Noname manuscript No.
(will be inserted by the editor)

Leapfrog time-stepping for Hermite methods

Arturo Vargas* · **Thomas Hagstrom** ·
Jesse Chan · **Tim Warburton**

Received: date / Accepted: date

Abstract We introduce Hermite-leapfrog methods for first order linear wave systems. The new Hermite-leapfrog methods pair leapfrog time-stepping with the Hermite methods of Goodrich and co-authors [7]. The new schemes stagger field variables in both time and space and are high-order accurate for equations with smooth solutions and coefficients. We provide a detailed description of the method and demonstrate that the method conserves variable quantities. Higher dimensional versions of the method are constructed via tensor products. Numerical evidence and rigorous analysis in one space dimension establish stability and high-order convergence. Experiments demonstrating efficient implementations on a graphics processing unit are also presented.

Keywords High order · Hermite · Leapfrog

1 Introduction

Simulations of wave propagation play a crucial role in science and engineering. In applications to geophysics, they are the engine of many seismic imaging algorithms. For electrical engineers, they can be a useful tool for the design of radars and antennas. In these applications achieving high fidelity simulations is challenging due to the inherent issues in modeling highly oscillatory waves and the associated high computational cost of high-resolution simulations. Thus the ideal numerical

*Corresponding author: A. Vargas
Lawrence Livermore National Laboratory,
7000 East Ave, Livermore, CA 94550
E-mail: vargas45@llnl.gov

T. Hagstrom
Department of Mathematics, Southern Methodist University, Dallas, TX 75275

J. Chan
Department of Computational and Applied Mathematics, Rice University, 6100 Main Street, Houston, TX 77005

T. Warburton
Department of Mathematics, Virginia Tech, Blacksburg, VA 24060

method should be able to capture high-frequency waves and be suitable for parallel computing.

The staggered finite-difference time-domain (FDTD) method has been widely adopted as the method of choice for simulating electromagnetic and acoustic wave propagation on structured grids [3,9]. In the context of electromagnetics the method was first pioneered by K.S. Yee in [17]; while the discretization for the acoustic wave equation was introduced by Virieux in [6] and Levander in [10]. The underlying idea of staggered FDTD methods is to stagger field variables both in time and space and propagate the solution using leapfrog time-stepping [11]. This type of discretization is known to be efficient and easy to implement but is typically only second order accurate and struggles to resolve highly-oscillatory waves. Fourth order accurate variants of the method have been proposed in the literature [16,18] but our work differs as we present methods of arbitrary order.

In contrast to FDTD methods where derivatives are approximated via difference formulas, the Hermite methods of Goodrich and co-authors [7] combine Hermite-Birkhoff interpolation and a staggered (dual) grid to produce high-order numerical methods for first order hyperbolic problems. The resulting "Hermite methods" carry out a localized polynomial reconstruction within cells of a structured grid and propagate the solution completely independent of neighboring cells via Hermite-Taylor time-stepping. The remarkable feature of Hermite methods is that the CFL condition of the resulting scheme is independent of the approximation order [7]. A formal introduction to these methods may be found in [7,13, 15].

Though Hermite methods are relatively new, several variations have been proposed to enhance their applicability and efficiency. To enable geometric flexibility, Hermite methods have been combined with discontinuous Galerkin methods [5]. Adaptive variants of Hermite methods have been introduced by means of p-adaptivity in [4], and preliminary work on h-adaptivity has been presented in [2]. Variations of the method which do not require a dual grid have been introduced in [15], and a flux conservative Hermite method has been introduced in [8].

Inspired by the favorable features of Hermite methods, this work presents a new variant which combines leapfrog time stepping and staggered solutions in space and time. The resulting "Hermite-leapfrog" scheme may be viewed as a high order variant of the Yee scheme for linear wave systems. We present this work as complimentary to the recent work of Appelö and co-authors [1] in which a similar time-stepping scheme was introduced for the second order acoustic wave equation. In this work we focus on first order linear wave systems and note that an early version appeared in the thesis of Vargas [13].

The remainder of this article proceeds as follows. We start with a detailed description in one space dimension. We then demonstrate that the method conserves variable quantities, establishing stability, and prove general error estimates in one space dimension. In addition we look at the dispersion relation, which provides an explanation for aspects of the method's accuracy observed in the numerical experiments. Experimental convergence rates are reported for both constant and spatially varying wave speed. In two dimensions we discuss incorporating zero Dirichlet boundary conditions and demonstrate the method's ability to resolve highly oscillatory waves. Lastly, we demonstrate computational efficiency on a graphics processing unit.

2 Description of the method

To introduce the method, we consider the following one-dimensional linear wave system

$$\begin{aligned} \frac{\partial p}{\partial t} &= c\frac{\partial v}{\partial x}, \quad \frac{\partial v}{\partial t} = c\frac{\partial p}{\partial x} \\ p(x,t_0) &= f(x), \quad v(x,t_0+\Delta t/2) = g(x). \end{aligned} \tag{1}$$

In this example, c corresponds to the speed of the propagating wave. The degrees of freedom of the Hermite-leapfrog method are the function value and first m derivatives at the nodes of a structured grid. An m^{th} order scheme discretizes the equations by staggering pressure and velocity approximations in both space and time. The discretization of the pressure occurs on a primary grid Ω while the discretization of the velocity term is maintained on a staggered dual grid $\tilde{\Omega}$. For the purpose of introducing the method we assume that initial conditions are given at grid-points staggered in both time and space. We define the primary grid as a collection of K equidistant points

$$\Omega = \{x_j : \quad x_j = x_{\min} + jh_x, \quad j = 0,\ldots,K-1\}, \tag{2}$$

analogously the dual grid, which holds velocity approximations, is defined as

$$\tilde{\Omega} = \{x_{j+1/2} = x_{\min} + \left(j+\frac{1}{2}\right)h_x, \quad j = 0,\ldots,K-1\}. \tag{3}$$

Approximations of each variable are carried out through local Taylor series expansions at each node x_j. The expansion takes the form

$$u(x) \approx u_j(x) = \sum_{i=0}^{m} \tilde{u}_i \left(\frac{x-x_j}{h}\right)^i, \tag{4}$$

where $\tilde{u}_i$ corresponds to scaled approximations of the i^{th} derivative of the function $u(x)$ evaluated at x_j.

$$\tilde{u}_i = \frac{h^i}{i!}\frac{d^i u}{dx^i}(x_j), \quad i = 0,\ldots m.$$

Next, we derive the time-stepping algorithm of the Hermite-leapfrog scheme by considering the continuous infinite temporal series for the pressure variable at t and $t+\Delta t$.

$$p(x,t+\Delta t) = \sum_{j=0} \frac{(\Delta t/2)^j}{j!}\frac{\partial^j p(x,t+\frac{\Delta t}{2})}{\partial t^j}, \tag{5a}$$

$$p(x,t) = \sum_{j=0} \frac{(-\Delta t/2)^j}{j!}\frac{\partial^j p(x,t+\frac{\Delta t}{2})}{\partial t^j}. \tag{5b}$$

Subtracting equations (5a) and (5b) yields

$$p(x,t+\Delta t) - p(x,t) = \sum_{j=1,odd} 2\frac{(\Delta t/2)^j}{j!}\frac{\partial^j p(x,t+\frac{\Delta t}{2})}{\partial t^j}, \tag{6}$$

where the limit of the sum is determined by the order of the method. Temporal derivatives are then exchanged for spatial derivatives by means of the Cauchy-Kowalevsky recurrence relation [7]; in the case of constant coefficients the recurrence relation simplifies to

$$\frac{\partial^r p}{\partial t^r} = c^r \frac{\partial^r v}{\partial x^r}, \quad r \in \{1, 3, 5, \dots\}. \tag{7}$$

yielding the time-stepping algorithm

$$\frac{\partial^s p(x, t+\Delta t)}{\partial x^s} = \frac{\partial^s p(x,t)}{\partial x^s} + 2 \sum_{j=1,odd} \frac{(c\Delta t/2)^j}{j!} \frac{\partial^{j+s} v(x, t+\Delta t/2)}{\partial x^{j+s}}. \tag{8}$$

The discrete version at each node x_j is then given by replacing the pressure and velocity terms with the local Taylor series approximations

$$\frac{\partial^s p_j^{n+1}(x)}{\partial x^s} \approx \frac{\partial^s p_j^n(x)}{\partial x^s} + 2 \sum_{j=1,odd}^{M} \frac{(c\Delta t/2)^j}{j!} \frac{\partial^{j+s} v_j^{n+1/2}(x)}{\partial x^{j+s}}. \tag{9}$$

The superscript is used to denote time step, and the subscript denotes the node around which the polynomial is expanded. Equality holds if M is assumed to be infinite otherwise the order of the approximation depends on the number of terms in the series.

Here we introduce the index s to correspond to the degree of freedom, $s \in \{0, \dots, m\}$, at each grid point. The key ingredient here is the approximation of the spatial derivatives (the right hand side of equation (8)). The approximation is carried out by means of Hermite-Birkhoff interpolation as done in the classic Hermite method [7]. The interpolation procedure constructs a polynomial $\tilde{v}_j$ by interpolating the function value and derivatives at $R_j = x_{j+1/2}$ $L_j = x_{j-1/2}$

$$\left.\frac{\partial^i \tilde{v}_j}{\partial x^i}\right|_{L_j} = \left.\frac{\partial^i v_j^L}{\partial x^i}\right|_{L_j}, \qquad \left.\frac{\partial^i \tilde{v}_j}{\partial x^i}\right|_{R_j} = \left.\frac{\partial^i v_j^R}{\partial x^i}\right|_{R_j}, \qquad i = 0, \dots, 2m+1.$$

Thus, the resulting polynomial reproduces the function value and first m derivatives at the left and right end points. Notably, the coefficients of the polynomial are the approximation of the function value and first $2m+1$ derivatives at the node x_j. As demonstrated in [15] this results in the following system

$$\begin{bmatrix} \mathbf{C}^L \\ \mathbf{C}^R \end{bmatrix} \tilde{\mathbf{v}} = \begin{bmatrix} \mathbf{v}^L \\ \mathbf{v}^R \end{bmatrix}, \tag{10}$$

where $\mathbf{C}^L, \mathbf{C}^R$ is defined as

$$\mathbf{C}^L_{ls} = \begin{cases} \left(\frac{L_j - \tilde{x}_j}{h}\right)^{s-l} \frac{1}{l!} \prod_{t=0}^{l-1} (s-t), & s \geq l \\ 0, & s < l \end{cases}$$

$$\mathbf{C}^R_{ls} = \begin{cases} \left(\frac{R_j - \tilde{x}_j}{h}\right)^{s-l} \frac{1}{l!} \prod_{t=0}^{l-1} (s-t), & s \geq l \\ 0, & s < l. \end{cases}$$

A full description of the interpolation procedure can be found in [7,13,15]. The coefficients of the polynomial are used to carry out the time stepping algorithm. Figure 1 provides an illustration of the stencil associated with the scheme. Next, we describe the stability and convergence properties of the method.

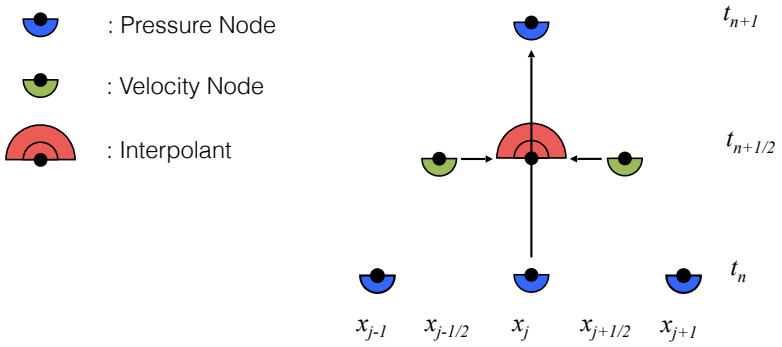


Fig. 1: The first step of the Hermite-leapfrog method is the use of Hermite-Birkhoff interpolation to approximate the function value and first $2m+1$ derivatives of the velocity field at node x_j. The solution of the pressure is then propagated via the Hermite-leapfrog time-stepping algorithm (equation (8)).

3 Stability and convergence

The essential properties of Hermite-Birkhoff interpolation used to establish stability and to estimate the error are proven in detail in the original paper [7]. First and foremost is the orthogonality lemma, which shows that the interpolation process is contracting in a certain Sobolev seminorm. Precisely, let $\mathcal{I}_m$ denote the Hermite interpolation operator employing derivative data up to order m; here we will not distinguish between interpolation on the two grids. We will also assume that all functions are periodic and sufficiently smooth. In one space dimension we use the Sobolev semi-inner-product

$$\langle f, g\rangle_{m+1} = \int \frac{\partial^{m+1} f}{\partial x^{m+1}} \cdot \frac{\partial^{m+1} g}{\partial x^{m+1}},$$

which we generalize in d dimensions to

$$\langle f, g\rangle_{m+1} = \int \frac{\partial^{(m+1)d} f}{\partial x_1^{m+1} \cdots \partial x_d^{m+1}} \cdot \frac{\partial^{(m+1)d} g}{\partial x_1^{m+1} \cdots \partial x_d^{m+1}}.$$

Then, as shown in [7], for any f and g we have

$$\langle \mathcal{I}_m f, g - \mathcal{I}_m g\rangle_{m+1} = 0. \tag{11}$$

Denoting by $|\cdot|_{m+1}$ the seminorm induced by these semi-inner-products, we thus have, by the Pythagorean Theorem

$$|f|_{m+1}^2 = |\mathcal{I}_m f|_{m+1}^2 + |f - \mathcal{I}_m f|_{m+1}^2. \tag{12}$$

The approximation properties of Hermite interpolants of smooth functions are generally what one expects. Restricting to $d = 1$ and using $\|\cdot\|$ to denote the L^2-norm we have, for mesh and function-independent constants, C

$$\|f - \mathcal{I}_m f\| \leq Ch^{2m+2} \|\frac{\partial^{2m+2} f}{\partial x^{2m+2}}\|, \tag{13}$$

$$\|\frac{\partial^{m+1}(f - \mathcal{I}_m f)}{\partial x^{m+1}}\| \leq Ch^{m+1} \|\frac{\partial^{2m+2} f}{\partial x^{2m+2}}\|. \tag{14}$$

We will also use (see [7]):

$$\|f - \mathcal{I}_m f\| \leq Ch^{m+1} \|\frac{\partial^{m+1}(f - \mathcal{I}_m f)}{\partial x^{m+1}}\|. \tag{15}$$

3.1 Conservation

We now construct variables which are conserved during the solution process. We begin by introducing the operators $\mathcal{S}_\pm$ defined via Fourier series

$$\widehat{\mathcal{S}_\pm f} = e^{\pm ic(k_1^2 + \ldots + k_d^2)^{1/2} \Delta t/2} \hat{f}(k), \tag{16}$$

where c denotes the speed of wave propagation. Note that for $d = 1$ the branch can be chosen so that these are simply shift operators

$$\mathcal{S}_\pm f(x) = f(x \pm c\Delta t/2), \quad d = 1,$$

where c is the speed of wave propagation. By Parseval's Theorem these operators preserve all Sobolev seminorms and in addition satisfy

$$\mathcal{S}_+ \mathcal{S}_- = \mathcal{S}_- \mathcal{S}_+ = I. \tag{17}$$

Using them we can write down a two-level conservation condition which holds at the continuous level.

3.1.1 One space dimension

For $d = 1$, note that (8) translated to Fourier variables leads to the exact solution formula

$$\begin{aligned} \hat{p}(k, t + \Delta t) &= \hat{p}(k, t) + 2 \sum_{j=1,\text{odd}} \frac{(ikc\Delta t/2)^j}{j!} \hat{v}(k, t + \Delta t/2) \\ &= \hat{p}(k, t) + \left(e^{ick\Delta t/2} - e^{-ick\Delta t/2} \right) \hat{v}(k, t + \Delta t/2), \end{aligned}$$

which, along with the analogous formulas for the evolution of v implies

$$p(x, t + \Delta t) = p(x, t) + \mathcal{S}_+ v(x, t + \Delta t/2) - \mathcal{S}_- v(x, t + \Delta t/2), \tag{18}$$

$$v(x, t + \Delta t/2) = v(x, t - \Delta t/2) + \mathcal{S}_+ p(x, t) - \mathcal{S}_- p(x, t). \tag{19}$$

Define

$$P_\pm(x,t) = p(x,t) \mp \mathcal{S}_\pm v(x, t - \Delta t/2), \tag{20}$$
$$V_\pm(x,t+\Delta t/2) = v(x,t+\Delta t/2) \mp \mathcal{S}_\pm p(x,t). \tag{21}$$

Then, using (17) we can rewrite (18)-(19) as

$$P_\pm(x,t+\Delta t) = \mp \mathcal{S}_\mp V_\pm(x,t+\Delta t/2), \tag{22}$$
$$V_\pm(x,t+\Delta t/2) = \mp \mathcal{S}_\mp P_\pm(x,t), \tag{23}$$

which yields the conservation conditions, valid for any Sobolev seminorm,

$$\|P_\pm(\cdot,t+\Delta t)\| = \|V_\pm(\cdot,t+\Delta t/2)\|, \tag{24}$$
$$\|V_\pm(\cdot,t+\Delta t/2)\| = \|P_\pm(\cdot,t)\|. \tag{25}$$

Now consider the approximate evolution. The essential point is that the Hermite data updated according to (8) is in fact the exact data for the update of the global piecewise polynomial Hermite interpolant so long as the CFL condition $c\Delta t/\Delta x < 1$ is satisfied; this simply follows from domain-of-dependence considerations. Therefore we can write the discrete evolution as

$$\begin{aligned} p^h(x,t+\Delta t) = {} & p^h(x,t) + \mathcal{I}_m \mathcal{S}_+ v^h(x,t+\Delta t/2) \\ & - \mathcal{I}_m \mathcal{S}_- v^h(x,t+\Delta t/2), \end{aligned} \tag{26}$$
$$\begin{aligned} v^h(x,t+\Delta t/2) = {} & v^h(x,t-\Delta t/2) + \mathcal{I}_m \mathcal{S}_+ p^h(x,t) \\ & - \mathcal{I}_m \mathcal{S}_- p^h(x,t), \end{aligned} \tag{27}$$

Defining $P^h_\pm$, $V^h_\pm$ as in (20)-(21) with the discrete variables p^h and v^h replacing the continuous ones and using the fact that $p^h = \mathcal{I}_m p^h$, $v^h = \mathcal{I}_m v^h$, we derive evolution equations

$$\begin{aligned} P^h_\pm(x,t+\Delta t) = {} & \mp \mathcal{I}_m \mathcal{S}_\mp V^h_\pm(x,t+\Delta t/2) \\ & \mp (1 - \mathcal{I}_m) \mathcal{S}_\pm V^h_\mp(x,t+\Delta t/2), \end{aligned} \tag{28}$$
$$\begin{aligned} V^h_\pm(x,t+\Delta t/2) = {} & \mp \mathcal{I}_m \mathcal{S}_\mp P^h_\pm(x,t) \\ & \mp (1 - \mathcal{I}_m) \mathcal{S}_\pm P^h_\mp(x,t). \end{aligned} \tag{29}$$

Now we get conservation conditions for the sums of the Sobolev seminorms of $P^h_\pm$ and $V^h_\pm$ using the orthogonality lemma (12). Define

$$Q^h(t) = \left| P^h_+(\cdot,t) \right|^2_{m+1} + \left| P^h_-(\cdot,t) \right|^2_{m+1} \tag{30}$$
$$R^h(t+\Delta t/2) = \left| V^h_+(\cdot,t+\Delta t/2) \right|^2_{m+1} + \left| V^h_-(\cdot,t+\Delta t/2) \right|^2_{m+1}. \tag{31}$$

Then

$$\begin{aligned} Q^h(t+\Delta t) = {} & \left| \mathcal{I}_m \mathcal{S}_- V^h_+(\cdot,t+\Delta t/2) \right|^2_{m+1} + \left| (1-\mathcal{I}_m) \mathcal{S}_+ V^h_-(\cdot,t+\Delta t/2) \right|^2_{m+1} \\ & + \left| \mathcal{I}_m \mathcal{S}_+ V^h_-(\cdot,t+\Delta t/2) \right|^2_{m+1} + \left| (1-\mathcal{I}_m) \mathcal{S}_- V^h_+(\cdot,t+\Delta t/2) \right|^2_{m+1} \\ = {} & \left| \mathcal{S}_- V^h_+(\cdot,t+\Delta t/2) \right|^2_{m+1} + \left| \mathcal{S}_+ V^h_-(\cdot,t+\Delta t/2) \right|^2_{m+1} \\ = {} & R^h(t+\Delta t/2). \end{aligned} \tag{32}$$

Similarly

$$R^h(t+\Delta t/2) = Q^h(t), \tag{33}$$

so we have for all n

$$Q^h(n\Delta t) = R^h((n+1/2)\Delta t) = Q^h(0). \tag{34}$$

3.1.2 Extension to d space dimensions

In higher dimensions we consider the generalization of (1)

$$\begin{aligned} \frac{\partial p}{\partial t} &= c\nabla\cdot\mathbf{v} \\ \frac{\partial \mathbf{v}}{\partial t} &= c\nabla p, \end{aligned} \tag{35}$$

though other systems, such Maxwell's equations, could be similarly treated. The formulas analogous to (8) are now

$$p(\mathbf{x}, t+\Delta t) = p(\mathbf{x},t) + 2\sum_{j=1,odd} \frac{(c\Delta t/2)^j}{j!}\nabla^{j-1}\nabla\cdot\mathbf{v}(\mathbf{x}, t+\Delta t/2), \tag{36}$$

$$\mathbf{v}(\mathbf{x}, t+\Delta t/2) = \mathbf{v}(\mathbf{x},t-\Delta t/2) + 2\sum_{j=1,odd} \frac{(c\Delta t/2)^j}{j!}\nabla^{j-1}\nabla p(\mathbf{x},t). \tag{37}$$

In Fourier space, denote the normalized $\mathbf{k}$-vector by

$$\hat{\mathbf{k}} = \frac{\mathbf{k}}{|k|},$$

noting that in the formulas below the undefined value at $\mathbf{k} = \mathbf{0}$ plays no role. However we will need to specify its value when defining the conserved quantities and we do so by setting it to be some arbitrarily-chosen unit d-vector. We then find

$$\begin{aligned} \hat{p}(\mathbf{k}, t+\Delta t) &= \hat{p}(\mathbf{k},t) \\ &\quad + \left(e^{ic(k_1^2+\ldots+k_d^2)^{1/2}\Delta t/2} - e^{-ic(k_1^2+\ldots+k_d^2)^{1/2}\Delta t/2}\right)\hat{\mathbf{k}}\cdot\hat{\mathbf{v}}(\mathbf{k}, t+\Delta t/2), \end{aligned} \tag{38}$$

$$\begin{aligned} \hat{\mathbf{v}}(\mathbf{k}, t+\Delta t/2) &= \hat{\mathbf{v}}(\mathbf{k},t-\Delta t/2) \\ &\quad + \left(e^{ic(k_1^2+\ldots+k_d^2)^{1/2}\Delta t/2} - e^{-ic(k_1^2+\ldots+k_d^2)^{1/2}\Delta t/2}\right)\hat{\mathbf{k}}\hat{p}(\mathbf{k},t). \end{aligned} \tag{39}$$

Defining $\mathcal{G}$ to be the operator given in Fourier space by multiplication by $\hat{\mathbf{k}}$ and by $\mathcal{G}^*$ its adjoint, noting also that $\mathcal{G}^*\mathcal{G} = I$ we rewrite the continuous evolution formulas. We also define

$$\mathbf{v}_d = \mathcal{G}\mathcal{G}^*\mathbf{v}, \quad \mathbf{v}_c = \mathbf{v} - \mathbf{v}_d$$

and note that

$$\mathcal{G}^*\mathbf{v}_d = \mathcal{G}^*\mathbf{v}, \quad \mathcal{G}^*\mathbf{v}_c = 0.$$

Then we have

$$p(\mathbf{x},t+\Delta t) = p(\mathbf{x},t) + \mathcal{S}_+\mathcal{G}^*\mathbf{v}_d(\mathbf{x},t+\Delta t/2) - \mathcal{S}_-\mathcal{G}^*\mathbf{v}_d(\mathbf{x},t+\Delta t/2), \tag{40}$$

$$\mathbf{v}_d(\mathbf{x},t+\Delta t/2) = \mathbf{v}_d(\mathbf{x},t-\Delta t/2) + \mathcal{G}\mathcal{S}_+p(\mathbf{x},t) - \mathcal{G}\mathcal{S}_-p(\mathbf{x},t), \tag{41}$$

$$\mathbf{v}_c(\mathbf{x},t+\Delta t/2) = \mathbf{v}_c(\mathbf{x},t-\Delta t/2). \tag{42}$$

Introducing

$$P_\pm(\mathbf{x},t) = p(\mathbf{x},t) \mp \mathcal{S}_\pm\mathcal{G}^*\mathbf{v}_d(\mathbf{x},t-\Delta t/2), \tag{43}$$

$$\mathbf{V}_\pm(\mathbf{x},t+\Delta t/2) = \mathbf{v}_d(\mathbf{x},t) \mp \mathcal{G}\mathcal{S}_\pm p(\mathbf{x},t), \tag{44}$$

we find

$$P_\pm(\mathbf{x},t+\Delta t) = \mp\mathcal{S}_\mp\mathcal{G}^*\mathbf{V}_\pm(\mathbf{x},t+\Delta t/2), \tag{45}$$

$$\mathbf{V}_\pm(\mathbf{x},t+\Delta t/2) = \mp\mathcal{G}\mathcal{S}_\mp P_\pm(\mathbf{x},t). \tag{46}$$

Using (42), (45)-(46) and the fact that

$$\|\mathcal{G}^*\mathbf{V}_\pm\| = \|\mathbf{V}_\pm\|$$

the conservation of arbitrary Sobolev seminorms of $\mathbf{v}_c$, $P_\pm$, and $\mathbf{V}_\pm$ follows:

$$\|P_\pm(\cdot,t+\Delta t)\| = \|\mathbf{V}_\pm(\cdot,t+\Delta t/2)\| \tag{47}$$

$$\|\mathbf{V}_\pm(\cdot,t+\Delta t/2)\| = \|P_\pm(\cdot,t)\| \tag{48}$$

$$\|\mathbf{v}_c(\cdot,t+\Delta t/2)\| = \|\mathbf{v}_c(\cdot,t-\Delta t/2)\|. \tag{49}$$

Similarly we can establish conservation conditions for the Hermite-leapfrog updates. The main new issue which arises is the fact that the interpolation operators do not commute with $\mathcal{G}$. Defining

$$\mathbf{v}_d^h = \mathcal{G}\mathcal{G}^*\mathbf{v}^h, \quad \mathbf{v}_c^h = \mathbf{v}^h - \mathbf{v}_d^h,$$

$$P_\pm^h = p^h \mp \mathcal{S}_\pm\mathcal{G}^*\mathbf{v}_d^h, \quad \mathbf{V}_\pm^h = \mathbf{v}_d^h \mp \mathcal{G}\mathcal{S}_\pm p^h,$$

we derive the following update formulas

$$P_\pm^h(\mathbf{x},t+\Delta t) = \mp\mathcal{I}_m\mathcal{S}_\mp\mathcal{G}^*\mathbf{V}_\pm^h(\mathbf{x},t+\Delta t/2) \mp(1-\mathcal{I}_m)\mathcal{S}_\pm\mathcal{G}^*\mathbf{V}_\mp^h(\mathbf{x},t+\Delta t/2), \tag{50}$$

$$\mathbf{V}_\pm^h(\mathbf{x},t+\Delta t/2) + \mathbf{v}_c^h(\mathbf{x},t+\Delta t/2) = \mp\mathcal{I}_m\left(\mathcal{G}\mathcal{S}_\mp P_\pm^h(\mathbf{x},t) - \mathbf{v}_c^h(\mathbf{x},t-\Delta t/2)\right) \mp(1-\mathcal{I}_m)\left(\mathcal{G}\mathcal{S}_\pm P_\mp^h(\mathbf{x},t) - \mathbf{v}_c^h(\mathbf{x},t-\Delta t/2)\right). \tag{51}$$

Define

$$Q^h(t) = \left|P_+^h(\cdot,t)\right|_{m+1}^2 + \left|P_-^h(\cdot,t)\right|_{m+1}^2 + \left|\mathbf{v}_c^h(\cdot,t-\Delta t/2)\right|_{m+1}^2 \tag{52}$$

$$R^h(t+\Delta t/2) = \left|\mathbf{V}_+^h(\cdot,t+\Delta t/2)\right|_{m+1}^2 + \left|\mathbf{V}_-^h(\cdot,t+\Delta t/2)\right|_{m+1}^2 + \left|\mathbf{v}_c^h(\cdot,t+\Delta t/2)\right|_{m+1}^2. \tag{53}$$

Then, since $\mathbf{v}_c^h$ is orthogonal in the Sobolev semi-inner products to the range of $\mathcal{G}$ we find as above

$$Q^h(t+\Delta t) = R^h(t+\Delta t/2), \tag{54}$$
$$R^h(t+\Delta t/2) = Q^h(t), \tag{55}$$

from which we finally conclude

$$Q^h(n\Delta t) = R^h((n+1/2)\Delta t) = Q^h(0). \tag{56}$$

3.2 Convergence in one dimension

We now exploit the energy estimates along with the approximation properties of Hermite interpolation (13)-(15) to establish convergence. We define the errors by

$$e_p = p - p^h, \quad e_v = v - v^h.$$

They satisfy the evolution formula

$$\begin{aligned} e_p(x,t+\Delta t) &= e_p(x,t) + \mathcal{I}_m\mathcal{S}_+ e_v(x,t+\Delta t/2) \\ &\quad -\mathcal{I}_m\mathcal{S}_- e_v(x,t+\Delta t/2) + (1-\mathcal{I}_m)p(x,t+\Delta t), \end{aligned} \tag{57}$$
$$\begin{aligned} e_v(x,t+\Delta t/2) &= e_v(x,t-\Delta t/2) + \mathcal{I}_m\mathcal{S}_+ e_p(x,t) \\ &\quad -\mathcal{I}_m\mathcal{S}_- e_p(x,t) + (1-\mathcal{I}_m)v(x,t+\Delta t/2). \end{aligned} \tag{58}$$

Introducing the variables corresponding to the conserved quantities

$$E_{p,\pm}(x,t) = e_p(x,t) \mp \mathcal{S}_\pm e_v(x,t-\Delta t/2), \tag{59}$$
$$E_{v,\pm}(x,t+\Delta t/2) = e_v(x,t+\Delta t/2) \mp \mathcal{S}_\pm e_p(x,t), \tag{60}$$

and following the previous calculations we derive

$$\begin{aligned} E_{p,\pm}(x,t+\Delta t) &= \mp\mathcal{I}_m\mathcal{S}_\mp E_{v,\pm}(x,t+\Delta t/2) \\ &\quad \mp(1-\mathcal{I}_m)\mathcal{S}_\pm E_{v,\mp}(x,t+\Delta t/2) \\ &\quad +(1-\mathcal{I}_m)(p(x,t+\Delta t)-p(x,t)), \end{aligned} \tag{61}$$
$$\begin{aligned} E_{v,\pm}(x,t+\Delta t/2) &= \mp\mathcal{I}_m\mathcal{S}_\mp E_{p,\pm}(x,t) \\ &\quad \mp(1-\mathcal{I}_m)\mathcal{S}_\pm E_{p,\mp}(x,t) \\ &\quad +(1-\mathcal{I}_m)(v(x,t+\Delta t/2)-v(x,t-\Delta t/2)). \end{aligned} \tag{62}$$

Noting that for sufficiently smooth solutions

$$|(1-\mathcal{I}_m)(p(x,t+\Delta t)-p(x,t))|_{m+1} = O\left(\Delta t \cdot h^{m+1}\right),$$
$$|(v(x,t+\Delta t/2)-v(x,t-\Delta t/2))|_{m+1} = O\left(\Delta t \cdot h^{m+1}\right),$$

and defining

$$\mathcal{E}_p(t) = |E_{p,+}(\cdot,t)|_{m+1}^2 + |E_{p,-}(\cdot,t)|_{m+1}^2,$$
$$\mathcal{E}_v(t+\Delta t/2) = |E_{v,+}(\cdot,t+\Delta t/2)|_{m+1}^2 + |E_{v,-}(\cdot,t+\Delta t/2)|_{m+1}^2,$$

we find, where here and in what follows C denotes a mesh-independent quantity which will depend on the exact solution,

$$\begin{aligned}\mathcal{E}_p(t+\Delta t) \leq \mathcal{E}_v(t+\Delta t/2) + C\Delta t \cdot h^{m+1}\sqrt{\mathcal{E}_v(t+\Delta t/2)} \\ +C\Delta t^2 h^{2m+2}, \end{aligned} \tag{63}$$

$$\mathcal{E}_v(t+\Delta t/2) \leq \mathcal{E}_p(t) + C\Delta t \cdot h^{m+1}\sqrt{\mathcal{E}_p(t)} + C\Delta t^2 h^{2m+2}. \tag{64}$$

Defining

$$\hat{\mathcal{E}}^n = \max_{j\leq n}\left(\mathcal{E}_p(j\Delta t), \mathcal{E}_n((j-1/2)\Delta t)\right)$$

we have by summing (63)-(64) over j

$$\hat{\mathcal{E}}^n \leq \hat{\mathcal{E}}^0 + Ct_n h^{m+1}\sqrt{\hat{\mathcal{E}}^n} + Ct_n\Delta t \cdot h^{2m+2}.$$

Assuming, as would be true for smooth initial data and initialization of the scheme by Hermite interpolation

$$\hat{\mathcal{E}}^0 \leq Ch^{2m+2},$$

we derive our first error estimate

$$\hat{\mathcal{E}}^n \leq C(1+t_n^2)h^{2m+2}. \tag{65}$$

Now return to (61)-(62), which we rewrite as

$$\begin{aligned} E_{p,\pm}(x,t+\Delta t) = \mp\mathcal{S}_\mp E_{v,\pm}(x,t+\Delta t/2) \\ \mp(1-\mathcal{I}_m)\left(\mathcal{S}_\pm E_{v,\mp}(x,t+\Delta t/2) - \mathcal{S}_\mp E_{v,\pm}(x,t+\Delta t/2)\right) \\ +(1-\mathcal{I}_m)(p(x,t+\Delta t) - p(x,t)), \\ E_{v,\pm}(x,t+\Delta t/2) = \mp\mathcal{S}_\mp E_{p,\pm}(x,t) \\ \mp(1-\mathcal{I}_m)\left(\mathcal{S}_\pm E_{p,\mp}(x,t) - \mathcal{S}_\mp E_{p,\pm}(x,t)\right) \\ +(1-\mathcal{I}_m)(v(x,t+\Delta t/2) - v(x,t-\Delta t/2)). \end{aligned}$$

Computing the L^2 norms and using (15) along with (65) we deduce

$$\begin{aligned}\|E_{p,\pm}(\cdot,t+\Delta t)\| \leq \|E_{v,\pm}(\cdot,t+\Delta t/2)\| + C(1+t)h^{2m+2}, \\ \|E_{v,\pm}(\cdot,t+\Delta t/2)\| \leq \|E_{p,\pm}(\cdot,t)\| + C(1+t)h^{2m+2}.\end{aligned}$$

Assuming sufficiently accurate initial data and in addition that for some $\eta > 0$, $\Delta t > \eta h$ we sum these inequalities and deduce our second error estimate:

$$\|E_{p,\pm}(\cdot,n\Delta t)\| \leq C(1+t_n^2)h^{2m+1}, \tag{66}$$

$$\|E_{v,\pm}(\cdot,(n-1/2)\Delta t)\| \leq C(1+t_n^2)h^{2m+1}. \tag{67}$$

Finally, we rewrite (59)-(60)

$$\begin{aligned} e_p(x,t) = E_{p,\pm} \pm \mathcal{S}_\pm e_v(x,t-\Delta t/2), \\ e_v(x,t+\Delta t/2) = E_{v,\pm}(x,t+\Delta t/2) \pm \mathcal{S}_\pm e_p(x,t), \end{aligned}$$

which yields after taking norms

$$\begin{aligned}\|e_p(\cdot,t)\| \leq \|e_v(\cdot,t-\Delta t/2)\| + C(1+t^2)h^{2m+1}, \\ \|e_v(\cdot,t+\Delta t/2) = \|e_p(\cdot,t)\| + C(1+t^2)h^{2m+1}.\end{aligned}$$

Summing this we get an estimate of the solution error:

$$\|e_p(\cdot, n\Delta t)\| \leq C(1+t_n^3)h^{2m}, \tag{68}$$
$$\|e_v(\cdot, (n-1/2)\Delta t)\| \leq C(1+t_n^3)h^{2m}. \tag{69}$$

We note that in the numerical experiments we do not see the cubic growth in t which appears in (68)-(69). There is a technical barrier to the extension of the error analysis to higher spatial dimensions, namely that the seminorm used in the stability analysis cannot control the full interpolation error due to the fact that it is zero for functions of fewer than d variables. However we have not noticed any degradation in convergence rates for our experiments with $d = 2, 3$. What the analysis misses is the fact that, for m even, we observe convergence at order $2m+2$ rather than $2m$. In the next section we study the dispersion relation for the method which suggests an explanation for this phenomenon.

3.3 Dispersion

As a complement to the stability and error analysis given above, we also consider the dispersive properties of the method. Here we again assume spatial periodicity and expand the discrete solution data in a Fourier series:

$$\tilde{P}^h(x_j, t) = \sum_k \hat{\tilde{P}}(k,t)e^{ikx_j},$$
$$\tilde{V}^h(x_{j+1/2}, t) = \sum_k \hat{\tilde{V}}(k,t)e^{ikx_{j+1/2}}.$$

Here $\tilde{P}^h$ and $\tilde{V}^h$ denote the $m+1$-vectors of approximate function and derivative data at the nodes so that $\hat{\tilde{P}}$ and $\hat{\tilde{V}}$ are also $m+1$-vectors. Written in these variables the evolution takes the form:

$$\hat{\tilde{P}}(k, t+\Delta t) = \hat{\tilde{P}}(k,t) + \hat{D}(k,\Delta t)\hat{\tilde{V}}(k, t+\Delta t/2), \tag{70}$$
$$\hat{\tilde{V}}(k, t+\Delta t/2) = \hat{\tilde{V}}(k, t-\Delta t/2) + \hat{D}(k,\Delta t)\hat{\tilde{P}}(k,t), \tag{71}$$

where the existence of matrix $\hat{D}$ follows from the translation-invariance of the method. Obviously the solutions can be expressed using the eigenvalues, κ, and eigenvectors, $\hat{w}$ of $\hat{D}$ which we will compare to the exact values,

$$\kappa_c = 2i\sin(ck\Delta t/2), \quad \hat{w}_c = (1 \; ik \; \ldots \; (ik)^m/m!)^T.$$

To make this comparison at leading order we rescale the problem so that $h = 1$ and $\Delta t = \lambda$ and make an expansion for $k \ll 1$. Using the accuracy properties of the Hermite interpolation operators we have

$$\hat{D}\hat{w}_c = \kappa_c \hat{w}_c + O(k^{2m+2}). \tag{72}$$

We thus deduce the perturbation equation

$$\hat{D}(\hat{w} - \hat{w}_c) = (\kappa - \kappa_c)\hat{w}_c + \kappa(\hat{w} - \hat{w}_c) + O(k^{2m+2}). \tag{73}$$

To leading order we may replace $\hat{D}(k)$ by $\hat{D}(0)$, $\hat{w}_c(k)$ by $\hat{w}_c(0) = e_1$, and, since $\kappa_c(0) = 0$, ignore $\kappa(\hat{w} - \hat{w}_c)$. We then have two possibilities:

i. 0 is a simple eigenvalue of $\hat{D}(0)$, in which case there is a solution to (73) satisfying

$$\kappa - \kappa_c = O(k^{2m+2}), \quad \hat{w} - \hat{w}_c = O(k^{2m+2}), \tag{74}$$

ii. 0 is not a simple eigenvalue of $\hat{D}(0)$ in which case we do not expect (74) to hold.

To apply this result we check for solutions for a second eigenvector corresponding to the eigenvalue 0.

$$\hat{D}(0)\hat{g} = 0, \tag{75}$$

or a generalized eigenvector

$$\hat{D}(0)\hat{g} = e_1. \tag{76}$$

Note that solutions of these equations correspond to the existence of a nonconstant polynomial $g(x)$ of degree $2m+1$ satisfying for $j = 0, \ldots, m$

$$\frac{d^j g}{dx^j}(1/2) - \frac{d^j g}{dx^j}(-1/2) = \frac{d^j g}{dx^j}(\lambda/2) - \frac{d^j g}{dx^j}(-\lambda/2) = 0,$$

or the above equations except

$$g(\lambda/2) - g(-\lambda/2) = 1.$$

Moreover, since a constant function solves the homogeneous equation, we can in addition assume that $g(0) = 0$. Expanding g

$$g = \sum_{j=1}^{2m+1} g_j x^j$$

we see that they represent $2m+2$ linear equations with $2m+1$ unknown coefficients. Moreover, due to symmetry the equations involving even order derivatives only involve the $m+1$ coefficients of the odd powers of x while equations involving odd order derivatives only involve the m coefficients of even powers. The size of these subsystems differs in the cases $m = 2n$ and $m = 2n+1$:

$m = 2n$: We have $2n+2$ equations for $2n+1$ coefficients of odd powers and $2n$ equations for coefficients of even powers. As we have checked numerically, these are only solvable by 0 when all equations are homogeneous. Thus 0 is a simple eigenvalue and we have a solution which is accurate to order $2m+2$.

$m = 2n+1$: We have $2n+2$ equations for $2n+2$ coefficients of odd powers and $2n+2$ homogeneous equations for $2n+1$ coefficients of even powers. Although the latter equations are overdetermined, they are always solved by 0. Numerically we have found that the equations for the coefficients of the odd powers is invertible; thus there is no second eigenvalue solving (75) but there is a generalized eigenvector solving (76). Thus we do not in general expect a solution which is accurate to order $2m+2$, but only $2m$ as proven above.

4 Numerical experiments in one-dimension

To verify the observations in the previous section, we present numerical experiments which assess the accuracy and performance of the Hermite-leapfrog method. To complement the experiments, we present comparisons with the classic Hermite method (Dual Hermite method as referred to in [15]). As a model equation we chose the following wave system

$$\begin{aligned} \frac{\partial p}{\partial t} &= -c^2(x)\rho\frac{\partial v}{\partial x} + z(x,t), \quad \rho\frac{\partial v}{\partial t} = -\frac{\partial p}{\partial x} \\ p(x,t_0) &= f(x), \quad v(x,t_0+\Delta t/2) = g(x), \end{aligned} \tag{77}$$

with periodic boundary conditions. Here $c(x)$ denotes the speed of the propagating wave and $z(x,t)$ is introduced to enforce a desired solution. For simplicity, ρ denotes a unit bulk modulus. Although we only consider a constant value for the bulk modulus, this restriction isn't necessary and the method could be applied to the case in which it is spatially varying. The time step, Δt is chosen by introducing a CFL constant $C_{CFL} \in (0,1)$, such that

$$\Delta t = 2C_{CFL}\frac{h}{c_{max}}.$$

Choosing C_{CFL} closer to one denotes a larger time step while choosing C_{CFL} closer to zero corresponds to taking a smaller time-step. The variable h corresponds to the length of the interpolation interval, and c_{max} denotes the maximum speed of the propagating wave.

4.1 Standing wave solution

As a first example we consider propagation of a standing wave with unit wave speed. The analytic solution is chosen to be

$$p(x,t) = \cos(2\pi t)\sin(2\pi x).$$

For these experiments the computational domain is defined to be the bi-unit interval and the solution is propagated to a final time of $T = 4.13$. Figure 2 reports the accuracy in the L^2 norm while Table 1 reports the observed rates of convergence. Numerically, we observe that even order Hermite-leapfrog schemes converge at rates of $O(h^{2m+2})$, as suggested by the analysis of the dispersion relation, while variation is observed if the method is of odd order. Furthermore, it can be observed that a Hermite-leapfrog scheme of even order provides a better approximation than the Dual Hermite method.

	$C_{CFL} = 0.1$				$C_{CFL} = 0.5$				$C_{CFL} = 0.9$			
Order - m	0	1	2	3	0	1	2	3	0	1	2	3
Hermite-leapfrog	1.90	1.96	5.97	5.65	1.94	1.91	5.98	7.09	2.01	2.02	6.00	5.87
Classic Hermite	-	2.58	4.90	6.84	0.10	2.78	4.83	6.95	0.46	2.78	4.83	6.88

Table 1: Observed L^2 rates of convergence for the Hermite-leapfrog scheme and classic Hermite methods of m^{th} order with varying CFL constants.

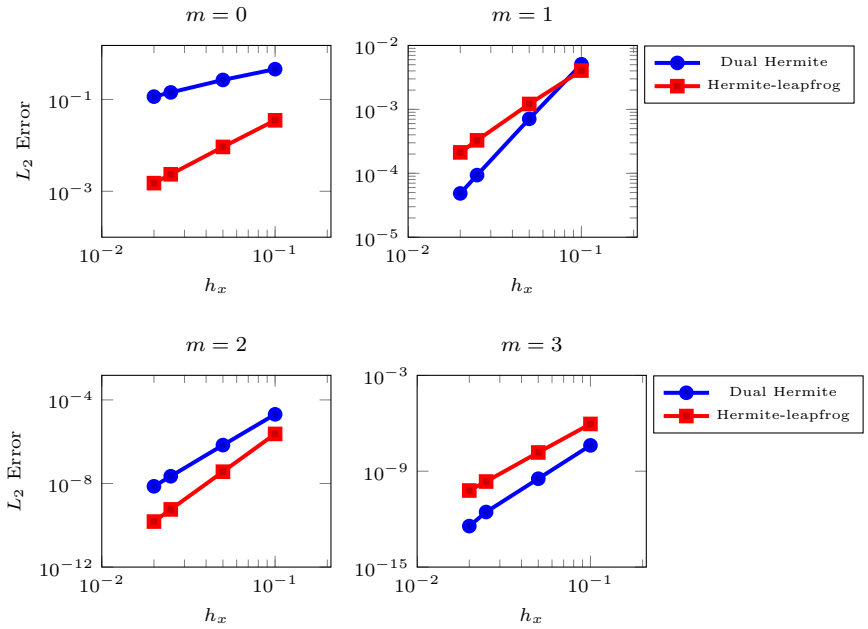


Fig. 2: L^2 errors of an m order Hermite-leapfrog and Dual Hermite methods when applied to the one-dimensional pressure-velocity system. The Hermite-leapfrog scheme provides a better approximation to the classic Hermite method when m is chosen to be even. For these experiments the CFL constant is set to $CFL = 0.9$ across all orders.

.

4.2 Spatially varying wave speed

As a second set of numerical experiments we consider a spatially varying wave speed. To simplify establishing the Cauchy-Kowalesky recurrence relation we take a pre-processing step and expand the coefficient at each grid point via a local Taylor expansion

$$c^2(x_i) = 1 + \sin(x_i)/2 \approx \sum_{i=0}^{2m+1} \tilde{c}_i \left(\frac{x - x_i}{h} \right),$$

where

$$\tilde{c}_i = \frac{h!}{i!} \frac{\partial^i c^2(x)}{\partial x^i}.$$

For these experiments the domain is chosen to be $[0, 2\pi]$ and the solution is propagated to a final time of $T = 3.2$. The analytic solution is chosen to be

$$p(x, t) = \sin(x - t). \tag{78}$$

Figure 3 reports the observed accuracy in the L^2 norm while Table 2 reports the observed rates of convergence. These numerical experiments suggest rates of $O(h^{2m+2})$ for even m and $O(h^{2m})$ for odd m.

4.2.1 Discontinuous coefficients

A key ingredient of the method is the Hermite interpolation step which acts as a smoothing operator. In the presence of discontinuous coefficients the interpolation step would reduce accuracy. A potential avenue for such problems could be to consider a hybrid scheme in which another method is used in areas with discontinuities. A hybrid Hermite discontinuous-Galerkin scheme has been presented in the literature [5,15] and is something we would like to explore in future work.

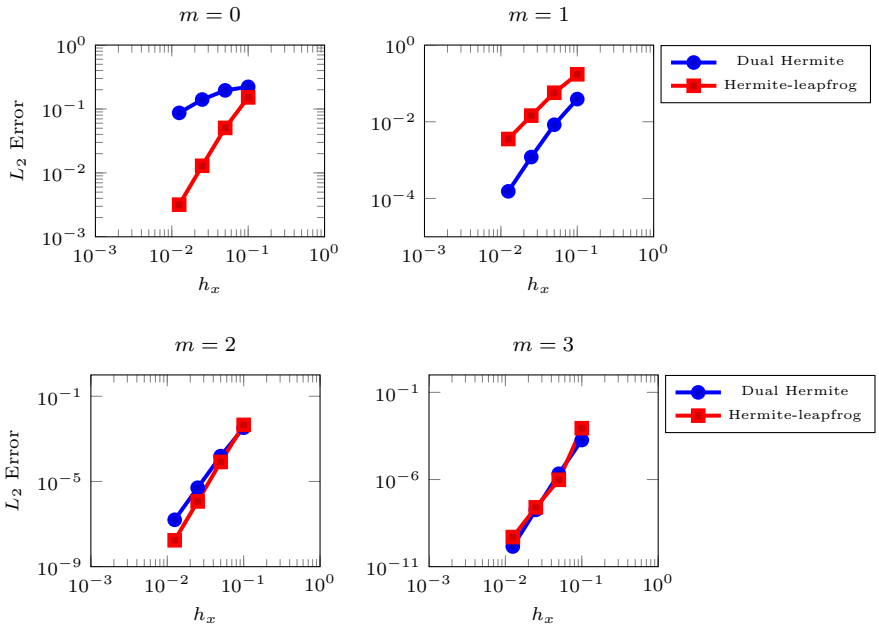


Fig. 3: The L^2 errors of various m order Hermite-leapfrog and Dual Hermite methods when applied to the one-dimensional pressure-velocity system with spatially varying wave speed. The Hermite-leapfrog scheme provides a better approximation than the Dual-Hermite method when m is chosen to be even. Here the CFL constant is set to $C_{CFL} = 0.9$.

.

	$C_{CFL} = 0.1$				$C_{CFL} = 0.5$				$C_{CFL} = 0.9$			
N	0	1	2	3	0	1	2	3	0	1	2	3
Hermite-leapfrog	1.98	1.97	5.98	5.94	1.98	1.97	5.99	5.88	1.98	1.98	6.02	5.81
Dual-Hermite	0.18	2.97	4.99	6.84	0.70	2.99	5.01	7.01	0.88	2.99	5.00	6.99

Table 2: Observed L^2 rates of convergence for various m order Hermite-leapfrog and Dual Hermite methods when applied to the pressure-velocity system with smoothly varying coefficients. The numerical experiments are carried out for various CFL constants.

4.3 A note on improving rates of convergence

The peculiar convergence behavior of the Hermite-leapfrog scheme has led us to investigate if modifications can be made which may yield more consistent convergence results. In particular we have found a modification to the method which leads to consistent $O(h^{2m+2})$ convergence rates. The derivation of the modified Hermite-leapfrog scheme originates from applying the scheme to the following advection equation,

$$u_t = u_x, \tag{79}$$

and tracing out local truncation errors. The analog of subtracting equations (5a) and (5b) for the advection equation yields a time-stepping scheme with local temporal truncation errors

$$\frac{\partial^r u\,(x,t+\Delta t)}{\partial x^r} = \frac{\partial^r u\,(x,t)}{\partial x^r} + \sum_{i=1+r,odd}^{2m+1} c_{i-r}\frac{\partial^i u\,(x,t+\Delta t/2)}{\partial x^i} + \mathcal{O}(\Delta t^{2m+3-r}), \tag{80}$$

$$\frac{\partial^l u\,(x,t+\Delta t)}{\partial x^l} = \frac{\partial^l u\,(x,t)}{\partial x^l} + \sum_{i=1+l,even}^{2m} c_{i-l}\frac{\partial^i u\,(x,t+\Delta t/2)}{\partial x^i} + \mathcal{O}(\Delta t^{2m+2-l}) \tag{81}$$

where it is important to observe that equation (80) is valid for even order derivatives, $r \in \{0,2,\dots\}$, and equation (81) is valid for odd order derivatives, $l \in \{1,3,\dots\}$. Here c_j are the coefficients from the Taylor series expansion. Drawing from the work of Appelö and co-authors in [1], we construct a time-stepping scheme by subtracting expansions (5a) and (5b) leading to

$$\frac{\partial^r u\,(x,t+\Delta t)}{\partial x^r} = -\frac{\partial^r u(x,t)}{\partial x^r} + \sum_{i=2+r}^{2m} c_{i-r}\frac{\partial^i u(x,t+\Delta t/2)}{\partial x^i} + \mathcal{O}(\Delta t^{2m+2-r}), \tag{82}$$

$$\frac{\partial^l u\,(x,t+\Delta t)}{\partial x^l} = -\frac{\partial^l u\,(x,t)}{\partial x^l} + \sum_{i=2+l}^{2m-1} c_{i-l}\frac{\partial^l u(x,t+\Delta t/2)}{\partial x^l} + \mathcal{O}(\Delta t^{2m+3-l}), \tag{83}$$

where the indices remain to be $r = 0,2,\dots$ and $l = 1,3,\dots$. As before, equation (82) is the local truncation error for even order derivatives, and (83) is the local truncation for odd order derivatives.

Spatial errors may be quantified by noting that Hermite-Birkhoff reconstruction leads to truncation errors of the form

$$\frac{\partial u^r(x,t)}{\partial x^r} = \frac{\partial \tilde{u}^r(x,t)}{\partial x^r} + \mathcal{O}(h^{2m+2-r}) \quad r = 0,2,4,\dots \tag{84}$$

$$\frac{\partial u^l(x,t)}{\partial x^l} = \frac{\partial \tilde{u}^l(x,t)}{\partial x^l} + \mathcal{O}(h^{2m+3-l}) \quad l = 1,3,5,\dots. \tag{85}$$

Here it is important to note that subsequent derivatives are approximated with the same order.

By using the time-stepping scheme given by equation (80) for even order derivatives and equation (83) for odd order derivatives we are choosing the best approximation within the framework. An immediate area of research would be to carry out a full truncation analysis, and a similar eigenvalue analysis to confirm $2m+2$ convergence rates. In the next section we present numerical evidence on stability and provide numerical estimates convergence rates for the modified Hermite-leapfrog method.

4.3.1 Modified Hermite-leapfrog: Advection Equation

To assess the accuracy and performance of the modified Hermite-leapfrog scheme, numerical experiments are carried out using the one-dimensional advection equation. For these experiments the domain is chosen to be the bi-unit interval and the analytic solution is chosen to be

$$u(x,t) = \sin(3\pi(x-t)).$$

Figure 4 reports the observed accuracy for orders $m = 1, 2, 3$ while Table 3 reports the observed rates of convergence, notably we observe consistent $O(h^{2m+2})$ rates of convergence.

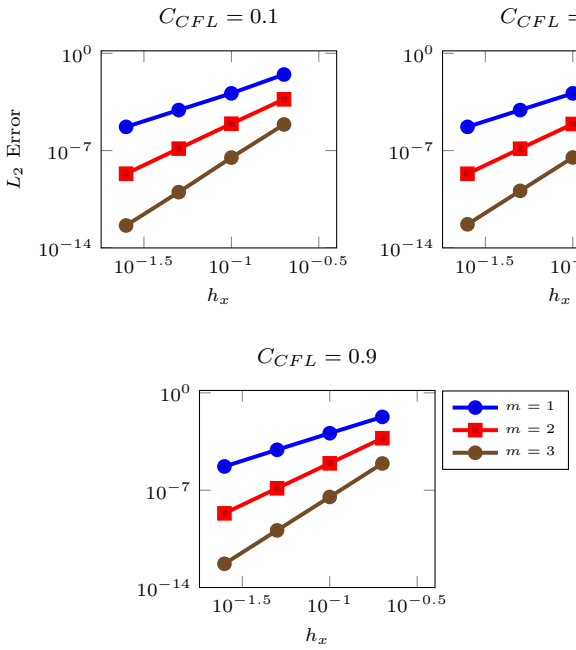


Fig. 4: The L^2 errors of various m order modified Hermite-leapfrog methods when applied to the one-dimensional advection equation with unit wave speed. The CFL constant is set to be $C_{CFL} = 0.9$ across all orders.

	$C_{CFL} = 0.1$			$C_{CFL} = 0.5$			$C_{CFL} = 0.9$		
Order - m	1	2	3	1	2	3	1	2	3
Modified Hermite-leapfrog	4.15	5.92	8.06	4.08	5.96	8.00	3.94	5.98	7.99

Table 3: Observed L^2 rates of convergence when using the modified Hermite-leapfrog scheme to solve the advection equation with unit wave speed.

4.3.2 Modified Hermite-leapfrog: Pressure-Velocity Equations

We continue our numerical experiments by applying the modified Hermite-leapfrog method to the pressure-velocity system. The drawback of this approach is that it requires to discretize the pressure and velocity on both the primary and dual grid. Approximations on both grids are necessary as the Cauchy-Kowalesky recurrence relation exchanges even time derivatives of the pressure term for spatial derivatives of the pressure term (odd time derivatives are exchanged for spatial derivatives of the velocity term). For these numerical experiments the analytic solution for the pressure term is chosen to be

$$p(x,t) = \cos(3\pi t)\sin(3\pi x).$$

The system is assumed to have unit wave speed and the solution is propagated to a final time of $T = 4.13$. Figure 5 reports the observed accuracy of the modified Hermite-leapfrog scheme under the L^2 norm. Table 4 reports the observed rates of convergence. The rates of convergence remain consistent with results observed with the advection equation, namely $O(h^{2m+2})$. Notably, the modified Hermite-leapfrog scheme is the most accurate scheme.

	$C_{CFL} = 0.1$			$C_{CFL} = 0.5$			$C_{CFL} = 0.9$		
Order - m	1	2	3	1	2	3	1	2	3
Modified Hermite-leapfrog	4.03	6.07	7.99	3.98	5.98	8.04	3.93	5.93	7.97
Hermite-leapfrog	1.04	5.95	5.07	1.07	5.95	5.64	1.21	5.98	6.50
Dual-Hermite	1.89	4.95	7.08	2.56	4.99	6.93	2.87	4.94	6.96

Table 4: Comparison of the L^2 rates of convergence for the modified Hermite-leapfrog, Hermite-leapfrog, and the Dual Hermite method when applied to the one-dimensional pressure-velocity system for orders $m = 1, 2, 3$.

5 Extending the method to higher dimensions

The Hermite-leapfrog method is easily extended to higher dimensions by means of a tensor product construction. For completeness we describe a two-dimensional extension; a three dimensional version follows a similar construction. To aid in the description of the method we consider the pressure-velocity wave system in two dimensions

$$\frac{\partial p}{\partial t} = -\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}, \tag{86}$$

$$\frac{\partial v}{\partial t} = -\frac{\partial p}{\partial x}, \quad \frac{\partial u}{\partial t} = -\frac{\partial p}{\partial y}. \tag{87}$$

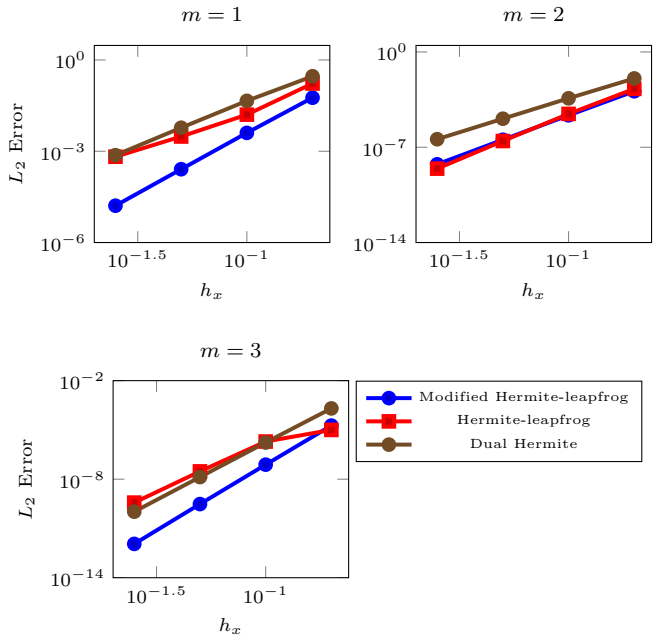


Fig. 5: L^2 errors for the modified Hermite-leapfrog, Hermite-leapfrog, and Dual Hermite scheme when applied to one-dimensional pressure velocity system.

.

For simplicity we assume the initial conditions are defined at a half-step apart

$$p(\mathbf{x}, t) = 0, \quad v(\mathbf{x}, \Delta t/2) = g(\mathbf{x}), \quad u(\mathbf{x}, \Delta t/2) = h(\mathbf{x}). \tag{88}$$

As in the one-dimensional version, the discretization of the equations occurs over two grids. The primary grid, Ω, serves to hold approximations of the pressure variable. The second dual grid $\tilde{\Omega}$ is introduced in order to store approximations of the velocity fields. The construction of each grid is performed by taking the tensor product of one-dimensional grids. Assuming a grid spacing of h_x and h_y in the x and y directions respectively, the solution at each node (x_j, y_j) is represented by the following tensor polynomial

$$u(\mathbf{x}) \approx u_{i,j}(\mathbf{x}) = \sum_{i=0}^{m} \sum_{j=0}^{m} \tilde{u}_{i,j} \left(\frac{x - x_i}{h_x} \right)^i \left(\frac{y - y_i}{h_y} \right)^j .$$

We emphasize that Hermite-leapfrog methods maintain approximations of the pressure solely on the primary grid. Approximations of the velocity variables are maintained on the dual grid; this is notably different than traditional staggered time domain finite difference methods which discretize variables on different grids. The time-stepping algorithm for the two-dimensional Hermite-leapfrog method is derived in a similar manner as in the one-dimensional case. Here we consider two-dimensional analogues of equations (5a) and (5b)

$$\frac{\partial^{s+l} p(x,y,t+\Delta t/2)}{\partial x^s \partial y^l} = \sum_{r=0} \frac{1}{r!} (\Delta t/2)^r \frac{\partial^{r+s+l} p(x,y,t)}{\partial t^r \partial x^s \partial y^l}, \tag{89}$$

$$\frac{\partial^{s+l} p(x,y,t-\Delta t/2)}{\partial x^s \partial y^l} = \sum_{r=0} \frac{1}{r!} (\Delta t/2)^r \frac{\partial^{r+s+l} p(x,y,t)}{\partial t^r \partial x^s \partial y^l}. \tag{90}$$

As before the update for the pressure term is derived by subtracting equations (89) and (90)

$$\begin{aligned} \frac{\partial^{s+l} p(x,y,t+\Delta t/2)}{\partial x^s \partial y^l} - \frac{\partial^{s+l} p(x,y,t-\Delta t/2)}{\partial x^s \partial y^l} = \\ - \sum_{r=1,odd} \frac{1}{r!} (\Delta t/2)^r \frac{\partial^{r+s+l} v_x(x,y,t)}{\partial x^{s+r} \partial y^{l+r}} \\ - \sum_{r=1,odd} \frac{1}{r!} (\Delta t/2)^r \frac{\partial^{r+s+l} v_y(x,y,t)}{\partial x^{s+r} \partial y^{l+r}}. \end{aligned} \tag{91}$$

where the indices s, l correspond to the order of spatial derivatives, i.e. $s, l \in \{0, \ldots, m\}$, the formulas for the velocity fields are derived analogously. As in the one-dimensional case; spatial derivatives are approximated by means of Hermite-Birkhoff interpolation. Given the tensor structure of the polynomial, the interpolant enables a dimension by dimension reconstruction. Figure 7 illustrates the staggering of the nodes as well as the reconstruction procedure; we refer the reader to [13] for further details the higher dimensional reconstruction procedure.

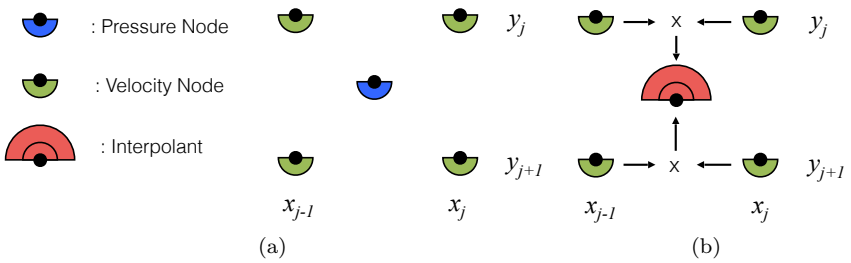


Fig. 6: Figure (a) illustrates an aerial view of the Hermite-leapfrog discretization for the pressure-velocity equations. We emphasize that the nodes are staggered a half-step apart. Figure (b) illustrates a dimension by dimension reconstruction of the velocity interpolant. The interpolants tensor product structure enables a dimension by dimension reconstruction.

.

5.1 Numerical experiments in two dimensions

To demonstrate the method's efficiency, we solve the acoustic wave equations with unit wave speed on the bi-unit square, $[-1, 1] \times [-1, 1]$, with periodic boundary

conditions. The pressure is chosen to have an analytic solution of

$$p(x, y, t) = \sin(\pi x)\sin(\pi y)\cos(\sqrt{2\pi}t).$$

Figure 7 reports observed accuracy in the L^2 norm for both the Hermite-leapfrog and Dual Hermite method. Table 5 reports observed rates of convergence with varying CFL constants, notably we observe similar convergence behavior as in the one-dimensional case.

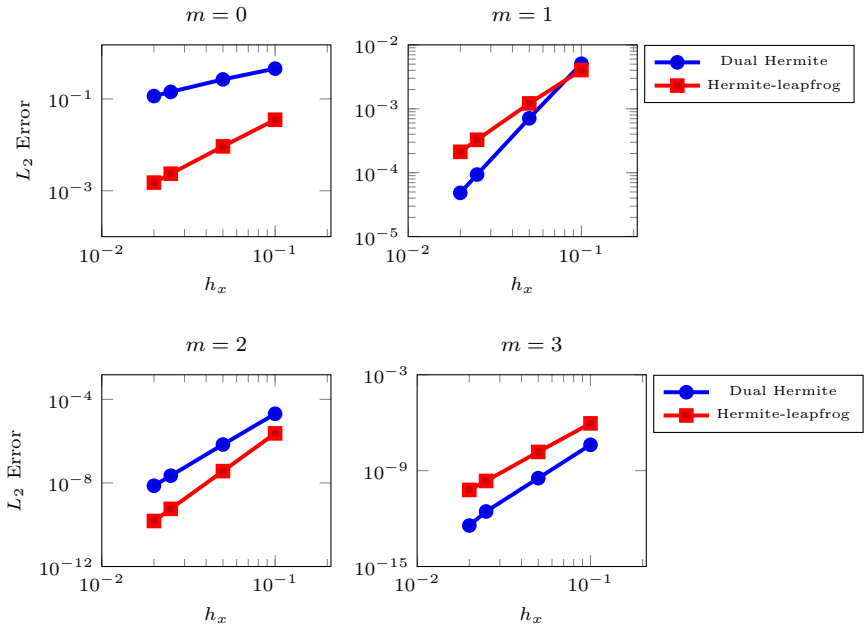


Fig. 7: L^2 errors for the modified Hermite-leapfrog, Hermite-leapfrog, and Dual Hermite scheme when applied to two-dimensional pressure velocity system.

.

	$C_{CFL} = 0.1$				$C_{CFL} = 0.5$				$C_{CFL} = 0.9$			
Order - m	0	1	2	3	0	1	2	3	0	1	2	3
Hermite-leapfrog	1.74	1.76	5.68	6.84	1.75	1.76	5.71	7.45	1.86	1.88	6.01	6.74
Dual-Hermite	-	2.37	4.76	6.83	-	2.44	4.78	6.76	-	2.67	4.80	6.77

Table 5: Comparison of L^2 rates of convergence for the m order Hermite-leapfrog, and the Dual Hermite method when applied to the one-dimensional pressure-velocity system.

5.2 Imposing reflective boundary conditions

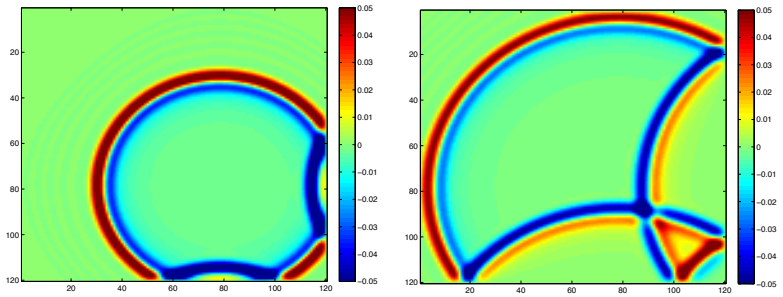
As a second example, we consider zero Dirichlet boundary conditions on the pressure variable. The complete set of boundary conditions may be derived by insisting the solution of the pressure term is zero at the boundary and differentiating in time; yielding

$$\begin{aligned} &p(x,y,t) = 0, \quad x = -1, 1, \quad y = -1, 1 \\ &u(x,y,t) = 0, \quad u_y(x,y,t) = 0, \quad y = -1, 1 \\ &v(x,y,t) = 0, \quad v_x(x,y,t) = 0, \quad x = -1, 1. \end{aligned} \tag{92}$$

The staggered discretization enables either the pressure or the velocity variables to be on boundary. For this example, we choose to discretize pressure in the interior of the domain and thus only the velocity is discretized on the boundary. The boundary conditions are imposed prior to propagating the pressure solution by filling in ghost point values of the velocity variables at the edge of the domain. The ghost point values are chosen to mirror the polynomials such that they are either odd around the boundary and thus agree with the homogeneous Dirichlet boundary conditions or so that they are even and satisfy homogeneous Neumann conditions. To illustrate reflecting boundaries we initialize the pressure variable with the following Gaussian pulse

$$p(x,y,0) = \exp\left(\frac{(x-0.3)^2 + (y-0.3)^2}{0.002}\right).$$

Figure 8 illustrates a reflection of the propagating wave on the South-East corner.



(a) (b)

Fig. 8: Figure (a) and (b) illustrate a wave reflected on the South-East corner as propagated by the Hermite-leapfrog scheme. Imposing reflective boundary conditions is accomplished by adding ghost nodes on the dual grid. The ghost points values are chosen to mirror the polynomials such that they are even or odd around the boundary and thus agreeing with the boundary conditions specified by the boundary conditions (equations (92)) .

5.3 Numerical experiments with Maxwell's equations

To illustrate our methods ability to resolve high frequency waves, we carry out numerical experiments with Maxwell's equation in the time domain. In particular we consider the transverse magnetic form with unit material coefficients

$$\begin{aligned} \frac{\partial H^x}{\partial t} &= -\frac{\partial E^z}{\partial y}, \\ \frac{\partial H^y}{\partial t} &= \frac{\partial E^z}{\partial x}, \\ \frac{\partial E^z}{\partial t} &= \frac{\partial H^y}{\partial x} - \frac{\partial H^x}{\partial y}. \end{aligned} \tag{93}$$

Here (H^x, H^y) corresponds to the magnetic fields, and E^z is the electric field. We take our domain of interest to be the bi-unit square, $\Omega = [-1, 1] \times [-1, 1]$, and the analytic solution to be

$$\begin{aligned} H^x(x, y, t) &= -\frac{\omega_y}{\omega_t} \sin(\omega_x x) \cos(\omega_y y) \sin(\omega_t t), \\ H^y(x, y, t) &= \frac{\omega_x}{\omega_t} \cos(\omega_x x) \sin(\omega_y y) \sin(\omega_t t), \\ E^z(x, y, t) &= \sin(\omega_x x) \cos(\omega_y y) \sin(\omega_t t). \end{aligned} \tag{94}$$

The parameters are chosen to be $\omega_x = 8\pi, \omega_y = 8\pi$, and $\omega_t = \sqrt{\omega_x^2 + \omega_y^2}$. The boundary conditions are imposed by the mirroring technique discussed in 5.2. To illustrate the resolving power of a high order numerical method, we choose the polynomial degree of the Hermite-leapfrog method to be $m = 3, 4, 5$, and 6. Figure 9 reports the error under the L_2 norm. At high orders we observe slightly more variation of rates of convergence and sensitivity to round-off, yet the method is still numerically stable. At high orders it becomes difficult to accurately estimate rates of convergence due to numerical round off and how fast the method converges. In practice, the accuracy of the method will depend on the frequency of the solution and the order of the method. As future work we are investigating techniques to improve accuracy at high order.

6 Acceleration on graphics processing units

Lastly, we demonstrate that the Hermite-Leapfrog method is well-suited for the graphics processing unit. In these numerical experiments we solve the three-dimensional acoustic wave equations

$$\begin{aligned} \frac{\partial p}{\partial t} &= -\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} - \frac{\partial w}{\partial z} \\ \frac{\partial v}{\partial t} = -\frac{\partial p}{\partial x}, \quad \frac{\partial u}{\partial t} &= -\frac{\partial p}{\partial y}, \quad \frac{\partial w}{\partial t} = \frac{\partial p}{\partial z}. \end{aligned} \tag{95}$$

on the unit cube, $[-1, 1] \times [-1, 1]$ with periodic boundary conditions. This section serves as an extension of the work carried out by Vargas et at. [14] wherein Hermite methods were tailored to the graphics processing unit. More precisely we revisit the two approaches outlined in [14]. The first is a split approach in two kernels are

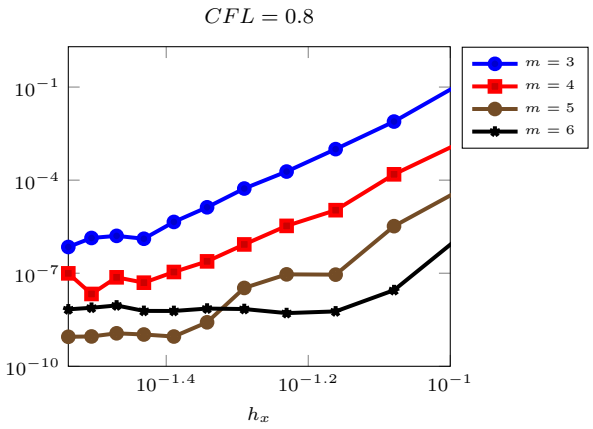


Fig. 9: L^2 errors for the Hermite-leapfrog method applied to Maxwell's equation.

employed. The first kernel reconstructions approximations of high order derivatives via Taylor series while the second kernel then applies leapfrog time-stepping to propagate the solution. The second approach combines the interpolation and evolution procedures into a monolithic kernel.

Our numerical experiments are carried out on a single node of the Southern Methodist University cluster "ManeFrame II". A single node consist of a NVIDIA Tesla P100-PCIE-16GB graphics card. The hardware has a theoretical bandwidth of 732 GB/s and potentially performs 4700 Gigaflops/sec in double precision. To obtain estimates of kernel bandwidth and gflops we employ the NVIDIA profiler which reports effective arithmetic bandwidth by flop_count_dp and memory bandwidth as the sum of *dram_read_throughput* and *dram_write_throughput*. Here bandwidth corresponds to the sum of bytes read and written to global memory by a GPU kernel. Our CUDA kernels are generated by using the OCCA language [12]. Lastly we compare time to solution with a dual socket Xeon CPU E5-2695 (2.10 GHz) as found on a Maneframe II node.

In studying GFLOP and effective bandwidth it is important to note that the devices theoretical capabilities are typically difficult to achieve in practice. To estimate a more "realistic" bandwidth we consider a simple vector copy wherein the entries of vector x are copied onto vector y. For sufficiently large N a streaming bandwidth of 534 GB/s was observed on the graphics card. Although this is not indicative of the peak performance it does serve as a representative of achievable peak performance numbers. Tables 10 and Tables 11 report the observed GFLOP and bandwidth performance.

In our numerical experiments it is clear that the interpolation procedure dominates the floating point operations. This is not too surprising as the operations are carried out as a series of matrix-matrix multiplications (tensor contractions). Similarly the kernel which evolves the velocity variables, V_x, V_y, V_z, has a high number of floating point operations as it carried out the time-stepping for three variables.

As OCCA enables code generation on various platforms we conclude our accelerator studies by presenting Table 6 which compares time per iteration across

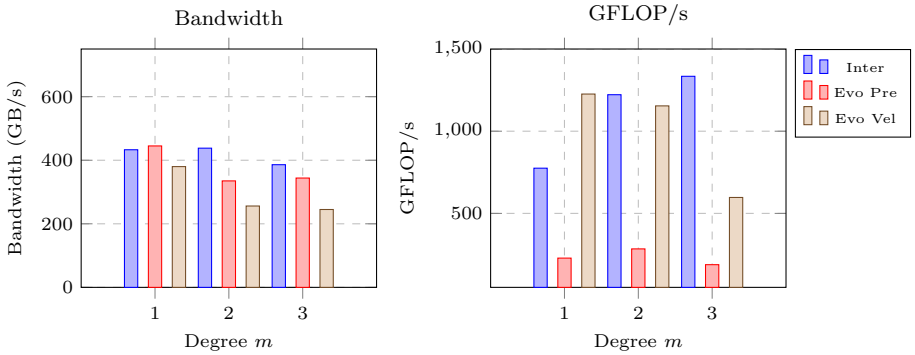


Fig. 10: Here we report the Bandwidth and GFLOP of a two kernel approach as outlined by Vargas and co-authors in [14].

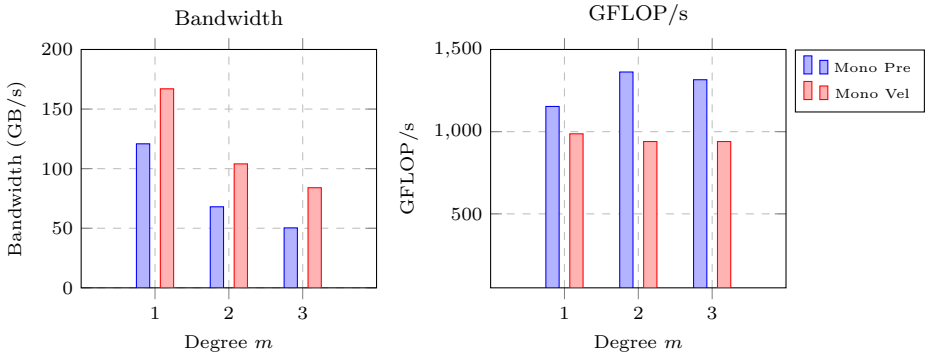


Fig. 11: Here we report the Bandwidth and GFLOP of using a monolithic kernel approach as outline by Vargas and co-authors in [14].

a P100 and the dual 16-core Xeon CPUs. In these experiments an iteration refers to propagating the pressure and velocity fields.

Order of the method - m	$m = 1$	$m = 2$	$m = 3$
OCCA::OpenMP	1.08 sec	1.30 sec	1.74 sec
OCCA::CUDA - Single Kernel	0.060 sec	0.07 sec	0.10 sec
OCCA::CUDA - Two Kernels	0.098 sec	0.19 sec	0.15 sec

Table 6: Comparison of time per iteration of Hermite kernels executed on the GPU and CPU. For orders $m = 1, 2, 3$ the number of grid points were chosen to be 200, 130, 100 points per Cartesian direction. Noticeably a single GPU kernel offers a better time to solution with advantage of relieving the need to store the interpolate on account of less data movement. These results are consistent with what has been observed in [14].

7 Summary

We have presented a variation of the classic Hermite method which uses leapfrog time-stepping to advance the solution for linear wave systems. The new Hermite-leapfrog method is demonstrated to be numerically stable, high-order in both time and space, and may time-step the solution independent of order. A detailed description of the method is provided in one and two dimensions, as well as a techniques for incorporating spatially varying coefficients and reflective boundary conditions. Numerical experiments suggest that the rate of convergence is dependent on whether the method is of odd or even order. To address the variation we introduce a modification that achieves consistent $O(h^{2m+2})$ rates of convergence.

Lastly, we accelerate the method using a graphics processing unit. We find that a monolithic kernel is ideal as it eliminates the need to explicitly store the interpolant and can provide a comparable time to solution to using two kernels which achieve a higher bandwidth. As a future research direction we plan to provide a theoretical justification for the exceptional convergence rates of the modified Hermite-leapfrog method.

Acknowledgements The authors would like to thank Daniel Appelö for the fruitful conversations. TH is supported in part by NSF Grant DMS-1418871. Any conclusions or recommendations expressed in this paper are those of the author and do not necessarily reflect the views of the NSF.J C is supported by NSF grants DMS-1719818 and DMS-1712639. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-JRNL-757049.

References

1. Appelo, D., Hagstrom, T., Vargas, A.: Hermite methods for the scalar wave equation. SIAM J. Sci. Comput. **40**, A3902–A3927 (2018)
2. Appelö, D., Inkman, M., Hagstrom, T., Colonius, T.: Hermite methods for aeroacoustics: Recent progress. In: 17th AIAA/CEAS Aeroacoustics Conference (32nd AIAA Aeroacoustics Conference), Portland, Oregon (2011)
3. Bencomo, M.J.: Discontinuous Galerkin and finite difference methods for the acoustic equations with smooth coefficients. Ph.D. thesis, Rice University (2015)
4. Chen, R., Hagstrom, T.: P-adaptive Hermite methods for initial value problems. ESAIM: Mathematical Modelling and Numerical Analysis **46**(3), 545–557 (2012)
5. Chen, X.R., Appelö, D., Hagstrom, T.: A hybrid Hermite–discontinuous Galerkin method for hyperbolic systems with application to Maxwells equations. Journal of Computational Physics **257**, 501–520 (2014)
6. Gauthier, O., Virieux, J., Tarantola, A.: Two-dimensional nonlinear inversion of seismic waveforms: Numerical results. Geophysics **51**(7), 1387–1403 (1986)
7. Goodrich, J., Hagstrom, T., Lorenz, J.: Hermite methods for hyperbolic initial-boundary value problems. Mathematics of computation **75**(254), 595–630 (2006)
8. Kornelus, A., Appelö, D.: Flux-conservative hermite methods for simulation of nonlinear conservation laws. Journal of Scientific Computing **76**(1), 24–47 (2018)
9. Kowalczyk, K., Van Walstijn, M.: A comparison of nonstaggered compact FDTD schemes for the 3D wave equation. In: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pp. 197–200. IEEE (2010)
10. Levander, A.R.: Fourth-order finite-difference P-SV seismograms. Geophysics **53**(11), 1425–1436 (1988)
11. LeVeque, R.J.: Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems. SIAM (2007)
12. Medina, D.: OKL: A unified language for parallel architectures. Ph.D. thesis, Rice University (2015)

13. Vargas, A.: Hermite methods for the simulation of wave propagation. Ph.D. thesis (2017)
14. Vargas, A., Chan, J., Hagstrom, T., Warburton, T.: Gpu acceleration of hermite methods for the simulation of wave propagation. In: Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016, pp. 357–368. Springer (2017)
15. Vargas, A., Chan, J., Hagstrom, T., Warburton, T.: Variations on Hermite methods for wave propagation. Communications in Computational Physics **22**(2), 303–337 (2017)
16. Xie, Z., Chan, C.H., Zhang, B.: An explicit fourth-order staggered finite-difference time-domain method for maxwell's equations. Journal of Computational and Applied Mathematics **147**(1), 75–98 (2002)
17. Yee, K.: Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Transactions on antennas and propagation **14**(3), 302–307 (1966)
18. Yefet, A., Petropoulos, P.G.: A staggered fourth-order accurate explicit finite difference scheme for the time-domain maxwell's equations. Journal of Computational Physics **168**(2), 286–315 (2001)