Available online at www.sciencedirect.com

SCIENCE DIRECT®

Wave Motion 41 (2005) 263–291

WAVE MOTION

www.elsevier.com/locate/wavemoti

Wave propagation using bases for bandlimited functions

G. Beylkin, K. Sandberg*

Department of Applied Mathematics, University of Colorado at Boulder, 526 UCB, Boulder, CO 80309-0526, USA

Received 7 January 2003; received in revised form 7 May 2004; accepted 10 May 2004

Available online 23 September 2004

Abstract

We develop a two-dimensional solver for the acoustic wave equation with spatially varying coefficients. In what is a new approach, we use a basis of approximate prolate spheroidal wavefunctions and construct derivative operators that incorporate boundary and interface conditions. Writing the wave equation as a first-order system, we evolve the equation in time using the matrix exponential. Computation of the matrix exponential requires efficient representation of operators in two dimensions and for this purpose we use short sums of one-dimensional operators. We also use a partitioned low-rank representation in one dimension to further speed up the algorithm. We demonstrate that the method significantly reduces numerical dispersion and computational time when compared with a fourth-order finite difference scheme in space and an explicit fourth-order Runge–Kutta solver in time.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Wave propagation; Prolate spheroidal wave functions; Bandlimited functions; Efficient operator representations; Matrix exponential; Spectral projectors; Acoustic wave equation

---

1. Introduction

In this paper we demonstrate how to use bases for bandlimited functions in algorithms of wave propagation. Using bandlimited functions allows us to achieve a low sampling rate while significantly reducing numerical dispersion. In addition, we show how to compute and use the matrix exponential as a propagator by employing separated and partitioned low rank representations.

Using bases for bandlimited functions is a significant departure from the usual approach in numerical analysis. For example, the standard notion of the order of approximation is not appropriate in its usual form since in our construction the basis itself is generated for a finite but arbitrary accuracy. We note that the methods we describe in this paper are applicable to many other problems.

---

* Corresponding author. Tel.: +1 303 492 0593; fax: +1 303 492 4066.
E-mail address: kristian.sandberg@colorado.edu (K. Sandberg).

0165-2125/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.wavemoti.2004.05.008

The first step in constructing a numerical scheme is to select a basis for representing solutions and operators. Typically, in spectral and pseudo-spectral methods, the trigonometric functions $\{e^{ik\pi x}\}_{k=0}^{N}$ have been used for periodic, and Legendre and/or Chebyshev polynomials for non-periodic problems. Instead, we consider bandlimited functions on an interval. A basis for bandlimited functions, the prolate spheroidal wave functions (PSWFs), was introduced in the 1960s by Slepian et al. in a series of papers [1–5]. Recently the generalized Gaussian quadratures became available in [6,7], making it possible to construct efficient numerical algorithms for such functions.

We review the construction of three bases for bandlimited functions. First we consider bases $\{e^{ic\theta_k x}\}_{k=1}^{N}$ on the interval $[-1, 1]$, where $|\theta\ |\quad 1$ are the nodes of the generalized Gaussian quadrature constructed for a given precision and bandlimit. We note that these functions are not necessarily periodic. Such bases may not be suitable for some numerical computations (heuristically, they correspond to the basis of monomials). For this reason, we also consider bases of approximate PSWFs and interpolating bases and use them in our computations.

There are at least two deficiencies of orthogonal polynomials in using them for numerical computations. First is the concentration of Gaussian nodes near the end points of the interval. Second is the sampling rate that never approaches, even asymptotically, the rate for periodic functions, namely, π versus two points per wavelength, see e.g. [8]. As it turns out, the nodes of the generalized Gaussian quadratures for exponentials do not concentrate excessively (the rate reported in [6] is in error, see Section 2.2) and the sampling rate asymptotically approaches the rate for periodic functions.

In recent preprints [9,10] the authors present a study of the PSWFs as a tool for solving PDEs. We note that our use of the PSWFs differs in several ways that have a significant impact on the performance. We first select the desired accuracy and then, for a given bandlimit, construct the (nearly) optimal quadratures for these parameters. Alternatively, for a selected accuracy and a given number of nodes, we find the largest possible bandlimit (see discussion in Section 2.2). We note that in [9,10] the number of nodes is selected proportional to the bandlimit, which is not the optimal choice. We also use a different approach to time evolution described below.

An important observation in using the PSWFs is that the norm of the derivative matrix based on bandlimited functions is smaller than that based on polynomials. In constructing derivative operators we incorporate boundary conditions into the derivative matrix. In the case of discontinuous interface conditions, these conditions are also incorporated into the derivative matrix in a way similar to [11]. We also use the spectral projector to remove spurious large eigenvalues and corresponding eigenspaces from the derivative operators, thus further reducing their norm. For time evolution we use a semigroup approach (that involves computing the matrix exponential) and compare it with the standard fourth-order Runge–Kutta method. We note that for time evolution one can also use the approach introduced in [12] or the spectral method in [13]. We will discuss approaches that avoid computing the matrix exponential explicitly elsewhere.

We write the acoustic equation as a first order system [14]. After discretizing the spatial operator, the equation takes the form of the system of linear first order ordinary differential equations:

$$\mathbf{u}_t = \quad \mathbf{u} + \mathbf{F}(t)$$

with the initial condition $\mathbf{u}(0) = \mathbf{u}_0$. In the case of time independent coefficients, the solution is given by

$$\mathbf{u}(t) = e^{tL}\,\mathbf{u_0} + \int_0^t e^{(t-\tau)L}\,\mathbf{F}(\tau)\,d\tau. \tag{1}$$

Using (1) for time evolution requires computing the matrix exponential $e^{\Delta tL}$ for a time step Δt. The computation of $e^{\Delta tL}$ and applying it to a function is costly in dimensions 2 and higher and, therefore, this approach is rarely used for numerical computations.

We use the separated representation introduced in [15] to represent the operator L for problems in two or higher dimensions. This representation significantly reduces the cost of computing the matrix exponential and matrix–vector multiplications. The separated representation of an operator in two or higher dimensions is given by a sum of products of operators acting in one dimension. We refer to the number of terms in the separated representation as the separation rank. The separation rank for the matrix exponential $e^{\Delta tL}$ grows with the size of the time step Δ, and we will see that a time step between one and two temporal periods is appropriate to control both the separation rank and the number of time steps. We note a typical time step in problems of wave propagation is a fraction of a temporal period.

We reduce the computational cost further by using the partitioned low-rank (PLR) representation for operators acting in one dimension. This representation is similar to the partitioned singular value decomposition considered in [16,17]. We note that both the separated and PLR representations are interesting on their own, with applications in other areas, e.g., computational quantum mechanics (see [15,18]).

We note that in [19,13] the authors present a spectral method for applying the matrix exponential without constructing such matrix. Our approach is competitive if the problem has to be solved repeatedly for the same model with different initial conditions. We will consider a comparison of the method in [19,13] with our approach separately.

We begin with a review of the bandlimited functions in Section 2 and construct derivative operators incorporating boundary and interface conditions in the following section. In Section 4 we provide several numerical examples demonstrating the accuracy of the derivative matrix based on bandlimited functions and also construct integration operators with respect to bandlimited functions. In the following section we review the separated representation and the PLR representation, and describe linear algebra algorithms for operators in these representation. We also introduce the PLR representation and describe linear algebra algorithms for operators in this representation. Finally, we apply these tools to solve the acoustic equation in two dimensions in Section 6 and give a number of numerical examples and comparisons.

2. Bandlimited functions and their approximations

In physical phenomena there is always a bound for both the spatial/time extent and the wavenumber/frequency range. However, a function cannot be compactly supported in both the space and the Fourier domain. In order to manage this apparent contradiction, it is natural to consider the basis of eigenfunctions of the space and band limiting operator. This has been the topic of a series of papers by Slepian et al. [1–5], which introduced the prolate spheroidal wave functions (PSWFs) as an eigensystem bandlimited in $[-\ ,\ ]$ and maximally concentrated within the space interval $[-1, 1]$.

The bandlimited periodic functions can be expanded into the Fourier basis $\{e^{ik\pi x}\}_{k=0}^{N}$ or, if we consider zero boundary conditions, into the basis $\{\sin\ (\pi(\ + 1))\ 2\}_{=1}$. However, in order to divide the computational domain into subdomains, we need to allow arbitrary boundary conditions on the subdomains, and neither the Fourier nor the sine basis are then acceptable. This motivates the introduction of a basis that can efficiently represent functions of the type for an arbitrary real value b, such that $|\ |$, where c is a fixed parameter, the bandlimit.

We note that solutions of equations of mathematical physics behave more like exponentials than polynomials. This provides a naive but compelling motivation for using bandlimited functions rather than polynomials, as a tool for approximating solutions. As we demonstrate, for a given accuracy, computing with bandlimited functions significantly reduces the computational cost.

aG

and $= (\ \ 2\pi)\ ^{*}\ \cdot$ [redacted]

$$(\psi)(\ ) = \frac{1}{\pi}\int_{-1}^{1}\frac{\sin(\ (\ \ -\ \ ))}{\ -\ }\psi(\ )\,\mathrm{d}\ \ .$$

The PSWFs are the eigenfunctions of the operators and [redacted]

$$\mu = \frac{\ }{2\pi}|\lambda|^{2}. \tag{3}$$

In our notation we may suppress the dependence of the eigenfunctions and eigenvalues on c.

Let us consider the spaces of bandlimited functions,

$$\mathsf{B}\ = \{\ \ \in\ \ ^{2}(\mathbb{R}) |\ \hat{\ }(\omega) = 0 \text{ for } |\omega| \geq\ \ \}.$$

The PSWFs form a complete basis in $^{2}([-1,1])$ and B [1]. The eigenfunctions $\psi\ (\ )$ are real and orthogonal on both $[-1, 1]$ and $\mathbb{R}$:

$$\int_{-1}^{1}\psi\ (\ )\psi\ (\ )\,\mathrm{d}\ \ = \delta \tag{4}$$

and

$$\int_{-\infty}^{\infty}\psi\ (\ )\psi\ (\ )\,\mathrm{d}\ \ = \frac{1}{\mu}\delta\ \ , \tag{5}$$

where μ are eigenvalues of the operator .

The PSWFs are uniformly bounded on $[-1, 1]$, $\|\psi\ \|\ _{\infty([-1,1])} \leq$, for some constant , for all $= 0, 1, \ldots$ The existence of

Proposition 2. *For* 0 *and* ϵ 0, *we construct nodes* -1 θ_1 θ_2 $\cdots$ θ 1 *and weights* 0, *such that for any* $\in [-1, 1]$:

$$\left| \int_{-1}^{1} e^{ictx}\, dt - \sum_{k=1}^{M} w_k\, e^{ic\theta_k x} \right| \quad \epsilon \tag{7}$$

and the number of nodes, M, is (*nearly*) *optimal. The nodes and weights maintain the natural symmetry,* $\theta \ = -\theta_{\ -\ +1}$ *and* $\ = \ _{\ -\ +1}$.

Thus, we can integrate all functions with using Proposition 2. The nodes and weights in Proposition 2 are computed as a function of the bandlimit 0 and the accuracy ϵ 0 and can be viewed as the generalized Gaussian quadratures for the bandlimited functions. We note that the algorithm in [7] identifies the nodes of the generalized Gaussian quadratures as zeros of the *discrete* prolate spheroidal wave functions (DPSWF) corresponding to small eigenvalues. For a study of DPSWFs we refer to [5].

2.2. On the distribution of nodes for Gaussian quadratures

As it is well known, nodes of Gaussian quadratures (both the usual and generalized) accumulate near the end points as the number of nodes grows. The rate of such accumulation has a critical influence in a variety of applications where quadratures are used either for integration or interpolation.

Although we compute the nodes and weights as in [7] by selecting first the bandlimit, *c*, and then computing the minimal (or nearly minimal) number of nodes, *M*, to achieve a given accuracy ϵ, once such quadratures are generated we use the number of nodes as the variable and $= (\ , \epsilon)$ to study node accumulation.

Let us consider the ratio

$$(\ , \epsilon) = \frac{\theta_2 - \theta_1}{\theta_{\lfloor\ 2\rfloor} - \theta_{\lfloor\ 2\rfloor - 1}}, \tag{8}$$

where "$\lfloor\ 2\rfloor$" denotes least integer part. Observing that the distance between nodes of the Gaussian quadratures changes monotonically from the middle of the interval toward the end points, and that the smallest distance is between the two nodes closest to an end point, this ratio can be used as a measure of node accumulation. For example, the distance between the nodes near the end points of the standard Gaussian quadratures for polynomials decreases as $O(1\ \ ^2)$, where *M* is the number of nodes, so that we have $(\ , \epsilon) = O(1\ \ )$.

Using the method in [7] example,[r,

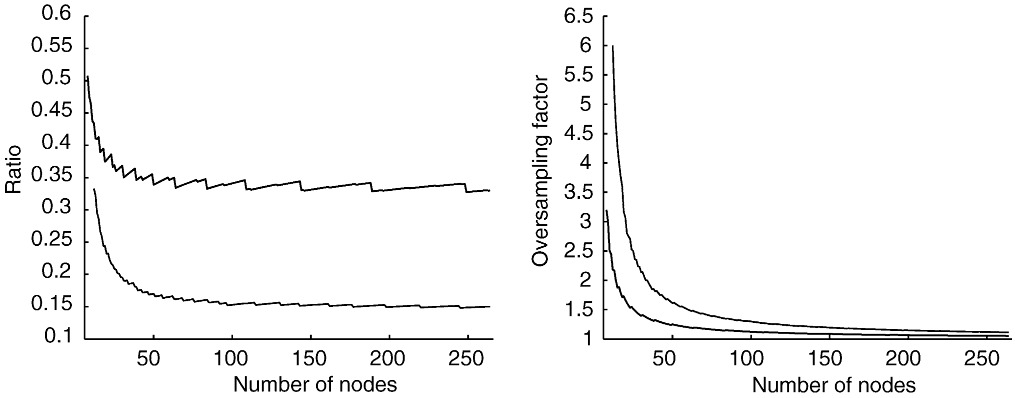

Fig. 1. The ratio (, ϵ) in (8) and the oversampling factor $\alpha($, $\epsilon)$ plotted against the number of nodes for quadratures of accuracy $\epsilon \approx 10^{-7}$ and $\approx 10^{-17}$

We have $E \subseteq {}^{\infty}([-1,1])$ and prove the following theorem (see Appendix A)

Theorem 3. *For every* ϵ 0 *and* , $\in B$ *there exists a function* $\tilde{}$ $\in E$, *such that* $\| - \tilde{} \|_{2([-1,1])}$ ϵ.

Any bandlimited function from E can be approximated by a linear combination of a finite number of exponentials in the form $e^{ic\theta_k x}$ where $|\theta| \leq 1$. The phases θ are chosen as nodes of the generalized Gaussian quadratures ([7, Theorem 6.1], see also [6]). Following [7], we use the quadrature nodes and weights to construct bases for E .

Theorem 4. *Consider* $\in E$,

$$( ) = \sum_{\in} \quad e^{ib_k x}$$

and let $\{\theta\}_{=1}$ *and* $\{ \}_{=1}$ *be quadrature nodes and weights for the bandlimit* 2 *and accuracy* ϵ^2. *Then there exist coefficients* $\{ \}_{=1}$ *and a constant A such that*

$$\left\| ( ) - \sum_{=1} \, e^{ic\theta_l x} \right\|_{\infty([-1,1])} \leq A \left(\sum_{\in} | \quad | \right) \epsilon.$$

The set of exponentials $\{e^{ic\theta_k x}\}_{k=1}^{M}$ may be viewed as a basis for bandlimited functions E with accuracy ϵ. The basis of exponentials has the obvious advantage of being easy to differentiate and integrate but these functions are far from being orthonormal and one must be careful using them for numerical computations. In this respect they are analogous to monomials as a basis for polynomials. In order to construct a basis analogous to the orthogonal polynomials, we turn to the PSWFs.

Instead of using the PSWFs directly, we choose to construct their approximations [7], as it is sufficient for our purposes. Given the bandlimit 0 and accuracy threshold ϵ 0, let us construct quadrature nodes and weights according to Theorem 4. We then solve the algebraic eigenvalue problem:

$$\sum_{=1} \, e^{ic\theta_m\theta_l} \, \boldsymbol{\Psi}_j(\theta_l) = \eta_j \boldsymbol{\Psi}_j(\theta_m) \tag{10}$$

and define the approximate PSWFs on $[-1,1]$ by

$$\Psi ( ) = \frac{1}{\eta} \sum_{=1} \, e^{icx\theta_l} \, \boldsymbol{\Psi}_j(\theta_l), \tag{11}$$

where $\psi(\theta)$ are the eigenvectors in (10).

The matrix in (10) does not have zero eigenvalues as can be easily checked numerically although we do not have a proof for this fact. We expect the eigenvalues $\{\eta\}_{=1}$ to approximate the first *M* eigenvalues $\{\lambda\}$ and eigenvectors of [redacted], where the relative error may be large. Since the absolute values of the first $\simeq 2 \quad \pi$ eigenvalues in (2) are very close, some of them are numerically indistinguishable (within the machine precision). As a result, we do not construct approximations to the individual PSWFs via (10) but, instead, approximate correctly the subspace spanned by these functions.

Let us consider the inner products of functions in (11):

$$= \int_{-1}^{1} \Psi ( ) \Psi ( ) \, d \tag{12}$$

for , $= 1, \ldots,$. We have the following proposition (see [7], Proposition 8.1).

3. Derivative matrices with boundary and interface conditions

Table 2

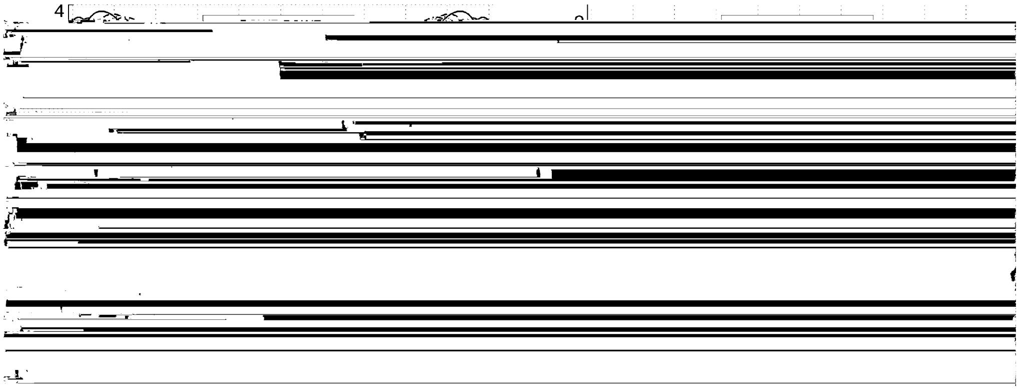

Fig. 5. Absolute (left) and relative (right) errors for the first derivative of the function in the interval $[-1, 1]$ with $| \ | \leq 16\pi$ using a basis of 32 approximate PSWFs.

10^{-10}, 10^{-7}, and 10^{-4}, with the corresponding bandlimits c set to 5.5π, 7π, 8.5π, and 10.5π, respectively. We differentiate the function $(\ ) =$

where ⊗ denotes the Kronecker product, $\{\boldsymbol{A}\ (\ _1,\ '_1)\}_{=1}$ and $\{\ (\ _2,\ '_2)\}_{=1}$ are $\times$ matrices, $\|\boldsymbol{A}\ \| = 1$, $\|\ \| = 1$, $_{,}$ 0, and

$$\left\|\ (\ _1,\ '_1,\ _2,\ '_2) - \sum_{=1}^{,} \boldsymbol{A}\ (\ _1,\ '_1) \otimes\ (\ _2,\ '_2)\right\| \leq \epsilon.$$

The number of terms in the representation, *r*yt2 1 Tf 1.03185.2.54(<

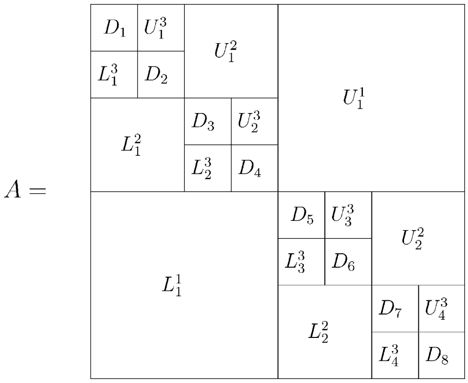

Fig. 9. Matrix subdivision for the 3-level PLR representation. The diagonal blocks are stored as full matrices and whereas the off-diagonal blocks are of low rank and are represented accordingly.

5.2. *The partitioned low rank (PLR) representation*

The partitioned low rank (PLR) representation is a simplification of the partitioned singular value decomposition (PSVD) introduced in [16,29,30], and used for spectral projectors in [17]. The PSVD is simplified by dropping the requirement of orthogonality between vectors as implied by the SVD and using a much simpler algorithm for rank reduction. The PSVD and PLR are more flexible than wavelet decompositions and are applicable to a wider class of matrices. In particular, the exponential of a matrix with pure imaginary spectrum and the bandlimited derivative matrix constructed in Section 4 are of high rank, dense, non-Toeplitz, with entries oscillatory as functions of indices. Unlike operators with real, negative spectrum, exponentials of such operators are not necessarily compressible via the wavelet transform while the PLR representation is efficient even when wavelet or multiwavelet transforms are dense. In Section 6 we apply PLR representation to exponentials of operators with pure imaginary spectrum (propagators) and its representation remains efficient for propagation over 1–2 periods (wavelengths).

The PLR representation is defined recursively by splitting a matrix into four blocks. The two diagonal blocks are split further, whereas the two off-diagonal blocks are maintained using a low rank representation of the form $\sum \sigma$ *. The 3-level PLR representation is illustrated in Fig. 9, where we use , and to denote the diagonal, upper and lower blocks of the partitioned matrix at different levels. This notation is convenient when describing linear algebra operations in the PLR representation.

In all our computations for a given accuracy ϵ 0, we seek an approximation $\tilde{\boldsymbol{A}}$ of an operator A such that $\|\boldsymbol{A} - \tilde{\boldsymbol{A}}\|$ ϵ, where $\| \cdot \|$ is an operator norm. For many operators in the PLR representation, the coefficients of

$\boldsymbol{A}\boldsymbol{A}$

Let a matrix A be in an m-level PLR representation. In order to compute the matrix–vector product A , it is clear from Fig. 9 that there are two types of matrix–vector multiplications we need to evaluate. We need to compute the dense matrix–vector products for the diagonal blocks and the matrix–vector products $\tilde{}$ and $\tilde{}$, where $\tilde{} \in \mathbb{C}$, $=$ 2 . If $= \sum_{=1}$ *, then the matrix–vector product is computed as

$$= \sum_{=1} \langle \, , \, \rangle \, . \qquad (29)$$

The cost of such matrix–vector multiplication is O( ). Assuming that the rank of all off-diagonal blocks is the same, the total cost of computing A is then estimated as $2 \; ^{2} + \sum_{=1} 2$. If the total size of A is $= 2$,

6. Solution of the acoustic equation in two dimensions

Let us consider the acoustic equation

$${}' \; {}_{``} = {}^{1}$$

a numerical scheme for solving the homogeneous acoustic Eq. (30) in two dimensions with time independent coefficients and boundary conditions. We proceed via the following steps:

(1) Construct the derivative matrices representing [illegible] and [illegible] using the results in Section 3.
(2) Construct separated representations of the multiplication operators 1 $\kappa(\, , \,)$ and $\sigma(\, , \,)$
(3) Construct blocks of the 3×3 spatial operator in (31) and use the algorithm in Section 5.3 to reduce the separation rank of each block.
(4) Select the time step Δ (see below) and compute the matrix exponential $e^{\Delta tL}$ using the scaling and squaring algorithm (see e.g. [31]). The linear combinations and products of the matrix blocks given in the separated representation are computed using the methods described in Section 5.
(5) Compute the solution $\mathbf{u}(\,) = e^{\Delta tL}\,\mathbf{u}(t_{k-1})$, starting from $\mathbf{u}(\,_0) = \mathbf{u}(0)$, for $\; = 1, \ldots, \;_{\text{time}}$.

We refer to this algorithm as the method of bandlimited bases (MBB) with the exponential propagator (EP). We note that for time dependent boundary conditions the problem can be reduced to that with zero boundary conditions and a forcing term.

If the factors in the separated representation (which are ordinary matrices) are large, we use the PLR representation described in Section 5.2 to speed up the computations in Steps 4 and 5 above. As discussed in Section 4, the norm of the derivative projectors can be greatly reduced by using spectral projectors. For example, to construct projected versions of [illegible] and [illegible] (derivative operators in the [illegible]-direction), we form the block matrix,

$$ = \begin{bmatrix} 0 & _0 \\ & 0 \end{bmatrix},$$

where D and $\;_0$ are constructed as in Section 3. The boundary conditions are enforced only for the function u in (31) which results in the structure of the matrix L above. We then construct the projected operator,

$$_{\text{proj}} = \sum_{|\lambda \; | \leq} \lambda \; \mathbf{e} \; \mathbf{f}^{T},$$

where $\mathbf{e}$ [illegible]

the second comparison we write the acoustic equation (30) as a first order system:

$$\begin{bmatrix} \\ , \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{\kappa}\left[\frac{}{}\left(\sigma \frac{}{} \right) + \frac{}{}\left(\sigma \frac{}{} \right) \right] \\ & 0 \end{bmatrix} \begin{bmatrix} \\ , \end{bmatrix} + \Bigg[$$

[redacted]

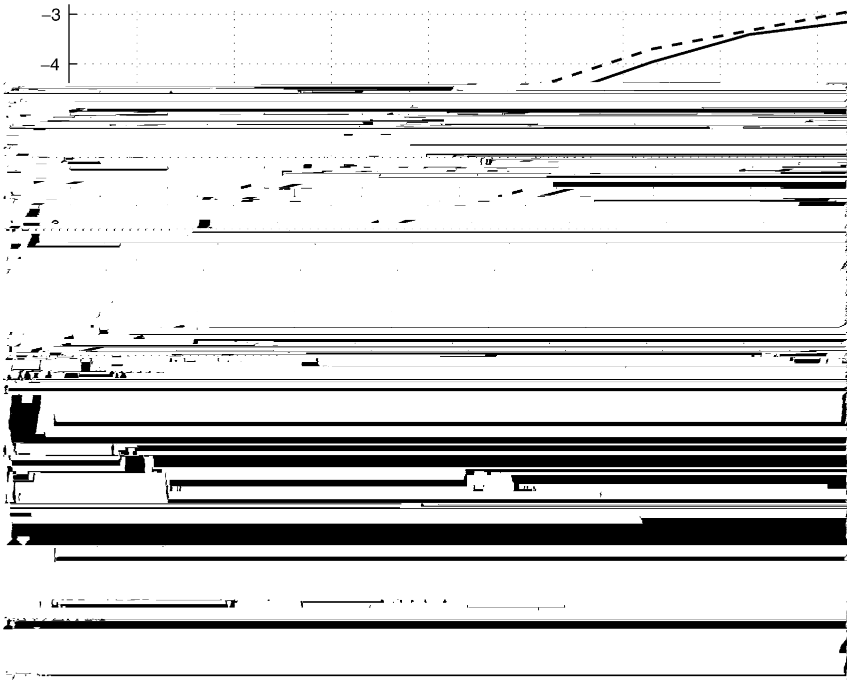

Fig. 10. Relative error in the max-norm for approximating the solution to (34) (top) and the computational time (bottom).

discretization as for the MBB with EP, but use the RK4 solver in time with the timestep Δt 128. The results are shown in Fig. 10.

In order for the finite difference fourth order scheme to reach similar accuracy, we need more than 1024 samples in space corresponding to an oversampling factor of approximately 22 (for periodic functions) and a timestep $= \Delta t$ 128. With this sampling rate, the computational time per characteristic period is almost 3 min, or more than 5000 times slower than using the MBB with EP. However, such oversampling factor is significantly larger than is typically used. For this reason, in the next experiment we solve the same equation, but use 400 samples in space for the fourth order scheme, corresponding to an oversampling factor of approximately 8.7 (compared for the Nyquist frequency for periodic functions), and a timestep Δt 32. The results are shown in Fig. 11. In this experiment, the computational times for the two methods are comparable, but the MBB with EP is significantly more accurate.

In the next experiment, we demonstrate that the cost of improving accuracy is small for the MBB with EP. Let us fix $= 19.5$, and solve the model problem (34) using the MBB with EP for the bandlimit $= 20\pi$ with 52, 56, 60, 64, and 68 nodes. For all solutions, we use the time step $\Delta t = \sqrt{2}$ 20 (approximately 1.4 characteristic periods). The result is shown in Fig. 12.

We observe that using 60 nodes takes approximately two times longer than using 52 nodes but gives approximately 4 more digits of accuracy. We also note that the error increases linearly over time.

6.1.2. Numerical dispersion

Due to inaccuracies of differentiation, the different Fourier modes of a pulse propagate with different speeds. After some time the shape of the pulse deteriorates.

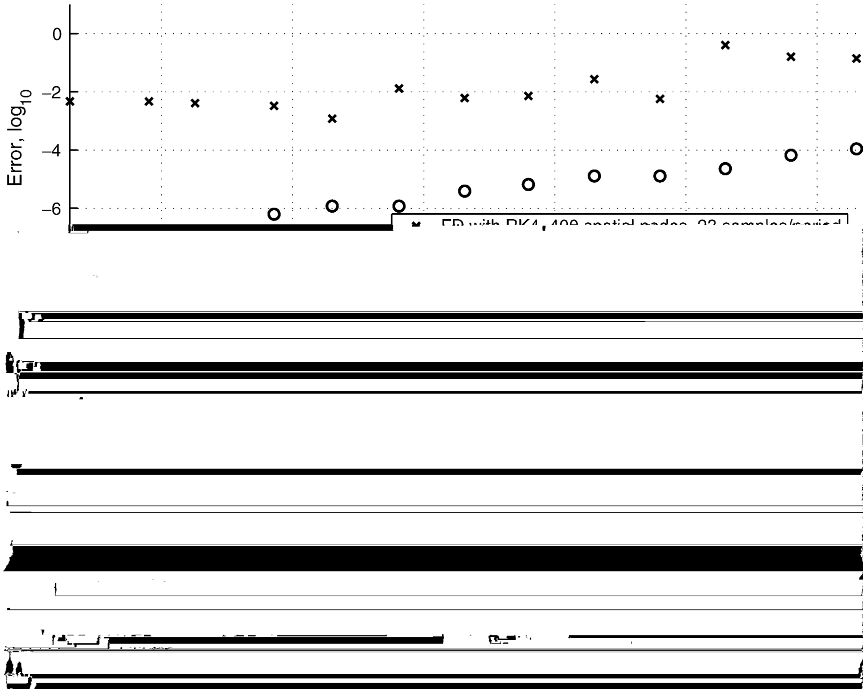

Fig. 11. Relative error ($\log_{10}$) in the max-norm for approximating the solution to (34) (top), and the computational time (bottom).

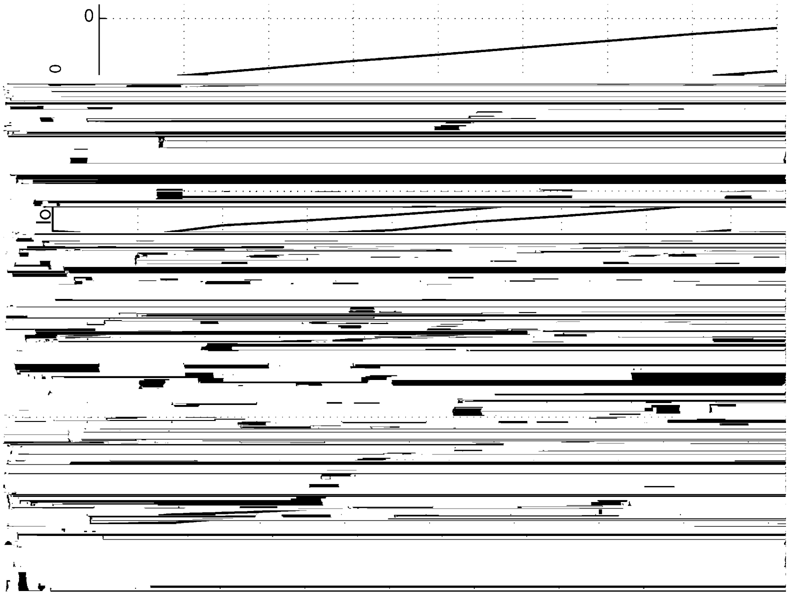

Fig. 12. Relative error in the max-norm for approximating the solution to (34) for = 19.

To examine this phenomenon, let us consider the wave equation in one dimension, $u_t + u_x = 0$, the solutions of which correspond to right-traveling waves. Solutions of this equation take the form:

$$u(x, t) = \mathrm{e}^{\mathrm{i}\omega(x-ct)},$$

which we refer to as a Fourier mode of frequency ω traveling to the right with velocity c. Exact differentiation of this solution yields

$$\frac{\partial u}{\partial x} = \mathrm{i}\omega\,\mathrm{e}^{\mathrm{i}\omega(x-ct)}.$$

If the error in the representation of the differentiation operator is of the form

$$\frac{\partial u}{\partial x} \simeq \mathrm{i}\, f(\omega)\,\mathrm{e}^{\mathrm{i}\omega(x-ct)},$$

then the Fourier mode propagates with the velocity $f(\omega)/\omega$. Unless $f(\omega) = \omega$, which corresponds to the exact differentiation, the Fourier modes of different frequencies travel with different velocities. For example, in the case of the second order centered finite difference approximation of the derivative, $f(\omega) = \sin(\omega)$.

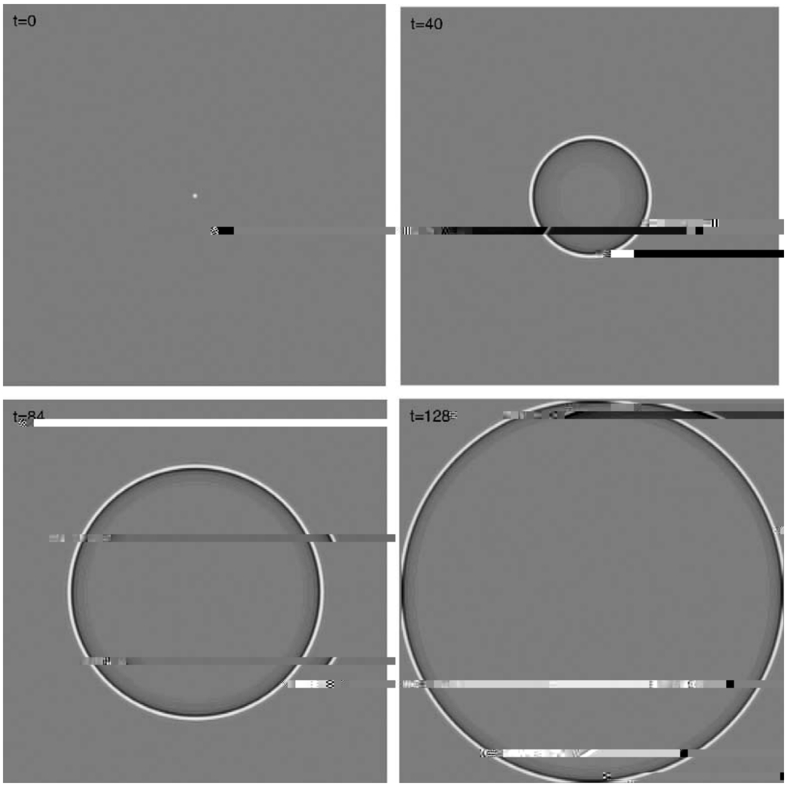

Fig. 13. Solution of (35) using the MBB with EP. The shape of the pulse is maintained throughout the propagation.

6.1.3. Numerical results for variable coefficients

Let us consider the acoustic equation with variable coefficients. Since we do not have an analytical solution, we simply display a sequence of images and study the shape of the pulse as it propagates throughout the domain. Let us solve

$$\begin{aligned} &\ = \frac{1}{\kappa(\ )}(\ \ +\ \ ), \quad (\ ,\ ) \in (-1,1)\times(-1,1), \qquad (\ ,\ ,0) = \mathrm{e}^{-1000(\mathrm{x}^2+\mathrm{y}^2)},\\ &\ (\pm 1,\ ) = \ (\ ,\pm 1) = \ (\ ,\ ,0) = 0, \end{aligned} \tag{36}$$

where

$$\kappa(\ ) = \frac{1}{1 - \sin(\pi(\ +1))\ 2}.$$

The solution is a sharp pulse originating at the origin of the domain, and expanding outwards in the medium with varying velocity. For the MBB with EP, we construct 128 quadrature nodes and weights for the bandlimit $\ = 54\pi$. We set the accuracy in the construction to $\epsilon = 10^{-7}$. We use the time step $\Delta\ = 2\pi$ corresponding to propagating over two characteristic wavelengths. This choice of parameters yields the separation rank either $\ = 7$ or 8 for the blocks of the exponential operator. Using the PLR representation for computing $\mathrm{e}^{\Delta\mathrm{tL}}$ u is in this case approximately 25% faster than using the dense representation of matrices in one dimension. The gain .8(one)-tely

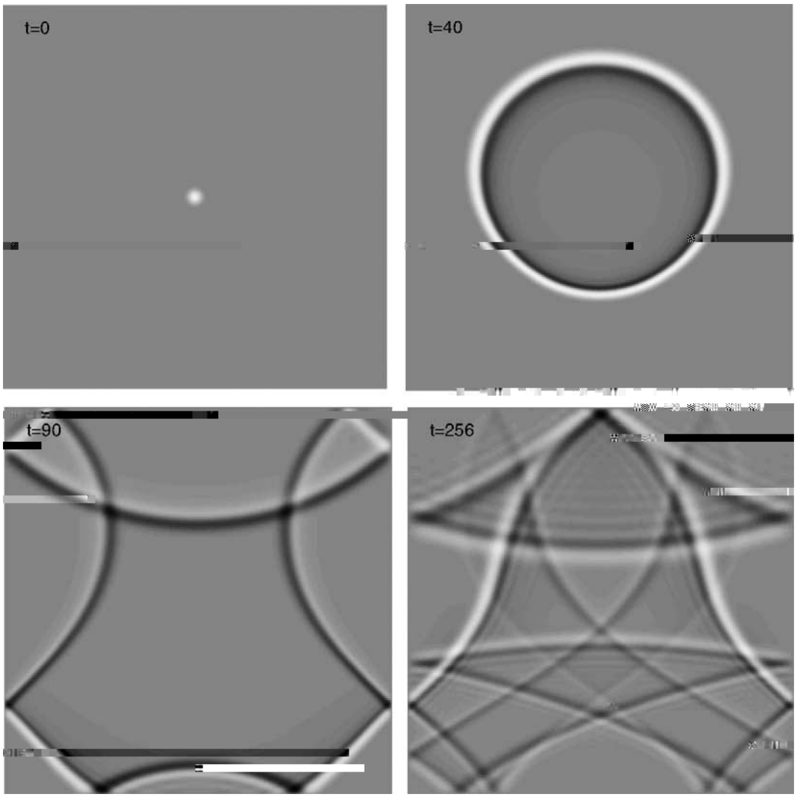

Fig. 16. Solution of (35) using the FD with RK4. Note the ripples which are caused by numerical dispersion.

to propagating over one-tenth of the characteristic wavelength. This sampling rate gives the two schemes approximately the same computational time. The results are shown as sequences of images in Figs. 15 and 16.

Both solutions behave qualitatively in the same way by propagating faster in the upper part of the domain where the wave velocity is higher. We note that for the MBB with EP, the shape of the pulse is maintained. For the FD with RK4, the pulse begins to noticeably deteriorate, as the error accumulates due to the numerical dispersion.

where lim $_{\to\infty}$ $(\ ) = 0$ for all $\in [-1, 1]$. We choose *N* sufficiently large, such that $\| \quad \|_{\infty[-1,1]} \quad \epsilon\ 2\sqrt{2}$ and define for $\in [-1, 1]$

$$\tilde{\imath}(\ ) = \sum_{=1} \mathrm{e}^{\mathrm{i}b_k x},$$

where $= 2\ \sigma(\ )$. Then $\tilde{\imath}$ is bounded on $[-1, 1]$ and $|\ (\ ) - \tilde{\imath}(\ )| \quad \epsilon\ 2\sqrt{2}$ almost everywhere.

Next we consider a function $\in \mathsf{B}$. Then, since $\mathsf{B} \cap {}^1(\mathbb{R}$

[14] J. Bazer, R. Burridge, Energy partition in the reflection and refraction of plane waves, SIAM J. Appl. Math. 34 (1978) 78–92.

[15] G. Beylkin, M.J. Mohlenkamp, Numerical operator calculus in higher dimensions, Proc. Natl. Acad. Sci. USA 99 (16) (2002) 10246–10251.

[16] P. Jones, J. Ma, V. Rokhlin, A fast direct algorithm for the solution of the Laplace equation on regions with fractal boundaries, J. Comput. Phys. 113 (1) (1994).

[17] G. Beylkin, N. Coult, M.J. Mohlenkamp, Fast spectral projection algorithms for density-matrix computations, J. Comput. Phys. 152 (1) (1999) 32–54.

[18] G. Beylkin, M.J. Mohlenkamp, Algorithms for numerical analysis in high dimensions, University of Colorado, Applied Mathematics Preprint # 519, 2004, SIAM J. Sci. Comp. (2004), in press.

[19] H. Tal-Ezer, R. Kosloff, An accurate and efficient scheme for propagating the time dependent Schrödinger equation, The J. Chem. Phys. 81 (9) (1984) 3967–3971.

[20] D. Slepian, Some comments on Fourier analysis, uncertainty and modeling, SIAM Rev. 25 (3) (1983) 379–393.

[21] Y. Meyer, Wavelets and operators, Cambridge Studies in Advanced Mathematics, vol. 37, Cambridge University Press, Cambridge, 1992 (transl.; from the 1990 French original by D.H. Salinger).

[22] T. Rivlin, Chebyshev Polynomials, 2nd ed., Wiley, 1990.

[23] D. Kosloff, H. Tal-Ezer, A modified Chebyshev pseudospectral method with an O($^{-1}$) time step restriction, J. Comput. Phys. 104 (2) (1993) 457–469.

[24] J.S. Hesthaven, P.G. Dinesen, J.P. Lynov, Spectral collocation time-domain modeling of diffractive optical elements, J. Comput. Phys. 155 (2) (1999) 287–306.

[25] B. Fornberg, A Practical Guide to Pseudospectral Methods, Cambridge University Press, 1995.

[26] J.P. Boyd, Chebyshev and Fourier Spectral Methods, 2nd ed., Dover, Mineola, NY, 2001.

[27] L. Greengard, Spectral integration and two-point boundary value problems, SINUM 28 (4) (1991) 1071–1080.