Hermite Methods in Time

Rujie Gu and Thomas Hagstrom

1 Introduction

Over the past decade a number of works have appeared which exploit the unique properties of Hermite-Birkhoff interpolation in space to construct arbitrary-order discretization methods for hyperbolic [1, 2, 4–7, 10, 14, 16–19] as well as Schrödinger [3] equations. The precise form of the interpolant in a single cell, which here we write in one dimension labelled t , is

$$u(t) \approx \mathcal{J}u(t) \in \Pi^{2m+1}, \quad t \in (t_{j-1}, t_j), \quad (1)$$

$$\frac{d^k}{dt^k} \mathcal{J}u(t_\ell) = \frac{d^k u}{dt^k}(t_\ell); \quad k = 0, \dots, m, \quad \ell = j-1, j, \quad (2)$$

where Π^{2m+1} denotes the polynomials of degree $2m + 1$. (In higher dimensions one uses a tensor-product cell interpolant based on vertex data consisting of mixed derivatives of order through m in each Cartesian coordinate.)

In contrast, there has been little work on analogous methods for time discretization. A recent exception is the manuscript by Liu et al. [15]. They develop methods for second-order semilinear hyperbolic equations using interpolants of the form (1)–(2) combined with a reformulation of the evolution problem using exact solutions of the linear part. They demonstrate excellent long-time performance.

The outline of the paper is as follows. In Sect. 2 we list a few properties of piecewise Hermite-Birkhoff interpolation. In Sect. 3 we construct the time-stepping

R. Gu · T. Hagstrom (✉)
Southern Methodist University, Dallas, TX, USA
e-mail: rujieg@smu.edu; thagstrom@smu.edu

schemes and establish some basic results, with a few numerical experiments described in Sect. 4.

2 Basic Properties of Hermite-Birkhoff Interpolation

Hermite interpolants have a number of interesting properties which make them very attractive for the solution of differential equations; see, e.g., [2]. Here we will mainly use the simplest. Precisely, for $t \in (t_{j-1}, t_j)$, the Peano representation of the local error can be easily derived by noting that $e = u - \mathcal{I}u$ solves the two point boundary value problem

$$\frac{d^{2m+2}e}{dt^{2m+2}} = \frac{d^{2m+2}u}{dt^{2m+2}}, \quad \frac{d^k e}{dt^k} = 0, \quad t = t_{j-1}, t_j, \quad k = 0, \dots, m. \quad (3)$$

Thus

$$e(t) = \int_{t_{j-1}}^{t_j} K_j(t, s) \frac{d^{2m+2}u}{dt^{2m+2}}(s) ds, \quad (4)$$

where the kernel K_j is the Green's function for (3). Simple scaling arguments combined with the transformation $t = t_{j-1} + zh_j$ then show that $e = O(h_j^{2m+2})$ where $h_j = t_j - t_{j-1}$ is the time step. A fundamental feature of piecewise Hermite interpolation is the following orthogonality property. For **any** functions $v(t), w(t)$

$$\int_{t_{j-1}}^{t_j} \frac{d^{m+1} \mathcal{I}v}{dt^{m+1}}(t) \cdot \frac{d^{m+1} (w - \mathcal{I}w)}{dt^{m+1}}(t) dt = 0, \quad (5)$$

which in particular implies that interpolation reduces the H^{m+1} seminorm.

3 Time-Stepping Methods

We begin by considering the initial value problem for a first-order system ordinary differential equations:

$$\frac{du}{dt} = f(u, t), \quad u(t_0) = u_0, \quad u(t) \in \mathbb{R}^d. \quad (6)$$

Given a discrete time sequence $t_j > t_{j-1}$, $j = 1, \dots, N$, with time steps $h_j = t_j - t_{j-1}$ we write down the Picard integral formulation of the time evolution over

a single step

$$u(t_j) = u(t_{j-1}) + \int_{t_{j-1}}^{t_j} f(u(s), s) ds. \quad (7)$$

The construction of our time integration formula proceeds in three steps. We denote by v_j the approximation to $u(t_j)$.

1. Given v_{j-1} and assuming for the moment that v_j is known, use the differential equation to compute m scaled derivatives of its solution, $V_\ell(t)$, satisfying $V_\ell(t_\ell) = v_\ell$, $\ell = j, j-1$. Setting

$$F_\ell(t) = f(V_\ell(t), t), \quad (8)$$

these are recursively defined by the formula

$$\frac{d^k V_\ell}{dt^k}(t_\ell) = \frac{d^{k-1} F_\ell}{dt^{k-1}}(t_\ell), \quad k = 0, \dots, m. \quad (9)$$

2. Construct the Hermite-Birkhoff interpolant of this data; that is the polynomial $P_{j-1/2}(t; v_{j-1}, v_j)$ of degree $2m+1$, satisfying

$$\frac{d^k P_{j-1/2}}{dt^k}(t_\ell; v_{j-1}, v_j) = \frac{d^k V_\ell}{dt^k}(t_\ell); \quad \ell = j-1, j, \quad k = 0, \dots, m.. \quad (10)$$

3. Approximate (7) by replacing $u(t_\ell)$ by v_ℓ and replacing the integral by a $q+1$ -point quadrature rule with f evaluated at the Hermite interpolant:

$$v_j = v_{j-1} + h_j \sum_{k=0}^q w_k f(P_{j-1/2}(t_{j,k}; v_{j-1}, v_j)). \quad (11)$$

4. Solve (11) for v_j . Note that this is a system of d nonlinear equations for any m ; that is, unlike standard implicit Runge–Kutta methods, the size of the nonlinear system is independent of the order.

We remark that we have not studied in detail the unique solvability of (11) in the stiff case. In our numerical experiments we used the solution at the current time step as an initial approximation for Newton iterations and simply accepted the solution to which the iterates converged.

To emphasize the ideas we write down some specific examples of methods with $m = 1$ and $m = 2$ making the simplifying assumption of autonomy; that is $f = f(u)$. The derivation of methods of arbitrary order is straightforward and the formulas can be trivially obtained using software capable of symbolic computations. To apply them at higher order one must evaluate higher derivatives of f , which is also possible using automatic differentiation tools [11].

Example ($m = 1$) Set

$$\tau = \frac{t - t_{j-1}}{h_j}.$$

Now the interpolant $P_{j-1/2}(t; v_{j-1}, v_j)$ is given by:

$$P_{j-1/2}(t; v_{j-1}, v_j) = \sum_{k=0}^3 a_k \tau^k, \quad (12)$$

where

$$\begin{aligned} a_0 &= v_{j-1}, \quad a_1 = h_j f(v_{j-1}), \\ a_2 &= 3(v_j - v_{j-1}) - h_j (2f(v_{j-1}) + f(v_j)), \\ a_3 &= -2(v_j - v_{j-1}) + h_j (f(v_{j-1}) + f(v_j)). \end{aligned} \quad (13)$$

We next introduce a quadrature rule which is exact for polynomials of degree 3. Possible choices include the 2-point Gauss-Legendre (14) rules, or the 3-point Gauss-Radau (15) or Gauss-Lobatto rules. Note that by using two different rules we obtain a possible error indicator. Here are the two different methods used below. Note that the methods are identical if f is linear.

$$v_j = v_{j-1} + \frac{h_j}{2} \left(f(P_{j-1/2}(\alpha_-; v_{j-1}, v_j)) + f(P_{j-1/2}(\alpha_+; v_{j-1}, v_j)) \right), \quad (14)$$

$$\begin{aligned} v_j &= v_{j-1} + \frac{h_j}{36} \left(\beta_+ f(P_{j-1/2}(\gamma_-; v_{j-1}, v_j)) + \beta_- f(P_{j-1/2}(\gamma_+; v_{j-1}, v_j)) \right. \\ &\quad \left. + 4f(v_j) \right), \end{aligned} \quad (15)$$

$$\alpha_{\pm} = \left(1 \pm \frac{1}{\sqrt{3}} \right), \quad \gamma_{\pm} = \frac{4 \pm \sqrt{6}}{10}, \quad \beta_{\pm} = 16 \pm \sqrt{6}. \quad (16)$$

A time step is executed by solving the nonlinear system, (14) or (15), for v_j .

Example ($m = 2$) Now we also need the second time derivative of u ,

$$\frac{d^2 u}{dt^2} = \frac{d}{dt} f(u) = J(u) f(u), \quad (17)$$

where $J(u)$ is the Jacobian derivative. The Hermite interpolant can now be written:

$$P_{j-1/2}(t; v_{j-1}, v_j) = \sum_{k=0}^5 a_k \tau^k, \quad (18)$$

where

$$\begin{aligned} a_0 &= v_{j-1}, \quad a_1 = h_j f(v_{j-1}), \quad a_2 = \frac{h_j^2}{2} J(v_{j-1}) f(v_{j-1}), \\ a_3 &= 10(v_j - v_{j-1}) - h_j(6f(v_{j-1}) + 4f(v_j)) + \frac{h_j^2}{2}(-3J(v_{j-1})f(v_{j-1}) + J(v_j)f(v_j)), \\ a_4 &= -15(v_j - v_{j-1}) + h_j(8f(v_{j-1}) + 7f(v_j)) + \frac{h_j^2}{2}(3J(v_{j-1})f(v_{j-1}) - J(v_j)f(v_j)), \\ a_5 &= 6(v_j - v_{j-1}) - 3h_j(f(v_{j-1}) + f(v_j)) + \frac{h_j^2}{2}(-J(v_{j-1})f(v_{j-1}) + J(v_j)f(v_j)). \end{aligned} \quad (19)$$

Again we can now use, for example, the 3-point Gauss-Legendre or 4-point Gauss-Radau quadrature rules to produce the equation we must solve for v_j .

3.1 Stability and Consistency

The consistency of the method is a straightforward consequence of its construction, and its linear stability properties can also be established.

Theorem 1 *Assume that the quadrature rule has positive weights and is exact for polynomials of degree $2m + 1$. Then the implicit Hermite method is A-stable and accurate of order $2m + 2$.*

Proof Assume that f is smooth and that $u(t) \in C^{2m+2}(0, T)$. Using (4), standard estimates for quadrature errors, and the Picard formula (7) we find for the truncation error

$$\tau_j = \frac{u(t_j) - u(t_{j-1})}{h_j} - \sum_k w_k f(P_{j-1/2}(t_{j,k}; u(t_{j-1}), u(t_j))), \quad (20)$$

$$|\tau_j| = \frac{1}{h_j} \left| \int_{t_{j-1}}^{t_j} f(u(s)) ds - h_j \sum_k w_k f(P_{j-1/2}(t_{j,k}; u(t_{j-1}), u(t_j))) \right|$$

$$\begin{aligned}
&\leq \frac{1}{h_j} \left| \int_{t_{j-1}}^{t_j} f(u(s)) ds - h_j \sum_k w_k f(u(t_{j,k})) \right| \\
&\quad + \left| \sum_k w_k (f(u(t_{j,k})) - f(P_{j-1/2}(t_{j,k}; u(t_{j-1}), u(t_j)))) \right| \\
&\leq Ch_j^{2m+2}.
\end{aligned} \tag{21}$$

Now consider the Dahlquist test problem, $f(u) = \lambda u$. In this case all quadrature rules which are exact for the Hermite interpolant produce the same method. As interpolation is linear, we have that the coefficients of the interpolant are linear combinations $h_j^k \lambda^k v_{j-1}$ and $h_j^k \lambda^k v_j$, $k = 0, \dots, m$. The Picard integral then increases the powers of $h_j \lambda$ by one so that the implicit system (11) can be rearranged to:

$$Q_+(h_j \lambda) v_j = Q_-(h_j \lambda) v_{j-1} \Rightarrow v_j = \frac{Q_-(h_j \lambda)}{Q_+(h_j \lambda)} v_{j-1}, \tag{22}$$

where $Q_{\pm}(h_j \lambda)$ are polynomials of degree $m + 1$. Consistency implies

$$e^{h_j \lambda} = \frac{Q_-(h_j \lambda)}{Q_+(h_j \lambda)} + O((h_j \lambda)^{2m+3}). \tag{23}$$

The only rational function of the given degree with this accuracy is the diagonal Padé approximant. We thus conclude that our methods are A-stable [12]. $\diamond$

4 Numerical Experiments

Our first experiments treat standard problems from the ode literature and are restricted to the fourth and sixth order methods described above with either Gauss-Legendre or Gauss-Radau quadrature. Our practical implementations employ the classical Aitken algorithm adapted to Hermite interpolation to directly evaluate $P_{j-1/2}(t_{j,k}, v_{j-1}, v_j)$ and solve (11) using Newton's method with the Jacobian of the implicit system approximated by finite differences. For adaptive computations we

1. Compute v_j using the Gauss-Radau-based formulas,
2. Compute a residual, ρ_j , by substituting v_j into the Gauss-Legendre-based formulas.

We then adjust the time step by the simple rule

$$h_{j+1} = \left(\frac{\text{tol}}{\rho_j} \right)^{1/(2m+3)} h_j, \quad (24)$$

while also imposing a minimum time step.

Our final experiment examines the use of the method for evolving spectral discretizations of initial-boundary value problems for the Schrödinger equation.

4.1 *Arentsorf Orbit*

We first consider the problem of computing a periodic solutions of the restricted three-body problem which we reformulate as a first-order system of four variables:

$$\frac{d^2 y_1}{dt^2} = y_1 + 2 \frac{dy_2}{dt} - (1 - \mu) \frac{y_1 + \mu}{\left((y_1 + \mu)^2 + y_2^2 \right)^{3/2}} - \mu \frac{y_1 - (1 - \mu)}{\left((y_1 - (1 - \mu))^2 + y_2^2 \right)^{3/2}},$$

$$\frac{d^2 y_2}{dt^2} = y_2 - 2 \frac{dy_1}{dt} - (1 - \mu) \frac{y_2}{\left((y_1 + \mu)^2 + y_2^2 \right)^{3/2}} - \mu \frac{y_2}{\left((y_1 - (1 - \mu))^2 + y_2^2 \right)^{3/2}},$$

$$\mu = 0.012277471, \quad y_1(0) = .994, \quad \frac{dy_1}{dt}(0) = y_2(0) = 0,$$

$$\frac{dy_2}{dt}(0) = -2.01585106379082\dots, \quad T = 17.06521656015796\dots$$

(For graphs of the solution see [13, Ch. II].)

We note that this problem is not considered to be stiff. The main difficulty is a need for very small time steps when the orbits approach the singularities of f . However, we use it to verify convergence at the design order when (woefully inefficient) uniform time steps are employed and to test the utility of our naive time step adaptivity algorithm.

Results for fixed (small) time steps are displayed in Table 1. We observe that convergence is at design order and that the results for the two quadrature formulas are comparable, though the fourth order Radau method is somewhat more accurate than Gauss-Legendre with roles reversed at sixth order. The sixth order methods are more accurate with larger time steps. The error is simply $\sqrt{(y_1(T) - y_1(0))^2 + (y_2(T) - y_2(0))^2}$.

Results for adaptive computations with $m = 2$ are shown in Table 2. Obviously, the adaptive methods lead to a very significant reduction in the number of time steps; an accuracy of 10^{-7} is achieved with 264 steps of the adaptive method

Table 1 Convergence with fixed time steps for the Arentsorf orbit problem

Gauss-Legendre $m = 1$				Gauss-Radau $m = 1$				Gauss-Legendre $m = 2$				Gauss-Radau $m = 2$			
h	Error	Rate	h	Error	Rate	h	Error	h	Error	Rate	h	Error	Rate	h	Rate
5.69(-4)	2.30(-4)		5.69(-4)	9.65(-5)		8.53(-4)	2.98(-6)	8.53(-4)	4.46(-6)		8.53(-4)	4.46(-6)			
4.88(-4)	1.24(-4)	4.01	4.88(-4)	5.21(-5)	4.00	6.83(-4)	7.82(-7)	6.83(-4)	1.16(-6)	6.03	6.83(-4)	1.16(-6)	6.03		
4.27(-4)	7.26(-5)	4.01	4.27(-4)	3.05(-5)	4.00	5.69(-4)	2.62(-7)	5.69(-4)	3.88(-7)	6.02	5.69(-4)	3.88(-7)	6.02		
3.79(-4)	4.53(-5)	4.01	3.79(-4)	1.91(-5)	4.00	4.88(-4)	1.04(-7)	4.88(-4)	1.53(-7)	6.01	4.88(-4)	1.53(-7)	6.01		
3.41(-4)	2.97(-5)	4.01	3.41(-4)	1.25(-5)	4.00	4.27(-4)	4.66(-8)	4.27(-4)	6.87(-8)	6.01	4.27(-4)	6.87(-8)	6.01		
3.10(-4)	2.03(-5)	4.00	3.10(-4)	8.54(-6)	4.00	3.79(-4)	2.30(-8)	3.79(-4)	3.39(-8)	6.01	3.79(-4)	3.39(-8)	6.01		
2.84(-4)	1.43(-5)	4.00	2.84(-4)	6.03(-6)	4.00	3.41(-4)	1.22(-8)	3.41(-4)	1.80(-8)	6.00	3.41(-4)	1.80(-8)	6.00		

Table 2 Time steps and error as a function of tolerance for adaptive solutions of the Arentsorf problem with $m = 2$

Tol	Error	Steps
10^{-6}	$8.21(-3)$	65
10^{-8}	$1.85(-5)$	136
10^{-10}	$1.15(-8)$	264

Fig. 1 Solution and time step history for the van der Pol oscillator with tolerance 10^{-8}

while 35,000 uniform steps are required. Due to the sensitivity of the problem, the global error is much larger than the error tolerance, but is reduced in proportion to it.

4.2 Van der Pol Oscillator

Our second example is the van der Pol oscillator problem, which again we rewrite as a first order system:

$$\frac{d^2 y}{dt^2} = \epsilon^{-1} \left((1 - y^2) \frac{dy}{dt} - y \right), \quad (25)$$

$$\epsilon = 10^{-6}, \quad y(0) = 2, \quad \frac{dy}{dt}(0) = 0.$$

We solve up to $T = 11$ using the adaptive method with $m = 2$. We plot the solution and the time step histories for a tolerance of 10^{-10} in Fig. 1. Note that very small steps are needed to resolve the fast transitions, while the problem is quite stiff in the regions where y is nearly constant. Plots for the other tolerances tested, 10^{-6} and 10^{-10} , are similar though the number of time steps required varies.

Fig. 2 Left: Relative errors for NLS with various time steps and $m = 3$. Right: Relative errors for NLS with $h = .01$ and varying m

4.3 Schrödinger Equation

Lastly, we apply the method to evolve a Fourier pseudospectral discretization of the nonlinear Schrödinger equation. Precisely we consider the real problem

$$\frac{\partial v}{\partial t} = -\frac{\partial^2 w}{\partial x^2} - (v^2 + w^2)w, \quad \frac{\partial w}{\partial t} = \frac{\partial^2 v}{\partial x^2} + (v^2 + w^2)v, \quad (26)$$

for $x \in (-8, 8)$, $t \in (0, 3)$ with periodic boundary conditions $\frac{\partial v}{\partial x} = \frac{\partial w}{\partial x} = 0$ at $x = \pm 1$. We approximate the periodization of the exact solitary wave solution

$$\begin{aligned} v(x, t) &= \sqrt{50} \cos(rx - st) \cdot \text{sech}(5(x - ct)), \\ w(x, t) &= \sqrt{50} \sin(rx - st) \cdot \text{sech}(5(x - ct)), \end{aligned} \quad (27)$$

with $c = 2\pi$, $r = \pi$, $s = \pi^2 - 25$. We note that the amplitude of the solitary wave is reduced by about 17 digits at a distance of 8 from its peak so that the interaction with periodic copies is negligible over the simulation time. We use 512 Fourier modes in the computation of the derivatives and experiments show that this is sufficient to represent the solitary wave to machine precision. The implicit system was solved using Newton iterations each time step. In Fig. 2 we present results for $m = 3$ (8th order) with varying time step and for m varying from 1 to 5 (order 4 through 12) with $h = 10^{-2}$. In both cases we observe rapid convergence. We also tabulate the errors at the final time and calculate the convergence rates when $m = 3$ in Table 3. The results are clearly consistent with the design order.

Table 3 Relative errors for the Fourier pseudospectral discretization of the NLS (26) with solitary wave solution (27)

$m = 3$			$h = 0.01$	
h	Error	Rate	m	Error
3.0(−2)	4.0(−3)		1	1.2(−2)
2.0(−2)	6.7(−5)	10.1	2	5.0(−5)
1.5(−2)	4.9(−6)	9.1	3	1.8(−7)
1.0(−2)	1.8(−7)	8.1	4	8.6(−10)
7.5(−3)	2.2(−8)	7.3	5	1.2(−10)
6.0(−3)	3.2(−9)	8.7		
5.0(−3)	9.4(−10)	6.8		

5 Conclusions and Future Work

In conclusion, we have demonstrated that Hermite-Birkhoff interpolation can be used to develop singly-implicit A-stable timestepping methods of arbitrary order. A number of possible generalizations and improvements to the method are possible. These include

- 1. Stability analysis for variable coefficient or nonlinear problems using the projection properties (5);
- 2. Improved time step/order adaptivity;
- 3. Preconditioning of the implicit system for applications to partial differential equations such as spectral/pseudospectral discretizations of equations of Schrödinger type (e.g. integration preconditioners [8, 9]);
- 4. Development of IMEX schemes combining Hermite and Taylor polynomials.

Acknowledgements This work was supported in part by NSF Grant DMS-1418871. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

1. Appelö, D., Hagstrom, T.: On advection by Hermite methods. *Pac. J. Appl. Math.* **4**,125–139 (2012)

2. Appelö, D., Hagstrom, T.: Solving PDEs with Hermite Interpolation. *Lecture Notes in Computational Science*, pp. 31–49. Springer, Berlin (2015)

3. Appelö, D., Kreiss, G., Wang, S.: An explicit Hermite-Taylor method for the Schrödinger equation. *Commun. Comput. Phys* **21**, 1207–1230 (2017)

4. Appelö, D., Hagstrom, T., Vargus, A.: Hermite methods for the scalar wave equation. *SIAM J. Sci. Comput.* **40**, A3902–A3927 (2018)

5. Chen, R., Hagstrom, T.: P-adaptive Hermite methods for initial value problems. *ESAIM: Math. Model. Numer. Anal.* **46**, 545–557 (2012)

6. Chen, X., Appelö, D., Hagstrom, T.: A hybrid Hermite-discontinuous Galerkin method for hyperbolic systems with application to Maxwell’s equations. *J. Comput. Phys.* **257**, 501–520 (2014)

7. Chidwagyai, P., Nave, J.-C., Rosales, R., Seibold, B.: A comparative study of the efficiency of jet schemes. *Int. J. Numer. Anal. Model.-B* **3**, 297–306 (2012)
8. Coutsias, E., Hagstrom, T., Hesthaven, J., Torres, D.: Integration preconditioners for differential operators in spectral τ methods. *Houst. J. Math.* **1995**, 21–38 (1996). Special issue: ICOSAHOM
9. Coutsias, E., Hagstrom, T., Torres, D.: An efficient spectral method for ordinary differential equations with rational function coefficients. *Math. Comp.* **65**, 611–635 (1996)
10. Goodrich, J., Hagstrom, T., Lorenz, J.: Hermite methods for hyperbolic initial-boundary value problems. *Math. Comp.* **75**, 595–630 (2006)
11. Griewank, A.: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia (2000)
12. Hairer, E., Wanner, G.: *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*. Springer, New York (1996)
13. Hairer, E., Norsett, S., Wanner, G.: *Solving Ordinary Differential Equations I, Nonstiff Problems*. Springer, New York (1992)
14. Kornelus, A., Appelö, D.: Flux-conservative Hermite methods for simulation of nonlinear conservation laws. *J. Sci. Comput.* **76**, 24–47 (2018)
15. Liu, C., Iserles, A., Wu, W.: Symmetric and arbitrarily high-order Birkhoff-Hermite time integrators and their long-time behaviour for solving nonlinear Klein-Gordon equations. *J. Comput. Phys.* **356**, 1–30 (2018)
16. Seibold, B., Rosales, R., Nave, J.-C.: Jet schemes for advection problems. *Discrete Contin. Dyn. Syst. Ser. B* **17**, 1229–1259 (2012)
17. Vargas, A., Chan, J., Hagstrom, T., Warburton, T.: GPU Acceleration of Hermite Methods for Simulation of Wave Propagation. *Lecture Notes in Computational Science*, pp. 357–368. Springer, Berlin (2017)
18. Vargas, A., Chan, J., Hagstrom, T., Warburton, T.: Variations on Hermite methods for wave propagation. *Commun. Comput. Phys.* **22**, 303–337 (2017)
19. Vargas, A., Hagstrom, T., Chan, J., Warburton, T.: Leapfrog time-stepping for Hermite methods. *J. Sci. Comput.* **80**, 289–314 (2019)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.