Concurrency and Computation: Practice and Experience

WILEY

SPECIAL ISSUE PAPER OPEN ACCESS

How Accurate is Richardson's Error Estimate?

Carl Christian Kjelgaard Mikkelsen[1] | Lorién López-Villellas[2]

[1]Department of Computer Science, Umeå University, Umeå, Sweden | [2]Departamento de Informática e Ingeniería de Sistemas/Aragón Institute for Engineering Research (I3A), Universidad de Zaragoza, Zaragoza, Spain

Correspondence: Carl Christian Kjelgaard Mikkelsen (spock@cs.umu.se)

Received: 14 February 2025 | **Revised:** 6 August 2025 | **Accepted:** 3 September 2025

Funding: This work was supported by eSSENCE, the Swedish e-Science collaboration. This work was also supported by MCIN/AEI/10.13039/501100011033 (grant PID2022-136454NB-C22), and by Government of Aragon (T58_23R research group).

Keywords: discretization errors | modeling errors | practical error estimation | Richardson extrapolation | rounding errors

ABSTRACT

We consider the fundamental problem of estimating the difference between the exact value T and approximations A_h that depend on a single real parameter h. It is well-known that if the error $E_h = T - A_h$ satisfies an asymptotic expansion, then we can use Richardson extrapolation to approximate E_h. In this paper, our primary concern is the accuracy of Richardson's error estimate R_h, that is, the size of the relative error $(E_h - R_h)/E_h$. In practice, the computed value $\hat{A}_h$ is different from the exact value A_h. We show how to determine when the computational error $A_h - \hat{A}_h$ is irrelevant and how to estimate the accuracy of Richardson's error estimate in terms of Richardson's fraction F_h. We establish monotone convergence theorems and derive upper and lower bounds for T in terms of A_h and R_h. We classify asymptotic error expansions according to their practical value rather than the order of the primary error term. We present a sequence of numerical experiments that illustrate the theory. Weierstrass's function is used to define a sequence of smooth problems for which it is impractical to apply Richardson's techniques.

1 | Introduction

Consider the fundamental problem of improving a model of a physical phenomenon. Let P denote the value of a parameter that can be measured in the real world, and let T denote the value that is predicted by our model. We wish to compute the modeling error $P - T$ so that we may adjust our model accordingly. Our task is complicated by the fact that we cannot compute the exact value of T. Instead, we rely on approximations A_h that depend on a single real parameter h, such as the time step used to integrate the system of differential equations that defines our model. In practice, errors such as rounding and truncation errors ensure that the computed value $\hat{A}_h$ is different from the true value of A_h, and we have to contend with the fact that we cannot be certain that

$$P - \hat{A}_h \approx P - T = P - \hat{A}_h - (T - A_h) - (A_h - \hat{A}_h) \qquad (1)$$

is a good approximation unless we are certain that the discretization error $E_h = T - A_h$ is much smaller than $P - \hat{A}_h$ and that the computational error $A_h - \hat{A}_h$ is irrelevant. If the discretization error E_h satisfies an asymptotic expansion of the form

$$E_h = \alpha h^p + \beta h^q + o(h^q), \quad h \to 0_+ \qquad (2)$$

then we can use Richardson extrapolation to approximate E_h [1, 2]. In particular, we know that Richardson's error estimate R_h given by

$$R_h = \frac{A_h - A_{2h}}{2^p - 1} \qquad (3)$$

is a good approximation of E_h when h is sufficiently small, but how do we recognize this threshold, and when can we no longer ignore the difference between A_h and $\hat{A}_h$? In this paper, we derive theorems that will allow us to identify the range of h-values where

R_h is a good approximation of the discretization error E_h and the computational error $A_h - \hat{A}_h$ is irrelevant. Moreover, we show how to estimate the accuracy of Richardson's error estimate, that is, the value of the relative error $(E_h - R_h)/E_h$ without knowing the true value of T. Central to these results is the behavior of Richardson's fraction F_h given by

$$F_h = \frac{A_{2h} - A_{4h}}{A_h - A_{2h}}. \tag{4}$$

It is natural to focus on the primary error term αh^p at the expense of the secondary error term βh^q. Why should one care about the value of a term that is ultimately insignificant? In this paper, we shall demonstrate the utility of knowing the value of both p and q as well as the signs of α and β.

Our main results appear in Section 2 and our numerical experiments are presented in Section 3. As Richardson extrapolation sees heavy use in the field of computational fluid dynamics, we use Section 4 to relate our findings to the procedures that have become standard in this field. In Section 5, we shall discuss practical implications of our work. Some of our experiments build on a set of elementary results that might not be well-known. We have relegated these matters to the appendix, where we supply either a reference or an elementary proof.

This paper is an extended version of a paper that was presented at PPAM 2024 [3]. At the time, we were primarily preoccupied with the need for accuracy and smoothness in numerical simulations, and these topics shall remain central. After all, we cannot expect that $\hat{A}_h$ will behave in a manner that is amenable to analysis if the computational error $A_h - \hat{A}_h$ is too large and it is no surprise that useful error expansions are more likely to exist if the functions defining our model are many times differentiable. However, as we worked on this manuscript, we eventually understood that smoothness is not enough, and we eventually found smooth problems for which it is impractical to apply Richardson's techniques.

2 | Main Results

Consider the problem of computing a target value $T \in \mathbb{R}$ using approximations A_h that depend on a single real parameter $h > 0$. Our fundamental assumption is that the error $E_h = T - A_h$ has an asymptotic expansion. Our goal is to develop theorems that will allow us to determine when the computational error $A_h - \hat{A}_h$ is irrelevant, and estimate the error $\hat{E}_h = T - \hat{A}_h$ as accurately as possible.

Definition 1. Let $\{p_j\}_{j=1}^{\infty} \subset (0, \infty)$ denote a strictly increasing sequence. We say that E_h has an asymptotic expansion of order m with respect to $\{p_j\}_{j=1}^{\infty}$ if there exists $\{\alpha_j\}_{j=1}^{m} \subset \mathbb{R}$ such that

$$E_h = \sum_{j=1}^{m} \alpha_j h^{p_j} + o(h^{p_m}), \quad h \to 0_+ \tag{5}$$

We are mainly interested in the case where there exists $\alpha \neq 0$ and $\beta \neq 0$ and real exponents $0 < p < q$ such that

$$E_h = \alpha h^p + \beta h^q + o(h^q), \quad h \to 0_+ \tag{6}$$

However, we wish to entertain the possibility that the order m might be one or even zero. We therefore make the following definition.

Definition 2. We shall classify asymptotic expansions as follows:

1. We say that E_h has an asymptotic expansion of Type III if

$$E_h = \alpha h^p + \beta h^q + o(h^q), \quad h \to 0_+. \tag{7}$$

2. We say that E_h has an asymptotic expansion of Type II if

$$E_h = \alpha h^p + o(h^p), \quad h \to 0_+. \tag{8}$$

3. We say that E_h has an asymptotic expansion of Type I if

$$E_h = o(1), \quad h \to 0_+. \tag{9}$$

It is no surprise that expansions of Type I have little practical value, but the difference between expansions of Type II and Type III shall prove significant. To stress these differences, it is convenient to explore the expansions in descending order.

2.1 | Error Expansions of Type III

We begin by deriving a simple representation of Richardson's fraction F_h and Richardson's error estimate R_h, which were both defined in Section 1. These results will allow us to describe the behavior of the exact value of F_h and R_h.

Lemma 1. *Assume that $E_h = T - A_h$ has an asymptotic expansion of Type III. Let m be given by*

$$m = q - p > 0 \tag{10}$$

and let $(c_1, c_2, c_3) \in \mathbb{R}^3$ be given by

$$c_1 = \frac{\beta}{\alpha}, \quad c_2 = \frac{2^q - 1}{2^p - 1} c_1, \quad c_3 = 2^m c_2. \tag{11}$$

Then

$$\frac{R_h}{E_h} = \frac{1 + c_2 h^m + o(h^m)}{1 + c_1 h^m + o(h^m)}, \quad h \to 0_+ \tag{12}$$

and

$$F_h = 2^p \frac{1 + c_3 h^m + o(h^m)}{1 + c_2 h^m + o(h^m)}, \quad h \to 0_+. \tag{13}$$

Proof. We begin by examining Richardson's error estimate R_h. By definition, there is a function $g_1(h) = o(h^q)$ such that

$$E_h = T - A_h = \alpha h^p + \beta h^q + g_1(h). \tag{14}$$

It follows that

$$E_{2h} = T - A_{2h} = 2^p \alpha h^p + 2^q \beta h^q + g_1(2h). \tag{15}$$

We conclude that

$$\begin{aligned} A_h - A_{2h} &= (T - A_{2h}) - (T - A_h) \\ &= (2^p - 1)\alpha h^p + (2^q - 1)\beta h^q + g_1(2h) - g_1(h) \end{aligned} \tag{16}$$

or equivalently

$$R_h = \frac{A_h - A_{2h}}{2^p - 1} = \alpha h^p + \frac{2^q - 1}{2^p - 1}\beta h^q + g_2(h) \tag{17}$$

where

$$g_2(h) = \frac{g_1(2h) - g_1(h)}{2^p - 1} = o(h^q) \tag{18}$$

because $g_1(h) = o(h^q)$. This allows us to write

$$\begin{aligned} \frac{R_h}{E_h} &= \frac{\alpha h^p + \frac{2^q-1}{2^p-1}\beta h^q + g_2(h)}{\alpha h^p + \beta h^q + g_1(h)} = \frac{1 + c_2 h^m + z_2(h)}{1 + c_1 h^m + z_1(h)}, \\ z_i(h) &= \frac{g_i(h)}{\alpha h^p}, \quad i \in \{1, 2\} \end{aligned} \tag{19}$$

where $z_i(h) = o(h^m)$ because $g_i(h) = o(h^q)$.

We shall now examine Richardson's fraction F_h. Equation (16) implies that

$$A_{2h} - A_{4h} = 2^p(2^p - 1)\alpha h^p + 2^q(2^q - 1)\beta h^q + g_1(4h) - g_1(2h). \tag{20}$$

It follows that

$$\begin{aligned} F_h &= \frac{A_{2h} - A_{4h}}{A_h - A_{2h}} \\ &= \frac{2^p(2^p - 1)\alpha h^p + 2^q(2^q - 1)\beta h^q + g_1(4h) - g_1(2h)}{(2^p - 1)\alpha h^p + (2^q - 1)\beta h^q + g_1(2h) - g_1(h)} \end{aligned} \tag{21}$$

This allows us to write

$$F_h = 2^p \frac{1 + c_3 h^m + z_3(h)}{1 + c_2 h^m + z_4(h)} \tag{22}$$

where

$$z_3(h) = \frac{g_1(4h) - g_1(2h)}{2^p(2^p - 1)\alpha h^p} = o(h^m), z_4(h) = \frac{g_1(2h) - g_1(h)}{(2^p - 1)\alpha h^p} = o(h^m). \tag{23}$$

because $g_1(h) = o(h^q)$. This completes the proof. □

We shall use the following theorem to estimate the accuracy of the approximation $E_h \approx R_h$ in the general case where T is unknown.

Theorem 1. *Assume that* $E_h = T - A_h$ *has an asymptotic expansion of Type III. Let* m *be given by*

$$m = q - p > 0 \tag{24}$$

and let $(c_4, c_5) \in \mathbb{R}^2$ *be given by*

$$c_4 = \frac{2^q - 2^p}{2^p - 1}\frac{\beta}{\alpha}, \quad c_5 = (2^q - 1)c_4 \tag{25}$$

Then the following statements are true

$$\frac{R_h}{E_h} \to 1, \quad h \to 0_+, \quad F_h \to 2^p, \quad h \to 0_+. \tag{26}$$

We also have

$$\frac{(R_h - E_h)/E_h}{h^m} \to c_4, \quad h \to 0_+, \quad \frac{F_h - 2^p}{h^m} \to c_5, \quad h \to 0_+. \tag{27}$$

Proof. By Lemma 1 we have

$$\frac{R_h}{E_h} = \frac{1 + c_2 h^m + o(h^m)}{1 + c_1 h^m + o(h^m)} \to 1, \quad h \to 0_+ \tag{28}$$

and

$$F_h = 2^p \frac{1 + c_3 h^m + o(h^m)}{1 + c_2 h^m + o(h^m)} \to 2^p, \qquad h \to 0_+. \tag{29}$$

It remains to analyze the details of the convergence. In each case, the problem reduces to the analysis of a function of the form

$$f(h) = \frac{1 + a(h)h^m}{1 + b(h)h^m} \tag{30}$$

where

$$a(h) \to a, \quad h \to 0_+, \quad b(h) \to b, \quad h \to 0_+. \tag{31}$$

We have

$$f(h) - 1 = \frac{1 + a(h)h^m}{1 + b(h)h^m} - \frac{1 + b(h)h^m}{1 + b(h)h^m} = \frac{(a(h) - b(h))h^m}{1 + b(h)h^m}. \tag{32}$$

It follows immediately that

$$\frac{f(h) - 1}{h^m} \to (a - b), \quad h \to 0_+. \tag{33}$$

In the case of R_h/E_h we have

$$a - b = c_2 - c_1 = \left(\frac{2^q - 1}{2^p - 1} - 1\right)\frac{\beta}{\alpha} = \frac{2^q - 2^p}{2^p - 1}\frac{\beta}{\alpha} = c_4 \tag{34}$$

and in the case of F_h we have

$$a - b = 2^p(c_3 - c_2) = 2^p(2^m - 1)\frac{2^q - 1}{2^p - 1}\frac{\beta}{\alpha} = (2^q - 1)\frac{2^q - 2^p}{2^p - 1}\frac{\beta}{\alpha} = c_5. \tag{35}$$

This completes the proof. □

What is the practical value of Theorem 1? At first glance, it is merely a statement about the behavior of F_h and E_h/R_h in *exact* arithmetic. In practice, computational errors such as rounding and truncation errors prevent us from computing the exact value of A_h, and we have to accept the fact that $A_h \neq \hat{A}_h$. At this point, we stress that the behavior of R_h and F_h is controlled entirely by the difference between successive approximations. In particular, as long as the *computed* value of F_h behaves in a manner that is consistent with Theorem 1, then there is no reason to believe that the difference between (A_h, A_{2h}) and $(\hat{A}_h, \hat{A}_{2h})$ is significant compared with $A_h - A_{2h}$. In particular, if we have reason to believe that p is an integer, then the only possible value is typically easy to recognize because $F_h \to 2^p$ and as long as

$$2^p - \hat{F}_h \approx c_5 h^m \tag{36}$$

is a good approximation, then we have no reason to believe that the computational error is significant compared with the discretization error that we seek to estimate. In this range, we have no evidence that higher-order terms are significant, and we can ignore the difference between A_h and $\hat{A}_h$. In general, we do not know the value of c_5, but a human will often find it is easy to recognize when

$$\log|2^p - \hat{F}_h| \approx \log|c_5| + m\log(h) \tag{37}$$

is a good approximation. We simply plot $\log|2^p - \hat{F}_h|$ as a function of $\log(h)$ and investigate if the graph is a straight line. This process also yields q as $q = m + p$, where m is the slope of the straight line.

Theorem 1 allows us to relate the accuracy of Richardson's error estimate, that is, the relative error $(E_h - R_h)/E_h$, to the behavior of Richardson's fraction. In particular, we have the following result.

Theorem 2. *Assume that E_h has an asymptotic expansion of Type III. Then*

$$\frac{(E_h - R_h)/E_h}{2^p - F_h} \to \frac{1}{2^q - 1}, \quad h \to 0_+. \tag{38}$$

Proof. Let $m = q - p$. Then by Theorem 1 there are constants c_4 and c_5 such that

$$\begin{aligned}\frac{(E_h - R_h)/E_h}{2^p - F_h} &= \left(\frac{(E_h - R_h)/E_h}{h^m}\right) \Big/ \left(\frac{2^p - F_n}{h^m}\right)\\ &\to \frac{c_4}{c_5} = \frac{1}{2^q - 1} h \to 0_+\end{aligned} \tag{39}$$

This completes the proof. □

Theorem 2 addresses the titular question of our paper. Using Theorem 2 we conclude that the approximation

$$\frac{E_h - R_h}{E_h} \approx \frac{1}{2^q - 1}(2^p - F_h) =: C_h \tag{40}$$

will eventually be good. Here, the right-hand side can be computed in terms of the approximations A_h, A_{2h}, and A_{4h}, while the error $E_h = T - A_h$ cannot be computed without knowing T. Alekseev et al. [4] observe that the existing theory of Richardson extrapolation does not include inequalities of the form $|E_h| \leq C$ for computable constants C. We observe that while we cannot use Equation (40) to bound the error E_h with certainty, we at least have the ability to estimate the relative error $(E_h - R_h)/E_h$ and we can also estimate

$$E_h \approx \frac{1}{1 - C_h} R_h. \tag{41}$$

Lemma 1 suggests that the convergence of F_h and $(E_h - R_h)/h^m$ is eventually one-sided and monotone, but our assumptions are not quite strong enough to reach this conclusion. However, there is enough information to establish the following result.

Theorem 3. *Assume that $E_h = T - A_h$ has an asymptotic expansion of Type III. Then there are two distinct possibilities depending on the sign of $\frac{\beta}{\alpha}$.*

1. *If $\frac{\beta}{\alpha} < 0$, then*

$$F_{2h} < F_h < 2^p, \quad \frac{R_{2h}}{E_{2h}} < \frac{R_h}{E_h} < 1 \tag{42}$$

for h sufficiently small.

2. *If $\frac{\beta}{\alpha} > 0$, then*

$$F_{2h} > F_h > 2^p, \quad \frac{R_{2h}}{E_{2h}} > \frac{R_h}{E_h} > 1 \tag{43}$$

for h sufficiently small.

Proof. As in the proof of Theorem 1, the proof reduces to the analysis of a function of the form

$$f(h) = \frac{1 + a(h)h^m}{1 + b(h)h^m} \tag{44}$$

where

$$a(h) \to a, \quad h \to 0_+, \quad b(h) \to b, \quad h \to 0_+. \tag{45}$$

In the case of E_h/R_h we have $a - b = c_4$ and in the case of F_h we have $a - b = c_5$. In either case, we have

$$\operatorname{sign}(a - b) = \operatorname{sign}(\beta/\alpha). \tag{46}$$

We shall now show that

$$\operatorname{sign}[f(2h) - f(h)] = \operatorname{sign}(\beta/\alpha) \neq 0 \tag{47}$$

when h is sufficiently small. By the definition of f, we have

$$f(2h) - f(h) = \frac{1 + 2^m a(2h)h^m}{1 + 2^m b(2h)h^m} - \frac{1 + a(h)h^m}{1 + b(h)h^m} = \frac{T(h)}{N(h)} \tag{48}$$

where

$$T(h) = (1 + 2^m a(2h)h^m)(1 + b(h)h^m) - (1 + a(h)h^m)(1 + 2^m b(2h)h^m). \tag{49}$$

and

$$N(h) = (1 + b(h)h^m)(1 + 2^m b(2h)h^m). \tag{50}$$

We must have $N(h) > 0$ for h sufficiently small because $b(h) \to b$ for $h \to 0_+$. We therefore concentrate on the expression for $T(h)$. We find that

$$\begin{aligned}T(h) = {}& 2^m(a(2h) - b(2h))h^m - (a(h) - b(h)h^m + 2^m[a(2h)b(h)\\ &- a(h)b(2h)]h^{2m}.\end{aligned} \tag{51}$$

It follows that

$$\frac{T(h)}{h^m} \to (2^m - 1)(a - b) \neq 0, \quad h \to 0_+. \tag{52}$$

We conclude that

$$\operatorname{sign}\left(\frac{T(h)}{N(h)}\right) = \operatorname{sign}(a - b) = \operatorname{sign}(\beta/\alpha) \tag{53}$$

when h is sufficiently small. For such h, it now follows that

$$f(2h) > f(h) \tag{54}$$

when $\beta/\alpha > 0$ and that

$$f(2h) < f(h) \tag{55}$$

when $\beta/\alpha < 0$. This completes the proof. □

In practice, we often compute A_h for $h = h_k = 2^{-k}h_0$ for a suitable range of k. In this case, Theorem 3 shows that the convergence of both the relative error $(E_h - R_h)/E_h$ and Richardson's fraction is eventually both one-sided and monotone. We shall use this observation to establish bounds on the target value T. The following theorem shows that there is a distinct difference between the case of $\beta/\alpha < 0$ and the case of $\beta/\alpha > 0$.

Theorem 4. *Assume that $E_h = T - A_h$ has an asymptotic expansion of Type III. Then the following statements are true*:

1. *If $\beta/\alpha < 0$ and h is sufficiently small, then*

$$T < A_h + R_h < A_h \tag{56}$$

when $R_h < 0$ and

$$A_h < A_h + R_h < T \tag{57}$$

when $R_h > 0$.

2. *If $\beta/\alpha > 0$ and h is sufficiently small, then*

$$A_h + R_h < T < A_h \tag{58}$$

when $R_h < 0$ and

$$A_h < T < A_h + R_h \tag{59}$$

when $R_h > 0$.

Proof. The proof is a direct application of Theorems 1 and 3. By Theorem 1 we have that R_h and E_h eventually have the same sign simply because $R_h/E_h \to 1$ as $h \to 0_+$. We now split the proof into the case of $\beta/\alpha < 0$ and the case of $\beta/\alpha > 0$.

1. Assume that $\beta/\alpha < 0$. By Theorem 3 we have

$$0 < R_h/E_h < 1 \tag{60}$$

for all sufficiently small h. We must now distinguish between $R_h < 0$ and $R_h > 0$.

a. If $R_h < 0$ then we must have $E_h < R_h$. It follows that

$$T = A_h + E_h < A_h + R_h < A_h \tag{61}$$

b. If $R_h > 0$ then we must have $E_h > R_h$. It follows that

$$T = A_h + E_h > A_h + R_h > A_h \tag{62}$$

2. Assume that $\beta/\alpha > 0$. By Theorem 3 we have

$$1 < R_h/E_h \tag{63}$$

for all sufficiently small h. We must now distinguish between $R_h < 0$ and $R_h > 0$.

a. If $R_h < 0$, then we must have $R_h < E_h < 0$. It follows that

$$A_h + R_h < A_h + E_h < A_h \tag{64}$$

and since $T = A_h + E_h$ we have established inequality (58).

b. If $R_h > 0$, then we must have $0 < E_h < R_h$. It follows that

$$A_h < A_h + E_h < A_h + R_h \tag{65}$$

and since $T = A_h + E_h$ we have established inequality (59).

This completes the proof. □

What is the real value of Theorem 4? It allows us to establish bounds for the target value T in terms of numbers that we can compute. It is interesting to note that while we can bracket the target value T between A_h and $A_h + R_h$ when $\beta/\alpha > 0$, we can only establish either lower or upper bounds for T when $\beta/\alpha < 0$.

2.2 | Error Expansions of Type II

We shall now explore the limitations of error expansions of Type II.

Lemma 2. *If the discretization error E_h has an asymptotic expansion of Type II, then there are functions $\epsilon_i(h) = o(1)$ for $i \in \{1, 2, 3\}$ such that*

$$\frac{E_h}{R_h} = \frac{1 + \epsilon_1(h)}{1 + \epsilon_2(h)}. \tag{66}$$

and Richardson's fraction F_h satisfies

$$F_h = 2^p \frac{1 + \epsilon_3(h)}{1 + \epsilon_2(h)}. \tag{67}$$

Proof. By definition, there is a function $g_1(h) = o(h^p)$ such that

$$E_h = \alpha h^p + g_1(h). \tag{68}$$

It follows that

$$E_{2h} = 2^p \alpha h^p + g_1(2h). \tag{69}$$

We conclude that

$$A_h - A_{2h} = (2^p - 1)\alpha h^p + g_1(2h) - g_1(h) \tag{70}$$

or equivalently

$$R_h = \frac{A_h - A_{2h}}{2^p - 1} = \alpha h^p + g_2(h), \quad g_2(h) = \frac{g_1(2h) - g_1(h)}{2^p - 1} = o(h^p) \tag{71}$$

It follows that

$$\frac{E_h}{R_h} = \frac{1 + \epsilon_1(h)}{1 + \epsilon_2(h)} \tag{72}$$

where

$$\epsilon_i(h) = \frac{g_1(h)}{\alpha h^p} = o(1), \quad i \in \{1, 2\} \tag{73}$$

because $g_i(h) = o(h^p)$. We shall now derive an expression for Richardson's fraction. We have

$$F_h = \frac{A_{2h} - A_{4h}}{A_h - A_{2h}} = \frac{2^p(2^p - 1)\alpha h^p + g_1(4h) - g_1(2h)}{(2^p - 1)\alpha h^p + g_1(2h) - g_1(h)} = 2^p \frac{1 + \epsilon_3(h)}{1 + \epsilon_2(h)} \tag{74}$$

where

$$\epsilon_3(h) = \frac{g_2(2h)}{2^p \alpha h^p} = o(1) \tag{75}$$

because $g_2(h) = o(h^p)$. This completes the proof. □

The following result is an immediate consequence of Lemma 2.

Theorem 5. *Assume that the error E_h has an asymptotic expansion of Type II. Then the following statements are true*:

$$\frac{R_h}{E_h} \to 1, \quad h \to 0_+, \quad F_h \to 2^p, \quad h \to 0_+. \tag{76}$$

The distinct difference between expansions of Type III and Type II is that convergence of both $(E_h - R_h)/E_h$ and F_h can be irregular when the expansion is of Type II. In particular, there is no reason to expect that the quality of the approximation $E_h \approx R_h$ is tightly correlated with the quality of the approximation $F_h \approx 2^p$.

2.3 | Error Expansions of Type I

Suppose that we only know that E_h has an asymptotic expansion of Type I. Then we cannot define Richardson's error estimate, because we do not know which value to assign to p. As for Richardson's fraction, we have

$$A_h - A_{2h} = T - A_{2h} - (T - A_h) = -(E_h - E_{2h}) \tag{77}$$

which allows us to write

$$F_h = \frac{E_{2h} - E_{4h}}{E_h - E_{2h}}. \tag{78}$$

Since we only know that $E_h = o(1)$, we have no reason to expect that F_h will behave in a regular manner.

3 | Numerical Experiments

This section contains a sequence of numerical experiments that exhibit the strength and limitations of Richardson extrapolation as well as the theory developed in this paper. The software needed to execute every experiment and generate all figures from scratch is freely available for anonymous download. Please read the Data Availability Statement before continuing beyond this point.

3.1 | Experiments Based on Numerical Differentiation

In this section, we consider the problem of computing the derivative $T = f'(x)$ of a differentiable function $f : \mathbb{R} \to \mathbb{R}$, using the simple approximation A_h given by

$$A_h = \frac{f(x+h) - f(x)}{h}. \tag{79}$$

Let $m \in \mathbb{N}$. If $f \in C^m$, then Taylor's theorem implies that $E_h = T - A_h$ has an asymptotic expansion of the form

$$E_h = T - A_h = \sum_{j=1}^{m-1} \alpha_j h^j + o(h^{m-1}), \quad h \to 0, \quad h \neq 0 \tag{80}$$

where

$$\alpha_j = \frac{1}{(j+1)!} f^{(j+1)}(x). \tag{81}$$

We shall use anti-derivatives of Weierstrass's function to illustrate the behavior of error expansions of Type I, II, and III. Weierstrass's function $f : \mathbb{R} \to \mathbb{R}$ is everywhere continuous and nowhere differentiable. Section A.0.1 contains a short discussion of the technical details related to Weierstrass's function and its implementation.

3.1.1 | An Error Expansion of Type I

Let $f_1 : \mathbb{R} \to \mathbb{R}$ be given by

$$f_1(x) = \frac{1}{\pi} \sum_{n=0}^{\infty} (ab^{-1})^n \sin(b^n \pi x), \quad (a, b) = \left(\frac{1}{2}, 3\right). \tag{82}$$

Then $f_1 \in C^1 \setminus C^2$ and f_1' is Weierstrass's function. We conclude that E_h has an asymptotic expansion of Type I. The MATLAB function `wf23dif_type1` approximates $T = f_1'(x)$ at $x = 1.2$ using A_h for $h = h_k = 2^{1-k}$ and $k = \{1, 2, \dots, 25\}$. It prints a table of relevant numbers on the screen and generates Figure 1.

We observe that while the discretization error E_h decreases as k increases, Richardson's error estimate R_h is a poor approximation of E_h and there is no discernible pattern to the behavior of Richardson's fraction F_h. At first glance, this is hardly surprising as we are using a function f_1 of class $C^1 \setminus C^2$. However, our implementation of f_1 only sums finitely many terms and is therefore indistinguishable from the natural realization of a function of class C^∞. Therefore, the true value of this example is to demonstrate the existence of a function f of class C^∞ that is so poorly behaved that we cannot reliably estimate the target $T = f'(x)$ using the approximation A_h.

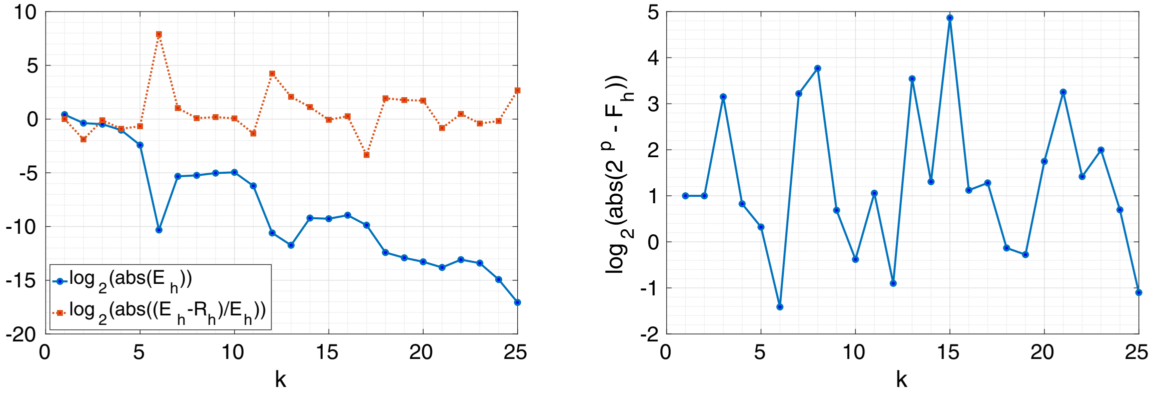

FIGURE 1 | The figures generated by the function `wf23dif_type1`. On the left: The discretization error E_h decays in an irregular manner, and Richardson's error estimate R_h is a poor approximation of E_h. On the right: Richardson's fraction F_h behaves in an irregular manner.

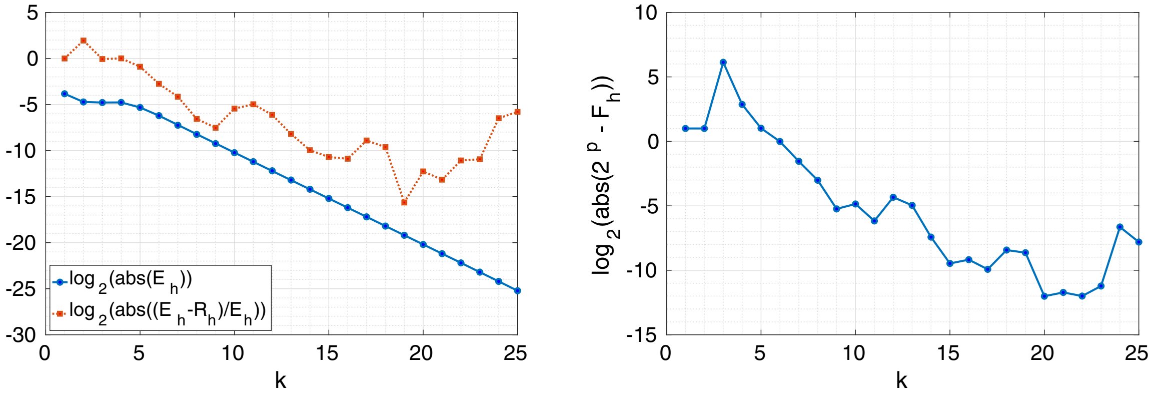

FIGURE 2 | The figures generated by the function `wf23dif_type2`. On the left: The discretization error E_h decays in a regular manner, and Richardson's error estimate R_h is a fair approximation of E_h. On the right: Richardson's fraction F_h tends to 2, but not in a monotone manner.

3.1.2 | An Error Expansion of Type II

Let $f_2 : \mathbb{R} \rightarrow \mathbb{R}$ be given by

$$f_2(x) = -\frac{1}{\pi^2}\sum_{n=0}^{\infty}(ab^{-2})^n \cos(b^n \pi x), \quad (a,b) = \left(\frac{1}{2}, 3\right). \tag{83}$$

Then $f_2 \in C^2 \setminus C^3$ and f_2'' is Weierstrass's function. We conclude that E_h has an asymptotic expansion of Type II. The MATLAB function `wf23dif_type2` approximates $T = f_2'(x)$ at $x = 1.2$ using A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, 2, \dots, 25\}$. It prints a table on the screen and generates Figure 2.

We observe that the discretization error E_h decreases as k increases and that Richardson's fraction F_h approaches 2 in an irregular manner. In particular, there is no obvious correlation between the behavior of Richardson's fraction F_h and the accuracy of the approximation $E_h \approx R_h$, which peaks at $k = 19$. As in the previous example, we must appreciate the fact that our implementation of f_2 only computes a partial sum of the infinite series. It is therefore indistinguishable from the natural realization of a function of class C^∞. The true value of this example is therefore to exhibit a function that is of class C^∞, but behaves as a function of class $C^2 \setminus C^3$.

3.1.3 | An Error Expansion of Type III

Let $f_3 : \mathbb{R} \rightarrow \mathbb{R}$ be given by

$$f_3(x) = -\frac{1}{\pi^3}\sum_{n=0}^{\infty}(ab^{-3})^n \sin(b^n \pi x), \quad (a,b) = \left(\frac{1}{2}, 3\right). \tag{84}$$

Then $f_3 \in C^3 \setminus C^4$ and f_3''' is Weierstrass's function. We conclude that E_h has an asymptotic expansion of Type III. The MATLAB function `wf23dif_mwe3` approximates $T = f_3'(x)$ at $x = 1.2$ using A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, 2, \dots, 25\}$. It prints a table on the screen and generates Figure 3.

The discretization error E_h decays rapidly and the quality of the approximation $E_h \approx R_h$ increases as k increases and $k \in \{2, 3, \dots, 19\}$. We observe that the behavior of Richardson's fraction is consistent with an asymptotic expansion of Type III with $(p, q) = (1, 2)$ until rounding errors become significant after $k = 19$. We stress that our implementation of f_3 only sums finitely many terms, hence it is indistinguishable from the natural realization of a function that is of class C^∞. It follows that the conditions needed to secure an asymptotic expansion of Type III are more than satisfied.

3.2 | Examples Based on Numerical Integration

In this section, we consider the problem of computing the integral

$$T = \int_a^b f(x)dx \tag{85}$$

using the composite trapezoidal rule with uniform step size h, that is,

$$A_h = \frac{h}{2}\sum_{i=0}^{n-1}\big(f(x_i) + f(x_{i+1})\big), \quad nh = (b-a). \tag{86}$$

It is well-known that if $f \in C^{2k+1}([a,b],\mathbb{R})$, then there exists $\{\alpha_j\}_{j=1}^{k} \subset \mathbb{R}$ such that

$$E_h = \sum_{j=1}^{k} \alpha_j h^{2j} + O(h^{2k+1}), \quad h \to 0_+. \tag{87}$$

We shall vary the function f to demonstrate various interesting aspects of this problem. In particular, we shall exhibit functions that are not smooth and smooth functions for which we cannot afford to reach the asymptotic range.

3.2.1 | Monotone Convergence of F_h From Below

This is an extension of an example included in the original paper [3]. Consider the problem of computing the integral $T = \int_0^1 \exp(x)dx = \exp(1) - 1 \approx 1.7183$ using the composite trapezoidal rule. The MATLAB function `rint_below` computes A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, 2, \dots, 20\}$. It prints a table on the screen and generates Figure 4.

The table supports the statement that F_h tends to 4 from below. It follows that $\beta/\alpha < 0$ and we must have $E_h/R_h > 1$, and this fact is also supported by the table. The experiment is consistent with an asymptotic expansion of Type III with $(p,q) = (2,4)$ and to the naked eye the asymptotic range appears to be $k \in \{2, \dots, 13\}$ because $\log_2|2^p - F_h|$ is essentially a linear function of k on this range. However, if we consider the evolution of the relative error $(E_h - R_h)/E_h$, we see a slight bend at $k = 13$ which suggests that rounding errors are starting to be relevant at this point. Since $\frac{\beta}{\alpha} < 0$ we cannot bracket T between A_h and $A_h + R_h$, but we can conclude that

$$T < A_h + R_h < A_h \tag{88}$$

for h sufficiently small. These inequalities are satisfied by the computed values that correspond to $k \in \{2, 3, \dots, 12\}$.

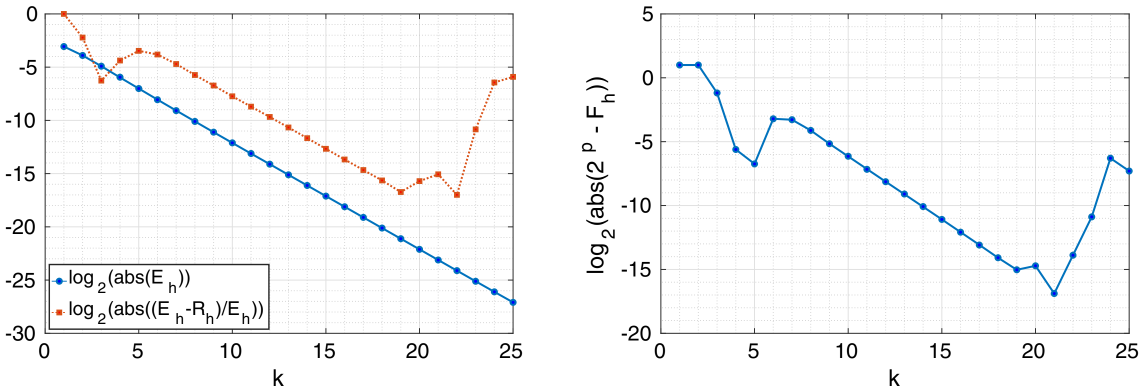

FIGURE 3 | The figures generated by the function `wf23dif_type3`. On the left: The discretization error E_h decays rapidly, and Richardson's error estimate R_h is a good approximation of E_h. On the right: Richardson's fraction F_h behaves in a manner consistent with an error expansion of Type III with $(p,q) = (1,2)$ until rounding errors become significant after $k = 19$.

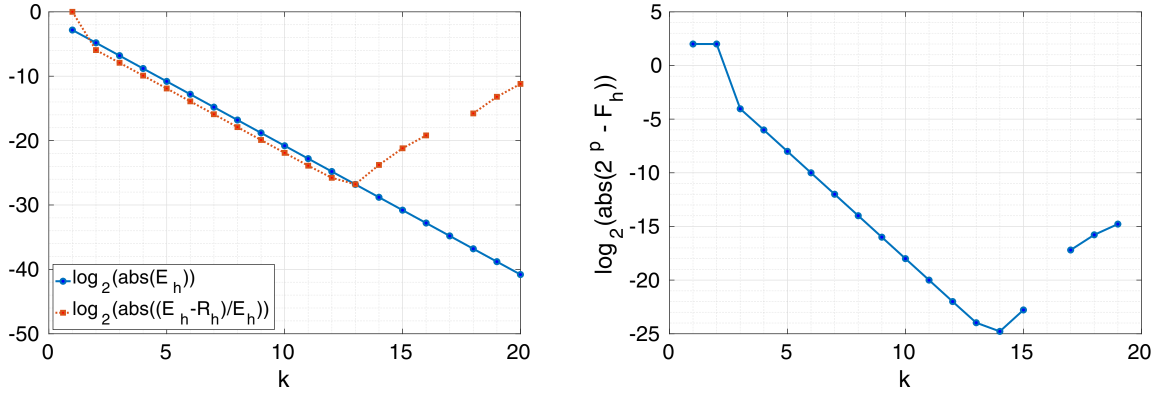

FIGURE 4 | The figures generated by the function `rint_below`. On the left: The discretization error E_h decays rapidly and the quality of the approximation $E_h \approx R_h$ increases monotonically until $k = 13$. On the right: Richardson's fraction F_h tends to 4 in a regular manner until $k = 13$. The occasional data point is missing as one cannot plot $\log_2(0)$ on a finite canvas.

3.2.2 | Monotone Convergence of F_h From Above

Consider the problem of computing the integral $T = \int_0^1 \sin(x)\exp(x)dx = [\frac{1}{2}(\sin(x) - \cos(x))\exp(x)]_0^1 \approx 0.9093$ using the composite trapezoidal rule. The MATLAB function `rint_above` computes A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, 2, \ldots, 20\}$. It prints a table on the screen and generates Figure 5.

The table supports the statement that F_h tends to 4 from above. It follows that $\beta/\alpha > 0$ and we must have $E_h/R_h < 1$, and this fact is also supported by the table. We find that the behavior of Richardson's fraction is consistent with an asymptotic expansion of Type III with $(p, q) = (2, 4)$, and to the naked eye, the asymptotic range appears to be $k \in \{2, \ldots, 13\}$. However, if we consider the evolution of the relative error $(E_h - R_h)/E_h$, we see that rounding errors can be felt already at $k = 13$. Since $\frac{\beta}{\alpha} > 0$ we can bracket T between A_h and $A_h + R_h$ and since R_h is negative we have

$$A_h + R_h < T < A_h \tag{89}$$

for h sufficiently small. These inequalities are satisfied by the computed values corresponding to $k \in \{2, 3, \ldots, 12\}$.

3.2.3 | Reduction of p Due to a Lack of Smoothness

This example was included in the original paper [3]. Let $f : [0, 1] \to \mathbb{R}$ denote the function $f(x) = \sqrt{x}$ and consider the problem of computing the integral $T = \int_0^1 f(x)dx = \frac{2}{3}$ using the trapezoidal rule. The MATLAB function `rint_sqrt` computes A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, 2, \ldots, 25\}$. It prints a table on the screen and generates Figure 6.

From the table, it is clear that $p = 2$ is impossible, while $p = 1.5$ is a plausible candidate. We find that the behavior of Richardson's fraction is consistent with an asymptotic expansion of Type III with $(p, q) = (1.5, 2)$ and the asymptotic range covers $k \in \{2, 3, \ldots, 23\}$.

We mention in passing that low-order methods are more practical than high-order methods in the sense that the observed asymptotic range tends to be larger for low-order methods than for high-order methods. This is due to the fact that the natural implementation of the function $h \to F_h = (A_{2h} - A_{4h})/(A_h - A_{2h})$ suffers from subtractive cancellation when h is sufficiently small. This issue is more acute for high-order methods than for low-order methods, because A_h tends to T much more rapidly for high-order methods than for low-order methods.

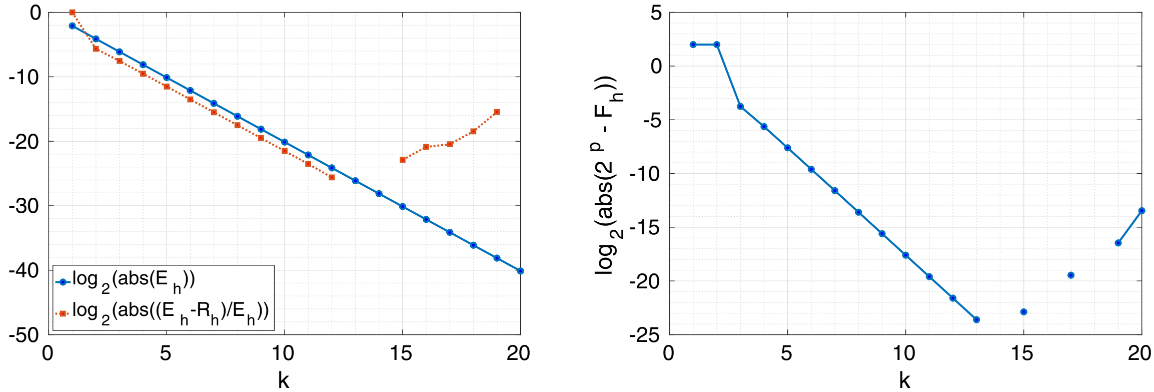

FIGURE 5 | The figures generated by the function `rint_above`. On the left: The discretization error E_h decays rapidly and the quality of the approximation $E_h \approx R_h$ increases monotonically until $k = 12$. On the right: Richardson's fraction tends to 4 in a regular manner until $k = 13$. The occasional data point is missing as one cannot plot $\log_2(0)$ on a finite canvas.

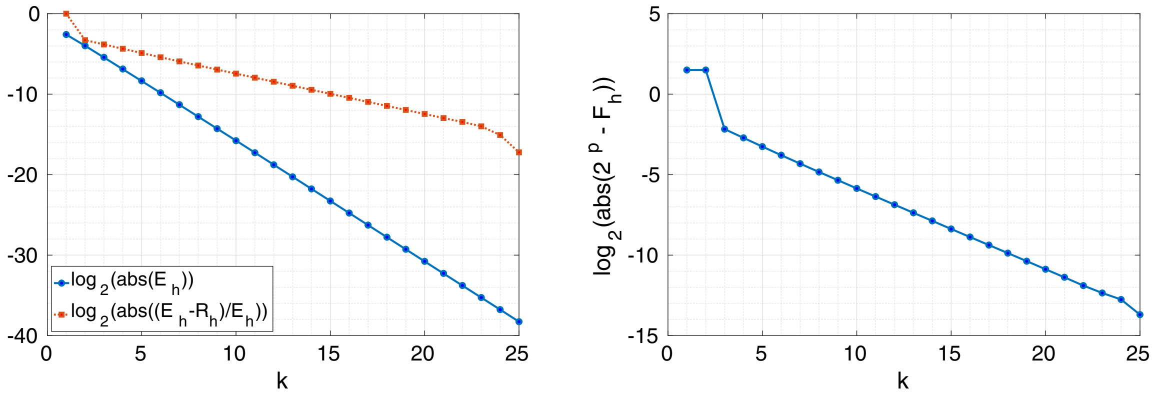

FIGURE 6 | The figures generated by the function `rint_sqrt`. On the left: E_h decays rapidly and R_h is always at least a fair approximation of E_h. On the right: The asymptotic range continues until $k = 24$, where rounding errors can no longer be ignored.

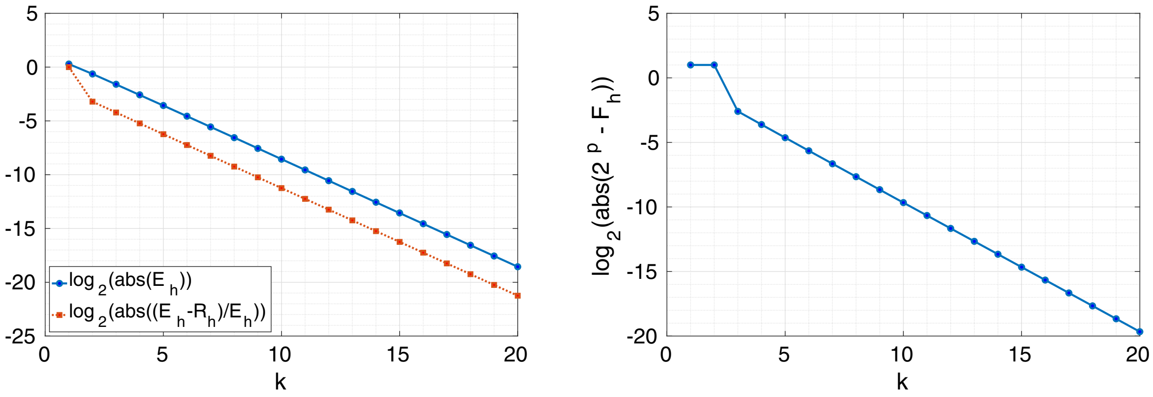

FIGURE 7 | The figures generated by the function `rint_bad`. On the left: Superficially, all is well as E_h decreases and R_h is always a good approximation of E_h, but the programming error has changed the slope. On the right: The asymptotic range is large and covers $k \in \{3, 4, \dots, 20\}$.

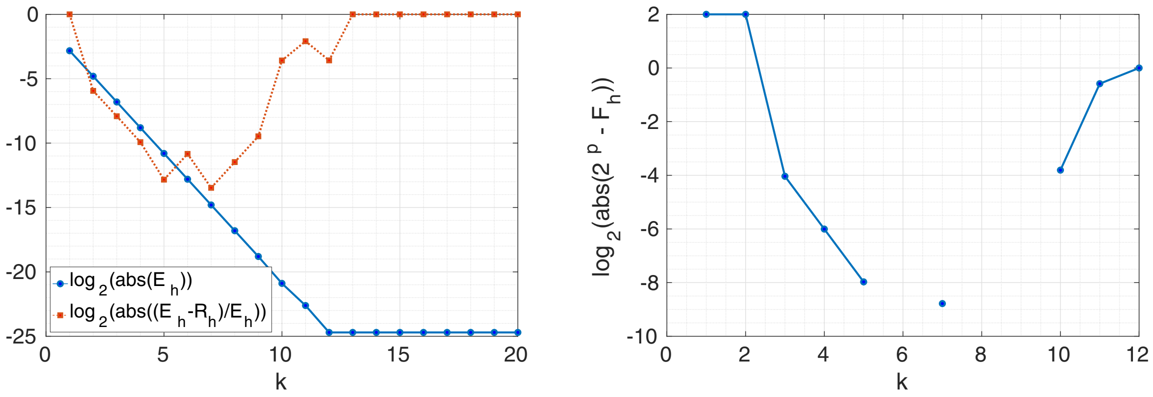

FIGURE 8 | The figures generated by the function `rint_single`. On the left: E_h decays rapidly until the limit imposed by the use of single precision is reached. R_h is a good approximation of E_h initially, but the quality decays rapidly. On the right: We are barely able to identify an asymptotic range.

3.2.4 | Reduction of p Due to a Programming Error

In this section, our target value $T = \int_0^1 \exp(x)dx$ is the same as in Section 3.2.1, but we have deliberately introduced an index error into the software so that the very last term of the sum that defines the composite trapezoidal rule is omitted. The error is proportional to h, and this reduces the order of the primary error term from $p = 2$ to $p = 1$. The MATLAB function `rint_bad` computes the erroneous values of A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, \dots, 20\}$. It prints a table of relevant data on the screen and generates Figure 7.

We see that the experiment supports an error expansion of Type III with $(p, q) = (1, 2)$ rather than the intended goal of $(p, q) = (2, 4)$. In particular, if the error estimates are calculated using the value $p = 2$, then they will be too small and we risk returning plausible approximations A_h that do not meet the user's accuracy goal.

3.2.5 | A Simple Calculation that Cannot be Completed in Single Precision

In this section, our target value $T = \int_0^1 \exp(x)dx$ is the same as in Sections 3.2.1 and 3.2.4, but we attempt to reduce the runtime by evaluating the integrand using single rather than double precision floating point arithmetic. The MATLAB function `rint_single` computes the values of the composite trapezoidal rule A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, \dots, 20\}$ using single precision arithmetic. It prints a table on the screen and generates Figure 8.

As expected, the accuracy has been reduced, but we achieve a relative error $(T - A_h)/T$ that is comparable to the single precision unit roundoff $u = 2^{-24}$. However, the fundamental problem is that our ability to estimate $E_h = T - A_h$ without knowing T has been severely compromised. One could argue that the asymptotic range consists of $k \in \{2, 3, 4\}$, but three points are hardly enough to trace a straight line, and we have no reason to trust Richardson's error estimate outside of this range and certainly not at $k = 12$ where $|E_h|$ reaches its minimal value.

This example stresses the point that Richardson's error estimate and the behavior of Richardson's method hinge on the *difference* between successive approximations. As we approach the limits imposed by our problem and our choice of working precision, there is no longer enough information to compute accurate error estimates and recognize their quality.

This example should not be viewed as an objection to the use of single-precision arithmetic to accelerate select applications. There are situations, such as iterative refinement for systems of linear equations [5] or general quasi-Newton methods for solving systems of non-linear equations [6, 7], where it is quite safe to compute the *correction* of a good approximation using reduced precision arithmetic.

3.2.6 | A Trivial Integral With No Error Estimate

Let $f : [-\pi, \pi] \rightarrow \mathbb{R}$ be given by $f(x) = \sin(x)$ and consider the problem of computing the trivial integral $T = \int_{-\pi}^{\pi} f(x)dx = 0$ using the composite trapezoidal rule. The MATLAB function `rint_null` computes A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, 2, \dots, 20\}$. It prints a table on the screen and generates Figure 9.

By design, the exact value A_h is 0 for all k and the computed value $\hat{A}_h$ consists entirely of round-off errors. Therefore, it is no surprise that Richardson's error estimate provides a poor approximation of the actual error. Moreover, the irregular behavior of Richardson's fraction reminds us of Section 3.1.1 where the expansion was only of Type I.

3.2.7 | A Family of Integrals for Which the Asymptotic Range Cannot be Reached

Let f denote Weierstrass's function corresponding to the choice of $(a, b) = (\frac{1}{2}, 3)$ and consider the problem of computing the target value $T = \int_0^{\frac{1}{2}} f(x)dx$ using the composite trapezoidal rule. The MATLAB function `rint_wf23` computes A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, 2, \dots, 20\}$. It prints a table on the screen and generates Figure 10.

We observe that the error decays in an irregular manner and that Richardson's error estimate is a poor approximation of the error. Moreover, Richardson's fraction behaves in an irregular manner. At first glance, these observations are hardly surprising, because f is nowhere differentiable. However, our implementation of f only sums finitely many terms, hence it is indistinguishable from the natural realization of a function that is of class C^{∞}. It is therefore natural to experiment with different partial sums of Weierstrass's function.

The function `rint_wf23n5` considers the integral of the first $n = 5$ terms of Weierstrass's function. It computes A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, 2, \dots, 20\}$, prints a table on the screen and generates Figure 11.

The evolution of Richardson's fraction suggest that the asymptotic range consists of $k \in \{10, 11, \dots, 17\}$ and this range is included in the range where the relative error $(E_h - R_h)/E_h$ is decreasing, that is, $k \in \{5, 6, \dots, 17\}$.

The function `rint_wf23n10` considers the integral of the first $n = 10$ terms of Weierstrass's function. It computes A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, 2, \dots, 20\}$, prints a table of relevant numbers on the screen, and generates Figure 12.

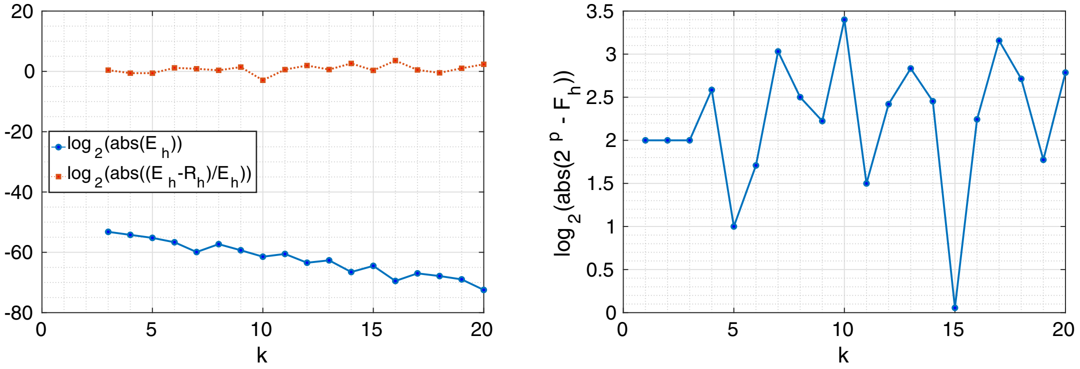

FIGURE 9 | The figures generated by the function `rint_null`. On the left: The computed values of E_h and R_h consist entirely of roundoff errors. On the right: The computed value of F_h behaves in a very irregular manner.

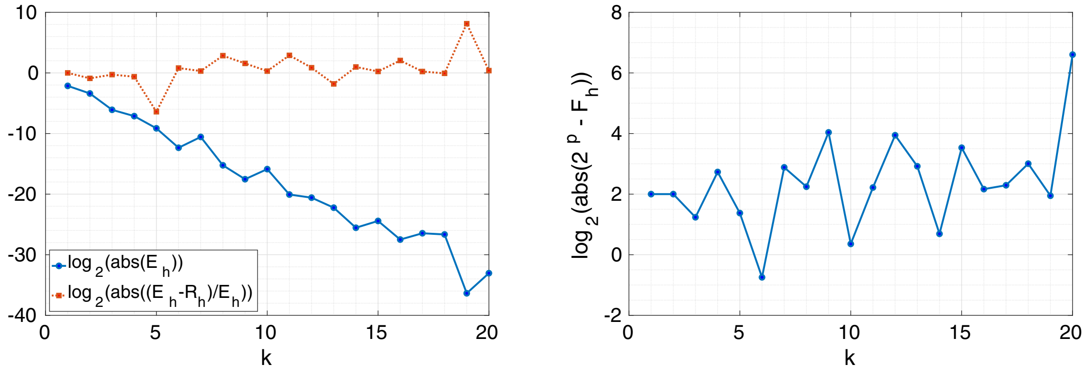

FIGURE 10 | The figures generated by the function `rint_wf23`. On the left: E_h decays rapidly, but R_h is a bad approximation of E_h. On the right: F_h behaves in a very irregular manner.

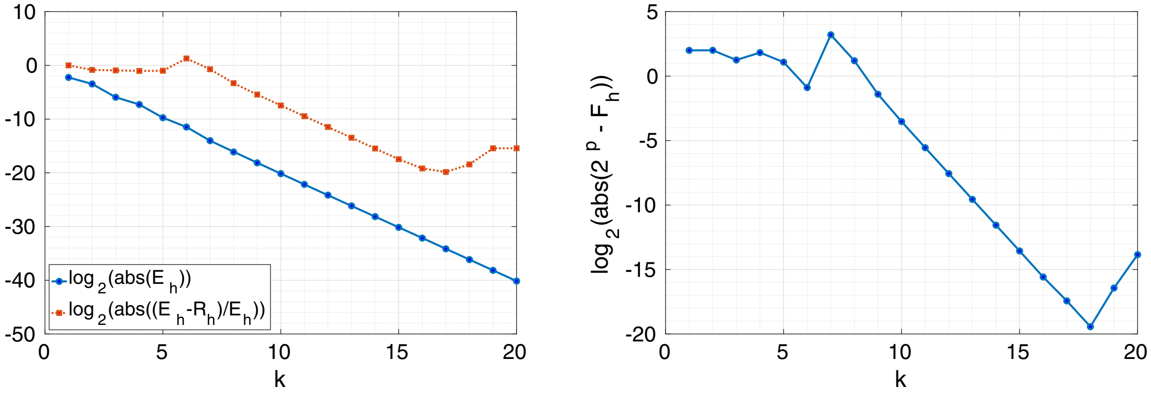

FIGURE 11 | The figures generated by the function `rint_wf23n5`. On the left: E_h decays rapidly and R_h is eventually a good approximation of E_h. On the right: F_h evolves in a manner that is consistent with an asymptotic expansion of Type III with $(p, q) = (2, 4)$.

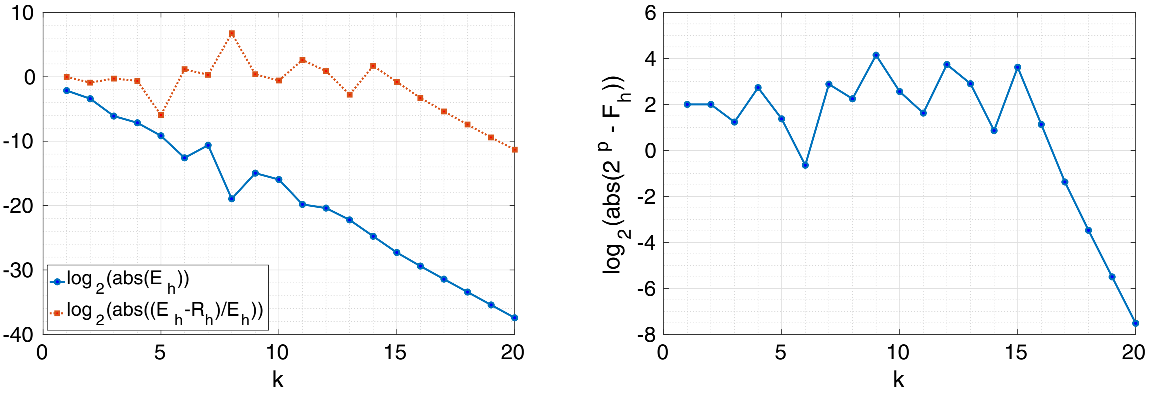

FIGURE 12 | The figures generated by the function `rint_wf23n10`. On the left: E_h decays rapidly, but R_h is not a fair approximation of E_h unless k is large. On the right: The asymptotic range only covers $k \in \{17, 18, 19, 20\}$.

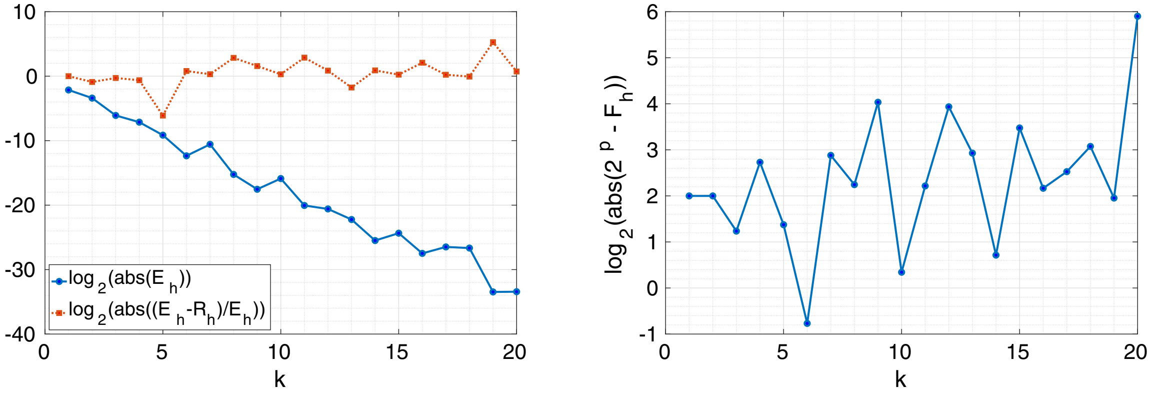

FIGURE 13 | The figures generated by the function `rint_wf23n15`. On the left: E_h decays rapidly, but R_h is never a good approximation of E_h. On the right: All evidence of an asymptotic range has vanished.

We observe that the inclusion of the extra terms has delayed the onset of the asymptotic range to $k = 17$, and while it continues until $k = 20$, we have to accept the fact that the price for obtaining reliable error estimates has been increased substantially compared with the case of $n = 5$. What is the real issue? As we increase the number of terms, we dramatically increase the size of the coefficients that define the error expansion. We therefore need a substantially smaller step size before we can ignore the higher-order terms. It follows that the onset of the asymptotic range is delayed.

The function `rint_wf23n15` considers the integral of the first $n = 15$ terms. It computes A_h for $h = h_k = 2^{1-k}$ and $k \in \{1, 2, \dots, 20\}$, prints a table of relevant numbers on the screen, and generates Figure 13.

We observe that the error E_h continues to decay, but Richardson's error estimate R_h is a poor approximation of E_h. Moreover, the behavior of Richardson's fraction is irregular, and we cannot recognize any asymptotic range. This is an example of a function that is of class C^∞, but so ill-behaved that we cannot reliably estimate the error for the trapezoidal rule using Richardson's techniques.

3.3 | Experiments Based on Numerical Solution of Differential Equations

We begin by considering the problem of simulating the motion of a system of atoms moving in a force field subject to a set of constraints. In this case, Newton's 2nd law takes the form of the following system of differential algebraic equations

$$q'(t) = v(t) \tag{90}$$

$$M v'(t) = f(q(t)) - G(q(t))^T \lambda(t) \tag{91}$$

$$g(q(t)) = 0 \tag{92}$$

The vector q represents the position of the atoms. The vector v represents the velocities of the atoms. The function f represents the force acting on the atoms. The non-singular diagonal matrix M lists the masses of the atoms. The function G is the Jacobian of the constraint function g, and λ is a vector of Lagrange multipliers. In the field of molecular dynamics, the standard algorithm for this problem is the SHAKE algorithm [8]. It uses a pair of staggered grids with uniform step size h and takes the form

$$v_{n+1/2} = v_{n-1/2} + hM^{-1}\left(f(q_n) - G(q_n)^T \lambda_n\right) \tag{93}$$

$$q_{n+1} = q_n + hv_{n+1/2} \tag{94}$$

$$g(q_{n+1}) = 0 \tag{95}$$

The constraint equation (95) is usually a non-linear equation with respect to the vector λ_n of Lagrange multipliers.

3.3.1 | Molecular Dynamics Using GROMACS

This experiment was included in the original paper [3]. We utilized GROMACS v2021 to conduct experiments on the behavior of hen egg white lysozyme submerged in water within a cubic simulation box, following Justin Lemkul's Lysozyme in Water GROMACS Tutorial [9]. Several steps were taken to prepare the system for production simulation: First, ions were introduced to achieve electrical neutrality. Subsequently, energy minimization was performed using the steepest descent algorithm until the maximum force reached below 1000.0 kJ/(molůnm). Following this, the system underwent 100 ps of equilibration in an NVT ensemble to stabilize temperature, followed by another 100 ps of equilibration in an NPT ensemble to stabilize pressure. The described process was replicated using two different force fields, OPLS-AA/L and CHARMM36. We conducted production simulations of 1 ps for both force fields, using

$$n \in \{250, 500, 1000, 1100 : 100 : 2000, 3000 : 1000 : 16000\} \tag{96}$$

steps to cover this interval. Moreover, we used two different values of the tolerance τ for the SHAKE algorithm, namely $\tau \in \{10^{-4}, 10^{-12}\}$. For each experiment, we computed the total kinetic and potential energy of the system at the end of the simulation. The function `gromacs_figures` will generate Figures 14 and 15 for the OPLS-AA/L force field and similar figures for CHARMM36.

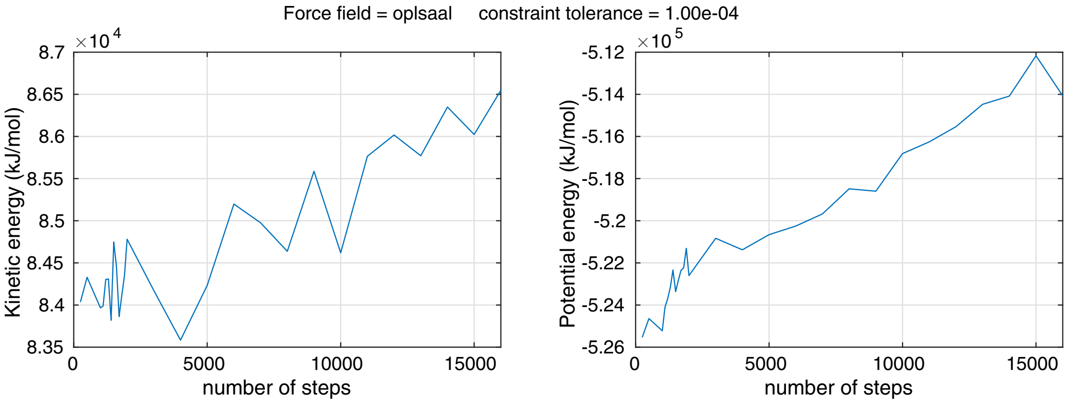

FIGURE 14 | The evolution of the kinetic and potential energy of a system as the number of time steps used to cover 1 ps of real time. On the left: The total kinetic energy. On the right: The total potential energy. In each case, it is the rapid changes in the energy (when the number of time steps is small) that are noteworthy.

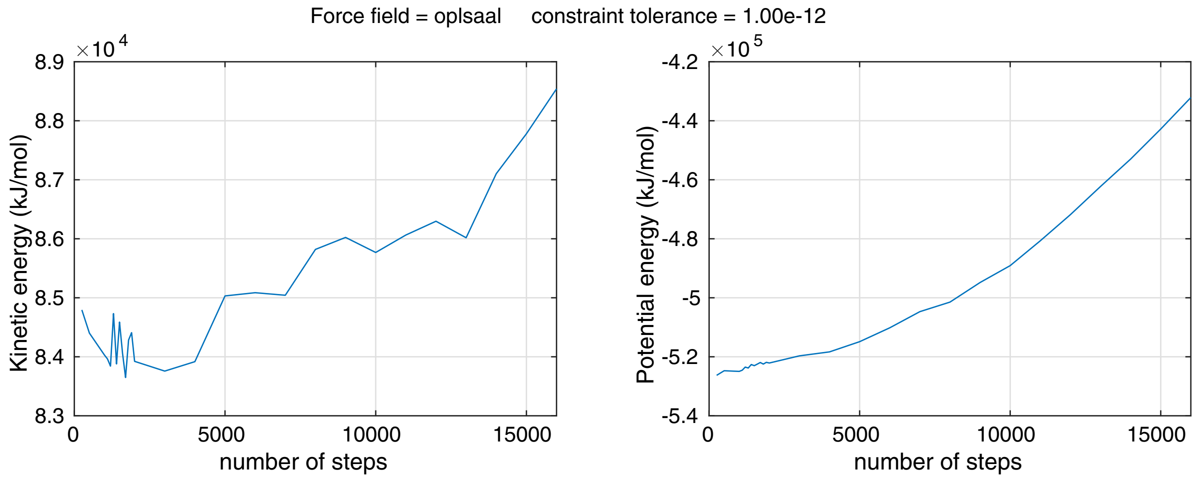

FIGURE 15 | The evolution of the kinetic and potential energy of a system as the number of time steps used to cover 1 ps of real time. On the left: The total kinetic energy. On the right: The total potential energy. The potential energy shows very minor fluctuations, but we continue to see rapid changes in the total kinetic energy.

These figures display the total potential and kinetic energy at the end of the simulation as a function of the total number n of time steps used to cover the interval. The figures present several features of interest. Firstly, the potential energy and especially the kinetic energy exhibit violent oscillations when the tolerance is large, that is, $\tau = 10^{-4}$. The amplitude of the oscillations is reduced when $\tau = 10^{-12}$. We expect the solution of the underlying differential algebraic equation to behave nicely, but we do not expect the computed approximation to follow suit unless τ is very small. Secondly, the total energy grows linearly with the number of time steps. This is not surprising as we expect the rounding error to grow with the number of operations. Thirdly, if the computed energies for tol $= 10^{-12}$ followed an asymptotic expansion of Type III or even Type II, then the commonly used time step of 1 fs ($n = 1000$ in this case) is *not* well inside the asymptotic range. Why is this? If we were in the asymptotic range, then $\hat{A}_h \approx T - \alpha h^p$ would be a good approximation for some $\alpha \neq 0$ and $p > 0$. In particular, the value of $\hat{A}_h$ should behave in a *monotone* manner, and the tiny oscillations that we have recorded should not be present. This does not mean that the simulation is wrong, but we do not have the ability to assert that rounding errors are irrelevant, and we cannot estimate the discretization error using Richardson extrapolation.

We cannot say with certainty why we cannot apply Richardson's techniques in this setting, but we can point to two necessary conditions that are not necessarily satisfied when conducting a GROMACS simulation. Specifically, the tolerance τ might not be small enough, and the functions used to model the force-fields might not be sufficiently differentiable. The next two sections present experiments that demonstrate that each condition is necessary for the success of Richardson's techniques.

3.3.2 | Simulating an Ion-Trap

This is an extended version of the experiment conducted in the original paper. Here, we consider the smallest degree of smoothness that would allow us to apply Richardson's techniques. In molecular dynamics, it is common to ignore the interaction between atoms that are far away. This can be done by setting force-fields to zero outside a sufficiently large ball centered at the source. There is more than one way to achieve this, and the online documentation for non-bonded interactions in GROMACS 2024 [10] discusses the use of force-fields that have jump-discontinuities at the cut-off or force-fields that are of class C^0 or C^1. The GROMACS team finds that their switching function can produce artificially large forces in the switching region, which is why they do not recommend switching Coulomb interactions using their switching functions. However, they find that switching the Lennard-Jones interactions using their switching functions produces acceptable results.

It is obvious that changing the switching functions changes the dynamics, but can these changes be quantified? Can we assert that rounding errors are still irrelevant, and can we estimate the discretization error so that changes to the model can be evaluated?

To explore the importance of smoothness, we have simulated the motion of a set of identical ions moving in a liquid. The ions repel each other electrostatically, but they are pulled towards the origin by independent and identical springs that obey Hooke's law. The friction between each ion and the liquid is proportional to its velocity. The friction drains the kinetic energy and ensures that the ions eventually come to rest in a stable configuration. Let $\boldsymbol{f}$ denote the force-field generated by an ion located at the origin with charge q. Then

$$\boldsymbol{f}(\boldsymbol{r}) = cq\boldsymbol{r}/r^3, \quad r = \|\boldsymbol{r}\|_2 \tag{97}$$

where $c > 0$ is a suitable constant.

A sequence of MATLAB functions explores the use of switching functions that are either discontinuous or of class C^n for modest values of n. Each experiment starts with the same initial condition, and the target value T is always the total kinetic energy after 5 s of real time. Each function tracks the evolution of Richardson's fraction as a function of the time step

$h_k = 2^{1-k}h_1$ where $h_1 = 5 \times 2^{-9}s$ using Runge–Kutta methods of order $p \in \{1, 2, 3, 4\}$.

The modified force-fields all have the form

$$\boldsymbol{f}_s(\boldsymbol{r}) = \boldsymbol{f}(\boldsymbol{r})s(\|\boldsymbol{r}\|_2) \tag{98}$$

where s is a scalar switching function that assumes values in $[0, 1]$.

The script `iontrap_no_cut` does not modify the electrostatic force fields, and the $m = 4$ ions will eventually form a regular tetrahedron with edge length $\rho > 0$. The evolution of Richardson's fractions suggests that the discretization error for each method has an asymptotic expansion of Type III, see Figure 16.

The function `iontrap_jump` uses a switching function g that has a jump discontinuity and satisfies $g(r) = 1$ for $r < 0.5\rho$ and $g(r) = 0$ for $r \geq 0.5\rho$. It generates Figure 17.

In each case, the evolution of Richardson's fraction is irregular, and we are reminded of the Type I expansion presented in Section 3.1.1. We find no evidence of an asymptotic expansion of Type II or Type III. In particular, there is no reason to trust Richardson's error estimate.

The MATLAB functions `iontrap_Ck` where $k \in \{0, 1, 2, 3, 4\}$ use switching functions g that are of class C^k. The switching functions are given by

$$g(r) = \psi_k\left(\frac{b - x}{b - a}\right), \quad a = 0.5\rho, \quad b = 0.95\rho \tag{99}$$

where $\psi_k \in C^k$ is given by Lemma 3.

We shall examine the outcome of these experiments one by one until a pattern emerges. The function `iontrap_C0` uses a switching function of class C^0 and generates Figure 18.

The evolution of Richardson's fraction is consistent with an asymptotic expansion of Type III for $p = 1$, Type II for $p = 2$, and Type I for $p \in \{3, 4\}$. In particular, we can identify an asymptotic range only for $p = 1$.

The function `iontrap_C1` uses a switching function of class C^1 and generates Figure 19.

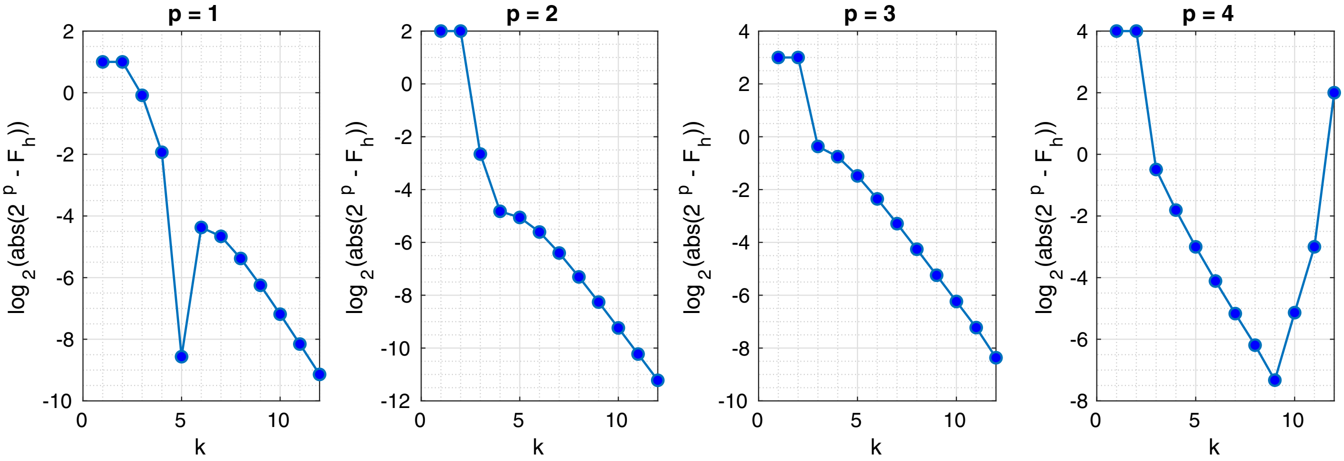

FIGURE 16 | The evolution of F_h for the total kinetic energy at a fixed time for a system of ions computed using Runge–Kutta methods of order $p \in \{1, 2, 3, 4\}$ and Coulomb potentials with infinite range.

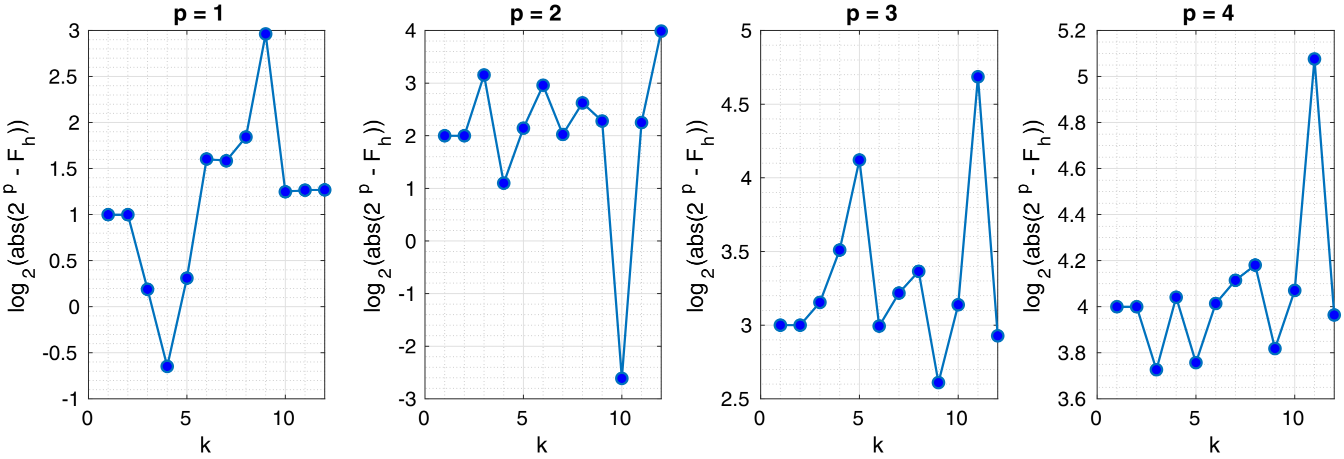

FIGURE 17 | The evolution of F_h for an iontrap when the switching function has a jump discontinuity. We use 4 different Runge–Kutta methods of order $p \in \{1, 2, 3, 4\}$. There is not a single method that shows any evidence of an asymptotic expansion of Type III or even Type II.

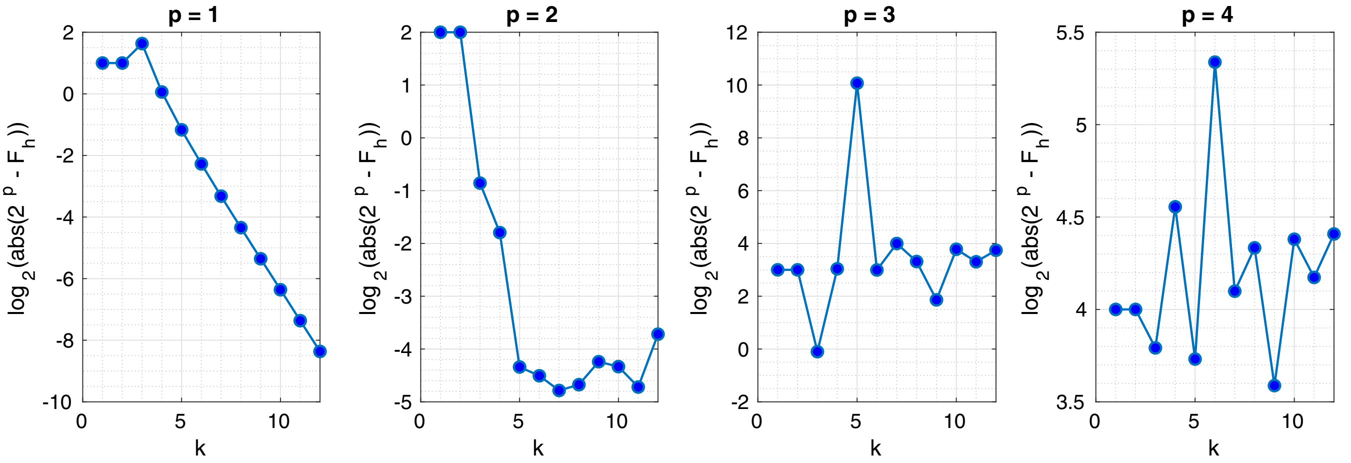

FIGURE 18 | The evolution of F_h for an iontrap where the switching function is of class C^0 using 4 different Runge–Kutta methods of order $p \in \{1, 2, 3, 4\}$. The evolution of F_h is consistent with a Type III expansion for $p = 1$ and with a Type II expansion for $p = 2$.

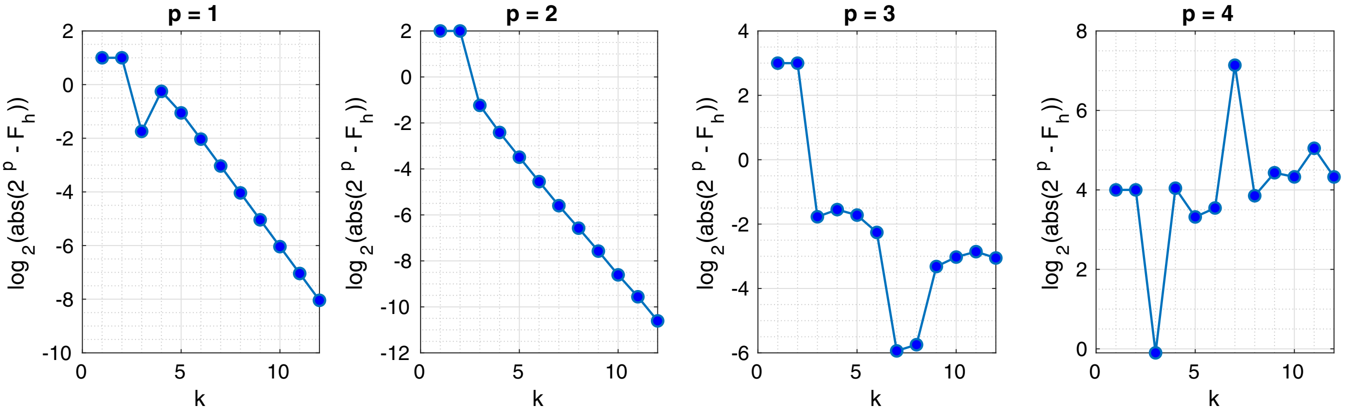

FIGURE 19 | The evolution of F_h for an iontrap where the switching function is of class C^1 using 4 different Runge–Kutta methods of order $p \in \{1, 2, 3, 4\}$. The evolution of F_h is consistent with a Type III expansion for $p \in \{1, 2\}$ and a Type II expansion for $p = 3$.

The evolution of Richardson's fraction is consistent with an asymptotic expansion of Type III for $p \in \{1, 2\}$, Type II for $p = 3$, and Type I for $p = 4$. In particular, we can now identify an asymptotic range for both $p = 1$ and $p = 2$.

The general pattern we observe is summarized in Table 1. We note that the evolution of Richardson's fraction for a method of order p is consistent with an error expansion of Type III, when the switching function is at least of class C^{p-1}. We find that this is a very modest requirement.

3.3.3 | Computing the Range of a Howitzer

This experiment is a variation of an experiment included in the original paper [3]. There, we considered the problem of identifying the type of shell fired by a specific howitzer by computing the maximum range of a shot for a selection of shells with different drag coefficients. Here, we consider the problem of computing the range of a single shell. Superficially, this is a simpler problem, but here we exhibit an issue associated with the atmospheric model that was irrelevant for the original paper.

We continue to treat the shell as a point mass moving in a plane. The position of the shell is $\boldsymbol{r} = (x_1, x_2)$ and the velocity is $\boldsymbol{v} = (x_3, x_4)$. The muzzle of the howitzer is located at $(0, 0)$. The positive x_1 axis is horizontal and directed towards the target on the firing range, and the positive x_2 axis is directed upwards. Sea-level corresponds to $x_2 = 0$. The gravity field is constant and directed downwards. Newton's equation of motion takes the form

$$\boldsymbol{r}'(t) = \boldsymbol{v}(t) \tag{100}$$

$$m\boldsymbol{v}'(t) = -\frac{1}{2} A \rho c_D(\nu) \|\boldsymbol{v}\|_2 \boldsymbol{v}(t) + m\boldsymbol{g} \tag{101}$$

where m is the mass of the shell, A is the area of the cross section of the shell, ρ is the density of the atmosphere, c_D is the drag coefficient of the shell, a is the speed of sound, $\nu = \|\boldsymbol{v}\|_2 / a$ is the shell's Mach number and $\boldsymbol{g} = (0, -g)$ is the acceleration due to gravity. The density ρ as well as the speed of sound a are functions of the shell's height x_2 above sea-level. These values will be computed using the international standard atmospheric model, which is implemented in MATLAB as `atmosisa`. The initial condition is

$$\boldsymbol{x}(0) = (x_1(0), x_2(0), x_3(0), x_4(0)) = (0, 0, v_0 \cos(\theta), v_0 \sin(\theta)) \tag{102}$$

where $v_0 = \|\boldsymbol{v}(0)\|_2$ is the muzzle velocity of the shell and θ is the elevation of the barrel of the howitzer. We compute the range of

TABLE 1 | Summary of the different switching functions used by the MATLAB functions used to simulate an iontrap, as well as our classification of the evolution of Richardson's fraction for the total kinetic energy at the end of each simulation using Runge–Kutta methods of order $p \in \{1, 2, 3, 4\}$.

		Classification of the convergence			
MATLAB function	**Switching function**	**RK1**	**RK2**	**RK3**	**RK4**
`iontrap_no_cut`	no switching function	III	III	III	III
`iontrap_jump`	jump discontinuity	I	I	I	I
`iontrap_C0`	C^0	III	II	I	I
`iontrap_C1`	C^1	III	III	II	I
`iontrap_C2`	C^2	III	III	III	II
`iontrap_C3`	C^3	III	III	III	III
`iontrap_C4`	C^4	III	III	III	III
`iontrap_Cinf`	C^∞	III	III	III	III

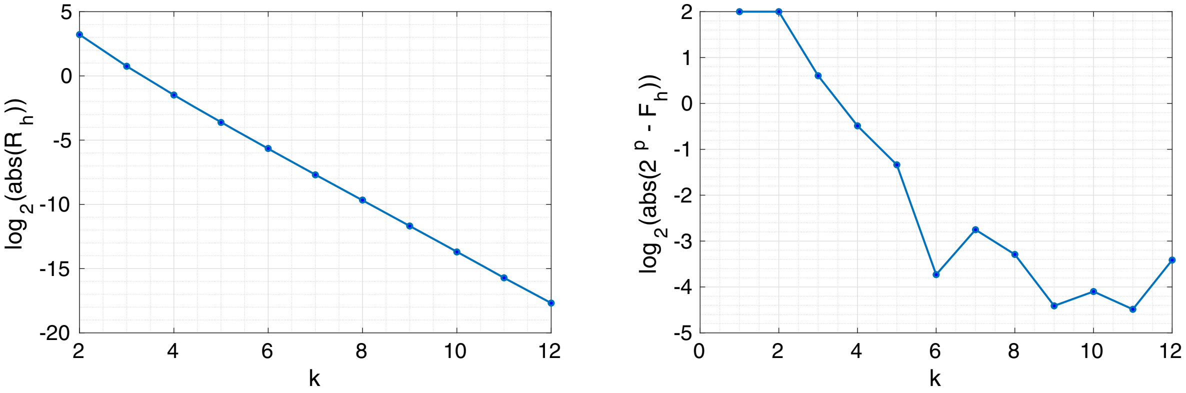

FIGURE 20 | The figures generated by the function `range_mwe1`. On the left: Richardson's error estimate decays rapidly. On the right: The behavior of Richardson's fraction is at best reminiscent of a Type II expansion.

a shot by solving a non-linear equation

$$g(\boldsymbol{x}(t)) = 0 \tag{103}$$

with respect to t. The event function g is given by

$$g(\boldsymbol{x}) = x_2 \tag{104}$$

We do this by carefully adjusting the final time step to place the shell directly on the ground. We will be using a one-step method $\boldsymbol{y}_{k+1} = \boldsymbol{\phi}(\boldsymbol{y}_k, h)$ to compute approximations $\boldsymbol{y}_k$ of $\boldsymbol{x}(t_k)$. If $g(\boldsymbol{y}_k)$ and $g(\boldsymbol{y}_{k+1})$ have different signs, then we use the bisection method to solve the non-linear equation

$$g(\boldsymbol{\phi}(\boldsymbol{y}_k, \rho h)) = 0 \tag{105}$$

with respect to $\rho \in (0, 1)$. We approximate the range of the shell using the first component of $\boldsymbol{\phi}(\boldsymbol{y}_k, \rho h)$, that is,

$$A_h = \phi_1(\boldsymbol{y}_k, \rho h) \tag{106}$$

The corresponding flight time from the muzzle of the gun to the point of impact is $t(\rho) = t_k + \rho h$. We terminate the bisection algorithm when the current search bracket (a, b) satisfies $|a - b| < ut_k/h$, where u is the unit roundoff and h is the current time step. Why is this reasonable? If $\rho_i \in (a, b)$, then $t(\rho_1) - t(\rho_2) = (\rho_1 - \rho_2)h$ satisfy $|t(\rho_1) - t(\rho_2)| < ut_k$ and further iterations are unlikely to reduce the error as the spacing between floating point numbers is $O(ut_k)$ in the vicinity of t_k.

In this paper, we use a model of Rheinmetall's Panzerhaubitze 2000 L52 155 mm howitzer that fires shells with a Type G7 drag function [11]. Our model is implemented in `pzh2000` using parameters extracted from a variety of publicly available sources. We use Heun's method to integrate the equations of motion. The elevation is fixed at $\theta = \frac{\pi}{3}$ radians, that is, 60°.

The function `range_mwe1` uses cubic spline interpolation to approximate the drag coefficient. It tracks the evolution of Richardson's fraction for $h = h_k = 2^{2-k}s$ and $k \in \{1, 2, \dots, 12\}$ and generates Figure 20.

To our surprise, we found no evidence of an asymptotic error expansion of Type III, and we had no reason to trust the error estimate.

To eliminate the question of differentiability completely, we interpolated the drag coefficient using a function of class C^∞. The function `range_mwe2` computes the corresponding range A_h for $h = h_k = 2^{2-k}s$ and $k \in \{1, 2, \dots, 12\}$, prints a table on the screen and generates Figure 21.

Again, there is no evidence of an asymptotic expansion of Type III and no reason to trust the error estimate.

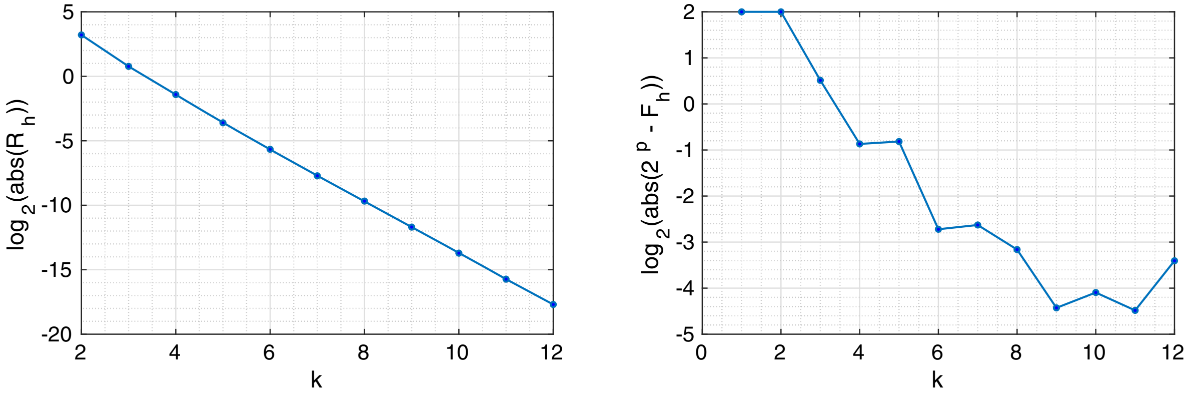

FIGURE 21 | The figures generated by the function `range_mwe2`. On the left: Richardson's error estimate decays rapidly. On the right: There is still no evidence of an asymptotic expansion of Type III.

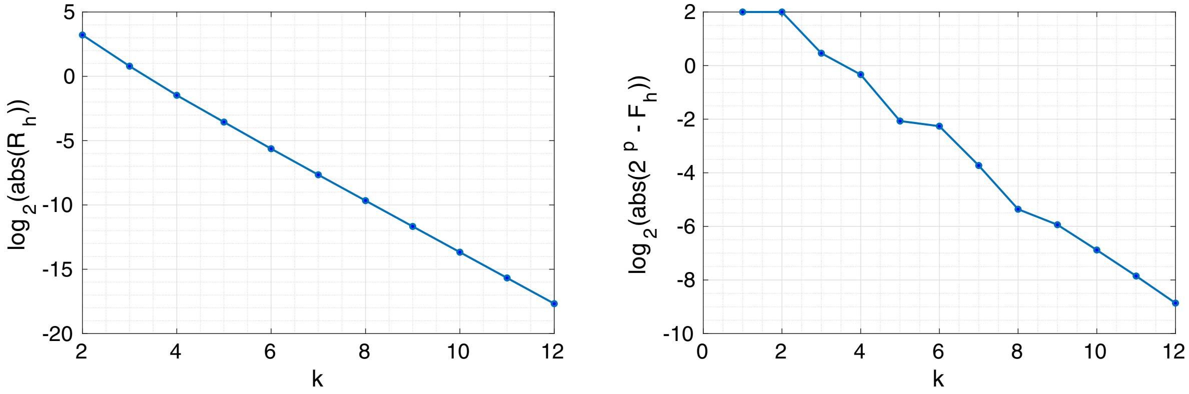

FIGURE 22 | The figures generated by the function `range_mwe3`. On the left: Richardson's error estimate continues to decay rapidly. On the right: We can finally identify an asymptotic range.

The problem was eventually traced to our use of the international standard atmospheric model. This model divides the atmosphere into layers. The first layer stretches from sea-level to a height of 11,000 meters. The density and the speed of sound are both continuous functions of the height above sea-level, but they are not differentiable at the boundary between the first and the second layer, and this irregularity is enough to destroy our ability to estimate the discretization error. In the original paper, we simulated shots with a lower elevation, and the shells did not cross into the second layer.

How do we circumvent this problem? Following Gear and Østerby [12], we adjust the time steps as needed to place the shell directly on the boundary between the two layers, that is, we change the event function to

$$g(x) = x_2(11000 - x_2) \tag{107}$$

and find the first three solutions of $g(x) = 0$. In general, all but three time steps will have the same size h.

The function `range_mwe3` uses a smooth drag coefficient and solves the extended event equation accurately. It tracks the evolution of Richardson's fraction for $h = h_k = 2^{2-k}s$ and $k \in \{1, 2, \dots, 12\}$, prints a table on the screen and generates Figure 22.

We now find evidence of an asymptotic expansion of Type III with $(p, q) = (2, 3)$, and we are confident that we can estimate the error.

We found it reasonable to assume that a smooth drag coefficient was overkill, and so we returned to an interpolant of class C^2. The function `range_mwe4` uses cubic spline interpolation to approximate the drag coefficient, but continues to solve the extended event equation accurately. It tracks the evolution of Richardson's fraction for $h = h_k = 2^{2-k}s$ and $k \in \{1, 2, \dots, 12\}$, prints a table on the screen and generates Figure 23.

We continue to find evidence of an error expansion of Type III with $(p, q) = (2, 3)$, and we remain confident that we can estimate the error accurately.

We observe that the size of the error estimate $R_h \approx 2^{-10}$ meters as we enter the asymptotic range at $k = 8$ is significantly smaller than the kill-radius (about 50 meters) of a 155 mm high explosive shell.

4 | Related Work in Computational Fluid Dynamics

Richardson extrapolation is used extensively in the field of computational fluid dynamics. In this field, the term "grid

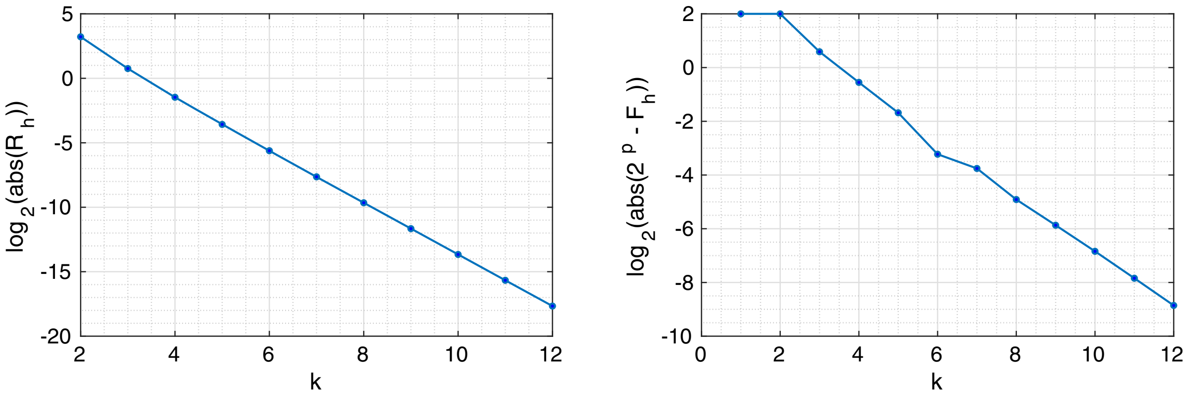

FIGURE 23 | The figures generated by the function `range_mwe4`. On the left: Richardson's error estimate decays rapidly. On the right: We can still identify an asymptotic range, that is, $k \in \{8, 9, 10, 11, 12\}$.

convergence study" is used to describe the process of finding suitable grids and estimating the target value T from A_h. Professor Patrick J. Roache has made a major effort to establish guidelines for such studies, and his textbook on verification and validation in computational fluid dynamics remains a cornerstone of this field [13]. In this section, we shall relate the work of Roache to the results presented in this paper.

Roache operates with a minimum of three approximations f_1, f_2 and f_3 where f_1 corresponds to the finest grid, f_2 corresponds to a coarser grid and f_3 corresponds to the coarsest grid. The spacing between the nodes on the ith grid is denoted h_i. Roache [13] (page 114, Equation (5.6.1)) defines a grid convergence index (GCI) as follows

$$\mathrm{GCI}_{12} = F_S \frac{|f_1 - f_2|}{r^p - 1} \tag{108}$$

where $r = h_2/h_1$ is the grid refinement ratio and F_S is a factor of safety that is carefully chosen to increase the probability that T is contained in the interval I given by

$$I = (f_1 - \mathrm{GCI}_{12}, f_1 + \mathrm{GCI}_{12}) \tag{109}$$

Roache [13] (Section 5.9.2) recommends the choice of $F_S = 3$ when the study is based on just two grids and $F_S = 1.25$ when the study is based on at least three grids. Roache [13] (page 134, Equation (5.10.5.2)) states that if r is constant, then

$$\mathrm{GCI}_{12} \approx r^p \mathrm{GCI}_{23} \tag{110}$$

indicates that the asymptotic range has been achieved.

Roache has greater flexibility than we, as we have fixed $r = 2$ and $(f_1, f_2, f_3) = (A_h, A_{2h}, A_{4h})$. In our setting, we find that the grid convergence index reduces to

$$\mathrm{GCI}_{12} = F_S |R_h| \tag{111}$$

and Equation (110) reduces to the statement that

$$F_h \approx 2^p \tag{112}$$

To what extent does our analysis support the experience and recommendations of Roache? If E_h has an asymptotic expansion of Type III, then we find that F_h approaching 2^p strongly suggests that R_h is an increasingly good approximation of E_h. Why so? By Theorem 2 we have

$$\frac{(E_h - R_h)/E_h}{2^p - F_h} \rightarrow \frac{1}{2^q - 1}, \quad h \rightarrow 0_+ \tag{113}$$

which implies that the approximation

$$(E_h - R_h)/E_h \approx \frac{1}{2^q - 1}(2^p - F_h) \tag{114}$$

will eventually be good. However, if the asymptotic expansion is only of Type I or Type II, then we cannot issue any definitive statements on this matter.

If the error expansion is of Type III, then we can use Theorem 4 to evaluate the interval recommended by Roache, that is,

$$I = (A_h - F_S|R_h|, A_h + F_S|R_h|) \tag{115}$$

If $\beta/\alpha > 0$, then we can say with certainty that $T \in I$ when h is sufficiently small, because we know that T is eventually bracketed by A_h and $A_h + R_h$. It then follows that

$$A_h - F_S|R_h| < T < A_h + F_S|R_h| \tag{116}$$

for any value of the safety factor $F_S \geq 1$. If $\beta/\alpha < 0$ and $R_h < 0$, then $T < A_h + R_h < A_h$ and T cannot lie in the upper half of I. If $\beta/\alpha < 0$ and $R_h > 0$, then $A_h + R_h < T$ and T cannot lie in the lower half of I. If the error expansion is not of Type III, then we cannot issue any definite statements on this matter.

5 | Practical Implications and Future Directions

Our purpose is to develop algorithms and software that will allow other scientists to model the physical world. Their goal is to refine their models and improve their predictions. Let P denote the value of a parameter that can be measured in the real world, and let T denote the corresponding value obtained from the model. Can the modeling error $P - T$ be computed? We cannot compute the exact value of T, but we often implement software that returns approximations A_h that depend on a single real parameter $h > 0$, say, the time step used to integrate a system of differential equations. A host of computational errors ensures that the computed value of $\hat{A}_h$ is different from the exact value A_h. We now write the modeling error as a sum of three terms, that is,

$$P - T = (P - \hat{A}_h) - (A_h - \hat{A}_h) - (T - A_h) \tag{117}$$

and consider the user's problem of reducing $|P - T|$ below a given threshold. The user is not certain $P - \hat{A}_h$ dominates the right-hand side, because the calculation is complex and the model has been refined several times since it was first introduced. What can we do to help? In this paper, our primary contribution has been to explain the value of having an expansion with three terms, that is,

$$E_h = \alpha h^p + \beta h^q + o(h^q) \tag{118}$$

We have shown how to use Richardson's fraction F_h to identify the asymptotic range where computational errors are irrelevant and where Richardson's error estimate R_h is a good approximation of the error E_h. Our most valuable contribution is the realization that

$$(E_h - R_h)/E_h \approx \frac{1}{2^q - 1}(2^p - F_h) \tag{119}$$

is eventually a good approximation. This observation is useful because it allows us to estimate the accuracy of Richardson's error estimate using values that are readily available, but there is another implication that is potentially far more important. In the past, high-order methods with $q = p + 1$ have been developed. Might we not use the order conditions differently and deliberately choose $p = 1$ and q as large as possible? It seems plausible that such a method would have a very large asymptotic range that would start almost immediately and while Richardson's error estimates would be large, they would also be very accurate, because $2^p - F_h = o(h^{q-p})$ so that $(2^p - F_h)/(2^q - 1)$ would decay rapidly. While the implicit trapezoidal rule for solving ordinary differential equations has $(p, q) = (2, 4)$, it remains to determine if methods with $q = p + k$ can be developed for any positive integer k.

The numerical experiments included in this paper contain several features of interest.

Our use of Weierstrass's function demonstrates that there are C^∞ functions that are so ill-behaved that we cannot use Richardson's techniques to estimate the discretization error even in the context of very simple calculations such as numerical differentiation and integration, see Sections 3.1 and 3.2.7. The ion trap discussed in Section 3.3.2 reveals the price of using switching functions that are not many times differentiable. If the switching functions are not continuous, then we have no hope of estimating the discretization error. If the switching functions are of class C^0, then we can only estimate the discretization for Euler's explicit method. If we wish to use Heun's method, then we must use switching functions of class C^1 or better. In general, the pattern is very clear. If the switching functions are not sufficiently smooth compared with the order of our method, then we lose the ability to perform error estimation. Our calculations might still be accurate, but we cannot make this determination.

The howitzer discussed in Section 3.3.3 reveals a practical problem. We were able to obtain reliable error estimates using Heun's method and cubic spline interpolation as soon as we respected the discontinuities included in the atmospheric model. Once we entered the asymptotic range, we found error estimates that were far smaller than the kill-radius of the shell. In the context of external ballistics, we do not need tiny error estimates; we merely need accurate error estimates that are comparable to the kill-radius of the shell. While many of the details are classified, it is known that professional ballisticians have previously used a Runge–Kutta–Fehlberg method of order 7 with a fixed time step, and they interpolated the drag coefficient using either linear interpolation or piece-wise cubic Hermite interpolation [14, 15]. We have no reason to doubt that their results are accurate, but in view of our experiments, we find it doubtful that they achieve 7th-order accuracy. Moreover, we have no reason to believe that Richardson's error estimate is reliable because the solution of their system of differential equations is of class at most C^3 and because we need to adjust the time step to place the shell exactly on the boundary between the different layers of the atmosphere. We conjecture that equally useful results with reliable error estimates can be obtained at lower cost using a method of low order. It remains to implement the 6 DOF and 7 DOF models specified by NATO's standardization recommendation STANREC 4618 [16] and determine the validity of this conjecture.

When validating software for solving differential equations, it is common to use the method of manufactured solutions. A solution is chosen, and the corresponding inhomogeneous term is derived and fed to the solver. If the solver returns a good approximation of the known solution within the available time, then the software is regarded as good. When developing the software, we often use every trick in the book to improve the parallel performance and improve the overall flop rate. We might discard small terms, skip the last few iterations, or use linear interpolation instead of a more complex scheme. Our deliberate use of single precision to accelerate a calculation at the expense of a reliable error estimate, see Section 3.2.5, demonstrates the dangers of too much optimization. We may well retain the ability to compute a good approximation of a *manufactured* solution, but do we still have the extra bits that are required to reliably estimate the accuracy when the solution is not known?

Acknowledgments

We are pleased to acknowledge the work done by the reviewers of our original paper [3]. Their questions and suggestions helped us refine our analysis and extend our work. We are happy to acknowledge the work done by the reviewers of this extended paper. Their comments allowed us to improve the presentation. This work was supported by eSSENCE, the Swedish e-Science collaboration. This work was also supported by MCIN/AEI/10.13039/501100011033 (grant PID2022-136454NB-C22), and by Government of Aragon (T58_23R research group).

Disclosure

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The software needed to repeat every experiment and generate every figure in this paper from scratch is available for anonymous download from our GitHub repository: https://github.com/spockcc/ccpe2025-richardson.

The names of relevant MATLAB functions are written using a typewriter font, for example, `rint_wf23`. Most functions will print relevant tables on the screen, and the volume of data far exceeds that which can be included in this paper. We therefore ask the reader to execute the experiments and verify the few statements that refer to these tables. The repository includes `.mat` files with the outcome of large experiments. Should these files be deleted, then the first call to a function will regenerate the `.mat` file and the second call will produce the tables or figures.

References

1. L. F. Richardson and G. J. A. Gaunt VIII, “The Deferred Approach to the Limit,” *Philosophical Transactions of the Royal Society A* 226, no. 636–646 (1927): 299–361, https://doi.org/10.1098/rsta.1927.0008.

2. Z. Zlatev, I. Dimov, I. Faragó, and Á. Havasi, *Richardson Extrapolation: Practical Aspects and Applications* (De Gruyter, 2018).

3. C. C. Kjelgaard Mikkelsen and L. López-Villellas, “The Need for Accuracy and Smoothness in Numerical Simulations,” in *Parallel Processing and Applied MathematicsUmeå University*, ed. R. Wyrzykowski, J. Dongarra, E. Deelman, and K. Karczewski (Springer Nature Switzerland, 2025), 3–16.

4. A. K. Alekseev, A. E. Bondarev, and A. E. Kuvshinnikov, “A Comparison of the Richardson Extrapolation and the Approximation Error Estimation on the Ensemble of Numerical Solutions,” in *Computational Science – ICCS 2021Keldysh Institute of Applied Mathematics, RAS, Moscow, Russia*, ed. M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot (Springer International Publishing, 2021), 554–566.

5. E. Carson and N. J. Higham, “Accelerating the Solution of Linear Systems by Iterative Refinement in Three Precisions,” *SIAM Journal on Scientific Computing* 40, no. 2 (2018): A817–A847, https://doi.org/10.1137/17M1140819.

6. C. T. Kelley, “Newton’s Method in Mixed Precision,” *SIAM Review* 64, no. 1 (2022): 191–211, https://doi.org/10.1137/20M1342902.

7. C. C. Kjelgaard Mikkelsen, L. L pez-Villellas, and P. Garc a-Risue o, “Newton’s Method Revisited: How Accurate Do We Have to Be?,” *Concurrency and Computation: Practice and Experience* 36, no. 10 (2023): e7853, https://doi.org/10.1002/cpe.7853.

8. J. P. Ryckaert, G. Ciccotti, and H. J. Berendsen, “Numerical Integration of the Cartesian Equations of Motion of a System With Constraints: Molecular Dynamics of n-Alkanes,” *Journal of Computational Physics* 23, no. 3 (1977): 327–341.

9. J. A. Lemkul, “From Proteins to Perturbed Hamiltonians: A Suite of Tutorials for the GROMACS-2018 Mol. Sim. Pack, v1.0,” *Journal of Computational Molecular Science* 1, no. 1 (2019): 5068.

10. https://manual.gromacs.org/2024.0/reference-manual/functions/nonbonded-interactions.html.

11. “Original source: Ballistics Research Laboratory, Aberdeen Proving Ground,” https://jbmballistics.com/ballistics/downloads/downloads.shtml.

12. C. W. Gear and O. Østerby, “Solving Ordinary Differential Equations With Discontinuities,” *ACM Transactions on Mathematical Software* 10, no. 1 (1984): 23–44.

13. P. J. Roache, *Verification and Validation* (Hermosa Publishers, 2009).

14. P. Wey, D. Corriveau, T. A. Saitz, R. dW, and P. Strömbäck, “BALCO 6/7-DoF Trajectory Model,” https://www.researchgate.net/publication/311858619_BALCO_67-DoF_Trajectory_Model.

15. C. A. Rabbath and D. Corriveau, “A Comparison of Piecewise Cubic Hermite Interpolating Polynomials, Cubic Splines and Piecewise Linear Functions for the Approximation of Projectile Aerodynamics,” *Defence Technology* 15, no. 5 (2019): 741–757.

16. “The Six/Seven Degrees of Freedom Guide Projectile Trajectory Model. NATO STANREC 4618,” in *Releasable to Citizens of PFP Countries and NATO Allies Through Their National DODs* (NATO’s Standardization Office (NSO), 2016).

17. K. Weierstrass, *Abhandlungen Aus der Functionenlehre* (Verlag von Julius Springer, 1886), 97–100.

18. G. H. Hardy, “Weierstrass’s Nondifferentiable Function,” *Transactions of the American Mathematical Society* 17, no. 301 (1917): 325.

Appendix A

Auxiliary Results

This section contains the details needed to construct our numerical examples. We include a detailed proof only when we do not know a reference.

Weierstrass’s Function

We shall use Weierstrass’s function to illustrate several aspects of Richardson extrapolation. This section contains the details that are relevant. Let $a \in (0,1)$ and let $b > 1$ and consider the function $f_0 : \mathbb{R} \rightarrow \mathbb{R}$ given by

$$f_0(x) = \sum_{n=0}^{\infty} a^n \cos(b^n \pi x) \tag{A1}$$

Weierstrass [17] showed that this function is everywhere continuous but nowhere differentiable when b is an odd integer such that $ab > 1 + \frac{3}{2}\pi$. Hardy [18] extended Weierstrass’s result to any $b > 0$ and $ab > 1$. In this paper, we shall exclusively use the case of $a = \frac{1}{2}$ and $b = 3$.

The problem of implementing Weierstrass’s function accurately presents at least one feature of interest. In particular, if $\theta \geq 2^{55}$, then the distance between the double precision representation of θ and the next double precision number is at least 8 and therefore greater than 2π. In this case, there is no reason to expect that the cosine function will return an accurate approximation of $y = \cos(\theta)$. We find that MATLAB’s function `cos` returns $\hat{y} = -0.2956$ for $\theta = 2^{55}\pi$ while the exact value is $y = 1$. We shall therefore rely on the fact that if T_k is the kth Chebyshev polynomial of the first kind, then

$$\forall \theta \in \mathbb{R} \;:\; T_k(\cos(\theta)) = \cos(k\theta). \tag{A2}$$

It follows that

$$\cos(b^{n+1}\pi x) = T_b(\cos(b^n \pi x). \tag{A3}$$

In the case of $b = 3$, we have $T_3(x) = 4x^3 - 3x$.

The series for f_0 can be integrated term-wise as many times as we like. This follows from Weierstrass’s M-test using the geometric series to establish absolute and uniform convergence.

Switching Functions

In molecular dynamics, it is common to nullify a force-field $\boldsymbol{r} \rightarrow \boldsymbol{f}(\boldsymbol{r})$ outside a sufficiently large sphere centered at the source. This is typically done using a switching function $s : [0, \infty) \rightarrow [0, 1]$ which satisfies $s(r) = 1$ for $r < a$ and $s(r) = 0$ for $b < r$ where $0 < a < b$. The new force-field $\boldsymbol{f}_s$ is given by

$$\boldsymbol{f}_s(\boldsymbol{r}) = \boldsymbol{f}(\boldsymbol{r})s(r) \tag{A4}$$

There is more than one way to construct switching functions, and in this paper, our only goal is to control the smoothness of the new force-field in a manner that is as simple as possible. In particular, we are not interested in the computational efficiency, and we shall make no effort to minimize the number of arithmetic operations.

Switching Functions of Class C^k

Lemma 3. *Let $k \in \mathbb{N} \cup \{0\}$ and let $\phi_k : \mathbb{R} \to \mathbb{R}$ and $\psi_k : \mathbb{R} \to \mathbb{R}$ be given by*

$$\phi_k(x) = \begin{cases} 0 & x \leq 0, \\ x^{k+1} & x > 0. \end{cases} \tag{A5}$$

and

$$\psi_k(x) = \frac{\phi_k(x)}{\phi_k(x) + \phi_k(1-x)} \tag{A6}$$

Then ϕ_k and ψ_k are of class C^k and ψ_k switches from 0 to 1 in the sense that

1. *$\psi_k(x) = 0$ for $x \leq 0$*
2. *ψ_k is strictly increasing on the interval $(0, 1)$*
3. *$\psi_k(x) = 1$ for $1 \leq x$.*

Proof. It is clear that $\phi_0 \in C^0$ because the statement

$$\forall x \in \mathbb{R} \ : \ \phi_0(x) = \max\{0, x\} = \frac{1}{2}(x + |x|) \tag{A7}$$

exhibits ϕ_0 as a linear combination of continuous functions. We shall now prove that $\phi_k \in C^1$ for all $k \in \mathbb{N}$ with

$$\forall x \in \mathbb{R} \ : \phi'_k(x) = (k+1)\phi_{k-1}(x). \tag{A8}$$

This statement will immediately imply that $\phi_k \in C^k$. It is clear that ϕ_k is differentiable for all $x \neq 0$ and

$$\phi_k(x) = 0 \tag{A9}$$

for $x < 0$ and

$$\phi'_k(x) = (k+1)x^k \tag{A10}$$

for $0 < x$. In either case, we have

$$\phi'_k(x) = (k+1)\phi_{k-1}(x) \tag{A11}$$

It remains to consider the case of $x = 0$. Let therefore $h \neq 0$ be given. It follows

$$\frac{\phi_k(h) - \phi_k(0)}{h} = \begin{cases} 0 & h < 0, \\ h^k & 0 < h. \end{cases} \tag{A12}$$

This implies that ϕ_k is differentiable at $x = 0$ and

$$\phi'_k(0) = 0 = (k+1)\phi_{k-1}(0). \tag{A13}$$

We conclude that $\phi_k \in C^k$. This completes the analysis of ϕ_k. The function ψ_k is defined for all x, simply because

$$\forall x \in \mathbb{R} \ : \ \phi_k(x) + \phi_k(1-x) > 0. \tag{A14}$$

It follows immediately that $\psi_k \in C^k$ and for any $x \in \mathbb{R}$ we have

$$\begin{aligned} \psi_k(x) &= \frac{\phi'_k(x)(\phi_k(x) + \phi_k(1-x)) - \phi_k(x)(\phi'_k(x) - \phi'_k(1-x))}{(\phi_k(x) + \phi_k(1-x))^2} \\ &= \frac{\phi'_k(x)\phi_k(1-x) + \phi_k(x)\phi'_k(1-x)}{(\phi_k(x) + \phi_k(1-x))^2} \end{aligned} \tag{A15}$$

This shows that $\psi'_k(x) > 0$ for $x \in (0, 1)$ because $\phi_k(t) > 0$ and $\phi'_k(t) > 0$ for $t > 0$. We conclude that ψ_k is strictly increasing on $(0, 1)$. For $x \leq 0$ we have $\psi_k(x) = 0$ because $\phi_k(x) = 0$ and for $x \geq 1$ we have $\psi_k(x) = 1$ because $\phi_k(1-x) = 0$. This completes the proof. □

Now consider the problem of switching a function $f \in C^m(\mathbb{R}, \mathbb{R})$ to zero. Let $a < b$ be given and let $g : \mathbb{R} \to \mathbb{R}$ be given by

$$g(x) = f(x)\psi_k\left(\frac{b-x}{b-a}\right) \tag{A16}$$

where ψ_k is defined by Lemma 3. Then g is of class C^p where $p = \min\{m, k\}$ and $g(x) = f(x)$ for $x < a$ and $g(x) = 0$ for $b < x$.

Switching Functions of Class C^∞

Lemma 4. *Let $\phi : \mathbb{R} \to \mathbb{R}$ and $\psi : \mathbb{R} \to \mathbb{R}$ be given by*

$$\phi(x) = \begin{cases} 0 & x \leq 0, \\ \exp(-x^{-1}) & x > 0. \end{cases} \tag{A17}$$

and

$$\psi(x) = \frac{\phi(x)}{\phi(x) + \phi(1-x)}. \tag{A18}$$

Then ϕ and ψ are of class C^∞ and ψ satisfies the following statements:

1. *$\psi(x) = 0$ for $x \leq 0$*
2. *ψ is strictly increasing on the interval $(0, 1)$*
3. *$\psi(x) = 1$ for $1 \leq x$.*

Proof. Let $p_k : \mathbb{R} \to \mathbb{R}$ denote the polynomials given by

$$p_0(t) = 1 \tag{A19}$$

$$p_k(t) = t^2(p_{k-1}(t) - p'_{k-1}(t)), \quad k \in \mathbb{N}, \tag{A20}$$

and let the functions $g_k : \mathbb{R} \to \mathbb{R}$ be given by

$$g_k(x) = \begin{cases} 0 & x \leq 0, \\ p_k(x^{-1})\exp(-x^{-1}) & 0 < x \end{cases} \tag{A21}$$

By definition, $\phi(x) = g_0(x)$ for all x. We shall now prove that g_{k-1} is differentiable for any $k \in \mathbb{N}$ and

$$\forall x \in \mathbb{R} \ : \ g'_{k-1}(x) = g_k(x) \tag{A22}$$

This statement will immediately imply that $\phi \in C^\infty$. Let therefore $k \in \mathbb{N}$ be given. It is clear that g_{k-1} is differentiable for $x \neq 0$ with $g'_{k-1}(x) = 0 = g_k(x)$ for $x < 0$ and

$$\begin{aligned} g'_{k-1}(x) &= x^{-2}\left[p_{k-1}(x^{-1}) - p'_{k-1}(x^{-1})\right]\exp(-x^{-1}) \\ &= p_k(x^{-1})\exp(-x^{-1}) = g_k(x) \end{aligned} \tag{A23}$$

for $0 < x$. It remains to consider the case of $x = 0$. Let $h \neq 0$ be given. It follows that

$$\frac{g_{k-1}(h) - g_{k-1}(0)}{h} = \begin{cases} 0 & h < 0, \\ h^{-1}p_{k-1}(h^{-1})\exp(-h^{-1}) & 0 < h. \end{cases} \tag{A24}$$

We conclude that g_{k-1} is differentiable with $g'_{k-1}(0) = 0$. Here, it is critical that p_{k-1} is a polynomial. This shows that g_{k-1} is differentiable for all x with derivative g_k. It follows that $\phi \in C^\infty$.

We shall now analyze the function ψ. By definition, $\phi(x) \geq 0$ for all x and $\phi(x) > 0$ for all $x > 0$. This ensures that

$$\phi(x) + \phi(1-x) \geq \phi(x) > 0 \tag{A25}$$

for $x > 0$ and

$$\phi(x) + \phi(1-x) \geq \phi(1-x) > 0 \tag{A26}$$

for $0 \leq x$. This shows that ψ is defined for every $x \in \mathbb{R}$. It follows immediately that $\psi \in C^\infty$, because $\phi \in C^\infty$. It remains to analyze the growth of ψ. For any $x \in \mathbb{R}$ we find that

$$\begin{aligned}\psi'(x) &= \frac{\phi'(x)(\phi(x)+\phi(1-x)) - \phi(x)(\phi'(x) - \phi'(1-x))}{[\phi(x)+\phi(1-x)]^2} \\ &= \frac{\phi'(x)\phi(1-x) + \phi(x)\phi'(1-x)}{[\phi(x)+\phi(1-x)]^2}.\end{aligned} \tag{A27}$$

From this expression we see that $\psi'(x) > 0$ for $x \in (0,1)$, because $\phi(t) > 0$ and $\phi'(t) > 0$ for all $t > 0$. We conclude that ψ is strictly increasing for $x \in (0,1)$. Finally, we note that $\psi(x) = 0$ for any $x \leq 0$ because $\phi(x) = 0$ for $x \leq 0$ and $\psi(x) = 1$ for $1 \leq x$ because $\phi(1-x) = 0$ for $1 \leq x$. This completes the proof. □

Now consider the problem of smoothly truncating a function $f : \mathbb{R} \to \mathbb{R}$. Let $a < b$ be given and let $g : \mathbb{R} \to \mathbb{R}$ be given by

$$g(x) = f(x)\psi\left(\frac{b-x}{b-a}\right) \tag{A28}$$

where ψ is defined by Lemma 4. Then g is as smooth as f and $g(x) = f(x)$ for $x < a$ and $g(x) = 0$ for $b < x$.

Smooth Interpolation

Let $f \in C^m([a,b],\mathbb{R})$ and let

$$a < x_1 < x_2 < \cdots < x_n < b \tag{A29}$$

denote a set of n nodes in (a,b) and consider the problem of finding a function

$$g \in C^\infty([a,b],\mathbb{R}) \tag{A30}$$

such that

$$\forall i \in \{1, 2 \ldots, n\}\ \forall j \in \{0, 1, \ldots, m\} \ : \ g^{(j)}(x_i) = f^{(j)}(x_i) \tag{A31}$$

Our goal is simplicity rather than computational efficiency, and we shall make no effort to reduce the number of arithmetic operations.

We begin by constructing a smooth "tent" function g_j centered at the node x_j. We have deliberately made the unusual choice of $a < x_1$ and $x_n < b$ rather than the natural choice of $a = x_1$ and $x_n = b$. This allows us to define "dummy" nodes $x_0 = a$ and $x_{n+1} = b$. We are therefore free to concentrate on a single node x_i where $i \in \{1, 2, \ldots, n\}$, secure in the knowledge that there is a node x_{i-1} to our left and a node x_{i+1} to our right.

Lemma 5. *Let $a < b < c$ and letψ be as in Lemma 4. Let $\psi_\pm : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ denote the functions given by*

$$\psi_-(x) = \psi\left(\frac{x-a}{b-a}\right), \quad \psi_+(x) = \psi\left(\frac{x-c}{b-c}\right) \tag{A32}$$

and

$$g(x) = \begin{cases} \psi_-(x) & x < b, \\ 1 & x = b, \\ \psi_+(x) & x > b \end{cases} \tag{A33}$$

Then $\psi_\pm \in C^\infty$ and $g \in C_c^\infty$ has $\mathrm{supp}(g) \subseteq [a,c]$ *and*

$$\forall k \in \mathbb{N} \ : \ g^{(k)}(b) = 0. \tag{A34}$$

Proof. We observe that $g(b) = 1$ implies that $g \in C$. The chain rule implies that $\psi_\pm \in C^\infty$ because $\psi \in C^\infty$. It follows that g is infinitely differentiable for $x \neq b$ with

$$\forall j \in \mathbb{N} \ : \ g^{(j)}(x) = \psi^{(j)}\left(\frac{x-a}{b-a}\right)(b-a)^{-j} \tag{A35}$$

for $x < b$

$$\forall j \in \mathbb{N} \ : \ g^{(j)}(x) = \psi^{(j)}\left(\frac{x-c}{b-c}\right)(b-c)^{-j} \tag{A36}$$

for $b < x$. The mean value theorem now implies that $g^{(j-1)}$ is differentiable at $x = b$ with derivative 0. This completes the proof. □

We can now define an interpolant of class C^∞ that matches f and as many of its derivatives as we like at the nodes.

Lemma 6. *Let $f \in C^m([a,b],\mathbb{R})$ and let $a < x_1 < x_2 < \cdots < x_n < b$ denote a set of n internal nodes. Define $x_0 = a$ and $b = x_{n+1}$. For each $i \in \{1, 2, \ldots, n\}$, let $p_i : \mathbb{R} \to \mathbb{R}$ denote the Taylor polynomial of order m of f at the point x_i and let$g_i : \mathbb{R} \to \mathbb{R}$ denote the tent centered at x_i and defined by Lemma 5 with $(a,b,c) = (x_{i-1}, x_i, x_{i+1})$. Define $g : \mathbb{R} \to \mathbb{R}$ as follows*

$$g(x) = \sum_{i=1}^{n} g_i(x)p_i(x). \tag{A37}$$

Then $g \in C^\infty$ and

$$\forall i \in \{1, 2, \ldots, n\}\ \forall j \in \{1, 2, \ldots, m\} \ : \ g^j(x_i) = f^{(j)}(x_i). \tag{A38}$$

Proof. It is clear that $g \in C^\infty$ because it is a finite sum of smooth functions defined for all $x \in \mathbb{R}$. By Leibniz's rule, we have

$$g^{(j)}(x) = \sum_{i=1}^{n}\sum_{k=1}^{j}\binom{j}{k} g_i^{(k)}(x)p_i^{(j-k)}(x) \tag{A39}$$

Let $x = x_l$ denote one of the internal nodes. Since g_i has support in $[x_{i-1}, x_{i+1}]$ we automatically have $g_i^{(j)}(x_l) = 0$ for all $i \neq l$ and any j and the sum reduces to

$$g^{(j)}(x_l) = \sum_{k=0}^{j}\binom{j}{k} g_l^{(k)}(x_l)p_l^{(j-k)}(x_l) \tag{A40}$$

By Lemma 5 we have $g_l(x_l) = 1$ and $g_l^{(k)}(x_l) = 0$ for $k > 0$. It follows that the sum reduces to the single term corresponding to $k = 0$, that is,

$$g^{(j)}(x_l) = g_l(x_l)p_l^{(j)}(x_l) = f^{(j)}(x_l) \tag{A41}$$

as required. This completes the proof. □