The Wavelet-Galerkin Methods for PDEs

Joshua Im, Colin Neaville

June 21, 2025

Abstract

The wavelet-Galerkin method (WGM) is a wavelet-based method of finding approximate solutions to partial differential equations. The method involves approximating the solution of the PDE as a finite linear combination of scaling functions in V_0:

$$u(x,t) \approx \sum_{k=-M}^{N} \widehat{U}_k(t)\phi(x-k).$$

The coefficients $\widehat{U}_k(t)$ are then solved for such that an integral condition called the Galerkin condition is satisfied. This condition ensures that the approximate solution is the best solution in the subspace of V_0 in which the solution is approximated.

Since the WGM yields an approximate solution, it will not satisfy the PDE exactly. The error is especially significant for PDEs in which the solution decays slowly or involves shocks or high frequencies.

Here we discuss a few methods found in the literature which can be applied to solve PDEs for which the unmodified WGM does not provide an accurate solution. The first method involves scaling the domain of the PDE by a factor of $1/\epsilon$ by the change of variables $y = x/\epsilon$, where $\epsilon \ll 1$ is small. This effectively expands the features of the solution, so that they can be accurately captured by the scaling functions in V_0. The second method involves scaling the scaling functions down by a factor of 2^n by approximating the solution in V_n instead of in V_0:

$$u(x,t) \approx \sum_{k=-M_n}^{N_n} \widehat{U}_k(t)\phi_{n,k}(x).$$

Of course, the MRA which is used to approximate the solution has an effect on the accuracy of the WGM. Different scaling functions have different properties such as continuity, smoothness, and rate of decay at infinity. We will therefore investigate the accuracy of solutions obtained using different scaling functions as the basis for the approximation. In particular, we investigate accuracy when using the Shannon versus Daubechies scaling functions.

1 Background

1.1 Shannon and Daubechies Scaling Functions

In this section, we briefly describe each of the scaling functions to be used in our investigation of the WGM.

The Shannon scaling function has the closed form:

$$\phi^{(\text{sha})}(x) = \text{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} & x \neq 0 \\ 1 & x = 0. \end{cases}$$

Unlike other scaling functions such as the Haar and linear spline scaling functions, $\phi^{(\text{sha})}$ has the advantage of being continuous and smooth and is thus able to more accurately represent smooth solutions to PDEs. The disadvantage of using $\phi^{(\text{sha})}$ is that it is not compactly supported and has a relatively slow rate of decay at infinity. It should therefore in theory not be ideal to solve a PDE using the Shannon scaling function when the solution is expected to be compactly supported or decay quickly. We shall investigate how the slow decay rate of $\phi^{(\text{sha})}$ affects solutions obtained using the WGM.

The family of Daubechies scaling functions are obtained in the following manner. We begin by choosing the coefficients p_k associated with the member of the family of scaling functions to be constructed (e.g. Db4, Db6) and define the polynomial $P(z) = (1/2)\sum_k p_k z^k$. We then define

$$\phi_0(x) = \phi^{(\text{Haar})}(x) = \chi_{[0,1)}(x) \tag{1}$$

$$\phi_n(x) = \sum_k p_k \phi_{n-1}(2x - k) \text{ for } n \geq 1. \tag{2}$$

This sequence of functions can be shown to converge pointwise and in mean to a function $\phi^{(\text{Daub})}$ which, together with the sequence of spaces $\text{span}\{\phi(2^j x - k), k \in \mathbb{Z}\} \subset L^2(\mathbb{R})$, forms an MRA called the Daubechies MRA. As we work up the Daubechies hierarchy, the scaling function becomes increasingly continuously differentiable. The smoothness of the scaling function is significant to the WGM, as the solutions to many PDEs are highly smooth and therefore require smooth scaling functions to be approximated accurately. We shall investigate how the differing smoothness of members of the Daubechies family of scaling functions affects the accuracy of the WGM by comparing resulting solutions obtained using the Db4 and Db6 scaling functions.

1.2 The Wavelet-Galerkin Method

We now give a detailed description of how to apply the WGM to a general PDE. We follow the process described in [9] with clarifying details filled in.

Suppose that $U(x,t)$ is a solution to the PDE

$$F(U, U_t, \dots, U_x, U_{xx}, \dots) = 0. \tag{3}$$

We want to approximate the solution U in terms of scaling functions.

$$U(x,t) \approx \sum_{k\in\mathbb{Z}} \widehat{U}_k(t)\phi(x-k) \tag{4}$$

Note that this expansion in terms of scaling functions is not exact as this approximation is in V_0 and only so many details of the solution can be captured by functions in V_0.

We could instead approximate the solution with a function in $V_n := \text{span}\{\phi_{n,k}(x), k \in \mathbb{Z}\} = \text{span}\{2^{n/2}\phi(2^n x - k), k \in \mathbb{Z}\}$. As n increases, the approximation of U in V_n becomes more accurate, though requires heavier computation. The details for approximation in V_n are explained in subsequent sections.

Here we project U onto V_0 obtaining (4). By projecting onto V_0, we are assuming the smallest scale of variation to be the integers. There are multiple methods which can be used to apply these methods to a problem in which a smaller scale is required. This process is illustrated concretely in the section on numerical implementation of the Wavelet-Galerkin method.

We now further approximate U by removing all but finitely many terms from the projection. This is equivalent to projecting the approximation of U in V_0 to a finite-dimensional subspace of V_0. This gives

$$U(x,t) \approx \sum_{k=-M}^{N} \widehat{U}_k(t)\phi(x-k). \tag{5}$$

We can now see why the decay rate of U is important. If U is compactly supported, then U should be zero outside of some bounded interval, so we can include only the terms which contribute to within that bounded interval. Therefore, in the case that U is compactly supported, the equality in (5) can be made to be exact. If U is not compactly supported, fast decay would still give an approximation with high accuracy as we are dropping the terms that contribute less to U. As the decay rate decreases, the accuracy will also decrease.

Let $\widehat{U}$ be the approximation of U by (5). Since U is the solution to the equation (3) and $\widehat{U}$ is an approximation of it, we would want to make $\widehat{U}$ also satisfy the equation with a very small error. That is,

$$F(\widehat{U}, \widehat{U}_t, \dots, \widehat{U}_x, \widehat{U}_{xx}, \dots) \approx 0.$$

The value of F with $\widehat{U}$ is not actually zero, as $\widehat{U}$ is a finite-dimensional projection of U, and there is therefore an approximation error. There is a residue $R(x,t)$ resulting in the difference of the values of F. That is,

$$\begin{aligned} R(x,t) &= F(\widehat{U}, \widehat{U}_t, \dots, \widehat{U}_x, \widehat{U}_{xx}, \dots) - F(U, U_t, \dots, U_x, U_{xx}, \dots) \\ &= F(\widehat{U}, \widehat{U}_t, \dots, \widehat{U}_x, \widehat{U}_{xx}, \dots). \end{aligned}$$

Hence we want $\widehat{U}$ to satisfy $R(x,t) \approx 0$. Since it is hard to find such $\widehat{U}$, we reduce the problem to a weaker condition, called the Galerkin condition.

We say that $\widehat{U}$ satisfies the Galerkin condition if $\langle R(x,t), v\rangle = 0$ for each $v \in V_0$. That is,

$$\int_{-\infty}^{\infty} R(x,t)\cdot\phi(x-k)\,dx = 0 \tag{6}$$

for each $k \in \mathbb{Z}$. This is a natural analogue inside V_0 to solving the equation weakly in $L^2(\mathbb{R})$. Such $\widehat{U}$ making $R(x,t)$ satisfying the Galerkin condition will be the best approximate solution to (3) within V_0.

As we have the approximation of U by (5), we need to find the corresponding coefficients $\widehat{U}_k$ for each k. Since the expansion $\widehat{U}$ runs from $k = -M,\ -M+1,\ \ldots,\ N$, for each $k \in \{-M, -M+1, \ldots, N\}$, we use the Galerkin's condition. If k is fixed, then the equation

$$\int_{-\infty}^{\infty} F\left(\sum_{j=-M}^{N} \widehat{U}_j(t)\phi(x-j), \ldots\right) \cdot \phi(x-k) = 0$$

will give an equation with unknowns $\widehat{U}_{-M}$, $\widehat{U}_{-M+1}$, $\ldots$, $\widehat{U}_N$. Repeating the process for each $k \in \{-M, -M+1, \ldots, N\}$, we get a system of equations with $M+N+1$ equations and $M+N+1$ unknowns, which are $\widehat{U}_{-M}$, $\widehat{U}_{-M+1}$, $\ldots$, $\widehat{U}_N$.

Solving for these coefficients will give $\hat{U}$, and hence the approximation to the solution of the equation (3). If F is nonlinear, then the system will be nonlinear in the coefficients $\widehat{U}_k(t)$, which can be solved using iterative methods. If F is linear, then the system will also be linear in the coefficients, which is easy to solve.

2 Improving the approximations

Before proceeding with the details of solving particular PDEs with the wavelet Galerkin method we shall investigate methods by which the accuracy of solutions can be improved.

Recall that our approximation (5) is a projection of $U(x,t)$ onto the subspace V_0 of $L^2(\mathbb{R})$. Since V_0 is a low-resolution subspace of $L^2(\mathbb{R})$, $\widehat{U}$ can only capture limited features of U. Some factors that influence the amount of error resulting from the projection onto V_0 include:

- How much the energy of U is contained in V_0,
- The smoothness and steepness of U,
- Whether U is compactly supported and its rate of decay if its support is not finite.

If most of the energy of U is contained in V_0, projecting U to V_0 loses very little, while if a lot of energy is outside of V_0 (e.g. in high frequencies) then the projection may miss important parts of U. If U itself is smooth and slow-varying, the functions $\phi(x-k) \in V_0$ can still represent it well, even though they are slowly-varying and with low frequency. However, if U has high-frequency behaviors like shocks and jumps, then V_0 cannot capture all of those features. Finally, if U is compactly supported, then only finitely $\phi(x-k)$ functions will be needed, but if U is not, then truncating $\widehat{U}$ from $-M$ to N will introduce noticeable errors.

In the case where we need a projection with higher accuracy, there are two methods which can be applied.

2.1 Approximation inside V_n

Since the subspaces form a sequence

$$V_0 \subset V_1 \subset V_2 \subset \cdots \subset L^2(\mathbb{R}),$$

for nonnegative integers n and m, V_n is finer than V_m if and only if $n > m$. Thus V_0 is the coaresest space, which is expected that that the projection of U onto V_0 only captures low-frequency parts of U. To capture more detail, we instead use V_n for some positive integer n. Since the projection of U onto V_n is of the form

$$\widehat{U}(t) = \sum_{k=-M}^{N} \widehat{U}_k(t) 2^{n/2} \phi(2^n x - k).$$

Since the scaling functions $\phi(2^n x - k)$ in V_n are compressed compared to the scaling functions $\phi(x-k)$ in V_0, this allows us to capture higher frequency components of U, improving the accuracy of the approximation. This is because since the projection error is given by

$$\|U - \widehat{U}\|_{L^2(\mathbb{R})},$$

increasing n will reduce the error. Precisely, if $U \in H^s(\mathbb{R})$ where $H^s(\mathbb{R})$ is the Sobolev space of smoothness s, then (see [8])

$$\|U - \widehat{U}\|_{L^2(\mathbb{R})} \leq C 2^{-ns} \|U\|_{H^s} \tag{7}$$

where n is the scale level of the subspace V_n used in the approximation. Thus by choosing a sufficiently large n, we can make the error arbitrarily small. However, more basis functions are needed. If the finite projection of U onto V_0 ran from $k = -M$ to N, there were $M+N+1$ basis functions, but in V_n, k should run approximately from $-2^n \cdot M$ to $2^n \cdot N$, so there should be approximately $2^n(M+N+1)$ basis functions. For simplicity, we say that the sum runs from $-M_n$ to N_n. The amount of basis functions will approximately multiply by 2 if we decide to project onto the next finer space, i.e. V_{n+1}.

Since the Galerkin condition is satisfied if $\langle R(x,t), v \rangle = 0$ for each $v \in V_n$, we should have

$$\int_{-\infty}^{\infty} R(x,t) \cdot \phi(2^n x - k)\, dx = 0 \tag{8}$$

for each $k \in \mathbb{Z}$. Here the scaling factor $2^{n/2}$ is omitted since the right-hand side is zero. Now, the equation

$$\int_{-\infty}^{\infty} F\left(\sum_{j=-M_n}^{N_n} \widehat{U}_j(t) \phi(2^n x - j), \dots \right) \cdot \phi(2^n x - k) = 0$$

will give the system of ODEs with unknowns $\widehat{U}_{-M_n}$, $\widehat{U}_{-M+1}$, $\dots$, $\widehat{U}_{N_n}$. Thus as n increases, computational cost also increases, as the system of ODEs will involve more variables and coefficients.

2.2 Scaling the equation

Recall that approximation the solution U by the projection onto V_0 had a disadvantage that the scaling functions $\phi(x-k)$ could not capture the high-frequency behaviors. Instead of using compressed scaling functions, we may rescale the domain of the equation to improve the approximation.

Suppose, for simplicity, $U(x,t)$ is a solution to the equation (3). Define a new function

$$\widetilde{U}(y,t) = U(x,t),$$

where $x = \epsilon y$ and $\epsilon \ll 1$ is a small scaling factor. This process is called modulation and used frequently in solving PDEs (see [7], [1]). By this process, the partial derivatives becomes $\widetilde{U}_y = \epsilon U_x$, $\widetilde{U}_{yy} = \epsilon^2 U_{xx}$, and so on. Note that the time derivatives are unaffected by the rescaling, and only spatial derivatives introduce powers of ϵ. Thus

$$U_x = \frac{1}{\epsilon}\widetilde{U}_y$$

$$U_{xx} = \frac{1}{\epsilon^2}\widetilde{U}_{yy},$$

and so on. Thus the equation (3) becomes

$$F\left(\widetilde{U}\left(\epsilon y,t\right), \widetilde{U}_t\left(\epsilon y,t\right), \ldots, \frac{1}{\epsilon}\widetilde{U}_y(\epsilon y,t), \frac{1}{\epsilon^2}\widetilde{U}_{yy}(\epsilon y,t), \ldots\right) = 0. \tag{9}$$

The advantage is that if $U(x,t)$ has rapid oscillations, its frequency in $\widetilde{U}(y,t)$ is much lower. That is, $\widetilde{U}$ varies much more slowly in y. High-frequency behaviors are reduced with frequency, so now the scaling functions in V_0 will capture much more behaviors of $\widetilde{U}$ than the behaviors of U. Thus there is no need to use scaling functions in V_n for higher n, and the accuracy generally increases. For the error, in the original equation, we had the inequality (7). This corresponds to the inequality

$$\|\widetilde{U} - \widehat{\widetilde{U}}\|_{L^2(\mathbb{R})} \leq C\epsilon \|U\|_{H^s}$$

with $n = 0$, assuming U, $\widetilde{U} \in H^s(\mathbb{R})$ (see [7], [1]), which requires justification but formally is. This suggests that domain scaling is roughly equivalent to increasing n in frequency resolution.

However, since the domain is stretched, we need more basis functions. In the original projection, if we used basis functions from $k = -M$ to N, then in the scaling, these functions are used only to represent $-\epsilon M$ to ϵN in the original equation. Thus we need approximately $1/\epsilon$ multiple of the number of the basis function we use for the original approximation, which is still a lot of them.

2.3 Comparison of two methods and further improvements

The two methods both have advantages and disadvantages. Projection onto V_n does not change the equation (3) itself, so the domain is fixed and Galerkin formulation directly

applies. However, the amount of basis functions doubles at every level, so the number of calculations increases exponentially.

Rescaling the domain will significantly reduce the amount of basis functions needed, and will still capture high-frequency behaviors. However, the coefficients of the equation change, and we still need a lot of basis functions.

If the solution U does not contain high-frequency features everywhere, there is no need to use the compressed scaling functions on the interval in which does not contain high-frequency features. In this case, if the approximation is done by projection onto V_n, it is possible to combine scaling functions from different levels V_n. In the interval where U is smooth, we can use scaling functions for lower levels, i.e. V_n with small n, then we only have to use compressed scaling functions, i.e. V_n with large n where U has high-frequency behaviors. For rescaling the domain, we may only stretch the interval with high-frequency behaviors. This local projection depending on the behavior of U will reduce the computational cost, but in this paper we focus on a single-level approximation using V_n for a fixed n.

In practice, domain rescaling is useful when the entire solution exhibits high-frequency structure. However, when high-frequency components are spatially localized, an adaptive projection into different V_n regions is often more efficient.

3 Examples

3.1 Heat equation

Consider the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \qquad x \in (0,1),\, t > 0 \tag{10}$$

with the boundary conditions

$$u(0,t) = u(1,t) = 0$$

and initial condition

$$u(x,0) = \sin(\pi x).$$

The solution of this problem is clearly going to involve a minimum scale of variation much smaller than the scale of the integers. We must therefore either scale the PDE up or scale the scaling functions down.

To scale the PDE, we make the change of variables $y = 2^\ell x$ where ℓ is an positive integer. This results in

$$\frac{\partial U}{\partial t} = 4^\ell \frac{\partial^2 U}{\partial y^2}, \qquad y \in (0, 2^\ell),\, t > 0 \tag{11}$$

where $U(y,t)$ is the solution $u(x,t)$ rewritten in terms of the new variable y. The boundary conditions become

$$U(0,t) = U(2^\ell, t) = 0$$

and initial conditions

$$U(y,0) = \sin(2^{-\ell}\pi y).$$

We approximate the solution $U(y,t)$ in V_0:

$$U(y,t) \approx \widehat{U}(y,t) = \sum_{k=-M}^{N} \widehat{U}_k(t)\phi(y-k). \tag{12}$$

Taking the spatial derivatives and time derivatives gives the following equations:

$$\frac{\partial^2 \widehat{U}}{\partial y^2} = \sum_{k=-M}^{N} \widehat{U}_k(t)\phi''(y-k) \tag{13}$$

$$\frac{\partial \widehat{U}}{\partial t} = \sum_{k=-M}^{N} \widehat{U}'_k(t)\phi(y-k). \tag{14}$$

Note that the equation (11) is equivalent to $F(U) = 0$ where F is the linear operator $\partial/\partial t - 4^\ell(\partial^2/\partial y^2)$. For equation (11), the residual $R(x,t)$ is defined as $R(x,t) = \widehat{U}_t - 4^\ell \widehat{U}_{yy}$. Now fix $k \in \{-M, -M+1, \ldots, N\}$. Galerkin condition (6) gives

$$\int_0^{2^\ell} \left(\sum_{j=-M}^{N} \widehat{U}'_j(t)\phi(y-j) - 4^\ell \sum_{j=-M}^{N} \widehat{U}_j(t)\phi''(y-j) \right) \phi(y-k)\, dy = 0,$$

where the integral runs from 0 to 2^ℓ since the equation (11) was posed on $(0, 2^\ell)$. Expanding the equation and interchanging the integral and the sum, we obtain

$$\sum_{j=-M}^{N} \widehat{U}'_j(t) \int_0^{2^\ell} \phi(y-j)\phi(y-k)\, dy - \sum_{j=-M}^{N} \widehat{U}_j(t) \cdot 4^\ell \int_0^{2^\ell} \phi''(y-j)\phi(y-k)\, dy = 0.$$

Define

$$T_{kj} = \int_0^{2^\ell} \phi(y-j)\phi(y-k)\, dy \tag{15}$$

$$S_{kj} = 4^\ell \int_0^{2^\ell} \phi''(y-j)\phi(y-k)\, dy. \tag{16}$$

These integrals must be computed numerically. Galerkin's condition becomes

$$\sum_{j=-M}^{N} T_{kj}\widehat{U}'_j(t) = \sum_{j=-M}^{N} S_{kj}\widehat{U}_j(t),$$

which is a first-order ODE on $\widehat{U}_{-M}(t)$, $\widehat{U}_{-M+1}(t)$, $\ldots$, $\widehat{U}_N(t)$. We have $M+N+1$ equations for each $k \in \{-M, -M+1, \ldots, N\}$, resulting in a system of first-order ODES

$$\begin{pmatrix} T_{-M,-M} & \cdots & T_{-M,N} \\ \vdots & \ddots & \vdots \\ T_{N,-M} & \cdots & T_{N,N} \end{pmatrix} \begin{pmatrix} \widehat{U}'_{-M}(t) \\ \vdots \\ \widehat{U}'_N(t) \end{pmatrix} = \begin{pmatrix} S_{-M,-M} & \cdots & S_{-M,N} \\ \vdots & \ddots & \vdots \\ S_{N,-M} & \cdots & S_{N,N} \end{pmatrix} \begin{pmatrix} \widehat{U}_{-M}(t) \\ \vdots \\ \widehat{U}_N(t) \end{pmatrix}. \tag{17}$$

This can be expressed succinctly as

$$T\widehat{\boldsymbol{U}}' = S\widehat{\boldsymbol{U}}.$$

The initial condition is

$$\widehat{U}_k(0) = \int_0^{2^\ell} \sin(2^{-\ell}\pi y)\phi(y-k)\ dy$$

for each $k \in \{-M, -M+1, \ldots, N\}$. The equation (17) is solvable and will give the coefficients $\widehat{U}_{-M}(t)$, $\widehat{U}_{-M+1}(t)$, $\ldots$, $\widehat{U}_N(t)$, hence to approximation of the solution of equation (10) by (12).

If instead of scaling the PDE, we approximate the solution in V_n for some $n > 0$ we obtain

$$\widehat{u}(x,t) = \sum_{k=-M}^{N} \widehat{U}_k(t)\phi_{n,k}(x).$$

The Galerkin condition requires that

$$\sum_{j=-M}^{N} \left(\widehat{U}_j'(t) \int_0^1 \phi_{n,j}(x)\phi_{n,k}(x)dx - \widehat{U}_j(t) \int_0^1 \phi_{n,j}''(x)\phi_{n,k}(x)dx \right) = 0.$$

Define matrices $\widetilde{T}^n$ and $\widetilde{S}^n$ by

$$\widetilde{T}^n_{kj} = \int_0^1 \phi_{n,j}(x)\phi_{n,k}(x)dx$$

$$\widetilde{S}^n_{kj} = \int_0^1 \phi_{n,j}''(x)\phi_{n,k}(x)dx.$$

Then the Galerkin condition can be reduced to

$$\widetilde{T}^n\widehat{\boldsymbol{U}}' = \widetilde{S}^n\widehat{\boldsymbol{U}}.$$

Once again, the integrals $\widetilde{T}^n_{kj}$ and $\widetilde{S}^n_{kj}$ must be computed numerically.

3.2 Burgers' equation

Consider the viscous Burgers' equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \qquad x \in (0,1),\ t > 0 \tag{18}$$

with the boundary conditions

$$u(0,t) = u(1,t) = 0$$

and initial condition

$$u(x,0) = \sin(\pi x).$$

If we make the change of variables $y = 2^\ell x$, the problem becomes

$$\frac{\partial U}{\partial t} + 2^\ell U \frac{\partial U}{\partial y} = 4^\ell \nu \frac{\partial^2 u}{\partial y^2}, \qquad y \in (0, 2^\ell),\, t > 0 \tag{19}$$

with boundary and initial conditions

$$U(0,t) = u(2^\ell, t) = 0$$

$$U(y,0) = \sin(2^{-\ell}\pi y).$$

where $U(y,t)$ is the solution written in terms of the new variable y. The approximation of the solution $U(y,t)$ by the Wavelet-Galerkin method is given by

$$U(y,t) \approx \widehat{U}(y,t) = \sum_{k=-M}^{N} \widehat{U}_k(t)\phi(y-k). \tag{20}$$

Then the derivatives $\widehat{U}_t$, $\widehat{U}_x$, and $\widehat{U}_{xx}$ are

$$\frac{\partial \widehat{U}}{\partial t} = \sum_{k=-M}^{N} \widehat{U}_k'(t)\phi(y-k) \tag{21}$$

$$\frac{\partial \widehat{U}}{\partial y} = \sum_{k=-M}^{N} \widehat{U}_k(t)\phi'(y-k) \tag{22}$$

$$\frac{\partial^2 \widehat{U}}{\partial y^2} = \sum_{k=-M}^{N} \widehat{U}_k(t)\phi''(y-k) \tag{23}$$

and the nonlinear term $\widehat{U} \cdot \widehat{U}_y$ becomes

$$\widehat{U} \cdot \widehat{U}_y = \left(\sum_{k=-M}^{N} \widehat{U}_k(t)\phi(y-k) \right) \cdot \left(\sum_{k=-M}^{N} \widehat{U}_k(t)\phi'(y-k) \right)$$

For equation (18), the residual $R(y,t)$ is defined as

$$R(y,t) = \sum_{k=-M}^{N} \widehat{U}_k'(t)\phi(y-k) + 2^\ell \left(\sum_{k=-M}^{N} \widehat{U}_k(t)\phi(y-k) \right) \cdot \left(\sum_{k=-M}^{N} \widehat{U}_k(t)\phi'(y-k) \right)$$
$$- 4^\ell \nu \sum_{k=-M}^{N} \widehat{U}_k(t)\phi''(y-k).$$

The Galerkin condition states that

$$\int_0^{2^\ell} R(y,t)\phi(y-k)dy = 0$$

for each $k \in \{-M, -M+1, \ldots, N\}$. To apply this condition we must compute the integrals

$$T_{kj} = \int_0^{2^\ell} \phi(y-j)\phi(y-k)\ dy$$

$$S_{kj} = 4^\ell \int_0^{2^\ell} \phi''(y-j)\phi(y-k)\ dy,$$

which are (15) and (16), and also the integral

$$\Omega_{kij} = 2^\ell \int_0^{2^\ell} \phi(y-i)\phi'(y-j)\phi(y-k)dy.$$

for each k. The numerical value of these T_{kj}, S_{kj}, and Ω_{kij} integrals will be used to find the coefficients $\widehat{U}_{-M}(t)$, $\widehat{U}_{-M+1}(t)$, $\ldots$, $\widehat{U}_N(t)$, and hence the approximation $\widehat{U}$. Due to the nonlinear term $\widehat{U} \cdot \widehat{U}_y$, the resulting term of equations for $\widehat{U}_k(t)$ is nonlinear and generally requires iterative numerical methods.

If we instead solve the problem with higher resolution wavelets, we obtain

$$u(x,t) = \sum_{k=-M}^{N} \widehat{U}_k(t)\phi_{n,k}(x)$$

and the Galerkin condition becomes

$$\sum_{j=-M}^{N} \widehat{U}_j'(t) \int_0^1 \phi_{n,j}(x)\phi_{n,k}(x)dx + \sum_{j=-M}^{N}\sum_{m=-M}^{N} \widehat{U}_j(t) \int_0^1 \phi_{n,j}(x)\phi_{n,m}'(x)\phi_{n,k}(x)dx$$
$$- \nu \sum_{j=-M}^{N} \widehat{U}_j(t) \int_0^1 \phi_{n,j}''(x)\phi_{n,k}(x)dx = 0.$$

This equation can be solved by iterative methods once the integrals are computed numerically.

4 Numerical implementations

We now attempt to solve the heat equation numerically using the wavelet-Galerkin method. The exact process of solving numerically depends on the choice of multi-resolution analysis. We shall use the Shannon and Daubechies MRAs.

In the Shannon MRA, the scaling function has the exact form

$$\phi(x) = \mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}.$$

Since (11) is posed on the interval $(0, 2^\ell)$ and $\phi(x)$ has a period of 2 in the Shannon MRA, it is reasonable to set $M = 0$ and $N = 2^\ell - 1$ so that $k \in \{0, 1, 2, \ldots, 2^\ell - 1\}$. The initial conditions are given by the integral

$$\widehat{U}_k(0) = \int_0^{2^\ell} \sin(2^{-\ell}\pi y)\frac{\sin(\pi(y-k))}{\pi(y-k)}dy.$$

The integrals T_{kj} and S_{kj} in (15) and (16) become

$$T_{kj} = \int_0^{2^\ell} \frac{\sin(\pi(y-j))\sin(\pi(y-k))}{\pi^2(y-j)(y-k)} dy.$$

$$S_{kj} = 4^\ell \int_0^{2^\ell} \frac{\left(2\left(y-j\right) - \pi^2\left(y-j\right)^3\right)\sin\left(\pi\left(y-j\right)\right) - 2\pi\left(y-j\right)^2\cos\left(\pi\left(y-j\right)\right)}{\pi\left(y-j\right)^4} \times \frac{\sin(\pi(y-k))}{\pi(y-k)} dy.$$

For the Daubechies MRA, we once again set $M = 0$ and $N = 2^\ell$. The Daubechies scaling function has no closed form, though it is possible to obtain arbitrarily accurate approximations of $\phi^{(\text{Daub})}$ by repeatedly iterating the defining equation (1) where the coefficients p_k depend on which scaling function is being constructed. For both the Db2 and Db6 wavelets, we use the 6th iteration of the equation to obtain an approximation $\phi_6^{(\text{Daub})}$ of the scaling function. The initial conditions and relevant integrals are then given by

$$\widehat{U}_k(0) = \int_0^{2^\ell} \sin(2^{-\ell}\pi y)\phi_6^{(\text{Daub})}(y-k)dy$$

$$T_{kj} = \int_0^{2^\ell} \phi_6^{(\text{Daub})}(y-j)\phi_6^{(\text{Daub})}(y-k)dy$$

$$S_{kj} = \int_0^{2^\ell} \phi_6^{(\text{Daub})\prime\prime}(y-j)\phi_6^{(\text{Daub})}(y-k)dy$$

All that is left to do is compute these integrals numerically and solve the resulting system of ODEs. In MATLAB, we use the trapezoidal sum function with a mesh consisting 2^{10} equally points to evaluate each of the integrals and use Runge-Kutta $(4, 5)$ to solve the system of ODEs. We solve the PDE up to time $t = 0.25$ with 10 time steps.

The figures below show the $t = 0.25$ profiles of the exact versus approximate solutions due to Wavelet-Galerkin for each MRA and various values of ℓ. The dashed line represents the approximate solution and the solid line the exact solution.

These solutions have exactly the expected behavior. Increasing the value of ℓ yields more accurate solutions in all MRAs since more details are able to be captured by the scaling functions in V_0 the more the equation is expanded.

According to figure 1, the Shannon scaling function, due to its slow rate of decay, requires large values of ℓ to accurately approximate the compactly supported solution. However, the solutions obtained using the Shannon scaling function are smooth since $\phi^{(\text{Sha})}$ is highly smooth.

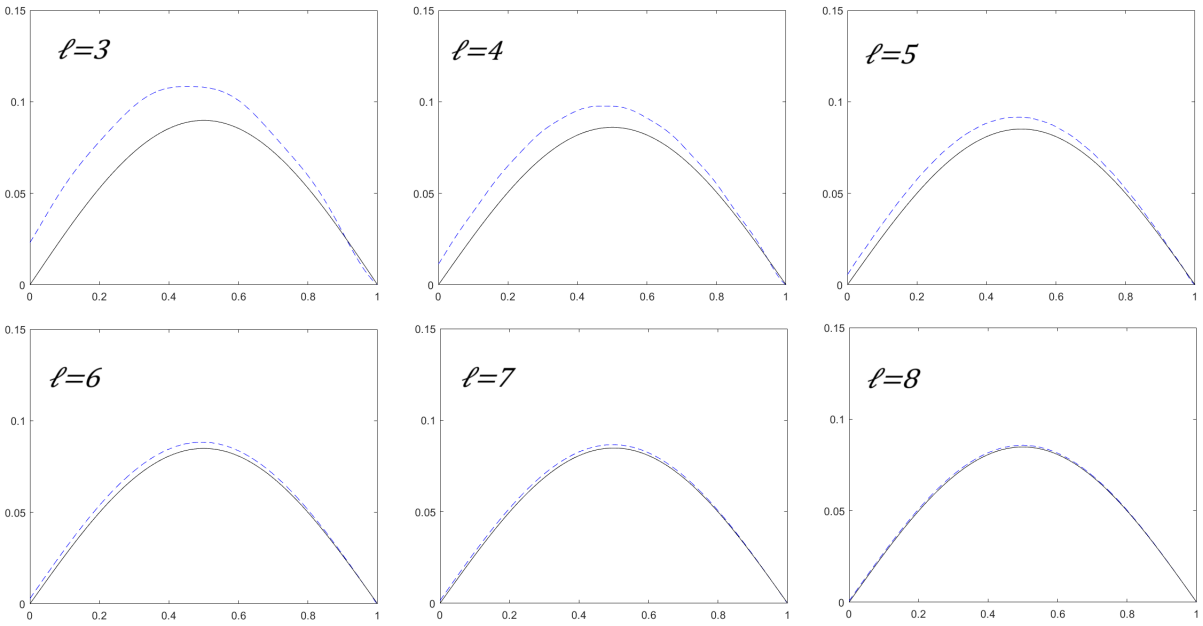

Figure 1: Approximate solutions of the heat equation at $t = 0.25$ using the Shannon scaling function.

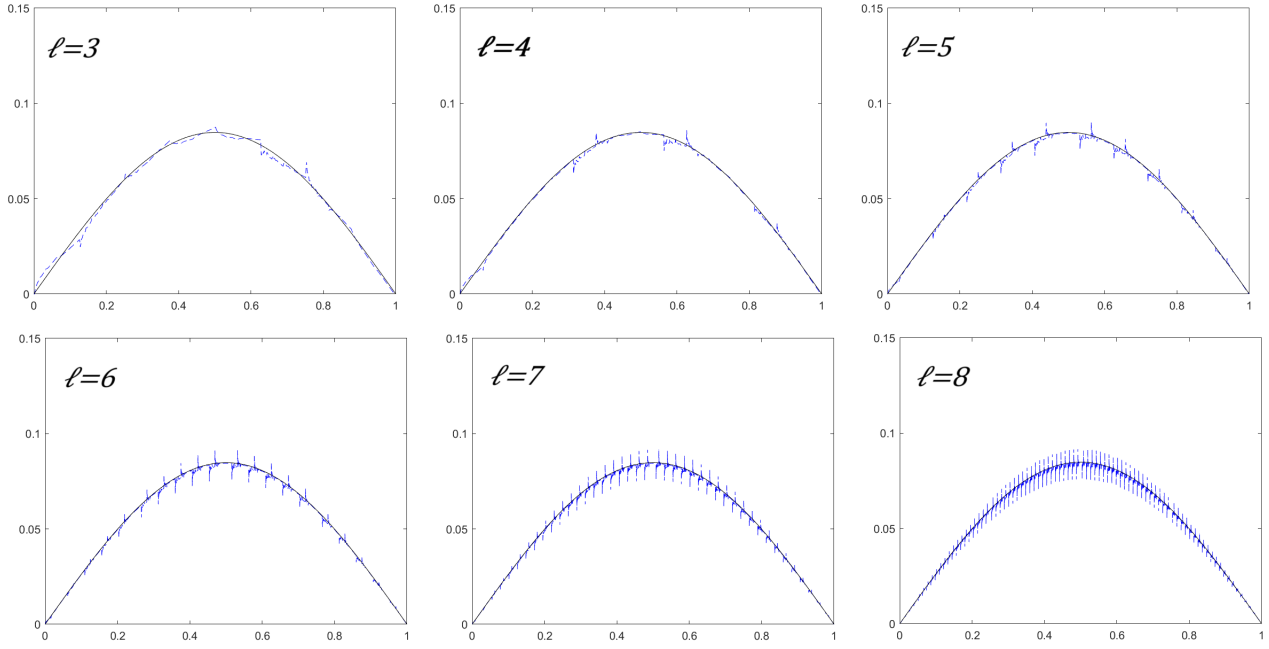

Figure 2: Approximate solution at $t = 0.25$ using the Daubechies-2 (Db2) scaling function.

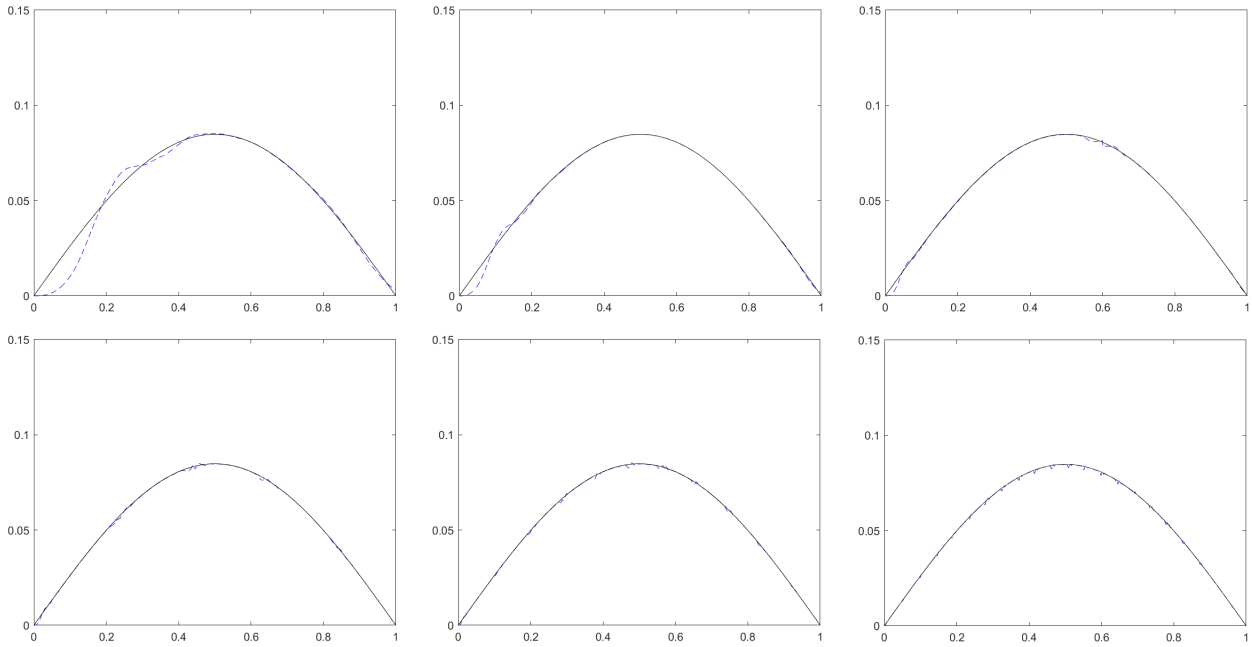

Figure 3: Approximate solutions at $t = 0.25$ using the Daubechies-6 (Db6) scaling function.

The solutions obtained using the Daubechies wavelets are very accurate even for small values of ℓ due to the compact support of $\phi^{(\text{Daub})}$. However, they contain some spiky artifacts due to the limited degree of smoothness of the $\phi^{(\text{Daub})}$. This is especially apparent for the solution obtained using the Db2 scaling function which is not even once continuously differentiable. Better solutions could be obtained by using Daubachies scaling functions with a higher degree of smoothness (e.g. Db12) though this is more computationally expensive as smoother Daubechies scaling functions also have larger support lengths.

References

[1] Grégoire Allaire. Two-scale convergence and homogenization of periodic structures. In *School on Homogenization*, Trieste, Italy, September 1993. ICTP, Trieste. Lecture notes presented at the ICTP School on Homogenization, September 6–17, 1993.

[2] Kevin Amaratunga and John R. Williams. Wavelet-galerkin solution of boundary value problems. *Archives of Computational Methods in Engineering*, 1997.

[3] S. H. Behiry, J. R. Cannon, H. Hashish, and A. I. Zayed. Comparison of a wavelet-galerkin procedure with a crank-nicolson-galerkin procedure for the diffusion equation subject to the specification of mass. *Journal of Computational and Applied Mathematics*, 98(2):245–270, 1998.

[4] Albert Boggess and Francis J. Narcowich. *A First Course in Wavelets with Fourier Analysis*. Wiley, Hoboken, NJ, 2nd edition, 2009.

[5] Ming-Quayer Chen, Chyi Hwang, and Yen-Ping Shih. The computation of wavelet-galerkin approximation on a bounded interval. *International Journal for Numerical Methods in Engineering*, 39(17):2921–2944, 1996.

[6] Simon Jones and Mathias Legrand. The wavelet-galerkin method for solving pde's with spatially dependent variables. In *Proceedings of the 19th International Congress on Sound and Vibration (ICSV19)*, Vilnius, Lithuania, 2012. HAL Id: hal-00719744.

[7] Pius Kirrmann, Guido Schneider, and Alexander Mielke. The validity of modulation equations for extended systems with cubic nonlinearities. *Proceedings of the Royal Society of Edinburgh Section A: Mathematics*, 122(1-2):85–91, 1992.

[8] Stéphane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[9] Sam Qian and John Weiss. Wavelets and the numerical solution of partial differential equations. *Journal of Computational Physics*, 106:155–175, 1993.