

A FINITE VOLUME METHOD FOR SOLVING PARABOLIC EQUATIONS ON LOGICALLY CARTESIAN CURVED SURFACE MESHES*

DONNA A. CALHOUN[†] AND CHRISTIANE HELZEL[‡]

Abstract. We present a second order finite volume scheme for the constant-coefficient diffusion equation on curved parametric surfaces. While our scheme is applicable to general quadrilateral surface meshes based on smooth or piecewise smooth coordinate transformations, our primary motivation for developing the present scheme is to solve diffusion problems on a particular set of circular and spherical meshes introduced in [D. A. Calhoun, C. Helzel, and R. J. LeVeque, *SIAM Rev.*, 50 (2008), pp. 723–752] for the discretization of hyperbolic problems. These grids are generated from mappings of a single Cartesian grid and were designed to have nearly uniform cells sizes and avoid the pole singularity associated with polar or spherical grid mappings. The present method for parabolic equations offers several advantages. It does not require analytic metric terms, shows second order accuracy on our disk and sphere grids, can be easily coupled to existing finite volume solvers for logically Cartesian meshes, and handles general mixed boundary conditions. Our parabolic scheme should appeal to researchers in the fields of geophysical fluid dynamics, computational biology, and any other discipline that requires the solution of parabolic equations on quadrilateral surface meshes. In this article, we present several numerical examples demonstrating the accuracy of the scheme, and then use the scheme to solve advection-reaction-diffusion equations modeling biological pattern formation on surfaces.

Key words. finite volume methods, parabolic PDEs, curved surfaces, pattern formation, Laplace–Beltrami operator

AMS subject classifications. 65M08, 58J35, 35K05

DOI. 10.1137/08073322X

1. Introduction. We present a finite volume method for the solution of parabolic problems on smooth parametric surfaces. The main ingredient of this method is a finite volume discretization of the surface Laplacian on a logically Cartesian surface mesh. While the method can be adapted for use on general quadrilateral grids based on smooth or piecewise smooth mappings, we focus our attention here on a set of mappings for circular and spherical regions that were designed to have nearly uniform cell sizes and to avoid the pole singularities in polar or spherical grid mappings [12]. The chief advantages of our method are that the scheme does not require analytic metric terms and can be easily coupled with existing finite volume hyperbolic solvers for logically Cartesian surfaces meshes. We also show how to include general mixed boundary conditions in the parabolic scheme and how to couple it with advection solvers to solve advection-reaction-diffusion equations on surfaces.

Our main motivation for the present work is to solve diffusion equations on the circular and spherical grids introduced in [12]. These grids have the useful property that they are obtained by a mapping of a single Cartesian grid in computational space and do not use multiblock data structures or unstructured meshes. Hence, existing numerical methods for single logically Cartesian grids, including those which make

*Received by the editors August 20, 2008; accepted for publication (in revised form) July 31, 2009; published electronically November 13, 2009.

<http://www.siam.org/journals/sisc/31-6/73322.html>

[†]Corresponding author. DEN/DM2S/SFME/LTMF, Bt. 454, Commissariat à l’Energie Atomique (C.E.A.), Centre de Saclay, Gif-sur-Yvette 91191, France (donna.calhoun@cea.fr).

[‡]Department of Mathematics, Ruhr-University Bochum, Bochum, Germany (christiane.helzel@rub.de).

use of adaptive mesh refinement, can be easily adapted for use on these grids. In [12], we showed how to solve hyperbolic equations on these circular and spherical grids and demonstrated that we can obtain accurate results for the Euler equations on a disk, the shallow water wave equations on the sphere, and acoustic equations on a mesh with embedded cylinders. The presence of metric discontinuities in these mappings does not appear to degrade the solutions to such problems, suggesting that these mappings may be useful for a variety of hyperbolic problems.

In this paper, we seek a finite volume discretization of the surface Laplacian that complements our hyperbolic solvers cited above and is suitable for solving parabolic problems on surface grids. In [12], we briefly discuss the approximation of a reaction-diffusion equation on the sphere, but in that presentation we rely on analytic metric terms to formulate the finite volume fluxes for the parabolic equations. Since our hyperbolic discretization for quadrilateral surface meshes relies only on quantities that can be easily computed from the physical location of mesh vertices, we seek a discretization for the surface Laplacian that also relies only on the discrete geometry of our mesh. Such a discretization would fit nicely into our framework for solving hyperbolic problems and would complement our existing solvers for PDEs on our disk and sphere grids, making these grids useful to a wide variety of application areas.

Existing literature on discretizations of the surface Laplacian, otherwise known as the Laplace–Beltrami operator, contains many references to methods based on finite element and finite difference discretizations on unstructured triangle meshes. Dziuk, in [20], was the first author to present a finite element discretization of elliptic partial differential equations on surfaces using a triangular mesh and linear elements. In a more recent paper, Dziuk and Elliot used this approach to discretize parabolic equations on surfaces [21]. Pinkall and Polthier constructed an approximation for computing minimal surfaces that led to the well-known “cotan” formula for the Laplacian at vertices of triangular surface meshes [43]. More recent articles from the computer graphics and computer-aided geometric design literature describe similar approaches to discretizing the surface Laplacian [53, 39, 57, 37, 18]. Xu [57] summarizes several such discretizations, including the cotan formula, for triangular meshes. For our purposes, the main drawback to these formulas for triangles is that there appears to be no clear way to extend them to quadrilateral cells. Subdividing the quadrilateral cell into triangles is not a satisfying solution, as the discretization for each quadrilateral mesh cell will depend on how that cell is subdivided. Xu [37] recently addressed this concern by developing a discretization for the Laplace–Beltrami operator on quadrilaterals. However, the purpose of that paper, as with many articles coming from the graphics and geometric design community, is to derive a formally consistent discretization of the differential operator, not necessarily to solve PDEs. As it turns out, this requirement of formal consistency can be much more difficult to satisfy than that of the convergence of solutions to elliptic or parabolic equations. It is known, for example, that finite volume schemes for general meshes may not be formally consistent, yet can still lead to second order convergence results [52, 30, 22].

In another approach to solving parabolic problems on surfaces, one does not assume an underlying surface mesh, but instead views the surface as implicitly defined by a level-set function embedded in three-dimensional space [46, 7, 49, 24, 2, 17]. The parabolic problem is then written in terms of the three space variables, but formulated in such a way that the solution to the three-dimensional problem coincides with the desired solution on the implicitly defined surface. These methods have the obvious advantage that they avoid the need for a surface parametrization and hence can be

used for very general, possibly evolving, surfaces. This method has recently gained popularity in computational biology, where highly convoluted, multiply connected surfaces are common but may not be easily defined or parameterized [49, 42, 48]. Since we assume at the outset, however, that we have a surface parameterization, these methods do not appear to offer us any particular advantages and, in any case, do not couple in an obvious way with our existing hyperbolic solvers for quadrilateral surface meshes.

Finally, there is an active research community dedicated to the development of schemes based on “diamond-cell” approximations and “discrete duality finite volume” (DDFV) principles for solving diffusion problems on unstructured finite volume meshes in two- and three-dimensional Euclidean space [1, 3, 8, 14, 16, 27, 15, 19, 50, 28, 36, 31]. For a comparison of more than 20 of these schemes, see [25]. For many of these schemes, convergence proofs establish first order accuracy, although second order accurate results are usually seen in practice, even on very skewed meshes or for problems with highly anisotropic or discontinuous diffusion tensors [19, 3, 8]. To take advantage of the extensive practical and theoretical results available for these schemes, we could view the problem of surface diffusion as an anisotropic diffusion process in which the diffusion tensor is derived from a coordinate transformation rather than from physical considerations. This is essentially the approach we took in [12], where we solved a reaction-diffusion system on our sphere grid using a diamond-cell scheme on a uniform Cartesian mesh and a diffusion tensor derived analytically from the sphere mapping. Although we are not familiar with other specific efforts to apply diamond-cell or DDFV schemes directly in this way, our experience suggests that these methods should generally work well. A major drawback to this approach is that for general mappings it can be difficult to derive the algebraic expressions needed to define the metric tensor.

In this paper, we take an alternative viewpoint and consider surface diffusion that is physically isotropic (i.e., the principle diffusion directions are aligned with coordinate directions), but make use of the fact that the diamond-cell and DDFV schemes referenced above are specifically designed to provide accurate discretizations for general unstructured meshes. We use as our starting point a semidiscrete finite volume approximation of the Laplace–Beltrami operator on a quadrilateral surface mesh. By replacing the analytic metric terms that appear in this expression with their discrete equivalents, we derive a nine-point stencil for the Laplace–Beltrami operator that requires only the location of mesh cell centers and vertices and in particular does not require surface normals or analytic expressions for the metric. We combine this discretization with an explicit time stepping scheme to solve parabolic equations on our disk and sphere grids, and provide numerical evidence showing that these solutions are second order accurate if we take into account discontinuities in the derivatives of the metric for these grids and reduce to first order if we ignore these discontinuities. The scheme we derive is closely related to the diamond-cell schemes referenced above. We could have based our scheme on the DDFV methods of Hermeline and others [27, 19], but since these schemes require twice as many unknowns as the diamond-cell schemes, their use can be practically justified only as part of an implicit time stepping strategy requiring the solution to a linear system and that can take full advantage of the superior matrix properties that the DDFV methods offer.

A second goal of this paper is to show that the resulting scheme has low computational overhead and can be coupled easily into an existing solver for hyperbolic problems. We avoid solving linear systems by using an explicit Runge–Kutta–Chebyshev

(RKC) solver of Sommeijer, Shampine, and Verwer [51] and show that the RKC scheme requires fewer function calls (or matrix-vector multiplies, in the case of the implicit solver) than either the forward Euler or an implicit method based on BICG-stab. To demonstrate how our scheme can be easily used with existing hyperbolic solvers, we couple our parabolic solver with the wave-propagation algorithms available in CLAWPACK [32] to solve an advection-reaction-diffusion problem on the disk.

In what follows, we first describe our disk and sphere grids and then describe the general fractional step scheme for solving advection-reaction-diffusion problems. In section 3.1.1, we describe our Laplace–Beltrami discretization and follow this with a brief discussion of the advection scheme on surface grids. In section 3.1.2 on implementation details, we briefly describe the RKC method for explicit time stepping and various components needed to achieve second order accuracy, including modifications near metric discontinuities and implementation of general boundary conditions. Then, in section 4, we present an accuracy study for the discretization of parabolic problems on the disk, the hemisphere, and the sphere. Finally, we present test calculations for chemotaxis in a petri dish and for Turing patterns on a surface. In the appendices, we compare our scheme with the cotan formula [43, 20] of Dzuik and others for triangular meshes (Appendix A) and Hermeline’s scheme [27] for flat polygonal meshes (Appendix B).

2. Disk and sphere grid mappings. In this section we describe the grids that are used in our test calculations. These are grids for the circular disk and sphere that were introduced in [12]. We also describe a grid for a surface that can be obtained by a mapping of the sphere grid. All of these grids are constructed to avoid the pole singularity and to have nearly uniform cell sizes. The ratio of largest to smallest grid cell sizes is nearly two and does not increase under grid refinement. An additional advantage of these grids is that they are described by a mapping of a *single* rectangular mesh in computational space. This greatly simplifies the implementation of numerical methods (particularly those which use adaptive mesh refinement (AMR)) and the postprocessing and visualization of the results.

A polar coordinate grid for the disk as well as a longitude-latitude grid for the sphere are, of course, obvious examples of logically Cartesian grids for the disk and sphere and are often used in applications. However, the pole singularity leads to severe time step restriction when using explicit numerical methods which are typically used for the approximation of hyperbolic PDEs.

2.1. Mappings for the disk. In this section, we describe a mapping which maps the computational domain $[-1, 1] \times [-1, 1]$ to the unit disk. To describe the mapping, we focus our attention on the region of the square in which computational coordinates (ξ, η) satisfy $|\xi| \leq \eta$. This region corresponds to the upper triangular region between the two diagonals of the square. We refer to this region as the *north sector* of the computational domain. The mapping in this sector is based on the idea that we can map a horizontal line segment between points $(-d, d)$ and (d, d) , with $d = \max(|\xi|, |\eta|)$, to a circular arc of radius $R(d)$, passing through points $(-D(d), D(d))$ and $(D(d), D(d))$. The center of the circle on which this arc lies is given by $(x_0, y_0) = (0, D(d) - \sqrt{R(d)^2 - D(d)^2})$. A point (ξ, η) in the north sector is mapped to the unit disk by the mappings $X_n(\xi, \eta)$ and $Y_n(\xi, \eta)$ given by

$$(2.1) \quad \left. \begin{aligned} X_n(\xi, \eta) &= D(\eta) \frac{\xi}{\eta}, \\ Y_n(\xi, \eta) &= D(\eta) - \sqrt{R(\eta)^2 - D(\eta)^2} + \sqrt{R(\eta)^2 - X_n(\xi, \eta)^2}, \end{aligned} \right\} \quad |\xi| \leq \eta.$$

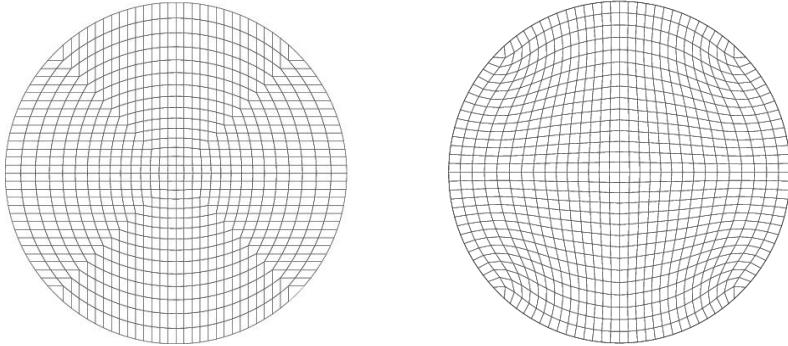


FIG. 2.1. Grids for the circular disk. On the left is a grid for the disk obtained using (2.2). This mapping is not smooth across the two diagonal lines bisecting the circle. On the right is a grid which is optimized with respect to smoothness.

The mapping to other sectors is easily obtained by negating and/or swapping the arguments (ξ, η) or functions (X_n, Y_n) . For example, the mapping in the west sector is given by $X_w(\xi, \eta) = Y_n(\eta, -\xi)$ and $Y_w(\xi, \eta) = -X_n(\eta, -\xi)$. MATLAB scripts for this and other mappings are given in [12].

There are many choices for $R(d)$ and $D(d)$ that lead to useful grids. For example, one choice that is well suited for the unit disk is

$$(2.2) \quad \begin{aligned} D(d) &= \frac{d}{\sqrt{2}}, \\ R(d) &= 1. \end{aligned}$$

This choice leads to the grid on the left in Figure 2.1. Also in Figure 2.1, we show an example of an optimally smooth grid obtained using mesh generation techniques [29]. By comparing this grid with ours, we see that, with the exception of the metric discontinuity along the diagonals, our grid is quite similar to the optimally smooth one, and much simpler to generate. In either case, once the location of the grid cells is known, the same algorithms for the discretization of PDEs can be applied. As we will see, our grids will require simple modifications at metric discontinuities to obtain second order accuracy.

2.2. A mapping for the unit sphere. The above mapping for the unit disk can be used directly to create a grid for the unit hemisphere or sphere. To parameterize the hemisphere, for example, we could simply describe the hemispherical surface as a function of physical locations $(X(\xi, \eta), Y(\xi, \eta))$ from the disk mapping. However, using the formula for $D(d)$ given in (2.2) will lead to very elongated cells near the equator. To remedy this, we seek a $D(d)$ which compresses azimuthal grid lines near the outer edge of the disk mappings. An example of a function $D(d)$ which does this is

$$(2.3) \quad D(d) = \frac{\sin(\pi d/2)}{\sqrt{2}}.$$

The plot at the left in Figure 2.2 shows a grid generated using this $D(d)$, along with $R(d) = 1$.

In general, if we are given a mapping $X(\xi, \eta)$ and $Y(\xi, \eta)$ from the square $[-1, 1] \times [-1, 1]$ to the unit circle, we can define mapping functions $X_s(\xi, \eta)$, $Y_s(\xi, \eta)$, and

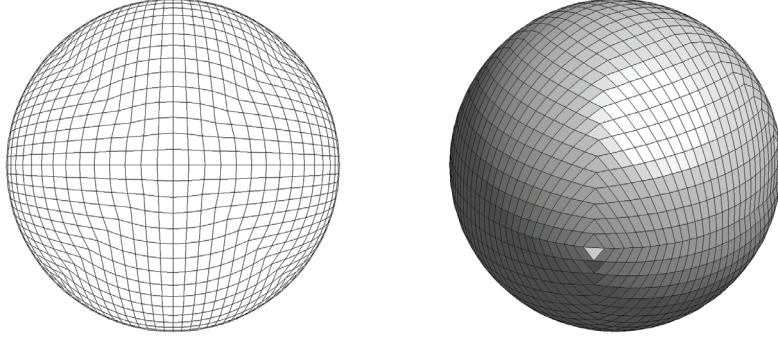


FIG. 2.2. On the left is a grid for the disk which redistributes points near the boundary using (2.3) and which is used to construct the sphere grid shown in the right picture. The mapping used to construct the sphere grid is not smooth at the equator or along the diagonals inherited from the disk mapping.

$Z_s(\xi, \eta)$ from the rectangular region $[-3, 1] \times [-1, 1]$ to the sphere as

$$(2.4) \quad \begin{aligned} X_s(\xi, \eta) &= \begin{cases} X(-(\xi + 2), \eta) & \text{if } |\xi + 2| \leq 1, \\ X(\xi, \eta) & \text{if } |\xi| \leq 1, \end{cases} \\ Y_s(\xi, \eta) &= Y(\xi, \eta), \\ Z_s(\xi, \eta) &= \begin{cases} -\sqrt{1 - (X(-(\xi + 2), \eta)^2 + Y(\xi, \eta)^2)} & \text{if } |\xi + 2| \leq 1, \\ \sqrt{1 - (X(\xi, \eta)^2 + Y(\xi, \eta)^2)} & \text{if } |\xi| \leq 1. \end{cases} \end{aligned}$$

Using functions $X(\xi, \eta)$ and $Y(\xi, \eta)$ based on $D(d)$ as defined in (2.3) leads to the sphere mesh shown in Figure 2.2.

Both our disk and sphere grids parameterize the surface using piecewise smooth mappings from computational space. For the disk and sphere mappings, metric discontinuities occur along the diagonals of the computational grid, and for the sphere mapping, we also have discontinuities along the equator of the sphere. As we will see, it is important to take into account these discontinuities when discretizing parabolic problems.

2.3. A mapping of the hemisphere to a surface. Grids for domains that can be described by a mapping from the sphere (or disk) can easily be constructed as an extension of the grids discussed so far. Here we give an example for a so-called supershape proposed by Gielis [23]. Gielis suggested a class of formulas to study forms in plants and other organisms. Here we describe one particular surface and later show how we can easily apply our numerical scheme to solving reaction-diffusion equations on this interesting shape.

To describe the supershape, we define functions

$$(2.5) \quad r_i(\alpha) = (|\cos(m_i\alpha/4)| + |\sin(m_i\alpha/4)|)^{-1}, \quad i = 1, 2.$$

The surface is obtained by a spherical product of the form

$$(2.6) \quad \begin{aligned} X(\phi, \theta) &= r_1(\theta) \cos \theta r_2(\phi) \cos \phi, \\ Y(\phi, \theta) &= r_1(\theta) \sin \theta r_2(\phi) \cos \phi, \\ Z(\phi, \theta) &= r_2(\phi) \sin \phi, \end{aligned}$$

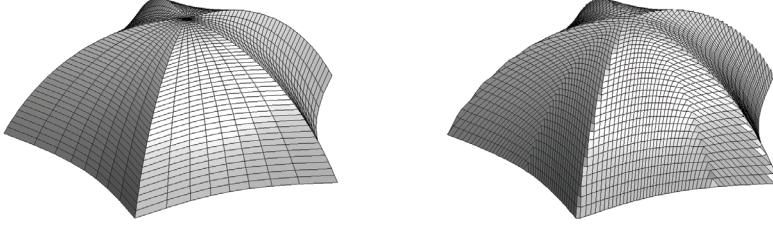


FIG. 2.3. The mesh on the left is based on a longitude-latitude discretization of the surface described by (2.5), (2.6) with $m_1 = 5$, $m_2 = 2.5$. The right plot shows the same surface, constructed from a mapping of our sphere grid. Only the upper half of the supershape mesh is shown.

with latitude $-\pi/2 \leq \phi \leq \pi/2$ and longitude $-\pi \leq \theta \leq \pi$. The parameter values m_i in (2.5) specify the number of symmetric sectors. The particular choice $m_1 = 5$ and $m_2 = 2.5$ leads to the surfaces shown in Figure 2.3. In the left plot of that figure, the surface is discretized using spherical coordinates (ϕ, θ) .

The longitude-latitude discretization shown in Figure 2.3 exhibits the dramatic effects of the pole singularity present when mapping the supershape from spherical coordinates. Using our discretization of the sphere, we can obtain a mesh for this surface in which mesh cells are much more uniformly spaced. To obtain the grid shown in Figure 2.3, we first map each point in the computational space to a hemisphere using our hemisphere mapping, and then map the corresponding (ϕ, θ) coordinates to the supershape. It appears that the supershape grid appears to resolve the ridges much less cleanly than the polar grid. However, the fact that the five ridges on the supershape align exactly with radial coordinate lines of the polar grid depends only on our fortuitous choice of number of radial coordinate lines. In this paper, we do not do anything special to treat the discontinuities in curvature on the supershape.

3. Finite volume methods for advection-reaction-diffusion problems on surfaces. In this section, we describe a numerical method for the approximation of advection-reaction-diffusion equations of the form

$$(3.1) \quad \frac{\partial q}{\partial t} + \nabla \cdot f(q) = \nabla^2 q + G$$

on a surface S embedded in \mathbf{R}^3 . Here $q(\mathbf{x}, t)$ is a vector valued quantity defined on the surface, $f(q)$ is the flux function, and G is a source term. In biological applications, for example, this source term is used to model reactions between interacting species. Throughout this paper, we use $\nabla \cdot$ and ∇^2 to denote the divergence and Laplacian operators on the surface. A flat surface is included as a special case.

We use a fractional step approach in which we split the approximation of (3.1) into subproblems which we solve in an alternating fashion. These two subproblems, solved one after the other, are

$$(3.2) \quad \frac{\partial q}{\partial t} + \nabla \cdot f(q) = 0,$$

$$(3.3) \quad \frac{\partial q}{\partial t} = \nabla^2 q + G.$$

The solution to the hyperbolic subproblem is approximated using the wave propagation algorithms described in [33] and implemented in CLAWPACK [32]. In [12], we

presented the wave propagation algorithm for the discretization of hyperbolic problems on the circle and the sphere grid. A brief description can also be found in section 3.2 of this paper.

3.1. Solving the reaction-diffusion problem. In the following subsection, we describe a finite volume discretization of the Laplace–Beltrami operator that can be used for general smooth quadrilateral surface meshes. Then, in section 3.1.2, we describe the explicit time stepping strategy for evolving the reaction-diffusion problem. In section 3.1.3, we describe how to obtain accurate values at mesh cell nodes, and finally in section 3.1.4, we describe how to impose mixed physical boundary conditions (in the case of the disk or hemisphere), or artificial boundary conditions (in the case of the sphere) on the boundary of the computational domain.

3.1.1. A finite volume approximation of the surface Laplacian. To derive a finite volume approximation to the Laplace–Beltrami operator, we start by assuming the existence of a smooth surface parameterization given by

$$(3.4) \quad T(\xi, \eta) := (X(\xi, \eta), Y(\xi, \eta), Z(\xi, \eta))^T.$$

For now, we will assume that this mapping is differentiable. Using $T(\xi, \eta)$, we can define the vectors tangent to coordinate lines on the surface as

$$(3.5) \quad \mathbf{t}_{(1)} := \left(\frac{\partial X}{\partial \xi}, \frac{\partial Y}{\partial \xi}, \frac{\partial Z}{\partial \xi} \right)^T, \quad \mathbf{t}_{(2)} := \left(\frac{\partial X}{\partial \eta}, \frac{\partial Y}{\partial \eta}, \frac{\partial Z}{\partial \eta} \right)^T.$$

The surface metric tensor $a_{\alpha\beta}$ is defined in terms of the $\mathbf{t}_{(\alpha)}$ as

$$(3.6) \quad a_{\alpha\beta} = \mathbf{t}_{(\alpha)} \cdot \mathbf{t}_{(\beta)}, \quad \alpha, \beta = 1, 2,$$

and the conjugate tensor $a^{\alpha\beta}$ is given as

$$(3.7) \quad a^{11} = \frac{a_{22}}{a}, \quad a^{12} = a^{21} = -\frac{a_{12}}{a}, \quad a^{22} = \frac{a_{11}}{a},$$

where a is the determinant of $a_{\alpha\beta}$ and is given by $a = a_{11}a_{22} - (a_{12})^2$.

Using the conjugate tensor, we can define a set of conjugate vectors $\mathbf{t}^{(\alpha)}$ in terms of $\mathbf{t}_{(\alpha)}$ as

$$(3.8) \quad \begin{aligned} \mathbf{t}^{(1)} &:= a^{11}\mathbf{t}_{(1)} + a^{12}\mathbf{t}_{(2)}, \\ \mathbf{t}^{(2)} &:= a^{21}\mathbf{t}_{(1)} + a^{22}\mathbf{t}_{(2)}. \end{aligned}$$

It is easy to verify that $\mathbf{t}^{(\alpha)}$ are space vectors tangent to the surface. Furthermore, we can show that

$$(3.9) \quad \mathbf{t}^{(\alpha)} \cdot \mathbf{t}_{(\beta)} = \begin{cases} 1 & \text{if } \alpha = \beta, \\ 0 & \text{otherwise.} \end{cases}$$

From this, it follows that $\mathbf{t}^{(\alpha)} \cdot \mathbf{t}^{(\beta)} = a^{\alpha\beta}$. What is important for our purposes is that these conjugate vectors provide us with surface directions normal to the tangents $\mathbf{t}_{(\alpha)}$ and hence normal to surface coordinate lines.

We now apply the divergence theorem for surfaces to construct a finite volume approximation to the integral of the surface Laplacian $\nabla^2 q = \nabla \cdot \nabla q$. The surface divergence theorem is given by

$$(3.10) \quad \int_S \nabla^2 q \, dS = \int_{\partial S} \nabla q \cdot \boldsymbol{\nu} \, ds,$$

where S is a patch on the surface, ∂S is the boundary of the patch, and $\boldsymbol{\nu}$ is a unit vector tangent to the surface and normal to ∂S .

We construct our finite volume approximation to (3.10) by first approximating normal fluxes at the boundary of a small patch S and then approximating the line integral of these fluxes along the boundary. To illustrate this, we let S be an infinitesimal surface control volume defined over the rectangular region $C_0 = [\xi_0, \xi_0 + \Delta\xi] \times [\eta_0, \eta_0 + \Delta\eta]$ in computational space. From here on we restrict the description to quadrilateral grid cells. As an example, we consider the edge of the control volume corresponding to $\xi = \xi_0$. We refer to this as the left edge of the control volume S . A unit vector normal to this edge is given by

$$(3.11) \quad \boldsymbol{\nu}^{(1)} := \frac{\mathbf{t}^{(1)}}{\|\mathbf{t}^{(1)}\|} = \frac{\mathbf{t}^{(1)}}{\sqrt{a^{11}}},$$

where $\|\cdot\|$ is the usual Cartesian product. The normal derivative of q at this edge can be computed by taking the dot product of $\boldsymbol{\nu}^{(1)}$ with the gradient of q . If we assume that the gradient of q has no components in the direction normal to the surface, we can express it in Cartesian coordinates as

$$(3.12) \quad \left(\frac{\partial q}{\partial X}, \frac{\partial q}{\partial Y}, \frac{\partial q}{\partial Z} \right)^T = \frac{\partial q}{\partial \xi} \mathbf{t}^{(1)} + \frac{\partial q}{\partial \eta} \mathbf{t}^{(2)}.$$

The normal derivative of q at the left edge can then be computed as

$$(3.13) \quad \begin{aligned} \frac{dq}{dn} \Bigg|_{\xi=\xi_0} &= \left(\frac{\partial q}{\partial \xi} \mathbf{t}^{(1)} + \frac{\partial q}{\partial \eta} \mathbf{t}^{(2)} \right) \cdot \boldsymbol{\nu}^{(1)} \\ &= \frac{1}{\sqrt{a^{11}}} \left(a^{11} \frac{\partial q}{\partial \xi} + a^{21} \frac{\partial q}{\partial \eta} \right). \end{aligned}$$

For a finite volume approximation, we are interested in an integral of the normal flux (3.13) over the left edge of S . To first order, this integral can be approximated by multiplying (3.13) by the physical length of the edge. In general, infinitesimal lengths ds on the surface can be computed from the surface metric $a_{\alpha\beta}$ using

$$(3.14) \quad ds^2 = a_{11} d\xi^2 + 2a_{12} d\xi d\eta + a_{22} d\eta^2.$$

Along the edge $\xi = \xi_0$, we have $d\xi = 0$, so that the finite edge length can be approximated by $ds \approx \sqrt{a_{22}} \Delta\eta$. The integral of the flux in (3.13) over the left edge of the control volume dS can then be approximated as

$$(3.15) \quad ds \frac{dq}{dn} \Bigg|_{\xi=\xi_0} \approx \sqrt{a} \left(a^{11} \frac{\partial q}{\partial \xi} + a^{21} \frac{\partial q}{\partial \eta} \right) \Delta\eta,$$

where we have used (3.7) to write $(\sqrt{a_{22}} \Delta\eta) / \sqrt{a^{11}} = \sqrt{a} \Delta\eta$.

To derive a semidiscrete approximation to (3.10), we approximate the integrals of the normal fluxes across the remaining three edges of S , orient all normals so that they are directed out of the control volume, and express the finite volume approximation

to the integral of the surface Laplacian as

$$(3.16) \quad \begin{aligned} \int_S \nabla^2 q \, dS &= \int_{\partial S} \nabla q \cdot \boldsymbol{\nu} \, ds \\ &\approx \left\{ \sqrt{a} \left(a^{11} \frac{\partial q}{\partial \xi} + a^{21} \frac{\partial q}{\partial \eta} \right) \Big|_{\xi=\xi_0+\Delta\xi} - \sqrt{a} \left(a^{11} \frac{\partial q}{\partial \xi} + a^{21} \frac{\partial q}{\partial \eta} \right) \Big|_{\xi=\xi_0} \right\} \Delta\eta \\ &\quad + \left\{ \sqrt{a} \left(a^{12} \frac{\partial q}{\partial \xi} + a^{22} \frac{\partial q}{\partial \eta} \right) \Big|_{\eta=\eta_0+\Delta\eta} - \sqrt{a} \left(a^{12} \frac{\partial q}{\partial \xi} + a^{22} \frac{\partial q}{\partial \eta} \right) \Big|_{\eta=\eta_0} \right\} \Delta\xi. \end{aligned}$$

Using the approximation $dS \approx \sqrt{a} \Delta\xi\Delta\eta$, one can also view the above as an approximation to the integral of the Laplace–Beltrami operator over the computational cell C_0 . In differential form, the Laplace–Beltrami operator is given by

$$(3.17) \quad \nabla^2 q = \frac{1}{\sqrt{a}} \frac{\partial}{\partial \xi} \left\{ \sqrt{a} \left(a^{11} \frac{\partial q}{\partial \xi} + a^{12} \frac{\partial q}{\partial \eta} \right) \right\} + \frac{\partial}{\partial \eta} \left\{ \sqrt{a} \left(a^{21} \frac{\partial q}{\partial \xi} + a^{22} \frac{\partial q}{\partial \eta} \right) \right\}.$$

Obtaining discrete metric terms. Because in general, our mesh is nonorthogonal, our discretization will involve function values at both centers and nodes of an underlying Cartesian mesh. We discretize the computational domain $[a_x, b_x] \times [a_y, b_y]$ using a uniform Cartesian mesh with mesh spacing $\Delta\xi$ and $\Delta\eta$. Mesh cell corners or *nodes* are given by $(\hat{\xi}_i, \hat{\eta}_j)$, defined by $\hat{\xi}_i = a_x + (i-1)\Delta\xi$ and $\hat{\eta}_j = a_y + (j-1)\Delta\eta$ for (i, j) in the index space $[1, \dots, M_x+1] \times [1, \dots, M_y+1]$. Cell centers (ξ_i, η_j) are defined over the index space $[0, \dots, M_x+1] \times [0, 1, \dots, M_y+1]$. For problems with periodic boundary conditions, or for our sphere example, cell centers are defined over the entire index space as $\xi_i = a_x + (i-1/2)\Delta\xi$ and $\eta_j = a_y + (j-1/2)\Delta\eta$. For problems in which we impose Dirichlet or flux conditions at physical boundaries, we set $\xi_0 = a_x$, $\xi_{M_x+1} = b_x$, $\eta_0 = a_y$, and $\eta_{M_y+1} = b_y$.

Associated with each node and cell center in the computational space are nodes and centers of mesh cells in the physical domain. Using the notation suggested by Hermeline and others who have developed methods based on diamond-cell approximations or discrete duality principles [16, 27, 19], we refer to the physical points associated with (ξ_i, η_j) as the *primal points*, and physical points associated with $(\hat{\xi}_i, \hat{\eta}_j)$ as the *dual points*. The primal mesh cell P_{ij} is the cell whose center is \mathbf{x}_{ij} . Analogously, the dual mesh cell D_{ij} is the cell with center $\hat{\mathbf{x}}_{ij}$. Note that \mathbf{x}_{ij} and $\hat{\mathbf{x}}_{ij}$ are generally not the centers of mass of the primal and dual grid cell, respectively. Figure 3.1 shows the locations of the primal and dual nodes.

At primal and dual points, we define the primal value of function q as $q_{ij} = q(\xi_i, \eta_j)$. Dual values are defined analogously as $\hat{q}_{ij} = q(\hat{\xi}_i, \hat{\eta}_j)$. For a finite volume scheme, these values should approximate the average value of the function q over the primal or dual mesh cells.

We let $S_{i-1/2,j}$ denote the primary cell edge defined as the straight line segment (not necessarily on the surface) connecting primal nodes $\hat{\mathbf{x}}_{i,j+1}$ and $\hat{\mathbf{x}}_{ij}$. Similarly, we let $\hat{S}_{i-1/2,j}$ denote the edge of the dual cell connecting dual nodes $\mathbf{x}_{i-1,j}$ and \mathbf{x}_{ij} . Associated with these edges are positively oriented vectors (with respect to our Cartesian mapping)

$$(3.18) \quad \mathbf{t}_{i-1/2,j} := \hat{\mathbf{x}}_{i,j+1} - \hat{\mathbf{x}}_{ij}, \quad \hat{\mathbf{t}}_{i-1/2,j} := \mathbf{x}_{i-1,j} - \mathbf{x}_{i-1,j}.$$

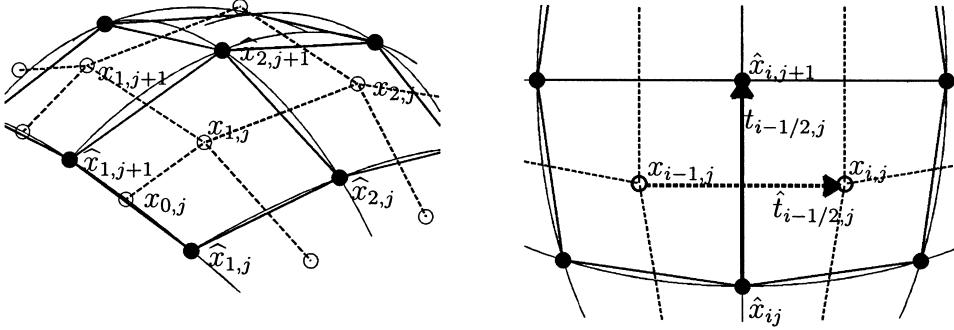


FIG. 3.1. Diagrams showing the location of primal and dual nodes on a quadrilateral surface mesh. The open circles are the primal points \mathbf{x}_{ij} (i.e., dual nodes), and the closed circles are the dual points $\hat{\mathbf{x}}_{ij}$ (i.e., primal nodes). Dual nodes are the corners of primal mesh cells, and primal nodes are the corners of dual mesh cells. If physical boundary conditions are to be imposed at the computational boundary, primal mesh points are located directly on the physical boundary, as shown in the left figure. The right figure shows a general region on the mesh and the vectors $\mathbf{t}_{i-1/2,j}$ (solid arrow at primal mesh cell edge $S_{i-1/2,j}$) and $\hat{\mathbf{t}}_{i-1/2,j}$ (dashed arrow at dual cell edge $\hat{S}_{i-1/2,j}$).

The lengths of these vectors (and hence of associated edges) are given by

$$(3.19) \quad |S_{i-1/2,j}| := \| \mathbf{t}_{i-1/2,j} \|, \quad |\hat{S}_{i-1/2,j}| := \| \hat{\mathbf{t}}_{i-1/2,j} \|,$$

where $\| \cdot \|$ is the Euclidean norm in \mathbb{R}^3 . Similarly, we define the primary mesh cell edge $S_{i,j-1/2}$ as the edge between primal nodes $\hat{\mathbf{x}}_{i+1,j}$ and $\hat{\mathbf{x}}_{ij}$. The associated dual mesh cell edge $\hat{S}_{i,j-1/2}$ is the line segment between dual nodes \mathbf{x}_{ij} and $\mathbf{x}_{i,j-1}$. Vectors associated with these edges are given by

$$(3.20) \quad \mathbf{t}_{i,j-1/2} := \hat{\mathbf{x}}_{i+1,j} - \hat{\mathbf{x}}_{ij}, \quad \hat{\mathbf{t}}_{i,j-1/2} := \mathbf{x}_{ij} - \mathbf{x}_{i,j-1},$$

and their associated lengths $|S_{i,j-1/2}|$ and $|\hat{S}_{i,j-1/2}|$ are defined as above.

Using these definitions, we approximate the metric tensor $a_{\alpha\beta}$ and its conjugate. For example, at edge $S_{i-1/2,j}$, we have

$$(3.21) \quad \begin{aligned} a_{11}^{(i-1/2,j)} &= (\hat{\mathbf{t}}_{i-1/2,j}) \cdot (\hat{\mathbf{t}}_{i-1/2,j}), \\ a_{12}^{(i-1/2,j)} &= a_{21}^{(i-1/2,j)} = (\hat{\mathbf{t}}_{i-1/2,j}) \cdot (\mathbf{t}_{i-1/2,j}), \\ a_{22}^{(i-1/2,j)} &= (\mathbf{t}_{i-1/2,j}) \cdot (\mathbf{t}_{i-1/2,j}). \end{aligned}$$

The determinant of $a_{\alpha\beta}^{(i-1/2,j)}$ is given by

$$(3.22) \quad a^{(i-1/2,j)} = \det(a_{\alpha\beta}^{(i-1/2,j)}) = \| (\hat{\mathbf{t}}_{i-1/2,j}) \times (\mathbf{t}_{i-1/2,j}) \|^2,$$

and its conjugate $a_{(i-1/2,j)}^{\alpha\beta}$ is defined as in (3.7). Metric terms at the remaining three edges of the control volume S are defined analogously.

To define the partial derivatives of q in (3.16), we may, without loss of generality, assume that for our underlying mapping we have $\Delta\xi = \Delta\eta = 1$. Then derivatives of q at edge $S_{i-1/2,j}$ can be approximated by

$$(3.23) \quad \left. \frac{\partial q}{\partial \xi} \right|_{S_{i-1/2,j}} \approx q_{i,j} - q_{i-1,j} := \Delta q_{i-1/2,j}, \quad \left. \frac{\partial q}{\partial \eta} \right|_{S_{i-1/2,j}} \approx \hat{q}_{i,j+1} - \hat{q}_{i,j} := \Delta \hat{q}_{i-1/2,j}.$$

At the edge $S_{i,j-1/2}$ we have analogous definitions

$$(3.24) \quad \frac{\partial q}{\partial \xi} \Big|_{S_{i,j-1/2}} \approx \widehat{q}_{i+1,j} - \widehat{q}_{ij} := \Delta \widehat{q}_{i,j-1/2}, \quad \frac{\partial q}{\partial \eta} \Big|_{S_{i,j-1/2}} \approx q_{ij} - q_{i,j-1} := \Delta q_{i,j-1/2}.$$

Finally, we will make use of the vector identities

$$(3.25) \quad \begin{aligned} (\widehat{\mathbf{t}}_{i-1/2,j}) \cdot (\widehat{\mathbf{t}}_{i-1/2,j}) &= |\widehat{S}_{i-1/2,j}|^2, \\ (\mathbf{t}_{i-1/2,j}) \cdot (\mathbf{t}_{i-1/2,j}) &= |S_{i-1/2,j}|^2, \\ (\widehat{\mathbf{t}}_{i-1/2,j}) \cdot (\mathbf{t}_{i-1/2,j}) &= |\widehat{S}_{i-1/2,j}| |S_{i-1/2,j}| \cos(\theta_{i-1/2,j}), \\ \|(\widehat{\mathbf{t}}_{i-1/2,j}) \times (\mathbf{t}_{i-1/2,j})\| &= |\widehat{S}_{i-1/2,j}| |S_{i-1/2,j}| \sin(\theta_{i-1/2,j}), \end{aligned}$$

where $\theta_{i-1/2,j}$ is the angle between $\widehat{\mathbf{t}}_{i-1/2,j}$ and $\mathbf{t}_{i-1/2,j}$.

Using the above definitions and the vector identities, the flux in (3.15) is given by

$$(3.26) \quad \int_{S_{i-1/2,j}} \frac{\partial q}{\partial n} dL \approx p_{i+1/2,j} \Delta q_{i+1/2,j} + \widehat{u}_{i+1/2,j} \Delta \widehat{q}_{i+1/2,j},$$

and the integral of the Laplacian given in (3.16) is approximated as

$$(3.27) \quad \begin{aligned} \int_{P_{ij}} \nabla^2 q dS &\approx I_{ij}^{lb}(q) := (p_{i+1/2,j} \Delta q_{i+1/2,j} + \widehat{u}_{i+1/2,j} \Delta \widehat{q}_{i+1/2,j}) \\ &\quad - (p_{i-1/2,j} \Delta q_{i-1/2,j} + \widehat{u}_{i-1/2,j} \Delta \widehat{q}_{i-1/2,j}) \\ &\quad + (p_{i,j+1/2} \Delta q_{i,j+1/2} + \widehat{u}_{i,j+1/2} \Delta \widehat{q}_{i,j+1/2}) \\ &\quad - (p_{i,j-1/2} \Delta q_{i,j-1/2} + \widehat{u}_{i,j-1/2} \Delta \widehat{q}_{i,j-1/2}), \end{aligned}$$

where

$$(3.28) \quad \begin{aligned} p_{i-1/2,j} &:= \left(\frac{|S_{i-1/2,j}|}{|\widehat{S}_{i-1/2,j}|} \right) \csc(\theta_{i-1/2,j}), \\ \widehat{u}_{i-1/2,j} &:= -\cot(\theta_{i-1/2,j}). \end{aligned}$$

This formula is closely related to that used by Hermeline [27] for approximating the Laplacian on flat polygonal meshes, and the well-known ‘‘cotan’’ formula used by Pinkall, Desbrun, and their coworkers [43, 18, 39] for approximating the surface Laplacian. In the appendices, we show the connection between (3.28) and these other formulas.

To obtain the pointwise Laplace–Beltrami operator, we divide the above approximation by the area of the primal cell to get

$$(3.29) \quad \nabla^2 q \approx L_{ij}(q) := \frac{1}{\text{Area}(P_{ij})} I_{ij}^{lb}(q).$$

3.1.2. Explicit time stepping scheme for the diffusion problem. The time stepping strategy we describe here is well suited for our disk and sphere grids. In particular, we take advantage of the fact that our mappings for these surfaces generate

grids with nearly uniform cell sizes, making the use of explicit time discretizations practical.

We solve the parabolic equation given by

$$(3.30) \quad q_t = \nabla^2 q + G(q, \dots),$$

where $\nabla^2 q$ is approximated using (3.29) and G may depend on the solution, the spatial location on the surface, or time.

We define time levels as $t_n = n\Delta t$, for $n = 0, 1, \dots$. The computed solution at time level n is then denoted q^n . If we were to use a forward Euler scheme, our time marching would take the form

$$(3.31) \quad q_{ij}^{n+1} = q_{ij}^n + \Delta t (L_{ij}(q^n) + G_{ij}^n),$$

where q_{ij}^n is the approximation

$$(3.32) \quad q_{ij}^n \approx \frac{1}{\text{Area}(P_{ij})} \int_{P_{ij}} q(\mathbf{x}, t) dS.$$

The source term is then typically evaluated as $G_{ij}^n \equiv G(q_{ij}^n, \dots)$. For our disk and sphere grids, pointwise values at primal points are good approximations to the cell average, except in the nearly triangular cells along the diagonal. In those cells, the location of the primal node approximates the location of the centroid only to first order.

Because of the severe time constraints imposed by the numerical stability requirements of the forward Euler scheme, we in fact use a more sophisticated explicit time marching scheme designed specifically for reaction-diffusion equations. This second order accurate scheme is part of a class of Runge–Kutta–Chebyshev (RKC) methods proposed by Sommeijer, Shampine, and Verwer [51] and allows for a time step $\Delta t \approx \Delta\xi$. The RKC solver, available on the web, also has low storage requirements, making it an attractive alternative to implicit schemes, even for very large systems of equations. For more details on this scheme, see [51, 34]. In section 4.2, we show that the RKC solver is competitive with a standard implicit solver for parabolic equations.

3.1.3. Obtaining dual cell values. The accuracy of our overall scheme relies crucially on the fact that we can accurately obtain the dual values needed by (3.27). If these values are only first order accurate, even along lines, then the overall scheme will be only first order accurate. In what follows, we assume that for our disk and hemisphere grids we have $M_x = M_y$. For the sphere grid, we assume that $M_x = 2M_y$. These assumptions allow us to easily handle the metric discontinuities that appear along diagonal lines in the computational domains for these grids.

For regions of our disk and sphere mappings away from discontinuities in the metric, we can obtain the dual cell values required by (3.27) by simply averaging primal cell values to primal nodes using

$$(3.33) \quad \hat{q}_{ij} = \frac{1}{4} (q_{i-1,j} + q_{i-1,j-1} + q_{i,j-1} + q_{i,j}) \approx \frac{1}{\text{Area}(D_{ij})} \int_{D_{ij}} q(\mathbf{x}, t) dS.$$

This is a second order approximation to the cell average over the dual cell D_{ij} as long as the function q is a smooth function of the computational variables (ξ, η) and the computational grid is uniformly discretized.

Our disk and sphere mappings are discontinuous at diagonals of the computational grid, and so $q(\xi, \eta)$ exhibits a discontinuity in the derivative along the diagonals, even if the function in physical space is smooth across diagonals. Using (3.33) in these cells reduces the overall accuracy of the solution to first order. To restore second order accuracy for these meshes, we can take advantage of our uniform discretization on the computational grid and the fact that the discontinuity lies exactly on the diagonals of the computational grid. Along the diagonal, we expect the solution to be smooth, and since computational variables corresponding to dual and primal nodes are equidistributed along the diagonals, we can compute accurate dual cell values using only the two neighboring primal values along the diagonal. This average is given by

$$(3.34) \quad \begin{aligned} \hat{q}_{ii} &= \frac{1}{2} (q_{i,i} + q_{i-1,i-1}), \\ \hat{q}_{i,M_x-i+2} &= \frac{1}{2} (q_{i-1,M_x-i+2} + q_{i,M_x-i+1}), \end{aligned}$$

where we have assumed that $M_x = M_y$. For the sphere, diagonal values are defined analogously, where the same idea applies on each hemisphere of the sphere grid. This simple modification restores overall accuracy of the scheme to second order; see section 4.

For problems in which we have physical boundary conditions to impose, the primal values are assumed to lie exactly on the boundary, so it is natural to obtain dual boundary values using an average based only on primal boundary values. For the disk and hemisphere grids, we define

$$(3.35) \quad \begin{aligned} \hat{q}_{1,j} &= \frac{1}{2} (q_{0,j-1} + q_{0,j}), \\ \hat{q}_{M_x+1,j} &= \frac{1}{2} (q_{M_x+1,j-1} + q_{M_x+1,j}), \\ \hat{q}_{i,1} &= \frac{1}{2} (q_{i-1,0} + q_{i,0}), \\ \hat{q}_{i,M_y+1} &= \frac{1}{2} (q_{i-1,M_y+1} + q_{i,M_y+1}). \end{aligned}$$

This is an accurate approximation as long as the domain boundary is smooth, as is the case with our disk or hemisphere grid.

For the sphere grid, we use the same idea to compute dual cell values that lie along the equator, since our sphere mapping is not smooth across the equator. In section 3.1.4, we explain how we compute values for a set of fictitious primal cells, that we place directly on the equator for the purpose of using (3.35) to obtain dual cell values.

In the next section, we show how to impose physical boundary conditions to obtain primal boundary values for general boundary conditions.

3.1.4. Boundary conditions for the diffusion problem. Because of the way we position primal and dual points exactly on physical boundaries, Dirichlet conditions can be applied by simply evaluating boundary conditions at these points. In this case, it is not in fact necessary to use (3.35). Below, we describe how we handle more general physical boundary conditions for the disk and hemisphere grids. Also in this section, we describe how we treat metric discontinuities at the equator of the sphere grid.

Physical boundary conditions. For problems with physical boundaries, we can impose general boundary conditions of the form

$$(3.36) \quad aq + b \frac{\partial q}{\partial n} = c,$$

where a , b , or c can vary along the boundary. To discretize this equation, we first define vectors containing boundary values for the primal and dual cells. Starting with the values $q_{1,0}$ and $q_{1,1}$ and working around the Cartesian boundary counterclockwise, we construct vectors q^p of primal boundary values and q^b , the closest interior values to these boundary values, as

$$(3.37) \quad q^b = [q_{1,1}, q_{2,1}, \dots, q_{M_x,1}, q_{M_x,1}, q_{M_x,2}, \dots, q_{M_x,M_y}, q_{M_x,M_y}, \dots, q_{1,1}]^T,$$

$$(3.38) \quad q^p = [q_{1,0}, q_{2,0}, \dots, q_{M_x,0}, q_{M_x+1,1}, q_{M_x+1,2}, \dots, q_{M_x+1,M_y}, \dots, q_{0,1}]^T.$$

The entries of each vector are given by q_k^p and q_k^b , for $k = 1, 2, \dots, N_p$, $N_p = 2(M_x + M_y)$. Note that in the definition of q^b we have intentionally included corner entries twice.

Similarly, we construct vectors p and \hat{u} of the same length using the coefficients defined in (3.28). These vectors are given by

$$(3.39) \quad \begin{aligned} p &= \left[\frac{p_{1,1/2}}{|S_{1,1/2}|}, \frac{p_{2,1/2}}{|S_{2,1/2}|}, \dots, \frac{p_{M_x,1/2}}{|S_{M_x,1/2}|}, \frac{p_{M_x+1/2,1}}{|S_{M_x+1,1}|}, \dots, \frac{p_{1/2,1}}{|S_{1/2,1}|} \right]^T, \\ \hat{u} &= \left[\frac{\hat{u}_{1,1/2}}{|S_{1,1/2}|}, \frac{\hat{u}_{2,1/2}}{|S_{2,1/2}|}, \dots, \frac{\hat{u}_{M_x,1/2}}{|S_{M_x,1/2}|}, \frac{\hat{u}_{M_x+1/2,1}}{|S_{M_x+1,1/2}|}, \dots, \frac{\hat{u}_{1/2,1}}{|S_{1/2,1}|} \right]^T. \end{aligned}$$

Their entries are denoted by p_k and \hat{u}_k for $k = 1, 2, \dots, N_p$. To ensure that our resulting expression for the normal derivative at the boundary has the correct sign, we define

$$(3.40) \quad s_k^d = \begin{cases} -1, & 1 \leq k \leq M_x \text{ or } M_x + M_y + 1 \leq k \leq 2M_x + M_y, \\ 1, & \text{otherwise,} \end{cases}$$

and simplify the notation that follows by redefining the entries of \hat{u} from (3.39) as

$$(3.41) \quad \hat{u}_k \leftarrow s_k^d \hat{u}_k.$$

Finally, we construct a vector \hat{q}^d containing dual boundary values. Starting with $\hat{q}_{1,1}$, we again work counterclockwise around the computational grid to get

$$(3.42) \quad \hat{q}^d = [\hat{q}_{1,1}, \hat{q}_{2,1}, \dots, \hat{q}_{M_x+1,1}, \hat{q}_{M_x+1,2}, \dots, \hat{q}_{M_x+1,M_y+1}, \hat{q}_{M_x,M_y+1}, \dots, \hat{q}_{1,3}]^T.$$

We have intentionally omitted the expected final entry $\hat{q}_{1,2}$. Below, we treat this value as an unknown which we solve for in a separate step. The vector in (3.42) has entries \hat{q}_k^d for $k = 1, 2, \dots, N_p - 1$.

Using q^p , \hat{q}^d , q^b and coefficient vectors p and \hat{u} , we discretize the boundary conditions in (3.36) as

$$(3.43) \quad \begin{aligned} aq_k^p + b(p_k(q_k^p - q_k^b) + \hat{u}_k(\hat{q}_{k+1}^d - \hat{q}_k^d)) &= c, \quad k = 1, 2, \dots, N_p - 2, \\ aq_{N_p-1}^p + b(p_{N_p-1}(q_{N_p-1}^p - q_{N_p-1}^b) + \hat{u}_{N_p-1}(\alpha - \hat{q}_{N_p-1}^d)) &= c, \\ aq_{N_p}^p + b(p_{N_p}(q_{N_p}^p - q_{N_p}^b) + \hat{u}_{N_p}(\hat{q}_1^d - \alpha)) &= c. \end{aligned}$$

We can also rewrite condition (3.35) using these vectors to get

$$(3.44) \quad \begin{aligned} q_1^p + q_{N_p}^p - 2\hat{q}_1^d &= 0, \\ q_k^p + q_{k-1}^p - 2\hat{q}_k^d &= 0, \quad k = 2, \dots, N_p - 1, \\ q_{N_p}^p + q_{N_p-1}^p - 2\alpha &= 0. \end{aligned}$$

Solving for q_k^p in (3.43) and substituting the resulting expressions into (3.44), we obtain a linear system for unknowns \hat{q}_k^d and an unknown α given by

$$(3.45) \quad \begin{pmatrix} \mathbf{T} & \mathbf{w} \\ \mathbf{v}^T & u \end{pmatrix} \begin{pmatrix} \hat{q}^d \\ \alpha \end{pmatrix} = \begin{pmatrix} \mathbf{f}_T \\ f_\alpha \end{pmatrix}.$$

The matrix $\mathbf{T} \in \mathbf{R}^{(N_p-1) \times (N_p-1)}$ is a tridiagonal matrix which we can invert using a fast solver. The entries \mathbf{w} and \mathbf{f}_T are column vectors of length $N_p - 1$, and \mathbf{v}^T is a row vector, also of length $N_p - 1$. Entries u and f_α are scalars.

Using (3.45), it is straightforward to show that for arbitrary α the solution \hat{q}^d can be written as a linear combination of a time dependent solution ϕ_{td}^d and a time independent solution ϕ_{tid}^d as

$$(3.46) \quad \hat{q}^d = \phi_{td}^d + \alpha \phi_{tid}^d,$$

where

$$(3.47) \quad \mathbf{T} \phi_{td}^d = \mathbf{f}_T \quad \text{and} \quad \mathbf{T} \phi_{tid}^d = -\mathbf{w}.$$

Using this form of the solution, we can then solve for α to get

$$(3.48) \quad \alpha = \frac{f_\alpha - \mathbf{v}^T \phi_{td}^d}{u + \mathbf{v}^T \phi_{tid}^d}.$$

The solution ϕ_{td}^d depends on the time dependent solution q^p through the right-hand-side vector \mathbf{f}_T , and so must be solved for at each step. However, the solution ϕ_{tid}^d depends only on the mesh geometry, and so can be solved for in a preprocessing step.

To describe the entries of the tridiagonal matrix T , the vectors \mathbf{v} , \mathbf{w} , and \mathbf{f}_T , and scalars u and f_α , we first define two sets of constants. These are given by

$$(3.49) \quad \left. \begin{aligned} t_k &:= \frac{b\hat{u}_k}{a + bp_k}, & t_0 &:= t_{N_p} \\ r_k &:= \frac{c + bp_k q_k^b}{a + bp_k}, & r_0 &:= r_{N_p} \end{aligned} \right\} \quad k = 1, \dots, N_p.$$

Using the constants t_k , the entries of the tridiagonal matrix \mathbf{T} are given by

$$(3.50) \quad \begin{aligned} T_{k,k-1} &= t_k, & k &= 2, \dots, N_p - 1, \\ T_{k,k} &= t_k - t_{k-1} - 2, & k &= 1, \dots, N_p - 1, \\ T_{k,k+1} &= -t_k, & k &= 1, \dots, N_p - 2; \end{aligned}$$

the vectors \mathbf{v} and \mathbf{w} are given by

$$(3.51) \quad \begin{aligned} w_1 &= t_{N_p}, & w_{N_p-1} &= -t_{N_p-1}, & v_k &= w_k = 0, & k &= 2, \dots, N_p - 2; \\ v_1 &= -t_{N_p}, & v_{N_p-1} &= t_{N_p-1}, \end{aligned}$$

and the scalar u is given by

$$(3.52) \quad u = t_{N_p} - t_{N_p-1} - 2.$$

The right-hand-side vector \mathbf{f}_T and scalar f_α are given in terms of constants r_k as

$$(3.53) \quad \begin{aligned} (f_T)_k &= -(r_{k-1} + r_k), \quad k = 1, \dots, N_p - 1, \\ f_\alpha &= -(r_{N_p-1} + r_{N_p}). \end{aligned}$$

RIGHT-HAND-SIDE FUNCTION FOR THE RKC SOLVER.

- In a preprocessing step, construct the matrix \mathbf{T} and solve the tridiagonal system $\mathbf{T}\phi_{tid}^d = -\mathbf{w}$.
- Call the RKC solver with a right-hand-side function $F_{rkc}(t, q)$, which takes as input the current time and state variable q and returns the right-hand side $L_{ij}(q) + G(q, \dots)$.
- The steps performed in this right-hand-side function are as follows:
 1. Solve $\mathbf{T}\phi_{tid}^d = f^T$, where f^T is constructed using q^p extracted from grid data q passed from the RKC solver.
 2. Compute α using (3.48).
 3. Compute $\hat{q}^d = \phi_{tid}^d + \alpha\phi_{tid}^d$.
 4. Use (3.43) and \hat{q}^d to evaluate primal boundary values q^p .
 5. On the grid interior, average primal values q_{ij} to dual values \hat{q}_{ij} using (3.33) and (3.34). For dual boundary values, use (3.35) or the solution \hat{q}^d computed in step 3 directly.
 6. Compute $L_{ij}(q)$ at each nonboundary primal value over the entire grid.
 7. Compute any source term values $G(q, \dots)$ and add them to $L_{ij}(q)$.

- Return to the RKC solver a vector containing the Laplacian plus source terms for each interior primal mesh cell. The same algorithm can be used for both problems with physical boundaries (i.e., the disk or hemisphere) and for the sphere grid.

Ensuring flux continuity across the equator for sphere grids. A sphere does not have physical boundaries, but our sphere mapping is not smooth across the equator, and so to preserve the accuracy of our proposed scheme we explicitly ensure the continuity of fluxes at the equator using an approach analogous to that used for physical boundaries.

On the computational domain, we use the layer of ghost cell values bordering the boundary of the computational grid to supply primal values needed to evaluate the Laplacian. These primal values can be easily obtained by

$$(3.54) \quad \left. \begin{aligned} q_{0,j} &= q_{M_x,j} \\ q_{M_x+1,j} &= q_{1,j} \end{aligned} \right\}, \quad j = 1, \dots, M_y,$$

$$\left. \begin{aligned} q_{i,0} &= q_{M_x-i+1,0} \\ q_{i,M_y+1} &= q_{M_x-i+1,M_y+1} \end{aligned} \right\}, \quad i = 1, \dots, M_x.$$

To formulate the linear system needed to solve for dual values at the equator, we first introduce fictitious primal boundary points directly on the equator, between these dual points. On the computational grid, we place these points along the bottom and top of the right half of the grid, the left edge, and along the interior center line. We can think of these primal boundary values as bordering the upper-hemisphere of

the sphere grid. Furthermore, we allow the ghost cell values (excluding corner ghost cells) and interior values just left of the center-line to form a continuous array of ghost cell values around this hemisphere grid.

In direct analogy with how we imposed the physical boundary conditions on the hemisphere, we construct 1-d arrays for the sphere grid that start at the lower left corner of the hemisphere grid (the point $(-1, -1)$, in computational space), and traverse the boundary of the right half of the computational grid counterclockwise, returning to $(-1, -1)$ via the center-line $\xi = -1$. These data arrays have $N_p = 2(M_x/2 + M_y)$ points, where we are assuming that for the sphere mapping, $M_x = 2M_y$. For the upper hemisphere, we construct the vector q^u , containing primal values, in exactly the same manner as we did in (3.38). The primal ghost cell values (i.e., values in cells in the lower hemisphere that share an edge with the upper hemisphere) are stored in the same manner in a vector q^ℓ . We also define the single vector q^p of the fictitious primal boundary values we introduced above. Dual cell values along the equator are stored in \hat{q}^d , and as before, we use α to represent the last value in this array and let $\alpha = \hat{q}_{M_x/2+1,2}$.

To get coefficient vectors analogous to those defined in (3.39), we will need to recompute the coefficients used in (3.27) for primal boundary points in our fictitious array of boundary points, not from ghost cells. We create a set of coefficient vectors p^u and p^ℓ for the cells bordering the equator in the upper and lower hemisphere. We do not need to apply the sign change in (3.41), since we match only fluxes, not normal derivatives.

Using these definitions, we define fluxes at primal cell edges that lie on the equator. For $k = 1, 2, \dots, N_p - 2$, these fluxes are given by

$$(3.55) \quad \begin{aligned} F_k^u &:= p_k^u(q_k^p - q_k^u) + \hat{u}_k^u(\hat{q}_{k+1}^d - \hat{q}_k^d), & k = 1, 2, \dots, N_p - 2. \\ F_k^\ell &:= p_k^\ell(q_k^p - q_k^\ell) + \hat{u}_k^\ell(\hat{q}_{k+1}^d - \hat{q}_k^d), \end{aligned}$$

Fluxes for $k = N_p - 1$ and $k = N_p$ are given in terms of α , just as in (3.43).

These fluxes are directed out of their respective cells, and so, to ensure continuity of fluxes across the boundary, we impose

$$(3.56) \quad F_k^u + F_k^\ell = 0, \quad k = 1, 2, \dots, N_p.$$

Solving for q_k^p and substituting this value into (3.44), we obtain a tridiagonal system for the dual values along the equator, where now the coefficients t_k and r_k appearing in (3.50)–(3.53) are given by

$$(3.57) \quad \left. \begin{aligned} t_k &:= \frac{\hat{u}_k^u + \hat{u}_k^\ell}{p_k^u + p_k^\ell}, & t_0 &:= t_{N_p} \\ r_k &:= \frac{p_k^u q_k^u + p_k^\ell q_k^\ell}{p_k^u + p_k^\ell}, & r_0 &:= r_{N_p} \end{aligned} \right\} \quad k = 1, \dots, N_p.$$

The dual values are obtained in exactly the same fashion as they were in section 3.1.4.

3.2. Solving the advection problem. Many PDEs of importance in scientific fields include advection terms as well as diffusion terms. In section 5.1, we present a numerical approach to solving a chemotaxis model in a petri dish. This model involves both advection and diffusion of transported chemo-attractants.

Here we briefly describe how we solve the hyperbolic subproblem

$$\partial_t q + \nabla \cdot f(q) = 0$$

for the scalar quantity q on our disk and sphere grids. The details of this approach were described in [12] and are based on an approach for curvilinear grids found in [33]. We perform this update on the primal grid cells.

A finite volume method can be applied on a grid cell P_{ij} using the integral form of the conservation law

$$\frac{d}{dt} \int_{P_{ij}} q(\mathbf{x}, t) dS = - \int_{\partial P_{ij}} f(q) \cdot \boldsymbol{\nu} ds,$$

where $\boldsymbol{\nu}$ is again the outward pointing unit normal vector at the boundary ∂P_{ij} of the grid cell P_{ij} and $f \cdot \boldsymbol{\nu}$ is the flux normal to the boundary. This leads to a finite volume method of the form

$$Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{\text{Area}(P_{ij})} \sum_{k=1}^N |S_k| \check{F}_k^n,$$

where Q_{ij}^n is the cell average of the conserved quantity in grid cell P_{ij} at time t_n , \check{F}_k^n represents a numerical approximation to the average normal flux across the k th side of the grid cell, and $|S_k|$ is the length of the k th side.

In our applications, a hyperbolic problem arises as a subproblem in the modeling of chemotaxis. The equation has the form

$$(3.58) \quad \frac{\partial q}{\partial t} + \alpha \nabla \cdot \left(\frac{\nabla v}{(1+v)^2} q \right) = 0,$$

where $v = v(x, y, t)$ is assumed to be a given function in this fractional step and α is a given constant parameter.

Following [55], we rewrite (3.58) as a transport equation of the form

$$(3.59) \quad \frac{\partial q}{\partial t} + \frac{\partial}{\partial x} (U(x, y, t) q) + \frac{\partial}{\partial y} (V(x, y, t) q) = 0,$$

where the velocities U and V are given by

$$\begin{aligned} U &= \frac{\alpha}{(1+v)^2} \frac{\partial v}{\partial x}, \\ V &= \frac{\alpha}{(1+v)^2} \frac{\partial v}{\partial y}. \end{aligned}$$

A finite volume advection scheme requires the velocity normal to cell interfaces. Using the flux expression (3.15), we can write this normal derivative as

$$(3.60) \quad ds \frac{\partial v}{\partial n} \approx \sqrt{a} \left(a^{11} \frac{\partial v}{\partial \xi} + a^{12} \frac{\partial v}{\partial \eta} \right) \Delta \eta.$$

Just as with the diffusion fluxes, we can substitute discrete geometric terms here and obtain the flux formula in (3.26).

In this paper, the conserved quantity arising in the approximation of the hyperbolic subproblem is a scalar variable. The ideas presented here extend to systems of hyperbolic equations, in which q is a vector valued quantity. In [12], for example, we describe numerical methods for the shallow water wave equations on the sphere.

3.3. Computing the area of mesh cells. To compute the average Laplacian over a mesh cell, one needs to have an approximation to the mesh cell area. For flat, quadrilateral grids, one can easily compute the area of mesh cells P_{ij} by triangulating the quadrilateral mesh cell and summing the areas of the resulting two triangles, or directly by using formulas for areas of polygons. For general curved surfaces, however, the first approach can lead to ambiguous values for the area, depending on how one does the triangulation. Direct formulas for computing the area of quadrilateral surface patches on general surfaces do not exist. Since we do not want to assume the existence of an underlying metric, we instead approximate the surface locally by assuming that the surface spanning the mesh cell has the bilinear function

$$(3.61) \quad S(u, v) = c_{00} + c_{01}u + c_{10}v + c_{11}uv, \quad 0 \leq u, v \leq 1,$$

where coefficients $c_{\ell k} \in \mathbf{R}^3$ are computed from known locations of the mesh cell corners. Using corner points $\hat{\mathbf{x}}_{i+k,j+\ell}$ for $k, \ell = 0, 1$, we get

$$(3.62) \quad \begin{aligned} c_{00} &= \hat{\mathbf{x}}_{ij}, & c_{01} &= \hat{\mathbf{x}}_{i+1,j} - \hat{\mathbf{x}}_{ij}, & c_{10} &= \hat{\mathbf{x}}_{i,j+1} - \hat{\mathbf{x}}_{ij}, \\ c_{11} &= \hat{\mathbf{x}}_{i+1,j+1} - \hat{\mathbf{x}}_{i+1,j} - \hat{\mathbf{x}}_{i,j+1} + \hat{\mathbf{x}}_{ij}. \end{aligned}$$

The area of the surface element spanned by $S(u, v)$ can then be computed by integrating the surface element. This leads to the approximation

$$(3.63) \quad \text{Area}(P_{ij}) \approx \int_0^1 \int_0^1 \|S_u \times S_v\| \, dudv = \int_0^1 \int_0^1 \|(c_{01} + c_{11}v) \times (c_{10} + c_{11}u)\| \, dudv.$$

To compute the right-hand-side integral accurately, we used a fourth order quadrature rule based on Gauss-Legendre nodes and weights.

4. Numerical accuracy study. In this section, we demonstrate the accuracy of the Laplace–Beltrami operator when used as a discretization for parabolic equations on general manifolds. The accuracy of the hyperbolic solver for the advection equation on a sphere was demonstrated in [12]. As already discussed, we will use the RKC solver for time stepping.

To test the scheme for general surfaces, we construct an artificial solution $q(\mathbf{x}, t)$ of the form

$$(4.1) \quad q(\mathbf{x}, t) = A(t)Q_e(\mathbf{x}),$$

where \mathbf{x} is a point on the surface. Then $q(\mathbf{x}, t)$ satisfies

$$(4.2) \quad q_t = \nabla^2 q + \frac{dA}{dt} Q_e(\mathbf{x}) - A(t) \nabla^2 Q_e(\mathbf{x}).$$

For the test cases considered here, we use the simple time component

$$(4.3) \quad A(t) = \exp(-5t).$$

While this does not rigorously test the accuracy of the time marching scheme, our focus here is to test the Laplace–Beltrami discretization. For the flat disk, we define the spatial component $Q_e(\mathbf{x})$ as

$$(4.4) \quad Q_e(\mathbf{x}) = \sum_{k=1}^K \exp(-10 r_k^2), \quad r_k = \|\mathbf{x} - \mathbf{w}_k\|,$$

where \mathbf{w}_k is a point on the disk. For the sphere, we use a similar function given by

$$(4.5) \quad Q_e(\mathbf{x}) = \sum_{k=1}^K \exp(-10\theta_k^2), \quad \theta_k = \cos(\mathbf{w}_k \cdot \mathbf{x}),$$

where \mathbf{w}_k is a unit vector normal to the surface of the sphere. To avoid inadvertently biasing the distribution of Gaussians toward favorable locations on our disk and sphere meshes, we choose the \mathbf{w}_k and θ_k randomly.

To evaluate the source term in (4.2), we have to be able to evaluate $\nabla^2 Q_e(\mathbf{x})$ over each mesh cell for general surfaces. Since our discretization of the parabolic equation is based on an integral form of the equations, we can take advantage of the divergence theorem to approximate the Laplacian in the source term as

$$(4.6) \quad \nabla^2 Q_e(\mathbf{x}_{ij}) \approx \frac{1}{\text{Area}(P_{ij})} \int_{P_{ij}} \nabla^2 Q_e(\mathbf{x}, t) dS = \frac{1}{\text{Area}(P_{ij})} \int_{\partial P_{ij}} \nabla Q_e(\mathbf{x}) \cdot \boldsymbol{\nu} dL.$$

To obtain an accurate approximation, we compute normal fluxes (given by (3.13)) along cell edges using known analytic metric terms and then integrate these fluxes along the four cell edges of S using a high order Gauss quadrature rule. This procedure allows us to test the accuracy of our scheme for any surface for which we can easily obtain the vector $\mathbf{t}^{(\alpha)}$, defined in (3.8).

Using the RKC solver described in section 3.1.2, we time step the equation given in (4.2). If we let $Q_{ij}(t)$ be the average values of $q(\mathbf{x}, t)$ over the primal mesh cell P_{ij} , then a spatial discretization of (4.2) can be written as

$$(4.7) \quad \frac{d}{dt} Q_{ij}(t) = L_{ij}(Q(t)) + \frac{dA}{dt} Q_e(\mathbf{x}) - \frac{A(t)}{\text{Area}(P_{ij})} \int_{\partial P_{ij}} \nabla Q_e(\mathbf{x}) \cdot \boldsymbol{\nu} dL,$$

where the area used to compute both $L_{ij}(Q)$ and the source term is computed using the bilinear approximation given in section 3.3. We compute the solution to time $T_{final} = 0.2$ using a time step $\Delta t = \Delta\xi = \Delta\eta$.

Errors at $T = T_{final}$ at each grid point are computed by comparing our computed solution to the exact solution evaluated at the primal points. This error is given by

$$(4.8) \quad e_{ij}(T) = Q_{ij}^N - A(T) Q_e(\mathbf{x}_{ij}).$$

We compute the 1-norm and inf-norm of the grid function e_{ij} as

$$(4.9) \quad \|e_{ij}(T)\|_1 = \frac{1}{\text{Total Area}} \sum_{i,j} |e_{ij}(T)| \text{Area}(P_{ij})$$

and

$$(4.10) \quad \|e_{ij}(T)\|_\infty = \max_{i,j} |e_{ij}(T)|,$$

where the total area is the surface area of the disk, hemisphere, or sphere.

4.1. Errors on the disk, hemisphere, and sphere. In this section, we show convergence rates for the disk, the hemisphere, and the sphere. While our choice of artificial solution allows us to test more general mappings, we choose to focus on these three to demonstrate the accuracy of our boundary conditions for the disk and hemisphere, and the accuracy for the boundary conditions for handling the discontinuities

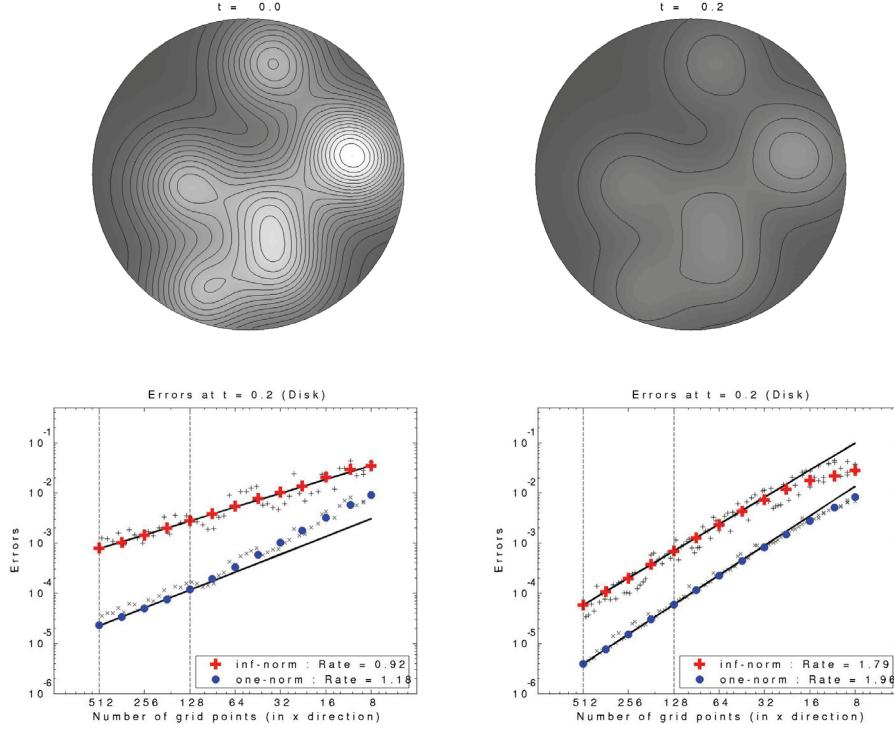


FIG. 4.1. The top set of plots show the initial conditions and final solution used to test convergence of the present numerical scheme for solving a general parabolic equation with source term on the disk map. The plot at the top left shows the initial distribution of randomly placed Gaussians in (4.4), and the top right plot shows the solution at time $T_{final} = 0.2$. The solution was computed on a 128×128 grid. The bottom set of plots show convergence rates for the solution on the disk. The bottom left plot shows the rates for the scheme in which no special averaging was used to obtain dual cell values in diagonal cells. The bottom right-hand plot shows the convergence rate if the averaging described in (3.34) is used. The solid circles and crosses are the errors for the solution computed using the initial conditions shown in the top panels. The scattered "x" and "+"'s are the errors obtained when these initial conditions are rotated, with respect to the underlying mesh, through arbitrary angles about the origin. The solid black lines are the best-fit lines through the errors shown by the solid symbols. For the left plot the convergence rates are 0.92 for the inf-norm and 1.18 for the 1-norm. For the right plot, the convergence rates are 1.79 for the inf-norm and 1.96 for the 1-norm. See Tables 4.1 and 4.2 for a subset of the values shown in these plots.

along the equator of our sphere mapping. We expect that applying our discretization and boundary condition to more general smooth mappings should lead to results similar to those we show here.

For the convergence test, we choose the \mathbf{w}_k 's and the θ_k 's in (4.4) and (4.5) randomly. For the disk, we set $K = 8$ (i.e., we distribute eight Gaussians over the unit disk); for the hemisphere, we set $K = 15$; and for the sphere, we set $K = 23$. To obtain the convergence rate for a particular mapping, we fix the random placement of Gaussians for a series of 12 grids ranging in size from $M_y = 8$ to $M_y = 512$. The initial conditions for the particular random solution chosen are shown in Figures 4.1, 4.2, and 4.3. In those figures, we also show the solution at time $T_{final} = 0.2$, the time at which we compute the errors. For the time step required by the RKC solver, we choose $\Delta t = \Delta\xi$. For the disk and hemisphere grids, we impose flux boundary conditions consistent with our artificial solution.

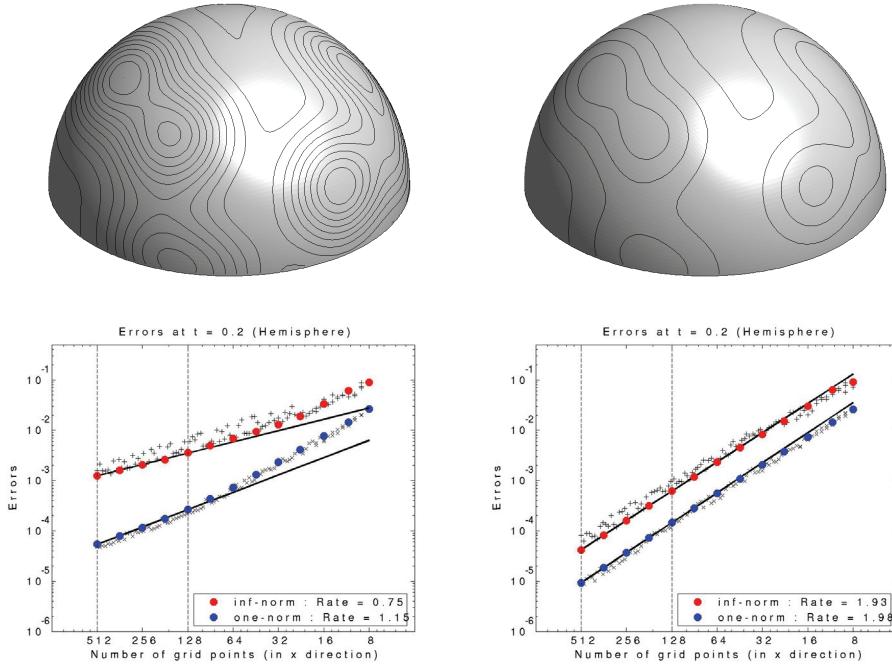


FIG. 4.2. *Initial conditions, final solution, and convergence rates for the hemisphere example.* The convergence rates for the scheme in which no special averaging was used in the diagonal cells (bottom left plot) is 0.75 for the \inf -norm and 1.15 for the 1-norm. When diagonal averaging was used, the convergence rates are 1.93 for the \inf -norm and 1.98 for the 1-norm. See Figure 4.1 for a detailed description of each plot, and Tables 4.1 and 4.2 for a subset of the values shown in these plots.

In Figures 4.1, 4.2, and 4.3, we show the errors computed at our final time $T_{final} = 0.2$ using initial conditions described above. We also present a subset of our error results in Tables 4.1 and 4.2. The convergence rates are computed as the best-fit line through errors on the five grids in the range $[128, 512]$, the range we believe exhibits the asymptotic behavior of the numerical scheme. To see the effects of the mesh on these errors, we also compute the errors that result when we rotate the initial conditions through a random angle about the z -axis for the disk and hemisphere grids, or through an arbitrary axis through the center of the sphere grid. These errors are shown on the same plots as the convergence results described above.

In all three convergence plots, we see that our scheme, when used with the modifications we describe for the disk, hemisphere, and sphere, is nearly second order in the 1-norm (1.96, 1.98, and 1.98 for the disk, hemisphere, and sphere, respectively). The \inf -norms are slightly less than second order (1.79, 1.93, and 1.92, respectively). Moreover, we see that the magnitude of the error is relatively insensitive to the location of the solution relative to the underlying mesh. The errors obtained using rotated initial conditions do not show much variation, suggesting that the presence of the near-triangular cells on the diagonals does not have significant impact on the errors.

Finally, to see the importance of taking into account the metric discontinuities when computing vertex values, we also compute solutions without the modifications we described to account for these discontinuities. For the disk and hemisphere grids,

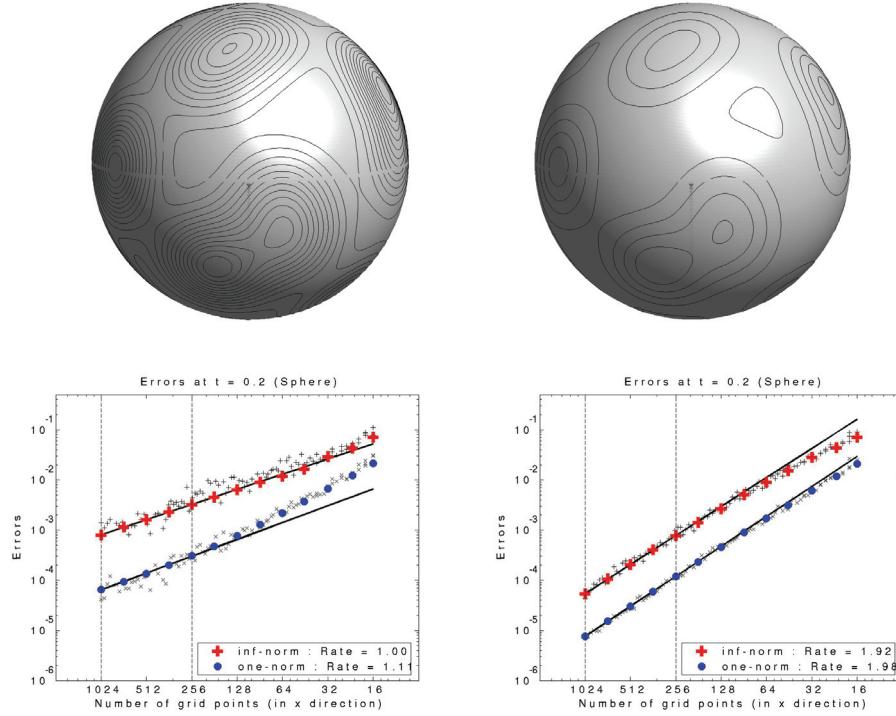


FIG. 4.3. *Initial conditions, final solution and convergence rates for the sphere example. The convergence rates for the scheme in which no special averaging was used for dual cells along the equator (bottom left plot) are 1.00 for the inf-norm and 1.11 for the 1-norm. For the right plot, the convergence rates are 1.92 for the inf-norm and 1.98 for the 1-norm. See Figure 4.1 for a detailed description of each plot and Tables 4.1 and 4.2 for a subset of values shown in these plots.*

we consider the effects of omitting the modification given in (3.34). For the sphere grid, we consider only the effect of omitting the internal boundary conditions described in section 3.1.4. From the plots in which we omit the diagonal averaging and the equator modification, we see that the solutions are only first order. These results are consistent with the theoretical results presented in [8] for anisotropic diffusion problems with discontinuities.

The second order accuracy of the hyperbolic solver for smooth solutions of the advection equation on the sphere was confirmed in [11]. Accuracy studies for the shallow water equations on the sphere can be found in [12].

4.2. Efficiency of the RKC solver. The numerical method we describe here for solving the parabolic equation on the disk, hemisphere, and sphere relies on the fact that we can efficiently update the solution using the explicit RKC time stepping scheme. To assess the efficiency of this scheme for our problem, we compared it with forward Euler and the TR-BDF2 scheme, an implicit one-step two-stage L-stable scheme based on a hybrid of the trapezoidal rule and a second order backwards differentiation formula [33]. To measure the efficiency of each of the three update methods, we tracked the number of function calls required for each time stepping scheme to reach a fixed time $T_{final} = 0.2$. All of the simulations were done on the sphere example.

TABLE 4.1

1-norm errors for the disk, hemisphere, and sphere solutions shown in Figures 4.1, 4.2, and 4.3. The convergence rates between successive grids are shown in parenthesis. The errors shown here are a subset of those shown in Figures 4.1, 4.2, and 4.3, with the $O(h)$ solutions corresponding to the left plots in these figures, and the $O(h^2)$ plots corresponding to the right convergence plots in these figures.

$\ e(T)\ _1$	Disk		Hemisphere		Sphere	
M_y	$O(h)$	$O(h^2)$	$O(h)$	$O(h^2)$	$O(h)$	$O(h^2)$
64	3.30e-4	2.25e-4	7.32e-4	5.55e-4	7.66e-4	4.58e-4
(32/64)	(1.64)	(1.87)	(1.67)	(1.89)	(1.50)	(1.90)
128	1.19e-4	5.90e-5	2.69e-4	1.44e-4	3.05e-4	1.19e-4
(64/128)	(1.47)	(1.93)	(1.44)	(1.95)	(1.32)	(1.94)
256	4.95e-5	1.51e-5	1.16e-4	3.67e-5	1.36e-4	3.01e-5
(128/256)	(1.27)	(1.97)	(1.21)	(1.97)	(1.17)	(1.98)
512	2.29e-5	3.90e-6	5.48e-5	9.24e-6	6.50e-5	7.58e-6
(256/512)	(1.11)	(1.95)	(1.08)	(1.99)	(1.07)	(1.99)

TABLE 4.2

Inf-norm errors for the disk, hemisphere, and sphere solutions shown in Figures 4.1, 4.2, and 4.3. Notation as in Table 4.1.

$\ e(T)\ _\infty$	Disk		Hemisphere		Sphere	
M_y	$O(h)$	$O(h^2)$	$O(h)$	$O(h^2)$	$O(h)$	$O(h^2)$
64	5.35e-3	2.32e-3	6.78e-3	2.32e-3	6.36e-3	2.65e-3
(32/64)	(0.95)	(1.66)	(0.92)	(1.83)	(0.89)	(1.74)
128	2.77e-3	6.92e-4	3.55e-3	6.13e-4	3.21e-3	7.68e-4
(64/128)	(0.95)	(1.75)	(0.93)	(1.92)	(0.99)	(1.79)
256	1.42e-3	1.98e-4	2.06e-3	1.60e-4	1.60e-3	2.05e-4
(128/256)	(0.96)	(1.81)	(0.79)	(1.94)	(1.00)	(1.90)
512	7.89e-4	5.81e-5	1.24e-3	4.19e-5	8.00e-4	5.34e-5
(256/512)	(0.85)	(1.77)	(0.73)	(1.93)	(1.00)	(1.94)

The time step chosen for each scheme depended on the stability requirements of the scheme. Hence, for forward Euler, we used $\Delta t_{fe} = 0.1\Delta\xi^2$, and for the RKC solver and the implicit scheme, we used $\Delta t = \Delta\xi$. We ran the simulation on grid sizes ranging from $M_x = 16$ to $M_x = 1024$, and for each method we tracked how many function calls were needed to reach $T_{final} = 0.2$. For the explicit time stepping schemes, a “function call” involves those steps outlined in the right-hand-side algorithm. For the implicit solver, we modify this routine slightly so that it is a suitable matrix-vector multiply for BICG-stab, the iterative method we used to solve the two linear systems appearing in the TR-BDF method. All three time stepping schemes are second order, and for a given grid all three schemes produce essentially the same 1-norm error.

In Figure 4.4, we show the results of these simulations and see that for coarse grids (i.e., $M_x < 100$) the RKC solver required many more function calls than either forward Euler or the TR-BDF. This may be partially explained by the fact that we used the same absolute and relative tolerances (required parameters for the RKC

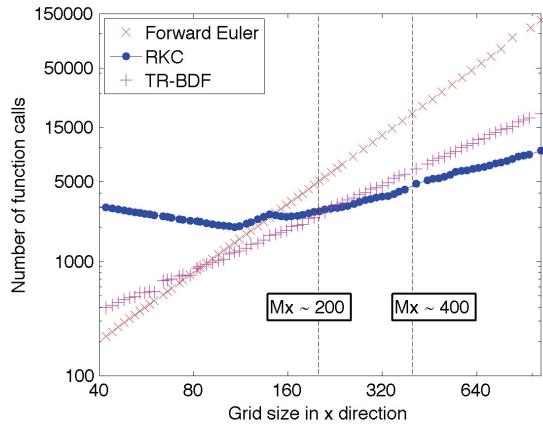


FIG. 4.4. Efficiency of the forward Euler, RKC, and implicit TR-BDF scheme for the sphere example. Each curve plots the number of function calls versus grid size in the x direction for the sphere. The labeled vertical lines correspond to calculations on a 200×100 sphere grid (approximately 2700 function calls needed for both the RKC solver and the implicit solver) and on a 400×200 grid (approximately 4800 function calls needed for the RKC solver versus 6300 for the implicit solver).

solver) for all grids. On coarse grids, these tolerances may have been too strict for the level of grid resolution, and so the RKC solver had to perform many more steps to achieve the desired level of accuracy.

Once the number of grid points in the x direction exceeded roughly 200 (i.e., $M_x > 200$), the RKC solver required far fewer function calls than either of the other two schemes. For example, on the 400×200 grid, the RKC solver required about 4800 function calls, whereas TR-BDF required about 6300, and forward Euler about 20000. We conclude that the RKC solver is an attractive time stepping approach for our spatial discretization and is superior not only to the standard explicit approach, but to implicit solvers as well.

5. Applications.

5.1. Chemotaxis in a petri dish. Chemotaxis is an interesting example of pattern formation that can be observed for bacteria. We refer to Murray [41, Chapter 5] and Tyson and coworkers [54, 55] for detailed analytical and numerical investigations of models that are based on experimental studies by Budrene and Berg [9, 10]. A key property of many bacteria is that in the presence of certain chemicals (chemoattractants), they move preferentially towards higher concentration of the chemical [41]. Here we restrict our consideration to simulations of a simplified nondimensionalized model [41, section 5.8], whose numerical discretization already contains all the basic components needed in more complex models.

Numerical simulations for chemotaxis models appearing in the literature have been typically carried out on rectangular domains with periodic or no-flux boundary conditions [55, 13]. This geometry is obviously chosen because it simplifies the construction of numerical schemes. By using our disk grids along with the method outlined in section 3.2, we can perform simulations which more closely resemble the results of experiments carried out in a circular petri dish.

The mathematical model is a system of PDEs of mixed type containing compo-

nents that model transport, diffusion, and reactions and is given by

$$(5.1) \quad \begin{aligned} \frac{\partial u}{\partial t} &= d_u \nabla^2 u - \alpha \nabla \cdot \left(\left(\frac{\nabla v}{(1+v)^2} \right) u \right) + \rho u(\delta - u), \\ \frac{\partial v}{\partial t} &= \nabla^2 v + \beta u^2 - uv. \end{aligned}$$

The variables u and v (which depend on position and time) describe concentrations of cell density and chemoattractant, respectively. Diffusion terms model the random motion of each component. The chemotaxis term (the second term on the right-hand side of the first equation in (5.1)) can be viewed as an advection term. It models a directed motion of the cell density u in response to a concentration gradient of v . Reaction terms model the interaction of the different components, e.g., growth of cells, release of chemoattractant, or consumption of nutrients.

In this model we have the free parameters α , β , δ , ρ , and d_u . Depending on the values of these parameters, one can observe different patterns, either concentric rings or spotted rings. For the simulations we perform here, we use fixed parameter values

$$d_u = 0.25, \quad \beta = 0.2, \quad \delta = 20, \quad \rho = 0.01$$

and set the parameter value α to either $\alpha = 2.25$ (continuous rings) or $\alpha = 5$ (spotted rings). We set the radius of the petri dish to $R = 15\sqrt{2}$. For our simulations, we initialize the bacteria distribution by computing mesh cell averages of a function which is 1 inside a circle of radius 0.5, and 0 outside this circle. The initial distribution of bacteria is set to these cell-averaged values. Initially, there is no chemoattractant present.

In Figure 5.1 we show results obtained by solving (5.1) for the above choice of parameter values. We made two sets of simulations, one on our disk grids, described in section 2 and shown in Figure 2.1, and one on the optimized smooth grid, also shown in Figure 2.1. Both grids are 150×150 . We use our algorithm directly on the smooth grid, since the computational domain is again a single logically Cartesian grid, and we do not require the analytic mapping. For the smooth grid, we assume that there are no metric discontinuities. We show numerical results for both striped and spotted patterns on both grids at a fixed time $T = 75$.

The numerical results show that the radial symmetry of the solution is well approximated even on our relatively coarse grid. Furthermore, we see little difference in the results on our disk grid and those on the smooth grid, suggesting that our algorithm and grids are robust, despite the presence of metric discontinuities. Finally, the boundary conditions are not adversely affecting the solution, unlike boundary conditions imposed on the boundaries of rectangular domains.

5.2. Turing patterns on general curved surfaces. Reaction-diffusion equations arise in a large variety of mathematical models describing pattern formation for biological applications [41, 40, 38, 4]. Most results reported in the literature discuss pattern formation on flat two-dimensional surfaces. The few efforts in the biology community to simulate patterns on curved surfaces have been for very simple surfaces such as cones or spheres [44, 56]. The solution techniques presented for these simple shapes all rely on knowledge of the parameterization, making it difficult to extend those methods to more general surfaces. To demonstrate that our diffusion scheme can easily solve problems on general smooth surfaces, we present the results of calculations of Turing patterns on the supershape presented in section 2.3.

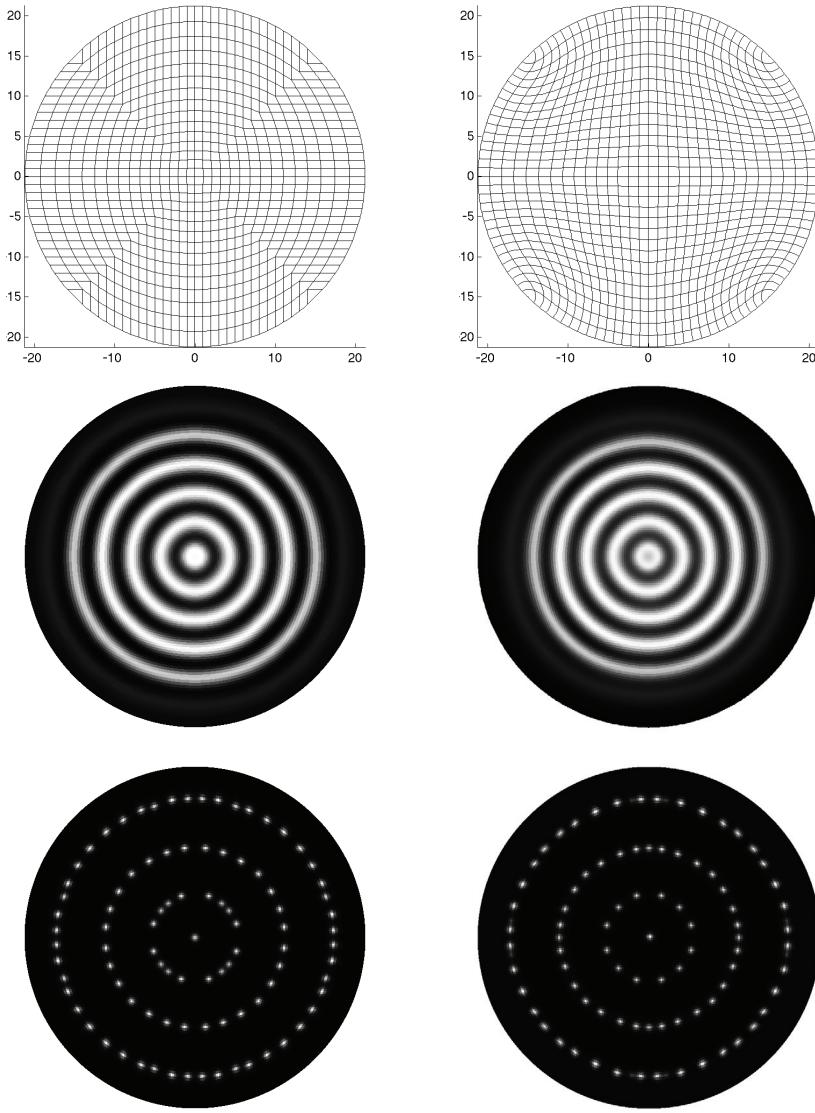


FIG. 5.1. Formation of concentric rings of stripes (middle) or spots (bottom) for chemotaxis in a petri dish computed on a 150×150 grid. The solutions in the left column were computed on the grid at the top left, and those in the right column were computed using the smooth optimized grid on the right. For each solution, the u species is shown at time $t = 75$.

We consider Turing models with two interacting species u and v using the model equations from [5, 56] which have the form

$$(5.2) \quad \begin{aligned} \frac{\partial u}{\partial t} &= D\delta\nabla^2 u + \alpha u(1 - \tau_1 v^2) + v(1 - \tau_2 u), \\ \frac{\partial v}{\partial t} &= \delta\nabla^2 v + \beta v \left(1 + \frac{\alpha\tau_1}{\beta} uv\right) + u(\gamma + \tau_2 v), \end{aligned}$$

where $(u, v) = (0, 0)$ is a unique spatially uniform steady state for $\gamma = -\alpha$. Turing

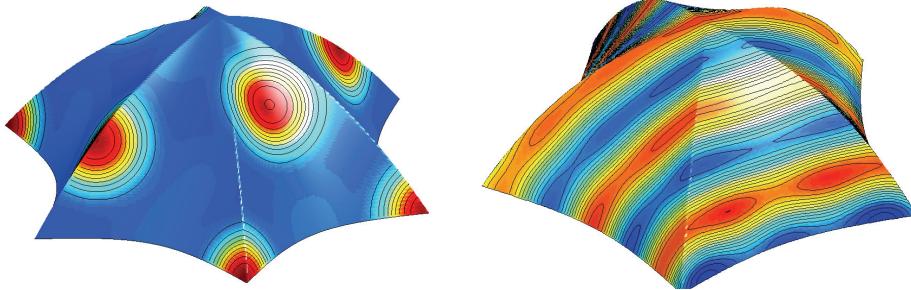


FIG. 5.2. Spot and stripe patterns computed on 200×200 grids. Parameter values used for the spots in (5.2): $\delta = 0.0045$, $D = 0.516$, $\tau_1 = 0.02$, $\tau_2 = 0.2$, $\alpha = 0.899$, $\beta = -0.91$, $\gamma = -\alpha$. Parameter values used for the stripes in (5.2): $\delta = 0.0021$, $D = 0.516$, $\tau_1 = 3.5$, $\tau_2 = 0$, $\alpha = 0.899$, $\beta = -0.91$, $\gamma = -\alpha$.

showed that for certain parameter values the steady state was linearly stable in the absence of diffusion but unstable in the presence of diffusion. This diffusion-driven instability leads to the formation of different patterns, including striped and spotted patterns.

For our calculations, we solve the above coupled system of parabolic equations on two different supershapes—one with five symmetric arms and one with six. To initialize the simulations, we initialized u and v with random values between $-1/2$ and $1/2$. On the physical boundaries of the supershape, we imposed no-flux boundary conditions. We ran the simulation until a steady state solution was reached. The results of the calculations on each supershape are shown in Figure 5.2. The left plot shows particular regularity, especially with respect to the location of spots at the tips of the supershape arms. In the right plot, we see the significant effect that the boundary conditions have on the resulting pattern. The interaction between the no-flux boundary conditions and the nonconstant curvature of the surface appear to be constraining the steady state solution in such a way that the striped pattern exhibits breaks or defects.

6. Conclusions. We have described a finite volume discretization for parabolic problems on smooth surfaces parameterized using a quadrilateral grid. The standard version of the algorithm is based on an update of cell centered values (i.e., average values on primal cells) and an averaging procedure to calculate vertex values (i.e., average values on dual mesh cells) and can be applied to general smooth or piecewise smooth grids.

We have in particular considered grids for the disk and the sphere, which were introduced in [12]. While these quadrilateral grids have some valuable properties, the mapping is nonsmooth in some regions. In order to retain second order accuracy for the approximation of parabolic problems, we have introduced a suitable calculation of dual values. Also, we implemented general Robin boundary conditions. We provided numerical evidence for the second order convergence of our scheme for the solution to parabolic test problems on a disk, hemisphere, and sphere grid. We also showed that our method works quite well for simulating chemotaxis in a petri dish and reaction-diffusion on a general surface mesh.

While we demonstrated our diffusion algorithm for the sphere grids that we introduced in [12], our method should extend quite naturally to other parameterizations

of the sphere. In particular, there is increasing interest in the atmospheric community in the cubed sphere grids introduced by Sadorny [47] (see also [45]) in place of latitude-longitude grids. The cubed sphere grid is also piecewise smooth, and so the techniques introduced here for preserving second order accuracy should work equally well on that grid.

Finally, we hope that the solution technique presented here will appeal to computational biologists interested in simulating pattern formation on general surfaces. It would be especially interesting to reproduce the patterns on insect wings and fish described in [35, 5, 6]. To aid in this and other research efforts, our code will be made available to interested researchers.

Appendix A. Connection to the “cotan” formula. Here, we discuss the connection between our approximation in (3.27) and the “cotan” formula, widely used in the graphics and geometric-aided design communities for approximating the Laplace–Beltrami operator on triangular meshes.

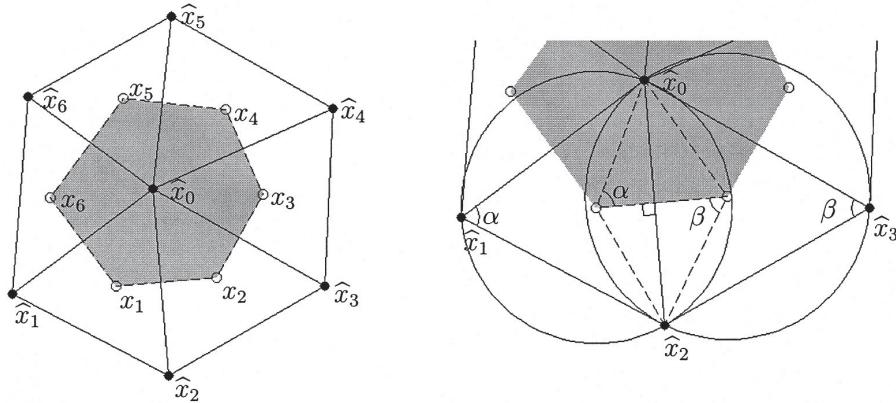


FIG. A.1. Flat triangular mesh showing a dual cell (shaded polygonal cell) over which one can approximate the Laplacian at the vertex point x_0 . The right plot shows the angles that appear in the cotan formula, a widely used approximation to the Laplace–Beltrami operator on triangular surface meshes.

In practice, the cotan formula is viewed as an approximation to the surface Laplacian at vertices of a triangular mesh. However, it can also be viewed as a finite volume discretization over a dual mesh cell, such as the one shown by the shaded area in Figure A.1 [18, 57, 26]. Under the assumption that we have Delaunay triangulation in which all triangles are acute, the circumcenter of each triangle is interior to the triangle and so can be chosen as primal node for that triangle. Adapting the flux formula we derived in (3.26) to the mesh shown in Figure A.1, we express the outgoing normal flux at edge $[x_1, x_2]$ as

$$(A.1) \quad \int_{[x_1, x_2]} \frac{\partial q}{\partial n} dL \approx \frac{|x_1 - x_2|}{|\hat{x}_0 - \hat{x}_2|} (q(\hat{x}_2) - q(\hat{x}_0))$$

since $\theta = \pi/2$. For the flat mesh shown in Figure A.1, it is easy to show that the ratio of these lengths reduces to

$$(A.2) \quad \frac{|x_1 - x_2|}{|\hat{x}_0 - \hat{x}_2|} = \frac{1}{2} (\cot \alpha_{0,2} + \cot \beta_{0,2}),$$

where $\alpha_{0,2}$ and $\beta_{0,2}$ are the angles at vertices \hat{x}_1 and \hat{x}_3 , respectively, shown in Figure A.1(right). Constructing fluxes at each edge of the dual cell and summing them up leads to the cotan formula for the integral of the Laplacian over the dual mesh cell shown in Figure A.1. This formula is given by

$$(A.3) \quad \int_{D_0} \nabla^2 q \, dA \approx \sum_{j=1}^6 \frac{1}{2} (\cot(\alpha_{0,j}) + \cot(\beta_{0,j})) (q(\hat{x}_j) - q(\hat{x}_0)).$$

The cotan formula for a vertex \hat{x}_i on a general triangular mesh is given by

$$(A.4) \quad \int_{D_i} \nabla^2 q \, dA \approx \sum_{j \in N(i)} \frac{1}{2} (\cot(\alpha_{ij}) + \cot(\beta_{ij})) (q(\hat{x}_j) - q(\hat{x}_i)),$$

where $N(i)$ is the set of vertices in a one-ring neighborhood around \hat{x}_i .

Appendix B. Connection to other schemes for flat unstructured meshes.

For flat manifolds, the approximation to the Laplacian in (3.27) is essentially the same as that described by Hermeline and others for general unstructured meshes [27, 19, 15]. To see the connection between our scheme and the one presented by Hermeline in [27], we define unit tangent vectors $\boldsymbol{\tau}_p$ and $\boldsymbol{\tau}_d$ at the right edge of the primal cell P_{ij} as

$$(B.1) \quad \boldsymbol{\tau}_p := \frac{\mathbf{t}_{i+1/2,j}}{|S_{i+1/2,j}|}, \quad \boldsymbol{\tau}_d := \frac{\widehat{\mathbf{t}}_{i+1/2,j}}{|\widehat{S}_{i+1/2,j}|}.$$

For flat, polygonal meshes, the normal vector $\boldsymbol{\nu}$ to the right edge of P_{ij} is unique and can be written as

$$(B.2) \quad \boldsymbol{\nu} = a \boldsymbol{\tau}_d + b \boldsymbol{\tau}_p.$$

From this expression for the normal, we can obtain an approximation to the integral of the flux over a straight edge as

$$(B.3) \quad \begin{aligned} \int_{S_{i+1/2,j}} \nabla q \cdot \boldsymbol{\nu} \, dL &= a \int_{S_{i+1/2,j}} \nabla q \cdot \boldsymbol{\tau}_d \, dL + b \int_{S_{i+1/2,j}} \nabla q \cdot \boldsymbol{\tau}_p \, dL \\ &\approx a \frac{|S_{i+1/2,j}|}{|\widehat{S}_{i+1/2,j}|} \Delta q_{i+1/2,j} + b \Delta \widehat{q}_{i+1/2,j}. \end{aligned}$$

To determine coefficients a and b , we construct the linear system

$$(B.4) \quad \begin{pmatrix} \boldsymbol{\nu} \cdot \boldsymbol{\tau}_d \\ \boldsymbol{\nu} \cdot \boldsymbol{\tau}_p \end{pmatrix} = \begin{pmatrix} 1 & \cos(\theta) \\ \cos(\theta) & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \cos(\phi) \\ 0 \end{pmatrix},$$

where $\phi = \pi/2 - \theta$ is the angle between $\boldsymbol{\tau}_d$ and $\boldsymbol{\nu}$ and θ is the angle defined as in (3.25). Replacing $\cos(\phi)$ in (B.4) with $\sin(\theta)$ and solving for a and b , we get

$$(B.5) \quad \boldsymbol{\nu} = \csc(\theta) \boldsymbol{\tau}_d - \cot(\theta) \boldsymbol{\tau}_p,$$

from which we can recover our normal flux approximation in (3.26). Making the substitution $\cos(\theta) = \sin(\phi)$ leads to

$$(B.6) \quad \boldsymbol{\nu} = \sec(\phi) \boldsymbol{\tau}_d - \tan(\phi) \boldsymbol{\tau}_p,$$

from which we obtain exactly Hermeline's flux approximation, used in [27] for polygonal, flat meshes.

Acknowledgments. We wish to thank François Hermeline for showing us the connection between (3.27) and the cotan formula, Stéphane Gounand (LTMF/CEA) for generating the smooth grid shown in Figure 2.1, and Christophe Le Potier (SMTS/CEA) for many useful discussions on diamond-cell and DDFV schemes. Finally, we thank Randall J. LeVeque and Jing-Rebecca Li for their careful reading of an early draft and their many useful comments.

REFERENCES

- [1] I. AAVATSMARK, G. T. EIGESTAD, B. T. MALLISON, AND J. M. NORDBOTTEN, *A compact multipoint flux approximation method with improved robustness*, Numer. Methods Partial Differential Equations, 24 (2008), pp. 1329–1360.
- [2] D. ADALSTEINSSON AND J. A. SETHIAN, *Transport and diffusion of material quantities on propagating interfaces via level set methods*, J. Comput. Phys., 185 (2003), pp. 271–288.
- [3] B. ANDREIANOV, F. BOYER, AND F. HUBERT, *Discrete duality finite volume schemes for Leray–Lions-type elliptic problems on general 2D meshes*, Numer. Methods Partial Differential Equations, 23 (2007), pp. 145–195.
- [4] R. E. BAKER, E. A. GAFFNEY, AND P. K. MAINI, *Partial differential equations for self-organization in cellular and developmental biology*, Nonlinearity, 21 (2008), pp. 251–290.
- [5] R. BARRIOS, C. VAREA, J. ARAGÓN, AND P. MAINI, *A two-dimensional numerical study of spatial pattern formation in interacting Turing systems*, Bull. Math. Biol., 61 (1999), pp. 483–505.
- [6] R. A. BARRIOS, R. E. BAKER, B. VAUGHAN, K. TRIBUZY, M. R. DE CARVALHO, R. BASSANEZI, AND P. K. MAINI, *Modeling the skin pattern of fishes*, Phys. Rev. E, 79 (2009), paper 031908.
- [7] M. BERTALMIO, L. CHENG, S. OSHER, AND G. SAPIRO, *Variational problems and partial differential equations on implicit surfaces*, J. Comput. Phys., 174 (2001), pp. 759–780.
- [8] F. BOYER AND F. HUBERT, *Finite volume method for 2D linear and nonlinear elliptic problems with discontinuities*, SIAM J. Numer. Anal., 46 (2008), pp. 3032–3070.
- [9] E. O. BUDRENE AND H. BERG, *Complex patterns formed by motile cells in Escherichia coli*, Nature, 349 (1991), pp. 630–633.
- [10] E. O. BUDRENE AND H. BERG, *Dynamics of formation of symmetrical patterns of chemotactic bacteria*, Nature, 376 (1995), pp. 49–53.
- [11] D. A. CALHOUN, C. HELZEL, AND R. J. LEVEQUE, *A finite volume grid for solving hyperbolic problems on the sphere*, in Hyperbolic Problems: Theory, Numerics, Applications, Proceedings of the 11th International Conference on Hyperbolic Problems, Lyon, 2006, Springer, Berlin, 2008, pp. 355–362.
- [12] D. A. CALHOUN, C. HELZEL, AND R. J. LEVEQUE, *Logically rectangular grids and finite volume methods for PDEs in circular and spherical domains*, SIAM Rev., 50 (2008), pp. 723–752.
- [13] A. CHERTOCK AND A. KURGANOV, *A second-order positivity preserving central-upwind scheme for chemotaxis and haptotaxis models*, Numer. Math., 111 (2008), pp. 169–205.
- [14] W. J. COIRIER, *An Adaptively-Refined, Cartesian, Cell-Based Scheme for the Euler and Navier–Stokes Equations*, Ph.D. thesis, Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 1994.
- [15] Y. COUDIÈRE, T. GALLOUËT, AND R. HERBIN, *Discrete Sobolev inequalities and L_p error estimates for finite volume solutions of convection diffusion equations*, ESAIM Math. Model. Numer. Anal., 35 (2004), pp. 767–778.
- [16] Y. COUDIÈRE, J. P. VILA, AND P. VILLEDIEU, *Convergence rate of a finite volume scheme for a two dimensional convection-diffusion problem*, ESAIM Math. Model. Numer. Anal., 33 (1999), pp. 493–516.
- [17] A. DEMLOW AND G. DZIUK, *An adaptive finite element method for the Laplace–Beltrami operator on implicitly defined surfaces*, SIAM J. Numer. Anal., 45 (2007), pp. 421–442.
- [18] M. DESBRUN, M. MEYER, P. SCHRÖDER, AND A. BARR, *Implicit fairing of irregular meshes using diffusion and curvature flow*, in Proceedings of the 26th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 99), Los Angeles, CA, ACM, New York, 1999, pp. 317–324.
- [19] K. DOMELEVO AND P. OMNES, *A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids*, ESAIM Math. Model. Numer. Anal., 39 (2005), pp. 1203–1249.
- [20] G. DZIUK, *Finite elements for the Beltrami operator on arbitrary surfaces*, in Partial Differential Equations and Calculus of Variations, S. Hildebrandt and R. Leis, eds., Lecture Notes in Math. 1357, Springer, New York, 1988, pp. 142–155.

- [21] G. DZIUK AND C. M. ELLIOT, *Surface finite elements for parabolic equations*, J. Comput. Math., 25 (2007), pp. 385–407.
- [22] R. EYMARD, T. GALLOUËT, AND R. HERBIN, *The finite volume method*, in *Handbook of Numerical Analysis*, Vol. VII, P. Ciarlet and J. L. Lions, eds., North-Holland, Amsterdam, 2000, pp. 713–1020.
- [23] J. GIELIS, *A generic geometric transformation that unifies a wide range of natural and abstract shapes*, Amer. J. Botany, 90 (2003), pp. 333–338.
- [24] J. GREER, *An improvement of a recent Eulerian method for solving PDEs on general geometries*, J. Sci. Comput., 29 (2006), pp. 321–352.
- [25] R. HERBIN AND F. HUBERT, *Benchmark on Discretization Schemes for Anisotropic Diffusion Problems on General Grids*, in *Finite Volumes for Complex Applications V: Problems and Perspectives*, R. Eymard and J.-M. Hérard, eds., ISTE/Wiley, New York, 2008, pp. 659–692.
- [26] F. HERMELINE, *private discussions*, CEA/DIF, DSSI/SNEC, Commissariat à l’Energie Atomique (C.E.A.), Bruyères le Châtel, France, 2007.
- [27] F. HERMELINE, *A finite volume method for the approximation of diffusion operators on distorted meshes*, J. Comput. Phys., 160 (2000), pp. 481–499.
- [28] F. HERMELINE, *Approximation of diffusion operators with discontinuous tensor coefficients on distorted meshes*, Comput. Methods Appl. Mech. Engrg., 192 (2003), pp. 1939–1959.
- [29] W. HUANG, *Variational mesh adaptation: Isotropy and equidistribution*, J. Comput. Phys., 174 (2001), pp. 903–924.
- [30] W. P. JONES AND K. R. MENZIES, *Analysis of the cell-centered finite volume method for the diffusion equation*, J. Comput. Phys., 165 (2000), pp. 45–68.
- [31] C. LEPOTIER, *Schéma volumes finis monotone pour des opérateurs de diffusion fortement anisotropes sur des maillages de triangles nonstructurés*, C. R. Acad. Sci. Paris, 341 (2005), pp. 787–792.
- [32] R. J. LEVEQUE, *CLAWPACK*, software package, available online at <http://www.amath.washington.edu/~claw/>.
- [33] R. J. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, UK, 2002.
- [34] R. J. LEVEQUE, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, SIAM, Philadelphia, 2007.
- [35] S. S. LIAW, C. C. YANG, R. T. LIU, AND J. T. HONG, *Turing model for the patterns of lady beetles*, Phys. Rev. E, 64 (2001), paper 041909.
- [36] K. LIPNIKOV, M. SHASHKOV, D. SVYATSKII, AND Y. VASSILEVSKI, *Monotone finite volume schemes for diffusion equations on unstructured triangular and shape-regular polygonal meshes*, J. Comput. Phys., 227 (2007), pp. 492–512.
- [37] D. LIU, G. XU, AND Q. ZHANG, *A discrete scheme of Laplace–Beltrami operator and its convergence over quadrilateral meshes*, Comput. Math. Appl., 55 (2008), pp. 1081–1093.
- [38] P. K. MAINI AND H. G. OTHMER, EDS., *Mathematical Models for Biological Pattern Formation*, IMA Vol. Math. Appl. 121, Springer, New York, 2000.
- [39] M. MEYER, M. DESBRUN, P. SCHRÖDER, AND A. H. BARR, *Discrete differential geometry operators for triangulated 2-manifolds*, in *Mathematical Visualization*, Scientific Visualization, Numerical Mathematics, Scientific Computing, Computer Graphics, M. Peters, ed., Springer-Verlag, Heidelberg, Germany, 2003, pp. 35–57.
- [40] J. D. MURRAY, *Mathematical Biology I: An Introduction*, 3rd ed., Springer, New York, 2007.
- [41] J. D. MURRAY, *Mathematical Biology II: Spatial Models and Biomedical Applications*, 3rd ed., Springer, New York, 2008.
- [42] I. NOVAK, F. GAO, Y. CHOI, D. RESASCO, J. SCHAFF, AND B. SLEPCHENKO, *Diffusion on a curved surface coupled to diffusion in the volume: Application to cell biology*, J. Comput. Phys., 226 (2007), pp. 1271–1290.
- [43] U. PINKALL AND K. POLTHIER, *Computing discrete minimal surfaces and their conjugates*, Exper. Math., 2 (1993), pp. 15–36.
- [44] R. G. PLAZA, F. SÁNCHEZ-GARDUÑO, P. PADILLA, R. A. BARRO, AND P. K. MAINI, *The effect of growth and curvature on pattern formation*, J. Dynam. Differential Equations, 16 (2004), pp. 1093–1121.
- [45] W. M. PUTMAN AND S. J. LIN, *Finite-volume transport on various cubed-sphere grids*, J. Comput. Phys., 227 (2007), pp. 55–78.
- [46] S. RUUTH AND B. MERRIMAN, *A simple embedding method for solving partial differential equations on surfaces*, J. Comput. Phys., 227 (2008), pp. 1943–1961.
- [47] R. SADOURNY, *Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids*, Monthly Weather Rev., 100 (1972), pp. 211–224.

- [48] I. V. SBALZARINI, A. HAYER, A. HELENUS, AND P. KOUMOUTSAKOS, *Simulations of (an)isotropic diffusion on curved biological surfaces*, Biophys. J., 90 (2006), pp. 878–885.
- [49] P. SCHWARTZ, D. ADALSTEINSSON, P. COLELLA, A. P. ARKIN, AND M. ONSUM, *Numerical computation of diffusion on a surface*, Proc. Natl. Acad. Sci. USA, 102 (2005), pp. 11151–11156.
- [50] Z. SHENG AND G. YUAN, *A nine point scheme for the approximation of diffusion operators on distorted quadrilateral meshes*, SIAM J. Sci. Comput., 30 (2008), pp. 1341–1361.
- [51] B. SOMMEIJER, L. SHAMPINE, AND J. VERWER, *RKC: An explicit solver for parabolic PDEs*, J. Comput. Appl. Math., 88 (1997), pp. 315–326.
- [52] E. SÜLI, *Convergence of finite volume schemes for Poisson's equation on nonuniform meshes*, SIAM J. Numer. Anal., 28 (1991), pp. 1419–1430.
- [53] G. TURK, *Generating textures on arbitrary surfaces using reaction-diffusion*, Computer Graphics, 25 (1991), pp. 289–298.
- [54] R. TYSON, S. LUBKIN, AND J. D. MURRAY, *Model and analysis of chemotactic bacterial patterns in a liquid medium*, J. Math. Biol., 38 (1999), pp. 357–375.
- [55] R. TYSON, L. G. STERN, AND R. J. LEVEQUE, *Fractional step methods applied to a chemotaxis model*, J. Math. Biol., 41 (2000), pp. 455–475.
- [56] C. VAREA, J. ARAGON, AND R. BARRIO, *Turing patterns on a sphere*, Phys. Rev. E, 60 (1999), pp. 4588–4592.
- [57] G. XU, *Discrete Laplace–Beltrami operators and their convergence*, Comput. Aided Geom. Design, 21 (2004), pp. 767–784.