IMA Journal of Numerical Analysis (2016) **36**, 824–850
doi:10.1093/imanum/drv033
Advance Access publication on July 2, 2015

The FFTRR-based fast decomposition methods for solving complex biharmonic problems and incompressible flows

Aditi Ghosh and Prabir Daripa*
Department of Mathematics, Texas A&M University, College Station, TX 77843, USA
*Corresponding author: prabir.daripa@math.tamu.edu aditi.ghosh@math.tamu.edu

[Received on 21 March 2014; revised on 25 March 2015]

In this work, we present several computational results on the complex biharmonic problems. First, we derive fast Fourier transform recursive relation (FFTRR)-based fast algorithms for solving Dirichlet- and Neumann-type complex Poisson problems in the complex plane. These are based on the use of FFT, analysis-based RRs in Fourier space, and high-order quadrature methods. Our second result is the application of these fast Poisson algorithms to solving four types of inhomogeneous biharmonic problems in the complex plane using decomposition methods. Lastly, we apply these high-order accurate fast algorithms for the complex inhomogeneous biharmonic problems to solving Stokes flow problems at low and moderate Reynolds number. All these algorithms are inherently parallelizable, though only sequential implementations have been performed. These algorithms have theoretical complexity of the order $\mathcal{O}(\log N)$ per grid point, where N^2 is the total number of grid points in the discretization of the domain. These algorithms have many other desirable features, some of which are discussed in the paper. Numerical results have been presented which show performance of these algorithms.

Keywords: complex Biharmonic equation; complex Poisson equation; fast algorithms; stokes equations; incompressible flows; FFT; recursive relations; numerical implementation in Matlab.

1. Introduction

Biharmonic problems arise in many fields of classical physics and engineering, including the theory of elasticity, slow viscous flows, sedimentation and so on. They also arise as an intermediary step in solving some linear and nonlinear elliptic and parabolic problems such as problems involving the Navier–Stokes equations and so on; see Guazzelli & Morris (2012), Love (1927) and Muskhelishvili (1977). Therefore, there has been extensive research on the development of efficient numerical algorithms to solve such problems using various methods such as finite difference/finite element methods; see Bjorstad (1983), Braess & Peisker (1986), Cheng *et al.* (2000), Lai (2005), Monk (1987) and Peisker (1988), and integral equation methods; see Greenbaum *et al.* (1992), Greengard & Kropinski (1998), Greengard *et al.* (1996), Kropinski (1999) and Mayo (1992), to name a few of the methods.

To the best of our knowledge, however, there are no solvers to-date of the boundary value problems for the inhomogeneous biharmonic equation

$$(\partial_z \partial_{\bar{z}})^2 \omega = f(z, \bar{z}),$$

in the complex z-plane which is the physical $x - y$ plane through the assignment $z = x + iy$. We refer this equation as the complex biharmonic equation below. When the source term f is real and prescribed boundary values are real, we recover the conventional inhomogeneous biharmonic problem in the physical plane. Thus, the complex biharmonic equation is a more general case, and can be used to solve many problems in the plane including problems in plane elasticity and Stokes flow. Here the problems are in the physical plane, and can as well be solved with many other fast solvers for real biharmonic equation. However, many problems in plane elasticity (see Muskhelishvili, 1977; Wang *et al.*, 1991) are formulated as complex biharmonic problems whose solutions can be computed using the fast algorithms developed in this paper. Our effort in developing these algorithms is forward-looking, and we hope that they find applications in a variety of areas including Stokes's flow. It is conjectured here that this equation can play an important role in applied mathematics, in particular in fluid dynamics and elasticity theory, analogous to the role of Cauchy–Riemann equations in two-dimensional potential flow.

In this paper, fast algorithms for the complex inhomogeneous biharmonic problems are built from using the decomposition method in which each of these problems is decomposed into two 'complex' Poisson problems, and also one complex homogeneous biharmonic problem in some cases as we will see later. Our algorithms for the Poisson problems are derived from analysis of singular integral representations of their solutions based on classical Green's function approach. Such representations for the complex Poisson problems were made available only very recently, as recently as the year 2008, by Begehr (2007b). The use of quadrature methods to compute the complex singular integrals in these representations is computationally very expensive for large-scale problems and has poor accuracy. The asymptotic complexity of straightforward implementation of the quadrature method is usually $\mathcal{O}(N^2)$ per grid point, where N^2 is the total number of grid points in the discretization of the domain which is taken to be the unit disk. In order to reduce this complexity and to obtain high-order accurate solutions, we first develop fast Fourier transform recursive relation (FFTRR)-based fast complex Poisson solver. This makes use of the FFT and analysis-based RRs in the Fourier space resulting in a complexity of the order $\mathcal{O}(\log N)$ per grid point. This complexity is a significant improvement over $\mathcal{O}(N^2)$ per grid point. This FFTRR-based approach to solving 'complex problems' involving Cauchy–Riemann and Beltrami equations (Daripa, 1992, 1993) and real Poisson problems (Borges & Daripa, 2001) has been introduced earlier by one of the authors (P.D.). We summarize some features of these algorithms.

(1) The algorithms have very low computational complexity: $\mathcal{O}(\log N)$ per grid point.

(2) The constant hidden behind the order estimate of computational complexity of these algorithms is also very small as shown in this paper.

(3) The RRs allow one to define higher-order integration methods in the radial direction without the inclusion of additional grid points.

(4) The order of accuracy of the solution primarily depends on the numerical integration scheme of one-dimensional integrals involved in the RRs of these fast algorithms. It is second (third)-order when trapezoidal (Simpson's) integration formulae are used. The order of accuracy can be increased using Euler–Maclaurin formulas for integration as shown in this paper. The complexity estimate of these algorithms is independent of the order of accuracy.

(5) These algorithms are parallelizable by construction (see Borges & Daripa, 2000, 2001).

(6) A prior selection of the number of grid points based on desired accuracy.

(7) These algorithms are easily implemented.

These algorithms for the complex Poisson and the complex inhomogeneous biharmonic problems have been implemented using MATLAB. The Matlab code has been tested using some canonical examples, and most of the features of the algorithms mentioned above have been validated. Using our fast biharmonic solver, steady-state solutions of Navier–Stokes equations have been computed within a disk for a variety of boundary conditions at low and moderate Reynolds numbers. The accuracy of these solutions have been validated against known solutions of these problems in the literature. Numerical results are also presented, validating the error estimate and its use in a prior selection of mesh size based on a prescribed error. This has been done for the Poisson problems and the extension to biharmonic case is straightforward.

The paper is then laid out as follows. In Section 2, we define the Dirichlet and Neumann problems for Poisson equations in the complex plane and present integral representations of their solutions. Some theorems along with their proofs and RRs resulting from the analysis of these integral representations are given in this section. Similar development of the integral representations, RRs and theorems with regard to the four biharmonic problems are given in Section 3. These theorems and RRs are used to develop the fast algorithms for Poisson and biharmonic problems which are presented along with their computational complexities in Section 4. The MATLAB code encapsulating these algorithms has been used to perform some numerical calculations. Numerical results on the validation and accuracy check of these algorithms from computation of some canonical Poisson and biharmonic problems are given in Section 5. In Section 6, a crude but useful error estimate for the fast Poisson algorithms is given. Numerical results are presented validating the error estimate and its use in a prior selection of mesh size based on a prescribed error. In Section 7, these biharmonic algorithms have been applied to solving steady Navier–Stokes equations inside the unit disk at low and moderate Reynolds numbers. Numerical results are presented there. Finally, we conclude in Section 8.

2. Complex Poisson problems

In this section, we describe the mathematical formulation similar to the previous works of Daripa and collaborators (Daripa, 1992, 1993; Borges & Daripa, 2001) for the FFT and RR-based fast and high-order accurate algorithms to solve complex Poisson problems in the unit disc in the complex plane. This is necessary in order to build fast and high-order accurate algorithms for the complex inhomogeneous biharmonic problems which is discussed in Section 3. We should mention here that the existence of solutions to problems (P1) and (P2) defined below is given under the assumption that the inhomogeneous term $f \in L_1(\mathbf{D};\mathbb{C})$ (see Begehr, 2007a). However, we develop the algorithms for constructing solutions of these problems numerically under the assumption that f is expandable in a Fourier series. This is a stronger restriction than the original theorem, and hence justified.

2.1 *Dirichlet problem for the complex Poisson equation in the unit disk*

The Dirichlet problem for the complex Poisson equation in the unit disc $\mathbf{D} = \{z \in \mathbb{C}, |z| \leqslant 1\}$ in the complex plane is given by

$$\partial_z \partial_{\bar{z}} \omega = f(z,\bar{z}) \quad \text{in } \mathbf{D}, \qquad \omega = \gamma \quad \text{on } \partial\mathbf{D}. \tag{P1}$$

Note here that f in general is a function of both the variables z and $\bar{z}$. Below, for the sake of convenience, we write $f(z)$ even though it is a function of both the variables. This problem is uniquely solvable for $f \in L_1(\mathbf{D};\mathbb{C})$, $\gamma \in C(\partial D;\mathbb{C})$ (see Begehr, 2007a). If u is the solution of the homogeneous problem,

$$\partial_z \partial_{\bar{z}} u = 0 \quad \text{on } \mathbf{D}, \qquad u = \gamma - v \quad \text{on } \partial\mathbf{D}, \tag{2.1}$$

where v satisfies the equation

$$\partial_z \partial_{\bar{z}} v = f, \tag{2.2}$$

then the solution of the problem (P1) is given by

$$\omega = u + v. \tag{2.3}$$

A principal solution of (2.2) is obtained using standard Green's function method (see Greenberg, 1971; Borges & Daripa, 2001; Begehr, 2007b) and can be written as

$$v(z) = 4 \iint_{\mathbf{D}} G(z,\zeta) f(\zeta)\, \mathrm{d}\xi\, \mathrm{d}\eta, \tag{2.4}$$

where

$$G(z,\zeta) = \frac{1}{2\pi} \log|\zeta - z|, \tag{2.5}$$

is the free space Green's function for the Laplace equation.

Specifically, we expand $\omega(.)$ the solution of problem (P1), in terms of Fourier series, and derive the Fourier coefficients of the associated singular integrals in terms of RRs utilizing one-dimensional integrals in the radial direction. To demonstrate this, we evaluate the singular integral $v(z = r\,e^{\mathrm{i}\alpha})$; $0 < \alpha \leqslant 2\pi$ (see Borges & Daripa, 2001).

THEOREM 2.1 If $v(z)$ is the principal solution of the (P1) problem and $f(r\,e^{\mathrm{i}\alpha}) = \sum_{n=-\infty}^{\infty} f_n(r)\, e^{\mathrm{i}n\alpha}$ with $z = r\,e^{\mathrm{i}\alpha}$, then the Fourier coefficients $v_n(r)$ are given by

$$v_n(r) = \begin{cases} 4\displaystyle\int_0^r \rho \log r f_0(\rho)\,\mathrm{d}\rho + 4\int_r^1 \rho \log \rho f_0(\rho)\,\mathrm{d}\rho & \text{if } n = 0,\\ -2\displaystyle\int_0^r \left(\frac{\rho}{r}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho - 2\int_r^1 \left(\frac{r}{\rho}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho & \text{if } n > 0,\\ 2\displaystyle\int_0^r \left(\frac{r}{\rho}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho + 2\int_r^1 \left(\frac{\rho}{r}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho & \text{if } n < 0. \end{cases} \tag{2.6}$$

Proof. We introduce the following notations below:

$$B(\sigma, r) = \{z \in \mathbb{C} : |z - \sigma| < r\}, \quad \bar{B}(\sigma, r) = \{z \in \mathbb{C} : |z - \sigma| \leqslant r\}.$$

$$\Omega_0^r = B(0, r), \quad \Omega_{r-\epsilon}^{r+\epsilon} = B(0, r+\epsilon) - B(0, r-\epsilon), \quad \breve{\Omega}_{r-\epsilon}^{r+\epsilon} = \Omega_{r-\epsilon}^{r+\epsilon} - B(z, \epsilon).$$

Hence,

$$\begin{aligned} v(z) = \frac{2}{\pi} \iint_{\mathbf{D}} \log|\zeta - z| f(\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta &= \frac{2}{\pi} \lim_{\epsilon\to 0} \iint_{\Omega_0^{r-\epsilon}} \log|\zeta - z| f(\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta \\ &+ \frac{2}{\pi} \lim_{\epsilon\to 0} \iint_{\breve{\Omega}_{r-\epsilon}^{r+\epsilon}} \log|\zeta - z| f(\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta + \frac{2}{\pi} \lim_{\epsilon\to 0} \iint_{\Omega_{r+\epsilon}^1} \log|\zeta - z| f(\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta. \end{aligned} \tag{2.7}$$

Let

$$I^\epsilon = \frac{2}{\pi} \iint_{\breve{\Omega}_{r-\epsilon}^{r+\epsilon}} \log|\zeta - z| f(\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta.$$

Then

$$|I^{\epsilon}| \leqslant \frac{2}{\pi} \sup_{\zeta \in \breve{\Omega}_{r-\epsilon}^{r+\epsilon}} (|f(\zeta)\log|\zeta - z||)(\pi(r+\epsilon)^2 - \pi(r-\epsilon)^2) \leqslant 8 \sup_{\zeta \in \breve{\Omega}_{r-\epsilon}^{r+\epsilon}} |f(\zeta)|\epsilon|\log\epsilon|.$$

Therefore, the second integral in (2.7), namely $\lim_{\epsilon\to 0} I^{\epsilon} = 0$. This simplifies the representation of $v(z = r\,e^{\mathrm{i}\alpha})$ in (2.7) whose Fourier coefficients $v_n(r)$ are now given by

$$v_n(r) = \frac{1}{\pi^2} \lim_{\epsilon\to 0} \int_0^{2\pi} e^{-\mathrm{i}n\alpha} \left(\iint_{\Omega_0^{r-\epsilon}} \log|\zeta - z| f(\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta + \iint_{\Omega_{r+\epsilon}^{1}} \log|\zeta - z| f(\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta \right) \mathrm{d}\alpha. \quad (2.8)$$

Then it follows that

$$v_n(r) = \iint_{\Omega_0^r} f(\zeta) Q_n(r,\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta + \iint_{\Omega_r^1} f(\zeta) Q_n(r,\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta, \quad (2.9)$$

$$Q_n(r,\zeta) = \frac{1}{\pi^2} \int_0^{2\pi} e^{-\mathrm{i}n\alpha} \log|\zeta - z|\,\mathrm{d}\alpha. \quad (2.10)$$

For $z = r\,e^{\mathrm{i}\alpha}$, $\zeta = \rho\,e^{\mathrm{i}\theta}$, $r > \rho$, we have

$$Q_n(r,\zeta) = \begin{cases} -\dfrac{1}{\pi|n|}\left(\dfrac{\rho}{r}\right)^{|n|} e^{-\mathrm{i}n\theta} & \text{if } n \neq 0, \\ \dfrac{2}{\pi}\log r & \text{if } n = 0. \end{cases}$$

Similarly, for $r < \rho$ we have

$$Q_n(r,\zeta) = \begin{cases} -\dfrac{1}{\pi|n|}\left(\dfrac{r}{\rho}\right)^{|n|} e^{-\mathrm{i}n\theta} & \text{if } n \neq 0, \\ \dfrac{2}{\pi}\log \rho & \text{if } n = 0. \end{cases}$$

For $n > 0$, we have

$$\begin{aligned} \iint_{\mathbf{D}} f(\zeta) Q_n(r,\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta &= \iint_{\Omega_0^r} f(\zeta) Q_n(r,\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta + \iint_{\Omega_r^1} f(\zeta) Q_n(r,\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta \\ &= -2\int_0^r \left(\frac{\rho}{r}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho - 2\int_r^1 \left(\frac{r}{\rho}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho. \end{aligned} \quad (2.11)$$

Similar calculation for $n < 0$, $\quad n = 0$, respectively, yields

$$v_n(r) = 2\int_0^r \left(\frac{r}{\rho}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho + 2\int_r^1 \left(\frac{\rho}{r}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho, \quad (2.12)$$

$$v_0(r) = 4\int_0^r \rho\log r f_0(\rho)\,\mathrm{d}\rho + 4\int_r^1 \rho\log\rho f_0(\rho)\,\mathrm{d}\rho. \quad (2.13)$$

□

The solution u of the homogeneous problem (2.1) is obtained using the Poisson integral formula and with $z = re^{i\alpha}$, it is given by

$$u(r,\alpha) = \frac{1}{2\pi}\int_0^{2\pi} \phi(\theta)K(r,\alpha-\theta)\,\mathrm{d}\theta.$$

Here $K(\rho,\theta)$ is the Poisson kernel and $\phi(\theta) = \gamma(\theta) - v(\theta)$ on $\partial\mathbf{D}$. If ϕ_n are the Fourier coefficients of $\phi(\theta)$, then the Fourier coefficients $u_n(r,\alpha)$ of $u(z)$ are given by (see Borges & Daripa, 2001)

$$u_n(r) = r^{|n|}\phi_n. \tag{2.14}$$

Using (2.6) and (2.14), the solution ω of the Dirichlet problem (P1) is obtained from its Fourier coefficients $\omega_n = u_n + v_n$ (see (2.3)). As shown in the next section, development for the Neumann problem is similar and the computation of solution of the Neumann problem involves the same integrals as in (2.6).

2.2 *Neumann problem for the complex Poisson equation in the unit disk*

We consider here the complex Poisson equation with Neumann boundary condition in the unit disk $\mathbf{D}$ in the complex plane. The integral representation of the solution of this problem is given in Begehr (2007a). Our fast algorithm is derived from analysis of the integrals involved in this representation. The following theorem taken from Begehr (2007a) gives this representation.

THEOREM 2.2 The Neumann problem for the complex Poisson equation defined by

$$\partial_z\partial_{\bar z}w = f \quad \text{in } \mathbf{D}, \quad \partial_\nu w = g \quad \text{on } \partial\mathbf{D}, \tag{P2}$$

$f \in L_1(\mathbf{D};\mathbb{C}),\ g \in (\partial\mathbf{D};\mathbb{C}),\ (1/2\pi\mathrm{i})\int_{\partial\mathbf{D}} w(z)(\mathrm{d}z/z) = k \in \mathbb{C}$, is uniquely solvable iff

$$\frac{1}{4i}\int_{\partial\mathbf{D}} g(\zeta)\frac{\mathrm{d}\zeta}{\zeta} = \int_{\mathbf{D}} f(\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta. \tag{2.15}$$

The solution is given by

$$w(z) = k + \frac{1}{4\pi\mathrm{i}}\int_{\partial\mathbf{D}} G_3(z,\zeta)g(\zeta)\frac{\mathrm{d}\zeta}{\zeta} - \frac{1}{\pi}\iint_D G_3(z,\zeta)f(\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta, \tag{2.16}$$

where

$$G_3(z,\zeta) = \log|(1-z\bar\zeta)(\zeta - z)|^{-2} \tag{2.17}$$

is the Green's function for the Neumann problem and k is an arbitrary constant.

We rewrite

$$w(z) = k + u_1^N(z) + I_1^N(z) + v(z), \tag{2.18}$$

where

$$u_1^N(z) = -\frac{1}{\pi i}\int_{\partial D}\log|1 - z\bar\zeta|g(\zeta)\frac{\mathrm{d}\zeta}{\zeta}, \quad I_1^N(z) = \frac{2}{\pi}\iint_D \log|1 - z\bar\zeta|f(\zeta)\,\mathrm{d}\xi\,\mathrm{d}\eta, \tag{2.19}$$

and $v(z)$ is given by (2.4). Using the approach given in Daripa (1992), we analyse the integrals involved in the formula (2.16) to design the fast algorithm. Towards this end, we expand the solution $w(\cdot)$ in terms of Fourier series, and express its radius dependent Fourier coefficients in terms of Fourier coefficients of the boundary data g and of the inhomogeneous term f, the latter appearing as integrands of one-dimensional integrals. This is embodied in the following theorem proof of which can be easily obtained.

THEOREM 2.3 If $w(r,\alpha)$ is the solution of the Neumann problem in Theorem 2.2, $z = r\,e^{\mathrm{i}\alpha}$, $f(r\,e^{\mathrm{i}\alpha}) = \sum_{n=-\infty}^{\infty} f_n(r)\,e^{\mathrm{i}n\alpha}$, and $g(e^{\mathrm{i}\alpha}) = \sum_{n=-\infty}^{\infty} g_n e^{\mathrm{i}n\alpha}$, then the Fourier coefficients $w_n(r)$ can be written as

$$w_n(r) = k\delta_{n0} + u_{1,n}^N(r) + I_{1,n}^N(r) + v_n(r), \tag{2.20}$$

where

$$\delta_{n0} = \begin{cases} 0 & \text{if } n \neq 0, \\ 1 & \text{if } n = 0, \end{cases} \qquad u_{1,n}^N(r) = \begin{cases} g_n \dfrac{r^{|n|}}{|n|} & \text{if } n \neq 0, \\ 0 & \text{if } n = 0, \end{cases} \tag{2.21}$$

$$I_{1,n}^N(r) = \begin{cases} -2\displaystyle\int_0^1 f_n(\rho)\frac{(r\rho)^{|n|}}{|n|}\rho\,\mathrm{d}\rho & \text{if } n \neq 0, \\ 0 & \text{if } n = 0, \end{cases} \tag{2.22}$$

and expressions for $v_n(r)$ are given in Theorem 2.1.

Both the problems (Dirichlet and Neumann) thus require evaluations of the integrals (2.6). To improve upon the computational complexity for solving these problems, the integrals in (2.6) are evaluated using RRs which are derived next.

2.3 *Recursive relations*

We develop some RRs below. We discretize the unit disc into $M \times N$ grid points with M equidistant points in the radial direction and N equidistant points in the angular direction. We define the following integrals:

$$\left.\begin{aligned} &p_{1,n}(r) = -2\int_0^r \left(\frac{\rho}{r}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho, && p_{2,n}(r) = -2\int_r^1 \left(\frac{r}{\rho}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho && \text{if } n > 0, \\ &s_{1,n}(r) = 2\int_0^r \left(\frac{r}{\rho}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho, && s_{2,n}(r) = 2\int_r^1 \left(\frac{\rho}{r}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho && \text{if } n < 0, \\ &t_{1,0}(r) = 4\int_0^r \left(\frac{\rho}{r}\right) f_0(\rho)\,\mathrm{d}\rho, && t_{2,0}(r) = 4\int_r^1 \rho\log\rho f_0(\rho)\,\mathrm{d}\rho. && \end{aligned}\right\} \tag{2.23}$$

COROLLARY 2.4 It follows that $p_{1,n}(0) = p_{2,n}(1) = s_{1,n}(0) = s_{2,n}(1) = t_{1,0}(0) = t_{2,0}(1) = 0$.

COROLLARY 2.5 Let $0 = r_1 < r_2 < \cdots < r_M = 1$ and $r_j > r_i$, $1 \leqslant i,j \leqslant M$. Define

$$A_n^{i,j} = 2\int_{r_i}^{r_j} \left(\frac{R_A}{\rho}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho, \quad B_n^{i,j} = 2\int_{r_i}^{r_j} \left(\frac{\rho}{R_B}\right)^n \frac{\rho}{n} f_n(\rho)\,\mathrm{d}\rho \tag{2.24}$$

and

$$A_0^{i,j} = 4\int_{r_i}^{r_j} \left(\frac{\rho}{r_j}\right) f_0(\rho)\,\mathrm{d}\rho, \quad B_0^{i,j} = 4\int_{r_i}^{r_j} \rho\log\rho f_0(\rho)\,\mathrm{d}\rho, \tag{2.25}$$

where

$$R_A = \begin{cases} r_i & \text{if } n > 0, \\ r_j & \text{if } n < 0, \end{cases} \qquad R_B = \begin{cases} r_j & \text{if } n > 0, \\ r_i & \text{if } n < 0. \end{cases}$$

Then for $1 \leqslant i, j, l \leqslant M$, $r_l < r_i < r_j$ and using Corollary 2.4 we have the following RRs:

$$\left.\begin{aligned} p_{1,n}(r_i) &= \left(\frac{r_l}{r_i}\right)^n p_{1,n}(r_l) - B_n^{l,i}, \quad p_{2,n}(r_i) = \left(\frac{r_i}{r_j}\right)^n p_{2,n}(r_j) - A_n^{i,j} \quad \text{if } n > 0, \\ s_{1,n}(r_i) &= \left(\frac{r_i}{r_l}\right)^n s_{1,n}(r_l) + A_n^{l,i}, \quad s_{2,n}(r_i) = \left(\frac{r_j}{r_i}\right)^n s_{2,n}(r_j) + B_n^{l,j} \quad \text{if } n < 0, \\ t_{1,0}(r_i) &= \left(\frac{r_l}{r_i}\right) t_{1,0}(r_l) + A_0^{l,i}, \quad t_{2,0}(r_i) = t_{2,0}(r_j) + B_0^{i,j} \end{aligned}\right\} \tag{2.26}$$

and

$$v_n(r_l) = \begin{cases} (r_l \log r_l) t_{1,n}(r_l) + t_{2,n}(r_l) & \text{if } n = 0, \\ p_{1,n}(r_l) + p_{2,n}(r_l) & \text{if } n > 0, \\ s_{1,n}(r_l) + s_{2,n}(r_l) & \text{if } n < 0. \end{cases} \tag{2.27}$$

Proof. Simple algebraic manipulation of the relations in (2.23–2.25) proves the above result in Corollary 2.5. □

COROLLARY 2.6 For $r_j > r_i$, $1 \leqslant i, j \leqslant M$, and using the RRs we have

$$v_n(r_l) = \begin{cases} r_l \log r_l \left(\displaystyle\sum_{i=2}^{l} \left(\frac{r_i}{r_l}\right) A_n^{i-1,i} \right) + \displaystyle\sum_{i=l}^{M-1} B_n^{i,i+1} & \text{if } n = 0, \\ -\displaystyle\sum_{i=2}^{l} \left(\frac{r_i}{r_l}\right)^n B_n^{i-1,i} - \displaystyle\sum_{i=l}^{M-1} \left(\frac{r_l}{r_i}\right)^n A_n^{i,i+1} & \text{if } n > 0, \\ \displaystyle\sum_{i=2}^{l} \left(\frac{r_l}{r_i}\right)^n A_n^{i-1,i} + \displaystyle\sum_{i=l}^{M-1} \left(\frac{r_i}{r_l}\right)^n B_n^{i,i+1} & \text{if } n < 0. \end{cases} \tag{2.28}$$

Proof. Algebraic manipulation of the relations in (2.26) yields Corollary 2.6. □

3. Biharmonic problems

In this section, we describe the mathematical foundation for the fast algorithms to solve complex inhomogeneous biharmonic equations with four types of boundary conditions (see Begehr, 2007a). Each of these four problems is uniquely solvable under conditions that are mentioned in the sections below. This has been established by Begehr (2008). Solutions of two (see problems (D2) and (D4) in Sections 3.2 and 3.4) of these four problems are based on decomposing each of these two problems into two complex Poisson problems, whereas each of the remaining two problems (see problems (D1) and (D3) in Sections 3.1 and 3.3) require solutions of two complex Poisson problems and one complex homogeneous biharmonic problem. We should mention here that the existence of solutions to problems (D1), (D3) and (D4) defined below is given under the assumption that the inhomogeneous term $f \in L_1(\mathbf{D};\mathbb{C})$ (see Begehr, 2007a). However, we develop the algorithms for constructing solutions of these problems numerically under the assumption that f is expandable in a Fourier series. This is a stronger restriction than the original theorem, and hence justified.

3.1 *Dirichlet problem of type* (D1)

The first problem for the biharmonic equation, we consider is the following Dirichlet (D1) problem (to be called (D1) biharmonic problem henceforth).

$$\left.\begin{aligned} (\partial_z\partial_{\bar z})^2\omega &= f && \text{in } \mathbf{D},\\ \omega &= h_0 && \text{on } \partial\mathbf{D},\\ \omega_{\bar z} &= h_1 && \text{on } \partial\mathbf{D},\end{aligned}\right\}\tag{D1}$$

which is uniquely solvable (see Begehr (2008)) for $f \in L_1(\mathbf{D};\mathbf{C})$, $h_0 \in C^2(\partial\mathbf{D},\mathbb{C})$ and $h_1 \in C(\partial\mathbf{D},\mathbb{C})$. The solution to this problem is based on the decomposition of the problem into two complex Poisson problems $(3.1)_1$, $(3.1)_2$, and a complex homogeneous biharmonic problem (H1) which is a homogeneous version of the biharmonic problem (D1). We write $\omega = \omega_1 + \omega_2$, where ω_1 and ω_2 satisfy

$$\left.\begin{aligned} (\partial_z\partial_{\bar z})u &= f && \text{in } \mathbf{D},\\ u &= 0 && \text{on } \partial\mathbf{D},\end{aligned}\right\} \quad \left.\begin{aligned} (\partial_z\partial_{\bar z})\omega_1 &= u && \text{in } \mathbf{D},\\ \omega_1 &= h_0 && \text{on } \partial\mathbf{D}.\end{aligned}\right\}\tag{3.1}$$

$$\left.\begin{aligned} (\partial_z\partial_{\bar z})^2\omega_2 &= 0 && \text{in } \mathbf{D},\\ \omega_2 &= 0 && \text{on } \partial\mathbf{D},\\ \omega_{2\bar z} &= h_1 - \omega_{1\bar z} = h_1' && \text{on } \partial\mathbf{D}.\end{aligned}\right\}\tag{H1}$$

Thus, the method of solving the above biharmonic problem (D1) involves the following steps: (i) Solve the complex Poisson problem $(3.1)_1$ using the method of Section 2 to obtain $u(z)$ which is an input for the complex Poisson problem $(3.1)_2$; (ii) Then solve the complex Poisson problem $(3.1)_2$ to obtain $\omega_1(z)$ in $\mathbf{D}$ and $\omega_{1\bar z}$ on $\partial\mathbf{D}$ which is required in the boundary data of the (H1) problem defined above; (iii) Then finally solve the (H1) biharmonic problem to obtain ω_2 as described below. Finally, combining these two solutions ω_1 and ω_2, we obtain solution $\omega(z)$ of the (D1) biharmonic problems.

The solution to the (H1) problem is given by the following boundary integral (see Begehr, 2007a).

$$\omega_2(z) = \frac{(1-|z|^2)}{2\pi i}\int_{\partial\mathbf{D}} g_1(z,\zeta)h_1'(\zeta)\,\mathrm{d}\bar\zeta, \quad g_1(z,\zeta) = \frac{1}{1-z\bar\zeta} + \frac{1}{1-\bar z\zeta} - 1.$$

Using similar approach as before, this boundary integral is also analysed using Fourier series of the integrands that lead to design of fast algorithms. The result of this analysis is embodied in the following theorem. We omit the proof here because it can be easily obtained.

THEOREM 3.1 If $\omega_2(r,\alpha)$ is the solution of the (H1) problem for $z = r\,e^{i\alpha}$ in the unit disc and if $h_1'(e^{i\alpha}) = \sum_{n=-\infty}^{\infty} b_n\, e^{in\alpha}$, then the nth Fourier coefficient $\omega_{2,n}(r)$ of $\omega_2(r,\cdot)$ can be written as

$$\omega_{2,n}(r) = \begin{cases} -b_{1+n}r^{|n|}(1-r^2) & \text{if } n \neq 0,\\ -b_1(1-r^2) & \text{if } n = 0.\end{cases}\tag{3.2}$$

We discuss the design of the fast algorithm for the construction of solution of the complex biharmonic problem 3.1 in Section 4.3.

3.2 *Dirichlet problem of type* (D2)

We consider here the second Dirichlet (D2) biharmonic problem (to be called (D2) biharmonic problem henceforth). This problem is sometimes called Riquier and also Navier problem.

$$\left.\begin{array}{ll} (\partial_z\partial_{\bar z})^2\omega = f & \text{in } \mathbf{D}, \\ \omega = h_0 & \text{on } \partial\mathbf{D}, \\ \partial_z\partial_{\bar z}\omega = h_2 & \text{on } \partial\mathbf{D}. \end{array}\right\} \tag{D2}$$

This problem has boundary conditions different from the (D1) harmonic problem given above, and is uniquely solvable (see Begehr, 2008) for $f \in L_2(\mathbf{D};\mathbb{C})$, $h_0 \in C(\partial\mathbf{D};\mathbb{C})$ and $h_2 \in C(\partial\mathbf{D};\mathbb{C})$. We note here that this (D2) biharmonic problem can be decomposed into the following two complex Poisson problems.

$$\left.\begin{array}{ll} (\partial_z\partial_{\bar z})u = f & \text{in } \mathbf{D}, \\ u = h_2 & \text{on } \partial\mathbf{D}, \end{array}\right\} \quad \left.\begin{array}{ll} (\partial_z\partial_{\bar z})\omega = u & \text{in } \mathbf{D}, \\ \omega = h_0 & \text{on } \partial\mathbf{D}. \end{array}\right\} \tag{3.3}$$

These are Dirichlet (P1) type complex Poisson problems discussed in Section 2.1. The fast Poisson algorithm discussed in Section 4.1 based on the theory of Section 2.1 is used twice in succession, first for solving $(3.3)_1$ for u and then for solving $(3.3)_2$ for ω.

3.3 *Dirichlet–Neumann problem of type* (D3)

The third problem for the biharmonic equation we consider is the following Dirichlet–Neumann (D3) problem (to be called (D3) biharmonic problem henceforth).

$$\left.\begin{array}{ll} (\partial_z\partial_{\bar z})^2\omega = f & \text{in } \mathbf{D}, \\ \omega = h_0 & \text{on } \partial\mathbf{D}, \\ \partial_\nu\omega = h_1 & \text{on } \partial\mathbf{D}. \end{array}\right\} \tag{D3}$$

This problem is uniquely solvable for $f \in L_1(\mathbf{D};\mathbb{C})$, $h_0 \in C^2(\partial\mathbf{D};\mathbf{C})$, and $h_1 \in C^1(\partial\mathbf{D};\mathbf{C})$. The method for solving (D3) biharmonic problem is the same as for the problem (D1) (see Section 3.1), except that problem (H1) there is replaced by following (H2) problem.

$$\left.\begin{array}{ll} (\partial_z\partial_{\bar z})^2\omega_2 = 0 & \text{in } \mathbf{D}, \\ \omega_2 = 0 & \text{on } \partial\mathbf{D}, \\ \partial_\nu\omega_2 = h_1 - \partial_\nu\omega_1 = h_1' & \text{on } \partial\mathbf{D}. \end{array}\right\} \tag{H2}$$

The solution to the (H2) problem is given by the following boundary integral (see Begehr, 2007a).

$$\omega_2(z) = -\frac{(1-|z|^2)}{4\pi \mathrm{i}}\int_{\partial\mathbf{D}} g_1(z,\zeta)h_1'(\zeta)\frac{\mathrm{d}\zeta}{\zeta}.$$

THEOREM 3.2 If $\omega_2(r,\alpha)$ is the solution of the (H2) problem in $\mathbf{D}$, and $h_1'(e^{i\alpha}) = \sum_{n=-\infty}^{\infty} b_n\, e^{\mathrm{i}n\alpha}$, then the Fourier coefficients $\omega_{2,n}(r)$ of $\omega_2(r,\cdot)$ can be written as

$$\omega_{2,n}(r) = \begin{cases} -\dfrac{(1-r^2)}{2} r^{|n|} b_n & \text{if } n \neq 0, \\ -\dfrac{(1-r^2)}{2} b_0 & \text{if } n = 0. \end{cases} \tag{3.4}$$

Proof. The proof is similar to the proof of Theorem 3.1. □

3.4 *Dirichlet–Neumann problem of type* (D4)

The fourth problem for the biharmonic equation we consider is the following Dirichlet–Neumann (D4) problem (to be called (D4) biharmonic problem henceforth).

$$\left.\begin{array}{ll} (\partial_z\partial_{\bar z})^2 w = f & \text{in } \mathbf{D}, \\ w = h_0 & \text{on } \partial\mathbf{D}, \\ \partial_\nu w_{z\bar z} = h_2 & \text{on } \partial\mathbf{D}. \end{array}\right\} \tag{D4}$$

This problem is also uniquely solvable for $f \in L_1(\mathbf{D},\mathbb{C})$, $h_0,\ h_2 \in (\partial\mathbf{D};\mathbb{C})$ iff

$$\frac{1}{4\mathrm{i}} \int_{\partial\mathbf{D}} h_2(\zeta) \frac{\mathrm{d}\zeta}{\zeta} = \int_{\mathbf{D}} f(\zeta)\, \mathrm{d}\xi\, \mathrm{d}\eta.$$

This problem can similarly be decomposed into two Poison problems as follows:

$$\left.\begin{array}{ll} (\partial_z\partial_{\bar z})u = f, & \text{in } \mathbf{D}, \\ \partial_\nu u = h_2, & \text{on } \partial\mathbf{D}, \end{array}\right\} \quad \left.\begin{array}{ll} (\partial_z\partial_{\bar z})w = u, & \text{in } \mathbf{D}, \\ \omega = h_0, & \text{on } \partial\mathbf{D}. \end{array}\right\} \tag{3.5}$$

As before, the fast Poisson algorithm discussed in Section 4.1 is used twice in succession, first for solving the complex Neumann Poisson problem $(3.5)_1$ for u and then for solving the complex Dirichlet Poisson problem $(3.5)_2$ for ω.

4. Fast algorithms

We first build FFT and RRs-based fast algorithms for the complex Poisson problems in Section 4.1, followed by the description of the fast algorithms for the biharmonic problems in Section 4.3. It is worth recalling discretization of the unit disk $\mathbf{D}$ using $M \times N$ grid points, M in the radial direction and N in the azimuthal direction.

4.1 *Fast algorithm for the complex Poisson problems*

We first consider the (P1) problem.
Initialization: Choose M and $N = 2^n$, where n is an integer. Define $K = N/2$.
Inputs: $M, N, \gamma(e^{2\pi \mathrm{i}k/N}), f(r_l\, e^{2\pi \mathrm{i}k/N})$, $l \in [1, M]$, $k = 1 \cdots N$ and $n \in [-K+1, K]$.
Step 1. Compute the Fourier coefficients γ_n, and $f_n(r_l)$, $n \in [-K+1, K]$, $l \in [1, M]$.

Step 2. Compute $A_n^{i,i+1}$, $A_0^{i,i+1}$, $B_n^{i,i+1}$, $B_0^{i,i+1}$ for $i \in [1, M-1]$, $n \in [-K+1, K]$ using (2.24)$_1$, (2.24)$_2$, (2.25)$_1$, and (2.25)$_2$.

Step 3. Using (2.26) in Corollary 2.5, compute the RRs as shown in the flowchart.

```
Set $p_{1,n}(r_1 = 0) = 0$,
for $n = 1, \dots, K$
    for $l = 2, \dots, M$
        $p_{1,n}(r_l) = (\frac{r_{l-1}}{r_l})^n p_{1,n}(r_{l-1}) - B_n^{l-1,l}$
    end
end
Set $p_{2,n}(r_M = 1) = 0$
for $n = 1, \dots, K$
    for $l = (M-1), \dots, 1$
        $p_{2,n}(r_l) = (\frac{r_l}{r_{l+1}})^n p_{2,n}(r_{l+1}) - A_n^{l,l+1}$
    end
end
Set $s_{1,n}(r_1 = 0) = 0$
for $n = -K, \dots, -1$
    for $l = 2....M$
        $s_{1,n}(r_l) = (\frac{r_l}{r_{l-1}})^n s_{1,n}(r_{l-1}) + A_n^{l-1,l}$
    end
end
Set $s_{2,n}(r_M = 1) = 0$
for $n = -K, \dots, -1$
    for $l = (M-1), \dots, 1$
        $s_{2,n}(r_l) = (\frac{r_{l+1}}{r_l})^n s_{2,n}(r_{l+1}) + B_n^{l,l+1}$
    end
end
Set $t_{1,0}(r_1 = 0) = 0$,
for $l = 2, \dots, M$
    $t_{1,0}(r_l) = (\frac{r_{l-1}}{r_l}) t_{1,0}(r_{l-1}) + A_0^{l-1,l}$
end
Set $t_{2,0}(r_M = 1) = 0$
for $l = (M-1), \dots, 1$
    $t_{2,0}(r_l) = t_{2,0}(r_{l+1}) + B_0^{l,l+1}$
end
```

Step 4. Compute $v_n(r_l)$ for $n \in [-K+1, K]$, $l \in [1, M]$ using (2.27).

Step 5. Compute $u_n(r_l)$, $n \in [-K+1, K]$, $l \in [1, M]$ using (2.14).

Step 6. Finally, compute $\omega(r_l\, e^{2\pi \mathrm{i} k/N})$ by inverting $(v_n(r_l) + u_n(r_l))$ using FFT.

The sequential algorithm for the (P2) problem is similar.

4.2 *Algorithmic complexity*

The computational complexity of the above fast algorithm for the complex Poisson equation in the unit disc is presented below.

Step	Operation count
1	Computation of γ_n and f_n by $(M+1)$ times application of FFT, each time to N data points contributes $\mathcal{O}(MN\log N)$
2	Computation of $A^{i,i+1}_{n,1}$ and $B^{i,i+1}_{n,1}$, $i \in [1, M-1]$ contributes $\mathcal{O}(MN)$
3	Computation of $p^{(1)}_{1,n}(r_l)$, $p^{(1)}_{2,n}(r_l)$, $s^{(1)}_{1,n}(r_l)$, $s^{(1)}_{2,n}(r_l)$, $t^{(1)}_{1,0}(r_l)$, $t^{(1)}_{2,0}(r_l)$, $l \in [1, M]$ contributes $\mathcal{O}(MN)$
4	Computation $v_n(r_l)$, $l \in [1, M]$ contributes $\mathcal{O}(MN)$
5	Computation $u_n(r_l)$, $l \in [1, M]$ contributes $\mathcal{O}(MN)$
6	Computation of $\omega(r_l\, e^{\frac{2\pi ik}{N}})$, $k \in [1, N]$ using FFT contributes $\mathcal{O}(MN\log N)$

We see from above that the algorithm has complexity of the order $\mathcal{O}(MN\log N)$ for a total of $M \times N$ degrees of freedom (i.e., number of grid points). Thus, it has computational complexity of the order $\mathcal{O}(\log N)$ per degree of freedom. For the Neumann problem also, it is easy to see that the complexity remains the same.

4.3 *Fast algorithms for the complex inhomogeneous biharmonic problems*

For (D2) and (D4) biharmonic problems, we apply the above fast Poisson algorithm for the (P1) problem twice in succession. Since the algorithms for biharmonic problems (D1) and (D3) are similar, below we outline the steps of the fast algorithm for solving (D1) biharmonic problem and skip the details for (D3) biharmonic problem.

Initialization: Choose M and $N = 2^n$ where n is an integer. Define $K = N/2$.

Inputs: $M, N, h_0(e^{2\pi ik/N}), f(r_l\, e^{2\pi ik/N}), l \in [1, M], k \in [1, N]$ and $n \in [-K+1, K]$.

Step 1. Compute the Fourier coefficients $h_{0,n}(r_l), f_n(r_l), n \in [-K+1, K], l \in [1, M]$.

Step 2. Compute $u_n(r_l),\ n \in [-K+1, K], l \in [1, M]$ using the Poisson algorithm.

Step 3. Compute the Fourier coefficients $u_n(r_l),\ n \in [-K+1, K], l \in [1, M]$ using FFT.

Step 4. Compute $\omega_{1,n}(r_l), n \in [-K+1, K], l \in [1, M]$ using the Poisson algorithm.

Step 5. Compute the $\omega_{2,n}(r_l), n \in [-K+1, K], l \in [1, M]$ using Theorem 3.1.

Step 6. Finally, compute $\omega(r_l\, e^{\frac{2\pi ik}{N}})$ by inverting $(\omega_{1,n}(r_l) + \omega_{2,n}(r_l))$ using FFT.

It is easy to see that this algorithm has the same computational complexity, since it uses our fast complex Poisson solver twice in succession.

5. Numerical results

In this section, numerical results obtained using above fast Poisson and biharmonic algorithms on several problems are presented and discussed. Numerical implementation of these fast algorithms were done in MATLAB, and computations were performed using double precision (16 digit) arithmetics. We have implemented the trapezoidal, Simpson's and Euler–Maclaurin formulas (Sidi & Israeli, 1988) for numerical integration, and determined the accuracy of the fast solvers. Euler–Maclaurin formulas are used with two points-based numerical integration of one-dimensional integrals that appear in the fast algorithms discussed in Sections 4.1 and 4.3. Computations have been performed for several values of M and N. In all the examples to be presented below, the source terms f and all boundary data associated with the Poisson and biharmonic problems have been taken to be C^∞ functions, so that the solutions of these problems are also C^∞. Since the fast algorithms use Fourier expansions of the source term and of the solutions of these problems in the azimuthal direction with grid size $2\pi/N$, the solvers have exponential rate of convergence in the grid size $2\pi/N$. It has been found more than sufficient to use $N = 32$ as the solutions of these problems using our algorithms have been found to remain the same within machine round-off error for $N > 32$ with M fixed. Therefore, the order of accuracy of the solutions has been determined for several increasing values of M by evaluating relative errors in L_∞ norm. The error and order of accuracy for several values of M using different numerical integration schemes will be shown below. We first show the performance of our algorithms on complex Poisson problems followed by complex biharmonic problems.

5.1 *The complex Poisson problem*

We consider Dirichlet (P1) and Neumann (P2) complex Poisson problems (see Section 2) with appropriate inhomogeneous terms f and boundary conditions, so that the solutions of both the problems are given by

$$\omega(z) = z^{3/2}\bar{z}^{5/2} + \mathrm{i}z^{5/2}\bar{z}^{3/2}, \tag{5.1}$$

which is refereed to as Example 1 in the rest of this paper. Example 2 which has been taken from Borges & Daripa (2001), the solution is given by

$$\omega(r,\theta) = 3\,e^{r(\cos\theta+\sin\theta)}(r\cos\theta - (r\cos\theta)^2)(r\sin\theta - (r\sin\theta)^2) + 5. \tag{5.2}$$

For Example 1, Table 1 shows relative error and order of accuracy in azimuthal grid size $(1/(M-1))$ for several increasing values of M with the value of N fixed at 64. Similar results are shown in Table 2 for Example 2. These tables show the fourth-order accuracy in the azimuthal grid size of the algorithms. This is a marked improvement over second-order accuracy of the fast Poisson solver of Borges & Daripa (2001), where trapezoidal and Simpson's integration formula were used. Therefore, in these tables we do not show the results with trapezoidal and Simpson's rules.

5.2 *The complex biharmonic problem*

Now we consider complex inhomogeneous biharmonic problems with different boundary conditions and apply the Algorithm 4.3 to solve them. We should recall from Section 3 that four problems (D1), (D2), (D3) and (D4) discussed there can be grouped into two classes from the view point of their solution procedure: Group 1 consisting of problems (D1) and (D3), and Group 2 consisting of problems (D2)

TABLE 1 *Relative errors and orders of accuracy of the Poisson solvers for Example* 1 (*see solution* (5.1)) *using Euler–Maclaurin formula and* $N = 64$

	Dirichlet Poisson problem (P1)		Neumann Poisson problem (P2)	
M	$\|\cdot\|_\infty$	Order of accuracy	$\|\cdot\|_\infty$	Order of accuracy
16	3.7×10^{-5}	—	3.49×10^{-5}	—
32	2.03×10^{-6}	4.18	1.26×10^{-6}	4.79
64	1.19×10^{-7}	4.09	6.78×10^{-8}	4.21
128	7.21×10^{-9}	4.05	4.34×10^{-9}	3.97
256	4.43×10^{-10}	4.02	2.8×10^{-10}	3.96
512	2.75×10^{-11}	4.01	1.78×10^{-11}	3.97

TABLE 2 *Relative errors and orders of accuracy of the algorithms for Example* 2 (*see solution* (5.2)) *using Euler–Maclaurin formula and* $N = 64$

Dirichlet Poisson problem (P1)		
M	$\|\cdot\|_\infty$	Order of accuracy
16	3.63×10^{-4}	—
32	5.26×10^{-6}	4.95
64	2.39×10^{-7}	4.46
128	5.03×10^{-9}	4.78
256	2.99×10^{-10}	4.07
512	1.82×10^{-11}	4.03

TABLE 3 *Relative error for* $f(z) = 12(\bar{z} + \mathrm{i}z)$ *using Euler–Maclaurin formula for the biharmonic problem* (D1) *with* $N = 64$

	Trapezoidal		Euler–Maclaurin	
M	$\|\cdot\|_\infty$	Order	$\|\cdot\|_\infty$	Order
16	3.4×10^{-2}	—	1.2×10^{-3}	—
32	6.7×10^{-3}	2.3	6.9×10^{-5}	4.1
64	3.2×10^{-3}	1.1	4.8×10^{-6}	3.9
128	9.2×10^{-4}	1.8	3.4×10^{-7}	3.8
256	2.2×10^{-4}	2.1	2.1×10^{-8}	4.0
512	5.8×10^{-5}	1.9	1.7×10^{-9}	3.6

and (D4). Each of the group 1 problems require solution of a homogeneous biharmonic problem in their solution procedure by decomposition method, whereas each of the Group 2 problems do not require this (see Section 3). Therefore, we show results for only one problem from each of these two classes.

First, we consider the biharmonic problems (D1) and (D2) with $f(z) = 12(\bar{z} + \mathrm{i}z)$, and their respective boundary conditions $h_0 = z + \mathrm{i}\bar{z}, h_1 = 3 + 2\mathrm{i}\bar{z}^2$ (for (D1) problem) and $h_0 = \bar{z} + \mathrm{i}z, h_2 = 6(\bar{z} + \mathrm{i}z)$ (for (D2) problem). The trapezoidal and Euler–Maclaurin formulas have been used for numerical integration in the radial direction for several values of M with $N = 64$. A second-order convergence using trapezoidal rule and a fourth-order convergence using Euler–Maclaurin rule are observed in both of these tables (Table 3 and 4).

Computations have been performed with many more source and boundary terms on all four problem types, and similar results on the error and order of accuracy have been observed. We just show one more set of results obtained with a different set of source (f) and boundary data, but with biharmonic problems (D3) and (D4). The source term taken for both the problems is given by $f(z, \bar{z}) = 72(z^2\bar{z} + z\bar{z}^2)$, and their respective boundary conditions are given by $h_0 = z + i\bar{z}$, $h_1 = 7z + 7\bar{z}$ for (D3) and $h_0 = \bar{z} + z$, $h_2 = 60\bar{z} + 60z$ for (D4). The source term used here is symmetric: $f(z, \bar{z}) = f(\bar{z}, z)$, as opposed to the one used before where it was not symmetric. Table 5 shows the relative errors and order of accuracy obtained with Euler–Maclaurin formulas (trapezoidal results are similar as before) for both of these problems which support that these biharmonic algorithms give fourth-order accuracy results.

Next we compare numerical asymptotic complexity with our theoretical complexity $O(N^2 \log N)$, where N^2 is the total number of grid points. Numerical complexity has been calculated based on CPU

TABLE 4 *Relative error with $f(z) = 12(\bar{z} + \mathrm{i}z)$ using Euler–Maclaurin formula for the biharmonic problem* (D2) (*see* (D2)) *with* $N = 64$

	Trapezoidal		Euler–Maclaurin	
M	$\\|\cdot\\|_\infty$	Order	$\\|\cdot\\|_\infty$	Order
16	2.7×10^{-2}	—	3.7×10^{-4}	—
32	7.1×10^{-3}	1.9	1.7×10^{-5}	4.39
64	1.8×10^{-3}	1.9	1.1×10^{-6}	3.95
128	4.5×10^{-4}	2.0	6.6×10^{-8}	4.05
256	1.14×10^{-4}	1.9	4.2×10^{-9}	3.97
512	2.8×10^{-5}	2.1	2.6×10^{-10}	4.01

TABLE 5 *Relative error for $f(z) = 72(z\bar{z}^2 + z^2\bar{z})$ using Euler–Maclaurin formula for the* (D3) *and* (D4) *problem with* $N = 64$

	For the biharmonic problem (D3)		for the biharmonic problem (D4)	
M	$\\|\cdot\\|_\infty$	Order	$\\|\cdot\\|_\infty$	Order
16	4.3×10^{-3}	—	6.4×10^{-5}	—
32	3.8×10^{-4}	3.5	7.6×10^{-6}	3.1
64	2.6×10^{-5}	3.8	4.6×10^{-7}	4.0
128	1.8×10^{-6}	3.8	2.7×10^{-8}	4.1
256	1.3×10^{-7}	3.7	1.6×10^{-9}	4.1
512	8.4×10^{-9}	3.9	1.02×10^{-10}	4.0

TABLE 6 *CPU timings and estimates for the constant c using Euler–Maclaurin example for* (D2) *and* (D4) *problems*

	For the biharmonic problem (D2)		For the biharmonic problem (D4)	
$M = N$	CPU time	c	CPU time	c
16	0.23	3.24×10^{-4}	0.25	3.52×10^{-4}
32	0.27	7.61×10^{-5}	0.28	7.89×10^{-5}
64	0.43	2.52×10^{-5}	0.45	2.64×10^{-5}
128	1.54	1.94×10^{-5}	1.55	1.95×10^{-5}
256	8.53	2.35×10^{-5}	8.55	2.35×10^{-5}
512	58.61	3.58×10^{-5}	58.79	3.58×10^{-5}

time required to solve two benchmark problems, namely biharmonic problems (D2) and (D3), for different values of N using Euler–Maclaurin integration scheme. The number of grid points in the radial and azimuthal directions have been taken to be same, N, which is the correct way to estimate of the complexity numerically for the purpose of comparison with the theoretical one. From this, we also compute the constant, c hidden in the order estimate by dividing the total computation time by $N^2 \log N$. The CPU timings and the estimate of the constant c for these two problems, (D2) and (D3), have been tabulated in Table 6 for several values of N. The CPU timings versus $N^2 \log N$ is plotted in Fig. 1(a), which

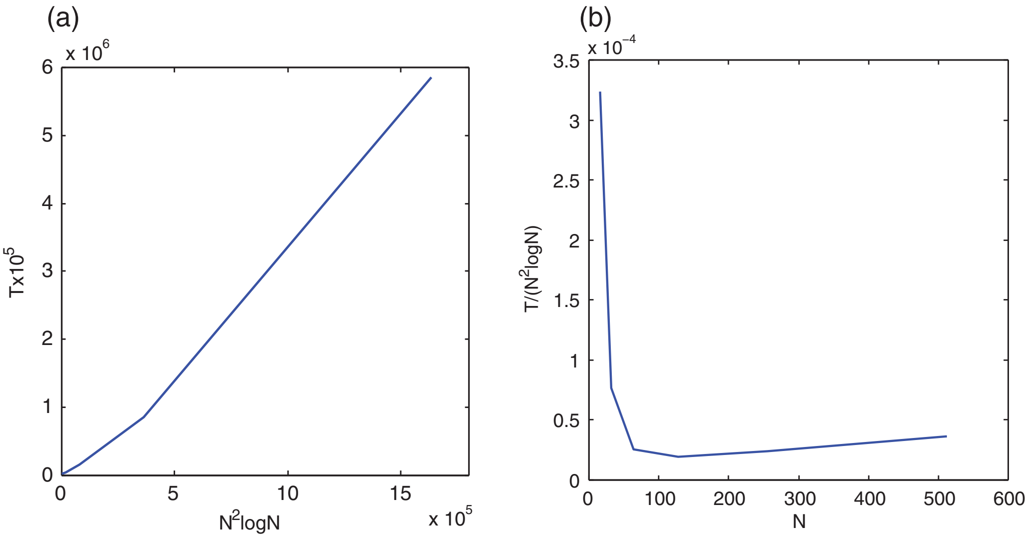

FIG. 1. Plots of the CPU timings and the constant $c = T/(N^2 \log N)$ for the fast algorithms applied to the biharmonic problems (D2) and (D3) problems using the Euler–Maclaurin formula. Plots for (D2) and (D3) problems in each of the (a) and (b) are indistinguishable from each other. (a) Plot of CPU timings versus $N^2 \log N$, where N^2 is the total number of points in the discretization of the domain. (b) Plot of $T/(N^2 \log N)$ versus N, where N^2 is the total number of points in the discretization of the domain.

is consistent with the theoretical complexity. More importantly, this figure shows that the constant c hidden behind the complexity estimate is very small (note that the abscissa of the plot is scaled up by 10^{-5}), which is further confirmed by the plot of computed values of c versus N in the Fig. 1(b). This is another significant advantage that these fast algorithms offer.

6. Error estimate for the singular integrals

For a given problem and choice of integration formula for the one-dimensional integrals in the fast algorithm, free parameters available to control the numerical error are M and N. The goal here is to show how M is determined *a priori* from prescribed values of N and maximum desired error ϵ_{p} in the solution of the problem. For this, we choose the principal solution $v(z)$ defined by the singular integral (2.4). We first show below the estimate for the error ϵ in its numerical evaluation of $v(z)$ by the fast algorithm.

LEMMA 6.1 For nonzero r and infinitely differentiable function f in $\mathbf{D}$, if each integral in step 2 of the fast algorithm (see Section 4.1) is computed with an error bound δ, then the error ϵ in the numerical evaluation of the principal solution $v(z)$, defined by the singular integral (2.4), is bounded by $2\delta(M-1)(K+2)/K$, where N is the total number of points in the azimuthal direction, M is the total number of points in the radial direction and $K = N/2$.

We provide some notations here before going to the proof. For $r_i < r_{i+1}$, the error in computing the integral $(n/2)A_n^{i,i+1}, (n/2)B_n^{i,i+1}, A_0^{i,i+1}, B_0^{i,i+1}$ are given by $\epsilon_{1,n}^{i,i+1}, \epsilon_{2,n}^{i,i+1}, \epsilon_{1,0}^{i,i+1}, \epsilon_{2,0}^{i,i+1}$ and $(A_n^{i,i+1})_{\mathrm{app}}$, $(B_n^{i,i+1})_{\mathrm{app}}$ are the approximate values of $A_n^{i,i+1}, B_n^{i,i+1}$, respectively. It follows from the RRs (2.26) in

Section 2.3.

$$p_{1,n}(r_l) = -\sum_{i=2}^{l}\left(\frac{r_i}{r_l}\right)^n B_n^{i-1,i}, \quad p_{2,n}(r_l) = -\sum_{i=l}^{M-1}\left(\frac{r_l}{r_i}\right)^n A_n^{i,i+1}, \quad n>0,$$

$$s_{1,n}(r_l) = \sum_{i=2}^{l}\left(\frac{r_l}{r_i}\right)^n A_n^{i-1,i}, \quad s_{2,n}(r_l) = \sum_{i=l}^{M-1}\left(\frac{r_i}{r_l}\right)^n B_n^{i,i+1}, \quad n<0,$$

$$t_{1,0}(r_l) = \sum_{i=l}^{M-1} A_0^{i-1,i}, \quad t_{2,0}(r_l) = \sum_{i=l}^{M-1} B_0^{i,i+1}.$$

Next we give the proof of the lemma.

Proof. Recall from Corollary 2.4 that $p_{2,n}(r_M=1)=s_{2,n}(r_M=1)=t_{2,0}(r_M=1)=0$. It follows from Theorem 2.1 that for $z=r_l\,e^{2\pi \mathrm{i}j/N}$, $l=2,3,\ldots,M$, $j=1,2,\ldots,N$

$$\begin{aligned}
v(r_l\,e^{2\pi \mathrm{i}j/N}) &= \sum_{n=-K+1}^{K}(v_n(r_l))\,e^{2\pi n\mathrm{i}j/N} = \sum_{n=1}^{K}(p_{1,n}(r_l)+p_{2,n}(r_l))\,e^{2\pi n\mathrm{i}j/N} \\
&\quad + \sum_{n=-K+1}^{-1}(s_{1,n}(r_l)+s_{2,n}(r_l))\,e^{2\pi n\mathrm{i}j/N} + (r_l\log r_l)t_{1,0}(r_l)+t_{2,0}(r_l) \\
&= \sum_{n=1}^{K}\left(-\sum_{i=2}^{l}\left(\frac{r_i}{r_l}\right)^n (B_n^{i-1,i})_{\mathrm{app}}\right)e^{2\pi n\mathrm{i}j/N} + \sum_{n=1}^{K}\left(-\sum_{i=l}^{M-1}\left(\frac{r_i}{r_l}\right)^n (A_n^{i,i+1})_{\mathrm{app}}\right)e^{2\pi n\mathrm{i}j/N} \\
&\quad + \sum_{n=-K+1}^{-1}\left(\sum_{i=2}^{l}\left(\frac{r_l}{r_i}\right)^n (A_n^{i-1,i})_{\mathrm{app}}\right)e^{2\pi n\mathrm{i}j/N} + \sum_{n=-K+1}^{-1}\left(\sum_{i=l}^{M-1}\left(\frac{r_i}{r_l}\right)^n (B_n^{i,i+1})_{\mathrm{app}}\right)e^{2\pi n\mathrm{i}j/N} \\
&\quad + \sum_{i=2}^{l} r_l\log r_l[(A_0^{i-1,i})_{\mathrm{app}}] + \sum_{i=l}^{M-1}[(B_0^{i,i+1})_{\mathrm{app}}] + \epsilon,
\end{aligned}$$

where

$$\begin{aligned}
|\epsilon| &\leqslant \sum_{n=1}^{K}\frac{2}{n}\sum_{i=2}^{l}\left(\frac{r_i}{r_l}\right)^n|\epsilon_{2,n}^{i-1,i}| + \sum_{n=1}^{K}\frac{2}{n}\sum_{i=l}^{M-1}\left(\frac{r_l}{r_i}\right)^n|\epsilon_{1,n}^{i,i+1}| + \sum_{n=-K+1}^{-1}\frac{2}{n}\sum_{i=2}^{l}\left(\frac{r_l}{r_i}\right)^n|\epsilon_{1n,1}^{i-1,i}| \\
&\quad + \sum_{n=-K+1}^{-1}\frac{2}{n}\sum_{i=l}^{M-1}\left(\frac{r_i}{r_l}\right)^n|\epsilon_{2,n}^{i,i+1}| + \sum_{i=2}^{l}|r_l\log r_l|\epsilon_{1,0}^{i-1,i}| + \sum_{i=2}^{l}|\epsilon_{2,0}^{i,i+1}|.
\end{aligned}$$

We have for $2<i<l, r_i<r_l$ and for $l<i<M, r_l<r_i$. If each integral is computed within an error bounded by δ, where

$$\delta = \max_{i,n,\rho}\left\{|\epsilon_{1,0}^{i-1,i}|, |\epsilon_{2,0}^{i,i+1}|, |\epsilon_{2,n}^{i-1,i}|, |\epsilon_{1,n}^{i,i+1}|, |\epsilon_{2,n}^{i,i+1}|, |\epsilon_{1,n}^{i-1,i}|\right\}. \tag{6.1}$$

Then

$$|\epsilon| \leqslant \sum_{n=1}^{K} \frac{2\delta}{n}(M-1) + \sum_{n=1}^{K} \frac{2\delta}{n}(M-1) + \sum_{n=-K+1}^{-1} \frac{2\delta}{n}(M-1) + \sum_{n=-K+1}^{-1} \frac{2\delta}{n}(M-1) + 2\delta(M-1)$$

$$= 2\delta(M-1) + 4\delta(M-1)\left(\sum_{n=1}^{K}\frac{1}{n} - \sum_{n=1}^{K-1}\frac{1}{n}\right) = \frac{2\delta(M-1)(K+2)}{K}. \tag{6.2}$$

This concludes the proof of the lemma. Therefore, for $|\epsilon| < \epsilon_{\mathrm{p}}$ we have

$$\frac{2\delta(M-1)(K+2)}{K} < \epsilon_{\mathrm{p}}, \tag{6.3}$$

from which an estimate for minimum value of M can be obtained as shown below through two examples. In the case of trapezoidal rule, it follows from (6.1) that δ in (6.3) is given by

$$\delta = \max_{i,n,\rho} \frac{(\Delta r)^3}{12}\left\{\frac{\partial^2}{\partial\rho^2}\left(\rho f_n(\rho)\left(\frac{r_{i+1}}{\rho}\right)^n\right), \frac{\partial^2}{\partial\rho^2}\left(\rho f_n(\rho)\left(\frac{\rho}{r_i}\right)^n\right), \frac{\partial^2}{\partial\rho^2}(4\rho\log\rho f_0(\rho))\right\}. \tag{6.4}$$

□

6.1 *Examples*

The above estimate has been validated using many examples. We give below only two simple case studies.

PROBLEM 6.2 We consider the function $f(z) = 12\bar{z}$. Hence,

$$f_n(\rho) = \begin{cases} 12 & \text{if } n = -1, \\ 0 & \text{if } n \neq -1. \end{cases}$$

Therefore, it follows from (6.4)

$$\delta = \max_{i,n,\rho} \frac{(\Delta r)^3}{12}\left\{\frac{\partial^2}{\partial\rho^2}\left(\frac{12\rho^3}{r_{i+1}}\right), \frac{\partial^2}{\partial\rho^2}(12 r_i \rho)\right\} = 6(\Delta r)^3. \tag{6.5}$$

It follows from (6.3) and (6.5)

$$(M-1)^2 > \frac{12(K+2)}{\epsilon_{\mathrm{p}} K}. \tag{6.6}$$

PROBLEM 6.3 We consider the function $f(z) = 6z\bar{z}^2$. Hence,

$$f_n(\rho) = \begin{cases} 6\rho^3 & \text{if } n = 1, \\ 0 & \text{if } n \neq 1. \end{cases}$$

Therefore, it follows from (6.1)

$$\delta = \max_{i,n,\rho} \frac{(\Delta r)^3}{12}\left\{\frac{\partial^2}{\partial\rho^2}\left(\frac{12\rho^3}{r_{i+1}}\right), \frac{\partial^2}{\partial\rho^2}\left(\frac{6\rho^5}{r_i}\right)\right\} = 10(\Delta r)^3. \tag{6.7}$$

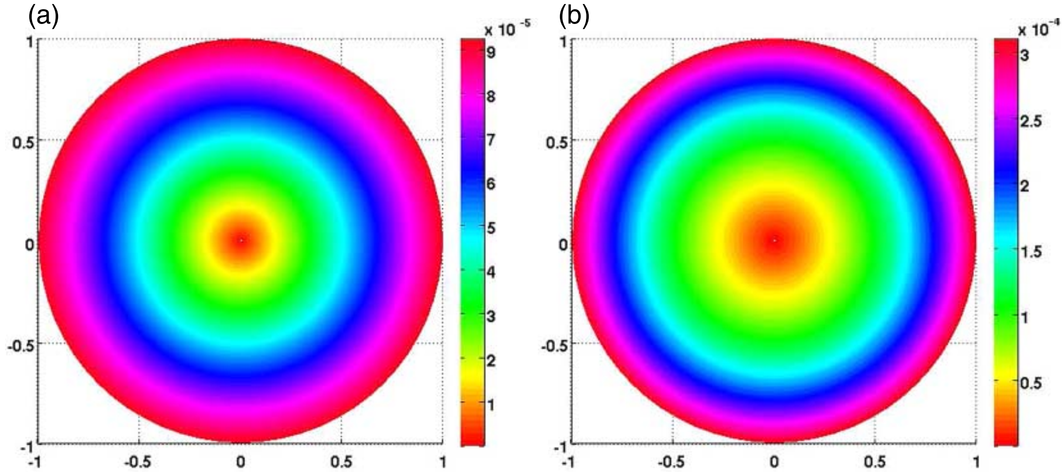

FIG. 2. The point-wise error inside the disk for two problems. This is a colour plot on screen. (a) Error plot for with $f(z) = 12\bar{z}$, $N = 64, M = 256$. (b) Error plot with $f(z) = 6z^2\bar{z}$, $N = 64, M = 128$.

It follows from (6.3) and (6.7)

$$(M-1)^2 > \frac{20(K+2)}{\epsilon_\mathrm{p} K}. \tag{6.8}$$

For the above two problems, we have numerically evaluated $v(z)$ (see (2.4)) inside the disk using the fast algorithm using minimum values of M determined by the above estimates ((6.6) for problem 1 and (6.8) for problem 2) for several choices of ϵ_p, and also evaluated $v(z)$ exactly since exact integration of the singular integral (2.4) is possible for both the problems, thereby obtained numerical errors in the computed values of $v(z)$ for both the problems. We have found that the numerical error in the computed values of $v(z)$ inside the disk at all grid points is always less than the prescribed error ϵ_p for the problems. For brevity, we show only two figures.

For a prescribed error $\epsilon_\mathrm{p} = 0.0001$ and $N = 64$ for problem 1, we obtain $M = 358$ from the theoretical estimate (6.6). Figure 2 shows point-wise error plot inside the disk for this problem when $N = 64, M = 256$. We see that the error does not exceed this prescribed error. Similarly for problem 2, for a prescribed error $\epsilon_\mathrm{p} = 0.001$ and $N = 64$, we obtain $M = 147$ from the theoretical estimate (6.8). Figure 2(b) shows point-wise error plot inside the disk for this problem when $N = 64, M = 128$. We see that the error is less than the prescribed error. Note for both problems, the numerical error is within prescribed errors with values of M less than the estimates.

7. Application to low and moderate Reynolds number flow

We apply our fast algorithms for the biharmonic problems to study steady and viscous incompressible flows inside a cylinder. Such flows have been studied by many (see Mills, 1977, for example). The practical application of these types of problems arises in recirculation of fluids in cavities and in confined ventilation (see Mills, 1977). The governing equations for such flows are given by

$$(\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \nabla\cdot(R^{-1}\nabla\mathbf{u}), \quad \nabla\cdot\mathbf{u} = 0, \tag{7.1}$$

where p is the pressure, R is the Reynolds number and $\mathbf{u}$ is the velocity. The velocity components (u_r, u_θ) in terms of stream function ψ are given by

$$u_r = \frac{1}{r}\frac{\partial \boldsymbol{\psi}}{\partial \theta}, \quad u_\theta = -\frac{\partial \boldsymbol{\psi}}{\partial r}. \tag{7.2}$$

The scalar vorticity φ satisfies the Poisson equation $\varphi = -\Delta\psi$. Taking curl of both sides of (7.1) gives

$$\Delta^2\psi = -RJ[\psi, \Delta\psi], \tag{7.3}$$

where the Jacobian $J[\psi, \Delta\psi] = (1/r)(\partial_r\psi\,\partial_\theta\Delta\psi - \partial_r\Delta\psi\,\partial_\theta\psi)$. For $R \to 0$, equation (7.3) is the homogeneous biharmonic equation. In terms of generalized derivatives, $\Delta^2 = (\partial_z\partial_{\bar{z}})^2/16$. Therefore, a typical flow problem inside the cylinder for prescribed boundary values of ψ and $\partial\psi/\partial r$ is then given by (see Kuwahara & Imai, 1969; Mills, 1977)

$$\left.\begin{aligned} (\partial_z\partial_{\bar{z}})^2\psi &= -\frac{R}{16}J[\psi, \psi_{z\bar{z}}] && \text{in } r < 1,\\ \psi &= f_1(\theta) && \text{on } r = 1,\\ \frac{\partial\psi}{\partial r} &= f_2(\theta) && \text{on } r = 1, \end{aligned}\right\} \tag{7.4}$$

where the Reynolds number $R = Ur/\nu$ based radius r of the cylinder, U is the speed of rotation of the part of the cylinder wall and ν is the kinematic viscosity of the fluid. With slight abuse of notation, we use above the same notation ψ when $\psi(x, y)$ is written as a function of $z = (x + \mathrm{i}\,y)$ and $\bar{z} = (x - \mathrm{i}\,y)$. The problem (7.4) above is the biharmonic problem (D3) (see Section 3), except that the inhomogeneous term in the biharmonic equation $(7.4)_1$ depends on the solution itself. Therefore, the problem (7.4) is solved using an iteration scheme in which the fast algorithm discussed in Section 4.3 is used to solve the (D3) biharmonic problem at every iteration until some suitable convergence criterion is met.

We apply this iterative method and the fast algorithm developed in previous sections to specific fluid flow problems which will validate the application potential of our algorithms. We first consider slow creeping Stokes flows when $R \to 0$. In this case, the above problem (7.4) reduces to the (D3) *homogeneous* biharmonic problem. This is first solved using our fast algorithm discussed in Section 4.3. The solution of this zero Reynolds number problem serves as an initial guess to solve the problem (7.4) for flows with nonzero Reynolds number. In particular, we are first interested in flows with low Reynolds number, for which we follow a simple iteration method described in Greengard & Kropinski (1998) and Mills (1977).

We start with an initial guess $\psi^{(0)}$ obtained from the solution of Stokes flow, and then at each $(k+1)$th stage we solve

$$\left.\begin{aligned} \psi^{(k+1)}_{z\bar{z}z\bar{z}} &= -\frac{R}{16}J[\psi^{(k)}, \psi^{(k)}_{z\bar{z}}] && \text{in } r < 1,\\ \psi^{k+1} &= f_1(\theta) && \text{on } r = 1,\\ \frac{\partial\psi^{(k+1)}}{\partial r} &= f_2(\theta) && \text{on } r = 1, \end{aligned}\right\} \tag{7.5}$$

using the fast algorithm for the inhomogeneous biharmonic problem (D3) as mentioned above. The vorticity φ is obtained from $\varphi = -\Delta\psi$ using numerical approximation of the Laplacian. The Jacobian is obtained using the central difference formula on mesh points inside the disk and backward difference, forward difference for points on the boundary. We continue the iteration until the convergence criterion $(\|\psi^{k+1}\| - \|\psi^{k}\|)/\|\psi^{k+1}\| <$ tol is met for some suitable choice for the value of the tolerance. In our case, we used tol $= 3 \times 10^{-4}$.

In our numerical experiments, the above iteration method failed for Reynolds number beyond 4. This is well known as others have also experienced the same problem with this iteration method (Mills, 1977; Greengard & Kropinski, 1998) in the past. Therefore, we modify our iteration method for $R > 4$. We use a relaxation factor as in the Gauss–Seidel SOR method (see Mills, 1977). We start with our initial guess as before to obtain $\psi^{(0)}(z)$ and use the fast algorithm to compute the iterate $\psi^{(k+1)}(z)$. For convergence, we use two relaxation factors α and β for fields φ and ψ. To update the values of φ and ψ, we use

$$\varphi_{n,l}^{(k+1)} = \alpha\varphi_{n,l}^{\star(k+1)} + (1-\alpha)\varphi_{n,l}^{(k)}, \quad \psi_{n,l}^{(k+1)} = \beta\psi_{n,l}^{\star(k+1)} + (1-\beta)\psi_{n,l}^{(k)}.$$

The starred quantities denote the values obtained at each iterative step. The relaxation factor helps in convergence, and suitable choices for α and β are taken to be 0.3 and 0.5, respectively. The iteration is continued until tolerance is met. This iteration method works for problems with moderate Reynolds number, but becomes unstable for problems with high Reynolds number. Similar performance of this iteration method has been observed in Mills (1977). Results on several flow problems are discussed below.

The flow problem we solve has boundary data $f_2(\theta) = \cos\theta\,\sin\theta$ and $f_1(\theta) = 0$, taken from Greengard & Kropinski (1998). The flow problem is solved for two values of R: 10 and 150. The number of iterations required to obtain converged solution within a tolerance = 3×10^{-4} is 4 at $R = 10$ and 26 at $R = 150$, respectively. Streamline plots at $R = 10$ and $R = 150$ are shown in Fig. 3(a,b), respectively. Similar plots for vorticity are shown in Fig. 4(a,b), respectively. The flow pattern is symmetric about x- and y-axes, as it should be due to the same symmetry in the boundary data $f_2(\theta)$. The plots agree with those obtained by Greengard & Kropinski (1998). This flow was investigated for $0 \leqslant R \leqslant 150$. Significant change in the vorticity pattern is observed with increasing Reynolds number.

Now we consider the moving wall problem with boundary condition $\psi(r = 1) = 0$, and discontinuous boundary condition

$$\frac{\partial\psi}{\partial r} = \begin{cases} -1, & 0 \leqslant \theta < \pi, \\ 0, & \pi \leqslant \theta < 2\pi. \end{cases} \tag{7.6}$$

This flow problem was solved for $0 \leqslant R \leqslant 20$. The computed streamline patterns at $R = 0$ and $R = 10$ are shown in Fig. 5(a,b), respectively. Nonsymmetric flow patterns are observed here, and the centre of the vortex is seen shifted to the direction of the flow. The number of iterations required at $R = 10$ is 27. For computation of this flow problem, the number of grid points used is taken more (see the figure caption) than for the other problems due to the presence of discontinuity in the boundary condition of this problem (see (7.6)). This problem has also been studied by Mills (1977) and Mabey (1957), and our results are in excellent agreement with their results.

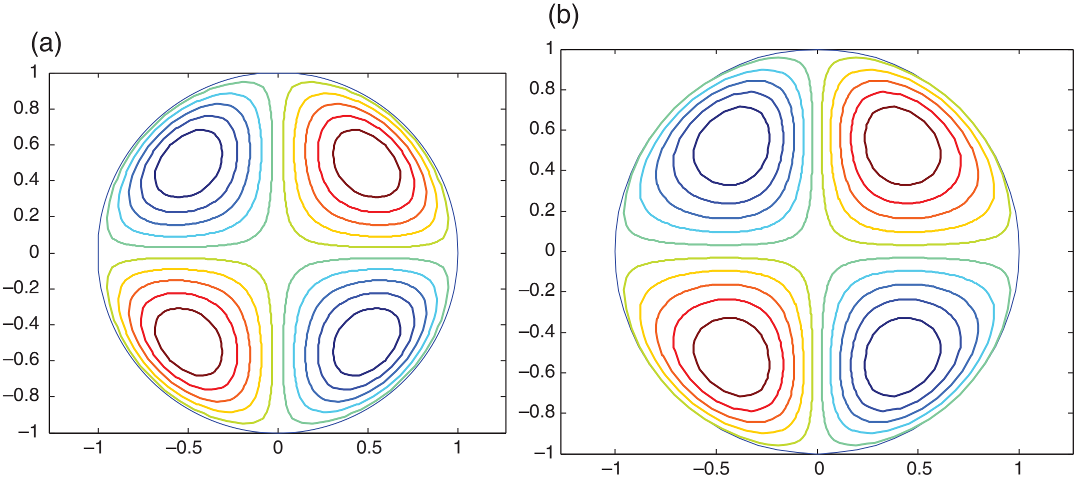

FIG. 3. Streamline patterns for flows with $f_1(\theta) = 0$, and $f_2(\theta) = \cos\theta \sin\theta$. Computations have been performed with parameter values $N = 64, M = 64$ using the fast Algorithm 4.3 in a (D3) biharmonic problem. This is a colour plot on screen. (a) Streamline patterns for $R = 10$. (b) Streamline patterns for $R = 150$.

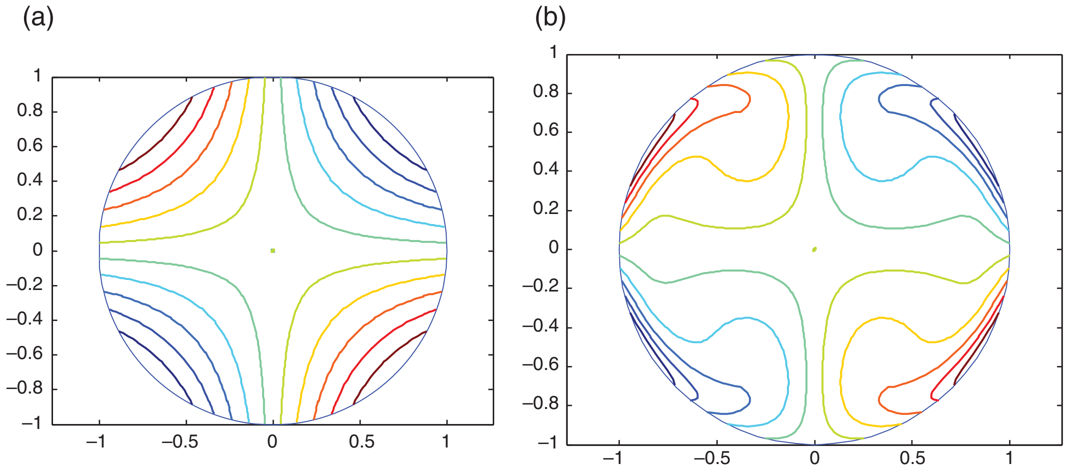

FIG. 4. Level sets of vorticity for flows with $f_1(\theta) = 0$, and $f_2(\theta) = -\cos\theta \sin\theta$. Computations have been performed with parameter values $N = 64, M = 64$ using the fast Algorithm 4.3 in a (D3) biharmonic problem. This is a colour plot on screen. (a) Level sets of vorticity for $R = 10$. (b) Level sets of vorticity for $R = 150$.

Next we show our results on the following outflow inflow problem (taken from Mills (1977)), in which ψ is prescribed on the boundary as a piecewise continuous data.

$$\frac{\partial \psi}{\partial r} = 0, \quad 0 < \theta < 2\pi, \tag{7.7}$$

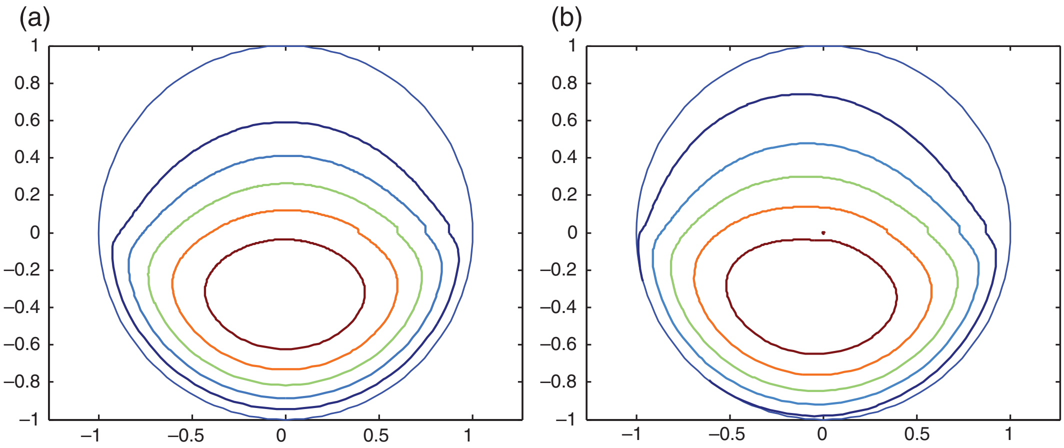

FIG. 5. Streamlines for flows with $f_1(\theta) = 0$ and discontinuous $f_2(\theta)$ given by (7.6). Computations have been performed with parameter values $N = 128$, $M = 129$ using the fast Algorithm 4.3 in a (D3) biharmonic problem. This is a colour plot on screen. (a) Streamline patterns for $R = 0$. (b) Streamline patterns for $R = 10$.

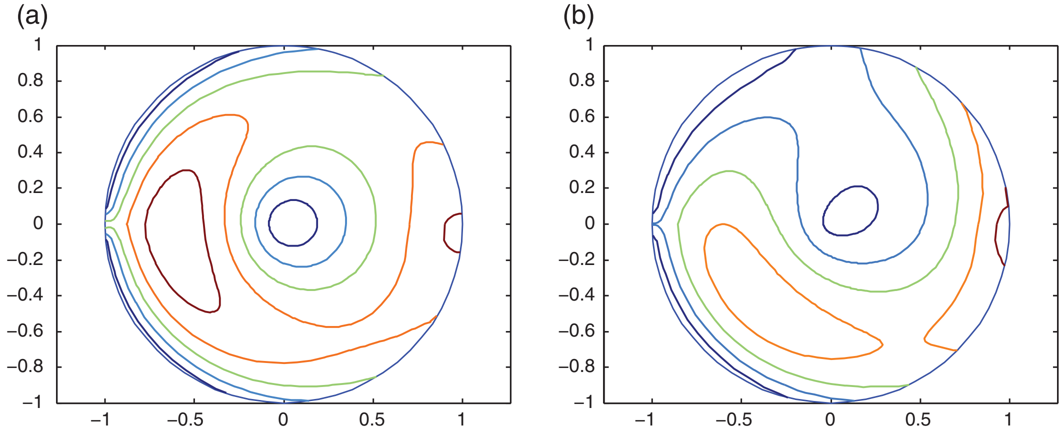

FIG. 6. Streamlines for flows with $f_1(\theta) = 0$ and discontinuous $f_2(\theta)$ given by (7.8). Computations have been performed with parameter values of $N = 128, M = 65$ using the fast algorithm 4.3 in a (D3) biharmonic problem. This is a colour plot on screen. (a) streamline patterns for $R = 0.009$. (b) Streamline patterns for $R = 0.02$.

$$\psi = \begin{cases} 1 + \dfrac{(\theta - \alpha)}{\epsilon}, & \alpha - \epsilon < \theta < \alpha + \epsilon, \\ 2, & \alpha + \epsilon < \theta < \beta - \epsilon, \\ 1 + \dfrac{(\beta - \theta)}{\epsilon}, & \beta - \epsilon < \theta < \beta + \epsilon, \\ 0, & \beta + \epsilon < \theta < 2\pi + \alpha - \epsilon. \end{cases} \tag{7.8}$$

In our computation, $\alpha = 0$, $\epsilon = \pi/32$, $\beta = \pi$ and $R = U\epsilon/\nu$ is the Reynolds number, where U is the speed and $U\epsilon$ the flow across the arc intercepted by ϵ. The number of iterations required to compute this flow at $R = 0.009$ and $R = 0.02$ is 13 and 20, respectively. Computed streamline patterns at $R = 0.009$ and $R = 0.02$ are shown in Fig. 6(a,b), respectively, which agree very well with that obtained by Dennis (1975).

8. Conclusions

We have developed FFTRR-based fast and high-order accurate algorithms for solving the complex Poisson and the complex inhomogeneous biharmonic problems. Algorithms based on these same principles have been developed before by Borges & Daripa (2000, 2001), Daripa (1992, 1993) and Daripa & Mashat (1998) for other types of elliptic problems. For ease of reference in the future for algorithms based on these principles, we have named these algorithms for the first time in this paper as FFTRR-based algorithms. These algorithms have many desirable features which are listed in the Section 1. These algorithms for the complex Poisson problems have been applied here to solve four classes of inhomogeneous complex biharmonic problems using the decomposition methods, resulting in efficient and accurate fast biharmonic solvers. These biharmonic solvers have been applied to solve steady, incompressible slow viscous flow problems for low to moderate Reynolds number within circular cylinders. The computational results agree very well with the existing results on these problems. These algorithms can also be applied to solving similar problems in arbitrary domains using domain embedding technique (Badea & Daripa, 2001, 2002, 2003, 2004), which is a topic of future research.

In closing, we mention the fast method for rectangular domains developed by Ben-Artzi *et al.* (2008). There are even more fourth-order accurate methods for solving Poisson's equation on regular regions with the same $O(\log n)$ operation count per mesh point, where the constant behind the complexity estimate is small. Performance and various other features of the algorithms proposed here need to be compared with that of a variety of other fourth-order accurate methods that have been developed by many in the past for solving the biharmonic equation on a disk with uniform meshes with the same asymptotic $O(\log n)$ operation count per mesh point. This is a topic of future research and falls outside the scope of this paper.

On theoretical ground, one can easily see that these algorithms, similar to other algorithms developed by Borges & Daripa (2000, 2001), Daripa (1992, 1993) and Daripa & Mashat (1998), offer several *distinct advantages over the existing ones* with similar accuracy and complexity: (i) analytical methods (i.e., exact RRs) in conjunction with FFT provides accurate and computationally efficient algorithms; (ii) these algorithms allow *a priori* selection of the uniform mesh sizes from prescribed desired error in the solution (see Section 6); (iii) compared with methods based on discretization of the PDE over regular mesh in polar coordinates, there is no need for any special treatment near coordinate singularity at the origin; (iv) solutions locally near any ring in the disk can be computed without computing solutions everywhere else. This property is built into the algorithms and will be exemplified in the future; (v) the algorithms automatically simplifies tremendously when solving the real biharmonic equation with boundary data real (i.e., when inhomogeneous term f and the boundary conditions are real). This can be easily worked out by using the fact that $f_n = \overline{f_{-n}}$ for real f; (vi) these FFTRR-based algorithms by their very construction allow construction of exact explicit solutions of a certain class of problems, in particular when Fourier coefficients of the source term f can be written explicitly as a function of the radial coordinate. This can be easily seen by analyzing the algorithms given in this paper. This will be exemplified in the future.

Acknowledgements

A.G. thanks the Department of Mathematics at Texas A&M University for teaching assistantship during her graduate study when this work was completed. The authors are grateful to the reviewers for their constructive criticisms which have helped us to improve the paper.

References

Badea, L. & Daripa, P. (2001) On a boundary control approach to domain embedding methods. *SIAM J. Control Optim.*, **40**, 421–449.

Badea, L. & Daripa, P. (2002) A domain embedding/boundary control method to solve elliptic problems in arbitrary domains. *IEEE Conference on Decision and Control*, vol. 3, pp. 3004–3009.

Badea, L. & Daripa, P. (2003) On a Fourier method of embedding domains using an optimal distributed control. *Numer. Algorithms*, **32**, 261–273.

Badea, L. & Daripa, P. (2004) A domain embedding method using the optimal distributed control and a fast algorithm. *Numer. Algorithms*, **36**, 95–112.

Begehr, H. (2007a) Biharmonic Green functions. *Matematiche (Catania)*, **61**, 395–409.

Begehr, H. (2007b) Basic boundary value problems in complex analysis. *J. Appl. Funct. Anal.*, **2**, 57–71.

Begehr, H. (2008) Six biharmonic Dirichlet problems in complex analysis. *Function Spaces in Complex and Clifford Analysis* (L. H. Son & W. Tutschke eds). Hanoi: National University Publ. Hanoi, pp. 243–252.

Ben-Artzi, M., Croisille, J.-P. & Fishelov, D. (2008) A fast direct solver for the biharmonic problem in a rectangular grid. *SIAM J. Sci. Comput.*, **31**, 303–333.

Bjorstad, P. (1983) Fast numerical solution of the biharmonic Dirichlet problem on rectangles. *SIAM. J. Numer. Anal.*, **20**, 626–668.

Borges, L. & Daripa, P. (2000) A parallel version of a fast algorithm for singular integral transforms. *Numer. Algorithms*, **23**, 71–96.

Borges, L. & Daripa, P. (2001) A fast parallel algorithm for the Poisson equation on a disk. *J. Comput. Phys.*, **169**, 151–192.

Braess, D. & Peisker, P. (1986) On the numerical solution of the biharmonic equation and the role of squaring matrices for preconditioning. *IMA J. Numer. Anal.*, **6**, 393–404.

Cheng, X.-L., Han, W. & Huang, H.-C. (2000) Some mixed finite element methods for biharmonic equation. *J. Comput. Appl. Math.*, **126**, 91–109.

Daripa, P. (1992) A fast algorithm to solve nonhomogeneous Cauchy–Riemann equations in the complex plane. *SIAM J. Sci. Statist. Comput.*, **13**, 1418–1432.

Daripa, P. (1993) A fast algorithm to solve the Beltrami equation with applications to quasiconformal mappings. *J. Comput. Phys.*, **106**, 355–365.

Daripa, P. & Mashat, D. (1998) Singular integral transforms and fast numerical algorithms. *Numer. Algorithms*, **18**, 133–157.

Dennis, S. C. (1975) Application of the series truncation method to two-dimensional internal flows. *Proceedings of the Fourth International Conference on Numerical Methods in Fluid Dynamics*, vol. 4, pp. 138–143.

Greenbaum, A., Greengard, L. & Mayo, A. (1992) On the numerical solution of the biharmonic equation in the plane. *Physics D*, **60**, 216–225. Experimental mathematics: computational issues in nonlinear science (Los Alamos, NM, 1991).

Greenberg, M. D. (1971) *Application of Green's Functions in Science and Engineering*. Berlin: Prentice-Hall. Green's function.

Greengard, L. & Kropinski, M. C. (1998) An integral equation approach to the incompressible Navier–Stokes equations in two dimensions. *SIAM J. Sci. Comput.*, **20**, 318–336.

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

GUAZZELLI, É. & MORRIS, J. F. (2012) *A Physical Introduction to Suspension Dynamics*. Cambridge Texts in Applied Mathematics. Cambridge: Cambridge University Press.

KROPINSKI, M. C. A. (1999) Integral equation methods for particle simulations in creeping flows. *Comput. Math. Appl.*, **38**, 67–87.

KUWAHARA, K. & IMAI, I. (1969) Steady viscous flow within a circular boundary. *Phys. Fluids.*, **12**, 94–101.

LAI, M. (2005) Fast direct solver for the biharmonic equation on a disk and its application to incompressible flows. *Appl. Math. Comput.*, **164**, 679–695.

LOVE, A. (1927 *A Treatise on the Mathematical Theory of Elasticity*. New York: Dover Press, reprinted 1994.

MABEY, D. (1957) Fluid dynamics. *J. Royal Aeronaut. Soc.*, **61**, 181–198.

MAYO, A. (1992) The rapid evaluation of volume integrals of potential theory on general regions. *J. Comput. Phys.*, **100**, 236–245.

MILLS, R. D. (1977) Computing internal viscous flow problems for the circle by integral methods. *J. Fluid Mech.*, **79**, 609–624.

MONK, P. (1987) A mixed finite element method for the biharmonic equation. *SIAM J. Numer. Anal.*, **24**, 737–749.

MUSKHELISHVILI, N. (1977) *Some Basic Problems of the Mathematical Theory of Elasticity*. Berlin: Springer (English translation).

PEISKER, P. (1988) On the numerical solution of the first biharmonic equation. *RAIRO Modél. Math. Anal. Numér.*, **22**, 655–676.

SIDI, A. & ISRAELI, M. (1988) Quadrature methods for periodic singular and weakly singular Fredholm integral equations. *J. Sci. Comput.*, **3**, 201–231.

WANG, Z., WEI, X. & GAO, X. (1991) Solution of the plane stress problems of strain-hardening materials described by power-law using the complex pseudo-stress function. *Appl. Math. Mech.*, **12**, 481–492.