Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 202 (2005) 391–410
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Laplace's equation and the Dirichlet–Neumann map: a new mode for Mikhlin's method ☆

Johan Helsing *, Eddie Wadbro [1]

Numerical Analysis, Centre for Mathematical Sciences, Lund University, P.O. Box 118, SE-221 00 Lund, Sweden

Received 18 March 2004; accepted 30 June 2004
Available online 27 August 2004

Abstract

Mikhlin's method for solving Laplace's equation in domains exterior to a number of closed contours is discussed with particular emphasis on the Dirichlet–Neumann map. In the literature there already exist two computational modes for Mikhlin's method. Here a new mode is presented. The new mode is at least as stable as the previous modes. Furthermore, its computational complexity in the number of closed contours is better. As a result, highly accurate solutions in domains exterior to tens of thousands of closed contours can be obtained on a simple workstation.

MSC: 65R20; 77C05
Keywords: Laplace's equation; Exterior problem; Multiply connected domains; Integral equations; Fast solvers; Dirichlet–Neumann map

1. Introduction

Solving Laplace's or the biharmonic equation in domains exterior to a large number of closed contours or surfaces is a common task in many branches of applied mathematics. Often, in 2D, these problems can be reduced to boundary value problems in the theory of analytic functions and, by particular choices of representations or by other means, recast as Fredholm second kind integral equations. Examples can be

☆ This work was supported by the Swedish Research Science Council under contract 621-2001-2799.
* Corresponding author. Tel.: +46 46 222 33 72; fax: +46 46 222 4623.
E-mail addresses: helsing@maths.lth.se (J. Helsing), eddiew@it.uu.se (E. Wadbro).
[1] Present address: Division of Scientific Computing, Department of Information Technology, Uppsala University, P.O. Box 337, SE-751 05 Uppsala, Sweden.

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2004.06.024

found in materials science (microstructural evolution and particle coarsening), electrostatics, elasticity, and fluid dynamics (Stokes flow). See chapters IV and V in Mikhlin [19] for a classic reference and [1–3,6,7,11,13,16,18,22,24,26–28] for more recent work and applications.

Numerical methods based on integral equations are, as indicated, natural choices for the problems just mentioned. When the boundary data correspond to solutions with sources and sinks inside the contours, standard methods for Laplace's and the biharmonic equations seem to be those of Mikhlin and Sherman [19,20]. These similar methods were popularized in the 1990s by Greenbaum et al. [5] and by Greengard et al. [7] who presented Mikhlin's method and Sherman's method (in constrained formulation) with two computational modes: one unpreconditioned and one preconditioned. Despite the use of the fast multipole method [4,8,23], these modes have some unwanted properties regarding complexity and stability which become apparent when the number of closed contours is large. In practice, the preconditioned modes are often chosen [1–3,6,16,28]. For large problems they may be applied to smaller, overlapping, subproblems involving a few hundred closed contours [1,2,28]. The overall resulting accuracy in the solution is then, perhaps, only one per cent [1].

This paper presents a new computational mode for Mikhlin's method in the Dirichlet–Laplace setting. The new mode is stable for large problems. Its complexity in execution time and storage is almost linear in the number of discretization points and also in the number of closed contours. At least this holds for a problem class which is common in materials science. Solutions with a controlled relative error of, for example, less than 10^{-8} can be achieved in domains exterior to 30,000 closed contours on a simple workstation. No overlapping subproblems are involved. The leading idea is to reformulate Mikhlin's original method in a way which makes it more symmetric and then to apply efficient right preconditioners on two levels: first to the main system which results from discretizing the integral equation and then to a smaller Schur complement system appearing within the main-level preconditioner. The main-level preconditioner is a modified approximation to the preconditioner used in the preconditioned mode of [5]. Its purpose is to convert the main system into a system whose condition number stays bounded not only when the number of discretization points grow due to mesh refinement but also when they grow due to an increased number of closed contours. The Schur complement system preconditioner is an approximation to a certain logarithm matrix, capturing the ill-conditioning of the underlying mathematical problem. Its inverse is cheaply constructed using a modified version of an existing scheme for system matrices resulting from the discretization of some first kind Fredholm integral equations arising in potential theory [21].

The paper is organized as follows: Section 2 contains the problem statement. Section 3 reviews Mikhlin's method and related integral equation methods for Laplace's equation in multiply connected domains. Section 4 covers discretization and matrix partitioning. The three modes – the two modes of [5] and our new mode – are presented in Section 5. The computation of the Dirichlet–Neumann map in a post-processing step and the construction of test problems are described in Sections 6 and 7. Section 8 presents numerical results including a very thorough study of numerical errors and large-scale computations of unprecedented size. The paper ends with a discussion in Section 9.

2. Problem statement

Let D be an infinite, multiply connected, two dimensional domain exterior to M closed smooth contours $L_1, \ldots, L_M$. We refer to L_k as *boundary components* and call their union L. The normal of L, inwards D, is denoted ν.

The exterior Dirichlet problem for Laplace's equation with boundary data $f(Q)$ is to find a function $U(P)$ which satisfies:

$$\Delta U(P) = 0, \quad P \in D, \tag{1}$$

$$\lim_{D\ni P\to Q} U(P) = f(Q), \quad Q \in L. \tag{2}$$

A condition regarding the behavior in the far field is needed in order for this problem to have a unique solution. We require that there is a real number λ such that

$$| U(P) | \leqslant \lambda, \quad P \in D, \tag{3}$$

that is, $U(P)$ is bounded.

Once $U(P)$ is found, its normal derivative $g(Q)$ on L can be computed

$$g(Q) = \lim_{t\to 0^+} v(Q)\cdot\nabla U(Q + tv(Q)), \quad Q \in L. \tag{4}$$

The mapping $f(Q) \to g(Q)$ is called the Dirichlet–Neumann map. Its efficient computation on domains with large M is our topic.

In the remainder of this paper there will be no distinction made between a point $P = (x, y) \in \mathbb{R}^2$ and a point $z = x + \mathrm{i}y \in \mathbb{C}$. The real and imaginary parts of a complex number w are denoted $\Re\{w\}$ and $\Im\{w\}$. Further, ζ and τ always denote points located on L while z and z_k are reserved for points not on L. The boundary components have positive orientations and their lengths are $|L_k|$. In the context of integration, an incremental element of arc is $\mathrm{d}\sigma$. In the context of discretization, N is the total number of discretization points.

3. Integral equations of Mikhlin type

Mikhlin, in paragraph 31 of his book [19], suggests that the solution $U(P)$ to (1)–(3) be regarded as the real part of an analytic function $\varphi(z)$

$$U(P) = \Re\{\varphi(z)\}. \tag{5}$$

The function $\varphi(z)$ is represented as

$$\varphi(z) = \varphi^*(z) + \sum_{k=1}^{M} a_k \log(z - z_k), \quad z \in D, \tag{6}$$

where z_k is an arbitrary point inside L_k and $\varphi^*(z)$ is a single-valued function in D. To ensure that (3) is satisfied, the coefficients a_k need to be subjected to the condition

$$\sum_{k=1}^{M} a_k = 0. \tag{7}$$

The function $\varphi^*(z)$ is sought in the form of a Cauchy-type integral and a certain constant

$$\varphi^*(z) = \frac{1}{2\pi\mathrm{i}} \int_L \frac{\mu(\zeta)}{\zeta - z}\,\mathrm{d}\zeta + \frac{1}{2\pi}\int_L \mu(\zeta)\,\mathrm{d}\sigma, \tag{8}$$

where the density $\mu(\zeta)$ is real, that is, $\mu(\zeta) : L \to \mathbb{R}$. Solving (1)–(3) now reduces to finding $\mu(\zeta)$ and a_k. Replicating the arguments in [19] one arrives at the system of equations

$$\begin{aligned} &\mu(\tau) - \frac{1}{\pi}\int_L \mu(\zeta)\Im\left\{\frac{\mathrm{d}\zeta}{\zeta - \tau}\right\} - \frac{1}{\pi}\int_L \mu(\zeta)\,\mathrm{d}\sigma - 2\sum_{k=1}^{M} a_k \log | \tau - z_k | = -2f(\tau), \\ &\sum_{k=1}^{M} a_k = 0, \quad \int_{L_k} \mu(\zeta)\,\mathrm{d}\sigma = 0, \quad k = 1, \ldots, M-1. \end{aligned} \tag{9}$$

Having solved (9) for $\mu(\zeta)$ and a_k, the solution $U(P)$ to (1)–(3) is recovered from (5), (6) and (8). This approach is used in [5]. A particular aspect of (9), which, as we shall see later, can have large implications on stability properties of numerical algorithms, is a certain lack of symmetry; in the last line of (9) the Mth boundary component L_M is not included.

The representation (8) of $\varphi^*(z)$ is not the only possibility. Another possible representation is

$$\varphi^*(z) = \frac{1}{2\pi \mathrm{i}} \int_L \frac{\mu^*(\zeta)}{\zeta - z}\, \mathrm{d}\zeta + c_0, \tag{10}$$

where c_0 is a constant and $\mu^*(\zeta) : L \to \mathbb{R}$. Starting from this representation one can arrive at the system of equations

$$\begin{aligned} &\mu^*(\tau) - \frac{1}{\pi} \int_L \mu^*(\zeta) \Im\left\{\frac{\mathrm{d}\zeta}{\zeta - \tau}\right\} - 2c_0 - 2\sum_{k=1}^{M} a_k \log \mid \tau - z_k \mid = -2f(\tau), \\ &\sum_{k=1}^{M} a_k = 0, \quad \frac{1}{2 \mid L_k \mid} \int_{L_k} \mu^*(\zeta)\, \mathrm{d}\sigma = 0, \quad k = 1, \ldots, M. \end{aligned} \tag{11}$$

Having solved (11) for $\mu^*(\zeta)$, a_k, and c_0, the solution $U(P)$ to (1)–(3) is recovered from (5), (6) and (10). One can observe a higher degree of symmetry in the system (11) than in the system (9); all M boundary components in (11) are treated in the same way. The connection between (9) and (11) is explored further in Appendix A.

Yet another integral equation based approach to (1)–(3) starts with the representation of $U(P)$ as a combination of single and double layer potentials. The resulting Fredholm equation is used in some commercial software, see [17] and the discussion in Section 9.

4. Discretization and matrix partitioning

We intend to solve the systems (9) and (11) using a Nyström algorithm based on the trapezoidal quadrature rule.

The discretization of (9) with N_k points, denoted by ζ_j^k, on each boundary component L_k generates the system:

$$\mu_i^l - \frac{1}{\pi} \sum_{k=1}^{M} h_k \sum_{j=1}^{N_k} \mu_j^k \left[\Im\left\{ \frac{\zeta_j^{k\prime}}{\zeta_j^k - \zeta_i^l} \right\} + 1 \right] - 2\sum_{k=1}^{M} a_k \log \mid \zeta_i^l - z_k \mid = -2f(\zeta_i^l), \tag{12}$$

$$\sum_{k=1}^{M} a_k = 0, \tag{13}$$

$$\sum_{j=1}^{N_k} \mu_j^k h_k = 0, \quad k = 1, \ldots, M-1. \tag{14}$$

Here h_k is the step in arclength on L_k, $\zeta_j^{k\prime} = \mathrm{d}\zeta/\mathrm{d}\sigma(\zeta_j^k)$, and $\mu_j^k = \mu(\zeta_j^k)$. Appropriate limits, $\Im\{\zeta_j^{k\prime}/(\zeta_j^k - \zeta_i^l)\} = \Im\{\zeta_i^{l\prime\prime}/(2\zeta_i^{l\prime})\}$, are taken when $\zeta_j^k = \zeta_i^l$. The system (12)–(14) is the same as the system (25) of [5].

Remark 1. In some situations, such as when boundary components have high local curvature or are closely gathered together, it might be advantageous to adopt a nonuniform mesh spacing. This is easily

accomplished by employing suitable stretchings of the arclength coordinate [5] or by using composite Gaussian quadrature and a special technique to evaluate layer potentials close to their sources [9].

The discrete Eqs. (12)–(14) can be written in block form

$$\begin{bmatrix} I-K & B \\ C & D \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} -2\mathbf{f} \\ 0 \end{bmatrix}. \tag{15}$$

Here $\boldsymbol{\mu}$ is the vector of the unknown density values, $\mathbf{a}$ is the vector of the unknown coefficients, and $\mathbf{f}$ is the vector of given boundary values. The $N\times N$ matrix K corresponds to the interactions of the double layer potentials and the $N\times M$ matrix B represent the logarithmic terms in (12). The $M\times M$ matrix D and the $M\times N$ matrix C represent the constraint equations (13) and (14).

The discretization of (11) generates a system analogous to (12)–(14):

$$\mu_i^{*l} - \frac{1}{\pi}\sum_{k=1}^{M} h_k \sum_{j=1}^{N_k} \mu_j^{*k} \Im\left\{\frac{\zeta_j^{k\prime}}{\zeta_j^k - \zeta_i^l}\right\} - 2c_0 - 2\sum_{k=1}^{M} a_k \log|\zeta_i^l - z_k| = -2f(\zeta_i^l), \tag{16}$$

$$\sum_{k=1}^{M} a_k = 0, \tag{17}$$

$$\frac{1}{2|L_k|}\sum_{j=1}^{N_k} \mu_j^{*k} h_k = 0, \quad k = 1,\ldots,M. \tag{18}$$

In block form this system reads

$$\begin{bmatrix} I-\tilde{K} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}^* \\ \begin{bmatrix} c_0 \\ \mathbf{a} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} -2\mathbf{f} \\ 0 \end{bmatrix}. \tag{19}$$

Here the $N\times N$ matrix $\tilde{K}$ corresponds to the interactions of the double layer potentials in (16). The $(M+1)\times N$ matrix $\tilde{C}$ and the $(M+1)\times(M+1)$ matrix $\tilde{D}$ represent the constraint Eqs. (17) and (18). The $N\times(M+1)$ matrix $\tilde{B}$ can be written

$$\tilde{B} = [-2\mathbf{e}_N \; B], \tag{20}$$

where $\mathbf{e}_N$ is a column vector with N elements all equal to one.

5. Modes

The numerical method presented in [5] solves the system (15) making use of the GMRES iterative solver [25] accelerated by the fast multipole method [4,8,23]. Actually, the method in [5] is presented with two computational modes – one unpreconditioned and one preconditioned. In the rest of this paper we denote the unpreconditioned mode of [5] by Mode I and the preconditioned mode of [5] by Mode II. These two modes will now be presented more in detail together with our new mode for the system (19). Our new mode will be denoted Mode III.

5.1. Mode I

In Mode I, the linear system (15) is solved iteratively as it stands using GMRES. At each iteration a matrix–vector multiplication has to be computed. This can be done in $O(N+M)$ operations making use

of the fast multipole method for the blocks $I - K$ and B and since the blocks C and D are sparse. As we shall see in Section 8.3, the complexity of Mode I is not necessarily $O(N + M)$. The condition number of the system matrix in Mode I is observed to grow linearly with M, given that the relative sizes of and distances between the boundary components are not significantly changed as M increases, see Subsection 5.1 of [3] and Subsection 2.1.1 of [16]. If these conditions are violated the condition number could grow even faster. The number of GMRES iterations needed to meet a certain tolerance in Mode I may, therefore, depend on M.

5.2. Mode II

In Mode II, the linear system (15) is solved iteratively using a left preconditioner in GMRES. The preconditioner is

$$P = \begin{bmatrix} I & B \\ C & D \end{bmatrix}. \tag{21}$$

Thus, rather than solving the system (15) as it stands, one solves the system

$$\begin{bmatrix} I & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} I - K & B \\ C & D \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} I & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} -2\mathbf{f} \\ 0 \end{bmatrix}. \tag{22}$$

At each iteration for (22) it is necessary to solve a linear system with the preconditioner P of (21) as system matrix. To solve such a system

$$\begin{bmatrix} I & B \\ C & D \end{bmatrix} \begin{bmatrix} \mathbf{x}_\mu \\ \mathbf{x}_a \end{bmatrix} = \begin{bmatrix} \mathbf{b}_\mu \\ \mathbf{b}_a \end{bmatrix}, \tag{23}$$

one first forms the $M \times M$ Schur complement S of D in P

$$S = D - CB. \tag{24}$$

One then obtains $\mathbf{x}_a$ by solving

$$S\mathbf{x}_a = \mathbf{b}_a - C\mathbf{b}_\mu, \tag{25}$$

and $\mathbf{x}_\mu$ from the relation

$$\mathbf{x}_\mu = \mathbf{b}_\mu - B\mathbf{x}_a. \tag{26}$$

We refer to (22) as the *main system* and to the Schur complement system (25) as the *inner system* in Mode II. The inner system is solved by Gaussian elimination. Backsolving requires $O(M^2)$ operations each time the inner system is solved. The total cost of one iteration for the main system is therefore $O(N + M^2)$. The initial LU-factorization of S, however, requires $O(M^3)$ operations.

5.3. Mode III

In Mode III, the linear system (19) is solved iteratively using right preconditioners in GMRES. In contrast to Mode II, not only the main system is solved iteratively but also the inner system. In this way we overcome the $O(M^3)$ cost of factorizing S which makes Mode II impractical for domains with large M. In addition, as we shall see in Section 8.2, the combination of right preconditioning and the fact that (19) is more symmetric than (15) will make Mode III much more stable than Mode II.

5.3.1. The main system in Mode III

One could construct a preconditioner to the system (19) very similar to P of (21), namely

$$\tilde{P} = \begin{bmatrix} I & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix}. \tag{27}$$

The $(M+1)\times(M+1)$ Schur complement $\tilde{S}$ of $\tilde{D}$ in $\tilde{P}$ has the form

$$\tilde{S} = \tilde{D} - \tilde{C}\tilde{B}. \tag{28}$$

The structure of the matrix $\tilde{S}$ is simple. The element $\tilde{S}_{11}$ is zero. The elements $\tilde{S}_{1i}$ and $\tilde{S}_{i1}$, $i = 2,\ldots,M+1$, are one. The remaining elements of $\tilde{S}$ can be seen as discrete versions of integrals of the type

$$\frac{1}{|L_k|}\int_{L_k} \log|\zeta - z_i|\,\mathrm{d}\sigma. \tag{29}$$

For $i \neq k$ we make the following observation:

$$\frac{1}{|L_k|}\int_{L_k} \log|\zeta - z_i|\,\mathrm{d}\sigma \approx \frac{\log|z_k - z_i|}{|L_k|}\int_{L_k}\mathrm{d}\sigma = \log|z_k - z_i|\,. \tag{30}$$

This approximation is exact if L_k is a circle centered at z_k, according to the mean value theorem for harmonic functions. The approximation should also be reasonable when L_k does not deviate too much from a circle and, more importantly, when L_k is located far away from z_i.

We now use (30) to build an alternative preconditioner $\hat{P}$ given by

$$\hat{P} = \begin{bmatrix} I & \hat{B} \\ \tilde{C} & \tilde{D} \end{bmatrix}, \tag{31}$$

where $\hat{B}$ is an approximation to $\tilde{B}$ of (20) formed in the following way: all elements in $\tilde{B}$, except for those in the first column, have the form $-2\log|\zeta_j^k - z_i|$. These elements are in $\hat{B}$ replaced by $-2\log|z_k - z_i|$ whenever $i \neq k$. The Schur complement $\hat{S}$ of $\tilde{D}$ in $\hat{P}$ is then

$$\hat{S} = \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & \tilde{S}_{22} & \log|z_1 - z_2| & \cdots & \log|z_1 - z_M| \\ 1 & \log|z_2 - z_1| & \tilde{S}_{33} & \cdots & \log|z_2 - z_M| \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \log|z_M - z_1| & \log|z_M - z_2| & \cdots & \tilde{S}_{(M+1)(M+1)} \end{bmatrix}, \tag{32}$$

where $\tilde{S}_{ii}$ refers to the ith diagonal element of $\tilde{S}$ of (28).

Thus, rather than solving (19) as it stands, we apply $\hat{P}$ as a right preconditioner and solve the system

$$\begin{bmatrix} I - K & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix}\begin{bmatrix} I & \hat{B} \\ \tilde{C} & \tilde{D} \end{bmatrix}^{-1}\boldsymbol{\omega} = \begin{bmatrix} -2\mathbf{f} \\ 0 \end{bmatrix} \tag{33}$$

for the unknown $N + M + 1$ vector $\boldsymbol{\omega}$. We then obtain the solution to (19) from

$$\begin{bmatrix} \boldsymbol{\mu}^* \\ \begin{bmatrix} c_0 \\ \mathbf{a} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} I & \hat{B} \\ \tilde{C} & \tilde{D} \end{bmatrix}^{-1}\boldsymbol{\omega}. \tag{34}$$

5.3.2. Applying the preconditioner in Mode III

At each iteration for (33) it is necessary to solve a linear system with the preconditioner $\hat{P}$ of (31) as system matrix. Such a system is solved in a similar manner as a corresponding system in Mode II, see (23)–(26). However, for the equation corresponding to the inner system (25), that is

$$\hat{S}\mathbf{x}_{ca} = \mathbf{b}_{ca} - \tilde{C}\mathbf{b}_{\mu}, \tag{35}$$

we do not factorize $\hat{S}$. Instead we apply GMRES for (35) together with a right preconditioner whose inverse $\hat{S}_P^{-1}$ we construct explicitly.

The Schur complement $\hat{S}$ of (32), acting as system matrix in (35), can be written in block form

$$\hat{S} = \begin{bmatrix} 0 & \mathbf{e}_M^{\mathrm{T}} \\ \mathbf{e}_M & F \end{bmatrix}. \tag{36}$$

Here $\mathbf{e}_M$ is a column vector with M elements all equal to one, $\mathbf{e}_M^{\mathrm{T}}$ is its transpose, and F denotes the bottom right $M \times M$ submatrix in (32). Our plan for $\hat{S}_P^{-1}$ is to find an approximation $\hat{F}^{-1}$ to F^{-1}, and then to compute $\hat{S}_P^{-1}$ as the inverse of (36) using the Schur–Banachiewicz inverse formula [12] and replacing F^{-1} with $\hat{F}^{-1}$ whenever it occurs. The result is

$$\hat{S}_P^{-1} = \begin{bmatrix} -\alpha & \alpha\mathbf{v}^{\mathrm{T}} \\ \alpha\mathbf{u} & \hat{F}^{-1} - \alpha\mathbf{u}\mathbf{v}^{\mathrm{T}} \end{bmatrix}, \tag{37}$$

where $\alpha = (\mathbf{e}_M^{\mathrm{T}}\hat{F}^{-1}\mathbf{e}_M)^{-1}$, $\mathbf{u} = \hat{F}^{-1}\mathbf{e}_M$, and $\mathbf{v}^{\mathrm{T}} = \mathbf{e}_M^{\mathrm{T}}\hat{F}^{-1}$.

For the construction of $\hat{F}^{-1}$ we observe that F resembles a discretization of the kernel of a single-layer potential, whose inverse is the Laplacian operator. One could therefore expect that the matrix F, and perhaps also the matrix $\hat{S}$, has a condition number which grows linearly with M. (In Section 8.2 we shall see that this can indeed be the case.) Fortunately, there exists easily computed approximations to inverses of discretized single-layer potentials based on geometrically local properties. We shall use a modification of a construction suggested by Nabors et al. [21]. Our $\hat{F}^{-1}$ is basically their ALGORITHM 5.1. The chief difference is that our choice of "finest-level nonempty cubes" depends on system size M, but is independent of the cube (or box in 2D) hierarchy of the fast multipole algorithm subsequently used to compute matrix–vector multiplication. Further, we do not simultaneously solve for several rows or columns in the approximate inverse, but for one at a time. Our construction reads in its entirety:

Algorithm 1. Construction of $\hat{F}^{-1}$
Let p be the integer nearest to $\mathtt{min}(M, 10\log_{10}(M))$.
Fill the $M \times M$ matrix $\hat{F}^{-1}$ with zeros.
Let $z_1,\ldots,z_M$ be the points appearing in the definition of F.
for $i = 1,\ldots,M$ **do**
  Let i_{j+1} be the index of the jth nearest point to z_i. Let $i_1 = i$.
  Form the $p \times p$ matrix G with elements $G_{\mathrm{jk}} = F_{i_j i_k}$.
  Solve the system $Gx = b$, with $b = (1,0,\ldots,0)^{\mathrm{T}}$.
  for $j = 1,\ldots,p$ **do**
    Set the i_jth element in column i of $\hat{F}^{-1}$ equal to x_j.
  end
end

Note that the matrix $\hat{F}^{-1}$ is sparse and contains at most pM nonzero elements. The formula $p = \min(M, 10\log_{10}(M))$ in Algorithm 1 is determined experimentally and found to work well for a wide range of domains.

We proceed with a numerical experiment in order to illustrate the efficiency of $\hat{S}_P^{-1}$ of (37). We compare the number of GMRES iterations needed to solve a sequence of systems

$$\hat{S}\mathbf{x} = \mathbf{b} \tag{38}$$

for **x**, with and without right preconditioning. The geometric data used to construct the matrices $\hat{S}$ in these systems is generated from the test domains described in Section 7, below. The elements of **b** are chosen randomly in [−1,1]. Note that, for a given test domain, the system (38) and the inner system (35) have the same system matrices – only the right hand sides differ. If $\hat{S}_P^{-1}$ is efficient for (38), where the right hand side is random, one could expect that $\hat{S}_P^{-1}$ also is efficient for (35), where the right hand side varies with the iterations in the main system. The number of iterations needed to reach a relative error of less than 10^{-8} in the residual of (38) is shown as a function of system size M in Fig. 1. The relative errors in the solutions are shown in Fig. 2. The reference solutions for **x** are computed by solving the unpreconditioned systems down to relative residuals of less than 10^{-12}. Fig. 1 shows that the number of iterations for the unpreconditioned system grows approximately as $M^{0.3}$ while the number of iterations for the preconditioned system grows much slower. The speedup in the preconditioned system is about a factor of 20 for large M. This should be very promising for Mode III. Fig. 2 shows that the computed accuracy of the solution in the preconditioned system always is higher than the requested tolerance in the solver. The accuracy seems to increase with system size.

To sum up Mode III: each iteration for the inner system requires $O(M\log(M))$ operations. If there are k_{initer} such iterations, the total cost of one iteration for the main system is $O(N + k_{\text{initer}}M\log(M))$. We have reasons to believe that k_{initer} grows only slowly with M for many problems of interest. The initial cost for setting up the preconditioner is $O(M(\log(M))^3)$.

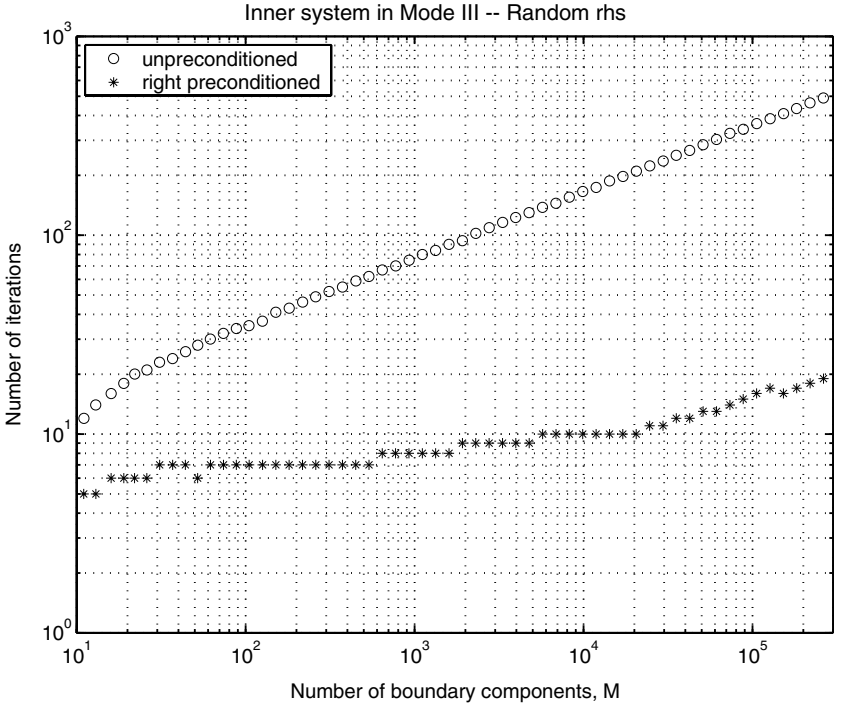

Fig. 1. Number of iterations for (38). The effect of right preconditioning is studied for a sequence of systems with random right hand sides.

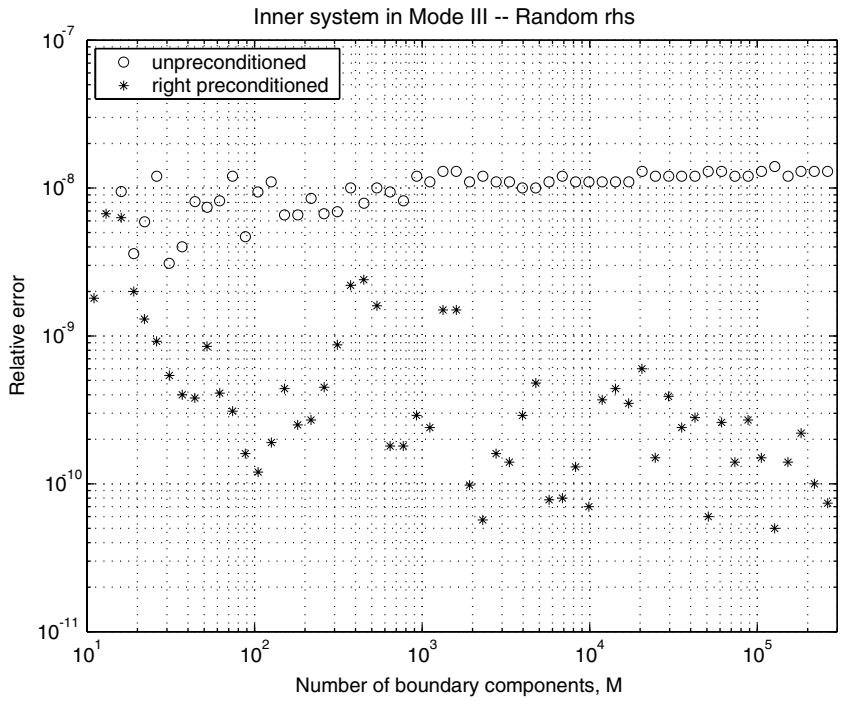

Fig. 2. Accuracy of solutions for (38).

6. The Dirichlet–Neumann map

All three modes in Section 5 compute quantities from which the solution to Laplace's equation in multiply connected exterior domains, that is, the mapping $f(Q) \to U(P)$, can be directly evaluated via (5), (6) and (8) or (10). In order to get the Dirichlet–Neumann map $f(Q) \to g(Q)$, as we set out to do in Section 2, we also need to compute the normal derivative (4) of $U(P)$ on L, that is, the mapping $U(P) \to g(Q)$, and then compose these two maps.

The function $U(P)$ represented by (5), (6) and (8) or (10) is a sum of logarithmic terms, the real part of a Cauchy-type integral, and a constant. The normal derivative of the logarithmic terms can be computed analytically. The normal derivative of the real part of a Cauchy-type integral, in this case the real part of $\varphi^*(z)$ of (8) or (10), can be computed numerically as the tangential derivative of its harmonic conjugate. We obtain the harmonic conjugate by computing $\Im\{\varphi^*(\zeta)\}$ on L using the alternate point trapezoidal rule, see [5] for details. The tangential derivative is obtained by Fourier approximation and the FFT. We use the routines DCFFTF and DCFFTB from Netlib.

7. The test problems

The test domains in our numerical examples are domains exterior to M ellipses. A typical domain is shown in Fig. 3. The area fraction of the ellipses is around 29%. The ellipses are placed sufficiently separated as to ensure a relative accuracy of 10^{-8} in the Dirichlet–Neumann map using 128 discretization points per boundary component L_k. The formal construction of the test domains, used in the numerical experiment of Section 5.3.2 and in the large-scale problems of Sections 8.2 and 8.3, reads

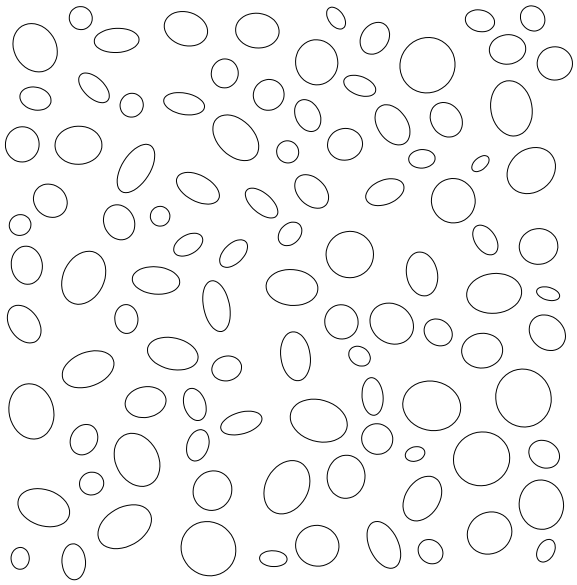

Fig. 3. A typical domain with 100 boundary components.

1. Let $r_{\min} = \sqrt{1/(35M)}$ and let $\gamma = 1.35$.
2. Place M circles C_k in the unit square:
   - Choose M circle radii randomly in $[r_{\min}, 3r_{\min}]$.
   - Construct a sorted vector $\mathbf{r}$ of circle radii in descending order.
   - Place M circles C_k with radii r_k, sequentially picked from $\mathbf{r}$, randomly in the unit square in such a way that the distance d_{jk} between the centers of any two circles C_j and C_k is at least $d_{jk} = \gamma\max(r_j, r_k) + \min(r_j, r_k)$.
3. Place an ellipse E_k inside each circle C_k:
   - Choose the inclination angle, θ_k, of ellipse E_k randomly in $[0, 2\pi]$.
   - Set the length of the major axis, α_k, of ellipse E_k equal to r_k.
   - Choose length of the minor axis, β_k, of ellipse E_k randomly in $[0.5\alpha_k, \alpha_k]$.

The points z_k, introduced in (6) and needed for the construction of various preconditioners, are placed at the centers of the ellipses E_k.

As for Dirichlet boundary data $f(Q)$ in the test problems, we choose the Gibbs–Thomson boundary condition

$$f(\tau) = \kappa(\tau), \quad \tau \in L, \tag{39}$$

where $\kappa(\tau)$ denotes curvature. This type of boundary condition is typical in certain material studies. One example is the simulation of particle coarsening during Ostwald ripening, where the evolution of interface boundaries is determined by the Dirichlet–Neumann map [1,2,5,28].

8. Numerical results

This section presents numerical examples comparing the performance of, and highlighting the differences between, the three computational modes of Section 5. The underlying computer programs are implemented in Fortran 77. The numerical experiments are performed on a SunBlade 100 workstation.

8.1. Example 1 – Verification and reproduction of earlier results

We start with the same setup as in Example 1 of Greenbaum et al. [5]. The Dirichlet problem (1)–(3) is solved in a domain exterior to six ellipses, see Fig. 4. Each boundary component L_k, $k = 1,\ldots,6$ is parameterized by:

$$\begin{aligned} x(\phi) &= c_x + a\cos\theta\cos\phi - b\sin\theta\sin\phi, \quad 0 \leqslant \phi \leqslant 2\pi, \\ y(\phi) &= c_y + a\cos\theta\sin\phi + b\sin\theta\cos\phi, \quad 0 \leqslant \phi \leqslant 2\pi. \end{aligned} \tag{40}$$

The boundary data $f(Q)$ is obtained by choosing an exact solution of the form

$$U(P) = c + \sum_{k=1}^{6} d_k \log(|z - s_k|^2), \tag{41}$$

where the points s_k lie inside the ellipses. The precise centers (c_x,c_y), eccentricities a,b, and inclination angles θ of the ellipses in (40) and the positions of the points $s_k = (s_x + \mathrm{i}s_y)_k$ in (41) are given in Table 1. The coefficients in (41) are taken as

$$c = 1.0, \quad d_k = k - \tfrac{7}{2}, \quad k = 1,\ldots,6. \tag{42}$$

The points z_k, introduced in (6), are chosen as $z_k = (c_x + \mathrm{i}c_y)_k$. In this way $z_k \neq s_k$.

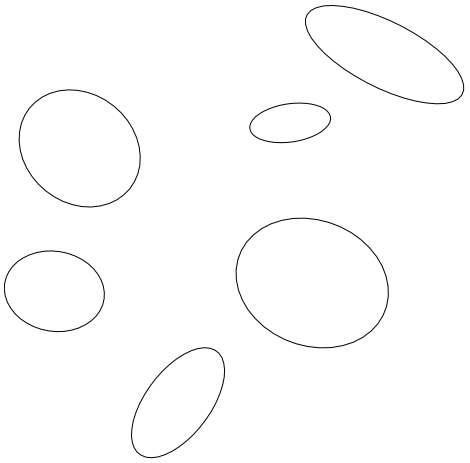

Fig. 4. The domain exterior to six ellipses used in Example 1.

Table 1
Data for Example 1. Each row contains data for one ellipse in (40) and (41)

a	b	c_x	c_y	θ	s_x	s_y
0.3626	0.1881	0.1621	0.5940	3.3108	0.1	0.5
0.5061	0.6053	−1.7059	0.3423	0.5778	−1.6	0.4
0.6051	0.7078	0.3577	−0.9846	4.1087	0.3	−0.9
0.7928	0.3182	1.0000	1.2668	2.6138	0.95	1.2
0.3923	0.4491	−1.9306	−1.0663	4.4057	−1.85	−1.0
0.2976	0.6132	−0.8330	−2.1650	5.7197	−0.8	−2.1

Table 2
Relative error in the Dirichlet–Neumann map for Mode I at a number of tolerances and refinements

N_k	Tolerances		
	10^{-4}	10^{-8}	10^{-12}
32	3.242E − 03	3.239E − 03	3.239E − 03
64	1.576E − 04	1.028E − 05	1.028E − 05
128	1.574E − 04	1.247E − 08	1.757E − 09
256	1.576E − 04	1.247E − 08	1.650E − 12

Table 3
Relative error in the Dirichlet–Neumann map for Mode II at a number of tolerances and refinements

N_k	Tolerances		
	10^{-4}	10^{-8}	10^{-12}
32	3.239E − 03	3.239E − 03	3.239E − 03
64	2.940E − 05	1.028E − 05	1.028E − 05
128	2.841E − 05	1.292E − 08	1.757E − 09
256	2.998E − 05	1.383E − 08	1.682E − 12

Table 4
Relative error in the Dirichlet–Neumann map for Mode III at a number of tolerances and refinements

N_k	Tolerances		
	10^{-4}	10^{-8}	10^{-12}
32	3.240E − 03	3.239E − 03	3.239E − 03
64	8.696E − 05	1.028E − 05	1.028E − 05
128	8.639E − 05	2.044E − 08	1.757E − 09
256	8.640E − 05	2.037E − 08	1.957E − 12

The solution to (1)–(3) and the Dirichlet–Neumann map (4) are computed, using the three modes of Section 5. We test a number of different tolerances for the relative error in the norm of the residual or pseudoresidual in the GMRES solver, and a number of refinements in the discretization. Tables 2–4 present the relative errors in the Dirichlet–Neumann map for Mode I, Mode II, and Mode III, respectively. We note that all three modes produce very similar results.

8.2. Example 2 – Accuracy for large domains

This example investigates how the accuracy in the Dirichlet–Neumann map achieved by the three modes of Section 5 depends on the number of boundary components M in a domain. The Dirichlet problem (1)–(3) is solved for a number of test problems with different sizes M, constructed according to the description in Section 7. The Dirichlet–Neumann map is computed. All GMRES tolerances (for relative errors of norms of residuals and pseudoresiduals) are set to 10^{-8} and 128 discretization points per boundary component are used. The reference solution is computed in Mode I with tolerance set to 10^{-12} and with 256 discretization points per boundary component.

The relative errors in the Dirichlet–Neumann map for the three modes are shown in Fig. 5. Here one can make a very interesting observation. The relative error for Mode II is uncontrolled as M grows, while the relative errors for Mode I and Mode III are controlled. The solution in Mode II is unsatisfactory for

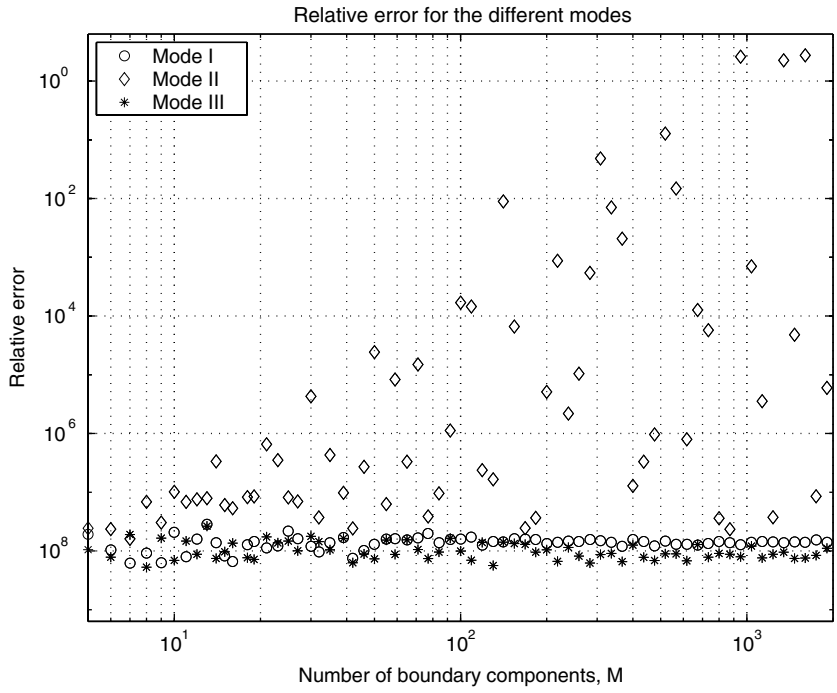

Fig. 5. Relative error in the Dirichlet–Neumann map as a function of problem size.

$M > 500$ even if we are willing to accept low precision. A partial explanation is offered in Fig. 6. This figure shows the condition numbers of the Schur complements S and $\hat{S}$, acting as system matrices in the inner systems of Mode II and Mode III, as functions of M. The Schur complement S in Mode II has a condition number that grows at an alarming rate and somehow seems to depend on the choice of the special boundary component L_M which is not included in the constraint equation on the last line of (9). The Schur complement $\hat{S}$ in Mode III has a condition number that grows linearly with M, as conjectured in Section 5.3.2. Fig. 6 thus suggests that if one wants to use the preconditioning strategy of Mode II it should, for stability reasons, be applied to our new system (11) rather than to Mikhlin's original system (9).

We note that when Mode II was used in [5], the largest problem size was $M = 200$ and the question of precision in solutions for larger problems was not addressed.

8.3. Example 3 – Large-scale computations and complexity

This example investigates how the computing time of the three modes of Section 5 depends on the number of boundary components M in a domain. The Dirichlet problem (1)–(3) is, again, solved for a number of test problems with different sizes M, constructed according to the description in Section 7. The systems are solved with all GMRES tolerances set to 10^{-8} and with 128 discretization points per boundary component. For $M < 2000$, the test problems are a subset of those in Example 2 (every other problem is chosen). The largest test domain included in the study is depicted in Fig. 7.

Timing results for the three modes are compared in Fig. 8. One can see that our new Mode III always outperforms Mode I in terms of speed and that its asymptotic complexity is better. Comparison with Mode II may not be so relevant since the solutions produced in Mode II are unreliable for large problems. Still, for small problems, where Mode II performs better, our new Mode III is only slightly slower. A comparison

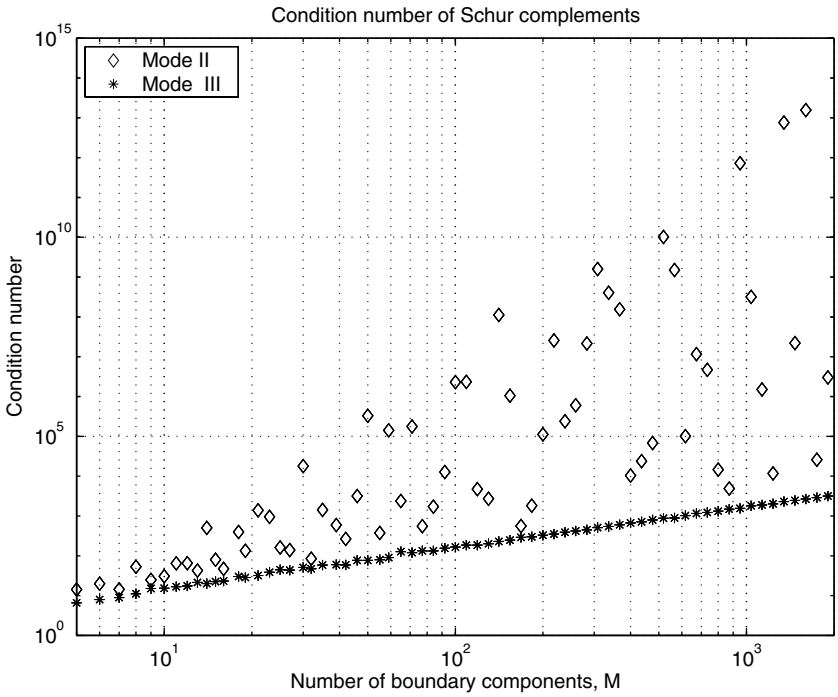

Fig. 6. Condition numbers of the Schur complements S of (24) and $\hat{S}$ of (32) acting as system matrices in the inner systems of Mode II and Mode III.

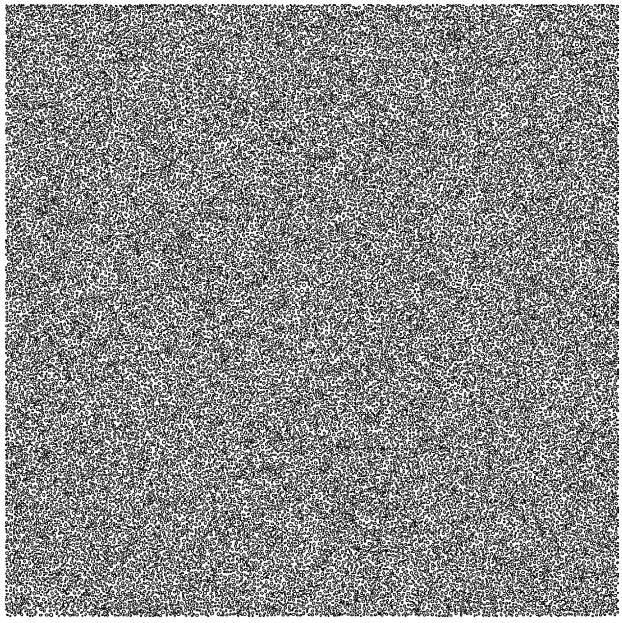

Fig. 7. A test domain containing 30,444 boundary components. A blow-up part of the domain could look like Fig. 3.

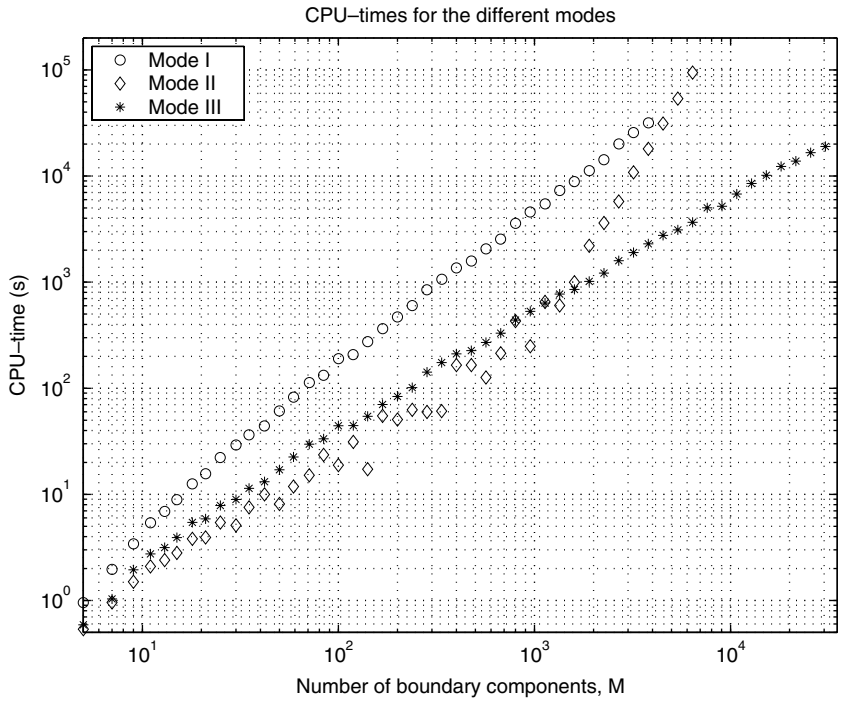

Fig. 8. Computing time in seconds as a function of problem size.

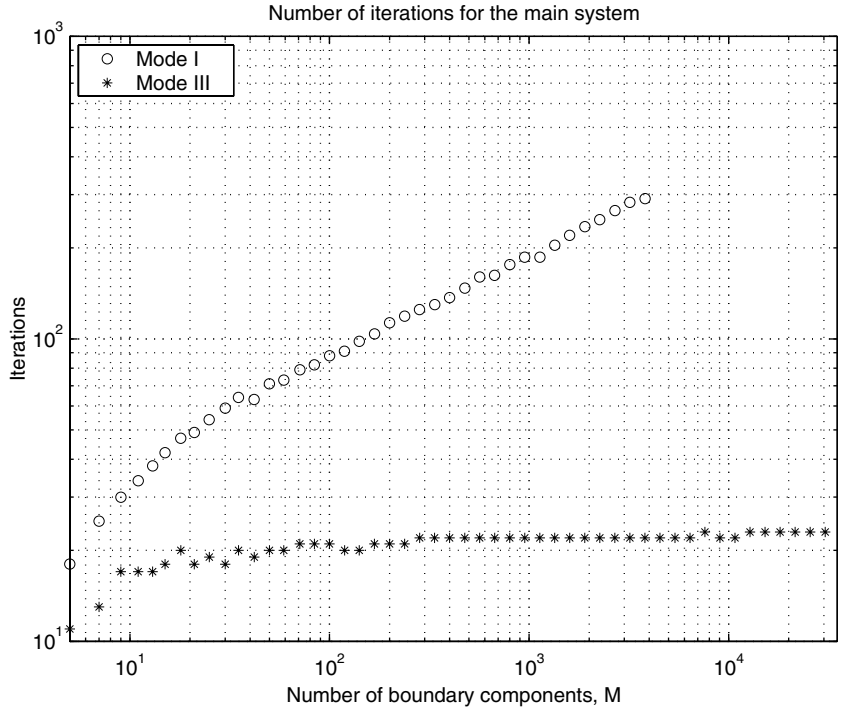

Fig. 9. Iterations for the main system as a function of problem size.

of Figs. 5, 6, and 8 shows that the wiggly shape of the curve for Mode II in Fig. 8 is correlated with the condition number of the Schur complement S and the accuracy in the solution. The seemingly rapid execution for some problem sizes M is a consequence of that few iterations are needed to meet the GMRES tolerance. The accuracy is bad for these problems and the condition number of the Schur complement S is particularly high.

The timings for Mode I and Mode III in Fig. 8 chiefly reflect the number of GMRES iterations needed for the main systems. These numbers are compared separately in Fig. 9. The number of iterations in Mode I clearly grows with problem size, although not as bad as $O(M^{0.5})$ which is a pessimistic estimate for many iterative methods on systems whose system matrices have condition numbers that grow as $O(M)$. The number of iterations for the main system in Mode III is nearly independent of M. This indicates that our alternative preconditioner $\hat{P}$ of (31) makes the system (33) behave like a discretized Fredholm second kind equation in the sense that its condition number is independent of the number of discretization points N. Note, however, that we can increase N in two different ways – either by increasing M and holding N_k constant, or by increasing N_k and holding M constant. In the latter situation, which is more honest test for if a certain equation behaves as if it were of Fredholm second kind but less relevant for large-scale problems, Mode I and Mode III behave similarly.

The number of iterations needed to solve the inner system (35) in Mode III is shown as a function of M in Fig. 10. (The largest number for all main iterations at a given problem size.) The dependence on M is, as predicted in Section 5.3.2, very similar to that of the system (38) shown as stars in Fig. 1. This indicates that Mode III should work well for test problems with up to several hundreds of thousands of boundary components.

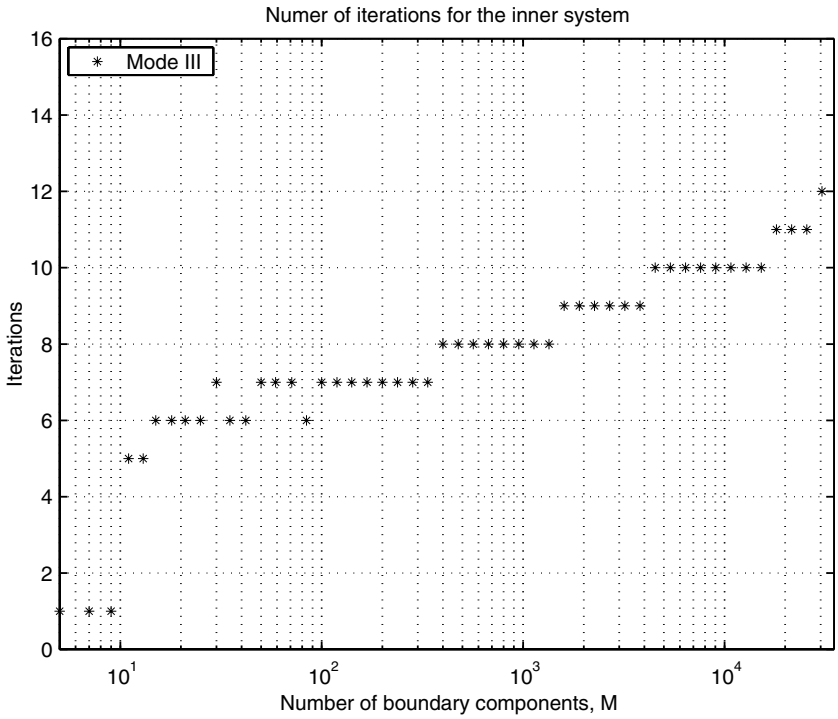

Fig. 10. Inner iterations in Mode III as a function of problem size.

9. Conclusions and extensions

We have reformulated Mikhlin's classic method [19] for the solution of Laplace's equation in multiply connected exterior domains with Dirichlet boundary data. The new formulation differs from Mikhlin's original formulation in that all boundary components are treated in the same manner and in that the discretized integral equation has one extra unknown. Based on the new formulation we have constructed a new computational mode, denoted Mode III. Our Mode III differs from Mode I of [5] in that it does use a preconditioner. It differs from Mode II of [5] in that it is stable and avoids costly LU-factorization.

Most previous investigators who need to solve the Dirichlet problem have settled for Mode II [1,2,16,28], sometimes applied on overlapping sub-problems – a strategy which introduces an error "generally better than one per cent" [1]. Neither Mode I nor Mode II is particularly good for large problems. Mode I is quite slow already for small problems and its complexity is worse than linear in M. Mode II is faster for small problems, but its complexity is cubic in M and it exhibits instabilities. In contrast, our new Mode III has almost linear complexity in M, it is always faster than Mode I, it is faster than Mode II for medium-sized and large problems (the precise cross-over size depends among other things on the requested tolerance and details in the implementation of the fast multipole method), and it seems to produce results with user specified accuracy. At least this holds for the problem class described in Section 7. It is therefore our hope that our new Mode III will replace Mode I and Mode II as the standard computational mode for Mikhlin's method when used on larger problems. On small problems, where Mode II can be justified, we hope, for stability reasons, that it will be applied to the more symmetric and new system (11) rather than to the original system (9).

We end with a few words regarding future work. An interesting possibility, mentioned in Section 3, is to abandon Mikhlin's method completely and to represent the solution to Laplace's equation as a combination of single and double layer potentials. Judging from the experiments presented in [17] it seems as if algorithms based on this approach behave similarly as Mode I. Another idea concerns the removal of arbitrariness within Mikhlin's method. The representation of the solution requires that a point z_k is placed inside each boundary component L_k. What is the optimal placement of this point? Replacing (6) by, for example,

$$\varphi(z) = \varphi^*(z) + \sum_{k=1}^{M} a_k \int_{L_k} \log(z-\zeta)\,\mathrm{d}\sigma, \quad z \in D, \tag{43}$$

would remove the need for z_k here, although some kind of points z_k would be needed for the construction of the preconditioner $\hat{P}$ of (31). Removal of arbitrarily placed points, used by classic authors when finding Fredholm second kind equations for other boundary value problems of potential theory, has previously shown to be successful [10,11]. Possible further applications of the ideas put forward in this paper include elasticity and Stokes flow [3,6,7,13,24], where Sherman's method, in constrained formulation, could be presented with a new computational mode, and problems where the Dirichlet data is given on open arcs [14,15].

Appendix A. Connection between the systems (9) and (11)

Consider the systems (9) and (11). Given a solution $\mu(\zeta)$, a_k to (9) we can construct a solution to (11). We commence by defining

$$c_0 \equiv \frac{1}{2\pi}\int_L \mu(\zeta)\,\mathrm{d}\sigma = \frac{1}{2\pi}\int_{L_M} \mu(\zeta)\,\mathrm{d}\sigma. \tag{A.1}$$

We then define $\mu^*(\zeta) : L \to \mathbb{R}$ by

$$\mu^*(\zeta) \equiv \begin{cases} \mu(\zeta), & \zeta \in L_k, \quad k = 1, \ldots, M-1, \\ \mu(\zeta) - \frac{2\pi}{|L_M|} c_0, & \zeta \in L_M. \end{cases} \tag{A.2}$$

It can be verified that $\mu^*(\zeta)$, c_0 together with the coefficients a_k solve the system (11). Conversely, given a solution $\mu^*(\zeta)$, c_0, a_k to (11), it is possible to construct a solution to (9) by keeping the coefficients a_k and using (A.2) to define $\mu(\zeta)$.

It is proven in paragraph 31 of [19] that (9) has a unique solution. Assume now that we have two solutions to (11); $\{\mu^{*(1)}, c_0^{(1)}, a_k^{(1)}\}$ and $\{\mu^{*(2)}, c_0^{(2)}, a_k^{(2)}\}$. From each of these solutions it is possible to construct a solution to (9); $\{\mu^{(1)}, a_k^{(1)}\}$ and $\{\mu^{(2)}, a_k^{(2)}\}$. Uniqueness of the solution to (9) directly gives that:

$$\mu^{(1)} = \mu^{(2)}, \tag{A.3}$$

$$a_k^{(1)} = a_k^{(2)}, \quad k = 1, \ldots, M. \tag{A.4}$$

A simple calculation shows that $c_0^{(1)} = c_0^{(2)}$:

$$2\pi c_0^{(1)} = \frac{1}{|L_M|} \int_{L_M} 2\pi c_0^{(1)} \, \mathrm{d}\sigma + \int_{L_M} \mu^{*(1)}(\zeta) \, \mathrm{d}\sigma = \int_{L_M} \mu^{(1)}(\zeta) \, \mathrm{d}\sigma, \tag{A.5}$$

$$2\pi c_0^{(2)} = \frac{1}{|L_M|} \int_{L_M} 2\pi c_0^{(2)} \, \mathrm{d}\sigma + \int_{L_M} \mu^{*(2)}(\zeta) \, \mathrm{d}\sigma = \int_{L_M} \mu^{(2)}(\zeta) \, \mathrm{d}\sigma. \tag{A.6}$$

Finally we get that $\mu^{*(1)}(\zeta) = \mu^{*(2)}(\zeta)$:

$$\mu^{*(1)}(\zeta) = \begin{cases} \mu^{(1)}(\zeta), & \zeta \in L_k, \quad k = 1, \ldots, M-1, \\ \mu^{(1)}(\zeta) - \frac{2\pi}{|L_M|} c_0^{(1)}, & \zeta \in L_M, \end{cases} \tag{A.7}$$

$$\mu^{*(2)}(\zeta) = \begin{cases} \mu^{(2)}(\zeta), & \zeta \in L_k, \quad k = 1, \ldots, M-1, \\ \mu^{(2)}(\zeta) - \frac{2\pi}{|L_M|} c_0^{(2)}, & \zeta \in L_M. \end{cases} \tag{A.8}$$

It is now possible to conclude that the solution to (11) is unique.

References

[1] N. Akaiwa, D.I. Meiron, Two-dimensional late-stage coarsening for nucleation and growth at high-area fractions, Phys. Rev. E 54 (1) (1996) R13–R16.

[2] N. Akaiwa, K. Thornton, P.W. Voorhees, Large-scale simulations of microstructural evolution in elastically stressed solids, J. Comput. Phys. 173 (1) (2001) 61–86.

[3] G. Biros, L. Ying, D. Zorin, A fast solver for the Stokes equation with distributed forces in complex geometries, J. Comput. Phys. 193 (1) (2003) 317–348.

[4] J. Carrier, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm for particle simulations, SIAM J. Sci. Stat. Comp. 9 (4) (1988) 669–686.

[5] A. Greenbaum, L. Greengard, G.B. McFadden, Laplace's equation and the Dirichlet–Neumann map in multiply connected domains, J. Comput. Phys. 105 (2) (1993) 267–278.

[6] L. Greengard, M.C. Kropinski, Integral equation methods for Stokes flow in doubly-periodic domains, J. Eng. Math. 48 (2) (2004) 157–170.

[7] L. Greengard, M.C. Kropinski, A. Mayo, Integral equation methods for Stokes flow and isotropic elasticity in the plane, J. Comput. Phys. 125 (2) (1996) 403–414.

[8] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (2) (1987) 325–348.

[9] J. Helsing, Thin bridges in isotropic electrostatics, J. Comput. Phys. 127 (1) (1996) 142–151.

[10] J. Helsing, On the interior stress problem for elastic bodies, J. Appl. Mech. T. ASME 67 (4) (2000) 658–662.
[11] J. Helsing, A. Jonsson, Stress calculations on multiply connected domains, J. Comput. Phys. 176 (2) (2002) 456–482.
[12] H.V. Henderson, S.R. Searle, On deriving the inverse of a sum of matrices, SIAM Rev. 23 (1) (1981) 53–60.
[13] T.Y. Hou, J.S. Lowengrub, M.J. Shelley, Boundary integral methods for multicomponent fluids and multiphase materials, J. Comput. Phys. 169 (2) (2001) 302–362.
[14] S.D. Jiang, V. Rokhlin, Second kind integral equations for the classical potential theory on open surfaces I: analytical apparatus, J. Comput. Phys. 191 (1) (2003) 40–74.
[15] S.D. Jiang, V. Rokhlin, Second kind integral equations for the classical potential theory on open surfaces II, J. Comput. Phys. 195 (1) (2004) 1–16.
[16] H.-J. Jou, P.H. Leo, J.S. Lowengrub, Microstructural evolution in inhomogeneous elastic media, J. Comput. Phys. 131 (1) (1997) 109–148.
[17] MadMax Optics Inc., User's Manual FMM Toolbox (2D) for MATLAB®, Version 1.1, 2002.
[18] G.B. McFadden, P.W. Voorhees, R.F. Boiswert, D.I. Meiron, A boundary integral method for the simulation of two-dimensional particle coarsening, J. Sci. Comp. 1 (2) (1986) 117–144.
[19] S.G. Mikhlin, Integral Equations and their Applications to Certain Problems in Mechanics, Mathematical Physics and Technology, second ed., Pergamon Press, London, 1964.
[20] N.I. Muskhelishvili, Some Basic Problems of the Mathematical Theory of Elasticity, P. Noordhoff Ltd, Groningen, 1953.
[21] K. Nabors, F.T. Korsmeyer, F.T. Leighton, J. White, Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory, SIAM J. Sci. Comput. 15 (3) (1994) 713–735.
[22] H. Power, The completed double layer boundary integral equation method for two-dimensional Stokes flow, IMA J. Appl. Math. 51 (2) (1993) 123–145.
[23] V. Rokhlin, Rapid solution of integral equations of classical potential theory, J. Comput. Phys. 60 (2) (1985) 187–207.
[24] D. Russell, Z.J. Wang, A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, J. Comput. Phys. 191 (1) (2003) 177–205.
[25] Y. Saad, M.H. Schultz, GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comp. 7 (3) (1986) 856–869.
[26] J. Tausch, J. White, Second-kind integral formulations of the capacitance problem, Adv. Comput. Math. 9 (1–2) (1998) 217–232.
[27] K. Thornton, J. Ågren, P.W. Voorhees, Modelling the evolution of phase boundaries in solids at the meso- and nano-scales, Acta Mater. 51 (19) (2003) 5675–5710.
[28] K. Thornton, N. Akaiwa, P.W. Voorhees, Large-scale simulations of Ostwald ripening in elastically stressed solids: I. Development of microstructure, Acta Mater. 52 (5) (2004) 1353–1364.