A Fast Poisson Solver of Arbitrary Order Accuracy in Rectangular Regions

Appeared in SIAM J. Scientific Computing, Vol. 19(3), pp. 933-952, May 1998

A. Averbuch† M. Israeli‡ L. Vozovoi‡

† School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel

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

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 large scale problems.

The method is based on a pseudo-spectral 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 non-periodic inhomogeneous terms.

The solution of a boundary value problem is obtained in the 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.

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 (b.v.p.) for constant coefficient elliptic equations in rectangular regions. After solving an inhomogeneous equation with some "convenient" boundary conditions one must solve, in a correction step, a homogeneous problem with specified (Dirichlet or Neumann) boundary conditions. As a resulting boundary distribution may not satisfy required compatibility conditions at the corner points, singularities may arise in the process of the solution, destroying the convergence rate even if the final solution is smooth.

When the computational region is discretized on a grid $N \times N$ 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", see e.g. [1].

Application of high-order (pseudo) spectral methods, based on global expansions into orthogonal, e.g. Chebyshev, polynomials to the solution of elliptic equations results in a full matrix problem. The cost of inverting of such a matrix using the best current algorithms amounts to $O(N^3)$ operations [13]. 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 is discretized on a non-uniform grid. For time-dependent problems, when elliptic equations arise due to a time-discretization procedure, non-uniform 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, see [10].

Unlike spectral methods based on orthogonal polynomials, the Fourier method can be applied to the solution of differential equations with any resolution without loosing accuracy due to accumulation of round-off errors. As the differential operators are represented in the Fourier space by diagonal matrices, the integration reduces to dividing the expansion coefficients by 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 requires $10N^2 \log_2 N$ operations.

However, the efficient application of conventional spectral Fourier methods is restricted to the case of periodic domains. Finite Fourier series of a non-periodic function, having discontinuous periodic extension, converges very slowly inside the region and exhibits $O(1)$ spurious oscillations near the boundaries (the Gibbs phenomenon). G. Sköllermo [4] 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 of 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 trigonometric series

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

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 by summing up two solutions, $u = \tilde{u} + U_1$.

This algorithm gives the solution with the 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 not periodic, are subject to the Gibbs phenomenon when they are represented by trigonometric series.

In this paper we propose a direct high order method for the solution of the Poisson equation. As in [4] we use the Fourier method in combination with the polynomial subtraction technique in order to accelerate convergence of the Fourier series. The principle difference of the present approach is that this technique is applied to the boundary function, when solving a b.v.p. rather than to the forcing function in the right hand side of Eq.(1.1).

The solution of the Poisson equation is obtained in two steps.

1. Construct the particular solution to Eq. (1.1) using the Local Fourier Basis (LFB) method of [5],[6]. The boundary conditions at this stage are not specified. The cost of this step is $O(N_e^2 \log N_e)$ operations, where $N_e = N + 2n$ and $n = O(1)$.

2. Solve an auxiliary b.v.p. The basic computational step here is the application of the FFT to special solutions of the Laplace equation which become trigonometric series when specialized to the boundaries. Fast convergence of the Fourier series is achieved by removing discontinuities at the corner points using subtraction functions. These functions satisfy the Laplace equation, so that the subtraction procedure does not generate non-periodic inhomogeneous terms.

The solution of a b.v.p. is obtained in the 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.

Finally, the solution of Eq.(1.1) is found as a superposition of particular and homogeneous parts, at the cost of $O(N^2)$ operations.

The present algorithm allows one to treat efficiently both periodic and non-periodic analytic solutions. Moreover, it can be easily modified in order to obtain rapidly convergent singular (non-analytic) solutions, having discontinuities at the corner points. These discontinuities result from the fact that relations for the second and/or higher derivatives of the boundary distribution at the corner points are incompatible with the Laplace equation. After removing discontinuities using properly constructed subtraction functions, the smooth (analytic) part of the solution is computed with high accuracy as in the "regular" case.

The paper is organized as follows. In section 2, we construct a special solution to Dirichlet (or Neumann) b.v.p. based on separation of variables, in the case of periodic boundary conditions. In section 3, we introduce a subtraction polynomial technique, generalized for the two-dimensional case. This technique is implemented to transform a generic non-periodic 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 analytic solution as well as in a non-analytic case, when the solution has singularities at the corner points.

Finally, the algorithm for the solution of the Poisson equation is described in section 5. The performance of the present method is compared with that of the recent adaptive multidomain spectral algorithm by L. Greengard and J.-Y. Lee [11].

2 A Boundary Value Problem. Periodic Boundaries

We are interested in solving the Laplace equation

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

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

The boundary conditions may be either Dirichlet

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

or Neumann type:

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

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

The boundary functions

$$\phi_1(x) = \Phi(x,0), \qquad \phi_3(x) = \Phi(x,1)$$

$$\phi_2(y) = \Phi(0,y), \qquad \phi_4(x) = \Phi(1,y) \tag{2.3}$$

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

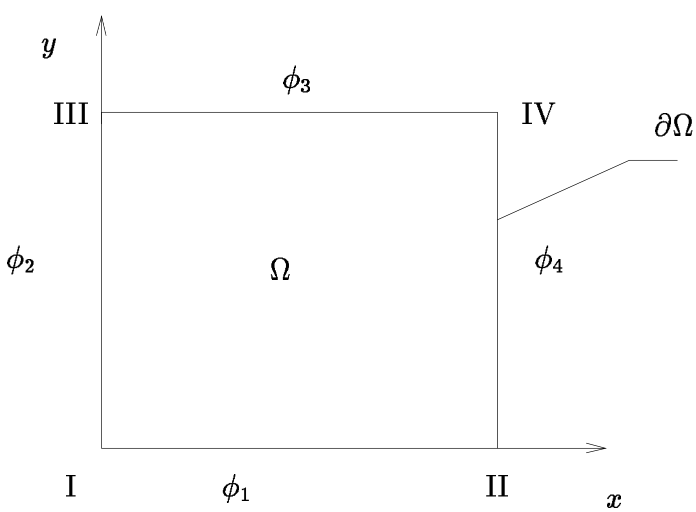


Figure 1: Computational region and boundary functions

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 b.v.p. Suppose that the boundary functions ϕ_i have smooth (C^∞- continuous) periodic extensions beyond the corner points. and $\phi_i = 0$ at these points. Then we can approximate the solution to (2.1), (2.2) by the (exponentially convergent) finite trigonometric/hyperbolic series as follows:

$$u(x,y) = \sum_{n=1}^{N} c_n^{(1)} h_n(x, 1-y) + \sum_{n=1}^{N} c_n^{(2)} h_n(y, 1-x) + \sum_{n=1}^{N} c_n^{(3)} h_n(x,y) + \sum_{n=1}^{N} c_n^{(4)} h_n(y,x). \tag{2.5}$$

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 non-zero boundary values on one side of the rectangle Fig.1 and vanishes on the other three sides. Thus, a function corresponding to the first sum has a non-trivial 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 $c_n^{(1)}$, $n = 1, ..., N$.

   2.2 For each $y = y_j$, $j = 1, ...N$ 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 resultant 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 operations of step 2 successively for 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 involves $O(N^2)$ multiplications and N times application of the DST routine of the size N, that is $5N^2 log_2 N$ 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 chose boundary functions which have zero first derivative 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 $c_n^{(i)},\ i = 1,..,4$ for the boundary distribution on each side of the rectangular. As in the Dirichlet case, the construction can be performed *independently* for each side since the normal derivative vanishes on the other three sides due to the choice of the basis functions.

Similar construction can be fulfilled in the case of 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:

$$\begin{aligned} u(x,y) = &\sum_{n=1}^{N} c_n^{(1)} \cos\ \lambda_n x\ \frac{\sinh\ \lambda_m(1-y)}{\sinh\ \lambda_m} + \sum_{n=1}^{N} c_n^{(2)} \sin\ \lambda_n y\ \frac{\cosh\ \lambda_m(1-x)}{\lambda_m\ \sinh\ \lambda_m} \\ &+ \sum_{n=1}^{N} c_n^{(3)} \cos\ \lambda_n x\ \frac{\sinh\ \lambda_m y}{\sinh\ \lambda_m} + \sum_{n=1}^{N} c_n^{(4)} \sin\ \lambda_n y\ \frac{\cosh\ \lambda_m x}{\lambda_m\ \sinh\ \lambda_m}. \end{aligned} \tag{2.9}$$

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

3 Non periodic boundaries

In the generic case of non periodic boundary distribution the finite trigonometric series exhibits $O(1)$ spurious oscillations near the boundaries, which decay slowly inwards 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 accelerated considerably by removing from the given function f a function F that has the same discontinuities on the boundaries. As a result, the Fourier series of the continuous component $g = f - F$ converges much faster than for the original discontinuous function f.

In one dimension this procedure can be implemented as follows.

Algorithm II

1. Project $f(x)$, $0 \le x \le 1$ onto a uniform collocation grid $x_i = (i-1)\Delta$, $1 \le i \le N+1$.

2. Compute several initial *even* derivatives $f^{(2)}, f^{(4)}, \ldots, f^{(2r)}$ at the end points $x = 0$ 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, see [3]. Another, much simpler but sufficiently accurate alternative is described in Appendix 1 (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

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

4. Extend anti-symmetrically g over half-period as follows:

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

$\square$

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 ([2], Chap. 1(5)). A particular version of this technique in one dimension was implemented in [3] for the solution of stiff differential equations with high accuracy.

We propose here a generalization of the subtraction technique of A.N. Krylov onto two dimensions. 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 Eq.(2.1). Therefore, the algorithm of section 2 can be applied efficiently and with high accuracy to obtain $\tilde{u}$, then the original solution can be easily reconstructed by adding the known subtraction function U.

To summarize, the algorithm in two dimensions 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:

$$\text{at } x = 0,\ 1 \qquad \phi_1^{(2)}, \phi_1^{(4)}, \ldots, \phi_1^{(2r)}; \qquad \phi_3^{(2)}, \phi_3^{(4)}, \ldots, \phi_3^{(2r)}$$

$$\text{at } y = 0,\ 1 \qquad \phi_2^{(2)}, \phi_2^{(4)}, \ldots, \phi_2^{(2r)}; \qquad \phi_4^{(2)}, \phi_4^{(4)}, \ldots, \phi_4^{(2r)}. \tag{3.3}$$

(for example, using the procedure of Appendix 1).

3. Construct a function $U^{(r)}(x, y)$ which satisfy Eq.(2.1) and the following conditions at the corner points:

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

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

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:

$$\tilde{\phi}_1^{(r)}(x) = \Phi(x, 0) - \Psi^{(r)}(x, 0), \qquad \tilde{\phi}_3^{(r)}(x) = \Phi(x, 1) - \Psi^{(r)}(x, 1)$$

$$\tilde{\phi}_2^{(r)}(y) = \Phi(0, y) - \Psi^{(r)}(0, y), \qquad \tilde{\phi}_4^{(r)}(x) = \Phi(1, y) - \Psi^{(r)}(1, y). \tag{3.6}$$

5. Extend the boundary functions $\tilde{\phi}_i^{(r)}$ anti-symmetrically over half-period as in (3.2).

6. Compute $\tilde{u}(x, y)$ by expanding 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

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

□

Constructing the subtraction function $U^{(r)}$ at all N^2 interior points requires $O(N^2)$ 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 by means of the Algorithm I (step 6).

4 Construction of the subtraction function $U^{(r)}$

In this section we construct an 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. *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

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

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

2. *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 Eq. (2.1).

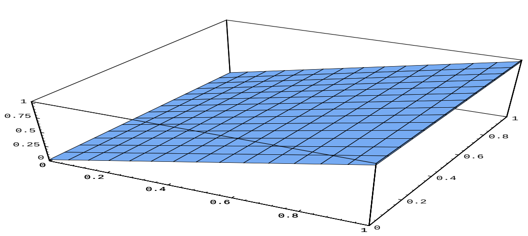
Consider both these cases.

4.1 Analytic Case

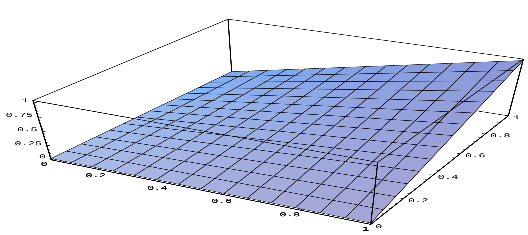
The function $U^{(r)}(x, y)$ can be sought in the form

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

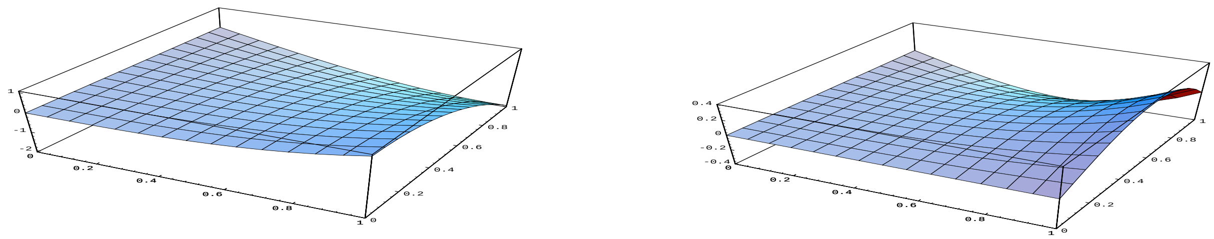
where $z = x + iy = \rho e^{i\theta}$ is the complex variable. Evidently, this is an analytic harmonic function which satisfies Eq.(2.1). Several first polynomials z_l are plotted in Fig. 2.



$\mathcal{R}e \;\; h_1^I$



$\mathcal{I}m \;\; h_2^I$



$\mathcal{R}e\ \ h_3^I$ $\qquad\qquad$ $\mathcal{I}m\ \ h_4^I$

Figure 2: Plot of several initial harmonic polynomials $h_l^I(z)$

The complex coefficients $c_l = a_l + i b_l$ are uniquely determined by the "matching conditions" (3.4), (3.4a) at the corner points. Actually, it is sufficient to match the derivatives only in one direction, say in x (relations (3.4)), then (3.4a) hold due to the compatibility conditions (4.1). The number of (complex) terms in the polynomial expansion (4.3) which is required to satisfy $4 \times (r+1)$ relations (3.4) at four corner points is $M = 2\,(r+1)$.

For example, in the lowest order $r = 0$ we have

$$a_0 = \Phi(0,0), \quad b_0 = 0, \quad a_1 = \Phi(1,0) - \Phi(0,0), \quad b_1 = \Phi(0,0) - \Phi(0,1),$$

$$a_2 = 0, \quad b_2 = \frac{1}{2}\left(\Phi(1,0) + \Phi(0,1) - \Phi(0,0) - \Phi(1,1)\right).$$

To illustrate Algorithm III we consider the solution of Eqs.(2.1), (2.2) generated by a singular charge of unit strength located at the point $x = x_0,\ y = y_0$ outside Ω: $u(x,y) = \ln\ \rho$, $\rho = \sqrt{(x-x_0)^2 + (y-y_0)^2}$. 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.

Figs. 3 and 4 show the original boundary distribution $\phi_1(x) = \Phi(x,0)$ at $y = 0$, and the "smoothed" distributions $\tilde{\phi}_1(x) = \Phi(x,0) - U^{(r)}(x,0)$ for $r = 0, 1$ and 2;

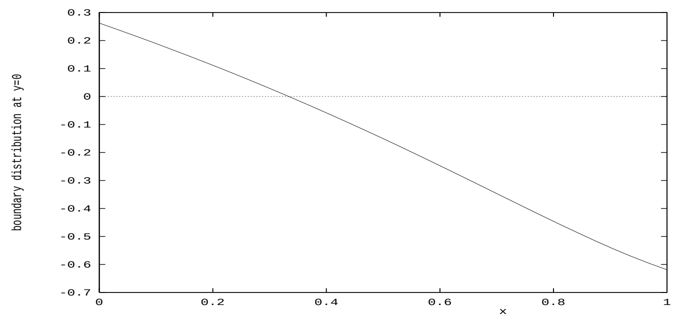


Figure 3: An original boundary function $\Phi(x, 0)$.

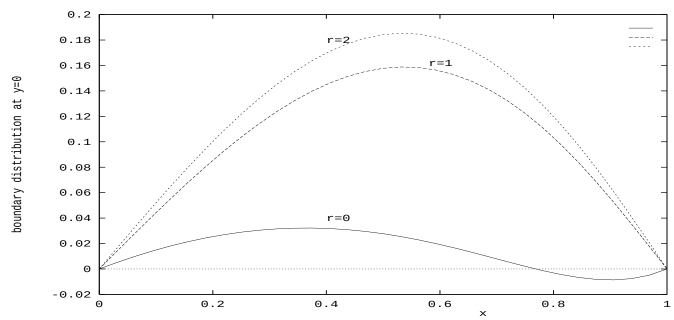


Figure 4: Distributions on the boundary $y = 0$ after subtraction a function $U^{(r)}(x, 0)$ for $r = 0, 1$ and 2.

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 [8] by the multipole method using the trapezoidal quadrature rule for the numerical integration of integral equations.

4.2 Singular Case

Suppose now that the boundary function $\Phi(x, y)$ is discontinuous at one or more corner points. In such a case any numerical method, including high order methods, converges slowly as the number of degrees of freedom grows. The present approach enables one to get

$N\times N$	$r=0$		$r=1$		$r=2$		method of [8]
	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)

Table 1: Maximum and RMS errors in the computed solution of the Laplace equation using the subtraction function $U^{(r)}$ in (4.30) with $r=0,1$ and 2.

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

$$U_s(x,y)=\ln z\ \sum_{l=1}^{q} c_l z^l. \tag{4.4}$$

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 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 $x=0,\ y=0$, that is the relation (4.2) holds. This jump can be removed by subtracting the function

$$U_s(x,y)=\bar{A}\,\mathcal{I}m\left\{\ln z\right\}=\bar{A}\,\theta. \tag{4.5}$$

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 the 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 Eq.(4.1) are violated for the second derivative, that is

$$\phi_1''(0)+\phi_2''(0)=A_2\neq 0. \tag{4.6}$$

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

$$U_\sigma(x,y)=\mathcal{R}e\left\{c_2 z^2\ln z\right\},\qquad c_2=a_2+ib_2. \tag{4.7}$$

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

$$U_\sigma(x,y)=-\frac{A_2}{\pi}\left[2xy\ln\rho+\theta(x^2-y^2)\right]. \tag{4.8}$$

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

$$\text{at } y=0,\qquad \theta=0,\ U_\sigma=0,\ \frac{\partial^2 U_\sigma}{\partial x^2}=0$$

$$\text{at } x = 0, \qquad \theta = \frac{\pi}{2},\ U_\sigma = \frac{A_2 y^2}{2},\ \frac{\partial^2 U_\sigma}{\partial y^2} = A_2. \tag{4.9}$$

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

$$\text{at } \ x = y = 0, \qquad \frac{\partial^2 U_\sigma}{\partial x^2} + \frac{\partial^2 U_\sigma}{\partial y^2} = A_2 \tag{4.10}$$

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

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

$$U_\sigma(x, y) = \mathcal{R}e\left\{c_{2k} z^{2k} \ln z\right\}, \qquad c_{2k} = a_{2k} + i b_{2k}.$$

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 Eq.(5.1) with any (arbitrary) boundary conditions.

2. Compute on the boundaries the difference

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

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 Eq.(5.1) as

$$u = u_p + u_h \tag{5.3}$$

□

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

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

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

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 [12], p. 795) one can easily re-arrange the summation in (5.6) to get 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 described algorithm for computation of the particular solution is similar to the "local solver" procedure proposed in [11].

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

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

Thus, the fast computation of the particular solution in the form (5.6) requires the discretization on the non-uniform grid (5.7), while the homogeneous solution is computed on equally spaced (Fourier) nodes. In this case the implementation of the 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 finite trigonometric series along with a smoothing procedure near the boundaries (the Local Fourier Basis (LFB) method of [5], [6]). 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 2. 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 the 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 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 2 due to the equation.

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).

Once the particular solution is found, the computation is reduced to the solution of the Laplace equation with Dirichlet b.c. $\Phi = -u_p$ 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.5, right). The reconstructed solution $u = u_p + u_h$ to (5.9) is plotted in Fig.6.

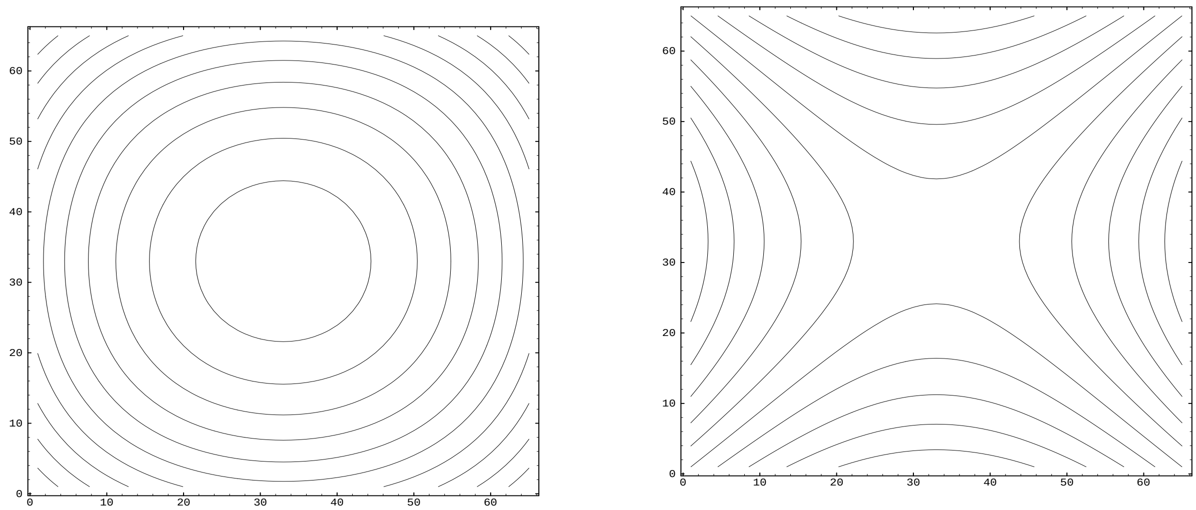


Figure 5. The particular solution of (5.9) computed by the LFB method (left).
. The smooth part of the homogeneous solution (right).

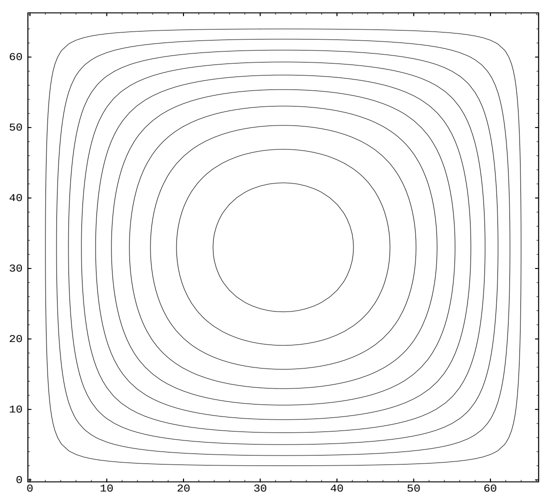


Figure 6. The complete solution of (5.9).

The maximum error in the numerical solution is shown in Table 2. The order of the subtraction procedure $r = 0, 1$ and 2. The last column gives the error in the solution obtained by the second order finite-difference method. The accuracy of the present method with the resolution 8×8 is even better than that of the finite-difference method on a grid 1024×1024.

Example 2 (A Gaussian distribution)

$N \times N$	$r=0$	$r=1$	$r=2$	f-d
8×8	2.4 (-5)	2.9 (-8)	5.0 (-10)	1.7 (-3)
16×16	5.7 (-6)	1.8 (-9)	1.3 (-12)	4.5 (-4)
32×32	1.4 (-6)	1.1 (-10)	1.4 (-14)	1.1 (-4)
64×64	3.6 (-7)	6.7 (-12)	5.1 (-16)	2.8 (-5)
1024×1024				2.7 (-7)

Table 2: The maximum errors in the computed solution using Algorithm IV ($r = 0, 1$ and 2) and the second order finite-difference scheme (the last column).

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

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

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 L. Greengard and J.-Y. Lee [11]. They developed a direct Poisson solver which implements an adaptive decomposition of the computational domain (using a quad-tree data structure) in order to get 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 attained by using the singular solutions of the Laplace equation (dipole and charge layers). The effect of singular 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

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

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 forth 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 [13].

The number of subdomains grows like $M = 3l + 1$, where l is the level of the quad-tree decomposition. Note that if the source is located at the point of intersection of four subdomains, as in the present example, then the number of subdomains increases much faster with the number of level 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 ALC method are related to $p = 42$, $K = 16$, and several numbers of refinement levels l. On each new level the resolution is improved by a factor of 2. 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 computation count is based on the expressions (5.8) and (5.12).

	present GF method			ALC method of [11]		
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

Table 3: The accuracy and the operation count (# of operations $\times 10^5$) in Example 2 of the present global Fourier (GF) method and the adaptive local Chebyshev (ALC) method of [12]

In the particular case of Eq.(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 getting 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. by shifting forth by one refinement level (this 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 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 especially 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 have 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 looses its advantage of adaptivity. Ultimately, when the sources spread out uniformly the number of sumdomains 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.7).

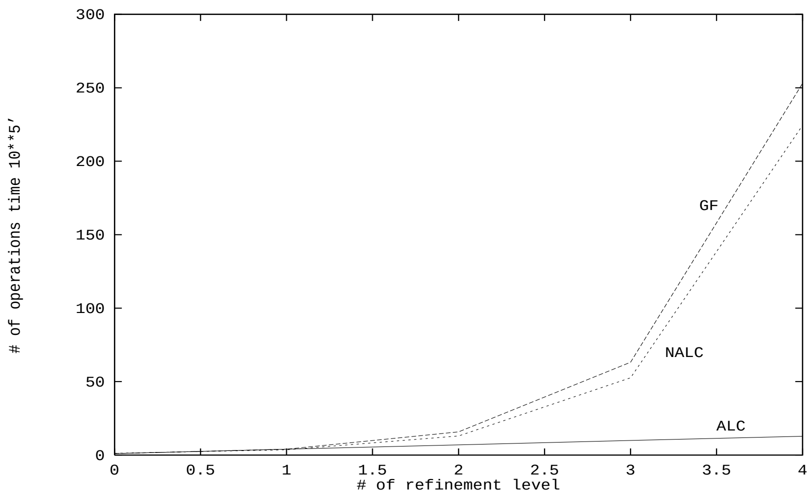


Figure 7. The operation count as a function of resolution for the adaptive local Chebyshev (ALC), nonadaptive local Chebyshev (NALC) and global Fourier (GF) methods.

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

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.8.

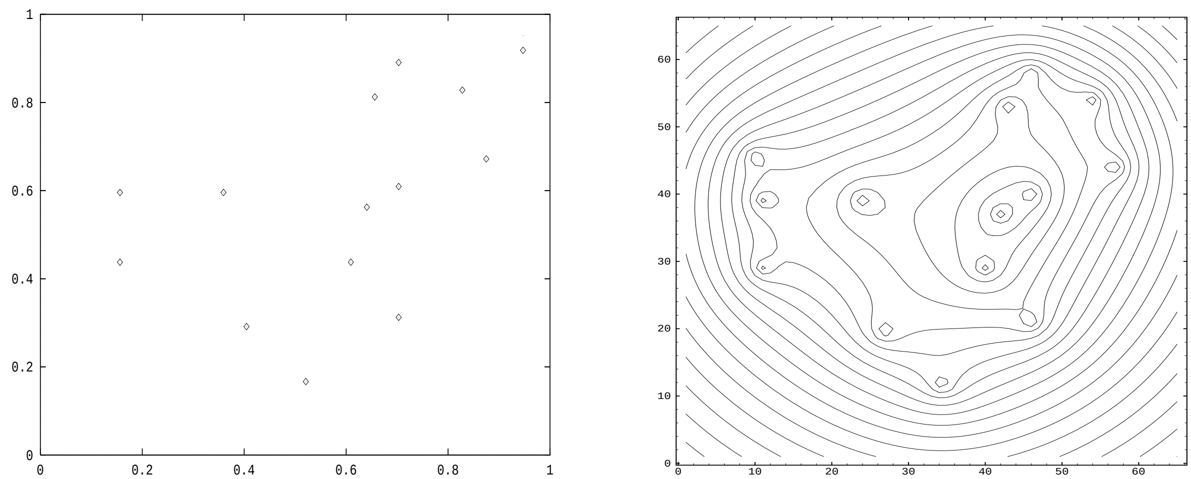


Figure 8. The random distribution of Gaussian sources (left) and a contour plot of the solution (right).

The performance of both GF and ALC methods is compared in Fig.8. The computational work is plotted against the discretization tolerance of the source function in Eq.(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 typically is 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.8 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 Eq.(5.13).

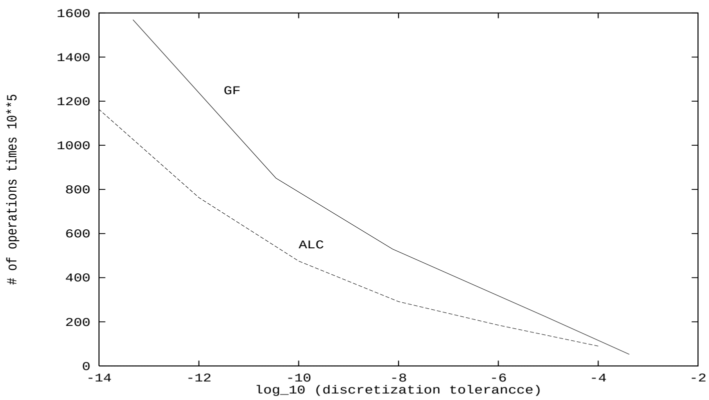


Figure 8. Performance of the GF and the ALC methods in Example 3.

6 Conclusions and Generalizations

The algorithm of this paper is a direct global Poisson solver in rectangular regions. It is based on a pseudo-spectral Fourier method along with a subtraction polynomial technique. The solution procedure is essentially analytic. The main computational tool here is the Fast Fourier Transform.

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 [11], 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 [14].

The generalization to three dimensions is straightforward. The subtraction functions has to be 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 3-D solution.

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

The present Poisson solver can be also employed for the solution of problems in irregular geometries in combination with the mapping [7] or the embedding [9] techniques.

7 Appendix 1

There several techniques which allow one to evaluate accurately derivatives of a not-periodic function $f(x)$, $x \in [a, b]$ discretized on $N+1$ equispaced discrete set of points $x_j = j\,\Delta$; $\Delta = (b-a)/N$, $0 \le j \le N$. One technique is based on the Fourier-Gegenbauer re-expansion method. The detailed description of this technique can be found in [3]. If the function is known on an extended interval $[a_1, b_1] \supset [a, b]$, then the derivatives at a point $x_0 \in [a, b]$ can be evaluated, within spectral accuracy, using the Local Fourier Basis method, see [5].

In this work we used expansion of the function into Taylor series near the boundaries to compute its several initial derivatives. This procedure is found to be accurate enough to preserve the high accuracy of the present method. It can be also regarded as the Newton interpolation formula.

Given the vector of nodal values $f_j = f(x_j)$, find the first p derivatives of the function f at the boundary point $x_0 = a$. The algorithm includes the following steps:

- Take $s \ge p$ grid points adjacent to the boundary $x_0 = a$.
- Define

$$y_k = f^{(k)}, \qquad A_{jk} = \frac{(j\Delta)^k}{k!}, \qquad b_k = f_k - f_0.$$

- Solve the linear system

$$A\,y = b$$

To compute accurately second and fourth derivatives, that are used in the present algorithm, the optimal size of the stencil is $s = 6 - 8$ points.

References

[1] R. W. Hockney, The potential calculation and some applications, *Methods In Computational Physics*, **9**, Academic Press, N-Y, 1970, p. 135-211.

[2] L.V. Kantorovich, V.I. Krylov, Approximate methods of higher analysis, Groningen, the Netherlands, 1958.

[3] L. Vozovoi, M. Israeli, A. Averbuch, Analysis and application of Fourier-Gegenbauer method to stiff differential equations, to appear in *SIAM, J. Numerical Analysis.*

[4] G. Skölermo - A Fourier method for numerical solution of Poisson's equation, *Math. Comput.* (USA), **29**, No 131, (1975), pp. 697-711.

[5] M. Israeli, L. Vozovoi, A. Averbuch, Spectral multi-domain technique with Local Fourier Basis, *J. Scientific Computing*, **8**, N 2, p. 135-149, (1993).

[6] L. Vozovoi, M. Israeli, A. Averbuch, Spectral multi-domain technique with Local Fourier Basis II: decomposition into cells. *J. Scientific. Computing*, **9**, No. 3, p. 311-326, (1994).

[7] L. Vozovoi, M. Israeli, A. Averbuch, Multidomain Local Fourier Method for PDEs in Complex Geometries, to appear in *J. Comput. Applied Math.*

[8] V. Rokhlin, Rapid solution of integral equations of classical potential theory, *J.Comput.Phys.*, **60**, No. 2, p.187-207, (1985).

[9] A. Mayo, The fast solution of Poisson's and biharmonic equations on irregular regions - *SIAM J.Numer.Anal.*, **21**, No. 2, p.285-299, (1984).

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

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

[12] M. Abramowitz, I.A. Stegun - Handbook of Mathematical Functions, Dover, 1970.

[13] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang - Spectral methods in Fluid dynamics, *Springer-Verlag*, (1989).

[14] L. Vozovoi, M. Israeli, A. Averbuch, High order fast elliptic solver in rectangular regions - *Technical report 846, Technion - Israel Institute of Technology*, 1995.