ESAIM: M2AN 46 (2012) 545–557
DOI: 10.1051/m2an/2011050

ESAIM: Mathematical Modelling and Numerical Analysis
www.esaim-m2an.org

P-ADAPTIVE HERMITE METHODS FOR INITIAL VALUE PROBLEMS*

Ronald Chen[1] and Thomas Hagstrom[2]

Abstract. We study order-adaptive implementations of Hermite methods for hyperbolic and singularly perturbed parabolic initial value problems. Exploiting the facts that Hermite methods allow the degree of the local polynomial representation to vary arbitrarily from cell to cell and that, for hyperbolic problems, each cell can be evolved independently over a time-step determined only by the cell size, a relatively straightforward method is proposed. Its utility is demonstrated on a number of model problems posed in $1+1$ and $2+1$ dimensions.

Mathematics Subject Classification. 65M70, 65M12.

Received September 29, 2009
Published online January 11, 2012.

1. Introduction

We consider the numerical solution of hyperbolic or singularly perturbed parabolic evolution equations

$$u_t = F(u, \mathcal{D}u, \epsilon\mathcal{D}^2 u, x, t) \tag{1.1}$$

in $d+1$ dimensions. Here $u(x,t) \in \mathbb{R}^m$ and $\mathcal{D}$, $\mathcal{D}^2$ denote the arrays of first and second order space derivatives. As the focus of this work is on adaptivity in space, we assume u is L-periodic in x, that is $u(x+jLe_k, t) = u(x,t)$ for any integer j and standard unit basis vector $e_k \in \mathbb{R}^d$, which we will write as $x \in \mathbb{T}^d(L)$. As recognized more than thirty years ago in the pioneering works on spectral and pseudospectral methods [9,11,16], difficult problems in wave propagation can be most efficiently treated with spectral or high-order spatial discretizations. In addition, for problems exhibiting localized pulses or sharp fronts, adaptivity in space and time is also needed. The vast majority of work on adaptive schemes, however, has focused on local mesh refinement and local time-stepping, so-called h-adaptivity. A notable exception to this is Demkowicz' and coworkers' development of hp-adaptive solvers for elliptic boundary value problems [7]. The goal of this work is to exploit the unique features of Hermite discretizations of initial-value problems to develop straightforward and, we believe, efficient purely P-adaptive methods.

Keywords and phrases. Adaptivity, high-order methods.

* *The research of the second author was supported in part by NSF Grants DMS-06010067, DMS-0929241 and ARO Grants DAAD19-03-1-0146, W911NF-09-1-0344. Any conclusions or recommendations expressed in this paper are those of the author and do not necessarily reflect the views of NSF or ARO.*

[1] Department of Mathematics and Statistics, The University of New Mexico, MSC03 2150, Albuquerque, 87131-0001 NM, USA. xqroy@math.unm.edu.

[2] Department of Mathematics, Southern Methodist University, PO Box 750156, Dallas, 75275-0156 TX, USA. thagstrom@smu.edu.

Article published by EDP Sciences

© EDP Sciences, SMAI 2012

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

We begin with a brief review of the construction and analysis of arbitrary-order Hermite methods, emphasizing the fact that large time steps can be taken in each computational cell, independent of the method order and independent of data in neighboring cells, once the Hermite interpolant of the vertex data has been computed. We also derive new results on the convergence of the Hermite interpolants in the limit of infinite degree for bandlimited functions. We then develop and test a relatively simple strategy for locally adapting the degree of the interpolants and the time-stepping algorithm. As we can treat the polynomial evolution problem independently in each cell, there is very little overhead required by the proposed technique. Indeed, the basic formulation would allow us to use degrees and time-stepping procedures in each cell limited only by the data available to construct the cell interpolant. The procedure we use constrains the degree of the data used in each cell to guarantee that the interpolation process decreases a certain seminorm. Simple numerical experiements with the transport equation and Burgers equation are presented to demonstrate the potential of our approach.

2. Hermite methods

A Hermite method in d space dimensions uses staggered computational cells of hypercubes. The standard, fixed-order method is constructed as follows. The degrees-of-freedom are the coefficients of a tensor-product degree md polynomial at each node. That is, at a node $(x_{1,k_1},\dots,x_{d,k_d}) \equiv x^{[k]}$ we approximate u by a tensor-product polynomial

$$u \approx \sum_{j_1=0}^{m} \cdots \sum_{j_d=0}^{m} c_{j_1 j_2 \dots j_d}^{k_1 k_2 \dots k_d}(t) \left(\frac{(x_1 - x_{1,k_1})}{h_1}\right)^{j_1} \cdots \left(\frac{(x_d - x_{d,k_d})}{h_d}\right)^{j_d} \tag{2.1}$$

or, using the usual multiindex notation

$$\frac{h^j}{j!} D^j u(x^{[k]}, t) \approx c_{[j]}^{[k]}(t). \tag{2.2}$$

Here h_i is the grid spacing in the ith coordinate, which we assume to be uniform. At a time step, t_n, we construct, at the midpoint of a cell, $(x_{1,k_1} + h_1/2, \dots, x_{d,k_d} + h_d/2) \equiv x^{[k+1/2]}$, the degree $(2m+1)d$ Hermite interpolant of the 2^d vertex polynomials, which we denote by $Q^{[k+1/2]}(t_n)$. We now consider the evolution problem (1.1) projected onto the degree $(2m+1)d$ tensor-product polynomials with initial data $Q^{[k+1/2]}(t_n)$:

$$\frac{dQ^{[k+1/2]}}{\mathrm{d}t} = \mathcal{T}_{2m+1} F(Q^{[k+1/2]}, \mathcal{D}Q^{[k+1/2]}, \epsilon \mathcal{D}^2 Q^{[k+1/2]}, x, t), \tag{2.3}$$

where $\mathcal{T}$ is the projection onto Taylor polynomials

$$\mathcal{T}_{2m+1} w(x,t) = \sum_{j_1=0}^{2m+1} \cdots \sum_{j_d=0}^{2m+1} \frac{D^j w(x^{[k+1/2]}, t)}{j!} (x - x^{[k+1/2]})^j. \tag{2.4}$$

Clearly, (2.3) represents a closed system of ordinary differential equations for $(2m+2)^d$ polynomial coefficients. We approximately evolve it to time $t_{n+1/2}$ using possibly multiple substeps of some single-step method of Runge-Kutta type, or, for linear autonomous problems, temporal Taylor series. On its completion we finally obtain vertex data on the dual grid by a further projection,

$$c_{[j]}^{[k+1/2]}(t_{n+1/2}) = \frac{h^j}{j!} D^j Q^{[k+1/2]}(x^{[k+1/2]}, t_{n+1/2}), \quad j_k = 0, \dots, m. \tag{2.5}$$

The process can then be repeated on the dual grid to produce $c_{[j]}^{[k]}(t_{n+1})$.

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

Note that the evolution step is completely local to each cell. Thus no global storage of stage values is needed, nor is any communication of information between cells. If large steps can be taken, which is possible for hyperbolic problems, Hermite methods are essentially optimal from the standpoint of storage and communications. We note that in this aspect Hermite methods can be viewed as arbitrary-order generalizations of central schemes [15]. However, we have not yet attempted to correctly formulate them for problems requiring shock-capturing as in [15].

2.1. Stability and convergence

A complete analysis of Hermite methods for linear hyperbolic systems is presented in [10]. Assume that (2.3) is solved exactly. For linear autonomous systems this can be done as the temporal Taylor series terminates. In other cases we assume it is solved with sufficient accuracy, which can be justified *via* the stability theorem, but we will ignore this complication here.

If the system is hyperbolic, the domain of dependence of the solution at $(x^{[k+1/2]}, t_{n+1/2})$ is contained in the region $|x - x^{[k+1/2]}| \leq \frac{c}{2}\Delta t$ where $\Delta t = t_{n+1} - t_n$ is a full time step and c is the maximum wave speed. Assuming a CFL restriction

$$c\Delta t \leq \min_i h_i \tag{2.6}$$

the solution of (2.3) evaluated at the dual grid node $x^{[k+1/2]}$ is in fact a high-order approximation to the Taylor projection of the exact solution of the evolution equation (1.1) with piecewise polynomial initial data whose restriction to cell $[k+1/2]$ is $Q^{[k+1/2]}$. If we denote that function by w we have

$$c_{[j]}^{[k+1/2]}(t_{n+1/2}) - \frac{h^j}{j!} D^j w(x^{[k+1/2]}, t_{n+1/2}) = O(h^{2m+1}\Delta t). \tag{2.7}$$

Stability follows from the fact that the Hermite interpolation process decreases a seminorm of the solution. The basic lemma, proven in [10] *via* a simple application of integration by parts, is as follows.

Lemma 2.1. *Let f,g be smooth periodic functions, $\mathcal{I}f$ the degree $(2m+1)d$ piecewise Hermite interpolant of the data $D^j f$, $0 \leq j_k \leq m$ on the nodes, $x^{[k]}$, $\mathcal{I}g$ the analogous interpolant of g, and define*

$$(f,g)_{[m+1]} = \left(\prod_{k=1}^{d} D_k^{m+1} f, \prod_{k=1}^{d} D_k^{m+1} g \right)_{L^2}. \tag{2.8}$$

Then

$$(\mathcal{I}f, g - \mathcal{I}g)_{[m+1]} = 0, \tag{2.9}$$

$$\|f\|_{[m+1]}^2 = \|\mathcal{I}f\|_{[m+1]}^2 + \|f - \mathcal{I}f\|_{[m+1]}^2. \tag{2.10}$$

Directly, we conclude that the Hermite interpolation process decreases the seminorm $\|\cdot\|_{[m+1]}$. This fundamental lemma, combined with standard estimates of interpolation error, can be directly turned into a convergence proof. Precisely, for the method described here, the proofs of Theorems 4.1 and 6.1 in [10] can be adapted to prove that the approximate solution coverges at order $2m + \frac{1}{2}$, which can be improved to order $2m + 1$ for linear, constant coefficient systems.

A point of emphasis is the fact that, under the assumption that the local evolution problem (2.3) is solved with sufficient accuracy, the outer time step is limited only by the wave speed and the cell size; it is independent of the polynomial degree. In addition, the stability restrictions for the inner time step, that is the local time steps taken within each cell, have a favorable dependence on m. For a standard spectral method, the norm of the differentiation matrix must grow like m^2, and thus the CFL constraint scales like m^{-2} [12]. Although we have shown how to reduce these norms to $O(m)$ by using dual grid filters for discontinuous Galerkin spectral elements [18], with Hermite methods no special actions are required; the differentiation matrix is always $O(m)$.

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

The reason is that we only differentiate the cell polynomial in the cell center where Bernstein's inequality implies for polynomials of degree m [5]

$$\left|\frac{\mathrm{d}p}{\mathrm{d}x}\right| \leq \frac{2m}{h}\max_{\text{cell}}|p|. \tag{2.11}$$

(We have not used this fact to formally prove the stated bounds, but we have observed them in practice).

Naturally, these results cannot strictly hold for parabolic equations, as the domain of dependence is the entire spatial domain. However, numerical experiments with the heat equation, the convection-diffusion equation, and the Navier-Stokes equations suggest [8]

$$\Delta t \propto \min\left(\frac{h_i}{c_{\max}}, \frac{h_i^2}{\epsilon m}\right). \tag{2.12}$$

Only in a shock or shear layer, where $\frac{h_i}{m} = O(\epsilon)$, is the additional restriction important. Again, the m-dependence is significantly better than for standard spectral element methods; see, *e.g.*, [17] for a discussion of discontinuous Galerkin discretizations.

2.2. *P*-convergence

The focus of the analysis in [10] is on the h-convergence of the Hermite schemes. Given that our intention here is to improve accuracy by increasing the degree rather than decreasing the cell size, it is of interest to study the convergence of the Hermite interpolation process in this limit. To that end we take $d = 1$ and assume without loss of generality that $h = 2$. The Peano kernel formula for the interpolation error is

$$f(x) - (\mathcal{I}f)(x) = \int_{-1}^{1} D^{2m+2}f(t)K_m(t,x)\mathrm{d}t, \tag{2.13}$$

where we assume $f \in C^\infty([-1,1])$. An interesting characterization of K_m follows from the recognition that $e = f - \mathcal{I}f$ satisfies the boundary value problem

$$D^{2m+2}e = D^{2m+2}f, \quad D^j e(-1) = D^j e(1) = 0, \quad j = 0,\ldots,m. \tag{2.14}$$

Thus K_m is simply the Dirichlet Green's function for D^{2m+2}. As shown in [4] upper bounds for the error are given by

$$\|e\|_{L^\infty([-1,1])} \leq \frac{1}{(2m+2)!}\|D^{2m+2}f\|_{L^\infty([-1,1])}. \tag{2.15}$$

Specializing to $f = \mathrm{e}^{\mathrm{i}\omega x}$, we can use the Hermite error formula (*e.g.* [6]):

$$\mathrm{e}^{\mathrm{i}\omega x} - (\mathcal{I}\mathrm{e}^{\mathrm{i}\omega\cdot})(x) = \frac{(x^2-1)^{m+1}}{2\pi i}\int_C \frac{\mathrm{e}^{\mathrm{i}\omega z}}{(z^2-1)^{m+1}(z-x)}\mathrm{d}z, \tag{2.16}$$

where C is a contour surrounding the real interval $[-1,1]$. Following, for example, Weideman and Trefethen [19], we consider contours defined by

$$\left|\, z^2 - 1 \,\right| = c > 1. \tag{2.17}$$

By direct computation we find that if $z = r\mathrm{e}^{\mathrm{i}\theta}$ these contours are parametrized by

$$r = \sqrt{\cos 2\theta + \sqrt{c^2 - \sin^2 2\theta}}. \tag{2.18}$$

In particular the maximum value of the imaginary part is $\sqrt{c-1}$. We thus derive the bound for ω large

$$|\mathrm{e}^{\mathrm{i}\omega x} - (\mathcal{I}\mathrm{e}^{\mathrm{i}\omega\cdot})(x)| \leq \frac{1}{2\pi\sqrt{c-1}}\mathrm{e}^{|\omega|\sqrt{c-1}-(m+1)\ln c}. \tag{2.19}$$

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

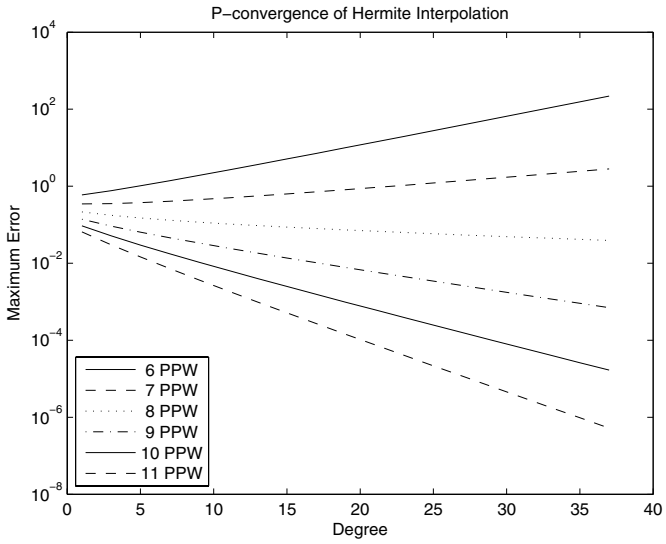


FIGURE 1. Convergence of Hermite interpolation with increasing degree. Here PPW stands for degrees-of-freedom per wavelength, as in all cases we are interpolating on a single cell.

As $|\omega|/\pi$ is the number of wavelengths and $2m+2$ is the number of degrees of freedom we write

$$|\omega| = \frac{2\pi}{\alpha}(m+1), \tag{2.20}$$

where α is the number of degrees-of-freedom per wavelength. Convergence as $m \to \infty$ is guaranteed if

$$\min_{c>1}\left(\frac{2\pi}{\alpha}\sqrt{c-1} - \ln c\right) < 0. \tag{2.21}$$

The minimum occurs when

$$\frac{\pi}{\alpha\sqrt{c-1}} = \frac{1}{c} \to \frac{\pi}{\alpha} = \frac{\sqrt{c-1}}{c}, \tag{2.22}$$

and thus we require

$$2\frac{c-1}{c} - \ln c < 0 \to c > c_0 = 4.9216\ldots \tag{2.23}$$

On this interval the formula relating α and c at the minimum, (2.22), implies that α increases with c. Thus our sufficient condition for convergence is

$$\alpha > \frac{c_0}{\sqrt{c_0-1}}\pi \approx 2.4853\pi \approx 7.8077\ldots \tag{2.24}$$

This condition is apparently sharp. In Figure 1 we display the maximum interpolation errors for the functions $e^{4\pi i(m+1)x/\alpha}$ on $[0,1]$ for $\alpha = 6-11$ and $0 \le m \le 18$, that is for all odd degree interpolants from 1 through 37. We have divergence for $\alpha = 6$ and $\alpha = 7$ and convergence for $\alpha = 8-11$. For larger values of m our implementation suffers from poor conditioning. In practice we have limited m to a maximum value of 11 in all of our implementations, that is a maximum degree of 23. (In the experiments here we take $m_{\max} = 8$).

We conclude, then, that Hermite interpolation requires almost two and one half times as many degrees-of-freedom per wavelength as corresponding methods based on Chebyshev interpolation, which require π points-per-wavelength, and in fact more than the 6 points-per-wavelength required when using equispaced nodes [19].

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

This is not surprising as we have removed sampling points from the center of the interval. However, we believe that the possibility for reduced timesteps and communications enabled by the Hermite approach compensate for this defect. It would also be of interest to compute the dispersion relation, which is more directly relevant than the interpolation error. This has been done for continuous and discontinuous Galerkin methods by Ainsworth [1,2], but it has not yet been done for Hermite schemes.

2.3. Cost

The computational costs of the Hermite methods are dominated by two operations, the construction of the interpolant and the evaluation of the right-hand side of (2.3). Using the tensor product structure, the interpolation may be carried out dimension-by-dimension. For example, if $d = 2$, the data consists of $(m+1)^2$ values at each vertex. We may interpolate along the two edges parallel to the x_1-axis, solving $2m+2$ independent one-dimensional Hermite interpolation problems associated with each power of x_2. As the cost of the one-dimensional interpolation is that of a matrix-vector multiplication, $O(m^2)$, this step requires $O(m^3)$ flops. The second step then involves an additional $2m+2$ one-dimensional interpolation problems in the x_2-direction, costing an additional $O(m^3)$ flops. Extending this argument to d-dimensions we conclude

$$C_{\text{interpolation}} = O(m^{d+1}). \tag{2.25}$$

The cost of evaluating the right-hand side of (2.3) depends on the structure of F. For a linear, constant-coefficient system the cost is linear in the number of coefficients, $O(m^d)$. For a system with product nonlinearities, the cost scales with the cost of multiplying degree $(2m+1)^d$ tensor product polynomials. The direct algorithm for accomplishing this exploiting the tensor-product structure costs $O(m^{d+1})$ flops. However, for m large FFTs can be used to reduce the cost to $O(m^d \ln m)$. For general nonlinearities we use a recursive algorithm inspired by automatic differentiation techniques [13]. We illustrate it by the example of computing

$$P = \mathcal{T}_{2m+1} \mathrm{e}^Q, \tag{2.26}$$

where Q is a degree $(2m+1)^d$ tensor product polynomial. The starting point is the differential equation

$$D_j P = \mathcal{T}_{2m+1}\left((D_j Q) P\right). \tag{2.27}$$

Taking, for example, $j = d$, (2.27) implies a recursion for the coefficients. Specializing to the node $x_{k,j_k} = 0$ and writing P and Q as

$$P = \sum_{j_d=0}^{2m+1} p_{j_d}(x_1,\ldots,x_{d-1}) \left(\frac{x_d}{h_d}\right)^{j_d}, \quad Q = \sum_{j_d=0}^{2m+1} q_{j_d}(x_1,\ldots,x_{d-1}) \left(\frac{x_d}{h_d}\right)^{j_d} \tag{2.28}$$

we have

$$j_d \cdot p_{j_d} = \sum_{j'=1}^{j_d} j' \cdot q_{j'} \cdot p_{j_d - j'}, \;\; j_d = 1,\ldots,2m+1. \tag{2.29}$$

Directly, given p_0 this allows the computation of all the polynomials p_{j_d} in $O(m^2)$ multiplications of tensor-product polynomials in dimension $d-1$. To compute p_0 we would apply the analogous recursions in one lower dimension. To begin with we compute the coefficients $p_{j_1 0\ldots 0}$ by

$$j_1 \cdot p_{j_1 0\ldots 0} = \sum_{j'=1}^{j_1} j' \cdot q_{j' 0\ldots 0} \cdot p_{j_1 - j' 0\ldots 0}, \quad p_{0\ldots 0} = \mathrm{e}^{q_{0\ldots 0}}. \tag{2.30}$$

The cost of the direct implementation of this method is dominated by (2.29) and is $O(m^{d+2})$. If the multiplications are replaced by FFTs for m large this becomes $O(m^{d+1} \ln m)$. Noting that the recursion in (2.29) has

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

a convolutional form, the algorithm of Hairer *et al.* [14] can be adapted to reduce this to $O(m^d \ln^3 m)$, though again the advantage of the fast algorithm is only likely to be felt for m rather large. Griewank [13] has shown that all of the standard transcendental functions can be evaluated in this way. Letting s be the total number of evaluations of F required to advance (2.3) to the next time level we have a total cost per degree-of-freedom depending on the structure of F and the algorithm employed given by

$$C_{\text{total}} = \begin{cases} O(s) + O(m), & \text{linear constant coefficient} \\ O(sm) \leftrightarrow O(s \ln m) + O(m), & \text{product nonlinearity} \\ O(sm^2) \leftrightarrow O(sm \ln m) \leftrightarrow O(s \ln^3 m) + O(m), & \text{general nonlinearity.} \end{cases} \tag{2.31}$$

We note that an alternative pseudospectral approach to evaluating F is possible, but we have not yet tried it.

3. Adaptive implementation in 1 + 1 dimensions

We now set $d = 1$ and consider the implementation of an order adaptive strategy. Functionally, we simply admit the possibility that at the current time level t_n the coefficient data at the node x_k extends to some previously-determined degree m_k. At $x_{k+1/2}$ we then could compute the Hermite interpolant of the degree m_k polynomial at x_k and the degree m_{k+1} polynomial at x_{k+1} which would have degree $m_k + m_{k+1} + 1$. However, we instead set

$$\bar{m}_{k+1/2} = \min\{m_k, m_{k+1}\} \tag{3.1}$$

and construct the degree $2\bar{m}_{k+1/2} + 1$ Hermite interpolant of the function values and derivatives through order $\bar{m}_{k+1/2}$ at each node. Directly this means that we use all of the available data from the node where the polynomial is of lower degree and ignore the highest degree coefficients at the other. The motivation for this choice is Lemma 2.1, which we realize can be applied cell-by-cell. It implies that the local cell interpolant satisfies

$$\int_{x_k}^{x_{k+1}} \left(D^{\bar{m}_{k+1/2}+1} Q^{[k+1/2]}(x) \right)^2 \mathrm{d}x \leq \int_{x_k}^{x_{k+1}} \left(D^{\bar{m}_{k+1/2}+1} f(x) \right)^2 \mathrm{d}x \tag{3.2}$$

for any function f whose derivatives through order $\bar{m}_{k+1/2}$ agree with the nodal data. We now evolve this polynomial as in the nonadaptive case using (2.3).

To adaptively choose the degree we consider the truncation step (2.5). The full polynomial has degree $2\bar{m}_{k+1/2} + 1$, roughly double what is being carried on the nodes. We simply truncate at whatever order is suggested by the tolerance, τ. That is, find the smallest $m_{k+1/2}$ such that

$$\max_{j > m_{k+1/2}} \frac{h^j}{j!} \left| D^j Q^{[k+1/2]}(x^{[k+1/2]}, t_{n+1/2}) \right| < \tau. \tag{3.3}$$

(We also place a limit on the global maximum order). We impose the same strategy when marching from $t_{n+1/2}$ to t_{n+1}.

3.1. Application to the transport equation

As a first test of the method we solve the transport equation,

$$u_t = u_x, \quad u(x, 0) = \mathrm{e}^{-x^2}, \tag{3.4}$$

imposing periodic boundary conditions on the interval $[-10, 10]$ and solving over two periods, $T = 40$. We take $h = .25$ and $\Delta t = 0.9h$. The time-stepping is *via* Taylor series and is carried out to match the local spatial order. That is, as in [10], we use an order $2\bar{m}_{k+1/2} + 1$ temporal Taylor series. Note that this time-stepping procedure is completely local to each cell, so there is no issue in choosing different temporal orders in different cells. The Taylor method could easily be replaced by multiple substeps of some other Runge–Kutta formula as in [3].

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

TABLE 1. Errors and orders for various tolerances: transport equation in $1+1$ dimensions.

τ	Maximum error	$m_{\max}$	m_{ave}
10^{-3}	1.221×10^{-5}	4	1.556
10^{-4}	5.767×10^{-7}	5	1.914
10^{-5}	1.132×10^{-7}	6	2.333
10^{-6}	2.588×10^{-8}	7	2.778
10^{-7}	7.394×10^{-9}	8	3.420
10^{-8}	4.373×10^{-10}	8	3.803
10^{-9}	3.022×10^{-11}	8	4.049

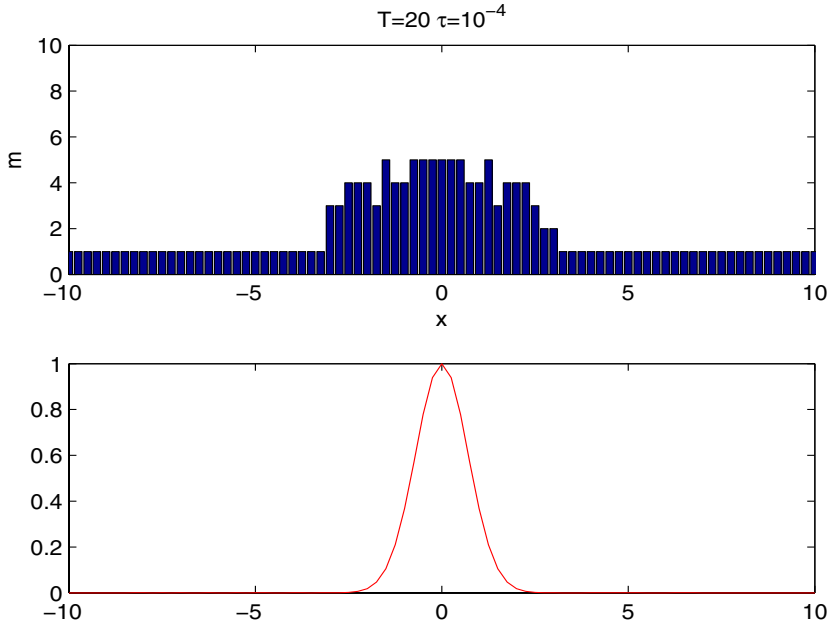


FIGURE 2. Solution and degree for the transport equation in $1+1$ dimensions.

Again as the local evolution problems are independent, the size of the substeps can be chosen independently in each cell. We set the maximum value of m to be 8 and the minimum to be 1 so that the local method order varies between 3 ($m = 1$) and 17. At $t = 0$, m_k is chosen based on the interpolation of the initial condition. Varying the tolerance, τ, between 10^{-3} and 10^{-9} we obtain the results summarized in Table 1. See also Figure 2 for a plot of the solution and m_k when $\tau = 10^{-4}$.

We see that the actual error in each case is more than an order of magnitude less than the tolerance, indicating that a less stringent cutoff criterion could be used. For the coarsest tolerance the average density of degrees-of-freedom is almost 2.7 times less than the maximum. Exponential convergence is observed as a decrease in the error of a factor of 10 requires roughly a fixed increase in the number of degrees-of-freedom over the range of tolerances considered.

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

TABLE 2. Comparison of Fourier pseudospectral and adaptive order Hermite solutions for various tolerances and viscosities: Burgers equation in $1+1$ dimensions.

ϵ	t	τ	$\|u_{\text{adapt}} - u_{PS}\|_\infty$	$m_{\max}$	m_{ave}
10^{-2}	1	10^{-3}	8.32×10^{-5}	7	1.32
10^{-2}	2	10^{-3}	1.80×10^{-3}	8	1.23
10^{-2}	1	10^{-6}	2.99×10^{-10}	8	3.19
10^{-2}	2	10^{-6}	6.09×10^{-6}	8	2.45
10^{-4}	1	10^{-3}	9.0×10^{-3}	8	1.43
10^{-4}	2	10^{-3}	3.0×10^{-2}	8	1.38
10^{-4}	1	10^{-6}	9.2×10^{-3}	8	3.22
10^{-4}	2	10^{-6}	3.2×10^{-2}	8	2.62

3.2. Application to Burgers equation

As a second example we solve Burgers equation

$$u_t + uu_x = \epsilon u_{xx}, \quad u(x,0) = \sin x \tag{3.5}$$

on $[0, 2\pi]$, $0 \le t \le 2$. Note that the shock formation time for the inviscid equations is $t = 1$. We choose a uniform grid with $h = \pi/80$, set the maximum value of m to be 8 (maximum order 17) and solve for $\epsilon = 10^{-2}, 10^{-4}$ with $\tau = 10^{-3}, 10^{-6}$. As the time step restriction is now proportional to $h^2/(m\epsilon)$ we chose $\Delta t = h/50$, though for $\epsilon = 10^{-4}$ we could take larger steps. If we define an effective cell Reynolds number based on the finest degree-of-freedom density allowed

$$R_c = u_{\max} \frac{h}{\epsilon \cdot (m_{\max} + 1)} \tag{3.6}$$

we compute a value of .44 when $\epsilon = 10^{-2}$ and 44 when $\epsilon = 10^{-4}$. Thus the latter case is certainly underresolved and we do not expect to achieve the error tolerances. However we will achieve reasonable accuracy and our solutions will not display spurious oscillations. This demonstrates the robustness of the *P*-adaptive Hermite methods. Although, based on the results from Section 1.2 we may expect that adaptive implementations of standard spectral element methods could require a little less resolution in the shock layer, the advantage of the proposed scheme is that there is essentially no overhead.

Approximate error data is generated by comparing the solutions to those computed with a Fourier pseudospectral method in space [11] evolved in time with Matlab's ode45 routine. Absolute and relative error tolerances for ode45 were set at 10^{-9}. With $\epsilon = 10^{-2}$ we used 1280 points for the pseudospectral method and 40 960 with $\epsilon = 10^{-4}$. To verify the accuracy of the pseudospectral computations we repeated them with twice as many grid points and the tolerances for the ode solver reduced to 10^{-11}. These indicate that the pseudospectral solutions have errors below 10^{-9} at the times indicated except for $t = 2$ and $\epsilon = 10^{-4}$, when a maximum norm difference of 10^{-4} was recorded. As this is orders-of-magnitude smaller than the Hermite error in that case we deem the error data to be reliable.

Details of the results are presented in Table 2 for both the approximate shock formation time, $t = 1.0016$, and the final time, $t = 2$. Decreasing the tolerance by a factor of 10^{-3} increases the number of degrees of freedom by less than a factor of 3. The error tolerances were approximately achieved in the case of $\epsilon = 10^{-2}$. For $\epsilon = 10^{-4}$ larger errors persist due to the limits placed on R_c. However, with the adaptive strategy we do avoid oscillations at the shock, as is readily apparent in the graphs. (See Figs. 3–6).

We also plot the solution and method order at $t = 1$ and $t = 2$ for $\epsilon = 10^{-2}$ and $\epsilon = 10^{-4}$ computed with $\tau = 10^{-3}$.

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

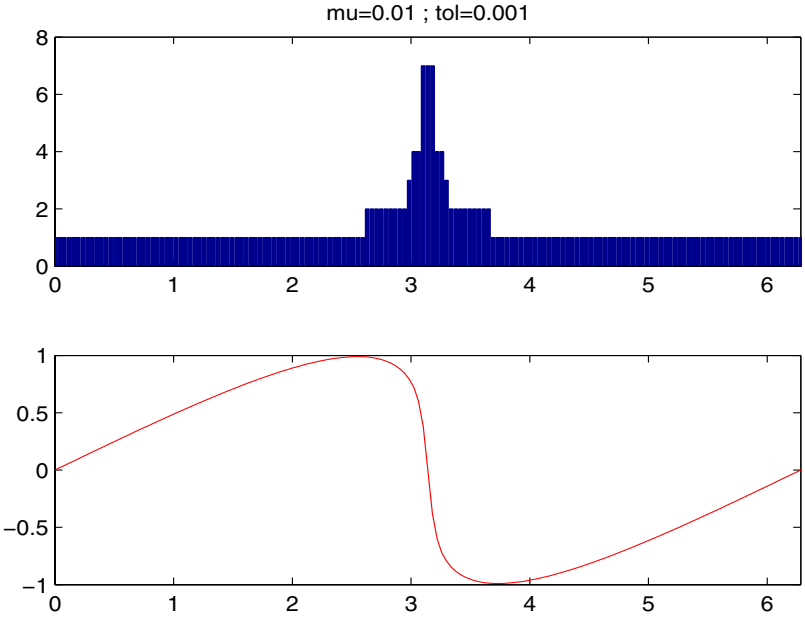


FIGURE 3. Solution and degree at $t = 1$ for Burgers equation in $1 + 1$ dimensions.

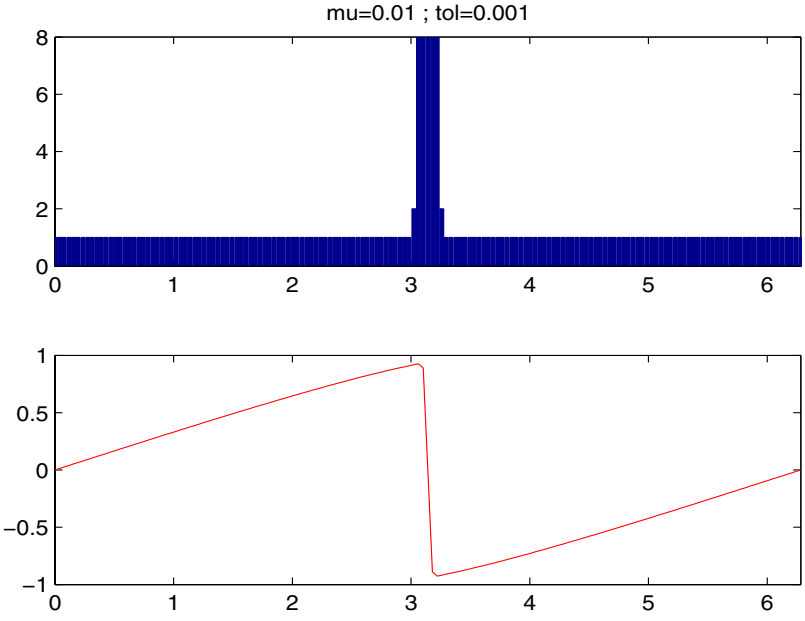


FIGURE 4. Solution and degree at $t = 2$ for Burgers equation in $1 + 1$ dimensions.

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

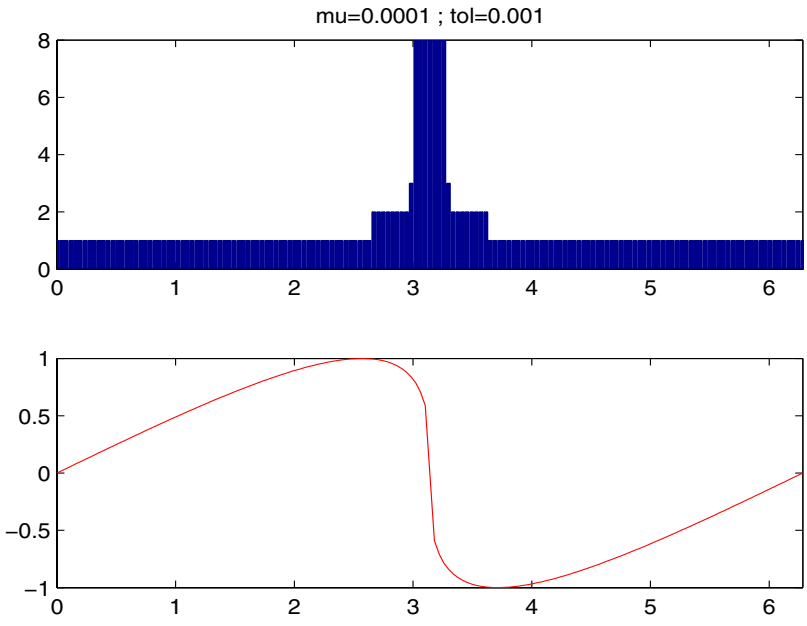


FIGURE 5. Solution and degree at $t = 1$ for Burgers equation in $1 + 1$ dimensions.

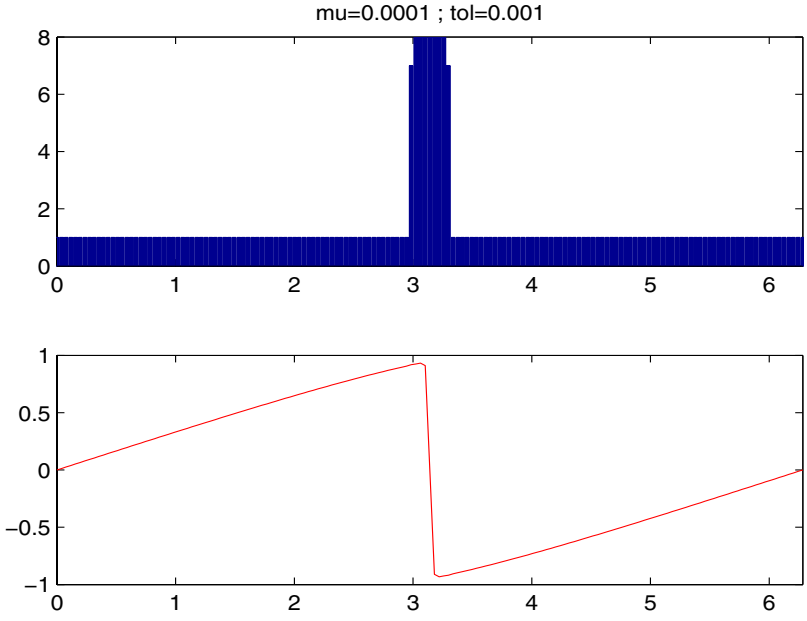


FIGURE 6. Solution and degree at $t = 2$ for Burgers equation in $1 + 1$ dimensions.

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

4. Extensions to 2 space dimensions

The interpolation technique used above has a natural extension to any number of space dimensions. For simplicity we restrict our attention here to the case $d = 2$. We simply define

$$\bar{m}_{x,[i+1/2,j+1/2]} = \min\left(m_{x,[i,j]}, m_{x,[i,j+1]}, m_{x,[i+1,j]}, m_{x,[i+1,j+1]}\right), \tag{4.1}$$

$$\bar{m}_{y,[i+1/2,j+1/2]} = \min\left(m_{y,[i,j]}, m_{y,[i,j+1]}, m_{y,[i+1,j]}, m_{y,[i+1,j+1]}\right), \tag{4.2}$$

and compute the tensor-product Hermite interpolant, $Q^{[i+1/2,j+1/2]}$ of the vertex data using mixed derivatives up through order $\bar{m}_{x,[i+1/2,j+1/2]}$ in x and up through order $\bar{m}_{y,[i+1/2,j+1/2]}$ in y. This will result in a cell polynomial of degree $2\bar{m}_{x,[i+1/2,j+1/2]}+1$ in x and $2\bar{m}_{y,[i+1/2,j+1/2]}+1$ in y. The interpolation is a stabilizing step in that Lemma 2.1 applied in both variables implies the analogue of (3.2):

$$\begin{aligned}\int_{x_i}^{x_{i+1}}\int_{y_j}^{y_{j+1}} \left(D_x^{\bar{m}_{x,[i+1/2,j+1/2]}+1} D_y^{\bar{m}_{y,[i+1/2,j+1/2]}+1} Q^{[i+1/2,j+1/2]}(x,y)\right)^2 \mathrm{d}x\mathrm{d}y \\ \leq \int_{x_i}^{x_{i+1}}\int_{y_j}^{y_{j+1}} \left(D_{x,\bar{m}_x^{[i+1/2,j+1/2]}+1} D_y^{\bar{m}_{y,[i+1/2,j+1/2]}+1} f(x,y)\right)^2 \mathrm{d}x\mathrm{d}y\end{aligned} \tag{4.3}$$

for any function f whose mixed derivatives through order $\bar{m}_{x,[i+1/2,j+1/2]}$ in x and order $\bar{m}_{y,[i+1/2,j+1/2]}$ in y agree with the nodal data.

Having computed the cell interpolant, we evolve the data using (2.3). To truncate we seek to satisfy the analogue of (3.3) with the smallest values of $m_{x,[i+1/2,j+1/2]}$, $m_{y,[i+1/2,j+1/2]}$

$$\max_{j_1 > m_{x,[i+1/2,j+1/2]}} \frac{h_1^{j_1}}{j_1!}\frac{h_2^{j_2}}{j_2!} \left| D_x^{j_1} D_y^{j_2} Q^{[i+1/2,j+1/2]}(x_{i+1/2}, y_{j+1/2}, t_{n+1/2}) \right| < \tau, \tag{4.4}$$

$$\max_{j_2 > m_{y,[i+1/2,j+1/2]}} \frac{h_1^{j_1}}{j_1!}\frac{h_2^{j_2}}{j_2!} \left| D_x^{j_1} D_y^{j_2} Q^{[i+1/2,j+1/2]}(x_{i+1/2}, y_{j+1/2}, t_{n+1/2}) \right| < \tau. \tag{4.5}$$

4.1. Application to the transport equation

As a first demonstration of the method in $2+1$ dimensions we have solved the transport equation

$$u_t + \cos\theta \cdot u_x + \sin\theta \cdot u_y = 0, \quad (x,y) \in [-25,25] \times [-25,25], \tag{4.6}$$

with $\theta = \frac{\pi}{3}$, periodic boundary conditions, and initial data consisting of a Gaussian

$$u(x,y,0) = \mathrm{e}^{-(x^2+y^2)}. \tag{4.7}$$

We solved up to $t = 100$ using the adaptive method described above with a maximum value of $m_{\max} = 8$ (17th order) and tolerances of 10^{-3},10^{-5}, 10^{-7}, and 10^{-9}. The mesh width was $h = \frac{5}{8}$, $\Delta t \approx .8h$.

The results are summarized in Table 3. Generally, the L^2 error is at or below the desired tolerance, though the maximum error is quite a bit larger. That said, the method is seen to be very efficient. Even with a tolerance of 10^{-9}, which produced a maximum error of approximately 10^{-6}, the total number of degrees-of-freedom, as indicated by m_{ave}, exceeds those of the first order method on the same grid by a mere 28%.

5. Conclusions

We believe this work demonstrates the potential of P-adaptive implementations of Hermite methods. However, a number of additional developments are needed to improve its reliability and efficiency. These include:

(i) analysis of stability;

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press

TABLE 3. Errors and orders at $t = 100$ for various tolerances: transport equation in $2 + 1$ dimensions.

τ	L^2 Error	Maximum error	$m_{\max}$	m_{ave}
10^{-3}	3.10×10^{-4}	1.51×10^{-1}	8	0.06
10^{-5}	3.24×10^{-6}	1.80×10^{-3}	8	0.14
10^{-7}	6.07×10^{-8}	1.52×10^{-5}	8	0.20
10^{-9}	1.42×10^{-9}	1.04×10^{-6}	8	0.28

(ii) justification of the choice of the cutoff criterion (3.3) in terms of truncation error analysis;
(iv) combination with h-refinement and local time stepping when P-refinement fails to achieve the desired accuracy.

We hope to treat these issues in subsequent work.

Acknowledgements. David Gottlieb was a pioneer in the development and application of high-order methods to difficult problems in scientific computing. His influence can be seen throughout our work, and his absence has cast a shadow over the entire community. Beyond his accomplishments in mathematics, he was a true gentleman whose kindness and encouragement touched many of us. Thus it is an honor to contribute to this volume dedicated to his memory, and to acknowledge his many contributions, both scientific and personal, to the advancement of our field.

REFERENCES

[1] M. Ainsworth, Discrete dispersion relation for hp-version finite element approximation at high wave number. *SIAM J. Numer. Anal.* **42** (2004) 553–575.
[2] M. Ainsworth, Dispersive and dissipative behavior of high-order discontinuous Galerkin finite element methods. *J. Comput. Phys.* **198** (2004) 106–130.
[3] D. Appelö and T. Hagstrom, Experiments with Hermite methods for simulating compressible flows: Runge-Kutta time-stepping and absorbing layers, in *13th AIAA/CEAS Aeroacoustics Conference*. AIAA (2007).
[4] G. Birkhoff, M. Schultz and R. Varga, Piecewise Hermite interpolation in one and two variables with applications to partial differential equations. *Numer. Math.* **11** (1968) 232–256.
[5] P. Borwein and T. Erdélyi, *Polynomials and Polynomial Inequalities*. Springer-Verlag, New York (1995).
[6] P. Davis, *Interpolation and Approximation*. Dover Publications, New York (1975).
[7] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszynski, W. Rachowicz and A. Zdunek, *Computing with hp-Adaptive Finite Elements*. Applied Mathematics & Nonlinear Science, Chapman & Hall/CRC, Boca Raton (2007).
[8] C. Dodson, *A high-order Hermite compressible Navier-Stokes solver*. Master's thesis, The University of New Mexico (2003).
[9] B. Fornberg, On a Fourier method for the integration of hyperbolic equations. *SIAM J. Numer. Anal.* **12** (1975) 509–528.
[10] J. Goodrich, T. Hagstrom and J. Lorenz, Hermite methods for hyperbolic initial-boundary value problems. *Math. Comput.* **75** (2006) 595–630.
[11] D. Gottlieb and S.A. Orszag, *Numerical Analysis of Spectral Methods*. SIAM, Philadelphia (1977).
[12] D. Gottlieb and E. Tadmor, The CFL condition for spectral approximations to hyperbolic initial-boundary value problems. *Math. Comput.* **56** (1991) 565–588.
[13] A. Griewank, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia (2000).
[14] E. Hairer, C. Lubich and M. Schlichte, Fast numerical solution of nonlinear Volterra convolutional equations. *SIAM J. Sci. Statist. Comput.* **6** (1985) 532–541.
[15] G.-S. Jiang and E. Tadmor, Nonoscillatory central schemes for multidimensional hyperbolic conservation laws. *SIAM J. Sci. Comput.* **19** (1998) 1892–1917.
[16] H.-O. Kreiss and J. Oliger, Comparison of accurate methods for the integration of hyperbolic equations. *Tellus* **24** (1972) 199–215.
[17] F. Lörcher, G. Gassner and C.-D. Munz, An explicit discontinuous Galerkin scheme with local time-stepping for general unsteady diffusion equations. *J. Comput. Phys.* **227** (2008) 5649–5670.
[18] T. Warburton and T. Hagstrom, Taming the CFL number for discontinuous Galerkin methods on structured meshes. *SIAM J. Numer. Anal.* **46** (2008) 3151–3180.
[19] J. Weideman and L. Trefethen, The eigenvalues of second-order differentiation matrices. *SIAM J. Numer. Anal.* **25** (1988) 1279–1298.

https://doi.org/10.1051/m2an/2011050 Published online by Cambridge University Press