UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Fast Wavelet Based Algorithms for Linear Evolution Equations

Bjorn Engquist
Stanley Osher
Sifen Zhong

February 1992

CAM Report 92-10

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

Fast Wavelet Based Algorithms for Linear Evolution Equations

Bjorn Engquist, Stanley Osher, and Sifen Zhong[1]
Department of Mathematics
University of California, Los Angeles
Los Angeles, CA 90024

Abstract

We devise a class of fast wavelet based algorithms for linear evolution equations whose coefficients are time independent. The method draws on the work of Beylkin, Coifman, and Rokhlin [1] which they applied to general Calderon-Zygmund type integral operators. We apply a modification of their idea to linear hyperbolic and parabolic equations, with spatially varying coefficients. A reduction of asymptotic complexity over standard methods is obtained when applied to hyperbolic equations in one space dimension and parabolic equations in multidimensions.

Key words: Wavelets, hyperbolic, parabolic, numerical methods.

1991 MSC subject classification: Primary 65M06, secondary 65M12

1. Introduction. During the last few years a number of fast computational algorithms have been developed for elliptic problems. These are techniques for which the number of arithmetic operations needed are close to linear as a function of the number of unknowns. Examples of algorithms of such complexity are multigrid methods and the so-called fast Poisson solvers. The fast multipole method and wavelet based methods for elliptic problems formulated as integral equations also belong to this category [8], [1].

There has not been the same progress for hyperbolic and parabolic methods. In general classical numerical techniques for these problems are already optimal.

[1]Research supported by ONR Grant #N00014-91-J-1034.

Consider a system of evolution equations.

$$\begin{aligned} &\partial_t u + L(x, \partial_x)u = f(x), \quad x \in \Omega \subset \mathbf{R}^d, \quad t > 0, \\ &u(x, 0) = u_0(x), \end{aligned} \tag{1.1}$$

with boundary conditions, where L is a differential operator.

An explicit discretization of this problem typically takes the form,

$$\begin{aligned} &u_j^n \approx u(x_j, t_n), \qquad t_n = n\Delta t, \\ &\qquad\qquad\qquad\quad x_j = (j_1\Delta x_1, \dots, j_d\Delta x_d) \\ &u^{n+1} = Au^n + F, \\ &u^0 = u_0, \\ &u, F \in \mathbf{R}^{N^d}, \qquad \Delta t = \text{const. } |\Delta x|^r. \end{aligned} \tag{1.2}$$

The vector u^n contains all the unknowns u_j^n at time level t_n. For simplicity we shall assume $j_\nu = 1, 2, \dots, N$ in all dimensions $\nu = 1, \dots, d$.

The matrix A is $(N^d \times N^d)$ with the number of elements $\neq 0$ in each row and each column bounded by a constant. Every time step requires $\mathcal{O}(N^d)$ arithmetic operations and the overall complexity for a time interval of $\mathcal{O}(1)$ is of the same order as the number of unknowns, $\mathcal{O}(N^{d+r})$.

There are, however, some fast methods based on the analytic form of the solution operator. In [3] the multidimensional heat operator, with u_0 and f both zero, but with inhomogeneous boundary data given at M points for N time levels, was treated. There the closed form of the solution evaluated at M points at each time level N was obtained in $O(NM)$ rather than $O(N^2M^2)$ operations. Also, in [4], the same authors obtained an algorithm for evaluating the sum of N Gaussians at M arbitrarily distributed points in $O(N+M)$ operations. So far, their interesting method appears to need an explicit analytic representation of the heat kernel, effectively ruling out variable coefficient problems.

The formula (1.2) has a simple closed form solution

$$u^n = A^n u_0 + \sum_{\nu=0}^{n-1} A^\nu F. \tag{1.3}$$

This form can be used to compute the solution $A^n u_0$, for $F = 0$, in $\log n$ steps, ($n = 2^m$, m integer; here and throughout, $\log n = \log_2 n$) by repeated squaring of A : $A, A^2, A^4, A^8, \dots, A^{2^m}$.

Unfortunately the later squarings involve almost dense matrices and the overall complexity is $O(N^{3d}\log N)$ which is larger than that using (1.2) directly.

For an appropriate representation of A in a wavelet basis all of the powers A^ν may be approximated by sparse matrices and the algorithm using repeated squaring should then be advantageous.

We shall consider the following algorithms for the computation of the closed form solution (1.3) of the inhomogeneous problem in $m = \log n$ steps,

$$\begin{aligned} &B := SAS^{-1} \\ &C := I \\ &\left.\begin{aligned} C &:= TRUNC(C + BC, \varepsilon) \\ B &:= TRUNC(BB, \varepsilon) \end{aligned}\right\} \quad \text{(iterate } m \text{ steps)} \\ &u^n := S^{-1}(BSu^0 + CSF) \end{aligned} \tag{1.4}$$

The matrix S corresponds to a fast transform of wavelet type and the truncation operator sets elements in a matrix to zero if their absolute value is below a given threshold.

$$\tilde{A} = TRUNC(A, \varepsilon) : \begin{cases} \tilde{a}_{ij} = a_{ij} & |a_{ij}| \geq \varepsilon \\ \tilde{a}_{ij} = 0 & |a_{ij}| < \varepsilon \end{cases} \tag{1.5}$$

It is easy to see that algorithm (1.4) is equivalent to (1.3) for $\varepsilon = 0$. This is not so for $\varepsilon > 0$. We shall however show that it is possible to choose ε small enough for the result of (1.4) to be arbitrarily close to (1.3) but still with very few arithmetic operations.

For a fixed predetermined accuracy level the computational complexity to calculate a one dimensional hyperbolic equation can be reduced from the standard $O(N^2)$ to $O(N(\log N)^3)$. The extra cost per time step is minimal. This also makes it possible, as a curiosity, to use algorithms which are unstable in the traditional sense.

Our technique is more favorable for parabolic problems. A d-dimensional explicit calculation with standard complexity $O(N^{d+2})$ may be reduced to $O(N^d(\log N)^3)$.

The algorithm (1.4) can be extended to some problems with time dependent data. In this case, we clearly need to compress the information in the data such that not all the $O(N^{d+r})$ values in, e.g. the inhomogeneous term $f(x_j, t_n)$ are needed.

One simple but important application of this type is from optics or electro-magnetic scattering with a time periodic source. If k points are needed to resolve one time period,

we can group k time steps together

$$u^{n+k} = A^k u^n + \sum_{j=0}^{k-1} A^j F_{n+k-j-1}. \tag{1.6a}$$

where

$$F_n = \Delta t f(t_n). \tag{1.6b}$$

This equation is now of the type (1.2) with time step $k\Delta t$ and with inhomogeneous term

$$F = \sum_{j=0}^{k-1} A^j F_{n+k-j-1}. \tag{1.6c}$$

In sections 2 and 3 we shall discuss the analytical properties of the algorithm. Numerical examples are presented in section 4.

2. Hyperbolic Problems. Consider first the simple one dimensional scalar advection equation,

$$\begin{aligned} \partial_t u + a\partial_x u = 0, \quad & a > 0 \\ u(x,0) = u_0(x), \quad & 0 \le x \le 1. \end{aligned} \tag{2.1}$$

The functions u_0 and thus u are assumed to be 1-periodic in x. The solution of (2.1) is given by:

$$u(x,t) = u_0(x - at). \tag{2.2}$$

The different rows of A^ν in a numerical solution of (2.1) will represent approximations of the Green's function G below,

$$\begin{aligned} u(x,t) &= \int_{-\infty}^{\infty} G(x,y,t) u_0(y) dy, \\ u(x,t) &= \int_{-\infty}^{\infty} \delta(x - y - at) u_0(y) dy. \end{aligned} \tag{2.3}$$

Let φ_J be a truncated wavelet expansion of a δ-function with an orthonormal set of compactly supported wavelets,

$$\delta(x) \sim \varphi_J(x) = \sum d_k^j 2^{(\frac{\nu-j}{2})} \psi(2^{\nu-j}x - k + 1) + s_1^\nu \varphi(x)$$

The choices of $\psi(x)$ and the resulting $\varphi(x)$ will be discussed below. Assume that the rows of A^ν are discrete δ-functions, i.e. just one element is nonzero and large. For each level

$j = 1, 2, \ldots, J$ there are only a finite number of $d_k^j \neq 0$ since the wavelets are compactly supported. With $J = m = \log N$ there is only $\log N$ of all $d_k^j \neq 0$. Thus each row in B, (1.4), has $\log N$ elements, $b_{jk} \neq 0$. The matrix B^2 is also a transform of an idealized matrix A^ν and will have $N \log N$ elements different from zero. This means that each iteration step in the algorithm (1.4) produces $O(N(\log N)^2)$ flops when $F = 0$. We have assumed that calculations are only carried out for those B^2 elements which are different from zero. In practice a slightly larger number of elements needs to be computed and then truncated. This corresponds to the case when the location of the δ-functions is only approximately known. Compare the wavelet technique for Burgers' equation by Maday, Perrier, and Ravel [6].

Each row of C, (1.4), is a transform of a step function,

$$\tilde{c}(x) = \begin{cases} \text{const.} & 0 \leq x \leq at, \\ 0, & \text{else} \end{cases}$$

Since $\tilde{c}(x) \not\equiv$ constant only at $x = at$, this function can also be represented by $\log N$ wavelets and thus *the overall cost is* $O(N(log\ N)^3)$.

In numerical computations the rows of A^ν are only approximations of δ-functions. If an upwind scheme,

$$\begin{aligned} u_j^{n+1} &= u_j^n - \lambda(u_j^n - u_{j-1}^n), \\ u_j^0 &= u_0(x_j), \qquad j = 1, 2, \ldots N, \\ \lambda &= a\Delta t/\Delta x < 1, \end{aligned} \tag{2.4}$$

is used A will have the form,

$$A = \begin{bmatrix} 1-\lambda & 0 & \cdots & & & \lambda \\ \lambda & 1-\lambda & 0 & \cdots & & 0 \\ 0 & \lambda & 1-\lambda & 0 & \cdots & 0 \\ & & & & & \\ & & & & & \\ 0 & \cdots & 0 & \lambda & & 1-\lambda \end{bmatrix}$$

The matrix A^ν will have Toeplitz structure. Each row is still an approximation of a δ-function. The first order smoothing effect of (2.4) is given by the modified equation, [5],

$$\partial_t u + a\partial_x u = (a\Delta x/2)\partial_x^2 u. \tag{2.5}$$

Equation (2.5) is parabolic with a fundamental solution of the form,

$$G(x-y,t) = (2\pi a \Delta x t)^{-\frac{1}{2}} \exp(-(x-y-at)^2/(2a\Delta x t)). \tag{2.6}$$

Compare the solution formula for parabolic problems (3.2).

Each row of A^ν is thus a close approximation to the function $G(x-y,t)$ above. The computational complexity of the algorithm (1.4) depends on how many wavelets are needed to represent $G(x-y,t)$ as a function of $x, (0 \le t \le T)$ with a given accuracy.

Higher order accurate (say order 2p-1) dissipative finite difference approximations to (2.1) are usually modelled by the equation

$$u_t + a u_x = (-1)^{p+1} k_p (\Delta x)^{2p-1} \left(\frac{\partial}{\partial x}\right)^{2p} u. \tag{2.7}$$

with $k_p \ge \delta > 0, \delta$, independent of Δx.

The fundamental solution for this parabolic equation is:

$$G_p(x,t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\xi \exp(i\xi(x-at) - k_p(\Delta x)^{2p-1}\xi^{2p}t)).$$

The key, simple estimate we shall obtain here (and which we certainly do not claim is new) is:

$$\left| x^{m+1} \left(\frac{\partial}{\partial x}\right)^m G_p(x+at,t) \right| \le C_{m,p} \tag{2.8}$$

uniformly in $0 < t$ and Δx and for all nonnegative integers m.

Proof of 2.8. We wish to bound

$$\begin{aligned}
&\frac{1}{2\pi} \int_{-\infty}^{\infty} (i\xi)^m x^{m+1} e^{i\xi x - k_p(\Delta x)^{2p-1} t\xi^{2p}} d\xi. \\
&= \frac{i}{2\pi} \int_{-\infty}^{\infty} e^{i\xi x} \left(\frac{\partial}{\partial \xi}\right)^{m+1} \left[\xi^m e^{-k_p(\Delta x)^{2p-1}\xi^{2p}t}\right] d\xi \\
&= \frac{i}{2\pi} \int_{-\infty}^{\infty} \left[\exp\left(\frac{i\xi x}{(t(\Delta x)^{2p-1}k_p)^{1/2p}}\right)\right] \left[\left(\frac{\partial}{\partial \xi}\right)^{m+1} \left[\xi^m e^{-\xi^{2p}}\right]\right] d\xi
\end{aligned}$$

The result is now clear. Also, an inspection of the right hand side of the above shows that $C_{m,p}$ can be chosen to be arbitrarily small if $t(\Delta x)^{2p-1}$ is large enough.

Discrete estimates analogous to (2.8) uniform in powers of A, are needed so that the compression method described below is valid.

Remark R1. Let the general space dependent coefficient, one dimensional system of hyperbolic equations

$$u_t + A(x)u_x = C(x)u,$$

where u is an ℓ vector, A is a uniformly diagonalizable smooth $\ell \times \ell$ matrix, with all real eigenvalues $\lambda_i(x)$, and $C(x)$ is smooth, be approximated by a dissipative finite difference scheme of order $2p-1$. Typically, its model equation is a systems version of (2.1)

$$u_t + A(x)u_x = C(x)u + (-1)^{p+1}(\Delta x)^{2p-1}P(x, \frac{\partial}{\partial x})u$$

where $(-1)^{p+1}P(x, \frac{\partial}{\partial x})$ is a $2p$ order elliptic operator. A more involved argument shows that the fundamental solution satisfies an estimate of the type (2.8) with the expression $x+at$ replaced appropriately by solutions of $\frac{d\tilde{x}}{dt} = \lambda_i(\tilde{x}) \quad \tilde{x}(0) = x, \quad i = 1, \ldots, \ell$ and with $C_{m,p}$ possibly growing in time like $C_{m,p}e^{kt}$ for k fixed.

Our numerical procedure involves the compression of the matrix A^ν, which for the purpose of analysis only, we shall view as the discretization of the fundamental solution for either (2.5) or (2.7),

$$(A^n)_{jk} = G(x_j, y_k, t^n)$$

where the interval $[0,1]$ is discretized via

$$x_j = \frac{j}{N}, \quad j = 1, \ldots, N, \quad N = 2^\nu,$$

$[0,1] \times [0,1]$ is discretized via (x_j, y_k), and $t^n = n\Delta t = n\lambda\Delta x, \quad n = 0, 1, \ldots$.

We now adapt the terminology, notation, and results of [1] to this unsteady problem (1.1).

Finite difference schemes approximating (1.1), e.g. (2.4) are regarded as acting on a vector $\{s_k^0\}_{k=1}^N$ which is to be viewed as approximating $u(x,0)$ on the finest scale:

$$s_k^0 = 2^{\frac{\nu}{2}} \int \varphi(2^\nu x - k + 1)u(x,0)dx$$

All functions, both continuous and discrete, are extended periodically:

$$u(x,t) \equiv u(x+1,t)$$
$$s_{k+N}^0 \equiv s_k^0$$

etc.

The function φ satisfies

$$\varphi(x) = \sum_{p=0}^{2m-1} h_{p+1}\varphi(2x-p)$$

The function $\psi(x)$ which will generate an orthonormal basis is obtained via

$$\psi(x) = \sum_{p=0}^{2m-1} g_{p+1}\varphi(2x-p)$$

with $g_p = (-1)^{p-1}h_{2m-p+1},\ p = 1,\dots,2m$ and $\int \varphi(x)dx = 1$.

The coefficients $\{h_p\}_{p=1}^{2m}$ are generally chosen so that

$$\psi_{j,k}(x) = 2^{-\frac{j}{2}}\psi(2^{-j}x - k + 1),$$

for j,k integers, form an orthonormal basis and in addition, the function $\psi(x)$ has m vanishing moments

$$\int \psi(x)x^{\ell}dx = 0, \quad \ell = 0,1,\dots,m-1.$$

Also we define

$$\varphi_{jk} = 2^{-\frac{j}{2}}\varphi(2^{-j}x - k + 1).$$

Finally, we assume as in [1] that there exists a real constant $\tau_m(\tau_1 = \frac{1}{2})$ such that the following conditions are satisfied:

$$\int \varphi(x+\tau_m)x^{\ell}dx = 0 \quad \text{for} \quad \ell = 1,\dots,m-1,$$

and $\int \varphi(x)dx = 1$.

In this case the quadrature formula becomes:

$$s_k^0 = \frac{1}{\sqrt{N}}(f(\frac{k-1+\tau_m}{N}) + O(N^{-(m+1)}))$$

and the initial discretization error is $O(N^{-(m+1)})$ up to uniform translation.

The decomposition of the vector $\{s_1^0, \ldots, s_{2^\nu}^0\}$ into the basis we use to compute with comes via

$$\begin{array}{ccccccc} \{s_k^0\} & \longrightarrow & \{s_k^1\} & \longrightarrow & \{s_k^2\} & \cdots \longrightarrow & \{s_k^\nu\} \\ & \searrow & \{d_k^1\} & \searrow & \{d_k^2\} & \cdots \searrow & \{d_k^\nu\} \end{array}$$

This is implemented in $O(N)$ operations using:

$$s_k^j = \sum_{p=1}^{p=2m} h_p s_{p+2k-1}^{j-1}$$

$$d_k^j = \sum_{p=1}^{p=2m} g_p s_{p+2k-1}^{j-1}$$

and the s_k^j, d_k^j are viewed as periodic sequences with period $2^{\nu-j}$.

The coordinates in the orthonormal basis consist of

$$[d_1^1, \ldots, d_{\frac{N}{2}}^1,\ d_1^2, \ldots d_{\frac{N}{4}}^2, \ldots, d_1^n,\ \ s_1^n].$$

The inverse mapping can also be done in $O(N)$ operations.

Each of the s_k^j is thought of as approximating

$$s_k^j = 2^{(\frac{\nu-1}{2})} \int f(x)\varphi(2^{\nu-j}x - k + 1)dx =$$
$$2^{-(\frac{\nu-j}{2})}\Big[f(2^{-\nu+j}(k-1+\tau_m))$$
$$+ O(N^{(-\nu+j)(m+1)})\Big]$$

while each d_k^j is thought of as approximating

$$d_k^j = 2^{(\frac{\nu-j}{2})} \int f(x)\psi(2^{\nu-j}x - k + 1)dx.$$

The numerical procedure effectively transforms the approximate discretization of the matrix $G(x_j, y_k, t^n)$ which is $(A^n)_{jk}$. Estimate (2.8) (corresponding to (4.5) and (4.6) of [1], uniform in all parameters, indicates (via an argument of [1]) that truncating A^n by

removing elements of a band of width $b \geq 2m$ around a shifted diagonal (and its periodic extension) i.e., those for which

$$|j - k - a\lambda n| \geq b > 2m,$$

which replaces A^n by $A^{n,b}$, leads to an estimate

$$\|A^n - A^{n,b}\| \leq \frac{C}{b^m} \log(N)$$

for C depending only on G.

Figure 2 shows what the nonzero elements in $A^{n,b}$ look like in the transformed basis for the variable coefficient case when $a = a(x)$ in (2.1).

Our examples are, of course, academic. The gain will come for highly oscillatory problems for which a large number of grid points are needed.

It also follows easily that for large N and fixed precision ε, only $O(N \log N)$ elements will be greater than ε. Alternatively, by discarding all elements that are smaller than a fixed threshhold we compress it to $O(N \log N)$ elements. Again following the discussion in [1], we note that this naive approach is to construct the full matrix in the wavelet basis and then to threshhold. Clearly this is an $O(N^2)$ operation.

Since we have, *à priori*, the structure of the singularities of the matrix A^ν the relevant coefficients can be evaluated by using the quadrature formulas. Estimate (2.8) guarantees that this procedure requires $O(N \log N)$ operations.

Remark R2. It is interesting to note that certain so called unstable difference schemes can be used without any drastic loss of efficiency. If (2.1) is approximated by,

$$\begin{aligned} u_j^{n+1} &= u_j^n - \lambda(u_{j+1}^n - u_{j-1}^n)/2, \\ u_j^0 &= u_0(x_j), \quad j = 1, 2, \ldots, N \end{aligned} \tag{2.9}$$

the algorithm is not stable for any fixed $\lambda > 0$, see e.g. [7].

The approximation does converge if $\Delta t \leq C\Delta x^2$, $(\lambda \leq C\Delta x)$ with an amplification factor $1 + \mathcal{O}(\Delta t)$. The number of timesteps for $t = \mathcal{O}(1)$ calculation will be large, $n = \mathcal{O}(\Delta x^{-2}) = \mathcal{O}(N^2)$. This is devastating for the standard explicit algorithm (1.2) but will only affect the complexity of (1.4) by a constant factor. The number of iterations (m in (1.4)) will increase from $\log(N)$ to $\log(N^2)$.

Our approach is in general not as favorable for multidimensional hyperbolic systems,

$$\begin{aligned} &\partial_t u + \sum_{j=1}^{d} A_j(x)\partial_{x_j} u = f(x), \quad x \in \mathbf{R}^d, \\ &u(x,0) = u_0(x). \end{aligned} \tag{2.10}$$

When u is a scalar or if the system can be diagonalized the algorithm (1.4) works well. The solution is given by integration along characteristics and the support of the Green's function is a small number of points (see Remark (R1) above). In the idealized case each row of A^ν consists of a fixed number of δ-functions. Its wavelet representation will have $\log(N^d)$ nonzero terms. The overall complexity for (1.4) is then $\mathcal{O}((\log N)^3 N^d)$ when the knowledge of the location of the δ-functions is used. This is better than the standard $\mathcal{O}(N^{d+1})$ estimate.

In general, however, the Green's function for (2.6) has a support with positive volume in $\mathbf{R}^d$ and with a singular support of positive measure in Hausdorff dimension $d-1$. The representation of the singular support consists of $\mathcal{O}(N^{d-1})\delta$-functions in each row of A^ν. This corresponds to $\mathcal{O}(\log(N)N^{d-1})$ wavelets and the overall algorithm contains at least $(\mathcal{O}(\log N)^2 N^{2d-1})$ wavelets.

For general multidimensional problems and for very large time the new algorithm is still of interest in special cases, e.g., if the solution is needed for a large number of different data u_0, f.

3. Parabolic Problems. The Green's function for parabolic problems is smooth in contrast to the hyperbolic case. The pure initial value problem for the heat equation,

$$\begin{aligned} &\partial_t u = \Delta u, \quad t > 0,\ x \in R^d, \\ &u(x,0) = u_0(x), \end{aligned} \tag{3.1}$$

has a solution of the form,

$$u(x,t) = (4\pi t)^{-d/2} \int_{R^d} \exp(-|x-y|^2/4t) u_0(y) dy. \tag{3.2}$$

In bounded domains the kernel has to be changed slightly depending on the boundary conditions. For positive $t(= n\Delta t)$ each row in A^n is always an approximation of segments of regular functions.

Our new technique is in general more favorable for parabolic problems than hyperbolic ones. The structure of the matrix B in (1.4) is simpler. When t increases the kernel becomes smoother and α_{jk} can be truncated to zero for all k when j is large enough.

Explicit methods for (3.1) also requires more operations than for hyperbolic problems when the standard method is used. This follows from the parabolic stability requirement,

$$\Delta t \leq \text{const. } |\Delta x|^2. \tag{3.3}$$

The new technique is only marginally affected by the constraint (3.3). Compare here the discussion above for unstable hyperbolic methods.

In more general higher order multidimensional parabolic cases the fundamental solution of, e.g.,

$$u_t + (-\Delta)^d u = 0$$

is

$$G_d(x,t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} d\xi \exp(i\xi \cdot x - |\xi|^{2d}t).$$

This is merely a multidimensional and rescaled version of the fundamental solution used in (2.8), and a simpler, but multidimensional version of (2.8) is just:

$$\left||x|^{m+1} D_x^m G_d(x,t)\right| \leq C_{md}.$$

Moreover C_{md} is arbitrarily small if t is large enough (this of course requires the nonexistence or other special behavior of lower order terms).

The matrix compression technique is easy here (for periodic problems without boundary conditions) because the significant terms of $[A^\nu]$ lie near the main diagonal and its periodic extension in one dimension. In two space dimensions (as is usual for elliptic operators), we also need to consider diagonals $i = j \pm kN$ for $0 < k \leq d$. Recall A is an $N^2 \times N^2$ matrix in 2D.

It is clear that à priori thresholding (to obtain $O(\epsilon)$ precision) near the image of these diagonals will give us an $O(N^d(\log N)^3)$ operation for each evaluation of the solution, where d is the number of space dimensions for the problem.

4. Numerical Experiments. The algorithm (1.4) was applied to hyperbolic problems in one space dimensions and to one and two dimensional parabolic problems. Various difference approximations and wavelet spaces were used. We present results concerning the accuracy of the calculations and the sparsity of $(SAS^{-1})^n$.

4.1 Hyperbolic problems. Consider the following scalar hyperbolic problem:

$$\begin{aligned} \partial_t u + a(x)\partial_x u &= f(x) \\ u(x,0) &= u_0(x) \end{aligned} \tag{4.1a}$$

with periodic boundary conditions ($0 \le x \le 1$). We made the following choices:

$$a(x) = 0.5 + 0.115\sin(4\pi x) \tag{4.1b}$$

$$f(x) = \cos(4\pi x) \tag{4.1c}$$

$$u_0(x) = \sin(4\pi x) \tag{4.1d}$$

In the discretization, $\Delta x = 1/1024$ and $\Delta t/\Delta x = 1$. The wavelet transform operator S uses the Daubechies-8 wavelets, which have 8 coefficients and have 4 vanishing moments. Finite difference schemes of order 1,2,3,4, and 5 of accuracy are tested.

These finite difference schemes are obtained as follows. In each interval

$$I_{\nu-\frac{1}{2}} = \{x/(\nu-1)\Delta x \le x \le \nu\Delta x\} \tag{4.2}$$

a polynomial of degree k is constructed. This polynomial interpolates the two points $(x_{\nu-1}, u^n_{\nu-1})$ and (x_ν, u^n_ν) and $k-1$ of its neighbors. If k is even these interpolation points go from $x_{\nu-\frac{k}{2}}$ to $x_{\nu+\frac{k}{2}}$. If k is odd they go from $x_{\nu-(\frac{k-1}{2})-1}$ to $x_{\nu+(\frac{k-1}{2})}$. This gives us a reconstruction function which is a polynomial of degree k in each $I_{\nu-\frac{1}{2}}$ and is continuous, but generally not differentiable at the boundary points $x_{\nu-1}$ and x_ν. We call this function $R^{n,k}(x)$

To approximate (4.1) at the grid points (x_ν, t^{n+1}) we solve (4.1) "exactly" with initial data

$$u_{\Delta x}(x, t^n) = R^{n,k}(x) \tag{4.3}$$

for $t^n \le t \le t^{n+1}$, evaluate the solution at (x_ν, t^{n+1}), and set $u^{n+1}_\nu = u_{\Delta x}(x_\nu, t^{n+1})$. We require $\frac{\Delta t}{\Delta x}\|a\|_\infty < 1$, so the solution depends only on data in $I_{\nu-\frac{1}{2}}$ if $a(x) > 0$ and $I_{\nu+\frac{1}{2}}$ if $a(x) < 0$.

In the special case when $a(x) = a$, constant, then

$$\begin{aligned} u^{n+1}_\nu &= R^{n,k}(x_\nu - a\Delta t) \\ &\quad + \int_{t^n}^{t^{n+1}} f(x_\nu - a(t^{n+1} - s))ds \end{aligned} \tag{4.4}$$

In the case when $f \equiv 0$ we get some familiar schemes: For $k = 1$ this is just the first order accurate upwind difference scheme (2.4). For $k = 2$ this is just the classical Lax-Wendroff second order accurate three point scheme, see e.g. [7]. For $k = 3, 4, 5$ the schemes are less studied, but are known to be L^2 stable, see e.g. [9] and the references therein.

For variable coefficients the result is

$$u_{\Delta x}(x_\nu, t^{n+1}) = R^{n,k}(x_\nu(t^n)) + \int_{t^n}^{t^{n+1}} f(x_\nu(t^{n+1} - s))ds \tag{4.5a}$$

where $x_\nu(t)$ solves

$$\frac{dx_\nu}{dt} = a(x_\nu), \quad t^n \le t \le t^{n+1} \tag{4.5b}$$

$$x_\nu(t^{n+1}) = x_\nu. \tag{4.5c}$$

A fourth order Runge-Kutta method is used to integrate the O.D.E. (4.5b,c) and Simpson's rule is used to evaluate the integral in (4.5a). The result of this approximation to the right side of (4.5a) is defined to be u_ν^{n+1}.

Returning to the present case the computations ran 13 steps until $t = 8$, that is, $(SAS^{-1})^{2^{13}}$ was computed.

At each step n the number of elements of A^n and $(SAS^{-1})^n$ whose absolute values are greater than 10^{-4} is shown in table 1. This is for methods whose order of accuracies go from one through five. The compression ratio decreases with the order of accuracy of the scheme. The results are also plotted on Figure 1.

These significant elements are located near the sub-diagonal corresponding to the characteristic curve which is known a priori. The image of these locations in $(SAS^{-1})^n$, shown on figure 2, has total length of $O(N \log N)$ elements where $N = 1024$.

In the computation of $(SAS^{-1})^n$, first, from the knowledge of the PDE, we figure out the structure of the singularities of A and its image in $(SAS^{-1})^n$. Then we compute $(SAS^{-1})^{2n} = (SAS^{-1})^n * (SAS^{-1})^n$ considering only the elements in a neighborhood of the singularities of A. In particular, we define the neighborhood of a singularity to be locations whose distance from the singularity are less than or equal to 5. If the singularities lie on a subdiagonal and its periodic extension we take neighborhoods forming a subband of bandwidths 11, 13, 15, and 17 (the wavelet filters have 8 elements). This bandwidth is independent of the time t (the step n) and the size of the problem. The errors due to

the subband truncation, measured by $\|u^n - \tilde{u}^n\|/\|u^n\|$, are shown in table 2b. Table 2a shows the relative error between the subband truncation and the exact solution. Here and throughout, "$\|\cdot\|$" denotes the ℓ^2 norm. Table 2c shows the relative error between the subband truncation and untruncated under grid refinement for the various orders. Unsurprisingly, since the relative length of the subband which is preserved decreases linearly with grid size, the error increases, but only slightly under this process.

We note that the compression (as seen in Figure 1 and Table 1) is better for odd order than for even order schemes. This is perhaps not surprising since (2.7) models schemes of odd order accuracy. Singularities behave a bit differently for even order (say order $= 2p$) schemes. These are modeled by

$$\begin{aligned} u_t + au_x = \ell_p(\Delta x)^{2p} \left(\frac{\partial}{\partial x}\right)^{2p+1} u \\ + (-1)^p k_p(\Delta x)^{2p+1} \left(\frac{\partial}{\partial x}\right)^{2p+2} u \end{aligned} \tag{4.6}$$

where $k_p > 0$ and ℓ_p are nonzero constants. The odd order dispersive term above may tend to spread singularities of the fundamental solution spuriously.

Finally table 3 shows the relative error due to truncation when the band width of the subband is 9, 11, 13, 15 and 17 for the methods of first through fifth order. Figures 3a and 3b compare the truncated versus the approximate solutions due to truncation of bandwidth 9 for the first and second order methods (the truncated graphs are dotted). As described in the previous paragraph, the 1st order method has a smoother truncation error and is hence more compressible by the wavelet representation.

4.2 Unstable schemes. For theoretical interest, we apply the method to a finite difference scheme which is unstable for $\frac{\Delta t}{\Delta x} = \lambda > 0$

$$u_j^{n+1} = u_j^n - \lambda(u_{j+1}^n - u_{j-1}^n)/2, \tag{4.7a}$$

$$u_j^0 = u_0(x_j). \tag{4.7b}$$

The amplification factor of this scheme is

$$1 - \lambda i \ \sin\theta = r(e^{i\theta}), \quad -\pi < \theta \leq 1 \tag{4.8}$$

so

$$|r(e^{i\theta})| = (1 + \lambda^2 \sin^2\theta)^{\frac{1}{2}}.$$

This means that if

$$\Delta t \leq 2c(\Delta x)^2 \tag{4.9}$$

for some $c > 0$, then

$$\|A^n\|_{l^2} \leq e^{cn\Delta t} \tag{4.10}$$

The restriction (4.9) means that the operation count for this explicit method would be $O(N^3)$ if we were silly enough to use it. However our compression method allows for an operation count of $O(N(\log N)^3)$ for the reasons described above.

Table 4 shows the number of elements in A^n and $(SAS^{-1})^n$ whose absolute values are greater than 10^{-3}. We choose a bigger threshold here since we took $\frac{\Delta t}{(\Delta x)^2} = 1$ and $n\Delta t = 2$, so $\|A^n\|$, as estimated in (4.10) grows to be roughly e when we are finished computing.

The error as measured by $\frac{\|u^n - \tilde{u}^n\|}{\|u^n\|}$ (subband truncation using bandwidth 11) was 0.0136.

We also performed convergence studies as we refined the grid for this method. Figures (4a,b,c) compare the numerical (untruncated) using dots versus exact solution for $m = 128, 256, 512$ grid points. The result indicates a second order method, as it should, since $\Delta t = (\Delta x)^2$. Figures (5a,b,c) compare the truncated bandwidth (using dots) vs the untruncated for this method for $m \neq 128, 256$, and 512 grid points.

The relative error decreases with mesh refinement. The truncation error equation associated with this scheme involves limited antidiffusion. Perhaps this accounts for this behavior.

4.3 System of hyperbolic equations. We apply the method to solving the system of hyperbolic equations:

$$\partial_t \begin{bmatrix} v \\ w \end{bmatrix} + \begin{bmatrix} a & 0 \\ 0 & -a \end{bmatrix} \partial_x \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.11a}$$

on $0 \leq x \leq 1, t \geq 0$ with the boundary conditions and initial conditions:

$$\begin{aligned} v(0,t) &= w(0,t) \\ w(1,t) &= v(1,t) \\ v(x,0) &= v_0(x) = \sin(4\pi x) \\ w(x,0) &= w_0(x) = \cos(2\pi x) \end{aligned} \tag{4.11b}$$

the coefficient a is chosen to be constant:

$$a = 0.115.$$

The numerical method used is the first order accurate upwind method described above. The results are similar to the scalar case, except the structure of the singularities in the matrices is more complicated. We have to keep track of reflections of singularities at the boundaries which is quite simple in this case. The number of elements in A^n and $(SAS^{-1})^n$ whose absolute values are greater than 10^{-4} is shown on table 5, and is plotted on figure 6. The relative error due to the subband of width 11 truncation, measured by $\|u^n - \tilde{u}^n\|/\|u^n\|$, is 0.0149.

The structure of the elements whose absolute values are greater than 10^{-4} of A^{2048} and $(SAS^{-1})^{2048}$ is shown in figures (7a,c), while Figure (7b) shows the image of a subband of bandwidth 11 in $(SAS^{-1})^{2048}$.

4.4 Parabolic problems. We do experiments on the following parabolic problem:

$$\begin{aligned} \partial_t u &= \partial_x(a(x)\partial_x u) + f(x) \\ u(x,0) &= u_0(x) \end{aligned} \tag{4.12}$$

with periodic boundary conditions ($0 \le x \le 1$). We made the following choices:

$$\begin{aligned} a(x) &= 0.5 + 0.25\sin(2\pi x) \\ f(x) &= -\pi^2\cos(2\pi x)^2 + \pi^2(0.5 + 0.25\sin(2\pi x))\sin(2\pi x) \\ u_0(x) &= \sin(4\pi x) \end{aligned}$$

The discrete setting and the wavelets are the same as in the hyperbolic problem. We use the simple explicit central difference scheme (4.13)

$$\begin{aligned} u_j^{n+1} = u_j^n + \frac{\Delta t}{(\Delta x)^2}\Delta_-(a(x_j)\Delta_+ u_j) \\ + \Delta t f(x_j) \end{aligned} \tag{4.13}$$

where

$$\Delta_{\mp} u_j = \mp(u_{j\mp 1} - u_j)$$

with $\Delta t/(\Delta x)^2 = 0.25$. The number of significant elements in A^n and $(SAS^{-1})^n$ is shown on table 6, and is plotted on figure 8.

For the parabolic problem, the large elements of A are in the neighborhood of the main diagonal. Their wavelet transform image is shown in figure 9. The relative error due to subband truncation was 0.0025.

4.5 Two-dimensional parabolic problems. We consider the following problem:

$$\partial_t u = a_{11}\partial_{xx}u + 2a_{12}\partial_{xy}u + a_{22}\partial_{yy}u$$
$$u(x,y,0) = u_0(x,y)$$

with periodic boundary conditions ($0 \le x \le 1,\ 0 \le y \le 1$). We choose

$$a_{11}(x,y) = 0.5 + 0.25\sin(2\pi x)$$
$$a_{12}(x,y) = 0.115\sin(2\pi x)\cos(2\pi y)$$
$$a_{22}(x,y) = 0.5 + 0.25\cos(2\pi y)$$
$$u_0(x,y) = \sin(4\pi x) + \cos(8\pi x).$$

We use a standard two-dimensional explicit central difference scheme. The two-dimensional data $u_{j,k},\ j = 1 \ldots N_1,\ k = 1 \ldots N_2$ forms a one-dimensional vector in the following way

$$\{u_{1,1} \ldots u_{1,N_2}, u_{2,1} \ldots u_{2,N_2}, \ldots, u_{N_1,1} \ldots u_{N_1,N_2}\}.$$

To reduce the size of the problem, N_2 is much less than N_1. In particular we took $N_1 = 128,\ N_2 = 8$ that is, $\Delta x = \frac{1}{128}\ \Delta y = \frac{1}{8}$.

The compression worked quite well. Table 7 shows the number of elements in A^n on $(SAS^{-1})^n$ whose absolute values are greater than 10^{-4}. The relative error due to subband truncation was 0.0066.

Acknowledgement

The authors would like to thank V. Rokhlin for suggesting both this problem and this approach to it, An Jiang for her kind help with some numerical calculations, and an anonymous referee for some constructive comments.

Bibliography

[1] G. Beylkin, R. Coifman and V. Rokhlin, "Fast wavelet transforms and numerical algorithms I", Comm. Pure Appl. Math., **44** (1991), pp. 141-184.

[2] I. Daubechies, "Orthonormal basis of compactly supported wavelets", Comm. Pure, Appl. Math. **41** (1988), pp. 909-996.

[3] L. Greengard and J. Strain, "A fast algorithm for the evaluation of heat potentials", Comm. Pure. Appl. Math. **43** (1990), pp. 949-964.

[4] L. Greengard and J. Strain, "The fast Gauss transform", SIAM J. Sci. Stat. Comput, Vol 12 **1** (1991), pp. 79-94.

[5] G. Hedstrom, "The rate of convergence of some difference schemes", SIAM J. Numer. Analys **5** (1968), pp. 363-406.

[6] Y. Maday, V. Perrier, and J.C. Ravel, "Adaptativité dynamique sur based'onde letts pour l'approximation d'équations aux dérivées partielles", to appear C.R. Acad. Sci. Paris.

[7] R.D. Richtmyer and K.W. Morton, "Difference methods for initial-value problems", Interscience Publishers (1967)

[8] V. Rokhlin, "Rapid solution of integral equations of classical potential theory", J. Comp. Phys. **60** (1985), pp. 187-207.

[9] A. Iserles and G. Strang, "The optimal accuracy of difference schemes", Trans. Amer. Math. Soc. **277** (1983), p. 779.

time steps	order 1 A^n	order 1 $(SAS^{-1})^n$	order 2 A^n	order 2 $(SAS^{-1})^n$	order 3 A^n	order 3 $(SAS^{-1})^n$	order 4 A^n	order 4 $(SAS^{-1})^n$	order 5 A^n	order 5 $(SAS^{-1})^n$
1	2048	48564	3072	48174	4096	49538	5120	49280	6144	49988
2	3072	51172	5120	52916	7168	53394	9216	54260	11264	54442
4	5120	50258	9216	54404	11264	53618	13588	53628	14736	52840
8	9216	48830	13996	52734	15502	53014	18596	54018	18732	54038
16	14886	45326	20822	53192	18480	52766	23462	55992	21372	54778
32	21376	40110	27860	54700	21788	51988	28370	57582	24796	55582
64	29728	31538	37032	56650	25656	51808	34702	60710	28188	56050
128	41190	22160	49420	58668	30462	51294	42602	63630	32192	56458
256	56652	15268	66054	58828	36074	47750	52600	67522	36344	56298
512	78586	10320	89838	55850	43358	41950	66560	71408	41820	55614
1024	113954	6836	130366	52256	54574	35810	89112	73374	50456	53532
2048	155624	4640	173238	47966	63564	30254	107704	75466	56242	50956
4096	211340	3126	229458	44002	74226	25544	130084	75416	62384	47858
8192	284318	2104	304854	40970	84896	21204	155688	74288	68536	43734

Table 1: Hyperbolic equation: the number of elements in A^n and $(SAS^{-1})^n$ whose absolute values are greater than 10^{-4}

B = 11

	order 1	order 2	order 3	order 4	order 5	
error	.1621	.0125	.0080	.0129	.0121	(a)
	order 1	order 2	order 3	order 4	order 5	
error	.0044	.0122	.0077	.0127	.0119	(b)
m	order 1	order 2	order 3	order 4	order 5	(c)
1024	.0044	.0122	.0077	.0127	.0119	
512	.0030	.0081	.0062	.0079	.0073	
256	.0017	.0050	.0046	.0065	.0054	
128	.0006	.0040	.0021	.0032	.0031	

B = 13

	order 1	order 2	order 3	order 4	order 5	
error	.1602	.0079	.0056	.0125	.0082	(a)
	order 1	order 2	order 3	order 4	order 5	
error	.0025	.0075	.0055	.0124	.0081	(b)
m	order 1	order 2	order 3	order 4	order 5	(c)
1024	.0025	.0075	.0055	.0124	.0081	
512	.0019	.0052	.0040	.0057	.0058	
256	.0013	.0032	.0031	.0047	.0039	
128	.0005	.0024	.0013	.0018	.0019	

B = 15

	order 1	order 2	order 3	order 4	order 5	
error	.1604	.0068	.0045	.0090	.0090	(a)
	order 1	order 2	order 3	order 4	order 5	
error	.0019	.0064	.0042	.0089	.0089	(b)
m	order 1	order 2	order 3	order 4	order 5	(c)
1024	.0019	.0064	.0042	.0089	.0089	
512	.0011	.0041	.0030	.0041	.0042	
256	.0006	.0025	.0018	.0032	.0026	
128	.0003	.0020	.0007	.0013	.0012	

B = 17

	order 1	order 2	order 3	order 4	order 5	
error	.1600	.0034	.0024	.0072	.0063	(a)
	order 1	order 2	order 3	order 4	order 5	
error	.0009	.0028	.0022	.0071	.0062	(b)
m	order 1	order 2	order 3	order 4	order 5	(c)
1024	.0009	.0028	.0022	.0071	.0062	
512	.0004	.0017	.0012	.0020	.0020	
256	.0002	.0009	.0005	.0010	.0008	
128	.0000	.0005	.0000	.0002	.0000	

Table 2: Hyperbolic equation: (a) the error measured by $\frac{\|u^n - \tilde{u}^n\|}{\|u^n\|}$ composed with the exact solution; (b) due to the truncation only; (c) due to the truncation only under grid refinement.

	order 1	order 2	order 3	order 4	order 5
B=9	.0248	.0331	.0330	.0378	.0374
B=11	.0044	.0122	.0077	.0127	.0119
B=13	.0025	.0075	.0055	.0124	.0081
B=15	.0019	.0064	.0042	.0089	.0089
B=17	.0009	.0028	.0022	.0071	.0062

Table 3: (a) Errors measured by $\frac{\|u^n - \tilde{u}\|}{\|u^n\|}$ due to truncation for various bandwidths and first through fifth orders.

	order 1	order 2	order 3	order 4	order 5
B=9	.1771	.0333	.0330	.0379	.0374
B=11	.1621	.0125	.0080	.0129	.0121
B=13	.1602	.0079	.0056	.0125	.0082
B=15	.1604	.0068	.0045	.0090	.0090
B=17	.1600	.0034	.0024	.0072	.0063

Table 3: (b) Errors measured by $\frac{\|u_{exact} - \tilde{u}\|}{\|u_{exact}\|}$ due to truncation for various bandwidths and first through fifth orders.

n	A^n	$(SAS^{-1})^n$
1	512	512
2	750	512
4	1024	1336
8	1024	1764
16	1024	2328
32	1024	3060
64	1024	4026
128	2048	5273
256	2048	6302
512	2560	7447
1024	3432	8360
2048	4566	9305
4096	6330	9266
8192	9362	10557
16384	14332	13346
32768	23872	19255
65536	41490	29649
131072	74750	46595
262144	132916	64558
524288	132454	106197
1048576	132304	110240
2097152	130164	115276

Table 4: Hyperbolic equation "unstable scheme": the number of elements in A^n and $(SAS^{-1})^n$ whose absolute values are greater than 10^{-3}

n	A^n	$(SAS^{-1})^n$
1	2048	19351
2	3074	22569
4	5126	25327
8	6154	26440
16	9228	25804
32	13332	25747
64	19488	22364
128	27692	18985
256	37948	14064
512	52308	10116
1024	72814	8110
2048	98456	5685

Table 5: System of hyperbolic equations: The number of elements in A^n and $(SAS^{-1})^n$ whose absolute values are greater than 10^{-4}.

n	A^n	$(SAS^{-1})^n$
1	3072	15194
2	5120	17342
4	8462	19136
8	11682	19328
16	16214	18775
32	21900	17622
64	30126	14389
128	41434	10387
256	56756	7392
512	78078	5073
1024	106976	3554
2048	146466	2396
4096	199878	1658
8192	272050	1082

Table 6: Parabolic equation: the number of elements in A^n and $(SAS^{-1})^n$ whose absolute values are greater than 10^{-4}

n	A^n	$(SAS^{-1})^n$
1	6632	34190
2	16612	52941
4	40210	72420
8	72360	87381
16	105802	84827
32	146292	67912
64	198480	46856
128	269882	31925
256	365456	21497
512	491936	13653
1024	658800	8703
2048	891144	5271
4096	1048576	3373
8192	1048576	1981

Table 7: 2D-parabolic equation: the number of elements in A^n and $(SAS^{-1})^n$ whose absolute values are greater than 10^{-4}

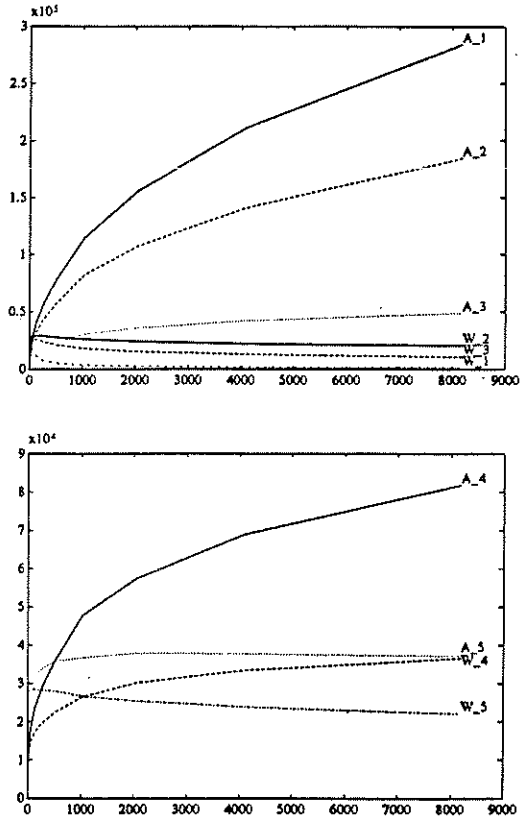

Figure 1: Hyperbolic equation: the number of elements in A^n and $(SAS^{-1})^n = w^n$ whose absolute values are greater than 10^{-4}

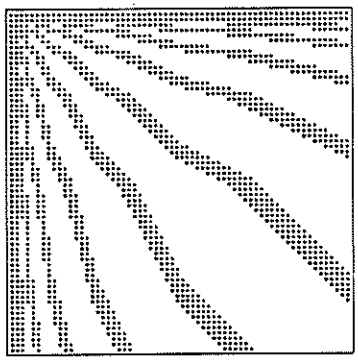

Figure 2: Hyperbolic equation: the pattern of significant elements in $(SAS^{-1})^n$.

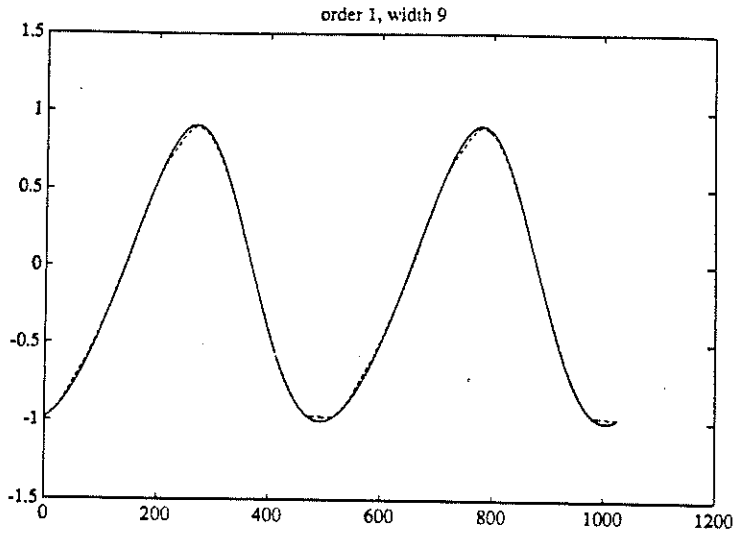

Figure 3a: Truncation versus nontruncated approximate solution, first order method truncated at bandwidth 9. (Truncated is dotted).

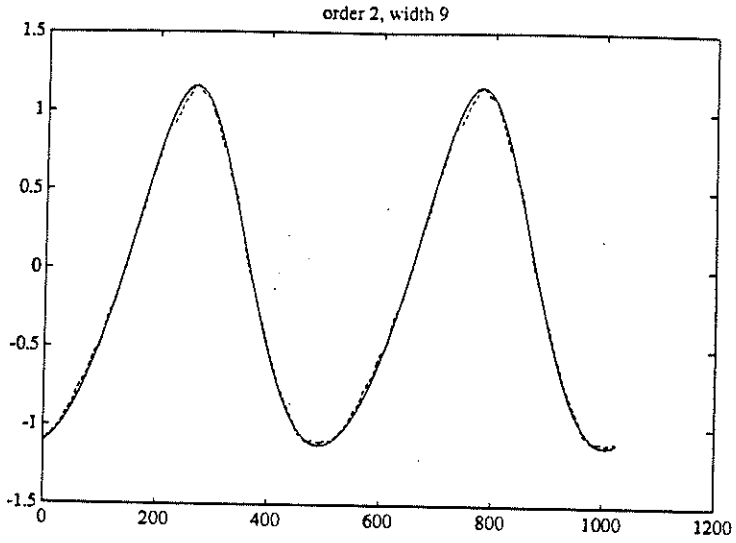

Figure 3b: Truncated versus nontruncated approximate solution, second order method, truncated at bandwidth 9. (Truncated is dotted).

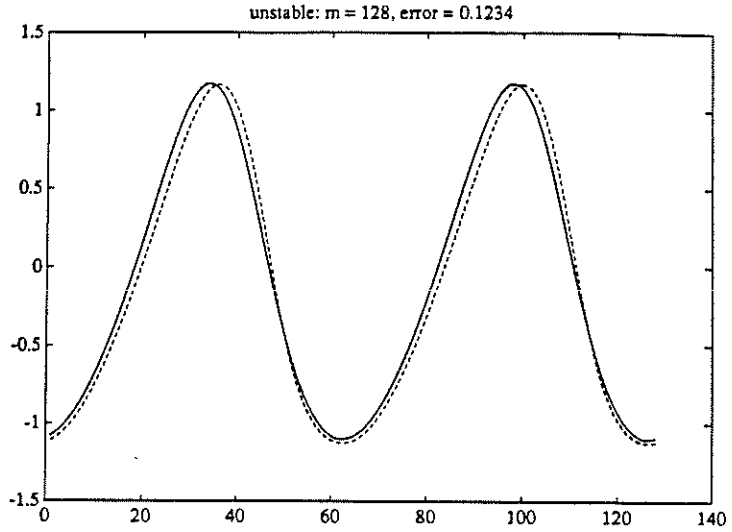

Figure 4a: Exact vs approximate solution, "unstable scheme", $m = 128$

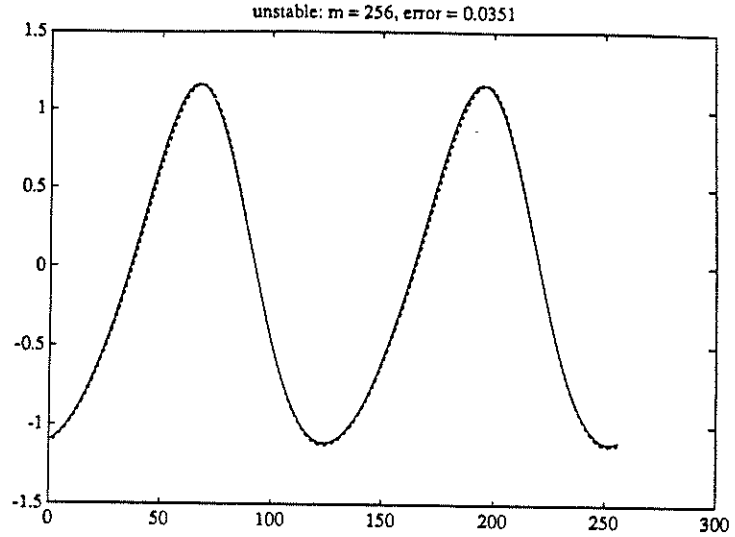

Figure 4b: Exact vs approximate solution "unstable scheme", $m = 256$

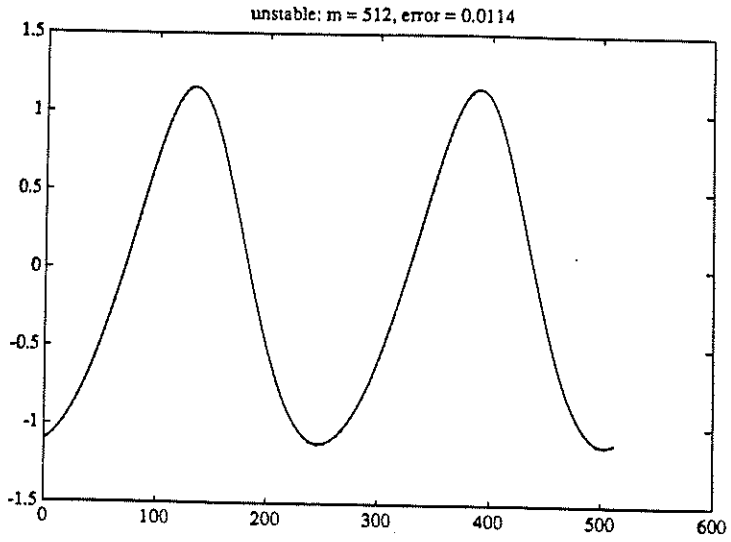

Figure 4c: Exact vs approximate solution "unstable scheme", $m = 512$

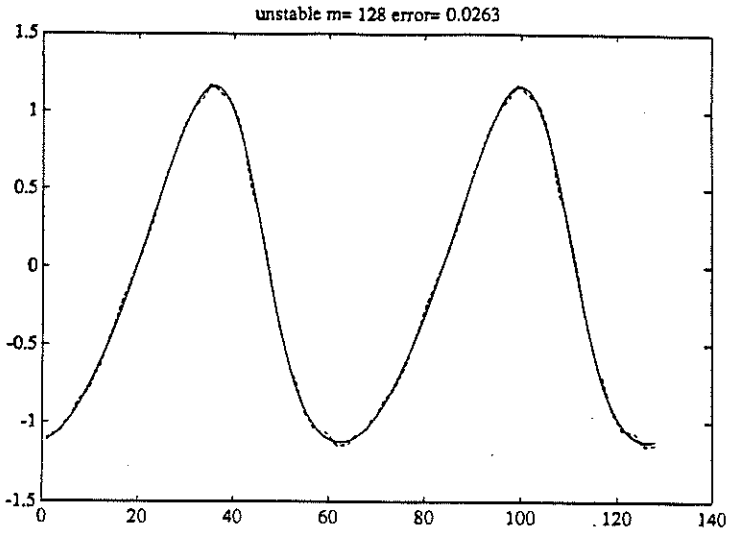

Figure 5a: Truncated bandwidth 11 vs untruncated for the "unstable scheme", $m = 128$

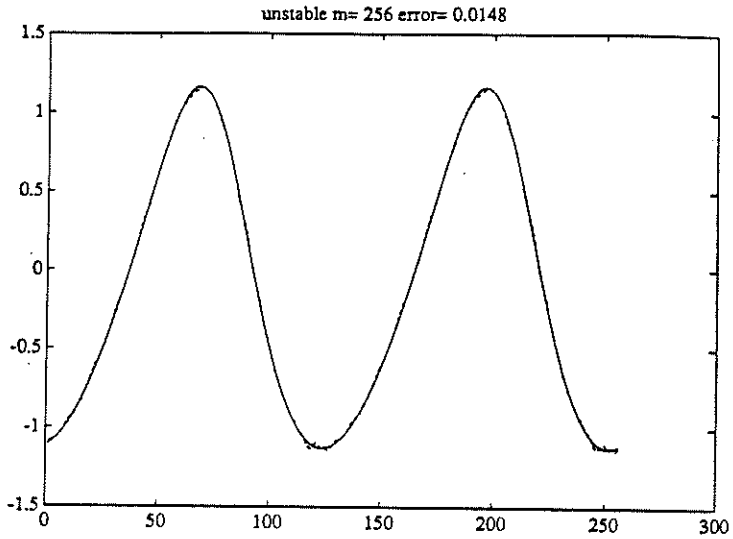

Figure 5b: Truncated bandwidth 11 vs untruncated for the "unstable scheme", $m = 256$

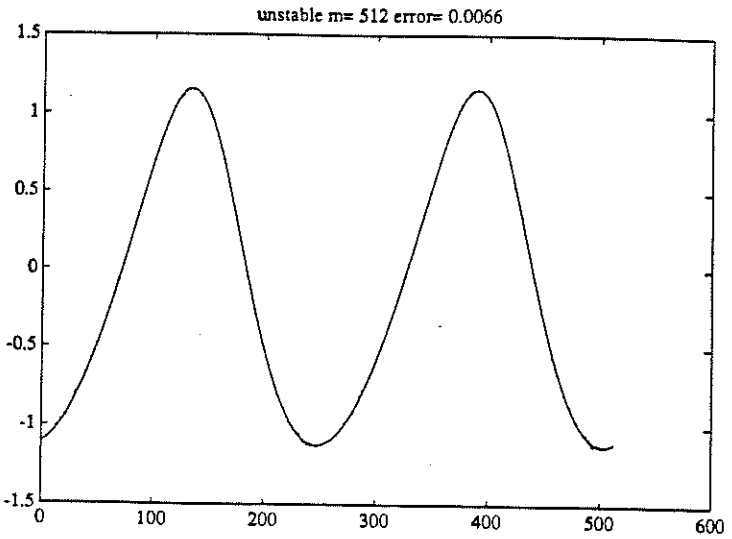

Figure 5c: Truncated bandwidth 11 vs untruncated for the "unstable scheme", $m = 512$

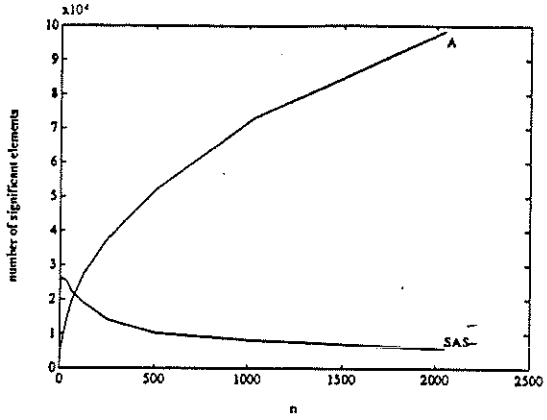

Figure 6: System of hyperbolic equations: The number of elements in A^n and $(SAS^{-1})^n$ whose absolute values are greater than 10^{-4}.

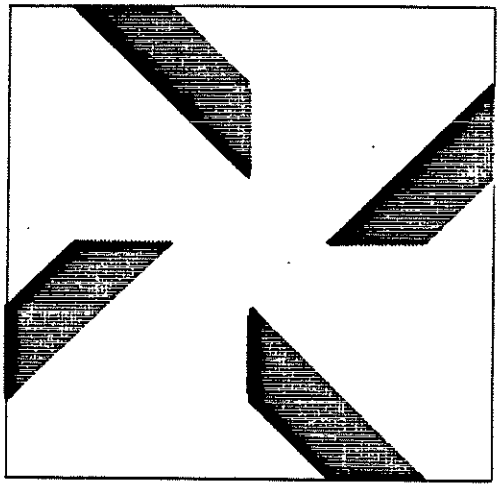

Figure 7a: System of hyperbolic equations pattern of significant elements $(7, 10^{-4})$ for A^n, $n = 2048$

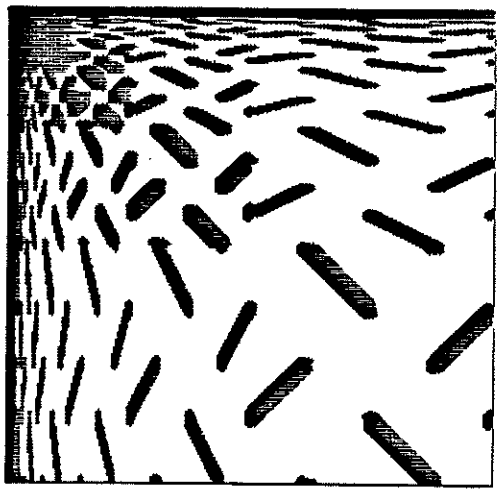

Figure 7b: System of hyperbolic equations pattern of significant element for $(SAS^{-1})^n$, $n = 2048$. image of bandwidth 11 around singular support

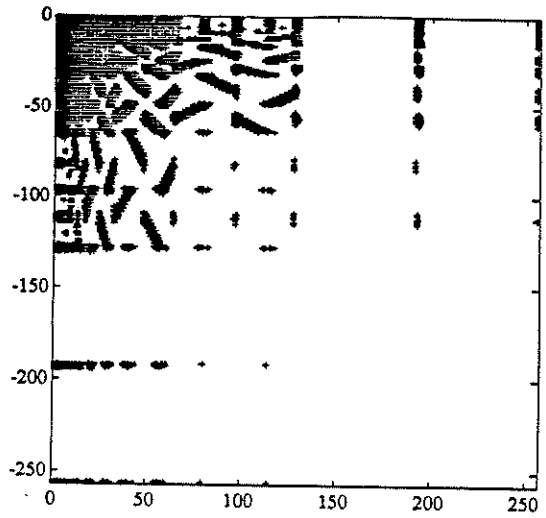

Figure 7c: System of hyperbolic equations pattern of significant elements ($\geq 10^{-4}$) for (SAS^{-1}), $n = 2048$

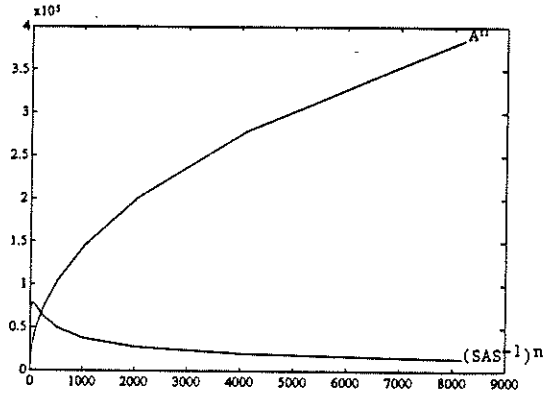

Figure 8: Parabolic equation: the number of elements in A^n and $(SAS^{-1})^n$ whose absolute values are greater than 10^{-4}

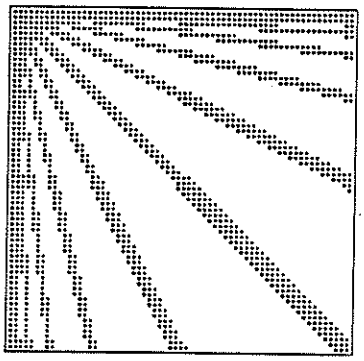

Figure 9: Parabolic equation: the pattern of significant elements in $(SAS^{-1})^n$.