and the solutions of some related linear systems. The particular cases of collocation and Wronskian matrices of Legendre polynomials, Gegenbauer polynomials, Chebyshev polynomials of the first and second kind, and rational Jacobi polynomials have been also considered. Numerical examples have illustrated the accuracy of the performed computations.
- We have provided a method to obtain the bidiagonal factorization of the Wronskian matrix of Bessel polynomials and the bidiagonal factorization of the Wronskian matrix of Laguerre polynomials. This method can be used to compute with high relative accuracy the singular values, inverses, as well as the solutions of some linear systems related to the Wronskian matrices of the considered bases. Numerical examples illustrating the theoretical results have been included.
- We have design algorithms for computing the bidiagonal factorization of the Wronskian matrix of Bernstein basis of polynomials and the bidiagonal factorization of the Wronskian matrices of other related bases, such as the Bernstein basis of negative degree or the negative binomial basis. We have also shown that these algorithms can be used to perform with high relative accuracy some algebraic computations with these Wronskian matrices, such as the calculation of their inverses, their eigenvalues or their singular values, and the solutions of some related linear systems. Numerical experiments have illustrated the theoretical results.
- We have provided algorithms to construct the bidiagonal factorization of the Wronskian matrix of geometric basis and the bidigiagonal factorization of the Wronskian matrix of Poisson basis. We have also shown that these algorithms can be used to perform accurately some algebraic computations with these Wronskian matrices, such as the calculation of their inverses, eigenvalues or singular values, and the solutions of some related linear systems. Moreover, we have included numerical experiments illustrating the theoretical results.
- The complexity of all the algorithms with which we have solved the mentioned algebraic problems is comparable to that of traditional LAPACK algorithms, which, as we have illustrated, deliver no such accuracy.

Future work

Gram matrices appear in applications as diverse as in the finite element method, in the model fitting of the covariance structure, in machine learning (see [5]) and in many of the fundamental problems of interpolation and approximation which lead to interesting related linear algebra computations. Unfortunately, Gram matrices are often ill-conditioned and therefore the aforementioned computations lose accuracy as the dimension of the problem increases.

Let us recall that given a Hilbert space U under a inner product $\langle \cdot, \cdot \rangle$ and an $(n + 1)$ -dimensional subspace V generated by a basis (f_0, \dots, f_n) , the computation of the best approximation in V , with respect to the norm defined in U , of a given $u \in U$ is $v = \sum_{i=0}^n c_{i-1} f_i$, where $c = (c_1 \dots, c_{n+1})^T$ is the solution of the linear system $Mc = b$ and $M = (M_{i,j})_{1 \leq i, j \leq n+1}$ is the Gram matrix such that

$$M_{i,j} := \langle f_{i-1}, f_{j-1} \rangle$$

and $b = (b_i)_{1 \leq i \leq n+1}$ with $b_i := \langle f_{i-1}, u \rangle$.

Hilbert matrices $H_n = (1/(i + j - 1))_{1 \leq i, j \leq n+1}$ are well-known notoriously ill-conditioned Gram matrices corresponding to monomial bases $(1, x, \dots, x^n)$ with respect to the inner product: $\langle f, g \rangle = \int_0^1 f(t)g(t) dt$. In [62], accurate computations with these matrices are obtained by using an elegant representation of them as a product of nonnegative bidiagonal matrices. As far as the authors' knowledge, up to now, computations with high relative accuracy (HRA) using Gram matrices of other bases have not been achieved.

Thanks to the acquired experience during the development of the thesis in obtaining the factorization of totally positive matrices as the product of totally positive bidiagonal matrices, we are going to focus on finding the bidiagonal factorization of the Gram matrices corresponding to the bases with which we have worked on this thesis, such as the Bernstein bases of positive and negative degree, the geometric basis and the Poisson basis. In this sense, we are also going to study other bases such as the Ball basis or bases that form fundamental solutions of differential equations. In addition, we would like to study the applicability of the obtained factorizations for solving differential equations, in the finite element method and in machine learning, among others.

Recently, dual bases have been deeply studied by many authors and several interesting applications have been found, especially in some approximation problems related to numerical analysis and infographics (see [101] and the references therein). Among other properties, it has been shown that Gram matrices are the change of basis matrix between a given basis and its dual basis. In particular, the constrained dual Bernstein basis of degree n is (D_0^n, \dots, D_n^n) satisfying $\langle D_i^n, B_j^n \rangle = \delta_{i,j}$ for $i, j = 0, \dots, n$ (cf. [101, 102, 66, 67]) and then

$$(B_0^n, \dots, B_n^n)^T = M(D_0^n, \dots, D_n^n)^T$$

(see Lemma 1 of [69]) where M is the Gram matrix of the Bernstein basis of degree n (B_0^n, \dots, B_n^n) .

Following this interesting line of research, we believe that obtaining the bidiagonal factorization of the Gram matrices of the negative degree Bernstein basis or the Ball basis could allow defining important properties of the corresponding dual basis. This could help us explore and extend the well-known properties of the Bernstein dual basis to such dual bases.

To conclude, we think that it would also be of great interest to obtain the bidiagonal factorization of the collocation, Wronskian and Gram matrices of the polynomial spline bases.

Impact factor of publications

- [73] E. Mainar, J.M. Peña, B. Rubio, Evaluation and subdivision algorithms for general classes of totally positive rational bases, *Computer Aided Geometric Design* **81** (2020).
The JCR journal impact factor of Computer Aided Geometric Design in 2018 is 1.517 (Q1, Mathematics, Applied).
- [74] E. Mainar, J.M. Peña, B. Rubio, Accurate bidiagonal decomposition of collocation matrices of weighted φ -transformed systems, *Numerical Linear Algebra Appl.* **e2295** (2020).
The JCR journal impact factor of Numerical Linear Algebra with Applications in 2018 is 1.281 (Q1, Mathematics).
- [39] R. Gonzalez, E. Mainar, E. Paluzo, B. Rubio, Neural-Network-Based Curve Fitting Using Totally Positive Rational Bases, *Mathematics* **8**, 2197 (2020).
The JCR journal impact factor of Mathematics in 2019 is 1.747 (Q1, Mathematics).
- [75] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with Wronskian matrices, *Calcolo* **58**, 1 (2021).
The JCR journal impact factor of Calcolo in 2019 is 1.603 (Q1, Mathematics, Applied).
- [76] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with collocations and Wronskian matrices of Jacoby polynomials, *Journal of Scientific Computing* **87**, 77 (2021).
The JCR journal impact factor of Scientific Computing in 2019 is 2.228 (Q1, Mathematics, Applied).

Co-authorship justification

The author of this thesis has led the line of work in all the publications that compose this thesis. This author's contribution is mainly embodied in the following tasks:

- Study of the state of the art.
- Investigation.
- Conceptualization.
- Methodology.

- Software.
- Numerical experiments.
- Analysis and discussion of the experimental results.
- Writing of the manuscripts.

Bibliography

- [1] M. Abadi, et al, Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467 (2016).
- [2] T. Ando, Totally positive matrices, *Linear Algebra Appl.* **90**, 165–219 (1987).
- [3] G. Aumann, Corner cutting curves and a new characterization of Bézier and B-splines curves, *Computer Aided Geometric Design* **14** , 449–474 (1997).
- [4] U. Bashir, J. Ali, Monotone Rational Trigonometric Interpolation, *International Journal of Mathematical and Computational Sciences* **8**, 295–297 (2014).
- [5] R. Beals, R. Wong, Special functions and orthogonal polynomials, *Cambridge University Press*, Cambridge (2016).
- [6] P. Bentler, M. Jamshidian, Gram matrices arise in covariance structure model, *Applied Psychological Measurement* **18**, 79-94 (1993).
- [7] C. Bishop and C. Roach, Fast curve fitting using neural networks, *Review of Scientific Instruments* **63**, 4450-4456 (1992).
- [8] A. Björck, Numerical Methods for Least Squares Problems, *Siam* **51** (1996).
- [9] J.M. Carnicer, J.M. Peña, Shape preserving representations and optimality of the Bernstein basis, *Adv. Comput. Math.* **1**, 173–196 (1993).
- [10] J.M. Carnicer, J.M. Peña, Totally positive bases for shape preserving curve design and optimality of B-splines, *Comput. Aided Geom. Design* **11**, 633–654 (1994).
- [11] J.M. Carnicer, E. Mainar, J.M. Peña, Representing circles with five control points, *Computer Aided Geometric Design* **20**, 501– 511 (2003).
- [12] J.M. Carnicer, E. Mainar, J.M. Peña, A totally positive basis for circle approximations, *Revista de la Real Academia de Ciencias Exactas, Físicas y Naturales. Serie A. Matemáticas* **113** (2019).
- [13] P. de Castel'jau, Courbes et surfaces a poles, Thecnical report, A Citroën, Paris (1963).
- [14] J. Delgado, J.M Peña, A Comparison of Different Progressive Iteration Approximation Methods, *Mathematical Methods for Curves and Surfaces*, 136–152 (2010).
- [15] J. Delgado, J.M. Peña, Accurate computations with collocation matrices of rational bases, *Appl. Math. Comput.* **219**, 4354–4364 (2013).

- [16] J. Delgado, J.M. Peña, Fast and accurate algorithms for Jacobi-Stirling matrices, *Appl. Math. Comput.* **236**, 253–259 (2014).
- [17] J. Delgado, J.M. Peña, Accurate computations with collocation matrices of q-Bernstein polynomials, *SIAM J. Matrix Anal. Appl.* **36**, 880–893 (2015).
- [18] J. Delgado, G. Peña, J.M. Peña, Accurate and fast computations with positive extended Schoenmakers-Coffey matrices, *Numerical Linear Algebra with Applications* **23**, 1023–1031 (2016).
- [19] J. Delgado, J.M. Peña, Accurate computations with Lupaş matrices, *Appl. Math. Comput.* **303**, 171–177 (2017).
- [20] J. Delgado, H. Orera, J.M. Peña, Accurate Algorithms with Laguerre Matrices, *Numer Linear Algebra App.* **e2217** (2018).
- [21] J. Delgado, H. Orera, J.M. Peña, Accurate Algorithms for Bessel Matrices, *J. Sci. Comput.* **80**, 1264–1278 (2019).
- [22] J. Demmel, S. Eisenstat, I. Slapnicar, K. Veselik, Z. Dramc, Computing the singular value decomposition with high relative accuracy, *Linear Algebra Appl.* **299**, 21–80 (1999).
- [23] J. Demmel, P. Koev, The accurate and efficient solution of a totally positive generalized Vandermonde linear system, *SIAM J. Matrix Anal. Appl.* **27**, 42–52 (2005).
- [24] J. Demmel, I. Dumitriu, O. Holtz, P. Koev, Accurate and efficient expression evaluation and linear algebra, *Acta Numer.* **17**, 87–145 (2008).
- [25] L. Diaz, B. Rubio, J.A. Albajez, J.A. Yagüe, E. Mainar, M. Torralba, Trajectory Definition with High Relative Accuracy (HRA) by Parametric Representation of Curves in Nano-Positioning Systems, *Micromachines* **10**, 597 (2019).
- [26] C. Disibuyuk, R. Goldman, A Unifying Functional Structure for Polar Forms and for Functional Bernstein-Bezier Curves, *J. Approx. Theory* **192**, 234–249 (2015).
- [27] S.M. Fallat, C.R. Johnson, Totally Nonnegative Matrices, Princeton University Press, Princeton, NJ, *Princeton Series in Applied Mathematics* (2011).
- [28] G. Farin, Curves and Surfaces for Computer Aided Geometric Design (Fifth Ed.), Academic Press, Boston (2002).
- [29] G. Farin, Rational Curves and Surfaces, *Mathematical Methods in Computer Aided Geometric Design*, Academic Press Professional, Inc. 215–238 (1989).
- [30] R.T. Farouki, T.N.T. Goodman, On the optimal stability of the Bernstein basis, *Mathematics of Computation* **65**, 1553–1566 (1996).
- [31] R.T. Farouki, The Bernstein polynomial basis: A centennial retrospective, *Computer Aided Geometric Design* **29**, 379–419 (2012).
- [32] H. N. Fitter, A.B. Pandey, D.D. Patel, J.M. Mistry, A Review on Approaches for Handling Bezier Curves in CAD for Manufacturing, *Procedia Engineering* **97**, 1155–1166 (2014).

- [33] F.P. Gantmacher, M.G. Krein, *Oscillation Matrices and Kernels and Small Vibrations of Mechanical Systems* (revised ed.), *AMS Chelsea, Providence, RI* (2002).
- [34] M. Gasca, J.M. Peña, Total positivity and Neville elimination, *Linear Algebra Appl.* **165**, 25–44 (1992).
- [35] M. Gasca, J.M. Peña, Total Positivity, *QR* Factorization, and Neville Elimination, *SIAM Journal on Matrix Analysis and Application* **14**, 1132–1140 (1993).
- [36] M. Gasca, J.M. Peña, A matricial description of Neville elimination with applications to total positivity, *Linear Algebra Appl.* **202**, 33–53 (1994).
- [37] M. Gasca, J.M. Peña, On factorizations of totally positive matrices, in: M. Gasca, C.A. Micchelli (Eds.), *Total Positivity and Its Applications*, Kluwer. Academic Publishers, Dordrecht, The Netherlands, 109–130 (1996).
- [38] M. Gasca, Introduction to Bézier and spline curves, *Shape preserving representations of curves and surfaces*, Nova Science Publ., New York, 1–14, (1999).
- [39] R. Gonzalez, E.Mainar, E.Paluzo, B.Rubio, Neural-Network-Based Curve Fitting Using Totally Positive Rational Bases, *Mathematics* **8**, 2197 (2020).
- [40] R. Goldman R, Polya’s Urn Model and Computer Aided Geometric Design, *SIAM. J. on Algebraic and Discrete Methods* **6**, 1–28 (1985).
- [41] R. Goldman, Urn models, Approximations, and Splines, *J. Approx. Theory* **54**, 1–66 (1988).
- [42] R. Goldman, The rational Bernstein bases and the multirational blossoms, *Comput. Aided Geom. Des.* **16**, 701–738 (1999).
- [43] D. Gonsor, M. Neamtu M, Non-polynomial polar forms, in: *Curves and surfaces in geometric design*, Chamonix Mont-Blanc, A.K Peters, M.A Wellesley, 193–200 (1994).
- [44] T.N.T. Goodman, . Shape preserving representations, in: Lyche, T. and Skumaker, L.L.,eds., *Mathematical Methods in CAGD*, Academic Press, Boston, 337–357 (1989).
- [45] T.N.T. Goodman, C.A. Michheli, Corner cutting algorithms for the Bézier representation of free from curves, *Linear Algebra Appl.* **99**, 225–252 (1988).
- [46] G.H Golub, C.F Van Loan, *Matrix computations* (3rd edition), The John Hopkins University Press (1996).
- [47] E. Grosswald, *Bessel Polynomials*, Springer (1978).
- [48] G. Groza, N. Pop, Approximate solution of multipoint boundary value problems for linear differential equations by polynomial functions, *Journal of Difference Equations and Applications* **14**, 1289–1309 (2008).
- [49] L.W. Han, Y. Chu, Z.-Y. Qiu, Generalized Bézier curves and surfaces based on Lupaş q-analogue of Bernstein operator, *J. Comput. Appl. Math.* **261**, 352–363 (2014).
- [50] L.W, Han, LW, Y.S, Wua, Y, Chua, Weighted Lupaş q-Bézier curves, *Comput. Appl. Math.* **308**, 318–329 (2016).

- [51] R.E. Hartwig, Applications of the Wronskian and Gram matrices of $t^i e^{\lambda kt}$, *Linear Algebra and its Applications* **43**, 229–241 (1982).
- [52] M. Hoffmann, L. Várady, Free-form Surfaces for scattered data by neural networks, *Journal for Geometry and Graphics* **2**, 1–6 (1998).
- [53] N.J. Highman, Accuracy and Stability of Numerical Algorithms, second ed. *SIAM*, Philadelphia, PA.
- [54] L.W. Han, Y. S. Wua, Y. Chua, Weighted Lupaş q-Bézier curves, *J. Comput. Appl. Math.* **308**, 318–329 (2016).
- [55] A. Iglesias, A. Gálvez, A. Avila, Discrete Bézier Curve Fitting with Artificial Immune Systems, *Intelligent Computer Graphics*, 59–75 (2013).
- [56] A. Iglesias, A. Gálvez, M. Collantes, Global-support rational curve method for data approximation with bat algorithm, *IFIP International Conference on Artificial Intelligence Applications and Innovations*, 191–205 (2015).
- [57] C. Itu, O. A. Öchsner, S. Vlase, M. Marin, Improved rigidity of composite circular plates through radial ribs, *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications* **233** (2018).
- [58] S. Karlin, Total Positivity, Stanford University Press, Stanford, California **1** (1968).
- [59] D.P. Kingma, B. J. Adam: A Method for Stochastic Optimization, arXiv (2014).
- [60] R.C. Kirby, Fast simplicial finite element algorithms using Bernstein polynomials, *Numer. Math.* **117**, 631–652 (2011).
- [61] P. Koev, Accurate eigenvalues and SVDs of totally nonnegative matrices (2005), *SIAM J. Matrix Anal. Appl.* **27**, 1–23.
- [62] P. Koev, Accurate computations with totally nonnegative matrices, *SIAM J. Matrix Anal.* **29**, 731–751 (2007).
- [63] P. Koev, <http://www.math.sjsu.edu/~koev/software/TNTTool.html>
- [64] T. H. Koornwinder, R. S. C. Wong, R. Koekoek, R. F. Swarttouw, Orthogonal Polynomials, in: F. W. J. Olver, D. M. Lozier, R. F. Boisver, C.W. Clark (Eds.), *NIST Handbook of Mathematical Functions*, Cambridge University Press, Cambridge, Chapter 18 (2010).
- [65] G.R.G Lanckriet, N. Crisitanini, P. Barlett, L.E. Ghaoui, M.I Jordan, Learning the kernel matrix with semidefinite programming, *Journal Machine Learning Research* **5**, 27–72 (2004).
- [66] S. Lewanowicz, P. Woźny, Multi-degree reduction of tensor product Bézier surfaces with general boundary constraints, *Appl. Math. Comput.* **217**, 4596–4611 (2011).
- [67] S. Lewanowicz, P. Woźny, Bézier representation of the constrained dual Bernstein polynomials, *Appl. Math. Comput.* **218**, 4580–4586 (2011).
- [68] C. Loucera, A. Gálvez, A. Iglesias, Simulated annealing algorithm for Bézier curve approximation, *Proc. of Cyberworlds, IEEE Computer Society*, 182–189 (2014).

- [69] L. Lu, Gram matrix of Bernstein basis: Properties and applications, *J. Comput. Appl. Math.* **280**, 37–41 (2015).
- [70] E. Mainar, J.M. Peña, Corner cutting algorithms associated with optimal shape preserving representations, *Comput. Aided Geom. Design* **16**, 883–906 (1999).
- [71] E. Mainar, J.M. Peña, Accurate computations with collocation matrices of a general class of bases, *Numerical Linear Algebra With Applications* **25** (2018).
- [72] E. Mainar, J.M. Peña, B. Rubio. Accurate least squares fitting with a general class of shape preserving bases, *Monografías Matemáticas García de Galdeano* ISBN 978-84-1340-039-6, 183–182 (2019).
- [73] E. Mainar, J.M. Peña, B. Rubio, Evaluation and subdivision algorithms for general classes of totally positive rational bases, *Computer Aided Geometric Design* **81** (2020).
- [74] E. Mainar, J.M. Peña, B. Rubio, Accurate bidiagonal decomposition of collocation matrices of weighted φ -transformed systems, *Numerical Linear Algebra Appl.* **e2295** (2020).
- [75] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with Wronskian matrices, *Calcolo* **58**, 1 (2021).
- [76] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with collocations and Wronskian matrices of Jacoby polynomials, *Journal of Scientific Computing* **87**, 77 (2021).
- [77] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with Wronskian matrices of Bessel and Laguerre polynomials. Under review.
- [78] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with Wronskian matrices of Bernstein and related bases. Under review.
- [79] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with Wronskian matrices of geometric and Poisson bases. Under review.
- [80] E. Mainar, J.M. Peña, J. Sánchez-Reyes, Shape preserving alternatives to the rational Bézier model, *Computer Aided Geometric Design* **18**, 37–60 (2001).
- [81] C. Manni, F. Pelosi, M. L. Sampoli, Generalized B-splines as a tool in isogeometric analysis, *Comput. Methods Appl. Mech. Engrg.* **200**, 867–881 (2011).
- [82] A. Marco, J.J. Martínez, A fast and accurate algorithm for solving Bernstein–Vandermonde linear systems, *Linear Algebra Appl.* **422**, 616–628 (2007).
- [83] A. Marco, J.J. Martínez, Accurate computations with Said–Ball–Vandermonde matrices, *Linear Algebra Appl.* **432** 2894–2908 (2010).
- [84] A. Marco, J.J. Martínez, Polynomial least squares fitting in the Bernstein basis, *Linear Algebra Appl.* **433**, 1254–1264 (2010).
- [85] A. Marco, J. J. Martinez, Accurate computations with totally positive Bernstein-Vandermonde matrices, *Electronic Journal of Linear Algebra* **26**, 357–380 (2013).

- [86] A. Marco, J. J. Martínez, Bidiagonal decomposition of rectangular totally positive Said-Ball-Vandermonde matrices: Error analysis, perturbation theory and applications, *Linear Algebra Appl.* **495**, 90–107 (2016).
- [87] A. Marco, J. J. Martínez, Accurate computation of the Moore–Penrose inverse of strictly totally positive matrices, *Journal of Computational and Applied Mathematics* **350**, 299–308 (2019).
- [88] J.J Martínez, J.M. Peña, Fast algorithms of Bjorck-Peeyra type for solving Cauchy Vandermonde systems, *Appl. Numer. Math.* **26**, 343–352 (1998).
- [89] J.Mehnen, K. Weinert, Discrete NURBS-Surface Approximation using an Evolutionary Strategy, *Reihe Computational Intelligence* **87** (2001).
- [90] G. Morin, R. Goldman, A subdivision scheme for Poisson curves and surfaces, *Comput. Aided Geom. Des* (2000), **17**, 813–833.
- [91] M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of Machine Learning, *The MIT Press*.
- [92] J.M. Peña, Shape preserving representations for trigonometric polynomial curves, *Comput. Aided Geom. Design* **14**, 5–11 (1997).
- [93] A. Pinkus, Totally positive matrices, Cambridge Tracts in Mathematics **181**, *Cambridge University Press*, Cambridge (2010).
- [94] S. Rao, Regularized Least Square: Tikhonov Regularization test for Hilbert matrix, MATLAB Central File Exchange.
- [95] Á. Rót, Control point based exact description of trigonometric/hyperbolic curves, surfaces and volumes, *J. Comput. Appl. Math.* **290**, 74–91 (2015).
- [96] J. Sánchez-Reyes, Harmonic rational Bézier curves, p-Bézier curves and trigonometric polynomials, *Comput. Aided Geom. Design* **15**, 909–923 (1998).
- [97] Z. Šir, B. Jüttler, On de Casteljaü-type algorithms for rational Bézier curves, *J. Comput. Appl. Math.* **288**, 244–250 (2015).
- [98] V.T. Tang, K. Tanawat, Using Rational B-Spline Neural Networks for Curve Approximation, Proceedings of the 7th WSEAS International Conference on Mathematical Methods and Computational Techniques In Electrical Engineering, 42–50 (2005).
- [99] V.V Volkov, V.I Erokhin, V.V Kakaev, A.Y Onufrei, Generalizations of Tikhonov’s regularized method of least squares to non-Euclidean vector norms, *Computational Mathematics and Mathematical Physics* **57**, 1416–1426 (2017).
- [100] Z.-Q. Wang, B.-Y. Guo, Jacobi rational approximation and spectral for differential equations of degenerate type, *Math. Comp.* **77**, 883–907 (2008).
- [101] P. Wozny, S. Lewanowicz, Multi-degree reduction of Bézier curves with constraints, using dual Bernstein basis polynomials, *Comput. Aided Geom. Design* **26**, 566–579 (2009).
- [102] P. Wozny, Construction of dual bases, *J. Comput. Appl. Math.* **245**, 75-85 (2013).