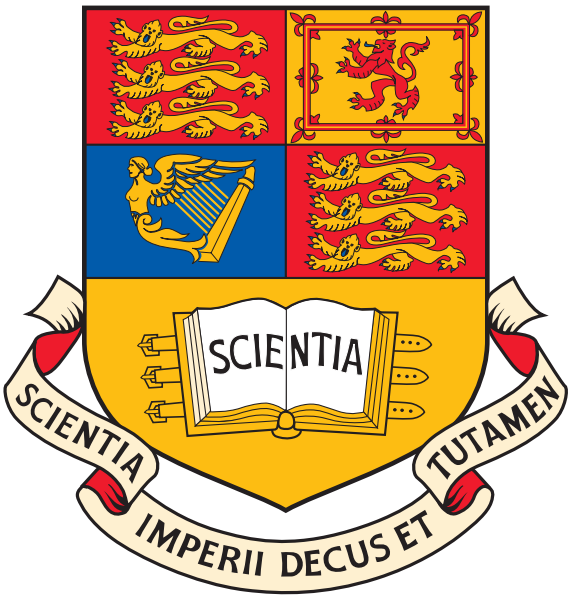
Imperial College London
Department of Earth Science & Engineering

Galerkin projection of discrete fields via supermesh construction

Patrick E. Farrell

September 2009



Supervised by:
Dr. Matthew D. Piggott
Prof. Christopher C. Pain
Dr. Gerard J. Gorman

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computational Physics
of Imperial College London
and the Diploma of Imperial College London

Declaration

I herewith certify that all material in this dissertation which is not my own work has been properly acknowledged.

Patrick E. Farrell

Abstract

Interpolation of discrete fields arises frequently in computational physics. This thesis focuses on the novel implementation and analysis of Galerkin projection, an interpolation technique with three principal advantages over its competitors: it is optimally accurate in the L_2 norm, it is conservative, and it is well-defined in the case of spaces of discontinuous functions. While these desirable properties have been known for some time, the implementation of Galerkin projection is challenging; this thesis reports the first successful general implementation.

A thorough review of the history, development and current frontiers of adaptive remeshing is given. Adaptive remeshing is the primary motivation for the development of Galerkin projection, as its use necessitates the interpolation of discrete fields. The Galerkin projection is discussed and the geometric concept necessary for its implementation, the supermesh, is introduced. The efficient local construction of the supermesh of two meshes by the intersection of the elements of the input meshes is then described. Next, the element-element association problem of identifying which elements from the input meshes intersect is analysed. With efficient algorithms for its construction in hand, applications of supermeshing other than Galerkin projections are discussed, focusing on the computation of diagnostics of simulations which employ adaptive remeshing. Examples demonstrating the effectiveness and efficiency of the presented algorithms are given throughout. The thesis closes with some conclusions and possibilities for future work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Contents

1.1 Introduction

That the universe may be modelled by its inhabitants is a remarkable fact. There is no obvious axiom from which it follows that the universe we inhabit

should not merely be ordered, but be comprehensible to its limited denizens. We should count ourselves lucky that this is so (Deutsch, 1998).

Mathematics is the language in which the laws of physics are written. In particular, physical laws are written in the language of calculus, and are formulated as differential equations.

The universe has been very kind to allow such a compact and elegant representation of itself. There is, however, a catch. Although we humans may write down (approximations to) the laws of physics, in most cases of interest we cannot actually solve them.

Therefore, the business of computing approximate solutions to differential equations is of fundamental importance. This is achieved by a process known as *discretisation*. While qualitative properties of the exact solutions are sometimes discernible, an experimentalist or an engineer requires quantitative information to evaluate a theory or design. The quantitative prediction of physical phenomena is often referred to as scientific computing, and considered a sub-branch of numerical analysis.

While interest in scientific computing has grown exponentially since the widespread availability of digital computers, it has long been of concern to humanity. Consider the Antikythera mechanism, a remarkable analog calculator from ancient Greece designed to calculate astronomical predictions (Freeth et al., 2006, 2008). The Antikythera mechanism is a mechanical manifestation of a discretisation of the equations describing the apparent motion of the planets and stars; nowadays, our models are instantiated on general-purpose programmable computers. Our interest in these topics has been smoldering for millennia; the recent explosion of research in this matter is merely its bursting to flame.

The ability to simulate physical phenomena has placed great power in the hands of engineers. The development of the nuclear bomb and the moon landings both depended utterly on numerical calculation. Indeed, both of these projects spurred onward the development of computational hardware and techniques: one of the first general-purpose electronic computers, ENIAC, was employed in calculations for the Manhattan project. Previously, designing an aircraft required extensive use of expensive wind tunnels. Computational fluid dynamics is now an essential tool in the design of aeronautic vehicles such as the supersonic car ThrustSSC (Morgan et al., 1999); SpaceShipOne, the first privately-funded spaceplane, was de-

signed entirely without the use of wind tunnel experiments (Linehan, 2008). Structural simulation allows for the study of how cars deform in crashes, and how bridges and buildings behave under load. Numerical simulation provides your daily weather forecast.

Scientific computing is exciting because of the confluence of relevancy and infancy. While the discipline has ancient roots, most of the algorithms that make the approximation of physical solutions possible have been invented since 1950 (Trefethen, 2008). One of the major trends in the field today is the introduction of adaptive algorithms: the user specifies the goal of a computation, and the algorithm modifies the quality of the approximation to achieve it. In his predictions for the future of scientific computing, Trefethen (2000) conjectures that this trend will dominate the solution of most numerical problems by 2050.

Since the discretisation of differential equations frequently involves the subdivision of space into a mesh, one mechanism of adaptive discretisation is to change the mesh in response to some error estimator. Of this class of methods, known as h-adaptivity, adaptive remeshing is the most flexible: it allows for arbitrary changes between the original and adapted meshes. This flexibility is particularly important when anisotropic phenomena must be efficiently represented; by allowing for the mesh to align with the curvature of these phenomena, the same accuracy may be attained for significantly less computational cost (Morgan et al., 1991; Piggott et al., 2009).

Since the original and adapted meshes are in general entirely different, the question of how to *interpolate* data from the original mesh to the adapted mesh therefore arises. This interpolation problem is described as *discrete* since it involves two discrete meshes, as opposed to the discrete approximation of a continuous quantity. This question is the focus of this thesis. An optimally accurate projection method for interpolating data from one mesh to another, called *Galerkin projection*, is developed. Its theoretical properties and implementation are discussed. The implementation fundamentally relies on a geometrical construct called a *supermesh*, the mesh of the intersections of the elements of the input meshes (figure 1.1). While the desirable properties of Galerkin projection for discrete interpolation problems have been known for some time, its implementation has proven challenging: this thesis presents the first successful general implementation.

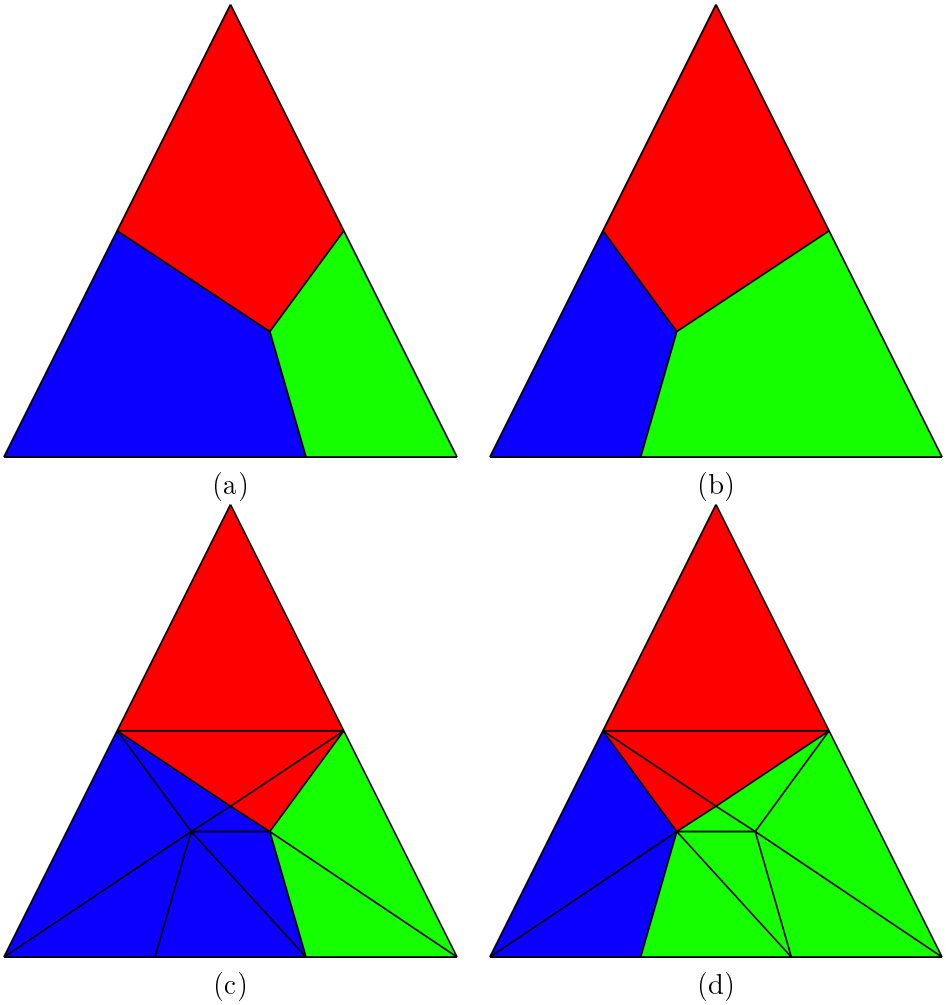


Figure 1.1: (a) and (b): Two quadrilateral meshes. (c): A triangular supermesh of (a) and (b), coloured to show the elements of (a). (d): The same supermesh of (a) and (b), coloured to show the elements of (b).

1.2 Review of anisotropic adaptive remeshing

Historically, numerical analysts concerned themselves with *a priori* error bounds of particular numerical schemes, i.e. asymptotic analyses of the order of convergence of a discretisation with respect to some discretisation parameter such as mesh sizing h or polynomial order p. However, such *a priori* error bounds do not provide useful estimates of the simulation error of a particular physical system on a particular mesh for a specified norm: they merely describe how that error behaves as the discretisation is modified. Since such *a priori* error bounds involve the unknown exact solution, they are, in general, not computable.

In the late 1970s, the pioneering work of Babuška and Rheinboldt laid the foundations for *a posteriori* error estimates (Babuška and Rheinboldt, 1978a,b). In contrast to *a priori* bounds, *a posteriori* error estimates involve only the approximate computed solution and data from the problem, and are thus computable (or approximately so, if they involve the solution of an auxiliary problem). These error estimates can then be used in an adaptive loop, modifying the discretisation until some user-specified error criterion is reached. Most *a posteriori* error estimation literature deals with estimating the error in the natural norm induced by the bilinear form of the problem, the energy norm. For a review of *a posteriori* error estimation with emphasis on energy norm estimation see the books of Verfürth (Verfürth, 1996) and Ainsworth and Oden (Ainsworth and Oden, 2000). The goal-oriented adaptive framework of Rannacher and co-workers, which estimates the error in the computation of a given goal functional, is detailed in Becker and Rannacher (2001) and Bangerth and Rannacher (2003).

Once *a posteriori* estimates have been computed, there are many possible ways of modifying the discretisation to achieve some error target. These include h-adaptivity, which changes the connectivity of the mesh (Berger and Colella, 1989); p-adaptivity, which increases the polynomial order of the approximation (Babuška and Suri, 1994); and r-adaptivity, which relocates the vertices of the mesh while retaining the same connectivity (Budd et al., 2009). Combinations of these methods are also possible (e.g., Houston and Süli (2001); Ledger et al. (2003)).

1.2.1 Some definitions

While many of these definitions will be familiar, some of the terminology used imply different concepts in different communities. For example, mesh adaptivity is frequently used as a synonym for hierarchical refinement among the hierarchical refinement community. Similarly, adaptive remeshing is occasionally taken to mean global remeshing, whereas here global remeshing is considered to be a subclass of adaptive remeshing. For clarity, the definitions used throughout this review are given below.

simplex A d-dimensional simplex is the convex hull of $d+1$ points not in an affine space of dimension $d-1$. In two dimensions, a simplex is a triangle; in three dimensions, a tetrahedron.

mesh Let Ω be a subset of $\mathbb{R}^d$ with a boundary consisting of $d-1$-dimensional polytopes. A set of convex polytopes $\mathcal{T}$ is a mesh of Ω if

1. Ω is the union of the elements of $\mathcal{T}$.
2. Every element $K \in \mathcal{T}$ has positive d-measure.
3. The intersection of any two elements of $\mathcal{T}$ has zero d-measure.

In this work, emphasis is placed on simplicial meshes, i.e. where $\mathcal{T}$ is a set of d-dimensional simplices. A grid is a synonym for a mesh.

mesh generation Mesh generation is the act of constructing a mesh $\mathcal{T}$ of Ω, given a description of the boundary $\partial\Omega$. It is generally expected that this mesh will satisfy a sizing requirement, which in the most general case is encoded in a metric field. Two of the main algorithms for simplicial mesh generation are Delaunay triangulation and the advancing front method. Mesh generation frequently uses a background mesh to specify its sizing requirements.

metric tensor field A metric tensor field is a tensor-valued function associating a symmetric positive-definite tensor to every point in a domain Ω:

$$\mathcal{M} : \Omega \to \mathbb{R}^{d \times d}.$$

The field is described as a metric as it induces a definition of distance. For a parameterisation γ of a curve Γ, the length of Γ with respect to

$\mathcal{M}$ is defined as

$$l_{\mathcal{M}}(\Gamma) = \int_0^1 \sqrt{\gamma'(t)^T \mathcal{M}(\gamma(t))\gamma'(t)}dt,$$

where $\gamma' = d\gamma/dt$. The distance between two points is the infimum over such curves. A metric tensor may be used to encode the desired lengths of a mesh by adopting the convention that the generated mesh should have all edges with edge length 1 when measured with respect to the metric. The advantage of encoding it in this manner rather than a scalar-valued sizing function is that this allows for the desired edge length to vary directionally, i.e. the metric can encode anisotropic sizing specifications.

metric space A metric space is $\mathbb{R}^d$ equipped with a sense of distance induced by a metric field $\mathcal{M}$.

Delaunay triangulation $\mathcal{T}$ is a Delaunay triangulation of Ω if the circumcircle (circumsphere) associated with each element is empty, i.e. contains no other vertices of $\mathcal{T}$. The Delaunay triangulation enjoys many optimality properties (George and Borouchaki, 1998).

advancing front method An advancing front method is a technique for mesh generation which incrementally constructs the mesh by marching a front of free sides into the domain. The initial front is given by the boundary discretisation.

h-adaptivity h-adaptivity is the act of changing the connectivity of the computational mesh, possibly adding or removing vertices, or applying operations which modify the topology of the mesh such as edge swaps.

r-adaptivity r-adaptivity is the act of changing the locations of the existing vertices of the computational mesh without changing the connectivity.

mesh adaptivity Mesh adaptivity is the act of changing the computational mesh, encompassing h-adaptivity and r-adaptivity.

p-adaptivity p-adaptivity is the act of changing the local polynomial order of the basis functions associated with a given mesh.

hierarchical refinement Hierarchical refinement is one mechanism for h-adaptivity. This consists of partitioning current elements or coarsening patches of selected elements. For a review of hierarchical refinement strategies, see Behrens (2006). Adaptive mesh refinement (AMR) is sometimes used as a synonym (e.g. Jones and Plassmann (1997)), but AMR generally connotes the use of Cartesian meshes (e.g. Berger and Colella (1989)), where it is sometimes referred to as quadtree and octree refinement in two and three dimensions. In other works, hierarchical refinement encompasses the use of certain kinds of p-adaptivity, where the adapted function space is a superset of the non-adapted function space; in this work, it is used only to mean the hierarchical h-refinement of meshes.

adaptive remeshing In contrast to hierarchical refinement, adaptive remeshing is a subclass of mesh adaptivity methods which construct an adapted mesh which in general may be entirely different from the previous mesh. Thus, maximum flexibility is allowed in the meshes constructed by such an algorithm. This approach is sometimes referred to as m-adaptivity (Löhner, 1995b), or is considered as a subclass of h-adaptive methods (Zienkiewicz and Taylor, 2000a; Piggott et al., 2005). Global remeshing, local remeshing and mesh optimisation are three mechanisms for constructing such an adapted mesh.

global remeshing Having constructed a sizing specification from an error analysis of the computed approximate solution on a previous mesh, global remeshing is the act of generating an entirely new mesh of the same domain satisfying the sizing specification.

local remeshing Global remeshing regenerates the mesh of the entire domain. By contrast, local remeshing is a mechanism of adaptive remeshing in which cavities of elements are removed and the hole remeshed. These cavities are identified by measuring their conformity to a given sizing specification.

mesh optimisation In contrast to global remeshing, where the previous mesh is used merely to describe the sizing specifications, mesh optimisation is a mechanism of adaptive remeshing which deforms the previous mesh to the adapted mesh by a sequence of local operations.

The mesh quality is measured by a functional measuring how close the mesh is to the mesh encoded in the sizing specification. The local mesh modification operations are applied successively to optimise this functional.

1.2.2 Scope of this review

This review focuses on adaptive remeshing in multiple dimensions, with particular emphasis on the exploitation of anisotropic solution features. Since the meshes produced are not constrained by the previous mesh, this approach allows for maximum flexibility in adapting to solution features. However, this flexibility comes at a cost: guiding the adaptive remeshing procedure (choosing what mesh to construct), executing the adaptation (constructing the chosen mesh) and data transfer of solution fields (from the previous mesh to the newly adapted mesh) become more complicated than with hierarchical refinement.

This review does not discuss hierarchical refinement. Some hierarchical refinement approaches attempt to generate anisotropic child elements (e.g., Apel et al. (2004); Richter (2009)); these are not discussed. Nor does this review discuss those r-adaptive algorithms which attempt to generate anisotropic meshes (e.g. Brackbill (1993); Schneider and Jimack (2006)). This review also does not discuss mesh generation except in the context of global remeshing; for a review of mesh generation, see George and Borouchaki (1998), Thompson et al. (1999) or Frey and George (2008).

1.2.3 Adaptive remeshing technology

In this section, techniques for the construction of an adapted mesh given a sizing specification are reviewed. The construction of the sizing specification is described in §1.2.4.

Adaptive remeshing in one dimension to optimally distribute nodes for the interpolation of a given function can be traced back to the equidistribution principle of de Boor (de Boor, 1973). However, adaptive remeshing procedures in multiple dimensions require an automated mesh generation capability, and thus the development of these algorithms had to wait until robust, automatic mesh generation algorithms were available. The first anisotropic adaptive remeshing method published in the literature was Peraire et al.

(1987). This work applied global remeshing to the solution of the stationary two-dimensional Euler equations. Aside from its significance to adaptive remeshing, the paper discusses an important advance in the development of advancing front mesh generation. The mesh sizing is controlled by two scalar fields and one vector field: a mesh sizing function, a mesh aspect ratio and a stretching direction. These parameters are computed from the eigendecomposition of the Hessian of the density. The highest aspect ratio reported in the examples is 6 and the largest mesh used had approximately 10^3 elements.

Adaptive remeshing was first applied to transient simulations in Löhner (1988). The algorithm was applied to transient fluid-structure interaction computations. The maximum allowed aspect ratio was set to 5. The author comments that the gain over uniformly fine grids depends on the degree of anisotropy present in the solutions, but estimates the speedup as lying between 10 and 50. The author also notes briefly that the repeated interpolation necessitated by adaptive remeshing may be diffusive. Löhner (1989) extends the error indicator used so that it is normalised to be dimensionless and comments that applying a gradation algorithm to the mesh parameters and bounding the element sizes results in adapted meshes that are more suitable for the computational simulation. This work also couples the adaptive remeshing with hierarchical refinement techniques, as hierarchical refinement was more easily parallelisable on the vector machines available. Löhner comments in the conclusion that the questions of interpolation and conservation through adaptive remeshing deserve further scrutiny.

The next advance was to apply the same method to stationary (Peraire et al., 1988; Morgan et al., 1991) and transient (Löhner, 1990) computations in three dimensions. The adapted mesh is described by three scalar and two vector fields: the mesh sizing, aspect ratios, and stretching directions. The meshes used in the computations reported had approximately 10^5 elements. Peraire et al. (1988) does not report aspect ratio statistics, but Löhner (1990) limits it to 1.5. The reason for this limit is not explained.

Mavriplis (1990) extended adaptive remeshing to viscous Navier-Stokes simulations in two dimensions. The construction of the adapted mesh is achieved by means of a Delaunay triangulation. The author comments that the aspect ratios required for resolving viscous boundary-layer flows are several orders of magnitude higher than those achieved previously. Since the

Delaunay triangulation will generally produce triangles with low aspect ratio, the Delaunay triangulation is performed in a mapped space, where the mapping is induced from the stretching vector and magnitude. This extends the ideas presented in the previous literature where a local coordinate transformation is used during the advancing front point insertion. However, in this work the stretching ratios were not computed from the curvatures of the flow solution, but were taken as the ratios from the initial, hand-generated, mesh.

The key idea of forming the triangulation entirely in a metric space, with the coordinate mapping given by a metric tensor, was first published in Vallet (1990). An isotropic triangulation of unit edge length (a so-called unit mesh) is constructed via Delaunay triangulation in the mapped space, which is then mapped back to give a non-equilateral anisotropic mesh in Euclidean space. Therefore, the desired mesh distribution (sizing, aspect ratio and orientation) is elegantly encoded in a single mathematical object. This insight provided the basis for most of the future extensions and applications of anisotropic adaptive remeshing. The metric tensor is derived from the Hessian of a key variable of the solution. The connection between interpolation error and differential geometry was further developed in D'Azevedo (1991) and D'Azevedo and Simpson (1991). By the mid-90s this idea was well known, as evidenced in the reviews of Simpson (Simpson, 1994), Löhner (Löhner, 1995b) and Baker (Baker, 1997). Löhner comments that the state of the art of mesh generation at the time was not yet able to routinely and robustly generate elements with aspect ratios on the order of 10^3. Baker (Baker, 1997) notes two difficulties with adaptive remeshing: the first is that while hierarchical refinement only needs element-level indicators measuring the error in some norm, adaptive remeshing requires the specification of a mesh sizing field, which is a much more difficult task. The second criticism relates to the stability of the mesh adaptation procedure for stationary flows; Baker comments that a stable adaptation process is much more difficult to achieve when the mesh changes globally rather than by hierarchical refinement, which is inherently local. He suggests that these reasons underlie the popularity of hierarchical refinement over adaptive remeshing. (The convergence of adaptive remeshing was discussed in one dimension in Pryce (1989)). Curiously, the fact that adaptive remeshing is much more technically difficult is unmentioned. By contrast, Zhu and Zienkiewicz (1997)

strongly endorse adaptive remeshing:

> To achieve an optimal mesh for a given accuracy using adaptive h refinement or h-p refinement, mesh generation executed by an automatic mesh generator or an automatic mesh enrichment procedure is crucial. Although the adaptive procedure is simpler if mesh enrichment is employed, which keeps the refinement on the previously used meshes, partial or complete remeshing appears to be more efficient for a large class of problems as the desired accuracy can be obtained in fewer adaptive analysis steps. For many practical problems such as simulation of forming processes, optimum design, and problems arising in fluid dynamics, remeshing is usually inevitable in the process of finite element approximation. An adaptive procedure with an automatic mesh generation capability is therefore the natural approach to be used in the finite element analysis of these problems.

In the terminology used in this review, "adaptive h refinement" would be referred to as h-adaptivity.

While optimisation of meshes has been studied for several decades (e.g., Kennon and Dulikravich (1986)), the first published work discussing mesh optimisation with respect to a metric appears to be that of Briere de l'Isle and George (1995). In this work, iterations of node relocation, edge removal, node insertion and node deletion are performed in three dimensions with respect to a given metric field to optimise a functional measuring the conformity of the mesh to the metric. Each optimisation operation is only performed if the quality of the mesh (measured as the quality of the worst element involved in the operation) improves. In Bossen and Heckbert (1996), iterations of node relocation, node insertion, node deletion and edge swaps are performed in two dimensions. No mesh functional is used; therefore, it is not a true mesh optimisation method. Instead, nodes are marked as inactive if the node relocation has little effect on their positions. The algorithm is not employed in an adaptive loop. Borouchaki et al. (1997a) also apply edge swapping and node relocation to improve the result of a Delaunay triangulation governed by a metric. This work was again confined to two dimensions. Borouchaki et al. (1997b) applies the algorithm to a viscous Navier-Stokes simulation, achieving aspect ratios on the order of 10^2. Peraire and Morgan

(1997) apply mesh optimisation operations in three dimensions to generate anisotropic meshes, but the process is not guided by a metric; instead, it is driven by a scalar function measuring the distance to a surface along which the user specifies that the elements should be anisotropic.

An authoritative review of the state of the art of mesh generation and adaptive remeshing is given in the book of George and Borouchaki (George and Borouchaki, 1998). The authors also extensively discuss mesh optimisation in two and three dimensions. They comment that in their experience, mesh optimisation iterations after mesh generation are generally unnecessary in two dimensions, but are important to mesh quality in three dimensions. The authors propose and examine several mesh quality functionals, and enumerate local optimisation operations in two and three dimensions.

One refinement of global remeshing was the development of local remeshing (Hassan et al., 1998, 2000). Remeshing the entire domain can be expensive, especially if the areas to be changed comprise a small fraction of the domain. Furthermore, by changing the mesh everywhere, unnecessary interpolation errors are introduced. Instead, the normalised second derivatives of a key single variable are employed to identify regions where the mesh should be adapted. Then, these regions are removed to form cavities. A mesh generation algorithm is then called to mesh each cavity with respect to a sizing function determined from the previous solution. Further examples of this technique are presented in Hassan et al. (2007). The authors comment that for their application (aeronautics), the non-conservative character of collocation interpolation does not affect the quality of the results. Local remeshing is particularly suited to simulations with moving boundaries, as the changing geometry typically distorts small areas of the domain. By only remeshing where necessary, the computational procedure becomes much more efficient.

Since the late 1990s, mesh optimisation has come to be the most popular approach for adaptive remeshing. Freitag and Ollivier-Gooch (1997) was very influential in popularising mesh optimisation, but the algorithm is not performed with respect to a metric tensor field. It appears that Buscaglia and Dari (1997) was the first to apply pure mesh optimisation (i.e., no global remeshing) with respect to a metric in two dimensions. Dolejší (1998) also appears to have developed a similar approach and compares its computational efficiency to hierarchical refinement. Vasilevskii and Lipnikov (1999) suggested the use of a modified mesh quality functional which explicitly

accounts for both element size and shape and extended the analysis of the convergence of adaptive mesh optimisation algorithms. The authors apply it to the solution of the Navier-Stokes equations around an airfoil and comment that the rigorous application of adaptive remeshing techniques remained an open question. While preliminary results of an in-development mesh optimisation algorithm were reported in Dompierre et al. (1998), Agouzal et al. (1999) appears to be the first to have robustly extended these techniques to three dimensions. Both Tam et al. (2000) and Pain et al. (2001) appear to have independently developed implementations of anisotropic mesh optimisation to three dimensions; none of Agouzal et al., Tam et al. or Pain et al. cite each other. All of the applications in this thesis apply the mesh optimisation algorithms described in Vasilevskii and Lipnikov (1999) and Pain et al. (2001) in two and three dimensions respectively.

Since these developments, numerous other groups have reported the implementation of the core ideas. These are briefly reported. Dompierre et al. (2002), implementing the algorithm of Habashi et al. (2000), report aspect ratios of 10^7 in two dimensions. The authors investigate the sensitivity of the converged adapted mesh to the initial mesh chosen, and conclude that the final quality of the adapted mesh is independent of this choice. Bottasso (2004) compares different mesh quality functional choices. The author concludes that, of the functionals considered, a functional combining the edge lengths and inscribed radius is the most effective. Gruau and Coupez (2005) develop metrics to create initial meshes that resolve geometrical interfaces between subdomains found in material forming studies. The method is then applied to metal forging simulations in Boussetta et al. (2006). Li et al. (2005) emphasise the accurate placement of new nodes on three-dimensional CAD data to conform to the geometry description. Remacle et al. (2005) apply anisotropic adaptive remeshing to discontinuous Galerkin simulations. The authors develop an algorithm to specify a metric which is aligned with discontinuities in the solution. Acikgoz and Bottasso (2007) contrast a simulated annealing optimisation algorithm with the usual Gauss-Seidel approach. The authors conclude that simulated annealing allows the optimisation to escape local minima. Sahni et al. (2006) and Sahni et al. (2008) combine anisotropic adaptive remeshing with semi-structured boundary layer meshes where the presence of boundary layers are expected *a priori*. This is achieved by decomposing the metric into wall-normal and

wall-tangential sizing information and applying constrained mesh optimisation operations to adapt the boundary layer mesh while retaining its structure. Compère et al. (2008) apply mesh optimisation to multi-phase fluid simulations. The authors comment that global remeshing is generally faster than their implementation of local mesh optimisation. However, they prefer local mesh optimisation for transient simulations because the mesh can remain unchanged in most of the domain. This minimises the interpolation error introduced. Park and Darmofal (2008) discuss the combination of metric-based adaptivity and cut-cell methods for complex geometries. This allows the remeshing step to be simplified as it does not need to exactly conform to the geometry. Pagnutti and Ollivier-Gooch (2008) report the development of a two-dimensional mesh optimisation procedure. They abandon the use of the node relocation, claiming it is too expensive; however, they do not present any quantitative evidence for this claim. Nguyen et al. (2009) apply the anisotropic centroidal Voronoi tessellation algorithm of Du and Wang (2005) to boundary layer resolution in two-dimensional convection-diffusion problems. Aubé et al. (2009) validate an anisotropic mesh optimisation procedure against boundary-layer wind tunnel data for a high-rise building in China.

1.2.4 Metric formation

The flexibility of adaptive remeshing comes at a cost. Adaptive remeshing is more complicated to guide than hierarchical or p-refinement. To guide hierarchical or p-refinement, one needs element-level indicators measuring the contribution of the element to some quantification of the error. By contrast, the input to the adaptive remeshing algorithm is a metric specifying the sizing and orientation of the desired output mesh. This extra step of computing what mesh would (approximately) give a desired target error is the characteristic challenge of guiding the adaptive remeshing algorithm.

1.2.4.1 Interpolation-based metrics

Most of the work to date has been guided by considerations of interpolation error. In one dimension, for interpolation using p^{th} order Lagrange polynomials, this is related to the $(p+1)^{th}$ derivative of the function being interpolated; in multiple dimensions, this extends to the tensor of $(p+1)^{th}$

order partial derivatives of the function being interpolated. For linear interpolation ($p = 1$), the interpolation error depends on the Hessian (the matrix of second-order partial derivatives), which naturally induces a metric in which to form a unit mesh. Given that interpolation of a function over a triangulation is fundamental to many numerical schemes, it is somewhat surprising that new developments are still being published, even for piecewise linear interpolation over triangles. For an excellent historical chronology of interpolation, see the review of Meijering (Meijering, 2002).

Peraire et al. (1987) compute the desired mesh sizing, aspect ratio and orientation from the Hessian of a key variable of the solution, in this case density. This is justified with a heuristic argument that shows that if the solution were nodally exact, then the error can be approximated using the second derivatives of the exact solution (a modern form of the argument is given in Frey and Alauzet (2005)). This *a priori* argument is employed as an *a posteriori* error indicator by approximating the second derivatives of the exact solution with the second derivatives of the computed solution. D'Azevedo (1991) and D'Azevedo and Simpson (1991) consider the problems of generating the optimal mesh to achieve a specified interpolation error of a given analytical function in the L_∞-norm and H_1-seminorm respectively. These questions are resolved by computing coordinate transformations in which an equilateral triangular mesh is optimal when considered in Euclidean space. These coordinate transformations again depend on the Hessian of the function to be interpolated. This result also extends to bilinear quadratic elements; see D'Azevedo (1999).

Throughout this chapter, the *optimal mesh* is defined to be that mesh which minimises some upper bound of the interpolation error in some norm. Therefore, optimality is defined with respect not only to a norm, but to an error bound also; different authors may define different optimal meshes or metrics for the same norm, depending on the form of the error bound employed.

Rippa (1992) further extends these results by providing theoretical justification for the observation that anisotropy can be beneficial for interpolation, when the anisotropy is aligned with the eigenvectors of the Hessian of the function to be interpolated, as these give the principal directions of curvature of the function. Rippa states that the rule of thumb drawn from the Bramble-Zlámal and Babuška-Aziz error bounds (Bramble and Zlámal,

1970; Babuška and Aziz, 1976) that anisotropy is harmful to interpolation accuracy is based on the implicit assumption that the second derivatives of the function to be interpolated are of equal magnitude. When this is not the case, this conclusion no longer follows from these bounds. Apel and Dobrowolski (1992) develop anisotropic error estimates for the H_1-seminorm in two dimensions.

A significant practical advance was the development of a technique for superimposing the anisotropic mesh requirements derived from several fields by Castro-Díaz et al. (1995), removing the necessity of choosing a single variable to guide the adaptive algorithm. This metric intersection is further described in Castro-Díaz et al. (1997) and Borouchaki et al. (1997a).

Apel (1999) proposes several alternate quasi-interpolation operators which enable the proof of error estimates on anisotropic meshes. Formaggia and Perotto (2001) develop anisotropic interpolation error estimates in the H_1-seminorm and L_2-norm for functions in $H^1(\Omega)$, by considering Clément or Scott-Zhang quasi-interpolation operators in place of the usual Lagrange interpolation operator, which is not necessarily defined for such functions (Clément, 1975; Scott and Zhang, 1990). Since the H_1-seminorm of interpolation error becomes unbounded as the maximum angle of a triangle approaches π, interpolation error estimates should reflect this asymptotic behaviour. However, the error bounds discussed previously do not capture this behaviour, and thus require a maximal angle condition to retain their relevancy. The authors are able to remove the typical maximal angle condition by developing error bounds which display the correct asymptotic behaviour, but comment that the estimate of Apel (1999) is more accurate for a right-angled triangle. These are then used to derive *a priori* and *a posteriori* error estimates for elliptic problems in Formaggia and Perotto (2003).

The manuscript of Shewchuk (2002a) discusses error bounds and quality measures for mesh generation. Emphasis is placed on error bounds that are informative not just in the asymptotic limit, but are useful for guiding mesh optimisation. The relationship between the interpolation error in the L_∞-norm, interpolation error in the H_1-seminorm, and stiffness matrix conditioning is investigated. The author demonstrates examples where the ideal elements for each of these considerations disagree.

In Coudière et al. (2002), the authors discuss the isotropic interpolation of functions that are piecewise regular, with discontinuities along a $(d-1)$-

dimensional manifold in between the areas of regularity. They define the order of convergence α to relate the interpolation error in the L_p norm (p finite) to the number of elements in the mesh. The authors claim that for isotropic mesh adaptivity, the order of convergence is bounded by

$$\alpha \leq \frac{d/p}{d-1}, \tag{1.1}$$

where d is the space dimension. This is a rather severe limit; it implies a maximum convergence order of 3/4 for the L_2 norm in three dimensions. (For smoother functions, higher convergence rates are expected.) The authors give numerical evidence that this bound does not apply to anisotropic mesh adaptivity. Dervieux et al. (2003) consider this as strong motivation in favour of the application of anisotropic adaptive remeshing to problems with possibly discontinuous solutions. Alauzet (2008) shows that adaptive remeshing recovers the theoretical second-order convergence of shock-capturing methods in the presence of discontinuities, whereas uniform refinement fails to attain the predicted convergence order.

In the meshing algorithms described above, a metric tensor (itself represented by an interpolant on a mesh) is used to encode the desired mesh to be constructed. Dervieux et al. (2003) and Courty et al. (2006) extend this idea by considering a continuously-defined metric as the abstract representative of a discrete mesh. By posing the problem in a continuous manner, this approach allows for the calculus of variations to be applied to the problem of determining the optimal mesh for various problems; the authors apply it to finding the optimal mesh on which to interpolate a specified function in a given L_p norm in two dimensions. This is extended to three dimensions in Alauzet et al. (2006b). Alauzet et al. (2008) comment that adapting to the L_2 norm instead of the L_∞ norm is very important for their application (sonic boom reduction) as the output functional depends strongly on weak phenomena; the L_∞ norm concentrates on the strongest shocks, while the L_2 norm is more sensitive to weaker variations. It appears that Chen et al. (2007) has independently derived the same metric formulation.

Recent developments have focussed on the application of anisotropic adaptive remeshing to higher-order methods ($p > 1$). There have been several heuristic approaches published. The method of Belhamadia et al. (2004) reconstructs a higher-order Hermite approximation of the solution by ap-

plying a derivative recovery algorithm to the numerical approximation. The difference between the recovered Hermite approximation and the numerical solution is then considered to be the error and the mesh modified by optimisation operations to reduce it. This is applied in two dimensions to the solution of the Stefan phase-change problem in Belhamadia et al. (2004) and to the bidomain model for electrocardiology in Belhamadia (2008). An alternative approach is described in Pagnutti and Ollivier-Gooch (2009). The algorithm consists of modelling the error as the $(p+1)^{th}$ term in the Taylor expansion of the field to be interpolated. These higher-order derivatives are recovered and the metric components computed to approximate the $(p+1)^{th}$ power of these derivatives. This approximation is performed by a computation involving the Fourier coefficients of the derivatives, expressed in spherical coordinates. While heuristic, the authors present numerical evidence that it improves the order of convergence for quantities of interest for flow past an airfoil. The authors claim that "we are the only authors to extend Hessian-based anisotropic refinement to higher order methods", which is not the case. More rigorous approaches are presented in Cao (2008) and Huang (2005). Cao (2008) proves new anisotropic interpolation error estimates and applies these to the problem of metric formation, developing a formula for the optimal metric for k^{th} order Lagrange interpolation in the $W^{m,p}$ seminorm in two dimensions in terms of generalised anisotropic diagnostics of higher-order derivatives. The results generalise earlier bounds developed in Cao (2005) and Cao (2007). These anisotropic diagnostics extend the notion of the orientation and aspect ratio of the derivatives (given, for the second derivatives, by the eigendecomposition) to derivatives greater than 2. Huang (2005) also considers this problem and develops alternative expressions for the optimal metric. The formulae of Cao (2008) are expressed in terms of physically meaningful quantities and are therefore easier to understand, but the expressions of Huang (2005) have the advantage of being written for arbitrary dimension. A survey incorporating this development is given in Huang (2006).

For transient phenomena, if the mesh is adapted solely to well-represent the solution fields at the time of adaptation, it will in general lag behind the dynamics as they evolve, possibly compromising the suitability of the mesh. Since these bounds on interpolation error are for a function not changing in time, the adaptive remeshing must be modified to take into account tran-

sient phenomena, for the mesh produced by the adaptive remeshing must be suitable for the computation until the next invocation. One way to achieve this is with a goal-based approach, which determines the necessary spatial and temporal resolution to resolve some functional to a desired degree of accuracy (see §1.2.4.2). However, this requires the computation of an adjoint (or dual) solution. An adjoint-free alternative, described in Alauzet et al. (2003), is to introduce a new iteration in the solver loop. Suppose the solution has been computed up to time T and that the task at hand is to generate a suitable mesh for the interval $[T, T+\Delta T]$, where ΔT is the adaptivity period; typically $\Delta T = n\Delta t$ with n between 10 and 20. The algorithm timesteps forward until $T+\Delta T$, computing a metric for each timestep using the formulations described above. These metrics are superimposed through time, producing a metric at $T + \Delta T$ which is suitable for representing all the intermediate dynamics over the interval. The mesh is then adapted to this intersected metric and the computation restarted at time T. This procedure is then iterated until the mesh and solution produced have converged together.

1.2.4.2 Goal-based metrics

One of the major advances in the numerical solution of partial differential equations in the 1990s was the development of goal-based error estimation by Rannacher and co-workers. Rather than estimate the error in an energy or L_p norm, this method gives computable error estimators for quantities of the form

$$J(u) - J(u_h),$$

where u is the exact solution to some variational problem, u_h is a Galerkin approximation in some finite-dimensional subspace, and J is a user-supplied functional of the output to be computed. This technique applies to both linear and nonlinear PDEs, and linear and nonlinear functionals (Bangerth and Rannacher, 2003). This approach requires the solution of a linearised adjoint problem. The power and utility of this framework is evident: it gives a quantitative measure of the amount of computational effort required to compute some desired goal output of a simulation to the desired accuracy. This information can be exploited to compute the goal output much more cheaply than by merely controlling the error in an energy or L_p norm.

However, the goal-based framework naturally supplies element-level error indicators, associating an element with its contribution to the error in the functional. Therefore, it is usually combined with hierarchical or p- adaptivity (or a combination).

The first combination of goal-based adaptivity and anisotropic adaptive remeshing was published by Venditti (Venditti and Darmofal, 2003) and elaborated in his thesis (Venditti, 2002). The approach was applied to two-dimensional stationary flow. The orientation and aspect ratio information is derived from the Hessian of the Mach number, while information on desired mesh sizing is computed for each element by a combination of the current mesh size and an adjoint-related factor. The choice of Mach number is arbitrary; in his conclusions, the author states that ideally the anisotropic information of aspect ratio and orientation should be derived from adjoint criteria. The effectivity of the approach is demonstrated on the examples of lift and drag past an airfoil.

Power et al. (2006) apply a similar approach to time-dependent problems in ocean modelling. The adjoint is approximately computed by taking large timesteps forward and backward ($\sim 10 - 20\times$ the simulation timestep) to minimise the computational effort devoted to the auxiliary problem. For each prognostic field, nodal weights are computed from the adjoint. These measure the relevance to the goal and are used to weight the Hessian of the field. This approach is further elaborated in Power (2008).

The most rigorous published result to date is that presented in Formaggia et al. (2004). Element-level indicators, involving the forward residual and adjoint error, are computed to provide an error estimate for a given functional. In contrast to the previous approaches, which combined the orientation and stretching from the Hessian with a heuristic weighting given by the adjoint, the approach taken here explicitly minimises these element-level indicators with respect to element stretching and orientation. The solution to this optimisation problem turns out to be related to the eigendecomposition of a matrix whose components are a function of the first derivatives of the adjoint solution. The algorithm is demonstrated for the stationary two-dimensional Stokes and advection-diffusion-reaction problems.

It appears that A. Loiselle of INRIA has also developed an adaptive remeshing strategy for functional outputs (Loseille, 2008); however, as of the time of writing, no publications in English are yet available.

1.2.5 Related topics

1.2.5.1 Hessian recovery

As can be seen in §1.2.4, almost every adaptive remeshing scheme relies somewhere on the recovery of derivatives of the discrete solution, usually the Hessian. Since the algorithm is generally applied to piecewise linear fields, the second derivative is formally zero on the element interiors and undefined at the element boundaries. Therefore, some recovery procedure must be applied to compute an approximation to the Hessian. Several methods have been proposed to recover the Hessian $\mathcal{H}$ from a piecewise linear scalar field u. In this subsection, attention is restricted to piecewise linear interpolants. For a quantitative comparison of Hessian recovery methods, see Lipnikov and Vasilevskii (2006) and Vallet et al. (2007).

The idea behind quadratic fitting (Vallet et al., 2007) is to locally approximate the piecewise linear approximation to a function by a smooth quadratic polynomial, then differentiate that polynomial analytically. The local quadratic approximation at a node is obtained by performing a least-squares fit on the coefficients of the quadratic polynomial over the nodes in a patch of elements surrounding that node. Let $\Delta = \{\delta_1, \delta_2, ...\} = \{1, x, y, xy, x^2, ...\}$ be the set of basis functions for a quadratic polynomial ($|\Delta| = 6$ for two dimensions, 11 for three). For a given node in the mesh, let $\mathcal{K}$ be the set of neighbouring nodes in the patch around that node. The coefficients of the quadratic approximation Q are the solution of the linear system

$$P^T P Q = P^T B,$$

where $P_{jl} = \delta_l(\mathcal{K}_j)$, and $B_j = u(\mathcal{K}_j)$. The patch must contain sufficient nodes to constrain the least-squares fit. The second derivatives of this quadratic polynomial are then taken as the approximation to the second derivatives of the solution field at the node. For the 2D case, Vallet et al. (2007) found this method was the most accurate and robust of the methods compared. In the domain interior, this method is exact for constant, linear and quadratic underlying solution fields. In this author's experience, this method can suffer from oscillations reminiscent of Runge's phenomenon in the vicinity of sharp interfaces, such as those present in multimaterial simulations.

The superconvergent patch recovery (SPR) method (Zienkiewicz and Zhu, 1992; Zhang and Zhu, 1995) is motivated by the observation that there exist points within an element at which the derivative is one order more accurate than the theoretical convergence rate for direct differentiation of the basis functions. Such points are called superconvergent points. Given the values of the derivative at these points, the algorithm computes an approximation to the derivative of the field at the mesh vertices which demonstrates higher-order convergence everywhere, not just at the superconvergent points. The idea is to construct a continuous polynomial expansion of the derivative of the field in a patch of elements surrounding the node at which the value is desired. The polynomial is formed by performing a least-squares fit of the derivative at the superconvergent points within the elements in the patch. The polynomial is then evaluated at the node to give the higher-order accurate value for the derivative. This process can be applied recursively to obtain higher order derivatives of a field. By comparing the recovered gradient with the direct derivative of the finite element solution, this method has been widely applied to yield error indicators for h-refinement (Zhu and Zienkiewicz, 1997).

Buscaglia and Dari (1997) apply Green's formula and lump the mass matrix to give an equation for the Hessian for a node n as

$$\mathcal{H}_{ij}^n = \frac{-1}{2M_n^L} \int_\Omega \left(\frac{\partial u}{\partial x_i}\frac{\partial \phi_n}{\partial x_j} + \frac{\partial u}{\partial x_j}\frac{\partial \phi_n}{\partial x_i} \right) \, dV + \frac{1}{M_n^L} \int_{\partial\Omega} \frac{\partial u}{\partial n}\phi_n \, dS,$$

where Ω is the domain of integration, ϕ_n the basis function, and M_n^L is the n^{th} diagonal entry of the row-summed lumped mass matrix. The boundary term is usually neglected and an extrapolation algorithm used to compute values of the Hessian on the boundary (Buscaglia and Dari, 1997; Alauzet, 2003). In the numerical comparison of Lipnikov and Vasilevskii (2006), this recovery algorithm yields the lowest error in the L_∞ norm for an adaptive remeshing loop; however, the difference between this and the double lumped Galerkin projection described below was found to be marginal.

Pain et al. (2001) uses a double lumped Galerkin projection to compute the Hessian as

$$\mathcal{H}_{ij}^n = \frac{1}{M_n^L} \int_\Omega \phi_n \frac{\partial q_i}{\partial x_j} dV,$$

where M^L is the row-summed lumped mass matrix and q_i is a piecewise

linear projection of the first derivative given by

$$q_i = \frac{1}{M_n^L} \int_\Omega \phi_n \frac{\partial u}{\partial x_i} dV. \tag{1.2}$$

The off-diagonal entries are then averaged to enforce symmetry of the computed Hessian. The mass matrix is lumped for computational efficiency. Bank and Xu (2003) suggest the application of a multigrid-like smoothing operator to post-process the lumped Galerkin projection; however, Lipnikov and Vasilevskii (2006) demonstrate numerically that applying this smoothing operator negatively affects the L_∞ error and convergence rate of an adaptive remeshing loop for several analytical cases and recommend against its application in adaptive remeshing.

The work presented in this thesis applies the double lumped Galerkin projection as described in Pain et al. (2001).

1.2.5.2 Gradation

A metric derived from error considerations may yield sudden changes in desired mesh edge length, due to the nature of the problem being resolved. Such sudden changes are undesirable in a mesh. For example, sudden changes in mesh sizing can cause the spurious reflection of waves (Bažant, 1978; Bangerth and Rannacher, 2001). Therefore, a mesh gradation algorithm is often applied to smooth out sudden variations in the metric, which results in more gradual changes in mesh spacing.

Various gradation algorithms have been introduced to solve this problem. Löhner (1996) uses various functions of distance to point sources where edge length is specified by the user to control the isotropic sizing function for an advancing front mesh generator. Owen and Saigal (2000) apply natural neighbour interpolation to smooth sudden variations in an isotropic sizing function. Persson (2006) bounds the gradient of an isotropic sizing function by solving a partial differential equation. Borouchaki et al. (1998) introduced two gradation algorithms for scalar isotropic mesh sizing functions, bounding the gradient of the sizing function or the ratio of the length of two adjacent edges, along with anisotropic generalisations of these.

Li et al. (2004) gave an anisotropic generalisation of the algorithm presented in Borouchaki et al. (1998) to bound the ratio of two adjacent edge lengths. Remacle et al. (2005) describes this as a crucial part of an anisotropic

adaptive algorithm for discontinuous Galerkin methods. In the anisotropic generalisation given in Borouchaki et al. (1998), when considering an edge PQ between two nodes P and Q, only the length associated with the direction PQ was bounded. Li et al. (2004) extends this to bound the ratio of edge lengths in all directions at P and Q.

There appears to be no quantification of the improvement arising from these gradation algorithms in the literature; however, engineering experience indicates their usefulness.

1.2.5.3 Parallelisation

Most practical computations are too large or too computationally expensive to store or run on a single processor. Therefore, if an algorithm is to be applied to real-world situations, it must be parallelised.

Clearly, adaptive remeshing in parallel is intimately related to parallel mesh generation. However, reviewing parallel mesh generation would be a chapter in itself. Therefore, papers discussing parallel mesh generation will only be discussed where relevant to adaptive methods.

Two of the key issues when considering a parallel adaptive remeshing algorithm are the synchronisation of interfaces and load-balancing the adapted mesh. Since the interface between two processors must be consistent on both, this constraint must be enforced in the remeshing procedure. If the output mesh of this procedure has a mesh density very different to the input mesh, the parallel decomposition must also change to balance the computational load across the processors available.

Coupez et al. (2000) deal with interface consistency by first locking the shared regions of the mesh. The unlocked regions are then remeshed and the parallel decomposition perturbed away from the existing interface by element exchanges (as opposed to a graph repartitioning strategy). The procedure is then iterated until all of the mesh has been adapted (if necessary). All the examples presented in the paper are isotropic. The maximum number of processors used in the examples is 32, with a parallel efficiency of 0.18 on 32 processors for a two-dimensional case.

Freitag et al. (1999) take a different approach. Here, the parallelism is more fine-grained: rather than locking the shared regions, the vertices of the (globally distributed) mesh are coloured using a graph-colouring algo-

rithm. The algorithm consists of sweeping through the vertices of the mesh of the same colour in sequence, applying the optimisation operations to these vertices, and then synchronising across the processors. This approach requires invasive changes to a serial mesh optimisation algorithm and incurs a relatively large number of small communications.

Alauzet et al. (2006a) interleave optimisation operations and communication in a different manner. When coarsening a mesh by removing a vertex, the set of mesh entities to be affected is computed. If these entities are all local, the operation is carried out; otherwise, it is stored as pending in a local buffer. When all processors have finished carrying out their possible optimisations, the entire set of mesh entities to be affected by a vertex removal is transferred to one processor so that the pending operation can be carried out. When refining, the refinement operation is carried out and an update notification buffered. When all refinement operations have occurred, the buffers are exchanged and each processor synchronises its copies of mesh entities with the operations performed on them by other processors. Load balancing is achieved with the Zoltan library (Devine et al., 2002).

Gorman (2003) takes a similar approach to Coupez et al. (2000). Again, the shared regions are locked; the unshared regions are updated; and the partitioning modified so that regions which require further adaptation are not partitioned, and are therefore free to be improved. The author emphasises how this parallelisation strategy can be applied without modification to a serial mesh optimisation algorithm. Load balancing is achieved with the ParMETIS graph partitioning library (Karypis and Kumar, 1999). In Gorman et al. (2009), scaling results up to 1024 processors are presented; the efficiency of the parallel adaptive algorithm for a three-dimensional fluid dynamics problem was 0.6 for 1024 processors, relative to the time taken on 64 processors.

The method presented in Lipnikov and Vassilevski (2003) has several features which would appear to hamper its scalability. Firstly, the entire mesh is made known to every process, rather than each storing a part. Secondly, the decomposition is serialised onto the root process; only the remeshing part is truly parallelised. Thirdly, the mesh is re-gathered onto the root processor in between decomposition and remeshing iterations. The largest parallel examples shown run on 8 processors. The authors comment that for 8 processors, communication starts to dominate computation.

Lepage et al. (2004, 2006) discuss the trade-offs in designing a parallel algorithm for a distributed-memory machine, rather than a shared-memory machine. The approach presented also locks the interface between meshes while mesh optimisation iterations take place. ParMETIS is used for load-balancing. An interesting feature of the algorithm presented is that node relocation operations (which in this algorithm are the last mesh optimisation operation performed) are terminated if any of the processors has finished its adaptive step. This premature termination improves parallel scalability, at the cost of poorer-quality meshes. The entire background mesh is stored on every process to facilitate interpolation of the metric during the adaptive step. The meshes used in the CFD solver employ prisms on walls with no-slip boundary conditions; special attention is paid to the preservation of these. The examples shown demonstrate a parallel efficiency of approximately 0.68 on 8 processors.

Tremel et al. (2007) describe the parallel implementation of a local remeshing algorithm. Like Alauzet et al. (2006a), the set of elements to be modified is transferred between processors so that each continuous set lies entirely on one process; then these may be remeshed in parallel. However, the element sets to be remeshed may be arbitrarily large, as they are cavities wherein the elements are of insufficient quality. The cavities are remeshed with the Delaunay triangulation algorithm of Weatherill and Hassan (1994). The example presented runs on 24 processors.

Finally, the parallelisation described in Park and Darmofal (2008) combines features of both Freitag et al. (1999) and Alauzet et al. (2006a). Like Alauzet et al. (2006a), the optimisation operations are performed in two sweeps: first those whose effect is entirely local, and then those which require communication. Like Freitag et al. (1999), these communication-requiring operations are performed by colour given by a graph colouring algorithm. After the adaptive remeshing, ParMETIS is used to rebalance the load.

It appears that there is little agreement on the optimal approach to the parallelisation of adaptive remeshing. Unfortunately, a quantitative analysis is difficult because few papers discussing parallelisation present scaling results beyond 64 processors. The author is unaware of any published papers demonstrating parallel scaling comparable to the excellent results recently presented in the hierarchical refinement community (Burstedde et al., 2008). A set of community benchmarks, designed to allow the quantitative com-

parison of different approaches, would be a highly worthwhile exercise.

1.2.5.4 Interpolation

As mentioned previously, the application of adaptive remeshing divides naturally into three sub-problems. The first, discussed in §1.2.3, is how to generate a mesh matching a given sizing specification. The second, reviewed in §1.2.4, is how to define the sizing specification from an approximate numerical solution. The third, discussed here, is how to interpolate any necessary data from the previous mesh to the adapted one.

This problem has received less attention from the adaptive remeshing community, with collocation Lagrange interpolation (interpolation by basis function evaluation; see §2.3.1) almost universally used. Many papers in the adaptive remeshing literature do not even mention its use.

There are several good reasons for this. The drawbacks of collocation interpolation can be summarised as having suboptimal interpolation error, its unsuitability for discontinuous fields, and its lack of conservation. Firstly, for stationary problems, the interpolated solution is only used as an initial guess for the next solve, so any errors introduced in the interpolation have a minimal effect. Secondly, even for transient simulations, the interpolation error introduced is often acceptably low, provided the adapted mesh is suitable for the representation of the data. Thirdly, its unsuitability for discontinuous solutions and its loss of conservation are unimportant for the majority of applications of adaptive remeshing.

Nevertheless, there are good reasons to consider the mesh-to-mesh interpolation problem. Firstly, computing the interpolation with optimal accuracy in the L_2 norm is an interesting mathematical question in its own right. Secondly, Lagrange or Lagrange-like interpolation is unsuited to discontinuous Galerkin methods, which are increasingly popular. For these cases, Lagrange interpolation is not defined, and the averaging inherent in Lagrange-like pseudo-interpolation operators is diffusive and cannot exploit discontinuous functions in the target function space. Thirdly, Lagrange interpolation is inherently nonconservative, which is a key requirement for the discretisation of certain problems. Without a conservative interpolation operator available, adaptive remeshing cannot be applied to these problems. As the development of optimally accurate, conservative interpolation oper-

ators is the subject of chapters 2, 3 and 4, a discussion of the literature relevant to conservation through interpolation is deferred to these chapters. This discussion focuses on interpolation in the context of adaptive remeshing, whereas §2.1 discusses interpolation in other contexts, in particular ALE and semi-Lagrangian methods.

The standard method, collocation interpolation, consists of evaluating the previous solution at the locations of the nodes in the adapted mesh, and taking these values as the coefficients of the associated shape functions. As basis function evaluation is trivially available for any finite element method, the only difficulty is the problem of mesh association: the identification of which basis functions to evaluate for a given node in the adapted mesh, i.e. to identify in which element of the previous mesh each node of the adapted mesh lies. The relevant element is referred to as the parent element of the node.

Peraire et al. (1993) discuss interpolation between meshes in the context of non-nested multigrid methods. The authors observe that Galerkin projection is optimal in the L_2 norm, note that its assembly necessitates computing the inner products of the basis functions of both meshes, and comment that this computation is very difficult because the basis functions are defined on different supports. No mention of mesh intersection is made; however, the authors demonstrate that if the inner products are approximated with numerical quadrature on the donor mesh, the resulting approximate projection is still conservative. Despite this conservation property, the use of this procedure to compute the inner products is discouraged as it is very inaccurate.

Löhner (1995a) discusses the mesh association problem in detail. The author discusses brute-force searching, methods of subdividing space, and develops an advancing-front vicinity searching algorithm. The algorithm exploits the connectivity of the target and donor meshes. Since adjacent nodes in the target will lie in nearby elements in the donor mesh, the algorithm uses the parenthood information for nodes which have already been interpolated to provide clues for the search for the parent of unprocessed nodes.

George and Borouchaki (1998) discuss the necessity of solution interpolation after adaptive remeshing, note the non-conservative character of collocation interpolation (see §2.3.1), and propose the use of the Galerkin projection

from mesh to mesh by means of mesh intersection. Galerkin projection is the optimally accurate projection in the L_2 norm, and is conservative, but its implementation is very difficult. The fundamental reason for this difficulty is that the method requires the computation of the inner products of the basis functions of the two meshes. In order to compute these exactly, the supermesh of the two meshes must be constructed (see §2.2), which is quite involved. (Standard numerical quadrature approaches are inadequate; see §3.4.) Although they comment that in their experience this provides a satisfactory algorithm for solution transfer, they give no examples. The reader is referred to a technical report by R. Ouachtaoui to be published in 1997 for further discussion; it appears, however, that this technical report was never published. Geuzaine et al. (1999) also discuss the Galerkin projection between two-dimensional meshes; however, rather than integrating over the supermesh, the integrals appear to be computed over the target mesh. This is less accurate than assembling over the supermesh, and therefore should be referred to as an approximate Galerkin projection. A similar approach is taken by Parent et al. (2008).

A restricted implementation of Galerkin projection was applied to mesh optimisation in Remacle et al. (2006). The difficulty of assembling the Galerkin system is circumvented by tightly coupling the optimisation operations and the projection. Since for each operation, the small patch of elements affected by this operation is known, the projection can be computed without any mesh association. This localisation is only possible if the associated function space is discontinuous, for otherwise the solution values in the patch are coupled to solution values outside it. It is not clear from the paper on what mesh the Galerkin system is assembled: on the old patch, the new patch, or their supermesh. No mention of element intersection is made.

El Hraiech et al. (2005) describe the desirability of the Galerkin projection for structural analysis, but comment that the construction of the supermesh was not yet feasible. Therefore, the integral is performed over a subdivision of the target mesh, with the hope being that computing the inner products of the basis functions on the refined target mesh is sufficiently accurate to assemble a useful Galerkin system. However, the basis functions of the donor mesh are (in general) discontinuous piecewise polynomials over any given element of the target mesh. A similar scheme is evaluated in §3.4. It

was found that the inner products were unacceptably inaccurate, even at the maximum refinement level everywhere. Therefore, this scheme is likely to be of limited use.

Davies et al. (2007) employ a cubic interpolation operator to transfer data between meshes in an adaptive remeshing scheme. The authors comment that linear interpolation fails to accurately interpolate features at a density interface in a thermochemical mantle convection simulation, and that its diffusive and nonconservative character degrades the quality of the solution. However, Davies et al. (2007) incorrectly claim that the cubic interpolation operator is conservative; it should be more accurately described as consistently conservative, i.e. conservative in the limit of target mesh refinement.

Various authors have developed special interpolation algorithms which preserve some desired properties of the interpolant, typically the divergence constraint on the velocity in fluid simulations. In the context of model coupling, Chippada et al. (1998) and Carey et al. (2001) enforce the divergence constraint on the interpolated velocity field by post-processing an interpolated field. This process is analogous to a familiar pressure projection algorithm and requires the solution of a Poisson equation. Balsara (2001) develops a divergence-free reconstruction algorithm for meshes produced from hierarchical refinement, where there exists an integer refinement ratio between the meshes. Bochev and Shashkov (2005) recover a vector potential from the velocity field on the previous mesh, which is then interpolated. The discrete curl operator on the adapted mesh is then applied to yield the divergence-free interpolant.

If the boundary of the domain is modified during the adaptive remeshing procedure (see §1.2.5.5), then the interpolation procedure must take this into account. (If any node of the adapted mesh lies outside the previous mesh, then it is more properly referred to as an extrapolation procedure.) Löhner (1995a) discusses this case, but this extension of the interpolation procedure is insufficient if the underlying space of the meshes differs in more drastic ways, such as if the meshes are different discretisations of a non-flat 2-manifold in $\mathbb{R}^3$. This problem frequently arises in domain decomposition methods, such as fluid-structure interaction problems where the fluid and structure subproblems are solved separately. Significant effort has been invested in the development of mesh association algorithms for such cases. Additionally, it appears that the interpolation algorithms used in surface

projection methods are more advanced than the interpolation algorithms typically used in adaptive remeshing, as conservation of the boundary conditions applied is often important for these applications.

Maman and Farhat (1995) propose a scheme for matching points between meshes by defining the *associate* of a point to be its projection onto the target mesh in the direction normal to the element. However, as shown in Jiao et al. (1999), this is not always well-defined. In Jiao et al. (1999), the associate of a point is defined to be the nearest point in the target mesh. van Brummelen (2008) comments that while this allows for a very efficient implementation, it causes difficulty for higher-order discretisations as the association is insufficiently smooth. Jiao and Heath (2004b) propose the use of the averaged normals to define the associate of a point; this is applied to constructing supermeshes for surface mesh data transfer in Jaiman et al. (2006). However, this association technique is specific to piecewise-linear geometrical representations. van Brummelen (2008) discuss the extension of the averaged-normal projection to higher-order geometrical representations. Rather than averaging, as in Jiao and Heath (2004b), the normal vector field is smoothed by the solution of a modified Helmholtz equation. The equation is solved using the same order that represents the geometry; this guarantees that the approximate solution is sufficiently smooth to transfer data between the higher-order meshes.

Farhat et al. (1998) discuss a conservative load transfer algorithm for fluid-structure interactions, using the point association algorithm of Maman and Farhat (1995). Data transfer from the structural computation to the fluid computation is performed with collocation interpolation, while an approach very similar to Galerkin projection is developed for transfer from the fluid computation to the structural computation.

It appears that both Heinstein and Laursen (2003) and Jiao and Heath (2004b) independently developed the natural extension to this algorithm, which is to assemble the mixed mass matrix over the supermesh. The supermesh is constructed by projecting the points of one surface onto their associate on the other and intersecting the resulting meshes. Both of these implementations only deal with two-dimensional surface meshes.

Farrell et al. (2009) was the first to present the application of supermeshing to adaptive remeshing, and the first to describe a bounded variant of the Galerkin projection. Since the development of the algorithms de-

scribed in this thesis, a technical report from INRIA has been published on the application of supermeshing to two-dimensional simulations (Alauzet and Mehrenberger, 2009). The projection algorithm described is not the optimally-accurate Galerkin projection and is specific to piecewise linear fields. Rather than computing the inner products of the basis functions, the integral of the field and its gradient is computed for each element of the target mesh. This computation is achieved through the construction of the supermesh. The nodal values of the solution are then obtained by averaging the elemental information.

The mesh association literature described above associates nodes of the target mesh with elements of the donor mesh, which is the natural problem to solve for collocation interpolation. However, for Galerkin projection, the natural association is that between elements of the target and donor meshes. An algorithm for computing such an association is given in chapter 4.

1.2.5.5 Boundary treatment

One aspect of preprocessing required by most numerical discretisations is the approximation of the true domain $\hat{\Omega}$ on which the partial differential equation is posed by a domain Ω on which an approximate solution may be computed (Strang and Fix, 1973). When the mesh is adapted, there are two main choices available for dealing with how Ω conforms to $\hat{\Omega}$: to retain the initial discrete geometry, or to modify the adapted mesh to conform to a different (and hopefully better) approximation to $\hat{\Omega}$, Ω'.

The first choice, which retains the initial geometry, is the simplest as it requires no integration between the remesher and the source of geometrical information. This is the approach used in all the examples of this thesis. This choice makes sense where an optimised representation of the domain has been computed as a preprocessing step and it is desired that this representation is retained throughout (e.g., Gorman et al. (2006)). This also simplifies the post-adaptive interpolation considerably as conservation of constant functions is possible because the volume has not changed. Also, no extrapolation is necessary.

The second approach, which is to modify the adapted mesh to better conform to $\hat{\Omega}$, is appropriate where the geometry of $\hat{\Omega}$ is available from a CAD system. This is particularly important for p-refinement, where convergence

can stall as p is increased if the order of approximation of the geometry is not also increased (Luo et al., 2002; Szabó et al., 2004). This also allows for the coarsening of geometrical details in regions where coarse meshes are desired for efficiency reasons: if the initial geometry is to be retained, then the mesh can only be coarsened up to the co-planar surfaces describing it. If the region is refined later in the simulation and the geometrical description is required, it can be re-acquired from the CAD data available. A series of papers by Shephard and co-workers describes the details of how mesh optimisation operations may be modified to consider the geometrical surface description (Li et al., 2003; Shephard et al., 2005; Li et al., 2005). Alternatively, an approximation to the analytical surface can be computed from the given initial boundary description by a surface reconstruction method and the adapted mesh fitted to this (Lipnikov and Vassilevski, 2005). Hughes and co-workers have introduced the concept of isogeometric analysis, where the CAD geometry is represented exactly in the analysis by NURBS basis functions; furthermore, following the isoparametric approach, the solution space for the prognostic variables is chosen to be the same space which represents the geometry (Hughes et al., 2005). Thus, the geometry is represented exactly, even on the coarsest discretisation. It remains to be seen whether this approach will influence future directions in adaptive remeshing.

The integration of the mesh optimisation procedure with CAD data is not currently available, and is deferred to future work.

1.3 Contributions of this thesis

1.3.1 Novel research

This thesis presents several novel results:

- The first three-dimensional implementation of Galerkin projection (§3.3).
- The first application of supermeshing to adaptive remeshing (§2.4.3).
- The development of *a posteriori* error estimates for Galerkin projection (or any supermesh-based projection) (§2.3.3.2).
- The development of a bounded variant of the Galerkin projection for piecewise linear fields (§2.3.4).

- The development of a novel element-element association algorithm to efficiently identify pairs of intersecting elements (chapter 4).
- The observation that the supermesh forms a common discrete superspace of the function spaces associated with the input meshes (§5.2).
- The development of an algorithm for computing suitable meshes for common interpolation (§5.3).
- The first algorithm for directional integration on fully unstructured meshes (§5.4.1.3). This is particularly important for ocean modelling, where vertical integrals are key diagnostic quantities to be computed from the solution.

1.4 Some common notation

d dimension, $d \in \{1, 2, 3\}$

Ω a polyhedral d-dimensional domain, $\Omega \subset \mathbb{R}^d$

μ the d-dimensional Lebesgue measure (length, area, or volume)

$\mathcal{T}_A$ a mesh of Ω indexed by A

ϕ_A a basis function associated with $\mathcal{T}_A$

$\phi_A^{(k)}$ the k^{th} basis function associated with $\mathcal{T}_A$

Φ_A the set of basis functions associated with $\mathcal{T}_A$, $\Phi_A = \left\{\phi_A^{(k)}\right\}_k$

$\mathcal{V}_A$ the function space associated with $\mathcal{T}_A$

K_A an element of $\mathcal{T}_A$

$|\mathcal{T}_A|$ the number of elements in $\mathcal{T}_A$

$\mathcal{N}_A$ the set of nodes of $\mathcal{T}_A$

$\mathcal{F}_A$ the $(d-1)$-dimensional facets of $\mathcal{T}_A$ (points, edges, or faces)

$\mathcal{T}_D$ in interpolation problems, the donor mesh

$\mathcal{T}_T$ in interpolation problems, the target mesh

Π_{TD} an interpolation operator from $\mathcal{T}_D$ to $\mathcal{T}_T$

$\mathcal{T}_S$ the supermesh of $\mathcal{T}_D$ and $\mathcal{T}_T$

$\mathcal{T}_S^K$ the fragment of the supermesh associated with element K

$\mathcal{T}_P$ a parent mesh (of the supermesh), either $\mathcal{T}_D$ or $\mathcal{T}_T$

q_D a function in $\mathcal{V}_D$ to be interpolated

q_T its interpolant in $\mathcal{V}_T$

$q_D^{(k)}$ the coefficient of q_D associated with $\phi_A^{(k)}$

η the error in the interpolation, $q_D - q_T$

M_T the mass matrix associated with $\mathcal{V}_T$, the Gram matrix of Φ_T

M_T^L the lumped mass matrix associated with $\mathcal{V}_T$, obtained by row-summing M_T

M_{TD} the mixed mass matrix mapping from $\mathcal{V}_D$ to $\mathcal{V}_T$

$\mathcal{M}$ a metric field encoding a mesh

Chapter 2

Conservative and bounded interpolation operators

Abstract

Adaptive remeshing on unstructured meshes is a popular tool for reducing the computational cost of numerical simulations. Unstructured meshes are often preferred in mesh adaptivity as they allow for greater geometric flexibility and arbitrary anisotropy in resolving simulation features. However, such mesh adaptivity suffers from a significant drawback: the interpolation errors caused by interpolating from the old mesh to the new mesh typically destroy conservation of quantities important to the physical accuracy of the simulation (e.g., density, volume fraction, tracer concentration). This work presents several novel interpolation operators between general unstructured meshes via the construction of an intermediate supermesh. Particular attention is paid to the development of a novel bounded conservative interpolation operator. Additionally, the presented interpolation operators are well defined in the case where the basis functions of the target and/or donor meshes are discontinuous, a significant advantage over collocation interpolation. The performance of the conservative interpolation operators are compared against collocation interpolation using the underlying basis functions.

This chapter is derived from and expands upon
Farrell et al. (2009) and Farrell and Maddison (2009).

Contents

2.1 Introduction

Mesh adaptivity algorithms enable a simulation to dynamically focus resolution where and when it is important, potentially allowing for large savings in computational costs for a given level of error (Morgan et al., 1991). However, one drawback of adaptive remeshing is the necessity of interpolating solution fields from the previous mesh to the newly adapted mesh. Such interpolation destroys conservation of important physical quantities such as density, volume fractions, or tracer concentrations.

Therefore, several conservative interpolation operators are proposed: interpolation operators that preserve the global integrals of the solution fields. To construct such an interpolation operator, an auxiliary *supermesh* is defined and constructed; the supermesh is a mesh that may be interpreted as the union of its parent meshes, the original mesh and the newly adapted mesh. By constructing a supermesh, the interpolation operators allow that the target and donor meshes may be unrelated, and one or both may have discontinuous basis functions. The algorithm requires no structural relationship between the two meshes (e.g., a hierarchical relationship), but can be trivially extended to exploit such a relationship if it is available.

The purpose of the supermesh is to facilitate the use of projection operators as conservative mesh-to-mesh interpolators. Along with Galerkin projection, a bounded, conservative, minimally diffusive interpolation scheme for the special case of linear elements is developed.

This chapter is laid out as follows. A review of previous work in this area follows in the remainder of this section. The definition of a supermesh is given in §2.2. Its use in conservative interpolation is explained in §2.3. The problem of constructing the supermesh is deferred to chapter 3. Section §2.4 discusses some numerical examples demonstrating the interpolation operators introduced. The chapter closes with some conclusions.

2.1.1 Background to conservative interpolation

Preservation of conservation properties under discretisation is absolutely vital in some application areas. For example, in numerical weather prediction and climate simulation, statistics are sought rather than deterministic solution trajectories; preserving the conservation of appropriate quantities is known to improve the long time scale accuracy of results (Arakawa and Lamb, 1981; Sadourny, 1975; Ringler and Randall, 2002; Cullen, 2007; Thuburn, 2008). Gear (1992) states that the failure to maintain certain invariants can lead to physically impossible solutions. As the conservative discretisation forces the discrete solution onto a manifold on which the analytical solution lies, conservative lower-order schemes can be more accurate than nonconservative higher-order schemes; de Frutos and Sanz-Serna (1997) present such an example. Discretisations which preserve qualitative structures of the analytical solution are discussed in Budd and Piggott (2003).

The natural algorithm for interpolating from a donor mesh to a target mesh, called collocation interpolation, is to evaluate the numerical solution defined on the donor mesh at each node of the target mesh using the basis functions of the donor mesh (Löhner, 1995a). Collocation interpolation is known to erode minima and maxima, and is not conservative (Farrell et al., 2009). If the target mesh is discontinuous, collocation interpolation ignores the possibility of exploiting discontinuities in the output to better represent the field: the output is always continuous. If the donor mesh is discontinuous, this procedure is not well-defined, as nodes of the target mesh may lie along discontinuities of the solution on the donor mesh, and so the value of the solution at a physical point in space is not uniquely defined. While it is possible to ignore this difficulty with a pseudo-interpolation operator, the averaging inherent in this procedure renders it unsuitable for use in discontinuous numerical simulations.

The purpose of this chapter, or indeed this thesis, is not to discuss conservative discretisation methods for PDEs. Rather, the assumption here is that such a method is available and one wishes to couple this with a mesh adaptivity algorithm which makes necessary the use of mesh-to-mesh interpolation.

Some of the earliest work on conservative mesh-to-mesh interpolation grew out of the development of arbitrary Lagrangian-Eulerian (ALE) methods (Hirt et al., 1974). ALE methods make Lagrangian timesteps where mesh nodes and elements are advected with the flow, and then periodically this mesh is manipulated (rezoned) into a more optimal configuration to remove distortion or skewness. Solution variables are then interpolated (remapped) between configurations. This rezoning and remapping procedure can be applied continuously at every timestep, or only occasionally, governed by how distorted the mesh has become. An advantage of rezoning at every timestep is that the configuration of the mesh before and after rezoning can be assumed to represent only a small perturbation. This allows the remapping to be performed via a local operation where solution variables are exchanged through cell faces. However, this may necessitate a timestep restriction. A conservative interpolation step in an ALE algorithm is presented in Margolin and Shashkov (2003) which is based on partitioning cells of the updated mesh into components of cells from the old mesh and 'swept regions' from neighbouring cells, with material then fluxed between cells. Alternatively,

this interpolation step can be recast as a parameterised advection problem, where the wide literature on flux limiters and correctors may be applied to design interpolation schemes which minimise spurious dissipation and overshoots (Smolarkiewicz and Rasch, 1991). In Garimella et al. (2007), a conservative algorithm for polyhedral meshes is described, and in Kucharik and Shashkov (2007) a local algorithm is proposed that is able to deal with changing connectivity in Voronoi meshes.

Global (or integral) remapping assumes no linkage between target and donor meshes. Therefore, it is a more complex and costly problem but has the advantages of allowing complete flexibility in the two meshes, and in principle allows rezoning to be performed less often, freeing the method from the timestep restrictions imposed. Methods in this class typically rely on being able to calculate the volume of the intersection of old and new cells. In Dukowicz (1984), a conservative interpolation method is proposed that assumes piecewise constant fields and simplifies the problem of computing the volume of intersection of old and new cells into a surface integral by invoking the divergence theorem. However, the first order nature of the algorithm leads to excessive diffusion. In Dukowicz and Kodis (1987) the approach is extended to higher order to improve its diffusive characteristics.

Bailey (1987) presents an approximate implementation of Galerkin projection in two dimensions for triangular meshes. The right hand side of the Galerkin system (2.18) is approximated on the target mesh by assuming the solution is piecewise constant over the area of intersection. This algorithm is similar to the more widely-known algorithms of Franklin et al. (1994) and Grandy (1999), in that it only uses the volume of the polygon of intersection, as opposed to performing the computation of the inner products exactly by meshing the polygon of intersection. The approximation retains the conservative nature of the Galerkin projection, but the accuracy of the scheme will be affected. The author also proposes a bounded variant of Galerkin projection for piecewise linear fields, which exploits the fact that if the mass matrix in (2.18) is lumped by row-summing, the resulting projection is bounded (lemma 2.4). The approximate Galerkin projection is used where the solution is naturally bounded, while the lumped mass solution is used where the Galerkin projection exhibits overshoots and undershoots.

Semi-Lagrangian methods are another popular class of numerical method where an interpolation step is crucial (Staniforth and Côté, 1991). Here an

advection step is recast in Lagrangian form as a trajectory calculation, followed by an interpolation of data from the old mesh to trajectory departure points. Parameterised advection can be used at this stage to preserve monotonicity properties (Smolarkiewicz and Rasch, 1991). In Scroggs and Semazzi (1995), cell centred values in a semi-Lagrangian method are conserved through the use of a first order conservative interpolation scheme based upon weighted averages on a quadrilateral mesh. Phillips and Williams (2001) extend this work with schemes that are of higher order and reduce issues with excessive numerical diffusion. In Iske and Käser (2004), a conservative semi-Lagrangian method on unstructured and adaptive meshes is described. The method combines a particle-based semi-Lagrangian approach with a finite volume scheme on adaptive unstructured Voronoi meshes. An intersection algorithm on two-dimensional Voronoi cells is used to ensure a conservative scheme. A timestep restriction is required to ensure that all cells being intersected are convex.

Chesshire and Henshaw (1994) discusses conservative interpolation of fluxes between overlapping structured meshes. The interpolation coefficients are assumed to be free parameters; then, constraints are derived on the coefficients to ensure that the interpolation is conservative. Grandy (1999) discusses conservative remapping through the calculation of the volume of the intersection of overlapping polyhedra. The remapping algorithm used is first order and assumes the field to be constant in the donor element. The algorithm differs from that of Dukowicz (1984) in the manner in which the volume of the intersection is computed. (Very similar algorithms were developed by Bailey (1987) and Franklin et al. (1994), although Grandy appears to be unaware of these works.) A general overview of methods for calculating whether polyhedra intersect and constructing that intersection is given in Mount (1997). George and Borouchaki (1998) proposed the use of Galerkin projection for mesh-to-mesh interpolation in the context of unstructured adaptive meshes. This work was the first to note that the Galerkin projection can be computed exactly by computing the inner products over the region of intersection. They mention that from their experience this gives a suitable answer to the solution transfer problem, in terms of conservation and accuracy, but give no examples.

In Jiao and Heath (2004a) the term common-refinement is used as a synonym for the supermesh of two meshes which data is being transferred

between. They compare several methods for transferring data once the common-refinement has been constructed. These include standard pointwise interpolation using the underlying basis functions, cubic spline interpolation, area weighted averaging, L_2 (Galerkin) minimisation, and minimisation using an H_1 Sobolev norm. Comparisons in one dimension and two dimensions on uniform quadrilateral grids show that the common-refinement based scheme with L_2 minimisation on linear basis functions has errors which grow more slowly with iteration number than cubic interpolation, and so for repeated data transfers between grids it is more accurate than the non-conservative method. This paper also deals with data transfer between surface meshes that do not necessarily coincide. In one dimension projection via an appropriate Sobolev norm is shown to introduce a smoothing effect which assists in reducing undershoots and overshoots which may be present with L_2 minimisation.

Rüter et al. (2007) discuss the transfer of solution gradients between primal and dual meshes via a weighted average approach in two dimensions and use the term supermesh for the intersection between the two meshes over which the transfer takes place. However, they do not actually discuss supermesh construction; instead, they assume that the two meshes are hierarchically related, and hence the computation of the supermesh is trivial. In the adaptive approach for discontinuous methods described in Remacle et al. (2005), a local interpolation problem is solved at each mesh modification, ensuring conservation (Remacle et al., 2006). Alauzet (2008) mentions the importance of conservative interpolation in the simulation of the Euler equations. Recently, a similar approach to interpolation by supermeshing has been proposed for two-dimensional interpolation between triangular meshes (Alauzet and Mehrenberger, 2009).

To the best of the author's knowledge, this work (presented in Farrell et al. (2009)) is the first to present a minimally diffusive bounded interpolation algorithm between unrelated unstructured meshes.

In this work, attention is confined to volume meshes of the domain $\Omega \subset \mathbb{R}^d, d \in \{2, 3\}$ which are comprised of convex polytopes. For a discussion of conservative interpolation between two-dimensional surface meshes in $\mathbb{R}^3$ using a supermesh approach, the reader is referred to Jiao and Heath (2004b); Jaiman et al. (2006).

2.2 Supermeshes

Let $\mathcal{T}_D$, $\mathcal{T}_T$ be two (arbitrarily unstructured) volume meshes of the same polyhedral domain $\Omega \subset \mathbb{R}^d$, with nodes $\mathcal{N}_D, \mathcal{N}_T$, and edges $\mathcal{F}_D, \mathcal{F}_T$ respectively. K will refer to an element of a mesh. Define an edge of an element in a d-dimensional mesh to be a $(d-1)$-dimensional surface facet (in 2 dimensions, lines; in 3 dimensions, faces). In this work, attention is confined to linear geometric simplices, i.e. the shape functions used to represent positions are linear. Ω is assumed to be polyhedral; the extension to curved boundaries presents additional complexities which will be considered in future work.

Definition 2.1. *Define a* supermesh $\mathcal{T}_S$ *of* $\{\mathcal{T}_D, \mathcal{T}_T\}$ *as a mesh of* Ω *such that:*

- $\mathcal{N}_S \supseteq \mathcal{N}_D \cup \mathcal{N}_T$*;*
- $\mu(K_S \cap K) \in \{0, \mu(K_S)\} \ \forall \ K_S \in \mathcal{T}_S, K \in \mathcal{T}_P, \mathcal{T}_P \in \{\mathcal{T}_D, \mathcal{T}_T\}$*;*

where μ *is the* d*-dimensional measure function (length, area or volume).*

In words, the first requirement states that any node in a parent mesh must be present in the supermesh. The second requirement states that for every element in the supermesh, the intersection of that element with any element of a parent mesh must either be a set of measure zero, or the whole element.

The existence of such a supermesh is proved by construction in chapter 3. Clearly, a supermesh is not unique.

The utility of a supermesh $\mathcal{T}_S$ is that it provides a decomposition of elements in $\mathcal{T}_D$ and $\mathcal{T}_T$ as elements in $\mathcal{T}_S$. This is encoded in the following result (figure 2.1).

Lemma 2.1. *For every element* K_S *in a supermesh* $\mathcal{T}_S$ *of input meshes* $\{\mathcal{T}_D, \mathcal{T}_T\}$*, in each parent input mesh* $\mathcal{T}_P \in \{\mathcal{T}_D, \mathcal{T}_T\}$ *there exists exactly one element with an intersection of nonzero measure with* K_S*. This element is called the* parent *element of* K_S *in* $\mathcal{T}_P$*.*

Proof. By assumption, $\mathcal{T}_S$ meshes the same domain Ω as $\mathcal{T}_P$. Therefore, for a given element K_S, its measure may be written as

$$\mu(K_S) = \sum_{K_P \in \mathcal{T}_P} \mu(K_S \cap K_P), \tag{2.1}$$

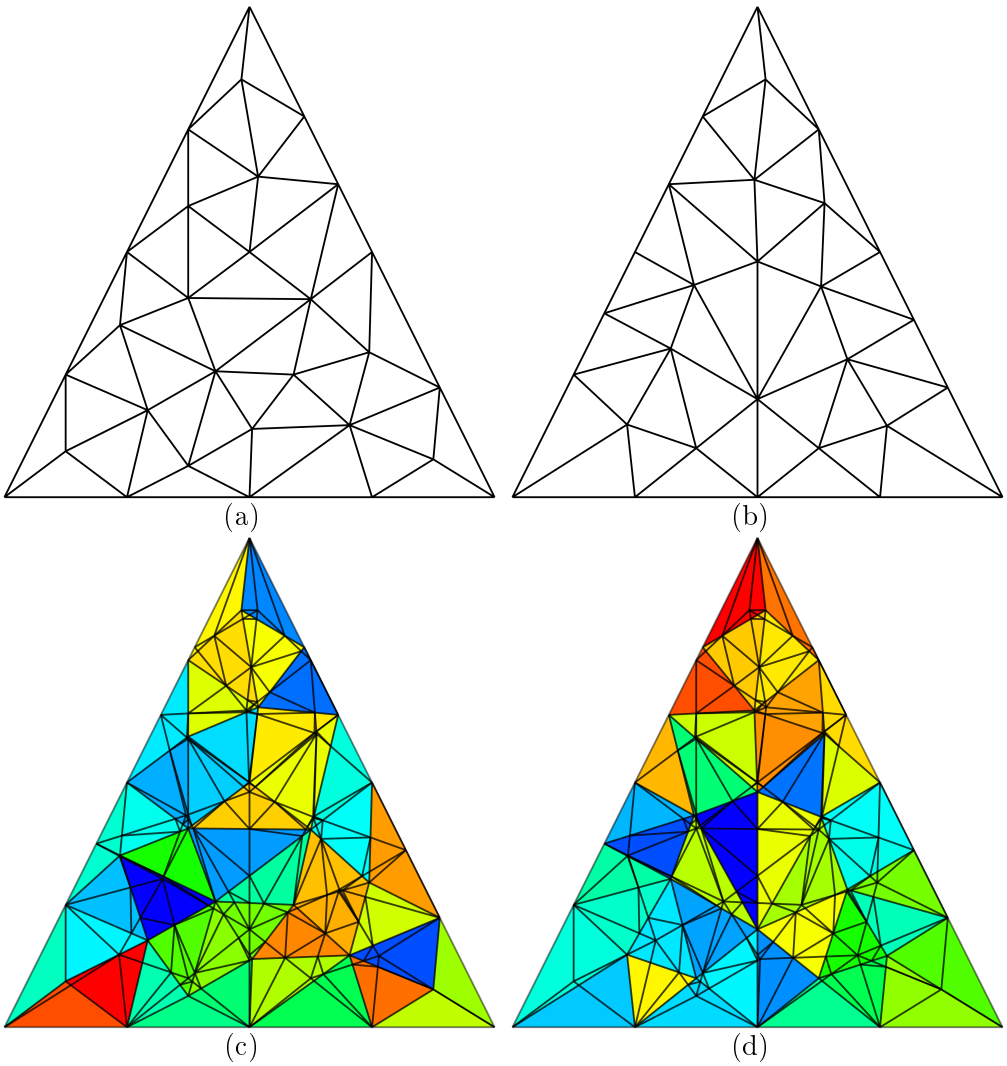


Figure 2.1: Two meshes and their supermesh. (a) Input mesh $\mathcal{T}_D$. (b) Input mesh $\mathcal{T}_T$. (c) A supermesh $\mathcal{T}_S$ of $\mathcal{T}_D$ and $\mathcal{T}_T$, coloured with the mapping χ_{SD} (equation (2.5)). (d) The same supermesh, coloured with the mapping χ_{ST}. The colours in (c) and (d) identify the parent elements of each element in the supermesh. Note that each element of the supermesh is completely contained within an element in each parent mesh (lemma 2.1).

where K_P is an element in the parent mesh. Suppose there exist K_{P_1}, K_{P_2} such that $\mu(K_S \cap K_{P_1}) \neq 0, \mu(K_S \cap K_{P_2}) \neq 0$. Then by the second property of definition 2.1,

$$\mu(K_S \cap K_{P_1}) = \mu(K_S), \tag{2.2}$$

$$\mu(K_S \cap K_{P_2}) = \mu(K_S). \tag{2.3}$$

However, this contradicts (2.1): the right hand side of equation (2.1) is greater than the left hand side. Therefore there can be at most one $K_P \in \mathcal{T}_P$ with intersection of nonzero measure. A similar argument shows that there must be at least one intersection of nonzero measure. □

The size of the supermesh may be estimated as follows. The intersection of each intersecting pair of elements from the target and donor meshes must be representable as the union of supermesh elements; each intersection must in turn be triangulated to form a mesh over which quadrature may be performed. The number of intersections k is bounded:

$$\min(|\mathcal{T}_D|, |\mathcal{T}_T|) \leq k \leq |\mathcal{T}_D||\mathcal{T}_T|. \tag{2.4}$$

In two dimensions, if the elements of the input meshes are convex polygons of n vertices, then the intersection is a convex polygon of at most $2n$ vertices (Mount, 1997). A convex polygon of $2n$ vertices may be minimally triangulated into $2n - 2$ triangles. In three dimensions, the problem is significantly harder. Given two tetrahedra, the intersection has at most 8 faces (Preparata and Shamos, 1985, theorem 7.2). A 3-polytope with 8 faces can have at most 12 vertices (Seidel, 2004, page 499). Computing the size of the minimal triangulation of a convex polyhedron is NP-complete (Below et al., 2004). However, the size of the minimal triangulation of a polyhedron with n vertices is bounded above by $\binom{n}{2} - 2n + 3$ (Edelsbrunner et al., 1990), where $\binom{n}{2} = \frac{n!}{2!(n-2)!}$. Therefore, the number of elements of the supermesh in the worst case is bounded above by $C_d|\mathcal{T}_D||\mathcal{T}_T|$, with $C_2 = 4$ and $C_3 = 45$. For most practical pairs of meshes, this bound is very pessimistic.

Let $\mathcal{P}(\mathcal{T}_S)$ denote the power set of $\mathcal{T}_S$, the set of all subsets of elements in the supermesh. For the purposes of a conservative interpolation algorithm between a donor mesh $\mathcal{T}_D$ and a target mesh $\mathcal{T}_T$, the following maps are constructed (see figure 2.1):

- $\chi_{SD} : \mathcal{T}_S \to \mathcal{T}_D$, taking an element in $\mathcal{T}_S$ to its parent element in the donor mesh; and
- $\chi_{ST} : \mathcal{T}_S \to \mathcal{T}_T$, taking an element in $\mathcal{T}_S$ to its parent element in the target mesh; and
- $\chi_{DS} : \mathcal{T}_D \to \mathcal{P}(\mathcal{T}_S)$, taking an element in $\mathcal{T}_D$ to its child elements in the supermesh; and
- $\chi_{TS} : \mathcal{T}_T \to \mathcal{P}(\mathcal{T}_S)$, taking an element in $\mathcal{T}_T$ to its child elements in the supermesh.

These maps are defined by

$$\chi_{SD}(K_S) = K_D \iff \mu(K_S \cap K_D) = \mu(K_S), \tag{2.5}$$

$$\chi_{DS}(K_D) = \{K_S \mid \chi_{SD}(K_S) = K_D\}, \tag{2.6}$$

with χ_{ST} and χ_{TS} defined similarly. The existence of these maps follows from lemma 2.1 above.

2.3 Interpolation

Let $\mathcal{T}_D$, $\mathcal{T}_T$ be two meshes as described above. In this work, $\mathcal{T}_D$ is to be the donor mesh, and $\mathcal{T}_T$ is the target mesh onto which data from $\mathcal{T}_D$ should be interpolated. Let $q_D \in \mathcal{V}_D$ be a function to be interpolated.

2.3.1 Collocation interpolation

The development of the novel interpolation operators is motivated by considering the obvious approach, collocation interpolation.

Collocation interpolation is the interpolation derived from the solution values of the donor mesh. For each node $n_T \in \mathcal{N}_T$ in the target mesh $\mathcal{T}_T$, a containing element K_D is identified in the donor mesh $\mathcal{T}_D$, and the solution q_D is evaluated at the physical location of the target node n_T. Such an element K_D may be identified by an advancing front algorithm (Löhner, 1995a) or by an R-tree spatial indexing algorithm (Guttman, 1984).

While cheap to implement, this algorithm suffers from several serious drawbacks:

- Conservation. In general, the integral of the interpolant on the target mesh is not the same as the integral of the field on the donor mesh. For some applications, such as for long-term geophysical fluid dynamics, conservation is crucial (Arakawa and Lamb, 1981; Cullen, 2007; Thuburn, 2008). While it is possible to enforce conservation via an *ad hoc, a posteriori* correction, such procedures have undesirable consequences for the quality of the solution (Takacs, 1988).

- Erosion of maxima and minima. In general, the minimum and maximum values of the field will be lost during collocation interpolation (Davies et al., 2007).

- Continuity. Discontinuous discretisation methods are becomingly increasingly popular. However, collocation interpolation is unsuitable for such methods as the solution values are not pointwise well-defined. As the donor mesh $\mathcal{T}_D$ is queried by physical location, the output will be continuous, assuming the solution on the donor mesh is even well-defined at that location. While some pseudo-interpolation operator may be applied, the continuity of the output precludes the exploitation of discontinuities, restricting the range of the interpolation operator to the continuous subspace of the function space. This loss of discontinuous information is particularly unfortunate in the case of adaptive remeshing; upon each adapt, all discontinuous information is lost.

2.3.2 The Grandy conservative interpolation operator

Let q_D be a function whose integral is to be conserved, i.e.

$$\int_\Omega q_D \, dV = \int_\Omega \Pi_{TD}[q_D] \, dV, \tag{2.7}$$

where Π_{TD} is the projection operator to be described. This projection operator was first described in Franklin et al. (1994), and independently rediscovered in Grandy (1999); a similar scheme (where the donor basis functions are assumed piecewise constant, but the target basis functions are not) was proposed in Bailey (1987). As the work of Grandy is the most widely known, this interpolation operator is herein referred to as the Grandy interpolation operator.

Equipped with the mappings defined in section §2.2, let us form a discrete version of equation (2.7) over the meshes $\mathcal{T}_D$ and $\mathcal{T}_T$:

$$\sum_{K_D \in \mathcal{T}_D} \int_{K_D} q_D \, dV = \sum_{K_T \in \mathcal{T}_T} \int_{K_T} \Pi_{TD}[q_D] \, dV. \tag{2.8}$$

K_T may be expressed as the union of its children elements:

$$\sum_{K_D \in \mathcal{T}_D} \int_{K_D} q_D \, dV = \sum_{K_T \in \mathcal{T}_T} \left(\sum_{K_S \in \chi_{TS}(K_T)} \int_{K_S} \Pi_{SD}[q_D] \, dV \right), \tag{2.9}$$

where Π_{SD} is the projection operator applied to the supermesh.

Now, it is clear from the definition of χ_{TS} that

$$\chi_{TS}(K) \cap \chi_{TS}(K') \neq \emptyset \iff K = K', \tag{2.10}$$

and so the problem of interpolating conservatively from $\mathcal{T}_D$ to $\mathcal{T}_T$ reduces to interpolating from $\mathcal{T}_D$ to $\mathcal{T}_S$ in a conservative manner, such that

$$\sum_{K_D \in \mathcal{T}_D} \int_{K_D} q_D \, dV = \sum_{K_S \in \mathcal{T}_S} \int_{K_S} \Pi_{SD}[q_D] \, dV. \tag{2.11}$$

Therefore, any supermesh interpolation operator Π_{SD} that conserves the integral (i.e., Π_{SD} satisfies (2.11)) induces a conservative interpolation operator Π_{TD} that satisfies equation (2.8) by means of the map χ_{TS}.

One such operator Π_{SD} that satisfies (2.11) is

$$\int_{K_S} \Pi_{SD}[q_D] \, dV := \omega_{K_S} \int_{\chi_{SD}(K_S)} q_D \, dV, \tag{2.12}$$

where

$$\omega_{K_S} := \int_{K_S} 1 \, dV \Big/ \int_{\chi_{SD}(K_S)} 1 \, dV. \tag{2.13}$$

ω_{K_S} defines the fraction of the integral of the parent element to contribute to the child element. Since with this definition of ω_{K_S} it holds that

$$\int_{K_D} q_D \, dV = \sum_{K_S \in \chi_{DS}(K_D)} \int_{K_S} \Pi_{SD}[q_D] \, dV \tag{2.14}$$

for every element $K_D \in \mathcal{T}_D$, conservation is retained. Π_{SD} thus induces a

conservative mapping Π_{TD} by the argument above.

Given the elemental integrals of the function q_D, the nodal values may be recovered by assuming that the function is constant over an element. This is the same assumption as made in Grandy (1999), and it will have severe consequences for the accuracy of the solution, as shall be seen in §2.4. Let

$$q_T|_{K_T} = \int_{K_T} q_D \, dV \; / \; \mu(K_T), \tag{2.15}$$

where $\mu(K_T)$ is the measure of the element. A standard Galerkin projection can then be then applied to convert the piecewise constant elemental values to a representation by piecewise linear basis functions.

The algorithm is summarised as follows.

- Compute the supermesh $\mathcal{T}_S$ and mappings χ_{SD}, χ_{TS} (equations (2.5) and (2.6)).
- For every $K_T \in \mathcal{T}_T$, compute its integral value as the sum of the integral values of its children elements (equations (2.12), (2.13)).
- Given the elemental integrals of the function q and assumption (2.15), compute its nodal values by means of a $P_0 \to P_1$ Galerkin projection.

By construction, this scheme is conservative, but its nodal accuracy is hampered by assumption (2.15). As explained in Grandy (1999), the scheme is first-order.

2.3.3 Galerkin projection

Let us consider equation (2.7) in a weak integral sense:

$$\int_\Omega q_D \phi_T^{(k)} \, dV = \int_\Omega q_T \phi_T^{(k)} \, dV, \tag{2.16}$$

for each basis function $\phi_T^{(k)}$ associated with mesh $\mathcal{T}_T$. This will conserve the integral if the constant function 1 is contained in the span of Φ_T, the set of basis functions associated with $\mathcal{T}_T$.

In this work, no strong Dirichlet boundary conditions are applied within the interpolation procedure. If strong Dirichlet boundary conditions are applied, the test space is modified so that the constant function 1 is not in the

span of the basis functions, and thus the projection will not be conservative. If Dirichlet boundary conditions are applied weakly, then the interpolation procedure developed within this work may be applied without modification. It may be possible to modify the test space in a manner analogous to Hubbard et al. (2009) so as to both conserve the integral and enforce strong Dirichlet boundary conditions.

Let $q_D \in \mathcal{V}_D$ represent a function to be interpolated, and let $q_T \in \mathcal{V}_T$ represent its interpolant on mesh $\mathcal{T}_T$. Now replace q_D and q_T with their finite element representations:

$$\int_\Omega \sum_i q_D^{(i)} \phi_D^{(i)} \phi_T^{(k)} \, dV = \int_\Omega \sum_j q_T^{(j)} \phi_T^{(j)} \phi_T^{(k)} \, dV, \tag{2.17}$$

for i and j ranging over the sets of basis functions Φ_D and Φ_T respectively, and $q_P^{(k)}$ representing the component of q_P associated with $\phi_P^{(k)}$. This gives rise to the matrix equation

$$M_T q_T = M_{TD} q_D, \tag{2.18}$$

where

$$(M_T)_{ij} = \int_\Omega \phi_T^{(i)} \phi_T^{(j)} dV, \tag{2.19}$$

and

$$(M_{TD})_{ij} = \int_\Omega \phi_T^{(i)} \phi_D^{(j)} dV. \tag{2.20}$$

M_{TD} represents a mixed mass matrix between meshes $\mathcal{T}_D$ and $\mathcal{T}_T$. Equation (2.18) may then be solved using a standard iterative solver to compute the nodal values of q_T.

The mixed mass matrix is assembled by means of decomposing elements of $\mathcal{T}_T$ into their child elements in the supermesh using the mapping χ_{TS} and computing the appropriate intersection integrals with the elements of the donor mesh given by χ_{SD}. There is no need to explicitly store M_{TD}; its action on q_D may be computed element-by-element by looping over the elements of $\mathcal{T}_T$. Note that since the polynomial degrees of Φ_T and Φ_D are known, the minimal order quadrature rule required to assemble the system exactly is also known.

Again, by construction, this scheme is conservative, and its nodal accuracy is not hampered by assumptions such as equation (2.15). However, it can

suffer from oscillations, as will be demonstrated later.

2.3.3.1 Optimality of the Galerkin projection

Galerkin projection is referred to as a projection because it is optimal in the L_2 norm, i.e.

$$||q_D - q_T||_2 = \min_{q \in \mathcal{V}_T} ||q_D - q||_2 \,. \tag{2.21}$$

The L_2 norm of $q_D - q$ is minimised if $\nabla ||q_D - q||_2 = 0$. Expanding the definition of the L_2 norm,

$$\nabla \int_\Omega (q_D - q)^2 \, dV = \; 0 \tag{2.22}$$

$$\implies \int_\Omega \frac{\partial}{\partial q^{(i)}} (q_D - q)^2 \, dV = \; 0, \; \forall \; i \tag{2.23}$$

$$\implies \int_\Omega 2\phi_T^{(i)} (q_D - q) \, dV = \; 0, \; \forall \; i \tag{2.24}$$

and the Galerkin projection described above is recovered.

2.3.3.2 *A posteriori* error computation

An elegant feature of supermesh-based interpolation is that the error between the donor function and its interpolant is exactly computable (ignoring roundoff).

As shown later in lemma 5.1, the supermesh provides a function superspace of the function spaces associated with the input meshes. This implies that the projection operator obtained by collocation interpolation,

$$\Pi_{SP} : \mathcal{V}_P \rightarrow \mathcal{V}_S \; , P \in \{T, D\} \tag{2.25}$$

is the identity operation, i.e. $\Pi_{SP}(q) = q$. By closure, the difference between two functions in $\mathcal{V}_S$ is also an element of $\mathcal{V}_S$, and therefore the projection error may be computed as

$$\eta = \Pi_{SD}(q_D) - \Pi_{ST}(q_T) = q_D - q_T. \tag{2.26}$$

The evaluation of Π_{SP} is trivial: since the parenthood mapping from each element in $\mathcal{T}_S^K$ is already stored to facilitate the construction of M_{TD},

no searching need be performed; only the evaluation of the parent basis functions is required.

The computation of η is exact, ignoring roundoff error in the evaluation of the projection operator Π_{SP}. Needless to say, this is a very attractive property.

As η is available as a function, any desired norm may be taken to quantify the error. Since the Galerkin projection is optimal in the L_2 norm, the L_2 norm is a sensible choice. If the projection were modified so that it were optimal in the H_1 norm, as in Jiao and Heath (2004a), then the H_1 norm should be chosen.

In practice, this *a posteriori* error computation is more useful for discontinuous fields, as q_T is computable element-by-element and therefore so is η. The *a posteriori* error computation can still be applied for continuous fields, but either the supermesh must be stored or recomputed, as q_T requires a global mass matrix solve and so the entire supermesh must be assembled before it is computable.

2.3.3.3 Numerical order of convergence

A numerical experiment was performed to investigate the observed order of convergence of the Galerkin projection. A given scalar field ζ is evaluated on the donor and target meshes. Although the donor and target meshes are topologically unrelated, they share the same characteristic mesh size h. The Galerkin projection from the donor mesh to the target mesh is computed. The error is then computed as described in §2.3.3.2. This process is repeated with pairs of meshes of different sizes. The meshes were generated with Gmsh (Geuzaine and Remacle, 2009).

Four functions were used:

$$\zeta_1(x, y) = x^2 + 2y + 3, \tag{2.27}$$

$$\zeta_2(x, y) = 5y^3 + x^2 + 2y + 3, \tag{2.28}$$

$$\zeta_3(x, y) = \exp x^2 + 2y, \tag{2.29}$$

$$\zeta_4(x, y) = \sin x + \cos y. \tag{2.30}$$

Convergence results for functions ζ_1-ζ_4 with basis function order p varying from 1 to 3 are shown in figures (2.2-2.5). The $O(h^{p+1})$ expected order of

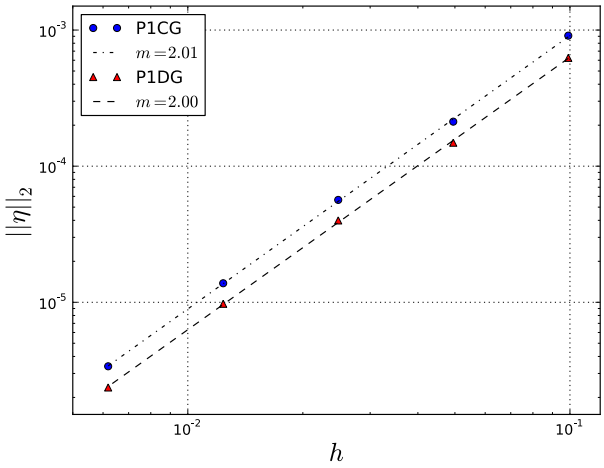


Figure 2.2: Convergence results for the L_2 error of ζ_1 as a function of mesh sizing h for linear basis functions. The error is $O(h^2)$, as expected. Since ζ_1 is quadratic, the error for higher-order basis functions is on the order of numerical zero.

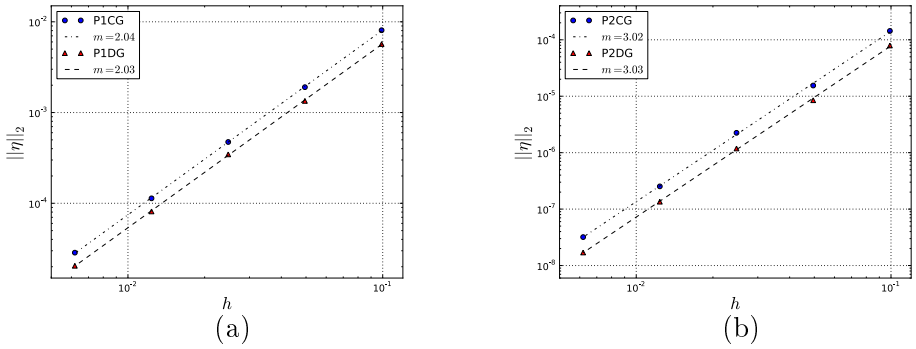
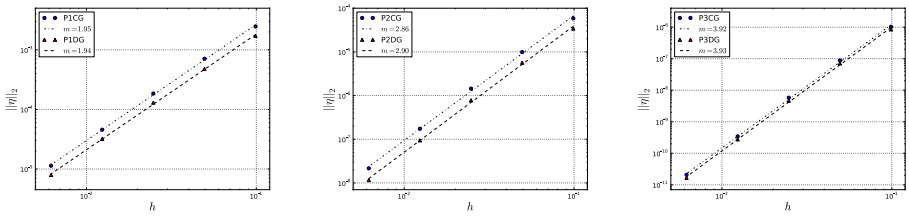
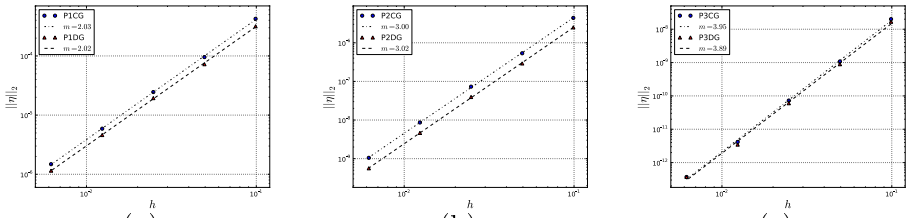


Figure 2.3: Convergence results for the L_2 error of ζ_2 as a function of mesh sizing h for (a) linear basis functions and (b) quadratic basis functions. The error is $O(h^{p+1})$, as expected. Since ζ_2 is cubic, the error for higher-order basis functions is on the order of numerical zero.



(a) (b) (c)

Figure 2.4: Convergence results for the L_2 error of ζ_3 as a function of mesh sizing h for (a) linear basis functions, (b) quadratic basis functions and (c) cubic basis functions. The error is $O(h^{p+1})$, as expected.



(a) (b) (c)

Figure 2.5: Convergence results for the L_2 error of ζ_4 as a function of mesh sizing h for (a) linear basis functions, (b) quadratic basis functions and (c) cubic basis functions. The error is $O(h^{p+1})$, as expected.

h	$\|\|v\|\|$	$\|\|\eta\|\|$	$\|\|v\|\| \,/\, \|\|\eta\|\|$
0.1	9.98×10^{-4}	4.24×10^{-4}	2.36
0.05	2.31×10^{-4}	9.56×10^{-5}	2.42
0.025	6.23×10^{-5}	2.45×10^{-5}	2.54
0.0125	1.48×10^{-5}	5.89×10^{-6}	2.51
0.00625	3.68×10^{-6}	1.48×10^{-6}	2.48

Table 2.1: L_2 norm of the interpolation error in collocation interpolation ($||v||$) and Galerkin projection ($||\eta||$), and their ratio, for the function ζ_4 represented with P1 basis functions. Galerkin projection is optimal in the L_2 norm, so any error above that measures the suboptimality of the collocation interpolation scheme. Collocation interpolation is approximately 2.5× worse for this field.

convergence is observed in the numerical results. The figures for ζ_1 with basis function order $p > 1$ and ζ_2 with basis function order $p > 2$ are not shown as these are representable exactly and the error is on the order of numerical zero (10^{-12} – 10^{-14}), independent of h.

2.3.3.4 The accuracy of collocation interpolation

Lemma 5.1 also allows for the computation of the error in any interpolation operator. To analyse the suboptimality of collocation interpolation, the L_2 interpolation error for both collocation interpolation and Galerkin projection was computed for a series of unstructured quasi-uniform meshes.

Let η be the error in the Galerkin projection and let v be the error in collocation interpolation. For each mesh, the function ζ_4 defined above (equation (2.30)) was applied on the donor mesh by nodal evaluation and transferred to the target mesh. Since $||\eta||$ is optimal, the ratio $||v|| \,/\, ||\eta||$ measures the suboptimality of collocation interpolation. The meshes were generated with Gmsh (Geuzaine and Remacle, 2009). This was repeated with piecewise linear, quadratic and cubic basis functions Φ_D and Φ_T.

Results are shown in table 2.1, table 2.2 and table 2.3. As can be seen, the error in collocation interpolation is only a small multiple of the error in Galerkin projection, which confirms its utility for those cases where conservation is unimportant and the field is continuous. Therefore, for efficiency, Galerkin projection should be used where it must be used, and collocation interpolation used otherwise.

h	$\|\|v\|\|$	$\|\|\eta\|\|$	$\|\|v\|\| \,/\, \|\|\eta\|\|$
0.1	4.94×10^{-6}	4.40×10^{-6}	1.12
0.05	5.93×10^{-7}	5.35×10^{-7}	1.10
0.025	7.98×10^{-8}	7.31×10^{-8}	1.09
0.0125	9.43×10^{-9}	8.63×10^{-9}	1.09
0.00625	1.14×10^{-9}	1.05×10^{-9}	1.09

Table 2.2: L_2 norm of the interpolation error in collocation interpolation ($||v||$) and Galerkin projection ($||\eta||$), and their ratio, for the function ζ_4 represented with P2 basis functions. Galerkin projection is optimal in the L_2 norm, so any error above that measures the suboptimality of the collocation interpolation scheme. Collocation interpolation is approximately 1.1× worse for this field.

h	$\|\|v\|\|$	$\|\|\eta\|\|$	$\|\|v\|\| \,/\, \|\|\eta\|\|$
0.1	3.05×10^{-8}	2.01×10^{-8}	1.52
0.05	1.68×10^{-9}	1.08×10^{-9}	1.56
0.025	1.17×10^{-10}	7.26×10^{-11}	1.61
0.0125	6.64×10^{-12}	4.16×10^{-12}	1.59
0.00625	4.57×10^{-13}	3.72×10^{-13}	1.29

Table 2.3: L_2 norm of the interpolation error in collocation interpolation ($||v||$) and Galerkin projection ($||\eta||$), and their ratio, for the function ζ_4 represented with P3 basis functions. Galerkin projection is optimal in the L_2 norm, so any error above that measures the suboptimality of the collocation interpolation scheme. Collocation interpolation is approximately 1.5× worse for this field.

2.3.4 Bounded minimally-diffusive conservative projection

In the Galerkin projection method (§2.3.3), after assembling the right hand side $M_{TD}q_D$ with the supermesh, a mass matrix M_T is inverted to solve for q_T. First, it is shown that using the lumped mass matrix M_T^L in place of M_T has no effect on the conservation properties of the interpolation for Lagrange elements.

First, a convenient expression for the integral of a field is needed.

Lemma 2.2. *Let $q_T \in \mathcal{V}_T$. Let M_T be the mass matrix. Then*

$$\int_\Omega q_T \, dV = \sum_i \left(M_T q_T\right)^{(i)} . \tag{2.31}$$

Proof. Exploit the fact that the basis functions form a partition of unity and that M_T is symmetric.

$$\begin{aligned}
\int_\Omega q_T \, dV &= \sum_i q_T^{(i)} \int_\Omega \phi_T^i \, dV && (2.32)\\
&= \sum_i q_T^{(i)} \int_\Omega \phi_T^i \left(\sum_j \phi_T^j\right) \, dV && (2.33)\\
&= \sum_i \sum_j q_T^{(i)} (M_T)_{ij} && (2.34)\\
&= \sum_i \sum_j q_T^{(j)} (M_T)_{ji} && (2.35)\\
&= \sum_i \left(M_T q_T\right)^{(i)}. && (2.36)
\end{aligned}$$

□

With that result, it is now possible to prove that lumping the mass matrix by row-summing has no effect on conservation.

Lemma 2.3.

$$\sum_i \left(M_T q_T\right)_i = \sum_i \left(M_T^L q_T\right)_i. \tag{2.37}$$

Proof.

$$\sum_i (M_T q_T)^{(i)} = \sum_i \sum_j q_T^{(i)} (M_T)_{ij} \tag{2.38}$$

$$= \sum_i (\sum_j (M_T)_{ij}) q_T^{(i)} \tag{2.39}$$

$$= \sum_i (M_T^L q_T)^{(i)}. \tag{2.40}$$

□

For the remainder of this section, attention is confined to linear Lagrange basis functions.

Lumping the mass matrix has no effect on conservation, but it has two other major effects: it bounds the resulting solution, as noted in Bailey (1987), and it adds an artificial numerical diffusion (Zienkiewicz and Taylor, 2000a, pg. 476). The next lemma deals with the boundedness of the lumped Galerkin projection for linear basis functions.

Lemma 2.4. *Suppose Φ_D and Φ_T are piecewise linear polynomials. Then for a lumped Galerkin projection,*

$$q_T^{(k)} \leq \max_{\vec{x} \in \Omega} q_D \quad \forall k \in \mathcal{N}_T. \tag{2.41}$$

Proof.

$$\int_\Omega q_D \phi_T^{(k)} \, dV \leq \int_\Omega (\max q_D) \phi_T^{(k)} \, dV, \tag{2.42}$$

$$= (\max q_D) \int_\Omega \phi_T^{(k)} \, dV \tag{2.43}$$

since $\max q_D$ is constant over Ω. Since $\phi_T^{(k)} \geq 0$, and $\phi_T^{(k)}$ is not everywhere zero, $\int_\Omega \phi_T^{(k)} \, dV > 0$, and

$$q_T^{(k)} = \int_\Omega q_D \phi_T^{(k)} \, dV / \int_\Omega \phi_T^{(k)} \, dV \leq \max q_D. \tag{2.44}$$

□

A similar result holds for the lower bound.

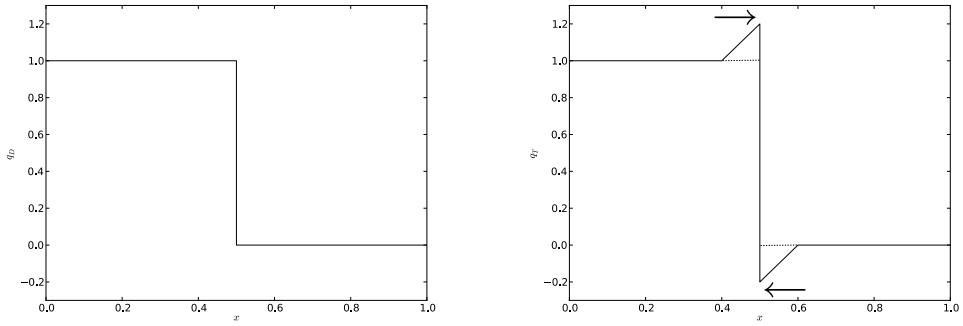


Figure 2.6: The idea behind the bounded variant of Galerkin projection. The left-hand side shows a donor function and the right-hand side shows its unbounded Galerkin projection. The bounded algorithm identifies the overshoots and undershoots (dashed lines) and acts to diffuse them to where they can be absorbed.

To note how this interpolant erodes maxima and minima, consider interpolating from a mesh to itself. It is clear from lemma 2.4 that the maximum is attained only if

$$\int_\Omega q_D \phi_T^{(k)} \, dV = \int_\Omega (\max q_D) \phi_T^{(k)} \, dV, \tag{2.45}$$

for some basis function $\phi_T^{(k)}$, which happens only if $q_D = \max q_D$ over the support of the basis function. Therefore, if the maximum is attained only at a single point, that maximum will be lost in the lumped Galerkin projection.

It follows from the above argument that if only boundedness and conservation are required, the lumped mass matrix can be used on the left hand side of equation (2.18). However, if boundedness, conservation and minimal diffusivity are required, more work must be done.

The fundamental idea of the following algorithm is to compute the consistent Galerkin interpolant, and then selectively apply numerical diffusion to the resulting interpolant to bound it within the bounds of the field q_D on the original mesh (figure 2.6). The minimal amount of numerical diffusion is applied to bound the interpolant, and hence the algorithm is described as a bounded minimally diffusive projection. This algorithm is similar to the one presented in Bailey (1987); in that algorithm, the consistent Galerkin solution is used where the Galerkin projection is naturally bounded, and

the lumped solution used where the Galerkin projection exhibits overshoots and undershoots. The existence of a bounded conservative interpolant is given by the fact that the solution to the lumped version of equation (2.18) satisfies these properties.

To bound the interpolant, it is necessary to know what the bounds at each node of the new mesh should be. If the bounds are known *a priori*, they may be specified by the user; otherwise, the bounds may be computed from the lumped Galerkin projection, or from the linear interpolant of the field. Let $\max q_D$ and $\min q_D$ be the upper and lower bounds of the interpolant.

To measure the deviation from boundedness, define a discrete field q_{dev} defined pointwise such that at each node

$$q_{\text{dev}} = \begin{cases} q_T - \max q_D, & q_T > \max q_D \\ q_T - \min q_D, & q_T < \min q_D \\ 0, & \min q_D \leq q_T \leq \max q_D \end{cases} \tag{2.46}$$

so that boundedness is achieved when $q_{\text{dev}} = 0$ everywhere. Say that a node has absorptive capacity if it lies strictly within the solution bounds. The algorithm applies diffusion to the deviation field to spread it to nodes with absorptive capacity. This is done in such a way that maintains the integral of the interpolant.

At each iteration, a new field q_{alt} is solved for such that

$$M_T^L q_{\text{alt}} = M_T q_{\text{dev}}. \tag{2.47}$$

This operation is trivial due to the lumping of the mass matrix on the left-hand side. The interpolant is then modified such that

$$q_T \leftarrow q_T - q_{\text{dev}} + q_{\text{alt}}. \tag{2.48}$$

Note that by lemma 2.3, this operation has no effect on the integral of q_T. In this update, the numerical diffusion introduced by lumping the mass matrix is exploited to spread the deviation to neighbouring nodes (Zienkiewicz and Taylor, 2000b).

Due to the restriction to linear basis functions, there exists a conservative bounded interpolant; that is, sufficient absorptive capacity exists to absorb the deviation from boundedness. At each diffusion step, the deviation is

spread to neighbouring nodes; if those nodes lie within the bounds, some of the deviation is absorbed; otherwise the deviation is spread further at the next iteration. Therefore, the L_1 norm of the deviation vector forms a nonincreasing sequence. However, since it is possible to have a separation between nodes with absorptive capacity and nodes with deviation from boundedness, it is not guaranteed that the L_1 norm will decrease at any given iteration. As each diffusion step propagates the deviation one edge away from the node with deviation, the deviation will eventually reach the node with absorptive capacity and be reduced, provided the mesh is connected. Thus, the algorithm will converge, albeit non-monotonically. The convergence may be accelerated by upwinding appropriately to direct the diffusion of the deviation field into regions with absorptive capacity, and by applying successive over-relaxation, but these are not discussed here for reasons of brevity. In the event of stalling, it is possible to take the deviation away and place it directly at the nodes with absorptive capacity, without regard to spatial locality; this will introduce some diffusion, but will achieve boundedness. If such a step is performed after a suitable number of diffusive steps, the spurious numerical diffusion added will be negligible.

This update is iterated until the deviation field is zero, or some tolerance is reached. In the examples shown later, satisfactory convergence to 10^{-10} is achieved with approximately 1000 iterations. These iterations are cheap, each requiring only a matrix-vector multiplication, an array division, and two vector additions.

2.4 Examples

2.4.1 Numerical order of convergence of the bounded method

Using the full mass matrix in the Galerkin projection gives second order convergence for piecewise linear basis functions, while using the lumped mass matrix reduces this to first order; therefore, since the bounded variant selectively applies the lumped mass matrix to bound the interpolant, the expected order of convergence lies between one and two.

A numerical experiment was performed to investigate the observed order of convergence of the bounded Galerkin projection. The same four functions

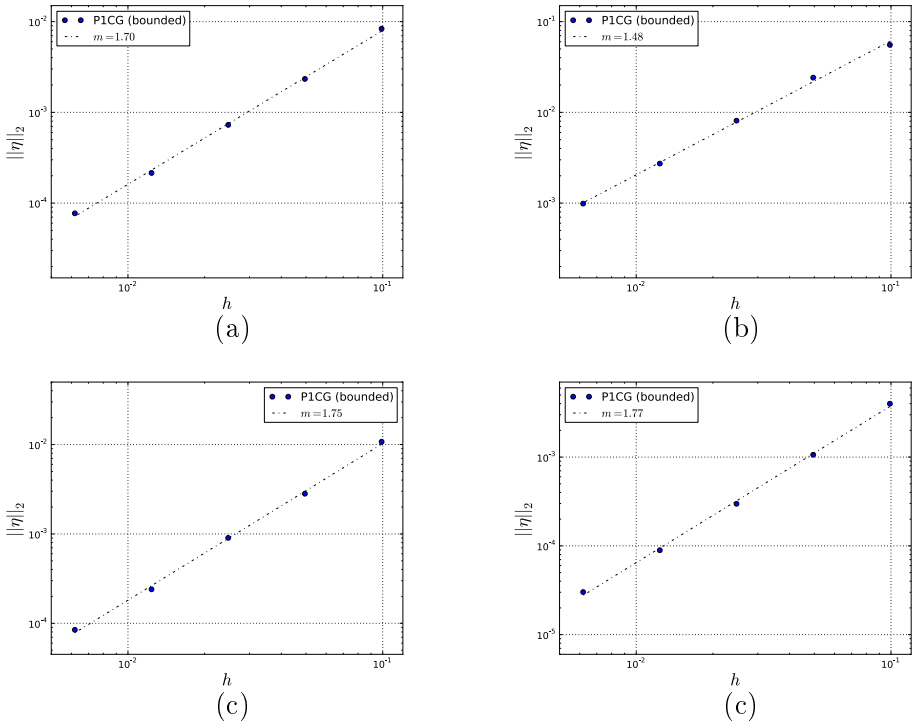


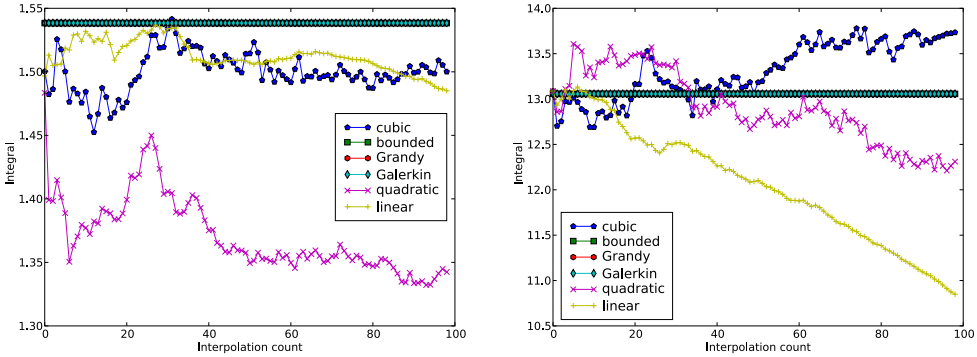
Figure 2.7: Convergence results for the L_2 error in bounded Galerkin projection of (a) ζ_1 (b) ζ_2 (c) ζ_3 (d) ζ_4 as a function of mesh sizing h for P1 basis functions. The error is between first and second order, as expected.

ζ_1-ζ_4 were used (equation 2.30), and the same meshes and procedure as described in §2.3.3.3.

Results are shown in figure 2.7. As expected, the observed order of convergence lies between one and two.

2.4.2 Repeated interpolation

To test the effectiveness of the interpolation operators presented, 100 simplicial meshes of the domain $\Omega = [-3, 3]^2$ are generated. The first mesh is populated with initial fields by nodally interpolating an analytical expression. The discrete functions are then interpolated onto each mesh in turn. The first mesh was a structured mesh with 10000 nodes, to give a reasonable representation of the initial fields, while all the other meshes have 1000 nodes randomly placed with a uniform distribution. These meshes were generated



(a) (b)

Figure 2.8: (a) & (b): integrals of fields p_T and q_T. The three schemes expected to be conservative (Grandy, Galerkin and bounded) are indeed conservative. The other schemes (linear, quadratic and cubic) are not.

with Triangle (Shewchuk, 1996). Randomly generated meshes are used to stress the interpolation operators; it is to be emphasised that the meshes used in this experiment are deliberately under-resolved and not adapted in any sense to the representation of the fields on them.

Two fields are initialised on the first mesh; the first, p, is a hat function centred at the origin of radius 0.7:

$$p(x,y) = \begin{cases} 1, & \sqrt{x^2+y^2} \leq 0.7 \\ 0, & \text{otherwise} \end{cases} \tag{2.49}$$

while the second, q, is the `peaks` function from MATLAB,

$$q(x,y) = 3(1-x)^2 \mathrm{e}^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right)\mathrm{e}^{-x^2-y^2} - \frac{1}{3}\mathrm{e}^{-(x+1)^2-y^2}, \tag{2.50}$$

as used in Jiao and Heath (2004a).

To investigate the performance of the algorithms, the integral of the interpolant, its bounds, and the L_2 error against the analytical formula are recorded for each interpolation.

To solve the Galerkin projection, the Conjugate Gradient algorithm is used (Balay et al., 1997) with a relative residual tolerance of 10^{-10}. The mass matrix is preconditioned with Symmetric Successive Over-Relaxation. For

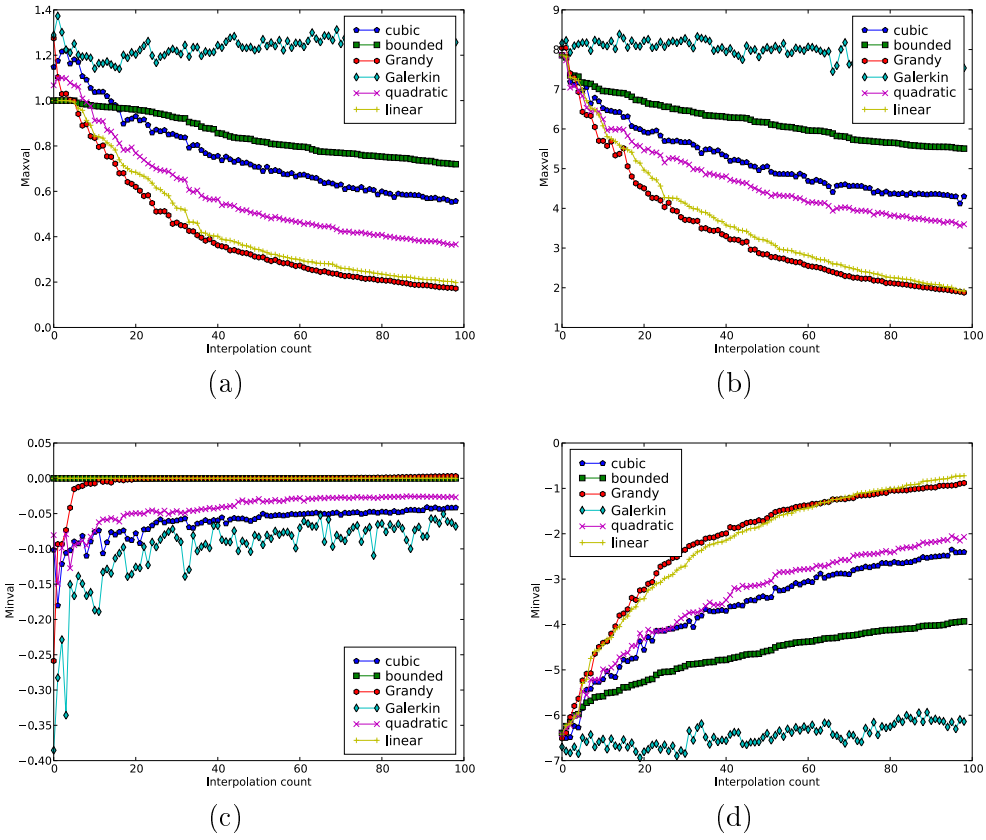


Figure 2.9: (a) & (b): maximum value of fields p_T and q_T. (c) & (d): minimum value of fields p_T and q_T. In both cases, the bounded algorithm does an excellent job of retaining the bounds of the fields. Grandy interpolation and linear interpolation do a poor job of retaining the maximum value for both examples presented.

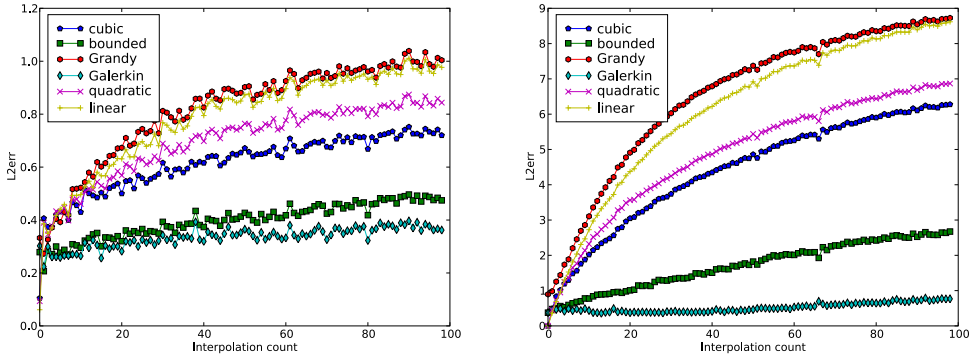


Figure 2.10: (a) & (b): L_2 error of fields p_T and q_T against analytical nodal interpolants. Galerkin projection is the best method considered, with the bounded scheme second best.

the bounded algorithm, the bounds are computed as the least conservative of lumped Galerkin projection and collocation interpolation. 1000 boundedness iterations are used. Also compared are a quadratic fitting scheme which performs a least-squares fit of a quadratic polynomial to the donor function and evaluates this fit at the nodes of the target mesh, as described in Vallet et al. (2007), and an analogous unpublished cubic fitting scheme.

The integral results are presented in figures 2.8a and b. As can be seen, the three conservative interpolation operators presented here all conserve the integral of the field, regardless of the unsuitability of the meshes; while collocation interpolation, quadratic fitting and cubic fitting rapidly change the integral.

The boundedness results are presented in figures 2.9a-d. The bounded scheme (§2.3.4) and collocation interpolation keep the interpolant within the original bounds, while the Grandy scheme (§2.3.2), consistent Galerkin projection (§2.3.3), quadratic fitting and cubic fitting introduce oscillations. This is particularly visible in the p function. The bounded scheme best preserves the bounds, as the maximum and minimum values are closest to horizontal lines.

The accuracy results are presented in figures 2.10a and b. The accuracy of the Grandy scheme is greatly hampered by assumption (2.15). As expected, the Galerkin interpolant is the interpolant that minimises the L_2 error. Note that the Galerkin interpolant minimises the L_2 error between interpolants,

rather than the L_2 error against the analytical nodal interpolant, which is what is plotted here. The bounded algorithm has a slightly higher error, as expected. The error of the quadratic fitting and cubic fitting schemes is intermediate between Galerkin projection and collocation interpolation.

These results demonstrate the effectiveness of the bounded interpolation algorithm presented in this work, as it is simultaneously conservative, bounded, and more accurate than all the other methods analysed except for Galerkin projection.

2.4.3 Adaptive example

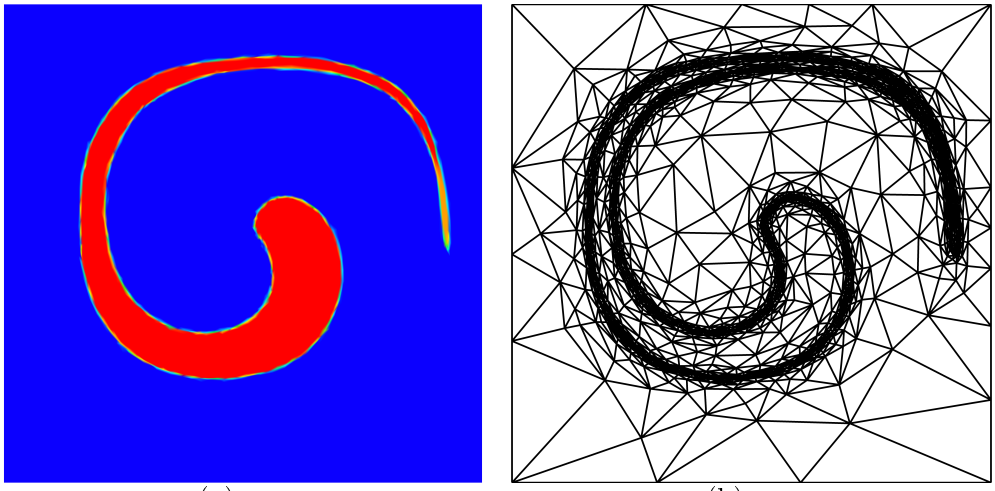
An adaptive example is presented to demonstrate the utility of the conservative, bounded interpolation algorithm. The test is drawn from Rudman (1997). An initially circular volume fraction with radius $\frac{\pi}{5}$ centred on $\left(\frac{\pi}{2}, \frac{\pi+1}{5}\right)$ is advected around the domain $\Omega = [0, \pi]^2$ with a prescribed velocity $(\cos(x)\sin(y), -\sin(x)\cos(y))$. A bounded, control volume advection algorithm is used. Details of the algorithm are forthcoming in Wilson (2009).

As volume fractions typically exhibit sharp, anisotropic interfaces, adaptive remeshing is well-suited to this problem. However, the physics of multimaterial simulations places severe constraints on the numerical methods used.

The simulation is run for 15000 timesteps at a CFL number of 0.1, with adaptive remeshing invoked every 10 timesteps. The timestep is automatically adjusted after every adapt to maintain a constant CFL number. The error metric used to guide the adaptive algorithm is calculated from the material volume fraction using the algorithm described in Pain et al. (2001). The two-dimensional adaptive remeshing algorithm described in Vasilevskii and Lipnikov (1999) is used to update the mesh.

The simulation was initially configured to use collocation interpolation. However, this results in nonphysical exchanges of volume between the materials represented by the volume fraction (see figure 2.12a). Therefore, a conservative interpolation algorithm must be employed. When Galerkin projection is employed, its non-boundedness very quickly leads to physically impossible results (see figure 2.12b).

Bounded Galerkin projection solves both of these problems: it maintains the integral of the material volume fraction (up to the accuracy of the ad-



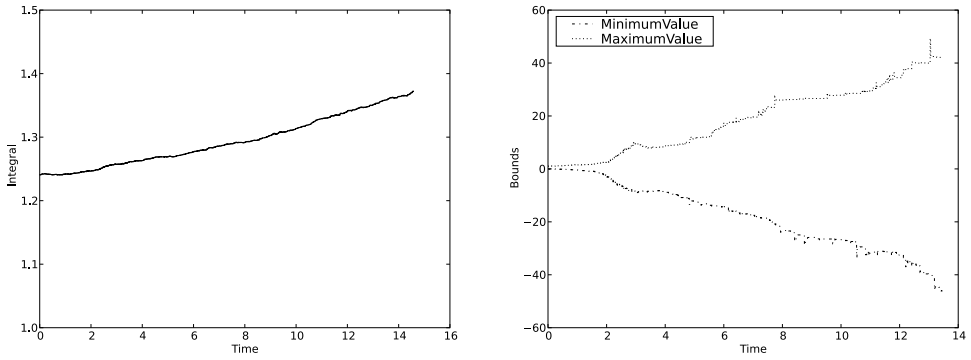
(a) (b)

Figure 2.11: (a) & (b): The advected material volume fraction and (b) the mesh at $t = 14.6$. Bounded Galerkin projection was used for this simulation.

vection algorithm used; figure 2.13a), while simultaneously preserving the physical bounds of the field (figure 2.13b).

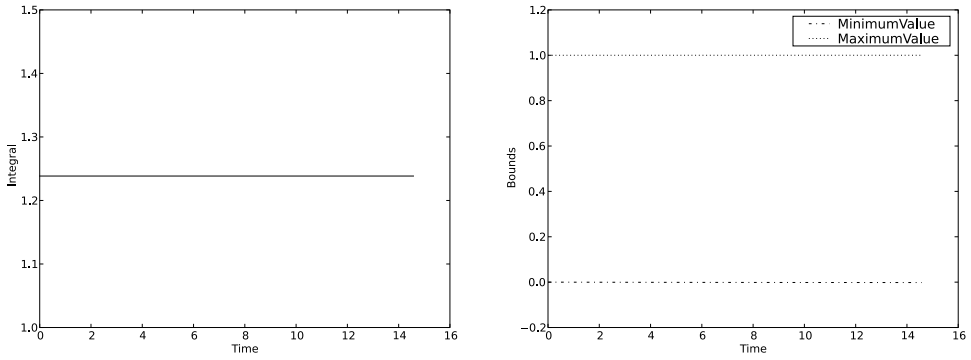
2.5 Conclusions

A bounded minimally-diffusive conservative interpolation algorithm for general unstructured meshes has been described. The algorithm conserves quantities to the accuracy of machine precision and the linear solver tolerances, and is thus suitable for use in application areas demanding conservative discretisation methods, such as nuclear reactor simulations (Pain et al., 2005a) and long-term geophysical fluid dynamics (Piggott et al., 2008; Slingo et al., 2009). With the availability of conservative interpolation, an objection to the use of unstructured adaptive algorithms is removed. This allows the more widespread application of adaptive remeshing and the associated computational savings. Although the examples presented here are two-dimensional, these algorithms apply unmodified to the three-dimensional case, provided a supermesh construction algorithm is available. Future work will involve investigating projection methods which conserve higher-order moments.



(a) (b)

Figure 2.12: (a) Integral of the material volume fraction as a function of time, for collocation interpolation. Note the nonphysical exchange of material mass. (b) Bounds of the material volume fraction as a function of time, for Galerkin projection. Note the nonphysical negative volume fraction.



(a) (b)

Figure 2.13: (a) Integral of the material volume fraction as a function of time, for bounded conservative interpolation. (b) Bounds of the material volume fraction as a function of time, for bounded Galerkin projection.

Chapter 3

Supermesh construction

Abstract

As demonstrated in the previous chapter, Galerkin projection offers several extremely attractive features over collocation interpolation. These properties have been well-known for over 10 years, since George and Borouchaki (1998). However, these methods have not gained widespread popularity because of the difficulty of assembling the associated linear system, which involves the solution of some problems of computational geometry: the construction of the supermesh. This chapter presents two algorithms for the construction of the supermesh. The first presents a proof-of-concept implementation in two dimensions based upon a transformation to a constrained Delaunay triangulation. The second is a practical, efficient and local algorithm which extends to two and three dimensions.

This chapter is derived from and expands upon Farrell et al. (2009) and Farrell and Maddison (2009).

Contents

3.1 Introduction

To assemble the right-hand side of equation (2.18), it is necessary to integrate the products of the basis functions of $\mathcal{T}_D$ and $\mathcal{T}_T$. Over each element of $\mathcal{T}_T$, the basis functions of $\mathcal{T}_D$ are piecewise polynomials. Therefore, if the products of the basis functions are evaluated at each of the quadrature points of $\mathcal{T}_T$, the integrals will not in general be exact, as quadrature schemes are typically exact for a specified polynomial order, not piecewise polynomials. This error in the assembly of M_{TD} causes the loss of conservation and accuracy properties, which is highly undesirable.

To circumvent this, a *supermesh* $\mathcal{T}_S$ is formed, a mesh of the intersections of the elements of $\mathcal{T}_D$ and $\mathcal{T}_T$, as defined in definition 2.1. Over each element of $\mathcal{T}_S$, the basis functions of $\mathcal{T}_D$ and $\mathcal{T}_T$ are polynomials, not piecewise polynomials, and their product may therefore be integrated exactly.

This chapter presents two algorithms for the construction of the supermesh. The initial approach taken in Farrell et al. (2009) was to convert

the problem of supermesh construction to that of a constrained Delaunay triangulation (CDT) (§3.2). This conversion is made possible by lemma 3.1, a novel result. While this made possible the proof-of-concept demonstration of the effectiveness of Galerkin projection in Farrell et al. (2009), it was not efficient and did not extend naturally to three dimensions. Therefore, a superior local supermeshing algorithm was developed, as described in §3.3. This enabled the first three-dimensional implementation of Galerkin projection (Farrell and Maddison, 2009).

3.2 Supermesh construction by transformation to a constrained Delaunay triangulation

Given two input meshes $\mathcal{T}_D$ and $\mathcal{T}_T$, the objective is to construct a supermesh satisfying definition 2.1. The following result shows how this problem may be converted to that of a constrained meshing problem.

Lemma 3.1. *Let $\mathcal{T}_D, \mathcal{T}_T$ be two arbitrarily unstructured meshes of the same polyhedral domain Ω, with nodes $\mathcal{N}_D, \mathcal{N}_T$ and edges $\mathcal{F}_D, \mathcal{F}_T$. Then any triangulation $\mathcal{T}_S$ of Ω where the presence of all nodes $n \in \mathcal{N}_D \cup \mathcal{N}_T$ and all edges $f \in \mathcal{F}_D \cup \mathcal{F}_T$ is enforced is a supermesh of $\{\mathcal{T}_D, \mathcal{T}_T\}$, as defined by definition 2.1.*

Proof. The first condition of definition 2.1 is satisfied by assumption.

Let $K_S \in \mathcal{T}_S$ be an element of the resulting supermesh, and let $\mathcal{T}_P$ be a parent input mesh, either $\mathcal{T}_D$ or $\mathcal{T}_T$. Since the supermesh discretises the same domain Ω as $\mathcal{T}_P$, there must exist at least one element $K_P \in \mathcal{T}_P$ with an intersection of nonzero measure with K_S. By assumption, the presence of the edges of K_P are enforced in the mesh $\mathcal{T}_S$. Therefore, K_S cannot cross the edges of K_P, and is thus wholly contained within K_P: $V(K_S \cap K_P) = V(K_S)$. It therefore follows that the intersection of K_S with any other $K'_P \in \mathcal{T}_P \setminus \{K_P\}$ must have zero measure, and therefore condition 2 of definition 2.1 is satisfied. □

The utility of this result derives from the fact that the problem of such a constrained meshing problem is well-studied in the case where the output is composed of simplicial elements: it is referred to as a constrained Delaunay triangulation (CDT) (Lee and Lin, 1986; Chew, 1989, 1993; Ruppert, 1995;

Shewchuk, 2002b). In this work, the two-dimensional mesh generator Triangle is used (Shewchuk, 1996). By transforming the problem of supermesh construction into that of a constrained triangulation, lemma 3.1 allows us to use widely available, robust algorithms for the problems of computational geometry inherent in the conservative interpolation algorithms presented. Note that a simplicial supermesh can be used for the purposes of conservative interpolation between two meshes composed of quadrilaterals or hexahedra, as it is merely used for the purposes of computing $M_{\mathcal{T}_T\mathcal{T}_D}$.

Lemma 3.1 leads directly to the following algorithm for the construction of a simplicial supermesh.

Algorithm 3.1. *Input: $\mathcal{T}_D$ and $\mathcal{T}_T$. Output: $\mathcal{T}_S$, a supermesh of the input meshes.*

1. $\mathcal{N} \leftarrow \mathcal{N}_\mathcal{D} \cup \mathcal{N}_\mathcal{T}$

2. $\mathcal{F} \leftarrow \mathcal{F}_D \cup \mathcal{F}_T$

3. $\{\mathcal{N}, \mathcal{F}\}$ *forms an imperfect planar straight line graph (PSLG) in two dimensions, or an imperfect piecewise linear complex (PLC) in three dimensions.*

4. *Give $\{\mathcal{N}, \mathcal{F}\}$ as input to a constrained Delaunay triangulation algorithm such as that described in Shewchuk (1996).*

By imperfect PSLG it is meant a planar straight line graph where some of the specified edges intersect. Some constrained Delaunay triangulation algorithms are capable of detecting such intersections and dividing the edges appropriately; Triangle is so capable. If the mesh generator used is not capable of such division, the intersecting edges must be divided and extra nodes added as a pre-processing step before calling the mesh generation algorithm.

With the appropriate data structures, the complexity of the set union operation is linear in the size of its inputs and therefore the transformation to a PSLG takes linear time. Efficient algorithms for the construction of the CDT are log-linear in time (Chew, 1989).

3.2.1 Parenthood mapping construction

Algorithm 3.1 constructs a supermesh from two given input meshes, but it does not explicitly discuss how to construct the parenthood mappings χ_{SD} and χ_{ST}. One possible approach to construct the parenthood mappings is to use an R-tree algorithm (Guttman, 1984; Manolopoulos et al., 2005). This section discusses an alternative where the input to the constrained Delaunay triangulation is annotated so that these mappings may be recovered from the output of the mesh generator.

Mesh generators typically allow each edge to be annotated with an integer value. This is used for associating edges with regions where boundary conditions should be applied.

Suppose χ_{SP} is to be constructed, for $\mathcal{T}_P$ a parent input mesh with nodes $\mathcal{N}_P$ and edges $\mathcal{F}_P$. Each edge $f \in \mathcal{F}_P$ is annotated as follows. Let K, K' be the two elements in $\mathcal{T}_P$ that share f. If f is on the boundary of the domain, take K' to be some positive sentinel value (such as a number greater than the number of elements in the input mesh). Since each edge may only be annotated with precisely one integer, these two integers must be encoded into one. Such bijections $C : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ are well known from proofs of the countability of the rationals $\mathbb{Q}$. In this work, bijection #7 from Bradley (2005) is taken:

$$C(m, n) = m + (m + n - 2)(m + n - 1)/2, \tag{3.1}$$

with inverse

$$C^{-1}(q) = (M(q), 1 + L(q) - M(q)), \tag{3.2}$$

$$L(q) = \lfloor 1/2 + \sqrt{2q - 1} \rfloor, \tag{3.3}$$

$$M(q) = q - (L(q) - 1)(L(q))/2. \tag{3.4}$$

f is annotated with $C(K, K')$ for all $f \in \mathcal{F}_P$. All edges which only appear in the other input mesh are left unannotated, so that the annotations associated with $\mathcal{T}_P$ propagate correctly through the mesh generation procedure. (This therefore necessitates calling the mesh generation algorithm twice, once for each input mesh; this is a consequence of only allowing one annotation per edge. This is a particular detail of only allowing one annotation per edge. Because of this, the mesh generator should be deterministic.)

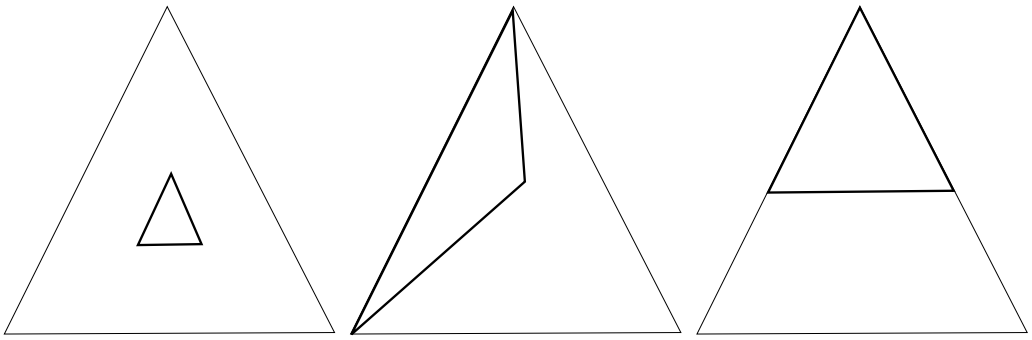


Figure 3.1: Three possible cases of edge annotations. Normal line is the parent element, bold line is the child element. (a) No annotations available. (b) One annotation available. (c) More than one annotation available.

This input is then passed to the mesh generator to yield $\mathcal{T}_S$.

Consider an element $K_S \in \mathcal{T}_S$. There are 3 possible cases (figure 3.1):

- No annotations available. This happens if and only if no edge of the child element shares a segment with an edge of the parent element. Such an element is guaranteed to have the same parent as any of its neighbours, so the resolution of its parenthood is deferred until other the cases have been dealt with; then, its neighbours are consulted to determine its parenthood.

- One annotation available. This happens if and only if one edge of the child element shares a segment with an edge of the parent element. Since the edge encodes the only two possible parent elements, a containment test of the kind described in Löhner (1995a) is applied to decide in which parent element the vertex opposite the annotated edge lies.

- More than one annotation available. This happens if and only if more than one edge of the child element shares a segment with an edge of the parent element. As each edge annotation encodes at most two possible parents, the parent of this element is the intersection of the annotations of the annotated edges.

Thus the mapping χ_{SP} is constructed. This is trivially inverted to give the map χ_{PS}, if it is required.

The complexity of the edge annotation algorithm is linear in the number of edges of the input meshes. The complexity of the parenthood resolution algorithm is linear in the number of elements of the supermesh.

3.3 Local supermeshing

The previous section shows that meshing the regions of intersection is equivalent to enforcing the existence of the nodes and edges of the two meshes in the supermesh; thus, the problem of constructing a supermesh can be converted into a constrained meshing problem which may be solved with a constrained Delaunay triangulation (CDT).

While this conveniently passes the geometric work of constructing the supermesh to well-known, robust and widely available algorithms, it has several downsides. Firstly, the input to the CDT is imperfect, in the sense that the edges whose existence is enforced intersect with each other. While Triangle is capable of handling imperfect input, the author is unaware of any such program that is available for solving the imperfect CDT in three dimensions. Secondly, the parenthood mappings must be established: for each element in the supermesh, the parent elements in $\mathcal{T}_D$ and $\mathcal{T}_T$ must be identified. Thirdly, this approach constructs the whole supermesh at once, and is hence referred to as global supermeshing. While practical for smaller inputs, storing the entire supermesh in main memory becomes prohibitively expensive for larger problem sizes.

In this section, an alternative approach called local supermeshing is proposed. Here, the supermesh is formed by meshing each intersection in turn. This approach is particularly suited to interpolation between spaces of discontinuous functions, as both the assembly and solve can be performed entirely locally on a given element of $\mathcal{T}_T$ by exploiting the block-diagonal structure of the mass matrix.

3.3.1 Intersection identification

To form a supermesh intersection by intersection, it is first necessary to identify the intersecting pairs of elements of $\mathcal{T}_D$ and $\mathcal{T}_T$. The existence of a suitable geometric intersection predicate is assumed (for more details, see Mount (1997)).

The naïve brute-force approach performs $O(|\mathcal{T}_D||\mathcal{T}_T|)$ intersection tests, which is clearly undesirable. The element pairs may be first filtered by an R-tree algorithm, which only considers pairs with intersecting bounding boxes for intersection testing (Guttman, 1984; Manolopoulos et al., 2005). The construction of the R-tree takes $O(|\mathcal{T}_D| \log |\mathcal{T}_D|)$ time, and the query for each element of $\mathcal{T}_T$ takes on average $O(\log |\mathcal{T}_D|)$ time, for a total time of $O((|\mathcal{T}_D| + |\mathcal{T}_T|) \log |\mathcal{T}_D|)$. This figure ignores the actual bounding box intersection tests performed ascending and descending the tree, which will render it sensitive to the number of intersections between the meshes. Neither of these algorithms exploits the mesh-based structure of $\mathcal{T}_D$ and $\mathcal{T}_T$. By exploiting the fact that the elements are not merely sets of polytopes but that they form meshes of known connectivity, it is possible to develop a novel algorithm for intersection identification which performs $O(|\mathcal{T}_D| + k)$ intersection tests, where k is the number of intersecting elements between $\mathcal{T}_D$ and $\mathcal{T}_T$. If an initial seed is supplied to the algorithm, this reduces to $O(k)$. This algorithm is described in more detail in chapter 4.

3.3.2 Intersection construction

Once the intersecting elements in $\mathcal{T}_D$ are identified for a given $K_T \in \mathcal{T}_T$, these intersections must be meshed so that the quadrature of the products of the basis functions may be performed.

Various intersection construction algorithms are available, depending on the specifics of the elements used. For general convex polytopes, algorithms are available in two dimensions (Shamos and Hoey, 1976) and three dimensions (Chazelle, 1992). One of the simplest is the Sutherland-Hodgman clipping algorithm (Sutherland and Hodgman, 1974).

The Sutherland-Hodgman algorithm is illustrated in figure 3.2. The subject polygon is initialised to be one of the input polygons, and the other is designated the clipping polygon. Each edge of the clipping polygon is considered in turn. The edge, when extended, forms a line. An orientation test is performed for each vertex of the subject polygon to determine whether it is on the positive or negative side of the line, or collinear. If its orientation is positive or collinear, it is retained, while if it is negative, it is discarded. Furthermore, if one vertex of an edge in the subject polygon is retained while the other vertex is discarded, then that edge must intersect the clipping line,

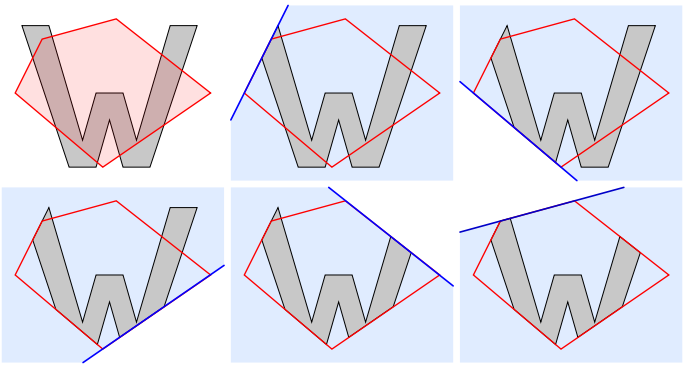


Figure 3.2: The Sutherland-Hodgman clipping algorithm (Sutherland and Hodgman, 1974). The subject polygon is initialised to the 'W' and the clipping polygon is the pentagon. Each edge of the clipping polygon discards the subject vertices outside it, possibly creating new vertices for any intersecting edges. Figure credit: Wikipedia (public domain).

so the point of intersection between the subject edge and the clipping line is computed and inserted into the output vertices. The list of output vertices then forms the subject polygon to be compared to the next edge of the clipping polygon. In this manner, vertices of the subject polygon which are external to the clipping polygon are systematically removed.

A similar approach is taken in three dimensions. Each face of the clipping polyhedron is considered in turn. For each vertex of the subject polyhedron, an orientation test is performed to decide whether the vertex should be discarded or retained. Again, if an edge has one vertex discarded and one vertex retained, then a new vertex is formed at the intersection of the edge and the clipping plane. The new vertices produced are then connected with edges to form a new face of the subject polyhedron. This procedure is continued until all the faces of the clipping polyhedron have discarded those parts of the subject polyhedron outside the intersection.

A disadvantage of the Sutherland-Hodgman approach is that the algorithm only returns the vertices of the polytope of the intersection. In order to assemble the integrals of the mixed mass matrix, this polytope must be meshed. If the two input elements are convex, the intersection polytope is also convex, and thus the Delaunay triangulation of the vertices of the polytope is a mesh of the intersection.

An alternative approach, adopted in this work, naturally constructs the

mesh of the polytope during the intersection procedure. This algorithm follows that of Eberly (2001). The list of output polytopes is initialised to be (a simplicial decomposition of) one of the input polytopes, and the other is designated the clipping polytope. Again, each edge of the clipping polytope is considered in turn. Each polytope in the output list is clipped against the edge; the clipping procedure returns a list of simplices that are to replace the considered polytope. In this manner, at every step of the intersection procedure, the intersection is represented as a list of simplices. Therefore, the output of the procedure consists of a list of simplices which constitute a mesh of the intersection. By design, the clipping procedure need only consider the intersection of a simplex with a half-space; this can be divided into a handful of possible cases and the solution for each devised on paper. This approach eliminates the need for costly post-processing of the resulting intersection polytope.

3.4 Adaptive quadrature approach

Clearly, supermeshing is not the only way to compute the inner products of the basis functions of the target and donor meshes: it is merely the only way to compute them exactly (ignoring roundoff). An alternative is to evaluate the basis functions of the donor mesh at the quadrature points of the target mesh and compute the integrals using numerical quadrature. In this vein, an experiment was performed to test the practicality of computing the integrals in this manner. A very similar approach was advocated in El Hraiech et al. (2005); there, the authors subdivide the elements of the target mesh into several sub-elements to compute a more accurate approximation of the integrals.

The adaptive quadrature package `CUBPACK` (Cools and Haegemans, 2003) was chosen as the numerical quadrature scheme, as this appeared to offer implementations of the most recent research in numerical quadrature. The example used was the one presented in §3.6.2. The adaptive quadrature algorithm was used to compute the inner products of the basis functions of the two meshes. The target relative error was set to 1% of the integral and the maximum number of integrand evaluations for each element in the target mesh was set to 10^5.

It was found that the integral computation performed for each element

reached the maximum number of integrand evaluations, 10^5; the target relative error of 1% was never reached. As such, the system of equations was inaccurately assembled, resulting in conservation errors on the order of 10^{-4}, or 0.02%. By contrast, the integral error arising from local supermeshing was on the order of 10^{-16}. The interpolant constructed with supermeshing took less than a second, while the adaptive quadrature approach took over 15 minutes. The adaptive quadrature approach was therefore over 10^3 times slower than projection by supermeshing, for a poorer result. The experiment was not attempted on other examples due to the prohibitive computational cost.

It is to be emphasised that the outcome of this experiment does not reflect on the quality of the algorithms developed in `CUBPACK`. Over each element, the integral to be computed is a discontinuous piecewise polynomial; it is merely the case that supermeshing is a vastly more efficient method for computing these particularly difficult integrals to within an acceptable error, even for this simple case. However, it would appear that the method of El Hraiech et al. (2005) is not practical for such integrals.

3.5 Summary

The Galerkin projection algorithm is summarised as follows.

Algorithm 3.2. *Galerkin projection.*

1. *Identify the intersecting pairs of elements.*
2. *For each element $K \in \mathcal{T}_T$:*
   a) *Assemble the contribution to the mass matrix M_T.*
   b) *For each intersecting element $K_D \in \mathcal{T}_D$:*
      i. *Form $\mathcal{T}_S^K$, the mesh of the region of intersection.*
      ii. *Assemble the contribution to the mass matrix M_{TD} by integrating the basis functions of K and K_D over $\mathcal{T}_S^K$.*
      iii. *Apply this to q_D to form the contribution to the right-hand side of equation (2.18).*
   c) *If $\mathcal{T}_T$ is discontinuous, perform the local solve of equation (2.18).*
3. *If $\mathcal{T}_T$ is continuous, perform the global solve of equation (2.18).*

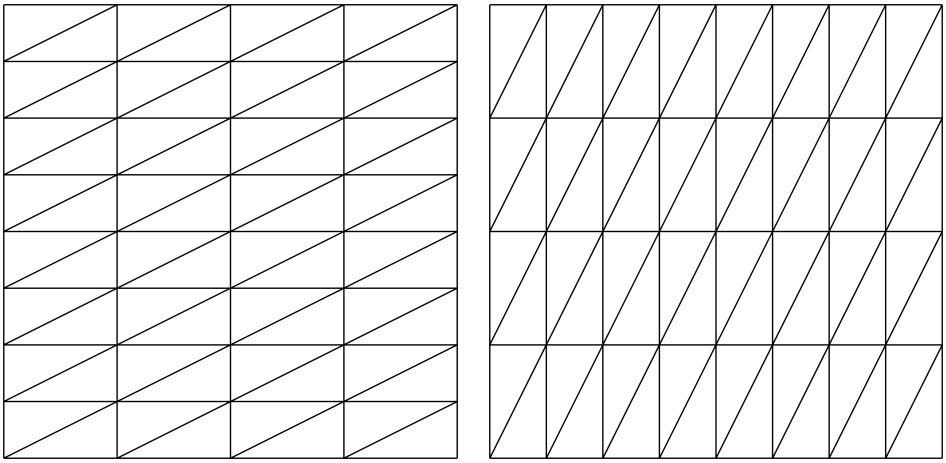


Figure 3.3: Two two-dimensional meshes used in the profiling analysis of Galerkin projection.

These algorithms have been implemented in the open-source CFD/GFD framework Fluidity/ICOM (Mansoorzadeh et al., 1998; Piggott et al., 2008).

3.6 Examples

3.6.1 Profiling results

An experiment was conducted to investigate the scaling of Galerkin projection with problem size. For a given size of problem, two different regular structured meshes of equal numbers of elements were generated and a $\mathrm{P1_{DG}}$ field projected back and forth between them 10 times. The time taken for this problem size was then computed as the total CPU time (measured by the `cpu_time` intrinsic of Fortran 90) divided by the number of projections. Therefore, the reported timings include the cost of I/O, the intersection finder, constructing the supermesh, assembling the Galerkin system and solving it. In two dimensions, for a given problem size n, the meshes were generated by dividing the unit square into $n \times 2n$ and $2n \times n$ subdivisions (figure 3.3). In three dimensions, for a given problem size n, the meshes were generated by dividing the unit cube into $n \times 2n \times n$ and $n \times n \times 2n$ subdivisions. This procedure was repeated for several different problem sizes in two and three dimensions.

The experiment was conducted in serial on a two-processor quad-core Intel E5530 2.4 GHz machine with 12Gb of RAM. The implementation of the algorithm was compiled with version 10.1 of the Intel compiler suite and used the Hoard memory allocator (Berger et al., 2000).

The results are shown in figure 3.4. As can be seen, the time taken by the algorithm is linear in the size of the input meshes. In two dimensions, the algorithm takes approximately 0.12 ms of CPU time per element in the mesh; in three dimensions, 0.15 ms of CPU time per element. Despite the inherent complexity of supermeshing in three dimensions, the procedure is almost as fast in three dimensions as in two; this is because significant development effort was invested in optimising the three-dimensional intersector using Oprofile (Levon and Elie, 2009). For the largest mesh of 165888 tetrahedral elements (663552 degrees of freedom), the Galerkin projection took 25.8 s, which is judged to be sufficiently fast for practical use. Note that in the context of adaptive remeshing, the cost could be reduced dramatically by supplying the projection algorithm with information about unchanged elements from the adaptive remeshing algorithm. If only 10% of the elements of the mesh change, then it is unnecessary to supermesh or assemble the mixed mass matrix over the other 90%, and thus the cost of Galerkin projection would be reduced by a factor of approximately 10.

3.6.2 Two-dimensional square

To demonstrate the applicability of Galerkin projection to interpolation between discontinuous function spaces, two $\mathrm{P2_{DG}}$ meshes (discontinuous quadratic polynomial basis functions) were generated of the domain $\Omega = [0,1]^2 \subset \mathbb{R}^2$ using Netgen[1] (Schöberl, 1997). The discontinuous field

$$\psi(x,y) = \begin{cases} 0, & x \leq \frac{1}{2}, \\ 1, & x \geq \frac{1}{2}, \end{cases} \tag{3.5}$$

is applied by nodal evaluation on the donor mesh. The donor mesh is arranged so that element faces align with the discontinuity along the line $x = 0.5$, while the target mesh does not. The donor and target meshes consist of 202 and 208 elements respectively.

[1] `http://www.hpfem.jku.at/netgen/`

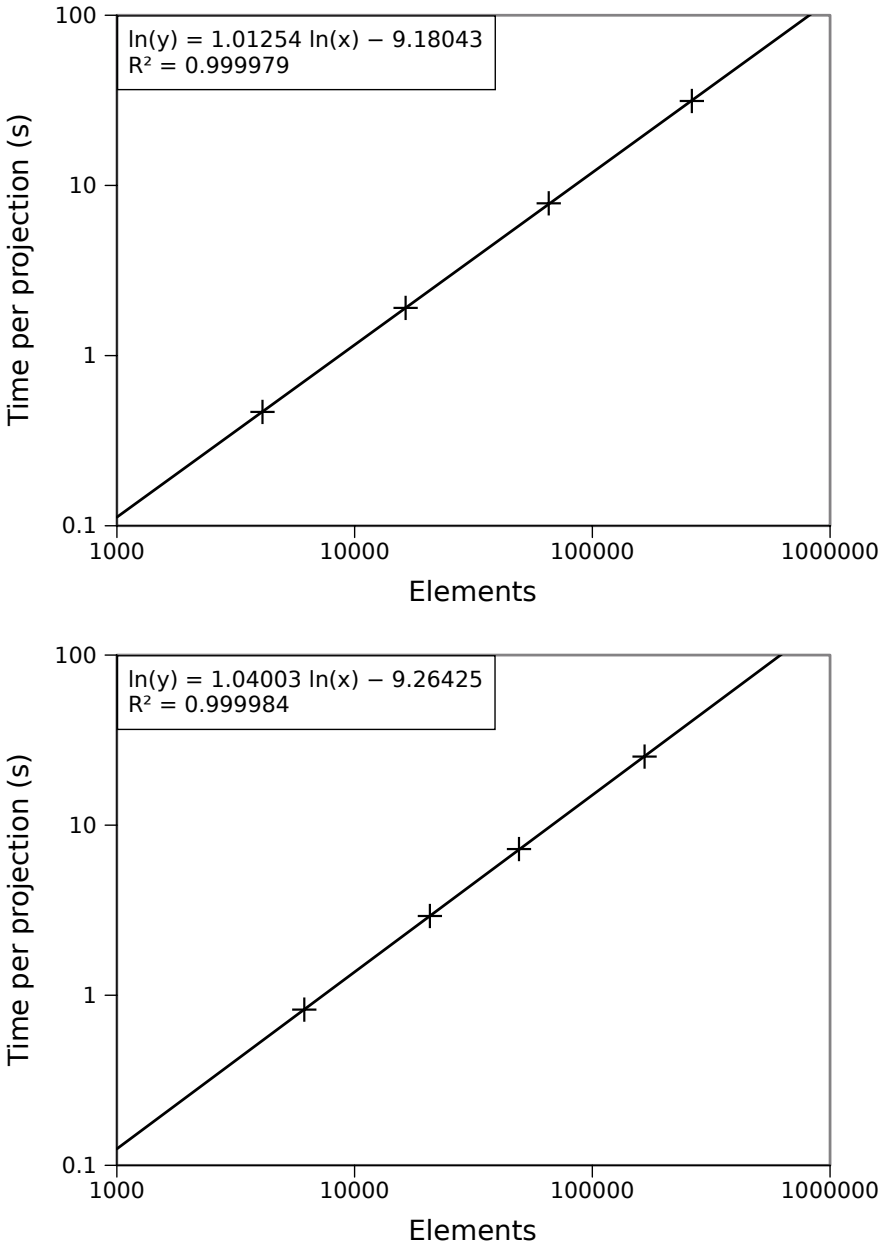
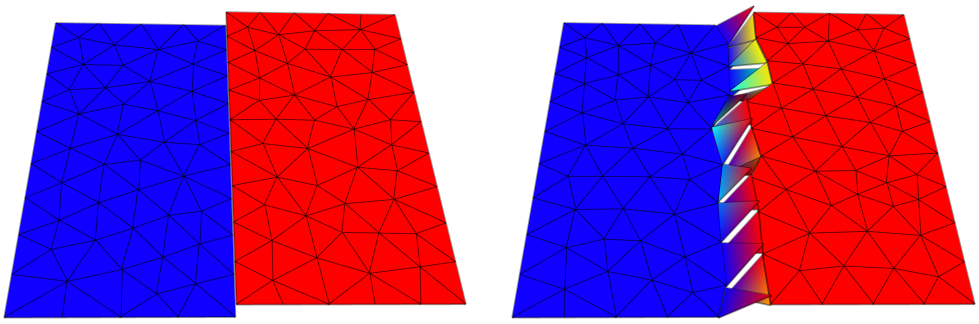


Figure 3.4: Profiling results for the experiment described in section §3.6.1 in two (above) and three (below) dimensions. The time taken by the algorithm is linear in the size of the inputs. In two dimensions, the algorithm takes approximately 0.12 ms of CPU time per element in the mesh; in three dimensions, 0.15 ms of CPU time per element.



(a) (b)

Figure 3.5: $z = \psi(x, y)$ on (a) the donor mesh and (b) the target mesh.

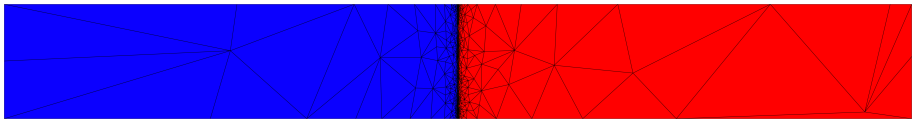


Figure 3.6: Initial condition for temperature for the two-dimensional lock exchange problem. The bounds on temperature are $[-0.5, 0.5]$.

The result can be seen in figure 3.5. The interpolant is discontinuous around the region of the line $x = 0.5$, as a result of the resolution on the target mesh not being aligned with the discontinuity of the original function. The integral of ψ is conserved to one part in 10^{16}, within 64-bit floating point precision. The supermesh consists of 1600 elements. As triangles are convex polygons, the intersection of two triangles is a convex polygon with at most 6 vertices, a convex hexagon. Since any convex hexagon can be meshed with 4 triangles, only 4 triangles of the supermesh ever exist at once.

3.6.3 Two-dimensional lock exchange

With the development of Galerkin projection, combinations of techniques that were previously infeasible become not just possible but useful.

To demonstrate this, a two-dimensional lock exchange simulation was conducted in the domain $\Omega = [0, 0.8] \times [0, 0.1]$. The lock exchange problem consists of two fluids of different density that are initially separated by a gate or 'lock'. The gate is removed at $t = 0$ and the buoyancy force drives

two gravity currents that propagate in opposite directions, with the denser fluid flowing under the lighter fluid.

The equations solved were the incompressible Navier-Stokes equations subject to the Boussinesq approximation for velocity and pressure, and the advection-diffusion equation for temperature. The equation set was closed with the addition of a linear equation of state. A no-slip boundary condition was enforced at the bottom boundary and no-normal flow was enforced on all other boundaries. The kinematic viscosity coefficient ν was set to 10^{-6}, to give a Reynolds number of 790 by the definition of Özgökmen et al. (2007). The initial condition for temperature (figure 3.6) was set to

$$T(x, y, t = 0) = \begin{cases} -1/2, & \text{if } x < 0.4, \\ 1/2, & \text{otherwise.} \end{cases} \tag{3.6}$$

The velocity and pressure fields were discretised with the mixed continuous-discontinuous Galerkin $\mathrm{P1_{DG}}$-P2 finite element discretisation (Cotter et al., 2009b). This element pair promises to be excellent for geophysical applications, as it satisfies the Ladyzhenskaya-Babuška-Brezzi stability condition and excellently represents geostrophic balance (Cotter et al., 2009a). The advection-diffusion equation is discretised using a control volume scheme with the Sweby limiter (Sweby, 1984; Wilson, 2009). Crank-Nicolson timestepping was used with a timestep Δt of 0.025 s. The simulation was terminated after 24 seconds of simulation time. The mesh was adapted every 5 timesteps with the algorithm of Vasilevskii and Lipnikov (1999). The metric was computed to control the L_∞ norm of the interpolation error associated with temperature. The target interpolation error was set to 0.025 °C. A minimum edge length of 10^{-4} m and a maximum edge length of 0.5 m were enforced on the metric.

This simulation requires Galerkin projection for two reasons. Firstly, collocation interpolation is undefined for the discontinuous velocity field, so if one wishes to combine adaptive remeshing with a discontinuous discretisation then an alternative to collocation interpolation necessary. Here, Galerkin projection is used for velocity. Secondly, the discretisation of the advection-diffusion equation preserves the integral to machine precision and the bounds approximately; therefore, it is highly desirable to preserve these properties through the interpolation step. Therefore, the bounded Galerkin

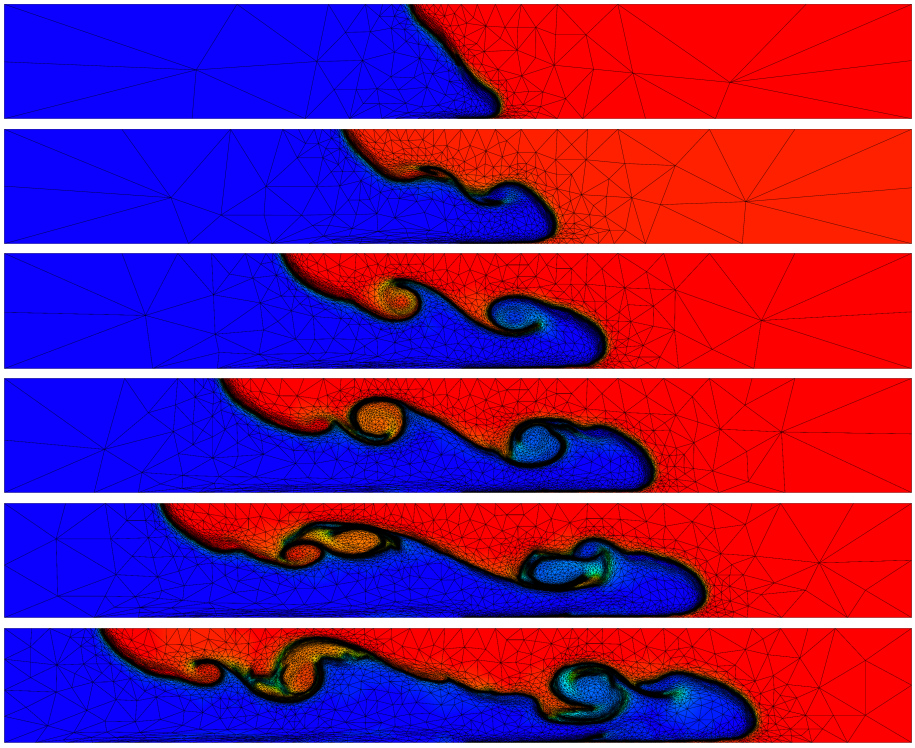


Figure 3.7: Simulation results for the lock exchange at times 4, 8, 12, 16, 20, 24 s. The bounds for temperature are $[-0.5, 0.5]$. Note the mesh adapting to resolve the temperature interface. The combination of adaptive remeshing and discontinuous Galerkin methods would be impossible with collocation interpolation.

projection of §2.3.4 is employed for temperature.

Simulation results are displayed in figure 3.7. The simulation spends 1.5% of runtime identifying intersecting elements using the algorithm of chapter 4, and 7.5% of runtime in the Galerkin projections. The adaptive remeshing algorithm ensures that the mesh resolution is appropriately placed to resolve the temperature interface. The velocity of the front is in good agreement with the benchmark data of Härtel et al. (2000). As can be seen in figure 3.8, both the discretisation and the interpolation step are conservative. Galerkin projection supplies a key component in rendering possible the combination of discontinuous discretisations and adaptive remeshing.

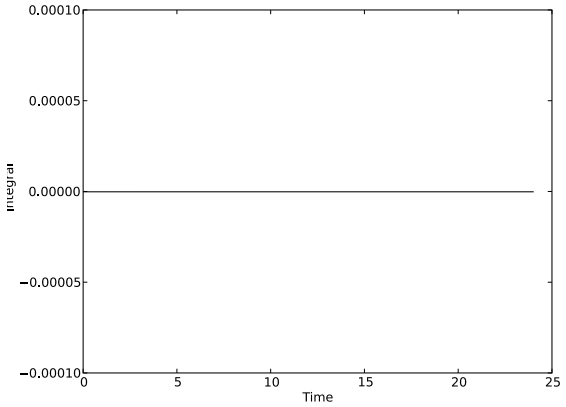


Figure 3.8: Integral of the temperature field for the two-dimensional lock exchange simulation.

3.6.4 Three-dimensional annulus

To demonstrate the application of this technique to a three-dimensional problem, a laboratory-scale annulus was simulated. The equations solved were the incompressible Navier-Stokes subject to the Boussinesq approximation for velocity and pressure and the advection-diffusion equation. The $P1_{DG}$-P2 finite element was chosen for the velocity-pressure discretisation, and the temperature was represented with $P1_{DG}$ basis functions. The simulation was conducted using annulus parameters as described in table 1 of Read (2003). A tanh-stretched mesh as described in Farnell and Plumb (1975) was used for the simulation, via the decomposition of hexahedra into tetrahedra (Tanizume et al., 1990). The model was initialised using output from the Met. Office/Oxford Rotating Annulus Laboratory Simulation (MORALS) finite difference annulus model, run in axisymmetric mode (Farnell and Plumb, 1975; Hignett et al., 1985).

Figure 3.9 shows the system state after 2000s of simulation time, after which a wavenumber three baroclinic wave has developed. In order to demonstrate the application of discontinuous interpolation to this simulation, a target mesh was generated via local mesh modifications as in Pain et al. (2001), with a metric tensor derived from the Hessian of projections of the discontinuous simulation fields to a continuous mesh.

The $P1_{DG}$ temperature field of this simulation was interpolated between

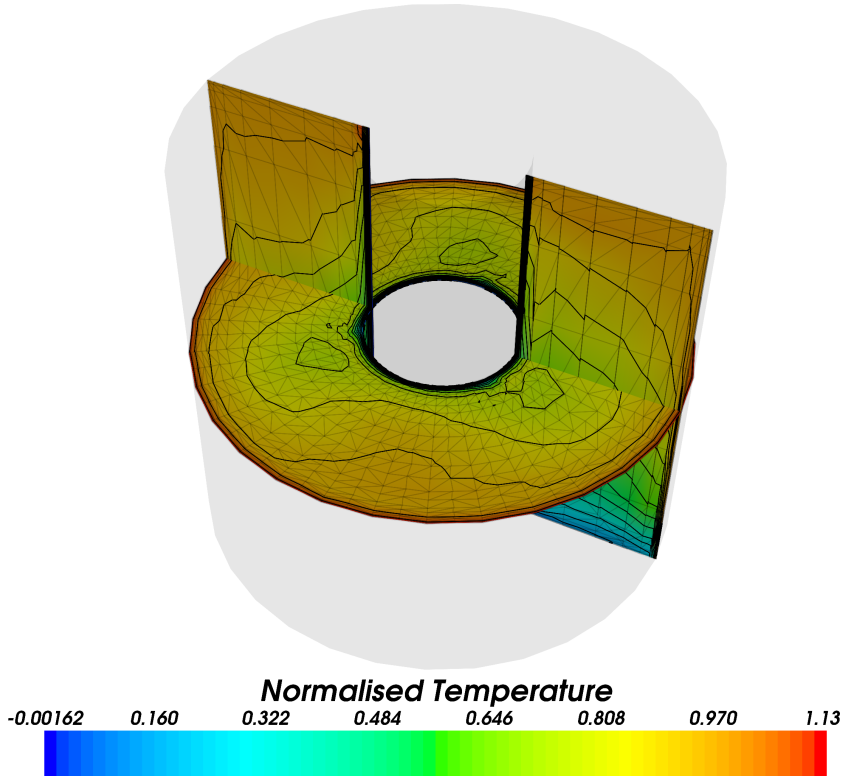
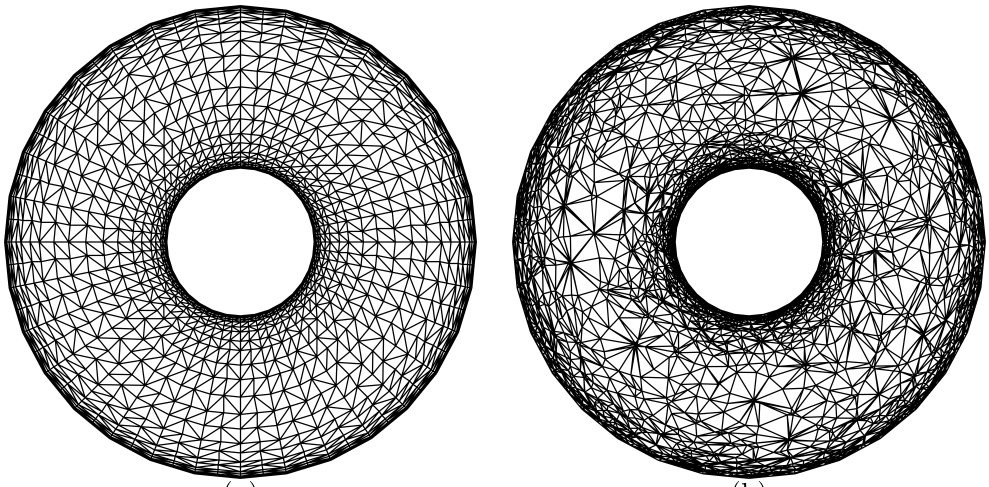


Figure 3.9: Horizontal and vertical slices through the annulus simulation after 2000s. Normalised temperature $\bar{T} = T/(T_B - T_A)$, where T_B and T_A are the temperatures of the outer and inner walls respectively. A wavenumber 3 solution can be seen.



(a) (b)

Figure 3.10: Mid-height horizontal slices though (a) the donor mesh and (b) the target mesh, used in the interpolation of the discontinuous temperature field. Note that the slice on the right is though a fully unstructured three-dimensional mesh. Hence, the apparent quality of the two-dimensional slice is not an indication of the quality of the full volume mesh.

(a) (b)

Figure 3.11: Normalised temperature $\bar{T} = T/(T_B - T_A) = 0.6$ isotherm, (a) before and (b) after interpolation, where T_B and T_A are the temperatures of the outer and inner walls respectively.

the meshes using algorithm 3.2. Figure 3.11 shows an isotherm before and after the interpolation. In particular, it can be seen that the transition between the boundary layer and the fluid bulk is represented on the donor mesh with a clear discontinuity as a result of the field being under-resolved in these regions. This discontinuous information is preserved following the interpolation to the target mesh, with a reduction in the jumps between elements as a result of the target mesh having resolution concentrated towards the boundaries.

The volume integral of the temperature field on the target mesh was conserved to one part in 10^{14}, with the error attributable to roundoff error in both the supermesh construction and the solution of equation (2.18). The donor and target meshes contained 23,232 and 136,051 elements respectively, with a total of 7,934,047 elements in the entire supermesh. The supermesh was constructed locally. The maximum number of supermesh elements stored in memory at once was 69, demonstrating the ability of this approach to be applied to larger problems.

3.6.5 Three-dimensional water collapse

Simulations of multimaterial flows are numerically challenging. The interfaces of the material volume fractions recording the materials are sharp and anisotropic, and this must be reflected in the mesh upon which these flows are discretised. Furthermore, the discretisation must be conservative and bounded; otherwise nonphysical phenomena such as mass exchange may occur.

Adaptive remeshing was applied to an unsteady multimaterial simulation of a water column collapsing in air under gravity. The equations solved were the incompressible Navier-Stokes equation and the advection equation for the evolution of the material volume fraction.

The simulation was conducted in the domain $\Omega = [-0.5, 0.5] \times [-0.5, 2] \times [-0.5, 0.5]$. A material volume fraction representing water is initialised to be 1 in the region $[-0.5, -0.25] \times [-0.5, 0] \times [-0.5, 0]$ and zero elsewhere. No-normal flow was imposed on velocity on all boundaries except for the top. At the top, a homogeneous Neumann boundary condition was imposed on velocity, and a homogeneous Dirichlet boundary condition was imposed on pressure. The $\mathrm{P0_{DG}}$-$\mathrm{P1_{CV}}$ element pair was used for the velocity-pressure

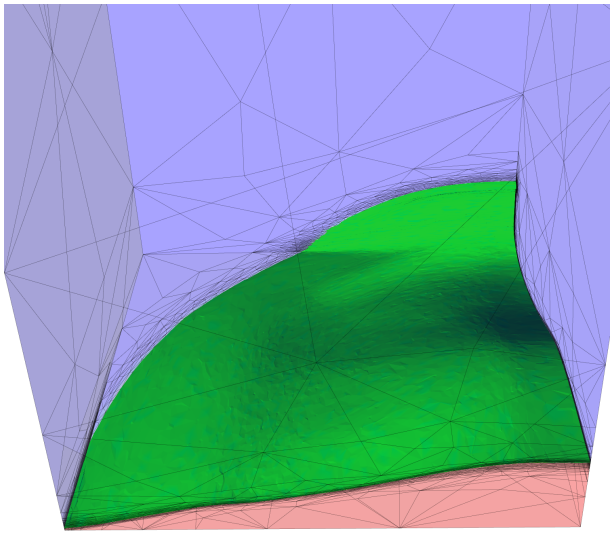


Figure 3.12: Isosurface of the material volume fraction field at time $t = 0.43$.

discretisation; the HyperC control volume face value algorithm was used for the advection equation (Leonard, 1991). The motivation for this element pair and the discretisation is described in Wilson (2009). Crank-Nicolson timestepping was used, with an initial timestep of $\Delta t = 2 \times 10^{-4}$. The timestep was adapted through the simulation to maintain a CFL number of 2.5. The simulation was terminated at $t = 0.9$. The material volume fraction representing water was chosen as the field to guide adaptivity. A three-dimensional extension of the metric formation algorithm of Formaggia and Perotto (2003); Micheletti and Perotto (2006) was used to control the H_1-seminorm of the interpolation error; the target error was chosen to be $\tau = 25$. A minimum edge length of 0.001 was enforced to constrain the adaptive algorithm. The mesh was adapted every 10 timesteps. To spread resolution ahead of the dynamics, the metric tensor formed was advected forward for one adaptivity period and superimposed with itself. This has the effect of extending resolution to where the interface will be over the course of the adaptivity period. Since the velocity field was discontinuous, Galerkin projection was used to interpolate it from each previous mesh to the corresponding adapted mesh. As conservation and boundedness of the material volume fraction are crucial, the bounded Galerkin projection was employed for this field.

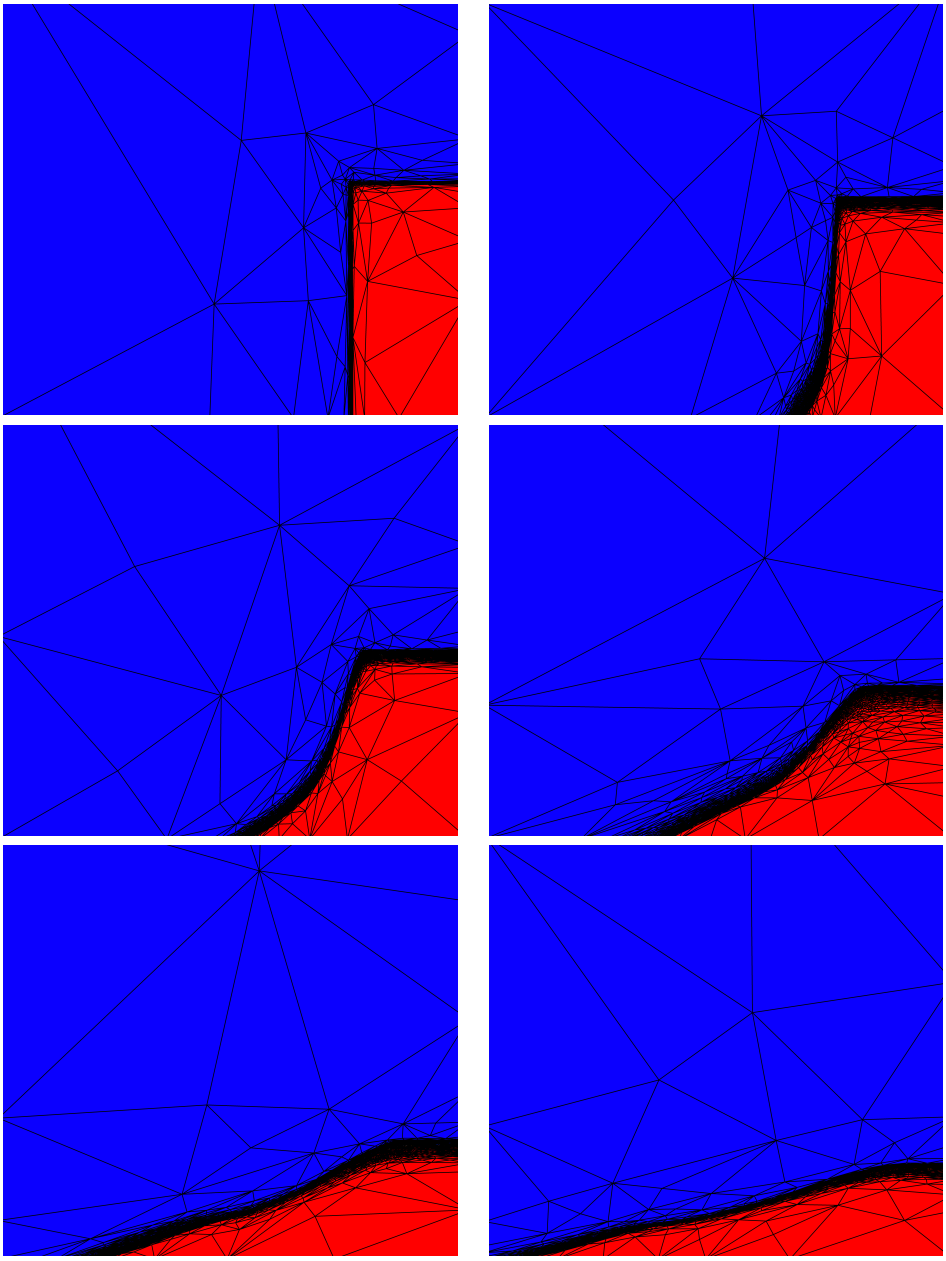
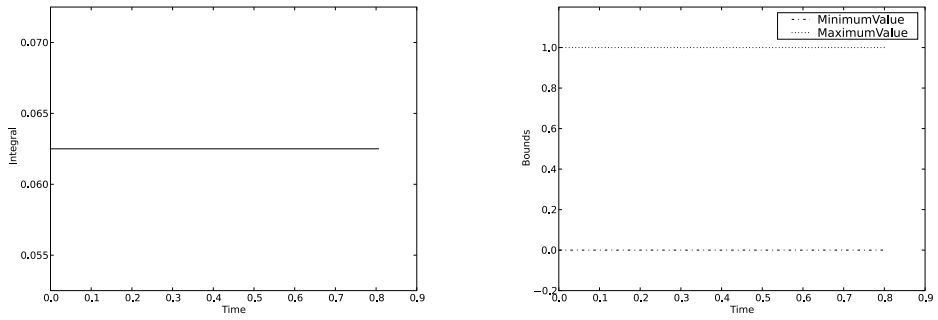


Figure 3.13: The material volume fraction and mesh at times $t = 0, 0.09, 0.17, 0.25, 0.33, 0.39$.



(a) (b)

Figure 3.14: (a) Integral and (b) bounds of the material volume fraction field for the three-dimensional water column collapse. Note that both the discretisation and interpolation are conservative and bounded.

Simulation results are displayed in figures 3.12 and 3.13. The simulation spent 0.4% of runtime identifying intersecting elements using the algorithm of chapter 4, and spent 12.3% of runtime in the Galerkin projections. Again, the adaptive remeshing ensures that the mesh resolution is focussed on the material interface. Maintaining this accuracy in the material interface would be prohibitively expensive on any fixed mesh. As can be seen in figure 3.14, both the discretisation and the interpolation step preserve the integral and bounds of the material volume fraction. This combination of a discontinuous discretisation, conservative and bounded advection, and adaptive remeshing would have been impossible if not for the development of Galerkin projection.

3.7 Conclusions

A robust, efficient supermesh construction algorithm has been proposed. With the development of local supermeshing, the algorithm can scale to larger problem sizes than those feasible with global supermeshing. Several examples have been shown which demonstrate the practicality of this approach in two and three dimensions. The availability of this algorithm makes possible the exploitation of Galerkin projection between unrelated unstructured meshes, and thus has many applications in model initialisation, adaptive remeshing, and other tasks requiring interpolation.

Chapter 4

Intersection reporting between meshes of connected domains

Abstract

Given two meshes $\mathcal{T}_R$ and $\mathcal{T}_B$ (coloured *red* and *blue*) of a connected finite domain $\Omega \subset \mathbb{R}^d$, an efficient algorithm for reporting all intersecting pairs of elements is given. The algorithm does not assume that $\Omega \subset \mathbb{R}^2$, nor that Ω is simply connected, nor that $\mathcal{T}_R$ or $\mathcal{T}_B$ are simplicial. The algorithm performs $O(|\mathcal{T}_B|+k)$ intersection tests, where k is the number of intersections. If an initial seed is supplied as input, then the algorithm performs $O(k)$ intersection tests.

This chapter expands upon Farrell and Maddison (2009).

Contents

4.1 Introduction

Given two sets $\mathcal{T}_R$ and $\mathcal{T}_B$ of geometric objects, coloured *red* and *blue*, a frequently arising problem in computational geometry is to identify or count the bichromatic intersections: finding pairs $(K_R, K_B) \in \mathcal{T}_R \times \mathcal{T}_B$ such that $K_R \cap K_B \neq \emptyset$ (Agarwal and Sharir, 1990; Chazelle, 1993; Chan, 1994; Mount, 1997; Gupta et al., 2005). This is intimately related to the problem of computing intersecting pairs of elements within a single set (Shamos and Hoey, 1976; Bentley and Ottmann, 1979; Hopcroft et al., 1983; Reichling, 1988; Gupta et al., 1999; Agarwal et al., 2001; Ezra and Sharir, 2004). In particular, this problem arises in the context of local supermeshing for the assembly of the Galerkin projection (see chapter 3).

In this work, an algorithm for the bichromatic intersection problem is proposed where $\mathcal{T}_R$ and $\mathcal{T}_B$ are not merely sets of polytopes in $\mathbb{R}^d$, but meshes of a finite, connected domain Ω. By developing an algorithm for the more specific case, the mesh connectivity information can be exploited to minimise the number of intersection tests performed. It is also possible to easily extend the algorithm to non-connected domains. In two dimensions, a mesh is a special kind of planar subdivision, which was dealt with for simply connected convex domains by Guibas and Seidel (1987) and for simply connected possibly non-convex domains by Finke and Hinrichs (1995).

The algorithm proposed here is based upon an advancing front approach (George, 1971; Lo, 1985; Löhner and Parikh, 1988; Peraire et al., 1988; Bonet and Peraire, 1991). The closest related algorithm is that presented in Löhner

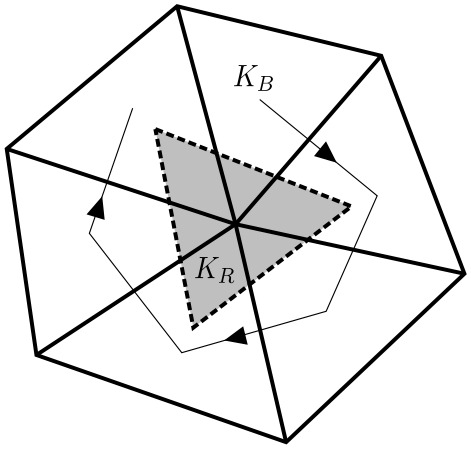


Figure 4.1: The idea behind the intersection finding algorithm. Suppose it is desired to compute the intersections in $\mathcal{T}_B$ for a given K_R (dashed lines). Starting from a seed element K_B, the algorithm advances a front through the elements of $\mathcal{T}_B$ until all the intersecting elements have been found. The seed element K_B is found by advancing a front through $\mathcal{T}_R$.

(1995a) which gives an algorithm for computing an intersecting element in $\mathcal{T}_R$ for each node in $\mathcal{T}_B$.

4.2 Intersection reporting by advancing fronts

The fundamental idea of the algorithm is that, for a given element $K_R \in \mathcal{T}_R$, if at least one intersecting $K_B \in \mathcal{T}_B$ is known, then it is possible to compute the set of all intersecting elements in $\mathcal{T}_B$ by searching in an advancing front of elements around K_B (figure 4.1). This follows from the connectedness of the set of intersections. This information can be used to start the search front for the neighbours of K_R, since any neighbour of K_R will necessarily intersect with at least one of the elements in $\mathcal{T}_B$ that intersect with K_R. Therefore, the algorithm consists of a traversal through $\mathcal{T}_R$, with the intersections of each $K_R \in \mathcal{T}_R$ determined by an advancing front algorithm through $\mathcal{T}_B$.

Throughout, the existence of a suitable intersection predicate is assumed. For the usual case where both meshes are composed of convex elements, this problem is discussed in Chazelle and Dobkin (1980); Dobkin and Kirkpatrick (1983, 1985, 1990). If both meshes are composed of triangles, the triangle

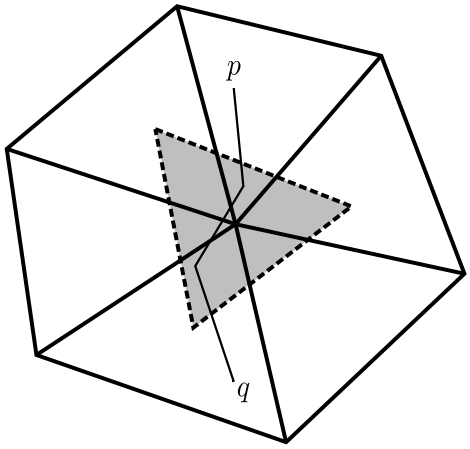


Figure 4.2: For a given $K_R \in \mathcal{T}_R$ (dashed lines, interior shaded), the corresponding set of intersecting elements in $\mathcal{T}_B$, I_{K_R}, (solid lines) forms a connected subdomain of Ω (lemma 4.1). A path is constructed between any two points p and q by routing through K_R.

intersection predicate given in Möller (1997) may be used; if both meshes are composed of tetrahedra, the tetrahedron intersection predicate given in Ganovelli et al. (2002) may be used. At the cost of generating false positives, the intersection predicate may be that the axis-aligned bounding boxes of the elements intersect. This predicate is a necessary but not sufficient condition for intersection. This predicate is much cheaper, and may therefore be efficient if the cost of false positives is low. The algorithm developed below is robust to a predicate which is necessary but not sufficient. In the examples presented later, the bounding box predicate is used.

$K' \in \mathcal{T}$ is a *neighbour* of $K \in \mathcal{T} \iff K' \cap K \neq \emptyset$. Note that this includes the case where K' and K share a face, edge, or node. Let the neighbourhood of K, $N(K)$, be all elements of $\mathcal{T}$ satisfying this definition.

To justify the use of advancing fronts in searching for the intersections for a particular K_R, the following result shall be used (see figure 4.2).

Lemma 4.1. *For each $K_R \in \mathcal{T}_R$, the intersecting elements in $\mathcal{T}_B$*

$$I_{K_R} = \{K_B \in \mathcal{T}_B \mid K_R \cap K_R \neq \emptyset\} \tag{4.1}$$

forms a connected subdomain $\Omega_{K_R} \subseteq \Omega$.

Proof. Let $p \in K_B^{(1)} \in I_{K_R}$ and $q \in K_B^{(2)} \in I_{K_R}$ be two points. The goal is to construct a path between them to demonstrate connectedness. By construction, there exist points $s_1 \in K_B^{(1)} \cap K_R$ and $s_2 \in K_B^{(2)} \cap K_R$. Since the elements $K_R, K_B^{(1)}$ and $K_B^{(2)}$ are connected, there exist paths joining p to s_1, s_1 to s_2, and s_2 to q. Therefore, a path between p and q is constructed by concatenating the paths $p\vec{s}_1$, $s_1\vec{s}_2$ and $s_2\vec{q}$ (figure 4.2). Therefore, I_{K_R} is connected. □

The next step is to show that an advancing front sweep around a given intersecting $K_B \in I_{K_R}$ finds all elements of I_{K_R}. This may be seen as a variation of a topological sweep algorithm (Edelsbrunner and Guibas, 1989).

First, the advancing front algorithm for determining the intersections of $K_R \in \mathcal{T}_R$, given some starting $K_B \in I_{K_R}$, is discussed.

Algorithm 4.1. *Advancing front intersection detection.*

1. $I \leftarrow \{K_B\}$
2. $F \leftarrow N(K_B)$
3. *while* $|F| \neq 0$*:*
   a) *remove a neighbour* K' *from* F
   b) *if* $K_R \cap K' \neq \emptyset$*:*
      i. $I \leftarrow I \cup \{K'\}$
      ii. $F \leftarrow F \cup N(K')$
4. *Return* I

Lemma 4.2. *Suppose* $K_B \in I_{K_R}$ *is given. Let* I *be constructed by algorithm 4.1. Then* $I = I_{K_R}$.

Proof. Let $K^* \in I$. Then $K^* \in I_{K_R}$ as $K_R \cap K^* \neq \emptyset$ by construction. So $I \subseteq I_{K_R}$.

Let $K^* \in I_{K_R}$. Suppose $K^* \notin I$. Let $p_1 \in K_R \cap K_B$ and let $p_2 \in K_R \cap K^*$, and let l be a path contained in I_{K_R} joining p_1 and p_2; such a path exists by lemma 4.1. Define L as

$$L = \{K_B \in \mathcal{T}_B \mid l \cap K_B \neq \emptyset\}. \tag{4.2}$$

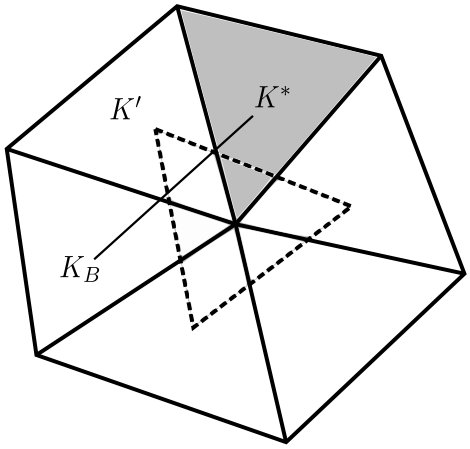


Figure 4.3: Suppose the output I of algorithm 4.1 is not complete, i.e. $I \subsetneq I_{K_R}$. Then it has missed an element K^* (shaded) that neighbours I. Let K' be its neighbour in I. Since K^* and K' are neighbours, K^* is tested for membership of I_{K_R} when K' is added to I. Therefore, $K^* \in I$ and $I_{K_R} \subseteq I$. Since $I \subseteq I_{K_R}$, $I = I_{K_R}$.

L is the elements of $\mathcal{T}_B$ through which l passes. Define the missed elements M as

$$M = L \cap (I_{K_R} \setminus I). \tag{4.3}$$

That is, M is the elements that have been missed by the above algorithm. Suppose M is not empty. There exists at least one $K^* \in M$ such that it has a neighbour $K' \in I$, for one endpoint of the line l is in I (the proof is deferred until lemma 4.3). This is illustrated in figure 4.3. However, K^* must be in I because K^* was appended to the front of neighbours for consideration F when K' was added to I, so K^* was tested for intersection with K_R and added to I. Therefore, no such K^* exists. Therefore $M = \emptyset$, and $I_{K_R} \supseteq I$. □

The proof of the claim used in lemma 4.2 is now given.

Lemma 4.3. *If $I \subsetneq I_{K_R}$, there exist $K' \in I$ and $K^* \in M$ such that K' and K^* are neighbours.*

Proof. It is possible to decompose l into pieces contained in each element

with which it intersects:

$$l = \cup_{K_B \in L}\ l \cap K_B. \tag{4.4}$$

Define a function $f : l \to \{0, 1\}$ by

$$f(x) = \begin{cases} 0 & x \notin I, \\ 1 & x \in I. \end{cases} \tag{4.5}$$

Then f is discontinuous, with jumps occurring at the boundaries of the elemental decomposition of l.

If M is nonempty, then f attains the value 0 at some point. However, f is fixed to be 1 at the endpoint of l, p_2: therefore it must jump in value from 0 to 1 at some elemental boundary. Let K^* be the element upon which f attains 0 and let K' be its neighbour upon which f attains 1. □

Given a *seed element* $K_B \in I_{K_R}$, algorithm 4.1 constructs the whole of I_{K_R}. It can be seen that the algorithm is efficient in the sense that for each K_R, the number of extraneous intersection tests is restricted to the ring of elements in $\mathcal{T}_B$ neighbouring I_{K_R}. Let c be the maximal element neighbourhood degree of $\mathcal{T}_B$, that is

$$c = \max_{K_B \in \mathcal{T}_B} |N(K_B)|. \tag{4.6}$$

Then the number w of intersection tests performed is bounded by

$$w \leq c \cdot |I_{K_R}|. \tag{4.7}$$

and so the work done is $O(|I_{K_R}|)$, linear in the number of intersections.

The question now arises of how to compute the initial seed K_B. Suppose that I_{K_R} has been computed. Let K'_R be a neighbour of K_R. Then by the definition of neighbourhood, there exists a point $p \in K_R \cap K'_R$. As $\mathcal{T}_R$ and $\mathcal{T}_B$ mesh the same domain Ω, it follows that the volume occupied by the elements of I_{K_R} is a superset of the volume occupied by K_R and thus there exists $K'_B \in I_{K_R}$ such that $p \in K'_B$. Then $K'_R \cap K'_B$ and K'_B provides the initial seed for the computation of $I_{K'_R}$.

Therefore, the algorithm sweeps through the mesh $\mathcal{T}_R$ from neighbour to

neighbour, exploiting the known intersections of K_R to provide the seed for the computation of the intersections of K'_R. A seed for the computation of the first intersection set is acquired by a brute force search through the elements of $\mathcal{T}_B$, or it may be supplied as input to the algorithm.

The intersection reporting algorithm is summarised as follows.

Algorithm 4.2. *Intersection reporting between $\mathcal{T}_R$ and $\mathcal{T}_B$.*

1. *Generate an initial seed for $K_R^{(1)}$ by brute-force search through $\mathcal{T}_B$.*
2. *Apply algorithm 4.1 to compute $I_{K_R^{(1)}}$.*
3. *While there exists an unprocessed element in $\mathcal{T}_R$:*
   a) *Move from a processed element K_R to an unprocessed neighbour K'_R.*
   b) *Compute an initial seed by searching through I_{K_R}.*
   c) *Apply algorithm 4.1 to compute $I_{K'_R}$.*

The connectedness of Ω guarantees that it is possible to move from neighbour to neighbour to process the whole of $\mathcal{T}_R$. The computation of the initial seed is linear in the number of elements in $\mathcal{T}_B$. Let k be the total number of intersections:

$$k = \sum_{K_R \in \mathcal{T}_R} |I_{K_R}|. \tag{4.8}$$

The number of intersection tests performed in the loop (3) of algorithm 4.2, W, is given by

$$W = \sum_{K_R \in \mathcal{T}_R} O\left(|I_{K_R}|\right) = O\left(\sum_{K_R \in \mathcal{T}_R} |I_{K_R}|\right) = O(k). \tag{4.9}$$

So the number of intersection tests performed is $O\left(|\mathcal{T}_B| + k\right)$. If an initial seed can be provided as input to the algorithm, then the number of intersection tests performed formally reduces to $O(k)$. If the two meshes are the input and output to a mesh optimisation algorithm, an initial seed can generally be provided with minimal changes to the optimisation library.

Note that this algorithm can be extended to non-connected domains, by applying it to each connected subdomain in turn.

A problem can arise with the use of a necessary but not sufficient intersection predicate, such as the bounding box intersection test. If the initial

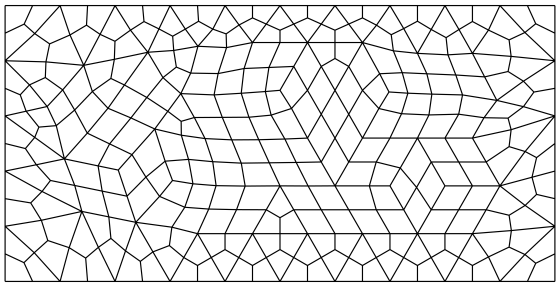


Figure 4.4: An example quadrilateral mesh used in the scaling analysis of the advancing front intersection finding algorithm.

seed K_B chosen does not actually intersect with K_R, then the algorithm is not guaranteed to find all elements of I_{K_R}. One way to circumvent this problem is to use all of the intersecting elements of a processed neighbour $I_{K'_R}$, rather than merely taking the first intersecting element in the set.

4.3 Examples

The number of intersection tests performed in intersection reporting using the advancing front algorithm was tested for its scaling with problem size in both two and three dimensions, using quadrilateral and tetrahedral elements.

4.3.1 Two-dimensional domain

The reporting algorithm was applied to the two-dimensional rectangular domain $[0, 2] \times [0, 1]$. Unstructured quadrilateral mesh pairs of comparable size were generated using the mesh generator GiD[1], with element counts ranging from 224 to 1,044,954. The advancing front intersection reporting algorithm was applied using a bounding box intersection predicate, and the resulting set of intersections verified against alternative intersection reporting algorithms: brute force where computationally feasible, and an R-tree spatial indexing algorithm when the brute force approach became impractical (Guttman, 1984; Manolopoulos et al., 2005). As counting the number of predicates performed by the R-tree does not give a fair estimate of the work it is doing, comparison against the R-tree is deferred until section §4.3.3.

[1] `http://gid.cimne.upc.es/`

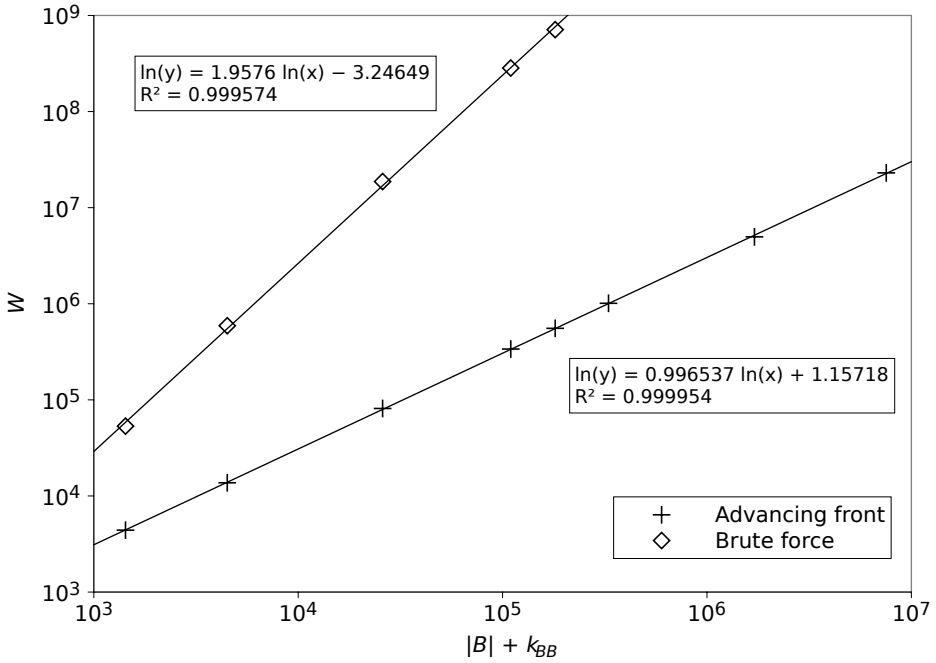


Figure 4.5: Number of intersection tests performed, W, against target mesh element count $|\mathcal{T}_B|$ plus the number of bounding box element intersections k_{BB}, for the advancing front intersection reporting algorithm and for a simple brute force approach applied to unstructured quadrilateral meshes in a rectangular domain.

The number of bounding box intersections k_{BB} was computed by summing the number of intersections for each element in $\mathcal{T}_R$.

Figure 4.5 shows W against $(|\mathcal{T}_B| + k)$ for each pair of meshes for the advancing front algorithm and, where available, the corresponding number of intersection tests performed in the brute force intersection reporting. The brute force algorithm exhibits quadratic scaling of W with $(|\mathcal{T}_B| + k)$ to within 2.5%. The advancing front exhibits linear scaling of W with $(|\mathcal{T}_B|+k)$ to within 1%, as expected.

4.3.2 Three-dimensional multiply-connected domain

In order to demonstrate that the algorithm extends to higher dimensions and multiply-connected domains, the algorithm was applied to a domain consisting of a cube of unit size, with a cubic region of half-unit size removed from its centre to form a cubic shell, as shown in figure 4.6. Unstructured

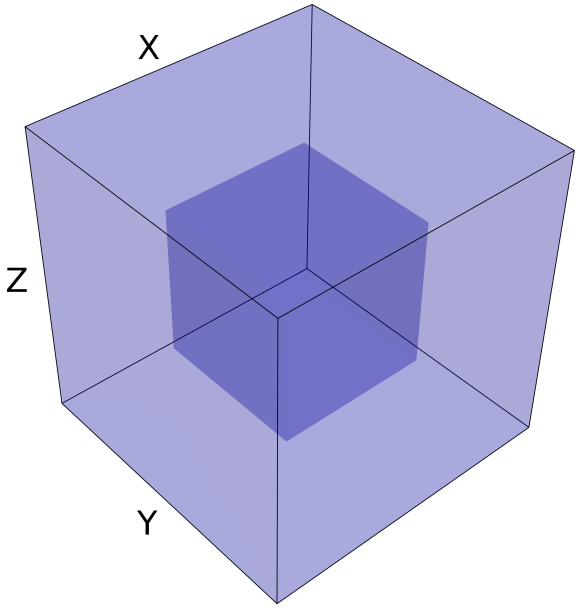


Figure 4.6: The geometry of the cubic shell: the volume contained between two concentric cubes, one of unit size and one of half unit size.

tetrahedral meshes of varying node counts were generated by the addition of random points throughout the shell with a uniform distribution, followed by constrained Delaunay tetrahedralisation using the three-dimensional mesh generator Tetgen[2] (Si and Gärtner, 2005).

Pairs of meshes were generated for input shell nodes counts of $2^n, n \in \{8, \ldots, 15\}$. For each pair, the advancing front intersection reporting algorithm was applied using a bounding box intersection predicate. As in the two-dimensional example, the resulting set of intersections was verified for completeness by comparison against alternative intersection reporting algorithms. For node counts in the range 256 to 4096, the set of intersections was verified against a brute force reporting algorithm, and for node counts of 8192 and above was verified against an R-tree spatial indexing algorithm. The number of bounding box intersections k_{BB} was computed by summing the number of intersections for each element in $\mathcal{T}_R$.

Figure 4.7 shows W against $(|\mathcal{T}_B| + k_{BB})$ for each pair of meshes for the advancing front algorithm and, where available, the corresponding number of intersection tests performed in the brute force intersection reporting. The

[2] `http://tetgen.berlios.de`

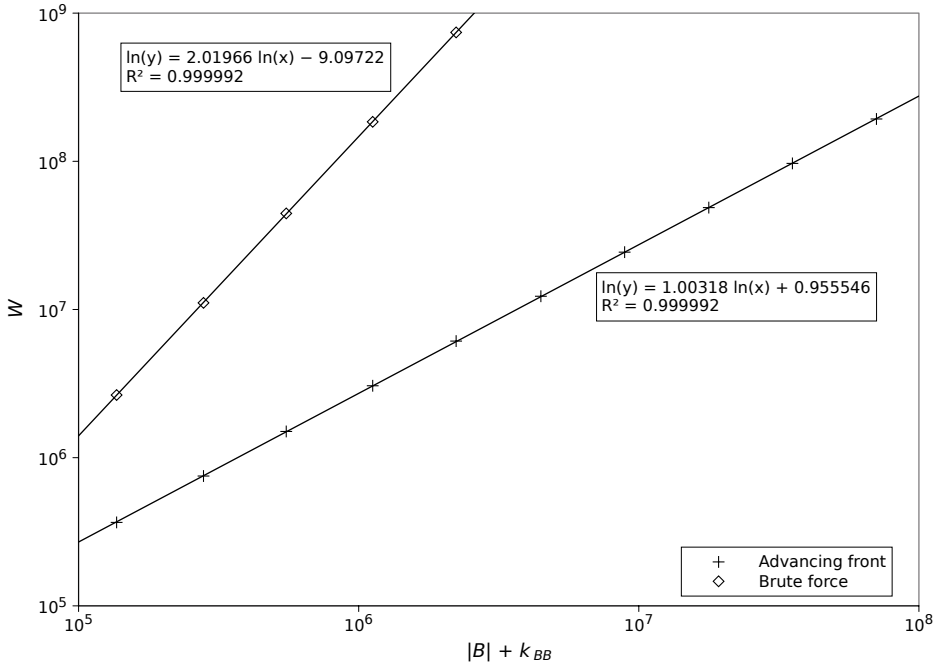


Figure 4.7: Number of intersection tests performed, W, against target mesh element count $|\mathcal{T}_B|$ plus the number of bounding box intersections k_{BB}, for the advancing front intersection reporting algorithm and for a simple brute force approach applied to unstructured tetrahedral meshes in a three-dimensional multiply-connected domain.

brute force algorithm exhibits quadratic scaling of W with $(|\mathcal{T}_B| + k_{BB})$ to within 1%. The advancing front exhibits linear scaling of W with $(|\mathcal{T}_B| + k_{BB})$, also to within 2%, as expected.

4.3.3 Comparison against the R-tree algorithm

An R-tree is a data structure commonly used in computer science for spatial indexing. Let U be a set of objects in space: $u \subset \mathbb{R}^d \; \forall \; u \in U$. From the bounding boxes of each $u \in U$, an R-tree may be constructed, which allows for the efficient resolution of spatial queries, such as the identification of objects inside or intersecting with a given bounding box. The R-tree itself consists of a tree of hierarchically nested, possibly overlapping, bounding boxes. It is similar to a quadtree or octree, but more flexible, which is advantageous when considering meshes composed of anisotropic elements whose length scales may vary by orders of magnitude. When a query is performed, the search algorithm starts at the root of the tree and checks each of its children for intersection against the query volume. The search is then recursively applied on all of the child nodes which intersect with the query volume. For more details, see Guttman (1984) and Manolopoulos et al. (2005). Note that the R-tree algorithm does not exploit any connectivity graph between the objects it indexes (as such a connectivity graph of U does not, in general, exist).

Grandy (1999) employs an algorithm which is similar in spirit to an R-tree approach to filter the number of pairs of elements to be tested for intersection, but this connection is not explicitly mentioned in the work. Bonet and Peraire (1991) present a tree-based algorithm for geometric intersection and searching problems and apply this to advancing front mesh generation.

The previous examples only compare the number of predicates performed against the brute-force approach. This is because of the fundamentally different nature of the R-tree algorithm: it makes no sense to count just the number of predicates performed, as constructing and querying the R-tree involves significant computational work that is not captured by such a diagnostic. Therefore, to compare the advancing front approach with an R-tree algorithm, the CPU time taken by the advancing front and R-tree algorithms for the series of meshes described above was measured. The experiment was performed on a dual-processor dual-core Intel Xeon 2.8 GHz machine with

$\lvert\mathcal{T}_B\rvert + k$	Advancing front	R-tree	Ratio
1429	0.008	0.03	3.75
4497	0.02	0.08	4.00
25880	0.13	0.59	4.53
109630	0.52	2.59	4.98
180880	0.83	4.20	5.06
329833	1.56	8.00	5.13
1706998	7.50	65.50	8.73
7535744	31.97	249.83	7.81

Table 4.1: CPU times taken by the advancing front and R-tree algorithms for the series of unstructured quadrilateral meshes. The ratio column reports the time taken by the R-tree algorithm divided by the time taken for the advancing front algorithm. The advancing front algorithm is clearly superior, being approximately 8 times faster for the larger examples.

2G of RAM. The CPU time was measured with the `cpu_time` intrinsic of Fortran 90. The R-tree implementation was provided by version 1.3.2 of the SpatialIndex[3] C++ library.

The results are shown in table 4.1 and table 4.2. As can be seen, the advancing front algorithm is considerably more efficient. This accords with expectations, as the advancing front algorithm exploits more information about the structure of its inputs; the R-tree approach does not use the connectivity of the meshes to reduce its computational burden.

4.4 Conclusions

In this work, an intersection reporting algorithm for the identification of bichromatic intersections between meshes of connected domains has been proposed. This problem has applications in a range of areas, including computational geometry, computer graphics, and numerical simulation. In particular, the algorithm is a key component of any efficient implementation of Galerkin projection for interpolation between unrelated meshes. The algorithm is output-sensitive, in that the number of intersection tests scales with the number of intersections to be reported.

[3] `http://trac.gispython.org/spatialindex/wiki`

$\lvert\mathcal{T}_B\rvert + k$	Advancing front	R-tree	Ratio
21680	0.05	0.08	1.60
54910	0.13	0.24	1.85
136702	0.36	0.73	2.02
279220	0.74	1.82	2.46
551238	1.55	4.44	2.86
1122063	3.34	11.5	3.44
2227674	6.83	27.21	3.98
4470779	14.67	65.81	4.49
8900131	30.09	164.92	5.48
17557563	61.77	432.72	7.00
34912502	121.57	1001.54	8.24
69660697	250.50	3036.22	12.12

Table 4.2: CPU times taken by the advancing front and R-tree algorithms for the series of unstructured tetrahedral meshes. The ratio column reports the time taken by the R-tree algorithm divided by the time taken for the advancing front algorithm. The advancing front algorithm is clearly superior, being more than 12 times faster for the largest example.

Chapter 5

Diagnostics of adaptive simulations

Abstract

Adaptive remeshing can greatly enhance the accuracy of a numerical simulation, given a fixed amount of computational resources. However, the meshes on which the simulation data are represented vary greatly through time, complicating the computation of diagnostics of these simulations and the analysis of the results. Two new approaches to this problem are offered. By exploiting the efficient algorithms given in previous chapters for the construction of supermeshes, the first explicitly constructs a function space which is a superset of the function spaces of the meshes considered. This can be used to exactly compute sums, differences and averages of functions (up to roundoff error). The second computes a mesh suitable for the common interpolation of fields in the input function spaces; this approach has a significantly reduced cost relative to the first approach when exactness is unnecessary. Several examples of both approaches are given.

A publication derived from this chapter is in preparation.

Contents

5.1 Introduction

Let $q(x,t)$ be the numerical solution of some system of partial differential equations equipped with suitable initial and boundary conditions. Take as a desired model output the computation of the difference between the solution at some time T and the initial condition:

$$\psi(x,T) = q(x,T) - q(x,0). \tag{5.1}$$

If no adaptive remeshing is employed, computing ψ is trivial; computing it is merely a loop over the nodes of the (static) mesh. In the case where non-hierarchical adaptive remeshing is employed, such as the method described in Pain et al. (2001), $q(x,T)$ and $q(x,0)$ may be stored on entirely different meshes and so the computation of their difference is no longer obvious.

Throughout this chapter, it is assumed that for any mesh, the basis functions Φ are such that for a given element K, any refinement (subdivision) of K can also represent the basis functions associated with K exactly. For example, both continuous and discontinuous Lagrange polynomials satisfy this refinement property. Let $\mathcal{V}_0$ be the function space associated with the

mesh at time 0, and let $\mathcal{V}_T$ be the function space associated with the mesh at time T. The problem above is then restated as computing the difference between two functions $q(x,0) \in \mathcal{V}_0$ and $q(x,T) \in \mathcal{V}_T$.

One approach is to interpolate the data onto one of the meshes, and to compute the desired diagnostics on it. However, in general, $\mathcal{T}_0$ and $\mathcal{T}_T$ will be entirely different and the interpolation error introduced will pollute the diagnostic computed. If the solution must be represented exactly, it is therefore necessary to compute a common superspace of $\mathcal{V}_0$ and $\mathcal{V}_T$. If an exact answer is unnecessary, a common mesh suitable for representing functions from both function spaces must still be computed, i.e. compute a mesh $\mathcal{T}_C$ such that the interpolation error from both $\mathcal{V}_0$ and $\mathcal{V}_T$ to $\mathcal{V}_C$ is minimised. These are the problems addressed in this chapter.

5.2 Forming a function superspace

To form a common function superspace of two function spaces $\mathcal{V}_1$ and $\mathcal{V}_2$, it is necessary and sufficient to form a function space $\mathcal{V}_S$ such that the basis functions of $\mathcal{V}_1$ and $\mathcal{V}_2$ may be represented exactly in $\mathcal{V}_S$. Since the basis functions are associated with the elements of $\mathcal{T}_i, i = 1, 2$, the mesh associated with $\mathcal{V}_S$ must preserve the structure of the elements of both input meshes: that is, given an element K in either input mesh, it must be possible to represent K exactly as the union of elements of the mesh $\mathcal{T}_S$. The supermesh of $\mathcal{T}_1$ and $\mathcal{T}_2$ provides such a decomposition.

Recall the definition of a supermesh given in definition 2.1. Let $\mathcal{N}_i$ be the set of nodes of $\mathcal{T}_i$. Let K be an element of $\mathcal{T}_i$ and let K_S be an element of $\mathcal{T}_S$. Define a *supermesh* $\mathcal{T}_S$ of $\{\mathcal{T}_1, \mathcal{T}_2\}$ as a mesh of Ω such that:

- $\mathcal{N}_S \supseteq \mathcal{N}_1 \cup \mathcal{N}_2$;
- $\mu(K_S \cap K) \in \{0, \mu(K_S)\} \; \forall \; K_S \in \mathcal{T}_S, K \in \mathcal{T}_i$;

where μ is the d-dimensional measure (length, area or volume function). The utility of this construction is that it allows us to decompose elements in $\mathcal{T}_i$ as the union of elements of $\mathcal{T}_S$. $\mathcal{T}_S$ is equipped with the same order basis functions as $\mathcal{V}_1$ and $\mathcal{V}_2$ to form $\mathcal{V}_S$.

Lemma 5.1. *Let ϕ be a basis function of $\mathcal{T}_i$. Let $\mathcal{T}_S$ be a supermesh of $\{\mathcal{T}_i\}$. Then $\phi \in \mathcal{V}_S$ and therefore $\mathcal{V}_S \supseteq \mathcal{V}_1 \cup \mathcal{V}_2$.*

Proof. The support of ϕ is a set of elements in $\mathcal{T}_i$. By the definition of the supermesh, for each element K in the support of ϕ, it is possible to decompose K into a set of elements of $\mathcal{T}_S$. Since the basis functions support the refinement property described in §5.1, ϕ can be exactly represented on this decomposition. Outside the support, ϕ is zero, which can also be exactly represented. Therefore, $\phi \in \mathcal{V}_S$ and the result follows. □

Since the supermesh supplies a superspace of the function spaces of the two input meshes, there is no interpolation error in interpolating functions from either of these meshes onto the supermesh. The supermesh is therefore the natural arena for the exact computation of the model output (equation (5.1)).

The extension to the case where the polynomial order varies over elements, such as in *p*- or *hp*-adaptivity, is briefly discussed. For an element K of an input mesh, let $p(K)$ be the the lowest row of Pascal's triangle which contains a term used in the basis functions (Zienkiewicz and Taylor, 2000a, fig. 8.5). For example, bilinear shape functions on quadrilateral elements have $p(K) = 2$, due to the bilinear xy term, while linear shape functions on simplicial elements have $p(K) = 1$. For each simplicial element $K_S \in \mathcal{T}_S$, its parent in $\mathcal{T}_i$ is denoted as $P_i(K_S)$. K_S is equipped with Lagrange basis functions of order $\max(p(P_1(K_S), p(P_2(K_S))))$. This choice guarantees that the span of the basis functions of the supermesh element contains the span of the parent basis functions confined to that supermesh element. If either of the parent elements are equipped with discontinuous basis functions, then K_S should also be equipped with discontinuous basis functions. With this choice of $\mathcal{V}_S$, it is easy to see that an analogous result to lemma 5.1 holds.

5.3 Forming a common mesh for interpolation

The approach detailed in §5.2 forms an exact superspace, but requires a significant amount of computational effort. Additionally, if fields on many meshes are to be manipulated (e.g., for time averaging), all those meshes must be supermeshed, which would be prohibitively expensive. An alternative approximate approach is developed for the case where exactness is unnecessary, but no obvious choice for a common mesh exists.

Given a sequence of meshes $\mathcal{T}_i, i = 1, \ldots, n$, the goal is to compute a mesh

$\mathcal{T}_C$ which will well-represent functions from $\mathcal{V}_i$ for all i. It is required that this mesh should have fine resolution wherever any of its input meshes have fine resolution, so that the fine detail will not be lost upon interpolation to $\mathcal{V}_C$. That is, for any point $x \in \Omega$, for a given direction $\vec{v}$, it is required that the edge length in $\mathcal{T}_C$ be less than or equal to the minimum of the edge lengths of $\{\mathcal{T}_i\}$ at x in the direction $\vec{v}$. To motivate this criterion, suppose that on each mesh $\mathcal{T}_i$ there is a field ψ_i which is to be used in some computation. With basis functions of degree p, the error in the computation of ψ_i (measured in a suitable norm) is $O(h_i^{p+1})$, where h_i is the mesh length scale of $\mathcal{T}_i$. By ensuring that at each point in the domain,

$$h_C \leq \min_i(h_i), \tag{5.2}$$

the aim is to ensure that the interpolation error introduced is no worse than $O(h_i^{p+1})$; that is, that the error involved in interpolating onto $\mathcal{T}_C$ is of the same order as the error in the input fields.

The idea of the following algorithm is to intersect the edge length requirements of an analytical representation of each mesh in $\{\mathcal{T}_i\}$ and to supply this intersected mesh specification to an adaptive remeshing library. As an interpolation error has been introduced in order to transfer the tensor fields encoding the mesh edge lengths onto a common mesh so that they may be intersected, this procedure is iterated until convergence. The output mesh will then satisfy the edge lengths criterion described above. Obviously, if the nodes and edges do not align then there will be error in the interpolation onto the common mesh; if this is unacceptable, then the algorithm described in §5.2 should be used.

For a given mesh $\mathcal{T}_i$, it is possible to encode the edge lengths and element orientations as a piecewise-constant symmetric positive-definite metric tensor field $\mathcal{M}_i$ on $\mathcal{T}_i$ (figure 5.1; Vallet (1990); George and Borouchaki (1998)). This analytical representation of the mesh is well known and is widely used to facilitate the automated manipulation of meshes. For example, adaptive remeshing libraries typically take in a piecewise-linear metric tensor field to guide the adaptive procedure (Pain et al., 2001; Agouzal et al., 1999; George and Borouchaki, 1998). The metric tensor for a given element can be computed by the polar decomposition of the Jacobian of the affine transformation mapping the ideal element $\widehat{K}$ to the physical element K (Formaggia

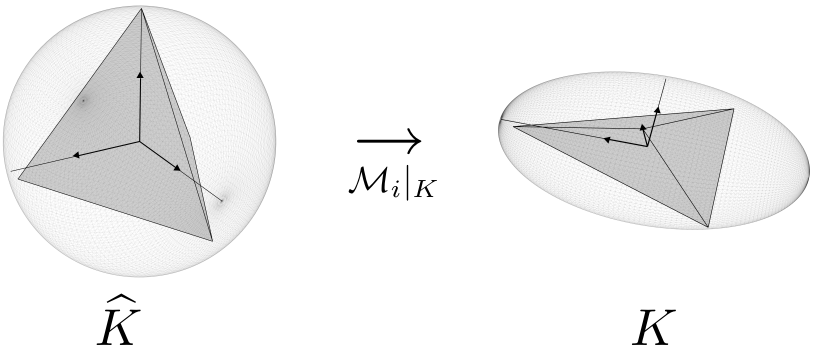


Figure 5.1: The geometric properties of each element in a simplicial mesh can be represented as piecewise-constant symmetric positive-definite metric. The metric is computed from the polar decomposition of the Jacobian of the affine transformation mapping the ideal element $\widehat{K}$ to the physical element K.

and Perotto, 2001; Micheletti and Perotto, 2006).

For a given symmetric positive-definite metric tensor H, consider the unit ball under the inner product induced by this tensor, that is:

$$\mathcal{B}_H = \left\{ x \in \mathbb{R}^d \mid \sqrt{x^T H x} = 1 \right\}. \tag{5.3}$$

In two dimensions, $\mathcal{B}_H$ forms an ellipse; in three dimensions, an ellipsoid. The orientation of the ellipsoid represents the eigenvectors, while the eigenvalues encode the size of the ellipsoid along those directions. These observations motivate an algorithm for combining the edge length requirements of two tensors H_1 and H_2 (Castro-Díaz et al., 1997). The algorithm computes the intersection of two metric tensors by computing an approximation to the ellipsoid of maximal measure contained within both (figure 5.2). The output tensor therefore satisfies the edge length requirements of both its inputs in every direction. For a description of the algorithm to compute the intersection, see Castro-Díaz et al. (1997); Borouchaki et al. (1997a).

The algorithm to construct $\mathcal{T}_C$ proceeds as follows.

Algorithm 5.1. *Pseudo-supermesh construction.*

$k \leftarrow 1$.

Choose $\mathcal{T}_C^k = \mathcal{T}_1$.

For each mesh $\mathcal{T}_i$*, compute the metric representing it,* $\mathcal{M}_i$.

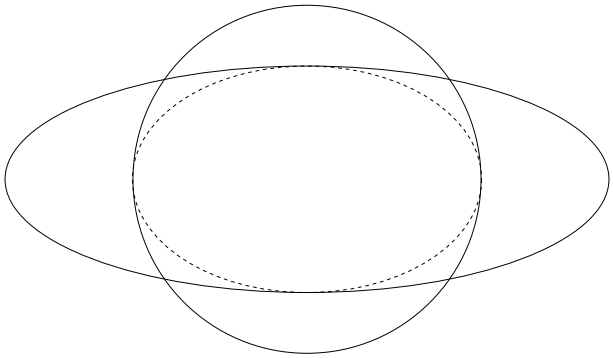


Figure 5.2: Given two metrics, the metric intersection procedure of Castro-Díaz et al. (1997) combines their edge length requirements by computing an approximation to the contained ellipse (ellipsoid in three dimensions) of maximal measure.

Until the adaptive procedure converges:

1. *For each mesh, interpolate* $\mathcal{M}_i$ *onto* $\mathcal{T}_C^k$ *and intersect their requirements to give* $\mathcal{M}_C^{k+1}$.

2. *Supply* $\mathcal{M}_C^{k+1}$ *to the adaptive remeshing library to form* $\mathcal{T}_C^{k+1}$.

3. $k \leftarrow k+1$.

The procedure terminates when the mesh $\mathcal{T}_C^k$ satisfies the mesh sizing requirements encoded in $\mathcal{M}_C^{k+1}$; this is determined by the functional used in the adaptive procedure itself (Pain et al., 2001; Agouzal et al., 1999). Typically, this procedure converges in 3 to 5 iterations, although it can take more if the initial mesh $\mathcal{T}_C^1$ is unsuitable.

5.4 Examples

5.4.1 Supermeshing

5.4.1.1 Interpolation error quantification

This technique has already been used earlier in this thesis to quantify the suboptimality of collocation interpolation in §2.3.3.4.

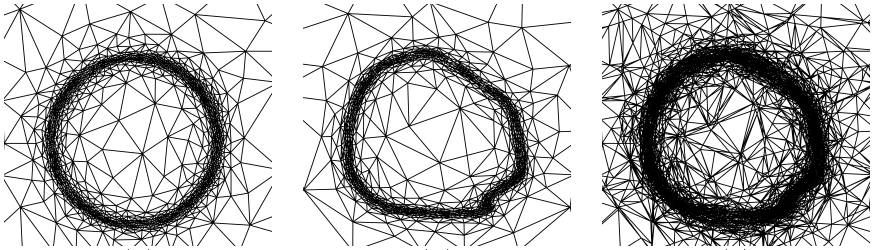


Figure 5.3: (a) Mesh adapted to initial conditions. (b) Mesh adapted to final solution. (c) A supermesh of the initial and final meshes.

5.4.1.2 Difference from an analytical solution

In the advection test case of Rudman (1997), a material volume fraction is advected with a prescribed velocity for N timesteps before reversing the flow and advecting further for N additional timesteps. If there were no discretisation errors, the final state of the volume fraction would be equal to the initial condition. Therefore, the discretisation error introduced may be quantified by the difference between the final and initial conditions. However, as the adaptive remeshing algorithm described in Vasilevskii and Lipnikov (1999) is applied, the initial and final meshes are different. Therefore, in order to compute the difference and thus the discretisation error exactly, the supermesh of the initial and final meshes is constructed (figure 5.3). The exact computation of the discretisation error in this case would be impossible without such a construction. Thus, it is possible to rigorously compare the accuracy and efficiency of different advection algorithms and adaptive remeshing methods.

5.4.1.3 Vertical integration

The vertical integration of a quantity is a diagnostic commonly used in analysing flow fields in geophysical applications. It is key to the determination of advective transports, as a proxy for measurements of upwelling or downwelling, and for computing depth integrated stream functions. Vertical integration on a mesh that is structured in the z-direction is trivial, requiring no interpolation. Vertical integration over a general mesh with no prescribed columnar structure, such as used by the vertically unstructured model ICOM described in Pain et al. (2005b) and Piggott et al. (2008),

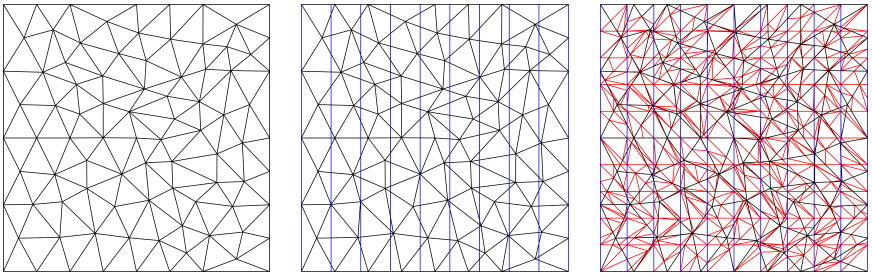


Figure 5.4: (a) A two-dimensional donor mesh containing fields to be vertically integrated. (b) The extrusion of nodes from a one-dimensional target mesh through the 2D domain (in blue). (c) The resulting supermesh (all of red, blue and black) via which the vertical integration of fields on the donor mesh can be performed.

requires a projection.

To compute the vertically-integrated function for a general unstructured mesh, the projection of the vertically-integrated function onto the function space associated with the surface mesh is computed. This is performed by generating a supermesh between the original unstructured mesh, and an appropriate vertically structured mesh, and using this to perform a Galerkin projection of the vertical integral onto the surface mesh. Let $\mathcal{T}_H$ be a $(d-1)$-dimensional horizontal surface mesh of $\delta\Omega \subset \mathbb{R}^{(d-1)}$. Let $\mathcal{T}_D$ be a (possibly unrelated) d-dimensional volume mesh of Ω with associated function space $\mathcal{V}_D$. Let Π be the extrusion operator that extrudes the surface mesh such that $\Pi(\mathcal{T}_H) \supseteq \mathcal{T}_D$. Given a function $v_D \in \mathcal{V}$, a vertically integrated field $v_H \in \mathcal{V}_H$ can be defined such that

$$\int_{\partial\Omega} v_H \phi_H^{(i)} \, dA = \int_{\partial\Omega} \left[\int_z v_D \, dz \right] \phi_H^{(i)} \, dA, \tag{5.4}$$

$$= \int_{\partial\Omega} \int_z \left[v_D \Pi(\phi_H^{(i)}) \right] \, dz dA \tag{5.5}$$

$$= \int_\Omega v_D \Pi(\phi_H^{(i)}) \, dV \tag{5.6}$$

for all surface basis functions $\phi_H^{(i)}$, with $\Pi(\phi_H^{(i)})$ defined as the vertical extrusion of $\phi_H^{(i)}$ which is constant over the vertical. Replacing v_D and v_H by

their basis function expansions, the equations become

$$M_H v_H = M_{HD} v_D, \tag{5.7}$$

where

$$(M_H)_{ij} = \int_{\partial\Omega} \phi_H^{(i)} \phi_H^{(j)} dA, \tag{5.8}$$

and

$$(M_{HD})_{ij} = \int_{\Omega} \Pi(\phi_H^{(i)}) \phi_D^{(j)} dV. \tag{5.9}$$

The assembly of M_{HD} can be computed using the supermesh of $\Pi(\mathcal{T}_H)$ and $\mathcal{T}_D$, as for each element in the supermesh the basis functions of $\mathcal{V}_H$ and $\mathcal{V}_D$ are polynomials.

Note that since the constant function $1 \in \mathcal{V}_H$, the projection of the vertically-integrated function preserves its integral.

Obviously, the accuracy of the representation of the vertically-integrated function depends on the choice of the surface mesh $\mathcal{T}_H$. One choice is to use the surface mesh associated with $\mathcal{T}_D$. A more accurate but also more expensive choice would be to project all the elements of $\mathcal{T}_D$ onto $\partial\Omega$ and then use the supermesh (or perhaps the pseudo-supermesh) of these to form the surface mesh onto which to project.

The vertical integration diagnostic was applied to a simulation of the rotating thermally driven annulus (Hide and Mason, 1975). The simulation was conducted using ICOM (Piggott et al., 2008) with the adaptive remeshing algorithm of Pain et al. (2001). The configuration is as described in Wordsworth et al. (2008), with differential heating of the tank inner and outer side-walls at temperatures T_A and T_B respectively ($T_B - T_A > 0$), and with 22° sloping top and bottom boundaries (shallow at annulus centre). Other system parameters are as given in Wordsworth et al. (2008) table I (fluid 1) at rotation rate $\Omega = 1.3$ rad / s. For these parameters the dynamics exhibit a chaotic irregular flow pattern (Wordsworth et al., 2008). The vertical integrals of the horizontal components of velocity were computed as per equation (5.7), and used to compute a vertically integrated stream function, shown in figure 5.4.1.3. Note how, for this system configuration, two distinct trains of eddies are observed, with cyclonic eddies towards the mid-radius and highly unstable anti-cyclonic eddies towards the tank inner wall. These two trains are clearly observed in the vertically integrated stream

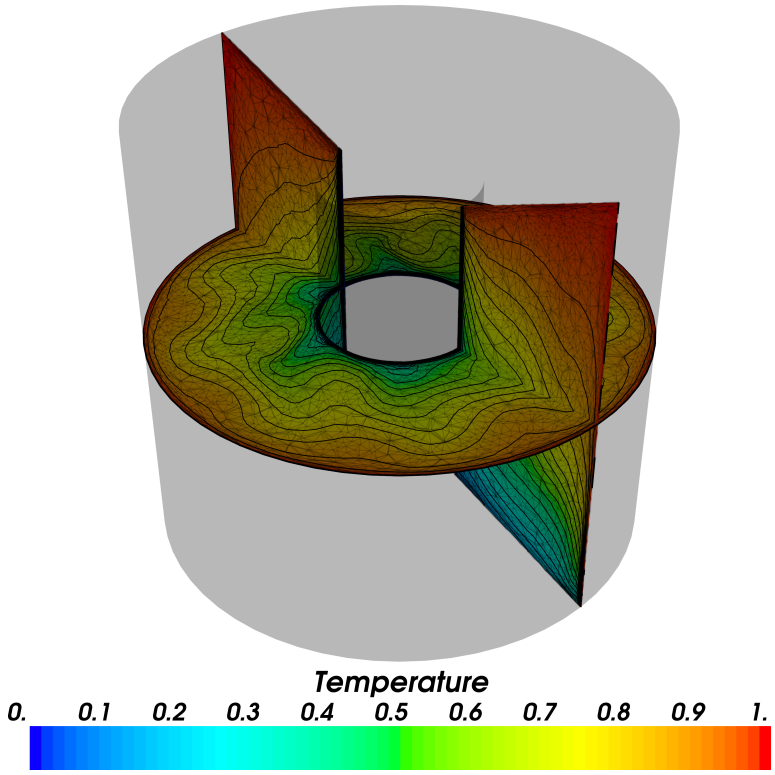


Figure 5.5: Simulation of the thermally driven annulus with sloping lower and upper boundaries, in an irregular flow regime. Normalised temperature $(T - T_A)/(T_B - T_A)$ shown, with 20 contours.

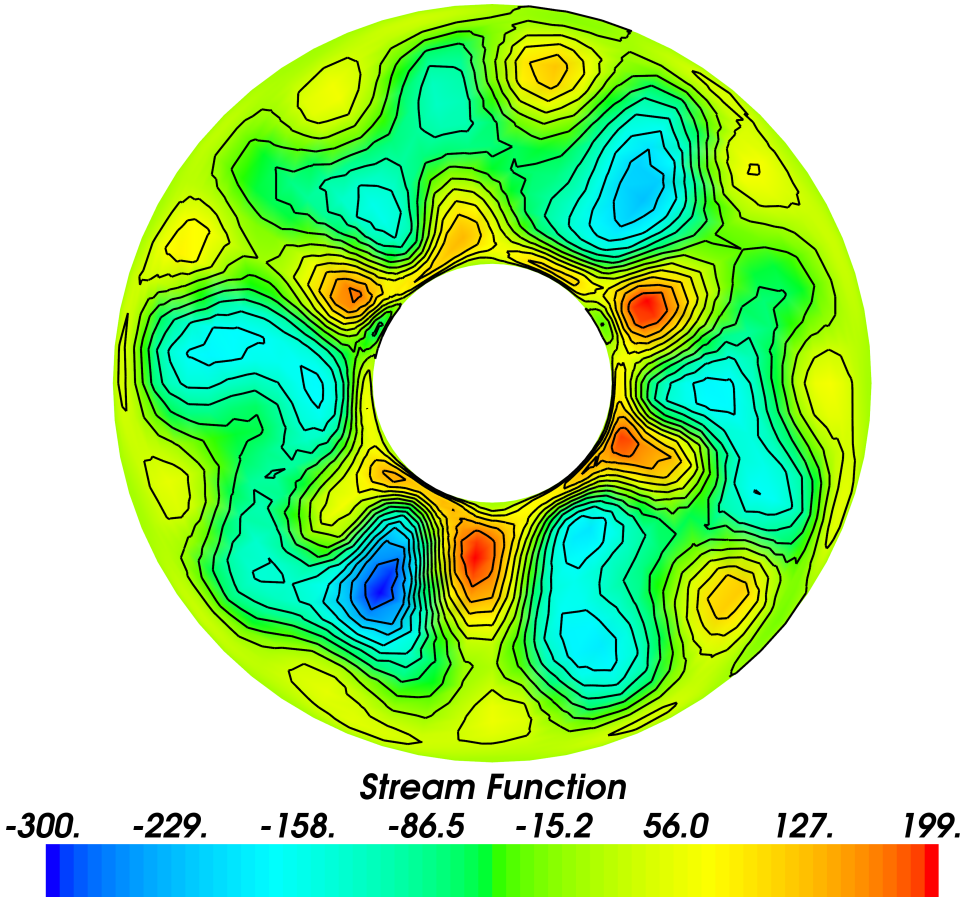


Figure 5.6: Vertically integrated stream function for an ICOM simulation of the thermally driven annulus with sloping lower and upper boundaries, in an irregular flow regime. Two trains of eddies can be seen: cyclonic eddies at the mid-radius (negative stream function) and highly unstable anti-cyclonic eddies towards the inner wall (positive stream function). Stream function units are cm^2 / s, with 16 contours shown.

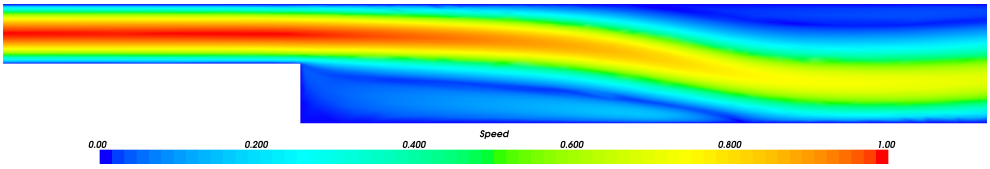


Figure 5.7: A typical backward-facing step simulation. An inflow boundary condition is specified on the left-hand side, an outflow boundary condition on the right-hand side, and no-slip imposed elsewhere. The flow separates at the step and reattaches downstream.

function diagnostic. Without the use of the supermesh, rigorously computing this vertically integrated stream function on a vertically-unstructured mesh would be impossible.

5.4.2 Pseudo-supermeshing

5.4.2.1 Time averaging

In fluid dynamical simulations, time averaging is necessary to compute a decomposition of the velocity into a mean and a fluctuating component, which are key diagnostic quantities. Since a simulation may perform a large number of remeshings, it is computationally expensive to perform an intersection of all simulation meshes. It is therefore appropriate to instead make use of an approximate pseudo-supermesh, in order to minimise the interpolation error introduced when time averaging.

Pseudo-supermeshing for time averaging was applied to an adaptive mesh simulation of the backward-facing step. The backward-facing step (figure 5.7) is a popular problem for investigating the simulation of the separation and reattachment of turbulent flows, as accurate experimental results for a wide range of flow regimes exist (Armaly et al., 1983). For a review of the use of the backward-facing step case for comparative studies of different strategies for numerical simulation, see Candy (2008, §8.7.2). One of the most important diagnostics of the backward-facing step problem is the reattachment length of the time-averaged flow; therefore, to compute this diagnostic, the time-averaged velocity must be computed.

A simulation of the three-dimensional backward-facing step at Reynolds number $Re = 10^3$ (using the definition of Armaly et al. (1983)) was performed on 64 processors. The stabilised P1-P1 element pair was used to

discretise velocity and pressure. The Crank-Nicolson timestepping scheme was used. The timestep was automatically adapted to maintain a maximum CFL number of 2. The mesh was adapted every 20 timesteps, and contained approximately 5 million nodes. The metric formulation of Pain et al. (2001) was used to control the L_∞ norm of the interpolation error of velocity.

The simulation was initialised from rest. The simulation was spun up for 70 time units. From $t_0 = 70$ to $t_1 = 102$, the computed velocity field was recorded every 0.5 time units. These velocity snapshots were then used to compute the time-averaged velocity field for the purposes of computing the reattachment length. The length of the domain in the x-direction is 30 units, and the inflow velocity is of order 1; therefore, this time period is approximately one flushing cycle, two complete cycles after initialisation.

A pseudo-supermesh was constructed from the snapshot meshes and the velocity fields were interpolated onto this mesh using collocation interpolation. This interpolation is complicated by the fact that each snapshot of velocity and the pseudo-supermesh is defined on a different mesh with a different parallel domain decomposition. The naïve approach of interpolating from each subdomain to each subdomain is quadratic in the number of subdomains; to circumvent this, a bounding-box intersection predicate was employed to filter the number of source subdomains to be considered. With the velocity fields interpolated onto the pseudo-supermesh, the time-averaged velocity $\bar{u}$ was then calculated:

$$\bar{u}(\vec{x}) = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} u(\vec{x}, t)\ dt, \tag{5.10}$$

where $[t_0, t_1] = [70, 102]$ (figure 5.8). This integral was evaluated by Gaussian quadrature. Note that such an evaluation would be exceedingly difficult if the available velocity fields were not on the same mesh.

The reattachment length was defined to be the length from the step at which the zero-isosurface of the x-component of $\bar{u}$ intersects with the bottom boundary. This quantity was computed from $\bar{u}$ using the VTK library (Schroeder et al., 2006). For the simulation described, the reattachment length was approximately 10 times the step height, which is consistent with the value given in the literature (Le et al., 1997). A more rigorous analysis would involve repeating the experiment for a range of Reynolds numbers and comparing the reattachment length of multiple model runs. It may be

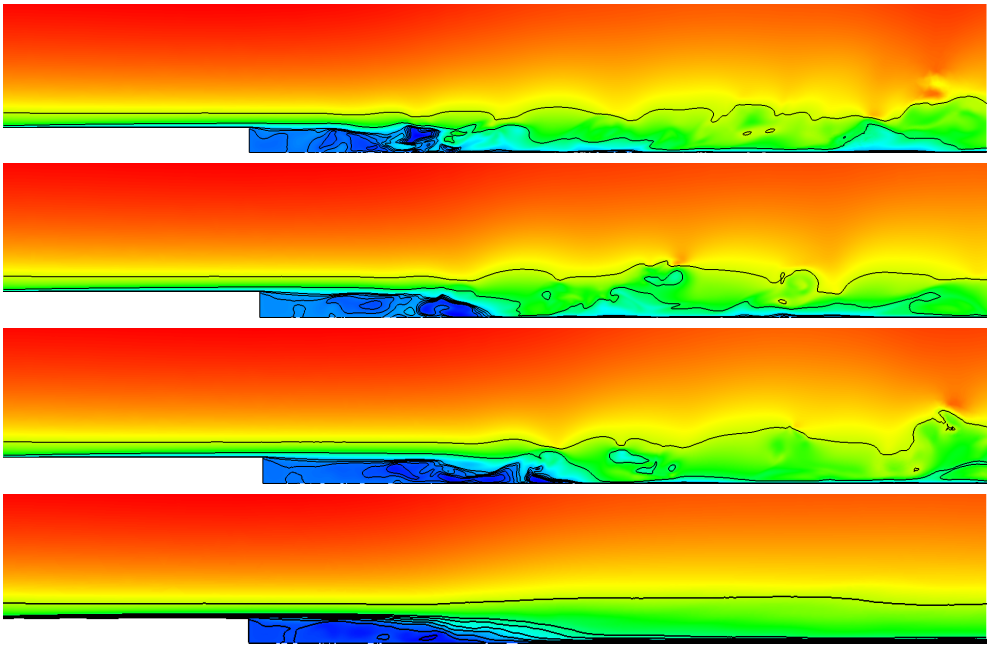


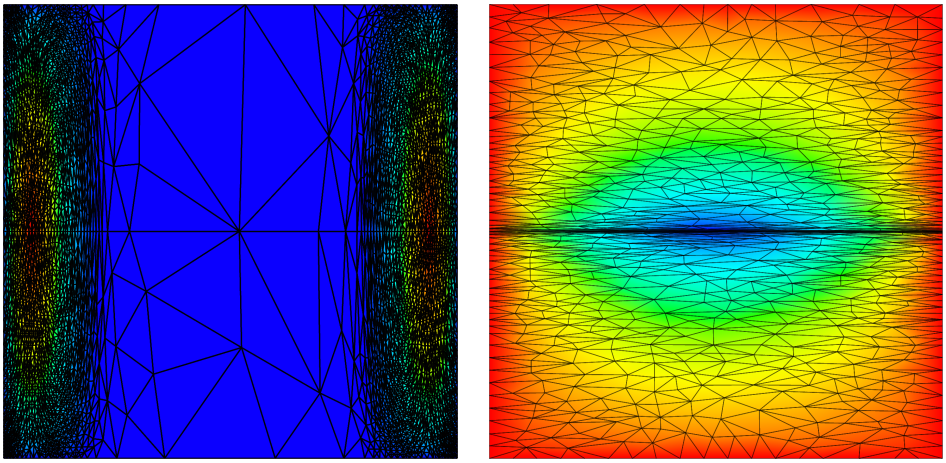
Figure 5.8: A view along the plane $y = 2$ of the x-component of velocity at times $t = 70$, $t = 86$, $t = 102$, and the x-component of the time-averaged velocity $\bar{u}$.

seen that pseudo-supermeshing is an efficient method to produce a common mesh suitable for the interpolation of fields from multiple meshes, such as is necessary in time-averaging.

5.4.2.2 Adjoint computations on different meshes

Adjoint equations arise naturally in the control and optimisation of physical systems (Gunzburger, 2003). An adjoint equation is a linear differential equation which is associated with a (possibly nonlinear) forward equation. For a given physical system, its associated adjoint equation may be solved to efficiently calculate gradient values with respect to a particular functional (Giles and Pierce, 2000). The adjoint equation is also fundamental to the theory of goal-based error estimation (Becker and Rannacher, 2001; Bangerth and Rannacher, 2003).

There are two approaches to compute the adjoint of a physical system (Gunzburger, 2003, §2.9). The first is to discretise the forward equations, then differentiate these discrete forward equations to form the adjoint equa-



(a) (b)

Figure 5.9: Forward and adjoint solutions of the problem described in §5.4.2.2.

tions to be solved. This differentiation may be performed with the aid of an automatic differentiation tool (Griewank, 2000), or with the Independent Set Perturbation method (Fang et al., 2009a). The second is to differentiate the continuous forward equations to yield continuous adjoint equations, and then to discretise these separately. One advantage of the differentiate-then-discretise approach is that it allows for the use of different meshes for the adjoint problem; "it is clear that adjoint or sensitivity systems are often best discretized using a different grid than that used for the flow" (Gunzburger, 2003, page 60).

Fang et al. (2006) implements such an approach where the forward and adjoint equations are solved separately on different sequences of adapting meshes. This approach necessitates the interpolation of the forward and adjoint solutions onto a common mesh. The author comments that control of the interpolation error by choosing a suitable common mesh is crucial for the success of the technique. This strategy of solving the forward and adjoint equations on separate meshes is also advocated elsewhere in the literature (e.g., Korotov (2007); Rüter et al. (2007)).

To demonstrate the application of pseudo-supermeshing to such problems, the example of Richter (2001, §6.3) is considered. The Poisson problem with

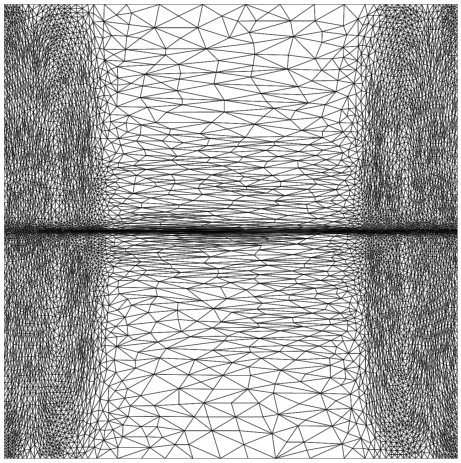


Figure 5.10: The pseudo-supermesh of the two meshes shown in figure 5.9.

homogeneous Dirichlet boundary conditions,

$$-\nabla^2 u = f, \tag{5.11}$$

is solved on $\Omega = (-1, 1)^2$. The source term f is chosen such that

$$u = \left(1 - x^2\right)\left(1 - y^2\right)\exp\left(-x^{-4}\right), \tag{5.12}$$

which satisfies the boundary conditions $u|_{\partial\Omega} = 0$. For the forward solution, the mesh was adapted with the algorithm described in Vasilevskii and Lipnikov (1999) to control the L_∞ norm of the interpolation error in representing u with a target error of 10^{-4}. Such an adapted mesh is shown in figure 5.9a. As the adaptive algorithm is not goal-based and does not incorporate information from the adjoint solution, the mesh is merely adapted to optimise the representation of the function u.

The corresponding adjoint equation is solved for the adjoint solution associated with the functional

$$J(u) = \int_{-1}^{1} u(x, 0) \quad dx. \tag{5.13}$$

As the equation is self-adjoint, the solution procedure merely consists of replacing the right-hand side of the discretised forward equation with the functional evaluated for each basis function. The anisotropy of the adjoint

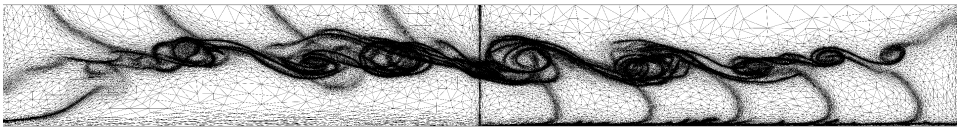


Figure 5.11: The pseudo-supermesh of the lock exchange snapshots. Notice the resolution placed at the location of the density interface for each snapshot.

solution is entirely different to that of the forward solution, and therefore the adjoint solution requires a different mesh for its efficient representation. Again, the adaptive procedure of Vasilevskii and Lipnikov (1999) was applied to adapt the mesh to the interpolation error of the adjoint solution; the target interpolation error was set to 2×10^{-4}. Such an adapted mesh is shown in figure 5.9b.

Figure 5.10 shows the pseudo-supermesh of the meshes individually adapted to the forward and adjoint solutions. The length scale of the mesh is everywhere the minimum of the length scales of the input meshes. This mesh is therefore suitable for the common interpolation of the forward and adjoint solutions for computations depending on both.

5.4.2.3 Proper Orthogonal Decomposition

Regardless of the computational resources available, there will always be models of physical systems which are too complex to simulate. One approach to circumvent this difficulty is to systematically reduce the complexity of the model while retaining its key features. This process is referred to as model order reduction. The Proper Orthogonal Decomposition (POD) is a very popular tool for achieving model reduction (Schilders et al., 2008). POD involves computing a small set of basis functions which captures the key dynamics of the system.

It has previously been observed that it is necessary to interpolate the solution fields to a common mesh for the purposes of computing a POD basis if adaptive remeshing is used (Fang et al., 2008, 2009b). Since constructing the POD basis involves the singular value decomposition of the data matrix, the snapshots comprising the data matrix must lie inside the same function

space.

A brief description of the proper orthogonal decomposition is given. Let $\psi_i \in \mathcal{V}_C$ be snapshots of a field, $i = 1, \ldots, m$. Let n be the number of basis functions associated with $\mathcal{V}_C$. The mean $\bar{\psi}$ is computed as

$$\bar{\psi} = \frac{1}{m} \sum_{i=1}^{m} \psi_i, \tag{5.14}$$

and the deviation from the mean is computed as

$$\tilde{\psi}_i = \psi_i - \bar{\psi}, \tag{5.15}$$

for each snapshot i. Let A be the $n \times m$ data matrix formed from these vectors as

$$A = \left[\begin{array}{c|c|c} & & \\ \tilde{\psi}_1 & \tilde{\psi}_2 & \ldots \\ & & \end{array} \right]. \tag{5.16}$$

The POD basis functions are computed as the eigenvectors of the $m \times m$ matrix $A^T A$. For each eigenvector, the corresponding eigenvalue is referred to as its energy. For more details, see Schilders et al. (2008).

To demonstrate the utility of pseudo-supermeshing for adaptive POD, 7 snapshots of the density field of a two-dimensional lock exchange simulation were taken from a simulation similar to a simulation described in Härtel et al. (2000) (figure 5.12). Using the definition of the Reynolds number given in Härtel et al. (2000), the Reynolds number is approximately 7 $\times 10^2$. As the simulation adapts to resolve the density interface using the algorithm described in Vasilevskii and Lipnikov (1999), the meshes for the snapshots are entirely unrelated to each other. Therefore, in order to provide a common mesh upon which the decomposition may be computed, the pseudo-supermesh of the meshes of the snapshots is assembled using the algorithm of §5.3. The pseudo-supermesh is visible in figure 5.11.

The density fields are then interpolated onto the pseudo-supermesh. For simplicity, only the POD basis corresponding to density was formed. The computed POD basis functions are given in order of descending energy in figure 5.12. These POD basis functions could then be used to form a reduced-order model of the system.

As can be seen, the POD basis functions exhibit fine detail in the eddy re-

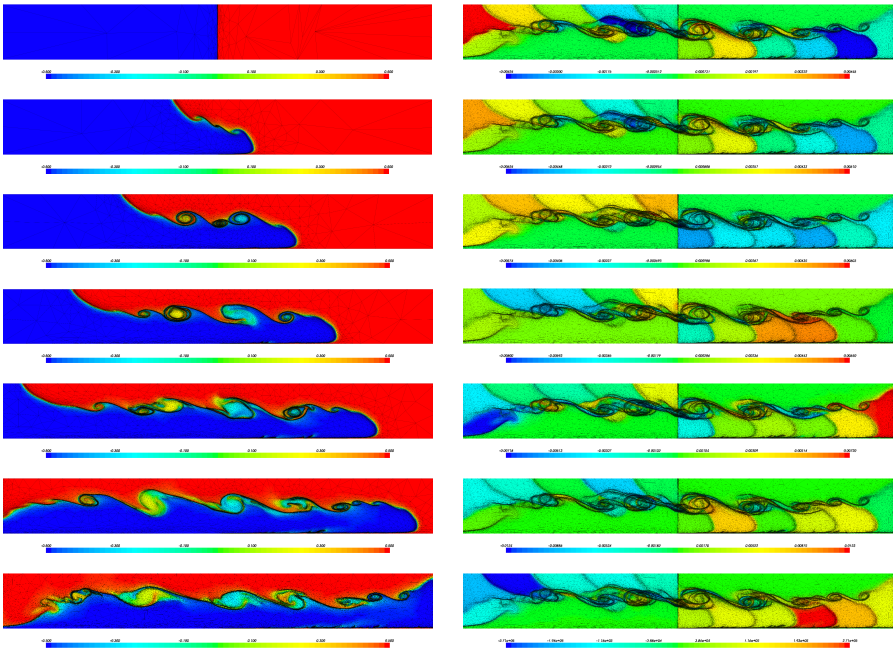


Figure 5.12: (a) Snapshots of a two-dimensional lock exchange problem. (b) Computed POD basis functions on the pseudo-supermesh, in order of decreasing energy.

gions contained in the snapshots. Therefore, the mesh upon which the POD basis is computed must be sufficiently fine to resolve them. The pseudo-supermesh satisfies this criterion in an efficient manner, i.e. achieving the resolution with a uniform mesh would be prohibitively expensive.

5.5 Conclusions

Two techniques for the computation of diagnostics of adapting simulations have been presented. The first exploits the fact that the supermesh induces a function space that is a common superspace of the function spaces of its input meshes. The second builds a pseudo-supermesh for diagnostics where exactness is unnecessary. The utility of both techniques has been demonstrated by several examples for each.

Chapter 6

Epilogue

6.1 Summary of presented work

This thesis has discussed the analysis, implementation and applications of Galerkin projection and supermesh construction.

In chapter 1, a thorough review of the history, development and current frontiers of adaptive remeshing was given. This chapter explained the context in which my study of mesh-to-mesh interpolation came about, and motivated the developments of the subsequent chapters.

In chapter 2, the history of attempts to develop conservative interpolation operators between restricted classes of meshes was given. Assuming the availability of supermesh construction, the Galerkin projection was described and analysed, along with a bounded variant for piecewise-linear fields.

Having demonstrated the utility of supermeshing, chapter 3 focussed on algorithms for its construction. The development of local supermeshing allowed for the practical implementation of the projections described in the previous chapter. Several examples were given to illustrate the possibility of combining techniques such as adaptive remeshing and discontinuous Galerkin methods which were previously impossible.

Chapter 4 discussed the element-element association problem for arbitrarily different meshes of the same domain. By carefully exploiting the available information about the connectivity of the meshes, an algorithm was developed which is more than an order of magnitude faster than an alternative algorithm which does not exploit connectivity.

In chapter 5, other applications of supermeshing were investigated. It was found that supermesh construction is very useful for the computation

of diagnostics in simulations where adaptive remeshing is applied, as it allows for the construction of a common function superspace. An additional technique was proposed for the case where the exactness of supermeshing is unnecessarily expensive. The utility of these approaches was demonstrated on several examples of practical interest.

The algorithms developed in this thesis have been implemented in parallel and run on up to 1024 processors on HECToR, the UK National Supercomputing Service.

6.2 Possible applications

The algorithms presented in this thesis are potentially applicable to a wide variety of problems. In particular, the properties of the Galerkin projection make possible the application of adaptive remeshing to situations where it was previously infeasible. As demonstrated in §3.6, Galerkin projection allows for adaptive remeshing to be used in simulations with discontinuous Galerkin discretisations. Galerkin projection also allows for the use of adaptive remeshing in simulations where conservation of key quantities is a non-negotiable requirement for the discretisation. Examples of such systems include long-term climate prediction, nuclear criticality simulations, and multimaterial and multiphase flow modelling.

Model coupling is another possible area of application. As different models can impose different and incompatible restrictions on the meshes to be used, model coupling often involves the repeated transfer of data back and forth between different meshes. Here, the properties of the Galerkin projection make it ideal for this task. In particular, if data is to be repeatedly transferred back and forth between the same two meshes, it is possible to cache the sparse mixed mass matrix for efficiency.

For similar reasons, Galerkin projection may also be employed to enable a single model to use different meshes for different fields. As observed in §5.4.2.2, it is sometimes desirable to discretise the forward and adjoint equations on different meshes. This observation potentially carries through to other situations where a single model solves several separate equations; it may be advantageous to discretise the separate equations on separate meshes, and to communicate between them by supermeshing.

6.3 Future work

The cutoff point for the end of a thesis is usually arbitrary, and this is certainly the case here. While the core theory and implementation has been developed in this thesis, there are several possible extensions and improvements.

6.3.1 Curved boundaries

As discussed in §1.2.5.5, it is frequently desirable to change the approximation to the geometry when the mesh is adapted. This raises some interesting questions for the task of transferring data between these meshes. Consider the transfer of the constant function 1. If the transfer preserves constant functions, then it will not be conservative, and vice versa. Alauzet and Mehrenberger (2009) choose to preserve constant and linear functions at the expense of conservation.

The main reason for deferring this study to future work is that the adaptive remeshing libraries used in this thesis do not support CAD integration and so retain the initial geometry.

6.3.2 Property-preserving projections

While Galerkin projection preserves the integral of the field, it is often desirable to preserve other properties, especially if effort has been made in the discretisation to preserve them. Examples of such properties might include higher-order moments, solenoidality, irrotationality, or key physical balances such as geostrophic balance. A general way to incorporate such constraints is through the use of Lagrange multipliers (Carey et al., 2001). Future work could include the implementation and analysis of such constrained projections. In particular, the question of whether preserving such properties through the interpolation step qualitatively improves the solution remains unresolved.

6.3.3 Boundedness through optimisation

The algorithm presented in §2.3.4 relies on spreading overshoots and undershoots from node to node until they can be absorbed. Since this process is incremental and local, convergence can sometimes be slow. An alternative

approach is to cast the problem as a constrained optimisation problem:

$$J(q_T, \bar{q}_T) \to \min, \tag{6.1}$$

subject to

$$\int_\Omega q_T \, dV = \int_\Omega \bar{q}_T \, dV, \tag{6.2}$$

and

$$q_{\min} \leq \bar{q}_T \leq q_{\max}, \tag{6.3}$$

where q_T is the Galerkin projection, $\bar{q}_T$ is the bounded interpolant, $J(q_T, \bar{q}_T)$ is a cost functional to be minimised, and $q_{\min}$ and $q_{\max}$ are the bounds which $\bar{q}_T$ is to satisfy. The advantage of this approach is that it is possible to apply the rich theory of optimisation to the problem of computing a bounded interpolant. By doing so, this approach is not dependent on local incremental modifications to compute $\bar{q}_T$ from q_T, and could therefore be much faster.

Some experiments in this direction were performed using the Algencan optimisation library (Andreani et al., 2007, 2008). Let $\underline{q}_T$ and $\underline{\bar{q}}_T$ be the functions interpreted as vectors in $\mathbb{R}^n$, where n is the number of degrees of freedom of the function space. The cost functional was chosen to be

$$J(q_T, \bar{q}_T) = \frac{1}{2} \left|\left| \underline{q}_T - \underline{\bar{q}}_T \right|\right|_2^2, \tag{6.4}$$

the squared L_2 norm of the difference of the vectors. The conservation constraint was enforced by demanding that

$$M_T^L \cdot \underline{\bar{q}}_T = M_T^L \cdot \underline{q}_T, \tag{6.5}$$

where M_T^L is the row-summed lumped mass matrix of the target function space, interpreted as a vector in $\mathbb{R}^n$. This follows from lemma 2.2. With this choice of functional, equations (6.1-6.3) may be interpreted as a linear least-squares optimisation problem.

The initial results were promising, but not satisfactory. The procedure was applied to two- and three-dimensional water collapse simulations akin to that described in §3.6.5. In two dimensions, the procedure works well, and is significantly faster than the bounded interpolation scheme described in §2.3.4. However, as can be seen from figure 6.1, the optimisation proce-

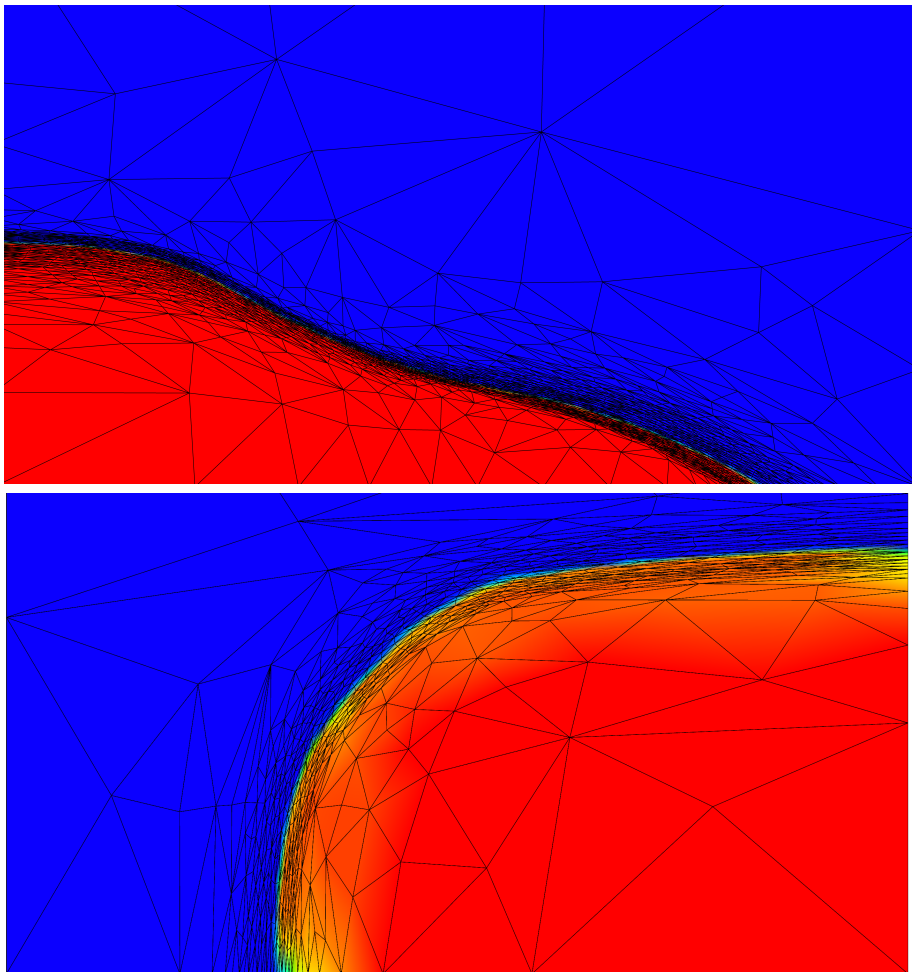


Figure 6.1: Initial experiments with boundedness through optimisation. Above: a two-dimensional water collapse simulation. Below: a three-dimensional water collapse simulation, as described in section §3.6.5, viewed from the bottom. While the volume fraction field is bounded and conservative, the optimisation approach causes spuriously low values inside the volume of water in the three-dimensional simulation. A better functional would penalise such behaviour.

dure causes spuriously low values inside the volume of the water column in the three-dimensional simulation. More research is necessary to investigate this behaviour and to decide upon a better cost functional with which to optimise.

6.3.4 Adaptive interpolation

As described in §1.1, there has been a broad trend in scientific computing towards incorporating adaptive error control loops into numerical algorithms. The most challenging aspect of designing an adaptive loop is usually the estimation of the relevant error. However, as discussed in §2.3.3.2, the structure of the discrete Galerkin projection lends itself naturally to a rigorous quantification of the interpolation error introduced. This could be exploited for an adaptive interpolation algorithm which modifies the target mesh until the interpolation error introduced in the projection is less than some user-specified tolerance.

6.3.5 Supermesh assembly

Interpolation is usually employed to transfer data from a donor function space to a target function space so that it may be used in computations in the target function space. However, is this actually necessary? Supermeshing may provide an alternative viewpoint, one which obviates the need for interpolation altogether.

Consider as an illustrative example the Poisson equation

$$-\nabla^2 u = f_D \qquad \text{in } \Omega, \tag{6.6}$$

$$u = 0 \qquad \text{on } \partial\Omega, \tag{6.7}$$

where the source term f_D is the discrete solution of another equation solved with a different model. For the argument, the particular details of the differential operator on the left-hand side and boundary conditions are unimportant. Problems of this nature arise frequently in computational mechanics when coupling different computational models. For example, one might wish to couple an unstructured finite element ocean model with a structured finite difference sea ice model which is only capable of solving its equations on a regular mesh. Other examples include coupling a biological activity

model with an ocean model, coupling an atmospheric chemistry model with a meteorological model, and coupling an electrocardiological model of the heart with a structural model of its motion.

If the problem is discretised on a mesh $\mathcal{T}_T$ of Ω, it leads to the following variational problem: find $u \in \mathcal{V}_T$ such that for all $\phi_T \in \mathcal{V}_T$,

$$\int_\Omega \nabla u \cdot \nabla \phi_T \, dV - \int_{\partial\Omega} \hat{n} \cdot u \nabla \phi_T \, dA = \int_\Omega f_D \phi_T \, dV, \tag{6.8}$$

where f_D lies in a function space $\mathcal{V}_D$ associated with a different mesh $\mathcal{T}_D$ of Ω.

The difficulty to be overcome is in the computation of the right-hand side of equation (6.8). One approach to this problem is to compute the Galerkin projection $f_T \in \mathcal{V}_T$ and to use this instead of f_D in equation (6.8). Since its Galerkin projection f_T has the property that

$$\int_\Omega f_T \phi_T \, dV = \int_\Omega f_D \phi_T \, dV, \tag{6.9}$$

the right-hand side of equation (6.8) will be the same whether f_T or f_D is used. Therefore, in a useful sense, the interpolation is lossless.

The above procedure may be viewed as somewhat wasteful, however; the assembly and solve of the mass matrix in the Galerkin projection are unnecessary, as the right-hand side of the Galerkin projection system is precisely the desired right-hand side of equation (6.8). Therefore, there is no need to interpolate in any sense; the supermesh allows for the direct computation of the desired functionals of f_D, without the need for any intermediate representation in $\mathcal{V}_T$.

This second approach, of using the supermesh to compute the desired functionals of f_D, also extends to equations other than equation (6.8). Consider the Galerkin approximation of the advection of a passive tracer τ by a velocity field u_D. The equation is to be solved in $\mathcal{V}_T$, but the velocity field u_D lies in $\mathcal{V}_D$. The assembly routine must assemble the matrix

$$C_{ij}^K = \int_K \phi_T^{(i)} \left(u_D \cdot \nabla \phi_T^{(j)} \right) \, dV, \tag{6.10}$$

for each element $K \in \mathcal{T}_T$ (Donea and Huerta, 2003, equation 2.14). As above, one approach is to compute the Galerkin projection $u_T \in \mathcal{V}_T$ of u_D

using the supermesh $\mathcal{T}_S$ of $\mathcal{T}_T$ and $\mathcal{T}_D$, and use u_T instead of u_D in equation (6.10). However, information is lost through this procedure; the resulting C^K will not be exact due to interpolation error. The alternative approach is to compute equation (6.10) as the sum of integrals over the elements of $\mathcal{T}_S^K$: since the operator

$$\Pi_{SD} : \mathcal{V}_D \to \mathcal{V}_S \tag{6.11}$$

is the identity operation by lemma 5.1, the velocity field u_D may be represented exactly on $\mathcal{T}_S$ and so C^K may be computed without any interpolation error induced by an intermediate representation.

If the goal is to compute some functional of a function on a different mesh, rather than to compute some close representative of it, it is in general possible to do this by integrating the supermesh construction algorithm with the finite element assembly routine. Thus, with the same amount of geometric work as Galerkin projection (though more assembly), it is possible to perform finite element computations involving different meshes in a manner entirely free of interpolation error.

Clearly, such an approach is more expensive than either collocation interpolation and computing the Galerkin projection, as more work in assembly is required; however, it answers those critics of adaptive remeshing who would claim that the error introduced in interpolation renders other adaptive techniques preferable. It is perhaps ironic that a thesis which deals with interpolation between meshes ends on a note which suggests the possible circumvention of its necessity.

Acknowledgements

The research described in this thesis would have been impossible without supervisors and colleagues who are generous with their time, a family which is generous in its support, and a string of lucky coincidences.

I first met Gerard Gorman in the offices of the Irish Centre for High-End Computing; I was working there as a system administrator, managing the cluster. I had no idea that the friendship we struck up would lead to an invitation to study in London, and ultimately to this thesis.

Committing to a Ph.D. is a gamble: a much bigger gamble than most people realise at the time. Any number of things can go wrong: you may not like the topic, or financial constraints may hinder your focus, or you may not get on well with your supervisors. I have been blessed with the finest supervision for which I could ask. Under the triumvirate of Matthew Piggott, Gerard Gorman and Christopher Pain, I have never lacked for advice, support or funding. They have my lasting gratitude. In particular, Matthew Piggott was the ideal combination of counsellor, editor, librarian, travel agent and teacher.

To imply, however, that my supervision ended with my formal supervisors would be a gross injustice. I have been the lucky beneficiary of countless hours of discussion and explanation with my colleagues of the Applied Modelling and Computation Group. In particular, David Ham, Stephan Kramer, Colin Cotter and Cian Wilson deserve special mention for their patient instruction. David Ham is one of those rare men who by sheer force of willpower change the world around them; without his efforts to restructure the model, this work would have been greatly hindered. The efficient stewardship of Tim Bond greatly helped in keeping the show on the road. My podmates on the third floor shared the frustrations and joys of computational research, and provided much practical wisdom on dealing with the

model. A discussion with Fangxin Fang on interpolation for Proper Orthogonal Decomposition sparked my initial interest in Galerkin projections. If engineering does not work out for Johnny Bull, he can surely make a career as a chef; Hannah Hiester earned a special place in my affections by baking wheat-free cakes and brownies.

Much of this work was performed in collaboration with James Maddison of the University of Oxford. James is one of the most intelligent men I have yet met. Most of the good ideas of this thesis (or at least the better ones) arose in conversation with him. I earnestly hope our future collaborations will be as fruitful.

As an institution, Imperial College London has been a fantastic environment in which to study. I would like to thank the department of Earth Science & Engineering for the Janet Watson scholarship which funded my research, and for the magnanimous award of the Janet Watson memorial prize for excellence in postgraduate research. The support of the Imperial College High Performance Computing Service is gratefully acknowledged; Simon Burbidge and Matt Harvey are both casually brilliant at their jobs. Their management of the College's cluster is superb (and I speak as a former cluster administrator).

Due to the generosity of my supervisors, I had the opportunity to travel widely to scientific conferences, both in the UK and abroad, and meet other researchers. I am grateful to Konstantin Lipnikov for supplying the two-dimensional mesh optimisation library used in this work. Frédéric Alauzet asked some insightful questions which caused me to reflect more on floating-point robustness. Stefano Micheletti and Simona Perotto have provided useful guidance on the extension of their metric tensor formulation to three dimensions. I would like to thank the organisers of the 2009 CBMS conference on adaptive finite element methods for partial differential equations, Guido Kanschat and Wolfgang Bangerth, for funding me to attend their summer school; the study of adaptive finite element methods is one of the most exciting areas of scientific computing today, and to have it explained and expounded by Rolf Rannacher was a great privilege.

None of my work would have been possible without the love and support of my family and my girlfriend, Carol. The support of my family, both financial and emotional, was instrumental in my ability to focus on the task at hand. My parents supplied boundless encouragement in my efforts to finish this thesis. The last word of thanks goes to Carol – her kind soul furnished me with a world outside of work, and thereby kept me sane.

References

A

N. Acikgoz and C. L. Bottasso. Metric-driven mesh optimization using a local simulated annealing algorithm. *International Journal for Numerical Methods in Engineering*, 71(2):201–223, 2007. doi: 10.1002/nme.1904. 14

P. K. Agarwal and M. Sharir. Red-blue intersection detection algorithms, with applications to motion planning and collision detection. *SIAM Journal on Computing*, 19(2):297–321, 1990. doi: 10.1137/0219020. 96

P. K. Agarwal, M. de Berg, S. Har-Peled, M. H. Overmars, M. Sharir, and J. Vahrenhold. Reporting intersecting pairs of polytopes in two and three dimensions. In *Algorithms and Data Structures*, pages 122–134. Springer, 2001. doi: 10.1007/3-540-44634-6_12. 96

A. Agouzal, K. Lipnikov, and Y. Vasilevskii. Adaptive generation of quasi-optimal tetrahedral meshes. *East–West Journal of Numerical Mathematics*, 7(4):223–244, 1999. 14, 114, 116

M. T. Ainsworth and J. T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. Wiley, New York, 2000. ISBN 978-0-471-29411-5. url: `http://eu.wiley.com/WileyCDA/WileyTitle/productCd-047129411X.html`. 5

F. Alauzet. *Adaptation de maillage anisotrope en trois dimensions. Application aux simulations instationnaires en mécanique des fluides*. PhD thesis, Université Montpellier II, Montpellier, France, 2003. 23

F. Alauzet. High-order methods and mesh adaptation for Euler equations. *International Journal for Numerical Methods in Fluids*, 58:1069–1076, 2008. doi: 10.1002/fld.1739. 18, 44

F. Alauzet and M. Mehrenberger. P1-conservative solution interpolation on unstructured triangular meshes. Technical Report RR-6804, INRIA Rocquencourt, Rocquencourt, Le Chesnay, France, 2009. url: `http://hal.inria.fr/inria-00354509/en/`. 33, 44, 132

F. Alauzet, P. L. George, B. Mohammadi, P. J. Frey, and H. Borouchaki. Transient fixed point-based unstructured mesh adaptation. *International Journal for Numerical Methods in Fluids*, 43(6-7): 729–745, 2003. doi: 10.1002/fld.548. 20

F. Alauzet, X. Li, E. S. Seol, and M.S. Shephard. Parallel anisotropic 3D mesh adaptation by mesh modification. *Engineering with Computers*, 21(3):247–258, 2006a. doi: 10.1007/s00366-005-0009-3. 26, 27

F. Alauzet, A. Loseille, A. Dervieux, and P. Frey. Multi-dimensional continuous metric for mesh adaptation. In P. P. Pébay, editor, *Proceedings of the 15th International Meshing Roundtable*, pages 191–214, Birmingham, Alabama, 2006b. Springer. doi: 10.1007/978-3-540-34958-7_12. 18

F. Alauzet, S. Borel-Sandou, L. Daumas, A. Dervieux, Q. Dinh, S. Kleinveld, A. Loseille, Y. Mesri, and G. Rogé. Multi-model and multi-scale optimization strategies. Application to sonic boom reduction. *European Journal of Computational Mechanics*, 17(1-2):191–214, 2008. doi: 10.3166/remn.17.245-269. 18

R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. On augmented Lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization*, 18(4):1286–1309, 2007. doi: 10.1137/060654797. 133

R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. Augmented Lagrangian methods under the constant positive linear dependence constraint qualification. *Mathematical Programming*, 111 (1):5–32, 2008. doi: 10.1007/s10107-006-0077-1. 133

T. Apel. Interpolation of non-smooth functions on anisotropic finite element meshes. *Mathematical Modelling and Numerical Analysis*, 33 (6):1149–1185, 1999. doi: 10.1051/m2an:1999139. 17

T. Apel and M. Dobrowolski. Anisotropic interpolation with applications to the finite element method. *Computing*, 47(3-4):277–293, 1992. doi: 10.1007/BF02320197. 17

T. Apel, S. Grosman, P. K. Jimack, and A. Meyer. A new methodology for anisotropic mesh refinement based upon error gradients. *Applied Numerical Mathematics*, 50(3-4):329–341, 2004. doi: 10.1016/j.apnum.2004.01.006. 9

A. Arakawa and V. R. Lamb. A potential enstrophy and energy conserving scheme for the shallow water equations. *Monthly Weather Review*, 109(1):18–36, 1981. doi: 10.1175/1520-0493(1981)109<0018:APEAEC>2.0.CO;2. 40, 49

B. F. Armaly, F. Durst, J. C. F. Pereira, and B. Schönung. Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics*, 127:473–496, 1983. doi: 10.1017/S0022112083002839. 122

M. S. Aubé, W. G. Habashi, H. Wang, and D. Torok. On the impact of anisotropic mesh adaptation on computational wind engineering. *International Journal for Numerical Methods in Fluids*, 2009. doi: 10.1002/fld.2109. Accepted, in press. 15

B

I. Babuška and A. K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13(2):214–226, 1976. doi: 10.1137/0713021. 17

I. Babuška and W. C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM Journal on Numerical Analysis*, 15(4): 736–754, 1978a. doi: 10.1137/0715049. 5

I. Babuška and W. C. Rheinboldt. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in*

Engineering, 12(10):1597–1615, 1978b. doi: 10.1002/nme.1620121010. 5

I. Babuška and M. Suri. The p and h-p versions of the Finite Element method, basic principles and properties. *SIAM Review*, 36(4):578, 1994. doi: 10.1137/1036141. 5

R. H. Bailey. An algorithm for the conservative interpolation of data between two-dimensional structured or unstructured triangular meshes. Master's thesis, University College Swansea, 1987. 42, 43, 49, 60, 61

T. J. Baker. Mesh adaptation strategies for problems in fluid dynamics. *Finite Elements in Analysis and Design*, 25(3-4):243–273, 1997. doi: 10.1016/S0168-874X(96)00032-7. 11

S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997. 65

D. S. Balsara. Divergence-free adaptive mesh refinement for magnetohydrodynamics. *Journal of Computational Physics*, 174(2):614–648, 2001. doi: 10.1006/jcph.2001.6917. 31

W. Bangerth and R. Rannacher. Adaptive finite element techniques for the acoustic wave equation. *Journal of Computational Acoustics*, 9(2):575–591, 2001. doi: 10.1142/S0218396X01000668. 24

W. Bangerth and R. Rannacher. *Adaptive finite element methods for differential equations*. ETH Zürich Lectures in Mathematics. Birkhäuser, 2003. ISBN 3764370092. 5, 20, 124

R. E. Bank and J. Xu. Asymptotically exact a posteriori error estimators, Part II: general unstructured grids. *SIAM Journal on Numerical Analysis*, 41(6):2313–2332, 2003. doi: 10.1137/S0036142901398751. 24

Z. P. Bažant. Spurious reflection of elastic waves in nonuniform finite element grids. *Computer Methods in Applied Mechanics and Engineering*, 16(1):91–100, 1978. doi: 10.1016/0045-7825(78)90035-X. 24

R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10: 1–102, 2001. doi: 10.1017/S0962492901000010. 5, 124

J. Behrens. *Adaptive Atmospheric Modeling*, volume 54 of *Lecture Notes in Computational Science and Engineering*. Springer, 2006. 8

Y. Belhamadia. A time-dependent adaptive remeshing for electrical waves of the heart. *IEEE Transactions on Biomedical Engineering*, 55 (2):443–452, 2008. doi: 10.1109/TBME.2007.905415. 19

Y. Belhamadia, A. Fortin, and É. Chamberland. Anisotropic mesh adaptation for the solution of the Stefan problem. *Journal of Computational Physics*, 194(1):233–255, 2004. doi: 10.1016/j.jcp.2003.09.008. 18, 19

A. Below, J. A. De Loera, and J. Richter-Gebert. The complexity of finding small triangulations of convex 3-polytopes. *Journal of Algorithms*, 50(2):134–167, 2004. doi: 10.1016/S0196-6774(03)00092-0. 47

J. L. Bentley and T. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computers*, 28(9):643–647, 1979. doi: 10.1109/TC.1979.1675432. 96

E. D. Berger, K. S. McKinley, R. D. Blumofe, and P. R. Wilson. Hoard: a scalable memory allocator for multithreaded applications. *ACM SIGPLAN Notices*, 35(11):117–128, 2000. doi: http://doi.acm.org/10.1145/356989.357000. 83

M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82(1):64–84, 1989. doi: 10.1016/0021-9991(89)90035-1. 5, 8

P. Bochev and M. Shashkov. Constrained interpolation (remap) of divergence-free fields. *Computer Methods in Applied Mechanics and Engineering*, 194(2-5):511–530, 2005. doi: 10.1016/j.cma.2004.05.018. 31

J. Bonet and J. Peraire. An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems. *International*

Journal for Numerical Methods in Engineering, 31(1):1–17, 1991. doi: 10.1002/nme.1620310102. 96, 107

H. Borouchaki, P.-L. George, F. Hecht, P. Laug, and E.. Saltel. Delaunay mesh generation governed by metric specifications. Part I: Algorithms. *Finite Elements in Analysis and Design*, 25(1-2):61–83, 1997a. doi: 10.1016/S0168-874X(96)00057-1. 12, 17, 115

H. Borouchaki, P.-L. George, and B. Mohammadi. Delaunay mesh generation governed by metric specifications Part II. Applications. *Finite Elements in Analysis and Design*, 25(1-2):85–109, 1997b. doi: 10.1016/S0168-874X(96)00065-0. 12

H. Borouchaki, F. Hecht, and P. J. Frey. Mesh gradation control. *International Journal for Numerical Methods in Engineering*, 43(6):1143–1165, 1998. doi: 10.1002/(SICI)1097-0207(19981130)43:6<1143::AID-NME470>3.0.CO;2-I. 24, 25

F. J. Bossen and P. S. Heckbert. A pliant method for anisotropic mesh generation. In *5th International Meshing Roundtable*, pages 63–74, 1996. 12

C. L. Bottasso. Anisotropic mesh adaption by metric-driven optimization. *International Journal for Numerical Methods in Engineering*, 60 (3):597–639, 2004. doi: 10.1002/nme.977). 14

R. Boussetta, T. Coupez, and L. Fourment. Adaptive remeshing based on a posteriori error estimation for forging simulation. *Computer Methods in Applied Mechanics and Engineering*, 195(48-49): 6626–6645, 2006. doi: 10.1016/j.cma.2005.06.029. 14

J. U. Brackbill. An adaptive grid with directional control. *Journal of Computational Physics*, 108(1):38–50, 1993. doi: 10.1006/jcph.1993.1161. 9

D. M. Bradley. Counting the positive rationals: A brief survey, 2005. url: `http://www.citebase.org/abstract?id=oai:arXiv.org:math/0509025`. 75

J. H. Bramble and M. Zlámal. Triangular elements in the finite element method. *Mathematics of Computation*, 24(112):809–820, 1970. url: http://www.jstor.org/stable/2004615. 16

E. Briere de l'Isle and P. L. George. Optimization of tetrahedral meshes. In I. Babuška, W. D. Henshaw, J.E. Oliger, J.E. Flaherty, J.E. Hopcroft, and T. Tezduyar, editors, *IMA Volumes in Mathematics and its Applications*, volume 75, pages 97–128. Springer, 1995. 12

C. J. Budd and M. D. Piggott. Geometric integration and its applications. In P. G. Ciarlet and F. Cucker, editors, *Foundations of Computational Mechanics*, volume XI of *Handbook of Numerical Analysis*, pages 35–139. Elsevier, 2003. ISBN 0444703667. 40

C. J. Budd, W. Huang, and R. D. Russell. Adaptivity with moving grids. *Acta Numerica*, 18:111–241, 2009. doi: 10.1017/S0962492906400015. 5

C. Burstedde, O. Ghattas, G. Stadler, T. Tu, and L. C. Wilcox. Towards adaptive mesh PDE simulations on petascale computers. In *Proceedings of TeraGrid 08*. Las Vegas, Nevada, 2008. url: http://teragrid.org/events/teragrid08/Papers/papers/86.pdf. 27

G. C. Buscaglia and E. A. Dari. Anisotropic mesh optimization and its applications in adaptivity. *International Journal for Numerical Methods in Engineering*, 40(22):4119–4136, 1997. doi: 10.1002/(SICI)1097-0207(19971130)40:22<4119::AID-NME254>3.0.CO;2-R. 13, 23

C

A. S. Candy. *Subgrid scale modelling of transport processes*. PhD thesis, Imperial College London, 2008. 122

W. Cao. On the error of linear interpolation and the orientation, aspect ratio, and internal angles of a triangle. *SIAM Journal of Numerical Analysis*, 43(1):19–40, 2005. doi: 10.1137/S0036142903433492. 19

W. Cao. Anisotropic measures of third order derivatives and the quadratic interpolation error on triangular elements. *SIAM Journal of*

Scientific Computation, 29(2):756–781, 2007. doi: 10.1137/050634700. 19

W. Cao. An interpolation error estimate in $\mathbb{R}^2$ based on the anisotropic measures of higher order derivatives. *Mathematics of Computation*, 77 (261):265–286, 2008. doi: 10.1090/S0025-5718-07-01981-3. 19

G. F. Carey, G. Bicken, V. Carey, C. Berger, and J. Sanchez. Locally constrained projections on grids. *International Journal for Numerical Methods in Engineering*, 50(3):549–577, 2001. doi: 10.1002/1097-0207(20010130)50:3<549::AID-NME35>3.0.CO;2-S. 31, 132

M. J. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaptation for flow simulations. *International Journal for Numerical Methods in Fluids*, 25(4):475–491, 1997. doi: 10.1002/(SICI)1097-0363(19970830)25:4<475::AID-FLD575>3.0.CO;2-6. 17, 115, 116

M.J. Castro-Díaz, F. Hecht, and B. Mohammadi. New progress in anisotropic grid adaptation for inviscid and viscous flows simulations. Technical Report RR-2671, INRIA Rocquencourt, Rocquencourt, Le Chesnay, France, 1995. url: `http://hal.inria.fr/inria-00074019/en/`. 17

T. M. Chan. A simple trapezoid sweep algorithm for reporting red/blue segment intersections. In *Proceedings of the 6th Canadian Conference on Computational Geometry*, pages 263–268, 1994. 96

B. Chazelle. An optimal algorithm for intersecting three-dimensional convex polyhedra. *SIAM Journal on Computing*, 21(4):671–696, 1992. doi: 10.1137/0221041. 78

B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete and Computational Geometry*, 9(1):145–158, 1993. doi: 10.1007/BF02189314. 96

B. Chazelle and D. P. Dobkin. Detection is easier than computation. In *Proceedings of the 12th annual ACM symposium on Theory of Computing*, pages 146–153, New York, NY, USA, 1980. ACM. ISBN 0-89791-017-6. doi: 10.1145/800141.804662. 97

L. Chen, P. Sun, and J. Xu. Optimal anisotropic meshes for minimizing interpolation errors in l^p-norm. *Mathematics of Computation*, 76(257): 179–204, 2007. doi: 10.1090/S0025-5718-06-01896-5. 18

G. Chesshire and W. D. Henshaw. A scheme for conservative interpolation on overlapping grids. *SIAM Journal on Scientific Computing*, 15(4):819–845, 1994. doi: 10.1137/0915051. 43

L. P. Chew. Constrained Delaunay triangulations. *Algorithmica*, 4(1): 97–108, 1989. doi: 10.1007/BF01553881. 73, 74

L. P. Chew. Guaranteed-quality mesh generation for curved surfaces. In *Proceedings of the 9th annual symposium on Computational Geometry*, pages 274–280. ACM New York, NY, USA, 1993. 73

S. Chippada, C. N. Dawson, M. L. Martínez, and M. F. Wheeler. A projection method for constructing a mass conservative velocity field. *Computer Methods in Applied Mechanics and Engineering*, 157(1-2): 1–10, 1998. doi: 10.1016/S0045-7825(98)80001-7. 31

P. Clément. Approximation by finite element functions using local regularization. *RAIRO Analyse Numérique*, 9:77–84, 1975. 17

G. Compère, E. Marchandise, and J.-F. Remacle. Transient adaptivity applied to two-phase incompressible flows. *Journal of Computational Physics*, 227(3):1923–1942, 2008. doi: 10.1016/j.jcp.2007.10.002. 15

R. Cools and A. Haegemans. Algorithm 824: CUBPACK: a package for automatic cubature; framework description. *ACM Transactions on Mathematical Software*, 29(3):287–296, 2003. doi: 10.1145/838250.838253. 80

C. J. Cotter, D. A. Ham, and C. C. Pain. A mixed discontinuous/continuous finite element pair for shallow-water ocean modelling. *Ocean Modelling*, 26(1-2):86–90, 2009a. doi: 10.1016/j.ocemod.2008.09.002. 86

C. J. Cotter, D. A. Ham, C. C. Pain, and S. Reich. LBB stability of a mixed Galerkin finite element pair for fluid flow simulations. *Journal of Computational Physics*, 228(2):336–348, 2009b. doi: 10.1016/j.jcp.2008.09.014. 86

Y. Coudière, B. Palmerio, A. Dervieux, and D. Leservoisier. Accuracy barriers in mesh adaptation. Technical Report RR-4528, INRIA Sophia Antipolis, Sophia Antipolis, FR 06902, 2002. url: `http://www.inria.fr/rrrt/rr-4528.html`. 17

T. Coupez, H. Digonnet, and R. Ducloux. Parallel meshing and remeshing. *Applied Mathematical Modelling*, 25(2):153–175, 2000. doi: 10.1016/S0307-904X(00)00045-7. 25, 26

F. Courty, D. Leservoisier, P.-L. George, and A. Dervieux. Continuous metrics and mesh adaptation. *Applied Numerical Mathematics*, 56(2): 117–145, 2006. doi: 10.1016/j.apnum.2005.03.001. 18

M. Cullen. Modelling atmospheric flows. *Acta Numerica*, 16:67–154, 2007. doi: 10.1017/S0962492906290019. 40, 49

D

D. R. Davies, J. H. Davies, O. Hassan, K. Morgan, and P. Nithiarasu. Investigations into the applicability of adaptive finite element methods to two-dimensional infinite Prandtl number thermal and thermochemical convection. *Geochemistry Geophysics Geosystems*, 8(5), 2007. doi: 10.1029/2006GC001470. 31, 49

E. D'Azevedo. On optimal bilinear quadrilateral meshes. *Engineering with Computers*, 15(3):219–227, 1999. doi: 10.1007/s003660050017. 16

E. F. D'Azevedo. Optimal triangular mesh generation by coordinate transformation. *SIAM Journal on Scientific and Statistical Computing*, 12(4):755–786, 1991. doi: 10.1137/0912040. 11, 16

E. F. D'Azevedo and R. B. Simpson. On optimal triangular meshes for minimizing the gradient error. *Numerical Mathematics*, 59(1):321–348, 1991. doi: 10.1007/BF01385784. 11, 16

C. de Boor. Good approximation by splines with variable knots. In *Spline Functions and Approximation Theory*, volume 21 of *International Series of Numerical Mathematics*, pages 57–72. Birkhäuser, Basel, 1973. 9

J. de Frutos and J. M. Sanz-Serna. Accuracy and conservation properties in numerical integration: the case of the Korteweg-de Vries equation. *Numerische Mathematik*, 75(4):421–445, 1997. doi: 10.1007/s002110050247. 40

A. Dervieux, D. Leservoisier, P.-L. George, and Y. Coudière. About theoretical and practical impact of mesh adaptation on approximation of functions and PDE solutions. *International Journal for Numerical Methods in Fluids*, 43(5):507–516, 2003. doi: 10.1002/fld.503. 18

D. Deutsch. *The Fabric of Reality*. Penguin, 1998. ISBN 014027541X. 2

K. Devine, E. Boman, R. Heaphy, B. Hendrickson, and C. Vaughan. Zoltan data management services for parallel dynamic applications. *Computing in Science and Engineering*, 4(2):90–97, 2002. doi: 10.1109/5992.988653. 26

D. P. Dobkin and D. G. Kirkpatrick. Fast detection of polyhedral intersection. *Theoretical Computer Science*, 27(3):241–253, 1983. doi: 10.1016/0304-3975(82)90120-7. 97

D. P. Dobkin and D. G. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra. *Journal of Algorithms*, 6(3): 381–392, 1985. doi: 10.1016/0196-6774(85)90007-0. 97

D. P. Dobkin and D. G. Kirkpatrick. Determining the separation of preprocessed polyhedra: a unified approach. In *17th International Colloquium on Automata, Languages and Programming*, pages 400–413, Warwick, England, 1990. Springer. 97

V. Dolejší. Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes. *Computing and Visualization in Science*, 1(3):165–178, 1998. doi: 10.1007/s007910050015. 13

J. Dompierre, P. Labbé, F. Guibault, and R. Camarero. Proposal of benchmarks for 3D unstructured tetrahedral mesh optimization. In *Proceedings of the 7th International Meshing Roundtable*, pages 459–478, Dearborn, Michigan, 1998. 14

J. Dompierre, M. G. Vallet, Y. Bourgault, M. Fortin, and W. G. Habashi. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part III. Unstructured meshes. *International Journal for Numerical Methods in Fluids*, 39(8):675–702, 2002. doi: 10.1002/fld.357. 14

J. Donea and A. Huerta. *Finite element methods for flow problems*. John Wiley & Sons Ltd Chichester, UK, 2003. 136

Q. Du and D. Wang. Anisotropic centroidal Voronoi tessellations and their applications. *SIAM Journal on Scientific Computing*, 26(3):737–761, 2005. doi: 10.1137/S1064827503428527. 15

J. K. Dukowicz. Conservative rezoning (remapping) for general quadrilateral meshes. *Journal of Computational Physics*, 54(3):411–424, 1984. doi: 10.1016/0021-9991(84)90125-6. 42, 43

J. K. Dukowicz and J. W. Kodis. Accurate conservative remapping (rezoning) for arbitrary Lagrangian-Eulerian computations. *SIAM Journal on Scientific and Statistical Computing*, 8(3):305–321, 1987. doi: 10.1137/0908037. 42

E

D. H. Eberly. *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*. Morgan Kaufmann, 2001. 80

H. Edelsbrunner and L. Guibas. Topologically sweeping an arrangement. *Journal of Computer and System Sciences*, 38(1):165–194, 1989. doi: 10.1016/0022-0000(89)90038-X. 99

H. Edelsbrunner, F. P. Preparata, and D. B. West. Tetrahedrizing point sets in three dimensions. *Journal of Symbolic Computation*, 10 (3/4):335–348, 1990. doi: 10.1016/S0747-7171(08)80068-5. 47

A. El Hraiech, H. Borouchaki, P. Villon, and L. Moreau. Mechanical field interpolation. In L. M. Smith, F. Pourboghrat, J. Cao, T. B. Stoughton, J.-W. Yoon, M. F. Shi, C.-T. Wang, and L. Zhang, editors, *Proceedings of the 6th International Conference and Workshop*

on Numerical Simulation of 3D Sheet Metal Forming Process, pages 201–208, Detroit, Michigan, 2005. AIP. doi: 10.1063/1.2011218. 30, 80, 81

E. Ezra and M. Sharir. Counting and representing intersections among triangles in three dimensions. In *SCG '04: Proceedings of the 20th annual symposium on Computational Geometry*, pages 210–219, New York, NY, USA, 2004. ACM. ISBN 1-58113-885-7. doi: 10.1145/997817.997851. 96

F

F. Fang, M. D. Piggott, C. C. Pain, G. J. Gorman, and A. J. H. Goddard. An adaptive mesh adjoint data assimilation method. *Ocean Modelling*, 15(1-2):39–55, 2006. doi: 10.1016/j.ocemod.2006.02.002. 125

F. Fang, C. C. Pain, I. M. Navon, M. D. Piggott, G. J. Gorman, P. E. Farrell, P. A. Allison, and A. J. H. Goddard. A POD reduced-order 4D-Var adaptive mesh ocean modelling approach. *International Journal for Numerical Methods in Fluids*, 60(7):709–732, 2008. doi: 10.1002/fld.1911. 127

F. Fang, C. C. Pain, I. M. Navon, G. J. Gorman, M. D. Piggott, and P. A. Allison. The Independent Set Perturbation adjoint method: a new method of differentiating mesh based fluids models. Submitted to International Journal for Numerical Methods in Fluids, 2009a. 125

F. Fang, C. C. Pain, I. M. Navon, G. J. Gorman, M. D. Piggott, P. A. Allison, P. E. Farrell, and A. J. H. Goddard. A POD reduced order unstructured mesh ocean modelling method for moderate Reynolds number flows. *Ocean Modelling*, 28(1-3):127–136, 2009b. doi: 10.1016/j.ocemod.2008.12.006. 127

C. Farhat, M. Lesoinne, and P. Le Tallec. 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, 157(1-2):95–114, 1998. doi: 10.1016/S0045-7825(97)00216-8. 32

L. Farnell and R. A. Plumb. Numerical integration of flow in a rotating annulus I: axisymmetric model. Occasional Note Met O 21 75/3, December 1975. unpublished. 88

P. E. Farrell and J. R. Maddison. Conservative interpolation between volume meshes by local Galerkin projection, 2009. Submitted to Computer Methods in Applied Mechanics and Engineering. 39, 71, 73, 95

P. E. Farrell, M. D. Piggott, C. C. Pain, and G. J Gorman. Conservative interpolation between unstructured meshes via supermesh construction. *Computer Methods in Applied Mechanics and Engineering*, 198(33-36):2632–2642, 2009. doi: 10.1016/j.cma.2009.03.004. 32, 39, 41, 44, 71, 72, 73

U. Finke and K. H. Hinrichs. Overlaying simply connected planar subdivisions in linear time. In *Proceedings of the 11th annual symposium on Computational Geometry*, pages 119–126. ACM New York, NY, USA, 1995. 96

L. Formaggia and S. Perotto. New anisotropic a priori error estimates. *Numerische Mathematik*, 89(4):641–667, 2001. doi: 10.1007/s002110100273. 17, 114

L. Formaggia and S. Perotto. Anisotropic error estimates for elliptic problems. *Numerische Mathematik*, 94(1):67–92, 2003. doi: 10.1007/s00211-002-0415-z. 17, 92

L. Formaggia, S. Micheletti, and S. Perotto. Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection–diffusion–reaction and the Stokes problems. *Applied Numerical Mathematics*, 51(4):511–533, 2004. doi: 10.1016/j.apnum.2004.06.007. 21

W. R. Franklin, V. Sivaswami, D. Sun, M. S. Kankanhalli, and C. Narayanaswami. Calculating the area of overlaid polygons without constructing the overlay. *Cartography and Geographic Information Science*, 21(9):81–89, 1994. doi: 10.1559/152304094782610260. 42, 43, 49

T. Freeth, Y. Bitsakis, X. Moussas, J. H. Seiradakis, A. Tselikas, H. Mangou, M. Zafeiropoulou, R. Hadland, D. Bate, A. Ramsey, M. Allen, A. Crawley, P. Hockley, T. Malzbender, D. Gelb, W. Ambrisco, and M. G. Edmunds. Decoding the ancient Greek astronomical calculator known as the Antikythera Mechanism. *Nature*, 444(7119): 587–591, 2006. doi: 10.1038/nature05357. 2

T. Freeth, A. Jones, J. M. Steele, and Y. Bitsakis. Calendars with Olympiad display and eclipse prediction on the Antikythera Mechanism. *Nature*, 454(7204):614–617, 2008. doi: 10.1038/nature07130. 2

L. Freitag, M. Jones, and P. Plassmann. A parallel algorithm for mesh smoothing. *SIAM Journal on Scientific Computing*, 20(6):2023–2040, 1999. doi: 10.1137/S1064827597323208. 25, 27

L. A. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering*, 40(21):3979–4002, 1997. 13

P. J. Frey and F. Alauzet. Anisotropic mesh adaptation for CFD computations. *Computer methods in applied mechanics and engineering*, 194:5068–5082, 2005. doi: 10.1016/j.cma.2004.11.025. 16

P.J. Frey and P.L. George. *Mesh Generation. Application to finite elements*. Wiley, 2nd edition, 2008. 9

G

F. Ganovelli, F. Ponchio, and C. Rocchini. Fast tetrahedron-tetrahedron overlap algorithm. *Journal of Graphics Tools*, 7(2):17–26, 2002. url: `http://jgt.akpeters.com/papers/GanovelliPonchioRocchini02/`. 98

R. Garimella, M. Kucharik, and M. Shashkov. An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes. *Computers and Fluids*, 36(2):224–237, 2007. doi: 10.1016/j.compfluid.2006.01.014. 42

C. W. Gear. Invariants and numerical methods for ODEs. *Physica D: Nonlinear Phenomena*, 60(1-4):303–310, 1992. doi: 10.1016/0167-2789(92)90246-J. 40

J. A. George. *Computer implementation of the finite element method.* PhD thesis, Stanford University, Stanford, CA, USA, 1971. 96

P. L. George and H. Borouchaki. *Delaunay Triangulation and Meshing: Application to Finite Elements.* Hermes, 1998. 7, 9, 13, 29, 43, 71, 114

C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. doi: 10.1002/nme.2579. 54, 57

C. Geuzaine, B. Meys, F. Henrotte, P. Dular, and W. Legros. A Galerkin projection method for mixed finite elements. *IEEE Transactions on Magnetics*, 35(3):1438–1441, 1999. doi: 10.1109/20.767236. 30

M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65(3-4):393–415, 2000. doi: 10.1023/A:1011430410075. 124

G. J. Gorman. *Parallel Anisotropic Unstructured Mesh Optimisation and its Applications.* PhD thesis, Imperial College London, London, UK, 2003. 26

G. J. Gorman, M. D. Piggott, C. C. Pain, C. R. E. de Oliveira, A. P. Umpleby, and A. J. H. Goddard. Optimisation based bathymetry approximation through constrained unstructured mesh adaptivity. *Ocean Modelling*, 12(3-4):436–452, 2006. doi: 10.1016/j.ocemod.2005.09.004. 33

G.J. Gorman, C. C. Pain, M.D. Piggott, A.P. Umpleby, J.R. Maddison, and P.E. Farrell. Dynamically load-balanced anisotropic mesh adaptivity. In preparation, 2009. 26

J. Grandy. Conservative remapping and region overlays by intersecting arbitrary polyhedra. *Journal of Computational Physics*, 148(2):433–466, 1999. doi: 10.1006/jcph.1998.6125. 42, 43, 49, 51, 107

A. Griewank. *Evaluating derivatives: principles and techniques of algorithmic differentiation.* Frontiers in Applied Mathematics. SIAM, 2000. ISBN 0898714516. 125

C. Gruau and T. Coupez. 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. *Computer Methods in Applied Mechanics and Engineering*, 194(48-49):4951–4976, 2005. doi: 10.1016/j.cma.2004.11.020. 14

L. J. Guibas and R. Seidel. Computing convolutions by reciprocal search. *Discrete and Computational Geometry*, 2(1):175–193, 1987. doi: 10.1007/BF02187878. 96

M. D. Gunzburger. *Perspectives in Flow Control and Optimization.* Advances in Design and Control. Society for Industrial Mathematics, 2003. ISBN 089871527X. 124, 125

P. Gupta, R. Janardan, and M. Smid. Efficient algorithms for counting and reporting pairwise intersections between convex polygons. *Information Processing Letters*, 69(1):7–13, 1999. doi: 10.1016/S0020-0190(98)00187-2. 96

P. Gupta, R. Janardan, and M. Smid. Computational geometry: Generalized intersection searching. In D. Mehta and S. Sahni, editors, *Handbook of Data Structures and Applications*, pages 1–17. Chapman & Hall/CRC, Boca Raton, FL, 2005. 96

A. Guttman. R-trees: a dynamic index structure for spatial searching. *ACM SIGMOD Record*, 14(2):47–57, 1984. doi: 10.1145/971697.602266. 48, 75, 78, 103, 107

H

W. G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M.-G. Vallet. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles. *International Journal for Numerical Methods in Fluids*, 32(6):725–744, 2000. doi: 10.1002/(SICI)1097-0363(20000330)32:6<725::AID-FLD935>3.0.CO;2-4. 14

C. Härtel, E. Meiburg, and F. Necker. Analysis and direct numerical simulation of the flow at a gravity-current head. Part I. Flow topology and front speed for slip and no-slip boundaries. *Journal of Fluid Mechanics*, 418(1):189–212, 2000. 87, 128

O. Hassan, E. J. Probert, and K. Morgan. Unstructured mesh procedures for the simulation of three-dimensional transient compressible inviscid flows with moving boundary components. *International Journal for Numerical Methods in Fluids*, 27(1-4):41–55, 1998. doi: 10.1002/(SICI)1097-0363(199801)27:1/4<41::AID-FLD649>3.0.CO;2-5. 13

O. Hassan, E. J. Probert, K. Morgan, and N. P. Weatherill. Unsteady flow simulation using unstructured meshes. *Computer Methods in Applied Mechanics and Engineering*, 189(4):1247–1275, 2000. doi: 10.1016/S0045-7825(99)00376-X. 13

O. Hassan, K. Morgan, and N. Weatherill. Unstructured mesh methods for the solution of the unsteady compressible flow equations with moving boundary components. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1859): 2531–2552, 2007. doi: 10.1098/rsta.2007.2020. 13

M. W. Heinstein and T. A. Laursen. A three dimensional surface-to-surface projection algorithm for non-coincident domains. *Communications in Numerical Methods in Engineering*, 19(6):421–432, 2003. doi: 10.1002/cnm.601. 32

R. Hide and P. J. Mason. Sloping convection in a rotating fluid. *Advances in Physics*, 24(1):47–100, 1975. doi: 10.1080/00018737500101371. 119

P. Hignett, A. A. White, R. D. Carter, W. D. N. Jackson, and R. M. Small. A comparison of laboratory measurements and numerical simulations of baroclinic wave flows in a rotating cylindrical annulus. *Quarterly Journal of the Royal Meteorological Society*, 111(467):131–154, 1985. doi: 10.1002/qj.49711146705. 88

C. W. Hirt, A. A. Amsden, and J. L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computa-*

tional Physics, 14(3):227, 1974. doi: 10.1016/0021-9991(74)90051-5. 41

J. E. Hopcroft, J. T. Schwartz, and M. Sharir. Efficient detection of intersections among spheres. *The International Journal of Robotics Research*, 2(4):77, 1983. doi: 10.1177/027836498300200405. 96

P. Houston and E. Süli. *hp*-Adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems. *SIAM Journal on Scientific Computing*, 23(4):1226–1252, 2001. doi: 10.1137/S1064827500378799. 5

W. Huang. Metric tensors for anisotropic mesh generation. *Journal of Computational Physics*, 204(2):633–665, 2005. doi: 10.1016/j.jcp.2004.10.024. 19

W. Huang. Mathematical principles of anisotropic mesh adaptation. *Communications in Computational Physics*, 1(2):276–310, 2006. url: `http://www.global-sci.com/issue/contents/1/issue2.html`. 19

M. E. Hubbard, M. J. Baines, and P. K. Jimack. Consistent Dirichlet boundary conditions for numerical solution of moving boundary problems. *Applied Numerical Mathematics*, 59(6):1337–1353, 2009. doi: 10.1016/j.apnum.2008.08.002. 52

T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41): 4135–4195, 2005. doi: 10.1016/j.cma.2004.10.008. 34

I

A. Iske and M. Käser. Conservative semi-Lagrangian advection on adaptive unstructured meshes. *Numerical Methods for Partial Differential Equations*, 20(3):338–411, 2004. doi: 10.1002/num.10100. 43

J

R. K. Jaiman, X. Jiao, P. H. Geubelle, and E. Loth. Conservative load transfer along curved fluid–solid interface with non-matching meshes. *Journal of Computational Physics*, 218(1):372–397, 2006. doi: 10.1016/j.jcp.2006.02.016. 32, 44

X. Jiao and M. T. Heath. Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. *International Journal for Numerical Methods in Engineering*, 61:2402–2427, 2004a. 43, 54, 65

X. Jiao and M. T. Heath. Overlaying surface meshes, Part I: Algorithms. *International Journal on Computational Geometry and Applications*, 14(6):379–402, 2004b. 32, 44

X. Jiao, H. Edelsbrunner, and M. T. Heath. Mesh association: Formulation and algorithms. In *Proceedings of the 8th International Meshing Roundtable*, South Lake Tahoe, California, 1999. 32

M. T. Jones and P. E. Plassmann. Parallel algorithms for adaptive mesh refinement. *SIAM Journal on Scientific Computing*, 18(3):686–708, 1997. doi: 10.1137/S106482759528065X. 8

K

G. Karypis and V. Kumar. Parallel multilevel series k-way partitioning scheme for irregular graphs. *SIAM Review*, 41(2):278–300, 1999. doi: 10.1137/S0036144598334138. 26

S. R. Kennon and G. S. Dulikravich. Generation of computational grids using optimization. *AIAA Journal*, 24(7):1069–1073, 1986. doi: 10.2514/3.9393. 12

S. Korotov. Error control in terms of linear functionals based on gradient averaging techniques. *Computing Letters*, 3(1):35–44, 2007. doi: 10.1163/157404007779994241. 125

M. Kucharik and M. Shashkov. Extension of efficient, swept-integration-based conservative remapping method for meshes with

changing connectivity. *International Journal for Numerical Methods in Fluids*, 56(8):1359–1365, 2007. doi: 10.1002/fld.1577. 42

L

H. Le, P. Moin, and J. Kim. Direct numerical simulation of turbulent flow over a backward-facing step. *Journal of Fluid Mechanics*, 330: 349–374, 1997. 123

P. D. Ledger, K. Morgan, J. Peraire, O. Hassan, and N. P. Weatherill. The development of an *hp*-adaptive finite element procedure for electromagnetic scattering problems. *Finite Elements in Analysis and Design*, 39(8):751–764, 2003. doi: 10.1016/S0168-874X(03)00057-X. 14th Robert J. Melosh Competition. 5

D. T. Lee and A. K. Lin. Generalized Delaunay triangulation for planar graphs. *Discrete and Computational Geometry*, 1(1):201–217, 1986. doi: 10.1007/BF02187695. 73

B. P. Leonard. The ultimate conservative difference scheme applied to unsteady one-dimensional advection. *Computer Methods in Applied Mechanics and Engineering*, 88(1):17–74, 1991. doi: 10.1016/0045-7825(91)90232-U. 92

C. Y. Lepage, A. St.-Cyr, and W. G. Habashi. Parallel unstructured mesh adaptation on distributed-memory systems. In *Proceedings of the 34th AIAA Fluid Dynamics Conference and Exhibit*, 2004. 26

C. Y. Lepage, A. St-Cyr, and W. G. Habashi. MPI parallelization of unstructured mesh adaptation. In C. Groth and D. W. Zingg, editors, *Proceedings of the Third International Conference on Computational Fluid Dynamics*, pages 727–732. Springer, 2006. doi: 10.1007/3-540-31801-1_105. 27

J. Levon and P. Elie. Oprofile: a system profiler for Linux, 2009. url: `http://oprofile.sourceforge.net`. 83

X. Li, M. S. Shephard, and M. W. Beall. Accounting for curved domains in mesh adaptation. *International Journal for Numerical Methods in Engineering*, 58(2):247–276, 2003. doi: 10.1002/nme.772. 34

X. Li, J. Remacle, N. Chevaugeon, and M. S. Shephard. Anisotropic mesh gradation control. In *Proceedings of the 13th International Meshing Roundtable*, Williamsburg, Virginia, 2004. 24, 25

X. Li, M. S. Shephard, and M. W. Beall. 3D anisotropic mesh adaptation by mesh modification. *Computer Methods in Applied Mechanics and Engineering*, 194:4915–4950, 2005. doi: 10.1016/j.cma.2004.11.019. 14, 34

D. Linehan. *SpaceShipOne: An Illustrated History*. Zenith Press, 2008. ISBN 076033188X. 3

K. Lipnikov and Y. Vasilevskii. Analysis of Hessian recovery methods for generating adaptive meshes. In P. P. Pébay, editor, *Proceedings of the 15th International Meshing Roundtable*, pages 163–171, Birmingham, Alabama, 2006. Springer. doi: 10.1007/978-3-540-34958-7_10. 22, 23, 24

K. Lipnikov and Y. Vassilevski. Parallel adaptive solution of 3D boundary value problems by Hessian recovery. *Computer Methods in Applied Mechanics and Engineering*, 192(11-12):1495–1513, 2003. doi: 10.1016/S0045-7825(02)00654-0. 26

K. Lipnikov and Y. Vassilevski. On discrete boundaries and solution accuracy in anisotropic adaptive meshing. In B. W. Hanks, editor, *Proceedings of the 14th International Meshing Roundtable*, pages 312–324, San Diego, California, 2005. Springer. doi: 10.1007/3-540-29090-7_19. 34

S. H. Lo. A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical Methods in Engineering*, 21(8): 1403–1426, 1985. doi: 10.1002/nme.1620210805. 96

R. Löhner. An adaptive finite element solver for transient problems with moving bodies. *Computers and Structures*, 30(1):303–317, 1988. doi: 10.1016/0045-7949(88)90236-2. 10

R. Löhner. Adaptive remeshing for transient problems. *Computer Methods in Applied Mechanics and Engineering*, 75(1-3):195–214, 1989. doi: 10.1016/0045-7825(89)90024-8. 10

R. Löhner. Three-dimensional fluid-structure interaction using a finite element solver and adaptive remeshing. *Computing Systems in Engineering*, 1(2):257–272, 1990. doi: 10.1016/0956-0521(90)90012-A. 10

R. Löhner. Robust, vectorized search algorithms for interpolation on unstructured grids. *Journal of Computational Physics*, 118(2):380–387, 1995a. doi: 10.1006/jcph.1995.1107. 29, 31, 41, 48, 76, 96

R. Löhner. Mesh adaptation in fluid mechanics. *Engineering Fracture Mechanics*, 50(5-6):819–831, 1995b. doi: 10.1016/0013-7944(94)E0062-L. 8, 11

R. Löhner. Extensions and improvements of the advancing front grid generation technique. *Communications in Numerical Methods in Engineering*, 12(10):683–702, 1996. doi: 10.1002/(SICI)1099-0887(199610)12:10<683::AID-CNM983>3.0.CO;2-1. 24

R. Löhner and P. Parikh. Generation of three-dimensional unstructured grids by the advancing-front method. *International Journal for Numerical Methods in Fluids*, 8(10):1135–1149, 1988. doi: 10.1002/fld.1650081003. 96

A. Loseille. *Adaptation de maillage anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluides. Application à la prédiction haute-fidélité du bang sonique.* PhD thesis, Université Pierre et Marie Curie, Paris VI, 2008. 21

X. Luo, M. S. Shephard, J. F. Remacle, R. M. O'Bara, M. W. Beall, B. A. Szabó, and R. Actis. *p*-Version mesh generation issues. In *Proceedings of the 11th International Meshing Roundtable*, pages 343–354, Ithaca, New York, 2002. 34

M

N. Maman and C. Farhat. Matching fluid and structure meshes for aeroelastic computations: A parallel approach. *Computers & Structures*, 54(4):779–785, 1995. doi: 10.1016/0045-7949(94)00359-B. 32

Y. Manolopoulos, A. Nanopoulos, A.N. Papadopoulos, and Y. Theodoridis. *R-trees: Theory And Applications*. Springer, 2005. 75, 78, 103, 107

S. Mansoorzadeh, C. C. Pain, C. R. E. de Oliveira, and A. J. H. Goddard. Finite element simulations of incompressible flow past a heated/cooled sphere. *International Journal for Numerical Methods in Fluids*, 28(6):903–915, 1998. doi: 10.1002/(SICI)1097-0363(19981030)28:6<903::AID-FLD746>3.0.CO;2-O. 82

L. G. Margolin and M. Shashkov. Second-order sign-preserving conservative interpolation (remapping) on general grids. *Journal of Computational Physics*, 184(1):266–298, 2003. doi: 10.1016/S0021-9991(02)00033-5. 41

D. J. Mavriplis. Adaptive mesh generation for viscous flows using Delaunay triangulation. *Journal of Computational Physics*, 90(2):271–291, 1990. doi: 10.1016/0021-9991(90)90167-Y. 10

E. Meijering. A chronology of interpolation: from ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90 (3):319–342, 2002. doi: 10.1109/5.993400. 16

S. Micheletti and S. Perotto. Reliability and efficiency of an anisotropic Zienkiewicz-Zhu error estimator. *Computer Methods in Applied Mechanics and Engineering*, 195(9-12):799–835, 2006. doi: 10.1016/j.cma.2005.02.009. 92, 115

T. Möller. A fast triangle-triangle intersection test. *Journal of Graphics Tools*, 2(2):25–30, 1997. url: `http://jgt.akpeters.com/papers/Moller97/`. 98

K. Morgan, J. Peraire, J. Peiró, and O. Hassan. The computation of three-dimensional flows using unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 87(2-3):335–352, 1991. doi: 10.1016/0045-7825(91)90012-U. 3, 10, 39

K. Morgan, O. Hassan, and N. Weatherill. Why didn't the supersonic car fly? *Mathematics Today*, 35(5):110–114, 1999. Winner of the Catherine Richards prize. 2

D. M. Mount. Geometric intersection. In J. E. Goodman and J. O'Rourke, editors, *Handbook of discrete and computational geometry*, pages 615–630. CRC Press, Inc., Boca Raton, FL, USA, 1997. ISBN 0-8493-8524-5. 43, 47, 77, 96

N

H. Nguyen, M. Gunzburger, L. Ju, and J. Burkardt. Adaptive anisotropic meshing for steady convection-dominated problems. *Computer Methods in Applied Mechanics and Engineering*, 2009. doi: 10.1016/j.cma.2009.05.001. 15

O

S. J. Owen and S. Saigal. Surface mesh sizing control. *International Journal for Numerical Methods in Engineering*, 47(497):511, 2000. doi: 10.1002/(SICI)1097-0207(20000110/30)47:1/3<497::AID-NME781>3.0.CO;2-H. 24

T. M. Özgökmen, T. Iliescu, P. F. Fischer, A. Srinivasan, and J. Duan. Large eddy simulation of stratified mixing in two-dimensional dam-break problem in a rectangular enclosed domain. *Ocean Modelling*, 16 (1-2):106–140, 2007. doi: 10.1016/j.ocemod.2006.08.006. 86

P

D. Pagnutti and C. Ollivier-Gooch. Delaunay-based anisotropic mesh adaptation. In R. V. Garimella, editor, *Proceedings of the 17th International Meshing Roundtable*, pages 141–157, Pittsburgh, Pennsylvania, 2008. Springer. doi: 10.1007/978-3-540-87921-3_9. 15

D. Pagnutti and C. Ollivier-Gooch. A generalized framework for high order anisotropic mesh adaptation. *Computers & Structures*, 87(11-12):670–679, 2009. doi: 10.1016/j.compstruc.2008.11.008. Fifth MIT Conference on Computational Fluid and Solid Mechanics. 19

C. C. Pain, A. P. Umpleby, C. R. E. de Oliveira, and A. J. H. Goddard. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Computer Methods in Applied Mechanics and Engineering*, 190(29-30):3771–3796, 2001. doi: 10.1016/S0045-7825(00)00294-2. 14, 23, 24, 68, 88, 111, 114, 116, 119, 123

C. C. Pain, J.L.M.A. Gomes, M.D. Eaton, C.R.E. de Oliveira, and A.J.H. Goddard. A model of heat transfer dynamics of coupled multiphase-flow and neutron-radiation: application to a nuclear fluidized bed reactor. *International Journal of Numerical Methods for Heat & Fluid Flow*, 15(8):765–807, 2005a. doi: 10.1108/09615530510625084. 69

C. C. Pain, M. D. Piggott, A. J. H. Goddard, F. Fang, G. J. Gorman, D. P. Marshall, M. D. Eaton, P. W. Power, and C. R. E. de Oliveira. Three-dimensional unstructured mesh ocean modelling. *Ocean Modelling*, 10(1-2):5–33, 2005b. doi: 10.1016/j.ocemod.2004.07.005. 117

G. Parent, P. Dular, J. P. Ducreux, and F. Piriou. Using a Galerkin projection method for coupled problems. *IEEE Transactions on Magnetics*, 44(6):830–833, 2008. doi: 10.1109/TMAG.2008.915798. 30

M. A. Park and D. L. Darmofal. Parallel anisotropic tetrahedral adaptation. In *Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2008. 15, 27

J. Peraire and K. Morgan. Unstructured mesh generation including directional refinement for aerodynamic flow simulation. *Finite Elements in Analysis & Design*, 25(3-4):343–356, 1997. doi: 10.1016/S0168-874X(96)00055-8. 12

J. Peraire, M. Vahdati, K. Morgan, and O.C Zienkiewicz. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 72(2):449–466, 1987. doi: 10.1016/0021-9991(87)90093-3. 9, 16

J. Peraire, J. Peiró, L. Formaggia, K. Morgan, and O. C. Zienkiewicz. Finite element Euler computations in three dimensions. *Interna-*

tional Journal for Numerical Methods in Engineering, 26(10):2135–2159, 1988. doi: 10.1002/nme.1620261002. 10, 96

J. Peraire, J. Peiró, and K. Morgan. Finite element multigrid solution of Euler flows past installed aero-engines. *Computational Mechanics*, 11(5):433–451, 1993. doi: 10.1007/BF00350098. 29

P.-O. Persson. Mesh size functions for implicit geometries and PDE-based gradient limiting. *Engineering with Computers*, 22(2):95–109, 2006. doi: 10.1007/s00366-006-0014-1. 24

T. N. Phillips and A. J. Williams. Conservative semi-Lagrangian finite volume schemes. *Numerical Methods for Partial Differential Equations*, 17(4):403–425, 2001. doi: 10.1002/num.1019. 43

M. D. Piggott, C. C. Pain, G. J. Gorman, P. W. Power, and A. J. H. Goddard. h, r, and hr adaptivity with applications in numerical ocean modelling. *Ocean Modelling*, 10(1-2):95–113, 2005. doi: 10.1016/j.ocemod.2004.07.007. 8

M. D. Piggott, G. J. Gorman, C. C. Pain, P. A. Allison, A. S. Candy, B. T. Martin, and M. R. Wells. A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes. *International Journal for Numerical Methods in Fluids*, 56(8):1003–1015, 2008. doi: 10.1002/fld.1663. 69, 82, 117, 119

M. D. Piggott, P. E. Farrell, C. R. Wilson, G. J. Gorman, and C. C. Pain. Anisotropic mesh adaptivity for multi-scale ocean modelling. Philosophical Transactions of the Royal Society A, 2009. In press. 3

P. W. Power. *Error Measures for Finite Element Ocean Modelling.* PhD thesis, Imperial College London, Kensington, London, UK, 2008. 21

P. W. Power, C. C. Pain, M. D. Piggott, G. J. Gorman, D. P. Marshall, A. J. H. Goddard, and I. M. Navon. Adjoint goal-based error norms for adaptive mesh ocean modelling. *Ocean Modelling*, 15(1-2):3–38, 2006. doi: 10.1016/j.ocemod.2006.05.001. 21

F. P. Preparata and M. I. Shamos. *Computational geometry: an introduction.* Springer, second edition, 1985. ISBN 0387961313. 47

J. D. Pryce. On the convergence of iterated remeshing. *IMA Journal of Numerical Analysis*, 9(3):315–335, 1989. doi: 10.1093/imanum/9.3.315. 11

R

P. L. Read. A combined laboratory and numerical study of heat transport by baroclinic eddies and axisymmetric flows. *Journal of Fluid Mechanics*, 489:301–323, 2003. doi: 10.1017/S002211200300524X. 88

M. Reichling. On the detection of a common intersection of k convex objects in the plane. *Information Processing Letters*, 29(1):25–29, 1988. doi: 10.1016/0020-0190(88)90127-5. 96

J.-F. Remacle, X. Li, M. S. Shephard, and J. E. Flaherty. Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering*, 62(7):899–923, 2005. doi: 10.1002/nme.1196. 14, 24, 44

J.-F. Remacle, S. S. Frazão, X. Li, and M. S. Shephard. An adaptive discretization of shallow-water equations based on discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, 52(8):903–923, 2006. doi: 10.1002/fld.1204. 30, 44

T. Richter. Funktionalorientierte Gitteroptimierung für die Finite Elemente Methode. Master's thesis, University of Heidelberg, 2001. 125

T. Richter. A posteriori error estimation and anisotropy detection with the dual-weighted residual method. *International Journal for Numerical Methods in Fluids*, 2009. doi: 10.1002/fld.2016. 9

T. D. Ringler and D. A. Randall. A potential enstrophy and energy conserving numerical scheme for solution of the shallow-water equations on a geodesic grid. *Monthly Weather Review*, 130(5):1397–1410, 2002. doi: 10.1175/1520-0493(2002)130<1397:APEAEC>2.0.CO;2. 40

S. Rippa. Long and thin triangles can be good for linear interpolation. *SIAM Journal of Numerical Analysis*, 29(1):257–270, 1992. doi: 10.1137/0729017. 16

M. Rudman. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids*, 24(7):671–691, 1997. doi: 10.1002/(SICI)1097-0363(19970415)24:7<671::AID-FLD508>3.0.CO;2-9. 68, 117

J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18(3):548–585, 1995. doi: 10.1006/jagm.1995.1021. 73

M. Rüter, S. Korotov, and C. Steenbock. Goal-oriented error estimates based on different FE-spaces for the primal and the dual problem with applications to fracture mechanics. *Computational Mechanics*, 39(6): 787–797, 2007. doi: 10.1007/s00466-006-0069-2. 44, 125

S

R. Sadourny. The dynamics of finite-difference models of the shallow-water equations. *Journal of the Atmospheric Sciences*, 32(4):680–689, 1975. doi: 10.1175/1520-0469(1975)032<0680:TDOFDM>2.0.CO;2. 40

O. Sahni, J. Müller, K. E. Jansen, M. S. Shephard, and C. A. Taylor. Efficient anisotropic adaptive discretization of the cardiovascular system. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5634–5655, 2006. doi: 10.1016/j.cma.2005.10.018. John H. Argyris Memorial issue, part II. 14

O. Sahni, K. E. Jansen, M. S. Shephard, C. A. Taylor, and M. W. Beall. Adaptive boundary layer meshing for viscous flow simulations. *Engineering with Computers*, 24(3):267–285, 2008. doi: 10.1007/s00366-008-0095-0. 14

W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, editors. *Model order reduction: theory, research aspects and applications*, volume 13 of *The European Consortium for Mathematics in Industry*. Springer, 2008. ISBN 978-3-540-78840-9. 127, 128

R. Schneider and P. K. Jimack. Anisotropic mesh adaption based on a posteriori estimates and optimisation of node positions. In G. Lube

and G. Rapin, editors, *Proceedings of the International Conference on Boundary and Interior Layers*, Germany, 2006. 9

J. Schöberl. NETGEN: An advancing front 2D/3D-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1 (1):41–52, 1997. doi: 10.1007/s007910050004. 83

W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. KitWare, Inc., fourth edition, 2006. ISBN 193093419X. url: `http://www.vtk.org`. 123

L. R. Scott and S. Zhang. Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Mathematics of Computation*, 54(190):483–493, 1990. url: `http://www.jstor.org/stable/2008497`. 17

J. S. Scroggs and F. H. M. Semazzi. A conservative semi-Lagrangian method for multidimensional fluid dynamics applications. *Numerical Methods for Partial Differential Equations*, 11(5):445–452, 1995. doi: 10.1002/num.1690110503. 43

R. Seidel. Convex hull computations. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 495–512. Chapman & Hall/CRC, 2004. ISBN 1584883014. 47

M. I. Shamos and D. Hoey. Geometric intersection problems. In *Proceedings of the 17th Annual Symposium on Foundations of Computer Science*, pages 208–215, 1976. 78, 96

M. S. Shephard, J. E. Flaherty, K. E. Jansen, X. Li, X. Luo, N. Chevaugeon, J. F. Remacle, M. W. Beall, and R. M. O'Bara. Adaptive mesh generation for curved domains. *Applied Numerical Mathematics*, 52 (2-3):251–271, 2005. doi: 10.1016/j.apnum.2004.08.040. 34

J. R. Shewchuk. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied Computational Geometry: Towards Geometric Engineering*, pages 203–222. Springer, 1996. doi: 10.1007/BFb0014474. 65, 74

J. R. Shewchuk. What is a good linear element? Interpolation, conditioning, and quality measures. In *Proceedings of the 11th International Meshing Roundtable*, pages 115–126, Ithaca, New York, 2002a. 17

J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22 (1-3):21–74, 2002b. doi: 10.1016/S0925-7721(01)00047-5. 74

H. Si and K. Gärtner. Meshing piecewise linear complexes by constrained Delaunay tetrahedralizations. In B. W. Hanks, editor, *Proceedings of the 14th International Meshing Roundtable*, pages 147–163, San Diego, California, 2005. Springer. doi: 10.1007/3-540-29090-7_9. 105

R. B. Simpson. Anisotropic mesh transformations and optimal error control. *Applied Numerical Mathematics*, 14(1-3):183–198, 1994. doi: 10.1016/0168-9274(94)90025-6. 11

J. Slingo, K. Bates, N. Nikiforakis, M. D. Piggott, M. Roberts, L. Shaffrey, I. Stevens, P. Vidale, and H. Weller. Developing the next generation climate system models: challenges and achievements. *Philosophical Transactions of the Royal Society A*, 367(1890):815–831, 2009. doi: 10.1098/rsta.2008.0207. 69

P. K. Smolarkiewicz and P. J. Rasch. Monotone advection on the sphere: an Eulerian versus semi-Lagrangian approach. *Journal of the Atmospheric Sciences*, 48(6):793–810, 1991. doi: 10.1175/1520-0469(1991)048<0793:MAOTSA>2.0.CO;2. 42, 43

A. Staniforth and J. Côté. Semi-Lagrangian integration schemes for atmospheric models: a review. *Monthly Weather Review*, 119(9):2206–2223, 1991. doi: 10.1175/1520-0493(1991)119<2206:SLISFA>2.0.CO;2. 42

W. G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Series in Automatic Computation. Prentice-Hall, 1973. 33

I. E. Sutherland and G. W. Hodgman. Reentrant polygon clipping. *Communications of the ACM*, 17(1):32–42, 1974. doi: 10.1145/360767.360802. 78, 79

P. K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis*, 21(5):995–1011, 1984. doi: 10.1137/0721062. 86

B. Szabó, A. Düster, and E. Rank. The p-version of the finite element method. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 1, pages 119–139. Wiley, 2004. 34

T

L. L. Takacs. Effects of using a posteriori methods for the conservation of integral invariants. *Monthly Weather Review*, 116(3):525–545, 1988. doi: 10.1175/1520-0493(1988)116<0525:EOUAPM>2.0.CO;2. 49

A. Tam, D. Ait-Ali-Yahia, M. P. Robichaud, M. Moore, V. Kozel, and W. G. Habashi. Anisotropic mesh adaptation for 3D flows on structured and unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 189(4):1205–1230, 2000. doi: 10.1016/S0045-7825(99)00374-6. 14

Y. Tanizume, H. Yamashita, and E. Nakamae. Tetrahedral elements generation using topological mapping and spacedividing for 3-D magnetic field FEM. *IEEE Transactions on Magnetics*, 26(2):775–778, 1990. doi: 10.1109/20.106432. 88

J. F. Thompson, B. Soni, and N. P. Weatherill. *Handbook of grid generation*. CRC press, 1999. 9

J. Thuburn. Some conservation issues for the dynamical cores of NWP and climate models. *Journal of Computational Physics*, 227(7):3715–3730, 2008. doi: 10.1016/j.jcp.2006.08.016. 40, 49

L. N. Trefethen. Predictions for scientific computing fifty years from now. *Mathematics Today*, 36(2):53–57, 2000. 3

L. N. Trefethen. Numerical analysis. In T. Gowers, editor, *Princeton Companion to Mathematics*, pages 604–615. Princeton University Press, 2008. Section IV.21. 3

U. Tremel, K. A. Sorensen, S. Hitzel, H. Rieger, O. Hassan, and N. P. Weatherill. Parallel remeshing of unstructured volume grids for CFD applications. *International Journal for Numerical Methods in Fluids*, 53(8):1361, 2007. doi: 10.1002/fld.1195. 27

V

M.-G. Vallet. Génération de maillages anisotropes adaptés – application à la capture de couches limites. Technical Report RR-1360, INRIA Rocquencourt, Rocquencourt, Le Chesnay, France, 1990. url: `http://www.inria.fr/RRRT/RR-1360.html`. 11, 114

M.-G. Vallet, C.-M. Manole, J. Dompierre, S. Dufour, and F. Guibault. Numerical comparison of some Hessian recovery techniques. *International Journal for Numerical Methods in Engineering*, 72(8):987–1007, 2007. doi: 10.1002/nme.2036. 22, 67

E. H. van Brummelen. Mesh association by projection along smoothed-normal-vector fields: Association of closed manifolds. *International Journal for Numerical Methods in Engineering*, 73(4):493–520, 2008. doi: 10.1002/nme.2085. 32

Y. Vasilevskii and K. Lipnikov. An adaptive algorithm for quasioptimal mesh generation. *Computational Mathematics and Mathematical Physics*, 39(9):1468–1486, 1999. 13, 14, 68, 86, 117, 126, 127, 128

D. A. Venditti. *Grid adaptation for functional outputs of compressible flow simulations*. PhD thesis, Massachusetts Institute of Technology, 2002. 21

D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, 187:22–46, 2003. doi: 10.1016/S0021-9991(03)00074-3. 21

R. Verfürth. *A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. John Wiley and Sons Ltd., 1996. ISBN 0-471-96795-5. 5

W

N. P. Weatherill and O. Hassan. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering*, 37(12):2005–2039, 1994. doi: 10.1002/nme.1620371203. 27

C. R. Wilson. *Modelling multiple-material flows on adaptive unstructured meshes.* PhD thesis, Imperial College London, London, UK, 2009. 68, 86, 92

R. D. Wordsworth, P. L. Read, and Y. H. Yamazaki. Turbulence, waves, and jets in a differentially heated rotating annulus experiment. *Physics of Fluids*, 20(12):126602, 2008. doi: 10.1063/1.2990042. 119

Z

Z. Zhang and J. Zhu. Analysis of the superconvergent patch recovery technique and a posteriori error estimator in the finite element method (I). *Computer Methods in Applied Mechanics and Engineering*, 123(1-4):173–187, 1995. doi: 10.1016/0045-7825(95)00780-5. 23

J. Z. Zhu and O. C. Zienkiewicz. A posteriori error estimation and three-dimensional automatic mesh generation. *Finite Elements in Analysis & Design*, 25(1-2):167–184, 1997. doi: 10.1016/S0168-874X(96)00037-6. 11, 23

O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method. Volume I: The Basis.* Butterworth-Heinemann, fifth edition, 2000a. 8, 60, 113

O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method. Volume III: Fluid Dynamics.* Butterworth-Heinemann, fifth edition, 2000b. 62

O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part I: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7): 1331–1364, 1992. doi: 10.1002/nme.1620330702. 23