Wavelets for 1D PDEs

ME5204 Finite Element Analysis (course project)

Patel Shrey Rakeshkumar

Department of Mechanical Engineering,
Indian Institute of Technology Madras, Chennai - 600036, India.
me21b138@smail.iitm.ac.in

Abstract

This report explores the application of wavelet-based methods in finite element analysis (FEA) for solving partial differential equations (PDEs). Wavelets, known for their ability to capture local information efficiently, are integrated with traditional FEA methods to achieve improved accuracy and flexibility in solving PDEs. The primary goal of this study is to investigate the potential of wavelet bases, specifically using multi-scale hierarchical Schauder bases, as a tool for refining solutions to PDEs. The study introduces a novel approach that combines coarse-grid FEA solutions with wavelet refinement, leading to more detailed information in localized regions of the mesh. We analyze the convergence behavior by examining the change in the gain function, ϵ_{gain} , as additional wavelet layers are introduced. The results suggest that the wavelet method provides a computationally feasible alternative to traditional methods, offering modularity and enhanced local resolution while avoiding the computational expense of fully wavelet-based solutions.

1 Introduction

Wavelets are a relatively recent mathematical innovation used to decompose data, functions, and differential operators into components, each associated with a resolution that aligns with its scale. Their versatility has led to significant advancements in applying wavelet bases to solve partial differential equations (PDEs). Several competing formulations have emerged, including wavelet-Galerkin methods, wavelet-collocation techniques, and reproducing kernel methods. A comprehensive survey of wavelet-based methods for solving PDEs can be found in [1].

Recently, the wavelet-Galerkin method has garnered substantial attention due to its generality, provided by the Galerkin framework, and the flexibility it offers in implementing and testing alternative bases. There has been a shift from traditional Daubechies wavelets, known for their L^2 orthogonality, toward wavelets with weaker orthogonality properties, such as biorthogonal and semi-orthogonal wavelets.

This project explores the development of multiscale finite elements using a hierarchical Schauder basis, leveraging orthogonality concerning the energy norm with a nodal change of basis. In one dimension (1D), the uniform stability of the Schauder basis in $\mathscr{H}^1$ ensures uniformly bounded condition numbers—independent of the refinement level—for elliptic operators in the Galerkin framework.

In section 2, I have talked about the PDEs in consideration for this project purposes and mathematics behind wavelets. In section 3, I have talked about discretization approach for the

multi-scale finite elements. in that section i have mention how to create M, K, and F matrices for multi-scale finite elements. In section 4, I have shown how to calculate desired variable with the help of wavelet basis with a simple example also I have brifly explained about the stopping condition for intra elemental mesh convergence. Then section 4 contains conclusions followerd by references.

2 Governing equations and weak form

In this project, We will be focusing on the partial differential equation of the form given in the Eq. (1).

$$-\epsilon\nabla^2 u + \gamma u = f \text{ on } \Omega, \tag{1}$$

with Dirichlet boundary conditions and Neumann boundary conditions

$$u = u_i \text{ on } \Gamma_{D_i}, \tag{2}$$

$$\frac{\partial u}{\partial x} = v_j \text{ on } \Gamma_{N_j}, \tag{3}$$

The weak form of the problem, after introducing the finite dimensional subspace, $\mathcal{V}_k$, is

$$\int_\Omega \varepsilon\nabla u_k \cdot \nabla \mathbf{\Phi_k}\, \mathrm{d}\mathbf{\Omega} + \int_{\mathbf{\Omega}} \gamma \mathbf{u_k}\mathbf{\Phi_k}\, \mathrm{d}\mathbf{\Omega} = \int_{\mathbf{\Omega}} \mathbf{f}\mathbf{\Phi_k}\, \mathrm{d}\mathbf{\Omega} + \int_{\mathbf{\Gamma}} \epsilon\nabla \mathbf{u_k}\mathbf{\Phi_k}\mathrm{d}\mathbf{\Gamma} \tag{4}$$

Now, we will examine the mathematics behind wavelets which can be found in [2] Here, $u_k \in \mathcal{H}_0^1$, and $\mathbf{\Phi_k}$ is a finite dimensional basis for the subspace $\mathcal{V}_k$ at the scale k. In matrix form, this may be written as

$$[\mathbf{M_k} + \mathbf{K_k}]\mathbf{u_k} = \mathbf{F_k} \tag{5}$$

where $\mathbf{M_k}$ is mass matrix, and $\mathbf{K_k}$ is stiffness matrix associated with scale k. $\mathbf{M_k}$, $\mathbf{K_k}$ and $\mathbf{F_k}$ all are functions of $\mathbf{\Phi_k}$.

Now, consider an alternative basis for $\mathcal{V}_k$,

$$\mathbf{\Psi_k} = \mathbf{W_k^T}\mathbf{\Phi_k} \tag{6}$$

where $\mathbf{W_k}$ is a non-singular $N \times N$ matrix. From this transformation, $u_k = \Delta u^T \mathbf{\Psi}$ may also be found by solving

$$[\mathbf{M_k}(\mathbf{\Psi_k}, \mathbf{\Psi_k}) + \mathbf{K_k}(\mathbf{\Psi_k}, \mathbf{\Psi_k})]\mathbf{\Delta u_k} = \mathbf{F_k}(\mathbf{\Psi_k}) \tag{7}$$

for the multi-scale representation of the field, Δu. The linear system, $\mathbf{A_k^\Psi}$, that derives from the change of basis may also be viewed as the linear system obtained by preconditioning $\mathbf{A_k^\Phi}$ with $\mathbf{W_k}$,

$$\mathbf{A_k^\Psi} = \mathbf{W_k^T}\mathbf{A_k^\Phi}\mathbf{W_k}. \tag{8}$$

The matrix W_k is a wavelet transform and provides the mechanism for decomposing a field into multiple scales. In order to understand the role of the wavelet transform, consider that

$$\mathcal{V}_0 \subset \mathcal{V}_1 \subset \cdots \subset \mathcal{V}_k \subset \cdots \tag{9}$$

is a one-sided sequence of nested finite-dimensional subspaces of $\mathcal{H}$, Hilbert space, such that $\overline{\bigcup \mathcal{V}_k} = \mathcal{H}$. Now, define the "coarse-grid" wavelet subspace, $\mathcal{W}_0 = \mathcal{V}_0$ and, for $k \geq 1$, choose $\mathcal{W}_k$ so that completion of subspace $\mathcal{V}_{k-1}$ may be achieved via a direct sum,

$$\mathcal{V}_k = \mathcal{V}_{k-1} \oplus \mathcal{W}_k. \tag{10}$$

Here $\oplus$ denotes a direct sum, not an orthogonality direct sum.

The wavelet transform may be written recursively in terms of a two-scale transform as

$$\mathbf{W_k} = \mathbf{T_K} \begin{bmatrix} T_{k-1} & 0 \\ 0 & I_{k-1} \end{bmatrix} \cdots \begin{bmatrix} T_1 & 0 \\ 0 & I_1 \end{bmatrix} \tag{11}$$

where I_k is the $N_\Psi \times N_\Psi$ identity matrix with $N_\Psi = dim(\mathcal{W}_K)$. Here, T_k is a two-scale transform such that

$$\begin{Bmatrix} \mathbf{\Phi_{k-1}} \\ \mathbf{\Psi_k} \end{Bmatrix} = \mathbf{T_k^T \Phi_k} \tag{12}$$

As an aside, one very important aspect of this formulation is that it does not require the decomposition matrix $\mathbf{W_k^{-1}}$. This is significant because it permits the relaxation of strict orthogonality in the selection of the wavelet bases.

This basis consists of the elements of the linear finite element basis at multiple scales where the relationship between $\mathbf{\Psi_k}$ and $\mathbf{\Phi_k}$ is simply

$$\psi_k^j = \phi_k^{2j-1}, \quad j = 1, \ \ldots, \mathrm{Nnp}_k, \tag{13}$$

where Φ_k is the linear finite element basis. Here the superscript j indicates the node number, k indicates the scale, and Nnp_k is the number of node points at scale k. The Schauder basis is illustrated in Figure 1.

With the relationship between basis elements in the Schauder basis given in the Eq. (13), the construction of the two-scale transformation may be written in terms of the grid spacing. In order to define the wavelet transformation, let $\mathbf{H_k}$ be the $(\mathrm{Nnp}_k - 1) \times (\mathrm{Nnp}_k - 1)$ matrix $\mathbf{H_k} = (\mathbf{h_k^{j,j'}})_{\mathbf{j',j}}$ and let $\mathbf{G_k}$ be the $(\mathrm{Nnp}_k - 1) \times (\mathrm{Nnp}_{k-1})$ matrix $\mathbf{G_k} = (\mathbf{g_k^{j,j'}})_{\mathbf{j',j}}$. Then, the two-scale transformation for the Schauder basis is given by

$$T_k = [\mathbf{H_k} | \mathbf{G_k}]. \tag{14}$$

The function values $h_k^{j,j'}$ and $g_k^{j,j'}$ for ϕ_{k-1}^j and ψ_k^j respectively at the node points $x_k^{j'}$ are given by

$$h_k^{j,j'} = \begin{cases} \frac{\Delta^{j'}}{\Delta_{k-1}^k}, & j' = 2j-1 \\ 1, & j' = 2j \\ \frac{\Delta^{j'+1}}{\Delta_{k-1}^k}, & j' = 2j+1 \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

$$g_k^{j,j'} = \delta_{j',2j-1}. \tag{16}$$

In one dimension, the element length from x_k^{j-1} to x_k^j is defined as $\Delta_k^j = x_k^j - x_k^{j-1}$.

The action of $h_k^{j,j'}$ applied at scale k is to average, while $g_k^{j,j'}$ acts to inject at scale k . This can be seen in Figure 1 where the relationship between basis elements is given by

$$\phi_{k-1}^j = \sum_{j'} h_k^{j,j'} \phi_k^{j'} \tag{17}$$

$$\psi_k^j = \sum_{j'} g_k^{j,j'} \psi_k^{j'} \tag{18}$$

With the Schauder basis defined. The stability of the Schauder basis are extensively explained in [2].

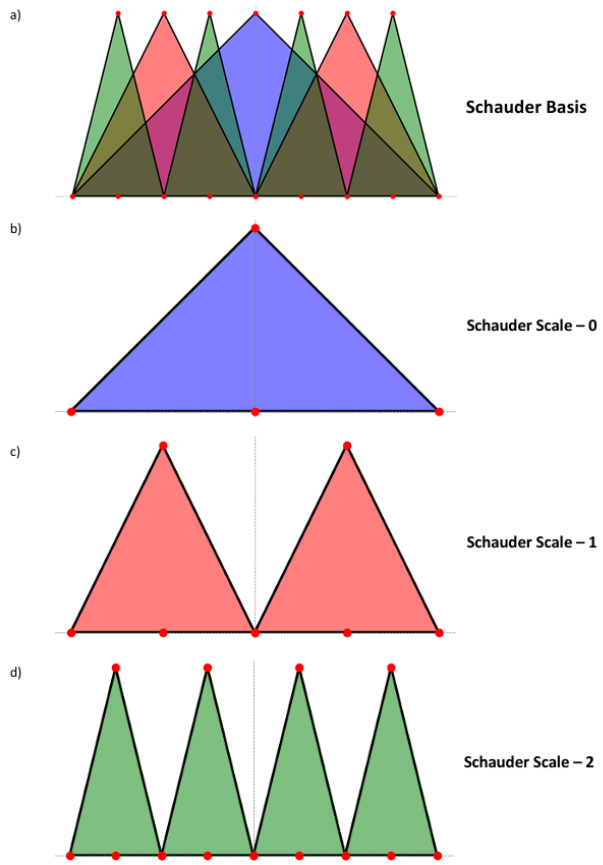

Figure 1: The hierarchical Schauder basis for scale k = 2 with 8 elements and 9 nodes showing a the composite basis, b the basis functions for scale-0, c the basis functions for scale-1, d the basis functions for scale-2

3 Discretization approach - The multi-scale finite element

This section outlines the formulation of 1-D multi-scale finite elements based on the hierarchical Schauder basis. A change of basis is implemented at the element level to enable the use of the well-established finite element assembly procedure.

The process begins by selecting an element in the finite element basis and then applying wavelets within the parametric space [-1,1]. The multi-scale element formulation starts with the linear finite element, where the shape functions are:

$$N_i = \frac{1}{2}(1 - \xi_i \xi). \tag{19}$$

Here ξ represents the natural coordinate, ξ_i denotes the nodal values, i = 0, 1, and $\xi_0 = 1$, $\xi_1 = -1$. The natural coordinate varies over the range $-1 \leq \xi \leq 1$.

The concept of scale is introduced at the element level by incorporating successive layers of wavelet functions. At Scale-1, a wavelet shape function is defined such that it equals 1 at $\xi = 0$. Each subsequent scale introduces twice as many wavelet shape functions as the preceding scale. Consequently, after k levels of wavelets, the total number of wavelet shape functions will be $2^k - 1$.

At each new level, every sub-element from the previous level is divided into two sub-elements of equal size. This hierarchical refinement allows for the systematic addition of detail at finer scales.

The structure and progression of these wavelet layers are illustrated in Figure 2, providing a clearer understanding of the scaling concept.

At Scale-1, the pseudo-wavelet is

$$\psi_1(\xi) = \begin{cases} 1+\xi & \text{if} \quad -1 \le \xi \le 0 \\ 1-\xi & \text{if} \quad 0 \le \xi \le 1 \end{cases} \tag{20}$$

More generally, the pseudo-wavelets for the multi-scale formulation can be expressed in terms of the translates and dilates of $\psi(\xi)$ as:

$$\psi_k(\xi) = \psi(\tilde{\xi}), \tag{21}$$

where

$$\tilde{\xi} = 2^{k-1}(1+\xi) - 2j - 1 \tag{22}$$

and

$$\psi(\tilde{\xi}) = \begin{cases} \text{ascending if } \xi \in [2^{2-k}j - 1, 2^{2-k}j + 2^{1-k} - 1] \\ \text{descending if } \xi \in [2^{2-k}j + 2^{1-k} - 1, 2^{2-k}j + 2^{2-k} - 1] \end{cases} \tag{23}$$

Here k indicates the scale, and j indicates the translates in the element parametric space.

The derivatives of the shape functions produce constant functions that are orthogonal to the derivatives of the pseudo-wavelets across all scales. Furthermore, the derivatives of the pseudo-wavelets correspond to Haar wavelets, as depicted in Figure 2.

The 1-D multi-scale operators

For each element, it is necessary to construct the K, M, and F matrices to solve the desired PDEs. While the fundamental principles underlying the construction of these matrices remain similar to traditional approaches, a different methodology is required to accommodate the multi-scale framework effectively.

The levels within a particular element can be divided into two sections: 1) the coarse-grid (scale-0), and 2) the finer-grid or wavelet-grid. To compute the M, K, and F matrices, it is essential to understand how the shape functions interact both within their respective sections and with those of other sections. We will start with the K matrix, followed by the M matrix, and conclude with the F matrix. Only the coarse-scale terms contribute to the element assembly process, as all internal pseudo-wavelet functions require only point evaluations and do not depend on information from outside the element.

Note: The provided code is capable of solving PDEs where ϵ and γ are functions of x. However, for simplicity and ease of illustration, we assume ϵ and γ to be constant within a given element in the explanation below.

The computation of the stiffness for the model problem follows a straightforward procedure, beginning with the coarse-grid stiffness.

At the element level (i.e., Scale-0), this calculation is simple and can be performed using the standard finite element analysis (FEA) procedure.

$$K_0^e = \frac{\epsilon}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \tag{24}$$

where h is the node spacing for the coarse-grid.

Considering the intra-layer pseudo-wavelet shape functions, we observe that they do not interact with each other, as illustrated in Figure 2. Therefore, we can assign their values as 0.

For inter-layer interactions, we note that the active region of each pseudo-wavelet is fully covered by the pseudo-wavelets from the preceding layer, or it does not interact at all. As a result, their

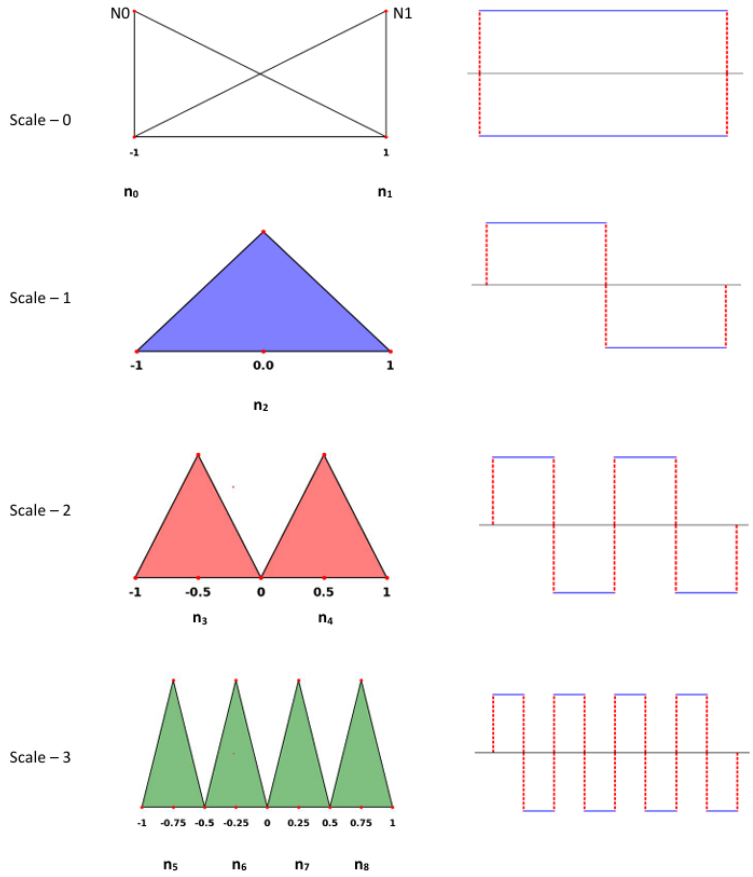

Figure 2: Basis elements and their piecewise derivatives for the one-dimensional multi-scale element with three refinement scales. The derivatives have been scaled by $1/k$ for scales $k = 1, 2, 3$

stiffness contributions are effectively nullified. Thus, stiffness is only present at the interactions of pseudo-wavelets within the same layer. Consequently, for the finer grid, the stiffness matrix forms a diagonal structure.

$$K_k^{2+2^{k-1}+j} = \frac{2^{k+1}\epsilon}{h}. \tag{25}$$

The resulting multi-scale element-level stiffness matrix is

$$K^e = \frac{\varepsilon}{h}\begin{bmatrix} 1 & -1 & & & & \\ -1 & 1 & & & & \\ & & 4 & & & \\ & & & 8 & & \\ & & & & \ddots & \\ & & & & & 2^{k+1} \end{bmatrix}. \tag{26}$$

The computation of the mass matrix is more involved, as it requires the calculation of inner products that involve the basis functions across scales. For example, when calculating the mass element for two intra-level shape functions, the result will be zero since they do not interact with each other.

When considering the intra-level shape functions, their interactions can be computed using the following formulations:

$$M_{00} = \int_{\Omega^e} \gamma N_0 N_0 d\Omega,\ M_{01} = \int_{\Omega^e} \gamma N_0 N_1 d\Omega, \cdots$$

$$M_{22} = \int_{\Omega^e} \gamma \psi_1^2 \psi_1^2 d\Omega,\ M_{23} = \int_{\Omega^e} \gamma \psi_1^2 \psi_2^3 d\Omega, \cdots, M_{ij} = \int_{\Omega^e} \gamma \psi^i \psi^j d\Omega \tag{27}$$

The overall structure of the Mass matrix has illustrated in Figure 3

The F matrix has been calculated with the help of the below equations.

$$F_0 = \int_{\Omega^e} f N_0 d\Omega,\ F_1 = \int_{\Omega^e} f N_1 d\Omega,\ F_2 = \int_{\Omega^e} f \psi_1^2 d\Omega,\ \cdots, F_i = \int_{\Omega^e} f \psi^i d\Omega \tag{28}$$

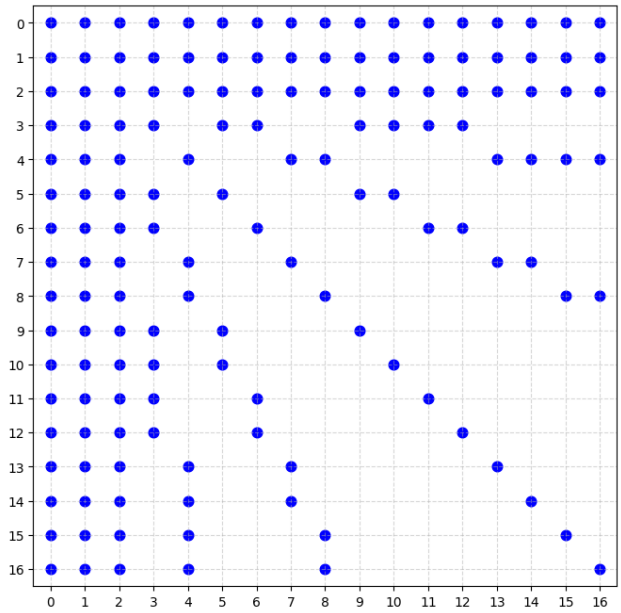

Figure 3: Finger diagonal structure of multi-scale element mass matrix showing non-zero entries

The code adopts an elemental approach for the computation and assembly of the standard coarse-grid matrices M, K, and F. However, for the finer grid, I have chosen a more nodal approach rather than an elemental one, since, unlike the coarse grid, many functions interact with each other within a particular sub-element.

Thus, the code processes the pseudo-wavelet layers iteratively, calculating the M, K, and F matrices for all interactions with the lower layer shape functions, updating these matrices accordingly. The calculations are only performed when two shape functions have a non-empty intersecting active zone. For example, when considering ψ_i and ψ_j, the code first identifies the intersecting active zone for both shape functions. This zone is then mapped into the parametric space, and numerical integration is used to compute the necessary integrals.

To determine the active zones of the shape functions, I am utilizing the formulations given in Equations (21), (22), and (23).

In standard FEA, we need to assemble the M, K, and F matrices using elemental values. However, this is not required in the case of wavelets, as they are local phenomena. Wavelet bases do not

rely on information from outside the element, so there is no need to assemble global matrices. Instead, we can solve the equations locally and directly obtain the coefficient multipliers of the wavelet shape functions, Δu_k. However, unlike traditional FEA, where each nodal value is associated with a single shape function, the wavelet basis involves multiple shape functions contributing to a single nodal value. Therefore, instead of assembling the M, K, and F matrices like in traditional FEA, we need to assemble the final solutions to obtain the desired results. These results can be computed using Equation (29).

$$\mathbf{u_k^{1+2^{k-1}+j}} = \sum_{\mathbf{i=0}}^{\mathrm{Nnpe-1}} \mathbf{N_i u_0^i} + \sum_{\mathbf{k=1}}^{\mathrm{Nscale}} \psi_{\mathbf{k}}^{\mathbf{1+2^{k-1}+j}} \mathbf{\Delta u_k^{1+2^{k-1}+j}} \tag{29}$$

The stopping condition for the wavelet-based method differs slightly from the one used in standard FEA. In this approach, we use Equation (30) to evaluate the stopping condition.

$$\epsilon_{\text{gain}} = \frac{\mathbf{\Delta u_k^{1+2^{k-1}+j}}}{||\mathbf{u_k}||} \tag{30}$$

I refer to this term as the gain, as it quantifies the amount of new information introduced by each additional layer of wavelets. The gain also indicates the extent to which the values of the latest layer's variables are influenced by the wavelets of that layer, as opposed to being a combination of contributions from previous layers. If the gain, ϵ_{gain}, falls below a predefined threshold, we can stop adding further wavelet layers.

The following algorithm can be used to compute the nodal values of the variables.

Algorithm 1: Multi-scale Solution Algorithm

1. Form the coarse-grid operators K_0, M_0, and F_0, and solve the coarse-grid problem:

$$U_0 = [K_0 + M_0]^{-1} F_0$$

2. For each element, inject one scale and solve for the wavelet coefficient, Δu.

$$\Delta u_k = [K_k + M_k]^{-1} F_k$$

3. Calculate u_k with the help of Eq. (29).

4. Calculate the gain ϵ_{gain} with Eq. (30) and compare with $\epsilon_{\text{desired}}$. If $\epsilon_{\text{gain}} \geq \epsilon_{\text{desired}}$ return to step 2 and include additional wavelet layers.

4 Numerical Example

In this section, we present the results obtained from using the wavelet basis for solving the given problem.

For this numerical example, we set $\gamma = 1$, $\epsilon = 1$, and the forcing function $f = \pi^2 sin(\pi x) + sin(\pi x)$, with Dirichlet boundary conditions $u(0) = 0$ and $u(2) = 0$. The computational domain is divided into two elements: [0,1] and [1,2] along the x-axis.

The analytical solution to the problem, derived from Equation (1), is given by:

$$u(x) = sin(\pi x)$$

.

This allows us to compare the analytical solution with the numerical results obtained using the wavelet basis. The comparison for different scales of wavelets is shown in Figure 4.

As observed, for Scale-0 (the coarse-grid solution), the result is a straight line parallel to the x-axis. This outcome occurs because no wavelet layers are applied, and solving the coarse grid leads to zero values for all three nodes. Essentially, all the information within the element is lost in this case.

However, as we increase the number of wavelet layers, the numerical solution begins to closely approximate the analytical solution. This indicates that the wavelets are effectively capturing the local information, progressively refining the solution and providing more accurate results. This behavior demonstrates that the wavelet basis is correctly gathering information locally and improving the accuracy of the numerical solution as more layers are added.

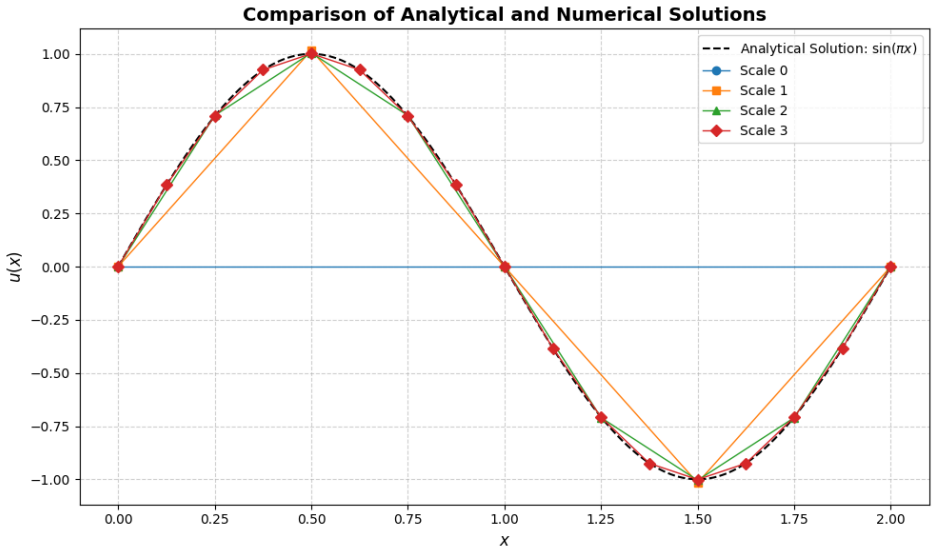

Figure 4: Analytical solution and scale solutions for k = 0, 1, 2, 3

The next aspect we will discuss is equivalent to the mesh-convergence analysis. In Figure 7, we observe how the value of ϵ_{gain} , derived from Equation (30), changes for a particular element in the given problem setup.

For the first wavelet layer, we see that ϵ_{gain}=1. This can be explained by the fact that the values of the nodal variables at this layer are entirely dependent on the new wavelet layer, as there was no prior information available for these nodes. As we continue to increase the number of scales (wavelet layers), ϵ_{gain} progressively decreases and asymptotically approaches zero.

In this particular example, we observe that ϵ_{gain} reaches near zero after the introduction of five wavelet layers. This indicates that the elemental mesh has essentially converged after five levels of wavelets. Beyond this point, adding additional layers does not significantly improve the value of ϵ_{gain}. Therefore, we can conclude that the mesh has reached convergence, and further refinement of the wavelet layers does not yield substantial benefits.

It is also noteworthy that both elements in our problem exhibit symmetry, and as a result, they follow the same convergence pattern. This symmetry ensures that the convergence behavior is consistent across both elements.

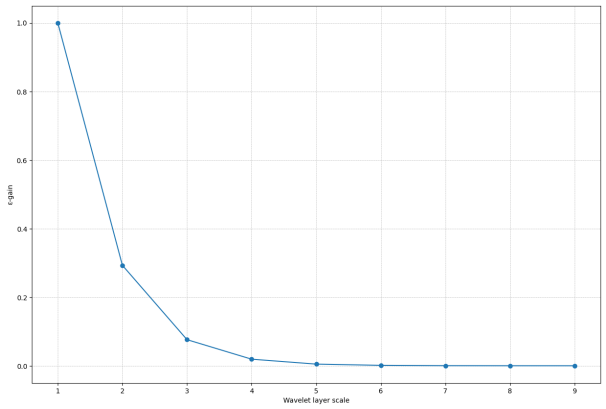

Figure 5: Linear Scale

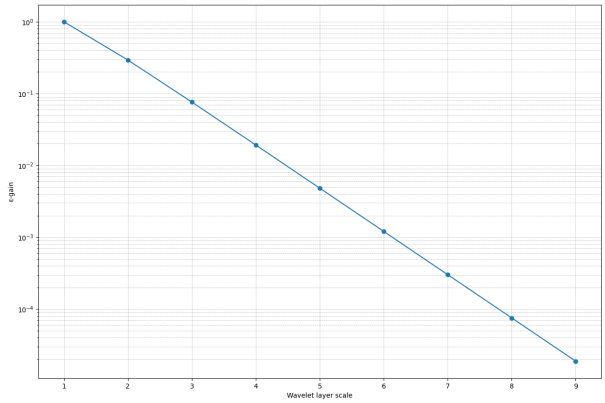

Figure 6: Logarithmic Scale

Figure 7: Change in ϵ_{gain} as we increase the Scale

Inference

Wavelets serve as a powerful tool for capturing local information with high precision. However, replacing the traditional finite element analysis (FEA) basis entirely with wavelets is impractical due to the heavy computational burden associated with the Schauder mass matrix, which is expensive to manage and invert. Instead, wavelets prove to be more effective when used in conjunction with standard FEA bases.

A practical approach is to solve the partial differential equations (PDEs) using the conventional FEA basis on a coarse grid, storing the relevant information, and then applying wavelets to obtain finer, more detailed solutions in the necessary regions. This hybrid approach is computationally viable and offers significant modularity in meshing. It allows for targeted refinement without the need for global refinement, thus enhancing both efficiency and flexibility.

An important observation is that, as the number of wavelet layers increases, the elemental mass and stiffness matrices grow exponentially. Therefore, it is crucial to understand the relationship between the number of wavelet layers and the gain, ϵ_{gain}, as this will inform decisions about how much refinement is necessary. From our analysis, we observe that ϵ_{gain} follows an almost linear curve with respect to the number of wavelet layers. This trend suggests that as the number of wavelet shape functions increases, the change in ϵ_{gain} becomes linear, allowing us to predict and control the convergence behavior more effectively.

Moreover, this behavior highlights the diminishing returns of additional wavelet layers beyond a certain point, suggesting that an optimal number of layers can be determined for a given problem. This leads to more efficient computation by avoiding unnecessary complexity once convergence is achieved.

Overall, the integration of wavelets with traditional FEA methods not only enhances the solution's accuracy but also offers a flexible, computationally efficient way to address complex problems, especially when local refinement is required.

5 Conclusions

This study demonstrates that wavelets can significantly improve the accuracy of finite element solutions by providing a mechanism for local refinement without the need for global mesh refinement. While replacing the traditional FEA basis entirely with wavelets is computationally expensive due

to the complexity of mass and stiffness matrix computations, combining wavelets with FEA offers a viable and efficient solution. By solving the PDEs on a coarse grid with standard FEA and applying wavelets to refine the solution locally, we achieve a detailed representation of the solution in the necessary regions while maintaining computational efficiency.

Our analysis of the convergence behavior, particularly through the evaluation of ϵ_{gain} , highlights that the benefits of additional wavelet layers diminish after a certain number of layers. This asymptotic behavior allows us to determine an optimal level of wavelet refinement, preventing unnecessary computational complexity. The linear relationship observed between the number of wavelet layers and ϵ_{gain} further strengthens the predictability and control over the solution process.

In conclusion, the wavelet-FEA hybrid approach offers a promising avenue for solving complex PDEs, providing better resolution where needed without excessive computational overhead. Future work may focus on optimizing the integration of wavelets with adaptive meshing techniques and extending the method to higher-dimensional problems for broader applications in engineering and physics.

References

[1] Mark Christon, Timothy Trucano, Joe Weatherby, Roy Baty, S. Burns, David Roach, David Womble, and Thomas Voth. An investigation of wavelet bases for grid-based multi-scale simulations. 06 1999.

[2] Mark A Christon and DW Roach. The numerical performance of wavelets for pdes: the multi-scale finite element. *Computational Mechanics*, 25(2):230–244, 2000.