Journal of Scientific Computing (2023) 97:11
https://doi.org/10.1007/s10915-023-02324-0

Optimal Parameters for Numerical Solvers of PDEs

Gianluca Frasca-Caccia[1] · **Pranav Singh**[2]

Received: 9 December 2022 / Revised: 20 July 2023 / Accepted: 15 August 2023 /
Published online: 7 September 2023

Abstract
In this paper we introduce a procedure for identifying optimal methods in parametric families of numerical schemes for initial value problems in partial differential equations. The procedure maximizes accuracy by adaptively computing optimal parameters that minimize a defect-based estimate of the local error at each time step. Viable refinements are proposed to reduce the computational overheads involved in the solution of the optimization problem, and to maintain conservation properties of the original methods. We apply the new strategy to recently introduced families of conservative schemes for the Korteweg-de Vries equation and for a nonlinear heat equation. Numerical tests demonstrate the improved efficiency of the new technique in comparison with existing methods.

Keywords Parameter optimization · Finite difference methods · Conservation laws · KdV equation · Nonlinear diffusion equation

Mathematics Subject Classification 90C31 · 65M06 · 65M20 · 65L05

1 Introduction

Many highly effective methods for initial value problems in partial differential equations appear as parametric families of numerical schemes. These include exponential splittings (see, e.g., [8, 25]), where the free parameters constitute the coefficients of the splitting, and rational Krylov methods [20], where the free parameters are the poles of rational approximants.

A new technique that uses symbolic algebra to develop bespoke finite difference methods that preserve multiple local conservation laws has been recently introduced in [14]. This approach has been further refined in [16], and new families of conservative schemes have been introduced for a range of partial differential equations (PDEs) in [13–16]. These numerical schemes feature certain free parameters that can be arbitrarily chosen without compromising the preservation of the conservation laws.

✉ Gianluca Frasca-Caccia
gfrascacaccia@unisa.it

[1] Dipartimento di Matematica, Università degli Studi di Salerno, Via Giovanni Paolo II 132, 84084 Fisciano, SA, Italy

[2] Department of Mathematical Sciences, University of Bath, 6 West, Claverton Down, Bath BA2 7AY, UK

A convenient choice of the free parameters yields numerical solutions with superior accuracy in all these cases. Coefficients of exponential splittings are typically determined a-priori using algebraic means in the pursuit of high order accuracies [28] and may be specialized for specific PDEs [31]. Optimal pole selection for rational Krylov methods remains an active area of research and strategies include a-priori choices based on analytical reasoning [17] and a-posteriori fitting [7]. Optimal parameters for the finite difference methods in [13–16] are identified using a brute force sweep through the entire parameter space, and comparisons against reference solutions show that suitable choices of the parameters yield errors up to 20 times smaller than existing methods for the proposed benchmark tests.

In practice, the optimal choice of such free parameters depends heavily on initial conditions and may also vary with time-step. Consequently, while the results in [13–16] do highlight the potential advantages of choosing good parameters, there is no known algorithm for identifying them. In order to overcome this issue we propose here a new approach for adaptively identifying optimal parameters for families of numerical schemes for PDEs, where convenient values are not known a priori.

For obtaining estimates of the optimal parameters we adaptively minimize an estimate of the local error introduced by the time integrator. In order for this approach to be effective, we assume throughout the paper that the spatial approximation is accurate and that the error in the solution is mainly due to the time discretization. This is not too restrictive an assumption, as in many instances PDEs are approximated very accurately in space, using for example spectral semidiscretizations. In the case of finite difference schemes, this amounts to either considering higher order discretizations in space, or restricting attention to cases where $\Delta t \gg \Delta x$. Large time-steps reduce computational expenses and are generally desirable, except for potential stability concerns. In particular, $\Delta t \gg \Delta x$ is a typical setting when using implicit schemes.

In the proposed approach, at each time-step of a single step numerical scheme, we seek to compute the optimal parameters that minimize the local error. This requires a reasonably accurate but inexpensive estimate of the local error and its dependence on the parameters. For an a posteriori estimate of the local error, we resort to the "defect" based approach outlined in [6]. In the context of backward error analysis, the defect measures the discrepancy between the differential equation satisfied by the numerical solution and the original equation [30]. Defect based error estimates have been utilized widely in the development of time-adaptive methods for ordinary differential equations (ODEs) (see, e.g., [12, 21]) and PDEs [3, 4, 6], but, to the best of our knowledge, these have not been employed for the estimation of optimal parameters.

In this paper we do not aim for adaptive time stepping. Indeed, unlike adaptive techniques for choosing time-steps, where the local error can be assumed to decrease monotonically with the time-step, in the proposed approach an optimization problem needs to be solved for finding the values of the parameters. The optimization problem for minimizing the local error estimate is solved in an efficient manner by computing the defect on a coarser (but still accurate) spatial grid, and utilizing an iterative method with a Gauss–Newton approximation to the Hessian for achieving fast local convergence [29].

We apply the new procedure to families of schemes introduced in [14] for the Korteweg de Vries (KdV) and a nonlinear heat (NLH) equation. The main feature of these schemes is that each of them preserves specific discretizations of some conservation laws. However, since the discrete conservation laws also depend on the parameters, these cannot be preserved by using an adaptive approach. Where conservation of these properties is of paramount importance, we suggest a more conservative version of the algorithm that uses fixed parameters derived from a sequence of values obtained adaptively.

Although the conservation properties of the schemes in [14] confer them good stability and accuracy, by applying the technique proposed in this paper we achieve significantly higher accuracy with moderate overheads. Despite the defect based approach being asymptotic in the time-step, Δt, in practice the procedure also works well in the large time-steps regime and, in some cases, also confers a notable stability advantage.

In Sect. 2 we discuss the validity of a defect based approximation of the local error for the purpose of adaptively identifying optimal parameters. In Sect. 3 we outline the defect based approach used for finding optimal values of free parameters in a numerical scheme, and introduce the two algorithms briefly outlined above. In Sect. 4, we apply the new techniques to families of conservative schemes introduced in [14] for the KdV equation and the NLH equation, giving explicit expressions for the defect. In Sect. 5, we show numerical results that demonstrate the effectiveness of the proposed algorithms in finding good estimates of the optimal parameters, together with their higher accuracy and efficiency in comparison to a default choice of the parameters and other schemes from the literature. Conclusive remarks are presented in Sect. 6.

2 Defect Based Approximation of Local Error

We consider a PDE,

$$\partial_t u(t) = \mathcal{A}(u(t)), \qquad t \geq 0, \qquad u(0) = u_0 \in \mathcal{H}, \tag{1}$$

written as an Initial Value Problem on the Hilbert space $\mathcal{H}$, where $\mathcal{A} : \mathcal{H} \to \mathcal{H}$. Boundary conditions and non-autonomous PDEs can also be incorporated into our approach in a straightforward manner, as demonstrated with concrete examples in Sect. 4.2.

Following spatial discretization, the solution of (1) is approximated by the solution of the system of ODEs,

$$\mathcal{D}_t \boldsymbol{u}(t) = A(\boldsymbol{u}(t)), \qquad t \geq 0, \quad \boldsymbol{u}(0) = \boldsymbol{u}_0 \in \mathbb{R}^M, \tag{2}$$

where here and henceforth $\mathcal{D}_z$ denotes the total derivative with respect to z, and $\boldsymbol{u}(t)$ represents a finite dimensional approximation of $u(t)$. For instance, this could involve a finite difference approximation on a uniform grid on the domain $[a, b]$ with Dirichlet boundaries,

$$x_m = a + m\Delta x, \qquad m = 0, \ldots, M+1, \qquad \Delta x = (b-a)/(M+1).$$

Let T be the final time of integration,

$$t_n = n\Delta t, \qquad n = 0, \ldots, N, \quad \text{and} \quad \Delta t = T/N$$

be the time nodes and stepsize, respectively, $u_{m,n}$ an approximation of $u(x_m, t_n)$, and $\boldsymbol{u}_n$ the column vector whose m-th entry is $u_{m,n}$.

The exact solution of (2) is described by the flow $\mathcal{E} : \mathbb{R}^+ \times \mathbb{R}^M \to \mathbb{R}^M$,

$$\boldsymbol{u}(t) = \mathcal{E}(t, \boldsymbol{u}_0).$$

Similarly, a single step numerical scheme for (2) can be described by the numerical flow,

$$\boldsymbol{u}_{n+1} = \Phi(\Delta t, \boldsymbol{u}_n).$$

Note that the numerical flow Φ also exists for implicit methods, even if not specified in an explicit form.

In this manuscript, we consider numerical schemes in the form

$$\boldsymbol{u}_{n+1} = \Phi(\Delta t, \boldsymbol{u}_n, \chi), \qquad \Phi : \mathbb{R}^+ \times \mathbb{R}^M \times \Omega \to \mathbb{R}^M, \tag{3}$$

where Ω is a compact subset of $\mathbb{R}^K$, and Φ depends on a vector of free parameters $\chi \in \Omega$ that effect the accuracy of the scheme. In our theoretical discussion we assume that the vector field A, the exact flow $\mathcal{E}$ and the numerical flow Φ are smooth with respect to all arguments and that the method,

$$\boldsymbol{u}_{n+1} = \Phi(\Delta t, \boldsymbol{u}_n, \chi(t_n)), \tag{4}$$

is stable and convergent for arbitrary choices of $\chi : \mathbb{R}^+ \to \Omega$.

The local error in the numerical method (3) is defined as

$$\mathcal{L}(\Delta t, \boldsymbol{u}_n, \chi) = \Phi(\Delta t, \boldsymbol{u}_n, \chi) - \mathcal{E}(\Delta t, \boldsymbol{u}_n). \tag{5}$$

In general, $\mathcal{L}(\Delta t, \boldsymbol{u}_n, \chi)$ is not a computable quantity since the exact solution $\mathcal{E}(\Delta t, \boldsymbol{u}_n)$ is not available in practice. Consequently, we resort to defect-based approximations (see [5, 6]) to obtain a posteriori estimates. The *defect* or *residual* of Φ,

$$\mathcal{R}(\Delta t, \boldsymbol{u}_n, \chi) = \mathcal{D}_{\Delta t}\Phi(\Delta t, \boldsymbol{u}_n, \chi) - A(\Phi(\Delta t, \boldsymbol{u}_n, \chi)), \tag{6}$$

quantifies the extent to which the numerical flow Φ fails to satisfy (2).

To study the accuracy of the defect (6) as an approximation of the local error, we resort to the following nonlinear variation-of-constant formula, valid for nonlinear parabolic PDEs and time-reversible equations [4, 11].

Lemma 1 (Gröbner–Alekseev formula) *The analytical solutions of the following initial value problems*

$$\begin{cases} \mathcal{D}_t \boldsymbol{u}(t) = H(t, \boldsymbol{u}(t)) = G(\boldsymbol{u}(t)) + R(t, \boldsymbol{u}(t)), & 0 \le t \le T, \\ \boldsymbol{u}(0) = \boldsymbol{u}_0, \end{cases}$$

$$\begin{cases} \mathcal{D}_t \boldsymbol{u}(t) = G(\boldsymbol{u}(t)), & 0 \le t \le T, \\ \boldsymbol{u}(0) = \boldsymbol{u}_0, \end{cases}$$

are related through the nonlinear variation-of-constants formula

$$\mathcal{E}_H(t, \boldsymbol{u}_0) = \mathcal{E}_G(t, \boldsymbol{u}_0) + \int_0^t \partial_2 \mathcal{E}_G(t-\tau, \mathcal{E}_H(\tau, \boldsymbol{u}_0)) \cdot R(\tau, \mathcal{E}_H(\tau, \boldsymbol{u}_0)) \mathrm{d}\tau, \quad 0 \le t \le T.$$

Here and henceforth, $\partial_k f$ denotes the Fréchet derivative of a function f with respect to the k-th argument.

Applying Lemma 1 to (2) and (6) yields the following formula for the local error (5),

$$\begin{aligned} \mathcal{L}(\Delta t, \boldsymbol{u}_n, \chi) &= \int_0^{\Delta t} \partial_2 \mathcal{E}\left(\Delta t - \tau, \Phi(\tau, \boldsymbol{u}_n, \chi)\right) \cdot \mathcal{R}(\tau, \boldsymbol{u}_n, \chi)\, \mathrm{d}\tau \\ &=: \int_0^{\Delta t} \Theta(\tau, \boldsymbol{u}_n, \chi)\, \mathrm{d}\tau. \end{aligned} \tag{7}$$

Theorem 1 *Given a numerical scheme* Φ *(3) of order* p*, the* defect-based estimator

$$L(\Delta t, \boldsymbol{u}_n, \chi) := \frac{\Delta t}{p+1} \mathcal{R}(\Delta t, \boldsymbol{u}_n, \chi), \tag{8}$$

is an asymptotically correct *estimator of the local error (5), uniformly for* $\chi \in \Omega$*, i.e.*

$$\|L(\Delta t, \boldsymbol{u}_n, \chi) - \mathcal{L}(\Delta t, \boldsymbol{u}_n, \chi)\| \leq C\Delta t^{p+2},$$

with C *independent of* χ.

Proof Since the method is of order p, then $\mathcal{L}(\Delta t, \boldsymbol{u}_n, \chi) = \mathcal{O}(\Delta t^{p+1})$, and $\Theta(\tau, \boldsymbol{u}_n, \chi) = \mathcal{O}(\tau^p)$ for any Δt, $\tau \in [0, \Delta t]$, and $\chi \in \Omega$. Taylor expanding Θ in τ around 0, there exists $\xi_1 \in [0, \tau]$ such that

$$\Theta(\tau, \boldsymbol{u}_n, \chi) = \frac{\tau^p}{p!}\partial_1^p \Theta(0, \boldsymbol{u}_n, \chi) + \frac{\tau^{p+1}}{(p+1)!}\partial_1^{p+1}\Theta(\xi_1, \boldsymbol{u}_n, \chi). \tag{9}$$

Therefore,

$$\begin{aligned}\mathcal{L}(\Delta t, \boldsymbol{u}_n, \chi) &= \int_0^{\Delta t} \Theta(\tau, \boldsymbol{u}_n, \chi)\,\mathrm{d}\tau \\ &= \frac{\Delta t^{p+1}}{(p+1)!}\partial_1^p \Theta(0, \boldsymbol{u}_n, \chi) + \int_0^{\Delta t} \frac{\tau^{p+1}}{(p+1)!}\partial_1^{p+1}\Theta(\xi_1, \boldsymbol{u}_n, \chi)\,\mathrm{d}\tau\,. \end{aligned} \tag{10}$$

Due to smoothness of Θ, $\partial_1^{p+1}\Theta$ is continuous and is bounded over the compact set $[0, \Delta t] \times \Omega$, so that we may define

$$\widetilde{C} = \max_{(\xi,\chi)\in[0,\Delta t]\times\Omega} \left\|\partial_1^{p+1}\Theta(\xi, \boldsymbol{u}_n, \chi)\right\| < \infty.$$

Note that by definition of Θ in (7), $\Theta(\Delta t, \boldsymbol{u}_n, \chi) = \partial_2\mathcal{E}\,(0, \Phi(\Delta t, \boldsymbol{u}_n, \chi))\cdot\mathcal{R}(\Delta t, \boldsymbol{u}_n, \chi) = \mathcal{R}(\Delta t, \boldsymbol{u}_n, \chi)$. Using (10) and applying (9) with $\tau = \Delta t$, for some $\xi_1, \xi_2 \in [0, \Delta t]$,

$$\begin{aligned}&\left\|\mathcal{L}(\Delta t, \boldsymbol{u}_n, \chi) - \frac{\Delta t}{p+1}\mathcal{R}(\Delta t, \boldsymbol{u}_n, \chi)\right\| = \left\|\mathcal{L}(\Delta t, \boldsymbol{u}_n, \chi) - \frac{\Delta t}{p+1}\Theta(\Delta t, \boldsymbol{u}_n, \chi)\right\| \\ &= \left\|\int_0^{\Delta t} \frac{\tau^{p+1}}{(p+1)!}\partial_1^{p+1}\Theta(\xi_1, \boldsymbol{u}_n, \chi)\,\mathrm{d}\tau - \frac{\Delta t^{p+2}}{(p+1)!(p+1)}\partial_1^{p+1}\Theta(\xi_2, \boldsymbol{u}_n, \chi)\right\| \\ &\leq \left\|\int_0^{\Delta t} \frac{\tau^{p+1}}{(p+1)!}\partial_1^{p+1}\Theta(\xi_1, \boldsymbol{u}_n, \chi)\,\mathrm{d}\tau\right\| + \left\|\frac{\Delta t^{p+2}}{(p+1)!(p+1)}\partial_1^{p+1}\Theta(\xi_2, \boldsymbol{u}_n, \chi)\right\| \\ &\leq \left[\frac{1}{(p+2)!} + \frac{1}{(p+1)!(p+1)}\right]\widetilde{C}\Delta t^{p+2} =: C\Delta t^{p+2},\end{aligned}$$

where

$$C = \frac{2p+3}{(p+2)!(p+1)}\widetilde{C},$$

is independent of Δt and χ. Therefore, indeed

$$\|L(\Delta t, \boldsymbol{u}_n, \chi) - \mathcal{L}(\Delta t, \boldsymbol{u}_n, \chi)\| \leq C\Delta t^{p+2},$$

uniformly for any value of $\chi \in \Omega$.

Remark 1 Theorem 1 is a generalisation of the results of [6], where the defect based estimator (8) is shown to be an asymptotically correct estimator for the local error. An important distinction here is the correctness of this estimator uniformly with respect to the parameters χ, which is essential for applications in optimal parameter selection. Note that the compactness of the parameter set Ω is crucial for the proof, but in practice is typically not a severe restriction.

Remark 2 Formula (8) provides an estimate of the local error of the time integrator. Nevertheless, there is no guarantee that the global error behaves in a similar way. This may occur even when accurate space discretizations are considered, e.g., caused by instabilities and constraints on the ratio of the discretization stepsizes. However, in Sect. 5 we show an example where the proposed approach, based on finding the parameters χ minimizing the quantity L in (8), prevents the occurrence of instabilities.

3 Defect Based Identification of Optimal Parameters

In this section, we propose the use of the defect based error estimate (8) for finding, at every time-step, optimal parameters $\chi_n^* \in \Omega \subset \mathbb{R}^K$, defined as

$$\chi_n^* = \arg\min_{\chi\in\Omega} \|L(\Delta t, \boldsymbol{u}_n, \chi)\|_2 \,, \tag{11}$$

where $\boldsymbol{u}_n$ and Δt are fixed. The result of Theorem 1 assures us that

$$\mathcal{L}(\Delta t, \boldsymbol{u}_n, \chi_n^*) = L(\Delta t, \boldsymbol{u}_n, \chi_n^*) + C\Delta t^{p+2},$$

and guarantees that the choice of parameters χ_n^* keeps the true local error, $\mathcal{L}$, close to $L(\Delta t, \boldsymbol{u}_n, \chi_n^*)$ in the asymptotic limit $\Delta t \to 0$ and, therefore, small.

Remark 3 The application of defect based error estimates for choosing optimal parameters differs from their application in context of time-adaptivity in a couple of crucial aspects.

1. Since L is neither monotonous in χ, nor are we interested in asymptotic limits for small χ (unlike the case of Δt in context of time-adaptivity), the defect based estimate $L(\Delta t, \boldsymbol{u}_n, \chi)$ needs to be computed for multiple values of χ within an optimization routine.
2. The perturbation of L by a term ρ independent of χ has no effect on χ^*. This is in contrast to time-adaptivity where we seek the largest Δt^* such that $\|L(\Delta t^*, \boldsymbol{u}_n, \chi)\|_2 < \delta$ for some user specified tolerance δ, and $\rho \neq 0$ effects the choice of Δt^*.

The first observation in Remark 3 suggests that the application of defect based estimates for choosing optimal parameters can be prohibitively expensive. However, the second suggests that we can resort to inexpensive approximations of the defect and still hope to arrive at a good choice of parameters.

In Sect. 3.2, we see that under reasonable assumptions, the number of optimization steps is not expected to be large and just a few steps of Gauss–Newton iteration are ever required. In the large time-step regime the approximation of defect on a coarse spatial grid also proves to be sufficient for the purposes described here. Overall, this leads to a procedure for identifying optimal parameters with very reasonable overheads, producing highly efficient schemes.

3.1 Optimization Procedure

In practice, we minimize the square of the defect,

$$\chi_n^* = \arg\min_{\chi\in\Omega} f(\chi), \qquad f(\chi) = \frac{1}{2}\|\mathcal{R}(\Delta t, \boldsymbol{u}_n, \chi)\|_2^2 \,,$$

using the gradient,

$$g = \left(\mathcal{D}_\chi \mathcal{R}(\Delta t, \boldsymbol{u}_n, \chi)\right)^\top \mathcal{R}(\Delta t, \boldsymbol{u}_n, \chi), \tag{12}$$

where $\mathcal{D}_\chi \mathcal{R}$ is the Jacobian of the defect with respect to χ, and a Gauss–Newton approximation to the Hessian,

$$\nabla^2 f \approx H_{\mathrm{GN}} := \left(\mathcal{D}_\chi \mathcal{R}(\Delta t, \boldsymbol{u}_n, \chi)\right)^\top \left(\mathcal{D}_\chi \mathcal{R}(\Delta t, \boldsymbol{u}_n, \chi)\right). \tag{13}$$

Utilizing the Gauss–Newton Hessian in the context of trust region algorithms yields a sequence of parameters χ_n^k that quickly converges to the optimal χ_n^*, with reliable global convergence properties [26]. At the same time, the procedure remains relatively inexpensive for a small number of parameters, K, since we only need to compute the first derivatives of the defect with regards to χ. These can be computed either analytically or approximately using finite differences.

The defect, $\mathcal{R}(\Delta t, \boldsymbol{u}_n, \chi_n^k)$, is computed using (6). This requires the computation of a temporary solution, $\widetilde{\boldsymbol{u}}_{n+1}^k = \Phi(\Delta t, \boldsymbol{u}_n, \chi_n^k)$, and of $\mathcal{D}_{\Delta t}\Phi(\Delta t, \boldsymbol{u}_n, \chi_n^k)$, the latter of which can be computed analytically as outlined with concrete examples in Sect. 4.

Note that, in general, at each iteration a trust region algorithm may be used to compute $\mathcal{R}$ at candidate parameters $\chi = \widetilde{\chi}_n^{k+1}$ before deciding to accept or reject the candidate and/or update the trust region radius Δ_n^k. For a detailed introduction to trust region algorithms, we refer the reader to [26].

3.2 Practical Considerations for Efficiency

The evaluations of defect can be very expensive, as they require the computation of the temporary solution $\widetilde{\boldsymbol{u}}_{n+1}^k$ at every iteration. Each of these is as expensive as a step of the original numerical solver Φ. However, in practice we identify χ_n^* by optimizing the defect on a coarse spatial grid, resorting to the fine computational grid only for evaluating $\boldsymbol{u}_{n+1}$ once χ_n^* has been identified.

The coarse grid is obtained as a subgrid of the fine grid with resolution $r\Delta x$, with r an integer that divides $M+1$. Let $\mathcal{P}_r$ denote the projection operator from the fine grid to the coarse grid, defined as

$$\mathcal{P}_r : \mathbb{R}^M \to \mathbb{R}^{(M+1)/r+1}, \qquad \widehat{\boldsymbol{u}}_n = \mathcal{P}_r(\boldsymbol{u}_n), \tag{14}$$

where

$$\begin{aligned} \boldsymbol{u}_n &= (u_{0,n}, u_{1,n}, \ldots, u_{i,n}, \ldots, u_{M,n}, u_{M+1,n}), \\ \widehat{\boldsymbol{u}}_n &= (u_{0,n}, u_{r,n}, \ldots, u_{ir,n}, \ldots, u_{M+1-r,n}, u_{M+1,n}). \end{aligned}$$

At the k-th iteration of Gauss-Newton algorithm, we evaluate the defect (6) on the coarse grid, as

$$\mathcal{R}(\Delta t, \widehat{\boldsymbol{u}}_n, \chi_n^k) = \mathcal{D}_{\Delta t}\Phi(\Delta t, \widehat{\boldsymbol{u}}_n, \chi_n^k) - A(\Phi(\Delta t, \widehat{\boldsymbol{u}}_n, \chi_n^k)), \qquad \widehat{\boldsymbol{u}}_n = \mathcal{P}_r(\boldsymbol{u}_n).$$

This requires the computation of $\widetilde{\boldsymbol{u}}_{n+1}^k = \Phi(\Delta t, \widehat{\boldsymbol{u}}_n, \chi_n^k)$. On a grid with resolution $r\Delta x$, the dimension of the problem is reduced by a factor r. This typically leads to a significant speedup in the computation of χ_n^*. This speedup is expected to be particularly pronounced in 2 or 3 dimensional problems, where the coarse grid is smaller by factors of r^2 and r^3, respectively, than the finer computational grid.

For a method of order p in space and time, a $\mathcal{O}(r^p \Delta x^p)$ term of error is introduced in the evaluation of the defect. This error is negligible if $r\Delta x \ll \Delta t$ and is not expected to have a significant effect on the estimate of the optimal parameters χ_n^* in light of Remark 3.

Remark 4 With larger values of r, the additional cost of identifying χ_n^* becomes marginal, while the advantages of identifying good parameters could still be significant. We can expect, however, that once r is large enough such that $r\Delta x \ll \Delta t$ is no longer valid, spatial discretization errors will start to dominate and the computation of defect may become too inaccurate to be useful. In light of these observations, as a rule of thumb, the largest r we recommend is the largest divisor of $M+1$ such that $r < \Delta t/\Delta x$.

Lipschitz continuity. Further gains can be obtained by exploiting temporal smoothness of the optimal parameters χ^*. In the context of some methods it may be reasonable to assume that the optimal parameters are described by a Lipschitz function $\chi^*(t)$, i.e. $|\chi_n^* - \chi_{n-1}^*| \leq \widetilde{C}\Delta t$ for some $\widetilde{C} < \infty$ independent of n. For small enough $\widetilde{C}\Delta t$, χ_{n-1}^* is close enough to χ_n^*. Thus, the previous value of the optimal parameter serves as a good first guess for the next time-step, $\chi_n^0 = \chi_{n-1}^*$. With $\widetilde{C}$ and Δt small enough, χ_n^0 can be expected close enough to χ_n^* so that conditions of trust-region are guaranteed to be satisfied and fast convergence is guaranteed. In such a case, it suffices to use the simple Newton-type iteration,

$$\chi_n^{k+1} = \chi_n^k - H_{\mathrm{GN}}^{-1}\, g, \qquad \chi_n^0 = \chi_{n-1}^*, \quad n > 0, \tag{15}$$

in place of the trust-region algorithm, where g and H_{GN} are given by (12) and (13), respectively. In practice, just a couple of steps of (15) suffice, with the exception of the first iteration ($n = 0$), when the arbitrary value of the initial guess may be very far from the optimal one, requiring more optimization steps and the use of trust-regions.

Gauss–Newton algorithm is iterated until the stopping criterion,

$$|\chi_n^{k+1} - \chi_n^k| < \mathrm{tol},$$

is satisfied for a suitable tolerance. Then we set $\chi_n^* = \chi_n^{k+1}$ and the solution at the next time-step is obtained on the finer computational grid as

$$\boldsymbol{u}_{n+1} = \Phi(\Delta t, \boldsymbol{u}_n, \chi_n^*).$$

The overall procedure introduced in this section is summarized in Algorithm 1.

Assuming that the method Φ in (4) is stable and convergent for a time-dependent choice of parameters $\chi : \mathbb{R}^+ \to \Omega$, it follows that the time-adaptive procedure is also stable and has the same order of convergence.

3.3 Modifications for Conservative Schemes

Algorithm 1 is very effective in improving accuracy of standard numerical methods with free parameters and of geometric integrators that preserve a structure that is independent of the parameters. However, some numerical schemes, e.g. the schemes described in [14], preserve conservation laws that depend explicitly on the parameters. Since the optimal parameters generated by Algorithm 1 change from time-step to time-step, it undermines the preservation of the conservation laws of these schemes.

We present an alternative approach for these numerical schemes, which post-processes the time-dependent optimal parameters and uses constant-in-time values of the parameters e.g. by using the mean-values. This allows numerical schemes with parameter-dependent conservation laws to respect their conservation laws as well, while benefitting from low local errors. We modify Algorithm 1 as follows.

Algorithm 1: Adaptively identifying optimal parameters

output: $\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N, \chi_0^*, \chi_1^*, \ldots, \chi_{N-1}^*$.
input : $\chi_0^0 \in \Omega, \boldsymbol{u}_0, \Phi, \Delta t, N, \Delta_0^0$, MaxIter, $\mathrm{tol}_1, \mathrm{tol}_2, r$.
for $n \leftarrow 0$ **to** $N-1$ **do**
  $\widehat{\boldsymbol{u}}_n = \mathcal{P}_r(\boldsymbol{u}_n)$;
  $k \leftarrow 0$;
  while $k \leq$ MaxIter **and** $|\chi_n^{k+1} - \chi_n^k| >$ tol **do**
    $\mathcal{R} \leftarrow \mathcal{R}(\Delta t, \widehat{\boldsymbol{u}}_n, \chi_n^k)$;
    $\mathcal{D}_\chi \mathcal{R} \leftarrow \mathcal{D}_\chi \mathcal{R}(\Delta t, \widehat{\boldsymbol{u}}_n, \chi_n^k)$;
    $g \leftarrow \left(\mathcal{D}_\chi \mathcal{R}\right)^\top \mathcal{R}$;
    $H_{\mathrm{GN}} \leftarrow \left(\mathcal{D}_\chi \mathcal{R}\right)^\top \left(\mathcal{D}_\chi \mathcal{R}\right)$;
    $\chi_n^{k+1}, \Delta_n^{k+1} \leftarrow$ **TrustRegion**$(\chi_n^k, \Delta_n^k, g, H_{\mathrm{GN}}, \widehat{\boldsymbol{u}}_n, \Phi)$;
    $k \leftarrow k+1$;
  end
  $\chi_n^* \leftarrow \chi_n^k$;
  $\boldsymbol{u}_{n+1} \leftarrow \Phi(\Delta t, \boldsymbol{u}_n, \chi_n^*)$;
  $\chi_{n+1}^0 \leftarrow \chi_n^*$;
  $\Delta_{n+1}^0 \leftarrow \Delta_n^k$;
end

1. Project the initial condition on a grid with resolution $r\Delta x$.
2. Find the optimal χ_0^* and use it to advance a step in time on the coarse grid. Iterate this step till the final time, obtaining the optimal parameters χ_n^* at $n = 0, \ldots, N-1$ steps.
3. Compute the average optimal parameters as $\overline{\chi}^* = \frac{1}{N} \sum_{n=0}^{N-1} \chi_n^*$.
4. Compute the numerical solution on the full computational grid using the average parameters, $\boldsymbol{u}_{n+1} = \Phi(\Delta t, \boldsymbol{u}_n, \overline{\chi}^*)$, for $n = 0, \ldots, N-1$.

We summarize this conservative version of our procedure in Algorithm 2. Note that the numerical solution obtained on the coarse grid, $\widehat{\boldsymbol{u}}_n$, is used only for estimating the optimal parameters and later discarded.

Since our estimate of the optimal parameters in Algorithm 2 relies on the solution of the problem on a coarser grid, this algorithm is also affected by the accumulation of errors in space, which may be particularly pronounced for large values of r. On the other hand, the parameters χ_n^* need not be identified with too high an accuracy since we only need a good average choice, $\overline{\chi}^*$.

Considering the average parameters for solving the problem on the fine grid is a good option particularly for solutions with simple and smooth dynamics (e.g. travelling waves) as the values obtained at the different time-step are all reasonably close to each other.

Remark 5 Note that at the end of step 2, the full sequence of optimal parameters, $\chi_0^*, \ldots, \chi_{N-1}^*$, and local error estimates are available to the user, and it is possible to consider alternative and more suitable options than the average values of the parameters. This could be particularly useful in cases where the solution changes its type substantially and also the parameters values vary in a wide range along the integration. For example, one may choose the value corresponding to a particularly delicate stage of the time evolution.

Algorithm 2: Identifying fixed optimal parameters

output: $\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N, \chi_0^*, \chi_1^*, \ldots, \chi_{N-1}^*$.
input : $\chi_0^0 \in \Omega, \boldsymbol{u}_0, \Phi, \Delta t, N, \Delta_0^0$, MaxIter, $\mathrm{tol}_1, \mathrm{tol}_2, r$.
$\widehat{\boldsymbol{u}}_0 \leftarrow \mathcal{P}_r(\boldsymbol{u}_0)$;
for $n \leftarrow 0$ **to** $N-1$ **do**
  $k \leftarrow 0$;
  while $k \leq$ MaxIter **and** $|\chi_n^{k+1} - \chi_n^k| >$ tol **do**
    $\mathcal{R} \leftarrow \mathcal{R}(\Delta t, \widehat{\boldsymbol{u}}_n, \chi_n^k)$;
    $\mathcal{D}_\chi \mathcal{R} \leftarrow \mathcal{D}_\chi \mathcal{R}(\Delta t, \widehat{\boldsymbol{u}}_n, \chi_n^k)$;
    $g \leftarrow \left(\mathcal{D}_\chi \mathcal{R}\right)^\top \mathcal{R}$;
    $H_{\mathrm{GN}} \leftarrow \left(\mathcal{D}_\chi \mathcal{R}\right)^\top \left(\mathcal{D}_\chi \mathcal{R}\right)$;
    $\chi_n^{k+1}, \Delta_n^{k+1} \leftarrow$ **TrustRegion**$(\chi_n^k, \Delta_n^k, g, H_{\mathrm{GN}}, \widehat{\boldsymbol{u}}_n, \Phi)$;
    $k \leftarrow k+1$;
  end
  $\chi_n^* \leftarrow \chi_n^k$;
  $\widehat{\boldsymbol{u}}_{n+1} \leftarrow \Phi(\Delta t, \widehat{\boldsymbol{u}}_n, \chi_n^*)$;
  $\chi_{n+1}^0 \leftarrow \chi_n^*$;
  $\Delta_{n+1}^0 \leftarrow \Delta_n^k$;
end
$\overline{\chi}^* \leftarrow \frac{1}{N} \sum_{n=0}^{N-1} \chi_n^*$;
for $n \leftarrow 0$ **to** $N-1$ **do**
  $\boldsymbol{u}_{n+1} \leftarrow \Phi(\Delta t, \boldsymbol{u}_n, \overline{\chi}^*)$;
end

4 Approximation of Defect for Specific Schemes

In this section we consider the application of the proposed approach to two partial differential equations—the KdV equation and a nonlinear heat equation—with suitable initial and boundary conditions. In particular, we present the computation of defect (6) for numerical schemes introduced in [14], which is required in Algorithms 1 and 2.

We define the forward shifts in space and time,

$$S_{\Delta x}(u_{m,n}) = u_{m+1,n}, \qquad S_{\Delta t}(u_{m,n}) = u_{m,n+1},$$

respectively, the forward difference and average operators,

$$D_{\Delta x} = \frac{S_{\Delta x} - I}{\Delta x}, \quad D_{\Delta t} = \frac{S_{\Delta t} - I}{\Delta t}, \quad \mu_{\Delta x} = \frac{S_{\Delta x} + I}{2}, \quad \mu_{\Delta t} = \frac{S_{\Delta t} + I}{2},$$

and the centred difference operator,

$$D_{2k,\Delta x} = D_{\Delta x}^{2k} S_{\Delta x}^{-k}, \qquad D_{2k-1,\Delta x} = D_{\Delta x}^{2k-1} S_{\Delta x}^{-k} \mu_{\Delta x},$$

approximating the space derivatives of degree $2k$ and $2k-1$, respectively, with second order of accuracy. Action of these operators on vectors is defined entrywise. Moreover, we denote with $\circ$ the Hadamard product whose action is entrywise multiplication of vectors.

For the two equations considered here, families of second order finite difference methods that depend on one or more free parameters have been introduced in [14] by means of a strategy based on the fact that, just like total divergences form the kernel of an Euler operator [27], there exists a discrete Euler operator whose kernel is the space of the difference divergences [23, 24]. We consider in this section some of these families of schemes.

Remark 6 We restrict attention to the setting where $\Delta t > \Delta x$, and the parameters χ featured in the numerical schemes from [14] are $\mathcal{O}(\Delta t^2)$. These small parameters χ correspond to perturbation terms that have no counterpart in the continuous problem, and whose contributions vanish in the limit $\Delta t \to 0$. Thus, it is reasonable to restrict our search to a neighbourhood of $\mathbf{0}$ of size $\mathcal{O}(\Delta t^2)$, $\Omega \subset \overline{B}_{C\Delta t^2}(\mathbf{0}; \mathbb{R}^K)$, for some constant $C > 0$. Therefore, when applying the algorithms outlined in Sect. 3 we can expect that $\chi = \mathbf{0}$ is a reasonable initial guess, and we can find convenient values of the parameters by using the simple iteration (15) without the aid of trust region methods.

Remark 7 The schemes considered here are all implicit. We assume that $\mathcal{J}$, the Jacobian operator defined by the partial derivatives of the scheme with respect to $\boldsymbol{u}_{n+1}$, is never singular. Solutions can then be obtained by iteration of Newton's method.

An important property of the numerical schemes considered here is that they preserve some conservation laws. Conservation laws are defined as total divergences,

$$\mathcal{D}_x F + \mathcal{D}_t G,$$

that vanish on solutions of the PDE. The functions F and G are the flux and the density of the conservation law and depend on x, t, u and its derivatives. The methods in [14] preserve second order approximations of specific conservation laws in the form

$$D_{\Delta x}\widetilde{F}(\boldsymbol{x}, \boldsymbol{u}_n, \boldsymbol{u}_{n+1}, \chi) + D_{\Delta t}\widetilde{G}(\boldsymbol{x}, \boldsymbol{u}_n, \chi) = 0,$$

where here and henceforth tildes represent approximations of the corresponding continuous terms, and $\boldsymbol{x}$ is the column vector whose m-th entry is x_m.

4.1 Schemes for KdV Equation

In this section we apply the approach outlined in Sect. 3 to parametric families of schemes for the KdV equation,

$$u_t + \left(\tfrac{1}{2}u^2 + u_{xx}\right)_x = 0, \tag{16}$$

with initial condition

$$u(x, 0) = u_0(x).$$

For simplicity, we restrict attention to periodic or zero boundary conditions. However, as shown in Sect. 4.2, the entire discussion can also be adapted to boundary conditions of a different type. The KdV equation has infinitely many independent conservation laws. The first three, in increasing order,

$$\begin{aligned}
&\mathcal{D}_x F_1 + \mathcal{D}_t G_1 \equiv \mathcal{D}_x\left(\tfrac{1}{2}u^2 + u_{xx}\right) + \mathcal{D}_t u = 0,\\
&\mathcal{D}_x F_2 + \mathcal{D}_t G_2 \equiv \mathcal{D}_x\left(\tfrac{1}{3}u^3 + uu_{xx} - \tfrac{1}{2}u_x^2\right) + \mathcal{D}_t\left(\tfrac{1}{2}u^2\right) = 0,\\
&\mathcal{D}_x F_3 + \mathcal{D}_t G_3 \equiv \mathcal{D}_x\left(\tfrac{1}{4}u^4 + u_x u_t - uu_{xt} + u^2 u_{xx} + u_{xx}^2\right) + \mathcal{D}_t\left(\tfrac{1}{3}u^3 + uu_{xx}\right) = 0,
\end{aligned} \tag{17}$$

describe the local conservation laws of mass, momentum and energy, respectively. For this equation we define the semidiscrete operator A in (2) in the most natural way, as

$$A(\boldsymbol{u}(t)) = -D_{1,\Delta x}\left(\tfrac{1}{2}\boldsymbol{u}^2(t) + D_{2,\Delta x}\boldsymbol{u}(t)\right). \tag{18}$$

The results obtained are independent of the particular form of this operator under spatial discretization, as we work under the assumption that the leading source of error is given by the time integration.

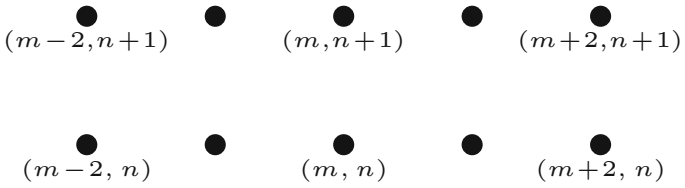

Fig. 1 10-point stencil for schemes EC(α) (19) and MC(β, γ) (22)

Energy conserving methods

We consider here the following family of mass and energy-conserving methods described in [14],

$$D_{\Delta t}(\boldsymbol{u}_n) + D_{\Delta x}\left(S_{\Delta x}^{-1}\mu_{\Delta x}\psi(\boldsymbol{u}_n, \boldsymbol{u}_{n+1}, \alpha)\right) = 0, \tag{19}$$

where

$$\psi(\boldsymbol{u}_n, \boldsymbol{u}_{n+1}, \alpha) = \tfrac{1}{6}(\boldsymbol{u}_{n+1}^2 + \boldsymbol{u}_n^2 + \boldsymbol{u}_n \circ \boldsymbol{u}_{n+1}) + D_{2,\Delta x}\mu_{\Delta t}\boldsymbol{u}_n + \alpha D_{\Delta t} D_{1,\Delta x}\boldsymbol{u}_n.$$

We denote with EC(α) the schemes in Eq. (19), where, according to Remark 6, α is a free parameter in a neighbourhood of 0, $\Omega \subset \overline{B}_{C\Delta t^2}(0;\mathbb{R}) = [-C\Delta t^2, C\Delta t^2]$ for some constant $C > 0$. For any choice of the parameter $\alpha \in \Omega$ schemes EC(α) are defined on the 10-point stencil in Fig. 1, they are implicit and second order accurate. Moreover, they have a discrete version of the conservation law of the mass (17) given by their definition (19) with,

$$\widetilde{F}_1 = S_{\Delta x}^{-1}\mu_{\Delta x}\psi(\boldsymbol{u}_n, \boldsymbol{u}_{n+1}, \alpha), \qquad \widetilde{G}_1 = \boldsymbol{u}_n.$$

and satisfy the discrete energy conservation law,

$$\begin{aligned}
&D_{\Delta x}\big(\widetilde{F}_3\big) + D_{\Delta t}\big(\widetilde{G}_3\big) = 0,\\
&\widetilde{F}_3 = \psi(\boldsymbol{u}_n, \boldsymbol{u}_{n+1}, \alpha)\circ S_{\Delta x}^{-1}\psi(\boldsymbol{u}_n, \boldsymbol{u}_{n+1}, \alpha) + \alpha(D_{\Delta t}\boldsymbol{u}_n)\circ(S_{\Delta x}^{-1}D_{\Delta t}\boldsymbol{u}_n)\\
&\qquad + S_{\Delta x}^{-1}\big((D_{\Delta x}\mu_{\Delta t}\boldsymbol{u}_n)\circ(D_{\Delta t}\mu_{\Delta x}\boldsymbol{u}_n) - (\mu_{\Delta x}\mu_{\Delta t}\boldsymbol{u}_n)\circ(D_{\Delta x}D_{\Delta t}\boldsymbol{u}_n)\big),\\
&\widetilde{G}_3 = \tfrac{1}{3}\boldsymbol{u}_n^3 + \boldsymbol{u}_n \circ D_{2,\Delta x}\boldsymbol{u}_n.
\end{aligned} \tag{20}$$

Being implicit methods, an explicit expression for Φ in (3) is not available. Nevertheless, an analytical expression of its time derivatives can be obtained by substituting (3) in (19) and differentiating. This yields,

$$\mathcal{D}_{\Delta t}\Phi(\Delta t, \boldsymbol{u}_n, \alpha) = -[\Delta t\mathcal{J}]^{-1}(D_{1,\Delta x}\psi(\boldsymbol{u}_n, \boldsymbol{u}_{n+1}, 0)), \tag{21}$$

where $\boldsymbol{u}_{n+1} = \Phi(\Delta t, \boldsymbol{u}_n, \alpha)$, and $\mathcal{J}$ denotes the Jacobian matrix defined in Remark 7.

Optimal values of α are then computed according to (11), where the defect is calculated according to (6) with (18) and (21).

Remark 8 As noted in Sect. 3.3, since the value of α changes at every time-step in Algorithm 1, it cannot preserve the conservation laws (19) and (20) of EC(α) because they depend on α. However, since the boundary conditions are conservative, summing the entries of the vectors in (19) and (20) yields

$$\sum_m u_{m,n+1} = \sum_m u_{m,n},$$

$$\sum_m \left(\tfrac{1}{3}u_{m,n+1}^3 + u_{m,n+1} D_{2,\Delta x} u_{m,n+1}\right) = \sum_m \left(\tfrac{1}{3}u_{m,n}^3 + u_{m,n} D_{2,\Delta x} u_{m,n}\right).$$

Therefore, EC(α) also preserves the following approximations of the global mass and energy:

$$\Delta x \sum_m u_{m,n}, \qquad \Delta x \sum_m \left(\tfrac{1}{3}u_{m,n}^3 + u_{m,n} D_{2,\Delta x} u_{m,n}\right).$$

These two global invariants are independent of α, and therefore they are conserved by both algorithms introduced in Sect. 3.

Momentum conserving methods

A two-parameter family of mass and momentum conserving schemes described in [14] is

$$\begin{aligned} D_{\Delta t}(\boldsymbol{u}_n) + D_{\Delta x}\Big\{ &\tfrac{1}{6}\big((S_{\Delta x}^{-1}\mu_{\Delta t}\boldsymbol{u}_n)^2 + (S_{\Delta x}^{-1}\mu_{\Delta t}\boldsymbol{u}_n)\circ(\mu_{\Delta t}\boldsymbol{u}_n) \\ &+(\mu_{\Delta t}\boldsymbol{u}_n)^2\big) + D_{\Delta x}^2 S_{\Delta x}^{-2}\mu_{\Delta x}\mu_{\Delta t}\boldsymbol{u}_n + D_{\Delta t} D_{\Delta x} S_{\Delta x}^{-1}(\beta\boldsymbol{u}_n + \gamma D_{2,\Delta x}\boldsymbol{u}_n)\Big\} = 0. \end{aligned} \tag{22}$$

We denote with MC(β, γ) the two-parameter family of schemes (22). For any value of the parameters $(\beta, \gamma) \in \Omega \subset \overline{B}_{C\Delta t^2}(0; \mathbb{R}^2)$, $C > 0$, the schemes MC(β, γ) are defined on the 10-point stencil in Fig. 1, are implicit and second order accurate. Solutions of MC(β, γ) satisfy the local mass conservation law given by (22), and the local momentum conservation law,

$$\begin{aligned} &D_{\Delta x}\big(\widetilde{F}_2\big) + D_{\Delta t}\big(\widetilde{G}_2\big) = 0, \\ &\widetilde{F}_2 = \tfrac{1}{3}(\mu_{\Delta t}\boldsymbol{u}_n)\circ(S_{\Delta x}^{-1}\mu_{\Delta t}\boldsymbol{u}_n)\circ(S_{\Delta x}^{-1}\mu_{\Delta t}\mu_{\Delta x}\boldsymbol{u}_n) + \tfrac{1}{2}(\mu_{\Delta t}\boldsymbol{u}_n)\circ(D_{2,\Delta x}\mu_{\Delta t}\boldsymbol{u}_n) \\ &\quad + \tfrac{1}{2}(S_{\Delta x}^{-1}\mu_{\Delta t}\boldsymbol{u}_n)\circ(D_{\Delta x}^2\mu_{\Delta t}\boldsymbol{u}_n) - \tfrac{1}{2}(D_{1,\Delta x}\boldsymbol{u}_n)^2 + \beta\rho(\boldsymbol{u}_n, \boldsymbol{u}_{n+1}) \\ &\quad + \gamma\sigma(\boldsymbol{u}_n, \boldsymbol{u}_{n+1}), \\ &\widetilde{G}_2 = \tfrac{1}{2}\boldsymbol{u}_n \circ \left(\boldsymbol{u}_n + D_{2,\Delta x}\left(\beta\boldsymbol{u}_n + \gamma D_{2,\Delta x}\boldsymbol{u}_n\right)\right), \end{aligned} \tag{23}$$

where

$$\begin{aligned} \rho(\boldsymbol{u}_n, \boldsymbol{u}_{n+1}) &= S_{\Delta x}^{-1}\big\{(\mu_{\Delta x}\mu_{\Delta t}\boldsymbol{u}_n)\circ(D_{\Delta x}D_{\Delta t}\boldsymbol{u}_n) - \tfrac{1}{2}D_{\Delta t}\big((\mu_{\Delta x}\boldsymbol{u}_n)\circ(D_{\Delta x}\boldsymbol{u}_n)\big)\big\}, \\ \sigma(\boldsymbol{u}_n, \boldsymbol{u}_{n+1}) &= \tfrac{1}{2}D_{\Delta t}\big\{S_{\Delta x}^{-1}\big((D_{\Delta x}\boldsymbol{u}_n)\circ(D_{\Delta x}^2\boldsymbol{u}_n)\big) - \boldsymbol{u}_n\circ(D_{\Delta x}^3 S_{\Delta x}^{-2}\boldsymbol{u}_n)\big\} \\ &\quad + \Big\{(\mu_{\Delta t}\boldsymbol{u}_n)\circ(D_{\Delta t}D_{\Delta x}^3 S_{\Delta x}^{-2}\boldsymbol{u}_n) - S_{\Delta x}^{-1}\big((D_{\Delta x}\mu_{\Delta t}\boldsymbol{u}_n)\circ(D_{\Delta t}D_{\Delta x}^2\boldsymbol{u}_n)\big)\Big\}. \end{aligned}$$

As also these schemes are fully implicit, we substitute (3) in (22) and differentiate in time. This yields,

$$\mathcal{D}_{\Delta t}\Phi(\Delta t, \boldsymbol{u}_n, \beta, \gamma) = [\Delta t\mathcal{J}]^{-1}A(\mu_{\Delta t}\boldsymbol{u}_n). \tag{24}$$

The defect and the local error estimate (8) are then computed using (18) and (24).

Remark 9 Summation of (22) and (23) gives that

$$\Delta x \sum_m u_{m,n}, \qquad \Delta x \sum_m \left(\tfrac{1}{2}u_{m,n}(u_{m,n} + \beta D_{2,\Delta x}u_{m,n} + \gamma D_{4,\Delta x}u_{m,n})\right),$$

are the discretizations of the global mass and momentum, respectively. The discrete global mass is independent of the parameters, and therefore it is conserved by both Algorithms 1 and 2. However, the discrete global momentum and the local conservation laws (22) and (23) are only preserved by Algorithm 2.

4.2 Schemes for a Nonlinear Heat Equation

In this section we consider the nonlinear heat equation,

$$u_t = \tfrac{1}{2}(u^2)_{xx}, \tag{25}$$

with initial condition and Dirichlet boundary conditions,

$$u(x,0) = u_0(x), \quad u(a,t) = \varphi_L(t), \quad u(b,t) = \varphi_R(t).$$

Equation (25) has only two independent conservation laws,

$$\mathcal{D}_x F_1 + \mathcal{D}_t G_1 \equiv \mathcal{D}_x(-uu_x) + \mathcal{D}_t(u) = 0, \tag{26}$$

$$\mathcal{D}_x F_2 + \mathcal{D}_t G_2 \equiv \mathcal{D}_x\left(\tfrac{1}{2}u^2 - xuu_x\right) + \mathcal{D}_t(xu) = 0. \tag{27}$$

We denote with CS(λ) the one-parameter family of methods described in [14], given by

$$D_{\Delta t}(\boldsymbol{u}_n + \lambda D_{2,\Delta x}\boldsymbol{u}_n) = \tfrac{1}{2} D_{2,\Delta x}(\boldsymbol{u}_n \circ \boldsymbol{u}_{n+1}). \tag{28}$$

The scheme CS(λ) has the following discrete versions of the conservation laws (26) and (27),

$$D_{\Delta x}\widetilde{F}_1 + D_{\Delta t}\widetilde{G}_1 = 0, \quad \widetilde{F}_1 = -S_{\Delta x}^{-1}\left(\tfrac{1}{2}\boldsymbol{u}_n \circ \boldsymbol{u}_{n+1} - \lambda D_{\Delta t}\boldsymbol{u}_n\right), \quad \widetilde{G}_1 = \boldsymbol{u}_n, \tag{29}$$

$$D_{\Delta x}\widetilde{F}_2 + D_{\Delta t}\widetilde{G}_2 = 0, \qquad \widetilde{G}_2 = \boldsymbol{x} \circ (\boldsymbol{u}_n + \lambda D_{2,\Delta x}\boldsymbol{u}_n),$$

$$\widetilde{F}_2 = S_{\Delta x}^{-1}\left(\mu_{\Delta x}(\tfrac{1}{2}\boldsymbol{u}_n \circ \boldsymbol{u}_{n+1}) - (\mu_{\Delta x}\boldsymbol{x}) \circ D_{\Delta x}(\tfrac{1}{2}\boldsymbol{u}_n \circ \boldsymbol{u}_{n+1})\right). \tag{30}$$

In order to evaluate the defect, we consider a centred difference space discretization of (25) in the form (2) with

$$A(\boldsymbol{u}(t),t) = \tfrac{1}{2}\left(\hat{D}_{2,\Delta x}\boldsymbol{u}^2(t) + \varphi_L^2(t)\boldsymbol{e}_1 + \varphi_R^2(t)\boldsymbol{e}_M\right), \tag{31}$$

where $\boldsymbol{e}_j$ is the j-th unit vector, and we define $\hat{D}_{2,\Delta x} = D_{2,\Delta x}|_{\varphi_L=\varphi_R=0}$ in order to isolate the contribution of the boundary conditions. The methods in (28) are linearly implicit so they can be written in the form (3) with

$$\begin{aligned}\Phi(\Delta t, \boldsymbol{u}_n, \lambda) = [\Delta t\mathcal{J}]^{-1}\big\{&\tfrac{\Delta t}{2\Delta x^2}\left(\varphi_L(t_{n+1})\varphi_L(t_n)\boldsymbol{e}_1 + \varphi_R(t_{n+1})\varphi_R(t_n)\boldsymbol{e}_M\right)\\ &+(1+\lambda\hat{D}_{2,\Delta x})\boldsymbol{u}_n - \lambda\big((\varphi_L(t_{n+1})-\varphi_L(t_n))\boldsymbol{e}_1 + (\varphi_R(t_{n+1})-\varphi_R(t_n))\boldsymbol{e}_M\big)\big\}.\end{aligned} \tag{32}$$

Differentiating (32) in time yields,

$$\begin{aligned}\mathcal{D}_{\Delta t}\Phi(\Delta t,\boldsymbol{u}_n,\lambda) = [\Delta t\mathcal{J}]^{-1}\big\{&\tfrac{1}{2}\hat{D}_{2,\Delta x}(\boldsymbol{u}_n \circ \boldsymbol{u}_{n+1}) - \lambda(\varphi_L'(t_{n+1})\boldsymbol{e}_1 + \varphi_R'(t_{n+1})\boldsymbol{e}_M)\\ &+ \tfrac{1}{2\Delta x^2}\big[(\varphi_L(t_{n+1}) + \Delta t\varphi_L'(t_{n+1}))\varphi_L(t_n)\boldsymbol{e}_1\\ &+ (\varphi_R(t_{n+1}) + \Delta t\varphi_R'(t_{n+1}))\varphi_R(t_n)\boldsymbol{e}_M\big]\big\},\end{aligned}$$

which, together with (31), is utilized in (8) and (6) to estimate the local error.

Remark 10 If the boundary conditions are conservative, summing the entries of the vectors in (29) and (30) yields

$$\sum_m u_{m,n+1} = \sum_m u_{m,n},$$

$$\sum_m x_m(u_{m,n+1} + \lambda D_{2,\Delta x}u_{m,n+1}) = \sum_m x_m(u_{m,n} + \lambda D_{2,\Delta x}u_{m,n}),$$

giving the following approximations of the global invariants

$$\Delta x \sum_m u_{m,n}, \qquad \Delta x \sum_m x_m (u_{m,n} + \lambda D_{2,\Delta x} u_{m,n}).$$

Since the former is independent of λ, this is a conserved invariant when using either of Algorithms 1 and 2. In general, the latter is conserved only by Algorithm 2. However, if the boundary conditions are such that

$$\sum_m x_m D_{2,\Delta x} u_{m,n} = 0,$$

the conservation of

$$\Delta x \sum_m x_m u_{m,n}$$

is also guaranteed by Algorithm 1. On a uniform spatial grid, this is achieved, for example, with zero boundary conditions.

5 Numerical Examples

In this section we consider a range of benchmark problems for the KdV Eq. (16) and the nonlinear heat Eq. (25), and investigate the performance of the methods described in Sect. 4 with optimal parameters obtained by the two algorithms introduced in Sect. 3. Comparisons between different numerical schemes are based on:

- Relative error in the solution at the final time $t = T$, defined as

$$\frac{\|\boldsymbol{u}_N - u_{\text{exact}}(T)\|}{\|u_{\text{exact}}(T)\|},$$

  where $\|\cdot\|$ denotes the discrete L^2 norm and u_{exact} is the solution of (1).
- Error in the variation of the global densities. If the method preserves the k-th conservation law,

$$D_{\Delta x} \widetilde{F}_k(\boldsymbol{x}, \boldsymbol{u}_n, \boldsymbol{u}_{n+1}, \chi) + D_{\Delta t} \widetilde{G}_k(\boldsymbol{x}, \boldsymbol{u}_n, \chi) = 0,$$

  the error in the global variation of G_k is defined as

$$\text{Err}_k = \Delta x \max_{n=1,\dots,N} \left| (\boldsymbol{e}_{M+1} - \boldsymbol{e}_1)^T \widetilde{F}_k(\boldsymbol{x}, \boldsymbol{u}_n, \boldsymbol{u}_{n+1}, \chi) + \mathbf{1}^T D_{\Delta t} \widetilde{G}_k(\boldsymbol{x}, \boldsymbol{u}_n, \chi) \right|, \quad (33)$$

  where $\boldsymbol{e}_j$ denotes the j-th column vector of the standard basis of $\mathbb{R}^{M+1}$, and $\mathbf{1} \in \mathbb{R}^{M+1}$ is the column vector with all entries equal to one.
  When the boundary conditions are periodic, we consider instead the maximum error on the k-th global invariant, defined as

$$\text{Err}_k = \Delta x \max_{n=1,\dots,N} \left| \mathbf{1}^T \left(\widetilde{G}_k(\boldsymbol{x}, \boldsymbol{u}_n, \chi) - \widetilde{G}_k(\boldsymbol{x}, \boldsymbol{u}_0, \chi) \right) \right|. \quad (34)$$

- Computational cost, measured in terms of computation time in seconds.

For each of the family of schemes described in Sect. 4, we consider the following choices of the vector of free parameters:

- $\chi = \mathbf{0}$, default choice, fixed at each step.

- $\chi = \chi^*$, globally optimal fixed value that minimizes the solution error for this specific problem. This value is obtained by brute force search based on empirical comparisons with the exact solution and is not available a priori. Thus, this choice of parameters does not constitute a reasonable numerical algorithm and we do not provide the computation time for it since it is excessive.
- $\{\chi^*_{r,n}\}_{n\geq 0}$, sequence of values that minimize the local error at each time-step obtained using Algorithm 1 projecting on a grid with resolution $r\Delta x$.
- $\chi = \overline{\chi}^*_r$, fixed value obtained using Algorithm 2 with projection on a coarse grid with spatial resolution $r\Delta x$.

As discussed in Remark 6, we apply both Algorithms 1 and 2 with the simplified Gauss–Newton step (15).

For all the experiments in this section, Δt and Δx are such that $4 < \Delta t/\Delta x < 10$. In order to verify the validity of the observations in Remark 4, we apply the proposed algorithms with $r = 1, 2, 4, 10$.

5.1 KdV Equation

In this section we solve the KdV equation (16), comparing schemes in Sect. 4.1 with different choices of the parameters against schemes known in literature – namely, the multisymplectic method,

$$D_{\Delta x}\left\{\tfrac{1}{2}\mu_{\Delta x}(\mu_{\Delta x}\mu_{\Delta t}u_{m-2,n})^2 + D_{2,\Delta x}\mu_{\Delta t}u_{m-1,n}\right\} + D_{\Delta t}\left\{\mu^3_{\Delta x}u_{m-2,n}\right\} = 0,$$

and the narrow box scheme,

$$D_{\Delta x}\left\{\tfrac{1}{2}(\mu_{\Delta t}u_{m-1,n})^2 + \mu_{\Delta t}(D_{2,\Delta x}u_{m-1,n})\right\} + D_{\Delta t}\left\{\mu_{\Delta x}u_{m-1,n}\right\} = 0,$$

introduced in [1, 2]. Both of these schemes preserve a discrete conservation law of the mass, given by their definition, but not of the momentum or the energy. These schemes are more compact than those defined in Sect. 4.1 and not centred on the grid. Therefore, we evaluate the error in the global momentum and energy according to (34) with

$$\widetilde{G}_2 = \tfrac{1}{2}(\mu_{\Delta x}u_{m-1,n})^2, \quad \widetilde{G}_3 = \tfrac{1}{3}(\mu_{\Delta x}u_{m-1,n})^3 + (\mu_{\Delta x}u_{m-1,n})D_{2,\Delta x}(\mu_{\Delta x}u_{m-1,n}).$$

For methods EC(α) (resp. MC(β, γ)) we evaluate the error (34) in the conservation of the momentum (resp. the energy) with $\widetilde{F}_2$ and $\widetilde{G}_2$ (resp. $\widetilde{F}_3$ and $\widetilde{G}_3$) given in (23) (resp. (20)) with all parameters set to zero. As a first numerical test we consider the motion of a soliton. The initial condition is obtained from the exact solution on $\mathbb{R}$,

$$u(x,t) = 3\,\mathrm{sech}^2\left(\tfrac{1}{2}(x - t + 5)\right),$$

evaluated at $t = 0$. We solve this problem over $[a, b] = [-20, 20]$ till the final time $T = 10$, and set a grid with $\Delta x = 0.05$ and $\Delta t = 0.4$.

We first find estimates for the optimal parameters of the schemes EC(α) and MC(β, γ) by applying the two new algorithms with projection (14) on grids with spatial resolution $r\Delta x$, $r = 1, 2, 4, 10$. In Fig. 2 we show the sequences of optimal values given by Algorithms 1 and 2, respectively, at each time step of scheme EC. Similarly, in Fig. 3 we show the sequences obtained at each time step of the scheme MC.

Since the solution only travels along the same direction with constant speed, after a few initial steps the sequences of parameters stabilize around constant values.

The sequences obtained by the two algorithms with $r = 1, 2, 4$, are all very close to each other (within a maximum distance of $5 \cdot 10^{-3}$). In agreement with Remark 4, those obtained

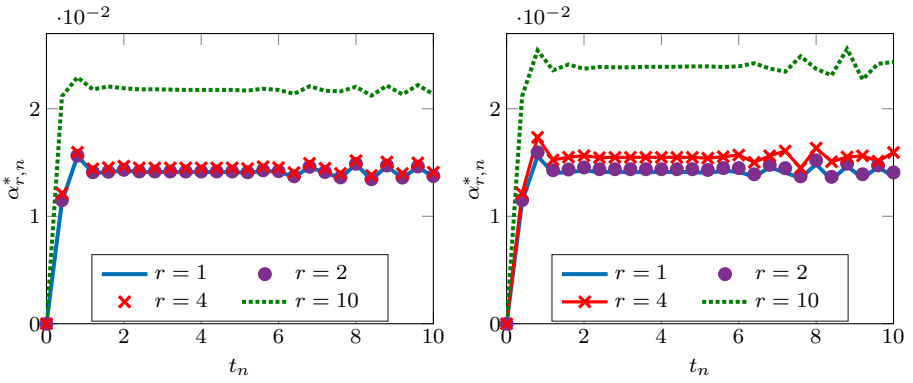

Fig. 2 One-soliton problem for KdV (16). Sequence of parameters $\{\alpha^*_{r,n}\}_{n\geq 0}$ for EC(α), (19), obtained using Algorithms 1 (left) and 2 (right) with different values of r

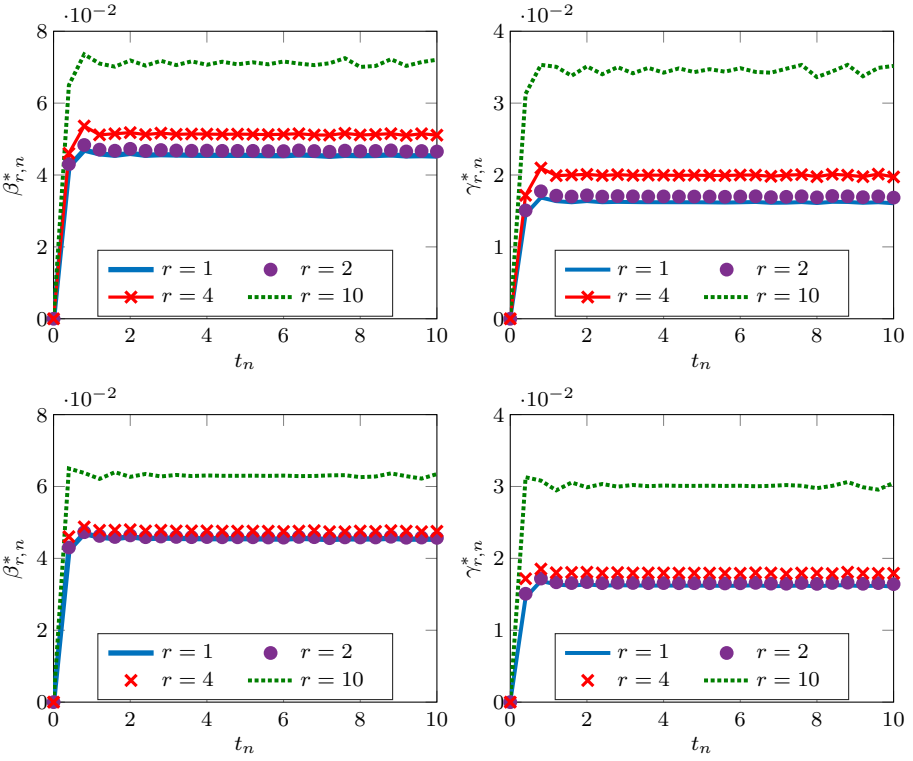

Fig. 3 One-soliton problem for KdV (16). Sequence of parameters $\{(\beta^*_{r,n}, \gamma^*_{r,n})\}_{n\geq 0}$ for MC(β, γ), (22), obtained using Algorithms 1 (top) and 2 (bottom) with different values of r

Table 1 One-soliton problem for KdV (16)

Method	Err_1	Err_2	Err_3	Sol. err	Time
$\mathrm{EC}(\alpha^*_{1,n})_{n\geq 0}$	1.88e−13	3.24e−04	6.25e−13	0.0139	32.33
$\mathrm{EC}(\alpha^*_{2,n})_{n\geq 0}$	1.92e−13	3.27e−04	6.43e−13	0.0139	14.43
$\mathrm{EC}(\alpha^*_{4,n})_{n\geq 0}$	1.31e−13	3.50e−04	4.51e−13	0.0133	8.06
$\mathrm{EC}(\alpha^*_{10,n})_{n\geq 0}$	2.03e−13	0.0010	1.06e−12	0.0092	6.51
$\mathrm{EC}(\overline{\alpha}^*_1) \equiv \mathrm{EC}(0.014)$	3.11e−13	3.28e−04	1.16e−12	0.0140	42.12
$\mathrm{EC}(\overline{\alpha}^*_2) \equiv \mathrm{EC}(0.014)$	2.84e−13	3.44e−04	8.14e−12	0.0136	16.39
$\mathrm{EC}(\overline{\alpha}^*_4) \equiv \mathrm{EC}(0.015)$	1.97e−13	4.15e−04	6.91e−13	0.0120	9.65
$\mathrm{EC}(\overline{\alpha}^*_{10}) \equiv \mathrm{EC}(0.024)$	1.10e−13	0.0013	5.81e−13	0.0113	7.10
$\mathrm{MC}(\beta^*_{1,n}, \gamma^*_{1,n})_{n\geq 0}$	9.33e−13	0.0017	4.90e−04	0.0135	37.38
$\mathrm{MC}(\beta^*_{2,n}, \gamma^*_{2,n})_{n\geq 0}$	6.64e−13	0.0017	5.08e−04	0.0140	16.26
$\mathrm{MC}(\beta^*_{4,n}, \gamma^*_{4,n})_{n\geq 0}$	3.32e−13	0.0019	5.70e−04	0.0161	8.00
$\mathrm{MC}(\beta^*_{10,n}, \gamma^*_{10,n})_{n\geq 0}$	1.92e−13	0.0028	9.45e−04	0.0242	6.95
$\mathrm{MC}(\overline{\beta}^*_1, \overline{\gamma}^*_1) \equiv \mathrm{MC}(0.045, 0.016)$	5.61e−13	3.03e−12	5.01e−04	0.0135	50.07
$\mathrm{MC}(\overline{\beta}^*_2, \overline{\gamma}^*_2) \equiv \mathrm{MC}(0.046, 0.017)$	5.12e−13	3.33e−12	5.08e−04	0.0135	17.56
$\mathrm{MC}(\overline{\beta}^*_4, \overline{\gamma}^*_4) \equiv \mathrm{MC}(0.048, 0.018)$	4.90e−13	4.35e−12	5.36e−04	0.0138	9.11
$\mathrm{MC}(\overline{\beta}^*_{10}, \overline{\gamma}^*_{10}) \equiv \mathrm{MC}(0.063, 0.030)$	7.14e−13	4.44e−12	8.00e−04	0.0181	7.74
EC(0)	1.39e−13	1.54e−04	3.27e−13	0.0376	6.07
MC(0, 0)	2.34e−13	4.67e−13	3.04e−04	0.0446	6.07
$\mathrm{EC}(\alpha^*) \equiv \mathrm{EC}(0.020)$	1.39e−13	8.26e−04	4.14e−13	0.0085	N/A
$\mathrm{MC}(\beta^*, \gamma^*) \equiv \mathrm{MC}(0.055, 0.031)$	1.23e−12	7.27e−12	7.57e−04	0.0083	N/A
Multisymplectic	9.52e−13	6.40e−06	2.73e−04	0.0447	8.55
Narrow box	2.22e−12	2.39e−06	2.90e−04	0.0434	7.24

Errors in solution and conservation laws, and computation time

with $r = 10 > \Delta t/\Delta x$ show the effect of the space error on the coarser grid. This shows that the accuracy in the solution can be compromised when r is too large. Nevertheless, these parameters are not too far from the values obtained on the computational grid, and still show a good level of accuracy in all cases, further reducing the computational overheads involved in the solution of the optimization problem.

In Table 1 we compare different schemes. The results obtained show that:

- The values of the maximum error on the k-th global invariant, Err_k, $k = 1, 2, 3$, evaluated according to (34), show that schemes EC and MC with fixed values of the parameters exactly preserve two conservation laws, and therefore two global invariants.
- According to Remark 8, due to the conservative boundary conditions, the sequence of schemes $\mathrm{EC}(\alpha^*_{r,n})_{n\geq 0}$ preserves the global mass and the global energy. Similarly, according to Remark 9, schemes $\mathrm{MC}(\beta^*_{r,n}, \gamma^*_{r,n})_{n\geq 0}$ preserve the global mass but not the global momentum.
- The computational cost of both of the proposed algorithms decreases as r increases, i.e., as the resolution of the grid (see Eq. (14)) used for the solution of the optimization problem increases. With respect to solving the optimization problem on the same grid of the differential problem ($r = 1$), the overall computational time reduces to less than a half when $r = 2$, and to less than a fourth when $r = 4$, making the computation cost of

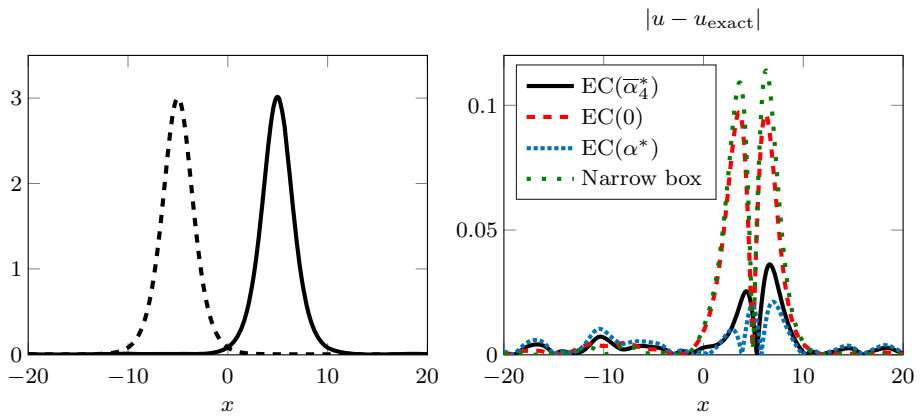

Fig. 4 One-soliton problem for KdV (16). Left: Initial condition (dashed line) and solution of EC($\overline{\alpha}_4^*$). Right: Pointwise error for different schemes at the final time

the methods proposed in this paper comparable to that of the other schemes in literature. As expected, setting $r = 10$, the computational time further reduces.

- All the approximations obtained with the sequences of parameters given by Algorithms 1 and 2 are more accurate than the solutions of the schemes from the literature and of EC(0) and MC(0,0). However, schemes EC(α^*) and MC(β^*, γ^*), where the parameters are obtained by brute force, are more precise. This shows that there exist sequences of parameters yielding higher accuracy than those obtained using the new algorithms. This is due to the fact that our approach is based on a minimization of (8) as an estimate of the local error. This improves local error but does not necessarily lead to a minimization of the global error.
- The accuracy of the solutions obtained using the two new algorithms is only marginally affected by the different choices of $r < \Delta t/\Delta x$. However, for $r = 10 > \Delta t/\Delta x$ the spatial error has a visible effect on the sequences of parameters obtained, as is evident in Figs. 2 and 3. In this case, the accuracy in the solution is only marginally affected but in general this may not be the case. In agreement with Remark 4, setting $r = 4$ gives the best compromise between reliability and speed of computation.
- Algorithms 1 and 2 with $r = 4$ are significantly more efficient than the schemes from the literature: while computation times are comparable, the solution error is roughly three times lower.

In the left plot of Fig. 4 we show the initial profile and, as an example, the solution of EC($\overline{\alpha}_4^*$) at the final time. In the right plot we show the absolute error given by EC($\overline{\alpha}_4^*$) at every point, in comparison with EC(0), EC(α^*), and the narrow box scheme. For all schemes, the bulk of the error is detected around the final location of the soliton and is due to a delay introduced by all numerical schemes. However, the maximum error introduced by EC($\overline{\alpha}_4^*$) is less than the 40% of that given by EC(0) and by the narrow box scheme, and only slightly larger than the error given by EC(α^*).

As a second numerical test we consider the interaction of two solitons over $[a, b] = [-30, 30]$ and till time $T = 15$. The initial condition is obtained from the exact solution on

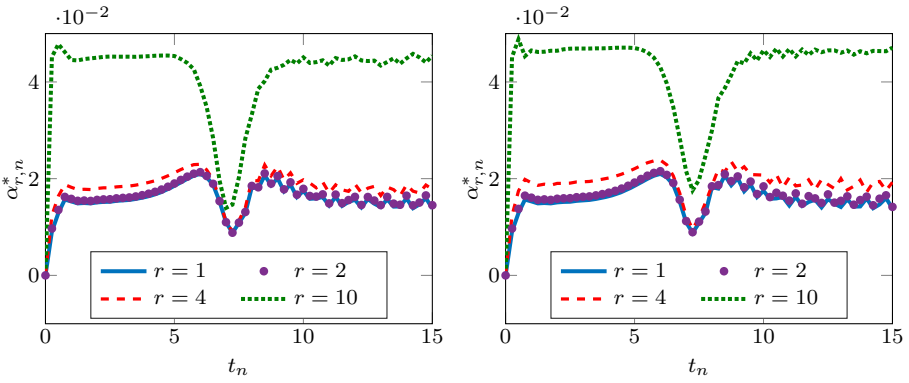

Fig. 5 Two-soliton problem for KdV (16). Sequence of parameters $\{\alpha^*_{r,n}\}_{n\geq 0}$ for EC(α), (19), obtained using Algorithms 1 (left) and 2 (right) with different values of r

$\mathbb{R}$,

$$u(x,t) = \frac{12(c_1 - c_2)(c_1 \cosh^2 \xi_2 + c_2 \sinh^2 \xi_1)}{\left((\sqrt{c_1} - \sqrt{c_2}) \cosh(\xi_1 + \xi_2) + (\sqrt{c_1} + \sqrt{c_2}) \cosh(\xi_1 - \xi_2)\right)^2},$$

with

$$\xi_1 = \frac{c_1}{2}(x + d_1 - c_1 t), \quad \xi_2 = \frac{c_2}{2}(x + d_2 - c_2 t).$$

We set

$$c_1 = 2, \qquad c_2 = 1, \qquad d_1 = 17, \qquad d_2 = 10, \qquad \Delta x = 0.05, \qquad \Delta t = 0.25.$$

In Figs. 5 and 6 we show the sequences of parameters obtained at each time step of schemes EC and MC, respectively. Again, all the sequences obtained by solving the optimization problems on a grid up to four times coarser are very close (within a distance of $6 \cdot 10^{-3}$) to those obtained on the computational grid.

We notice that the parameters rapidly change when the solitons interact, but only slowly variate before and after the interaction.

In Table 2 we compare the accuracy, efficiency and conservation properties of different schemes. The same observations done for the case of the motion of a solitary wave hold also in this case. As before, the computational times of the new algorithms with $r = 4$ are comparable to those of the methods from the literature, and their greater efficiency is evident through solution errors that are much lower.

In the left plot of Fig. 7, we show the initial condition together with the solution of the sequence MC$(\beta^*_{4,n}, \gamma^*_{4,n})_{n\geq 0}$. In the right plot, we show the pointwise error of MC$(\beta^*_{4,n}, \gamma^*_{4,n})_{n\geq 0}$, MC(0,0), and MC$(\beta^*, \gamma^*)$ at the final time. We omit the results for the multisymplectic and the narrow box scheme, as they are similar to that of MC(0,0). The methods obtained using the approaches introduced in this paper are very accurate around the final location of the faster soliton, where the bulk of the error given by MC$(0, 0)$ and by the schemes from the literature is located. However, the error around the slower soliton is larger and small oscillations can be seen far from the solitons. MC(β^*, γ^*) gives a slightly smaller error around the slower soliton, but more oscillations are introduced.

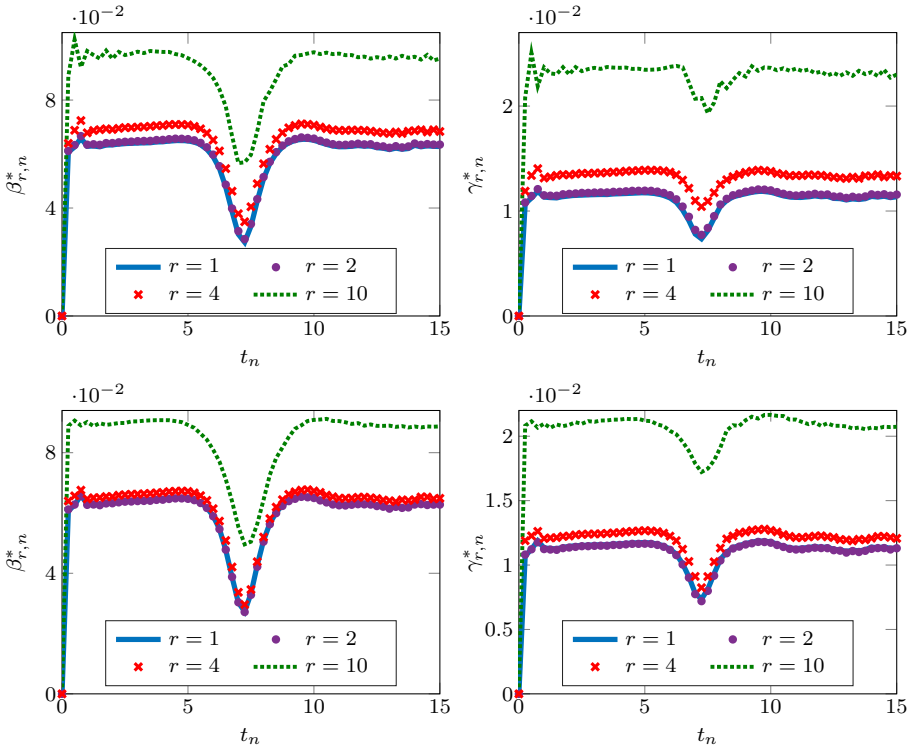

Fig. 6 Two-soliton problem for KdV (16). Sequence of parameters $\{(\beta^*_{r,n}, \gamma^*_{r,n})\}_{n\geq 0}$ for MC(β, γ), (22), obtained using Algorithms 1 (left) and 2 (right) with different values of r

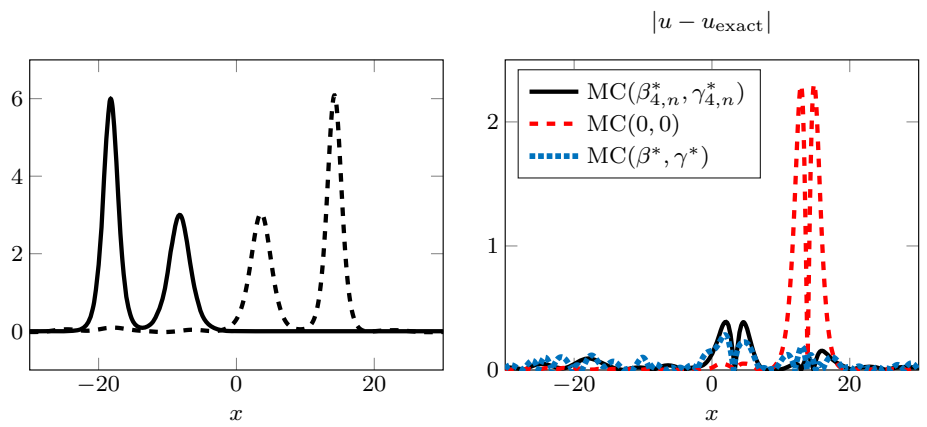

Fig. 7 Two-soliton problem for KdV (16). Left: Initial condition (dashed line) and solution of MC$(\beta^*_{4,n}, \gamma^*_{4,n})_{n\geq 0}$. Right: Pointwise error for different schemes at the final time

Table 2 Two-soliton problem for KdV (16)

Method	Err_1	Err_2	Err_3	Sol. err	Time
$\mathrm{EC}(\alpha^*_{1,n})_{n\geq 0}$	2.81e−13	0.0490	4.79e−12	0.1898	364.25
$\mathrm{EC}(\alpha^*_{2,n})_{n\geq 0}$	2.10e−13	0.0453	5.91e−12	0.1856	127.11
$\mathrm{EC}(\alpha^*_{4,n})_{n\geq 0}$	6.57e−13	0.0287	9.82e−12	0.1664	85.69
$\mathrm{EC}(\alpha^*_{10,n})_{n\geq 0}$	3.30e−13	0.1439	4.04e−12	0.1275	84.41
$\mathrm{EC}(\overline{\alpha}^*_1) \equiv \mathrm{EC}(0.016)$	5.90e−13	0.0697	5.85e−12	0.1897	474.40
$\mathrm{EC}(\overline{\alpha}^*_2) \equiv \mathrm{EC}(0.017)$	1.99e−13	0.0637	2.56e−12	0.1841	166.42
$\mathrm{EC}(\overline{\alpha}^*_4) \equiv \mathrm{EC}(0.019)$	5.12e−13	0.0384	6.12e−12	0.1604	97.08
$\mathrm{EC}(\overline{\alpha}^*_{10}) \equiv \mathrm{EC}(0.043)$	8.95e−13	0.1768	1.45e−11	0.1250	93.81
$\mathrm{MC}(\beta^*_{1,n}, \gamma^*_{1,n})_{n\geq 0}$	1.21e−12	0.4128	0.9188	0.0834	416.18
$\mathrm{MC}(\beta^*_{2,n}, \gamma^*_{2,n})_{n\geq 0}$	2.06e−12	0.4187	0.9183	0.0825	175.44
$\mathrm{MC}(\beta^*_{4,n}, \gamma^*_{4,n})_{n\geq 0}$	2.49e−12	0.4404	0.9165	0.0812	94.32
$\mathrm{MC}(\beta^*_{10,n}, \gamma^*_{10,n})_{n\geq 0}$	3.24e−12	0.5410	0.9082	0.1252	86.30
$\mathrm{MC}(\overline{\beta}^*_1, \overline{\gamma}^*_1) \equiv \mathrm{MC}(0.060, 0.011)$	1.49e−12	1.66e−12	0.9269	0.0874	480.98
$\mathrm{MC}(\overline{\beta}^*_2, \overline{\gamma}^*_2) \equiv \mathrm{MC}(0.061, 0.011)$	2.38e−12	1.85e−11	0.9266	0.0873	177.06
$\mathrm{MC}(\overline{\beta}^*_4, \overline{\gamma}^*_4) \equiv \mathrm{MC}(0.062, 0.012)$	1.15e−12	1.48e−11	0.9256	0.0856	92.42
$\mathrm{MC}(\overline{\beta}^*_{10}, \overline{\gamma}^*_{10}) \equiv \mathrm{MC}(0.085, 0.021)$	4.22e−12	1.39e−11	0.9198	0.0918	86.05
EC(0)	4.44e−13	0.2186	5.44e−12	0.3208	78.11
MC(0, 0)	2.03e−13	6.04e−13	0.8567	0.3884	74.59
$\mathrm{EC}(\alpha^*) \equiv \mathrm{EC}(0.034)$	7.07e−13	0.0951	1.01e−11	0.0683	N/A
$\mathrm{MC}(\beta^*, \gamma^*) \equiv \mathrm{MC}(0.147, 0.065)$	7.90e−12	7.10e−11	0.8318	0.0689	N/A
Multisymplectic	2.91e−12	0.0081	0.8807	0.3885	94.28
Narrow box	4.50e−12	0.0041	0.8160	0.3825	82.44

Errors in solution and conservation laws, and computation time

5.2 Nonlinear Heat Equation

In this section we apply the new algorithms to the nonlinear heat Eq. (25) using the methods CS(λ) from Sect. 4.2.

We consider here two benchmark problems with (weak) energy solutions that are not classic solutions, having at least one point of non differentiability. Although in Sect. 3 smoothness of the solution is assumed, we show that the strategies introduced in this paper are also effective in this setting.

In order to converge, explicit and implicit finite difference methods for (25) in literature typically require $\Delta t = \mathcal{O}(\Delta x^2)$ and $\Delta t = \mathcal{O}(\Delta x)$, respectively [9, 10, 18, 19, 22]. Under such small time-step restrictions, the Crank-Nicolson method applied to the semidiscretization (31) turns out to be very accurate and efficient. However, it fails to converge for the two benchmark tests in this section with $\Delta t > \Delta x$. Such instabilities may also occur when using a CS(λ) method with a default choice of the parameter, $\lambda = 0$. In contrast, we find that the two proposed algorithms in Sect. 3, based on optimization of defect based error estimate, are able to avoid these instabilities.

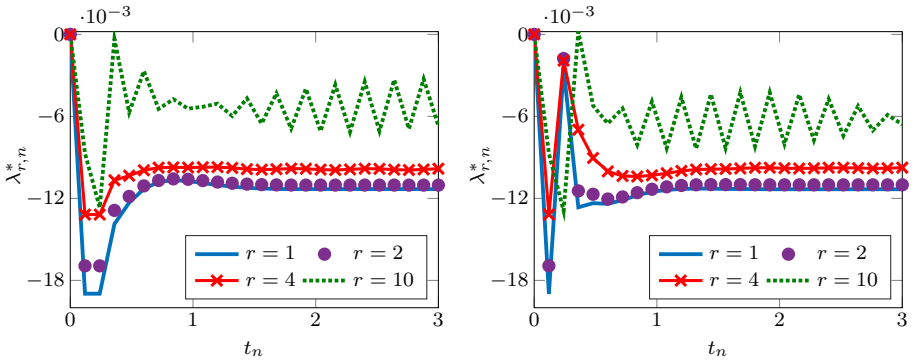

Fig. 8 NLH (25) with initial and boundary conditions (35). Sequence of parameters $\{\lambda^*_{r,n}\}_{n\geq 0}$ for CS(λ), (28), obtained using Algorithms 1 (left) and 2 (right) with different values of r

The first benchmark problem is given by Eq. (25) with initial and boundary conditions,

$$u(x,0) = 0, \qquad u(0,t) = t, \qquad u(5,t) = 0, \qquad (x,t) \in [0,5] \times [0,3]. \tag{35}$$

The solution of this problem is a linear wave travelling in an undisturbed medium with unit speed,

$$u_{\text{exact}}(x,t) = \begin{cases} t - x, & \text{if } t > x, \\ 0, & \text{otherwise.} \end{cases}$$

We discretize the initial boundary value problem described by (25) and (35), choosing $\Delta x = 0.025$ and $\Delta t = 0.12$. The graphs in Fig. 8 show the sequences of parameters obtained from Algorithms 1 and 2 at each time step, where the search of the optimal parameters is carried out after applying the projection (14) with spatial resolutions of $r\Delta x$ with $r = 1, 2, 4, 10$. Although the values of the parameters given by the two algorithms are different for the first time-steps, for the smallest values of r the two procedures converge to the same value.

In Table 3 we compare the different choices for the parameter λ. The error in the conservation laws is calculated according to (33) and the figures in the table show that these are preserved to machine accuracy by all the methods that use a fixed value of the free parameter.

The two algorithms introduced in this paper give equally accurate solutions. By increasing r, the computation time decreases, while the accuracy in the solution is only marginally affected. Moreover, both new algorithms avoid instabilities that occur when $\lambda = 0$. This is shown on the left of Fig. 9 where, as an example, we show the solution of CS($\overline{\lambda}^*_4$) (marks at every tenth grid point) and CS(0) at the final time.

On the right of Fig. 9, we plot the pointwise errors given by CS(λ^*) and CS($\overline{\lambda}^*_4$). The error obtained with $\lambda = \overline{\lambda}^*_4$ is almost entirely located at the interface where the solution is non differentiable. Fixing $\lambda = \lambda^*$, the L^2 error is lower and the solution is more accurate around the interface. However, spurious oscillations are seen where the true solution is smooth.

The second benchmark problem is (25) with

$$u(x,0) = \left(1 - \frac{x^2}{6}\right)_+, \qquad u(-6,t) = u(6,t) = 0, \qquad (x,t) \in [-6,6] \times [0,9], \tag{36}$$

Table 3 NLH (25) with initial and boundary conditions (35)

λ	Err_1	Err_2	Solution error	Time
$\{\lambda^*_{1,n}\}_{n\geq 0}$	0.1258	0.0190	0.0054	0.136
$\{\lambda^*_{2,n}\}_{n\geq 0}$	0.1182	0.0169	0.0052	0.069
$\{\lambda^*_{4,n}\}_{n\geq 0}$	0.1030	0.0132	0.0047	0.055
$\{\lambda^*_{10,n}\}_{n\geq 0}$	0.0817	0.0127	0.0027	0.035
$\overline{\lambda}^*_1 = -0.0115$	1.18e−13	2.98e−14	0.0054	0.122
$\overline{\lambda}^*_2 = -0.0110$	8.06e−14	3.20e−14	0.0052	0.077
$\overline{\lambda}^*_4 = -0.0096$	1.15e−13	4.48e−14	0.0046	0.046
$\overline{\lambda}^*_{10} = -0.0064$	6.68e−14	5.12e−14	0.0031	0.028
$\lambda = 0$	6.54e−14	9.24e−15	7.5017	0.016
$\lambda^* = -0.0044$	9.24e−14	3.27e−14	0.0023	N/A

Errors in solution and conservation laws and computation time

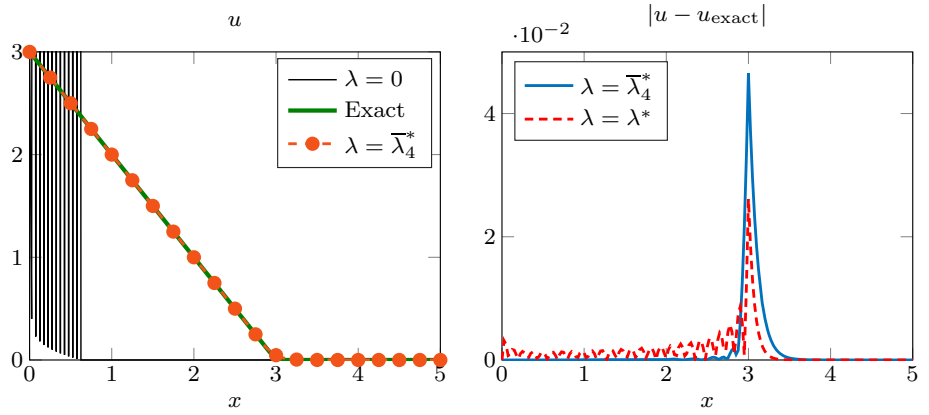

Fig. 9 NLH (25) with initial and boundary conditions (35). Exact and numerical solutions of CS(λ), (28), with $\lambda = 0$ and $\lambda = \overline{\lambda}^*_4$ (left). Solution error for $\lambda = \overline{\lambda}^*_4$ and $\lambda = \lambda^*$ (right)

where $f_+ = \max(f, 0)$. The solution of this problem is the Barenblatt profile,

$$u_{\text{exact}}(x,t) = (t+1)^{-1/3}\left(1 - \frac{x^2}{6(t+1)^{2/3}}\right)_+ .$$

This solution has compact support and is not differentiable at the interface points, which move outward at a finite speed. We solve this problem with $\Delta x = 0.02$ and $\Delta t = 0.09$.

We show in Fig. 10 that in this case the two proposed algorithms generate sequences of parameters that approach a small negative value for all the considered values of r.

Table 4 shows that all the schemes preserve a discrete version of the global invariants. When the value of the parameter is fixed during the iteration, this is a consequence of the preservation of the local conservation laws. When a sequence of different parameters is used, this is instead due to the zero boundary conditions (see Remark 10). Although in this case the conservation laws are not preserved locally, the error in the solution is the lowest.

Table 4 NLH (25) with initial and boundary conditions (36)

λ	Err_1	Err_2	Solution error	Time
$\{\lambda^*_{1,n}\}_{n\geq 0}$	2.00e−13	2.55e−14	1.82e−04	8.18
$\{\lambda^*_{2,n}\}_{n\geq 0}$	1.41e−13	2.43e−14	1.41e−04	1.73
$\{\lambda^*_{4,n}\}_{n\geq 0}$	3.43e−13	3.28e−14	1.45e−04	1.51
$\{\lambda^*_{10,n}\}_{n\geq 0}$	3.62e−13	3.20e−14	1.15e−04	1.35
$\overline{\lambda}^*_1 = -6.81\text{e}{-04}$	5.05e−14	3.05e−14	5.20e−04	5.88
$\overline{\lambda}^*_2 = -5.64\text{e}{-04}$	2.70e−14	2.72e−14	4.53e−04	1.64
$\overline{\lambda}^*_4 = -4.38\text{e}{-04}$	2.79e−14	3.10e−14	3.78e−04	1.26
$\overline{\lambda}^*_{10} = -3.86\text{e}{-04}$	2.36e−14	2.93e−14	3.45e−04	1.14
$\lambda = 0$	3.59e−14	2.64e−14	0.2989	0.59
$\lambda^* = -2.32\text{e}{-04}$	3.85e−14	3.61e−14	2.62e−04	N/A

Errors in solution and conservation laws and computation time

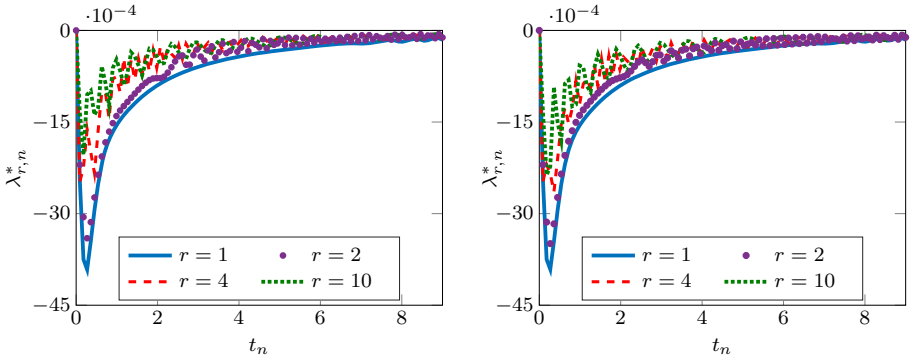

Fig. 10 NLH (25) with initial and boundary conditions (35). Sequence of parameters $\{\lambda^*_{r,n}\}_{n\geq 0}$ for CS(λ), (28), obtained using Algorithms 1 (left) and 2 (right) with different values of r

The figures in Table 4 show that with increasing r the computation time decreases while the accuracy is not significantly affected even for $r = 10$. This is a consequence of the fact that the sequence of parameters obtained for different values of r quickly approach each other after just a few time-steps (see Fig. 10). Once again, we find in Fig. 11 (left) that the sequence of parameters $\{\lambda^*_{4,n}\}_{n\geq 0}$ obtained from Algorithm 1 (marks at every tenth grid point) allows us to avoid the numerical instability that occurs under the default choice, $\lambda = 0$.

On the right half of Fig. 11 we show the solution error of CS($\lambda^*_{4,n}$)$_{n\geq 0}$ and of CS(λ^*). The solution error obtained using the sequence of parameters given by Algorithm 1 is mainly located at the points where the solution is not differentiable. Although $\lambda = \lambda^*$ minimizes the L^2 norm of the error with respect to any other fixed value of λ, we notice that the error at the interface can be further reduced by changing the parameter at every time-step. Moreover, for $\lambda = \lambda^*$, relatively large components of error appear near ± 2.5 where the solution is otherwise smooth. These are not visible in the solution errors obtained using either the adaptive sequences given by Algorithm 1 or the fixed parameters given by Algorithm 2.

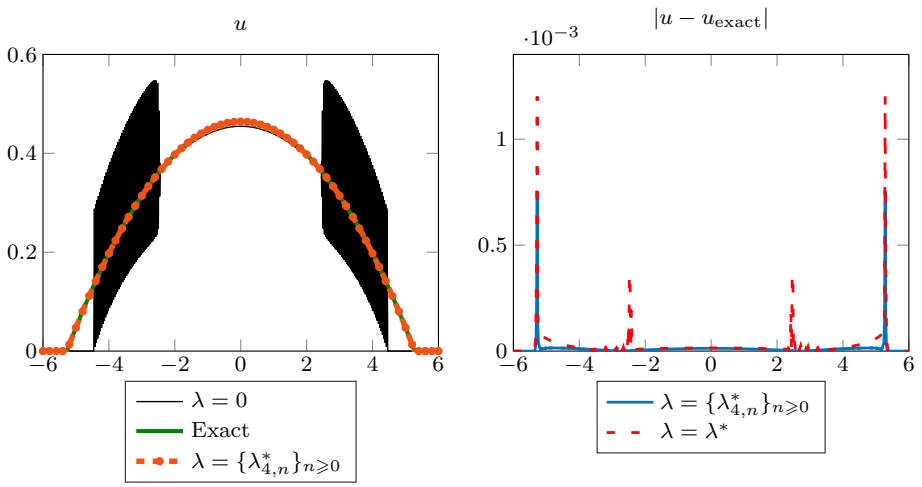

Fig. 11 NLH (25) with initial and boundary conditions (36). Exact and numerical solutions of CS(λ), (28), with $\lambda = 0$ and $\lambda = \{\lambda^*_{4,n}\}_{n\geq 0}$ (left). Solution error for $\lambda = \lambda = \{\lambda^*_{4,n}\}_{n\geq 0}$ and $\lambda = \lambda^*$ (right)

6 Conclusions

In this paper we have proposed two approaches for identifying an optimal method in a parameter dependent family of numerical schemes, based on a minimization of the defect as an estimate of the local error. The first approach uses different (adaptive) values of the parameters at every time-step. In the second approach fixed values of the parameters are derived from a sequence. The latter approach does not compromise parameter depending conservation properties of geometric integrators.

The new algorithms solve an optimization problem at each time-step in order to identify the optimal values of the parameter. In principle, this can increase the computational cost of the original method prohibitively. However, in the large time-step regime, it is possible to solve the optimization problem on coarser spatial grids without compromising the accuracy of the optimal parameters, significantly decreasing the computational time.

The new approaches have been applied to families of schemes for the KdV equation and a nonlinear heat equation that preserve local conservation laws. The proposed numerical tests show that, on one hand, the new strategies effectively identify very accurate methods in each considered family of schemes. On the other hand, introducing a coarse grid for the solution of the optimization problem tremendously improves the efficiency of the new strategies. Overall, the computational time is comparable to that of other schemes in literature, while the accuracy of the proposed approach is much superior.

Acknowledgements The authors would like to thank the Isaac Newton Institute for Mathematical Sciences for support and hospitality during the programme Geometry, Compatibility and Structure Preservation in Computational Differential Equations, when work on this paper was undertaken. This work was supported by EPSRC grant number EP/R014604/1. The first author is member of the INdAM Research group GNCS.

Funding Open access funding provided by Università degli Studi di Salerno within the CRUI-CARE Agreement.

Data Availability All data generated or analyzed during this study are included in this manuscript.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ascher, U.M., McLachlan, R.I.: Multisymplectic box schemes and the Korteweg-de Vries equation. Appl. Numer. Math. **48**, 255–269 (2004). https://doi.org/10.1016/j.apnum.2003.09.002
2. Ascher, U.M., McLachlan, R.I.: On symplectic and multisymplectic scheme for the KdV equation. J. Sci. Comput. **25**, 83–104 (2005). https://doi.org/10.1007/s10915-004-4634-6
3. Auzinger, W., Hofstätter, H., Koch, O., Kropielnicka, K., Singh, P.: Time adaptive Zassenhaus splittings for the Schrödinger equation in the semiclassical regime. Appl. Math. Comput. **362**, 124550 (2019). https://doi.org/10.1016/j.amc.2019.06.064
4. Auzinger, W., Hofstätter, H., Koch, O.: Symmetrized local error estimators for time-reversible one-step methods in nonlinear evolution equations. J. Comput. Appl. Math. **356**, 339–357 (2019). https://doi.org/10.1016/j.cam.2019.02.011
5. Auzinger, W., Koch, O., Thalhammer, M.: Defect-based local error estimators for splitting methods, with application to Schrödinger equations, Part I: the linear case. J. Comput. Appl. Math. **236**(10), 2643–2659 (2012). https://doi.org/10.1016/j.cam.2012.01.001
6. Auzinger, W., Koch, O., Thalhammer, M.: Defect-based local error estimators for splitting methods, with application to Schrödinger equations, Part II: higher-order methods for linear problems. J. Comput. Appl. Math. **255**, 384–403 (2013). https://doi.org/10.1016/j.cam.2013.04.043
7. Berljafa, M., Güttel, S.: The RKFIT algorithm for nonlinear rational approximation. SIAM J. Sci. Comput. **39**, A2049–A2071 (2017). https://doi.org/10.1137/15M1025426
8. Blanes, S., Casas, F., Murua, A.: Splitting and composition methods in the numerical integration of differential equations. Bol. Soc. Esp. Mat. Apl. **45**, 89–145 (2008). https://doi.org/10.1016/j.cam.2010.06.018
9. Del Teso, F.: Finite difference method for a fractional porous medium equation. Calcolo **51**, 615–638 (2014). https://doi.org/10.1007/s10092-013-0103-7
10. Del Teso, F., Endal, J., Jakobsen, E.: Robust numerical methods for nonlocal (and local) equations of porous medium type. Part II: schemes and experiments. SIAM J. Numer. Anal. **56**, 3611–3647 (2018). https://doi.org/10.1137/18M1180748
11. Descombes, S., Thalhammer, M.: The Lie-Trotter splitting method for nonlinear evolutionary problems involving critical parameters. An exact local error representation and application to nonlinear Schrödinger equations in the semi-classical regime. IMA J. Numer. Anal. **33**, 722–745 (2013). https://doi.org/10.1093/imanum/drs021
12. Enright, W.: A new error-control for initial value solvers. Appl. Math. Comput. **31**, 288–301 (1989). https://doi.org/10.1016/0096-3003(89)90123-9
13. Frasca-Caccia, G., Hydon, P.E.: Locally conservative finite difference schemes for the modified KdV equation. J. Comput. Dyn. **6**, 162–179 (2019). https://doi.org/10.3934/jcd.2019015
14. Frasca-Caccia, G., Hydon, P.E.: Simple bespoke preservation of two conservation laws. IMA J. Numer. Anal. **40**, 1294–1329 (2020). https://doi.org/10.1093/imanum/dry087
15. Frasca-Caccia, G., Hydon, P.E.: Numerical preservation of multiple local conservation laws. Appl. Math. Comput. **403**, 126203 (2021). https://doi.org/10.1016/j.amc.2021.126203
16. Frasca-Caccia, G., Hydon, P.E.: A new technique for preserving conservation laws. Found. Comput. Math. **22**, 477–506 (2022). https://doi.org/10.1007/s10208-021-09511-1
17. Göckler, T., Grimm, V.: Uniform approximation of φ-functions in exponential integrators by a rational Krylov subspace method with simple poles. SIAM J. Matrix Anal. Appl. **35**(4), 1467–1489 (2014). https://doi.org/10.1137/140964655

18. Graveleau, J.L., Jamet, P.: A finite difference approach to some degenerate nonlinear parabolic equations. SIAM J. Appl. Math. **20**, 199–223 (1971). https://doi.org/10.1137/0120027
19. Gurtin, M.E., MacCamy, R.C., Socolovsky, E.: A coordinate transformation for the porous media equation that renders the free-boundary stationary. Quart. Appl. Math. **47**, 345–358 (1984). https://doi.org/10.1090/QAM/757173
20. Güttel, S.: Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection. GAMM-Mitteilungen **36**(1), 8–31 (2013). https://doi.org/10.1002/gamm.201310002
21. Higham, D.J.: Robust defect control with Runge–Kutta schemes. SIAM J. Numer. Anal. **26**(5), 1175–1183 (1989). https://doi.org/10.1137/0726065
22. Hoff, D.: A linearly implicit finite-difference scheme for the one-dimensional porous medium equation. Math. Comp. **45**, 23–33 (1985). https://doi.org/10.2307/2008047
23. Hydon, P.E., Mansfield, E.L.: A variational complex for difference equations. Found. Comput. Math. **4**, 187–217 (2004). https://doi.org/10.1007/s10208-002-0071-9
24. Kupershmidt, B.A.: Discrete Lax equations and differential-difference calculus, vol. 123. Société mathématique de France (1985)
25. McLachlan, R.I., Quispel, G.R.W.: Splitting methods. Acta Numer. **11**, 341–434 (2002). https://doi.org/10.1017/S0962492902000053
26. Nocedal, J., Wright, S.J.: Numerical Optimization, second edn. Springer, New York, NY, USA (2006). doi: https://doi.org/10.1007/978-0-387-40065-5
27. Olver, P.J.: Applications of Lie Groups to Differential Equations, vol. 107, 2nd edn. Springer Science & Business Media, New York (1993). https://doi.org/10.1007/978-1-4684-0274-2
28. Omelyan, I., Mryglod, I., Folk, R.: Symplectic analytically integrable decomposition algorithms: classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations. Comput. Phys. Commun. **151**(3), 272–314 (2003). https://doi.org/10.1016/S0010-4655(02)00754-3
29. Schaback, R.: Convergence analysis of the general Gauss-Newton algorithm. Numer. Math. **46**, 281–309 (1985). https://doi.org/10.1007/BF01390425
30. Shampine, L.F.: Error estimation and control for ODEs. J. Sci. Comput. **25**, 3–16 (2005). https://doi.org/10.1007/s10915-004-4629-3
31. Singh, P.: Sixth-order schemes for laser-matter interaction in the Schrödinger equation. J. Chem. Phys. **150**(15), 154111 (2019). https://doi.org/10.1063/1.5065902