Multiscale Inversion of Elliptic Operators

Amir Averbuch, Gregory Beylkin,
Ronald Coifman, and Moshe Israeli

Abstract. A fast adaptive algorithm for the solution of elliptic partial differential equations is presented. It is applied here to the Poisson equation with periodic boundary conditions. The extension to more complicated equations and boundary conditions is outlined.

The purpose is to develop algorithms requiring a number of operations proportional to the number of significant coefficients in the representation of the r.h.s. of the equation. This number is related to the specified accuracy, but independent of the resolution. The wavelet decomposition and the conjugate gradient iteration serve as the basic elements of the present approach.

The main difficulty in solving such equations stems from the inherently large condition number of the matrix representing the linear system that result from the discretization. However, it is known that periodized differential operators have an effective diagonal preconditioner in the wavelet system of coordinates. The condition number of the preconditioned matrix is $O(1)$ and, thus, depends only weakly on the size of the linear system.

The nonstandard form (nsf) is preferable in multiple dimensions since it requires $O(1)$ elements to represent the operator on all scales. Unfortunately, the preconditioned nsf turns out to be dense. This obstacle can be avoided if in the process of solving the linear system, the preconditioner is applied separately before and after the operator (to maintain sparsity).

A constrained version of the preconditioned conjugate gradient algorithm is developed in wavelet coordinates. Only those entries of the conjugate directions which are in the set of significant indices are used.

The combination of the above-mentioned elements yields an algorithm where the number of operations at each iteration is proportional to the number of elements. At the same time, the number of iterations is bounded by a constant.

§1 Introduction

In this paper we describe the components of a fast adaptive method for solving elliptic equations with periodic boundary conditions as well as de-

Signal and Image Representation in Combined Spaces
Y. Y. Zeevi and R. R. Coifman (Eds.), pp. 341–359.

ISBN 0-12-777830-6

velop a framework for solving problems with general boundary conditions. Let us consider the partial differential equation

$$\mathcal{L}u = f \quad x \in \mathbf{D} \subset \mathbf{R}^{\mathrm{d}}, \tag{1.1}$$

with the boundary condition

$$\mathcal{B}u|_{\partial\mathbf{D}} = g, \tag{1.2}$$

where $\mathcal{L}$ is an elliptic operator,

$$\mathcal{L}u = -\sum_{i,j=1,\ldots,d} (a_{ij}(x)\ u_{x_i})_{x_j} + b(x)\ u, \tag{1.3}$$

and $\mathcal{B}$ is the boundary operator,

$$\mathcal{B}u = \alpha u + \beta \frac{\partial u}{\partial N}. \tag{1.4}$$

We assume that the boundary $\partial\mathbf{D}$ is "complicated." As a practical matter we are interested in dimensions $\mathrm{d} = 1, 2, 3$ though our considerations are valid in higher dimensions as well.

We adopt a classical approach to this problem which, until now, was not practical from the numerical point of view. We consider the following steps for solving the problem in (1.1) and (1.2):

1. We generate a function f_{ext}, a smooth extension of f outside the domain $\mathbf{D}$, such that f_{ext} is compactly supported in a rectangular box $\mathbf{B}$, $\mathbf{D} \subset \mathbf{B} \subset \mathbf{R}^{\mathrm{d}}$, and $f = f_{\mathrm{ext}}$ for $x \in \mathbf{D}$.

2. We solve the problem

$$\mathcal{L}u = f_{\mathrm{ext}} \quad x \in \mathbf{B} \tag{1.5}$$

with periodic boundary conditions.

3. Given the solution u_{ext} of (1.5), we look for the solution of (1.1) as

$$u = u_{\mathrm{ext}} + v \tag{1.6}$$

which yields the homogeneous equation

$$\mathcal{L}v = 0 \quad x \in \mathbf{D} \tag{1.7}$$

with the boundary condition

$$\mathcal{B}v|_{\partial\mathbf{D}} = g - \mathcal{B}u_{\mathrm{ext}}|_{\partial\mathbf{D}}, \tag{1.8}$$

which we solve by boundary integral methods.

In order to realize the preceding steps we need to develop:

1. An algorithm for extending the function f outside the domain $\mathbf{D}$.
2. An efficient method for solving (1.5).
3. An efficient method for solving the boundary integral equation derived from (1.7) and (1.8).
4. An effective algorithm for generating a representation of the solution v of (1.7) and (1.8) once the boundary integral equation is solved.

It is only recently that fast methods for solving the problem (1.7) and (1.8) using boundary integral equations have been developed, namely, Fast Multipole Method (FMM) in [9, 8, 7] and BCR algorithm in [6]. It is our understanding that 1 and 4 are solvable and the main difficulty resides with 2. We will address the algorithmic issues of 1, 3, and 4 elsewhere; and in this paper, we will concentrate on constructing an algorithm for solving (1.5).

§2 Approach

Our goal in solving (1.5) is to develop an adaptive algorithm where the number of operations will be proportional to the number of significant coefficients in the representation of f_{ext}. The usual solution procedure by current numerical methods requires discretization of the r.h.s. and of the solution in terms of a grid or a basis, such that the representations will resolve all features of interest. This might require a large number of grid points or elements not only near the singularities of the functions involved but also in the regions of smooth behavior, thus requiring proportionally large number of operations. Current adaptive procedures (for example, adaptive grids or irregular elements) are cumbersome, especially in higher dimensions and imply a considerable overhead both on the algorithmical and programming levels.

Our approach is based on using properties of representations of functions in wavelet bases and allows us to obtain a simple adaptive algorithm.

Let us illustrate it by considering Poisson's equation

$$\Delta u = f \quad x \in \mathbf{B} \tag{2.1}$$

with periodic boundary conditions where (with a slight abuse of notation) we used f instead of f_{ext} to denote the source term. The source term f may be discontinuous in the domain $\mathbf{B}$.

Let us consider an MRA of $\mathbf{L}^2(\mathbf{R}^d)$,

$$\cdots \subset \mathbf{V}_2 \subset \mathbf{V}_1 \subset \mathbf{V}_0 \subset \mathbf{V}_{-1} \subset \mathbf{V}_{-2} \subset \cdots \tag{2.2}$$

where $\mathbf{V}_j$ is a subspace of an MRA spanned by translations of the scaling function,

$$\phi_{j,\mathbf{k}}(x) = 2^{-jd/2}\phi(2^{-j}x_1 - k_1)\,\phi(2^{-j}x_2 - k_2)\ldots\phi(2^{-j}x_{\mathrm{d}} - k_{\mathrm{d}}), \quad (2.3)$$

where $x = (x_1, \ldots, x_{\mathrm{d}})$ and $k = (k_1, \ldots, k_{\mathrm{d}}) \in \mathbf{Z}^{\mathrm{d}}$. The function ϕ is the scaling function of MRA of $\mathbf{L}^2(\mathbf{R})$.

Let us define the subspaces $\mathbf{W}_j$ as orthogonal complements of $\mathbf{V}_j$ in $\mathbf{V}_{j-1}$,

$$\mathbf{V}_{j-1} = \mathbf{V}_j \oplus \mathbf{W}_j, \quad (2.4)$$

and represent the space $\mathbf{L}^2(\mathbf{R}^{\mathrm{d}})$ as a direct sum

$$\mathbf{L}^2(\mathbf{R}^{\mathrm{d}}) = \mathbf{V}_n \bigoplus_{j \le n} \mathbf{W}_j \quad (2.5)$$

where $\mathbf{V}_n$ is the subspace corresponding to the coarsest scale n. Let us define wavelets $\psi^{\sigma}_{j,\mathbf{k}}(x)$ which form an orthonormal basis of the subspaces $\mathbf{W}_j$, $j \le n$. We consider

$$\varphi^{\rho}(x) = \begin{cases} \psi(x) & \text{for } \rho = 1, \\ \phi(x) & \text{for } \rho = 0, \end{cases} \quad (2.6)$$

where ϕ and ψ are the scaling function and the wavelet of the MRA of $\mathbf{L}^2(\mathbf{R})$. The wavelet ψ typically has several vanishing moments. Then it follows that

$$\psi^{\sigma}_{j,\mathbf{k}}(x) = 2^{-jd/2}\varphi^{\rho_1}(2^{-j}x_1 - k_1)\,\varphi^{\rho_2}(2^{-j}x_2 - k_2)\ldots\varphi^{\rho_{\mathrm{d}}}(2^{-j}x_{\mathrm{d}} - k_{\mathrm{d}}), \quad (2.7)$$

where the multi-index $\sigma = (\rho_1, \rho_2, \ldots, \rho_{\mathrm{d}})$, $\sigma \neq \mathbf{0}$ and ρ_k, $k = 1, 2, \ldots, \mathrm{d}$, take values of either one or zero.

We also define projection operators on the subspaces $\mathbf{V}_j$, $j \le n$,

$$P_j : \mathbf{L}^2(\mathbf{R}) \to \mathbf{V}_j, \quad (2.8)$$

as follows

$$(P_j f)(x) = \sum_{\mathbf{k}} \langle f, \phi_{j,\mathbf{k}} \rangle \phi_{j,\mathbf{k}}(x) \quad (2.9)$$

while on the subspaces $\mathbf{W}_j$, $j \le n$,

$$Q_j : \mathbf{L}^2(\mathbf{R}) \to \mathbf{W}_j, \quad (2.10)$$

we define

$$Q_j = P_{j-1} - P_j = \sum_{\mathbf{k}} \sum_{\sigma} \langle f, \psi^{\sigma}_{j,\mathbf{k}} \rangle \psi^{\sigma}_{j,\mathbf{k}}(x), \quad (2.11)$$

where

$$\langle f, g\rangle = \int_{-\infty}^{\infty} f(x)\, g(x) dx. \tag{2.12}$$

The sum over σ is finite and the number of terms is $2^{\mathrm{d}} - 1$ for each $\mathbf{k}$.

Let us represent the source term f and the solution u in (2.1) in the wavelet basis,

$$f(x) = \sum_{j\le n}\sum_{\mathbf{k}}\sum_{\sigma} f^{\sigma}_{j,\mathbf{k}}\psi^{\sigma}_{j,\mathbf{k}}(x) + \sum_{\mathbf{k}} s^{f}_{n,\mathbf{k}}\phi_{n,\mathbf{k}}(x), \tag{2.13}$$

$$u(x) = \sum_{j\le n}\sum_{\mathbf{k}}\sum_{\sigma} u^{\sigma}_{j,\mathbf{k}}\psi^{\sigma}_{j,\mathbf{k}}(x) + \sum_{\mathbf{k}} s^{u}_{n,\mathbf{k}}\phi_{n,\mathbf{k}}(x), \tag{2.14}$$

where

$$f^{\sigma}_{j,\mathbf{k}} = \langle f, \psi^{\sigma}_{j,\mathbf{k}}\rangle,\ \ u^{\sigma}_{j,\mathbf{k}} = \langle u, \psi^{\sigma}_{j,\mathbf{k}}\rangle,\ \ s^{f}_{n,\mathbf{k}} = \langle f, \phi_{n,\mathbf{k}}\rangle,\ \text{and}\ \ s^{u}_{n,\mathbf{k}} = \langle u, \phi_{n,\mathbf{k}}\rangle. \tag{2.15}$$

We now define the ϵ-accuracy subspace for f to be the subspace on which f may be represented with accuracy ϵ, namely,

$$M^{\epsilon}_{\mathrm{r.h.s}} = \mathbf{V}_n \bigcup \{\mathrm{span}\{\psi^{\sigma}_{j,\mathbf{k}}\} \,|\, (j,\mathbf{k},\sigma)\ :\ |f^{\sigma}_{j,\mathbf{k}}| > \epsilon\}, \tag{2.16}$$

and observe that the ϵ-accuracy subspace for the solution

$$M^{\epsilon}_{\mathrm{sol}} = \mathbf{V}_n \bigcup \{\mathrm{span}\{\psi^{\sigma}_{j,\mathbf{k}}\} \,|\, (j,\mathbf{k},\sigma)\ :\ |u^{\sigma}_{j,\mathbf{k}}| > \epsilon\} \tag{2.17}$$

may be estimated given $M^{\epsilon}_{\mathrm{r.h.s}}$. It may be verified that

Proposition 1. *Let*

$$u(x) = \sum_{j}\sum_{\mathbf{k}}\sum_{\sigma} u^{\sigma}_{j,\mathbf{k}}\psi^{\sigma}_{j,\mathbf{k}}(x) + \text{constant} \tag{2.18}$$

be the solution of

$$\Delta u = \psi^{\sigma'}_{j',\mathbf{k}'} \quad x \in \mathbf{B} \tag{2.19}$$

with periodic boundary conditions. For any $\epsilon > 0$ there exist $\lambda > 0$ and $\mu > 0$ such that all indices $(j,\mathbf{k},\sigma)$ corresponding to the significant coefficients of the solution, $|u^{\sigma}_{j,\mathbf{k}}| \ge \epsilon$, satisfy $|\mathbf{k} - \mathbf{k}'| \le \lambda$ and $|j - j'| \le \mu$.

The size of $\mu > 0$ and $\lambda > 0$ depends on the particular choice of basis and, of course, on ϵ. Given $M^{\epsilon}_{\mathrm{r.h.s}}$, we may construct the set $M_{\lambda,\mu}$ as a (λ,μ)-neighborhood of $M^{\epsilon}_{\mathrm{r.h.s}}$. According to Proposition 1, $M^{\epsilon}_{\mathrm{sol}} \subset M_{\lambda,\mu}$. We note that estimating the subspace amounts to constructing a mask which contains indices of significant coefficients.

Instead of estimating $M_{\lambda,\mu}$ directly, we may use an iterative approach. For example, solving directly on $M^{\epsilon}_{\text{r.h.s}}$ produces a solution $\tilde{u}$ with accuracy $\tilde{\epsilon} > \epsilon$. Applying the Laplacian to $\tilde{u}$, we generate $\tilde{f}$. Estimating the ϵ-accuracy subspace for $\tilde{f}$, we may use it to continue the iteration to improve the accuracy of the solution. In other words, the mask for $M^{\epsilon}_{\text{sol}}$ may be generated iteratively.

There are three main features in our approach to solve (2.1):

1. **Estimation of the ϵ-accuracy subspace for the solution.** Our first step is to explicitly estimate the subspace $M^{\epsilon}_{\text{sol}}$ given $M^{\epsilon}_{\text{r.h.s}}$. For elliptic operators, the dimension of $M^{\epsilon}_{\text{sol}}$ is proportional to that of $M^{\epsilon}_{\text{r.h.s}}$.

2. **Preconditioning of the operator.** A simple diagonal preconditioner is available for periodized differential operators in the wavelet bases [4, 5] which yields a condition number of $O(1)$. We will show below how to construct simple preconditioners in wavelet bases for more general operators.

3. **Constrained iterative solver.** We use the preconditioned Conjugate Gradient (CG) method which we constrain to the subspace estimated at Step 1, *e.g.* $M_{\lambda,\mu}$. The CG method requires only a constant number of iterations due to preconditioning at Step 2, whereas the cost of each iteration is proportional to the dimension of $M^{\epsilon}_{\text{sol}}$ provided we succeed to limit the number of operations required for the application of the operator (matrix) in the CG method (see below).

Steps 1–3 constitute an adaptive algorithm for solving Poisson's equation.

§3 Outline of the algorithm

Let us consider the projection L_0 of the periodized operator Δ on $\mathbf{V}_0$, the finest scale under consideration,

$$L_0 = P_0 \Delta P_0, \tag{3.1}$$

and L_s and L_{ns}, its standard (s-form) and non-standard forms (ns-form) [6].

One of the difficulties in solving (1.5) stems from the inherently large condition number of the linear system resulting from the discretization of (1.5). As was shown in [4] and [5], using a diagonal preconditioner in the wavelet system of coordinates yields a linear system with the condition number typically less than 10, independently of its size. Let $\mathcal{P}$ denote such a diagonal preconditioner.

In [5] the s-form is used to solve the two-point boundary-value problem. Alternatively, we may use the ns-form. Some care is required at this point since the preconditioned ns-form is dense unlike the s-form, which remains sparse. Thus, in the process of solving the linear system, it is necessary to apply the preconditioner and the ns-form sequentially in order to maintain sparsity. The ns-form is preferable in multiple dimensions since, for example, differential operators require $O(1)$ elements for representation on all scales (see *e.g.* [4]).

We develop a constrained (see below) preconditioned CG algorithm for solving (1.5) in an adaptive manner. Both the s-form and the ns-form may be used for this purpose but it appears that using the ns-form is more efficient, especially if compactly supported wavelets are used and high accuracy is required.

Let us consider (1.5) in the wavelet system of coordinates

$$L_{ns} u_w = f_w, \tag{3.2}$$

where f_w and u_w are representations of f and u in the wavelet system of coordinates. This equation should be understood to include the rules for applying the ns-form (see [6]).

Let us rewrite (3.2) using the preconditioner $\mathcal{P}$ as

$$\mathcal{P} L_{ns} \mathcal{P} v = \mathcal{P} f_w, \tag{3.3}$$

where $\mathcal{P} v = u$. For example, for the second derivative the preconditioner $\mathcal{P}$ is as follows:

$$\mathcal{P}_{il} = \delta_{il} 2^j \tag{3.4}$$

where $1 \le j \le n$ is chosen depending on i, l so that $n - n/2^{j-1} + 1 \le i, l \le n - n/2^j$, and $\mathcal{P}_{nn} = 2^n$.

The periodized operator Δ has the null space of dimension one which contains constants. If we use the full decomposition (over all n scales) in the construction of the ns-form, then the null space coincides with the subspace $\mathbf{V}_n$, which in this case has dimension one (see [5]). This allows us to solve (3.3) on the range of the operator,

$$\bigoplus_{1 \le j \le n} \mathbf{W}_j \tag{3.5}$$

where the linear system (3.3) is well conditioned.

Remark. Operators with variable coefficients. As in the case of the Laplacian, the ϵ-accuracy subspace for the solution may be estimated using corresponding subspaces for the r.h.s and the coefficients. Essentially, we consider the union of such subspaces as a starting point for constructing $M_{\lambda,\mu}$. These estimates may be revised in the process of iteration.

§4 Preconditioner for the operator $-\Delta + \text{Const}$

An "efficient" preconditioner is an essential element in the present approach. In a more restricted sense, "efficient" means insensitive to the size of the problem.

Let us demonstrate how to construct a diagonal preconditioner for the sum of operators $-\Delta + \text{Const}$ in the wavelet bases. We observe that if A and B are diagonal operators with diagonal entries a_i and b_i, then the diagonal operator with entries $1/(a_i + b_i)$ (provided $a_i + b_i \neq 0$) is an ideal preconditioner.

In our case, the operator $-\Delta$ is not diagonal but we know a good diagonal preconditioner for it in wavelet bases (3.4). Let us use this preconditioner instead of $-\Delta$ for the purpose of constructing a preconditioner for $-\Delta + \text{Const}$, where $\text{Const} > 0$. We note that in wavelet bases the identity operator remains unchanged. We restrict $\text{Const} \cdot I$, where I is the identity operator, to the subspace

$$\bigoplus_{1 \leq j \leq n} \mathbf{W}_j \tag{4.1}$$

and construct a preconditioner on this subspace.

We obtain

$$\mathcal{P}_{il} = \frac{\delta_{il}}{\sqrt{2^{-2j} + \text{Const}}} \tag{4.2}$$

where $1 \leq j \leq n$ is chosen depending on i, l so that $n - n/2^{j-1} + 1 \leq i, l \leq n - n/2^j$, and $\mathcal{P}_{nn} = 1/\sqrt{2^{-2n} + \text{Const}}$. The square root appears in (4.2) in order to symmetrize the application of the preconditioner as shown in the previous section. In Table 1 we illustrate the effect of preconditioning of the operator $-\frac{d^2}{dx^2} + \text{Const}$ by the diagonal matrix (4.2).

Remark. If we consider an operator $-\Delta + V$, where V is an operator of multiplication by a function $V(x)$, a similar construction may be obtained on fine scales. On fine scales where the function $V(x)$ does not change significantly over the support of wavelets, we may consider the diagonal operator V^{diag},

$$\left(V^{\text{diag}} \psi^{\sigma}_{j,\mathbf{k}}\right) = V(x_{j,\mathbf{k}}) \psi^{\sigma}_{j,\mathbf{k}}, \tag{4.3}$$

where $x_{j,\mathbf{k}}$ is a point within the support of the wavelet $\psi^{\sigma}_{j,\mathbf{k}}$. Using V^{diag} instead of V, we obtain the preconditioner in a manner outlined above. We will address the problem of constructing preconditioners for operators of the form $-\Delta + V$ elsewhere.

Const	κ	κ_p
$7.1{\cdot}10^{-0}$	$2.4{\cdot}10^{0}$	2.1
$7.1{\cdot}10^{-1}$	$1.5{\cdot}10^{1}$	6.3
$7.1{\cdot}10^{-2}$	$1.4{\cdot}10^{2}$	9.4
$7.1{\cdot}10^{-3}$	$1.3{\cdot}10^{3}$	9.5
$7.1{\cdot}10^{-4}$	$6.7{\cdot}10^{3}$	7.5
$7.1{\cdot}10^{-5}$	$1.5{\cdot}10^{4}$	5.0

Table 1. Condition numbers κ before and κ_p after preconditioning of the operator $-\frac{d^2}{dx^2} + \text{Const}$ in the basis of Daubechies' wavelets with six vanishing moments. There are 8 scales and the matrix size is 256×256.

§5 Initial numerical experiments

Several numerical experiments have been performed in one dimension to verify the main points of the procedure outlined in this paper. A two- and three-dimensional version of an adaptive Poisson solver is being implemented and the results will be reported elsewhere.

In our one-dimensional experiment we set out to verify that using the CG method we can: (1) maintain the sparsity of the conjugate directions (and all other auxiliary vectors) used in the method, this is essential for establishing the sought for operation count, and: (2) solve the problem with a number of iterations that do not depend on the size of the problem.

In order to be able to make a convergence test independently of discretization errors, we first construct the solution and then apply the operator to obtain the right-hand side of the equation. We then solve the equation with a given accuracy ϵ and compare the results with the exact solution to verify the performance of the algorithm.

As the solution we chose initially the vector

$$u_i = \sin(2\pi i)/N, \tag{5.1}$$

where $i = 0, \ldots, N-1$, the size $N = 2^n$ represents the size of the vector in the ordinary system of coordinate and n indicates the maximum number of scales in the problem. A more interesting problem is given by the solution vector

$$u_i = \tanh(\omega \sin(2\pi i)/N), \tag{5.2}$$

where ω is a parameter. By choosing ω we control the steepness of the slope of the function and, thus, the necessary number of scales to represent this vector in a wavelet basis up to some accuracy ϵ.

The initial experiments with very smooth solution vectors demonstrated a severe problem in the algorithm which is exhibited in Figures 1–5. For example, consider the wavelet coefficients of the solution vector for $\omega = 0.8, N = 1024$, analyzed with Daubechies' basis with 10 vanishing moments. Obviously, both the solution and the initial approximation have a very small number of significant coefficients concentrated in the coarse scales. (The wavelet coefficients are usually exhibited by plotting the magnitude against the number of the coefficient in the following sequence: the fine-scale wavelet coefficients numbered (in the present case) 1 to 512 appear first in the plot of the wavelet decomposition, immediately followed by the coarser scale with 256 coefficients, 513 to 768 followed by the third level with 128, from 769 to 896 then the levels containing the next 64, 32, 16, 8, 4, 2, 1 coefficients respectively, bringing us to 1023; the last coefficient, 1024, is the mean.)

Figure 1 shows only the largest coefficients (1010–1024), other coefficients are not visible on this scale. The other four figures show the behavior of the 64 coefficients in level four (897–960). The first iteration result, ans_1_iter, vanishes for this level, iteration 3 shows a significant growth, iteration five is even larger, iteration seven drops by an order of magnitude, iteration nine has again vanishingly small coefficients on this scale. We can summarize this behavior and our solution in the following way:

Constrained iterative solver. In order to solve (3.3) we apply the Conjugate Gradient method constrained to the subspace $M_{\lambda,\mu}$. Without such constraint the conjugate directions become "dense" at early stages of the iteration only to become small outside the subspace $M_{\lambda,\mu}$ later. Thus, constraining the solution to a subspace is critical for an adaptive algorithm.

In applying this "constrained" conjugate gradient method in the wavelet coordinates, we generate only those entries of conjugate directions which are in the set of significant indices which define the subspace $M_{\lambda,\mu}$. This yields an algorithm where the number of operations at each iteration is proportional to the number of elements of $M_{\lambda,\mu}$. The number of iterations is $O(1)$ and, thus, the overall number of operations is proportional to the number of significant coefficients of f, *i.e.* the dimension of $M^{\epsilon}_{\text{r.h.s}}$.

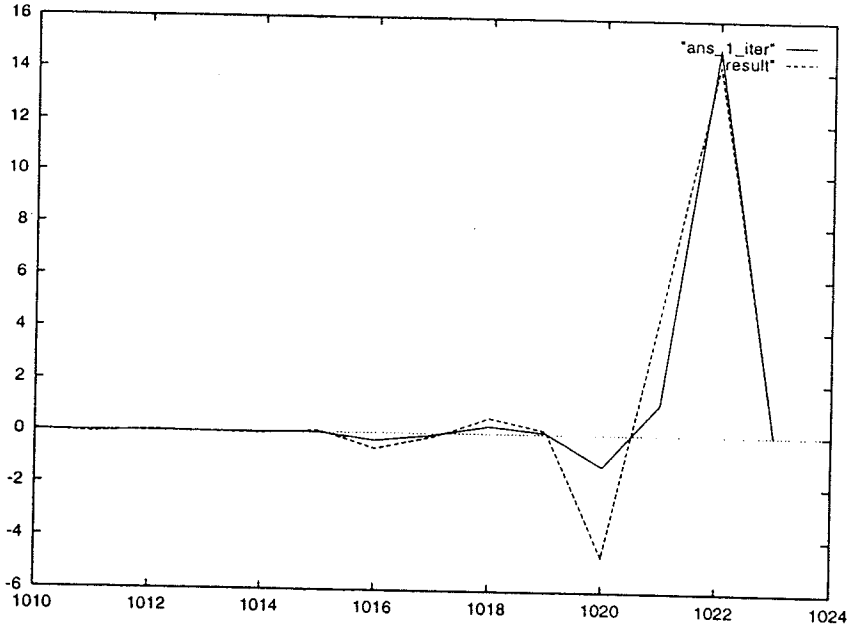

Figure 1. Compare the final result with the result after 1 iteration for $\omega = 0.8$, 1024 points, Daubechies 20, no skip.

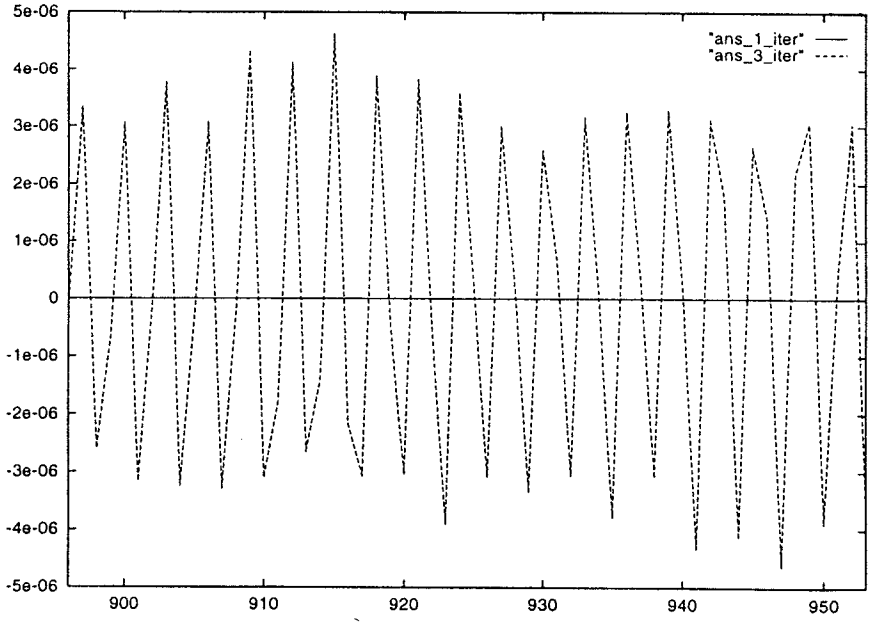

Figure 2. Compare the results after 1 and 3 iterations for $\omega = 0.8$, 1024 points, Daubechies 20, no skip.

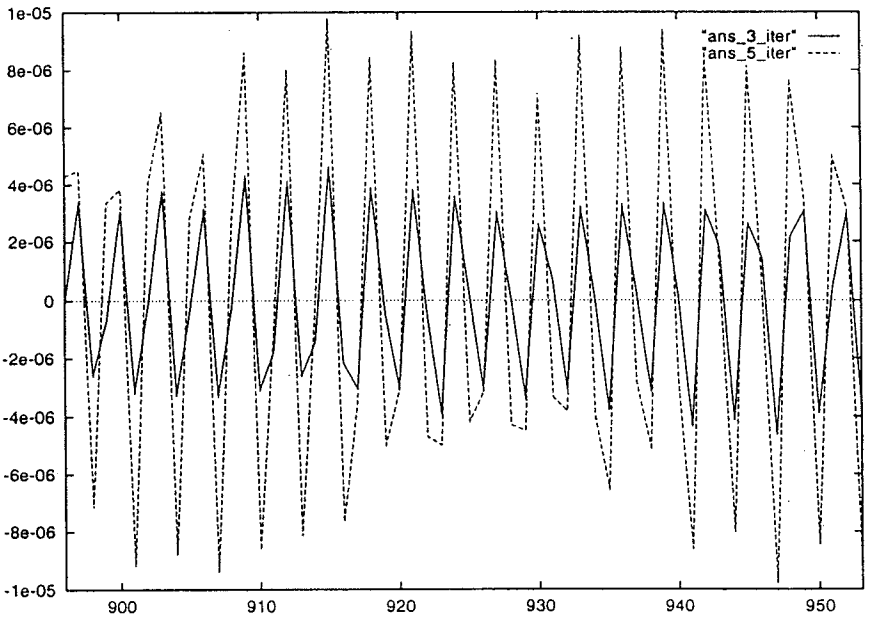

Figure 3. Compare the results after 3 and 5 iterations for $\omega = 0.8$, 1024 points, Daubechies 20, no skip.

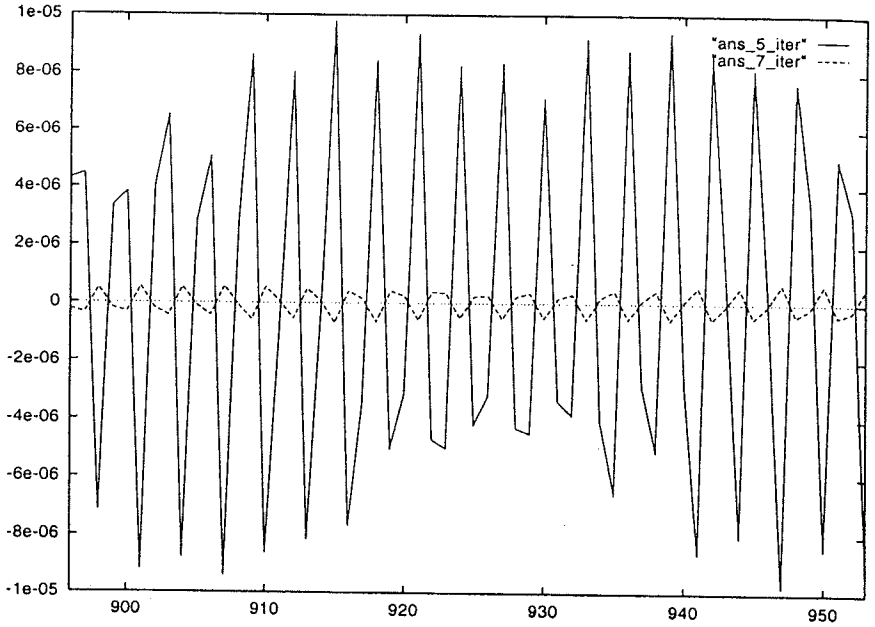

Figure 4. Compare the results after 5 and 7 iterations for $\omega = 0.8$; 1024 points, Daubechies 20, no skip.

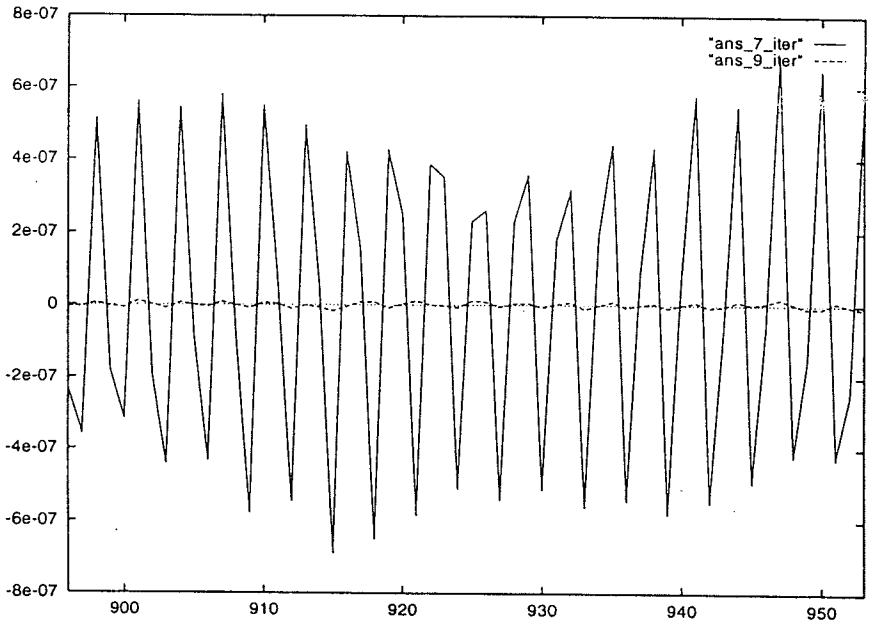

Figure 5. Compare the results after 7 and 9 iterations for $\omega = 0.8$, 1024 points, Daubechies 20, no skip.

# scales to skip	# iterations	$\|\ \|_\infty$	$\|\ \|_2$
0	36	10^{-13}	10^{-14}
1	36	0.45×10^{-9}	0.32×10^{-9}
2	34	0.96×10^{-8}	0.52×10^{-8}
3	32	0.17×10^{-6}	0.83×10^{-7}
4	27	0.30×10^{-5}	0.13×10^{-5}
5	17	0.48×10^{-4}	0.21×10^{-4}
6	9	0.80×10^{-3}	0.34×10^{-3}

Table 2. Coiflet 12, $2^{10} \times 2^{10}$ matrix.

§6 Further numerical experiments

In the second set of one-dimensional experiments we verify that using the "constrained CG" method we can maintain the sparsity of the conjugate directions. At the same time we must check that the convergence rate of the method was not damaged by the restriction of the search directions to a small subspace. In general we cannot expect success, however in the present case there are a number of special "mitigating circumstances."

For the smooth problems considered in the previous set of experiments the coefficients are concentrated in the coarse scales, so a very effective "mask" is obtained by setting to zero a number of fine scales. We demonstrate the results in Tables 2, 3, and 4 where we show the number of iterations of the constrained CG method as a function of the number of scales skipped (where 0 means all levels are used, 1 means the fine level was skipped, 2 means the two finest levels were skipped, etc.) and the accuracy achieved as measured against the exact solution. In the present approach skipping some levels means that the answer itself is represented with a small number of scales and the accuracy will be affected adversely. A more elaborate mask is used in our later experiments [1].

We observe that the number of iterations did not increase at all as a consequence of the constraint. Furthermore, in all cases checked it was a monotonically decreasing function of the number of scales skipped. We can conclude that at least for the case of smooth-forcing functions and constant coefficient operators the CPCG (constrained preconditioned conjugate gradient approach) results in a number of operations proportional to the number of significant coefficients of the forcing. More elaborate masks appropriate for non-smooth functions are explored in the next paper in this sequence [1].

# scales to skip	# iterations	$\|\ \|_\infty$	$\|\ \|_2$
0	29	10^{-14}	10^{-15}
1	28	0.20×10^{-13}	0.13×10^{-13}
2	28	0.17×10^{-11}	0.81×10^{-12}
3	26	0.97×10^{-10}	0.52×10^{-10}
4	23	0.64×10^{-8}	0.33×10^{-8}
5	16	0.40×10^{-6}	0.20×10^{-6}
6	9	0.25×10^{-4}	0.12×10^{-4}
7	5	0.14×10^{-2}	0.77×10^{-3}
8	3	0.65×10^{-1}	0.38×10^{-1}

Table 3. Daubechies 12, $2^{10} \times 2^{10}$ matrix.

# scales to skip	# iterations	$\|\ \|_\infty$	$\|\ \|_2$
0	22	10^{-14}	10^{-14}
1	22	10^{-14}	10^{-14}
2	22	10^{-14}	10^{-14}
3	21	10^{-14}	10^{-14}
4	19	0.36×10^{-13}	0.17×10^{-13}
5	14	0.34×10^{-10}	0.17×10^{-10}
6	9	0.33×10^{-7}	0.17×10^{-7}
7	5	0.27×10^{-4}	0.15×10^{-4}
8	3	0.13×10^{-1}	0.77×10^{-2}

Table 4. Daubechies 20, $2^{10} \times 2^{10}$ matrix.

Acknowledgment. The research of A. Averbuch, R. Coifman, and M. Israeli was supported by U.S-Israel Binational Science Foundation Grant # 92-00269/1. The research of M. Israeli was supported also by the Fund for the Promotion of Research at the Technion. The research of G. Beylkin was partially supported by ARPA grant F49620-93-1-0474 and ONR grant N00014-91-J4037.

References

[1] Averbuch, A., G. Beylkin, R. Coifman, and M. Israeli, Fast adaptive algorithm for elliptic equations with periodic boundary conditions, to appear.

[2] Averbuch, A., M. Israeli, and L. Vozovoi, Spectral multidomain technique with local Fourier basis, *J. Scientific Computing* **8**(2) (1993), 135–149.

[3] Aharoni, G., A. Averbuch, R. Coifman, and M. Israeli, Local cosine transform – A method for the reduction of the blocking effect in JPEG, *J. Math. Imaging and Vision*, Special Issue on Wavelets, **3** (1993), 7–38.

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

[5] Beylkin, G., On wavelet-based algorithms for solving differential equations, in *Wavelets: Mathematics and Applications*, J. J. Benedetto and M. W. Frazier (eds.), CRC Press, 1994, 449–466.

[6] Beylkin, G., R. R. Coifman, and V. Rokhlin, Fast wavelet transforms and numerical algorithms I, *Comm. Pure and Appl. Math.*, **44** (1993), 141–183. Yale University Technical Report YALEU/DCS/RR-696, August 1989.

[7] Carrier, J., L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations, *SIAM J. Scientific and Statistical Computing* **9**(4), 1988. Yale University Technical Report, YALEU/DCS/RR-496 (1986).

[8] Greengard, L. and V. Rokhlin, A fast algorithm for particle simulations, *J. Comp. Phys.* **73**(1) (1987), 325–348.

[9] Rokhlin, V., Rapid solution of integral equations of classical potential theory, *J. Comp. Phys.* **60**(2) (1985).

Amir Averbuch
School of Mathematical Sciences
Tel Aviv University
Tel Aviv 69978, Israel
amir@math.tau.ac.il

Gregory Beylkin
Program in Applied Mathematics
University of Colorado at Boulder
Boulder, CO 80309-0526
beylkin@julia.colorado.edu

Ronald Coifman
Department of Mathematics
P.O. Box 208283
Yale University
New Haven, CT 06520-8283
coifman@jules.math.yale.edu

Moshe Israeli
Faculty of Computer Science
Technion-Israel Institute of Technology
Haifa 32000, Israel
israeli@cs.technion.ac.il