LUND UNIVERSITY

Approximate inverse preconditioners for some large dense random electrostatic interaction matrices

Helsing, Johan

Link to publication

Total number of authors:
1

APPROXIMATE INVERSE PRECONDITIONERS FOR SOME LARGE DENSE RANDOM ELECTROSTATIC INTERACTION MATRICES [*]

JOHAN HELSING [†]

*Numerical Analysis, Centre for Mathematical Sciences, Lund University, Box 118,
S-221 00 LUND, Sweden. email: helsing@maths.lth.se*

Abstract.

A sparse mesh-neighbour based approximate inverse preconditioner is proposed for a type of dense matrices whose entries come from the evaluation of a slowly decaying free space Green's function at randomly placed points in a unit cell. By approximating distant potential fields originating at closely spaced sources in a certain way, the preconditioner is given properties similar to, or better than, those of a standard least squares approximate inverse preconditioner while its setup cost is only that of a diagonal block approximate inverse preconditioner. Numerical experiments on iterative solutions of linear systems with up to four million unknowns illustrate how the new preconditioner drastically outperforms standard approximate inverse preconditioners of otherwise similar construction, and especially so when the preconditioners are very sparse.

AMS subject classification (2000): 65F10,65R20,65F35,78A30.

Key words: dense matrices, integral equations, preconditioners, sparse approximate inverses, iterative methods, potential theory.

1 Introduction.

The purpose of this paper is to construct an efficient sparse approximate inverse preconditioner M to a $n \times n$ matrix A whose entries are given by

$$(1.1) \qquad A_{ij} = \begin{cases} -\log |z_i - z_j|, & i \neq j, \\ -\log |r_i|, & i = j, \end{cases}$$

where z_i are n somehow randomly placed points in a unit square centered at the origin in the complex plane and where each r_i is a number in $(0, d_i]$ with d_i being the distance between the point z_i and its nearest neighbour. We observe that computing an off-diagonal entry A_{ij} corresponds, up to a factor of $-1/2\pi$, to evaluating the free space Green's function for the Laplacian in two dimensions with argument $z_i - z_j$.

Interaction matrices of the form (1.1), with the Green's function possibly replaced with its three dimensional analogue, may occur as system matrices in a variety of situations in materials science and electrical engineering, for example, particle coarsening and capacitance extraction [14, 15, 18, 19]. The particular application which drives this work concerns Laplace's equation and the Dirichlet-Neumann map in domains exterior to a number n of closed contours – a problem which, in turn, has relevance for microstructural evolution, see Ref. [13] and references therein. Here, the ill-conditioning of the system matrix becomes a serious problem when solving linear systems iteratively as n grows large. In a typical example, without preconditioning, the number of iterations needed to meet a certain relative residual grows approximately as $O(n^{0.3})$. See Figure 2.2 below, uppermost curves.

We shall construct a sparse inverse preconditioner for matrices of the form (1.1) which, together with the full, that is, not restarted, GMRES iterative solver [21] and the fast multipole method (FMM) [10], has good asymptotic properties in the limit of large n. A similar quest, for a system matrix originating from the discretization of an integral equation for Laplace's equation in three dimensions, was mentioned as an interesting open problem in an influential paper by Vavasis in 1992 [22]. To our knowledge it has only received scattered attention. Reasons for this may include that the problems at large n could be most pronounced in two dimensions. Should the Greens' function in (1.1) be replaced with a more rapidly decaying one, established preconditioners are reported to work satisfactorily [15, 18, 20, 23]. Furthermore, efficient linear solvers for large systems involve other key components besides preconditioners. The choice of iterative method, here for example SQMR in conjunction with a symmetrized preconditioner rather than GMRES, could be equally important [4]. Yet an option, when z_i are somehow ordered, is to avoid iterations completely and to use a fast direct solver. Among papers that, more broadly, investigate rapidly applicable inverse preconditioners or inverses for large dense symmetric matrices coming from discretized integral equations with possibly slowly decaying kernels we particularly mention references [9, 12, 17]. These papers discuss preconditioners/inverses which are not sparse in a classical sense, but could be considered as data-sparse. Hackbusch [12] constructs rapidly invertible approximations of dense matrices in terms of hierarchical, so called, $\mathcal{H}$-matrices, and works out some details for a one-dimensional example involving the logarithmic kernel. Ford and Tyrtyshnikov [9] use a single inverse Kronecker product preconditioner and construct a fast scheme based on Kronecker product approximation, wavelet compression, and conjugate gradients or GMRES for the special case of z_i being placed on a grid that is logically equivalent to the Cartesian product of some one-dimensional grids. Martinsson and Rokhlin [17] go one step further and, at a cost of roughly 20 FMM matrix-vector multiplications, construct a compressed factorization of the inverse which can be applied at a minimal cost and with high relative accuracy to any right-hand side for the special case of z_i being placed on a smooth one-dimensional contour in the plane.

The inverse preconditioner we propose, for z_i randomly placed, is denoted

WBAI(k) in general and M_{WBAI} when reference is made to the actual matrix. The only user specified parameter is its bandwidth k. The setup cost is $O(nk^3)$. The cost of storage and application is $O(nk)$. We show, experimentally, that for A of (1.1) with z_i completely randomly placed in the unit square, r_i randomly selected in $(0, 0.5d_i]$, $n \leq 4000000$, $k = 20$, and a random right-hand side $\mathbf{b}$, the number of full GMRES iterations needed to solve $AM_{\mathrm{WBAI}}\mathbf{u} = \mathbf{b}$ and to compute $\mathbf{x} = M_{\mathrm{WBAI}}\mathbf{u}$ to a given relative accuracy increases only as $O(\log(n))$, or slower. This is often a substantial improvement over the only other competitor with similar applicability and setup cost we have found in the literature – the DBAI(k) preconditioner. We also test our new preconditioner WBAI(k) on some other matrices of the form (1.1) where the points z_i are placed in a random, but to a certain degree mutually repulsive, manner and where the numbers r_i are chosen differently. The results here are equally encouraging.

2 Sparse approximate inverses.

2.1 General ideas and the sparsity patterns $\mathcal{S}_{\mathrm{neig}}$ and $\mathcal{S}_{\mathrm{dom}}$.

Sparse approximate inverse preconditioners are matrices M that are sparse and make either of the compositions MA or AM in some sense close to the identity matrix I. This may mean that the composition is close to I in Frobenius norm, or is close to I plus a low rank perturbation, or has a small cluster radius. The matrix A could, in general, be almost any large matrix stemming from some discretization of a partial differential equation or an integral equation. See Benzi [2] for an excellent general review of the field and Chen [6] for details pertaining to dense linear systems, such as ours.

We construct right unsymmetric preconditioners M. The sparsity pattern $\mathcal{S}$ of M is the set of all index pairs (i, j) for which the entries M_{ij} are nonzero. There are numerous ways to determine sparsity patterns: statically, that is *a priori*, or dynamically, that is, during the course of construction of M. See, further [2, 3, 6, 7, 22] and references therein. In this paper we will chiefly take a local geometric approach and define the mesh-neighbour sparsity pattern $\mathcal{S}_{\mathrm{neig}}$ associated with A of (1.1) as follows: Let $j = 1, 2, \ldots, n$; for a given j, let $\mathcal{S}_{\mathrm{neig}}$ contain the k index pairs (i, j) corresponding to the k smallest values of the distance $|z_i - z_j|$, $i = 1, 2, \ldots, n$. We say that an M based on $\mathcal{S}_{\mathrm{neig}}$ has bandwidth k. The columns of M each have k nonzero entries. The rows of M on average have k nonzero entries. We also define the dominant sparsity pattern $\mathcal{S}_{\mathrm{dom}}$ associated with A of (1.1) as follows: Let $j = 1, 2, \ldots, n$; for a given j, let $\mathcal{S}_{\mathrm{dom}}$ contain the k index pairs (i, j) corresponding to the k largest entries in column j of A^{-1}.

The sparsity pattern $\mathcal{S}_{\mathrm{dom}}$ appears to be more efficient than $\mathcal{S}_{\mathrm{neig}}$, once constructed. Its setup cost $O(n^3)$, however, greatly limits its applicability in large-scale computations. The setup cost for $\mathcal{S}_{\mathrm{neig}}$ is only $O(n)$.

2.2   Properties of the matrix A and a particular choice.

The off-diagonal entries A_{ij} of (1.1) come from the evaluation of the free space Green's function for the Laplacian in two dimensions. We shall often talk about the distance between an off-diagonal entry A_{ij} and the diagonal. An entry A_{ij} is said to be far away from the diagonal when the distance $|z_i - z_j|$ is large. An entry A_{ij} is said to be close to the diagonal when the distance $|z_i - z_j|$ is small. The entries A_{ij} with indices in $\mathcal{S}_{\mathrm{neig}}$ are, thus, close to the diagonal. While entries A_{ij} of (1.1) with indices in $\mathcal{S}_{\mathrm{neig}}$ are large in modulus, one cannot generally say that the entries decay as the distance to the diagonal increases, unless the unit square is scaled with a factor of $1/\sqrt{2}$, in which case the decay is slow. Matrices of the form (1.1) are symmetric, but not necessarily positive definite for general choices of r_i.

In our development of an efficient sparse approximate inverse preconditioner for matrices of the form (1.1) we shall rely on a class of realizations where the points z_i are independent of each other with uniform density in the unit square and the numbers r_i are randomly chosen in $(0, 0.5d_i]$ with uniform density. We denote any matrix with these characteristics "the matrix A_{I}". In the numerical examples at the end of Section 3 we shall also investigate two other classes of matrices of the form (1.1) which do not share the characteristics of A_{I}. We denote any matrix in these classes "the matrix A_{II}" or "the matrix A_{III}".

The spectrum of the matrix A_{I} shows a close resemblance to the reciprocal of that of minus the Laplacian on the unit square. For p being an integer, the eigenvalues corresponding to the p^2 eigenvectors of lowest frequencies for the Laplacian are
$$(2.1) \qquad \lambda(\Delta) = \left\{ -\pi^2(i^2 + j^2) \right\}_{i,j=1,2,\ldots,p} .$$

Numerical experiments suggest that the spectrum of A_{I}, with $n = p^2$, is approximately given by
$$(2.2) \qquad \lambda(A_{\mathrm{I}}) \approx \left\{ \frac{1.2p^2}{i^2 + j^2} \right\}_{i,j=1,2,\ldots,p} .$$

If we were to multiply each entry of A_{I} with a factor $-1/(2\pi p^2)$, as to make the off-diagonal entries of A_{I} resemble those in a discretization of the free-space Green's function for the Laplacian where quadrature weights $w_{ij} = 1/p^2$ are used, the spectral approximation (2.2) would change accordingly and the connection between the two spectra is further elucidated.

2.3   Submatrices of the matrix A.

Before venturing into the construction of various sparse approximate inverses M with sparsity patterns $\mathcal{S}$ we shall define some useful submatrices of A of (1.1). For this, first let the set of m indices $\mathcal{S}_m^{(0)}$ be given by $\mathcal{S}_m^{(0)} = \{i : 1 \le i \le m\}$ and let the set of k indices $\mathcal{S}^{(j)}$ be given by $\mathcal{S}^{(j)} = \{i : (i, j) \in \mathcal{S}\}$. Now let $\hat{A}^{(j)}$ be a $k \times k$ symmetric submatrix of A containing entries of A which are close to A_{jj} and arranged in a particular order. More precisely, define a one-to-one mapping

q from $\mathcal{S}_k^{(0)}$ onto $\mathcal{S}^{(j)}$ in such a way that

$$(2.3) \qquad \begin{aligned} q(i) = j\,, \quad i = 1\,, \\ |z_{q(i)} - z_j| \geq |z_{q(i-1)} - z_j|\,, \quad i = 2, 3\ldots, k\,. \end{aligned}$$

Then let

$$(2.4) \qquad \hat{A}_{il}^{(j)} = A_{q(i)q(l)}\,, \quad i, l \in \mathcal{S}_k^{(0)}\,.$$

The matrix $\hat{A}^{(j)}$ contains k entries from each of k different columns of A. Let $A_0^{(j)}$ be a rectangular $(n - k) \times k$ submatrix of A containing the remainder of the entries in the k columns of A from which the entries of $\hat{A}^{(j)}$ were collected. More precisely, let r be a one-to-one mapping from $\mathcal{S}_{n-k}^{(0)}$ onto $\mathcal{S}_n^{(0)} - \mathcal{S}^{(j)}$. Then let

$$(2.5) \qquad {A_0^{(j)}}_{il} = A_{r(i)q(l)}\,, \quad i \in \mathcal{S}_{n-k}^{(0)}, l \in \mathcal{S}_k^{(0)}\,.$$

We remark that the particular ordering of the rows and columns of $\hat{A}^{(j)}$, given by the mapping q, is not essential for the construction of the preconditioners LSAI(k) and DBAI(k) of Sections 2.4 and 2.5 below. It simplifies, however, the construction of our new preconditioner WBAI(k) in Section 3.

2.4 Mesh-neighbour based least squares approximate inverse (LSAI).

When the entries of M are chosen as to minimize the Frobenius norm of $AM - I$ one speaks of a least squares approximate inverse preconditioner, here denoted LSAI(k) in general and M_{LSAI} when reference is made to the actual matrix. The k nonzero entries in column j of M_{LSAI} can be obtained by solving the $n \times k$ overdetermined linear system

$$(2.6) \qquad \begin{bmatrix} \hat{A}^{(j)} \\ A_0^{(j)} \end{bmatrix} \hat{\mathbf{m}}^{(j)} = \begin{bmatrix} \mathbf{e} \\ 0 \end{bmatrix},$$

in least squares sense. Here $\mathbf{e}$ is a column vector with k entries all equal to zero except for the first entry, which is one. The correspondence between the index of a particular entry in the vector $\hat{\mathbf{m}}^{(j)}$ and its row index in column j of M_{LSAI} is given by the mapping q of of (2.3).

The left images of Figure 2.1 show the spectra of $A_{\text{I}} M_{\text{LSAI}}$ with $k = 20$ and $n = 7569$ for $\mathcal{S}_{\text{neig}}$ and for $\mathcal{S}_{\text{dom}}$. The spectra are rather similar and clustered around unity. Note the difference in scales on the x- and y-axes. A series of experiments show that the spectra grow slowly towards the left with increasing n. A drawback with LSAI(k) is its setup cost. For a dense matrix and a general sparsity pattern this cost is $O(n^2 k^2)$, making LSAI(k) extremely expensive to use in large-scale computations even though, of course, it could be amortized if many right-hand sides are to be solved. For a sparsity pattern such as $\mathcal{S}_{\text{neig}}$, where the normal equations corresponding to different columns of M_{LSAI} may have system matrices with many entries in common, it is possible to achieve a setup cost of $O(n^2 k + n k^3)$. This cost is, however, also high.

The left image of Figure 2.2 illustrates the efficiency of using LSAI(k), once M_{LSAI} has been constructed, when iteratively solving linear systems involving
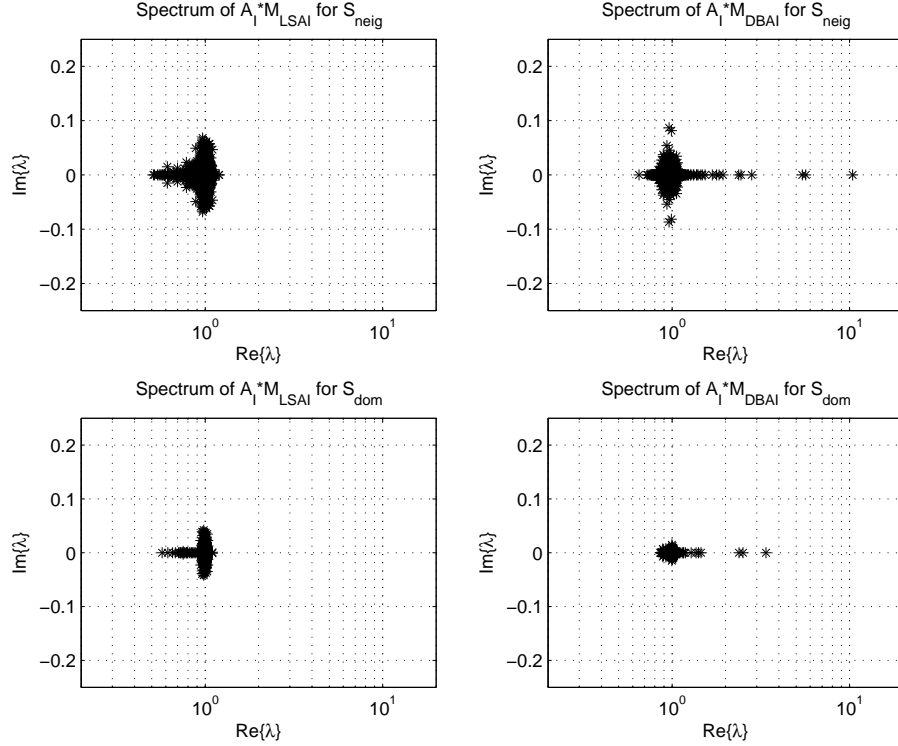
Figure 2.1: Eigenvalues $\lambda(A_\mathrm{I}M)$ for compositions of the matrix A_I with different sparse approximate inverse preconditioners and sparsity patterns. The system size n, corresponding to the number of random points z_i, is $n = 7569$ and the bandwidth is $k = 20$. All four spectra remain essentially unchanged if the unit square is scaled with a factor of $1/\sqrt{2}$.

the matrix A_I. This figure is presented for the purpose of comparison with other preconditioners. The high setup cost makes computations involving LSAI(k) even more time consuming than using no preconditioner at all. Anyhow, one can see that a larger bandwidth k gives fewer iterations for a fixed n, although the improvement with increasing k gets progressively smaller.

2.5   Mesh-neighbour based diagonal block approximate inverse (DBAI).

When the entries of M are chosen by requiring that $(AM)_{ij} = I_{ij}$ hold for all index pairs in $\mathcal{S}$ one speaks of a diagonal block approximate inverse preconditioner, here denoted DBAI(k) in general and M_DBAI when reference is made to the actual matrix. The DBAI(k) preconditioner could, equivalently, be viewed as a modification of the LSAI(k) preconditioner where the matrix $A_0^{(j)}$ of (2.6) is replaced by a matrix of zeros. The k nonzero entries in column j of M_DBAI
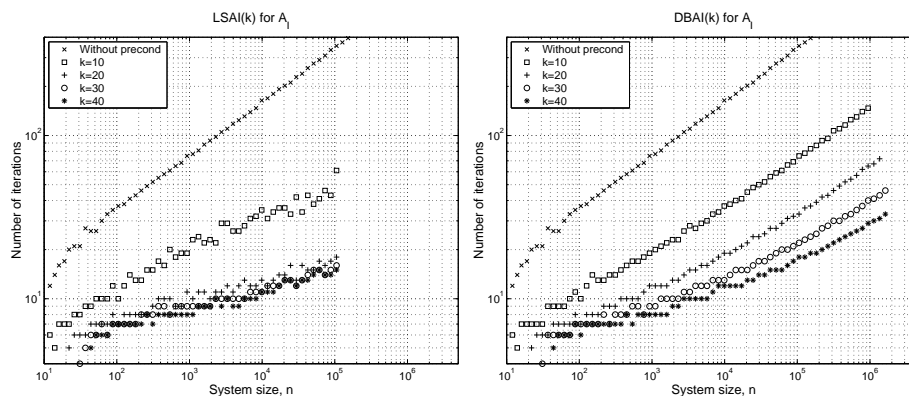
Figure 2.2: Left, number of full GMRES iterations needed to solve systems $A_\mathrm{I}\mathbf{x} = \mathbf{b}$ with a stopping criterion threshold of 10^{-8} in the relative residual, with and without LSAI(k) preconditioning. The components of the right-hand side vectors $\mathbf{b}$ are chosen randomly in $[-1, 1]$. The sparsity pattern is $\mathcal{S}_\mathrm{neig}$. Right, the same thing with DBAI(k) preconditioning.

can thus be obtained by solving the $k \times k$ linear system

$$(2.7) \qquad\qquad \hat{A}^{(j)}\hat{\mathbf{m}}^{(j)} = \mathbf{e}\,.$$

The right images of Figure 2.1 show the spectra of $A_\mathrm{I}M_\mathrm{DBAI}$ with $k = 20$ and $n = 7569$ for $\mathcal{S}_\mathrm{neig}$ and for $\mathcal{S}_\mathrm{dom}$. These spectra are not as well clustered as those of $A_\mathrm{I}M_\mathrm{LSAI}$. A series of experiments show that they grow towards the right with increasing n. The setup cost for DBAI(k) is only $O(nk^3)$, making some version of it a reasonably popular choice in various applications. See, for example, [1, 16, 19, 20, 24]

The right image of Figure 2.2 illustrates the asymptotic efficiency of using DBAI(k) when iteratively solving linear systems involving the matrix A_I. A comparison to the results with LSAI(k) in the left image shows that, once M_LSAI and M_DBAI are constructed, LSAI(k) seems more efficient than DBAI(k).

REMARK 2.1. *The terminology in this field has not yet crystallized. The name DBAI(k), used also in [1, 6, 16], may be misleading. The approximate inverse is not block diagonal. An alternative name is MN(k), meaning mesh neighbour, used in [22, 24], and referring more to how $\mathcal{S}$ is constructed than to how the entries of M are computed. We stick to the acronym DBAI(k) which reflects the fact that the entries of M_DBAI are computed by solving the linear systems (2.7) whose system matrices $\hat{A}^{(j)}$ of (2.4) could be thought of as diagonal blocks of A.*

3   A new preconditioner WBAI, in between LSAI and DBAI.

The spectra of Figure 2.1 and the comparison of the images of Figure 2.2 indicate that the LSAI(k) preconditioner, once constructed, should be more efficient than the DBAI(k) preconditioner. The DBAI(k), on the other hand,

has a much lower setup cost. A natural idea would be to find something in between these preconditioners, capturing the best features of the two. This idea is by no means new. Carpentieri, Duff, Giraud, Made, and Sylvand [4, 5], for example, describe a scheme where a dense matrix A is sparsified prior to the construction of a LSAI type preconditioner. In this way, the bulk of the entries of $A_0^{(j)}$ are set to zero. The setup cost is reduced. When the off-diagonal entries of A come from a rapidly decaying Green's function such a truncated Green's function LSAI approach could work well. Unfortunately, matrices of the form (1.1) have an underlying Green's function which is not rapidly decaying. A truncated LSAI approach does, therefore, not seem very promising if one has good asymptotics for large n in mind. The right image of Figure 2.2 illustrates this point further. Irrespective of the bandwidth k, the number of iterations eventually grows as a power law in system size n.

3.1   Approximation of the far field.

The truncated Green's function LSAI approach tries to make the approximate inverse M efficient by keeping the correct values for a few of the entries of the submatrices $A_0^{(j)}$ while setting the bulk of the entries to zero. Here we shall do something different. We shall neither keep nor zero individual entries of $A_0^{(j)}$, but instead approximate entire matrices $A_0^{(j)}$ with something simple which makes the setup cost for a LSAI type preconditioner the same as for a DBAI type preconditioner. The key observation is that the Green's function, for large arguments, is a harmonic function which is approximately constant at small scales. Therefore a submatrix $A_0^{(j)}$, which contains entries far away from the diagonal in the sense described in Section 2.2, has low rank. In particular, the entries on a given row of $A_0^{(j)}$ tend to be similar.

If we explicitly form the normal equations corresponding to the LSAI equation (2.6) we get

$$(3.1) \qquad \left[\hat{A}^{(j)} \hat{A}^{(j)} + A_0^{(j)\mathrm{T}} A_0^{(j)} \right] \hat{\mathbf{m}}^{(j)} = \hat{A}^{(j)} \mathbf{e} \,,$$

where we have used that $\hat{A}^{(j)}$ is symmetric. We argued, in the preceding paragraph, that the matrix $A_0^{(j)}$ has low rank and approximately constant rows. We now approximate $A_0^{(j)}$ using the rank-one approximation

$$(3.2) \qquad A_0^{(j)} \approx \mathbf{a}\mathbf{u}^{\mathrm{T}} \,,$$

where $\mathbf{u}$ is a column vector with k entries all equal to one and $\mathbf{a}$ is a column vector with $n-k$ unknown entries. Substituting (3.2) into (3.1) and multiplying with $\hat{A}^{(j)-1}$ from the left we arrive at

$$(3.3) \qquad \left[\hat{A}^{(j)} + ||\mathbf{a}||_2^2 \hat{A}^{(j)-1} \mathbf{u}\mathbf{u}^{\mathrm{T}} \right] \hat{\mathbf{m}}^{(j)} = \mathbf{e} \,.$$

This equation is just a simple rank-one modification of the DBAI equation (2.7). Once $\hat{A}^{(j)}$ is constructed and the scalar $||\mathbf{a}||_2$ is determined, the cost of solving (3.3) using a direct method and the Sherman-Morrison formula is the same

as the cost of solving the DBAI equation (2.7) to leading order in k. For good performance one can use LDL^T or Cholesky factorization for $\hat{A}^{(j)}$, depending on the situation. In our numerical examples we settle for LU-factorization.

We have already observed a close connection between the LSAI(k) and the DBAI(k) preconditioners; if one replaces the matrix $A_0^{(j)}$ with zeros in the LSAI equation (2.6) one recovers the DBAI equation (2.7). The DBAI(k) can, thus, be viewed as a special case of a generalized LSAI(k) where the 2-norm, entering the definition of the Frobenius norm if written in terms of individual matrix columns, is replaced by a weighted 2-norm where the k first weights are set to one and the remaining weights are set to zero. Perhaps the most efficient norm is somewhere in between these two extremes – more weight should be given to the upper equations of (2.6), corresponding to entries of AM that are close to the diagonal in the sense described in Section 2.2, and less weight should be given to the lower equations of (2.6)?

3.2   *A weighted 2-norm.*

Now define the weighted norm

$$(3.4) \qquad\qquad ||\mathbf{x}||_W = \sqrt{\mathbf{x}^\mathrm{T} W^\mathrm{T} W \mathbf{x}} \,,$$

$$(3.5) \qquad\qquad W_{ij} = \left\{ \begin{array}{ll} 0\,, & i \neq j\,, \\ 1/i\,, & i = j \leq k\,, \\ 1/k\,, & i = j > k\,, \end{array} \right.$$

and let WLSAI(k) denote a weighted LSAI(k) preconditioner obtained by minimizing the residual of (2.6) in the norm given by (3.4). The left images of Figure 3.1 show the spectra of $A_\mathrm{I} M_\mathrm{WLSAI}$, where A_I is the same as in Figure 2.1. A comparison with the left images of Figure 2.1 indicates that WLSAI(k) does achieve somewhat better clustering of the eigenvalues than does LSAI(k). This is particularly apparent for the lower left images involving $\mathcal{S}_\mathrm{dom}$.

We now proceed to modify the preconditioner implied by (3.3) as to incorporate the weighted norm. Starting with (2.6), using $||\cdot||_W$ for minimizing the residual, and repeating the steps of Section 3.1 we arrive at

$$(3.6) \qquad\qquad \left[\hat{A}^{(j)} + ||\mathbf{a}||_2^2 \hat{W}^{-2} k^{-2} \hat{A}^{(j)-1} \mathbf{u}\mathbf{u}^\mathrm{T} \right] \hat{\mathbf{m}}^{(j)} = \mathbf{e}\,,$$

where $\hat{W}$ is the $k \times k$ submatrix given by W_{ij}, $i, j \leq k$. Equation (3.6) will be used for computing the nonzero entries of our new sparse approximate inverse preconditioner, denoted WBAI(k). As for choosing the scalar $||\mathbf{a}||_2$, experiments showed that it should be a rapidly decaying function of k. We settled for

$$(3.7) \qquad\qquad ||\mathbf{a}||_2^2 = 4 \cdot (n-k) \cdot 10^{-\frac{k}{4\log_{10}(n)}} \,,$$

which is used in all examples involving WBAI(k) in this paper. The second image of Figure 3.1 shows the spectrum of $A_\mathrm{I} M_\mathrm{WBAI}$, where A_I is the same as in Figure 2.1. The spectrum of $A_\mathrm{I} M_\mathrm{WBAI}$ stretches further out into the imaginary
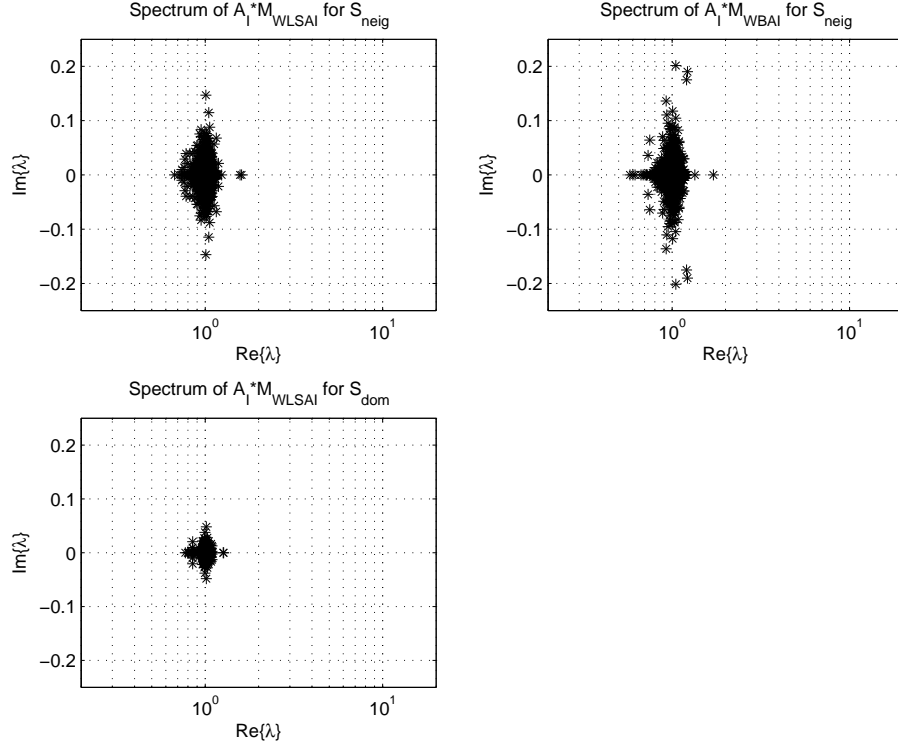
Figure 3.1: Same as in Figure 2.1, except for that the sparse approximate inverses of Section 3.2 are used. Note the difference in scales on the x- and y-axes.

plane than the other spectra, but the real parts of the eigenvalues are well clustered. Furthermore, a series of experiments show that the spectrum grows only slowly with increasing n. A comparison of the second image of Figure 3.1 with the first images of Figure 2.1 and Figure 3.1 shows that the clustering achieved by WBAI(k) is rather similar to those achieved by LSAI(k) and WLSAI(k). A comparison with the second image of Figure 2.1 shows that WBAI(k) achieves a much better clustering than does its chief competitor DBAI(k).

At this point we wish to make four remarks:

- The matrix W of (3.5), which gives most weight to the first equation and progressively less weight to the following equations of (2.6), is the same for all $\hat{A}^{(j)}$. This is so since the ordering of the rows of all $\hat{A}^{(j)}$ is the same with respect to the distances $|z_{q(i)} - z_j|$, see Section 2.3.

- The determination of the matrix W and the determination of the scalar $||\mathbf{a}||_2$ have to be done in tandem. We experimented with several weight matrices similar to W of (3.5), including such that took the distances $|z_{q(i)} - z_j|$ into account in a more elaborate fashion, in combination with

other expressions for the scalar $||\mathbf{a}||_2$, without observing any significant changes in performance. A simultaneously optimal choice for W and $||\mathbf{a}||_2$ would depend on both the particular system matrix and the strategy for determining $\mathcal{S}$. We settled for the choices (3.5) and (3.7) since they are reasonably simple and gave good performance for several classes of system matrices of the form (1.1), see Section 3.3.

- Minor changes in $\mathcal{S}$ have larger effect on the performance of WBAI(k) than minor changes in W and $||\mathbf{a}||_2$, see Section 3.4.

- The derivation of WBAI(k) involves the rank-one approximation (3.2). The unknown vector $\mathbf{a}$ is never determined entry-wise, but enters into the scheme only via its Euclidean norm. Constant block approximation of off-diagonal matrix-blocks that minimize the distance for the energy norm to an associated discrete nonoscillatory Green's functions have been used in the quite opposite context of finding full block-constant approximate inverses to certain sparse matrices. See Guillaume, Huard, and Calvez [11] for such an example where the number of iterations needed to meet a given residual, after preconditioning, increases as $O(n^{0.25})$.

3.3 Further numerical examples.

Figure 3.2 illustrates the asymptotic efficiency of using our new preconditioner WBAI(k) when iteratively solving linear systems involving the matrix A_{I}. The left image should be compared to the results with LSAI(k) and DBAI(k) of Figure 2.2. The difference between WBAI(k) and DBAI(k) is striking. Especially for $k \geq 20$. The number of full GMRES iterations with WBAI(k) grows as $O(\log(n))$, or slower. For $k = 20$ and $n = 1358104$, the number of full GMRES iterations decreases from 72 to 17 as we switch from DBAI(k) to WBAI(k), while the setup cost, the computational cost per iteration, and the achieved accuracy are unchanged. The difference between WBAI(k) and LSAI(k), on the other hand, is smaller. That is, for all n where LSAI data are available. Upon close comparison of the left image of Figure 3.2 and the left image of Figure 2.2 one can see a tendency for WBAI(k) to perform better than LSAI(k) at large n. It thus seems as if we have succeeded in making a new preconditioner combining, or even improving on, the best features of LSAI(k) and DBAI(k).

The right image of Figure 3.2 contains error estimates: reference solutions $\mathbf{x}_{\mathrm{ref}}$ for various system sizes n are obtained by iteratively solving unpreconditioned systems $A_{\mathrm{I}}\mathbf{x} = \mathbf{b}$ with a stopping criterion threshold of 10^{-13} in the relative residual. Restarts are used whenever the GMRES solver stagnates. The solution $\mathbf{x}_{\mathrm{ref}}$, for a given n, is then used as "exact" solution for the purpose of computing error estimates for the other five solutions, obtained with a stopping criterion threshold of 10^{-8} and no restarts. Given the approximate spectral information of (2.2) and considering that $\mathbf{b}$ is random, one could, perhaps, make the assumption that $\mathbf{x}_{\mathrm{ref}}$ has better quality than the other five solutions for a given n. If this holds, then the right image of Figure 3.2 indicates that the
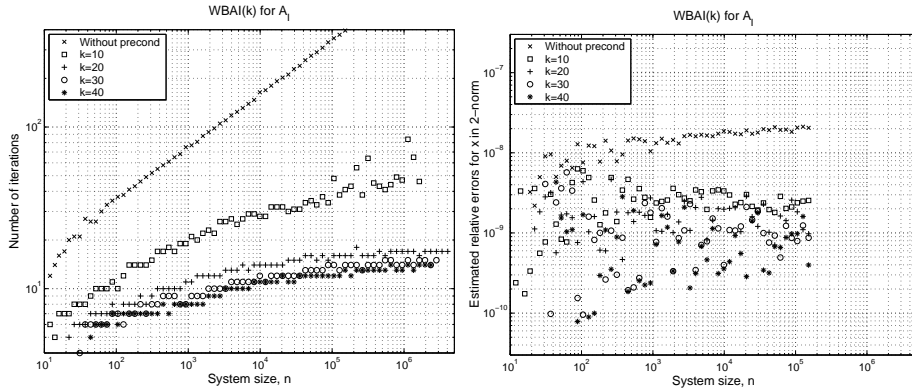
Figure 3.2: Left, same as in Figure 2.2 except for that WBAI(k) preconditioning is used. Right, estimated relative errors in the computed solutions $\mathbf{x}$. Reference solutions for $\mathbf{x}$ are obtained by solving unpreconditioned systems with a stopping criterion threshold of 10^{-13}.

quality of the solutions obtained from the preconditioned systems is at least as good as that of the solutions obtained from the unpreconditioned system.

We now compare the DBAI(k) and the WBAI(k) preconditioners on two other classes of system matrices of the form (1.1). One class of matrices is obtained by letting the points z_i no longer be completely randomly placed in the unit square, but taken as the centers of ellipses in a certain polydisperse suspension. The ellipses vary in relative size and eccentricity within certain limits. See Section 7 of Ref. [13] for details on how the polydisperse suspension is generated. The numbers r_i are still randomly selected in $(0, 0.5d_i]$. We denote a matrix with these characteristics A_{II}. The other class of matrices is obtained by keeping the same strategy for placing the points z_i as for A_{II}, but changing the mechanism for determining the numbers r_i. These numbers now lie in an interval approximately given by $[0.12d_i, 0.59d_i]$ and they are no longer independent random variables, but exhibit a complicated type of dependence. See Section 5.3 of Ref. [13] for details on how r_i are computed. We denote a matrix with these characteristics A_{III}. The matrix A_{III} is the matrix denoted F in eq. (36) of Ref. [13], multiplied with minus one.

It is hard to say which of the matrices A_{I}, A_{II}, and A_{III} is the most difficult to precondition. As we shall see, when used as system matrices and without preconditioning, they all behave similarly in our test. Still, there are major differences in their characteristics: The diagonal entries in A_{I} and A_{II} can be arbitrarily large while the diagonal entries in A_{III} are bounded for a fixed n. Entries close to the diagonal in A_{I} can be arbitrarily large while entries close to the diagonal in A_{II} and A_{III} are bounded for a fixed n. Numerical experiments suggest that the matrices A_{I} and A_{II} are positive definite while the matrix A_{III} is indefinite. Jou, Leo, and Lowengrub in Section 2.1.1 of Ref. [14] comment on the conditioning of a system of linear equations where, in a sense, a matrix of the form (1.1) is the leading matrix. These authors report that small r_i and small
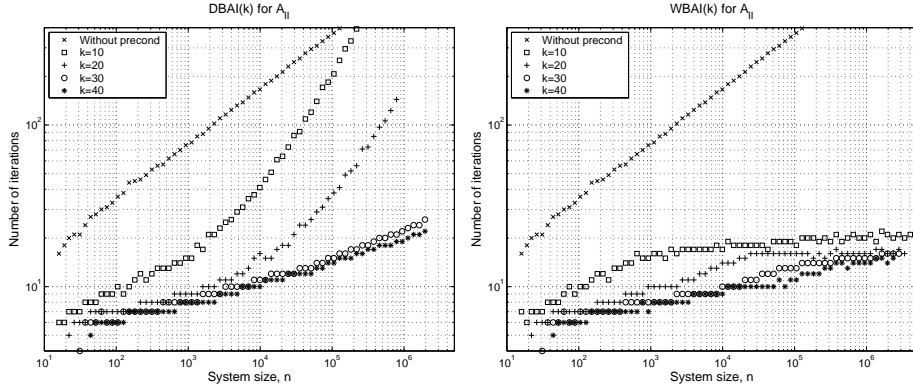
Figure 3.3: Number of full GMRES iterations needed to solve systems $A_{\mathrm{II}}\mathbf{x} = \mathbf{b}$ with a stopping criterion threshold of 10^{-8} in the relative residual. The components of the right-hand side vectors $\mathbf{b}$ are chosen randomly in $[-1, 1]$. The sparsity pattern is $\mathcal{S}_{\mathrm{neig}}$. Left, DBAI($k$) preconditioning. Right, WBAI(k) preconditioning.
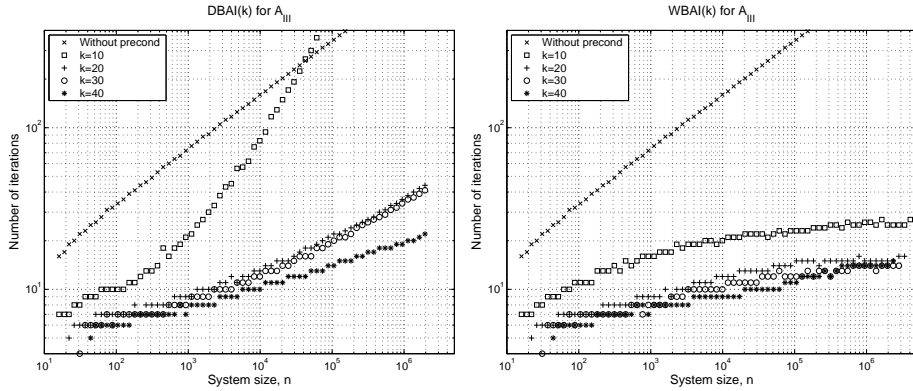

Figure 3.4: Same as in Figure 3.3, except for that A_{II} is replaced with A_{III}.

distances between some points z_i increase the condition number of their entire system matrix for a fixed n. Perhaps then the matrix A_{I} is the most difficult to deal with and the matrix A_{III} is the easiest.

Figure 3.3 compares the asymptotic efficiency of using DBAI(k) to that of using WBAI(k) when iteratively solving linear systems involving the matrix A_{II}. We note that DBAI(k) runs into great difficulties with A_{II} for large n and $k \leq 20$, much greater than with A_{I}. Our new preconditioner WBAI(k), on the other hand, performs slightly better than with A_{I}, for which it was constructed. This seems very promising since our objective is to find an efficient sparse approximate inverse for a general matrix of the form (1.1).

Figure 3.4 compares the asymptotic efficiency of using DBAI(k) to that of using WBAI(k) when iteratively solving linear systems involving the matrix A_{III}.
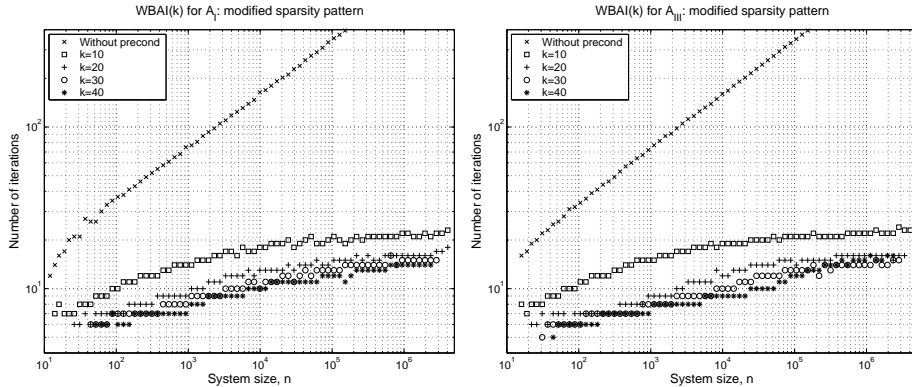
Figure 3.5: Same as in Figure 3.2, left image, and Figure 3.4, right image, except for that the modified $\mathcal{S}_{\mathrm{neig}}$ of Section 3.4 is used.

The left image shows that the number of iterations for DBAI(k) depends on k in an irregular way. The right image shows, yet again, that our new preconditioner WBAI(k) is more efficient than DBAI(k) for large n and especially so for small k, which are of greatest interest in applications.

3.4   Minor changes in the sparsity pattern $\mathcal{S}_{\mathrm{neig}}$.

The sparsity pattern $\mathcal{S}_{\mathrm{neig}}$ associated with matrices of the form (1.1) is not optimal. We would rather use the dominant sparsity pattern $\mathcal{S}_{\mathrm{dom}}$, could it only be constructed in reasonable time for large systems. A close inspection of $\mathcal{S}_{\mathrm{dom}}$ for various realizations of A_{I} reveals that, for a given j, index pairs (i,j) of $\mathcal{S}_{\mathrm{dom}}$ correspond to the points z_i with smallest distances $|z_i - z_j|$ subjected in some sense to a condition that z_i are evenly distributed around z_j.

In order to construct a sparsity pattern with a low setup cost which better mimics the observed $\mathcal{S}_{\mathrm{dom}}$ we made a mild modification of $\mathcal{S}_{\mathrm{neig}}$. In brevity, among the k index pairs (i,j) for a given j we scrutinized those five pairs corresponding to the largest values of the distance $|z_i - z_j|$. Under certain conditions we replaced some of these pairs with new pairs corresponding to points z_i with slightly larger values of $|z_i - z_j|$ so that the total number of index pairs for a given j still was k and so that that the points z_i corresponding to these pairs were more evenly distributed around z_j. We omit the details of this *ad hoc* procedure and turn our attention to Figure 3.5 which displays results from test involving this modification of $\mathcal{S}_{\mathrm{neig}}$, the preconditioner WBAI(k), and the matrices A_{I} and A_{III}. Comparison with the left image of Figure 3.2 and the right image of Figure 3.4 shows a decrease in the number of iterations needed for convergence for the smallest bandwidth $k = 10$. The improvement is particularly large for the matrix A_{I}. The number of necessary iterations here seems to decrease with a factor of three for system sizes $n \geq 10^6$.

3.5 Notes on the implementation.

All numerical experiments in this paper were performed on a regular Sun Blade 100 workstation. The experiments involving spectra were done in MATLAB. The other experiments were done with codes implemented in Fortran 77. See Section 4.6 of Ref. [8] for some details of the implementation of the FMM [10]. The internal tolerance in the FMM was set to 10^{-13} in all experiments.

4 Discussion.

We set out to find an efficient right sparse approximate inverse preconditioner of bandwidth k for dense $n \times n$ matrices A of the form (1.1) with good asymptotic properties in the limit of large n. The preconditioner should be broadly useful, that is, it should be efficient for a wide range of choices of the points z_i and the numbers r_i. The preconditioner should be simple to use and free from any user specified parameter apart from k. We chose to develop the preconditioner in a GMRES setting, well aware of that for particular system matrices SQMR or conjugate gradients might be more efficient iterative solvers. See [3] for details on the extra work that has to be invested in the preconditioner to make it compatible with a symmetric Krylov solver.

The idea was to construct a mesh-neighbour based LSAI type preconditioner at a small setup cost. For this, we took advantage of the special structure of matrices of the form (1.1). Inner products of various vectors corresponding to distant potential fields originating at closely spaced sources were approximated by identical constants, determined by a number $||a||_2$, thus reducing the setup cost to that of a DBAI type preconditioner. We also observed that all equations entering into the overdetermined linear systems used for computing the columns of the preconditioner were not equally important. As a consequence we used a weighted norm for minimization of the residuals.

During the development of our preconditioner we relied on a class of matrices where the points z_i were completely randomly placed in the unit cell and where the numbers r_i were independent random variables with uniform density in $(0, 0.5d_i]$. Based on a large set of experiments, not presented in the paper, we arrived at the formula (3.7) for $||a||_2$. We named our preconditioner WBAI(k) and tested it on three classes of matrices with various strategies for choosing z_i and r_i, and compared with results using DBAI(k). In all cases, huge improvement over DBAI(k) was found and especially so for large n and small k, which is the situation most relevant for large-scale applications. The numerical experiments showed that for $k \geq 20$, or even $k \geq 10$ together with a modified sparsity pattern, the number of full, that is, not restarted, GMRES iterations needed to reach an estimated relative error in 2-norm of less than 10^{-8} in the solution $\mathbf{x}$ to a system $A\mathbf{x} = \mathbf{b}$, where $\mathbf{b}$ is random, often grew as $O(\log(n))$. In some cases the number of iterations grew even slower, making the properties of the system $A\mathbf{x} = \mathbf{b}$, after preconditioning, resemble those of a discretized Fredholm second kind integral equation. If A of the form (1.1) is a system matrix originating from the discretization of a Fredholm first kind equation (with singular kernel) one

could say that we have found an affordable way of regularizing that equation numerically.

There exists a rich literature on experimental results regarding the efficiency of various sparse approximate inverse preconditioners and iterative solvers for various dense linear systems. A particular feature of the present work is that it presents systematic and highly resolved studies of the dependence of speed and estimated accuracy on system size n. Sometimes the differences between preconditioners become pronounced only at large n. Their behaviour for small n may not be indicative of their large-scale behaviour. Our examples involve system matrices with dimensions up to $n = 4055271$. As for further work, it would be interesting to study the effect of approximating the far field in LSAI type preconditioners in some three dimensional, or other, context where the Green's function decays more rapidly than in the present study.

REFERENCES

1. E. Bängtsson and M. Neytcheva, *Algebraic preconditioning versus direct solvers for dense linear systems as arising in crack propagation problems*, Comm. Numer. Meth. Engng, 21 (2005), pp. 73–81.

2. M. Benzi, *Preconditioning Techniques for Large Linear Systems: A Survey*, J. Comput. Phys., 182 (2002), pp. 418–477.

3. B. Carpentieri, I. S. Duff, and L. Giraud, *Sparse pattern selection strategies for Robust Frobenius-norm minimization preconditioners in electromagnetism*, Numer. Linear Algebra Appl., 7 (2000), pp. 667–685.

4. B. Carpentieri, I. S. Duff, L. Giraud, and M. M. M. Made, *Sparse symmetric preconditioners for dense linear systems in electromagnetism*, Numer. Linear Algebra Appl., 11 (2004), pp. 753–771.

5. B. Carpentieri, I. S. Duff, L. Giraud, and G. Sylvand, *Combining fast multipole techniques and an approximate inverse preconditioner for large electromagnetism calculations*, SIAM J. Sci. Comput., 27 (2005), pp. 774–792.

6. K. Chen, *An analysis of sparse approximate inverse preconditioners for boundary integral equations*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1058–1078.

7. E. Chow, *A priori sparsity patterns for parallel sparse approximate inverse preconditioners*, SIAM J. Sci. Comput., 21 (2000), pp. 1804–1822.

8. J. Englund and J. Helsing, *Stress computations on perforated polygonal domains*, Engng Anal. Boundary Elem., 27 (2003), pp. 533–546.

9. J. M. Ford and E. E. Tyrtyshnikov, *Combining Kronecker product approximation with discrete wavelet transform to solve dense, function related linear systems*, SIAM J. Sci. Comput., 25 (2003), pp. 961–981.

10. L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

11. P. H. Guillaume, A. Huard, and C. Le Calvez, *A block constant approximate inverse for preconditioning large linear systems*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 822-851.

12. W. Hackbusch, *A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: introduction to $\mathcal{H}$-matrices*, Computing, 62 (1999), pp. 89-108.

13. J. Helsing and E. Wadbro, *Laplace's equation and the Dirichlet–Neumann map: a new mode for Mikhlin's method*, J. Comput. Phys., 202 (2005), pp. 391–410.

14. H.-J. Jou, P.H. Leo, and J.S. Lowengrub, *Microstructural evolution in inhomogeneous elastic media*, J. Comput. Phys., 131 (1997), pp. 109–148.

15. S. Kapur and D. E. Long, *Large-scale capacitance calculation*, in Proc. 37th Design Automation Conf., June 2000, pp. 744-749.

16. J.-F. Lee, R. Lee, R. J. Burkholder, *Loop star basis functions and a robust preconditioner for EFIE scattering problems*, IEEE Trans. Antennas and Propagation, 51 (2003), pp. 1855–1863.

17. P. G. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, J. Comput. Phys., 205 (2005), pp. 1–23.

18. K. Nabors, S. Kim, and J. White, *Fast Capacitance Extraction of General Three-Dimensional Structures*, IEEE T. Microw. Theory., 40 (1992), pp. 1496–1506.

19. K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White, *Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory*, SIAM J. Sci. Comput., 15 (1994), pp. 713–735.

20. M. Nilsson, *A Fast Multipole Accelerated Block Quasi Minimum Residual Method for Solving Scattering from Perfectly Conducting Bodies*, in Proc. IEEE AP-S Int. Symp. Vol. 4, Magdy F. Iskander, ed., Salt Lake City, Utah, 2000, pp. 1848–1851.

21. Y. Saad and M.H. Schultz, *GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856–869.

22. S. A. Vavasis, *Preconditioning for boundary integral equations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 905–925.

23. K. G. Wang, M. E. Glicksman, and K. Rajan, *Modeling and simulation for phase coarsening: A comparison with experiment*, Phys. Rev. E, 69 (2004): Art. No. 061507 Part 1.

24. W. Yu, Z. Wang, X, Hong, *Preconditioned multi-zone boundary element analysis for fast 3D electric simulation*, Engng Anal. Boundary Elem., 28 (2004), pp. 1035–1044.