Commun. Comput. Phys.
doi: 10.4208/cicp.260915.281116a

Vol. **22**, No. 2, pp. 303-337
August 2017

Variations on Hermite Methods for Wave Propagation

Arturo Vargas[1,*], Jesse Chan[2], Thomas Hagstrom[3] and Timothy Warburton[2]

[1] *Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005, USA.*
[2] *Department of Mathematics, Virginia Tech, Blacksburg, VA 24061, USA.*
[3] *Department of Mathematics, Southern Methodist University, Dallas, TX 75275, USA.*

Communicated by Chi-Wang Shu

Received 26 September 2015; Accepted (in revised version) 28 November 2016

Abstract. Hermite methods, as introduced by Goodrich et al. in [15], combine Hermite interpolation and staggered (dual) grids to produce stable high order accurate schemes for the solution of hyperbolic PDEs. We introduce three variations of this Hermite method which do not involve time evolution on dual grids. Computational evidence is presented regarding stability, high order convergence, and dispersion/dissipation properties for each new method. Hermite methods may also be coupled to discontinuous Galerkin (DG) methods for additional geometric flexibility [4]. An example illustrates the simplification of this coupling for Hermite methods.

AMS subject classifications: 65M70

Key words: High-order methods, hyperbolic problems, numerical analysis.

1 Introduction

The computational simulation of wave propagation is central to geophysical applications, such as seismic imaging and exploration, the modeling of seismic waves induced by earthquakes, and problems in structural acoustics. However, the numerical modeling of intermediate frequency waves is known to be challenging for many standard low-order methods, requiring a large number of points per wavelength to adequately resolve oscillatory behavior. Additionally, the simulation of propagating waves using low order methods is typically subject to significant non-physical (numerical) dissipation and dispersion. High order methods have the advantage of both rapid convergence

*Corresponding author. *Email addresses:* Arturo.Vargas@rice.edu (A. Vargas), jesse.chan@caam.rice.edu (J. Chan), thagstrom@mail.smu.edu (T. Hagstrom), tcew@vt.edu (T. Warburton)

 ©2017 Global-Science Press

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

and decreased numerical dissipation [10, 19] compared to low order methods, especially for problems in intermediate frequency wave propagation [14, 25]. High order methods also tend to have a high number of operations per data access, yielding a computational structure well-suited to modern computing architectures [21, 23, 24].

Hermite methods, as introduced by Goodrich et al. in [15], are high order methods for wave propagation which represent the solution using a piecewise polynomial basis by collocating the solution and its derivatives on a structured grid. Solution and derivative information at grid nodes is then used to reconstruct and evolve the solution in time on a staggered (dual) grid. Hermite methods are provably stable and high order accurate for hyperbolic equations, including problems with varying coefficients.

Furthermore, though the reconstruction step requires the access of non-local data at neighbor nodes, the computation of derivatives then depends only locally on the reconstructed data at each node. This is advantageous for high order or multi-stage timestepping methods compared to finite difference methods, where neighboring data must be accessed each time derivatives are approximated. This structure has also been noted to be well-suited for parallel implementations on modern architectures [2, 11]. Hermite schemes, which were initially introduced for Cartesian domains, have also been coupled with discontinuous Galerkin (DG) schemes for numerical simulations on complex geometries [4]. They have also been applied to problems in aeroacoustics [2], electromagnetics [4], and fluid dynamics [18].

The Hermite schemes of Goodrich et al. [15] are one instance of a broader family of methods involving collocation of the solution and its derivatives. Other methods in this family include shape-preserving methods [13, 28, 36] and jet schemes [5, 26, 33], which use Hermite interpolation in conjunction with semi-Lagrangian techniques to solve advective problems. These differ from the Hermite schemes discussed here in terms of the characteristic time evolution procedure; however, the analysis and stability of both Hermite and jet schemes both rely primarily on properties of Hermite interpolation under high order Sobolev seminorms.

Sections 1.1 and 2 present a generalized view of Hermite methods, and motivate new one-step Hermite schemes based on variations in the reconstruction procedure. These procedures also aim to simplify the implementation and coupling of Hermite and DG schemes [4]. Section 3 presents numerical experiments which confirm the high order convergence and stability of each method for the advection equation in one dimension. Section 4 extends each method to two space dimensions and includes numerical results for the two-dimensional advection and acoustic wave equations.

1.1 Time evolution

In this section, we introduce one-dimensional Hermite schemes for the approximation of an evolving solution and its derivatives at a collection of points over an interval $[a,b)\subset\mathbb{R}$. Each Hermite scheme presented has a timestep restriction based only on the domain of dependence for hyperbolic partial differential equations. For simplicity of presentation,

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

we illustrate this using the 1D periodic scalar advection equation

$$\begin{aligned}&\frac{\partial u}{\partial t}=c\frac{\partial u}{\partial x},\\&u(x,t_0)=\mathbf{u}_0(x),\\&u(a,t)=u(b,t),\end{aligned}$$

where c is a constant advection speed and $u_0(x)$ is a smooth initial condition. Hermite methods may be extended in a straightforward manner to non-uniform grids and more general systems of equations with variable coefficients [15], though these details are omitted for brevity.

We define first a primary grid Ω as a collection of K equispaced points

$$\Omega=\{x_m:\ x_m=a+mh_x,\ m=0,\cdots,K-1\},$$

where K is the number of grid points on the interval (a,b) and $h_x=(b-a)/K$ denotes the spacing between the nodes. For periodic domains, we assume that $x_{m+K}=x_m$.

Next, we introduce the interpolation length scale h (distinct from the grid spacing h_x). We assume that a smooth function $u(x)$ is well-approximated over some interval (L_m,R_m) with size $h=R_m-L_m$ by a degree $\tilde{N}$ expansion around some point $\tilde{x}_m$. This expansion takes the form

$$u(x)\approx\tilde{u}_m(x)=\sum_{j=0}^{\tilde{N}}\tilde{\mathbf{u}}_j\left(\frac{x-\tilde{x}_m}{h_x}\right)^j.$$

The vector $\mathbf{u}$ contains Hermite degrees of freedom, which are scaled spatial derivatives at $\tilde{x}_m$

$$\tilde{\mathbf{u}}_j=\frac{h_x^j}{j!}\frac{d^j\tilde{u}_m(x)}{dx^j}\bigg|_{\tilde{x}_m},\quad j=0,\cdots,\tilde{N}.$$

For convenience, we express the advection operator applied to $u_m(x)$ as an expansion around x_m

$$c\frac{\partial\tilde{u}_m(x)}{\partial x}=\sum_{j=0}^{\tilde{N}}\mathbf{w}_j\left(\frac{x-\tilde{x}_m}{h_x}\right)^j,\qquad\mathbf{w}_j=\frac{h_x^j}{j!}\frac{\partial^j}{\partial x^j}\left(a\frac{\partial\tilde{u}_m(x)}{\partial x}\right)\bigg|_{\tilde{x}_m}.$$

The coefficients $\mathbf{w}_j$ are related to $\mathbf{u}_j$ through the derivative matrix $\mathbf{D}$

$$\mathbf{w}=c\mathbf{D}\mathbf{u},\qquad\mathbf{D}_{ij}=\begin{cases}\frac{i+1}{h_x}, & j=i+1,\\0, & \text{otherwise},\end{cases}\qquad 0\le i,j\le\tilde{N}.$$

This yields a semi-discrete system for the degrees of freedom $\mathbf{u}_j$

$$\frac{\mathrm{d}\tilde{\mathbf{u}}(t)}{\mathrm{d}t}=-c\mathbf{D}\tilde{\mathbf{u}}(t).$$

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

For equations with constant coefficients we approximate $u(x,t_n+dt)$ for some timestep $dt>0$ by solving this semi-discrete system. This is achieved in [15] using a temporal Taylor series: assuming an expansion centered around time t_n,

$$\tilde{u}_m(x,t)=\sum_{j=0}^{\tilde{N}}\sum_{k=0}^{\tilde{N}-j}\tilde{\mathbf{U}}_{jk}\left(\frac{x-\tilde{x}_m}{h_x}\right)^j\left(\frac{t-t_n}{dt}\right)^k,$$

and using the Cauchy-Kowalevski relation, the coefficients $\mathbf{U}_{jk}$ may be shown to satisfy

$$\tilde{\mathbf{U}}_{jk}=\frac{c}{kdt}\frac{(j+1)}{h_x}\tilde{\mathbf{U}}_{j+1,k-1},\qquad j=0,\cdots,\tilde{N}-k.$$

or more succinctly using matrix-vector notation

$$\tilde{\mathbf{U}}_{(\cdot,k)}=\frac{c}{kdt}\mathbf{D}\tilde{\mathbf{U}}_{(\cdot,k-1)},$$

where $\tilde{\mathbf{U}}_{(\cdot,k)}$ refers to the kth column of the array $\tilde{\mathbf{U}}$. Inserting these coefficients into the temporal Taylor expansion and evaluating at time t_n+dt yields an update for the solution $u_m(x,t_n)$. In practice, this is computed using Algorithm 1 [3, p109], which may be generalized to linear autonomous systems in multiple dimensions. Alternatively, more standard time integration techniques may be used to advance the solution forward in time. This evolution of the solution from t_n to $t_{n+1}=t_n+dt$ may be represented by the application of some update matrix $\mathbf{T}^{dt}$ to the degrees of freedom at time t_n, such that

$$\tilde{\mathbf{u}}^{n+1}=\mathbf{T}^{dt}\tilde{\mathbf{u}}^n,\qquad \tilde{u}_m(x,t_n)=\sum_{j=0}^{\tilde{N}}\tilde{\mathbf{u}}_j^n\left(\frac{x-\tilde{x}_m}{h_x}\right)^j.$$

Algorithm 1 Time evolution procedure for 1D scalar advection.

procedure TEMPORAL TAYLOR SERIES EVALUATION
  $\mathbf{w}=\tilde{\mathbf{u}}^n$
  for $\ell=\tilde{N},\tilde{N}-1,\cdots,0$ **do**
    $\mathbf{w}=\mathbf{w}+\frac{dt}{1+\ell}(-c\mathbf{D})\tilde{\mathbf{u}}^n$
  $\tilde{\mathbf{u}}^{n+1}=\mathbf{w}$

As noted in [15], for the scalar advection equation, the Taylor expansion in time yields an exact evolution of the approximation to $\mathbf{u}_0$, so long as the domain of dependence at t_0+dt lies within the interval (L_m,R_m). This translates into a degree-independent timestep restriction

$$cdt<\frac{h}{2},\qquad h=\min\{\tilde{x}_m-L_m,R_m-\tilde{x}_m\}.$$

If L_m and R_m are chosen symmetrically around x_m, h the interpolation interval reduces to $(L_m,R_m)=(x_m-h,x_m+h)$.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

The same domain of dependence argument motivates timestep restrictions for "tent-pitching" space-time finite element methods, which use a coupled discretization in both time and space. The stability of the space-time formulation results in a similar causal timestep restriction [12].

1.2 Local Hermite interpolation

The above evolution procedure hinges on a degree $\tilde{N}$ polynomial representation of some smooth function $u(x)$ in an interval (L_m,R_m) which accurately approximates the solution and its derivatives at the point $\tilde{x}_m$. This is addressed in [15] using degree N Hermite interpolation to produce a degree $\tilde{N}=2N+1$ reconstruction at specific points $\tilde{x}_m$.

The degree $\tilde{N}$ Hermite interpolant of $u(x)$ over (L_m,R_m) (which we refer to as $\tilde{u}_m(x)$) is constructed by specifying $(N+1)$ scaled spatial derivatives at the left and right endpoints of the interval (L_m,R_m)

$$\mathbf{u}_j^L=\frac{h_x^j}{j!}\frac{d^ju}{dx^j}\bigg|_{L_m},\qquad \mathbf{u}_j^R=\frac{h_x^j}{j!}\frac{d^ju}{dx^j}\bigg|_{R_m},\qquad j=0,\cdots,N.$$

The resulting expansions around L_m,R_m

$$u_m^L(x)=\sum_{j=0}^{N}\mathbf{u}_j^L\left(\frac{x-L_m}{h_x}\right)^j,\qquad u_m^R(x)=\sum_{j=0}^{N}\mathbf{u}_j^R\left(\frac{x-R_m}{h_x}\right)^j$$

coincide with the value and first N derivatives of $u(x)$ at each point.

The Hermite interpolant $\tilde{u}_m(x)$ is a polynomial of order $2N+1$ over (L_m,R_m), which is defined by interpolating the $(N+1)$ solution and derivative values at each endpoint. $\tilde{u}_m(x)$ is represented using the following expansion

$$\tilde{u}_m(x)=\sum_{j=0}^{2N+1}\tilde{\mathbf{u}}_j\left(\frac{x-\tilde{x}_m}{h_x}\right)^j,$$

where $\tilde{\mathbf{u}}_j$ are values of the solution and $2N+1$ scaled derivative values at $\tilde{x}_m$. These coefficients may be determined by solving the interpolation problem

$$\frac{\partial^i\tilde{u}_m}{\partial x^i}\bigg|_{L_m}=\frac{\partial^iu_m^L}{\partial x^i}\bigg|_{L_m},\qquad \frac{\partial^i\tilde{u}_m}{\partial x^i}\bigg|_{R_m}=\frac{\partial^iu_m^R}{\partial x^i}\bigg|_{R_m},\qquad i=0,\cdots,2N+1.$$

This results in the system

$$\begin{bmatrix}\mathbf{C}^L\\ \mathbf{C}^R\end{bmatrix}\tilde{\mathbf{u}}=\begin{bmatrix}\mathbf{u}^L\\ \mathbf{u}^R\end{bmatrix},\tag{1.1}$$

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

where the constraint matrices $\mathbf{C}^L,\mathbf{C}^R\in\mathbb{R}^{(N+1)\times(2N+2)}$ enforce conditions on $u_m(x)$ and its $2N+1$ derivatives at the points L_m,R_m

$$\mathbf{C}^L_{mn}=\begin{cases}\left(\frac{L_m-\tilde{x}_m}{h_x}\right)^{n-m}\frac{1}{m!}\prod_{s=0}^{m-1}(n-s), & n\geq m,\\ 0, & n<m,\end{cases}$$

$$\mathbf{C}^R_{mn}=\begin{cases}\left(\frac{R_m-\tilde{x}_m}{h_x}\right)^{n-m}\frac{1}{m!}\prod_{s=0}^{m-1}(n-s), & n\geq m,\\ 0, & n<m.\end{cases}$$

For convenience, we define $\mathbf{H}$

$$\mathbf{H}=\begin{bmatrix}\mathbf{C}^L\\ \mathbf{C}^R\end{bmatrix}^{-1}$$

as the interpolation matrix which maps solution and scaled derivative data at the endpoints of the interval (L_m,R_m) to a degree $2N+1$ expansion at the node $\tilde{x}_m$. Variations in the construction of $\mathbf{H}$ result in different interpolation/reconstruction schemes.

For the remainder of this work, we refer to the expansion of $\tilde{u}_m(x)$ as a Hermite reconstruction at $\tilde{x}_m$. This expansion may then be evolved in time a distance of dt using the evolution matrix $\mathbf{T}^{dt}$.

2 Hermite methods in one dimension

Having introduced evolution and interpolation procedures, one-dimensional Hermite methods may now be specified. Given some grid Ω, Hermite approximation spaces are associated with each grid. The degree N space $\mathcal{P}^N_\Omega$ is defined to be

$$\mathcal{P}^N_\Omega=\left\{\bigoplus_{m=0}^{K-1}\mathcal{P}^{2N+1}[x_m,x_{m+1})\right\}\cap C^N[a,b].$$

which consists of piecewise polynomials of degree $2N+1$ with N globally continuous derivatives at each node $x_m\in\Omega$. Each $u(x)\in\mathcal{P}^N_\Omega$ is defined by $(N+1)$ pieces of Hermite interpolation data at each node x_m

$$\mathbf{U}_{mj}=\frac{h_x^j}{j!}\frac{\partial^j u}{\partial x^j}\bigg|_{x_m},\qquad j=0,\cdots,N+1,\qquad m=0,\cdots,K-1.$$

For brevity, we refer to $\mathbf{U}_{(m,\cdot)}$, the vector of derivatives at the node x_m, as $\mathbf{U}_m$.

Finally, in addition to the evolution and interpolation operators $\mathbf{T}^{dt}$ and $\mathbf{H}$, respectively, we introduce the restriction operator $\mathbf{R}\in\mathbb{R}^{(N+1)\times(2N+1)}$ such that $\mathbf{R}_{ij}=\delta_{ij}$ and multiplication of a vector by $\mathbf{R}$ extracts the first $N+1$ entries of that vector.

Hermite methods march forward in time by combining three procedures over one or more stages:

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

1. **Interpolation** using points in the primary grid to produce Hermite reconstructions of higher degree $\tilde{N}$ centered around points $\tilde{x}_m$.

2. **Evolution** of the higher degree Hermite reconstructions at points $\tilde{x}_m$ forward in time to $\tilde{t}=t_n+dt$.

3. **Restriction** of the solution by truncating the degree of the polynomial expansion at $\tilde{x}_m$.

We emphasize that for each Hermite method, the stable timestep restriction is independent of the degree of approximation N. This is due to the fact that, by evolving the solution in time using a temporal Taylor series, a single update step of a Hermite method may be interpreted as the composition of the *exact* evolution of piecewise polynomial data with a *projection* in a seminorm which is preserved by the solution [15]. In particular, Hermite interpolation in one space dimension results in the projection onto piecewise degree-$2N+1$ polynomial in the H^{N+1} seminorm. Thus, as described in Section 1.1, the timestep restriction is determined only by the domain of dependence of the equation and the interval of the Hermite reconstruction.

We refer to the collection of reconstruction points $\tilde{x}_m$ as an auxiliary grid $\tilde{\Omega}$ on which the solution is evolved in time. Different schemes use differing combinations of interpolation and evolution procedures, which are summarized in Table 1. Different Hermite schemes may also vary parameters of the interpolation process. For example, the Hermite schemes of Goodrich et al. [15] (referred to henceforth in this paper as Dual Hermite schemes) produce a Hermite reconstruction centered between two grid points, which are represented over an auxiliary grid consisting of midpoints of the primary grid. These reconstructions are then evolved forward in time and truncated. The resulting auxiliary grid data may then be used to compute Hermite reconstructions at the original primal points, which are then evolved in time and truncated to complete a single timestep.

We introduce here the Virtual Hermite method, which is equivalent to the Dual Hermite method with a timestep of size zero on the auxiliary grid. As a result, the interpolations to and from the auxiliary grid may be combined, resulting in a step which interpolates and reconstructs on the same primary grid.

The Central and Upwind Hermite methods aim to interpolate and reconstruct on the same primary grid through redefinitions of the interpolation operator. The Central scheme expands the interval of interpolation, producing a Hermite reconstruction at a point x_m using information at neighboring nodes x_{m-1},x_{m+1}, while the Upwind Hermite scheme uses data from a single neighbor to produce a directional reconstruction.

While the implementation of the time evolution operator changes from problem to problem, the interpolation procedure remains the same irregardless of the equation solved. For this reason, we focus on the description of the interpolation procedure for three specific Hermite methods in the following sections. Common to each Hermite scheme is a timestep restriction which is independent of the polynomial degree N.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

Table 1: Overview of interpolation and evolution operations over different stages for various Hermite method, where $\mathbf{U}_m^n$ refers to the Hermite solution at the nth timestep and mth grid point. Note that the stable time step restriction results in a different dt for each method.

	Stage 1		Stage 2	
	Interpolate	Evolve	Interpolate	Evolve
Dual	$\mathbf{U}_m^n, \mathbf{U}_{m+1}^n \to \tilde{\mathbf{U}}_{m+1/2}^n$	$[t_n, t_n+dt)$	$\mathbf{U}_{m-1/2}^{n+1}, \mathbf{U}_{m+1/2}^n \to \tilde{\mathbf{U}}_m^n$	$[t_n, t_n+dt)$
Virtual	$\mathbf{U}_m^n, \mathbf{U}_{m+1}^n \to \tilde{\mathbf{U}}_{m+1/2}^n$	$[t_n, t_n)$	$\mathbf{U}_{m-1/2}^{n+1}, \mathbf{U}_{m+1/2}^n \to \tilde{\mathbf{U}}_m^n$	$[t_n, t_n+dt)$
Central	$\mathbf{U}_{m-1}^n, \mathbf{U}_{m+1}^n \to \tilde{\mathbf{U}}_m^n$	$[t_n, t_n+dt)$		
Upwind	$\mathbf{U}_{m-1}^n, \mathbf{U}_m^n \to \tilde{\mathbf{U}}_m^n$	$[t_n, t_n+dt)$		

For each one-dimensional Hermite method discussed, the timestep restriction is presented only for the scalar advection equation. An extension to $k \times k$ systems of hyperbolic equations

$$\frac{\partial u}{\partial t} = A\frac{\partial u}{\partial x}, \qquad u \in \mathbb{R}^k, \qquad A \in \mathbb{R}^{k\times k}$$

results in timestep restrictions which are identical to the scalar advection equation, except that the wavespeed c is replaced by the spectral radius $\rho(A)$. A similar analysis extends these results to variable coefficient problems, and both are described in more detail in [15].

In the following sections, we introduce the Dual, Virtual, Central, and Upwind Hermite methods in more detail.

2.1 Dual Hermite method

Hermite methods were originally introduced by Goodrich, Hagstrom, and Lorenz in [15], using "primal" and "dual" grids to facilitate Hermite reconstructions at specific points. We refer to this specific Hermite method as the *Dual* Hermite method. The Dual Hermite method introduces the auxiliary grid $\tilde{\Omega}$, which is taken to be a dual or co-volume grid

$$\tilde{\Omega} = \{\tilde{x}_{m+1/2} = a + (m+1/2)h_x, \ \ m = 0, \cdots, K-1\}$$

such that the nodes of $\tilde{\Omega}$ are staggered a distance of $h_x/2$ between the nodes of Ω. For periodic grids, the dual grid also satisfies $\tilde{x}_{m+1/2+K} = \tilde{x}_{m+1/2}$. We associate also an approximation space $\mathcal{P}_{\tilde{\Omega}}^N$ to the dual grid

$$\mathcal{P}_{\tilde{\Omega}}^N = \left\{ \bigoplus_{m=1}^{K-1} \mathcal{P}^{2N+1}[\tilde{x}_{m-1/2}, \tilde{x}_{m+1/2}) \right\} \cap C^N[a,b].$$

Each $q(x) \in \mathcal{P}_{\tilde{\Omega}}^N$ is defined by degrees of freedom $\mathbf{Q}_{mj}$

$$\mathbf{Q}_{m+1/2,j} = \frac{h_x^j}{j!}\frac{\partial^j q}{\partial x^j}\bigg|_{\tilde{x}_{m+1/2}}, \qquad j = 0, \cdots, N+1, \qquad m = 0, \cdots, K-1.$$

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

In the Dual Hermite method, the interpolation length scale is $h_x/2$, such that the timestep restriction on both primary and dual grids is

$$cdt < h_x/2.$$

The utility of the dual grid comes in considering the Hermite reconstruction at $x_m \in \Omega$, which interpolates at the points $L_m = x_m - h_x/2$ and $R_m = x_m + h_x/2$, such that

$$(L_m, R_m) = \left(x_m - \frac{h_x}{2}, x_m + \frac{h_x}{2}\right) = (\tilde{x}_{m-1/2}, \tilde{x}_{m+1/2}).$$

In other words, nodal data at points on the primary grid is used to produce a Hermite reconstruction $\tilde{u}_m(x)$ at each point $\tilde{x}_m$ on the dual grid. This also defines the interpolation length scale $h = h_x/2$, implying a time step restriction of $cdt < h_x/2$.

Denoting the vector of nodal data at a point $x_{m\pm1/2}$ as $\mathbf{Q}_{m\pm1/2}$, the interpolation procedure is

$$\tilde{\mathbf{U}}_m = \mathbf{H}\begin{bmatrix} \mathbf{Q}_{m-1/2} \\ \mathbf{Q}_{m+1/2} \end{bmatrix},$$

where $\tilde{\mathbf{U}}_m$ represents a degree $2N+1$ expansion at a point x_m on the primary grid.

Suppose $u^n(x) \in P_\Omega^N$ is the solution at timestep n with degrees of freedom $\mathbf{U}_{mj}^n$. The Dual Hermite method interpolates the primary grid solution at x_m and x_{m+1} to the dual grid and evolves it in time. The degree $2N+1$ solution is then truncated, producing $v_{m+1/2}^{n+1/2}(x) \in P_{\tilde{\Omega}}^N$. Denoting degrees of freedom for $v_{m+1/2}^{n+1/2}(x)$ as $\mathbf{Q}_{m+1/2}^{n+1/2}$, this step of a Hermite method may be expressed as

$$\mathbf{Q}_{m+1/2}^{n+1/2} = \mathbf{R}\mathbf{T}^{dt}\mathbf{H}\begin{bmatrix} \mathbf{U}_{m-1}^n \\ \mathbf{U}_{m+1}^n \end{bmatrix}, \quad m = 0, \cdots, K-1.$$

In order to update the solution on the primary grid, the process is repeated, except that data from the dual grid is transferred to the primary grid before being evolved in time

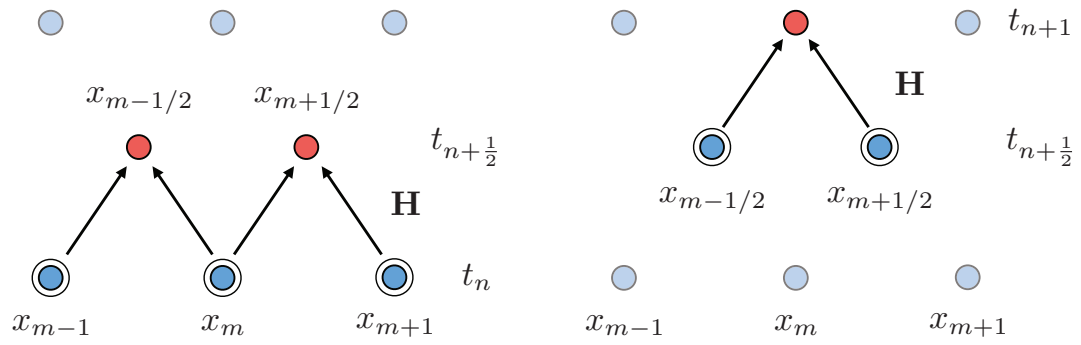
$$\mathbf{U}_m^{n+1} = \mathbf{R}\mathbf{T}^{dt}\mathbf{H}\begin{bmatrix} \mathbf{Q}_{m-1/2}^{n+1/2} \\ \mathbf{Q}_{m+1/2}^{n+1/2} \end{bmatrix}, \quad m = 0, \cdots, K-1.$$

The complete process illustrated in Fig. 1.

2.2 Virtual Hermite method

The Virtual Hermite method is motivated by the fact that two timesteps in the Dual Hermite method may be collapsed into a single update step on the primary grid involving nodal data at points x_{m-1}, x_m, x_{m+1}, bypassing explicit time evolution on the dual grid. This requires the formation of the time evolution operator explicitly, which unfortunately depends on physical and discretization parameters, and may vary between timesteps for

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) Primary to dual grid (b) Dual to primary grid

Figure 1: Interpolation procedures to and from primary and dual grids for the Dual Hermite method. Nodes which contribute data to the reconstruction are circled.

nonlinear problems. We propose the Virtual Hermite method to avoid the explicit construction of $\mathbf{T}^{dt}$ for the dual grid.

The Virtual Hermite method is identical to the Dual Hermite method except for time-evolution on the dual grid. In the Dual Hermite method, the solution on both the primary and dual grids is evolved with timestep dt. The Virtual Hermite method skips one evolution step, taking a timestep of $dt=0$ on the dual grid. The evolution operator $\mathbf{T}^{dt}$ on the dual grid then becomes the identity matrix, and the two steps of the Dual Hermite method may be collapsed into a single update step

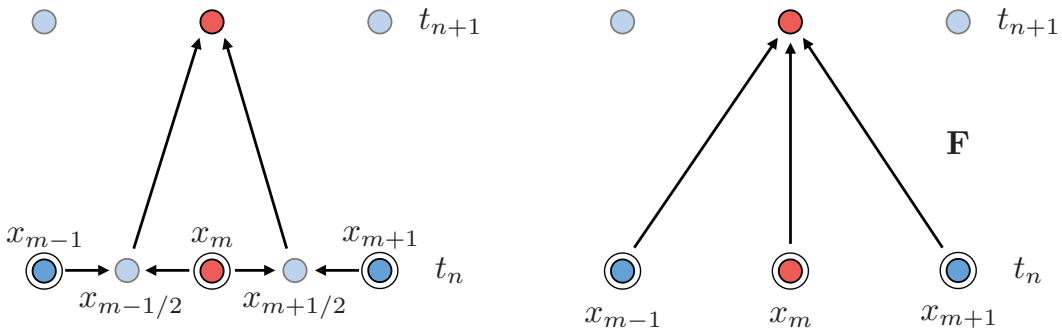
$$\mathbf{U}_m^{n+1}=\mathbf{F}\begin{bmatrix}\mathbf{U}_{m-1}^n\\ \mathbf{U}_m^n\\ \mathbf{U}_{m+1}^n\end{bmatrix},\qquad \mathbf{F}=\mathbf{R}\mathbf{T}^{dt}\mathbf{H}\begin{bmatrix}\boxed{\mathbf{RH}} & \mathbf{0}\\ \mathbf{0} & \boxed{\mathbf{RH}}\end{bmatrix},$$

where $\mathbf{0}\in\mathbb{R}^{(N+1)\times(N+1)}$ and $\mathbf{RH}\in\mathbb{R}^{(N+1)\times(2N+2)}$.

The matrix $\mathbf{F}$ resembles the co-volume filter analyzed in [35], which projects the solution on a primary grid to and from a staggered dual grid, suppressing spurious gradients of the solution on the primal grid. The Virtual Hermite method uses a similar procedure, where $\mathbf{F}$ maps Hermite data of degree N to Hermite data of degree $2N+1$ by transferring to and from the dual grid. We expect Virtual Hermite solutions to resemble filtered Dual Hermite solutions, which is supported by numerical experiments in Section 3.2.

The Virtual Hermite method obeys the timestep restriction $cdt<h_x/c$. This is the same restriction observed for a single step of the Dual Hermite method, since the interpolation length scale $h=h_x/2$ is the same in both cases. However, by forming $\mathbf{F}$ and using a single update step on the primary grid, the Virtual Hermite method eliminates the need to explicitly store and evolve dual grid solutions. We refer to $\mathbf{F}$ as an operator with a 3-node stencil since degree N data from the three nodes x_{m-1},x_m,x_{m+1} is required to produce degree $2N+1$ data at x_m, as shown in Fig. 2.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) Virtual dual nodes (b) Collapsed dual nodes

Figure 2: Virtual Hermite interpolation procedure, which transfers to and from an auxiliary grid (left) or as a single reconstruction step involving a three-node stencil (right). Nodes which contribute data to the reconstruction are circled.

We note that it is also possible to produce higher degree reconstructions using the same 3-node stencil. One approach is to directly using the degree $2N+1$ reconstructions at the two dual grid nodes to produce a final reconstruction of degree $4N+3$ at x_m, instead of truncating the dual grid reconstructions. Another option is to directly interpolate the $3(N+1)$ solution and derivative values at each node to produce a degree $3N+2$ reconstruction at x_m. In both cases, the higher degree reconstruction may then be evolved in time using a higher order scheme; however, we do not observe significantly improved convergence rates under such a procedure, and for some values of N, higher degree reconstructions results in an unstable scheme. Section 3.2 describes an alternative way to determine a higher degree reconstruction based on optimization of discrete dispersion and dissipation relations [34].

2.3 Central Hermite method

While the Virtual Hermite method removes the need to update the solution in time on the dual grid, the Central Hermite method sidesteps the use of an dual grid altogether by defining the Hermite reconstruction at x_m through interpolation at neighboring points

$$(L_m,R_m)=(x_m-h_x,x_m+h_x)=(x_{m-1},x_{m+1}).$$

As a result, the interpolation length scale is $h=h_x$ and the timestep restriction is

$$cdt<h_x.$$

Then, nodal data at x_m is constructed using data from nodes at x_{m-1},x_{m+1}. In other words, Hermite interpolation at a node on the primary grid is performed using its two neighbors and then evolved in time, resulting in an update step

$$\mathbf{U}_m=\mathbf{R}\mathbf{T}^{dt}\mathbf{H}\begin{bmatrix}\mathbf{U}_{m-1}\\ \mathbf{U}_{m+1}\end{bmatrix}.$$

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

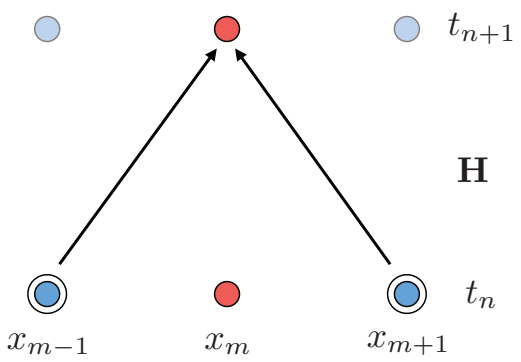


Figure 3: Central Hermite interpolation procedure, where a Hermite polynomial is constructed centered at x_i using nodal information from x_{i+1} and x_{i-1}. Nodes which contribute data to the reconstruction are circled.

The Central Hermite method may thus also be interpreted as two decoupled Dual Hermite methods on grids of size $2h_x$. The timestep restriction and numerical results in Section 3 also confirm this interpretation.

We note that by increasing the size of the interpolation interval h, the resulting timestep restriction increases independently of the grid spacing h_x. However, doing so also decreases the quality of the interpolation procedure, and Section 3.2 describes deleterious effects on the error and spectra of the resulting method.

The Central Hermite interpolation operator results in the 2-node stencil of Fig. 3, since nodal information from x_{m-1} and x_{m+1} is required to construct information at a x_m. Computationally, a smaller stencil results in fewer memory accesses for the reconstruction. While the difference between the Central Hermite and Virtual Hermite stencils in one space dimension is small, the difference becomes more pronounced in multiple dimensions. For a degree N Hermite method in d dimensions, each node in the stencil requires $(N+1)^d$ accesses, and a Central Hermite stencil contains 4 nodes in 2D, and 8 nodes in 3D, while the Virtual Hermite stencil contains 9 nodes in 2D and 27 nodes in 3D.

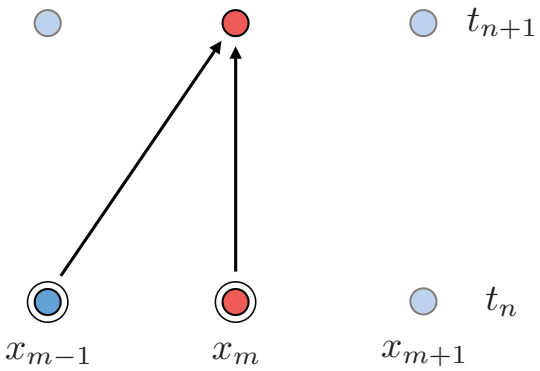
2.4 Upwind Hermite method

Each Hermite method presented has utilized a centered stencil, where solution values and derivatives are interpolated in a symmetric fashion around the reconstruction point. The Upwind Hermite method constructs instead a directional or one-sided Hermite reconstruction. This concept was used in [15] to enforce boundary conditions, though the use of such reconstructions may also take advantage of the directional nature of hyperbolic equations [9,22].

As shown in Fig. 4, an Upwind Hermite reconstruction $\tilde{u}_m(x)$ may be defined at x_m by interpolating solution and derivative values at the endpoints of the interval (x_{m-1},x_m)

$$\left.\frac{\partial^i \tilde{u}_m}{\partial x^i}\right|_{x_{m-1}} = \left.\frac{\partial^i u_{m-1}}{\partial x^i}\right|_{x_{m-1}}, \qquad \left.\frac{\partial^i \tilde{u}_m}{\partial x^i}\right|_{x_m} = \left.\frac{\partial^i u_m}{\partial x^i}\right|_{x_m}, \qquad i=0,\cdots,2N+1.$$

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) One-dimensional reconstruction

Figure 4: Interpolation procedure in one space dimension, where a Hermite polynomial is constructed centered at x_m using additional nodal information from and x_{m-1}. Nodes which contribute data to the reconstruction are circled.

The solution to this problem results in a degree $2N+1$ reconstruction at x_m. However, since the solution and derivative values at x_m are interpolated, the first $(N+1)$ coefficients at x_m remain unchanged, and only the remaining $(N+1)$ coefficients need to be computed. This may be done by multiplying the derivative values at x_{m-1},x_m by the last $(N+1)$ rows of the interpolation matrix $\mathbf{H}$. A downwind interpolation operator may be defined in a similar manner using information at (x_m,x_{m+1}) to produce a reconstruction at x_m.

For the advection equation specifically, the domain of influence is biased, such that the solution at time t_n+dt at the point x_m depends only on x_m-cdt at time t_n. Redefining the width of the interpolation interval for the Upwind Hermite method as $h=x_m-x_{m-1}=h_x$, the same domain of dependence arguments used previously imply that the method is stable if $cdt<h_x$. The timestep restriction of the Upwind Hermite method then matches that of the Central Hermite method, with the caveat that this result is specific to scalar advection equations and the sign of c. For example, for $c>0$, a downwind reconstruction would be unstable due to the fact that the interval (x_m,x_{m+1}) does not contain the domain of dependence for x_m at any time greater than t_n. We also note that, by similar arguments made in [15], the time evolution of the Upwind Hermite method by temporal Taylor series is also exact.

The Upwind Hermite method also requires special treatment when directionality is not readily apparent, such as for systems of hyperbolic equations. In one space dimension, the procedure may be adapted to reconstruct upwind and downwind characteristic variables, similar to the approach used in WENO reconstructions [27, 29]. However, the effectiveness of the characteristic approach does not appear to extend to all systems of equations in higher dimensions, as discussed in Section 4.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

2.5 Stability and convergence

The analysis carried out by Goodrich et al. [15] for the Dual Hermite method demonstrated that stability results from the contractive property of the Hermite interpolation procedure when measured using the $\|\cdot\|_{N+1}$ semi-norm defined as

$$\|\cdot\|_{N+1} = \|D^{N+1}(\cdot)\|_{L^2}.$$

Here D^{N+1} is the degree $N+1$ derivative operator. The interpolation operator, $\mathcal{I}$, satisfies

$$\|\mathcal{I}f\|_{N+1} \leq \|f\|_{N+1}, \tag{2.1}$$

for all periodic functions f in a degree $N+1$ Sobolev space.

Building on the stability of the interpolation procedure, inequality 2.1 is key in proving convergence for the Dual Hermite method. The proof of convergence is carried out by considering a truncation analysis in the $\|\cdot\|_{N+1}$ semi-norm which is then related to the L_2 norm through properties of the interpolation operators [15]. The arguments of stability and convergence can be extended to the methods presented in this work.

Recalling that the Virtual Hermite method is derived from the Dual Hermite method by taking a step size of zero on the dual grid; the arguments on stability and convergence for the Virtual Hermite method would follow directly from the analysis of the Dual Hermite method.

The Central Hermite method employs the same interpolation operators as the Dual Hermite method, thus the stability arguments remain the same. Since a dual grid is not introduced the convergence analysis will have to be carried out on the primary grid.

In contrast to the Central and Virtual Hermite methods, the Upwind Hermite method uses a directionally biased stencil. However, this directional procedure is still a form of Hermite-Birkhoff interpolation, which suggest that similar stability and error analyses can be performed for the Upwind Hermite method. As in the Central Hermite method, the analysis for convergence will have to be maintained on the primary grid.

A formal analysis of the new Hermite method variants is beyond the scope of this paper. In future work we intend to explore convergence arguments which do not require a dual grid.

3 Numerical experiments in 1D

To compare the performance of the new Hermite methods, we examine convergence rates and qualitative behavior for the Virtual. Central, and Upwind Hermite methods. Numerical results are shown for the periodic constant-coefficient scalar advection equation on the interval $[-1,1]$, using the Taylor expansion discussed in Section 1.1 to evolve in time.

We introduce also a CFL constant $C>0$ such that $dt = Ch/c$, where h is the size of the interpolation interval (for Virtual Hermite methods, $h = h_x/2$, while for Central and Upwind Hermite methods, $h = h_x$), and c denotes the wave speed. $C \approx 1$ sets the timestep

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

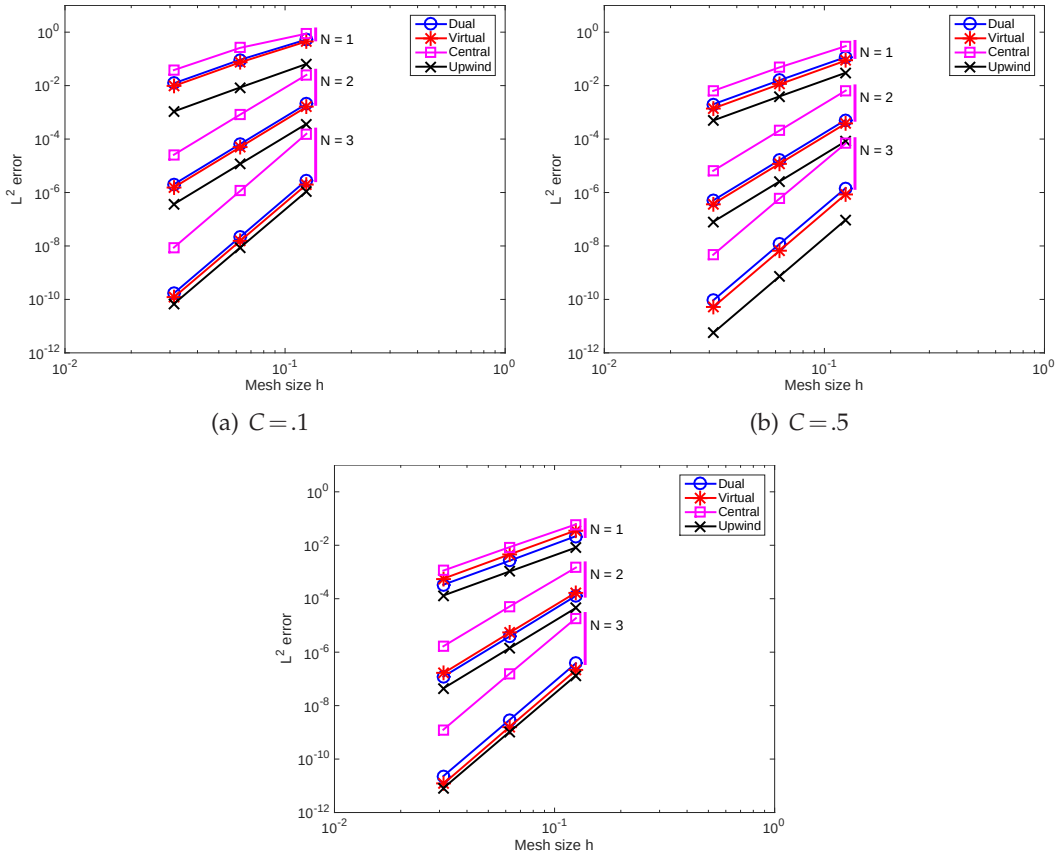
as large as possible based on the timestep restriction for each method, while $C \ll 1$ results in more timesteps than necessary as implied by stability. Since a filter-like step (the Hermite reconstruction) is applied at each timestep, small values of the CFL constant C (i.e. smaller timesteps than strictly necessary) may result in stronger filtering than necessary and larger errors.

3.1 Convergence rates

We report convergence rates for the one-dimensional scalar advection equation with speed $a=1$ and solution

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

We vary the CFL constant between $C=.1$, $C=.5$, and $C=.9$ and calculate L^2 errors at time $T=10$. For both methods, the error is smaller the closer C is to 1 as shown in Fig. 5.



(a) $C=.1$ (b) $C=.5$

(c) $C=.9$

Figure 5: Convergence of L^2 errors for Hermite methods for the one-dimensional advection equation with a smooth sine solution.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

Table 2: L^2 rates of convergence for Hermite methods for the one-dimensional advection equation with a smooth sinusoidal solution.

	$C=.1$			$C=.5$			$C=.9$		
N	1	2	3	1	2	3	1	2	3
Dual	2.72	4.99	7.02	2.93	5.0	6.98	3.02	5.02	7.01
Virtual	2.67	5.0	7.00	2.96	5.0	7.02	2.99	5.01	7.07
Central	1.71	4.94	7.06	2.62	4.98	6.92	2.92	4.98	6.99
Upwind	2.94	4.99	6.98	2.96	5.0	7.02	3.02	5.03	7.03

Convergence rates are reported in Table 2, and except for the lowest order case $N=1$ and smallest CFL constant $C=.1$, optimal $\mathcal{O}(h^{2N+1})$ rates of convergence were observed for all methods. Additionally, at higher orders of approximation, errors for the Virtual and Upwind Hermite method are very similar to those of the Dual Hermite method. For the Central Hermite method, the error is roughly a factor of 2^N greater than that of the other methods.

The growth of error in time is also examined for each Hermite method. Error estimates for discretizations of transient hyperbolic problems typically contain two terms which characterize spatially-dependent and time-dependent errors, respectively. Standard bounds are of the form

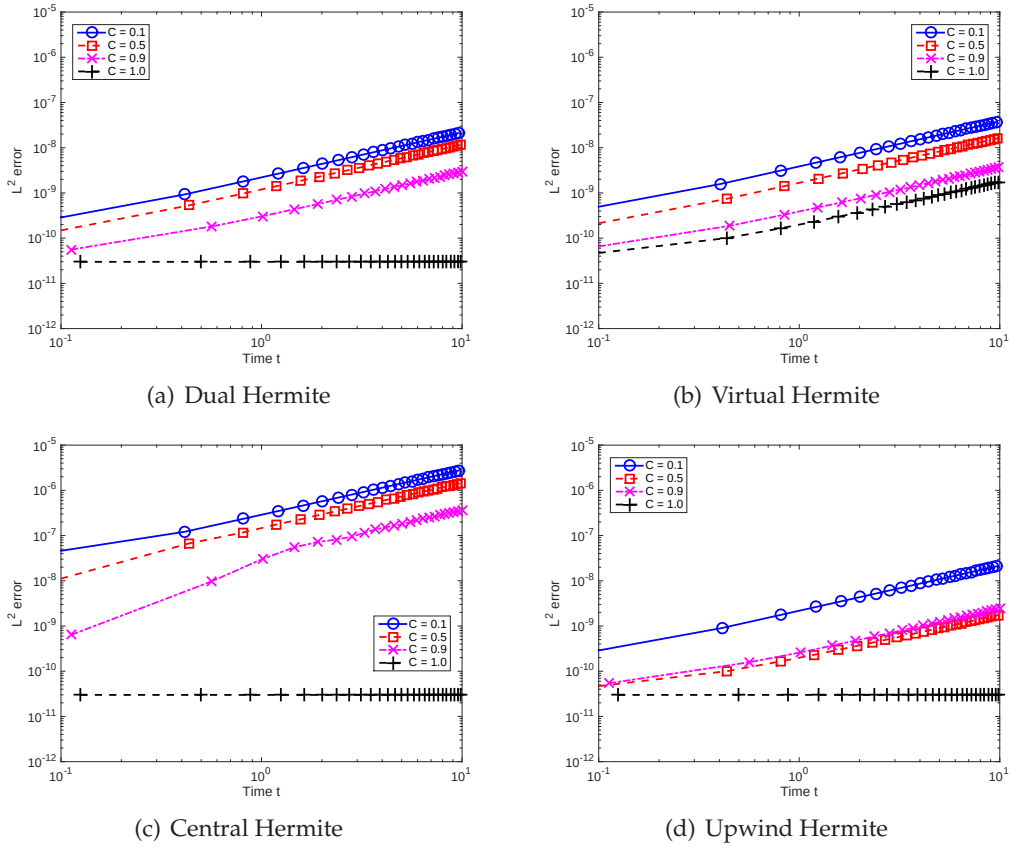
$$e(T) \leq (C_1 + C_2 T) h^{r(N)}$$

where $e(T)$ is some measure of error at time T, and $r(N)$ is some rate of convergence depending on the degree of approximation [8,19]. We confirm the linear growth of error in time for all Hermite methods in Fig. 6, with the exception of the case when $C=1$, which is discussed in Section 3.2. The growth of error for the Upwind and Dual Hermite methods is very similar. While the Central Hermite method develops larger errors than the Virtual Hermite method, the long-time rate of growth is identical for each value of C for both methods. Moreover, the Central Hermite method with $K=32$ results in time-dependent errors very similar to the Virtual Hermite method for $K=16$, indicating a strong dependence of the error on the size of the interpolation interval.

Growth of the error in time is often described in terms of dispersive and/or dissipation mechanisms intrinsic to numerical methods [30]. This may be illustrated by advecting an under-resolved function over several periods; the effect of numerical diffusion will be to smooth the profile out as time increases. Fig. 7 shows advection of the periodic Gaussian pulse initial condition $e^{-4\sin(\pi x)^2}$ over 5 periods for orders of approximation $N=1,2,3$ and a grid of 8 nodes.

As expected, all methods display diffusive behavior at low orders of approximation and low values of C, which is improved as N and C increase. The Dual and Upwind Hermite methods appear to be the least diffusive, though the difference between each method is small at higher orders of approximation. On a coarse $K=8$ mesh, the Central Hermite scheme behaves particularly poorly, displaying both spurious oscillations and significant numerical diffusion for all C.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) Dual Hermite (b) Virtual Hermite

(c) Central Hermite (d) Upwind Hermite

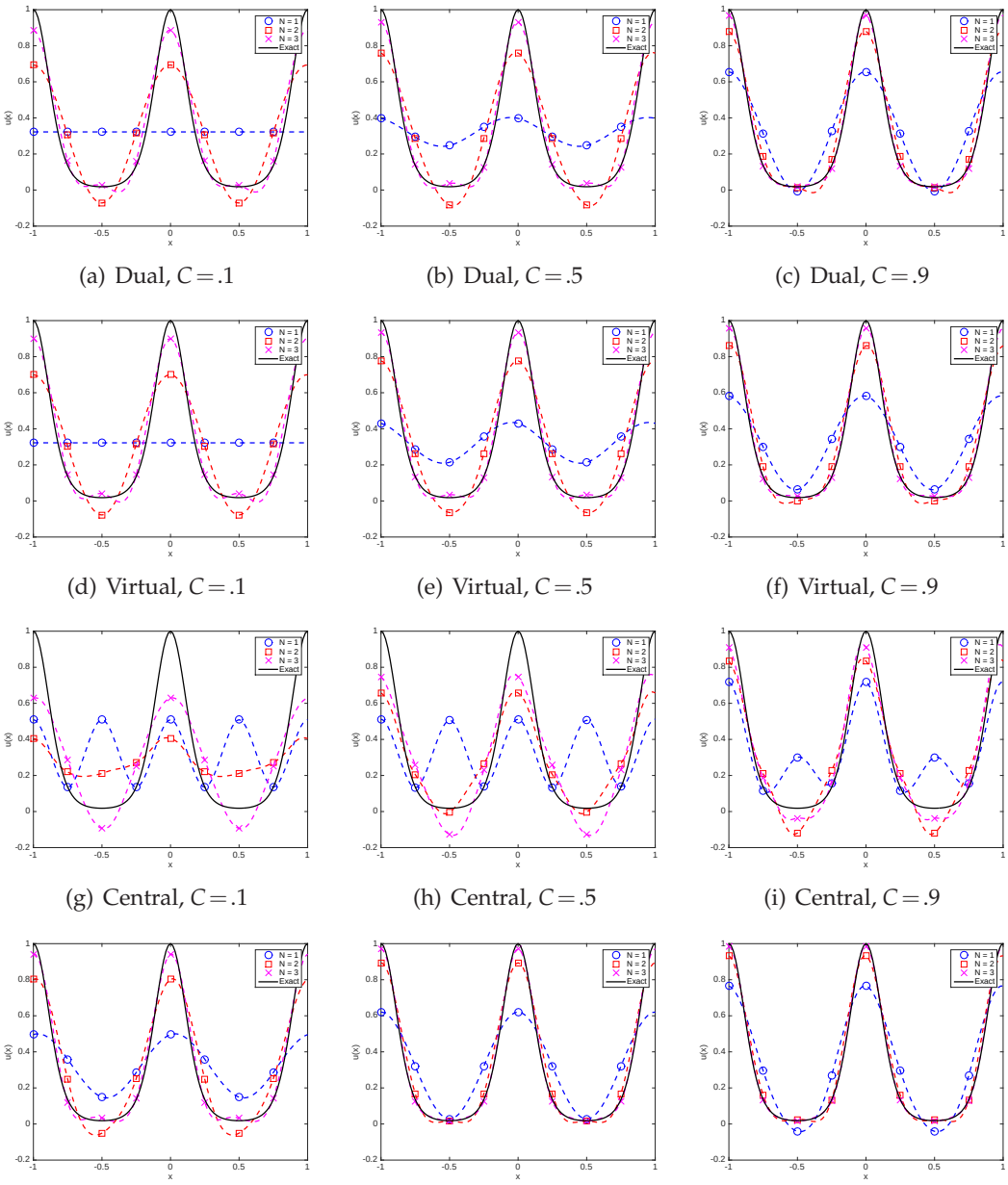
Figure 6: Growth of L^2 error in time for various Hermite schemes with $N=3$, $K=16$ in one dimension. The advection equation is solved up to time $T=10$ with a smooth sine solution, and the L^2 error is computed at each timestep.

Increasing to a finer mesh $K=16$, the Central Hermite scheme behaves comparably to the Virtual Hermite scheme. Qualitatively, the behavior of the Central Hermite scheme for $K=16$ resembles that of the Dual Hermite scheme for $K=8$. The L^2 errors for advection of a Gaussian, while not identical, are very close — for $N=2$, the Central Hermite scheme with $K=16$ results in an L^2 error of .0841903, while the Dual Hermite scheme with $K=8$ results in an L^2 error of .0839913. This is expected since, for Central with $K=16$ and Dual with $K=8$, the timestep restrictions and interpolation intervals are identical.

3.2 Spectra and dispersion/dissipation relations

Numerical experiments confirm the high order convergence of each method; however, the qualitative behavior of each method in convecting an under-resolved solution varies significantly. We seek to further analyze this behavior by computing the spectra of the update matrix and dispersion/dissipation relations for each Hermite method.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) Dual, $C=.1$ (b) Dual, $C=.5$ (c) Dual, $C=.9$

(d) Virtual, $C=.1$ (e) Virtual, $C=.5$ (f) Virtual, $C=.9$

(g) Central, $C=.1$ (h) Central, $C=.5$ (i) Central, $C=.9$

(j) Upwind, $C=.1$ (k) Upwind, $C=.5$ (l) Upwind, $C=.9$

Figure 7: Advection of a periodic Gaussian pulse by Hermite schemes with various CFL constants C on a grid of $K=8$ cells.

We define the update operator $\mathbf{S}$ such that, for solution degrees of freedom $\mathbf{U}^n$ at time t_n, the application of $\mathbf{S}$ evolves the solution at time t_n+dt

$$\mathbf{U}^{n+1}=\mathbf{S}\mathbf{U}^n.$$

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

Both the Dual and Central Hermite method march forward by $dt = Ch_x/c$ over a single timestep (the Dual Hermite method defines $dt = Ch_x/(2c)$, but takes timesteps on both primary and dual grids). Since the Virtual Hermite method takes the timestep to be $dt = Ch_x/(2c)$, we analyze $\mathbf{S}^2$ for the Virtual grid (corresponding to taking two timesteps instead of one) in order to normalize how far in time the update operator evolves the solution.

For the Dual, Central, and Upwind Hermite methods, setting $C = 1$ exactly (with respect to machine precision) results in exact evolution of the solution, though this is unique to constant coefficient equations. As a consequence, the update operator $\mathbf{A}$ becomes exactly equal to a circulant shift matrix. As a result, the order of convergence of each of these methods with $C = 1$ increases to $\mathcal{O}(h^{2N+2})$, and coincides with Hermite interpolation estimates given in [15, Lemma 3.1]. This same exact evolution property does not hold for the Virtual Hermite method.

The spectra in Fig. 8 suggest that the Virtual and Dual Hermite methods should behave similarly, as the eigenvalues of $\mathbf{A}$ are distributed similarly for both methods. Eigenvalues which lie on the unit circle are typically of the form $e^{i\omega}$, and are related to the non-dissipative propagation of modes of the form $e^{i\omega(x-ct)}$. For example, for $C = .1$, the eigenvalues fall closest to the unit circle around the point $(1,0)$, corresponding to the non-dissipative propagation of modes with small ω (low frequency modes). The remaining spectra lie within the unit circle, indicating dissipation of under-resolved modes. In contrast, the spectra for the Central Hermite method clusters not only around $(1,0)$ but also around $(-1,0)$, suggesting that under-resolved high frequency modes may be propagated without dissipation. These spurious modes may explain the behavior of the Central Hermite method observed in Fig. 7, where propagation of a Gaussian on a coarse grid resulted in "spurious" oscillatory behavior which remained over several periods of advection.

A study of the dispersion and dissipation error for the Dual Hermite method was reported in [20] using a modified equation and Bloch wave analysis in one dimension, which we adapt and apply to the Hermite methods introduced in this work. Dispersion and dissipation properties of Hermite methods depend mainly on the properties of one-dimensional Hermite interpolation matrix $\mathbf{H}$

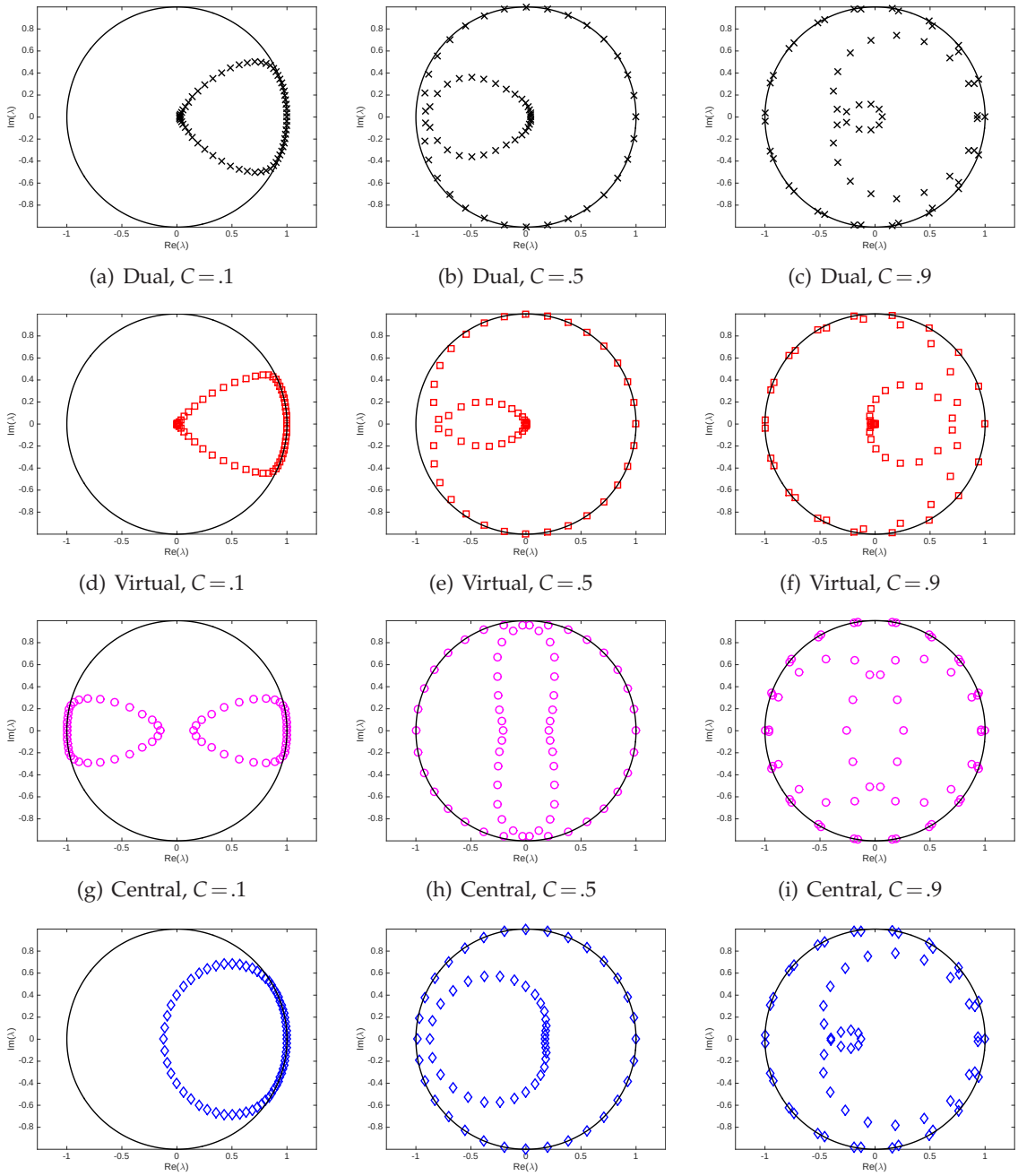
$$\mathbf{H} = [\mathbf{H}_L, \mathbf{H}_C, \mathbf{H}_R],$$

where $\mathbf{H}_C, \mathbf{H}_L, \mathbf{H}_R$ act on Hermite data associated with a given node and it's left/right neighbors to produce a reconstruction.

For a periodic grid, $\mathbf{S} \in \mathbb{R}^{K(N+1)\times K(N+1)}$ is a block tridiagonal matrix

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_L & \mathbf{S}_R & & \cdots & \mathbf{S}_L \\ \mathbf{S}_L & \mathbf{S}_C & \mathbf{S}_R & & \\ & \mathbf{S}_L & \mathbf{S}_C & \ddots & \\ & & \ddots & \ddots & \\ \mathbf{S}_R & & & \mathbf{S}_L & \mathbf{S}_C \end{bmatrix},$$

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) Dual, $C=.1$ (b) Dual, $C=.5$ (c) Dual, $C=.9$

(d) Virtual, $C=.1$ (e) Virtual, $C=.5$ (f) Virtual, $C=.9$

(g) Central, $C=.1$ (h) Central, $C=.5$ (i) Central, $C=.9$

(j) Upwind, $C=.1$ (k) Upwind, $C=.5$ (l) Upwind, $C=.9$

Figure 8: Spectra of the update matrix for each Hermite scheme at various CFL constants C. The order of approximation and grid size are fixed to be $N=3$ and $K=16$, respectively.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

where $\mathbf{S}_L=\mathbf{T}\mathbf{H}_L$, and similarly for $\mathbf{S}_C,\mathbf{S}_R$. For the Central and Upwind Hermite methods, $\mathbf{S}_C$ and $\mathbf{S}_R$ are zero, respectively.

We perform a fully discrete Bloch analysis to examine dispersive and dissipative properties of each Hermite method. This is done by representing the wave solution $e^{ik(x-ct)}$ in the Hermite basis, and noting that the solution is shifted in both space and time by scaling with a complex exponential

$$u(x,t+dt)=u(x,t)e^{-ikcdt}, \qquad u(x+h,t)=u(x,t)e^{ikh}.$$

Assuming a uniform grid spacing h, the discrete evolution of the interpolated exact solution at a node x_m from time t_n to t_n+dt is then given by

$$\left(e^{-ikh}\mathbf{S}_L+\mathbf{S}_C+e^{ikh}\mathbf{S}_R\right)\mathbf{u}_m^n=\lambda_h\mathbf{u}_m^{n+1}.$$

Since the timestep restrictions for the Dual and Virtual Hermite methods are $dt=Ch/(2c)$ as opposed to $dt=Ch/c$, the dispersion relations are measured over two timesteps. For the Dual Hermite method, this implies that $\mathbf{S}$ captures the evolution of the solution from the primal to dual grid, then back to the primal grid. For the Virtual Hermite method, this requires taking two timesteps and substituting for $\mathbf{S}$ the matrix $\mathbf{S}^2$, which is block pentadiagonal.

An eigenvalue problem may be solved for the discrete Floquet multiplier λ_h, whose real and imaginary parts correspond to numerical dispersion and dissipation, respectively. For each Hermite method, we measure the relative error between the discrete and the true Floquet multiplier

$$E_{kh}=\frac{\left|\lambda_h-e^{-ikcdt}\right|}{\left|e^{-ikcdt}\right|}$$

over a single timestep as a function of kh, the order of approximation N and the CFL constant $C=.1,.5,.9$.

Fig. 9 shows the error E_{kh} over a range of kh. The Central Hermite method shows the largest errors, while Upwind Hermite shows the smallest errors. The errors for the Virtual and Dual methods lie in-between, with the Dual Hermite displaying smaller errors at low N. Smaller values of the CFL constant C increase the dispersion and dissipation error, though the effect is less noticeable as N increases.

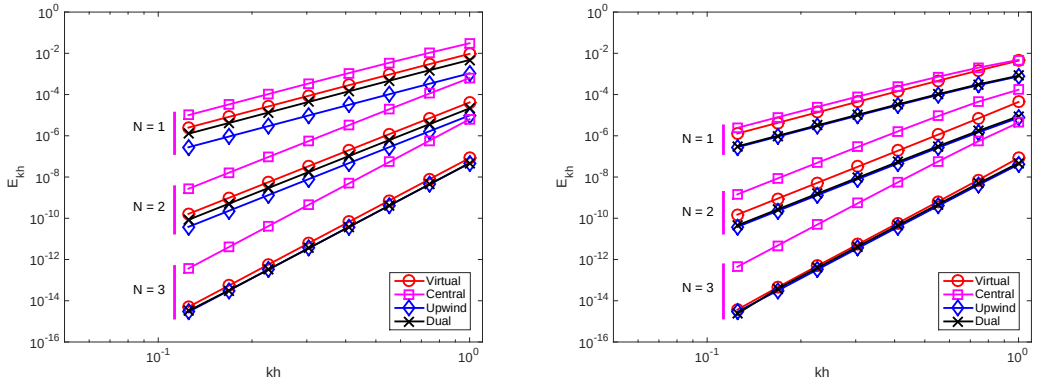
For each method, the error E_{kh} is observed to follow

$$\frac{\left|\lambda_h-e^{-ikcdt}\right|}{\left|e^{-ikcdt}\right|}\approx C_N\left(\frac{kh}{c}\right)^{2N+2},$$

where N is the number of degrees of freedom per node, and the underlying Hermite approximation space is of degree $2N+2$.[†].

[†]If we seek instead the error in the discrete and exact wavenumbers $|\omega-\omega_h|$, we recover convergence rates of $\mathcal{O}(h^{2N+1})$ and $\mathcal{O}(h^{2N+3})$ for the real and imaginary parts, respectively. In comparison, DG with covolume filtering achieves rates of $\mathcal{O}(h^{2N+2})$ and $\mathcal{O}(h^{2N+3})$ for the real and imaginary parts, respectively [35], while Galerkin methods result in rates of either $2N+1$ or $2N+3$ (for N even or odd) under a degree N approximation space [1].

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) $C=.1$ (b) $C=.9$

Figure 9: Dispersive and dissipative errors $\left|\lambda_h - e^{-ikcdt}\right| / \left|e^{-ikcdt}\right|$ for each Hermite scheme, with $N=1,2,3$. The computed errors are observed to behave as $\mathcal{O}(kh/c)^{2N+2}$.

3.2.1 Optimizing dispersive and dissipative errors

Finally, motivated by Dispersion Relation Preserving (DRP) finite difference schemes [34], dispersive and dissipative errors may be improved through optimization of entries of the interpolation matrix. As mentioned in Section 2.2, the Virtual Hermite method produces a degree $2N+1$ reconstruction using degree N data from three nodes, though there is sufficient data to define a higher $3N+2$ degree reconstruction. We define the interpolation matrix $\tilde{\mathbf{H}} \in \mathbb{R}^{(3N+3)\times(3N+3)}$

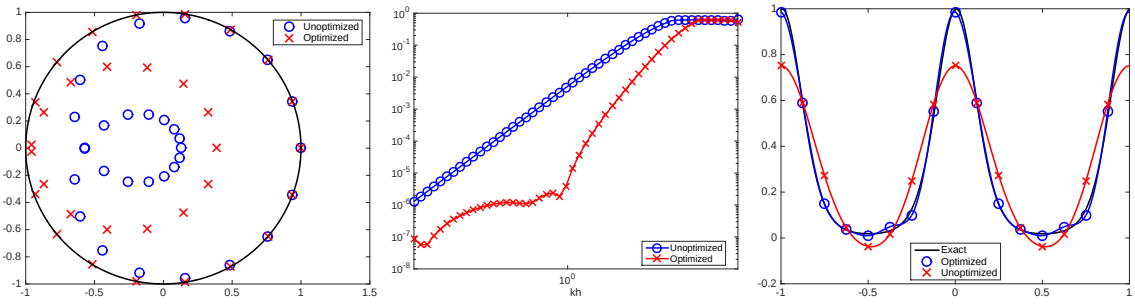
$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}_2 \end{bmatrix},$$

where $\mathbf{H}$ is the Virtual Hermite interpolation matrix. DRP schemes enforce a fixed order of approximation for a given finite difference stencil, while using additional degrees of freedom to optimize the dispersion relation. Similarly, fixing the first $2N+2$ rows of $\tilde{\mathbf{H}}$, the reconstruction implied by $\tilde{\mathbf{H}}$ is enforced to match that of the Virtual Hermite reconstruction for the first $2N+2$ coefficients, while entries of the matrix $\mathbf{H}_2$ (which determine higher order coefficients) are used to minimize dispersion and dissipation errors. The entries of $\mathbf{H}$ depend on the ratio between $L_m - \tilde{x}$ (or $R_m - \tilde{x}$) and as the grid spacing h. Since this ratio is constant as a function of h, $\mathbf{H}_2$ does not change drastically as a grid is refined. However, the optimization does appear to be sensitive to the value of C.

To demonstrate the effect of optimization, we compare the Virtual Hermite method to an optimized scheme for $N=1$ and CFL constant $C=.9$. We produce the optimized submatrix $\mathbf{H}_2$ by minimizing the real and imaginary parts of the relative dispersion error for the advection equation

$$\left(\frac{\mathrm{Re}\left(\lambda_h - e^{-ikcdt}\right)}{\mathrm{Re}\left(e^{-ikcdt}\right)}\right)^2 + \left(\frac{\mathrm{Im}\left(\lambda_h - e^{-ikcdt}\right)}{\mathrm{Im}\left(e^{-ikcdt}\right)}\right)^2$$

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) Spectra of **S** (b) E_{kh} (c) Convection over 5 periods

Figure 10: Spectra, dispersion/dissipation errors, and convection of the initial condition $e^{-4\sin(\pi x)^2}$ over 5 periods. Results are shown for both standard and optimized Virtual Hermite methods with $N=1$, $K=16$, and $C=.9$. Optimization is done on a coarse $K=8$ mesh.

with $c=1$ and $K=8$ grid cells. The same optimized submatrix $\mathbf{H}_2$ is then used on a finer $K=16$ grid, and computed solutions for the Virtual and optimized Hermite methods are compared for the initial condition $e^{-4\sin(\pi x)^2}$ in Fig. 10. The spectra of the update matrix **S** and dispersion/disspation errors E_{kh} are also compared in Fig. 10. The dispersion error and spectra are shown to be significantly improved, and numerical results indicate that under-resolved features are convected with greater accuracy.

Unfortunately, the benefits of such an approach appear to be limited to low orders of approximation. At higher orders, optimization did not reduce the dispersion and dissipation error E_{kh} significantly compared to the unoptimized scheme. Additionally, the stability of such an approach is not guaranteed for Hermite methods (compared to DRP schemes, which optimized over symmetric stencils to guarantee stability). For example, for $N=1$, the spectral radius of the update matrix for the unoptimized scheme was computed to be $\rho(\mathbf{S})=1$ to machine precision. For the $N=1$ optimized scheme, $\rho(\mathbf{S})\approx 1.0005$, and strict enforcement of $\rho(\mathbf{S})\leq 1$ resulted in either non-convergence of the optimization problem or subpar dispersion and dissipation properties. Further study is required to address these issues.

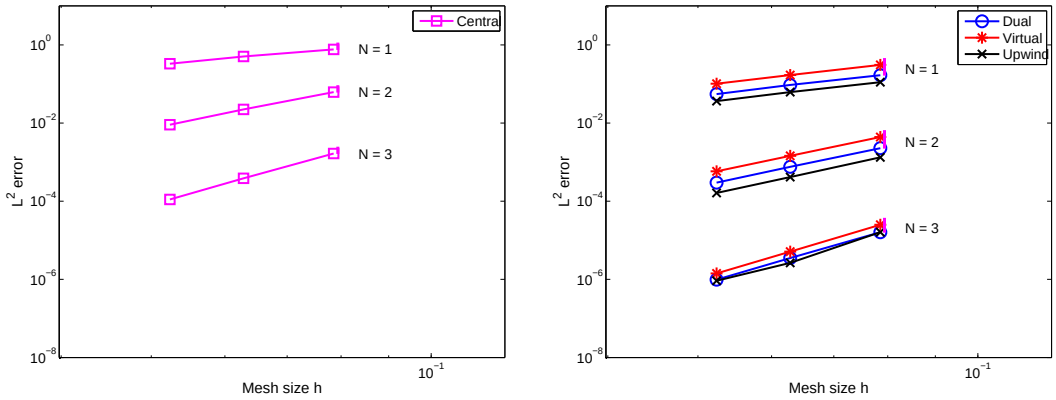
3.3 Variable coefficients and nonlinearities

We complete our experiments in one dimension by extending each new Hermite method to equations with spatially varying coefficients and nonlinearities. The Cauchy-Kowalewski procedure, discussed in Section 1.1, is applied to the example with spatially varying coefficients while numerical integration is used for the nonlinear equation.

3.3.1 Variable coefficients

As a first example we consider an advection equation with spatially varying coefficients as used in [15],

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) $C=0.9$ (b) $C=0.9$

Figure 11: L^2 rates of convergence for Hermite methods for the one-dimensional advection equation with variable coefficients and a time-dependent forcing function. The CFL constant is fixed set to $C=0.9$.

$$\frac{\partial u}{\partial t}=\left(1+\frac{\sin(x)}{2}\right)\frac{\partial u}{\partial x}+f(t,x).$$

The forcing term is chosen such that the analytic solution is $u=\cos(16(x+t))$. The domain is chosen to be $[0,2\pi]$ and the wave is propagated to $T=1.25$. The number of cells is varied between 80, 100, and 120. Fig. 11 reports the accuracy of the methods for a fixed CFL constant, C=0.9. Results for lower CFL constants are similar, with the exception that Hermite methods of order $N=1$ exhibit increased dispersion due to the interpolation procedure (Section 3.2). Table 3 presents the observed rates of convergence for the Hermite methods under varying CFL constants.

Table 3: L^2 rates of convergence for Hermite methods for the one-dimensional advection equation with variable coefficients and a time-dependent forcing function.

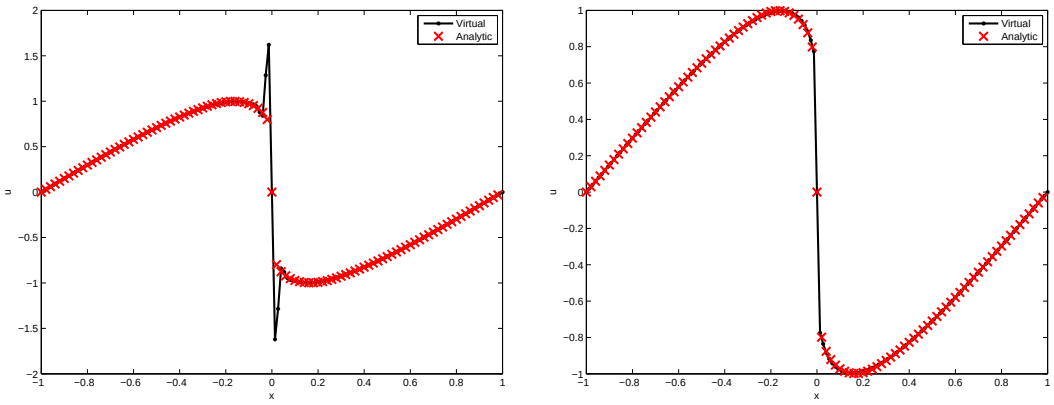
	$C=.1$			$C=.5$			$C=.9$		
N	1	2	3	1	2	3	1	2	3
Dual	2.05	4.93	7.01	2.52	5.00	6.94	2.74	5.00	6.88
Virtual	2.11	4.92	7.00	2.57	5.00	7.01	2.72	4.98	7.07
Central	0.98	4.55	7.05	1.55	4.80	6.90	2.11	4.74	6.70
Upwind	2.58	4.98	6.98	2.76	5.00	6.98	2.81	4.98	7.01

3.3.2 Nonlinear equations

Our second example considers the viscous Burger's equation [17],

$$\frac{\partial u}{\partial t}=-u\frac{\partial u}{\partial x}+\epsilon\frac{\partial^2 u}{\partial x^2}.$$

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) Under Resolved Transition (b) Resolved Transition

Figure 12: Approximation of the viscous Burger's equation with an under resolved approximation and a resolved approximation with a viscous term set to $\epsilon = 1e-3$ using the Virtual Hermite method.

The initial condition is set to $u(x,0) = -\sin(\pi x)$ on the bi-unit domain with periodic boundary conditions. For nonlinear equations, the classic temporal Taylor series expansion requires repeated temporal differentiation of the right-hand side. The repeated differentiation spawns many new terms. In contrast, using ODE integration requires only a single right-hand side evaluation [15–17]. The spatial order of convergence is preserved as long as a sufficiently accurate time stepping procedure is employed.

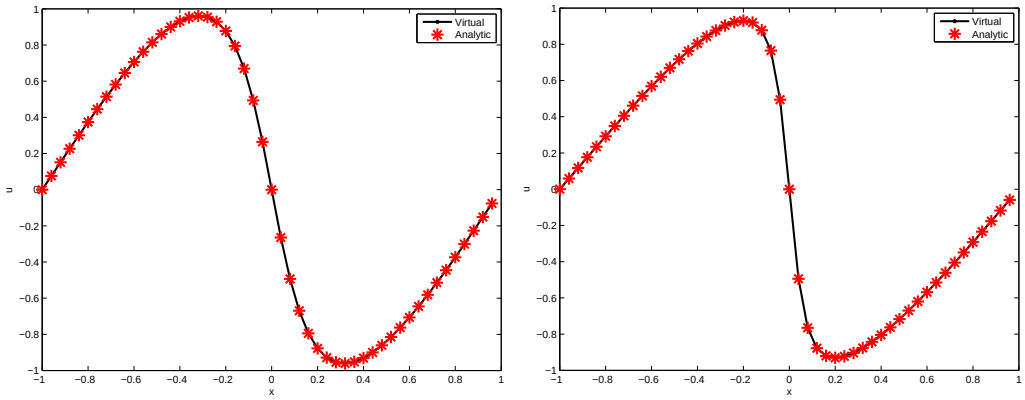
The solution develops a narrow transition zone at $T = 0.3$, thus we examine the approximation before and after the transition forms. To illustrate the resolution of Hermite methods, Fig. 12 shows an under resolved and resolved transition at $T = 0.35$ using 50 and 150 grid points for a small epsilon value ($1e-3$). A single fourth order Runge-Kutta time step is taken using a CFL constant of 0.1 and a scheme of order $N = 2$. Hermite methods with a symmetric stencil were able to adequately resolve the transition. In the presence of diffusion, a purely biased Upwind Hermite method is not stable due to its directional reconstruction. Computational experiments suggest that the Upwind Hermite method may be stable if a dynamic stencil could be used to accommodate changes in the directionality of the wave, which we intend to explore in future work.

To study accuracy and convergence, we set $\epsilon = 0.02$ and vary the number of cells between 25, 35, 45, and 55. Table 4 reports the accuracy before and after the transition, and Fig. 13 illustrates the solutions.

4 Extension to two dimensions

Each Hermite method may be extended to higher dimensions naturally through a tensor product construction. In this work, we take the grid Ω to be the tensor product of one-dimensional grids. Assuming grid spacings h_x, h_y in the x and y directions, respectively,

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) $T=0.2$ (b) $T=0.35$

Figure 13: Approximation of the viscous Burger's equation before ($T=0.2$) and after the transition ($T=3.5$) using the Virtual Hermite method with 50 grid points and the viscous term set at ($\epsilon=0.02$).

Table 4: L^2 rates of convergence for Hermite methods for the one-dimensional viscous burgers equation with a smooth sinusoidal solution before ($T=0.2$) and after the transition ($T=0.35$).

	$T=.2$				$T=.35$			
K	25	35	45	55	35	45	55	65
Dual (N=2)	3.3(-5)	6.75(-6)	1.87(-6)	6.24(-7)	1.9(-4)	6.02(-5)	2.21(-5)	8.36(-6)
Rate	-	4.7	5.1	5.5	-	4.68	4.98	5.83
Virtual (N=2)	6.22(-5)	1.31(-5)	3.78(-6)	1.33(-6)	3.59(-4)	1.16(-4)	4.54(-5)	1.88(-5)
Rate	-	4.6	4.9	5.2	-	4.48	4.69	5.26
Central (N=2)	7.16(-4)	1.65(-4)	5.26(-5)	2.04(-5)	3.6(-3)	1.3(-3)	5.52(-4)	2.26(-4)
Rate	-	4.4	4.5	4.7	-	4.07	4.26	4.47

each point $(x_m,y_m)\in\Omega$ admits the tensor-product expansion

$$u_m(x,y)=\sum_{j=0}^{N}\sum_{k=0}^{N}\mathbf{u}_{jk}\left(\frac{x-x_m}{h_x}\right)^j\left(\frac{y-y_m}{h_y}\right)^k.$$

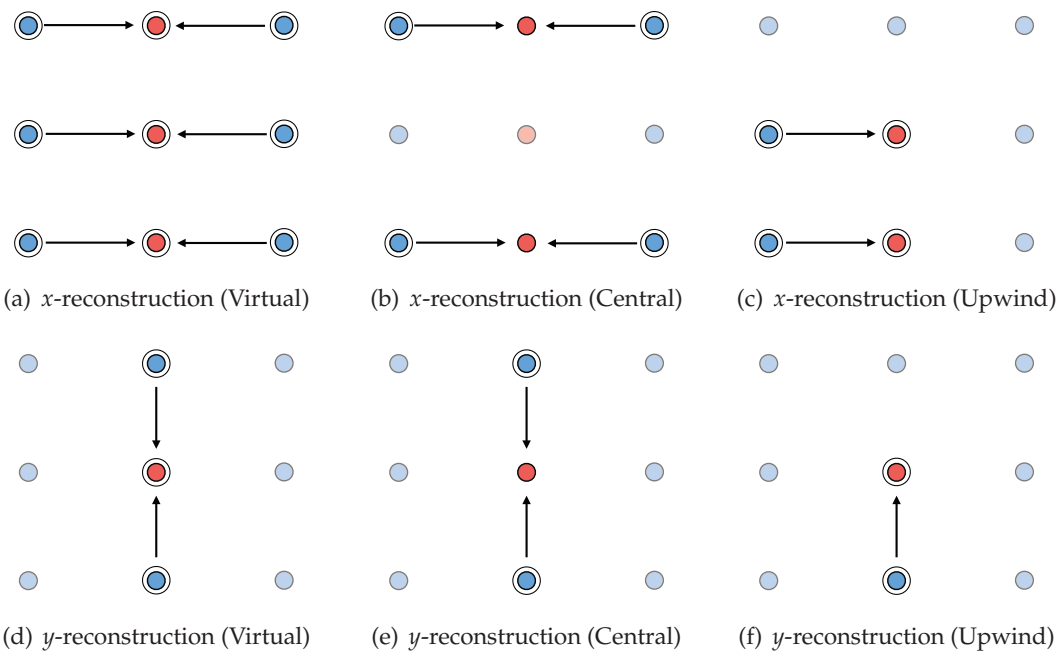
For linear autonomous equations, a temporal Taylor series may be used to evolve the solution in time. Algorithm 2 describes this process for the two-dimensional scalar advection equation, using derivative matrices $\mathbf{D}_x,\mathbf{D}_y$ for the x and y coordinates, respectively. Due to the tensor-product nature of the Hermite interpolants in higher dimensions, the Taylor series must be of order $d(2N+1)$ to be exact in d-dimensions [15]. Numerical experiments indicate that reducing the degree of the Taylor expansion in time results in a tighter timestep restriction; however, this only decreases the restriction by some constant factor, which is independent of the order of approximation. In all experiments, the increase in the order of the Taylor expansion did not correspond with a significant decrease in error.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

Algorithm 2 Time evolution procedure for 2D scalar advection. $\tilde{N}$ may be taken to be $d(2N+1)$ for exact time evolution.

procedure TWO-DIMENSIONAL TEMPORAL TAYLOR SERIES EVALUATION
  $\mathbf{w} = \tilde{\mathbf{u}}^n$
  for $\ell = \tilde{N}, \tilde{N}-1, \cdots, 0$ **do**
    $\mathbf{w} = \mathbf{w} + \frac{dt}{1+\ell}(-c\mathbf{D}_x - c\mathbf{D}_y)\tilde{\mathbf{u}}^n$
  $\tilde{\mathbf{u}}^{n+1} = \mathbf{w}$

While time evolution is extended in a straightforward way regardless of spatial dimension, interpolation operators in higher dimensions are defined through applications of 1D interpolation operators along each coordinate direction. The application of operators for the Virtual and Central Hermite methods is illustrated in Fig. 14, and we refer the reader to [15] for more details on the extension of the Dual Hermite method to multiple dimensions. The Upwind Hermite reconstruction may be adapted to the advection equation in higher dimensions by considering the direction of advection along each coordinate, though the direction of the reconstruction will depend on the sign of c_x at each point.



(a) x-reconstruction (Virtual) (b) x-reconstruction (Central) (c) x-reconstruction (Upwind)

(d) y-reconstruction (Virtual) (e) y-reconstruction (Central) (f) y-reconstruction (Upwind)

Figure 14: Two-dimensional Hermite reconstruction stencils. The Upwind reconstruction assumes $c_x, c_y > 0$. Nodes which contribute data to a reconstruction are circled.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

4.1 Numerical experiments in two dimensions

We consider two model problems in two space dimensions: the periodic advection equation

$$\frac{\partial u}{\partial t}+c_x\frac{\partial u}{\partial x}+c_y\frac{\partial u}{\partial x}=0.$$

where $\mathbf{c}=(c_x,c_y)$ is a unit vector, and the isotropic wave equation in first order form

$$\frac{1}{c^2}\frac{\partial p}{\partial t}=-\nabla\cdot\mathbf{u}, \qquad \frac{\partial \mathbf{u}}{\partial t}=-\nabla p,$$

where c is a specified wavespeed, p is pressure and $\mathbf{u}=(u,v)$ is the velocity. The CFL condition for the two-dimensional advection equation is

$$\|\mathbf{c}\|dt<h,$$

while the wave equation depends on the maximum wavespeed in each coordinate direction. For the non-dimensional isotropic wave equation above, this results in the CFL condition $cdt<h$.

We note that behavior of the Upwind Hermite method is reported only for the advection equation. While the Upwind Hermite method may be extended to hyperbolic systems through a characteristic-based approach, numerical experiments with the isotropic wave equation indicated that the method resulted in a timestep restriction depending on the degree of approximation. We intend to explore additional generalizations of the Upwind Hermite method to systems of equations in multiple dimensions in the future.

As in one dimension, we introduce a CFL constant C such that $dt=Ch$, and examine the behavior of each Hermite method at various values C. Fig. 15 shows the L^2 convergence of the Virtual and Central Hermite methods for the advection equation. We take the advection speeds c_x,c_y and exact solution to be

$$c_x=\cos(\pi/3), \quad c_y=\sin(\pi/3), \qquad u(x,y,t)=\sin(\pi(x-c_xt))\sin(\pi(y-c_yt)).$$

and compute L^2 errors for isotropic grids of $8\times 8,16\times 16$, and 32×32 nodes.

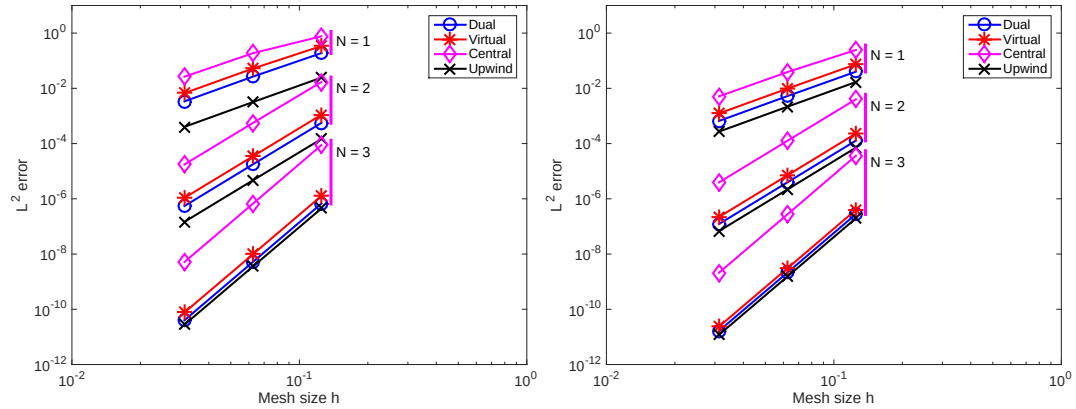
Unlike Hermite methods in one space dimension, a temporal Taylor series of degree $d(2N+1)$ is required for exact time evolution in d dimensions [15]. All experiments use exact time evolution; however, decreasing the order of the temporal Taylor series to $2N+1$ did not result in a significant decrease in error, though the stable timestep restriction decreases by a factor of .5. L^2 rates of convergence are reported in Table 5 for $T=1$. As in the one-dimensional case, the L^2 error is observed to converge at a rate close to h^{2m+1}.

We repeat the Dual, Virtual, and Central Hermite convergence experiments for the periodic wave equation in 2D, using the exact solution

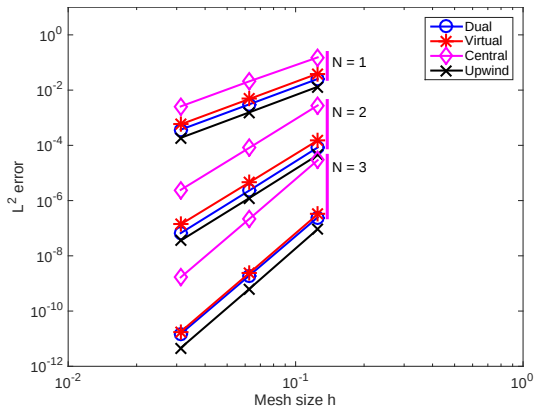
$$p(x,y,t)=\sin(\pi x)\sin(\pi y)\cos(\sqrt{2}\pi t).$$

Upwind Hermite results are not reported, since a straightforward application of the two-dimensional upwind Hermite reconstruction does not yield a stable procedure for the

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) $C=.1$ (b) $C=.5$



(c) $C=.9$

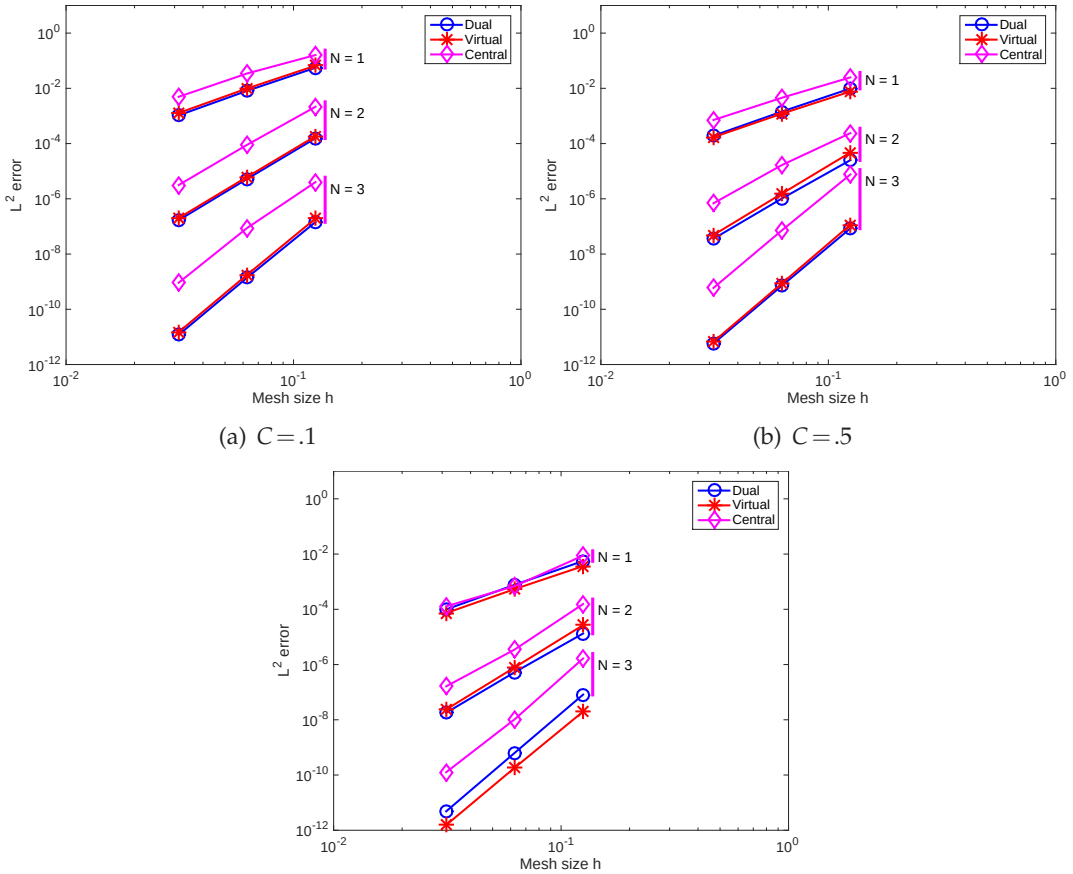
Figure 15: Convergence of L^2 errors for each Hermite scheme for the two-dimensional advection equation with a smooth sinusoidal solution.

Table 5: L^2 rates of convergence of the Dual, Virtual, and Central Hermite methods for the advection equation in two dimensions.

	$C=.1$			$C=.5$			$C=.9$		
N	1	2	3	1	2	3	1	2	3
Dual	2.90	5.00	7.01	2.97	5.02	7.02	3.05	5.15	7.00
Virtual	2.83	5.00	7.01	2.95	5.02	7.03	3.04	5.04	7.06
Central	2.43	4.96	7.03	2.79	5.00	7.04	2.95	5.08	7.04
Upwind	2.96	4.98	6.99	2.98	5.02	7.01	3.07	5.14	7.14

wave equation. Convergence experiments are repeated for the set of grids used for advection, and Fig. 16 plots the L^2 errors in p at time $T=1$ for the Dual, Virtual, and Central Hermite methods. The Dual and Virtual Hermite methods produce errors of very similar

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) $C=.1$ (b) $C=.5$

(c) $C=.9$

Figure 16: Convergence of L^2 errors for the Dual, Virtual, and Central Hermite schemes for the isotropic wave equation in two dimensions.

Table 6: L^2 rates of convergence for the isotropic wave equation in 2D.

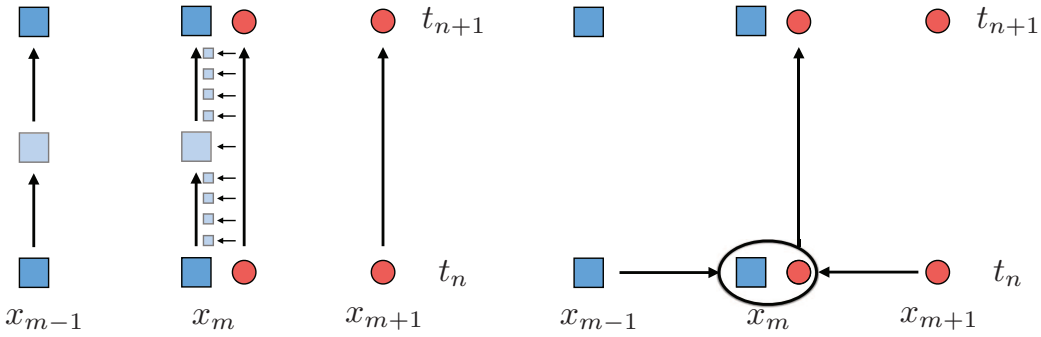
	$C=.1$			$C=.5$			$C=.9$		
N	1	2	3	1	2	3	1	2	3
Dual	2.86	4.93	6.79	2.84	4.73	6.92	2.91	4.77	7.02
Virtual	2.85	4.93	6.84	2.75	4.96	7.03	2.82	5.07	6.83
Central	2.51	4.71	6.05	2.56	4.22	6.82	3.05	4.95	6.84

magnitude, while the error for the Central Hermite method is larger by a factor of roughly 2^N as observed in 1D. Surprisingly, at $C=.9$ and $N=3$, the error for the Virtual Hermite method is lower than that of the Dual Hermite method. The L^2 rates of convergence are reported in Table 6.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

4.2 Coupling with Discontinuous Galerkin methods

Hermite methods may also be coupled to Discontinuous Galerkin methods in order to tackle more complicated geometries and boundary conditions. In [4], coupling conditions between DG and the Hermite method are constructed for both the primary and auxiliary grids using a least squares reconstruction and high order finite difference stencils. Since the Hermite methods introduced in this work do not require staggered grids, the transfer of information is simplified. We will refer to the order of approximation for the DG method as m. Hermite methods may transfer information to DG methods through the numerical flux. Due to a timestep restriction of $\mathcal{O}(h/m^2)$ for DG compared to the $\mathcal{O}(h)$ timestep restriction for Hermite methods, multiple DG substeps must be taken for each Hermite timestep. The following numerical experiments use a 4th-order Runge-Kutta scheme with 5 stages, and necessitates the evaluation of the numerical flux for each stage. To maintain high order convergence, we compute flux contributions by evaluating the high order Hermite interpolant in time, as shown in Fig. 17. The coupling from DG to Hermite is more fragile, as high order derivative information must be determined from the DG solution. In the following experiments, these derivatives are provided via a patch reconstruction at Hermite nodes [37]. Projection onto the Hermite basis directly yields high order Hermite coefficients; alternatively, these coefficients may then be determined by taking derivatives of the reconstructed polynomial.



(a) Hermite-DG coupling (4-stage RK with 2 substeps)

(b) DG-Hermite coupling (patch recovery)

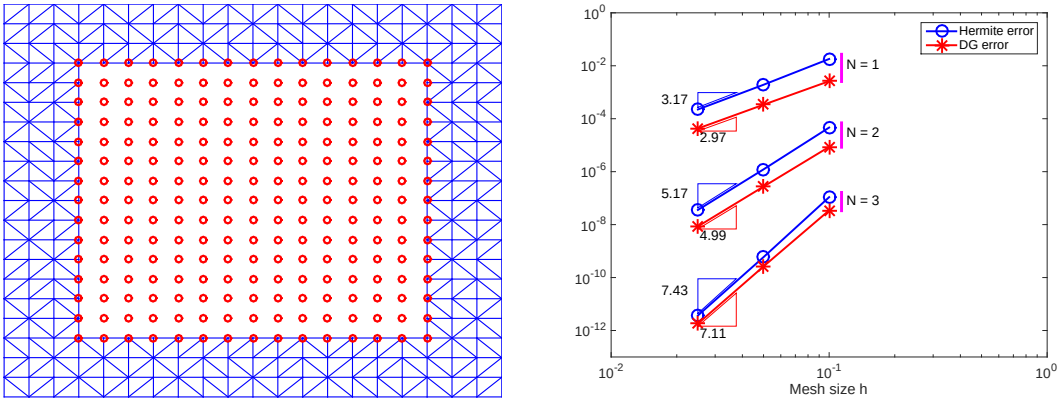
Figure 17: Coupling between Hermite and DG methods without staggered grids (DG nodes are squares, while Hermite nodes are circles).

We compute L^2 errors and convergence rates using the smooth solution

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

for wavespeed $c = \sqrt{1/2}$. On non-overlapping grids, a polynomial of degree $2N+1$ is constructed using both Hermite and DG data on neighboring elements. Since the best

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press



(a) Coupled mesh (b) Wave equation L^2 errors

Figure 18: (Left) Coarsest coupled mesh used for coupling Virtual Hermite and DG (Hermite nodes are circled). (Right) Convergence of L^2 errors in 2D for the isotropic wave equation.

possible convergence rate for DG is $\mathcal{O}(h^{m+1})$,‡ we take the order of approximation for DG to be $m=2N$ (where N is the degree of the Hermite method) in order to preserve the $\mathcal{O}(2N+1)$ convergence rate.

Results are shown in Fig. 18 for N=1,2,3 for the coupled Virtual Hermite-DG method, using $C=.8$ for the Hermite timestep. For $N>2$, convergence was limited by the order of the DG Runge-Kutta scheme, and a smaller timestep must be taken to recover optimal convergence rates. Similar observations were made when coupling Dual Hermite and DG methods [4]. Convergence results were similar when coupling Central Hermite and DG, though the Hermite error increases slightly.

The approximation of Hermite coefficients using DG becomes less accurate at higher orders, as roughly an order of convergence is lost per derivative with patch recovery methods. Formally, this should not affect the convergence rate of the coupled scheme, but it can affect the accuracy. A salient alternative to patch recovery is Smoothness-Increasing Accuracy-Conserving (SIAC) postprocessing [7, 31], which produces smooth reconstructions of the solution which converge with rate $\mathcal{O}(h^{2m+1})$. Under such a method, optimal convergence rates could be preserved using DG and Hermite methods with degrees $m=N$. The postprocessing of higher order derivatives may also yield additional accuracy in Hermite coefficients [32].

5 Conclusions and future work

We have presented a generalization of Hermite methods for periodic problems, and have investigated three new methods within this framework and compared their performance

‡Optimal convergence rates for upwind DG are typically observed in practice, and are provable on specific classes of meshes [6]. However, on general meshes, DG methods can expect at most $\mathcal{O}(h^{m+1/2})$.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

to the original Hermite method in the literature. The original Dual Hermite method results in a two-node stencil in one space dimension, and requires time integration on both primal and staggered (dual) grids. The Virtual Hermite method increases the stencil to three nodes, but avoids explicit storage of staggered grid degrees of freedom by fusing operations on the auxiliary and primary grid together. The Central Hermite method modifies the interpolation procedure in order to avoid a staggered grid, and in doing so, maintains a two-node stencil and doubles the timestep restriction. However, to achieve a specific error resolution, the Central Hermite method requires twice as many degrees as the original Dual Hermite method.

Additionally, the Central Hermite method may suffer from the propagation of spurious modes. The Upwind Hermite method achieves a resolution close to that of the Dual Hermite method, while maintaining the same two-node stencil and timestep restriction as the Central Hermite method. However, the stability of Upwind Hermite schemes does not appear to generalize in a straightforward manner to systems of equations in higher dimensions. Finally, since the Virtual, Central, and Upwind Hermite methods do not require dual grids, the coupling between Hermite and DG methods is simplified.

Future work will expand on the theoretical framework of each new method. In addition, we will explore stable extensions of upwind Hermite reconstructions to multidimensional wave problems and to use these simplified methods to produce efficient many-core parallel implementations in two and three space dimensions.

Acknowledgments

The authors wish to acknowledge the Matlab codes of the Hermite training library CHIDES (`http://www.chides.org`), as well as helpful discussions with Daniel Appelö.

Arturo Vargas is supported by an NSF graduate fellowship. Jesse Chan and T. Warburton are supported by NSF (award number DMS-1216674). Thomas Hagstrom is supported by NSF (award number DMS-1418871).

References

[1] Mark Ainsworth. Dispersive behaviour of high order finite element schemes for the one-way wave equation. *Journal of Computational Physics*, 259:1–10, 2014.

[2] Daniel Appelö, Matthew Inkman, Thomas Hagstrom, and Tim Colonius. Hermite methods for aeroacoustics: Recent progress. In *17th AIAA/CEAS Aeroacoustics Conference (32nd AIAA Aeroacoustics Conference), Portland, Oregon*, 2011.

[3] Claudio Canuto, M Youssuff Hussaini, Alfio Quarteroni, and Thomas A Zang. *Spectral Methods: Fundamentals in Single Domains*. Springer Berlin Heidelberg, 2006.

[4] Xi Ronald Chen, Daniel Appelö, and Thomas Hagstrom. A hybrid Hermite-discontinuous Galerkin method for hyperbolic systems with application to maxwell's equations. *Journal of Computational Physics*, 257:501–520, 2014.

[5] Prince Chidyagwai, Jean-Christophe Nave, Rodolfo Ruben Rosales, and Benjamin Seibold. A comparative study of the efficiency of jet schemes. *arXiv preprint arXiv:1104.0542*, 2011.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

[6] Bernardo Cockburn, Bo Dong, and Johnny Guzmán. Optimal convergence of the original DG method for the transport-reaction equation on special meshes. *SIAM Journal on Numerical Analysis*, 46(3):1250–1265, 2008.
[7] Bernardo Cockburn, Mitchell Luskin, Chi-Wang Shu, and Endre Süli. Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. *Mathematics of Computation*, 72(242):577–606, 2003.
[8] Bernardo Cockburn and Chi-Wang Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.
[9] Richard Courant, Eugene Isaacson, and Mina Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics*, 5(3):243–255, 1952.
[10] Michel O Deville, Paul F Fischer, and Ernest H Mund. *High-order methods for incompressible fluid flow*, volume 9. Cambridge University Press, 2002.
[11] Evan T Dye. Performance analysis and optimization of Hermite methods on Nvidia GPUs using CUDA. Master's thesis, University of New Mexico, 2015.
[12] Richard S Falk and Gerard R Richter. Explicit finite element methods for symmetric hyperbolic equations. *SIAM Journal on Numerical Analysis*, 36(3):935–952, 1999.
[13] Karsten Fischer. Convective difference schemes and Hermite interpolation. *International Journal for Numerical Methods in Engineering*, 12(6):931–940, 1978.
[14] Bengt Fornberg. The pseudospectral method: Comparisons with finite differences for the elastic wave equation. *Geophysics*, 52(4):483–501, 1987.
[15] John Goodrich, Thomas Hagstrom, and Jens Lorenz. Hermite methods for hyperbolic initial-boundary value problems. *Mathematics of computation*, 75(254):595–630, 2006.
[16] Thomas Hagstrom and Daniel Appelö. Experiments with Hermite methods for simulating compressible flows: Runge-Kutta time-stepping and absorbing layers. In *13th AIAA/CEAS Aeroacoustics Conference*, 2007.
[17] Thomas Hagstrom and Daniel Appelö. Solving pdes with Hermite interpolation. In *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, pages 31–49. Springer, 2015.
[18] Thomas Hagstrom, Daniel Appelo, Tim Colonius, Matthew Inkman, and Chang Youn Jang. Simulation of compressible flows using Hermite methods. *The Journal of the Acoustical Society of America*, 131(4):3429–3429, 2012.
[19] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, volume 54. Springer, 2007.
[20] Chang Young Jang, Daniel Appelö, Tim Colonius, Thomas Hagstrom, and Matthew Inkman. An analysis of dispersion and dissipation properties of Hermite methods and its application to direct numerical simulation of jet noise. In *18th AIAA/CEAS Aeroacoustics Conference (33rd AIAA Aeroacoustics Conference)*, page 2240, 2012.
[21] Andreas Klöckner, Tim Warburton, Jeff Bridge, and Jan S Hesthaven. Nodal discontinuous Galerkin methods on graphics processors. *Journal of Computational Physics*, 228(21):7863–7882, 2009.
[22] P Lesaint and PA Raviart. On a finite element method for solving the neutron transport equation. *Publications mathématiques et informatiques de Rennes*, (S4):1–40, 1974.
[23] Stefano Markidis, Jing Gong, Michael Schliephake, Erwin Laure, Alistair Hart, David Henty, Katherine Heisey, and Paul Fischer. OpenACC acceleration of the nek5000 spectral element code. *International Journal of High Performance Computing Applications*, page

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press

1094342015576846, 2015.
[24] David S Medina, Amik St-Cyr, and Timothy Warburton. High-order finite-differences on multi-threaded architectures using OCCA. *arXiv preprint arXiv:1410.1387*, 2014.
[25] Jens Markus Melenk and S Sauter. Wavenumber explicit convergence analysis for Galerkin discretizations of the helmholtz equation. *SIAM Journal on Numerical Analysis*, 49(3):1210–1243, 2011.
[26] Jean-Christophe Nave, Rodolfo Ruben Rosales, and Benjamin Seibold. A gradient-augmented level set method with an optimally local, coherent advection scheme. *Journal of Computational Physics*, 229(10):3802–3827, 2010.
[27] Jianxian Qiu and Chi-Wang Shu. On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes. *Journal of Computational Physics*, 183(1):187–209, 2002.
[28] Philip J Rasch and David L Williamson. On shape-preserving interpolation and semi-Lagrangian transport. *SIAM journal on scientific and statistical computing*, 11(4):656–687, 1990.
[29] Yu-Xin Ren, Hanxin Zhang, et al. A characteristic-wise hybrid compact-WENO scheme for solving hyperbolic conservation laws. *Journal of Computational Physics*, 192(2):365–386, 2003.
[30] KV Roberts and NO Weiss. Convective difference schemes. *Mathematics of Computation*, pages 272–299, 1966.
[31] Jennifer Ryan, Chi-Wang Shu, and Harold Atkins. Extension of a postprocessing technique for the discontinuous Galerkin method for hyperbolic equations with application to an aeroacoustic problem. *SIAM Journal on Scientific Computing*, 26(3):821–843, 2005.
[32] Jennifer K Ryan and Bernardo Cockburn. Local derivative post-processing for the discontinuous Galerkin method. *Journal of Computational Physics*, 228(23):8642–8664, 2009.
[33] Benjamin Seibold, Jean-Christophe Nave, and Rodolfo Ruben Rosales. Jet schemes for advection problems. *arXiv preprint arXiv:1101.5374*, 2011.
[34] Christopher KW Tam and Jay C Webb. Dispersion-relation-preserving finite difference schemes for computational acoustics. *Journal of computational physics*, 107(2):262–281, 1993.
[35] Timothy Warburton and Thomas Hagstrom. Taming the CFL number for discontinuous Galerkin methods on structured meshes. *SIAM Journal on Numerical Analysis*, 46(6):3151–3180, 2008.
[36] David L Williamson and Philip J Rasch. Two-dimensional semi-Lagrangian transport with shape-preserving interpolation. *Monthly Weather Review*, 117(1):102–129, 1989.
[37] Olgierd Cecil Zienkiewicz and Jian Zhong Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7):1331–1364, 1992.

https://doi.org/10.4208/cicp.260915.281116a Published online by Cambridge University Press