Chebyshev pseudospectral method for wave equation with absorbing boundary conditions that does not use a first order hyperbolic system

F. S. V. Bazán
Department of Mathematics,
Federal University of Santa Catarina,
88040-900 - Florianópolis SC, Brasil
e-mail: fermin@mtm.ufsc.br

Abstract

The analysis and solution of wave equations with absorbing boundary conditions by using a related first order hyperbolic system has become increasingly popular in recent years. At variance with several methods which rely on this transformation, we propose an alternative method in which such hyperbolic system is not used. The method consists of approximation of spatial derivatives by the Chebyshev pseudospectral collocation method coupled with integration in time by the Runge-Kutta method. Stability limits on the timestep for arbitrary speed are calculated and verified numerically. Furthermore, theoretical properties of two methods by Jackiewicz and Renaut are derived, including, in particular, a result that corrects some conclusions of these authors. Numerical results that verify the theory and illustrate the effectiveness of the proposed approach are reported.

Key words. Wave equation, Chebyshev pseudospectral methods, eigenvalue stability, pseudo-eigenvalues.

1 Introduction

We consider some numerical methods for the 1D wave equation

\[
\begin{align*}
  u_{tt} &= c^2 u_{xx}, & 0 < x < 1, \ t > 0, \\
  u(x,0) &= f(x), & 0 < x < 1, \\
  u_t(x,0) &= 0, & 0 < x < 1
\end{align*}
\]  

(1)

\[
\begin{align*}
  u_t(0, t) - cu_x(0, t) &= 0, & t > 0 \\
  u_t(1, t) + cu_x(1, t) &= 0, & t > 0
\end{align*}
\]  

(2)

Boundary conditions like these are introduced to avoid or absorb spurious reflections on the boundary, see, e.g., [3, 5, 9, 11, 12, 15, 19].
The pseudospectral collocation approach [1, 6, 8, 10, 11, 13, 16, 18] has become an efficient way to generate several methods for solving time dependent partial differential equations (PDEs) due to its high precision and relatively lower computational cost compared with finite difference methods. These methods start by approximating spatial derivatives by the Chebyshev pseudospectral collocation (CPS) method, giving rise to a set of ordinary differential equations (ODEs) in which only the first time derivative appears. This set, referred to as semidiscrete system, is then integrated in time using a great variety of schemes.

As for the initial and boundary problem (1)-(2), a subclass of CPS methods that has been increasingly exploited in recent years is the one in which the analysis and solution of the problem is done by using a related first order hyperbolic system of PDEs [4, 11, 10, 15]. Although these methods allow certain degree of flexibility for improvement of the accuracy because integration in time can be done by using an appropriate method for ODEs, some care is required when using them as the overall accuracy of computed solutions depends strongly on the chosen method and as the solution \( u \) is calculated from approximations to \( u_t \). For this reason, methods that do not use such hyperbolic system of PDEs and that allow the calculation of \( u \) without using any estimate of \( u_t \) seem desirable. It is the object of the present paper to propose such a method.

Essentially, we transform the PDE into a first order system of ODEs and then integrate in time by using the fourth-order Runge-Kutta (RK) method. Although the order of accuracy in time can be considered low when compared with the one of spatial derivatives, this is not a serious difficulty since highly accurate solutions can be constructed by taking the timestep fairly small while maintaining the storage requirements within reasonable bounds.

The paper is structured as follows. In Section 2 we reintroduce some methods of Jackiewicz and Renaut [10, 11] and then describe the proposed method. Theoretical properties of all methods are described here, including, in particular, a result that corrects some conclusions in [11]. Stability of the proposed method as well as the stability limits on the timestep for arbitrary speed are included in Section 3. In Section 4 we give a numerical example that illustrates the effectiveness of the proposed approach. The paper ends with some concluding remarks in Section 5.

2 Methods and properties

As commented before, the methods we are interested in, use approximations to first order spatial derivatives calculated by the Chebyshev pseudospectral collocation method. From here on we use as collocation points the so-called Chebyshev-Gauss-Lobatto points, numbered from left to right on the interval [0,1], defined by

\[
0 = x_0 < x_1 < \cdots < x_n = 1, \quad \text{with} \quad x_j = \frac{1}{2}[1 - \cos(j\pi/n)], \quad j = 0, \ldots, n. \tag{3}
\]

The objective of this section is to review some methods by Jackiewicz and Renaut [10, 11], as well as to describe the proposed method along with underlying properties. To this end, we start by describing some definitions and notation.

Throughout the paper the first order \( (n + 1) \times (n + 1) \) Chebyshev differentiation matrix associated with these collocation points is denoted by \( D \). The columns (resp., rows) of \( D \) are
denoted by $d_j$ (resp., $l_j$). Let $D_1, D_2,$ and $D_3$ be matrices defined by

$$D_1 = [d_2, \ldots, d_n], \quad D_2 = [l_2, \ldots, l_n]^T, \quad E = [d_1 \; 0_{(n+1) \times (n-1)} - d_{n+1}],$$

(4)

where $0_{m \times n}$ denotes the $m \times n$ matrix having all entries equal to zero; when $m = n$ we simply write $0_n$. Accordingly, $I_{n+1}$ denotes the identity matrix of order $n + 1$.

### 2.1 First order hyperbolic system-based methods

These methods rely on a conversion of the second order partial differential equation to a first order hyperbolic system of PDEs. With the notation $u_1(x, t) = u_t(x, t)$, and $u_2(x, t) = u_x(x, t)$, problem (1)-(2) is converted to the first order hyperbolic system

$$\begin{align*}
\left\{ \begin{array}{l}
u_1, t = c^2 u_2, x, \quad 0 < x < 1, \quad t > 0, \\
u_2, t = u_1, x, \quad 0 < x < 1, \quad t > 0,
\end{array} \right.
\end{align*}$$

(5)

subject to the initial and boundary conditions

$$\begin{align*}
u_1(x, 0) &= 0, \quad 0 \leq x \leq 1, \\
u_2(x, 0) &= f_x(x), \quad 0 \leq x \leq 1, \\
u_1(0, t) - cu_2(0, t) &= 0, \quad t > 0 \\
u_1(1, t) + cu_2(1, t) &= 0, \quad t > 0
\end{align*}$$

(6)

Approximation of spatial first order derivatives via the differentiation matrix $D$ leads to several first order semidiscrete systems of ODEs which depend on the way the boundary conditions are incorporated into the model. To simplify the notation, here and throughout we use $u_1(x_j, t), \quad 0 \leq j \leq n,$ to denote both the exact value and the corresponding approximation arising from solving the related semidiscrete system; the same comment applies to $u_2(x_j, t)$ and $u(x_j, t)$. We shall concentrate on two methods by Jackiewicz and Renaut [11, 10] which are briefly described as follows.

If the boundary conditions are incorporated by eliminating the variables $u_2(0, t)$ and $u_2(1, t)$ taking $u_2(0, t) = u_1(0, t)/c$ and $u_2(1, t) = -u_1(1, t)/c$, the semidiscrete system is

$$\begin{align*}
\left\{ \begin{array}{l}
U_{1,t} = A_1 U_1, \quad t > 0, \\
U_1(0) = [0_{1 \times n+1} \; f_x(x_1), \ldots, f_x(x_{n-1})]^T
\end{array} \right.
\end{align*}$$

(7)

where

$$A_1 = \begin{bmatrix}
c d_1 0_{(n+1) \times (n-1)} - cd_{n+1} \\
D(2 : n, 1 : n + 1) \\
D_{(n-1) \times (n-1)}
\end{bmatrix},$$

and

$$U_1 = [u_1(x_0, t), \ldots, u_1(x_n, t), u_2(x_1, t), \ldots, u_2(x_{n-1}, t)]^T.$$
As in [11], MATLAB notation is being used. If, on the other hand, we choose \( u_1(0, t) = cu_2(0, t) \) and \( u_1(1, t) = -cu_1(1, t) \), the resulting semidiscrete system is

\[
\begin{align*}
U_2(t) &= A_2 U_2, \ t > 0, \\
U_2(0) &= [0_{1 \times (n-1)} f_2(x_0), \ldots, f_2(x_n)]^T
\end{align*}
\]  

where

\[
A_2 = \left[ \begin{array}{c|c}
0_{(n-1) \times (n-1)} & c^2 D(2 : n, 1 : n + 1) \\
D(1 : n + 1, 2 : n) & cd_1 0_{(n+1) \times (n-1)} - cd_{n+1}
\end{array} \right],
\]

and

\[
U_2 = [u_1(x_1, t), \ldots, u_1(x_{n-1}, t), u_2(x_0, t), \ldots, u_2(x_n, t)]^T.
\]

We now observe that because time integration of these systems does not give approximations to the solution \( u \), but rather to \( u_1 \) and \( u_2 \), Jackiewicz and Renaut proposed the following two methods to calculate \( u \) from \( u_1 \)

**Method 1:**

1. Step 1: Calculate approximations to \( u_1 \) by the fourth-order RK method.
2. Step 2: For \( j = 0, \ldots, n \) and \( v \geq 0 \), calculate

\[
u(x_j, t_{v+3}) = u(x_j, t_v) + \Delta t [\beta_0 u_1(x_j, t_v) + \beta_1 u_1(x_j, t_{v+1}) + \beta_2 u_1(x_j, t_{v+2}) + \beta_3 u_1(x_j, t_{v+3})],
\]

where \( \beta_0 = -\frac{51}{8}, \beta_1 = \frac{171}{8}, \beta_2 = -\frac{153}{8} \) and \( \beta_3 = \frac{57}{8} \).

The second method proposes integration in time by using a fourth-order diagonally implicit multistage integration method (DIMSIM). In these methods, approximations to \( u \) at time \( t_{v+1} \) are calculated by using approximations to \( u_1 \) at interior points of the interval \([t_v, t_{v+1}]\) (called internal stages). The method is as follows.

**Method 2:**

For \( j = 0, \ldots, n \) and \( v \geq 0 \), calculate

\[
u(x_j, t_{v+1}) = u(x_j, t_v) + \Delta t [\gamma_0 u_1(x_j, t_v + c_1 \Delta t) + \gamma_1 u_1(x_j, t_v + c_2 \Delta t) + \gamma_2 u_1(x_j, t_v + c_3 \Delta t) + \gamma_3 u_1(x_j, t_v + c_4 \Delta t)]
\]

where \( \gamma_0 = \gamma_3 = \frac{13}{48}, \gamma_1 = \gamma_2 = \frac{11}{48}, c_1 = \frac{1}{8}, c_2 = \frac{3}{8}, c_3 = \frac{5}{8}, c_4 = \frac{7}{8} \).

The principal characteristic of Method 2 is that the internal stages \( u_1(x_j, t_v + c_i \Delta t), \ i = 1, \ldots, 4, \) as well as the estimates \( u(x, t) \) are all \( O(\Delta t^4) \) accurate; the reader is referred to [2, 10] for details. As a consequence, Method 2 should outperform Method 1; explanation about this can be found in [10].
A comment is required concerning a conclusion in a paper by Jackiewicz and Renaut [11]. In that paper the authors analyze numerically the spectra of matrices \(A_1\) and \(A_2\) and conclude, among other things, that the spectrum of \(A_1\) differs from the spectrum of \(A_2\) (see Section 2 in [11]). The following proposition shows this conclusion is incorrect \(^1\).

**Proposition 1** System matrices \(A_1\) and \(A_2\) are similar.

**Proof:** Define
\[
S = \begin{bmatrix} 0_{(n-1)\times(n+1)} & c^2 I_{n-1} \\
I_{n+1} & 0_{(n+1)\times(n-1)} \end{bmatrix}.
\]

It follows that \(S\) is non singular and that its inverse is
\[
S^{-1} = \begin{bmatrix} 0_{(n+1)\times(n-1)} & I_{n+1} \\
\frac{1}{c^2} I_{n-1} & 0_{(n-1)\times(n+1)} \end{bmatrix}.
\]

Using these matrices it is straightforward to see that
\[
A_1 = S^{-1} A_2 S,
\]
as claimed. \(\square\)

**Remark:** It is worth noting that the statement of the above proposition continues to hold irrespective of the grid points being used, i.e., we might use, for instance, Legendre points, and the assertion will be still valid. Additional consequences of this proposition will be commented in Section 4.

The eigenvalues of these matrices, which come come in complex conjugate pairs, are very sensitive to small perturbations such as rounding errors, except for some “outlier eigenvalues”. This is illustrated in Figure 1. Notice that for \(n = 31\) the eigenvalues of \(A_1\) match well those of \(A_2\); this is not the case when \(n = 63\). These results do not contradict Prop. 1, of course, they simply illustrate that matrices \(A_1\) and \(A_2\) are far from normal.

A useful measure of closeness to normality of \(A \in \mathbb{R}^{m \times m}\) is the Jordan condition number of the corresponding eigenproblem. If \(A\) has a Jordan decomposition \(A = VAV^{-1}\), the 2-norm Jordan condition number is
\[
\kappa(V) = \|V\|_2 \|V^{-1}\|_2.
\]
A a consequence, the smaller \(\kappa(V)\), the closer \(A\) is to normality.

As for the Jordan condition numbers of \(A_1\) and \(A_2\), for \(c = 1\), \(n = 31, 63\), we have
\[
\kappa(V_1) \approx 6.5194e + 11, \quad 8.5281e + 14, \quad \kappa(V_2) \approx 6.5191e + 11, \quad 9.4652e + 14
\]
which explains the behavior of eigenvalues displayed in Figure 1.

\(^1\)After contacting Z. Jackiewicz, we now know that the analysis in [11] relies on a “system matrix \(A_1\)” that differs from that in eq. (7) (eq. (2.6) in [11]) and arises from one of the so many ways of incorporating the boundary conditions into the model. This clarifies what is claimed in [11] on the spectra of matrices \(A_1\) and \(A_2\) and related issues.
An important consequence of Proposition 1 is that the Jordan condition numbers of the eigenproblems associated with system matrices $A_1$ and $A_2$ relate to each other by a constant factor that depends on the speed $c$. This is made precise in the following proposition.

**Proposition 2** Let $\kappa(V_1)$, $\kappa(V_2)$ be the Jordan condition numbers of the eigenvalue problems associated with $A_1$ and $A_2$, respectively. These numbers relate to each other by a constant factor not larger than $\Xi \geq 1$ defined by

$$\Xi = \begin{cases} \frac{c^2}{c} & \text{if } c > 1, \\ 1 & \text{if } c = 1, \\ \frac{1}{c^2} & \text{if } c < 1. \end{cases}$$

**Proof:** Let $A_1 = V_1 A V_1^{-1}$ be a Jordan decomposition of $A_1$. By Prop. 1, the 2-norm Jordan condition number $\kappa(V_2)$ satisfies

$$\kappa(V_2) = \|SV_1\|_2\|V_1^{-1}S^{-1}\|_2 \leq \|S\|_2\|S^{-1}\|_2\|V_1\|_2\|V^{-1}\|_2 = \kappa(S)\kappa(V_1). \quad (11)$$

Now observe that if $c > 1$, because the columns of matrix $S$ are orthogonal, we have $\|S\|_2 = c^2$, $\|S^{-1}\|_2 = 1$, and hence $\kappa(S) = c^2$. A similar reasoning leads to $\kappa(S) = 1/c^2$ if $c < 1$, and $\kappa(S) = 1$ if $c = 1$. Insertion of this value in (11) shows that $\kappa(V_2) \leq \Xi \kappa(V_1)$. If we reverse the roles of $V_1$ and $V_2$ in the above procedure we obtain $\kappa(V_1) \leq \Xi \kappa(V_2)$. What $\Xi \geq 1$ follows from the definition itself. This ends the proof. \hfill $\Box$

The conclusion that can be drawn from Prop. 1 and the above numerical results is that the eigenproblems associated with system matrices $A_1$ and $A_2$ are either almost equally conditioned when $c \neq 1$ or equally conditioned when $c = 1$, and that for any choice of the speed $c$, the corresponding condition numbers can be extremely huge when $n$ is sufficiently large. From the practical point of view, this means that the stability properties of time integration methods for the semidiscrete systems (7) and (8) should not differ significantly.
2.2 Block companion approach

To describe the proposed method we need some notation. Let \( \bar{u} = [u(x_0, t), \ldots, u(x_n, t)]^T \). Since the inner product of \( j \)th row of \( D \) and \( \bar{u} \) approximates \( u(x_{j-1}, t), \ 1 \leq j \leq n+1 \), the boundary conditions (2) imply that

\[
I_1^T \bar{u} \approx u(x_0, t) = u_t(x_0, t)/c, \quad I_{n+1}^T \bar{u} \approx u_x(1, t) = -u_t(1, t)/c.
\]

In view that

\[
\bar{u}_{tt} = c^2 \bar{u}_{xx} \approx c^2 D^2 \bar{u},
\]

using (12) we obtain

\[
D^2 \bar{u} = d_1 I_1^T \bar{u} + (d_2 I_2^T \bar{u} + \cdots + d_n I_n^T \bar{u}) + d_{n+1} I_{n+1}^T \bar{u}
\approx [d_1/c \ 0_{(n+1) \times (n-1)} - d_{n+1}/c] \bar{u}_t + D_1 D_2 \bar{u}.
\]

Neglecting the approximation error we obtain the set of second order system of ODEs

\[
\bar{u}_{tt} = c E \bar{u}_t + c^2 D_1 D_2 \bar{u}.
\]

If we introduce

\[
U_3 = [u_t(x_0, t), \ldots, u_t(x_n, t), u(x_0, t), \ldots, u(x_n, t)]^T,
\]

a first order system in companion block form associated with (13) is

\[
\begin{align*}
U_3 &= A_3 U_3, \\
U_3(0) &= [0_{1 \times (n+1)}, f(x_0), \ldots, f(x_n)]^T,
\end{align*}
\]

where

\[
A_3 = \begin{bmatrix}
    c E & c^2 D_1 D_2 \\
    I_{n+1} & 0_{n+1}
\end{bmatrix}
\]

(15)

with \( D_1, D_2 \) and \( E \) being defined in (4). Observe that \( A_3 \) is now of order \( 2n+2 \), as opposed to matrices \( A_1 \) and \( A_2 \) which are of order \( 2n \). In spite of this, an important characteristic of the semidiscrete system (14) is that the related system matrix contains essentially the same information as \( A_1 \) and \( A_2 \). This is the subject of the following proposition.

**Proposition 3** Except for a zero eigenvalue of multiplicity two, \( \lambda(A_3) \), the spectrum of matrix \( A_3 \), equals \( \lambda(A_1) \), the spectrum de \( A_1 \).

**Proof:** We first recall a well known result from numerical linear algebra concerning eigenvalues. Let \( A \in \mathbb{R}^{m \times q} \) and \( B \in \mathbb{R}^{q \times m}, \ m \geq q \). Then except for \( (m-q) \) zero eigenvalues, we have that \( \lambda(AB) = \lambda(BA) \).

Now observe that \( A_3 \) can be rewritten as

\[
A_3 = \begin{bmatrix}
    c E & c^2 D_1 D_2 \\
    I_{n+1} & 0_{n+1}
\end{bmatrix}
\]

(15)
Application of the above property to the right hand side of the previous equality gives that, except for two zero eigenvalues, we have

$$\lambda(A_3) = \lambda\left( \begin{bmatrix} I_{n+1} & cE \\ D_2 & I_{n+1} \end{bmatrix} \begin{bmatrix} c^2 D_1 \\ 0 \end{bmatrix}_{(n+1) \times (n-1)} \right) = \lambda(A_1),$$

which proves the assertion of the proposition. \[\square\]

Another point to be discussed here is the dependence of the spectral radius of the system matrices on constant $c$. To highlight the dependence of the system matrix $A_3$ on $n$ and $c$, we shall use the notation $A_3(n, c)$ instead of $A_3$. Our goal is to relate the spectrum of $A_3(n, c)$ for arbitrary $c > 0$ to the spectrum of $A_3(n, 1)$.

**Proposition 4** Let $\rho(A_3(n, c))$ denote the spectral radius of the system matrix $A_3(n, c)$ defined in (15). Then $\lambda(A_3(n, c)) = c \lambda(A_3(n, 1))$, and thus

$$\rho(A_3(n, c)) = c \rho(A_3(n, 1)),$$

and the same property continues to hold if $A_3$ is replaced with $A_1$ or $A_2$.

**Proof:** Let $A_4 = ZA_3(n, c)Z^{-1}$ where $Z$ is defined by

$$Z = \begin{bmatrix} \frac{1}{c} I_{n+1} & 0 \\ 0 & I_{n+1} \end{bmatrix}.$$  

A simple calculation reveals that $A_4 = cA_3(n, 1)$. This proves that $\lambda(A_3(n, c)) = c \lambda(A_3(n, 1))$, which in turn proves (16); the last part of the statement of the proposition is a consequence of Prop. 3. \[\square\]

The method proposed in this work relies on the observation that matrix $A_4$ is better scaled than $A_3$, in which case the eigenvalue problem of the former is better conditioned than the eigenvalue problem of the latter [7]. Essentially, we change the system coordinates $U_3$ in (14) to $U_4 = ZU_3$, which leads to the semidiscrete system

$$\begin{cases}
U_{4,t} = A_4 U_4, \\
U_4(0) = [0_{1 \times (n+1)}, f(x_0), \ldots, f(x_n)]^T,
\end{cases}$$

and propose to integrate in time by the fourth-order Runge-Kutta method. The method, which we denote by CPS-RK, can be described as follows

**Method 3 (CPS-RK):** For $v \geq 0$, calculate

$$U_4(t_{v+1}) = U_4(t_v) + \frac{\Delta t}{6}(F_1 + 2F_2 + 2F_3 + F_4)$$.
where

\[ F_1 = A_4 U_4(t_v), \quad F_2 = A_4 \left( U_4(t_v) + \frac{\Delta t}{2} F_1 \right) \]
\[ F_3 = A_4 \left( U_4(t_v) + \frac{\Delta t}{2} F_2 \right), \quad F_4 = A_4 \left( U_4(t_v) + \Delta t F_3 \right). \]

The principal feature of using the semidiscrete system (17) is that now \( u(x_j, t) \) can be calculated directly from initial data, as opposed to the approaches of Jackiewicz and Renaut which estimate \( u \) from approximations to \( u_1 \) and \( u_2 \). As far as accuracy is concerned, CPS-RK will certainly outperform Method 1 but nothing can be affirmed with respect to its performance compared with Method 2 because both methods are fourth-order accurate. Even so, it is worth emphasizing that CPS-RK has the capability to construct highly accurate approximations by simply taking the stepsize \( \Delta t \) fairly small.

3 Stability of the proposed method

In this section we discuss the choice of the maximum stepsize that assures stable integration for a given spatial grid. A discussion about this involving matrices \( A_1 \) and \( A_2 \) is carried out by Jackiewicz and Renaut in [11]. They analyze the spectra and \( \epsilon \)-pseudospectra of these matrices and conclude that the maximum stepsize that assures stability of his RK-based method can be chosen by requiring the spectrum of the system matrix scaled by \( \Delta t \) to lie within the stability region of the method. Limits on allowable stepsizes for stability of both formulations for speed \( c = 1 \) are also reported (Table 1 in [11]). Ignoring the fact that most part of these reported values are incorrect because they contradict the statement of Proposition 1, what must be observed here is that the conclusion of these authors concerning the appropriateness of eigenvalue analysis for stability is actually correct. This is supported by the results displayed in Figure 2. With regard to the stability limits on the stepsizes for the proposed method, they can be found by coupling the line of analysis of Jackiewicz and Renaut with Prop. 4. They are displayed in Table 1 for several values of \( n \). Observe that in the sense of “eigenvalue stability”, these values apply to all matrices \( A_i, i = 1, 2, 3, 4 \), since by Prop. 4, all these matrices share the same spectral radius.

Since these stability limits were shown to hold in practice in [11] when solving the semidiscrete system involving matrix \( A_2 \), it remains to verify whether the same occurs in conjunction with the system matrix \( A_4 \) of the proposed method. As demonstrated in [14] and illustrated in [17, 18], the right way to carry out such verification is via pseudoeigenvalues. In words, this requires choosing the timestep in such a way that the pseudospectra of \( A_4 \) lie within a distance \( O(\epsilon) + \Delta t \) of the scaled region of stability as \( \epsilon \to 0 \) and \( \Delta t \to 0 \) [14, 18]. The spectra and \( \epsilon \)-pseudospectra of all matrices \( A_i \) for \( n = 31 \) and the scaled region of stability of RK method are displayed in Figure 2.

Several conclusions can be drawn from Figure 2. First, it shows that the eigenvalues of matrices \( A_1 \) and \( A_2 \) are almost equally sensitive to perturbation, as predicted in theory (see Prop. 2), and that among the eigenvalue problems associated with matrices \( A_4 \ (i = 1 : 4) \), the most sensitive to perturbation is the one associated with matrix \( A_3 \). Moreover, in view that the \( \epsilon \)-pseudospectra of \( A_4 \) are well within the scaled region of stability, the most important
Table 1: Maximum stable stepsize for arbitrary speed $c$, for matrices $A_i$, $i = 1, 2, 3, 4$ of several orders, according to eigenvalue stability analysis

<table>
<thead>
<tr>
<th>$n + 1$</th>
<th>$c\Delta t_{\text{max}}$ for $A_i$, $i = 1 : 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.61413877092330</td>
</tr>
<tr>
<td>8</td>
<td>0.23643588530295</td>
</tr>
<tr>
<td>16</td>
<td>0.06906986008198</td>
</tr>
<tr>
<td>32</td>
<td>0.01680029485722</td>
</tr>
<tr>
<td>64</td>
<td>0.00410202619991</td>
</tr>
<tr>
<td>128</td>
<td>0.00101143197421</td>
</tr>
</tbody>
</table>

Conclusion is that the pseudoeigenvalue stability criterion is indeed satisfied, thus verifying that the $\Delta t_{\text{max}}$ calculated using the standard eigenvalue analysis is appropriate in this case.

Figure 2: Region of absolute stability of fourth-order Runge-Kutta method scaled by $1/\Delta t_{\text{max}}$ and spectra and $\epsilon$-pseudospectra of system matrices $A_1$ (top left) $A_2$ (top right), $A_3$ (bottom left), and $A_4$ (bottom right), for $\epsilon = 10^{-9}, 10^{-8}, \ldots, 10^{-3}$ (from inner to outer), $n = 31$, $c = 5$, and $\Delta t_{\text{max}} = 0.01680$
4 Numerical example

We give a numerical example to illustrate the effectiveness of the proposed method. The example addresses the solution of a 1D wave equation with speed $c = 5$ and initial condition defined by

$$u(x, 0) = e^{-100(x-0.5)^2}$$

The problem was solved on the interval $[0, 0.12]$, $n + 1 = 64$, and $\Delta t = 0.004/c = 0.0008$ (see Table 1). This test problem is taken from [11] in which the solution is computed on the interval $[0,0.6]$ with $c = 1$ and $\Delta t = 0.004$. For comparison, we also solve the problem using Method 1 and Method 2. For Method 2 we provide exact internal stages $u_1(x_j, t_v + c_i \Delta t)$ (see, (10)), and we do so to illustrate that this method should not outperform CPS-RK. All numerical computation were carried out in MATLAB. In our implementation we use a slightly modified version of function cheb.m by Trefethen [18] (see also, Trefethen and Embree [17, Pag. 410].

The numerical results of PCS-RK method are presented in Figure 3, in which the exact solution is plotted by solid line and approximate solutions by small circles.

![Figure 3: Exact and CPS-RK-based approximate solution for 1D wave equation](image)

To assess the performance of the methods, the error of the approximate solutions was cal-
culated. This is simple to do since for 1D problems the exact solution is known in closed form. Figure 4 presents the error in absolute value of numerical solutions calculated by the tested methods. The numerical results are clear: even that Method 2 is shown in an advantageous light because exact internal stages were provided, it does not outperform CPS-RK, as expected.

![Figure 4: Error in absolute value of approximate solutions at time level $t = 0.12$](image)

5 Concluding remarks

We have presented a method for numerical simulation of 1D wave propagation with absorbing boundary conditions that relies on the Runge-Kutta method and that does not use a first order hyperbolic system. Doing so, we maintained the simplicity and accuracy of the Runge-Kutta method and avoided the difficulty of recovering the solution $u$ from low order approximations to derivatives $u_t$, which is typical of methods like Method 1 and Method 2. Stability of the proposed method was also discussed and the stability limits on the stepsize for arbitrary speed were calculated and verified in practice.

The method we have presented can be extended to 2D wave equations without difficulties. Indeed, a block companion approach for 2D problems appears naturally following the same line of reasoning as the one used in Section 3. This extension was not included here because the error of the approximate solutions for 2D problems as well as the assessment of how much reflection is avoided at the artificial numerical boundaries is not as simple to estimate as for the 1D case. This is the subject of future work.

Acknowledgments. This research was sponsored by CNPq Brasil, grants 300487/94-O(NV), 473481/008-1. The author wishes to thank Z. Jackiewicz for providing clarification concerning the definition of system matrices $A_1$ and $A_2$ in [11].
References


