MATRIX APPROXIMATIONS AND SOLVERS USING TENSOR PRODUCTS AND NON-STANDARD WAVELET TRANSFORMS RELATED TO IRREGULAR GRIDS *

J. M. FORD†, I. V. OSELEDETS ‡, AND E. E. TYRTYSHNIKOV §

Abstract. Dense large-scale matrices coming from integral equations and tensor-product grids can be approximated by a sum of Kronecker products with further sparsification of the factors via discrete wavelet transforms, which results in reduced storage and computational costs and also in good preconditioners in the case of uniform one-dimensional grids. However, irregular grids lead to a loss of approximation quality and, more significantly, to a severe deterioration in efficiency of the preconditioners that have been considered previously (using a sparsification of the inverse to one Kronecker product or an incomplete factorization approach). In this paper we propose to use non-standard wavelet transforms related to the irregular grids involved and we show, using numerical examples, that the new transforms provide better compression than the Daubechies wavelets. A further innovation is a scaled two-level circulant preconditioner that performs well on irregular grids. Our proposed approximation and preconditioning techniques have been applied to a hypersingular integral equation modelling flow around a thin aerofoil, and make it possible to solve linear systems with more than 1 million unknowns in 15-20 minutes even on a personal computer.

1. Introduction. Integral equation methods inevitably require numerical treatment of matrices that are dense and usually possess no explicit structure (like that of Toeplitz matrices). The main difficulty in many practical problems is that the dimensions of these matrices may be very large – several hundred thousands or even millions. In spite of several modern approaches to solving problems on these scales [7, 8, 10, 13, 16, 20, 21], there is still a great need for faster practical algorithms.

All the cited approaches capitalize, explicitly or implicitly, on approximation of the original matrices by "simpler" matrices of hierarchical block multilevel structure. Thus, the original matrix never appears as a full array of its entries and is dealt with only through a procedure of fast approximate matrix-vector multiplication. Such a procedure may need relatively little storage for data coming from the original matrix, and methods of acquisition of these data are often referred to as *data-compression* procedures.

We consider here the cases where the problem domain is the tensor-product of lower dimension domains. In these cases it is possible to avoid any complicated block structures by using Kronecker-product approximations [23, 24] implemented using the incomplete cross approximation algorithm [21] (see matrix theory arguments in [6, 5], an adapted version in [4], a similar algorithm with interpolation arguments in [1]). Moreover, the corresponding Kronecker factors can be further compressed by the block low-rank constructions of [9] or by wavelet sparsification [4], which results in almost linear matrix-vector complexity and superlinear compression rate.

In this paper we pursue the approach of [4] and propose better data-compression procedures in the case of irregular one-dimensional grids (*twice better* in the numerical examples considered here). The new procedures are based on non-standard wavelet

*This work was supported by EPSRC Postdoctoral Research Fellowship ref: GR/R95982/01 and by the Russian Fund of Basic Research (grant 02-01-00590).

†Mathematics Department, UMIST, PO Box 88, Manchester M60 1QD, UK (j.ford@umist.ac.uk).

‡Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Street, 8, Moscow(ivan@bach.inm.ras.ru).

§Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Street, 8, Moscow(tee@inm.ras.ru).

transforms constructed for the given irregular grids [12].

We also present a *scaled two-level circulant* preconditioner with good performance on irregular grids (where the preconditioners of [4] proved to be very inefficient).

The logic of our approach is very simple. Given a linear system $Ax = b$ (by which we mean, "given b and a procedure to calculate any entry of A") and an allowed bound ε on the relative error (in the Frobenius norm), we perform the following steps:

(A) Approximate A with a sum of Kronecker products [1]

$$B = \sum_{k=1}^{r} U_k \otimes V_k, \tag{1.1}$$

$$||B - A||_F \leq \varepsilon ||A||_F, \tag{1.2}$$

where U_k and V_k are of dimensions $p \times p$ and $q \times q$ respectively, and $n = pq$ is the dimension of matrix A. For simplicity, we assume below that $p = q = n^{1/2}$.

(B) Apply the non-standard wavelet transform with a prescribed number of vanishing moments m to each Kronecker factor:

$$P_k = WU_kW^T, Q_k = WV_kW^T, 1 \leq k \leq r. \tag{1.3}$$

Here W is the matrix of a non-standard wavelet transform of degree m. P_k and Q_k are pseudo-sparse matrices. Selecting an appropriate threshold $\tau = \tau(\varepsilon, P_k, Q_k)$ and setting to zero all elements in P_k and Q_k that are smaller in modulus that τ, we finally approximate B with a matrix

$$C = W^{-T} \otimes W^{-T} DW \otimes W \approx B, D = \sum_{k=1}^{r} P_k^\tau \otimes Q_k^\tau, \tag{1.4}$$

that approximates B with a desired accuracy ε :

$$||B - C||_F \leq \varepsilon \leq ||B||_F. \tag{1.5}$$

(C) Construct a preconditioner M^{-1} for the matrix A. There are several options. In this paper we use a block circulant preconditioner with scaling.

(D) Apply GMRES to solve

$$CM^{-1}y = b. \tag{1.6}$$

Output $M^{-1}y$ as an approximation to the exact solution x.

It is expected that $r \ll n$. Thus, the format (1.1) requires storage of only $2rn \ll n^2$ numbers. Step (A) allows us to store an approximation to A in operative memory.

Formulation and proof of upper estimates on r were first proposed in [23, 24] and then, under different assumptions, in [9]. Kronecker approximations of low Kronecker rank can be computed efficiently by a version of LU decomposition with a special dynamic choice of pivots (the approach was presented in [21] and an adapted detailed description can be found in [4]). Since only the entries of some $O(r)$ rows and columns of A are used during this algorithm, we call it the *incomplete cross approximation* algorithm.

[1] $[u_{kl}] \otimes V$ is a block matrix of the form $[u_{kl}V]$.

The purpose of step (B) is two-fold: to decrease the memory required to store matrix B and to reduce matrix-by-vector multiplication costs. Indeed, the multiplication of B by a vector takes $O(n^{3/2})$ operations, which is already much better than a standard $O(n^2)$ rule, but still time consuming (even for n of several thousands). Steps (A) and (B) allow us to multiply A by a vector with any desired accuracy very quickly. But, since the number of GMRES iterations can be large (especially in the case of irregular grids), we should find an appropriate preconditioner to reduce it. This is the purpose of step (C).

This paper can be viewed as a natural sequel to [4]. A major innovation is the adoption of non-standard wavelet transforms [12] specially constructed for the given irregular grids. These transforms are presented in Section 2. Numerical tests in [12] and application of these transforms to the Kronecker factors in Step (B) prove that the non-standard wavelet transforms provide better data compression with the same level of accuracy even for uniform grids. Another novelty is a scaled circulant preconditioner described in Section 3.

In Section 5, we apply the above steps (A)-(D) to solve a hypersingular integral equation modelling a flow around a thin rectangular-shaped aerofoil [2, 11]. Theory and proof for numerical schemes for this equation are still not complete. So numerical experiments with increasing grid sizes may help to find a better numerical scheme and show up what should be expected and then might be proved. Such experiments have not been conducted previously and are possible now only due to the new computation techniques presented in this paper.

2. Non-standard wavelet transforms. Classical wavelets and their application in multiresolution analysis of data are inherently related to uniform grids and the Fourier transform. However, in real-life applications non-uniform grids are often required. In the irregular grid cases, construction of functions and transforms with properties analogous to classical wavelets (for example, wavelets of the Daubechies family [3]) requires a totally different approach.

The non-standard wavelets that we use here are constructed in the following way (see [12] for more details). Given a grid $x_i,\ i = 1, \dots, n+k+1$, on an interval and a subgrid $\widetilde{x}_i,\ i = 1, \dots, N+k+1$, with $N < n$, we introduce B-splines (see, for example, [15, 25]) of order k as follows:

$$B_i(x) = [x_i; \dots; x_{i+k+1}](y-x)_+^k, \quad i = 1, \dots, n,$$

$$\widetilde{B}_i(x) = [\widetilde{x}_i; \dots; \widetilde{x}_{i+k+1}](y-x)_+^k, \quad i = 1, \dots, N.$$

The square brackets symbolize the divided differences in the y variable, and $(z)_+ = z$ if $z \geq 0$ and 0 otherwise.

Define V and $\widetilde{V}$ as the spans of $B_i,\ i = 1, \dots, n$, and $\widetilde{B}_i,\ i = 1, \dots, N$, respectively. Notice that V and $\widetilde{V}$ are merely the spaces of spline functions on the grids x_i and $\widetilde{x}_i$. Since $\widetilde{x}_i$ is a subgrid of x_i, we conclude that $\widetilde{V}$ is a subspace V. Consequently,

$$\widetilde{B}_i = \sum_s r_{is} B_s. \tag{2.1}$$

The coefficients r_{is} are called *refinement* coefficients. The wavelet space W is defined as a complement of $\widetilde{V}$ in V. Let us take any convenient space W with the known basis

$$\widetilde{\psi}_i = \sum_s \beta_{is} B_s, \quad i = 1, ..., n-N, \tag{2.2}$$

and then "improve" it (redefine $\widetilde{\psi}_i$) using *the lifting scheme* of Sweldens [17]:

$$\widetilde{\psi}_i = \sum_s \beta_{is} B_s - \sum_{j=j_{min}}^{j=j_{max}} \alpha_{ij} \widetilde{B}_j, \quad i = 1, \ldots, n-N.$$

To define the unknown coefficients α_{ij} (the so-called *lifting coefficients*) we demand that the functions $\widetilde{\psi}_i$ have a prescribed number (say, m) of vanishing moments:

$$\int \widetilde{\psi}_i x^p dx = 0, \quad p = 0, \ldots, m.$$

To make the number of equations be equal to the number of unknowns, set

$$j_{max} = j_{min} + m.$$

The prescription for m vanishing moments gives us a linear system for the lifting coefficients, which is explicitly solved in [12].

THEOREM 2.1.

$$\alpha_{ij} = \sum_s \beta_{is} [x_i; \ldots; x_{i+k+1}] P_j(x), \tag{2.3}$$

where $P_j(x)$ is a polynomial of degree $m+k+1$ such that

$$P_j(\widetilde{x}_r) = \begin{cases} q_j(\widetilde{x}_r), & j_{\min} \le r \le j, \\ 0, & j < r \le j_{\max} + k + 1, \end{cases} \tag{2.4}$$

$$q_j(x) = \begin{cases} (\widetilde{x}_j - \widetilde{x}_{j+k+1}) \prod_{l=j+1}^{j+k} (x - \widetilde{x}_l), & k \ge 1, \\ (\widetilde{x}_j - \widetilde{x}_{j+1}), & k = 0. \end{cases}$$

This theorem directly yields the following algorithm for the lifting coefficients.

ALGORITHM 2.1. *Given*

- *a grid $x_i, i = 1, \ldots, n+k+1$,*
- *a subgrid $\widetilde{x}_i, i = 1, \ldots, N+k+1$,*
- *arrays $j_{min}(i),\ i = 1, \ldots, n-N$, and $\beta_{is},\ i = 1, \ldots, n-N,\ s = s_0, \ldots, s_1$,*

calculate the lifting coefficients by the following FORTRAN-like code:

```
do i = 1, n − N
   do j = j_min(i), j_min(i) + m
      do r = j_min(i), j_min(i) + m + k + 1
         Calculate q_j(x̃_r).
      end do
      Calculate P_j(x) using Newton interpolation formulae.
      Calculate α_ij = Σ_{s=s_0}^{s_1} β_is [x_i; ...; x_{i+k+1}] P_j(x).
   end do
end do
```

Also we need to calculate the refinement coefficients. They satisfy linear equations

$$\int \widetilde{B}_i x^p dx = \sum_{s=s_0}^{s_1} \int r_{is} B_s x^p dx, p = 0, \ldots, s_1 - s_0 - 1,$$

which coincide with those for the lifting coefficients. Hence, the refinement coefficients can be found by the same algorithm.

To construct the discrete transform, we take a vector

$$a = [a_1, \ldots, a_n]^\top$$

and relate it to an expansion of some function $f \in V$:

$$f = \sum_{i=1}^{n} a_i \phi_i,$$

where the $\phi_i,\ i = 1, \ldots, n$, represent a *dual* basis to the basis of $B_i \in V$. Denote by φ_i and ψ_i the dual bases for $\widetilde{B}_i$ and $\widetilde{\psi}_i$, respectively. Then

$$f = \sum_{i=1}^{n} a_i \phi_i = \sum_{i=1}^{N} c_i \varphi_i + \sum_{i=1}^{n-N} d_i \psi_i, \tag{2.5}$$

and the sought-after transform reads

$$a \to (c^T, d^T)^T,$$

$$c = [c_1, \ldots, c_N]^\top, \quad d = [d_1, \ldots, d_{n-N}]^\top.$$

Algorithm 2.2. (One level of wavelet transform.) *Given*

- *vector components* $a_i,\ i = 1, \ldots, n$,
- *number of vanishing moments* m,
- *spline order* k,
- *arrays of coefficients*
  $r_{is},\ i = 1 \le i \le N, \quad \alpha_{ij},\ 1 \le i \le N, \quad \beta_{is}, 1 \le i \le n - N$,
- *array of indices* $j_{min}(i),\ 1 \le i \le n - N$,

calculate the transformed vector components $z_i,\ i = 1, \ldots, n$, *as follows*:

```
do i = 1, N
   z_i = Σ_s r_is a_s .
end do
do i = 1, n-N
   z_{i+N} = Σ_s β_is a_s − Σ_{j=j_min(i)}^{j_min(i)+m} α_ij z_j .
end do
```

Algorithms (2.1) and (2.2) implement a one-level transform. For computation of the l-level transform, these algorithms should be applied l times, provided the sequence of embedded grids is supplied.

We need also the *inverse wavelet transform* and *inverse transposed transform*. In the general case, the transform given by the algorithm 2.2 cannot be inverted explicitly. Consider in more detail the case $k = 1$. Assume for simplicity that the total number of points is odd and $\widetilde{x}_i = x_{2i-1},\ \beta_{is} = \delta_{(2i-1)s},\ N = (n-1)/2$. In this case,

$$\widetilde{B}_i = r_{i1} B_{2i-1} + r_{i2} B_{2i} + r_{i3} B_{2i+1}.$$

The matrix of the transform can be written as

$$W = LRP,$$

where P is a permutation matrix such that

$$P[a_1, \ldots, a_n]^\top = [a_2, a_4, \ldots, a_{n-1}, a_1, a_3, \ldots, a_n]^\top,$$

R is a block matrix

$$R = \begin{bmatrix} \mathcal{D} & \mathcal{B} \\ 0 & I \end{bmatrix}, \quad \mathcal{D} = \text{diag } (r_{12}, ..., r_{(n-1)2}),$$

$\mathcal{B}$ is a bidiagonal matrix of dimension $N \times (n - N)$ with the entries

$$\mathcal{B}_{ii} = r_{i1}, \quad \mathcal{B}_{i,i+1} = r_{i3}, \qquad i = 1, \ldots, N.$$

The matrix L is a block matrix of the form

$$L = \begin{bmatrix} I & 0 \\ -\mathcal{A} & I \end{bmatrix}$$

where the non-zero entries of $\mathcal{A}$ are the lifting coefficients:

$$\mathcal{A}_{ij} = \alpha_{ij}, \quad i = 1, \ldots, n - N, \quad j = j_{min}(i), ..., j_{min}(i) + m.$$

The inverse transform has the following form:

$$W^{-1} = P^\top R^{-1} L^{-1},$$

$$L^{-1} = \begin{bmatrix} I & 0 \\ \mathcal{A} & I \end{bmatrix}, \quad R^{-1} = \begin{bmatrix} \mathcal{D}^{-1} & -\mathcal{D}^{-1}\mathcal{B} \\ 0 & I \end{bmatrix}.$$

The inverse transposed transform is

$$W^{-\top} = L^{-\top} R^{-\top} P, \quad L^{-\top} = \begin{bmatrix} I & \mathcal{A}^\top \\ 0 & I \end{bmatrix}, \quad R^{-\top} = \begin{bmatrix} \mathcal{D}^{-1} & 0 \\ -B^\top \mathcal{D}^{-1} & I \end{bmatrix}. \tag{2.6}$$

The matrices $\mathcal{D}$, $\mathcal{A}$, $\mathcal{B}$ are all banded. Therefore, W, W^{-1}, W^{-T} can each be multiplied by a vector in $O(n)$ operations.

The matrix transform of a $p \times p$ matrix Z is defined to be

$$\widetilde{Z} = WZW^\top. \tag{2.7}$$

Now, consider $B = \sum_{k=1}^{r} U_k \otimes V_k$. We apply the wavelet transform to each Kronecker factor:

$$P_k = WU_kW^\top, \quad Q_k = WV_kW^\top.$$

Then, we select a threshold τ and replace matrices P_k and Q_k with sparse matrices P_k^τ and Q_k^τ respectively, and come to an approximation to matrix B:

$$C = \left(W^{-1} \otimes W^{-1}\right) \left(\sum_{k=1}^{r} P_k^\tau \otimes Q_k^\tau \right) \left(W^{-\top} \otimes W^{-\top}\right). \tag{2.8}$$

It can easily be verified that

$$||\sum_{k=1}^{r} P_k \otimes Q_k - \sum_{k=1}^{r} P_k^\tau \otimes Q_k^\tau||_F \le \varepsilon_W ||\sum_{k=1}^{r} P_k \otimes Q_k||_F, \tag{2.9}$$

where

$$\varepsilon_W = \frac{\sum_{k=1}^{r}(||P_k - P_k^\tau||_F||Q_k||_F + ||P_k||_F||Q_k - Q_k^\tau||_F)}{||\sum_{k=1}^{r} P_k \otimes Q_k||_F}$$

is easy to compute. The non-standard transform is *not orthogonal*, therefore we should write

$$||C - B||_F \le \gamma \varepsilon_W ||B||_F, \tag{2.10}$$

where γ is a "non-orthogonality factor". Numerical experiments show that it is of order of tens. When using an orthogonal transform like that of Daubechies, $\gamma = 1$. But the ε_W appears to be much smaller in the case of non-standard transform, and the overall error is smaller.

If B is an approximation to A such that

$$||B - A||_F \le \varepsilon_K ||A||_F,$$

the error of approximation of A by C is estimated as

$$||C - A||_F \le (\varepsilon_K + \gamma\varepsilon_W + \gamma\varepsilon_K\varepsilon_W). \tag{2.11}$$

Selecting r and τ properly, we can maintain a desired approximation accuracy for A. An important result of Step (B) is better data compression, but this appears as a by-product. The main purpose of this step is reduction of the matrix-vector multiplication costs. If ν denotes the number of non-zero entries in all P_k^τ and Q_k^τ, then C can be multiplied by vector in $O(\nu\sqrt{n})$ operations.

3. Scaled circulant preconditioners. Using the above constructions, we can multiply A by a vector rapidly and with sufficient accuracy. Thus, applying GMRES or PCG to solve a given a linear system with the coefficient matrix A, we can implement any single iteration in a fast way. However, the number of GMRES iterations is observed to become especially large when we get to irregular grids, and we really need to find a suitable preconditioner.

Two options (called ILUT and IKT) were considered in [4], both using the format (1.1) as the starting point for constructing a preconditioner. The ILUT is the incomplete factorization with dynamic thresholding in the spirit of [14]. The IKP is a sparsified inverse to one (most significant, in effect) Kronecker-product term in the sum. For irregular grids, however, the IKP appears to be ineffective while ILUT requires too much memory.

Another idea is to construct a preconditioner using the matrix A directly, but using only $O(n)$ of its entries. We propose to rejuvenate and adapt the well-known constructions of multilevel circulant preconditioners (see [19, 18, 22]). However, straightforward application of circulants is efficient only for uniform grids. A novelty with irregular grids is that the circulants remain useful when constructed for a suitably scaled matrix. Thus, we begin with a scaling

$$\hat{A} = D_1 A D_2 \tag{3.1}$$

using appropriate diagonal matrices D_1 and D_2. If the diagonal entries of A are positive, we can choose $D_1 = D_2$ so that all the diagonal entries of $\hat{A}$ are equal to 1 (other possibilities are related with equalizing the norms of columns and rows of $\hat{A}$).

The *optimal two-level circulant preconditioner* Q for $\hat{A}$ is a block circulant with circulant blocks satisfying

$$||\hat{A} - Q||_F = \min_{\hat{Q} \in \mathcal{T}} ||\hat{A} - \hat{Q}||_F, \tag{3.2}$$

where $\mathcal{T}$ is the set of all two-level circulants with the same dimensions as those of $\hat{A}$. Since $\hat{A}$ is a two-level matrix, its entries $\hat{a}_{ij}$ can be pointed to by multi-indices (i_1, j_1) and (i_2, j_2), where (i_1, j_1) positions a block containing a_{ij}, whereas (i_2, j_2) specifies a place inside a block. Then, the first column entries $q_{(i_1,i_2)}$ (indexed in the same way) of Q are given by the formulae [19]

$$q_{(i_1,i_2)} = \frac{1}{n} \left(\sum_{l=0}^{p-1} \sum_{k=0}^{p-1} \hat{a}_{(l,i_1+l),(k,i_2+k)} \right), \tag{3.3}$$

where the entries of $\hat{A}$ are thought of as periodic in all four indices.

A difficulty is that computation of Q via (3.3) needs $O(n^2)$ operations, which is unacceptable. We propose to construct an *approximate optimal* preconditioner. According to (3.3), we compute the mean values for all entries belonging to the i_2-th periodical diagonal in each block along the i_1-th periodical block diagonal. This suggests that we can replace the mean value of the sequence of length n by the mean value of some subsequence, in the hope that it approximates the true mean value.

When Q is constructed, we come up with a preconditioner of the form

$$M = D_1^{-1} Q D_2^{-1}. \tag{3.4}$$

Two-level circulants are diagonalized by the 2D Fourier transform:

$$Q = \frac{1}{n}(F^* \otimes F^*)\Lambda(F \otimes F), \tag{3.5}$$

where F is the Discrete Fourier Transform matrix and Λ is a diagonal matrix of the eigenvalues of C. Consequently,

$$M^{-1} = \frac{1}{n} D_2 (F^* \otimes F^*)\Lambda^{-1}(F \otimes F) D_1 \tag{3.6}$$

is an *explicit* preconditioner for matrix A. Using the Fast Fourier Transform, we can multiply F by a vector in $O(n \log n)$ operations, and hence, M^{-1} can be multiplied by a vector in $O(n \log(n))$ operations.

4. Examples of function-related matrices. In this section we illustrate the improved compression properties of the new, grid-based wavelet transform by means of some example matrices defined on irregular 1-D grids. In each case, we compare the compression achieved using standard Daubechies transforms with that obtained using the non-standard transforms of Section 2 with $k = 1$ and $m = 4$. To do this we set to zero entries that fall below a threshold of 10^{-6} and then compare the numbers of non-zero entries in each matrix. The computational complexity of transforming vectors and matrices using the non-standard wavelets is approximately the same (actually a little less) as for a Daubechies order 3 transform, but the Daubechies transform has

fewer vanishing moments and so we would expect poorer compression even on a regular grid. We anticipate that the order 4 Daubechies transform would give compression similar to that of the non-standard wavelets on a regular grid (but at somewhat higher computational cost) and to perform rather less well on irregular grids.

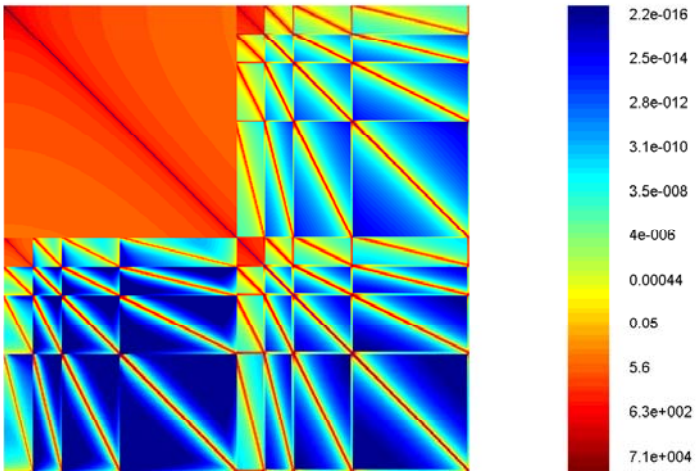

FIG. 4.1. *Function-related matrix from Example 1 with $p = 1$. Top left: original matrix; top right: Daubechies order 3 ransform; bottom left: new, nonstandard transform with 4 vanishing moments; bottom right: Daubechies order 4 transform.*

4.1. Example 1. Figure 4.1 shows the $n \times n$ matrix

$$a_{ij} = \begin{cases} 0.0 & \text{if } i = j \\ 1/|x_i - x_j|^p & \text{otherwise} \end{cases} \tag{4.1}$$

(with $p = 1$) defined on the grid $x_i = 1 - \cos(i\pi/2n)$, together with its image under three different wavelet transforms. The smoothness of the original matrix (top left in the diagram) gives rise to a large proportion of small entries when a wavelet transform is applied. The non-standard transform (bottom left) has a a larger proportion of very small entries than either the Daubechies 3 (top right) or Daubechies 4 (bottom right) transforms.

We can see the improved compression more clearly by comparing the sparsity patterns of approximations to the transformed matrices when a threshold of 10^{-6} is imposed. Figure 4.2 shows matrices approximated using Daubechies and non-standard wavelets. The number of non-zero entries using non-standard wavelets is less than two thirds of that using the Daubechies order 4 transform and only about half of that using Daubechies 3 wavelets. This is typical of other function-related matrices defined on this grid. In particular, we tested matrices of the form (4.1) for $p = 1/2, 1, 3/2, 2, 5/2$. In each case, the non-standard wavelet transform gave substantially better compression for a given approximation accuracy. Figure 4.3 illustrates this by showing the Frobenius norm of the error (obtained by applying an inverse transform to the sparsified matrix and comparing with the original dense matrix), plotted against the compression rate (i.e. the fraction of non-zero entries in the sparse matrix). here we can see that, even after taking into account the "non-orthogonality factor", the

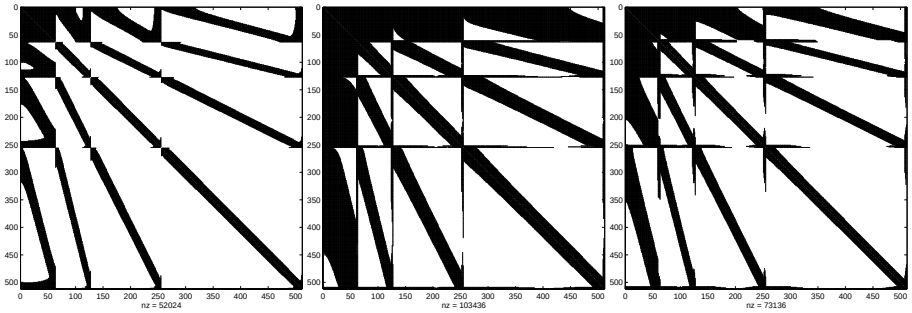

FIG. 4.2. *Approximations to the wavelet-transforms of the function-related matrix from Example 1, with $p = 1$. Left: new, nonstandard transform with 4 vanishing moments; centre: Daubechies order 3 transform; right: Daubechies order 4 transform.*

non-standard wavelets give savings in storage of about 40%. Other values of p give similar results.

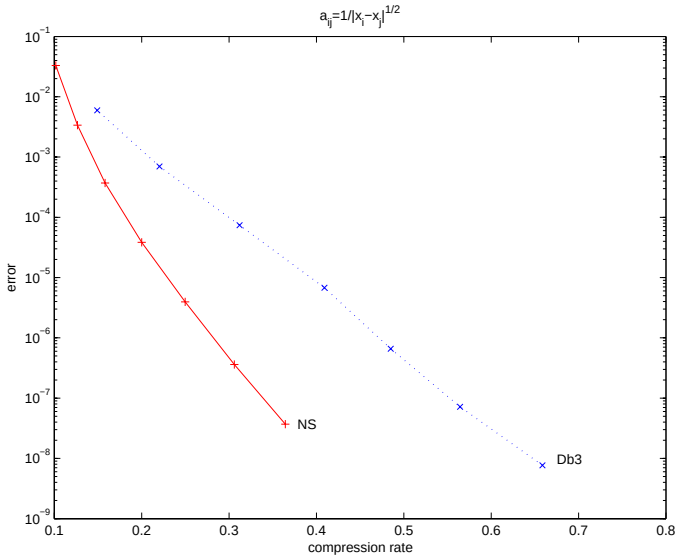

FIG. 4.3. *Frobenius norm a posteriori error plotted against compression ratio for function-related matrix from Example 1 with $p = 1/2$, using non-standard (NS) and Daubechies order 3 (Db3) wavelets.*

4.2. Example 2. Consider the $n \times n$ matrix

$$a_{ij} = \begin{cases} 0.0 & \text{if } i = j \\ -\ln|x_i - x_j| & \text{otherwise.} \end{cases}$$

defined on the grid $x_i = \ln(i)/\ln(n)$. Figure 4.2 shows the approximate matrices (with a threshold of 10^{-6}) using Daubechies 3 (left) and non-standard wavelets. The number of non-zero entries using non-standard wavelets is less than half of that using the Daubechies transform. Experiments using different functions defined on this grid, gave similar results with savings of 50% in the number of non-zero entries when the

non-standard wavelets are used being typical. Even when a "non-orthogonality factor" of 10 is included (i.e. the non-standard transformed matrix is approximated using a threshold of 10^{-7}) the savings in storage when using the non-standard transform are between 30% and 40%.

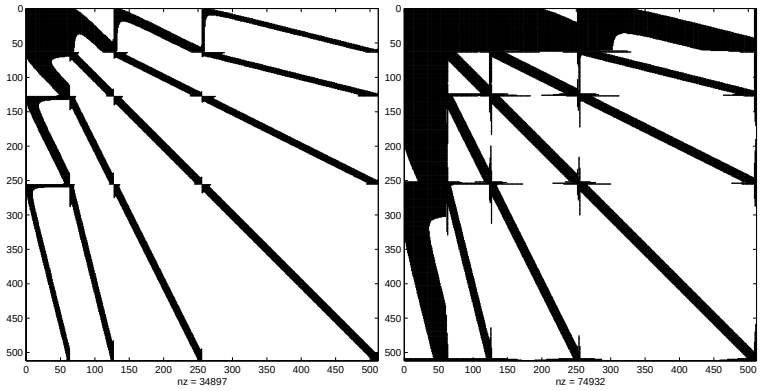

FIG. 4.4. *Approximations to the wavelet-transforms of the function-related matrix from Example 2. Left: new, nonstandard transform with 4 vanishing moments; right: Daubechies order 3 transform.*

5. Application to a hypersingular integral equation on a plate. Let us apply our proposed method to numerical solution of the following hypersingular integral equation on a plate (modelling a flow around a thin rectangular-shaped aerofoil):

$$\int\limits_{\Pi}\int \frac{u(\overrightarrow{x})}{|\overrightarrow{x}-\overrightarrow{x_0}|^3}dx = f(\overrightarrow{x_0}), \qquad \overrightarrow{x_0} \in \Pi = [0,\ 1] \times [0,\ 1]. \tag{5.1}$$

The right-hand side f is given and u is unknown. The integral in (5.1) has a strong singularity and should be treated as the Hadamard finite part integral [2, 11].

Following [2, 11], we discretize (5.1) by the collocation method: take a rectangular grid $z^i,\ i = 1, \ldots, (p+1)^2$, on Π as the Cartesian product of one-dimensional grids $x_i,\ i = 0, \ldots, p$, and $y_i,\ i = 0, \ldots, p$, $n = p^2$, and select the *collocation points*

$$w^i = (w_{k(i)}, w_{l(i)}), \quad i = (k(i)-1)p + l(i)$$

with uniquely defined integers $k(i), l(i)$ in the range from 1 to p. Node w^i belongs to $[x_{k(i)-1}, x_{k(i)}] \times [y_{l(i)-1}, y_{l(i)}]$. In the result, we approximate the function $u(\overrightarrow{x})$ with a piece-wise constant function and derive a linear system for the unknowns $u(w^i) = u_i$:

$$Au = f, \tag{5.2}$$

$$u = [u_1, ..., u_n]^\top, \quad f = [f(w^1), ..., f(w^n)]^\top,$$

$$a_{ij} = \int\limits_{\Pi_j}\int \frac{1}{|x - w^i|^3}dx, \quad \Pi_j = [x_{k(j)-1}, x_{k(j)}] \times [y_{l(j)-1}, y_{l(j)}].$$

The integrals can be calculated analytically [2, 11]:

$$a_{ij} = -\sqrt{\frac{(w_{k(i)} - x_{k(j)})^2 + (w_{l(i)} - y_{l(j)})^2}{(w_{k(i)} - x_{k(j)})(w_{l(i)} - y_{l(j)})}} + \sqrt{\frac{(w_{k(i)} - x_{k(j)-1})^2 + (w_{l(i)} - y_{l(j)})^2}{(w_{k(i)} - x_{k(j)-1})(w_{l(i)} - y_{l(j)})}}$$

$$+\sqrt{\frac{(w_{k(i)} - x_{k(j)})^2 + (w_{l(i)} - y_{l(j)-1})^2}{(w_{k(i)} - x_{k(j)})(w_{l(i)} - y_{l(j)-1})}} - \sqrt{\frac{(w_{k(i)} - x_{k(j)-1})^2 + (w_{l(i)} - y_{l(j)-1})^2}{(w_{k(i)} - x_{k(j)-1})(w_{l(i)} - y_{l(j)-1})}}.$$

Further, we take the same grids in both directions and select the grid w^i so that it is the Cartesian product of one-dimensional grids denoted by $w_i,\ i = 1, ..., p$.

Consider the following cases:

(a) Uniform grid:

$$x_i = i/p, \quad i = 0, \ldots, p, \tag{5.3}$$

$$w_i = (i - 0.5)/p, \quad i = 1, \ldots, p. \tag{5.4}$$

(b) Chebyshev grid:

$$x_i = (1 - \cos\frac{\pi i}{p})/2, \quad i = 0, \ldots, p, \tag{5.5}$$

$$w_i = (1 - \cos\frac{\pi(i - 0.5)}{p})/2, \quad i = 1, \ldots, p. \tag{5.6}$$

The choice of Chebyshev grids is inspired by theory for the analogous one-dimensional equation. In the two-dimensional case, theory is not well-developed. It is only known that the quadrature formula of rectangles (which was used) for the uniform grid approximates the hypersingular integral and the estimate

$$|I(\overrightarrow{x_0}) - I_{app}(\overrightarrow{x_0})| \le C_\delta |h \ln h|,$$

($h = 1/p$ is a grid step and δ is a distance between $\overrightarrow{x_0}$ and the border of the rectangle Π) is valid. The constant C_δ tends to infinity as $\overrightarrow{x_0}$ approaches the border. There are no theorems about the convergence of particular numerical schemes (even for the uniform grid). Therefore, numerical treatment for small h (and large n) would give a better insight into properties of approximation techniques for this problem.

TABLE 5.1
Numerical results on the irregular grid.

n	16129	65025	261121	1046529
r	20	22	25	20
ε	$9.7 \cdot 10^{-8}$	$8.4 \cdot 10^{-8}$	$9.8 \cdot 10^{-8}$	$8.1 \cdot 10^{-6}$
Matrix-by-vector time	0.3 sec	1.6 sec	7.7 sec	15.6 sec
Number of iterations	28	30	33	38
Prec. construction time	4.0 sec	16.4 sec	1.1 min	4.4 min
Solution time	11.2 sec	59.4 sec	5.7 min	14.5 min
Relative solution error	$5.8 \cdot 10^{-7}$	$1.1 \cdot 10^{-6}$	$9.9 \cdot 10^{-7}$	$2.8 \cdot 10^{-5}$

Table 5.1 presents some numerical results for the Chebyshev (irregular) grid. "The relative solution error" is the error of the solution of the linear system. To measure

it, we set the right-hand side of the algebraic equation to be the sum of the first, fifth and tenth columns of A, so we know the exact solution and can report on the accuracy in the most reliable way.

In Table 5.2 we compare the performance of GMRES with scaled (approximate optimal) two-level circulant preconditioner and without it.

TABLE 5.2
Scaled circulant preconditioner for the Chebyshev grid.

n	16129	65025	261121	1046529
Number of iterations(without prec.)	137	336	> 600	> 600
Number of iterations(with prec.)	28	30	33	38

Let us compare the convergence of the approximate solution to the exact solution of the initial integral equation. In order to do this, we set f to be the right-hand side for $u = (\overrightarrow{x})_1$. The point-wise errors of the solution on the uniform and Chebyshev grids ($p = 255$) are given in Fig. 5.1.

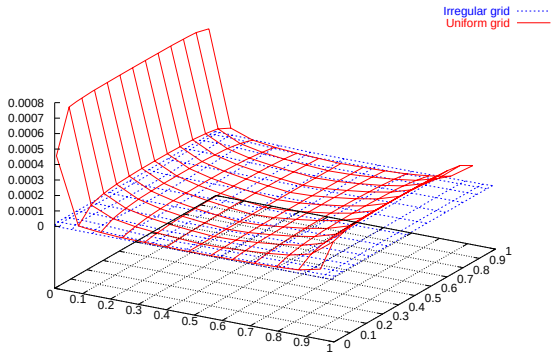

FIG. 5.1. *Point-wise error of the solution for different grids*

The L_2 and maximum errors of the approximate solution are plotted in Fig. 5.2. Clearly, the irregular grid gives much better convergence. (Theoretical estimates of the convergence rate are lacking.)

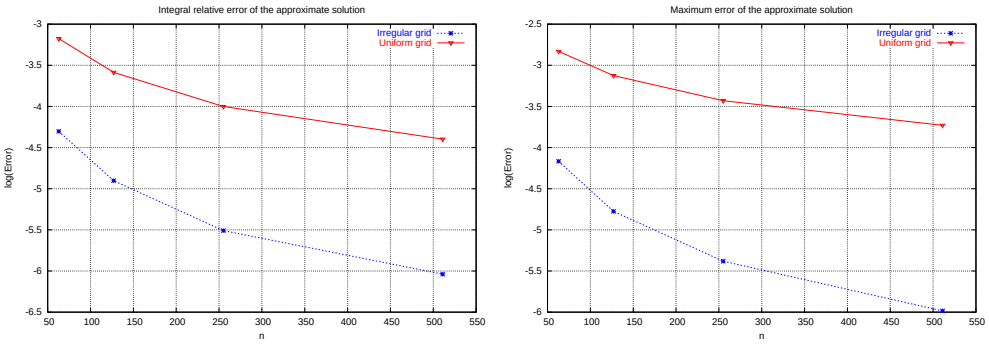

FIG. 5.2. L_2 *and maximum errors against* p

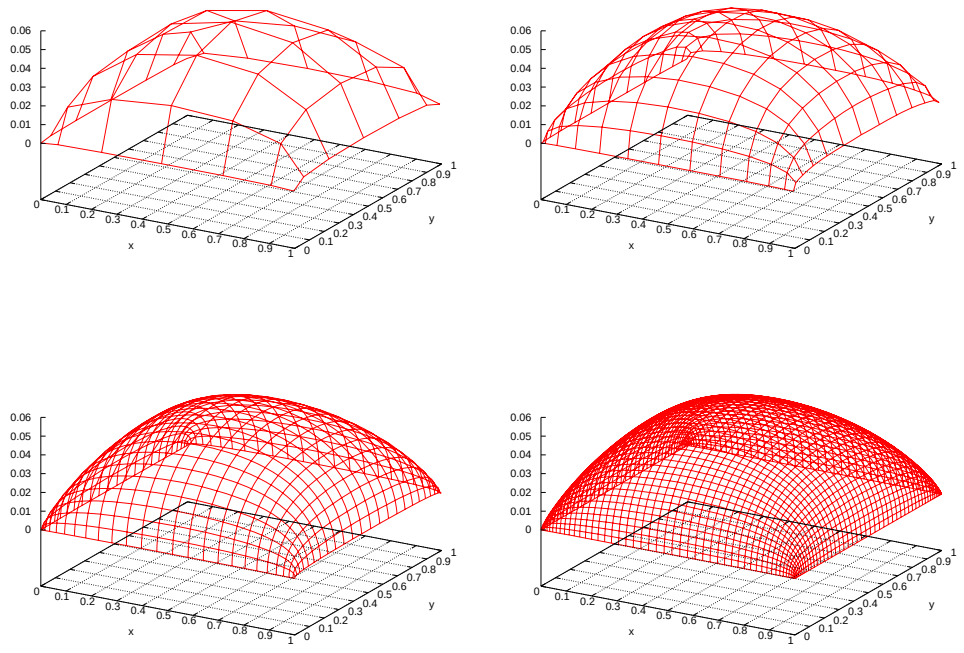

FIG. 5.3. *Solution on the irregular grid ($p = 63, 127, 255, 511$).*

The tests above show that the irregular grid is better than the uniform grid. But, the right-hand-side was very special (it was not even bounded), so the results obtained do not fit completely the physical problem (although they are very encouraging). From physics of the problem, it is known that the solution u should tend to zero for any bounded f. All the results below are obtained for $f = 1$. We plot $-u$ instead of u. Fig. 5.3 contains the results for the irregular grid and different numbers of grid points. In Fig. 5.4 analogous results are given for the uniform grid.

It can be seen, that in both cases the numerical solution stabilizes. To give some numerical estimate of the convergence rate, we plot the L_2 solution norm in Fig. 5.5 for both grids. It is clear that the irregular grid furnishes much better convergence. Notice an interesting difference in the behavior of L_2 norms: in the case of uniform grids it decreases as p increases, whereas in the case of irregular grids it increases.

6. Conclusion. We have developed an approximation technique that allows certain dense matrices to be stored using only a small fraction of the memory that would be required for their explicit representation. We have been concerned with matrices that are defined in terms of a function on a multi-dimensional grid. Provided that the grid can be expressed as the tensor product of two or more 1-dimensional grids, the matrix can be approximated by a sum of Kronecker products by means of the Incomplete Cross Approximation algorithm of Tyrtyshnikov [21]. A further reduction in storage requirements can be achieved by applying a suitable wavelet transform and setting to zero entries that fall below a chosen threshold.

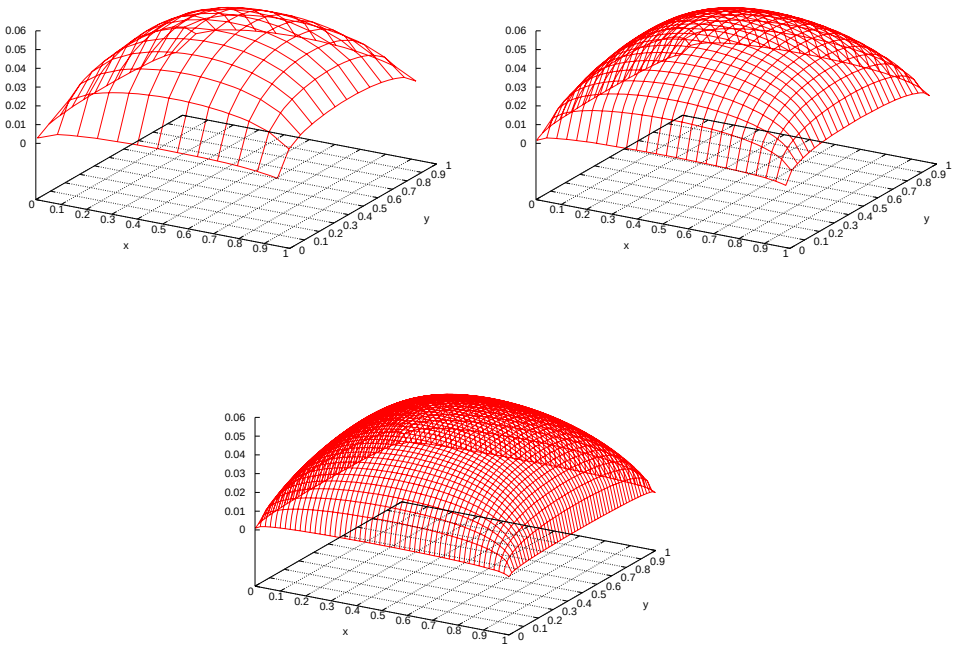

FIG. 5.4. *Solution on the uniform grid ($p = 127, 255, 511$).*

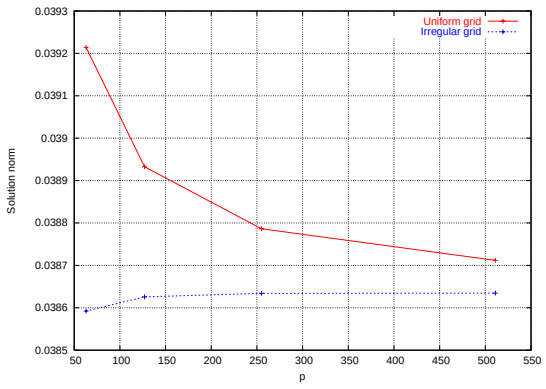

FIG. 5.5. *L_2-norm of the solution*

For regular 1-D grids standard wavelets, such as the (orthogonal) Daubechies family give satisfactory results [4], but when the underlying grids are non-uniform these are considerably less effective. In this paper we make use of the algorithm of Oseledets [12], which enables custom-built biorthogonal wavelets to be designed to give good approximation of functions defined on any given 1-D grid. In experiments, we have confirmed that these new wavelets give improved compression of matrices derived from functions on irregular 1-D grids with a reduction in storage requirements

of up to 50% compared with using Daubechies wavelets with similar computational complexity.

Having thus expressed the matrix as a sum of Kronecker products of sparse matrices, we can solve the corresponding linear system by an iterative method such as GMRES or PCG. The Kronecker product representation enables each matrix-vector product to be computed at low cost, but a preconditioner is required in order to keep the number of iterations in check. Preconditioners that were developed previously for regular grids [4] are ineffective in the non-uniform case, so we have instead adapted the multilevel circulant preconditioners first proposed in [19].

The most important new features in our approach are:

- the use of non-standard wavelets, designed to suit the underlying 1-D grids, to improve compression of the Kronecker factors;
- adaptation of multilevel circulant preconditioners for matrices on non-uniform grids by introducing a diagonal scaling.

Our combination of Kronecker-product approximation, non-standard wavelet compression and scaled block circulant preconditioning provides an efficient method of solving dense systems of equations arising from discretization on nonuniform grids. We have demonstrated its effectiveness by solving a hypersingular integral equation using finer grids than has previously been possible. In this example the ability to use irregular grids is important, since the (irregular) Chebyshev grid gives much better convergence than does a uniform grid.

We are confident that the ideas presented in this paper will be useful in solving a wide range of problems involving integral equations over multi-dimensional domains.

REFERENCES

[1] M. Bebendorf and S. Rjasanow, Adaptive low-rank approximation of collocation matrices, *Computing*, 70:1–24, 2003.

[2] S. M. Belotserkovsky, I. K. Lifanov, *Method of discrete vortices*, CRC Press, 1992.

[3] I. Daubechies, Orthonormal bases of compactly supported wavelets, *Communications of Pure and Applied Mathematics*, 41(7): 909-996, 1988.

[4] J. M. Ford, E. E. Tyrtyshnikov, Combining Kronecker product approximation with discrete wavelet transforms to solve dense, function-related systems, *SIAM J. Sci. Comp.*, 25(3):961–981, 2003.

[5] S. A. Goreinov, E. E. Tyrtyshnikov, The maximal-volume concept in approximation by low-rank matrices, *Contemporary Mathematics*, 208:47–51, 2001.

[6] S. A. Goreinov, E. E. Tyrtyshnikov, N. L. Zamarashkin, A Theory of Pseudo-Skeleton Approximations, *Linear Algebra Appl.* 261: 1–21, 1997.

[7] W. Hackbusch, A sparse matrix arithmetic based on $\mathcal{H}$-matrices. I. Introduction to $\mathcal{H}$-matrices, *Computing*, **62**:89–108, 1999.

[8] W. Hackbusch, B. N. Khoromskij, A sparse $\mathcal{H}$-matrix arithmetic. II. Application to multi-dimensional problems, *Computing*, **64:**21–47, 2000.

[9] W. Hackbusch, B. N. Khoromskij, and E. E. Tyrtyshnikov, Hierarchical Kronecker tensor-product approximations, Max-Panck-Institut für Mathematik in den Naturwissenschaften, Leipzig, Preprint No. 35, 2003.

[10] W. Hackbusch, Z. P. Nowak, On the fast matrix multiplication in the boundary elements method by panel clustering, *Numer. Math.* **54 (4)**:463–491, 1989.

[11] I. K. Lifanov, *Singular integral equations and discrete vortices*, VSP, 1996.

[12] I. Oseledets, *Application of B-splines and divided differences for construction of fast discrete wavelet transforms on irregular grids*, to appear.

[13] V. Rokhlin, Rapid solution of integral equations of classical potential theory, *J. Comput. Physics* 60: 187–207, 1985.

[14] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, An International Thomson Publishing Company, Boston, 1996.

[15] L. Schumaker, *Spline functions : basic theory*, Wiley, New York, 1981.

[16] X. Sun, N. P. Pitsianis, A matrix version of the fast multipole method, *SIAM Review*, Vol. 43, No. 2, pp. 289–300, 2001.
[17] W. Sweldens, The lifting scheme: A custom design construction of biorthogonal wavelets, *Appl. Comput. Harmon. Anal.*, 3(2): 186-200, 1996.
[18] V.V.Strela and E.E.Tyrtyshnikov, Which circulant preconditioner is better? *Math. Comput.* 65(213): 137–150, 1996.
[19] E. Tyrtyshnikov, Optimal and superoptimal circulant preconditioners, *SIAM J. Matrix Anal. Appl.*, Vol. 13, No. 2, pp. 459-473, 1992.
[20] E. E. Tyrtyshnikov, Mosaic–skeleton approximations, *Calcolo*, 33 (1-2): 47–57, 1996.
[21] E. E. Tyrtyshnikov, Incomplete cross approximation in the mosaic-skeleton method, *Computing* 64(4):367–380, 2000.
[22] E. Tyrtyshnikov and R.Chan, Spectral Equivalence and Proper Clusters for Boundary Element Method Matrices, *Int. J. Numer. Meth. Engnr.* 49:1211–1224, 2000.
[23] E. E. Tyrtyshnikov, Tensor approximations of matrices generated by asymptotically smooth functions, *Sbornik: Mathematics*, Vol. 194, No. 6, 200), pp. 147–160 (in Russian).
[24] E. E. Tyrtyshnikov, Kronecker-product approximations for some function-related matrices, *Linear Algebra Appl.*, in press.
[25] V.A.Vasilenko, *Spline finctions: theory, algorithms, codes*, Nauka, Novosibirsk, 1983.