A finite difference Poisson solver for irregular geometries

Z. Jomaa*   C. Macaskill†

(Received 8 August 2003, revised 21 January 2004)

Abstract

The motivation for this work comes from the development of a 3D quasi-geostrophic Contour Advective Semi-Lagrangian model for vortex interaction in the ocean. The existing code is limited to circular cylindrical geometry and uses polar coordinates. We wish to extend the method to more general cross-sections. The crucial aspect is the solution of the Poisson equation that allows the determination of the stream function from the potential vorticity at each time-step, as this is the part of the algorithm that must be performed on a grid: the advection of potential vorticity contours is fully Lagrangian and hence is easily modified for irregular domains. We develop a 2D algorithm

---

*School Maths & Statistics, University of Sydney, AUSTRALIA. mailto:ziadj@maths.usyd.edu.au

†School Maths & Statistics, University of Sydney, AUSTRALIA.

for inverting the Poisson equation for the stream function on an arbitrarily shaped domain, in the special case when the boundary is a streamline, as is the case for our problem. However, the method is also valid for non-zero Dirichlet boundary conditions. The approach uses finite differences with the domain embedded in a rectangular Cartesian grid. We show that the algorithm is second-order accurate and provide several numerical examples.

Contents

1   Introduction

The Poisson equation is one of the fundamental equations in mathematical physics. It occurs in a broad range of applications including acoustics, electromagnetism and fluid mechanics: our specific application is the solution of the Poisson equation that recovers the stream function from the vorticity in inviscid vortex dynamics. Our aim is to solve the Poisson equation on an irregularly shaped, but smooth, domain as a critical part of ongoing work on

vortex dynamics, extending previous work in a circular domain [5]. As the boundary of this domain is a streamline, we limit our approach to Dirichlet boundary conditions.

There are many different approaches to this problem in the literature. The immersed boundary method [12] uses a δ-function on the domain boundary to enforce a no flow boundary condition; see [11] for details. A related approach called the immersed interface method is a second-order numerical method designed to preserve the jump condition at the interface [3]. An alternative approach, using boundary integral techniques, has been explored in a sequence of papers [7, 8, 9, 10].

Liu, Fedkiw and Kang [4] develop a first-order accurate symmetric discretization of the variable-coefficient Poisson equation in the presence of an irregular interface across which the variable coefficient, the solution and the derivative of the solution may have jumps. These ideas were extended by Gibou et al. [1] to give a second-order accurate symmetric discretization for the Poisson equation with Dirichlet boundary conditions.

Here we follow Johansen and Colella [2]. In their approach, the physical domain is embedded in a rectangular domain, where it is assumed that the unknown takes zero values outside the physical boundary—therefore there is in general a jump at the boundary. Johansen and Colella solved the Poisson equation with Dirichlet boundary conditions, as we do here. However, their treatment of the boundary in the 2D problem uses a local area fit to determine the boundary treatment, which is also second-order accurate, but gives rise to larger errors at moderate grid-point density than the current method. Furthermore, the current technique uses the simpler idea of applying the 1D boundary treatment over x and y separately. In principle, we can therefore directly extend the current approach to 3D, but we have not yet attempted this.

In this paper we present a numerical method for solving the Poisson equation with Dirichlet boundary conditions on a bounded 2D region Ω.

Our approach uses a finite difference discretization. The Poisson equation is discretized at each grid-point. For grid-points away from the boundary the algorithm uses the standard five-point discretization for second derivatives with second-order truncation error. The slopes at the edge points are computed to second-order accuracy (that is, using a quadratic fit), so that the second derivatives at grid-points just inside the boundary are in fact only first-order accurate. On the boundary, this first-order truncation error produces a third-order accurate solution; however the overall solution (that is, at interior points) is second-order accurate. This at first sight seems inconsistent, in that it might be adequate to use linear fitting at the boundaries to ensure uniform error across the domain, but such a procedure leads to a coefficient of error at the boundary that is large enough to dominate the calculation. The discretization used here gives rise to a pentadiagonal matrix system, as for a rectangular domain, but matrix elements corresponding to boundary points depend on the local form of the boundary.

For the remainder of this paper, we will give details of the algorithm. In Section 2, we describe the discretization in 1D and provide some analysis of the accuracy of the method. We explain that the extension to 2D is straightforward. Finally, we show some of the numerical results in Section 3.

2   Mathematical Formulation

Our aim is to solve the Poisson equation

$$\nabla^2 \psi = f(x, y), \tag{1}$$

on a domain Ω with Dirichlet boundary conditions $\psi = g(x, y)$ given on the boundary $\partial\Omega$.

2.1   1D case

To solve the 2D problem we make use of the following 1D analysis, employed first in the x-direction and then in the y-direction. Furthermore, in the 1D case we can explicitly derive the error involved, so as to understand how boundary error contributes to the overall error.

The 1D Poisson equation is just

$$\frac{d^2\psi}{dx^2} = f(x)\,. \tag{2}$$

and we choose a uniform grid over the domain $x \in [e, d]$. However, we take the domain of interest Ω to be the interval $x \in [x_{\mathrm{jumpL}}, x_{\mathrm{jumpR}}]$, with specified Dirichlet conditions at these (typically non-grid) points. Outside the interval we set $\psi = 0$, so that in general there is a discontinuity at each of x_{jumpL} and x_{jumpR}: with zero boundary conditions this will reduce to a jump in slope. Then we label the points between the jumps so that $e = x_0 < x_{\mathrm{jumpL}} < x_1 < x_2 < \cdots < x_{N-2} < x_{N-1} < x_{\mathrm{jumpR}} < d = x_N$, the distance between x_0 and x_{jumpL} is $\alpha_L \Delta x$ and the distance between x_{jumpR} and x_N is $\alpha_R \Delta x$ (Figure 1).

We denote by $x_{i+1/2}$ the midpoint of the interval $[x_i, x_{i+1}]$. For all the points in $\Omega \backslash [x_0, x_1] \cup [x_{N-1}, x_N]$, we use centered differences to approximate the gradient at the midpoint:

$$\psi'_{k+1/2} = \frac{\psi_{k+1} - \psi_k}{\Delta x}\,. \tag{3}$$

To approximate $d\psi/dx$ at $x = x_{1/2}$ (in the left-most interval), we fit a quadratic polynomial through the values ψ_{jumpL}, ψ_1 and ψ_2, and evaluate the corresponding slope at $x = x_{1/2}$ (Figure 1).

$$\psi'_{1/2} = \frac{1}{\Delta x}\left[-\frac{2}{(1-\alpha_L)(2-\alpha_L)}\psi_{\mathrm{jumpL}} + \frac{1+\alpha_L}{1-\alpha_L}\psi_1 - \frac{\alpha_L}{2-\alpha_L}\psi_2\right]\,. \tag{4}$$
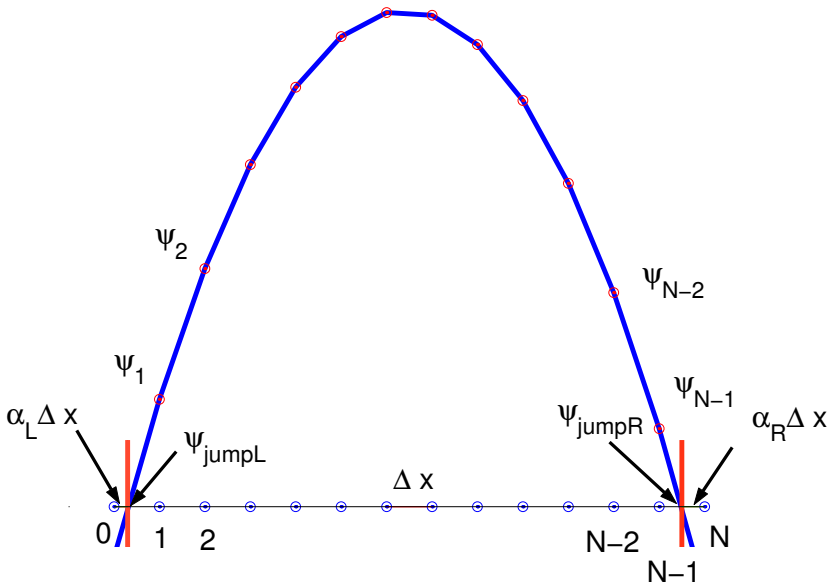
FIGURE 1:   Diagram of the second-order stencil for the gradient at each end of the interval. A quadratic is fitted to the values of ψ at the two neighbouring points and boundary point.

Similarly, to approximate $d\psi/dx$ at $x = x_{N-1/2}$, we fit a quadratic polynomial through the values ψ_{jumpR}, ψ_{N-1} and ψ_{N-2}, and evaluate the corresponding slope at $x = x_{N-1/2}$ (Figure 1).

The discretization of equation (2) gives rise to a tridiagonal matrix equation for the unknown ψ on the interior grid-points; for the exterior grid-points we set $\psi = 0$. At each grid-point $x = x_k$, (2) is discretized by the 1D standard second derivative

$$\frac{1}{\Delta x}\left[\left(\frac{\psi_{k+1} - \psi_k}{\Delta x}\right) - \left(\frac{\psi_k - \psi_{k-1}}{\Delta x}\right)\right] = f_k. \tag{5}$$

However at $x = x_1$, equation (2) becomes

$$\frac{1}{(\Delta x)^2}\left[\frac{2}{(1-\alpha_L)(2-\alpha_L)}\psi_{\text{jumpL}} - \frac{2}{1-\alpha_L}\psi_1 + \frac{2}{2-\alpha_L}\psi_2\right] = f_1. \tag{6}$$

A similar discretization applies at $x = x_{N-1}$.

2.1.1   Error estimate for the 1D case

This analysis follows the ideas of Johansen and Colella [2], but carries through the evaluation of the error terms explicitly so that it becomes clear how the boundary truncation error contributes to the overall error in the solution.

We define the second derivative operator

$$(L\xi)_{i+1} = \frac{H_{i+3/2} - H_{i+1/2}}{\Delta x} = \tau_{i+1}, \tag{7}$$

where H is the first derivative of ξ. The truncation error

$$\tau_i = f_i - (L\psi^e)_i,$$

where ψ^e is the exact solution. The error $\xi = \psi - \psi^e$ satisfies the following system of equations:

$$L\xi = \tau, \quad \xi_0 = \xi_N = 0. \tag{8}$$

Solving (8) for ξ, we obtain

$$\xi_1 = (1 - \alpha_L)(\Delta x)^2 \left[\frac{H_{N-1/2}}{\Delta x} - \frac{(2 - \alpha_L)}{2}\tau_1 - \sum_{k=2}^{N-1} \tau_k \right]. \tag{9}$$

Similarly for ξ_{N-1}; but for $2 \leq k \leq N - 2$

$$\xi_k = (\Delta x)^2 \left[(k - \alpha_L)\frac{H_{N-1/2}}{\Delta x} \right.$$
$$\left. - (1 - \alpha_L)\left(\frac{(2 - \alpha_L)}{2}\tau_1 + \sum_{m=2}^{N-1}\tau_m \right) - \sum_{m=1}^{k-1}\sum_{p=m+1}^{N-1}\tau_p \right]. \tag{10}$$

$$H_{N-1/2} = \Delta x \left\{ -(1 - \alpha_L)\left[\frac{(2 - \alpha_L)}{2}\tau_1 + \sum_{j=2}^{N-1}\tau_j \right]\left[\frac{1 + \alpha_R}{1 - \alpha_R} - \frac{\alpha_R}{2 - \alpha_R} \right] \right.$$
$$\left. - \frac{1 + \alpha_R}{1 - \alpha_R}\sum_{j=1}^{N-2}\sum_{k=j+1}^{N-1}\tau_k + \frac{\alpha_R}{2 - \alpha_R}\sum_{j=1}^{N-3}\sum_{k=j+1}^{N-1}\tau_k \right\}$$
$$\times \left[1 + (N - 1 - \alpha_L)(\frac{1 + \alpha_R}{1 - \alpha_R}) - (N - 2 - \alpha_L)\frac{\alpha_R}{2 - \alpha_R} \right]^{-1}. \tag{11}$$

Here $H_{N-1/2}$ is $\mathcal{O}((\Delta x)^2)$, so that the total error at the end-points ξ_1 and ξ_{N-1} is $\mathcal{O}((\Delta x)^3)$. However, for $2 \leq k \leq N - 2$ the error at internal grid-points, ξ_k, is $\mathcal{O}((\Delta x)^2)$. This internal error in fact dominates and gives rise to overall error of order $\mathcal{O}((\Delta x)^2)$ due to the small coefficients for ξ_1 and ξ_{N-1}.

2.2   2D case

Consider the 2D Poisson equation (1), and let Ω be any irregular 2D shape inscribed within a rectangle with boundary $\partial\Omega$ at which Dirichlet conditions $\psi(x, y) = g(x, y)$ are specified. As $\psi = 0$ outside the physical domain, there may be jumps on $\partial\Omega$. We denote by $x_{i+1/2,j}$ the midpoint of the interval $[x_{i,j}, x_{i+1,j}]$ (that is, in the x-direction) and by $y_{i,j+1/2}$ the midpoint of $[y_{i,j}, y_{i,j+1}]$. The generalization of the 2D method is straightforward from the 1D case and it is simple to implement since it allows a dimension by dimension application of the numerical method. For all the interior points where there are no jumps in x, we use centered differences to approximate $\partial\psi/\partial x$ at the midpoint:

$$\psi'_{i+1/2,j} = \frac{\psi_{i+1,j} - \psi_{i,j}}{\Delta x}. \tag{12}$$

Similarly, for all the interior points where there are no jumps in y, we discretize $\partial\psi/\partial y$:

$$\psi'_{i,j+1/2} = \frac{\psi_{i,j+1} - \psi_{i,j}}{\Delta y}. \tag{13}$$

For any points where there is a jump in x to be incorporated in the discretization, we approximate $\partial\psi/\partial x$ at $x = x_{i+1/2,j}$ by fitting a quadratic polynomial through the values $\psi_{I,j}$, $\psi_{i+1,j}$ and $\psi_{i+2,j}$, and then evaluating its slope at $x = x_{i+1/2,j}$. Similarly, for any points where there is a jump in y, we need to approximate $\partial\psi/\partial y$ at $y = y_{i,j+1/2}$ by fitting a quadratic polynomial through the values $\psi_{i,J}$, $\psi_{i,j+1}$ and $\psi_{i,j+2}$, and evaluating its slope at $y = y_{i,j+1/2}$.

3   Numerical result

In this section, we present four simple problems to demonstrate the algorithm. The first problem is the 1D Poisson equation $\psi_{xx} = \exp(x)$. Figure 2 shows that the analytic expression for the error is very accurate (left) and the algorithm is second-order accurate with the RMS and maximum errors
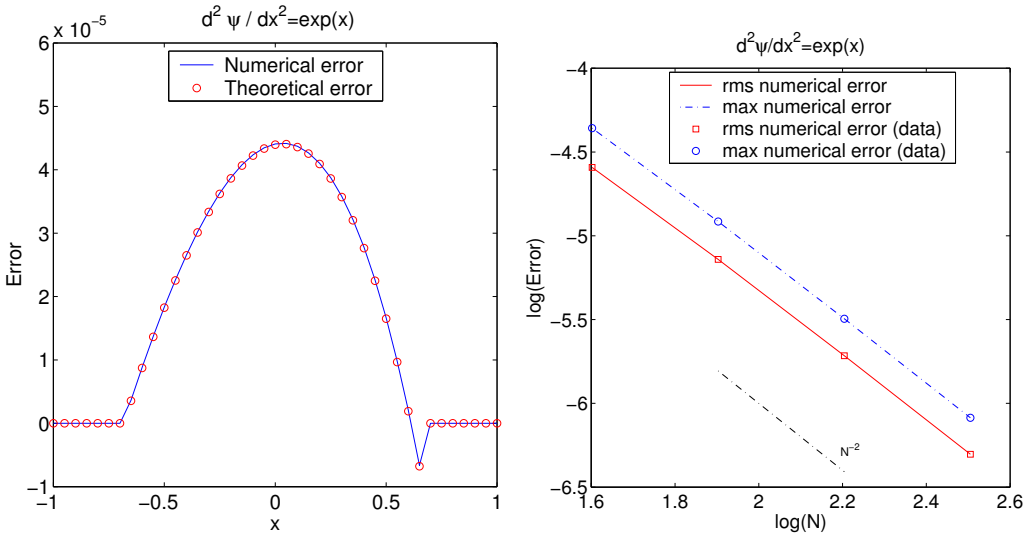
FIGURE 2: Plot comparing numerical and theoretical error (left) and plot of the RMS and maximum errors (right) for the 1D Poisson equation with $f(x) = \exp(x)$.
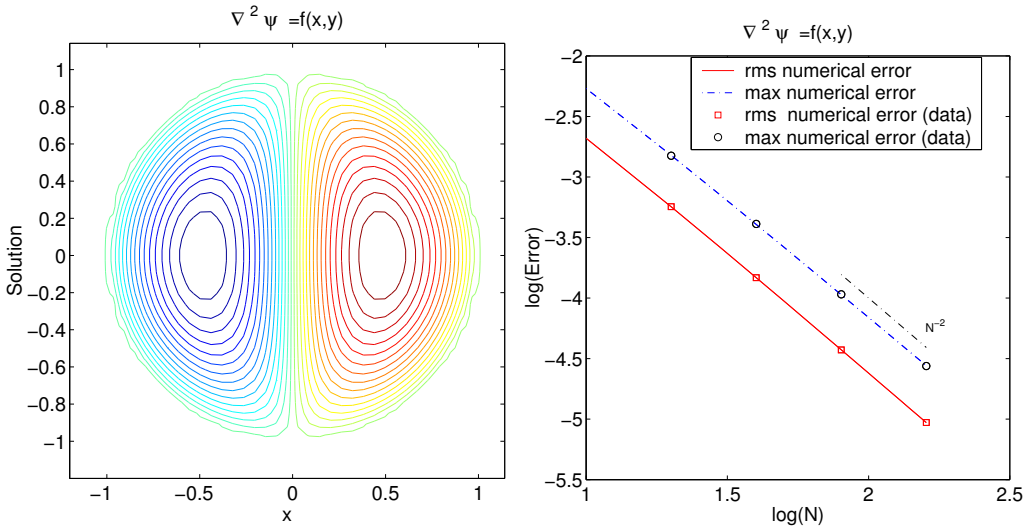
FIGURE 3: Plot of the numerical solution (left) and plot of the RMS and maximum errors (right) for the Poisson equation on the interior of a circle of radius one with $f(x, y) = \text{sgn}(x)$ and with zero Dirichlet boundary condition.
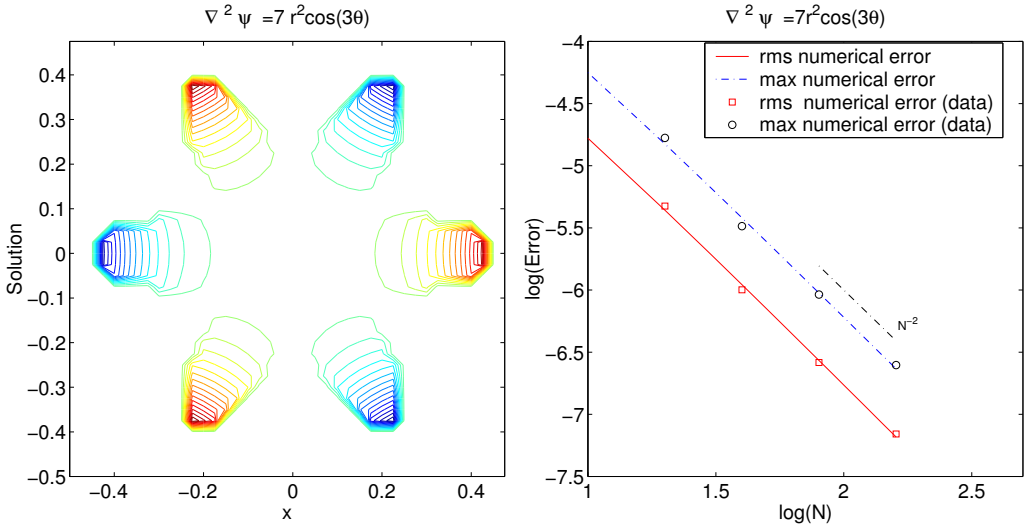
FIGURE 4: Plot of numerical solution (left) and plot of the RMS and maximum errors (right) for the Poisson equation on $\Omega = \{(r,\theta); r \leq 0.30 + 0.15\cos 6\theta\}$ with $f(r,\theta) = 7r^2\cos 3\theta$ on the interior of the domain and non-zero Dirichlet boundary conditions chosen to ensure continuity with the interior solution, that is, $(\psi(r,\theta) = r^4\cos 3\theta)$

converging as $\mathcal{O}((\Delta x)^2)$. The second test problem is the 2D Poisson equation (1) with Ω the interior of the unit circle and zero Dirichlet boundary condition on $\partial\Omega$. We take $f(x,y) = \mathrm{sgn}(x)$. Figure 3 shows the numerical solution (left) and the RMS and maximum errors (right) demonstrating the second-order accuracy of the method. The third problem is the 2D Poisson equation $\psi_{xx} + \psi_{yy} = 7r^2\cos 3\theta$ on $\Omega = \{(r,\theta); r \leq 0.30 + 0.15\cos 6\theta\}$ and non-zero Dirichlet boundary condition on $\partial\Omega$, consistent with the given form of the Poisson equation. Figure 4 shows the numerical solution (left) and the RMS and maximum errors (right) which again shows second-order accuracy.

The final problem is the 2D Poisson equation $\nabla^2\psi = -2\cos(x+y)$ on the unit circle, with Dirichlet boundary condition chosen to ensure continuity
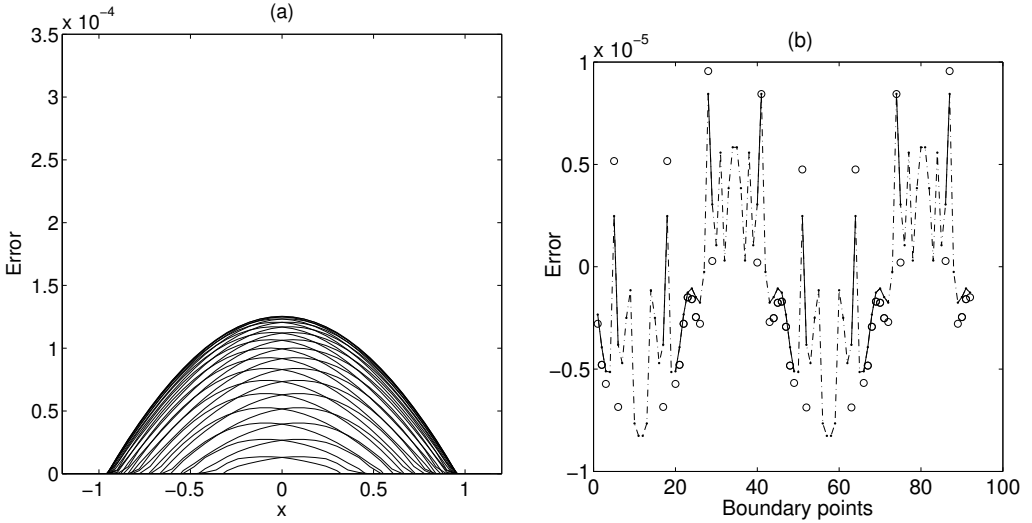
FIGURE 5: The 2D problem $\nabla^2 \psi = -2\cos(x+y)$, with $N\Delta x = 2.4$ and $N = 40$ (panel (a) and (b)), subject to $\psi = \cos(x+y)$ on the boundary of the unit circle. In (a) plots of error at constant values of y. In (b) the actual numerical errors are shown at all points on the boundary, these are taken as running clockwise from $(x, y) = (-1, 0)$ (labelled 1–96) and the errors are shown as a dash-dotted curve, where there are sequences of adjacent non-corner points, the dash-dotted line for the numerical error is replaced with a solid line. For non-corner points, the 1D full quadratic error of (10) for the x-direction and the equivalent in the y-direction are shown as open circle.

from within, that is: $\psi = \cos(x + y)$. Outside the unit circle we set $\psi = 0$. In the first panel 5(a) the numerical error for the whole domain is plotted. In panel (b) we show the boundary error (with $N = 40$, so there are 96 boundary points). As we only have available the 1D expressions, we can only make comparison at points on the boundary which essentially use 1D contribution (non-corner points). Although the agreement is not exact, we see a quite reasonable estimation of the errors at the boundaries.

4   Conclusion

In this paper, we present a finite-difference method for the solution of the Poisson equation. For the 1D case we have determined explicit forms for the error involved, following the idea of Johansen and Colella [2]. In agreement with them [2] we find that the end-point error is $\mathcal{O}((\Delta x)^3)$ and so has negligible influence, with the total error being dominated by the internal error. Although no expression for error have been derived for the 2D case, we have found that for boundary points that are not corner points (that is, the quadratic fitting contribution at the boundary point is only one directional for non-corner point), the 1D error expressions, in either the x- or y-directions as appropriate, give good estimates for the 2D boundary error. If local grid refinement is to be used together with our method then issues like matching of the solution at grid interfaces will need to be considered (for more details, see [6]).

References

[1] Frederic Gibou, Ronald P. Fedkiw, Li-Tien Cheng and Myungjoo Kang, A second-order accurate symmetric discretization of the Poisson

equation on irregular domains, *J. Comp. Phys.* 176, (2002) 205–227. C715

[2] H. Johansen and P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, *J. Comp. Phys.* 147, (1998) 60–85. C715, C719, C726

[3] R. J. LeVeque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.* 31, 1019 (1994). C715

[4] X. Liu, R. Fedkiw and M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, *J. Comp. Phys.* 160, (2000) 151–178. C715

[5] C. Macaskill, W.E.P. Padden and D.G. Dritschel, The CASL algorithm for quasi-geostrophic flow in a cylinder, *J. Comp. Phys.* 188/1, (2003) 232–251. C714

[6] D. Martin and K. Cartwright, Solving Poisson's equation using adaptive mesh refinement, Electronics Research Laboratory Memorandum UCB/ERL M96/66, University of California, Berkeley, October 1996. C726

[7] A. Mayo, The fast solution of Poisson's and the biharmonic equations in irregular domains, *SIAM J. Numer. Anal.* 21(2), 285 (1984). C715

[8] A. Mayo, The rapid evaluation of volume integrals of potential theory on general regions, *J. Comp. Phys.* 100, 236 (1992) C715

[9] A. Mayo and A. Greenbaum, Fast parallel iterative solution of Poisson's and the biharmonic equations on irregular regions, *SIAM J. Sci. Stat. Comput.* 13(1), 101 (1992) C715

[10] A. McKenney, L. Greengard, and A. Mayo, A fast Poisson solver for complex geometries, *J. Comp. Phys.* 118, 348 (1995). C715

[11] C. S. Peskin and B. F. Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries, *J. Comp. Phys.* 105, 33 (1993). C715

[12] C. Peskin, Numerical analysis of blood flow in the heart, *J. Comp. Phys.* 25, 220 (1977). C715