NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2016; **00**:1–17
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla

Composite-Grid Multigrid for Diffusion on the Sphere

James H. Adler[1], Ilya Lashuk[2], Scott P. MacLachlan[3]*

[1]*Department of Mathematics, Tufts University, 503 Boston Ave. Medford, MA 02155*
[2]*Department of Mathematics, The Pennsylvania State University, State College, PA 16802*
[3]*Department of Mathematics and Statistics, Memorial University of Newfoundland and Labrador, St. John's, Newfoundland and Labrador A1C 5S7, Canada*

SUMMARY

Recently, there has been much interest in the solution of differential equations on surfaces and manifolds, driven by many applications whose dynamics take place on such domains. While increasingly powerful algorithms have been developed in this field, many straightforward questions remain, particularly in the area of coupling advanced discretizations with efficient linear solvers. In this paper, we develop a structured refinement algorithm for octahedral triangulations of the surface of the sphere. We explain the composite-grid finite-element discretization of the Laplace-Beltrami operator on such triangulations, and extend the fast adaptive composite-grid (FAC) scheme to provide efficient solution of the resulting linear system. Supporting numerical examples are presented, including the recovery of second-order accuracy in the case of a non-smooth solution. Copyright © 2016 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: Laplace-Beltrami operator, multigrid, fast adaptive composite-grid (FAC) algorithm, finite-element discretizations on surfaces

1. INTRODUCTION

Discretization of scalar elliptic partial differential equations (PDEs) over regions in the plane or volumes in $\mathbb{R}^3$ is, in general, a well-understood subject, with viable finite-difference [1, 2], finite-element [3, 4], and finite-volume [5] approaches seen in standard textbooks and references. Mesh-refinement techniques are also well-understood in this setting, with numerous strategies and algorithms appearing in the literature (e.g. [6–11]). For standard scalar elliptic problems, on both uniform and refined meshes, solution of the resulting linear systems is typically accomplished efficiently using multigrid methods [12, 13], particularly algebraic multigrid (AMG) [14, 15].

A wide body of recent research has been devoted to extending classical discretization approaches to PDEs posed on surfaces and manifolds. Natural techniques that parameterize a surface based on regular coordinates in the plane are one option [16, 17], but are limited to simple surfaces and introduce problematic singularities, both in the underlying surface meshes (as, for example, at the poles of the sphere when using latitude-longitude grids) and the discretized differential operators.

*Correspondence to: Department of Mathematics and Statistics, Memorial University of Newfoundland and Labrador, St. John's, Newfoundland and Labrador A1C 5S7, Canada. E-mail: smaclachlan@mun.ca

Contract/grant sponsor: The work of SM was partially supported by an NSERC discovery grant. The work of IL was partially supported by the NSF; contract/grant number: DMS-1418843

Contract/grant sponsor: This work was partially supported by the NSF; contract/grant number: DMS-1015370, DMS-1216972

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls *[Version: 2010/05/13 v2.00]*

Embedding methods are another approach, exemplified by the closest point method (CPM) of Ruuth and coauthors [18, 19], which inherits attractive numerical properties by appropriately embedding the surface in a surrounding volume and extending the PDE to that volume. Alternatively, one can aim to *discretize the surface* directly, by defining an approximating triangulated surface, and then applying standard finite-element (or finite-volume) approaches to discretize the PDE on the triangulated surface; see the review article [20] and the many references therein.

Within each class of discretization methodology, specific advances have been made, particularly in the development of fast solution algorithms. For example, for the finite-volume discretization of the Laplace-Beltrami operator on the surface of the sphere using latitude-longitude meshes, an optimal multigrid method is known using alternating line relaxation and a transformed restriction operator [21]. Multigrid solvers have also been developed for CPM discretizations [22]. For uniform triangulations of surfaces, [23, 24] demonstrate the effectiveness of simple multigrid methods, using natural finite-element interpolation operators and simple point smoothing. Motivated by the problem of radiation dose planning (as explained in Section 2), this paper aims to extend these results to locally refined triangulations of the sphere for problems with highly localized sources, using the fast adaptive composite-grid (FAC) methodology.

The FAC methodology was first proposed in the 1980's [25, 26] to extend optimal multigrid performance to meshes generated by structured adaptive mesh refinement algorithms. Many such algorithms generate meshes that are *semi-structured*, which can be characterized by having distinct "levels" of refinement with regular structure within each level. While such systems can be solved using algebraic multigrid (AMG) [14], this is naturally inefficient as AMG makes no use of the existing structure in the mesh. FAC, in contrast to AMG, explicitly requires that the composite grid consist of distinct mesh levels, each of which can be considered independently of the others. FAC multigrid cycles consist of independent processing on each level, coupled by intergrid-transfer operators as in standard multigrid, and can be done both multiplicatively (as in [26]) or additively (asynchronously) [27, 28].

The main goal of this paper is to bring together the aspects of mesh refinement and fast multigrid solution for PDEs dominated by the Laplace-Beltrami operator with localized sources on the surface of the sphere. Our motivating application, radiation dose planning, is explained in Section 2, including the Fokker-Planck limit in which the Laplace-Beltrami operator is obtained. In Section 3, we propose a structured mesh refinement algorithm, which is later demonstrated to restore second-order convergence when the source function is singular. Finite-element discretization is used on the resulting non-uniform triangulation of the sphere, discussed in Section 4. In Section 5, the FAC methodology is explained and adapted into a solution algorithm for this discretization. Supporting numerical results are presented in Section 6, including for an example with a non-smooth solution where mesh refinement is expected to be most beneficial. Conclusions and comments on future work are given in Section 7.

2. MOTIVATING APPLICATION: RADIATION DOSE PLANNING

As a motivating application, we consider the radiation dose planning problem, which can be posed as an optimal control problem, $\min_{g|\mathcal{L}(f)=g} J(f)$, where f is the phase-space density of charged particles in tissue. In this model, $J(f)$ is a cost function representing irradiation of cancerous tissue, whose minimum value of 0 is achieved when f is sufficiently large within all identified cancerous tissue in the domain, $\Omega \subset \mathbb{R}^3$, and zero (or sufficiently small) in all healthy tissue. The constraint $\mathcal{L}(f) = g$ represents the solution of the Linear Boltzmann Transport Equation,

$$\frac{1}{c}\frac{\partial f}{\partial t} + \omega \cdot \nabla f + \sigma_t(\vec{x}, E)f - \mathcal{K}_s f = 0,$$

subject to initial conditions $f(\vec{x}, \omega, E, t_0) = f_0(\vec{x}, \omega, E)$ and boundary conditions, $f(\vec{x}, \omega, E, t) = g(\vec{x}, \omega, E, t)$ for $\vec{x} \in \partial\Omega$. To define notation, $f(\vec{x}, \omega, E, t)$ is the phase-space density of particles at location $\vec{x} \in \Omega$, moving in direction $\omega \in S$, with energy E at time t. Here, S is the unit sphere

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

DOI: 10.1002/nla

in $\mathbb{R}^3$. Key parameters in the problem are the particle speed, c, the probability of interaction per unit distance traveled, σ_t, and the scattering kernel, $\mathcal{K}_s$. For charged particles, the mean free path, $\overline{\lambda} = 1/\sigma_t$, is relatively small, but most interactions result in relatively small changes in angle, direction, and energy. The control variable, $g(\vec{x}, \omega, E, t)$, represents the choice of a suitable configuration of beams of charged particles to properly irradiate the cancerous tissue. In real-world applications there are, of course, many practical constraints on the possible values of g imposed by available equipment and medical safety. Efficient solution algorithms for such optimization problems often rely on efficient solution algorithms for the constraining PDEs, and it is this "forward problem" that we concentrate on in this paper.

To first simplify this model, we consider the time-independent case of mono-energetic transport (constant c and E), and assume that all interactions result in scattering, so that

$$\mathcal{K}_s f(\vec{x}, \omega) = \left(\overline{\lambda}\right)^{-1} \int_S \frac{p(\omega \cdot \omega')}{2\pi} f(\vec{x}, \omega') d\omega',$$

where $p(\omega \cdot \omega')/2\pi$ is the probability density for scattering from direction $\omega' \in S$ to direction ω, depending only on the angle between unit vectors ω and ω'. In this setting, the governing Boltzmann Equation simplifies to

$$\omega \cdot \nabla f(\vec{x}, \omega) = \mathcal{K}_s f(\vec{x}, \omega) - \left(\overline{\lambda}\right)^{-1} f(\vec{x}, \omega) = \left(\overline{\lambda}\right)^{-1} \left(\int_S \frac{p(\omega \cdot \omega')}{2\pi} f(\vec{x}, \omega') d\omega' - f(\vec{x}, \omega) \right) \quad (1)$$

and the domain of f is now $\Omega \times S$. The balance in Equation (1) between the advective term on the left and the scattering term on the right depends on the properties of the probability density, $p(\mu)$. Of particular interest is the expected value of $\mu = \omega \cdot \omega'$, defined as $\overline{\mu} = \int_{-1}^{1} \mu p(\mu) d\mu$, which gives the expected value of the cosine of the angle between an incident direction, ω', and a scattered direction, ω. In many models of the scattering of charged particles, the scattering occurs very frequently, giving a small mean free path, $\overline{\lambda}$, but each interaction results in only a small deflection, giving $\overline{\mu} \approx 1$. The Fokker-Planck limit considers the case where $\overline{\lambda} \to 0$ and $\overline{\mu} \to 1$ in a way such that the transport mean-free path, $\overline{\lambda}_{tr} = \frac{\overline{\lambda}}{1-\overline{\mu}} = \frac{2}{T} > 0$, is fixed. In this case, [29] shows that the right-hand side of (1) converges weakly to $\frac{1}{2\overline{\lambda}_{tr}} \Delta_\omega f$, the Laplace-Beltrami operator on the sphere. The resulting equation,

$$\omega \cdot \nabla f(\vec{x}, \omega) = \frac{1}{2\overline{\lambda}_{tr}} \Delta_\omega f, \quad (2)$$

is known as the Fokker-Planck equation.

In recent years, several contributions have been made to the literature on both the discretization and efficient solution of Equation (1), primarily drawing on insight from the limiting case in Equation (2). Discretization in space is typically handled using upwind finite differences or DG-type discretizations, yielding a natural downstream ordering of the resulting linear systems. In angle, several discretizations of the unit sphere have been considered, including the S_n (discrete ordinates) discretization [30], popular in the computational transport literature, but finite-difference and Galerkin finite-element methods have also been used. In [31], an *angular multigrid method* was proposed for the one-dimensional analogue of (1), with semi-coarsening in the angular variable and a "downstream" Gauss-Seidel relaxation used to resolve the spatial variation. This was extended to a two-dimensional problem in [32], for a finite-difference discretization in "flatland", where the domain of f is $[0, 1]^2 \times [0, 2\pi]$; a domain-decomposition preconditioner for this problem was also considered in [33]. Similar work has also been done for three-dimensional scattering, in the case of variable E, [34–37], considering the S_n discretization scheme. In [38, 39], a uniform-grid finite-element discretization of S is used, coupled with a finite-difference or discontinuous-Galerkin discretization of the angular terms. A general conclusion from these works is that efficient and accurate simulation of the Boltzmann Equation for charged particles is possible, provided that a fast solver is available for an accurate discretization of the angular part of the operator.

Considering the motivating application of radiation dose planning, a natural inefficiency arises in the use of uniform discretizations of the sphere, since the typical boundary conditions impose

Copyright © 2016 John Wiley & Sons, Ltd.

Prepared using nlaauth.cls
DOI: 10.1002/nla

beams of charged particles that are highly localized in direction. Thus, in this paper, we consider the question of mesh refinement on the sphere, restricted to the Laplace-Beltrami operator (or its reaction-diffusion analogue). While extensions to the full Fokker-Planck and linear Boltzmann equation are necessary for the resulting algorithm to be applied in the context of the radiation dose planning problem, we leave these for future work. In what follows, we propose a structured mesh refinement strategy that addresses (reaction-)diffusion on the sphere with localized sources, a natural finite-element discretization that complements this approach, and we extend the FAC multigrid approach to these meshes and this discretization. The proposed discretization and refinement scheme are similar to those recently proposed for neutral-particle transport in [40].

3. STRUCTURED REFINEMENT OF THE SPHERE

Let S be the unit sphere in $\mathbb{R}^3$. We henceforth consider the solution of PDEs on the sphere of the form

$$-\Delta_\omega u(\omega) + \alpha u(\omega) = g(\omega) \quad \forall \omega \in S, \tag{3}$$

where Δ_ω represents the Laplace-Beltrami operator on S, and $\alpha \geq 0$ is a constant. We consider specifically the case where $g(\omega)$ is a function either with localized support or where it is very small except over a small area on S. Our goal is to define a non-uniform triangulation of S such that the variation of g (and, consequently, u) is efficiently resolved by the triangulation. When discretizing Equation (3) using a finite-element approach (discussed below in Section 4), this is essentially equivalent to saying that the triangulation yields a good discrete approximation for the solution, $u(\omega)$.

In this paper, we consider the case where the local structure in $g(\omega)$ is known in advance and, thus, structured refinement approaches are more natural than adaptive mesh refinement. We start with a "base" triangulation of the sphere, $\mathcal{T}_0$. Natural choices are to use an octahedron or icosahedron as the "zeroth" mesh, with all points of the polyhedron lying on S, although refinements of such meshes or alternate base triangulations could also be used. Given an existing mesh, our general (local-then-global) refinement algorithm is considered to have 3 parts:

1. Mark triangles for local refinement
2. Refine marked triangles
3. Refine all triangles (including those created in Step 2).

Whenever a triangle is refined in this algorithm, we consider uniform refinement, bisecting each edge of the triangle to create new nodes, interconnecting these nodes to create 4 new sub-triangles, and then projecting these nodes onto the surface of the sphere. If multiple levels of refinement are to be done simultaneously, either in Step 2 or 3 above, they are done by sequentially bisecting the edges the appropriate number of times.

Considering the case where $g(\omega)$ is a point source (or similar) located at the North pole of the sphere, one instance of this strategy will be to consider the refinement of an octahedron with one vertex located at the North pole. In the algorithm above, we will then mark all triangles adjacent to the North pole for local refinement, in addition to a global refinement. Figure 1 shows the resulting triangulations when each refinement step is a single level, dividing each refined triangle into four subtriangles. Multiple simultaneous levels of refinement are also possible within this algorithm, as is reversing the order of the global and local refinement. Both of these allow for finer-grained control of the relative density of points in the refined region and those away from the refinement. Figure 2 shows the resulting meshes from the "reverse" (global-then-local) refinement strategy, where a global refinement step is first applied, then the resulting elements adjacent to the North pole are refined a second time. In comparison to the meshes in Figure 1, we see a more focused refinement pattern around the North pole, while the global base mesh extends further into the Northern hemisphere.

The leftmost image in Figure 2 highlights an important property of the resulting triangulations, which is that they are naturally non-conforming, in the sense that the local refinement step introduces

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

DOI: 10.1002/nla

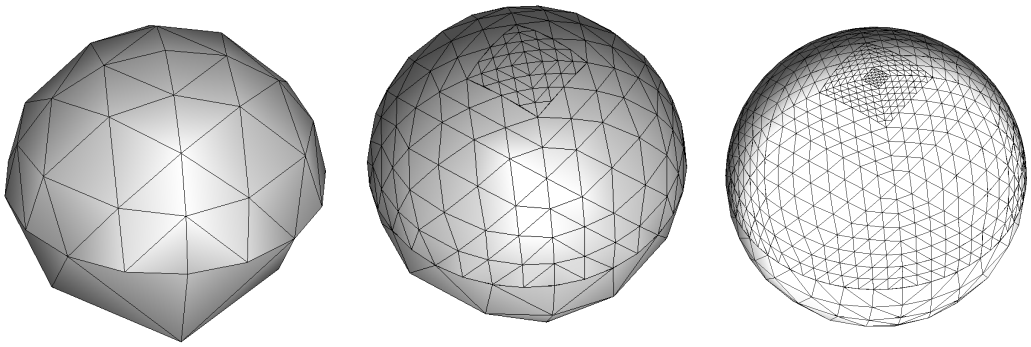

Figure 1. Three meshes obtained by applying the local-then-global refinement strategy where marked triangles are those adjacent to the North pole.

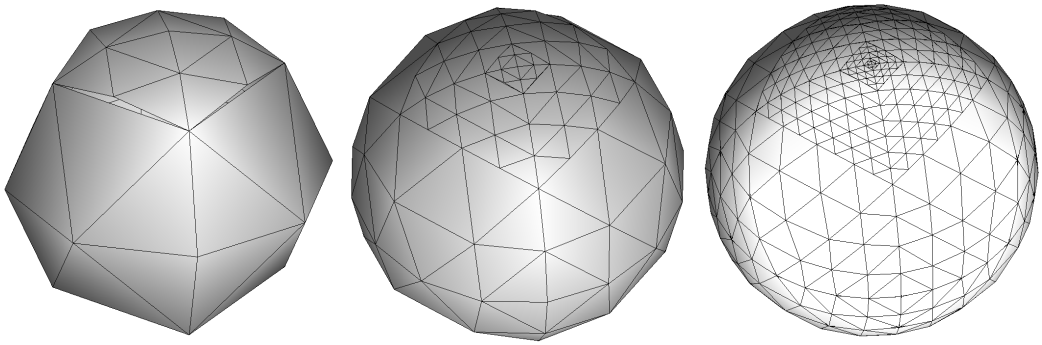

Figure 2. Three meshes obtained by applying the global-then-local refinement strategy where marked triangles are those adjacent to the North pole after a global refinement is performed.

"hanging nodes." Due to the projection of such nodes onto the surface of the sphere, these nodes "disconnect" the triangulation; while an additional face could be inserted to fill this area, we choose not to and, instead, simply enforce continuity in the finite-element approximation defined below. This avoids the introduction of elements with poor shape regularity that would otherwise negatively impact the discrete problem to be solved.

In what follows, we view such refined meshes as a hierarchy of patches centred at the North pole. If ℓ steps of the refinement algorithm are performed, we can consider a quasi-uniform global mesh, $\mathcal{T}_{\ell,0}$, which is formed by taking the base mesh, $\mathcal{T}_0$, (an octahedron for the refined meshes shown in Figures 1 and 2) and refining it ℓ times. We then define a sequence of ℓ "patches", with $\mathcal{T}_{\ell,1}$ denoting the first refinement patch around the North pole, starting from the elements created by the first local refinement step that have also been uniformly refined ℓ times. Thus, if h_0 is a representative mesh-width for the global mesh, $\mathcal{T}_{\ell,0}$, then the representative meshwidth for $\mathcal{T}_{\ell,1}$ is $h_1 \approx h_0/2$, indicating that the triangles in $\mathcal{T}_{\ell,1}$ are at one level of refinement finer than those in $\mathcal{T}_{\ell,0}$. Note that, due to the projection in both the local and global refinement steps, there will be variations in edge lengths and triangle areas that would not be present for refinement in the plane. Continuing, we define $\mathcal{T}_{\ell,k}$ to be the elements created in the k^{th} local refinement step, uniformly refined ℓ times. Following the above argument, the representative meshwidth for $\mathcal{T}_{\ell,k}$ is $h_k \approx h_{k-1}/2$, and the elements in $\mathcal{T}_{\ell,k}$ can be seen to be uniformly refined k times relative to the global mesh, $\mathcal{T}_{\ell,0}$.

It is important for what follows to emphasize that each patch, $\mathcal{T}_{\ell,k}$, contains all elements that are covered by $\mathcal{T}_{\ell,k+1}$ (and all finer patches), but at the same level of resolution as the non-overlapped triangles in $\mathcal{T}_{\ell,k}$. In order to properly define the triangles active in composite meshes of varying levels of refinement, we also define the "overlap" between $\mathcal{T}_{\ell,k}$ and $\mathcal{T}_{\ell,k+1}$ as those elements in $\mathcal{T}_{\ell,k}$

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

DOI: 10.1002/nla

that are covered by $\mathcal{T}_{\ell,k+1}$. Formally, one way to express this is as

$$\mathcal{T}'_{\ell,k} = \{T \in \mathcal{T}_{\ell,k} \mid \text{ all nodes of } T \text{ are also nodes of triangles in } \mathcal{T}_{\ell,k+1}\}.$$

For a given number of levels of refinement, ℓ, the composite mesh can then be defined as

$$\mathcal{S}_\ell = \left(\bigcup_{k=0}^{\ell-1} \left(\mathcal{T}_{\ell,k} \setminus \mathcal{T}'_{\ell,k} \right) \right) \cup \mathcal{T}_{\ell,\ell}. \tag{4}$$

While the notation for $\mathcal{S}_\ell$ would be slightly more transparent with an alternate definition for $\mathcal{T}_{\ell,k} \setminus \mathcal{T}'_{\ell,k}$, the fast adaptive composite-grid multigrid method discussed below is more natural using this notation. An illustration of this definition of the composite mesh is given in Figure 3; while this figure is restricted to the plane, it matches the connectivity of the composite mesh on the Northern hemisphere of the sphere using the global-then-local refinement pattern as shown in Figure 2. Note that in the particular case of refinement by a single level of refinement in each local/global step, we have the natural relationship that $\mathcal{T}'_{\ell,k} = \mathcal{T}_{\ell-1,k+1}$, when the latter is well-defined (i.e., when $\ell - 1 \geq k + 1$), but that this relationship does not hold for general refinement patterns. As such, we cannot directly simplify the notation in (4), unless we make additional assumptions.

4. COMPOSITE-GRID FINITE-ELEMENT DISCRETIZATION

To discretize Equation (3), we consider a piecewise linear finite-element discretization on the "composite grid" formed by the refinement process discussed above. Discretization then follows by the surface finite-element method (see, e.g. [20]), which we adapt to suit the composite-grid setting.

On any given (flat) triangular face, T, with normal direction, $\vec{n}_T$, we define the *tangential gradient* on T, as $\nabla_T u = \nabla \overline{u} - (\nabla \overline{u} \cdot \vec{n}_T)\, \vec{n}_T$, where $\overline{u}$ is a smooth extension of $u : T \to \mathbb{R}$ into a neighbourhood of T in $\mathbb{R}^3$. We note that if T happens to lie parallel to the $xy-$plane, for example, this reduces simply to the two-dimensional gradient, extended by zero in the z-direction. This allows us to define the weak form of the Laplace-Beltrami operator in (3) on a triangulation $\mathcal{S}_\ell$ as finding $u \in H^1(\mathcal{S}_\ell)$ such that

$$a_\ell(u, v) = \sum_{T \in \mathcal{S}_\ell} \int_T (\nabla_T u \cdot \nabla_T v + \alpha u v) = \sum_{T \in \mathcal{S}_\ell} \int_T g_\ell v = \langle g_\ell, v \rangle_\ell \quad \forall v \in H^1(\mathcal{S}_\ell), \tag{5}$$

where $H^1(\mathcal{S}_\ell)$ is defined in the natural way. In what follows, we also make use of the space $C^0(\mathcal{S}_\ell)$ for functions mapping from $\mathcal{S}_\ell$ into $\mathbb{R}$. This space is defined in the natural way both within any triangle, $T \in \mathcal{S}_\ell$, and across common edges between triangles, but extended to the non-conforming situation at boundaries between two levels of refinement in $\mathcal{S}_\ell$. In this case, continuity is still enforced on the non-conforming edges in $\mathcal{S}_\ell$, by defining arc-length parameterizations $e_k : [0, 1] \to \mathcal{S}_\ell$ and $e_{k+1} : [0, 1] \to \mathcal{S}_\ell$ of the single (coarse) edge and multiple (fine) edges, respectively, with $e_k(0) = e_{k+1}(0)$ and $e_k(1) = e_{k+1}(1)$ marking the two common nodes, and requiring that $f(e_k(t)) = f(e_{k+1}(t))$ for $0 \leq t \leq 1$ for all functions $f \in C^0(\mathcal{S}_\ell)$. There are, of course, many possible parameterizations of the two sets of edges; we assume both e_k and e_{k+1} are arc-length parameterizations in order to impose the natural continuity condition that the function value at the midpoint of a coarse edge matches that at the middle node in a refinement of that edge.

Our surface finite-element discretization is, then, defined on a triangulation, $\mathcal{S}_\ell$, by taking

$$\mathcal{V}_\ell = \left\{ u_\ell \in C^0(\mathcal{S}_\ell) \mid \forall T \in \mathcal{S}_\ell, u_\ell \text{ is linear on } T \right\}, \tag{6}$$

giving a standard piecewise linear approximation on each triangle in $\mathcal{S}_\ell$. The discrete weak form over $\mathcal{S}_\ell$ is then defined by restricting (5) to $\mathcal{V}_\ell$, i.e., finding $u_\ell \in \mathcal{V}_\ell$ such that $a_\ell(u_\ell, v_\ell) = \langle g_\ell, v_\ell \rangle_\ell$ for all $v_\ell \in \mathcal{V}_\ell$. Note that in the non-conforming case, the requirement that $u_\ell \in C^0(\mathcal{S}_\ell)$ "slaves"

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

DOI: 10.1002/nla

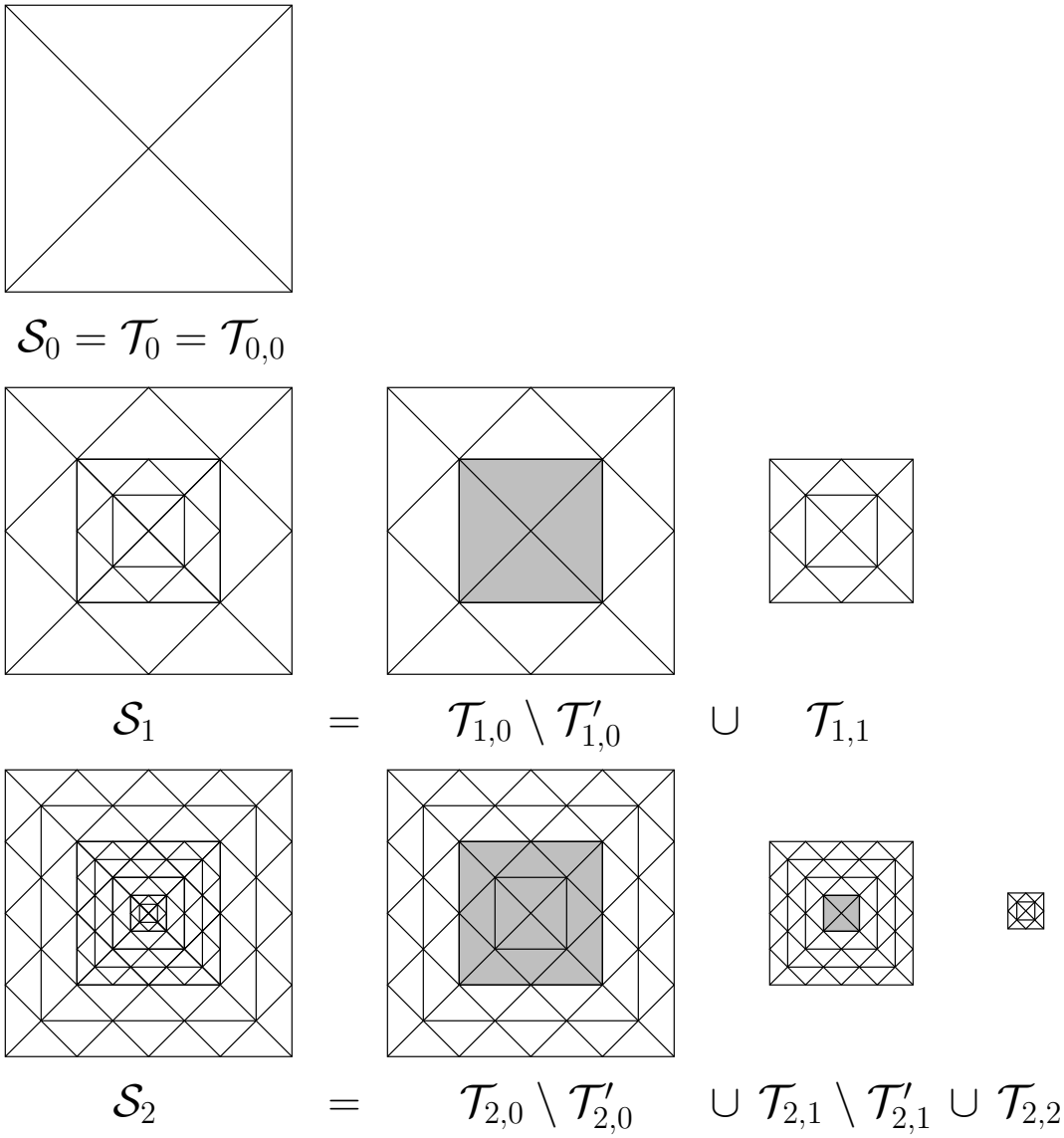

Figure 3. Decomposition of composite meshes, $\mathcal{S}_0$, $\mathcal{S}_1$, and $\mathcal{S}_2$ into patches for global-then-local refinement pattern on a square in the plane.

any hanging nodes in the triangulation to take the interpolated values at corresponding points on the unrefined side of the interface. This weak form presumes, however, that the source term g_ℓ is defined naturally on $\mathcal{S}_\ell$, which is not the case since $g : S \to \mathbb{R}$ in the Laplace-Beltrami Equation in (3). In order to achieve best-possible order of approximation, we must ensure that g_ℓ approximates g in a suitable way (see, for example, [20,41]). In what follows, we extend the source function g to its approximation $g_\ell : \mathcal{S}_\ell \to \mathbb{R}$ by projecting values from S onto $\mathcal{S}_\ell$. In particular, for a point, $x \in \mathcal{S}_\ell$, we draw the straight-line path from the origin through x to a point $\omega \in S$, and fix $g_\ell(x) = g(\omega)$.

Note also that in the case where $\alpha = 0$, we must impose an additional constraint on both u_ℓ and g_ℓ in order to ensure unique solvability. We extend the natural conditions from the case of planar domains, requiring that $\langle g_\ell, 1 \rangle_\ell = 0$, yielding a consistent linear system for the finite-element approximation. To ensure uniqueness of u_ℓ, we can either fix the value of u_ℓ at a point in $\mathcal{S}_\ell$ or impose a similar orthogonality condition. In the experiments to follow (only when $\alpha = 0$), we choose the former and fix the value of u_ℓ at the South pole to be zero.

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

DOI: 10.1002/nla

In what follows, we make use of two decompositions of $\mathcal{V}_\ell$. A natural overlapping decomposition is to write $\mathcal{V}_\ell = \oplus_{k=0}^{\ell} \mathcal{V}_{\ell,k}$, where

$$\mathcal{V}_{\ell,k} = \left\{ u_k \in C^0(\mathcal{T}_{\ell,k}) \mid u_k \text{ is piecewise linear on } \mathcal{T}_{\ell,k} \right\}.$$

To represent $\mathcal{V}_\ell$ without overlap, we introduce the subspaces

$$\mathcal{W}_{\ell,k} = \left\{ u_k \in C^0(\mathcal{T}'_{\ell,k}) \mid u_k \text{ is piecewise linear on } \mathcal{T}'_{\ell,k} \right\},$$

noting that $\mathcal{W}_{\ell,k} \subset \mathcal{V}_{\ell,k}$ (since $\mathcal{T}'_{\ell,k} \subset \mathcal{T}_{\ell,k}$) and that $\mathcal{W}_{\ell,k} \subset \mathcal{V}_{\ell,k+1}$ (since $\mathcal{T}_{\ell,k+1}$ covers $\mathcal{T}'_{\ell,k}$). An equivalent, but non-overlapping, decomposition of $\mathcal{V}_\ell$ is then given by

$$\mathcal{V}_\ell = \mathcal{V}_{\ell,\ell} \oplus_{k=0}^{\ell-1} \left(\mathcal{V}_{\ell,k} \setminus \mathcal{W}_{\ell,k} \right),$$

where boundaries between levels of refinement are implicitly constrained to match the coarser-level representation due to the requirement that $\mathcal{V}_\ell \subset C^0(\mathcal{S}_\ell)$. In what follows, we will make use of two natural finite-element interpolation operators, $Q_k : \mathcal{W}_{\ell,k} \to \mathcal{V}_{\ell,k}$ and $P_k : \mathcal{W}_{\ell,k} \to \mathcal{V}_{\ell,k+1}$. While we can consider these both as operators acting on the function spaces and as matrices acting on vectors of coefficients for basis expansions, in what follows we will exclusively consider the matrix representation of these operators.

5. FAST ADAPTIVE COMPOSITE GRID METHOD

To solve the discretized linear system on the composite mesh, $\mathcal{S}_\ell$,

$$\text{Find } u_\ell \in \mathcal{V}_\ell \text{ such that } a_\ell(u_\ell, v_\ell) = \langle g_\ell, v_\ell \rangle_\ell \quad \forall v_\ell \in \mathcal{V}_\ell, \tag{7}$$

we use the Fast Adaptive Composite Grid (FAC) methodology [25–28]. This is based on the overlapping composite-grid decomposition of $\mathcal{S}_\ell$, where each "patch," $\mathcal{T}_{\ell,k}$, serves as a level in the multigrid hierarchy.

There are many variations on the FAC methodology. The original FAC algorithm of [25, 26] is based on a decomposition of the mesh into overlapping levels, with the discrete weak form solved sequentially on each level (from coarsest to finest). To break this sequentiality, asynchronous FAC (AFAC) methods were proposed and studied in [27,28,42], where approximations for each level are computed in parallel and differences of these approximations are used to correct the composite-grid approximation. AFACx replaces solves on each level within AFAC with the application of simple relaxation approaches. Here, we consider an "FAC V-cycle", where each level in the composite grid is treated by a simple relaxation scheme (lexicographical Gauss-Seidel), but the processing is sequential, as in standard multigrid. To enable the V-cycle to be used as a preconditioner for the conjugate gradient algorithm, we use symmetric relaxation ordering and a V(1,1) cycling scheme.

To fully describe the FAC V-cycle, we define two hierarchies of discrete operators corresponding to the constituent pieces of the discretized weak form in (7). Restricting to $\mathcal{T}_{\ell,k}$, we define the system matrix A_k and right-hand side, $\mathbf{g}_k$ by the weak form

$$\sum_{T \in \mathcal{T}_{\ell,k}} \int_T (\nabla_T u \cdot \nabla_T v + \alpha u v) = \sum_{T \in \mathcal{T}_{\ell,k}} \int_T g v,$$

for $u, v \in \mathcal{V}_{\ell,k}$, where, again, we project $g : S \to \mathbb{R}$ onto each triangle, T, in order to define the integrals on the right-hand side. Similarly, restricting to $\mathcal{T}'_{\ell,k}$, we define the system matrix $\tilde{A}_k$ and right-hand side, $\tilde{\mathbf{g}}_k$ by the weak form

$$\sum_{T \in \mathcal{T}'_{\ell,k}} \int_T (\nabla_T u \cdot \nabla_T v + \alpha u v) = \sum_{T \in \mathcal{T}'_{\ell,k}} \int_T g v,$$

for $u, v \in \mathcal{W}_{\ell,k}$, with the corresponding projection on the right-hand side.

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

DOI: 10.1002/nla

While the multigrid interpolation operators were defined above, some subtlety exists in the FAC restriction step in order to form the corresponding residual on each coarser level after relaxation on the next finer level of the composite-grid hierarchy. To simplify this, consider restriction from level k in the hierarchy. After relaxation on level k, the $\mathcal{V}_{\ell,k}$ residual is naturally defined as $\mathbf{r}_k = \mathbf{g}_k - A_k\mathbf{u}_k$, where $\mathbf{u}_k$ is the discrete approximation after relaxation. Two residuals must be considered over $\mathcal{V}_{\ell,k-1}$, that over $\mathcal{W}_{\ell,k-1}$, $\tilde{\mathbf{r}}_{k-1} = \tilde{\mathbf{g}}_{k-1} - \tilde{A}_{k-1}\tilde{\mathbf{u}}_{k-1}$, and that over $\mathcal{V}_{\ell,k-1}$ itself, $\mathbf{r}_{k-1} = \mathbf{g}_{k-1} - A_{k-1}\mathbf{u}_{k-1}$, where $\tilde{\mathbf{u}}_{k-1}$ and $\mathbf{u}_{k-1}$ are the existing approximations on that level. Note that $\mathbf{r}_{k-1}$ and $\tilde{\mathbf{r}}_{k-1}$ should agree over the interior of the patch $\mathcal{T}'_{\ell,k-1}$, but they will differ along the boundary of $\mathcal{T}'_{\ell,k-1}$ (and only $\mathbf{r}_{k-1}$ will be defined on $\mathcal{T}_{\ell,k-1} \setminus \mathcal{T}'_{\ell,k-1}$). The restriction step can, thus, be expressed as

$$\mathbf{r}_{k-1} \leftarrow \mathbf{r}_{k-1} + Q_{k-1}\left(P_k^T\mathbf{r}_k - \tilde{\mathbf{r}}_{k-1}\right). \tag{8}$$

In this expression, the term $P_k^T\mathbf{r}_k$ is the natural restriction of the (updated) grid k residual to $\mathcal{T}'_{\ell,k-1}$, while applying Q_{k-1} to this vector simply extends it (by zero) to all of $\mathcal{T}_{\ell,k-1}$. The difference $\mathbf{r}_{k-1} - Q_{k-1}\tilde{\mathbf{r}}_{k-1}$ is, in contrast, zero within the patch, and equal to the existing residual on $\mathcal{T}_{\ell,k-1} \setminus \mathcal{T}'_{\ell,k-1}$. Thus, the restriction step in (8) appropriately accounts for the combined purpose of mesh $\mathcal{T}_{\ell,k-1}$, to both provide a correction to the refined mesh $\mathcal{T}_{\ell,k}$ and an approximation to the solution on the non-overlapped mesh, $\mathcal{T}_{\ell,k-1} \setminus \mathcal{T}'_{\ell,k-1}$. Proper treatment at the boundary of the patch is provided by the natural finite-element interpolation operators, Q_{k-1} and P_k.

Pseudo-code for the FAC V-cycle is given below as Algorithm 1. On the "pre-relaxation" side of the cycle, this essentially matches a standard multigrid V-cycle, with a relaxation step and a fine-grid residual calculation; the only difference is in the restriction step which takes the form in (8), above, rather than a standard restriction operation. On the global mesh, $\mathcal{T}_{\ell,0}$, a standard geometric multigrid V(1,1)-cycle is used as the solver, progressing through coarsened meshes, $\mathcal{T}_{\ell-1,0}, \mathcal{T}_{\ell-2,0}, \ldots$ until the original base mesh is reached, where either an exact solve is performed or a few sweeps of relaxation are used to approximate the coarsest-grid solution. The "post-relaxation" side of the cycle is also standard, both for the global mesh and the composite grid. In order to ensure symmetry of the resulting cycle, we use a forward-backward ordering of relaxation, with forward lexicographical sweeps used for pre-relaxation and reverse lexicographical sweeps used for post-relaxation. When used as a preconditioner for the conjugate gradient algorithm, the right-hand side vectors, $\mathbf{g}_k$ and $\tilde{\mathbf{g}}_k$, on all meshes are replaced with the current CG residual on each mesh, and zero initial guesses are used for $\mathbf{u}_k$ and $\tilde{\mathbf{u}}_k$ on all meshes. We use a single FAC V(1,1)-cycle as the preconditioner when doing so.

Algorithm 1 FAC V-cycle Pseudo-code

```
for k = ℓ, ℓ − 1, . . . 1 do
    Relax on interior of T_{ℓ,k}
    Compute residual on T_{ℓ,k}
    Update residual for T_{ℓ,k−1}
end for
Regular V-cycle on T_{ℓ,0}
for k = 1, 2, . . . , ℓ do
    Interpolate and add correction from T_{ℓ,k−1}
    Relax on interior of T_{ℓ,k}
end for
```

To illustrate Algorithm 1, we consider the cycle on $\mathcal{S}_2$, as shown in the bottom line of Figure 3. The algorithm begins with a sweep of relaxation on $\mathcal{T}_{2,2}$, after which a residual is calculated. The restriction step to $\mathcal{T}_{2,1}$, as in Equation (8), can be thought of in two pieces. Outside of the patch $\mathcal{T}'_{2,1}$, we seek to directly approximate the solution and, so, define the current residual there by $\mathbf{r}_1 - Q_1\tilde{\mathbf{r}}_1$. Within the patch, we aim to compute a correction to the current approximation and, thus, augment this by the restricted residual over the patch, $Q_1P_2^T\mathbf{r}_2$. A similar sequence is then followed on $\mathcal{T}_{2,1}$, with a sweep of relaxation, a residual calculation, and restriction via Equation (8) to the global mesh $\mathcal{T}_{2,0}$. After a V-cycle on the global mesh, a correction over the patch $\mathcal{T}'_{2,0}$ is interpolated to $\mathcal{T}_{2,1}$. After

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

DOI: 10.1002/nla

relaxation on $\mathcal{T}_{2,1}$, a correction over the patch $\mathcal{T}'_{2,1}$ is interpolated to $\mathcal{T}_{2,2}$, followed by relaxation on this finest level.

6. NUMERICAL EXPERIMENTS

For the numerical results presented here, we use a C++ implementation of the FAC algorithm, using the modular finite-element library MFEM [43] for managing the discretization, mesh, and interpolation operators. All tests were run in serial on a single node of the Tufts High-Performance Computing Research Cluster, using a 2.2 GHz Intel Xeon CPU and 128 GB available RAM, with code compiled with gcc using full optimizations. For comparison, we present numerical results using the algebraic multigrid package BoomerAMG [44]. In both cases, we use the preconditioned conjugate gradient implementation in hypre [45] to accelerate convergence.

6.1. Smooth source term

Our first numerical experiments consider the reaction-diffusion analogue on the sphere,

$$-\frac{\tau}{2}\Delta_\omega u(\omega) + u(\omega) = g(\omega), \tag{9}$$

where $g(\omega)$ is calculated so that the solution is given by

$$u(\theta,\phi) = \frac{1}{2}\sum_{i=0}^{\infty}(2i+1)e^{-i(i+1)\tau/2}P_i\left(\cos\theta\right),$$

where $\omega \in S$ is written as $\omega = (\theta,\phi)$ for polar angle, θ, and azimuthal angle, ϕ, $P_i(x)$ is the i^{th} Legendre polynomial on $[-1,1]$, and $\tau = 0.001$. This solution arises as the fundamental solution of the heat equation on the sphere, with initial data as a (scaled) delta function at the North pole ($\theta = 0$), after time τ has elapsed. Thus, Equation (9) models the behaviour of the first time-step in an implicit-in-time integration of the heat equation with time-step τ.

As in the planar case, the fundamental solution to the heat equation is in $C^\infty(S)$; thus, we expect an optimal order of approximation already on a uniform grid. Since we use a piecewise linear approximation, we expect second-order convergence in the L_2 error. However, since the solution is highly peaked around $\theta = 0$, we expect to see some benefit, in terms of the accuracy per degree of freedom, in the adaptive meshes considered above. Figure 4 plots the relative L_2 error for a uniform mesh and two refined meshes, generated by the local-then-global ("standard") and global-then-local ("reverse") refinement patterns discussed above and pictured in Figures 1 and 2, respectively. All three methods clearly offer convergence with errors proportional to the inverse of the number of degrees of freedom. For the uniform mesh, this is equivalent to second-order convergence since the surface of the sphere is two-dimensional. As expected, the locally refined meshes offer better accuracy per degree-of-freedom by concentrating elements towards the poles, where the greatest variation in the solution occurs. At 12.5 million degrees of freedom using the local-then-global refinement pattern, the resulting solution has a relative error in this measure of 1.9×10^{-4}, compared with that of 3.3×10^{-3} when using 16.7 million degrees of freedom on a uniform mesh, showing a clear advantage for the refined meshes.

The advantages in accuracy of the locally refined meshes are, of course, only of benefit if there is no significant additional cost associated with the solution of the resulting linear systems. As a baseline for comparison, Table I presents results for geometric multigrid with rediscretization used for coarse-grid operators on the uniform mesh discretization. In this table, and those that follow, we report grid sizes measured by the number of degrees of freedom in the resulting linear system (i.e., the number of nodes in the mesh), the relative error in the discrete solution, measured in the L_2 norm (as displayed in Figure 4), the setup time for the geometric multigrid method, and both the solve time and number of iterations for the multigrid-preconditioned conjugate gradient algorithm. The setup time includes the total time required to sequentially refine the octahedral base mesh to each

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

DOI: 10.1002/nla

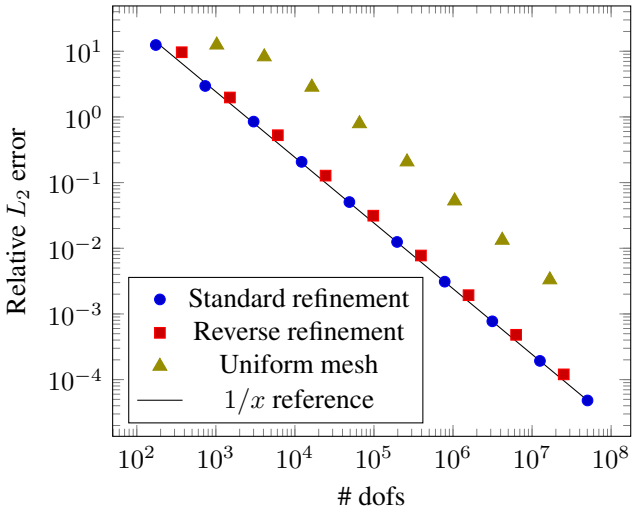

Figure 4. Convergence plot, smooth solution for uniform, local-then-global (standard), and global-then-local (reverse) refinement patterns.

# dofs	relative L_2 error	setup time	solve time	# of PCG iter.
16386	2.84e+00	0.12	0.014	7
65538	7.89e-01	0.49	0.063	8
262146	2.07e-01	2.06	0.27	8
1048578	5.25e-02	8.68	1.26	9
4194306	1.32e-02	35.26	5.46	9
16777218	3.31e-03	143.2	23.46	9

Table I. Finite-element accuracy and solver performance for example with smooth solution on uniform meshes.

level of refinement in the multigrid hierarchy, discretize the equation on each of these meshes, and determine the grid-transfer operators between levels of the hierarchy. Two noticeable conclusions are possible from this data. First, that both setup and solve times scale roughly linearly with the number of degrees of freedom, as is expected. Secondly, the total time-to-solution is dominated by the setup time, which is consistently a factor of six or more times more costly than the PCG solve time. This illustrates a fundamental difficulty of working with triangulations of surfaces, that mesh refinement and discretization require substantially more effort than they do on planar regions, even when using uniform meshes.

Results for the FAC algorithm applied to the locally refined meshes are shown in Tables II (for the local-then-global refinement pattern) and III (for the global-then-local refinement pattern). Most notable from these results is that there is no particular "penalty" to pay in terms of FAC setup and solve times when using these refined meshes. For both refinement patterns, setup time scales roughly linearly with the number of degrees of freedom, as do solve times; again, there is clearly a higher cost to the setup phase of the algorithm, by the same factor of about six or more, except on the finest mesh of the local-then-global refinement pattern, with approximately 50 million degrees of freedom, where the factor is only five. Note also that the local-then-global refinement pattern offers slightly better accuracy per degree of freedom than the global-then-local pattern.

A further comparison for the efficiency of the FAC algorithm is against the performance of algebraic multigrid (AMG), for which we use the BoomerAMG code from the Hypre package [44, 45] with standard parameters. Tables IV and V detail solver timings and iteration counts for this comparison. For clarity, we break out the mesh-refinement and finite-element discretization times in the column labelled "Matrix/RHS setup" from the actual setup time consumed by the AMG

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

DOI: 10.1002/nla

# dofs	relative L_2 error	setup time	solve time	# of PCG iter.
48894	5.05e-02	0.39	0.053	9
196094	1.25e-02	1.61	0.23	9
785406	3.10e-03	6.77	0.97	9
3143678	7.72e-04	28.32	4.12	9
12578814	1.93e-04	114.6	17.61	9
50323454	4.81e-05	477	95.09	9

Table II. Finite-element accuracy and FAC solver performance for example with smooth solution on meshes following the local-then-global refinement pattern.

# dofs	relative L_2 error	setup time	solve time	# of PCG iter.
24450	1.28e-01	0.19	0.026	9
98050	3.13e-02	0.77	0.11	9
392706	7.76e-03	3.27	0.47	9
1571842	1.93e-03	13.80	1.95	9
6289410	4.81e-04	55.65	8.28	9
25161730	1.20e-04	228.4	35.64	9

Table III. Finite-element accuracy and FAC solver performance for example with smooth solution on meshes following the global-then-local refinement pattern.

# dofs	Matrix/RHS setup	AMG setup	Solve	# of of PCG iter.
48894	0.70	0.08	0.11	11
196094	2.03	0.34	0.51	12
785406	7.37	1.34	2.32	12
3143678	28.4	5.33	9.75	12
12578814	113.7	21.04	39.8	12

Table IV. AMG solver performance for example with smooth solution on meshes following the local-then-global refinement pattern.

algorithm, computing the AMG coarse meshes and interpolation operators, as well as the Galerkin coarse-grid operators. Comparing the timings for the local-then-global refinement pattern in Table IV to those for FAC in Table II, we see that the cost for the construction of the composite mesh and finite-element assembly of the matrix and right-hand side is roughly the same as the setup cost for the full FAC algorithm. Additional to this is the cost of the AMG setup and, then, a higher solve cost for AMG, due both to a small increase in the number of iterations and a larger increase in the cost-per-iteration. We note also in this case that the lower memory overhead of the FAC algorithm allows us to solve a problem with 50 million degrees of freedom using FAC, which we could not do within the memory available to us with AMG. Similar behaviour is also seen for the global-then-local refinement pattern, with results for AMG in Table V for comparison to those in Table III. Here, there is a consistent but anomalous penalty of about 20% in CPU time for the construction of the composite mesh and finite-element assembly within the AMG code compared to that in the FAC code. This is due to an increase in time in the composite mesh generation function within our finite-element package, MFEM, for this particular refinement pattern in the code that calls AMG. While it should be possible to eliminate this difference, it is small enough that we have not done so in the results reported here. There is also a larger increase in solve time (that cannot be readily eliminated) for the AMG approach compared to the local-then-global refinement pattern, about a factor of 2.5 compared to a factor of 2, coming from increased complexity in the AMG hierarchy for this refinement pattern.

Overall, two main themes emerge from these results. First, there is no substantial penalty in terms of computational cost per degree of freedom for solving the problems on refined meshes using FAC

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

DOI: 10.1002/nla

# dofs	Matrix/RHS setup	AMG setup	Solve	# of of PCG iter.
24450	0.51	0.04	0.05	11
98050	1.27	0.15	0.25	12
392706	4.43	0.67	1.13	12
1571842	16.81	2.75	4.83	12
6289410	67.01	10.8	19.98	12
25161730	267.5	45.39	90.67	12

Table V. AMG solver performance for example with smooth solution on meshes following the global-then-local refinement pattern.

compared to geometric multigrid on uniform meshes. This, then, validates the use of refined meshes for problems with highly localized solutions, such as the example considered here. Furthermore, there is a clear benefit to the use of FAC in this setting over the use of a black-box approach, such as algebraic multigrid. In particular, the practical costs of the full setup of the FAC algorithm are naturally included in those of assembling the composite-mesh finite-element discretization. This coupled with the greatly improved efficiency of the resulting FAC solve phase, in comparison to either the combined AMG setup and solve phases or just the AMG solve phase, shows the true algorithmic potential of using a semi-structured approach with measured speedup factors of two to four times and notably lower memory requirements.

6.2. Impulse source term

In contrast to the example with a smooth solution considered above, we now consider the problem

$$\begin{aligned} -\Delta_\omega u &= \sum_{i=0}^{\infty} \frac{2i+1}{4\pi} P_i(\cos\theta) - \frac{1}{4\pi}, \\ u(\pi,\phi) &= 0 \quad \forall\phi, \end{aligned}$$

which has exact solution

$$u = -\frac{1}{2\pi}\log(\sin(\frac{\theta}{2})).$$

We note that the series term on the right-hand side represents the Dirac delta function at the North pole, and should be interpreted in the distributional sense, rather than as a convergent sum. In the distributional sense, its integral over S is one and, so, the constant shift on the right-hand side is necessary so that the total source term then integrates to zero for consistency. As noted above, the condition imposed at the South pole, $u(\pi,\phi) = 0$, guarantees uniqueness; at the discrete level, this is imposed just as a typical Dirichlet boundary condition would be, albeit at a single point. Figure 5 displays the relative error measured in the L_2 norm between the finite-element solutions and the continuum solution, calculated by omitting those elements adjacent to the North pole to avoid numerical issues evaluating the solutions there. As expected, we now see the clear benefit in rate of convergence given by the adaptive meshes in contrast to the first-order convergence of the uniform mesh (since the mesh-size scales as the inverse square root of the number of degrees of freedom in the uniform-mesh discretization).

Tables VI and VII provide timing statistics for FAC and AMG applied to this problem on the refined mesh using just the local-then-global refinement pattern for brevity. Aside from a slight increase in the number of iterations for each (from 9 to 12 for FAC, and 12 to 15 for AMG), the details strongly mirror those for the smooth solution presented above. Notably, the setup times between the two approaches are roughly equal, with a small added amount for FAC, noticeable only on the finest meshes (corresponding to the time needed to generate the FAC interpolation operators on top of the composite-mesh finite-element assembly). The solve times for FAC are again about a factor of one-half of those for AMG, or about one-third if the AMG setup and solve times are counted together. As before, the lower memory footprint of FAC allows solution of the largest

Copyright © 2016 John Wiley & Sons, Ltd.

Prepared using nlaauth.cls
DOI: 10.1002/nla

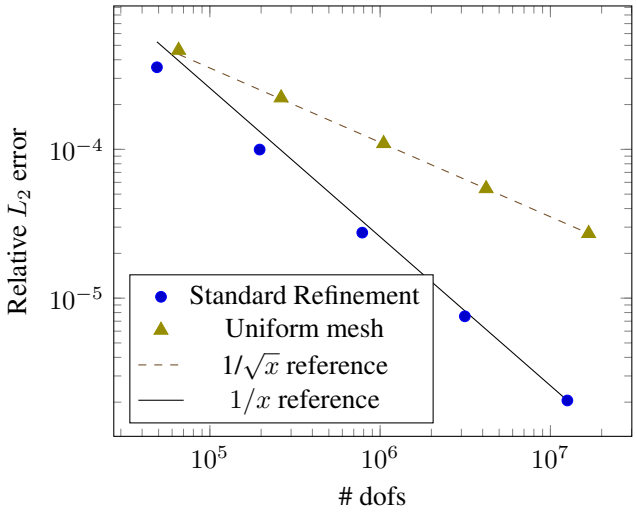

Figure 5. Convergence plot, impulse source term for uniform and local-then-global (standard) refinement patterns.

# dofs	relative L_2 error	setup time	solve time	# of PCG iter.
48894	3.56e-04	0.31	0.058	10
196094	9.97e-05	1.30	0.28	11
785406	2.76e-05	5.54	1.16	11
3143678	7.55e-06	23.45	4.91	11
12578814	2.05e-06	94.94	22.86	12
50323454	5.54e-07	401.3	123.5	12

Table VI. Finite-element accuracy and FAC solver performance for example with impulse source term on meshes following local-then-global refinement pattern.

# dofs	Matrix/RHS setup	AMG setup	Solve	# of of PCG iter.
48894	0.63	0.09	0.14	14
196094	1.68	0.34	0.59	14
785406	5.99	1.37	2.65	14
3143678	23.13	5.41	11.83	15
12578814	91.99	21.22	45.33	14

Table VII. AMG solver performance for example with impulse source term on meshes following local-then-global refinement pattern.

problem size, with over 50 million degrees of freedom, while AMG could not solve this problem within the memory available.

6.3. Two-spot problem

To demonstrate the flexibility of the FAC algorithm, we next consider an example with a smooth solution that changes rapidly at two points in the domain. To construct this example, we again consider Equation (9), now with source function $g(\omega)$ chosen so that the solution is given by

$$u(\theta,\phi) = \frac{1}{2}\sum_{i=0}^{\infty}(2i+1)e^{-i(i+1)\tau/2}P_i\left(\sin\theta\cos\phi\right) + \frac{1}{2}\sum_{i=0}^{\infty}(2i+1)e^{-i(i+1)\tau/2}P_i\left(\cos\theta\right),$$

with $\tau = 0.001$. This solution arises as a linear combination of fundamental solutions to the heat equation with initial data as the sum of two delta functions, one at the North pole ($\theta = 0$), and one

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

DOI: 10.1002/nla

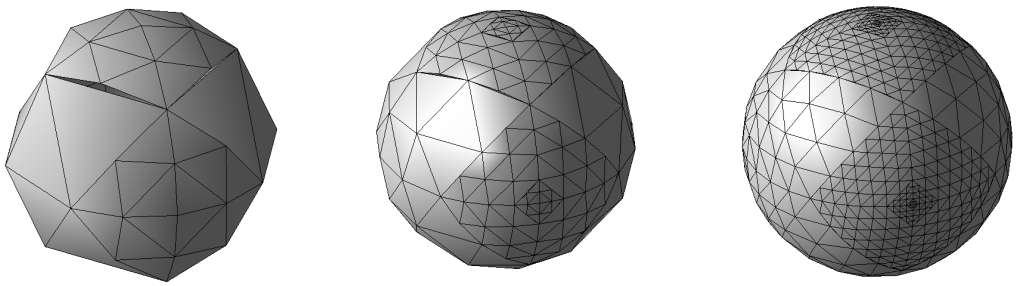

Figure 6. Three meshes obtained by applying the global-then-local refinement strategy for two-spot problem.

# dofs	relative L_2 error	setup time	solve time	# of PCG iter.
32514	1.81e-01	0.27	0.032	8
130562	4.43e-02	1.06	0.15	9
523266	1.10e-02	4.54	0.64	9
2095106	2.73e-03	18.88	2.66	9
8384514	6.80e-04	76.77	11.33	9
33546242	1.70e-04	311.9	48.63	9

Table VIII. Finite-element accuracy and FAC solver performance for two-spot problem.

# dofs	Matrix/RHS setup	AMG setup	Solve	# of of PCG iter.
32514	0.60	0.05	0.08	11
130562	1.63	0.22	0.33	12
523266	5.89	0.91	1.52	12
2095106	22.50	3.65	6.47	12
8384514	90.24	14.36	26.52	12
33546242	369.5	73.0	140.31	12

Table IX. AMG solver performance for two-spot problem.

on the equator ($\theta = \pi/2$, $\phi = 0$). Thus, this example models the behaviour of the first time-step in an implicit-in-time integration of the heat equation with time-step τ and a "two-spot" initial condition. To match the variation in the solution, we use a corresponding two-spot refinement pattern, with sample meshes pictured in Figure 6. In this example, we focus on the global-then-local refinement pattern, simply because this allows for two disjoint regions of refinement. Note now that each refinement patch consists of two (disconnected) pieces, but that the FAC algorithm as given above remains well-defined in such a case.

FAC and AMG solver performance are detailed in Tables VIII and IX, respectively, with accuracy proportional to the number of degrees of freedom verified by the relative L_2 errors reported in Table VIII. Similarly to the global-then-local refinement pattern results for the example in Section 6.1, these results show a consistent anomaly in the time required for the mesh construction and finite-element assembly in the AMG results. Aside from this, the same general conclusions can be drawn, with the FAC solve phase being about 2.5 times faster than that of the AMG solve phase, and over four times faster than the combined AMG setup and solve phases for the finest mesh with over 33.5 million degrees of freedom.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we extend the FAC algorithm to finite-element discretizations on triangulated surfaces, particularly focused on structured-grid refinements of the sphere. The FAC algorithm is shown to

Copyright © 2016 John Wiley & Sons, Ltd.

Prepared using nlaauth.cls
DOI: 10.1002/nla

be well-suited for such problems, with the composite-grid transfer operation naturally expressed in terms of two finite-element interpolation operators. For problems with localized sources, improved finite-element accuracy is seen using refined meshes, including restoration of effectively second-order accuracy for a problem with singular source.

Given the wide body of research currently ongoing into the accurate discretization of PDEs posed on surfaces or manifolds, a natural direction for future work is the comparison of FAC and multigrid algorithms for surface finite-element discretizations with solution algorithms for the closest-point method or other discretization approaches. Additionally, the work presented here was motivated by the development of fast algorithms for solution of the Boltzmann Transport Equation in (or close to) the Fokker-Planck limit. A key piece of future work is the coupling of the FAC algorithm presented here with more general scattering kernels and a suitable spatial discretization to address that problem in more detail.

REFERENCES

1. Strikwerda JC. *Finite difference schemes and partial differential equations*. Second edn., Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2004, doi:10.1137/1.9780898717938.
2. LeVeque RJ. *Finite difference methods for ordinary and partial differential equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007, doi:10.1137/1.9780898717839. Steady-state and time-dependent problems.
3. Brenner S, Scott L. *The mathematical theory of finite element methods*, *Texts in Applied Mathematics*, vol. 15. Springer-Verlag: New York, 1994.
4. Braess D. *Finite Elements*. Cambridge University Press: Cambridge, 2001. Second Edition.
5. Eymard R, Gallouët T, Herbin R. Finite volume methods. *Handbook of numerical analysis, Vol. VII*. Handb. Numer. Anal., VII, North-Holland, Amsterdam, 2000; 713–1020.
6. Berger MJ, Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics* 1984; **53**(3):484 – 512, doi:http://dx.doi.org/10.1016/0021-9991(84)90073-1.
7. Gui W, Babuska I. The h, p, and h-p versions of the finite element method in 1 dimension, part iii. the adaptive hp version. *Numer. Math.* 1986; **49**:659–683, doi:10.1007/BF01389733.
8. Bell J, Berger M, Saltzman J, Welcome M. Three-dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM Journal on Scientific Computing* 1994; **15**(1):127–138, doi:10.1137/0915008.
9. Verfürth R. A posteriori error estimation and adaptive mesh-refinement techniques. *J. Comput. Appl. Math.* May 1994; **50**(1-3):67–83, doi:10.1016/0377-0427(94)90290-9.
10. Morin P, Nochetto RH, Siebert KG. Convergence of adaptive finite element methods. *SIAM Review* 2002; **44**(4):631–658, doi:10.1137/S0036144502409093.
11. Bank RE, Holst M. A new paradigm for parallel adaptive meshing algorithms. *SIAM Review* 2003; **45**(2):291–323, doi:10.1137/S003614450342061.
12. Briggs WL, Henson VE, McCormick SF. *A Multigrid Tutorial*. SIAM Books: Philadelphia, 2000. Second edition.
13. Trottenberg U, Oosterlee CW, Schüller A. *Multigrid*. Academic Press: London, 2001.
14. Ruge JW, Stüben K. Algebraic multigrid (AMG). *Multigrid Methods*, *Frontiers in Applied Mathematics*, vol. 3, McCormick SF (ed.). SIAM: Philadelphia, PA, 1987; 73–130.
15. Stüben K. An introduction to algebraic multigrid. *Multigrid*, Trottenberg U, Oosterlee C, Schüller A (eds.). Academic Press: London, 2001; 413–528.
16. Swarztrauber PN. The direct solution of the discrete Poisson equation on the surface of a sphere. *J. Computational Phys.* 1974; **15**:46–54.
17. Floater MS, Hormann K. Surface parameterization: a tutorial and survey. *Advances in multiresolution for geometric modelling*. Math. Vis., Springer, Berlin, 2005; 157–186, doi:10.1007/3-540-26808-1-9.
18. Ruuth SJ, Merriman B. A simple embedding method for solving partial differential equations on surfaces. *J. Comput. Phys.* 2008; **227**(3):1943–1961, doi:10.1016/j.jcp.2007.10.009.
19. Macdonald CB, Ruuth SJ. The implicit closest point method for the numerical solution of partial differential equations on surfaces. *SIAM J. Sci. Comput.* 2009/10; **31**(6):4330–4350, doi:10.1137/080740003.
20. Dziuk G, Elliott CM. Finite element methods for surface PDEs. *Acta Numer.* 2013; **22**:289–396.
21. Barros SRM. Multigrid methods for two- and three-dimensional Poisson-type equations on the sphere. *J. Comput. Phys.* 1991; **92**(2):313–348, doi:10.1016/0021-9991(91)90213-5.
22. Chen Y, Macdonald CB. The closest point method and multigrid solvers for elliptic equations on surfaces. *SIAM J. Sci. Comput.* 2015; **37**(1):A134–A155, doi:10.1137/130929497.
23. Bey J. Finite-volumen- und mehrgitterverfahren für elliptische randwertprobleme. PhD Thesis, Eberhard-Karls-Universität Tübingen, Tübingen, Germany 1997.
24. Landsberg C, Voigt A. A multigrid finite element method for reaction-diffusion systems on surfaces. *Comput. Vis. Sci.* 2010; **13**(4):177–185, doi:10.1007/s00791-010-0136-2.
25. McCormick SF. Fast adaptive composite grid (FAC) methods: Theory for the variational case. *Defect Correction Methods: Theory and Applications*, Böhmer K, Stetter HJ (eds.). Computing Suppl. 5, Springer–Verlag: Vienna, 1984; 115–121.
26. McCormick S, Thomas J. The fast adaptive composite grid (FAC) method for elliptic equations. *Mathematics of Computation* 1986; **46**(174):439–456.

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

27. Lee B, McCormick SF, Philip B, Quinlan DJ. Asynchronous fast adaptive composite-grid methods: numerical results. *SIAM J. Sci. Comput.* 2003; **25**(2):682–700.
28. Lee B, McCormick SF, Philip B, Quinlan DJ. Asynchronous fast adaptive composite-grid methods for elliptic problems: theoretical foundations. *SIAM J. Numer. Anal.* 2004; **42**(1):130–152.
29. Börgers C, Larsen EW. On the accuracy of the Fokker-Planck and Fermi pencil beam equations for charged particle transport. *Medical Physics* Oct 1996; **23**(10):1749–59.
30. Lewis EE, Miller WF. *Computational Methods of Neutron Transport*. American Nuclear Society: La Grange Park, IL, 1993.
31. Morel JE, Manteuffel TA. An angular multigrid acceleration technique for S_n equations with highly forward peaked scattering. *Nuclear Science and Engineering* 1991; **107**:330–342.
32. Börgers C, MacLachlan S. An angular multigrid method for monoenergetic particle beams in Flatland. *J. Comp. Phys.* 2010; **229**:2914–2931.
33. Börgers C. A fast iterative method for computing particle beams penetrating matter. *J. Comput. Phys.* 1997; **133**(2):323–339.
34. Lee B. A novel multigrid method for SN discretizations of the mono-energetic Boltzmann transport equation in the optically thick and thin regimes with anisotropic scattering. I. *SIAM J. Sci. Comput.* 2009/10; **31**(6):4744–4773, doi:10.1137/080721480.
35. Lee B. Improved multiple-coarsening methods for SN discretizations of the Boltzmann equation. *SIAM J. Sci. Comput.* 2010; **32**(5):2497–2522, doi:10.1137/080742476.
36. Lee B. Space-angle-energy multigrid methods for S_n discretizations of the multi-energetic Boltzmann equation. *Numer. Linear Algebra Appl.* 2012; **19**(4):773–795, doi:10.1002/nla.808.
37. Lee B. A multigrid framework for S_n discretizations of the Boltzmann transport equation. *SIAM J. Sci. Comput.* 2012; **34**(4):A2018–A2047, doi:10.1137/110841199.
38. Gao H, Zhao H. A fast-forward solver of radiative transfer equation. *Transport Theory Statist. Phys.* 2009; **38**(3):149–192, doi:10.1080/00411450903187722.
39. Gao H, Zhao H. Analysis of a numerical solver for radiative transport equation. *Math. Comp.* 2013; **82**(281):153–172, doi:10.1090/S0025-5718-2012-02605-6.
40. Lau CY, Adams ML. Discrete-ordinates quadrature based on linear and quadratic discontinuous finite elements over spherical quadrilaterals. *Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*, Nashville, Tennessee, April 19-23, 2015.
41. Dziuk G. Finite elements for the Beltrami operator on arbitrary surfaces. *Partial differential equations and calculus of variations*, *Lecture Notes in Math.*, vol. 1357. Springer, Berlin, 1988; 142–155, doi:10.1007/BFb0082865. URL `http://dx.doi.org/10.1007/BFb0082865`.
42. McCormick S, Quinlan D. Asynchronous multilevel adaptive methods for solving partial differential equations on multiprocessors: performance results. *Parallel Comput.* 1989; **12**(2):145–156.
43. MFEM. Modular finite element methods library 2016. Http://mfem.org.
44. Henson V, Yang U. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics* 2002; **41**:155–177.
45. hypre. High performance preconditioners 2016. Http://www.llnl.gov/CASC/hypre/.

Copyright © 2016 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls