

# PDE and Wavelets: Problems and Issues

Daniel D. Beatty

September 18, 2003

## Contents

<b>1</b>	<b>PDE in General</b>	<b>3</b>
1.1	Classic Methods . . . . .	3
1.1.1	Elliptical Differential Equations . . . . .	4
1.1.2	Parabolic Differential Equations . . . . .	4
1.1.3	Hyperbolic Differential Equations . . . . .	5
1.1.4	Explicit Methods/ Iterative Methods . . . . .	5
1.1.5	Implicit Methods . . . . .	6
1.1.6	Galerkin . . . . .	7
1.2	Problems . . . . .	8
1.2.1	Semi-Infinite String Problem . . . . .	8
1.2.2	Heat Diffusion . . . . .	9
1.2.3	Diffusion - Convection . . . . .	10
<b>2</b>	<b>Related Work</b>	<b>10</b>
2.1	Gregory Beylkin . . . . .	10

2.2	Turbulence and Navier Stokes . . . . .	11
2.3	KL Transform Approach . . . . .	12
2.4	Galerkin Approach . . . . .	13
2.5	. . . . .	13

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 likelihood that the matrix can be 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, oscillatory 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-hyperbolic. 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 Poisson’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 categories: Dirichlet, Neumann, and Robin’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 Robin’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:

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

- $\phi_{i,j+1} = p\phi_{i+1,j} + q\phi_{i,j} + r\phi_{i-1,j}$  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 method's speed comes from the next step being computed from the current and previous steps. However, the 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  $\Delta x$  and  $\Delta t$ . 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  $\theta$  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  $\theta$
  - 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  $\lambda$  such that  $\lambda \in [0, 1]$
2. Pick  $\Delta x$  and  $\Delta t$  and assign grid points
3. Use computational molecule to generate equation.
4. Solve the matrix

### 1.1.6 Galerkin

The Galerkin method supposes that a complete orthonormal system  $\{v_j\}_j$  is defined on  $L^2([0, 1])$  and every  $v_j$  is  $C^2$  on  $[0, 1]$ . The boundary conditions of the  $v_j$  is typically defined as well. The solution approximation is then defined on the span of this orthonormal system. For example,  $u_s = \sum_{k \in \Lambda} (x_k v_k) \in S$  such that  $S$  is a span of  $v_j : j \in \Lambda$ , and  $x_j$  is a scalar. The catch is that  $u_s$  should behave as true solution a system of linear equations, i.e. a vector itself. The linear equations are the implicit set of equations for solving a PDE or ODE.

Frazier takes this one step further to show a parallel from Galerkin to a conventional implicit form. First he shows:

- $\langle Lu_s, v_j \rangle = \langle f, v_j \rangle \forall j \in \Lambda$  such that  $\langle f, g \rangle = \int_0^1 f(t)g(\bar{t})dt$
- Furthermore:  $\langle L(\sum_{k \in \Lambda} x_k v_k), v_j \rangle = \langle f, v_j \rangle \forall j \in \Lambda$  leading to
- $\sum_{k \in \Lambda} \langle Lv_k, v_j \rangle x_k = \langle f, v_j \rangle \forall j \in \Lambda$

The final connection is that each element of a matrix  $A$  defined  $A = (a_{j,k})_{j,k \in \Lambda}$  is a scalar defined by  $\langle Lv_k, v_j \rangle$ . This yields the following equality

$$(a_{j,k})_{j,k \in \Lambda} = \langle Lv_k, v_j \rangle \text{ and } \sum_{k \in \Lambda} a_{j,k} x_k = y_j \forall j \in \Lambda$$

- $x$  is a vector  $(x_k)_{k \in \Lambda}$
- $y$  is a vector  $(y_k)_{k \in \Lambda}$
- $A$  is a matrix with rows and columns indexed by  $\Lambda$
- $a_{j,k}$  is an individual element of  $A$

With Galerkin, for all subsets in  $\Lambda$  we obtain an approximation  $u_s \in S \rightarrow u$ . This is done by solving  $Ax = y$  and using  $x$  to determine  $u_s$ . One of the tricks is finding the  $v_j$ 's and  $x_j$ 's such that the equations are satisfied.

## 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  $u_{tt} = c^2 u_{xx} \quad \forall x \in (0, \infty) \text{ and } \forall t \in (0, \infty)$
- BC  $u(0, t) = 0$
- IC  $u(x, 0) = f(x)$



- $u_t(x, 0) = g(x)$
- general solution:  $u(x, t) = \frac{1}{2}[f(x - ct) + f(x + ct)] + \frac{1}{2c} \int^{x+ct} x - ct g(\zeta) d\zeta$
- $c^2 u_{xx}$  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 artificial constant. In the case of heat diffusion, there is a constant  $\alpha$  which is defined as the diffusivity constant. Diffusivity is defined by the following:

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

such that  $K$  is the thermal constant,  $\tau$  is the density and  $\sigma$  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.

- $u_t = \alpha^2 u_{xx} - vu_x$  such that
- $u_t$  is the change in time
- $\alpha^2 u_{xx}$  is the diffusion component
- $vu_x$  is the convection component
- $v$  is the velocity of the convection current
- $\alpha$  is the diffusivity 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:

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

## 2 Related Work

### 2.1 Gregory Beylkin

One example applied mathematician in the field of wavelets is Gregory Beylkin of the University of Colorado. He has published several papers on the topic, and few have similar work to this paper. Some of his quotes deserve some explanation, and other do not. The main point is to use the wavelet transform to diagonalize the matrix for fast sparse methods to work.

One quote, “Fast algorithms for applying these operators to functions, solving integral equations. The operators which can be efficiently treated using representations in the wavelet bases include Calderon-Zygmund and pseudo-differential operators.” [3] This quote references two mathematical forms introduced in the early half of the 20th century. Calderon-Zygmund was difference operator that divided the vector in two each time it operated. The more generic version, the pseudo-difference operator was introduced shortly thereafter. Both introduced methods of numerically differentiating equations. Beylkin used kernel  $K$  to illustrate the use of the wavelet transform:

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

This kernel allows for the mathematical explanation of how wavelets fit in scheme of solving PDE's. The bottom line is best said in Beylkin's own words, “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 matrices. ... 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. ”

Indeed, in Beylkin's examples the wavelets diagonalize the matrices for which it works on. Beylkin's multi-resolution scheme is unique in the way it provides the diagonalization process. It performed successively on each average term. However, the averaging and differencing is applied to portions in the vertical and horizontal components. Thus some of these parts are also forced to zero.

Beylkin claims in another paper that wavelets support rapid application of dense matrix. In some cases, Beylkin claims to achieve operations in  $O(N^2)$  operations. In general it is clear that wavelets offer the ability to reduce the overall size of the input, and therefore reduce the time necessary to compute the result. It is important to note that the input is an equivalent form to the original, and the new form is simply more compact.

On both integral applications, it is known that the algorithms have hidden recursive components that tend to make the algorithm grow exponentially. In cases described by Farlow [7], integral equations are solved in a similar manner to PDE's by using their numerical equivalence forms.

Lastly, Beylkin points out that wavelets have several advantages over the Fast Fourier Transform. Amongst them is the cost of the transform itself. The wavelet transform in 1-D space has a cost of nearly  $O(N)$ . where the FFT is of order  $O(N^2)$  Another is translation invariance. The wavelet transform does not require translation invariance, where as the FFT does. This gives the wavelet transform some adaptability to numerical applications.

## 2.2 Turbulence and Navier Stokes

Jaffard, Meyer, and Ryan showed exploration into the basis functions useful for solving PDE problems such as Navier-Stokes and other turbulence problems. Amongst the first attempts for solving Navier-Stokes used the Battle-Federbush and Shannon basis. One of the keys for this problem was the absence of boundary conditions.

The idea was to apply the Battle-Fererbush basis with the Galerkin method to solve Navier Stokes. This concept was first published by Zenin in 1981. One of the difficulties is getting the matrix solution to diagonal form. The Shannon basis does not cause the non-diagonal elements to decay rapidly. In some cases, neither does Battle-Federbush.

The main issues for wavelets and PDE solutions in some opinions is to address complicated geometries. Others see that local refinements can enhance or be used in place of multi-grid algorithms in some cases. Again the key is to make the solution matrix converge to diagonal and sparse system.

## 2.3 KL Transform Approach

In the solution of the PDE in implicit form, the bottom line is to solve a linear equation. One approach is to use the Karhunen-Loeve transform. As stated by Wickerhauser [10] the overall goal this solution method is to acquire the singular values, eigenvalues, and eigenvectors. From these answers, the solution to the matrix is trivial. Methods for accomplishing this are factor analysis, principle component analysis, singular value decomposition, and the KL transform.

The point of wavelets with the KL-Transform was well stated "The approximate factor analysis algorithm is the search through a library of orthonormal basis for the one whose H is closest to that of the Karhunen-Loève basis, and cases of fast search methods the result is a fast approximate Karhunen-Loève algorithm. "[10]

So what problem is solved by the KL transform? "Two problems solved by principle orthogonal decomposition. First, distinguishing elements from a collection by making d measurements. Second, inverting a complicated map from a p-parameter configuration space to d-dimensional measurement space. "[10] What does this mean? Usual cases for the extraction of singular values come measurements taken. The parameters necessary to get these values by certain formulae are extracted by their singular values which the KL transform can acquire. In the case of PDE's, these parameters represents a mapping to eigenvalues, eigenvectors, and singular values. As stated, "the Karhunen-Loève basis eigenvectors are also called principle orthogonal components or principal factors, and computing them for a given ensemble X is also called factor analysis." [10]

Where do wavelets come in? "It is possible to build a library of more than  $2^d$  fast transforms U of  $R^d$  to use for "x" points.' Question: What does this mean? This means that wavelets can be used to construct the fast transforms U which make up the "x". From these basis, a particular wavelet basis can be selected as the best choice. Or so this idea is implying.

The method of using wavelets to acquire the KL transform are stated as follows:

- Expand N vectors  $\{X_n \in R^d : n = 1, 2, \dots, N\}$  into wavelet packets coefficients:  $O(Nd \log d)$
- Summing squares into the variance tree:  $O(d \log d)$
- Searching the variance tree for a best basis:  $O(d + d \log d)$
- Sorting the best basis vectors into decreasing order  $O(d \log d)$

- Diagonalizing the auto-covariance matrix of the top  $d'$  best basis vectors  $O(d'^3)$

Total complexity of the Approximate Karhunen-Loève basis:  $O(Nd \log d + d'^3)$ . Since  $d' \ll d$ , it safe to say the approximate solution is faster than the conventional one.

The approximate Karhunen-Loève transform of one vector

- Computing the wavelet packet coefficients of one vector  $O(d \log d)$
- Applying the  $d' \times d'$  matrix  $K'^*$  :  $O(d'^2)$

Updating the approximate Karhunen-Loève basis

- Expanding one vector into wavelet packet coefficients
- Adding the coefficients into the means tree
- Adding the squared coefficients into the squares tree
- Forming the variance tree and computing the new information costs
- Searching the variance tree for the joint best basis

The classification in large data sets apply to both rogues' gallery problem, fingerprint classification problem, and rank reduction for complex classifiers.

## 2.4 Galerkin Approach

## 2.5

## 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
- [10] Mladen Victor Wickerhauser *Adapted Wavelet Analysis from Theory to Software* copyright 2001 by the Society for Industrial and Applied Mathematics Philadelphia, PA 19104-2688