Higher-order explicit schemes based on the Method of characteristics for hyperbolic equations with crossing straight-line characteristics

T.I. Lakoba*, J.S. Jewell

Department of Mathematics and Statistics,
University of Vermont, Burlington, VT 05405, USA

January 27, 2021

Abstract

We develop Method of characteristics schemes based on explicit Runge–Kutta and pseudo-Runge–Kutta third- and fourth-order solvers along the characteristics. Schemes based on Runge–Kutta solvers are found to be strongly unstable for certain physics-motivated models. In contrast, schemes based on pseudo-Runge–Kutta solvers are shown to be only weakly unstable for periodic boundary conditions and essentially stable for the more physically relevant nonreflecting boundary conditions. Our implementation of nonreflecting boundary conditions does not rely on interpolation.

Keywords: Method of characteristics, Coupled-mode equations, Higher-order methods.

*tlakoba@uvm.edu, 1 (802) 656-2610

1 Introduction

In this work we will develop higher-order explicit numerical schemes that use the Method of characteristics (MoC) to solve systems of hyperbolic partial differential equations of the form:

$$\mathbf{y}^+_t + \mathbf{y}^+_x = \mathbf{f}^+\left(\mathbf{y}^+, \mathbf{y}^-\right), \qquad \mathbf{y}^-_t - \mathbf{y}^-_x = \mathbf{f}^-\left(\mathbf{y}^+, \mathbf{y}^-\right). \tag{1a}$$

Here $\mathbf{y}^\pm$ and $\mathbf{f}^\pm$ are vectors of respective lengths $N^\pm$ and subscripts denote partial differentiation. Functions $\mathbf{f}^\pm$ are, in general, nonlinear. Moreover, they, in principle, may contain small diffusion-like terms $\mathbf{y}^\pm_{xx}$. We will briefly comment on the latter possibility in the concluding section of this work, but in the main part of it we will assume that system (1a) is non-dissipative.

Systems of the form (1a) can be obtained by a simple change of variables from a slightly more general system

$$\mathbf{y}_t + \mathbf{A}\mathbf{y}_x = \mathbf{f}\left(\mathbf{y}\right), \tag{1b}$$

where vectors $\mathbf{y}$ and $\mathbf{f}$ are obtained by stacking, respectively, $\mathbf{y}^+$ with $\mathbf{y}^-$ and $\mathbf{f}^+$ with $\mathbf{f}^-$, and the diagonalization of $\mathbf{A}$ is

$$c_1\mathbf{I}_{(N^+ + N^-)} + c_2\mathbf{\Sigma}, \qquad \text{with} \quad \mathbf{\Sigma} \equiv \mathrm{diag}(\mathbf{I}_{N^+}, -\mathbf{I}_{N^-}), \tag{1c}$$

$\mathbf{I}_N$ being the $N \times N$ identity matrix, and $c_{1,2}$ being some constants. Equations of the form (1a) or (1b) describe physical models where there are two groups of waves that propagate with constant and different velocities $c_\pm = c_1 \pm c_2$, in the notations of (1c). An extensive list of physical applications in linear and nonlinear optics, plasma physics, atomic physics, and relativistic field theory where such models arise is found in Section 2 of [1] (Refs. [4,5,20–29,44–64] there); see also Refs. [2]–[13] here and Ref. [14] about applications to power transmission lines. Let us note that the Klein–Gordon equation, which serves as a simplified model for many dispersive wave problems:

$$u_{xx} - c^{-2}u_{tt} = g(u, u_x, u_t), \tag{2}$$

where $c = \mathrm{const}$ and g is some differentiable function of its arguments, can also be written in form (1b) [15] with $N^+ = N^- = 2$. Numerical solution of various extensions of this well-studied model has recently attracted atention [16, 17, 18]. The schemes that we propose below are applicable to all of those extension.

In the framework of the MoC, each of the two (systems of) equations in (1a) is transformed, by the respective change of variables

$$(x, t) \to (\xi^\pm, t), \qquad \xi^\pm = x \mp t, \tag{3a}$$

to an ordinary differential equation (ODE) in t, and then is solved by an ODE numerical solver along the respective characteristic:

$$\mathbf{y}_t^{\pm} = \mathbf{f}^{\pm} \qquad \text{along} \;\; \xi^{\pm} = \text{const.} \tag{3b}$$

Therefore, the accuracy of an MoC-based scheme follows that of the ODE solver. Our goal is to develop schemes of order higher than two for the *quasi*-ODE system (3). We emphasize the word 'quasi': the fact that each of the ODEs in (3b) is solved along a different characteristic introduces several nontrivial modifications to the standard procedure of solving a system of (true) ODEs numerically.

Before presenting a motivation for this work, it is important to state its *limitations*. First, even though the body of physical applications described by Eqs. (1) is substantial, many hyperbolic systems in fluid and gas dynamics, where the propagation speed depends on the wave amplitude, cannot be described by (1a). Thus, for them, the schemes that we will develop in this work, cannot be applied without modification. For such systems, a well-studied family of semi-Lagrangian methods uses interpolation across a given time level to account for the grid "distortion" caused by the characteristics' curvature. Many highly accurate interpolation [19, 20] and characteristic-backtracking [21] schemes have been used primarily in applications of semi-Lagrangian methods to problems with *one* characteristic. We think that it should be possible (albeit not straightforward) to combine these techniques with the higher-order schemes which we develop here for *two crossing* characteristics; this is a topic for future work. The second limitation is that our approach requires a substantial modification for problems in multiple spatial dimensions; see Section 9 for more detail.

In this and the next paragraph we present the motivation for this work. For long-time simulations of system (1), it is essential that the numerical scheme distort its asymptotic dispersion relation

$$\omega = \pm k, \qquad |k| \gg 1 \tag{4}$$

as little as possible; here k and ω are the wavenumber and frequency of Fourier modes $\mathbf{y}^{\pm} \sim \exp[i(kx - \omega t)]$. The reason is that in schemes distorting (4), high-k Fourier modes propagate with incorrect group velocities and also experience spuriously high reflection from the boundaries back into the computational domain, thereby contaminating the solution (see, e.g., [22]). Most finite-difference and collocation-type schemes do not preserve the asymptotic dispersion relation (4) even approximately; see, e.g., [23, 24], whereas MoC-based schemes for systems with straight-line characteristics preserve it inherently (see, e.g., [23, 1]). This is their main advantage over finite-difference and collocation-type methods. (As a result, MoC-based schemes are able to compute solutions with steep front very accurately; see Section 8.)

The main disadvantage of MoC-based schemes is that until now, only first- and second-order accurate explicit MoC schemes for systems with crossing characteristics, such as (1a), have been known [23, 25]. There exist fourth-order MoC schemes based on implicit Runge–Kutta (RK) [26] and multistep [27] solvers for (1a). They are implemented as multi-stage predictor–corrector methods and hence are, as far as implementation is concerned, explicit; yet, it is expected that a method based on an explicit solver of the same order would be faster.

It is the purpose of this work to address the aforementioned disadvantage of MoC schemes and develop 3rd- and 4th-order methods based on explicit ODE solvers for systems (1a). In addition to the obvious benefit that a higher-order method is more accurate (or faster for the same accuracy) than a lower-order one, a higher-order method is also expected to *automatically* provide the following benefit. In many physical applications, systems of the form (1a) describe waves that propagate without dissipation. Therefore, a desirable property of a numerical scheme would be to introduce as little dissipation or energy gain as possible. The only explicit ODE solver that introduces no dissipation is the leap-frog (or any of its equivalent forms). However, when used with the MoC framework, it produces a strongly unstable method for energy-preserving systems of the form (1a) [1]. Other explicit ODE solvers, such as RK, introduce either small dissipation or gain; the amount of this dissipation/gain for non-stiff ODEs is known to decrease as the order of the solver increases. Therefore, one should expect that an MoC-based scheme using a high-order ODE solver will preserve the energy of the system (concentrated at low wavenumbers) better than a lower-order scheme. This is the expected additional benefit of 3rd- and 4th-order accurate MoC-based schemes, mentioned above. As for high wavenumbers, their temporal evolution is analyzed separately in this work.

A convergent numerical scheme must be not only accurate (consistent), but also stable. Therefore, in this and the next three paragraphs, we will outline our approach to analyzing the stability of an MoC-based scheme. The standard way is to apply the von Neumann stability analysis to the linearized version of (1a):

$$\widetilde{\mathbf{y}}_t + \mathbf{\Sigma}\,\widetilde{\mathbf{y}}_x = \mathbf{P}\,\widetilde{\mathbf{y}}, \tag{5a}$$

where $\widetilde{\mathbf{y}}$ is a small deviation of the solution from some constant-in-x solution $\mathbf{y}_0$, $\mathbf{\Sigma}$ is defined in (1c), and $\mathbf{P} = \partial\mathbf{f}/\partial\mathbf{y}|_{\mathbf{y}=\mathbf{y}_0}$ is the Jacobian matrix. For future reference, we mention that it has the structure

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}^{++} & \mathbf{P}^{+-} \\ \mathbf{P}^{-+} & \mathbf{P}^{--} \end{pmatrix}, \tag{5b}$$

where $\mathbf{P}^{+-} = \partial\mathbf{f}^{+}/\partial\mathbf{y}^{-}|_{\mathbf{y}=\mathbf{y}_0}$ etc. For schemes based on the Method of lines, the outcome of the von Neumann analysis depends on the scheme itself and on the eigenvalues of $\mathbf{P}$; therefore, in that case, applying the von Neumann analysis to a scalar equation would suffice. This, however, is not

so for MoC-based schemes, because matrix $\mathbf{\Sigma}$ also enters the equation determining the amplification factor of the scheme. This was shown for first- and second-order MoC-based schemes in [1] and will be reported for third-order schemes in Sections 3.2 and 6.2. *Thus, it would be incorrect to perform a stability analysis of an MoC-based scheme based only on the eigenvalues of the Jacobian matrix* $\mathbf{P}$. Therefore, below we present an alternative approach that we followed.

We limited our numerical stability analysis of system (5) only to those $\mathbf{P}$ for which that system is stable *physically*; i.e., its solutions of the form $\widetilde{\mathbf{y}} \propto \exp[ikx - i\omega t]$ with all $k = O(1)$ do not grow in time exponentially. When $N^+ = N^- = 1$ (see (1c)), it is straightforward to find all $\mathbf{P}$ for which (5) is physically stable. However, already for $N^+ = N^- = 2$, the problem of determining all possibilities for $\mathbf{P}$ where (5a) is physically stable becomes nontrivial. We do not address it here but instead adopt the following simplified approach. We consider a certain family of models from nonlinear optics whose linearization leads to a system (5a) with over 30 different forms of $\mathbf{P}$ with $N^+ = N^- = 2$. Of those, we identify 4 *distinct* (i.e., not reducible to one another) $\mathbf{P}$'s that lead to a physically stable (5a). In addition, the linearization of the Klein–Gordon equation with $g \equiv u$ and written in the form (1b) (see Eqs. (2.6)–(2.8) in [15]), yields one more $\mathbf{P}$. Thus, we have collected a "bank" of 5 different $\mathbf{P}$'s for which (5a) is physically stable, and which are listed in Section 2. We then apply the von Neumann analysis to system (5a) with $\mathbf{\Sigma} = \mathrm{diag}(\mathbf{I}_2, -\mathbf{I}_2)$ for each of these five $\mathbf{P}$'s. We will declare a given MoC scheme to be numerically stable (or weakly unstable; see Section 3) if the von Neumann analysis reveals no instability (or, respectively, only weak instability) for all these five $\mathbf{P}$'s, as well as for all cases with $N^+ = N^- = 1$. Admittedly, such an analysis cannot predict numerical instability of a scheme for *all* possible cases of (5a); however, it does so for all *physically stable cases known to us*. For any other case, the same analysis as outlined in Section 3 (and originally presented in [1] for lower-order MoC schemes) will determine the von Neumann stability of the scheme.

In the previous paragraph we mentioned that the von Neumann analysis may reveal weak numerical instability of a scheme. This means that for sufficiently high (but not necessarily the highest) wavenumbers, the growth rate of the numerical error is $O(h)$, where h is the discretization step in space and time. As was demonstrated in [28], nonreflecting BC applied to low-order MoC-based schemes can *suppress* the weak numerical instability. We found, by numerical simulations, that the same conclusion also holds for the higher-order MoC schemes that are determined to be weakly unstable by the von Neumann analysis. On the other hand, if a scheme is found to be strongly unstable (with the growth rate $O(1)$) by the von Neumann analysis, then it remains such for nonreflecting BC.

Therefore, after presenting each of the new higher-order MoC schemes, we will report for it the results of von Neumann analysis and accordingly classify the scheme as weakly or strongly unstable (for periodic BC). Then, in Sections 3 and 7, we demonstrate that nonreflecting BC indeed suppress weak, but not strong, numerical instability. When weak instability for wavenumbers satisfying $1 \ll |k| \ll k_{\max}$ is suppressed by nonreflecting BC, there still remains a much weaker numerical instability for $|k| \ll 1$ (corresponding to the ODE limit of (5a)) and for $|k| \approx k_{\max}$. The growth rate of that instability decreases with the ODE solver's order: for 2nd- and 3rd-order solvers it is $O(h^3)$; for the 4th-order (and a 5th-order, not considered here), it is $O(h^5)$; etc. We will ignore this very weak instability, as it can only affect results of ultra-long time simulations.

The organization of the main part of this work is as follows. In Section 2, we present the explicit forms of matrix $\mathbf{P}$ for which we will perform the von Neumann analysis of our new MoC schemes applied to system (5a). In Section 3, we will describe the construction of a MoC scheme with a third-order RK solver. In what follows we will refer to a MoC scheme that uses an 'XYZ' ODE solver as MoC-XYZ; thus, Section 3 presents details on MoC-RK3. It is shown that while the MoC-RK3 is only weakly unstable for periodic BC for $N^+ = N^- = 1$, it can be strongly unstable for $N^+ = N^- = 2$. Similar conclusions are reached about MoC-RK4 in Section 4. Therefore, in Section 5, we turn to a different family of ODE solvers, known as pseudo-RK (pRK) methods, which were first proposed in [29]. These can be thought of as a hybrid between RK and multistep methods. We also present modifications of pRK methods that can better preserve conserved quantities. Section 6 presents details on the construction and von Neumann stability analysis of MoC-pRK3 and MoC-pRK4 methods. Unlike MoC-RK schemes, the MoC-pRK ones are found to be only weakly unstable for periodic BC, and thus are expected to be stabilized by nonreflecting BC. In order to demonstrate this, one needs to develop an algorithm of imposing non-periodic BC in the MoC-pRK scheme. We present such algorithms in Section 7 and then verify that nonreflecting BC indeed suppress weak instability of the MoC-pRK schemes. In Section 8 we demonstrate that the accuracy of the developed MoC schemes indeed follows that of their respective ODE solvers. In Section 9, we summarize our results, discuss their extensions, and briefly compare our main result with that of [26].

Given the considerable length of the paper, we recommend that the reader who is interested only in the main idea of the MoC-(p)RK schemes *limit their reading* to: Sections 3.1, Figs. 2 and 3 and the paragraph immediately after it, Sections 5.1, 5.4, 6.1, and Figs. 7 and 8. Pseudocodes for the MoC-pRK schemes are presented in Appendix B and the actual codes, in Ref. [30].

2 Explicit forms of P in (5a)

As noted in the Introduction, we will consider only the cases $N^{\pm} = 1$ and $N^{\pm} = 2$, as these are the ones for which we are familiar with physical applications. We are interested only in those systems where the solution is physically stable, i.e. has no exponentially growing Fourier harmonics $\exp[ikx]$ for $k = O(1)$. In this section, we present only the forms of the corresponding $\mathbf{P}$ matrices, which for brevity we will refer to as "stable", along with brief comments. Details on how these matrices are obtained are in Appendix A. In what follows, we will use the common notation of 2×2 Pauli matrices:

$$\boldsymbol{\sigma}_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \boldsymbol{\sigma}_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \boldsymbol{\sigma}_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \boldsymbol{\sigma}_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}; \tag{6a}$$

as well as (uncommon) ones for matrices that will appear in 4×4 $\mathbf{P}$ matrices below:

$$\boldsymbol{\sigma}_4 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \frac{1}{2}(\boldsymbol{\sigma}_1 + i\boldsymbol{\sigma}_2), \quad \boldsymbol{\sigma}_5 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \frac{1}{2}(\boldsymbol{\sigma}_1 - i\boldsymbol{\sigma}_2), \quad \boldsymbol{\sigma}_6 = \boldsymbol{\sigma}_1 + \boldsymbol{\sigma}_4, \quad \boldsymbol{\sigma}_7 = i\boldsymbol{\sigma}_2 + \boldsymbol{\sigma}_4\,. \tag{6b}$$

In the $N^{\pm} = 1$ case, all physically stable $\mathbf{P}$'s can be reduced (see Appendix A) to $\mathbf{P} = i\,\boldsymbol{\sigma}_1$.

As explained in the Introduction, in the $N^{\pm} = 2$ case, we had to deduce stable $\mathbf{P}$'s from physical models. The first one comes from the linear Klein–Gordon equation (with $g \equiv u$ in (2)). Casting equations (2.6)–(2.8) of [15] into form (5a), one can show that in this case,

$$\mathbf{P}_1 = \frac{1}{2} \begin{pmatrix} \boldsymbol{\sigma}_5 & -\boldsymbol{\sigma}_7 \\ -\boldsymbol{\sigma}_7 & \boldsymbol{\sigma}_5 \end{pmatrix}. \tag{7}$$

The next model accounts for various forms of *linear* coupling among components of electromagnetic field in two waveguides (two optical fibers in close proximity to one another [2, 3] or two gratings with different periods in the same waveguide [4]). The corresponding stable $\mathbf{P}$ is:

$$\mathbf{P}_2 = i \begin{pmatrix} \boldsymbol{\sigma}_2 & a\,\boldsymbol{\sigma}_0 \\ a\,\boldsymbol{\sigma}_0 & \boldsymbol{\sigma}_2 \end{pmatrix}, \qquad a \in [0, \infty)\,. \tag{8}$$

The other stable matrices come from various models describing *nonlinear* coupling of co- and counter-propagating electromagnetic fields at distinct frequencies in optical fibers that support propagation of two modes (polarizations); see [6]–[13], [31, 32]. The corresponding stable matrices are:

$$\mathbf{P}_3 = i \begin{pmatrix} \boldsymbol{\sigma}_2 & \boldsymbol{\sigma}_2 \\ \boldsymbol{\sigma}_2 & \boldsymbol{\sigma}_2 \end{pmatrix} + i\,a \begin{pmatrix} \boldsymbol{\sigma}_2 & \mathbf{O} \\ \mathbf{O} & \boldsymbol{\sigma}_2 \end{pmatrix}, \qquad a \in [-3/2,\, 7/4]\,; \tag{9}$$

$$\mathbf{P}_4 = \begin{pmatrix} -i\boldsymbol{\sigma}_2 & -\boldsymbol{\sigma}_6 \\ \boldsymbol{\sigma}_6 & i\boldsymbol{\sigma}_2 \end{pmatrix} + a \begin{pmatrix} \boldsymbol{\sigma}_4 & \mathbf{O} \\ \mathbf{O} & -\boldsymbol{\sigma}_4 \end{pmatrix}, \qquad a \in [-1,\, 3/4]\,; \tag{10}$$

$$\mathbf{P}_5 = i \begin{pmatrix} \boldsymbol{\sigma}_2 & \boldsymbol{\sigma}_2 \\ \boldsymbol{\sigma}_2 & \boldsymbol{\sigma}_2 \end{pmatrix} + a \begin{pmatrix} \boldsymbol{\sigma}_4 & -2\boldsymbol{\sigma}_4 \\ -2\boldsymbol{\sigma}_4 & \boldsymbol{\sigma}_4 \end{pmatrix}, \qquad a \in [-1/2,\, 0]\,; \tag{11}$$

here and below $\mathbf{O}$ denotes the zero matrix of appropriate dimensions.

3 MoC schemes with RK3 solver

Here we will first derive a third-order accurate scheme based on an RK3 ODE solver. Then we will show that this scheme is strongly numerically unstable for some of the $\mathbf{P}$ matrices with $N^{\pm} = 2$. Finally, we will mention other versions of the same scheme; however, all of them will still be numerically unstable for the same $\mathbf{P}$ matrices. There are three reasons why we will dwell quite substantially on this unstable scheme:

- While describing its relatively simple setup, we will emphasize a number of *key points* that will repeatedly occur in subsequent schemes, some of which will be essentially stable;
- The MoC-RK3 scheme is the only one suitable to start the stable 4th-order scheme developed in Section 6;
- Finally, there are physically important systems for which the MoC-RK3 (and MoC-RK4, considered in Section 4) are stable.

3.1 Derivation of the scheme

The stencil for MoC-RK3 scheme for system (1a) is shown in Fig. 1. Discretization sizes in space and time are $\Delta x = \Delta t \equiv h$. Then at any time level $t_n = nh$, the characteristics $\xi^{\pm} = \text{const}$, defined in (3a), are guaranteed to cross at the grid nodes, $x_m = mh$. As will be explained below, the solution at the nodes denoted by the two filled circles and three filled ellipses at level n will be needed to determine the solution at node (x_m, t_{n+1}), denoted by an open circle. The different roles played by the solution at filled circular and elliptical nodes will be explained after Eqs. (21).

An *important condition* that we will require of any MoC scheme developed below is that *it avoids interpolation of the solution along nodes at any one time level.* This is because such an interpolation is likely to introduce numerical diffusion and dispersion to the scheme, which are precisely the effects one wants to avoid by using the MoC. An MoC-RK scheme will solve each of the equations in (3b) using a particular RK solver along the respective characteristics. RK solvers of order higher than second are known to require evaluation of $\mathbf{f}^{\pm}$ at an intermediate time level $t = t_{n+b}$ with $b \in (0,\, 1)$. As we will show below, this will require (approximate) evaluation of $\mathbf{y}^{-}$ at $t = t_{n+b}$ when solving the ODE for $\mathbf{y}^{+}$ along $\xi^{+} = \text{const}$, and vice versa. Such an evaluation of $\mathbf{y}^{-}$ can be done only along the

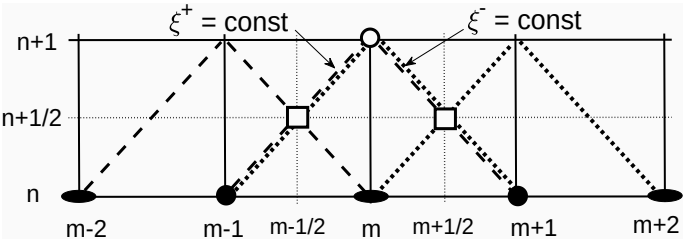

Figure 1: Stencil for MoC-RK3 scheme. Vertical axis is time. Actual nodes are at the intersection of the (solid) grid lines, while *virtual* nodes (open squares) are at the intersection of thin dotted lines. Thick dashed and dotted lines show the directions of the characteristics. Nodes connected by dashed (dotted) characteristics are used to determine $(\mathbf{y}^+)_m^{n+1}$ (respectively, $(\mathbf{y}^-)_m^{n+1}$).

characteristic $\xi^- = \text{const}$, via the second equation in (3b). It is clear from Fig. 1 that the only value of b that will not require interpolation along nodes of time level t_n is $b = 1/2$. The corresponding "virtual node" where $\mathbf{y}^-$ and $\mathbf{y}^+$ will be evaluated are shown by open squares. Therefore, *among all possible RK3 solvers we need to consider only those that involve the intermediate level* $t_{n+1/2}$ *and no other intermediate level.*

Such a well-known RK3 solver for the ODE

$$\mathbf{y}_t = \mathbf{f}(\mathbf{y}, t) \tag{12}$$

is:

$$(\boldsymbol{\kappa}_1)^n = \mathbf{f}(\mathbf{y}^n, t_n), \quad (\boldsymbol{\kappa}_2)^n = \mathbf{f}\left(\mathbf{y}^n + \frac{h}{2}(\boldsymbol{\kappa}_1)^n,\, t_{n+1/2}\right), \quad (\boldsymbol{\kappa}_3)^n = \mathbf{f}\left(\mathbf{y}^n - h\,(\boldsymbol{\kappa}_1)^n + 2h\,(\boldsymbol{\kappa}_2)^n,\, t_{n+1}\right); \tag{13a}$$

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \frac{h}{6}\left((\boldsymbol{\kappa}_1)^n + 4(\boldsymbol{\kappa}_2)^n + (\boldsymbol{\kappa}_3)^n\right); \tag{13b}$$

where $\mathbf{y}^n$ is the solution at time level t_n and $\boldsymbol{\kappa}_{1,2,3}$ are known as "stage derivatives". When applied to ODEs (3b), the first stage derivatives acquire the following form:

$$(\boldsymbol{\kappa}_1^{\pm})_{m\mp1}^n = \mathbf{f}^{\pm}\left((\mathbf{y}^+)_{m\mp1}^n, (\mathbf{y}^-)_{m\mp1}^n\right). \tag{14}$$

Note that the (m, n) index of a stage derivative is, and in what follows will be, that of the node where the "old" solution at time level t_n is taken to obtain the "new" solution at node (m, n).

The following remark about (14) contains *the key idea* of how a solver for an ODE (12) can be applied to a system of ODEs (3) *defined along different characteristics.* To illustrate that idea, let us focus on $\boldsymbol{\kappa}_1^+$. Note that the evolution of $\mathbf{y}^+$ along $\xi^+ = \text{const}$ is determined by $\mathbf{f}^+(\mathbf{y}^+, \mathbf{y}^-)$. Comparing this with (12), one sees that the role of t in $\mathbf{f}(\mathbf{y}, t)$ is played by $\mathbf{y}^-(t)$. That is, in the expression for $\boldsymbol{\kappa}_1^+$, t_n in (13a) must be interpreted as $\mathbf{y}^-(\xi^+, t_n)$, where ξ^+ is the constant value on the characteristic. From Fig. 1, one sees that one should take $\mathbf{y}^-(\xi^+, t_n) \equiv (\mathbf{y}^-)_{m-1}^n$ in the

expression for $(\boldsymbol{\kappa}_1^+)_{m-1}^n$. Similarly, when solving the ODE (3b) for $\mathbf{y}^-$, one interprets $\mathbf{y}^+$ as the counterpart of t in (12), and hence $\mathbf{y}^+(\xi^-, t_n) \equiv (\mathbf{y}^+)_{m+1}^n$.

Following the same idea, one obtains expressions for the other stage derivatives:

$$(\boldsymbol{\kappa}_2^+)_{m-1}^n = \mathbf{f}^+ \left((\mathbf{y}^+)_{m-1}^n + \frac{h}{2} (\boldsymbol{\kappa}_1^+)_{m-1}^n \,,\ (\mathbf{y}^-)_{m-1/2}^{n+1/2} \right) ; \tag{15a}$$

$$(\boldsymbol{\kappa}_2^-)_{m+1}^n = \mathbf{f}^- \left((\mathbf{y}^+)_{m+1/2}^{n+1/2} \,,\ (\mathbf{y}^-)_{m+1}^n + \frac{h}{2} (\boldsymbol{\kappa}_1^-)_{m+1}^n \right) ; \tag{15b}$$

$$(\boldsymbol{\kappa}_3^+)_{m-1}^n = \mathbf{f}^+ \left((\mathbf{y}^+)_{m-1}^n - h\,(\boldsymbol{\kappa}_1^+)_{m-1}^n + 2h\,(\boldsymbol{\kappa}_2^+)_{m-1}^n \,,\ (\mathbf{y}_{(2)}^-)_m^{n+1} \right) ; \tag{16a}$$

$$(\boldsymbol{\kappa}_3^-)_{m+1}^n = \mathbf{f}^- \left((\mathbf{y}_{(2)}^+)_m^{n+1} \,,\ (\mathbf{y}^-)_{m+1}^n - h\,(\boldsymbol{\kappa}_1^-)_{m+1}^n + 2h\,(\boldsymbol{\kappa}_2^-)_{m+1}^n \right) . \tag{16b}$$

Notations $(\mathbf{y}^\pm)_{m\pm1/2}^{n+1/2}$ and $(\mathbf{y}_{(2)}^\pm)_m^{n+1}$ stand for a certain approximation of the respective $\mathbf{y}$, as will be defined shortly. Given the stage derivatives, one finds the solution at the next time level by a counterpart of (13b):

$$(\mathbf{y}^\pm)_m^{n+1} = (\mathbf{y}^\pm)_{m\mp1}^n + \frac{h}{6} \left((\boldsymbol{\kappa}_1^\pm)_{m\mp1}^n + 4(\boldsymbol{\kappa}_2^\pm)_{m\mp1}^n + (\boldsymbol{\kappa}_3^\pm)_{m\mp1}^n \right) . \tag{17}$$

We now introduce notations that will be extensively used below:

$$(\mathbf{y}_{(1)}^\pm)_m^{n+1} = (\mathbf{y}^\pm)_{m\mp1}^n + h\,(\boldsymbol{\kappa}_1^\pm)_{m\mp1}^n ; \tag{18a}$$

$$(\mathbf{y}_{(2)}^\pm)_m^{n+1} = \frac{1}{2} \left[(\mathbf{y}^\pm)_{m\mp1}^n + (\mathbf{y}_{(1)}^\pm)_m^{n+1} + h\,\mathbf{f}^\pm \left((\mathbf{y}_{(1)}^+)_m^{n+1},\, (\mathbf{y}_{(1)}^-)_m^{n+1} \right) \right] . \tag{18b}$$

They provide, respectively, the first- and second-order accurate approximations of the solution at node $(m, n+1)$. The first one is obtained by the simple Euler (SE) solver, and the second one, by the modified Euler (ME) solver (a.k.a. explicit trapezoid rule), applied along respective characteristics $\xi^\pm = \text{const}$. Note that $(\mathbf{y}_{(2)}^\pm)_m^{n+1}$ are used in (16); this will be explained shortly.

We now describe how the approximations $(\mathbf{y}^\pm)_{m\pm1/2}^{n+1/2}$ in (15) are computed, using $(\mathbf{y}^-)_{m-1/2}^{n+1/2}$ as the specific example. Note that this variable in (15a) is meant to denote $\mathbf{y}^-(\xi^+, t_{n+1/2})$, as was explained after (14). Since its exact value is not available, one needs to use its approximation. A *key observation* is that it suffices for this approximation to have error $O(h^3)$. Indeed, such an error will introduce an error $h \cdot O(h^3) = O(h^4)$ in the solution (17), and this is consistent with the fact that the sought third-order accurate solution $(\mathbf{y}^+)_m^{n+1}$ has a local truncation error $O(h^4)$. For the same reason, it suffices to use the approximate solution (18b) instead of the exact values $\mathbf{y}^\pm(\xi^\mp, t_{n+1})$ in (16).

Now, the desired approximation of $(\mathbf{y}^-)^{n+1/2}_{m-1/2}$ is given by the Taylor expansion (see Fig. 1):

$$\begin{aligned}(\mathbf{y}^-)^{n+1/2}_{m-1/2} &= (\mathbf{y}^-)^n_m + \frac{h}{2}\left[(\mathbf{y}^-)^n_m\right]' + \frac{(h/2)^2}{2}\left[(\mathbf{y}^-)^n_m\right]'' + O(h^3) \\ &= (\mathbf{y}^-)^n_m + \frac{h}{2}\,\mathbf{f}^-\left((\mathbf{y}^+)^n_m, (\mathbf{y}^-)^n_m\right) + \frac{h^2}{8}\left[(\mathbf{y}^-)^n_m\right]'' + O(h^3)\,. \end{aligned} \tag{19}$$

We know the first two terms in the last expression, but not the third. We find it from the fact that the solution defined in (18b) has the Taylor expansion

$$(\mathbf{y}^-_{(2)})^{n+1}_{m-1} = (\mathbf{y}^-)^n_m + h\left[(\mathbf{y}^-)^n_m\right]' + \frac{h^2}{2}\left[(\mathbf{y}^-)^n_m\right]'' + O(h^3)\,. \tag{20}$$

Solving (20) for $[(\mathbf{y}^-)^n_m]''$ and substituting the result into (19) yields:

$$(\mathbf{y}^-)^{n+1/2}_{m-1/2} = \frac{3}{4}(\mathbf{y}^-)^n_m + \frac{1}{4}(\mathbf{y}^-_{(2)})^{n+1}_{m-1} + \frac{h}{4}(\boldsymbol{\kappa}^-_1)^n_m\,; \tag{21a}$$

here we have used (14) and also omitted the $O(h^3)$ contribution. By the same token, the similar term in (15b) is computed as

$$(\mathbf{y}^+)^{n+1/2}_{m+1/2} = \frac{3}{4}(\mathbf{y}^+)^n_m + \frac{1}{4}(\mathbf{y}^+_{(2)})^{n+1}_{m+1} + \frac{h}{4}(\boldsymbol{\kappa}^+_1)^n_m\,. \tag{21b}$$

By using different filled symbols — circles and ellipses — in Fig. 1, we indicated that the former contributed to the solution at node $(n+1, m)$ via the variables that were explicitly listed in the stage derivatives (14)–(16), while the latter contributed via the auxiliary solutions (21).

To summarize, Eqs. (14)–(18) and (21) define an MoC-RK3 scheme that is a counterpart of the ODE solver (13). The reason why this is *a*, and not *the*, counterpart scheme is that approximate solutions $\mathbf{y}^\pm_{(1)}$ and $\mathbf{y}^\pm_{(2)}$ could, in principle, be computed by different first- and second-order schemes; other "degrees of freedom" in obtaining $(\mathbf{y}^\pm)^{n+1}_m$ will be mentioned in Section 3.3. However, using any of those degrees of freedom will not favorably affect stability of the resulting MoC-RK3 scheme. We will address this issue in the next subsection.

Let us make two comments on using BC for the MoC-RK3. First, periodic BC require one to define two nodes outside of each boundary (i.e., $m = 0, -1$ and $m = M+1, M+2$). This is done simply by:

$$(\mathbf{y}^\pm)_{1-j} \equiv (\mathbf{y}^\pm)_{M-j+1} \qquad \text{and} \qquad (\mathbf{y}^\pm)_{M+j} \equiv (\mathbf{y}^\pm)_j\,; \qquad j = 1, 2. \tag{22}$$

Second, the treatment of physically more relevant nonreflecting BC

$$\mathbf{y}^+(0,t) = \mathbf{b}_{\text{left}}(t), \qquad \mathbf{y}^-(L,t) = \mathbf{b}_{\text{right}}(t) \tag{23a}$$

where $[0,\, L]$ is the spatial domain where (1a) is solved, proceeds slightly differently in that some of $\mathbf{y}_{(2)}$'s in (16) and (21) are replaced by the exact BC values

$$(\mathbf{y}^+)_1^n = \mathbf{b}_{\text{left}}(t_n), \qquad (\mathbf{y}^-)_M^n = \mathbf{b}_{\text{right}}(t_n), \qquad n \geq 1. \tag{23b}$$

Specifically, $(\mathbf{y}^-_{(2)})_M^{n+1}$ in (16a) with $m = M$ and $(\mathbf{y}^+_{(2)})_1^{n+1}$ in (16b) with $m = 1$ are replaced with the exact values $(\mathbf{y}^-)_M^{n+1}$ and $(\mathbf{y}^+)_1^{n+1}$, respectively, from (23b). Also, when computing $(\mathbf{y}^-_{(2)})_1^{n+1}$ in (21a) and $(\mathbf{y}^+_{(2)})_M^{n+1}$ in (21b), one similarly uses the respective exact boundary values instead of $(\mathbf{y}^+_{(1)})_1^{n+1}$ and $(\mathbf{y}^-_{(1)})_M^{n+1}$ in (18b).

3.2 Von Neumann analysis of the MoC-RK3 scheme

When carrying out von Neumann analysis of various schemes using the linear model equation (5), it will be convenient to define:

$$z = hk, \qquad \widetilde{\mathbf{y}}_m^n = \begin{pmatrix} (\widetilde{\mathbf{y}}^+)_m^n \\ (\widetilde{\mathbf{y}}^-)_m^n \end{pmatrix}, \qquad \mathbf{Q} = \exp\left[-i\mathbf{\Sigma} z\right], \tag{24}$$

where k is the wavenumber, so that $z \in [-\pi, \pi)$, and tilde here and below denotes variables obtained by linearization. Then the linearization of (17) can be written as:

$$\widetilde{\mathbf{y}}_m^{n+1} = \left[\mathbf{Q} + \frac{h}{6}\,(\mathbf{K}_1 + 4\mathbf{K}_2 + \mathbf{K}_3)\right] \widetilde{\mathbf{y}}_m^n \equiv \mathbf{\Phi}(z)\, \widetilde{\mathbf{y}}_m^n\,. \tag{25}$$

The expression for $\mathbf{K}_1$ is found from the linearization of (14):

$$\mathbf{K}_1 = \mathbf{Q}\,\mathbf{P}\,. \tag{26}$$

Note that the factor $\mathbf{Q}$ has appeared because the m-index of $\widetilde{\mathbf{y}}_m^n$ is shifted by 1 compared to the indices of $(\mathbf{y}^{\pm})_{m\mp1}^n$ on the r.h.s. of (14).

We will now briefly outline the derivation of $\mathbf{K}_2$ and then will state the form of $\mathbf{K}_3$, whose derivation follows similar lines. Linearization of (15a) yields:

$$\left(\widetilde{\boldsymbol{\kappa}}_2^+\right)_{m-1}^n = \mathbf{P}^{++}\left(\left(\widetilde{\mathbf{y}}^+\right)_{m-1}^n + \frac{h}{2}\,\mathbf{K}_1^+ \widetilde{\mathbf{y}}_m^n\right) + \mathbf{P}^{+-}\left(\widetilde{\mathbf{y}}^-\right)_{m-1/2}^{n+1/2}; \tag{27}$$

where $\mathbf{K}_1^+$ denotes the top $N^+ \times N$ block of $\mathbf{K}_1$, with $N \equiv (N^+ + N^-)$. The derivation of $\left(\widetilde{\boldsymbol{\kappa}}_2^+\right)_{m-1}^n$ is completed by computing $\left(\widetilde{\mathbf{y}}^-\right)_{m-1/2}^{n+1/2}$ from the linearization of (21a) and (18); this is quite tedious but straightforward. Combined with a similar calculation for $\left(\widetilde{\boldsymbol{\kappa}}_2^-\right)_{m+1}^n$, this yields:

$$\mathbf{K}_2 = \mathbf{P}_{\text{diag}}\left(\mathbf{Q} + \frac{h}{2}\,\mathbf{K}_1\right) + \frac{1}{8}\,\mathbf{P}_{\text{offdiag}}\left(8\mathbf{I} + (4\mathbf{I} + h\mathbf{P})\; h\mathbf{P}\;\right), \tag{28}$$

where

$$\mathbf{P}_{\text{diag}} = \begin{pmatrix} \mathbf{P}^{++} & \mathbf{O} \\ \mathbf{O} & \mathbf{P}^{--} \end{pmatrix}, \qquad \mathbf{P}_{\text{offdiag}} = \begin{pmatrix} \mathbf{O} & \mathbf{P}^{+-} \\ \mathbf{P}^{-+} & \mathbf{O} \end{pmatrix}, \tag{29}$$

and we have omitted the obvious $(N \times N)$ dimension of the identity matrix. In obtaining the second term in (28), Eq. (29) of [1] was used:

$$(\widetilde{\mathbf{y}}_{(2)})_m^{n+1} = \frac{1}{2}\,[\mathbf{Q} + (\mathbf{I} + h\mathbf{P})\mathbf{Q}(\mathbf{I} + h\mathbf{P})\,]\;(\widetilde{\mathbf{y}})_m^n\,. \tag{30}$$

Similarly,

$$\mathbf{K}_3 = \mathbf{P}_{\rm diag}\,(\mathbf{Q} - h\,\mathbf{K}_1 + 2h\,\mathbf{K}_2) + \frac{1}{2}\,\mathbf{P}_{\rm offdiag}\,(\mathbf{Q} + (\mathbf{I} + h\,\mathbf{P})\;\mathbf{Q}\;(\mathbf{I} + h\,\mathbf{P})\,)\;. \tag{31}$$

The amplification factor of the numerical error is the maximum absolute value of the eigenvalues, $\max|\lambda|$, of matrix $\mathbf{\Phi}(z)$ in (25). Given a matrix $\mathbf{P}$, the amplification factor is found numerically (say, by Matlab) from Eqs. (25), (26), (28)–(31) in about one second for $z \in [-\pi,\,\pi)$ and is shown for $\mathbf{P} = \mathbf{P}_1$ and $\mathbf{P} = \mathbf{P}_2$ in Fig. 2(a,c), respectively. The results for $\mathbf{P} = \mathbf{P}_3$ and $i\boldsymbol{\sigma}_1$ are qualitatively similar to that shown in Fig. 2(c), while that for $\mathbf{P} = \mathbf{P}_4$ and $\mathbf{P} = \mathbf{P}_5$ is similar to that shown in Fig. 2(a). The feature to note is the narrow peak of height ~ 0.35 near $z = \pi$. That peak, as well as a minor difference between those peaks in the cases of $\mathbf{P}_1$ and $\mathbf{P}_4$, are illustrated in Fig. 2(b). The growth rates of the numerical error that follow from Figs. 2(a) and 2(c) are $O(1)$ and $O(h)$, corresponding, respectively, to strong and weak numerical instabilities. For example, Fig. 2(a) suggests that

$$\max_z |\lambda| = 1 + c\,h, \tag{32a}$$

where c is some constant ($c \sim 0.35$). Then the error in $n = t/h$ time steps grows by the factor of

$$(1 + c\,h)^{t/h} \approx e^{ct}, \tag{32b}$$

whence c is the growth rate. Similarly, the growth rate predicted by Fig. 2(c) is approximately $0.25h$ (note that in Fig. 2(c), the vertical axis label has a higher power of h in the denominator than that in Figs. 2(a,b)).

Figures 3(a,b) show Fourier spectra of the numerical solution found by direct numerical simulations of Eqs. (5a) with matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ and with periodic boundary conditions (BC). Since (5a) is linear, spectra of the numerical error of any nonlinear system whose linearized matrix coincides with the respective $\mathbf{P}$ will look the same as in Fig. 3. In Section 3.2. of [1] it was explained why Figs. 3(a) and 3(b) look similar to Figs. 2(a) and 2(c), respectively. There is, however, a minor difference between Figs. 3(a) and 2(a) in that the numerical simulations reveal a slight growth of the error for most z away from 0 and π (which appears as a "shelf" in Fig. 3(a)), which does not seem to be predicted by the von Neumann analysis. However, we verified that this growth is linear (as opposed to exponential) in time and occurs because for each z in that range, matrix $\mathbf{\Phi}(z)$ has

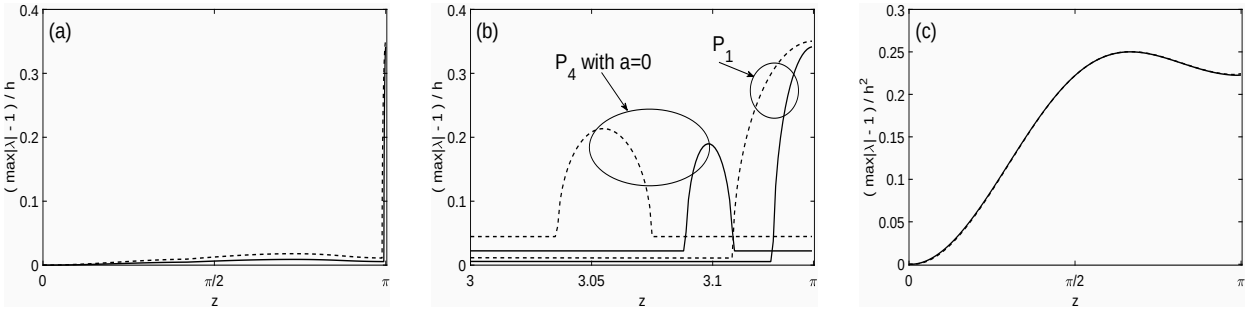

Figure 2: Amplification factor of matrix $\mathbf{\Phi}(z)$ in (25) (for the MoC-RK3) for matrices $\mathbf{P}_1$ (panels (a,b)) and $\mathbf{P}_2$ with $a = 1$ (c). Note a different power of h in the ratio plotted in (a,b) versus that in (c). The plots for $z \in [-\pi, 0]$ are symmetric to those shown above and hence are not presented. Results for two difference values of h are shown to illustrate the trend; $h = 0.05$ (solid) and $h = 0.1$ (dashed). Panel (b) shows a close-up of (a) near $z = \pi$ for two $\mathbf{P}$ matrices.

two pairs of eigenvalues that are almost the same. Such "almost repeated" eigenvalues, leading to a "shelf" in the Fourier spectrum, appear only for $\mathbf{P}_1$ but not for $\mathbf{P}_2$ through $\mathbf{P}_5$.

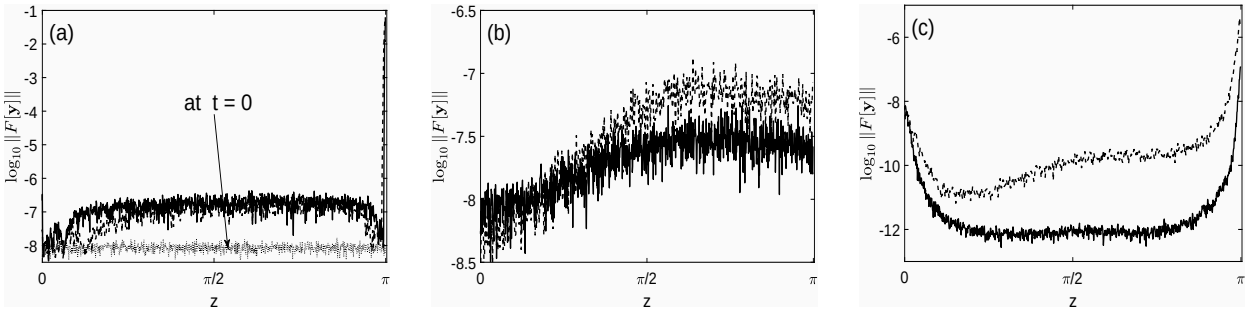

Figure 3: Fourier spectra of the numerical solution by the MoC-RK3 of (5a) with $\mathbf{P}_1$ (a) and $\mathbf{P}_2$ with $a = 1$ (b,c); $h = 0.05$ — solid; $h = 0.1$ — dashed. The initial condition is the white Gaussian noise with standard deviation 10^{-10}. The spectrum at $t = 0$ is shown in (a) only. Note different vertical scales in all three panels. Panels (a,b) are for periodic BC, while (c) is for nonreflecting BC (where the solution is multiplied by a spatial window vanishing at the end points before being Fourier-transformed). Simulation parameters: $L = 100$ in all panels; $t = 50$ (a); $t = 100$ (b); $t = 300$ (c) for better visibility. The notation $\| \dots \|$ stands for the ℓ^2-norm of the four-component vector.

The main conclusion of this subsection is that while the developed MoC-RK3 scheme is *essentially* numerically stable for all physically stable systems (1a) with $N^{\pm} = 1$ and for some systems with $N^{\pm} = 2$ ($\mathbf{P} = \mathbf{P}_2, \mathbf{P}_3$), it is unstable for a number of $N^{\pm} = 2$ systems ($\mathbf{P} = \mathbf{P}_1, \mathbf{P}_4, \mathbf{P}_5$), including the Klein–Gordon equation. In the previous sentence we need to clarify the word 'essentially', since Figs. 2(c) and 3(b) exhibit a weak, but still non-negligible numerical instability near the "middle" of the displayed spectral window (i.e., for $k \sim k_{\max}/2$). In [28] it was shown for the MoC-SE and MoC-ME schemes (i.e., Eqs. (18a) and (18b), respectively) that such a weak instability, appearing for periodic BC, can be suppressed by nonreflecting BC. The latter BC are more physically relevant

than periodic ones as they occur, e.g., in the problem of a field incident on a boundary of a medium described by Eqs. (1). Figure 3(c) illustrates that such instability suppression also takes place for the MoC-RK3 scheme (14)–(17). We did not carry out an analysis of this phenomenon for the MoC-RK3 scheme because: (i) It will be considerably more technical than such an analysis for the MoC-ME in [28], given that the MoC-RK3 is more complicated than MoC-ME; (ii) It will not provide any new insight compared to what was shown in [28]; and (iii) It will distract the reader's attention from our main task, which is the development of an MoC scheme that is stable (for nonreflecting BC) for *all* $\mathbf{P}$ matrices listed in Section 2.

It remains to point out two more aspects of the (in)stability of the schemes developed in this work. Both of these aspects were pointed out in [28] in relation to second-order MoC schemes. First, while nonreflecting BC suppress numerical instability "in the bulk" of the spectral window, they are unable to suppress it near $z = 0$ and $z = \pi$. The instability near $z = 0$ is very weak and is inherited from that of the ODE solver applied to a conservative ODE. For example, its growth rate is $O(h^3)$ for RK2 (such as ME) and RK3 solvers and $O(h^5)$ for RK4 and RK5 solvers. Such an instability is, therefore, inconsequential for most but ultra-long ($t \gtrsim 10^6$) simulations with $h \sim 0.01$ and hence will be ignored in what follows. Second, the nonreflecting BC are unable to suppress a *strong numerical* instability near $z = \pi$, such as that seen in Fig. 2(a). That is, if simulations that led to that Figure are repeated with nonreflecting BC, the peak near $z = \pi$ will be essentially the same as in Fig. 2(a) (plot not shown to save space). Thus, the MoC-RK3 is numerically strongly unstable for some of the systems (5): specifically, for those with $\mathbf{P} = \mathbf{P}_1, \mathbf{P}_4, \mathbf{P}_5$, regardless of the BC used.

3.3 Other versions of the MoC-RK3 scheme

In the development of the scheme in Section 3.1, one has the following three degrees of freedom. First, $\mathbf{y}^{\pm}_{(2)}$ in (18b) could be computed by a different second-order MoC scheme, e.g., one based on the midpoint solver.[1] Second, one could use another RK3 solver where one stage derivative is computed at the time level $t_{n+1/2}$. The only such solver with three stages is the Strong Stability Preserving RK3 [33], which for the ODE (12) is given by:

$$(\boldsymbol{\kappa}_1)^n = \mathbf{f}(\mathbf{y}^n, t_n), \quad (\boldsymbol{\kappa}_2)^n = \mathbf{f}\left(\mathbf{y}^n + h(\boldsymbol{\kappa}_1)^n,\, t_{n+1}\right), \quad (\boldsymbol{\kappa}_3)^n = \mathbf{f}\left(\mathbf{y}^n + \frac{h}{4}\left((\boldsymbol{\kappa}_1)^n + (\boldsymbol{\kappa}_2)^n\right),\, t_{n+1/2}\right); \tag{33a}$$

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \frac{h}{6}\left((\boldsymbol{\kappa}_1)^n + (\boldsymbol{\kappa}_2)^n + 4(\boldsymbol{\kappa}_3)^n\right). \tag{33b}$$

[1] The leapfrog solver could, in principle, be another possibility. However, in [1, 28] it was shown that the corresponding MoC scheme can be strongly numerically unstable for both periodic and nonreflecting BC. Hence we do not consider an MoC-leapfrog scheme as a viable option.

Third, $(\mathbf{y}^{\pm})^{n+1/2}_{m\pm1/2}$ in (21) could be found not by Taylor expansion but by using the MoC-ME with half step (see Fig. 1). Several combinations of the above degrees of freedom were tried in [34]. However, all of them yielded results that are qualitatively similar to those shown in Figs. 2 and 3. Therefore, we conclude that the MoC-RK3 scheme cannot be used to simulate a significant number of systems (1a) with $N^{\pm} = 2$, and hence we need to look for a different scheme.

4 MoC schemes with RK4 solver

The scheme presented in this section has the same stability problem as the MoC-RK3. That is, it can be used to stably simulated equations (1a) with $N^{\pm} = 1$ and with $N^{\pm} = 2$ when the linearization matrices are $\mathbf{P}_2$ or $\mathbf{P}_3$, but not the other three cases. Therefore, we will only state the equations of this scheme based on the classical RK4 (cRK4) solver. We will then make a *hypothesis* as to what causes the instability and hence, in Section 6, will be able to formulate stable MoC schemes of 3rd and 4th orders.

The first two stage derivatives of the MoC-cRK4 are given by (14) and (15). The other two are given by:

$$(\boldsymbol{\kappa}_3^{+})^{n}_{m-1} = \mathbf{f}^{+}\left((\mathbf{y}^{+})^{n}_{m-1} + \frac{h}{2}\,(\boldsymbol{\kappa}_2^{+})^{n}_{m-1}\,,\ (\mathbf{y}^{-})^{n+1/2}_{m-1/2}\right)\,; \tag{34a}$$

$$(\boldsymbol{\kappa}_3^{-})^{n}_{m+1} = \mathbf{f}^{-}\left((\mathbf{y}^{+})^{n+1/2}_{m+1/2}\,,\ (\mathbf{y}^{-})^{n}_{m-1} + \frac{h}{2}\,(\boldsymbol{\kappa}_2^{-})^{n}_{m+1}\right)\,; \tag{34b}$$

$$(\boldsymbol{\kappa}_4^{+})^{n}_{m-1} = \mathbf{f}^{+}\left((\mathbf{y}^{+})^{n}_{m-1} + h\,(\boldsymbol{\kappa}_3^{+})^{n}_{m-1}\,,\ (\mathbf{y}^{-}_{(3)})^{n+1}_{m}\right)\,; \tag{35a}$$

$$(\boldsymbol{\kappa}_4^{-})^{n}_{m+1} = \mathbf{f}^{-}\left((\mathbf{y}^{+}_{(3)})^{n+1}_{m}\,,\ (\mathbf{y}^{-})^{n}_{m+1} + h\,(\boldsymbol{\kappa}_3^{-})^{n}_{m+1}\right)\,. \tag{35b}$$

Finally,

$$(\mathbf{y}^{\pm})^{n+1}_{m} = (\mathbf{y}^{\pm})^{n}_{m\mp1} + \frac{h}{6}\,\left((\boldsymbol{\kappa}_1^{\pm})^{n}_{m\mp1} + 2(\boldsymbol{\kappa}_2^{\pm})^{n}_{m\mp1} + 2(\boldsymbol{\kappa}_3^{\pm})^{n}_{m\mp1} + (\boldsymbol{\kappa}_4^{\pm})^{n}_{m\mp1}\right)\,. \tag{36}$$

In (35), $(\mathbf{y}^{\pm}_{(3)})^{n+1}_{m}$ need to be computed with local error $O(h^4)$; see the paragraph before (19). Such a solution is available via the MoC-RK3. Similarly, $(\mathbf{y}^{\pm})^{n+1/2}_{m\pm1/2}$ in (34) and the counterparts of (15) for $(\boldsymbol{\kappa}_2^{\pm})^{n}_{m\mp1}$ also need to be computed with the local error $O(h^4)$. Given $(\mathbf{y}^{\pm}_{(3)})^{n+1}_{m}$, this can be done following the derivation of (21). The result is (see Appendix C in [34]):

$$(\mathbf{y}^{\pm})^{n+1/2}_{m\pm1/2} = \frac{1}{2}\,(\mathbf{y}^{\pm})^{n}_{m} + \frac{1}{2}\,(\mathbf{y}^{\pm}_{(3)})^{n+1}_{m\pm1} + \frac{h}{8}\,\mathbf{f}^{\pm}\left((\mathbf{y}^{+})^{n}_{m}, (\mathbf{y}^{-})^{n}_{m}\right) - \frac{h}{8}\,\mathbf{f}^{\pm}\left((\mathbf{y}^{+}_{(3)})^{n+1}_{m\pm1}, (\mathbf{y}^{-}_{(3)})^{n+1}_{m\pm1}\right)\,. \tag{37}$$

Von Neumann analysis of the MoC-cRK4 follows the lines presented in Section 3.2 and produces qualitatively similar results. Namely, this scheme is strongly unstable for $\mathbf{P} = \mathbf{P}_1$, $\mathbf{P}_4$, and $\mathbf{P}_5$. We

also tried to use a different RK4 solver (the first one listed in [35]), but obtained the same negative results.

Thus, to develop an essentially stable alternative to the MoC-RK3 and MoC-RK4 schemes, it is important to reflect on the question: *Why did they fail to be stable*? We do not know an answer to this. However, from our extensive experimentation with various versions of these schemes, we have surmised that the culprit may be the fact that they all required evaluation of the solution at a virtual node such as $(m+1/2, n+1/2)$; see Fig. 1. We have accepted this hypothesis and therefore sought methods that require solution values *only at the actual nodes* (like the MoC-ME (18b)). This led us to consider the pseudo-RK methods, described next.

5 Review of pseudo-RK solvers for ODE (12)

Here we will present ODE solvers that we will use in the next section to develop MoC schemes free of the strong instability observed for the MoC-RK3 and MoC-RK4. We will begin by stating the 3rd-order pseudo-RK (pRK) solver originally proposed by Byrne and Lambert [29] and then will present a new 4th-order solver, derived in [34]. We will then describe modified versions of pRK3 and pRK4 methods, proposed by Nakashima [36]. The usefulness of those modified pRK solvers will become clear when we consider their stability regions. Finally, we will point out advantages of using pRK solvers over the multi-step Adams–Bashforth ones.

5.1 pRK solvers based on Byrne–Lambert's idea

The framework of the methods proposed in [29] is:

$$\mathbf{y}^{n+1} = \mathbf{y}^n + h \sum_{j=0}^{s} \sum_{i=1}^{m} c_{ji} \, (\boldsymbol{\kappa}_i)^{n-j}, \tag{38a}$$

where c_{ji} as well as a_{il} and b_i in (38b) below, are certain constants, and the stage derivatives have the form:

$$(\boldsymbol{\kappa}_i)^{n-j} = \mathbf{f}\left(\mathbf{y}^{n-j} + \sum_{l=1}^{i-1} a_{il} \, (\boldsymbol{\kappa}_l)^{n-j}, \, t_n + h \, b_i \right) . \tag{38b}$$

Parameter s is the number of time levels *before* the current one which "contribute" their stage derivatives κ_i towards the solution at the new time level. Constants c_{ji} and a_{il} are entirely determined by b_i. According to the *hypothesis* stated at the end of Section 4, b_i must be an integer (and typically 0 or 1). The 3rd-order solver (38) with $s = 1$, $b_1 = 0$, and $b_2 = 1$ was found in [29]; we will refer to it as 'the pRK3':

$$(\boldsymbol{\kappa}_1)^n = \mathbf{f}(\mathbf{y}^n, t_n), \qquad (\boldsymbol{\kappa}_2)^n = \mathbf{f}\left(\mathbf{y}^n + h \, (\boldsymbol{\kappa}_1)^n, \, t_{n+1}\right) ; \tag{39a}$$

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \frac{h}{12}\left(13(\boldsymbol{\kappa}_1)^n + 5(\boldsymbol{\kappa}_2)^n - (\boldsymbol{\kappa}_1)^{n-1} - 5(\boldsymbol{\kappa}_2)^{n-1}\right) . \tag{39b}$$

The 4th-order solver presented in [29] had $s = 1$ (as does (39)), $b_1 = 0$, and could use b_2 and b_3 as free parameters. If one sets $b_2 = 1$, one must set b_3 to equal another integer. Setting $b_3 = 2$ would be awkward in view of the future use of the solver in the MoC framework. Indeed, suppose one solves for $(\mathbf{y}^+)^{n+1}$; then one would need to first estimate $(\mathbf{y}^-)^{n+2}$, as it "replaces" t_{n+2} in the ODE solver (see text after (14)). One can alternatively set $b_3 = -1$. However, this produces no advantage over, and uses more function evaluations per step than, a solver with $s = 2$ that was originally derived in [34] and which has the form ('the pRK4'):

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \frac{h}{24}\left(37(\boldsymbol{\kappa}_1)^n + 9(\boldsymbol{\kappa}_2)^n - 14(\boldsymbol{\kappa}_1)^{n-1} - 18(\boldsymbol{\kappa}_2)^{n-1} + (\boldsymbol{\kappa}_1)^{n-2} + 9(\boldsymbol{\kappa}_2)^{n-2}\right) , \tag{40}$$

with $\boldsymbol{\kappa}_{1,2}$ being defined in (39a).

5.2 pRK solvers based on Nakashima's idea

Nakashima proposed [36] computing $(\boldsymbol{\kappa}_i)^n$ for $i \geq 2$ in (38b) using not only $(\boldsymbol{\kappa}_l)^n$ $(l < i)$, but also the solution and selected stage derivatives from the previous time level. The general form of the Nakashima pRK (NpRK) solver can be found in [36]; its special cases were explored in [37] and [38]. Here we present only a 3rd- and a 4th-order solver with $b_1 = 0$ and $b_2 = 1$. For both solvers, $\boldsymbol{\kappa}_1$ is computed as in (39a) (and all previous instances) and

$$(\boldsymbol{\kappa}_2)^n = \mathbf{f}\left(\mathbf{y}^n + \Lambda\big(\mathbf{y}^n - \mathbf{y}^{n-1}\big) + h\,\left(a_{20}(\boldsymbol{\kappa}_1)^{n-1} + a_{21}(\boldsymbol{\kappa}_1)^n\right), t_{n+1}\right) . \tag{41}$$

Then the NpRK3 solution is computed as:

$$\mathbf{y}^{n+1} = \mathbf{y}^n + h\,\left(c_{11}(\boldsymbol{\kappa}_1)^n + c_{12}(\boldsymbol{\kappa}_2)^n + c_{21}(\boldsymbol{\kappa}_1)^{n-1}\right) ; \tag{42a}$$

$$a_{21} = 2 + a_{20}, \quad \Lambda = -1 - 2a_{20}, \quad c_{11} = 2/3, \quad c_{12} = 5/12, \quad c_{21} = -1/12\,. \tag{42b}$$

Note that unlike in the Byrne–Lambert pRK solvers (with fixed b_i's), here one has a free parameter, a_{20}.

The NpRK4 solution is computed as:

$$\mathbf{y}^{n+1} = \mathbf{y}^n + h\,\left(c_{11}(\boldsymbol{\kappa}_1)^n + c_{12}(\boldsymbol{\kappa}_2)^n + c_{21}(\boldsymbol{\kappa}_1)^{n-1} + c_{22}(\boldsymbol{\kappa}_2)^{n-1} + c_{31}(\boldsymbol{\kappa}_1)^{n-2}\right) . \tag{43a}$$

Here either Λ or c_{22} can be chosen as free parameters. If Λ is free, then

$$a_{20} = -\frac{1+\Lambda}{2}, \quad a_{21} = \frac{3-\Lambda}{2}, \quad c_{11} = \frac{7}{6}, \quad c_{12} = \frac{3}{8}, \quad c_{21} = -\frac{5}{24}, \quad c_{22} = -\frac{3}{8}, \quad c_{31} = \frac{1}{24}\,; \tag{43b}$$

while if c_{22} is free, then

$$a_{20} = 2, \quad a_{21} = 4, \quad \Lambda = -5, \quad c_{11} = \frac{19 - 24c_{22}}{24}\,, \tag{43c}$$

and $c_{12,\,21,\,31}$ have the same values as in (43b). The role of the free parameters in (42) and (43) is revealed in the next subsection.

5.3 Stability regions of pRK and NpRK solvers

The hyperbolic PDEs that we consider in this work conserve the "energy" (or the L_2-norm) of the solution; see [1]. They may also conserve other quantities, such as the Hamiltonian. Eigenvalues of all $\mathbf{P}$ matrices listed in Section 2 lie on the imaginary axis. Therefore, one can expect that ODE solvers whose stability region boundaries have a high degree of tangency to the imaginary axis are preferred to be used in the MoC framework.[2] This is the feature that we will emphasize in this subsection.

In Fig. 4 we show stability regions of the RK3 (13), cRK4 (see Section 4), the pRK3 (39), and the pRK4 (40). The following observations are apparent. First, while the size of the stability region increases with the solver's order for the RK solvers, it decreases for the pRK ones. (The latter trend is the same as that for multi-step solvers.) In particular, the length of the interval along which the stability is tangent to the imaginary axis *appears*, in Fig. 4(a), to be smaller for the pRK4 than for the RK3. However, Fig. 4(b) shows that the situation is actually the opposite. (As a quantitative measure of the degree of tangency of a curve to the imaginary axis, one can use the exponent of h in the ratio $\mathrm{Re}(h\lambda)\,/\,\mathrm{Im}(h\lambda)$, where $(\mathrm{Re}(h\lambda),\, \mathrm{Im}(h\lambda))$ is a point on the curve. Then the degree of tangency of any RK3 and pRK3 solver can be shown to be 3, while that of any RK4 or pRK4 solvers is 5.) Based on Fig. 4(b), one can expect that the pRK4 should be able to preserve conserved quantities of an ODE better than both the pRK3 and RK3 solvers. This was confirmed in [34] for selected nonlinear, non-stiff ODEs. In Section 8 we will demonstrate a similar trend for a PDE.

Figure 5 illustrates the improvement that NpRK solvers can bring over the pRK ones. Tangency of the NpRK3 stability regions to the imaginary axis can be improved by a judicious choice of a_{20}; of the three values shown in Fig. 5, the solver with $a_{20} = 1.4$ has the best tangency. The tangency of the NpRK3 solvers can even exceed that of the RK3 one (see panel (b)). Correspondingly, it was shown in [34] that for the ODEs mentioned in the previus paragraph, the NpRK3 was able to preserve conserved quantities significantly better than the RK3. Analogous results hold for the

[2] Such solvers will at least do well in the limit $k \to 0$, where ∂_x in (1a) can be dropped and the PDE becomes an ODE. Whether they will do well for arbitrary k can only be determined on a case-by-case basis; see the footnote in Section 3.3 about the leapfrog solver, whose ODE stability region is a segment on the imaginary axis.

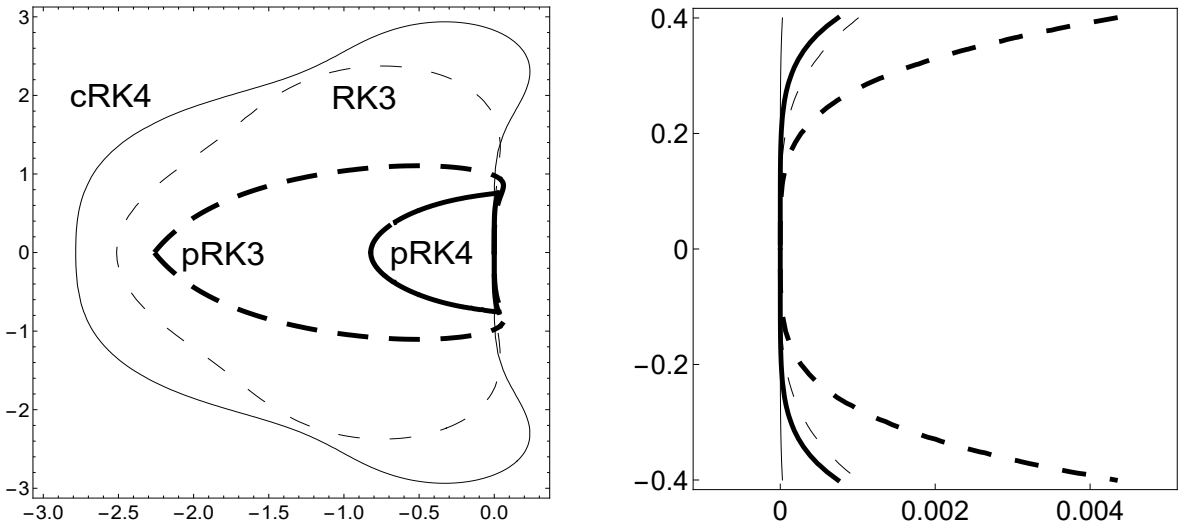

Figure 4: Stability regions of the four ODE solvers for the model equation $y' = \Lambda y$ (see any textbook on Numerical Analysis). As per standard notations, the horizontal and vertical axes denote $h\,\mathrm{Re}\Lambda$ and $h\,\mathrm{Im}\Lambda$, respectively. Various line styles denote the solvers as labeled in panel (a). Panel (b) shows a magnification of panel (a) near the imaginary axis. Line styles pertain to the same solvers in both panels.

NpRK4 solvers (not shown to save space), except that even the optimal of them (with $\Lambda = -5$ and $c_{22} = -0.15$) does not become quite as good as the cRK4 in preserving conserved quantities. In Section 8 we will show that optimizing values of the free parameters in the NpRK solvers can also improve performance of the MoC schemes based on these solvers. However, the optimal values of parameters are not, in general, those suggested by the stability plots and do vary depending on the PDE and its simulated solution.

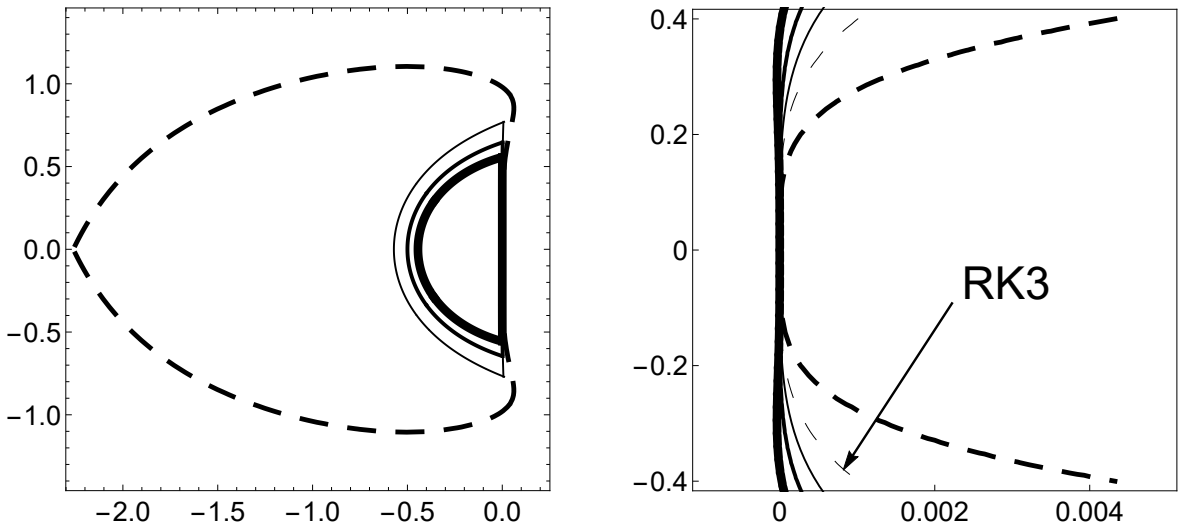

Figure 5: Stability regions of the pRK3 (thick dashed) and the NpRK3 (solid) with $a_{20} = 1.1$ (thin), 1.4 (medium), and 1.7 (thick). Panel (b) shows a magnification of panel (a) near the imaginary axis. The boundary of the stability region of the RK3 is also shown (in (b) only) for comparison.

5.4 Advantages of the pRK over Adams–Bashforth solvers

The pRK solvers (38) and their Nakashima variety are, essentially, a hybrid between RK and multi-step Adams–Bashforth solvers. They have two advantages over the latter solvers. First, one can verify that stability regions of the pRK solvers go closer to the imaginary axis than stability regions of the Adams–Bashforth solvers of respective orders. Second, and *more importantly*, the number of of previous time steps used in Adams–Bashforth solvers is greater than that number used in pRK solvers. When solving an ODE, this is not a problem (and may only moderately increase storage requirements). However, the *implementation* of MoC schemes *with non-periodic BC* becomes increasingly complicated as the number of previous time levels involved increases. In Section 7 we will see that increasing this number from one (in the MoC-pRK3) to two (in the MoC-pRK4) significantly complicates the implementation of the scheme. In comparison, an MoC-Adams–Bashforth-4 scheme would need three previous time levels, and coding it for any non-periodic BC would be very complicated.

6 MoC schemes of 3rd and 4th orders based on (N)pRK solvers for periodic BC

We will begin by presenting the schemes for the 3rd- and 4th-order MoC-pRK3 and MoC-NpRK3. We will then present results of the von Neumann analysis for these schemes and thereby arrive at one of our *key conclusions*: These schemes exhibit only a weak instability with growth rate $O(h)$, which is actually weaker than that of the MoC-ME. A demonstration that this instability disappears for nonreflecting BC, is postponed until Section 7.

6.1 MoC-pRK and MoC-NpRK schemes

In complete analogy with (17), (14), and (15) for the MoC-RK3, the MoC-pRK3 scheme is:

$$(\mathbf{y}^{\pm})_m^{n+1} = (\mathbf{y}^{\pm})_{m\mp1}^{n} + \frac{h}{12}\left(13(\boldsymbol{\kappa}_1^{\pm})_{m\mp1}^{n} + 5(\boldsymbol{\kappa}_2^{\pm})_{m\mp1}^{n} - (\boldsymbol{\kappa}_1^{\pm})_{m\mp2}^{n-1} - 5(\boldsymbol{\kappa}_2^{\pm})_{m\mp2}^{n-1}\right), \tag{44a}$$

where $\boldsymbol{\kappa}_1^{\pm}$ are given by (14) and

$$(\boldsymbol{\kappa}_2^{+})_{m-1}^{n} = \mathbf{f}^{+}\left((\mathbf{y}_{(1)}^{+})_m^{n+1}, (\mathbf{y}_{(2)}^{-})_m^{n+1}\right), \quad (\boldsymbol{\kappa}_2^{-})_{m+1}^{n} = \mathbf{f}^{-}\left((\mathbf{y}_{(2)}^{+})_m^{n+1}, (\mathbf{y}_{(1)}^{-})_m^{n+1}\right), \tag{44b}$$

with $(\mathbf{y}_{(1)}^{\pm})_m^{n+1}$ and $(\mathbf{y}_{(2)}^{\pm})_m^{n+1}$ being found with respective local errors $O(h^2)$ and $O(h^3)$ by (18). Once the stage derivatives $\boldsymbol{\kappa}_{1,2}^{\pm}$ are found at time level t_{n-1}, they are stored for one time step, to advance the solution from t_n to t_{n+1}; see the last two terms in (44a). The stencil for the MoC-pRK3

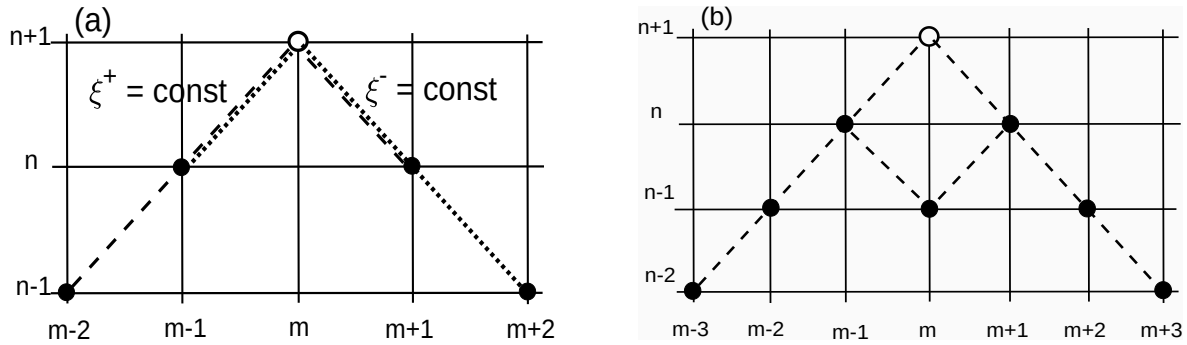

Figure 6: Stencils for the MoC-(N)pRK3 (a) and MoC-(N)pRK4 (b) schemes. Note that unlike in Fig. 1, there are no virtual nodes here. Filled (open) circles show nodes where the solution is available (is to be computed). See text for the explanation of line styles in (a).

is shown in Fig. 6(a). The dashed and dotted lines connect all the nodes required to compute $(\mathbf{y}^+)_m^{n+1}$ and $(\mathbf{y}^-)_m^{n+1}$, respectively.

The MoC-NpRK3 scheme based on the ODE solver (42a) has the same stencil, and is given by:

$$(\mathbf{y}^\pm)_m^{n+1} = (\mathbf{y}^\pm)_{m\mp1}^n + h\,\left(c_{11}(\boldsymbol{\kappa}_1^\pm)_{m\mp1}^n + c_{12}(\boldsymbol{\kappa}_2^\pm)_{m\mp1}^n + c_{21}(\boldsymbol{\kappa}_1^\pm)_{m\mp2}^{n-1}\right)\,, \tag{45a}$$

where $\boldsymbol{\kappa}_1^\pm$ are given by (14) and

$$(\boldsymbol{\kappa}_2^+)_{m-1}^n = \mathbf{f}^+\left(\overline{(\mathbf{y}^+)}_{m-1}^n, (\mathbf{y}_{(2)}^-)_m^{n+1}\right)\,,\quad (\boldsymbol{\kappa}_2^-)_{m+1}^n = \mathbf{f}^-\left((\mathbf{y}_{(2)}^+)_m^{n+1}, \overline{(\mathbf{y}^-)}_{m+1}^n\right)\,, \tag{45b}$$

$$\overline{(\mathbf{y}^\pm)}_m^n \equiv (\mathbf{y}^\pm)_m^n + \Lambda\left((\mathbf{y}^\pm)_m^n - (\mathbf{y}^\pm)_{m\mp1}^{n-1}\right) + h\,\left(a_{20}(\boldsymbol{\kappa}_1^\pm)_{m\mp1}^{n-1} + a_{21}(\boldsymbol{\kappa}_1^\pm)_m^n\right)\,, \tag{45c}$$

with the coefficients a_{ij} and c_{ij} computed as in (42b).

To start the third-order schemes (44) and (45), one can compute the solution at the time level $t_1 \equiv h$ by any second-order scheme, such as the MoC-ME (18b). Also, for periodic BC, a modification of these schemes near the edges of the computational domain, $m = 1$ and $m = M$, is done by (22).

The MoC-pRK4 scheme is:

$$(\mathbf{y}^\pm)_m^{n+1} = (\mathbf{y}^\pm)_{m\mp1}^n + \frac{h}{24}\,\left(37(\boldsymbol{\kappa}_1^\pm)_{m\mp1}^n + 9(\boldsymbol{\kappa}_2^\pm)_{m\mp1}^n - 14(\boldsymbol{\kappa}_1^\pm)_{m\mp2}^{n-1} - 18(\boldsymbol{\kappa}_2^\pm)_{m\mp2}^{n-1} + (\boldsymbol{\kappa}_1^\pm)_{m\mp3}^{n-2} + 9(\boldsymbol{\kappa}_2^\pm)_{m\mp3}^{n-2}\right)\,, \tag{46a}$$

where $\boldsymbol{\kappa}_1^\pm$ are given by (14) and

$$(\boldsymbol{\kappa}_2^+)_{m-1}^n = \mathbf{f}^+\left((\mathbf{y}_{(1)}^+)_m^{n+1}, (\mathbf{y}_{(3)}^-)_m^{n+1}\right)\,,\quad (\boldsymbol{\kappa}_2^-)_{m+1}^n = \mathbf{f}^-\left((\mathbf{y}_{(3)}^+)_m^{n+1}, (\mathbf{y}_{(1)}^-)_m^{n+1}\right)\,. \tag{46b}$$

The only difference between (46b) and (44b) is that for the former, we require a third-order (local error $O(h^4)$), not second-order, accurate solution, $(\mathbf{y}_{(3)}^\pm)_m^{n+1}$. It can be found by the MoC-pRK3 (44). Thus, finding a fourth-order accurate solution by the MoC-pRK4 requires first finding its

less accurate approximation by the MoC-pRK3. Fortunately for the implementation, these two methods share one stage derivative out of two: $\boldsymbol{\kappa}_1^{\pm}$. The stage derivatives, found at a given time level, are stored to be used at two subsequent time levels. The stencil for the MoC-pRK4 is shown in Fig. 6(b). Note that the solution at the node $(n-1, m)$ is needed to compute $(\mathbf{y}_{(3)}^{\pm})_m^{n+1}$, which is used by $(\boldsymbol{\kappa}_2^{\pm})_{m\mp1}^{n}$ in (46b).

The MoC-NpRK4 scheme has the same stencil and is given by:

$$(\mathbf{y}^{\pm})_m^{n+1} = (\mathbf{y}^{\pm})_{m\mp1}^{n} + h\,\left(c_{11}(\boldsymbol{\kappa}_1^{\pm})_{m\mp1}^{n} + c_{12}(\boldsymbol{\kappa}_2^{\pm})_{m\mp1}^{n} + c_{21}(\boldsymbol{\kappa}_1^{\pm})_{m\mp2}^{n-1} + c_{22}(\boldsymbol{\kappa}_2^{\pm})_{m\mp2}^{n-1} + c_{31}(\boldsymbol{\kappa}_1^{\pm})_{m\mp3}^{n-2}\right)\,, \tag{47a}$$

where $\boldsymbol{\kappa}_1^{\pm}$ are given by (14) and

$$(\boldsymbol{\kappa}_2^{+})_{m-1}^{n} = \mathbf{f}^{+}\left((\overline{\mathbf{y}^{+}})_{m-1}^{n}, (\mathbf{y}_{(3)}^{-})_m^{n+1}\right)\,, \quad (\boldsymbol{\kappa}_2^{-})_{m+1}^{n} = \mathbf{f}^{-}\left((\mathbf{y}_{(3)}^{+})_m^{n+1}, (\overline{\mathbf{y}^{-}})_{m+1}^{n}\right)\,, \tag{47b}$$

with $(\overline{\mathbf{y}^{\pm}})_{m+1}^{n}$ being given by (45c) and the coefficients a_{ij} and c_{ij} computed as in (43b) or (43c). Similarly to the difference between (46b) and (44b), in (47b) we require $(\mathbf{y}_{(3)}^{\pm})_m^{n+1}$ instead of $(\mathbf{y}_{(2)}^{\pm})_m^{n+1}$ in (45b). This solution can be found by either MoC-pRK3 or MoC-NpRK3.

To implement periodic BC for the MoC-pRK4 and MoC-NpRK4, one extends (22) to include $j = 3$.

6.2 von Neumann stability analysis of the MoC-(N)pRK schemes

We will show details for the MoC-pRK3 and will state only the final results for the MoC-NpRK3 and MoC-pRK4, as the corresponding details are analogous but more technically involved. These details for the MoC-pRK4 can be found in Section 6.2 of [34]. We did not work out this analysis for the MoC-NpRK4 since it is even more involved; however, direct numerical simulations with this scheme indicate that its stability is similar to that of the MoC-pRK4.

Since (44) is, essentially, a multi-step scheme, we introduce an "extended" vector:

$$(\widetilde{\mathbf{y}}_{\text{ext}})_m^n = \begin{pmatrix} \widetilde{\mathbf{y}}_m^n \\ \widetilde{\mathbf{y}}_m^{n-1} \end{pmatrix} \tag{48}$$

and seek to put the linearization of (44) into the form analogous to (25):

$$(\widetilde{\mathbf{y}}_{\text{ext}})_m^{n+1} \;\equiv\; \boldsymbol{\Phi}(z)\,(\widetilde{\mathbf{y}}_{\text{ext}})_m^n\,. \tag{49}$$

We begin by formally writing the linearization of (44a) as

$$(\widetilde{\mathbf{y}})_m^{n+1} = \left[[\mathbf{Q},\,\mathbf{O}] \;+\; \frac{h}{12}\left(13\mathbf{K}_1^{(0)} + 5\mathbf{K}_2^{(0)} - \mathbf{K}_1^{(-1)} - 5\mathbf{K}_2^{(-1)}\right)\right]\,(\widetilde{\mathbf{y}}_{\text{ext}})_m^n \;\equiv\; \boldsymbol{\Phi}_1(z)\,(\widetilde{\mathbf{y}}_{\text{ext}})_m^n \tag{50}$$

and will calculate the $N \times 2N$ matrices $\mathbf{K}_{1,\,2}^{(0),\,(-1)}$ one at a time, where N is the length of vector $\widetilde{\mathbf{y}}$.

Since from (14):

$$\left(\begin{array}{c} \widetilde{\boldsymbol{\kappa}}_1^+ \\ \widetilde{\boldsymbol{\kappa}}_1^- \end{array} \right)_m^n = \big[\mathbf{P},\, \mathbf{O}\big]\, (\widetilde{\mathbf{y}}_{\rm ext})_m^n\,, \qquad \left(\begin{array}{c} \widetilde{\boldsymbol{\kappa}}_1^+ \\ \widetilde{\boldsymbol{\kappa}}_1^- \end{array} \right)_m^{n-1} = \big[\mathbf{O},\, \mathbf{P}\big]\, (\widetilde{\mathbf{y}}_{\rm ext})_m^n\,, \tag{51a}$$

then, accounting for the lower indices of the $\boldsymbol{\kappa}_1$-terms in (44a), one has:

$$\mathbf{K}_1^{(0)} = \mathbf{Q}\,\big[\mathbf{P},\, \mathbf{O}\big]\,, \qquad \mathbf{K}_1^{(-1)} = \mathbf{Q}^2\,\big[\mathbf{O},\, \mathbf{P}\big]\,. \tag{51b}$$

To obtain linearization of $\boldsymbol{\kappa}_2$ in (44b), one first notices that linearization of the $\mathbf{y}_{(1)}^{\pm}$-terms in it is:

$$\begin{aligned} (\widetilde{\mathbf{y}}_{(1)})_m^{n+1} &= \mathbf{Q}\,\big(\big[\mathbf{I},\, \mathbf{O}\big] + h\,\big[\mathbf{P},\, \mathbf{O}\big]\,\big)\,(\widetilde{\mathbf{y}}_{\rm ext})_m^n \equiv \mathbf{S}_{(1)}^{(0)}\,(\widetilde{\mathbf{y}}_{\rm ext})_m^n\,, \\ (\widetilde{\mathbf{y}}_{(1)})_m^{n} &= \mathbf{Q}\,\big(\big[\mathbf{O},\, \mathbf{I}\big] + h\,\big[\mathbf{O},\, \mathbf{P}\big]\,\big)\,(\widetilde{\mathbf{y}}_{\rm ext})_m^n \equiv \mathbf{S}_{(1)}^{(-1)}\,(\widetilde{\mathbf{y}}_{\rm ext})_m^n\,. \end{aligned} \tag{52}$$

Note that $\mathbf{S}_{(1)}^{(0),\,(-1)}$ are $N \times 2N$ matrices. Linearizations of $\mathbf{y}_{(2)}^{\pm}$, previously given by (30), should now be written as:

$$\begin{aligned} (\widetilde{\mathbf{y}}_{(2)})_m^{n+1} &= \frac{1}{2}\left[\big[\mathbf{Q},\, \mathbf{O}\big] + \mathbf{S}_{(1)}^{(0)} + h\,\big[\mathbf{P},\, \mathbf{O}\big]\,\left(\begin{array}{c} \mathbf{S}_{(1)}^{(0)} \\ \mathbf{O} \end{array} \right)\right]\,(\widetilde{\mathbf{y}}_{\rm ext})_m^n \equiv \mathbf{S}_{(2)}^{(0)}\,(\widetilde{\mathbf{y}}_{\rm ext})_m^n\,, \\ (\widetilde{\mathbf{y}}_{(2)})_m^{n} &= \frac{1}{2}\left[\big[\mathbf{O},\, \mathbf{Q}\big] + \mathbf{S}_{(1)}^{(-1)} + h\,\big[\mathbf{O},\, \mathbf{P}\big]\,\left(\begin{array}{c} \mathbf{O} \\ \mathbf{S}_{(1)}^{(-1)} \end{array} \right)\right]\,(\widetilde{\mathbf{y}}_{\rm ext})_m^n \equiv \mathbf{S}_{(2)}^{(-1)}\,(\widetilde{\mathbf{y}}_{\rm ext})_m^n\,. \end{aligned} \tag{53}$$

Note that in the last terms in the square brackets, $\mathbf{O}$ is the zero matrix of dimension $N \times 2N$. With (52) and (53), the linearization of (44b) yields the expression for $\mathbf{K}_{(2)}^{(0)}$ in (50):

$$\mathbf{K}_{(2)}^{(0)} = \big[\mathbf{P}_{\rm diag},\, \mathbf{O}\big]\,\left(\begin{array}{c} \mathbf{S}_{(1)}^{(0)} \\ \mathbf{O} \end{array} \right) + \big[\mathbf{P}_{\rm offdiag},\, \mathbf{O}\big]\,\left(\begin{array}{c} \mathbf{S}_{(2)}^{(0)} \\ \mathbf{O} \end{array} \right)\,. \tag{54a}$$

Similarly, one obtains:

$$\mathbf{K}_{(2)}^{(-1)} = \mathbf{Q}\left(\big[\mathbf{O},\, \mathbf{P}_{\rm diag}\,\big]\,\left(\begin{array}{c} \mathbf{O} \\ \mathbf{S}_{(1)}^{(-1)} \end{array} \right) + \big[\mathbf{O},\, \mathbf{P}_{\rm offdiag}\,\big]\,\left(\begin{array}{c} \mathbf{O} \\ \mathbf{S}_{(2)}^{(-1)} \end{array} \right)\right)\,. \tag{54b}$$

Combining (51b) and (54) with (52) and (53) yields matrix $\boldsymbol{\Phi}_1(z)$ in (50). The remaining step of calculating $\boldsymbol{\Phi}(z)$ in (49) is straightforward because $(\widetilde{\mathbf{y}})_m^n = [\mathbf{I},\, \mathbf{O}]\,(\widetilde{\mathbf{y}}_{\rm ext})_m^n$, whence

$$\boldsymbol{\Phi}(z) \,=\, \left(\begin{array}{c} \boldsymbol{\Phi}_1(z) \\ \big[\mathbf{I},\, \mathbf{O}\,\big] \end{array} \right)\,. \tag{55}$$

Representative scaled amplification factors of the matrix (55) are shown in Fig. 7(a). These factors for the MoC-NpRK3 and MoC-pRK4 are shown in Fig. 7(b,c). Note that unlike Figs. 2(a,b), they predict only a weak instability, with growth rate $O(h)$, for the same $\mathbf{P}$-matrices for which the

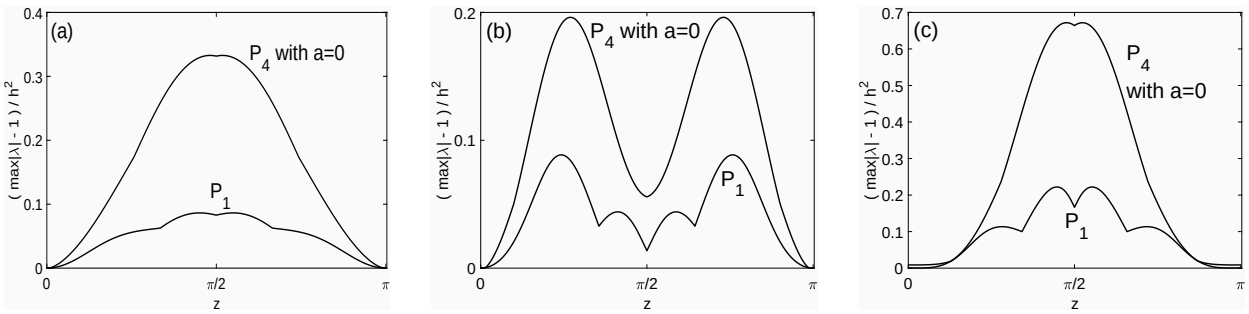

Figure 7: (a): Amplification factor of matrix $\mathbf{\Phi}(z)$ in (55) for the MoC-pRK3 for two representative $\mathbf{P}$-matrices. (b): Amplification factor for the MoC-NpRK3 with the "optimal" (see Section 5.3) value $a_{20} = 1.4$ in (42b). (c): Amplification factor for the MoC-pRK4. Amplification factors for other matrices listed in Section 2 are either similar to, or located between, the displayed ones. All curves are shown for a rather large value of the step size, $h = 0.05$. For smaller h, the corresponding curves look almost identical to those shown. For h as large at 0.1, these curves for some of $\mathbf{P}$-matrices look quantitatively different (but qualitatively the same). This is consistent with the fact [36, 38] that pRK solvers are less accurate (i.e., have a greater numeric constant in the $O(h^3)$ or $O(h^4)$ error terms) than the RK solvers of the same orders. In regards to panel (b), similar curves obtained for $h < 0.01$ are insensitive to a_{20} in a large range. For $h = 0.05$, they are close to the displayed curves for $a_{20} \in [0, 3]$.

MoC-RK3 and MoC-RK4 schemes had the strong instability with growth rate $O(1)$. Note that while the instability growth rate is greater for the MoC-pRK4 than for the MoC-pRK3 scheme, it is still some 30% smaller than that for MoC-ME, where the maximum of the curve corresponding to $\mathbf{P}_4$ with $a = 0$ is at 1 (see Fig. 2(a) in [1]).

These results are confirmed[3] by direct numerical simulations of Eqs. (5) with periodic BC, as shown in Fig. 8. They also show that: (i) the weak instability in the MoC-NpRKn ($n = 3, 4$) can be made weaker than that of the MoC-pRKn methods of the same order, and (ii) this instability is stronger for the MoC-(N)pRK4 than for the MoC-(N)pRK3. In the next section, we will demonstrate that the weak instability for wavenumbers in the "bulk" of the spectrum, i.e. for z away from 0 and π in Figs. 7 and 8, is suppressed when one uses nonreflecting BC. The verification that the numerical errors of all four schemes scale as the appropriate powers of h, is postponed until Section 8.

The results for matrix $\mathbf{P}_4$ shown in Figs. 8(a)–(c) quantitatively confirm the corresponding results in Figs. 7(a)–(c). For example, from Fig. 7(a) one finds that $\lambda(z = \pi/2) \approx 1 + 0.33h^2$, whence the growth rate of the corresponding harmonic is $0.33h$ (see (32)). This agrees very well with the growth of this harmonic by 7.3 orders of magnitude, as observed in Fig. 8(a). We verified that similar quantitative agreement takes place for all matrices except $\mathbf{P}_1$ for the schemes reported

[3] qualitatively for $\mathbf{P}_1$ and quantitatively for the other $\mathbf{P}$-matrices; see next paragraph

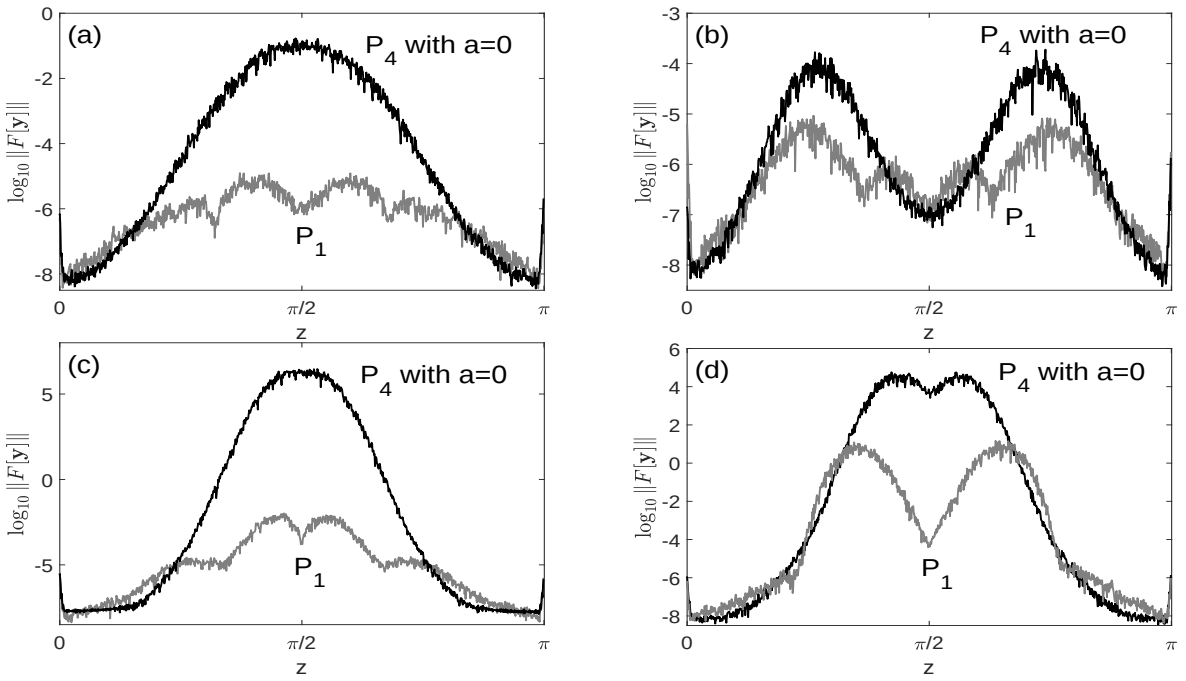

Figure 8: (a)–(c): Fourier spectra of numerical solutions of Eqs. (5) corresponding to the von Neumann results shown in Fig. 7(a–c). (d): Same, but for MoC-NpRK4. Simulation parameters: $L = 100$, $h = 0.05$, $t = 1000$. The initial condition is the white Gaussian noise with standard deviation 10^{-10}. The $\mathbf{y}^{\pm}_{(3)}$ solutions used by the MoC-NpRK4 are computed by the MoC-NpRK3. The free parameter in the MoC-NpRK3 was set to $a_{20} = 1.4$ (see (42b) and Fig. 5) for panels (b) and (d); the remaining free parameter in the MoC-NpRK4 was set to $c_{22} = -0.2$ (see (43c)).

in panels (a)–(c) of Figs. 7 and 8. (When the $\mathbf{P}$-matrix depended on parameter a, we verified the agreement only for two random values of a.) The reason that the amplified noise spectrum in simulations using matrix $\mathbf{P}_1$ agrees only qualitatively with the results of the von Neumann analysis is similar to that described in relation to the discrepancy between Figs. 3(a) and 2(a). Namely: for each z, the spectrum of the corresponding matrix $\mathbf{\Phi}(z)$ in (49) contains pairs of complex eigenvalues that differ by an amount much smaller than $O(h)$. This leads to an oscillatory rather than monotonic growth of the harmonics' amplitudes, with the "swing" of the oscillations reaching some two orders of magnitude.

7 MoC-pRK schemes with nonreflecting BC

Implementation of (N)pRK schemes for ODEs is straightforward in the sense that the general algorithm is to be modified only at the first few time levels. This is common for multi-step solvers and is accomplished by computing the solution at those levels by a single-step solver. For the MoC-(N)pRK schemes, applied to PDEs, the situation becomes considerably more complicated: one also needs to handle nodes adjacent to the boundaries separately from the in-bulk algorithm.

Let us note, however, that this is a common feature of *all* finite-difference schemes due to the one-sided approximation to spatial derivatives being different from symmetric approximations, which are typically used in the bulk of the grid.

In this section we will present the *ideas* of how nonreflecting BC (23) can be imposed for the MoC-pRK3 and MoC-pRK4. The slightly more general partially nonreflecting BC,

$$\mathbf{y}^+(0,t) = R_{\text{left}}\mathbf{y}^-(0,t) + \mathbf{b}_{\text{left}}(t), \qquad \mathbf{y}^-(L,t) = R_{\text{right}}\mathbf{y}^+(L,t) + \mathbf{b}_{\text{right}}(t)\,, \tag{56}$$

are imposed similarly. Also, BC (23) (or (56)) are imposed similarly for the MoC-NpRK3 and MoC-NpRK4 schemes, respectively. Even though the idea for the MoC-pRK4 case is conceptually the same as that for the MoC-pRK3 one, it is technically more complex. Therefore, we will present them separately and in both cases focus only on the left boundary, since the right one is handled in an analogous way. The *implementation* of these ideas in a code is yet another nontrivial task; it is outlined in Appendix B, where a GitHub link to the actual codes is also given.

7.1 Nonreflecting BC for the MoC-pRK3 (44), (14)

The stencil for the MoC-pRK3 in the vicinity of the left boundary is shown in Fig. 9(a). We now explain how the solution is found at the time levels with $n \geq 2$ given the initial condition at the time level with $n = 1$ and for $m = 1, \ldots, M$. First, one computes, for future use, the solution at the virtual node $(n = 1, m = 0)$, i.e., $(\mathbf{y}^{\pm})^1_0$, by the 3rd-order Lagrange extrapolation:

$$(\mathbf{y}^{\pm})^n_0 = 3(\mathbf{y}^{\pm})^n_1 - 3(\mathbf{y}^{\pm})^n_2 + (\mathbf{y}^{\pm})^n_3, \qquad n = 1, \tag{57}$$

whose order is consistent with the global error, $O(h^3)$, of the scheme. *We stress* that the extrapolation is used to compute a point outside the boundary *only* at the time level t_1. This is because in general, we do not assume that the solution is smooth (in x), which is the assumption implied in (57). Moreover, it is unknown (i.e., requires a separate investigation) how such an extrapolation, if done for all n, would affect stability of the scheme. Therefore, for all $n \geq 2$, finding the solution at $m = 0$ will be done by another method, which will be described two paragraphs below.

Second, the solution at $n = 2$, $m = 1, \ldots, M$, required to start the MoC-pRK3, is computed by the MoC-ME scheme, whose error at one step is $O(h^3)$. This error will only propagate to all subsequent levels, but will not accumulate (unlike the local truncation errors), and hence is consistent with the desired $O(h^3)$ global error of the MoC-pRK3. Third, at $n = 3$, the solution $(\mathbf{y}^+)^3_{m\geq 2}$ and $(\mathbf{y}^-)^3_{m\geq 1}$ is found by the in-bulk algorithm, while $(\mathbf{y}^+)^3_1$ is found from the BC (23b). (Recall that we concentrate on the vicinity of the left boundary only.) Note that determining $(\mathbf{y}^+)^3_2$ requires, as per the $\boldsymbol{\kappa}^+_{1,2}$-terms in (44a), the solution $(\mathbf{y}^{\pm})^1_0$, which has been determined at the first step above.

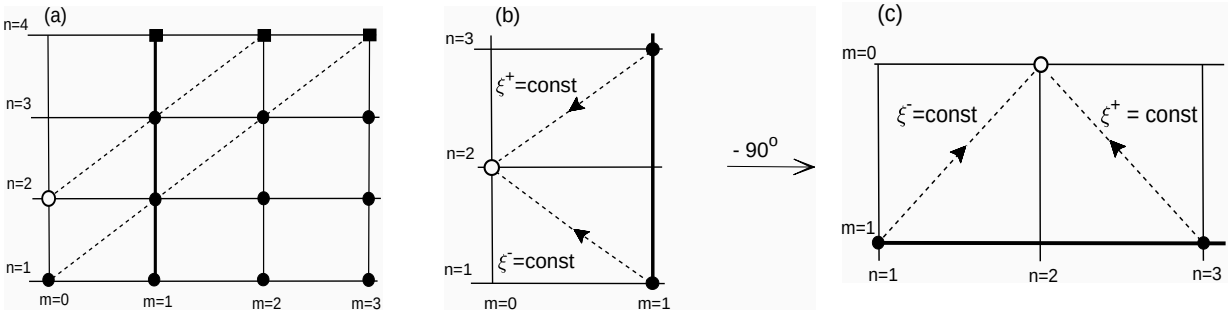

Figure 9: Schematics illustrating the treatment of the nonreflecting BC at the left boundary for the MoC-pRK3. See text for details.

A nontrivial extension of the algorithm is required to compute $(\mathbf{y}^+)_2^4$, as this requires the yet undetermined solutions $(\mathbf{y}^\pm)_0^2$. As we have emphasized above, these solutions will be computed by a method *different* from the extrapolation (57). Note that these solutions affect $(\mathbf{y}^+)_2^4$ only via $h\,(\boldsymbol{\kappa}^+_{1,2})_0^2$ and therefore need to be found only with local error $O(h^3)$ in order to guarantee that the local truncation error of $(\mathbf{y}^+)_2^4$ is $h \cdot O(h^3) = O(h^4)$. This suggests that $(\mathbf{y}^\pm)_0^2$ can be found by the MoC-ME. However, it is clear from Fig. 9(a) that the standard MoC-ME, whose stencil is given by the three circles in Fig. 1, cannot be used for this purpose. The *key trick* that enables the use of the MoC-ME is to employ the *rotated stencil*, as shown in Figs. 9(b,c). Indeed, the solutions $(\mathbf{y}^\pm)_1^1$ and $(\mathbf{y}^\pm)_1^3$ have already been found at the previous steps. Then, rotating the stencil shown in Fig. 9(b) by $-90°$, one obtains the standard MoC-ME stencil in Fig. 9(c). The corresponding equations are (for $n = 2$):

$$(\mathbf{y}^\pm_{(1)})_0^n = (\mathbf{y}^\pm)_1^{n\pm1} \mp h\,\mathbf{f}^\pm\big(\,(\mathbf{y}^+)_1^{n\pm1}, (\mathbf{y}^-)_1^{n\pm1}\,\big)\,, \tag{58a}$$

$$(\mathbf{y}^\pm_{(2)})_0^n = \frac{1}{2}\,\Big[(\mathbf{y}^\pm)_1^{n\pm1} + (\mathbf{y}^\pm_{(1)})_0^n \mp h\,\mathbf{f}^\pm\big(\,(\mathbf{y}^+_{(1)})_0^n, (\mathbf{y}^-_{(1)})_0^n\,\big)\Big]\,. \tag{58b}$$

Note that the negative sign in front of h in the expressions for $\mathbf{y}^+_{(1),(2)}$ occurs because the corresponding "steps" are taken in the negative direction along the characteristic $\xi^+ = \text{const}$; see Fig. 9(b). At the right boundary, the signs in front of the $\mathbf{f}^\pm$-terms switch compared to those in (58).

Now that $(\mathbf{y}^\pm)_0^2$ (and the analogous solutions just outside the right boundary), and hence the solutions $(\mathbf{y}^+)_2^4$ (and $(\mathbf{y}^-)_{M-1}^4$), have been found, the remaining solutions at time level with $n = 4$ are found by the in-bulk algorithm. The solution for all time levels with $n > 4$ follows the same pattern, and the pseudocode is shown in Appendix B.

It is reasonable to ask if in the above algorithm one could avoid using virtual nodes (at the left boundary, those are nodes with $m = 0$) altogether by computing *only* the solutions $(\mathbf{y}^-)_1^{n\geq3}$ and $(\mathbf{y}^+)_2^{n\geq3}$ by the MoC-RK3 (and similarly at the right boundary), while computing the rest of the solution by the in-bulk MoC-pRK3. The answer to this is 'no'. The reason is that even with these

four nodes out of the entire grid being computed by the unstable MoC-RK3 scheme (see Section 3b) renders this "combined MoC-(RK3 & pRK3)" scheme similarly unstable. We verified this by simulations with $\mathbf{P} = \mathbf{P}_4$ with $a = 0$ in [34] (see Sec. 5.9 there).

On the other hand, simulations following the "MoC-pRK3 only" scheme with the boundary treatment described above show that the mild numerical instability for the intermediate wavenumbers is completely suppressed: compare Figs. 8(a) and 10(a). (In this regard, we will comment on a relatively small growth — by about an order of magnitude over $t = 1000$ — that is visible as the "bumps" around the sharp "dips" at the left and right edges of Fig. 10(a) for $\mathbf{P} = \mathbf{P}_1$. We verified that this growth is *linear*, not exponential, in time, and therefore can affect only ultra-long simulations, on the order of many millions of time units.)

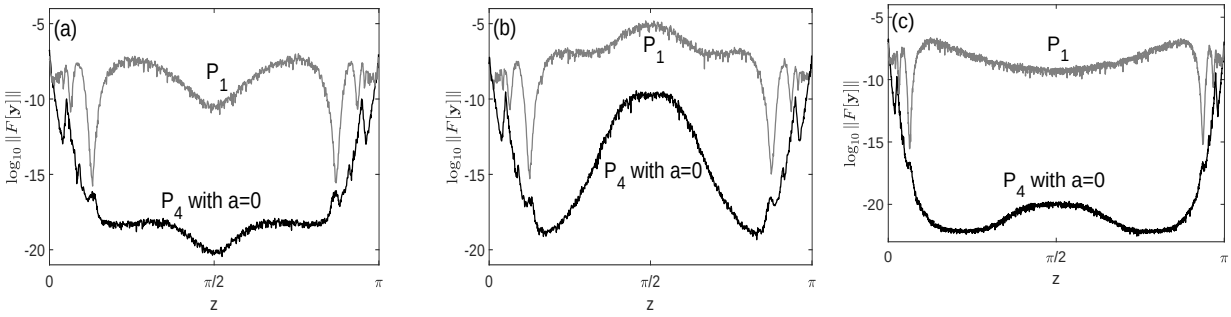

Figure 10: (a), (b): Same schemes and parameters as in Figs. 8(a,c), but with homogeneous nonreflecting BC (23). (c): Same as (b), but $h = 0.025$.

7.2 Nonreflecting BC for the MoC-pRK4 (46), (14)

The stencil for the MoC-pRK4 scheme in the vicinity of the left boundary is shown in Fig. 11(a). The initial condition is given at the time level with $n = 1$ for $m = 1, \ldots, M$. Since the pRK4 is a three-step solver, it requires the solution at three time levels to start the calculations. These solutions are to be computed with local accuracy $O(h^4)$. The only available option to obtain such a solution is the MoC-RK3. Even though that scheme can be strongly unstable when carrying out calculations for $t = O(1)$, it is acceptable to use it just for two time levels. Next, similarly to the situation with the MoC-pRK3, one obtains the solutions at the virtual nodes ($n = 1, 2$; $m = 0, -1$) by the 4th-order Lagrange extrapolation:

$$(\mathbf{y}^{\pm})^n_m = 4(\mathbf{y}^{\pm})^n_{m+1} - 6(\mathbf{y}^{\pm})^n_{m+2} + 4(\mathbf{y}^{\pm})^n_{m+3} - (\mathbf{y}^{\pm})^n_{m+4}; \qquad n = 1, 2; \quad m = 0, -1\,. \tag{59}$$

We emphasize that, as for the MoC-pRK3, this extrapolation will *not* be used at subsequent time levels. The solution at time level t_4 now has all the ingredients to be computed by the in-bulk algorithm (46); see Fig. 11(a).

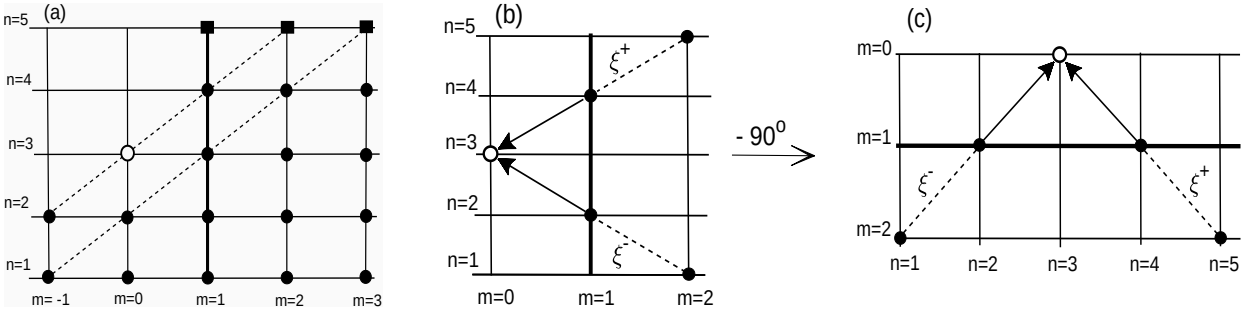

Figure 11: Schematics of computing the MoC-pRK4 solution with nonreflecting BC at the left boundary. (a) Filled squares indicate that a nontrivial computation involving nodes outside the boundary first occurs at the time level with $n = 5$. The open circle shows the node where this nontrivial computation is required. (b) Stencil to compute $(\mathbf{y}^{\pm})_0^3$ by the MoC-pRK3. (c) Same stencil rotated by $-90°$.

A nontrivial step first occurs in computing the solution at the next time level, t_5. It will contain the trick with the "rotated stencil" as in the MoC-pRK3 case, *as well as an additional twist.* To find $(\mathbf{y}^{+})_2^5$, one requires $(\mathbf{y}^{\pm})_0^3$. In analogy with the MoC-pRK3 case, it needs to be computed with error $O(h^4)$. At first sight, we have two schemes that could accomplish that task: the MoC-RK3 and MoC-pRK3. However, even the stability considerations aside, the MoC-RK3 scheme *cannot* be used. Indeed, one is unable to use the regular (i.e., not rotated) stencil for it since, as explained in the last paragraph of Section 3.1, that would require the knowledge of $(\mathbf{y}^{+})_{-1}^3$, which is not available. Similarly, if one instead uses the rotated stencil, that would require the knowledge of $(\mathbf{y}^{+})_0^4$, which is not available, either. Therefore, the only available option is to find $(\mathbf{y}^{\pm})_0^3$ is by the MoC-pRK3 with the rotated stencil, as shown in Figs. 11(b) and (c). In this stencil, we already know the solution at nodes $(n, m) = (1, 2)$, $(2, 1)$, $(3, 2)$, and $(4, 1)$. However, we do not know the solution $\mathbf{y}^{+}$ at node $(5, 2)$, because this is precisely the solution that we need $(\mathbf{y}^{\pm})_0^3$ for!

The way out of this seemingly vicious circle follows from the observation that the $(\mathbf{y}^{+})_2^5$ which is required to compute $(\mathbf{y}^{\pm})_0^3$ by the MoC-pRK3 needs to be computed only with the local error $O(h^3)$. (In contrast, the $(\mathbf{y}^{+})_2^5$ which will be computed by the MoC-pRK4 must have the local error $O(h^5)$.) Thus, to compute $(\mathbf{y}^{\pm})_0^3$ by the "rotated MoC-pRK3", it will suffice to compute $(\mathbf{y}^{+})_2^5$ by the MoC-ME. This can be done readily using the regular (i.e., not rotated) MoC-ME stencil and the available solution at time level t_4. Thus, to compute $(\mathbf{y}^{\pm})_0^3$, one proceeds as follows. First, compute $(\mathbf{y}^{\pm}_{(2)})_2^5$ by the MoC-ME (18) using $(\mathbf{y}^{\pm})_{1,3}^4$. Then, compute $(\mathbf{y}^{\pm})_0^3$ by the rotated MoC-pRK3 using the stencil shown in Figs. 11(b,c). The corresponding equations are (for $n = 3$):

$$(\mathbf{y}^{\pm})_0^n = (\mathbf{y}^{\pm})_1^{n\pm1} \mp \frac{h}{12}\left(13(\boldsymbol{\kappa}_1^{\pm})_1^{n\pm1} + 5(\boldsymbol{\kappa}_2^{\pm})_1^{n\pm1} - (\boldsymbol{\kappa}_1^{\pm})_2^{n\pm2} - 5(\boldsymbol{\kappa}_2^{\pm})_2^{n\pm2}\right), \tag{60a}$$

with $\boldsymbol{\kappa}_1^{\pm}$ being given by (14) and

$$(\boldsymbol{\kappa}_2^{+})_1^{n+1} = \mathbf{f}^{+}\big((\mathbf{y}_{(1)}^{+})_0^{n},\, (\mathbf{y}_{(2)}^{-})_0^{n} \big), \qquad (\boldsymbol{\kappa}_2^{-})_1^{n-1} = \mathbf{f}^{-}\big((\mathbf{y}_{(2)}^{+})_0^{n},\, (\mathbf{y}_{(1)}^{-})_0^{n} \big), \tag{60b}$$

$$(\boldsymbol{\kappa}_2^{+})_2^{n+2} = \mathbf{f}^{+}\big((\mathbf{y}_{(1)}^{+})_1^{n+1},\, (\mathbf{y}^{-})_1^{n+1} \big), \qquad (\boldsymbol{\kappa}_2^{-})_2^{n-2} = \mathbf{f}^{-}\big((\mathbf{y}^{+})_1^{n-1},\, (\mathbf{y}_{(1)}^{-})_1^{n-1} \big), \tag{60c}$$

where $\mathbf{y}_{(1)}^{\pm}$ and $\mathbf{y}_{(2)}^{\pm}$ are computed by (58). Note that, as previously in (58), the "steps" along the $\xi^{+} = \text{const}$ characteristics are taken with increment $(-h)$, not h.

Having computed $(\mathbf{y}^{\pm})_0^3$, one can then compute $(\mathbf{y}^{+})_{2\le m\le M}^5$ and $(\mathbf{y}^{-})_{1\le m\le M-1}^5$ by the in-bulk algorithm (46), while $(\mathbf{y}^{+})_1^5$ and $(\mathbf{y}^{-})_M^5$ are supplied by the BC (23b). For the purpose of generalizing this step for $n \ge 5$, we note that after computing the solution at level n (in Fig. 11(a), $n = 4$) for $m = 1, \dots, M$, one then needs to compute the solution at the nodes $(n-1, 0)$ and $(n-1, M+1)$, as that will be needed to advance to level $(n+1)$. In particular, having found the solution $(\mathbf{y}^{\pm})_m^5$, $m = 1, \dots, M$, we then compute $(\mathbf{y}^{\pm})_0^4$ and $(\mathbf{y}^{\pm})_{M+1}^4$.

At time level t_6, we again need to deviate from the in-bulk algorithm when computing $(\mathbf{y}^{+})_2^6$ (and similarly at the right boundary), as now $(\mathbf{y}^{\pm})_{-1}^3$ is not available. The latter values are required with local accuracy $O(h^4)$ and can be found by the "rotated MoC-pRK3" using the already available solutions at nodes $(n, m) = (1, 1)$, $(2, 0)$, $(3, 1)$, $(4, 0)$, and $(5, 2)$. The trick with using the MoC-ME, as described two paragraphs above, is *not* needed here. The remaining steps will follow a similar pattern. Namely, once the solution is found at level t_n for $m = 1, \dots, M$, first compute $(\mathbf{y}^{\pm})_{0,M+1}^{n-1}$ by the "rotated MoC-pRK3" (60); then compute $(\mathbf{y}^{\pm})_{1\le M}^{n+1}$ by the MoC-pRK4; and, finally, compute $(\mathbf{y}^{\pm})_{-1,M+2}^{n-1}$ by the "rotated MoC-pRK3". One is now ready to compute $(\mathbf{y}^{\pm})_{1\le M}^{n+2}$; and so on. The corresponding pseudocode is presented in Appendix B.

Comparison of Figs. 8(a,c) with 10(a,b) shows that the ability of nonreflecting BC to suppress the instability for the MoC-pRK4 is less than that for MoC-pRK3. In particular, for $h = 0.05$, this suppression is sufficient to eliminate the instability for some $\mathbf{P}$-matrices (namely, $\mathbf{P}_4$), but not all of them: for $\mathbf{P}_1$, the harmonics near $|k| = k_{\max}/2$ are seen to still grow by some three orders of magnitude over $t = 1000$. In [28], we showed analytically that for the MoC-SE and MoC-ME schemes, the smaller h (or, more precisely, hL), the stronger the unstable modes get suppressed by the nonreflecting BCs. While we do not carry out a similar analysis for the MoC-pRK schemes (which would have required a separate study), we can hypothesize that the same phenomenon should take place for them. This hypothesis is confirmed by numerical simulations: when we used $h = 0.025$ instead of $h = 0.05$, the instability near $|k| = k_{\max}/2$ got suppressed; see Fig. 10(c). The "bumps" on both sides of the sharp "dips" near the edges of the figure pertain to harmonics that grow linearly, not exponentially, in time, and hence do not affect any but ultra-long simulations; see a similar note

for the MoC-pRK3.

8 Numerical verification

Here we will verify that the MoC-(N)pRK schemes developed in this work indeed have the accuracy declared. Due to space limitation, we will explicitly do so only for the fourth-order schemes ([34] contains numerical results pertaining to the third-order schemes). However, note that the MoC-pRK4 scheme uses the MoC-pRK3 solution at an intermediate step. Therefore, the fact that the MoC-pRK4 solution is shown to have the error $O(h^4)$ *implies* that the MoC-pRK3 solution must have the error of at most $O(h^3)$. Similarly, since the MoC-pRK4 scheme uses the MoC-RK3 solution to start the calculations, our results will also imply that the error of the MoC-RK3 solution is at most $O(h^3)$.

Our verification will be restricted to two equations, one nonlinear and one linear. The former is the Gross–Neveu model [39] of the relativistic field theory, written in the form (1a):

$$\begin{aligned} u^+_t + u^+_x &= i(\,|u^-|^2 u^+ + (u^-)^2 (u^+)^*\,) - i u^-, \\ u^-_t - u^-_x &= i(\,|u^+|^2 u^- + (u^+)^2 (u^-)^*\,) - i u^+. \end{aligned} \tag{61}$$

Numerical schemes for (61) have attracted considerable attention in the last few years; see, e.g., [40, 41] and references therein. We will simulate two of its exact soutions. The first one is the standing soliton [42]:

$$u^\pm_{\rm sol}(x,t) \;=\; \sqrt{1-\Omega}\;\frac{\cosh(\beta x) \;\pm\; i\mu\,\sinh(\beta x)}{\cosh^2(\beta x) - \mu^2\,\sinh^2(\beta x)}\;\exp[-i\Omega t + i\phi_0], \qquad \Omega \in (0,1); \tag{62a}$$

with $\phi_0 = \text{const}$ and

$$\beta = \sqrt{1-\Omega^2}, \qquad \mu = \sqrt{(1-\Omega)/(1+\Omega)}\,. \tag{62b}$$

The second solution is the soliton moving with velocity V. It is obtained from (62) by a transformation related to the Lorentz transformation:

$$u^\pm_{\rm mov}(x,t) \;=\; \left(\frac{\sqrt{\Gamma+1}+\sqrt{\Gamma-1}}{\sqrt{2}}\right)^{\pm 1}\; u^\pm(x_{\rm mov},\, t_{\rm mov}), \tag{63a}$$

where

$$\Gamma = 1/\sqrt{1-V^2}, \qquad x_{\rm mov} = \Gamma\,(x - x_0 - Vt), \quad t_{\rm mov} = \Gamma\,(t - V(x-x_0)) \tag{63b}$$

and $x_0 = \text{const}$. Equation (63a) can be obtained from that found in, e.g., [43] by a simple dependent variable transformation. The Gross–Neveu model admits, in particular, two conserved quantities, charge Q and Hamiltonian H:

$$Q = \int_{-\infty}^{\infty} \left(|u^+|^2 + |u^-|^2\right)\, dx\,; \tag{64a}$$

$$H = \int_{-\infty}^{\infty} \left(-i\big((u^+)^* u_x^+ - (u^-)^* u_x^-\big) - \frac{1}{2}\left(u^+(u^-)^* + u^-(u^+)^*\right)^2 + \left(u^+(u^-)^* + u^-(u^+)^*\right) \right) dx\,. \tag{64b}$$

For the soliton (62), they take on values:

$$Q = 2\,\frac{\sqrt{1-\Omega^2}}{\Omega}, \qquad H = 2\,\ln\frac{\Omega}{1-\sqrt{1-\Omega^2}}\,. \tag{64c}$$

This model has, in the notations of Section 2, $N^{\pm} = 1$. Therefore, the MoC-RK4 (and MoC-RK3) scheme(s) will be stable or only weakly unstable for it (depending on the boundary conditions used). However, we will not test their performance relative to those of the MoC-(N)pRK schemes due to, again, space limitation, as well as in order to maintain our focus on the latter schemes. It should be noted that the 4th-order method proposed in [40] is (unlike the MoC-(p)RK4) restricted to using periodic BC; it was demonstrated in [41] that certain solutions (62) exhibit numerical instability for periodic (but not for nonreflecting) BC.

The second equation is the linear Klein–Gordon equation (2) with $c = 1$ and $g \equiv u$. We simulated it as system (5a) with $\mathbf{P} = \mathbf{P}_1$, given by (7), with the purpose being to demonstrate an excellent ability of the MoC-pRK4 to accurately compute very steep spatial fronts.

In all simulations in this section we set the length of the computational domain to $L = 100$.

8.1 Gross–Neveu with periodic BC

The standing soliton (62) is zero to numerical precision at the boundaries of the computational domain $x \in [-L/2,\, L/2)$, and hence it satisfies periodic BC. Two facts that we intend to demonstrate about the MoC-pRK4 scheme here are: (i) that its error scales as $O(h^4)$ and (ii) that it preserves conserved quantities (64) up to order $O(h^5)$ for moderately long times.[4] For these demonstrations, we simulate the solitons (62) with $\Omega = 0.3$ and 0.6, shown in Fig. 12, up to $t = 200$. A technical aspect about computing the numerical error of a soliton solution over such a moderately long time is addresses in Appendix C.

The dependence of the errors of the MoC-pRK4 and MoC-NpRK4 schemes on h is shown in Fig. 13(a) and is seen to be $O(h^4)$. The corresponding dependences of relative errors in Q and H are shown in Fig. 14(a,b) and are seen to be $O(h^5)$. Note that this dependence is the same as the well-known scaling of the errors of conserved quantities computed by the RK4 solvers for ODEs. Finally, Fig. 14(c) illustrates that one can significantly improve the ability of the MoC-NpRK4 scheme to preserve conserved quantities by optimizing the free parameter in the NpRK4 solver (see Section 5.2).

[4] As we noted in Section 6, for *very* long times, the numerical solution with periodic BC will be affected by the weak instability, which will occur primarily for wavenumbers in the interval $(k_{\max}/4,\, 3k_{\max}/4)$.

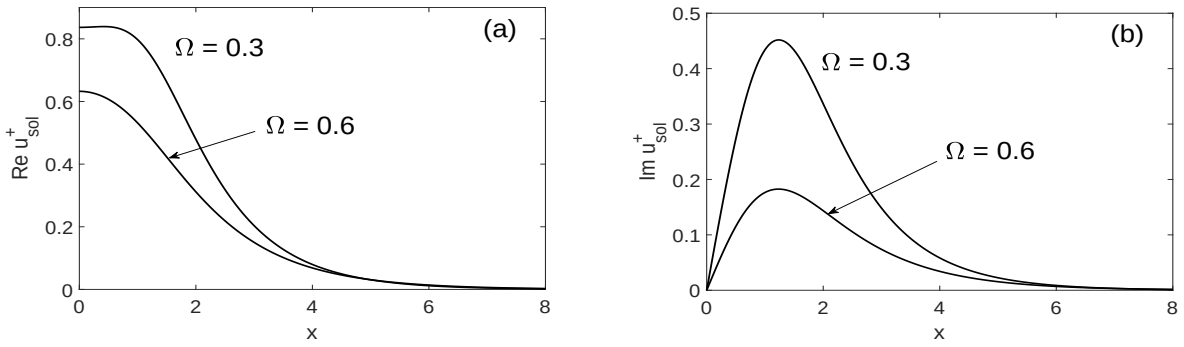

Figure 12: Real and imaginary parts of $u^+_{\rm sol}(x,0)$ for two values of Ω used in Section 8. By symmetry, $\mathrm{Re}\, u^+_{\rm sol}(-x,0) = \mathrm{Re}\, u^+_{\rm sol}(x,0)$, $\mathrm{Im}\, u^+_{\rm sol}(-x,0) = -\mathrm{Im}\, u^+_{\rm sol}(x,0)$.

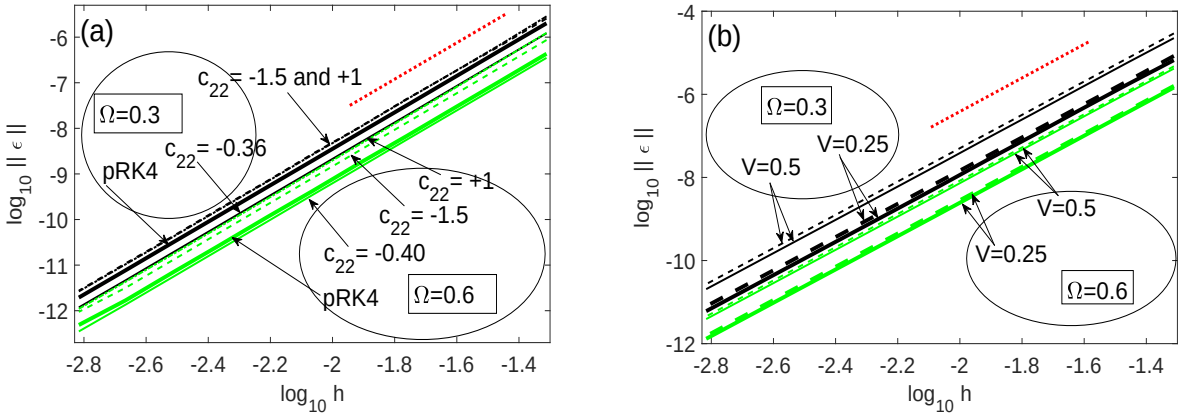

Figure 13: (Color online) (a): Error (see (78b)) of MoC-pRK4 (thick line) and MoC-NpRK4 (thin lines) for the solitons with $\Omega = 0.3$ (black) and 0.6 (green) of the Gross–Neveu model with periodic BC versus h. Thin dashed and dashed–dotted lines correspond to $c_{22} = -1.5$ and 1.0, respectively. The lines for $c_{22} = -1.5$ and 1.0 for $\Omega = 0.3$ are indistinguishable in the plot. The thin solid lines are for the optimal values of c_{22} (see Fig. 14): -0.36 for $\Omega = 0.3$ and -0.40 for $\Omega = 0.6$. The red dotted line has the slope of 4. Simulations were done for $h = L/2^{11+0.5j}$, $j = 0, \ldots, 10$. (b): Same, but for nonreflecting BC and for the MoC-pRK4 only. Simulation parameters (except t; see text) and colors are the same as in (a). Thick and thin lines are for $V = 0.25$ and 0.5, respectively, where V is the velocity of the incident soliton. Solid and dashed lines are for $x_0 = -0.4$ and $+0.4$, where x_0 is the position of the soliton center at $t = 0$ relative to the left boundary of the computational domain.

Let us note that Figs. 13(a) and 14(a,b) show that while optimization of the free parameter in the MoC-NpRK4 leads to a significant reduction of the drifts of conserved quantities, it has only minor effect on the error of the solution. This is explained by the fact that the error at $t = 200$ is computed relative *not* to the exact soliton (62) but to the soliton whose Ω is adjusted (see Appendix C) according to the varying Q (or, equivalently, H: see (64c)). Thus, the error whose norm is plotted in Fig. 13(a) does not contribute to the drift of Q and H.

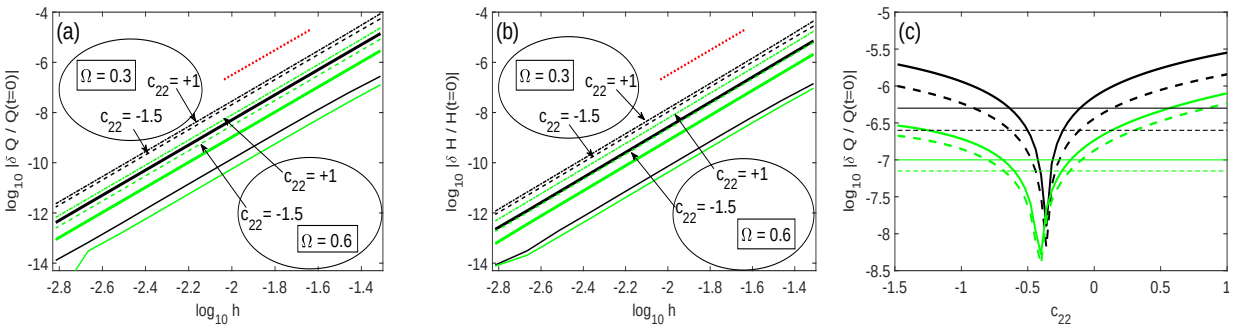

Figure 14: (Color online) (a) and (b): Similar to Fig. 13(a), but for the errors in Q (a) and H (b). The red dotted line has the slope of 5. In (b), the lines corresponding to the MoC-pRK4 for $\Omega = 0.3$ and MoC-NpRK4 with $c_{22} = -1.5$ for $\Omega = 0.6$ can be distinguished only in color. (c): Thick lines: Errors in Q (solid) and H (dashed) versus c_{22} for MoC-NpRK4 for $\Omega = 0.3$ (black) and $\Omega = 0.6$ (green); $h = 0.025$. The thin horizontal lines with respective colors and line styles show the errors for the MoC-pRK4.

8.2 Gross–Neveu with nonreflecting BC

In order to confirm that the algorithm of handling nonreflecting BC presented in Section 7 preserves the order $O(h^4)$ of the MoC-pRK4 scheme, we simulated the entering of a soliton (63) into a medium governed by Eqs. (61). The corresponding BC, and the initial condition consistent with them, are:

$$u^+(0,t) = u^+_{\rm mov}(0,t), \qquad u^-(L,t) = 0\,; \tag{65a}$$

$$u^\pm(x,0) = u^\pm_{\rm mov}(x,0)\,. \tag{65b}$$

For reasons explained in Appendix C, the simulation time needs to be much shorter than that in Section 8.1, and we used $t = 5$. During this simulation time, and for the parameters V and x_0 of the incident soliton reported in the caption to Fig. 13(b), the field at the boundary is essentially nonzero, as follows from Fig. 12. The error for the MoC-pRK4 is shown in Fig. 13(b) and is seen to scale as $O(h^4)$. We did not show corresponding results for the MoC-NpRK4 scheme because in the pulse-entering problem, one is not concerned with preservation of the conserved quantities, as they, naturally, vary as the pulse enters the medium.

8.3 Linear Klein–Gordon

Here we demonstrate convergence of the MoC-pRK4 scheme for Eq. (2) with $c = 1$ and $g \equiv u$ and an initial condition with steep fronts (see Fig. 15(a)):

$$u(x,0) = \exp\big[-x^q\big], \quad q = 10 \text{ or } 20; \qquad u_t(x,0) = 0. \tag{66}$$

In what follows we limit out simulation time to $t = 10$, whereby the field does not have the time to reach the boundaries of the computational window located at $x = \pm L/2$; recall that $L = 100$ in this

section. Therefore, the corresponding exact solution coincides with that on the infinite line:

$$u(x,t) = \int_{-\infty}^{\infty} e^{ikx}\, \widehat{u}_0(k) \cos(\sqrt{k^2+1}\, t)\, dk, \qquad \widehat{u}_0(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ikx} u(x,0)\, dx\,. \tag{67}$$

The transformation between the linear Klein–Gordon equation and its form (5) is given by:

$$y_1^{\pm} = (\pm p - u)/2, \quad y_2^{\pm} = (v \pm w)/2, \tag{68}$$

where $v = u_t$ and $w = -u_x$, and p is an auxiliary variable satisfying compatibility conditions: $p_t = u_x + w$ and $p_x = u_t - v$ (see [15]), with $p(x,0) = 0$. In our simulations, we had vectors $\mathbf{y}^{\pm}$ satisfy the nonreflecting BC (23) with $\mathbf{b}_{\text{left,right}} = 0$.

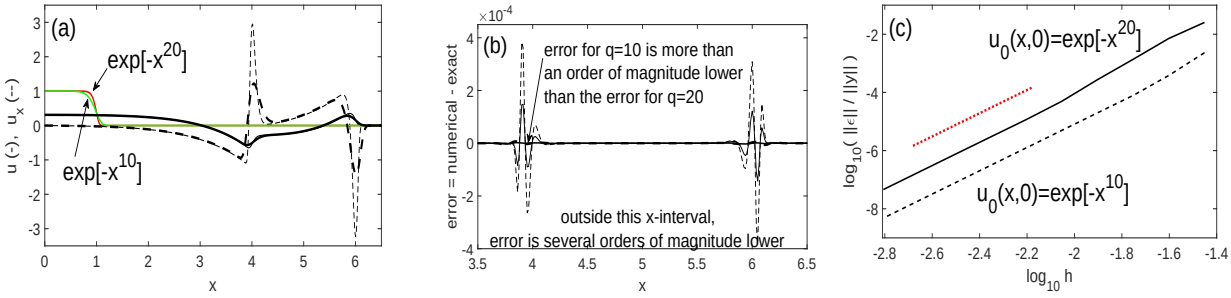

Figure 15: (Color online) (a): Solution (solid black lines) of (5) with (7) (equivalent to (2) with $g \equiv u$) for each of the initial conditions (66) (shown in color), obtained with $h = 0.0125$ at $t = 5$; it is symmetric with respect to $x = 0$. Solution at $t = 10$ it is similar but more spread out. Since in some applications, it is either u_t or u_x that has physical meaning, we also show u_x (dashed lines). Thick (thin) black lines correspond to the solution for the initial condition with $q = 10$ ($q = 20$). These two lines are almost indistinguishable in the plot. (b) Error for u and u_x of the solutions shown in (a); line styles mean the same as in (a). (c) Relative ℓ_2-error of the solution of (5), (7) at $t = 10$ for both initial conditions. The red dotted line has slope four. Since the $y_2^{\pm}$-components of the solution involve $w = -u_x$, the ℓ_2-norm of the $\mathbf{y}$-error is proportional to the Sobolev norm of the u-error.

Figure 15(a,b) shows that the MoC-PRK4 scheme computes the steep fronts of the solution very accurately, and Fig. 15(c) further shows that even for those steep fronts, the scheme preserves the $O(h^4)$ order of the error not only in the maximum norm of u but also in its Sobolev norm.

9 Summary and discussion

We have considered systems of (N^+, N^-) coupled first-order hyperbolic equations of the form (1a) in one spatial dimension. We have constructed, for the first time to our knowledge, 3rd- and 4th-order MoC schemes based on the explicit ODE solvers. When employing RK solvers, we found (in Sections 3 and 4) that while the resulting MoC-RK schemes were essentially stable[5] for $N^{\pm} = 1$ and

[5]Here the word 'essentially' refers to the fact that instability with the respective growth rates $O(h^3)$ and $O(h^5)$ still remains (see the end of Section 1), but can be ignored as inconsequential for most applications.

for some cases with $N^{\pm}=2$, they were strongly unstable for other $N^{\pm}=2$ cases, including that of the important Klein–Gordon equation. Based on our extensive experimentation with various ODE solvers, we hypothesized that the common feature of all such unstable schemes is that they involve virtual nodes (i.e., nodes off the grid; see Fig. 1). Consequently, we proposed to use pRK solvers (Section 5) instead of RK ones, and the resulting MoC-pRK schemes (Section 6) were found to be only weakly unstable, with the instability growth rate being $O(h)$, for periodic BC. Importantly, we further demonstrated (in Section 7) that this weak instability disappears for nonreflecting BC (23), which are more common in physical applications than periodic ones. It should be noted that implementation of nonreflecting BC required non-trivial treatment of the near-boundary nodes, as explained in Section 7. A link to the codes themselves is found at [30]. For partially reflecting BC (56), which generalize the nonreflecting BC, our MoC-pRK schemes are implemented similarly. The MoC-pRK schemes are capable of resolving sharp fronts of the solution without introducing spurious oscillations (see Section 8.3).

In the remainder of this section we will discuss extensions of our schemes to systems more general than (1). First of all, MoC-pRK schemes are easily extended from (1a) to the following system with *three* characteristics:

$$\mathbf{y}^+_t+\mathbf{y}^+_x=\mathbf{f}^+\left(\mathbf{y}^+,\mathbf{y}^-,\mathbf{n}\right),\qquad \mathbf{y}^-_t-\mathbf{y}^-_x=\mathbf{f}^-\left(\mathbf{y}^+,\mathbf{y}^-,\mathbf{n}\right),\qquad \mathbf{n}_t=\mathbf{f}^{\mathrm{med}}\left(\mathbf{y}^+,\mathbf{y}^-,\mathbf{n}\right). \tag{69}$$

They arise when the waves $\mathbf{y}^{\pm}$ interact with a medium, described by variable $\mathbf{n}$; examples are electromagnetic propagation in distributed-feedback semiconductor lasers (e.g., [44, 5] and references therein) and Stimulated Brillouin Scattering (see, e.g., [45]). Below we illustrate the extension of the schemes developed in Section 6 to systems (69) using the MoC-pRK3 as an example; it is conceptually similar for the MoC-pRK4.

The extension of the general algorithm (i.e., disregarding boundaries) is straightforward: Eqs. (14) are replaced with

$$(\boldsymbol{\kappa}_1^j)_m^n=\mathbf{f}^j\left((\mathbf{y}^+)_m^n,(\mathbf{y}^-)_m^n,(\mathbf{n})_m^n\right),\qquad j\in\{\pm,\,\mathrm{med}\}; \tag{70a}$$

Eqs. (18) are supplemented with

$$(\mathbf{n}_{(1)})_m^{n+1}=(\mathbf{n})_m^n+h\,(\boldsymbol{\kappa}_1^{\mathrm{med}})_m^n;\qquad (\mathbf{n}_{(2)})_m^{n+1}=\frac{1}{2}\left[(\mathbf{n})_m^n+(\mathbf{n}_{(1)})_m^{n+1}+h\,\mathbf{f}^{\mathrm{med}}\left((\mathbf{y}^+_{(1)})_m^n,(\mathbf{y}^-_{(1)})_m^n,(\mathbf{n}_{(1)})_m^n\right)\right]; \tag{70b}$$

in Eqs. (44b), the third argument, $(\mathbf{n}_{(2)})_m^{n+1}$, is added to $\mathbf{f}^{\pm}$ and one also sets

$$(\boldsymbol{\kappa}_2^{\mathrm{med}})_m^n=\mathbf{f}^{\mathrm{med}}\left((\mathbf{y}^+_{(2)})_m^{n+1},(\mathbf{y}^-_{(2)})_m^{n+1},(\mathbf{n}_{(1)})_m^{n+1}\right); \tag{70c}$$

finally, Eqs. (44a) are supplemented with

$$(\mathbf{n})_m^{n+1} = (\mathbf{n})_m^n + \frac{h}{12}\left(13(\boldsymbol{\kappa}_1^{\text{med}})_m^n + 5(\boldsymbol{\kappa}_2^{\text{med}})_m^n - (\boldsymbol{\kappa}_1^{\text{med}})_m^{n-1} - 5(\boldsymbol{\kappa}_2^{\text{med}})_m^{n-1}\right). \tag{70d}$$

To illustrate the new (compared to that treated in Section 7.1) issue that arises in the computation of the outside-the-boundary nodes, let us consider the computation of the fields at node $(n, m) = (2, 0)$; see Fig. 9. In order to compute $(\mathbf{y}^{\pm})_0^2$, one requires the knowledge of $(\mathbf{n}_{(1)})_0^2$. The following two key realizations should be made here. First, one *cannot* compute $(\mathbf{n}_{(1)})_0^2$ using the rotated stencil in Fig. 9(b,c) and the first equation in (70b) because it would have required an equation for $\mathbf{n}_x$, which is not available (see (69)). Second, $(\mathbf{n}_{(1)})_0^2$ does *not* need to be computed using specifically the stencil in Fig. 9(b,c); it just needs to be computed *in some way* that is consistent with (69). Such a way is to simply use the first equation in (70b) with $n = 1$ and $m = 0$, all ingredients for which have already been found. Finally, one computes $(\mathbf{n}_{(2)})_0^2$ using the second equation in (70b) with $n = 1$ and $m = 0$.

Another generalization of Eqs. (1) includes advection-dominated problems where small diffusion is present in $\mathbf{f}^{\pm}$. Numerical implementation of such terms is straightforward: one simply computes $\mathbf{y}_{xx}^{\pm}$ within one time level. However, their presence changes the Jacobian matrix $\mathbf{P}$ in Eqs. (5) so that its eigenvalues now lie in the left half of the complex plane. This will require redoing stability analysis of the MoC-RK and MoC-pRK schemes with a new $\mathbf{P}$. It is even possible that the strong instability of the MoC-RK schemes could be suppressed by dissipation. On the other hand, one should keep in mind that the stability region of MoC-pRK schemes can be smaller than that MoC-RK schemes: see Fig. 4.

Let us now comment on the issue of generalizing higher-order MoC schemes to two spatial dimensions. A well-known method to apply the MoC of order up to 2 on a rectangular spatial grid is by using operator splitting (OS), applying a one-dimensional scheme along one dimension at a time over a substep Δt_i, where $\sum_{i=1}^{s} \Delta t_i = \Delta t$ and s is the number of stages in OS. However, if one intends to avoid interpolation, one must use Δt_i that are integer multiples of the spatial steps $\Delta x = \Delta y = \Delta t$. The problem is that no OS scheme with such Δt_i is known to have the order higher than 2, and using a second-order accurate OS will annihilate the advantage of using a higher-order MoC scheme per each spatial dimension. Therefore, the extension of higher-order MoC schemes to several spatial dimensions is a nontrivial open problem.

Finally, let us comment on the issue of comparison of the MoC-pRK4, proposed in this work, with the MoC scheme using an *implicit* RK4 solver, proposed in [26]. For brevity, we will refer to it

as the MoC-iRK4 scheme. Let us note that it implements the implicit step as a predictor-corrector, with the number of corrector stages advocated in [26] being three. First, and most importantly, we note that stability of the MoC-iRK4 has not been studied analytically, and, to our knowledge, applications of that scheme were reported only to equations (1a) with $N^{\pm} = 1$ or $N^{\pm} = 2$ with $\mathbf{P} = \mathbf{P}_2$ [2]. Thus, the question whether MoC-iRK4 is stable for $N^{\pm} = 2$ and the other four $\mathbf{P}$-matrices listed in Section 2, remains open. We will address it in a future publication.

Therefore, here we will limit our task to comparison of the number of function evaluations (FE) per variable in the MoC-iRK4 and MoC-pRK4 schemes, which should give one a rough idea about relative execution times for the two schemes. The predictor stage of the MoC-iRK4, given by Eqs. (8) in [26], requires 2 FEs (with $f(x_n, y_n)$ being saved from the previous step), and each corrector stage, given by Eqs. (7) there, requires 2 more. Thus, for one predictor and three corrector stages (see above), one needs 8 FEs per variable. In the MoC-pRK4, one computes $\boldsymbol{\kappa}_1$ by (14) and $\boldsymbol{\kappa}_2$ by (46b) at each step, which amounts to 2 FEs. Next, the computation of $\boldsymbol{\kappa}_2$ requires $\mathbf{y}_{(3)}$, which in turn requires *its* $\boldsymbol{\kappa}_1$ (same as above) and $\boldsymbol{\kappa}_2$ (Eqs. (44b)); this is 1 more FE. Finally, the latter $\boldsymbol{\kappa}_2$ requires $\mathbf{y}_{(2)}$, which takes 1 more FE as per (18b). In total, the MoC-pRK4 requires 4 FEs per variable per step, and hence its execution time should be approximately half that of the MoC-iRK4.

Let us also note that application of the MoC-iRK4 to systems of the form (69) required interpolation of solution to nodes off the grid (as in Fig. 1), whereas the MoC-pRK (70) does not require such an interpolation.

References

[1] T.I. Lakoba, Z. Deng, Stability analysis of the numerical Method of characteristics applied to a class of energy-preserving hyperbolic systems. Part I: Periodic boundary conditions, J. Comput. Appl. Math. 356 (2019) pp. 67–80.

[2] W.E.P. Padden, C.M. de Sterke, D.C. Psaila, Nonlinear pulse propagation in twin-core-fiber rocking filters, Phys. Rev. E 52 (1995) pp. 4401–4409.

[3] H.H.B. Rocha, J.C. Sales, W.B. de Fraga, A. da Conceição Ferreira, J.L.S. Lima, C.S. Sobrinho, J.W.M. Menezes, A.S.B. Sombra, Signal coupling in nonlinear hybrid optical structures: a numerical approach, Int'l Microwave Optoelectron. Conf. (IMOC 2009) pp. 611–614.

[4] I.M. Merhasin, B.A. Malomed, Four-wave solitons in Bragg cross-gratings, J. Opt. B: Quantum Semiclass. Opt. 6 (2004) pp. S323–S332.

[5] H. Ghafouri-Shiraz, Distributed feedback laser diodes and optical tunable filters, Wiley, Chichester, 2003; Secs. 2.4 and 2.3.4.

[6] H.G. Winful, Pulse compression in optical fiber filters, Appl. Phys. Lett. 46 (1985) pp. 527–529.

[7] C.J. McKinstrie, H. Kogelnik, G.G. Luther, L. Schenato, Stokes-space derivations of generalized Schrödinger equations for wave propagation in various fibers, Opt. Express 15 (2007) pp. 10964–10983.

[8] S. Pitois, G. Millot, S. Wabnitz, Nonlinear polarization dynamics of counterpropagating waves in an isotropic optical fiber: theory and experiments, J. Opt. Soc. B 18 (2001) pp. 432–443.

[9] V.V. Kozlov, J. Nuno, S. Wabnitz, Theory of lossless polarization attraction in telecommunication fibers, J. Opt. Soc. B 28 (2011) pp. 100–108.

[10] A.V. Mikhailov, S. Wabnitz, Polarization dynamics of counterpropagating beams in optical fibers, Opt. Lett. 15 (1990) pp. 1055–1057.

[11] S. Wabnitz, Chiral polarization solitons in elliptically birefringent spun optical fibers, Opt. Lett. 34 (2009) pp. 908–910.

[12] S. Wabnitz, Cross-polarization modulation domain wall solitons for WDM signals in birefringent optical fibers, IEEE Photon. Technol. Lett. 21 (2009) pp. 875–877.

[13] V.E. Zakharov, A.V. Mikhailov, Polarization domains in nonlinear media, JETP Lett. 45 (1987) pp. 349–35

[14] T. Kauffmann, I. Kocar, J. Mahseredjian, New investigations on the method of characteristics for the evaluation of line transients, Electr. Pow. Syst. Res. 160 (2018) pp. 243–250.

[15] T.J. Bridges, S. Reich, Multi-symplectic integrators: numerical schemes for Hamiltonian PDEs that conserve symplecticity, Phys. Lett. A 284 (2001) pp. 184–193.

[16] X. Zhao, On error estimates of an exponential wave integrator sine pseudospectral method for the Klein–Gordon–Zakharov system, Numer. Methods Partial Differ. Equ. 32 (2016) pp. 266–291.

[17] W. Yi, X. Ruan, C. Su, Optimal resolution methods for the Klein–Gordon–Dirac system in the nonrelativistic limit regime, J. Sci. Comput. 79 (2019) pp. 1907–1935.

[18] B. Ji, L. Zhang, X. Zhou, Conservative compact finite difference scheme for the N-coupled nonlinear Klein–Gordon equations, Numer. Methods Partial Differ. Equ. 35 (2019) pp. 1056–1079.

[19] M. El-Amrani, M. Seaïd, A finite element modified Method of Characteristics for convective heat transport, Numer. Methods Partial Differ. Equ. 24 (2008) pp. 776–798.

[20] J.-M. Qiu, C.-W. Shu, Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow, J. Comput. Phys. 230 (2011) pp. 863–889.

[21] S. Bak, High-order characteristic-tracking strategy for simulation of a nonlinear advection–diffusion equation, Numer. Methods Partial Differ. Equ. 35 (2019) pp. 1756–1776.

[22] T. Colonius, Numerically nonreflecting boundary and interface conditions for compressible flow and aeroacoustic computations, AIAA J. 35 (1997) pp. 1126–1133.

[23] H. Wang, M. Al-Lawatia, A.S. Telyakovskiy, Runge–Kutta characteristic methods for first-order linear hyperbolic equations, Numer. Methods Partial Differ. Equ. 13 (1997) pp. 617–661.

[24] M. Alhawwary, Z.J. Wang, Fourier analysis and evaluation of DG, FD and compact difference methods for conservation laws, J. Comput. Phys. 373 (2018) pp. 835–862.

[25] P.L. Roe, M. Arora, Characteristic-based schemes for dispersive waves I. The Method of Characteristics for smooth solutions, Numer. Methods Partial Differ. Equ. 9 (1993) pp. 459–505.

[26] C.M. de Sterke, K.R. Jackson, B.D. Robert, Nonlinear coupled-mode equations on a finite interval: a numerical procedure, J. Opt. Soc. Am. B 8 (1991) pp. 403–412.

[27] J. Chi, A. Fernandez, L. Chao, Comprehensive modeling of wave propagation in photonic devices, IET Commun. 6 (2012) pp. 473–477.

[28] T.I. Lakoba, Z. Deng, Stability analysis of the numerical Method of characteristics applied to a class of energy-preserving hyperbolic systems. Part II: Nonreflecting boundary conditions, J. Comput. Appl. Math. 356 (2019) pp. 267–292.

[29] G.D. Byrne, R.J. Lambert, Pseudo-Runge–Kutta methods involving two points, J. Assoc. Comput. Mach. 13 (1966) pp. 114–123.

[30] https://github.com/jsjewell/MoC-pRK-codes .

[31] J.E. Sipe, C.M. de Sterke, B.J. Eggleton, Rigorous derivation of coupled mode equations for short, high-intensity grating-coupled, co-propagating pulses, J. Mod. Opt. 49 (2002) pp. 1437–1452.

[32] E. Assemat, A. Picozzi, H.-R. Jauslin, D. Sugny, Hamiltonian tools for the analysis of optical polarization control, J. Opt. Soc. B 29 (2012) pp. 559–571.

[33] C.-W. Shu, S. Osher, Efficient implementation of Essentially Non-oscillatory shock-capturing schemes, J. Comp. Phys. 77 (1988) pp. 439–471.

[34] J.S. Jewell, Higher-order Runge–Kutta type schemes based on the Method of Characteristics for hyperbolic equations with crossing characteristics, M.S. Thesis, University of Vermont, 2019, https://scholarworks.uvm.edu/graddis/1028/ .

[35] J.C. Butcher, Numerical methods for ordinary differential equations, 2nd Ed., Wiley, Chichester, 2008; p. 180.

[36] M. Nakashima, On pseudo-Runge–Kutta methods with 2 and 3 stages, Publ. RIMS, Kyoto Univ. 18 (1982) pp. 895–909.

[37] H. Shintani, On pseudo-Runge–Kutta methods of the third kind, Hiroshima Math. J. 11 (1981) pp. 247–254.

[38] T.H. Lim, A third-order Nakashima pseudo-Runge–Kutta method, Sunway Acad. J. 10 (2014) pp. 36–45.

[39] D.J. Gross, A. Neveu, Dynamical symmetry breaking in asymptotically free field theories, Phys. Rev. D 10 (1974) pp. 3235–3253.

[40] S.-C. Li, X.-G. Li, F.-Y. Shi, Time-splitting methods with charge conservation for the nonlinear Dirac equation, Numer. Methods Partial Differ. Equ. 33 (2017) pp. 1582–1602.

[41] T.I. Lakoba, Study of instability of the Fourier split-step method for the massive Gross–Neveu model, J. Comput. Phys. 402 (2020) p. 109100.

[42] S.Y. Lee, T.K. Kuo, A. Gavrielides, Exact localized solutions of two-dimensional field theories of massive fermions with Fermi interactions, Phys. Rev. D 12 (1975) pp. 2249–2253.

[43] A. Alvarez, B. Carreras, Interaction dynamics for the solitary waves of a nonlinear Dirac model, Phys. Lett. A 86 (1981) pp. 327–332.

[44] N.G.R. Broderick, C.M. de Sterke, K.R. Jackson, Coupled mode equations with free carrier effects: a numerical solution, Opt. Quantum Electron 26 (1994) pp. S219–S234.

[45] R.W. Boyd, Nonlinear optics, Academic, San Diego, 1992; Secs. 8.3, 8.6, 10.6.

Appendix A: Derivation of stable P matrices and their physical context

A.1 Case $N^+ = N^- = 1$

The obvious necessary condition for system (5a) to be stable for $k = O(1)$ is that it be stable for $k = 0$, which amounts to $\mathbf{P}$ having only imaginary eigenvalues. This yields the allowed form of $\mathbf{P}$:

$$\mathbf{P} = ia_1\boldsymbol{\sigma}_1 + ia_2\boldsymbol{\sigma}_2, \tag{71a}$$

where $a_{1,2} \in \mathbb{R}$. Terms $ia_0\boldsymbol{\sigma}_0$ and $ia_3\boldsymbol{\sigma}_3$ with $a_{0,3} \in \mathbb{R}$ are absent in (6) since they can be removed by a phase transformation: $\mathbf{y}^{\pm} \to \mathbf{y}^{\pm} \exp[-i(a_0 t \pm a_3 x)]$. It is straightforward to verify that for (71a), the plane-wave solution, proportional to $\exp[i(kx - \omega t)]$, is stable for all $k \in \mathbb{R}$.

Now, (71a) can be transformed into a form where only one of $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ is present. For example, if originally $a_2 \neq 0$, then a similarity transformation with matrix $r\boldsymbol{\sigma}_0 + \boldsymbol{\sigma}_3$, where $r = i\left(\sqrt{(a_1/a_2)^2 + 1} - (a_1/a_2)\right)$, sets $a_2 = 0$. Note that this transformation does not affect matrix $\boldsymbol{\Sigma} \equiv \boldsymbol{\sigma}_3$ in (5a). Therefore, for $N^+ = N^- = 1$, one can take, without loss of generality:

$$\mathbf{P} = i\boldsymbol{\sigma}_1\,. \tag{71b}$$

In the context of coupled modes in a waveguide, the coefficients a_1 and a_2 in (71a) describe coupling via spatial modulation of, respectively, refractive index and gain/loss in both counter-propagating [5, 6] and co-propagating [2, 31] geometries. Matrix (71b) also arises in the one-dimensional relativistic field theory; examples are the Gross–Neveu and Thirring models.

A.2 Case $N^+ = N^- = 2$

Here we will consider two groups of models describing propagation of electromagnetic field in optical fibers. The field vector in a fiber has two components, referred to as polarizations. Typically, they propagate with slightly different speeds; this phenomenon is known as birefringence.

The two polarizations can be coupled linearly by spatial modulation of the fiber's refractive index. In addition, there can be linear coupling to the field in another, closely placed fiber. Finally, there can be nonlinear coupling, via the refractive index's nonlinear part, to a co- or counter-propagating field in the same fiber, as well as between two polarizations of the same field.

The first group contains just one model, where two polarizations in one fiber are coupled linearly to each other as well as to respective polarizations in a closely placed fiber [2]. The corresponding matrix has the form:

$$\mathbf{P} = i \begin{pmatrix} \boldsymbol{\sigma}_1 & a\,\boldsymbol{\sigma}_0 \\ a\,\boldsymbol{\sigma}_0 & \boldsymbol{\sigma}_1 \end{pmatrix}, \qquad a \in [0, \infty)\,, \tag{72}$$

where parameter a accounts for the relative strength of the two types of coupling. In order to make the form of the diagonal blocks of $\mathbf{P}$ matrices of both groups the same, we cast (72) into an equivalent form (8). This can be achieved by a similarity transformation with matrix

$$\mathbf{S} = \begin{pmatrix} \mathbf{V}^+ & \mathbf{O} \\ \mathbf{O} & \mathbf{V}^- \end{pmatrix}, \tag{73}$$

where $\mathbf{V}^\pm$ are some invertible matrices. One can show that (73) is *the only* similarity transformation that does not affect matrix $\boldsymbol{\Sigma}$ in (5a).

The second group comprises several models, each describing nonlinear coupling between polarizations of counter- and co-propagating fields. For a summary, see [32] and also below. The same or closely related models had been also considered in [7], although equations derived there were not put in the form (74). The general form of these models is:

$$\mathbf{s}_t^\pm \pm \mathbf{s}_x^\pm = \mathbf{s}^\pm \times \mathbf{J_c}\,\mathbf{s}^\mp + \mathbf{s}^\pm \times \mathbf{J_s}\,\mathbf{s}^\pm\,, \tag{74}$$

where $\mathbf{s}^\pm \equiv \left(s_1^\pm, s_2^\pm, s_3^\pm\right)^T$ are the Stokes vectors of the two fields, 'T' stands for transposition and $\mathbf{J_{c,s}}$ are matrices accounting, respectively, for cross- and self-interaction of $\mathbf{s}^\pm$. The real-valued components of the Stokes vector are defined in terms of the complex-valued field vector $\vec{E} = (E_1,\, E_2)^T$ as:

$$s_j = \left(\vec{E}^*\right)^T \boldsymbol{\sigma}_j\,\vec{E}, \qquad j = 1, 2, 3; \tag{75}$$

where $E_{1,2}$ are the polarizations and '$*$' stands for complex conjugation. In the context of counter-propagating geometry, superscripts '$+$' and '$-$' refer to forward- and backward-propagating fields. In the context of co-propagating geometry, the same identification can be made formally, as described after Eq. (1c). The form of matrices $\mathbf{J_{c,s}}$ has been derived for five

different physical models. Below we use the order of entries of these matrices consistent with definition (75).

The first model describes two counter-propagating fields in an isotropic fiber, in which case [8]:

$$\mathbf{J_c} = \mathrm{diag}(-2, -2, 0), \qquad \mathbf{J_s} = \mathrm{diag}(-1, -1, 0). \tag{76a}$$

The next two models describe counter- and co-propagating fields in a randomly birefringent fiber, where, respectively [9]:[6]

$$\mathbf{J_c} = \mathrm{diag}(1, -1, -1), \qquad \mathbf{J_s} = \mathbf{O}, \tag{76b}$$

and

$$\mathbf{J_c} = \mathrm{diag}(1, 1, 1), \qquad \mathbf{J_s} = \mathbf{O}\,. \tag{76c}$$

The last two models describe interaction of counter- and co-propagating fields in a spun, highly birefringent fiber, where the respective matrices are [10, 11]:

$$\mathbf{J_c} = (1 - a_{\mathrm{fiber}})\,\mathrm{diag}(1, -1, -2), \quad \mathbf{J_s} = \frac{3}{2}\,a_{\mathrm{fiber}}\,\mathrm{diag}(0, 0, 1), \qquad a_{\mathrm{fiber}} \in [-1/2,\, 1]; \tag{76d}$$

and [12]:

$$\mathbf{J_c} = (1 - a_{\mathrm{fiber}})\,\mathrm{diag}(1, -1, -2) - 3\,\mathrm{diag}(1, 1, 0), \quad \mathbf{J_s} = \frac{3}{2}\,a_{\mathrm{fiber}}\,\mathrm{diag}(0, 0, 1), \qquad a_{\mathrm{fiber}} \in [-1/2,\, 1];\ . \tag{76e}$$

Parameter $a_{\mathrm{fiber}} = 1 - (3/2)\cos^2\varphi_{\mathrm{fiber}}$ in these equations characterizes the relative strength of the spinning and birefringence, with φ_{fiber} being the degree of ellipticity of the polarization eigenmodes of the fiber ($\varphi_{\mathrm{fiber}} = 0$ and $\pi/2$ correspond to linear and circular eigenmodes, respectively). Note that a_{fiber} in (76d), (76e) is not the same as a in (9)–(11), although the two are related. Model (76d) with $a_{\mathrm{fiber}} = 0$ was earlier derived in [13] in a different physical context.

Two remarks about Eqs. (76) are in order. First, the numeric values of entries of $\mathbf{J_{c,\,s}}$ are dictated by fundamental physical properties of the nonlinear refractive index of the fiber and *cannot* be taken arbitrarily, as, say, $\mathrm{diag}(a_1, a_2, a_3)$. Second, these entries cannot be scaled relative to one another; e.g., $\mathbf{J_c} = \mathrm{diag}(1, -1, -1)$ cannot be transformed to $\mathbf{J_c} = \mathrm{diag}(1, -1, -2)$ by any linear transformation of the Stokes vectors $\mathbf{s}^{\pm}$.

[6] Expressions (76b), (76c) correct a typo that resulted in an incorrect order of entries in [9].

In deriving stable $\mathbf{P}$ matrices from models (76), we follow the approach of [1]. Namely, we consider six stationary solutions of (76):

$$s_j^+ = 1, \quad s_j^- = \pm s_j^+ \qquad \begin{array}{l}\text{for one of } j = 1, 2, \text{ or } 3, \text{ with} \\ \text{the other two components of } \mathbf{s}^\pm \text{ being } 0.\end{array} \tag{77}$$

For brevity, we will refer to these solutions as $(j\pm)$, where j and $\pm$ correspond to the particular choice of the component and the sign in (77). For each model in (76) we obtain $\mathbf{P}$ from linearization about each of these solutions; this results in 30 matrices. The explicit form of $\mathbf{P}$ allows us to determine which of solutions (77) are physically stable for a given model in (76); this is done by numerically finding the dispersion relation $\omega(k)$ for the plane wave solution $\propto \exp[i\omega t - ikx]$. For model (76a), the only physically stable solution is (3+), and its $\mathbf{P}$ corresponds to $\mathbf{P}_3$ with $a = -1/2$. For model (76b), the stable solutions are (1+), (2−), (3−), and the corresponding matrices can be reduced to $\mathbf{P}_3$ in (9) with $a = 0$ by a similarity transformation (73). The stable solutions (1+), (2+), (3+) of model (76c) result in the same matrix. The stable solutions (1+), (2−) of model (76d) result (either directly or via a similarity transformation) in $\mathbf{P}_4$ in (10), and the stable solution (3−) results in $\mathbf{P}_3$ with $a \in [-1/2,\, 7/4]$. The stable solution (1+) and (2+) of model (76e) result in $\mathbf{P}_5$ in (11), while the stable solution (3+) results in $\mathbf{P}_3$ with $a \in [-3/2,\, 3/2]$. All solutions not mentioned above are physically unstable, and hence their $\mathbf{P}$ matrices are not considered in this work. (This includes matrices $\mathbf{P}_4$ and $\mathbf{P}_5$ with values of parameter a that correspond to values of a_{fiber} allowed by (76d) and (76e) but that are outside of the intervals for a listed in (10) and (11).)

Appendix B: Pseudocodes and codes for MoC-pRK{3,4} schemes with nonreflecting BC

To improve visual clarity, in these pseudocodes we will *not* use boldface font for variables.

Recall that the nodes of the grid are numbered with $1 \le m \le M$. Then the virtual nodes nearest to the grid on the left and right are numbered as $m = 0$ and $m = M + 1$, respectively, and the next-to-nearest nodes are numbered as $m = -1$ and $m = M + 2$. Notations like $y_{[1\,:\,M]}$ will refer to *all* nodes with $1 \le m \le M$, while those like $y_{[1,\,M]}$ will refer to *only two* nodes $m = 1$ and $m = M$. At each time level except the first, the algorithms first handle virtual nodes outside the grid and then those inside the grid; the corresponding groups of steps are labeled as OG and IG. This order is dictated by the fact that the OG calculations are performed at *earlier* time levels.

Equation numbers for the computation of a specific variable (e.g., (14) for $\boldsymbol{\kappa}_1$) are listed only once per pseudocode. Also, some of these listed equations pertain only to the left boundary; their counterparts for the right boundary are obtained straightforwardly.

Schematics of the pseudocodes are illustrated in Fig. 16. Only the left boundary and the flow of the computation of the components along the *positive* characteristic are shown; for the right boundary and the negative characteristic they are analogous. The emphasis of this schematic is on the OG calculations, and hence only the $m = 2, 3$ nodes for the IG calculation are shown.

We note that these pseudocodes use a different organization of OG calculations than the pseudocodes presented in [34].

Finally, the actual Matlab codes for both schemes are found on GitHub [30]. The main groups of logical steps that these codes have while the pseudocodes below do not are: (i) reassignment of κ's computed at previous time levels (e.g., as n is increased, one reassigns $\kappa^{n-1} \to \kappa^{n-2}$, etc.) and (ii) storage of near-boundary values.

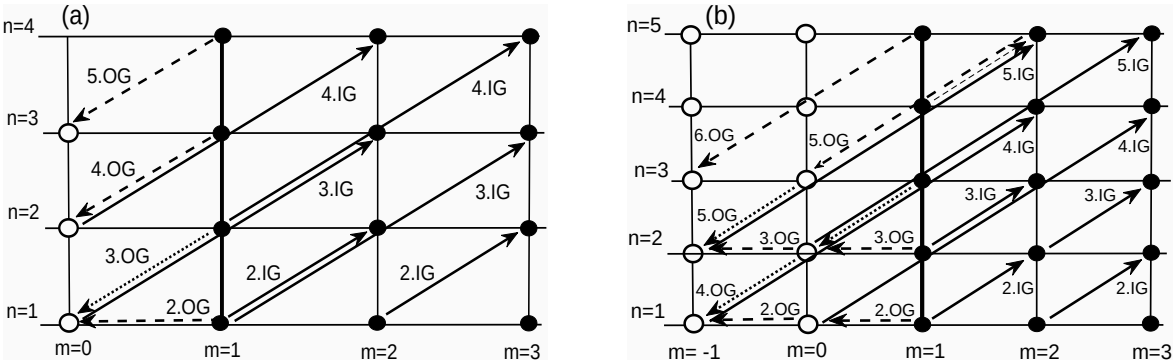

Figure 16: Schematics illustrating the pseudocodes. Filled (open) circles represent real (virtual) nodes. Solid arrows show the flow of an inside-the-grid calculation along the positive characteristic. Dashed (dotted) arrows indicate an outside-the-grid calculation (retrieval of stored data) to compute (retrieve) the virtual node to which the arrow is pointing. The numbers in front of 'OG' and 'IG' and next to the arrows correspond to the time levels as numbered in the pseudocodes. Panels (a) and (b) correspond to the MoC-pRK3 and MoC-pRK4 schemes, respectively. In panel (b), two labels are missing due to lack of space: 4.OG for the arrow from node $(n, m) = (3, 1)$ to $(2, 0)$ and 5.OG from $(4, 1)$ to $(5, 2)$. Note that while the latter arrow (thin dashed) is located inside the grid, it belongs to the 'OG' group.

MoC-pRK3 pseudocode

---

while $n <$ nmax

  if $n = 1$

    IG: $(y^{\pm})^n_{[1::M]}$ given by initial condition

IG: Compute $(\kappa_1^+)_{[1::M-1]}^n$ and $(\kappa_1^-)_{[2::M]}^n$ by (14)

else if $n = 2$

OG: Compute $(y^\pm)_{[0,\,M+1]}^{n-1}$ by (57)

IG: Compute $(y^\pm)_{[1::M]}^n$ by MoC-ME (18b) and BC (23b)

IG: Compute $(\kappa_2^+)_{[1::M-1]}^{n-1}$ and $(\kappa_2^-)_{[2::M]}^{n-1}$ by (44b)

IG: Compute $(\kappa_1^+)_{[1::M-1]}^n$ and $(\kappa_1^-)_{[2::M]}^n$

else $(n \geq 3)$

OG: **if** $n = 3$

Retrieve $(y^\pm)_{[0,\,M+1]}^{n-2}$ from $n = 1$

else

Compute $(y^\pm)_{[0,\,M+1]}^{n-2}$ by the rotated MoC-ME (58)

end

OG: Compute $(\kappa_{1,2}^+)_0^{n-2}$ and $(\kappa_{1,2}^-)_{M+1}^{n-2}$

IG: Compute $(y_{(2)}^\pm)_{[1::M]}^n$ by MoC-ME

IG: Compute $(\kappa_2^+)_{[1::M-1]}^{n-1}$ and $(\kappa_2^-)_{[2::M]}^{n-1}$

IG: Compute $(y^\pm)_{[1::M]}^n$ by MoC-pRK3 (44a)

IG: Compute $(\kappa_1^\pm)_{[1::M-1]}^n$ and $(\kappa_1^-)_{[2::M]}^n$

end

end while

MoC-pRK4 pseudocode

---

while $n <$ nmax

if $n = 1$

IG: $(y^\pm)_{[1::M]}^n$ given by initial condition

IG: Compute $(\kappa_1^+)_{[1::M-1]}^n$ and $(\kappa_1^-)_{[2::M]}^n$ by (14)

IG: Compute $(y_{(1)}^+)_{[1::M-1]}^{n+1}$ and $(y_{(1)}^-)_{[2::M]}^{n+1}$ by (18a)

(% These will be used at the *next* n to compute κ_2 at *this* n.)

else if $n = 2$ or $n = 3$

OG: Compute $(y^\pm)_{[-1,\,0,\,M+1,\,M+2]}^{n-1}$ by (59)

OG: **if** $n = 3$

Compute $(\kappa_1^{(+)})_0^{n-2}$ and $(\kappa_1^{(-)})_{M+1}^{n-2}$

Compute $(\kappa_2^{(+)})_0^{n-2}$ and $(\kappa_2^{(-)})_{M+1}^{n-2}$

end

IG: Compute $(y^\pm)_{[1::M]}^n$ by MoC-RK3 (17) and BC (23b)

IG: Compute $(\kappa_2^+)_{[1::M-1]}^{n-1}$ and $(\kappa_2^-)_{[2::M]}^{n-1}$ by (46b)

IG: Compute $(\kappa_1^+)_{[1::M-1]}^n$ and $(\kappa_1^-)_{[2::M]}^n$

IG: Compute $(y_{(1)}^+)_{[1::M-1]}^{n+1}$ and $(y_{(1)}^-)_{[2::M]}^{n+1}$

end

else $(n \geq 4)$

OG: **if** $n > 4$

Compute $(y^{\pm}_{(2)})^{n}_{[2,\,M-1]}$ by the MoC-ME

end

OG: **if** $n = 4$

Retrieve $(y^{\pm})^{n-2}_{[0,\,M+1]}$

else $(n \geq 5)$

Compute $(y^{\pm})^{n-2}_{[0,\,M+1]}$ by the rotated MoC-pRK3 (60)

end

OG: Compute $(\kappa_1^{(+)})^{n-2}_{0}$ and $(\kappa_1^{(-)})^{n-2}_{M+1}$

OG: Compute $(\kappa_2^{(+)})^{n-2}_{0}$ and $(\kappa_2^{(-)})^{n-2}_{M+1}$

OG: **if** $n = 4$ or $n = 5$

Retrieve $(y^{\pm})^{n-3}_{[-1,\,M+2]}$

else $(n \geq 6)$

Compute $(y^{\pm})^{n-2}_{[-1,\,M+2]}$ by the rotated MoC-pRK3

end

OG: Compute $(\kappa_1^{(+)})^{n-3}_{-1}$ and $(\kappa_1^{(-)})^{n-3}_{M+2}$

OG: Compute $(\kappa_2^{(+)})^{n-3}_{-1}$ and $(\kappa_2^{(-)})^{n-3}_{M+2}$

IG: Compute $(y^{\pm}_{(3)})^{n}_{[-1\,::\,M]}$ by the MoC-pRK3 (44a)

IG: Compute $(\kappa_2^{+})^{n-1}_{[1::\,M-1]}$ and $(\kappa_2^{-})^{n-1}_{[2::\,M]}$

IG: Compute $(y^{\pm})^{n}_{[1::\,M]}$ by MoC-pRK4 (46a) and BC

IG: Compute $(\kappa_1^{+})^{n}_{[1::\,M-1]}$ and $(\kappa_1^{-})^{n}_{[2::\,M]}$

IG: Compute $(y^{+}_{(1)})^{n+1}_{[1::\,M-1]}$ and $(y^{-}_{(1)})^{n+1}_{[2::\,M]}$

end

end while

Appendix C: Technical considerations for computing numerical error in Section 8

If computed by the naive formula

$$\epsilon^{\pm}_{\text{naive}} \,=\, \max_{x} \left| u^{\pm}(x,t) - u^{\pm}_{\text{sol}}(x,t) \right|, \tag{78a}$$

where $u^{\pm}_{\text{sol}}(x,t)$ is given by (62), the error $\epsilon^{\pm}_{\text{naive}}$ would grow in time due to the following. The scheme's discretization error at every time step causes the soliton parameters Ω, V, ϕ_0, and x_0 to drift. (For the standing soliton (62), only Ω and ϕ_0 will drift due to symmetry considerations.) As the discretization error is approximately constant in time for the constant soliton profile, this drift can be assumed to have approximately constant rate. A drift of ϕ_0 at a rate $\dot{\phi}_0$ causes the error, computed by (78a), to grow linearly in time, since $|\exp[i\dot{\phi}_0 t] - 1| \propto t$ for $|\dot{\phi}_0 t| \ll 1$. Similarly, a drift of Ω at a constant rate $\dot{\Omega}$ causes the error to grow quadratically in time, since

now the soliton's phase is $\int_0^t \Omega(t')dt' = \Omega(0)t + \dot{\Omega}t^2/2$. Since, in general, both ϕ_0 and Ω drift, the time dependence of the error depends on the relation between $\dot{\phi}_0$ and $\dot{\Omega}t$. A typical result is shown in Fig. 17(a). Such a growth of the error due to the phase drift would mask an error occurring due to the changes in the soliton's profile and which, for practical purposes, could be deemed more essential than the phase error.

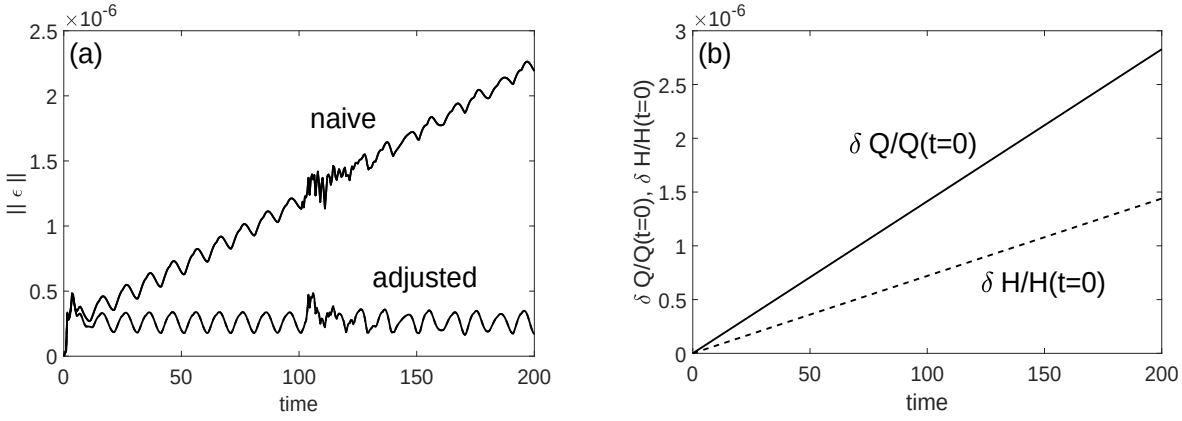

Figure 17: (a): Evolution of the solution error of MoC-NpRK4 with $c_{22} = 1$ and $h = 0.025$ with periodic BC. (b): Evolution of errors of Q and H for the same simulation.

Thus, to avoid this growth, we computed the error as

$$\epsilon^{\pm} \,=\, \max_x \, |u^{\pm}(x,t) - u^{\pm}_{\text{sol, adj}}(x,t)| \,, \qquad \|\epsilon\| = \sqrt{(\epsilon^+)^2 + (\epsilon^-)^2} \,, \tag{78b}$$

where in the last term of the first expression, parameters Ω and ϕ_0 of the soliton are being continuously adjusted. This adjustment proceeds as follows. First, at each time step, one computes the soliton's charge Q_{comp} from (64a) and the numerical solution $u^{\pm}(x,t)$. As illustrated in Fig. 17(b), Q_{comp} drifts in time due to the scheme not being exactly conservative. Given the first relation in (64c), one infers that

$$\Omega_{\text{comp}} = 1/\big(1 + (Q_{\text{comp}}/2)^2\big) \,. \tag{79a}$$

Second, one measures the phase $\phi_{\text{comp}}(t)$ of the computed solution as the phase of $(u^+(0,t) + u^-(0,t))$. (For the exact soliton (62), the phase of $(u^+(x,t) + u^-(x,t))$ would equal $(-\Omega t)$ uniformly in x.) Then in (78b), one sets

$$u^{\pm}_{\text{sol}}(x,t) \,=\, \sqrt{1-\Omega_{\text{comp}}}\, \frac{\cosh(\beta_{\text{comp}}\, x) \,\pm\, i\mu_{\text{comp}}\, \sinh(\beta_{\text{comp}}\, x)}{\cosh^2(\beta_{\text{comp}}\, x) - \mu^2_{\text{comp}}\, \sinh^2(\beta_{\text{comp}}\, x)}\, \exp[i\phi_{\text{comp}}(t)] \,, \tag{79b}$$

where β_{comp} and μ_{comp} are related to Ω_{comp} by (62b).

In order to use (79a) to compute the error of the soliton entering the medium, as in Section 8.2, one would have to adjust not only Ω and ϕ_0, but also V and x_0. We do not do so for two

reasons. First, it is considerably more complicated than adjusting just Ω and ϕ_0 of the standing soliton; this, in particular, is due to the fact that as the soliton is entering the medium, small changes in Ω and ϕ_0 become coupled to those of V and x_0. Moreover, there is actually no need to carry out a long-term simulation of a soliton entering the medium to evaluate performance of a numerical scheme. Indeed, simulating the standing soliton over a long time was needed *not* to confirm its accuracy, but rather to verify that it can preserve conserved quantities to a given degree in long-term simulations. In the pulse-entering problem, one is not concerned with preservation of the conserved quantities, as they, naturally, vary as the pulse enters the medium. Therefore, in this case, we computed the error only up to $t = 5$ using the "naive" formula (78a) and did not compute the error in Q and H.