An Overview of Some Fast Algorithms with Applications: Some Open Problems and Challenges

Prabir Daripa*
Department of Mathematics
Texas A&M University
College Station, TX–77843
email: daripa@math.tamu.edu

Abstract

Some recently developed fast algorithms for some second order elliptic problems in complex two-dimensions are briefly reviewed. Scope of development of other algorithms based on similar ideas for other second and higher order elliptic problems in two- and three-dimensions is discussed. Various possible applications of these algorithms are discussed and some problems using these algorithms that have been solved are mentioned. Some open problems are mentioned.

1 Introduction

With the advent of modern computers over past few decades, there have been enormous strides in the development of algorithmic tools that can be used to explore the frontiers of science and to solve many real world problems of engineering interests. The journey in this direction has just begun and there can be little doubt that in the decades to follow, there will be further great advances in the development of robust successful algorithms. This is more so in light of the advances that perhaps will take place in next few decades in areas of speed, capacity, and logical design of electronic computers. Development of smarter computers at a current torrid pace gives further hope that significant breakthrough advances in the area of algorithms will likely happen in near future.

Algorithms, these days are central concepts to the modern view of what could constitute a "thinking device". To be more precise, an algorithm in itself is usually considered a systematic procedure to solve a problem and can be implemented on computers through some logical steps. Basic concept of algorithms is as old as the times of Euclid or possibly the beginning of mankind and there are numerous examples of classical algorithmic procedures such as Euclid's throughout mathematics. But it is perhaps remarkable that despite the historic origins of numerous specific examples of algorithms, the precise formulation of the concept of a general algorithm dates only from this century. In fact, the concept of algorithms

*Author for correspondence (E-mail: Prabir.Daripa@math.tamu.edu)

that are currently practiced all started around 1930s and successful implementation of these algorithms have allowed us to tackle problems that were unimaginable in recent past.

With the design and improvement of modern computers and development of new architectures, there is a demand on the flexibility of the algorithms so that these can be adapted on most computers with little modifications and be able to utilize the processors most efficiently. For example, a parallel computer, as opposed to a serial one, has a very great number of separate calculations carried out concurrently, and results of these largely autonomous operations are only intermittently combined together to give contributions to the overall calculation. The motivation for this type of computer architecture comes largely from an attempt to imitate the operation of the nervous system. Whereas there is no difference in principle between a parallel and a serial computers (both are in effect Turing Machines), there can be a significant difference in the efficiency, or speed, of the calculation as a whole.

Regardless of what kind of computers are used, usually one needs to have *insights* to construct a successful algorithm because an algorithm in itself does not guarantee that it will work and testing the validity of the algorithm is a nontrivial and almost always a non-algorithmic task. Therefore, worries about solving a problem efficiently do not stop with the construction of an algorithm. There is also another aspect which is equally important which I prefer to refer as fragility or robustness of the algorithm. Some algorithms are fragile in the sense that the slightest mutation of the algorithm can render it completely useless and such algorithms can rarely be improved upon. On the other hand, there are robust algorithms which can be improved upon, corrected or altered as often as one pleases. This is the kind of algorithms that are most often desirable in solving most problems of academic and industrial interest and this is what interests us most also. There are many other factors which qualifies the characteristics of an algorithm which should be paid attention to by all those who are engaged in either development or the use of such algorithms. Some of these factors are, for example, asymptotic complexity, number of operations involved per degree of freedom, accuracy, ease of implementation, fragility, easy of parallelizability, range of applicability, and so on.

These are among the issues that I should be attempting to address during my invited talk in the context of some recently developed application driven algorithms along with some applications. It is important to mention here that it is most often a tall order to ask for definitive answers to many of the questions discussed above and to design algorithms with all desirable features. Therefore, various algorithms with different features are usually designed towards the same goal and choice of a specific algorithm is often dictated by many competing factors.

2 Basic Philosophy and Some Results

It is well known that mathematical formulations of many applied problems require solving elliptic and parabolic problems. There are many ways to solve elliptic and parabolic problems in simple as well as complex geometry. Parabolic problems when discretized in time reduces to a sequence of elliptic problems, one at each time level. Therefore, we focus on algorithms, in particular the ones developed by Daripa and collaborators ([1], [2], [3], [5], [6], [7], [8], [9], [10], [11], [12]) for solving elliptic problems and some applications of these algorithms in this paper. It should be mentioned that there are many articles and surveys on numerical treatment of elliptic problems and we do not intend to survey these in this brief paper. Some references are however provided which by no means is exhaustive.

We have developed some analysis based fast algorithms for solving elliptic equations such as Poisson and Helmholtz equations within a unit disk (also exterior case and within an annulus). In these methods, it is assumed that the inhomogeneous terms in these equations are not known explicitly in functional form. These fast algorithms having similar complexity can be derived using classical "separation of variables" technique with some further analysis as well as using Green's function approach. These algorithms are very accurate, has $O(\ln N)$ complexity per degree of freedom, a small number of operations per degree of freedom and very easy to implement. These algorithms are also inherently parallelizable by construction.

These elliptic equations within a disk when Fourier transformed in circumferential direction reduces to ordinary differential equations (ODE) for the Fourier coefficients in the radial direction. Usually, these ordinary differential equations can be solved numerically. However, one-dimensional integral representations of solutions of these ODEs have some nice properties. The use of these properties along with the use of FFT's for computing the above mentioned Fourier transforms gives rise to one type of fast algorithms mentioned in the previous paragraph. An alternate approach to derive similar types of fast algorithms is based on representation of solutions of the elliptic equations in terms of free space Green's functions. The solution involves singular integrals whose direct evaluation by any of the well known quadrature techniques will be computationally slow and less accurate. An analysis of these integrals using the Fourier series of the inhomogeneous term (which appears in the kernel of the singular integral) and the properties of these convolution-type integrals leads to an algorithm which is of the same type as the one mentioned above. Needless to say, this approach requires computing the complementary part of the solution using any one of the many existing fast algorithms such as fast multipole method etc. It is worth mentioning here that if the inhomogeneous term in the elliptic equation is a sum of delta functions, then the fast multipole method can also used to solve the inhomogeneous elliptic equations efficiently, even though the implementation may not be as easy.

We have mentioned above the basic principles behind our algorithm in the context of two-dimensional elliptic problems in circular regions: be it interior, exterior or annular. The same principles can also be used for the development of similar fast algorithms for other second order elliptic as well as higher order elliptic equations in similar geometries. Three-dimensional analogue of these problems can also be developed within spherical geometry based on similar principles. Some of these topics are the subject of works currently in progress.

These algorithms have obvious limitations of circular or spherical geometry. Therefore questions regarding their range of applicability arise since most practical problems are in complex geometry. In this regard, we should mention that these algorithms can be used to solve problems in many complex geometries using domain embedding and/or domain overlapping methods. For example, elliptic problems in a geometry whose boundary is a union of the arcs of different circles can possibly be solved using the above algorithm and Schwarz's alternating method. Large scale elliptic problems in 2D and 3D complex shaped domains with inclusions and holes can also be solved using domain embedding method and our fast algorithm. A simple approach could be based on computation of particular solutions using a fast approach such as our algorithm in the larger but regular geometry and then to solve for the solution of the homogeneous problem within the given complex shaped boundary with appropriate boundary data using efficient solvers based on either boundary element or multipole or any other suitable method. The approach to be discussed below in brief uses a combination of either boundary control or optimal distributed control and our fast algorithm. Even though this approach is illposed, good approximate solutions can be constructed using rather coarse grids. In this approach, there is no extra need for solving the corresponding homogeneous equation and the specified boundary data is satisfied only approximately. Our approach has optimal order of complexity and easy to implement. There are other approaches also such as a combination of domain decomposition and fictitious domain ideas with the distributed Lagrange multipliers technique or boundary element approach. We do not discuss such approaches in this paper.

We have applied our fast algorithm so far to grid generation problem, quasiconformal mappings, design of subcritical airfoils, and modeling of blood flow through catheterized artery. All of these problems required solving elliptic problems within a circular or annular geometry after some appropriate mappings. We have also applied our fast algorithms in conjunction with either boundary control or optimal distributed control to solve 2D elliptic problems in complex shaped domains with holes. Currently further works are in progress some of which will be mentioned later in this section.

Theory, numerics and applications of these algorithms will be presented in detail in the conference. Due to space limitations, we mention below only one typical example very briefly and direct the readers to appropriate references which are either in print or in press (see the bibliography at the end). The approach below involves application of fast algorithms within a disk with domain embedding and boundary control to solve the problem.

The following example concerns the Dirichlet problem

$$\begin{aligned} &\Delta y - \sigma^2 y = f \text{ in } \omega \\ &y = g_\gamma \text{ on } \gamma, \end{aligned} \tag{2.1}$$

where $\omega \subset \mathbf{R}^2$ is bounded by the straight lines $x_1 = -\pi/2$, $x_1 = \pi/2$, $x_2 = -1.5$ and the curve $y = 0.5 + \cos(x + \pi/2)$. We approximate the solution of this problem by a solution of the Dirichlet problem

$$\begin{aligned} &\Delta y(v) - \sigma^2 y(v) = f \text{ in } \Omega \\ &y(v) = v \text{ on } \Gamma, \end{aligned} \tag{2.2}$$

in which the domain Ω is the disc centered at the origin with the radius of 2.3 (see Figure 1, (a)). We have taken $\sigma^2 = 0.75$ in numerical computations.

We approximate the functions f and v by the discrete Fourier transforms

$$\begin{aligned} f(r,\theta) &= \sum_{k=-n/2}^{n/2-1} f_k(r) e^{ik\theta}, \\ v(\theta) &= \sum_{k=-n/2}^{n/2-1} v_k e^{ik\theta}. \end{aligned} \tag{2.3}$$

Then the solution of problem (2.2),

$$y(v) = y_f + y'(v) \tag{2.4}$$

can also be written as a discrete Fourier transform

$$
\begin{aligned}
y_f(r,\theta) &= \sum_{k=-n/2}^{n/2-1} y_k(r)e^{ik\theta}, \\
y'(v)(r,\theta) &= \sum_{k=-n/2}^{n/2-1} y'_k(r)e^{ik\theta},
\end{aligned}
\tag{2.5}
$$

where the Fourier coefficients $y_k(r)$ and $y'_k(r)$ are given by

$$
\begin{aligned}
y_k(r) &= -\int_0^r \rho K_k(\sigma r)I_k(\sigma\rho)f_k(\rho)d\rho - \int_r^R \rho I_k(\sigma r)K_k(\sigma\rho)f_k(\rho)d\rho + \\
&\quad \frac{I_k(\sigma r)}{I_k(\sigma R)}\int_0^R \rho K_k(\sigma R)I_k(\sigma\rho)f_k(\rho)d\rho, \\
y'_k(r) &= \frac{I_k(\sigma r)}{I_k(\sigma R)}v_k.
\end{aligned}
\tag{2.6}
$$

Above, R is the radius of the disc, and I_k and K_k are the modified Bessel functions of first and second kind, respectively. A fast algorithm is proposed in [1], which using equations (2.6) and the fast Fourier transforms evaluates y_f and $y'(v)$ in (2.5) at the nodes of a mesh on the disc Ω with n equidistant nodes in the tangential direction and l equidistant nodes in the radial direction. It should be mentioned that the approach to the development of the fast algorithm in [1] is an alternative to the approach taken in [9], [6].

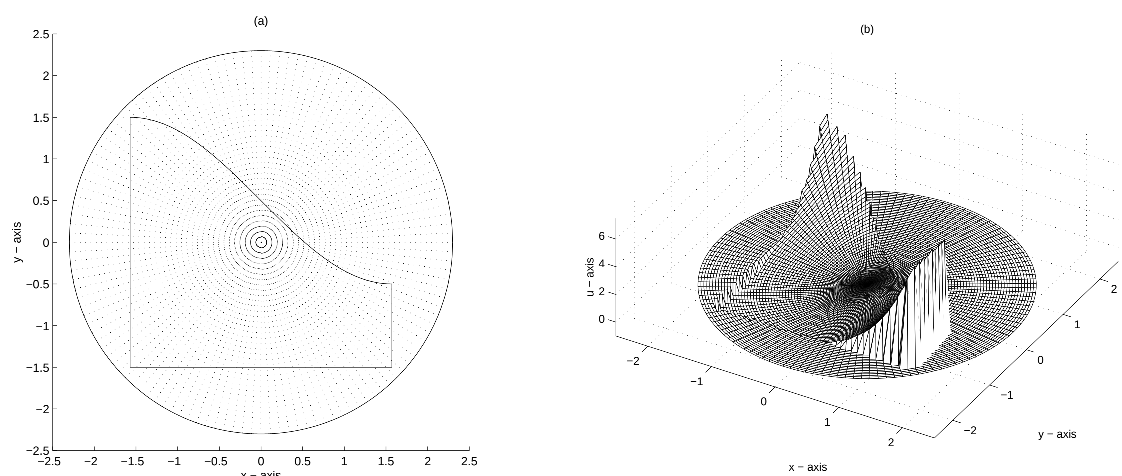

Figure 1. (a) - Domains, (b) - Exact solution.

It is worth noting from (2.3) that the finite dimensional space U of controls is the space of real periodic functions defined on $[0, 2\pi]$ which can be written as a Fourier transform with the terms $-n/2, \cdots, 0, \cdots, n/2-1$. Since the controls v are real functions, it follows from (2.3) that $v_i = \bar{v}_{-i}$ for $i = 1, \cdots, n/2-1$ and v_0 is real provided we choose $v_{-n/2} \in R$. Consequently, a basis of U is given by the functions: φ_0 which has the Fourier coefficient $v_0 = 1$ and the other ones being zero, $\varphi_{-n/2}$ which has the Fourier coefficient $v_{-n/2} = 1$, the other ones being zero, and φ_j, $-n/2+1 \le j \le n/2-1$, $j \ne 0$, have the Fourier coefficients $v_j = 1+i$, $v_{-j} = 1-i$ with the rest being zero. The boundary γ is discretized with m equidistant points and, the values of y_f and $y'(\varphi_)$ at the mesh points of the boundary γ are obtained by interpolation of function values at mesh points on Ω.

For numerical purposes, we have taken $f(x_1, x_2) = (2+x_1(1-\sigma^2))e^{x_1} + (2+x_2(1-\sigma^2))e^{x_2}$ and $g_\gamma(x_1, x_2) = x_1e^{x_1} + x_2e^{x_2}$ in (2.1). Then problem (2.1) has the exact solution $y(x_1, x_2) = x_1e^{x_1} + x_2e^{x_2}$ which is shown in Figure 1(b). In order to assess the effect of various extensions of the function f outside of ω on the numerical results, we have taken for this example only two types of extensions: (i) extending f using the above formula in ω; (ii) extending f by zero (see Figure 2). Figure 3 shows absolute errors at the mesh nodes in the domain ω when $n = 128$ and $\delta_r = 0.01$ for the two extensions of f .

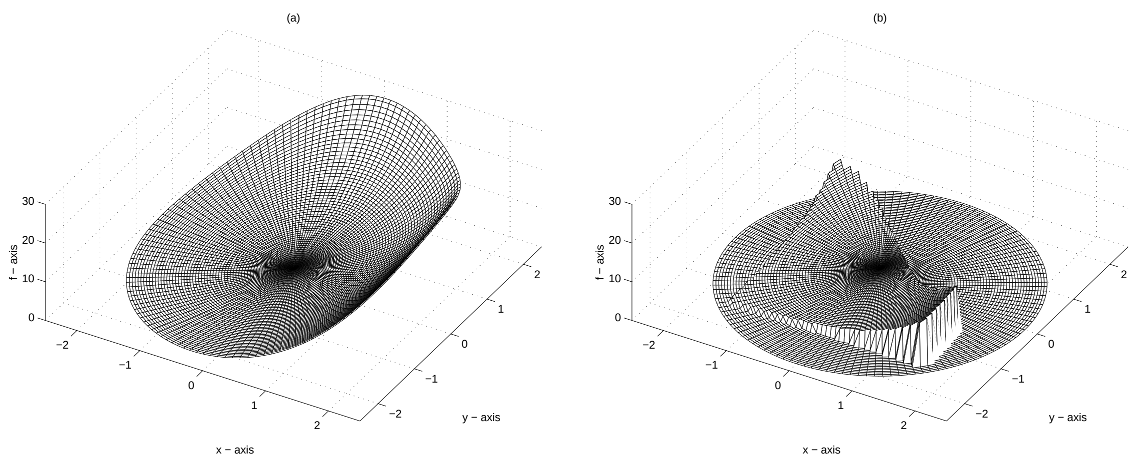

Figure 2. Extension of f by: (a) - the formula in the domain ω, (b) - zero.

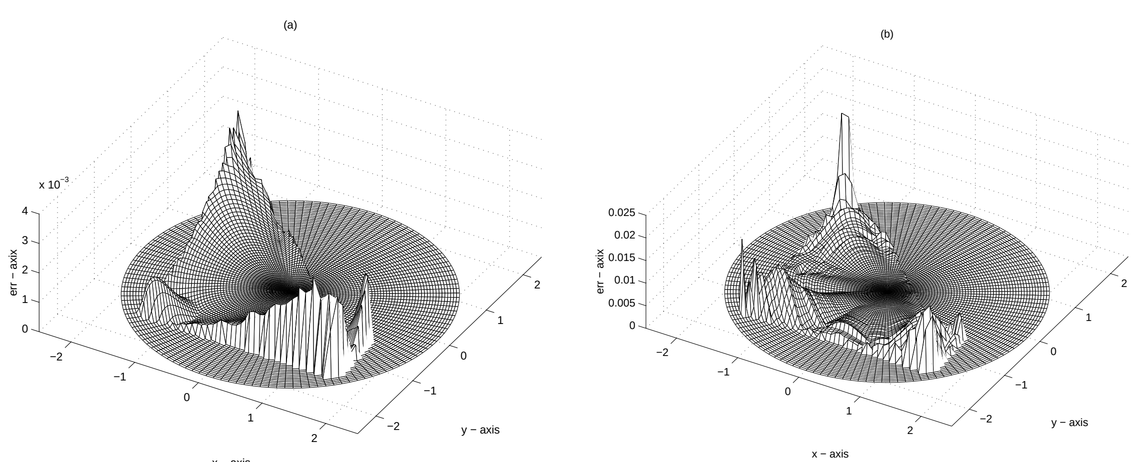

Figure 3. Errors in the domain when f is extended by:
(a) - the formula in the domain ω, (b) - zero

We have also applied this algorithm to the exterior problems with complex boundary (see [1]). We have also recently implemented optimal distributed control instead of boundary control in [2] to solving similar problems, and have also used in [4] embedding within a rectangular domains instead of circular domains.

We have applied similar fast algorithms developed originally in [8] and refined later in [10] for efficient and accurate evaluations of singular integrals to solving quasiconformal mappings of simply and doubly connected domains reported in [9] and [11]. We have also recently implemented a parallel version of this fast algorithm for singular integrals [5]. These integrals have wide applications including inverse design of airfoils which we have carried out in [7]. Recently we have applied these sequential and parallel versions of these algorithms for inverse design of airfoils. In [6] we report design of sequential and parallel fast algorithms for solving Poisson equation within a unit disk. Recently, we have modified this algorithm for an annular region and have applied to modelling blood flow in catheterized artery [12]. A condensed version of this article appears in this volume.

The applications scope of these algorithms to solving real world problems in complex shaped domains are abundant and we have barely begun to solve some of such problems. Any interested reader is welcome to solve some of the problems from the author's arsenal of many interesting problems. Also use of these algorithms in combination with domain decomposition/overlapping domains/fictitious domains and Schwarz alternating method or use of Lagrange multipliers or use of boundary/optimal distributed controls or boundary elements are wide open.

Acknowledgment: This Material is based in part upon work supported by the Texas Advanced Research Program under Grant No. TARP-97010366-030. Part of the results presented here has been obtained in collaboration with my colleague Lori Badea of Institute of Mathematics, Romanian Academy of Sciences.

References

[1] L. Badea and P. Daripa, *On a boundary control approach to domain embedding method*, SIAM J. Cont. Opt., To Appear.

[2] L. Badea and P. Daripa, *A domain embedding method using optimal distributed control and a fast algorithm*, J. Comp. Phys., submitted (2000).

[3] L. Badea and P. Daripa, *A fast algorithm for two-dimensional elliptic problems,* SIAM J. Num. Anal., submitted (2000).

[4] L. Badea and P. Daripa, *A spectral domain embedding method using distributed optimal control,* J. Comp. Phys., submitted (2000).

[5] L. Borges and P. Daripa, *A parallel version of a fast algorithm for singular integral transforms*, Numer. Algor., 23(1), pp. 71-96, 2000.

[6] L. Borges and P. Daripa, *A Fast Parallel Algorithm for the Poisson Equation on a Disk*, J. Comp. Phys., To appear.

[7] P. Daripa, *On applications of a complex variable method in compressible flows*, J. Comp. Phys., 88 (1990), pp. 337–361.

[8] ——, *A fast algorithm to solve nonhomogeneous Cauchy-Riemann equations in the complex plane*, SIAM J. Sci. Stat. Comput., 6 (1992), pp. 1418–1432.

[9] ——, *A fast algorithm to solve the Beltrami equation with applications to quasiconformal mappings*, J. Comp. Phys., 106 (1993), pp. 355–365.

[10] P. Daripa and D. Mashat, *Singular integral transforms and fast numerical algorithms*, Num. Algor., 18 (1998), pp. 133–157.

[11] P. Daripa and D. Mashat, *An efficient and novel numerical method for quasiconformal mappings of doubly connected domains*, Num. Algor., 18 (1998), pp. 159-178.

[12] P. Daripa and R. Dash, *Application of a fast algorithm to study oscillatory blood flow in an eccentric catheterized artery*, Jour. Engg. Math., submitted (2000).

[13] K. Hwang, *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, New York, NY, 1993.