PDE and Wavelets: Problems and Issues

Daniel D. Beatty

September 3, 2003

Contents

In most of the literature, wavelets are used to condition the difference equation matrix, and yield a sparser matrix. This is useful for increasing the likely hood that the matrix can solved. Sparse matrices, like those generated by the wavelet transform, are less complex and easier to solve in the case of implicit forms. The reason is that sparse and well conditioned matrices allow fast algorithms to be applied. The goal of this chapter of this thesis is to explore these sparse matrix producing and conditioning techniques that wavelets offer in comparison to traditional methods.

1 PDE in General

Computational Scientists consider partial differential equations (pde) important their ability to solve problems as they relate equilibrium, diffusion states, oscillitory systems. Most of the conventional algorithms are costly for performing these tasks. Therefore the challenge of how to compute these faster has computational scientist concerned. This why the use of wavelets are considered.

1.1 Classic Methods

In the classic sense computational scientists agree that there are three basic classification of partial differential equations: elliptical, parabolic, and hyperbolic. Some consider the hyperbolic to be made of a fourth category, ultra-hyerpbolic. Regardless, these kind of problems are grouped together for the similarities in properties and boundary conditions. Also, three of these types have their own usefulness as they model physical phenomena.

For example, elliptical PDEs have examples in heat expansion, electrostatic charge, and other source expansion equations. Parabolic PDEs model diffusion of gases or fluids. Oscillating states such as electromagnetic fields, some control theory, vibrating strings, and electronic communication systems all involve hyperbolic PDEs to model their behavior.

The difference in partial differential equations is defined from their general form:

$Au_{xx} + Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu = G$

For case where $B^2 - 4AC > 0$, then the PDE is hyperbolic, and the solution has an oscillatory nature to it. In cases, where $B^2 - 4AC < 0$, then the PDE is elliptical. If there is an equivalence to zero, then the PDE is parabolic. The effect this has on classic methods (analytical and numerical) is how the PDE is represented and eventually solved.

In addition to the above classifications, their limits or boundaries also have a classification. Boundary conditions are defined either on the dependent variable or the gradient of the dependent variable. If the boundary conditions leaves the region open, then the problem is an initial value problem.

Each solution (analytic or numerical) is tailored to the equation; however, the three main categories will have similar solutions due to similarities in the problem. In numerical terms there are two general mechanisms for solving these equations: explicit and implicit. Explicit tends favor initial

value problems, and has difficulties with round off errors. Implicit handles error quite well; however, complexity exists increases depending on the number points to be solved for. A third method also exists, Monte Carlo. However, the Monte Carlo method is more for "simulating the results of differential or integral equations.

1.1.1 Elliptical Differential Equations

Elliptical PDEs can be represented with Poison's formula. In the homogenous case, and Laplace's equation is the representing equation.

$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$

Also, it is typical that a polygon is chosen to define the boundary conditions. Typically, the polygon is defined by either constants either in the first derivative or normal space. Simplest is a rectangle specified in the first derivative for the right and top edges and constants for the lower and left edges. Also, the central difference formula is used to setup a linear set equations to be solved. [1]. The trick is to set up the matrix, and then solve it by any means (classic or wavelet based).

Boundary conditions for elliptical PDEs consist of three catagories: Dirichlet, Neumann, and Robbin's. Dirichlet Boundary conditions are specified by either functions or constants on the solution function itself. Neumann Boundary conditions specify against the derivative of the function based on some either independent or dependent variable. The combination of the two is called Robbin's Boundary Conditions. What does mean?

Boundary conditions define the starting point and limits for solving PDEs. This is necessary to have enough constants in the solution for a solution to be calculated. Different boundary condition types have different method of solution to account for the boundary conditions.

1.1.2 Parabolic Differential Equations

Parabolic type of differential equations usefully describe diffusion and fluid mechanics. Some analytical methods useful for solving parabolic equations include substitution of variables, Laplace and Fourier Transforms. Examples of Parabolic equations include heat diffusion, diffusion - convection, and Navier-Stokes.

Boundary conditions for parabolic PDEs are less formal than elliptical PDEs. Boundary conditions are specified either across the boundary, on the boundary, or a hybrid of the two. The boundary conditions are typically spelled by example in temperature equations. However; it is conceivable that other diffusion problems have similar issues.

1.1.3 Hyperbolic Differential Equations

Hyperbolic type of differential equations are prevalent in equations for oscillating phenomena. This includes electromagnetic fields and vibrating strings. In its analytical form, the D'Alembert Solution is an textbook example method. The D'Alembert Solution is an example the canonical form and it use in solving PDE problems.

In some literature canonical form is used to generate a sparse domain. The general rule for using canonical form comes from the D'Alembert Solution which is as follows:

1. Replace (x,t) by new canonical coordinates.
2. Solve the transformed equations.
3. Transform the solution into the original coordinates
4. Substitute the general solution into the IC's the acquire the constants.

One set of boundary conditions to watch out for are ones that can vary either their strength or position. Examples:

- $u_n + \lambda u = g(t)$
- $u_n = g(t)$
- $u = g(t)$

Some the boundary conditions that Farlow [7] refers to are controlled end points, force specified boundaries, and elastic attachments. In the case of force specified points, the boundary point itself can move in position. In the elastic attachment problem, the force can change on each of the boundary points. While, controlled end points are straight forward functions or constants. In each case, the force, position, and function of the boundary conditions must be accounted for through out the solution of the PDE.

1.1.4 Explicit Methods/ Iterative Methods

Explicit methods are used to calculate one result after the next of partial differential equations. One thing that both implicit and explicit methods have in common is to transform the PDE in difference equations. At the heart of both methods is the central, backward, and forward difference approximation formulae. The central difference method is applied to part of a grid with accuracy of $O(x^2)$ and $O(t^2)$. To illustrate the explicit method, a parabolic PDE used for reference. That PDE is as follows:

- $\phi_{i,j+1} = \phi_{i,j+1} + \frac{2\alpha^2 \Delta t}{\Delta x^2}(\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j})$

- $\phi_{i,j+1} = p\phi_{i+1,j} + q\phi_{i,j} + r\phi i - 1, j$ such that $p, q, r > 0$ and $p + q + r \leq 1$

The basic method is as follows:

1. Start with the index value of 0 (i=0 at the initial value)
2. Find the solution for all of x for $t = t_{i+1}$ by explicit formula.
3. Establish the boundary conditions with respect for $u_{t_{i+1},x}$ by boundary condition approximation formula.
4. Repeat steps 2 through 4 until i = n (where n is the maximum index).

In this case the explicit method is used to solve a PDE where parts of it have been transformed into its central difference equivalent. The boundary conditions must also be computed in cases where they are not constants. In cases where the boundary conditions are specified on a derivative, those conditions must be placed in their difference equation form. Note that for each iteration, the boundary condition must be computed.

This computation is useful for cases where speed is necessary. This methods speed comes from the next step being computed from the current and previous steps. However, this explicit form is unstable, and tends to yield results inconsistent with the boundaries. In order for this form to be stable :

Limits on the explicit Method and alternatives by implicit methods are as follows:

1. Imposed limits on Δx and Δt. Dependences in explicit methods are directly limited to 3 values of the many values which it theoretically should.
2. The implicit method is approximately the second derivative $\frac{\partial^2\theta}{\partial x^2}|_{i,j}$ "by the finite difference formula involving θ at an advanced time (t_{j+1})" [1]. A mid-point is computed using a central - difference formula
   $\frac{\partial\theta}{\partial t}|_{i,j+1/2} = \frac{\theta_{i,j+1}\theta_{i,j}}{\Delta t}$
3. The 2nd p.d. applied with central-difference formula. There is catch with a weighting parameters.
4. Variable Weighted Implicit Formula can be used with the following conditions:
   - weighting factor θ
   - more than one unknown variable at the time step $j + 1$

1.1.5 Implicit Methods

One area where explicit and implicit methods differ is the arrangement of the linear equations used to solve the system. Typically, explicit methods utilize the previous solution to determine

the boundary conditions of the current solution. While in a sense both linear equations, implicit methods lend themselves to simultaneous solution more than explicit ones.

An example of an implicit method is the Crank-Nicolson Method. The idea is to solve by use of a system of equations arrived at by converting the PDE into a system of difference equations.

Method:

1. Pick some value for λ such that $\lambda \in [0, 1]$
2. Pick Δx and Δt and assign grid points
3. Use computational molecule to generate equation.
4. Solve the matrix

1.2 Problems

The previous subsection discussed PDEs in general and how to solve them. Listed in this subsection are some common PDE problems. The selection has at least one problem each of the three PDE categories. In this section, the classic method of solution is applied. However, the wavelet methods are saved for the next section. Implicit solutions are generally chosen where accuracy is required, and explicit methods are shown when speed and complexity are necessary.

In order to make these solutions a few formulae need to be defined. These formulae are the central difference, forward difference, and backward difference formulae.

Central Difference Formulae:

1. $f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}$
2. $f''(x) \approx \frac{f(x+h)-2f(x)+f(x-h)}{h^2}$

Backward Difference Formula:

- $f'(x) \approx \frac{f(x)-f(x-h)}{h}$

Forward Difference Formula

- $f'(x) \approx \frac{f(x+h)-f(x)}{h}$

1.2.1 Semi-Infinite String Problem

The semi-infinite string problem is a typical resonance problem. Newton's physical laws derive the equation based on external forces, friction forces, restoration forces, and net forces. In the simplest form, the problem is solved for net forces only. Even this problem requires a matrix to solve it. The problem is defined mathematically as:

- PDE $u_{tt} = c^2 u_{xx}$ $\forall x \in (0, \infty)$ and $\forall t \in (0, \infty)$
- BC $u(0, t) = 0$
- IC $u(x, 0) = f(x)$
- $u_t(x, 0) = g(x)$
- general solution: $u(x, t) = \frac{1}{2}[f(x - ct) + f(x + ct)] + \frac{1}{2c}\int^{x+ct} x - ctg(\zeta)d\zeta$
- $c^2 u_{xx}$ is the net force due to the tension on the string.
- u_{tt} represents the longitudinal or torsional vibrations on the string.

There is a conventional solution that comes from the central difference, and forward difference formulae. The rest is rather simple algebra. One key issue is the boundary conditions. These must be solved to establish the constants in the matrix. Once these are established, the solution can be arrived at by conventional methods. The conventional algebra is as follows:

- $u_{xx} = u[i, j + 1] - u[i, j] + u[i, j - 1]$
- $u_{tt} = u[i + 1, j] - u[i, j] + u[i - 1, j]$
- $u_t = (u[i + 1, j] - u[i, j])$
- $u[t, 0] = 0$
- $u[0, x] = f(x)$
- $u_t[0, x] = g(x) = u[1, x] - u[0, x]$
- $u[1, x] = g(x) + f(x)$
- $u[i + 1, j] + (c^2 - 1)u[i, j] + u[i - 1, j] - c^2 u[i, j + 1] - c^2 u[i, j - 1] = 0$

1.2.2 Heat Diffusion

The heat diffusion or heat conduction equation defines heat in a solid at any point and any time within the domain. Diffusion comes from a heat source, and may come from an artifical constant. In the case of heat diffusion, there is a constant α which is defined as the diffusitivity constant. Diffusitivity is defined by the following:

$\alpha = \frac{K}{\tau\sigma}$

such that K is the thermal constant, τ is the density and σ is the specific heat. The heat diffusion problem is defined as follows:

- $u_t = \alpha^2 u_{xx}$ such that
- u_t is the change in temperature with respect to time.
- u_{xx} is the concavity of the temperature

1.2.3 Diffusion - Convection

The big idea for diffusion - convection regards convection currents effects on the diffusion process. In particular, this address substances diffusing in areas where currents exists. One case is a gaseous substance being released in an air conditioned room or wind area. Another case is the release of a liquid substance into a river.

- $u_t = \alpha^2 u_{xx} - v u_x$ such that
- u_t is the change in time
- $\alpha^2 u_{xx}$ is the diffusion component
- $v u_x$ is the convection component
- v is the velocity of the convection current
- α is the diffusitivity constant

The solution to this problem is best done by considering the general solution, and then filling in the constants with the boundary conditions. The general solution is as follows:

- $u_t = \alpha^2 u_{xx} - v u_x$
- $u_t = u[i+1, j] - u[i, j]$
- $u_x = u[i, j+1] - u[i, j]$
- $u_{xx} = u[i, j+1] - 2u[i, j] + u[i, j-1]$
- $u_t - \alpha^2 u_{xx} - v u_x = 0$
- $u[i+1, j] - u[i, j] - \alpha^2 u[i, j+1] + \alpha^2 2u[i, j] - \alpha^2 u[i, j-1] - vu[i, j+1] + vu[i, j] = 0$
- $u[i+1, j]\alpha^2 u[i, j+1] - \alpha^2 u[i, j-1] - vu[i, j+1] + (2\alpha^2 + v - 1)u[i, j] = 0$

2 Related Work

2.1 Gregory Beylkin

2.1.1 BVP

Role of orthonomal basis in computation:

Example of Calderon-Zygmund and pseudo-differential operators.

"Fast algorithms for applying these operators to functions, solving integral equations. The operators which can be efficently treated using representations in the wavelet bases include Calderon-Zygmund and pseudo-differential operators."

Again G. Beylkin used his Kernel operation to achieve a wavelet transform:

$T(f)(x) = \int K(x,y)f(y)dy$

The above has the effect of a convolution operator, which implies the wavelet transform via convolution.

One of the issues for differential equations (ordinary or partial) is the condition number. Usually, the step size for differential equation is chosen to be small for accuracy. However, this produces a system of equations which is dense, the condition number is not likely to be high. The condition number has direct effect as to the rate of convergence for the solutions of these equations.

"If our starting point is a differential equation with boundary conditions then the wavelet system of coordinates there is a diagonal preconditioned which allows us to perform algebraic manipulations only with the sparse the sparse matrices."

Beylkin's Method was applied to multi-grid methods with a sample Green's function. The idea is to use the wavelet bases as a preconditioner in the multi-grid scheme. "Orthonormal wavelet bases provide a very convenient tool for implementing the preconditions.

"Matrix L has the condition number $O(N^2)$. "

"The goal is to construct the matrix L^{-1} numerically in $o(-\log \epsilon N)$ operations, where ϵ is the desired accuracy."

"The kernel of the inverse operator for the problem has a sparse representation in wavelet bases since kernel satisfies the estimates of the type in .... ".

"The wavelets play an auxiliary role in that they provide a system of coordinates where the condition numbers of the sparse matrices (involved in the computations) under control. "

2.1.2 AMS

Theme: For a number of operators we may compute the non-standard form in the wavelet bases by solving a small system of linear equations.

Example: Non-standard form of the operator:

Compute $\alpha_{i,l}^j$, $\beta_{i,l}^j$ and $\gamma_{i,l}^j$ of A_j, B_j, and Γ_j where $i, l \in Z$ for the operator $\frac{d}{dx}$.

- $\alpha_{i,l}^j = 2^{-j} \int \psi(2^{-j} - i)\psi\prime(2^{-j}x - l)2^- jdx = 2^{-j}\alpha_{i-l}$
- $\beta_{i,l}^j = 2^{-j} \int \psi(2^{-j} - i)\phi\prime(2^{-j}x - l)2^- jdx = 2^{-j}\beta_{i-l}$
- $\gamma_{i,l}^j = 2^{-j} \int \phi(2^{-j} - i)\psi\prime(2^{-j}x - l)2^- jdx = 2^{-j}\gamma_{i-l}$

where

- $\alpha_l = \int_{-\infty}^{\infty} \psi(x - l)\frac{d}{dx}\psi(x)dx$
- $\beta_l = \int_{-\infty}^{\infty} \psi(x - l)\frac{d}{dx}\phi(x)dx$
- $\gamma_l = \int_{-\infty}^{\infty} \phi(x - l)\frac{d}{dx}\psi(x)dx$

For example using

- $\phi(x) = \sqrt{2}\sum_{k=0}^{k-1} h_k\phi(2x - k)$
- $\psi(x) = \sqrt{2}\sum_{k=0}^{k-1} g_k\phi(2x - k)$
- $\alpha_i = 2\sum_k \sum_{k\prime} g_k g_{k\prime} r2i + k - k\prime$
- $\beta_i = 2\sum_k \sum_{k\prime} g_k h_{k\prime} r2i + k - k\prime$
- $\gamma_i = 2\sum_k \sum_{k\prime} h_k g_{k\prime} r2i + k - k\prime$

$r_l = \int_{-infty}^{\infty} \phi(x - l)\frac{d}{dx}\phi(x)dx$

Conclusions, eye catchers, and questions

1. The representation of d/dx is completely determined by the coefficents r_l or more to the point by the representation on the subspace V_0.
2. The coefficents r_l depend only on the auto-correlation function of the scaling functions ϕ, rather than the scaling function itself.
3. Justification of one of the conclusions (2) is that integral depends on $|\phi(\xi)|^2$.
4. Also, how does Beylkin get to this point?

Proposition: The goal is to reduce the computation of the coefficients r_l to solving a system of linear algebraic equations.

Given:

$$r_l = \int_{-\infty}^{\infty} \phi(x-l)\frac{d}{dx}\phi(x)dx$$

$$r_l = \int_{-\infty}^{\infty} |\phi(\xi)|^2(i\xi)e^{-il\xi}d\xi$$

Proposition 1: If the given exists, then the following coefficents $r_l, l \in Z$ satisfy the following system of linear algebraic equations:

$$r_l = 2(r_{2l} + \frac{1}{2}\sum_{k=1}^{L/2}\alpha_{2k-1}(r_{2l-2k+1} + r_{2l+2k-1}))$$

$$\sum_l lr_l = -1$$

such that $a_{2k-1} = 2\sum_{i}^{L-2k} = 0h_ih_{i+2k-1}$ $k \in U[1, L/2]$. are the auto-correlation coefficients of the filter H.

If $M \geq 2$, then a and b have an unique solution with a finite of non-zero r_l, namely $r_l \neq 0 \forall l \in [-L+2, L-2]$ and $r_l = -r_l$

2.1.3 Fast Wavelet Transforms and Numerical Algorithms I

Introduction

Reasons for Wavelets in Numerical Computations The article makes the following claims within the first paragraph.

1. Rapid Application of dense matrices: The article claims that direct application of an $N \times N$ matrix to a vector is an $O(N^2)$ operation.

2. Integral Applications: These are also claimed as numerically expensive with hidden recursively generated vectors which large quickly. The complexity of these operations are claimed as $O(N^3)$ in complexity.

3. Finite Difference and Finite Element potential: These items are claimed to be devices for reducing a partial differential equation to a sparse linear system with an inherently high condition number.

4. Fast Fourier Transform Limitation: The FFT and most related algorithms suffer from excessive costs of their transform. Typically, these transforms have complexity $O(N \log(N)$. Also, these schemes suffer from a fragile exactness due to the properties of the operator.

5. Fast Approximate Algorithms exploiting analytic properties: Such operators are claimed to be applicable to arbitrary vectors. As a result, the article claims that these operators "do not require that the operators in question be translation invariant, and are considerably more adaptable than the algorithms base on the FFT and its variants."

6. Fast Numerical Applications: The wavelet method normally requires O(N) operations and is directly applicable to all Calderon-Zygmund, pseudo-differential operators, and general operators with a specified degree of accuracy (finite accuracy).

Methods of the Paper The paper used the Haar Wavelet Transform, and mechanisms first developed by Stromberg, Meyer, and Ingrid Daubechies. Operators included in the study are integral, Dirichlet and Neumann boundary value problems for elliptical partial differential equations, and Legendre series. The idea like most wavelet schemes is to use the wavelet transform to make the matrix operator very sparse. Once that is the case, the operation against a vector is $O(N)$ in complexity. The construction is claimed to be $O(N^2)$ with the exception of structures whose singularities are known a priori. In the case of the exception, the compression operator is an order O(N) procedure.

Mechanisms provided that the article claims to provide for evaluating integral operators. A standard and non-standard form. The non-standard scheme claims to extend the standard form leading to an $O(N)$ scheme.

The paper is organized with section concepts. The first concept is the Haar Wavelet Basis. Second are relevant facts regarding wavelets. Third is a description of the integral operators for which we obtain an order N algorithm. Included with the third is a billinear operator and its description. Fourth is the complexity analysis. Finally, the numerical applications of wavelets are presented.

Properties of Wavelets One property that the article hits on is the fact the Haar Wavelet does not drop off very fast. In the case of the paper, the Daubechies Wavelet Basis functions are used.

References

[1] Singiresu S. Rao *Applied Numerical Methods for Engineers and Scientists* published Prentice Hall Upper Saddle River, NJ 07458 copyright 2002

[2] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery *Numerical Methods in C* Published by the Press Syndicate of the University of Cambridge The Pitt Building, Trumpington Street, Cambridge CB2 1RP 40 West 20th Street, New York, NY 10011-4211, USA Copyright Cambridge University Press 1988, 1992

[3] G. Beylkin *On wavelet-based algorithms for solving differential equations*

[4] G. Beylkin *Wavelets and Fast Numerical Algorithms*

[5] G. Beylkin, R. Coifman, and V. Rokhlin *Fast Wavelet Transforms and Numerical Algorithms I, Article in Communications on pure and applied mathematics* copyright 1991 Wiley, New York

[6] G. Beylkin, D. Gines and L. Vozovoi *Adaptive Solution of Partial Differential Equations in Multiwavelet Bases*

[7] Stanly J. Farlow *Partial Differential Equations for Scientists and Engineers* copywrite 1993 Dover Publications, Inc Mineola, N.Y. 11501

[8] Murray R. Spiegel, *Theory and Problems of Advanced Mathematics for Engineers and Scientist* copy-write 1996 McGraw-Hill

[9] Stephane Jaffard, Yves Meyer, and Robert D. Ryan *Wavelets: Tools for Science and Technology* copyright 2001 by the Society for Industrial and Applied Mathematics Philadelphia, PA 19104-2688