Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 199 (2004) 763–775
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Stable, high-order discretization for evolution of the wave equation in 2 + 1 dimensions

Stephen Wandzura *

HRL Laboratories, LLC, Computational Physics, 3011 Malibu Canyon Rd., Malibu, CA 90265-4799, USA

Received 4 December 2003; received in revised form 8 March 2004; accepted 22 March 2004
Available online 7 May 2004

Abstract

In a previous paper, it was shown that consistent high-order discretization can make a Lax–Wendroff method stable in the presence of small cells near the boundary in one space and one time dimension. Here I show that a straightforward extension of this approach to two space dimensions is similarly successful.

Keywords: Small-cell problem; Stability

1. Introduction

Solutions to the wave equation,

$$\left(\nabla^2 - \frac{\partial^2}{\partial t^2}\right)\psi(\vec{x},t) = 0, \tag{1}$$

satisfy the following identity relating the field at three equally spaced times:

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

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

The well-known Lax–Wendroff method is based on this identity and is usually implemented using some finite difference approximations to powers of ∇^2. Alpert et al. [1] suggested various high-order discretizations of Eq. (2) that can give stable time marching in one space and one time dimension, even with small cells (size $< \Delta t$) in parts of the mesh. Such discretizations have been further analyzed by Li and Greengard

* Tel.: +310/317-5462.
E-mail address: wandzura@hrl.com (S. Wandzura).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2004.03.011

[3] showing that a certain class ("strongly consistent") of such schemes have the desired properties. In a previous paper by Visher et al. [4], a prescription for an explicit discretization of Eq. (2) in one space dimension, along with a projection operator that enforces the desired boundary conditions, was given. Numerical examples showed that alternated use of the resultant propagation and projection operators can give stable results with discretization error that decreases as a high power of the discretization scale. A key ingredient of that approach is recognizing the implicit identity operator (L_0) when Eq. (2) is solved for $\psi(\vec{x}, \Delta t)$:

$$\psi(\vec{x}, \Delta t) = 2L_{\Delta t}\psi(\vec{x}, 0) - L_0\psi(\vec{x}, -\Delta t). \tag{4}$$

Discretizing L_0 by the same prescription as is used for $L_{\Delta t}$ results in a low pass filter which facilitates stability. In this paper, I show that a straightforward implementation of this approach in two space dimensions is similarly successful in some situations. However, I have not formulated any underlying stability theory.

2. Discretization

In this section the prescription used for numerical evolution of the wave equation with boundary conditions is given. This duplicates, to a large extent, the presentation of [4], and is included for completeness. A few aspects that arise only in more than one space dimension are detailed.

2.1. General discretization prescription

To use this method, we need to discretize linear operators: the "propagation" operator $L_{\Delta t}$ and interpolation/extrapolation/differentiation operators. All operators acting on a field are discretized as follows. The field is assumed to be tabulated at a collection of points $\vec{x}_j$, which need not be (but mostly are) on a regular lattice. The result of a linear function a on some known fields $f_k(\vec{x})$ (taken here to be polynomials) are denoted as r_k:

$$(a, f_k) = r_k. \tag{5}$$

Linear functions are approximated by

$$(a, f) \approx \sum_j \hat{a}_j f(\vec{x}_j). \tag{6}$$

The coefficients $\hat{a}_j$ are computed by minimizing the weighted sum of squares

$$\sum_j w_j \hat{a}_j^2, \tag{7}$$

subject to the constraints that the result be exact for the fields f_k:

$$\sum_j \hat{a}_j f_k(\vec{x}_j) = r_k. \tag{8}$$

(Obviously the number of discretization points used must not be less than the number of functions f_k.) The coefficients are given by

$$\hat{a}_j = \frac{1}{w_j} \sum_k \lambda_k f_k(\vec{x}_j), \tag{9}$$

where the λ_k solve

$$\sum_{k'} \left[\sum_j \frac{1}{w_j} f_k(\vec{x}_j) f_{k'}(\vec{x}_j) \right] \lambda_{k'} = r_k. \tag{10}$$

In the case of equally spaced points, the w_j may be taken to be equal. For nonregular meshes, it seems reasonable to take them to be the weights of a suitable (i.e., of order commensurate with that of the discretization) quadrature rule for the integral of a function over the propagation region. In one space dimension, this was shown to result in reductions of about an order of magnitude in discretization error for a typical range of discretization scales and orders.

2.2. Precomputation

Suppose we want to compute the time evolution in a spatial domain S, with suitable initial conditions and boundary conditions on the boundary ∂S. The steps of the precomputation are:

- Choose the functions for which the discretization is to be exact. In our examples, we choose all polynomials of degree not exceeding some chosen value d.
- Find the minimum stencil size for which the free space discretization will be stable.
- Choose discretization points in S. For most of S, these should be on a lattice, so that the propagation operator can be applied by convolution. (I have used only square lattices, but it is conceivable that discretization on a triangular lattice could be somewhat more efficient, because of higher symmetry.) An irregular mesh may be used to resolve better the field near the boundary ∂S. (This option is not explored numerically in this paper.)
- Classify the discretization points as "bulk" or "border". Bulk points are those for which each point of the stencil of the propagation operator corresponds to a discretization point on the square lattice. Border points are the rest.
- For each of the border points, compute a discretization of the propagation operator using the point itself and its neighbors. In some cases, I have found that it is necessary to use more neighbors in the border region than in the bulk stencil to preserve stability. For a few cases (with high discretization order and Neumann boundary condition), even the number of discretization points in the bulk had to be increased. This did not occur in the $(1+1)d$ examples studied in [4]. The issue of the number of points required to maintain stability is an important subject for future research.
- Pick points on the boundary, and compute differentiators/extrapolators, using neighboring discretization points, to them, for the desired boundary conditions. For example, for Dirichlet conditions, construct extrapolators for the field and integer powers of ∇^2.
- Decompose the outer product of the constraint vectors so that the projection operator can be applied efficiently.

2.3. Propagation operator in bulk region

For efficiency in precomputation and convolution, most of the propagation domain is discretized by tabulation of the field on a square lattice

$$\psi_{ijn} \approx \psi(i\Delta x, j\Delta x, n\Delta t). \tag{11}$$

The propagation operator

$$L_{\Delta t} \equiv \cosh \sqrt{(\Delta t)^2 \nabla^2} \tag{12}$$

is discretized according to the general procedure above. The result of applying L to polynomial "testing" functions is (evaluated at the original)

$$(L_{\Delta t} x^m y^n)(0,0,t) = (\Delta t)^{m+n} A(m,n) X\left(\frac{m+n}{2}\right), \tag{13}$$

where

$$A(m,n) \equiv \frac{1}{2\pi}\int_0^{2\pi} \mathrm{d}\theta \cos^m(\theta)\sin^n(\theta) \tag{14}$$

$$= \begin{cases} \frac{(m-1)!!(n-1)!!}{2^{\frac{m+n}{2}}\left(\frac{m+n}{2}\right)!}, & m \text{ and } n \text{ even}, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

and

$$X(m) \equiv (2m)^2 \int_0^\infty \mathrm{d}x\, x^{2m-1} \log\frac{\sqrt{1-x^2}+1}{x} = \frac{[(2m)!!]^2}{(2m)!}. \tag{16}$$

The values of Eq. (13) are used to compute the coefficients associated with the nearest M lattice points to the origin, so that

$$(L_{\Delta t}\psi)(x,y,t) \approx \sum_{ij} \hat{L}_{ij}(\tau)\psi(x+i\Delta x, y+j\Delta x, t), \tag{17}$$

where $\tau = \Delta t/\Delta x$. In order to make the discretization exact for all polynomials of degree not greater than d, M must be at least as big as the number $(d+l)(d+2)/2$ of such polynomials. As in one dimension, it must usually be greater to give stable propagation. Also, we insist on a stencil that has "square" (D_4) symmetry, so that the allowed values of M are

$$\{5, 9, 13, 21, 25, 29, 37, 45, 49, 57, 61, 69, 81, 89, 97, \ldots\}.$$

To determine if a discretization of L is stable, we look at the function

$$\hat{\alpha}(k_x, k_y, \tau) \equiv \sum_{ij} \hat{L}_{ij} \cos(k_x i)\cos(k_y j). \tag{18}$$

As in one dimension, propagation is stable only if both

$$\left|\hat{\alpha}(k_x,k_y,\tau) \pm \sqrt{\hat{\alpha}^2(k_x,k_y,\tau) - \hat{\alpha}(k_x,k_y,0)}\right| \leqslant 1 \tag{19}$$

for all $\pi \leqslant k_x, k_y \leqslant \pi$ (the first Brillouin zone). If a particular discretization is unstable, a larger value of M must be used.

2.4. Propagation operator in the border region

In the border region, the propagation operator is discretized differently than in the bulk, because (by definition) a bulk stencil centered on a border point will not entirely overlay discretization points. The requirement that the discretized operator give the correct answer (Eq. (13)) for polynomials up to the chosen degree remains the same, however. The general discretization method given in Section 2.1 is used, choosing the near neighbor discretization points to each border point. (These will usually include

both border and bulk points.) This must, in general, be done independently for each border point, but symmetries of the problem and the discretization point set could be used to reduce the necessary precomputation. The number of points used can be the same as that (M) in the bulk region, but in some cases I have found empirically that more points are necessary to avoid instability. If a finer discretization were used near the boundary, I would expect this necessity to have a greater tendency to occur.

2.5. Boundary condition projection operator

This step exhibits the biggest change from one to two space dimensions, because in one dimension, there are only two boundary points. In two dimensions, we need to pick a set of point on the boundary ∂S at which to impose the boundary conditions. A reasonable approach is to take their separation to be commensurate with the spacing of discretization points near the boundary. For example, with the discretization points all on a square lattice, I use boundary points along ∂S separated by the distance nearly equal to the lattice constant. In the case of a nonuniform mesh, points separated on the boundary by a distance commensurate with the smallest spacing in the nearby propagation region would be appropriate. For each boundary point, extrapolation/differentiation estimators are computed for the quantities constrained by desired boundary condition and powers of the Laplacian operating on these. For example, for Dirichlet conditions, none computes estimators for

$$\nabla^{2p}\psi(\vec{x}) \approx \sum_i b_i \psi(\vec{x}_i) \tag{20}$$

for $p = 0, \ldots, [d/2]$, $\vec{x}$ is a boundary point, d is the degree of the discretization, and the sum is taken over those i for which $\vec{x}_i$ is a near neighbor of $\vec{x}$. For Neumann conditions, we use computed estimators for $\nabla^{2p}\hat{n}\cdot\nabla\psi(\vec{x})$, where $\hat{n}$ is the normal to ∂S and $p = 0, \ldots, [d/2]$. In discretization of boundary conditions, I have always used the same number of points $\vec{x}_j$ as used for the propagation operator in the border region, which may be more than the number used in the bulk region, as described in Section 2.4. Labeling the constraint vectors for all discretization points $\vec{x}$ on the boundary and all p by an index l, the discretized boundary conditions are

$$\sum_i b_{li}\psi(\vec{x}_i) = c_l. \tag{21}$$

For each boundary point, there will be multiple values of the index l. For example, for a discretization of degree $d = 2$ and Dirichlet boundary conditions, there will be two values of l for each point, one to enforce the correct value of ψ, and the other to enforce the correct value of $\nabla^2\psi$. For homogeneous boundary conditions, the c_l vanish, of course. To apply inhomogeneous boundary conditions, one must have high-order approximations to powers of $(\partial/\partial t)^2$ acting on the boundary conditions. Application of projection operators is facilitated by decomposing the outer product matrix

$$B_{lm} \equiv \sum b_{li} b_{mi}, \tag{22}$$

so that linear equations with the coefficient matrix B can be solved quickly. By virtue of the locality of the boundary conditions, the matrix B is sparse and is amenable to sparse decomposition by nested dissection [2]. Then, to apply the projection operator to a particular vector ψ we solve for the coefficients α_l,

$$\sum_m B_{lm}\alpha_m = c_l - \sum_i b_{li}\psi(\vec{x}_i), \tag{23}$$

and make the replacement

$$\psi(\vec{x}_i) \leftarrow \psi(\vec{x}_i) + \sum_i \alpha_l b_{li}. \tag{24}$$

(I use a single index i to label the points because the border points need not be on a regular lattice.)

2.6. Evolution

Having discretizations of all the operators in hand, time stepping is done by alternating the application of the discretization of the evolution formula

$$\psi(\vec{x}, \Delta t) = 2L_{\Delta t}\psi(\vec{x}, t) - L_0\psi(\vec{x}, t - \Delta t) \tag{25}$$

with the discretization of the boundary projection operation

$$\psi(\vec{x}, t) \leftarrow P_b\psi(\vec{x}, t), \tag{26}$$

as described in Section 2.5. The discretization of the identity operator L_0 is a high-order low-pass filter. As emphasized in [4], this notation is *not* arbitrary – L_0 is exactly the $\Delta t \to 0$ limit of $L_{\Delta t}$, and likewise for their discretizations.

3. Numerical examples

In this section, I report the results of four examples of the method, exhibiting discretizations of various degrees (2–8) that give stable results, and the convergence of discretization error with density of sample points. First, the bulk propagation is studied on a square region with periodic boundary conditions (BCs). Then, I show two examples with homogeneous BCs, a square with Dirichlet BCs on two opposite sides and Neumann BCs on the other two, and a disk with Neumann BCs. Finally, I give an example of inhomogeneous BCs: a plane wave solution with Dirichlet BCs on a circle.

3.1. Bulk propagation

The stencils for $d = 2$, $\tau = \Delta t/\Delta x = 1$, and $M = 21$ are

$$\hat{L}_1 = \begin{pmatrix} 0 & 0.0268657 & 0.0656716 & 0.0268657 & 0 \\ 0.0268657 & 0.143284 & 0.18209 & 0.143284 & 0.0268657 \\ 0.0656716 & 0.18209 & 0.220896 & 0.18209 & 0.0656716 \\ 0.0268657 & 0.143284 & 0.18209 & 0.143284 & 0.0268657 \\ 0 & 0.0268657 & 0.0656716 & 0.0268657 & 0 \end{pmatrix}, \tag{27}$$

$$\hat{L}_0 = \begin{pmatrix} 0 & -0.041791 & 0.00895522 & -0.041791 & 0 \\ -0.041791 & 0.110448 & 0.161194 & 0.110448 & -0.041791 \\ 0.00895522 & 0.161194 & 0.21194 & 0.161194 & 0.00895522 \\ -0.041791 & 0.110448 & 0.161194 & 0.110448 & -0.041791 \\ 0 & -0.041791 & 0.00895522 & -0.041791 & 0 \end{pmatrix}. \tag{28}$$

These are the smallest stencils (for the given values of d and τ which result in stable propagation. In Fig. 1, the stability function

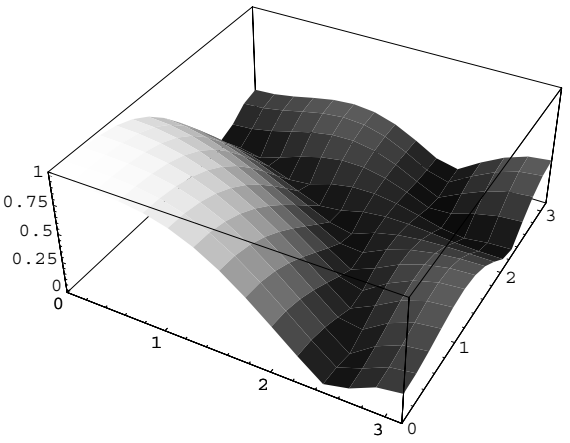

Fig. 1. Stability function for the stencils given in Eqs. (27) and (28).

$$\mathscr{S}(k_x,k_y) \equiv \max\left|\hat{\alpha}(k_x,k_y,\tau) \pm \sqrt{\hat{\alpha}^2(k_x,k_y,\tau) - \hat{\alpha}(k_x,k_y,0)}\right| \tag{29}$$

is shown. Since $\mathscr{S}(0,0) = 1$ and $S(k_x,k_y) < 1$ for all other values of k_x, k_y, the propagation is stable. As a contrast, the stability function for $M = 13$ is plotted in Fig. 2. Near the corner of the Brillouin zone, $k_x = k_y = \pi$, it exceeds 1, indicating instability (cf. Fig. 1).

In Table 1 the stencil sizes for $\tau = 1$ and $2 \leqslant d \leqslant 8$ are listed. The convergence of the discretization is examined by computing the deviation from the exact solution

$$\psi(x,y,t) = \cos(2\pi x)\sin(4\pi y)\cos(2\sqrt{5}\pi t). \tag{30}$$

Tabulations of the integral of the square of the error as a function of the number N of discretization points in each direction are given in Table 2 for various orders of discretization and $\tau = \Delta t/\Delta x = 1$. Periodic boundary conditions were enforced in the usual way. 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 prop-

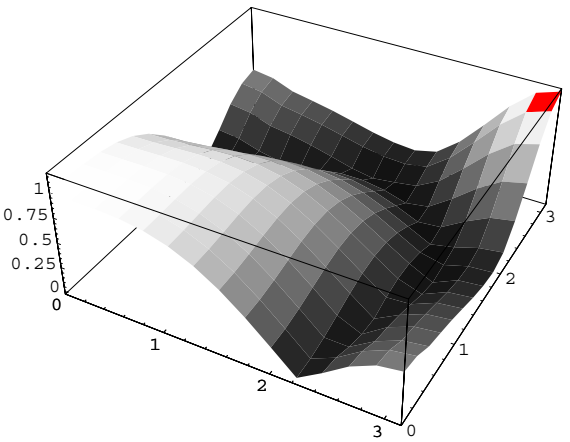

Fig. 2. Stability function for the stencils with $M = 13$.

Table 1
Minimum stencil sizes, M, needed for stability ($\tau = 1$)

d	M
2	21
4	25
6	49
8	81

Table 2
Integrated square error at $t = 1/2$ ($N/2$ steps) for a square grid, periodic boundary conditions

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
16		2.6×10^{-4}	7.1×10^{-6}	7.2×10^{-7}
32	1.1×10^{-3}	2.0×10^{-6}	3.1×10^{-9}	2.6×10^{-11}
48	3.0×10^{-4}	8.9×10^{-8}	2.7×10^{-11}	4.8×10^{-14}
64	1.1×10^{-4}	9.4×10^{-9}	9.0×10^{-13}	5.2×10^{-16}
80	4.6×10^{-5}	1.6×10^{-9}	6.3×10^{-14}	1.5×10^{-17}
96	2.3×10^{-5}	3.8×10^{-10}	7.2×10^{-15}	8.4×10^{-19}
112	1.3×10^{-5}	1.1×10^{-10}	1.1×10^{-15}	7.4×10^{-20}
128	7.5×10^{-6}	3.9×10^{-11}	2.3×10^{-16}	1.1×10^{-20}
Convergence	3.7	7.6	11.6	15.4

agation is $0 \leqslant x, y \leqslant 1$ and $N/2$ time steps are used so that the solution is computed up to $t = 1/2$. The convergence values given in the tables is p, determined by the fit of the integrated square error $\epsilon^2 = cN^{-p}$ for the range of N. The same data are plotted in Fig. 3. All discretizations were verified to be stable by running for thousands of time steps from white noise initial conditions and monitoring $\int \psi^2(x, y, t)\mathrm{d}x\,\mathrm{d}y$.

3.2. Square region with mixed homogeneous boundary conditions

In this section, I give numerical results for the wave equation in a square domain with homogeneous Dirichlet boundary conditions on two opposing sides ($x = 0$ and $x = 1$) and homogeneous Neumann conditions on the other two sides ($y = 0$ and $y = 1$). I compute the error by deviation from the exact solution

$$\psi(x, y, t) = \sin\pi x \cos\pi y \cos\sqrt{2}\pi t. \tag{31}$$

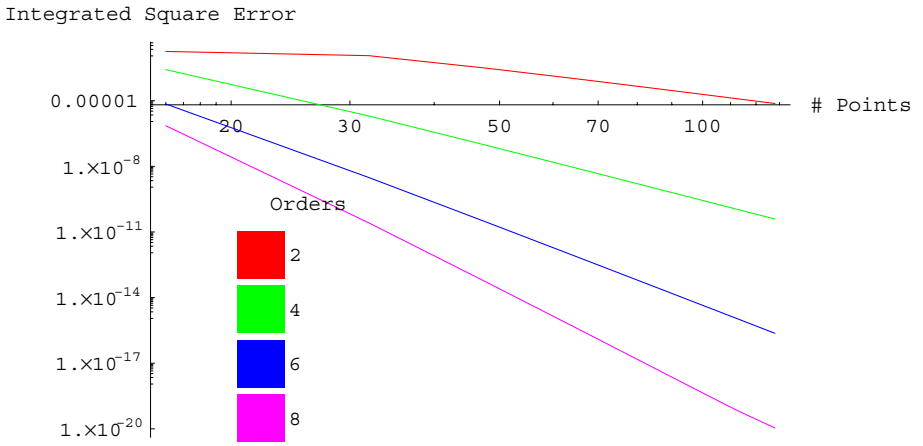

Fig. 3. Integrated square error at $t = 1/2$ ($N/2$ steps) for a square grid, periodic boundary conditions.

In order to achieve stability, it was necessary to use more points for the discretization of the border propagation and boundary condition operators for all degrees of discretization. For degrees 6 and 8, more points in the bulk region were also required. The same number of points were used for all these discretizations. The number of points used is tabulated in Table 3. Tabulations of the integral of the square of the error as a function of the number N of discretization points in each direction are given in Table 4 for various orders of discretization and $\tau = 1$. The convergence values given in the table are p, determined by the the fit of the integrated square error $\epsilon^2 = cN^{-p}$ for the range of N. The same data are plotted in Fig. 4. All discretizations were verified to be stable by running for hundreds of time steps from white noise initial conditions and monitoring $\int \psi(x,y,t)^2 \mathrm{d}x\,\mathrm{d}y$.

Table 3
Number of points used for the discretization of border propagation and boundary condition operators in square propagation region, and the amount by which these exceed the number of points used for the bulk propagation operator

d	M	$M - M_{\text{bulk}}$
2	21	0
4	33	8
6	89	32
8	163	64

Table 4
Integrated square error, $\tau = 1$, mixed boundary conditions

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
16	6.9×10^{-2}	5.6×10^{-4}	5.7×10^{-4}	
24	1.4×10^{-2}	2.0×10^{-5}	3.9×10^{-7}	4.6×10^{-7}
32	5.5×10^{-3}	1.3×10^{-6}	7.3×10^{-9}	6.0×10^{-10}
40	2.9×10^{-3}	1.7×10^{-7}	5.5×10^{-10}	2.7×10^{-11}
48	1.5×10^{-3}	9.0×10^{-8}	1.7×10^{-10}	4.4×10^{-13}
56	1.1×10^{-3}	7.3×10^{-9}	8.0×10^{-12}	7.2×10^{-15}
Convergence	3.3	8.7	13.8	20.5

Fig. 4. Integrated square error at $t = 1/2$ ($N/2$ steps) for a square grid, mixed boundary conditions.

3.3. Circular region with homogeneous Neumann boundary conditions

In this section, I give numerical results for the wave equation in a circular domain with homogeneous Neumann conditions at $\tau = a \approx 1.84118$ (the first root of $J_1(r)$). I compute the error as the deviation from the exact solution

$$\psi(r,t) = J_0(r)\cos t. \tag{32}$$

The circular region of propagation is inscribed in an $N \times N$ square grid. The range of time was $0 \leqslant t \leqslant 2\pi$. The value of τ changed somewhat so that $t = 2\pi$ was reached in an integral number of time steps. For sparse discretization, τ was somewhat less than 1, approaching one as more points were used. Again it was necessary to use more points (than those used for periodic BCs) for the discretization of the operators. The same number of points M were used for all these discretization of border propagation operators and boundary projection operators. The number of points used is tabulated in Table 5. For computation of the degree 8 discretization coefficients, extended precision arithmetic was used. All time stepping was done in normal double precision. Tabulations of the integral of the square of the error as a function of the number N of discretization points in each direction are given in Table 6 for various orders of discretization and $\tau = 1$. They are plotted in Fig. 5. The convergence values given in the tables is p, determined by the fit of the integrated square error $\epsilon^2 = cN^{-p}$ for the range of N.

3.4. Circular region with inhomogeneous Dirichlet boundary conditions

Finally, I give results for a plane wave

$$\psi(x,y,t) = \cos\pi(x-t), \tag{33}$$

using inhomogeneous Dirichlet BCs on a circle of radius 1. As in the previous example, the range of time was $0 \leqslant t \leqslant 2\pi$ and value of τ changed somewhat so that $t = 2\pi$ was reached in an integral number of time steps. The number of points used is tabulated in Table 7. For computation of the degrees 6 and 8 dis-

Table 5
Number of points used for the discretization of border propagation and boundary condition operators in circular propagation region with homogeneous Neumann BCs, and the amount by which these exceed the number of points used for the bulk propagation operator

d	M	$M - M_{\text{bulk}}$
2	21	0
4	33	8
6	65	16
8	193	96

Table 6
Integrated square error, $\tau = 1$, circular boundary with Neumann conditions

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
16	4.0×10^{-4}	4.5×10^{-7}		
24	3.0×10^{-5}	8.2×10^{-9}	6.0×10^{-11}	8.6×10^{-13}
32	3.0×10^{-6}	9.2×10^{-10}	4.2×10^{-13}	1.7×10^{-15}
40	8.4×10^{-7}	5.1×10^{-11}	2.0×10^{-14}	4.6×10^{-17}
48	2.8×10^{-7}	1.4×10^{-10}	1.9×10^{-15}	1.9×10^{-17}
56	1.1×10^{-7}	3.3×10^{-12}	1.3×10^{-16}	1.4×10^{-19}
Convergence	6.6	8.7	15.0	17.1

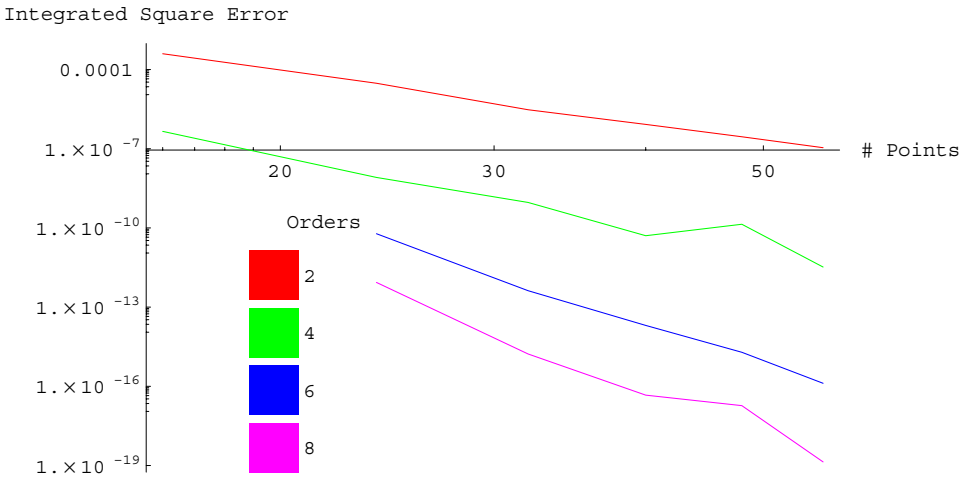

Fig. 5. Integrated square error, $\tau \approx 1$, circular boundary with Neumann conditions.

Table 7
Number of points used for the discretization of border propagation and boundary condition operators in circular propagation region with inhomogeneous Dirichlet BCs, and the amount by which these exceed the number of points used for the bulk propagation operator

d	M	$M - M_{\text{bulk}}$
2	21	0
4	33	8
6	65	16
8	129	118

cretization coefficients, extended precision arithmetic was used. All time stepping was done in normal double precision. Note that, for degree 8, fewer points were required than for homogeneous Neumann conditions (Section 3.3). This is significant, because one might think it harder to achieve stability with inhomogeneous BCs. Instead, it seems that inhomogeneous BCs are no harder than homogeneous ones, but that Dirichlet BCs are easier than Neumann.

Table 8
Integrated square error, $\tau \approx 1$, circular boundary with inhomogeneous Dirichlet conditions

N	$d = 2$	$d = 4$	$d = 6$	$d = 8$
16	1.4×10^{-2}	4.2×10^{-4}		
24	2.1×10^{-3}	1.2×10^{-5}	1.5×10^{-7}	
32	2.3×10^{-4}	1.7×10^{-7}	7.0×10^{-17}	
40	1.3×10^{-4}	1.2×10^{-8}	7.7×10^{-11}	
48	5.9×10^{-5}	1.3×10^{-9}	4.2×10^{-12}	4.0×10^{-14}
56	4.4×10^{-5}	6.2×10^{-10}	6.2×10^{-13}	2.7×10^{-15}
64			6.3×10^{-14}	7.8×10^{-16}
72				8.8×10^{-17}
80				1.9×10^{-17}
88				3.0×10^{-19}
Convergence	4.8	11.4	14.5	17.7

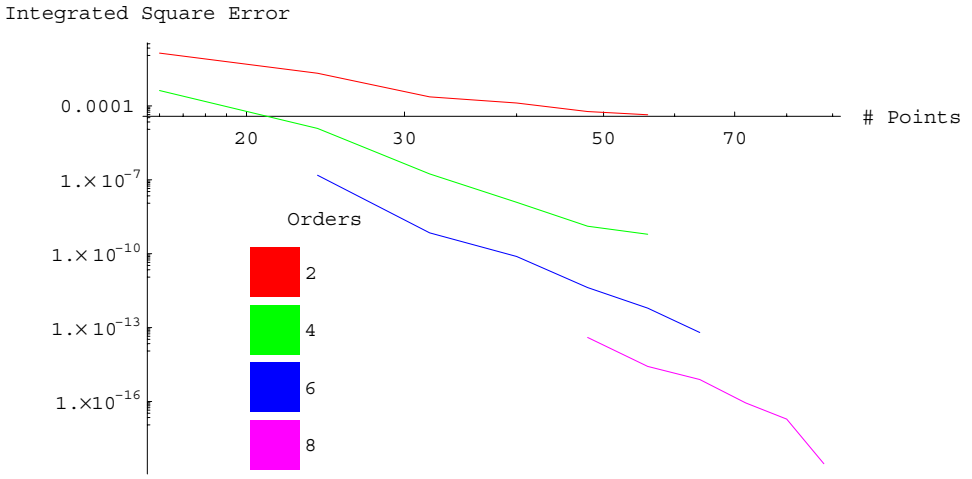

Fig. 6. Integrated square error, $\tau \approx 1$, circular boundary with inhomogeneous Dirichlet conditions.

Tabulations of the integral of the square of the error as a function of the number N of discretization points in each direction are given in Table 8 for various orders of discretization and $\tau = 1$. They are plotted in Fig. 6. The convergence values given in the tables is p, determined by the fit of the integrated square error $\epsilon^2 = cN^{-p}$.

4. Conclusions

The results presented here show how the approach of AGH [1], as implemented in [4], works in $2+1$ dimensions. Although I have not used a tapered mesh, small cells are implicit because of the proximity of the boundary points to the border points. This paper substantiates our previous claim that the method is formulated in a way that is straightforward to implement in any number of dimensions. It further motivates study of the issues raised in [4], namely

General stability analysis: Although I have demonstrated stable high-order examples, the general conditions for this behavior remain unknown. This problem is formidable but very important.

First order formulation: The advantages and disadvantages of using ψ and $\dot{\psi}$ at two time slices instead of just ψ at three time slices.

Parametric efficiency analysis: The optimization of choices such as discretization density, degree of approximation and time step size for desired accuracy.

Tailoring of border mesh: Because the examples of this paper all have fields without singularities at the boundaries, tapering of the mesh was not indicated. For boundaries that induce singularities, efficiency and accuracy will depend on the choice of the border mesh.

Finally, it raises a new issue that did not appear in $1+1$ dimensions, the larger number of points needed in the border region (compared to the bulk) to maintain stability. The dependence of this on discretization order, density, and mesh tapering will be very important in practical applications.

Acknowledgements

This work supported in part by DARPA/DSO under contract #MDA972-01-C-0006.

References

[1] B.K. Alpert, L.F. Greengard, Thomas 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] J.-R. Li, L. Greengard, Strongly consistent marching schemes for the wave equation, Journal of Computational Physics, submitted. http://dx.doi.org/10.1016/j.jcp.2004.01.021.
[4] John Visher, Stephen Wandzura, Amanda White, Stable, high-order discretization for evolution of the wave equation in $1+1$ dimensions, Journal of Computational Physics 194 (2003) 395–408.