Univ. Beograd. Publ. Elektrotehn. Fak.
Ser. Mat. 14 (2004), 86–93.

PRECONDITIONING IN A WAVELET BASIS AND ITS APPLICATION TO SOME BOUNDARY VALUE PROBLEMS

Marija Rašajski

Standard finite difference methods applied to the boundary value problem $a\,(x)\,u''\,(x) + b\,(x)\,u'\,(x) + c\,(x)\,u\,(x) = f\,(x),\ u\,(0) = 0,\ u\,(1) = 0$, lead to linear systems with large condition numbers. Solving a system, i.e. finding the inverse of a matrix with a large condition number can be achieved by some iterative procedure in a large number of iteration steps. By projecting the matrix of the system into the wavelet basis, and applying a diagonal preconditioner, we obtain a matrix with a small condition number. Computing the inverse of such a matrix requires fewer iteration steps, and that number does not grow significantly with the size of the system. Numerical examples, with various operators, are presented to illustrate the effect preconditioners have on the condition number, and the number of iteration steps.

1. INTRODUCTION

Consider the boundary value problem

$$a(x)u''(x) + b(x)u'(x) + c(x)u(x) = f(x),\ x \in (0,1)$$
$$u(0) = u(1) = 0.$$

Discretizing the equation on the uniform grid with N points, and applying some standard finite difference scheme, we get the problem of solving a (large) system of linear equations. The matrix of the system is sparse, usually tridiagonal, with a large condition number, which grows with the dimension of the matrix. (The condition number of a matrix is the ratio of the largest and the smallest singular values.) For example, the condition number of the standard matrix corresponding to the second derivative (with the stencil $(1, -2, 1)$) grows as N^2. The large condition number of a matrix tells us that it would be necessary to perform a large

1991 Mathematics Subject Classification: 65F35, 65L10
Keywords and Prases: Ordinary differential equations, wavelet transform

number of iterative steps to solve the corresponding system. The method we show in this paper keeps the condition number under control. We show that for certain types of problems the condition number does not change much if the size of the system is increased.

The idea is to project the operator into the wavelet domain, then apply preconditioner, which is a diagonal matrix in this case, find the inverse matrix, and project back the solution. We show in several examples the condition number before and after preconditioning, and the number of iterative steps for computing the inverse operator with the prescribed accuracy.

2. SOME BASIC NOTIONS OF THE WAVELET THEORY

Let us briefly introduce the notion of wavelets and the multiresolution analysis ([**1**], [**3**], [**4**]).

Definition 1. *Multiresolution analysis is a decomposition of the Hilbert space $L^2(R)$ into a chain of closed subspaces*

$$\ldots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots$$

such that:

1. $\bigcap\limits_{j\in Z} V_j = \{0\}$ *and* $\bigcup\limits_{j\in Z} V_j$ *is dense in* $L^2(R)$.
2. *For any* $f \in L^2(R)$ *and any* $j \in Z$, $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$.
3. *For any* $f \in L^2(R)$ *and any* $k \in Z$, $f(x) \in V_0 \Leftrightarrow f(x-k) \in V_0$.
4. *There exists a scaling function* $\varphi \in V_0$ *such that* $\{\varphi(x-k), k \in Z\}$ *is an orthonormal basis of* V_0.

For practical purposes we define the finest and the coarsest level of the resolution leading to the following chain of subspaces:

$$V_0 \subset V_1 \subset V_2 \subset \ldots \subset V_n, \quad V_n \in L^2(R).$$

It is easy to see that the scaling function satisfies the dilation equation

$$\varphi(x) = \sum_{k=0}^{N-1} c_k \sqrt{2}\varphi(2x-k), \quad c_k = \int_R \varphi(x)\sqrt{2}\varphi(2x-k)\,dx.$$

Filter coefficients $h_k = \frac{1}{\sqrt{2}} c_k$ are often used instead of the coefficients c_k. These filter coefficients are all we need in order to apply the wavelet transform. This fact is the key feature of the wavelet transform, since the most of the wavelets in use today are not given in the closed form.

Let W_j be the orthogonal complement of V_j in V_{j+1}, $V_{j+1} = V_j \oplus W_j$. We get

$$V_n = V_0 \oplus W_0 \oplus W_1 \oplus ... \oplus W_{n-1}.$$

Let us define the translations and dilations of the scaling function φ by: $\varphi_{jk} = 2^{\frac{j}{2}} \varphi\left(2^j x - k\right)$, $k, j \in Z$. Then, $V_j = \text{span}\left\{\varphi_{jk}, k \in Z\right\}$. There exists a function ψ, called the wavelet, such that $W_j = \text{span}\left\{\psi_{jk}, k \in Z\right\}$, where $\psi_{jk} = 2^{\frac{j}{2}} \psi\left(2^j x - k\right)$, $k, j \in Z$, and

$$\psi(x) = 2 \sum_{k=0}^{N-1} (-1)^{N-k-1} h_{N-k-1} \varphi(2x - k).$$

The following statements are equivalent:

1. $\left\{1, x, x^2, \ldots, x^{p-1}\right\}$ are the linear combinations of the functions $\varphi(x - l)$.
2. $\left\| f - \sum\limits_l s_l \varphi\left(2^j x - l\right) \right\| \leq C 2^{-jp} \left\| f^{(p)} \right\|$, with $s_l = 2^j \int\limits_R f(x) \varphi\left(2^j x - l\right) dx$.
3. $\int\limits_R x^m \psi(x)\, dx = 0$, for $m = 0, 1, \ldots, p-1$, the vanishing moment property.
4. $\int\limits_R f(x) \psi\left(2^j x\right) dx \leq C 2^{-jp}$.

Projecting the function f into the space V_j gives the approximation of that function at the corresponding level, and projecting it into the W_j gives the details. Representation of a function f in basis

$$\left\{\varphi_{0k}, \psi_{jk}, j \in \{0, 1, ..., n-1\}, k \in Z\right\}$$

is given by

$$f(x) \approx \sum_{k \in Z} s_k^0 \varphi_{0k}(x) + \sum_{j=0}^{n-1} \sum_{k \in Z} d_k^j \psi_{jk}(x).$$

Coefficients s_k^0 and d_k^j are computed from inner products $\langle f, \varphi_{0k} \rangle$ and $\langle f, \psi_{jk} \rangle$. Because of the recursive nature of the equation defining the scaling function, we get the key feature of the wavelet transform. The integrals defining the inner products are never computed, but instead, the coefficients of the expansion are acquired using only the filter coefficients and approximation coefficients from the previous level. Starting from the point values of the function f at 2^n equally spaced points x_k, $k \in \{0, 1, ..., 2^n - 1\}$, we obtain the wavelet coefficients in n steps of decomposition. At one level of decomposition, starting from 2^j approximation coefficients, two sets of coefficients are computed: 2^{j-1} approximation coefficients, and 2^{j-1} detail coefficients.

In this paper we use so-called periodized wavelets, which are suitable for functions defined on an interval, and that is what we are dealing with in this paper.

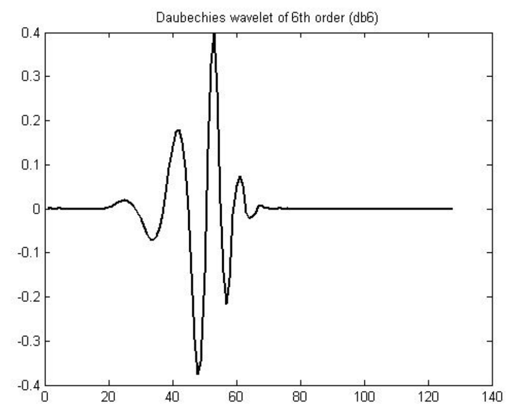

Figure 1: Wavelet Db6

The wavelet transform is usually performed via pyramid algorithm, because of its efficiency. Another way is by constructing the matrix W of the wavelet transform. Then, the wavelet transform of a vector v is achieved by matrix multiplication Wv. Wavelet transform is orthogonal, and therefore $W^{-1} = W^T$. Application of the wavelet transform to a matrix M is achieved by computing WMW^T.

3. THE INVERSE OPERATOR IN THE WAVELET BASIS

The first step is the periodization of the matrix corresponding to the differential operator of a given problem. The periodizations means the "wrapping around" of the coefficients in the matrix. For example consider the matrix D_2 corresponding to the periodized second derivative:

$$D_2 = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 & 0 & 1 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 1 & -2 \end{bmatrix}.$$

Diagonal preconditioning can work for any finite difference matrix, corresponding to a periodized differential operator [2].

Consider the discretized problem $Lu = f$, where L is a difference operator. Let L_p be the periodized matrix corresponding to L (the size of the matrices is N). Following the work of [2], but with the different ordering of the matrix of the wavelet transform, we find the inverse of the operator L_p (and then L). Briefly, we apply wavelet transform to the matrix L_p, $W(L_p) = WL_pW^T$. It is easily shown that the first column of the matrix $W(L_p)$ is a zero column, thus the problem is reduced to finding the inverse of a matrix B (of size $N-1$). Since the wavelet transform is

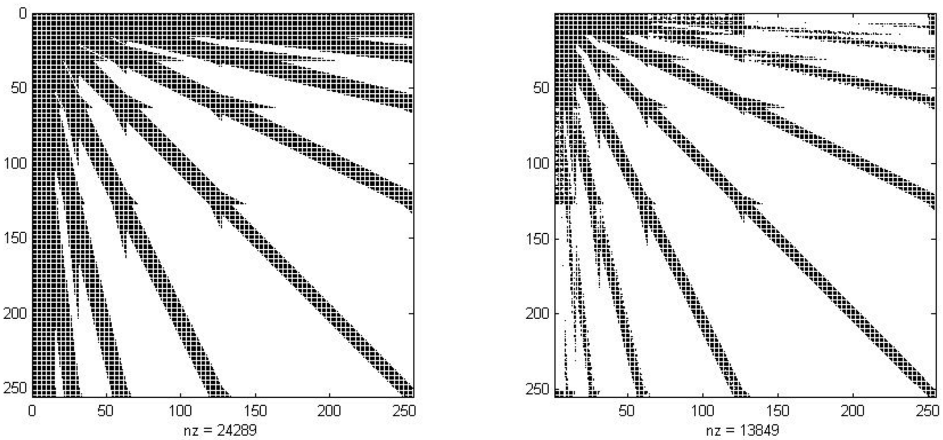

Figure 2: Operators

orthogonal, the condition number does not change after transformation. We apply preconditioning to the matrix B (in the wavelet domain), and by that obtain a matrix with a condition number which does not change much if N is increased. The important thing to mention is that both, the matrix B and its inverse B^{-1}, are sparse matrices. Their sparsity pattern, with the entries with the absolute value greater than 10^{-9}, are shown in Figure 2.

The preconditioner that we use in this paper is a diagonal matrix with the powers of 2 on the diagonal (with the size of $N = 2^n$):

$$\begin{aligned} P &= diag(\underbrace{2^n, 2^n}_{2}, \underbrace{2^{n-1}, 2^{n-1}}_{2}, \underbrace{2^{n-2}, 2^{n-2}, 2^{n-2}, 2^{n-2}}_{2^2}, \ldots \\ & \qquad \ldots, \underbrace{2^i, 2^i, \ldots, 2^i, 2^i}_{2^{n-i}}, \ldots, \underbrace{2^2, \ldots, 2^2}_{2^{n-2}}, \underbrace{2, \ldots, 2}_{2^{n-1}}). \end{aligned}$$

The preconditioning of a given matrix X is achieved by matrix multiplication: $X_P = PXP$. When we compute the inverse matrix X_P^{-1}, we get the inverse $X^{-1} = PX_P^{-1}P$.

4. NUMERICAL EXAMPLES - THE CONDITION NUMBER

The following tables show the condition numbers of various operators with and without preconditioning (kp and k, respectively). Daubechies wavelets of orders $3, 4, 6, 8$ are used in these examples. The size of the matrices is denoted by N. The same diagonal preconditioner P is used for all operators. We see from the tables that, when preconditioning is applied to the operators with dominating coefficient with the second derivative, the resulting condition number is practically independent of the size of the problem. From other two tables we see that this does not hold for the operators in which the function with the second derivative is not that dominant. But, the condition number does not grow that much if we increase the dimension of the problem. Practically, it stays under control, and regulates the number of the iterations needed for computing the inverse of the operator.

Operator:		$Lu \equiv u''$		$Lu \equiv 25u'' - xu$	
Wavelet	N	k	kp	k	kp
db3	64	4.153451e+002	9.085871	4.151735e+002	9.087316
db4	64	4.153451e+002	6.697578	4.151735e+002	6.698821
db6	64	4.153451e+002	5.261028	4.151735e+002	5.263435
db8	64	4.153451e+002	4.992119	4.151735e+002	4.994146
db3	128	1.660380e+003	10.01903	1.659683e+003	10.02027
db4	128	1.660380e+003	6.990916	1.659683e+003	6.992004
db6	128	1.660380e+003	5.289686	1.659683e+003	5.292075
db8	128	1.660380e+003	4.997137	1.659683e+003	4.999176
db3	256	6.640518e+003	10.84062	6.637711e+003	10.84169
db4	256	6.640518e+003	7.218768	6.637711e+003	7.219714
db6	256	6.640518e+003	5.303512	6.637711e+003	5.305874
db8	256	6.640518e+003	4.998509	6.637711e+003	5.000558
db3	512	2.656107e+004	11.56212	2.654980e+004	11.56303
db4	512	2.656107e+004	7.398805	2.654980e+004	7.399627
db6	512	2.656107e+004	5.310329	2.654980e+004	5.312670
db8	512	2.656107e+004	4.998877	2.654980e+004	5.000933

Operator:		$Lu \equiv \left(x^2+1\right)u'' + u$		$Lu \equiv \left(x^3+1\right)u'' - 3x^2u' + 3xu$	
Wavelet	N	k	kp	k	kp
db3	64	6.446490e+002	16.08324	6.662824e+002	16.41038
db4	64	6.446490e+002	10.26399	6.662824e+002	9.616045
db6	64	6.446490e+002	10.17642	6.662824e+002	9.894164
db8	64	6.446490e+002	9.694400	6.662824e+002	9.830358
db3	128	2.659956e+003	19.95304	2.781817e+003	19.78578
db4	128	2.659956e+003	13.71226	2.781817e+003	12.20485
db6	128	2.659956e+003	14.22476	2.781817e+003	14.04340
db8	128	2.659956e+003	12.09301	2.781817e+003	11.85213
db3	256	1.085734e+004	28.25572	1.143952e+004	27.31360
db4	256	1.085734e+004	20.28615	1.143952e+004	18.31685
db6	256	1.085734e+004	22.17994	1.143952e+004	22.15356
db8	256	1.085734e+004	19.07107	1.143952e+004	17.53844
db3	512	4.399693e+004	43.94376	4.657174e+004	41.81697
db4	512	4.399693e+004	36.06684	4.657174e+004	32.66964
db6	512	4.399693e+004	37.64197	4.657174e+004	37.83663
db8	512	4.399693e+004	33.61918	4.657174e+004	30.49973

5. NUMERICAL EXAMPLES - THE NUMBER OF ITERATIONS IN COMPUTING THE INVERSE MATRIX

The next thing to mention is the algorithm for computing the inverse matrix.

Proposition 1. [**3**] *Consider the sequence of matrices* X_k, $X_{k+1} = 2X_k - X_kAX_k$ *with* $X_0 = \alpha A^*$, *where* A^* *is the adjoint matrix and* α *is chosen so that the largest eigenvalue of* αA^*A *is less than two. Then the sequence* X_k *converges to the generalized inverse* $A^\dagger$.

When dealing with operators in a wavelet basis, the inverse can be computed in at most $O(N \log^2 N \log C)$ operations, if the operator is in the standard form, and in $O(N \log C)$, if the operator is in the so-called non-standard form (C is the condition number of the matrix). In this paper we use the standard form.

Let us introduce the following notations: $L_1u \equiv u''$, $L_2u \equiv 25u'' - xu$, $L_3u \equiv (x^3+1)u'' - 3x^2u' + 3xu$, $L_4u \equiv (x^2+1)u'' + u$, $L_5u \equiv (x+1)u'' + xu' - u$. For operators L_1, L_2, L_3, L_4, L_5 we computed the inverse operators and the number of iterations is shown in the tables. L is the matrix of the discretized operator, B is the matrix obtained from L by the application of the wavelet transform, as described before, and $B_P = PBP$, where P is the preconditioner. In the columns below B_P^{-1}, B^{-1}, L^{-1} are the numbers of the iterations performed to compute those matrices. The initial matrix X_0 (from the Proposition 1) is set to be $X_0 = \frac{1.92}{\sigma_1} B_P^* B_P$, where σ_1 is the largest singular value of the matrix $B_P^* B_P$. The iterations are performed until the condition $\|B_P X_k - I\| < \varepsilon$ is satisfied.

L_1	$\varepsilon = 10^{-4}$	wavelet=	db6
N	B_P^{-1}	B^{-1}	X^{-1}
31	7	16	20
63	8	20	24
127	8	24	28
255	8	28	32
511	8	32	36

L_1	$\varepsilon = 10^{-8}$	wavelet=	db6
N	B_P^{-1}	B^{-1}	X^{-1}
31	8	17	21
63	9	21	25
127	9	25	29
255	9	29	33
511	9	33	37

L_2	$\varepsilon = 10^{-8}$	wavelet=	db6
N	B_P^{-1}	B^{-1}	X^{-1}
31	8	17	21
63	9	21	25
127	9	25	29
255	9	29	33
511	9	33	> 75

L_2	$\varepsilon = 10^{-8}$	wavelet=	db3
N	B_P^{-1}	B^{-1}	X^{-1}
31	10	17	21
63	10	21	25
127	10	25	29
255	11	29	33
511	11	33	> 75

L_3	$\varepsilon = 10^{-4}$	wavelet=	db6
N	B_P^{-1}	B^{-1}	X^{-1}
31	9	17	22
63	9	22	26
127	10	26	30
255	12	30	34
511	13	34	38

L_3	$\varepsilon = 10^{-2}$	wavelet=	db6
N	B_P^{-1}	B^{-1}	X^{-1}
31	8	16	21
63	8	21	25
127	9	25	29
255	11	29	33
511	12	33	37

L_3	$\varepsilon = 10^{-8}$	wavelet=	db6
N	B_P^{-1}	B^{-1}	L^{-1}
31	10	18	23
63	10	23	27
127	11	27	31
255	13	31	35
511	14	35	> 75

L_3	$\varepsilon = 10^{-8}$	wavelet=	db3
N	B_P^{-1}	B^{-1}	L^{-1}
31	11	18	23
63	12	23	27
127	12	27	31
255	13	31	35
511	15	35	> 75

L_4	$\varepsilon = 10^{-8}$	wavelet=	db6
N	B_P^{-1}	B^{-1}	L^{-1}
31	10	18	22
63	10	22	26
127	11	27	31
255	13	31	35
511	14	35	> 75

L_5	$\varepsilon = 10^{-8}$	wavelet=	db6
N	B_P^{-1}	B^{-1}	L^{-1}
31	10	18	22
63	10	22	26
127	11	26	30
255	13	30	34
511	14	34	38

6. CONCLUSIONS AND THE FUTURE WORK

There are diagonal preconditioners for the boundary value problem represented in the wavelet basis. Thus, it is possible to perform all necessary algebraic calculations with sparse matrices with a small condition number. The inverse matrix is also sparse in the wavelet domain, and that property speeds up the calculations. Solving the corresponding linear system requires $O(N)$ operations. The wavelets of higher order, the smoother wavelets, give smaller condition numbers, and fewer iteration steps in the algorithm of finding the inverse matrix.

What remains to be explored is the application of the similar methods to the partial differential equations. Also, the representation of operators in the non-standard form and the implementation of fast algorithms for matrices of that form, are some of the objectives.

REFERENCES

1. I. Daubechies: *Ten Lectures on Wavelets.* Number 61 in CBMS-NSF series in Applied Mathematics. SIAM, Philadelphia, 1992.
2. G. Beylkin: *On wavelet-based algorithms for solving differential equations.* In *Wavelets: mathematics and Applications*, pages 449–466.
3. G. Beylkin: *Wavelets, multiresolution analysis and fast numerical algorithms.* ICASE Lecture notes, 1993.
4. G. Beylkin, R.Coifman, V. Rokhlin: *Fast wavelet transforms and numerical algorithms I.* Comm. Pure Appl. Math., **44** (1991), 141–183.
5. M. Dorobantu: *Wavelet-based Algorithms for Fast PDE Solvers.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 1995.

Faculty of Electrical Engineering,
Bulevar kralja Aleksandra 73,
Beograd, Serbia and Montenegro

E–mail: rasajski@etf.bg.ac.yu

(Received March 3, 2004)