Paper for CRC Press

On wavelet-based algorithms for solving differential equations.

G. Beylkin
Program in Applied Mathematics
University of Colorado at Boulder, Boulder, CO 80309-0526

I Introduction

The role of the orthonormal wavelet bases in solving integral equations has been studied in [4], where it was observed that wide classes of operators have sparse representations in the wavelet bases thus permitting a number of fast algorithms for applying these operators to functions, solving integral equations, etc. The operators which can be efficiently treated using representations in the wavelet bases include Calderón-Zygmund and pseudo-differential operators. Let us here summarize several points important for further considerations and refer to [4], [5], [1] for the details. If we consider an integral operator (Calderón-Zygmund or pseudo-differential operator),

$$T(f)(x) = \int K(x,y) f(y) dy, \tag{1.1}$$

and construct its matrix representation in a two-dimensional wavelet basis, then we find that the rate of decay of the size of entries as a function of the distance from the diagonal in the sub-blocks of such representation is faster than that of the original kernel. The rate of decay depends on the number of vanishing moments of the basis functions. For example, let the kernel satisfy the conditions

$$|K(x,y)| \leq \frac{1}{|x-y|}, \tag{1.2}$$

$$|\partial_x^M K(x,y)| + |\partial_y^M K(x,y)| \leq \frac{C_0}{|x-y|^{1+M}} \tag{1.3}$$

for some $M \geq 1$. Then by choosing the wavelet basis with M vanishing moments, the matrices of coefficients $\alpha^j_{i,l}$, $\beta^j_{i,l}$, $\gamma^j_{i,l}$ of the representation of the kernel K in the non-standard form (see [4]) satisfy the estimate

$$|\alpha^j_{i,l}| + |\beta^j_{i,l}| + |\gamma^j_{i,l}| \leq \frac{C_M}{1+|i-l|^{M+1}}, \tag{1.4}$$

for all

$$|\ i-l\ | \geq 2M. \tag{1.5}$$

And, if in addition to (1.2) and (1.3),

$$| \int_{I\times I} K(x,y)\ dxdy \,| \leq C|I| \tag{1.6}$$

holds for all dyadic intervals I (the so-called "weak cancellation condition"), then (1.4) is valid for all i, l. Thus, for a given accuracy, the representations of operators which satisfy (1.2) and (1.3) are sparse since we may use banded versions of $\alpha^j_{i,l}$, $\beta^j_{i,l}$, $\gamma^j_{i,l}$ for computing.

We note that considering a banded approximation directly for the kernel satisfying (1.2) does not lead to a satisfactory numerical approximation. The method of [4] uses the smoothness of the matrix away from the diagonal to increase the rate of decay. Parts of the matrix which can be well approximated by the low degree polynomials are represented by small coefficients in the wavelet system of coordinates since the basis functions have vanishing moments. Once a sparse representation is obtained, fast algorithms are available for a variety of tasks associated with solving integral equations, for example $O(N)$ algorithm for solving the integral equation or an iterative algorithm for constructing the generalized inverse in $O(N)$ operations [4], [5], [1].

In this paper we address a question "what are the implications of using wavelet bases for solving differential equations?" if our starting point is a differential equation rather than an integral equation. The same problem often may be posed both as a problem of solving the boundary value problem for a differential equation and as a problem of solving an integral equation. From the point of view of numerical analysis one would observe a significant difference in these formulations. The discretization of a differential equation leads to a sparse linear algebraic system with a "large" condition number of the corresponding matrix, whereas the discretization of an integral operator leads to a dense matrix with "small" condition number. More precisely, the condition numbers of matrices representing differential operators usually have a polynomial growth with the reduction of the step size (i.e., with the increase of the size of the system). For example, the condition number of the matrix of the second order finite difference operator grows as $1/h^2$ (or N^2), where h is the step size (N is the number of points of discretization). On the other hand, the condition number of the matrix of the linear algebraic system obtained by discretizing the second kind integral equation does not grow with the reduction of the step size (usually it actually improves somewhat) but the matrix is dense (full).

We recall that the condition number of a matrix is defined as the ratio of the largest and the smallest singular values. If matrix has a null space (the actual null space or a null space for a given accuracy), then by the condition number we understand the ratio of the largest singular value to the smallest singular value *above* the theshold of accuracy. The condition number controls the rate of convergence of a number of iterative algorithms for solving linear systems; for example the number of iterations of the conjugate gradient method is $O(\sqrt{\kappa})$, where κ is the condition number of the matrix.

Naively, solving the linear system of size $N \times N$ obtained by discretizing the second kind integral equations seems to require $O(N^2)$ operations. Solving the sparse linear system obtained by representing differential operators also seems to require $O(N^2)$ operations since the number of iterations is of $O(N)$.

As it is shown in [4], the dense matrices obtained by discretizing the second kind integral equations may be replaced by sparse matrices in the wavelet system of coordinates (for operators satisfying the conditions (1.2) and (1.3), for example), thus leading to $O(N)$ algorithms for solving such integral equations. As we demonstrate in this paper, if our starting point is a differential equation with boundary conditions then in the wavelet system of coordinates there is a *diagonal* preconditioner which allows us to perform algebraic manipulations only with the sparse matrices whose condition number is $O(1)$, thus also leading to $O(N)$ algorithms for solving the corresponding linear systems.

We describe a method for solving the two point boundary value problem for elliptic differential operators in the wavelet "system of coordinates". To illustrate the difference between our approach and the existing numerical methods for solving the two-point boundary value problems of this kind, such as multigrid (see e.g. [7]) or multilevel (hierarchical) methods or very simple and elegant algorithm of [9], we construct the Green's function (the inverse operator) in $O(N)$ operation. We note that the numerical methods mentioned above allow us to find the solution of the problem in $O(N)$ operations. However, since the ordinary matrix representation of the Green's function requires $O(N^2)$ significant entries, fast algorithms for its construction are not readily available. Our method permits solving the problem in $O(N)$ operations as well, but since the representation of the Green's function in the wavelet bases requires (for a given accuracy) only $O(N)$ entries, we concentrate on describing a fast algorithm for its construction.

Once the Green's function is obtained, finding the solution reduces to the matrix-vector multiplication, which in the wavelet system of coordinates is an $O(N)$ procedure. In addition, if the entries of the vector are values of a smooth and non-oscillatory function then the vector is sparse in the wavelet system of coordinates. In this case the number of operations to apply the Green's function to a vector is proportional to the number of significant coefficients of this vector in the wavelet system of coordinates. We illustrate these properties further by considering a modification of Crank-Nicolson method, which we convert into an explicit and adaptive scheme in the wavelet system of coordinates.

The main tool in our approach is the diagonal preconditioning [2] available for the periodized differential operator in the wavelet bases. The idea of preconditioning has long been one of the main ideas in the multilevel and multigrid methods. Among a great number of papers on the preconditioning we would like to note [6], where authors explicitly consider orthonormal chains of subspaces (similar to that of the multiresolution analysis) in order to construct the multilevel preconditioners. Apparently unfamiliar with the multiresolution analysis and the wavelet bases, they remark that "in practice, an orthonormal basis ... is seldom available". In fact, the orthonormal wavelet bases provides a very convenient tool for implementing the preconditioners. Moreover, since the inverse operator is sparse in the wavelet bases, it is possible to construct it numerically in $O(N)$ operations.

S. Jaffard in [10] gives a theoretical analysis of solving the elliptic boundary value problem in the wavelet bases and considers the diagonal preconditioning. However, he does not provide a practical method. In this paper our considerations are restricted to the two-point boundary value problems, since a practical construction of the wavelet bases in an arbitrary domain is not available at this time. We approach the multidimensional problems using the alternating directions technique, which is modified since we are able to construct numerically the Green's functions of the two-point boundary value problems. We note that our use of the diagonal preconditioning differs from that in [10] since we apply it to the periodized differential operators and solve the boundary value problem by rank-one perturbation.

For simplicity, we consider the ordinary $O(h^2)$ finite-difference scheme for the two-point boundary value problem, and use the periodized wavelets only to perform the "linear algebra". Such approach enables us to make a clear comparison with other techniques. On the other hand, it also carries some of the limitations of the finite-difference scheme. A more consistent approach which uses the wavelet bases of the interval [8] to achieve an approximation of order h^p, where p is arbitrary, is currently being developed and will be described elsewhere.

II The two-point boundary value problem

Let us consider the two-point boundary value problem

$$\mathcal{L}u \equiv \frac{d}{dx}\left(a(x)\,\frac{du}{dx}\right) = f(x) \tag{2.1}$$

with the Dirichlet boundary conditions $u(0) = u(1) = 0$. We assume that a is a sufficiently smooth function and $a(x) > 0$, $x \in (0,1)$. The method that we describe is applicable to more general elliptic operators, e.g.

$$\mathcal{L}u = \frac{d}{dx}\left(a(x)\,\frac{du}{dx}\right) - b(x)u, \tag{2.2}$$

where $b(x) > 0$.

Discretizing this problem on a staggered grid, we obtain the following system of linear algebraic equations

$$a_{i-\frac{1}{2}}u_{i-1} - (a_{i-\frac{1}{2}} + a_{i+\frac{1}{2}})u_i + a_{i+\frac{1}{2}}u_{i+1} = h^2 f_i, \quad i = 1,\ldots,N, \tag{2.3}$$

where $u_i = u(x_i)$, $a_{i+\frac{1}{2}} = a(x_{i+\frac{1}{2}})$, $f_i = f(x_i)$, $x_i = ih$ and $x_{i+\frac{1}{2}} = (i+\frac{1}{2})h$ and where we explicitly set $u_0 = u_{N+1} = 0$.

We write (2.3) as

$$\mathbf{Lu} = \mathbf{f}, \tag{2.4}$$

where the $N \times N$ matrix $\mathbf{L}$ is as follows

$$\mathbf{L} = \begin{pmatrix} -(a_{\frac{1}{2}} + a_{\frac{3}{2}}) & a_{\frac{3}{2}} & 0 & \cdots & 0 & 0 & 0 \\ a_{\frac{3}{2}} & -(a_{\frac{3}{2}} + a_{\frac{5}{2}}) & a_{\frac{5}{2}} & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & a_{N-\frac{3}{2}} & -(a_{N-\frac{3}{2}} + a_{N-\frac{1}{2}}) & a_{N-\frac{1}{2}} \\ 0 & 0 & 0 & \cdots & 0 & a_{N-\frac{1}{2}} & -(a_{N-\frac{1}{2}} + a_{N+\frac{1}{2}}) \end{pmatrix}. \tag{2.5}$$

There are two reasons for the condition number of the matrix $\mathbf{L}$ to be large. If $a(x) = 1$ in (2.1) and (2.5), then we obtain the central difference matrix representation of the second derivative d^2/dx^2. It is clear that the matrix $\mathbf{L}$ has the condition number $O(N^2)$. On the other hand, noting that the size of the function a might be different in the subintervals of $(0,1)$, we observe that the condition number of the operator of multiplication by the function $a(x)$ could be arbitrarily large.

Our goal is to construct the matrix $\mathbf{L}^{-1}$ numerically in $O(-\log\epsilon\ N)$ operations, where ϵ is the desired accuracy. This seemingly hopeless task (it is easy to check for small N that the matrix $\mathbf{L}^{-1}$ is dense in the ordinary representation) has, in fact, a simple solution in the wavelet system of coordinates.

The kernel of the inverse operator for the problem (2.1) (the Green's function for the Dirichlet problem for an elliptic operator) has a sparse representation in the wavelet bases since such kernel satisfies the estimates of the type in (1.2), (1.3) (see [4]). Let us show how to construct $\mathbf{L}^{-1}$ numerically by starting with the matrix $\mathbf{L}$.

III Reduction to the periodized problem

In the wavelet bases the preconditioner for the periodized differential operator is a diagonal matrix. The condition number of the rescaled operator is $O(1)$ and depends only on the choice of the basis [2]. Moreover, any finite difference matrix representation of periodized differential operators may be rescaled by a diagonal preconditioner. We use this fact to solve the two-point boundary value problem using a fairly standard discretization scheme in (2.4). The wavelets play an auxiliary role in that they provide a system of coordinates where the condition numbers of the sparse matrices (involved in the computations) are under control. We use such "mixed" approach for two reasons. First, it provides a simple way to see the advantages of computing in the wavelet bases. Second, it provides a practical way to significantly improve the performance of commonly used finite difference schemes.

In order to use periodized differential operators, we consider the matrix $\mathbf{L}$ as a finite rank perturbation of a periodized matrix. Indeed, we have

$$\mathbf{L} = \mathbf{A} - a_{\frac{1}{2}}\mathbf{e}_1\mathbf{e}_N^T - a_{N+\frac{1}{2}}\mathbf{e}_N\mathbf{e}_1^T, \tag{3.1}$$

where

$$\mathbf{A} = \begin{pmatrix} -(a_{\frac{1}{2}} + a_{\frac{3}{2}}) & a_{\frac{3}{2}} & 0 & \cdots & 0 & 0 & a_{\frac{1}{2}} \\ a_{\frac{3}{2}} & -(a_{\frac{3}{2}} + a_{\frac{5}{2}}) & a_{\frac{5}{2}} & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & a_{N-\frac{3}{2}} & -(a_{N-\frac{3}{2}} + a_{N-\frac{1}{2}}) & a_{N-\frac{1}{2}} \\ a_{N+\frac{1}{2}} & 0 & 0 & \cdots & 0 & a_{N-\frac{1}{2}} & -(a_{N-\frac{1}{2}} + a_{N+\frac{1}{2}}) \end{pmatrix}, \tag{3.2}$$

and the unit vectors $\mathbf{e}_1$, $\mathbf{e}_N$,

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \mathbf{e}_N = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}. \tag{3.3}$$

In this section we consider the case where the size of the function a does not change significantly over the interval $(0, 1)$. To illustrate the effect of the diagonal preconditioning in the wavelet system of coordinates, let us set $a = 1$ and consider $\mathbf{A} = \mathbf{D}$,

$$\mathbf{D} = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 & 1 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 1 & -2 \end{pmatrix}. \tag{3.4}$$

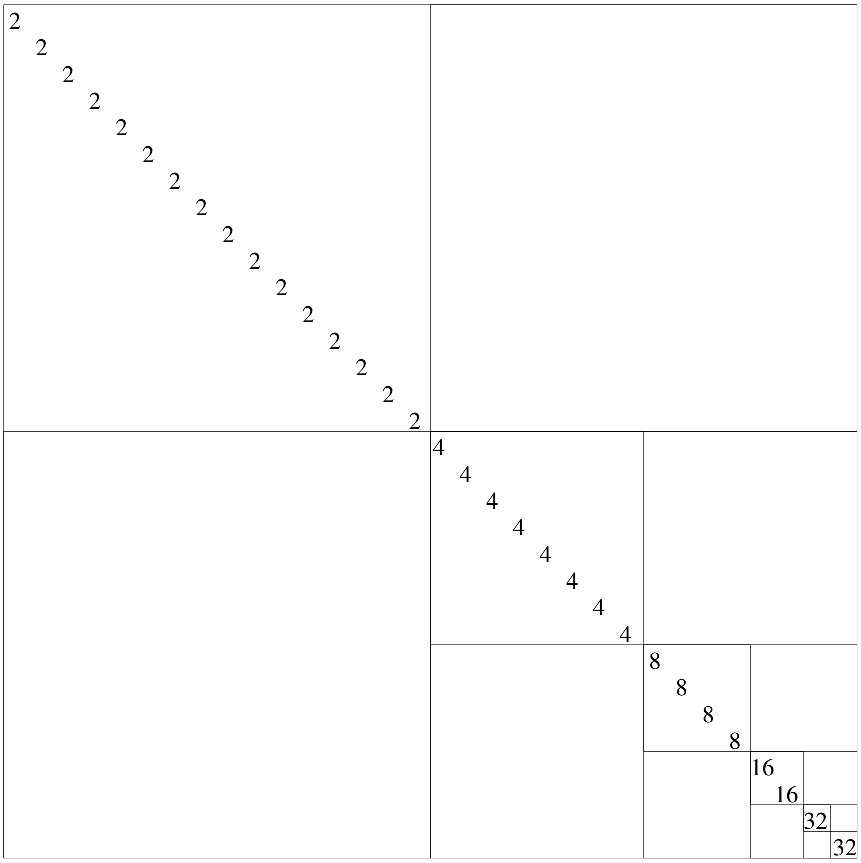

Figure 1: An example ($n = 5$) of the diagonal martix $\mathbf{P}$ used to rescale the matrix of the periodized second derivative $\mathbf{D}_w$ in the wavelet system of coordinates.

In the following two examples we compute the standard form $\mathbf{D}_w$ of the periodized second derivative $\mathbf{D}$ of size $N \times N$, where $N = 2^n$, and rescale it by the diagonal matrix $\mathbf{P}$,

$$\mathbf{D}_w^p = \mathbf{P}\mathbf{D}_w\mathbf{P}.$$

where $\mathbf{P}_{il} = \delta_{il} 2^j$, $1 \leq j \leq n$, and where j is chosen depending on i, l so that $N - N/2^{j-1} + 1 \leq i, l \leq N - N/2^j$, and $\mathbf{P}_{NN} = 2^n$. The matrix P is illustrated in Figure 1.

The following tables compare the original condition number κ of $\mathbf{D}$ (and $\mathbf{D}_w$ since it is obtained by an orthogonal transformation) and κ_p of $\mathbf{D}_w^p$. Since matrices $\mathbf{D}$, $\mathbf{D}_w$ and $\mathbf{D}_w^p$ have a null space (of dimension one), the condition numbers are computed on the range of these matrices.

N	κ	κ_p
32	$0.10409 \cdot 10^3$	8.021
64	$0.41535 \cdot 10^3$	9.086
128	$0.16605 \cdot 10^4$	10.019
256	$0.66405 \cdot 10^4$	10.841
512	$0.26562 \cdot 10^5$	11.562
1024	$0.10625 \cdot 10^6$	12.197

Table 1.

Condition numbers of the matrix of periodized second derivative (with and without preconditioning) in the system of coordinates associated with Daubechies' wavelets with three vanishing moments $M = 3$.

N	κ	κ_p
32	$0.10409 \cdot 10^3$	5.2002
64	$0.41535 \cdot 10^3$	5.2610
128	$0.16605 \cdot 10^4$	5.2897
256	$0.66405 \cdot 10^4$	5.3035
512	$0.26562 \cdot 10^5$	5.3103
1024	$0.10625 \cdot 10^6$	5.3137

Table 2.

Condition numbers of the matrix of periodized second derivative (with and without preconditioning) in the system of coordinates associated with Daubechies' wavelets with six vanishing moments $M = 6$.

Let us now describe a method for solving the two-point boundary value problem in the wavelet system of coordinates. Denoting the matrix of the discrete wavelet transform by **W** (though the actual transform is applied via a pyramid algorithm, see e.g. [4]) and observing that $\mathbf{W}$ is an orthogonal transformation, we rewrite (2.4) and (3.1) in the wavelet system of coordinates,

$$\left(\mathbf{A}_w - a_{\frac{1}{2}}\hat{\mathbf{e}}_1\hat{\mathbf{e}}_N^T - a_{N+\frac{1}{2}}\hat{\mathbf{e}}_N\hat{\mathbf{e}}_1^T\right)\hat{\mathbf{u}} = \hat{\mathbf{f}} \tag{3.5}$$

where

$$\mathbf{A}_w = \mathbf{WAW}^* \tag{3.6}$$

?

$$\hat{\mathbf{u}} = \mathbf{Wu} \tag{3.7}$$

$$\hat{\mathbf{f}} = \mathbf{Wf} \tag{3.8}$$

and $\hat{\mathbf{e}}_l = \mathbf{We}_l$, $l = 1, N$.

Computing the discrete (periodized) wavelet transform of a vector of size $N = 2^n$ and using n scales, we obtain on the most sparse scale a single coefficient for differences and a single coefficient for averages which we call the total average. We note that the total average of a vector is proportional to the direct sum of the elements of the vector. The sum of the entries in the rows of the matrix $\mathbf{A}$ is identically zero and, therefore, the matrix $\mathbf{A}_w$ has the following structure,

$$\mathbf{A}_w = \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{c}^T & 0 \end{pmatrix}, \tag{3.9}$$

where $\mathbf{B}$ is an $(N-1) \times (N-1)$ full rank matrix with the condition number proportional to N^2. Let us now determine the vector $\mathbf{c}^T$. If we compute $\mathbf{A}_w$ by first applying the transformation to the columns of $\mathbf{A}$ we obtain the last row of the transformed matrix as

$$\rho(a_{N+\frac{1}{2}} - a_{\frac{1}{2}})(\mathbf{e}_1^T - \mathbf{e}_N^T), \tag{3.10}$$

where ρ is a factor which depends on the size of the matrix $\mathbf{A}$. In order to obtain $\mathbf{A}_w$, we have to transform further by applying the wavelet transform to the rows of the intermediate result. Thus, we obtain

$$(\mathbf{c}^T, 0) = \rho(a_{N+\frac{1}{2}} - a_{\frac{1}{2}})(\hat{\mathbf{e}}_1^T - \hat{\mathbf{e}}_N^T). \tag{3.11}$$

Let us intoduce the following notation

$$\hat{\mathbf{e}}_l = \begin{pmatrix} \mathbf{r}_l \\ \rho \end{pmatrix}, \quad l = 1, N, \tag{3.12}$$

where $\mathbf{r}_l$ are vectors of size $N-1$ and ρ is a scalar factor (common to both vectors),

$$\hat{\mathbf{u}} = \begin{pmatrix} \mathbf{d} \\ s \end{pmatrix}, \tag{3.13}$$

and

$$\hat{\mathbf{f}} = \begin{pmatrix} \mathbf{f}^d \\ f^s \end{pmatrix}. \tag{3.14}$$

Also, let $2a = a_{\frac{1}{2}} + a_{N+\frac{1}{2}}$, $\alpha = a_{\frac{1}{2}}/(2a)$, $\beta = a_{N+\frac{1}{2}}/(2a)$, so that $\alpha + \beta = 1$. We now rewrite (3.5) as

$$\left[\begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{c}^T & 0 \end{pmatrix} - 2a \begin{pmatrix} \alpha \mathbf{r}_1 \mathbf{r}_N^T + \beta \mathbf{r}_N \mathbf{r}_1^T & \rho(\alpha \mathbf{r}_1 + \beta \mathbf{r}_N) \\ \rho(\alpha \mathbf{r}_N^T + \beta \mathbf{r}_1^T) & \rho^2 \end{pmatrix} \right] \begin{pmatrix} \mathbf{d} \\ s \end{pmatrix} = \begin{pmatrix} \mathbf{f}^d \\ f^s \end{pmatrix}, \tag{3.15}$$

where

$$\mathbf{c}^T = 2a\rho(\beta - \alpha)(\mathbf{r}_1^T - \mathbf{r}_N^T). \tag{3.16}$$

By eliminating s,

$$s = -\frac{(\alpha \mathbf{r}_1^T + \beta \mathbf{r}_N^T)\mathbf{d}}{\rho} - \frac{f^s}{2a\rho^2}, \tag{3.17}$$

we obtain the $(N-1) \times (N-1)$ system of linear algebraic equations for $\mathbf{d}$,

$$\left[\mathbf{B} + 2a(\alpha^2 \mathbf{r}_1 - \beta^2 \mathbf{r}_N)(\mathbf{r}_1^T - \mathbf{r}_N^T) \right] \mathbf{d} = \mathbf{f}^d - \frac{f^s}{\rho} \left(\alpha \mathbf{r}_1 + \beta \mathbf{r}_N \right). \tag{3.18}$$

Our method for solving the two-point boundary value problem (2.1) is based on the fact that the matrix $\mathbf{B}^{-1}$ is sparse and could be computed in $O(N)$ operation. We will construct the matrix $\mathbf{B}^{-1}$ in the next Section and here let us assume that it is available. Given the matrix $\mathbf{B}^{-1}$, we solve (3.18) using Sherman-Morrison formula for the rank-one update of the inverse matrix. We obtain

$$\mathbf{d} = \left[\mathbf{B}^{-1} - \sigma \mathbf{B}^{-1}(\alpha^2 \mathbf{r}_1 - \beta^2 \mathbf{r}_N)(\mathbf{r}_1 - \mathbf{r}_N)^T \mathbf{B}^{-1} \right] \left[\mathbf{f}^d - \frac{f^s}{\rho} \left(\alpha \mathbf{r}_1 + \beta \mathbf{r}_N \right) \right] \tag{3.19}$$

where

$$\sigma = \frac{2a}{1 + 2a(\mathbf{r}_1 - \mathbf{r}_N)^T \mathbf{B}^{-1}(\alpha^2 \mathbf{r}_1 - \beta^2 \mathbf{r}_N)}. \tag{3.20}$$

Remark 1. The condition number of the sparse matrix $\mathbf{B}$ after rescaling by $\mathbf{P}$ is $O(1)$ as is illustrated in Tables 1 and 2. Thus, the linear system (3.18) may be solved using (3.19) by the standard iterative methods (e.g. conjugate gradient) in $O(N)$ operations since using (3.19) only involves finding the solution of the linear system $\mathbf{Bx} = \mathbf{y}$.

We look for the inverse operator in the form

$$\mathbf{L}^{-1} = \begin{pmatrix} \mathbf{\Gamma} & \mathbf{p} \\ \mathbf{q}^T & \gamma \end{pmatrix}, \tag{3.21}$$

and we obtain

$$\mathbf{\Gamma} = \left[\mathbf{B}^{-1} - \sigma \mathbf{B}^{-1}(\alpha^2 \mathbf{r}_1 - \beta^2 \mathbf{r}_N)(\mathbf{r}_1 - \mathbf{r}_N)^T \mathbf{B}^{-1} \right], \tag{3.22}$$

$$\mathbf{p} = -\frac{1}{\rho}\left[\mathbf{B}^{-1}\left(\alpha\mathbf{r}_1 + \beta\mathbf{r}_N\right) - \sigma\kappa_1\mathbf{B}^{-1}(\alpha^2\mathbf{r}_1 - \beta^2\mathbf{r}_N)\right], \tag{3.23}$$

$$\mathbf{q}^T = -\frac{1}{\rho}\left[\left(\alpha\mathbf{r}_1^T + \beta\mathbf{r}_N^T\right)\mathbf{B}^{-1} - \sigma\kappa_2(\mathbf{r}_1 - \mathbf{r}_N)^T\mathbf{B}^{-1}\right], \tag{3.24}$$

$$\gamma = \frac{1}{\rho^2}\left(\kappa_3 - \sigma\kappa_2\kappa_1 - \frac{1}{2a}\right), \tag{3.25}$$

where

$$\kappa_1 = (\mathbf{r}_1 - \mathbf{r}_N)^T\mathbf{B}^{-1}(\alpha\mathbf{r}_1 + \beta\mathbf{r}_N) \tag{3.26}$$

$$\kappa_2 = \left(\alpha\mathbf{r}_1^T + \beta\mathbf{r}_N^T\right)\mathbf{B}^{-1}(\alpha^2\mathbf{r}_1 - \beta^2\mathbf{r}_N), \tag{3.27}$$

and

$$\kappa_3 = \left(\alpha\mathbf{r}_1^T + \beta\mathbf{r}_N^T\right)\mathbf{B}^{-1}\left(\alpha\mathbf{r}_1 + \beta\mathbf{r}_N\right). \tag{3.28}$$

All matrix-vector multiplications in (3.22)-(3.28) involve sparse matrix $\mathbf{B}^{-1}$ and sparse vectors $\mathbf{r}_1$ and $\mathbf{r}_N$. Thus, the problem of constructing $\mathbf{L}^{-1}$ is reduced to that of computing the matrix $\mathbf{B}^{-1}$.

IV Computing inverse of the periodized operator

We start by rescaling the $(N-1) \times (N-1)$ matrix $\mathbf{B}$ by the diagonal matrix $\mathbf{P}$, where $\mathbf{P}_{il} = \delta_{il} 2^j$, $1 \leq j \leq n$, and where j is chosen depending on i, l so that $N - N/2^{j-1} + 1 \leq i, l \leq N - N/2^j$ (see Figure 1). We have

$$\mathbf{B}_p = \mathbf{P}\mathbf{B}\mathbf{P}, \tag{4.1}$$

and the condition number of the matrix $\mathbf{B}_p$ is $O(1)$ (see Tables 1 and 2). The matrices $\mathbf{B}$ and $\mathbf{B}_p$ are sparse matrices which is illustrated in Figures 2 and 3. Also, the matrices $\mathbf{B}$ and $\mathbf{B}_p$ are full rank.

Our main tool in computing the inverse matrix $\mathbf{B}_p^{-1}$ is the iterative algorithm [11]

$$\mathbf{X}_{l+1} = 2\mathbf{X}_l - \mathbf{X}_l \mathbf{B}_p \mathbf{X}_l, \tag{4.2}$$

which is initialized by setting

$$\mathbf{X}_0 = \alpha \mathbf{B}_p^*, \tag{4.3}$$

where α is chosen so that $0 < \alpha < 2/\sigma_1$, where σ_1 is the largest singular value of $\mathbf{B}_p$.

For the full rank matrices the iteration (4.2) converges to $\mathbf{B}_p^{-1}$. The number of iterations is proportional to the logarithm of the condition number of the matrix $\mathbf{B}_p$ and, thus, is of $O(1)$. For the full rank matrices the iteration (4.2) is self-correcting and we use this property as described below.

The iteration (4.2) provides an $O(N)$ algorithm to compute the inverse matrix if $\mathbf{B}_p$, $\mathbf{B}_p^{-1}$ and all the intermediate matrices $\mathbf{X}_l$, have a sparse representation in the wavelet basis [3], [5], [1]. Since we know in advance that $\mathbf{B}^{-1}$ is sparse in the wavelet basis (for a given accuracy ϵ), we only need to maintain sparsity of the intermediate matrices $\mathbf{X}_l$.

Since the iteration is self-correcting, we first compute the low-accuracy inverse by removing all entries with absolute value less than a given threshold (e.g. 10^{-2}) after each iteration. Once the iteration converges we improve the accuracy of the inverse matrix by continuing the iteration and decreasing the threshold of accuracy. The sparsity of the resulting matrix $\mathbf{B}_p^{-1}$ is illustrated in Figure 3.

Finally, to obtain $\mathbf{B}^{-1}$, we have

$$\mathbf{B}^{-1} = \mathbf{P}\mathbf{B}_p^{-1}\mathbf{P}. \tag{4.4}$$

We note that since the matrix $\mathbf{P}$ is a diagonal matrix, there is no loss of accuracy in computing via (4.1) or (4.4), since only the operation of multiplication is involved. In our particular case the multiplication is by the powers of 2 only and, thus, no rounding errors are introduced.

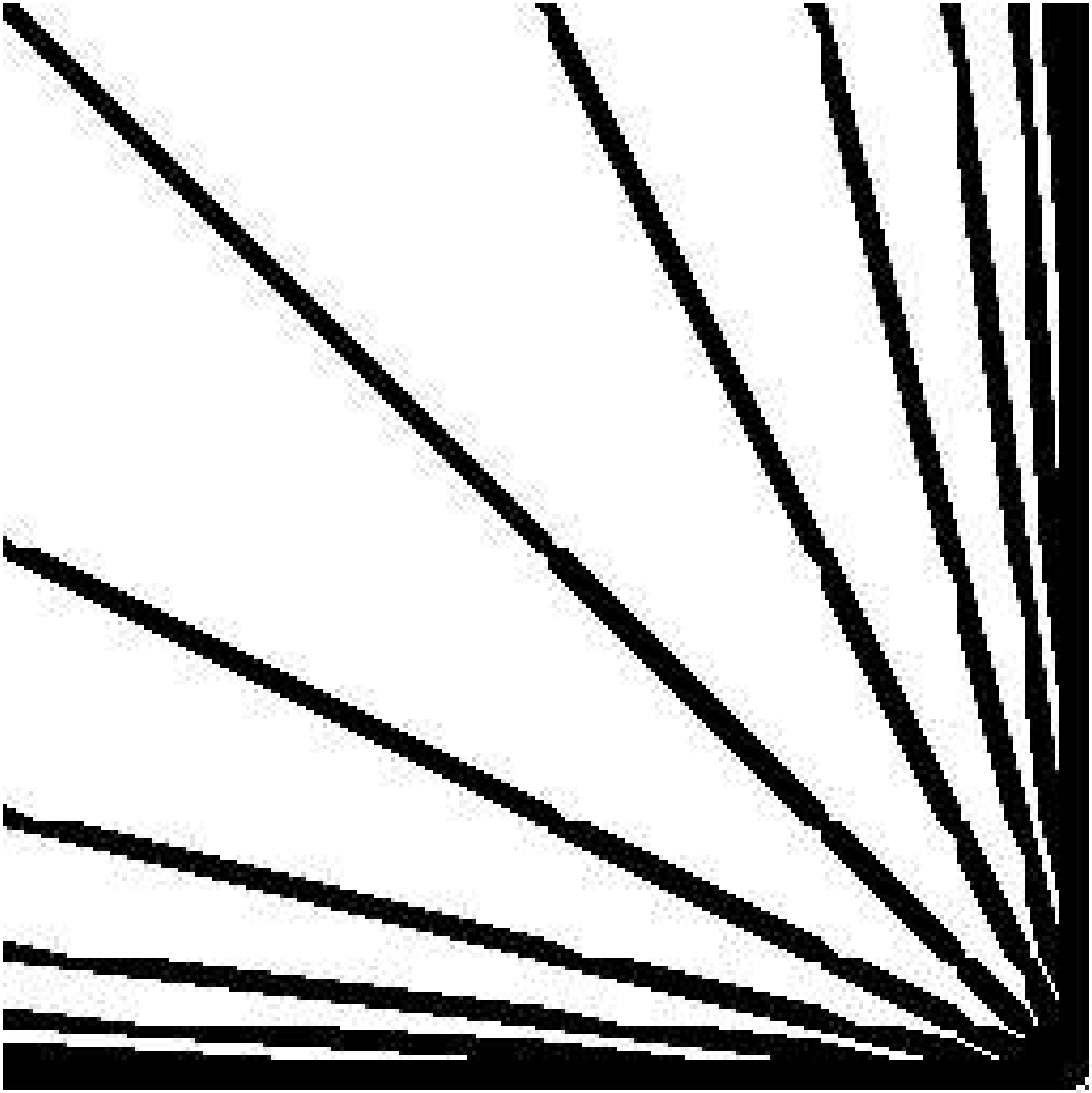

Figure 2: Matrix $\mathbf{B}$ (in the case $\mathbf{A} = \mathbf{D}$) of size 255×255 in the system of coordinates associated with the basis of Daubechies' wavelets with 3 vanishing moments. Entries with the absolute value greater than 10^{-14} are shown black.

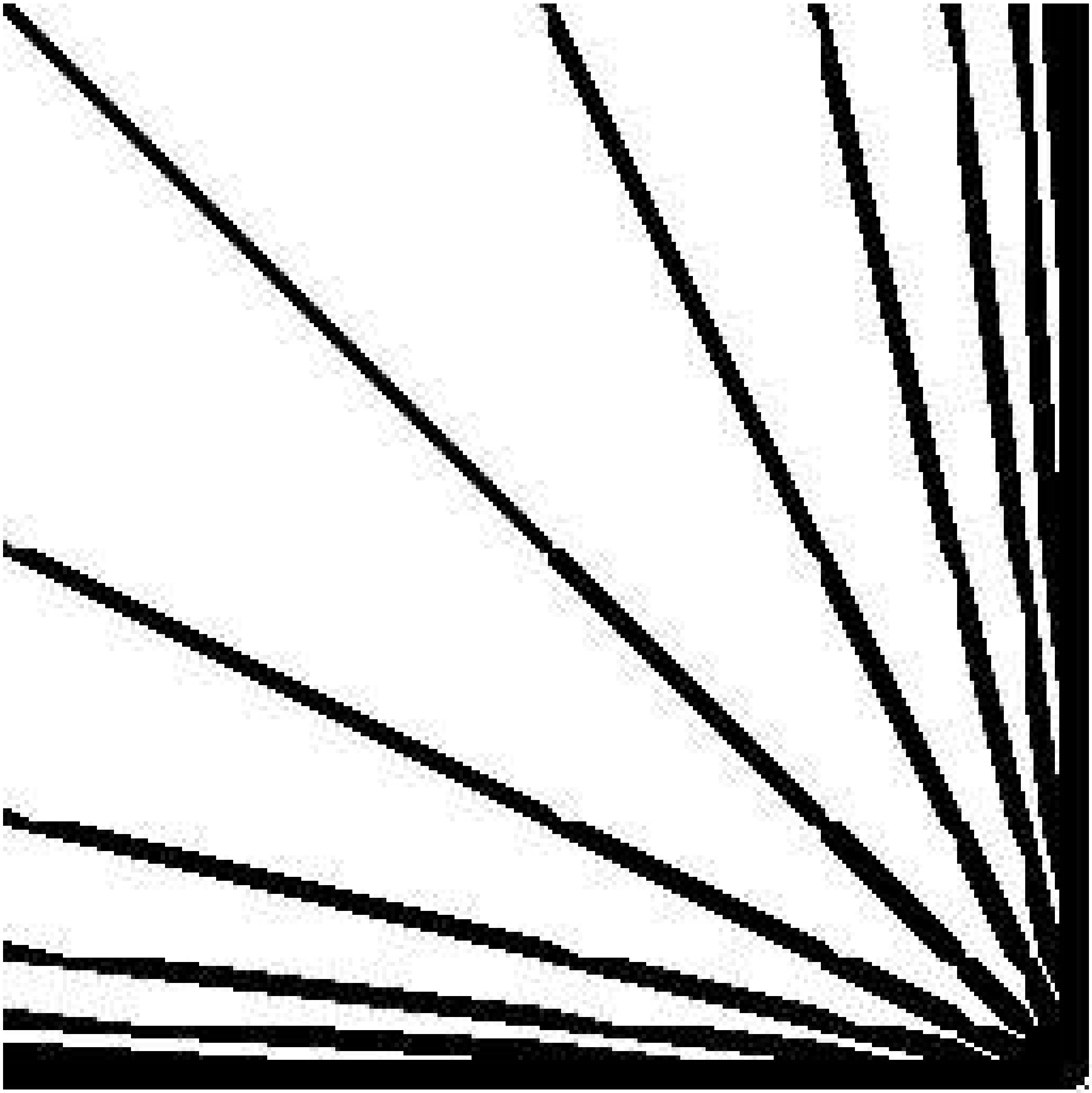

Figure 3: Matrix $\mathbf{B}^{-1}$ computed via iterative algorithm of this Section with diagonal rescaling. Entries with the absolute value greater than 10^{-9} are shown black and the matrix verifies $||\mathbf{B}\mathbf{B}^{-1} - \mathbf{I}||$, $||\mathbf{B}^{-1}\mathbf{B} - \mathbf{I}|| \approx 10^{-9}$.

V Various extensions

V.1 Preconditioning to compensate for variations in a.

In Section III we were assuming that the function a does not change significantly over the interval $(0, 1)$. If a is such that the finite difference scheme in (2.3) is appropriate for solving the two-point boundary value problem, then we rescale (2.3) by multiplying the matrix of the system in (2.3) on both sides by the diagonal matrix

$$\mathbf{P}_a = \operatorname{diag}\left(\frac{1}{\sqrt{a_1}}, \frac{1}{\sqrt{a_2}}, \ldots, \frac{1}{\sqrt{a_N}}\right). \tag{5.1}$$

We obtain instead of (2.3),

$$\frac{a_{i-\frac{1}{2}}}{\sqrt{a_{i-1}a_i}} v_{i-1} - \frac{a_{i-\frac{1}{2}} + a_{i+\frac{1}{2}}}{a_i} v_i + \frac{a_{i+\frac{1}{2}}}{\sqrt{a_i a_{i+1}}} v_{i+1} = h^2 \frac{f_i}{\sqrt{a_i}}, \quad i = 1, \ldots, N, \tag{5.2}$$

where

$$v_i = u_i \sqrt{a_i} \quad i = 1, \ldots, N, \tag{5.3}$$

This corresponds to considering the operator

$$\frac{1}{a(x)} \frac{\partial}{\partial x}\left(a(x)\frac{\partial u}{\partial x}\right) \tag{5.4}$$

instead of the operator $\mathcal{L}$ in (2.1).

If a is sufficiently smooth, then we have

$$\frac{a(x - \frac{1}{2}h)}{\sqrt{a(x-h)a(x)}} = 1 + O(h^2), \tag{5.5}$$

$$\frac{a(x - \frac{1}{2}h) + a(x + \frac{1}{2}h)}{a(x)} = 2 + O(h^2), \tag{5.6}$$

and

$$\frac{a(x + \frac{1}{2}h)}{\sqrt{a(x)a(x+h)}} = 1 + O(h^2). \tag{5.7}$$

Thus, the matrix $\mathbf{L}$ corresponding to (5.2) may be written as

$$\mathbf{L} = \mathbf{L}_0 + h^2\mathbf{R} \tag{5.8}$$

where

$$\mathbf{L}_0 = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -2 \end{pmatrix}. \tag{5.9}$$

We note that in computing entries of the matrix $h^2\mathbf{R}$ via $h^2\mathbf{R} = \mathbf{L} - \mathbf{L}_0$ one should be careful to obtain the sufficient number of significant digits.

Given the operator $\mathbf{L}_0^{-1}$, we have

$$\mathbf{L}^{-1} = \mathbf{L}_0^{-1}(\mathbf{I} + h^2\mathbf{L}_0^{-1}\mathbf{R})^{-1} \tag{5.10}$$

and, therefore, we need to compute $(\mathbf{I} + h^2\mathbf{L}_0^{-1}\mathbf{R})^{-1}$. Again we use the iteration in Section IV and note that if the largest singular value of the operator $\mathbf{T} = -h^2\mathbf{L}_0^{-1}\mathbf{R}$ is less than one, then the iteration in Section IV takes a particular simple form since we have

$$(\mathbf{I} - \mathbf{T})^{-1} = \prod_{j=0}^{\infty}(\mathbf{I} + \mathbf{T}^{2^j}). \tag{5.11}$$

V.2 Additional remarks

Remark 2. It is clear that the generalized inverse $\mathbf{D}^{-1}$ (which on the range of the matrix $\mathbf{D}$ is the matrix $\mathbf{B}^{-1}$ in Figure 3) plays a special role in our approach. Therefore, we may compute this matrix in advance. Considering $\mathbf{A}$ in (3.2) as a perturbation of $\mathbf{D}$, we may use the iteration in Section IV to compute the inverse (similar to that in (5.10)). In fact, the matrix $\mathbf{D}^{-1}$ may be stored and used in the non-standard form [4], which will result in additional efficiency of the computation.

Remark 3. Our approach uses an $O(h^2)$ finite-difference scheme in the wavelet system of coordinates. We may use Richardson extrapolation to improve the accuracy of the solution and, also, of the inverse operator.

VI Wavelet-based adaptive Crank-Nicolson scheme

Let us now consider some of the implications of the fact that the Green's functions of the elliptic two-point boundary value problems are available numerically as sparse matrices. As an example, we consider the implicit Crank-Nicolson scheme to solve

$$u_t = \mathcal{L}u \tag{6.1}$$

with the Dirichlet boundary conditions $u(t,0) = u(t,1) = 0$ and the initial condition $u(0,x) = u_0(x)$. Approximating (6.1) by a system of ordinary differential equations, we obtain

$$\frac{d\mathbf{u}}{dt} = \frac{1}{h_x^2}\mathbf{L}\mathbf{u}, \tag{6.2}$$

where the matrix $\mathbf{L}$ is given in (2.5) and h_x is the step size in x-coordinate. Applying trapezoidal rule to (6.2) (in time), we obtain the Crank-Nicolson method

$$\mathbf{u}^{(n+1)} - \mathbf{u}^{(n)} = \frac{h_t}{2h_x^2}(\mathbf{L}\mathbf{u}^{(n+1)} + \mathbf{L}\mathbf{u}^{(n)}), \tag{6.3}$$

where h_t is the step size in time. We have

$$(\mathbf{I} - \frac{h_t}{2h_x^2}\mathbf{L})\ \mathbf{u}^{(n+1)} = (\mathbf{I} + \frac{h_t}{2h_x^2}\mathbf{L})\ \mathbf{u}^{(n)}, \tag{6.4}$$

a well known implicit scheme for solving (6.1).

In the standard Crank-Nicolson method the inverse matrix is never computed since it is a dense matrix. Instead, one solves a tridiagonal linear system at each time step.

We note that the matrix $(\mathbf{I} - \frac{h_t}{2h_x^2}\mathbf{L})^{-1}$ is sparse in the wavelet system of coordinates and may be computed explicitly by a procedure similar to that described in the previous sections. Thus, by converting (6.4) into the wavelet basis, we obtain an explicit scheme by computing $(\mathbf{I} - \frac{h_t}{2h_x^2}\mathbf{L})^{-1}$. We have

$$\hat{\mathbf{u}}^{(n+1)} = \mathbf{C}\ \hat{\mathbf{u}}^{(n)}, \tag{6.5}$$

where

$$\mathbf{C} = (\mathbf{I} - \frac{h_t}{2h_x^2}\mathbf{W}\mathbf{L}\mathbf{W}^*)^{-1}\ (\mathbf{I} + \frac{h_t}{2h_x^2}\mathbf{W}\mathbf{L}\mathbf{W}^*), \tag{6.6}$$

and

$$\hat{\mathbf{u}} = \mathbf{W}\mathbf{u} \tag{6.7}$$

and $\mathbf{W}$ denotes the matrix of the discrete wavelet transform. The matrix $\mathbf{C}$ is a sparse matrix of the structure similar to that of matrices in Figures 2 and 3.

Such approach applied to the higher order schemes might have serious advantages. Also, the advantages of the conversion from an implicit to an explicit scheme are apparent if the vector $\hat{\mathbf{u}}^{(n)}$ is sparse in the wavelet system of coordinates. In this case the number of operations per time-step is proportional to the number of the significant entries of the vector $\hat{\mathbf{u}}^{(n)}$.

The vectors $\hat{\mathbf{u}}^{(n)}$ become more and more sparse in the wavelet system of coordinates as n (time) increases since the oscillatory modes which correspond to larger eigenvalues of the operator $\mathcal{L}$ decay faster than those corresponding to the smaller eigenvalues. Thus, the speed of the computation via (6.5) is adaptable to the regularity of the solution which is not the case for the standard scheme.

As an example let us consider computing the smallest eigenvalue of the operator $\mathcal{L}$. One of the ways to compute the smallest eigenvalue of the operator $\mathcal{L}$ is to use (6.4) and renormalize the solution after each time-step. Only the modes corresponding to the smallest eigenvalue will remain as a part of the solution after several time-steps. We notice that the eigenvector which corresponds to the smallest eigenvalue is smooth and non-oscillatory. Let us choose the initial condition to be a constant vector. Since such vector is sparse in the wavelet system of coordinates (in fact, it might be represented by just one number) and since we know that the solution of (6.1) remains smooth and non-oscillatory, we conclude that a time-step requires only $O(1)$ operations.

We note that generalizing this approach to the multidimensional case and using the Crank-Nicolson scheme with alternating directions, it appears feasible to obtain a fast method for computing the smallest eigenvalues for the multidimensional elliptic operators.

Acknowledgments.

The research was partially supported by ONR grant N00014-91-J4037 and a grant from Chevron Oil Field Research Company.

References

[1] B. Alpert, G. Beylkin, R. R. Coifman, and V. Rokhlin. Wavelets for the fast solution of second-kind integral equations. *SIAM Journal of Scientific and Statistical Computing*, 14(1):159–174, 1993. Technical report, Department of Computer Science, Yale University, New Haven, CT, 1990.

[2] G. Beylkin. On the representation of operators in bases of compactly supported wavelets. *SIAM J. Numer. Anal.*, 29(6):1716–1740, 1992.

[3] G. Beylkin, R. R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms II. in progress.

[4] G. Beylkin, R. R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Comm. Pure and Appl. Math.*, 44:141–183, 1991. Yale University Technical Report YALEU/DCS/RR-696, August 1989.

[5] G. Beylkin, R. R. Coifman, and V. Rokhlin. Wavelets in Numerical Analysis. In *Wavelets and Their Applications*, pages 181–210. Jones and Bartlett, 1992.

[6] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Mathematics of Computation*, 55(191):1–22, 1990.

[7] W. L. Briggs. *A Multigrid Tutorial.* SIAM, Philadelphia, Pennsylvania, 1987.

[8] A. Cohen, I. Daubechies, B. Jawerth, and P. Vial. Multiresolution analysis, wavelets and fast algorithms on an interval. *Comptes Rendus Acad. Sc. Paris*, 1992.

[9] L. Greengard and V. Rokhlin. On the Numerical Solution of Two-Point Boundary Value Problems. Yale University Technical Report, YALEU/DCS/RR-692 (1989).

[10] S. Jaffard. Wavelet methods for fast resolution of elliptic problems. *SIAM Journal on Numerical Analysis*, 29(4):965–986, 1992.

[11] G. Schulz. Iterative Berechnung der reziproken Matrix. *Z. Angew. Math. Mech.*, 13:57–59, 1933.