An Operator-Customized Wavelet-Finite Element Approach for the Adaptive Solution of Second-Order Partial Differential Equations on Unstructured Meshes

by

Stefan F. D'Heedene

Burgerlijk Ingenieur Architect
Katholieke Universiteit Leuven, 2000

SUBMITTED TO THE
DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN THE FIELD OF COMPUTATIONAL ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2005

Signature of Author ....
Department of Civil and Environmental Engineering
January 12, 2005

Certified by ....
Kevin Amaratunga
Associate Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by ....
Andrew J. Whittle
Chairman, Departmental Committee on Graduate Students

An Operator-Customized Wavelet-Finite Element Approach for the Adaptive Solution of Second-Order Partial Differential Equations on Unstructured Meshes

by

Stefan F. D'Heedene

Submitted to the Department of Civil and Environmental Engineering on January 14, 2005 in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in the Field of Computational Engineering

ABSTRACT

The *Finite Element Method* (FEM) is a widely popular method for the numerical solution of *Partial Differential Equations* (PDE), on multi-dimensional unstructured meshes. Lagrangian finite elements, which preserve C^0 continuity with interpolating piecewise-polynomial shape functions, are a common choice for second-order PDEs. Conventional single-scale methods often have difficulty in efficiently capturing fine-scale behavior (e.g. singularities or transients), without resorting to a prohibitively large number of variables. This can be done more effectively with a multi-scale method, such as the *Hierarchical Basis* (HB) method. However, the HB FEM generally yields a multi-resolution stiffness matrix that is coupled across scales.

We propose a powerful generalization of the Hierarchical Basis: a second-generation wavelet basis, spanning a Lagrangian finite element space of any given polynomial order. Unlike first-generation wavelets, second-generation wavelets can be constructed on any multi-dimensional unstructured mesh. Instead of limiting ourselves to the choice of primitive wavelets, effectively HB detail functions, we can tailor the wavelets to gain additional qualities.

In particular, we propose to customize our wavelets to the problem's operator. For any given linear elliptic second-order PDE, and within a Lagrangian FE space of any given order, we can construct a basis of compactly supported wavelets that are orthogonal to the coarser basis functions with respect to the weak form of the PDE. We expose the connection between the wavelet's vanishing moment properties and the requirements for operator-orthogonality in multiple dimensions. We give examples in which we successfully eliminate all scale-coupling in the problem's multi-resolution stiffness matrix. Consequently, details can be added locally to a coarser solution without having to re-compute the coarser solution.

This quality can be exploited in the adaptive solution of a wide range of problems. By using an adaptive operator-customized wavelet basis, we achieve an optimal solution speed for problems with concentrated local errors. We illustrate this with the computation of a two-dimensional Green's Function on a bounded domain. We also apply our adaptive solution technique to speed up barrier option valuation, governed by a multi-dimensional diffusion-convection-reaction PDE with varying coefficients.

Thesis Supervisor: Kevin Amaratunga
Title: Associate Professor of Civil and Environmental Engineering

Acknowledgements

I would like to thank the following individuals and institutions for their support:

My research advisor, Professor Kevin Amaratunga, for his guiding and friendly advice, supporting commitment, and trust.

My former and current fellow students, especially Julio Castrillón-Candás and Ragunathan Sudarshan, for the enlightening discussions on research and life.

The Belgian American Educational Foundation, for supporting me with a fellowship.

The National Science Foundation, for supporting this work under Grant No. 9984619.

The Department of Civil and Environmental Engineering at MIT, for their generous commitment to my education.

My friends and roommates, for their encouragement and understanding.

My family, for their loving trust.

To my mother

Biographical Note

Stefan D'Heedene grew up in Belgium, where he obtained the degree of Burgerlijk Ingenieur Architect at the Katholieke Universiteit Leuven *Magna Cum Laude* in June 2000. He started his doctoral studies at the Massachusetts Institute of Technology in September 2000 with a *Victor De Corte* Fellowship. In 2001, he was awarded the *H. Van Waeyenbergh of the Hoover Foundation for the Development of the University of Leuven* Fellowship by the Belgian American Educational Foundation. In 2004, he received a *Financial Technology Option* Certificate from the Massachusetts Institute of Technology.

Contents

Tables

Figures

1. Introduction

1.1 Background

The *Finite Element Method* (FEM) (e.g. Bathe, 1996 or Zienkiewicz *et al.*, 2000) is a widely popular method for the numerical solution of problems described by *Partial Differential Equations* (PDE) over complicated multi-dimensional geometries. With the growth in computational power and storage capacity, FE models have become increasingly large-scale. In particular, problems that exhibit behavior over a range of scales may be better handled by a multi-scale method than by a simple single-scale method. We have in mind problems with geometrical anomalies (e.g. holes), material anomalies (e.g. boundary layers), or detailed features in the loads or initial condition (e.g. Green's function, wave front). For such problems, the mesh resolution can be increased adaptively, only where needed. Hence, a given solution accuracy can be obtained with a reduced problem size. In addition, multi-resolution methods, such as the multi-grid method, can improve a FE system's iterative solving speed.

In the nineties, a more flexible multi-resolution technique, the *Hierarchical Basis* (HB) FEM, has been proposed (Yserentant, 1992) as an alternative to the multi-grid pre-conditioner. This method in essence consists of a change from the usual single-scale

FEM basis to a multi-resolution basis of HB functions that span the same space. More recently, these Hierarchical Basis functions have been proposed for adaptive refinement methods (Krysl *et al.*, 2003). Indeed, whereas other adaptive mesh refinement methods require either re-meshing or the resolution of hanging nodes, the HB method, by contrast, performs mesh refinement in a natural way. When adding detail functions to a coarser basis, we do not need to change the stiffness matrix of the coarser problem, but can just plug-in the sub-matrix corresponding to the new detail functions. However, new details generally cannot be added to a coarser solution without re-computing the entire solution. Indeed, in general, the HB FEM stiffness matrix is fully coupled across scales. Achieving decoupling between the detail part and the coarser part of the multi-resolution stiffness matrix is the primary goal of this dissertation. Scale-decoupling will greatly facilitate adaptive refinements. In addition, scale-decoupling will yield an optimal solution speed for problems with high local concentration of the solution error.

Parallel to the development of Hierarchical Bases, the use of wavelet functions in PDE simulations has been proposed (see e.g. Amaratunga *et al.*, 1993, 1994, 1997, Beylkin *et al.*, 1992, Dahlke *et al.*, 1993), because wavelets can lead to fast, hierarchical and locally adaptive algorithms. However, their application in FEM analysis was hindered by the 'signal processing' nature of traditional wavelet constructions (see e.g. Daubechies, 1988, Mallat, 1988, Meyer, 1985, Strang *et al.*, 1996). Indeed, traditional wavelets consist of scaled and shifted versions of a single function on a regularly spaced one-dimensional grid over a theoretically unbounded domain. Therefore, they cannot be constructed on meshes commonly encountered in FEM analysis. This major restriction on wavelet theory has been eliminated by the discovery of the *lifting scheme* (Sweldens, 1996), and *stable completion* (Carnicer *et al.*, 1996). These new construction methods have led to a generalization of traditional wavelets to the wider class of *second-generation wavelets,* which can be built on irregularly spaced, unstructured, multi-dimensional meshes over bounded domains.

We now can look at the Hierarchical Bases from a wavelet perspective. The generalization of the multi-wavelet concept to second-generation wavelets has led to the construction of a family of Lagrangian wavelet bases (Strang *et al.,* 1995, and Castrillón-Candás *et al.,* 2001). They are piecewise polynomial of any given order, and flexible to build on irregularly spaced, unstructured, multi-dimensional meshes over bounded domains. They span the same space as single-scale Lagrangian finite elements, commonly used for the analysis of second-order PDEs. In their simplest form these wavelets correspond to traditional HB functions. However, we can customize these wavelets to generate additional qualities for our multi-resolution basis.

In other research, wavelets (detail functions) have been customized to be orthogonal to all scaling functions (regular shape functions), with the intention of stabilizing the multi-resolution basis. Such orthogonality is not a natural quality of traditional HB functions. In many cases, each of these *orthogonal* wavelets had support all over the domain, albeit decaying fast enough to enable a local approximation (e.g. Vassilevski *et al.*, 1997). In other proposals each wavelet was in effect compact (Strang *et al.*, 1996, p.257, or Dahmen *et al.*, 1999). Wavelets that are orthogonal to scaling functions, or even feature additional vanishing moments, have been proposed for applications ranging from system matrix compression based on operator smoothness to system pre-conditioning. However, such orthogonal wavelets in general do not generate full scale-decoupling in the stiffness matrix of a second-order PDE.

Our intent is to facilitate adaptive refinement schemes for large-scale problems with local features. For this, we desire a full decoupling between the detail parts and the coarser part of the multi-resolution stiffness matrix. At the same time, we would like to keep the wavelet functions compactly supported. Indeed, if we achieve scale-decoupling with compact wavelets (detail functions), cheaply computed details may be added locally to a coarse solution without having to re-compute the coarse solution. We achieve such scale-decoupling if and only if our wavelets are made *operator-orthogonal* to all scaling functions, with respect to the weak form of the PDE. In general, traditional Hierarchical

Basis functions do not scale-decouple the stiffness matrix, except for the first-order HB applied to a trivial one-dimensional Poisson's Equation. We will focus specifically on problems described by linear elliptic second-order PDEs. Indeed, adaptive refinement becomes much more challenging for nonlinear problems. Note that the *orthogonality* described in the previous paragraph is in fact a special case of *operator-orthogonality.* Indeed, it corresponds to operator-orthogonality with respect to the identity operator.

Other researchers have proposed the construction of an operator-orthogonal wavelet basis. Jawerth and Sweldens derived a basis of one-dimensional compact wavelets that are operator-orthogonal with respect to non-trivial second-order elliptic operators (Jawerth *et al.*, 1993). However, their basis does not span a Lagrangian finite element space, and their method is not readily extendible to higher-dimensional problems. Dahlke and Weinreich proposed the construction of one- and two-dimensional wavelets operator-orthogonal with respect to non-trivial second-order elliptic operators (Dahlke *et al.*, 1994). However, they used a basis of first-generation wavelets, restricted to regular grids over unbounded domains.

We will propose a method to customize Lagrangian FE wavelets – on irregular, unstructured meshes over bounded domains – such that they are compact and operator-orthogonal with respect to *any* linear elliptic second-order operator of our choosing. Then, we will apply this method to exploit scale-decoupling in one- and two-dimensional adaptive refinement applications.

1.2 Outline

The following chapter, Chapter 2, discusses the benefits and limitations of the Hierarchical Basis FEM. Hierarchical Basis functions handle adaptive refinements in a natural manner (Krysl, 2002), without hanging-node issues. In addition, it is well-known that the stiffness matrix for the one-dimensional Laplace operator, using a first-order Lagrangian HB, is scale-decoupled, even entirely diagonal. This greatly facilitates adaptive refinement. However, for any other second-order operator, for higher-order bases, as well as for two-dimensional problems, the HB stiffness matrix is fully coupled across scales.

Next, in Chapter 3, we present second-generation wavelet theory and a wavelet framework for the FEM. We apply the multi-wavelet idea to second-generation wavelets (Strang *et al.,* 1995, and Castrillón-Candás *et al.,* 2001), and build a wavelet framework for Lagrangian finite element basis functions of any given order, on unstructured, irregular, one-dimensional or two-dimensional (triangular) meshes. This inexpensive wavelet construction method is based on the lifting scheme (Sweldens, 1996) and stable completion (Carnicer *et al.*, 1996). In their simplest form, these wavelets correspond to traditional HB functions. However, we have the control to tailor the wavelet functions to our needs.

Then, we will use this framework to customize wavelets to any given second-order operator. We will cover all one-dimensional operators in Chapter 4, and all two-dimensional operators in Chapter 5. In particular, we propose wavelets (i.e. detail functions) that are orthogonal to the scaling functions (i.e. coarse basis functions) with respect to the bilinear form induced by the operator, or *operator-orthogonal* in short. We will study the influence of operator type on the compactness of customized wavelets. Based on this analysis, we will propose implementation schemes that can handle different operator types and accommodate any unstructured mesh.

In Chapter 6, we analyze the complexity of our customized wavelet method, illustrated by a two-dimensional Green's function example. For problems with a high local concentration in the solution error, such as our example, we achieve an optimal solution cost of $O(J)$, where J is the number of levels of refinement. This clearly outperforms the Hierarchical Basis method.

In Chapter 7, we subsequently apply our customized wavelet method to a barrier option pricing problem, to show the generality and effectiveness of our approach. This dynamic problem is governed in the spatial domain by a one-dimensional or two-dimensional diffusion-convection-reaction PDE with varying coefficients. The barrier feature causes a local concentration in the solution error that can be effectively exploited by a highly adaptive method. We assume that finer solution details cannot pop-up in a zone with no significant coarser detail coefficients, i.e. that details will be nested over consecutive levels of refinement. Then, we can on each level use the details of a coarser scale to determine where to compute finer details.

Finally, we conclude with research extensions in Chapter 8. We briefly discuss the expected benefits of applying our method to 3D applications. We also mention the extension of our wavelet framework and customization for Lagrangian finite elements solving second-order PDEs, to the Hermite finite elements solving fourth-order problems.

2. Hierarchical Basis FEM

2.1 HB Refinement

In this section, we demonstrate the benefits and limitations of *Hierarchical Basis* (HB) adaptive refinement. Let us consider the following example problem: the Poisson's Equation over the one-dimensional domain $[0,1]$, subject to homogeneous Dirichlet boundary conditions:

$$-\frac{\partial^2 u(x)}{\partial x^2} = f(x) \quad \left| \begin{array}{l} u(0)=0 \\ u(1)=0 \end{array} \right. \tag{2.1}$$

The forcing function on the right-hand-side has a discontinuity and is plotted in Figure 2-1. We can choose to numerically solve this *Partial Differential Equation* (PDE), with a *Finite Element Method* (FEM) (see e.g. Bathe, 1996). For example, with a mesh of eight linear Lagrangian elements, shown in Figure 2-2, we find a solution plotted in Figure 2-3. Note that Lagrangian finite elements support interpolating piecewise polynomial shape functions of a given order, guaranteeing C^0 continuity over nodes that connect different elements. They are a popular choice for second-order PDEs. If we now desire higher

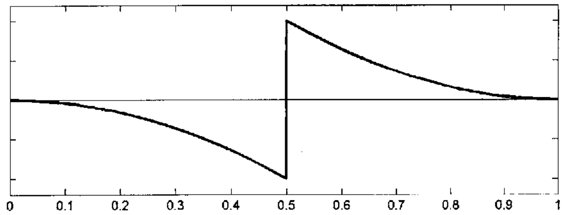


Figure 2-1: Forcing function used in Equation (2.1).

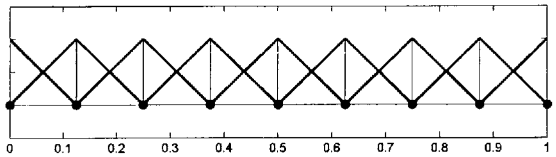


Figure 2-2: Eight Lagrangian finite elements of first order.

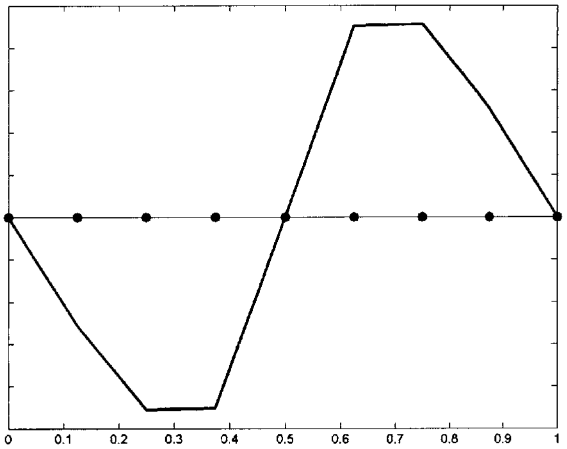


Figure 2-3: FE solution of Equation (2.1) on the grid shown in Figure 2.2.

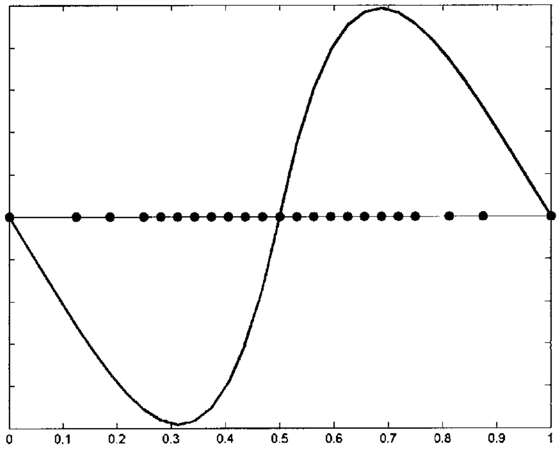


Figure 2-4: FE solution of Equation (2.1) on an adaptively refined grid.

solution accuracy, a common option is to solve the problem on a mesh with a higher resolution, a refined mesh. As shown in Figure 2-4, the problem's new size can be reduced by only increasing the mesh's resolution locally where needed. Indeed, the solution error is much higher close to the forcing discontinuity than near the boundary. We refer to this selective refinement as *adaptive refinement*, or *non-uniform refinement*. Such refinement can produce a solution with the same degree of accuracy as the solution on a uniformly refined mesh, while keeping the problem size – and hence the solution cost – low.

Adaptive refinements are most commonly achieved by h-refinement, or element-refinement. Also p-refinement can be used, though this is often more difficult (see Zienkiewicz *et al.*, 2000). When desiring one more degree of freedom for the mesh of four elements in Figure 2-5, h-refinement replaces an element of the coarser mesh by two finer elements. This not only adds one degree of freedom to the FE stiffness matrix, but

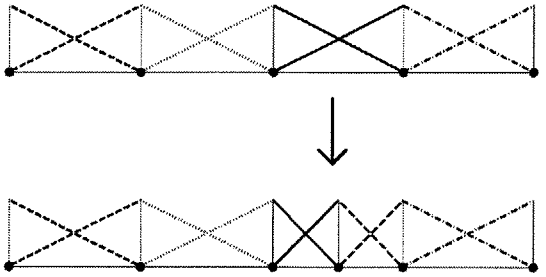


Figure 2-5: Adaptive h-refinement of a one-dimensional first-order Lagrangian FE basis.

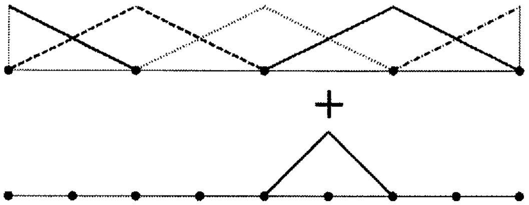


Figure 2-6: Adaptive refinement of a one-dimensional first-order Hierarchical Basis.

also changes the matrix entries for the existing degrees of freedom that form the connection between the new and the old mesh. The same goes for the right-hand-side vector. Moreover, if we re-arrange the stiffness matrix to place the new degree of freedom at the bottom right, we can assess that the matrix is coupled between the part corresponding to the coarser mesh and the new part:

$$\mathbf{A}_{\text{coarse}}\mathbf{u}_{\text{coarse}} = \mathbf{f}_{\text{coarse}}$$
$$\rightarrow \begin{bmatrix} \overline{\mathbf{A}}_{\text{coarse}} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{A}_{\text{detail}} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{u}}_{\text{coarse}} \\ \mathbf{u}_{\text{detail}} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{f}}_{\text{coarse}} \\ \mathbf{f}_{\text{detail}} \end{bmatrix} \tag{2.2}$$

In Equation (2.2), the coupling term $\mathbf{C} \neq \mathbf{0}$, and the bars above entries denote a changed value relative to the coarser system above. This means that in order to add this new detail to the problem, we expect to re-compute the entire solution.

However, let us now view the four elements as five basis functions instead, shown in Figure 2-6. Each basis function is built up of shape functions and is an interpolating piecewise polynomial associated with one degree of freedom. Evidently, this is merely a different perspective on the same FEM and yields the same solution. When we want to refine this basis, we can keep the coarser basis functions, and throw in a new finer basis function, associated with the new degree of freedom. Such refinement is referred to as Hierarchical Basis (HB) refinement. It has been recently proposed as a natural refinement method (Krysl *et al.*, 2002), based on the earlier groundbreaking work on the Hierarchical Basis FEM (Yserentant, 1992). Because we keep the coarser basis functions in our basis without alterations, we do not need to update the stiffness matrix entries for

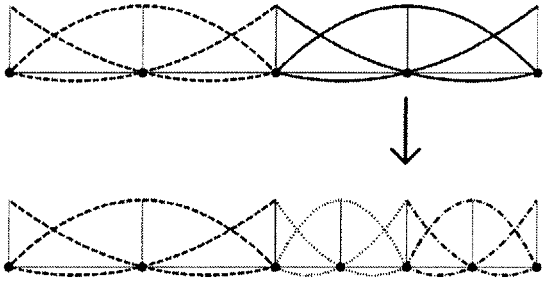


Figure 2-7: Adaptive h-refinement of a one-dimensional second-order Lagrangian FE basis.

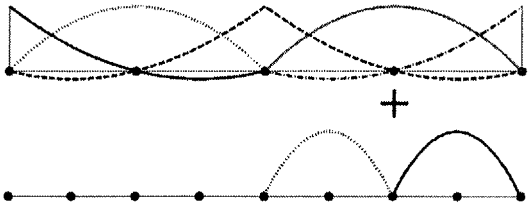


Figure 2-8: Adaptive refinement of a one-dimensional second-order Hierarchical Basis.

the existing degrees of freedom. In addition, for this particular problem, we find that there is no coupling between the part corresponding to the coarser mesh and the new part:

$$\mathbf{A}_{\text{coarse}}\mathbf{u}_{\text{coarse}} = \mathbf{f}_{\text{coarse}}$$
$$\rightarrow \begin{bmatrix} \mathbf{A}_{\text{coarse}} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{\text{detail}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\text{coarse}} \\ \mathbf{d}_{\text{detail}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\text{coarse}} \\ \mathbf{g}_{\text{detail}} \end{bmatrix}$$

We can check that for linear HB functions:

$$\int \frac{\partial \varphi_{\text{detail}}}{\partial x} \frac{\partial \varphi_{\text{coarse}}}{\partial x} dx = 0$$

The advantage of scale-decoupling is that we can add new details to the problem, without having to re-compute the coarser solution. This can save much work for problems requiring a high degree of adaptivity, where the size of the sub-problem associated with new details can be significantly smaller than the size of the coarser problem. This scale-decoupling property is a well-known quality of the linear HB functions in combination with the one-dimensional Poisson's Equation. However, such decoupling between the coarse and detail part is not the case for HB refinements in general.

Indeed, let us revisit the one-dimensional Poisson's Equation described by Equation (2.1), but now use quadratic (second-order Lagrangian) finite elements. Also for this FE problem, adaptive refinements can increase the solution accuracy while limiting the problem size. Again, as illustrated in Figures 2-7 and 2-8, we can avoid updates to the coarser part of the stiffness matrix, by using HB refinement. With h-refinement, we would replace one element by two new elements, thus introducing two new degrees of freedom. If we view the two elements, instead, as a basis of five basis functions (of two distinct types), we can achieve the same refinement, by adding in two HB detail functions. As in the case of a linear basis, the refinement basis functions are finer interpolating piecewise polynomial basis functions of the same order. They are shown in Figure 2-8. However, for the choice of a quadratic basis and a one-dimensional Poisson's Equation, we do not have scale-decoupling in the stiffness matrix. Indeed, we find:

$$\begin{aligned} &\mathbf{A}_{\text{coarse}}\mathbf{u}_{\text{coarse}} = \mathbf{f}_{\text{coarse}} \\ \rightarrow &\begin{bmatrix} \mathbf{A}_{\text{coarse}} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{A}_{\text{detail}} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{u}}_{\text{coarse}} \\ \mathbf{d}_{\text{detail}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\text{coarse}} \\ \mathbf{g}_{\text{detail}} \end{bmatrix} \end{aligned}$$

With $\mathbf{C} \neq \mathbf{0}$, because for quadratic HB functions, we have in general:

$$\int \frac{\partial \varphi_{\text{detail}}}{\partial x} \frac{\partial \varphi_{\text{coarse}}}{\partial x} dx \neq 0$$

Thus, in order to add these new details to the problem, we are forced to re-compute the entire solution.

Next, we consider a problem, different from Poisson's Equation: a Helmholtz Equation over the one-dimensional domain $[0,1]$, subject to homogeneous Dirichlet boundary conditions:

$$-\frac{\partial^2 u(x)}{\partial x^2} + Ku(x) = f(x) \quad \left| \begin{aligned} u(0) &= 0 \\ u(1) &= 0 \end{aligned} \right.$$

The forcing function $f(x)$ remains as plotted in Figure 2-1. We also use again a first-order Lagrangian FEM with linear shape functions. As in the case for Poisson's Equation, we do not have to update the entries of the coarser part of the stiffness matrix, if we use the linear HB refinements discussed above. However, unlike the Poisson case, we have now full coupling between the coarse part and the new detail part of the stiffness matrix. Indeed, we find:

$$\int \frac{\partial \varphi_{\text{detail}}}{\partial x} \frac{\partial \varphi_{\text{coarse}}}{\partial x} dx + K \int \varphi_{\text{detail}} \varphi_{\text{coarse}} dx \neq 0$$

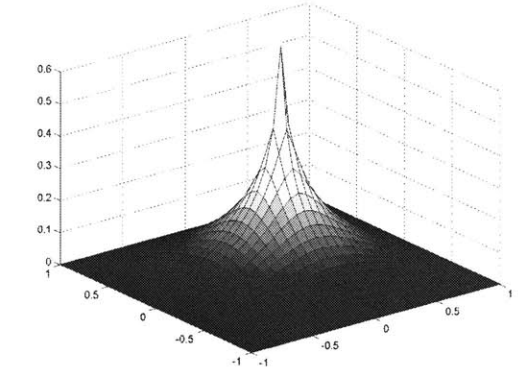


Figure 2-9: FE solution of Equation (2.3) with first-order Lagrangian triangular elements.

Thus, we have to re-compute the entire solution, when adding refinements.

Finally, let us consider the Poisson's Equation over a two-dimensional domain, subject to homogeneous Dirichlet boundary conditions, and a delta function as forcing function:

$$-\nabla^2 u(x,y) = \delta_{x-x_c,y-y_c} \qquad u(x,y)\big|_\Gamma = 0 \tag{2.3}$$

Note that the solution of this problem is the Green's function for the Laplace operator on a bounded domain. We solve this problem with a first-order Lagrangian FEM on a triangular mesh. Each element has three linear interpolating shape functions: one per degree of freedom. In view of the FE solution plotted in Figure 2-9, it again makes sense to increase the density of the mesh only locally around the delta, where the solution error is concentrated. Note that we will graphically demonstrate adaptive refinement on a mesh of equilateral elements, but the discussion is applicable to any triangular FE mesh. To add with h-refinement one degree of freedom to the elements shown in Figure 2-10, we replace one element by four elements. We thereby introduce five new degrees of freedom, four of which are *hanging nodes* that require extra conditions to be coupled to the neighboring coarser degrees of freedom. Thus, we need to replace parts of the coarser stiffness matrix as well as introduce cumbersome additional equations. If we view the ten linear elements of Figure 2-10 instead as ten piecewise linear basis functions, each associated with a degree of freedom, refinement becomes much more natural. Indeed, as shown in Figure 2-11, we can add that additional degree of freedom

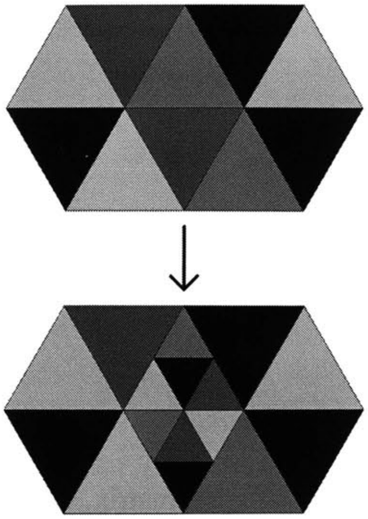


Figure 2-10: Adaptive h-refinement of a two-dimensional first-order Lagrangian FE basis.

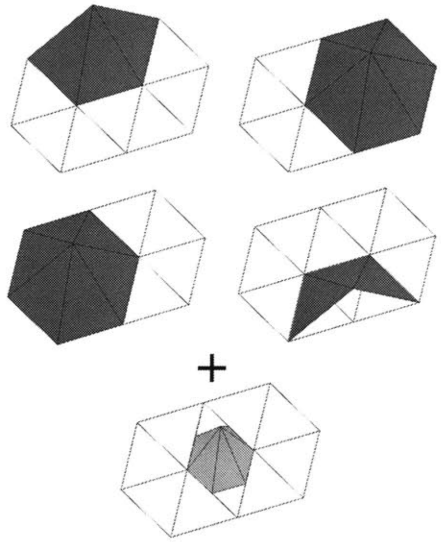


Figure 2-11: Adaptive refinement of a two-dimensional first-order Hierarchical Basis.

by just adding one finer HB basis function. As we have stated before, with HB refinement we do not need to change the coarser part of the stiffness matrix. However, we do have coupling between the coarser part of the stiffness matrix and the detail part. This is because for the two-dimensional Poisson's Equation with linear HB we have:

$$\iint\left(\frac{\partial\varphi_{\text{detail}}}{\partial x}\frac{\partial\varphi_{\text{coarse}}}{\partial x}+\frac{\partial\varphi_{\text{detail}}}{\partial y}\frac{\partial\varphi_{\text{coarse}}}{\partial y}\right)dxdy \neq 0$$

Thus, though HB refinements are natural and avoid cumbersome implementation issues, in general we do not have the additional quality of scale-decoupling.

2.2 Hierarchical Basis FEM

We now can see that the multi-resolution basis, shown in Figure 2-12, spans exactly the same function space as a finest resolution single-scale FEM basis. We can construct this full Hierarchical Basis by adding uniform HB refinements to a coarser basis, uniformly and over multiple levels of refinement. Because they span the same space, substituting the single-scale basis functions by the Hierarchical Basis functions for the chosen trial and test functions of the FEM, will not change the FE solution. As a consequence, the HB FEM inherits single-scale FEM properties, such as rate of

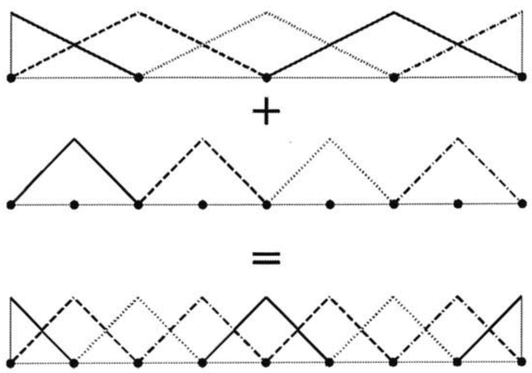


Figure 2-12: A first-order Hierarchical Basis.

convergence. Changing basis does, however, transform the stiffness matrix, forcing vector and solution vector to a multi-resolution format. Not only does this format yield more natural adaptive refinement methods (see section 2.1), it also can improve the matrix' properties (see section 2.3).

As shown in the previous section, a Hierarchical Basis can be constructed for a one- or multi-dimensional Lagrangian finite element space of any given order. The guiding principle is to keep coarser basis functions unchanged, and to add for every new degree of freedom a finer version of the same basis functions, as if we were building a single-scale FE basis on the high-resolution mesh.

The Hierarchical Basis is truly *hierarchical*. Indeed, on every point in the domain the presence of a certain scale basis function will guarantee the presence of all coarser basis functions. However, the number of levels of refinement attained does not have to be uniform over the domain (although it has to be uniform over each element). Thus, while the coarse basis covers the full domain, the details could be distributed adaptively.

Although the HB method was initially presented for regularly spaced meshes – using midpoint refinement –, the method is broad enough to cover any refined mesh. The most important restriction for a Hierarchical Basis mesh is that all finer mesh points have to lie on the straight line connecting neighboring coarser mesh points. This requirement will automatically be satisfied for a *subdivision mesh*. Such mesh can be generated by subdividing a coarse mesh, for each new refinement level adding a new degree of freedom between each pair of neighboring degrees of freedom. Meshes can be *irregular*, which means that the spacing between all nodes is not of a fixed size, and that also refinement does not have to be done mid-point. Two-dimensional meshes do not have to be *structured* either. Indeed, the number of edges connected to a node (i.e. the node valence) of the coarsest level can be different than six.

2.3 HB Pre-Conditioner

The Hierarchical Basis FEM has originally been proposed to serve as an effective pre-conditioner to the single-scale FEM problem (Yserentant, 1992). Indeed, the HB stiffness matrix has a significantly lower condition number than the equivalent single-scale stiffness matrix. Note that the HB method is similar in spirit to the popular multi-grid method, but more flexible to implement. A low condition number reduces the number of iterations needed to converge to a solution with an iterative method, such as the Conjugate Gradient method. We will discuss this in more detail in chapter 6, where we will compare the complexity of the Hierarchical Basis FEM with our proposed wavelet method.

2.4 Conclusion

The Hierarchical Basis perspective on the FEM yields several advantages. A HB framework can handle adaptive refinements in a natural and simple manner, without cumbersome implementation issues such as hanging nodes. For the one-dimensional Poisson's Equation, the use of a linear HB results in a completely decoupled system. However, in general, the HB FEM stiffness matrix is coupled between parts of different resolution. Consequently, a coarser solution has to be re-computed when new details are added. Furthermore, the Hierarchical Basis multi-resolution format pre-conditions the FEM stiffness matrix, such that the problem can be solved much faster with an iterative method.

To overcome the limitations of the traditional Hierarchical Basis, while preserving its flexibility and effectiveness in refinement, we propose a powerful generalization of the HB FEM, based on a second-generation wavelet framework.

3. Wavelet Framework

3.1 Generalized HB: Wavelet Basis

To overcome the limitations of the Hierarchical Basis FEM, we will consider HB from the broader perspective of wavelet theory.

The $H^1 \subset C^0$ function f, e.g. the FEM solution of a second-order PDE, can be projected onto a finite element subspace, spanned by single-scale basis functions of a specific resolution (see Figure 3-1). Because these compactly supported single-scale basis functions can fully represent the function's projection f_j on a specific scale j, we call them *scaling functions*. Scaling functions are denoted by the symbol $\phi_{j,k}$, associated with a specific scale, or level of resolution, j, and each corresponding with a degree of freedom, or node, k. They are shown in Figure 3-2, in which every round point represents a k-node. The basis functions discussed in Chapter 2 – spanning a Lagrangian finite element space of any given order and spatial dimension – are scaling functions. However, note that not all imaginable single-scale basis functions are acceptable as scaling functions in a wavelet theory context. Specifically, scaling functions have to satisfy a Scaling Equation – a refinement equation that relates any coarser scaling function to finer scaling functions. The Scaling Equation guarantees that a coarse

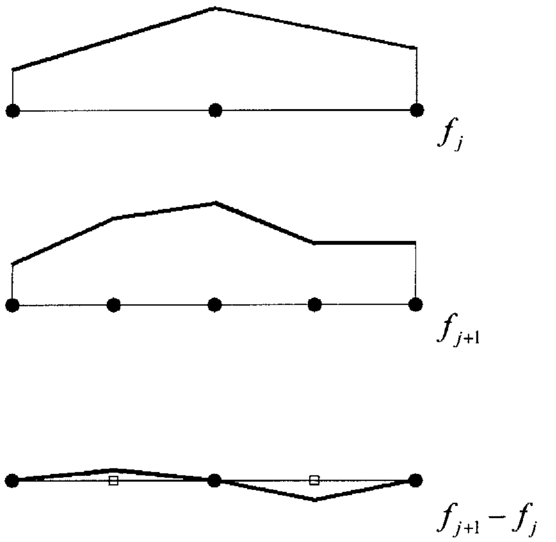


Figure 3-1: Representation of function f on two different scales; difference of the two.

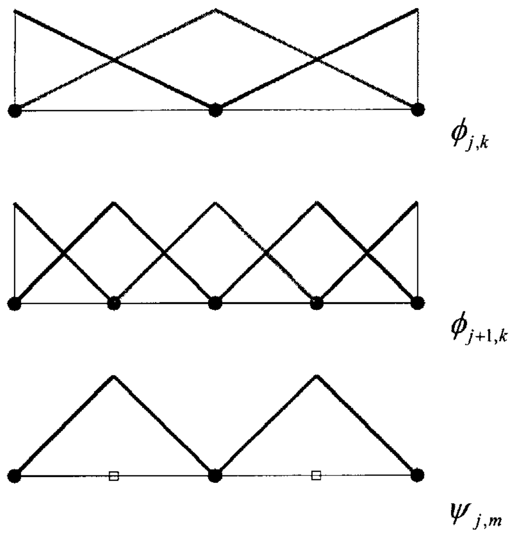


Figure 3-2: Scaling functions for two different scales; wavelet functions.

solution can also be represented by basis functions of a higher resolution. We will give this relation for the Lagrangian finite element space in section 3.2. Now consider the projection f_j of this function f on level j scaling functions, and the finer projection f_{j+1} of the same function f on scaling functions of level $j+1$:

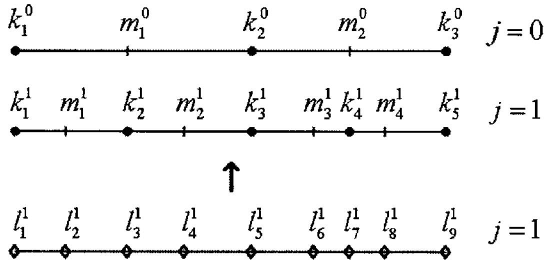
$$f_j = \sum_k \lambda_{j,k} \phi_{j,k}$$

$$f_{j+1} = \sum_k \lambda_{j+1,k} \phi_{j+1,k}$$

The projection coefficients $\lambda_{j,k}$ and $\lambda_{j+1,k}$ are called *scaling coefficients*. The difference between these two representations is given by:

$$f_{j+1} - f_j = \sum_m \gamma_{j,m} \psi_{j,m}$$

This difference space is spanned by a basis of compactly supported detail functions, which we may call *wavelet functions*. We denote them by the symbol $\psi_{j,m}$. They are associated with a specific scale, or level of resolution, j, and they correspond each to a degree of freedom, or node, m. The m-nodes are represented in Figure 3-2 by small squares, whereas a round point indicates a k-node. The projection coefficients $\gamma_{j,m}$ are called *wavelet coefficients*. The wavelets can be chosen to be simple finer scaling functions sitting on the m-nodes. Thus, as a primitive choice, we have $\psi_{j,m} = \phi_{j+1,m}$.


k_1^0 m_1^0 k_2^0 m_2^0 k_3^0 $j=0$
k_1^1 m_1^1 k_2^1 m_2^1 k_3^1 m_3^1 k_4^1 m_4^1 k_5^1 $j=1$
l_1^1 l_2^1 l_3^1 l_4^1 l_5^1 l_6^1 l_7^1 l_8^1 l_9^1 $j=1$

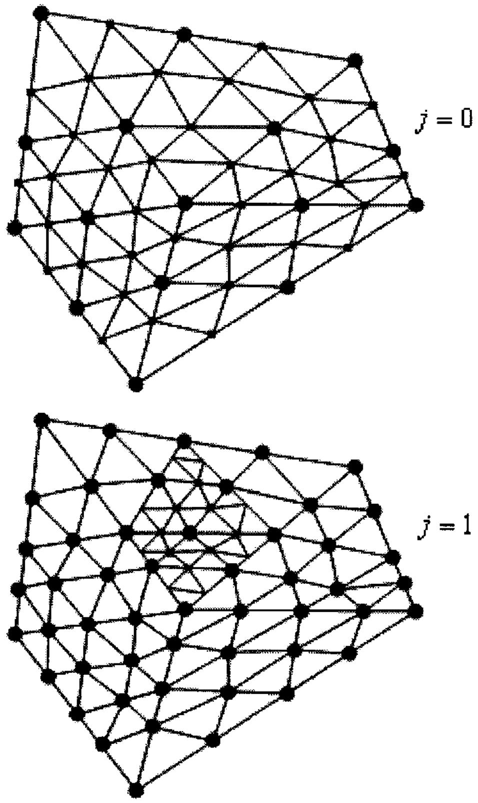


Figure 3-3: Partitioning of one-dimensional mesh into k- and m-nodes of levels 1 and 0.

Figure 3-4: Level 0 and level 1 partitions of adaptively refined two-dimensional mesh.

Indeed, we can verify that these detail functions span the difference between f_{j+1} and f_j. If we have scaling functions of the Lagrangian finite element space, such a wavelet choice results in a traditional Hierarchical Basis, on an irregularly spaced, unstructured mesh. However, we will show further in the Wavelet Equation of section 3.2 that this is not the only possible wavelet choice. Indeed, we will be able to construct wavelet functions customized to our needs.

The mesh supporting a wavelet basis can be obtained by splitting up a fine single-scale mesh in partitions of different levels, according to a spatial hierarchy. Examples are given in Figures 3-3 and 3-4, where both meshes support quadratic basis functions. The partitions are nested, which means that a partition of a certain level must contain all nodes of coarser level partitions. Levels will be numbered by j, with level 0 being the coarsest level partition possible on the grid. Per level j, nodes are denoted by the variable l. This set has two subsets: the nodes that are part of the partition at coarser level

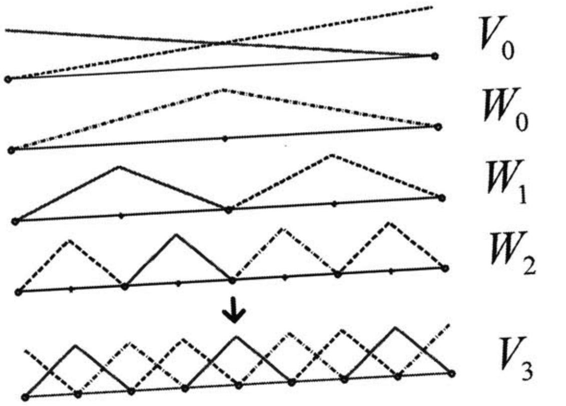


Figure 3-5: First-order Lagrangian primitive wavelet basis on regular 1D mesh.

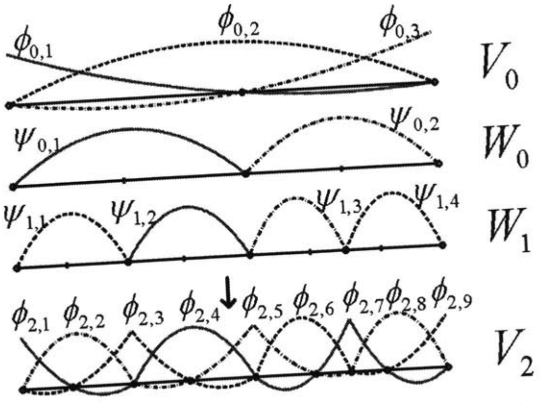


Figure 3-6: Second-order Lagrangian primitive wavelet basis on irregular 1D mesh.

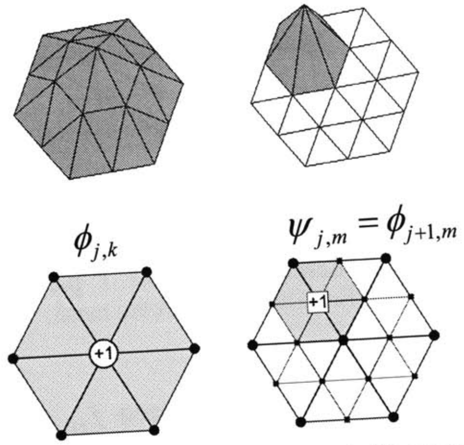


Figure 3-7: First-order Lagrangian primitive wavelet basis on regular 2D mesh.

$j-1$, are denoted by *k*, while the nodes that are not, are denoted by *m*. As we discussed in Chapter 2, at level 0, and on all finer levels, *m*-nodes have to lie on the straight line connecting the neighboring *k*-nodes. This can be guaranteed by building the mesh with subdivision. Note that we do not need to attain the same level of refinement everywhere over the mesh. Adaptive refinements, in which some partitions do not include all *m*-nodes, are entirely permitted. This is illustrated in Figure 3-4.

In a FEM context, on each level j, the mesh can be divided into a set of connecting *elements*, delimited by the points of discontinuity of the derivative of the scaling functions $\phi_{j,k}$ on that level. The support of an element will be denoted in this dissertation by the symbol $\Omega_{j,el}$.

We refer to the space of scaling functions at level j as V_j, and the space of wavelet functions at level j as W_j. The approximation space V_j, spanned by a single-scale basis of scaling functions, equals the direct sum of the coarser level wavelet space W_{j-1} and the approximation space V_{j-1}. This can be iterated upon until the coarsest level, level 0, is reached, resulting in a full multi-resolution decomposition of the space V_J of level J:

$$V_J = W_{J-1} \oplus V_{J-1} = W_{J-1} \oplus W_{J-2} \oplus \ \dots \ \oplus W_0 \oplus V_0$$

In Figures 3-5 and 3-6, only the coarsest level – level 0 – has scaling functions $\phi_{0,k}$ located at the k-nodes, while both coarsest and finer levels contain wavelet functions $\psi_{j,m}$, built around the m-nodes. On each level, the scaling functions form a complete basis for a piecewise polynomial of order n on that partition. The wavelets, at the other hand, only span the difference between two piecewise polynomials of different resolution.

3.2 Second-Generation Wavelet Construction

Traditional wavelet bases could only be constructed on regularly spaced and essentially unbounded domains, since traditional wavelets (and scaling functions) were required to be shifted and scaled versions of one single function (respectively). However, recent developments in wavelet theory allow for the construction of so-called *second-generation wavelets* on irregularly spaced, unstructured meshes over multi-dimensional, bounded domains. This wavelet construction relies on two important relations between functions of different resolutions: a Scaling Equation and a Wavelet Equation.

First, we have the Scaling Equation, a relationship between the scaling functions of different levels (see Figures 3-8 and 3-9):

$$\phi_{j,k} = \phi_{j+1,k} + \sum_m h^0_{j,k,m} \ \phi_{j+1,m} \qquad \forall j \text{ and } \forall k \tag{3.1}$$

The filter coefficients $h^0_{j,k,m}$ can vary with j across scales and with k over the domain. They will be equal to the function value of $\phi_{j,k}$ at the node m, because of the

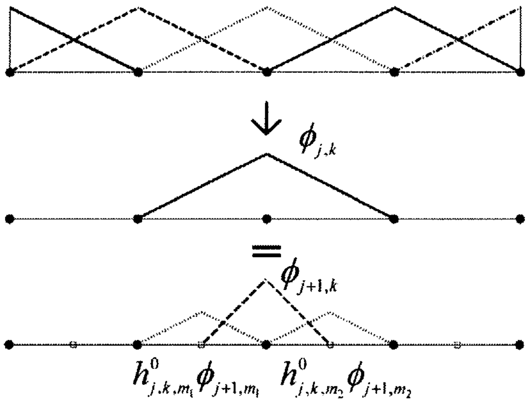


Figure 3-8: Scaling Equation for first-order wavelet basis.

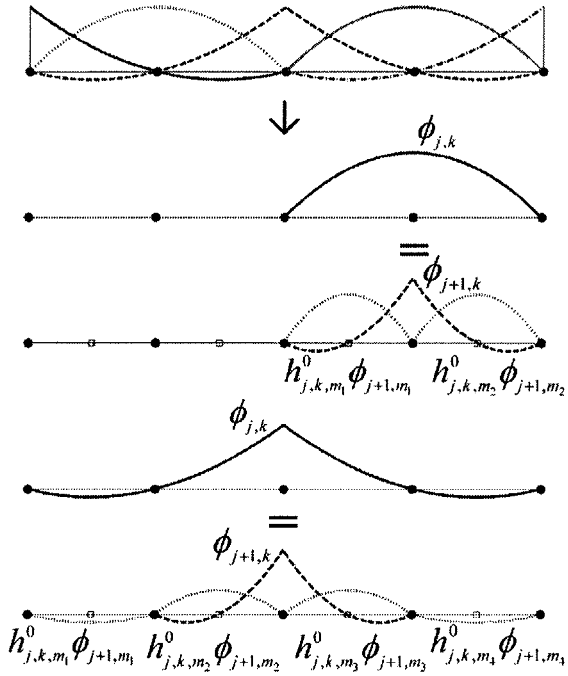

$\phi_{j,k}$
=
$\phi_{j+1,k}$
$h^0_{j,k,m_1}\phi_{j+1,m_1}$
$h^0_{j,k,m_2}\phi_{j+1,m_2}$
$\phi_{j,k}$
=
$\phi_{j+1,k}$
$h^0_{j,k,m_1}\phi_{j+1,m_1}$
$h^0_{j,k,m_2}\phi_{j+1,m_2}$
$h^0_{j,k,m_3}\phi_{j+1,m_3}$
$h^0_{j,k,m_4}\phi_{j+1,m_4}$

Figure 3-9: Scaling Equation for second-order wavelet basis.

interpolating nature of Lagrangian scaling functions. All filter coefficients $h^0_{j,k,m}$ can be collected into a sparse matrix $\mathbf{H}^0_j$ per level j. Bases of higher polynomial order n will have multiple types of scaling functions (and wavelets) per element (see e.g. Figure 3-9), which can be grouped in n-dimensional *multi-wavelet* vectors (Strang *et al.,* 1995, and Castrillón-Candás *et al.,* 2001). The Scaling Equation, Equation (3.1), is illustrated in Figures 3-8 and 3-9 for a one-dimensional regular grid, but is also valid for two-dimensional and irregular-spaced unstructured meshes.

Next, we can build wavelet functions with the Wavelet Equation based on the *lifting scheme* discovered by Sweldens (Sweldens, 1996) (see Figures 3-10 and 3-11):

$$\psi_{j,m} = \phi_{j+1,m} - \sum_k s_{j,k,m}\,\phi_{j,k} \qquad \forall j \text{ and } \forall m \tag{3.2}$$

Each wavelet is constructed by lifting a primitive wavelet – which is chosen to be a simple scaling function from a finer level $\phi_{j+1,m}$ – with scaling functions $\phi_{j,k}$ from its neighborhood. This relation too will vary both across scales and over the domain. We

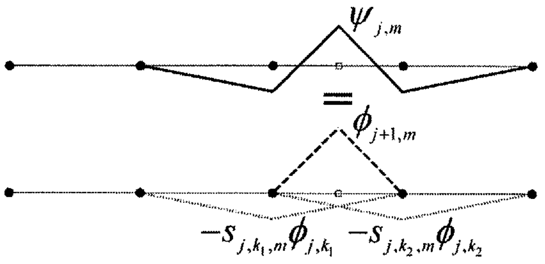


Figure 3-10: Wavelet Equation with lifting, for first-order wavelet basis.

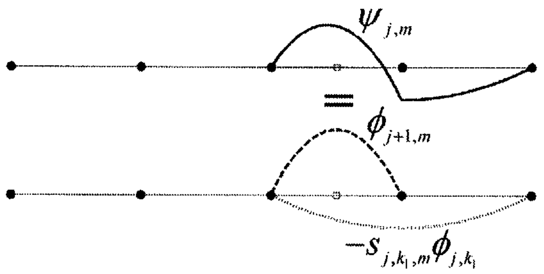


Figure 3-11: Wavelet Equation with lifting, for second-order wavelet basis.

(3.3)

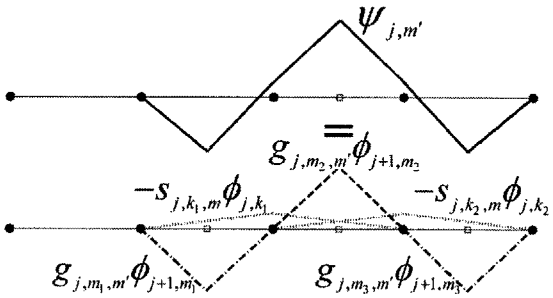

$\psi_{j,m'}$
=
$g_{j,m_2,m'}\phi_{j+1,m_2}$
$-s_{j,k_1,m}\phi_{j,k_1}$
$-s_{j,k_2,m}\phi_{j,k_2}$
$g_{j,m_1,m'}\phi_{j+1,m_1}$
$g_{j,m_3,m'}\phi_{j+1,m_3}$

Figure 3-12: Wavelet Equation with stable completion, for first-order wavelet basis.

can group all *lifting coefficients* $s_{j,k,m}$ into a sparse matrix $\mathbf{S}_j$ per level. An important realization is that this relation gives us the opportunity to design the wavelets by choosing appropriate lifting coefficients $s_{j,k,m}$. Examples of wavelets constructed with this scheme are given in Figures 3-10 and 3-11. Figure 3-10 illustrates a wavelet lifted with two neighboring scaling functions, extending the wavelet's support over three linear elements, while Figure 3-11 shows a wavelet lifted with only one scaling function, keeping its support within one quadratic element.

Note that choosing all lifting coefficients zero defaults to a traditional HB. We will call this choice a *primitive wavelet* choice. We will make extensive use of the free parameters in Equation (3.2) to tailor the wavelets to fit our needs. However, we will find further that we need additional capacity to custom-design our wavelets.

Around the same time as the discovery of the lifting scheme, a related method for wavelet design was proposed (Carnicer *et. al.*, 1996), called *stable completion.*

Introducing this concept to our framework comes down to including $g_{j,m,m'}$ in Equation (3.2) (see Figure 3-12):

$$\psi_{j,m'} = \sum_{m} g_{j,m,m'}\, \phi_{j+1,m} - \sum_{k} s_{j,k,m'}\, \phi_{j,k} \qquad \forall j \text{ and } \forall m'$$

This effectively builds a linear combination of primitive wavelets $\phi_{j+1,m}$, and then lifts this combination to create the resulting wavelet $\psi_{j,m'}$. However, an important restriction to the method is that the matrix $\mathbf{G}_j$, formed by the stable completion coefficients $g_{j,m,m'}$, has to be fully invertible. Only primitive wavelets within a local neighborhood should be used in the construction in order to ensure a compactly supported wavelet and a sparse matrix $\mathbf{G}_j$. An example of stable completion is given in Figure 3-12. Note that Equation (3.3) is a special case of Equation (3.2), with:

$$g_{j,m,m'} = \delta_{m-m'} \quad \forall j, m, m'$$

This construction generates a wider class of wavelets, generally referred to as second-generation wavelets. They can easily be constructed in closed-form on multi-dimensional unstructured meshes, which favors their use as basis functions in a FEM. Provided that $\mathbf{G}_j$ is indeed invertible, these wavelet bases (including the scaling functions) are guaranteed to span the same space as a single level space on the finest level, V_J.

Also guaranteed is the existence of a set of *dual* scaling functions $\tilde{\phi}_{j,k}$ and wavelet functions $\tilde{\psi}_{j,m}$. They fulfill the following *bi-orthogonality conditions* with respect to the primary scaling functions and wavelets:

$$\begin{aligned}
&\left\langle \phi_{j,k}, \tilde{\psi}_{j,m} \right\rangle = 0 \quad \forall j, k \text{ and } m \\
&\left\langle \psi_{j,m}, \tilde{\phi}_{j,k} \right\rangle = 0 \quad \forall j, m \text{ and } k \\
&\left\langle \phi_{j,k_1}, \tilde{\phi}_{j,k_2} \right\rangle = \delta_{k_1-k_2} \quad \forall j, k_1 \text{ and } k_2 \\
&\left\langle \psi_{j,m_1}, \tilde{\psi}_{j,m_2} \right\rangle = \delta_{m_1-m_2} \quad \forall j, m_1 \text{ and } m_2
\end{aligned} \qquad (3.4)$$

Note that the brackets in Equation (3.4) denote the L^2-inner product. It can be shown that also the duals follow a scaling function and wavelet relation, featuring filters from the primary relations, Equations (3.1) and (3.3):

$$\begin{aligned}&\tilde{\phi}_{j,k} = \tilde{\phi}_{j+1,k} + \sum_{m'} s_{j,k,m'}\tilde{\psi}_{j,m'} \quad \forall j \text{ and } \forall k \\ &\sum_{m'} g_{j,m,m'}\tilde{\psi}_{j,m'} = \tilde{\phi}_{j+1,m} - \sum_{k} h^{0}_{j,k,m}\tilde{\phi}_{j+1,k} \quad \forall j \text{ and } \forall m\end{aligned} \tag{3.5}$$

Dual scaling functions span a dual space $\tilde{V}_j$, which is generally different from the primary space V_j. A dual wavelet space $\tilde{V}_j$ can be defined similarly. The dual functions are much less smooth than the piecewise polynomial primary functions, and are generally not known in closed form, but this is not required for our analysis.

The projection of a C^0 function f on a primary (or dual) wavelet basis will result in a set of scaling function coefficients $\lambda_{j,k}$ (or $\tilde{\lambda}_{j,k}$) and wavelet coefficients $\gamma_{j,m}$ (or $\tilde{\gamma}_{j,m}$). We may use the dual (or primary) wavelet basis functions to extract these coefficients from the function f. For example, using Equations (3.4):

$$f = \sum_k \lambda_{0,k}\phi_{0,k} + \sum_{j=0}^{\infty}\sum_m \gamma_{j,m}\psi_{j,m}$$

$$\text{with} \quad \lambda_{0,k} = \left\langle f, \tilde{\phi}_{0,k}\right\rangle \quad \forall k \qquad \text{and} \qquad \gamma_{j,m} = \left\langle f, \tilde{\psi}_{j,m}\right\rangle \quad \forall j,m$$

When a wavelet function is orthogonal to all polynomials of order n (in the L^2 inner product), the n^{th} moment of that wavelet vanishes, and so do all moments from $n-1$ down to zero. Hence, that wavelet has $n+1$ *vanishing moments*. Having vanishing moments in the dual wavelets is beneficial for reducing the norm of the projection of a smooth function on the space of primary wavelets, W_j. For a Lagrangian wavelet basis of order n, we have for a pure polynomial p of order n:

$$p = \sum_k \lambda_{0,k}\phi_{0,k} = \sum_k \lambda_{0,k}\phi_{0,k} + \sum_{j=0}^{\infty}\sum_m \gamma_{j,m}\psi_{j,m}$$

$$\gamma_{j,m} = \left\langle p, \tilde{\psi}_{j,m}\right\rangle = 0 \quad \forall j,m$$

Therefore, we know that the dual wavelets will have at least $n+1$ vanishing moments. Similarly, vanishing moments in the primary wavelets reduce the projection on the dual space, $\tilde{W}_j$. Indeed, the magnitude of the wavelet coefficients, $\gamma_{j,m}$, decays as $O\left(f^{(q)}h_j^q\right)$ where q is the number of vanishing moments of $\tilde{\psi}_{j,m}$ and h_j is the characteristic support of the basis functions at level j (see e.g. Strang *et al.*, 1996). The primitive wavelets corresponding with the Lagrangian Hierarchical Basis have no vanishing moments.

All primary and dual transforms on the wavelet and scaling function coefficients can be derived by integrating Equations (3.1), (3.3) and (3.5) against the function f. For example, the *discrete wavelet transform* analyzes primary coefficients from finer primary coefficients:

$$\sum_{m'} g_{j,m,m'}\gamma_{j,m'} = \lambda_{j+1,m} - \sum_{k} h^0_{j,k,m}\,\lambda_{j+1,k} \qquad \forall j \text{ and } \forall m$$
$$\lambda_{j,k} = \lambda_{j+1,k} + \sum_{m'} s_{j,k,m'}\,\gamma_{j,m'} \qquad \forall j \text{ and } \forall k$$

And the *dual discrete wavelet transform* analyzes dual coefficients from finer dual coefficients:

$$\tilde{\lambda}_{j,k} = \tilde{\lambda}_{j+1,k} + \sum_{m} h^0_{j,k,m}\,\tilde{\lambda}_{j+1,m} \qquad \forall j \text{ and } \forall k$$
$$\tilde{\gamma}_{j,m'} = \sum_{m} g_{j,m,m'}\,\tilde{\lambda}_{j+1,m} - \sum_{k} s_{j,k,m'}\,\tilde{\lambda}_{j,k} \qquad \forall j \text{ and } \forall m'$$

These wavelet transforms can be grouped into square transform matrices:

$$\mathbf{T}_j = \begin{bmatrix} \mathbf{L}_j \\ \mathbf{H}_j \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{S}_j \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{G}_j^{-1}\mathbf{H}_j^0 & \mathbf{G}_j^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{P}_j^k \\ \mathbf{P}_j^m \end{bmatrix}$$
$$\tilde{\mathbf{T}}_j = \begin{bmatrix} \tilde{\mathbf{L}}_j \\ \tilde{\mathbf{H}}_j \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{S}_j & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{H}_j^0 \\ \mathbf{0} & \mathbf{G}_j \end{bmatrix} \begin{bmatrix} \mathbf{P}_j^k \\ \mathbf{P}_j^m \end{bmatrix} \qquad (3.6)$$

Matrices $\mathbf{P}_j^k$ and $\mathbf{P}_j^m$ are permutation matrices. They re-order the level $j+1$ k-nodes in a group of k-nodes and a group of m-nodes on level j. The matrix $\mathbf{T}_j$ consists of a low-channel $\mathbf{L}_j$ and a high-channel $\mathbf{H}_j$, and performs column-wise a discrete wavelet transform over all coefficients from level $j+1$ to j. Similarly, the matrix $\tilde{\mathbf{T}}_j$ is made up of a low-channel $\tilde{\mathbf{L}}_j$ and a high-channel $\tilde{\mathbf{H}}_j$, and executes column-wise a dual discrete wavelet transform, and $\tilde{\mathbf{T}}_j^T$ row-wise a dual discrete wavelet transform. Thus:

$$\begin{bmatrix}\boldsymbol{\lambda}_{j-1}\\ \boldsymbol{\gamma}_{j-1}\\ \boldsymbol{\gamma}_{j}\end{bmatrix}=\mathbf{T}_{j-1}\begin{bmatrix}\boldsymbol{\lambda}_{j}\\ \boldsymbol{\gamma}_{j}\end{bmatrix}\qquad \text{and}\qquad \begin{bmatrix}\boldsymbol{\lambda}_{j}\\ \boldsymbol{\gamma}_{j}\end{bmatrix}=\mathbf{T}_{j}\boldsymbol{\lambda}_{j+1}\qquad \text{etc...}$$

Note that $\mathbf{T}_{j-1}$ only transforms the level j scaling coefficients $\boldsymbol{\lambda}_j$, and not the wavelet coefficients $\boldsymbol{\gamma}_j$, previously generated by $\mathbf{T}_j$. Of course, the same goes for the dual transform matrices $\tilde{\mathbf{T}}_j$. Because of the bi-orthogonality conditions given in Equations (3.4), we have the following important property:

$$\tilde{\mathbf{T}}_j^T\mathbf{T}_j=\mathbf{I}$$

For compactly supported wavelets, the sparse $\tilde{\mathbf{T}}_j$ transform can be applied to a vector with $O(N)$ operation cost, where N is the number of degrees of freedom, or the dimension of the problem. Moreover, this computation can even be done in-place, if $\mathbf{G}_j=\mathbf{I}$. However, the $\mathbf{T}_j=\left(\tilde{\mathbf{T}}_j\right)^{-1}$ matrix will be fully populated in general, and thus $\mathbf{T}_j$ cannot be cheaply applied to a vector. Only for compactly supported wavelets constructed with only lifting and *without* stable completion, i.e. $g_{j,m,m'}=\delta_{m-m'}$ or $\mathbf{G}_j=\mathbf{I}$, will the $\mathbf{T}_j$ transform be $O(N)$, and in-place.

3.3 Wavelet Basis FEM

A well-posed boundary value problem specified by a linear second-order elliptic Partial Differential Equation can be solved numerically with a conventional Galerkin Finite Element Method. For notational simplicity, we assume homogeneous Dirichlet boundary conditions around the domain Ω:

$$\begin{aligned}
&-\nabla\cdot(\mathbf{P}\nabla u)+\mathbf{q}\cdot\nabla u+ru=f\\
&\rightarrow\quad \mathbf{A}\,\mathbf{u}=\mathbf{f}\\
&\qquad\text{with}\quad \mathrm{A}[\mathrm{k'},\mathrm{k}]=a(\varphi_k,\varphi_{k'})=\int_\Omega\left(\mathbf{P}\nabla\varphi_k\cdot\nabla\varphi_{k'}+(\mathbf{q}\cdot\nabla\varphi_k)\varphi_{k'}+r\varphi_k\varphi_{k'}\right)d\Omega\\
&\qquad\text{and}\quad \mathrm{f}[\mathrm{k'}]=b(\varphi_{k'})=\int_\Omega f\varphi_{k'}d\Omega
\end{aligned}\tag{3.7}$$

In Equation (3.7), the coefficient matrix $\mathbf{P}$, vector $\mathbf{q}$ and scalar r may be functions of the spatial coordinates, as are u and f. We choose the set of test functions $\varphi_{k'}$,

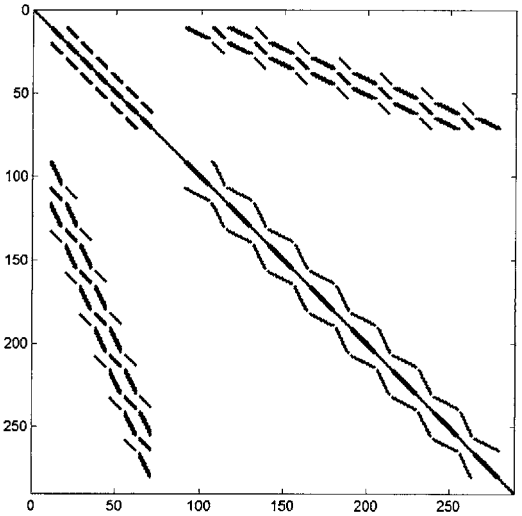


Figure 3-13: Sparsity of two-level first-order HB FEM stiffness matrix for 2D Poisson.

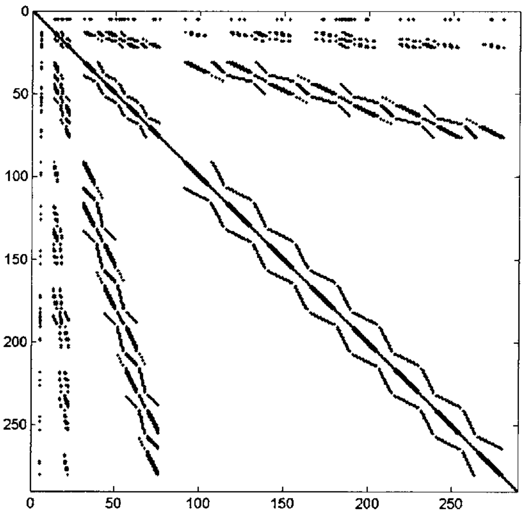


Figure 3-14: Sparsity of full-resolution first-order HB FEM stiffness matrix for 2D Poisson.

weighing the approximation error, to coincide with the set of trial functions φ_k. With this choice, the method is referred to as a *Ritz-Galerkin* method. In particular, we choose the commonly used Lagrangian FE basis functions φ_k. They are widely used in the FEM community, flexible over geometries and dimensions, and formulated in closed-form. Moreover, we showed in the previous paragraphs how to draw a multi-resolution wavelet basis from these basis functions. Therefore, we will not search for a solution as a linear combination of single-scale (finest level J) scaling functions:

$$u = \sum_k \lambda_{J,k}\phi_{J,k}$$

But, instead, we will choose a multi-resolution basis for the basis functions φ_k and $\varphi_{k'}$. With a multi-resolution basis of only two scales, we have:

$$u = \sum_k \lambda_{J-1,k}\phi_{J-1,k} + \sum_m \gamma_{J-1,m}\psi_{J-1,m}$$

With a full multi-resolution basis, down to level 0, we can write the solution as:

$$u = \sum_k \lambda_{0,k}\phi_{0,k} + \sum_{j=0}^{J-1}\sum_m \gamma_{j,m}\psi_{j,m}$$

These wavelet bases span the same solution space as the single-scale basis. Therefore, the FE solution remains the same. However, a multi-resolution system can feature additional qualities: it can reduce the solution cost or facilitate adaptive methods. We can use wavelet theory from paragraph 3.2 to construct transformations between the single-scale system of level J and the corresponding two-level multi-resolution system:

$$\begin{aligned}
&\mathbf{A}_J\mathbf{u}_J = \mathbf{f}_J \\
&\Leftrightarrow \tilde{\mathbf{T}}_{J-1}\mathbf{A}_J\tilde{\mathbf{T}}_{J-1}^T\mathbf{T}_{J-1}\mathbf{u}_J = \tilde{\mathbf{T}}_{J-1}\mathbf{f}_J \quad \text{because } \tilde{\mathbf{T}}_{J-1}^T\mathbf{T}_{J-1} = \mathbf{I} \\
&\Leftrightarrow \mathbf{A}_J^{[J-1]}\mathbf{u}_J^{[J-1]} = \mathbf{f}_J^{[J-1]} \quad \text{with } \mathbf{A}_J^{[J-1]} = \tilde{\mathbf{T}}_{J-1}\mathbf{A}_J\tilde{\mathbf{T}}_{J-1}^T \\
&\qquad\qquad\qquad\qquad\qquad\quad\;\; \mathbf{u}_J^{[J-1]} = \mathbf{T}_{J-1}\mathbf{u}_J \\
&\qquad\qquad\qquad\qquad\qquad\quad\;\; \mathbf{f}_J^{[J-1]} = \tilde{\mathbf{T}}_{J-1}\mathbf{f}_J
\end{aligned}$$

Superscript $[J-1]$ signifies a multi-resolution entity with coarsest scale $J-1$. The multi-resolution system above has the following internal structure, with sparsity shown in Figure 3-13:

$$\begin{bmatrix} \mathbf{A}_{J-1} & \mathbf{C}_{J-1} \\ \mathbf{D}_{J-1} & \mathbf{\Delta}_{J-1} \end{bmatrix}\begin{bmatrix} \boldsymbol{\lambda}_{J-1} \\ \boldsymbol{\gamma}_{J-1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{J-1} \\ \mathbf{t}_{J-1} \end{bmatrix}$$

The sub-matrix $\mathbf{A}_{J-1}$ is the single-scale stiffness matrix of a level $J-1$ FEM system, basically a coarser version of the FE matrix $\mathbf{A}_J$:

$$\mathbf{A}_{J-1}\mathbf{u}_{J-1} = \mathbf{f}_{J-1}$$

The sub-matrix $\mathbf{\Delta}_{J-1}$ contains the wavelet-wavelet interaction of level $J-1$. The sub-matrices $\mathbf{C}_{J-1}$ and $\mathbf{D}_{J-1}$ couple the finer wavelet (detail) part of the system with the coarser scaling function part of the system. Note that the single-scale and multi-resolution stiffness matrices are generally not symmetric, since $a(\varphi_k, \varphi_{k'})$ is not in general. If $a(\varphi_k, \varphi_{k'})$ is in fact symmetric, we have $\mathbf{D}_{J-1} = \mathbf{C}_{J-1}^T$. The symbol $\mathbf{u}_j$ denotes the solution of a level j system, while $\boldsymbol{\lambda}_j$ stands for the projection of a (possibly finer) solution onto a level j scaling function space.

The transforms $\tilde{\mathbf{T}}_{J-i}$ can be repeated until $J-i=0$, and full multi-resolution is achieved:

$$\begin{aligned}
&\mathbf{A}_J\mathbf{u}_J = \mathbf{f}_J \\
&\Leftrightarrow \mathbf{A}_J^{[J-i]}\mathbf{u}_J^{[J-i]} = \mathbf{f}_J^{[J-i]} \quad \text{with } \mathbf{A}_J^{[J-i]} = \tilde{\mathbf{T}}_{J-i}\ldots\tilde{\mathbf{T}}_{J-1}\mathbf{A}_J\tilde{\mathbf{T}}_{J-1}^T\ldots\tilde{\mathbf{T}}_{J-i}^T \\
&\qquad\qquad\qquad\qquad\qquad\quad\;\; \mathbf{u}_J^{[J-i]} = \mathbf{T}_{J-i}\ldots\mathbf{T}_{J-1}\mathbf{u}_J \\
&\qquad\qquad\qquad\qquad\qquad\quad\;\; \mathbf{f}_J^{[J-i]} = \tilde{\mathbf{T}}_{J-i}\ldots\tilde{\mathbf{T}}_{J-1}\mathbf{f}_J
\end{aligned}$$

The internal structure of this full multi-resolution system is plotted in Figure 3-14, and given by:

$$\begin{bmatrix} \mathbf{A}_{J-i} & \mathbf{C}_{J-i,\mathbf{A}_{J-i}} & \cdots & \mathbf{C}_{J-1,\mathbf{A}_{J-i}} \\ \mathbf{D}_{J-i,\mathbf{A}_{J-i}} & \boldsymbol{\Delta}_{J-i} & & \mathbf{C}_{J-1,\boldsymbol{\Delta}_{J-i}} \\ \vdots & & \ddots & \\ \mathbf{D}_{J-1,\mathbf{A}_{J-i}} & \mathbf{D}_{J-1,\boldsymbol{\Delta}_{J-i}} & & \boldsymbol{\Delta}_{J-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_{J-i} \\ \boldsymbol{\gamma}_{J-i} \\ \vdots \\ \boldsymbol{\gamma}_{J-1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{J-i} \\ \mathbf{t}_{J-i} \\ \vdots \\ \mathbf{t}_{J-1} \end{bmatrix}$$

Of course, we could also construct directly this multi-resolution system, without transforming the finer single-scale system, since we have all scaling functions and wavelets in closed-form. This system can be solved for $\mathbf{u}_J^{[J-i]}$, which will produce the solution $\mathbf{u}_J$, after performing inverse wavelet transforms on $\mathbf{u}_J^{[J-i]}$. The cost of the transform operations will remain $O(N)$ since we only use $\tilde{\mathbf{T}}_{J-i}$ and do not need to compute or apply $\mathbf{T}_{J-i}$. Indeed, we can compute the single-scale solution $\mathbf{u}_J$ as:

$$\mathbf{u}_J = \tilde{\mathbf{T}}_{J-1}^T \ldots \tilde{\mathbf{T}}_{J-i}^T \mathbf{u}_J^{[J-i]}$$

The cost of matrix assembly, solution and transformations will be discussed in detail in Chapter 6, where we present the complexity of our wavelet method.

It is important to note that in general only $\boldsymbol{\lambda}_J$ equals $\mathbf{u}_J$ (the solution to $\mathbf{A}_J \mathbf{u}_J = \mathbf{f}_J$), while all coarser $\boldsymbol{\lambda}_{J-i}$ differ from $\mathbf{u}_{J-i}$ (the solution to $\mathbf{A}_{J-i}\mathbf{u}_{J-i} = \mathbf{f}_{J-i}$) unless the coupling matrices $\mathbf{C}_j$ are zero. This is generally not the case for the primitive wavelet choice of a traditional Hierarchical Basis.

3.4 Operator-Customized Wavelets

We can now use the second-generation wavelet framework described in paragraph 3.2 to modify our wavelet basis and customize it to the operator. Indeed, we will propose the construction of a basis of compactly supported wavelets that are operator-orthogonal to all scaling functions. That is, we wish to satisfy the following condition for *operator orthogonality*:

$$C_{j,k,m} = a\left(\psi_{j,m}, \phi_{j,k}\right) = 0 \qquad \forall m,k \text{ and } 0 \le j \le J-1$$

The resulting transform $\tilde{\mathbf{T}}_{J-i}$ will effectively eliminate the coupling between scales, i.e. $\mathbf{C}_{J-i}=\mathbf{0}$. This is in contrast with a Hierarchical Basis (see e.g. Figure 3-13), where we have $\mathbf{C}_{J-i}\neq\mathbf{0}$ in general (with the exception of the linear HB for the one-dimensional Poisson's Equation). Note that if $\mathbf{C}_j=\mathbf{0}$ for every level j, there will not be any coupling between wavelets of different levels either, since both scaling functions and wavelets are themselves linear combinations of finer level scaling functions. Thus:

$$\mathbf{C}_j=\mathbf{0} \quad \Rightarrow \quad \mathbf{C}_{j,\mathbf{\Lambda}_0}=\mathbf{0} \text{ and } \mathbf{C}_{j,\mathbf{\Delta}_{j-i}}=\mathbf{0} \quad \forall i=1\ldots j$$

The scheme to derive our wavelet transform will essentially be based upon the mesh geometry (the connectivity in particular) and general enough to scale-decouple any relevant operator for a Lagrangian finite element space of any given order.

The thus achieved decoupling across scales in the multi-resolution system can be very suitable for adaptive methods. Indeed, if $\mathbf{C}_{J-1}=\mathbf{0}$ we can find $\mathbf{u}_J$ from simply adding details – the solution of $\mathbf{\Delta}_{J-1}\boldsymbol{\gamma}_{J-1}=\left(\mathbf{t}_{J-1}-\mathbf{D}_{J-1}\boldsymbol{\lambda}_{J-1}\right)$ – to a coarser solution $\mathbf{u}_{J-1}=\boldsymbol{\lambda}_{J-1}$, without having to update this coarser solution. For a symmetrical $\mathbf{A}_J$ this procedure simplifies because $\mathbf{D}_{J-1}=\mathbf{0}$. Compact operator-orthogonal wavelets could be particularly useful in solving large-scale problems where the solution error has large local concentrations, and this solution can be computed adaptively by assuming zero details everywhere else. Then, the dimensions of $\mathbf{\Delta}_{J-1}$ will be much smaller than $\mathbf{A}_{J-1}$, so not having to re-compute the coarser $\mathbf{u}_{J-1}=\boldsymbol{\lambda}_{J-1}$ will surely pay off.

Note that a special case of operator-orthogonality is orthogonality. We define an *orthogonal* wavelet basis as a wavelet basis for which:

$$\int_{\Omega}\psi_{j,m}\phi_{j,k}d\Omega=0 \qquad \forall m,k \text{ and } 0\le j\le J-1$$

This is the same as saying the basis is operator-orthogonal with respect to the identity operator. Such a basis will yield a scale-decoupled mass-matrix. If we can build an operator-orthogonal basis for any given operator, we can also generate an orthogonal basis. Note that in our definition of orthogonality, the wavelets do not have to be orthogonal to each other. Some other authors use the term *semi-orthogonal* – indicating orthogonality across scales, but not over the spatial domain – for what we will simply call *orthogonal*.

We stress the importance of keeping our customized wavelets within compact and local support. If the wavelets would not be compactly supported, but spread all over the domain, our transform costs would dramatically increase, since none of the wavelet transforms would be sparse. It then becomes impossible to transform a multi-resolution solution to a full single-scale solution with only $O(N)$ cost.

3.5 Conclusion

To overcome the limitations of the Hierarchical Basis FEM, we generalized this method to a more powerful Wavelet Ritz-Galerkin FEM. We applied second-generation wavelet theory to construct multi-resolution bases that span the Lagrangian finite element space of any given order, on multi-dimensional unstructured meshes. The construction method allows us to tailor the wavelet basis to our needs. In general the multi-resolution stiffness matrix is coupled across scales. However, by making our wavelets operator-orthogonal to the scaling functions, with respect to the problem's operator, we can eliminate all scale-coupling.

In the following chapter, we will propose the construction of a compact operator-orthogonal wavelet basis for one-dimensional problems, by starting with the Poisson's Equation and then generalizing this to any second-order PDE. Next, we will repeat the wavelet customization for two-dimensional problems.

4.

1D Wavelet Customization

4.1 Poisson's Equation

Now that we have the building tools to construct and tailor a wavelet basis, we will use them to customize a wavelet basis to any given second-order operator. We first start with a simple one-dimensional Poisson's Equation over the domain $[x_1, x_2]$:

$$-\frac{\partial^2 u}{\partial x^2} = f \quad \left| \begin{array}{l} u(x_1) = 0 \\ u(x_2) = 0 \end{array} \right. \tag{4.1}$$

For simplicity, let us assume for now homogeneous Dirichlet boundary conditions. We may solve this problem with a Ritz-Galerkin FEM, with both trial functions φ_k and test functions $\varphi_{k'}$ spanning a Lagrangian finite element space of any given order n:

$$\begin{aligned} &\mathbf{A}\,\mathbf{u} = \mathbf{f} \\ &\mathrm{A}[\mathrm{k',k}] = a(\varphi_k, \varphi_{k'}) = \int_\Omega \frac{\partial \varphi_k}{\partial x} \frac{\partial \varphi_{k'}}{\partial x} dx \\ &\mathrm{f}[\mathrm{k'}] = b(\varphi_{k'}) = \int_\Omega f \varphi_{k'} dx \end{aligned}$$

The mesh can be irregular-spaced, as long as the FEM problem is well-conditioned. Note that a highly irregular mesh could be dealt with by treating it as a non-uniformly refined mesh. We now choose for φ_k a multi-resolution wavelet basis instead of a single-scale basis, both spanning the Lagrangian finite element space of any given order n. We will propose, for any given order n, the construction of a specific compact wavelet basis that is operator-orthogonal to the scaling functions, with respect to the Laplace operator. Thus, all wavelets and scaling functions need to satisfy:

$$a\left(\psi_{j,m},\phi_{j,k}\right)=\int_{\Omega}\frac{\partial\psi_{j,m}}{\partial x}\frac{\partial\phi_{j,k}}{\partial x}dx=0 \qquad \forall j,m,k$$

It is known in Hierarchical Basis theory that a multi-resolution basis of primitive linear hats scale-decouples the FEM stiffness matrix for the Poisson's Equation on a one-dimensional regular grid. Thus the linear primitive wavelets are operator-orthogonal to the scaling functions with respect to the Laplace operator on a one-dimensional regular grid. This property extends to linear hats on irregular meshes as well. However, this property does not naturally extend to higher-order primitive wavelet bases (i.e. all wavelets are chosen to be simple scaling functions from a finer level) (see e.g. Figure 4-3). As our first main contribution, we propose the construction of a compact operator-orthogonal wavelet basis for any given order n, on an irregular mesh. We can achieve this by simply lifting our primitive wavelets, such that each wavelet $\psi_{j,m}$ has a number of vanishing moments equal to $n-1$, where n is the order of the Lagrangian finite element space spanned by $\phi_{j,k}$.

The proof starts with the following assessment of an *inheritance of vanishing moments* property. Indeed, a wavelet's derivative inherits vanishing moments from the compact support and vanishing moments of the wavelet itself, based on the integration by parts formula:

$$\int_\Omega \frac{\partial \psi}{\partial x} dx = \psi\Big|_{x_1}^{x_2} = 0$$

$$\int_\Omega x\frac{\partial \psi}{\partial x} dx = x\psi\Big|_{x_1}^{x_2} - \int_\Omega \psi dx = 0 \quad \Leftrightarrow \quad \int_\Omega \psi dx = 0$$

$$\int_\Omega x^2\frac{\partial \psi}{\partial x} dx = x^2\psi\Big|_{x_1}^{x_2} - \int_\Omega 2x\psi dx = 0 \quad \Leftrightarrow \quad \int_\Omega x\psi dx = 0$$

See Chapter 3 for an explanation of the term *vanishing moment.* If a wavelet is zero-valued at the integration boundaries, the derivative of the wavelet will have one vanishing moment. In addition, each vanishing moment in the wavelet will result in an additional vanishing moment in the derivative of the wavelet. Now, note that the following is a *sufficient condition* for operator-orthogonality between wavelets and scaling functions of order n:

$$\int_\Omega \frac{\partial \psi_{j,m}}{\partial x} t\, dx = \int_{\Omega_{j,el}} \frac{\partial \psi_{j,m}}{\partial x} t\, dx = 0 \qquad \forall j,m,k \qquad m \in \Omega_{j,el} \tag{4.2}$$

In Equation (4.2), t is any pure polynomial of order $n-1$ in x. Indeed, for any given order n, $\partial\phi_{j,k}/\partial x$ will be a pure polynomial of order $n-1$ over the support of $\psi_{j,m}$, if we keep the support of the wavelet $\psi_{j,m}$ within the element that contains its m-node ($\Omega_{j,el} \mid m \in \Omega_{j,el}$), and not outside that element. Let us define a scaling function $\phi_{j,k}$ to be *interior* with respect to a set of elements $\Omega_{j,set}$, if the function's support lies completely within those elements (i.e. $k \in \Omega_{j,set}$ and also $k \notin \Omega_{j,total\setminus set}$). In the remainder of this paragraph, this definition for the term *interior scaling function* applies, with $\Omega_{j,set}$ as the single element $\Omega_{j,el}$ that contains the m-node associated with the primitive wavelet. Now, we can easily check that, for any given order n, we have exactly $n-1$ interior scaling functions for each element $\Omega_{j,el}$. With these $n-1$ scaling functions per element $\Omega_{j,el}$, we can simply lift each primitive wavelet, using Equation (3.2), such that we obtain $n-1$ vanishing moments in the resulting wavelet $\psi_{j,m}$. This wavelet will have no support outside $\Omega_{j,el}$, and because of the inheritance of vanishing moments, $\partial\psi_{j,m}/\partial x$ will have n vanishing moments. Hence, we have satisfied the sufficient condition for operator-orthogonality for our wavelet basis. For first-order Lagrangians, we do not need

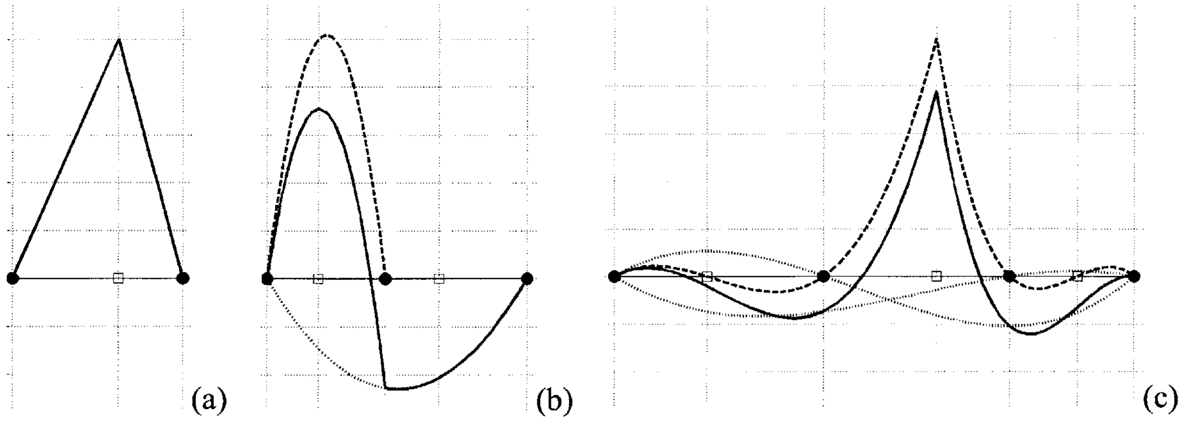


Figure 4.1: First- (a), second- (b), third-order (c) wavelets customized to Laplace operator.

any vanishing moments: the linear element has no interior scaling functions and the primitive wavelets of a Hierarchical Basis (see Figure 4-1) are naturally operator-orthogonal. In the second-order case, we lift each primitive wavelet with the one interior scaling function to enforce one vanishing moment (compare Figure 4-1b with Figure 3-6). Thus, the first derivative of such wavelet inherits two vanishing moments, making it orthogonal to the derivative of the scaling functions, which means that the resulting wavelet is operator-orthogonal. In the third-order case, we can lift with the two interior scaling functions to enforce two vanishing moments (see Figure 4-1). This results in three vanishing moments in the wavelet's derivative, and hence operator-orthogonality. We leave the higher-order cases for the reader's pleasure. It is important to stress that nowhere in our proof we have assumed a regular-spaced grid. Therefore, our construction is applicable to any irregular one-dimensional mesh.

If we use these Laplace-operator-orthogonal wavelet bases in a Ritz-Galerkin FEM solving Poisson's Equation (Equation (4-1)), we obtain full decoupling across scales (i.e. $\mathbf{C}_j = \mathbf{0}$ and $\mathbf{D}_j = \mathbf{C}_j^T = \mathbf{0}$). Moreover, the stiffness matrix will in this case also be decoupled over the domain within each level, since wavelets associated with different elements do not overlap. Hence, all detail matrices $\mathbf{\Delta}_j$ are block-diagonal with blocks of dimension n by n, and the resulting multi-resolution stiffness matrix $\mathbf{A}_J^{[0]}$ will therefore be block-diagonal as well (see Figure 4-2). Consequently, if we would apply a

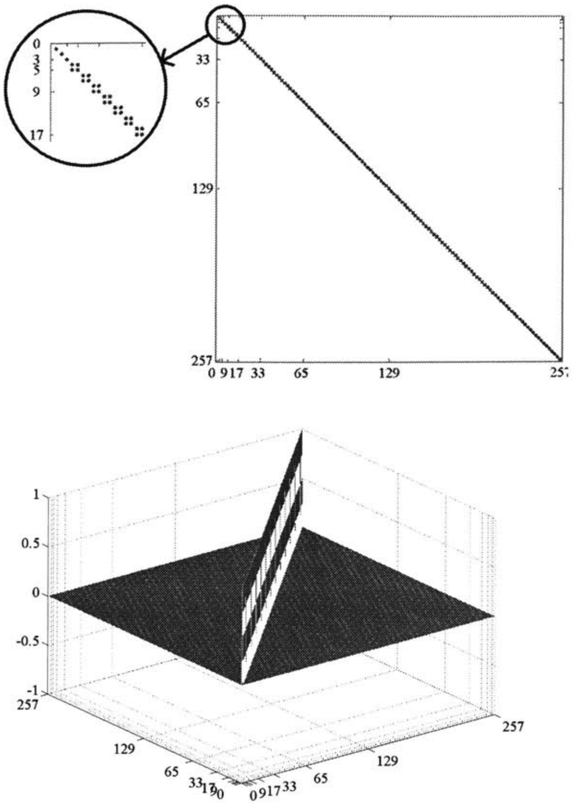


Figure 4-2: Stiffness matrix for second-order Laplace-customized wavelets.

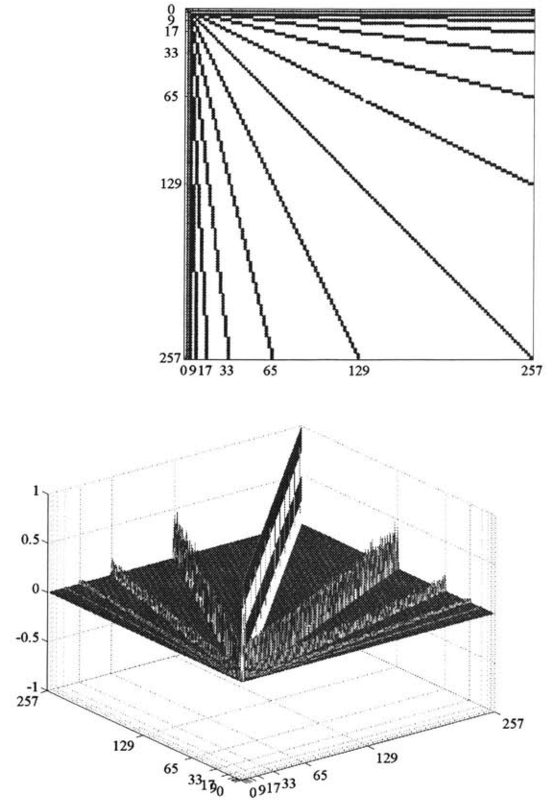


Figure 4-3: Stiffness matrix for second-order HB.

simple diagonal pre-conditioner to $\mathbf{A}_J^{[0]}$ (corresponding to a mere rescaling of the basis), $\mathbf{A}_J^{[0]}$ would have a very low condition number that remains constant with increasing problem dimensions (see Figure 4-4).

What happens at the boundary? At a Dirichlet boundary, we will force all wavelets to be zero-valued on the boundary. Note that for one-dimensional problems no primitive wavelet sits right on the boundary, because the boundaries are defined by *k*-nodes only and not by *m*-nodes. In addition, Laplace-operator-customized wavelets immediately next to the boundary will be always zero-valued on the boundary, since we only lift with interior scaling functions. As a result, homogeneous or inhomogeneous Dirichlet boundary conditions will be entirely absorbed by the scaling functions on the boundary nodes. Consequently, we can solve for those scaling coefficients independently of the rest of the system. Neumann boundaries are handled slightly differently. Such boundary conditions will be absorbed by both the scaling functions near and at the

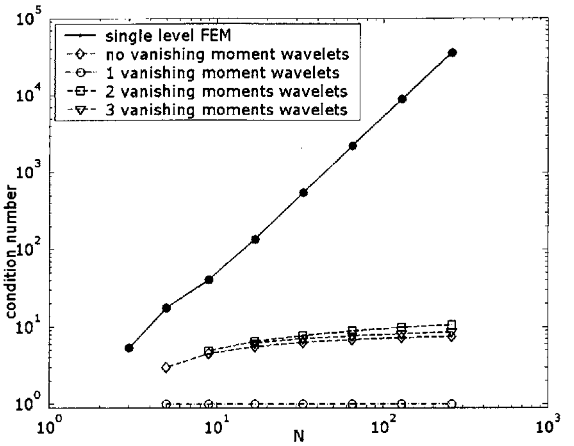


Figure 4-4: Condition number of quadratic wavelet FE matrix for Poisson's Equation.

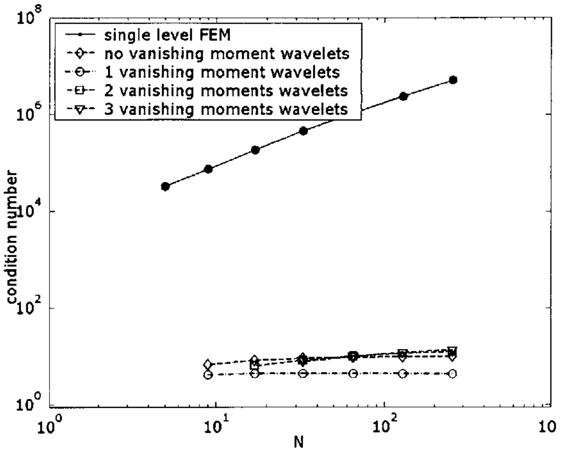


Figure 4-5: Condition number of quadratic wavelet FE matrix for non-Poisson PDE.

boundary, and the wavelets near the boundary. We can follow exactly the same procedure as for a single-scale FEM with a Neumann boundary.

In this paragraph, we have demonstrated how to build a basis of compactly supported Lagrangian finite-element wavelets of any given order, operator-orthogonal to all scaling functions with respect to the Laplace operator, on irregularly spaced one-dimensional grids.

4.2 Second-Order Partial Differential Equations

How can we extend this achievement to other more general linear elliptic second-order operators? Consider the following one-dimensional PDE with varying coefficients over the domain $[x_1, x_2]$. At this point, we assume for simplicity homogeneous Dirichlet boundary conditions:

$$-\frac{\partial}{\partial x}\left(p\frac{\partial u}{\partial x}\right)+q\frac{\partial u}{\partial x}+ru=f \quad \left|\begin{array}{l} u(x_1)=0 \\ u(x_2)=0 \end{array}\right. \tag{4.3}$$

Now, we wish to satisfy the following condition for *operator-orthogonality* with respect to the operator of this PDE:

$$C_{j,k,m}=a\left(\psi_{j,m},\phi_{j,k}\right)=\int_{\Omega}\left(p\frac{\partial\psi_{j,m}}{\partial x}\frac{\partial\phi_{j,k}}{\partial x}+q\frac{\partial\psi_{j,m}}{\partial x}\phi_{j,k}+r\psi_{j,m}\phi_{j,k}\right)dx=0$$

$$\forall m,k \text{ and } 0\le j\le J-1$$

The primitive linear hat (first-order Lagrangian) wavelet basis is not operator-orthogonal with respect to this general operator (except for the special case of Poisson's Equation). Nor any of the higher-order primitive wavelet bases are naturally operator-orthogonal. However, if we would use Laplace-operator-orthogonal wavelets in a Ritz-Galerkin FEM solving a general PDE, we would find only minimal coupling across scales. It is important to note that this applies to any order of Laplace-operator-orthogonal basis, as constructed in the previous section (see Figure 4-1). In contrast, the primitive wavelet bases all yield stiffness matrices that are highly coupled across scales. As a corollary, we have found for a non-Poisson PDE a small and relatively constant condition number of the Laplace-operator-orthogonal (pre-conditioned) multi-resolution system matrix $\mathbf{A}_J^{[0]}$, contrasting a faster growing condition number for the primitive multi-resolution system (see Figure 4-5). Although there surely could be benefits in using a wavelet basis with 'approximate' decoupling power, we still pursue our goal of complete operator-orthogonality with respect to any given operator.

4.2.1 Non-Lagrangian Wavelet Basis

Such operator-orthogonality has been achieved by Jawerth and Sweldens for Equation (4.3) with both q and r equal to zero (Jawerth *et al.*, 1993). They also applied this to the Helmholtz Equation, which is treated as a special case of the former. Their method is based on the application of anti-derivatives to a basis of naturally orthogonal wavelets. Looking at their work from our wavelet framework's perspective, we propose to construct operator-orthogonal wavelets as follows. All wavelets need to satisfy the following condition (note that p and t are functions in x):

$$a\left(\psi_{j,m},\phi_{j,k}\right)=\int_{\Omega}\frac{\partial\psi_{j,m}}{\partial x}\frac{\partial\phi_{j,k}}{\partial x}p\,dx=\int_{\Omega}\frac{\partial\psi_{j,m}}{\partial x}t\,dx=0 \quad \text{with} \quad t=\frac{\partial\phi_{j,k}}{\partial x}p$$

If we now make sure that t is nothing but pure polynomial of order $n-1$ over the support of $\psi_{j,m}$ and we give $\psi_{j,m}$ $n-1$ vanishing moments, we are guaranteed to have operator-orthogonality. This has been shown in paragraph 4.1. To satisfy the first condition, we are forced to relax the choice of piecewise polynomial basis functions that span a Lagrangian finite element space. However, we could work with scaling functions that are derived from a set of parent scaling functions $\phi^*_{j,k}$ that do span a Lagrangian finite element space of any given order n. For example, we can build new scaling functions $\phi_{j,k}$ satisfying the following relations:

$$\frac{\partial \phi_{j,k}}{\partial x} p = t = \sum_{el} c_{el} t_{el} \quad \Leftrightarrow \quad \phi_{j,k} = \int_{\Omega} \frac{t}{p} dx = \sum_{el} c_{el} \int_{\Omega} \frac{t_{el}}{p} dx$$

where t_{el} is obtained from $\dfrac{\partial \phi^*_{j,k}}{\partial x} = \sum_{el} t_{el} \quad \Leftrightarrow \quad \phi^*_{j,k} = \sum_{el} \int_{\Omega} t_{el} dx$

In these equations, the function t is a discontinuous order $n-1$ piecewise polynomial, which resembles the first derivative of the scaling function $\phi^*_{j,k}$ corresponding to a Lagrangian finite element space of given order n. The function is made up of polynomial pieces t_{el} per element, which are zero outside of that element. Those pieces simply summed together form the first derivative of $\phi^*_{j,k}$. However, in the function t, each piece is multiplied by a parameter c_{el} such that the integral of t/p is zero wherever $\phi^*_{j,k}$ is zero. This ensures the compactness of the resulting $\phi_{j,k}$. In the linear case $n=1$, for example, the sum of all plain pieces t_{el} (with $c_{el}=1$) is a piecewise constant function spanning two elements, positive over the left element and negative over the right element. The running integral of this function is exactly a compactly supported linear hat. Also the function t will be piecewise constant, but with heights depending on the function p. The thus created compactly supported scaling functions $\phi_{j,k}$ will in general not anymore be piecewise polynomial, though they are based upon piecewise polynomial $\phi^*_{j,k}$. However, because the scaling functions $\phi_{j,k}$ are based upon parent scaling functions $\phi^*_{j,k}$ with a scaling relation, they too will satisfy a scaling relation Equation (3.1). Wavelets can be constructed with the lifting scheme of Equation (3.2). Moreover, we can lift each primitive wavelet (equal to a finer level scaling function $\phi_{j+1,m}$) with $n-1$ interior scaling functions, to enforce $n-1$ vanishing moments while staying within the support of the primitive wavelet's element $\Omega_{j,el}$. Thus, t will be pure polynomial of order $n-1$ over the support of each wavelet $\psi_{j,m}$, and because of the inheritance of vanishing moments property, we have now an operator-orthogonal wavelet basis. Note that, since we did not need to make any assumptions on the grid, this method can be applied to problems on any irregularly spaced one-dimensional mesh.

Jawerth and Sweldens limited their discussion to modified linear hats on regular grids. The paper mentions that their method does allow for bases derived from higher order functions, but those would be based upon splines instead of our Lagrangian finite element space. Such wavelets would be smoother, but less compact than our Lagrangian-based higher-order extension. Thus, we were able to make a contribution in extending their method to a higher-order Lagrangian finite element space, by applying our results from section 4.1.

However, in the following section, we will continue our search for an operator-orthogonal wavelet basis that spans a truly Lagrangian finite element space of any given order. We choose to remain specifically within the Lagrangian family, because of the generality of these bases, their flexibility and their ease in implementation. Moreover, we wish to rely on well-established FEM theory (e.g. regarding convergence). In addition, the method presented in this section would not be easily extended to multi-dimensional problems.

4.2.2 Non-compact Wavelet Basis

Lifting alone can achieve decoupling across scales for any multi-resolution matrix $\mathbf{A}_J^{[J-1]}$. This property holds for any given operator, a Lagrangian finite element space of any given order, and on multi-dimensional unstructured meshes. We will refer to the property as the *decoupling power of lifting*. It can easily be verified with a dimensionality assessment. Consider the two-level multi-resolution FEM system with a traditional Hierarchical Basis of any given order, for the general PDE of Equation (4.3) (or a two-dimensional problem for that matter):

$$\begin{bmatrix} \mathbf{A}_{J-1} & \mathbf{C}_{J-1} \\ \mathbf{D}_{J-1} & \mathbf{\Delta}_{J-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_{J-1} \\ \boldsymbol{\gamma}_{J-1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{J-1} \\ \mathbf{t}_{J-1} \end{bmatrix}$$

In general, we know that $\mathbf{C}_{J-1} \neq \mathbf{0}$, and it is our goal to eliminate coupling in the multi-resolution stiffness matrix. The wavelets in a Hierarchical Basis method are all primitive wavelets, which may still be lifted with non-zero $s_{j,k,m}$. Doing so results in the following stiffness matrix, based on Equations (3.6):

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{S}_{J-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{J-1} & \mathbf{C}_{J-1} \\ \mathbf{D}_{J-1} & \mathbf{\Delta}_{J-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{S}_{J-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{J-1} & \mathbf{C}'_{J-1} \\ \mathbf{D}'_{J-1} & \mathbf{\Delta}'_{J-1} \end{bmatrix}$$

$$\begin{aligned} & \mathbf{C}'_{J-1} = \mathbf{C}_{J-1} - \mathbf{A}_{J-1}\mathbf{S}_{J-1} \\ \text{with} \quad & \mathbf{D}'_{J-1} = \mathbf{D}_{J-1} - \mathbf{S}_{J-1}\mathbf{A}_{J-1} \\ & \mathbf{\Delta}'_{J-1} = \mathbf{\Delta}_{J-1} + \mathbf{S}_{J-1}\mathbf{A}_{J-1}\mathbf{S}_{J-1} - \mathbf{S}_{J-1}\mathbf{C}_{J-1} - \mathbf{D}_{J-1}\mathbf{S}_{J-1} \end{aligned}$$

The sub-matrix $\mathbf{\Delta}'_{J-1}$ contains the wavelet-wavelet interaction of lifted wavelets (instead of the primitive wavelets), and sub-matrix $\mathbf{C}'_{J-1}$ is the scale-coupling we now have to eliminate. Since $\mathbf{A}_{J-1}$ has full rank (it is a finite element stiffness matrix in its own right), we are guaranteed to find a lifting coefficient matrix $\mathbf{S}_{J-1}$ that will make $\mathbf{C}'_{J-1} = \mathbf{0}$, explicitly $\mathbf{S}_{J-1} = \mathbf{A}_{J-1}^{-1}\mathbf{C}_{J-1}$. However, in general $\mathbf{S}_{J-1}$ will not be sparse (except in the Poisson case of paragraph 4.1). The corresponding lifted wavelets are operator-orthogonal to all scaling functions with respect to the problem's operator, but are in general not compact, boasting support over the entire domain (see e.g. Figure 4-6).

In wavelet literature, non-compact basis functions have been proposed to achieve basis orthogonality (i.e. operator-orthogonality with respect to the unity operator, scale-decoupling the mass-matrix). For this special case, a truncation of the lifted wavelets can produce wavelets that are relatively compact and approximately orthogonal (e.g. Lounsbery *et al.*, 1997). Indeed, the perfectly operator-orthogonal wavelets have a fast decay away from the primitive wavelet. Consequently, the error incurred by enforcing a compact wavelet support is small enough for their use in wavelet pre-conditioners for multi-resolution FE systems (e.g. Vassilevski *et al.*, 1997). It is important to point out that these methods rely on lifting only, and do not make use of stable completion.

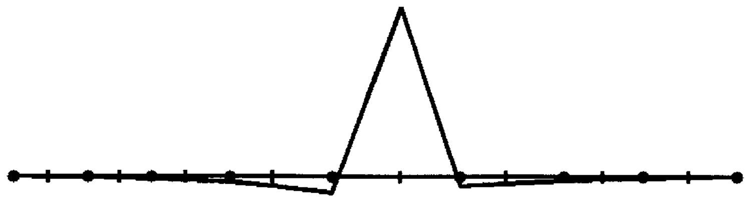


Figure 4-6: Non-compact lifting-only wavelet customized to a general operator.

As discussed in Chapter 3, we focus on building a basis of compactly supported wavelets, since only those can guarantee low-cost solution transforms. In addition, we do not want to incur truncation errors.

4.2.3 Compact Wavelet Basis

A compactly supported perfectly orthogonal wavelet basis has been built using stable completion, combined with lifting. For example, Strang and Nguyen give an example of a compact piecewise-linear one-dimensional orthogonal wavelet on a regular grid (Strang *et al.*, 1996, p.257). Further, Dahmen and Stevenson (Dahmen *et al.*, 1999) specifically use stable completion in their construction of compact wavelet bases, spanning a Lagrangian finite element space, on multi-dimensional grids. These bases, however, are not designed to be operator-orthogonal with respect to the problem's operator, but rather to be simply orthogonal and have additional vanishing moments, for the purpose of typical matrix compression for smooth functions. Unless, hypothetically speaking, the problem's stiffness matrix is the mass-matrix, using an orthogonal wavelet basis will not decouple the scales in the stiffness matrix. We propose to use the framework of lifting and stable completion to enforce operator-orthogonality with respect to *any* given operator. Then, we can take advantage of scale-decoupling in the stiffness matrix to facilitate adaptive refinement methods.

Focusing now on the one-dimensional problem specifically, we will show how we can assemble compact operator-customized wavelets as a linear combination of at most three primitive wavelets, and interior scaling functions. Indeed, consider the set of three

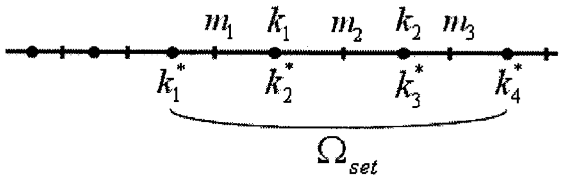


Figure 4-7: Customized wavelet's support of three first-order elements.

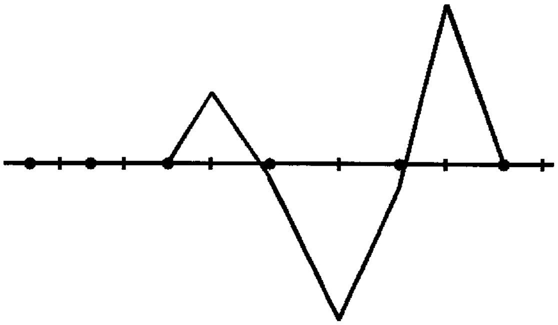


Figure 4-8: Wavelet customized to general PDE, on support of Figure 4-7.

neighboring primitive wavelets ϕ_{j+1,m_1}, ϕ_{j+1,m_2}, ϕ_{j+1,m_3} (associated with m-nodes $\mathbf{m}_{set}$). The compact set of all elements in which any of these functions has non-zero support is denoted by $\Omega_{j,set}$. We are looking to construct a compact operator-customized wavelet that falls entirely within this support. All scaling functions that overlap with $\Omega_{j,set}$ are denoted by $\phi_{j,\mathbf{k}^*_{set}}$ (associated with k-nodes $\mathbf{k}^*_{set}$). From these scaling functions, we consider the subset of functions $\phi_{j,\mathbf{k}_{set}}$ that are all interior with respect to $\Omega_{j,set}$ (associated with k-nodes $\mathbf{k}_{set}$, a subset of $\mathbf{k}^*_{set}$). In Figure 4-7, an example is given for basis functions of a first-order Lagrangian finite element space, with $\mathbf{m}_{set}=[m_1,m_2,m_3]$, $\mathbf{k}^*_{set}=[k_1,k_2,k_3,k_4]$ and $\mathbf{k}_{set}=[k_2,k_3]$. As discussed in Chapter 3, we can build wavelets using lifting and stable completion, with Equation (3.3). If we are looking to construct a wavelet within $\Omega_{j,set}$, we may use stable completion with the primitive wavelets $\phi_{j+1,\mathbf{m}_{set}}$ and lifting with the interior scaling functions $\phi_{j,\mathbf{k}_{set}}$, without leaving $\Omega_{j,set}$. Our new wavelet has to be made operator-orthogonal to only the scaling functions that overlap with $\Omega_{j,set}$. We thus need to satisfy c constraints with dof degrees of freedom. Each constraint corresponds with the enforcement of operator-orthogonality between the new wavelet and one of c overlapping scaling functions $\phi_{j,\mathbf{k}^*_{set}}$. The degrees of freedom are the weights $g_{j,\mathbf{m}_{set},m'_{set}}$ of the three primitive wavelets $\phi_{j+1,\mathbf{m}_{set}}$ and the weights $s_{j,\mathbf{k}_{set},m'_{set}}$ of the lifted scaling functions $\phi_{j,\mathbf{k}_{set}}$ that will create this new wavelet with Equation (3.3). In particular, we need to satisfy:

$$\underset{c\times dof}{\mathbf{M}_{\Omega_{j,set}}} \underset{dof\times sol}{\begin{bmatrix} s_{j,\mathbf{k}_{set},\mathbf{m}'_{set}} \\ g_{j,\mathbf{m}_{set},\mathbf{m}'_{set}} \end{bmatrix}} = \mathbf{0} \qquad \text{with} \qquad \underset{c\times dof}{\mathbf{M}_{\Omega_{j,set}}} = \Big[\underset{c\times dof}{a\left(\phi_{j,\mathbf{k}^*_{set}}, \phi_{j,\mathbf{k}_{set}}\right) \;\; a\left(\phi_{j,\mathbf{k}^*_{set}}, \phi_{j+1,\mathbf{m}_{set}}\right)} \Big]$$

$$\text{and} \quad \left| \begin{array}{l} dof = \text{length}\left(\mathbf{m}_{set}\right) + \text{length}\left(\mathbf{k}_{set}\right) \\ c = \text{length}\left(\mathbf{k}^*_{set}\right) \\ sol = \text{length}\left(\mathbf{m}'_{set}\right) \geq dof - c \end{array} \right. \tag{4.4}$$

The mere dimensions of the interaction matrix $\mathbf{M}_{\Omega_{j,set}}$ ensure that Equation (4.4) has a number of solution vectors $sol \geq dof - c$. This matrix is actually a small sub-matrix of the Hierarchical Basis stiffness matrix $\mathbf{A}_J^{[J-1]}$ (where wavelets are primitive). The appropriate $g_{j,\mathbf{m}_{set},m'_{set}}$ and $s_{j,\mathbf{k}_{set},m'_{set}}$ per solution m'_{set} can be found by computing the null-space of the interaction matrix $\mathbf{M}_{\Omega_{j,set}}$. The dimension of the null-space determines the number of linearly independent solutions sol. Irrespective of the order, n, or the operator, this local problem will always have at least one non-trivial solution, resulting in at least one operator-orthogonal wavelet within the support of $\Omega_{j,set}$. Indeed, because of the geometry and our well-made choice of a set of three primitive wavelets, we are guaranteed to find:

$$\left. \begin{array}{l} \text{length}\left(\mathbf{k}^*_{set}\right) - \text{length}\left(\mathbf{k}_{set}\right) = 2 \\ \text{length}\left(\mathbf{m}_{set}\right) = 3 \end{array} \right\} \Rightarrow dof - c = 1$$

Note that if we had chosen only one or two primitive wavelets for stable completion, we would not have been guaranteed a solution. For a wavelet customized to a general given operator, we effectively need to add one primitive wavelet for each lifting tail we want to cut off, to create a truly compact wavelet. For our first-order Lagrangian example of Figure 4-7, we can check that we have at least one solution within $\Omega_{j,set}$:

$$\begin{array}{l} dof = \text{length}\left(\mathbf{m}_{set}\right) + \text{length}\left(\mathbf{k}_{set}\right) = 5 \\ c = \text{length}\left(\mathbf{k}^*_{set}\right) = 4 \\ sol = \text{length}\left(\mathbf{m}'_{set}\right) \geq dof - c = 1 \end{array} \qquad \text{with} \qquad \begin{array}{l} \mathbf{m}_{set} = \left[m_1, m_2, m_3\right] \\ \mathbf{k}_{set} = \left[k_2, k_3\right] \\ \mathbf{k}^*_{set} = \left[k_1, k_2, k_3, k_4\right] \end{array}$$

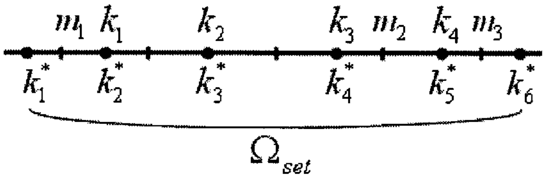


Figure 4.9: Customized wavelet's support of more than three first-order elements.

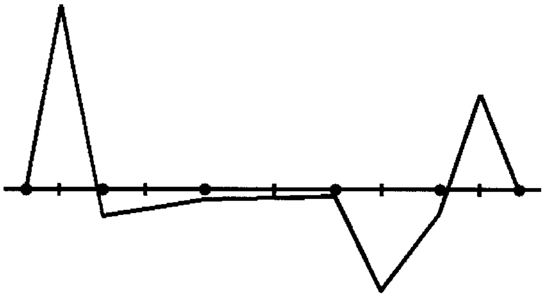


Figure 4.10: Wavelet customized to general PDE, on support of Figure 4.9.

In Figure 4-8, we show an example of such a compact wavelet customized to a given general operator.

What happens if the m-nodes associated with the three primitive wavelets are not adjacent (see Figure 4-9)? In that case, we make sure that $\Omega_{j,set}$ contains the three primitive m-nodes without any gaps, and we look for a wavelet that has a compact continuous support over $\Omega_{j,set}$. For every additional constraint imposed by an overlapping scaling function, we will have one more lifting degree of freedom. Indeed, every additional k-node added to $\mathbf{k}_{set}^*$ is added to $\mathbf{k}_{set}$ as well. Thus, we are again guaranteed to find at least one operator-orthogonal wavelet, made up from the three primitive wavelets and scaling functions, within this less compact support (see Figure 4-10).

It is important to note that a compact customized wavelet $\psi_{j,m'}$ will be also operator-orthogonal to a non-compact wavelet $\psi_{j,m''}$, if its support is entirely contained within the lifted (non-primitive) part of this non-compact wavelet. Indeed, in that area, the non-compact wavelet is only composed of scaling functions. Thus, also a *non-compact* operator-orthogonal wavelet will be 'compact' in terms of its interaction with all *compact* customized wavelets in the interaction matrix $\mathbf{\Delta}_j$. Indeed, we have:

$$
\begin{aligned}
&a\left(\psi_{j,m'},\psi_{j,m''}\right)\\
&=\sum_{m} g_{j,m,m''}\, a\left(\psi_{j,m'},\phi_{j+1,m}\right)-\sum_{k} s_{j,k,m''}\, a\left(\psi_{j,m'},\phi_{j,k}\right)\\
&=\sum_{m} g_{j,m,m''}\, a\left(\psi_{j,m'},\phi_{j+1,m}\right) \qquad \text{because} \quad a\left(\psi_{j,m'},\phi_{j,k}\right)=0\\
&=0 \qquad \text{because } \psi_{j,m'} \text{ does not overlap with any } \phi_{j+1,m}
\end{aligned}
$$

However, a non-compact customized wavelet will be in general not operator-orthogonal to another non-compact wavelet. For example, the wavelets discussed in paragraph 4.2.2 would yield a non-sparse detail matrix $\mathbf{\Delta}_j$ unless they are truncated. Thus, having one single non-compact wavelet among a set of compact wavelets is much more convenient than having several non-compact wavelets. Indeed, in that case we do not have to rely on truncation to achieve a sparse scale-decoupled stiffness matrix.

4.2.4 Special Operators

Now that we have shown how to build wavelets that are operator-orthogonal with respect to any given operator, we will make the connection between the general operator case and the Laplace operator case. Why have the Laplace-customized wavelets from paragraph 4.1 a smaller support than wavelets customized to a more general operator? This is due to the *inheritance of vanishing moments* property, discussed in paragraph 4.1. A key element in the construction of a wavelet basis customized to the Laplace operator was the constraint on the wavelet's support $\Omega_{j,set}$. Indeed, keeping $\Omega_{j,set}$ limited to one element $\Omega_{j,el}$ guaranteed pure polynomial scaling functions over the wavelet's support. However, if we increase the wavelet's support beyond one element – to deal with general operators, as we did in paragraph 4.2.3 – scaling functions cannot anymore be assumed pure polynomial over the customized wavelet's support $\Omega_{j,set}$. Nevertheless, we are always able to build pure polynomials over $\Omega_{j,set}$, by taking simple linear combinations of the scaling functions, spanning a Lagrangian finite element space of a given order n. Indeed, away from the boundary, we can build – using all scaling functions associated with $\mathbf{k}^{*}_{set}$ – any pure polynomial of order n over the entire wavelet support $\Omega_{j,set}$. This means that we can find $pol = n+1$ linearly independent linear combinations of scaling functions, each determined by a vector $b_{j,\mathbf{k}^*_{set},n}$, satisfying on $\Omega_{j,set}$:

$$\underset{1\times pol}{\left[\sum_{\mathbf{k}^*_{set}} \phi_{j,k_i} b_{j,k_i,n}\right]_{\Omega_{j,set}}} = \underset{1\times pol}{\left[x^0 \quad \dots \quad x^n\right]_{\Omega_{j,set}}} \quad \text{and} \quad pol = n+1$$

This property directly affects the rank of the interaction matrix $\mathbf{M}_{\Omega_{j,set}}$ for the wavelets within this support (see Equation (4.4)) customized to the Laplace operator in particular. To show this, let us first make an adjustment to this matrix. Indeed, we know that in order to benefit from higher-order vanishing moments in the wavelet's derivative, we have to enforce vanishing moments in the wavelet itself. Thus, we add to $\mathbf{M}_{\Omega_{j,set}}$ $n-1$ rows, corresponding with constraints that enforce $van = n-1$ vanishing moments on all customized wavelet solutions. This yields the following problem:

$$\underset{(c+van)\times dof}{\mathbf{M}_{\Omega_{j,set}}} \underset{dof\times sol}{\begin{bmatrix} s_{j,\mathbf{k}_{set},\mathbf{m}'_{set}} \\ g_{j,\mathbf{m}_{set},\mathbf{m}'_{set}} \end{bmatrix}} = \mathbf{0} \qquad \text{with} \qquad \underset{(c+van)\times dof}{\mathbf{M}_{\Omega_{j,set}}} = \underset{(c+van)\times dof}{\begin{bmatrix} a\left(\phi_{j,\mathbf{k}^*_{set}}, \phi_{j,\mathbf{k}_{set}}\right) & a\left(\phi_{j,\mathbf{k}^*_{set}}, \phi_{j+1,\mathbf{m}_{set}}\right) \\ \left\langle \mathbf{x}, \phi_{j,\mathbf{k}_{set}} \right\rangle & \left\langle \mathbf{x}, \phi_{j+1,\mathbf{m}_{set}} \right\rangle \end{bmatrix}}$$

$$\text{where} \quad \mathbf{x} = \begin{bmatrix} 1x^0 \\ \vdots \\ (n-1)x^{n-2} \end{bmatrix} \quad \text{and} \quad \left| \begin{array}{l} dof = \text{length}\left(\mathbf{m}_{set}\right) + \text{length}\left(\mathbf{k}_{set}\right) \\ c = \text{length}\left(\mathbf{k}^*_{set}\right) \\ van = n-1 \\ sol = \text{length}\left(\mathbf{m}'_{set}\right) \geq dof - \left(c + van\right) \end{array} \right.$$

Now, we can apply the *inheritance of vanishing moments* principle. Because primitive wavelets $\phi_{j+1,\mathbf{m}_{set}}$ and interior scaling functions $\phi_{j,\mathbf{k}_{set}}$ all lie entirely within $\Omega_{j,set}$, we know that each is Laplace-operator-orthogonal to both x^0 and x^1. In addition, any higher-order inner-product with respect to the operator, up to x^n, (e.g. $a\left(x^n, \phi_{j+1,\mathbf{m}_{set}}\right)$) can be eliminated with a moment of this function (e.g. $\phi_{j+1,\mathbf{m}_{set}}$) from one of the additional rows in $\mathbf{M}_{\Omega_{j,set}}$:

$$\int_\Omega \frac{\partial x^n}{\partial x} \frac{\partial \phi_{j+1,\mathbf{m}_{set}}}{\partial x} dx = n x^{n-1} \phi_{j+1,\mathbf{m}_{set}} \Big|_{x_1}^{x_2} - n \int_\Omega (n-1) x^{n-2} \phi_{j+1,\mathbf{m}_{set}} dx$$

$$\Rightarrow \int_\Omega \frac{\partial x^n}{\partial x} \frac{\partial \phi_{j+1,\mathbf{m}_{set}}}{\partial x} dx + n \int_\Omega (n-1) x^{n-2} \phi_{j+1,\mathbf{m}_{set}} dx = 0$$

And consequently:

$$\underset{dof\times(c+van)}{\mathbf{M}^{T}_{\Omega_{j,set}}}\ \underset{(c+van)\times pol}{\mathbf{B}} = \mathbf{0} \qquad \text{with} \quad \mathbf{B} = \underset{(c+van)\times pol}{\begin{bmatrix} b_{\mathbf{k}^{\bullet}_{set},\mathbf{n}} \\ \mathbf{L} \end{bmatrix}} \quad \text{and} \quad \mathbf{L} = \underset{van\times pol}{\begin{bmatrix} 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & n \end{bmatrix}} \tag{4.5}$$

We can check that the rank of $\mathbf{B}$ equals $pol = n+1$. This allows us to eliminate $pol = n+1$ constraints from our problem defined by $\mathbf{M}_{\Omega_{j,set}}$, but we had to first add $van = n-1$ vanishing moment constraints. Thus, for the Poisson's Equation we will find $pol - van = 2$ more operator-orthogonal wavelets, compactly supported within $\Omega_{j,set}$, than for a general non-Laplace operator. We will find at least $dof - c + 2$ Laplace-customized wavelets within $\Omega_{j,set}$. For example, for first-order Lagrangian scaling functions ($n = 1$), over three elements $\Omega_{j,set}$ as in Figure 4-7, we will find $5-4+2=3$ Laplace-operator-orthogonal wavelets within that support, in contrast to just $5-4=1$ wavelet customized to a more general operator. For a Lagrangian finite element space of any given order *n*, limiting the support to just one element $\Omega_{j,el}$ as in Figure 4-1, always yields $dof - c + 2 = (2n-1)-(n+1)+2 = n$ Laplace-customized wavelets per element. This is consistent with our findings in paragraph 4.1 of a Laplace-operator-orthogonal wavelet basis, with all wavelets $\psi_{j,m}$ remaining within their respective element. Thus, we have shown how the construction of a Laplace-customized wavelet basis can be regarded as a very special case of a general construction method.

Note that the construction of higher-order Laplace-customized wavelets does not strictly require adding vanishing moment constraints to the interaction matrix $\mathbf{M}_{\Omega_{j,set}}$. Equation (4.5) with $b_{j,\mathbf{k}^{\bullet}_{set},n}$ corresponding to only x^0 and x^1, and without $\mathbf{L}$, would yield the same (number of) solutions. Indeed, the gain for a Laplace operator over a non-Laplace operator is $(n+1)-(n-1)=2$, irrespective of the order n. The null-space of the interaction matrix $\mathbf{M}_{\Omega_{j,set}}$ without vanishing moment constraints equals the null-space of $\mathbf{M}_{\Omega_{j,set}}$ from Equation (4.4). However, the longer derivation will be relevant for the two-dimensional problem and does illustrate the generality of the approach. In addition, we can use this derivation to show that the higher-order wavelets customized to the one-dimensional Laplace operator will have a specific number of vanishing moments. Note also that a similar derivation can be done for pure orthogonality (i.e. with respect to the identity operator), as opposed to operator-orthogonality with respect to the Laplace operator. In that case, however, we have to exactly add as many constraints as we wish to eliminate, even for higher-dimensional problems. Thus, we do not benefit from using the $\mathbf{B}$ matrix in the construction of an orthogonal wavelet basis.

The Poisson's Equation is not the only PDE for which we can build operator-orthogonal wavelets with less than three primitive wavelets. Indeed, consider the following Diffusion-Convection Equation, Equation (4.3) with a non-constant coefficient *p*, but with *q* constant and *r* equal to zero:

$$-\frac{\partial}{\partial x}\left(p\frac{\partial u}{\partial x}\right)+\frac{\partial u}{\partial x}=f \quad \left|\begin{array}{l} u(x_1)=0 \\ u(x_2)=0 \end{array}\right.$$

Because of the partition of unity property, our basis of scaling functions can always form a constant function, away from the boundary:

$$\underset{1\times 1}{\left[\sum_{\mathbf{k}^*_{set}}\phi_{j,k_i}\,b_{j,k_i,0}\right]_{\Omega_{j,set}}}=[1]_{\Omega_{j,set}}$$

Therefore – irrespective of the properties of the wavelet and without any vanishing moment constraints enforced on $\mathbf{M}_{\Omega_{j,set}}$ – we know that:

$$\underset{dof\times c}{\mathbf{M}^T_{\Omega_{j,set}}}\underset{c\times 1}{\mathbf{B}}=\underset{dof\times c}{\mathbf{M}^T_{\Omega_{j,set}}}\underset{c\times 1}{\left[b_{j,\mathbf{k}^*_{set},0}\right]}=\begin{bmatrix} a\left(\phi_{j,\mathbf{k}_{set}},1\right) \\ a\left(\phi_{j+1,\mathbf{m}_{set}},1\right)\end{bmatrix}=\mathbf{0}$$

$$\text{because}\quad a(\varphi,1)=\int_\Omega\left(p\frac{\partial\varphi}{\partial x}\frac{\partial 1}{\partial x}+\frac{\partial\varphi}{\partial x}1\right)dx=\varphi\big|_{x_1}^{x_2}=0$$

We can check that the rank of $\mathbf{B}$ equals one. Thus, we will find at least $dof-c+1$ wavelets customized to the Diffusion-Convection operator within $\Omega_{j,set}$. Hence, for a Lagrangian finite element space of any given order *n*, combining two primitive wavelets and lifting with only interior scaling functions, will yield at least $dof-c+1=1$ compact wavelet customized to this special operator.

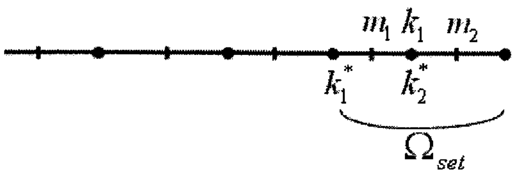


Figure 4-11: Customized wavelet's support adjacent to a Dirchlet boundary.

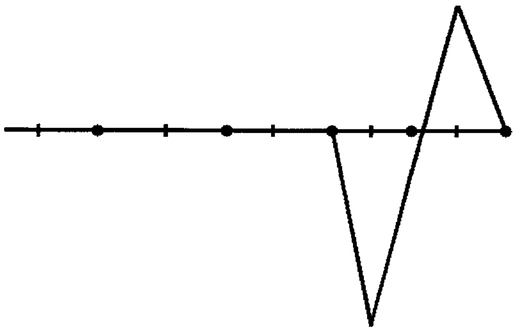


Figure 4-12: Wavelet customized to general PDE, on support of Figure 4-11.

4.2.5 Boundary Treatment

What happens at the boundary? As we discussed in paragraph 4.1, all wavelets near a boundary will be forced to be zero-valued at a Dirichlet boundary. This means that we cannot lift with Dirichlet boundary k-nodes. As a result, Dirichlet boundary conditions will be only carried by the boundary scaling functions. Consequently, we can solve for these scaling coefficients $\lambda_{j,\mathbf{k}_\Gamma}$ independently of the rest of the system, and eliminate all coupling between the scaling functions and the rest of the system. Thus, we do not need to make our wavelets operator-orthogonal to these boundary scaling functions. In effect, we take every $k_{\Gamma Dirichlet}$ out of both $\mathbf{k}_{set}$ and $\mathbf{k}^*_{set}$. Neumann boundaries are dealt with differently. We do allow wavelets to be lifted with scaling functions located on the Neumann boundary, and also force wavelets to be operator-orthogonal to those scaling functions. Hence, all $k_{\Gamma Neumann}$ are left in both $\mathbf{k}_{set}$ and $\mathbf{k}^*_{set}$. In conclusion, when $\Omega_{j,set}$ has one boundary, Dirichlet or Neumann, we can build at least one operator-orthogonal wavelet with only two primitive wavelets and the interior scaling functions (see e.g. Figures 4-11 and 4-12):

$$\left.\begin{array}{l}\text{length}\left(\mathbf{k}^*_{set}\right)-\text{length}\left(\mathbf{k}_{set}\right)=1\\ \text{length}\left(\mathbf{m}_{set}\right)=2\end{array}\right\}\Rightarrow dof-c=1$$

If $\Omega_{j,set}$ contains two boundaries, we can find at least one operator-orthogonal wavelet by lifting just one primitive wavelet. This is consistent with the non-local operator-orthogonal wavelet with support extending all over the domain, discussed in paragraph 4.2.2, shown to exist by the *decoupling power of lifting* property (see e.g. Figure 4-6).

For the special case analysis of the Laplace operator or Diffusion-Convection operator, presented in paragraph 4.2.4, the presence of a Dirichlet boundary will reduce the number of pure polynomials formed by the scaling functions. In effect, since the scaling functions on the Dirichlet boundary are not part of $\mathbf{k}_{set}^{*}$ (nor $\mathbf{k}_{set}$), they may not be included in the formation of pure polynomials over the wavelet's support. This results in a lesser rank of $\mathbf{B}$. For the Laplace operator, we have an effective gain of one extra solution for two primitive wavelets next to the Dirichlet boundary, while for the special Diffusion-Convection operator, we have no gain next to the Dirichlet boundary. Note that, alternatively, we can choose to add the Dirichlet constraint to $\mathbf{k}_{set}^{*}$, while preserving the number of pure polynomials, effectively a net-net situation.

Consequently, irrespective of the boundary, wavelets customized to the Laplace operator can always be build from only one primitive wavelet and interior scaling functions, while wavelets customized to the Diffusion-Convection operator can be build from only two primitive wavelet and interior scaling functions. Wavelets customized to more general operators will be constructed, within the domain, from three primitive wavelets and interior scaling functions, and from two primitive wavelets and interior scaling functions, next to a boundary.

4.3 Implementation

One important issue we have not yet addressed is how to guarantee an invertible stable completion matrix $\mathbf{G}_j$. Indeed, the construction of wavelets with only lifting guarantees the preservation of a full basis, whereas with stable completion this is conditional upon the invertibility of $\mathbf{G}_j$. If we fail to construct our wavelets by grouping primitive wavelets in a fully invertible operation, the wavelets will not anymore span the Lagrangian finite element space. This is of significant practical importance, because all wavelets are to be constructed and customized on-the-fly, by selecting local supports

$\Omega_{j,set}$, with appropriate $\mathbf{m}_{set}$, $\mathbf{k}_{set}$ and $\mathbf{k}^*_{set}$. Even though we may ultimately apply an adaptive solution method and use only few of the wavelets, we want to be sure that with all wavelets included we would obtain exactly the same solution as with a single-scale Lagrangian FEM.

Away from the boundary, in zones of the domain where the operator does not switch between the general and special cases, the compact operator-customized wavelets described in paragraphs 4.2.3 and 4.2.4 are guaranteed to form a full basis. For the general case and the case of the Poisson's Equation, the wavelet customization at the boundary, discussed in paragraph 4.2.5, completes this wavelet basis perfectly. However, for the special case of the Diffusion-Convection Equation, the boundaries force one of all customized wavelets to be non-compact. Indeed, because both the compact customized wavelets in the domain and the wavelets adjacent to the boundary are formed with two neighboring primitive wavelets (and interior scaling functions), we would be one wavelet short to form a complete basis. That extra wavelet can be taken as any of the primitive wavelets, lifted with all the scaling functions of the domain. Such non-compact customized wavelet has been discussed in paragraph 4.2.2. Though we can choose any of the primitive wavelets for this additional customized wavelet, for an adaptive method we would likely choose a primitive wavelet away from the zone with large solution error. A non-compact operator-orthogonal wavelet will still be 'compact' in terms of the interaction matrix $\mathbf{\Delta}_j$, as noted in paragraph 4.2.3, as long as its primitive support is compact and all other customized wavelets are compact. The non-compact wavelet for the Diffusion-Convection Equation is constructed from only one primitive wavelet, and it is the only non-compact wavelet in the basis. In addition, its function value decays fast, away from these primitive components. Thus, we expect this wavelet to be only included in an adaptive basis when the error concentration is high near these primitive components.

Furthermore, changes in operator could also cause some customized wavelets to be non-compact. For example, a zone of the domain governed by the Laplace operator,

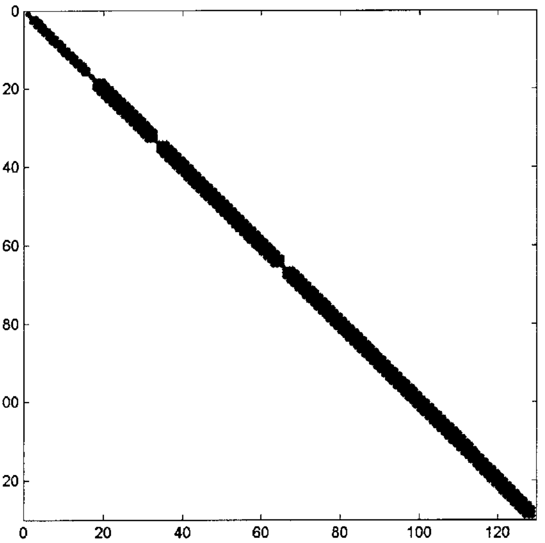


Figure 4-13: Four-level FE matrix of first-order wavelets customized to a general PDE.

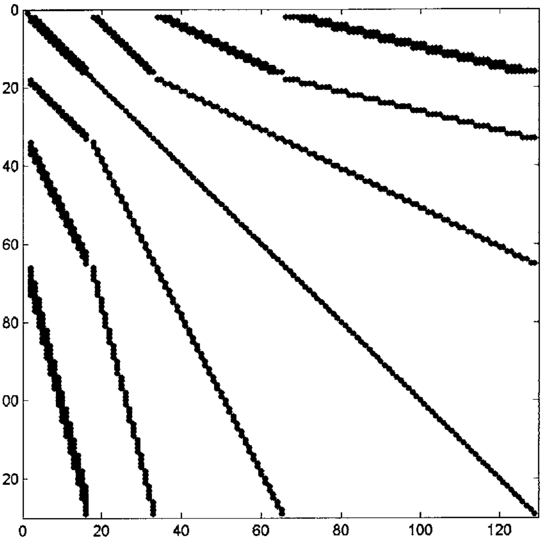


Figure 4-14: Four-level FE matrix of first-order HB for a general PDE.

surrounded by a zone governed by a more general operator, would yield two non-compact wavelets. Each would span the Laplace zone with scaling function components, and have two primitive wavelet components at one side of the Laplace zone and one primitive wavelet at the other side.

Examples of compact operator-orthogonal wavelets are given in Figure 4-1 (for the Laplace operator) and Figures 4-8, 4-12 (for more general operators). The consequence of applying such an operator-customized wavelet basis to the FEM analysis of a general PDE, is shown in the stiffness matrix of Figure 4-13. The stiffness matrix is fully decoupled across all scales. As shown in Figure 4-14, this is not the case when using a Hierarchical Basis for the same non-Poisson PDE.

We know from the analysis above that the support needed for each customized wavelet depends on the operator, and on possible boundary contact. We propose an implementation strategy that guarantees a full set of operator-orthogonal wavelets, independent of the type of operator (allowing even a combination of different operators over the domain) or the Lagrangian order n. Every wavelet will be the most compactly supported achievable. The proposed algorithm works as follows. First, go over every individual primitive wavelet, compute the null-space of the appropriate system, and if a solution exists assign the corresponding *m*-node to a newly added wavelet. Eliminate all primitive wavelets assigned in the first round. Second, go over every pair of primitive wavelets while skipping assigned *m*-nodes, compute the null-space of the appropriate system, and if a solution exists assign one of the corresponding *m*-nodes to a newly added wavelet. Eliminate all primitive wavelets assigned in the second round. Third, go over every trio of primitive wavelets while skipping assigned *m*-nodes, compute the null-space of the appropriate system, and if a solution exists assign one of the corresponding *m*-nodes to a newly added wavelet. All wavelets have been found and assigned. The algorithm can be written as follows:

```
$\mathbf{m}_{assigned} = []$
for iter1,2,3 do:
  $\mathbf{m}_{usable} = \mathbf{m}_{all} \setminus \mathbf{m}_{assigned}$
  for all $m_i \in \mathbf{m}_{usable}$ do:
      case iter1: $\mathbf{m}_{set} = m_i$
      case iter2: $\mathbf{m}_{set} = [m_i, m_{i+1}]$  $(\text{if } m_{i+1} \notin \mathbf{m}_{usable} \text{ return})$
      case iter3: $\mathbf{m}_{set} = [m_{i-1}, m_i, m_{i+1}]$  $(\text{if } m_{i-1}, m_{i+1} \notin \mathbf{m}_{usable} \text{ return})$
      $\Omega_{j,mset} = \{\Omega_{j,el} \mid m \in \mathbf{m}_{set}, m \in \Omega_{j,el}\}$
      $\Omega_{j,set} = \Omega_{j,mset} \cup \{\Omega_{j,el} \mid \Omega_{j,el} \text{ is surrounded by } \Omega_{j,mset}\}$
      $\mathbf{k}_{set} = \{k \mid k \in \Omega_{j,set}, k \notin \Omega_{j,total\setminus set}, k \notin \Gamma_{j,Dirichlet}\}$
      $\mathbf{k}^*_{set} = \{k \mid k \in \Omega_{j,set}, k \notin \Gamma_{j,Dirichlet}\}$
      solve: $\begin{bmatrix} a(\phi_{j,\mathbf{k}^*_{set}}, \phi_{j,\mathbf{k}_{set}}) & a(\phi_{j,\mathbf{k}^*_{set}}, \phi_{j+1,\mathbf{m}_{set}}) \end{bmatrix} \begin{bmatrix} s_{j,\mathbf{k}_{set}} \\ g_{j,\mathbf{m}_{set}} \end{bmatrix} = \mathbf{0}$
      if solution exist do: $\mathbf{m}_{assigned} = \mathbf{m}_{assigned} \cup m_i$
```

Note that, although this procedure requires for each level three passes over the domain, it still remains $O(N)$. For adaptive methods this algorithm could be applied in a modified version.

If we do know at start the specific type of operator and boundaries, we can directly solve for wavelets on the appropriate supports, with no need for the algorithm described above. Moreover, in that case, we may fix the stable completion coefficient of one of the primitive wavelets (e.g. the middle one) to a non-zero value (e.g. one). By doing so, we only have for each customized wavelet a cheap determined system to solve.

4.4 Conclusion

We proposed the construction of a basis of compact wavelets that are operator-orthogonal to all scaling functions, with respect to the Laplace operator, for a Lagrangian space of any given order. We based this construction on a property we called the inheritance of vanishing moments. After that, we proposed the construction of a basis of compact higher-order wavelets, operator-orthogonal with respect to a more general operator. Unfortunately, these bases do not span anymore a Lagrangian finite element space, and are not easily extended to multi-dimensional problems. Then, after assessing the option of building non-compact operator-orthogonal wavelets, we proposed the construction of a basis of compact wavelets that are made operator-orthogonal with respect to any given operator. We show how the Laplace operator is a special case that allows for slightly more compact operator-orthogonal wavelets. Finally, we discuss an implementation strategy that takes into account this difference in support.

With the techniques presented in this chapter in mind, we will now address the slightly more complicated task of constructing a basis of compact operator-customized wavelets for two-dimensional problems.

5. 2D Wavelet Customization

5.1 Poisson's Equation

Now that we can construct a basis of compact wavelets customized to a one-dimensional operator, we will show how to customize wavelets to a two-dimensional operator. A compact operator-orthogonal wavelet basis will yield a sparse scale-decoupled FE stiffness matrix for problems governed by this operator. Before handling more general second-order PDEs, let us first start with the two-dimensional Poisson's Equation over the domain Ω:

$$-\nabla^2 u = f \quad \text{with} \quad u|_\Omega = 0 \tag{5.1}$$

For simplicity, we assume for now homogeneous Dirichlet boundary conditions all around the boundary. We choose a Lagrangian finite element space of a given order n for both trial functions φ_k and test functions $\varphi_{k'}$:

$$\mathbf{A}\,\mathbf{u} = \mathbf{f}$$

$$\mathrm{A}[\mathrm{k',k}] = a(\varphi_k, \varphi_{k'}) = \iint_{\Omega} \left(\frac{\partial \varphi_k}{\partial x} \frac{\partial \varphi_{k'}}{\partial x} + \frac{\partial \varphi_k}{\partial y} \frac{\partial \varphi_{k'}}{\partial y} \right) dxdy$$
$$\mathrm{f}[\mathrm{k'}] = b(\varphi_{k'}) = \iint_{\Omega} f \varphi_{k'} dxdy$$

The mesh, made up of triangles, can be irregular-spaced and even unstructured, as long as the FEM problem is well-conditioned. We refer to Chapter 2 and Chapter 3, for details on acceptable meshes. We now can choose for φ_k a multi-resolution wavelet basis instead of a single-scale basis, both spanning the order n Lagrangian finite element space. We will use wavelet theory to construct a Lagrangian basis of compactly supported wavelets that are operator-orthogonal to all scaling functions with respect to the two-dimensional Laplace operator. That is, we wish to satisfy the following condition for *operator orthogonality* (with respect to the Laplace operator), with compactly supported $\psi_{j,m}$:

$$C_{j,k,m} = a(\psi_{j,m}, \phi_{j,k}) = \iint_{\Omega} \left(\frac{\partial \psi_{j,m}}{\partial x} \frac{\partial \phi_{j,k}}{\partial x} + \frac{\partial \psi_{j,m}}{\partial y} \frac{\partial \phi_{j,k}}{\partial y} \right) dxdy = 0$$
$$\forall m,k \text{ and } 0 \le j \le J-1$$

In Chapter 2, paragraph 2.1, we saw that the two-dimensional Hierarchical Basis – a basis of primitive wavelets, without lifting or stable completion (see e.g. Figure 2-11 or Figure 3-7) – lacks this quality. Even the linear hat wavelets are not naturally operator-orthogonal to the coarser basis functions, with respect to the two-dimensional Laplace operator. However, we can use lifting and stable completion to customize the wavelets to the operator. In particular, the reader can verify that the following wavelet equations yield compact first-order Lagrangian wavelets that are customized to the two-dimensional Laplace operator, on a regular-spaced mesh of right triangles, away from the boundary:

$$\begin{bmatrix} \psi_{j,k(1)} \\ \psi_{j,k(2)} \\ \psi_{j,k(3)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \phi_{j+1,m_1} \\ \phi_{j+1,m_2} \\ \phi_{j+1,m_3} \\ \phi_{j+1,m_4} \\ \phi_{j+1,m_5} \\ \phi_{j+1,m_6} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \phi_{j,k} \qquad (5.2)$$

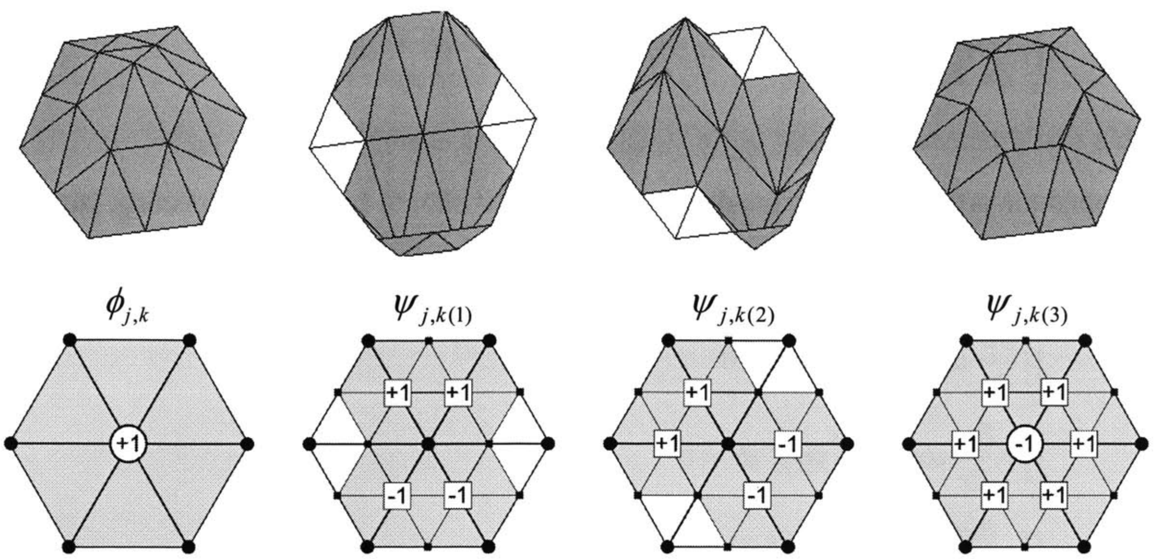


Figure 5-1: Scaling function, and wavelets customized to Laplace operator.

The wavelets of Equation (5.2) are shown in Figure 5-1. Note that we build three wavelets around each *k*-node, instead of one wavelet per *m*-node. The wavelets have a slightly larger support than the primitive wavelets shown in Figure 3-7, but still fall within the support of one scaling function, which is the minimal support possible. Because, away from the boundary, we always have three times the number of *m*-nodes as *k*-nodes (each triangle of level j is subdivided into four triangles on level $j+1$), we find that these wavelets span the full solution space within the domain.

Before presenting the details on the construction of compact Laplace-customized wavelets on an irregular-spaced mesh, we first recognize the following two-dimensional generalization of the *inheritance of vanishing moments* property for compactly supported wavelets:

$$\iint_{\Omega} \frac{\partial \psi}{\partial x} dxdy = \int \psi \Big|_{x_1}^{x_2} dy = 0$$

$$\iint_{\Omega} \frac{\partial \psi}{\partial y} dxdy = \int \psi \Big|_{y_1}^{y_2} dx = 0$$

Thus, a compactly supported wavelet would be operator-orthogonal to any plane $t = a_{0,0} + a_{1,0}x + a_{0,1}y$. Indeed:

$$\iint_\Omega \frac{\partial\psi}{\partial x}\frac{\partial f}{\partial x}+\frac{\partial\psi}{\partial y}\frac{\partial f}{\partial y}dxdy = a_{1,0}\iint_\Omega \frac{\partial\psi}{\partial x}dxdy + a_{0,1}\iint_\Omega \frac{\partial\psi}{\partial y}dxdy = 0 \tag{5.3}$$

This easily extends to higher order functions t. For example, for a compact wavelet to be operator-orthogonal to a quadratic function t, it should have one vanishing moment (i.e. zero integral), so that the first derivatives of the wavelet are orthogonal to any plane:

$$\iint_\Omega \frac{\partial\psi}{\partial x}dxdy = \int \psi\Big|_{x_1}^{x_2} dy = 0$$

$$\iint_\Omega y\frac{\partial\psi}{\partial x}dxdy = \int y\psi\Big|_{x_1}^{x_2} dy = 0$$

$$\iint_\Omega x\frac{\partial\psi}{\partial x}dxdy = \int x\psi\Big|_{x_1}^{x_2} dy - \iint_\Omega \psi\, dxdy = 0 \quad\Leftrightarrow\quad \iint_\Omega \psi\, dxdy = 0$$

Here we have only shown the conditions on the x-derivative; the conditions on the y-derivative are satisfied likewise. Similarly, a cubic function t requires the additional constraints (again shown only for the x-derivative):

$$\iint_\Omega x^2\frac{\partial\psi}{\partial x}dxdy = \int x^2\psi\Big|_{x_1}^{x_2} dy - \iint_\Omega 2x\psi\, dxdy = 0 \quad\Leftrightarrow\quad \iint_\Omega x\psi\, dxdy = 0$$

$$\iint_\Omega y^2\frac{\partial\psi}{\partial x}dxdy = \int y^2\psi\Big|_{x_1}^{x_2} dy = 0$$

$$\iint_\Omega xy\frac{\partial\psi}{\partial x}dxdy = \int xy\psi\Big|_{x_1}^{x_2} dy - \iint_\Omega y\psi\, dxdy = 0 \quad\Leftrightarrow\quad \iint_\Omega y\psi\, dxdy = 0$$

These constraints also satisfy the conditions on the wavelet's derivatives in y. We thus see that a compactly supported wavelet with one vanishing moment is operator-orthogonal to any quadratic $t = a_{0,0} + a_{1,0}x + a_{0,1}y + a_{1,1}xy + a_{2,0}x^2 + a_{0,2}y^2$. Similarly, a compactly supported wavelet with three vanishing moments (1, x and y) is operator-orthogonal to any cubic. The inheritance of vanishing moments property was key in finding compact operator-orthogonal Lagrangian wavelet bases for the one-dimensional Poisson's Equation, and is again key here. Note that in the two-dimensional case, the scaling functions are often *piecewise* polynomial over the support of the primitive wavelet (see e.g. Figure 3-7), whereas in the one-dimensional case, the scaling functions were *pure* polynomial over the support of the primitive wavelet (see Figure 4-1).

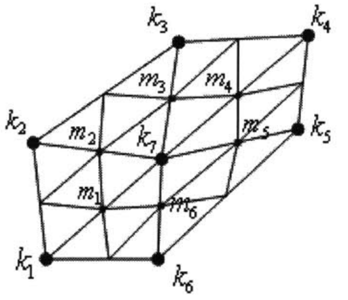


Figure 5-2: Customized wavelet's support of one scaling function.

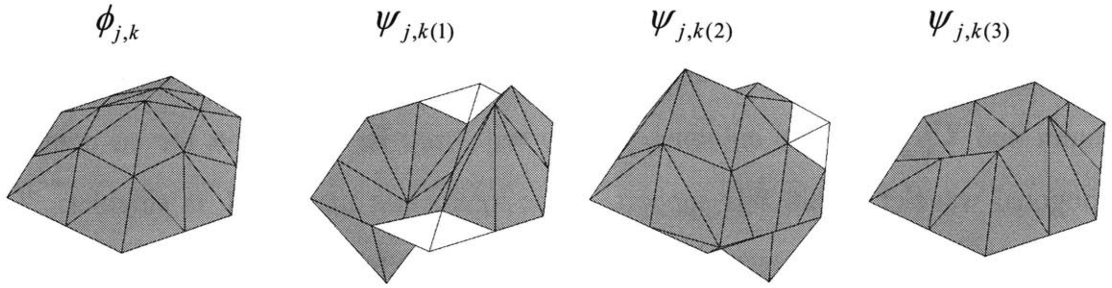


Figure 5-3: Scaling function, three wavelets customized to Laplace, on support of Figure 5-2.

However, just as we did in paragraph 4.2.3 for supports of multiple one-dimensional elements, we will effectively build pure polynomials t with linear combinations of scaling functions.

Let us go back now to the customization of first-order Lagrangian wavelets to the Laplace-operator on a two-dimensional irregular-spaced mesh. We will focus in this paragraph on the clear-cut first-order Lagrangian basis functions, and will leave the discussion of higher-order Lagrangian wavelet bases customized to the Laplace operator to paragraph 5.2.3. For example, consider the irregular-spaced mesh given in Figure 5-2 for a first-order Lagrangian finite element space. If we are looking to build an operator-orthogonal wavelet that lies entirely within the support of scaling function ϕ_{j,k_7}, we need to satisfy seven constraints with seven degrees of freedom. The constraints are that we have operator-orthogonality between the wavelet and the seven overlapping scaling functions ϕ_{j,k_1} to ϕ_{j,k_7}. We denote the set of nodes $[k_1,k_2,k_3,k_4,k_5,k_6,k_7]$ as $\mathbf{k}^*$. The degrees of freedom are the weights of the six primitive wavelets (located at the nodes

$\mathbf{m} = [m_1, m_2, m_3, m_4, m_5, m_6]$) and the lifted scaling function (on k_7) in Equation (3.3). Thus, we need to satisfy:

$$\underset{7\times 7}{\mathbf{M}}\begin{bmatrix} -s_{j,k_7,m'} \\ g_{j,\mathbf{m},m'} \end{bmatrix} = \mathbf{0} \quad \text{with} \quad \underset{7\times 7}{\mathbf{M}} = \underset{7\times 7}{\begin{bmatrix} A_{\mathbf{k}^*,k_7} & C_{\mathbf{k}^*,\mathbf{m}} \end{bmatrix}}$$

$$\text{where} \quad \begin{aligned} A_{k_i,k_7} &= \iint_\Omega \left(\frac{\partial \phi_{j,k_7}}{\partial x}\frac{\partial \phi_{j,k_i}}{\partial x} + \frac{\partial \phi_{j,k_7}}{\partial y}\frac{\partial \phi_{j,k_i}}{\partial y} \right) dxdy \\ C_{k_i,m_i} &= \iint_\Omega \left(\frac{\partial \phi_{j+1,m_i}}{\partial x}\frac{\partial \phi_{j,k_i}}{\partial x} + \frac{\partial \phi_{j+1,m_i}}{\partial y}\frac{\partial \phi_{j,k_i}}{\partial y} \right) dxdy \end{aligned} \tag{5.4}$$

Note that whereas $s_{j,k_7,m'}$ is a scalar, $g_{j,\mathbf{m},m'}$ has six rows. Similarly, $A_{\mathbf{k}^*,k_7}$ has one column and $C_{\mathbf{k}^*,\mathbf{m}}$ has six columns. There will only be a solution for an operator-orthogonal wavelet within this support, if the interaction matrix $\mathbf{M}$ is singular. The appropriate $g_{j,\mathbf{m},m'}$ and $s_{j,k_7,m'}$ per m' can then be found by computing the null-space of this matrix. To show that $\mathbf{M}$ is in fact singular, observe that we can build – using only the seven scaling functions ϕ_{j,k_1} to ϕ_{j,k_7} – any pure plane $t = a_{0,0} + a_{1,0}x + a_{0,1}y$ over the entire support of ϕ_{j,k_7}. This means that we can find three linearly independent linear combinations $b_{\mathbf{k}^*,n}$ of scaling functions that satisfy over the entire support of ϕ_{j,k_7}:

$$\underset{1\times 3}{\left[\sum_{k_i=k_1}^{k_7} \phi_{j,k_i} b_{k_i,n} \right]} = [1 \quad x \quad y] \quad n = 1,2,3 \tag{5.5}$$

Because ϕ_{j+1,m_1} to ϕ_{j+1,m_7} and ϕ_{j,k_7} all lie entirely within the support of ϕ_{j,k_7}, we know from Equation (5.3) that each is operator-orthogonal to the plane $t = a_{0,0} + a_{1,0}x + a_{0,1}y$ for any $a_{0,0}$, $a_{1,0}$ and $a_{0,1}$. In particular, this is true for the three planes $t = 1$, $t = x$ and $t = y$, built in Equation (5.5). Consequently:

$$\underset{7\times 7}{\mathbf{M}^T}\ \underset{7\times 3}{\mathbf{B}} = \mathbf{0} \quad \text{with} \quad \mathbf{B} = \begin{bmatrix} b_{\mathbf{k}^*,\mathbf{n}} \end{bmatrix}$$

Because of Equation (5.5), we know that the rank of $\mathbf{B}$ equals three. Therefore, we are guaranteed to have at least three operator-orthogonal wavelets within this support, which we can find by computing the null-space of $\mathbf{M}$. Consequently, we can find three wavelets for each k-node in the interior domain.

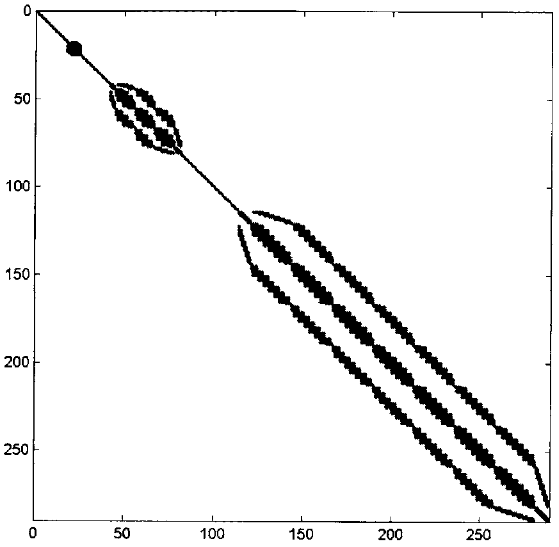


Figure 5-4: Four-level FE matrix of first-order wavelets customized to Poisson's Equation.

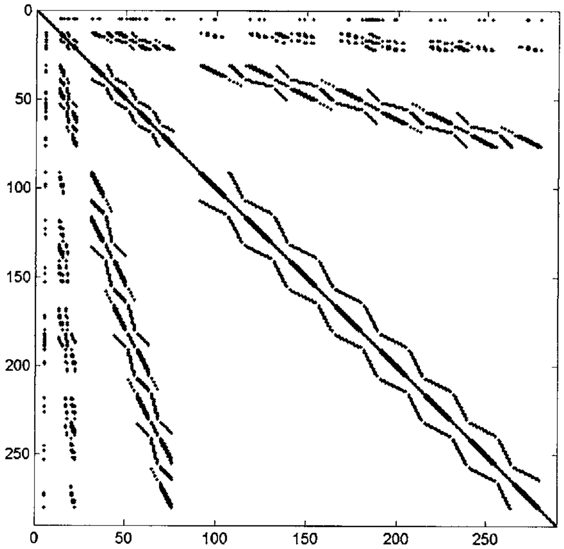


Figure 5-5: Four-level FE matrix of first-order HB for Poisson's Equation.

With the construction described above, we can build a piecewise linear (first-order Lagrangian) wavelet basis that is operator-orthogonal to all scaling functions, with respect to the Laplace operator, on an irregular mesh. Wavelets from such an operator-orthogonal basis are shown in Figure 5-3. We may use these wavelets for the Poisson's Equation on an irregular two-dimensional mesh as shown in Figure 5-6. Instead of the fully coupled stiffness matrix of Figure 5-5, we have now a sparse scale-decoupled stiffness matrix of Figure 5-4. Such decoupling across scales can be very useful in adaptive refinement schemes. Details can be added locally, without having to re-compute a coarser solution.

We leave the handling of boundaries, featuring Dirichlet or Neumann boundary conditions, to a more general discussion in paragraph 5.2.3.

Now that we have been acquainted with this special case for first-order Lagrangian wavelets customized to the Laplace operator, we will present the customization of Lagrangian finite element wavelets of any given order, to more general second-order operators.

5.2 Second-Order Partial Differential Equations

Consider the following two-dimensional second-order elliptic PDE with varying coefficients over the domain Ω, instead of Equation (5.1):

$$-\nabla\cdot\left(\mathbf{P}\nabla u\right)+\mathbf{q}\cdot\nabla u+ru=f \quad \text{with} \quad u\big|_{\Omega}=0 \tag{5.6}$$

We assume for now homogeneous Dirichlet boundary conditions. The mesh may be again unstructured and made up of irregular-spaced triangles, as long as it yields a well-conditioned single-scale FE problem. Just as for the Poisson's Equation, we can use a multi-resolution basis instead of a single-scale basis, for both test and trial functions. In particular, we will use a wavelet basis that spans a Lagrangian finite element space of any given order. The resulting multi-resolution stiffness matrix is in general not decoupled across scales, as discussed in Chapter 2. In order to achieve scale-decoupling, we need to make all wavelets operator-orthogonal to the scaling functions, with respect to the problem's operator. Thus, we wish to satisfy the following condition for *operator orthogonality* with respect to the operator of Equation (5.6) (see Chapter 3):

$$C_{j,k,m}=a\left(\psi_{j,m},\phi_{j,k}\right)=\iint_{\Omega}\mathbf{P}\nabla\psi_{j,m}\cdot\nabla\phi_{j,k}+\mathbf{q}\cdot\nabla\psi_{j,m}\phi_{j,k}+r\,\psi_{j,m}\phi_{j,k}dxdy=0$$
$$\forall m,k \quad \text{and} \quad 0\leq j\leq J-1$$

In addition, we will keep the support of the customized wavelets compact, such that the stiffness matrix remains sparse. We will use the wavelet construction framework of Chapter 3, and the techniques developed for one-dimensional problems in Chapter 4 to construct a basis of wavelets customized to any given elliptic second-order operator.

5.2.1 Non-compact Wavelet Basis

As discussed for one-dimensional operators in paragraph 4.2.2, we could rely on the *decoupling power of lifting* property to build such basis. The corresponding lifted wavelets are operator-orthogonal to all scaling functions, but are – for the two-dimensional problem – unfortunately not compactly supported. An example of such a non-compact customized wavelet is given in Figure 5-6. This first-order Lagrangian

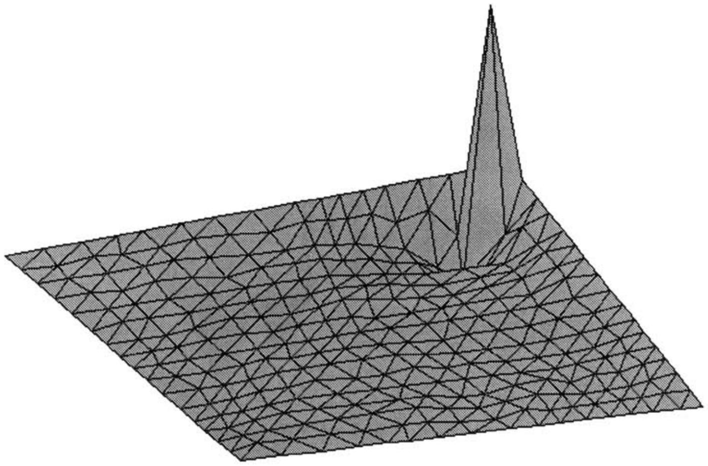


Figure 5-6: Non-compact lifting-only wavelet customized to Laplace.

finite element wavelet is customized to the Laplace operator, on a quadrilateral irregular mesh with Dirichlet boundary conditions. Whereas for the one-dimensional Laplace operator we did manage to build compact customized wavelets without stable completion, for the two-dimensional Laplace operator this is unfortunately not the case, as shown in paragraph 5.1. We want to keep our basis functions compact, to ensure a sparse stiffness matrix and sparse solution transforms, without resorting to basis function truncation or other approximations.

5.2.2 Compact Wavelet Basis

Thus, we will use both lifting and stable completion to construct our wavelets. For both one-dimensional and two-dimensional general elliptic second-order problems, we are provided with an upper bound for the support of customized wavelets, away from the boundary. Indeed, for one-dimensional problems, we have shown in paragraph 4.2.3 how we can always – for a Lagrangian finite element space of any given order – build a compact operator-orthogonal wavelet from a maximum of three neighboring primitive wavelets and interior scaling functions. Away from the boundary, such customized wavelets form a complete basis. Now, we will show how we can construct in a similar manner customized wavelets within compact two-dimensional supports $\Omega_{j,set}$, made up

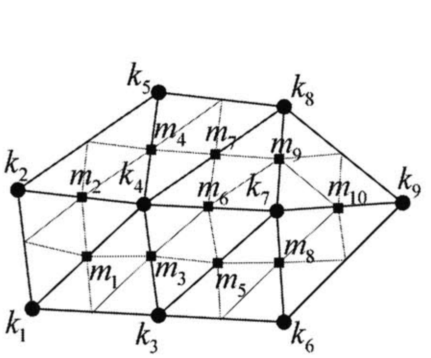


Figure 5-7: Customized wavelet's support of two neighboring scaling functions.

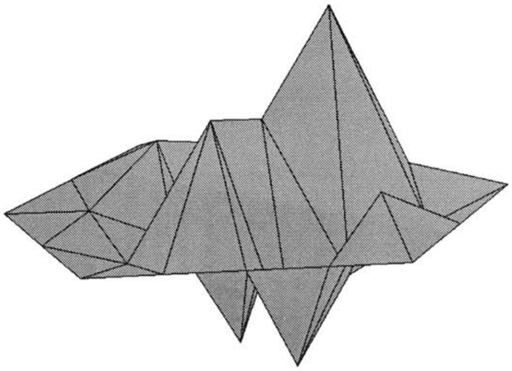


Figure 5-8: One of three wavelets customized to Helmholtz Equation on Figure 5-7 support.

of a set of elements. Consider $\phi_{j+1,\mathbf{m}_{set}}$ containing all primitive wavelets interior to $\Omega_{j,set}$, and $\phi_{j,\mathbf{k}^*_{set}}$ all scaling functions that overlap with $\Omega_{j,set}$. The subset of scaling functions that are interior with respect to $\Omega_{j,set}$ is denoted by $\phi_{j,\mathbf{k}_{set}}$. In Figure 5-7, an example is given for first-order Lagrangian wavelets. Grouping these primitive wavelets with stable completion and using only the interior scaling functions for lifting, we need to satisfy c constraints with dof degrees of freedom. Thus, we need to satisfy:

$$\underset{c\times dof}{\mathbf{M}_{\Omega_{j,set}}} \underset{dof\times sol}{\begin{bmatrix} -s_{j,\mathbf{k}_{set},\mathbf{m}'_{set}} \\ g_{j,\mathbf{m}_{set},\mathbf{m}'_{set}} \end{bmatrix}} = \mathbf{0} \qquad \text{with} \qquad \underset{c\times dof}{\mathbf{M}_{\Omega_{j,set}}} = \underset{c\times dof}{\left[a\left(\phi_{j,\mathbf{k}^*_{set}}, \phi_{j,\mathbf{k}_{set}}\right) \; a\left(\phi_{j,\mathbf{k}^*_{set}}, \phi_{j+1,\mathbf{m}_{set}}\right) \right]}$$

$$\text{and} \quad \left| \begin{array}{l} dof = \text{length}\left(\mathbf{m}_{set}\right) + \text{length}\left(\mathbf{k}_{set}\right) \\ c = \text{length}\left(\mathbf{k}^*_{set}\right) \\ sol = \text{length}\left(\mathbf{m}'_{set}\right) \geq dof - c \end{array} \right. \tag{5.7}$$

The appropriate $g_{j,\mathbf{m}_{set},m'_{set}}$ and $s_{j,\mathbf{k}_{set},m'_{set}}$ per solution m'_{set} can be found by computing the null-space of the interaction matrix $\mathbf{M}_{\Omega_{j,set}}$, which is actually a sub-matrix of $\mathbf{A}_J^{[J-1]}$ formed for primitive wavelets (i.e. traditional Hierarchical Basis). The dimension of the null-space determines the number of linearly independent solutions sol, which is the number of customized wavelets we can find within $\Omega_{j,set}$.

We are guaranteed to have a full-rank (rank sol) stable completion sub-matrix $g_{j,\mathbf{m}_{set},\mathbf{m}'_{set}}$, the part of $\mathbf{G}_j$ corresponding with $\Omega_{j,set}$. Indeed, if $g_{j,\mathbf{m}_{set},\mathbf{m}'_{set}}$ would not be full rank, we would be able to produce a solution to Equation (5.7) for which $g_{j,\mathbf{m}_{set},1} = \mathbf{0}$. Such wavelet, consisting of only scaling functions and no primitive wavelets whatsoever, can never be operator-orthogonal to the other scaling functions with respect to the problem's operator, since that would imply a singular single-scale FEM stiffness matrix.

The smallest support we can think of consists of one element. For a Lagrangian finite element space of any given order n, within a one-element support $\Omega_{j,set}$, we have $dof = 3n(n-1)/2 + (n-1)(n-2)/2 = (n-1)(2n-1)$ degrees of freedom to satisfy $c = n(n+3)/2 + 1$ constraints. Consequently, we are guaranteed to find merely $sol \geq dof - c = 3n(n-3)/2$ customized wavelets entirely within one element. Thus, only for Lagrangian wavelets of very high order, will we have such compact customized wavelets. For a first-order Lagrangian finite element space ($n = 1$), we do not find any solution within the support of one element ($sol \geq -3$).

If we extend this support with one element to two neighboring elements, we instead will have – for a Lagrangian finite element space of any given order n – $dof = n(3n-2) + (n-1)^2$ degrees of freedom and $c = n(n+2) + 1$ constraints. Therefore, we are guaranteed to find $sol \geq dof - c = 3n(n-2)$ customized wavelets entirely within the two-element support $\Omega_{j,set}$. For a first-order Lagrangian finite element space ($n = 1$), we do not find a solution within this support either ($sol \geq -3$).

Therefore, let us look for a solution within the compact support of one scaling function, sitting on a k-node with any given valence v. For this problem, we have $dof = vn(3n-1)/2 + vn(n-1)/2 + 1$ interior primitive wavelets and interior scaling functions, while $c = vn(n+1)/2 + 1$ overlapping scaling functions. Therefore, we are guaranteed to find $sol \geq dof - c = v3n(n-1)/2$ customized wavelets entirely within the compact support of this scaling function. As we have discussed in paragraph 5.1, for a first-order Lagrangian finite element space ($n = 1$), we do not find a wavelet solution within such small support ($sol \geq (v+1) - (v+1) = 0$), unless we are customizing to the Laplace operator, or another special operator. We are guaranteed to find $3v$ second-order Lagrangian wavelets $(n=2)$ within the support of one second-order Lagrangian scaling function.

Finally, since we did not find operator-orthogonal wavelets within the support of one scaling function, we will consider a larger support $\Omega_{j,set}$. We take the support of two neighboring scaling functions, sitting on two neighboring k-nodes with any given valence v_1 and v_2 respectively. This gives us, for any given Lagrangian order n, $dof = (v_1+v_2)n(3n-1)/2 - n(3n-2) + (v_1+v_2)n(n-1)/2 + 2 - (n-1)^2$ degrees of freedom to satisfy $c = (v_1+v_2)n(n+1)/2 + 2 - (n+1)^2$ constraints resulting in at least $sol \geq dof - c = (v_1+v_2)3n(n-1)/2 - 3n(n-2)$ solutions. Therefore, for a first-order Lagrangian finite element space ($n=1$), we can build at least three wavelets customized to any given operator within the support of two scaling functions ($sol \geq (v_1+v_2+1)-(v_1+v_2-2)=3$). Figure 5-8 shows one of three linearly independent wavelets customized to a general operator, on the irregular unstructured mesh section given in Figure 5-7. Note that this result is irrespective of the valences v_1 and v_2 of the two k-nodes. For a second-order Lagrangian finite element space ($n=2$), we can find $3(v_1+v_2)$ second-order Lagrangian wavelets within the support of two neighboring scaling functions. These are exactly the $3v_1$ wavelets contained in the support of the first scaling function alone, and the $3v_2$ wavelets contained in the second scaling function.

Thus, we have specified how we can find compact wavelets customized to a general operator. In paragraph 5.3, we will show that such compact wavelets do form a complete basis, for the first-order Lagrangian case. We are convinced that the same is true for two-dimensional Lagrangian finite element spaces of higher orders.

5.2.3 Special Operators

As in the one-dimensional case – wavelets customized to the Laplace operator are slightly more compact than wavelets customized to more general operators. We have illustrated this with the first-order Lagrangian wavelets in paragraph 5.1. To prove this for a Lagrangian finite element space of any given order, we will rely on the inheritance of vanishing moments property for two-dimensional wavelets, derived in paragraph 5.1.

Scaling functions will in general be piecewise polynomial and not pure polynomial over $\Omega_{j,set}$. However, we are always able to build pure polynomials over the entire wavelet's support, by taking simple linear combinations of the scaling functions. Indeed, away from the boundary, we can build – using all scaling functions associated with $\mathbf{k}^*_{set}$ – any polynomial of order n over the entire set of elements $\Omega_{j,set}$. This means that we can find $pol = (n+1)(n+2)/2$ linearly independent linear combinations $b_{\mathbf{k}^*_{set},n}$ of scaling functions, satisfying:

$$\underset{1\times pol}{\left[\sum_{\mathbf{k}^*_{set}} \phi_{j,k_i} b_{j,k_i,l}\right]_{\Omega_{j,set}}} = \begin{bmatrix} 1 & x & y & xy & x^2 & y^2 & \ldots & x^{n-i}y^i \end{bmatrix}_{\Omega_{j,set}}$$

We will now study the effect of this property on the rank of the interaction matrix $\mathbf{M}_{\Omega_{j,set}}$ associated with the particular Laplace operator (see Equation (5.7)). However, we will first make an adjustment to this matrix. Indeed, we know that in order to benefit from higher order vanishing moments in the wavelet's derivative, we have to enforce vanishing moments in the wavelet itself. Thus, for the Poisson's Equation, we add to $\mathbf{M}_{\Omega_{j,set}}$ exactly van rows that will enforce $van = n(n-1)/2$ vanishing moments on any customized wavelet solution:

$$\underset{(c+van)\times dof}{\mathbf{M}_{\Omega_{j,set}}} \underset{dof\times sol}{\begin{bmatrix} s_{j,\mathbf{k}_{set},\mathbf{m}'_{set}} \\ g_{j,\mathbf{m}_{set},\mathbf{m}'_{set}} \end{bmatrix}} = \mathbf{0} \qquad \text{with} \qquad \underset{(c+van)\times dof}{\mathbf{M}_{\Omega_{j,set}}} = \underset{(c+van)\times dof}{\begin{bmatrix} a\left(\phi_{j,\mathbf{k}^*_{set}}, \phi_{j,\mathbf{k}_{set}}\right) & a\left(\phi_{j,\mathbf{k}^*_{set}}, \phi_{j+1,\mathbf{m}_{set}}\right) \\ \left\langle \mathbf{xy}, \phi_{j,\mathbf{k}_{set}} \right\rangle & \left\langle \mathbf{xy}, \phi_{j+1,\mathbf{m}_{set}} \right\rangle \end{bmatrix}}$$

$$\text{where} \quad \mathbf{xy} = \begin{bmatrix} x^0 y^0 \\ \vdots \\ x^{n-2-i} y^i \end{bmatrix} \quad \text{and} \quad \left| \begin{array}{l} dof = \text{length}(\mathbf{m}_{set}) + \text{length}(\mathbf{k}_{set}) \\ c = \text{length}(\mathbf{k}^*_{set}) \\ van = n(n-1)/2 \\ sol = \text{length}(\mathbf{m}'_{set}) \ge dof - (c + van) \end{array} \right.$$

Because primitive wavelets $\phi_{j+1,\mathbf{m}_{set}}$ and interior scaling functions $\phi_{j,\mathbf{k}_{set}}$ all lie entirely within $\Omega_{j,set}$, we know that each is Laplace-operator-orthogonal to $x^0y^0 = 1$, $x^1y^0 = x$ and $x^0y^1 = y$. In addition, any higher-order inner-product with respect to the Laplace

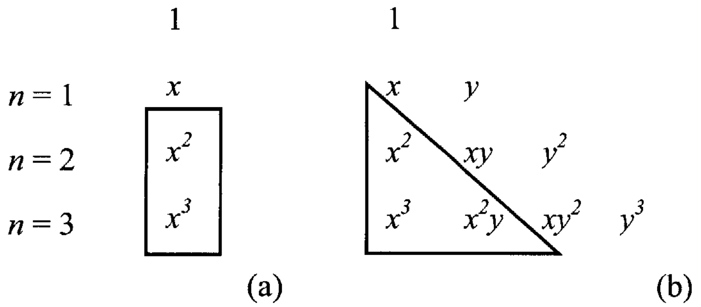


Figure 5-9: Derivative triangle for 1D problem (a), and 2D problem (b).

operator up to $x^{n-i}y^i$ (e.g. $a\left(x^{n-i}y^i,\phi_{j+1,\mathbf{m}_{set}}\right)$) can be eliminated with a moment of this function (e.g. $\phi_{j+1,\mathbf{m}_{set}}$) from one of the additional rows in $\mathbf{M}_{\Omega_{j,set}}$:

$$\iint_{\Omega}\frac{\partial x^{n-i}y^i}{\partial x}\frac{\partial\phi_{j+1,\mathbf{m}_{set}}}{\partial x}dxdy$$

$$=(n-i)\int_{\Omega}x^{n-1-i}y^i\phi_{j+1,\mathbf{m}_{set}}\Big|_{x_1}^{x_2}dy-(n-i)\iint_{\Omega}(n-1-i)\,x^{n-2-i}y^i\phi_{j+1,\mathbf{m}_{set}}dxdy$$

$$\Rightarrow\iint_{\Omega}\frac{\partial x^{n-i}y^i}{\partial x}\frac{\partial\phi_{j+1,\mathbf{m}_{set}}}{\partial x}dxdy+(n-i)\iint_{\Omega}(n-1-i)\,x^{n-2-i}y^i\phi_{j+1,\mathbf{m}_{set}}dxdy=0$$

This is possible because of the inheritance of vanishing moment property: if the wavelet has $van=n(n-1)/2$ vanishing moments, its derivative will have $(n+1)n/2$ vanishing moments, such that the wavelet becomes automatically operator-orthogonal with respect to $pol=(n+1)(n+2)/2$ polynomials. Thus:

$$\underset{dof\times(c+van)}{\mathbf{M}^T_{\Omega_{j,set}}}\;\underset{(c+van)\times pol}{\mathbf{B}}=\mathbf{0}\qquad\text{where}\quad\mathbf{B}=\underset{(c+van)\times pol}{\begin{bmatrix}b_{\mathbf{k}^{*}_{set},\mathbf{n}}\\ \mathbf{L}\end{bmatrix}}\quad\text{with multiplier}\;\underset{van\times pol}{\mathbf{L}}\;\text{well-chosen}$$

In this equation $\mathbf{B}$ has a rank of $pol=(n+1)(n+2)/2$. This allows us to effectively eliminate $(n+1)(n+2)/2$ constraints from our problem defined by $\mathbf{M}_{\Omega_{j,set}}$ for the Laplace operator, but we had to first add $n(n-1)/2$ vanishing moment constraints. Thus, the number of solutions gained over a problem governed by a more general operator is $pol-van=(n+2)(n+1)/2-n(n-1)/2=2n+1$, for a Lagrangian finite element space of order n. Hence, we will be able to build at least $sol=dof-c+(2n+1)$ wavelets customized to the Laplace operator within $\Omega_{j,set}$. Note that this finding is consistent with the discussion of the first-order wavelets ($n=1$) in paragraph 5.1, where we had $pol-van=3$ extra wavelet solutions. For second-order Lagrangians, we will find for the Poisson's Equation $pol-van=5$ extra wavelets for a given compact domain. We refer to the derivative triangle in Figure 5-9, to visualize why for two-dimensional problems the gain for the Laplace operator over general operators is dependent on the order n, whereas this is not the case for one-dimensional problems.

The Poisson case is not the only case in which we find more compact operator-orthogonal wavelets. Indeed, consider the following Diffusion-Convection Equation, basically Equation (5.6) with a non-constant coefficient matrix $\mathbf{P}$, but with constant vector $\mathbf{q}$, and r equal to zero:

$$-\nabla\cdot\left(\mathbf{P}\nabla u\right)+\mathbf{q}\cdot\nabla u=f \quad \text{with} \quad u\big|_{\Omega}=0$$

Because of the partition of unity property, our basis can always form a constant function. By an argument similar to the one-dimensional case in paragraph 4.2.4, this leads to at least $sol\geq dof-c+1$ operator-orthogonal wavelets within a compact set $\Omega_{j,set}$, for this special Diffusion-Convection operator.

Thus, for the first-order Lagrangian finite element space, we are unable to build customized wavelets contained within one element or two neighboring elements. On the increased support of one scaling function, we always find three wavelets customized to the Laplace operator, one wavelet customized to the special Diffusion-Convection operator, but none customized to a more general operator, regardless of the valence v of that scaling fucntion's *k*-node. Finally, on the combined support of two neighboring scaling functions, we can build six (namely three around each *k*-node) wavelets customized to the Laplace operator, at least four wavelets customized to the special Diffusion-Convection operator (of which two are more compactly supported around the *k*-nodes), and three wavelets customized to a more general operator. Away from the boundary, all necessary operator-orthogonal first-order Lagrangian wavelets can be found within such compact supports.

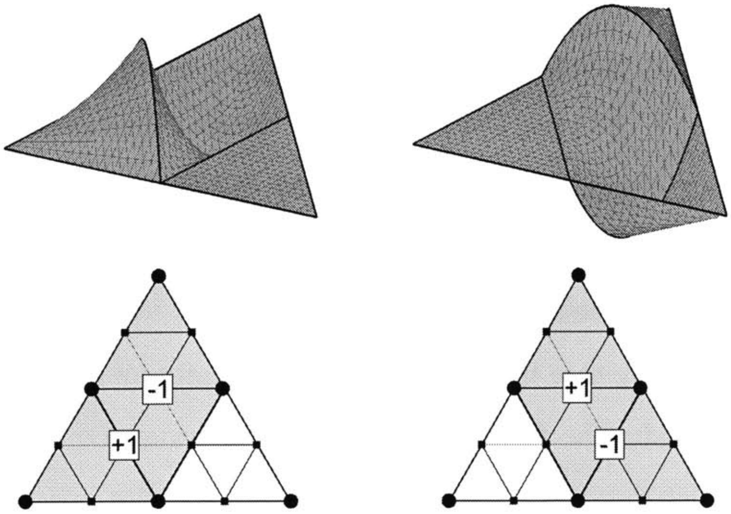


Figure 5-10: Compact quadratic wavelets customized to Laplace operator.

For the second-order Lagrangian finite element space, however, we can find two wavelets customized to the Laplace operator, within the support of one element (see Figure 5-10). These wavelets have one vanishing moment, and consist each of a linear combination of two of the three primitive wavelets interior to the quadratic element. We do not have any wavelets customized to the special Diffusion-Convection operator or a more general operator, within one element's support. If we take two neighboring elements instead, we find a total of five wavelets (four of which are contained within a single element) customized to the Laplace operator, one wavelet customized to the special Diffusion-Convection operator, and still no wavelets customized to a more general operator. Within the support of one scaling function, sitting on a *k*-node with valence v, we have at least $3v+5$ wavelets customized to the Laplace operator (of which $2v+(v-1)=3v-1$ are more compactly supported within one or two elements respectively). We have at least $3v+1$ wavelets ($v-1$ of which are contained within two neighboring elements) customized to the special Diffusion-Convection operator, and at least $3v$ wavelets customized to a more general operator. We leave the analysis of further increased wavelet support to the reader, since we have chosen to focus on the first-order Lagrangian wavelets for the implementation of our method. We are convinced, however, that as in the first-order case, we can build a basis of compact customized wavelets for any given second-order operator. Indeed, we need to find only twelve customized wavelets per scaling function, and with an average valence of six (true for both structured as for unstructured meshes), we have found already per scaling function eighteen wavelets customized to a general operator. Thus a subset of those eighteen is expected to form a full basis, away from the boundary.

It is clear that the special Poisson's and Diffusion-Convection Equations yield operator-orthogonal wavelets that are slightly more compact than in the general case. This quality is caused by the *inheritance of vanishing moments* property and the *partition of unity* property of our Lagrangian basis functions.

5.2.4 Boundary Treatment

Boundaries can be dealt with similarly as in the one-dimensional case. Note that in two-dimensional meshes, also m-nodes can be located on a boundary. The primitive wavelets associated with m-nodes on a Dirichlet boundary will not be used for stable completion with other primitive wavelets. They will each be included as primitive wavelets, not operator-orthogonal to the scaling functions, and then decoupled from the rest of the system, when applying the Dirichlet boundary conditions to the system. Within the support of one scaling function, sitting on a Dirichlet boundary k-node, we find exactly one first-order Lagrangian wavelet customized to the Laplace operator, but no first-order Lagrangian wavelet customized to the Diffusion-Convection operator or a more general operator. An example of a compact wavelet, customized to the Laplace operator, adjacent to a Dirichlet boundary of an irregular-spaced mesh, is given in Figure 5-11. The primitive wavelets associated with m-nodes on a Neumann boundary will be treated the same as the wavelets from the inner domain. Thus, within the support of one scaling function, sitting on a Neumann boundary k-node, we find three first-order Lagrangian wavelets customized to the Laplace operator, but no first-order Lagrangian wavelet customized to the Diffusion-Convection operator or a more general operator.

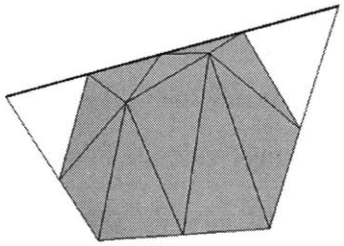


Figure 5-11: First-order Laplace-customized wavelet customized, near Dirichlet boundary.

5.3 Implementation

For simplicity, we will only discuss the implementation for first-order Lagrangian wavelets customized to Laplace the operator and to a general operator, on irregular, unstructured two-dimensional meshes. The most popular first-order Lagrangian finite element space will be also used in our FEM example of Chapter 6 and application of Chapter 7. We expect that a higher-order Lagrangian wavelet method can be implemented with a similar, albeit more advanced procedure.

For the Laplace-case, away from the boundary, we can build three wavelets around each *k*-node. The wavelets have a slightly larger support than the primitive wavelets, but still fall within the support of one scaling function, which is the minimal support possible. Because, away from the boundary, we always have three times the number of *m*-nodes as *k*-nodes, we needed to find on average three *m*-nodes per *k*-node. The three wavelets we find for one *k*-node are linearly dependent from the wavelets we find for all other *k*-nodes. Thus, away from the boundary, all wavelets together will span the full W_j, and yield an invertible stable completion matrix $\mathbf{G}_j$.

For a more general second-order operator, away from the boundary, we can build three wavelets around each pair of *k*-nodes (see Figures 5-7 and 5-8). These operator-orthogonal wavelets are less compact than the primitive wavelets or the Laplace-orthogonal wavelets but are all still compact. Including all these wavelets to construct a basis would result in three wavelets per *m*-node, which would be three times the number we need. Thus, we propose the following scheme to guarantee a full (and not redundant)

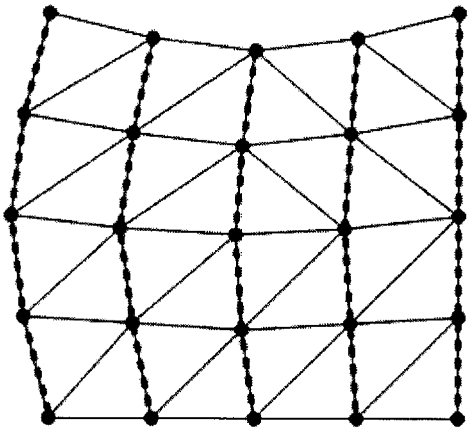


Figure 5-12: Non-crossing chains, connecting all the *k*-nodes.

wavelet basis (see Figure 5-12). We picture continuous chains of *k*-nodes stretched over the entire domain, with one and only one chain going through each k-node (chains never cross). An example could be the typical *minimum bandwidth* numbering scheme for a typical single-scale FEM implementation: all nodes have one and only one number, consecutive numbers correspond with neighboring nodes (or with nodes both on a boundary). Now, we only include customized wavelets associated with *k*-node pairs that are part of a chain. Indeed, if we were to close a figure with the *k*-node pairs, the set of corresponding customized wavelets would not anymore be linearly independent. With this procedure, we find exactly three wavelets per *k*-node – even on unstructured grids. All wavelets together span the full W_j, within the domain.

Thus, – away from the boundary – we could implement a wavelet customization procedure, independently of the nature of the operator (e.g. for mixed operator problems), by first iterating over all *k*-nodes, assigning the wavelets we find, and then iterating over the chains of *k*-nodes, and only using combinations of primitive wavelets that had not been used during the first run (since the supports overlap).

Boundaries slightly complicate this procedure. As discussed in paragraph 5.2.4, we can handle boundaries by excluding from $\mathbf{m}_{set}$, $\mathbf{k}_{set}$ and $\mathbf{k}^*_{set}$ all nodes on a Dirichlet boundary. An example of a Laplace-orthogonal wavelet adjacent to a Dirichlet boundary is given in Figure 5-11. Unlike the one-dimensional case, however, this construction still

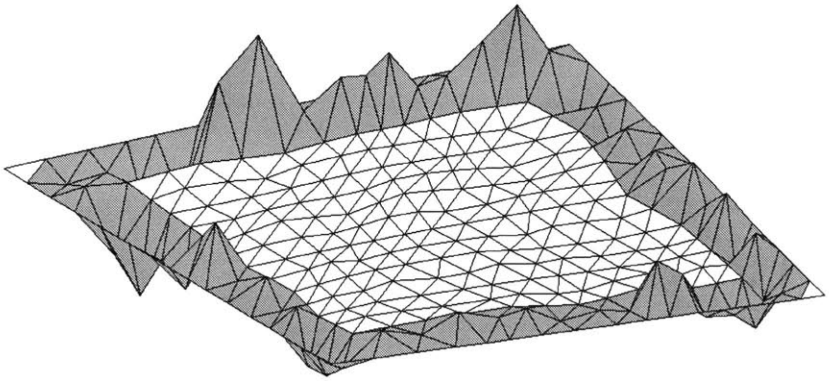


Figure 5.13: First-order wavelet customized to Laplace, along Dirichlet boundary.

can lead to local dependencies of operator-orthogonal wavelets in specific corners of the domain. We can eliminate those by always restricting to three the maximum number of wavelets per *k*-node or *k*-node pair. When we find more than three wavelets in a patch, we re-add (constraints associated with) Dirichlet boundary *k*-nodes to $\mathbf{k}_{set}^{*}$ such that we get only three wavelets. This effectively eliminates this local dependency.

However, for some meshes, we may need to include a single non-compact wavelet customized to the Laplace operator to form a full wavelet basis. This is similar to the single non-compact customized wavelet we needed to add for the one-dimensional Diffusion-Convection operator (see paragraph 4.3). Fortunately, we can construct this extra wavelet such that its support is restricted to the boundary area. An example of such an operator-orthogonal wavelet near the boundary, for an irregular mesh on a quadrilateral domain with Dirichlet boundary conditions, is given in Figure 5-13. This Laplace-orthogonal wavelet consists only of primitive wavelets next to the boundary (no lifting), and has no support away from the boundary. Consequently, this wavelet has only interaction with wavelets near the boundary. Thus, when using an adaptive method for a problem that has no significant solution error adjacent to the boundary, we may exclude this wavelet from our adaptive analysis. Alternatively, we could rely on the *decoupling power of lifting*, discussed in paragraph 5.2.1, to complete our wavelet basis with an operator-customized wavelet consisting of only one primitive wavelet (preferably

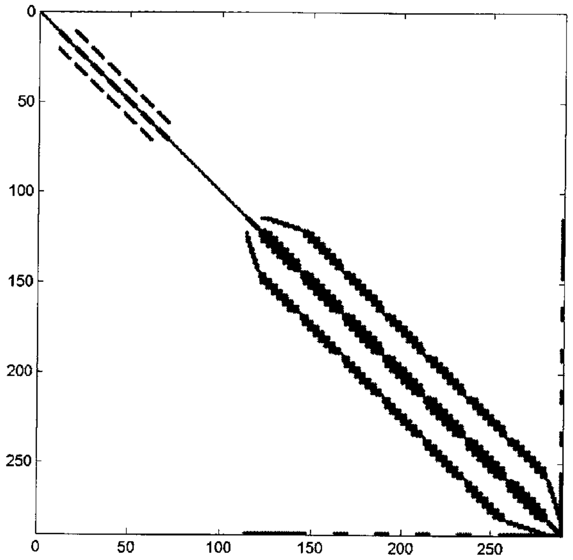


Figure 5-14: Two-level FE matrix of first-order wavelets customized to Poisson's Equation.

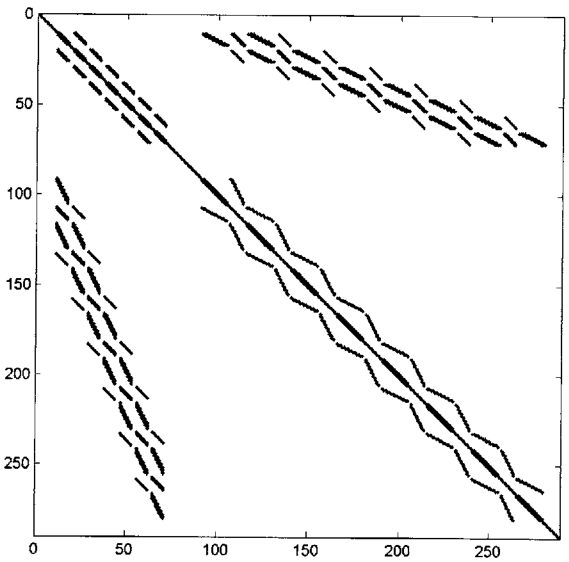


Figure 5-15: Two-level FE matrix of first-order HB for Poisson's Equation.

near the boundary), lifted with all the scaling functions over the domain (see Figure 5-6). This wavelet would have the benefit of fast decay away from the primitive wavelet, and also would be 'compact' in terms of the interaction matrix Δ_{J-1}. Note that for the Poisson's Equation on a triangular domain instead, we would not need such an extra wavelet. For the case of a more general operator, we do not need an extra boundary wavelet either, and all wavelets of the customized wavelet basis are truly compact.

In Figures 5-14 and 5-15, we compare, for the Poisson's Equation on the two-dimensional irregular mesh of Figure 5-13, the scale-decoupled stiffness matrices using an *Operator-Customized Wavelet Basis* (OCWB) or a *Hierarchical Basis* (HB). Note that the highest number Laplace-orthogonal wavelet in Figure 5-14 corresponds to the non-compact boundary wavelet shown in Figure 5-13. In Figure 5-4, we had already shown this example, but with an extra wavelet as shown in Figure 5-6, instead of the wavelet of Figure 5-13. As discussed in paragraph 4.2.3, this non-compact customized wavelet effectively behaves as a compact wavelet with respect to the sub-matrices Δ_j. Note that in Figures 5-14 and 5-15, we apply a full multi-resolution basis instead of the two-level multi-resolution basis of Figures 5-4 and 5-5.

5.4 Conclusion

In this Chapter, we have shown how to construct, for a Lagrangian finite element space of any given order, a basis of compact wavelets customized to any given second-order operator, on an irregular-spaced unstructured two-dimensional mesh. As in the one-dimensional case, the Poisson's Equation yields customized wavelets that have slightly smaller support than the wavelets customized to a more general second-order operator. A good understanding of this difference, caused by an *inheritance of vanishing moments* property, is necessary to ensure the completeness of our customized wavelet basis.

We will now first study the performance of such an *Operator-Customized Wavelet Basis* (OCWB) FEM, and illustrate it with the computation of the Green's function on a two-dimensional bounded domain. In Chapter 7, we will apply our OCWB method to a real-world barrier-option pricing problem, governed by a more general two-dimensional PDE.

6. Complexity Analysis: an Example

6.1 Green's Function Example

We will now demonstrate the application of an operator-orthogonal wavelet basis, with a standard problem: the computation of a Green's function $G(x,y)$ on a bounded domain. The Green's function for the Laplace operator can be computed as the solution to the following equation:

$$\nabla^2 u(x,y) = \delta_{x,y} \tag{6.1}$$

We choose the delta to be located in the center of the domain. We have homogeneous Dirichlet boundary conditions all around the square domain:

$$u(-1,y) = 0 \quad u(1,y) = 0 \quad u(x,-1) = 0 \quad u(x,1) = 0 \tag{6.2}$$

Green's functions (see e.g. Roach, 1982) can have many different applications. They may for example be used as auxiliary function in the solution of a boundary value problem for a series of different forcing functions. Indeed, such problems' solutions can all be found as specific integrals of the initially resolved Green's function. Green's functions are often numerically computed using a Fourier expansion. However, Fourier basis functions are supported all over the domain. In this chapter, we will use a basis of compactly supported wavelets instead, such that details can be added locally to the Green's function, only where needed.

We can use a FEM to numerically solve Equations (6.1) with (6.2). For this problem, we choose first-order Lagrangian elements, which is equivalent to a basis of linear hat functions. They are built on a regular-spaced mesh made up of right triangles. We choose a Ritz-Galerkin method, in which the trial and test basis functions are the same.

As we can see from the solutions for different resolution meshes, shown in Figures 6-2 and 6-4, the delta causes the bulk of the solution error to be centered in the middle of the domain. Thus, we could drastically reduce the problem size, while keeping practically the same accuracy, by refining the mesh only locally around the delta. Such refinement can be easily implemented with a multi-resolution basis, such as a *Hierarchical Basis* (HB). Using this adaptive basis for both test and trial functions results in the following multi-resolution system, coupled across scales:

$$\mathbf{A}_J^{[0]ad}\mathbf{u}_J^{[0]ad} = \mathbf{f}_J^{[0]ad}$$

$$\text{or equiv.} \begin{bmatrix} \mathbf{A}_0 & \mathbf{C}_{0,\mathbf{A}_0}^{ad} & \cdots & \mathbf{C}_{J-1,\mathbf{A}_0}^{ad} \\ \mathbf{C}_{0,\mathbf{A}_0}^{ad\ T} & \mathbf{\Delta}_0^{ad} & & \mathbf{C}_{J-1,\mathbf{\Delta}_0}^{ad} \\ \vdots & & \ddots & \\ \mathbf{C}_{J-1,\mathbf{A}_0}^{ad\ T} & \mathbf{C}_{J-1,\mathbf{\Delta}_0}^{ad\ T} & & \mathbf{\Delta}_{J-1}^{ad} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\gamma}_0^{ad} \\ \vdots \\ \boldsymbol{\gamma}_{J-1}^{ad} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{t}_0^{ad} \\ \vdots \\ \mathbf{t}_{J-1}^{ad} \end{bmatrix}$$

Figure 5-4 shows the sparsity of the non-adaptive HB stiffness matrix $\mathbf{A}_J^{[0]}$. The detail functions are basically primitive wavelets. However, we may also use an adaptive *Operator-Customized Wavelet Basis* (OCWB) to scale-decouple the multi-resolution system. To accomplish scale-decoupling, we construct a basis of wavelets that are operator-orthogonal to the scaling functions with respect to the Laplace operator. The customized wavelets on this square mesh of right triangles are given by Equation (6.3):

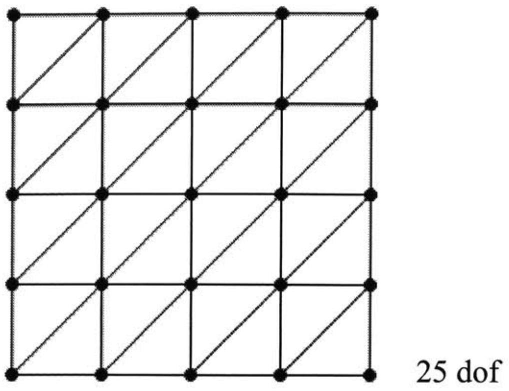


Figure 6-1: Level 0 mesh of *k*-nodes supporting scaling functions.

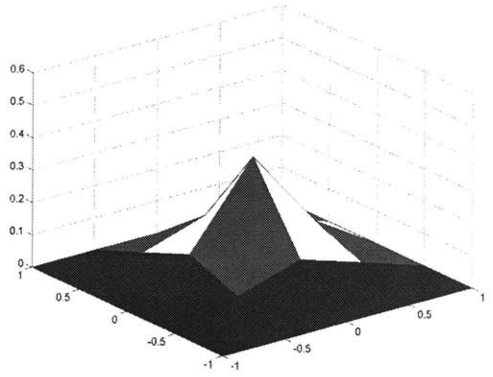


Figure 6-2: Solution on level 0 mesh.

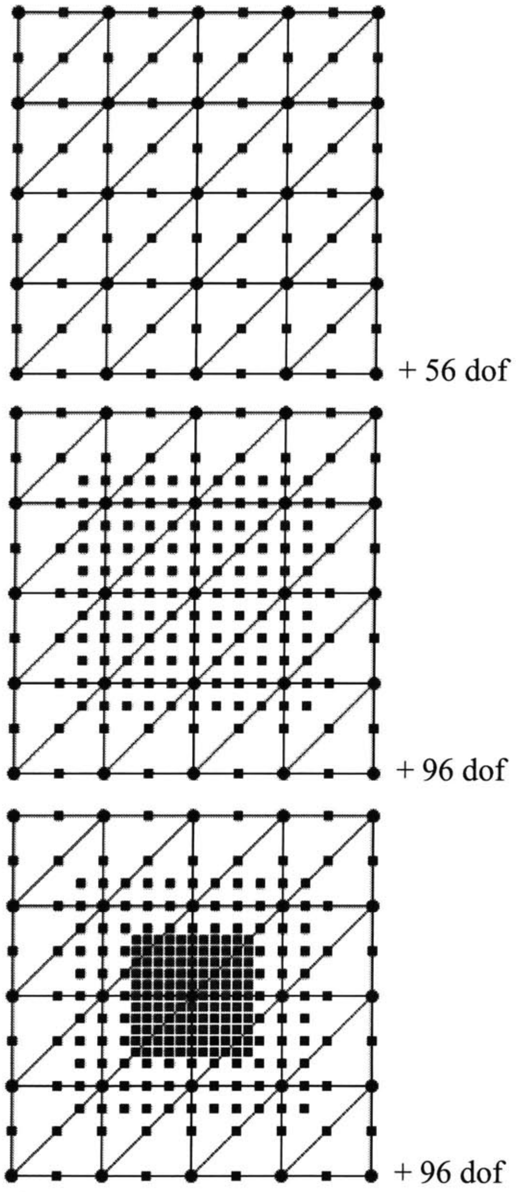


Figure 6-3: Level 0 non-adaptive, and level 1, 2 adaptive HB refinement.

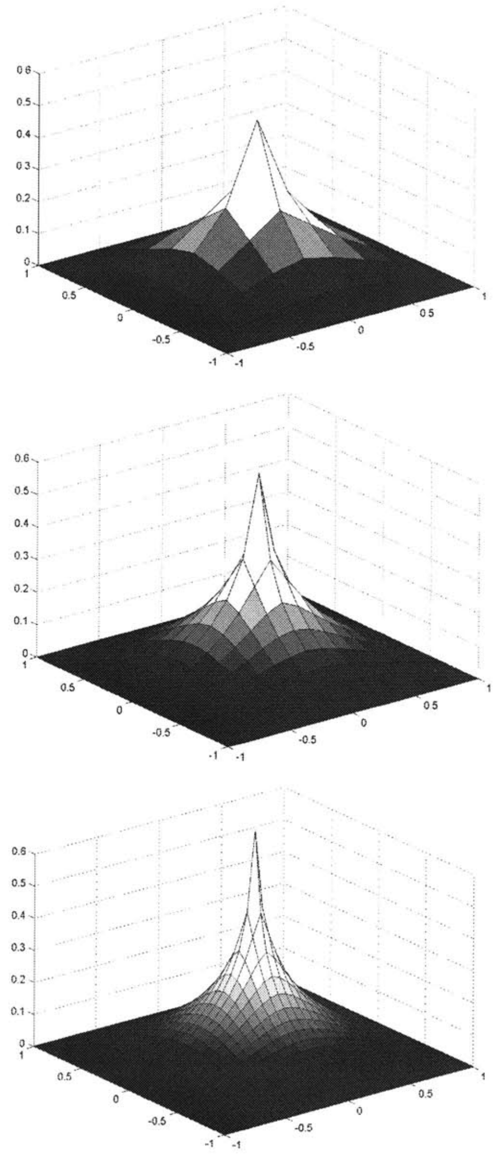


Figure 6-4: (Non)-adaptive solutions for HB refinement.

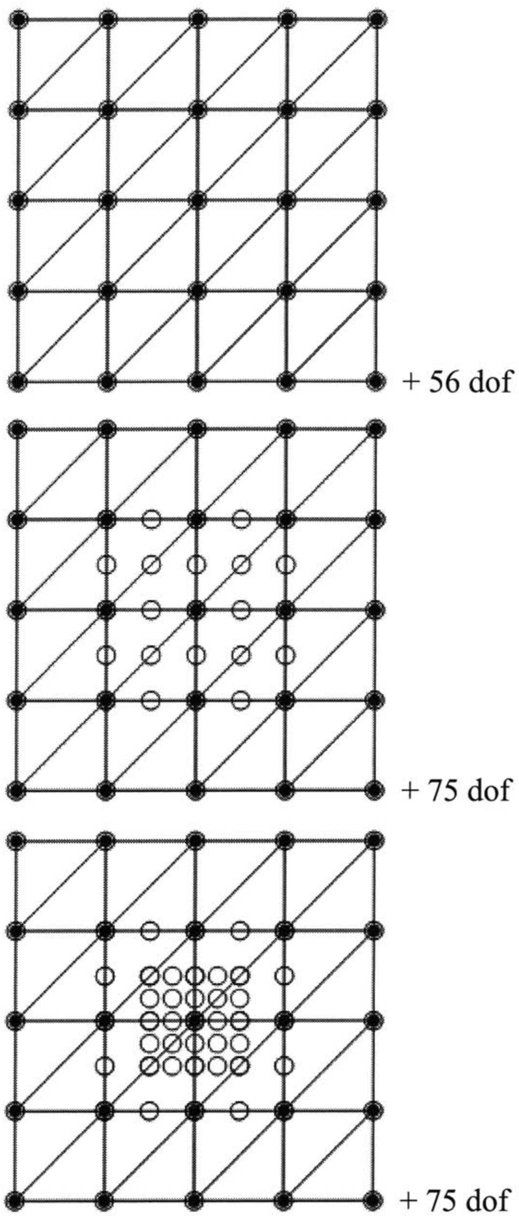


Figure 6-5: Level 0 non-adaptive, and level 1, 2 adaptive OCWB refinement.

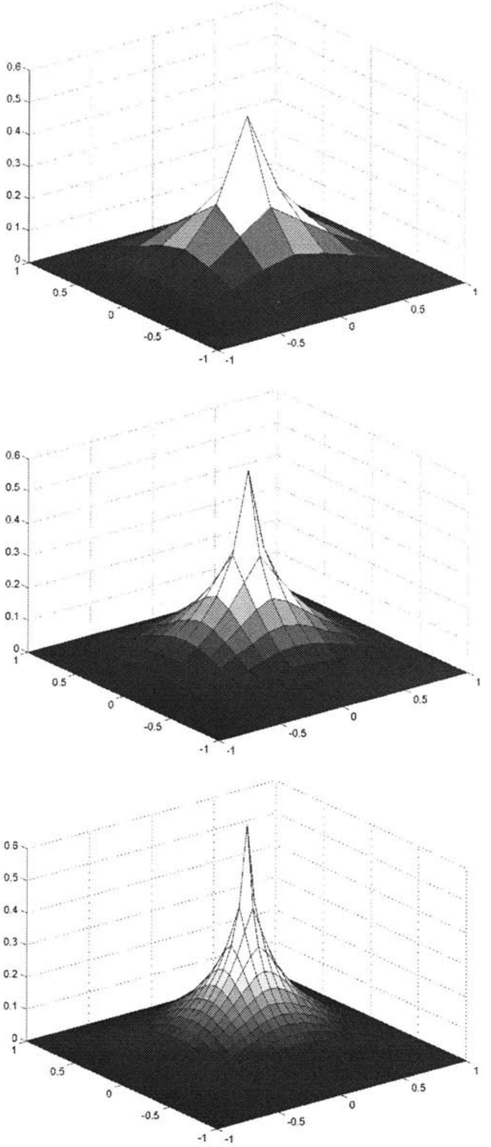


Figure 6-6: (Non)-adaptive solutions for OCWB refinement.

$$\begin{bmatrix} \psi_{j,k(1)} \\ \psi_{j,k(2)} \\ \psi_{j,k(3)} \end{bmatrix} = \begin{bmatrix} 0 & 2 & 1 & 0 & -2 & -1 \\ -2 & 0 & 1 & 2 & 0 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \phi_{j+1,m_1} \\ \phi_{j+1,m_2} \\ \phi_{j+1,m_3} \\ \phi_{j+1,m_4} \\ \phi_{j+1,m_5} \\ \phi_{j+1,m_6} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \phi_{j,k} \tag{6.3}$$

Using these basis functions for both test and trial functions results in the following adaptive multi-resolution system:

$$\mathbf{A}_J^{[0]ad}\mathbf{u}_J^{[0]ad} = \mathbf{f}_J^{[0]ad}$$

$$\text{or equiv.}\quad \begin{bmatrix} \mathbf{A}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Delta}_0^{ad} & & \mathbf{0} \\ \vdots & & \ddots & \\ \mathbf{0} & \mathbf{0} & & \boldsymbol{\Delta}_{J-1}^{ad} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\gamma}_0^{ad} \\ \vdots \\ \boldsymbol{\gamma}_{J-1}^{ad} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{t}_0^{ad} \\ \vdots \\ \mathbf{t}_{J-1}^{ad} \end{bmatrix}$$

Figure 5-4 shows the sparsity of the non-adaptive OCWB stiffness matrix $\mathbf{A}_J^{[0]}$.

Let us now compare an HB refinement with an OCWB refinement. We start with a coarse mesh, as shown in Figure 6-1. The problem on this resolution, defined as level $j = 0$, has a dimension of $N = 25$ degrees of freedom. To gain accuracy, we increase the resolution of the mesh. For the first level of refinement, level $j = 1$, we refine everywhere in the domain. Each primitive wavelet of the HB method is located on an *m*-node between two neighboring *k*-nodes of the coarser mesh (see Figure 6-3). In contrast, the customized wavelets of the OCWB method are grouped in trios around individual *k*-nodes of the coarser mesh (see Figure 6-5). Though, at the boundary these customized wavelets are arranged differently, as explained in Chapter 5. The wavelets on the Dirichlet boundary itself are chosen to be primitive wavelets. In addition, around each *k*-node at the boundary, we have one operator-orthogonal wavelet. The set is completed with one operator-orthogonal wavelet that runs adjacent to the boundary all around the domain. For both the HB method and the OCWB method, the problem's dimension increases with 56 added (wavelet) degrees of freedom to $N = 81$. The FE problem on this resolution yields identical solutions whether under HB or OCWB non-adaptive refinement. For all higher levels of refinement we opt to refine only locally, and to keep constant the number of degrees of freedom added per level. We define *telescopic refinement* as a nested multi-scale refinement in which each refinement level has a dimension that remains of constant order for increasing levels of refinement. As we will show further in this chapter, a scale-decoupled method is most effective for a problem

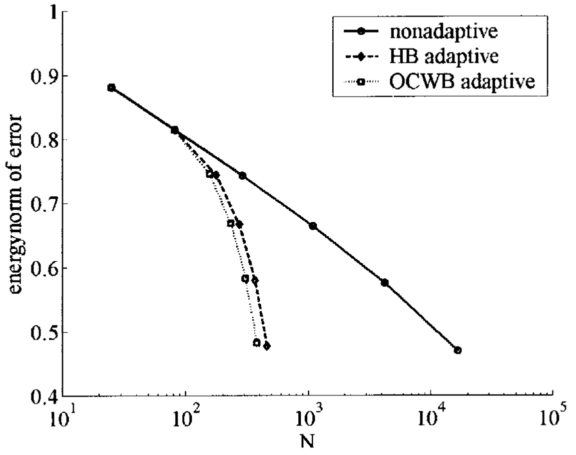


Figure 6-7: Error energy norm of Green's function solutions.

with telescopic refinement. On each level, we will refine only in the area around a patch of $5\times5=25$ k-nodes of that level's mesh, in the middle of the domain. For the HB method, this results in an addition of $(11\times11)-(5\times5)=96$ primitive wavelet degrees of freedom per level, as shown in Figures 6-3. For the OCWB method, we have $25(3)=75$ customized wavelets per level added, as illustrated in Figures 6-5. Note that we do not have to deal with any special boundary wavelets under this adaptive refinement. The plotted solutions per level of adaptive refinement are visually very close to the non-adaptive solution. Figure 6-7 compares the convergence of these adaptive methods with the convergence of the non-adaptive method. The norm of the numerical error $e_{\mathrm{J}}^{\mathrm{ad}}$ is plotted against the total problem size, taking the true solution as the non-adaptive solution $\mathbf{u}_7^{[0]}$ on a much finer mesh:

$$e_{\mathrm{J}}^{\mathrm{ad}}=\sqrt{\left(\begin{bmatrix}\mathbf{u}_7^{[0]ad}\\ \mathbf{0}\end{bmatrix}-\mathbf{u}_7^{[0]}\right)^T\mathbf{A}_7^{[0]}\left(\begin{bmatrix}\mathbf{u}_J^{[0]ad}\\ \mathbf{0}\end{bmatrix}-\mathbf{u}_7^{[0]}\right)}$$

Both the HB and the OCWB telescopic adaptive refinement methods yield comparable accuracy for significantly smaller problem sizes. However, the OCWB method can produce this solution much faster than the HB method. To prove this, we will make a detailed comparison of both method's operation costs of matrix assembly, solving and applicable solution transforms. Note that the storage cost needed for our method will be of the same order as the matrix assembly operation cost.

6.2 Complexity Analysis

6.2.2 Matrix Assembly Cost

The operation cost to assemble a single-scale stiffness matrix is of the order of the number of elements, or equivalently the number of degrees of freedom N. This is irrespective of whether the problem is one-dimensional or higher-dimensional. Indeed, we can build the matrix by iterating over the elements and plugging in the element's stiffness matrix, or by iterating over the degrees of freedom and plugging in the interaction of the corresponding basis function with overlapping basis functions. The construction of an element's stiffness matrix, or of a basis function's interaction vector, has a fixed operation cost, independent of the problem size.

If we were to directly construct a multi-resolution stiffness matrix for a Hierarchical Basis method, we would incur not a cost of $O(N)$, but instead a cost of $O(N \log N)$. Indeed, for each wavelet (degree of freedom) we would need to compute the interaction with the overlapping wavelets of the same level, as well as the interaction with the overlapping wavelets of each other level $j = 0, \ldots, J-1$ and the scaling functions at the coarsest level $j = 0$. Consequently, the operation cost would grow as $O(NJ)$. Now, for a one-dimensional problem, we have:

$$N = 2^J + 1 \quad \Rightarrow O(N) = O(2^J) \quad \Rightarrow O(\log_2 N) = O(J)$$

For n-dimensional problems, this generalizes to:

$$N = (2^J + 1)^n \quad \Rightarrow O(N) = O(2^{nJ}) \quad \Rightarrow O(\log_{2^n} N) = O(J)$$

Thus, the operation cost for a direct construction of the HB stiffness matrix would be $O(N \log N)$. However, we will preferably use an iterative method, such as the *Conjugate Gradient* (CG) method, to solve the HB system. For such iterative method, we do not need to have the multi-resolution matrix in explicit form, if we know a-priori how many levels of refinement we need. We can effectively implement an iterative multi-resolution method with only the fine resolution (level J) single-scale stiffness matrix and dual wavelet transforms. Indeed, we know that:

$$\mathbf{A}_J^{[0]} = \tilde{\mathbf{T}}_0 \ldots \tilde{\mathbf{T}}_{J-1} \mathbf{A}_J \tilde{\mathbf{T}}_{J-1}^T \ldots \tilde{\mathbf{T}}_0^T \tag{6.4}$$

As discussed above, the single-scale matrix can be directly constructed in $O(N)$ operations. For the dual wavelet transforms, we need to construct J square matrices $\tilde{\mathbf{T}}_j$. A transform on level $j-1$ applies only to the scaling function coefficients of the finer level j. Therefore, the size of each transform matrix grows with the transform's level $j-1$, as $\left(2^j+1\right)^n$. We have pointed out before (in Chapter 3) that the dual transform matrices $\tilde{\mathbf{T}}_j$ are sparse for any choice of compact wavelet constructed by equations (3.3), whereas the primary transform matrices $\mathbf{T}_j$ are only guaranteed to be sparse for compact wavelets that are constructed without stable completion. The cost of building a matrix row that corresponds with a wavelet is very small and independent of the problem size. It is merely placing an off-diagonal entry of one. The cost of building a matrix row that corresponds with a scaling function is also small and independent of the problem size. It consists of filling-in the coefficients of the very compact filter $h_{j,k,m}^0$. Consequently, the cost of constructing each transform matrix relates to the level as:

$$O\left(\left(2^j+1\right)^n\right) = O\left(\frac{N}{2^{n(J-j)}}\right) \quad \forall j = 1,\ldots,J$$

Thus, the total cost of constructing all J dual wavelet transforms will be $O(N)$:

$$O\left(N + \frac{N}{2^n} + \ldots + \frac{N}{2^{n(J-1)}}\right) = O(N) \tag{6.5}$$

We will also use the dual wavelet transforms to transform the multi-resolution solution back to single-scale format.

These costs are for a non-adaptive HB method. An adaptive method results in a decrease in problem size N_{ad}, for a given number of levels of refinement $j = J$. If we have telescopic refinement – as in our example of the Green's function – we have the following relation:

$$N_{ad} = C_1 + JC_2$$

And consequently:

$$O(N_{ad}) = O(J)$$

If we were to build directly the HB multi-resolution matrix for a problem with telescopic refinement, the operation cost would be $O(J^2)$. Instead, we again can use Equation (6.4), under the assumption we know a-priori exactly where and how far we will need refinement. However, it would be too expensive to use the non-adaptive stiffness matrix $\mathbf{A}_J$, since the assembly cost for this matrix is $O(N)$□ $O(N_{ad})$. Therefore, we propose to use a modified version of the non-adaptive $\mathbf{A}_J$. This modified stiffness matrix $\mathbf{A}_J^*$ has the format of a single-scale stiffness matrix, but the mesh has a varying resolution over the domain, corresponding to the finest level of refinement that is locally required for the adaptive method. To handle the jumps in resolution, we hem-in each patch of finer *k*-nodes with added 'hanging' k-nodes, and simply use green refinement to connect these redundant *k*-nodes to the surrounding *k*-nodes of the coarser mesh. Such green refinement is illustrated by Figure 6-8. In addition – to avoid redundant renumbering in our wavelet transforms – we do not number the modified matrix' degrees of freedoms according to the customary minimum-bandwidth rules. Instead, we first number the *k*-nodes of the coarsest level ($j = 0$) mesh, then the not-yet-numbered *k*-nodes of one level finer, etc... ending with the finest level $j = J$. It is important to note that we do not solve the problem on this 'refined' mesh, so we do not have to worry about the shape of the transition elements, nor the matrix' bandwidth. The modified matrix $\mathbf{A}_J^*$ is merely an implementation aid. Since this matrix is constructed as a regular stiffness matrix, with $N_{ad} + JC_{hanging}$ degrees of freedom, its assembly cost remains $O(J)$.

We now can build modified dual wavelet transform matrices associated with this matrix. Their construction is as described above for the non-adaptive case, except that a transform matrix on a particular level is only constructed for wavelets and scaling functions that overlap with the refinement zone on that level. Wavelets associated with the 'hanging' finer level *k*-nodes are not added in, since those degrees of freedom were

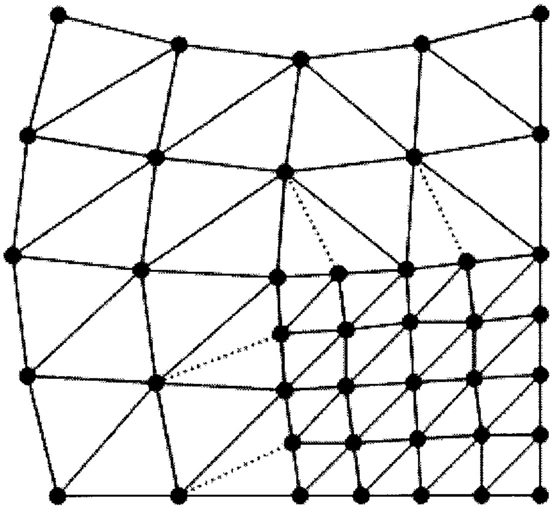


Figure 6-8: Green refinement around refinement zones.

added outside of the refinement zone. Wavelets outside of the refinement zone obviously are not included either. They could not even be built from the modified matrix $\mathbf{A}_J^*$. Scaling functions that overlap with the resolution jump can be easily constructed with Equation (3.1) from the finer scaling functions in the refinement zone and the finer scaling functions associated with the 'hanging' k-nodes. Scaling functions outside of the refinement zone effectively default to the scaling functions of the 'finer' level. Because of our convenient choice of numbering in the modified stiffness matrix $\mathbf{A}_J^*$, we do not need to renumber these degrees of freedom. Hence, they do not need to be included in the wavelet transform. Note that, although our adaptive method effectively has N_{ad} degrees of freedom, the modified system matrix has a slightly bigger size because of the added 'hanging' nodes. Consequently, the adaptive modified transform matrices $\tilde{\mathbf{T}}_j^{ad*}$ are not exactly square. We have:

$$\mathbf{A}_J^{[0]} = \tilde{\mathbf{T}}_0^{ad*} \ldots \tilde{\mathbf{T}}_{J-1}^{ad*} \mathbf{A}_0^* \tilde{\mathbf{T}}_{J-1}^{ad*T} \ldots \tilde{\mathbf{T}}_0^{ad*T}$$

In the case of telescopic refinement, each of the J dual wavelet transform matrices $\tilde{\mathbf{T}}_j^{ad*}$ has a fixed size, the number of wavelets and scaling functions overlapping with the refinement patch. Thus, the total cost of assembly of all wavelet transforms, and the modified stiffness matrix, is $O(J)$ for an adaptive HB method.

If instead we are using an OCWB multi-resolution method, we can directly build the multi-resolution matrix, as well as all wavelet transforms, with only $O(N)$ cost. We first need to determine for each customized wavelet the appropriate set of stable completion coefficients and lifting coefficients. Since practically all customized wavelets are constructed within compact support, this will be an $O(N)$ operation. The details of this customization are discussed in Chapter 5. For a wavelet on level j, we need to compute the interactions – with respect to the weak form of the operator – of functions related to a pre-selected support. The interactions between at one hand the level j scaling functions overlapping with this support, and at the other hand the internal level $j-1$ scaling functions (level j primitive wavelets) as well as the internal level j scaling functions are placed in an interaction matrix $\mathbf{M}_{\Omega_{j,set}}$ (see Equation (5.7)):

$$\underset{c\times dof}{\mathbf{M}_{\Omega_{j,set}}}=\underset{c\times dof}{\left[a\left(\phi_{j,\mathbf{k}_{set}^{*}},\phi_{j,\mathbf{k}_{set}}\right)\quad a\left(\phi_{j,\mathbf{k}_{set}^{*}},\phi_{j+1,\mathbf{m}_{set}}\right)\right]}$$

The null-space of this small matrix – of N -invariant size – yields the desired coefficients for the customization of the wavelet. Of course, knowing the dimension of the null-space of $\mathbf{M}_{\Omega_{j,set}}$ enables us to fix some of the coefficients, such that we can solve for the remaining ones in a fully determined system. For two-dimensional Poisson's Equation, we have one non-local wavelet, which runs along the boundary, as discussed in Chapter 5. This boundary wavelet has a support of $O\left(\sqrt{N}\right)$ and will clearly not affect the order of the entire wavelet customization operation:

$$O\left(C_1(N-1)+C_2\left(\sqrt{N}\right)\right)=O(N)$$

In parallel with the customization, we can build the dual wavelet transform matrices $\tilde{\mathbf{T}}_j$. We will need these transforms to transform the solution from multi-resolution to single-scale format. As in the HB case, all dual wavelet transforms can be built in $O(N)$ operation cost. The only difference is that, for the OCWB method, the cost of building a wavelet row is a slightly higher constant. Instead of placing an off-diagonal matrix entry of one – corresponding with a primitive wavelet –, we insert the stable completion and filter coefficients. After we have determined the stable completion and lifting

	single-scale	HB	HB	OCWB	OCWB
	direct solver	direct solver	iterative solver	direct solver	iterative solver
	N	$N\log N$	N	N	N
adaptive		J^2	J	J	J

Table 6-1: Assembly costs [order of], non-adaptive and with telescopic adaptivity.

coefficients, we can directly build the multi-resolution matrix $\mathbf{A}_J^{[0]}$. Unlike in the HB case, we only need to compute the local interaction between wavelets of the same level. Indeed, by forcing decoupling between wavelets and scaling function on one level, we have effectively eliminated also the coupling between those wavelets and all coarser level wavelets and scaling functions, as discussed in Chapter 3. This enables us to directly construct the multi-resolution matrix $\mathbf{A}_J^{[0]}$ with only $O(N)$ operation cost, whereas a direct construction of the stiffness matrix would cost $O(N\log N)$ for the Hierarchical Basis method.

The assembly costs for an adaptive OCWB method, with telescopic refinement are $O(J)$ – as in the adaptive HB case. Although we now do not need to explicitly construct a modified single-scale stiffness matrix $\mathbf{A}_J^*$, we can build all dual wavelet transforms $\tilde{\mathbf{T}}_j^{ad*}$ based on such matrix in $O(J)$ operations. However, if for the two-dimensional Poisson's Equation our refinement patch were to include the non-local boundary wavelet, the operation cost could be much higher than $O(J)$. This is not the case for problems such as our Green's function example, where the solution error is sufficiently small right on the boundary. In cases where telescopic refinement is appropriate and we do want to include this non-local boundary wavelet, we could limit the customization of the wavelet to where its support overlaps with the refinement patch. Outside this area, we assume the wavelet is operator-orthogonal to the scaling functions, without explicitly determining the stable completion and lifting coefficients. This does not affect the stiffness matrix, since the customized wavelet has no interaction with the scaling functions and it has no neighboring wavelets outside of the refinement patch. It does exclude part of the wavelet's participation from our solution, but that would be tolerable under our assumption of telescopic refinement. This modification would bring the assembly cost again down to $O(J)$. After determining the customization, we can directly construct the multi-resolution matrix in $O(J)$ operations as well. An overview of assembly costs is given in Table 6-1.

6.2.2 Solution Cost

The critical operation cost of the solving phase will distinguish the OCWB method from the HB method, and the adaptive methods from the non-adaptive. Although the methods we compare have substantial differences, they will rely on either of two well-known basic techniques for solving a system $\mathbf{Au} = \mathbf{f}$: Gaussian Elimination, or the Conjugate Gradient method.

Gaussian Elimination is a direct solving method (see Strang, 1993) in which the system is solved by transforming the augmented system matrix $[\mathbf{A} \mid \mathbf{f}]$ with elementary row operations into an upper triangular matrix, and then solving for $\mathbf{u}$ with back-substitution. The complexity of Gaussian Elimination depends directly on the sparsity of the matrix $\mathbf{A}$, and is of the order $O(Nb^2)$, in which N is the dimension and b is the bandwidth of the matrix $\mathbf{A}$. The *Conjugate Gradient* (CG) method is an iterative solving method (see Golub, 1996) for symmetric positive definite systems. Per iteration, a solution approximation, the corresponding residual and search direction are generated. The complexity of the CG method depends primarily on the matrix's condition number, a measure of how close the matrix is to being singular. The condition number κ is the square root of the ratio between the highest and smallest eigenvalue of the matrix. The number of iterations needed to obtain a sufficiently accurate solution is of the order $O(\sqrt{\kappa})$. The cost of one iteration is $O(N)$ for compactly supported basis functions, in which N is the dimension of $\mathbf{u}$. In fact – apart from a few additions and scalar multiplications –, this iteration cost is incurred by applying the sparse matrix $\mathbf{A}$ to the search direction vector (with dimension N), and by computing the inner product of this search direction vector and of the residual (with dimension N). Therefore, the CG solving cost is $O\left(N\sqrt{\kappa}\right)$.

A single-scale FE system can by default be solved with Gaussian Elimination, with a cost depending on the problem's spatial characteristics. The stiffness matrix' bandwidth is constant for a one-dimensional problem, $O\left(N^{1/2}\right)$ for a two-dimensional problem, and $O\left(N^{2/3}\right)$ for a three-dimensional problem. Thus, the solving costs would be $O(N)$, $O\left(N^2\right)$ and $O\left(N^{7/3}\right)$ respectively.

We now consider instead a non-adaptive Hierarchical Basis FE method. The Gaussian Elimination cost for this multi-resolution system is of the same order as for the single-scale system, for higher dimensional problems. For one-dimensional problems, the stiffness matrix' bandwidth has order $O\left(NJ^2\right)$. Indeed, each row – associated with a level j wavelet – has non-zero entries corresponding to the interaction of this wavelet with a compact set of scaling functions on level 0, compact set of wavelets on each of the $j-1$ coarser levels, compact set of wavelets on level j, and compact set of wavelets on each of the $J-1-j$ finer levels. Thus, the bandwidth grows by:

$$O\left(C_{J-1}+\ldots+C_j+\ldots+C_0\right)=O(J)$$

Note that in the case of linear hats and a Poisson's Equation, the cost would be only $O(N)$, since such HB method actually corresponds to an OCWB method for which $C_i=C_j\delta_{i-j}$, with δ_{i-j} the Kronecker delta. For two-dimensional problems ($n=2$), or three-dimensional problems ($n=3$), the bandwidth grows by:

$$O\left(N^{(n-1)/n}+\left(\frac{N}{2^n}\right)^{(n-1)/n}+\ldots+\left(\frac{N}{2^{n(J-1)}}\right)^{(n-1)/n}\right)=O\left(N^{(n-1)/n}\right)$$

However, the HB method is well-recognized for its performance as an effective stiffness matrix pre-conditioner. Because of this, the CG iterative method outperforms direct Gaussian elimination, for multi-dimensional problems. Note that per iteration, instead of explicitly constructing and applying the HB stiffness matrix $\mathbf{A}_J^{[0]}$ to the search direction vector p_i, we apply a series of dual wavelet transforms and the single-scale matrix to the search direction vector:

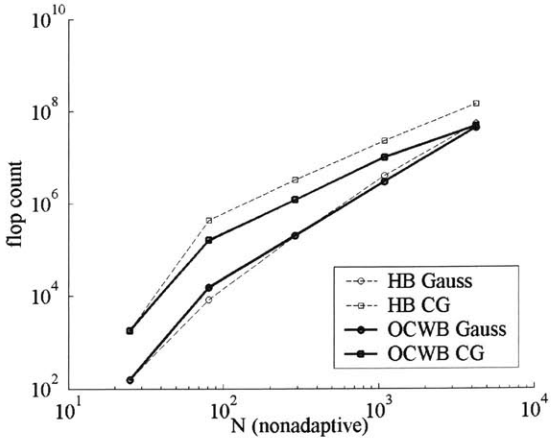


Figure 6-9: Solving cost vs. dimension of non-adaptive solution of Green's function.

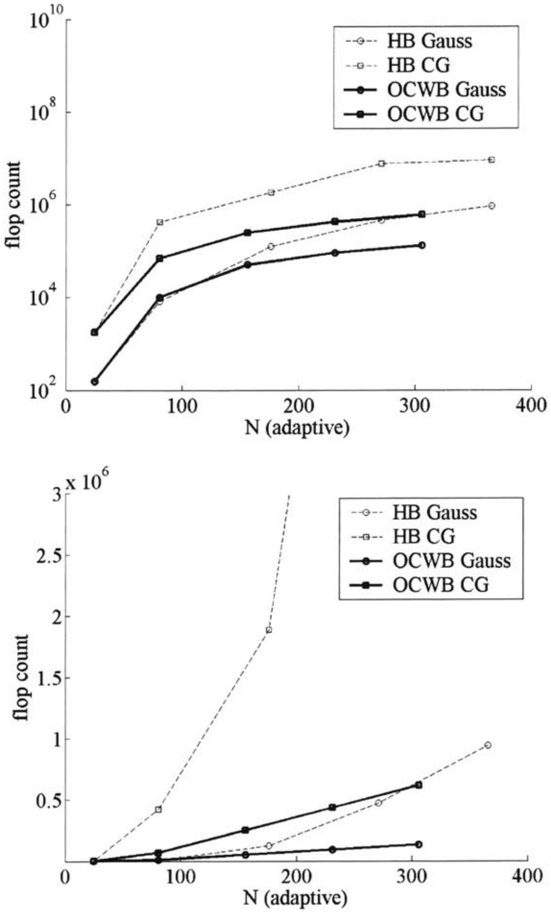


Figure 6-10: Solving cost vs. dimension of adaptive solution of Green's function.

$$\mathbf{A}_J^{[0]} p_i = \tilde{\mathbf{T}}_0 \dots \tilde{\mathbf{T}}_{J-1} \mathbf{A}_J \tilde{\mathbf{T}}_{J-1}^T \dots \tilde{\mathbf{T}}_0^T p_i$$

Thus, we can avoid a costly explicit assembly of the HB stiffness matrix. Because of the sparsity of the dual wavelet transforms and single-scale matrix $\mathbf{A}_J$, and in view of Equation (6.5), this is effectively an $O(N)$ operation. For one-dimensional problems, the stiffness matrix has a condition number of constant order, resulting in an $O(N)$ cost for the CG method. For two-dimensional problems, the condition number is known to be bounded by $C(\log N)^2$, whereas for three-dimensional problems by CN (Yserentant, 1992). Therefore, the CG method has a cost of order $O(N \log N)$ for two-dimensional problems, and $O(N^{3/2})$ for three-dimensional respectively.

Figure 6-9 illustrates the difference in computational performance between a Gaussian elimination method and a CG method for a non-adaptive HB solution of the two-dimensional Green function example discussed in this chapter. The floating operation count (MATLAB 5.3.1 *flops*) for both procedures is plotted against the problem's dimension. The CG method clearly outperforms Gaussian elimination asymptotically.

A major reason for the use of a multi-resolution method is its effectiveness in implementing an adaptive method. We have shown in Figure 6-7 how the exploitation of strong (telescopic) adaptivity does not hurt convergence, for our example problem. Comparing Figure 6-9 with Figure 6-10 shows how much we gain in solving speed by reducing the full dimension N to an adaptive dimension N_{ad}. Note that the x-axis of Figure 6-10 is now linear instead of logarithmic.

The cost of a direct solver for the adaptive HB method is $O\left(N_{ad}J^2\right)=O\left(J^3\right)$, for one-dimensional as well as multi-dimensional problems. Indeed, the bandwidth of the multi-resolution stiffness matrix will grow linearly with J only, since the number of wavelets on each level remains constant.

For the iterative CG method, we rely again on a modified single-scale stiffness matrix $\mathbf{A}_J^*$ and adaptive modified transforms $\tilde{\mathbf{T}}_j^{ad*}$. The iterative CG solving cost for a one-dimensional HB problem with telescopic refinement is $O\left(N_{ad}\right)=O\left(J\right)$. We do not have a close theoretical bound for the iterative solving cost for higher-dimensional problems, as it depends on the condition number of the adaptive HB stiffness matrix. We could assume the cost to be well above $O\left(N_{ad}\right)=O\left(J\right)$, since it is highly unlikely for the condition number to remain constant. We also consider $O\left(N_{ad}\log N\right)=O\left(J^2\right)$ an upper bound for two-dimensional problems, and $O\left(N_{ad}N^{1/2}\right)=O\left(J8^{J/2}\right)$ an upper bound for three-dimensional problems. As shown in Figure 6-10, our results for the two-dimensional Green function example – with telescopic refinement – support a value closer to $O\left(J^2\right)$ for two-dimensional problems.

To achieve an optimal solving speed of $O\left(N_{ad}\right)=O\left(J\right)$ for multi-dimensional problems with telescopic adaptivity, we can customize our detail basis functions to scale-decouple the multi-resolution stiffness matrix. By using an OCWB method, the stiffness

	single-scale direct solver	HB direct solver	HB iterative solver	OCWB direct solver	OCWB iterative solver
1D	N	$N\log^2 N$	N	N	N
adaptive		J^3	J	J	J
2D	NN	NN	$N\log N$	NN	$N\log N$ (*)
adaptive		J^3	J^2 (*)	J	J
3D	$N^{4/3}N$	$N^{4/3}N$	$NN^{1/2}$	$N^{4/3}N$	
adaptive		J^3		J	J

Table 6-2: Solution costs [order of], non-adaptive and with telescopic adaptivity.

matrix breaks down to a block-diagonal matrix, with each block containing the (limited bandwidth) wavelet interaction of one level. For a problem with telescopic refinement, the dimensions (as well as bandwidth) of those blocks are J-invariant. Thus, they can each be solved directly with a solving cost of constant order:

$$\begin{bmatrix} \mathbf{A}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Delta}_0^{ad} & & \mathbf{0} \\ \vdots & & \ddots & \\ \mathbf{0} & \mathbf{0} & & \boldsymbol{\Delta}_{J-1}^{ad} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\gamma}_0^{ad} \\ \vdots \\ \boldsymbol{\gamma}_{J-1}^{ad} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{t}_0^{ad} \\ \vdots \\ \mathbf{t}_{J-1}^{ad} \end{bmatrix} \rightarrow \left| \begin{array}{l} \mathbf{A}_0\boldsymbol{\lambda}_0 = \mathbf{f}_0 \Rightarrow \boldsymbol{\lambda}_0 \\ \boldsymbol{\Delta}_0^{ad}\boldsymbol{\gamma}_0^{ad} = \mathbf{t}_0^{ad} \Rightarrow \boldsymbol{\gamma}_0^{ad} \\ \vdots \\ \boldsymbol{\Delta}_{J-1}^{ad}\boldsymbol{\gamma}_{J-1}^{ad} = \mathbf{t}_{J-1}^{ad} \Rightarrow \boldsymbol{\gamma}_{J-1}^{ad} \end{array} \right.$$

Therefore, the total solving cost remains $O(J)$. In Figure 6-10, the solving cost of an adaptive OCWB method is compared with the cost of an adaptive HB method. In order to appreciate the significance of this difference, we have re-plotted the graph of Figure 6-10 with a linear instead of logarithmic scale for the flop count axis. Indeed, with an $O(J)$ solving cost, the OCWB truly outperforms the HB method for problems with telescopic refinement.

The non-adaptive OCWB method would have a direct solving cost of the same order as a single-scale direct method. Indeed, the cost will be dominated by the cost of solving the finest level detail block, which will have a dimension of $(3/4)N$ and a

(*) Based on numerical experiments.

	single-scale	HB	HB	OCWB	OCWB
	direct solver	direct solver	iterative solver	direct solver	iterative solver
	0	N	N	N	N
adaptive	0	J	J	J	J

Table 6-3: Transformation costs [order of], non-adaptive and with telescopic adaptivity.

bandwidth of order $O\left(N^{(n-1)/n}\right)$. Thus, the cost would be of order $O\left(N^{(3n-2)/n}\right)$, which is also the direct solving cost of a non-adaptive HB method. It is more difficult to gauge the cost of an iterative solver for the non-adaptive OCWB method. Indeed, this largely depends on the condition number of the finest level detail block, which is affected by the condition number of the stable completion matrix $\mathbf{G}_j$, of Equation (3.3). If this matrix is well-conditioned, we will have a slow-growing condition number in the detail blocks. Note that for a HB method, the stable completion matrix effectively is the unity matrix, with condition number one. For the HB method, the condition number of each detail block is constant. Notwithstanding this, it is possible for an OCWB iterative method to slightly outperform the HB iterative solver, because of the scale-decoupling effect. This is the case for the non-adaptive two-dimensional Green function example. Figure 6-9 compares the non-adaptive OCWB method's performance to the non-adaptive HB method. Thus, even if we have absolutely no adaptivity, we may still choose the OCWB over the HB method. An overview of all solving costs is given in Table 6-2.

6.2.3 Solution Transformation Cost

When using a multi-resolution method, we often need to transform the multi-resolution solution – generated in the solving stage – back to a single-scale format. Indeed, though for some problems only a local solution or solution properties are needed, many problems will call for an explicit sample of the entire solution. This is best achieved by applying J dual wavelet transforms to the multi-resolution solution:

$$\mathbf{u}_J = \tilde{\mathbf{T}}_{J-1}^T \ldots \tilde{\mathbf{T}}_0^T \mathbf{u}_J^{[0]}$$

For the adaptive methods, we may use modified dual wavelet transforms $\tilde{\mathbf{T}}_j^{ad*}$ instead. The total cost of this operation has the same order as the total cost of assembly of these transforms. This cost has been discussed in detail above. Thus, for both the HB as the OCWB method, the cost of transforming the solution will be $O(N)$ for non-adaptive problems, and $O(J)$ for problems with telescopic adaptive refinement (see Table 6-3).

6.2.4 Complexity Comparison

The total costs of the different methods are all critically determined by the cost of the solving phase, and are thus given in Table 6-2. Clearly an adaptive OCWB method is optimal for problems with highly local features in the solution, e.g. the two-dimensional Green function example, or the applications we will discuss in Chapter 7. For problems that have less-local details, or even no adaptivity whatsoever, the OCWB method may be outperformed by the HB iterative method, if the condition numbers of the stiffness matrix' detail parts grow too fast. Those depend on the condition number of the stable completion matrices $\mathbf{G}_j$. For a two-dimensional Poisson's Equation, we found this to be very well-behaved, and the non-adaptive OCWB slightly outperformed the non-adaptive HB iterative method. For the applications of Chapter 7, with a more general operator, we found the condition number to grow faster.

6.3 Refinement Strategy

When we anticipate local details and use an adaptive method such as the OCWB method, we still have to determine where and how deep to refine. For our Green function example, we had pre-specified the refinement zone, but for other applications, we might need to determine the refinement zones and depth a-posteriori. In light of this issue, it is important to stress that the OCWB method is truly a level-per-level method. We can start by solving the problem on a coarse mesh, without an a-priori decision on the number of levels or area of refinement. The assembly and solving of the detail parts can be done per level. Therefore, on each level a decision can be made on whether and where to refine.

Note that the same cannot be said for the HB method. For this method, an explicit construction of the multi-resolution stiffness matrix is expensive, and when details are added, the coarser solution needs to be re-computed. With an OCWB method, we can base refinement decisions on a-posteriori error detection. In the application of Chapter 7, we make the assumption that all details are nested over the levels. This means that we do not need to refine in areas where we have no coarser refinement, or where the computed coarser details were small enough. In that case, we can on each level j make an appropriate refinement decision based upon the previously computed details of the coarser level $j-1$. Convergence did not weaken under this assumption, for the problems in Chapter 7. This of course would be an incorrect assumption for problems were high-frequency details pop up in the solution, away from areas with coarser details. Note also that an operator-orthogonal wavelet basis is a full basis for the numerical error. Indeed, consider a coarse level j solution, and an OCWB of which we can add functions without having to re-compute the coarser solution. If the FE method converges, we know that this wavelet basis, from level j to ∞, must span exactly the difference between the coarse solution and the true solution, which is the numerical error. This is not the case for a HB basis, where the wavelets only together with the level j scaling functions span the numerical error, as well as the true solution for that matter. The operator-orthogonal wavelet basis spans the numerical error on its own.

6.4 Conclusion

We compared the computational complexity of our *Operator-Customized Wavelet Basis* (OCWB) FEM with the complexity of a *Hierarchical Basis* (HB) FEM and of a standard single-scale FEM. The cost of the solution phase proved to be the critical part of the total complexity for all three methods. We used both direct (Gaussian elimination) and iterative (Conjugate Gradient) solvers in our analysis. For strongly adaptive problems, problems with telescopic refinement in particular, the OCWB method achieves a truly optimal solving complexity of $O(J)$, where J is the number of refinement levels. This cannot be achieved by a HB method. We illustrated the theoretical results with the computation of a two-dimensional Green's function on a bounded domain. For this problem telescopic refinement yields good convergence. We found the adaptive OCWB to significantly outperform other adaptive methods such as HB. Even the non-adaptive OCWB slightly beats the non-adaptive HB, while well outperforming a single-scale FEM.

In the following chapter, we will apply an adaptive OCWB to a barrier option pricing problem, governed by a more general PDE, to demonstrate our method's breadth and power.

7. Application: Barrier Option Pricing

7.1 Barrier Option Pricing Problem

We now apply our wavelet method to a barrier-option pricing problem, in particular an up-and-out call option. We will customize a wavelet basis to scale-decouple an operator consisting of a diffusion, convection and reaction term with varying coefficients. An adaptive method can exploit the local concentration of error in the solution, to achieve critically fast and accurate pricing.

A call option (see e.g. Hull, 1989) is a contract that gives the owner the right but not the obligation to purchase an underlying asset at a pre-specified price, the *strike price*, at a given time in the future, the *option maturity*. The owner will only exercise his option at maturity if the option is *in-the-money* – if the asset price is higher than the strike price. The value of the option at maturity is straightforward: it is zero if the underlying asset is cheaper than the strike price and increases proportional to the asset price for asset prices above the strike price. The precise value of the option contract, *option price*, at any time before maturity can only be determined by making assumptions on the behavior of the underlying asset over time, and the risk-free interest rate. The *up-and-out barrier* feature imposes a constraint on the option contract: the contract will be void if the underlying asset price exceeds the barrier price any time before or at maturity. An option with special features such as a barrier is called an *exotic* option, as opposed to a regular *vanilla* option. For the pricing problem in this dissertation, we will assume for simplicity an underlying asset that does not pay dividends.

Options are commonly used in the financial world to hedge out asset risks, or to speculate in a leveraged fashion. Barrier options (with a reasonably placed barrier) have the benefit of being significantly cheaper, as the option ceases to exist when the underlying asset crosses a certain threshold. Note that an up-and-out barrier on a call option effectively cuts off a part of the option's payout where the payout is the largest, though in most cases unlikely to be realized. If, for example, a speculator believes the underlying asset price will not cross the barrier, he could buy this barrier option instead of a regular vanilla option without barrier, and save a lot of money to place his bet.

The underlying assets could be stock, indices, interest rates, foreign exchange rates, or anything that is liquid enough and carries sufficient investor interest.

Before the advent of close replication strategies, for each option contract two parties with opposite interests were needed. Currently, however, many option contracts are at one side neutralized by a replication strategy, attempting to exactly reproduce the option's payout at the time of exercise. The cost of such a replication strategy determines the fair price of that option. Unfortunately, any replication strategy, and therefore also the option value, relies on assumptions on the time-behavior of the underlying asset. Thus, your price is only as good as your assumptions. In addition, pricing also depends on the achieved accuracy of any numerical method used to solve the pricing problem, which in general depends on time. The more exact the price, the tighter the spread for the party knowing this price, the higher stable business and/or margin profits for that party. As a result, a fast and good pricing method is desirable.

In a first approach, we assume that the price of the underlying asset, *S*, follows a continuous geometric random walk:

$$dS = \mu S dt + \sigma S dw$$

In this equation, *w* denotes a Wiener process. We also assume a world without transaction costs or other trading frictions. This pricing model is called the Black-Scholes model (see Black and Scholes, 1972). The asset price S and the option price $u(S,t)$ are stochastic, whereas the asset volatility $\sigma(S,t)$, the asset drift $\mu(t)$, and the risk-free interest rate $r(t)$ are deterministic. Under these assumptions, the option price satisfies the following Black-Scholes partial differential equation:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 u}{\partial S^2} + rS\frac{\partial u}{\partial S} - ru = 0 \tag{7.1}$$

Note that the option price does not depend on the drift $\mu(t)$. The final condition of this partial differential equation is given by the option payout at maturity:

$$u(S,T) = \begin{cases} 0 & 0 \le S < K \\ S - K & K \le S < B \end{cases} \tag{7.2}$$

We assume that the option contract is void when the underlying asset becomes worthless:

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

Finally, the up-and-out barrier effectively imposes another boundary condition on the option price:

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

Equation (7.1) with final condition (7.2) and boundary conditions has a closed-form analytical solution for constant coefficients *r* and σ (see e.g. Haug, 1998):

$$\begin{aligned} S \le B: \quad u(S,t) = {} & S\left(N(d_1) - N(d_2)\right) \quad - S\left(\frac{B}{S}\right)^{\frac{2r}{\sigma^2}+1}\left(N(d_3) - N(d_4)\right) \\ & - Ke^{-r(T-t)}\left(N\left(d_1 - \sigma\sqrt{T-t}\right) - N\left(d_2 - \sigma\sqrt{T-t}\right)\right) \\ & + Ke^{-r(T-t)}\left(\frac{B}{S}\right)^{\frac{2r}{\sigma^2}-1}\left(N\left(d_3 - \sigma\sqrt{T-t}\right) - N\left(d_4 - \sigma\sqrt{T-t}\right)\right) \\ S > B: \quad u(S,t) = {} & 0 \end{aligned} \tag{7.3}$$

In these equations $N(d)$ is the cumulative normal distribution function:

$$N(d)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{d}e^{-\frac{1}{2}s^2}ds$$

with

$$d_1=\left(\ln\left(\frac{S}{K}\right)+\left(r+\frac{\sigma^2}{2}\right)(T-t)\right)/\left(\sigma\sqrt{T-t}\right)$$

$$d_2=\left(\ln\left(\frac{S}{B}\right)+\left(r+\frac{\sigma^2}{2}\right)(T-t)\right)/\left(\sigma\sqrt{T-t}\right)$$

$$d_3=\left(\ln\left(\frac{B^2}{SK}\right)+\left(r+\frac{\sigma^2}{2}\right)(T-t)\right)/\left(\sigma\sqrt{T-t}\right)$$

$$d_4=\left(\ln\left(\frac{B}{S}\right)+\left(r+\frac{\sigma^2}{2}\right)(T-t)\right)/\left(\sigma\sqrt{T-t}\right)$$

We do not have an analytical solution for time-dependent or asset-price-dependent coefficients $r(t)$ and $\sigma(S,t)$. For these problems we may use statistical methods such as Monte Carlo simulations, or numerical methods such as Binomial Trees, the Finite Difference Method (see e.g. Betaneli, 1998), or the *Finite Element Method* (FEM). We will focus in this dissertation on the FEM (see Winkler *et al.*, 2002), a fast and flexible method that can also be applied to higher-dimensional problems, such as Heston's model described in the next paragraph.

For vanilla *at-the-money* options – options with an underlying asset price close to the strike price – the Black-Scholes model performs very well. Most actively traded options in the market are at-the-money or not far away from that. For exotic options such as an up-and-out call, however, it is well known that the strict assumption of a deterministic volatility can lead to large pricing errors, even at-the-money.

Therefore, we now relax the assumption of a deterministic volatility $\sigma(S,t)$ to allow for a stochastic variance y. Instead of a Black-Scholes model, we have the following stochastic volatility model (Cox *et al.*, 1985):

$$dS=\mu Sdt+\sqrt{y}Sdw_1$$

$$dy=\kappa(\theta-y)dt+\xi\sqrt{y}dw_2$$

$$\operatorname{cov}[dw_1,dw_2]=\rho dt$$

In this equation, w_1 and w_2 denote two Wiener processes with correlation ρ. Here too, we assume a world without transaction costs or other trading frictions. The asset price S, the asset variance y, and the option price $u(S,y,t)$ are stochastic. The asset drift $\mu(t)$, long-term variance θ, rate of mean-reversion in variance κ, variance volatility ξ, the correlation ρ, and the risk-free interest rate $r(t)$ are deterministic. Under these assumptions, named Heston's model, the option price satisfies the following partial differential equation (Heston, 1993):

$$\frac{\partial u}{\partial t}+\frac{1}{2}\xi^2 y\frac{\partial^2 u}{\partial y^2}+\rho\xi yS\frac{\partial^2 u}{\partial y\partial S}+\frac{1}{2}yS^2\frac{\partial^2 u}{\partial S^2}+\kappa(\theta-y)\frac{\partial u}{\partial y}+rS\frac{\partial u}{\partial S}-ru=0 \tag{7.4}$$

Note that the option price again does not depend on the drift $\mu(t)$. We keep the following final and boundary conditions, imposed by the option contract:

$$\begin{aligned} &u(S,y,T)=\begin{cases}0 & 0\le S<K\\ S-K & K\le S<B\end{cases}\\ &u(0,y,t)=0\\ &u(B,y,t)=0 \end{aligned} \tag{7.5}$$

Equation (7.4) with conditions (7.5) has no closed-form analytical solution. As in the one-dimensional case, we may solve these problems with statistical methods such as Monte Carlo simulations, or numerical methods such as the *Finite Difference Method* (FDM) (see e.g. Kluge, 2002), or the *Finite Element Method* (FEM) (see Winkler *et al.*, 2002). Monte Carlo simulations generally are too slow, while the FDM can be difficult to implement on irregular-spaced meshes, caused by the exotic features of some options. In addition, the weak formulation of the FEM requires fewer smoothness constraints on the final conditions and boundary conditions. Since the barrier feature of the option causes a discontinuous, non-smooth final condition, we choose the FEM for Heston's model as well.

7.2 Operator-Customized Wavelet Basis FEM

We can numerically solve Equations (7.1) or (7.4) by first choosing an implicit discrete time-stepping scheme over t:

$$\frac{\partial u}{\partial t} = \frac{u^{t-1} - u^{t}}{\Delta t}$$
$$\frac{\partial^{n+m} u}{\partial S^{n} \partial y^{m}} = \frac{\partial^{n+m} u^{t-1}}{\partial S^{n} \partial y^{m}} \quad \forall n,m : n+m \leq 2, n \geq 0, m \geq 0$$

Then, we may use a FEM for the spatial coordinates S, or S and y respectively. Note that Equations (7.1) and (7.4) could be simplified by transforming the variable S:

$$x = \ln\left(\frac{S}{B}\right)$$

However, with the large local error caused by the discontinuity in our barrier option payoff at $S = B$, such variable transform significantly slows the FE convergence. We have the following system matrix entries in the spatial coordinate S, corresponding to the weak form of Equation (7.1):

$$a_{t-1}\left(\varphi_{trial}, \varphi_{test}\right) = \left(\int_{\Omega} \frac{1}{2} \sigma_{t-1}{}^{2} S^{2} \frac{\partial \varphi_{trial}}{\partial S} \frac{\partial \varphi_{test}}{\partial S} + \left(r_{t-1} - \sigma_{t-1}{}^{2}\right) S \frac{\partial \varphi_{trial}}{\partial S} \varphi_{test} - r_{t-1} \varphi_{trial} \varphi_{test} dS \right) \Delta t$$
$$+ \int_{\Omega} \varphi_{trial} \varphi_{test} dS \quad \forall t \qquad (7.6)$$

For Equation (7.4), we have instead an expression in spatial coordinates S and y:

$$a_{t-1}\left(\varphi_{trial}, \varphi_{test}\right) = \left(\iint_{\Omega} \begin{bmatrix} \frac{\partial \varphi_{trial}}{\partial S} & \frac{\partial \varphi_{trial}}{\partial y} \end{bmatrix} \mathbf{P} \begin{bmatrix} \frac{\partial \varphi_{test}}{\partial S} \\ \frac{\partial \varphi_{test}}{\partial y} \end{bmatrix} + \begin{bmatrix} \frac{\partial \varphi_{trial}}{\partial S} & \frac{\partial \varphi_{trial}}{\partial y} \end{bmatrix} \mathbf{q}_{t-1} \varphi_{test} - r_{t-1} \varphi_{trial} \varphi_{test} dS dy \right) \Delta t$$
$$+ \iint_{\Omega} \varphi_{trial} \varphi_{test} dS dy \quad \forall t \qquad (7.7)$$

The coefficients in this equation are given by:

$$\mathbf{P} = \frac{1}{2} \begin{bmatrix} S^{2} y & \rho \xi S y \\ \rho \xi S y & \xi^{2} y \end{bmatrix} \qquad \mathbf{q}_{t-1} = \begin{bmatrix} \left(r_{t-1} - y - \frac{\xi \rho}{2} \right) S \\ \kappa \theta - \frac{\xi^{2}}{2} - \left(\kappa + \frac{\xi \rho}{2} \right) y \end{bmatrix}$$

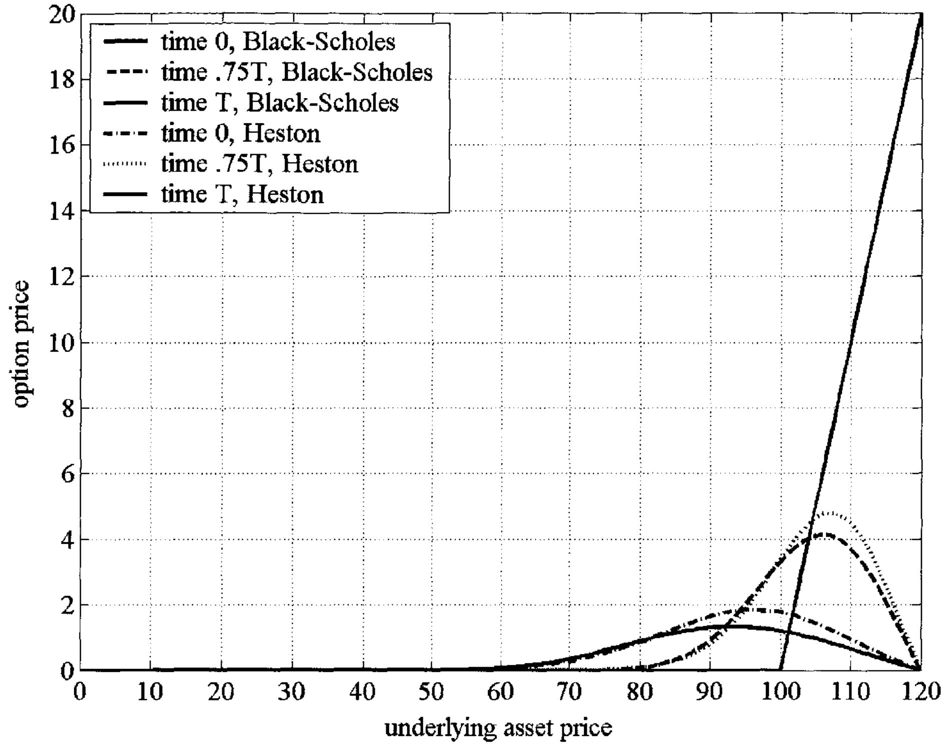


Figure 7-1: Option price vs. underlying asset price, at different times.

In general, both expressions will be dependent on t, but in our further example we will take the coefficients in Equations (7.1) and (7.4) constant in time. Linear hat – first-order Lagrangian – basis functions are chosen as FE test functions for the numerical solution, as well as trial functions. Since Heston's model does not impose any boundary constraints on y, we choose for y a lower boundary near $y = 0$ and a sufficiently high upper boundary, and restrain those boundaries with Neumann conditions, satisfying the weak form of Equation (7.4) on the boundary. Per time step, we will solve the following system for $\mathbf{u}_J^{t-1}$:

$$\begin{aligned}&\left(\mathbf{K}_J^{t-1}\Delta t + \mathbf{M}_J\right)\mathbf{u}_J^{t-1} = \mathbf{M}_J\mathbf{u}_J^{t}\\ &\Leftrightarrow \mathbf{A}_J^{t-1}\mathbf{u}_J^{t-1} = \mathbf{M}_J\mathbf{u}_J^{t}\end{aligned} \tag{7.8}$$

We work backward in time from the final condition $\mathbf{u}_J^t = \mathbf{u}^\#$ to $\mathbf{u}_J^0$ at time 0. For both the one-dimensional and the two-dimensional problem, the time step is chosen in function of the smallest spatial resolution (Equation (7.9)):

Black-Scholes	Heston		
1.1789	N/A		analytic value
1	1	T	option maturity
100	100	K	option strike
120	120	B	option barrier
0.1	0.1	r	risk-free rate
100	100	S	asset price
0	0	t	time
0.2	stochastic	σ	asset volatility
N/A	2.5	κ	variance mean-reversion rate
N/A	0.04	θ	long-term variance
N/A	0.5	ξ	variance volatility
N/A	-0.1	ρ	correlation

Table 7-1: Parameters used in the Black-Scholes and Heston model.

level	0	1	2	3	4	5	6	7	8
N	12	23	45	89	177	353	705	1409	2817
ΔS	10	5	2.5	1.25	0.625	0.313	0.156	0.078	0.039
Δt	1year	3months	23days	6days	34hours	9hours	128min	32min	8min
price	1.7475	1.4487	1.2526	1.1975	1.1836	1.1801	1.1792	1.1790	1.1789

Table 7-2: Numerical solutions of Black-Scholes PDE with Table 7-1 parameters.

level	0	1	2	3	4	5
N	84	299	1125	4361	17169	68129
ΔS	10	5	2.5	1.25	0.625	0.313
Δt	1year	3months	23days	6days	34hours	9hours
price	2.0422	1.9021	1.8045	1.7714	1.7596	1.7554

Table 7-3: Numerical solutions of Heston PDE with Table 7-1 parameters.

$$\Delta t = \frac{(\Delta S)^2}{100} \tag{7.9}$$

We assume an a-priori selection of the number of levels of refinement J, in order to fix the time step Δt in Equation (7.9). The mesh is regular-spaced, with right triangles for the two-dimensional problem. The coarsest – level 0 – mesh is eleven elements long in S and six elements wide in y. Note that we fixed the lower boundary for S at $S = 10$. Table 7-1 contains all the parameters chosen for our barrier option pricing problem. The solutions corresponding to different resolutions (number of levels of refinement) in Table 7-2 and Table 7-3 show the convergence for the non-adaptive one-dimensional and two-dimensional FEM problems. The FEM for the Black-Scholes model converges to the known analytical solution of 1.1789, computed with Equation (7.3). The FEM for Heston's model converges to a price of around 1.754, well above the Black-Scholes price. We do not have an analytical solution to the Heston's model, but we believe that this significant difference is caused by the different model assumptions. Figure 7-1 shows the option price as a function of the underlying asset at time zero, for the Black-Scholes model as well as the Heston model. The negative correlation between asset price and stochastic variance means that if the option becomes more in-the-money the volatility is likely to decrease, which reduces the risk of hitting the barrier. Reversely, when the option becomes out-of-the-money we are more likely to see bigger moves, which can only help us getting back in-the-money. This is one of the causes of the Heston model's higher price. Note that with a choice of parameters that eliminate the volatility's stochasticity ($\kappa = 0.0001$, $\xi = 0$ and $\rho = 0$), the two-dimensional FEM yields solutions that swiftly converge to those of the one-dimensional problems of equal spatial resolution, as given in Table 7-2.

Instead of a single-level FEM, we choose a multi-resolution basis for the trial and test functions. This produces an identical FE solution, but also allows for an adaptive reduction of the problem's dimensions. With a Hierarchical Basis, we would have a

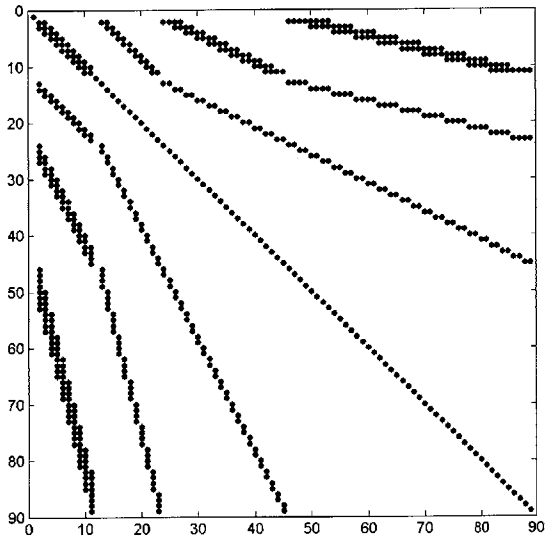


Figure 7-2: Four-level HB system or mass matrix for non-adaptive Black-Scholes model.

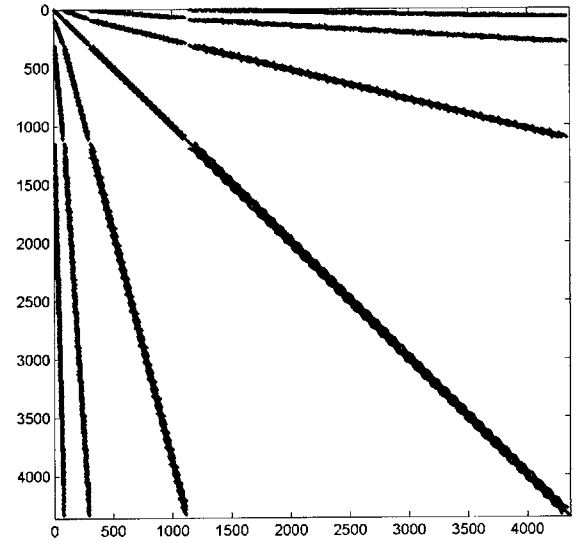


Figure 7-3: Four-level HB system or mass matrix for non-adaptive Heston model.

multi-resolution system matrix, fully coupled between scales. For example, for a two-level approach we have:

$$
\begin{aligned}
&\mathbf{A}_J^{t-1}\mathbf{u}_J^{t-1} = \mathbf{M}_J\mathbf{u}_J^t \\
&\Leftrightarrow \tilde{\mathbf{T}}_{J-1}\mathbf{A}_J^{t-1}\tilde{\mathbf{T}}_{J-1}^T\mathbf{T}_{J-1}\mathbf{u}_J^{t-1} = \tilde{\mathbf{T}}_{J-1}\mathbf{M}_J\tilde{\mathbf{T}}_{J-1}^T\mathbf{T}_{J-1}\mathbf{u}_J^t \\
&\Leftrightarrow \mathbf{A}_J^{t-1[J-1]}\mathbf{u}_J^{t-1[J-1]} = \mathbf{M}_J^{[J-1]}\mathbf{u}_J^{t[J-1]} \\
&\begin{bmatrix} \mathbf{A}_{J-1}^{t-1} & \mathbf{C}_{J-1}^{\mathbf{A},t-1} \\ \mathbf{D}_{J-1}^{\mathbf{A},t-1} & \boldsymbol{\Delta}_{J-1}^{\mathbf{A},t-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_{J-1}^{t-1} \\ \boldsymbol{\gamma}_{J-1}^{t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{J-1} & \mathbf{C}_{J-1}^{\mathbf{M}} \\ \mathbf{D}_{J-1}^{\mathbf{M}} & \boldsymbol{\Delta}_{J-1}^{\mathbf{M}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_{J-1}^{t} \\ \boldsymbol{\gamma}_{J-1}^{t} \end{bmatrix} \\
&\text{with } \mathbf{C}_{J-1}^{\mathbf{A},t-1}, \mathbf{D}_{J-1}^{\mathbf{A},t-1}, \mathbf{C}_{J-1}^{\mathbf{M}}, \mathbf{D}_{J-1}^{\mathbf{M}} \neq \mathbf{0}
\end{aligned}
$$

The sparsity patterns of the multi-resolution system matrix $\mathbf{A}_J^{t-1[0]}$ and mass matrix $\mathbf{M}_J^{[0]}$ (in full multi-scale format, instead of only two-level) are shown in Figure 7-2 for the one-dimensional and Figure 7-3 for the two-dimensional problem respectively. However, by choosing a wavelet basis – spanning the first-order Lagrangian finite element space – that is customized to the respective operators of Equation (7.1) and (7.4), we will decouple the multi-resolution system matrix across scales. We refer to this method as an *Operator-Customized Wavelet Basis* (OCWB) FEM. Such decoupling will be optimal for problems that allow for telescopically adaptive refinement.

Because of the asymmetry in the weak form (see Equations (7.6) and (7.7)), and consequently the asymmetry in our FE system matrix, we need a different customization

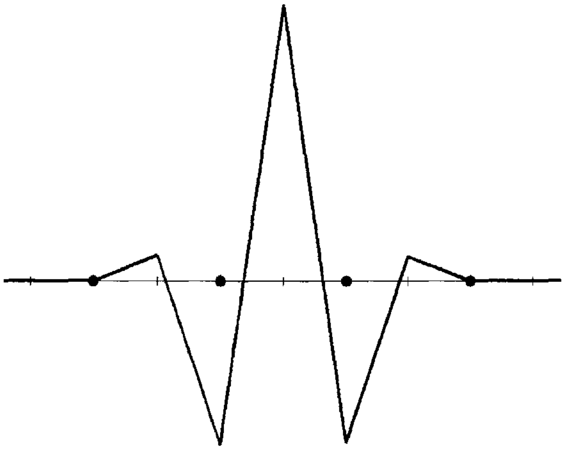


Figure 7-4: First-order FE wavelet customized to Black-Scholes operator.

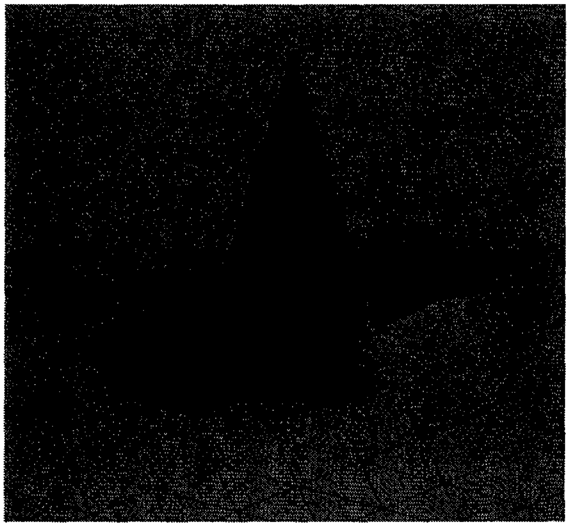


Figure 7-5: First-order FE wavelet customized to Heston operator.

for the FE test functions than the trial functions, in order to achieve full scale-decoupling in both the upper-right and lower-left of the system matrix. We choose our trial function wavelets to be operator-orthogonal to the scaling functions with respect to the weak form:

$$a_{t-1}\left(\psi_j^{t-1},\phi_j^{t-1}\right)=0 \quad \forall j,t$$

We refer to Chapter 5, for the construction of a compact wavelet basis with this property. The dual wavelet transform matrices $\tilde{\mathbf{T}}_j$ contain the filters associated with these customized wavelets. The wavelets have each a support that spans the support of two neighboring scaling functions, away from the boundary. Examples of these wavelets, for the parameters of Table 7-1, are shown in Figures 7-4 and 7-5. This wavelet choice eliminates all upper-right hand coupling in the multi-resolution matrix. The test functions are chosen to be operator-orthogonal with respect to the transpose of the weak form:

$$a_{t-1}\left(\phi_j^{t-1},\psi_j^{t-1}\right)=0 \quad \forall j,t \tag{7.10}$$

Consequently, the customized test function wavelets are built with another set of dual wavelet transforms $\tilde{\mathbf{T}}_j'$. They have the same support as the customized trial function

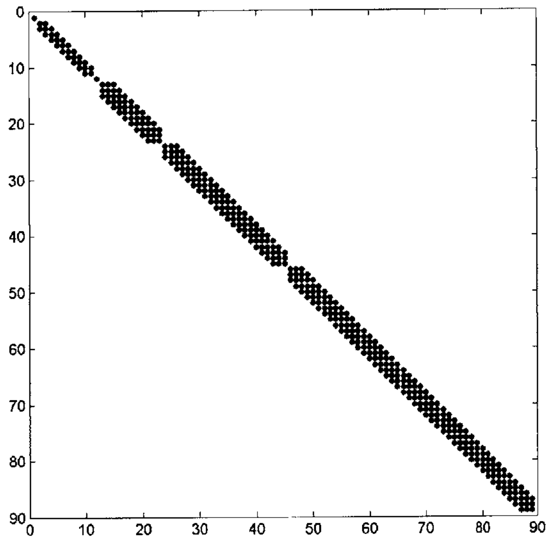


Figure 7-6: Four-level OCWB system matrix for non-adaptive Black-Scholes model.

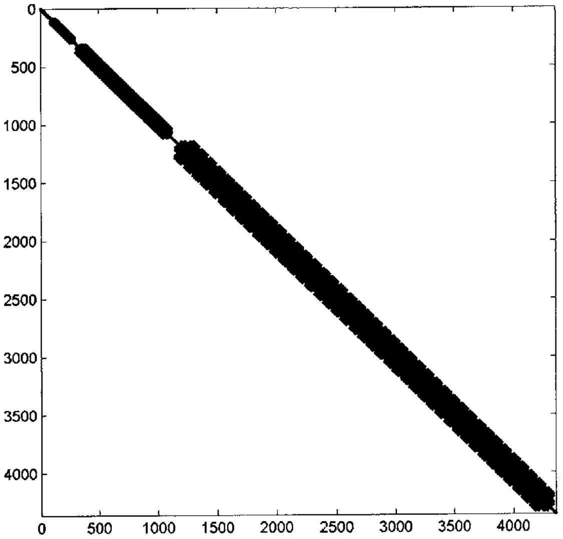


Figure 7-7: Four-level OCWB system matrix for non-adaptive Heston model.

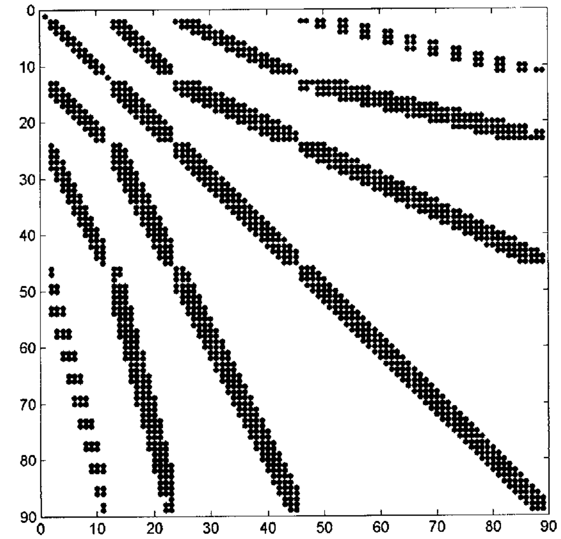


Figure 7-8: Four-level OCWB mass matrix for non-adaptive Black-Scholes model.

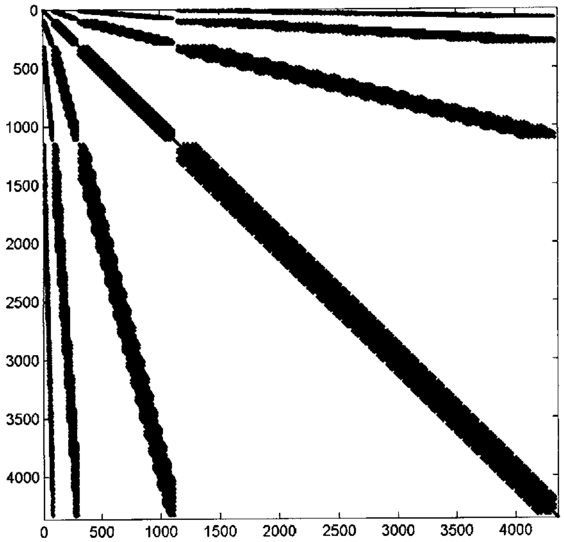


Figure 7-9: Four-level OCWB mass matrix for non-adaptive Heston model.

wavelets, and a similar shape. For a non-adaptive method both customized bases still span the same space, and thus we have effectively a Ritz-Galerkin method.

Note that alternatively we could have chosen for the test functions a primitive wavelet basis (HB). This would have resulted in a system matrix scale-decoupled in the upper-right while scale-coupled in the lower-left, which we still may solve as a scale-decoupled system, iterating from coarse to finer scales. However, this would be less easy to implement for adaptive methods, as the wavelets would have different supports, leading to a difference in wavelet number overlapping with any given refinement zone. For a two-level multi-resolution system, we now have:

$$\begin{aligned}
&\mathbf{A}_J^{t-1}\mathbf{u}_J^{t-1} = \mathbf{M}_J\mathbf{u}_J^t \\
&\Leftrightarrow \tilde{\mathbf{T}}_{J-1}^{\prime t-1}\mathbf{A}_J^{t-1}\left(\tilde{\mathbf{T}}_{J-1}^{t-1}\right)^T\mathbf{T}_{J-1}^{t-1}\mathbf{u}_J^{t-1} = \tilde{\mathbf{T}}_{J-1}^{\prime t-1}\mathbf{M}_J\left(\tilde{\mathbf{T}}_{J-1}^{t-1}\right)^T\mathbf{T}_{J-1}^{t-1}\mathbf{u}_J^t \\
&\Leftrightarrow \mathbf{A}_J^{t-1[J-1]}\mathbf{u}_J^{t-1[J-1]} = \mathbf{M}_J^{t-1[J-1]}\mathbf{u}_J^{t[J-1]} \\
&\begin{bmatrix} \mathbf{A}_{J-1}^{t-1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Delta}_{J-1}^{\mathbf{A},t-1} \end{bmatrix}\begin{bmatrix} \boldsymbol{\lambda}_{J-1}^{t-1} \\ \boldsymbol{\gamma}_{J-1}^{t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{J-1} & \mathbf{C}_{J-1}^{\mathbf{M},t-1} \\ \mathbf{D}_{J-1}^{\mathbf{M},t-1} & \boldsymbol{\Delta}_{J-1}^{\mathbf{M},t-1} \end{bmatrix}\begin{bmatrix} \boldsymbol{\lambda}_{J-1}^{t} \\ \boldsymbol{\gamma}_{J-1}^{t} \end{bmatrix} \\
&\text{with } \ \mathbf{C}_{J-1}^{\mathbf{M},t-1}, \mathbf{D}_{J-1}^{\mathbf{M},t-1} \neq \mathbf{0}
\end{aligned}$$

The multi-resolution matrices $\mathbf{A}_J^{t-1[0]}$ and $\mathbf{M}_J^{t-1[0]}$ (in full multi-resolution format) are shown in Figures 7-6 and 7-8 for the Black-Scholes model and in Figures 7-7 and 7-9 for the Heston model respectively. Instead of using just a two-level approach, we will benefit from a full multi-resolution approach (see Chapter 3):

$$\mathbf{A}_J^{t-1[0]}\mathbf{u}_J^{t-1[0]} = \mathbf{M}_J^{t-1[0]}\mathbf{u}_J^{t[0]} \tag{7.11}$$

Achieving scale-decoupling in system matrix $\mathbf{A}_J^{t-1[0]}$ results in a coupled mass matrix $\mathbf{M}_J^{t-1[0]}$. We believe it is impossible to decouple both at the same time. Consequently, the details on each particular level are affected by the coarser solution and lower-level details from previous time-steps. Thus, for each time step we need to solve for all levels of refinement, instead of being able to generate an answer per individual scale.

To start the iteration over time, we have to input the final condition $\mathbf{u}^{\#}$ on the right-hand-side of Equation (7.11). It would be too expensive to input this final condition in multi-resolution format. Indeed, transforming the single-scale final condition to multi-resolution format would involve applying the non-sparse wavelet transform matrices $\mathbf{T}_J^{t-1}$, with a higher than $O(N)$ operation cost (see Chapter 6), to the sampled final condition. Instead, we can apply a mass matrix $\mathbf{M}_J^{t-1(0)}$ specifically for multi-resolution test functions and single-level trial functions, and apply this to the final condition in single-level form:

$$\mathbf{A}_J^{t-1[0]}\mathbf{u}_J^{t-1[0]} = \mathbf{M}_J^{t-1[0)}\mathbf{u}_J^t \qquad \text{where} \quad \mathbf{M}_J^{t-1[0)} = \tilde{\mathbf{T}}_0^{\prime t-1}\ldots\tilde{\mathbf{T}}_{J-1}^{\prime t-1}\mathbf{M}_J$$

This operation has an $O(N)$ cost. We directly compute the integrals of the test wavelets against the final condition, without explicitly forming $\mathbf{M}_J^{t-1[0)}$. We choose to integrate the wavelets against the 'true' final condition, including the discontinuity at the barrier, and not against its approximation as a continuous function of the trial function space. As we will see further, this yields good convergence, even though the final condition is – unlike the FE solution – not contained in H^1 because of the discontinuity. To start the next and subsequent time steps, we can directly plug-in the previous multi-resolution solution $\mathbf{u}_J^{t[0]}$ in Equation (7.11), without any transform. The full multi-resolution mass matrix $\mathbf{M}_J^{t-1[0]}$ is applied to the solution vector. This operation will be $O(N)$, if we do not form $\mathbf{M}_J^{t-1[0]}$ explicitly, but apply it as a series of dual wavelet transforms and sparse single-scale $\mathbf{M}_J$:

$$\mathbf{M}_J^{t-1[0]}\mathbf{u}_J^{t[0]}=\tilde{\mathbf{T}}_0^{\prime t-1}\ldots\tilde{\mathbf{T}}_{J-1}^{\prime t-1}\mathbf{M}_J\left(\tilde{\mathbf{T}}_{J-1}^{t-1}\right)^T\ldots\left(\tilde{\mathbf{T}}_0^{t-1}\right)^T\mathbf{u}_J^{t[0]}$$

After the last time step, we can transform the multi-resolution solution back into single-scale format, by simply applying a series of dual wavelet transforms:

$$\mathbf{u}_J^0=\left(\tilde{\mathbf{T}}_{J-1}^0\right)^T\ldots\left(\tilde{\mathbf{T}}_0^0\right)^T\mathbf{u}_J^{0[0]} \tag{7.12}$$

This is in total an $O(N)$ operation, as discussed in Chapter 6. If we would be pricing an option that can be exercised before maturity – a so-called *American* option –, we would have to threshold the solution samples at each time-step. This cannot be done in multi-resolution format. However, for such pricing problems, we could every time-step transform the multi-resolution solution into single-scale format, as in Equation (7.12), with only $O(N)$ cost. The single-scale solution $\mathbf{u}_J^{t-1}$ can then be compared to another sample. For example, for an American option, we should threshold the solution against the final condition $\mathbf{u}^{\#}$ – the option's value on immediate exercise:

$$\mathbf{u}_J^{t-1} = \max\left(\mathbf{u}_J^{t-1},\mathbf{u}^{\#}\right)$$

Then, we apply the special mass matrix $\mathbf{M}_J^{t-1[0)}$ directly on the single-scale solution to obtain the right hand side of Equation (7.11). Indeed, it would be too expensive to transform the solution first to multi-resolution format. Note that in our example, we do not need to follow this longer procedure, since the option cannot be exercised before maturity.

When the parameters in Equations (7.1) or (7.4) are constant over time, the system matrix is identical for every time step. In that case, we only need to solve the system of Equation (7.11) one time, for example using an LU decomposition, and we can use this result to cheaply solve for all other time steps. Most applications, however, are expected to result in a time-dependent system that needs to be solved for each time step. Indeed, for time-invariant parameters, Equation (7.1) has an analytical closed-form solution, and a numerical method would not even be required.

7.3 Adaptive Method

We can further exploit the local concentration of error in the solution, caused by the discontinuity in the final condition. Indeed, we will use an adaptive method – with only local refinements where needed – instead of a non-adaptive method – with refinement everywhere. Such adaptivity can be easily implemented with a multiresolution wavelet basis. In addition, we can achieve a faster solution speed by using operator-orthogonal wavelet refinements, resulting in a scale-decoupled system matrix. As discussed in Chapter 6, this method works best when we have telescopic adaptivity, that is when we can keep constant the number of details added on each level of refinement. Then, we can solve the problem with only order $O(J)$ cost, where J is the number of levels of refinement. Note that we will rely on modified single-scale system matrices $\mathbf{A}_J^{*t-1}$ and a modified single-scale mass-matrix $\mathbf{M}_J^{*}$, as discussed in Chapter 6, to keep low the operation cost of the dual wavelet transforms. Our results show that for both the one-dimensional problem and the two-dimensional problem the solution details are local, nested over the levels and fast-decaying. When details are nested over the levels, i.e. when details will not surface in an area where there are no coarser details, we can on each level decide where to refine based on the coarser solution details. When we have scale-decoupling in particular, the coarser solution details do not depend on the finer

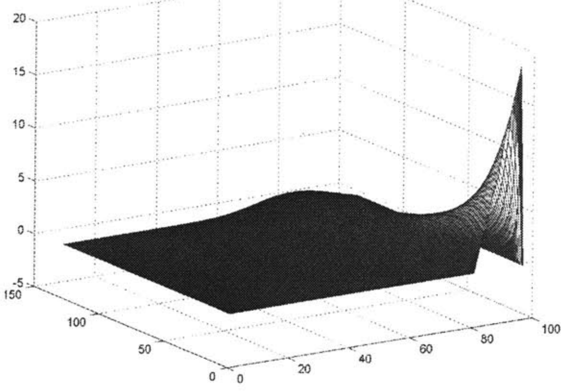


Figure 7-10: Adaptive Black-Scholes solution, function of asset price and time.

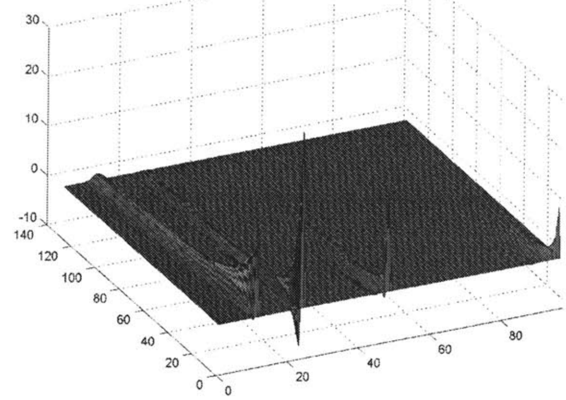


Figure 7-11: Corresponding scaling, wavelet (level 0, 1, 2) coefficients, in time.

details we are planning to add. This supports an adaptive refinement method for which on each level we only add details $\gamma_{j+1,m}^{t-1}$ in areas with significant coarser details $\gamma_{j,m}^{t-1}$.

We could choose to apply a diagonal pre-conditioner $\mathbf{D}^{t-1}$ to the multi-resolution system. Such pre-conditioner could be merged with the dual wavelet transforms, effectively inserting the re-scaling into the filters of the Wavelet and the Scaling Equation. For notational simplicity, we show the application of $\mathbf{D}^{t-1}$ to a two-level system:

$$\mathbf{D}'^{t-1}\tilde{\mathbf{T}}'^{t-1}_{J-1}\mathbf{A}_J^{t-1}\left(\tilde{\mathbf{T}}^{t-1}_{J-1}\right)^T\mathbf{D}^{t-1}\left(\mathbf{D}^{t-1}\right)^{-1}\mathbf{T}^{t-1}_{J-1}\mathbf{u}_J^{t-1}=\mathbf{D}'^{t-1}\tilde{\mathbf{T}}'^{t-1}_{J-1}\mathbf{M}_J\left(\tilde{\mathbf{T}}^{t-1}_{J-1}\right)^T\mathbf{D}^{t-1}\left(\mathbf{D}^{t-1}\right)^{-1}\mathbf{T}^{t-1}_{J-1}\mathbf{u}_J^{t}$$

We have not applied rescaling to the wavelet functions. We use a cut-off threshold χ_j to determine which coarser details $\gamma_{j,m}^{t-1}$ are significant enough to call for further refinement. The threshold decreases each level j, to facilitate convergence. For the one-dimensional problem, we apply a threshold of $\chi_j = 4^{-J}\tau^{-j}$ with a base of $\tau = 1.1$, to compute a solution with finest resolution J. We chose a base of 4 for the finest resolution, because the non-adaptive solution error converges at that rate. Since the two-dimensional non-adaptive solution converges at a slightly slower rate, we impose a threshold of $\chi_j = 3^{-J}\tau^{-j}$ with the same base of $\tau = 1.1$, to compute an adaptive solution for the two-dimensional problem. Under these parameters, we achieve similar convergence between the adaptive and non-adaptive methods.

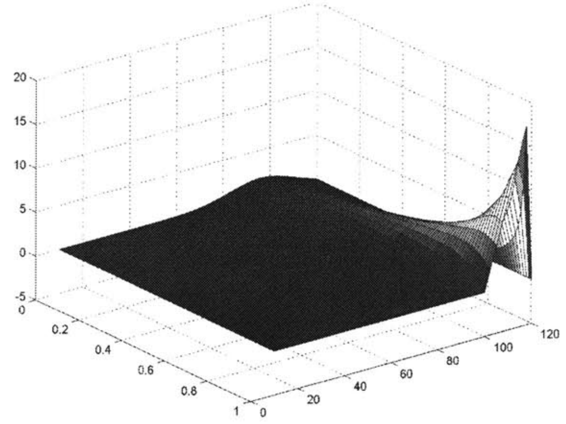


Figure 7-12: Adaptive Heston solution, function of asset price and time.

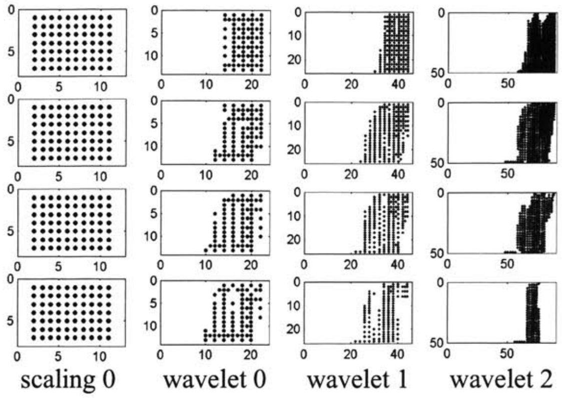


Figure 7-13: Significant OCWB scaling, wavelet coefficients; times *T*, 2*T*/3, *T*/3, 0.

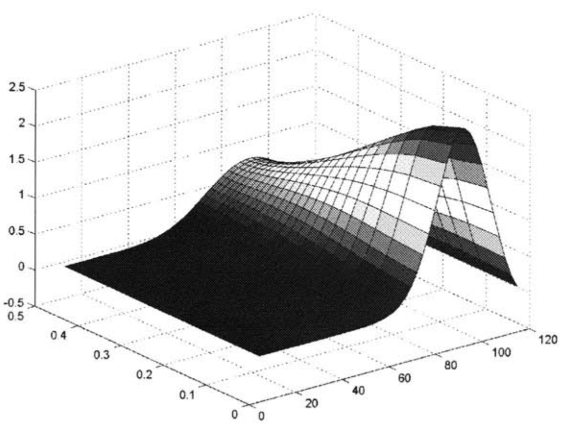


Figure 7-14: Adaptive Heston solution, function of asset price and volatility, at time 0.

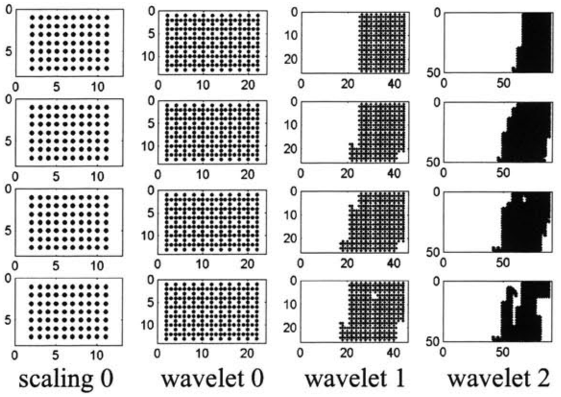


Figure 7-15: Computed OCWB scaling, wavelet coefficients; times *T*, 2*T*/3, *T*/3, 0.

In Figure 7-10 the adaptive solution of the one-dimensional Black-Scholes Equation is plotted, as a function of the underlying asset price and time. For every level, only a few of the OCWB detail functions are computed, depending on the significance of the coarser details. Figure 7-11 shows the fast decay of the wavelet coefficients over time, over the asset price away from the barrier, and with increasing level of refinement. The coefficients correspond to a level 3 customized wavelet basis, consisting of level 0 scaling functions, and level 0, 1 and 2 customized wavelets. Similarly, we show in Figure 7-12, the adaptive OCWB solution of the two-dimensional Heston model, as a function of the underlying asset price and time, for the value of the deterministic asset volatility. In Figure 7-14, the same solution is shown in function of the asset price and

level	0	1	2	3	4	5
N_{ad}	12	23	31	49	85	155
ΔS	10	5	2.5	1.25	0.625	0.313
Δt	1year	3months	23days	6days	34hours	9hours
price	1.7475	1.4487	1.2600	1.1991	1.1838	1.1801

Table 7-4: Adaptive OCWB solutions of Black-Scholes.

level	0	1	2	3	4
N_{ad}	84	299	597	1841	7611
ΔS	10	5	2.5	1.25	0.625
Δt	1year	3months	23days	6days	34hours
price	2.0422	1.9021	1.7994	1.7748	1.7589

Table 7-5: Adaptive OCWB solutions of Heston.

asset volatility for time zero. The discontinuity caused by the barrier at option maturity has visibly smoothened out over time. Also for this model, we compute only a sub-set of the OCWB detail functions, shown in Figure 7-15, depending on the significance of the coarser details, given in Figure 7-13. Again, the details decay over the asset price away from the barrier, and over time. However, there is less significant decay over volatility. We could decide to manually limit refinement to a zone close to the deterministic asset volatility. Note that this would be consistent with the local mesh refinement proposed for this area, in other research (see e.g. Kluge, 2002). However, in the results discussed in this dissertation, we have consistently based all adaptive refinement on the threshold rule described above.

The adaptive solutions of the Black-Scholes model and of the Heston model, by our OCWB method, are given for different numbers J of level of refinement in Tables 7-4 and 7-5 respectively. For both the 1D and the 2D case, the adaptive problem sizes are significantly smaller than the non-adaptive problem sizes. Note that in Tables 7-4 and 7-

5, the problem size N_{ad} denotes the adaptive problem size averaged over all time-steps. This problem size is – per time-step – the sum of the sizes of all sub-problems, each associated with a level j. The number of degrees of freedom remains quasi-constant (as opposed to doubles) per level for the 1D problem, and quasi-doubles (as opposed to quadruples) per level for the 2D problem. For an increasing highest resolution J, additional details are added on all levels j to achieve a higher overall accuracy. Comparing the results of Tables 7-4 and 7-5 with the non-adaptive solutions of Tables 7-2 and 7-3 respectively, shows that – with fully (1D) or partially (2D) telescopic refinement – we achieve similar convergence. We can solve independently the J smaller detail parts of the multi-resolution system matrix, with Gaussian Elimination. Hence, as discussed in Chapter 6, such adaptive OCWB method would be faster than a Hierarchical Basis adaptive or non-adaptive FE method.

Note that since the above described method of refinement works very well, we do not need to use the information contained in the multi-resolution solution $\mathbf{u}_J^{t[0]}$ of the previous time-step, nor the multi-resolution right-hand-side of Equation (7.11), $\mathbf{M}_J^{t-1[0]}\mathbf{u}_J^{t[0]}$, to determine where to refine at time $t-1$. For each time step, we make an independent decision, level per level, as to where to refine.

As an interesting corollary, the piecewise linear test wavelets, customized to these particular operators, vanish constant and linear functions. This property is shared by Equation (7.1) as well as Equation (7.4), and it will be explicitly shown for the two-dimensional problem. Imposing operator-orthogonality of wavelets to individual scaling functions, as in Equation (7.10), leads to operator-orthogonality to any linear combination of scaling functions. Consequently, test function wavelets away from a Dirichlet boundary are operator-orthogonal to the constant and linear functions:

$$\begin{aligned} &a_{t-1}\left(1,\psi_j^{t-1}\right)=0 \\ &a_{t-1}\left(S,\psi_j^{t-1}\right)=0 \\ &a_{t-1}\left(y,\psi_j^{t-1}\right)=0 \end{aligned} \qquad (7.13)$$

Test function wavelets near a Dirichlet boundary are operator-orthogonal to linear functions that satisfy the boundary condition. Equations (7.13) yield:

$$\left(\iint_{\Omega} -r_{t-1}\psi_j^{t-1} dSdy\right)\Delta t + \iint_{\Omega} \psi_j^{t-1} dSdy = 0$$

$$\left(\iint_{\Omega} \frac{S^2 y}{2}\frac{\partial \psi_j^{t-1}}{\partial S} + \frac{\rho\xi Sy}{2}\frac{\partial \psi_j^{t-1}}{\partial y} + \left(r_{t-1}S - Sy - \frac{\rho\xi S}{2}\right)\psi_j^{t-1} - r_{t-1}S\psi_j^{t-1} dSdy\right)\Delta t + \iint_{\Omega} S\psi_j^{t-1} dSdy = 0$$

$$\left(\iint_{\Omega} \frac{\rho\xi yS}{2}\frac{\partial \psi_j^{t-1}}{\partial S} + \frac{\xi^2 y}{2}\frac{\partial \psi_j^{t-1}}{\partial y} + \left(\kappa(\theta - y) - \frac{\rho\xi y}{2} - \frac{\xi^2}{2}\right)\psi_j^{t-1} - r_{t-1}y\psi_j^{t-1} dSdy\right)\Delta t + \iint_{\Omega} y\psi_j^{t-1} dSdy = 0$$

Under the reasonable assumptions of $\Delta t < 1$ and a risk-free interest rate $r_{t-1} < 1$, we can be sure that $r_{t-1}\Delta t \neq 1$. Furthermore, we assume that $(\kappa + r_{t-1})\Delta t \neq 1$ as well. Then, using integration by parts for the first derivatives of compact wavelets:

$$\iint_{\Omega} \psi_j^{t-1} dSdy = 0$$

$$\iint_{\Omega} S\psi_j^{t-1} dSdy = 0$$

$$\iint_{\Omega} y\psi_j^{t-1} dSdy = 0$$

And therefore the test function wavelets ψ_j^{t-1} vanish all functions that are constant or linear over the wavelet's support. Note that these vanishing moments are caused by, but not a sufficient condition for operator-orthogonality. Vanishing moments in the test wavelets will reduce the right hand side of Equation (7.11) for smooth solutions $\mathbf{u}_J^t$. Indeed, if $\mathbf{u}_J^t$ were linear over the support of a test wavelet the corresponding right-hand-side row entry would be zero:

$$\iint_{\Omega} \left(\sum_i \lambda_{J,i}^t \phi_{J,i}\right)\psi_j^{t-1} dSdy = 0$$

Since our solution smoothens over time, the right-hand-side of Equation (7.11) will diminish as well.

7.4 Conclusion

We successfully applied our *Operator-Customized Wavelet Basis* (OCWB) FEM to solve a barrier option pricing problem. Under Black-Scholes' assumptions, we can price such options by solving an iteration over time of one-dimensional PDE problems. If we allow for a stochastic volatility of the underlying asset, we can price them with an iteration of two-dimensional PDEs instead. These second-order PDEs consist of a diffusion, convection and reaction term, with time-dependent varying coefficients, and may be solved with a first-order Lagrangian FEM. For both the one-dimensional as the two-dimensional problem, the barrier introduces a local concentration of the solution error, which can be exploited by an adaptive method. An OCWB allows for scale-decoupled local refinements, whereas a *Hierarchical Basis* (HB) does not have this advantage. Because the number of details added at each level does not grow substantially, the OCWB FEM yields a solution speed close to the optimal speed of $O(J)$ per time-step, where J is the number of levels of refinement. This is not the case for an HB FEM.

We will now conclude this dissertation with a brief discussion of two possible extensions to this research, in Chapter 8.

8. Research Extensions

8.1 3D Problems

An obvious extension to this research would be the customization of wavelets for three-dimensional problems. As discussed in Chapter 6, for a problem with J levels of telescopic adaptive refinement, we have a solution method that has an operation cost of only $O(J)$, irrespective of the number of spatial dimensions. For one-dimensional problems, this does not beat a single-scale direct solver (Gaussian Elimination) or a Hierarchical Basis iterative solver (Conjugate Gradient Method). For two-dimensional problems, we outperform the $O(J^3)$ cost of the HB direct solver, as well as the between $O(J)$ and $O(J^2)$ cost of the HB iterative solver. Note that for our example in Chapter 6, we found the HB iterative solver to have a cost close to the upper range $O(J^2)$. For three-dimensional problems, we expect to outperform the other methods as well, probably even stronger. Indeed, we would have again an HB direct solving cost of $O(J^3)$, and we would expect an HB iterative solving cost of at least over $O(J)$ but below $O(J3^J)$. If the latter is effectively above $O(J^2)$, there is more to be gained in three-dimensional than in two-dimensional applications, for which local (telescopic) refinement is appropriate.

We do not expect any strong theoretical challenges in extending the one-dimensional and two-dimensional approach to three-dimensional problems. Indeed, we believe it is possible to construct a multi-resolution framework for basis functions of a three-dimensional Lagrangian finite element space of any given order. After establishing a scaling relation, we may use second-generation wavelet theory to build the wavelets. We also expect to be able to benefit from the compactness inherent to the mesh geometry to find enough compactly supported wavelets that satisfy the operator-orthogonality constraint, just as in the one-dimensional and two-dimensional case. In addition, we anticipate finding more compact support for wavelets customized to the Laplace operator, than for wavelets customized to more general operators. Indeed, we believe that also the inheritance of vanishing moments property is extendable to three-dimensional problems.

8.1 Hermite Finite Elements

An important extension to the work described in this dissertation is the customization of basis functions of the Hermite finite element space to higher-order operators. The Hermite finite element basis functions are pure polynomial over an element, with C^1 continuity (basis functions and their first derivatives are continuous) from an element to a neighboring element. Each node supports two degrees of freedom: a translation and a rotation component. The basis functions for a Lagrangian finite element space – the focus of our study – are pure polynomial over an element, but with only C^0 continuity (basis functions are continuous) from one element to a neighboring element. Each node has only a translation degree of freedom. Whereas Lagrangian finite element basis functions can be used to numerically solve second-order partial differential equations, the Hermite finite element basis functions can be used for fourth-order operators, for which a higher degree of basis function continuity is required. Sudarshan Ragunathan, of the Department of Civil and Environmental Engineering at MIT, is writing a doctoral dissertation on the customization of cubic Hermite wavelets to the bi-harmonic operator on quasi-regular multi-dimensional grids. With respect to the customization, it is interesting to note that the *inheritance of vanishing moments* property for the Laplace operator with Lagrangian basis functions extends to the bi-harmonic operator with Hermite basis functions, where vanishing moments are inherited over two levels of differentiation.

Bibliography

Amaratunga K. and Williams J. R. (1993), "Wavelet Based Green's Function Approach to 2D PDEs", *Engineering Computations*, Vol.10 No.4.

Amaratunga K., Williams J. R., Qian S. and Weiss J. (1994), "Wavelet-Galerkin Solutions for One Dimensional Partial Differential Equations", *International Journal for Numerical Methods in Engineering,* Vol.37, pp.2703-2716.

Amaratunga K. and Williams J. R. (1997), "Wavelet-Galerkin Solution of Boundary Value Problems", *Archives of Computational Methods in Engineering,* Vol.4 No.3, pp.243-285.

Amaratunga K. and Castrillón-Candás J. (2001), "Surface wavelets: a multiresolution signal processing tool for 3D computational modeling", *International Journal for Numerical Methods in Engineering,* Vol.52, pp.239-271.

Amaratunga K., Sudarshan R. and D'Heedene S. (2003), "Multiresolution Finite and Boundary Element Methods Based on Wavelet and Subdivision Theory", *Proceedings of the Seventh US National Congress on Computational Mechanics*, Albuquerque, NM.

Bathe K.J. (1996), *Finite Element Procedures*, Prentice-Hall.

Betaneli D. (1998), "Wavelets and PDEs: the Improvement of Computational Performance using Multi-Resolution Analysis", Doctoral Thesis, Massachusetts Institute of Technology, Department of Mathematics.

Beylkin G., Coifman R. and Rokhlin V. (1992), "Wavelets in Numerical Analysis", in *Wavelets and Their Applications*, Ruskai M. B. ed., Jones and Bartlett, Boston.

Black F. and Scholes M. (1972), "The Valuation of Option Contracts and a Test of Market Efficiency", *Journal of Finance*, Vol.27, pp.399-417.

Carnicer J., Dahmen W. and Pena J. (1996), "Local decompositions of refinable spaces", *Applied computational and harmonic analysis*, Vol. 3, pp.125-153.

Castrillón-Candás J. and Amaratunga K. (2003), "Spatially Adapted Multiwavelets and Sparse Representation of Integral Equations on General Geometries", *SIAM Journal on Scientific Computing*, Vol. 24, No.5, pp.1530-1566.

Cox J.C., Ingersoll J.E. and Ross S.A. (1985), "A Theory of the Term Structure of Interest Rates", *Econometrica*, Vol.53, pp.385-407.

Dahlke S. and Weinreich I. (1993), "Wavelet-Galerkin Methods: An Adapted Biorthogonal Wavelet Basis", *Constr. Approx.*, Vol.9, pp.237-262.

Dahlke S. and Weinreich I. (1994), "Wavelet Bases Adapted to Pseudo-Differential Operators", *Appl. Comp. Harmon. Anal*, Vol.1, No.3, pp.267-283.

Dahmen W., Prössdorf S. and Schneider R. (1994), "Multiscale Methods for Pseudo-Differential Equations on Smooth Closed Manifolds", in *Wavelets: Theory, Algorithms, and Applications*, eds. Chui *et al.*, Academic Press, pp.385-424.

Dahmen W., Stevenson R. (1999), "Element-by-element construction of wavelets satisfying stability and moment conditions", *SIAM Journal on Scientific Computing*, Vol.37, No.1, pp.319-352.

Daubechies I. (1988), "Orthonormal bases of compactly supported wavelets", *Communications on Pure and Applied Mathematics*, Vol.41, pp.909-996.

D'Heedene S., Amaratunga K. and Castrillón-Candás J. (2005), "Generalized Hierarchical Bases: A Wavelet-Ritz-Galerkin Framework for Lagrangian FEM", to appear in *Engineering Computations.*

Golub G. and Van Loan C. (1996), *Matrix Computations (Johns Hopkins Studies in the Mathematical Sciences)*, Johns Hopkins University Press.

Haug E. (1998), *Complete Guide to Option Pricing Formulas,* McGraw Hill.

Heston S. (1993), "A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options", *The Review of Financial Studies*, Vol.6 No.2, pp.327-343.

Hull J. (1989), *Options, Futures, and Other Derivatives*, Prentice Hall.

Jawerth B. and Sweldens W. (1993), "Wavelet multiresolution analyses adapted for the fast solution of boundary value ordinary differential equations", *Sixth Copper Mountain Conference on Multigrid Methods*, NASA Conference Publication 3224, pp.259-273.

Kluge T. (2002), "Pricing Derivatives in Stochastic Volatility Models using the Finite Difference Method", Diploma Thesis, Technische Universität Chemnitz, Fakultät für Mathematik.

Krysl P., Grinspun E. and Schröder P. (2003), "Natural hierarchical refinement for finite element methods", *International Journal for Numerical Methods in Engineering*, Vol.56, Issue 8, pp.1109-1124.

Lounsbery M., DeRose T., Warren J. (1997), "Multi-resolution Analysis for Surfaces of Arbitrary Type", *ACM TOG*, No.16 (1).

Mallat S.G. (1988), "A theory for multiresolution signal decomposition: the wavelet representation", *Communications on Pure and Applied Mathematics,* Vol.41, pp.674-693.

Meyer Y. (1985), "Principe d'incertitude, bases hilbertiennes et algebres d'operateurs", *Seminaire Bourbaki,* No.662.

Roach, G. F. (1982), *Green's Functions,* Cambridge University Press, Cambridge, Great Britain.

Schröder P. and Sweldens W. (1996), *Wavelets in Computer Graphics,* SIGGRAPH 96 course notes.

Simoens J. and Vandewalle S. (2003), "A Stabilized Lifting Construction of Wavelets on Irregular Meshes on the Interval", *SIAM Journal on Scientific Computing*, Vol.24, No.4, pp.1356-1378.

Strang G. (1993), *Introduction to Linear Algebra,* Wellesley-Cambridge Press, Wellesley, MA.

Strang G. and Strela V. (1995), "Finite element multiwavelets", *Proc. Maratea NATO Conference*, Kluwer.

Strang G. and Nguyen T. (1996), *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA.

Sudarshan R., D'Heedene S. and Amaratunga K. (2003), "A Multiresolution Finite Element Method Using Second Generation Hermite Multiwavelets", *Proceedings of the Second MIT Conference on Computational Fluid and Solid Mechanics*, Paper No. 409, Cambridge, MA.

Sweldens W. (1996), "The lifting scheme: A custom-design construction of biorthogonal wavelets", *Applied and Computational Harmonic Analysis,* Vol.3, pp.186-200.

Vassilevski, P. and Wang, J. (1997), Stabilizing the hierarchical basis by approximate wavelets, I: Theory, *Numer. Linear Alg. Appl.*, Vol.4, Nr.2, pp.103-126.

Winkler G., Apel T. and Wystup U. (2002), "Valuation of Options in Heston's Stochastic Volatility Model Using Finite Element Methods", in *Foreign Exchange Risk,* Risk Publications, London.

Yserentant H. (1986), "On the multi-level splitting of finite element spaces", *Numer. Math.,* Vol.49, pp.379-412.

Yserentant H. (1992), "Hierarchical Bases", *ICIAM 91, SIAM.*

Zienkiewicz O.C. and Taylor R.L. (2000), *Finite Element Method: Volume 1, 2 and 3,* Butterworth-Heinemann.