

Hermite Leap-Frog Methods for Waves

Tom Hagstrom SMU

Major contributors to this work: Daniel Appelö, U. of Colorado
Arturo Vargas, LLNL

Original codevelopers of Hermite methods for 1st order hyperbolic systems: John Goodrich (NASA GRC), Jens Lorenz (UNM)

Other contributors: Ronald Chen (HyPerComp) - hybrid Hermite-DG, P-adaptive
Jesse Chan (Rice), Tim Warburton (Virginia Tech) - GPU implementations and alternate forms
Chang Young Jang (SMU) - dispersion/dissipation characteristics of the method
Adeline Kornelus (UNM → Arizona State) - conservative formulations and shock capturing
Tim Colonius (Caltech) - compressible turbulence simulations with applications to jet aeroacoustics

Support: NSF, ARO

Charles Hermite 1822-1901

Hermitian Matrices We solve symmetrizable hyperbolic systems, but today's focus is on second order wave equations.

Transcendence of e We round to rationals.

Hermite Polynomials We don't use them at all.

Error Formula for Polynomial Interpolation We have used it to estimate the resolution limit of our method for high order.

Hermite Interpolation THE KEY INGREDIENT IN OUR METHODS!



What sort of Hermite interpolation am I talking about - in one dimension:

Construct a polynomial of degree $2m + 1$ on the interval $[x_1, x_2]$ satisfying:

$$\begin{aligned}\frac{d^j P}{dx^j}(x_1) &= f_{1,j}, & j = 0, \dots, m \\ \frac{d^j P}{dx^j}(x_2) &= f_{2,j}, & j = 0, \dots, m\end{aligned}$$

In 3 dimensions on the cell $[x_1, x_2] \times [y_1, y_2] \times [z_1, z_2]$ construct a tensor product polynomial of degree $2m + 1$ in each coordinate (total degree $6m + 3$) satisfying (now with multi-index notation):

$$D^\alpha P(x_j, y_k, z_l) = f_{j,k,l,\alpha}, \quad 0 \leq \alpha_i \leq m.$$

Three popular myths busted by methods based on Hermite Interpolation

- High degree interpolants of nonsmooth functions must oscillate (e.g. Gibbs phenomenon)
- High order polynomial element methods must have derivative matrices which scale like the square of the degree and thus time steps must be small compared with the ideal CFL condition
- Dahlquist's Theorems imply that high order A-stable ode solvers must involve multiple nonlinear solves each step.

A special feature of piecewise Hermite interpolation as defined above is that it is a **projection** in some seminorm. In one dimension for sufficiently smooth f and some partition $x_0 < x_1 < \dots < x_N$ define

$$\mathcal{I}f = P_k, \quad x_{k-1} < x < x_k$$

where P_k is the degree $2m + 1$ Hermite interpolant defined above. Define the semi-inner-product

$$\langle f, g \rangle_{m+1} = \int_{x_0}^{x_N} \frac{d^{m+1}f}{dx^{m+1}} \cdot \frac{d^{m+1}g}{dx^{m+1}}.$$

Then

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

which implies

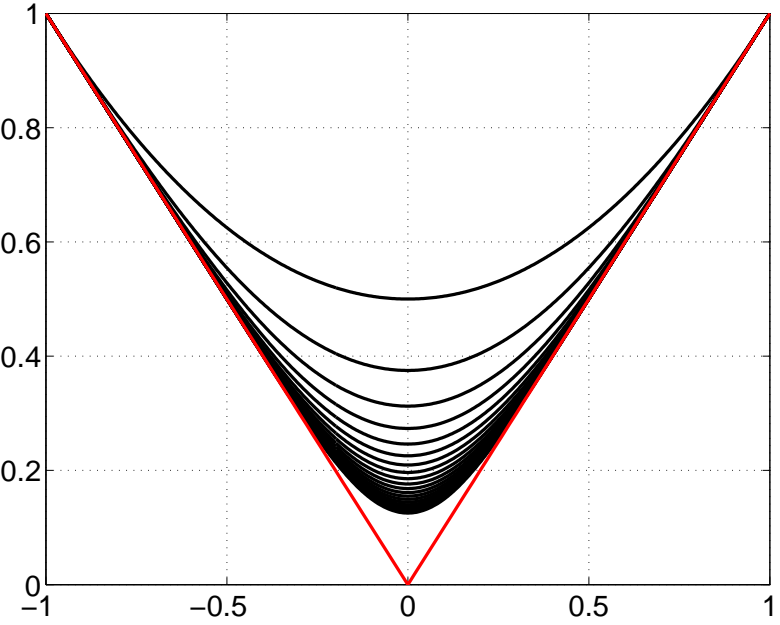
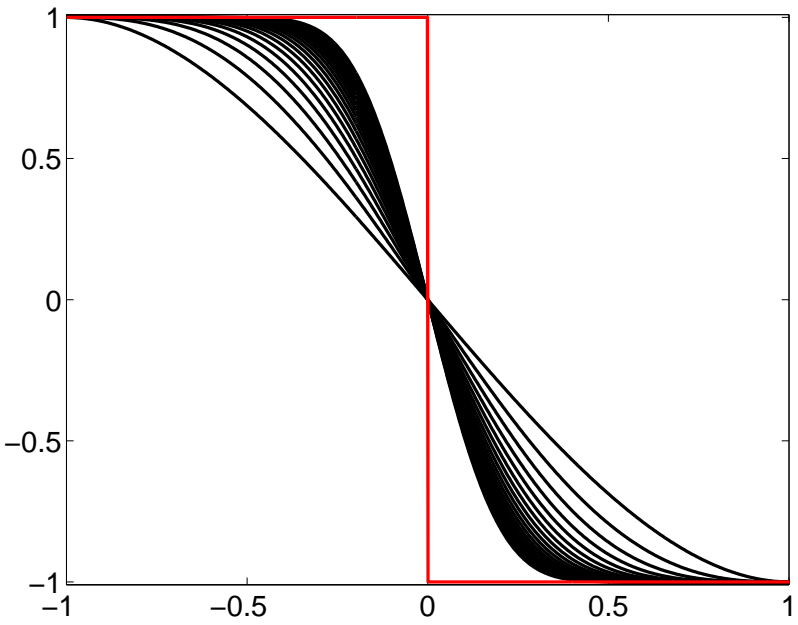
$$|f|_{m+1}^2 = |\mathcal{I}f|_{m+1}^2 + |f - \mathcal{I}f|_{m+1}^2$$

In higher dimension (e.g. 3) the same result holds with the inner product

$$\langle f, g \rangle_{m+1} = \int_{x_0}^{x_{N_1}} \int_{y_0}^{y_{N_2}} \int_{z_0}^{z_{N_3}} \frac{\partial^{3m+3} f}{\partial x^{m+1} \partial y^{m+1} \partial z^{m+1}} \frac{\partial^{3m+3} g}{\partial x^{m+1} \partial y^{m+1} \partial z^{m+1}}$$

Proof: integration by parts on each interval/cell.

This smoothing property applies to the Hermite interpolation of functions which are only piecewise smooth (so long as we don't use the singular points as nodes) - here we plot Hermite interpolating polynomials of degree $2m + 1, m = 1, \dots, 20$ of the step function $q(x) = -\text{sign}(x)$ and the absolute value function $|x|$.



Older work was on Hermite methods for first order symmetric hyperbolic systems:

$$u_t + Au_x + Bu_y + Cu_z = 0.$$

- High-order piecewise tensor-product polynomial methods using staggered rectangular/cuboidal cells and the Hermite interpolants defined above.
- Degrees-of-freedom are the degree $m \otimes \cdot \otimes m$ tensor-product Taylor polynomial at the vertices - the cell polynomial is the Hermite interpolant of these vertex polynomials.
- Each cell polynomial is evolved over a (half) time step to produce the updated Taylor polynomials at the dual cell vertices.

Basic properties proven in Goodrich, Hagstrom, Lorenz, *Math. Comp.*, 75, 2006, 595-630:

- Dissipative and of order $2m + 1$ in space - no additional filters are used beyond the inherent dissipation of the Hermite interpolation process;
- **Stable under the usual CFL restriction independent of order - i.e. if waves can't propagate from the cell boundary to the cell center in a half time step. Time step stability constraints are independent of the polynomial degree,** and thus they are at least an order of magnitude better than with standard spectral element methods (e.g. discontinuous Galerkin). Time-stepping is local to each cell - no data exchange between cells over a time step. **No matter how many stages and substeps of an RK method we use in a given cell there is no communication with neighboring cells or global stage storage.**

Leap-frog Hermite for the scalar wave equation - still use a staggered grid - the approximations $v(x, t_n)$ and $v(x, t_{n+1/2})$ are piecewise tensor-product Hermite interpolants on staggered grids. Now the evolution is given by:

$$\begin{aligned}v(\mathbf{x}, t_{n+1}) &= \mathcal{I}\mathcal{S}v(\mathbf{x}, t_{n+1/2}) - v(\mathbf{x}, t_n) \\v(\mathbf{x}, t_{n+3/2}) &= \mathcal{I}\mathcal{S}v(\mathbf{x}, t_{n+1}) - v(\mathbf{x}, t_{n+1/2})\end{aligned}$$

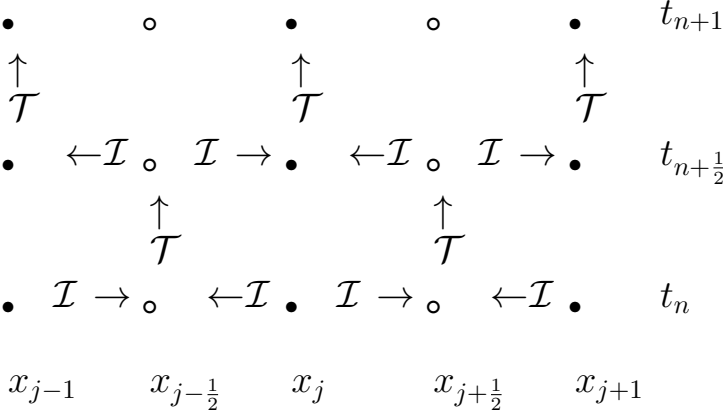
where \mathcal{S} is the **exact** leap-frog evolution operator - in Fourier space

$$\hat{\mathcal{S}} = \hat{\mathcal{S}}_+ + \hat{\mathcal{S}}_- \equiv e^{i|k|\Delta t/2} + e^{-i|k|\Delta t/2}$$

and \mathcal{I} is the Hermite interpolation operator.

How can we evaluate \mathcal{S} ? So long as the physical CFL condition is satisfied - that is so long as waves can't propagate from the cell edges to the cell center in a half time step - $\mathcal{S}v$ **at the cell center** is a polynomial in space-time. Since we only use the nodal data when we compute $\mathcal{I}\mathcal{S}v$ that's all we need!

Schematic description of the numerical process for a full time step. Solid circles represent the base mesh and open circles represent the dual mesh. \mathcal{I} is the Hermite interpolation operator and \mathcal{T} is the time evolution operator.



Stability and convergence - for standard finite element methods combined with leap-frog a great reference is Patrick's review article - Variational Methods for Time-Domain Wave Equations, Topics in Computational Wave Propagation, 201-264, Springer Lecture Notes in CSE Vol 31.

Here instead we combine energy conservation for the continuous problem with the projection property of Hermite interpolation.

Theorem: If $\Delta t < h/c$ and the solution of the wave equation is sufficiently smooth then the approximate solution produced by the leap-frog Hermite method converges at order $2m$.

Note: a similar result can be proven for Maxwell's equations and a staggered Hermite discretization:

$$\begin{aligned} E(\mathbf{x}, t_{n+1}) &= \mathcal{I}MH(\mathbf{x}, t_{n+1/2}) + E(\mathbf{x}, t_n) \\ H(\mathbf{x}, t_{n+3/2}) &= -\mathcal{I}ME(\mathbf{x}, t_{n+1}) + H(\mathbf{x}, t_{n+1/2}) \end{aligned}$$

Continuous energy for leap-frog:

Define for u a solution of the scalar wave equation:

$$U_{\pm}(\mathbf{x}, t) = u(\mathbf{x}, t) - \mathcal{S}_{\pm}u(\mathbf{x}, t - \Delta t/2)$$

Then

$$U_{\pm}(\mathbf{x}, t + \Delta t/2) = \mathcal{S}_{\mp}U_{\pm}(\mathbf{x}, t)$$

Recalling that $\hat{\mathcal{S}}_{\pm} = e^{\pm i|k|\Delta t/2}$ we see that all L^2 -based Sobolev norms of U_{\pm} are conserved.

For the approximate solution v we obtain a similar result in the seminorm for which \mathcal{I} is a projection:

$$V_{\pm}(\mathbf{x}, t_{n+1}) = \mathcal{I}\mathcal{S}_{\mp}V_{\pm}(\mathbf{x}, t_{n+1/2}) + (\mathcal{I} - 1)\mathcal{S}_{\pm}V_{\mp}(\mathbf{x}, t_{n+1/2})$$

which implies

$$|V_{+}(\cdot, t_{n+1})|_{m+1}^2 + |V_{-}(\cdot, t_{n+1})|_{m+1}^2 = |V_{+}(\cdot, t_{n+1/2})|_{m+1}^2 + |V_{-}(\cdot, t_{n+1/2})|_{m+1}^2$$

Proof outline:

1. Use the energy equalities to estimate the seminorm of the errors, $E_{\pm} = U_{\pm} - V_{\pm}$.

$$|E_{\pm}(\cdot, t)|_{m+1} = O(h^{m+1})$$

2. Use the energy estimate to prove convergence of the conserved quantities in L^2 -

$$\|E_{\pm}(\cdot, t_{n+1})\| \leq \|E_{\pm}(\cdot, t_{n+1/2})\| + O(h^{m+1}) \cdot \left(|E_{+}(\cdot, t_{n+1/2})|_{m+1} + |E_{-}(\cdot, t_{n+1/2})|_{m+1} \right)$$

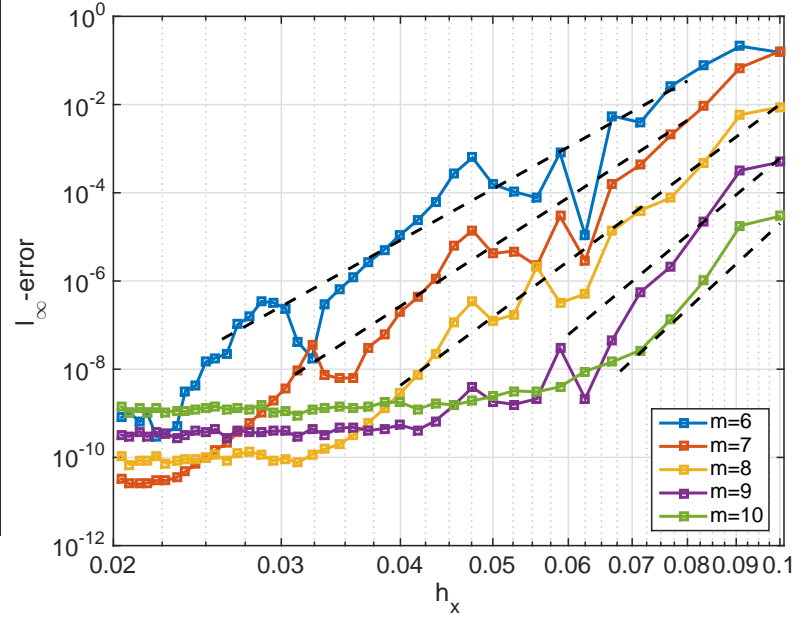
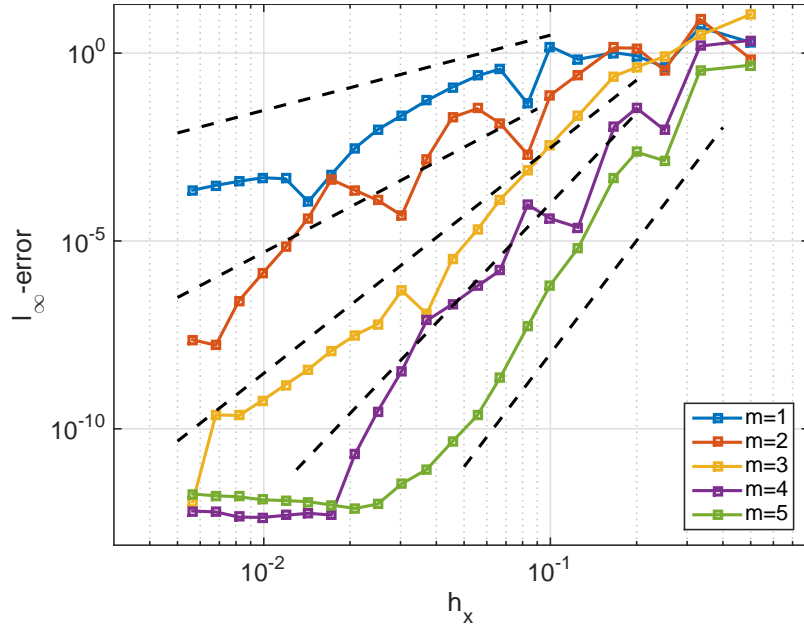
so that $\|E_{\pm}(\cdot, t_{n+1})\| = O(h^{2m+1})$.

3. Finally estimate the error itself:

$$\|e(\cdot, t_{n+1})\| \leq \|e(\cdot, t_{n+1/2})\| + \|E_{\pm}(\cdot, t_{n+1})\|$$

which yields the final result.

A simple numerical experiment - evolve $\sin(\kappa\pi x) \cdot \sin(\kappa\pi y) \cdot \sin(\sqrt{2}\kappa\pi t)$ on the unit square. $\kappa = 8$ for $m = 1, \dots, 5$ and $\kappa = 32$ for $m = 6, \dots, 10$. The straight lines correspond to the theoretical convergence rate $2m$.



One unique feature of Hermite schemes is that the time stepping is **purely local to each cell**. At high order we can take large steps without needing any intercell communication. Thus Hermite methods are good candidates for efficient implementation on many-core platforms such as gpus.

First experiments with an implementation of conservative Hermite methods on NVIDIA P100 gpus - comparison with a 36-core Broadwell CPU. Codes are written in OCCA (www.libocca.org) so that the target device (e.g. CPU using OpenMP or gpu using CUDA) can be determined at runtime. Here the grid sizes are 280^3 , 190^3 and 140^3 respectively.

m	1	2	3
GPU: Bandwidth (GB/s)	146	106	82
GPU: GFLOPS	1175	1259	1410
GPU: Time/step	.035	.052	.058
CPU: Time/step	.69	1.02	1.12

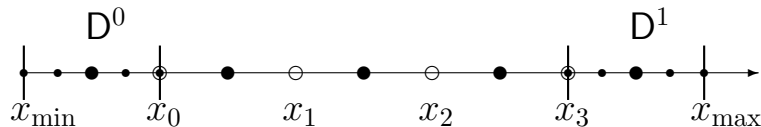
The difficulty with Hermite methods is the imposition of physical boundary conditions: one can only use mapped cuboidal elements which are inconvenient in complex geometry; **even in a mapped domain we generally use the equations to derive conditions for normal derivatives.** Recently Appelö and Henshaw have made progress on this problem.

A second possibility is to **fictitious boundaries** - a topic where Patrick has made many contributions and which is well-discussed in the review article I mentioned earlier. In that work the methods were of low order, but it may be possible to increase the order by

- i. Using higher order jump penalties - see, e.g., recent work by Stickle and Kreiss for DG methods.
- ii. Modifying the discrete evolution equation near the boundaries and correct in a neighborhood of the boundary using integral equations - see, e.g., Li and Greengard JCP 2004.

We have yet to try this with Hermite schemes.

What we have tried successfully for dissipative Hermite methods is to **use unstructured grids in the vicinity of complex geometry and discretize with a hybrid Hermite-discontinuous Galerkin method.** The method uses DG with small time steps on an unstructured mesh near physical boundaries, and Hermite on a structured mesh elsewhere. Experimentally this works, but we haven't proven stability or used it with the Leap-frog Hermite method discussed here.

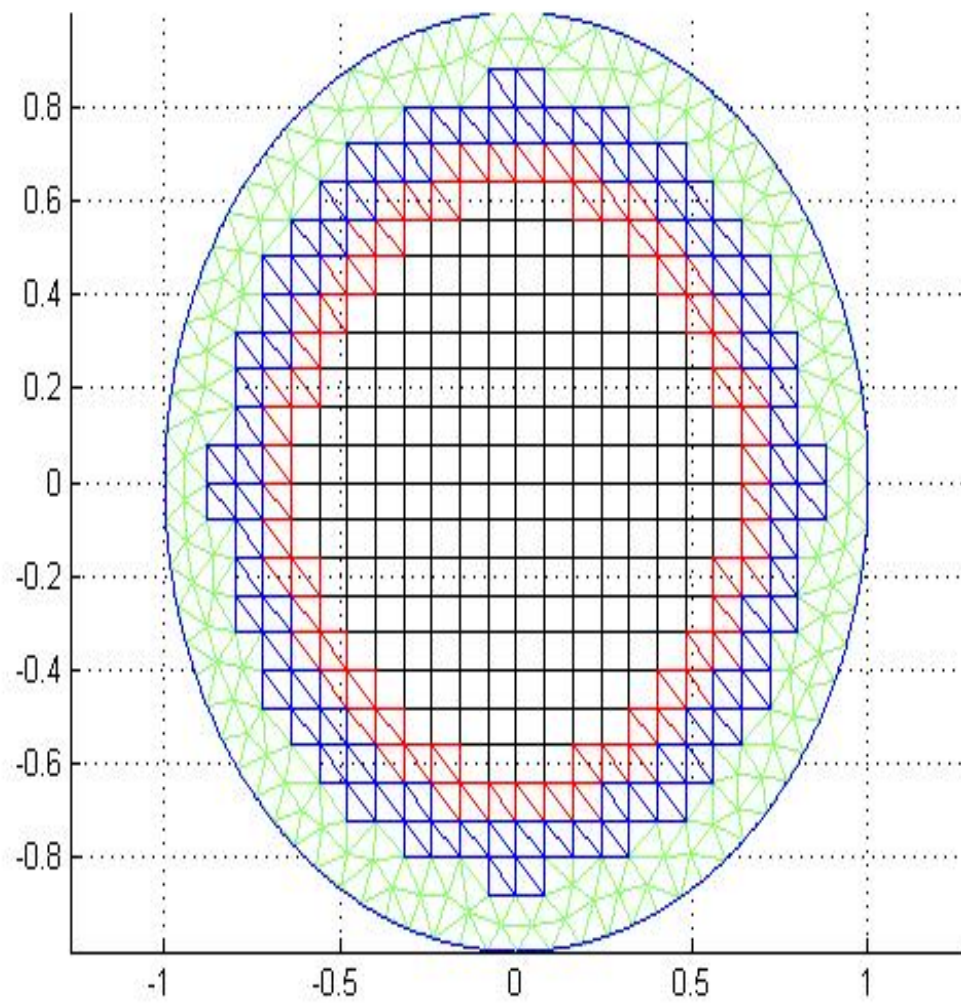


Hybrid grid in one dimension: The DG-grid $\mathcal{G}^{\text{DG}} = \{D^0, D^1\}$, consists of one element on each side of the interior Cartesian grid. The LGL nodes on the elements are denoted by small filled circles. The Cartesian grid is denoted by larger circles, the empty being the primal nodes and the filled being the dual nodes. The communication of Hermite data to the DG solver consists of constructing DG fluxes at x_0 and x_3 . The communication of DG data to the Hermite solver consists of evaluating derivatives centered at $x_{-1/2}$ and $x_{3+1/2}$ using the DG solution at the LGL nodes.

Example - evolution of a resonant mode of the unit disc:

$$\begin{aligned}
 E^z &= J_6(\alpha_6 r) \cos(6\theta) \cos(\alpha_6 t), \\
 H^x &= \frac{\sin(\alpha_6 t)}{\alpha_6} \left(\frac{6J_6(\alpha_6 r) \cos(\theta)}{r} \sin(6\theta) - \alpha_6 \sin(\theta) \cos(6\theta) \frac{(J_5(\alpha_6 r) - J_7(\alpha_6 r))}{2} \right), \\
 H^y &= \frac{\sin(\alpha_6 t)}{\alpha_6} \left(\frac{6J_6(\alpha_6 r) \sin(\theta)}{r} \sin(6\theta) - \alpha_6 \cos(\theta) \cos(6\theta) \frac{(J_5(\alpha_6 r) - J_7(\alpha_6 r))}{2} \right),
 \end{aligned}$$

and is a solution to the TM Maxwell's equation in a unit radius, cylindrical, metallic cavity. Here $J_l(z)$ is the l th Bessel function of the first kind and $\alpha_6 = 13.589290170541217$. This mode has six periods in the azimuthal direction and one "period" in the r direction.



m	h	K	CFL	Δt_{DG}	Error	rate
1	0.1	447	0.8	1.08(E-02)	3.57(E-02)	
1	0.08	666	0.8	9.03(E-03)	2.26(E-02)	2.1
1	0.05	1525	0.8	5.71(E-03)	6.09(E-03)	2.8
2	0.1	447	0.8	5.11(E-03)	3.20(E-04)	
2	0.08	666	0.8	4.29(E-03)	4.61(E-05)	8.7
2	0.05	1525	0.8	2.71(E-03)	3.92(E-06)	5.2
3	0.1	447	0.7	2.88(E-03)	6.80(E-06)	
3	0.08	666	0.7	2.42(E-03)	1.66(E-06)	6.3
3	0.05	1525	0.7	1.53(E-03)	7.21(E-08)	6.7
4	0.1	447	0.7	1.83(E-03)	1.37(E-08)	
4	0.08	666	0.7	8.56(E-04)	1.42(E-09)	10.1
4	0.05	1525	0.7	4.09(E-04)	1.68(E-11)	9.4

Maximum error at $T = 5$. Here h is the length of the sides on the square elements, K is the number of triangular element in \mathcal{G}^{DG} and \mathcal{G}^C , rate is the rate of convergence, m refers to the number of derivatives.

Conclusion: conservative Hermite methods have a number of interesting and unique properties relative to standard finite element methods for second order wave equations - we are still working on making them generally applicable in complex geometry.

For simple codes illustrating Hermite schemes (still under construction) see the Charles Hermite Interpolation Differential Equation Solver (CHIDES) - www.chides.org.

HAPPY BIRTHDAY PATRICK!