SIAM J. Sci. Comput.
Vol. 19, No. 3, pp. 933–952, May 1998

A FAST POISSON SOLVER OF ARBITRARY ORDER ACCURACY IN RECTANGULAR REGIONS*

A. AVERBUCH†, M. ISRAELI‡, AND L. VOZOVOI†

Abstract. In this paper we propose a direct method for the solution of the Poisson equation in rectangular regions. It has an arbitrary order accuracy and low CPU requirements which makes it practical for treating large-scale problems.

The method is based on a pseudospectral Fourier approximation and a polynomial subtraction technique. Fast convergence of the Fourier series is achieved by removing the discontinuities at the corner points using polynomial subtraction functions. These functions have the same discontinuities at the corner points as the sought solution. In addition to this, they satisfy the Laplace equation so that the subtraction procedure does not generate nonperiodic, nonhomogeneous terms.

The solution of a boundary value problem is obtained in a series form in $O(N \log N)$ floating point operations, where N^2 is the number of grid nodes. Evaluating the solution at all N^2 interior points requires $O(N^2 \log N)$ operations.

Key words. boundary value problem, Poisson equation, rectangular region, spectral method, corner discontinuities, polynomial subtraction

AMS subject classifications. 35J05, 45L10, 65P05

PII. S1064827595288589

1. Introduction.

An important step in the development of fast numerical solvers for elliptic equations in complicated domains is an algorithm for the solution of boundary value problems for constant coefficient elliptic equations in rectangular regions. After solving a nonhomogeneous equation with some "convenient" boundary conditions, one must solve, in a correction step, the homogeneous problem with specified (Dirichlet or Neumann) boundary conditions. As a resulting boundary distribution may not satisfy the required compatibility conditions at the corner points, singularities may arise in the process of the solution, thus destroying the convergence rate even if the final solution is smooth.

When the computational region is discretized on a $N \times N$ grid using some low-order (finite difference or finite element) scheme, the resulting system of linear algebraic equations is represented by a sparse matrix. Such a matrix can be inverted in $O(N^2)$ (or $O(N^2 \log_2 N)$ arithmetic operations using a "fast solver" [4]. However, since the method is of low order, the resolution N must be very large if a high accuracy is desired.

Application of high-order (pseudo) spectral methods, based on global expansions into orthogonal polynomials, e.g., Chebyshev polynomials, to the solution of elliptic equations, results in a full matrix problem. The cost of inverting such a matrix using the best current algorithms is $O(N^3)$ operations [2]. Besides, the accuracy decreases considerably as the dimension N^2 of the system grows due to accumulation of round-off errors.

When using the Chebyshev method, the fast computation of the expansion coefficients requires that the problem be discretized on a nonuniform grid. For time-dependent problems, when elliptic equations arise due to a time-discretization pro-

*Received by the editors July 5, 1995; accepted for publication (in revised form) July 5, 1996. http://www.siam.org/journals/sisc/19-3/28858.html

†School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel (amir@math.tau.ac.il).

‡Faculty of Computer Science, Technion, Haifa 32000, Israel.

cedure, nonuniform grids are associated with severe stability restrictions on the time step. Such grids are also more susceptible to the appearance of spurious solutions and other numerical artifacts [3].

Unlike spectral methods based on orthogonal polynomials, the Fourier method can be applied to the solution of differential equations with any resolution without losing accuracy due to accumulation of round-off errors. Since the differential operators are represented in the Fourier space by diagonal matrices, the integration is reduced to the division of the expansion coefficients by the corresponding wave numbers at the cost of $O(N^2)$ operations. The transformation from N^2 grid values to N^2 Fourier coefficients (and vice versa), using the discrete fast Fourier transform (FFT), requires $10N^2 \log_2 N$ operations.

The efficient application of the conventional spectral Fourier method is restricted to the case of periodic domains. A finite Fourier series of a nonperiodic function, having a discontinuous periodic extension, converges very slowly inside the region and exhibits $O(1)$ spurious oscillations near the boundaries (the Gibbs phenomenon). Skölermo [11] proposed a modification of the Fourier method for the solution of the Poisson equation

$$\Delta u = F \tag{1.1}$$

in the rectangle $\Omega = [0,1]^2$ with zero boundary conditions. The problem is solved in several steps:

1. Replace the forcing function F by a smoother function $\tilde{F}$. The later is obtained by subtracting a linear polynomial function, $P_1(x,y)$, having the same values at the corner points as $F(x,y)$.
2. Approximate the "smoothed" function $\tilde{F}$ by the trigonometric series

$$\sin k\pi x \ \sin l\pi y, \qquad k,\ l = 1, 2, \dots, N-1, \tag{1.2}$$

   which are the eigenfunctions of the Laplace operator. The expansion coefficients are computed by the fast sine transform.
3. Find the solution $\tilde{u}$ which corresponds to the "smoothed" source function using the relations between the Fourier coefficients for $\tilde{u}$ and $\tilde{F}$.
4. Find the solution, U_1, which corresponds to the polynomial subtraction function, P_1, using the same Fourier method.
5. Find the correct solution as a superposition $u = \tilde{u} + U_1$.

This algorithm gives the solution with accuracy $O(N^{-4})$.

The accuracy can be increased further, up to $O(N^{-6})$, by using a third-order subtraction polynomial, $P_3(x,y)$, in order to match both the function F and its second derivatives at the corner points. However, the algorithm becomes much more complicated and time consuming than in the previous case. Besides, the subtraction polynomial functions P_l, being themselves nonperiodic, are subject to the Gibbs phenomenon when they are represented by trigonometric series.

In [10] the homogeneous Laplace equation in the rectangular domain $\Omega = \{0 \leq x \leq \pi, 0 \leq y \leq a\}$ is solved using the Fourier method. The solution is split into four parts. Each part has nonzero boundary conditions only on one side of the domain (where it coincides with a given boundary function) and vanishes on the other three sides. The solution is constructed using separation of variables. For example, the part associated with the upper boundary has the form

$$u(x,y) = \sum_{n=1}^{\infty} a_n \frac{\sinh ny}{\sinh na} \sin nx, \tag{1.3}$$

where a_n are the Fourier coefficients of the boundary function, $f(x)$, on the upper side. The trigonometric series in (1.3) converges pretty fast inside the region due to the exponential factor $(\sinh ny)/(\sinh na) \sim e^{-n(a-y)}$. However, close enough to the boundary $y = a$ this series converges very badly for a general (not periodic) function $f(x)$, due to the Gibbs phenomenon.

The convergence of the trigonometric series is improved in [10] by replacing the solution u with a smoother one, $u^*(x, y) = u(x, y) - U(x, y)$. The function U is constructed such that it has the following properties:

1. U is a harmonic function, that is, $\Delta U = 0$;
2. it vanishes on all sides of the domain except for the upper side;
3. on the upper side, $U(x, a) = g(x)$ so that $f^*(x) = f(x) - g(x)$ and $f^*(0) = f^*(\pi) = 0$.

In the case of Dirichlet boundary conditions, the function $U(x, y)$ has been constructed in the form

$$U(x, y) = \phi(re^{-(a-y)}, x) - \phi(re^{-(a+y)}, x),$$

where $|r| < 1$, and

$$\phi(re^{-y}, x) = \text{Im } \left\{\log(1 + re^{iz})\right\}, \qquad z = x + iy.$$

A similar construction has been performed in the case of Neumann boundary conditions. For the "smoothed" function u^*, according to estimates of [10], a few hundred terms in the trigonometric expansion should give six-decimal accuracy apart from the boundary $y = 0$, and a few thousand terms will give this accuracy near a.

In this paper we propose a direct high-order method for the solution of the Poisson equation. It incorporates two main steps.

1. Compute a particular solution of (1.1), u_p, with some convenient (for example, zero or periodic) boundary conditions.
2. Compute a solution of the Laplace equation, u_h, such that upon adding it, $u = u_p + u_h$, one has a solution of the Poisson equation with the correct boundary conditions.

Thus, we concern ourselves with boundary conditions only when dealing with the Laplace equation.

In the first step, we construct the particular solution using the local Fourier basis (LFB) method of [5], [14]. The idea of this method is to project the source function in a smooth way on an extended domain. After that, we expand a smoothed source function into the rapidly converging Fourier series. Some details of this method are given in Appendix A. The cost of this step is $O(N_e^2 \log N_e)$ operations, where $N_e = N + 2n$ and $n = O(1)$.

In the second step, we solve an auxiliary boundary value problem to correct for the true boundary conditions. As in [11], [10] we use a subtraction technique in order to accelerate convergence of the Fourier series. The basic idea of our approach consists of removing all discontinuities of the boundary conditions which arise due to nonperiodicity of the boundary functions and nonsmoothness of the boundary. The later results in incompatibility of the boundary function with the homogeneous (Laplace) equation at the corner points (for example, the sum of second derivatives at the corner point is not equal to zero). The subtraction functions are constructed in the form of harmonic polynomial functions, z^n and $z^{2k} \log z$, so that the subtraction procedure does not generate nonperiodic, nonhomogeneous terms. The solution of a boundary value problem is obtained in a series form in $O(N \log N)$ arithmetic operations.

Evaluating the solution at all N^2 points in the whole domain requires $O(N^2 \log N)$ operations. For the fairly smooth (but not periodic) source function and boundary conditions, the solution can be computed with machine accuracy (in double precision) using some tens points in each direction.

The paper is organized as follows. In section 2, we construct a special solution to Dirichlet (or Neumann) boundary value problems based on separation of variables for periodic boundary conditions. In section 3, we introduce a subtraction polynomial technique generalized to be usable for two-dimensional problems. This technique is implemented to transform a generic nonperiodic problem to a periodic one with subsequent application of the foregoing algorithm. The harmonic subtraction function is constructed in section 4 in the case of an analytic solution and in a nonanalytic case, when the solution has singularities at the corner points. Finally, in section 5, the algorithm for the solution of the Poisson equation is described. The performance of the present method is compared with that of the recent adaptive multidomain spectral algorithm by Lee and Greengard [7].

2. Formulation of a problem. Periodic boundary conditions. We are interested in the solution of the Laplace equation

$$\Delta u = 0 \tag{2.1}$$

in the rectangle $\Omega = [0, 1]^2$.

The boundary conditions may be either of the Dirichlet

$$u = \Phi(x, y) \text{ on } \quad \partial\Omega \tag{2.2}$$

or of the Neumann type

$$\frac{\partial u}{\partial n} = \Phi(x, y) \text{ on } \quad \partial\Omega, \tag{2.2a}$$

where n is the internal normal to $\partial\Omega$. The mixed Dirichlet/Neumann-type boundary conditions can be considered as well.

The boundary functions

$$\begin{aligned} &\phi_1(x) = \Phi(x, 0), && \phi_3(x) = \Phi(x, 1), \\ &\phi_2(y) = \Phi(0, y), && \phi_4(x) = \Phi(1, y) \end{aligned} \tag{2.3}$$

are assumed to be smooth on each side, otherwise they are arbitrary. In particular, the function $\Phi(x, y)$ is not necessarily continuous at the corner points and/or it may not be compatible with the equation at these points.

The basis for the suggested approach is the use of rapidly convergent series expansions in terms of *harmonic functions* as follows:

$$u(x, y) = \sum_{n=0}^{M} c_n h_n(x, y), \tag{2.4}$$

where $h_n(x, y)$ satisfy the same Laplace equation as the solution $u(x, y)$. The coefficients c_n have to be found such that the boundary conditions (2.2) (or (2.2a)) are satisfied.

First we describe the algorithm as applied to the Dirichlet boundary value problem. Suppose that the boundary functions ϕ_i have smooth (C^∞- continuous) odd periodic extensions beyond the corner points and $\phi_i = 0$ at these points. Then we

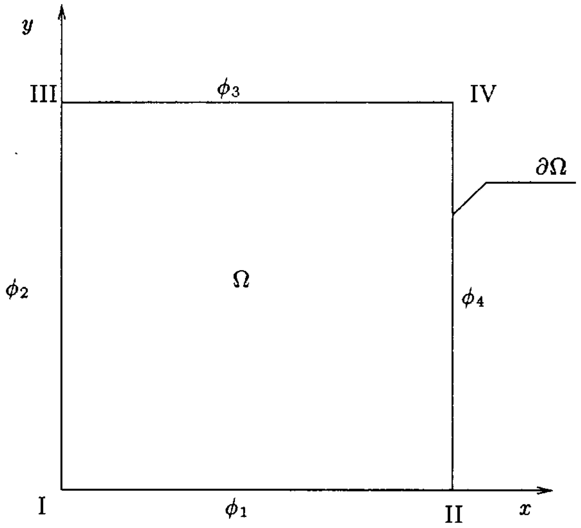

FIG. 1. *Computational region and the associated boundary functions.*

can approximate the solution to (2.1), (2.2) by the (exponentially convergent) finite trigonometric/hyperbolic series as follows:

(2.5)

$$u(x,y)=\sum_{n=1}^{N}c_n^{(1)}h_n(x,1-y)+\sum_{n=1}^{N}c_n^{(2)}h_n(y,1-x)+\sum_{n=1}^{N}c_n^{(3)}h_n(x,y)+\sum_{n=1}^{N}c_n^{(4)}h_n(y,x).$$

The basis functions $h_{n,m}$ are chosen to be the solutions of (2.1). They can be constructed in the form

$$h_n(x,y)=\sin\lambda_n x\,\frac{\sinh\lambda_n y}{\sinh\lambda_n},\qquad \lambda_n=2\pi n. \tag{2.6}$$

Each sum in (2.5) represents a function which has nonzero boundary values on one side of the rectangle in Fig. 1 and vanishes on the other three sides. Thus, a function corresponding to the first sum has a nontrivial distribution on the boundary $y=0$, the second sum on the boundary $x=0$, and so on.

The numerical algorithm consists of the following steps.

ALGORITHM I.

1. Discretization. Approximate Ω by the uniform grid $\Omega_g=\{(x_i,y_j)|x_i=(i-1)\Delta,\ y_j=(j-1)\Delta, 1\le i,j\le N+1\}$, where the grid spacing is determined as $\Delta=1/N$, with N a given power of two.
2. Construction of the first sum in (2.5):
   - 2.1. Apply the discrete sine transform (DST) to the function $\phi_1(x_i)$ on the boundary $y=0$ to obtain the expansion coefficients $c_n^{(1)},\ n=1,\dots,N$.
   - 2.2. For each $y=y_j,\ j=1,\dots,N$, do: multiply $c_n^{(1)}$ by the corresponding "complimentary" hyperbolic function of y

$$\tilde{c}_n^{(1)}=c_n^{(1)}\cdot\frac{\sinh\lambda_n y}{\sinh\lambda_n};$$

transform $\tilde{c}_n^{(1)}$ back to the physical space by using the inverse DST.

The resulting solution satisfies the prescribed boundary conditions at $y = 0$ and vanishes, according to the construction, on the other three boundaries.

3. Construction of the second, third, and fourth sums in (2.5). Perform the operations of step 2 successively for the other three boundaries. □

Notice that each sum in (2.5) is constructed independently using the expansion coefficients of the boundary distribution on a corresponding side.

Step 2.1 requires $5N \log_2 N$ operations. Step 2.2 requires $O(N^2)$ multiplications and N times application of the DST routine of size N, total of $5N^2 \log_2 N$ arithmetic operations. Therefore, the computational cost of this algorithm amounts to

$$20N^2 \log_2 N + O(N^2) \tag{2.7}$$

operations.

The case of Neumann boundary conditions (2.2a) can be treated in a similar way. Here we choose boundary functions which have the first derivative vanishing at the end (corner) points and seek the solution in the form (2.5) with the basis functions

$$h_n^I(x,y) = \cos\ \lambda_n x\, \frac{\cosh\ \lambda_n y}{\lambda_n \sinh\ \lambda_n}, \qquad \lambda_n = 2\pi n. \tag{2.8}$$

The discrete cosine transform can be used to obtain expansion coefficients $c_n^{(i)}, i = 1, \ldots, 4$, for the boundary distribution on each side of the rectangle. As in the Dirichlet case, the construction is performed *independently* for each side since the normal derivative vanishes on the other three sides due to the choice of the basis functions.

A similar construction is fulfilled in the case of the mixed Dirichlet/Neumann boundary conditions. For example, in the case $u(x,0) = \phi_1(x)$, $\frac{\partial u}{\partial y}(0,y) = \phi_2(y)$, $u(x,1) = \phi_3(x)$, $\frac{\partial u}{\partial y}(1,y) = \phi_4(y)$, and $u = 0$ at the corner points, the solution can be constructed in the form

$$u(x,y) = \sum_{n=1}^{N} c_n^{(1)} \cos\ \lambda_n x\, \frac{\sinh\ \lambda_m(1-y)}{\sinh\ \lambda_m} + \sum_{n=1}^{N} c_n^{(2)} \sin\ \lambda_n y\, \frac{\cosh\ \lambda_m(1-x)}{\lambda_m \sinh\ \lambda_m}$$

$$+ \sum_{n=1}^{N} c_n^{(3)} \cos\ \lambda_n x\, \frac{\sinh\ \lambda_m y}{\sinh\ \lambda_m} + \sum_{n=1}^{N} c_n^{(4)} \sin\ \lambda_n y\, \frac{\cosh\ \lambda_m x}{\lambda_m \sinh\ \lambda_m}. \tag{2.9}$$

As before, each sum in this expression can be computed independently (using the expansion coefficients $c_n^{(i)}$ of the corresponding boundary function ϕ_i) without affecting the other boundaries.

3. Nonperiodic boundary conditions. In the generic case of nonperiodic boundary distribution, the finite trigonometric series exhibits $O(1)$ spurious oscillations near the boundaries which decay slowly into the region (the Gibbs phenomenon). Therefore, the approximation (2.5) based on the functions (2.6), (2.8), or (2.9) converges poorly in Ω.

The rate of convergence can be considerably accelerated by removing, from the given function f, a function F such that it has the same discontinuities on the boundaries. As a result, the Fourier series of the continuous component $g = f - F$ converges much faster than that for the original discontinuous function f.

In one dimension this procedure can be implemented as follows.

Algorithm II.

1. Project $f(x), \phi \leq x \leq 1$, onto a uniform collocation grid $x_i = (i-1)\Delta, 1 \leq i \leq N+1$.
2. Compute several initial *even* derivatives $f^{(2)}, f^{(4)}, \ldots, f^{(2r)}$ at the endpoints $x = 0$ and $x = 1$ (here $f^{(s)}$ denotes the sth derivative of f). Several numerical techniques can be used for the evaluation of derivatives with high accuracy. One is based on the Fourier–Gegenbauer method [12]. Another, which is much simpler but which is a sufficiently accurate alternative, is described in Appendix A. Note that all results in the present paper are obtained using this simple procedure.
3. Subtract from $f(x)$ a known function $F(x)$, $g = f - F$, such that

$$\text{at } x = 0, 1 \qquad F = f, \ \ \frac{d^2F}{dx^2} = f^{(2)}, \ \ \frac{d^4F}{dx^4} = f^{(4)}, \ldots, \frac{d^{(2r)}F}{dx^{(2r)}} = f^{(2r)}. \tag{3.1}$$

4. Extend g antisymmetrically over a half-period as follows:

$$\tilde{g}(x) = \begin{cases} g(x), & 0 < x < 1, \\ -g(2-x), & 1 < x < 2. \end{cases} \qquad \square \tag{3.2}$$

The algorithm results in a smooth periodic function $\tilde{g}$ which can be represented by rapidly convergent trigonometric series. This approach is known as the method of A. N. Krylov ([6, Chap. 1(5)]). A particular version of this technique in one dimension was implemented in [12] for the solution of stiff differential equations with high accuracy.

We propose here a two-dimensional generalization of the subtraction technique of A. N. Krylov. The basic computational step consists of constructing the subtraction function $U(x, y)$ such that it itself satisfies the Laplace equation. Thus, the smooth part of the solution $\tilde{u} = u - U$ is also an harmonic function described by (2.1). Therefore, the algorithm in section 2 can be applied efficiently and with high accuracy in order to obtain $\tilde{u}$ and the original solution can be easily reconstructed by adding the known subtraction function U.

To summarize, the two-dimensional algorithm consists of the following steps.

Algorithm III.

1. Discretize the problem (2.1), (2.2) on a uniform grid $N_x \times N_y$.
2. Compute several initial even derivatives of the boundary functions ϕ_i at the corner points:

$$\begin{array}{lll} \text{at } \ x = 0, \ 1 & \phi_1^{(2)}, \phi_1^{(4)}, \ldots, \phi_1^{(2r)}; & \phi_3^{(2)}, \phi_3^{(4)}, \ldots, \phi_3^{(2r)}, \\ \text{at } \ y = 0, \ 1 & \phi_2^{(2)}, \phi_2^{(4)}, \ldots, \phi_2^{(2r)}; & \phi_4^{(2)}, \phi_4^{(4)}, \ldots, \phi_4^{(2r)} \end{array} \tag{3.3}$$

   (for example, using the procedure of Appendix A).
3. Construct the function $U^{(r)}(x, y)$ which satisfies (2.1) and the following conditions at the corner points:

$$\text{at } \ x = 0, \ y = 0 \qquad U^{(r)} = \phi_1, \quad \frac{\partial^2 U^{(r)}}{\partial x^2} = \phi_1^{(2)}, \ldots, \frac{\partial^{2r} U^{(r)}(x, 0)}{\partial x^{2r}} = \phi_1^{(2r)}, \tag{3.4a}$$

$$\frac{\partial^2 U^{(r)}}{\partial y^2} = \phi_2^{(2)}, \ldots, \frac{\partial^{2r} U^{(r)}}{\partial y^{2r}} = \phi_2^{(2r)}, \tag{3.4b}$$

   and similarly at the other three corners. Thus, the values of the function $U^{(r)}(x, y)$ and its several even derivatives at the corner points are matched with those of the boundary function $\Phi(x, y)$. We denote the boundary distribution $U^{(r)}|_{\partial\Omega} = \Psi^{(r)}(x, y)$.

4. Define a new function

$$\tilde{u}^{(r)} = u - U^{(r)} \tag{3.5}$$

whose boundary conditions are

$$\begin{aligned} &\tilde{\phi}_1^{(r)}(x) = \Phi(x,0) - \Psi^{(r)}(x,0), && \tilde{\phi}_3^{(r)}(x) = \Phi(x,1) - \Psi^{(r)}(x,1), \\ &\tilde{\phi}_2^{(r)}(y) = \Phi(0,y) - \Psi^{(r)}(0,y), && \tilde{\phi}_4^{(r)}(x) = \Phi(1,y) - \Psi^{(r)}(1,y). \end{aligned} \tag{3.6}$$

5. Extend the boundary functions $\tilde{\phi}_i^{(r)}$ antisymmetrically over a half-period as in (3.2).
6. Compute $\tilde{u}^{(r)}(x,y)$ by expanding it into the trigonometric-hyperbolic series (2.5) as described in the periodic case (see Algorithm I). Since the boundary functions $\tilde{\phi}_i^{(r)} \in C^{2r+1}$, the trigonometric series converge at least as $O(N^{-2(r+1)})$.
7. Reconstruct the sought solution $u(x,y)$ by summing up

$$u = \tilde{u}^{(r)} + U^{(r)}. \qquad \square \tag{3.7}$$

Constructing the subtraction function $U^{(r)}$ at all N^2 interior points requires an $O(N^2)$ operation (see section 4). Therefore, the most time-consuming step of this algorithm is the computation of a "smooth" solution $\tilde{u}(x,y)$ by $O(N^2 \log N)$ operations using Algorithm I in section 2 (step 6).

4. Construction of the subtraction function $U^{(r)}$. In this section we construct the auxiliary subtraction function, $U^{(r)}$, which satisfies the Laplace equation and matches the given boundary distribution, $\Phi(x,y)$, at the corner points as in (3.4).

We distinguish between two cases:

1. The analytic case. The distribution $\Phi(x,y)$ on the boundaries is continuous and consistent with the Laplace equation. That is, the following relations take place at the corner points:

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0, \qquad \frac{\partial^4 \Phi}{\partial x^4} - \frac{\partial^4 \Phi}{\partial y^4} = 0, \ldots \tag{4.1}$$

This corresponds to an analytic harmonic solution of (2.1).

2. The singular case. The distribution $\Phi(x,y)$ has jumps at the corner points, for example,

$$\phi_2(0) - \phi_1(0) = A \neq 0, \tag{4.2}$$

or it is incompatible with the harmonic relations (4.1). This corresponds to a singular solution of (2.1).

We consider both cases.

4.1. The analytic case. The function $U^{(r)}(x,y)$ is sought in the form

$$U^{(r)}(z) = \operatorname{Re}\left\{ \sum_{l=0}^{M} c_l^{(r)} \, z^l \right\}, \tag{4.3}$$

where $z = x + iy = \rho e^{i\theta}$ is a complex variable. Evidently, this is an analytic harmonic function which satisfies (2.1).

The complex coefficients $c_l^{(r)} = a_l^{(r)} + i b_l^{(r)}$ are uniquely determined by the "matching" conditions (3.4a), (3.4b) at the corner points. Actually, it is sufficient to match the derivatives only in one direction, say in x (relations (3.4a)); then (3.4b) holds due to the compatibility conditions (4.1). For a fixed r, we have $4\times(r+1)$ relations at four corner points. It must correspond to the number $2(M+1)$ of unknown coefficients $a_l^{(r)},\ b_l^{(r)}$ in the right-hand side of (4.3). One coefficient, b_0, must be zero for the real solution $U^{(r)}$. To balance the system we require that one of two coefficients in the highest-order term, $a_M^{(r)}$, also be zero (if we assume that $b_M^{(r)}=0$ and $a_M^{(r)}\neq 0$, then one can easily check that the resulting system is overdetermined, that is, the rows of a corresponding matrix are linearly dependent). Thus, we have $4\times(r+1)$ linearly independent relations for $2M$ unknown coefficients. It follows that $M=2(r+1)$.

For example, for $r=0$ the function $U^{(0)}$ in (4.3) writes

$$U^{(0)} = a_0 + a_1 x - b_1 y + a_2(x^2-y^2) - 2b_2 xy.$$

By setting in (3.5) $\tilde{u}^{(0)}=0$ at four corner points, we arrive at the following linear system:

$$a_0^{(0)} = \Phi_{\mathrm{I}},\ \ a_0^{(0)} + a_1^{(0)} + a_2^{(0)} = \Phi_{\mathrm{II}},$$
$$a_0^{(0)} - b_1^{(0)} - a_2^{(0)} = \Phi_{\mathrm{III}},\ \ a_0^{(0)} + a_1^{(0)} - b_1^{(0)} - 2b_2^{(0)} = \Phi_{\mathrm{IV}},$$

where subscripts $I,\ldots,IV$ denote the corresponding corner points as in Fig. 1. If we assume that $b_2^{(0)}=0$, then the last three equations in the above system are linearly dependent. By setting $a_2^{(0)}=0$, we obtain the following solution:

$$a_0^{(0)} = \Phi_{\mathrm{I}},\ \ a_1^{(0)} = \Phi_{\mathrm{II}} - \Phi_{\mathrm{I}},\ \ a_2^{(0)} = 0,$$
$$b_0^{(0)} = 0,\ \ b_1^{(0)} = \Phi_{\mathrm{I}} - \Phi_{\mathrm{III}},\ \ b_2^{(0)} = \frac{1}{2}\left(\Phi_{\mathrm{II}} + \Phi_{\mathrm{III}} - \Phi_{\mathrm{I}} - \Phi_{\mathrm{IV}}\right). \tag{4.4}$$

In the next order, $r=1$, the expansion coefficients are expressed in terms of the values of Φ and Φ'' at the corner points (prime denotes derivative with respect to x):

$$a_0^{(1)} = a_0^{(0)},\ \ a_1^{(1)} = a_1^{(0)} - \frac{1}{6}(2\Phi''_{\mathrm{I}} + \Phi''_{\mathrm{II}}),\ \ a_2^{(1)} = a_2^{(0)} - \frac{1}{2}\Phi''_{\mathrm{I}},$$
$$a_3^{(1)} = \frac{1}{6}(\Phi''_{\mathrm{II}} - \Phi''_{\mathrm{I}}),\ \ a_4^{(1)} = 0,$$
$$b_0^{(1)} = b_0^{(0)},\ \ b_1^{(1)} = b_1^{(0)} - \frac{1}{6}(2\Phi''_{\mathrm{I}} + \Phi''_{\mathrm{III}}),\ \ b_2^{(1)} = b_2^{(0)} - \frac{1}{2}(2\Phi''_{\mathrm{II}} - \Phi''_{\mathrm{III}}),$$
$$b_3^{(1)} = \frac{1}{6}(\Phi''_{\mathrm{I}} - \Phi''_{\mathrm{III}}),\ \ b_4^{(1)} = \frac{1}{24}(\Phi''_{\mathrm{II}} + \Phi''_{\mathrm{III}} - \Phi''_{\mathrm{I}} - \Phi''_{\mathrm{IV}}), \tag{4.5}$$

where $a_k^{(0)}$ and $b_k^{(0)}$ are defined in (4.4). Similarly one can obtain expressions of the expansion coefficients $a_l^{(r)},\ a_l^{(r)}$ in higher orders $r>1$.

To illustrate Algorithm III we consider the solution of (2.1), (2.2) generated by a singular charge of unit strength located at the point $x=x_0,\ y=y_0$ outside Ω: $u^{ex}(x,y)=\ln\rho\ ,\ \rho=\sqrt{(x-x_0)^2+(y-y_0)^2}$. The boundary distribution, $\Phi(x,y)$, is computed accordingly. Since the boundary functions have discontinuous periodic extensions beyond the corner points, immediate application of the Fourier transform would result in a low-order approximation. However, using the above technique we obtain very high accuracy even after a few subtraction steps.

The error in the numerical solution obtained using the subtraction function (4.3) for $r=0,1,2$ is listed in Table 1. The charge is located at $x_0=1.15,\ y_0=0.5$. The resolution is 32×32 and 64×64. The error decays as the order of smoothness r increases. The rate of convergence is higher, as expected, for larger r. The last column gives the error obtained at the same parameters in [9] using the multipole method and the trapezoidal quadrature rule for the numerical integration of integral equations.

TABLE 1

Maximum and root mean square (RMS) errors in the computed solution of the Laplace equation using the subtraction function $U^{(r)}$ in (4.3) *with* $r = 0, 1,$ *and* 2.

$$\varepsilon_{RMS} = \frac{1}{N^2}\left[\sum_{i=1}^{N}\sum_{j=1}^{N}(u_{ij} - u_{ij}^{ex})^2\right]^{1/2}$$

	$r = 0$		$r = 1$		$r = 2$		Method of [9]
$N \times N$	ε_{max}	ε_{RMS}	ε_{max}	ε_{RMS}	ε_{max}	ε_{RMS}	ε_{RMS}
32×32	8.8 (−7)	6.0 (−8)	5.6 (−9)	2.7 (−10)	3.2 (−10)	2.1 (−11)	9.0 (−5)
64×64	2.1 (−7)	7.3 (−9)	3.5 (−10)	8.3 (−12)	1.4 (−13)	3.5 (−15)	9.0 (−6)

4.2. The singular case. Suppose now that the boundary function $\Phi(x, y)$ is discontinuous at one or more corner points. In such a case any usual numerical method, including high-order methods, converges slowly as the number of degrees of freedom increases. The present approach enables one to get any prescribed order of accuracy by extracting the singular part of the solution and operating only on a smooth part. This can be achieved by using a subtraction function in the form

$$U_s(x, y) = \ln z \sum_{l=1}^{q} c_l z^l. \tag{4.6}$$

The complex coefficients c_l can be chosen such that U_s has the same discontinuity at the corner points as the boundary distribution Φ. At the same time, U_s remains a finite valued (not singular) function in $\bar{\Omega} = \Omega \bigcup \partial\Omega$.

First, consider the case when there is a jump at one corner point, say $x = 0,\ y = 0$, so that the relation (4.2) holds. This jump can be removed by subtracting the function

$$U_s(x, y) = \bar{A}\ \operatorname{Im}\{\ln z\} = \bar{A}\,\theta, \tag{4.7}$$

where $\bar{A} = 2A/\pi$. Evidently, U_s satisfies the Laplace equation as it is the imaginary part of the harmonic function $\ln z$. On the boundary $y = 0$, $U_s(x, 0) \equiv 0$ since $\theta = 0$. On the boundary $x = 0$ we have $\theta = \pi/2$ and $U_s(0, y) = A$. Therefore, the boundary distribution for the residual function $\tilde{u} = u - U_s$ does *not* have discontinuity and, thus, it can be constructed as in the regular case using Algorithm III. The solution to the original problem is completed by summing up $\tilde{u} + U_s$. The whole procedure can be carried out independently for each corner where the discontinuity takes place.

Let us assume now that the boundary distribution $\Phi(x, y)$ is continuous at the corner points, but the relations of (4.1) are violated for the second derivative. That is,

$$\phi_1''(0) + \phi_2''(0) = A_2 \neq 0. \tag{4.8}$$

An appropriate subtraction function in this case can be constructed in the form

$$U_\sigma(x, y) = \operatorname{Re}\left\{c_2 z^2 \ln z\right\}, \qquad c_2 = a_2 + ib_2. \tag{4.9}$$

After setting $a_2 = 0,\ b_2 = A_2/\pi$, we obtain

$$U_\sigma(x, y) = -\frac{A_2}{\pi}\left[2xy\ln\rho + \theta(x^2 - y^2)\right]. \tag{4.10}$$

The boundary conditions for $U_\sigma(x, y)$ are

$$\begin{aligned} &\text{at } y = 0 \qquad \theta = 0,\ U_\sigma = 0,\ \frac{\partial^2 U_\sigma}{\partial x^2} = 0,\\ &\text{at } x = 0 \qquad \theta = \frac{\pi}{2},\ U_\sigma = \frac{A_2 y^2}{2},\ \frac{\partial^2 U_\sigma}{\partial y^2} = A_2. \end{aligned} \tag{4.11}$$

Therefore, $U_\sigma(x,y)$ is continuous and finite valued everywhere, including at the pole $x=0,\ y=0$. The second derivatives satisfy the same relation

$$\text{at } \ x=y=0 \qquad \frac{\partial^2 U_\sigma}{\partial x^2}+\frac{\partial^2 U_\sigma}{\partial y^2}=A_2 \tag{4.12}$$

as that for the boundary distribution $\Phi(x,y)$. Consequently, $u-U_\sigma$ is an analytic harmonic function that can be constructed using Algorithm III.

Singularities in higher (even) derivatives can be removed in a similar way by using the subtraction functions in the form

$$U_\sigma(x,y)=\operatorname{Re}\left\{c_{2k}z^{2k}\ln z\right\}, \qquad c_{2k}=a_{2k}+ib_{2k}.$$

This technique will be employed in section 5 for the solution of the Poisson equation.

5. The Poisson equation. The solution of the Poisson equation

$$\Delta u=f \qquad \text{in } \ \Omega=[0,1]^2 \tag{5.1}$$

can be performed in several steps.

Algorithm IV.

1. Construct a particular solution u_p of (5.1) with any (arbitrary) boundary conditions.
2. Compute the difference on the boundaries

$$\tilde\Phi=\Phi-u_p \ \text{on} \ \ \partial\Omega. \tag{5.2}$$

3. Construct a solution u_h of the Laplace equation with the boundary conditions $\tilde\Phi(x,y)$ from (5.2) using the techniques of sections 2–4.
4. Find the solution of (5.1) as

$$u=u_p+u_h. \qquad \square \tag{5.3}$$

Thus, the solution of the homogeneous problem developed in previous sections is the key element of the present algorithm.

If the source function in (5.1) is a polynomial of order $n+m$

$$\Delta u=x^n y^m, \qquad n\le m, \tag{5.4}$$

then the particular solution to (5.4) can be constructed in the form of a $(m+n+2)$-degree polynomial as follows:

$$u_p=\sum_{k=0}^{[n/2]}(-1)^k\frac{n!\,m!}{(n-2k)!\,(m+2k+2)!}\,x^{n-2k}y^{m+2k+2}. \tag{5.5}$$

Any smooth function $f(x,y)$ can be approximated by the (exponentially convergent) truncated Chebyshev series

$$f(x,y)\approx\sum_{n=0}^{N}\sum_{m=0}^{N-n}\hat f_{nm}T_n(x)T_m(y). \tag{5.6}$$

Using the known matrix $\{\alpha_{nm}\}$ of the transformation

$$T_n(x)=\sum_{p=0}^{n}\alpha_{mp}x^p$$

(see [1, p. 795]), one can easily re-arrange the summation in (5.6) to obtain the coefficients $\{c_{nm}\}$ of the expansion

$$f(x,y) \approx \sum_{n=0}^{N} \sum_{m=0}^{N-n} c_{nm} x^n y^m.$$

A particular solution corresponding to this source function can be constructed as a superposition of the particular solutions in the form (5.5). The algorithm described for computation of the particular solution is similar to the "local solver" procedure proposed in [7].

The coefficients $\hat{f}_{nm}$ of the truncated Chebyshev expansion (5.7) can be obtained in $O(N^2 \log N)$ operation using the fast cosine transform provided that the collocation points are chosen in the Chebyshev nodes. On the interval $[a,b]$ these nodes are given by

$$x_j = \frac{a+b}{2} + \frac{b-a}{2} \cos \frac{\pi j}{N}, \qquad j = 0, \ldots, N. \tag{5.7}$$

Thus, the fast computation of the particular solution in the form (5.6) requires the discretization on the nonuniform grid (5.7), while the homogeneous solution is computed on equally spaced (Fourier) nodes. In this case the implementation of Algorithm IV involves the interpolation step (from Chebyshev to Fourier nodes and vice versa) in order to combine u_p and u_h.

An alternative approach for computation of the particular solution implemented in this paper, consists of expanding the source function into a finite trigonometric series along with a smoothing procedure near the boundaries (the LFB method of [5], [14]). The computational complexity of this method scales like

$$20N_e^2 \log_2 N_e + O(N_e^2), \tag{5.7.1}$$

where $N_e = N + 2n$. Here N^2 is the number of grid points in the computational domain Ω, and N_e^2 is the number of points in an extended domain $\Omega_e = \Omega \bigcup \omega$. An extra region ω, surrounding Ω, is required to perform the smoothing procedure when constructing the LFB; $n = O(1)$ is the number of points across ω. When $N \gg n$, which is always the case, the relative "extra" work (for applying the FFT on an extended region Ω_e rather than on the main region Ω) scales like $O(n/N)$. Some other technical details of the LFB method are given in Appendix B. The advantage of this approach is that both the particular and the homogeneous solutions are computed on the same uniform grid.

Combining estimations (2.7) and (5.7.1), the computational complexity of Algorithm IV is approximately

$$20N_e^2 \log_2 N_e + 20N^2 \log_2 N + O(N^2), \qquad N_e = N + 2n. \tag{5.8}$$

We illustrate this algorithm by giving several examples.

Example 1 (a constant forcing). Consider the Laplace equation with constant forcing and Dirichlet boundary conditions

$$\Delta u = 2, \qquad u|_{\partial\Omega} = 0. \tag{5.9}$$

Note that the second derivative of the corresponding solution is discontinuous at the corner points. Indeed, the sum $u_{xx} + u_{yy}$ must be zero according to the boundary conditions, while everywhere inside the region it is equal to two due to the equation.

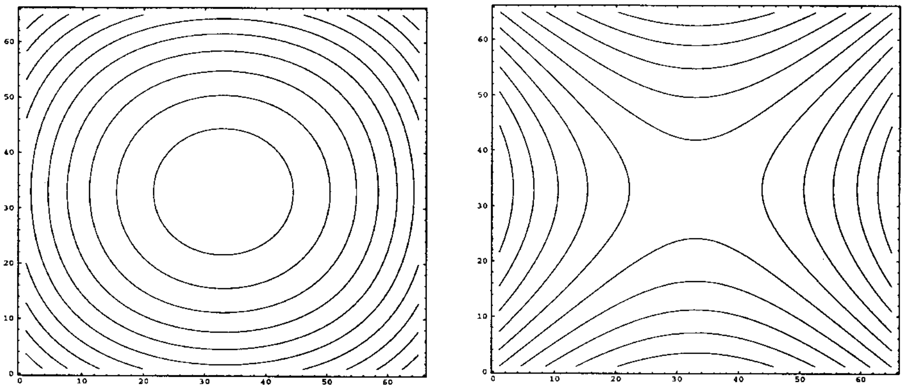

FIG. 2. *The particular solution of* (5.9) *computed by the LFB method (left). The smooth part of the homogeneous solution (right).*

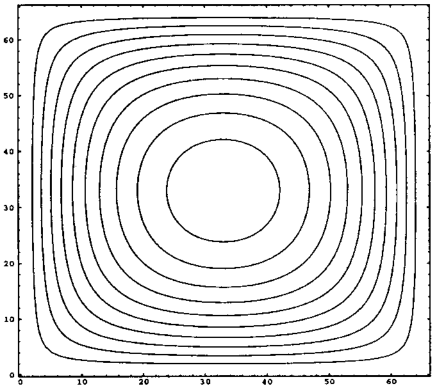

FIG. 3. *The complete solution of* (5.9).

In this simple case, the particular solution can be easily found analytically, e.g., $u_p = x^2$. However, we construct the particular solution using the general LFB approach. The plot of this solution is shown in Fig. 5 (left side).

Once the particular solution is found, the computation is reduced to the solution of the Laplace equation with Dirichlet boundary conditions $\Phi = -u_p$ on $\partial\Omega$. The boundary function Φ does not satisfy the compatibility conditions (4.1) at the corner points, which reflects the discontinuity in the second derivative of the solution u at the corner points. Therefore, the procedure of section 4.2 has to be applied at this stage in order to remove this discontinuity. After that the smooth part of the homogeneous solution is computed using Algorithm III of section 3 (it is plotted in Fig. 2, (right side)). The reconstructed solution $u = u_p + u_h$ to (5.9) is plotted in Fig. 3.

The maximum error in the numerical solution is shown in Table 2. The order of the subtraction procedure is $r = 0, 1$, and 2.

Example 2 (a Gaussian distribution). For our second example we consider the equation

$$\Delta u = 4\mu(\mu r^2 - 1)e^{-\mu r^2}, \tag{5.10}$$

in the box $\Omega = [-0.5, 0.5]^2$, where $r^2 = x^2 + y^2$. The exact solution is given by

$$u^{ex} = e^{-\mu r^2}. \tag{5.11}$$

Table 2
The maximum errors in the computed solution using Algorithm IV *for* $r = 0, 1, 2$.

$N \times N$	$r = 0$	$r = 1$	$r = 2$
8×8	2.4 (−5)	2.9 (−8)	5.0 (−10)
16×16	5.7 (−6)	1.8 (−9)	1.3 (−12)
32×32	1.4 (−6)	1.1 (−10)	1.4 (−14)
64×64	3.6 (−7)	6.7 (−12)	5.1 (−16)

Table 3
The accuracy and the operation count (# of operations $\times 10^5$*) in Example* 2 *of the present GF method and the ALC method of* [12].

	Present GF method			ALC method of [7]		
l	ε_{max}	# of points	Operat. count	ε_{max}	# of points	Operat. count
0	2.1 (−2)	$16^2 = 256$	1.2	2.31	256	1.0
1	1.7 (−7)	$32^2 = 1024$	3.6	5.1 (−4)	1024	4.0
2	6.0 (−15)	$64^2 = 4096$	13.0	7.9 (−7)	1792	6.9
3	3.2 (−15)	$128^2 = 16384$	52.6	4.7 (−11)	2560	9.9
4	4.1 (−15)	$256^2 = 65536$	224.5	3.8 (−15)	3328	12.8

The error and the approximate computational cost (according to a leading-order term in the estimate (5.8)) are listed in Table 3 in the case $\mu = 200$; the number of "extra" points (across the extended region) is $n = 8$. Other computational parameters are specified in the table.

This table also contains the results obtained for the same example in the recent work of Lee and Greengard [7]. They developed a direct Poisson solver which implements an adaptive decomposition of the computational domain (using a quad-tree data structure) in order to obtain a desired resolution near locations with sharp gradients. In each subdomain the local functions are approximated by Chebyshev expansions with the prescribed order accuracy. The coupling of the local solutions is achieved by using the singular solutions of the Laplace equation (dipole and charge layers). The effect of singularity layers is evaluated by the fast multipole method. We shall refer to this algorithm as an adaptive local Chebyshev (ALC) approach.

The computational requirements of the ALC method are approximately

$$MK^2\left(10\log K + 8K + 27\frac{p^2}{K^2} + 2K + O(1)\right). \tag{5.12}$$

Here, K is the order of the Chebyshev approximation in each subdomain, $M = N^2/K^2$ is the number of subdomains, and p is the order of the multipole expansion ($p \approx \log_2 \varepsilon$, where ε is the desired accuracy). The first term in (5.12) is the cost of the local expansion of the source function into finite Chebyshev series. The second term is the cost of evaluating the particular solution at the interface nodes. The third term is the complexity of the multipole expansion, and the fourth term is the cost of inverting the matrix of the Laplacian operator (discretized in the Chebyshev basis) by using the most efficient matrix-diagonalization approach [2]. Note that the estimate (5.12) does *not* take into account the "precomputation" work on coarser quad-tree levels. The latter consists of computing the grid values of the source function at new Chebyshev nodes and transforming from physical to spectral domain in order to evaluate the rate of convergence of the expansion coefficients.

Generally, the number of subdomains grows like $M = 3l + 1$, where l is the level of the quad-tree decomposition. In particular cases, when the source is located at

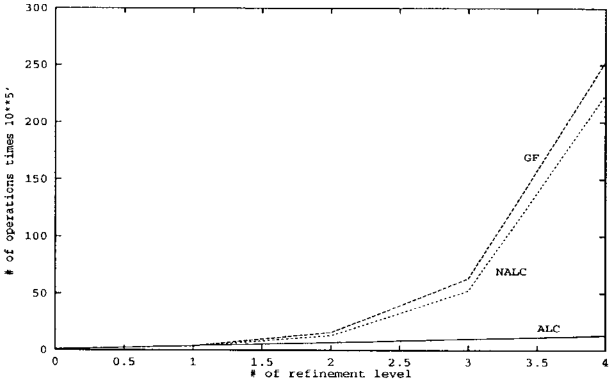

FIG. 4. *The operation count as a function of resolution for the ALC, nonadaptive local Chebyshev (NALC), and GF methods.*

the point of intersection of four subdomains, as in the present example, the number of subdomains increases much faster with l, like $M = 12(l-1)+4$. However, all estimates below are based on the previous, more optimistic relation $M = 3l + 1$.

The results in Table 3 for the ALC method are related to $p = 42$, $K = 16$, and several numbers of refinement levels l. On each new level the resolution is increased by a factor of two. For the present global Fourier (GF) method, the notion of "level" is used to denote an hierarchy of grids with the grid spacing $2^{-(l+4)}$. The operation count is based on the expressions (5.8) and (5.12).

In the particular case of (5.10) and $\mu = 200$, both the GF and the ALC methods give the maximum accuracy $\varepsilon \sim 10^{-15}$ by approximately the same number of operations, $13. \times 10^5$, while the performance of the GF method is better when obtaining a lower accuracy. The following remarks are important.

Remark 1. The advantage of an adaptive approach over a global one becomes significant for a sharper profile with larger μ. For example, for $\mu = 800$, the same accuracy as in Table 3 can be achieved by doubling the resolution (i.e., shifting by one refinement level), which follows from the scaling property of the exponential function (5.11). Then, the ALC method requires about 2.7 times less computations than the GF method. The reason is that the computational complexity of the GF method grows much faster with the number of level l than that of the ALC method; compare curves FG and ALC in Fig. 7.

Remark 2. The global nature of the Fourier method makes it particularly efficient in the case of multiple sources with high local gradients. Indeed, if a single "pick" is resolved with a certain accuracy, then approximately the same accuracy can be obtained for multiple "picks" when applying the same resolution. On the contrary, using an adaptive method one has to capture each particular "pick". Correspondingly, the resolution and CPU requirements grow as the number of "picks" increases. In effect, for numerous randomly distributed sources the ALC method loses its advantage of adaptability. Ultimately, when the sources spread out uniformly the number of subdomains on a level l of the quad-tree algorithm grows like $M = 4^l$. In such a case the computational complexity of the ALC algorithm is close to that of the GF method (curve NALC in Fig. 4).

The following example illustrates the capability of both approaches in the case of multiple sources.

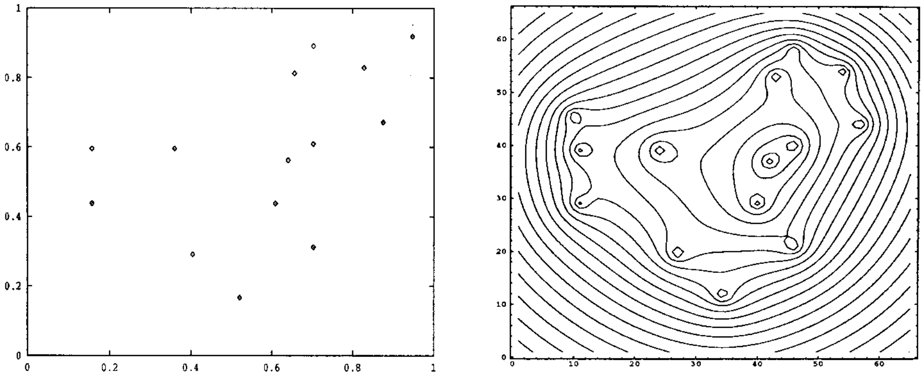

FIG. 5. *The random distribution of Gaussian sources (left) and a contour plot of the solution (right).*

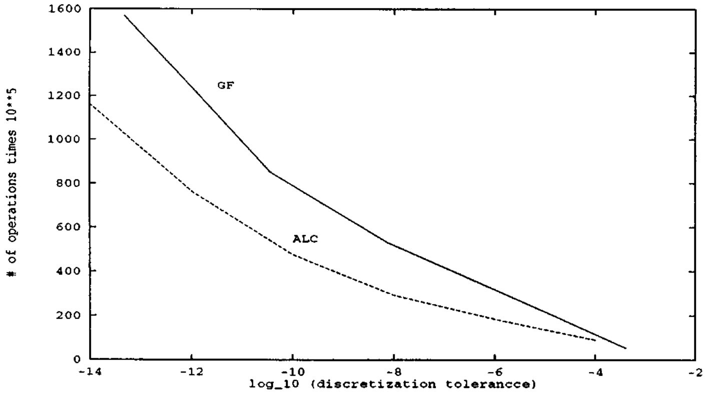

FIG. 6. *Performance of the GF and the ALC methods in Example* 3.

Example 3 (random distribution of Gaussian sources). The forcing function is formed by Gaussian sources

$$\Delta u = \sum_{i=1}^{14} e^{-\mu_i r_i^2}, \tag{5.13}$$

where $r_i^2 = (x - x_i)^2 + (y - y_i)^2$. Their centers (x_i, y_i) are randomly located in the square $[-0.4, 0.4]^2$, and $\mu_i = [1024, 16384]$. The computational domain is again the box $[-0.5, 0.5]^2$.

The distribution of the sources and the computed solution are plotted in Fig. 5.

The performance of both the GF and the ALC methods is compared in Fig. 6. The computational work is plotted as a function of the discretization tolerance of the source function in (5.13). In the ALC method an adaptive decomposition of the computational domain is performed until the local Chebyshev coefficients $\hat{f}^s_{n,m}$ of the source function $f^s(x, y)$ decay sufficiently fast according to the criterion

$$\sum_{n=0}^{K} |f^s_{n,K-n}| < \text{discretization tolerance.} \tag{5.14}$$

Here, $f^s(x, y)$ indicates the restriction of $f(x, y)$ to a subdomain s.

In the GF method, the resolution $N \times N$ is specified, while the discretization tolerance is computed as follows:

$$\text{discretization tolerance} = \frac{K}{N_e} \sum_{n=0}^{N_e} |f_{n,N_e-n}|, \tag{5.15}$$

where $N_e \times N_e$ is the number of grid points in an extended domain, and $N_e = N + 2n$. Since N_e is typically much greater than K, the coefficient $\frac{K}{N_e}$ is introduced to "equalize" the number of terms in both expressions (5.14) and (5.15). The results in Fig. 6 are computed for $K = 16$ (in the ALC method) and $n = 8$ (in the GF method).

We conclude for Example 3 that the performance of the GF method is not much worse (utmost by a factor of 1.5) than that of the ALC method. Furthermore, according to Remark 2, we expect that this method will be even more efficient for a larger number of source terms in the right-hand side of (5.13).

6. Conclusions and generalizations. The algorithm of this paper is a direct global Poisson solver in rectangular regions. It is based on a pseudospectral Fourier method and a subtraction polynomial technique. The solution procedure is essentially analytic. The main computational tool here is the FFT.

The method is shown to be fast and of high-order (in fact, arbitrary order) accuracy. Evaluating the solution at a given interior point requires $O(N \log N)$ operations. Constructing the solution at all N^2 interior points requires $O(N^2 \log N)$ operations. The performance is illustrated by solving the Poisson equation with the source function of a complicated structure. Multiple local sources with steep profiles, randomly located in the region, can be treated as efficiently as a single source. The present algorithm is particularly efficient in the case of multiple uniformly distributed sources.

The present method can be implemented (as a "local solver") in an adaptive setting, similar to [7], when combined with domain decomposition and singularity layers techniques. Preliminary results are encouraging.

This method can be also applied, with only slight modification, to the Helmholtz and modified Helmholtz equations [13].

The generalization to three dimensions is straightforward [17]. The subtraction function is constructed such that it matches the boundary distribution and its even derivatives first at the corner points, then at the edges of the cube. After that, the FFT is applied to compute the smooth part of the three-dimensional solution.

Problems with nonconstant coefficients can be treated using the present method in combination with an appropriate preconditioned iteration method.

The present Poisson solver also can be employed for the solution of problems in irregular geometries in combination with the mapping [16], [15], or the embedding [8] techniques.

Appendix A. In section 4 we constructed subtraction functions with parameters expressed in terms of known derivatives on the boundaries. Here we describe a simple numerical procedure for computation of derivatives. Given the vector of the grid values $f_j = f(x_j)$, $x_j = x_0 + j\Delta$, $\Delta = 1/N$, $j = 1, \dots, N$, compute the first p derivatives of the function f at the boundary point $x = x_0$.

Algorithm A (numerical evaluation of boundary derivatives).

1. Choose $s \geq p$ grid points, x_j, $j = 0, \dots, s-1$, adjacent to the boundary x_0.
2. Solve the linear system for s unknown values D_j

$$\sum_{k=1}^{s} D_j \, \frac{(j\Delta)^k}{k!} = f_j - f_0, \qquad j = 1, \dots, s.$$

3. Approximate $f^{(j)} \approx D_j$. □

Table 4

Relative errors in the derivatives $f^{(p)}$ of the function $f = e^{-\alpha x}$ at $x = 0$; $s = 9$, $\Delta x = 1/128$.

α	$p=1$	$p=2$	$p=3$	$p=4$	$p=5$	$p=6$
5	4.4 (−14)	6.3 (−12)	5.4 (−10)	3.3 (−8)	1.4 (−6)	4.7 (−5)
10	7.8 (−12)	5.7 (−10)	2.5 (−8)	8.0 (−7)	1.9 (−5)	3.5 (−4)
15	2.6 (−10)	1.2 (−8)	3.7 (−7)	7.8 (−6)	1.2 (−4)	1.5 (−3)
20	3.0 (−9)	1.1 (−7)	2.4 (−6)	3.8 (−5)	4.6 (−4)	4.2 (−3)

Table 5

Maximum relative errors in the Fourier approximation of the function (A.1); *the subtraction function $g^{(2r)}$ is constructed using numerical* (I) *or exact* (II) *derivatives at the boundary points.*

α_1	$r=0$	$r=1$	$r=2$	$r=3$	
	I, II	I, II	I, II	I	II
5	5.0 (−5)	6.0 (−9)	7.1 (−13)	5.9 (−15)	7.3 (−15)
10	2.0 (−4)	9.7 (−8)	4.5 (−11)	6.3 (−13)	5.6 (−13)
15	4.7 (−4)	4.9 (−7)	5.2 (−10)	6.7 (−12)	6.5 (−12)
20	8.3 (−4)	1.5 (−6)	2.9 (−9)	4.1 (−11)	3.6 (−11)

In fact, we approximate the derivatives by the corresponding divided differences. This algorithm is found to be sufficiently accurate to be consistent with the present method up to order 6–8.

We illustrate the accuracy of this procedure by computing several derivatives of the function $f(x) = e^{-\alpha x}$, $x \in [0, 1]$ at the boundary $x = 0$. Table 4 gives relative errors $|f^{(p)} - f^{(p)}_{ex}|/|f^{(p)}|_{max}$ in the first six derivatives. The mesh size $\Delta = 1/128$, the size of the stencil is $s = 9$.

The errors in the computed derivatives must affect the precision of the constructed subtraction functions and, thus, the overall accuracy of the present GF method. We illustrate this effect by computing the error in the Fourier approximation of the function

$$f(x) = \frac{\sinh\ \alpha_1(1-x)}{\sinh\ \alpha_1} - \frac{\sinh\ \alpha_2(1-x)}{\sinh\ \alpha_2}, \qquad x \in [0, 1], \tag{A.1}$$

whose periodic extension is of class C^0. First, we apply Algorithm II of section 3 to obtain a smoother function $g^{(2r)} = f - F^{(2r)}$, where $F^{(2r)}$ is a $(2r + 1)$-degree polynomial (its parameters are expressed via the first $2r$ derivatives of $f(x)$ on the boundaries $x = 0, 1$). Then we compute the Fourier coefficients $\hat{g}_k$ of a smoothed function $g^{(2r)}$ by applying the discrete fast Fourier transform (DFFT) to the grid values at the sampling points $x_j = (j - 1)\Delta$, $\Delta = 1/N$, $j = 1, \ldots, N$. Next, we reconstruct $g^{(2r)}$ in the physical space at the shifted points $\xi_j = x_j + \frac{\Delta}{2}$ (this can be done by applying the inverse DFFT routine to the "shifted" Fourier coefficients $\hat{g}_k e^{ik\tilde{\Delta}}$, where $\tilde{\Delta} = \frac{2\pi}{l}\ \frac{\Delta}{2}$, and l is the period). Finally, we reconstruct the original function $f = g^{(2r)} + F$ at the points $\{\xi_j\}$.

The results are shown in Table 5 for $N = 128$, $r = 0$ through 3, when using exact (case II) or numerical (case I) derivatives of the function (A.1) on the boundaries (see Table 4). One can conclude that the accuracy of the constructed approximation is practically the same in both cases I and II.

However, this simple procedure does not allow the computation of higher derivatives ($p > 6$) with suitable accuracy. Using a more accurate procedure would enable the MF method to construct higher-order subtraction functions in order to achieve faster convergence.

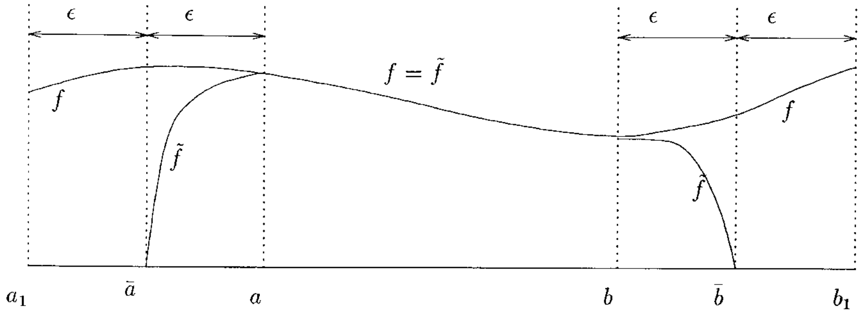

FIG. B1. *The projection operation.*

Appendix B. The detailed analysis of the LFB technique, as applied for the solution of problems in multidomain regions, is given in [5], [14]. Here we describe briefly only a part of this technique required for the solution of nonperiodic ODEs in one domain.

We illustrate the LFB approach by solving a simple one-dimensional differential equation

$$\frac{d^2u}{dx^2} = f(x), \qquad x \in [a, b]. \tag{B.1}$$

Assume that the source function $f(x)$ is known on an extended interval $x \in [a-2\epsilon, b+2\epsilon]$.

The computational algorithm consists of two main steps.

1. Project the source function $f(x)$ in a smooth way on the interval $x \in [\bar{a}, \bar{b}]$, where $\bar{a} = a-\epsilon$, $\bar{b} = b-\epsilon$. This results in a smooth function $\tilde{f}(x)$, $\sup\tilde{f} = [\bar{a}, \bar{b}]$, as in Fig. B1.
2. Apply a pseudospectral Fourier method to (B.1) with the source function $\tilde{f}$ in the right-hand side.

The projection procedure $\mathcal{P}$ consists of two "folding" operations across the extended boundaries $\bar{a}$ and $\bar{b}$:

$$\tilde{f} = \mathcal{P}f = \mathcal{F}_{\bar{a}}\mathcal{F}_{\bar{b}}f(x), \tag{B.2}$$

where the "folding" across the line $\bar{a}$ is defined as follows:

$$\mathcal{F}_{\bar{a}} = B(x)f(x) - B(2\bar{a}-x)f(2\bar{a}-x) \tag{B.3}$$

and similarly across the line $\bar{b}$. Here $B(x)$ is the window function which satisfies the following properties:

$$\sup B(x) = [a-2\epsilon,\, b+2\epsilon],$$

$$\begin{array}{ll} B^2(x) + B^2(2\bar{a}-x) = 1, & x \in [a-2\epsilon, a), \\ B(x) = 1, & x \in [a, b], \\ B^2(x) + B^2(2\bar{b}-x) = 1, & x \in (b, b+2\epsilon]. \end{array} \tag{B.4}$$

Inside the computational interval $[a, b]$ this function is equal to $B = 1$ and it smoothly decays outwards over a distance 2ϵ. Some specific forms of $B(x)$ and the influence of the size of the extra interval ϵ on the accuracy are tested in [5].

The smoothed function $\tilde{f}$ can be represented then by a rapidly convergent Fourier series

$$\tilde{f}(x) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \hat{f}_k \sin(\pi k x/\bar{l}), \qquad \bar{l} = \bar{b} - \bar{a}.$$

This allows us to apply accurately the Fourier method to the solution of (B.1) on the interval $x \in [\bar{a}, \bar{b}]$ with the smooth source function $\tilde{f}$. The expansion coefficients can be obtained by using the fast sine transform when the function $\tilde{f}(x)$ is discretized on a uniform grid.

According to the definitions (B.2), (B.3), and the properties of the window function (B.4), we have $\tilde{f} = f$ on $x \in [a, b]$. Therefore, the solution of this auxiliary problem coincides with the solution of interest $u(x)$ in the main region $x \in [a, b]$.

REFERENCES

[1] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1970.

[2] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, Berlin, New York, 1989.

[3] D. Gottlieb, *Issues in the application of high order schemes*, to appear.

[4] R. W. Hockney, *The potential calculation and some applications*, in Methods In Computational Physics 9, Academic Press, NY, 1970, pp. 135–211.

[5] M. Israeli, L. Vozovoi, and A. Averbuch, *Spectral multi-domain technique with Local Fourier Basis*, J. Sci. Comput., 8 (1993), pp. 135–149.

[6] L.V. Kantorovich and V.I. Krylov, *Approximate Methods of Higher Analysis*, Groningen, the Netherlands, 1958.

[7] J.-Y. Lee and L. Greengard, *A Direct Adaptive Poisson Solver of Arbitrary Order Accuracy*, DOE/CMCL Report 95-004, New York University, NY, 1995.

[8] A. Mayo, *The fast solution of Poisson's and biharmonic equations on irregular regions*, SIAM J. Numer. Anal., 21 (1984), pp. 285–299.

[9] V. Rokhlin, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1985), pp. 187–207.

[10] J. B. Rosser, *Fourier Series in Computer Age*, Technical Summary Report # 1401, Math. Research Center, Madison, WI, 1974.

[11] G. Skölermo, *A Fourier method for numerical solution of Poisson's equation*, Math. Comput., 29 (1975), pp. 697–711.

[12] L. Vozovoi, M. Israeli, and A. Averbuch, *Analysis and application of Fourier-Gegenbauer method to stiff differential equations*, SIAM, J. Numer. Anal., 33 (1996), pp. 1844–1863.

[13] A. Averbuch, M. Israeli, and L. Vozovoi, *On a fast direct elliptic solver by a modified Fourier method*, Numer. Algorithms, 15 (1997), pp. 287–313.

[14] L. Vozovoi, M. Israeli, and A. Averbuch, *Spectral multi-domain technique with Local Fourier Basis* II: *decomposition into cells*, J. Sci. Comput., 9 (1994), pp. 311–326.

[15] L. Vozovoi, M. Israeli, and A. Averbuch, *Application of the multidomain local Fourier method for CFD in complex geometries*, in Computational Fluid Dynamics (selected topics), D. Leutloff and R.C. Srivastava, eds., Springer-Verlag, Berlin, Heidelberg, New York, 1995, pp. 245–256.

[16] L. Vozovoi, M. Israeli, and A. Averbuch, *Multidomain local Fourier method for PDEs in complex geometries*, in J. Comput. Applied Math., 66 (1996), pp. 543–555.

[17] E. Braverman, M. Israeli, A. Averbuch, and L. Vozovoi, *A fast* 3-*D Poisson solver of arbitrary order accuracy*, J. Comput. Phys., 1998, submitted.