Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE @ DIRECT®

ELSEVIER

Journal of Computational and Applied Mathematics 166 (2004) 55–86

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Some aspects of adaptive grid technology related to boundary and interior layers

Graham F. Carey*, M. Anderson, B. Carnes, B. Kirk

ICES, The University of Texas, Austin, TX 78712, USA

Received 18 December 2002; received in revised form 16 June 2003

Abstract

We consider the use of adaptive mesh strategies for solution of problems exhibiting boundary and interior layer solutions. As the presence of these layer structures suggests, reliable and accurate solution of this class of problems using finite difference, finite volume or finite element schemes requires grading the mesh into the layers and due attention to the associated algorithms. When the nature and structure of the layer is known, mesh grading can be achieved during the grid generation by specifying an appropriate grading function. However, in many applications the location and nature of the layer behavior is not known in advance. Consequently, adaptive mesh techniques that employ feedback from intermediate grid solutions are an appealing approach. In this paper, we provide a brief overview of the main adaptive grid strategies in the context of problems with layers. Associated error indicators that guide the refinement feedback control/grid optimization process are also covered and there is a brief commentary on the supporting data structure requirements. Some current issues concerning the use of stabilization in conjunction with adaptive mesh refinement (AMR), the question of "pollution effects" in computation of local error indicators, the influence of nonlinearities and the design of meshes for targeted optimization of specific quantities are considered. The application of AMR for layer problems is illustrated by means of case studies from semiconductor device transport (drift diffusion), nonlinear reaction–diffusion, layers due to surface capillary effects, and shockwaves in compressible gas dynamics.

Keywords: Adaptive mesh refinement; Boundary layers; Internal layers; Multiscale behavior

1. Introduction

Boundary and interior layer structures in the solution are a familiar "feature" of certain classes of applications in engineering and science. Representative examples are the viscous boundary layer

* Corresponding author.
E-mail address: carey@cfdlab.ae.utexas.edu (G.F. Carey).

doi:10.1016/j.cam.2003.09.036

adjacent to a wall, edge effects near the boundary of a uniformly loaded shell structure, capillary meniscus boundary layers in the contact region of a thin film and a container, thermal boundary layers in heated fluids, interior shock layers in compressible gas dynamics, interior impurity layers in dopant diffusion for semiconductor implant processing, interior solution layers in the carrier concentration and field for semiconductor devices, reaction layers in heat and mass transfer, moving melt and reaction layers, layers in coupled physics problems such as electro-rheological and suspension flows, and so on. These layer structures may be thought of as local or embedded and their presence interpreted as a consequence of competing physical or chemical processes that dominate over different length scales on different regions of the problem domain.

For example, in the case of flow of a viscous fluid past a wall, the fluid adheres to the wall so the local effects of the boundary condition and viscosity are significant. For a uniform stream with kinematics viscosity v and velocity magnitude U past a flat plate of length l, there will be a thin viscous boundary layer adjacent to the plate if the Reynolds number $Ul/v \gg 1$ and an exterior region where the flow can be treated as essentially inviscid. Then viscous effects dominate at a small length scale in a layer adjacent to the boundary. The vorticity generated by the boundary condition varies rapidly as it is convected and diffused through the thin layer with the vorticity zero or changing slowly in the region exterior to the layer. Similarly, in the case of heat and mass transfer problems where convection and diffusion arise, these respective effects may compete to generate layer structures where diffusion dominates, and an exterior or outer region where convection dominates. In the electro-rheological case, the electric field changes the effective viscosity so that viscous boundary layers may be accentuated and in meniscus problems the interfacial effects associated with surface tension and curvature at the boundary compete with gravity. The interior layers in the device electron transport equation are a consequence of the electric field and the dopant impurity layer structures. Catalytic reactions in a domain can generate concentration layers at a boundary for large reaction rates. For high-speed compressible flows, shock layers form and thermal and concentration layers for the dissociating gas may also arise.

These multiscale ideas enhance our understanding of the complex interacting effects. Moreover, they can be exploited in developing simpler physical models that make the resulting problem more amenable to mathematical solution. This may involve "inspectional scaling analysis" to determine what effects dominate and which can be neglected from the physical model. Such reasoning was applied in [55] in developing the boundary layer equations describing flow adjacent to a wall. More formally, one can introduce the important space and time scales in a scaling of the governing equations to obtain nondimensional equations with parameters that can be used to characterize regions of interest (high or low Mach number, Rayleigh number and so on). In turn, these parameters often enter as the expansion parameters in regular or singular perturbation expansion schemes to yield simpler mathematical models. In particular, the use of singular perturbation techniques involves an expansion using a stretched variable in the inner layer region with another expansion at the normal scale in the outer region. A matched asymptotic "fit" is employed at the "interface". Each expansion involves use of the natural spatial scale for the associated region (e.g., stretched scales in the layers). Perturbation techniques and concepts from matched asymptotic analysis also contribute to our understanding of these interactions and, when analytic solutions can be composed in this way, they add significantly to our knowledge of the layer structure and properties. They also help explain why numerical approaches must be applied with due care, indicating why some methods fail and others succeed. In the present context, they provide insight into the question of constructing successful adaptive mesh strategies. For example, we can first solve the (simplified) boundary layer equations separately for the layer solution and then solve the (simplified) inviscid equations in the exterior region.

The above ideas can be viewed as an illustration of a domain decomposition strategy with different simplified physics in respective subdomains, such as the inner diffusive layer and outer convective regions, respectively. Such decompositions are frequently exploited in engineering analysis and design. The above examples can also be interpreted from the standpoint of multiple scales with different physics/chemistry active on the respective scales associated with these inner and outer subregions and with the subregion solutions matched in some sense with respect to these spatial scales. This idea underlies mathematical approaches using multiple scales and matched asymptotics in singular perturbation theory [27].

Both the subdomain and multiscale concepts are of great importance in developing effective numerical approaches to address these types of problems. Of central importance here is the idea of different physics dominating at different scales on subregions with appropriate “matching” in an interface zone. Therefore, these properties should be respected in the associated discretization strategies and solution algorithms. The focus of the present study is the role of adaptive mesh approaches in treating these layer problems. At the simplest level, we may elect to separate the space and time domains into subdomains corresponding to the regions where the different scales preside. We term such an approach a domain decomposition and can build and analyze discrete models constructed on the respective subdomains. For example, if the domain is simple then one can use a uniform fine grid in the inner layer and a uniform coarse grid in the exterior region [31]. Alternatively, one can use an optimization approach to design meshes that are graded appropriately into the layer regions [48,54] and adaptively timestep to resolve similar transient scale and accuracy issues. Adaptation of the spatial mesh will be the focus of the present study.

One of the great promises of local mesh refinement is optimization and control of the grid to tailor it to the problem being solved. The main approaches being developed along these lines involve adaptive refinement of cells in the grid using point insertion and subdivision or adaptive mesh redistribution. Pearson [50] was the first to explore local adaptive refinement for problems of boundary and interior layer type using a finite difference scheme and Carey [16] was apparently the first to investigate adaptive finite element mesh refinement approaches in 2D. In the earlier studies by Pearson, several problems exhibiting a variety of different layer behaviors were considered. The adaptive scheme involved adding points with an error indicator based on changes in the solution computed at the prior step. The 2D finite element work in [16] and related 1D work in [22] used local residuals to guide refinement.

Since adaptive mesh refinement (AMR) and coarsening via de-refinement (AMR/C) are logical strategies that should sensibly be part of the engineering analysis algorithm, then this is the preferred approach. It also provides a convenient data structure for accelerating the solver algorithm, “mining” simulation results and for fast remote visualization. Hence, it is clear that AMR/C should be intrinsic to the simulation goal and intimately tied to the analysis. Such strategies are also clearly needed to treat problems where boundary and interior layers arise in the solution because the inherent multiscale nature of these layer problems implies a similar scaling of the mesh structure. The location, nature and extent of the layer may not be known a priori so an adaptive feedback control strategy for the mesh and simulation will be essential to computing accurate and reliable results for such problems.

Our goal in this work is to describe the main concepts and issues related to AMR for such boundary and interior layer problems. We begin with transport systems and an example illustrating conditions on cell size/scale for stable, nonoscillating solutions. The use of local error estimates and error indicators provides a basis for feedback control to identify layer regions where refinement is needed. Related issues such as pollution control in error indicator computation, targeted or goal-oriented adaptive schemes, algorithms, and illustrative examples are considered. Mapping, mesh grading, grid redistribution and moving grid concepts are introduced. The need to maintain cell quality and permit enrichment leads to adaptive mesh refinement as an effective general approach as illustrated in the case studies.

2. Transport systems

To illustrate the issues for layer problems let us briefly consider the well-known transport equation

$$\frac{\partial u}{\partial t} + \alpha \cdot \nabla u - \nabla \cdot (\beta \nabla u) = f, \quad x \in \Omega,\ t > 0 \tag{1}$$

for a scalar field u such as temperature or species concentration in heat and mass transfer, respectively. Similar equations arise as systems for transport of vorticity in the stream-function vorticity equations or mass, momentum and energy in the Navier–Stokes equations, and electron and hole carrier concentrations in the semiconductor device transport equations.

For the stationary transport case, (1) simplifies accordingly to

$$\alpha \cdot \nabla u - \nabla \cdot (\beta \nabla u) = f, \quad x \in \Omega \tag{2}$$

in domain Ω, where α is the convective coefficient, β is the diffusion and f is the source term. For example, α is the velocity in heat or mass transfer and Navier–Stokes problems and it is the electric field in the device transport equation; β is the thermal conductivity, diffusivity, kinematic viscosity and mobility, respectively, in these applications; and f corresponds to the heat source, mass source, body force and collision recombination terms, respectively. In practice, α, β, and f may be functions of position, the field variable u, its derivatives, and of other field variables. In the case studies presented later, we have examples where $\alpha = \alpha(u)$, $f = f(u)$ and $\beta = \beta(|\nabla u|)$.

The "competition" between the respective convection, diffusion and reaction terms in (1) and (2) is of particular interest since it may lead to layer structures in the solution. To explore the nature of this interaction and the associated numerical approximation difficulties for layer problems, let us simplify (1) further by first considering the one-dimensional case with coefficients α, β positive constants

$$\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} - \beta \frac{\partial^2 u}{\partial x^2} = f, \quad 0 < x < 1,\ t > 0. \tag{3}$$

The source term f can obviously have a spatial scale structure where f is large and strongly varying in layer regions and more uniform elsewhere. This will usually induce a related multiscale behavior in u since obviously the governing equation defines a balance relation between various derivatives of u on the left and f on the right. Similarly, if f is zero, the equation implies a balance relation between the time rate of change of u and the spatial derivatives which can be interpreted in this context as an "equivalent" source term. Again, the local behavior of these spatial derivatives and their competition will influence the evolution of u in time and the same will be true of the approximate problem obtained after discretization.

If $\beta = 0$ in (3), the equation is hyperbolic and implies that the material derivative of u is equal to f. That is, $u_\xi = f$ and the evolution of u along the characteristic lines, $\xi = x - \alpha t$ (of slope $1/\alpha$ in the x, t plane) is determined by the behavior of f on the characteristics. If α is also zero the equation reduces further to an ordinary differential equation with respect to t at each point x (vertical characteristics). It is clear from these cases that behavior in time such as rapid transient layers, oscillations, decay and growth, depends on the form and sign of f.

If instead we let $\alpha \to 0$ with $\beta > 0$ fixed then the problem type is parabolic and the limiting equation is pure diffusion, so sharp layers and fronts will be diffused. If both α and β are nonzero, then these respective convective and diffusive behaviors may "compete" to generate layers for appropriate ranges of these parameters and choice of boundary conditions.

Since our focus here is on spatial layer behavior, let us consider in more detail the steady state problem for (3) with $f = 0$. Introducing the prime notation to denote differentiation, we have

$$\alpha u' - \beta u'' = 0, \quad 0 < x < 1, \tag{4}$$

which clearly states that the exact solution satisfies the balance $\alpha u' = \beta u''$ between convection and diffusion for any x in $(0, 1)$. This implies that as $\beta \to 0$ for fixed α either u is constant or u'' is locally large. For layer behavior here, we are clearly interested in the case $\alpha \gg \beta$ and a standard model problem for testing numerical discretization schemes is (4) with $u(0) = 0$ as the "inflow" condition and $u(1) = 1$ as the "hard outflow" condition. The analytic solution is monotone increasing, with $u \sim 0$ in most of the domain and rising to $u = 1$ through a boundary layer adjacent to $x = 1$. Recalling (3) and our discussion of the degenerate case $\beta = 0$, convection attempts to propagate the end condition $u = 0$ at $x = 0$ through the domain but fails to match the "redundant" condition at $x = 1$. For $\beta \neq 0$ but $\alpha \gg \beta$, convection will still dominate in the outer region $(0, 1-\delta)$, $0 < \delta \ll 1$, but the boundary condition at $x = 1$ is active and diffusion will be important in the layer $(1-\delta, 1)$.

If a standard central difference scheme or a Galerkin finite element scheme with linear elements is applied to (4), we have at interior node i of a uniform mesh of cell size h, simply

$$-\left(\frac{\alpha}{2h} + \frac{\beta}{h^2}\right) u_{i-1} + \frac{2\beta}{h^2} u_i + \left(\frac{\alpha}{2h} - \frac{\beta}{h^2}\right) u_{i+1} = 0. \tag{5}$$

It is easy to verify that this difference scheme is not monotone and will support an oscillation across the pair of cells adjacent to interior node i if the cell condition $\alpha h/\beta < 2$ is not satisfied. Note that $\alpha h/\beta$ corresponds to a cell form of the familiar Reynolds number for flows or Peclet number for mass transport applications. When this condition holds, system (5) becomes diagonally dominant and this matrix property is relevant to both the behavior of the solution and to the performance of solution algorithms. For $f = f(x)$ nonzero, we have f_i on the right in (5) and for $f = g(x) + \gamma u$ with γ constant we have g_i on the right and the diagonal contribution, $-\gamma u_i$, on the left in (4). Clearly, the nature of the reaction term in this latter case influences the monotone structure of the algebraic operator and therefore the oscillatory behavior. Similar reasoning applies to the Jacobian systems that arise in solving corresponding nonlinear problems.

Returning to the homogeneous case in (4) and (5), the cell condition implies mesh restriction $h < 2\beta/\alpha$ for a stable nonoscillatory approximation. That is, there is a mesh scale restriction that depends on the ratio β/α of the diffusion and convection coefficients. If we scale (4) by $1/\alpha$ we obtain $-(\beta/\alpha)u'' + u' = 0$ with $\beta/\alpha \ll 1$ clearly the small parameter of a singular perturbation problem. Setting $\varepsilon = \beta/\alpha$, the cell condition is equivalent to requiring a uniform mesh with $h < 2\varepsilon$ to resolve the layer scale. For computations with $h > 2\varepsilon$ the numerical solution is oscillatory and the oscillations become larger as h is increased. For $h < 2\varepsilon$ the approximation is monotone and converges with second order rate as $h \to 0$. Note that these results apply for *uniform* grids.

Computations with uniform fine grids that resolve the layer scale will give the desired result but are not practical for complex nonlinear applications in higher dimensions. Clearly, the mesh scale needed to resolve the inner layer is not needed to approximate the outer solution, so one obvious strategy would be to use a uniform fine mesh in the layer and a uniform coarse mesh in the outer region. However, this would create a more complicated mesh transition problem in higher dimensions (especially when the geometry is not simple) as well as possible associated numerical problems such as ill conditioning. The matched asymptotics approach suggests grading the mesh to match effects across the transition to the layer. Hence, the idea of adaptively redistributing a mesh or adaptively refining is natural for layer problems.

In the matched asymptotics approach for (4), a stretching transformation could be applied in the layer region of the form $\xi = (1 - x)/\varepsilon^{\gamma}$ where γ is related to the strength of the layer. As $1 - x \to \delta \sim \varepsilon^{\gamma}$, $\xi \to 1$. That is, the stretching transformation maps the layer to a stretched reference region of the same order as the outer region, emphasizing that both scales are equally relevant. The analytic solution to our example problem is $u(x) = (\mathrm{e}^{x/\varepsilon} - 1)/(\mathrm{e}^{1/\varepsilon} - 1)$ and for $\gamma = 1$, $\xi = (1 - x)/\varepsilon$, so we have $u \sim \mathrm{e}^{-\xi}$ in the stretched layer. This implies that the inner solution drops exponentially from 1 at $\xi = 0$ (or $x = 1$) to $1/e$ at $\xi = 1$, that is, at the layer edge ($x = 1 - \varepsilon$).

When we know the qualitative exponential behavior in the layer, we can use this behavior or a reasonable approximation of it to construct a mesh grading function for the discretized problem in the layer region. Hence, mapping and mesh grading functions provide one approach for resolving the layer scale. More generally, the layer location and nature may not be adequately known in advance. Therefore, a better strategy is to progressively refine a grid using a posteriori error indicators to guide the adaptive process in the layer. We elaborate on these approaches later.

3. Stabilization effects

The above observations notwithstanding, this recommended AMR/C approach is not yet the usual practice in numerical simulation. Instead of directly addressing the inadequate resolution of the layer by redistribution or refinement, work has focused instead on “stabilizing” the scheme to remove oscillations on coarse grids. This has been partly motivated by the hyperbolic nature of the degenerate problem ($\beta \to 0$) for which central differencing is unstable and single-cell upstream or upwind difference approximation of the convective term will yield a stable method. For $\beta \neq 0$, $\alpha \gg \beta$ this upwind scheme remains stable and the difference system is diagonally dominant. However, a simple truncation error analysis reveals that the resulting discrete model is only first-order accurate asymptotically. Furthermore, the leading truncation error term is dissipative. That is, this stabilization is achieved at the cost of a suboptimal rate of convergence and addition of significant artificial dissipation which degrades the approximation of the layer.

In fact, the above upwind stabilization scheme is equivalent to a second-order accurate approximation of a problem with diffusivity $\varepsilon + (h/2)$. Note that for $h > 2\varepsilon$ the added artificial diffusion exceeds the true diffusion so the layer approximation is exceedingly poor. If h is refined, the layer is resolved but the scheme is still of lower-order. Given that the interesting and important physical behavior is usually in the layer region, the preoccupation with low-order dissipative stabilizing schemes in most of the past two decades for these types of layer problems is baffling. Of course, if one is interested in accelerating an algorithm then this is a useful strategy on coarse and intermediate grids but this has been rarely the motivation. Clearly, the use of higher-order stabilized schemes is desirable and there has been some work on this subject. Nevertheless, much of the stabilization used in practical applications has been of the lower-order dissipative form. Of course there is a natural fit when these stabilization schemes are applied in conjunction with AMR.

The "success" of stabilization schemes in finite difference analysis has motivated construction of similar upwind weighted Petrov–Galerkin methods in finite element approximation of these problems. For example, the Petrov–Galerkin form of (4) follows from a weighted residual manipulation and integration by parts as: find $u_h \in H_h$ satisfying $u_h(0) = 0,\ u_h(1) = 1$ and such that

$$\int_0^1 (\varepsilon u_h' v_h' + u_h' v_h)\,\mathrm{d}x = 0 \tag{6}$$

for all admissible test functions $v_h \in W_h$ where H_h and W_h are the approximate solution (trial) and weight (test) spaces, respectively. The previous backward difference model can be duplicated by selecting u_h to be the piecewise linear basis $\phi_i(x)$ and v_h to be the sum of a piecewise linear basis function and a pair of quadratic "bubbles", $b(x)$ and $-b(x)$, on the elements upwind and downwind, respectively, from node i.

Other stabilization schemes based on manipulating the time truncation error in the transient problem using Lax–Wendroff/Taylor–Galerkin ideas have been developed. Least-squares finite elements are naturally dissipative and, motivated by this, least-squares residuals have been added to Galerkin formulations to construct Galerkin least-squares (GLS) approaches. There has been an effort to improve these approaches using certain so-called higher-order compact schemes [62]. Another approach is to introduce local approximations to the differential equation to develop improved difference schemes in the layer. For example, the Scharfetter–Gummel difference scheme for the semiconductor "drift-diffusion" transport equations uses a local exponential approximation to obtain a stabilized difference scheme. Some of the recent finite element work on "bubble functions" or use of an approximate local Green's function for stabilization is related to this idea. Finally, stabilized coarse grid finite difference and finite element schemes can be constructed by static condensation of fine grid nodes in the one-dimensional case (although this does not extend cleanly to higher dimensions) or by related residual bubble schemes.

In concluding this discussion of stabilization, it is worth noting that one can scale (4) by the integrating factor $\mathrm{e}^{-x/\varepsilon}$ and rewrite the resulting equation in conservative (self-adjoint) form as $-(\varepsilon \mathrm{e}^{-x/\varepsilon} u')' = 0$. Finite differencing this equation or a standard Galerkin formulation with linear elements yield similar three-point difference schemes that are nonoscillatory for all h and that exhibit exceptional accuracy and superconvergence of the approximation at the nodes.

4. Error indicators

Feature indicators such as pressure gradients in compressible flow have been widely used in practice to locate shock layers and grade the mesh locally. However, a more formal error analysis that leads to a posteriori error estimates and computable local error indicators is a preferable approach (e.g., see [2,5,7,18,65] and the references cited therein) Most such error indicator work has been directed to grading the mesh so that the error is equidistributed over the domain and is uniformly small. There are three main approaches: (1) truncation error or residual error indicators; (2) local problem solution; and (3) recovery indicators. The first and second classes are explicitly related to the error in locally approximating the governing differential equation in question. In the third class, local higher-order approximation of say, a gradient is used to determine an error indicator [70]. This latter approach may be based on some form of superconvergence behavior, smoothing, or extrapolation and need not be explicitly tied to the physical problem and governing equations [13,67].

In problems with layers, the quantity of interest may be a gradient such as the flux or stress at the wall adjacent to a boundary layer or may be the amplitude of the solution in an interior layer. The previous error equidistribution strategies may then be inefficient since they are designed to produce an error that is globally small. Instead, the adaptive refinement scheme may be modified to focus on the layer quantity while ensuring the mesh elsewhere is adequate to prevent pollution of the solution in the layer. We discuss this later and outline a new algorithm. First, however, let us illustrate the main ideas using a residual indicator approach.

For a linear differential equation $Lu = f$ and difference scheme with solution u_i, the error, $e_i = u(x_i) - u_h(x_i)$, at grid point i satisfies $L_h e_i = -\tau_i$ where L_h is the discretized operator and τ_i is the truncation error. Similarly, if u_h is the finite element approximation and $r = Lu_h - f$ is the residual, the error $e = u - u_h$ satisfies $Le = -r$. The local error is thereby related to the truncation error or residual. Hence, local patch post-processing schemes can be devised to compute truncation or residual error indicators. Locally refining the mesh to systematically reduce and equidistribute the residual indicator will control the error. To illustrate this point, consider the two-point problem in $0 < x < 1$

$$-\{[\alpha^{-1} + \alpha(x-\bar{x})^2]u'\}' = 2 - 2\alpha(x-\bar{x})[\tan^{-1}\alpha(x-\bar{x}) + \tan^{-1}\alpha\bar{x}]. \tag{7}$$

The exact solution with $u(0) = u(1) = 0$ is

$$u(x) = (1-x)[\tan^{-1}\alpha(x-\bar{x}) + \tan^{-1}\alpha\bar{x}], \tag{8}$$

where $\bar{x}$ determines the location of an interior layer and α controls the layer strength. The solution with $\bar{x} = 0.36$ and $\alpha = 100$ is shown in Fig. 1. After nine adaptive refinement steps from an initial uniform mesh of four quartic elements, we obtain a graded mesh of 25 elements with a graduated transition from either side into the layer and 12 elements in $(0.3125, 0.375)$ across the layer. The element residuals are used as error indicators in this example. The L_2 norm of the global residual is reduced by more then 2 orders of magnitude by the refinement and the L_2 norm of the global solution error is 3.0×10^{-5} on the final grid with an accurately resolved solution inside and outside the layer [23].

Good local upper and lower bounds on the error in appropriate norms are needed to further calibrate, for instance, the local error in the solution to the local residual for the differential equations. Rather than simply post-process the approximate solution and rank element or patch residuals in the mesh, one can construct local element, patch or subdomain boundary value problems $Le = -r$ on ω^* where ω^* indicates the local subregion. Of course, pollution effects may corrupt the boundary data for the local problem so appropriate measures need to be taken to circumvent such difficulties. For example, oscillations due to the boundary layer for (3) persist across the domain and can induce

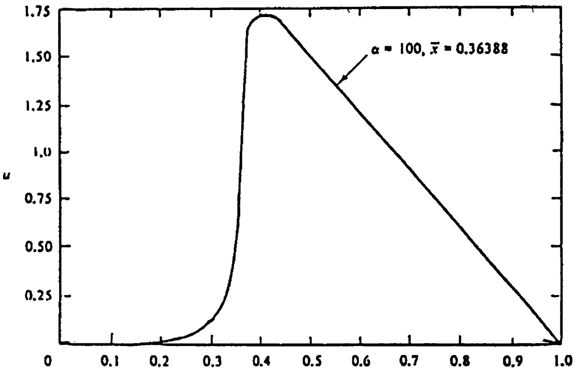

Fig. 1. Solution to Eq. (7) on an adaptively refined mesh illustrating the interior layer structure.

unnecessary refinement away from the layer. A subsequent "de-refinement" step will rectify this problem, but a preferable strategy may be to use the stabilization techniques to ensure a monotone approximation on the coarser grids. The dissipation will be reduced as refinement proceeds.

Indicators for boundary flux quantities such as local Nusselt number are of considerable practical interest in engineering analysis and design. Recovery indicators for boundary flux can be computed using the superconvergence properties of the Green–Gauss integral identities in association with the Galerkin finite element approximation as follows: Consider the two-dimensional stationary convection diffusion problem corresponding to (2) with source function f

$$-\nabla \cdot \beta \nabla u + \alpha \cdot \nabla u = f \quad \text{in } \Omega \tag{9}$$

and Dirichlet boundary condition $u = g$ on $\partial\Omega$.

The corresponding variational (Galerkin) problem is Find $u \in H^1(\Omega)$ with $u = g$ on $\partial\Omega$ and such that

$$\int_\Omega (\beta \nabla u \cdot \nabla v + \alpha \cdot \nabla u\, v)\, \mathrm{d}x = \int_\Omega f v\, \mathrm{d}x \tag{10}$$

for all admissible $v \in H_0^1(\Omega)$. The approximate problem follows on setting $u_h \in H^h \subset H^1(\Omega)$, $v_h \in H_0^h \subset H_0^1(\Omega)$ for u and v, respectively, so that

$$\int_\Omega (\beta \nabla u_h \cdot \nabla v_h + \alpha \cdot \nabla u_h\, v_h)\, \mathrm{d}x = \int_\Omega f\, v_h\, \mathrm{d}x. \tag{11}$$

Introducing the second Green–Gauss formula for the diffusion operator we have the identity

$$\int_\Omega \nabla \cdot (v \beta \nabla u)\, \mathrm{d}x = \int_\Omega \beta \nabla u \cdot \nabla v\, \mathrm{d}x + \int_\Omega \nabla \cdot (\beta \nabla u) v\, \mathrm{d}x, \tag{12}$$

so by Gauss' theorem in (12) and using (9)

$$\int_{\partial\Omega} \beta \nabla u \cdot \mathbf{n} v\, \mathrm{d}s = \int_\Omega \beta \nabla u \cdot \nabla v\, \mathrm{d}x - \int_\Omega f v\, \mathrm{d}x + \int_\Omega \alpha \cdot \nabla u\, v\, \mathrm{d}x. \tag{13}$$

Substituting u_h for u and v_h for v in the right of (13) we define the approximate projection σ^* accordingly by the relation

$$\int_{\partial\Omega_h} \sigma^* v_h \, \mathrm{d}s = \int_{\Omega_h} \beta \nabla u_h \cdot \nabla v_h \, \mathrm{d}x - \int_{\Omega_h} f \, v_h \, \mathrm{d}x + \int_{\Omega_h} \alpha \cdot \nabla u_h \, v_h \, \mathrm{d}x. \tag{14}$$

For $v_h = \phi_k$ with node k on the boundary, then v_h has support only on the strip S_h of elements adjacent to $\partial\Omega_h$ and expanding σ^* in a suitable basis on $\partial\Omega_h$, we have

$$\int_{\partial\Omega_h} \sigma_h^* \phi_k \, \mathrm{d}s = \int_{S_h} \beta \nabla u_h \cdot \nabla \, \phi_k \, \mathrm{d}x - \int_{S_h} f \, \phi_k \, \mathrm{d}x + \int_{S_h} \alpha \cdot \nabla u_h \, \phi_k \, \mathrm{d}x. \tag{15}$$

In particular, if we expand σ_h^* in the same basis as u_h and use Gauss–Lobatto quadrature on the left, we have an explicit formula for the nodal values for σ_k. The error indicator for the boundary flux follows as $\tilde{e}(x, y) = \sigma_h^*(x, y) - \sigma_h(x, y)$ where $\sigma_h(x, y)$ is $\beta \nabla u_h \cdot n$ evaluated on $\partial\Omega_h$.

5. Some practical issues

One-dimensional problems are special in the sense that element interfaces are knots whereas 2D elements share edges and in 3D they share edges and faces. Both simplex and tensor-product elements are commonly used in 2D and 3D. Let us first consider the 2D case. Given a triangulation of the domain, local refinement can be carried out by adding node points in the interior of an element and/or on edges and then subdividing the element. If points are added at the centroid of a triangle and this new node is connected to the vertices, the subdivision is local but the longest edge is not reduced and the interior subtriangles are slender. Delaunay edge swap operations will improve the cell quality. If, instead, the nodes are inserted at the midedges of a triangle and then joined, we obtain a quartet of congruent subtriangles, each similar to the parent triangle. Alternatively, the midpoint of the longest edge can be joined to the opposite vertex and to the midpoints of the other two edges [57]. A single diagonal swap separates these two configurations. In either case, the global mesh is now "nonconforming" since the new midedge nodes are shared with the three adjacent unrefined triangles. One can bisect these adjacent triangles by connecting midedge nodes to their opposite vertices or by similar propagating refinement strategies [52]. The end result is a conforming triangulation. Alternatively, one can simply refine designated elements to quartets of subelements where specified by an error indicator, and then enforce an interface constraint on any remaining "hanging" midedge nodes when solving the subsequent approximation problem.

A major issue in treating layers using adaptive grids is the need for anisotropic grids in dimension $d > 1$. For example, the grid for a boundary layer adjacent to a wall should be strongly graded in the direction normal to the wall. However, such grading may not be needed in the transverse directions, so the aspect ratio of the elements can be very large. If the usual cell subdivision strategies to quartets and octets are applied, then the mesh is refined both in the normal and transverse directions so the refinement scheme is inefficient and does not realize the full potential of adaptive gridding. If the general location of the layer is known in advance, then a practical grading to ameliorate this problem can be set up at the initial coarse mesh generation stage. Special-purpose mesh generators such as those used for exterior flow computations past airfoils include features for grading through the boundary layer. For example, slender quadrilateral and hexahedral elements in 2D and 3D, respectively, can be generated in the boundary layer region with the stretching aligned in the direction of the boundary and graded out to more uniform shape in the far field. Simplex elements can be used similarly.

Hybrid grids consisting of hexahedra, prisms and tetrahedra with transitional pyramids may be useful, particularly when the geometry and solution are complicated. For example, one may elect to use slender "flat" prisms with triangular faces of high aspect ratio in a grid region adjacent to a wing and to use tetrahedra in the far field. When standard adaptive refinement is applied to these "pregraded" meshes using local quadrisection or octasection of cells, the layer structure in the initial mesh is essentially preserved. It is clear, however, that this approach has several deficiencies: (1) it assumes specific a priori knowledge of the existence and location of a layer; (2) it places an additional burden on the mesh generator and mesh generation is a difficult problem in its own right; (3) it is less flexible and hence is more prone to lead to meshes containing cells that are ill-shaped with possible ill-conditioning and poor approximation properties.

Nevertheless, for applications where this approach can be used it is almost always superior for layer problems. General-purpose mesh generators, however, are usually designed to produce grids that have good cell quality for any reasonable geometry. Simply generating a valid hexahedral mesh to completion is a major challenge for complicated domains in industrial applications and tetrahedral meshes for the same problem will be completed but, in all likelihood, will contain slender tetrahedral "slivers" with large solid angles that are undesirable for simulations [29].

Delaunay swap algorithms can improve triangulations and mesh-smoothing algorithms are used both for hexahedral and tetrahedral mesh improvement. Clearly, these smoothing strategies will be detrimental in layer regions where suitable cells having high aspect ratio are really needed. A better approach is to generate a coarse mesh using the more flexible mesh generator to provide good cell quality and then adaptively refine in a way that will produce the desired efficient grading with high-aspect ratio cells in the layers. This implies that directional refinement will be enabled by permitting varied cell subdivision. That is, instead of refining a parent cell in 2D to a quartet of subcells, bisection to a pair of cells will also be permitted or quadrisection could be replaced completely by repeated bisection [59]. In turn, this implies that the error indicator should have a directional capability. A simple example would be a feature indicator that tested the curvature of the approximate solution along the directions of the principal axes of the element. This could be applied independently or in conjunction with another error indicator. Similar approaches can be applied to the element error obtained by solving a local boundary value problem or by the approach described in the next section that uses iterative enhancement to produce a more robust computable error estimate. If the error indicator is associated with an edge rather than an element, then a form of directional refinement may be naturally available since edges are directed. Subdivision of the adjacent cells follows subsequently and may lead to bisection, quadrisection or some other configuration depending on the underlying edge subdivision. The idea of directional subdivision can also be applied to p-type enrichment of the element in a similar manner. Beginning from a coarse mesh of bilinear elements, the tensor-product basis can be directionally enriched to a tensor-product of a quadratic basis function and a linear basis function based on the directional error indicator. This process can be continued to higher degree p. Of course, conformity requirements still apply across the edges between adjacent elements in standard formulations.

Discontinuous Galerkin schemes have become a popular research topic recently. Here the degrees of freedom are local to the element, so the global approximation is piecewise discontinuous. Continuity requirements are usually enforced weakly through element interface terms in an associated weak statement. On one hand, these schemes have an obvious drawback because the number of degrees of freedom are significantly increased and since the solver operations depend nonlinearly on the number of degrees of freedom, the computational cost increases very significantly indeed. On the other hand, since the approximation bases are completely local to the element, element-based adaptive refinement is slightly simpler in this setting. Furthermore, we can more conveniently handle mismatching meshes so adaptive DG schemes may be beneficial in transitioning from an inner layer subdomain to an outer subdomain. Abrupt transitions are to be avoided so this is an arguable attribute.

Most adaptive refinement schemes involve cell subdivision (h refinement). However, local accuracy can also be improved by increasing the degree p of the basis on an element (p refinement). Such p schemes work well when the local behavior is smooth [58]. If singularities are present then local h refinement toward the singularity can be combined with p refinement elsewhere. These are referred to as hp methods. The transition across the edge or face between elements of different degree can be handled in a manner analogous to the transition for the h refinement scheme. Although layers are abrupt and "visually" appear to have a singular nature, the matched asymptotics argument reveals that it is really a question of spatial scales. Nevertheless, the gradients and curvatures in the layer are large, and coarse meshes with standard p refinement will tend to generate oscillations unless the mesh size is sufficiently small.

In some cases, the qualitative local behavior of a singularity (and neighboring layer) or a boundary/interior layer may be known. This has been exploited in fracture mechanics by subtracting out the singularity or by constructing special bases for "singular elements". This approach could be generalized to include special bases for layer structures. One way to implement the idea is to use the standard finite element basis on a patch as a "cutoff function" to limit the extent of the layer function as follows: Assume for simplicity of exposition that the problem is one-dimensional and we have constructed a background coarse grid on which the location of the layer has been determined (using an indicator) or is known a priori: Let $s(x)$ describe the form of the layer solution and let grid point i be positioned near the approximate location of the layer. The piecewise-linear basis function $\phi_i(x)$ is zero on the boundary of the patch of elements adjacent to node i and therefore $\phi_i(x)$ is a suitable cutoff function. Accordingly, we define $s_i(x) = s(x)\phi_i(x)$ on the patch, with amplitude at node i to be determined. The approximation now has the standard piecewise polynomial behavior everywhere except over the elements adjacent to node i. The extension to higher dimensions for layer structures might involve a tensor-product of special function $s(x)$ with the constants or other functions. This idea may also be incorporated within the so-called meshless methods in a similar manner. While such approaches may produce a more efficient scheme for special purpose applications, it is debatable whether the additional complexity is needed if a good adaptive scheme and code are available.

In applications to equilibrium problems, the basic adaptive algorithms typically proceed as follows:

(i) an initial background grid is generated;
(ii) an approximate solution is computed;
(iii) local feature or error indicators are computed by post-processing the approximate solution on the current grid;
(iv) the grid is locally enriched by cell subdivision or degree enrichment.

Steps (ii)–(iv) are repeated until a desired error tolerance or stopping criteria are met. Such strategies are particularly useful for nonlinear problems since they offer an excellent strategy for a nested mesh continuation process that is usually very efficient since the solution on the previous mesh provides a good starting iterate for the next mesh. For a simple demonstration of this behavior on a nonlinear two-point boundary problem with an interior layer see [23].

The preceding algorithm is easily generalized to evolution problems by simply including local coarsening or "de-refinement" of the mesh in addition to local enrichment. For example, one can simply remove nodes in a triangulation and use a local Delaunay reconnection scheme or restore "parent" elements in a hierarchic quadtree or octree data structure to locally revert to a coarser grid. The latter strategy assumes a prior refinement but is well-suited to problems with propagating layers, where the grid is locally refined as the layer advances and then locally coarsened as the layer leaves cells. It is clear that in the refinement/coarsening algorithm the boundary approximation and other problem data are additional sources of "modeling error" that need to be quantified and controlled. This is not a new observation. Similar pollution problems arise when subdomain "windowing" or "rezoning" strategies are used to economize computations (see [17,25]).

Three-dimensional coupled nonlinear problems with layers are computationally intensive necessitating adaptive grids for efficiency of computation and economic memory usage. Parallel algorithms can be developed within this AMR setting to use processors assigned to subdomains and communication between processors at the subdomain boundaries. Layer problems obviously pose some special concerns for parallel computation: an initial grid may be partitioned to balance the computational and communication loads according to some metric [37,40] but subsequent refinement and coarsening to resolve the layer will lead to a strong imbalance across processors. Codes like Zoltan [36] are being developed to address dynamic partitioning needs but there are significant problems yet to be resolved. The problems are exacerbated when one considers coupled multiphysics applications on different domains that share an interior interface where refinement is taking place, as is the case in fluid/solid interaction problems. The question of appropriate metrics for determining a partition would appear to be inextricably tied to the nature of the problem being analyzed, the solution algorithm and the computer hardware. Approaches that are based exclusively on the grid geometry are of limited value.

Hence, in adaptive mesh treatment of problems with layer solution structure, the dynamic repartitioning code should have appropriate "rules" to guide the frequency of repartitioning and weights to guide the partitioning. This will require an additional layer of complexity for the partitioner to use the metrics for the parallel architecture in question. For multiphysics applications, layers may arise in each individual physics domain as well as at interfaces between domains. In this case, loosely decoupled algorithms with different adaptive grids graded into layers on the respective subdomains may be advantageous because of computational efficiency and the fact that the error indicators and refinement criteria can be tailored to the local physics behavior.

6. Error indicators and pollution control

As mentioned previously, the adaptive refinement process that grades the mesh into the layers is based on either a feature indicator such as a temperature gradient or an error indicator like a local truncation error estimate or local finite element residual. One can also solve a local boundary value problem to estimate the error corresponding to a local residual. There are obvious problems with using a feature indicator, so the main goal is to develop reliable and robust error indicators. One of the subtle problems here is the error indicator pollution issue that has been raised in recent years: in essence, the accuracy and reliability of the local error indicator will depend on the solution elsewhere to some extent and if singularities or layers nearby are not resolved adequately they will pollute the quality of the error indicator. For example, if a local boundary value problem is solved on an element or patch away from a boundary layer, oscillations due to the unresolved boundary layer will extend through the domain and corrupt the boundary data for the local error indicator calculations in the interior.

This may render an indicator calculation useless. Similar issues arise when one considers "targeted" or "goal-oriented" adaptivity that seeks to refine the mesh to meet a local error criterion at a point or in a specified subregion but deliberately permits the error elsewhere to be large rather than seek to equidistribute the error to be uniformly small. In this case, the pollution problem is particularly dangerous. One approach to address this is to construct an approximate Green's function or solve an auxiliary dual problem and thereby estimate pollution error (see [9,38,39,47,56]). However, it is clear that considerable additional work and coding may be required to accommodate these dual approaches. Hence, an open issue is the development of simple, efficient techniques that address possible pollution effects and remain computationally practical.

For problems where pollution is not an issue, the basic strategy outlined previously is appropriate and should be available as the main algorithm. However, when pollution is a concern then the following variation from the standard approach can be applied instead and fits more conveniently within the standard strategy [19]. The key idea is to carry out iterative enhancement on an intermediate global refinement to control the global polluting error effect and then to use this to compute a better error indicator for the local refinement. The scheme can be augmented by a multigrid V-cycle if desired. In this way, the global effect of polluting layers or singularities will not be "missed" by the algorithm. To all intents and purposes the scheme proceeds essentially as in the standard AMR algorithm. The main steps of an improved approach are as follows:

(i) Assume that an approximate solution $u^{(0)}$ has been computed for an elliptic boundary-value problem on initial mesh M_0 (M_0 may be unstructured). Make a uniform refinement to mesh M_1, interpolate or otherwise project the approximate solution $u^{(0)}$ to $\tilde{u}^{(1)}$ on M_1.

(ii) Do a few smoothing iterations on M_1. This is the basic step needed to compute the high-frequency error contributions on the finer grid [69]. A more complete multigrid scheme would include one or more subsequent V-cycles with restriction to the coarser grid and error prolongation back to the fine grid. Since the refinement to M_1 was uniform and the smoother is global, pollution is also being accommodated.

(iii) Let $\hat{u}^{(1)}$ be the approximate solution iterate so computed on M_1. Then the error $e = u - u^{(0)}$ on M_0 is approximated by $\hat{e} = \hat{u}^{(1)} - u^{(0)}$. This is a more robust error indicator for the scheme. Richardson extrapolation of $u^{(0)}$ and $\hat{u}^{(1)}$ may yield an improved result on elements away from singular points or layers. In some sense, the global smoother replaces the dual problem approach mentioned previously and appears computationally preferable.

(iv) Based on the computed error indicator in the previous step, refine M_0 locally by only subdividing those cells that meet the error criterion. The retained nodes (new and old) "inherit" the approximation from M_1.

The end result is a new adaptively refined mesh and the approach continues recursively as above. The approach also clearly applies to local polynomial enrichment schemes (p-type spectral refinement) and discontinuous Galerkin schemes.

The basic idea, as with other nested iteration schemes and MG schemes, is that the "exact" coarse grid solve is inexpensive and controls the low-frequency modes in the solution and error [12,14,15,28,43,45]. Subsequent smoothing iterations on the finer grids progressively capture the dominant high-frequency errors fast, while simultaneously providing increased accuracy on lower mode content [63]. The embedding of nested grids ensures that nodal values in the coarser meshes are repeatedly corrected at each smoothing iteration and the global smoothing step addresses the pollution issue.

7. Mesh redistribution and moving boundary problems

We conclude this part of the study with some brief comments on mesh redistribution for layer problems. Mesh redistribution is a standard strategy for improving the geometric quality of a grid without modifying the topology of the mesh and therefore not requiring manipulation of the data structure. This aspect was considered previously in the comments related to mesh smoothing. It also can obviously be designed to grade the mesh into interior and boundary layers. However, the geometric quality of the cells may then deteriorate as a result of this process to the point that accuracy of the underlying approximation is compromised or the calculations fail. In this case, the geometric properties of the mesh such as smoothness and orthogonality are often also considered together with local error indicators in controlling the grid. That is, the optimization objective function would also contain terms for grid quality and for error control. Another approach is to remove vertices or cells to maintain the cell quality as the mesh deforms.

Other more direct strategies are also possible. The grid can be redistributed by introducing a grading function $\xi(x)$ that depends on the solution. For example, a mesh could be graded to interpolate a monotone function with a layer such that the change in u is equidistributed on $[a,b]$ by introducing

$$\xi(x)=\int_a^x |u'|\,\mathrm{d}x \bigg/ \int_a^b |u'|\,\mathrm{d}x \tag{16}$$

and solving $\{\xi_i{=}i/N\}$ to compute $\{x_i\}$. In a similar way, we can seek $\xi(x)$ such that the error in some norm is minimized. The example $u(x)=-(x+\mu)^R+\mu^R(1-x)+(1+\mu)^R x$ with $\mu=10^{-2}$ and $R=-\frac{1}{4}$ has a boundary layer near $x=0$. The L^2 error obtained on (uniform versus redistributed) meshes of 4, 8, 16 and 32 linear elements are $(2.82\times10^{-1},\ 4.74\times10^{-2})$, $(1.48\times10^{-1},\ 1.18\times10^{-2})$, $(6.59\times10^{-2},\ 2.94\times10^{-3})$ and $(2.64\times10^{-2},\ 7.34\times10^{-4})$. More importantly, the layer approximation is superior in the redistributed mesh: The graded mesh on 32 elements has approximately half the elements in the interval $(0,0.1)$ and a rapid decrease in number of cells away from the layer. The same strategy can be extended to the solution of two-point boundary-value problems by simply using the current approximate solution u_h in the expression for the grading function. See [18,21] for more detailed discussion and associated algorithms.

The above ideas can be extended to problems that involve moving layers. Usually, the solution of the field problem and the mesh adjustment are decoupled at each timestep. A typical algorithm proceeds as follows: An initial mesh is first generated and adaptively refined or redistributed to best approximate the geometry and initial data. Next, the evolution problem is integrated through one or more timesteps using adaptive timestepping for error control. Periodically, during the integration process the mesh is moved (redistributed) based on the dynamics, a feature indicator or an error indicator. Finally, the solution is interpolated or projected to the new grid before integration proceeds. In the case of moving layer problems with sharp fronts, due caution should be exercised since the interpolation process may lead to dissipation errors that accumulate to degrade accuracy. Note that this standard approach also applies to moving boundary problems and to problems with moving interfaces such as the phase change boundary in Stefan problems, or a moving free surface between two fluids [10]. Moreover, there is again very little difference between this "moving mesh adjustment scheme" and one that is based exclusively on AMR/C. In the latter case, the algorithm proceeds in essentially the same way but the mesh adjustment is made by refining cells invaded by the advancing layer, while simultaneously coarsening cells being vacated by the layer. Level set methods [60] are also useful for treating free surface problems such as the moving fluid–gas interface.

A more unusual approach for propagating layers is to introduce a map to a moving frame and solve for both the map and the solution that minimize a specified objective function such as the L^2 residual of the transformed equation. In this approach, the mesh and solution are solved simultaneously at each timestep. Since grid point coordinates are unknowns and their location is advanced at each integration step, elements will usual deform and, in particular, grid points can "overtake" one another leading to tangled grids. Hence, interior barrier penalties are needed to maintain a valid (nontangled) grid, e.g., see [34,46]. In addition to the grid tangling problem, there are several other detractions to this approach. For instance, the formulation implies that a linear problem is transformed into a nonlinear problem and the number of solution variables is increased proportionally with the dimension so computational efficiency is a concern. More generally, the complexity of the approach and other complications such as treatment of complex geometry have discouraged practical use of these ideas but decoupled formulations can be devised that circumvent some of these difficulties.

Cell quality merits a few additional comments particularly in view of the situation with moving grids mentioned above. It is well known that triangles with large obtuse angles have poor approximation properties and that slender elements and rapid transitions in the mesh may also be associated with numerical conditioning problems in simulations. Hence, we expect there may be numerical difficulties arising here in treating boundary and interior layers.

Let us consider a patch of elements surrounding a single interior vertex node. Assume, for the moment, that we fix the boundary points of the patch and then seek to optimize the position of the interior vertex such that the objective function is to be minimized by moving the interior point alone. Setting the derivative of the objective function to zero yields a (usually nonlinear) algebraic equation that can be solved numerically using, for example, Newton's method with the constraint that the point remain inside the patch. Next, let us view this strategy as one step in a nonlinear Gauss–Seidel point relaxation scheme over the entire set of patches in the mesh. For each vertex node, there is an associated patch and these patches provide an overlapping patch coverage of the mesh. In precisely the same manner as the Gauss–Seidel point relaxation scheme for linear system solution, we can sweep over the vertices of the mesh and apply the previous local patch scheme successively to each vertex. (If an interior vertex is on a physical boundary then the vertex motion must be appropriately constrained to move on the boundary.) This defines an outer point relaxation iterative sweep with a constrained inner Newton scalar iteration for each patch. The outer sweeps continue until a convergence criterion or stopping tolerance is met. Successive over- and under-relaxation can also be applied [61,68]. Clearly, there are alternative descent schemes and solvers that can be applied to search for the minimizer. However, the above scheme is intuitively appealing and can be

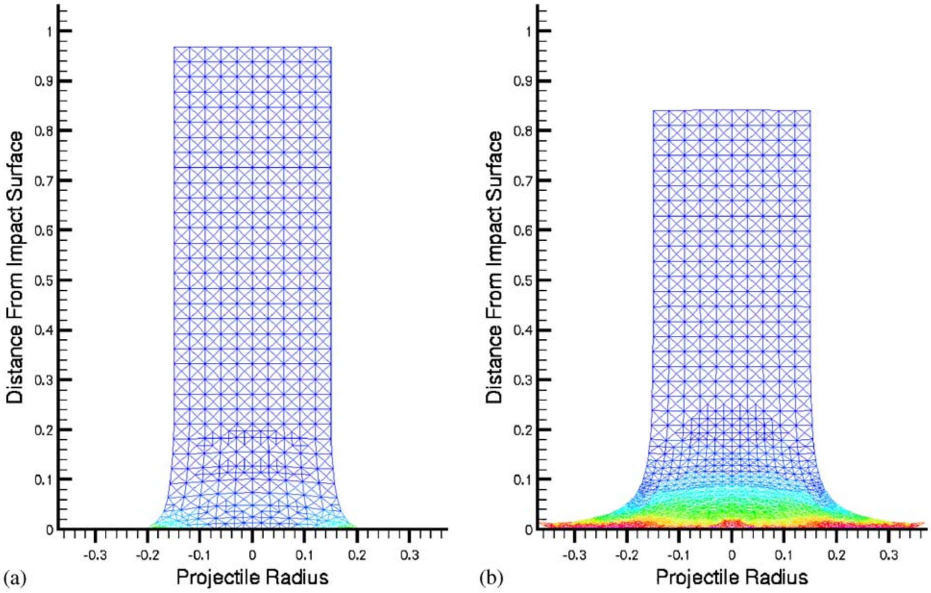

Fig. 2. Unstructured ALE for the Taylor Anvil problem. Grid at early (a) and late (b) stages in the computation, colored by plastic strain.

further specialized for problems exhibiting layer solutions that are directionally aligned (e.g., with a boundary).

Lagrangian formulations and related Lagrangian–Eulerian numerical approaches are utilized in certain classes of moving boundary problems such as impact problems involving large deformation and plastic flow. The elements deform and element shape quality will deteriorate locally to degrade accuracy. If explicit integration is used, the timestep will go to zero as any element in the mesh degenerates. Local grid refinement with coarsening is helpful in this case, but other strategies such as local smoothing of the grid or remeshing are also needed. The Taylor "anvil" benchmark problem for an elastic solid cylinder impacting a rigid boundary is a standard test problem in this application class. During impact, shockwaves traverse the cylinder and the cylinder and mesh deform significantly in the vicinity of the contact region as plastic flow and large local deformation occur. The effects on the mesh are indicated in Fig. 2 at two times during the process [48,49]. Even with the use of local refinement and element removal to improve degrading element shape quality, the mesh progressively deteriorates. Consequently, the timestep in this explicit calculation becomes prohibitively small and periodic mesh smoothing or remeshing is needed.

8. Case studies

In the following sections, we present AMR results for several applications involving solutions with layer structure.

8.1. Reaction–diffusion layers

The first example is for a boundary layer problem arising in a stationary reaction–diffusion problem

$$\begin{aligned}
&-\Delta u + \phi^2\, c(u) = f, \quad x \in \Omega,\\
&\frac{\partial u}{\partial n} = g_N, \quad s \in \Gamma_N,\\
&u = u_{\mathrm{D}}, \quad s \in \Gamma_{\mathrm{D}}, \qquad (17)
\end{aligned}$$

where the parameter ϕ^2, the Thiele modulus, is the ratio of the diffusion time scale to the reaction time scale. The general behavior for this type of problem is for a boundary layer to form along Γ_{D} when $\phi^2 \gg 1$, becoming more severe as $\phi^2 \to +\infty$. In the limit of $\phi^2 \to 0$, we obtain the diffusion solution, in which the boundary data on Γ_{D} has diffused to a steady-state solution.

In general, the reaction function $c(u)$ can be highly nonlinear. Because boundary layers are common to most forms of the reaction function, we will consider simply the linear case $c(u)=u$. Results for adaptive refinement for the nonlinear case can be found in [4,11,30,35,53].

A standard result for the linear case is that the error in the energy norm, defined by $\|v\|_E := (\int_\Omega |\nabla v|^2 + \phi^2 |v|^2 \,\mathrm{d}x)^{1/2}$, is equal to the norm of the residual of the differential equation, measured in the dual norm. Error indicators based on estimating the error in the energy norm, particularly in the singularly perturbed case of $\phi^2 \gg 1$, have been discussed in [1,66]. However, for nonlinear reaction functions c, this energy norm is not very useful, and we will be interested instead in the error in the equivalent $H^1(\Omega)$-seminorm, defined by $|v|_V := (\int_\Omega |\nabla v|^2 \,\mathrm{d}x)^{1/2}$, which is a natural norm on the space $V := \{v \mid v \in H^1(\Omega), \quad v|_{\Gamma_{\mathrm{D}}} = 0\}$.

The finite element solution u_h for problem (17) is computed using bilinear shape functions on quadrilateral elements in the standard way. The residual functional $R(u_h)$ is defined by

$$\langle R(u_h), v\rangle := \int_\Omega \{f\, v - \nabla u_h \cdot \nabla v - \phi^2 u_h v\} \,\mathrm{d}x + \int_{\partial\Omega} g_N v \,\mathrm{d}s, \quad v \in V. \qquad (18)$$

The global residual can be expressed as a sum of local residuals on each element K by the relation

$$\langle R(u_h), v\rangle := \sum_K \left\{ \int_K r_K v \,\mathrm{d}x + \int_{\partial K} R_K\, v \,\mathrm{d}s \right\}, \quad v \in V, \qquad (19)$$

where $r_K := f + \Delta u_h - \phi^2 u_h$ is the element residual and R_K is the edge jump residual defined on any edge $\gamma \subset \partial K$ with normal n_K by

$$R_K|_\gamma := \begin{cases} 0, & \gamma \in \partial K \cap \Gamma_{\mathrm{D}},\\ g_N - \nabla u_h \cdot n_K, & \gamma \in \partial K \cap \Gamma_N,\\ \frac{1}{2}\{\nabla u_h|_L - \nabla u_h|_K\} \cdot n_K, & \gamma \in \partial K \cap \partial L \neq \emptyset. \end{cases} \qquad (20)$$

We consider two simple error indicators, one based on computing the weighted norm of the flux jump residual, and another based on locally projecting the residual onto element quadratic bubble functions. The first is related to postprocessing the solution using superconvergence, while the second is based on local estimation of the residual, with coupling of global error through the Neumann boundary data.

The explicit flux element error indicator [41] is defined by

$$\eta_K^{\mathrm{FLUX}} := \left(\frac{h_K}{24} \int_{\partial K} |R_K|^2 \, \mathrm{d}s \right)^{1/2}, \tag{21}$$

where h_K is the element diameter.

The element residual error indicator in [8] is based on solution of a local element problem in the space of quadratic bubble functions, denoted by $B^2(K)$. The forcing data is the element residual and flux jump residual. Thus, we solve for $\psi_K^{\mathrm{RES}} \in B^2(K)$ such that

$$\int_K \nabla \psi_K^{\mathrm{RES}} \cdot \nabla v \, \mathrm{d}x = \int_K r_K \, v \, \mathrm{d}x + \int_{\partial K} R_K \, v \, \mathrm{d}s, \quad v \in B^2(K). \tag{22}$$

The local element error indicator is then

$$\eta_K^{\mathrm{RES}} := \left(\int_K |\nabla \psi_K^{\mathrm{RES}}|^2 \, \mathrm{d}x \right)^{1/2}. \tag{23}$$

In either case, we can define $\eta := (\sum_K \eta_K^2)^{1/2}$ to be the global error indicator.

We now present results for a boundary layer computation with domain and boundary given by

$$\begin{aligned} &\Omega := \{(x,y) \in \mathbb{R}^2 \mid y > 0, x^2 + y^2 < 1\}, \\ &\Gamma_N := \{(x,y) \in \mathbb{R}^2 \mid y = 0, -1 \leqslant x \leqslant 1\}, \\ &\Gamma_{\mathrm{D}} := \partial\Omega / \Gamma_N. \end{aligned} \tag{24}$$

The Dirichlet boundary data is defined in polar coordinates by the simple linear function $u_{\mathrm{D}}(s) := u_{\mathrm{D}}(\theta) := 0.5 + \theta/\pi$, for $0 \leqslant \theta \leqslant \pi$ and we take $g_N = 0$ and $f = 0$. We choose the Theile modulus to be $\phi^2 = 100$, which is sufficient to produce a boundary layer along the boundary Γ_{D}. All computations are assisted by adaptive algorithms from the deal.II library [6]. We use a stopping criterion of 1000 degrees of freedom.

We see in Fig. 3 that both error indicators lead to a very similar mesh that is graded into the boundary layer. The mesh refinement and solution history for the averaged flux error indicator is presented in Fig. 4. The similarity between the error indicators can be attributed to the fact that for smooth solutions, the largest contribution to the error, asymptotically, is the edge residuals, which

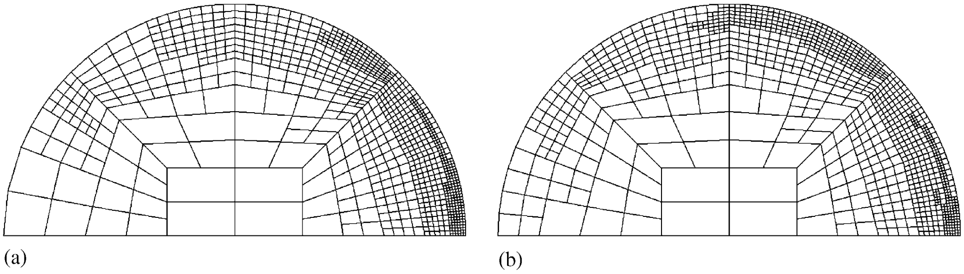

Fig. 3. Final adapted meshes for (17) using (a) averaged flux error indicator η_K^{FLUX} and (b) element residual error indicator η_K^{RES} with $\phi^2 = 100$.

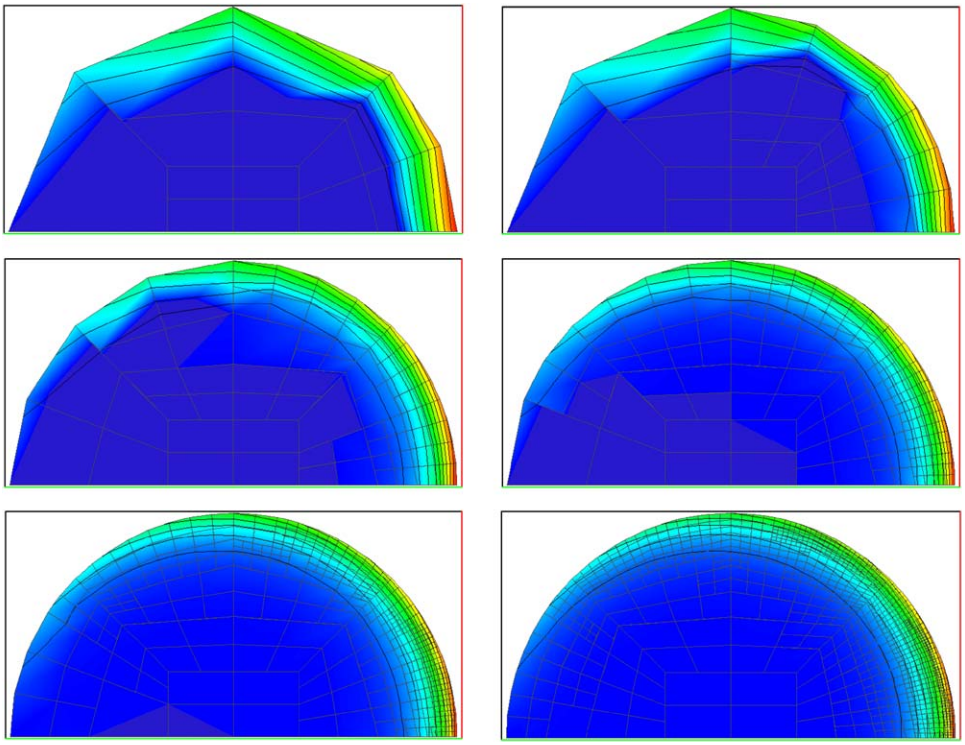

Fig. 4. Sequence of adaptive solutions to (17) with $\phi^2 = 100$ obtained using averaged flux residual error indicator. Contours range from zero to one.

is a special case for using odd polynomial degree elements. Since both error indicators include the flux jump residuals, they lead to the same asymptotic behavior. The element residual error indicator η_K^{RES} is more desirable if one wants an upper bound on the error.

In fact, for a simple problem on the unit square $(0,1)^2$ with $\Gamma_{\text{D}} := \{(x,y) \,|\, x = 1, 0 \leqslant y \leqslant 1\}$ and $u_{\text{D}} := 1$, we can compare the values of the two error indicators on the layer of cells in the boundary layer (which are the cells with the largest error indicator values) under *uniform* refinement. From Fig. 5, we see that asymptotically, the same values are obtained, while on the coarser mesh, the additional element residual term results in a larger value of the error indicator for the element residual error indicator.

8.2. Capillary surface layers

The behavior of the free surface of a fluid, or capillary surface, is governed by a combination of surface tension, gravity, adhesion to boundaries, and the geometry of these boundaries. Minimizing the sum of the associated energies for a static fluid (subject to a volume constraint) results in the Laplace–Young equation for the height u of an equilibrium capillary surface [32] and the contact angle natural boundary condition. In nondimensional form, the Laplace–Young equation can be written as

$$\nabla \cdot [k(|\nabla u|)\nabla u] = u, \tag{25}$$

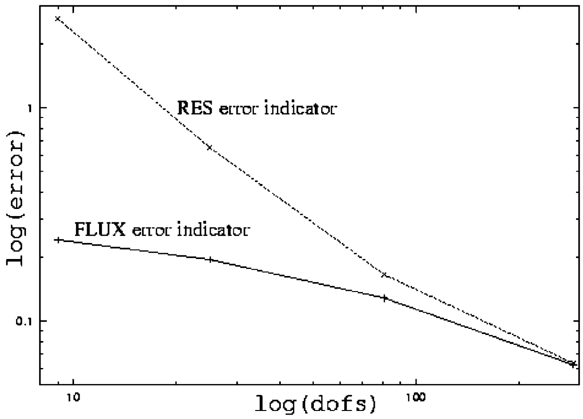

Fig. 5. Values of error indicators on elements within the boundary layer with $\phi^2 = 100$.

where $k(|\nabla u|) = (1 + |\nabla u|^2)^{-1/2}$. In contrast with reaction–diffusion equations (mentioned in the previous case study) where the nonlinearity is in the forcing term, the nonlinearity now occurs in the diffusion term.

The "constant contact angle" boundary condition is

$$k(|\nabla u|)\nabla u \cdot \hat{\mathbf{n}} = \cos\theta, \tag{26}$$

where θ is the contact angle between the surface and the boundary and is determined by the properties of the boundary and fluids involved.

For most liquids, surface tension effects cause either a raised or depressed layer adjacent to the boundary of a container. Typically, such surface tension effects are limited to regions within a small distance of the boundary, and these short length scale effects are often neglected for problems where larger-scale behavior is of interest. In other cases, it may be sufficient to consider a first-order linearization of the nonlinear problem [33]. However, in certain cases, the small-scale boundary layer effects are important. For example, when dipping electronic components into fluids, such as a solder, these effects may cause the contact line between the component and the fluid to be nonuniform in height, weakening the structure.

Also, the fluid behavior near a corner is known to depend strongly on the angle α of the corner, and the contact angle, θ. Depending upon the values of α and θ, the solution can be locally planar [42] or the solution may not extend continuously to the vertex [44]. If the form of the corner singularity is known, special basis functions can be used locally. However, for reentrant corners this information is not available. Therefore adaptive refinement is a logical choice in such situations.

Calculations here are performed using a continuous Galerkin formulation of the weak equation

$$\int_\Omega \left\{ \frac{\nabla u \cdot \nabla \phi}{\sqrt{1 + (\nabla u)^2}} + u\,\phi \right\} \mathrm{d}x = \int_{\partial\Omega} \cos\theta\phi\,\mathrm{d}s. \tag{27}$$

These equations are discretized using bilinear or biquadratic elements and a combination of successive approximation and Newton iteration for nonlinear convergence.

The deal.II library [6] is used for the calculations. The library automatically handles details such as hanging node constraints and ensures that adjacent elements differ by at most one refinement level. To estimate the error in each element the error indicator in (21) is used. The elements were then ranked and a fixed fraction (typically 20%) of the total number cells are refined at each stage. Similarly, the elements with lowest error are coarsened (typically around 5%). For efficiency, the solution on the coarse mesh is used as first iterate on the refined grid in the nonlinear iteration scheme. Due to the small problem size, it was possible to perform all calculations on either a single processor or a dual SMP machine. Consequently, there was no need for domain partitioning.

First, we consider the case of a component of square cross-section that is partially immersed in a fluid. Applying symmetry, the calculations were performed on the L-shaped domain shown in the bottom of Fig. 6: The edges of the inner square have the contact boundary condition (26) and the other boundaries have $\nabla u \cdot \boldsymbol{n} = 0$. The example shown in Fig. 6 is for a 2×2 square in a larger 10×10 domain with $\cos\theta = 0.5$. The initial grid consisted of 24 elements and was not graded in any way. The solution shows the characteristic depression of the fluid along the edges into the corner and significant mesh refinement near the corner singularity.

In a second study, the effect that a small sinusoidal deviation from a straight wall has on solution behavior is examined. Nonadaptive techniques have been used to suggest that such configurations produce almost 1D behavior with modified contact angle conditions [3]. In the example shown in Fig. 7, the disturbance has amplitude 0.2 and period 2π, the domain size is 8×8, and wall boundary condition is the contact condition with $\cos\theta = 0.5$. The results show that as expected the solution is similar to the one-dimensional case. It is evident that near the walls anisotropic refinement would be advantageous as boundary layers are evident. This was not possible using the deal.II library, but the isotropic refinement allowed a good representation of the behavior near the contact boundary and corner.

8.3. The compressible Navier–Stokes equations

In this section, we consider the use of AMR for high-speed compressible flows governed by the Navier–Stokes equations. Eqs. (28)–(30) describe the conservation of mass, momentum, and energy in a compressible gas [64]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{28}$$

$$\rho \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \right] = -\nabla P + \frac{1}{Re} \left[\Delta \boldsymbol{u} + \frac{1}{3} \nabla (\nabla \cdot \boldsymbol{u}) \right], \tag{29}$$

$$\rho \left[\frac{\partial T}{\partial t} + (\boldsymbol{u} \cdot \nabla) T \right] = -(\nabla \cdot \boldsymbol{u}) P + \frac{1}{Re} \left[\frac{\gamma}{Pr} \Delta T + \Phi(\nabla \boldsymbol{u}) \right]. \tag{30}$$

Here $\boldsymbol{u}$ is the fluid velocity, ρ is the density, T is the temperature, P is the pressure, and γ is the ratio of specific heats ($\gamma = 1.4$ for air at standard conditions). The equation set is closed with an

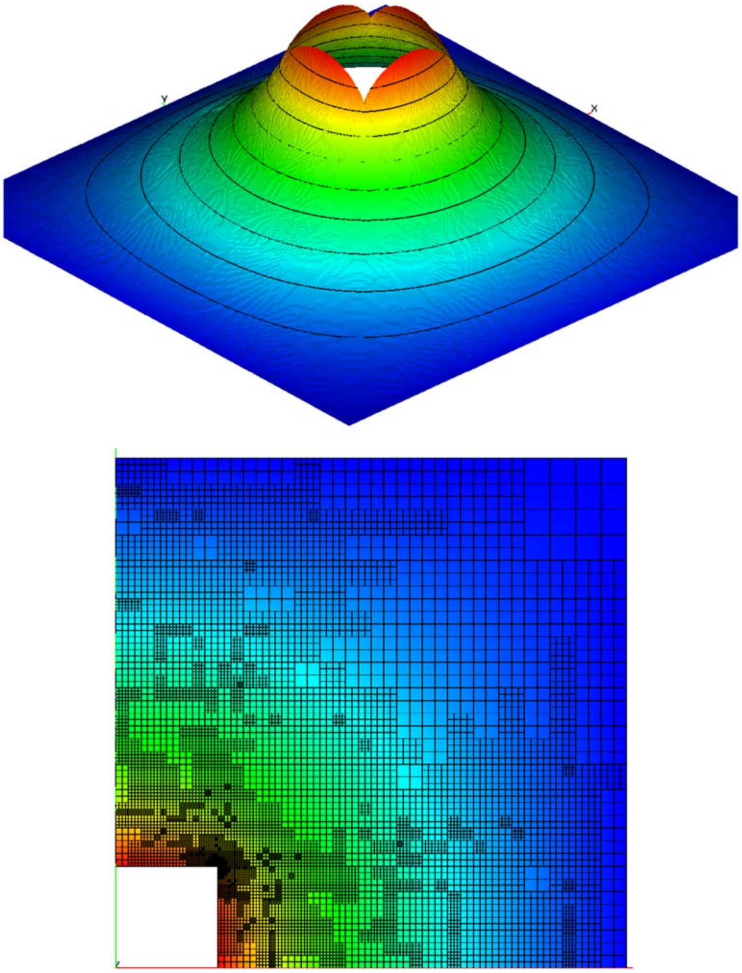

Fig. 6. Solutions for AMR/C solution of Laplace–Young equation near a reentrant corner.

equation of state, which for air at moderate temperatures is the familiar ideal gas equation: $P=\rho RT$. The viscous dissipation function Φ in (30) provides frictional heating in the presence of strong shear layers.

The equations have been nondimensionalized with respect to the freestream conditions and a characteristic length scale, and several parameters arise in the process. The Reynolds number ($Re=\rho UL/\mu$) represents the ratio of convective to diffusive forces and the Prandtl number ($Pr=\mu C_p/k$) is the ratio of viscous to thermal diffusion appear. The Mach number ($M=|\boldsymbol{u}|/c$) is defined as the ratio of fluid speed to the speed of sound and arises in the nondimensionalization of the boundary conditions. When $M<1$ everywhere the flow is subsonic and Eqs. (28)–(30) are parabolic. For supersonic flow ($M>1$), the equations are a mixed hyperbolic and parabolic set [51,64].

In the supersonic case, disturbances in the flowfield are unable to propagate upstream due to the finite propagation speed of pressure waves. In such cases, strong interior shock layers develop

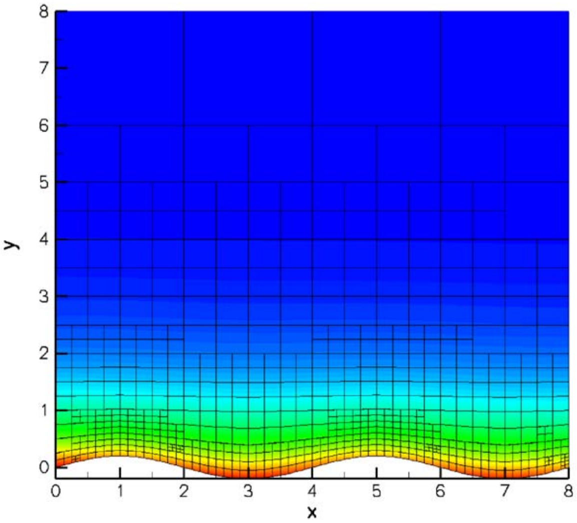

Fig. 7. AMR/C solution of Laplace–Young equation with a sinusoidal boundary.

across which flow properties are nearly discontinuous. In most problems of practical interest, the locations of such shock layers are not known a priori and indeed may change with time. Additionally, flow upstream and downstream of shocks is often relatively uniform. For these problems, very fine grids are required in the vicinity of the shock for proper resolution, and should transition rapidly to a coarse grid away from the layer. This enhances computational efficiency by not over-resolving uninteresting regions of the flowfield. For such applications AMR is very well suited.

Supersonic flow in a converging channel using AMR is shown in Fig. 8. For this (and subsequent test cases) the coupled system of equations is solved implicitly at each timestep. A GLS stabilization scheme is used, and any “hanging nodes” introduced by mesh refinement are constrained algebraically in the resulting linear system [18]. In this case, a pair of oblique shocks is set up from the leading edge of the channel constriction. Upstream of these shocks the flow is uniform, so a coarse mesh is adequate. The shocks intersect and reflect back toward the boundary of the channel, where they interact with the viscous boundary layer. A fine mesh is needed in the shock–shock and shock–boundary layer interaction regions.

Shockwave–boundary layer interaction is another important physical feature of supersonic flow that requires proper mesh resolution. In general, proper resolution of viscous boundary layers (required for accurate skin friction and heat transfer coefficient prediction) requires fine grids normal to solid boundaries. Additionally, the structure of boundary layer solutions can be strongly influenced by shockwaves in supersonic flow. In Fig. 8, the reflected shockwaves intersect the viscous boundary layer, causing a rapid increase in pressure. The boundary layer responds to this adverse pressure gradient by separating and creating the recirculation region located toward the rear of the profile in the bottom right portion of the figure. When the flow reattaches, there is often an increase in heat transfer to the surface, which must be accounted for in the design of supersonic vehicle components. Clearly, in this case, the details of the shockwave–boundary layer interaction is not

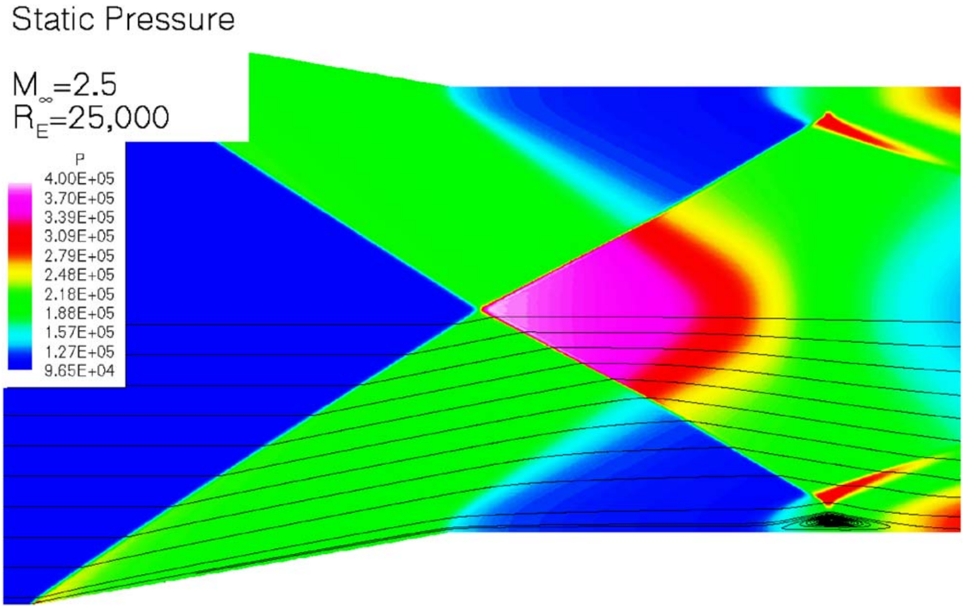

Fig. 8. Pressure contours in a converging channel for a freestream Mach number $M_\infty = 2.5$.

known in advance, and the mesh must evolve with the solution to provide adequate resolution in regions of interest.

Fig. 9 shows the AMR supersonic flow solution in a two-dimensional aircraft inlet at Mach number $M_\infty = 2$. Again, there are large regions of the domain upstream of the inlet that are uniform and require only a coarse mesh resolution. In this case study, the computations begin on a relatively fine initial mesh which is subsequently locally refined based on the flux-jump error indicator in Eq. (**??**), which is applied to all the variables in the coupled system. The refinement procedure does not support coarsening elements below the resolution of the initial mesh, a limitation in this case that results in an over-resolved mesh upstream of the inlet. The flowfield in this example is dominated by a number of intersecting shockwaves and a strong secondary shock that forms as compression waves coalesce from a separated, recirculating region.

On the top boundary, an outflow boundary condition is used to allow fluid to leave the domain. The boundary condition appears to behave well except in the immediate vicinity of the shock. In the shock region, there is excessive numerical dissipation from the shock-capturing algorithm, which causes an artificial reflection from the "outflow" boundary.

An interesting characteristic of shockwaves is that they form strong gradients normal to the wave, but properties parallel to the wave are relatively unaffected. This has the implication that cells in the shock layer subregion should be directionally refined so that degrees of freedom are introduced only when they will help resolve the flowfield. In the example presented here cells are uniformly refined, which may introduce unnecessary degrees of freedom parallel to the shock. This problem is more pronounced in three dimensions where high resolution is only required normal to the shock surface. As discussed previously, directional (or anisotropic) mesh refinement requires more complicated

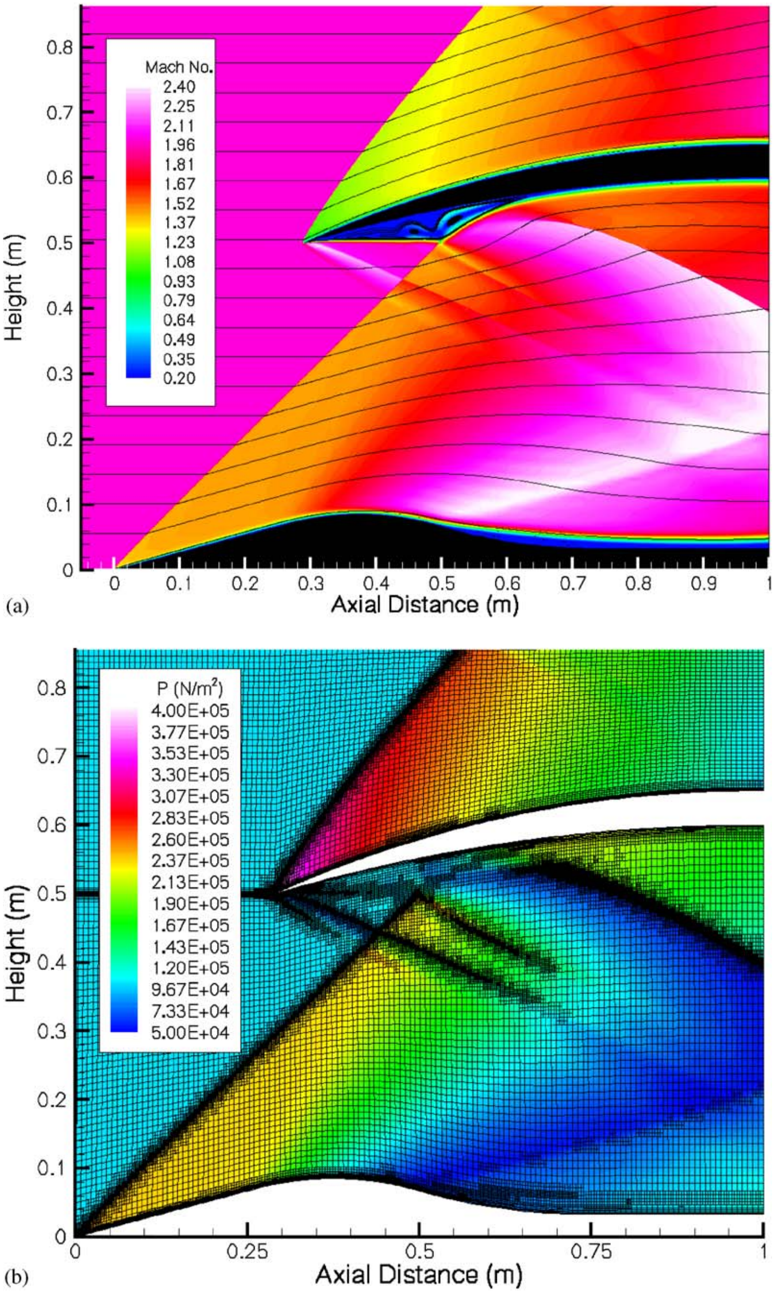

Fig. 9. Compressible flow in an inlet configuration. Freestream Mach number $M_\infty = 2$. Mach number contours with streamlines (a), pressure contours with the computational mesh (b).

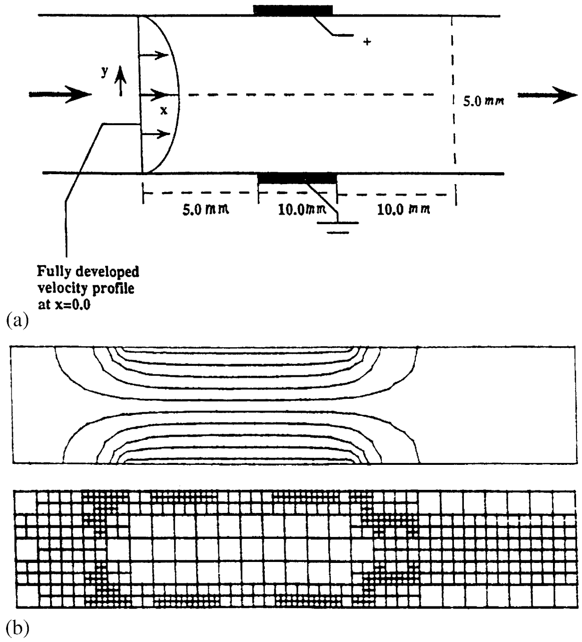

Fig. 10. Geometry (a) and resulting potential field with the adapted mesh (b) for ER fluids application.

software design and puts an additional burden on error indicators, requiring not only that they find cells where the error is large but also how to optimally refine them.

8.4. Coupled problems

Here we describe two examples involving solution of a coupled system of partial differential equations. Similar examples for coupled flow and heat or species transfer are common. A single grid is adaptively refined in each of the cases discussed. It is clear that, in general, one can have multiple layers in several field variables when such coupled problems are considered and in this case it may be efficient to have different adaptive grids (and error indicators) for the respective variables when an iteratively decoupled algorithm is used. However, this also implies that the projection of results between grids will be more complicated [20] as will any parallel algorithms on partitioned evolving AMR grids.

8.4.1. Electro-rheological flow

This example illustrates the use of AMR for resolving a shear-thinning layer and a nonoscillatory interior velocity profile for a generalized Newtonian flow model where the effective viscosity depends locally on the gradient of a potential field. The elementary example in Fig. 10 corresponds to flow of a suspension between two parallel walls with an applied electric field across a pair of plates in the mid-plane region. Particle orientation in the suspension responds to the applied field and the effect on the flow can be modeled by introducing a viscosity that depends on the local electric field. As a

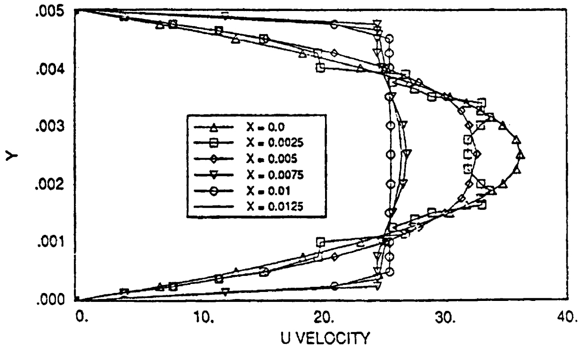

Fig. 11. Velocity profiles at a range of horizontal locations for an ER fluid [68].

result, the flow is "plug-like" in the interior center zone away from the plates and is shear-thinning near the plates, as evident in Fig. 11 which shows computed horizontal velocity profiles at several locations along the length of the plate. The section velocity changes from a quadratic profile upstream of the interaction zone to a bluff profile in the mid-region and back to a quadratic profile as the fluid leaves the interaction zone.

The AMR process not only resolves the shear thinning boundary layers rapidly and efficiently but, as seen in the figure, also refines cells in the interior regime where the fluid enters or leaves the interaction region between the plates. The interior AMR progressively removes oscillations in the velocity profile that are evident for coarser grids during the refinement/solution process.

The governing equations for this problem consist of the electrostatic potential equation and the Navier–Stokes equations with a field-dependent viscosity model. For a fixed grid calculation, the potential problem need be solved once only since coupling is one-way. However, in our case the grid is being adaptively refined based on a local residual indicator for the flow problem so the potential problem is also solved on the updated grids. However, this potential solve is very inexpensive.

8.4.2. *Drift-diffusion semiconductor device problem*

The next example also involves coupled solution with an electrostatic potential equation. The remaining PDE system consists of the drift-diffusion equations for transport of electrons and holes. The behavior of the solution is characterized by extreme interior layers as illustrated by the electrostatic potential solution in Fig. 12. Here the adapted grid is graphed on the solution surface for illustrative purposes. If the grid is not adequately refined, then oscillations occur due to the drift effect and evidence of slight oscillation can be still detected on careful examination of the figure. In this example we have employed an artificial diffusion approach to stabilize the strong effect of the drift term and this leads to some "smearing" in the layers. As refinement proceeds the artificial dissipation reduces accordingly. In this mathematical model, the coupling is two-way with the carrier concentrations entering on the right of the electrostatic potential equation and the electric field entering as the convection coefficient in the drift term of the carrier transport equations. Further details of this application are provided in [24,26].

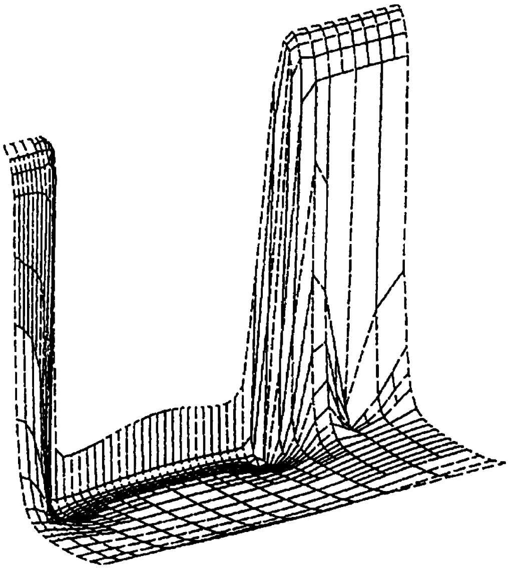

Fig. 12. Electrostatic potential surface plot using adaptive refinement and flux upwinding.

9. Concluding remarks

The widely different length scales arising in applications that exhibit boundary and interior layers make these problems more difficult and expensive to compute using standard numerical methods. Adaptive grid redistribution and refinement schemes provide an approach for addressing this difficulty. Not only is it possible to identify and resolve the location of layers automatically during computation via appropriate error indicators but the resulting discretized systems are often more robust and reliable accurate efficient simulation is enhanced. The ability of such methods to address different classes of layer problems (boundary layers, oscillatory interior layers, etc.) and especially the use of anisotropic and hybrid grids warrants further study. The effectiveness of error indicators, their sensitivity to the layer behavior and the treatment of pollution from the layers are also topics of current interest as are directional error indicators for large problems. Element quality remains a persistent concern especially when redistribution and moving grid techniques are applied. There is a significant overhead in implementing adaptive schemes. This occurs because of the complexity of the data structures and the consequent software requirements. The problem is exacerbated when one considers implementations in parallel computing environments where dynamic load balancing of the adapting grid must be considered. In all, this subject provides a rich area for continued research on boundary and interior layers that will be of great importance to the engineering science community.

Acknowledgements

This material is based upon work supported by Sandia National Laboratories Grant No. 56522.

The work has also been supported by DoD HPCMP PET activities under the terms of Contract No. N62306-01-D-7110 and in part by DoD Contract MSSU 060808-01090729.

References

[1] M. Ainsworth, I. Babuŝka, Reliable and robust a posteriori error estimation for singularly perturbed reaction–diffusion problems, SIAM J. Numer. Anal. 36 (2) (1999) 331–353.

[2] M. Ainsworth, J.T. Oden, A Posteriori Error Analysis in Finite Element Analysis, Wiley Interscience Publishers, New York, August 2000.

[3] M. Anderson, Boundary tracing for partial differential equations, Ph.D. Thesis, University of Western Australia, 2002.

[4] I. Babuška, W.C. Rheinboldt, Computational error estimates and adaptive processes for some nonlinear structural problems, Comput. Methods Appl. Mech. Engrg. 34 (1982) 895–937.

[5] I. Babuška, T. Strouboulis, The Finite Element Method and its Reliability, Oxford Science Publications, Oxford, UK, 2001.

[6] W. Bangerth, R. Hartmann, G. Kanschat, deal.II Differential Equations Analysis Library, Technical Reference, IWR, 2002. URL http://www.dealii.org.

[7] R.E. Bank, Pltmg: a Software Package for Solving Elliptic Partial Differential Equations: Users Guide 6.0. Frontiers in Applied Mathematics, SIAM, Philadelphia, Vol. 7, June 1990.

[8] R.E. Bank, A. Weiser, Some a posteriori error estimators for elliptic partial differential equations, Math. Comput. 44 (170) (1985) 283–301.

[9] R. Becker, H. Kapp, R. Rannacher, Adaptive finite element methods for optimal control of partial differential equations: Basic concept, SIAM J. Control Optim. 1 (2000) 113–13 239.

[10] G. Beckett, J.A. Mackenzie, M.L. Robertson, Moving mesh finite element method for the solution of two-dimensional stefan problems, J. Comput. Phys. 168 (2001) 500–518.

[11] M. Braack, R. Becker, R. Rannacher, Adaptive finite elements for reactive flows, in: H.G. Bock et al. (Eds.), ENUMATH 97 (Heidelberg), World Science Publishing, River Edge, NJ, 1998, pp. 206–213.

[12] D. Braess, W. Dahmen, A cascadic multigrid algorithm for the stokes problem, Numer. Math. 82 (1999) 179–191.

[13] J.H. Bramble, A.H. Schatz, Higher order local accuracy by averaging in the finite element method, Math. Comp. 31 (137) (1977) 94–111.

[14] A. Brandt, Multi-level adaptive solutions to boundary value problems, Math. Comp. 31 (1977) 333–391.

[15] W.L. Briggs, A multigrid tutorial, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1987.

[16] G.F. Carey, A mesh refinement scheme for finite element computations, J. Comput. Meth. Appl. Mech. Engrg. 7 (1) (1976) 93–105.

[17] G.F. Carey, Adaptive refinement and nonlinear fluid problems, J. Comput. Meth. Appl. Mech. Engrg. 17–18 (1979) 541–560.

[18] G.F. Carey, Computational Grids: Generation, Adaptation and Solution Strategies, Taylor & Francis, London, 1997.

[19] G.F. Carey, A robust efficient error indicator for adaptive refinement with pollution control, ICES Report, The University of Texas at Austin, November 2003.

[20] G.F. Carey, G. Bicken, V. Carey, C. Berger, J. Sanchez, Locally constrained projections, Internat. J. Numer Methods Engrg. 50 (2001) 549–577.

[21] G.F. Carey, H. Dinh, Grading functions and mesh redistribution, SIAM J. Numer. Anal. 50 (5) (1985) 1028–1040.

[22] G.F. Carey, B.A. Finlayson, Orthogonal collocation on finite elements, J. Chem. Engrg. Sci. 30 (1975) 587–596.

[23] G.F. Carey, D. Humphrey, Mesh refinement and iterative solution methods for finite element computations, Internat. J. Numer. Meth. Engrg. 17 (1981) 1717–1734.

[24] G.F. Carey, R. Richardson, C. Reed, B. Mulvaney, Circuit, Device and Process Simulation, Wiley, New York, 1996.

[25] G.F. Carey, M. Seager, Projection and iteration in adaptive finite element refinement, Internat. J. Numer. Meth. Engrg. 21 (1985) 1681–1695.

[26] G.F. Carey, M. Sharma, Semiconductor device simulation using adaptive refinement and flux upwinding, IEEE Trans. Comput. Aided Des. Integrated Circuits Systems 8 (6) (1989) 590–598.

[27] J. Cole, Perturbation Methods in Applied Mathematics, Blaisdell, New York, 1968.

[28] P. Deuflhard, Cascadic conjugate gradient methods for elliptic partial differential equations: algorithm and numerical results, in: D.E. Keyes and J. Xu (Eds.), Domain Decomposition Methods in Scientific and Engineering Computing, University Park, PA, 1993, Contemporary Mathematics, American Mathematical Society, Providence, RI, 1994, pp. 29–42.

[29] R.W Douglass, G.F. Carey, D.R. White, G.A. Hansen, Y. Kallinderis, N. Weatherill, Current views on grid generation: summaries of a panel discussion, Numer. Heat Transfer (Part B) 41 (2002) 211–237.

[30] D.J. Estep, M.G. Larson, R.D. Williams, Estimating the error of numerical solutions of systems of reaction–diffusion equations, Mem. Amer. Math. Soc. 146(696) (2000) viii+109.

[31] P.A. Farrell, A.F. Hegarty, J.J.H. Miller, E. O'Riordan, G.I. Shishkin, Singularly perturbed convection diffusion problems with boundary and weak interior layers, in: S. Wang, N. Fowkes (Eds.), Proceedings of BAIL2002, The University of Western Australia, 2002, pp. 115–120.

[32] R. Finn, Equilibrium Capillary Surfaces, Springer, New York, 1986.

[33] N.D. Fowkes, M.J. Hood, Surface tension effects in a wedge, Quart. J. Mech. Appl. Math. 51 (1998) 553–561.

[34] R.J. Gelinas, S.K. Doss, K. Miller, The moving finite element method: applications to general partial differential equations with multiple large gradients, J. Comput. Phys. 40 (1) (1981) 202–249.

[35] F. Hebeker, R. Rannacher, An adaptive finite element method for unsteady convection-dominated flows with stiff source terms, SIAM J. Sci. Comput. 21 (3) (1999) 799–818.

[36] B. Hendrickson, K. Devine, Dynamic load balancing in computational mechanics, Comput. Methods Appl. Mech. Engrg. 184 (2000) 485–500.

[37] B. Hendricksen, R. Leland, Multidimensional spectral load balancing, Technical Report, Technical Report SAND 93-0074, Sandia National Laboratories, January 1993.

[38] C. Johnson, Adaptive computational methods for differential equations, in: Ball, Hunt (Eds.), Proceedings from ICIAM99, Oxford University Press, Oxford, 2000, pp. 96–104.

[39] C. Johnson, R. Rannacher, M. Boman, Numerics and hydrodynamic stability—toward error control in computational fluid dynamics, SIAM J. Numer. Anal. 32 (4) (1995) 1058–1079.

[40] P.G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1) (1998) 359–392.

[41] D.W. Kelly, J.P.d.S.R. Gago, O.C. Zienkiewicz, I. Babuška, A posteriori error analysis and adaptive processes in the finite element method. I. Error analysis, Internat. J. Numer. Methods Engrg. 19 (11) (1983) 1593–1619.

[42] J.R. King, J.R. Ockendon, H. Ockendon, The Laplace–Young equation near a corner, Quart. J. Mech. Appl. Math. 52 (1) (1999) 73–97.

[43] B. Kirk, K. Lipnikov, G.F. Carey, Nested grid iteration for incompressible viscous flow and transport, Int. J. Comput. Fluid Dynamics 17 (2003) 253–262.

[44] N.J. Korevaar, On the behavior of a capillary surface at a re-entrant corner, Pacific J. Math. 88 (2) (1980) 379–385.

[45] S.F. McCormick, Multilevel adaptive methods for partial differential equations, Frontiers in Applied Mathematics, Vol. 6, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1989.

[46] A.C. Mueller, G.F. Carey, Continuously deforming finite elements, Internat. J. Numer. Methods Engrg. 21 (11) (1985) 2099–2126.

[47] J.T. Oden, S. Prudhomme, Goal-oriented error estimation and adaptivity for the finite element method, Comput. Math. Appl. 41 (2001) 735–756.

[48] A. Pardhanani, G.F. Carey, Optimization of computational grids, Numer. Methods Partial Differential Equations 4 (2) (1988) 95–117.

[49] A. Patra, A. Pehlivanov, D. Littlefield, G.F. Carey, J.T. Oden, Error indicators and adaptive finite element methods for large deformation lagrangian simulations of impacts, in: Proceedings of the Department of Defense High Performance Computing Users Group Conference. Monterey, CA, June 7–11 1999.

[50] C.E. Pearson, On a differential equation of boundary layer type, J. Math. Phys. 47 (1968) 134–154, 351–358.

[51] R. Peyret (Ed.), Handbook of Computational Fluid Mechanics, Academic Press, New York, 1996.

[52] A. Plaza, G.F. Carey, Local refinement of simplicial grids based on the skeleton, Appl. Numer. Math. 32 (2000) 95–218.

[53] J. Pousin, J. Rappaz, Consistency, stability, a priori and a posteriori errors for Petrov–Galerkin methods applied to nonlinear problems, Numer. Math. 69 (1994) 213–231.

[54] W. Prager, A note on the optimal choice of finite element grids, Comput. Methods Appl. Mech. Engrg. 6 (3) (1975) 363–366.

[55] L. Prandtl, On fluid motions with very small friction, Verhdlg. 3. intern. math. Kongr, also NACA Technical Mem. 452, 1904.

[56] R. Rannacher, Adaptive galerkin finite element methods for partial differential equations, J. Comput. Appl. Math. 128 (1–2) (2001) 205–233.

[57] M.C. Rivara, G. Iribarren, The 4-triangles longest-side partition of triangles and linear refinement algorithms, Math. Comput. 65 (216) (1997) 1485–1502.

[58] U. Rüde, The hierarchical basis extrapolation method, SIAM J. Sci. Statist. Comput. 13 (1) (1992) 307–318.

[59] H. Samet, The quadtree and related hierarchical data structures, Comput. Surveys 16 (2) (1984) 187–260.

[60] J.A. Sethian, Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science, Level Set Methods, Cambridge Monographs on Applied and Computational Mathematics, Vol. 3, Cambridge University Press, Cambridge, 1996.

[61] R.V. Southwell, Relaxation Methods in Theoretical Physics. A Continuation of the Treatise “Relaxation Methods in Engineering Science”, Clarendon Press, Oxford, 1956.

[62] W.F. Spotz, G.F. Carey, A high-order compact formulation for the 3D Poisson equation, Numer. Methods Partial Differential Equations 12 (2) (1996) 235–243.

[63] G. Strang, T. Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge Press, Wellesley, MA, 1996.

[64] J.C. Tannehill, D.A. Anderson, R.H. Pletcher, Computational Fluid Mechanics and Heat Transfer, Taylor & Francis, London, 1997.

[65] R. Verfürth, A Review of Posteriori Error Estimation & Adaptive Mesh-refinement Techniques, Wiley, New York, 1996.

[66] R. Verfürth, Robust a posteriori error estimators for a singularly perturbed reaction–diffusion equation, Numer. Math. 78 (1998) 479–493.

[67] L.B. Wahlbin, Superconvergence in Galerkin Finite Element Methods, Springer, Berlin, 1995.

[68] K.C. Wang, R. McLay, G.F. Carey, E-R Fluid Modeling, in: Carlson et al. (Eds.), Proc. 2nd Int. Conf. on Electro-Rheological Fluids, Technomic Publishers, PA, USA, 1989, pp. 41–52.

[69] D. Young, Iterative Solution of Large Linear Systems, Academic Press, New York, 1971.

[70] J.Z. Zhu, O.C. Zienkiewicz, Adaptive techniques in the finite element method, Comput. Appl. Numer. Methods 4 (1988) 197–204.