PAMM · Proc. Appl. Math. Mech. **7**, 2020065–2020066 (2007) / **DOI** 10.1002/pamm.200700443

Fast algorithms for PDEs in simple and complex geometries

Prabir Daripa[1,*]

[1] Department of Mathematics, Texas A&M University, College Station, TX-77843, USA.

A brief review of our fast algorithms is given in this short paper.

An Overview

We review some fast and accurate numerical techniques for solving elliptic equations in complicated domains developed by the author and his colleagues in recent past. These are based on a combination of fast algorithms for regular domains and various domain embedding techniques. The fast algorithms for regular domains are derived from analysis of integral equation approach for solving elliptic equations in two- and three-dimensions. These algorithms are very accurate and easy to implement on serial as well as parallel computers. For an irregular domain, first the domain is embedded in a regular domain and then the problem is solved in the regular domain using either boundary or distributed control techniques and the fast algorithm for the regular domain. There are more additional considerations in these methods that make the method more efficient and accurate. We do not go into these details in this short paper.

One of the key contributing ideas of the author in this broad set-up is the way fast algorithms in regular domains for various elliptic equations are derived. This algorithm was originally conceived during the course of subsonic airfoil design using complex Beltrami equation formulation of subsonic compressible flow equations [1]. We show the basic idea of the algorithm through the following simple model problem in the real plane.

$$-\nabla^2 u = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad u = g(\mathbf{y}), \quad \mathbf{y} \in \Gamma = \partial\Omega, \tag{1}$$

where Ω is a unit disk. The solution of this equation can be written as $u(\mathbf{x}) = v(\mathbf{x}) + F(\mathbf{x}), \quad \mathbf{x} \in \Omega$, where

$$F(\mathbf{x}) = -\frac{1}{2\pi}\int_\Omega \log|\mathbf{x} - \zeta|\, f(\zeta) d\zeta, \qquad \mathbf{x} \in \Omega, \tag{2}$$

and $v(\mathbf{x})$ is the solution of the following problem:

$$\nabla^2 v(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega, \quad v(\mathbf{y}) = g(\mathbf{y}) - F(\mathbf{y}), \quad \mathbf{y} \in \Gamma. \tag{3}$$

As seen above, computation of $u(\mathbf{x})$ requires computing $v(\mathbf{x})$ and $F(\mathbf{x})$ for $\mathbf{x} \in \Omega$. The function $v(\mathbf{x}), \mathbf{x} \in \Omega$ can be computed very efficiently in complicated domains using boundary element method. How does one compute singular integrals $F(\mathbf{x})$ efficiently and accurately ? Straight-forward computation by quadrature-based method is expensive because of $O(N^2)$ per point computational complexity where N is the total number of nodes in the domain. Moreover, the method is not very accurate due to singular nature of the kernel. Our fast and accurate algorithm in circular and annular domains arises from carrying out the computation of the integral in the Fourier domain in combination with careful analysis since the kernel is singular [8]. With M points in the radial direction and N points in the circular direction, the algorithm (for evaluation of singular integral) that results from this analysis involves evaluating $F(\mathbf{x})$ at each of these $M \times N$ points using FFT from its radius dependent Fourier coefficients, $F_k(r_j),\ k \in [-N/2, N/2-1],\ j \in [1, M]$ which, in turn, are obtained from recursive relations in the radial direction involving Fourier coefficients of the source term in these elliptic equations. Computational complexity behind use of recursive relations is less than $O(MN \log N)$ computational complexity of using of FFT a total of $2M$ times. This operation count is much smaller than $O(M^2N^2)$ which will be required if these integrals were to be evaluated directly at all these MN points using quadrature-based methods. Thus our FFT-Recursive-Relation based algorithm has theoretical computational complexity of order $O(\log N)$ per point. An additional advantage of our algorithm is the small number of operations required per point (i.e., the constant hidden in $O(\log N)$). In practice, computational complexity of our algorithm per point is close to $O(1)$.

Our algorithm easily extends to other elliptic equations in real and complex domains. Since the free-space Green's function of the elliptic operator and hence the singular kernel in the expression of F change with change in the elliptic operator, analysis is required for every elliptic equation in order to develop corresponding fast algorithm. Algorithms based on these and equivalent ideas have been developed for Cauchy-Reimann equations, Beltrami, Poisson, and Helmholtz equations [10]. The algorithms for bi-harmonic and non-constant coefficient elliptic equations are still in the development stage and will be published in time. Our method also easily extends to three-dimension for spherical domains.

* Corresponding author E-mail: daripa@math.tamu.edu, Phone: 979-845-12-4, Fax: 979-845-6028

Parallel Computation:

By construction, our algorithm for evaluating singular integrals offers good parallelization opportunities and a lower computational complexity when compared with methods based on quadrature rules. In [7], we have developed a parallel version of the fast algorithm by redefining the inherently sequential recurrences present in the original sequential formulation. The parallel version only utilizes a linear neighbor-to-neighbor communication path, which makes the algorithm very suitable for any distributed memory architecture. Numerical results and theoretical estimates there show good parallel scalability of the algorithm. In [8], a parallel algorithm for solving the Poisson equation with either Dirichlet or Neumann conditions is presented. The algorithm is highly parallelizable and the implementation is virtually architecture-independent. Theoretical estimates there show good parallel scalability of the algorithm, and numerical results there show the accuracy of the method for problems with sharp variations on inhomogeneous term.

Applications:

These fast algorithms can be used to design subsonic airfoils [1], to generate quasi-conformal mappings and quasi-conformal grids ([3], [5], [6]). We have also applied our fast algorithm to modeling blood flow in catheterized artery [4]. These algorithms can obviously be applied to many other problems. The author is currently implementing this method to solve Navier-Stokes' and Stokes equations in regular and complicated domains.

Complicated Domains:

Our fast algorithms for regular domains have been used in conjunction with domain embedding to solve problems in complicated domains (see [9], [10], [11], and [12]). In adapting our algorithm for irregular domains, a complicated domain is embedded in a regular domain to which our fast algorithm is useful. Then the problem is solved in the regular domain using our fast algorithm and least squares method (distributed or boundary control).

Adaptivity:

We are also working towards development of an adaptive version of our FFT-Recursive-Relation based algorithm which will allow even faster and accurate computations for problems where the source terms such as the function $f(x)$ in equation (1) can behave nonuniformly in the domain. For example it may be highly localized only in certain areas of the domain only. This is not so uncommon in many practical problems. Due to shortage of space, we do not go any further than this in this 2-page paper.

References

[1] P. Daripa, *On Applications of a Complex Variable Method in Compressible Flows*, J. Comp. Phys., **88**(2), 337-361 (1990).

[2] P. Daripa, *A Fast Algorithm to Solve Non-Homogeneous Cauchy-Riemann Equations in the Complex Plane*, SIAM J. Sci. Stat. Comput., **13**(6), 1418–1432 (1992).

[3] P. Daripa, *A Fast Algorithm to Solve the Beltrami Equation with Applications to Quasiconformal Mappings*, J. Comp. Phys., **106**(2), 355–365 (1993).

[4] P. Daripa, and R. Dash, *A Numerical Study of Pulsatile Blood Flow in An Eccentric Catheterized Artery Using a Fast Algorithm*, J. Engg. Math., 42(1), 1-16 (2002).

[5] P. Daripa, and D. Mashat, *Singular Integral Transforms and Fast Numerical Algorithms: I*, Numer. Algor., **18**, 133-157 (1998).

[6] m P. Daripa, and D. Mashat, *An Efficient and Novel Numerical Method for Quasiconformal Mappings of Doubly Connected Domains*, Numer. Algor., **18**, 159-178 (1998).

[7] P. Daripa, and L. Borges, *A Parallel Version of A Fast Algorithm For Singular Integral Transforms*, Numer. Algor., **23**(1), 71-96 (2000).

[8] P. Daripa, and L. Borges, *A Fast Parallel Algorithm for the Poisson Equation on a Disk*, J. Comp. Phys., **169**, 51-192 (2001).

[9] P. Daripa, and L. Badea, *On a Boundary Control Approach to Domain Embedding Method*, SIAM J. Cont. Opt., **40**(2), 421-449 (2001).

[10] P. Daripa, and L. Badea, *A Fast Algorithm for Two-Dimensional Elliptical Problems*, Numer. Algor., **30**(3-4), 199-239 (2002).

[11] P. Daripa, and L. Badea, *On a Fourier Method of Embedding Domains Using an Optimal Distributed Control*, Numer. Algor., **32**(2-4), 261-273 (2003).

[12] P. Daripa, and L. Badea, *A Domain Embedding Method Using Optimal Distributed Control and A Fast Algorithm*, Numer. Algor., **36**(2), 95-112 (2004).