Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 194 (2004) 395–408
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Stable, high-order discretization for evolution of the wave equation in 1 + 1 dimensions

John Visher [a], Stephen Wandzura [a,*], Amanda White [b]

[a] HRL Laboratories, LLC, 3011 Malibu Canyon Road, Malibu, CA 90265-4799, USA
[b] Pacific Northwest National Laboratory, P.O. Box 999, MS K5-12 Richland, WA 99352, USA

Received 18 March 2003; received in revised form 15 September 2003; accepted 15 September 2003

Abstract

We carry forward the approach of Alpert, Greengard, and Hagstrom to construct stable high-order explicit discretizations for the wave equation in one space and one time dimension. They presented their scheme as an integral form of the Lax–Wendroff method. Our perspective is somewhat different from theirs; our focus is on the *discretization* of the evolution formula rather than on its *form* (integral, differential, etc.). A key feature of our approach is the independent computation of three discretizations, one for bulk (away from boundaries) propagation, one for propagation near boundaries, and a projection operator to enforce boundary conditions. This is done in a way that is straightforward to extend to more space dimensions.

Keywords: Small-cell problem; Stability

1. Introduction

1.1. Background

This work is on numerical evolution of solutions to the wave equation,

$$\left(\nabla^2 - \frac{\partial^2}{\partial t^2}\right)\psi(\vec{x}, t), \tag{1}$$

along with appropriate initial and boundary conditions (BCs). We are particularly interested in obtaining a high-order discretization – one that has a discretization error that decreases rapidly with discretization density. There are three general ways to deal with the time evolution of such an equation:

* Corresponding author. Tel.: +1-310/317-5462.
E-mail addresses: visher@hrl.com (J. Visher), wandzura@hrl.com (S. Wandzura), amwhite@stanford.edu (A. White).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2003.09.028

- use a discretization of the spatial derivatives along with a ordinary differential equation solver, e.g. Runge–Kutta,
- discretize all the derivatives similarly, getting a system of equations that relate the field(s) at several times, or
- use an exact relation that relates the field(s) at multiple times, and discretize the linear operators that appear.

It seems natural to think that the last of these is most amenable to a high-order discretization. We use an example of this, the well known Lax–Wendroff method, based on the identity

$$\frac{1}{2}[\psi(\vec{x},\Delta t)+\psi(\vec{x},-\Delta t)] = L_{\Delta t}\psi(\vec{x},0) \equiv \cosh\sqrt{(\Delta t)^2\nabla^2}\psi(\vec{x},0), \tag{2}$$

$$= \sum_{m=0}^{\infty}\frac{(\Delta t)^{2m}\nabla^{2m}}{(2m)!}\psi(\vec{x},0), \tag{3}$$

because a hierarchy of discretizations can be obtained by representing the linear operator $L_{\Delta t}$ to the desired order. It is clear that Eq. (2) gives an *explicit* formula for $\psi(\vec{x},\Delta t)$ in terms of $\psi(\vec{x},0)$ and $\psi(\vec{x},-\Delta t)$, meaning that no linear equations need be solved to compute $\psi(\vec{x},\Delta t)$:

$$\psi(\vec{x},\Delta t) = 2L_{\Delta t}\psi(\vec{x},0) - \psi(\vec{x},-\Delta t). \tag{4}$$

The work of Alpert, Greengard, and Hagstrom (AGH) [1] was essentially concerned with various integral representations of $L_{\Delta t}$ in one, two, and three space dimensions. These integral representations suggest various approaches to discretization of $L_{\Delta t}$ and AGH showed that these can be high-order and stable.

We adopt the point of view that what is really important is the discretization of $L_{\Delta t}$ and not the representation from which it is derived. After all, the various representations are equivalent in the sense that they give the same result for any particular field $\psi(\vec{x})$.

1.2. Present approach

The important features of our approach are

- a straightforward discretization of the operator $L_{\Delta t}$,
- a high-order discretization of a projection operator which enforces the desired BCs independently of the discretization of $L_{\Delta t}$,
- a consistent discretization of the implicit identity operation on the first "time slice" $\psi(\vec{x},-\Delta t)$ on the right hand side of Eq. (4). (Simply put, we apply a high-order filter to $\psi(\vec{x},-\Delta t)$. This is essential to preserve stability.)
- applicability to both structured and unstructured meshes. (Of course, efficiency of the method depends on a regular mesh being used in as much of the modeled volume as possible.)

We march forward in time by alternating the application of the discretization of the evolution formula (Eq. (4) with the identity operator exhibited)

$$\psi(\vec{x},t+\Delta t) = 2L_{\Delta t}\psi(\vec{x},t) - L_0\psi(\vec{x},t-\Delta t), \tag{5}$$

with the boundary projection operation

$$\psi(\vec{x},t) \leftarrow P_{\mathrm{b}}\psi(\vec{x},t). \tag{6}$$

The discretization of the identity operator L_0 is a high-order low-pass filter. The notation is *not* arbitrary – L_0 is exactly the $\Delta t \to 0$ limit of $L_{\Delta t}$, and likewise for their discretizations.

It was not at all obvious to us that this plan would succeed in giving explicit stable evolution with the attempted order of convergence of discretization error. Our message is that numerical investigations demonstrate that, under some conditions, it does. Clearly, the approach deserves mathematical analysis, more sophisticated than that of which we are capable, of those conditions.

We ignore in this work the method of enforcing radiation or absorbing BCs. This is, of course, crucial for modeling of scattering problems, but involves issues that we consider totally separable from those discussed here. Accordingly, our examples are wave propagation on finite regions with simple (Dirichlet and Neumann) BCs. Our approach to discretization neither complicates nor simplifies the imposition of absorbing or radiation BCs. It is, for example, compatible with the fast method of AGH [3].

Our method is illustrated in one space dimension in this note. Similar results in $2+1$ dimensions have been obtained and will be reported separately.

2. Bulk propagation and stability

2.1. High order discretization

We first consider propagation on a homogeneous regular lattice without boundaries. This illustrates the discretization approach as well as affording a chance to apply the von Neumann method to analyze stability. The field ψ is discretized by tabulating its values on the spacetime lattice:

$$\psi_{mn} \approx \psi(m\Delta x, n\Delta t). \tag{7}$$

We wish to construct a high order discretization of the operator $L_{\Delta t}$, defined in Eq. (2). We do this by requiring the discretization to be exact for a suitable set of functions, namely polynomials up to some degree d. Clearly

$$\left(L_{\Delta t}\, x^k\right)(0) = \begin{cases} (\Delta t)^k & k \text{ even}, \\ 0 & k \text{ odd}. \end{cases} \tag{8}$$

The discretization of L is a set of $2M+1$ numbers $\hat{L}_m(\Delta t/\Delta x)$:

$$(L_{\Delta t}\psi)(x) \approx \sum_{m=-M}^{M} \hat{L}_m(\Delta t/\Delta x)\psi(x + m\Delta x), \tag{9}$$

requiring

$$\sum_{m=-M}^{M} \hat{L}_m(\tau) m^j = \beta_j \equiv \begin{cases} \tau^j & j \text{ even} \\ 0 & j \text{ odd} \end{cases}; \quad 0 \leqslant j \leqslant d. \tag{10}$$

If $2M = d$ this represents a square system of equations. However, we can (and usually must) allow $M > d/2$ to achieve stability. Before we discuss the treatment of the underdetermined systems, we consider the stability of the time evolution.

2.2. Stability analysis

Because the (discretized) propagation is invariant to translations in both space and time, we look for solutions

$$\psi_{mn} = \mathrm{e}^{\mathrm{i}(km+\omega n)}, \tag{11}$$

and find the "dispersion relation"

$$e^{i\omega} = \hat{\alpha}(k,\tau) \pm \sqrt{\hat{\alpha}^2(k,\tau) - \hat{\alpha}(k,0)}, \tag{12}$$

where

$$\tau \equiv \frac{\Delta t}{\Delta x}, \tag{13}$$

$$\hat{\alpha}(k,\tau) \equiv \hat{L}_0(\tau) + 2\sum_{m=1}^{M} \hat{L}_m(\tau)\cos(km). \tag{14}$$

Propagation is stable only if both

$$\left|\hat{\alpha}(k,\tau) \pm \sqrt{\hat{\alpha}^2(k,\tau) - \hat{\alpha}(k,0)}\right| \leqslant 1. \tag{15}$$

The stability condition motivates our approach to solving underdetermined equations for $\hat{L}_m(\tau)$. The $\hat{L}_m(\tau)$ are the coefficients of a Fourier series for $\hat{\alpha}(k,\tau)$. The requirement that the discretization be high-order (exact for polynomials up to degree d) means that the first $d+1$ derivatives (including the zeroth) of $\partial^j\hat{\alpha}(k,\tau)/\partial k^j$ at $k=0$ match those of the "true" α:

$$\alpha(k,\tau) \equiv \cos(k\tau). \tag{16}$$

Trying to force too many derivatives of $\hat{\alpha}$ to be correct at $k=0$ will cause the stability condition Eq. (15) to be violated at larger k. What we want then is to have $\hat{\alpha}(k,\tau)$ be a good approximation for small k and generally small for k large enough that it is not a good approximation. We are thus led to minimize

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} dk\hat{\alpha}^2(k,\tau) = \hat{L}_0^2(\tau) + 2\sum_{m=1}^{M}\hat{L}_m^2(\tau), \tag{17}$$

subject to the high-order constraints of Eq. (10).

2.3. Operator discretization

The minimization of Eq. (17) subject to the constraints of Eq. (10) is an elementary exercise in Lagrange multipliers. The solution is

$$\hat{L}_m = \sum_{j=0}^{d} \lambda_j m^j, \tag{18}$$

where the λ_j satisfy the linear equations

$$\lambda_0 + 2\sum_{m=1}^{M}\sum_{j'=0}^{d} m^{j'}\lambda_{j'} = \beta_0,$$

$$2\sum_{m=1}^{M}\sum_{j'=0}^{d} m^{j+j'}\lambda_{j'} = \beta_j, \tag{19}$$

where the β_j are defined in Eq. (10). We always use the same size stencils for $\hat{L}(\tau)$ and $\hat{L}(0)$.

2.4. Numerical examples

In this section, we summarize the approach, give some examples of the stencil radii M required for a given discretization ratio $\tau = \Delta t/\Delta x$ and degree d, and show measures of discretization error.

Before getting into examples, a special aspect of the 1 + 1 dimensional wave equation should be reviewed. In one space dimension, the operator $L_{\Delta t}$ is trivial:

$$L_{\Delta t}\psi(x,0) = \cosh\left(\Delta t\frac{\partial}{\partial x}\right)\psi(x,0) = \frac{1}{2}[\psi(x+\Delta t,0) + \psi(x-\Delta t,0)]. \tag{20}$$

Thus, if we choose an integral value of τ, the discrete representation $\hat{L}$ will also be trivial. Another way to put this is that any solution of the wave equation obeys

$$\psi(x,t+\Delta t) + \psi(x,t-\Delta t) = \psi(x+\Delta t,t) + \psi(x-\Delta t,t). \tag{21}$$

For noninteger τ however, $\hat{L}$ will be the sum of two high-order interpolators, because the point $x + \Delta t$ will lie between lattice points. Our examples use noninteger τ, to avoid the complete absence of discretization error. In higher dimensions, of course, the discretization of L is not so trivial for any τ. With noninteger τ, our one dimensional examples behave very much like those with any τ in higher dimensions.

The discretization recipe is straightforward:

1. Choose the desired degree d and lattice ratio $\tau = \Delta t/\Delta x$.
2. Find the minimum discretization radius M such that the stability conditions Eq. (15) are satisfied.

In Table 1 we list the minimum stencil radii for $\tau = 1/2, 3/2$ and $2 \leqslant d \leqslant 8$. For $\tau = 1/2$, the stencil are of minimum size to achieve the discretization order. For $\tau = 3/2$ and $d = 2$, the stencil is of minimum size to cover the past light cone. In these cases the system of Eq. (10) is square, and $\hat{L}_0$ is the identity matrix, which is to say there is no filtering of the field at the earliest $(t - \Delta t)$ time step. For higher order, the minimum stencil size grows not much faster than for $\tau = 1/2$.

The convergence of the discretization is examined by computing the deviation from the exact solution

$$\psi(x,t) = \cos 2\pi(x-t). \tag{22}$$

Tabulations of the integral of the square of the error as a function of the number N of discretization points used are given for various orders of discretization, $\tau = 1/2, 3/2$ in Tables 2 and 3. All calculations were done in machine precision (64 bit reals). Numbers smaller than 10^{-16} appear because of the squaring of the error. The domain of propagation is $0 \leqslant x \leqslant 1$ and N time steps are used, so that the solution is computed up to $t = \tau$. The convergence values given in the tables is p, determined by the fit of the integrated square error $\epsilon_k^2 = cN^{-p}$ for the range of N. All discretizations were verified to be stable by running for thousands of time steps from white noise initial conditions and monitoring $\int \psi(x,t)^2\,\mathrm{d}x$. Besides showing

Table 1
Minimum stencil radii, M, needed for stability

d	$\tau = 1/2$	$\tau = 3/2$
2	1	2
4	2	4
6	3	5
8	4	6

The size of the stencil is $2M + 1$.

Table 2
Integrated square error at $t = \tau = 1/2$ (N steps) for equally spaced points, periodic BCs

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
8	1.9×10^{-3}	1.0×10^{-5}	8.8×10^{-8}	9.4×10^{-10}
16	1.2×10^{-4}	4.2×10^{-8}	2.4×10^{-11}	1.7×10^{-14}
24	2.3×10^{-5}	1.6×10^{-9}	1.9×10^{-13}	2.7×10^{-17}
32	7.2×10^{-6}	1.7×10^{-10}	6.0×10^{-15}	2.7×10^{-19}
40	2.9×10^{-6}	2.8×10^{-11}	4.1×10^{-16}	7.7×10^{-21}
Convergence	4.0	8.0	11.9	15.9

Table 3
Integrated square error at $t = \tau = 3/2$ (N steps) for equally spaced points, periodic BCs

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
8	8.1×10^{-2}	3.1×10^{-2}	1.4×10^{-3}	4.8×10^{-5}
16	5.4×10^{-3}	1.8×10^{-4}	5.1×10^{-7}	1.2×10^{-9}
24	1.1×10^{-3}	7.4×10^{-6}	4.2×10^{-9}	2.0×10^{-12}
32	3.4×10^{-4}	7.6×10^{-7}	1.4×10^{-10}	2.1×10^{-14}
40	1.4×10^{-4}	1.3×10^{-7}	9.6×10^{-12}	5.8×10^{-16}
Convergence	4.0	7.7	11.7	15.6

the expected high-order convergence, these tables show the value of high-order discretization: for both values of τ the $N = 8$, $d = 8$ results are better than those for $N = 40$, $d = 2$.

3. Propagation near boundaries

The propagation formula Eq. (2) is valid anywhere in the propagation region, however if one tried to apply the "bulk" discretization near the boundaries, one would have to effectively evaluate the field outside of the propagation region. AGH called different ways of thinking of this: "Quadrature Schemes", "Interpolation Schemes", and "Extrapolation Schemes". In the end, it comes down to some discretization of Eq. (2) that uses field values only in the propagation region.

It seems as if the proper definition of the operator $L_{\Delta t}$ for field points less than Δt from the boundary might depend on the BCs. Nevertheless, we discretize this operator without regard to any BCs, using (again) the formula for monomials, Eq. (8). Near boundaries, we also might choose to have an irregular mesh. We call the points for which we use an "individually computed" discretization of L "border points". For each of these

$$(L_{\Delta t}\psi)(y) \approx \sum_{\text{neighbors } x} \hat{L}(y,x)\psi(x). \tag{23}$$

In general, the size of the discretization (the number of points x for which $\hat{L}(y,x)$ is nonzero) need not be equal to that in the bulk ($2M + 1$), or even to that for other border points y. Nevertheless, we have found that that we can achieve stability by using the same number of neighboring points in the border as in the bulk, for both completely regular meshes and those with points spaced more closely near the boundaries.

4. Enforcement of boundary conditions

As we said, a key feature of our method is *independent* discretization of the propagation and the BCs. We enforce BCs by applying a projection operator after each use of the stepping formula Eq. (5). The BCs are expressed as linear constraints

$$\sum_j b_{ij}\psi(x_j) = c_i. \tag{24}$$

We present numerical results for both homogeneous BCs ($c_i = 0$) and inhomogeneous BCs. Results are substantially similar. The BCs are enforced by applying the projection operation

$$\psi_j \leftarrow \bar{\psi}_j \equiv \psi_j + \sum_i \alpha_i b_{ij}, \tag{25}$$

where the coefficients α_i are determined by solution of the linear equations

$$\sum_{i'} \left(\sum_j b_{ij} b_{i'j} \right) \alpha_{i'} = c_i - \sum_j b_{ij}\psi_j. \tag{26}$$

In practice, one does a decomposition (e.g. LU) of the coefficient matrix $\sum_j b_{ij}b_{i'j}$ as a precomputation so that the equations can be solved efficiently at each time step.

4.1. Dirichlet boundary conditions

If there is a field discretization point x_i exactly on the boundary, the homogeneous Dirichlet BC is simply $\psi_i = 0$, and the part of the projection operation corresponding to that BC would seem to be simply setting that field value to zero. If x_i is not on the boundary, then b_{ij} would be the coefficients of a high-order extrapolation operator from the field points x_j to the boundary points x_i.

Our first attempt at this approach used only two boundary constraints of the form Eq. (24). The results were almost always unstable. The solution to this turned out to be very simple. If we want the numerical evolution to be locally exact for polynomials up to degree d we must enforce not only $\psi(x_i) = 0$, but also

$$\frac{\partial^l}{\partial t^l}\psi(x_i, t) = 0; \quad l = 0, \ldots, d. \tag{27}$$

In our formulation, the odd time derivatives are implicitly enforced by the application of the BC projection operator at the previous two time steps. For even l, we use

$$\frac{\partial^l}{\partial t^l}\psi(x_i, t) = \frac{\partial^l}{\partial x^l}\psi(x_i, t). \tag{28}$$

For nonzero l, the coefficients of the discretized boundary condition b_{ij} are thus (usually extrapolative) high-order discretization of

$$\left(\frac{\partial^2}{\partial x^2} \right)^{l/2} \psi(x_i, t).$$

In all cases we use the same discretization criteria as before – the coefficients are determined by the requirement that the discretization be exact for polynomial ψ up to degree d.

The BCs for higher derivatives of the field should not be regarded as "additional" or "extra". After all, to get an error that is proportional to a high power of Δt, one must make sure that boundary condition is obeyed to that order.

4.2. Neumann boundary conditions

Neumann BCs are handled in the analogous fashion, the coefficients b_{ij} being the extrapolative high-order discretization of

$$\frac{\partial^l}{\partial x^l}\psi(x_i,t),$$

for odd l.

4.3. Computation of boundary conditions coefficients

This is done exactly like the propagation operator. We minimize

$$\sum_j b_{ij}^2,$$

subject to the constraints that

$$\frac{\mathrm{d}^l}{\mathrm{d}x^l}(x-x_i)^k\bigg|_{x\to x_i} = \sum b_{ij}(x-x_i)^k, \tag{29}$$

for all required values of l and k.

5. Optional modification for unequally spaced points

The discretization criteria used for discretization of both the propagation operators and the boundary condition vectors can be summarized as minimization of

$$\sum_j A_j^2,$$

subject to the linear constraints

$$\sum_j A_j f_k(x_j) = r_k, \tag{30}$$

where $\{A_j\}$ is the discretization of some linear operator which gives the result r_k when acting on a function $f(x)$:

$$\int \mathrm{d}x A(x) f_k(x) = r_k. \tag{31}$$

The quantity being minimized corresponds to an trapezoidal-rule approximation to the integral of $A^2(x)$. It seems natural, in the case of unequally spaced points, to generalize the criteria to minimized the *weighted* sum of squares

$$\sum_j w_j A_j^2,$$

where $\{w_j\}$ are the weights for a quadrature rule having abscissae at the points x_j. The solution then becomes

$$A_j = \frac{1}{w_j} \sum_k \lambda_k f_k(x_j), \tag{32}$$

where the λ_k solve

$$\sum_{k'} \left[\sum_j \frac{1}{w_j} f_k(x_j) f_{k'}(x_j) \right] \lambda_{k'} = r_k. \tag{33}$$

Since the result will usually depend relatively weakly on the quadrature weights w_j, it does not seem necessary that a highly accurate rule be used, unless, of course, one actually needs to approximate an integral. In the examples presented in the next section, quadrature weights are used as described in the case of unequally spaced points. The answers were found to be somewhat better than using $w_j = 1$, although both were stable and had the expected rate of convergence.

6. Numerical examples

We exhibit convergence results for both homogeneous and inhomogeneous BCs.

6.1. Homogeneous boundary conditions

We solved the wave equation with mixed homogeneous BCs: Dirichlet at one end ($x = 0$) and Neumann at the other ($x = 1$). We computed the error by deviation from the exact solution

$$\psi(x,t) = \sin\frac{3\pi}{2} x \cos\frac{3\pi}{2} t. \tag{34}$$

Solutions were computed with various orders of discretization with $\tau = 1/2, 3/2$. We show results for both equally spaced points and for meshes that are tapered near the boundaries. The order of convergence is exhibited by the dependence of the integrated square error on the number N of discretization points used. The convergence values given in the tables is p, determined by the fit of the integrated square error $\epsilon_k^2 = cN^{-p}$ for the range of k. All discretizations were verified to be stable by running for thousands of time steps from white noise initial conditions and monitoring $\int \psi(x,t)^2 \, dx$.

In all the numerical examples, the radius of the stencil, M, is determined solely by the degree of the method. Even when points are not equally spaced, the values of M are given by Table 1.

6.1.1. Equally spaced points

The results for $\tau = 1/2$ are shown in Table 4. The results for $\tau = 3/2$ are shown in Table 5. These examples were run from $t = 0$ to 1, so that the number of time steps was N/τ. Both sets of data demonstrate good convergence, $p \geqslant 2d$. The superior efficiency of high-order discretization is again apparent, in both cases the coarsest discretization with $d = 8$ is more accurate than the finest discretization with $d = 2$ (see Table 6).

Table 4
Integrated square error after N/τ time steps for equally spaced points, $\tau = 1/2$

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
12	2.4×10^{-4}	9.9×10^{-6}	2.1×10^{-7}	6.2×10^{-9}
24	1.6×10^{-5}	3.2×10^{-9}	1.1×10^{-11}	2.7×10^{-14}
36	3.2×10^{-6}	4.1×10^{-11}	1.9×10^{-14}	1.3×10^{-17}
48	1.0×10^{-6}	2.1×10^{-12}	2.6×10^{-16}	5.3×10^{-20}
60	4.2×10^{-7}	2.7×10^{-13}	9.0×10^{-18}	6.7×10^{-22}
72	2.0×10^{-7}	5.8×10^{-14}	5.0×10^{-19}	2.2×10^{-23}
84	1.1×10^{-7}	1.6×10^{-14}	4.2×10^{-20}	1.2×10^{-24}
Convergence	4.0	10.4	15.0	18.7

Table 5
Integrated square error after N/τ time steps for equally spaced points, $\tau = 3/2$

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
18	1.4×10^{-3}	1.8×10^{-4}	1.8×10^{-6}	1.3×10^{-8}
36	2.6×10^{-5}	5.7×10^{-8}	8.5×10^{-11}	4.5×10^{-14}
54	4.3×10^{-6}	7.3×10^{-10}	2.1×10^{-13}	1.9×10^{-17}
72	1.3×10^{-6}	4.9×10^{-11}	3.2×10^{-15}	8.3×10^{-20}
90	5.0×10^{-7}	7.1×10^{-12}	1.3×10^{-16}	1.3×10^{-21}
108	2.4×10^{-7}	1.5×10^{-12}	9.9×10^{-18}	4.7×10^{-23}
126	1.2×10^{-7}	4.3×10^{-13}	1.2×10^{-18}	2.8×10^{-24}
Convergence	4.7	10.2	14.5	18.6

6.1.2. Unequally spaced points

The test with unequally spaced points near the boundaries used abscissae derived from Gauss Legendre quadrature rules. The unequally spaced points and the associated weights are tabulated in Table 6, assuming the last equally spaced point is $x_0 = 0$. The equally spaced points are separated by a unit length, with unit weights. The first point in the tapered set is placed at 1. The same point spacing and weights are used on both ends (which have different BCs). The BCs are enforced at locations tabulated in the column labeled "Endpoint." The resulting abscissae are mapped into the interval $0 < x < 1$. So that each case runs over the same time interval ($0 \leqslant t \leqslant 1$), τ, the ratio $\Delta t/\Delta x$ (in the regular part of the mesh) varies somewhat. Specifically, after choosing a nominal τ_0, the number of time steps is given by $N_t = N/\tau_0$, and then $\Delta t = 1/N_t$. The resulting τs approach τ_0 from below as N increases. In other words, τ_0 is the ratio of Δt to the average of Δx over the mesh. Tabulations of the integrated square error and the approximate convergence rate are given in Table 7 for $\tau = 1/2$ and in Table 8 for $\tau = 3/2$.

For the case $d = 8, \tau_0 = 3/2, N = 18$, the ratio of Δt to minimum point spacing is $\sim$7.58. The degree to which this apparently violates the Courant–Friedrichs–Lewy constraint $\Delta t/\Delta x \leqslant 1$ without losing stability demonstrates its utility, because it allows fine discretization near boundaries. Of course, more generally, one could regard the discretization as consistent with the CFL constraint in the sense that there are always points in the stencil outside the past light cone.

The use of a tapered mesh near the boundaries only results in less error for high discretization orders and fine discretization. This does *not* mean that tapered meshes in higher dimensions will not be useful – in one space dimensions there is no need to resolve more detail at the boundaries.

Table 6
Abscissae and weights for points near boundary

Stencil radius M	Endpoint	Abscissae x_j	Weights w_j
1	2.595389	2.453886	0.3589914
		1.885491	0.755935
		1.000000	0.9804617
2	3.858537	3.770901	0.2239182
		3.405347	0.5019379
		2.781822	0.7358097
		1.955575	0.9043423
		1.000000	0.9925291
3	5.127275	5.063807	0.1625074
		4.796124	0.3709135
		4.327688	0.5622999
		3.680294	0.7274223
		2.884197	0.8585373
		1.976621	0.9495098
		1.000000	0.9960851
4	6.397964	6.348216	0.1274905
		6.137416	0.2932146
		5.764537	0.4507562
		5.240223	0.5953526
		4.579545	0.7228262
		3.801508	0.8295069
		2.928494	0.9123247
		1.985616	0.9688968
		1.000000	0.9975957
5	7.669561	7.628656	0.1048763
		7.454908	0.2421512
		7.145873	0.3749203
		6.707496	0.5004101
		6.148301	0.6161636
		5.479169	0.7199254
		4.713124	0.8096750
		3.865075	0.8836655
		2.951530	0.9404565
		1.990268	0.9789427
		1.000000	0.9983748
6	8.941655	8.906923	0.08907104
		8.759192	0.2061271
		8.495590	0.3204653
		8.119757	0.4303203
		7.636963	0.5341377
		7.053982	0.6304578
		6.378998	0.7179281
		5.621487	0.7953204
		4.792081	0.8615482
		3.902424	0.9156818
		2.965004	0.9569613
		1.992981	0.9848072
		1.000000	0.9988285

Table 7
Integrated square error after N/τ_0 time steps for tapered mesh, $\tau_0 = 1/2$

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
12	4.6×10^{-4}	8.7×10^{-5}		
24	2.2×10^{-5}	2.3×10^{-8}	8.3×10^{-12}	1.8×10^{-14}
36	4.0×10^{-6}	2.9×10^{-10}	8.2×10^{-15}	2.4×10^{-18}
48	1.2×10^{-6}	1.6×10^{-11}	6.8×10^{-17}	5.9×10^{-21}
60	4.7×10^{-7}	1.8×10^{-12}	1.7×10^{-18}	6.8×10^{-23}
72	2.2×10^{-7}	3.2×10^{-13}	8.0×10^{-20}	1.9×10^{-24}
84	1.2×10^{-7}	7.5×10^{-14}	5.9×10^{-21}	2.2×10^{-25}
96			6.7×10^{-22}	7.8×10^{-27}
Convergence	4.2	10.7	16.7	20.2

Table 8
Integrated square error after N/τ_0 time steps for tapered mesh, $\tau_0 = 3/2$

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
18	5.1×10^{-3}			
36	1.3×10^{-4}	1.6×10^{-8}	7.5×10^{-13}	3.6×10^{-14}
54	1.6×10^{-5}	3.3×10^{-10}	9.2×10^{-16}	3.9×10^{-18}
72	3.6×10^{-6}	3.0×10^{-11}	6.0×10^{-17}	9.1×10^{-21}
90	1.2×10^{-6}	4.9×10^{-12}	5.0×10^{-18}	1.0×10^{-22}
108	5.0×10^{-7}	1.1×10^{-12}	6.1×10^{-19}	2.8×10^{-24}
126	2.4×10^{-7}	3.3×10^{-13}	9.9×10^{-20}	9.6×10^{-26}
144		1.1×10^{-13}	2.1×10^{-20}	2.4×10^{-27}
Convergence	5.1	8.5	12.1	22.4

Table 9
Integrated square error after N/τ_0 time steps for tapered mesh with inhomogeneous BCs, $\tau_0 = 1/2$

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
12	2.7×10^{-3}	5.7×10^{-3}		
24	1.4×10^{-4}	2.3×10^{-6}	5.4×10^{-10}	1.3×10^{-11}
36	2.6×10^{-5}	2.8×10^{-8}	8.0×10^{-13}	2.4×10^{-15}
48	7.9×10^{-6}	1.4×10^{-9}	8.2×10^{-15}	6.9×10^{-18}
60	3.2×10^{-6}	1.4×10^{-10}	2.4×10^{-16}	1.8×10^{-19}
72	1.5×10^{-6}	2.3×10^{-11}	1.4×10^{-17}	4.8×10^{-20}
84	8.1×10^{-7}	5.0×10^{-12}	1.2×10^{-18}	
96			1.4×10^{-19}	
Convergence	4.2	10.7	15.9	18.2

6.2. Inhomogeneous boundary conditions

The application of our method to problems with inhomogeneous BCs is straightforward. In this case the c_i of Eqs. (24)–(26) do not vanish. For a numerical example, we again choose the solution

$$\psi(x,t) = \cos 2\pi(x-t), \tag{35}$$

without using a cyclic mesh to establish periodic BCs. Instead we use the general BC projection technique to enforce the correct values of $\nabla^{2k}\psi(x,t)$ at $x = 0, 1$ with $k = 0, \ldots, d/2$, where d is the degree of the

Table 10
Integrated square error after N/τ_0 time steps for tapered mesh with inhomogeneous BCs, $\tau_0 = 3/2$

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
18	7.6×10^{-2}			
36	2.3×10^{-3}	3.4×10^{-6}	2.0×10^{-9}	2.1×10^{-11}
54	2.4×10^{-4}	4.8×10^{-8}	3.6×10^{-12}	3.1×10^{-15}
72	4.9×10^{-5}	2.9×10^{-9}	5.7×10^{-14}	7.3×10^{-18}
90	1.5×10^{-5}	3.6×10^{-10}	2.5×10^{-15}	3.7×10^{-19}
108	5.9×10^{-6}	6.9×10^{-11}	2.1×10^{-16}	
126	2.7×10^{-6}	1.7×10^{-11}	2.6×10^{-17}	
140		5.4×10^{-12}	4.4×10^{-18}	
Convergence	5.3	9.6	14.3	19.8

discretization. (This is an inhomogeneous Dirichlet BC.) Results are stable with the same parameter values and tapered mesh as those of Section 6.1.2. Results are tabulated in Tables 9 and 10. Errors are somewhat larger than the results with homogeneous BCs, but the convergence rates are comparable.

7. Conclusions

The results presented here constitute an encouraging validation of AGH [1]. They show high-order convergence with stability that is not spoiled by the presence of small cells near the boundary. No other approach to discretization of the wave equation has achieved this. The results also clearly exhibit the dramatic superiority in efficiency of high-order discretization which motivated this work. The method is formulated in a way that is straightforward to implement in any number of dimensions. A companion paper will demonstrate similar results in $2+1$ dimensions. The results also raise more questions than they answer. In this section we list some obvious directions for continued research.

7.1. Theorems on stability and convergence

Although we have demonstrated stable high-order examples, we do not know the general conditions for this behavior. We have not even proved that the overall high-order convergence is not spoiled by the interleaving of propagation step with boundary condition projection.

7.2. First order formulation

We have taken the Lax–Wendroff approach that relates the fields at three time slices and uses only the field as the dynamic variable. It is clear that the same approach can be used to generalize the formulation that relates the field and its first time derivative at two time slices:

$$\psi(\vec{x}, \Delta t) = \cosh \sqrt{(\Delta t)^2 \nabla^2}\psi(\vec{x}, 0) + \frac{\Delta t}{\sqrt{(\Delta t)^2 \nabla^2}} \sinh \sqrt{(\Delta t)^2 \nabla^2}\dot{\psi}(\vec{x}, 0), \tag{36}$$

$$\dot{\psi}(\vec{x}, \Delta t) = \cosh \sqrt{(\Delta t)^2 \nabla^2}\dot{\psi}(\vec{x}, 0) + \frac{\sqrt{(\Delta t)^2 \nabla^2}}{\Delta t} \sinh \sqrt{(\Delta t)^2 \nabla^2}\psi(\vec{x}, 0). \tag{37}$$

As before, one need not worry about any branch cuts, because the power series for the operators contain only integer powers of $(\Delta t)^2 \nabla^2$.

7.3. Efficiency analysis

Study is needed to gain insight as to the most efficient way to solve particular problems to a desired accuracy. Some obvious issues are listed in this section.

7.3.1. Parameter selection

For the solution to a given problem with a desired accuracy, it would be good to have justifiable procedure for selection of the many numerical parameters, including

- discretization density,
- degree of approximation, and
- time step.

7.3.2. Balance of precomputation with propagation computation

As in frequency domain computations, there will be some tradeoffs between precomputation and solution. For large problems, precomputation will be dominated by the discretization of the border propagation, the boundary constraints, and the decomposition of the coefficient matrix $\sum_j b_{ij} b_{i'j}$ which appears in Eq. (26). Because of its local structure, this decomposition can be accelerated by employing nested dissection [2].

7.4. Use of "wave discretization"

For frequency domain problems, the use of discretizations that are specified by minimization of RMS error over some spectrum of sin waves (rather than exactitude for low-order polynomials) results in more efficient calculation for a given accuracy. It seems likely that the same will apply to time domain problems. This should be investigated.

7.5. Nonuniform border points

We have shown that use of a tapered mesh near the boundary does not prevent stability and can result in smaller discretization error, even in one dimension which might be thought to be of marginal utility. How to best taper the mesh should be studied.

7.6. Summary

We have shown here that, in $1+1$ dimensions, good results can be obtained with all points evenly spaced, or with a tapered mesh near the boundary. We do not know how to optimize the placement of border points, however using the abscissae of reasonable quadrature rule near the boundary works somewhat better that equally spaced points.

References

[1] B.K. Alpert, L.F. Greengard, T. Hagstrom, An integral evolution formula for the wave equation, Journal of Computational Physics 162 (2000) 536–543.
[2] A. George, J.W.-H. Liu, Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall, Englewood Cliffs, NJ, 1981.
[3] B.K. Alpert, L.F. Greengard, T. Hagstrom, Rapid evaluation of nonreflecting boundary kernels for time-domain wave propagation, SIAM Journal of Numerical Analysis 37 (4) (2000) 1138–1164.