



A fast BIE iteration method for an arbitrary body in a flow of incompressible inviscid fluid

Antonio Scalia^a, Mezhlum A. Sumbatyan^{b,*}, Vitaly Popuzin^a

^a D.M.I., University of Catania, 95125 Viale A. Doria 6 (CT), Italy

^b Faculty of Mathematics, Mechanics and Computer Science, Southern Federal University, Milchakova Street 8a, 344090-Rostov-on-Don, Russia

ARTICLE INFO

Article history:

Received 23 October 2011

Received in revised form 28 April 2012

Keywords:

Boundary integral equations

Iteration algorithm

Töeplitz matrix

Complex-shape objects

ABSTRACT

The paper is concerned with the new iteration algorithm to solve boundary integral equations arising in boundary value problems of mathematical physics. The stability of the algorithm is demonstrated on the problem of a flow around bodies placed in the incompressible inviscid fluid. With a discrete numerical treatment, we approximate the exact matrix by a certain Töeplitz one and then apply a fast algorithm for this matrix, on each iteration step. We illustrate the convergence of this iteration scheme by a number of numerical examples, both for hard and soft boundary conditions. It appears that the method is highly efficient for hard boundaries, being much less efficient for soft boundaries.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

The Boundary Integral Equation (BIE) method is one of the most powerful techniques in applied mathematics, mechanics and physics. The basic advantage of this method is that such equations are more tractable from the mathematical point of view than the original differential equations. Typically, with a certain appropriate discretization these equations may be reduced to some linear algebraic systems whose matrix has a dense structure. Therefore, if one arranges a grid with N nodes then the matrix has N^2 nonzero elements, and classical approaches require $O(N^3)$ arithmetic operations that are too numerically expensive even when implemented on modern computers. Just for this reason there have been proposed recently some new “fast” methods. As a rule, they are based on some special numerical techniques connected with a representation of big data arrays by relatively small number of parameters in an “easy-to-use” form. Such approaches are related to some approximation methods which are in active use in statistics, speech recognition, image processing etc. In frames of such approaches, the computation time required to solve systems of linear algebraic equations generated by BIE is linear instead of quadratic or cubic in classical methods that is a big advantage when solving problems of huge mesh dimension.

There exist two general ways to solve this type of problems. The first of them is based on the well-known iterative solvers (such as conjugate gradients or GMRES), in a combination with fast matrix–vector multiplication methods on each iteration step, in order to accelerate the algorithm. The second way is to solve the problem directly. Another approach is founded upon inversion of the basic matrix. It can be done either directly or iteratively (for example, by using any version of Newton’s iterations). Then this matrix is used as a pre-conditioner for iterative solvers or directly as a product of the inverse matrix and the right-hand side of the equation.

The iteration algorithms require calculation of a matrix–vector product on each step of iterations. In the general case of matrices with dense structure it can be done in $O(N^2)$ operations with matrix storage of $O(N^2)$. Therefore, to achieve

* Corresponding author.

E-mail addresses: scalia@dmict.unict.it (A. Scalia), sumbat@math.rsu.ru (M.A. Sumbatyan), popuzin@dmict.unict.it (V. Popuzin).

a fast solver, one needs to apply any fast matrix–vector multiplication method. Some basic ideas of this sort have been proposed in [1,2] in 1987 that is known now as a fast multipole method (FMM). This method permits efficient matrix–vector multiplication in $O(N)$ operations with $O(N)$ storage dimension. Similar method was also developed in [3], this is known as a panel clustering method. The other, interpolation method, reduces the problem to a dense matrix of a smaller regular grid. The constructed matrix in this case is Töeplitz or block-Töeplitz one, for which there exist some fast matrix–vector multiplication algorithms based on the Fast Fourier Transform. This idea was considered by Nechepurenko in 1985 [4]. Further development in this direction was performed in [5], who applied these ideas to irregular grids. Other powerful methods are based upon wavelet analysis, when the initial matrix transforms to a pseudo-sparse form (or T -sparse, by A. Harten definition [6,7] which means that the matrix is sparse if one neglects its small elements). This idea is currently under active development [8,9]. Some other approaches are based upon a certain approximation of the initial matrix by a matrix of block structure, with the matrix rank equal to 1. This method, proposed by Tyrtysnikov in 1993 [10–12], is known as the mosaic-skeleton method. The basic ideas of this approach can also be found in the work of Voevodin [13].

All above methods are based upon a special decomposition of the initial matrix. Depending on the structure of the operator, as well as on disposition of the “source” and the “observation point”, the elements of the initial matrix are divided into two groups related to the so-called “long-range” and “short-range” interactions. Typically, the elements located near the principal diagonal correspond to “short-range” interaction, just these elements are associated with the singularities. All operations with such blocks are performed in a standard way. The matrix blocks, out of the diagonal, connected with the “long-range” action zone, may be approximated by some matrices of a specific structure. The algorithms discussed above operate just with these blocks.

The first attempts in this research direction were founded on some special analytical decomposition of the kernel in the considered integral equation. The evident lack of such an approach is that there is a need to construct a specific computational code for every specific kernel. Besides, some types of kernels do not permit the required decomposition. Recently, there have been created the algorithms which operate only with the elements of the matrix and do not use any information about the kernel [14,15]. In the present paper we use one of such methods, namely, “adaptive cross approximation” (ACA algorithm) [16]. In this algorithm the elements out of the principal diagonal are approximated by matrices of small rank. In practice, we used the C++ library HLibPro, to realize a fast matrix–vector multiplication by the ACA method [17–20].

The above survey of the modern fast matrix methods implies numerous applications of new efficient algorithms in many branches of applied mathematics and mechanics. In particular, in the present paper we propose a new iteration algorithm which is demonstrated on the problem of a flow around bodies placed in the incompressible inviscid fluid. With a discrete numerical treatment, we approximate the exact matrix by a certain Töeplitz one and then apply a fast algorithm for this matrix, on each iteration step. We illustrate the convergence of this iteration scheme by a number of numerical examples, both for hard and soft boundary conditions. It appears that the method is highly efficient for hard boundaries (indeed, the most natural case), being much less efficient for soft bodies (which are of less interest in practice).

2. Problem formulation and the basic BIE

Let us demonstrate the method we propose on the 2d problem about an arbitrary body placed in a flow of incompressible inviscid fluid. To be more specific, the flow is assumed stationary in time and uniform at infinity. Then the classical Boundary Element Method (BEM), or Boundary Integral Equation (BIE) method can be formulated here either in terms of velocity potential $\varphi(y)$, $y = (y_1, y_2)$, $\vec{v} = \text{grad } \varphi$, or in terms of stream function $\psi(y)$. Again, to be more specific, we treat the problem by using function φ . Then, the classical theory [21,22] determines potential φ' to the perturbed value of the velocity vector \vec{v} , at arbitrary point $y_0 = (y_1^0, y_2^0)$ in the flow, through the integral taken over the boundary contour l to the body under consideration:

$$\varphi'(y_0) = \int_l \left[\varphi(y) \frac{\partial G(y, y_0)}{\partial n_y} - \frac{\partial \varphi(y)}{\partial n_y} G(y, y_0) \right] dl_y. \quad (2.1)$$

If the boundary of the body is absolutely hard: $\partial \varphi / \partial n_y|_l = 0$ then (2.1) is simplified to

$$\varphi'(y_0) = \int_l \varphi(y) \frac{\partial G(y, y_0)}{\partial n_y} dl_y. \quad (2.2)$$

Here $\varphi = \varphi' + \varphi_\infty$, Green's function is $G(y, y_0) = -\ln r / (2\pi)$, $r = |y - y_0| = [(y_1 - y_1^0)^2 + (y_2 - y_2^0)^2]^{1/2}$, \vec{n}_y is the outer unit normal vector to contour l , $\varphi_\infty(y) = v_0(y_1 \cos \alpha + y_2 \sin \alpha)$, where α is the angle between axis y_1 and flow's velocity vector at infinity \vec{v}_0 .

It should be noted that representation (2.2) is valid only in the case when the velocity circulation is zero. We thus leave aside all questions related to the Kutta–Zhukovsky hypothesis if the body has a sharp edge on its boundary.

Now, by setting point x approaching to the boundary line from the fluid: $y_0 \rightarrow x \in l$, and taking into account the standard properties of the double layer potential [21,22], Eq. (2.2) is directly reduced to the following second-kind BIE of Fredholm type:

$$\frac{\varphi(x)}{2} - \int_l \varphi(y) \frac{\partial G(y, x)}{\partial n_y} dl_y = f(x), \quad x \in l, \quad (2.3a)$$

$$f(x) = \varphi_\infty(x) = v_0(x_1 \cos \alpha + x_2 \sin \alpha). \quad (2.3b)$$

Eq. (2.3) is valid for smooth contours only, but it can easily be rewritten in an appropriate form if there is a sharp edge on contour l [21].

The natural way to arrange a direct numerical treatment of the basic BIE (2.3) is to apply the so-called *collocation technique* [22]. One may arrange a dense set of nodes $x_j \in l$, $j = 1, \dots, N$, distributed over contour l along its full length: the same nodes for “internal” variable $y \in l$ and “external” variable $x \in l$. One then can associate the nodes for variable y with subscripts j , x_j ; and for variable x with subscripts i , x_i . If we approximate the integral in (2.3a) by a quadrature formula, the simplest one is indeed to put the integrand approximately constant over each elementary arc, then this leads to the following LAS (linear algebraic system):

$$C\varphi = f, \quad \sim \sum_{j=1}^N c_{ij}\varphi(x_j) = f_i, \quad i = 1, \dots, N; \quad f_i = v_0(x_1^i \cos \alpha + x_2^i \sin \alpha), \quad (2.4a)$$

$$c_{ii} = \frac{1}{2}; \quad j \neq i: \quad c_{ij} = -\frac{\partial G(x_j, x_i)}{\partial n_y} l_j = \frac{(x_1^j - x_1^i)n_1^j + (x_2^j - x_2^i)n_2^j}{(x_1^j - x_1^i)^2 + (x_2^j - x_2^i)^2} \frac{l_j}{2\pi}, \quad (2.4b)$$

where l_j denotes the length of the elementary arc. It should be noted that the integral in (2.3a), after discretization, gives a certain contribution to the diagonal terms c_{ii} . It is well known that this is defined by the value of the curvature of the boundary line l at node x_i . However, this contribution contains again a small factor l_j , being asymptotically small at $N \rightarrow \infty$, when compared with the term $c_{ii} = 1/2$ given by that present outside the integral.

3. The essence of the new fast numerical algorithm

Typically, LAS (2.4) can be treated directly, by the Gauss elimination process. This requires $O(N^3)$ arithmetic operations [23]. In the meantime, there is a case when the matrix $\{c_{ij}\}$ has a specific structure. We mean the case when the body placed in the flow is a circle, then with the set of the nodes, arranged with a uniform step h_θ over the polar angle, the matrix elements become $(x_1(\theta) = a \cos \theta, x_2(\theta) = a \sin \theta)$:

$$\begin{cases} x_1^j = a \cos \theta_j, \\ x_2^j = a \sin \theta_j, \end{cases} \quad \begin{cases} n_1^j = \cos \theta_j, \\ n_2^j = \sin \theta_j, \end{cases} \quad \begin{cases} \theta_j = \left(j - \frac{1}{2}\right) h_\theta, \quad h_\theta = 2\pi/N, \\ l_j = a h_\theta, \quad j = 1, \dots, N, \end{cases} \quad (3.1)$$

$$(x_1^j - x_1^i)n_1^j + (x_2^j - x_2^i)n_2^j = a[1 - \cos(\theta_j - \theta_i)] = a[1 - \cos[(i - j)h_\theta]],$$

$$(x_1^j - x_1^i)^2 + (x_2^j - x_2^i)^2 = 2a^2\{1 - \cos[(i - j)h_\theta]\}.$$

Therefore, the elements of the matrix depend on the difference of subscripts $(i - j)$, i.e. the matrix is of convolution type or, by other words, this is a Töeplitz matrix. Note that in this particular case of hard boundary the matrix is even degenerated since the numerator and the denominator cancel to each other. However, the Töeplitz structure of the matrix remains valid in all problems for round geometries, including with full (not only degenerated) structure studied in Section 4.

The principal idea of the method we propose here is to arrange, in the case of arbitrary geometry of the body, an iterative process when at each iteration step there is a need to solve a certain LAS with a Töeplitz matrix. This idea looks very attractive since the solution to Töeplitz LAS can be constructed more efficiently than in the case of matrices of general structure. More details of this property and existing approaches are discussed in Section 5.

Let us come back to the algorithm we propose. The required Töeplitz matrix, for arbitrary shape of boundary contour l , can be constructed if we change in matrix c_{ij} (2.4) all elements on every diagonal parallel to the principal one by the average value of all elements situated over this chosen diagonal. Such an approach yields a new matrix $\{c_{ij}^t\}$ with the elements:

$$c_{ij}^t = d_{i-j}, \quad d_{k+n-1,k} = \frac{\sum_{j=1}^{N-n+1} c_{j+n-1,j}}{N-n+1}, \quad n = 1, \dots, N; \quad k = 1, \dots, N-n+1; \quad (3.2)$$

$$d_{k,k+n-1} = \frac{\sum_{j=1}^{N-n+1} c_{j,j+n-1}}{N-n+1}, \quad n = 2, \dots, N; \quad k = 1, \dots, N-n+1.$$

This formal definition is clearly demonstrated here:

$$C = \begin{pmatrix} \bullet & \bullet & \bullet & e_1 & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & e_2 & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \ddots & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & e_{m-1} & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & e_m \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix} \xRightarrow{d = \frac{\sum_{j=1}^m e_j}{m}} C^t = \begin{pmatrix} \bullet & \bullet & \bullet & d & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & d & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \ddots & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & d & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & d \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix}.$$

With the so constructed Töeplitz matrix C^t let us represent LAS (2.4) in the equivalent form:

$$C^t \varphi = f + (C^t - C)\varphi. \quad (3.3)$$

This allows us to organize the iteration process

$$C^t \varphi_{q+1} = f + (C^t - C)\varphi_q, \quad q = 0, 1, \dots, \quad (3.4)$$

where $\varphi_0 = 0$ is accepted as the initial iteration step, which is obviously to solve the LAS with the Töeplitz matrix on the left-hand side and the true right-hand side.

Numerical experiments performed show that the rate of convergence is independent of the number of nodes, but is quite sensitive to the geometry of the boundary line l . This is quite natural from the physical point of view. In fact, the number of nodes must increase also in standard approaches of direct treatment if the geometry of the object becomes more complicated.

Generally speaking, for bodies of arbitrary complex geometry it is not so easy to give a strict proof of the convergence for the proposed iteration algorithm. However, this can be done for some particular classes of boundary contours. For example, let us restrict the consideration by the class of elliptic domains:

$$\begin{aligned} x_1^j &= a \cos \theta_j, & x_2^j &= b \sin \theta_j, & (a > b); & & l_j &= h_\theta \sqrt{a^2 \sin^2 \theta_j + b^2 \cos^2 \theta_j}, \\ n_1^j &= \frac{b \cos \theta_j}{\sqrt{a^2 \sin^2 \theta_j + b^2 \cos^2 \theta_j}}, & n_2^j &= \frac{a \sin \theta_j}{\sqrt{a^2 \sin^2 \theta_j + b^2 \cos^2 \theta_j}}, \\ \theta_j &= \left(j - \frac{1}{2}\right) h_\theta, & h_\theta &= 2\pi/N, & j &= 1, \dots, N. \end{aligned} \quad (3.5)$$

Then the elements of matrix C in (2.4) become

$$\begin{aligned} c_{ii} &= \frac{1}{2}; & j \neq i: & c_{ij} = \frac{(x_1^j - x_1^i)n_1^j + (x_2^j - x_2^i)n_2^j}{(x_1^j - x_1^i)^2 + (x_2^j - x_2^i)^2} \frac{l_j}{2\pi} \\ &= \frac{a(\cos \theta_j - \cos \theta_i)b \cos \theta_j + b(\sin \theta_j - \sin \theta_i)a \sin \theta_j}{a^2(\cos \theta_j - \cos \theta_i)^2 + b^2(\sin \theta_j - \sin \theta_i)^2} \frac{h_\theta}{2\pi} \\ &= \frac{ab h_\theta}{4\pi \left[(a^2 - b^2) \sin^2 \frac{\theta_i + \theta_j}{2} + b^2 \right]} = \frac{ab h_\theta}{4\pi \left[(a^2 - b^2) \sin^2 \frac{(i+j-1)h_\theta}{2} + b^2 \right]}, \end{aligned} \quad (3.6)$$

therefore, Eq. (3.2) implies

$$\begin{aligned} C^t &= \frac{1}{2}(I + D), & d_{k+n-1,k} &= \frac{ab h_\theta}{2\pi(N-n+1)} \sum_{j=1}^{N-n+1} \frac{1}{(a^2 - b^2) \sin^2 \frac{(2j+n-2)h_\theta}{2} + b^2}, \\ d_{k,k+n-1} &= d_{k+n-1,k} \quad (n = 2, \dots, N; k = 1, \dots, N-n+1); & d_{ii} &= 0, \end{aligned} \quad (3.7)$$

where D is evidently a Töeplitz matrix. Such a representation allows us to rewrite Eq. (3.3) as a second-kind matrix operator equation:

$$(I + D)\varphi = 2f + 2(C^t - C)\varphi, \implies \varphi = F + 2(I + D)^{-1}(C^t - C)\varphi, \quad F = 2(I + D)^{-1}f. \quad (3.8)$$

Let us estimate the elements of matrix D in (3.7) in the class of ellipses of small eccentricity, i.e. with $a = (1 + \varepsilon)b$, $\varepsilon \ll 1$, when the ellipse is very close to a circle. Let us keep in all expressions only linear terms over parameter ε . Then one deduces from (3.7):

$$\begin{aligned}
d_{k,k+n-1} &= d_{k+n-1,k} = \frac{h_\theta}{2\pi(N-n+1)} \sum_{j=1}^{N-n+1} \frac{2ab}{(a^2+b^2) - (a^2-b^2)\cos[(2j+n-2)h_\theta]} \\
&\approx \frac{h_\theta}{2\pi(N-n+1)} \sum_{j=1}^{N-n+1} \frac{1+\varepsilon}{1+\varepsilon\{1-\cos[(2j+n-2)h_\theta]\}} \\
&\approx \frac{h_\theta}{2\pi(N-n+1)} \sum_{j=1}^{N-n+1} \{1+\varepsilon\cos[(2j+n-2)h_\theta]\} \\
&= \frac{h_\theta}{2\pi} \left\langle 1 + \frac{\varepsilon}{N-n+1} \left\{ \frac{\sin[(N-n+2)h_\theta]}{\sin h_\theta} \cos[(N-n+1)h_\theta] - \cos[(n-2)h_\theta] \right\} \right\rangle \\
&= \frac{1}{N} \left\langle 1 + \frac{\varepsilon}{N-n+1} \left\{ \frac{1}{2} - \frac{\sin[2\pi(2n-3)/N]}{2\sin(2\pi/N)} - \cos[2\pi(n-2)/N] \right\} \right\rangle, \tag{3.9}
\end{aligned}$$

where we have used a tabular value of a finite sum for trigonometric functions [24].

For all estimates we introduce the standard normalized space l_∞ of dimension N , where the norms for arbitrary vector $c = \{c_j\}$ and arbitrary matrix $C = \{c_{kj}\}$ are, respectively [25,26]

$$\|c\| = \max_{1 \leq j \leq N} |c_j|, \quad \|C\| = \max_{1 \leq k \leq N} \sum_{j=1}^N |c_{kj}|. \tag{3.10}$$

Then

$$\begin{aligned}
\sum_{j=1}^N |d_{kj}| &= \sum_{n=2}^{N+1-k} |d_{k,k+n-1}| + \sum_{n=2}^k |d_{k,k-n+1}| = \sum_{n=2}^{N+1-k} |d_{k,k+n-1}| + \sum_{n=2}^k |d_{k+n-1,k}| \\
&= \frac{1}{N} \left(\sum_{n=2}^{N+1-k} + \sum_{n=2}^k \right) \left| 1 + \frac{\varepsilon}{N-n+1} \left\{ \frac{1}{2} - \frac{\sin[2\pi(2n-3)/N]}{2\sin(2\pi/N)} - \cos[2\pi(n-2)/N] \right\} \right|. \tag{3.11}
\end{aligned}$$

Let us notice that for $N \gg 1$

$$\begin{aligned}
&\left| \frac{\varepsilon}{N-n+1} \left\{ \frac{1}{2} - \frac{\sin[2\pi(2n-3)/N]}{2\sin(2\pi/N)} - \cos[2\pi(n-2)/N] \right\} \right| \\
&\leq \frac{\varepsilon}{N-n+1} \left\{ \frac{3}{2} + \frac{|\sin[2\pi(2N-2n+3)/N]|}{2\sin(2\pi/N)} \right\} \leq \frac{\varepsilon}{N-n+1} \left\{ \frac{3}{2} + \frac{2\pi(2N-2n+3)/N}{4\pi/N} \right\} \approx \varepsilon \ll 1, \tag{3.12}
\end{aligned}$$

hence the expression under the modulus in (3.11) is positive. Therefore,

$$\begin{aligned}
\sum_{j=1}^N |d_{kj}| &= \frac{1}{N} \left(\sum_{n=2}^{N+1-k} + \sum_{n=2}^k \right) \left\langle 1 + \frac{\varepsilon}{N-n+1} \left\{ \frac{1}{2} - \frac{\sin[2\pi(2n-3)/N]}{2\sin(2\pi/N)} - \cos[2\pi(n-2)/N] \right\} \right\rangle \\
&= \frac{N-1}{N} + \frac{\varepsilon}{N} \left(\sum_{n=2}^{N+1-k} + \sum_{n=2}^k \right) \frac{1}{N-n+1} \left\{ \frac{1}{2} - \frac{\sin[2\pi(2n-3)/N]}{2\sin(2\pi/N)} - \cos[2\pi(n-2)/N] \right\} \\
&\leq 1 - \frac{1}{N} + \frac{\varepsilon}{N} \sum_{n=2}^N \frac{1}{N-n+1} \left(\frac{3}{2} + \frac{1}{4\pi/N} \right) < 1, \quad (\varepsilon \ll 1), \implies \|D\| < 1. \tag{3.13}
\end{aligned}$$

Now, if in the operator equation of the second kind (3.8) the norm of operator $\|D\| < 1$, then, by Banach theorem [25], Eq. (3.8) is uniquely solvable and $\|(1+D)^{-1}\| \leq 1/(1-\|D\|)$ is finite.

Let us estimate the norm of operator $C^t - C$ in Eq. (3.8). Obviously, both these matrices have the constant elements $1/2$ on the principal diagonal which are cancelled when calculating their difference. Then, for the elements outside the principal diagonal one obtains from (3.7)

$$\begin{aligned}
&\frac{4\pi}{ab h_\theta} (d_{k+n-1,k} - c_{k+n-1,k}) \\
&= \frac{1}{N-n+1} \sum_{j=1}^{N-n+1} \frac{1}{(a^2-b^2) \sin^2 \frac{(2j+n-2)h_\theta}{2} + b^2} - \frac{1}{(a^2-b^2) \sin^2 \frac{(2k+n-2)h_\theta}{2} + b^2} \\
&= \frac{1}{N-n+1} \sum_{j=1}^{N-n+1} \left[\frac{1}{(a^2-b^2) \sin^2 \frac{(2j+n-2)h_\theta}{2} + b^2} - \frac{1}{(a^2-b^2) \sin^2 \frac{(2k+n-2)h_\theta}{2} + b^2} \right]
\end{aligned}$$

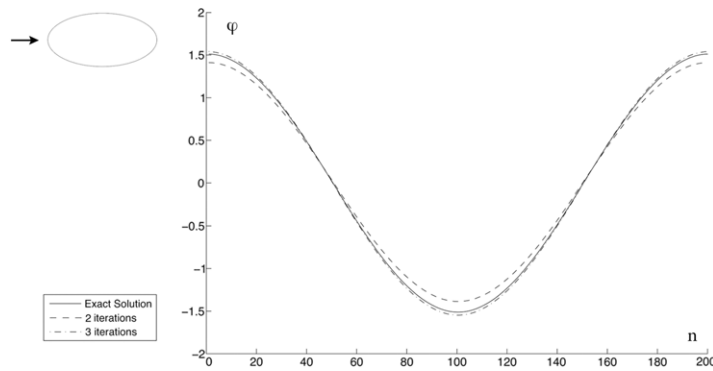


Fig. 1. Flow over a hard ellipse, $b/a = 0.5$, $N = 200$, $(1 \leq n \leq N)$. The first node is near the rightmost boundary point.

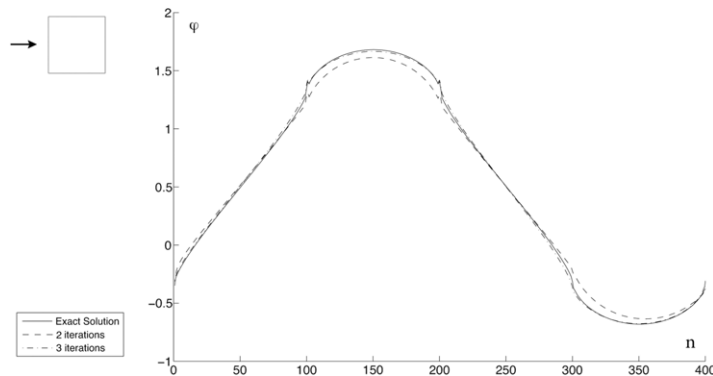


Fig. 2. Flow over a hard quadrate, $N = 400$, $(1 \leq n \leq N)$. The first node is near the left lowest boundary point.

$$= \frac{a^2 - b^2}{N - n + 1} \sum_{j=1}^{N-n+1} \frac{\sin^2 \frac{(2k+n-2)h_\theta}{2} - \sin^2 \frac{(2j+n-2)h_\theta}{2}}{\left[(a^2 - b^2) \sin^2 \frac{(2j+n-2)h_\theta}{2} + b^2 \right] \left[(a^2 - b^2) \sin^2 \frac{(2k+n-2)h_\theta}{2} + b^2 \right]}. \quad (3.14)$$

This implies

$$\begin{aligned} |d_{k+n-1,k} - c_{k+n-1,k}| &\leq \frac{ab h_\theta}{4\pi} \frac{a^2 - b^2}{N - n + 1} \sum_{j=1}^{N-n+1} \frac{2}{b^4} \approx \frac{2\varepsilon}{N}, \\ \Rightarrow \|C^t - C\| &= \max_{1 \leq k \leq N} \sum_{j=1}^N |d_{kj} - c_{kj}| \leq 2\varepsilon \ll 1, \end{aligned} \quad (3.15)$$

that yields the following final estimate in Eq. (3.8):

$$\varphi = F + 2(I + D)^{-1}(C^t - C)\varphi, \quad \|2(I + D)^{-1}(C^t - C)\| \leq 2\|(I + D)^{-1}\| \|C^t - C\| \leq \frac{2\varepsilon}{1 - \|D\|} \quad (3.16)$$

the value which can be attained arbitrarily small for small ε . Again, according to the classical Banach theory, if the last norm is less than unit value then the operator equation of the second kind (3.16) is uniquely solvable and its solution can be constructed by the standard successive iteration process, or equivalently, by Neumann's operator series [25].

When evaluating practical convergence of the proposed iteration algorithm, one may notice that this converges provided all the above estimates are valid. For example, the real range of convergence in the class of elliptic objects is much wider than those with small eccentricity. Further examples show that the convergence is perfect also for various geometries different from elliptic ones.

Figs. 1 and 2 demonstrate convergence of the iteration process for the ellipse with aspect ratio $\varepsilon = b/a = 1/2$ and for the quadratic domain. For such simple geometries 10 iteration steps provide the relative error less than 10^{-5} in the solution, when compared to the Gauss elimination technique. Note that in these and all forthcoming examples the enumeration of nodes is arranged counterclockwise when passing along the boundary contour.

Fig. 3 demonstrates a 8-leaves contour, for a correct treatment one needs to take at least 800 nodes, 100 nodes per every leaf.

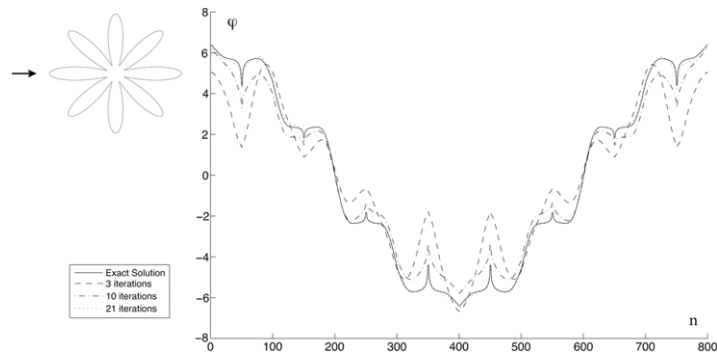


Fig. 3. Flow over a hard 8-leaves flower, $N = 800$, ($1 \leq n \leq N$). The first node is near the rightmost boundary point.

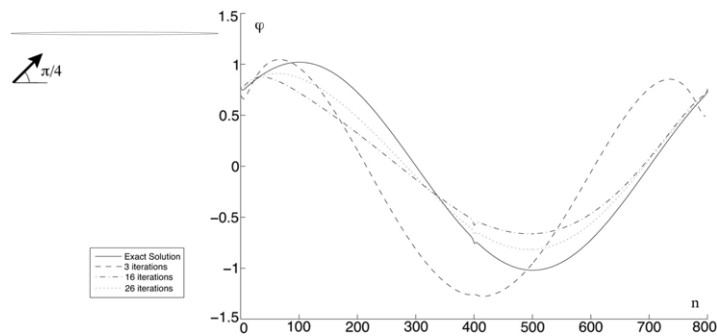


Fig. 4. Flow over a hard ellipsis, $b/a = 0.02$, $N = 800$, ($1 \leq n \leq N$). The first node is near the rightmost boundary point.

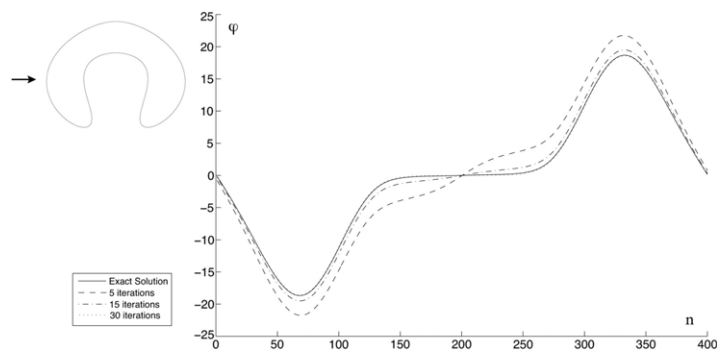


Fig. 5. Flow over a hard roll, $N = 400$, ($1 \leq n \leq N$). The first node is near the upper boundary point.

Since during the intensive investigation we could not find any geometry where the method proposed diverges, let us restrict the calculations by the class of ellipses with varying aspect ratio. By decreasing $\varepsilon = b/a$ up to value $\varepsilon = 0.02$ we come to the geometry demonstrated with its BIE solution in Fig. 4. With further decrease of parameter ε the calculations show that for extremely small values of ε the iteration process diverges. The critical value separating convergence from divergence is somewhere near $\varepsilon^* = 0.014$.

When looking at so stable convergence of the proposed algorithm, one may suppose that all considered examples are taken only for objects close, in some sense, to a round shape which exactly generates a real Töeplitz matrix. In order to study this question, let us consider two other geometries which seem absolutely different, both topologically and quantitatively, from any circular domain; see Figs. 5 and 6.

As a final remark in this section we note that the structure of the exact matrix C is such that its principal diagonal in (2.4a) $c_{ii} = 1/2$ is Töeplitz-like. Since the principal diagonal is usually dominant for well-posed matrices, one may assume that this property makes the iteration process convergent and stable. The next section demonstrates the convergence in the case when the principal diagonal is of more general structure.

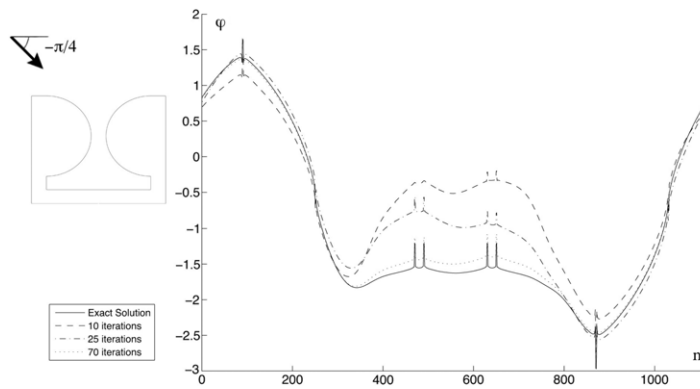


Fig. 6. Flow over a hard body of a complicated shape, $N = 1120$, ($1 \leq n \leq N$). The first node is close to the center of the lower boundary segment.

4. Flow over a soft body

If the boundary condition is $\varphi|_l = 0$, then Eq. (2.1) immediately yields the following integral equation

$$\int_l g(y) \ln |y - x| dl_y = f(x), \quad x \in l, \quad (4.1)$$

$$g(y) = \frac{\partial \varphi}{\partial n} \Big|_l = v_n|_l, \quad f(x) = -2\pi \varphi_\infty(x) = -2\pi v_0(x_1 \cos \alpha + x_2 \sin \alpha)$$

which is the basic BIE in the case of soft boundary.

The same discretization like in the case of hard body, applied to Eq. (4.1), leads to the following LAS

$$\sum_{j=1}^N c_{ij} g_j = f_i, \quad c_{ij} = \ln |x_j - x_i| \Delta l_j, \quad j \neq i, \quad (4.2a)$$

$$f_i = f(x_i), \quad x_i \in l, \quad i = 1, \dots, N.$$

In order to write out correctly the diagonal elements of the matrix here, $j = i$, one needs to calculate explicitly the contribution over the small arc:

$$c_{ii} = \int_{x_i - \frac{\Delta l_i}{2}}^{x_i + \frac{\Delta l_i}{2}} \ln |y - x_i| dy = 2 \int_0^{\Delta l_i/2} \ln y dy = \Delta l_i \left(\ln \frac{\Delta l_i}{2} - 1 \right). \quad (4.2b)$$

The same iteration scheme proposed in the previous section can be applied to system (4.2) too. Let us note that if the length of the elementary arc varies with i : $\Delta l_i \neq \text{const}$, then diagonal elements (4.2b) are different for different i , and so matrix c_{ij} is not of Töeplitz type.

The detailed numerical analysis shows that the proposed iteration algorithm converges in many cases also in the case of soft boundary, like for the hard contour. First of all, it is very interesting to find the critical value $\varepsilon^* = b/a$ for ellipses, if exists, analogous to that in the case of absolutely hard body. This is found to be around $\varepsilon^* = 0.15$. The comparison with $\varepsilon^* = 0.014$ from the previous section leads to the supposition that the developed iteration technique is less applicable in the soft case. Numerous examples considered confirm that this is so indeed. The quadratic domain gives the example when the method converges, the respective solution is reflected in Fig. 7. However, for geometries used in Figs. 5 and 6 the iterations diverge.

One may assume that the supposition indicated above is valid, namely that the Töeplitz-type principal diagonal predetermines convergence, therefore such a diagonal structure takes place for the hard body with good convergence but it is not Töeplitz for soft boundaries. However, on an example related to Fig. 6 the constant elementary arc length $\Delta l_j = \text{const}$, $\forall j$ can easily be arranged, but even in this case the iteration process diverges.

Let us try to improve the convergence of the iteration algorithm by passing to the indirect BIE method which reduces the Dirichlet boundary value problem to a second-kind integral equation. This means that one may seek the solution as a double-layer potential:

$$\varphi'(y_0) = \int_l u(y) \frac{\partial G(y, y_0)}{\partial n_y} dl_y, \quad (4.3)$$

where $y_0 = (y_1^0, y_2^0)$ is an arbitrary point in the flow, and $u(y)$ is a certain unknown function on the boundary contour. If the boundary of the body is absolutely soft: $\varphi|_l = 0$, $\varphi = \varphi' + \varphi_\infty$, then (4.3) implies

$$\frac{u(x)}{2} + \int_l u(y) \frac{\partial G(y, x)}{\partial n_y} dl_y = -\varphi_\infty(x), \quad x \in l, \quad (4.4)$$

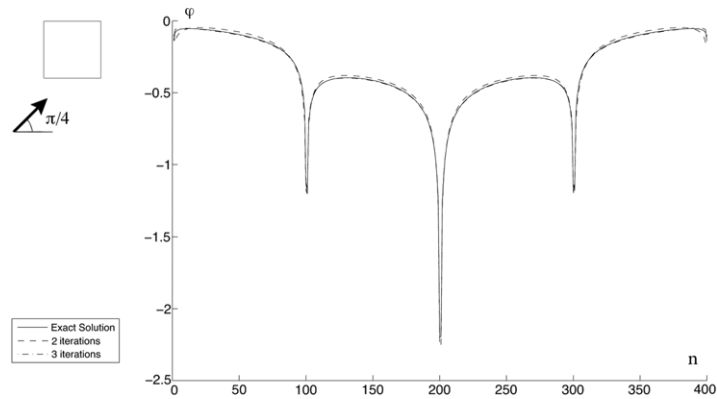


Fig. 7. Flow over a soft quadrate, $N = 400$, ($1 \leq n \leq N$). The first node is near the left lowest boundary point.

where the boundary property of the double-layer integral has again been applied. Let us note that, like with the direct BIE treatment for the rigid boundary, Eq. (4.4) is indeed the Fredholm integral equation of the second kind.

However, a detailed numerical analysis performed for Eq. (4.4) in its discrete form, analogous to what is presented by expressions (2.4), shows that convergence of the iteration scheme is even worse than with the treatment by the first-kind integral equation; see Eqs. (4.1) and (4.2). In particular, for elliptic obstacles iterations converge if $0.738 < b/a < 1$ only. It is completely unexpected, since the only difference between the integral operators on the left-hand sides of Eqs. (2.3) and (4.4) is opposite signs in front of the integral. This property indicates implicitly that such a poor behavior of the proposed algorithm in the case of soft boundary is connected with the physical nature of the flow for soft obstacles rather than with the type of the integral equation.

5. The Töeplitz solver

As follows from the previous sections, the proposed algorithm on the q -th iteration step requires to solve the following LAS

$$C^t g_q = f_q \quad (5.1)$$

where matrix C^t is Töeplitz and represents the respective matrix on the left-hand side of the operator equation for the iteration method proposed. By analogy, g_q is a notation for the respective vector of the solution, and f_q is a collective notation for the respective right-hand side. Besides, only in this section we accept the notation more standard for computational algorithms, when the elements of any vector vary over $(0, \dots, N-1)$, instead of $(1, \dots, N)$. The accepted notations imply that the matrix elements are constant along every diagonal parallel to the principal one, i.e.

$$C^t = \begin{pmatrix} c_0^t & c_1^t & \cdots & c_{N-2}^t & c_{N-1}^t \\ c_{-1}^t & c_0^t & c_1^t & & c_{N-2}^t \\ \vdots & c_{-1}^t & c_0^t & \ddots & \vdots \\ c_{-(N-2)}^t & & \ddots & \ddots & c_1^t \\ c_{-(N-1)}^t & c_{-(N-2)}^t & \cdots & c_{-1}^t & c_0^t \end{pmatrix}. \quad (5.2)$$

There are only $2N - 1$ different quantities in this matrix, hence there is a good chance to construct an appropriate fast solver which requires less than classical $O(N^3)$ arithmetic operations. In 1980s there was developed the idea to use a preconditioning technique in iterative algorithms (the most popular among them is the conjugate gradient method (CG)). It was shown that the complexity of this approach can be reduced to $O(MN \log N)$ operations, where M is the number of iterations which does not depend on the mesh size, being connected with the condition number of the matrix only. It will be clear from the forthcoming text that this gives the advantage to use the CG method on each iteration step of the proposed iterative scheme. Moreover, one can use vector g_q , constructed on the previous step of the algorithm, as an initial iteration for the CG procedure over the next iteration step. As will be shown, such an approach reduces also the number of iterations required to solve system (5.1). In our problem we have used the bi-conjugate gradient method with circulant preconditioning (PBCG) due to its stability and fast convergence. Another attractive feature of the PBCG method is that it is based on the short and clear recurrence relations

$$\begin{aligned}
r_0 &= f_q - C^t x_0, & r'_0 &= r_0, \\
p_1 &= P^{-1} r_0, \\
p'_1 &= (P^T)^{-1} r'_0, \\
\alpha_i &= (r'_{i-1}, P^{-1} r_{i-1}) / (C^t p_i, p'_i), \\
x_i &= x_{i-1} + \alpha_i p_i, \\
r_i &= r_{i-1} - \alpha_i C^t p_i, \\
r'_i &= r'_{i-1} - \alpha_i (C^t)^T p'_i, \\
\beta_i &= (r'_i, P^{-1} r_i) / (r'_{i-1}, P^{-1} r_{i-1}), \\
p_{i+1} &= P^{-1} r_i + \beta_i p_i, \\
p'_{i+1} &= (P^T)^{-1} r'_i + \beta_i p'_i.
\end{aligned} \tag{5.3}$$

Here vector r_i is the residual error and x_0 is the initial guess, f_q is the right-hand side from (5.1). In our case we use g_q constructed on the previous iteration of the main algorithm as x_0 . Matrix P is the preconditioner and it is assumed to be constructed easily. Moreover, the multiplication by P^{-1} should be done efficiently. In the case of circulant matrices both of these assumptions are valid. As noted in the Introduction, in all cases we used for this purpose a C++ code of the ACA algorithm as a part of the standard library HLibPro.

The circulant matrix P is a periodic Toeplitz matrix, where $p_{-k} = p_{N-k}$ for $1 \leq k \leq N-1$, i.e.

$$P = \begin{pmatrix} p_0 & p_1 & \cdots & p_{N-2} & p_{N-1} \\ p_{N-1} & p_0 & p_1 & & p_{N-2} \\ \vdots & p_{N-1} & p_0 & \ddots & \vdots \\ p_2 & & \ddots & \ddots & p_1 \\ p_1 & p_2 & \cdots & p_{N-1} & p_0 \end{pmatrix}. \tag{5.4}$$

It is well-known that any circulant matrix can be diagonalized via Fourier Transform: $P = F^* \Lambda F$, where Λ is a diagonal matrix containing all eigenvalues of P and can be obtained by applying FFT to the first column of P . The Fourier matrix F is

$$(F)_{kl} = \frac{1}{\sqrt{N}} e^{2\pi i k l / N}$$

where $i = \sqrt{-1}$ and the product of this matrix with any vector is a matrix representation of the Fast Fourier Transform. More detailed descriptions and notations for FFT techniques can be found in [27,28]. It is clear that the inverse to the matrix P is also a circulant matrix $P^{-1} = F \Lambda^{-1} F^*$, and the multiplication of such a matrix by any vector can be achieved by using the Fast Fourier Transform. So the computational cost is $O(N \log(N))$ arithmetic operations. There are several types of preconditioners for Toeplitz matrices. In our work we have used well-known T. Chan's circulant preconditioner with the following entries

$$p_j = \begin{cases} \frac{(N-j)c_j^t + j c_{j-N}^t}{N}, & 0 \leq j < N; \\ p_{N+j}, & 0 < -j < N. \end{cases} \tag{5.5}$$

To construct a fast matrix–vector multiplication algorithm for the Toeplitz matrix, one can embed C^t into a circulant matrix of size $2N \times 2N$ in the following way

$$S^t = \begin{pmatrix} C^t & \boxtimes \\ \boxtimes & C^t \end{pmatrix}. \tag{5.6}$$

Then the multiplication by a vector

$$\begin{pmatrix} C^t & \boxtimes \\ \boxtimes & C^t \end{pmatrix} \begin{pmatrix} g \\ o \end{pmatrix} = \begin{pmatrix} C^t g \\ \boxtimes g \end{pmatrix} \tag{5.7}$$

can be achieved in $O(2N \log(2N))$ operations via FFT. Here vector o is zero-vector and \boxtimes represents the blocks of the matrix required to construct a circulant matrix from C^t . Note, that vector $C^t g$ is obtained by cutting the resulting vector by its first N entries.

These two techniques applied in routine (5.3) give the fast and efficient algorithm to solve the Toeplitz LAS problem.

6. Conclusions

1. In the present work we propose a new numerical iteration method to solve BIEs arising in various boundary value problems of mathematical physics. The essence of the algorithm is demonstrated for a flow of inviscid incompressible fluid around a body, in the two-dimensional formulation. At each iteration step there is a need to solve a certain LAS with a Töeplitz matrix, which is constructed by a certain averaging along diagonals parallel to the principal one. As Töeplitz solvers we use some special techniques known from the literature, and give an estimate of respective computational expenses which shows an evident significant acceleration of the algorithm, when compared to standard Gauss elimination.
2. The convergence of the proposed iteration algorithm cannot be proved strictly for arbitrary geometry of the object. However, we demonstrate a proof in the class of ellipses with sufficiently small eccentricity. The scheme of the proof shows that the key point for the convergence is that operator $I + D$ in Eq. (3.8) must be invertible. As follows from the Fredholm theory [25], this property is valid for a wider class of boundary contours, not necessarily with the condition $\|D\| < 1$. Therefore, the practical significance of the method is that this is applicable not only for the class of elliptic objects of small eccentricity with the given proof.
3. The method demonstrates wonderful stability for hard boundaries of various types of complex geometry. There has been performed a special investigation, to find the cases when the algorithm diverges. We could only detect that this diverges for extremely elongated ellipses, and we did not succeed in finding any other examples.
4. Unfortunately, in the case of the Dirichlet boundary condition the efficiency of the algorithm is poorer. This is so when applied to the first-kind Fredholm integral equation of the direct BIE method, and unexpectedly even worse convergent when applied to the second-kind equation of the indirect method. Therefore, we come to the conclusion that probably the convergence is more dependent on the physical nature of the problem, rather than upon the kind of the integral equation.
5. In the case of perfect convergence, i.e. with the Neumann-type boundary condition, the number of iterations is insensitive with respect to the numerical grid dimension. This significantly depends on geometry of the body only. The more complex the geometry the more iterations are required. This means in practice that the complexity of the algorithm is free of the number of iterations, being determined only by the computational expenses to solve Töeplitz-type LAS.
6. It should be noted that a standard analysis of the computational expense of the proposed algorithm shows that its complexity is $O[2N \log(2N)]$. However, we do not insist that this is the most optimal by its cost. In particular, in frames of some type of the pre-corrected FFT algorithm [29,30], one can achieve a linear complexity in computer memory and $O(N \log N)$ arithmetic operations.

Acknowledgments

The present work has been supported in part by Italian Ministry of University (M.I.U.R) through its national and local projects, and in part by Russian Ministry for Education and Science, Project 16-516-11-6106. The authors also express their gratitude to two anonymous reviewers for their very helpful comments.

References

- [1] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, *J. Comput. Phys.* 73 (1987) 325–348.
- [2] V. Rokhlin, Rapid solution of integral equation of classical potential theory, *J. Comput. Phys.* 60 (1985) 187–207.
- [3] W. Hackbusch, Z.P. Novak, On the fast matrix multiplication in the boundary element method by panel clustering, *Numer. Math.* 54 (4) (1989) 463–491.
- [4] Yu.M. Nechepurenko, Fast numerically stable algorithms for a wide class of linear discrete transformations, Preprint 92, Dpt. Comp. Math., Russian Academy Sci., 1985 (in Russian).
- [5] A. Brandt, A.A. Lubrecht, Multilevel matrix multiplication and fast solution of integral equations, *J. Comput. Phys.* 90 (1990) 348–370.
- [6] A. Harten, Discrete multiresolution analysis and generalized wavelets, *J. Appl. Numer. Math.* 12 (1993) 153–193.
- [7] A. Harten, Adaptive multiresolution schemes for shock computations, UCLA CAM Report 93-06, April 1993.
- [8] G. Beylkin, R. Coifman, V. Rokhlin, Fast wavelet transform and numerical algorithms, *Comm. Pure Appl. Math.* 44 (1991) 141–183.
- [9] F.X. Canning, The impedance matrix localization (IML) method for moment-method calculations, *IEEE Antennas Propag. Mag.* 32 (1990) 18–30.
- [10] I. Oseledets, E. Tyrtshnikov, A unifying approach to construction of circulant preconditioners, *Linear Algebra Appl.* 418 (2006) 435–449.
- [11] E. Tyrtshnikov, Incomplete cross approximation in the mosaic-skeleton method, *Computing* 64 (2000) 367–380.
- [12] S.A. Goreinov, E.E. Tyrtshnikov, N.L. Zamarashkin, A theory of pseudo-skeleton approximations, *Linear Algebra Appl.* 261 (1997) 1–21.
- [13] V.V. Voevodin, On a certain method to reduce the dimension of matrices in solution of integral equations, in: *Numerical Analysis in FORTRAN*, Moscow University Press, Moscow, 1979, pp. 21–26.
- [14] P.G. Martinsson, V. Rokhlin, An accelerated kernel-independent fast multipole method in one dimension, *SIAM J. Sci. Comput.* 29 (3) (2007) 1160–1178.
- [15] M. Bebendorf, Approximation of boundary element matrices, *Numer. Math.* 86 (2000) 565–589.
- [16] M. Bebendorf, *Hierarchical Matrices*, Springer, Berlin, 2008.
- [17] S. Borm, L. Grasedyck, W. Hackbusch, Introduction to hierarchical matrices with applications, *Eng. Anal. Bound. Elem.* 27 (5) (2003) 405–422.
- [18] R. Kriemann, Parallel H -matrix arithmetics on shared memory systems, *Computing* 74 (2005) 273–297.
- [19] L. Grasedyck, R. Kriemann, S. Le Borne, Parallel blackbox H -LU preconditioning for elliptic boundary value problems, *Comput. Vis. Sci.* 11 (2008) 273–291.
- [20] <http://www.hlibpro.com>.
- [21] M. Bonnet, *Boundary Integral Equations Methods for Solids and Fluids*, John Wiley, NY, 1999.
- [22] M.A. Sumbatyan, A. Scalia, *Equations of Mathematical Diffraction Theory*, CRC Press, Boca Raton, Florida, 2005.
- [23] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, second ed., Cambridge University Press, Cambridge, 1992.

- [24] A.P. Prudnikov, Y.A. Brychkov, O.I. Marichev, *Integrals and Series* (Vol. 1), Gordon and Breach, Amsterdam, 1986.
- [25] L.V. Kantorovich, G.P. Akilov, *Functional Analysis*, Pergamon Press, NY, 1982.
- [26] R.A. Horn, C.R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
- [27] G. Strang, A proposal for Toeplitz matrix calculations, *Stud. Appl. Math.* 74 (1986) 171–176.
- [28] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.
- [29] J.R. Phillips, J.K. White, A precorrected-FFT method for electrostatic analysis of complicated 3-D structures, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 16 (1997) 1059–1072.
- [30] S.N. Fata, Fast Galerkin BEM for 3D potential theory, *Comput. Mech.* 42 (2008) 417–429.