INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2004; **61**:2402–2427
Published online 18 October 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1147

Common-refinement-based data transfer between non-matching meshes in multiphysics simulations

Xiangmin Jiao[‡] and Michael T. Heath[*,†]

Computational Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.

SUMMARY

In multiphysics simulations using a partitioned approach, each physics component solves on its own mesh, and the interfaces between these meshes are in general non-matching. Simulation data (e.g. jump conditions) must be exchanged across the interface meshes between physics components. It is highly desirable for such data transfers to be both numerically accurate and physically conservative. This paper presents accurate, conservative, and efficient data transfer algorithms utilizing a common refinement of two non-matching surface meshes. Our methods minimize errors in a certain norm while achieving strict conservation. Some traditional methods for data transfer and related problems are also reviewed and compared with our methods. Numerical results demonstrate significant advantages of common-refinement based methods, especially for repeated transfers. While the comparisons are performed with matching geometries, this paper also addresses additional complexities associated with non-matching surface meshes and presents some experimental results from 3-D simulations using our methods. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: multiphysics simulations; data transfer; common refinement; interpolation; conservation; non-matching meshes

1. INTRODUCTION

Detailed and accurate simulations of complex systems frequently require integration of multiple physical disciplines, such as fluid dynamics, solid mechanics, and combustion, in an efficient and accurate manner. The multidisciplinary nature of such systems suggests a partitioned approach, where each discipline solves its own equations over its own geometric domain, and jump conditions are exchanged periodically at the interface between domains [1–3]. In a typical simulation, the surface meshes at the interface are non-matching, meaning that their combinatorial structures differ and their geometric realizations may have discrepancies due to discretization or rounding errors. Such non-matching meshes and associated data transfer

[*] Correspondence to: M. T. Heath, Computational Science and Engineering, 2270 Digital Computer Laboratory MC-278, University of Illinois at Urbana-Champaign, 1304 W. Springfield Ave., Urbana, IL 61801, U.S.A.

[†] E-mail: heath@uiuc.edu

[‡] E-mail: jiao@uiuc.edu

Contract/grant sponsor: Center for Simulation of Advanced Rockets funded by the U.S. Department of Energy; contract/grant number: B523819

Received 25 April 2003
Revised 9 January 2004
Accepted 18 May 2004

Copyright © 2004 John Wiley & Sons, Ltd.

problems also arise in other scientific applications, such as adaptive meshing, multigrid methods, texture mapping, and shape morphing.

This paper addresses the problem of transferring data between non-matching interface meshes, with the focus on repeated data transfer in time-dependent multiphysics simulations. In these applications, data transfer must be numerically accurate and physically conservative. Numerical accuracy means small errors in a specific norm. Conservation corresponds to the equilibrium at the interface, which for many applications is required by the physical laws of the underlying problem, such as balance of forces or conservation of mass. For applications involving repeated data transfer, achieving both accuracy *and* conservation is particularly important and at the same time very difficult, because errors may accumulate across iterations, and a scheme that is both accurate *and* conservative tends to introduce smaller errors and deliver better convergence than non-conservative or inaccurate ones.

1.1. Problem description

In finite element and finite volume methods for numerical computations, data associated with a mesh are typically specified at a finite number of discrete points, referred to as *control points.* In finite element methods, data values are usually specified at the nodes. Together with the *basis functions* (or *shape functions*, typically piecewise linear or quadratic) associated with the nodes, these nodal values determine a piecewise polynomial interpolant over the underlying space of the mesh. In finite volume methods, data values are specified either at nodes or at cell centres, and an interpolant is defined piecewise over the *control volumes* of the mesh that surround or are incident on the control points.

This paper considers the problem of passing information from the control points of one mesh (the *source* mesh) onto the control points of another mesh (the *target* mesh). It focuses on cases where the interpolant over the mesh is given by a linear combination of basis functions associated with the control points, whose coefficients are their corresponding data values. These interpolants include those used in finite element methods and some variants of finite volume methods (for example, a first-order cell-centred finite volume method, where the control points are cell centres and the basis functions are piecewise constant). Let m be the number of source control points and n be the number of target control points. Let $f = \sum_{i=1}^{m} f_i \phi_i$ denote the *source function*, where $f_i = f(\mathbf{s}_i)$ and ϕ_i are the value and basis function, respectively, associated with the ith source control point $\mathbf{s}_i$, and let $g = \sum_{i=1}^{n} g_i \psi_i$ be the *target function*, where $g_i = g(\mathbf{t}_i)$ and ψ_i are associated with the ith target control point $\mathbf{t}_i$, respectively. Technically, the data transfer problem considered in this paper is to determine the g_i given the ϕ_i, f_i and ψ_i. The *residual* of data transfer is $f - g$. A norm of the residual measures accuracy, and the integral of the residual measures conservation.

1.2. Common refinement

This paper presents accurate and conservative methods for data transfer and efficient techniques for implementing them. Our data transfer methods use a data structure called the *common refinement* of two non-matching meshes. A common refinement of two meshes is a mesh composed of elements that subdivide the elements of both input meshes simultaneously, or, simply put, the intersections of the elements of the input meshes. A common refinement defines, and allows efficient query of, a unique nearby corresponding point on one side for every point on the other. Furthermore, it enables accurate integration of functions that depend on the shape functions of

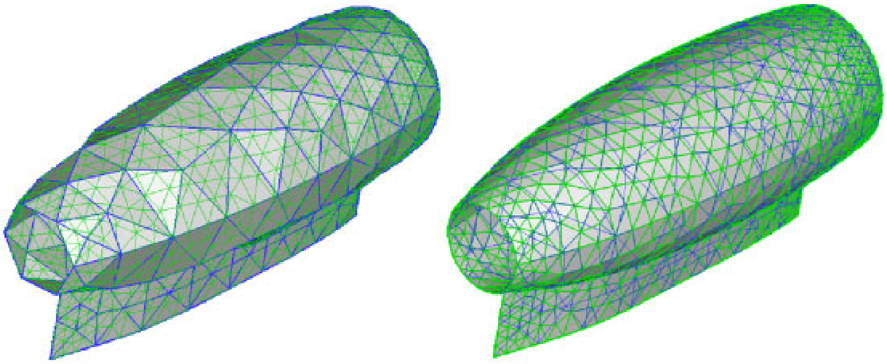

Figure 1. Common refinement of blue (coarse) and green (fine) meshes and its realizations on input surfaces.

the two meshes. For these reasons, the common refinement is important for our methods both theoretically and practically. An efficient and robust algorithm for constructing the common refinement for non-matching surface meshes is described in Reference [4]. Figure 1 shows an example of a common refinement of two triangular surface meshes modelling an engine nacelle of an aircraft. Readers are referred to Reference [4] for more details on the common refinement.

1.3. Organization

The remainder of the paper is organized as follows. Section 2 reviews some related problems and methods. Section 3 presents a simple weighted-residual formulation that minimizes the L_2 norm of the error and describes a common-refinement based discretization along with two *ad hoc* discretizations. Section 4 compares some interpolation techniques and discretized weighted-residual schemes and demonstrates the advantages of the common-refinement-based methods. Section 5 addresses issues arising from non-matching geometries of surface meshes, and presents experimental results with surface meshes. Section 6 analyses some weaknesses of the simple weighted-residual formulation and proposes a more advanced approach to overcome such weaknesses by minimizing Sobolev norms. Section 7 concludes the paper with a discussion of applications and future work.

2. RELATED PROBLEMS AND METHODS

This section reviews a few data transfer methods between meshes in multiphysics simulations and similar contexts. There is a vast amount of literature on this topic. Our objective is not to conduct an extensive survey as in References [3, 5], but to summarize previous methods that are most accurate, most widely used, or most relevant to our methods.

2.1. Pointwise interpolation and extrapolation

Interpolation (and sometimes extrapolation) is probably the most popular method for transferring data between meshes, and it is even sometimes used as a synonym for data transfer. In this paper, *interpolation*, or *pointwise interpolation*, refers to the process of determining and evaluating an interpolant to obtain values for some query points from given values at a finite number of control points, where the interpolant must evaluate to the given values at the control points. *Extrapolation* is required when a query point falls outside the range of the source mesh.

Pointwise interpolation can be categorized into two types. One type uses the source function as the interpolant, and is sometimes referred to as *consistent interpolation* [6] or *inverse isoparametric mapping* [3, 7]. Let $\mathbf{t} = \sum_i \phi_i(\mathbf{t})\mathbf{s}_i$ be a target control point, where the ϕ_i are the shape functions associated with the source mesh. Consistent interpolation assigns the value at $\mathbf{t}$ to be $g(\mathbf{t}) = \sum_i \phi_i(\mathbf{t}) f(\mathbf{s}_i)$. The other type constructs an interpolant using basis functions of order higher than that of the source basis functions, and will be referred to as *alternative interpolation*. A popular alternative interpolation is the thin-plate spline, which is obtained by minimizing the bending energy of a thin plate [5, 8] and has the form

$$H(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \|\mathbf{x} - \mathbf{x}_i\|^2 \ln \|\mathbf{x} - \mathbf{x}_i\|$$

where $\|\cdot\|$ denotes the Euclidean distance and the α_i are unknowns to be obtained by solving an $m \times m$ dense linear system. Utilizing no topological information, thin-plate splines are equally applicable to interpolation and extrapolation, similarly to a few other methods in that class (such as infinite-plate splines, finite-plate splines, and multiquadric-biharmonic [5, 8]). The cubic spline, obtained by enforcing continuity of second derivatives, is another widely used interpolant in geometric modelling and other scientific applications [9–12].

A major drawback of interpolation and extrapolation schemes is that they are not *strictly conservative*. When the shape functions are linear, the error in the conservation measure is of second order for consistent interpolation, but is in general non-zero and is sensitive to the second derivative of the input function, as shown in Appendix A. Alternative interpolation can deliver higher order accuracy but is still non-conservative. If the second derivative is large or does not exist, then the loss of information can be arbitrarily large, unless the extremes coincide with the nodes. For repeated transfers, accuracy and conservation measures may also degrade quickly over time, as shown in Section 4.

2.2. Area-weighted averaging

For cell-centred data, *area-weighted averaging* is the simplest form of conservative data transfer. In this method, the value of a target element is the weighted average of the values of the source elements in contact, where the weights are the areas of the intersections between the source and target elements. This method is sometimes called *conservative rezoning* (*remapping*) [13–15]. For the discretization of conservative rezoning, integrations are usually evaluated over the intersections of the source and target elements, or sometimes converted to boundary integration. This method is a special case of the common-refinement-based discretization of weighted-residual method that we describe later.

2.3. Mortar elements methods

Mortar element methods are general techniques for projecting data at interfaces between two or more subdomains in a non-conforming domain decomposition. In these methods, the discretized interfaces between subdomains are divided into the *mortar* side and the *non-mortar* side. Each side forms a discretization of the interfaces, and data are projected from the non-mortar side to the mortar side. The formulations of mortar element methods are also based on a weighted

Copyright © 2004 John Wiley & Sons, Ltd.

residual, where the weight functions are usually chosen from the space spanned by the basis functions of the mortar side. Most investigations of mortar element methods have focused on analyses of its *semidiscrete* form in the context of non-conforming domain decompositions [16–19]. Their discretizations face similar issues as the weighted-residual methods, which are the focus of Section 3.

2.4. Specialized methods

Pointwise interpolation and weighted residual are generic methods that are applicable to a wide range of problems. There are also methods designed for specific applications that do not fall directly into the above categories but frequently are variants or combinations of them. A particular example of specialized methods is load and motion transfer in fluid–solid interaction for aeroelasticity problems. In these problems, tractions from the fluid must be converted into nodal loads (equivalent to the load vectors in weighted residual methods) on the solid [6, 20]. Displacements of the solid are then transferred back to the fluid, typically using consistent interpolation. By taking advantage of knowledge about their specific applications, specialized methods can be more efficient than generic methods. However, they frequently must also face the same discretization issues as those of weighted residual methods and can also benefit from common-refinement-based discretizations.

3. L_2 MINIMIZATION

A simple weighted-residual formulation is to minimize the L_2 norm of the error. This section studies the properties of its semidiscrete form, assesses a couple of its unsuccessful discretizations, and presents a more accurate discretization based on the common refinement.

3.1. Semidiscrete formulation

Observe that the L_2 norm of $g-f$ is minimized if $\partial \int_\Omega (g-f)^2 \,\mathrm{d}x/\partial g_i = 0$ for $i=1,\dots,n$. Since

$$\frac{\partial \int_\Omega (g-f)^2 \,\mathrm{d}x}{\partial g_i} = \frac{\partial \int_\Omega (\sum_{j=1}^n \psi_j g_j)^2 \,\mathrm{d}x}{\partial g_i} - 2\,\frac{\partial \int_\Omega \sum_{j=1}^n \psi_j g_j f \,\mathrm{d}x}{\partial g_i} + \frac{\partial \int_\Omega f^2 \,\mathrm{d}x}{\partial g_i}$$

$$= 2\sum_{j=1}^n \int_\Omega \psi_i \psi_j \,\mathrm{d}x g_i - 2\int_\Omega \psi_i f \,\mathrm{d}x$$

this condition reduces to an $n \times n$ linear system $\mathbf{Mx} = \mathbf{b}$, where $\mathbf{x}$ is an unknown vector composed of the g_i, $\mathbf{M}$ is the *consistent mass matrix* (or simply *mass matrix*) with entries $\mathbf{M}_{ij} = \int_\Omega \psi_i \psi_j \,\mathrm{d}x$, and $\mathbf{b}$ is the *load vector* with components $\mathbf{b}_i = \int_\Omega \psi_i f \,\mathrm{d}x$. The preceding derivation is based on a minimum principle, or the Rayleigh–Ritz method, and this data transfer method will be referred to as L_2 *minimization*. The same set of equations can also be obtained based on the weak form, or the Galerkin method. Specifically, the equation $g = f$ is converted into a set of integral equations

$$\int_\Omega wg \,\mathrm{d}x = \int_\Omega wf \,\mathrm{d}x$$

where the w are *weight functions* (or *test functions*), chosen to be the target basis functions ψ_i. This derivation corresponds to an orthogonal projection of f onto the function space of the weight functions. The latter formulation is used in Reference [2]. Because of the equivalence of the two formulations, this method (along with many others) belongs to the *Rayleigh–Ritz–Galerkin* methods.

To investigate the accuracy of L_2 minimization, let $\tilde{g}$ denote the solution of L_2 minimization and $\hat{g}$ denote the solution from linear interpolation. From the Rayleigh–Ritz formulation, $\tilde{g}$ is optimal in the L_2 norm in the space spanned by the target shape functions, and hence $\|\tilde{g} - f\|_2 \leqslant \|\hat{g} - f\|_2$, which is $O(h_s^2 + h_t^2)$ from the analysis in Appendix A. Let F denote the analytical function that is discretized by the source function. If F'' is continuous and bounded, then $\|\tilde{g} - F\|_2 \leqslant \|\tilde{g} - f\|_2 + \|F - f\|_2$, which is also $O(h_s^2 + h_t^2)$. Both bounds are tight asymptotically, as can be verified by fixing one mesh while refining the other. A key advantage of L_2 minimization over pointwise interpolation is that it is conservative in the sense that $\int_\Omega \tilde{g}\, \mathrm{d}x = \int_\Omega f\, \mathrm{d}x$, because $\tilde{g} - f$ is orthogonal to any function v in the function space spanned by the ψ_i, i.e. $\int_\Omega (\tilde{g} - f) v\, \mathrm{d}x = 0$, and constant functions are in this space.

L_2 minimization has some nice properties. First, the mass matrix $\mathbf{M}$ is sparse and symmetric, which is obvious from the definition $\mathbf{M}_{ij} = \int_\Omega \psi_i \psi_j\, \mathrm{d}x$ and the near orthogonality among the ψ_i. $\mathbf{M}$ is also positive definite, because $\mathbf{x}^{\mathrm{T}}\mathbf{M}\mathbf{x} = \int_\Omega \tilde{g}^2\, \mathrm{d}x = \|\tilde{g}\|_2^2 \geqslant 0$, and $\|\tilde{g}\|_2^2 = 0$ if and only if $\tilde{g} = 0$. Second, the condition number of $\mathbf{M}$ is bounded by a constant that depends only on the mesh quality of the target mesh, so L_2 minimization is always well conditioned for well-shaped meshes, independent of the order of the shape functions or the number of elements. Because of these properties, the linear system $\mathbf{Mx} = \mathbf{b}$ can be solved efficiently using an iterative method. Furthermore, L_2 minimization has a stabilizing effect in the sense that $\|\tilde{g}\|_2 \leqslant \|f\|_2$ (i.e. it underestimates the 'energy' of the system), because $(\tilde{g} - f)^2 + \tilde{g}^2 = f^2 + 2(\tilde{g} - f)\tilde{g} = f^2$ (the Pythagorean theorem). Some of these properties are general to the Rayleigh–Ritz–Galerkin methods [21].

3.2. *Ad hoc discretizations*

To discretize L_2 minimization, the key question is how to integrate $\int_\Omega \psi_i \psi_j\, \mathrm{d}x$ and $\int_\Omega \psi_i f\, \mathrm{d}x$. A few discretization schemes have been used in the literature. However, these schemes lack mathematical foundation and undermine the accuracy and/or conservation properties of L_2 minimization.

3.2.1. Source-based discretization. In Reference [2], Cebral and Löhner computed $\int_\Omega \psi_i f\, \mathrm{d}x$ by numerical quadrature over elements of the *source* mesh and computed $\int_\Omega \psi_i \psi_j\, \mathrm{d}x$ over elements of the *target* mesh; see Reference [22, Chapter 8] for an introduction to numerical quadrature. The argument for this choice is that this discretization is conservative as in the semidiscrete form. Unfortunately, this scheme is sensitive to relative mesh resolution and is particularly poor if the source mesh is coarser than the target mesh. In general this scheme is zeroth-order accurate for a fixed relative mesh resolution: if the input meshes are refined simultaneously with a fixed ratio of mesh sizes, the error does not go to zero. As a consequence, this scheme cannot transfer even a constant function exactly, whereas the semidiscrete L_2 minimization is capable transferring linear functions exactly. For general functions, this discretization can cause high-frequency oscillations, as shown in Section 4. The reason for such oscillations is a violation of the regularity assumption of numerical quadrature: A quadrature rule in general assumes smoothness of the integrand function within an interval, and they can be arbitrarily

Copyright © 2004 John Wiley & Sons, Ltd.

inaccurate for non-smooth functions [22]. The target shape functions are piecewise polynomials within a source element for non-matching meshes, and their slopes become more and more steep as the meshes become more refined, which leads to unbounded errors in **b** for the source-based scheme.

3.2.2. Target-based discretization. Some older methods evaluate both $\int_\Omega \psi_i \psi_j \, dx$ and $\int_\Omega \psi_i f \, dx$ over elements of the *target* mesh [20]. Because f is piecewise polynomial within each target element, the regularity assumption is also violated in this discretization. However, if the exact function has continuous and small second derivative, then f is typically well behaved within a target element, and hence the discretization behaves reasonably well. In particular, it can transfer linear functions exactly, and is less oscillatory than source-based discretization for general functions. The drawback of the target-based discretization is that it is not strictly conservative, similar to pointwise interpolation.

3.3. Common-refinement-based discretization

The failures of the above two schemes naturally lead us to a common-refinement based discretization. Because the shape functions ϕ_i and ψ_i are polynomials within each source and target element, respectively, the functions $\psi_i f$ are polynomials within each subelement of a common refinement. Therefore, provided the degrees of the quadrature rules are sufficiently high, the load vector can be computed exactly by integrating $\psi_i f$ over the subelements. The mass matrix can also be computed exactly with numerical quadrature over target elements or subelements. For instance, if the source and target basis functions are linear, then the functions $\psi_i \psi_j$ and $\psi_i f$ are quadratic, and applying a quadrature rule of degree at least 2 integrates them exactly over the target elements and the subelements, respectively. With this discretization, L_2 minimization is truly conservative subject only to rounding errors, and is at least as accurate as consistent interpolation. When the shape functions are piecewise constant, this discretization of the weighted residual method reduces to area-weighted averaging.

One might ask whether exact integrations are necessary. Let k be the highest degree of a polynomial in the shape functions. In References [21, 23], it has been shown that for an elliptic PDE of order $2m$, a quadrature rule with degree at least $2(k-m)$ is generally required to avoid loss of convergence rate. The L_2 minimization is essentially a zeroth-order PDE, and hence in general it requires a quadrature rule of degree $2k$, which is sufficient to integrate the functions exactly. Besides potential loss of accuracy, an inexact quadrature rule may lead to ill-conditioned or even singular mass matrices; such an example is given in Appendix B. Therefore, exact integration is preferable for optimal accuracy and robustness.

4. NUMERICAL COMPARISON

This section compares the solutions of consistent interpolation, thin-plate splines, cubic splines, and three discretization schemes for L_2 minimization for repeated transfers, by investigating their solutions after transferring an analytic function back and forth between two meshes for a large number of iterations. While such a simple repeated transfer problem is unlikely to occur in practice, it serves as an effective 'microscope' for evaluating the methods, because weaknesses of the methods become magnified by the iterations and hence are easily identified. It also resembles the repeated exchange of jump conditions in coupled multiphysics simulations. Comparisons of conservative schemes with more realistic applications are described in Section 7.1 and in Reference [24], and those results are consistent with those obtained from the present tests.

4.1. Accuracy in one dimension

Our 1-D tests transfer Runge's function $1/(1+25x^2)$ [22, Chapter 7] back and forth between two uniform grids. This function is transferred from the coarse grid onto the fine grid, and then the solution is transferred back onto the coarse grid, which accounts for one iteration. The errors after power-of-two iterations are evaluated to investigate the growth of errors. The first test uses two grids with 32 and 45 nodes, and Figure 2 shows the results on the coarse grid after 2^i iterations, where i ranges up to 3 for the source-based scheme and up to 6 for other methods. Figure 3(a) plots the relative errors up to 2^{12} iterations, with the errors computed as $\|\mathbf{g}_{\text{numerical}} - \mathbf{g}_{\text{exact}}\|_2/\|\mathbf{g}_{\text{exact}}\|_2$, where $\mathbf{g}$ denotes a vector composed of values at the nodes of the coarse mesh. A second test uses two grids with 61 and 89 nodes, and Figure 3(b) plots the errors with these grids. For the discretizations of L_2 minimization, linear elements are used in all cases except for the 'common refinement 2' in Figure 3(b), which uses 30 and 44 quadratic elements on the two grids, respectively.

In our tests, cubic splines deliver the best results among the interpolation schemes, while thin-plate splines are nearly an order of magnitude less accurate than cubic splines, and consistent linear interpolation is an order of magnitude further less accurate. In a previous study Smith *et al.* [3], performed an extensive comparison of interpolation schemes commonly used in physical simulations, and consistent interpolation and thin-plate splines were among the best. Our results indicate that cubic splines are superior to the interpolation schemes that were considered in Reference [3] in terms of accuracy.

For L_2 minimization, the two-point Gaussian quadrature rule (of degree 3) is used over the element of the source, target, or common refinement for linear elements when integrating $\mathbf{M}_{ij}$ and $\mathbf{b}_i$, and the three-point Gaussian quadrature rule (of degree 5) is used for quadratic elements. In both cases, integrations are exact for $\mathbf{M}_{ij}$. Among the discretization schemes, the source-based introduced oscillations after one transfer, whereas the target-based introduced oscillations later; the oscillations grow quickly in both cases and cause the solutions to blow up. Note that grid refinement had no effect on the errors associated with the source-based scheme, confirming our claim that such a discretization is zeroth-order accurate.

The common-refinement based scheme, on the other hand, performed consistently well in both cases. It does not introduce any oscillation, as integrations are exact. The common-refinement scheme with linear shape functions is only second-order accurate, and hence it is expected to be less accurate than cubic splines for a single transfer. However, its associated errors grow much more slowly than other methods for repeated transfers, and it eventually ends up as accurate as cubic splines. The common-refinement scheme with quadratic elements is nearly as accurate as cubic splines. In terms of conservation, the error in the integrals is on the order of 10^{-12} for the common-refinement schemes compared with 10^{-2} using cubic splines after 2^{12} iterations. Therefore, the common-refinement based L_2 minimization improves accuracy and conservation substantially compared with traditional methods, especially for repeated data transfer.

4.2. Accuracy in two dimensions

The 2-D test transfers a function between two uniform Cartesian grids of 32×32 and 45×45 over the domain $[-3,3]\times[-3,3]$. This test used the peaks function in MATLAB [25], defined as

$$\text{peaks}(x,y) = 3(1-x)^2 \mathrm{e}^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right)\mathrm{e}^{-x^2-y^2} - \frac{1}{3}\mathrm{e}^{-(x+1)^2-y^2}$$

Copyright © 2004 John Wiley & Sons, Ltd.

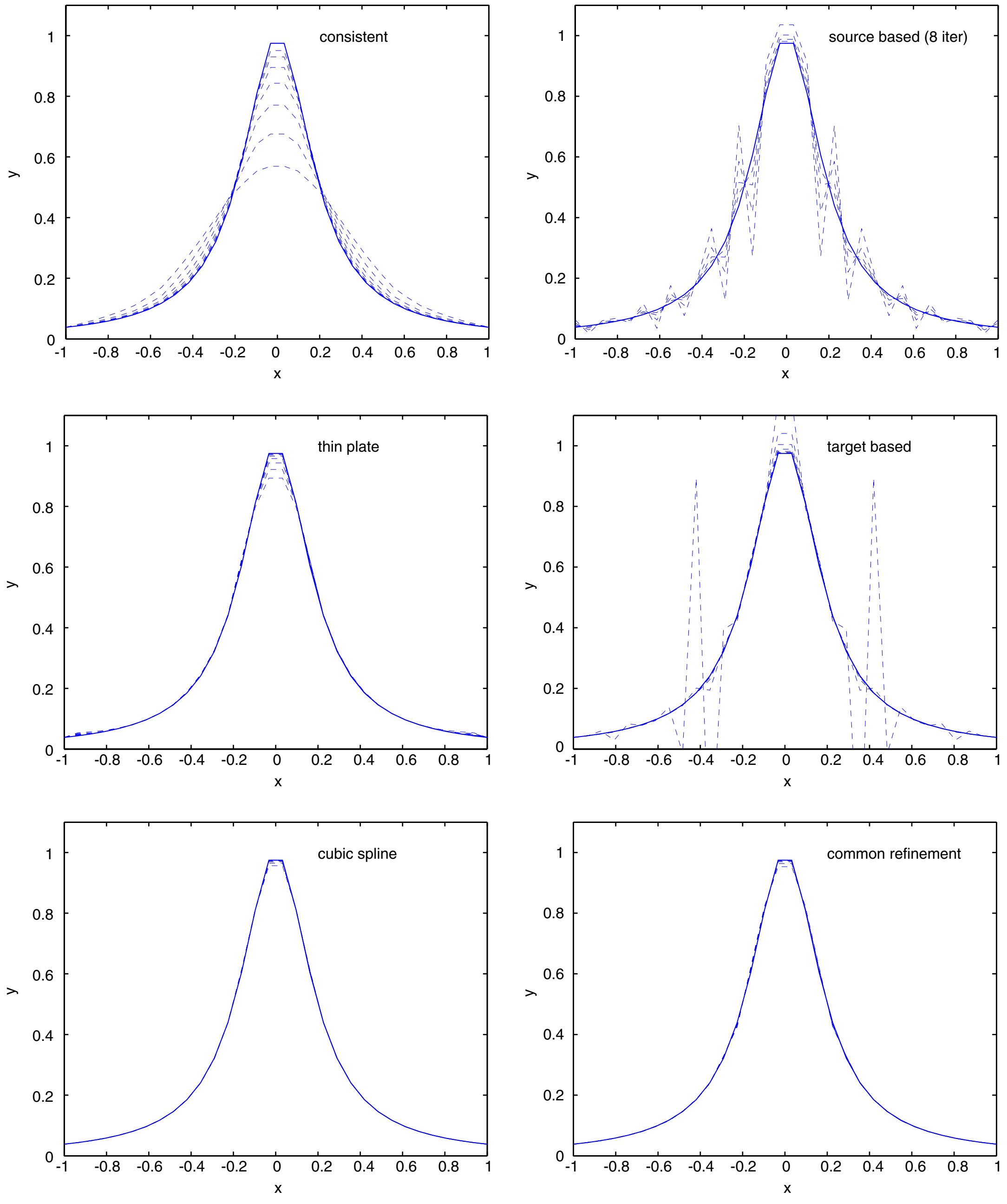

Figure 2. Solutions of repeated transfers of Runge's function up to 64 iterations.

Copyright © 2004 John Wiley & Sons, Ltd.

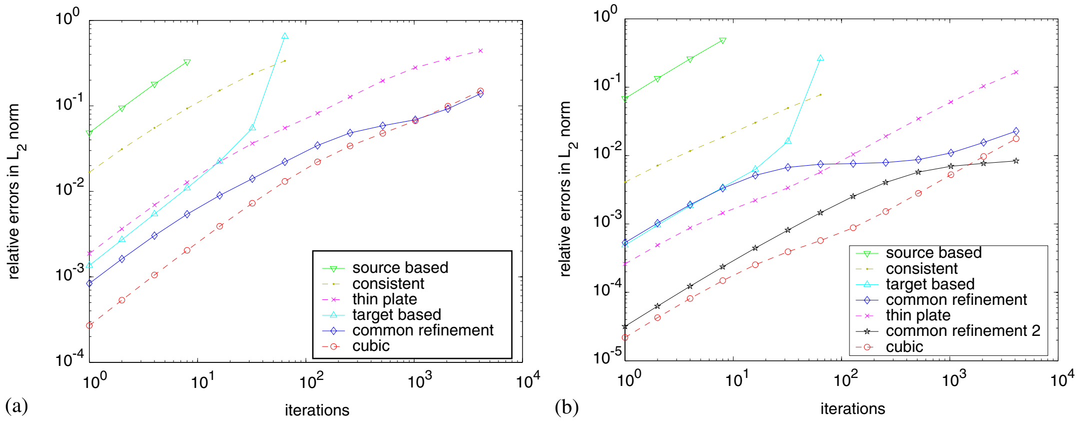

Figure 3. Relative errors in repeated transfer of Runge's function: (a) repeated transfer between 32 and 45 nodes; and (b) repeated transfer between 61 and 89 nodes.

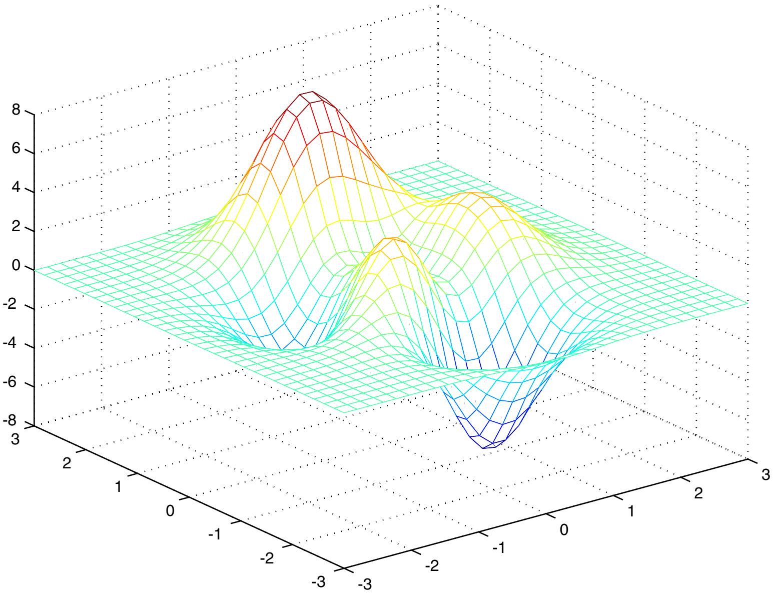

Figure 4. 3-D plot of peaks function.

which is obtained by translating and scaling Gaussian distributions. Figure 4 shows a 3-D plot of the function on the coarser grid. As for 1-D cases, this test also compares three interpolation schemes and three discretizations of L_2 minimization.

Figure 5 plots in logarithmic scales the relative errors of the resulting functions on the coarser mesh for up to 2^n iterations, where n is 9 for cubic splines and the common-refinement based scheme, 2 for the source-based, and 5 for the others. Smaller numbers of iterations were chosen

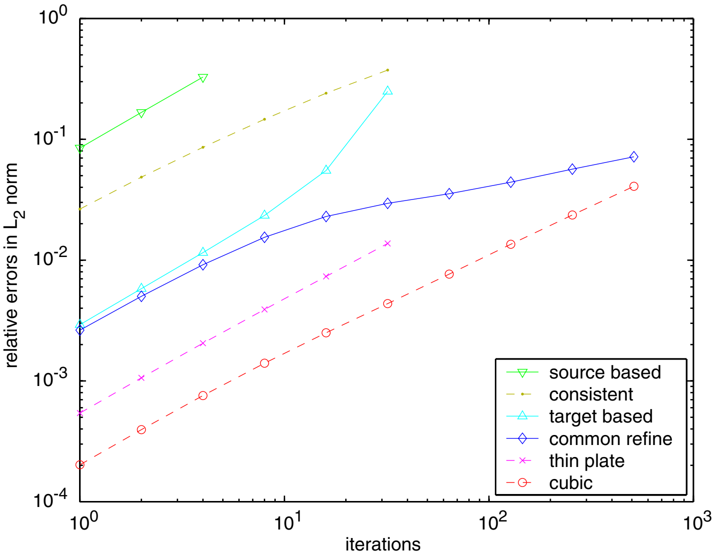

Figure 5. Relative errors in repeated transfer of peaks function.

when the errors became unreasonably large, or, in the case of thin-plate splines, the computations took too much time. Figure 6 shows the contours of the original function in solid lines and of the resulting functions after 2^i iterations in dashed lines, with i ranging up to 2 for the source-based and up to 5 for the other methods. Consistent with 1-D results, cubic splines deliver the best results among interpolation schemes, followed by thin-plate splines, which introduce some artefacts in the contours near boundaries. Among the minimization schemes, the source- and target-based introduce oscillations. The common-refinement-based scheme is not nearly as accurate as cubic splines initially but becomes comparable with cubic splines at the end. Using higher-order shape functions, the common-refinement scheme would deliver even better results, as already shown in one dimension.

4.3. Cost evaluation

Computation and storage costs are important considerations when choosing a numerical method. For large-scale simulations with potentially very large meshes, asymptotic performance is particularly important. Table I summarizes the asymptotic computational costs and required storage associated with data transfer. Among the methods, all except thin-plate splines require some sort of search operations, either to associate target nodes with source elements (for consistent or cubic splines), associate quadrature points of the target mesh with source elements (for the target based) or vice versa (for the source based), or ultimately to construct a common refinement of the meshes. In general, associating a point with an element of a mesh with n elements takes at least $O(\log n)$ time. However, associating all points would take $O(m+n)$ for well-shaped meshes by taking advantage of topological information and locality [4, 26].

After associating the meshes, consistent interpolation involves only evaluating some polynomials at each target point. Every other method requires evaluating some polynomials at either control or quadrature points to construct a linear system, and then solving the linear system.

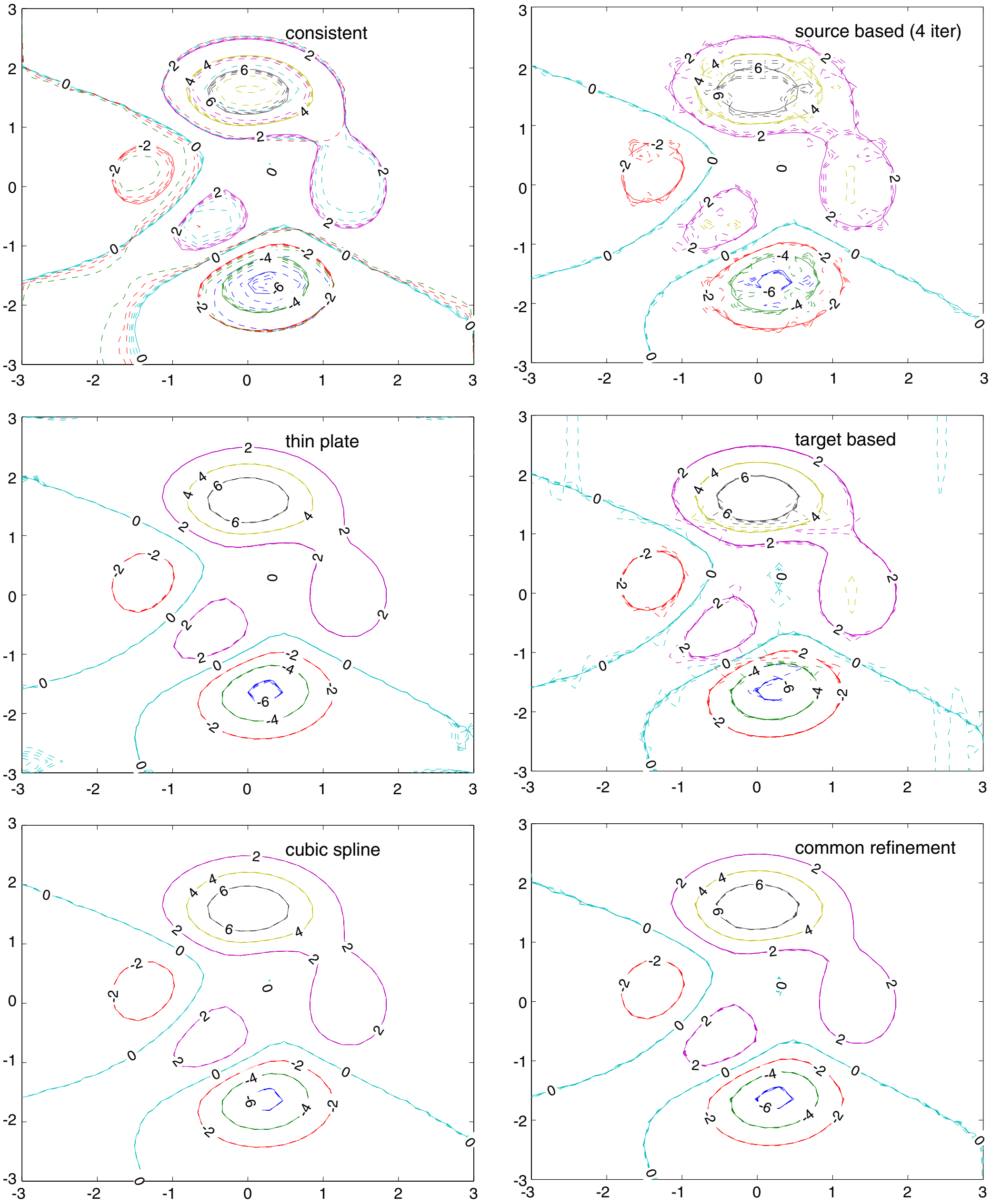

Figure 6. Contours of functions after 2^i iterations, where $i \leqslant 2$ for source-based and $\leqslant 5$ for others.

For L_2-minimization, the solutions of the linear systems are the values at the target control points, but for cubic and thin-plate splines, another round of evaluations of polynomials is necessary to obtain values at target points. The fourth column of Table I lists the total costs

Copyright © 2004 John Wiley & Sons, Ltd.

Table I. Asymptotic computation and storage costs of various methods. $O(S)$ indicates sparse solver.

	Search time		Solution time		
Method	Pointwise	Topological	Evaluation	Solve	Storage
Consistent	$O(\log m)$	$\approx O(m+n)$	$O(n)$	N/A	$O(m+n)$
Cubic	$O(\log m)$	$\approx O(m+n)$	$O(m+n)$	$O(S)$	$O(m+n)$
Thin plate	N/A	N/A	$O(m^2+mn)$	$O(m^3)$	$O(m^2+n)$
Source based	$O(\log n)$	$\approx O(m+n)$	$O(m+n)$	$O(S)$	$O(m+n)$
Target based	$O(\log m)$	$\approx O(m+n)$	$O(n)$	$O(S)$	$O(m+n)$
Common refinement	N/A	$\approx O(m+n)$	$\approx O(m+n)$	$O(S)$	$\approx O(m+n)$

Table II. Normalized execution times in one and two dimensions.

	Consistent	Cubic	Thin plate	Source	Target	Common refinement
1-D	1	5.88	13.87	6.60	6.65	7.12
2-D	1	4.3	1214.5	115.1	132.4	246.5

of evaluating polynomials before and after linear solves, and the fifth column lists the costs of linear solves. Because the thin-plate splines are global methods involving dense linear systems, the cost of evaluations is quadratic and that of linear solves is cubic in the input sizes. The linear systems in other methods are sparse, with the number of non-zero entries proportional to the input sizes, and hence they can be solved much more efficiently. The space complexity is quadratic for thin-plate splines but linear for all other methods.

Table II shows the execution times of the test cases in one and two dimensions, including both search and solution times, on a dual 2 GHz Power Mac G5 system. The calculations were carried out in MATLAB, which compiles a function M-file when the function is first executed [25]. To outweigh the overhead of runtime compilation, each type of transfer was run for a large number of iterations (2^{10} for 1-D and 2^5 for 2-D), and the execution times were normalized by dividing them by the time of consistent interpolation for each row. Consistent and cubic interpolations in 1-D are done using MATLAB's toolbox function interp1 with the methods '*linear' and '*spline', respectively, where the '$*$' indicates that the grid is equally spaced and monotonic for faster interpolation; the 2-D computations are done using interp2 with the corresponding methods. The codes for L_2 minimization and for thin-plate splines were written and manually optimized by the authors. For L_2 minimization, the mass matrices are stored using sparse-matrix format, and the majority of the execution time is spent on construction of the right-hand side of the system. For thin-plate splines, the code utilizes vector operations for polynomial evaluations and uses MATLAB's built-in dense linear solver, and the majority of the execution time is spent on the linear solver in two dimensions. Considering accuracy and efficiency, the cubic spline is the best choice for non-conservative transfer. Conservative transfer, in particular the common-refinement-based weighted-residual method, has the same order of time complexity as cubic interpolation but is two orders of magnitude more expensive than consistent interpolation and cubic splines. However, in the context of multiphysics simulations, the size of the interface is typically orders of magnitude smaller than the volume, and hence such an increase of a constant factor still leads to an insignificant increase of the overall cost, as observed from our experiences.

Copyright © 2004 John Wiley & Sons, Ltd.

5. TRANSFER BETWEEN SURFACE MESHES

The preceding discussions have assumed the geometries of the interfaces are perfectly matching. For curved interfaces, such as surface meshes, the underlying spaces of the meshes are no longer necessarily the same. In Reference [4], this issue was addressed in the context of constructing a common refinement for non-matching meshes by defining a continuous and bijective mapping between these surface meshes. In the common refinement, each subelement generally has two distinct realizations, which may have different areas and potentially different normal directions. In L_2 minimization, the functions $\psi_i f$ must be integrated over the subelements. The inconsistency in geometry of the input meshes raises the following question: Which realization should be used during integration? This question does not have a simple answer, unfortunately. To be more precise, in transferring a uniform function from one mesh onto another mesh, there are two questions with potentially contradictory answers:

1. Should the target function be uniform?
2. Should the integrals of the source and the target functions be equal?

These two questions correspond to the accuracy and conservation objectives of data transfer, respectively. The answers to these questions ideally should both be yes, and this can be achieved for meshes with matching geometry using weighted-residual methods. If the underlying spaces of the input meshes have different areas, however, then these two objectives in general contradict each other and must be adjusted to resolve this contradiction.

A few methods have been proposed in the literature to transfer data between meshes with non-matching geometry. In particular, a closest-point interpolation scheme is sometimes used, which interpolates the value for a target node from its closest point in the source mesh [27], and in the case of thin-plate splines, the problem would implicitly be treated as extrapolation. These schemes compromise conservation. The source-based conservative schemes, on the other hand, compromise accuracy and lead to oscillations, even with matching geometry. None of these approaches appears to be satisfactory. The problem ultimately boils down to the notion of conservation for non-matching meshes, which corresponds to equilibrium (or balance) at a well-characterized and physically unique interface. Interpolation schemes simply omit this equilibrium requirement. Existing conservative schemes, on the other hand, enforce equilibrium over two distinct surfaces, which is neither physically nor numerically justified. Furthermore, insisting on conservation but compromising accuracy is especially undesirable for general purpose data transfer methods because it may cause unexpected side effects. For example, if interface displacements are transferred from one mesh onto another, using such a scheme can cause the two meshes to separate because the one with smaller area will systematically move faster than the one with larger area.

To resolve the inconsistency, a better alternative is to introduce a unique *reference surface* between the input mesh surfaces, sometimes called a *virtual surface*. The source and target function are projected onto the reference surface, upon which conservation is imposed. This approach is more meaningful because equilibrium is now established over the reference surface, instead of two distinct surfaces. In practice, the reference surface can be chosen more or less freely. A rational choice, which is readily available to us through the common refinement, is a *convex combination* of the input meshes. Given input meshes S and T, a convex combination, denoted by $\alpha S + \beta T$, is defined as the point set

$$\alpha S + \beta T \equiv \{\mathbf{x} = \alpha \mathbf{s} + \beta T(\mathbf{s}) \mid \mathbf{s} \in S\}$$

Copyright © 2004 John Wiley & Sons, Ltd.

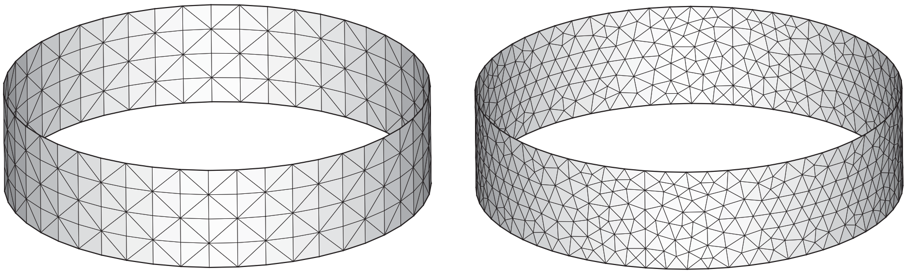

Figure 7. Two non-matching cylindrical surface meshes.

where α and β are two non-negative constants with $\alpha+\beta=1$, and $T(\mathbf{s})$ denotes the corresponding point of $\mathbf{s}$ inferred from the common refinement. Intuitively, if a line segment is drawn between every point $\mathbf{s}\in S$ and its corresponding point $T(\mathbf{s})$, then $\alpha S+\beta T$ contains a point from each of these line segments. Provided these line segments do not intersect, the resulting point set is a surface with no self-intersection. With this reference surface, L_2 minimization becomes minimization of

$$\int_{\alpha S+\beta T}(f-g)^2\,\mathrm{d}x$$

which is conservative in the sense that $\int_{\alpha S+\beta T} f\,\mathrm{d}x=\int_{\alpha S+\beta T} g\,\mathrm{d}x$.

To demonstrate the impact of non-matching geometry on numerical solutions, five tests were performed to transfer a sinusoidal function back and forth between two cylindrical surface meshes using five different schemes. Figure 7 shows the two meshes, one coarse and uniform while the other fine and non-uniform. One of our tests used consistent interpolation, which served as a reference for comparison. Another test used L_2 minimization without a reference surface, so that the mass matrix was computed using geometry of the target mesh while the load vector was computed using geometry of the source mesh. The remaining three tests used different combinations of reference surfaces: one used $\beta=1$ in both transfer directions, so the target mesh was the reference in either direction; another test used $\beta=0.5$ for both directions, so the average of the input surfaces was the reference; the fifth test used $\beta=0$ for forward direction and $\beta=1$ for backward direction, so that the uniform mesh was the reference. Figures 8(left) shows the errors in L_2 norm of the five schemes, where the curves for the three reference-surface schemes overlap with each other. Figures 8(right) shows the conservation measures evaluated on the uniform mesh. The common-refinement based schemes performed much better than consistent interpolation in both accuracy and conservation. Among the common-refinement based, the scheme without a reference surface achieved strict conservation but was not as accurate as those with a reference surface, and its order of accuracy depends on the mismatch of the geometry. When a reference surface is used, the accuracy did not appear to depend on the choice of the reference, but strict conservation is achieved only if there is a unique reference, on which the conservation measure should be computed.

6. SOBOLEV MINIMIZATION

L_2 minimization gives accurate and conservative solutions for many cases, but unfortunately it cannot always deliver satisfactory results. In particular, if the source function changes abruptly,

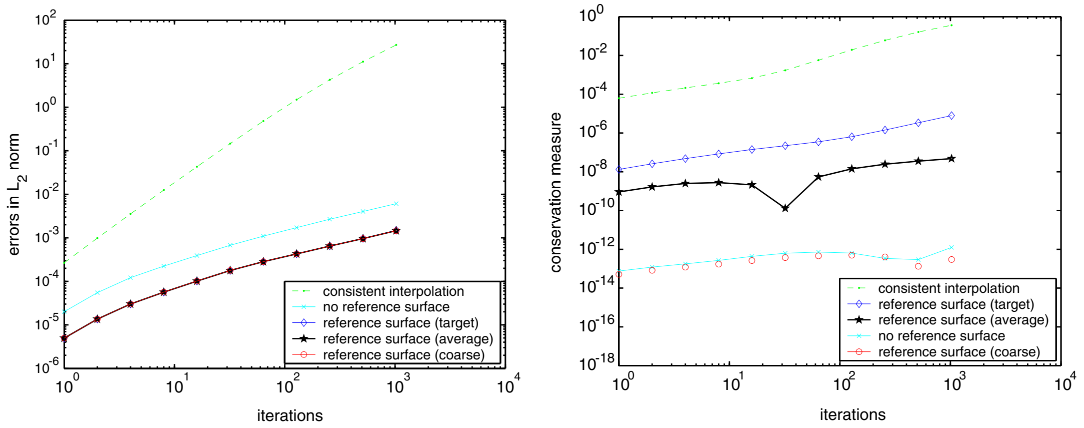

Figure 8. Errors in L_2 norm and conservation measures in transfer between surface meshes.

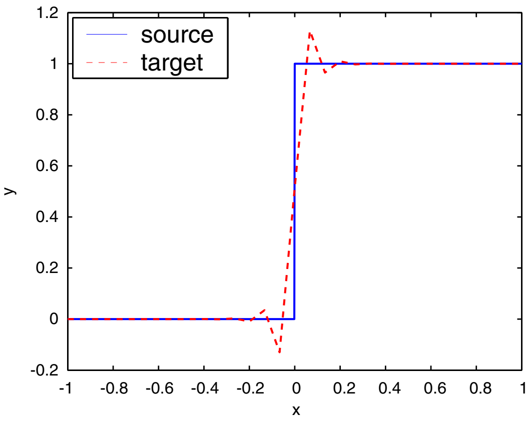

Figure 9. Gibbs phenomenon in minimizing L_2 norm.

this method may cause overshoots and undershoots, a behaviour similar to the Gibbs phenomenon seen in Fourier analysis. Figure 9 shows an example that passes a step function

$$f(x) = \begin{cases} 1 & \text{if } x \in [0, 1] \\ 0 & \text{if } x \in [-1, 0) \end{cases}$$

from a cell-centred grid onto a node-centred grid with 101 nodes. There are oscillations near the jump with an error of about 13%. Similar behaviour can also occur even when transferring data between two node-centred grids, especially when transferring from a fine grid onto a coarse one. These oscillations can be disruptive for some applications, where a solution as in Figure 10 is preferable.

Copyright © 2004 John Wiley & Sons, Ltd.

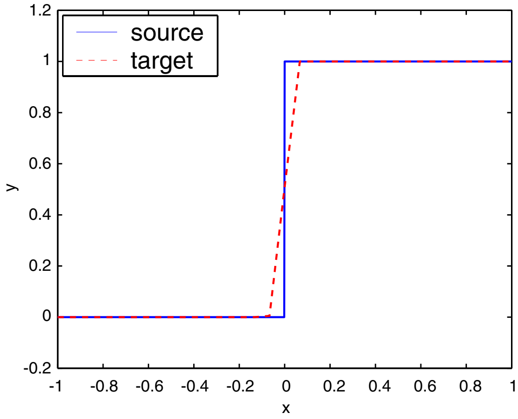

Figure 10. Resolution of Gibbs phenomenon.

A few approaches are feasible to resolve these overshoots and undershoots. For simple applications where jumps are isolated at well-characterized geometric features such as sharp edges of a surface, the simplest and most effective approach is to split the interface into patches along these features and to transfer data separately for each patch. Another approach is still to solve for the whole interface but introduce non-linear limiters to the linear solvers (such as in Reference [2]) or perform a postprocessing step to filter out oscillations after the solver (for example using similar schemes as in Reference [28]). This section pursues a different avenue by altering the objective function through minimizing a different norm, namely the Sobolev norm, to damp out the oscillations.

Let k be a constant integer. Given a function f whose derivatives of all orders less than or equal to k exist almost everywhere, the Sobolev norm of f is

$$\|f\|_{H^k} = \sqrt{\sum_{i=0}^{k} \alpha_i \|f^{(i)}\|_2^2}$$

where $f^{(i)}$ denotes the ith derivative of f, and α_i is its associated weight and is in general non-negative. The L_2 norm is a special case of the Sobolev norm with $k=0$. Intuitively, if the derivatives of the source function are known, a data transfer method can take advantage of them to obtain a more meaningful solution, and this section exploits this idea. For simplicity, this paper considers only the case $k=1$ and $\alpha_0=1$, i.e. minimizing

$$\|g-f\|_{H^1}^2 = \|g-f\|_2^2 + \alpha\|g'-f'\|_2^2$$

This method is referred to as *Sobolev minimization*. The first derivative of $f=\sum_i \phi_i f_i$ can be computed as $f'=\sum_i \phi_i' f_i$, obtained by differentiating f element-by-element, or as $f'=\sum_i \phi_i f_i'$, where f_i' is some derivative value at the node i supplied as input or computed using a certain approximation. The first approximation is referred to as *consistent differentiation*, and the second as *alternative differentiation*. Alternative differentiation can potentially deliver higher order of accuracy than consistent differentiation.

Assume that consistent differentiation is used for g. The Sobolev norm is minimized if $\partial \int_\Omega (g-f)^2 + \alpha\left(g'-f'\right)^2 \mathrm{d}x/\partial g_i = 0$ for $i = 1, \ldots, n$. Now

$$\frac{\partial \int_\Omega (g-f)^2 + \alpha(g'-f')^2\, \mathrm{d}x}{\partial g_i} = \frac{\partial \int_\Omega \left(\sum_{j=1}^n \psi_j g_j - f\right)^2 + \alpha\left(\sum_{j=1}^n \psi_j g'_j - f'\right)^2 \mathrm{d}x}{\partial g_i}$$

$$= 2\sum_{j=1}^n g_i \int_\Omega \psi_i\psi_j + \alpha\psi'_i\psi'_j\, \mathrm{d}x - 2\int_\Omega \psi_i f + \alpha\psi'_i f'\, \mathrm{d}x$$

which again leads to an $n \times n$ linear system $\mathbf{Sx} = \mathbf{b}$, where $\mathbf{S} = \mathbf{M} + \alpha\mathbf{K}$ is a weighted sum of the mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{K}$, with entries $\mathbf{K}_{ij} = \int_\Omega \psi'_i\psi'_j\, \mathrm{d}x$, and $\mathbf{b}$ is the modified load vector, with components $\mathbf{b}_i = \int_\Omega \psi_i f + \alpha\psi'_i f'\, \mathrm{d}x$. As with L_2 minimization, Sobolev minimization is also conservative, because $\int_\Omega (g-f)v + (g'-f')v'\, \mathrm{d}x = 0$ for any function v in the function space spanned by the ψ_i, and constant functions are in this space. Note that this argument holds independently of how f' is computed.

Sobolev minimization shares some properties with L_2 minimization: The matrix $\mathbf{S}$ is sparse and symmetric positive definite, and this scheme stabilizes the solution in that $\|g\|_{H^1} \leqslant \|f\|_{H^1}$. However, because the stiffness matrix $\mathbf{K}$ is singular, the matrix $\mathbf{S}$ is not necessarily well conditioned for large α. In general, α should be of order $O(h_t^2)$, as a larger α would lead to ill-conditioned systems as $h_t \to 0$. Furthermore, $\|f-g\|_{H^1}^2$ is proportional to $O(\alpha h_t^2 + h_t^4 + h_s^4)$, and an α of order $O(h_t^2)$ ensures the same order of accuracy as L_2 minimization. To discretize this method, a common-refinement-based scheme needs to be used and exact integration is preferred for optimal accuracy. When computing $\mathbf{S}_{ij}$, because the degree of the polynomials $\psi'_i\psi'_j$ is lower than that of $\psi_i\psi_j$, integrating to compute $\mathbf{K}_{ij}$ requires a smaller number of quadrature points than for computing $\mathbf{M}_{ij}$.

6.1. Smoothing effect

Sobolev minimization can effectively smooth out overshoots and undershoots, because such oscillations tend to have large derivatives, and Sobolev minimization inhibits unwanted oscillations by biasing the target function toward some prescribed derivatives. Figure 10 shows the resolution of the Gibbs phenomenon of the example in Figure 9 using Sobolev minimization, where $\alpha = h_t^2/6$, and f' is prescribed to be 0.

Sobolev minimization also has advantages in transferring data from a coarse grid onto a fine grid. Consider transferring a piecewise linear representation of a smooth function from a coarse grid with k intervals onto a fine grid with ck intervals, where c is a positive integer. When minimizing the L_2 norm, the target function would be identical to the source function, because the source basis functions are in the space of the target basis functions. By specifying continuously varying f', Sobolev minimization can construct a smoother representation on the target mesh. To illustrate this effect, a test is performed to transfer Runge's function from a grid with 11 nodes onto another grid with 61 nodes, both with linear shape functions. Figure 11(a) shows the target function after L_2 minimization, which is identical to the source function. Figure 11(b) shows the solution of Sobolev minimization, where $\alpha = 10h^2$ and centred differences (except for boundary nodes) are used to compute the derivatives. Both solutions are conservative and second-order accurate, but the function in Figure 11(b) is much smoother.

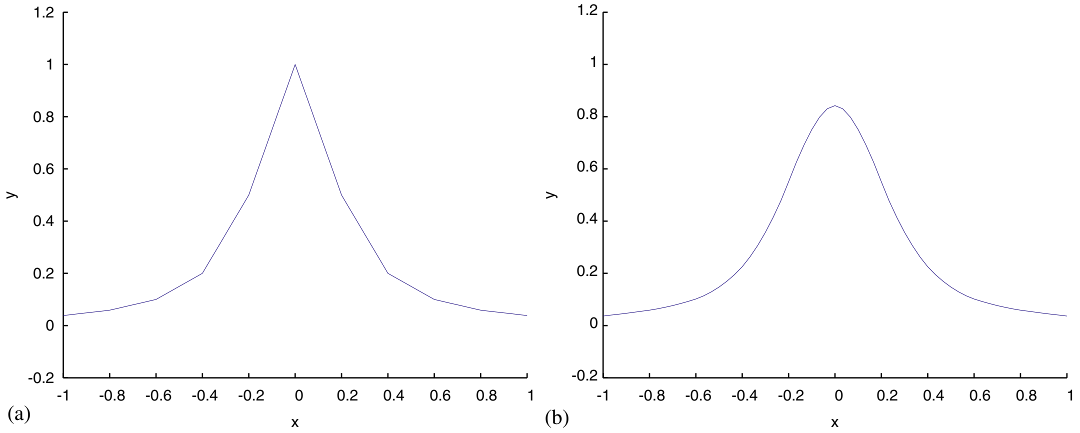

Figure 11. Smoothing effect of minimizing Sobolev norm: (a) minimizing L_2 norm; and (b) minimizing Sobolev norm.

6.2. Gradient on surface

To apply Sobolev minimization to surface meshes, the gradients of the source and target functions must be evaluated on a surface. For consistent differentiation, the gradient of a function f is $\nabla f = \sum_i \nabla\phi_i f_i$, where

$$\nabla\phi_i = \left(\frac{\partial\phi_i}{\partial x}, \frac{\partial\phi_i}{\partial y}, \frac{\partial\phi_i}{\partial z}\right)^{\mathrm{T}}$$

Within each element, ϕ_i is locally parameterized by two local (or *natural*) coordinates (ξ, η). Applying the chain rule leads to an underdetermined system $\mathbf{J}\nabla\phi_i = \mathbf{d}$, where J is the Jacobian matrix

$$\mathbf{J} = \begin{pmatrix} \dfrac{\partial x}{\partial\xi} & \dfrac{\partial y}{\partial\xi} & \dfrac{\partial z}{\partial\xi} \\ \dfrac{\partial x}{\partial\eta} & \dfrac{\partial y}{\partial\eta} & \dfrac{\partial z}{\partial\eta} \end{pmatrix}$$

and

$$\mathbf{d} = \left(\frac{\partial\phi_i}{\partial\xi}, \frac{\partial\phi_i}{\partial\eta}\right)^{\mathrm{T}}$$

The gradient $\nabla\phi_i$ can be solved as $\mathbf{J}^+\mathbf{d}$, where $\mathbf{J}^+$ is the pseudoinverse of $\mathbf{J}$ computed as $\mathbf{J}^{\mathrm{T}}(\mathbf{J}\mathbf{J}^{\mathrm{T}})^{-1}$ [22, Chapter 3]. This solution is consistent with that in $\mathbb{R}^2$ if $\partial z/\partial\xi = \partial z/\partial\eta = 0$, and it preserves the conservation property of Sobolev minimization as $\sum_i \nabla\phi_i = \mathbf{0}$. When computing $\nabla\phi_i$ for a subelement of the common refinement over a virtual surface $\alpha S + \beta T$, the matrix $\mathbf{J}$ is a linear combination of the Jacobian matrices of S and T, and $\mathbf{d}$ depends only on the source elements. All derivatives then can (and should) be computed over the source and target elements, and the subelements are used only for integration, not differentiation, so the condition number of the linear system does not rely on the mesh quality of the common refinement but

Copyright © 2004 John Wiley & Sons, Ltd.

only on those of the input meshes. This fact is worth noting because the common refinement can have arbitrarily narrow elements, which could otherwise lead to ill-conditioned systems.

6.3. Diagonal dominance of linear systems

By taking advantage of the free parameter α in the Sobolev norm, the structure of the linear system can be altered, so that its solution can be obtained more efficiently. For 1-D with linear basis functions, each element mass matrix $\mathbf{M}$ or element stiffness matrix $\mathbf{K}$ is a 2×2 symmetric matrix, and $\mathbf{M} + \alpha\mathbf{K}$ is a diagonal matrix if $\alpha = h_t^2/6$. This particular choice of α leads to an explicit data transfer algorithm that inverts only a diagonal matrix. Unfortunately, this diagonalization technique does not generalize to higher dimensions, because each element matrix then has at least three distinct off-diagonal entries but there is only one free parameter. Nevertheless, α can be chosen to make the matrix as diagonally dominant as possible, which tends to improve the convergence rate of iterative solvers. In our numerical experiments, the conjugate gradient method with Jacobi preconditioner frequently converges twice as fast for Sobolev minimization with this optimization compared with L_2 minimization.

7. DISCUSSION

This paper has analysed a number of numerical schemes for transferring data between non-matching meshes. Among the conventional methods, high-order interpolation schemes are accurate but not strictly conservative, although they are consistently conservative for smooth functions. The L_2 and Sobolev minimizations achieve both accuracy and conservation, and their common-refinement-based discretization are advantageous compared with other conventional methods for repeated data transfers. The conservative schemes for surface meshes presented in this paper are implemented in *Rocface*, the interface code for 3-D multiphysics simulations at the Center for Simulation of Advanced Rockets at the University of Illinois at Urbana-Champaign.

7.1. Applications

To demonstrate the effectiveness of the common-refinement based methods in multiphysics simulations, a fluid–solid interaction problem has been investigated to compare these methods with a popular load and motion transfer scheme in Reference [6]. The essence of this load projection scheme is to obtain the forces (loads) at target nodes as $G_j = \sum_i \psi_j(\mathbf{s}_i)F_i$, where the $F_i = \int_\Omega f\phi_i \, dx$ are the nodal loads at the source nodes $\mathbf{s}_i$, and the ψ_j are the target shape functions. This scheme is conservative in the sense that $\sum F_i = \sum G_j$. The motion transfer is done using consistent interpolation. In Reference [6], it was shown that this load and motion transfer scheme conserves virtual work. As shown in Reference [29] and in Appendix C, virtual work is also conserved by using L_2 minimization for both displacements and tractions under certain conditions.

Our experimental comparison uses a problem called *burning cavity*, where the solid is a ball with a spherical cavity in the middle filled with gas (the fluid). The fluid applies uniform pressure over the interface, which is a surface burning at a uniform speed, so that the solid deforms and regresses uniformly. The fluid and solid are solved using *Rocflo* and *Rocfrac*, respectively, both developed at the Center for Simulation of Advanced Rockets [30]. A tetrahedral mesh is used in the solid, whose interface mesh is triangular, and a block structured mesh is used in the fluid, whose interface mesh is quadrilateral. In one experiment, tractions (forces) and

Copyright © 2004 John Wiley & Sons, Ltd.

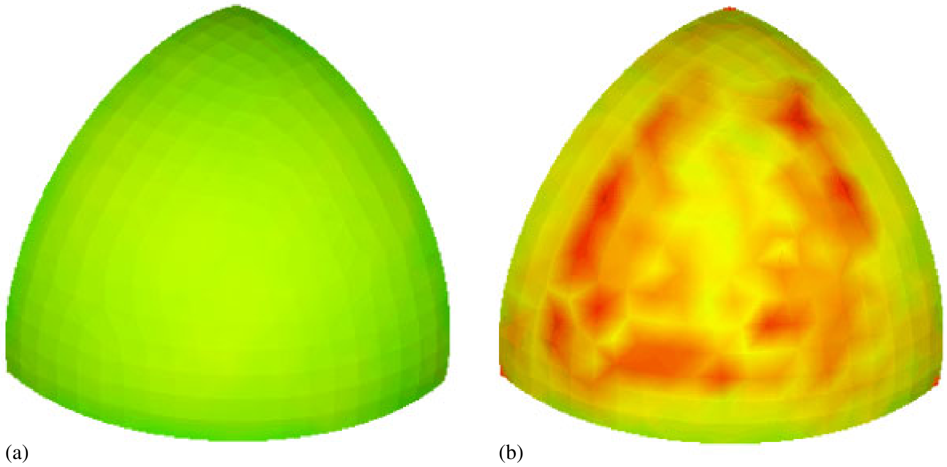

Figure 12. Numerical results of displacements under uniform pressure and regression in fluid–solid interaction: (a) common-refinement based L_2 minimization; and (b) conservative load and motion transfer in Reference [6].

regression are transferred from the fluid to the solid and displacements are transferred from the solid to the fluid, all using L_2 minimization with the common-refinement discretization. Different components of tractions and velocities are transferred independently of each other. In another experiment, the load transfer scheme in Reference [6] is used for tractions and consistent interpolations are used for velocities. Figure 12 shows the displacement after 500 iterations, where a more uniform result corresponds to smaller errors. The error on the left, which uses our common-refinement-based approach, is less than one percent compared against the analytical solution, whereas the error on the right is about twenty percent, demonstrating significant advantages of using common-refinement based approaches in multiphysics simulations. The errors are large in the latter method because its load transfer scheme is similar in spirit to the source-based discretization for computing load vectors, except that the source nodes are used instead of quadrature points. Another way to interpret the load transfer scheme in Reference [6] is to view it as the source-based scheme applied to the dual of the source mesh. In coupled simulations, the errors and oscillations associated with a source-based scheme worsen over time as in our simple test of repeated transfers. A more extensive comparison of conservative load transfer schemes has been conducted for two-dimensional fluid–solid interactions and will be published in another paper [24].

7.2. Future directions

The common-refinement-based methods developed in this paper allow accurate and conservative data transfer in numerical simulations. Among the methods, Sobolev minimization allows better flexibility and also delivers smoother results than L_2 minimization. This paper investigated adjusting the free parameter in Sobolev minimization to improve the efficiency of its solution. Monotonicity preservation is another important objective for many physical applications. For low-order conservative schemes, area-weighted averaging preserves monotonicity. For higher-order schemes, Sobolev minimization appears promising for preserving monotonicity,

Copyright © 2004 John Wiley & Sons, Ltd.

but selecting optimal values for the derivatives and control parameter is still a delicate issue that warrants further investigation. Besides Sobolev norms, some other norms are also feasible for minimization, such as the **A**-energy norm of a vector function $\|\mathbf{f}\|_{\mathbf{A}} \equiv \sqrt{\int_\Omega \mathbf{f}^{\mathrm{T}}\mathbf{A}\mathbf{f}\,\mathrm{d}x}$, but the linear systems for such a norm may be more expensive to solve for vector quantities (such as tractions), because all components can be interdependent and must be solved for simultaneously. A limitation of the common-refinement-based approach is that the geometry of the source and target meshes in principle should discretize the same geometry, and some applications may involve partially overlapping meshes [31–33] or physically disparate surfaces [27]. For better generality from a practical point of view, *Rocface* will be extended to provide support in such situations by gracefully resorting to less accurate or non-conservative schemes when the common refinement is not applicable.

APPENDIX A: ERROR BOUNDS FOR LINEAR INTERPOLATION

This section analyses the accuracy and conservation of pointwise interpolation with the focus on one-dimensional consistent interpolation with linear shape functions. Let h_s and h_t denote the mesh sizes of the source and the target meshes, respectively. Let the piecewise source function f be an approximation to an 'exact' function F, which is the limit of f as h_s tends to 0 and is typically unknown. For the sake of argument, assume that F has continuous and bounded second derivative F'', and the source function f agrees with F at the source control points. Let $\hat{g}$ denote the solution for g obtained from consistent interpolation, i.e. $\hat{g} = \sum \hat{g}(\mathbf{t}_i)\psi_i$, where $\hat{g}(\mathbf{t}_i) = f(\mathbf{t}_i)$. Although one can analyse the error in $\hat{g} - f$ directly [34], it is more convenient to start by analysing $\hat{g} - F$.

Assuming F'' is continuous and bounded, the following bounds on $f - F$ are shown in Section 1.6 of Reference [21]:

$$\|f - F\|_\infty \leqslant \tfrac{1}{8} h_s^2 \|F''\|_\infty \tag{A1}$$

$$\|f - F\|_2 \leqslant \frac{1}{\pi^2} h_s^2 \|F''\|_2 \tag{A2}$$

Using an argument similar to that in Reference [21], the following bounds on $\hat{g} - F$ can be obtained.

Lemma A.1
If F'' is continuous and bounded, then

(a) $\|\hat{g} - F\|_\infty \leqslant \frac{1}{8}(h_s^2 + h_t^2)\|F''\|_\infty$;
(b) $\|\hat{g} - F\|_2 \leqslant 1/\pi^2 h_t^2 \|F''\|_2 + \|1\|_2/8 h_s^2 \|F''\|_\infty$, where $\|1\|_2 \equiv \int_\Omega 1\,\mathrm{d}x$.

Proof
(a) Let $\Delta(x)$ be shorthand for $F(x) - \hat{g}(x)$, and $[a, b]$ be a typical interval of the target mesh. Now consider $\Delta(x)$ over $[a, b]$. Note that $\Delta(x)$ reaches its maximum at either $x = a$, $x = b$, or $x = c$ where $\Delta'(c) = 0$. For the former two cases, $\hat{g}(a) = f(a)$ and $\hat{g}(b) = f(b)$, and hence the error is bounded by $\frac{1}{8} h_s^2 \|F''\|_\infty \leqslant \frac{1}{8}(h_s^2 + h_t^2)\|F''\|_\infty$. For the latter, suppose $c - a \leqslant h_t/2 \leqslant b - c$. Expanding Δ in a Taylor series about a,

$$\Delta(a) = \Delta(c) + (c - a)\Delta'(c) + \tfrac{1}{2}(c - a)^2 \Delta''(w)$$

where $a < w < c$. Since $\Delta(a) \leqslant \frac{1}{8} h_s^2 \|F''\|_\infty$, $\Delta'(c) = 0$, and $\Delta''(x) = F''(x) - g_I''(x) = F''(x)$, this then leads to

$$|\Delta(c)| \leqslant \tfrac{1}{2}(c-a)^2 \Delta''(w) + \tfrac{1}{8} h_s^2 \|F''\|_\infty \leqslant \tfrac{1}{8} (h_s^2 + h_t^2) \|F''\|_\infty$$

(b) Consider the function

$$\Xi(x) \equiv \Delta(x) - \frac{x-a}{b-a} \Delta(b) - \frac{b-x}{b-a} \Delta(a)$$

which vanishes at both a and b. For simplicity, suppose $a = 0$ and $b = h_t$, and $\Xi(x)$ can be represented as a Fourier sine series:

$$\Xi(x) = \sum_{n=1}^{\infty} a_n \sin \frac{n\pi x}{h_t}$$

By direct computation,

$$\int_0^{h_t} \Xi^2 \, \mathrm{d}x = \frac{h_t}{2} \sum a_n^2$$

$$\int_0^{h_t} (\Xi'')^2 \, \mathrm{d}x = \frac{h_t}{2} \sum \left(\frac{n\pi}{h_t}\right)^4 a_n^2$$

Since

$$a_n^2 \leqslant \frac{h_t^4}{\pi^4} \left(\frac{n\pi}{h_t}\right)^4 a_n^2$$

for $n \geqslant 1$, summing on n,

$$\int_0^{h_t} \Xi^2 \, \mathrm{d}x \leqslant \frac{h_t^4}{\pi^4} \int_0^{h_t} (\Xi'')^2 \, \mathrm{d}x = \frac{h_t^4}{\pi^4} \int_0^{h_t} (F'')^2 \, \mathrm{d}x$$

This inequality holds for other intervals as well. Summing over all the intervals leads to

$$\int_\Omega \Xi^2 \, \mathrm{d}x \leqslant \frac{h_t^4}{\pi^4} \int_\Omega (F'')^2 \, \mathrm{d}x$$

Because

$$\left| \frac{x-a}{b-a} \Delta(b) + \frac{b-x}{b-a} \Delta(a) \right| \leqslant \max\{|\Delta(b)|, |\Delta(a)|\} \leqslant \|f - F\|_\infty$$

$$\|\Delta\|_2 \leqslant \|\Xi\|_2 + \left\| \left| \frac{x-a}{b-a} \Delta(b) + \frac{b-x}{b-a} \Delta(a) \right| \right\|_2 \leqslant \frac{1}{\pi^2} h_t^2 \|F''\|_2 + \frac{\|1\|_2}{8} h_s^2 \|F''\|_\infty \qquad \square$$

Now, it is then obvious that

$$\|\hat{g} - f\|_\infty \leqslant \left(\tfrac{1}{4}\,h_s^2 + \tfrac{1}{8}h_t^2\right)\|F''\|_\infty = O(h_s^2 + h_t^2) \tag{A3}$$

$$\|\hat{g} - f\|_2 \leqslant \frac{1}{\pi^2}(h_s^2 + h_t^2)\|F''\|_2 + \frac{\|1\|_2}{8}\,h_s^2\|F''\|_\infty = O(h_s^2 + h_t^2) \tag{A4}$$

These bounds are tight asymptotically, because if one of the meshes is fixed while the other is infinitely refined, the error is bounded by that of the fixed mesh, which is of second order. The error in the conservation measure is

$$\int_\Omega (\hat{g} - f)\,\mathrm{d}x \leqslant \|\hat{g} - f\|_1 \leqslant \|1\|_2\|\hat{g} - f\|_2 = O(h_s^2 + h_t^2) \tag{A5}$$

which is also of second order. Therefore, linear interpolation is *consistently conservative*, meaning that the error in the conservation measure goes to zero as h_s and h_t tend to zero. For alternative interpolation schemes, the error terms involving h_s are of higher order, but that involving h_t will stay the same. If F has discontinuities, the above analyses break down, and the accuracy will in general be of lower order, unless the discontinuities coincide with the source and target nodes.

APPENDIX B: SINGULARITIES DUE TO INEXACT QUADRATURE

Suppose the target mesh is a uniform 1-D grid. If the two-point Gaussian quadrature rule (of degree 3) is used, the element mass matrix is computed exactly

$$\frac{h}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

where h is the interval length of the target grid. If the one-point Gaussian quadrature rule (of degree 1) is used, the element mass matrix becomes

$$\frac{h}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

which is singular. The global mass matrix assembled from these 2×2 element matrices is also singular.

Singular matrices can also occur when using the two-point rule for 1-D grids with quadratic basis functions or when using the one-point rule for 2-D triangular meshes.

APPENDIX C: CONSERVATION OF VIRTUAL WORK

In fluid–solid interaction, displacement u is transferred from solid to fluid, and pressure p is transferred from fluid to solid. Provided that the basis functions for u and p are the same, conservation of virtual work is achieved if L_2 minimization is used to transfer u and p. Note

Copyright © 2004 John Wiley & Sons, Ltd.

that load (traction) transfer satisfies

$$\int_\Omega p^{(s)} \phi_i^{(s)} \, \mathrm{d}x = \int_\Omega p^{(f)} \phi_i^{(s)} \, \mathrm{d}x$$

where $f^{(s)}$ and $f^{(f)}$ denote a function f on the solid and fluid meshes, respectively. Then

$$\int_\Omega p^{(s)} u^{(s)} \, \mathrm{d}x = \sum_i \int_\Omega p^{(s)} \phi_i^{(s)} \, \mathrm{d}x \, u_i^{(s)} = \sum_i \int_\Omega p^{(f)} \phi_i^{(s)} \, \mathrm{d}x \, u_i^{(s)} = \int_\Omega p^{(f)} u^{(s)} \, \mathrm{d}x$$

On the other hand, motion (displacement) transfer satisfies

$$\int_\Omega u^{(f)} \phi_i^{(f)} \, \mathrm{d}x = \int_\Omega u^{(s)} \phi_i^{(f)} \, \mathrm{d}x$$

and then

$$\int_\Omega p^{(f)} u^{(f)} \, \mathrm{d}x = \sum_i \int_\Omega \phi_i^{(f)} u^{(f)} \, \mathrm{d}x \, p_i^{(f)} = \sum_i \int_\Omega \phi_i^{(f)} u^{(s)} \, \mathrm{d}x \, p_i^{(f)} = \int_\Omega p^{(f)} u^{(s)} \, \mathrm{d}x$$

It then follows that

$$\int_\Omega p^{(s)} u^{(s)} \, \mathrm{d}x = \int_\Omega p^{(f)} u^{(f)} \, \mathrm{d}x$$

ACKNOWLEDGEMENTS

The authors thank anonymous referees for their helpful comments in improving the presentation of this paper.

REFERENCES

1. Farhat C, Lesoinne M, Stern P. High performance solution of three-dimensional nonlinear aeroelastic problems via parallel partitioned algorithms: methodology and preliminary results. *Advances in Engineering Software* 1997; **28**(1):43–61.
2. Cebral JR, Löhner R. Conservative load projection and tracking for fluid-structure problems. *AIAA Journal* 1997; **35**(4):687–692.
3. Smith MJ, Hodges DH, Cesnik CES. Evaluation of computational algorithms suitable for fluid-structure interactions. *Journal of Aircraft* 2000; **37**(2):282–294.
4. Jiao X, Heath MT. Overlaying surface meshes, Part I and II. *International Journal of Computational Geometry & Applications* 2004, to appear.
5. Franke R. Scattered data interpolation: tests of some methods. *Mathematics of Computation* 1982; **38**(157):181–200.
6. Farhat C, Lesoinne M, LeTallec P. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering* 1998; **157**(1–2): 95–114.
7. Pidaparti RMV. Structural and aerodynamic data transformation using inverse isoparametric mapping. *Journal of Aircraft* 1992; **29**(3):507–509.
8. Duchon J. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, Schempp W, Zeller K (eds). Springer: Berlin, 1977; 85–100.
9. Bartels RH, Beatty JC, Barsky BA. *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*. Morgan Kaufmann: Los Altos, CA, 1987.

10. Späth H. *One Dimensional Spline Interpolation Algorithms*. AK Peters: Wellesley, MA, 1995.
11. Späth H. *Two Dimensional Spline Interpolation Algorithms*. AK Peters: Wellesley, MA, 1995.
12. de Boor C. *A Practical Guide to Splines*. Springer: Berlin, revised edition, 2001.
13. Dukowicz JK. Conservative rezoning (remapping) for general quadrilateral meshes. *Journal of Computational Physics* 1984; **54**(3):411–424.
14. Ramshaw JD. Conservative rezoning algorithm for generalized two-dimensional meshes. *Journal of Computational Physics* 1985; **59**(2):193–199.
15. Grandy J. Conservative remapping and region overlays by intersecting arbitrary polyhedra. *Journal of Computational Physics* 1999; **148**(2):433–466.
16. Arbogast T, Cowsar LC, Wheeler MF, Yotov I. Mixed finite element methods on nonmatching multiblock grids. *SIAM Journal on Numerical Analysis* 2000; **37**(4):1295–1315.
17. Belgacem FB. The mixed mortar finite element method for the incompressible stokes problem: convergence analysis. *SIAM Journal on Numerical Analysis* 2000; **37**(4):1085–1100.
18. Bernardi C, Maday Y, Patera AT. A new nonconforming approach to domain decomposition: the mortar element method. In *Nonlinear PDEs and Their Applications*, *Collège de France Seminar*, vol. XI, Brezis H, Lions JL (eds). Longman: New York, 1994; 13–51.
19. Braess D, Dahmen W. Stability estimates of the mortar finite element method for 3-dimensional problems. *East-West Journal of Numerical Mathematics* 1998; **6**(4):249–264.
20. Maman N, Farhat C. Matching fluid and structure meshes for aeroelastic computations: a parallel approach. *Computers and Structures* 1995; **54**(4):779–785.
21. Strang G, Fix GJ. *An Analysis of the Finite Element Method*. Prentice-Hall: Englewood Cliffs, NJ, 1973.
22. Heath MT. *Scientific Computing: An Introductory Survey* (2nd edn). McGraw-Hill: New York, 2002.
23. Fix GJ. Effects of quadrature errors in finite element approximation of steady state eigenvalue and parabolic problems. In *Mathematical Foundations of the Finite Element Method*. Babuska I, Aziz AK (eds). Academic Press: New York, 1972; 525–556.
24. Jaiman RK, Jiao X, Geubelle PH, Loth E. Assessment of conservative load transfer for fluid–solid interface with nonmatching meshes. *International Journal for Numerical Methods in Engineering* 2004, submitted.
25. Hanselman D, Littlefield B. *Mastering MATLAB 6*: *A Comprehensive Tutorial and Reference*. Prentice-Hall: Englewood Cliffs, NJ, 2001.
26. Jiao X, Edelsbrunner X, Heath MT. Mesh association: formulation and algorithms. In *8th International Meshing Roundtable*. Sandia Report, South Lake Tahoe, CA, 1999; 75–82.
27. Brown SA. Displacement extrapolations for CFD+CSM aeroelastic analysis. In *38th Structures, Structural Dynamics, and Material Conference*, *AIAA-97-1090* 1997; 291–300.
28. Gaitonde DV, Shang JS, Young JL. Practical aspects of higher-order numerical schemes for wave propagation phenomena. *International Journal for Numerical Methods in Engineering* 1999; **45**(12):1849–1869.
29. Cebral JR. Loose coupling algortihms for fluid–structure interaction. *Ph.D. Thesis*, George Mason University, June 1996.
30. Heath MT, Dick WA. Virtual prototyping of solid propellant rockets. *Computing in Science and Engineering* 2000; **2**(2):21–32.
31. Berger MJ. On conservation at grid interface. *SIAM Journal on Numerical Analysis* 1987; **24**(5):967–984.
32. Cai XC, Dryja M, Sarkis M. Overlapping nonmatching grid mortar element methods for elliptic problems. *SIAM Journal on Numerical Analysis* 1999; **36**(2):581–606.
33. Chesshire G, Henshaw WD. A scheme for conservative interpolation on overlapping grids. *SIAM Journal on Scientific Computing* 1994; **15**(4):819–845.
34. Cai XC. The use of pointwise interpolation in domain decomposition methods with non-nested meshes. *SIAM Journal on Scientific Computing* 1995; **16**(1):250–256.

Copyright © 2004 John Wiley & Sons, Ltd.