

A Supra-Convergent Finite Difference Scheme for the Variable Coefficient Poisson Equation on Fully Adaptive Grids

Chohong Min *

Frédéric Gibou †

Hector D. Ceniceros ‡

2nd February 2006

Abstract

We introduce a method for solving the variable coefficient Poisson equation on *fully* adaptive Cartesian grids that yields second order accuracy for the solutions *and* their gradients. We employ quadtree (in 2D) and octree (in 3D) data structures as an efficient means to represent the Cartesian grid, allowing for constraint-free grid generation. The schemes take advantage of sampling the solution at the nodes (vertices) of each cell. In particular, the discretization at one cell's node only uses nodes of two (2D) or three (3D) adjacent cells, producing schemes that are straightforward to implement. Numerical results in two and three spatial dimensions demonstrate supra-convergence in the L^∞ norm.

1 Introduction

The variable coefficient Poisson equation is a model at the core of diffusion dominated phenomena and therefore is a component used in the simulation of many important applications ranging from incompressible flows to semiconductor modeling to tissue engineering. Several approaches exist to solve numerically the variable coefficient Poisson equation on uniform grids in the case of regular domains (see e.g. [15] and the references therein), as well as in the case of irregular domains (see e.g. [10, 11, 14, 17, 18, 23, 24, 25] and the references therein). However, many physical problems have variations in scale and when solving these problems numerically, uniform Cartesian grids are limited in their ability to resolve small scales. Uniform grids in such situations are inefficient in terms of memory storage and CPU requirements since only small portions of the computational domain require fine resolutions. Therefore, it is highly desirable to lower the total number of grid cells involved in the discretization. Since their inception, adaptive mesh refinement techniques have provided a tool to systematically concentrate the computational effort where it is most needed, allowing for efficient resolution.

Adaptive mesh strategies are becoming popular, see e.g. [13, 30, 7] in the case of the study of incompressible flows. However, implementations based on recursive structures, such as quadtree/octree are less common. While in some cases (e.g. compressible flows) the mesh must be constrained to avoid non-physical effects (e.g. spurious shock reflections [5]), several applications including incompressible flows and the Stefan problem for example, do not require any special constraints and thus more optimal data structures can be used. Quadtree/Octree data structures have been proven to be optimal in such cases, as pointed out in [1].

Discretizations of an elliptic linear partial differential equation by finite differences, finite elements or the finite volume method result in a linear system to be solved. The numerical efficiency of finite element methods (F.E.M.) is in large part attributed to the fact that such formulations always yield symmetric linear systems, which are computationally inexpensive to invert, e.g. with preconditioned conjugate gradient methods [12, 27]. Moreover, the theoretical framework associated with F.E.M. makes these methods attractive. However, F.E.M. on adaptive meshes is often challenging to implement due to the large number of special cases to take into account when discretizing the equations. This stems from the fact that the support of the basis functions associated with one cell might involve an arbitrary number of neighboring cells, making a

*Mathematics Department, University of California, Santa Barbara, CA 93106.

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

‡Mathematics Department, University of California, Santa Barbara, CA 93106. Supported in part by NSF DMS-0311911.

general procedure difficult to write. Constraints can be introduced to limit the ratio between two adjacent cell sizes to at most two (graded trees). Although graded trees can be constructed from non-graded trees, such operations introduce a large amount of additional cells in the domain, significantly lowering the efficiency of the method.

Young *et al.* [31] introduced a finite element method employing adaptive mesh refinements for second-order variable-coefficient elliptic equations using a cut-cell representation of irregular domains. Almgren *et al.* [3] solved the variable coefficient Poisson equation in a block-structured adaptive grid in the context of the incompressible Navier-Stokes equations. Johansen and Colella [14] presented a cell-centered numerical 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 [6]. McCorquodale *et al.* presented a node-centered approach to solving the variable coefficient Poisson equation on irregular domains using the block-structured adaptive mesh refinement multigrid solver of Almgren [2, 4]. These schemes, however, do not consider fully adaptive meshes.

In [26], Popinet proposed a second order non-symmetric numerical method to study the incompressible Navier-Stokes equations using an octree data structure. In his method, a Poisson equation for the pressure needs to be solved to account for the incompressibility condition using a standard projection method (see e.g. [8]). The pressure is sampled at the center of each cell and the discretization of the Poisson equation requires interpolation procedures involving the pressure values at several adjacent cells. As a consequence, this discretization is intricate and yields a wide band in the linear system. Moreover, only graded trees were considered in [26].

Losasso *et al.* [21] proposed a first order accurate Poisson solver on fully adaptive grids and applied this solver to the motion of free surface flows. The work relied on the observation that, in the case of the Poisson equation, first order perturbations in the location of the solution produce consistent schemes (see [11]). Moreover, by limiting the Poisson solver to first order accuracy, a symmetric discretization that is computationally efficient to invert was achieved. This work was extended to second order accuracy in [20] using some of the ideas discussed by Lipnikov *et al.* in [19]. However, it is unclear whether or not the solution's gradients can be found to second order accuracy. In some applications (e.g. Stefan type problems), the accuracy of the solution's gradients determine the overall accuracy of the method and therefore a first order accurate solution may lead to a degradation in the accuracy.

In this paper we propose a second order accurate finite difference discretization for the variable coefficient Poisson equation on fully adaptive grids that also yields second order accuracy in the solution's gradients. The scheme is based on sampling the solution at the nodes of a cell. The discretization at one cell's node only uses nodes of two (2D) or three (3D) adjacent cells, producing schemes that are straightforward to implement. Examples in two and three spatial discretizations demonstrate second order accuracy in the L^∞ norms for the solution and its gradients.

2 Spatial Discretization

The domain is discretized into squares (in 2D) or cubes (in 3D) and we use a standard quadtree (in 2D) or octree (in 3D) data structure to represent this discretization. More precisely, consider the case depicted in Fig. 1 in two spatial dimensions: the entire domain is originally associated with the root of the tree and then split into four cells of equal sizes, called the children of the root. The discretization proceeds recursively, i.e. each cell can be in turn split into four children and so forth. A cell with no children is called a leaf. By definition, the level of the root is zero and is incremented by one every time a new generation of children is added. The anisotropy ratio of a cell is defined as the ratio between the width and length of that cell. The anisotropy ratio of a quadtree is the maximum of the anisotropy ratios. Hence, a quadtree of anisotropy ratio of one has squares only in the subdivision. Finally two cells are called neighbors if they share a common face. As mentioned in the introduction, discretizations found in finite element methods often limit the difference of level between two adjacent cells to one to simplify the calculations and the number of cases to consider. Popinet [26] also applies this constraint to his finite difference method, which leads to graded trees. In the method we propose in this work, we do not impose any constraints on the difference of level between two adjacent cells, yielding a fully adaptive mesh generation.

Similarly, in three spatial dimensions, the domain (root) is split in eight cubes (children) and each cell

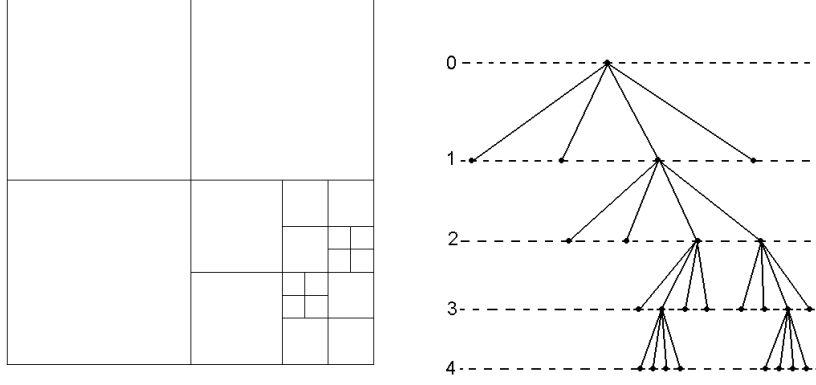


Figure 1: Discretization of a two dimensional domain (left) and its quadtree representation (right). The entire domain corresponds to the root of the tree (level 0), and each cell subdivided further points to its four children. In this example, the tree is not graded since the difference of level between neighboring cells can exceed one.

can be recursively split in a similar manner. The interested reader is referred to [28, 29] for more on octree data structures.

3 Accuracy Analysis

In the case of one dimensional second order differential equations on irregular grids, Manteuffel *et al.* [22] as well as Kreiss *et al.* [16] proved that standard central differencing schemes produce second order accurate solutions, even though local Taylor analysis dictates that the methods are at most first order accurate (see also Ciarlet [9]). In the case of the Poisson equation on irregular grids, supra-convergence has been observed numerically by Johansen and Colella [14]. Gibou *et al.* [11] also observed this fact in the case of the Poisson equation on uniform grids but irregular domains. In [21], Losasso *et al.* proposed a first order accurate scheme for solving the Poisson equation on fully adaptive grids in the context of the Navier-Stokes equations. They demonstrate that their scheme is globally first order accurate (consistent), even though the discretization at nonuniform mesh points is inconsistent. In fact, the different approximations of the pressure gradients in [21] result in consistent schemes, regardless of how the distance between the two adjacent cells involved in the discretization of the pressure gradients was accounted for. In this case, the scheme is therefore locally inconsistent on nonuniform meshes but reduces to a second order accurate discretization at locally uniform mesh points. This was explained by the fact that first order perturbations in the location produce a consistent method as demonstrated in [10, 11]. Johansen and Colella [14] provided a heuristic argument based on potential theory as to why schemes that are only first order accurate at locally nonuniform grid nodes can be globally second order accurate. This phenomena can be intuitively understood as follows: Consider a discretization Ω_h of a Cartesian domain $\Omega \in \mathbb{R}^d$, with N_{total} number of cells and $N_{nonunif}$ number of nonuniform cells. Here, we say that a cell is locally nonuniform if its size is different from the size of at least one of its neighbors whereas a cell is locally uniform if its size is equal to that of all of its neighbors. After the p^{th} refinement, the grid contains $N_{total} \cdot 2^{d \cdot p}$ cells, with $O(N_{nonunif} \cdot 2^{(d-1)p})$ nonuniform cells, since the set of nonuniform cells is of codimension one. Therefore, after the p^{th} refinement, there are $O(2^{d \cdot p})$ locally uniform cells and $O(2^{(d-1)p})$ locally nonuniform cells. In turn, the influence of nonuniform cells is absorbed by that of the uniform one, yielding a second order accurate scheme (see [14] for details). Based on this argument and on numerical evidence, we hypothesize in this paper that a strategy for deriving p^{th} order accurate finite difference schemes in the L^∞ norm, is to focus on designing schemes that are $(p-1)^{th}$ order accurate at locally nonuniform cells, which reduce to at least p^{th} order accurate schemes at locally uniform cells. In particular, in order to derive second order accurate schemes, we will focus on finding a consistent approximation at non-uniform cells.

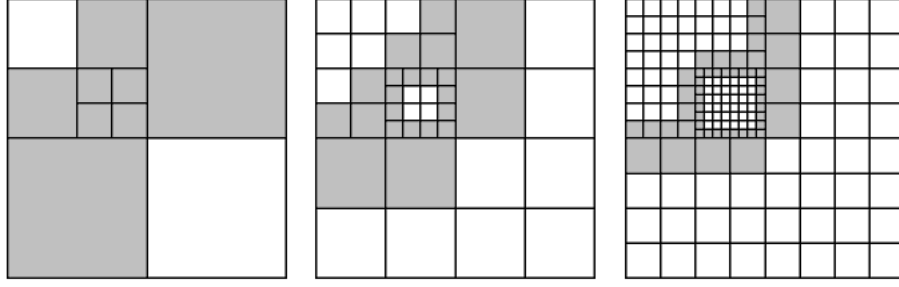


Figure 2: Example of refinement in two spatial dimensions. The total number of cells increases quadratically whereas the number of locally nonuniform cells (shaded) increases linearly. The contribution of nonuniform cells decreases relatively to that of uniform cells.

4 A Note on Center-Based and Node-Based Discretizations

There are mainly two standard choices for sampling the solution of an elliptic problem: sampling at the nodes or at the center of each cell. A cell-centered sampling often leads to a symmetric linear system, since the relation between neighbors is reflective. For example, in the case depicted in Fig. 3 (left), it is geometrically natural to define the discretization at c_4 in terms of c_2, c_3, c_5, c_6 . Likewise, the discretizations at c_2, c_3, c_5 and c_6 will all naturally include c_4 . In contrast, in the case of the node-based sampling depicted in Fig. 3 (right), it is geometrically natural to define the discretization at v_1 in terms of v_2, v_3, v_4, v_5, v_6 , and to define the discretization at v_6 in terms of v_4, v_5, v_7 . As a consequence, the equation for v_1 involves v_6 , but the discretization for v_6 does not involve v_1 , and thus produces nonsymmetric discretizations. In the case of cell-centered sampling, one can prove the following result:

Theorem 1 *Consider the discretization of the Poisson equation at the center of a cell C . If only adjacent cells are to be used, then there does not exist any locally consistent linear scheme on nonuniform Cartesian grids.*

Proof Referring to Fig. 3 (left), let u_i be the solution at c_i and consider the discretization at c_4 . For a linear consistent scheme to exist, we must be able to find the coefficients a_i such that

$$a_4 u_4 + a_2 u_2 + a_3 u_3 + a_5 u_5 + a_6 u_6 = u_{xx} + u_{yy} + O(h),$$

where h denotes the length of the edge of cell c_4 . A simple Taylor analysis implies that the coefficients a_i must satisfy the following linear system

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & -h & 0 & \frac{3h}{2} & -\frac{h}{2} \\ 0 & 0 & h & \frac{h}{2} & -\frac{3h}{2} \\ 0 & \frac{h^2}{2} & 0 & \frac{9h^2}{8} & \frac{h^2}{8} \\ 0 & 0 & 0 & \frac{3h^2}{4} & \frac{3h^2}{8} \\ 0 & 0 & \frac{h^2}{2} & \frac{h^2}{8} & \frac{9h^2}{8} \end{pmatrix} \begin{pmatrix} a_4 \\ a_3 \\ a_2 \\ a_5 \\ a_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix},$$

which does not have a solution. ■

In the same fashion, one can prove the nonexistence of any locally consistent linear methods even when all the neighboring cells' values of u and f are used (i.e. also including u_1, u_7 and f_1, f_7 in this example).

We caution the reader that this observation along with Theorem 1 does not imply that there does not exist any cell-centered linear schemes that are globally second order accurate. In fact, the finite element method and discretizations using the hierarchically nested mesh refinements of Berger and Olinger [6] provide obvious counter examples (see Fig. 4 for a locally inconsistent discretization that yields second order accurate

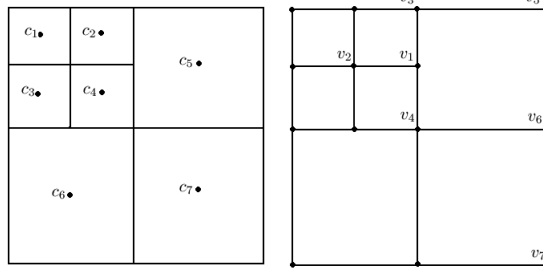


Figure 3: Cell-centered sampling (left) and node-based sampling (right).

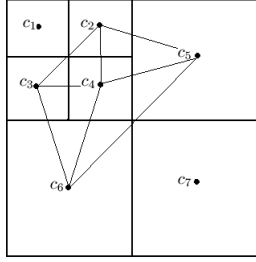


Figure 4: Standard triangulation in a F.E.M. framework. Using linear basis functions, the discretization at c_4 reads $-\frac{7}{6h^2} \cdot (u_2 + u_3 + u_5 + u_6) + \frac{28}{6h^2} \cdot u_4 = f_4$, and although it can be shown to be locally inconsistent in terms of Taylor expansions it leads to a globally second order accurate method.

solutions). The recent work of [20] is another example. Rather, this analysis serves as a guideline for deriving high order accurate methods: A sufficient condition for obtaining a globally second order accurate scheme is to first obtain a locally consistent scheme. In the case of a cell-centered scheme, one avenue is to increase the number of neighboring cells involved in the discretization as in [26]. However, this approach produces more intricate schemes and led the author in [26] to consider only graded trees to ease the implementation. Another possible framework is to sample the solution at the nodes, which we present next.

5 Laplace Equation

Consider the Laplace equation $\Delta u = f$ on a Cartesian domain Ω , with at least one Dirichlet boundary condition $u|_{\partial\Omega} = g$ on the domain's boundary $\partial\Omega$. In this section, we introduce schemes to solve this problem in \mathbb{R}^n , $n = 1, 2, 3$ that gives second order accuracy for the solutions and their gradients. Sampling the solution at the nodes produces a very efficient algorithm that can be applied in a dimension by dimension framework. Notably, the discretization at the node of one cell only involves some values of the cell itself and at most two of its neighboring cells, leading to methods that are straightforward to implement.

5.1 One Spatial Dimension

In the case of a nonuniform grid, the standard finite difference discretization for the one dimensional Laplace equation at a grid point x_i can be written as

$$\left(\frac{u_{i+1} - u_i}{s_{i+\frac{1}{2}}} - \frac{u_i - u_{i-1}}{s_{i-\frac{1}{2}}} \right) \cdot \frac{2}{s_{i-\frac{1}{2}} + s_{i+\frac{1}{2}}} = f_i, \quad (1)$$

where $s_{i-\frac{1}{2}} = x_i - x_{i-1}$. A simple Taylor analysis (with the standard abuse of notations) demonstrates that

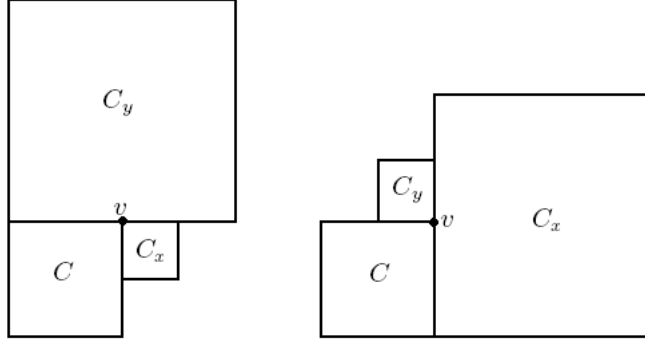


Figure 5: There are two possible geometric configurations for C , C_x and C_y . Either C and C_x are edge-aligned with respect to the node v (left) or C and C_y are edge-aligned with respect to the node v (right).

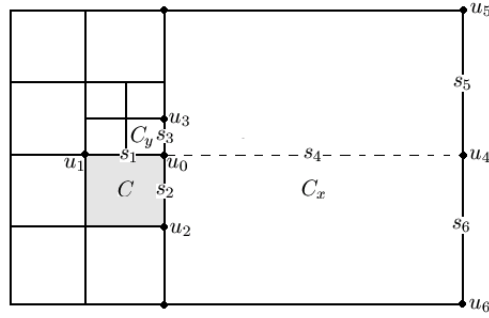


Figure 6: A configuration illustrating the nodes involved in the discretization at u_0 in the two dimensional case.

$$\left(\frac{u_{i+1} - u_i}{s_{i+\frac{1}{2}}} + \frac{u_i - u_{i-1}}{s_{i-\frac{1}{2}}} \right) \cdot \frac{2}{s_{i-\frac{1}{2}} + s_{i+\frac{1}{2}}} = f_i + \frac{s_{i+\frac{1}{2}} - s_{i-\frac{1}{2}}}{3} u_{xxx} + O(h^2),$$

where we take $h = \max_i s_{i+1/2}$. Therefore, the scheme appears to be second order accurate only when the grid is uniform. Again, Taylor analysis gives only sufficient conditions for accuracy. In fact, it has been shown (see e.g. [10, 16, 22]) that discretizations of this type are actually second order accurate.

5.2 Two Spatial Dimensions

The discretization in Section 5.1 can be applied in a dimension by dimension framework: consider a node u_0 of a cell C with a neighboring cell C_x in x -direction and a neighboring cell C_y in y -direction that both contain u_0 . Since C , C_x and C_y all include u_0 , either C_x or C_y should be edge-aligned with C (with respect to u_0) as shown in Fig. 5.

Referring to Fig. 6, to apply the one dimensional finite difference discretization at the node u_0 , an intermediate value u_4 is linearly interpolated from u_5 and u_6 :

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

Taylor analysis gives the following results for the standard discretizations in the x and y directions, respectively:

$$\left(\frac{u_4 - u_0}{s_4} - \frac{u_0 - u_1}{s_1} \right) \cdot \frac{2}{s_1 + s_4} = u_{xx} + \frac{s_5 s_6}{(s_1 + s_4) \cdot s_4} u_{yy} + O(h), \quad (2)$$

$$\left(\frac{u_3 - u_0}{s_3} - \frac{u_0 - u_2}{s_2} \right) \cdot \frac{2}{s_2 + s_3} = u_{yy} + O(h). \quad (3)$$

The analysis in Section 5.1 demonstrates that the method will be second order accurate if one can compensate for the spurious term involving u_{yy} in Equation (2). This is easily achieved by a linear combination of equations (2) and (3):

$$\left(\frac{u_4 - u_0}{s_4} - \frac{u_0 - u_1}{s_1} \right) \cdot \frac{2}{s_1 + s_4} + \left(\frac{u_3 - u_0}{s_3} - \frac{u_0 - u_2}{s_2} \right) \cdot \frac{2}{s_2 + s_3} \cdot w = f_0 + O(h), \quad (4)$$

where $w = 1 - \frac{s_5 s_6}{(s_1 + s_4) \cdot s_4}$. Note that the same strategy applies to the configuration in which C_x is edge-aligned with C . Also, not surprisingly, the weighted scheme reduces to the usual central scheme in the case of a locally uniform grid, i.e. $s_1 = s_4$, $s_2 = s_3$, and either $s_5 = 0$ or $s_6 = 0$ (in which case u_4 is simply not needed). This process can be applied in a dimension by dimension framework, making the discretization straightforward to implement.

The corresponding linear system is nonsingular in the practical case where the cells are isotropic (squares or cubes). In fact, it is easy to formulate and prove the following theorem stating that rectangular cells can be considered as well, given a mild constraint on the anisotropic ratio.

Theorem 2 *The matrix induced by equation 4 is an M-matrix if the anisotropic ratio of the quadtree is smaller than or equal to 2, and invertible if a Dirichlet boundary condition is imposed on at least one point.*

Proof By the assumption on the anisotropic the ratio, $s_5 + s_6 \leq 2s_4$. Since $s_5 s_6 \leq \frac{(s_5 + s_6)^2}{4}$, we have $0 \leq \frac{s_5 s_6}{(s_1 + s_4) \cdot s_4} \leq \frac{s_5 s_6}{s_4^2} \leq 1$, and $0 \leq w \leq 1$. Since w is nonnegative, all the coefficients of u_0 in Equation 4 have the same sign. Since every coefficient is multiplied equally to u_0 and its neighbors, the matrix is diagonally dominant. If a Dirichlet boundary condition is imposed at one node, then the linear system is strictly diagonally dominant at the node. By the Gerschgorin's Circle Theorem, the linear system is nonsingular. ■

5.3 Three Spatial Dimensions

In the case of three spatial dimensions, a cell C containing a node u_0 , has three neighboring cells C_x , C_y and C_z in x , y and z directions, respectively that also contain u_0 . Since C , C_x , C_y and C_z include the same node u_0 , either C_x or C_y or C_z should be edge-aligned with C , and one of the other two should be face-aligned with C as depicted in Fig. 7. In this case, two intermediate values u_4 and u_5 are first linearly interpolated from the neighboring nodes:

$$u_4 = \frac{u_7 s_8 + u_8 s_7}{s_8 + s_7},$$

and

$$u_5 = \frac{u_{11} s_{11} s_{12} + u_{12} s_{11} s_9 + u_9 s_{10} s_{12} + u_{10} s_{10} s_9}{(s_{11} + s_{10})(s_9 + s_{12})},$$

with u_i introduced in Fig. 7. Applying the one dimensional finite difference described in Section 5.1 in each spatial direction at u_0 gives the following truncation error:

$$\begin{aligned} \left(\frac{u_1 - u_0}{s_1} + \frac{u_4 - u_0}{s_4} \right) \frac{2}{s_1 + s_4} &= u_{xx} + u_{zz} \frac{s_7 s_8}{s_4 (s_1 + s_4)} + O(h), \\ \left(\frac{u_2 - u_0}{s_2} + \frac{u_5 - u_0}{s_5} \right) \frac{2}{s_2 + s_5} &= u_{yy} + u_{xx} \frac{s_9 s_{12}}{s_5 (s_2 + s_5)} \end{aligned} \quad (5)$$

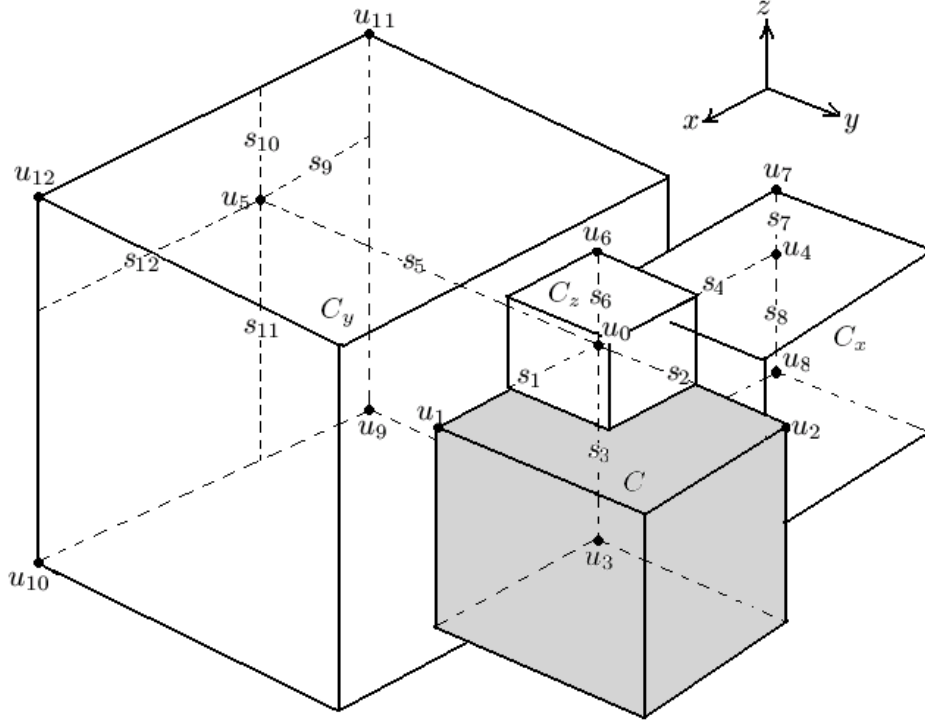


Figure 7: General three dimensional configuration: There are three neighboring cells C_x , C_y and C_z of the shaded cell C that include the node u_0 ; here C_x is face-aligned and C_z is edge-aligned.

$$+u_{zz} \frac{s_{10}s_{11}}{s_5(s_2+s_5)} + O(h), \quad (6)$$

$$\left(\frac{u_3 - u_0}{s_3} + \frac{u_6 - u_0}{s_6} \right) \frac{2}{s_3 + s_6} = u_{zz} + O(h). \quad (7)$$

As mentioned above, the scheme will be second order accurate if the spurious terms generated by the interpolations (the term in u_{zz} in equation (5) and the terms in u_{xx} and u_{zz} in equation (6)) are cancelled. This is achieved by a simple linear weighting of equations (5), (6) and (7):

$$\begin{pmatrix} \frac{u_1 - u_0}{s_1} + \frac{u_4 - u_0}{s_4} \\ \frac{u_2 - u_0}{s_2} + \frac{u_5 - u_0}{s_5} \\ \frac{u_3 - u_0}{s_3} + \frac{u_6 - u_0}{s_6} \end{pmatrix} \begin{pmatrix} \frac{2}{s_1 + s_4} \cdot \alpha \\ \frac{2}{s_2 + s_5} \\ \frac{2}{s_3 + s_6} \cdot \beta \end{pmatrix} = \Delta u + O(h), \quad (8)$$

with

$$\begin{aligned} \alpha &= 1 - \frac{s_{10}s_{11}}{s_5(s_2+s_5)}, \\ \beta &= 1 - \frac{s_9s_{12}}{s_5(s_2+s_5)} - \alpha \frac{s_7s_8}{s_4(s_1+s_4)}. \end{aligned}$$

Again, we note that this discretization reduces to the standard second order accurate scheme for the Laplace operator in the case where the grid is locally uniform, i.e. $s_1 = s_4$, $s_2 = s_5$, $s_3 = s_6$, $\alpha = 1$, $\beta = 1$. The linear system obtained is non singular on isotropic grid and even when relaxing slightly the condition on the anisotropic ratio:

Theorem 3 *The matrix induced by equation 8 is an M-matrix, if the anisotropic ratio of the octree is smaller than or equal to $\sqrt{2}$, and invertible if a Dirichlet boundary condition is imposed on at least one point.*

Proof By the assumption on the ratio, $s_{10} + s_{11} \leq \sqrt{2}s_5$. Since $s_{10}s_{11} \leq \frac{(s_{10}+s_{11})^2}{4}$, we have $0 \leq \frac{s_{10}s_{11}}{s_5(s_2+s_5)} \leq \frac{s_{10}s_{11}}{s_5^2} \leq \frac{1}{2}$, and $\frac{1}{2} \leq \alpha \leq 1$. In the same way, $0 \leq \frac{s_9s_{12}}{s_5(s_2+s_5)} \leq \frac{1}{2}$ and $0 \leq \frac{s_7s_8}{s_4(s_1+s_4)} \leq \frac{1}{2}$. Therefore $0 \leq \beta \leq 1$. Since α and β are nonnegative, all the coefficients of u_0 have the same sign. Since every coefficient is multiplied equally to u_0 and its neighbors, the matrix is diagonally dominant. If a Dirichlet boundary condition is imposed on one node, then the linear system is strictly diagonally dominant at the node. By the Gerschgorin's Circle Theorem, the linear system is nonsingular. ■

6 Variable Coefficient Poisson Equation

Consider again a Cartesian domain $\Omega \in \mathbb{R}^n$, $n = 1, 2, 3$ with boundary $\partial\Omega$ and the variable coefficient Poisson equation $\nabla \cdot (\rho \nabla u) = f$ on Ω with at least one Dirichlet boundary condition $u|_{\partial\Omega} = g$. We assume that the variable coefficient ρ is positive and bounded from below by some $\epsilon > 0$. The numerical methods described in Section 5 can be extended to the case of the variable Poisson equation to produce supra-convergent schemes on fully adaptive meshes. Moreover, the extension to the variable density does not increase the support of the schemes (i.e. the number of cells involved in the discretizations), producing schemes that are straightforward to implement in two and three spatial dimensions. Finally, just as in the case of the Laplace operator, the matrices associated with these discretizations are also M-matrices.

6.1 One Spatial Dimension

With the notation of Section 5.1, the standard discretization of the one-dimensional Poisson equation is

$$\left(\frac{u_{i-1} - u_i}{s_{i-\frac{1}{2}}} \cdot \frac{\rho_{i-1} + \rho_i}{2} + \frac{u_{i+1} - u_i}{s_{i+\frac{1}{2}}} \cdot \frac{\rho_{i+1} + \rho_i}{2} \right) \cdot \frac{2}{s_{i-\frac{1}{2}} + s_{i+\frac{1}{2}}} = f_i,$$

and yields second order accuracy.

6.2 Two Spatial Dimensions

The discretization described in Section 6.1 can be applied in a dimension by dimension framework and is sufficient to obtain second order accurate discretizations in the case where the nodes used in the discretization are aligned. In two spatial dimensions however, nodes are no longer necessarily aligned (see, e.g. Fig. 6) and a procedure similar to that introduced in the case of the Laplace operator must be applied in order to define a valid intermediate value. As with the Laplace discretization, we involve the two nodes u_5 and u_6 and their corresponding densities ρ_5 and ρ_6 . The discretizations for $(\rho u_x)_x$ and $(\rho u_y)_y$ along with their Taylor analysis are given respectively by

$$\left(\frac{u_1 - u_0}{s_1} \cdot \frac{\rho_1 + \rho_0}{2} + \frac{s_6 D_5 + s_5 D_6}{s_5 + s_6} \right) \cdot \frac{2}{s_1 + s_4} = (\rho u_x)_x + \frac{s_5 s_6}{(s_1 + s_4) s_4} (\rho u_y)_y + O(h), \quad (9)$$

and

$$\left(\frac{u_2 - u_0}{s_2} \cdot \frac{\rho_2 + \rho_0}{2} + \frac{u_3 - u_0}{s_3} \cdot \frac{\rho_3 + \rho_0}{2} \right) \cdot \frac{2}{s_2 + s_3} = (\rho u_y)_y + O(h), \quad (10)$$

with

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

Next, we cancel the spurious term $\frac{s_5 s_6}{(s_1 + s_4) s_4} (\rho u_y)_y$ by weighting appropriately equations (9) and (10):

$$\begin{aligned} &\left(\frac{u_1 - u_0}{s_1} \cdot \frac{\rho_1 + \rho_0}{2} + \frac{s_6 a_5 + s_5 a_6}{s_5 + s_6} \right) \cdot \frac{2}{s_1 + s_4} + \\ &\left(\frac{u_2 - u_0}{s_2} \cdot \frac{\rho_2 + \rho_0}{2} + \frac{u_3 - u_0}{s_3} \cdot \frac{\rho_3 + \rho_0}{2} \right) \cdot \frac{2}{s_2 + s_3} \cdot \left(1 - \frac{s_5 s_6}{(s_1 + s_4) s_4} \right) = f_0 + O(h). \end{aligned} \quad (11)$$

Theorem 4 *The matrix induced by Equation 11 is an M-matrix if the anisotropic ratio of the quadtree is smaller than or equal to 2, and invertible if at least one Dirichlet boundary condition is imposed.*

Proof The proof is a straightforward extension of the proof of Theorem 5.2. ■

6.3 Three Spatial Dimensions

The same process can be applied in three spatial dimensions. For example, referring to Fig. 7, the finite difference derived in one spatial dimension is applied dimension by dimension. If a neighboring point is not aligned in the principal x , y or z direction, a multilinear interpolation similar to that introduced in section 6.2 is performed on the finite difference approximations of ρu_x , ρu_y , and ρu_z :

$$\begin{aligned} \left(\frac{\rho_1 + \rho_0}{2} \cdot \frac{u_1 - u_0}{s_1} + D_4 \right) \frac{2}{s_1 + s_4} &= (\rho u_x)_x + \frac{s_7 s_8}{s_4(s_1 + s_4)} (\rho u_z)_z + O(h), \\ \left(\frac{\rho_2 + \rho_0}{2} \cdot \frac{u_2 - u_0}{s_2} + D_5 \right) \frac{2}{s_2 + s_5} &= (\rho u_y)_y + \frac{s_9 s_{12}}{s_5(s_2 + s_5)} (\rho u_x)_x + \frac{s_{10} s_{11}}{s_5(s_2 + s_5)} (\rho u_z)_z + O(h), \\ \left(\frac{\rho_3 + \rho_0}{2} \cdot \frac{u_3 - u_0}{s_3} + \frac{\rho_6 + \rho_0}{2} \cdot \frac{u_6 - u_0}{s_6} \right) \frac{2}{s_3 + s_6} &= (\rho u_z)_z + O(h), \end{aligned}$$

where

$$\begin{aligned} D_4 &= \frac{\rho_7 + \rho_0}{2} \cdot \frac{u_7 - u_0}{s_4} \cdot \frac{s_8}{s_8 + s_7} \\ &+ \frac{\rho_8 + \rho_0}{2} \cdot \frac{u_8 - u_0}{s_4} \cdot \frac{s_7}{s_8 + s_7}, \end{aligned}$$

and

$$\begin{aligned} D_5 &= \frac{\rho_{11} + \rho_0}{2} \cdot \frac{u_{11} - u_0}{s_5} \cdot \frac{s_{11} s_{12}}{(s_{11} + s_{10})(s_9 + s_{12})} \\ &+ \frac{\rho_{12} + \rho_0}{2} \cdot \frac{u_{12} - u_0}{s_5} \cdot \frac{s_{11} s_9}{(s_{11} + s_{10})(s_9 + s_{12})} \\ &+ \frac{\rho_9 + \rho_0}{2} \cdot \frac{u_9 - u_0}{s_5} \cdot \frac{s_{10} s_9}{(s_{11} + s_{10})(s_9 + s_{12})} \\ &+ \frac{\rho_{10} + \rho_0}{2} \cdot \frac{u_{10} - u_0}{s_5} \cdot \frac{s_{10} s_{12}}{(s_{11} + s_{10})(s_9 + s_{12})}. \end{aligned}$$

Next, a linear combination of these equations allows for the cancellation of the spurious terms $\frac{s_7 s_8}{s_4(s_1 + s_4)} (\rho u_z)_z$, $\frac{s_9 s_{12}}{s_5(s_2 + s_5)} (\rho u_x)_x$ and $\frac{s_{10} s_{11}}{s_5(s_2 + s_5)} (\rho u_z)_z$, giving the following consistent approximation:

$$\begin{aligned} &\left(\frac{\rho_1 + \rho_0}{2} \cdot \frac{u_1 - u_0}{s_1} + D_4 \right) \frac{2}{s_1 + s_4} \cdot \alpha + \\ &\left(\frac{\rho_2 + \rho_0}{2} \cdot \frac{u_2 - u_0}{s_2} + D_5 \right) \frac{2}{s_2 + s_5} + \\ &\left(\frac{\rho_3 + \rho_0}{2} \cdot \frac{u_3 - u_0}{s_3} + \frac{\rho_6 + \rho_0}{2} \cdot \frac{u_6 - u_0}{s_6} \right) \frac{2}{s_3 + s_6} \cdot \beta = f_0 + O(h), \end{aligned} \quad (12)$$

with

$$\begin{aligned} \alpha &= 1 - \frac{s_{10} s_{11}}{s_5(s_2 + s_5)}, \\ \beta &= 1 - \frac{s_9 s_{12}}{s_5(s_2 + s_5)} - \alpha \frac{s_7 s_8}{s_4(s_1 + s_4)}. \end{aligned} \quad (13)$$

Theorem 5 *The matrix induced by Equation 12 and Equation 13 is an M-matrix if the anisotropic ratio of the octree is smaller than or equal to $\sqrt{2}$, and invertible if at least one Dirichlet boundary condition is imposed.*

Proof The proof is a straightforward extension of the proof of Theorem 5.2. ■

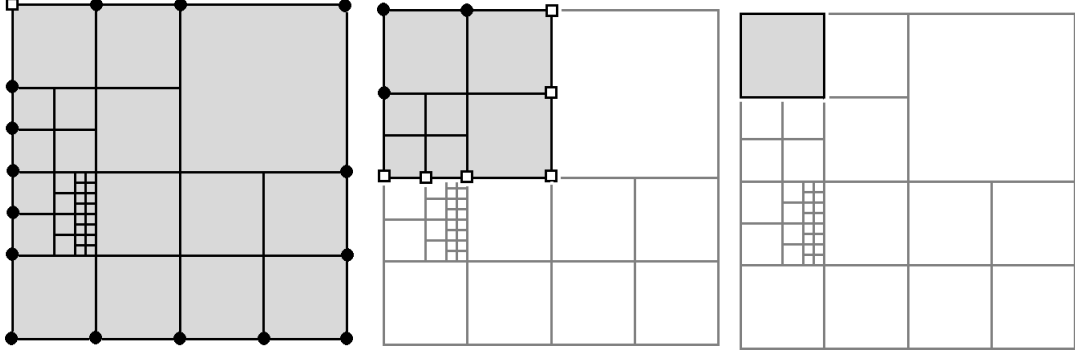


Figure 8: The three steps used to solve Poisson equation with Neumann boundary conditions. Here \square denotes a Dirichlet boundary condition, and \bullet denotes a Neumann boundary condition.

7 A Note on Neumann Boundary Conditions

In the case where Neumann boundary conditions are imposed on the entire domain's boundary, the system of equations constructed with equation 4 or equation 8 is singular since the addition of a constant is also a solution. However, in such a case, it is sufficient to be able to identify one solution since one is only interested in the solution's gradients in the case of "all Neumann boundary conditions" (see for example incompressible flows).

We propose a simple three-step procedure that yields solutions with second order accurate gradients: First, we impose a Dirichlet boundary condition at one corner of the domain (see Fig. 8 [left]) making the linear system nonsingular. Numerical experiments demonstrate that the solution is second order accurate, but that the solution's gradients are only first order accurate near the corner because the artificial Dirichlet boundary condition corrupts the accuracy of the gradients. A correction step follows to recover second order accuracy in the solution's gradients. For this, the Poisson equation is solved again, but only in a small portion of the domain containing the corner. For example, the shaded region in Fig. 8 (center) represents the portion of the domain that is solved with Dirichlet boundary conditions at the bottom and right edges using the solution found in the first step. In the third step, gradients at the nodes of the cell near the corner are updated using the solution of the second step (see Fig. 8 [right]). Numerical experiments demonstrate that the gradients are second order accurate in the L^∞ norm. We note that the cost of solving the Poisson equation in the second step is negligible since it only uses a small portion of the domain.

8 Computing Second Order Accurate Gradients

The gradients of the solution are calculated by a weighted average of the forward and backward differences. In the case where the neighboring nodes are not aligned with the principal axes, intermediate values are linearly interpolated, as discussed in section 5.2 and 5.3. The spurious error caused by the interpolation are successfully removed as detailed next.

8.1 Two Spatial Dimensions

Following the notations in figure 6, the gradient at u_0 is calculated by the following finite differences:

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

where

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

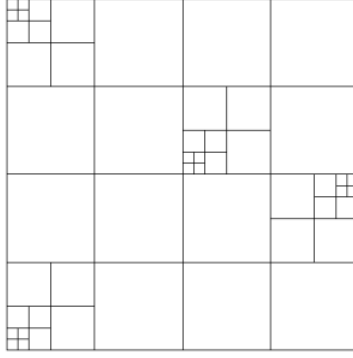


Figure 9: Domain $\Omega = [0, \pi]^2$ and original mesh used in example 9.1.1.

Note that the calculation of the gradient involves the same cells as those used in the discretization of the Laplace or the poisson equations, hence preserving the locality and ease of implementation of the method.

8.2 Three Spatial Dimensions

Following the notations of figure 7, the gradient at u_0 is calculated by the following finite differences:

$$\begin{aligned} u_x &= \frac{u_1 - u_0}{s_1} \cdot \frac{s_4}{s_1 + s_4} + \frac{u_0 - u_4}{s_4} \cdot \frac{s_1}{s_1 + s_4} + \frac{s_7 s_8}{2 s_4} \cdot \frac{s_1}{s_1 + s_4} \cdot u_{zz}, \\ u_y &= \frac{u_2 - u_0}{s_2} \cdot \frac{s_5}{s_2 + s_5} + \frac{u_0 - u_5}{s_5} \cdot \frac{s_2}{s_2 + s_5} + \frac{s_9 s_{12}}{2 s_5} \cdot \frac{s_2}{s_2 + s_5} \cdot u_{xx} + \frac{s_{10} s_{11}}{2 s_5} \cdot \frac{s_2}{s_2 + s_5} \cdot u_{zz}, \\ u_z &= \frac{u_6 - u_0}{s_6} \cdot \frac{s_3}{s_3 + s_6} + \frac{u_0 - u_3}{s_3} \cdot \frac{s_6}{s_3 + s_6}, \end{aligned}$$

The finite differences for u_{xx} , u_{yy} , and u_{zz} are given in section 5.3. Again, the calculation of the gradients involve the same cells that used in the discretization of the Laplace or the poisson equations.

9 Examples

In this section we present numerical evidence that confirms that the schemes described in this paper produce second order accuracy in the L^∞ norm for both the solution and its gradients, on highly irregular grids. In particular the difference of level between adjacent cells can be greater than one, illustrating the fact that the method is supra-convergent on *fully* adaptive meshes. The linear systems of equations are solved using the stabilized bi-conjugate gradient method with the incomplete LU preconditioner [27]. We note that, although other numerical algebra solvers could be used (GMRES, multigrid, etc.), we have not investigated this avenue at this time.

9.1 Laplace Equation

9.1.1 Isotropic Grid in Two Spatial Dimensions

Consider a domain $\Omega = [0, \pi]^2$ and u satisfying $\Delta u = f$ on Ω with an exact solution of $u(x, y) = e^{-x-y}$. We impose Dirichlet boundary conditions on $\partial\Omega$. Fig. 9 depicts the grid used and Table 1 demonstrates the second order accuracy of the solution and its gradients in the L^∞ norm. We note that the grid is highly non regular with a difference of level equal to 3 for some cells.

9.1.2 Anisotropic Grid in Two Spatial Dimensions

This example illustrates the fact that our method is supra-convergent even in the case where the ratio between the width and height of the original cell is greater than one. We test our method on the anisotropic

Effective Resolution	$\ u - u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	order
32^2	7.04×10^{-3}	—	3.42×10^{-2}	—
64^2	1.74×10^{-3}	2.01	1.41×10^{-2}	1.28
128^2	3.97×10^{-4}	2.13	4.38×10^{-3}	1.68
256^2	9.33×10^{-5}	2.09	1.18×10^{-3}	1.89
512^2	2.26×10^{-5}	2.05	3.09×10^{-4}	1.93

Table 1: Error analysis for example 9.1.1.

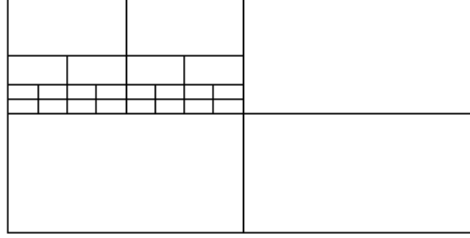


Figure 10: Domain $\Omega = [0, 2] \times [0, 1]$ and original mesh used in example 9.1.2.

grid of Fig. 10, with $\Omega = [0, 2] \times [0, 1]$. We take an exact solution of $u(x, y) = \sin(x)\sin(y)$ and impose Dirichlet boundary conditions. Table 2 illustrates the second order accuracy for the solution and its gradients in the L^∞ norm.

9.1.3 Comparisons between Adaptive Grid and Uniform Grid

Consider a domain $\Omega = [.1, 1]^2$ and u satisfying $\Delta u = f$ on Ω , with an exact solution of $u(r, \theta) = \sin(\frac{1}{r})$. We impose Dirichlet boundary conditions on $\partial\Omega$. Fig. 11 depicts the adaptive grid used and the exact solution. In particular, we chose to refine the grid near the regions where the solution presents large gradients. Tables 3 and 4 give the results in the L^∞ norm for the solution and its gradients in the case of the uniform and adaptive grids, respectively. In particular, we observe that the accuracy on the solution's gradients is slightly superior in the case of the adaptive grid with 1537 nodes than in the case of the uniform grid with 66049 nodes. Likewise, for about the same number of nodes, the accuracy in the case of the adaptive grid is significantly better than that of the uniform grid (10^{-3} .vs. 10^{-1}). Finally, the adaptive solution attains a converged regime with much less nodes than that of the uniform grid (121 nodes .vs. 4000 nodes).

9.1.4 Non Rectangular Domain

This example illustrates the fact that our method is supra-convergent in the case where the domain does not have a rectangular shape. Consider a domain Ω depicted in Fig. 12 and an exact solution $u(x, y) =$

Effective Resolution	$\ u - u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	order
16^2	1.49×10^{-2}	—	8.03×10^{-2}	—
32^2	3.44×10^{-3}	2.11	2.36×10^{-2}	1.76
64^2	7.36×10^{-4}	2.22	8.49×10^{-3}	1.47
128^2	1.79×10^{-4}	2.04	2.48×10^{-3}	1.77
256^2	4.37×10^{-5}	2.03	6.64×10^{-4}	1.90
512^2	1.07×10^{-5}	2.03	1.71×10^{-4}	1.95

Table 2: Error analysis for example 9.1.2.

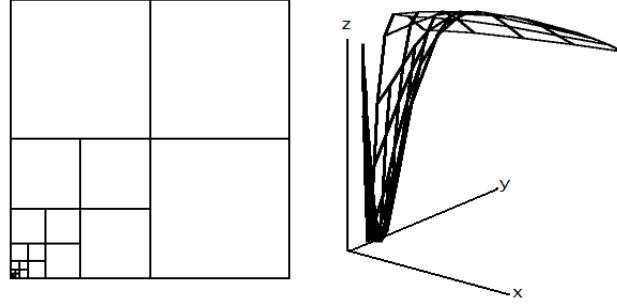


Figure 11: Left: Domain $\Omega = [0,1]^2$ with the original mesh refined near the origin where the solution presents stiff gradients and exact solution (right) used in example 9.1.3.

Number of Nodes	$\ u - u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	order
81	6.97×10^{-2}	—	1.02×10^1	—
289	1.78×10^{-2}	1.96	1.93×10^1	-.91
1089	1.53×10^{-2}	.21	1.17×10^1	.72
4225	4.31×10^{-3}	1.83	3.92×10^0	1.57
16641	1.11×10^{-3}	1.95	1.06×10^0	1.88
66049	2.78×10^{-4}	1.98	2.71×10^{-1}	1.97

Table 3: Error analysis for example 9.1.3 illustrating the limitations of a uniform grid to approximate stiff solutions.

Number of Nodes	$\ u - u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	order
39	8.66×10^{-2}	—	5.47×10^0	—
121	2.52×10^{-2}	1.77	1.78×10^0	1.63
417	7.13×10^{-3}	1.82	5.70×10^{-1}	1.63
1537	1.92×10^{-3}	1.89	1.77×10^{-1}	1.68
5889	4.97×10^{-4}	1.95	5.33×10^{-2}	1.73
23041	1.26×10^{-4}	1.97	1.55×10^{-2}	1.77
91137	3.18×10^{-5}	1.98	4.46×10^{-3}	1.80

Table 4: Error analysis for example 9.1.3 in the case of adaptive mesh refinement. The computational resources are focused on the regions of steep gradients.

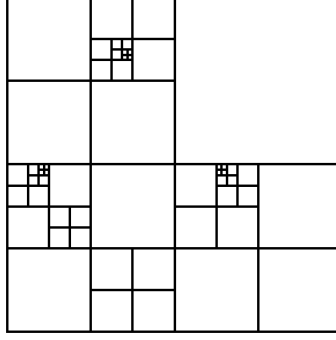


Figure 12: Domain Ω and original mesh used in example 9.1.4.

Effective Resolution	$\ u - u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	order
32^2	3.43×10^{-2}	—	1.71×10^{-1}	—
64^2	8.46×10^{-3}	2.02	5.44×10^{-2}	1.65
128^2	2.01×10^{-3}	2.07	1.44×10^{-2}	1.91
256^2	4.83×10^{-4}	2.05	3.63×10^{-3}	1.99
512^2	1.19×10^{-4}	2.02	9.10×10^{-4}	2.00

Table 5: Error analysis for example 9.1.4.

$\sin(x)\sin(y)$. Dirichlet boundary conditions are imposed on the domain's boundary. Table 5 illustrates the second order accuracy for the solution and its gradients in the L^∞ norm.

9.1.5 Neumann Boundary Condition

Consider a domain $\Omega = [0, \pi]^2$ and an exact solution $u(x, y) = \cos(x)\cos(y) - 1$. We impose Neumann boundary conditions on the boundary of the domain shown in Fig. 13. We apply the procedure described in Section 7 by first imposing a Dirichlet boundary condition at the origin. Table 6 demonstrates the second order accuracy of the solution in the L^∞ norm prior to fixing the localized errors. Fig. 13 depicts the error in the solution's gradients, illustrating a peak localized near the origin (center) that is removed (right) with the procedure described in section 7. Table 7 demonstrates the accuracy of the solution's gradients in the L^∞ norm, before and after fixing the localized errors.

9.1.6 Three Spatial Dimensions - Dirichlet boundary condition

Consider a domain $\Omega = [0, 1]^3$ and an exact solution $u(x, y, z) = e^{-x-y-z}$. We impose Dirichlet boundary conditions on $\partial\Omega$. Fig. 14 depicts the grid used and table 8 demonstrates the second order accuracy of the

Effective Resolution	$\ u - u_h\ _\infty$	Order
32^2	3.15×10^{-1}	—
64^2	7.86×10^{-2}	2.00
128^2	2.05×10^{-2}	1.94
256^2	5.46×10^{-3}	1.90
512^2	1.48×10^{-3}	1.89

Table 6: Error analysis for example 9.1.5.

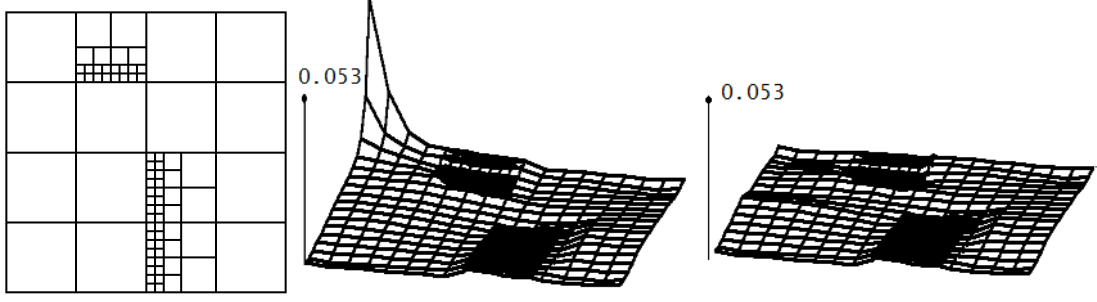


Figure 13: Domain $\Omega = [0, \pi]^2$ with the original mesh used in example 9.1.5 (left), plot of $\|\nabla u - \nabla u_h\|_\infty$ before (center) and after (right) applying the procedure described in section 7.

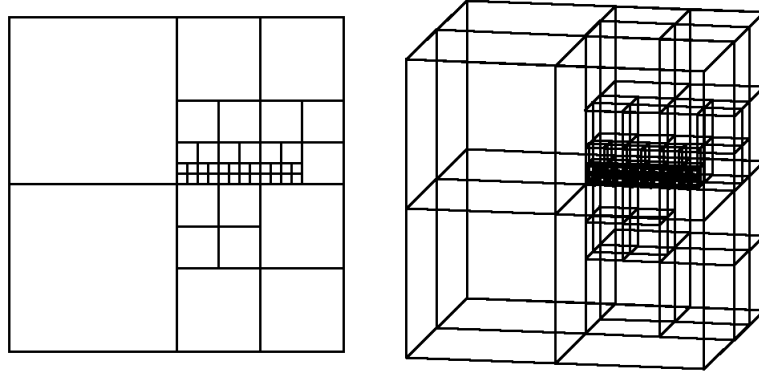


Figure 14: Domain $\Omega = [0, 1]^3$, the front (left) and side (right) views of the original mesh used in example 9.1.6.

solution and its gradients in the L^∞ norm.

9.1.7 Three Spatial Dimensions - Neumann Boundary Condition

Consider a domain $\Omega = [0, \pi]^3$ and an exact solution $u(x, y, z) = \cos(x) \cos(y) \cos(z) - 1$. We impose Neumann boundary conditions on the boundary of the domain depicted in Fig. 15. The procedure described in section 7 is applied by first imposing a Dirichlet boundary condition at the origin, then removing the spurious localized errors in gradients as explained in section 7. Table 9 demonstrates the second order accuracy in the solution's gradients in the L^∞ norm.

9.2 Variable Coefficient Poisson Equation

9.2.1 Two Spatial Dimensions

Consider the Poisson equation $\nabla \cdot (\rho \nabla u) = f$ on a domain $\Omega = [0, \pi]^2$ with $\rho(x, y) = \sin(x) \sin(y) + 2$ and an exact solution of $u(x, y) = \sin(x) + \sin(y)$. Dirichlet boundary conditions are imposed on the boundary $\partial\Omega$. Fig. 9 illustrates the original mesh and Table 10 demonstrates the second order accuracy of the solution and its gradients in the L^∞ norm.

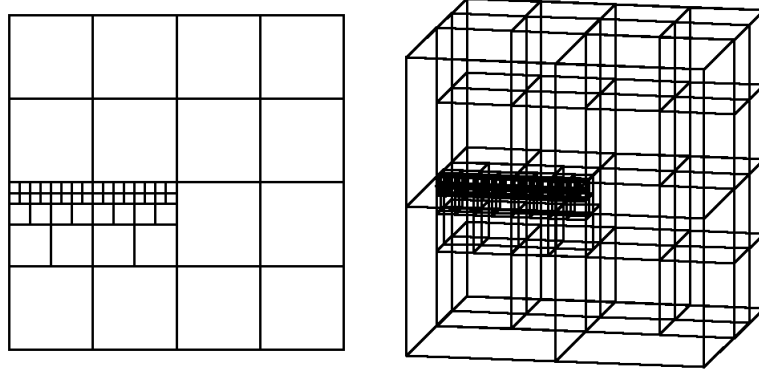


Figure 15: Domain $\Omega = [0, \pi]^3$, the front (left) and side (right) views of original mesh used in example 9.1.7.

9.2.2 Three Spatial Dimensions

Consider the Poisson equation $\nabla \cdot (\rho \nabla u) = f$ on a domain $\Omega = [0, \pi]^2$ with $\rho(x, y, z) = \sin(x + y + z) + 2$ and an exact solution of $u(x, y, z) = e^{-x^2 - y^2 - z^2}$. Dirichlet boundary conditions are imposed on the boundary $\partial\Omega$. Fig. 15 illustrates the original mesh and Table 11 demonstrates the second order accuracy of the solution and its gradients in the L^∞ norm.

10 Conclusion

We have proposed a finite difference algorithm for the Poisson equation that yields second order accuracy for the solutions and their gradients on fully adaptive grids. Sampling the solution at the nodes produces an efficient algorithm that can be applied in a dimension by dimension framework. At T-junctions, multilinear interpolations are used to generate intermediate values used in the discretizations. These intermediate values introduce spurious $O(1)$ errors that are successfully removed by simple weighting. Notably, the discretization at the node of one cell only involves nodes of two (in 2D) or three (in 3D) adjacent cells, yielding a method straightforward to implement. The linear systems obtained are nonsymmetric but are shown to be diagonally dominant. We have presented two- and three-dimensional results to demonstrate the second order accuracy of the method in the L^∞ norm for the solutions and their gradients. This method will serve as the basis for a second order accurate method to solve the Poisson and Heat equations on irregular domains and on fully adaptive grids. Future work will include the design of a simple second order accurate schemes for the Stefan problem and the incompressible Navier-Stokes equations on fully adaptive grids.

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 P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1989.
- [6] M. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.
- [7] H. D. Ceniceros and A. M. Roma. Study of long-time dynamics of a viscous vortex sheet with a fully adaptive non-stiff method. *Phys. Fluids*, 16(12):4285–4318, 2004.
- [8] A. Chorin. A Numerical Method for Solving Incompressible Viscous Flow Problems. *J. Comput. Phys.*, 2:12–26, 1967.
- [9] P. Ciarlet. *Introduction to Numerical Linear and Optimization*. Cambridge Texts in Applied Mathematics, 40 West 20th street, New York, NY 10011, 1998.
- [10] 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.
- [11] 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.
- [12] G. Golub and C. Loan. *Matrix Computations*. The John Hopkins University Press, 1989.
- [13] F. Ham, F. Lien, and A. Strong. A fully conservative second-order finite difference scheme for incompressible flow on nonuniform grids. *J. Comput. Phys.*, 117:117–133, 2002.
- [14] 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.
- [15] D. Kincaid and W. Cheney. *Numerical Analysis: Mathematics of Scientific Computing*. Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 2002.
- [16] H.O. Kreiss, H.-O. Manteuffel, T.A. Schwartz, B. Wendroff, and A.B. White Jr. Supra-convergent schemes on irregular grids. *Math. Comp.*, 47:537–554, 1986.
- [17] 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.
- [18] Z. Li. A fast iterative algorithm for elliptic interface problems. *SIAM J. Numer. Anal.*, 35:230–254, 1998.
- [19] 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.
- [20] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids (in press)*.
- [21] 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.
- [22] T. Manteuffel and A. White. The numerical solution of second-order boundary value problems on nonuniform meshes. *Math. Comput.*, 47 (176):511–535, 1986.
- [23] A. Mayo. The fast solution of poisson’s and the biharmonic equations on irregular regions. *SIAM J. Numer. Anal.*, 21:285–299, 1984.
- [24] 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.
- [25] A. McKenney and L. Greengard. A fast poisson solver for complex geometries. *J. Comput. Phys.*, 118:348–355, 1995.

- [26] S. Popinet. Gerris: A tree-based adaptive solver for the incompressible euler equations in complex geometries. *J. Comput. Phys.*, 190:572–600, 2003.
- [27] Y. Saad. *Iterative methods for sparse linear systems*. PWS Publishing, 1996. New York, NY.
- [28] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, New York, 1989.
- [29] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS*. Addison-Wesley, New York, 1990.
- [30] M. Sussman, A. S. Algre, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. An adaptive level set approach for incompressible two-phase flow. *J. Comput. Phys*, 148:81–124, 1999.
- [31] D. Young, R. Melvin, M. Bieterman, F. Johnson, S. Samant, and J. Bussioletti. A locally refined rectangular grid finite element method: Application to computational fluid dynamics and computational physics. *J. Comput. Phys.*, 92:1–66, 1991.

	Without Treating the Localized Errors		Treating the Localized Errors	
Effective Resolution	$\ \nabla u - \nabla u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	Order
32^2	4.66×10^{-1}	–	1.68×10^{-1}	–
64^2	1.46×10^{-1}	1.68	2.12×10^{-2}	2.99
128^2	5.80×10^{-2}	1.33	5.08×10^{-3}	2.06
256^2	2.63×10^{-2}	1.14	1.30×10^{-3}	1.96
512^2	1.26×10^{-2}	1.06	3.35×10^{-4}	1.96

Table 7: Error analysis for example 9.1.5.

Effective Resolution	$\ u - u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	order
32^3	3.22×10^{-3}	–	5.82×10^{-2}	–
64^3	7.03×10^{-4}	2.19	1.73×10^{-2}	1.75
128^3	1.82×10^{-4}	1.95	4.75×10^{-3}	1.87
256^3	4.47×10^{-5}	2.02	1.24×10^{-3}	1.93
512^3	1.10×10^{-5}	2.02	3.99×10^{-4}	1.64

Table 8: Error analysis for example 9.1.6.

	Without Treating the Localized Errors		Treating the Localized Errors	
Effective Resolution	$\ \nabla u - \nabla u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	Order
32^2	9.70×10^{-1}	–	9.70×10^{-1}	–
64^2	1.38×10^{-1}	2.82	1.12×10^{-1}	3.12
128^2	1.19×10^{-1}	0.22	2.44×10^{-2}	2.20
256^2	1.05×10^{-1}	0.18	6.34×10^{-3}	1.95
512^2	9.71×10^{-2}	0.11	1.60×10^{-3}	1.99

Table 9: Error analysis for example 9.1.7.

Effective Resolution	$\ u - u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	order
32^2	7.67×10^{-2}	–	3.10×10^{-1}	–
64^2	1.92×10^{-2}	2.00	7.41×10^{-2}	2.06
128^2	4.87×10^{-3}	1.98	1.89×10^{-2}	1.98
256^2	1.24×10^{-3}	1.97	4.74×10^{-3}	1.99
512^2	3.15×10^{-4}	1.98	1.19×10^{-3}	1.97

Table 10: Error analysis for example 9.2.1.

Effective Resolution	$\ u - u_h\ _\infty$	Order	$\ \nabla u - \nabla u_h\ _\infty$	order
32^3	5.02×10^{-2}	–	5.90×10^{-1}	–
64^3	1.25×10^{-2}	2.01	1.42×10^{-1}	2.06
128^3	3.11×10^{-3}	2.01	2.43×10^{-2}	2.54
256^3	8.10×10^{-4}	1.94	6.43×10^{-3}	1.92
512^3	2.06×10^{-4}	1.97	2.00×10^{-3}	1.68

Table 11: Error analysis for example 9.2.2.