# Wavelets as a Galerkin basis

Yakovlev A.S.

*Department of Computational Physics,*
*St Petersburg State University*

## 1 Introduction

The Galerkin method is one of the best known methods for finding numerical solutions to partial differential equations. Its simplicity makes it perfect for many applications. The Galerkin approach consists of finding a functional basis for the solution space of the equation, then projecting the solution on the functional basis, and minimizing the "residual" with respect to the functional basis.

The translates of a wavelet for all dilations form an unconditional orthonormal bases of $L^2(\mathbf{R})$ and the translates of a scaling function for all dilations form an unconditional orthonormal bases for $V_j \subset L^2(\mathbf{R})$, which is a great improvement over the standard polynomial basis or a trigonometric basis for the Galerkin Method. Variable resolution levels inherent with wavelets and multiresolution analysis allow for localizing small time scale variations of the solution. Furthermore, the properties of multiresolution analysis allow for fast switching of functional bases.

### 1.1 Overview

The goal of this project is to formulate a Wavelet-Galerkin Method to solve a partial differential equation numerically. The motivation for attempting this problem is that Wavelet- Galerkin Methods provide an improvement over standard Galerkin Methods by using a compactly supported orthogonal functional basis. Furthermore there are two ways to improve the approximation using a wavelet basis namely by increasing the resolution level and the order of the wavelet.

## 1.2 The Galerkin Method

Consider the following Boundary Value Problem (BVP)

$$L[u(x,y)] = 0 \text{ on } D(x,y), S(u) = 0 \text{ on } \partial D \tag{1}$$

where $L$ is a differentiation operator, $D(x,y)$ is the domain of the BVP, $\partial D$ is the boundary of the domain, and $S(u)$ the boundary conditions (BC). Let us assume that $u(x,y)$ can be represented accurately by a set of analytic functions $\{g_i(x,y)\}_{i=1}^{N}$. Then we can approximate the solution of the BVP as

$$u(x,y) \approx u_a(x,y) = u_0(x,y) + \sum_{i=1}^{N} a_i g_i(x,y)$$

where $u_0$ is chosen so as to satisfy the initial conditions. In order for $u_a(x,y)$ to be a good approximation, it must satisfy the differential equation. Thus if we apply (1) to $u_a$ we get

$$L[\sum_i a_i g_i(x,y)] + L[u_0(x,y)] = R(a_1, ..., a_N, x, y).$$

Since $u_a$ is not an exact solution of the BVP, $R$ will never be exactly zero, however we can minimize it with respect to a set of "weight functions." Galerkin Methods are a subclass of the so called Weighted Residual Methods, however, we shall not discuss that, and just use the simplest variant of Galerkin Method which is equivalent in our case to the Ritz-Raleigh Method. The Ritz-Raleigh Method minimizes the residual with respect to the set of approximating functions by requiring

$$\langle R(a_1, ..., a_N, x, y), g_i(x,y)\rangle_{L^2} = 0, \ i = 1, 2, ..., N.$$

Hence since the summation is finite and differentiation is linear,

$$\langle R(a_1, ..., a_N, x, y), g_i(x,y)\rangle_{L^2} = \sum_{j=1}^{N} a_j \langle L[g_j], g_i\rangle + \langle L[u_a], g_i\rangle = 0. \tag{2}$$

In this case one must solve the following matrix equation in order to find the coefficients $\{a_i\}_{i=1}^{N}$,

$$\begin{pmatrix} \langle L[g_1], g_1\rangle & \dots & \langle L[g_N], g_1\rangle \\ \vdots & \ddots & \vdots \\ \langle L[g_1], g_N\rangle & \dots & \langle L[g_N], g_N\rangle \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} \langle L[u_a], g_1\rangle \\ \vdots \\ \langle L[u_a], g_N\rangle \end{pmatrix}.$$

## 1.3 Multiresolution Analysis

Multiresolution Analysis is a sequence of closed spaces $\{V_j\}_{j=-\infty}^{\infty}$ (they are called approximation spaces) of special kind

1. $V_j \subset V_{j+1}$
2. $\bigcup_{j\in Z} V_j = L^2(\mathbf{R})$
3. $\bigcap_{j\in Z} V_j = \{0\}$
4. if $f(t) \in V_j \Rightarrow f(2t) \in V_{j+1}$
5. if $f(t) \in V_j \Rightarrow f(t-k) \in V_j$
6. single scaling function $\varphi$ defines orthonormal basis in corresponding subspace $V_j$ by scaling and translations

$$\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k).$$

For example, the set of spaces

$$V_j = \{f \in L^2(\mathbf{R}); \forall k \in \mathbf{Z} : f|_{[2^j k, 2^j(k+1)]} = const\}$$

is called Haar's multiresolution analysis. Now if we call $P_j$ - orthogonal projector on $V_j$ we will get

$$\lim_{j\to-\infty} P_j f = f \text{ for every } f \in L^2(\mathbf{R})$$

A method to construct the basis of this kind one can find in book [1]

## 1.4 Multiresolution Analysis and the Galerkin Method

Having a Multiresolution Analysis, $V_n, n \in Z$ with scaling function $\varphi(x)$, one can use $\varphi_{n,k}(x)$ as the basis functions for the Galerkin Method. We know that the set $\{\varphi_{n,k}(t) = 2^{n/2}\varphi(2^n t - k)\forall n, k \in \mathbf{Z}\}$ forms an orthonormal basis of $V_n$, thus at each approximation level $n$, one can take the orthogonal projection of $u$ onto $V_n$ in the following manner

$$u(x,y) \approx \Gamma_n u(x,y) = \sum_k a_{n,k}(y)\varphi_{n,k}(x), \qquad a_{n,k}(y) = \langle u(x,y), \varphi_{n,k}(x)\rangle$$

and this will provide an accurate approximation to $u$. Furthermore, for some $J$, $V_J$ will "capture" all details of the original function. It is important to note that $k$ is finite, since it would not make much sense to attempt to solve an infinite system of equations. Typically we would want to choose $k \in \{0, ..., 2n-$

1} in order to get a good sampling of the domain. Also the $a_{n,k}$, as it is given by the above projection, can only be used as initial fit for $y = 0$, since we shall re-calculate them through the Galerkin Method for a better fit to the BVP. Now if we apply this multiresolution analysis to (1) and force the condition in (2) we achieve the following,

$$\langle L[\Gamma_n u(x,y)] + L[u_0(x,y)], \varphi_{n,k} \rangle = 0 \tag{3}$$

However this is clearly an $1D$ approximation which would be quite useless to solve a $2D$ problem. A trick that can be used to make the approximation in (3) a plausible solution to a problem such as (1) is to let the coefficients $a_{n,k}$ vary with $y$. Hence we can sample the $y$ domain and for each $y_i$ we can solve the system resulting from (3), since it would be reduced to a $1D$ problem for a fixed $y$.

Thus, we arrive at the following matrix equation:

$$\mathbf{BA} = \mathbf{R} \tag{4}$$

where $B_{i,j} = \langle L[\varphi_{n,i}], \varphi_{n,j} \rangle$, $A_i = a_{n,i}(y)$, $R = -\langle L[u_0], \varphi_{n,j} \rangle$.

Solving (4) for each $y$, we can find the coefficients $a_{n,k}(y)$ and thus a solution to (1). Furthermore, multiresolution analysis allows easy conversion between different levels $n$, through the scaling relation,

$$\varphi_{j-1,l} = 2^{-1/2} \sum_{k \in \mathbf{Z}} p_k \varphi_{j,k}. \tag{5}$$

Consequently by solving the Galerkin system for a certain $n$, with a multiresolution analysis's scaling function's dilates and translates as the functional basis, we have solved the Galerkin system for all $m < n$ as well. The solution to the system of matrix equations in (4) can be simplified even further by applying a quadrature rule for easy evaluation of the inner products in R. It can be for example

$$\langle f(x), \varphi_{n,k}(x) \rangle \approx m f(k/2^n), \; m = \int_{-\infty}^{\infty} \varphi(x) dx$$

Quadrature rules exist for the fast calculation of products between scaling functions and their derivatives, but they are not as elementary and require rather involved methods, gives a good exposition on the matter.

A few notes are in order before concluding this section. First, the above method is a very rough approximation to a problem like (1), since in order to generate a very easy to solve matrix out of it we fix $y$, and linearize the $a_{n,k}$, by doing so we do not take into account the action of the differential operator $L$ on them. Thus for a PDE such as $u_{yy} = 0$ the above described method is rendered useless. However it's speed and simplicity, make it at least feasible

approach. Second, the MRA-Galerkin Method as it is does not account for the BCs of the BVP. It is impossible to force BCs on the spaces $V_j$ , thus the BCs cannot be applied. However, as we shall see in the next section, it is possible to construct similar spaces to those in an MRA in order to automatically satisfy the BC.

# 2 Solving differential equation

As shown in the previous section, applying properties of multiresolution analysis to the Galerkin Method offers a few improvements over traditional trial functions. However, it restricts our solutions to elements of $V_n$. In solving partial differential equations arising from physical phenomena solutions will belong to the class of finite energy surfaces, or $L^2(\mathbf{R}^2)$. Thus it would be more beneficial to have a scheme that will solve the PDE in $L^2(\mathbf{R}^2)$ rather than in $V_n \subset L^2(\mathbf{R})$. This is where the wavelets role is, since they can form orthogonal bases of $L^2(\mathbf{R}^k)$.

Let us see how a hierarchy of wavelet solutions to partial differential equations may be developed using scaling function bases. In order to demonstrate the wavelet technique, we consider the one dimensional second order differential equation

$$u_{xx} + \alpha u = f$$

where $u = u(x)$, $f = f(x)$ and $\alpha = \text{const}$. This equation is the one dimensional counterpart of Helmholtz's equation. To compare results let us choose an easy to calculate method of finite differences.

## 2.1 Finite difference solution of the periodic problem

Consider the problem

$$u_{xx} + \alpha u = f \tag{6}$$

where $u$ and $f$ are periodic functions in $x$. Let $d$ be the period. Then

$$u(0) = u(d)$$

$$f(0) = f(d)$$

Suppose now that we have an $n$ point discretization of the interval $[0, d]$, so that

$$u_i = u(i\Delta x)$$

$$f_i = f(i\Delta x)$$

where

$$i = 0, 1, 2 \ ... \ n-1$$

and

$$\Delta x = d/n.$$

The finite difference approximation to $u_{xx}$ is then

$$(u_{xx})_i = \frac{1}{\Delta x}(\frac{u_{i+1} - u_i}{\Delta x} - \frac{u_i - u_{i-1}}{\Delta x}) = \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2}$$

and so the discrete form of equation (6) is

$$u_{i+1} + (-2 + \alpha\Delta x)u_i + u_{i-1} = f_i(\Delta x)^2; \quad i = 0, 1, 2, ..n-1$$

or

$$u_{i+1} + ru_i + u_{i-1} = f_i(\Delta x)^2; \quad i = 0, 1, 2, ..n-1$$

where

$$r = -2 + \alpha\Delta x.$$

Noting that $u_{-1} = u_{n-1}$ and $u_0 = u_n$ this system of equations can be written in matrix form as

$$\begin{pmatrix} r & 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & r & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & r & 1 & \dots & 0 & 0 \\ 0 & 0 & 1 & r & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & r & 1 \\ 1 & 0 & 0 & 0 & \dots & 1 & r \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_4 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-2} \\ f_{n-1} \end{pmatrix}$$

or

$$CU = F.$$

Now $C$ is a circulant matrix, so that premultiplication of the vector $U$ by the matrix $C$ is the same as convolving $U$ with the vector $K$, where $K$ is the first column of $C$. $K$ is referred to as the convolution kernel. Thus

$$K * U = F.$$

This form of the equations is easily solved using the Fast Fourier Transform (FFT), since a convolution in physical space of two vectors is equivalent to a component by component multiplication in frequency space of the discrete Fourier transforms of the vectors. Here, and in Section 2.2 (next section), we use the FFT to solve the equations instead of Gaussian elimination because it results in a significant reduction in computational storage and time. Solution using the FFT takes $O(n \log_2 n)$ operations in contrast to the $O(b^2 n)$ operations required for a banded Gaussian elimination solution. Taking discrete Fourier transforms of both sides, therefore, we have

$$\hat{K}.\hat{U} = \hat{F}$$

where the . denotes component by component multiplication. Then the solution is obtained by taking the inverse Fourier transform of

$$\hat{U} = \hat{F}/\hat{K}$$

where / denotes component by component division. In the particular case where $\alpha = 0$, the rows of the matrix $C$ sum to zero. As a consequence of the singularity of $C$, we have

$$(\hat{K})_0 = 0$$

i.e. the first element of the vector $\hat{K}$ is zero, which leads to an undefined value for $(\hat{U})_0$. To circumvent this problem, we initially consider solutions $U$ whose mean over the period is zero. This means that $(\hat{U})_0 = 0$, so that we can set $(\hat{K})_0$ to be an arbitrary nonzero value to avoid division by zero, and then set $(\hat{U})_0 = 0$. Where Dirichlet boundary conditions require $U$ to have a nonzero mean, the mean may be calculated from the boundary conditions.

## 2.2 Wavelet-Galerkin solution of the periodic problem

The wavelet-Galerkin method entails representing the solution $u$ and the right hand side $f$ as expansions of scaling functions at a particular scale $m$. For the purposes of the current work, it will suffice to say that the scaling function $\varphi$ is defined by a dilation equation of the form

$$\varphi(x) = \sum_{k=-\infty}^{\infty} h_k \varphi(2x - k)$$

and that the values of the scaling function may be calculated using this recursion. Compactly supported scaling functions, such as those belonging to the Daubechies family of wavelets, have a finite number of nonzero filter coefficients $a_k$. We denote the number of nonzero filter coefficients by $N$.

The wavelet-Galerkin solution of the periodic problem is slightly more complicated than the finite difference solution, since the solution procedure consists of solving a set of simultaneous equations in wavelet space and not in physical space. This means that we have to transform the right hand side function into wavelet space, solve the set of simultaneous equations to get the solution in wavelet space, and then transform the solution from wavelet space back into physical space. We consider the same problem as before

$$u_{xx} + \alpha u = f$$

The wavelet-Galerkin approximation to the solution $u(x)$ at scale $m$ is

$$u(x) \approx \sum_k \tilde{c}_k 2^{m/2} \varphi(2^m x - k) \tag{7}$$

$\tilde{c}_k$ are the wavelet coefficients of u , *i.e.* they define the solution in wavelet space. The transformation from wavelet space to physical space (or vice versa) can be easily accomplished using the FFT if the wavelet expansion is expressed as a convolution of two vectors. To do this we make the substitution

$$y = 2^m x$$

Then we may write

$$U(y) = u(x) = \sum_k c_k \varphi(y - k) \qquad c_k = 2^{m/2} \tilde{c}_k. \tag{8}$$

Now $u(x)$ is periodic in $x$ with period $d$, so that $U(y)$ is periodic in $y$ with period $2md$. Assuming that $d \in \mathbf{Z}$, so that $n = 2md \in \mathbf{Z}$ then $c_k$ is also periodic with period $2md$.

We may now discretize $U(y)$, letting $y$ take only integer values. This gives us values of $u(x)$ at all the dyadic points $x = 2^{-m} y$ i.e. the discretization of $u(x)$ depends on the scale we have chosen (or vice versa). Thus

$$U_i = U(i\Delta y) = U(y_i) \qquad i = 0, 1, 2, .., n - 1.$$

Equation (8) may then be written as

$$u_i = \sum_k c_k \varphi_{i-k} = \sum_k c_{i-k} \varphi_k$$

where $\varphi_k = \varphi(k)$. In matrix form this becomes

$$\begin{pmatrix} U_0 \\ U_1 \\ U_2 \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ U_{n-1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & \cdots & \varphi_{N-2} & \cdots & \varphi_2 & \varphi_1 \\ \varphi_1 & 0 & 0 & \cdots & 0 & \cdots & \varphi_3 & \varphi_2 \\ \varphi_2 & \varphi_1 & 0 & \cdots & 0 & \cdots & \varphi_4 & \varphi_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \varphi_{N-2} & \varphi_{N-3} & \varphi_{N-4} & \cdots & \cdots & \cdots & 0 & 0 \\ 0 & \varphi_{N-2} & \varphi_{N-3} & \cdots & \cdots & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & \varphi_{N-3} & \cdots & \varphi_1 & 0 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ c_{n-1} \end{pmatrix}$$

where $N$ is the number of nonzero filter coefficients. Thus the solution $U$ is related to the vector of its wavelet coefficients $c$ by the convolution

$$U = K_\phi * c \tag{9}$$

where the convolution kernel $K_\phi$ is the first column of the scaling function matrix.

Similarly, the wavelet expansion for $f(x)$ is

$$f(x) = \sum_k \tilde{g}_k 2^{m/2} \varphi(2^m x - k) \qquad k \in \mathbf{Z} \tag{10}$$

so that

$$F(y) = f(x) = \sum_k g_k \varphi(y-k) \qquad g_k = 2^{m/2}\tilde{g}_k.$$

Here too, the wavelet coefficients $g_k$ are periodic. Thus the right hand side vector $F$ is related to the vector of its wavelet coefficients $g$ by

$$F = K_\phi * g. \tag{11}$$

If we now substitute the expansions of $u(x)$ and $f(x)$ into our original differential equation we have

$$\frac{\partial^2}{\partial x^2} \sum_k c_k \varphi(y-k) + \alpha \sum_k c_k \varphi(y-k) = \sum_k g_k \varphi(y-k)$$

$$\text{i.e.} \quad \sum_k c_k 2^{2m} \varphi''(y-k) + \alpha \sum_k c_k \varphi(y-k) = \sum_k g_k \varphi(y-k).$$

Taking the inner product of both sides with $\varphi(y-i)$; $j \in \mathbf{Z}$ gives

$$2^{2m} \sum_k c_k \int \varphi''(y-k)\varphi(y-j)dy + \alpha \sum_k c_k \int \varphi(y-k)\varphi(y-j)dy =$$

$$= \sum_k g_k \int \varphi(y-k)\varphi(y-j)dy$$

and since the orthogonality of the translates of the scaling function implies that

$$\int \varphi(y-k)\varphi(y-j)dy = \delta_{kj}$$

where $\delta_{kj}$ is the Kronecker delta, we may write

$$2^{2m} \sum_k c_k \Omega_{j-k} + \alpha c_j = g_j. \tag{12}$$

In equation (12),

$$\Omega_{j-k} = \int \varphi''(y-k)\varphi(y-j)dy$$

are the connection coefficients which can be obtained from the very beginning using the equation. Remembering that $c_k$ and $g_k$ have period $n$, the matrix form of equation (12) becomes

$$Tc = g$$

where

$$T = 2^{2m} \begin{pmatrix} \rho & \Omega_{-1} & \dots & \Omega_{2-N} & 0 & \dots & 0 & \Omega_{N-2} & \dots & \Omega_1 \\ \Omega_1 & \rho & \dots & \Omega_{3-N} & \Omega_{2-N} & \dots & 0 & 0 & \dots & \Omega_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \Omega_{N-2} & \Omega_{N-1} & \dots & \rho & \Omega_{-1} & \dots & \dots & \dots & \dots & 0 \\ 0 & \Omega_{N-2} & \dots & \Omega_1 & \rho & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots & \dots & \rho & \Omega_{-1} & \dots & \Omega_{2-N} \\ \Omega_{2-N} & 0 & \dots & \dots & \dots & \dots & \Omega_1 & \rho & \dots & \Omega_{3-N} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \Omega_{-1} & \Omega_{-2} & \dots & 0 & 0 & \dots & \Omega_{N-2} & \Omega_{N-3} & \dots & \rho \end{pmatrix}$$

$\rho$ derives from equality

$$\rho = \Omega_0 + 2^{-2m}\alpha$$

and

$$c^t = (c_0, c_1, c_2, ...c_{n-1})$$

$$g^t = (g_0, g_1, g_2, ...g_{n-1}).$$

Note that as in the finite difference method, the solution of the differential equation has been reduced to the solution of a set of simultaneous equations. This time too, we are left with a differential operator $T$ which is a circulant matrix. If we denote the convolution kernel of the matrix $T$ by $K_\Omega$ then

$$K_\Omega * c = g. \tag{13}$$

Taking Fourier transforms of equations (9), (11) and (13)

$$\hat{U} = \hat{K}_\phi \hat{c}$$

$$\hat{F} = \hat{K}_\phi . \hat{g}$$

$$\hat{K}_\Omega . \hat{c} = \hat{g}$$

from which

$$\hat{U} = \hat{K}_\phi . ((\hat{F}/\hat{K}_\phi)/\hat{K}_\Omega)$$

or

$$\hat{U} = \hat{F}/\hat{K}_\Omega .$$

Taking inverse Fourier transforms gives us the solution U.

# 3 Incorporation of boundary conditions

Boundary conditions may be incorporated using the capacitance matrix method ([8], [7] and others). Suppose that we are required to solve the problem

$$u_{xx} + \alpha u = f \quad \text{in } [a, b]$$

with the Dirichlet boundary conditions

$$u(a) = u_a,$$

$$u(b) = u_b.$$

Again, suppose that $u(x)$ and $f(x)$ are periodic with period $d$, where $0 < a < b < d$. $f$ can be made periodic by making it zero outside the interval $[a, b]$. If necessary, the function $f$ may be extended smoothly outside $[a, b]$ so as to make it periodic. We already know how to solve the differential equation with periodic boundary conditions. Let the solution to this problem be $v(x)$. We may get the solution $u$ to the differential equation with Dirichlet boundary conditions at $a$ and $b$ by adding in another function $w(x)$ such that

$$u = v + w. \tag{14}$$

Since $v_{xx} + \alpha v = f$ in $[a, b]$, we must require that

$$w_{xx} + \alpha w = 0 \quad \text{in } [a, b].$$

However, at $a$ and $b$, we may let $w_{xx}$ take such values as to make $u$ satisfy the given boundary conditions. Thus we need the solution to

$$w_{xx} + \alpha w = X \quad \text{in } [0, d]$$

where

$$X = X(x) = X_a \delta(x - a) + X_b \delta(x - b)$$

$X_a$ and $X_b$ are constants and $\delta(x)$ is the delta function at $x = 0$.

Now the Green's function $G(x)$ of the differential equation is given by

$$G_{xx} + \alpha G = \delta(x) \tag{15}$$

from which

$$w(x) = G * X = X_a G(x - a) + X_b G(x - b).$$

We may solve for the periodic Green's function very easily using the periodic solvers developed in sections 2.1 and 2.2. It then remains to find the values of $X_a$ and $X_b$ which make $u$ satisfy the given boundary conditions. From equations (14) and (15)

$$w(a) = X_a G(0) + X_b G(a - b) = u_a - v(a),$$

$$w(a) = X_a G(b-a) + X_b G(0) = u_b - v(b).$$

Thus $X_a$ and $X_b$ are given by the solution to the $2 \times 2$ system of equations

$$\begin{pmatrix} G(0) & G(a-b) \\ G(b-a) & G(0) \end{pmatrix} \begin{pmatrix} X_a \\ X_b \end{pmatrix} = \begin{pmatrix} u_a - v(a) \\ u_b - v(b) \end{pmatrix}.$$

This completes the solution of $w$. The solution $u$ to the given boundary value problem is then obtained from equation (14).

## 3.1 Offsetting of boundary sources to control error

A method for improving the capacitance matrix method by offsetting boundary sources was introduced by [7]. Placing the sources at the boundaries in the wavelet solution method leads to a large error due to the finite support of the delta function in wavelet space ,*i.e.* the number of nonzero wavelet coefficients. In fact, the support of the delta function in wavelet space is the same as the support of the scaling function that is used to define it, since the wavelet coefficients of $\delta(x) = \sum_k g_k \varphi(y-k)$ are given by

$$g_k = 2^m \varphi(-k).$$

We may avoid the effect of the finite support of the delta function in wavelet space by offsetting the boundary sources. The magnitude $s$ of the offset should be such that the number of discretization points involved is at least equal to the support of the scaling function. Suppose that

$$a_1 = a - s,$$

$$b_1 = b + s.$$

Then

$$X = X(x) = X_a \delta(x - a_1) + X_b \delta(x - b_1)$$

so that $X_a$ and $X_b$ are given by the solution to the system of equations

$$\begin{pmatrix} G(a-a_1) & G(a-b_1) \\ G(b-a_1) & G(b-b_1) \end{pmatrix} \begin{pmatrix} X_a \\ X_b \end{pmatrix} = \begin{pmatrix} u_a - v(a) \\ u_b - v(b) \end{pmatrix}. \tag{16}$$

Using this offset error can be reduced considerably up to $10^5$ times.

## 3.2 Comparison of results

The wavelet solutions have much better precision but take slightly slower than the finite difference solution owing to the need to transform the samples from physical space into wavelet space and back again. This overhead becomes less significant as the sample size increases. More importantly, there is a negligible variation in computation time as the support of the wavelet increases. Thus the D12 wavelet solution compares extremely favorably with the finite difference solution.

# 4 Conclusion

The wavelet solution method for partial differential equations has obvious practical applications in engineering, such as in the static and dynamic analysis of structures and the solution of the heat equation. In engineering problems, we often require a quick rough estimate of the solution at the preliminary stage, which may later be refined as the design or investigation progresses. Wavelets have the capability of providing a multilevel description of the solution . The multiresolution property of wavelets, along with their localization property, suggests that we may obtain an initial coarse description of the solution with little computational effort and then successively refine the solution in regions of interest with a minimum of extra effort. Preliminary research indicates that wavelets are a strong contender to finite elements in this respect, however, further research is still required on the subject.

The wavelet method has been shown to be a powerful numerical tool for the fast and accurate solution of partial differential equations. The procedure described here shows that the solution to the differential equation is related to the equation's right hand side by a sequence of discrete convolutions which can be rapidly performed using the Fast Fourier Transform. Although the FFT implies that the solution is periodic, we may incorporate non periodic boundary conditions using the periodic Green's function. Solutions obtained using the Daubechies 6, 8, 10 and 12 coefficient wavelets have been compared with the finite difference solution and the wavelet solutions have been found to converge much faster than the finite difference solution (see also [7]). Although the wavelet solutions require slightly more computational effort than the finite difference solution, the gains in accuracy, particularly with the higher order wavelets, far outweigh the increase in cost. Furthermore, wavelets have the capability of representing solutions at different levels of resolution, which makes them particularly useful for developing hierarchical solutions to engineering problems.

# References


[1] I. Daubechies *Ten lectures on wavelets*, Society for Applied Mathematics Philadelphia, Pennsylvania 1992.

[2] `www.mit.edu/people/christov/ downloads/wg-pde.pdf` .

[3] `ejde.math.swt.edu/conf-proc/10/l2/lopes.pdf` .

[4] `www.wavelet.org/phpBB2/viewtopic.php?p=3021` .

[5] `wavelets.mit.edu/Kevin/Papers/WGSolnBVP.pdf` .

[6] `wavelets.mit.edu/Kevin/Papers/1D_PDE.pdf` .

[7] Qian and Weiss *Wavelets and the numerical solution of partial differential equations*, J. Comp. Phys., 106, 1, 155-175 (1993).

[8] Proskurowski and Widlund *'On the numerical solution of Helmholtz's equation by the capacitance matrix method'*, Math. Comp., 30, 135, 433-468 (1976).