



An adaptive multilevel wavelet collocation method for elliptic problems

Oleg V. Vasilyev^{a,*}, Nicholas K.-R. Kevlahan^b

^a *Department of Mechanical Engineering, University of Colorado, 427 UCB, Boulder, CO 80309, USA*

^b *Department of Mathematics and Statistics, McMaster University, Hamilton, ON, Canada L8S 4K1*

Received 13 April 2004; received in revised form 25 October 2004; accepted 10 December 2004

Available online 28 January 2005

Abstract

An adaptive multilevel wavelet collocation method for solving multi-dimensional elliptic problems with localized structures is described. The method is based on multi-dimensional second generation wavelets, and is an extension of the dynamically adaptive second generation wavelet collocation method for evolution problems [Int. J. Comp. Fluid Dyn. 17 (2003) 151]. Wavelet decomposition is used for grid adaptation and interpolation, while a hierarchical finite difference scheme, which takes advantage of wavelet multilevel decomposition, is used for derivative calculations. The multilevel structure of the wavelet approximation provides a natural way to obtain the solution on a near optimal grid. In order to accelerate the convergence of the solver, an iterative procedure analogous to the multigrid algorithm is developed. The overall computational complexity of the solver is $O(\mathcal{N})$, where \mathcal{N} is the number of adapted grid points. The accuracy and computational efficiency of the method are demonstrated for the solution of two- and three-dimensional elliptic test problems.

© 2005 Elsevier Inc. All rights reserved.

Keywords: Wavelets; Lifting scheme; Second generation wavelets; Partial differential equations; Elliptic problem; Adaptive grid; Numerical method; Multilevel method; Multigrid method

1. Introduction

Elliptic partial differential equations (PDE) are common in many areas of physics and engineering. There have been several promising attempts to develop adaptive methods for these equations, most notably those based on adaptive finite elements [1–4]. However, little has been proved regarding the convergence of these

* Corresponding author.

E-mail addresses: oleg.vasilyev@colorado.edu (O.V. Vasilyev), kevlahan@mcmaster.ca (N.K.-R. Kevlahan).

schemes. In contrast, substantial progress has been made recently in proving convergence of wavelet-based adaptive methods for elliptic PDEs. In particular, it has been proved that the adaptive wavelet scheme converges for a variety of elliptic PDEs, as well as for singular integral equations [5]. Furthermore, it was shown that adaptive wavelet schemes are asymptotically optimal for elliptic equations, in the sense that the rate of convergence to the exact solution with respect to the number of degrees of freedom is the same as the rate of convergence of the best N -term approximation, which is obtained by retaining the N largest wavelet coefficients of the exact solution [6,7].

In recent years, there has been a growing interest in developing wavelet-based numerical algorithms for both elliptic problems [5–9] and evolution problems [10–23]. The existing wavelet-based numerical algorithms can be roughly classified as either adaptive wavelet Galerkin methods (AWGM) [6–14] or adaptive wavelet collocation methods (AWCM) [15–23]. The major difference between these approaches is that AWGM algorithms solve problems in wavelet coefficient space and, in general, can be considered as gridless methods, while AWCM solve problems in physical space on an adaptive computational grid. Two difficulties associated with AWGM are the treatment of nonlinearities and general boundary conditions, although different possibilities of dealing with these problems have been proposed [24–29]. AWCM, on the other hand, do not have these difficulties and the treatment of nonlinearities and general boundary conditions is a relatively straightforward task. The main advantage of AWGM is they generate a sparse operator representation [24], which is the reason research efforts were initially concentrated on the development of AWGM solvers. To the best of our knowledge this paper is the first attempt to develop a AWCM based elliptic solver.

The major strength of wavelet-based methods is their ability to adapt the computational grid (basis) to the solution. In AWCM every wavelet is uniquely associated with a collocation point, and thus grid adaptation is based simply on the analysis of wavelet coefficients: i.e. at any given time the computational grid consists of points corresponding to wavelets whose coefficients are greater than a given threshold (a parameter that controls the accuracy of the solution). With this adaptation strategy a solution is obtained on a near optimal grid for a given accuracy. This means that the compressed solution is obtained directly, as opposed being the result of a posteriori compression, as is done in data analysis. We emphasize here that the adaptation of the computational grid does not require additional effort, and consists merely in turning on and off wavelets at different locations and scales. Furthermore, grid adaptation is achieved by analyzing the solution, and not by applying ad hoc assumptions, as is often the case in conventional adaptive mesh algorithms [30,31].

Traditionally, wavelet-based numerical methods make use of first generation wavelets that are constructed by discrete (typically dyadic) dilation and translation of a single mother wavelet $\psi(x)$. This results in the construction of *first generation wavelets* [32,33] that are defined either in infinite or periodic domains. It is desirable in many engineering applications to have a larger class of wavelets that can be defined in general domains and/or on irregular sampling intervals. In order to achieve this, the translation and dilation relations of the first generation wavelets must be abandoned, and wavelets are constructed in physical space rather than in Fourier space. Recently, a new class of wavelets, currently referred to as *second generation wavelets* [34,35], has come to the fore. The main advantage of second generation wavelets is that they are constructed in the spatial domain and thus can be custom designed for the complex geometry and nonuniform sampling intervals commonly found in engineering and physical applications, e.g. heat transfer in complex geometry with localized heat source.

The general framework of the second generation AWCM has been developed by Vasilyev and Bowman [22] and Vasilyev [23] in the context of evolution problems. The objective of the present work is to extend the method to the solution of multi-dimensional elliptic problems with localized structures. Two different issues are addressed in this paper. The first concerns the development of a general adaptive elliptic solver that obtains a solution on an optimal (compressed) grid. The second is the development of an efficient multilevel elliptic solver for an adaptive, but fixed, computational grid. The multilevel structure of the wavelet

approximation provides a natural framework to establish an iterative algorithm, which is similar in spirit to multigrid methods [36]. These two techniques are then combined to produce an adaptive multilevel elliptic solver. The main differences between the proposed multilevel elliptic solver and multigrid methods are the structure of the nested grids and the use of wavelet interpolation for both prolongation and restriction operators.

The paper is organized as follows. Section 2 gives a brief introduction to the second generation wavelets and wavelet compression. The AWCМ for solving elliptic PDEs is introduced in Section 3. Finally, Section 4 presents some applications of the method to the solution of the Poisson equation, along with quantitative convergence results. The main results are summarized, and future research directions are outlined in Section 5.

2. Second generation wavelets

Second generation wavelets [34,35] are a generalization of biorthogonal wavelets [32,33] which are more easily applied to functions defined on domains more general than R^n . Second generation wavelets form a Riesz basis for L_2 space. The basis is local in both space and frequency, and can have many vanishing polynomial moments. However, the translation and dilation invariance of their biorthogonal cousins is lost. Despite the loss of two fundamental properties of wavelet bases, second generation wavelets retain the most useful features of biorthogonal wavelets, including the existence of a fast transform. In this section, we briefly summarize wavelet decomposition. For details of multiresolution wavelet analysis and construction of second generation wavelets we refer the reader to [34,35].

In this work we use tensor product second generation wavelets [23,35] that are constructed on a set of nested grids

$$\mathcal{G}^j = \{\mathbf{x}_{\mathbf{k}}^j \in \Omega : \mathbf{k} \in \mathcal{K}^j\}, \quad j \in \mathcal{J}, \quad (1)$$

where j is the level of resolution, \mathcal{J} is an integer index set associated with resolution levels, \mathcal{K}^j is some index set associated with scaling functions of level j , $\mathbf{k} = (k_1, \dots, k_n)$, and the grid points $\mathbf{x}_{\mathbf{k}}^j = (x_{1,k_1}^j, \dots, x_{n,k_n}^j)$ are constructed as a tensor product of uniformly or nonuniformly spaced one-dimensional nested grids [23]. Since each individual set of one-dimensional grids is nested ($x_{m,k_1}^j = x_{m,2k_1}^{j+1}$, $m = 1, \dots, n$) the resulting set of n -dimensional grids is also nested, i.e. $\mathcal{G}^j \subset \mathcal{G}^{j+1}$. Following the construction of second generation wavelets described in [23,35], we construct n -dimensional tensor product scaling functions $\phi_{\mathbf{k}}^j(\mathbf{x})$ ($\mathbf{k} \in \mathcal{K}^j$) and wavelets of different families $\psi_{\mathbf{l}}^{\mu,j}(\mathbf{x})$ ($\mathbf{l} \in \mathcal{L}^{\mu,j}$) such that a function $u(\mathbf{x})$ can be decomposed as

$$u(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{K}^0} c_{\mathbf{k}}^0 \phi_{\mathbf{k}}^0(\mathbf{x}) + \sum_{j=0}^{+\infty} \sum_{\mu=1}^{2^n-1} \sum_{\mathbf{l} \in \mathcal{L}^{\mu,j}} d_{\mathbf{l}}^{\mu,j} \psi_{\mathbf{l}}^{\mu,j}(\mathbf{x}), \quad (2)$$

where $\mathcal{L}^{\mu,j}$ is some index set associated with wavelets of family μ and level j . One may think of a wavelet decomposition as a multilevel or multiresolution representation of a function, where each level of resolution j (except the coarsest one) consists of wavelets $\psi_{\mathbf{l}}^j$ or family of wavelets $\psi_{\mathbf{l}}^{\mu,j}$ having the same scale but located at different positions. Note that scaling function coefficients represents a smoothed version of the function at the current scale, while the wavelet coefficients represent the details of the function between the current scale and the next finest scale. Also note that the wavelet transform is a recursive application of a single level wavelet transform starting either from finest or coarsest levels of resolution. Thus, even though each single level wavelet transform is tensorial, the overall multilevel n -dimensional wavelet transform is not a tensor product of n one-dimensional transforms. In addition, in n -dimensions there are $2^n - 1$ distinct n -dimensional wavelets [37].

The major strength of wavelet decomposition (2) is the ability to compress functions. For functions that contain isolated small scales on a large-scale background (i.e. intermittent functions), most wavelet coeffi-

coefficients are small. Thus, we can retain a good approximation, even after discarding a large number of wavelets with small coefficients. Intuitively, the coefficient $d_1^{\mu,j}$ is small unless the $u(\mathbf{x})$ has variation on the scale of j in the immediate vicinity of wavelet $\psi_1^{\mu,j}(\mathbf{x})$. More precisely, if we rewrite (2) as the sum of two terms composed of wavelets whose amplitudes are respectively above and below some prescribed threshold ϵ ,

$$u(\mathbf{x}) = u_{\geq}(\mathbf{x}) + u_{<}(\mathbf{x}), \tag{3}$$

where

$$u_{\geq}(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{X}^0} c_{\mathbf{k}}^0 \phi_{\mathbf{k}}^0(\mathbf{x}) + \sum_{j=0}^{+\infty} \sum_{\mu=1}^{2^n-1} \sum_{\substack{\mathbf{l} \in \mathcal{L}^{\mu,j} \\ |d_1^{\mu,j}| \geq \epsilon}} d_1^{\mu,j} \psi_1^{\mu,j}(\mathbf{x}), \tag{4}$$

$$u_{<}(\mathbf{x}) = \sum_{j=0}^{+\infty} \sum_{\mu=1}^{2^n-1} \sum_{\substack{\mathbf{l} \in \mathcal{L}^{\mu,j} \\ |d_1^{\mu,j}| < \epsilon}} d_1^{\mu,j} \psi_1^{\mu,j}(\mathbf{x}), \tag{5}$$

then, following Donoho [38], it can be shown that for a sufficiently smooth function $u(\mathbf{x})$

$$|u(\mathbf{x}) - u_{\geq}(\mathbf{x})| \leq C_1 \epsilon. \tag{6}$$

This implies that the number of significant wavelet coefficients \mathcal{N} is bounded by ϵ as

$$\mathcal{N} \leq C_2 \epsilon^{-\frac{n}{p}}, \tag{7}$$

where p is the order of the wavelet (the number of neighboring points used for wavelet construction during the prediction phase of wavelet transform [35]), n is the dimensionality of the problem and coefficients C_i depend on $u(\mathbf{x})$ (but are of order unity). Note that p controls the number of zero moments of the interpolating scaling function. Also note that the second generation wavelet transform is characterized by another parameter \tilde{p} (the number of neighboring points, or the stencil size, used for wavelet construction during the update phase of wavelet transform [35]), which controls the number of zero moments of the wavelets.

Combining (6) and (7) we have the following bound on an error in terms of \mathcal{N}

$$|u(\mathbf{x}) - u_{\geq}(\mathbf{x})| \leq C_3 \mathcal{N}^{-p/n}. \tag{8}$$

This error estimate is consistent with numerical experiments for both one-dimensional [22] and two-dimensional cases [23].

3. Numerical method

A linear elliptic PDE may be written in the general form,

$$\mathcal{L}\mathbf{u} = \mathbf{f}, \tag{9}$$

where \mathcal{L} is a linear elliptic operator (including boundary conditions), and \mathbf{f} is a source term. In the following, we describe an efficient multilevel AWCM for determining \mathbf{u} to within a specified residual tolerance $\|\mathcal{L}\mathbf{u} - \mathbf{f}\|_p < \epsilon$ given \mathcal{L} and \mathbf{f} . One notable feature of the method is that the minimum grid resolution is determined automatically, given the tolerance ϵ .

The numerical method is formally derived by evaluating the governing PDEs at collocation points. In order for the algorithm to resolve all the structures appearing in the solution, and yet be efficient in terms of minimizing the number of unknowns, the computational grid should adapt to reflect local structure of

the solution, i.e. high resolution computations should be carried out only in those regions where sharp transitions occur.

3.1. Grid adaptation

Grid adaptation occurs naturally in wavelet methods, e.g. [10,15]. To illustrate the algorithm, let us consider a function $u(\mathbf{x})$, defined on a closed n -dimensional rectangular domain Ω . Relations (6) and (8) give us the framework for representing a function with significantly fewer degrees of freedom, while still retaining a good approximation. However, in order to realize all the benefits of the wavelet compression, we need to be able to reconstruct $u_{\geq}(\mathbf{x})$ from the subset of \mathcal{N} significant grid points. We recall that every scaling function $\phi_{\mathbf{k}}^j(\mathbf{x})$, $\mathbf{k} \in \mathcal{K}^j$, is uniquely associated with a grid point $\mathbf{x}_{\mathbf{k}}^j$, while each wavelet $\psi_{\mathbf{l}}^{\mu,j}(\mathbf{x})$, $\mathbf{l} \in \mathcal{L}^{\mu,j}$ is uniquely associated with a corresponding collocation point, e.g. two-dimensional wavelets $\psi_{(l_1,l_2)}^{1,j}(\mathbf{x})$, $\psi_{(l_1,l_2)}^{2,j}(\mathbf{x})$, and $\psi_{(l_1,l_2)}^{3,j}(\mathbf{x})$ are respectively associated with $(x_{1,2l_1+1}^{j+1}, x_{2,l_2}^j)$, $(x_{1,l_1}^j, x_{2,2l_2+1}^{j+1})$, and $(x_{1,2l_1+1}^{j+1}, x_{2,2l_2+1}^{j+1})$ grid points. So once the wavelet decomposition is performed, each grid point is uniquely associated either with the wavelet or the scaling function at the coarsest level of resolution. Consequently, the collocation point should be omitted from the computational grid if the associated wavelet is omitted from the approximation. Note that for the stability of a reconstruction algorithm we need to keep all the grid points associated with the scaling function at the coarsest level of resolution. This procedure results in a set of nested adaptive computational grids $\mathcal{G}_{\geq}^j \subset \mathcal{G}^j$, such that $\mathcal{G}_{\geq}^j \subset \mathcal{G}_{\geq}^{j+1}$ for any $j < J - 1$, where J is the finest level of resolution present in the approximation (4).

It should be noted that an additional procedure, called the *perfect reconstruction check*, is necessary [23]. In order to illustrate the perfect reconstruction check procedure, let us consider the one step forward one-dimensional second generation wavelet transform

$$d_k^j = \frac{1}{2} \left(c_{2k+1}^{j+1} - \sum_l w_{k,l}^j c_{2k+2l}^{j+1} \right), \quad (10)$$

$$c_k^j = c_{2k}^{j+1} + \sum_l \tilde{w}_{k,l}^j d_{k+l}^j, \quad (11)$$

where $w_{k,l}^j$ and $\tilde{w}_{k,l}^j$ are coefficients associated with two stages of wavelet transform. In order to find the wavelet coefficient d_l^j we need to know only values of c_k^{j+1} at the grid point associated with the wavelet $\psi_l^j(x)$, i.e. x_{2l+1}^{j+1} , and the p nearest even grid points x_{2l+2n}^{j+1} . However, in order to calculate c_k^j we only need to know the value c_{2k}^{j+1} and the values of d_l^j that are above the threshold value, ϵ . Wavelet coefficients below threshold value are set to zero. In the higher dimensional case the situation is analogous. The only difference is that the n -dimensional wavelet transform consists of the sequential application of n one-dimensional wavelet transforms in x_i , $i = 1, \dots, n$, directions. Thus in order to find the grid points that are necessary for the calculation of the wavelet coefficient $d_1^{\mu,j}$, we start with the collocation point associated with $d_1^{\mu,j}$ and recursively, $i = n, \dots, 1$, add points that are needed to perform one step of the one-dimensional wavelet transform in the x_i direction at the locations that are added to perform the one-dimensional wavelet transforms in x_l , $l = i + 1, \dots, n$, directions. At the end of this recursive procedure we will have a *minimal set* of grid points that are necessary for calculation of wavelet coefficient $d_1^{\mu,j}$, provided that wavelet coefficients at other locations are either zero or negligible (below an a priori prescribed threshold). Fig. 1 illustrates the *minimal set* of grid points that are necessary for calculation of wavelet coefficient $d_1^{\mu,j}$ belonging to three different families of wavelets, i.e. $\mu = 1, 3$. Thus, if we a priori know what wavelet coefficients are zero or negligible, we can disregard the values of the function at these points. The pseudocode for the perfect reconstruction check procedure is shown in Algorithm 1. At the end of this procedure we have the complete mask

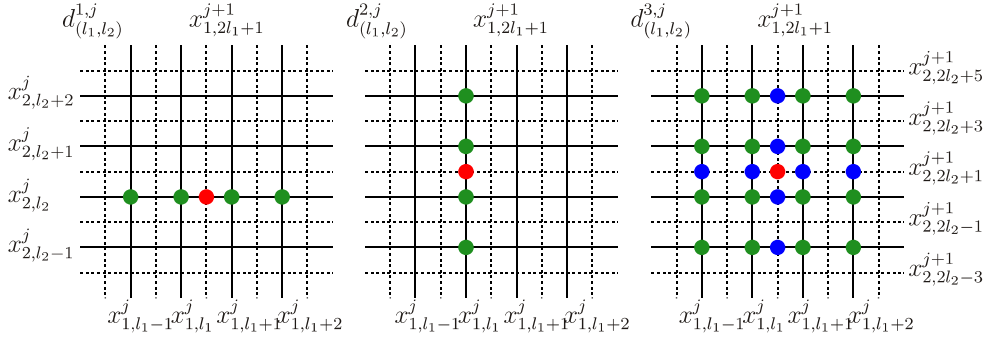


Fig. 1. Points at the coarser level j (marked \bullet) and finer level $j + 1$ (marked \bullet) where c_k^{j+1} are needed for calculation of the wavelet coefficient $d_1^{\mu,j}$, $\mu = 1, 3$ (marked \bullet) for two-dimensional wavelet transform with $p = 4$.

\mathcal{M} , from which we can easily construct a set of nested adaptive computational grids \mathcal{G}_\geq^j . The perfect reconstruction check procedure guarantees that all wavelet coefficients obtained by performing the wavelet transform on the adapted grid are the same as those found by performing the wavelet transform of $u_\geq(\mathbf{x})$ on the complete grid.

3.2. Global elliptic solver

The strategy for constructing the adaptive computational grid, or equivalently the mask \mathcal{M} , which is discussed in previous section assumes knowledge of the solution on the finest level and results in a compressed solution. However, one would like to obtain a compressed solution without the additional overhead of finding the solution everywhere on a non-adaptive grid. This overhead could be substantial if a solution is highly localized. In order to solve elliptic problems with highly localized solutions in an efficient manner, an iterative procedure of grid refinement should be used. One way to organize such a procedure is to start the calculations on a coarse grid. Once the solution is obtained, the computational grid must be extended to include grid points associated with wavelets whose coefficients are, or can possibly become, significant during the next iteration. In other words, as suggested by Liandrato and Tchamitchian [10], the computational grid should include not only points associated with wavelets whose coefficients are greater than the threshold ϵ , but also those points associated with wavelets belonging to an *adjacent zone*. We say that the wavelet $\psi_1^{\mu,j}(\mathbf{x})$ located at x_k^{j+1} belongs to the adjacent zone of wavelet $\psi_1^{\mu,j}(\mathbf{x})$ located at x_k^{j+1} if the following relations are satisfied:

Algorithm 1. [Reconstruction check procedure for the wavelet transform]

```

sample a function  $u(\mathbf{x})$  on a grid  $\mathcal{G}^J$  or  $\mathcal{G}_\geq$ 
perform forward wavelet transform
for all levels  $j = 1:1:J$ 
    create a mask  $\mathcal{M}$  for  $|d_1^{\mu,j}| \geq \epsilon$ 
end
include into mask  $\mathcal{M}$  all points at level  $j = 1$ 
for all levels  $j = J - 1:-1:1$ 
    extend mask  $\mathcal{M}$  to include the minimal set of points for  $|d_1^{\mu,j}| \geq \epsilon$ 
end
    
```

Algorithm 2. [Global elliptic solver]

```

initial guess ( $m = 0$ ) :  $\mathbf{u}_k^m$  and  $\mathcal{G}_{\geq}^m$ 
while  $m = 0$  or  $m > 1$  and [ $\mathcal{G}_{\geq}^m \neq \mathcal{G}_{\geq}^{m-1}$  or  $\|\mathbf{u}_k^m - \mathbf{u}_k^{m-1}\|_{\infty} > \delta_{\epsilon}$ ]
   $m = m + 1$ 
  perform forward wavelet transform for each component of  $\mathbf{u}_k^m$ 
  for all levels  $j = J:-1:1$ 
    create a mask  $\mathcal{M}$  for  $|d_1^{m,j}| \geq \epsilon$ 
  end
  extend the mask  $\mathcal{M}$  with adjacent wavelets (Eq. (12))
  perform the reconstruction check procedure (Algorithm 1)
  construct  $\mathcal{G}_{\geq}^{m+1}$ 
  if  $\mathcal{G}_{\geq}^{m+1} \neq \mathcal{G}_{\geq}^m$ 
    interpolate  $\mathbf{u}_k^m$  to  $\mathcal{G}_{\geq}^{m+1}$ 
  end if
  solve Eq. (9) using local multilevel elliptic solver (Algorithm 4)
end

```

$$|j - j'| \leq L, \quad |2^{j-j'} k_m - k'_m| \leq M, \quad m = 1, \dots, n, \quad (12)$$

where L determines the extent of which coarser and finer scales are included into the adjacent zone and M defines the width of the adjacent zone in physical space. The values of L and M affect the total number of collocation points present in the grid \mathcal{G}_{\geq} . For efficiency we should keep the number of collocation points in the adjacent zone as small as possible. We have found that the optimal values are $L = M = 1$. In other words, the adjacent zone includes only the nearest neighbors at the same, one above, and one below the resolution level associated with the current grid point. This grid refinement procedure may be continued iteratively until both the solution and grid converge.

The adaptive grid refinement procedure provides a way to obtain the solution on an optimal (compressed) grid. This strategy has been already tested for obtaining the compressed form of the initial conditions for evolution problems in different physical settings, e.g. [21–23,39–43]. What is different in this work is that the same procedure is applied in the context of elliptic PDEs. In this case the PDE must be solved during each grid iteration. For this purpose the elliptic multilevel wavelet collocation solver (see 3.2) is used. The pseudocode for the iterative global elliptic solver is shown in Algorithm 2. Note that \mathbf{u}_k^m denotes the vector function \mathbf{u} evaluated at the grid points $\mathbf{x}_k^j \in \mathcal{G}_{\geq}^m$ during m th iteration.

With such an algorithm the grid of collocation points is continuously refined to resolve the local structures that appear in the solution. Note that by omitting wavelets with coefficients below a threshold parameter ϵ we automatically control the error of approximation. Thus, the wavelet collocation method has another important feature: active control of the accuracy of the solution. The smaller ϵ is chosen to be, the smaller the error of the solution is. In typical applications the value of ϵ varies between 10^{-3} and 10^{-6} , assuming that the unknown dependent variables have been properly normalized. If the variables are not normalized, the threshold parameter for each variable is scaled by either the maximum value of the variable or an a priori prescribed scale. Note also that the smallest scale is also controlled by the parameter ϵ , since new small scales are added automatically as needed via the adjacent zone.

The algorithm can use different criteria for adaptation of the collocation grid. For example, one can construct a computational grid based on the analysis of wavelet coefficients of both the function and its

derivatives. If a system of equations is solved, the adaptation of the computational grid \mathcal{G}_{\geq}^m should be based on the analysis of wavelet coefficients associated with all dependent variables. The adaptive grid \mathcal{G}_{\geq}^m can be constructed as the union of irregular grids corresponding to each dependent variable. Note that the algorithm can be easily extended to the case where each variable is treated on a separate computational grid. The mapping from one grid to another can be achieved via wavelet interpolation. This may be important for problems where scales associated with different variables are considerably different.

3.3. Calculation of spatial derivatives on an adaptive grid

When solving PDEs numerically, it is important to obtain derivatives of a function from its values at collocation points. The procedure of finding derivatives, which takes advantage of the multiresolution wavelet decomposition, fast wavelet transform, and finite difference differentiation is discussed in detail by Vasilyev and Bowman [22] for the one-dimensional case and by Vasilyev [23] for multiple dimensions. In this paper, we briefly outline this procedure.

The differentiation procedure is based on the interpolating properties of second generation wavelets. We recall that wavelet coefficients $d_1^{\mu,j}$ measure the difference between the approximation of the function at the $j + 1$ level of resolution and its representation at the j level of resolution. Thus, if there are no points in the immediate vicinity of a grid point \mathbf{x}_k^j , i.e. $|d_m^{\mu,j}| < \epsilon$ for all the neighboring points, and points $\mathbf{x}_{(2k_1 \pm 1, 2k_2 \pm 1)}^{j+1}$ are not present in \mathcal{G}_{\geq}^{j+1} , then there exists some neighborhood of \mathbf{x}_k^j , Ω_k^j , where the actual function is well approximated by a wavelet interpolant based on c_m^j ($\mathbf{m} \in \mathcal{H}^j$), i.e.

$$\left| u(\mathbf{x}) - \sum_{\mathbf{m} \in \mathcal{H}^j} c_m^j \phi_m^j(\mathbf{x}) \right| \leq C_4 \epsilon, \quad \mathbf{x} \in \Omega_k^j. \tag{13}$$

Thus, differentiating this interpolant will give us the value of the derivative of the function at that particular location. Rewriting this interpolant as local Lagrange polynomial of order p , i.e. the same order as the wavelet, differentiating the polynomial, and evaluating it at \mathbf{x}_k^j location would result in local finite difference operator that uses the neighboring points of the interpolant on level j . Let us denote by \mathcal{D}_{\geq}^j a collection of such points at each level of resolution. The pseudocode for the procedure for finding derivatives at all grid points is given in Algorithm 3. At the end of this procedure we will have derivatives of the function at all grid points. The computational cost of calculating spatial derivatives will be roughly the same as the cost of forward and inverse wavelet transforms.

The accuracy of this differentiation procedure was examined by Vasilyev and Bowman [22] for the one-dimensional case and by Vasilyev [23] in multiple dimensions. It was shown that the error bound on the derivative is given by

$$|D_{x_i} u(\mathbf{x}) - D_{x_i} u_{\geq}(\mathbf{x})| \leq C_5 \mathcal{N}^{-(p-1)/n}, \tag{14}$$

where D_{x_i} stands for the derivative operator in the x_i direction. This relation was verified numerically for both one-dimensional [22] and two-dimensional cases [23]. Note that the error bound (14) is also correct for the second-order derivative if a symmetric stencil is used.

Algorithm 3. [Calculation of derivatives on the adapted grid]

```

perform forward wavelet transform for each component of  $\mathbf{u}_k^n$ 
for all levels  $j = 1:J - 1$ 
    perform one step of inverse wavelet transform for level  $j$ 
    find derivatives at grid points that belong to  $\mathcal{D}_{\geq}^j$ 
end
    
```

3.4. Multilevel iterative wavelet collocation elliptic solver

The algorithm presented in this section can be used as part of the elliptic solver to obtain the solution of an elliptic equation during iterative grid refinement as discussed in Section 3.1 or as a part of a differential constraint for the evolution problems, such as the continuity equation in the incompressible Navier–Stokes equations. In the latter case the Poisson equation for the pressure is solved in order to enforce the incompressibility condition.

The multilevel structure of the wavelet approximation gives us a natural framework to establish a V-cycle on an adaptive computational grid \mathcal{G}_{\geq} . We recall that the adaptive computational grid $\mathcal{G}_{\geq} = \mathcal{G}_{\geq}^J$ is constructed as a set of nested adaptive computational grids $\mathcal{G}_{\geq}^j \subset \mathcal{G}_{\geq}$, such that $\mathcal{G}_{\geq}^j \subset \mathcal{G}_{\geq}^{j+1}$ for any $j < J - 1$, where J is the finest level of resolution. This nested grid structure provides a framework that allows the approximation from coarser levels of resolution to be used to improve the approximation at the finest level.

The pseudocode for the full local elliptic solver (without grid adaptation) is given in Algorithm 4, where j_{\min} and J are respectively the lowest and the highest levels of resolution, v_1 and v_2 are respectively the number of pre- and post-relaxations, v_3 is the number of iterations of exact solver, ω_0 and ω_1 are the dumping parameters, and I_w^{j-1} and I_w^j are respectively the restriction and interpolation wavelet-based operators. The multilevel iterative Algorithm 4 is similar in spirit to multi-grid methods [36]. The main similarity lies in the fact that both multi-grid methods and multilevel solver utilize recursive *coarse grid correction* procedure. Both algorithms make use of low order *approximate* iterative solvers (smoothers) to relax on the error of the residual equation. Finally, both algorithms use *exact* (high order) solvers on the finest mesh to either correct approximate solution or to calculate the residual to be used by approximate solver.

The main differences between the proposed multilevel method and other multi-grid algorithms are in the details of the implementation. First, the structure of the nested grids is very different. In particular, in contrast to multi-grid methods, the lower level grid is not necessarily coarser at every region of the domain, and, thus, in some regions the grid could be identical for both finer and coarser levels. Secondly, lower-order wavelet differentiation is used for the approximate solver (smoother). Thirdly, wavelet interpolation and projection are used respectively for prolongation (injection) and restriction operators.

Algorithm 4. [Local multilevel elliptic solver using V-cycles]

```

while  $\|\mathbf{f}^J - \mathcal{L}\mathbf{u}_{\geq}^J\|_{\infty} > \delta_{\epsilon}$ 
   $\mathbf{r}^J = \mathbf{f}^J - \mathcal{L}\mathbf{u}_{\geq}^J$ 
  for all levels  $j = J:-1:j_{\min} + 1$ 
    do  $v_1$  steps of approximate solver for  $\mathcal{L}\mathbf{v}^j = \mathbf{r}^j$ 
       $\mathbf{r}^{j-1} = I_w^{j-1}(\mathbf{r}^j - \mathcal{L}\mathbf{v}^j)$ 
    enddo
  end
  Solve for  $j = j_{\min}$  level:  $\mathcal{L}\mathbf{v}^j = \mathbf{r}^j$ 
  for all levels  $j = j_{\min} + 1:+1:J$ 
     $\mathbf{v}^j = \mathbf{v}^j + \omega_0 I_w^j \mathbf{v}^{j-1}$ 
    do  $v_2$  steps of approximate solver for  $\mathcal{L}\mathbf{v}^j = \mathbf{r}^j$  enddo
  end
   $\mathbf{u}_{\geq}^J = \mathbf{u}_{\geq}^J + \omega_1 \mathbf{v}^J$ 
  do  $v_3$  steps of exact solver for  $\mathcal{L}\mathbf{u}^J = \mathbf{f}^J$  enddo
end

```

In the numerical results presented in this paper the same damping parameters have been used after each cycle. The weighted Jacobi second-order iterative solver was used as an approximate solver. The weight for the weighted Jacobi method was set to the theoretically predicted value of 2/3 for optimal convergence. Numerical experiments confirmed this weight to be the optimal for adaptive calculations as well. Finally, as in the usual multi-grid methods, either BI-CGSTAB [44] or GMRES [45] is used as an exact solver.

4. Results and discussion

In order to illustrate the accuracy and efficiency of the proposed numerical method, we will apply it to the solution of some two- and three-dimensional test problems. Both of these test problems are linear elliptic equations. The current method can be extended easily to include nonlinear equations by implementing the full approximation storage (FAS) algorithm [46]. FAS is the standard way of solving nonlinear elliptic PDEs using a multilevel or multigrid scheme. The process of grid adaptation would be roughly the same, except the adjacent zone definition may need to be modified to reflect the nature of the nonlinearity.

4.1. Problem formulation

4.1.1. Two-dimensional elliptic problem

For the first test problem we consider the two-dimensional Poisson equation

$$\Delta u = f, \quad (15)$$

where the operator Δ is the Laplacian operator

$$\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} \quad (16)$$

and f is the localized source chosen such a way that the solution of the Poisson equation is given by

$$u(x_1, x_2) = 1 + \exp\left(-\frac{(x'_1)^2 + (x'_2)^2}{2\mu_1}\right) + \exp\left(-\frac{(x''_1)^2}{2\mu_2} - \frac{(x''_2)^2}{2\mu_3}\right), \quad (17)$$

where $\mathbf{x}' = \mathbf{x} - \mathbf{x}_0$, $\mathbf{x}'' = \mathbf{R}(\mathbf{x} - \mathbf{x}_1)$, \mathbf{x}_0 and \mathbf{x}_1 are constants, and \mathbf{R} is a rotation matrix

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix},$$

defined by the angle θ . The initial and Dirichlet boundary conditions are obtained from the analytical solution (17). The problem is solved for parameter values $\mathbf{x}_0 = (0.2, 0.1)$, $\mathbf{x}_1 = (-0.25, -0.25)$, $\theta = 60^\circ$, $\mu_1 = 10^{-2}$, $\mu_2 = 10^{-3}$, $\mu_3 = 10^{-1}$.

4.1.2. Three-dimensional elliptic problem

As a second test problem we consider the three-dimensional Poisson equation (15) with the three-dimensional Laplacian operator

$$\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2}. \quad (18)$$

The localized source f is chosen such a way that the solution of the Poisson equation is given by

$$u(x_1, x_2) = 1 + \exp\left(-\frac{(x'_1)^2 + (x'_2)^2 + (x'_3)^2}{2\mu_1}\right) + \exp\left(-\frac{(x''_1)^2}{2\mu_2} - \frac{(x''_2)^2}{2\mu_3} - \frac{(x''_3)^2}{2\mu_4}\right), \tag{19}$$

where $\mathbf{x}' = \mathbf{x} - \mathbf{x}_0$, $\mathbf{x}'' = \mathbf{R}(\mathbf{x} - \mathbf{x}_1)$, \mathbf{x}_0 and \mathbf{x}_1 are constants, and \mathbf{R} is a rotation matrix

$$\mathbf{R} = \begin{bmatrix} \cos(\zeta) \cos(\eta) \cos(\xi) - \sin(\zeta) \sin(\xi) & \cos(\zeta) \cos(\eta) \sin(\xi) + \sin(\zeta) \sin(\xi) & -\cos(\zeta) \sin(\eta) \\ -\sin(\zeta) \cos(\eta) \cos(\xi) - \cos(\zeta) \sin(\xi) & -\sin(\zeta) \cos(\eta) \sin(\xi) + \cos(\zeta) \cos(\xi) & \sin(\zeta) \sin(\eta) \\ \sin(\eta) \cos(\xi) & \sin(\eta) \sin(\xi) & \cos(\eta) \end{bmatrix}$$

defined by the angles ξ , η , and ζ . The initial and Dirichlet boundary conditions are obtained from the analytical solution (19). The problem is solved for parameters values $\mathbf{x}_0 = (-0.5, -0.5, -0.5)$, $\mathbf{x}_1 = (0.2, 0.15, 0.1)$, $\xi = 45^\circ$, $\eta = -45^\circ$, and $\zeta = 30^\circ$, $\mu_1 = 5 \times 10^{-2}$, $\mu_2 = 5 \times 10^{-2}$, $\mu_3 = 2 \times 10^{-2}$, $\mu_4 = 5 \times 10^{-1}$.

4.2. Numerical results

4.2.1. Convergence of the global elliptic solver

We start by considering the adaptive grid refinement strategy discussed in Algorithm 2. The progressive adaptation of the computational grid \mathcal{G}^m for the first test problem is shown in Fig. 2 for wavelets of order $p = \tilde{p} = 6$ and wavelet threshold $\epsilon = 10^{-5}$. Comparing the grid structure with the converged solution shown in Fig. 3, it can be easily seen that the high resolution region simply follows the peaks in the solution, thus permitting proper resolution of the localized structures. Note that the computational grid for the cylindrical structure does not change between 5th and 6th iterations, indicating proper resolution of the structure for a given tolerance specified by the threshold parameter ϵ .

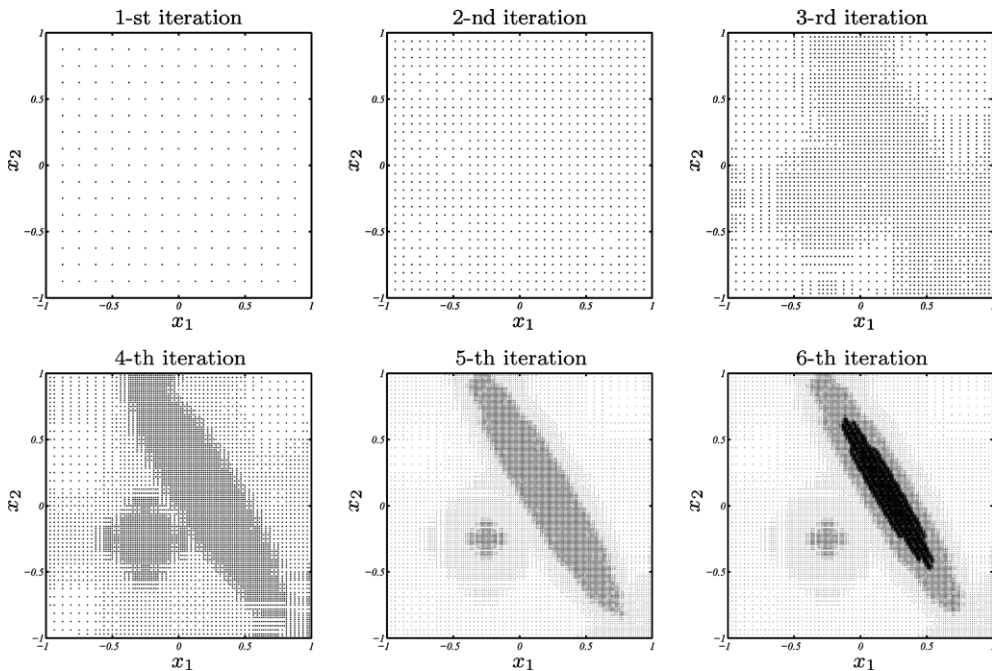


Fig. 2. Adaptive computational grids \mathcal{G}^m ($m = 1, \dots, 6$) for successive iterations ($\epsilon = 10^{-5}$, $p = \tilde{p} = 6$).

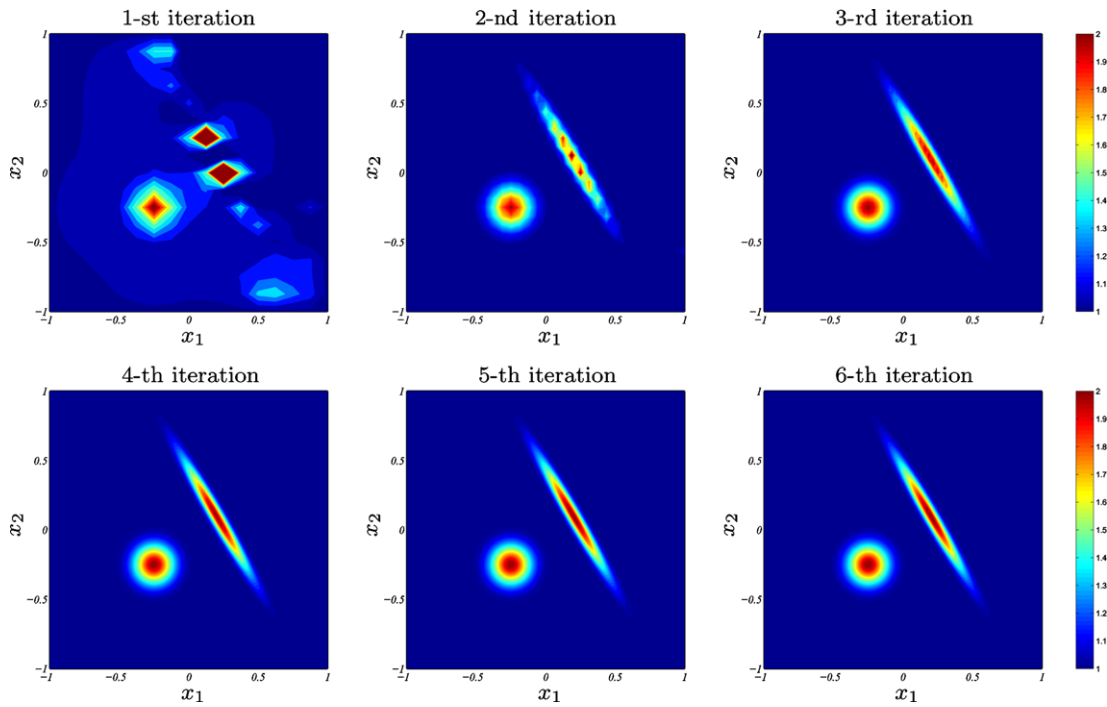


Fig. 3. Solution of elliptic problem (15) corresponding to adaptive computational grids shown in Fig. 2 ($\epsilon = 10^{-5}$, $p = \tilde{p} = 6$).

Figs. 2 and 3 provide a qualitative understanding of the global grid adaptation. In order to provide quantitative information, one needs to perform the convergence study for the algorithm. The convergence study for the adaptive wavelet algorithms with $\epsilon \neq 0$ should be distinguished from the refinement study. The latter is done by setting ϵ to zero and progressively refining the computational grid, i.e. increasing the maximum allowable level of resolution J . On the other hand, in the convergence study the maximum allowable level of resolution is not fixed and can be as high as needed. The convergence study is performed by progressively decreasing the threshold parameter ϵ and obtaining the globally converged solution for the specified ϵ . The decrease of ϵ will result in an increase in the number of grid points and number of levels of resolution. If the numerical method is convergent, then the computational error of the solution should be proportional to ϵ . Furthermore, the number of grid points should scale as predicted by (8), while the accuracy of the solution should scale according to (14). In order to increase computational savings, the threshold parameter ϵ was progressively decreased from 10^{-1} to the specified level. We found that the best strategy is simply to decrease the threshold by a factor of 10 each iteration until the desired value of ϵ is achieved. After that, the threshold is kept unchanged until both solution and computational grids are converged as described in Algorithm 2.

The results of the convergence study for the two-dimensional elliptic problem are presented in Fig. 4(a), where the pointwise l_∞ -error of the solution as a function of grid points is shown for different order wavelets. It is clearly seen that the convergence is consistent with analytical predictions of Eq. (14), shown by triangles. The dependence of the number of grid points \mathcal{N} on the threshold parameter ϵ is shown in Fig. 4(b). Once again, the dependence of \mathcal{N} on ϵ is consistent with theoretical estimate given in (8), as shown by the triangles. These figures clearly demonstrate the convergence of the numerical method with the decrease of ϵ . Note that the actual error of the solution is typically larger than ϵ , but is of the same order. Thus, by prescribing the value of ϵ , we can actively control the accuracy of the solution.

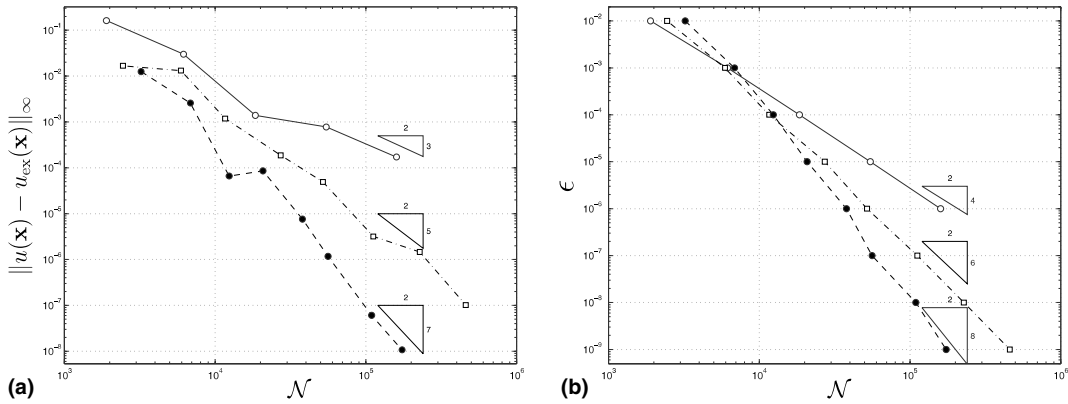


Fig. 4. The pointwise L_∞ -error of the solution of elliptic problem (15) (a) and the number of grid points as a function of ϵ (b) for different choices of parameters p , and \tilde{p} : $p = \tilde{p} = 4$ (○); $p = \tilde{p} = 6$ (□); $p = \tilde{p} = 8$ (●). \mathcal{N} is the actual number of grid points used in the calculations. The triangles indicate the slopes respectively predicted by Eqs. (14) and (8).

The results presented above clearly demonstrate that the solution converges with the decrease of ϵ . However, they do not show how fast the method converges. The efficiency of the algorithm is demonstrated in Fig. 5(a), which shows the rapid (an order of magnitude per iteration) convergence of the solution until ϵ stops decreasing as shown in Fig. 5(b). At that point the level of resolution continues to increase with iterations until all the structures are resolved and solution is converged. Note that the small dip in the error is misleading due to the fact that the solution is not fully resolved, and the error is calculated only at the points on the adaptive grid. Interpolation of the solution to the finer grid results in increase of the error, as clearly seen in Fig. 5(a). The progressive increase of the number of grid points \mathcal{N} and the maximum level of resolution J with iterations is respectively shown in Fig. 6(a) and (b).

Finally, in order to demonstrate the tremendous savings of the adaptive algorithm it is illustrative to compare the number of grid points used in the adaptive and nonadaptive methods with adequate resolution. This can be easily measured by the compression coefficient $\mathcal{C} = 1 - (\mathcal{N}/\mathcal{N}^J)$, where \mathcal{N} is the actual number of grid points used in the calculations and \mathcal{N}^J is the number of collocation points, required for the

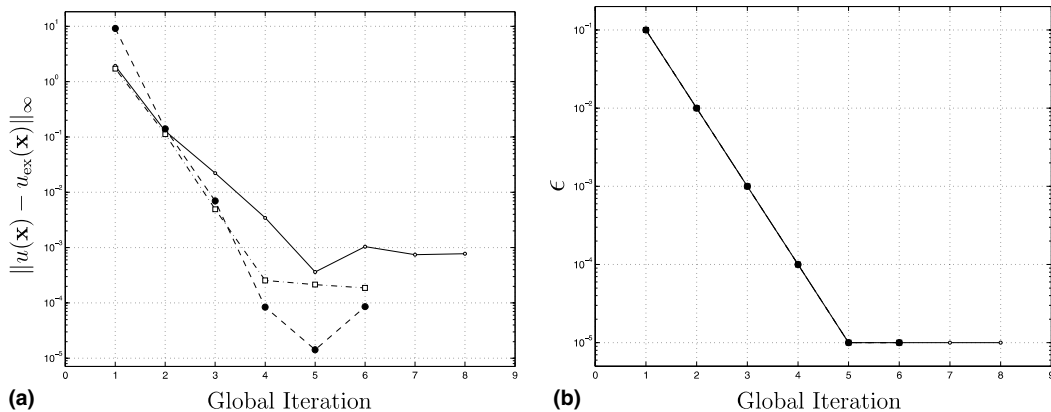


Fig. 5. The pointwise L_∞ -error of the solution of elliptic problem (15) (a) and the value of the threshold parameter ϵ (b) as a function of global iterations for $\epsilon = 10^{-5}$ and different choices of parameters p , and \tilde{p} : $p = \tilde{p} = 4$ (○); $p = \tilde{p} = 6$ (□); $p = \tilde{p} = 8$ (●).

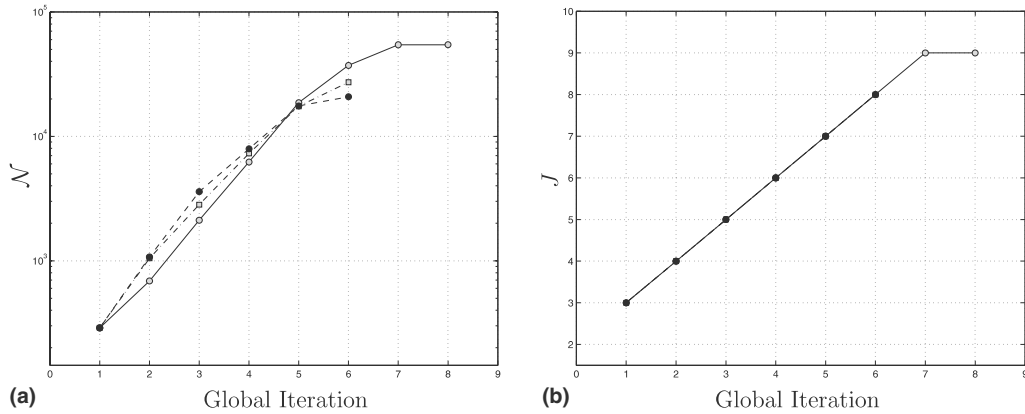


Fig. 6. The total number of grid points \mathcal{N} (a) and the maximum level of resolution J (b) as a function of global iterations for $\epsilon = 10^{-5}$ and different choices of parameters p , and \tilde{p} : $p = \tilde{p} = 4$ (\circ); $p = \tilde{p} = 6$ (\bullet); $p = \tilde{p} = 8$ (\square).

non-adaptive algorithm to solve the same problem with the comparable resolution. In other words, the compression coefficient measures the percentage of grid points that are not included to the adaptive grid \mathcal{G}'_{\geq} . The larger the compression coefficient, the more efficient the adaptive algorithm. A compression coefficient 0% indicates that there is no adaptation. As seen in Fig. 7(a) the compression coefficient monotonically increases with each iteration until the grid converges. Another way to look at the compression is to use the compression ratio $\mathbf{CR} = N^J / \mathcal{N}$, which measures the ratio of the total number of collocation points, required for the non-adaptive calculation, to the actual number of grid points used in the adaptive calculation (see Fig. 7). Note that when both solution and grid are converged, the number of grid points on adaptive grid is approximately twenty times less than on non-adaptive grid, i.e. $\mathbf{CR} \approx 20$. The compression ratio can be even higher for more localized structures. Finally, it should be noted that the execution time of the algorithm in its current implementation scales linearly with the number of grid points. Thus, the compression ratio is a direct measure of the algorithm efficiency.

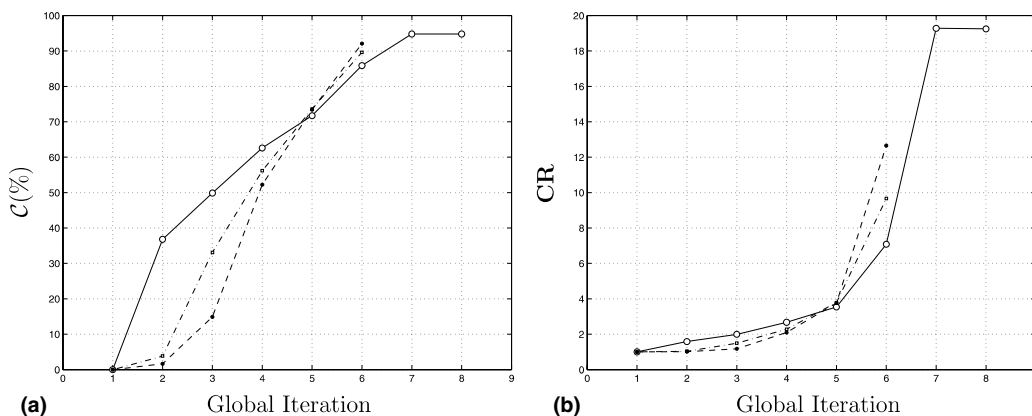


Fig. 7. The compression coefficient \mathcal{C} (a) and the compression ratio \mathbf{CR} (b) as a function of global iterations for $\epsilon = 10^{-5}$ and different choices of parameters p , and \tilde{p} : $p = \tilde{p} = 4$ (\circ); $p = \tilde{p} = 6$ (\bullet); $p = \tilde{p} = 8$ (\square).

4.2.2. *Convergence of the local multilevel elliptic solver*

The global convergence study presented in previous section only provides information about global convergence of the solution. Another factor that determines the overall efficiency of the method is the rate of the convergence of the local iterative multilevel elliptic solver discussed in Section 3.4. Due to similarities between proposed multilevel elliptic solver and multigrid methods, it is natural to expect linear convergence, i.e. the residual should decrease approximately by the same factor with each iteration. To demonstrate the convergence of the multilevel elliptic solver Figs. 8–10 show the l_∞ norm of the residual error as a function of local iterations (V-cycles) for different choices of the iteration parameters. In particular, Fig. 8 demonstrates the influence of the number of pre- (v_1) and post-relaxations (v_2) on the convergence of the multilevel solver. Obviously, two relaxations is sufficient to smooth the solution. Increasing the number of relaxations slightly decrease the error, but does not affect the rate of the convergence. The

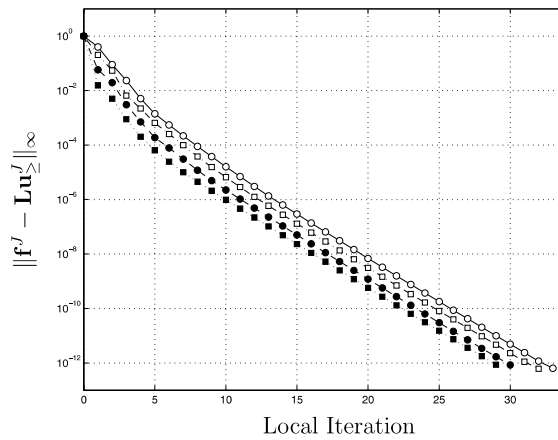


Fig. 8. The L_∞ -norm of the residual for the multilevel wavelet collocation solver as a function of local iterations (V-cycles) for $\epsilon = 10^{-5}$, $p = \bar{p} = 6$, $\omega_0 = 2/3$, $\omega_1 = 1$ and different choices of iterative parameters v_1 , v_2 , and v_3 : $v_1 = v_2 = 2$, $v_3 = 0$ (○); $v_1 = v_2 = 2$, $v_3 = 0$ (□); $v_1 = v_2 = 10$, $v_3 = 0$ (●); $v_1 = v_2 = 20$, $v_3 = 0$ (■).

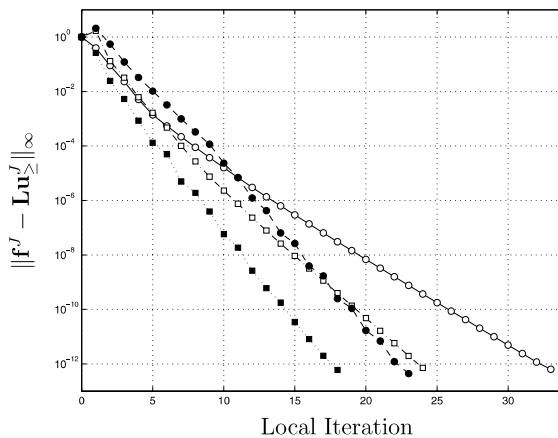


Fig. 9. The L_∞ -norm of the residual for the multilevel wavelet collocation solver as a function of local iterations (V-cycles) for $\epsilon = 10^{-5}$, $p = \bar{p} = 6$, $\omega_0 = 2/3$, $\omega_1 = 1$ and different choices of iterative parameters v_1 , v_2 , and v_3 : $v_1 = v_2 = 2$, $v_3 = 0$ (○); $v_1 = v_2 = 2$, $v_3 = 2$ (□); $v_1 = v_2 = 2$, $v_3 = 4$ (●); $v_1 = v_2 = 2$, $v_3 = 10$ (■).

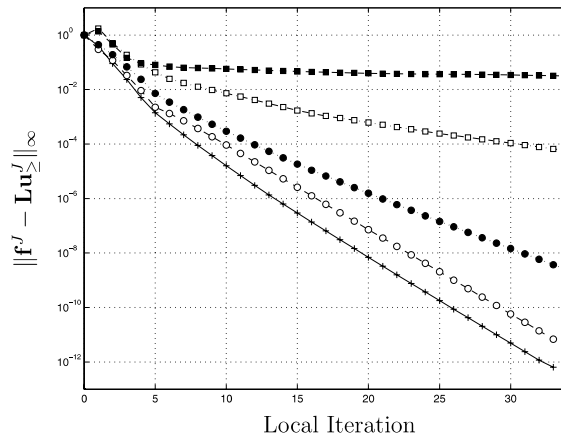


Fig. 10. The L_∞ -norm of the residual for the multilevel wavelet collocation solver as a function of local iterations (V-cycles) for $\epsilon = 10^{-5}$, $p = \tilde{p} = 6$, $v_1 = v_2 = 2$, $v_3 = 0$ and different choices of iterative parameters ω_0 , and ω_1 : $\omega_0 = 1$, $\omega_1 = 1$ (+); $\omega_0 = 0.75$, $\omega_1 = 1$ (○); $\omega_0 = 1.1$, $\omega_1 = 1$ (●); $\omega_0 = 1$, $\omega_1 = 0.75$ (□); $\omega_0 = 1$, $\omega_1 = 1.1$ (■).

convergence rate is more sensitive to the number of iterations of the exact solver (v_3), as clearly seen in Fig. 9. It goes from one order per three iterations for no iterations of the exact solver to two orders per three iterations with $v_3 = 10$. However, the computational cost of both BI-CGSTAB and especially GMRES is considerably more expensive. In addition, the increase of iterations v_3 of the exact solver based on GMRES, would make the Krylov space larger, which ultimately would result in an increase of memory use. For that reason we found it is better to use either GMRES with very few iterations of the exact solver or no iterations at all. Finally, the influence of the relaxation parameters ω_0 and ω_1 on the convergence of the method is demonstrated in Fig. 10. Due to optimality of the approximate solver in terms of smoothing the solution, both under-relaxation ($\omega < 1$) and over-relaxation ($\omega > 1$) of either ω_0 and ω_1 decreases the convergence of the residual. However, for other choices of elliptic operators it might not be so. For that reason the relaxation parameters could provide additional freedom for improving the convergence of the Algorithm 4.

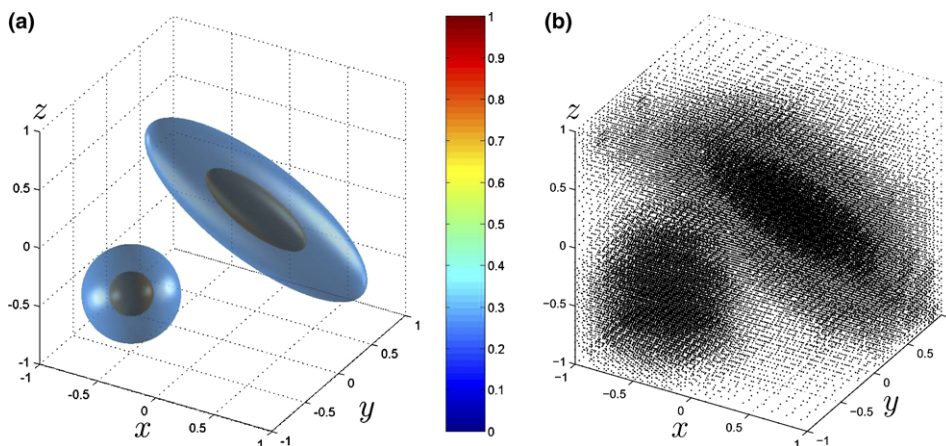


Fig. 11. The solution of three-dimensional elliptic problem (15) (isosurface levels are at 0.25 and 0.75) (a) and the corresponding adaptive computational grid \mathcal{G}_\geq (b) ($\epsilon = 10^{-5}$, $p = \tilde{p} = 6$).

4.2.3. *Three-dimensional results*

The proposed method works equally well for the solution of three-dimensional elliptic problems. The numerical solution of the three-dimensional elliptic problem is discussed in 4.1 and the corresponding computational grid is shown in Fig. 11. The results of the convergence study for the three-dimensional elliptic problem are presented in Fig. 12(a), where the pointwise l_∞ -error of the solution as a function of grid points is shown for different order wavelets. It is clearly seen that, as in two-dimensional case, the convergence is consistent with analytical predictions of Eq. (14), shown by triangles. The dependence of the number of grid points \mathcal{N} on the threshold parameter ϵ is shown in Fig. 12(b). Once again, the dependence of \mathcal{N} on ϵ is consistent with theoretical estimate given in (8), as shown by the triangles. The other global convergence results are similar to two-dimensional case. Finally, in order to demonstrate that the local convergence of the three-dimensional multilevel elliptic solver is similar to two-dimensional case, the l_∞ norm of the residual error as a function of local iterations (V-cycles) is shown in Fig. 13. The dependence of the rate of convergence on the relaxation parameters $\nu_1, \nu_2, \nu_3, \omega_0$, and ω_1 is similar to two-dimensional case.

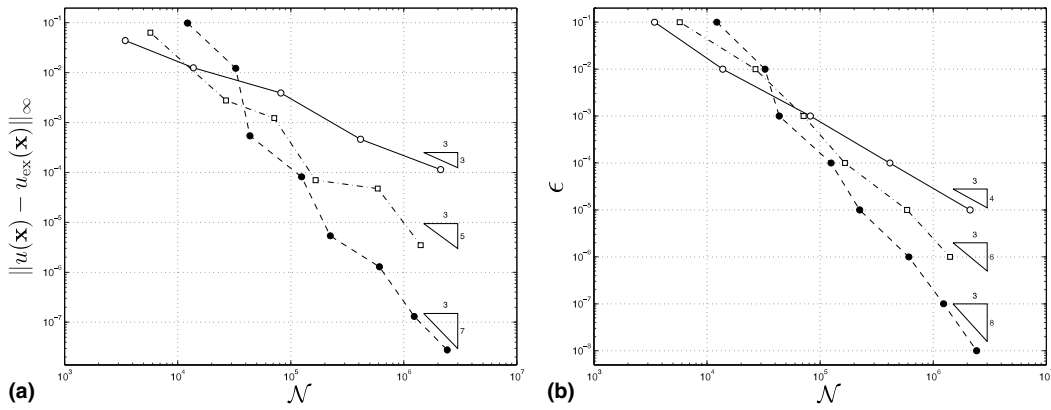


Fig. 12. The pointwise L_∞ -error of the solution of three-dimensional elliptic problem (15) (a) and the number of grid points as a function of ϵ (b) for different choices of parameters p , and \tilde{p} : $p = \tilde{p} = 4$ (○); $p = \tilde{p} = 6$ (□); $p = \tilde{p} = 8$ (●). \mathcal{N} is the actual number of grid points used in the calculations. The triangles indicate the slopes respectively predicted by Eqs. (14) and (8).

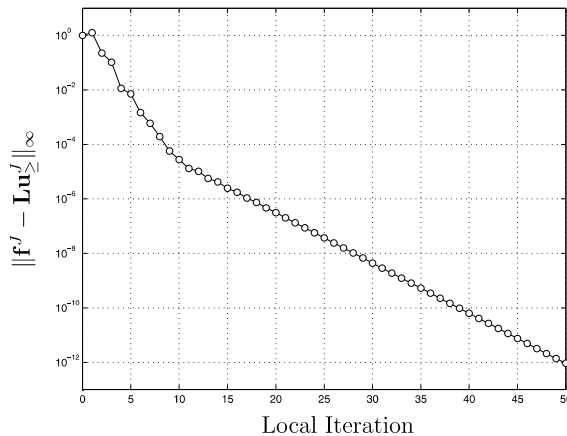


Fig. 13. The L_∞ -norm of the residual for the three-dimensional multilevel wavelet collocation solver as a function of local iterations (V-cycles) for $\epsilon = 10^{-5}$, $p = \tilde{p} = 6$, $\omega_0 = 2/3$, $\omega_1 = 1$, $\nu_1 = \nu_2 = 3$, $\nu_3 = 0$ (○).

5. Conclusions

The second generation wavelet collocation method [22,23] for solving evolution problems has been extended to the solution of elliptic problems using a multilevel method. Wavelet decomposition is used for grid adaptation and interpolation, while a $O(\mathcal{N})$ hierarchical finite difference scheme, which takes advantage of wavelet multilevel decomposition, is used for derivative calculations. An efficient adaptive algorithm for solving elliptic problems is developed. With this algorithm the solution is obtained on a near optimal grid for the prescribed tolerance that is controlled by wavelet threshold parameter ϵ . In order to accelerate the convergence of the adaptive method, an iterative multilevel procedure analogous to the multigrid algorithm is developed.

Despite similarities with multigrid methods, the wavelet multilevel iterative algorithm is different in several important ways. The major differences are in the structure of the nested grids, the use of wavelet interpolation for both prolongation and restriction operators, and the use of either BI-CGSTAB or GMRES methods as the exact solver. The accuracy and computational efficiency of this method are demonstrated for the solution of two- and three-dimensional Poisson equations with highly localized sources. Both global and local convergence results are presented. The theoretical prediction for the convergence is verified numerically. Linear convergence of the multilevel wavelet collocation solver is demonstrated. The results indicate that the computational grid and associated wavelets can efficiently adapt to the local irregularities of the solution in order to resolve sharp transition regions.

Finally, it should be noted that overall computational cost of the algorithm is data-structure (and hence problem size) dependent. In the current version of the algorithm, we used the simplest approach: a working array of the size of non-adaptive grid. This enabled us to perform the calculation on a subset of grid points with relative ease, although at the cost associated with a cache-inefficient algorithm for large problems. Working arrays are not efficient in terms of memory use, but they are efficient in terms of computational cost, since the computation cost scales linearly with the number of points used. We are currently developing an efficient data structure that is designed to take advantage of the special properties of the wavelet transform as well as computer architecture.

Another future area of research includes the generalization of the method to the solution of evolution problems in the space–time domain. This approach will provide a natural way of varying the time step based on location (in time and space), and spatial scale. The effective time step will be smaller in regions of rapid change (small local time scale), and larger in regions of slow change (large local time scale). This work is currently underway.

Acknowledgment

Partial support for the first author (O.V. Vasilyev) was provided by the National Science Foundation (NSF) under grants No. EAR-0242591, EAR-0327269 and ACI-0242457 and National Aeronautics and Space Administration (NASA) under grant No. NAG-1-02116. This support is gratefully acknowledged. The second author (N.K.-R. Kevlahan) was supported by the NSERC and gratefully acknowledges the use of SHARCNET computational resources.

References

- [1] I. Babuska, W.C. Rheinboldt, Error estimates for adaptive finite element computations, *SIAM J. Numer. Anal.* 15 (1978) 736–754.
- [2] I. Babuska, A. Miller, A feedback finite element method with a posteriori error estimation. part i: The finite element method and some basic properties of the a-posteriori error estimates, *Comp. Meth. Appl. Mech. Eng.* 61 (1987) 1–40.

- [3] R.E. Bank, A. Weiser, Some a posteriori error estimators for elliptic partial differential equations, *Math. Comput.* 44 (170) (1985) 283–301.
- [4] F.A. Bornemann, B. Erdmann, R. Kornhuber, A posteriori error estimates for elliptic problems in two and three space dimensions, *SIAM J. Numer. Anal.* 33 (3) (1996) 1188–1204.
- [5] S. Dahlke, W. Dahmen, R. Hochmuth, R. Schneider, Stable multiscale bases and local error estimation for elliptic problems, *Appl. Numer. Math.* 23 (1997) 21–47.
- [6] A. Cohen, W. Dahmen, R. DeVore, Adaptive wavelet methods for elliptic operator equations: Convergence rates, *Math. Comput.* 70 (233) (2001) 27–75.
- [7] A. Barinka, T. Barsch, P. Charton, A. Cohen, S. Dahlke, W. Dahmen, K. Urban, Adaptive wavelet schemes for elliptic problems – implementation and numerical experiments, *SIAM J. Sci. Comput.* 23 (3) (2001) 910–939.
- [8] R. Glowinski, A. Rieder, R.O. Wells, X. Zhou, A preconditioned CG-method for wavelet-Galerkin discretizations of elliptic problems, *Z. angew. Math. Mech.* 75 (1995) 683–684.
- [9] A. Cohen, R. Masson, Wavelet methods for second-order elliptic problems, preconditioning, and adaptivity, *SIAM J. Sci. Comput.* 21 (3) (1999) 1006–1026.
- [10] J. Liandrat, P. Tchamitchian, Resolution of the 1d regularized Burgers equation using a spatial wavelet approximation, Tech. Rep., NASA Contractor Report 187480, ICASE Report 90-83, NASA Langley Research Center, Hampton VA 23665-5225 (1990).
- [11] E. Bacry, S. Mallat, G. Papanicolaou, A wavelet based space-time adaptive numerical method for partial differential equation, *Math. Model. Numer. Anal.* 26 (7) (1992) 793–834.
- [12] G. Beylkin, J. Keiser, On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases, *J. Comput. Phys.* 132 (1997) 233–259.
- [13] J. Fröhlich, K. Schneider, An adaptive wavelet-vaguelette algorithm for the solution of PDEs, *J. Comput. Phys.* 130 (1997) 174–190.
- [14] M. Holmstrom, J. Walden, Adaptive wavelet methods for hyperbolic PDEs, *J. Sci. Comput.* 13 (1998) 19–49.
- [15] A. Harten, Adaptive multiresolution schemes for shock computations, *J. Comput. Phys.* 115 (1994) 319–338.
- [16] A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws, *Comm. Pure Appl. Math.* 48 (1995) 1305–1342.
- [17] O.V. Vasilyev, S. Paolucci, A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain, *J. Comput. Phys.* 125 (1996) 498–512.
- [18] W. Cai, J.Z. Wang, Adaptive multiresolution collocation methods for initial boundary value problems of nonlinear PDEs, *SIAM J. Numer. Anal.* 33 (1996) 937–970.
- [19] L. Jameson, A wavelet-optimized, very high order adaptive grid and order numerical method, *SIAM J. Sci. Comput.* 19 (6) (1998) 1980–2013.
- [20] M. Holmstrom, Solving hyperbolic PDEs using interpolating wavelets, *SIAM J. Sci. Comput.* 21 (2) (1999) 405–420.
- [21] O.V. Vasilyev, S. Paolucci, A fast adaptive wavelet collocation algorithm for multi-dimensional PDEs, *J. Comput. Phys.* 125 (1997) 16–56.
- [22] O.V. Vasilyev, C. Bowman, Second generation wavelet collocation method for the solution of partial differential equations, *J. Comp. Phys.* 165 (2000) 660–693.
- [23] O.V. Vasilyev, Solving multi-dimensional evolution problems with localized structures using second generation wavelets, *Int. J. Comp. Fluid Dyn.*, Special issue on High-resolution methods in Computational Fluid Dynamics 17 (2) (2003) 151–168.
- [24] G. Beylkin, On the representation of operators in bases of compactly supported wavelets, *SIAM J. Numer. Anal.* 29 (1992) 1716–1740.
- [25] Y. Maday, J.C. Ravel, Adaptativité par ondelettes: conditions aux limites et dimensions supérieures, *C.R. Acad. Sci. Paris* 315 (1992) 85–90.
- [26] J.-C. Xu, W.-C. Shann, Galerkin-wavelet methods for two-point boundary value problems, *Numer. Math.* 63 (1992) 123–144.
- [27] A. Cohen, I. Daubechies, Wavelets on the interval and fast wavelet transforms, *Appl. Comput. Harmon. Anal.* 1 (1993) 54–81.
- [28] L. Andersson, N. Hall, B. Jawerth, G. Peters, Wavelets on a closed subsets of the real line, in: L.L. Schumaker, G. Webb (Eds.), *Recent Advances in Wavelet Analysis*, Academic Press, New York, 1994, pp. 1–61.
- [29] W. Dahmen, A. Kunoth, Appending boundary conditions by Lagrange multipliers: Analysis of the LBB condition, *Numerische Mathematik* 88 (1) (2001) 9–42.
- [30] I. Babuska, J. Chandra, J.E. Flaherty, *Adaptive Computational Methods for Partial Differential Equations*, SIAM, Philadelphia, 1984.
- [31] J.E. Flaherty, *Adaptive Methods for Partial Differential Equations*, SIAM, Philadelphia, 1989.
- [32] I. Daubechies, Orthonormal bases of compactly supported wavelets, *Commun. Pure Appl. Math.* 41 (1988) 909–996.
- [33] A. Cohen, I. Daubechies, J. Feauveau, Bi-orthogonal bases of compactly supported wavelets, *Commun. Pures Appl. Math.* 45 (1992) 485–560.
- [34] W. Sweldens, The lifting scheme: A custom-design construction of biorthogonal wavelets, *Appl. Comput. Harmon. Anal.* 3 (2) (1996) 186–200.

- [35] W. Sweldens, The lifting scheme: A construction of second generation wavelets, *SIAM J. Math. Anal.* 29 (2) (1998) 511–546.
- [36] J.H. Bramble, *Multigrid Methods*, CRC Press, Boca Raton, FL, 1993.
- [37] I. Daubechies, *Ten Lectures on Wavelets*, No. 61 in CBMS-NSF Series in Applied Mathematics, SIAM, Philadelphia, 1992.
- [38] D.L. Donoho, *Interpolating wavelet transforms*, Tech. Rep. 408, Department of Statistics, Stanford University (1992).
- [39] O.V. Vasilyev, D.A. Yuen, Y.Y. Podladchikov, Applicability of wavelet algorithm for geophysical viscoelastic flow, *Geophys. Res. Lett.* 24 (23) (1997) 3097–3100.
- [40] O.V. Vasilyev, Y.Y. Podladchikov, D.A. Yuen, Modeling of compaction driven flow in poro-viscoelastic medium using adaptive wavelet collocation method, *Geophys. Res. Lett.* 25 (17) (1998) 3239–3242.
- [41] O.V. Vasilyev, Y.Y. Podladchikov, D.A. Yuen, Modeling of viscoelastic plume-lithosphere interaction using adaptive multilevel wavelet collocation method, *Geophys. J. Int.* 147 (3) (2001) 579–589.
- [42] O.V. Vasilyev, N.K.-R. Kevlahan, Hybrid wavelet collocation – Brinkman penalization method for complex geometry flows, *Int. J. Numer. Meth. Fluids* 40 (2002) 531–538.
- [43] N.K.-R. Kevlahan, O.V. Vasilyev, A three-dimensional adaptive wavelet method for fluid-structure interaction, *SIAM J. Sci. Comput.*, in press.
- [44] H.A. van der Vorst, BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 13 (2) (1992) 631–644.
- [45] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 7 (3) (1986) 856–869.
- [46] A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comp.* 31 (1977) 333–390.