Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 202 (2005) 281–297
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A finite element method for three-dimensional unstructured grid smoothing

Glen Hansen [a,*], Andrew Zardecki [a], Doran Greening [b], Randy Bos [b]

[a] *Computational Science Methods Group, Applied Physics Division, MS F645, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[b] *Materials Science Group, Applied Physics Division, MS F699, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

Received 2 December 2003; received in revised form 30 June 2004; accepted 30 June 2004
Available online 8 September 2004

Abstract

The finite element method is applied to grid smoothing in three-dimensional geometry, generalizing earlier results obtained for planar geometry. The underlying set of equations for the Cartesian components of grid coordinates, based on the notion of harmonic coordinates, has a natural variational formulation. To estimate the target metric tensor that drives the elliptic grid equations, the metric tensor components are computed on a coarse-grained grid. Numerical examples illustrating the proposed approach are presented together with results from the smoothness functional, which is used to measure the quality of the resulting grid.

MSC: 65M50; 51P05; 65N30
Keywords: Finite elements; Galerkin methods; Mesh generation; Elliptic smoothing

1. Introduction

The field of two-dimensional mesh generation and smoothing is well developed; many approaches exist that have a successful history in a variety of applications. However, issues still remain in this area, including the need for robust methods that are effective for applications which possess a wide disparity of length scales and that provide acceptable performance on problem domains that are geometrically complex. Spe-

* Corresponding author. Tel.: +1 505 667 0655.
E-mail addresses: ghansen@lanl.gov (G. Hansen), azz@lanl.gov (A. Zardecki), dgreening@lanl.gov (D. Greening), rbos@lanl.gov (R. Bos).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2004.06.026

cifically, geometric issues are particularly evident in the vicinity of convex and concave boundaries, as has been documented in the monograph of Knupp and Steinberg [1].

Past experience with two-dimensional grid smoothing [2] indicates that satisfactory results may often be obtained for problems with convex or concave boundaries by solving a set of Laplace–Beltrami equations. This system generalizes to higher dimensional spaces through the use of harmonic coordinates [3]. In the literature, the concept of harmonic maps between two Riemannian manifolds is well established. Nishikawa [4] provides a formulation of the harmonic map based on a variational principle. The earliest work devoted to mesh applications is contained in the articles of Brackbill and Saltzman [5] and Dvinsky [6].

Solutions of a biharmonic system may also be advantageous for particular applications in grid smoothing [7,8]. Altas et al. [9] develop a three-dimensional biharmonic approach on a structured mesh, and employ multigrid and Krylov iterative methods to solve the resultant algebraic system. Helenbrook [10] advances a biharmonic method based on a Galerkin finite element method and presents results on unstructured, two-dimensional meshes. Additionally, Sparis [8] proposes a biharmonic method that uses a preconditioned conjugate-gradient solution strategy for structured, two-dimensional problems. These approaches are promising, but distinct from the Laplace–Beltrami method described here by targeting three-dimensional structured applications (Altas), and two-dimensional problems (Helenbrook and Sparis).

In common with other popular elliptic approaches, the biharmonic method does not perform well on domains with highly curved boundaries. For example, Fig. 4 of Sparis [8] shows a structured two-dimensional mesh on such a domain (the ''Volkswagen Beetle coordinate grid''). In this example, the inner boundary conforms to the profile of the automobile, where the outer boundary is intended to conform to the wind tunnel enclosure. The mesh generated by this implementation of the biharmonic method recedes from strongly curved boundaries (the Volkswagen windshield and corners of the wind tunnel). Although the equation system is different, this effect is also present with classical Laplacian smoothing and many of the popular methods compared in Knupp's *Rouge's Gallery* [1]. Sparis suggests that ''the addition of appropriate source terms on the right hand side of the biharmonic equation at the apexes'' might be a possible remedy to the results he obtained.

Sparis' hypothesis further reinforces the development of the Laplace–Beltrami method detailed in this paper. Indeed, one way to view the proposed Laplace–Beltrami method is that it is based on adding ''appropriate source terms'' (or *control functions*) to the popular Winslow–Crowley elliptic method, as discussed in Section 3 of Hansen et al. [2]. The result of adding these terms appears promising. Notice the similarities between the Volkswagen grid in Fig. 4 of Sparis [8] and Hansen's horseshoe test problem [2], where the Laplace–Beltrami method was used to minimize the repulsion of the mesh away from the boundary feature. These examples both have highly-curved boundaries of similar character; the outer boundary is strongly convex while the inner is concave. For these cases, the implicit control functions of the Laplace–Beltrami approach appear to inhibit the tendency of the mesh to recede away from convex boundaries toward concave areas. As a final example of the effectiveness of the Laplace–Beltrami approach, a more challenging three-dimensional version of this horseshoe with an additional boundary curvature feature is discussed in Section 4 of this paper.

The goal of this paper is to extend a previous formulation developed for two-dimensional meshes [2], to address three-dimensional, unstructured domains. This extension is based on the elliptic Laplace–Beltrami system that has its origins in differential geometry and quasi-conformal mapping. In the earlier paper, it was shown that this basic approach relates several popular grid generation methods, including Laplacian smoothing [11], the Winslow–Crowley method [12,13], and work advanced by Thompson [14,15], Knupp [1], Khamayseh [16], Tipton [17], and others. The Laplace–Beltrami approach was shown to be successful for two-dimensional domains with highly-curved boundaries and was contrasted with an unstructured Laplacian method [2]. This paper continues this comparison, considering three-dimensional problems possessing highly-curved boundaries.

In many cases, effective algorithms for two-dimensional applications are not easily extended to three-dimensional problems, due to various limitations in their development. Undesirable artifacts, such as algorithm convergence issues or insufficient operator approximation order, often severely influence the results when the approach is extended to three dimensions. Mesh folding (the presence of non-convex elements or elements with a negative Jacobian value), which may or may not have been an issue in the two-dimensional approach, is a much larger concern in three dimensions due to the additional degree of freedom of the elements and node points.

The additional dimension usually aggravates any convex/concave boundary issues that were present in the two-dimensional implementation. In this case, boundary geometric issues tend to be more complex as they typically manifest themselves in two or more directions, vs. the simple planar convexity (or concavity) cases encountered in two-dimensional applications. Further, the analysis of three-dimensional phenomena in general is rather difficult due to the sheer magnitude of data involved. The implementation of algorithms and test suites are also significantly more challenging than the addition of another dimension would initially suggest. These issues tend to limit progress of three-dimensional approaches.

This paper proposes a finite element solution for the harmonic equations in three dimensions, thus generalizing the results obtained for planar grids. In Section 2, equations governing the grid smoothing operation are derived. In Section 3, these equations are cast into a finite element form. Numerical examples illustrating the proposed procedure are developed in Section 4. Finally, conclusions and recommendations are formulated in Section 5.

2. The equations for grid coordinates

Consider a volumetric domain Ω in three-dimensional Euclidean space (x,y,z) whose coordinates are written as $x^i = (x^1,x^2,x^3)$. In local coordinates $u^\alpha = (u^1,u^2,u^3)$, a length element is written as

$$\mathrm{d}s^2 = g_{\alpha\beta}\,\mathrm{d}u^\alpha\,\mathrm{d}u^\beta, \tag{1}$$

where the covariant components of the metric tensor are given as

$$g_{\alpha\beta} = \sum_{i=1}^{3} \frac{\partial x^i}{\partial u^\alpha}\frac{\partial x^i}{\partial u^\beta}. \tag{2}$$

Throughout this development, the Einstein convention implies that summation is performed over repeated local mesh coordinate indices.

The contravariant components $g^{\alpha\beta}$ form the matrix inverse with respect to $g_{\alpha\beta}$. Thus

$$g_{\alpha\gamma}g^{\gamma\beta} = \delta^\beta_\alpha, \tag{3}$$

where δ^β_α denotes the Kronecker delta symbol. The determinant $g = \det(g_{\alpha\beta})$ is always positive since the length element $\mathrm{d}s^2$ is positive.

In two-dimensional surface geometry, given the local coordinates u^α on a surface, it is convenient to discuss isothermal coordinates $\hat{u}^\alpha$. In these coordinates, the line element is proportional to the Euclidean form that is written as a sum of squares of coordinate increments. These isothermal coordinates, viewed as functions of coordinates u^α, satisfy the Laplace–Beltrami equations [18]. A generalization of these considerations to three or more dimensions leads to the concept of harmonic coordinates, introduced by Lanczos [19] within the framework of the Einstein equations for gravitational fields. The harmonic coordinates x^i, defined by the condition

$$\Delta x^i = 0, \quad i = 1,2,\text{or } 3, \tag{4}$$

may be obtained from a variational formulation involving harmonic maps [4,20]. To demonstrate this, consider a mapping $f: M \to N$ between two Riemannian manifolds M and N with local coordinates u^α and x^i, respectively. If the metric tensors of M and N are $g_{\alpha\beta}(u)$ and $h_{ij}(x)$, the energy functional of the map f is defined as

$$E[f] = \frac{1}{2}\int_M g^{\alpha\beta}(u)\frac{\partial f^i}{\partial u^\alpha}\frac{\partial f^j}{\partial u^\beta}h_{ij}\sqrt{g(u)}\,\mathrm{d}^3u, \tag{5}$$

where $x^i = f^i(u)$ are the local coordinates of the manifold N and $g(u) = \det(g_{\alpha\beta})$. A transformation $f(u)$ is referred to as a harmonic mapping if f satisfies the Euler equations of the variational problem corresponding to Eq. (5). The Euler equations of the energy functional read

$$\frac{1}{\sqrt{g(u)}}\frac{\partial}{\partial u^\alpha}\left(\sqrt{g(u)}g^{\alpha\beta}\frac{\partial f^i}{\partial u^\beta}\right) + g^{\alpha\beta}\Gamma^i_{jk}\frac{\partial f^j}{\partial u^\alpha}\frac{\partial f^k}{\partial u^\beta} = 0, \tag{6}$$

where Γ^i_{jk} are the Christoffel symbols of the second kind on the manifold N, formed with the aid of the metric tensor h_{ij}. To derive Eq. (6), note that the metric tensor $h_{ij}(x)$ may vary when the variation of the functional is evaluated. This leads to the term involving the Christoffel symbols.

Harmonic mapping theory is significant in grid generation because the elliptic grid generator is obtained as a special case of the harmonic map by assuming that the target manifold N is Euclidean [21]. Indeed, in this case, the Christoffel symbols in the second term of Eq. (6) vanish, resulting in the Laplace equation for the coordinates x^i

$$\Delta x^i \equiv \frac{1}{\sqrt{g}}\frac{\partial}{\partial u^\alpha}\left(\sqrt{g}g^{\alpha\beta}\frac{\partial x^i}{\partial u^\beta}\right) = 0. \tag{7}$$

This expression represents the Laplacian operator acting on the scalar variables x^i [18], and forms the governing system used for the proposed mesh smoothing approach.

The smoothness quality metric used in this study is based on a generalization of an approach advanced by Knupp et al. [22]. This metric may be extended to three dimensions by stating it in terms of the energy functional presented in Eq. (5). Given a Euclidean target manifold, the metric tensor h_{ij} becomes diagonal: $h_{ij} = \delta_{ij}$. In this case, the energy functional becomes

$$E[g] = \frac{1}{2}\int_M \sum_{i=1}^{3} g^{ii}(x)\sqrt{g(x)}\,\mathrm{d}^3x, \tag{8}$$

where

$$g^{ii}(x) = \frac{\partial x^i}{\partial u^\alpha}\frac{\partial x^i}{\partial u^\beta}g^{\alpha\beta}(u), \tag{9}$$

and the invariant volume element $\sqrt{g(x)}\,\mathrm{d}^3x = \sqrt{g(u)}\,\mathrm{d}^3u$. Eq. (8) is used as the smoothness quality criteria in the examples to be presented later. The covariant metric tensor $g_{ij}(x)$, a transform of the metric tensor $g_{\alpha\beta}(u)$, is assumed to be a *prescribed* function of coordinates. Following the ideas of Dvinsky [6] and Brackbill [23], one is free to specify $g_{\alpha\beta}(u)$ in a manner most appropriate to accomplish grid smoothing. This concept will be expanded in the following sections.

3. The finite element approximation

The Galerkin method forms the basis of the finite element approximation to the solution of the system of Eq. (7). In this approach, the elliptic system is multiplied by a sufficiently smooth function w to produce a weak form. After integration by parts one obtains

$$\int_{\Omega} \frac{\partial w}{\partial u^{\alpha}} \sqrt{g} g^{\alpha\beta} \frac{\partial x^{i}}{\partial u^{\beta}} \, \mathrm{d}^3 u - \int_{\partial\Omega} w \sqrt{g} g^{\alpha\beta} \frac{\partial x^{i}}{\partial u^{\beta}} \, \mathrm{d}s_{\alpha} = 0, \tag{10}$$

where $\mathrm{d}s_{\alpha} = n_{\alpha} \, \mathrm{d}s$ in the second term refers to the integration over the boundary of Ω ($\partial\Omega$), which is a covariant oriented line element. In this context, n^{α} are the contravariant components of the unit outward normal.

For the Dirichlet boundary value problem, the grid coordinates x^{i} are specified with the boundary conditions $x^{i} = \hat{x}^{i}$ on $\partial\Omega$. In this case, the set of equations

$$\int_{\Omega} \frac{\partial w}{\partial u^{\alpha}} \sqrt{g} g^{\alpha\beta} \frac{\partial x^{i}}{\partial u^{\beta}} \, \mathrm{d}^3 u = 0, \tag{11}$$

leads to a nonlinear algebraic system

$$\sum_{n=1}^{N} K_{mn}(a) a_{n}^{i} = 0, \quad i = 1, 2, \text{or } 3, \tag{12}$$

where the stiffness matrix K_{mn} is

$$K_{mn} = \int_{\Omega} \frac{\partial \psi_{m}}{\partial u^{\alpha}} \sqrt{g} g^{\alpha\beta} \frac{\partial \psi_{n}}{\partial u^{\beta}} \, \mathrm{d}^3 u. \tag{13}$$

In Eq. (12), the unknown coefficients a_{n}^{i} are the expansion coefficients of the coordinates x^{i} in terms of the basis functions $\psi_{n}(u)$

$$x^{i}(u) = \sum_{n=1}^{N} a_{n}^{i} \psi_{n}(u), \quad i = 1, 2, \text{or } 3. \tag{14}$$

Only first-order finite element basis functions are considered for this development. In the case of hexahedral elements, for example, trilinear basis functions are obtained by converting the functions

$$\psi_{i}(\xi, \eta, \zeta) = \frac{1}{8}(1 + \xi\xi_{i})(1 + \eta\eta_{i})(1 + \zeta\zeta_{i}), \quad i = 1, \ldots, 8 \tag{15}$$

defined on a master element $\Omega_{\xi} = \{\xi: -1 \leqslant \xi \leqslant 1, -1 \leqslant \eta \leqslant 1, -1 \leqslant \zeta \leqslant 1\}$, to the functions defined on element Ω_{e}. In Eq. (15), the variables ξ_{i}, η_{i} and ζ_{i} assume values of ± 1, depending on the nodal location.

The crucial step in the finite element scheme is the evaluation of the metric tensor in Eq. (13), which controls the properties of the final mesh. Consider the transformation of the master element Ω_{ξ} into element Ω_{e} given in terms of the node coordinates. In this case, x_{m}^{e}, y_{m}^{e}, z_{m}^{e} of the nodes N_{m}^{e}, $m = 1, \ldots, M$, form element Ω_{e}, or

$$\begin{aligned} x_{e} &= \sum\nolimits_{m=1}^{M} x_{m}^{e} \psi_{m}(\xi), \\ y_{e} &= \sum\nolimits_{m=1}^{M} y_{m}^{e} \psi_{m}(\xi), \\ z_{e} &= \sum\nolimits_{m=1}^{M} z_{m}^{e} \psi_{m}(\xi). \end{aligned} \tag{16}$$

The components of the *current* metric tensor $g_{\alpha\beta}^{e}$, associated with element Ω_{e}, are described by the equations

$$g_{\alpha\beta}^{e} = \sum_{m=1}^{M} \sum_{n=1}^{M} (x_{m}^{e} x_{n}^{e} + y_{m}^{e} y_{n}^{e} + z_{m}^{e} z_{n}^{e}) \frac{\psi_{m}}{\partial \xi^{\alpha}} \frac{\psi_{n}}{\partial \xi^{\beta}}. \tag{17}$$

Unfortunately, the use of the current metric tensor, being *descriptive* of the current grid, will not result in any smoothing of the mesh. Said another way, if one evaluates the definition of the metric (Eq. (17)) using

the current nodal coordinates, a metric tensor that describes the current mesh would be obtained. If this tensor is then used to compute new nodal coordinates using Eq. (12), these new values will not differ from the previous values and the mesh will not change. Therefore, to solve for the nodal coordinates of the smoothed mesh, one must supply a modified metric tensor that will produce the nodal coordinates desired in the solution. To address this issue in two-dimensional geometry, an estimate of this target (or prescriptive) metric tensor was obtained by considering the centers of elements neighboring Ω_e, which is analogous to calculating the metric tensor on a dual mesh. In the three-dimensional case, an element is often adjacent to a large number of elements of various shapes. In three-dimensions, it is usually more expedient to evaluate the target metric tensor on a synthetic element, which is best described by a procedure called *coarse-graining*. Coarse-graining is an equidistribution technique where each node N_m^e of an element e is replaced by a synthetic node that is located in a "more optimal" location than the original. This coarse-grained grid uses adjacency information, which for each node N_m^e is the set of neighboring nodes (those nodes connected to N_m^e by an element edge). Then, given the set neighbor nodes, the synthetic location of node N_m^e with the position $\mathbf{r}_m$ is obtained by averaging their coordinates.

The coarse-grained components of the metric tensor are again computed using Eq. (17); however, the coordinates (x_m^e, y_m^e, z_m^e) of node N_m^e are replaced by $(\overline{x}_m^e, \overline{y}_m^e, \overline{z}_m^e)$, which are the average of the coordinates of the nodes linked to N_m^e by an element edge. This average is unity weighted, calculated by summing the graph-connected adjacent node coordinates then normalizing by the number of connecting edges. In other words, for the vertex N_m^e, the coarse-grained position vector is

$$\overline{\mathbf{r}}_m^e = \frac{1}{N}\sum_{n=1}^{N} \mathbf{r}_{mn}, \tag{18}$$

where the sum includes all the vertices linked to node N_m^e by $\mathbf{r}_{mn}$.

In the example depicted in Fig. 1, a two-dimensional triangular element is defined by vertices A, B, and C. The coordinates of the vertex A are replaced by an arithmetic average of coordinates of vertices a, B, and C. Similarly, the coordinates of A, C, and d produce the coarse-grained coordinates of B, whereas the coordinates of A, B, a, b, and c yield the coarse-grained coordinates of C.

The coarse-graining operator is reminiscent of common Laplace grid smoothing [11], where the node positions are shifted to average positions. This averaging process is one form of a numerical approximation of the Laplace operator applied to a scalar. It is important to stress that using a Laplace operator to generate a synthetic metric for the finite element method is quite distinct from a Laplacian mesh smoothing algorithm. In Laplace smoothing, the converged average node positions completely define the ultimate

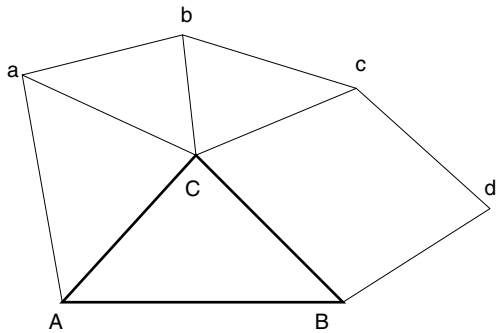

Fig. 1. The nodes of the triangular element ABC and their connectivity to other nodes determine the coarse-grained metric tensor.

smoothed grid. In the coarse grained metric approach, the average node positions only determine the target metric tensor that suggests a smoother mesh locally about element Ω_e. The ultimate position of the mesh nodes is still determined as a solution to the elliptic set of Eqs. (4).

Typically, to solve the system (4), one partitions the domain Ω into a collection of finite elements $\Omega = \{\Omega_e\}$, resulting in a nonlinear algebraic system (12). The nonlinearity arises from the dependence of the contravariant metric tensor $g^{\alpha\beta}$ on the coefficients a_n^i. This creates an issue in solution process; it is often challenging to solve nonlinear algebraic problems of this form. Furthermore, as this is a three-dimensional application, the total number of equations that must be solved is generally of respectable magnitude. Lastly, the desire for the ultimate application to begin calculation from a predictable initial mesh state often mandates a converged solution to the nonlinear problem. These three considerations typically lead to the requirement to use a robust, efficient parallel solver for the base nonlinear problem.

In this paper, a Newton–Krylov approach was used to solve the system (12). The implementation details were similar to the two-dimensional case [2]. To summarize the approach, Newton's method was applied to the nonlinear finite element problem [24], where the target metric is computed implicitly within the solution procedure. This approach is based on solving a system of nonlinear algebraic equations

$$\vec{F}(\vec{u}) = \begin{pmatrix} F_1(x_1, y_1, z_1, \ldots, x_N, y_N, z_N) \\ \vdots \\ F_N(x_1, y_1, z_1, \ldots, x_N, y_N, z_N) \end{pmatrix} = 0. \tag{19}$$

Given this form, the Newton–Krylov solution procedure is based on using a Krylov subspace method to solve the linear system

$$J^i \, \delta\vec{u}^i = -\vec{F}(\vec{u})^i, \tag{20}$$

where

$$J_{kl} = \sum_e J_{kl}^e = \sum_e \frac{\partial F_k^e(\vec{u})}{\partial u_l} \tag{21}$$

is the Jacobian matrix, $\vec{F}(\vec{u})^i$ is the residual vector for the ith iterate, and $\vec{u} = (x_1, y_1, z_1, \ldots, x_N, y_N, z_N)$ is the problem state vector. The solution may be advanced to the next iteration using the expression

$$\vec{u}^{i+1} = \vec{u}^i + \delta\vec{u}^i. \tag{22}$$

This form of the Jacobian is based on interchanging the differentiation operator and the finite element assembly process [25], which simplifies implementation of the solution procedure. Finally, the nonlinear equation template is

$$F_m^e(\vec{u}) = \sum_n \int_{D_e^\xi} \frac{\partial \varphi_m}{\partial \xi^\alpha} \frac{\partial \varphi_n}{\partial \xi^\beta} \, \tilde{g}_e^{\alpha\beta}(\vec{u}) \sqrt{\tilde{g}_e(\vec{u})} \, \mathrm{d}^3\xi \; u_n - f_m, \tag{23}$$

where the tilde over the metric factors (e.g. $\tilde{g}$) implies the use of the coarse-grained approximation, and f_m is the term arising from the Dirichlet boundary conditions for the mth equation.

As a stopping criterion, the nonlinear iteration is halted when the residual vector is sufficiently small. In this implementation, the convergence criteria is $\|\vec{F}^i(\vec{u})\|_2 / \|\vec{F}^0(\vec{u})\|_2 \leqslant 10^{-6}$, where $\vec{F}^0(\vec{u})$ is the residual vector of the initial "guess." A block preconditioner toolkit, BPKIT [26], was used to solve the linear system (20). For all the results presented here, the toolkit's incomplete LU factorization option ILU(1) was used as the local preconditioner, with two passes of symmetric successive over-relaxation (SSOR) used as a global preconditioner. There were 16 blocks used in the preconditioner, and the toolkit's flexible general minimal residual linear solver (FGMRES) was used as the Krylov solver.

4. Numerical examples

The examples presented in this section begin with a three-dimensional grid generated on an arc-extruded horseshoe domain. Variants of the horseshoe mesh were demonstrated to be challenging test problems for various two-dimensional approaches [1,2]. One of the more difficult aspects of this domain is the boundary curvature at the upper-left and upper-right areas of the top boundary, and the inner corners of the obstacle. To form the three-dimensional mesh, the planar horseshoe is extruded following a constant radius curve. This extrusion process forms a horseshoe-torus as illustrated in Fig. 2, and results in a third curvature feature along the extrusion direction.

Fig. 3 compares the converged results of the proposed finite element Laplace–Beltrami method with 500 Gauss–Seidel iterations of Laplacian smoothing on the horseshoe-torus domain. Note that the proposed method provides similar properties to that seen in the two-dimensional results [2], the additional curvature feature does not affect the mesh in a negative fashion. The Laplacian smoothing result, however, severely folds the mesh near the external obstacle, much like the two-dimensional results shown for this method.

Figs. 4 and 5 illustrate the smoothing of a larger grid contained within a spherical region. The original grid consists of prisms (extruded triangles) and hexahedral elements, and also exhibits several smoothing challenges. First, the outside boundary of the sphere is convex, the mesh here curves at a constant radius r around the outer surface. Laplacian smoothing approaches tend to uniformly "draw" the mesh away from this surface and attempt to collapse it toward the center of the sphere. The second feature to note is the cylindrical mesh pattern created by the axisymmetric prisms located around the vertical axis of the sphere. This topology increases the tendency of the mesh to collapse toward the center of the sphere. Finally, there is the transition region between the cylindrical mesh topology of the problem axis and the spherical topology present near the outside surface of the sphere. This feature, located roughly 1/4 of the way down from the top of the sphere and 1/2-way between the axis and the sphere surface, is most apparent in the unsmoothed mesh. This construct is a second degenerate form, where three hexahedral elements share a common edge, in this case normal to the plane of the view. Laplacian smoothing methods tend to collapse these elements toward the degenerate edge. In this particular case this effect is further enhanced as the degenerate edges join to form a circle in the plane perpendicular to the symmetry axis of the

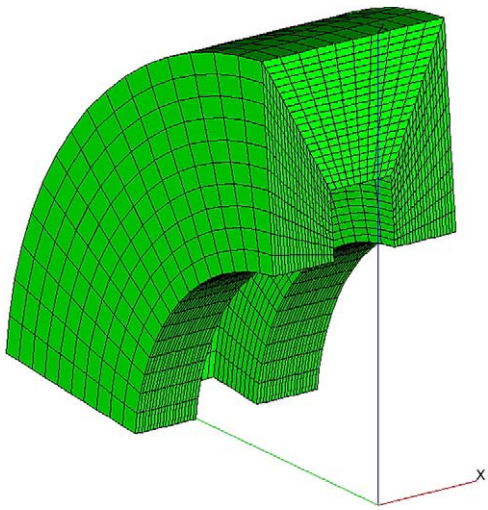

Fig. 2. Modified horseshoe example: A planar horseshoe-mesh extruded along a constant radius arc to form a torus (1/4 of torus shown).

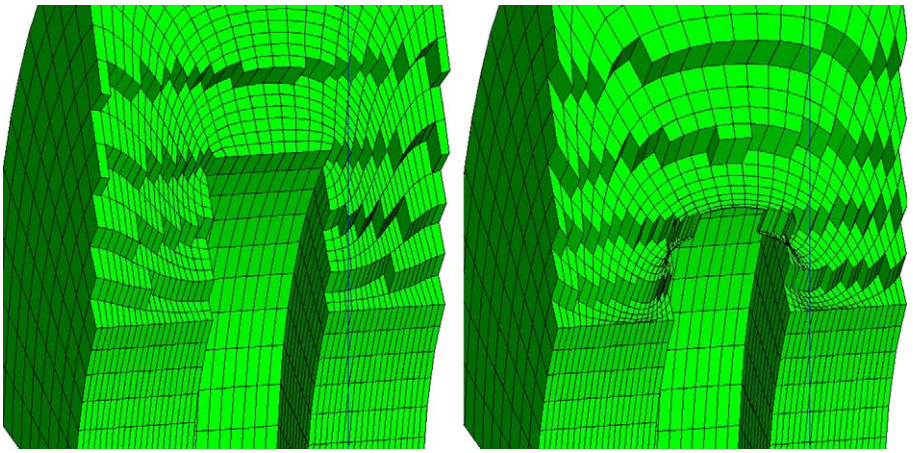

Fig. 3. Comparison of a horseshoe-torus mesh enhanced using the finite element Laplace–Beltrami method (left) and Laplacian smoothing (right). Note severe mesh folding near the internal obstacle when Laplace smoothing is used.

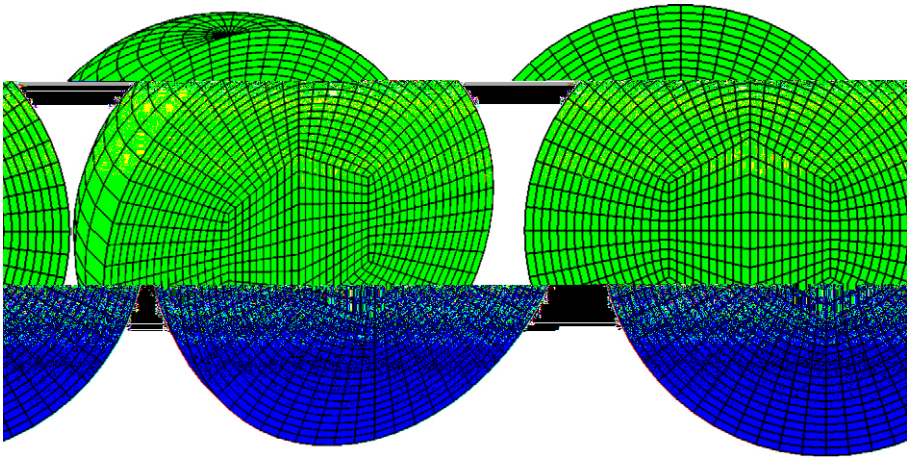

Fig. 4. The grid within a sphere consisting of prisms and hexahedral elements.

sphere. Laplace-based smoothing tends to collapse the aggregate mesh toward the center of the sphere, as shown in Fig. 6. Although the final result is generally smooth, it is likely not of use to a simulation application due to the excessive concentration of mesh near the axis of the problem along with the rarefaction near the surface of the sphere. In contrast, the finite element method easily overcomes the tendency of the mesh to collapse, as seen in Fig. 5. Indeed, this approach actually forces the mesh outward from its initial position toward the surface while smoothing the domain.

The previous examples suggest that the proposed finite element smoother is fully effective on three-dimensional boundary-fitted meshes that are enclosed by curved boundary surfaces. In this case, the horseshoe-torus is derived from a structured mesh, and the sphere example is block structured. The method does not utilize this structure in implementation; however, the question remains concerning the effectiveness of the approach for more traditional unstructured meshes. Fig. 7 depicts a cylindrical region that is capped by two hemispheres, which provides an example of an inherently unstructured tetrahedral grid. A planar cut

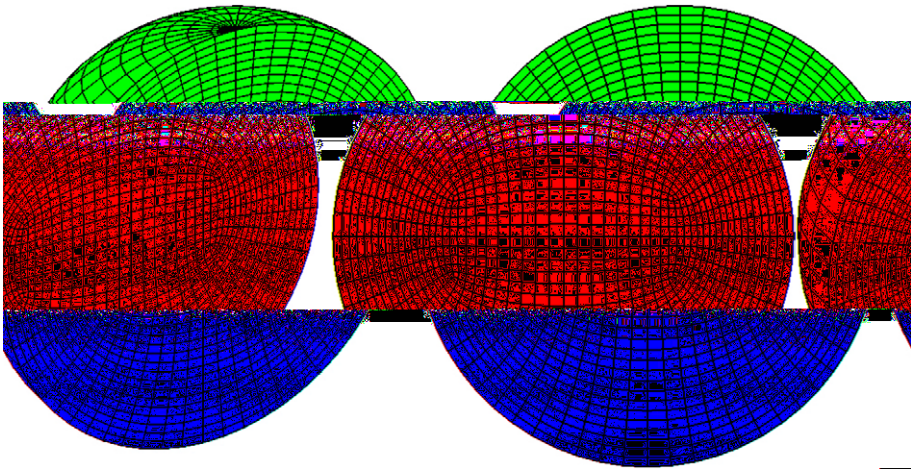

Fig. 5. The smoothed grid corresponding to the grid of Fig. 4.

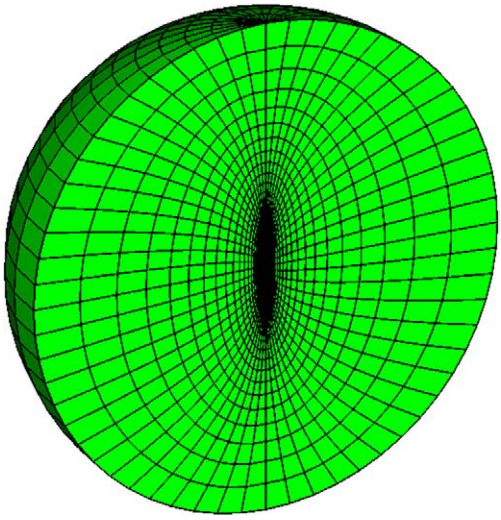

Fig. 6. Classic Laplacian smoothing applied to the sphere problem. Solution shown after 500 Gauss–Seidel iterations were performed.

through the region of the cylinder is shown in Fig. 8. The same planar cut obtained after grid smoothing is shown in Fig. 9. Visually, it is difficult to tell what has occurred during the smoothing process comparing these two views. Unlike two-dimensional diagrams where the effectiveness of the global algorithm is readily apparent, these three-dimensional cutaways expose only one surface of the interior mesh and a few node points. Secondly, the initial mesh is of relatively high quality; the smoother does not move any node point drastically as it converges. However, if one carefully compares the two meshes near the center and along the 3 o'clock axis, some smoothing is apparent (assume the clock sits on the cutaway surface with the 12 o'clock position pointing directly away from the user).

Fig. 10 shows an unstructured tetrahedral grid representing a turbine nozzle, courtesy of the *amira*™ software package by TGS, Inc. (www.tgs.com). A planar cut through this grid is shown in Fig. 11, whereas

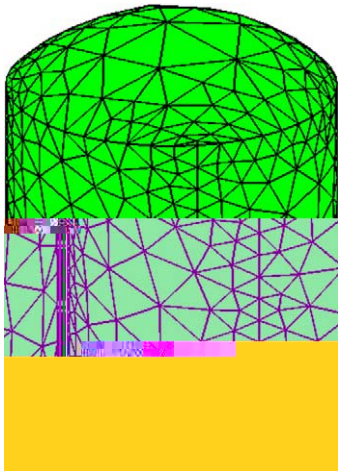

Fig. 7. Tetrahedral grid within a cylindrical surface with two spherical caps.

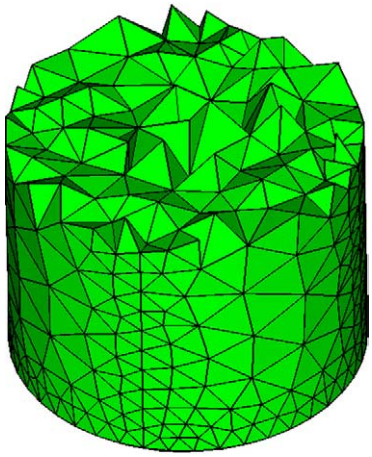

Fig. 8. A planar cut through the grid of Fig. 7.

a planar cut through the smoothed grid is shown in Fig. 12. In these figures, the effect of smoothing is more apparent. Particularly in the region directly opposite of the viewer, several nodes have been moved toward more optimal locations (evaluated visually). All the unstructured domains require a quantitative quality metric to indicate if real quality improvement has occurred, however.

The smoothness quality metric used in this study is based on a generalization of an approach advanced by Knupp et al. [22]. This metric may be extended to three dimensions by expressing it in terms of the contravariant components of the descriptive metric tensor (c.f., Eq. (8)). Table 1 lists the normalized smoothness functional (NSF) values for the results considered in this section. In this table, the values of the functional vs. the iteration number are normalized to unity by dividing the successive values by the values corresponding to the initial mesh. In all cases considered, the finite element smoother improved the quality of the input mesh, as measured by the smoothness functional.

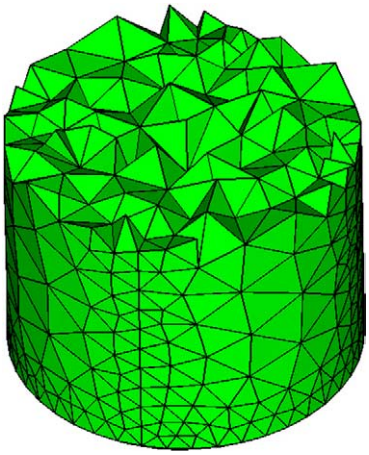

Fig. 9. The smoothed grid corresponding to the grid of Fig. 8.

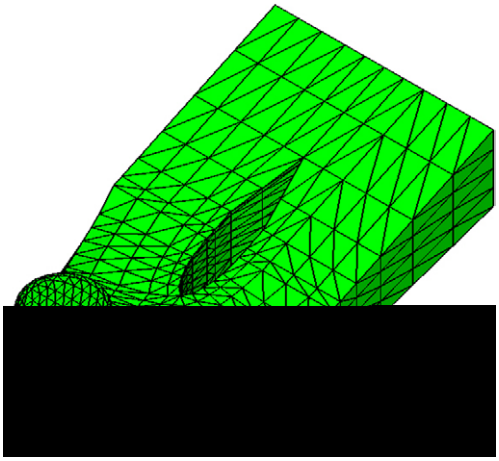

Fig. 10. The turbine tetrahedral grid. Image courtesy of the *amira*™ software package by TGS, Inc. (www.tgs.com).

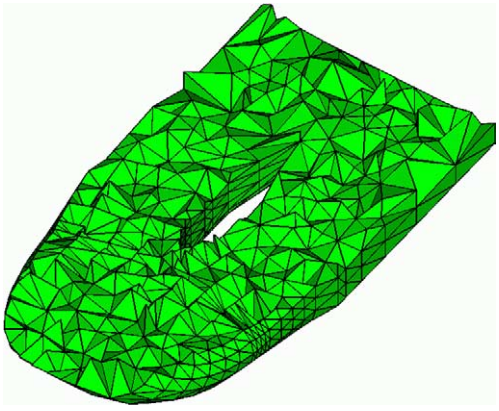

Fig. 11. A cut through the grid of Fig. 10.

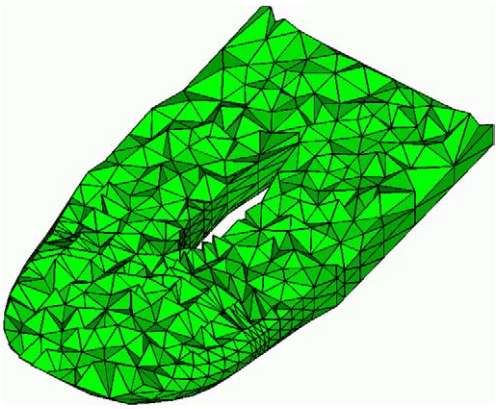

Fig. 12. The smoothed grid corresponding to the grid of Fig. 11.

Table 1
Value of the normalized smoothness functional (NSF) vs. Newton iteration number for the 3D examples considered

Grid	Number of unknowns	Newton iteration number					
		1	2	3	4	5	6
Turbine	1926	1.0000	0.8295	0.8296	0.8296	0.8296	0.8296
Cylinder	2202	1.0000	0.9227	0.9229	0.9229	0.9229	0.9229
Horseshoe-torus	4959	1.0000	0.9262	0.9065	0.9126	0.9129	0.9129
Sphere	55233	1.0000	0.9842	0.9461	0.9421	0.9417	0.9417

Table 2 compares the NSF values obtained using Laplacian and Laplace–Beltrami smoothing, respectively, on the example set. Based on this metric, the proposed Laplace–Beltrami method provided superior results, quantitatively, than the Laplacian method. The Laplace method failed to yield a qualitatively-acceptable mesh on both the horseshoe-torus and sphere problems. This result is indicated by missing entries in the table.

The failure of the Laplacian method is not surprising; the literature discusses the tendency of this approach to yield low-quality results for domains with large-degrees of boundary curvature [1,2,11,15]. Two-dimensional results further indicate that many popular methods do not perform well on these boundary-dominated problems [1,8,11]. These alternative methods were not examined for comparison purposes here, as one would expect that their performance would only degrade further due to the presence of the third dimension.

Fig. 13 shows the nonlinear convergence rate of Newton's method for the example set. The dotted line indicates the line denoting one order of magnitude reduction of the residual with each Newton iteration. The structured and semi-structured examples (the horseshoe-torus and sphere problems) indicate excellent performance, with near-logarithmic residual reduction. However, the unstructured cases (the turbine and cylinder) exhibit exceptional convergence rates. From this limited data, it is not possible to deduce if the convergence rate is a function of problem size. Indeed, the rates of both the smallest boundary-conformal problem considered (the horseshoe-torus with 4959 unknowns) compared with the largest (the sphere at 55,233 unknowns) are similar. Actually, similarities in the mesh topology (unstructured vs. (semi)-structured) appears to have a stronger influence on the convergence rate of Newton's method, at least for the examples considered.

Table 2
Value of the normalized smoothness functional (NSF) obtained from the Laplace–Beltrami (LB) method and classical Laplacian smoothing

Grid	NSF metric	
	Laplacian	LB
Turbine	0.9967	0.8296
Cylinder	0.9879	0.9229
Horseshoe-torus	–	0.9129
Sphere	–	0.9417

The LB results are fully-converged values and the Laplacian results are for 500 Gauss–Seidel iterations. Missing entries signify failure of the method (see text).

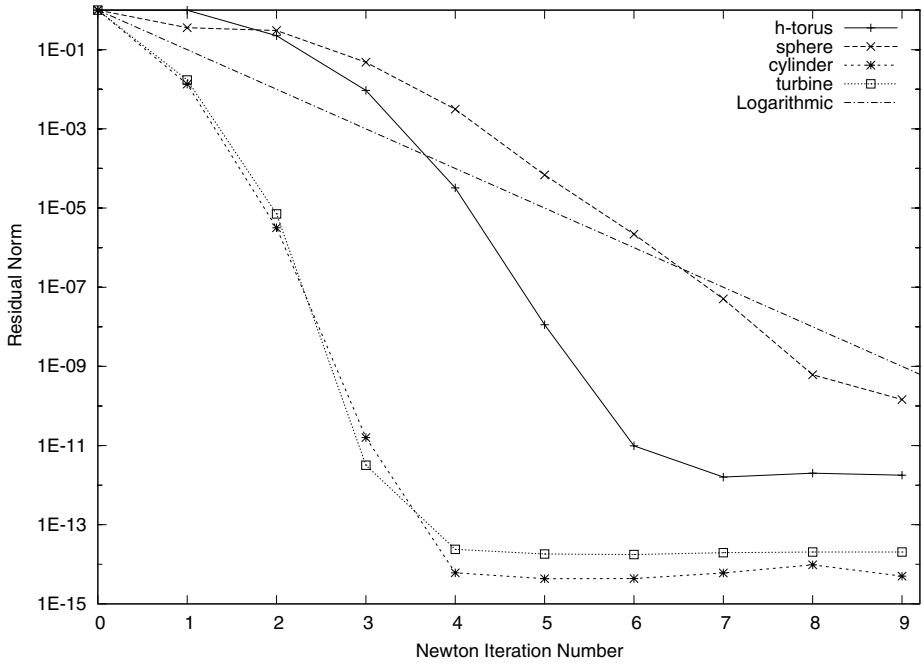

Fig. 13. Newton's method convergence plot of the example set.

The diagram set, Fig. 14, illustrates the linear convergence of FGMRES(50) on this set of problems, where 50 Krylov vectors were stored during the solution. The upper-left diagram shows the linear convergence history on the horseshoe-torus domain, where the vertical axis is the relative residual reduction metric calculated by the FGMRES algorithm from BPKIT. Each curve shows the FMGRES history for a particular Newton iteration number, which is also plotted to the right of the curves. Similarly, the upper-right diagram shows the linear convergence history on the sphere problem, with the lower-left and lower-right diagrams corresponding to the cylinder and turbine domains, respectively.

With the exception of the sphere problem, all other examples exhibit acceptable to excellent linear convergence behavior. The sphere problem exhibits numerous stalls in the FGMRES convergence behavior and also the inability to converge after two restarts (150 total FGMRES iterations). The sphere possesses several unique mesh constructs and curvature artifacts which may be responsible for this observed behavior. The number of unknowns in the sphere problem is also much greater than the other examples. Further

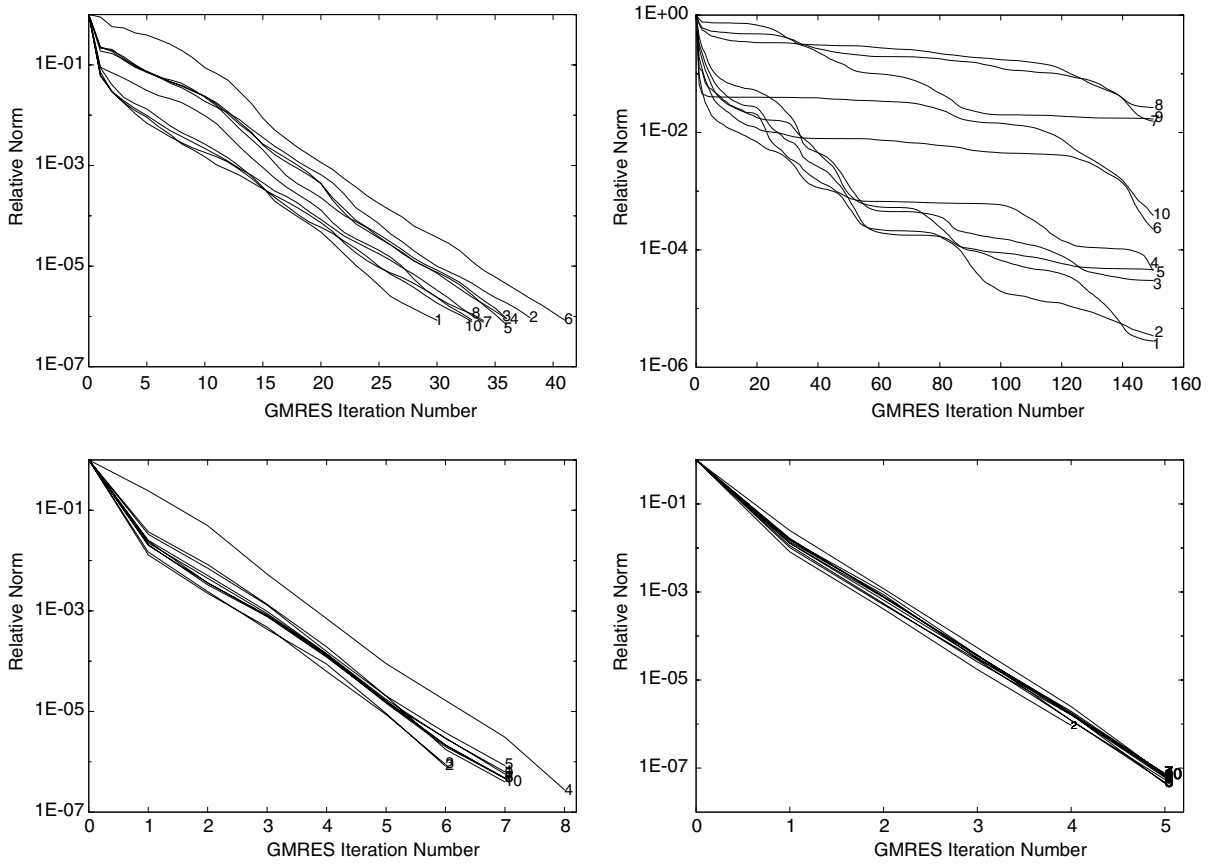

Fig. 14. Plots of the linear convergence of FGMRES(50) on the example set. The upper-left diagram is for the horseshoe-torus problem, the upper-right corresponds to the sphere, the lower-left is the cylinder, and the lower right is the turbine.

work is necessary to explore the reasons for this convergence behavior, and develop methodologies to mitigate this problem. As an interesting aside, this linear convergence issue does not appear to appreciably degrade the performance of Newton's method on the nonlinear problem (see Fig. 13).

In closing, these solution results are based on using the Newton–Krylov method to converge the discrete equation system. In general, full convergence is not always employed as a solution requirement in grid smoothing. In fact, many approaches are based on the use of various relaxation methods that are iteratively-applied until the desired visual result is achieved. For example, in a particular application, it may be that 100 iterations of a Gauss–Seidel procedure result in the desired mesh, i.e.,visually-smooth. Thus, the Newton–Krylov approach may or may not be the most desirable or attractive solution procedure for the actual implementation of the Laplace–Beltrami approach. Its use here is primarily motivated by two factors: (1) the need for a converged end-state to be used to compare and contrast results, and (2) repeatability of the solution results of the ultimate physics application may demand a consistent initial problem state.

5. Conclusions

This paper generalizes the finite element methodology developed previously for planar grids [2] to three-dimensions. The approach is based on a finite element solution to the set of Laplace–Beltrami equations for

an unstructured mesh. In a manner similar to the planar algorithm, the method is based on specifying a target metric, where the target metric is a prescription developed to equidistribute mesh characteristics locally in the neighborhood of each element. This prescriptive metric is refreshed each nonlinear iteration. This iterative process guides the final mesh solution to a higher-quality result relative to the initial mesh, given the quality criterion used for evaluation in this paper.

Unlike the two-dimensional algorithm, the target metric is estimated using a process called *coarse graining*, where a Laplacian-like algorithm is used to estimate a metric that is locally smoother than the current metric. This approach is much simpler to implement in three dimensions than the dual-mesh method employed for the planar problem. This coarse-grained approach appears to be fully satisfactory for the example set considered. Left for future activities is a detailed examination of more complex prescriptive approaches that are better suited to capturing disparate length scales and solution features within the eventual application. Furthermore, three-dimensional applications challenge solution methods because of the large number of unknowns in the algebraic system; they also possess challenging geometric characteristics that surpass the complexity of the sphere and horseshoe-torus examples considered here. Future work must also include a detailed analysis of the convergence issues discovered on the sphere problem, along with the development of an effective approach to address these issues.

Acknowledgement

This work was performed under the auspices of the US Department of Energy by Los Alamos National Laboratory under Contract W-7405-ENG-36 (LA-UR-03-8671).

References

[1] P. Knupp, S. Steinberg, Fundamentals of Grid Generation, CRC Press, Boca Raton, FL, 1994.
[2] G. Hansen, A. Zardecki, D. Greening, R. Bos, A finite element method for unstructured grid smoothing, J. Comput. Phys. 194 (2004) 611–631.
[3] V. Fock, The Theory of Space Time and Gravitation, Pergamon Press, New York, 1959.
[4] S. Nishikawa, Variational Problems in Geometry, American Mathematical Society, Providence, RI, 2002.
[5] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in two dimensions, J. Comput. Phys. 46 (1982) 342–368.
[6] A.S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifolds, J. Comput. Phys. 95 (1991) 450–476.
[7] G.B. Shubin, A.B. Stephens, J.B. Bell, Three dimensional grid generation using biharmonics, in: J.F. Thompson (Ed.), Numerical Grid Generation, Elsevier Science, New York, 1982, pp. 761–774.
[8] P.D. Sparis, A. Karkanis, Boundary-orthogonal biharmonic grids via preconditioned gradient methods, AIAA J. 530 (1992) 671–678.
[9] I. Altas, J. Erhel, M.M. Gupta, High accuracy solution of three-dimensional biharmonic equations, Num. Algorithms 29 (2002) 1–19.
[10] B.T. Helenbrook, Mesh deformation using biharmonic operator, Int. J. Num. Methods Eng. 56 (2003) 1007–1021.
[11] P.M. Knupp, Winslow smoothing on two-dimensional unstructured meshes, in: Seventh International Meshing Roundtable, Sandia National Laboratory, 1998, pp. 449–457.
[12] A.M. Winslow, Equipotential zoning of two dimensional meshes, Tech. Rep. UCRL-7312, Lawrence Livermore National Laboratory, 1963.
[13] W.P. Crowley, An equipotential zoner on a quadrilateral mesh, Tech. rep., Lawrence Livermore National Laboratory, memorandum, 1962.
[14] J.F. Thompson, F.C. Thames, C.W. Mastin, Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies, J. Comput. Phys. 15 (1974) 299–319.
[15] J.F. Thompson, Z.U.A. Warsi, C.W. Mastin, Numerical Grid Generation: Foundations and Applications, North Holland, New York, NY, 1985.
[16] A. Khamayseh, G. Hansen, Quasi-orthogonal grids with impedance matching, SIAM J. Sci. Comput. 22 (4) (2000) 1220–1237.
[17] R.E. Tipton, Grid optimization by equipotential relaxation, Tech. rep., Lawrence Livermore National Laboratory, July 1992.

[18] D.J. Struik, Lectures in Classical Differential Geometry, Dover, New York, 1961.
[19] K. Lanczos, Ein vereinfachenden Koordinatensystem für die Einsteinschen Gravitationsgleichungen, Phys. Zeitschrift 24 (1922) 537–539.
[20] F. Hélein, Harmonic Maps, Conservation Laws and Moving Frames, Cambridge University Press, Cambridge, 2002.
[21] V.D. Liseikin, Grid Generation Methods, Springer, Berlin, Heidelberg, New York, 1999.
[22] P. Knupp, L. Margolin, M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods, J. Comput. Phys. 176 (1) (2002) 93–128.
[23] J.U. Brackbill, An adaptive grid with directional control, J. Comput. Phys. 108 (1993) 38–50.
[24] X.-C. Cai, W.D. Gropp, D.E. Keyes, R.G. Melvin, D.P. Young, Parallel Newton–Krylov–Schwarz algorithms for the transonic full potential equation, SIAM J. Sci. Comput. 19 (1) (1998) 246–265.
[25] A. Stagg, G. Hansen, A. Zardecki, Comments on applying Newton's method to nonlinear finite element problems, Tech. Rep. LAUR-03-6657, Los Alamos National Laboratory, September 2003.
[26] E. Chow, M.A. Heroux, An object-oriented framework for block preconditioning, ACM Trans. Mathematical Software 24 (1998) 159–183.