12

On wavelet-based algorithms for solving differential equations

G. Beylkin

ABSTRACT We describe an order N method for computing the

12.1 Introduction

The role of the orthonormal wavelet bases in solving integral equations has been studied in [BCoR1], where it was observed that wide classes of operators have sparse representations in the wavelet bases thus permitting a number of fast

We note that considering a banded approximation directly for the kernel satisfying (12.2) does not lead to a satisfactory numerical approximation. The method of [BCoR1] uses the smoothness of the matrix away from the diagonal to in 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 [redacted]

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., [Bri]) or multilevel (hierarchical) methods or the very simple and elegant [redacted]

$O(N)$ operations. 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 [redacted] time. We approach the multidimensional problems using the alternating directions technique, which is modified since we are able to numerically construct the Green's functions of the two-point boundary value problems. We note that our use of the diagonal preconditioning differs from that in [J] 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 an 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 [Cetal] to achieve an approximation of order h^p, where p is arbitrary, is currently being developed and will be described elsewhere.

Let us consider the two-point boundary value problem where the $N \times N$ matrix $\mathbf{L}$ is as follows

$$\mathbf{L} = \begin{pmatrix} -(a_{1/2}+a_{3/2}) & a_{3/2} & 0 & \cdots & 0 & 0 & 0 \\ a_{3/2} & -(a_{3/2}+a_{5/2}) & a_{5/2} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{N-3/2} & -(a_{N-3/2}+a_{N-1/2}) & a_{N-1/2} \\ 0 & 0 & 0 & \cdots & 0 & a_{N-1/2} & -(a_{N-1/2}+a_{N+1/2}) \end{pmatrix}. \tag{12.11}$$

[illegible] reasons for the condition number of the matrix $\mathbf{L}$ to be large.

In order to use periodized differential operators, we consider the matrix $\mathbf{L}$ as a

$$\mathbf{L} = \mathbf{A} - a_{1/2}\mathbf{e}_1\mathbf{e}_N^T - a_{N+1/2}\mathbf{e}_N\mathbf{e}_1^T, \tag{12.12}$$

where

$$\begin{pmatrix} -(a_{1/2}+a_{3/2}) & a_{3/2} & 0 & \cdots & 0 & 0 & a_{1/2} \\ a_{3/2} & -(a_{3/2}+a_{5/2}) & a_{5/2} & \cdots & 0 & 0 & 0 \end{pmatrix} \tag{12.13}$$

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

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \mathbf{e}_N = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{12.14}$$

Table 12.1. 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	$0.10409 \cdot 10^3$	8.021
64	$0.41535 \cdot 10^3$	9.086
128	$0.16605 \cdot 10^4$	10.019
512	$0.26562 \cdot 10^5$	11.562
1024	[illegible]	[illegible]

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:

$$\mathbf{A}_w = \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{c}^T & 0 \end{pmatrix}, \tag{12.20}$$

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

$$\rho(a_{N+1/2} - a_{1/2})(\mathbf{e}_1^T - \mathbf{e}_N^T), \tag{12.21}$$

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

$$(\mathbf{c}^T, 0) = \rho(a_{N+1/2} - a_{1/2})(\hat{\mathbf{e}}_1^T - \hat{\mathbf{e}}_N^T). \tag{12.22}$$

Let us introduce the following notation:

$$\hat{\mathbf{e}}_l = \begin{pmatrix} \mathbf{r}_l \\ \rho \end{pmatrix}, \quad l = 1, N. \tag{12.23}$$

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

$$\hat{\mathbf{u}} = \begin{pmatrix} \mathbf{d} \\ s \end{pmatrix}, \tag{12.24}$$

$$\hat{\mathbf{f}} = \begin{pmatrix} \mathbf{f}^d \\ f^s \end{pmatrix}. \tag{12.25}$$

Also, let $2a = a_{1/2} + a_{N+1/2}$, $\alpha = a_{1/2}/(2a)$, $\beta = a_{N+1/2}/(2a)$, so that $\alpha + \beta = 1$. We now rewrite (12.16) as

$$\left[\begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{c}^T & 0 \end{pmatrix} - 2a \begin{pmatrix} \alpha \mathbf{r}_1 \mathbf{r}_N^T + \beta \mathbf{r}_N \mathbf{r}_1^T & \rho(\alpha \mathbf{r}_1 + \beta \mathbf{r}_N) \\ \rho(\alpha \mathbf{r}_N^T + \beta \mathbf{r}_1^T) & \rho^2 \end{pmatrix} \right] \begin{pmatrix} \mathbf{d} \\ s \end{pmatrix} = \begin{pmatrix} \mathbf{f}^d \\ f^s \end{pmatrix}, \tag{12.26}$$

where

$$\mathbf{c}^T = 2a\rho(\beta - \alpha)(\mathbf{r}_1^T - \mathbf{r}_N^T). \tag{12.27}$$

By eliminating s,

$$s = -\frac{(\alpha \mathbf{r}_1^T + \beta \mathbf{r}_N^T)\mathbf{d}}{\rho} - \frac{f^s}{2a\rho^2}, \tag{12.28}$$

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

[redacted]

REMARK 12.1 The condition number of the sparse matrix $\mathbf{B}$ after rescaling by $\mathbf{P}$ is $O(1)$ as is illustrated in Tables 12.1 and 12.2. Thus, the linear system (12.29) may be solved using (12.30) by the standard iterative methods (e.g., conjugate gradient) in $O(N)$ operations since using (12.30) only involves finding the solution of the linear system $\mathbf{Bx} = \mathbf{y}$. ∎

We look for the inverse operator in the form

[redacted]

and obtain

$$\boldsymbol{\Gamma} = \left[\mathbf{B}^{-1} - \sigma \mathbf{B}^{-1}(\alpha^2 \mathbf{r}_1 - \beta^2 \mathbf{r}_N)(\mathbf{r}_1 - \mathbf{r}_N)^T \mathbf{B}^{-1}\right], \tag{12.33}$$

[redacted]

$$\mathbf{q}^T = -\frac{1}{\rho}\left[(\alpha \mathbf{r}_1^T + \beta \mathbf{r}_N^T)\mathbf{B}^{-1} - \sigma \kappa_2 (\mathbf{r}_1 - \mathbf{r}_N)^T \mathbf{B}^{-1}\right], \tag{12.35}$$

$$\gamma = \frac{1}{\rho^2}\left(\kappa_3 - \sigma \kappa_2 \kappa_1 - \frac{1}{2a}\right), \tag{12.36}$$

where

$$\kappa_1 = (\mathbf{r}_1 - \mathbf{r}_N)^T \mathbf{B}^{-1}(\alpha \mathbf{r}_1 + \beta \mathbf{r}_N) \tag{12.37}$$

$$\kappa_2 = (\alpha \mathbf{r}_1^T + \beta \mathbf{r}_N^T)\mathbf{B}^{-1}(\alpha^2 \mathbf{r}_1 - \beta^2 \mathbf{r}_N), \tag{12.38}$$

and

[redacted]

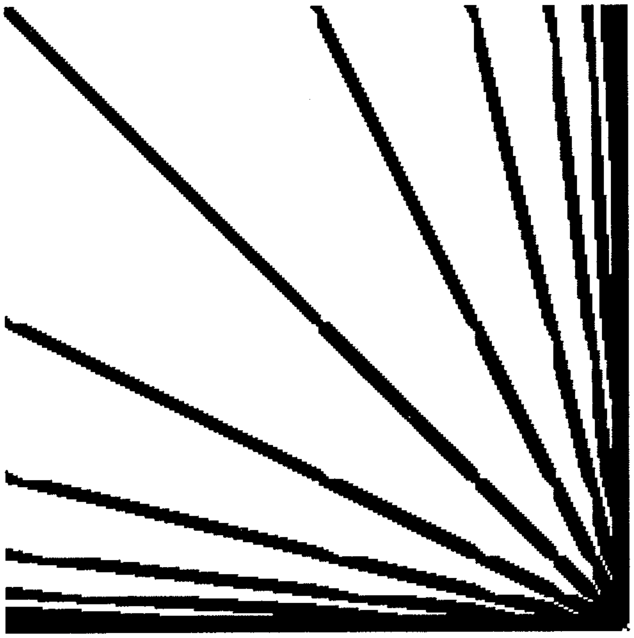

FIGURE 12.2
Matrix $\mathbf{B}$ (in the case $\mathbf{A} = \mathbf{D}$) of size 255×255 in the system of coordinates associated with the basis of Daubechies' wavelets with 3 vanishing moments. Entries with the absolute values greater than 10^{-14} are shown black.

[redacted]

and the condition number of the matrix $\mathbf{B}_p$ is $O(1)$ (see Tables 12.1 and 12.2). The matrices $\mathbf{B}$ and $\mathbf{B}_p$ are sparse matrices which is illustrated in Figure 12.2. 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 [S]

$$\mathbf{X}_{l+1} = 2\mathbf{X}_l - \mathbf{X}_l \mathbf{B}_p \mathbf{X}_l, \tag{12.41}$$

which is initialized by setting

$$\mathbf{X}_0 = \alpha \mathbf{B}_p^*, \tag{12.42}$$

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

For the full-rank matrices the iteration (12.41) 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 $O(1)$. For the full-rank matrices the iteration (12.41) is self-correcting and we use this property as described below.

[redacted]

The iteration (12.41) 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

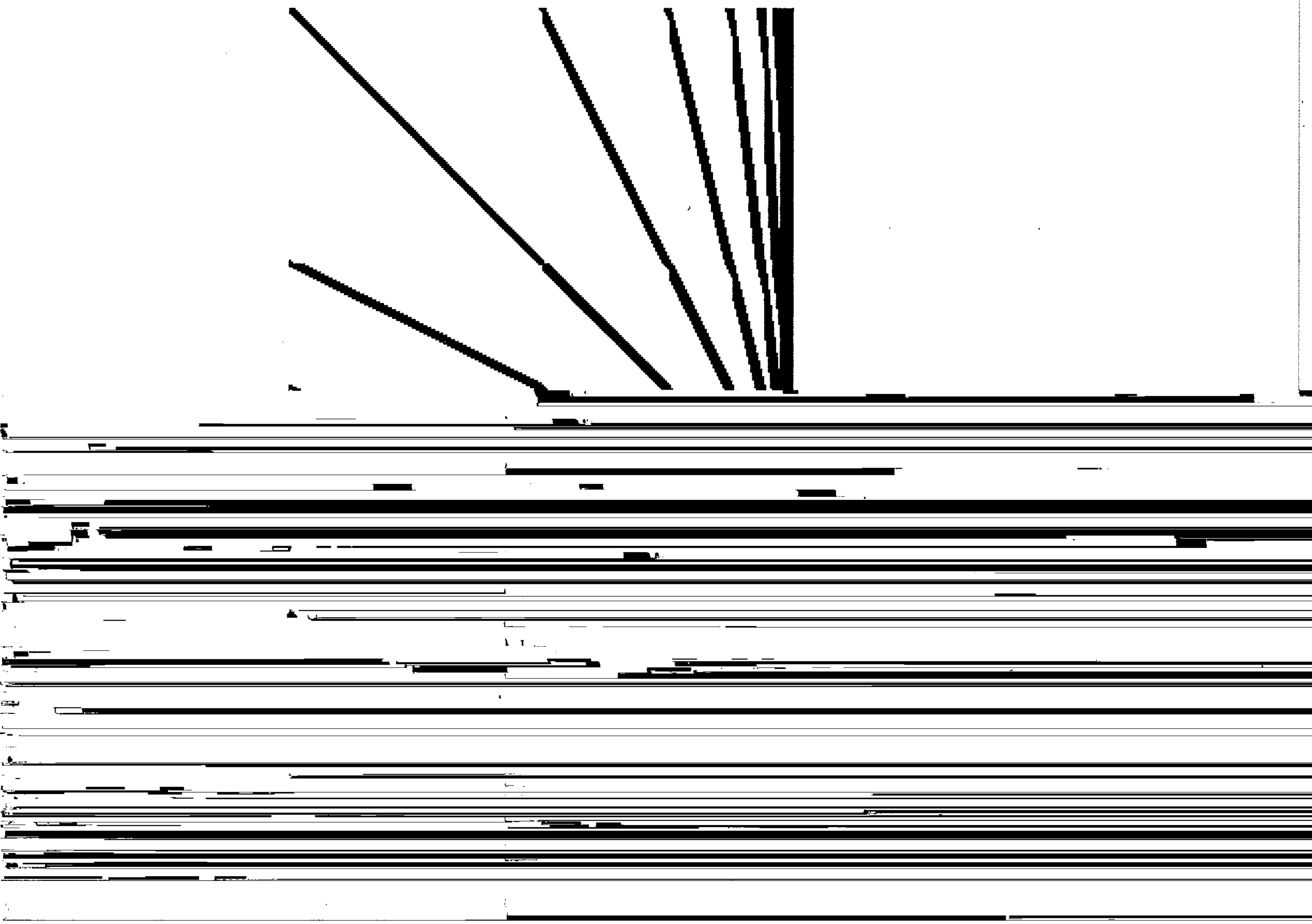

FIGURE 12.3
Matrix $\mathbf{B}^{-1}$ computed via iterative algorithm of this section with diagonal rescaling. Entries with the absolute value greater than 10^{-9} are shown black and the matrix verifies $\|\mathbf{B}\mathbf{B}^{-1}-\mathbf{I}\|$, $\|\mathbf{B}^{-1}\mathbf{B}-\mathbf{I}\| \approx 10^{-9}$.

is sparse in the wavelet basis (for a given accuracy ϵ), we only need to maintain sparsity of the intermediate matrices $\mathbf{X}$

12.5 Various extensions

12.5.1 Preconditioning to compensate for variations in a

In Section 12.3 we assumed that the function a does not change significantly over the interval $(0, 1)$. If a is such that the finite difference scheme in (12.9) is appropriate for solving the two-point boundary value problem, then we rescale (12.9) by multiplying the matrix of the system in (12.9) on both sides by the diagonal matrix

$$\mathbf{P}_a = \operatorname{diag}\left(\frac{1}{\sqrt{a_1}}, \frac{1}{\sqrt{a_2}}, \ldots, \frac{1}{\sqrt{a_N}}\right). \tag{12.44}$$

We obtain instead of (12.9),

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

where

$$v_i = u_i \sqrt{a_i} \quad i = 1, \ldots, N. \tag{12.46}$$

This corresponds to considering the operator

$$\frac{1}{a(x)} \frac{\partial}{\partial x}\left(a(x) \frac{\partial u}{\partial x}\right) \tag{12.47}$$

instead of the operator $\mathcal{L}$ in (12.7).

If a is sufficiently smooth, then we have

$$\frac{a(x-\frac{1}{2}h)}{\sqrt{a(x-h)a(x)}} = 1 + O(h^2), \tag{12.48}$$

$$\frac{a(x-\frac{1}{2}h) + a(x+\frac{1}{2}h)}{a(x)} = 2 + O(h^2), \tag{12.49}$$

and

$$\frac{a(x+\frac{1}{2}h)}{\sqrt{a(x)a(x+h)}} = 1 + O(h^2). \tag{12.50}$$

Thus, the matrix $\mathbf{L}$ corresponding to (12.45) may be written as

$$\mathbf{L} = \mathbf{L}_0 + h^2\mathbf{R}, \tag{12.51}$$

where

$$\mathbf{L}_0 = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -2 \end{pmatrix}. \qquad (12.52)$$

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 [redacted]

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

$$\mathbf{L}^{-1} = \mathbf{L}_0^{-1}(\mathbf{I} + h^2\mathbf{L}_0^{-1}\mathbf{R})^{-1} \qquad (12.53)$$

and, therefore, we need to compute $(\mathbf{I} + h^2\mathbf{L}_0^{-1}\mathbf{R})^{-1}$. Again we use the iteration in Section 12.4 and note that if the largest singular value of the operator $\mathbf{T} =$ [redacted] to solve

$$u_t = \mathcal{L}u \tag{12.55}$$

with the Dirichlet boundary conditions $u(t,0) = u(t,1) = 0$ and the initial condition $u(0,x) = u_0(x)$. Approximating (12.55) by a system of ordinary differential equations, we obtain

$$\frac{d\mathbf{u}}{dt} = \frac{1}{h^2}\mathbf{L}\mathbf{u}, \tag{12.56}$$

method

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 (12.58) and renormalize the solution after each time-step. Only the modes corresponding to the smallest eigenvalue will remain as a part of the solution

[illegible]

analysis, wavelets and fast algorithms on an interval," *C. R. Acad. Sci.*, sér. 1, to appear.

[GR] L. Greengard and V. Rokhlin. 1989. "On the numerical solution of two-point boundary value problems," Yale Univ. Technical Report, YALEU/DCS/RR-692.

[J] S. Jaffard. 1992. "Wavelet methods for fast resolution of elliptic problems," *SIAM J. Numer. Anal.*, **29**(4): 965–986.

[S] G. Schulz. 1933. "Iterative berechnung der reziproken matrix," *Z. Angew. Math. Mech.*, **13**: 57–59.