A Supra-Convergent Finite Difference Scheme for the Poisson and Heat Equations on Irregular Domains and Non-Graded Adaptive Grids

Han Chen * Chohong Min † Frédéric Gibou ‡

21st March 2006

Abstract

We present a finite difference scheme for solving the variable coefficient Poisson and heat equations on irregular domains with Dirichlet boundary conditions. We consider non-graded Cartesian grids, i.e. grids for which the difference in size between two adjacent cells is not constrained. We sample the solutions at the cell vertices (nodes) and use a quadtree data structure as an efficient means to represent the grids. The boundary of the irregular domain is represented by the zero value points of a level set function. For cells cut by the interface, both the interface's location and the value of the solution at the interface are found by quadratic interpolation. In the case of the heat equation, we use a second order implicit discretization in time to avoid the stringent time step restrictions associated with explicit schemes. This discretization can be applied in a dimension by dimension framework, producing a scheme that is straightforward to implement. Numerical results in two spatial dimensions demonstrate second order accuracy for both the solution and its gradient in the L^1 and L^∞ norms.

1 Introduction

The Poisson and the heat equations are two of the fundamental equations used in the modeling of diffusion dominated phenomena, ranging from electromagnetism to semi-conductor growth to fluid mechanics. A wide variety of approaches exist for solving the Poisson equation on irregular domains. In [23] Peskin introduced the immersed boundary method, which uses a δ-function that smears out the solution on a thin finite band around the interface. Leveque and Li introduced the immersed interface method [13], which is a second order accurate numerical method designed to preserve the jump condition at the interface. In [18, 20], Mayo *et al.* proposed a fast method using boundary integral techniques. Based on the Ghost Fluid Method of Fedkiw *et al.* [7], Liu *et al.* [15] developed 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 its derivative may have jumps. Gibou *et al.* [9] introduced a second order accurate symmetric discretization in the case where Dirichlet boundary conditions are imposed on the irregular domain. This work is based on using linear extrapolation to define ghost values for the solution across the interface. Third and fourth order accurate schemes were later proposed in Gibou *et al.* [8] by defining ghost values with quadratic or cubic extrapolations. In this case, the linear system becomes nonsymmetric. Jomaa and Macaskill [12] showed that in the case where ghost values are defined by linear extrapolations, the error near the boundary is in general large compared to the error found for regular domains. The also pointed out that defining ghost values by a quadratic extrapolation leads to errors comparable with those obtained for a regular domain.

Many physical problems have different scales and more often than not, only small portions of the computational domain require fine resolution. Uniform grids become inefficient in this case in terms of memory storage and CPU usage. Adaptive mesh strategies for elliptic partial differential equations, such as the

*Computer Science Department, University of California, Santa Barbara, CA 93106.
†Mathematics Department, University of California, Santa Barbara, CA 93106.
‡Mechanical Engineering Department & Computer Science Department, University of California, Santa Barbara, CA 93106.

Poisson equations, are often associated with the finite element method (see e.g. [6, 11]). Young *et al.* [30] introduced a finite element method employing adaptive mesh refinements for second order variable coefficient elliptic equations using a cut-cell representation of irregular domains. The finite element method has the advantage of a rigorous theoretical framework and a vast number of optimized commercial implementations. However, two factors that must be considered are the adaptive mesh generation for complicated domains and the efficiency of the organization of the resulting data structure. Generally, when applying the finite element method to moving boundary problems, one must take great care that the mesh generated is of good quality everywhere (see e.g. [28]).

Johansen and Colella [10] presented a cell-centered finite volume method for solving the variable coefficient Poisson equation on irregular domains using a multigrid approach and a block-grid algorithm related to the adaptive mesh refinement scheme of Berger and Oliger [5]. McCorquodale *et al.* [19] presented a node-centered finite difference approach for solving the variable coefficient Poisson equation on irregular domains using the block-structured adaptive mesh refinement and the multigrid solver of Almgren [2, 4, 3]. However, these patch based adaptive mesh refinement techniques restrict the adaptivity to block structured meshes, while a more efficient quadtree structure could be used as pointed out in Aftomis *et al.* [1].

Quadtree and octree (see e.g. [26, 27] for an introduction on quadtree/octree data structures) spatial discretizations have been used in a variety of approaches for solving the Poisson equation. Popinet [24] proposed a second order nonsymmetric numerical method to study the incompressible Navier-Stokes equations using an octree data structure for representing the spatial discretization. In this method, a Poisson equation for the pressure is solved to account for the incompressibility condition using a standard projection method. Only graded trees, i.e. trees for which the ratio between two adjacent cell sizes cannot exceed two, were considered in [24]. Non-graded octrees have been proposed in Losasso *et al.* [17] to obtain a first order accurate symmetric discretization of the Poisson equation. This work was then extended to second order accuracy in [16] using the work of Lipnikov [14]. Recently, Min *et al.* [22] introduced a supra-convergent scheme for solving the variable coefficient Poisson equation on a rectangular domain using non-graded grids represented by quadtrees/octrees. A hallmark of this method is that the solution as well as its gradients are second order accurate.

Based on the work of Min *et al.* [22], we consider in this paper the solution of the variable coefficient Poisson and the heat equations on irregular domains with Dirichlet boundary conditions. We embed the region of interest in a rectangular domain and consider non-graded Cartesian grids. The solution is sampled at the vertices (nodes) of each cell. At internal regular nodes, the Poisson or the heat equations are discretized with the standard second order accurate five-point finite difference scheme. At internal T-junction nodes, the value at the missing direct neighbor is defined by a linear interpolation of the adjacent values in the transverse direction. The spurious error induced by this interpolation is then properly balanced by a weighted combination of the discretizations in all Cartesian directions (see [21] for more details). For nodes adjacent to the interface, the interface's location as well as the Dirichlet boundary value at the interface are found by quadratic interpolations. The resulting linear system is nonsymmetric and solved using the stabilized bi-conjugate gradient method with the incomplete LU preconditioner [25]. In the case of the heat equation, the Crank-Nicolson scheme is used with a time step of $\Delta t = c\Delta x$, where $0 < c < 1$ and Δx is the size of the smallest cell. Numerical examples demonstrate that this method produces second order accuracy for the solution and its gradients in the L^1 and L^∞. This second order accuracy is particularly important in some applications (e.g. Stefan type problems), where the accuracy of the solution gradients determine the overall accuracy of the computation.

2 Equations

2.1 Poisson Equation

Consider a Cartesian computational domain, Ω, with exterior boundary, $\partial\Omega$, and a lower dimensional interface, Γ, which divides the computational domain into disjoint pieces, the interior region Ω^- and the exterior region Ω^+ (see figure 1). The regions are represented by a level set function ϕ, taken to be the signed distance function to the interface Γ. That is, Ω^- is represented by the set of points where $\phi < 0$ and Ω^+ is represented by the set of points where $\phi > 0$, whereas Γ is represented by the set of points where $\phi = 0$ The

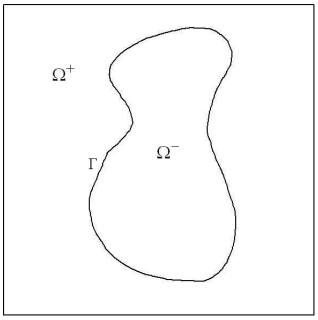

Figure 1: Schematic of two distinct regions Ω^- and Ω^+, separated by an interface Γ. The solution may present a kink at the interface.

variable coefficient Poisson equation is written as

$$\nabla \cdot (\rho(\vec{x})\nabla u(\vec{x})) = f(\vec{x}), \qquad \vec{x} \in \Omega, \tag{1}$$

where $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ is the gradient operator and where $\rho(\vec{x})$ is assumed to be continuous on each of the disjoint subdomains, Ω^- and Ω^+, but may be discontinuous across the interface Γ. Furthermore, $\rho(\vec{x})$ is assumed to be bounded below by a positive constant. On $\partial\Omega$, either Dirichlet or Neumann boundary conditions are specified. On the interface Γ, a Dirichlet boundary condition of $u_\Gamma = g(\vec{x})$ is specified. Thus, equation (1) decouples into two distinct equations, one on Ω^- and one on Ω^+, and the solutions can be obtained independently.

2.2 Heat Equation

Ignoring the effects of convection, the standard heat equation reads

$$\rho c_v T_t = \nabla \cdot (k\nabla T), \tag{2}$$

where T is the temperature, ρ is the density, c_v the specific heat at constant volume and k is the thermal conductivity. Assuming that ρ and c_v are constant allows equation (2) to be written as

$$T_t = \nabla \cdot (\hat{k}\nabla T), \tag{3}$$

where $\hat{k} = \frac{k}{\rho c_v}$.

3 Spatial Discretization on a Quadtree

The domain is discretized into squares and a quadtree data structure is used to represent this discretization. As depicted in figure 2 for a two dimensional domain, the entire domain is originally associated with the root of the tree, which has a level of zero by definition. Then this discretization proceeds recursively, i.e. each cell can be in turn split into four children which have one more level than their parent cell. A cell with no children is called a leaf. Two cells are called neighbors if they share a common face or part of a face. The interested reader is referred to [26, 27] for more on quadtree and octree data structures.

Due to the irregularity of the interface and the fact that large solution errors are often observed near the interface, we discretize the computational domain in such a way that the cell size is proportional to the absolute value of the level set function ϕ, i.e., the distance to the interface. We split a cell c if

$$\min_{v \in V} |\phi(v)| < \frac{\text{lip} * \text{diag}}{2}, \tag{4}$$

where v is a vertex of the current cell c, V is the set of all vertices of cell c, lip is the Lipschitz constant associated with ϕ, and diag is the diagonal length of the current cell (see Strain [29]). We impose that the finest resolution is obtained at cells cut by the interface.

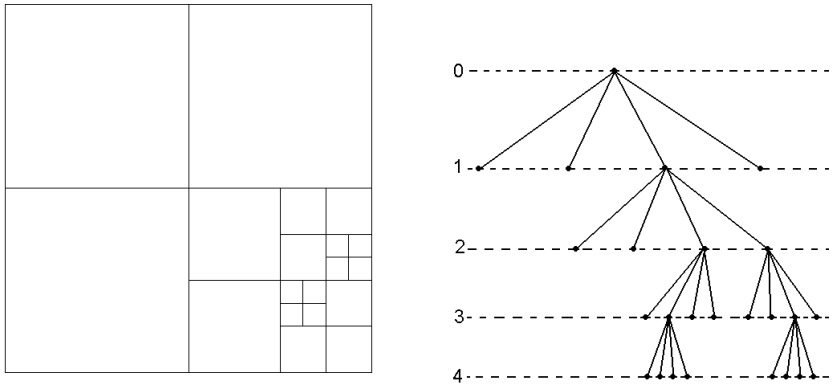

Figure 2: Discretization of a two dimensional domain (left) and its quadtree representation (right). The entire domain corresponds to the root of the tree (level 0), and each cell is subdivided into four children, in the order of lower-left, upper-left, lower-right, upper-right. In this example, the tree is not graded since the difference of levels between neighboring cells exceeds one.

By definition, a quadtree is graded if the difference between two adjacent cell levels is at most one. In this paper, we sample the solution at the cell vertices and we consider non-graded Cartesian grids,i.e. grids for which the level difference between two adjacent cells is not constrained. Typical spatial discretizations are presented in section 5.

4 Finite Difference Schemes

4.1 Variable Coefficient Poisson Equation on Regular Domains

We first recall the discretization from Min *et al.* [22] for the variable coefficient Poisson equation on regular domains. In two spatial dimensions, let us consider the finite difference discretization for a T-junction node without a direct right neighboring node as depicted in (figure 3). We note that discretization at T-junction nodes without a direct left or top or bottom adjacent node are derived in a similar manner.

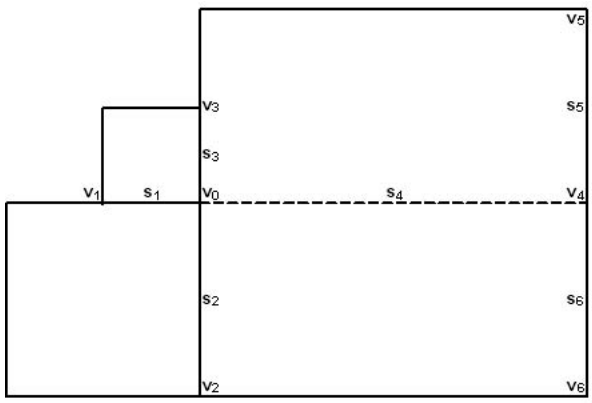

Figure 3: A configuration illustrating the nodes involved in the discretization at a T-junction node v_0.

The discretizations for $(\rho u_x)_x$ and $(\rho u_y)_y$ at node v_0, along with their Taylor analysis, are given by

$$\left(\frac{u_1 - u_0}{s_1} \cdot \frac{\rho_1 + \rho_0}{2} + \frac{s_6 D_5 + s_5 D_6}{s_5 + s_6}\right) \cdot \frac{2}{s_1 + s_4} = (\rho u_x)_x + \frac{s_5 s_6}{(s_1 + s_4)s_4} (\rho u_y)_y + O(h), \tag{5}$$

and

$$\left(\frac{u_2-u_0}{s_2}\cdot\frac{\rho_2+\rho_0}{2}+\frac{u_3-u_0}{s_3}\cdot\frac{\rho_3+\rho_0}{2}\right)\cdot\frac{2}{s_2+s_3}=(\rho u_y)_y+O(h), \tag{6}$$

respectively, where

$$\begin{aligned} D_5 &= \tfrac{u_5-u_0}{s_4}\cdot\tfrac{\rho_5+\rho_0}{2}, \\ D_6 &= \tfrac{u_6-u_0}{s_4}\cdot\tfrac{\rho_6+\rho_0}{2}. \end{aligned}$$

The spurious term $\frac{s_5 s_6}{(s_1+s_4)s_4}(\rho u_y)_y$ is cancelled by appropriately weighting equations (5) and (6) as:

$$\begin{aligned} &\left(\tfrac{u_1-u_0}{s_1}\cdot\tfrac{\rho_1+\rho_0}{2}+\tfrac{s_6 D_5+s_5 D_6}{s_5+s_6}\right)\cdot\tfrac{2}{s_1+s_4} &&+ \\ &\left(\tfrac{u_2-u_0}{s_2}\cdot\tfrac{\rho_2+\rho_0}{2}+\tfrac{u_3-u_0}{s_3}\cdot\tfrac{\rho_3+\rho_0}{2}\right)\cdot\tfrac{2}{s_2+s_3}\cdot\left(1-\tfrac{s_5 s_6}{(s_1+s_4)s_4}\right) &&= f_0+O(h). \end{aligned} \tag{7}$$

Not surprisingly, this weighted scheme (7) reduces to the usual five point scheme in the case of a regular node (figure 4):

$$\begin{aligned} &\left(\tfrac{u_1-u_0}{s_1}\cdot\tfrac{\rho_1+\rho_0}{2}+\tfrac{u_4-u_0}{s_4}\cdot\tfrac{\rho_4+\rho_0}{2}\right)\cdot\tfrac{2}{s_1+s_4} &&+ \\ &\left(\tfrac{u_2-u_0}{s_2}\cdot\tfrac{\rho_2+\rho_0}{2}+\tfrac{u_3-u_0}{s_3}\cdot\tfrac{\rho_3+\rho_0}{2}\right)\cdot\tfrac{2}{s_2+s_3} &&= f_0+O(h). \end{aligned} \tag{8}$$

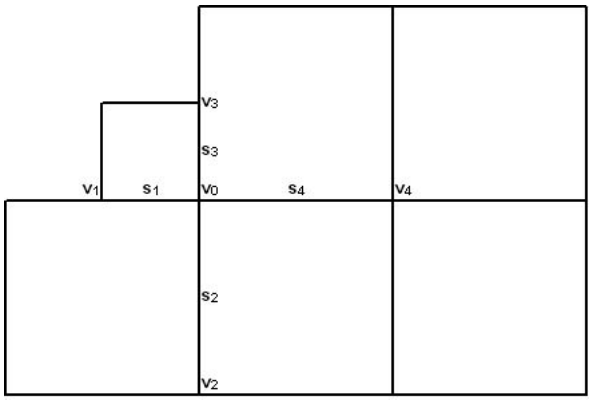

Figure 4: A configuration illustrating the nodes involved in the discretization at a regular node v_0.

In the case of the scheme (7), every coefficient is multiplied equally to u_0 and its neighbors and the absolute value of the sum of the coefficients in front of u_o is equal to the sum of those in front of all its neighboring nodes. When the interface is located between u_0 and one or more of its neighbors, the corresponding coefficient(s) in front of the neighbor node(s) is zero. This results in a diagonally dominant linear system to solve.

4.2 Heat Equation

We consider only the two dimensional constant coefficient heat equation:

$$T_t = k\Delta T, \tag{9}$$

noting that extension to the variable coefficient heat equation is straightforward. Explicit schemes impose a time step restriction of $\Delta t \leq \frac{(\theta\Delta x)^2}{2k}$ for stability, where Δx is the size of the smallest cell and θ is the smallest cell fraction for cells cut by the interface. Since θ can be arbitrarily small, so is the time step restriction Δt, which leads to unpractical schemes. Implicit schemes avoid such time step restrictions. We use the Crank-Nicholson scheme to discretize the heat equation (9):

$$(I-\frac{1}{2}\,\Delta t\; k\;\Delta)T^{n+1}=(I+\frac{1}{2}\,\Delta t\; k\;\Delta)T^n, \tag{10}$$

where I is the identity matrix and T^n is the solution at the n^{th} time step. The spatial discretization of the Laplace operator in Section 4.1, is used to approximate ΔT. The Crank-Nicolson scheme is unconditionally stable and second order accurate. The time step Δt can be chosen proportional to Δx.

4.3 Discretization Near the Interface

When one or more of the neighboring nodes involved in the spatial discretization, e.g. v_1, v_2, v_3, v_5, v_6 in figure 3, are outside the region with a positive ϕ value, the interface lies between the center node v_0 and the neighboring node(s). Since we exclude the cases that the interface crosses a T-junction by imposing that the smallest cells lie on the interface (see figures 6, 8, 10, 12, 14 and 16), we only consider the case when the node next to the interface is a regular node, as depicted in figure 5.

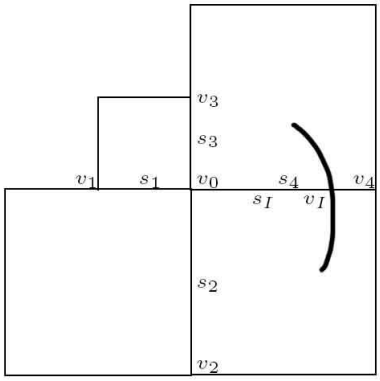

Figure 5: One neighboring node is outside the region and the center node is a regular node.

Suppose the interface intersects the segment $[v_0, v_4]$ at v_I. Denote by s_I the distance between v_I and v_0. s_I is approximated by finding the zero crossing of the quadratic interpolation in ϕ:

$$s_I = \begin{cases} \frac{-\phi_x + \sqrt{\phi_x^2 - 2\phi_{xx}\phi_0}}{\phi_{xx}} & \text{if } \phi_{xx} > \epsilon, \\ \frac{-\phi_x - \sqrt{\phi_x^2 - 2\phi_{xx}\phi_0}}{\phi_{xx}} & \text{if } \phi_{xx} < -\epsilon, \\ -\frac{\phi_0}{\phi_x} & \text{if } |\phi_{xx}| \leq \epsilon, \end{cases} \tag{11}$$

where ϵ is a small positive number to avoid division by zero, and ϕ_x and ϕ_{xx} are approximated at v_0 using second order central difference schemes:

$$\phi_x = \frac{s_1 \frac{\phi_4 - \phi_0}{s_4} + s_4 \frac{\phi_0 - \phi_1}{s_1}}{s_1 + s_4} \tag{12}$$

and

$$\phi_{xx} = \left(\frac{\phi_4 - \phi_0}{s_4} - \frac{\phi_0 - \phi_1}{s_1} \right) \frac{2}{s_1 + s_4}. \tag{13}$$

Then the standard five-point scheme is used to approximate $(\rho u_x)_x$ as in section 4.1:

$$(\rho u_x)_x = \left(\frac{u_I - u_0}{s_I} \cdot \frac{\rho_I + \rho_0}{2} - \frac{u_0 - u_1}{s_1} \cdot \frac{\rho_1 + \rho_0}{2} \right) \cdot \frac{2}{s_I + s_1}, \tag{14}$$

where the values of u_I and ρ_I are given by the Dirichlet boundary condition. Similarly, the discretizations at nodes with the interface crossing in different directions ($\mathbf{x}$, $-\mathbf{x}$, $\mathbf{y}$, $-\mathbf{y}$, $\mathbf{z}$ and $-\mathbf{z}$) can be performed independently, which makes the method straightforward to implement in a dimension by dimension framework.

4.4 Computing Second Order Accurate Gradients

To compute the gradients to second order accuracy at a T-junction node, an intermediate value (u_4 in figure 3) is linearly interpolated as:

$$u_4 = \frac{s_5 u_6 + s_6 u_5}{s_5 + s_6}. \tag{15}$$

The spurious error caused by the interpolation is successfully removed and the gradients are computed as:

$$\begin{aligned} u_x &= \frac{u_4 - u_0}{s_4} \cdot \frac{s_1}{s_1 + s_4} + \frac{u_0 - u_1}{s_1} \cdot \frac{s_4}{s_1 + s_4} - \frac{s_5 s_6 s_1}{2s_4(s_1 + s_4)} u_{yy}, \\ u_y &= \frac{u_3 - u_0}{s_3} \cdot \frac{s_2}{s_2 + s_3} + \frac{u_0 - u_2}{s_2} \cdot \frac{s_3}{s_2 + s_3}, \end{aligned} \tag{16}$$

where

$$u_{yy} = \left(\frac{u_3 - u_0}{s_3} + \frac{u_2 - u_0}{s_2} \right) \cdot \frac{2}{s_2 + s_3}.$$

When the node is a regular node (e.g., v_0 in figure 4), equation (16) reduces to the usual weighted average of the forward and backward differences:

$$\begin{aligned} u_x &= \frac{u_4 - u_0}{s_4} \cdot \frac{s_1}{s_1 + s_4} + \frac{u_0 - u_1}{s_1} \cdot \frac{s_4}{s_1 + s_4}, \\ u_y &= \frac{u_3 - u_0}{s_3} \cdot \frac{s_2}{s_2 + s_3} + \frac{u_0 - u_2}{s_2} \cdot \frac{s_3}{s_2 + s_3}, \end{aligned} \tag{17}$$

For nodes next to the interface (e.g., v_0 in figure 5), the following formula is used:

$$\begin{aligned} u_x &= \frac{u_I - u_0}{s_I} \cdot \frac{s_1}{s_1 + s_4} + \frac{u_0 - u_1}{s_1} \cdot \frac{s_I}{s_1 + s_I}, \\ u_y &= \frac{u_3 - u_0}{s_3} \cdot \frac{s_2}{s_2 + s_3} + \frac{u_0 - u_2}{s_2} \cdot \frac{s_3}{s_2 + s_3}, \end{aligned} \tag{18}$$

We note that the calculation of the gradient involves the same cells as those used in the discretization of the Poisson or heat equation, hence preserving the locality and ease of implementation of the method.

5 Examples

In this section we present numerical evidence that confirms the schemes described in this paper produce second order accuracy in the L^1 and L^∞ norms for both the solution and its gradients. In particular the difference of level between adjacent cells can be greater than one, illustrating the fact that the method is supra-convergent on non-graded adaptive grids. The linear systems of equations are solved using the stabilized bi-conjugate gradient method with an incomplete LU preconditioner (see e.g. [25]). We present the order of accuracy in the L^1 and L^∞ norms. In this paper, we solve the poisson or the heat equations in Ω^- only, noting that the procedure to obtain the solution on $\Omega+$ is similar.

5.1 Poisson Equation

We consider equation (1) in two spatial dimensions, where ρ is piecewise constant on each subdomain or spatially varying on each subdomain. Dirichlet boundary conditions are used on the interface and on the domain boundaries.

5.1.1 Example 1

Consider $\bigtriangledown \cdot (\rho \bigtriangledown u) = f$ on $\Omega = [-1, 1] \times [-1, 1]$ with an exact solution of $u = e^{-x^2 - y^2}$, where $\rho = 1$. The interface is given by the set of points where $\phi = \sqrt{x^2 + (y-1)^2} - 1.5 = 0$. The non-graded Cartesian grid and the interface are illustrated in figure 6. Numerical accuracy test results for the solution and its gradient are given in table 1 and table 2, respectively. The numerical solution on a grid with an effective resolution of 128×128 is plotted in figure 7.

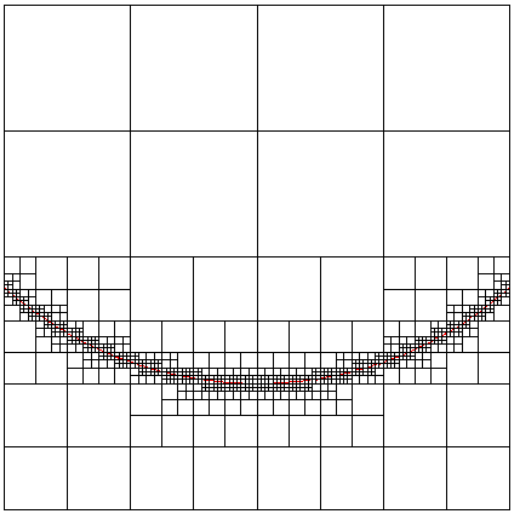

Figure 6: Domain $\Omega = [-1, 1] \times [-1, 1]$ and the spatial discretization used in example 5.1.1 and example 5.2.1.

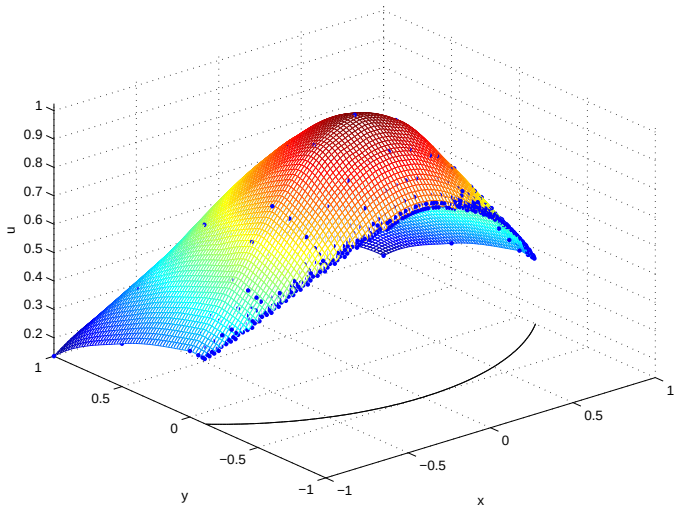

Figure 7: Graph of the solution in example 5.1.1. The dots represent the approximate solution and the mesh represents the exact solution.

effective resolution	L^∞ error	order	L^1 error	order
128×128	2.346×10^{-2}	—	1.061×10^{-3}	—
256×256	8.040×10^{-3}	1.545	2.584×10^{-4}	2.038
512×512	2.046×10^{-3}	1.974	6.557×10^{-5}	1.978
1024×1024	5.188×10^{-4}	1.980	1.658×10^{-5}	1.984

Table 1: Accuracy results for the solution u in example 5.1.1.

effective resolution	L^∞ error	order	L^1 error	order
128×128	3.466×10^{-1}	—	2.289×10^{-2}	—
256×256	6.687×10^{-2}	2.374	5.310×10^{-3}	2.108
512×512	1.341×10^{-2}	2.317	1.323×10^{-3}	2.005
1024×1024	2.919×10^{-3}	2.200	3.338×10^{-4}	1.987

Table 2: Accuracy results for the solution gradient ∇u in example 5.1.1.

5.1.2 Example 2

Consider $\bigtriangledown \cdot (\rho \bigtriangledown u) = f$ on $\Omega = [-1, 1] \times [-1, 1]$ with an exact solution of $u = \sin(x)\cos(y)$, where $\rho = e^{xy}$. The interface is a circle given by the set of points where $\phi = \sqrt{x^2 + y^2} - 0.75 = 0$. The non-graded Cartesian grid and the interface are illustrated in figure 8. Numerical accuracy test results for the solution and its gradient are given in table 3 and table 4, respectively. The numerical solution on a grid with an effective resolution of 128×128 is plotted in figure 9.

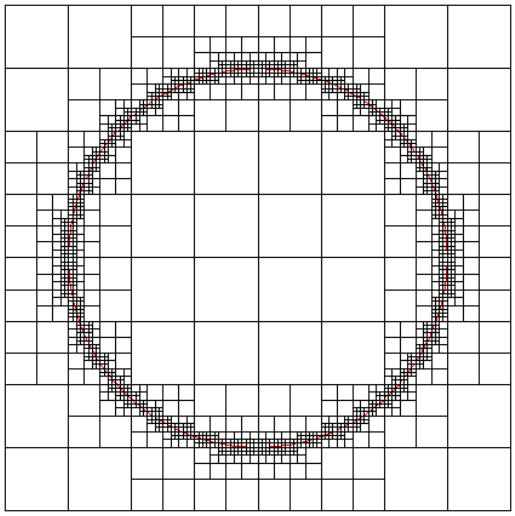

Figure 8: Domain $\Omega = [-1, 1] \times [-1, 1]$ and the spatial discretization used in example 5.1.2 and example 5.2.2.

effective resolution	L^∞ error	order	L^1 error	order
128×128	1.692×10^{-3}	—	1.590×10^{-4}	—
256×256	4.068×10^{-4}	2.056	3.599×10^{-5}	2.144
512×512	9.922×10^{-5}	2.036	8.502×10^{-6}	2.082
1024×1024	2.450×10^{-5}	2.018	2.069×10^{-6}	2.039

Table 3: Accuracy results for the solution u in example 5.1.2.

5.1.3 Example 3

Consider $\bigtriangledown \cdot (\rho \bigtriangledown u) = f$ on $\Omega = [-1, 1] \times [-1, 1]$ with an exact solution of $u = \sin(\pi x)\sin(\pi y)$, where $\rho = \sin(xy) + 2$. The interface is an ellipse given by the set of points where $\phi = x^2 + 4y^2 - 0.25 = 0$. The non-graded Cartesian grid and the interface are illustrated in figure 10. Numerical accuracy test results for

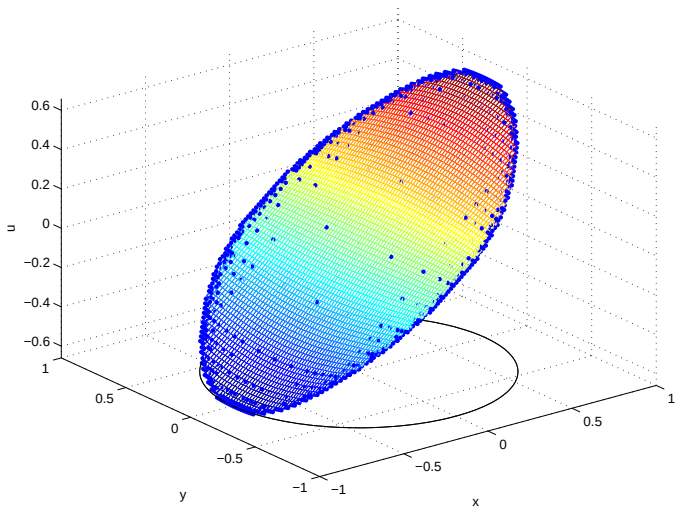

Figure 9: Graph of the solution in example 5.1.2. The dots represent the approximate solution and the mesh represents the exact solution.

effective resolution	L^∞ error	order	L^1 error	order
128×128	1.094×10^{-2}	—	4.422×10^{-3}	—
256×256	3.018×10^{-3}	1.858	1.085×10^{-3}	2.027
512×512	8.301×10^{-4}	1.862	2.678×10^{-4}	2.019
1024×1024	2.293×10^{-4}	1.856	6.647×10^{-5}	2.010

Table 4: Accuracy results for the solution gradient ∇u in example 5.1.2.

the solution and its gradient are given in table 5 and table 5, respectively. The numerical solution on a grid with an effective resolution of 128×128 is plotted in figure 11.

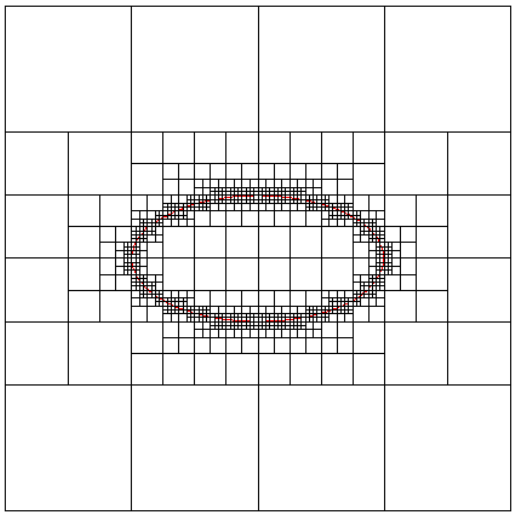

Figure 10: Domain $\Omega = [-1, 1] \times [-1, 1]$ and the spatial discretization used in example 5.1.3.

effective resolution	L^∞ error	order	L^1 error	order
128×128	3.408×10^{-3}	—	6.832×10^{-4}	—
256×256	9.025×10^{-4}	1.917	1.626×10^{-4}	2.071
512×512	2.253×10^{-4}	2.002	3.949×10^{-5}	2.042
1024×1024	5.615×10^{-5}	2.005	9.685×10^{-6}	2.028

Table 5: Accuracy results for the solution u in example 5.1.3.

effective resolution	L^∞ error	order	L^1 error	order
128×128	6.672×10^{-2}	—	2.681×10^{-2}	—
256×256	1.961×10^{-2}	1.766	6.808×10^{-3}	1.977
512×512	5.454×10^{-3}	1.846	1.714×10^{-3}	1.990
1024×1024	1.498×10^{-3}	1.864	4.302×10^{-4}	1.994

Table 6: Accuracy results for the solution gradient ∇u in example 5.1.3.

5.1.4 Example 4

Consider $\bigtriangledown \cdot (\rho \bigtriangledown u) = f$ on $\Omega = [-1, 1] \times [-1, 1]$ with an exact solution of $u = e^{xy}$, where $\rho = x^2 + y^2$. The interface is diamond shaped, given by the set of points where $\phi = |y| + 1.5|x| - 0.75 = 0$. The non-graded Cartesian grid and the interface are illustrated in figure 12. Numerical accuracy test results for the solution and its gradient are given in table 7 and table 8, respectively. The numerical solution on a grid with an effective resolution of 128×128 is plotted in figure 13.

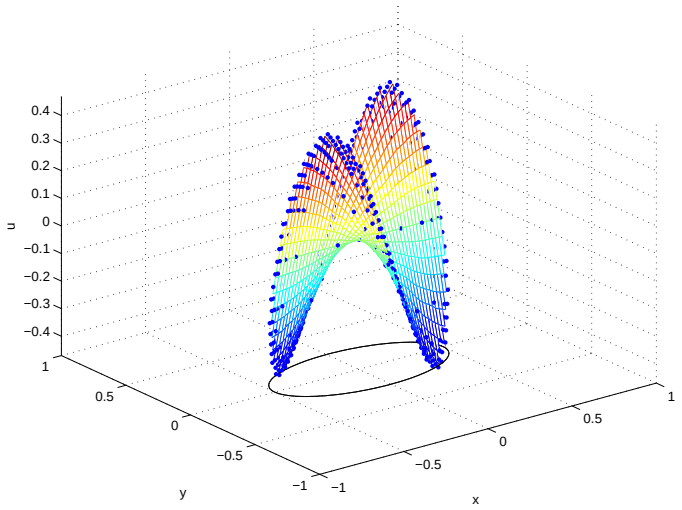

Figure 11: Graph of the solution in example 5.1.3. The dots represent the approximate solution and the mesh represents the exact solution.

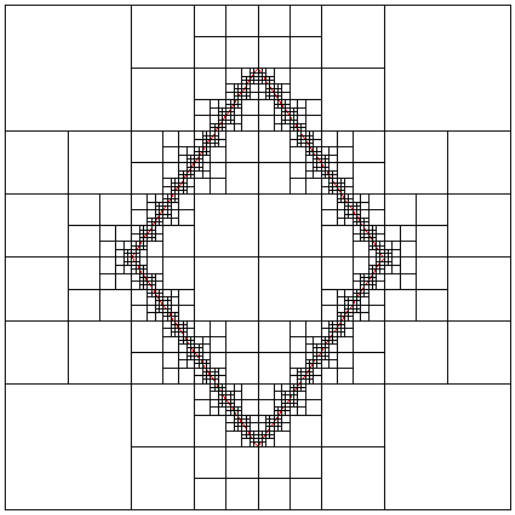

Figure 12: Domain $\Omega = [-1, 1] \times [-1, 1]$ and the spatial discretization used in example 5.1.4 and example 5.2.3.

effective resolution	L^∞ error	order	L^1 error	order
128×128	5.062×10^{-3}	—	5.298×10^{-4}	—
256×256	1.416×10^{-3}	1.838	1.387×10^{-4}	1.933
512×512	3.860×10^{-4}	1.875	3.508×10^{-5}	1.984
1024×1024	9.879×10^{-5}	1.966	8.735×10^{-6}	2.006

Table 7: Accuracy results for the solution u in example 5.1.4.

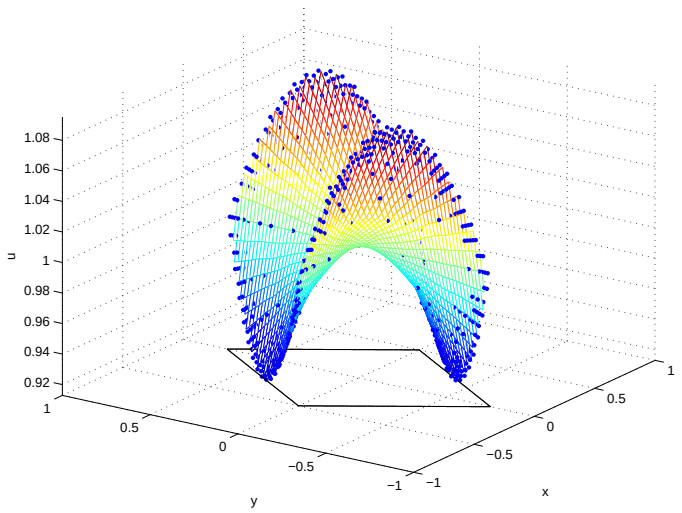

Figure 13: Graph of the solution in example 5.1.4. The dots represent the approximate solution and the mesh represents the exact solution.

effective resolution	L^∞ error	order	L^1 error	order
128×128	3.836×10^{-2}	—	1.583×10^{-2}	—
256×256	1.483×10^{-2}	1.371	4.642×10^{-3}	1.770
512×512	4.883×10^{-3}	1.603	1.252×10^{-3}	1.891
1024×1024	1.489×10^{-3}	1.713	3.227×10^{-4}	1.955

Table 8: Accuracy results for the solution gradient ∇u in example 5.1.4.

5.1.5 Example 5

Consider $\bigtriangledown \cdot (\rho \bigtriangledown u) = f$ on $\Omega = [-1,1] \times [-1,1]$ with an exact solution of $u = \cos(\pi x)\sin(\pi y)$, where $\rho = e^{xy}$. The interface is a cardioid given by the set of points where $\phi = ((3(x^2+y^2)-x)^2 - x^2 - y^2)/47 = 0$. Note that the interface is not even Lipschitz continuous and the singular point of the interface is the cusp point $(0,0)$. The non-graded Cartesian grid and the interface are illustrated in figure 14. Numerical accuracy test results for the solution and its gradient are given in table 9 and table 10, respectively. The numerical solution on a grid with an effective resolution of 128×128 is plotted in figure 15.

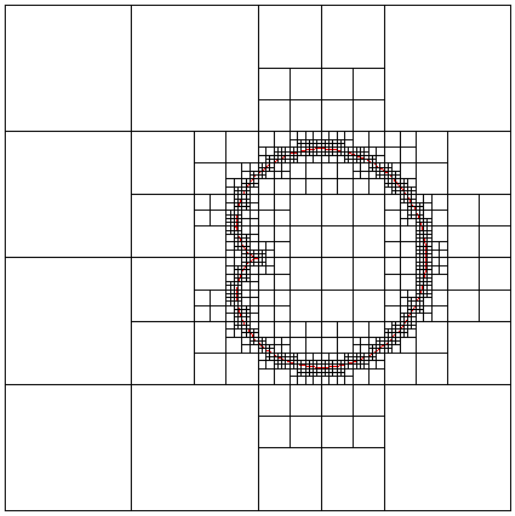

Figure 14: Domain $\Omega = [-1,1] \times [-1,1]$ and the spatial discretization used in example 5.1.5 and example 5.2.4.

effective resolution	L^∞ error	order	L^1 error	order
128×128	4.600×10^{-3}	—	4.538×10^{-4}	—
256×256	1.214×10^{-3}	1.922	1.148×10^{-4}	1.983
512×512	3.091×10^{-4}	1.973	2.885×10^{-5}	1.993
1024×1024	7.774×10^{-5}	1.991	7.236×10^{-6}	1.995

Table 9: Accuracy results for the solution u in example 5.1.5.

effective resolution	L^∞ error	order	L^1 error	order
128×128	5.676×10^{-2}	—	1.477×10^{-2}	—
256×256	1.582×10^{-2}	1.844	3.889×10^{-3}	1.925
512×512	4.329×10^{-3}	1.869	9.940×10^{-4}	1.968
1024×1024	1.176×10^{-3}	1.880	2.516×10^{-4}	1.982

Table 10: Accuracy results for the solution gradient ∇u in example 5.1.5.

5.1.6 Example 6

Consider $\bigtriangledown \cdot (\rho \bigtriangledown u) = f$ on $\Omega = [-1,1] \times [-1,1]$ with an exact solution of $u = e^{xy}$, where $\rho = x^2 + y^2$. The interface is star shaped, given by the set of points where $\phi = r - 0.5 - \frac{y^5+5x^4y-10x^2y^3}{3r^5} = 0$, where

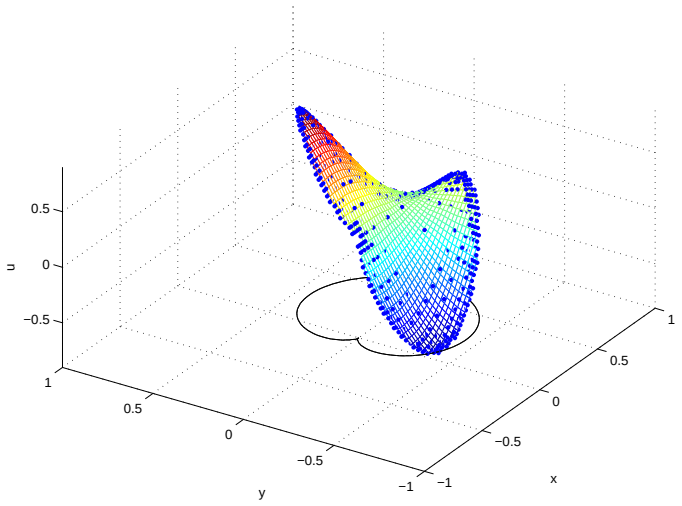

Figure 15: Graph of the solution in example 5.1.5. The dots represent the approximate solution and the mesh represents the exact solution.

$r = \sqrt{x^2 + y^2}$. The non-graded Cartesian grid and the interface are illustrated in figure 16. Numerical accuracy test results for the solution and its gradient are given in table 11 and table 12, respectively. The numerical solution on a grid with an effective resolution of 128×128 is plotted in figure 17.

effective resolution	L^∞ error	order	L^1 error	order
128×128	5.897×10^{-4}	—	6.999×10^{-5}	—
256×256	1.466×10^{-4}	2.008	1.600×10^{-5}	2.129
512×512	3.468×10^{-5}	2.080	3.837×10^{-6}	2.060
1024×1024	8.278×10^{-6}	2.067	9.393×10^{-7}	2.030

Table 11: Accuracy results for the solution u in example 5.1.6.

effective resolution	L^∞ error	order	L^1 error	order
128×128	1.683×10^{-2}	—	2.500×10^{-3}	—
256×256	4.237×10^{-3}	1.990	6.394×10^{-4}	1.967
512×512	1.029×10^{-3}	2.041	1.613×10^{-4}	1.987
1024×1024	3.356×10^{-4}	1.617	4.054×10^{-5}	1.992

Table 12: Accuracy results for the solution gradient ∇u in example 5.1.6.

5.1.7 Example 7

Consider $\bigtriangledown \cdot (\rho \bigtriangledown u) = f$ on $\Omega = [-2, 2] \times [-2, 2]$ with an exact solution of $u = \cos(\pi x)\sin(\pi y)$, where $\rho = 2 + \sin(\pi x)\cos(\pi y)$. The interface is given by the set of points where $\phi = 16y^4 - x^4 - 32y^2 + 9x^2 = 0$. The non-graded Cartesian grid and the interface are illustrated in figure 18. Numerical accuracy test results for the solution and its gradient are given in table 13 and table 14, respectively. The numerical solution on a grid with an effective resolution of 128×128 is plotted in figure 19.

5.2 Heat Equation

We consider equation (2) in two spatial dimensions, where ρ is a (possibly different) constant on each subdomain. Again, we only compute solutions on Ω^-. Dirichlet boundary conditions are used on the

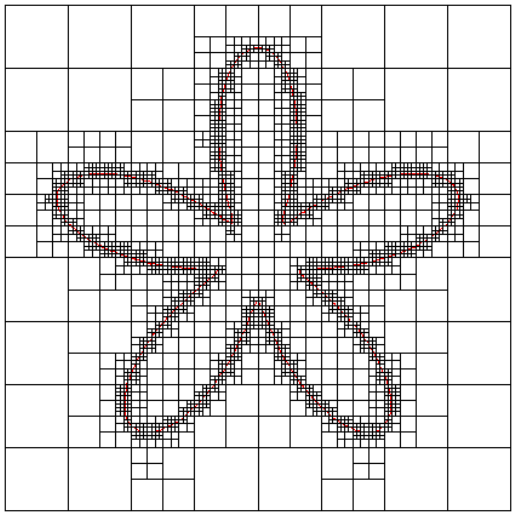

Figure 16: Domain $\Omega = [-1, 1] \times [-1, 1]$ and the spatial discretization used in example 5.1.6 and example 5.2.5.

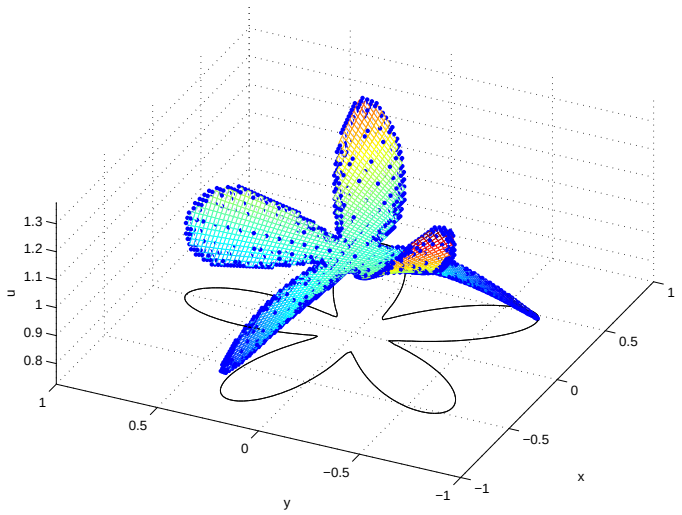

Figure 17: Graph of the solution in example 5.1.6. The dots represent the approximate solution and the mesh represents the exact solution.

effective resolution	L^∞ error	order	L^1 error	order
128×128	3.876×10^{-2}	—	4.420×10^{-3}	—
256×256	9.342×10^{-3}	2.053	1.033×10^{-3}	2.097
512×512	2.262×10^{-3}	2.046	2.473×10^{-4}	2.062
1024×1024	5.598×10^{-4}	2.014	6.048×10^{-5}	2.032

Table 13: Accuracy results for the solution u in example 5.1.7.

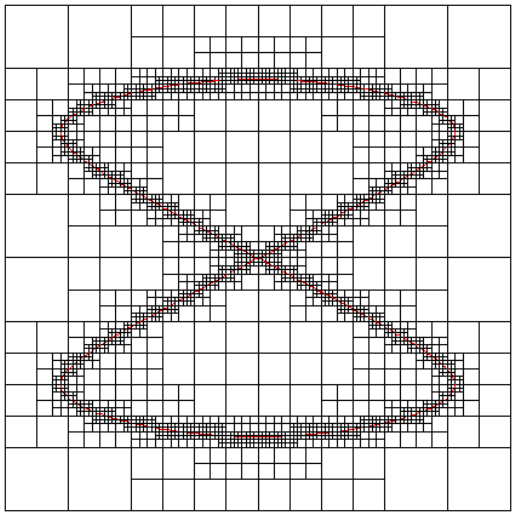

Figure 18: Domain $\Omega = [-2, 2] \times [-2, 2]$ and the spatial discretization used in example 5.1.7 and example 5.2.6.

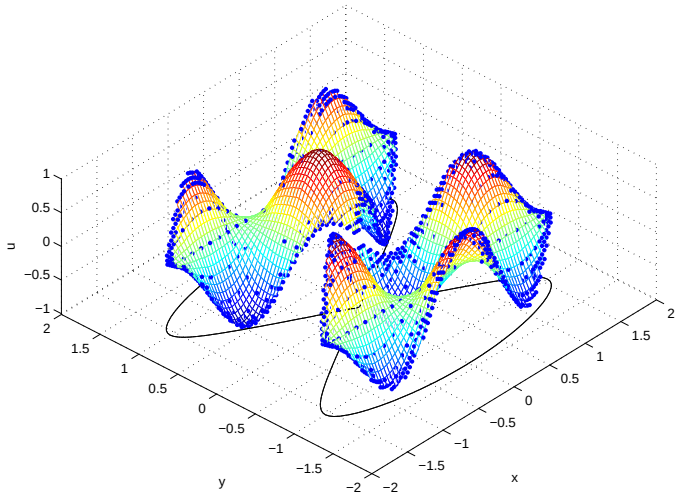

Figure 19: Graph of the solution in example 5.1.7. The dots represent the approximate solution and the mesh represents the exact solution.

effective resolution	L^∞ error	order	L^1 error	order
128×128	2.803×10^{-1}	—	8.066×10^{-2}	—
256×256	7.430×10^{-2}	1.915	1.981×10^{-2}	2.025
512×512	1.975×10^{-2}	1.911	4.896×10^{-3}	2.017
1024×1024	5.182×10^{-3}	1.931	1.220×10^{-3}	2.005

Table 14: Accuracy results for the solution gradient ∇u in example 5.1.7.

interface and domain boundaries. The time step in the Crank-Nicolson scheme is chosen as $\Delta t = \Delta x/2$, where Δx is the size of the smallest cell, so that an overall second order accuracy is obtained.

5.2.1 Example 1

Consider $u_t = \bigtriangledown \cdot (\rho \bigtriangledown u)$ on $\Omega = [-\pi, \pi] \times [-\pi, \pi]$ with an exact solution of $u = e^{-2\rho t} \sin(x) \cos(y)$, where $\rho = 1$. The interface is the same as in example 5.1.1 and figure 6. Numerical accuracy test results for the solution and its gradient are given in table 15 and table 16, respectively. The numerical solution at $t = 0.25$ on a grid with an effective resolution of 128×128 is plotted in figure 20.

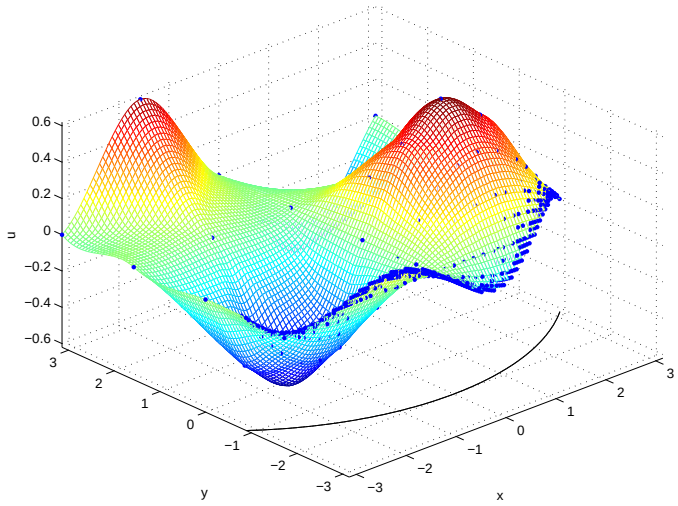

Figure 20: Graph of the solution in example 5.2.1. The dots represent the approximate solution and the mesh represents the exact solution.

effective resolution	L^∞ error	order	L^1 error	order
128×128	3.380×10^{-2}	—	1.409×10^{-3}	—
256×256	9.996×10^{-3}	1.758	4.308×10^{-4}	1.709
512×512	2.943×10^{-3}	1.764	1.120×10^{-4}	1.944
1024×1024	7.378×10^{-4}	1.996	2.798×10^{-5}	2.001

Table 15: Accuracy results for the solution u in example 5.2.1.

effective resolution	L^∞ error	order	L^1 error	order
128×128	3.629×10^{-1}	—	1.503×10^{-2}	—
256×256	9.796×10^{-2}	1.889	4.257×10^{-3}	1.820
512×512	3.027×10^{-2}	1.694	1.074×10^{-3}	1.986
1024×1024	8.212×10^{-3}	1.882	2.654×10^{-4}	2.017

Table 16: Accuracy results for the solution gradient ∇u in example 5.2.1.

5.2.2 Example 2

Consider $u_t = \bigtriangledown \cdot (\rho \bigtriangledown u)$ on $\Omega = [-1, 1] \times [-1, 1]$ with an exact solution of $u = e^{-2\pi^2 \rho t} \sin(\pi x) \sin(\pi y)$, where $\rho = 0.2$. The interface is the same as in example 5.1.2 and figure 8. Numerical accuracy test results for the solution and its gradient are given in table 17 and table 18, respectively. The numerical solution at $t = 0.25$ on a grid with an effective resolution of 128×128 is plotted in figure 21.

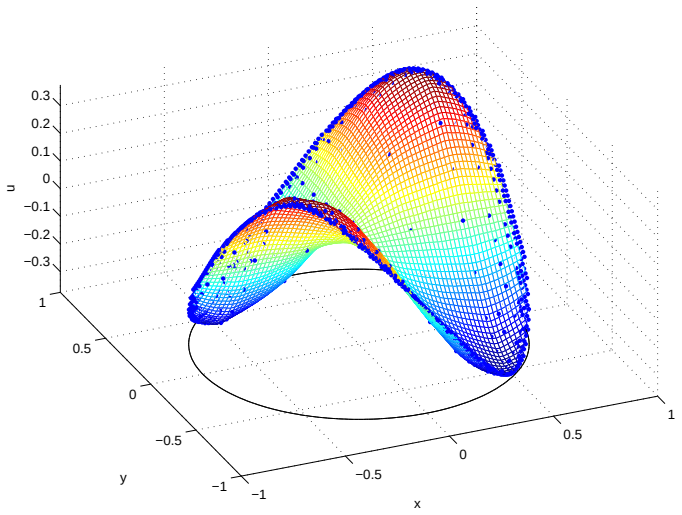

Figure 21: Graph of the solution in example 5.2.2. The dots represent the approximate solution and the mesh represents the exact solution.

effective resolution	L^∞ error	order	L^1 error	order
128×128	6.957×10^{-3}	—	1.021×10^{-3}	—
256×256	1.968×10^{-3}	1.822	2.182×10^{-4}	2.227
512×512	4.906×10^{-4}	2.004	5.069×10^{-5}	2.106
1024×1024	1.237×10^{-4}	1.988	1.224×10^{-5}	2.049

Table 17: Accuracy results for the solution u in example 5.2.2.

effective resolution	L^∞ error	order	L^1 error	order
128×128	6.459×10^{-2}	—	2.839×10^{-2}	—
256×256	1.892×10^{-2}	1.772	6.510×10^{-3}	2.124
512×512	5.304×10^{-3}	1.834	1.575×10^{-3}	2.047
1024×1024	1.395×10^{-3}	1.927	3.878×10^{-4}	2.022

Table 18: Accuracy results for the solution gradient ∇u in example 5.2.2.

5.2.3 Example 3

Consider $u_t = \bigtriangledown \cdot (\rho \bigtriangledown u)$ on $\Omega = [-1, 1] \times [-1, 1]$ with an exact solution of $u = e^{-2\pi^2 \rho t} \cos(\pi x) \sin(\pi y)$, where $\rho = 1$. The interface is the same as in example 5.1.4 and figure 12. Numerical accuracy test results for the solution and its gradient are given in table 19 and table 20, respectively. The numerical solution at $t = 0.25$ on a grid with an effective resolution of 128×128 is plotted in figure 22.

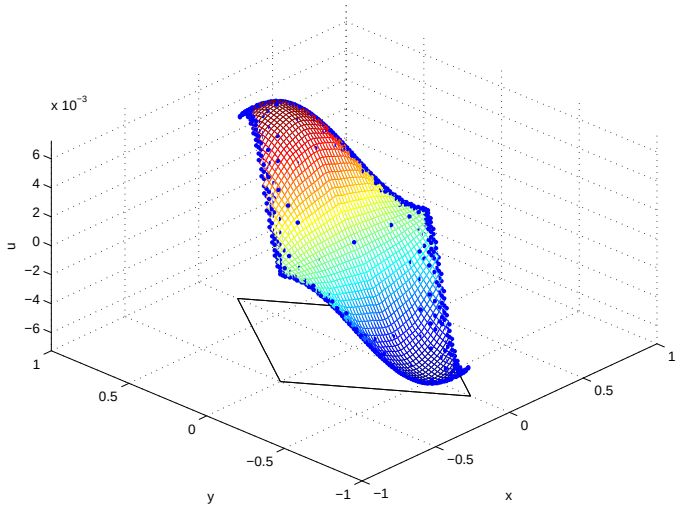

Figure 22: Graph of the solution in example 5.2.3. The dots represent the approximate solution and the mesh represents the exact solution.

effective resolution	L^∞ error	order	L^1 error	order
128×128	2.845×10^{-4}	—	2.183×10^{-5}	—
256×256	7.080×10^{-5}	2.007	4.810×10^{-6}	2.182
512×512	1.750×10^{-5}	2.016	1.114×10^{-6}	2.110
1024×1024	4.344×10^{-6}	2.011	2.683×10^{-7}	2.054

Table 19: Accuracy results for the solution u in example 5.1.3.

effective resolution	L^∞ error	order	L^1 error	order
128×128	1.966×10^{-3}	—	6.214×10^{-4}	—
256×256	4.992×10^{-4}	1.978	1.437×10^{-4}	2.112
512×512	1.292×10^{-4}	1.950	3.426×10^{-5}	2.069
1024×1024	3.388×10^{-5}	1.931	8.393×10^{-6}	2.029

Table 20: Accuracy results for the solution gradient ∇u in example 5.1.3.

5.2.4 Example 4

Consider $u_t = \bigtriangledown \cdot (\rho \bigtriangledown u)$ on $\Omega = [-1, 1] \times [-1, 1]$ with an exact solution of $u = e^{-2\rho t} \cos(x) \cos(y)$, where $\rho = 1$. The interface is the same as in example 5.1.4 and figure 14. Numerical accuracy test results for the solution and its gradient are given in table 21 and table 22, respectively. The numerical solution at $t = 0.25$ on a grid with an effective resolution of 128×128 is plotted in figure 23.

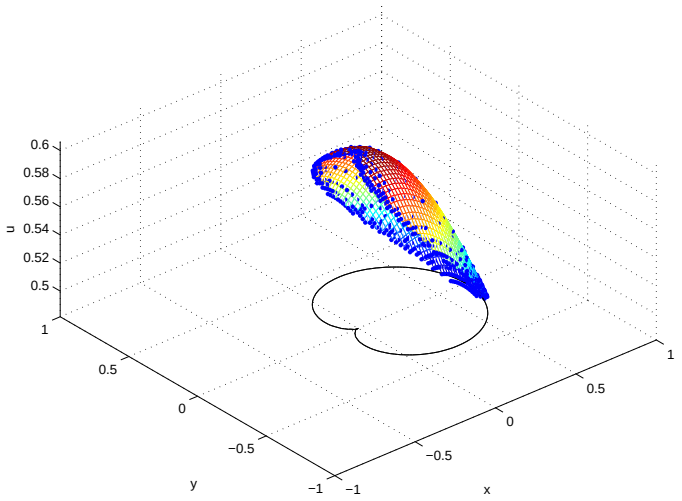

Figure 23: Graph of the solution in example 5.2.4. The dots represent the approximate solution and the mesh represents the exact solution.

effective resolution	L^∞ error	order	L^1 error	order
128×128	6.928×10^{-5}	—	8.809×10^{-6}	—
256×256	1.743×10^{-5}	1.991	2.168×10^{-6}	2.023
512×512	4.336×10^{-6}	2.007	5.351×10^{-7}	2.019
1024×1024	1.080×10^{-6}	2.006	1.327×10^{-7}	2.011

Table 21: Accuracy results for the solution u in example 5.2.4.

effective resolution	L^∞ error	order	L^1 error	order
128×128	7.953×10^{-4}	—	2.944×10^{-4}	—
256×256	2.254×10^{-4}	1.819	7.623×10^{-5}	1.949
512×512	6.181×10^{-5}	1.866	1.923×10^{-5}	1.987
1024×1024	1.662×10^{-5}	1.895	4.834×10^{-6}	1.992

Table 22: Accuracy results for the solution gradient ∇u in example 5.2.4.

5.2.5 Example 5

Consider $u_t = \bigtriangledown \cdot (\rho \bigtriangledown u)$ on $\Omega = [-1, 1] \times [-1, 1]$ with an exact solution of $u = e^{-\rho t}(\sin(x) + \cos(y))$, where $\rho = 8$. The interface is the same as in example 5.1.6 and figure 16. Numerical accuracy test results for the solution and its gradient are given in table 23 and table 24, respectively. The numerical solution at $t = 0.25$ on a grid with an effective resolution of 128×128 is plotted in figure 24.

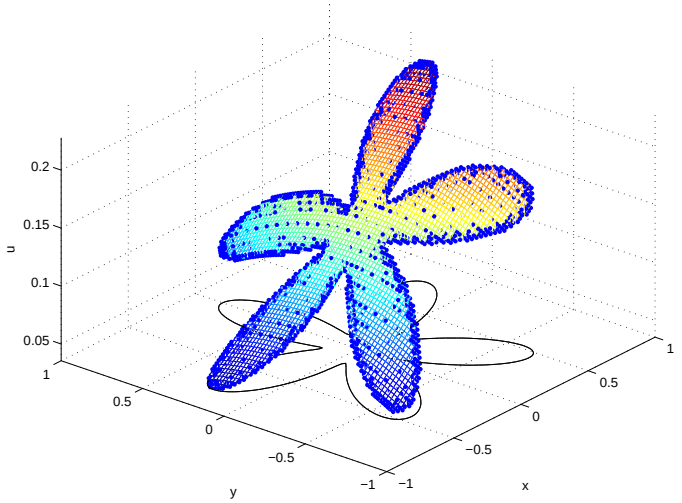

Figure 24: Graph of the solution in example 5.2.5. The dots represent the approximate solution and the mesh represents the exact solution.

effective resolution	L^∞ error	order	L^1 error	order
128×128	3.621×10^{-6}	—	6.389×10^{-7}	—
256×256	8.260×10^{-7}	2.132	1.700×10^{-7}	1.910
512×512	1.933×10^{-7}	2.095	3.983×10^{-8}	2.093
1024×1024	4.556×10^{-8}	2.085	9.652×10^{-9}	2.045

Table 23: Accuracy results for the solution u in example 5.2.5.

effective resolution	L^∞ error	order	L^1 error	order
128×128	1.073×10^{-4}	—	3.074×10^{-5}	—
256×256	3.889×10^{-5}	1.464	8.933×10^{-6}	1.783
512×512	7.729×10^{-6}	2.331	2.304×10^{-6}	1.955
1024×1024	2.284×10^{-6}	1.759	5.971×10^{-7}	1.948

Table 24: Accuracy results for the solution gradient ∇u in example 5.2.5.

5.2.6 Example 6

Consider $u_t = \bigtriangledown \cdot (\rho \bigtriangledown u)$ on $\Omega = [-2, 2] \times [-2, 2]$ with an exact solution of $u = e^{-2\pi^2 \rho t} \cos(\pi x) \sin(\pi y)$, where $\rho = 0.2$. The interface is the same as in example 5.1.7 and figure 18. Numerical accuracy test results for the solution and its gradient are given in table 25 and table 26, respectively. The numerical solution at $t = 0.25$ on a grid with an effective resolution of 128×128 is plotted in figure 25.

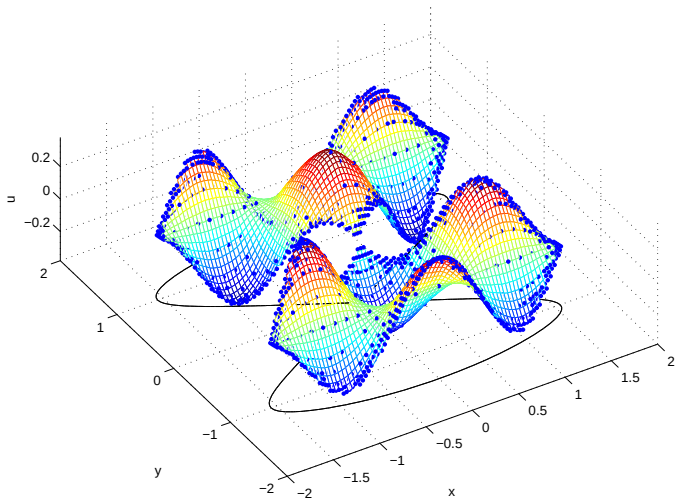

Figure 25: Graph of the solution in example 5.2.6. The dots represent the approximate solution and the mesh represents the exact solution.

effective resolution	L^∞ error	order	L^1 error	order
128×128	1.671×10^{-2}	—	1.982×10^{-3}	—
256×256	4.060×10^{-3}	2.041	4.666×10^{-4}	2.087
512×512	1.003×10^{-3}	2.017	1.126×10^{-4}	2.051
1024×1024	2.498×10^{-4}	2.006	2.767×10^{-5}	2.025

Table 25: Accuracy results for the solution u in example 5.2.6.

effective resolution	L^∞ error	order	L^1 error	order
128×128	1.089×10^{-1}	—	3.570×10^{-2}	—
256×256	2.767×10^{-2}	1.976	8.784×10^{-3}	2.023
512×512	7.574×10^{-3}	1.869	2.180×10^{-3}	2.011
1024×1024	2.029×10^{-3}	1.901	5.447×10^{-4}	2.001

Table 26: Accuracy results for the solution gradient ∇u in example 5.2.6.

6 Conclusions

We have proposed finite difference algorithms for the variable coefficient Poisson equation as well as for the heat equation on irregular domains and non-graded Cartesian grids. These schemes produce second order accuracy for the solution and its gradients. Sampling the solution at the nodes produces efficient and simple procedures that can be applied in a dimension by dimension framework. At T-junctions, linear interpolations are used to generate intermediate values used in the discretizations. These intermediate values introduce spurious $O(1)$ errors that are successfully cancelled by simple linear combinations of the discretizations in the transverse directions. T-junction nodes neighboring the irregular interface are excluded by imposing that the smallest cells lie on the interface. For nodes neighboring the interface, quadratic interpolation is used to find both the location of the interface and the value of the solution at the interface. The calculation of the solution gradients involves the same cells as those used in the discretization of the Poisson or heat equation, hence preserving the locality and ease of implementation of the method. In the case of the heat equation, we utilize an implicit time discretization to overcome the time step restrictions induced by explicit schemes. We have presented numerical results to demonstrate the second order accuracy in the L^1 and L^∞ norms for the solution and its gradients. Future work will include the design of a simple second order scheme for the Stefan problem.

References

[1] M. J. Aftosmis, M. J. Berger, and J. E. Melton. Adaptive Cartesian Mesh Generation. In *CRC Handbook of Mesh Generation (Contributed Chapter)*, 1998.

[2] A. Almgren. *A Fast Adaptive Vortex Method Using Local Corrections.* PhD thesis, University of California, Berkeley, 1991.

[3] A. Almgren, J. Bell, P. Colella, L. Howell, and M. Welcome. A conservative adaptive projection method for the variable density incompressible navier-stokes equations. *J. Comput. Phys.*, 142:1–46, 1998.

[4] A. Almgren, R. Buttke, and P. Colella. A fast adaptive vortex method in three dimensions. *J. Comput. Phys.*, 113:177–200, 1994.

[5] M. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

[6] I. B. (Ed.). *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations.* Springer-Verlag, New York, NY, 1995.

[7] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.*, 152:457–492, 1999.

[8] F. Gibou and R. Fedkiw. A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem. *J, Comput. Phys.*, 202:577–601, 2005.

[9] F. Gibou, R. Fedkiw, L.-T. Cheng, and M. Kang. A second–order–accurate symmetric discretization of the poisson equation on irregular domains. *J. Comput. Phys.*, 176:205–227, 2002.

[10] H. Johansen and P. Colella. A cartesian grid embedded boundary method for poisson's equation on irregular domains. *J. Comput. Phys.*, 147:60–85, 1998.

[11] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Cambridge Univ. Press, New York, NY, 1987.

[12] Z. Jomaa and C. Macaskill. The embedded finite difference method for the poisson equation in a domain with an irregular boundary and dirichlet boundary conditions. *J. Comput. Phys.*, 202:488–506, 2005.

[13] R. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources 31:1019–1044, 1994. *SIAM J. Numer. Anal.*, 31:1019–1044, 1994.

[14] K. Lipnikov, J. Morel, and M. Shashkov. Mimetic finite difference methods for diffusion equations on non-orthogonal non-conformal meshes. *J. Comput. Phys.*, 199:589–597, 2004.

[15] X. Liu, R. Fedkiw, and M. Kang. A boundary condition capturing method for Poisson's equation on irregular domains. *J. Comput. Phys.*, 154:151, 2000.

[16] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids (in press).*

[17] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH Proc.)*, pages 457–462, 2004.

[18] A. Mayo. The fast solution of poisson's and the biharmonic equations on irregular regions. *SIAM J. Numer. Anal.*, 21:285–299, 1984.

[19] P. McCorquodale, P. Colella, D. Grote, and J.-L. Vay. A node-centered local refinement algorithm for poisson's equation in complex geometries. *J. Comput. Phys.*, 201:34–60, 2004.

[20] A. McKenney and L. Greengard. A fast poisson solver for complex geometries. *J. Comput. Phys.*, 118:348–355, 1995.

[21] C. Min, F. Gibou, and H. Ceniceros. A supra-convergent finite difference scheme for the variable coefficient poisson equation on fully adaptive grids. *CAM report 05-29, Accepted in J. Comput. Phys.*

[22] C. Min, F. Gibou, and H. D. Ceniceros. A simple super-convergent finite different scheme for the vairable coefficient poisson equation on fully adaptive grids. *J. Comput. Phys.*, (in press).

[23] C. Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25:220–252, 1977.

[24] S. Popinet. Gerris: A tree-based adaptive solver for the incompressible euler equations in complex geometries. *J. Comput. Phys.*, 190:572–600, 2003.

[25] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing, 1996. New York, NY.

[26] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, New York, 1989.

[27] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS.* Addison-Wesley, New York, 1990.

[28] A. Schmidt. Computation of three dimensional dendrites with finite elements. *J. Comput. Phys.*, 125:293–312, 1996.

[29] J. Strain. Tree methods for moving interfaces. *J. Comput. Phys.*, 151:616–648, 1999.

[30] D. Young, R. Melvin, M. Bieterman, F. Johnson, S. Samant, and J. Bussoletti. A locally refined rectangular grid finite element method: Application to computational fluid dynamics and computational physics. *J. Comput. Phys.*, 92:1–66, 1991.