Computers and Fluids 251 (2023) 105740

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Extension of an exponential discretization scheme to multidimensional convection–diffusion problems

Víctor J. Llorente [a,b,1], Antonio Pascau [c,*]

[a] ETSIAE-UPM - School of Aeronautics, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros 3, E-28040 Madrid, Spain
[b] Center for Computational Simulation, Universidad Politécnica de Madrid, Campus de Montegancedo, Boadilla del Monte, 28660 Madrid, Spain
[c] Fluid Mechanics, I3A-EINA, University of Zaragoza, 50018 Zaragoza, Spain

ARTICLE INFO

Keywords:
Fluid mechanics
Multidimensional convection–diffusion problems
Transport equation
Exponential scheme
ENATE

ABSTRACT

ENATE (**E**nhanced **N**umerical **A**pproximation of a **T**ransport **E**quation) is a high-order exponential scheme for convection–diffusion problems, such as those that govern the transport of fluid properties in a flow field. The scheme was intended to be employed in fluid-related transport equations, although it can be used for any inhomogeneous second-order ordinary differential equation. The value of a variable ϕ at a generic point is related to those of adjacent nodes via an algebraic equation. Thus, a three-point stencil is associated to each node. The coefficients of this equation contain integrals of some fluid and flow parameters. One important property is that the scheme allows to obtain a machine-accurate solution of an inhomogeneous transport equation if these integrals can be obtained analytically. As the scheme is essentially one-dimensional, getting the machine-accurate solution of multidimensional problems is not guaranteed even in cases where ENATE integrals are analytic along each coordinate. In this regard the paper presents a simple way of getting solutions in multidimensional problems while still using the one-dimensional formulation. Moreover, if the problem is such that the solution is machine-accurate in the one-dimensional problem along coordinate lines, it will also be for the multidimensional domain. Two different methods of evaluating those terms that come out of the discretization will be explained and compared in various cases.

1. Introduction

The second-order ordinary differential equation (ODE),

$$p(x)\frac{\mathrm{d}^2\phi}{\mathrm{d}x^2} + q(x)\frac{\mathrm{d}\phi}{\mathrm{d}x} + r(x)\phi = S(x),$$

has been profusely studied within the applied mathematics, physics and engineering communities. It is an ODE that governs many physical phenomena in fluids, electromagnetism, solid mechanics and many other fields of physics and engineering related to the transport of physical variables in a spatial domain. From the perspective of numerical mathematics this transport equation contains several features that make it a challenging problem for robust and accurate discretization schemes [1].

ENATE is a high-order finite-volume scheme devised for a one-dimensional transport equation with variable coefficients and source [2]. It belongs to the family of exponential schemes, so called because the coefficients in the final algebraic equation contain exponentials of non-dimensional parameters. All these schemes are somehow based on the exact solution of the homogeneous one-dimensional transport equation. The first exponential scheme was proposed by Allen and Southwell [3] and later rediscovered by Ill'in [4] and Scharfetter and Gummel [5], all three in very different settings. Scharfetter et al.'s scheme was the starting point for the development of exponential schemes in multidimensional problems for particle motion in electronic devices and convection–diffusion problems in fluids [6,7].

The popularization of exponential schemes within the Computational Fluid Dynamics (CFD) community came in the 80s with Patankar's book [8]. An improvement over the original exponential formulation for CFD was obtained by Wong and Raithby [9] and Thiart [10] on considering a constant source across each finite volume (FV) of the discretization. These two schemes are very close to each other, the only differences being the definition of the non-dimensional parameters inside the exponentials and the extension of the interval around a generic node where the source is assumed to be constant. All schemes mentioned assume constant coefficients in the FV, although they can change from one FV to the next. The fact that the coefficients are taken to be piecewise constant over the whole domain reduces the accuracy of

* Corresponding author.
E-mail addresses: victorjavier.llorente@upm.es (V.J. Llorente), pascau@unizar.es (A. Pascau).
[1] This work was initiated while the author was at Fluid Mechanics Group and LIFTEC, CSIC-University of Zaragoza.

https://doi.org/10.1016/j.compfluid.2022.105740
Received 29 October 2021; Received in revised form 7 October 2022; Accepted 21 November 2022
Available online 26 November 2022

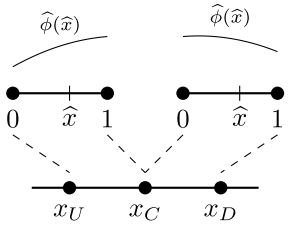

Fig. 1. Two intervals sharing C node.

the schemes. Recently, ten Thije Boonkkamp and Anthonissen derived a scheme named Finite Volume-Complete Flux (FV-CF) [11] in which the variation of the coefficients inside the control volume is taken into account. ENATE takes an alternative path to reach the same final discretization by using the non-dimensional version of the ODE. Both schemes share the same symbolic algebraic equation but the numerical treatment of its coefficients is quite different. A paper comparing both schemes has been published where the similarities and differences are highlighted [12]. A very close discretization to both FV-CF and ENATE but with constant diffusion coefficient has also been proposed by Angermann and Wang [13].

The schemes described so far were originally devised for one-dimensional differential equations. The algebraic discretization that originates from a one-dimensional differential equation can be easily computed with the Thomas Algorithm for tridiagonal matrices (TDMA) or its extension to pentadiagonal matrices (PDMA) for larger stencils. In two dimensions, the resulting matrix has non-zero diagonals that are not adjacent to the main diagonal as in one dimension. This precludes the use of TDMA or PDMA. In order to use these efficient solvers in two-dimensional problems a line-by-line calculation of the whole two-dimensional domain can be realized. When calculating the stencil of one coordinate the contribution of the other is moved to the source term. The domain is swept horizontally and vertically visiting every coordinate line where a tridiagonal/pentadiagonal system is solved for. Many CFD codes have this option where it is called line-by-line Gauss–Seidel or line-by-line Jacobi, depending on whether the newly calculated values of the previous line or those of the previous sweep are lumped in the source. As will be explained later in the paper the approach followed by ENATE for two-dimensional equations is to build two quasi-one-dimensional equations and to discretize them separately. Later, the two algebraic equations are added in order to make up the solved equation.

The paper is structured as follows. It starts with a brief description of ENATE whose main features are highlighted. Then, the extension of ENATE to multidimensions is detailed. Two ways of converting a 2D equation into two quasi-1D equations are explained. Although for unsteady one-dimensional equations the approach taken by ENATE is equivalent to that in two spatial dimensions, its particularities are commented on in a new section. Finally, some numerical tests are presented where the two 2D approaches are compared, showing the high accuracy of ENATE in multidimensional problems, either spatial or spatio-temporal. The paper finalizes with some hints on how the approach could be improved.

2. Brief description of ENATE

ENATE (**E**nhanced **N**umerical **A**pproximation of a **T**ransport **E**quation) is a high-order numerical scheme that under certain circumstances can provide machine-accurate solution of the steady one-dimensional transport equation written as a local boundary value problem (BVP)

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}x}\left(\rho u\phi - \Gamma\dfrac{\mathrm{d}\phi}{\mathrm{d}x}\right) = S_x, & x_{ub} < x < x_{db}, \\ \phi(x_{ub}) = \phi_{ub}, \quad \phi(x_{db}) = \phi_{db}, \end{cases}$$

for arbitrary diffusive, $\Gamma = \Gamma(x)$ and convective, $\rho u = \rho(x)u(x)$, coefficients and source, $S_x = S_x(x)$, that can also depend on the solution. Details on the derivation of ENATE are given in [2,14]. The whole domain is split in as many intervals as required. Two generic intervals, one between a *C*entral node and a *D*ownwind node, $x_D > x_C$, and the other between an *U*pwind node and a *C*entral node, $x_U < x_C$, are mapped onto a unity interval. To derive the algebraic discretization of the ODE the x coordinate of both intervals is transformed to $\hat{x} = (x - x_{ub})/(x_{db} - x_{ub})$, where ub and db stand for upwind boundary and downwind boundary respectively, different for each interval. For instance, $ub \equiv U$ for the interval UC, but $ub \equiv C$ for the interval CD, see Fig. 1. In order to obtain a non-dimensional equation all flow parameters are also normalized by their values at the upwind boundary of the interval. After some algebra a formula can be obtained in either interval as sum of the homogeneous and particular solutions. Subsequently, the diffusive fluxes coming from both intervals are matched at C, and an exact algebraic relation between the ϕ-values at C, U and D is obtained. We present the final expression,

$$\begin{aligned} &\left[(\rho u)_U \widetilde{k}_{UC} + (\rho u)_C \widetilde{k}_{CD} \exp \overline{P}_{x\,CD}\right] \phi_C = \\ &\left[(\rho u)_U \widetilde{k}_{UC} \exp \overline{P}_{x\,UC}\right] \phi_U + \left[(\rho u)_C \widetilde{k}_{CD}\right] \phi_D + \\ &\Delta x_{UC} \int_0^1 S_x \frac{IGE_{0\hat{x}}}{IGE_{01}} \mathrm{d}\hat{x}\bigg|_{UC} + \Delta x_{CD} \int_0^1 S_x \left(1 - \frac{IGE_{0\hat{x}}}{IGE_{01}}\right) \mathrm{d}\hat{x}\bigg|_{CD}, \end{aligned} \tag{1}$$

the interested reader is referred to [2] for details. The factors in the source integrals and in the nodal coefficients are

$$IGE_{0\hat{x}} = \int_0^{\hat{x}} \frac{1}{\widehat{\Gamma}} \left[\exp \int_{\hat{x}'}^1 P_x \,\mathrm{d}\hat{x}''\right] \mathrm{d}\hat{x}', \quad P_x = \frac{\rho u\, \Delta x}{\Gamma}, \quad \widehat{\Gamma} = \frac{\Gamma}{\Gamma_{ub}},$$

$$\widetilde{k} = \frac{1}{P_{x\,ub} IGE_{01}}, \quad \overline{P}_x = \int_0^1 P_x \,\mathrm{d}\hat{x},$$

where $P_{x\,ub}$ is the Péclet number based on values of ρu and Γ at the upwind boundary of the corresponding interval. Note that the Péclet number can vary across an interval, $\overline{P}_x$ being its average. All integrals are calculated in the unit interval. The formulation of the source contribution is different from that presented in [2] and it comes from applying integration by parts to the source term derived in that paper. As the treatment of the source is of importance in our approach to multidimensional problems, it is interesting to mention the asymptotic behaviour of the source terms as the two extreme cases, $P_x = 0$ and $P_x \to \pm\infty$, are approached. When Péclet number tends to zero and Γ is constant, the source integral is,

$$\int_0^1 S_x \frac{IGE_{0\hat{x}}}{IGE_{01}} \mathrm{d}\hat{x}\bigg|_{UC} = \int_0^1 S_x\, \hat{x}\, \mathrm{d}\hat{x}\bigg|_{UC},$$

as $IGE_{0\hat{x}}/IGE_{01}$ tends to $\hat{x}$. If, at the same time, the source is constant in the interval UC,

$$\int_0^1 S_x\, \hat{x}\, \mathrm{d}\hat{x}\bigg|_{UC} = \frac{1}{2} S_{x\,UC}.$$

The same factor comes out of the CD interval, so the total source contribution is then one half of the sum of the contributions from each interval. When Péclet goes to $+\infty$ the ratio of IGEs is one and there is only contribution from the upwind interval UC. If the source is constant the contribution is

$$\int_0^1 S_x \frac{IGE_{0\hat{x}}}{IGE_{01}} \mathrm{d}\hat{x}\bigg|_{UC} = \int_0^1 S_x \,\mathrm{d}\hat{x}\bigg|_{UC} = S_{x\,UC}.$$

If $P_x \to -\infty$ the source contribution is $S_{x\,CD}$ as CD is now the upwind interval. The task of the factor $IGE_{0\hat{x}}/IGE_{01}$ is always to weigh the source contribution of either interval, this ratio is drawn in Fig. 2. As Péclet increases, the weight of the upwind source grows at the expense of the downwind source.

In a general case of variable source and coefficients the source integral can be computed as a sum of functions of the average Péclet

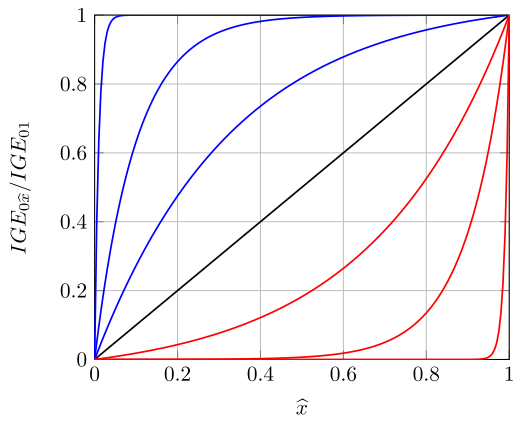

Fig. 2. IGE ratio over a unit interval for a constant Péclet regime. Black line, $P_x = 0$. Blue lines, $P_x = \{3, 10, 100\}$. Red lines, $P_x = \{-3, -10, -100\}$.

$\overline{P}_x$.

$$\int_0^1 S_x \left(1 - \frac{IGE_{0\hat{x}}}{IGE_{01}}\right) \mathrm{d}\hat{x} \approx \sum_{k=0}^{m} a_k \frac{F_k(\overline{P}_x)}{IGE_{01}},$$

where $F_0(\overline{P}_x) = (\exp \overline{P}_x - 1)/\overline{P}_x$ is the inverse of the Bernoulli function and $F_k(\overline{P}_x) = (kF_{k-1}(\overline{P}_x) - 1)/\overline{P}_x$, $k = 1, 2, \ldots, m$, m being the order of the Hermite polynomial that approximates the source. The coefficients a_k contain information of $\int S_x$, S_x, $\overline{P}_x$ and their derivatives at the interval edges [15].

It is also worth mentioning the values of the coefficients in the limit cases. The coefficients are given by

$$\begin{aligned} &A_C^x = (\rho u)_U \widetilde{k}_{UC} + (\rho u)_C \widetilde{k}_{CD} \exp \overline{P}_{x\,CD}, \\ &A_U = (\rho u)_U \widetilde{k}_{UC} \exp \overline{P}_{x\,UC}, \quad A_D = (\rho u)_C \widetilde{k}_{CD}. \end{aligned} \tag{2}$$

When $P_x = 0$ the coefficients satisfy

$$A_C^x = A_U + A_D,$$

and the actual values depend on the variation of $1/\Gamma$ across every interval. On the other hand, when $P_x \to +\infty$,

$$A_C^x = (\rho u)_C, \quad A_U = (\rho u)_U, \quad A_D = 0,$$

and when $P_x \to -\infty$,

$$A_C^x = -(\rho u)_C, \quad A_U = 0, \quad A_D = -(\rho u)_D.$$

3. Steady two-dimensional problems

In order to use ENATE in 2D problems two approaches were considered. The first approach was already developed in [12] and here we will give a brief outline. The second approach is a new contribution to the ENATE scheme.

3.1. Fluxes as pseudo-sources (FaP)

The nonhomogeneous steady-state transport equation in two spatial dimensions can be written as a local BVP,

$$\begin{cases} \dfrac{\partial}{\partial x}\left(\rho u\phi - \Gamma \dfrac{\partial \phi}{\partial x}\right) + \dfrac{\partial}{\partial y}\left(\rho v\phi - \Gamma \dfrac{\partial \phi}{\partial y}\right) = S, & (x, y) \in (x_{lb}, x_{rb}) \\ & \times (y_{bb}, y_{tb}), \\ \phi(x_{lb}, y) = \phi_{lb}(y), \quad \phi(x_{rb}, y) = \phi_{rb}(y), & y_{bb} \le y \le y_{tb}, \\ \phi(x, y_{bb}) = \phi_{bb}(x), \quad \phi(x, y_{tb}) = \phi_{tb}(x), & x_{lb} \le x \le x_{rb}, \end{cases}$$

where lb, rb, bb and tb stand for left boundary, right boundary, bottom boundary and top boundary respectively. The fluxes in each direction are defined by

$$\mathbf{f} = (f_x, f_y)^T := \left(\rho u\phi - \Gamma \frac{\partial \phi}{\partial x}, \rho v\phi - \Gamma \frac{\partial \phi}{\partial y}\right)^T.$$

In 2D, the notation employed for the upwind and downwind nodes is $(x_U, x_D)^T \equiv (x_W, x_E)^T$ for a line $y = \text{const}$, and $(x_U, x_D)^T \equiv (y_S, y_N)^T$ for a line $x = \text{const}$, see Fig. 3. Nodal points are the intersection of the two coordinate lines.

An approach to solve this equation is to split the 2D equation in two quasi-one-dimensional problems, where the derivative of the flux in one direction acts as a pseudo-source in the equation along the other,

$$\begin{aligned} &\frac{\partial}{\partial x}\left(\rho u\phi - \Gamma \frac{\partial \phi}{\partial x}\right) = S - \frac{\partial f_y}{\partial y} =: S_x, \\ &\frac{\partial}{\partial y}\left(\rho v\phi - \Gamma \frac{\partial \phi}{\partial y}\right) = S - \frac{\partial f_x}{\partial x} =: S_y. \end{aligned} \tag{3}$$

The ENATE scheme can then be applied to both equations. This approach is sometimes referred to as coordinate splitting. The problem with the pseudo-source is that it is solution-dependent and an iterative procedure is required. At the same time a quadrature must be employed for the associated integrals. ENATE starts by interpolating the integrands, S and $\partial f_{\{y,x\}}/\partial\{y, x\}$, with Hermite polynomials which are later integrated. This is the standard procedure in ENATE but any other quadrature would do. Hermite polynomials require the knowledge of the function and its derivatives at both interval edges. The integrals of the polynomial are a function of the polynomial coefficients and hence, of the derivatives at the nodes. For a third-order Hermite polynomial only the first derivative at the nodes is needed but if a seventh-order polynomial is employed, up to third derivatives are required. Neither $\partial f_x/\partial x$, $\partial f_y/\partial y$ nor their derivatives are directly available, so prior to evaluating the source integrals a way of obtaining them is called for. In this respect ENATE uses compact derivatives [16] to estimate the derivatives at a node once the function is known. For certain integrals such as $\overline{P}_{\{x,y\}}(x, y)$ and $\int_0^1 S_{\{x,y\}} \mathrm{d}\{\hat{x}, \hat{y}\}$, Compact Integration Rules (CIR) are employed [17]. CIR derives a linear combination of any integral in the working interval and its adjacents in terms of nodal values of the integrand. In contrast to Hermite polynomials, CIR is a quadrature free of derivatives. Once the two Eq. (3) are discretized via (1), they are combined and the 2D algebraic equation solved, see details in [12]. In some cases to be commented later the two equations are solved separately.

The complete sequence of steps to calculate the pseudo-source contribution in both directions is

1. Calculate ϕ with ENATE's 2D algebraic equation using the values of $\partial f_{\{y,x\}}/\partial\{y, x\}$ of the previous iteration.
2. Calculate $\partial\phi/\partial\{y, x\}$ with compact derivatives and then $f_{\{y,x\}}$.
3. Calculate $\partial f_{\{y,x\}}/\partial\{y, x\}$ with compact derivatives. This is the function to be integrated.
4. If Hermite polynomials are employed, calculate with compact derivatives as many derivatives as required by the polynomial order of the interpolant.
5. Compute the integrals.
6. Back to first step until convergence.

As just explained above there is a number of side calculations of the pseudo-source terms on top of ENATE's discretized equation computation. In the next paragraphs an alternative approach that avoids many of these extra calculations will be put forward.

3.2. REMEDIES

Let us use coordinate splitting to obtain two quasi-one-dimensional equations in the form

$$\begin{aligned} &\frac{\partial}{\partial x}\left(\rho u\phi - \Gamma \frac{\partial \phi}{\partial x}\right) = \frac{1}{2}S + \beta, \\ &\frac{\partial}{\partial y}\left(\rho v\phi - \Gamma \frac{\partial \phi}{\partial y}\right) = \frac{1}{2}S - \beta. \end{aligned} \tag{4}$$

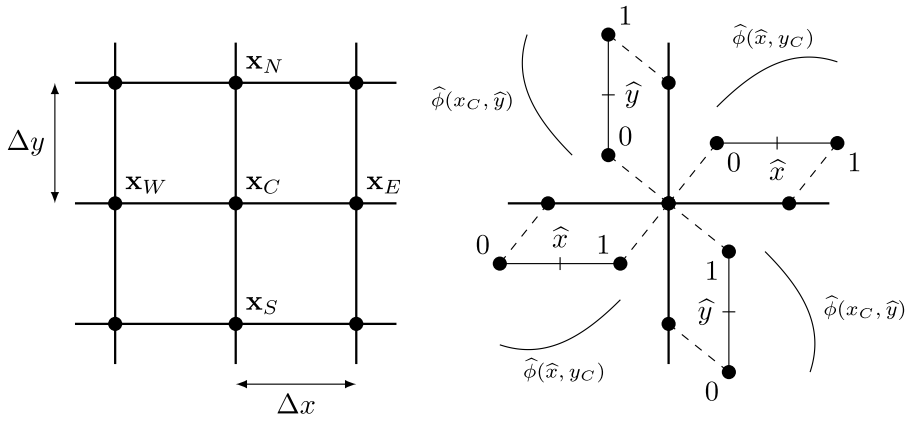

Fig. 3. Node labelling in a cartesian mesh (left). Normalized functions and intervals in two-dimensions (right) . Note that in the left figure $\mathbf{x}_C$ is a vector of coordinates, $\mathbf{x}_C = (x_C, y_C)^T$.

Instead of the pseudo-sources we now have an unknown scalar field, $\beta = \beta(x, y)$. The splitting of the source in two halves is totally arbitrary. In a specific problem it is very difficult to ascertain how much source is acting in one coordinate or another so the equal distribution is a conservative approach. This idea was put forward by Lee and Kim [18] where it was named as axial splitting. The joint solution of the two will also be solution of the original two-dimensional transport equation, at least at the continuum level. It is understood that β acts as a sort of source redistribution function between coordinates.

Consider a starting $\beta = \beta^{\text{ini}}$ for which the two equations are

$$\begin{aligned}\frac{\partial}{\partial x}\left(\rho u\phi_1 - \Gamma\frac{\partial\phi_1}{\partial x}\right) &= \frac{1}{2}S + \beta^{\text{ini}} =: S_x,\\ \frac{\partial}{\partial y}\left(\rho v\phi_2 - \Gamma\frac{\partial\phi_2}{\partial y}\right) &= \frac{1}{2}S - \beta^{\text{ini}} =: S_y.\end{aligned} \tag{5}$$

Our procedure initializes β within the domain with an inverse distance weighted-average, i.e.

$$\beta_C^{\text{ini}} = \frac{\sum_{l\in\mathcal{B}} \dfrac{\beta_l^{\text{bou}}}{d_{Cl}}}{\sum_{l\in\mathcal{B}} \dfrac{1}{d_{Cl}}}. \tag{6}$$

where d_{Cl} is the euclidean distance from the C point to the boundary l and β_l^{bou} is the boundary value of β at the point cut by the appropriate coordinate. In this paper Dirichlet boundary conditions are applied and Eq. (4) is used to compute β along the horizontal and vertical boundary lines $\mathcal{B}$. If β^{ini} is not the correct β, the solution of both equations will be different. The differences between the current values and the correct ones are

$$\Delta\beta = \beta - \beta^{\text{ini}}, \quad \Delta\phi_1 = \phi - \phi_1, \quad \Delta\phi_2 = \phi - \phi_2.$$

As we want to arrive at the solution given in Eq. (4) via successive updates, an equation for the increments can be obtained by subtracting Eq. (5) from Eq. (4), i.e.

$$\begin{aligned}\frac{\partial}{\partial x}\left(\rho u\Delta\phi_1 - \Gamma\frac{\partial\Delta\phi_1}{\partial x}\right) &= \Delta\beta,\\ \frac{\partial}{\partial y}\left(\rho v\Delta\phi_2 - \Gamma\frac{\partial\Delta\phi_2}{\partial y}\right) &= -\Delta\beta.\end{aligned} \tag{7}$$

There are three unknowns and two equations so another equation is needed to close the problem. It can be obtained from the fact that the final solution will be common to both equations, $\phi = \phi_1 + \Delta\phi_1 = \phi_2 + \Delta\phi_2$. $\Delta\phi_1$ can then be substituted in the first equation by $\Delta\phi_2 - (\phi_1 - \phi_2)$. Then,

$$\begin{aligned}\frac{\partial}{\partial x}\left(\rho u\Delta\phi_2 - \Gamma\frac{\partial\Delta\phi_2}{\partial x}\right) &= \Delta\beta + \frac{\partial f_{x(1-2)}}{\partial x},\\ \frac{\partial}{\partial y}\left(\rho v\Delta\phi_2 - \Gamma\frac{\partial\Delta\phi_2}{\partial y}\right) &= -\Delta\beta,\end{aligned} \tag{8}$$

where

$$f_{x(1-2)} = \rho u(\phi_1 - \phi_2) - \Gamma\frac{\partial(\phi_1 - \phi_2)}{\partial x},$$

is a numerical flux due to the difference of the numerical solutions in both directions independently. Now there are two equations and two unknowns because ϕ_1 and ϕ_2 are known from Eq. (5) discretized by (1). The correct $\Delta\beta$ would satisfy both equations at the same time, that is, $\Delta\phi_{2C}$ at a node C, calculated along $y = y_C$ line with the first equation, must be the same as $\Delta\phi_{2C}$ calculated along $x = x_C$ line with the second one. The discretized equations are

$$\begin{aligned}&A_C^x\Delta\phi_{2C} = A_W\Delta\phi_{2W} + A_E\Delta\phi_{2E} +\\ &A_C^x(\phi_{1C} - \phi_{2C}) - A_W(\phi_{1W} - \phi_{2W}) - A_E(\phi_{1E} - \phi_{2E}) +\\ &\Delta x_{WC}\int_0^1 \Delta\beta\,\frac{IGE_{0\widehat{x}}}{IGE_{01}}\mathrm{d}\widehat{x}\bigg|_{WC} + \Delta x_{CE}\int_0^1 \Delta\beta\left(1 - \frac{IGE_{0\widehat{x}}}{IGE_{01}}\right)\mathrm{d}\widehat{x}\bigg|_{CE},\end{aligned}$$

and

$$\begin{aligned}&A_C^y\Delta\phi_{2C} = A_S\Delta\phi_{2S} + A_N\Delta\phi_{2N} -\\ &\Delta y_{SC}\int_0^1 \Delta\beta\,\frac{IGE_{0\widehat{y}}}{IGE_{01}}\mathrm{d}\widehat{y}\bigg|_{SC} - \Delta y_{CN}\int_0^1 \Delta\beta\left(1 - \frac{IGE_{0\widehat{y}}}{IGE_{01}}\right)\mathrm{d}\widehat{y}\bigg|_{CN}.\end{aligned}$$

If the correct discrete field $\Delta\beta = \Delta\beta(x, y)$ were known the two equations would provide the same $\Delta\phi_2$ field. As $\Delta\beta$ is unknown moving towards the right β will take a number of iterative steps. We could start for instance with $\Delta\beta = 0$ and calculate $\Delta\phi_2$ with the first equation, using the second equation to calculate $\Delta\beta$. This approach was tested and it did not work out as expected, in fact, no converged solution was obtained, a reason will be given later. Another approach to update β that turned out to be efficient and relatively quick in terms of number of iterations was based on the characteristics of the $\Delta\beta$ source terms. It starts by summing up the two equations,

$$\begin{aligned}&(A_C^x + A_C^y)\Delta\phi_{2C} = A_W\Delta\phi_{2W} + A_E\Delta\phi_{2E} + A_S\Delta\phi_{2S} + A_N\Delta\phi_{2N} +\\ &A_C^x(\phi_{1C} - \phi_{2C}) - A_W(\phi_{1W} - \phi_{2W}) - A_E(\phi_{1E} - \phi_{2E}) +\\ &\Delta x_{WC}\int_0^1 \Delta\beta\,\frac{IGE_{0\widehat{x}}}{IGE_{01}}\mathrm{d}\widehat{x}\bigg|_{WC} + \Delta x_{CE}\int_0^1 \Delta\beta\left(1 - \frac{IGE_{0\widehat{x}}}{IGE_{01}}\right)\mathrm{d}\widehat{x}\bigg|_{CE} -\\ &\Delta y_{SC}\int_0^1 \Delta\beta\,\frac{IGE_{0\widehat{y}}}{IGE_{01}}\mathrm{d}\widehat{y}\bigg|_{SC} - \Delta y_{CN}\int_0^1 \Delta\beta\left(1 - \frac{IGE_{0\widehat{y}}}{IGE_{01}}\right)\mathrm{d}\widehat{y}\bigg|_{CN}.\end{aligned}$$

It looks like a discretized 2D transport equation with the $\Delta\beta$ contribution given by the last two lines. Note that the four $\Delta\beta$ terms come in pairs with opposite signs. If the flow conditions, the interval sizes, and the $\Delta\beta$ field were the same in both coordinate directions the integrals would cancel out and the net contribution would be zero. This is, of course, a very special situation but in any other case the contribution is likely to be small due to the opposite signs. According to this argument

an approximate discretized equation was built up with the $\Delta\beta$ terms neglected altogether. The resolved equation is thus

$$(A_C^x + A_C^y)\Delta\phi_{2C} = A_W\Delta\phi_{2W} + A_E\Delta\phi_{2E} + A_S\Delta\phi_{2S} + A_N\Delta\phi_{2N} + A_C^x(\phi_{1C} - \phi_{2C}) - A_W(\phi_{1W} - \phi_{2W}) - A_E(\phi_{1E} - \phi_{2E}). \quad (9)$$

The reason why the mentioned procedure of solving sequentially the two discretized equations did not work out was that on solving the first equation a term that can be significant is neglected. It is important to stress that in the approach just explained the β terms are neglected after the two equations are summed up. In that case the net contribution of all β terms is much smaller and neglecting them produces less error. Note that the most important factor that can make the β terms quite different is the interval size in both coordinates, as the IGE ratio will always lie between 0 and 1. If the interval sizes are very disparate, for the $\Delta\beta$ contribution to be of the same order in both coordinates the algebraic equation in the x-direction must be multiplied by $\Delta y/\Delta x$, or the y-direction equation by $\Delta x/\Delta y$, before adding them.

Notwithstanding the foregoing, a term that is not zero has been neglected and as a consequence the final solution cannot be reached in just one iteration. An iterative procedure will be in place. Note that the source that drives $\Delta\phi_2$ is the difference between ϕ_1 and ϕ_2 so when the difference gets to machine zero, $\Delta\phi_2$ will also be zero and the procedure will reach the final solution. Although the starting point was the same, Lee and Kim employed some very drastic simplifications for the β calculation that reduced the order of accuracy.

The updating procedure for β can be summarized in the following steps

1. Solve Eq. (9). A line-by-line Gauss–Seidel procedure is used with horizontal and vertical sweeps.
2. Obtain $\Delta\beta$ with any of the two Eq. (8).
3. Update β and recalculate ϕ_1 and ϕ_2 by solving (5) with ENATE.
4. Go to first step till convergence.

In the results section this procedure will be assessed regarding its computing time and its accuracy.

This approach has been named "**R**apid **E**valuation of **M**ultidimensional **E**quations with **D**istinct **I**ntegrals as **E**xtra **S**ources", REMEDIES, and the whole scheme as ENATE with REMEDIES. "Distinct Integrals" refer to the fact that $+\beta$ integral is in the x coordinate and $-\beta$ in the y coordinate. It is appropriate to mention that this procedure is not exclusive of ENATE, REMEDIES can be employed by any source treatment or other alternative discretization from which the coefficients of Eq. (9) can be obtained.

If a three-dimensional problem is dealt with an additional scalar function is needed. In fact, every time a new coordinate is considered a new scalar function turns up. Yet, the procedure already described can easily be extended just by taking $\alpha+\beta$ in the first coordinate, $-\alpha$ in the second and $-\beta$ in the third one. In this case two source contributions appear: f_x based on $\phi_1-\phi_2$ and f_z based on $\phi_3-\phi_2$. When $\phi_1=\phi_2=\phi_3$ the sources will be zero and $\Delta\alpha=\Delta\beta=0$.

4. Unsteady one-dimensional problems

The unsteady one-dimensional transport equation reads

$$\begin{cases} \dfrac{\partial\rho\phi}{\partial t} + \dfrac{\partial}{\partial x}\left(\rho u\phi - \Gamma\dfrac{\partial\phi}{\partial x}\right) = S, & x_{lb} < x < x_{rb}, \quad t > 0, \\ \phi(x_{lb}, t) = \phi_{lb}(t), \quad \phi(x_{rb}, t) = \phi_{rb}(t), & t > 0, \\ \phi(x, 0) = \phi^0(x), & x_{lb} \le x \le x_{rb}. \end{cases}$$

The ideas put forward in the previous section can be carried over to this equation just by considering a new spatial dimension $\eta = ct$ where c is an arbitrary velocity which is constant and positive. The unsteady one-dimensional convection–diffusion equation is rewritten as follows

$$\frac{\partial}{\partial\eta}(\rho c\phi) + \frac{\partial}{\partial x}\left(\rho u\phi - \Gamma\frac{\partial\phi}{\partial x}\right) = S. \quad (10)$$

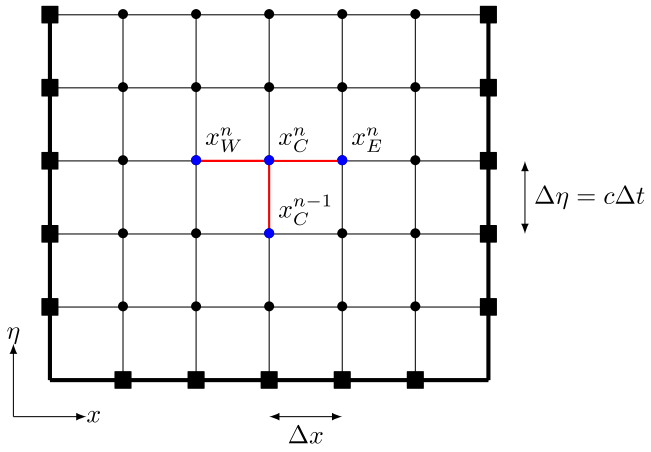

Fig. 4. Space–time discrete domain. ■ Boundary and initial values, • Inner values, Stencil of the scheme in red lines and blue dots.

We consider the temporal term akin to a diffusionless transport phenomenon with ρc a convection-like parameter. Then, the numerical solution of this equation can be computed using the same ideas as those explained above. We define the fluxes in (x,η)-domain as

$$\mathbf{f} = (f_x, f_\eta)^T := \left(\rho u\phi - \Gamma\frac{\partial\phi}{\partial x}, \rho c\phi\right)^T.$$

Considering the FaP approach, Eq. (10) is decomposed into two differential equations:

$$\begin{aligned} \frac{\partial}{\partial x}\left(\rho u\phi - \Gamma\frac{\partial\phi}{\partial x}\right) &= S - \frac{\partial f_\eta}{\partial\eta} =: S_x, \\ \frac{\partial}{\partial\eta}(\rho c\phi) &= S - \frac{\partial f_x}{\partial x} =: S_\eta. \end{aligned} \quad (11)$$

The η variable is discretized in intervals of size $c\Delta t$, $\eta_n = nc\Delta t$. The notation $\phi(x_j,\eta_n) = \phi_j^n$ will be employed. The first equation in (11) can be discretized at a time level n in the usual way. On the other hand, the discretization of the second equation in (11) is accomplished on integrating in time at a constant x-coordinate, x_C, from η_{n-1} to η_n. As this equation has no diffusion term the values of the coefficients and the source contribution are those stated earlier for $P\to+\infty$. The result is

$$\begin{aligned} &(\rho c)_C^n\phi_C^n - (\rho c)_C^{n-1}\phi_C^{n-1} = b_{\eta C}^n, \\ &b_{\eta C}^n = \int_{\eta_{n-1}}^{\eta_n} S_\eta\,\mathrm{d}\eta\bigg|_{x_C}. \end{aligned}$$

Finally, the two discrete equations can be added to arrive at an expression with a four-point stencil, see Fig. 4.

$$\begin{aligned} &-A_W^n\phi_W^n + \left[(\rho c)_C^n + A_C^n\right]\phi_C^n - A_E^n\phi_E^n = (\rho c)_C^{n-1}\phi_C^{n-1} + b_{\eta C}^n + b_{xC}^n, \\ &b_{xC}^n = \left[\Delta x_{WC}\int_0^1 S_x\frac{IGE_{0\hat{x}}}{IGE_{01}}\mathrm{d}\hat{x}\bigg|_{WC} + \Delta x_{CE}\int_0^1 S_x\left(1 - \frac{IGE_{0\hat{x}}}{IGE_{01}}\right)\mathrm{d}\hat{x}\bigg|_{CE}\right]_{\eta_n}, \end{aligned}$$

Defining the vectors

$$\begin{aligned} \boldsymbol{\rho c}^n &:= ((\rho c)_{lb}^n, \ldots, (\rho c)_C^n, \ldots, (\rho c)_{rb}^n)^T, \\ \boldsymbol{\phi}^n &:= \left(\phi_{lb}^n, \ldots, \phi_W^n, \phi_C^n, \phi_E^n, \ldots, \phi_{rb}^n\right)^T, \\ \mathbf{b}^n &:= \left(b_{\eta\,lb}^n + b_{x\,lb}^n, \ldots, b_{\eta C}^n + b_{xC}^n, \ldots, b_{\eta\,rb}^n + b_{x\,rb}^n\right)^T, \end{aligned}$$

and the matrices

$$\begin{aligned} \mathbf{A}^n &:= \mathrm{tridiag}(-A_W^n, A_C^n, -A_E^n), \\ \mathbf{C}^n &:= \mathrm{diag}(\boldsymbol{\rho c}^n) \quad . \end{aligned}$$

The system $(\mathbf{C}^n + \mathbf{A}^n)\boldsymbol{\phi}^n = \mathbf{C}^{n-1}\boldsymbol{\phi}^{n-1} + \mathbf{b}^n$, $n \geq 1$, is solved by the tridiagonal matrix algorithm (TDMA). For $n = 1$, $\boldsymbol{\phi}^0 = \phi^0(\mathbf{x})$ where $\mathbf{x} := (x_0, \ldots, x_C, \ldots, x_k)^T$. The velocity in the transformed temporal term will always be taken as 1 m/s but it will be kept in the formulae out of dimensional consistency.

In this alternative approach of considering an unknown scalar function $\beta(x, \eta)$ the algebraic derivations follow the same path as that for FaP, the reader may simply substitute the expressions for the sources.

5. Numerical tests

In this section some numerical tests will be presented in order to compare both approaches. Problems in two spatial dimensions and unsteady one-dimensional problems will be shown. All cases were run in a desktop with an Intel(R) Core(TM) i5-7400 CPU @ 3.00 GHz. All codes were written in Fortran 90.

The numerical cases were chosen with the idea of exploring both approaches in very different settings characterized by the Péclet number. The first test is a manufactured case where it will be shown that ENATE can get a machine-accurate solution with a minimum number of nodes under special but not trivial conditions. The second, third and fourth test cases encompass extreme cases of Péclet, that is, $P = 0$ in both coordinates, $|P| \to \infty$ in one coordinate and $|P| \to \infty$ in both coordinates. The additional last case is one proposed by ten Thije Boonkkamp and Anthonissen [11] who developed the FV-CF scheme, whose formulation is almost identical to ENATE. FV-CF was already compared with ENATE in 1D [12] and their performance in 2D is to be assessed. In all cases the mesh is uniform with the same interval size in both coordinates.

5.1. 2D machine-accurate solution

The equation solved in this case corresponds to a manufactured problem that represents the transport of temperature in a fluid with variable velocity. The equation is

$$\frac{\partial}{\partial x}\left(\rho c_p u T - k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\rho c_p v T - k\frac{\partial T}{\partial y}\right) = S,$$

where ρ is the density and c_p is the specific heat at constant pressure, both are constant. The actual equation solved is

$$\frac{\partial}{\partial x}\left(uT - \alpha\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(vT - \alpha\frac{\partial T}{\partial y}\right) = S^*,$$

with $\alpha = k/\rho c_p$ and $S^* = S/\rho c_p$. The velocity is variable and given by $u = y$ and $v = -x$. Two cases were run for $\alpha = \{10^{-2}, 10^{-4}\}$. The source is $S^*(x, y) = x^2 - y^2 - x$, such that the solution is $\phi = y(1 - x)$. The domain is $(0, 1) \times (0, 1)$.

With cubic Hermite and just one node in the centre of the square domain the difference between the computed solution and the exact one could be of order 10^{-16}, as long as there is a sufficient number of nodes for the approximating polynomials to be calculated exactly. In fact, this manufactured test case was especially chosen to show the capability of ENATE to obtain a machine-accurate solution for any number of grid points in a test case with analytic integrals. In the following it will be explained why the machine-accurate solution can be obtained.

The integrating factors included in the integrals of the coefficients are

$$\exp\int_{\hat{x}}^{1} P_x \,\mathrm{d}\hat{x}' = \exp\left(P_x(1-\hat{x})\right), \quad \exp\int_{\hat{y}}^{1} P_y \,\mathrm{d}\hat{y}' = \exp\left(P_y(1-\hat{y})\right),$$

as P_x depends only on y and P_y depends only on x. The IGEs ratios can be calculated exactly. For instance,

$$IGE_{0\hat{x}} = \int_0^{\hat{x}} \exp(P_x(1-\hat{x}'))\mathrm{d}\hat{x}' = \frac{1-\exp(-P_x\hat{x})}{P_x\exp(-P_x)},$$

$$IGE_{01} = \frac{1-\exp(-P_x)}{P_x\exp(-P_x)},$$

and, therefore,

$$\frac{IGE_{0\hat{x}}}{IGE_{01}} = \frac{1-\exp(-P_x\hat{x})}{1-\exp(-P_x)},$$

similarly in the other direction. Thus, $\widetilde{k}$ and $\overline{Pe}$ of Eq. (1) are known and A_W, A_E, A_S, A_N, A_C are exact in 2D.

The source and pseudo-sources are quadratic in x and y, so any polynomial of second degree or higher may approximate it exactly along one coordinate, in particular a third-degree Hermite polynomial (cubic Hermite), $S^*(\hat{x}, y_C) = \sum_{k=0}^{3} a_k\hat{x}^k$. After interpolating the integrand, all source contributions reduce to calculating integrals of the type $\int x^n \exp(-P_x x)\,\mathrm{d}x$. These integrals have analytic primitives that have been included in the code. In conclusion, whatever number of intervals is used the result of the integrals of both sources and pseudo-sources is exact. The only limitation is that a minimum number of nodes is required to be able to calculate derivatives exactly with CCS. As both coefficients and source integrals are exact, the code with ENATE scheme can provide for this example a machine-accurate solution. It must be stressed that this case was run with the general-purpose code, no previous analytic calculation was done by hand to be later coded. The computer program turned over machine-accurate results in all cases with cubic, quintic or septic Hermite polynomials. In Fig. 5 the l_2-norm vs. iteration number is shown for REMEDIES. Convergence is reached between 40–60 iterations with an average CPU time between 0.1 and 0.15 s. Septic Hermite is more sensitive to roundoff errors showing a less stable final value, that is slightly above quintic and cubic Hermite. On the other hand, Fap also reached an l_2 norm close to 10^{-16} but the convergence was very slow.

5.2. Poisson equation

The accuracy of ENATE either with FaP or with REMEDIES depends on many factors, particularly on the Péclet number based on the interval size. As part of the assessment of REMEDIES two cases with zero Péclet, i.e. a Poisson equation, were run. The Poisson equation is widely used in physics and specially in computational fluid dynamics to solve the pressure field [19,20].

The two cases presented were also studied by Zapata and Balam [21], where they used high-order compact schemes to solve the Poisson equation with different sources. Both cases solve

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = -S,$$

in a squared domain of unit side. The two sources, plotted in Fig. 6, and boundary conditions are such that the solution for the first case is

$$\phi(x, y) = \exp\left(-0.5(4\pi)^2((x-1/2)^2 + (y-1/2)^2)\right),$$

and for the second

$$\phi(x, y) = (x^3 - y^4 + x^2y^3)\,\sin 2\pi x\,\sin 2\pi y.$$

A uniform mesh in both coordinates was used with $\Delta x = \Delta y$. In these cases the accuracy of results will be measured by the l_2-norm of the vector difference between the computed results and the exact ones, $\|\phi_{comp} - \phi_{exa}\|_2 = \sum_i[(\phi_{comp_i} - \phi_{exa_i})^2\Delta x\Delta y]^{1/2}$.

In Fig. 7 the comparison between the two approaches, REMEDIES and FaP, is depicted. The norm is presented against the mesh size. The order of accuracy of both is the same but the actual values are slightly different with a gap of half an order of magnitude between them. The relative behaviour is a bit surprising because with cubic Hermite FaP is slightly better than REMEDIES but it is the other way round with quintic Hermite. The differences may only be caused by the buildup of errors due to the total number of arithmetic operations required to get to a converged solution.

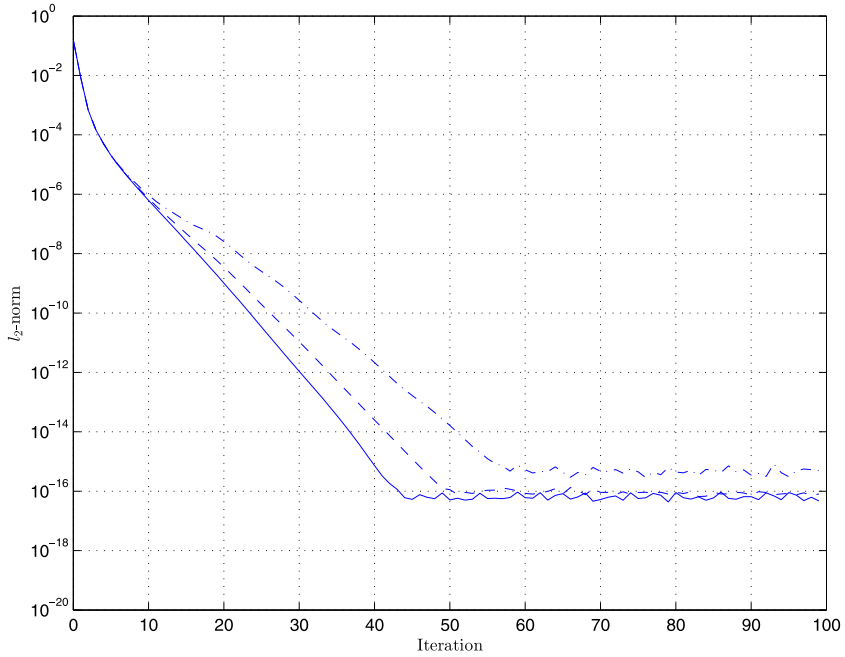

Fig. 5. l_2-norm of the error versus number of iterations for Case 5.1, 2D Machine-accurate solution, in REMEDIES for a 10 × 10 mesh and 1 sweep. Solid line with cubic Hermite. Dashed line with quintic Hermite. Dash-dotted line with septic Hermite.

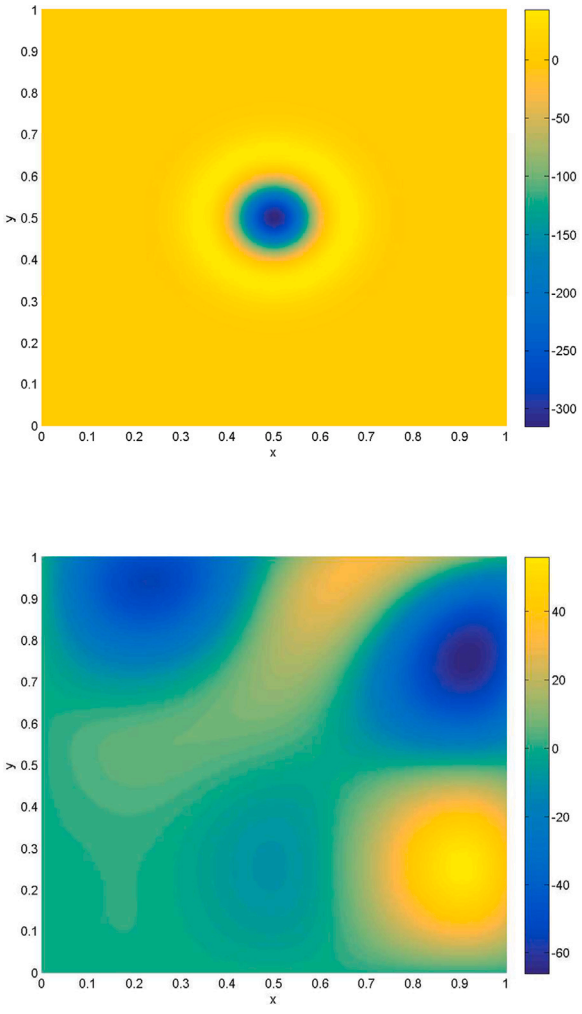

Fig. 6. Source term for the Poisson equation: first case (top), second case (bottom).

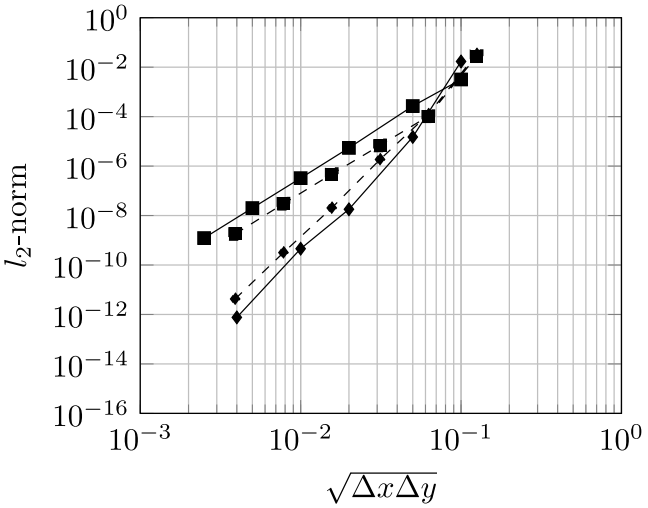

Fig. 7. l_2-norm of the error for the first Poisson problem. Solid lines are for REMEDIES and dashed lines correspond to FaP approach. ■ Cubic Hermite. ♦ Quintic Hermite.

FaP solves the algebraic equation resulting from the sum of Eq. (3) once discretized, whereas REMEDIES solves the algebraic equation for $\Delta\phi_2$ after solving for ϕ_1 and ϕ_2 along coordinate lines. CPU times will be presented later but we anticipate that REMEDIES is usually faster than FaP in reaching the converged solution.

Fig. 8 compares ENATE with REMEDIES with results from Zapata and Balam of similar order of accuracy. These authors present tables of l_2-norms for explicit, implicit and high-order implicit schemes, the names refer to the way the second derivatives are evaluated. In each category different approximations of the source function are considered. We chose for comparison those whose formal orders of accuracy are sixth, eighth and tenth respectively, named as EF3, IF3 and HIF3 in the paper. The theoretical orders of accuracy of cubic, quintic and septic Hermite are fourth, sixth and eighth but in this case, apart from cubic, the orders were closer to the others just mentioned. Cubic

Table 1
First Poisson problem. CPU time in seconds for different mesh sizes, number of sweeps, and Hermite polynomial to get a target l_2-norm = 10^{-6}. No entry means the approach does not reach the target.

Hermite	Sweeps	Mesh					
		50 × 50		100 × 100		200 × 200	
		FaP	REM	FaP	REM	FaP	REM
Cubic	1			15.771	0.575	262.090	6.906
	10			54.847	0.696	269.909	4.111
	50			30.730	1.350	281.100	21.750
	100			61.000	1.684	476.569	10.765
Quintic	1	4.938	0.004	23.733	0.516	453.636	12.042
	10	3.023	0.008	33.895	0.517	445.161	17.941
	50	14.590	0.150	58.400	1.380	425.110	22.490
	100	14.586	0.491	134.185	8.585	369.123	27.797
Septic	1	7.538	0.004	31.383	0.705	369.807	6.018
	10	4.616	0.007	90.719	0.353	434.121	19.869
	50	22.630	0.130	89.680	1.410	787.850	22.140
	100	44.465	0.602	209.496	8.108	691.379	30.655

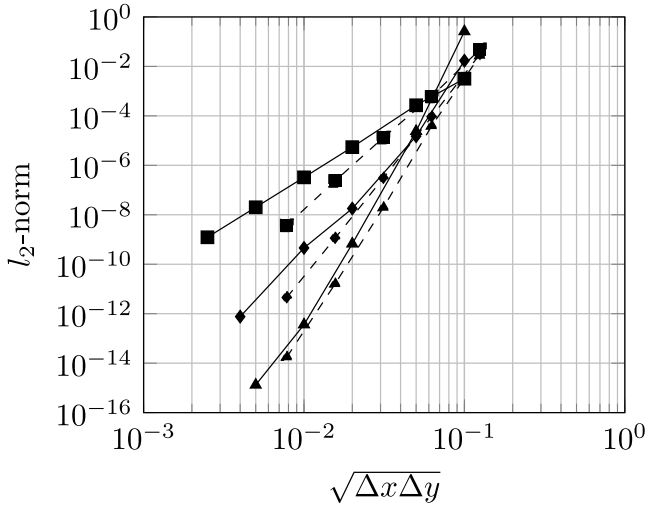

Fig. 8. l_2-norm of the error for the first Poisson problem. Solid lines are those of REMEDIES and dashed lines correspond to Zapata and Balam data of similar order of accuracy. ■ Cubic Hermite and EF3. ♦ Quintic Hermite and IF3. ▲ Septic Hermite and HIF3.

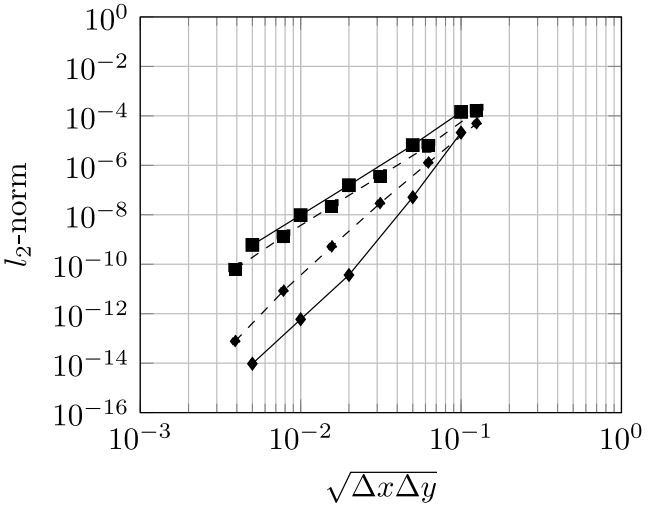

Fig. 9. l_2-norm of the error for the second Poisson problem. Solid lines are for REMEDIES and dashed lines correspond to FaP approach.

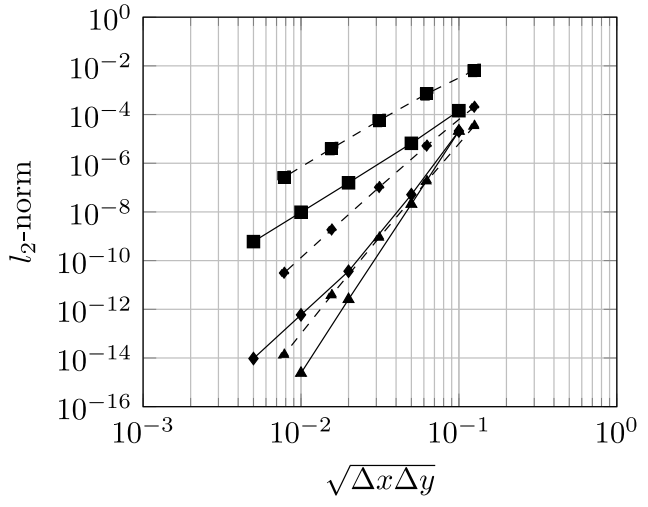

Fig. 10. l_2-norm of the error for the second Poisson problem. Solid lines are those of REMEDIES and dashed lines correspond to Zapata and Balam data of similar order of accuracy. ■ Cubic Hermite and EF3. ♦ Quintic Hermite and IF3. ▲ Septic Hermite and HIF3.

Hermite is the only Hermite polynomial that conforms to the formal order of accuracy. For a short range of mesh sizes quintic Hermite is eighth-order but in the last mesh becomes sixth-order. Septic Hermite is tenth-order over an ample range of mesh sizes and follows closely the convergence results of HIF3.

The second Poisson case is depicted in Fig. 9. Similar conclusions to those of the first Poisson case can be drawn from the comparison of REMEDIES and FaP. For cubic Hermite both approaches gave very close l_2 results. For quintic Hermite the differences were more noticeable, always in favour of the β treatment. It seems that the new terms in the interpolant introduced by quintic Hermite are more important in this case and the requirements of a larger number of iterations for FaP contribute to greater discretization errors. Yet, in the final range of interval sizes both behave as sixth-order.

Fig. 10 shows the comparison with Zapata and Balam schemes. All schemes present the formal order of accuracy except septic that is tenth order. For all interval sizes cubic Hermite is almost two orders of magnitude better than Zapata et al.'s comparable scheme. Quintic Hermite is two orders of magnitude better only for small interval sizes.

The explicit schemes of Zapata and Balam require the use of TDMA or PDMA to solve the final system of equations but due to the stencil of the implicit schemes, namely 21 points for both IF3 and HIF3 in 2D, a SOR procedure is followed. ENATE only employs TDMA, for a three-point stencil in each coordinate is employed, which also makes the coding much less burdensome.

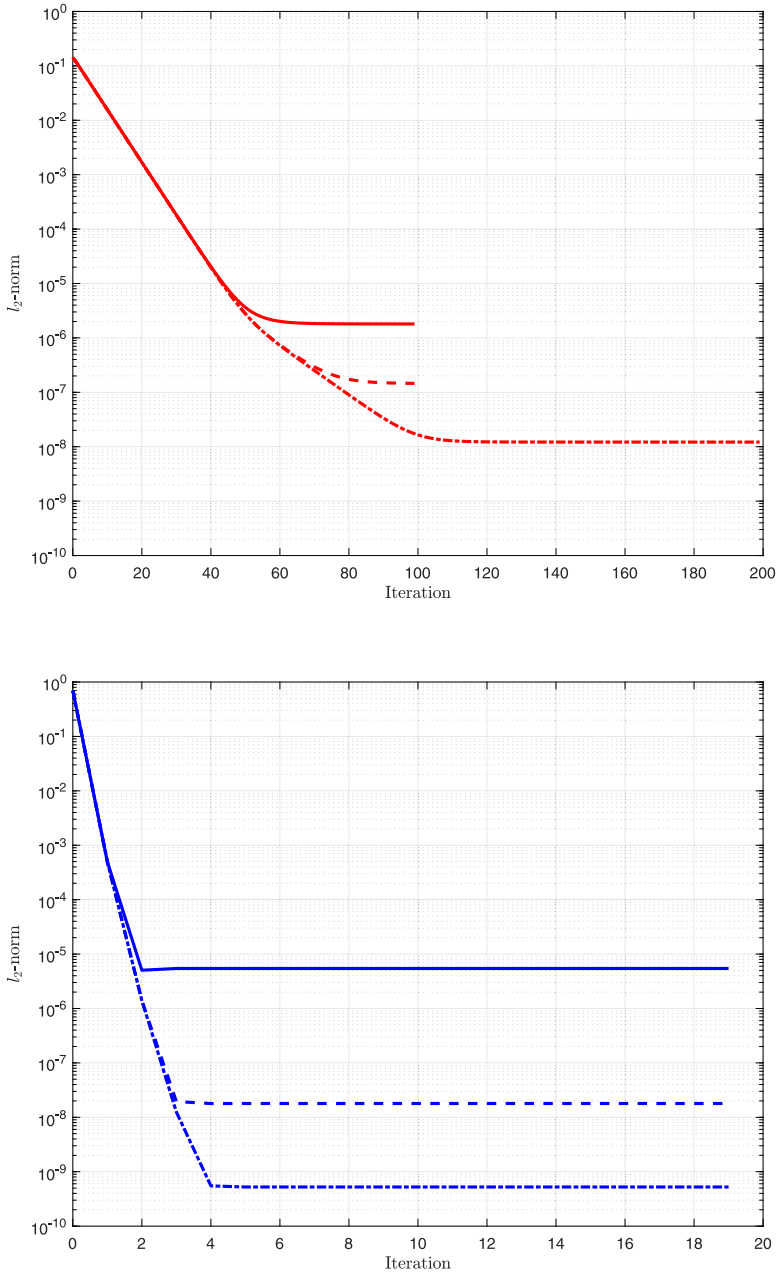

Fig. 11. First Poisson problem. l_2-norm of the error versus number of iterations in FaP (red line) and REMEDIES (blue line) for a 50 × 50 mesh and 100 sweeps. Solid line with cubic Hermite. Dashed line with quintic Hermite. Dash-dotted line with septic Hermite.

The l_2 data from Zapata and Balam are extracted directly from their paper where no information about the CPU time taken by each scheme is provided for these two cases. In the course of our research with REMEDIES it was found that the CPU time was strongly dependent on the Péclet number of the problem. The smaller this number is, the longer CPU time is needed to get the final solution due to the number of sweeps required to obtain a reasonably good estimation of $\Delta\phi_2$. A good estimation reduces the number of iterations. Remind that this equation is the only one in REMEDIES that is 2D, the equations for ϕ_1 and ϕ_2 are 1D.

In Tables 1 and 2 the CPU time taken in the two Poisson cases by FaP and REMEDIES is shown. Tables show the time taken to reach a desired l_2-norm by different Hermite polynomials as interpolants. The target norm was arbitrarily set to 10^{-6}. The code was run for a large number of iterations and the iteration at which l_2-norm was less than 10^{-6} for the first time was recorded. Times given at the tables were obtained by using the total time and the ratio between the iteration number of the target and the total number of iterations. One additional variable that is considered at the tables is the number of sweeps per iteration as explained in the theoretical description. Note that the right hand side of the discretized equations is only updated after an iteration with a prescribed number of inner sweeps is performed. Sweeps control the accuracy of the solved variable at the end of one iteration.

At earlier stages of this research it was also observed for REMEDIES that there existed a minimum number of sweeps below which the solution diverged. This happened when Péclet was very low, that is, for very fine meshes. With just one iteration in fine meshes a small number of sweeps do not provide an acceptable solution at the end of one iteration. The problem was solved by also considering the residual of the 2D discretized equation for $\Delta\phi_2$ at each point in the equation to obtain $\Delta\beta$. This modification allowed to obtain a converged solution whatever number of sweeps were set. Note that due to our procedure,

Table 2
Second Poisson problem. CPU time in seconds for different mesh sizes, number of sweeps, and Hermite polynomial to get a target l_2-norm = 10^{-6}. Null entry means the run diverged.

Hermite	Sweeps	Mesh					
		50 × 50		100 × 100		200 × 200	
		FaP	REM	FaP	REM	FaP	REM
Cubic	1	0.62	1.12	9.26	13.88	198.17	220.42
	10	2.33	0.42	13.49	6.04	104.12	98.55
	50	11.13	0.35	35.48	5.32	313.06	88.10
	100	22.03	0.38	71.34	5.26	386.06	86.86
Quintic	1	1.12	1.39	21.36	18.75	342.57	347.93
	10	3.76	0.42	25.61	6.34	237.11	128.20
	50	17.55	0.34	68.52	5.43	216.07	108.54
	100	34.78	0.35	135.85	5.17	193.38	106.05
Septic	1	1.68	1.67	30.12	28.57		
	10	5.60	0.48	40.22	8.46		137.44
	50	28.01	0.37	106.61	6.68		109.92
	100	55.60	0.38	210.06	6.43		106.00

i.e. sweeps+TDMA, an infinite number of sweeps would theoretically be required to obtain the exact solution in each iteration. With a finite number of sweeps there will always be an iteration error and, hence, a residual.

From the tables it can be seen that CPU times taken by REMEDIES are shorter than those by FaP, especially for relatively coarse grids. For the same number of inner sweeps the number of operations is similar in both approaches, but FaP requires more outer iterations than REMEDIES to obtain convergence (see Figs. 11 and 12). The number of operations required for both approaches is detailed in Appendix. In some test cases FaP is prohibitively slow partially due to the need of underrelaxation in the flux updates to control the oscillations that frequently appear in FaP. In order to dampen them an underrelaxation factor of 0.1 was always used, simply to be on the safe side. No studies were carried out to determine the optimal value of this factor for every case. It is very likely that the CPU time may be improved on optimizing the underrelaxation factor, although it will surely be case dependent.

There are some differences in CPU time between the two cases comparing FaP and REMEDIES. It seems that the source spatial distribution plays a significant role in the convergence process. For the second case, which contains a more complicated source, the differences in computing time for both approaches are smaller than in the first case. As the sources are more involved both updated outputs, β for REMEDIES and fluxes for FaP, are more spatially complex and the differences are reduced.

For this second Poisson case using REMEDIES there are small differences in the time taken by a given Hermite polynomial for 10, 50 or 100 sweeps, that is much less than that with 1 sweep. This gives the user some confidence regarding the chosen number of sweeps: between 10 and 100 they are not going to penalize much the run time. For the first Poisson case the longest run time is with 100 sweeps which shows that the number of optimal sweeps is critically dependent on the source distribution which consequently affects the solution pattern. According to this very limited study it can be concluded that an adequate number of sweeps is around 10. This discussion on the optimal number of sweeps is only relevant when both coordinates are elliptic, that is, contain a diffusion term. If one of them is parabolic/hyperbolic only one sweep is required.

5.3. Unsteady homogeneous one-dimensional convection–diffusion problem

The equation solved is an unsteady transport equation in a domain of unit length for a time $t \in (0, T]$, i.e.

$$\begin{cases} \dfrac{\partial \rho\phi}{\partial t} + \dfrac{\partial}{\partial x}\left(\rho u\phi - \Gamma \dfrac{\partial \phi}{\partial x}\right) = 0, & (x,t) \in (0,1)\times(0,T], \\ \phi(0,t) = 0, & t \in (0,T], \\ \phi(x,0) = \exp(5x)\sin(\pi x), & x \in (0,1). \end{cases}$$

with $\rho = 1$, $u = 0.1$, $T = 1$ and $\Gamma = 0.01$. The solution is $\phi(x,t) = \exp(5x - t(0,01\pi^2 + 0,25))\sin(\pi x)$.

As explained earlier, the case has been run with both approaches as a two-dimensional convection–diffusion equation with a pseudo-spatial coordinate $\eta = ct$, $c = 1$. A uniform mesh is employed in both coordinates with $\Delta\eta = \Delta x$.

In Fig. 13 norms of FaP and REMEDIES are compared against each other as well as with a traditional scheme like Crank–Nicolson. All schemes conform to their theoretical order of accuracy: Crank–Nicolson is second order, and all nth-order Hermite polynomials are of $(n+1)$th order of accuracy, at least over a decade. Cubic Hermite provides the same norms for both FaP and REMEDIES. Quintic-REMEDIES is roughly two orders of magnitude better than Quintic-FaP, both being sixth-order in part of the graph. Septic-REMEDIES is between eighth- and ninth-order.

The CPU time for REMEDIES to get a l_2-norm = 10^{-6} is well below one second even for the finest meshes, Table 3. Only a sweep is required to estimate the 2D variables which makes the difference in CPU time for both approaches more noticeable, always in favour of REMEDIES. The sweep is always forward in time and the t = cons. line can be calculated with TDMA. In the case of Poisson equation the connection of a generic point is the same with a north point which has not yet been updated as with a south point newly obtained in a south–north sweep. As a consequence, it takes more sweeps, south–north and north–south, to get a good value. This is only due to our strategy of sweeping the 2D domain line-by-line, had we used another procedure to invert the coefficient matrix the conclusions about the computer time associated to various Péclet numbers would have been different.

The convergence for every approach is presented in Fig. 14. As before, REMEDIES requires far less iterations than FaP. With high-order polynomials the convergence is less smooth for both approaches, even FaP-septic does not get a stable value. The reason is very likely to be related to the presence of a coordinate with no diffusion that makes the approaches very sensitive to the way the sources are treated, in this case via CCS. It is well known that CCS are more prone to instabilities as their degree increases.

5.4. Unsteady inhomogeneous one-dimensional convection problem

The equation solved is an unsteady inhomogeneous transport equation in a domain of unit length for a time $t \in (0, T]$, i.e.

$$\begin{cases} \dfrac{\partial \rho\phi}{\partial t} + \dfrac{\partial}{\partial x}(\rho u\phi) = -\dfrac{\phi}{\tau}, & (x,t) \in (0,1)\times(0,T], \\ \phi(0,t) = \exp(-t/\tau)\sin(-\pi u t), & t \in (0,T], \\ \phi(x,0) = \sin(\pi x), & x \in (0,1). \end{cases}$$

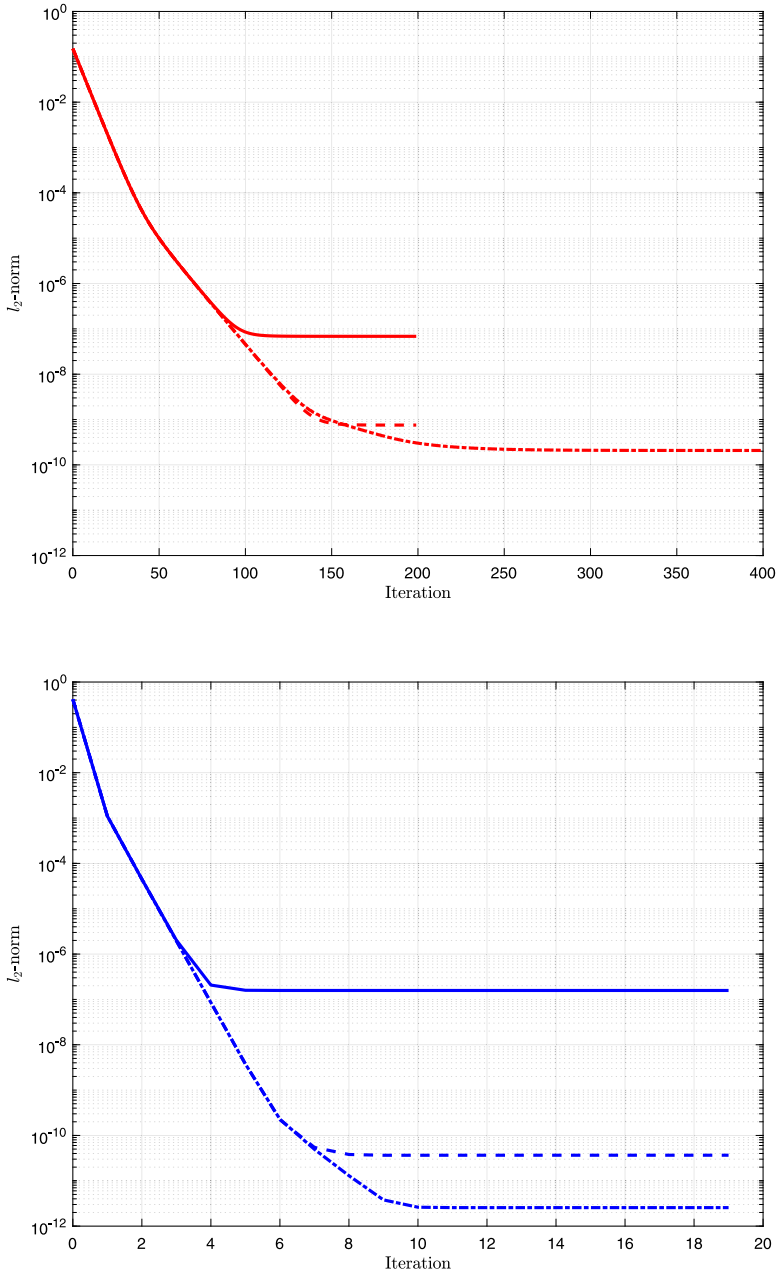

Fig. 12. Second Poisson problem. l_2-norm of the error versus number of iterations in FaP (red line) and REMEDIES (blue line) for a 50 × 50 mesh and 100 sweeps. Solid line with cubic Hermite. Dashed line with quintic Hermite. Dash-dotted line with septic Hermite.

Table 3
CPU time in seconds for different mesh sizes to get a target l_2-norm = 10^{-6} for Case 5.3, the sourceless unsteady equation. Cubic 50 × 50 does not reach the target l_2. Other null entries mean no convergence.

Hermite	Mesh					
	50 × 50		100 × 100		200 × 200	
	FaP	REM	FaP	REM	FaP	REM
Cubic			7.31	0.06		0.03
Quintic	0.43	0.03	47.28	0.08		0.22
Septic	2.06	0.03		0.09		

with $\rho = 1$, $u = 1$, $\tau = \{0.5, 0.05\}$, and $T = 1$. The exact solution is $\phi(x,t) = \exp(-t/\tau)\sin(\pi(x - ut))$. Parameter τ is a characteristic time of decay. This convection–reaction problem could be classified as a hyperbolic equation with a stiff–relaxation source term [22], similar to the Boltzmann equation but with the absence of the local equilibrium distribution [23].

To obtain accurate solutions with FaP the two coordinate lines had to be calculated separately. It can be easily checked that the sum of the two produces a discretized 2D equation identical to the standard upwind discretization in finite volumes. This gives rise to a solution pattern very diffusive associated to large amounts of numerical diffusion. Initially, the standard procedure was followed with FaP but

Table 4
CPU time in seconds for different mesh size, scheme, and Hermite polynomial to get a target l_2-norm = 10^{-6} fro Case 5.4, the stiff-source hyperbolic equation. Null entry means the run diverged.

Hermite	Scheme	Mesh					
		50 × 50		100 × 100		200 × 200	
		FaP	REM	FaP	REM	FaP	REM
Cubic	CCS	1.15	0.27	10.34	0.42	102.94	1.02
	CIR	0.86	0.23	7.20	0.39	78.03	0.93
Quintic	CCS	4.39	0.31	40.63	0.56		1.35
	CIR	2.05	0.22	19.65	0.41		1.02

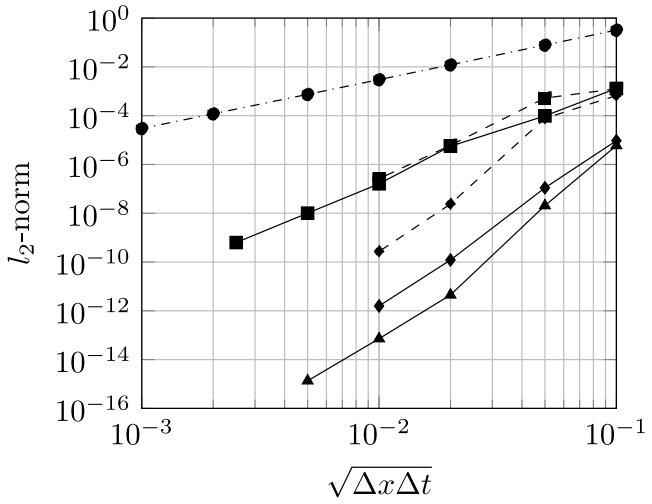

Fig. 13. l_2-norm of the error for Case 5.3, the sourceless unsteady equation. Solid lines are for REMEDIES and dashed lines are for FaP. • Crank–Nicolson ■ Cubic Hermite. ♦ Quintic Hermite. ▲ Septic Hermite.

after realizing the overly diffusive solution the procedure had to be changed and the two equations solved separately. Data from one were used in the source of the other.

The source depends linearly on the solution and has to be updated in every iteration. Remind that if Péclet goes to infinity the upwind integral of the source is the only contribution of the right-hand side. As CIR may also be employed to calculate this source integral, a thorough comparison between CIR and CCS in both approaches was carried out in this test case.

The existence of the solution-dependent source makes the equations for $\Delta\phi_2$ and $\Delta\phi_1$ contain an extra source that can be split in two, so the equations for $\Delta\phi_1$ and $\Delta\phi_2$ are

$$\begin{aligned} \frac{\partial}{\partial x}\left(\rho u \Delta\phi_1 - \Gamma \frac{\partial \Delta\phi_1}{\partial x}\right) &= -\frac{\Delta\phi_1}{2\,\tau} + \Delta\beta, \\ \frac{\partial}{\partial \eta}\left(\rho c \Delta\phi_2\right) &= -\frac{\Delta\phi_2}{2\,\tau} - \Delta\beta. \end{aligned} \tag{15}$$

The algebraic equation for $\Delta\phi_2$ differs from that presented in Eq. (9) given by

$$\begin{aligned} &(A_C^x + A_C^\eta)\Delta\phi_{2C} = A_W \Delta\phi_{2W} + A_E \Delta\phi_{2E} + A_S \Delta\phi_{2S} + A_N \Delta\phi_{2N} + \\ &A_C^x(\phi_{1C} - \phi_{2C}) - A_W(\phi_{1W} - \phi_{2W}) - A_E(\phi_{1E} - \phi_{2E}) - \\ &\frac{\Delta x_{WC}}{2}\int_0^1 \frac{\Delta\phi_1}{\tau}\,\mathrm{d}\hat{x}\bigg|_{WC} - \frac{\Delta\eta_{SC}}{2}\int_0^1 \frac{\Delta\phi_2}{\tau}\,\mathrm{d}\hat{y}\bigg|_{SC}. \end{aligned}$$

In our implementation the equation for $\Delta\phi_2$ is employed to update the solution. $\Delta\phi_1$ in the first integral of last line is considered to be equal to $\Delta\phi_2$ during the whole updating process. The reason is that in order to calculate $\Delta\beta$ we found convenient to have the same solution-dependent source in both Eq. (15).

In Fig. 15 the convergence pattern of two cases of cubic Hermite, one with no source in the RHS of $\Delta\phi_2$ equation and the other with it, is shown. Both runs start from scratch with a mesh of 500 × 500. The main difference between both approaches is in the initial increase of the norm during the first iterations. When including the extra source this increase is much less pronounced what makes the convergence quicker in the initial stages. In both runs there is a noticeable change of slope when the norm reaches the region of low values. This may be related to the relative importance of the neglected $\Delta\beta$ terms compared to the considered ones. When the neglected integrals are much less than the terms kept in the RHS associated to $\phi_1 - \phi_2$ the convergence is very quick, whereas it slows down considerably if they are of the same order. However, this change of slope is not seen in all cases tested, so further studies are required to ascertain its origin. For a large interval of the convergence curve the two curves lie on top of each other. A criterion to decide whether including or not the $\Delta\phi_2$ source can be the time taken for both approaches to reach a certain l_2-norm, for instance 10^{-6}. Well, the run with no source takes 23 s and that with source 30 s. At least in this case it is not worth to include it. It may well be that in other computational tests the inclusion of this source makes a favourable difference.

In this test problem the convergence pattern has some ups and downs for some combinations of meshes and Hermite polynomials. In Fig. 16 the l_2-norm is plotted against the number of iterations for two meshes 100 × 100 and 200 × 200 with quintic Hermite. This hilly behaviour always appears for quintic and septic Hermite polynomials and relatively fine grids. Moreover, in the downward part of the convergence hill the curve is mildly oscillatory that may suggest that the process of convergence is more unstable in these cases. For very coarse grids, 10 × 10 and 20 × 20, there is a slight rise in the norm during several iterations that can barely be noticed.

In Figs. 17 and 18 the l_2-norm is depicted with FaP and REMEDIES. Cubic Hermite behaves as a fourth order scheme. Quintic Hermite with FaP is sixth order but with REMEDIES is seventh-order for some intervals. Septic Hermite behaviour is pretty close to that of Quintic Hermite.

In Table 4 the CPU time to get down to $l_2 = 10^{-6}$ is compared for the two approaches with $u = 1$ and $\tau = 0.5$. In all mesh sizes REMEDIES is faster than FaP. For a given approach, CIR accelerates the convergence of ENATE with FaP or REMEDIES, as seen in Fig. 19 for a 50 × 50 mesh. However, it is less accurate than CCS in ENATE with REMEDIES. With another mesh size and τ parameter the performance is similar.

5.5. Wave travelling problem

The last test case is one with a bit more complicated source term.

$$\begin{cases} \dfrac{\partial \phi}{\partial t} + \dfrac{\partial}{\partial x}(u\phi) = -\dfrac{1}{\tau}\phi(1-\phi), & (x,t) \in (0,1)\times(0,0.5], \\ \phi(0,t) = \phi_0(t) = 0.8 + 0.2\sin(2\pi t), & t \in (0,0.5], \\ \phi(x,0) = \phi^0(x) = 0.8, & x \in (0,1). \end{cases}$$

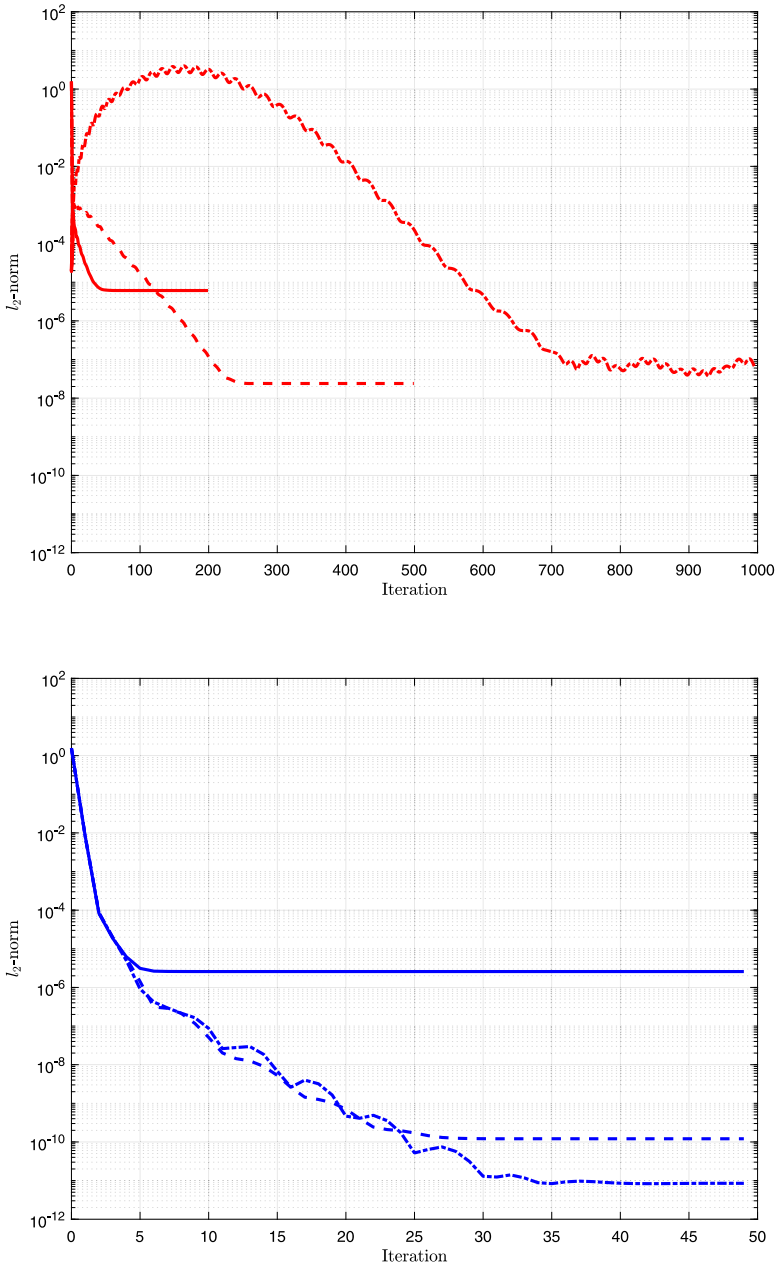

Fig. 14. l_2-norm of the error versus number of iterations for Case 5.3, the sourceless unsteady equation, in FaP (red line) and REMEDIES (blue line) for a 50 × 50 mesh. Solid line with cubic Hermite. Dashed line with quintic Hermite. Dash-dotted line with septic Hermite.

As in the previous example, τ is a characteristic time of decay. The solution is

$$\phi(x,t) = \begin{cases} \left[1 + \left(\dfrac{1}{\phi_0\,(t - x/u)} - 1\right)\exp\left(\dfrac{x}{u\tau}\right)\right]^{-1} & \text{for} \quad x < ut \\ \left[1 + \left(\dfrac{1}{\phi^0\,(x - ut)} - 1\right)\exp\left(\dfrac{t}{\tau}\right)\right]^{-1} & \text{for} \quad x \geq ut \end{cases}$$

This example was also chosen by ten Thije Boonkkamp and Anthonissen to assess the FV-CF scheme [11]. The solution of this case is only C^0 across the characteristic line $x = ut$. Both the fluxes and β are discontinuous at this line what makes this a challenging test case for our approach. Additionally, it was shown in [12] that FV-CF and ENATE were almost identical in 1D but differed considerably in 2D, and this case is a good test to assess the accuracy of our approach in comparison to FV-CF.

The exact solution decays in time except along the characteristic line that starts at $x = 0, t = 1/4$ along which ϕ is one and the source zero. This lack of decay initially transforms the sine function into a spiky function that eventually converts into a function of value one at points of the straight line $x = u(t-1/4)$ and measure zero in the x-domain. The assessment of both approaches will be performed in the x-domain for $t = 0.5$. No extra source due to the solution-dependent term has been included in the $\Delta\phi_2$ equation.

In Fig. 20 comparative results with a very coarse grid are depicted. Contrary to the Euler method, both ENATE approaches follow the spiky part of the solution but with significant over- and under-shoots, especially REMEDIES. With such a coarse grid the results are not very good and a possible solution for a high resolution could be the clustering of nodes near $x = u(t - 1/4)$.

In Fig. 21 the comparison between FV-CF and ENATE is presented. The values of the parameters employed are $\tau = 0.04$ and

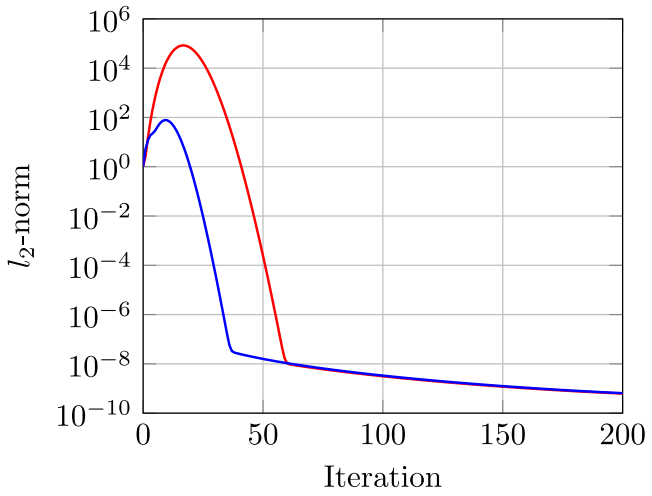

Fig. 15. l_2-norm of the error versus number of iterations for Case 5.4, the stiff-source hyperbolic equation, in REMEDIES for a 500 × 500 mesh. Red line with $\Delta\phi_2$ source. Blue line without $\Delta\phi_2$ source.

$u = 0.95$. We computed the l_1-norm of the error, $\|\phi_{comp} - \phi_{exa}\|_1 = \Delta x \sum_i |\phi_{comp_i} - \phi_{exa_i}|$, at $t = 0.5$ s. As shown, FV-CF is second-order and the two ENATE versions, FaP and REMEDIES, are fourth-order. The main difference between both approaches is the source treatment. Splitting Eq. (10) into two quasi-one-dimensional equations seems to work fine either with FaP or REMEDIES because both approaches use high-order quadratures. In the case of FV-CF, $\partial\phi/\partial t$ is only included as pseudo-source. Then the scheme produces a first-order ODE system solved by a second-order Crank–Nicolson method, see [11] for details.

The results presented with ENATE-REMEDIES have been carried out with CCS for the integrals, but FaP used CIR. Compact derivatives of fourth-order are used to evaluate the first derivative of the appropriate variables. Despite the discontinuities of the latter, fourth-order compact derivative is still able to provide a reasonably good estimation of the first derivative. In fact, fourth-order compact derivative estimation even provides usable values of second and third derivatives for Quintic and Septic Hermite. However, the use of compact derivatives of order higher than fourth caused a blowup of the calculations. When a combination of Hermite polynomials and compact derivatives of different orders is used, the order of accuracy of the results is the lower of the two, for example, Quintic Hermite (sixth-order) and fourth-order compact derivatives gives a fourth-order accuracy overall. There are huge differences in the CPU time. FaP took 50 min to reach the solution at $t = 0.5$ s with a 500×500 mesh whereas REMEDIES employed around 4 min.

6. Conclusions and future research

In this paper, multidimensional convection–diffusion problems were solved numerically by a novel high-order exponential scheme named ENATE. This FV-like scheme seeks the exact solution of the second-order one-dimensional ODE via a discrete equation with a three-point stencil solved by TDMA. The coefficients and the discrete source contain some integrals of the Péclet number, diffusion parameter and source.

In order to apply ENATE to steady-state two-dimensional transport equations, two approaches were used, one of them already developed in [12]. FaP consists of a coordinate splitting where $\partial f_x/\partial x$ and $\partial f_y/\partial y$ are pseudo-sources from the quasi-one-dimensional transport equations in the other coordinate. On the contrary, REMEDIES is a so-called axial splitting that, via an extra source β and successive corrections in the split differential equations, is able to reach the solution of the convection–diffusion problem with less number of iterations than FaP. For transient problems of one spatial dimension, we can solve the transport equation considering the temporal term as a diffusionless flux, so we might apply the same approaches as previously.

Five numerical tests were performed. The first example showed that even if we did not know the exact solution in 2D, we might obtain a machine-accurate solution with a minimum number of grid points in two-dimensional problems as long as Péclet number is constant along the coordinate lines and the source is a polynomial of third-degree or less (with cubic Hermite). In the example of the Poisson equation, FaP, REMEDIES and the schemes in [21] were compared. FaP was a bit more accurate than REMEDIES for Cubic Hermite but with Quintic it was the other way round, the difference being more noticeable. This drop in accuracy for FaP is likely due to the number of iterations that increase with Quintic. In fact, FaP with Septic was not plotted because it showed a chaotic behaviour, worsening Quintic predictions. With the implicit scheme in [21], no large differences are found except for the second case where REMEDIES is two orders of magnitude better with a much shorter computational stencil.

The final examples were unsteady. In the first one, with diffusion but no source, REMEDIES enhanced the accuracy in the solution and reduced the CPU time. The other two are diffusionless with a source that depends on the solution in different ways. In the first one, FaP was the best approach due to the treatment of the source by CIR, the accuracy with REMEDIES was deteriorated in comparison. Pretty small differences were found with both approaches in the decaying wave travelling example.

Generally speaking, this study shows than REMEDIES is a viable alternative to FaP, reducing considerably the run time and providing a more accurate solution in numerous cases.

Finally, since the scheme is employed through an iterative procedure, theoretical questions such as stability, consistency, and convergence should be addressed in upcoming versions of 2D ENATE.

CRediT authorship contribution statement

Víctor J. Llorente: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Antonio Pascau:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Supervision.

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.

Data availability

Data will be made available on request.

Acknowledgements

The study was supported by the European Union through FEDER funding and Diputación General de Aragón “Construyendo Europa desde Aragón” [Government of Aragon “Building Europe from Aragon”], support that is gratefully acknowledged.

Appendix. Floating-point operations

The calculations of arithmetical operations in ENATE with FaP and with REMEDIES take into account the number of sums, subtractions, multiplications, and divisions in all numerical procedures. The # symbol stands for number of arithmetical operations for a numerical procedure. In general, four numerical procedures take place in ENATE, namely:

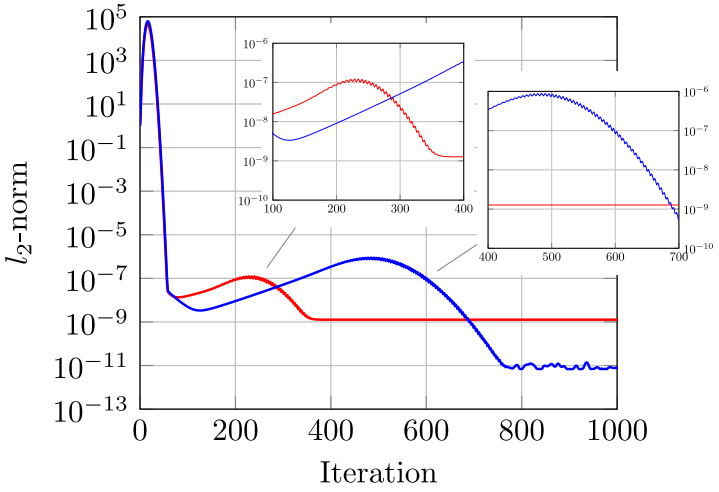

Fig. 16. l_2-norm of the error versus number of iterations for Case 5.4, the stiff-source hyperbolic equation, in REMEDIES with Quintic Hermite. Red line: 100×100, Blue line: 200×200.

Table A.1
Counting the number of sweeps.

Sweeping in x direction	p_x	Sweeping in y direction	p_y
east → west → east	1/2	south → north → south	1/2
east → west → east → west → east	1	south → north → south → north → south	1
⋮	⋮	⋮	⋮
east $\underbrace{\rightarrow \text{west} \rightarrow \ldots \rightarrow}_{a \text{ times}}$ east	$a/4$	south $\underbrace{\rightarrow \text{north} \rightarrow \ldots \rightarrow}_{b \text{ times}}$ south	$b/4$

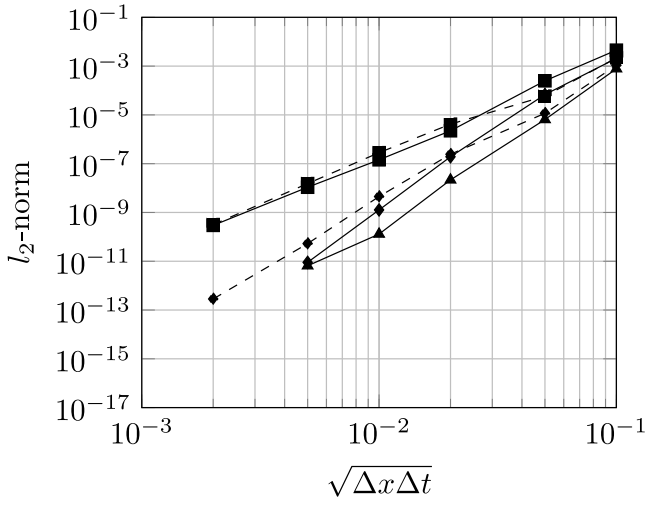

Fig. 17. l_2-norm of the error for Case 5.4, the hyperbolic equation, with $u = 1$ and $\tau = 0.05$. ■ Cubic Hermite. ♦ Quintic Hermite. ▲ Septic Hermite. Solid line, REMEDIES. Dashed line, FaP.

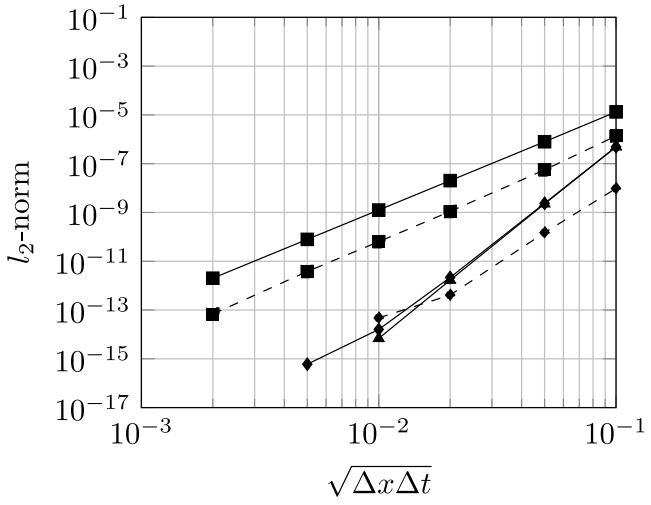

Fig. 18. l_2-norm of the error for Case 5.4, the stiff-source hyperbolic equation, with $u = 1$ and $\tau = 0.5$. ■ Cubic Hermite. ♦ Quintic Hermite. ▲ Septic Hermite. Solid line, REMEDIES. Dashed line, FaP.

Tri-Diagonal Matrix Algorithm, Central Compact Scheme, Compact Integration Rules, and Gauss–Seidel line-by-line method.

The Tri-Diagonal Matrix (Thomas) Algorithm implemented in the code has

$$\begin{array}{l}\#\text{Sums/subtractions} = 4m - 5\\ \#\text{Multiplications/divisions} = 6m - 6\\ \hline \#\text{TDMA}(m) = 10m - 11\end{array}$$

operations where m is the number of variables to solve and $m \neq 1$.

The Central Compact Schemes solve a tridiagonal linear system of equations, e.g. $\mathbf{M}\partial\boldsymbol{\phi}/\partial x \approx (1/\Delta x)\mathbf{Q}\boldsymbol{\phi}$, using the Thomas Algorithm. The total number of operations for the stencil, $\mathbf{Q}\boldsymbol{\phi}$, is

$$\begin{array}{l}\#\text{Sums/subtractions} = m + 4\\ \#\text{Multiplications/divisions} = 2m + 6\\ \hline \#\text{TDMA}(m)\\ \hline \#\text{CCS}_4(m) = 13m - 1\end{array}$$

for a 4th order scheme and

$$\begin{array}{l}\#\text{Sums/subtractions} = 3m + 4\\ \#\text{Multiplications/divisions} = 4m + 4\\ \hline \#\text{TDMA}(m)\\ \hline \#\text{CCS}_6(m) = 17m - 3\end{array}$$

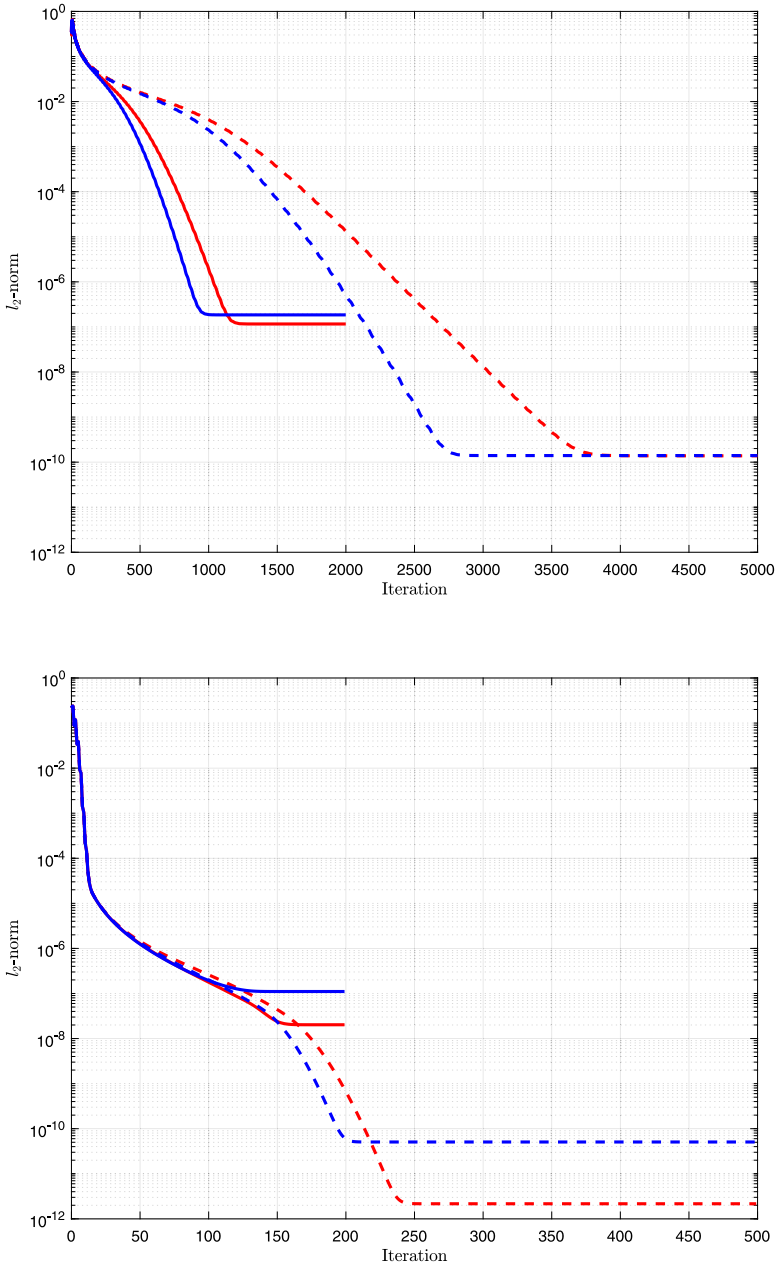

Fig. 19. l_2-norm of the error versus number of iterations for Case 5.4, the stiff-source hyperbolic equation, in FaP (top) and REMEDIES (bottom) for a 50 × 50 mesh. Red line with CCS. Blue line with CIR. Solid line with cubic Hermite. Dashed line with quintic Hermite.

Table A.2
Number of operations for ENATE with FaP.

Operations	Hermite					
	Cubic			Quintic		
	+ and −	× and /	Total	+ and −	× and /	Total
#A's	$3N'_xN'_y$	$N'_xN'_y + 4N''_xN''_y$	$\mathcal{O}(8\overline{N}^2)$	$3N'_xN'_y$	$N'_xN'_y + 4N''_xN''_y$	$\mathcal{O}(8\overline{N}^2)$
#Boundaries	$2N''_x + 2N''_y + 8$	$2N'_x + 2N'_y$	$\mathcal{O}(4N_x, 4N_y)$	$2N''_x + 2N''_y + 8$	$2N'_x + 2N'_y$	$\mathcal{O}(4N_x, 4N_y)$
#S^x	$[1 + 2\#\mathrm{CCS}_4(N_y)]N_x - 4$		$\mathcal{O}(26\overline{N}^2)$	$[1 + \#\mathrm{CCS}_6(N_y)]N_x - 4$		$\mathcal{O}(17\overline{N}^2)$
#S^y	$[1 + 2\#\mathrm{CCS}_4(N_x)]N_y - 4$		$\mathcal{O}(26\overline{N}^2)$	$[1 + \#\mathrm{CCS}_6(N_x)]N_y - 4$		$\mathcal{O}(17\overline{N}^2)$
#$\partial S^x/\partial x$				$N'_y\#\mathrm{CCS}_6(N_x) - 4$		$\mathcal{O}(17\overline{N}^2)$
#$\partial S^y/\partial y$				$N'_x\#\mathrm{CCS}_6(N_y) - 4$		$\mathcal{O}(17\overline{N}^2)$
#IS^x_{01}	$N'_y\#\mathrm{CIR}_4(N_x - 1)$		$\mathcal{O}(13\overline{N}^2)$	$N'_y\#\mathrm{CIR}_6(N_x - 1)$		$\mathcal{O}(17\overline{N}^2)$
#IS^y_{01}	$N'_x\#\mathrm{CIR}_4(N_y - 1)$		$\mathcal{O}(13\overline{N}^2)$	$N'_x\#\mathrm{CIR}_6(N_y - 1)$		$\mathcal{O}(17\overline{N}^2)$
#b's	$10N'_xN'_y$	$10N'_xN'_y$	$\mathcal{O}(20\overline{N}^2)$	$18N'_xN'_y$	$16N'_xN'_y$	$\mathcal{O}(34\overline{N}^2)$

Table A.3
Number of operations for ENATE with REMEDIES.

Operations	Hermite					
	Cubic			Quintic		
	+ and −	× and /	Total	+ and −	× and /	Total
#A's	$3N'_xN'_y$	$2N'_xN'_y + 8N''_xN''_y$	$\mathcal{O}(13\overline{N}^2)$	$3N'_xN'_y$	$N'_xN'_y + 4N''_xN''_y$	$\mathcal{O}(8\overline{N}^2)$
#Boundaries	$2N''_x + 2N''_y + 8$	$2N'_x + 2N'_y$	$\mathcal{O}(4N_x, 4N_y)$	$2N''_x + 2N''_y + 8$	$2N'_x + 2N'_y$	$\mathcal{O}(4N_x, 4N_y)$
#S^x	$N_xN_y - 2$	$N_xN_y - 2$	$\mathcal{O}(2\overline{N}^2)$	$N_xN_y - 2$	$N_xN_y - 2$	$\mathcal{O}(2\overline{N}^2)$
#S^y	$N_xN_y - 2$	$N_xN_y - 2$	$\mathcal{O}(2\overline{N}^2)$	$N_xN_y - 2$	$N_xN_y - 2$	$\mathcal{O}(2\overline{N}^2)$
#ϕ_1	N'_y#TDMA(N'_x)		$\mathcal{O}(10\overline{N}^2)$	N'_y#TDMA(N'_x)		$\mathcal{O}(10\overline{N}^2)$
#ϕ_2	N'_x#TDMA(N'_y)		$\mathcal{O}(10\overline{N}^2)$	N'_x#TDMA(N'_y)		$\mathcal{O}(10\overline{N}^2)$
#$\Delta\phi_2$	$5N'_xN'_y$	$3N'_xN'_y$	$\mathcal{O}(8\overline{N}^2)$	$5N'_xN'_y$	$3N'_xN'_y$	$\mathcal{O}(8\overline{N}^2)$
#$\Delta\beta$	$2N'_xN'_y$	$6N'_xN'_y$	$\mathcal{O}(8\overline{N}^2)$	$2N'_xN'_y$	$6N'_xN'_y$	$\mathcal{O}(8\overline{N}^2)$
#Update	$2N'_xN'_y$		$\mathcal{O}(2\overline{N}^2)$	$2N'_xN'_y$		$\mathcal{O}(2\overline{N}^2)$
#$\partial S^x/\partial x$	N'_y#CCS$_4$(N_x) − 4		$\mathcal{O}(13\overline{N}^2)$	N'_y#CCS$_6$(N_x) − 4		$\mathcal{O}(17\overline{N}^2)$
#$\partial S^y/\partial y$	N'_x#CCS$_4$(N_y) − 4		$\mathcal{O}(13\overline{N}^2)$	N'_x#CCS$_6$(N_y) − 4		$\mathcal{O}(17\overline{N}^2)$
#$\partial^2 S^x/\partial x^2$				N'_y#CCS$_6$(N_x) − 4		$\mathcal{O}(17\overline{N}^2)$
#$\partial^2 S^y/\partial y^2$				N'_y#CCS$_6$(N_y) − 4		$\mathcal{O}(17\overline{N}^2)$
#b's	$14N'_xN'_y$	$12N'_xN'_y$	$\mathcal{O}(26\overline{N}^2)$	$22N'_xN'_y$	$18N'_xN'_y$	$\mathcal{O}(40\overline{N}^2)$

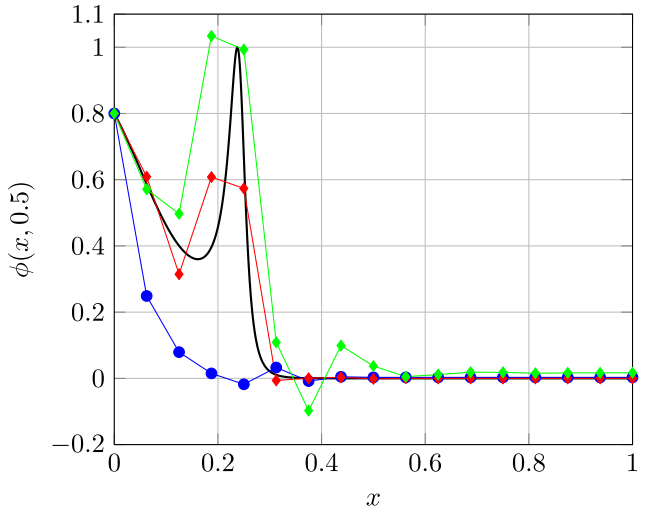

Fig. 20. Exact solution at $t = 0.5$ s (black line) for Case 5.5, the wave travelling equation. Numerical solutions in a mesh with $\Delta x = 0.0625$ m and time step $\Delta t = 0.0625$ s: Explicit Euler method (blue dots), ENATE with FaP (red diamonds) and ENATE with REMEDIES (green diamonds).

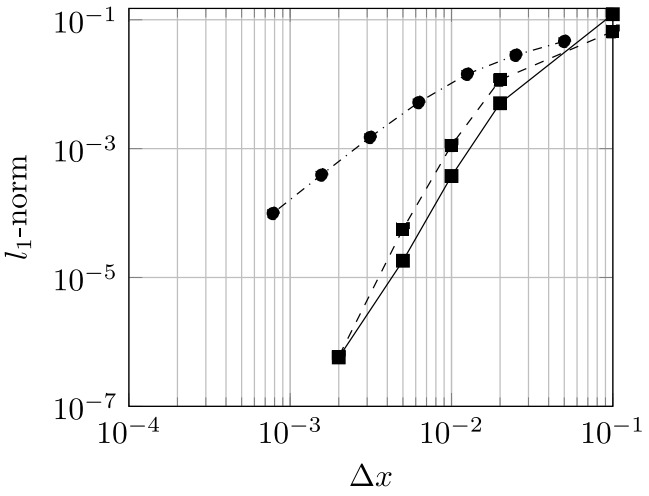

Fig. 21. l_1-norm of the error for Case 5.5, the wave travelling equation. • FV-CF scheme. ■ ENATE. Solid line, REMEDIES. Dashed line, FaP.

for a 6th order scheme. Similarly, the Compact Integration Rules use Thomas to solve $\mathbf{P}\int\boldsymbol{\phi}\mathrm{d}x \approx \Delta x\mathbf{R}\boldsymbol{\phi}$, for example, with a total number of operations for the stencil, $\mathbf{R}\boldsymbol{\phi}$, is

$$\#\text{Sums/subtractions} = m + 2$$

$$\#\text{Multiplications/divisions} = 2\ m$$

$$\frac{\#\text{TDMA}(m)}{\#\text{CIR}_4(m) = 13\ m - 9}$$

for a 4th order scheme and

$$\#\text{Sums/subtractions} = 3m + 2$$

$$\#\text{Multiplications/divisions} = 2m + 2$$

$$\frac{\#\text{TDMA}(m)}{\#\text{CIR}_6(m) = 15m - 7}$$

for a 6th order scheme.

On the other hand, given a Cartesian mesh, the number of nodes in the x direction is written as N_x, in the y direction as N_y, the total number of nodes as $N = N_xN_y$, and the geometric average nodes as $\overline{N} = \sqrt{N}$. By removing nodes at the boundaries (Dirichlet conditions), the number of inner nodes in the x direction is $N'_x = N_x - 2$ and $N'_y = N_y - 2$ in the y direction. Also is defined $N''_x = N'_x - 1$ and $N''_y = N'_y - 1$.

The building block of the code is the matrix solver performing a Gauss–Seidel line-by-line method. This means: (1) starting with a line (e.g. y = const.) where the neighbouring nodes are known from the previous iteration, (2) solving the system of equations by the Thomas Algorithm, and (3) following 1 for all lines. The resolution of all lines (vertical and horizontal) is named "one sweep". Defining p_x as the number of sweeps in x direction and p_y in y direction, see Table A.1, the number of operations for the Gauss–Seidel line-by-line is

$$\#\text{lbl-TDMA} = 4p_xN'_x\#\text{TDMA}(N'_y) + 4p_yN'_y\#\text{TDMA}(N'_x) + \#\text{Neighbour}$$

where $\#\text{Neighbour} = 24p_xN'_x + 24p_yN'_y$ are the solution points whose contribution is considered as source. Therefore, $\#\text{lbl-TDMA} \sim \mathcal{O}(80\widetilde{p}\overline{N}^2)$ with $\widetilde{p} = (p_x + p_y)/2$ is the average arithmetic sweep.

A.1. ENATE with FaP

In Table A.2 all operations are summarized except #lbl-TDMA, which is an operation in common to both Hermite quadratures. The total number of operations for ENATE is $\#\text{FaP-cubic} \sim \mathcal{O}((106+80\widetilde{p})\overline{N}^2)$ and $\#\text{FaP-quintic} \sim \mathcal{O}((140 + 80\widetilde{p})\overline{N}^2)$.

A.2. ENATE with REMEDIES

In Table A.3 all operations are summarized except #lbl-TDMA, which is an operation in common to both Hermite quadratures. The total number of operations for ENATE is #REMEDIES-cubic $\sim \mathcal{O}((107 + 80\widetilde{p})\overline{N}^2)$ and #REMEDIES-quintic $\sim \mathcal{O}((163 + 80\widetilde{p})\overline{N}^2)$.

References

[1] Roos HG, Stynes M, Tobiska L. Robust numerical methods for singularly perturbed equations. Springer series in computational mathematics, 2nd ed.. vol. 24, Berlin: Springer; 2008, http://dx.doi.org/10.1007/978-3-540-34467-4.

[2] Pascau A, Arici M. An accurate discretization for an inhomogeneous transport equation with arbitrary coefficients and source. Comput & Fluids 2016;125:101–15. http://dx.doi.org/10.1016/j.compfluid.2015.11.006.

[3] Allen D, Southwell R. Relaxation methods applied to determining the motion in two dimensions of a viscous fluid past a fixed cylinder. Q J Mech Appl Math 1955;8(2):129–45. http://dx.doi.org/10.1093/qjmam/8.2.129.

[4] Il'in AM. A difference scheme for a differential equation with a small parameter multiplying the highest derivative. Math Notes 1969;6:237–48. http://dx.doi.org/10.1007/BF01093706, (in Russian).

[5] Scharfetter DL, Gummel HK. Large-signal analysis of a silicon read diode oscillator. IEEE Trans Electron Devices 1969;16(1):64–77. http://dx.doi.org/10.1109/T-ED.1969.16566.

[6] Sacco R. Exponentially fitted shape functions for advection-dominated flow problems in two dimensions. J Comput Appl Math 1996;67(1):161–5. http://dx.doi.org/10.1016/0377-0427(95)00149-2.

[7] Mijalkovic S. Exponentially fitted discretization schemes for diffusion process simulation on coarse grids. IEEE Trans Comput Aided Des Integr Circuits Syst 1996;15(5):484–92. http://dx.doi.org/10.1109/43.506136.

[8] Patankar SV. Numerical heat transfer and fluid flow. UK: Taylor & Francis; 1980, http://dx.doi.org/10.1201/9781482234213.

[9] Wong HH, Raithby GD. Improved finite-difference methods based on a critical evaluation of the approximation errors. Numer Heat Transfer 1979;2(2):139–63. http://dx.doi.org/10.1080/10407787908913404.

[10] Thiart GD. Improved finite-difference scheme for the solution of convection-diffusion problems with the SIMPLEN algorithm. Numer Heat Tr B-Fund 1990;18(1):81–95. http://dx.doi.org/10.1080/10407799008944943.

[11] ten Thije Boonkkamp JHM, Anthonissen MJH. The finite volume-complete flux scheme for advection-diffusion-reaction equations. J Sci Comput 2011;46:47–70. http://dx.doi.org/10.1007/s10915-010-9388-8.

[12] Llorente VJ, ten Thije Boonkkamp JHM, Pascau A, Anthonissen MJH. Similarities and differences of two exponential schemes for convection-diffusion problems: The FV-CF and ENATE schemes. Appl Math Comput 2020;365:124700. http://dx.doi.org/10.1016/j.amc.2019.124700.

[13] Angermann L, Wang S. A super-convergent unsymmetric finite volume method for convection–diffusion equations. J Comput Appl Math 2019;358:179–89. http://dx.doi.org/10.1016/j.cam.2019.03.017.

[14] Pascau A. An exact discretization for a transport equation with piecewise-constant coefficients and arbitrary source. Comput & Fluids 2013;75:42–50. http://dx.doi.org/10.1016/j.compfluid.2013.01.009.

[15] Pascau A, Llorente VJ. Contributions to ENATE. Internal Report, Fluid Mechanics Group. University of Zaragoza.

[16] Lele SK. Compact finite difference schemes with spectral-like resolution. J Comput Phys 1992;103(1):16–42. http://dx.doi.org/10.1016/0021-9991(92)90324-R.

[17] Llorente VJ, Pascau A. Compact integration rules as a quadrature method with some applications. Comput Math Appl 2020;79(5):1241–65. http://dx.doi.org/10.1016/j.camwa.2019.08.038.

[18] Lee W, Kim DW. Localized axial Green's function method for the convection–diffusion equations in arbitrary domains. J Comput Phys 2014;275:390–414. http://dx.doi.org/10.1016/j.jcp.2014.06.050.

[19] Abdallah S. Numerical solutions for the pressure Poisson equation with Neumann boundary conditions using a non-staggered grid, I. J Comput Phys 1987;70(1):182–92. http://dx.doi.org/10.1016/0021-9991(87)90008-8.

[20] Abdallah S. Numerical solutions for the incompressible Navier–Stokes equations in primitive variables using a non-staggered grid, II. J Comput Phys 1987;70(1):193–202. http://dx.doi.org/10.1016/0021-9991(87)90009-X.

[21] Zapata MU, Balam RI. High-order implicit finite difference schemes for the two-dimensional Poisson equation. Appl Math Comput 2017;309:222–44. http://dx.doi.org/10.1016/j.amc.2017.04.006.

[22] Yong WA. Basic aspects of hyperbolic relaxation systems. In: Freistuhler H, Szepessy A, editors. Advances in the theory of shock waves. Progress in nonlinear differential equations and their applications, vol. 47, Boston, MA: Birkhauser; 2001, http://dx.doi.org/10.1007/978-1-4612-0193-9_4.

[23] Bhatnagar PL, Gross EP, Krook M. A model for collisional processes in gases. I. Small amplitude processes in charged and neutral one-component systems. Phys Rev 1954;94(3):511–25. http://dx.doi.org/10.1103/PhysRev.94.511.