# 8
# Discretization error

Recall that our definition of a mathematical model is a governing partial differential or integral equation along with its associate initial and boundary conditions. In scientific computing, one is concerned with finding approximate solutions to this mathematical model, a process that involves the discretization of both the mathematical model and the domain. The approximation errors associated with this discretization process are called discretization errors, and they occur in nearly every scientific computing simulation. The *discretization error* can be formally defined as the difference between the exact solution to the discrete equations and the exact solution to the mathematical model:

$$\varepsilon_h = u_h - \tilde{u}. \tag{8.1}$$

In the next chapter, it will be shown that the discretization error arises out of the interplay between the chosen discretization scheme for the mathematical model, the mesh resolution, the mesh quality, and the solution behavior. Discretization error is the most difficult type of numerical error to estimate reliably (i.e., accurately) and is usually the largest of the four numerical error sources discussed in Chapter 7.

The discretization error has two components: one that is locally-generated and one that is transported from other parts of the domain. The transported component is also called pollution error in the finite element literature (e.g., Babuska *et al.*, 1997). The existence of these two components was shown mathematically in Chapter 5 using the continuous error transport equation which relates the convergence of the numerical solution (i.e., the discretization error) to the consistency of the discretization scheme (i.e., the truncation error). It was showed that the discretization error is transported in the same manner as the underlying solution properties (e.g., it can be convected and diffused) and that it is locally generated by the truncation error.

An example of discretization error transport for the Euler equations is shown in Figure 8.1, which gives the discretization error in the density for the inviscid, Mach 8 flow over an axisymmetric sphere-cone (Roy, 2003). The flow is from left to right, and large discretization errors are generated at the bow shock wave where the shock and the grid lines are misaligned. In the subsonic (i.e., elliptic) region of the flow immediately behind the normal shock, these errors are convected along the local streamlines. In the supersonic (hyperbolic) regions these errors propagate along characteristic Mach lines and

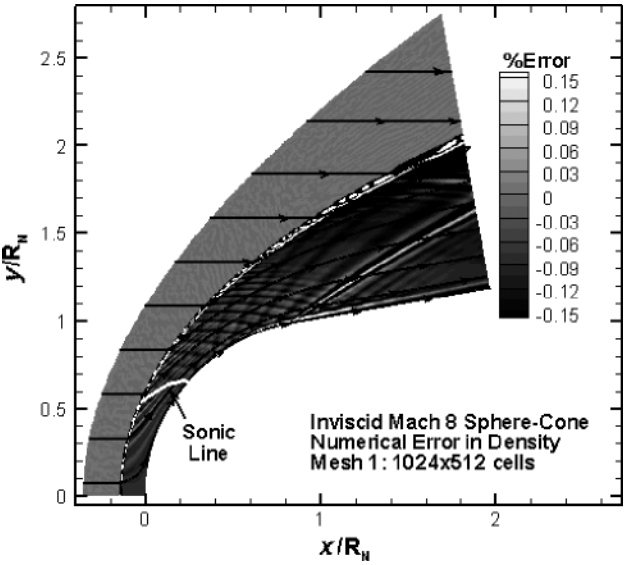

Figure 8.1 Contours of total estimated discretization error in density for the flow over an inviscid hypersonic sphere cone (from Roy, 2003). (See color plate section.)

reflect off the surface. Additional discretization error is generated at the sphere-cone tangency point (i.e., the point where the spherical nose joins the conical body), which represents a singularity due to the discontinuity in the surface curvature. Errors from this region also propagate downstream along the characteristic Mach line. An adaptation process which is driven by the global error levels would adapt to the characteristic line emanating from the sphere-cone tangency point, which is not desired. An adaptation process driven by the local contribution to the error should adapt to the sphere-cone tangency point, thus obviating the need for adaption to the characteristic line that emanates from it. Targeted mesh adaptation is discussed in detail in the next chapter.

This chapter contains an overview and classification of several approaches for estimating the discretization error, a summary of which is given by Roy (2010). Many of these approaches have arisen out of the finite element method, which due to its nature provides for a rigorous mathematical analysis. Other approaches to discretization error estimation seek to find an estimate of the exact solution to the mathematical model which is of a higher formal order of accuracy than the underlying discrete solution. Once this higher-order estimate is found, it can be used to estimate the discretization error in the solution.

While a wide range of approaches are discussed for estimating discretization error, particular attention is focused on Richardson extrapolation which relies on discrete solutions on two meshes to estimate the discretization error. The reason for this choice is straightforward: Richardson extrapolation is the only discretization error estimator that can be applied as a post-processing step to both local solutions and system response quantities from any discretization approach. The main drawback is the expense of generating and then computing another discrete solution on a systematically-refined mesh.

Regardless of the approach used for estimating the discretization error, the reliability of the resulting error estimate requires that the underlying numerical solution (or solutions) be in the asymptotic range. Achieving this asymptotic range can be surprisingly difficult, and confirming that it has indeed been achieved generally requires at least three discrete solutions. For complex scientific computing applications involving coupled, nonlinear, multi-dimensional, multi-physics equations, it is unlikely that the asymptotic range will be achieved without the use of solution adaptive procedures discussed in the next chapter.

The most common situation in scientific computing is when the discretization error estimate has been computed, but the reliability of that estimate is either (1) low because the asymptotic range has not been achieved or (2) unknown because three discrete solutions are not available. In these cases, the discretization error is more appropriately characterized as an epistemic uncertainty since the true value of the error is not known. Roache's grid convergence index converts the error estimate from Richardson extrapolation into an uncertainty by providing error bands.

Another important topic addressed in this chapter is the role of systematic mesh refinement for Richardson extrapolation-based discretization error estimators as well as for assessing the reliability of all discretization error estimators. The importance of systematic mesh refinement over the entire domain is discussed, along with approaches for assessing the systematic nature of the refinement. Issues with refinement in space and time, unidirectional refinement, fractional (or noninteger) refinement, and recommendations for refinement versus coarsening are also discussed. This chapter concludes with a discussion of some issues related to discretization error estimation that have not yet been adequately addressed by the research community.

## 8.1 Elements of the discretization process

The goal of the discretization process is to convert the mathematical model into a set of algebraic equations which can be readily solved on a digital computer. There are two distinct steps in this process: discretization of the mathematical model and discretization of the domain. While most applied scientific computing codes employ both aspects of discretization, some discretization schemes do not. For example, the method of weighted residuals does not include a domain discretization step (Huebner, 2001).

### *8.1.1 Discretization of the mathematical model*

Throughout this book, we have made numerous references to the mathematical model being composed of partial differential equations. However, this is not the only form that the governing equations can take in scientific computing. The partial differential equation form arises when the conceptual model (e.g., the conservation of energy principle) is applied to an infinitesimal element. If instead a control volume of finite size is used, then the divergence theorem can be used to convert the volume integral of the gradient of a quantity to fluxes of that quantity through the boundary. The resulting integral (or integro-differential) form of the governing equations is called the weak form of the equations since it admits solutions with discontinuities (i.e., weak solutions) that satisfy the differential equation only in a restricted sense. The weak form is more fundamental since it admits discontinuous (weak) solutions, whereas the differential form relies on the dependent variables being differentiable and therefore continuous (i.e., strong solutions).

In order to find the solution to a mathematical model, initial and/or boundary conditions must be supplied. In some cases, these initial and boundary conditions also take the form of differential equations, such as the Neumann (or gradient) boundary condition, where the gradient of a property normal to a boundary is specified. For example, the adiabatic (or zero heat flux) boundary condition for a heat conduction problem has the form

$$\left.\frac{\partial T}{\partial n}\right|_{\Gamma} = 0 \tag{8.2}$$

on the boundary $\Gamma$, where $n$ is defined to be normal to the boundary. To ensure that the boundary condition discretization does not have an adverse impact on the overall order of accuracy of the code, the formal order of accuracy of the boundary condition discretization must be the same as (or higher than) that of the interior discretization scheme. This is not to be confused with the order of the boundary condition itself (i.e., the highest derivative in the boundary condition), which must be at most one order less than the governing equation.

While there are a number of different discretization approaches, the approaches discussed in this section are the primary ones found in scientific computing software. The finite difference method (Section 8.1.1.1) employs the strong form of mathematical model, whereas the finite volume method (Section 8.1.1.2) and the finite element method (Section 8.1.1.3) employ the weak form of the equations. Other approaches not discussed here but which can also result in consistent and convergent discretizations include boundary element methods, spectral element methods, and pseudo-spectral methods.

### *8.1.1.1 The finite difference method*

Consider a general scalar partial differential equation in two spatial dimensions and time:

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = Q, \tag{8.3}$$

where *f, g*, and *Q* are all considered to be functions of *u*. In the finite difference method, the derivatives in Eq. (8.3) are replaced by finite differences which, for example, can be found using Taylor series expansions. If a first-order accurate forward difference in time is used, along with second-order accurate central differences in space, the finite difference discretization of Eq. (8.3) becomes:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} + \frac{f_{i+1,j}^{n} - f_{i-1,j}^{n}}{2\Delta x} + \frac{g_{i,j+1}^{n} - g_{i,j-1}^{n}}{2\Delta y} = Q_{i,j}^{n}. \tag{8.4}$$

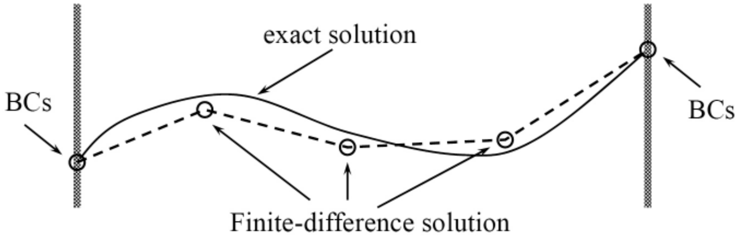

Figure 8.2 Simple example of a 1-D finite difference solution.

The superscripts denote discrete time location separated by the temporal step $\Delta t$, while the $i$ and $j$ subscripts denote discrete spatial locations separated by $\Delta x$ in the $x$ coordinate direction and $\Delta y$ in the $y$ coordinate direction, respectively. Due to the required ordering of spatial points, finite difference methods are generally implemented on structured grids. The above example employs a simple Cartesian mesh with constant spacing in the $x$ and $y$ directions and time. Finite difference methods can be applied to more general geometries with the use of transformations between physical space and a Cartesian computational space (e.g., see Chapter 9). See Richtmyer and Morton (1967), Tannehill *et al.* (1997), and Morton and Mayers (2005) for more details on finite difference methods.

A simple 1-D example of a typical finite difference solution is presented in Figure 8.2. While the exact solution exists for all points in the domain, the finite difference solution exists only at the discrete grid nodes. Interpolation between nodal values can be used to obtain the numerical solution between grid nodes, and the figure displays simple linear interpolation. Higher-order interpolations can be used to improve the accuracy between grid nodes, but care must be taken in interpreting any new extrema that arise. In any case, the interpolation will not produce values that are higher-order accurate than the underlying finite difference discretization.

### *8.1.1.2 The finite volume method*

In contrast to the finite element and finite difference methods, the finite volume method requires the governing equations in integro-differential form. One approach for obtaining the equations in this form is to start with a finite control volume when deriving the governing equations. However, if the equations are already available in differential form, then the integro-differential form can be found by integrating the differential equations over the finite control volume $\delta V$, as

$$\int\limits_{\Omega} \left( \frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} - Q \right) \mathrm{d}\Omega = 0, \tag{8.5}$$

where $\Omega$ is the domain.

Applying the divergence theorem to the terms involving $f$ and $g$ gives

$$\int\limits_{\Omega} \frac{\partial u}{\partial t} \mathrm{d}V + \int\limits_{\Gamma} \vec{\phi} \cdot \hat{n} \mathrm{d}\Gamma = \int\limits_{\Omega} Q \mathrm{d}V, \tag{8.6}$$

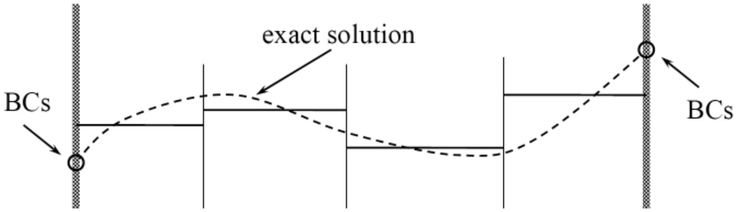

Figure 8.3 Simple example of a 1-D finite volume solution.

where $\Gamma$ is the surface of the control volume, $\hat{n}$ is the outward-pointing unit normal vector, and $\vec{\phi}$ is the vector composed of the flux components, which in 2-D Cartesian coordinates is:

$$\vec{\phi} = f\hat{\imath} + g\hat{\jmath}. \tag{8.7}$$

If the volume integrals are replaced by their exact averages over the cell, e.g.,

$$\bar{Q}_{i,j} = \frac{\int_{\Omega} Q \mathrm{d}\Omega}{\delta\Omega}, \tag{8.8}$$

and the area integrals replaced by their exact averages over the face, e.g.,

$$\bar{f}_{i+1/2,j} = \frac{\int_{\Gamma_{i+1/2,j}} f \mathrm{d}\Gamma}{\Gamma_{i+1/2,j}}, \tag{8.9}$$

then, assuming a 2-D Cartesian mesh, the integral equation can be rewritten as

$$\left.\overline{\frac{\partial u}{\partial t}}\right|_{i,j} + \frac{\bar{f}_{i+1/2,j} - \bar{f}_{i-1/2,j}}{\Delta x} + \frac{\bar{g}_{i,j+1/2} - \bar{g}_{i,j-1/2}}{\Delta y} = \bar{Q}_{i,j}. \tag{8.10}$$

For finite volume methods, the formal order of accuracy is related to the approximations used to represent cell averages (often simply by the cell-center value) and face averages. The face average fluxes are generally found by either interpolating neighboring cell center values (equivalent to a central differencing scheme on uniform meshes) or extrapolating in an appropriate upwind manner from neighboring cells, and these two procedures are collectively referred to as flux quadrature. Finite differences are usually used to discretize the unsteady term with respect to time ($\Delta t$). Additional information on the finite volume method can be found in Knight (2006) and Hirsch (2007).

An example of a finite volume solution in 1-D is presented in Figure 8.3. The discrete version of the integro-differential equation is applied in each of the regions shown. While the solution in each cell is generally considered to be a constant value, different order accurate quadratures can be used to determine interface flux values, resulting in higher-order schemes. Due to the manner in which these interface fluxes are calculated, the finite volume method satisfies local conservation of quantities from cell to cell.

### 8.1.1.3 The finite element method

Consider the 2-D steady heat conduction equation with a volumetric source term $Q$ as given by

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)-Q=0, \tag{8.11}$$

where $k$ is the thermal conductivity. Following closely the development of Zienkiewicz *et al.* (2005), we will apply the finite element method to Eq. (8.11), although we will omit the boundary conditions for simplicity. In the finite element method, approximate solutions are sought within a limited function space

$$T\approx\hat{T}=\sum_{i=1}^{n}N_i a_i, \tag{8.12}$$

where the $N_i$ are shape (or basis) functions that are functions of the independent variables only and the $a_i$ are unknown coefficients. Once the shape functions are chosen, then the finite element method involves solving for these $a_i$ coefficients on domain subvolumes (i.e., elements). In solving for these coefficients, the finite element method employs the weak form of the governing equation. This weak form can generally be obtained in one of two ways: using the method of weighted residuals (i.e., the Galerkin procedure) or by a variational formulation which requires that a variational principle exist for the problem of interest. Since variational principles do not exist for all scientific computing applications, we will employ the method of weighted residuals in this example.

If Eq. (8.11) is equal to zero everywhere in the domain, then we are free to integrate this equation over the domain volume $\Omega$ using an arbitrary function $v$,

$$\int_{\Omega} v\left[\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)-Q\right]\mathrm{d}x\mathrm{d}y=0, \tag{8.13}$$

where the functions $v$ are called test (or trial) functions. In the finite element approximation, these test functions are also limited to some subspace of functions as

$$v=\sum_{j=1}^{n}w_j b_j, \tag{8.14}$$

where the $w_j$ are functions of the independent variables and the $b_j$ are coefficients. Note that performing the integration in Eq. (8.13) requires that the solution $T(x, y)$ be $\mathrm{C}^1$ continuous, meaning that both the temperature and its slope are continuous. Applying integration by parts (i.e., using Green's identities) results in the weak form

$$\int_{\Omega}\left(\frac{\partial v}{\partial x}k\frac{\partial T}{\partial x}+\frac{\partial v}{\partial y}k\frac{\partial T}{\partial y}-vQ\right)\mathrm{d}x\mathrm{d}y-\int_{\Gamma} vk\left(\frac{\partial T}{\partial x}n_x+\frac{\partial T}{\partial y}n_y\right)\mathrm{d}\Gamma=0, \tag{8.15}$$

where

$$\frac{\partial T}{\partial n} = \frac{\partial T}{\partial x} n_x + \frac{\partial T}{\partial y} n_y \tag{8.16}$$

is the derivative normal to the domain boundary $\Gamma$. Since only first derivatives appear in Eq. (8.15), this weak form thus requires only $C^0$ continuity (i.e., allowing discontinuous slopes) to be integrable. It is convenient to choose the test functions $v$ such that they are zero on the boundary $\Gamma$, thus allowing Eq. (8.15) to be rewritten as simply

$$\int\limits_{\Omega} \left( \frac{\partial v}{\partial x} k \frac{\partial T}{\partial x} + \frac{\partial v}{\partial y} k \frac{\partial T}{\partial y} \right) \mathrm{d}x\mathrm{d}y = \int\limits_{\Omega} vQ\mathrm{d}x\mathrm{d}y. \tag{8.17}$$

The choice of $w_j = N_j$ results in the Galerkin method (also called the Bubnov–Galerkin method). Any other choice for $w_j$ is considered a Petrov–Galerkin scheme, and it can be shown that both the finite difference method and the finite volume method can be written as specific cases of Petrov–Galerkin schemes called point collocation and subdomain collocation, respectively (Zienkiewicz *et al.*, 2005).

Decomposing the domain into elements and using the Galerkin method (i.e., $w_j = N_j$), we can then write, for any given element $\Omega_k$,

$$b_j \left[ \int\limits_{\Omega_k} \left( \frac{\partial N_j}{\partial x} k \frac{\partial \hat{T}}{\partial x} + \frac{\partial N_j}{\partial y} k \frac{\partial \hat{T}}{\partial y} \right) \mathrm{d}x\mathrm{d}y - \int\limits_{\Omega_k} N_j Q \,\mathrm{d}x\mathrm{d}y \right] = 0, \quad j = 1, \ldots, n; \tag{8.18}$$

therefore we are free to omit the $b_j$ coefficients. Equation (8.18) represents $n$ different equations since there are $n$ terms in the summation for the test functions. Substituting the finite element approximation from Eq. (8.12) gives

$$\int\limits_{\Omega_k} \left[ \frac{\partial N_j}{\partial x} k \left( \sum_{i=1}^{n} \frac{\partial N_i}{\partial x} a_i \right) + \frac{\partial N_j}{\partial y} k \left( \sum_{i=1}^{n} \frac{\partial N_i}{\partial y} a_i \right) \right] \mathrm{d}x\mathrm{d}y$$
$$= \int\limits_{\Omega_k} N_j Q\mathrm{d}x\mathrm{d}y, \quad j = 1, \ldots, n. \tag{8.19}$$

Equations (8.19) form a system of $n$ equations on each element with the $n$ unknowns being the coefficients $a_i$. If the thermal conductivity $k$ and the heat source $Q$ are functions of $x$ and $y$ only, then Eqs. (8.19) are a linear system. If they are functions of the temperature, then this system of equations is nonlinear and requires linearization before solving in an iterative manner. The form of the shape functions will dictate the number of unknowns $n$ (i.e., degrees of freedom) on each element as well as the formal order of accuracy of the method. For more information on the finite element method, see Oden and Reddy (1976), Szabo and Babuska (1991), Hughes (2000), and Zienkiewicz *et al.* (2005).

A notional representation of a 1-D finite element solution is presented in Figure 8.4. This solution shows interfaces that are discontinuous (left interface), $C^0$ or value continuous

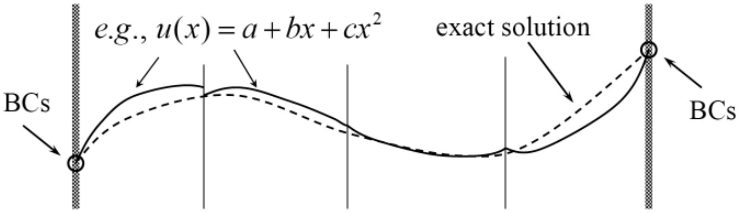

Figure 8.4 Simple example of a 1-D finite element solution.

(right interface), and $C^1$ or slope continuous (center interface). It is important to note that finite element methods enforce global conservation over the entire domain, but generally do not enforce conservation locally.

### *8.1.2 Discretization of the domain*

For steady problems, discretization of the domain involves decomposing the spatial domain into nodes, cells, or elements where the discrete equations will be applied. For unsteady problems, the temporal domain must also be discretized by selecting an appropriate time step $\Delta t$ to advance the simulations. This time step must be chosen to resolve the physical phenomena in question, to satisfy any stability criteria in the problem (recall the discussion of numerical stability in Chapter 5), and to achieve the desired level of temporal accuracy. The terms "mesh" and "grid" are often used to refer to the partitioned domain in space; however, for unsteady simulations the term "mesh" will also refer to the discretized temporal domain.

For complex, 3-D geometries, the generation of an appropriate spatial mesh can require significant time and effort. Two mesh-related issues that have a significant impact on the discretization error are mesh resolution and mesh quality. The effects of mesh resolution on the discretization error are clear when one considers that the truncation error serves as the local source for generating discretization error. (This relationship was initially shown in Chapter 5 and is further developed in this chapter during the discussion of error transport equations in Section 8.2.2.1.) The truncation error contains the discretization parameters ($\Delta x$, $\Delta y$, etc.) to various powers, with the lowest power determining the formal order of accuracy of the discretization scheme. Thus the discretization error can be reduced by simply refining the mesh. Mesh quality addresses factors such as cell aspect ratio, cell stretching ratio, cell skewness, and orthogonality of mesh lines at a boundary. Poor quality spatial meshes can result in a reduction in the order of accuracy and even non-convergence of the discrete solution in extreme cases. The relationship between mesh quality and the discretization error is investigated further in Chapter 9.

#### *8.1.2.1 Structured meshes*

There are two basic types of spatial mesh: structured and unstructured. In 3-D structured meshes, each node or element is uniquely defined by an ordered set of indices ($i$, $j$, and $k$).

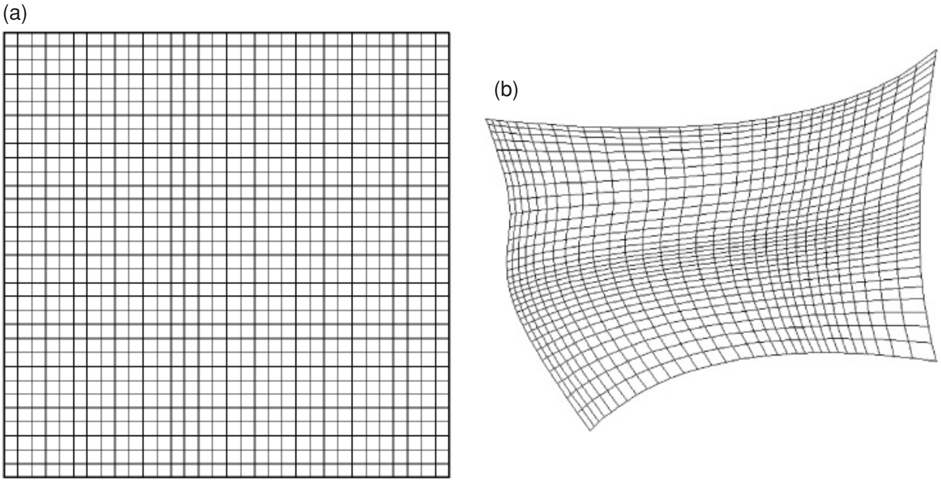

Figure 8.5 Structured mesh examples in 2-D: (a) Cartesian and (b) curvilinear.

Structured grids can either be Cartesian, stretched Cartesian, skewed Cartesian, or curvilinear. Cartesian grids have fixed spacing ($\Delta x$, $\Delta y$, and $\Delta z$) between orthogonal grid lines, whereas stretched Cartesian and skewed Cartesian relax the requirements for fixed spacing and orthogonality, respectively. Curvilinear structured grids allow the mesh to wrap around a surface and the spacing can vary over the domain. Examples of structured Cartesian and curvilinear meshes in 2-D are shown in Figure 8.5a and Figure 8.5b, respectively. The regular ordering of the structured grid generally makes it more efficient for computing numerical solutions than unstructured meshes, but at the cost of increased difficulty during mesh generation for complicated geometries.

#### *8.1.2.2 Unstructured meshes*

Unstructured meshes do not employ regular ordering of the nodes or elements. They can be composed of triangles, quadrilaterals, or arbitrary elements in 2-D, and pyramids, tetrahedrons, hexahedrons, prisms, and arbitrary elements in 3-D. There is no equivalent unstructured element type in 1-D as the nodes/elements in a 1-D mesh can always be indexed in an ordered manner. Two examples of unstructured meshes in 2-D are given in Figure 8.6. A general unstructured triangular mesh is shown in Figure 8.6a, while a hybrid mesh containing structured quadrilateral and triangular elements is shown in Figure 8.6b.

#### *8.1.2.3 Cartesian meshes*

Cartesian meshes can also be used for general geometries. For Cartesian grid methods, curved boundaries are usually handled with special boundary cells called cut-cells. An example of a Cartesian grid where the underlying mesh is structured is given in Figure 8.7a. This type of mesh could be handled by a structured grid code with special treatment of the curved boundaries. The more common example of a Cartesian grid

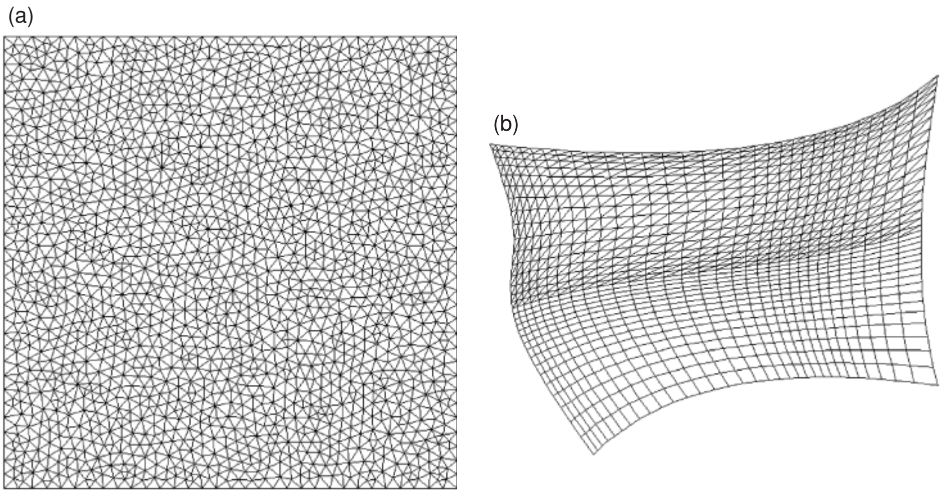

Figure 8.6 Unstructured meshes in 2-D: (a) general triangular and (b) hybrid structured/unstructured.

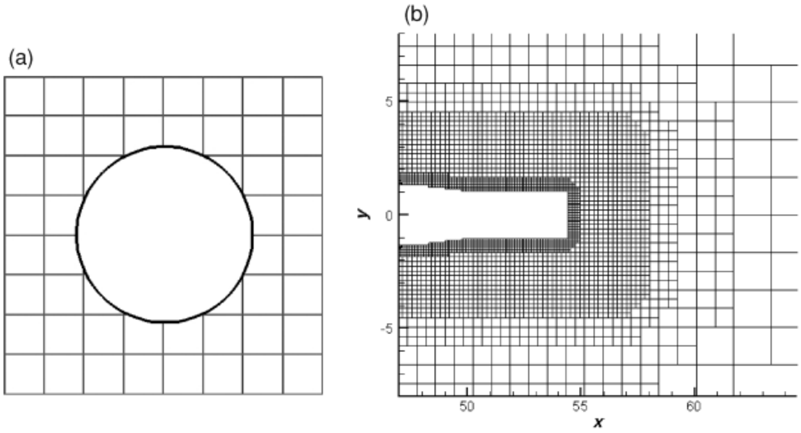

Figure 8.7 Cartesian grids in 2-D: (a) structured Cartesian and (b) unstructured Cartesian (Figure 8.7b is reproduced from Roy *et al.*, 2007).

where the underlying mesh is unstructured is given in Figure 8.7b. The unstructured nature of the grid allows clustering by subdividing selected cells, thus allowing more resolution of the surface geometry and/or solution features.

### *8.1.2.4 Mesh-free methods*

There are a number of mesh-free methods in the area of fluid mechanics which generally do not require a volume mesh in the domain, but may require surface meshes. These approaches thus have the advantage of greatly simplifying the grid generation process by reducing the dimensionality of the mesh to be generated by one. These methods take a Lagrangian approach, thus allowing individual vortices or fluid particles to travel through the domain and interact with each other. Some examples of grid-free methods include vortex methods, smoothed particle hydrodynamics, and lattice Boltzmann. The actual form of the underlying governing equations and the consistency/convergence of these methods with increasing number of vortices, particles, etc. has generally not been adequately addressed.

## 8.2 Approaches for estimating discretization error

There are many approaches available for estimating discretization error. These methods can be broadly categorized as *a priori* methods and *a posteriori* methods. The *a priori* methods are those that allow a bound to be placed on the discretization error before any numerical solution is even computed. In general, one looks to bound the discretization error by an equation of the form

$$\varepsilon_h \leq C(u)h^p, \tag{8.20}$$

where $\varepsilon_h$ is the discretization error, the function $C(u)$ usually depends on various derivatives of the exact solution, $h$ is a measure of the element size (e.g., $\Delta x$), and $p$ is the formal order of accuracy of the method. One approach to developing an *a priori* discretization error estimator is to perform a truncation error analysis for the scheme, relate the truncation error to the discretization error (as in Chapter 5), then develop some approximate bounds on the solution derivatives that comprise $C(u)$. The main failing of *a priori* error estimators is that $C(u)$ is extremely difficult to bound and, even when this is possible for simple problems, the resulting error estimate greatly over-estimates the true discretization error. *A priori* methods are generally only useful for assessing the formal order of accuracy of a discretization scheme. Current efforts in estimating the discretization error are focused on *a posteriori* methods. These methods provide an error estimate only after the numerical solution has been computed. They use the computed solution to the discrete equations (possibly including information from the problem being solved) to estimate the discretization error relative to the exact solution to the mathematical model.

The mathematical formalism that underlies the finite element method makes it fertile ground for the rigorous estimation of discretization error. Beginning with the pioneering work of Babuska and Rheinboldt (1978a), a tremendous amount of work has been done over the last 30 years on *a posteriori* estimation of discretization error by the finite element community (Ainsworth and Oden, 2000). The initial developments up to the early 1990s were concentrated on linear, elliptic, scalar mathematical models and focused on the $h$-version of finite elements. Early extensions of the *a posteriori* methods to parabolic and hyperbolic mathematical models were done by Eriksson and Johnson (1987) and Johnson and Hansbo (1992), respectively. Up to this point, *a posteriori* error estimation in finite elements was limited to analysis of the energy norm of the discretization error, which for

Poisson's equation can be written on element $k$ as:

$$|||\varepsilon_h|||_k = \left[ \int\limits_{\Omega_k} \left| \vec{\nabla} u_h - \vec{\nabla} \tilde{u} \right|^2 \mathrm{d}\Omega \right]^{1/2}. \tag{8.21}$$

The finite element method produces the numerical solution from the chosen set of basis functions that minimizes the energy norm of the discretization error (Szabo and Babuska, 1991). Taken locally, the energy norm can provide guidance on where adaptive refinement should occur. Taken globally, the energy norm provides a global measure of the overall optimality of the finite element solution. In the early 1990s, important extensions of *a posteriori* error estimators to system response quantities were found that require the solution to an adjoint, or dual, problem (e.g., Johnson and Hansbo, 1992). For additional information on *a posteriori* error estimation in finite element methods, see Babuska *et al.* (1986), Whiteman (1994), Ainsworth and Oden (1997, 2000), and Estep *et al.* (2000). A more introductory discussion of error estimation in finite element analysis is presented by Akin (2005).

In general, the level of reliability of *a posteriori* error estimation methods is strongly problem dependent. All of the discretization error estimators to be discussed here were originally developed for elliptic problems. As a result, they tend to work well for elliptic problems, but are not as well-developed for mathematical models that are parabolic or hyperbolic in nature. The level of complexity of the problem is also an important issue. The error estimators work well for smooth, linear problems with simple physics and geometries; however, strong nonlinearities, discontinuities, singularities, and physical and geometric complexity can significantly reduce the reliability and applicability of *a posteriori* discretization error estimation methods.

One approach for measuring the accuracy and reliability of a discretization error estimator is through the effectivity index $\theta$. Here we will define the *effectivity index* as a general norm of the estimated discretization error $\bar{\varepsilon}_h$ divided by the norm of the true error $\varepsilon_h$. The effectivity index can be computed locally by evaluating it on element or cell $k$,

$$\theta_k = \frac{\|\bar{\varepsilon}_h\|_k}{\|\varepsilon_h\|_k}, \tag{8.22}$$

or globally over the entire domain $\Omega$,

$$\theta = \frac{\|\bar{\varepsilon}_h\|_\Omega}{\|\varepsilon_h\|_\Omega}. \tag{8.23}$$

One important property that is desired in a discretization error estimator is consistency (i.e., asymptotic exactness) of the estimate, which means the estimated error must approach the true error (i.e., $\theta \rightarrow 1$) as the mesh is refined ($h \rightarrow 0$) or as the order of the scheme is increased ($p \rightarrow \infty$). The practical utility of a given error estimator can also be gauged by examining its effectivity index on coarse meshes. Finally, if asymptotic exactness of the discretization error estimator cannot be shown, then we prefer to have error estimates that are conservative (i.e., they over-estimate the error). Thus a discretization error estimator that over-estimates the error is more desirable than one which under-estimates it.

There are two types of discretization error estimator discussed in this section. In the first type, an estimate of the exact solution to the mathematical model (or possibly its gradient) is obtained that is of higher formal order of accuracy than the underlying solution. This higher-order estimate relies only on information from the discrete solution itself, and thus can often be applied in a post-processing manner. For mesh and order refinement methods, higher-order estimates can be easily obtained for system response quantities as well. Residual-based methods, by contrast, also incorporate information on the specific problem being solved into the error estimate. While their implementation in a scientific computing code is generally more difficult and code-intrusive, they have the potential to provide more detailed information on the discretization error and its various sources. The extension of residual-based methods to provide discretization error estimates in system response quantities generally requires the solution to an adjoint (or dual) problem.

### 8.2.1 Type I: Higher-order estimates

One approach to error estimation is to compare the discrete solution to a higher-order estimate of the exact solution to the mathematical model. While this approach uses only information from the discrete solution itself, in some cases, more than one discrete solution is needed, with the additional solutions being obtained either on systematically-refined/coarsened meshes or with different formal orders of accuracy.

#### 8.2.1.1 Mesh refinement methods

Mesh refinement methods are based on the general concept of Richardson extrapolation (Richardson 1911, 1927). Recall that in Chapter 5 we used a power series expansion to relate the discretization error in a system response quantity $\varepsilon_h = f_h - \tilde{f}$ to the mesh spacing parameter $h$, which for a convergent, $p$th-order discretization can be written as

$$f_h - \tilde{f} = g_p h^p + O(h^{p+1}), \tag{8.24}$$

where $f_h$ is the discrete solution and $\tilde{f}$ the exact solution to the mathematical model. The basic premise behind mesh refinement methods is to compute two discrete solutions on systematically-refined meshes, then use Eq. (8.24) to solve for an estimate of the exact solution to the mathematical model $\bar{f}$. As with all discretization error estimators, the estimate will only be reliable in the asymptotic range where the terms of order $h^{p+1}$ and higher can be neglected. In the asymptotic range, the estimate $\bar{f}$ is generally accurate to within order $p+1$. This higher-order estimate of $\tilde{f}$ can then be used to estimate the error in the discrete solutions. The main advantages of mesh refinement-based error estimators are that they can be applied as a post-processing step to any type of discretization and that they can be applied both to the solution and to any system response quantities. A more detailed discussion of Richardson extrapolation and its use as a discretization error estimator is presented in Section 8.3.

#### *8.2.1.2 Order refinement methods*

Order refinement methods are those which employ two or more discretizations on the same mesh but with differing formal orders of accuracy. The results from the two numerical solutions are then combined to produce a discretization error estimate. An early example of order refinement methods for error estimation is the Runge-Kutta–Fehlberg method (Fehlberg, 1969) for adaptive step size control in the solution of ordinary differential equations. This approach combines a basic fourth-order Runge-Kutta integration of the differential equations with an inexpensive fifth-order estimate of the error. Order refinement can be difficult to implement in finite difference and finite volume discretizations due to difficulties formulating higher-order accurate gradients and boundary conditions. Order refinement methods have been implemented within the context of finite elements under the name hierarchical bases (e.g., see Bank, 1996).

#### *8.2.1.3 Finite element recovery methods*

Recovery methods for estimating the discretization error were developed by the finite element community (e.g., Zienkiewicz and Zhu, 1987, 1992). For the standard *h*-version of finite elements with linear basis functions the solution is piece-wise linear; therefore, the gradients are only piece-wise constant and are discontinuous across the element faces. The user of a finite element code is often more interested in gradient quantities such as stresses than the solution itself, so most finite element codes provide for post-processing of these discontinuous gradients into piece-wise linear gradients using existing finite element infrastructure.

In some cases (see the discussion of superconvergence below), this reconstructed gradient is of a higher order of accuracy than the gradient found in the underlying finite element solution. Recall the definition of the energy norm of the discretization error given in Eq. (8.21). If the true gradient from the mathematical model is available, then this important error measure can be computed exactly. For the case where a reconstructed gradient is higher-order accurate than the finite element gradient, it can be used to approximate the true gradient in the energy norm. In addition to providing estimates of the discretization error in the solution gradients, due to their local nature, recovery methods are also often used as indicators of where solution refinement is needed in adaptive solutions (see Chapter 9).

In order to justify the use of the recovered gradient in the energy norm, it must in fact be higher-order accurate than the gradient from the finite element solution. This so-called *superconvergence property* can occur when certain regularity conditions on the mesh and the solution are met (Wahlbin, 1995) and results in gradients that are up to one order higher in accuracy than the underlying finite element gradients. For linear finite elements, the superconvergence points occur at the element centroids, whereas for quadratic finite elements, the location of the superconvergence points depends on the element topology. If the reconstructed gradient is superconvergent, and if certain consistency conditions are met by the gradient reconstruction operator itself, then error estimators based on this recovered gradient can be shown to be asymptotically exact with effectivity indices approaching unity (Ainsworth and Oden, 2000). While the superconvergence property appears to be a difficult one to attain for complex scientific computing applications, the discretization error estimates from some recovery methods tend to be "astonishingly good" for reasons that are not well-understood (Ainsworth and Oden, 2000).

Recovery methods have been shown to be most effective when the reconstruction step employs solution gradients rather than solution values. The *superconvergent patch recovery* (SPR) method (Zienkiewicz and Zhu, 1992) is the most widely-used recovery method in finite element analysis. Assuming the underlying finite element method is of order $p$, the SPR approach is based on a local least squares fitting of the solution gradient values at the superconvergence points using polynomials of degree $p$. The SPR recovery method was found to perform extremely well in an extensive comparison of *a posteriori* finite element error estimators (Babuska *et al.*, 1994). A more recent approach called *polynomial preserving recovery* (PPR) was proposed by Zhang and Naga (2005). In their approach, they use polynomials of degree $p+1$ to fit the solution values at the superconvergence points, then take derivatives of this fit to recover the gradient. Both the SPR and PPR gradient reconstruction methods can be used to obtain error estimates in the global energy norm and in the local solution gradients. Extensions to system response quantities must be done heuristically. For example, a 5% error in the global energy norm may be found to correspond to a 10% error in a system response quantity for a given class of problems.

### 8.2.2 Type II: Residual-based methods

Residual-based methods use the discrete solution along with additional information from the problem being solved such as the mathematical model, the discrete equations, or the sources of discretization error. Examples of residual-based methods are error transport equations (both continuous and discrete) and finite element residual methods. As is shown in the next section, all of these residual-based methods are related through the truncation error. The truncation error can be approximated either by inserting the exact solution to the mathematical model (or an approximation thereof) into the discrete equation or by inserting the discrete solution into the continuous mathematical model. The former is the discrete residual which is used in most discretization error transport equations. The latter is simply the definition of the finite element residual. The use of adjoint methods to extend Type II (residual-based) discretization error estimation methods to provide error estimates in system response quantities is also discussed.

#### 8.2.2.1 Error transport equations

Discretization error is transported through the domain in a similar fashion as the solution to the underlying mathematical model (Ferziger and Peric, 2002). For example, if a mathematical model governs the convection and diffusion of a scalar variable, then a discrete solution to the mathematical model will contain discretization error that is also convected and diffused. Babuska and Rheinboldt (1978a) appear to be the first to develop such a discretization error transport equation within the context of the finite element method. However, rather than solve this transport equation directly for the discretization error, the typical approach used in finite elements is to use this equation to either indirectly bound the error (explicit residual methods) or approximate its solution (implicit residual methods). Examples of the use of error transport equations for finite volume schemes can be found in Zhang *et al.* (2000), Cavallo and Sinha (2007), and Shih and Williams (2009).

### Continuous discretization error transport equation

The following development from Roy (2009) is applicable to any discretization approach and is based on the *generalized truncation error expression* developed in Chapter 5 as Eq. (5.12). Recall that the original (possibly nonlinear) governing equation operator $L(\cdot)$ and the discrete equation operator $L_h(\cdot)$ are exactly solved by $\tilde{u}$ (the exact solution to the original mathematical model) and $u_h$ (the exact solution to the discrete equations), respectively. Thus we can write:

$$L(\tilde{u}) = 0 \tag{8.25}$$

and

$$L_h(u_h) = 0. \tag{8.26}$$

Furthermore, the partial differential equation and the discretized equation are related through the *generalized truncation error expression* (Eq. (5.12), repeated here for convenience) as

$$L_h(u) = L(u) + TE_h(u), \tag{8.27}$$

which assumes some suitable mapping of the operators onto either a continuous or discrete space. Substituting $u_h$ into Eq. (8.27) and then subtracting Eq. (8.25) gives

$$L(u_h) - L(\tilde{u}) + TE_h(u_h) = 0. \tag{8.28}$$

If the equations are linear, or if they are linearized, then we have $L(u_h) - L(\tilde{u}) = L(u_h - \tilde{u})$. With the definition of the discretization error

$$\varepsilon_h = u_h - \tilde{u} \tag{8.29}$$

we can thus rewrite Eq. (8.28) as

$$L(\varepsilon_h) = -TE_h(u_h). \tag{8.30}$$

Equation (8.30) is the mathematical model that governs the transport of the discretization error $\varepsilon_h$ through the domain. Furthermore, the truncation error acting upon the discrete solution serves as a source term which governs the local generation or removal of discretization error and is a function of the local discretization parameters ($\Delta x$, $\Delta y$, etc.). Equation (8.30) is called the *continuous discretization error transport equation*. This equation can be discretized and solved for the discretization error in the solution variables assuming that the truncation error is known or can be estimated.

### Discrete discretization error transport equation

A discrete version of the discretization error transport equation can be derived as follows. First the exact solution to the mathematical model $\tilde{u}$ is substituted into Eq. (8.27) and then Eq. (8.26) is subtracted to obtain

$$L_h(u_h) - L_h(\tilde{u}) + TE_h(\tilde{u}) = 0. \tag{8.31}$$

If the equations are again linear (or linearized), then this equation can be rewritten as

$$L_h(\varepsilon_h) = -TE_h(\tilde{u}). \tag{8.32}$$

Equation (8.32) is the discrete equation that governs the transport of the discretization error $\varepsilon_h$ through the domain and is therefore called the *discrete discretization error transport equation*. This equation can be solved for the discretization error if the truncation error and the exact solution to the original partial differential equation (or a suitable approximation of it) are known.

### Approximating the truncation error

While the development of discretization error transport equations is relatively straightforward, questions remain as to the treatment of the truncation error, which acts as the local source term. The truncation error can be difficult to derive for complex, nonlinear discretization schemes such as those used for the solution to the compressible Euler equations in fluid dynamics. However, if the truncation error can be reliably approximated, then this approximation can be used as the source term for the error transport equation.

Here we present three approaches for approximating the truncation error, with the first two approaches beginning with the *generalized truncation error expression* given by Eq. (8.27) (Roy, 2009). In the first approach, the exact solution to the mathematical model $\tilde{u}$ is inserted into Eq. (8.27). Since this exact solution will exactly solve the mathematical model, the term $L(\tilde{u}) = 0$, thus allowing the truncation error to be approximated as

$$TE_h(\tilde{u}) = L_h(\tilde{u}). \tag{8.33}$$

Since this exact solution is generally not known, it could be approximated by plugging an estimate of the exact solution, for example from Richardson extrapolation or any other discretization error estimator, into the discrete operator:

$$TE_h(u_{\mathrm{RE}}) \approx L_h(u_{\mathrm{RE}}). \tag{8.34}$$

Alternatively, the solution from a fine grid solution $u_h$ could be inserted into the discrete operator for a coarse grid $L_{rh}(\cdot)$:

$$TE_h(\tilde{u}) = \frac{1}{r^p} TE_{rh}(\tilde{u}) \approx \frac{1}{r^p} TE_{rh}(u_h) = \frac{1}{r^p} L_{rh}(u_h). \tag{8.35}$$

Note that the subscript *rh* denotes the discrete operator on a grid that is a factor of *r* coarser in each direction than the fine grid. For example, $r = 2$ when the coarse mesh is formed by eliminating every other point in each direction of a structured mesh. This approach was used by Shih and Qin (2007) to estimate the truncation error for use with a discrete discretization error transport equation.

A second approach for estimating the truncation error is to insert the exact solution to the discrete equations $u_h$ into Eq. (8.27). Since this solution exactly solves the discrete equations $L_h(u_h) = 0$, we have

$$TE_h(u_h) = -L(u_h). \tag{8.36}$$

If a continuous representation of the solution is available then this evaluation is straightforward. In fact, the right hand side of Eq. (8.36) is the definition of the finite element residual that is given in the next section. For other numerical methods (e.g., finite difference and finite volume), a continuous projection of the numerical solution must be made in order to estimate the truncation error. For example, Sonar (1993) formed this residual by projecting a finite volume solution onto a finite element subspace with piece-wise linear shape functions.

A third approach that is popular for hyperbolic problems (e.g., compressible flows) is based on the fact that central-type differencing schemes often require additional numerical (artificial) dissipation to maintain stability and to prevent numerical oscillations. This numerical dissipation can either be explicitly added to a central differencing scheme (e.g., see Jameson *et al.*, 1981) or incorporated as part of an upwind differencing scheme. In fact, it can be shown that any upwind scheme can be written as a central scheme plus a numerical dissipation term (e.g., Hirsch, 1990). These two approaches can thus be viewed in the context of central schemes with the numerical dissipation contributions serving as the leading terms in the truncation error. While this approach may only be a loose approximation of the true truncation error, it merits discussion due to the fact that it can be readily computed with little additional effort.

### System response quantities

A drawback to the error transport equation approach is that it provides for discretization error estimates in the local solution variables, but not in system response quantities. While adjoint methods can be used to provide error estimates in the system response quantities (see Section 8.2.2.3), Cavallo and Sinha (2007) have developed a simpler approach that uses an analogy with experimental uncertainty propagation to relate the local solution errors to the error in the system response quantity. However, their approach appears to provide overly-conservative error bounds for integrated quantities since it does not allow for the cancellation of competing errors. An alternative approach would be to use the local error estimates to correct the local quantities, then compute the integrated quantity with these corrected values. This "corrected" integrated quantity could then be used to provide the desired discretization error estimate.

#### 8.2.2.2 *Finite element residual methods*

In a broad mathematical sense, a residual refers to what is left over when an approximate solution is inserted into an equation. In Chapter 5 we discussed the residual method for code order of accuracy verification where the residual was found by substituting the exact solution to the mathematical model into the discrete equations. In Chapter 7 we discussed iterative convergence in terms of the iterative residuals which are found by substituting an approximate iterative solution into the discrete equations. Consider now the general mathematical operator $L(\tilde{u}) = 0$ which is solved exactly by $\tilde{u}$. Because the finite element method provides for a continuous representation of the numerical solution $u_h$, it is natural to define the finite element residual in a continuous sense over the domain as

$$\Re(u_h) = L(u_h). \tag{8.37}$$

In a manner analogous to the development of the previous section, a continuous discretization error transport equation can be derived within the finite element framework (Babuska and Rheinboldt, 1978a). This so-called residual equation has three different types of term: (1) interior residuals that determine how well the finite element solution satisfies the mathematical model in the domain, (2) terms associated with any discretized boundary conditions on the domain boundary (e.g., Neumann boundary conditions), and (3) interelement residuals which are functions of the discontinuities in normal fluxes across element–element boundaries (Ainsworth and Oden, 2000). It is the treatment of these three terms that differentiates between explicit and implicit residual methods.

## Explicit residual methods

*Explicit residual methods* are those which employ information available from the finite element solution along with the finite element residuals to directly compute the error estimate. First developed by Babuska and Rheinboldt (1978b), explicit residual methods lump all three types of residual terms under a single, unknown constant. The analysis requires the use of the triangle inequality, which does not allow for cancellation between the different residual types. Due both to the use of the triangle inequality and the methods for estimating the unknown constant, explicit residual methods tend to be conservative estimates of the discretization error. They provide an element-wise estimate of the *local contribution* to the bound for the global energy norm of the error, but not a local estimate of the true error, which would include both local and transported components. Since explicit residual methods deal only with local contributions to the error, they can also be used for solution adaptation procedures. Stewart and Hughes (1998) have provided a tutorial on explicit residual methods and also discuss their relationship to *a priori* error estimation.

## Implicit residual methods

*Implicit residual methods* avoid the approximations required in explicit residual methods by seeking solutions to the residual equation which govern the transport and generation of the discretization error. In order to achieve nontrivial solutions to the global residual equation, either the mesh would have to be refined or the order of the finite element basis functions increased. Both of these approaches would be significantly more expensive than obtaining the original finite element solution and therefore are not considered practical. Instead, the global residual equation is decomposed into a series of uncoupled, local boundary value problems approximating the global equation. These local problems can be solved over a single element using the element residual method (Demkowicz *et al.*, 1984; Bank and Weiser, 1985) or over a small patch of elements using the subdomain residual method (Babuska and Rheinboldt, 1978a,b). The solution to the local boundary value problems provides the local discretization error estimate, while the global error estimate is simply summed over the domain. By directly treating all three types of terms that show up in the residual equation, implicit residual methods retain more of the structure of the residual equation than do the explicit methods, and thus should in theory provide tighter discretization error bounds.

#### *8.2.2.3 Adjoint methods for system response quantities*

Both error transport equations and finite element residual methods give localized estimates of the discretization error, which can then be combined through an appropriate norm to provide quantitative measures of the overall “goodness” of the discrete solutions. However, the scientific computing practitioner is often instead interested in system response quantities that can be post-processed from the solution. These system response quantities can take the form of integrated quantities (e.g., net flux through or force acting on a boundary), local solution quantities (e.g., maximum stress or maximum temperature), or even an average of the solution over some region.

Adjoint methods in scientific computing were initially used for design optimization problems (e.g., Jameson, 1988). In the optimization setting, the adjoint (or dual) problem can be solved for sensitivities of a solution functional (e.g., a system response quantity) that one wishes to optimize relative to some chosen design parameters. The strength of the adjoint method is that it is efficient even when a large number of design parameters are involved. In the context of optimization in scientific computing, adjoint methods can be thought of as constrained optimization problems where a chosen solution functional is to be optimized subject to the constraint that the solution must also satisfy the mathematical model (or possibly the discrete equations).

Adjoint methods can also be used for estimating the discretization error in a system response quantity in scientific computing applications. Consider a scalar solution functional $f_h(u_h)$ evaluated on mesh $h$. An approximation of the discretization error in this functional is given by

$$\varepsilon_h = f_h(u_h) - f_h(\tilde{u}). \tag{8.38}$$

Performing a Taylor series expansion of $f_h(\tilde{u})$ about the discrete solution gives

$$f_h(\tilde{u}) \cong f_h(u_h) + \left.\frac{\partial f_h}{\partial u}\right|_{u_h} (\tilde{u} - u_h), \tag{8.39}$$

where higher order terms have been neglected. Next, an expansion of the discrete operator $L_h(\cdot)$ is performed at $\tilde{u}$ about $u_h$:

$$L_h(\tilde{u}) \cong L_h(u_h) + \left.\frac{\partial L_h}{\partial u}\right|_{u_h} (\tilde{u} - u_h), \tag{8.40}$$

where $L_h(\tilde{u})$ is the discrete residual, an approximation of the truncation error from Eq. (8.33), and $\left.\frac{\partial L_h}{\partial u}\right|_{u_h}$ is the Jacobian which linearizes the discrete equations with respect to the solution. This Jacobian may already be computed since it can also be used to formulate implicit solutions to the discrete equations and for design optimization. Since $L_h(u_h) = 0$, Eq. (8.40) can be rearranged to obtain

$$(\tilde{u} - u_h) = \left[ \left.\frac{\partial L_h}{\partial u}\right|_{u_h} \right]^{-1} L_h(\tilde{u}). \tag{8.41}$$

Substituting this equation into Eq. (8.39) gives

$$f_h(\tilde{u}) \cong f_h(u_h) + \left.\frac{\partial f_h}{\partial u}\right|_{u_h} \left[ \left.\frac{\partial L_h}{\partial u}\right|_{u_h} \right]^{-1} L_h(\tilde{u}) \tag{8.42}$$

or

$$f_h(\tilde{u}) \cong f_h(u_h) + \Psi^{\mathrm{T}} L_h(\tilde{u}), \tag{8.43}$$

where $\Psi^{\mathrm{T}}$ is the row vector of discrete adjoint sensitivities. The adjoint sensitivities are found by solving

$$\Psi^{\mathrm{T}} = \left.\frac{\partial f_h}{\partial u}\right|_{u_h} \left[ \left.\frac{\partial L_h}{\partial u}\right|_{u_h} \right]^{-1}, \tag{8.44}$$

which can be put into the standard linear equation form by transposing both sides of Eq. (8.44)

$$\left[ \left.\frac{\partial L_h}{\partial u}\right|_{u_h} \right]^{\mathrm{T}} \Psi = \left[ \left.\frac{\partial f_h}{\partial u}\right|_{u_h} \right]^{\mathrm{T}}. \tag{8.45}$$

The adjoint solution provides the linearized sensitivities of the solution functional $f_h$ to perturbations in the discrete operator $L_h(\cdot)$. As such, the adjoint solution vector components are often referred to as the adjoint sensitivities. Equation (8.43) shows that the adjoint solution provides the sensitivity of the discretization error in the solution functional $f(\cdot)$ to the local sources of discretization error (i.e., the truncation error) in the domain. This observation will be used in Chapter 9 as the basis for providing solution adaptation targeted for solution functionals. Because the discrete operator $L_h(\cdot)$ is used above, this approach is called the *discrete adjoint method*. A similar analysis using expansions of the continuous mathematical operator $L(\cdot)$ and functional $f(\cdot)$ can be performed to obtain discretization error estimates using the *continuous adjoint method*. Both continuous and discrete adjoint methods also require appropriate formulations of initial and boundary conditions.

### Adjoint methods in the finite element method

While the use of explicit and implicit residual methods for finite elements has reached a certain level of maturity for elliptic problems (Ainsworth and Oden, 2000), the drawback to these methods is that they only provide error estimates in the energy norm of the discretization error. While the energy norm is a natural quantity by which to judge the overall goodness of a finite element solution, in many cases scientific computing is used to make an engineering decision with regards to a specific system response quantity (called "quantities of interest" by the finite element community). Extension of both the explicit and implicit residual methods to provide error estimates in a system response quantity generally requires the solution to the adjoint system.

In one approach (Ainsworth and Oden, 2000), the discretization error in system response quantity is bounded by the product of the energy norm of the adjoint solution and the energy norm of the error in the original solution. Assuming the solutions are asymptotic, the use of the Cauchy–Schwarz inequality produces a conservative bound. In this case, the discretization error in the system response quantity will be reduced at twice the rate of the solution error. In another approach (Estep *et al.*, 2000), the error estimate in the system response quantity for a class of reaction–diffusion problems is found as an inner product between the adjoint solution and the residual. This approach results in a more accurate (i.e., less conservative) error estimate at the expense of losing the rigorous error bound. For more information on error estimation using adjoint methods in finite elements, see Johnson and Hansbo (1992), Paraschivoiu *et al.* (1997), Rannacher and Suttmeier (1997), Estep *et al.* (2000), and Cheng and Paraschivoiu (2004).

### Adjoint methods in the finite volume method

Pierce and Giles (2000) have proposed a continuous adjoint approach that focuses on system response quantities (e.g., lift and drag in aerodynamics problems) and is not tied to a specific discretization scheme. They use the adjoint solution to relate the residual error in the mathematical model to the resulting error in the integral quantity of interest. Their approach also includes a defect correction step that increases the order of accuracy of the integral quantity. For example, if the original solution and the adjoint solution are both second-order accurate, then the corrected integral quantity will have an order of accuracy equal to the product of the orders of the original and adjoint solutions, or fourth order. Their approach effectively extends the superconvergence property of finite elements to other discretization schemes, and can also be used to further increase the order of accuracy of the integral quantities for the finite element method.

Venditti and Darmofal (2000) have extended the adjoint approach of Pierce and Giles (2000) to allow for the estimation of local mesh size contributions to the integral quantity of interest. Their approach is similar to that described earlier in this section, but expands the functional and discrete operator on a fine grid solution $u_h$ about a coarse grid solution $u_{rh}$. The solution is not required on the fine grid, only residual evaluations. Their approach is thus a discrete adjoint method rather than continuous adjoint. In addition, their focus is on developing techniques for driving a mesh adaptation process (see Chapter 9). Their initial formulation was applied to 1-D inviscid flow problems, but they have also extended their approach to 2-D inviscid and viscous flows (Venditti and Darmofal, 2002, 2003). While adjoint methods hold significant promise as discretization error estimators for solution functionals, they currently require significant code modifications to compute the Jacobian and other sensitivity derivatives and have not yet seen widespread use in commercial scientific computing software.

## 8.3 Richardson extrapolation

The basic concept behind Richardson extrapolation (Richardson, 1911, 1927) is as follows. If one knows the formal rate of convergence of a discretization method with mesh refinement, and if discrete solutions on two systematically refined meshes are available, then one can use this information to obtain an estimate of the exact solution to the mathematical model. Depending on the level of confidence one has in this estimate, it can be used either to correct the fine mesh solution or to provide a discretization error estimate for it. While Richardson's original work applied the approach locally over the domain to the dependent variables in the mathematical model, it can be readily applied to any system response quantity. There is, however, the additional requirement that the numerical approximations (integration, differentiation, etc.) used to obtain the system response quantity be at least of the same order of accuracy as the underlying discrete solutions.

Recall the definition of the discretization error in some general local or global solution variable $f$ on a mesh with spacing $h$,

$$\varepsilon_h = f_h - \tilde{f}, \tag{8.46}$$

where $f_h$ is the exact solution to the discrete equations and $\tilde{f}$ is the exact solution to the original partial differential equations. We can expand the numerical solution $f_h$ in either a Taylor series about the exact solution,

$$f_h = \tilde{f} + \frac{\partial \tilde{f}}{\partial h} h + \frac{\partial^2 \tilde{f}}{\partial h^2}\frac{h^2}{2} + \frac{\partial^3 \tilde{f}}{\partial h^3}\frac{h^3}{6} + O(h^4), \tag{8.47}$$

or simply a power series in $h$,

$$f_h = \tilde{f} + g_1 h + g_2 h^2 + g_3 h^3 + O(h^4), \tag{8.48}$$

where $O(h^4)$ denotes a leading error term on the order of $h$ to the fourth power. Moving $\tilde{f}$ to the left hand side allows us to write the discretization error for a mesh with spacing $h$ as

$$\varepsilon_h = f_h - \tilde{f} = g_1 h + g_2 h^2 + g_3 h^3 + O(h^4), \tag{8.49}$$

where the $g$ coefficients can take the form of derivatives of the exact solution to the mathematical model $\tilde{f}$ with respect to either the mesh size $h$ (as shown in Eq. (8.47)) or to the independent variables through the relationship with the truncation error (see Section 5.3.1). In general, we require numerical methods that are higher than first-order accurate, and thus discretization methods are chosen which cancel out selected lower-order terms. For example, if a formally second-order accurate numerical scheme is chosen, then the general discretization error expansion becomes

$$\varepsilon_h = f_h - \tilde{f} = g_2 h^2 + g_3 h^3 + O(h^4). \tag{8.50}$$

Equation (8.50) forms the basis for standard Richardson extrapolation, which is described next.

### *8.3.1 Standard Richardson extrapolation*

The *standard Richardson extrapolation* procedure, as originally formulated by Richardson (1911), applies exclusively to cases where the numerical scheme is formally second-order accurate and the mesh is systematically refined (or coarsened) by a factor of two. Consider a second-order discretization scheme that is used to produce numerical solutions on two meshes: a fine mesh with spacing $h$ and a coarse mesh with spacing $2h$. Since the scheme is second-order accurate, the $g_1$ coefficient is zero and the discretization error equation (Eq. (8.50)) can be rewritten as

$$f_h = \tilde{f} + g_2 h^2 + g_3 h^3 + O(h^4). \tag{8.51}$$

Applying this equation on two mesh levels $h$ and $2h$ gives

$$\begin{aligned} f_h &= \tilde{f} + g_2 h^2 + g_3 h^3 + O(h^4), \\ f_{2h} &= \tilde{f} + g_2 (2h)^2 + g_3 (2h)^3 + O(h^4). \end{aligned} \tag{8.52}$$

Eliminating $\tilde{f}$ from these equations and solving for $g_2$ yields

$$g_2 = \frac{f_{2h} - f_h}{3h^2} - \frac{7}{3} g_3 h + O(h^2). \tag{8.53}$$

Substituting Eq. (8.53) into the fine grid expansion of Eq. (8.52) and solving for $\tilde{f}$ gives

$$\tilde{f} = f_h + \frac{f_h - f_{2h}}{3} + \frac{4}{3} g_3 h^3 + O(h^4). \tag{8.54}$$

Combining the terms of order $h^3$ and higher in Eq. (8.54) with the exact solution yields

$$\bar{f} = \tilde{f} - \frac{4}{3} g_3 h^3 + O(h^4). \tag{8.55}$$

The Richardson extrapolated estimate can thus be found by inserting Eq. (8.55) in Eq. (8.54) to give

$$\bar{f} = f_h + \frac{f_h - f_{2h}}{3}. \tag{8.56}$$

Equation (8.56) is the standard Richardson extrapolation relationship, and provides an estimate $\bar{f}$ of the exact solution $\tilde{f}$ which is higher-order accurate than the underlying numerical scheme. This estimate will therefore converge to the exact solution faster than the numerical solutions themselves as the mesh is refined. A discretization error estimate derived from the Richardson extrapolation procedure will therefore be asymptotically consistent.

There is often confusion as to the order of accuracy of the Richardson extrapolation estimate. As shown in Eq. (8.55), the estimates from standard Richardson extrapolation are generally third-order accurate. In Richardson's original work (Richardson, 1911), he used this extrapolation procedure to obtain a higher-order accurate solution for the stresses in a masonry dam based on two second-order accurate numerical solutions. The original partial differential equation was Poisson's equation and he employed central differences which cancelled out the odd powers in the truncation error (i.e., $g_3 = 0$). His estimate for the exact solution was thus fourth-order accurate, as is clearly shown in Eq. (8.55) when $g_3 = 0$.

### 8.3.2 Generalized Richardson extrapolation

Richardson extrapolation can be generalized to $p$th-order accurate schemes and for two meshes systematically refined by an arbitrary factor. First consider the general discretization error expansion for a $p$th-order scheme:

$$\varepsilon_h = f_h - \tilde{f} = g_p h^p + g_{p+1} h^{p+1} + g_{p+2} h^{p+2} + \cdots. \tag{8.57}$$

Introducing the grid refinement factor as the ratio of the coarse to fine grid spacing, we have

$$r = \frac{h_{\text{coarse}}}{h_{\text{fine}}} > 1 \tag{8.58}$$

and the coarse grid spacing can thus be written as $h_{\text{coarse}} = r h_{\text{fine}}$. Choosing $h_{\text{fine}} = h$, the discretization error equations on the two meshes can be written as

$$\begin{aligned} f_h &= \tilde{f} + g_p h^p + g_{p+1} h^{p+1} + O(h^{p+2}), \\ f_{rh} &= \tilde{f} + g_p (rh)^p + g_{p+1} (rh)^{p+1} + O(h^{p+2}). \end{aligned} \tag{8.59}$$

As before, these equations can be used to eliminate the $g_p$ coefficient and solve for $\tilde{f}$ to give

$$\tilde{f} = f_h + \frac{f_h - f_{rh}}{r^p - 1} + g_{p+1} h^{p+1} \frac{r^p (r-1)}{r^p - 1} + O(h^{p+2}). \tag{8.60}$$

Again, combining terms of order $h^{p+1}$ and higher with the exact solution $\tilde{f}$ gives

$$\bar{f} = \tilde{f} - g_{p+1} \frac{r^p (r-1)}{r^p - 1} h^{p+1} + O(h^{p+2}). \tag{8.61}$$

Substituting this expression into Eq. (8.60) results in the generalized Richardson extrapolation estimate $\bar{f}$ :

$$\bar{f} = f_h + \frac{f_h - f_{rh}}{r^p - 1}. \tag{8.62}$$

As is shown clearly by Eq. (8.61), this estimate of the exact solution is generally $(p+1)$-order accurate estimate of the exact solution to the mathematical model $\tilde{f}$ unless additional error cancellation occurs in the higher-order terms (e.g., if the $g_{\mathrm{p}+1}$ coefficient is zero).

### 8.3.3 Assumptions

There are five basic assumptions required for Richardson extrapolation to provide reliable estimates of the exact solution to the mathematical model: (1) that *both* discrete solutions are in the asymptotic range, (2) that the meshes have a uniform (Cartesian) spacing over the domain, (3) that the coarse and fine meshes are related through systematic refinement, (4) that the solutions are smooth, and (5) that the other sources of numerical error are small. These five assumptions are each discussed in detail below.

#### 8.3.3.1 Asymptotic range

The formal order of accuracy of a discretization scheme is the theoretical rate at which the discretization error is reduced as the mesh is systematically refined. Recall that in Chapter 5 we used a continuous discretization error transport equation to relate the formal order of accuracy to the lowest order term in the truncation error. This lowest order term will necessarily dominate the higher-order terms in the limit as the mesh spacing parameter $h$ goes to zero. The dependent solution variables generally converge at the formal order of accuracy in the asymptotic range, as do any system response quantities of interest (unless of course lower-order numerical approximations are used in their evaluation). One should keep in mind that this asymptotic requirement applies not just to the fine mesh solution but to the coarse mesh solution as well. Procedures for confirming that the asymptotic range has been reached will be given in Section 8.4.2.

#### 8.3.3.2 Uniform mesh spacing

The discretization error expansion is in terms of a single mesh spacing parameter $h$. This parameter is a measure of the size of the discretization, and thus has units of length for spatial discretizations and time for temporal discretizations. This could be strictly interpreted as allowing only Cartesian meshes with spacing $h$ in each of the spatial coordinate directions. While this restriction seemingly prohibits the use of Richardson extrapolation for practical scientific computing applications, this is in fact not the case. Recall our previous discussions on the use of local or global mesh transformations and their impact on the order of accuracy in Section 5.4. Those findings also apply to the use of Richardson extrapolation. The spatial mesh quality can affect the formal order of accuracy of the method through interactions with the mesh resolution and the local solution behavior; this relationship is explored in detail in Chapter 9. Discrete transformations will not adversely impact the Richardson extrapolation procedure as long as they are of the same order of accuracy as the discretization scheme or higher. Thompson *et al.* (1985) note that there may be accuracy advantages to evaluating the discrete transformation metrics with the same underlying discretization used for the solution derivatives.

#### *8.3.3.3 Systematic mesh refinement*

An often overlooked requirement for the use of Richardson extrapolation is that the two mesh levels be systematically refined. Recall the definition of systematic mesh refinement given in Section 5.4, which requires that the mesh refinement be both uniform and consistent. Uniform refinement means that the mesh is refined by the same factor over the entire domain, which precludes the use of local refinement or adaptation during the Richardson extrapolation procedure. Consistent refinement requires that the mesh quality must either remain constant or improve with mesh refinement. Examples of mesh quality metrics include cell aspect ratio, cell skewness, and cell-to-cell stretching factor. Techniques for evaluating the uniformity and consistency of the mesh refinement are discussed in Section 8.7. For an example of how Richardson extrapolation can fail in the presence of nonuniform mesh refinement even in the asymptotic range, see Eca and Hoekstra (2009a).

#### *8.3.3.4 Smooth solutions*

As discussed earlier, the coefficients $g$ in the discretization expansion given by Eq. (8.57) are generally functions of the solution derivatives. As such, the Richardson extrapolation procedure will break down in the presence of discontinuities in any of the dependent variables or their derivatives. This is further complicated by the fact that the observed order of accuracy often reduces to first order or lower in the presence of certain discontinuities and singularities, regardless of the formal order of accuracy of the method for smooth problems (see Section 8.8.1).

#### *8.3.3.5 Other numerical errors sources*

Recall that the discretization error is defined as the difference between the exact solution to the discrete equations and the exact solution to the mathematical model. The exact solution to the discrete equations is never known due to round-off error, iterative error, and statistical sampling errors (when present). In practice, the available numerical solutions are used as surrogates for the exact solution to the discretized equations. If the other numerical error sources are too large, then they will adversely impact the Richardson extrapolation procedure since any extrapolation procedure will tend to amplify "noise" (Roache, 1998). A good rule of thumb is to ensure that all other sources of numerical error are at least two orders of magnitude smaller than the discretization error in the fine grid numerical solution (Roy, 2005; Eca and Hoekstra, 2009b).

### *8.3.4 Extensions*

This section describes three extensions of Richardson extrapolation when it is applied locally throughout the domain. The first addresses a method for obtaining the Richardson extrapolation estimate on all fine grid spatial points, while the second extends it to all fine mesh points in space and time. The third extension combines an extrapolation procedure with a discrete residual minimization and could be considered a hybrid extrapolation/residual method.

#### *8.3.4.1 Completed Richardson extrapolation in space*

If the Richardson extrapolation procedure is to be applied to the solution point-wise in the domain, then it requires that one obtain fine mesh solution values at the coarse grid points. For systematic mesh refinement with integer values for the refinement factor on structured grids, this is automatically the case. However, applying Richardson extrapolation to cases with integer refinement will result in estimates of the exact solution only at the coarse grid points. In order to obtain exact solution estimates at the fine grid points, Roache and Knupp (1993) developed the *completed Richardson extrapolation* procedure. Their approach requires interpolation of the fine grid correction (rather than the Richardson extrapolation estimate or the coarse grid solution) from the coarse grid to the fine grid. This interpolation should be performed with an order of accuracy at least as high as the underlying discretization scheme. When this fine grid correction is combined with the discrete solution on the fine grid, an estimate of the exact solution to the mathematical model is obtained that has the same order of accuracy as the Richardson extrapolation estimates on the coarse grid points.

#### *8.3.4.2 Completed Richardson extrapolation in space and time*

Richards (1997) further extended the completed Richardson extrapolation procedure of Roache and Knupp (1993). The first modification provides the higher-order estimate of the exact solution to be obtained on all the fine grid spatial points for integer refinement factors other than two. The second, more significant modification provides higher-order accurate estimates of the exact solution after a chosen number of coarse grid time steps. The approach allows for different formal orders of accuracy in space and time by choosing the temporal refinement factor in such a way as to obtain the same order of error reduction found in the spatial discretization. For a discretization that is formally $p$th-order accurate in space and $q$th-order accurate in time, the temporal refinement factor $r_{\mathrm{t}}$ should be chosen such that

$$r_{\mathrm{t}} = (r_{\mathrm{x}})^{p/q}\,, \tag{8.63}$$

where $r_{\mathrm{x}}$ is the refinement factor in space. This procedure is closely related to the combined order verification procedure discussed in Section 5.5.

#### *8.3.4.3 Least squares extrapolation*

Garbey and Shyy (2003) have developed a hybrid extrapolation/residual method for estimating the exact solution to the mathematical model. Their approach involves forming a more accurate solution by taking linear combinations of discrete solutions on multiple mesh levels using a set of spatially-varying coefficients. Spline interpolation is employed to obtain a smooth representation of this solution on a yet finer grid. The coefficients are then determined by a least squares minimization of the discrete residual formulated on this finer mesh. Their approach thus requires only residual evaluations on this finer mesh, which are significantly less expensive than computing a discrete solution on this mesh. This least squares extrapolated solution is demonstrated to be order $(p + 1)$ accurate, where $p$ is the formal order of accuracy of the method. The higher-order estimate of the exact solution to the mathematical model can be used as a local error estimator or to provide solution initialization within a multigrid-type iterative procedure.

### *8.3.5 Discretization error estimation*

While it may be tempting to use the Richardson extrapolated value as a more accurate solution than the fine grid numerical solution, this should only be done when there is a high degree of confidence that the five assumptions underlying Richardson extrapolation have indeed been met. In particular, the observed order of accuracy (which requires discrete solutions on three systematically-refined meshes, as discussed in Section 8.4.2) must first be shown to match the formal order of accuracy of the discretization scheme. In any case, when only two mesh levels are available, the asymptotic nature of the solutions cannot be confirmed, thus one is limited to simply using the Richardson extrapolated value to estimate the discretization error in the fine grid numerical solution.

Substituting the estimated exact solution from the generalized Richardson extrapolation expression (Eq. (8.62)) into the definition of the discretization error on the fine mesh (Eq. (8.46)) gives the estimated discretization error for the fine mesh (with spacing $h$),

$$\bar{\varepsilon}_h = f_h - \bar{f} = f_h - \left( f_h + \frac{f_h - f_{rh}}{r^p - 1} \right)$$

or simply

$$\bar{\varepsilon}_h = -\frac{f_h - f_{rh}}{r^p - 1}. \tag{8.64}$$

While this does provide for a consistent discretization error estimate as the mesh is systematically refined, there is no guarantee that the estimate will be reliable for any given fine mesh ($h$) and coarse mesh ($rh$) discrete solutions. Therefore, if only two discrete solutions are available then this numerical error estimate should be converted into a numerical uncertainty, as discussed in Section 8.5.

#### *8.3.5.1 Example: Richardson extrapolation-based error estimation*

An example of using the Richardson extrapolation procedure as an error estimator was presented by Roy and Blottner (2003). They examined the hypersonic, transitional flow over a sharp cone. The system response quantity was the heat flux distribution along the surface. The surface heat flux is shown versus the axial coordinate in Figure 8.8a for three

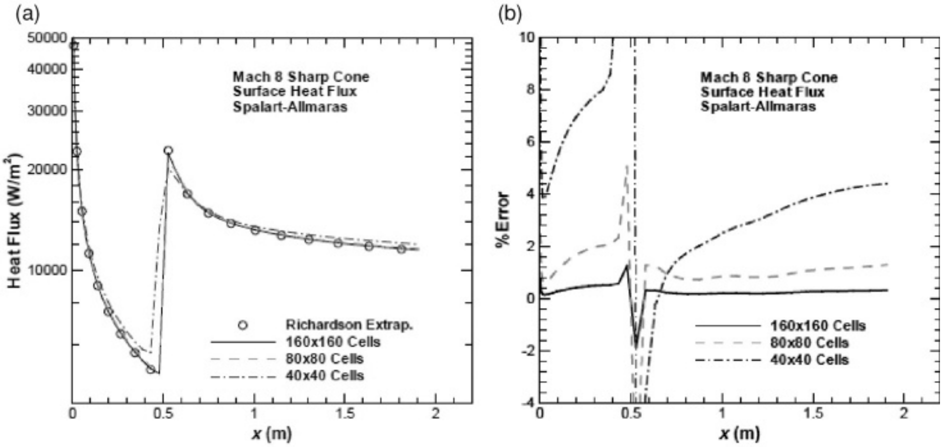

Figure 8.8 (a) Surface heat flux and (b) relative discretization error for the transitional flow over a sharp cone (Roy and Blottner, 2003).

systematically-refined mesh levels: fine (160×160 cells), medium (80×80 cells), and coarse (40×40 cells). Also shown are Richardson extrapolation results found from the fine and medium mesh solutions. The sharp rise in heat flux at $x = 0.5$ m is due to the specification of the location for transition from laminar to turbulent flow.

In Figure 8.8b, the Richardson extrapolation results are used to estimate the discretization error in each of the numerical solutions. Neglecting the immediate vicinity of the transition location, the maximum estimated discretization errors are approximately 8%, 2%, and 0.5% for the coarse, medium, and fine meshes, respectively. The solutions thus appear to be converging as $h \rightarrow 0$. Furthermore, these estimated errors display the expected $h^p$ reduction for these formally second-order accurate computations. In the turbulent region, the maximum errors are also converging at the expected rate giving error estimates of approximately 4%, 1% and 0.25%. More rigorous methods for assessing the reliability of discretization error estimates are addressed in Section 8.4.

### *8.3.6 Advantages and disadvantages*

The primary advantage that Richardson extrapolation holds over other discretization error estimation methods is that it can be used as a post-processing technique applied to any discretization scheme (finite difference, finite volume, finite element, etc.). In addition, it gives estimates in the total error, which includes both locally generated errors and those transported from other regions of the domain. Finally, it can be used for any quantity of interest including both local solution quantities and derived system response quantities (assuming that any numerical approximations have been made with sufficient accuracy).

There are, however, some disadvantages to using discretization error estimators based on Richardson extrapolation. First and foremost, they rely on having multiple numerical solutions in the asymptotic grid convergence range. This can place significant burdens on the grid generation process, which is already a bottleneck in many scientific computing applications. Furthermore, these additional solutions can be extremely expensive to compute. Consider the case where one starts with a 3-D mesh consisting of 1 million elements. Performing a mesh refinement with a refinement factor of two thus requires a solution on a mesh with 8 million elements. When one also accounts for the additional time steps or iterations required for this finer mesh, the solution cost easily increases by an order of magnitude with each refinement (note that integer refinement is not necessarily required, as discussed in Section 8.7.3).

The underlying theory of Richardson extrapolation requires smooth solutions, thus reducing the reliability of these error estimators for problems with discontinuities or singularities. In addition, the extrapolation procedure tends to amplify other sources of error such as round-off and iterative convergence error (Roache, 1998). Finally, the extrapolated quantities will not satisfy the same governing and auxiliary equations as either the numerical solutions or the exact solution. For example, if an equation of state is used to relate the density, pressure, and temperature in a gas, there is no guarantee that extrapolated values for density, pressure, and temperature will also satisfy this equation.

## 8.4 Reliability of discretization error estimators

One of the key requirements for reliability (i.e., accuracy) of any of the discretization error estimators discussed in this chapter is that the solution (or solutions) must be in the asymptotic range. This section provides a discussion of just what this asymptotic range means for discretization approaches involving both mesh ($h$) refinement and order ($p$) refinement. Regardless of whether $h$- or $p$-refinement is used, the *demonstration* that the asymptotic range has been achieved generally requires that at least three discrete solutions be computed. Demonstrating that the asymptotic range has been reached can be surprisingly difficult for complex scientific computing applications involving nonlinear, hyperbolic, coupled systems of equations. It is unlikely that the asymptotic range will be reached in such cases without the use of solution adaptation (see Chapter 9).

### *8.4.1 Asymptotic range*

The asymptotic range is defined differently depending on whether one is varying the mesh resolution or the formal order of accuracy of the discretization scheme. When mesh refinement is employed, then the asymptotic range is defined as the sequence of systematically-refined meshes over which the discretization error reduces at the formal order of accuracy of the discretization scheme. Examining the discretization error expansion for a $p$th-order accurate scheme given by Eq. (8.57), the asymptotic range is achieved when $h$ is sufficiently small that the $h^p$ term is much larger than any of the higher-order terms. Due to possible differences in the signs for the higher-order terms, the behavior of the discretization error outside of the asymptotic range is unpredictable. Demonstrating that the asymptotic range has been reached using systematic mesh refinement is achieved by evaluating the observed order of accuracy. The observed order assesses the behavior of the discrete solutions over a range of meshes and its evaluation is discussed in the next section.

For discretization methods involving order refinement, the asymptotic range is determined by examining the behavior of the numerical solutions with successively refined basis functions, all on the same mesh. As the basis functions are refined and the physical phenomena in the problem are resolved, the discrete solutions will eventually become better approximations of the exact solution to the mathematical model. Convergence is best monitored with error norms since convergence can be oscillatory with increased basis order. An example of hierarchical basis functions used within the finite element method is given by Bank (1996).

### *8.4.2 Observed order of accuracy*

The observed order of accuracy is the measure that is used to assess the confidence in a discretization error estimate. When the observed order of accuracy is shown to match the formal order, then one can have a high degree of confidence that the error estimate is reliable. In the discussion of the observed order of accuracy from Chapter 5, only two numerical solutions were required for its calculation because the exact solution was known. When the exact solution is not known, which is the case for solution verification, three numerical solutions on systematically-refined meshes are required to calculate the observed order of accuracy. For this observed order of accuracy to match the formal order of the discretization scheme, the requirements are the same as those given in Section 8.3.3 for Richardson extrapolation. When any of these requirements fail to be met, inaccurate values for the observed order of accuracy can be obtained (e.g., see Roy, 2003; Salas, 2006; and Eca and Hoekstra, 2009a).

#### *8.4.2.1 Constant grid refinement factor*

Consider a $p$th-order accurate scheme with numerical solutions on a fine mesh ($h_1$), a medium mesh ($h_2$), and a coarse mesh ($h_3$). For the case of a constant grid refinement factor, i.e.,

$$r = \frac{h_2}{h_1} = \frac{h_3}{h_2} > 1,$$

we can thus write

$$h_1 = h, \quad h_2 = rh, \quad h_3 = r^2 h.$$

Using the discretization error expansion from Eq. (8.57), we can now write for the three discrete solutions:

$$\begin{aligned} f_1 &= \tilde{f} + g_p h^p + g_{p+1} h^{p+1} + O(h^{p+2}), \\ f_2 &= \tilde{f} + g_p (rh)^p + g_{p+1} (rh)^{p+1} + O(h^{p+2}), \\ f_3 &= \tilde{f} + g_p (r^2 h)^p + g_{p+1} (r^2 h)^{p+1} + O(h^{p+2}). \end{aligned} \tag{8.65}$$

Neglecting terms of order $h^{p+1}$ and higher allows us to recast these three equations in terms of a locally-observed order of accuracy $\hat{p}$:

$$\begin{aligned} f_1 &= \bar{f} + g_p h^{\hat{p}}, \\ f_2 &= \bar{f} + g_p (rh)^{\hat{p}}, \\ f_3 &= \bar{f} + g_p (r^2 h)^{\hat{p}}, \end{aligned} \tag{8.66}$$

which will only match the formal order of accuracy if the higher order terms are indeed small. Subtracting $f_2$ from $f_3$ and $f_1$ from $f_2$ yields

$$f_3 - f_2 = g_p \left(r^2 h\right)^{\hat{p}} - g_p (rh)^{\hat{p}} = g_p r^{\hat{p}} h^{\hat{p}} \left(r^{\hat{p}} - 1\right) \tag{8.67}$$

and

$$f_2 - f_1 = g_p (rh)^{\hat{p}} - g_p h^{\hat{p}} = g_p h^{\hat{p}} \left(r^{\hat{p}} - 1\right). \tag{8.68}$$

Dividing Eq. (8.67) by (8.68) gives

$$\frac{f_3 - f_2}{f_2 - f_1} = r^{\hat{p}}. \tag{8.69}$$

Taking the natural log of both sides and solving for the observed order of accuracy $\hat{p}$ gives

$$\hat{p} = \frac{\ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right)}{\ln(r)}. \tag{8.70}$$

Consistent with the development of generalized Richardson extrapolation in Section 8.3.2, the Richardson extrapolated estimate of the exact solution $\bar{f}$ and the leading error term coefficient $g_p$ are now given in terms of the observed order of accuracy $\hat{p}$ by

$$\bar{f} = f_1 + \frac{f_1 - f_2}{r^{\hat{p}} - 1} \tag{8.71}$$

and

$$g_p = \frac{f_1 - \bar{f}}{h^{\hat{p}}}. \tag{8.72}$$

Note that it is only when the observed order of accuracy matches the formal order of the numerical scheme that we can expect the discretization error estimate given by Eq. (8.71) to be accurate. This is equivalent to saying that the solutions on all three meshes are in the asymptotic range and the higher-order terms in Eqs. (8.65) are small for all three meshes. In practice, when this locally observed order of accuracy is used for the extrapolation estimate, it is often limited to be in the range

$$0.5 \le \hat{p} \le p_{\mathrm{f}},$$

where $p_{\mathrm{f}}$ is the formal order of accuracy of the discretization scheme. Allowing the observed order of accuracy to increase above the formal order can result in discretization error estimates that are not conservative (i.e., they underestimate the error). Furthermore, as $\hat{p}$ approaches zero, the magnitude of the extrapolated estimate grows without bound.

#### *8.4.2.2 Non-constant grid refinement factor*

For the case of non-constant grid refinement factors

$$r_{12} = \frac{h_2}{h_1} > 1, \quad r_{23} = \frac{h_3}{h_2} > 1,$$

where $r_{12} \neq r_{23}$, the determination of the observed order of accuracy $\hat{p}$ is more complicated. For this case, the following transcendental equation (Roache, 1998) must be solved for $\hat{p}$:

$$\frac{f_3 - f_2}{r_{23}^{\hat{p}} - 1} = r_{12}^{\hat{p}} \left( \frac{f_2 - f_1}{r_{12}^{\hat{p}} - 1} \right), \tag{8.73}$$

This equation can usually be solved with a simple direct substitution iterative procedure as discussed in Chapter 7 to give

$$\hat{p}^{k+1} = \frac{\ln\left[ \left( r_{12}^{\hat{p}^k} - 1 \right) \left( \frac{f_3 - f_2}{f_2 - f_1} \right) + r_{12}^{\hat{p}^k} \right]}{\ln \left( r_{12} r_{23} \right)}, \tag{8.74}$$

where an initial guess of $\hat{p}^k = p_{\mathrm{f}}$ (the formal order of the scheme) can be used. Once the observed order of accuracy is found, the estimate of the exact solution $\bar{f}$ and the leading error term coefficient $g_p$ are given by Eqs. (8.71) and (8.72) by replacing the constant grid refinement factor with $r = r_{12}$.

#### *8.4.2.3 Application to system response quantities*

Recall that system response quantities are defined as any solution property derived from the solution to the mathematical model or its discrete approximation. Examples of common system response quantities in scientific computing are lift and drag in aerodynamics, heat flux through a surface in heat transfer analysis, and maximum stress in a structural mechanics problem. The observed order of accuracy for system response quantities can be evaluated by the approaches described earlier in this section. For this observed order of accuracy to match the formal order of the discretization scheme, there is an additional requirement beyond those described in Section 8.3.3 for Richardson extrapolation. This additional requirement pertains to the order of accuracy of any numerical approximations used to compute the system response quantity. When the system response quantity is an integral, a derivative, or an average, then the numerical approximations used in its evaluation must be of at least the same order of accuracy as the underlying discretization scheme. In most cases, integrated quantities and averages are better behaved and converge more rapidly with mesh refinement than local quantities. However, in some cases the errors due to the numerical quadrature can interact with the numerical errors in the discrete solution and adversely impact the observed order of accuracy computation. An example of

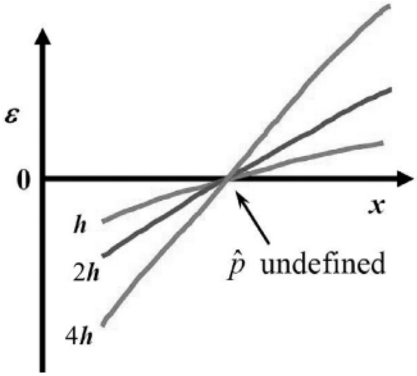

Figure 8.9 Simple example of how the observed order of accuracy computation will fail when applied locally over the domain: the observed order will be undefined at the crossover point.

these interactions for the computation of drag on an inviscid supersonic blunt body (mixed elliptic/hyperbolic) problem is given by Salas and Atkins (2009).

#### *8.4.2.4 Application to local quantities*

Problems often arise when the observed order of accuracy is evaluated on a point-by-point basis in the domain. A simple example of a case where this local evaluation of the order of accuracy will fail is given in Figure 8.9 which shows the discretization error on three different 1-D meshes. If the meshes are refined by a factor of two and the formal order of the scheme is first-order accurate, then we expect the discretization error to drop by a factor of two for each refinement. However, as can commonly occur in practical applications, the numerical solutions in one part of the domain can approach the exact solution from above while the numerical solutions in another part can approach it from below (Potter *et al.*, 2005). Even if we neglect any other sources of numerical error (such as round-off error), the observed order of accuracy at this crossover point will be undefined, even though the discretization error on all three meshes is exactly zero. When the observed order of accuracy is computed *near* this crossover point, the effects of other numerical error sources (e.g., round-off and iterative error) can become important. Such problems can be addressed by employing a globally evaluated observed order of accuracy (e.g., see Section 8.6.3.2).

Another example of the problems that can occur when examining the observed order of accuracy on a point-by-point basis through the domain was given by Roy (2003). The problem of interest is inviscid, hypersonic flow over a sphere-cone geometry. The mathematical character of this problem is elliptic immediately behind the normal shock wave that forms upstream of the sphere, but hyperbolic over the rest of the solution domain. The observed order of accuracy for the surface pressure is plotted versus the normalized axial distance based on three uniformly refined meshes in Figure 8.10. The finest mesh was 1024×512 cells and a refinement factor of two was used to create the coarse meshes. While a formally second-order accurate finite volume discretization was used, flux limiters were

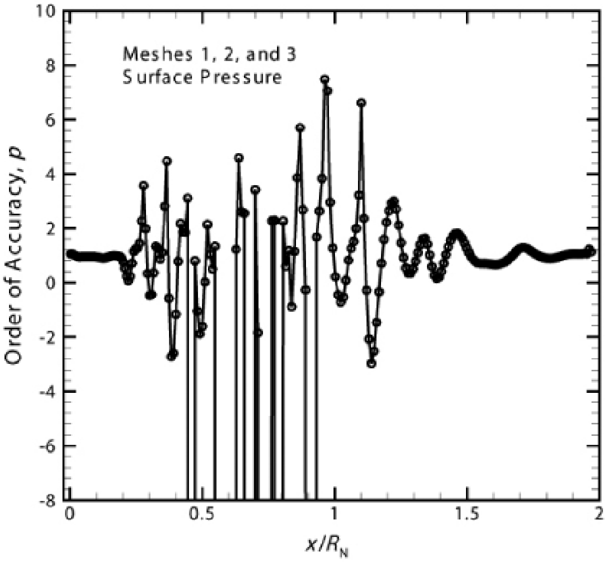

Figure 8.10 Observed order of accuracy in surface pressure for inviscid hypersonic flow over a sphere-cone geometry (from Roy, 2003).

employed in the region of the shock wave discontinuity to capture the shock in a monotone fashion by locally reducing the formal order of accuracy to one. The observed order of accuracy is indeed first order and well-behaved in the elliptic region (up to $x/R_N$ of 0.2). However, in the hyperbolic region, the observed order of accuracy is found to undergo large oscillations between $-4$ and $+8$, with a few locations being undefined. Farther downstream, the observed order again becomes well-behaved with values near unity. The source of these oscillations is likely the local characteristic waves generated when the shock moves from one grid line to another. (This is the same example that is given in Figure 8.1 showing the local and transported components of the discretization error.) Clearly extrapolation-based error estimates using the local order of accuracy would not be reliable in this region. As mentioned above, a globally-evaluated observed order of accuracy could be used in this case. Alternatively, the formal order of accuracy (here $p = 1$) could be used.

## 8.5 Discretization error and uncertainty

As discussed previously, when it has been demonstrated that the solutions are in the asymptotic range, then one can have confidence that the error estimate is reliable and therefore use the error estimate to correct the solution. While the calculation of the observed order of accuracy requires three systematically refined meshes, a fourth mesh level is recommended to confirm that the asymptotic range has indeed been reached. However, the much more common case is when the formal order does not match the observed order. In this case, the error estimate is much less reliable and should generally be converted into a numerical uncertainty. While the difference between the discrete solution and the (unknown) exact solution to the mathematical model is still truly a numerical error, because we do not know the true value of this error it should be represented as an uncertainty. Uncertainties due to such a lack of knowledge are called epistemic uncertainties and are distinct from aleatory (or random) uncertainties. They can be reduced by providing more information, in this case, additional computations on more refined meshes. The treatment of these numerical uncertainties is discussed in Chapter 13. In the next section we address some approaches for converting Richardson extrapolation estimates of the discretization error into epistemic uncertainties.

## 8.6 Roache's grid convergence index (GCI)

Before proceeding with a discussion of the grid convergence index (Roache, 1994), we first provide some motivation by considering the state of affairs in the early 1990s regarding the reporting of mesh refinement studies. This time period is before the appearance of numerical accuracy policies for journals dealing with scientific computing simulations. In some cases, authors mistakenly reported discretization error estimates by giving the relative difference between two discrete solutions computed on different meshes, i.e.,

$$\frac{f_2 - f_1}{f_1},$$

where $f_1$ is a fine mesh solution and $f_2$ a coarse mesh solution. This relative difference can be extremely misleading when used as an error estimate. To see why, consider the estimate of the relative discretization error (RDE) found from generalized Richardson extrapolation, which for the fine grid can be written as

$$\mathrm{RDE}_1 = \frac{f_1 - \bar{f}}{\bar{f}}. \tag{8.75}$$

Substituting the generalized Richardson extrapolation result from Eq. (8.62) into the above equation gives

$$\mathrm{RDE}_1 = \frac{f_1 - \left(f_1 + \frac{f_1 - f_2}{r^p - 1}\right)}{f_1 + \frac{f_1 - f_2}{r^p - 1}} = \frac{f_2 - f_1}{f_1 r^p - f_2}. \tag{8.76}$$

As an example, consider two numerical solutions where some quantity of interest $f$ has fine and coarse grid values of 20 and 21, respectively, for a relative difference between solutions of 5%. For a third-order accurate scheme with refinement factor $r = 2$, the error estimate based on Richardson extrapolation from Eq. (8.76) is 0.71%. However, for a first-order accurate numerical scheme with a grid refinement factor of 1.5, the error estimate based on Richardson extrapolation is 9.1%. Thus, a 5% relative difference in the two solutions can mean very different values for the relative discretization error, depending on the order of accuracy of the scheme and the grid refinement factor. This example illustrates the importance of accounting for the grid refinement factor $r$ and the order of accuracy $p$ when using extrapolation to estimate the discretization error. The desire to prevent misuse of this relative difference between discrete solutions as an error estimator led to the development of Roache's grid convergence index.

### *8.6.1 Definition*

Roache (1994) proposed the *grid convergence index*, or GCI, as a method for uniform reporting of grid refinement studies. Roache's stated goal in formulating the GCI is to achieve a 95% certainty (i.e., that it provide conservative uncertainty estimates in 19 out of 20 cases) over a wide range of applications. The original GCI is based on the often reported relative difference between two discrete solutions, but also properly accounts for the amount of grid refinement and the order of accuracy. The GCI takes the further step of converting the discretization error estimate into an uncertainty estimate using absolute values. The GCI for the fine grid numerical solution is defined as

$$\mathrm{GCI} = \frac{F_\mathrm{s}}{r^p - 1}\left|\frac{f_2 - f_1}{f_1}\right|, \tag{8.77}$$

where $F_\mathrm{s}$ is a factor of safety. When only two discrete solutions are available, then the formal order of accuracy of the discretization scheme is used in the definition of the GCI along with a factor of safety of $F_\mathrm{s} = 3$. However, when three discrete solutions are available then the observed order of accuracy can be computed. When the observed order matches the formal order of accuracy, then a less conservative factor of safety of $F_\mathrm{s} = 1.25$ can be used (Roache, 1998). If the GCI is re-derived in terms of the Richardson extrapolation estimate (as in Eq. (8.76)), then $f_1$ in the denominator will be replaced with the Richardson extrapolation estimate $\bar{f}$; however, this modification will only significantly affect the uncertainty estimate when the uncertainties themselves are large (Roy, 2001). When the implementation of Eq. (8.77) is used, the GCI returns a fractional estimate of the relative uncertainty in the fine grid solution. For example, a GCI value of 0.15 indicates an uncertainty due to the discretization process of 15% in the fine grid solution. As with all extrapolation-based approaches, systematic mesh refinement must be used.

It is important to include the factor of safety in the GCI. The GCI is based on the Richardson extrapolated value, which is itself an ordered estimate of the exact solution to the mathematical model (recall Eq. (8.61)). Thus we do not know *a priori* whether the estimated exact solution is above or below the true exact solution to the mathematical model. Consider Figure 8.11, which shows two numerical solutions ($f_1$ and $f_2$), the estimated exact solution from Richardson extrapolation $\bar{f}$, and the true exact solution $\tilde{f}$. In general, there is an equal chance that the true exact solution is above or below the estimated value. Thus a factor of safety of $F_\mathrm{s} = 1$ centered about the fine grid numerical solution $f_1$ will only provide 50% chance that the true error $\tilde{f}$ is within the uncertainty band. Increasing the factor of safety should increase the possibility that the true error will fall within the band. This argument is made simply to argue for the use of a factor of safety in converting the discretization error estimate into an epistemic uncertainty. It in no way implies that the

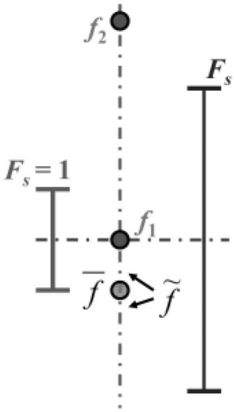

Figure 8.11 Factor of safety for extrapolation-based error bands (from Roy, 2005).

resulting uncertainty is a randomly distributed variable (i.e., an aleatory uncertainty). To reiterate the main point made in Section 8.4, the reliability of any discretization error or uncertainty estimate can only be determined by assessing the asymptotic nature of the discrete solutions. When the solutions are found to be far outside the asymptotic range, the reliability of the error/uncertainty estimate will likely be poor and its behavior erratic. In this case, no value for the factor of safety is guaranteed to be conservative.

### 8.6.2 Implementation

In order to avoid problems when the solution values are near zero, most recent implementations of the GCI omit the normalization by the fine grid solution in the denominator, i.e., the GCI is redefined as

$$\mathrm{GCI} = \frac{F_\mathrm{s}}{r^p - 1} \left| f_2 - f_1 \right|. \tag{8.78}$$

This implementation thus provides an uncertainty estimate for the fine grid solution in the same units as the solution itself. Roache (1998) provides clear guidelines for choosing the factor of safety when:

1 solutions on only two grids are available ($F_\mathrm{s} = 3$), or
2 solutions on three grids are available and the observed order of accuracy is calculated and shown to match the formal order of the scheme ($F_\mathrm{s} = 1.25$).

In the first case, when only two grid levels are available, GCI estimates should be used with caution since one has no information on whether the solutions are in (or even near) the asymptotic range. Far from the asymptotic range, all approaches for estimating discretization error or uncertainty will be unreliable (see Section 8.4). For the second case, when three solutions are available, if the observed order of accuracy matches the formal order, then $F_\mathrm{s} = 1.25$ is used and either the formal or observed order of accuracy can be employed

Table 8.1 *Proposed implementation of the GCI for solutions on three or more systematically refined grids using Eq. (8.78).*

| $\left\lvert \frac{\hat{p}-p_{\mathrm{f}}}{p_{\mathrm{f}}} \right\rvert$ | $F_{\mathrm{s}}$ | $p$ |
|---|---|---|
| $\leq 0.1$ | 1.25 | $p_{\mathrm{f}}$ |
| $> 0.1$ | 3.0 | $\min(\max(0.5, \hat{p}), p_{\mathrm{f}})$ |

(since they match, this choice will have little effect on the estimate). Difficulties arise when the observed and formal orders of accuracy do not agree. Furthermore, exactly how does one define agreement? Ideally, one would simply continue to systematically refine the mesh until the solutions are demonstrably asymptotic, possibly incorporating local mesh refinement to achieve the asymptotic range faster (see Chapter 9). When no additional solutions are possible due to resource limitations, then Roache (1998) provides anecdotal examples of how to apply the GCI for a wide range of situations.

We propose the following procedure for calculating the GCI when solutions on three or more systematically-refined meshes are available. In all cases, the nonnormalized GCI given by Eq. (8.78) is used. When the observed order of accuracy $\hat{p}$ agrees with the formal order $p_{\mathrm{f}}$ within 10%, then the formal order of accuracy along with a factor of safety of 1.25 is used in the GCI calculation. When the observed order of accuracy does not agree within 10%, then a factor of safety of three ($F_s = 3.0$) is used. Furthermore, the order of accuracy is limited between 0.5 and the formal order. Allowing the order of accuracy to be much larger than the formal order causes the uncertainty estimates to be unreasonably small since the GCI goes to zero as $p \to \infty$, while allowing the order of accuracy to go to zero causes the uncertainty estimate to approach infinity. These recommendations are summarized in Table 8.1. While these recommendations are "reasonable" (Roache, 2009), they require testing to see if they produce the desired 95% uncertainty bands on a wide range of problems.

### 8.6.3 Variants of the GCI

There are a number of different variants of the GCI that have been developed in recent years. These variations address different ways of computing the factor of safety and/or the order of accuracy used in the GCI calculation. In addition, special cases are often addressed depending on the behavior of the observed order of accuracy.

#### 8.6.3.1 Least squares method

The local calculation of the observed order of accuracy often exhibits large departures from the formal order of the discretization scheme. These variations can be due to a number of sources, including the discrete solutions not being asymptotic, errors transported from other regions (especially for hyperbolic problems), iterative error, round-off error, interpolation of solutions onto a common grid, and nonuniformity of the mesh refinement. Eca and Hoekstra (2002) developed a method for filtering out the "noise" from the observed order of accuracy calculation using a least squares fitting approach over four or more mesh levels. Recall the series expansion from generalized Richardson extrapolation from Eq. (8.57), which can be written for a general mesh level $k$ as

$$f_k = \bar{f} + g_p h_k^{\hat{p}}.$$

In their approach, they minimize the function

$$S(\bar{f}, g_p, \hat{p}) = \left\{ \sum\nolimits_{k=1}^{NG} \left[ f_k - \left( \bar{f} + g_p h_k^{\hat{p}} \right) \right]^2 \right\}^{1/2}, \tag{8.79}$$

where $k$ refers to a mesh level and $NG$ is the total number of mesh levels ($NG > 3$). This is accomplished by setting the derivatives of $S$ with respect to $\bar{f}$, $g_p$, and $\hat{p}$ to zero, which results in the following expressions:

$$\bar{f} = \frac{\sum_{k=1}^{NG} f_k - g_p \sum_{k=1}^{NG} h_k^{\hat{p}}}{NG}, \tag{8.80}$$

$$g_p = \frac{NG \sum_{k=1}^{NG} f_k h_k^{\hat{p}} - \left( \sum_{k=1}^{NG} f_k \right) \left( \sum_{k=1}^{NG} h_k^{\hat{p}} \right)}{NG \sum_{k=1}^{NG} h_k^{2\hat{p}} - \left( \sum_{k=1}^{NG} h_k^{\hat{p}} \right) \left( \sum_{k=1}^{NG} h_k^{\hat{p}} \right)}, \tag{8.81}$$

$$\sum\nolimits_{k=1}^{NG} f_k h_k^{\hat{p}} \ln(h_k) - \bar{f} \sum\nolimits_{k=1}^{NG} h_k^{\hat{p}} \ln(h_k) - g_p \sum\nolimits_{k=1}^{NG} h_k^{2\hat{p}} \ln(h_k) = 0. \tag{8.82}$$

Eca and Hoekstra (2002) employed a false position method to iteratively solve Eq. (8.82) for $\hat{p}$. The main drawback to the least squares method is that it requires solutions on four or more systematically refined grids. While their original approach actually applies the uncertainty estimate to the extrapolated value $\bar{f}$ (Eca and Hoekstra, 2002), subsequent studies have all applied the uncertainty estimate to the fine grid solution by using the order of accuracy found by solving Eq. (8.82) directly in the GCI calculation of Eq. (8.78) (e.g., see Eca *et al.*, 2005).

#### 8.6.3.2 *Global averaging method*

Cadafalch *et al.* (2002) employed a globally averaged value for the observed order of accuracy to provide local GCI estimates. Their approach can be summarized in the following five steps.

1 Interpolate the solutions from three systematically-refined meshes onto a common post-processing grid with a higher-order interpolation method.
2 Classify the nodes in the common mesh as either monotone $(f_3 - f_2)(f_2 - f_1) > 0$ or nonmonotone $(f_3 - f_2)(f_2 - f_1) < 0$ (they also treated a third case where the magnitude of this product was less than a tolerance of $10^{-30}$).
3 Compute the local observed order of accuracy for all of the monotone converging nodes.

4 Compute a global observed order of accuracy by simply averaging all of the local observed orders computed in step 3.
5 Compute the local GCI values at the monotone converging nodes using a factor of safety of $F_s = 1.25$ and with the global observed order of accuracy from step 4.

A significant limitation of this approach is that it does not provide uncertainty estimates at any nodes in the common mesh where the solutions converge in a nonmonotone fashion with mesh refinement. This case could be easily treated by simply computing the GCI at these non-monotone nodes using larger factor of safety ($F_s = 3$) along with the global observed order of accuracy.

### *8.6.3.3 Factor of safety method*

One of the drawbacks to the GCI is that the factor of safety only recognizes two cases: when the solutions are asymptotic ($F_s = 1.25$) and when they are not ($F_s = 3$). For solutions that are on the border of the asymptotic range (regardless of how it is defined), the uncertainty estimates will therefore differ by a factor of 2.4. It is natural to desire a smoother variation in the factor of safety, with increasing values as the solutions move further from the asymptotic range. In order to provide for such a smooth transition, Xing and Stern (2009) have developed the factor of safety method. Their approach is designed to address deficiencies (e.g., see Roache, 2003b) in an earlier version of their method (Stern *et al.*, 2001).

In their factor of safety method, Xing and Stern (2009) measure the distance to the asymptotic range in terms of a correction factor defined as

$$\mathrm{CF} = \frac{r^{\hat{p}} - 1}{r^{p_\mathrm{f}} - 1}, \tag{8.83}$$

where $\hat{p}$ is the observed order of accuracy and $p_\mathrm{f}$ the formal order. In the asymptotic range, CF will approach unity. The uncertainty estimate $U$ is then found from

$$U = \begin{cases} [\mathrm{FS}_1 \mathrm{CF} + \mathrm{FS}_0\,(1 - \mathrm{CF})]\,|\delta_\mathrm{RE}|, & 0 < \mathrm{CF} \le 1, \\ \frac{\mathrm{CF}}{2-\mathrm{CF}}\,[\mathrm{FS}_1\,(2 - \mathrm{CF}) + \mathrm{FS}_0\,(\mathrm{CF} - 1)]\,|\delta_\mathrm{RE}|, & 1 < \mathrm{CF} < 2, \end{cases} \tag{8.84}$$

where $\mathrm{FS}_0$ and $\mathrm{FS}_1$ are constants. The values for these constants are $\mathrm{FS}_0 = 2$ and $\mathrm{FS}_1 = 1.25$ and were determined by performing a statistical analysis for a wide range of problems and considering only the cases where $0 < \mathrm{CF} < 2$. The term $\delta_\mathrm{RE}$ is the discretization error estimate found from generalized Richardson extrapolation using the observed order of accuracy, i.e.,

$$\delta_\mathrm{RE} = \frac{f_2 - f_1}{r^{\hat{p}} - 1}. \tag{8.85}$$

The main limitations of this approach are that it only applies to $0 < \mathrm{CF} < 2$ and the uncertainty estimates approach infinity as CF → 2. For a formally second-order accurate method with a grid refinement factor of $r = 2$, the infinite uncertainty estimate occurs when $\hat{p} \approx 2.81$, a situation that can certainly occur in complex scientific computing applications.

### 8.6.4 Reliability of the GCI

Three recent studies have carefully examined the reliability of the GCI. Eca and Hoekstra (2002) examined a wide range of fluid flow problems using a large number of noninteger refined meshes (at least sixteen different grids for each case). They employed a locally evaluated observed order of accuracy and found that the GCI with a factor of safety of $F_s = 1.25$ worked well on the finer meshes. Cadafalch *et al.* (2002) examined five test cases for fluid flow (some with heat transfer) using four to seven meshes for each case with a grid refinement factor of two. In their case, they employed the global averaging method given in Section 8.6.3.2. They also found good results for the GCI with a factor of safety of 1.25 on finer meshes. Finally, Eca *et al.* (2004) looked at potential flow solutions around 2-D airfoils where exact solutions were available. They achieved good uncertainty estimates using the GCI when the locally evaluated observed order of accuracy was not significantly larger than the formal order of the method. See Pelletier and Roache (2006) for additional discussion of the effectiveness of the GCI.

## 8.7 Mesh refinement issues

Second only to the inability to achieve the asymptotic grid convergence range, the failure of extrapolation-based methods for estimating discretization error can often be tied to problems achieving systematic mesh refinement on the meshes in question (e.g., Baker, 2005; Salas, 2006). The use of locally-refined meshes, meshes refined in only one coordinate direction, and refinement where the refinement factor varies locally over the domain are all examples of improper approaches to mesh refinement. When combined with the extrapolation-based methods discussed earlier for error or uncertainty estimation, these approaches are virtually guaranteed to fail.

### 8.7.1 Measuring systematic mesh refinement

Recall the definition of *systematic mesh refinement* given in Section 5.4, which requires that the mesh refinement be both uniform and consistent. Uniform refinement requires that the mesh be refined by the same factor over the entire domain. This does not mean that the mesh itself must be uniform over the domain, only that the ratio of grid refinement from one mesh to another must not vary over the domain. Consistent refinement requires that the meshes should maintain the same grid quality (skewness, aspect ratio, stretching factor, etc.) or possibly provide for improved grid quality with mesh refinement.

A simple technique to ensure that two meshes retain the same volume ratio over the domain was developed by Roy *et al.* (2007) and applied to 3-D unstructured Cartesian meshes used for aerodynamic force and moment computations. The level of uniformity of the mesh refinement was evaluated by comparing cell volume ratios between the two successive mesh levels as follows. First, the local cell volumes were calculated and stored at the nodes. Next, the fine grid volume distribution was interpolated onto the coarse grid

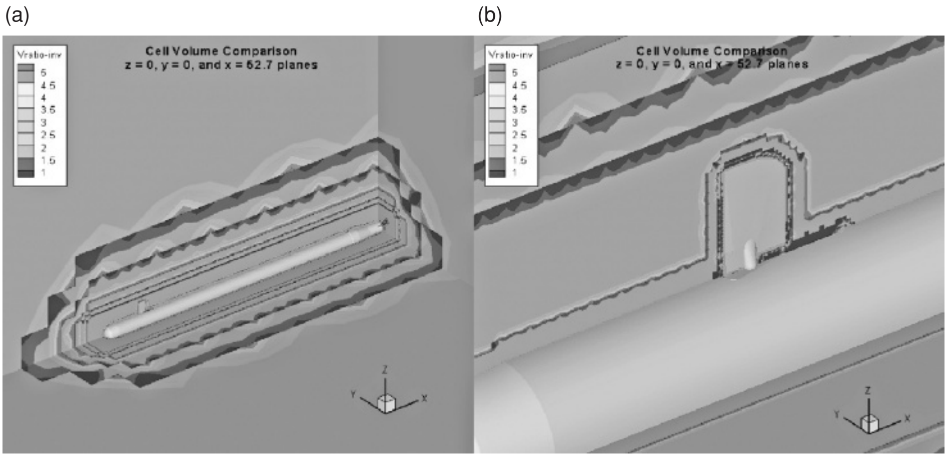

Figure 8.12 Contour plots of the cell volume ratio between the two unstructured Cartesian mesh levels for (a) the entire missile geometry and (b) a close-up of the canard (from Roy *et al.*, 2007). (See color plate section.)

with using an inverse distance function. Now that the coarse grid contains both fine and coarse grid cell volume information, the coarse to fine grid volume ratios can be calculated and examined. An example of the Cartesian mesh used here was given earlier in Figure 8.7b showing different layers of resolution. The volume ratios between two mesh levels are shown in Figure 8.12 and are fairly constant in the domain around a value of 1.6, with the exception of the regions where the Cartesian grid layers transition from one cell size to another. This approach could be easily extended to other mesh quality parameters such as skewness, aspect ratio, stretching factor, etc.

### *8.7.2 Grid refinement factor*

As discussed earlier, the *grid refinement factor* is a measure of the ratio of mesh length or time scales in each of the coordinate directions. The refinement factor is simple to compute for structured grids, while for unstructured grids the refinement factor can be computed as follows. Assuming that the mesh has been refined uniformly (see Section 8.7.1), the refinement factor can be computed as a function of the total number of cells or elements in the mesh by

$$r = \left(\frac{N_1}{N_2}\right)^{1/d}, \tag{8.86}$$

where $N_k$ is the number of cells/elements on mesh $k$ and $d$ is the dimensionality of the problem ($d = 1$ for 1-D, $d = 2$ for 2-D, etc.). Note that in the case where local mesh refinement is applied from one mesh to another, the grid refinement factor will vary over the domain and the numerical solutions should no longer be used in an extrapolation procedure or for the computation of the observed order of accuracy.

### 8.7.3 Fractional uniform refinement

It is not necessary to use grid refinement factors of two, a process referred to as grid doubling or grid halving, depending on whether one starts with the fine mesh or the coarse mesh. For simple mesh topologies (e.g., Cartesian meshes), grid refinement factors as small as $r = 1.1$ can be employed (Roache, 1998). Since the refinement must always be uniform, we will herein refer to this procedure as *fractional uniform refinement*. When refinement factors near one are used, one should take special care to ensure that round-off error does not become large relative to the difference between the two solutions. For example, solving a 2-D problem on two meshes of size $100\times100$ and $101\times101$ elements (i.e, $r = 1.01$) could easily result in solutions that differed by less than 0.01%. Single precision computations generally provide solution errors due to round-off of approximately 0.001%, so the expected round-off errors will be only one order of magnitude smaller than the discretization errors.

Using fractional uniform refinement has the added benefit that it increases the chances of getting discrete solutions on multiple meshes into the asymptotic grid convergence range. For example, consider a 3-D coarse mesh with 1 million elements which is in the asymptotic range. For three levels of mesh refinement with a factor of two, the fine mesh would require 64 million elements. If a refinement factor of 1.26 was used instead, then the fine mesh would have only 4 million cells and would thus be significantly less expensive to compute. However, non-integer grid refinement factors are difficult to apply to complex mesh topologies, especially those involving significant mesh stretching. For simulations using complex, structured meshes, the grid generation can sometimes make up the majority of the overall user's time to perform a given simulation. Thus, relying on the original grid generation procedure for grid refinement can be expensive; furthermore, it is difficult to enforce a constant grid refinement factor over the entire domain. Higher-order interpolation can be used for noninteger grid refinement. Here it is again better to start with the fine mesh and then coarsen (at least for structured meshes); however, this approach may not preserve the underlying surface geometry, especially in regions of high curvature.

When a grid refinement factor of two is employed on structured meshes, there is only significant effort involved in generating the fine mesh; the coarser meshes are found by simply removing every other point. The drawback is not only that the fine mesh may be unnecessarily expensive, but there is also an increased chance that the coarse mesh will be outside the asymptotic grid convergence range. Recall that all three meshes must be in the asymptotic range to demonstrate the reliability of the discretization error and uncertainty estimates.

Ideally, one would like the ability to simply create a grid and then have a grid generation tool generate a family of uniformly-refined meshes based on specified (possibly fractional) refinement factors. Such a capability would greatly improve the reliability of extrapolation-based error and uncertainty estimation procedures. To our knowledge, this automatic fractional uniform refinement capability does not exist in any commercially-available grid generators.

Finally, if the extrapolation (or error/uncertainty) estimation method is to be applied to the solution point-wise through the domain, then it is necessary to get fine grid information onto the coarse grid. This procedure is simple for integer refinement since the coarse grid is simply a subset of the fine grid. However, for fractional uniform refinement, fine mesh information must be interpolated to the coarse mesh points, possibly with an interpolation scheme that has a higher order of accuracy than the underlying numerical solutions (Roache and Knupp, 1993).

### *8.7.4 Refinement vs. coarsening*

In theory, it should not make a difference whether we start with the coarse mesh or the fine mesh. However, in practice, grid coarsening on structured meshes is often easier than grid refinement, especially for complex meshes. Here complex meshes are defined as those with complex geometries and/or significant grid clustering. For uniform meshes, refinement can be performed by simply averaging neighboring spatial location. For stretched meshes, this type of refinement will lead to discontinuities in the ratio of neighboring element sizes near the original coarse grid nodes. (Although, as Ferziger and Peric (1996) point out, the fraction of locations with discontinuous stretching factors will decrease with increasing mesh refinement.) A better strategy for stretched meshes is to use higher-order interpolation to obtain smooth stretching distributions; however, this process can be challenging on highly complex grids. The primary problems that arise during mesh refinement are due to a loss of geometric definition at object surfaces, especially at sharp corners. Furthermore, for structured grid approaches requiring point-to-point match-up at subdomain boundaries, the refinement strategy must ensure that these points are co-located. Thus for complex, structured meshes, it is often easier to simply start with the fine mesh and successively remove every other point in each of the coordinate directions.

For unstructured meshes, it is generally easier to start with the coarse mesh, then refine by subdividing the elements. This is due to the difficulties of merging elements in a manner that preserves the element type while enforcing the requirement of a constant grid refinement factor over the entire domain. While refinement on unstructured grids inherits all of the drawbacks of refinement for structured grids discussed above, there are currently efforts underway to make surface geometry information directly available to mesh refinement routines (e.g., see King *et al.*, 2006).

The choice of methods for refining the elements will determine the effective grid refinement factor. In two dimensions, triangular elements can easily be refined by connecting the midpoints of the edges as shown in Figure 8.13, thereby creating four new triangular elements of similar shape. This figure also shows the corresponding refinement strategy for three-dimensional tetrahedra, where the midpoints of each edge are connected, eventually resulting in eight smaller tetrahedra. In this case, the four new outer tetrahedra will be geometrically similar to the original tetrahedron, but the four interior tetrahedral will not. In both cases shown in Figure 8.13, the grid refinement factor is two. Refinement of quadrilateral and hexahedral cells is straightforward to perform by simple edge bisection.

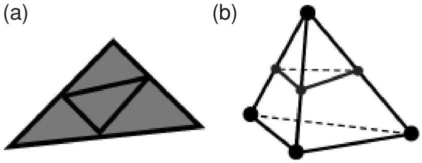

Figure 8.13 Refinement strategy for unstructured meshes: (a) 2-D triangles and (b) 3-D tetrahedra (from Roy, 2005).

### 8.7.5 Unidirectional refinement

It is sometimes the case that the discretization errors come primarily from just one of the coordinate directions. In such cases, it can be helpful to perform independent refinement in the coordinate directions to determine which one is the primary contributor to the overall discretization error. The approach discussed here is similar to the combined space-time method for code order of accuracy verification discussed in Section 5.5.3.2. For independent refinement in $x$ and $y$, we can write the expansion of the numerical solution on mesh $k$ about the exact solution to the original partial differential equation as

$$f_k = \tilde{f} + g_x(\Delta x_k)^p + g_y(\Delta y_k)^q + \cdots, \tag{8.87}$$

where the error terms for each coordinate direction are included. In order to keep the analysis general, the formal order of accuracy in the $x$ direction is assumed to be $p$ and the order of accuracy in the $y$ direction to be $q$, where the two may or may not be equal. Note that for some numerical schemes (e.g., the Lax–Wendroff scheme), a cross term $g_{xy}(\Delta x)^s(\Delta \mathrm{y})^{\mathrm{t}}$ may also be present.

Consider the case of solutions on two meshes levels (fine: $k = 1$ and coarse: $k = 2$) with refinement only in the $x$ direction by a factor of $r_x$. As the $\Delta x$ element size is refined, the term $g_y(\Delta y_k)^q$ will be constant. We are now unable to solve for an estimate of the exact solution $\tilde{f}$, but instead must solve for the quantity

$$\tilde{f}_x = \tilde{f} + g_y(\Delta y_k)^q, \tag{8.88}$$

which includes the error term due to the $\Delta y$ discretization. This term will simply be constant on the two meshes since the $\Delta y$ spacing does not change. Neglecting the higher-order terms into the estimated solution $\bar{f}$ results in the following two equations:

$$\begin{aligned} f_1 &= \bar{f}_x + g_x(\Delta x)^p, \\ f_2 &= \bar{f}_x + g_x(r_x \Delta x)^p, \end{aligned} \tag{8.89}$$

which can be solved for $\bar{f}_x$,

$$\bar{f}_x = f_1 + \frac{f_1 - f_2}{r_x^p - 1}, \tag{8.90}$$

and the leading $x$-direction error term,

$$g_x (\Delta x)^p = \frac{f_2 - f_1}{r_x^p - 1}. \tag{8.91}$$

Similarly, introducing a third solution ($k = 3$) with coarsening only in the $y$ direction allows us to solve for the $y$-direction error term,

$$g_y (\Delta y)^q = \frac{f_3 - f_1}{r_y^q - 1}. \tag{8.92}$$

The size of the two error terms from Eqs. (8.91) and (8.92) can then be compared to determine the appropriate direction for further mesh refinement. In addition, since $g_x$ and $g_y$ have been estimated, Eq. (8.87) can be used to obtain an estimate of $\tilde{f}$.

## 8.8 Open research issues

There are a number of issues which are currently active areas of research. These include the presence of singularities and discontinuities (especially in hyperbolic problems), oscillatory grid convergence, multi-scale models, and approaches for estimating discretization errors on coarse grids which are not in the asymptotic range.

### *8.8.1 Singularities and discontinuities*

Practical scientific computing applications are fraught with singularities and discontinuities. In some cases, they may take the form of relatively weak singularities/discontinuities such as discontinuous surface curvature in a linear elasticity analysis. Examples of strong singularities/discontinuities include shock waves in inviscid flow, the leading edge region in the flow over a sharp-nosed body, and the interface between two different materials in thermal or structural analysis problems. Nature tends to "smooth out" singularities such as the sharp leading edge that takes a more rounded appearance under high magnification or the shock wave that actually transports mass, momentum, and energy in the upstream direction via molecular motion. However, singularities and discontinuities in mathematical models are much more common than is often recognized, especially in idealized geometries and simplified mathematical models. Some examples include flow over a surface with discontinuous slope or curvature, loading of a structure with point loads, and loading of a plate with angled holes.

The presence of singularities and discontinuities can adversely impact our ability to obtain reliable estimates of the discretization error as well as our ability to compute the numerical solutions themselves. This is because all of the discretization error estimators (and most discretization schemes) require that the solutions be continuous and differentiable. Nevertheless, we still must compute numerical solutions to these problems and then estimate the discretization error in those solutions.

Work by Carpenter and Casper (1999) suggests that, for the Euler equations on sufficiently refined meshes, the simulation of flows containing shock waves will reduce to first-order accuracy, regardless of the formal order of the numerical scheme employed. This reduction to first-order accuracy in the presence of shock waves has also been observed by Roy (2001, 2003) for laminar and inviscid hypersonic flows. In these latter cases, flux limiters were employed to reduce a formally second-order accurate scheme to first order at a shock wave to prevent numerical oscillations.

Banks *et al.* (2008) provide an excellent explanation of why the formal order of accuracy of any discretization scheme will reduce to first-order accuracy or lower in the presence of discontinuities. They consider the inviscid Euler equations for compressible flows. The Euler equations admit discontinuous solutions such as shock waves, contact discontinuities, and slip lines. However, in the absence of viscosity which enforces the entropy condition in viscous flows, these discontinuous solutions are not unique (i.e., the Euler equations will admit infinitely many jump conditions). It is only in the limit of a vanishing viscosity that the correct Rankine–Hugoniot jump conditions are met. Note that this nonuniqueness generally does not pose a problem for numerical solutions to the Euler equations due to the presence of numerical viscosity in the form of even spatial derivatives in the truncation error. For nonlinear discontinuities such as shock waves, numerical schemes reduce down to first-order accuracy. For linear discontinuities (i.e., linearly degenerate waves) such as contact discontinuities and slip lines, Banks *et al.* (2008) perform a truncation error analysis to show that most discretization schemes reduce their formal order of accuracy to $p/(p+1)$, where $p$ is the formal order of the method for smooth problems. They also provide numerical examples to confirm their theoretical analysis.

Regardless of the discretization scheme employed, the reduction of the order of accuracy to first order in the presence of discontinuities results in a numerical scheme that is of mixed order. In this case, the scheme will be second-order accurate (or higher) in the smooth regions, but locally first-order accurate at the shock wave. While technically this scheme is formally first-order accurate for flows with shocks, there may be a significant range of mesh resolutions where the solutions exhibit second-order accuracy. This mixed order behavior can lead to non-monotone convergence of the solutions with grid refinement, as is demonstrated below. In order to analyze mixed first- and second-order schemes, Roy (2001, 2003) proposed the following expansion for the discretization error:

$$f_k = \bar{f} + g_1 h_k + g_2 h_k^2. \tag{8.93}$$

In this approach, terms accounting for the formal order of accuracy in both smooth regions ($p = 2$) and nonsmooth regions ($p = 1$) are included. If three mesh levels are available, then Eq. (8.93) can be solved for the two unknown coefficients $g_1$ and $g_2$ as well as the estimated exact solution $\bar{f}$. Consider the case where three discrete solutions $f_k$ are available on fine ($k = 1$), medium ($k = 2$), and coarse ($k = 3$) meshes, where the grid refinement factor is held constant between mesh levels (i.e., $r_{12} = r_{23} = r$). If we further arbitrarily set $h_1 = 1$ (this will simply require the absorbing of constants in the $g_k$ coefficients), then the

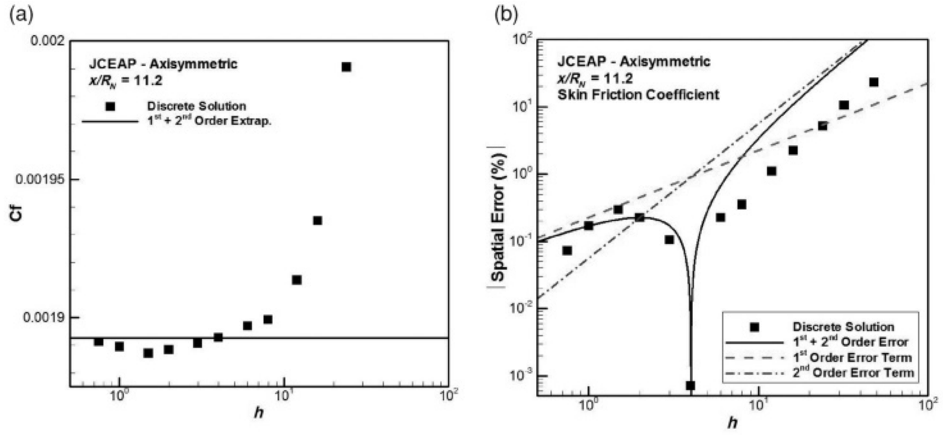

Figure 8.14 Dimensionless frictional resistance at one location on the body as a function of mesh spacing: (a) skin friction coefficient and (b) discretization error (from Roy, 2001).

expressions for the three unknowns become

$$g_1 = \frac{r^2(f_2 - f_1) - (f_3 - f_2)}{r(r-1)^2},$$
$$g_2 = \frac{(f_3 - f_2) - r(f_2 - f_1)}{r(r+1)(r-1)^2},$$
$$\bar{f} = f_1 + \frac{(f_3 - f_2) - (r^2 + r - 1)(f_2 - f_1)}{(r+1)(r-1)^2}. \quad (8.94)$$

These three expressions can be used not only to estimate the discretization error in the numerical solutions, but also to examine the behavior of the two different error terms in the discretization error expansion given by Eq. (8.93).

This mixed-order analysis method has been applied to the dimensionless frictional resistance (i.e., skin friction coefficient) at points along the surface of a sphere-cone geometry in laminar, hypersonic flow (Roy, 2001). The results are shown below in Figure 8.14a for the behavior of the skin friction coefficient at one location with mesh refinement. The solution displays nonmonotone behavior as the skin friction first decreases with mesh refinement and then begins to increase. Insight into why this nonmonotone behavior occurs can be found by examining the discretization error given in Figure 8.14b, which also shows the first- and second-order error terms from Eq. (8.93). In this case, both error terms have the same magnitude near $h = 4$, but they have opposite signs. This leads to error cancellation at $h = 4$ which manifests as a crossing of the estimated exact solution in Figure 8.14a. Since the two terms have opposite signs, when the mesh is sufficiently refined such that the first order term dominates, the skin friction coefficient displays a local minimum when plotted versus the mesh spacing $h$.

### *8.8.2 Oscillatory convergence with mesh refinement*

There has been much discussion in recent years on the presence of oscillatory convergence of numerical solutions with mesh refinement (e.g., see Coleman *et al.*, 2001; Celik *et al.*, 2005; and Brock, 2007). However, examination of the Taylor series expansion of the numerical solution about the exact solution the governing partial differential equations in terms of the mesh spacing parameter as given in Eq. (8.47), or even the more general series expansion from Eq. (8.48), shows that there can be only a single term that dominates the expansion *in the limit* as $h \to 0$. While this debate may not yet be settled, our opinion is that there can be no oscillatory convergence with mesh refinement in the asymptotic grid convergence range. The solution behavior being interpreted as oscillatory convergence is mostly likely due to a failure to achieve the asymptotic range (as discussed in Section 8.8.1), nonuniform refinement of the mesh (Eca and Hoekstra, 2002), or the presence of nonnegligible iterative and/or round-off errors. In the case of the former, we do not dispute the fact that the asymptotic range may be extremely difficult to achieve for complex scientific computing problems.

### *8.8.3 Multi-scale models*

A multi-scale model is one that resolves different physical phenomena at different length and/or time scales. For some multi-scale models, the governing equations actually change as the mesh is refined, thus making it difficult to separate out issues of verification (mathematics) from those of validation (physics). A classic example of a multi-scale model is the large eddy simulation (LES) of turbulent flows, which generally involves a spatial filtering operation to remove small-scale turbulent structures. The length scale used in this spatial filtering thus explicitly appears in the mathematical model. In most LES computations, this length scale is tied to the mesh spacing. Thus as one refines the mesh, it is equivalent to filtering the Navier–Stokes equations with a different filter length scale (i.e., a different mathematical model). The rigorous assessment of numerical errors and physical modeling errors thus becomes extremely difficult since both are tied to the mesh spacing. One possible remedy for this problem is to fix the filter width at the coarse grid spacing while performing mesh refinement to assess numerical errors (e.g., see Moin, 2007).

### *8.8.4 Coarse grid error estimators*

For scientific computing simulations where the physics and/or geometry are complex, the meshes are often under-resolved due to computational resource limitations. This is especially true for 3-D time-dependent problems. The ideal discretization error estimator is one that not only provides consistent error estimates as the mesh is refined, but also provides reliable error estimates on under-resolved coarse grids which are outside of the asymptotic grid convergence range. Although often buried deep within the details, all of the discretization error estimation methods discussed in this chapter require the solution (or solutions) to be within the asymptotic range. The reliability of discretization error estimators when the discrete solutions are not in (and possibly far from) the asymptotic range is an area for further investigation.

## 8.9 References

Ainsworth, M. and J. T. Oden (1997). A posteriori error estimation in finite element analysis, *Computer Methods in Applied Mechanics and Engineering*. **142**(1–2), 1–88.

Ainsworth, M. and J. T. Oden (2000). *A Posteriori Error Estimation in Finite Element Analysis*, New York, Wiley Interscience.

Akin, J. E. (2005). *Finite Element Analysis with Error Estimators*, Burlington, Elsevier.

Babuska, I. and A. Miller (1984). Post-processing approach in the finite element method – Part 3: A posteriori error estimates and adaptive mesh selection, *International Journal for Numerical Methods in Engineering*. **20**(12), 2311–2324.

Babuska, I. and W. C. Rheinboldt (1978a). A posteriori error estimates for the finite element method, *International Journal for Numerical Methods in Engineering*. **12**, 1597–1615.

Babuska, I. and W. C. Rheinboldt (1978b). Error estimates for adaptive finite element computations, *SIAM Journal of Numerical Analysis*. **15**(4), 736–754.

Babuska, I., O. C. Zienkiewicz, J. Gago, E. R. Oliveira (1986). *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, Chichester, Wiley.

Babuska, I., T. Strouboulis, and C. S. Upadhyay (1994). A model study of the quality of a posteriori error estimators for linear elliptic problems. Error estimation in the interior of patchwise uniform grids of triangles, *Computer Methods in Applied Mechanics and Engineering*. **114**(3–4), 307–378.

Babuska, I., T. Strouboulis, T. Gangaraj, and C. S. Upadhyay (1997). Pollution error in the h-version of the finite element method and local quality of the recovered derivative, *Computer Methods in Applied Mechanics and Engineering*. **140**(1–2), 1–37.

Baker, T. J. (2005). *On the Relationship between Mesh Refinement and Solution Accuracy*, AIAA Paper 2005–4875.

Bank, R. E. (1996). Hierarchical bases and the finite element method, *Acta Numerica*. **5**, 1–45.

Bank, R. R. and A. Weiser (1985). Some a posteriori error estimators for elliptic partial differential equations, *Mathematics of Computation*. **44**, 283–301.

Banks, J. W., T. Aslam, and W. J. Rider (2008). On sub-linear convergence for linearly degenerate waves in capturing schemes, *Journal of Computational Physics*. **227**, 6985–7002.

Barth, T. J. and M. G. Larson (2002). A-posteriori error estimation for higher order Godunov finite volume methods on unstructured meshes, In *Finite Volumes for Complex Applications III*, R. Herbin and D. Kroner (eds.), London, HERMES Science Publishing Ltd., 41–63.

Brock, J. S. (2007). *Bounded Numerical Error Estimates for Oscillatory Convergence of Simulation Data*, AIAA Paper 2007–4091.

Cadafalch, J., C. D. Perez-Segarra, R. Consul, and A. Oliva (2002). Verification of finite volume computations on steady-state fluid flow and heat transfer, *Journal of Fluids Engineering*. **24**, 11–21.

Carpenter, M. H. and J. H. Casper (1999). Accuracy of shock capturing in two spatial dimensions, *AIAA Journal*. **37**(9), 1072–1079.

Cavallo, P. A. and N. Sinha (2007). Error quantification for computational aerodynamics using an error transport equation, *Journal of Aircraft*. **44**(6), 1954–1963.

Celik, I., J. Li, G. Hu, and C. Shaffer (2005). Limitations of Richardson extrapolation and some possible remedies, *Journal of Fluids Engineering*. **127**, 795–805.

Cheng, Z. and M. Paraschivoiu (2004). A posteriori finite element bounds to linear functional outputs of the three-dimensional Navier–Stokes equations, *International Journal for Numerical Methods in Engineering*. **61**(11), 1835–1859.

Coleman, H. W., F. Stern, A. Di Mascio, and E. Campana (2001). The problem with oscillatory behavior in grid convergence studies, *Journal of Fluids Engineering*. **123**, 438–439.

Demkowicz, L., J. T. Oden, and T. Strouboulis (1984). Adaptive finite elements for flow problems with moving boundaries. Part I: Variational principles and a posteriori estimates, *Computer Methods in Applied Mechanics and Engineering*. **46**(2), 217–251.

Eca, L. and M. Hoekstra (2002). An evaluation of verification procedures for CFD applications, *24th Symposium on Naval Hydrodynamics*, Fukuoka, Japan, July 8–13, 2002.

Eca, L. and M. Hoekstra (2009a). Error estimation based on grid refinement studies: a challenge for grid generation, *Congress on Numerical Methods in Engineering*, Barcelona, Spain, June 29–July 2, 2009.

Eca, L. and M. Hoekstra (2009b). Evaluation of numerical error estimation based on grid refinement studies with the method of the manufactured solutions, *Computers and Fluids*. **38**, 1580–1591.

Eca, L., G. B. Vaz, J. A. C. Falcao de Campos, and M. Hoekstra (2004). Verification of calculations of the potential flow around two-dimensional foils, *AIAA Journal*. **42**(12), 2401–2407.

Eca, L., M. Hoekstra, and P. Roache (2005). *Verification of Calculations: an Overview of the Lisbon Workshop*, AIAA Paper 2005–4728.

Eriksson, K. and C. Johnson (1987). Error-estimates and automatic time step control for nonlinear parabolic problems. Part 1, *SIAM Journal of Numerical Analysis*. **24**(1), 12–23.

Estep, D., M. Larson, and R. Williams (2000). *Estimating the Error of Numerical Solutions of Systems of Nonlinear Reaction-Diffusion Equations*, Memoirs of the American Mathematical Society, Vol. **146**, No. 696, Providence, American Mathematical Society.

Fehlberg, E. (1969). *Low-Order Classical Runge-Kutta Formulas with Step Size Control and their Application to some Heat Transfer Problems*, NASA Technical Report 315, National Aeronautics and Space Administration, July 1969.

Ferziger, J. H. and M. Peric (1996). Further discussion of numerical errors in CFD, *International Journal for Numerical Methods in Fluids*. **23**(12), 1263–1274.

Ferziger, J. H. and M. Peric (2002). *Computational Methods for Fluid Dynamics*, 3rd edn., Berlin, Springer-Verlag.

Garbey, M. and W. Shyy (2003). A least square extrapolation method for improving solution accuracy of PDE computations, *Journal of Computational Physics*. **186**(1), 1–23.

Hirsch, C. (1990). *Numerical Computation of Internal and External Flows: Volume 2, Computational Methods for Inviscid and Viscous Flows*, Chichester, Wiley.

Hirsch, C. (2007). *Numerical Computation of Internal and External Flows: the Fundamentals of Computational Fluid Dynamics*, 2nd edn., Oxford, Butterworth-Heinemann.

Huebner, K. H. (2001). *The Finite Element Method for Engineers*, New York, Wiley.

Huebner, K. H., D. L. Dewhirst, D. E. Smith, and T. G. Byrom (2001). *The Finite Element Method of Engineers*, 4th edn., New York, John Wiley and Sons.

Hughes, T. J. R. (2000). *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, 2nd edn., Mineola, Dover.

Jameson, A. (1988). Aerodynamic design via control theory, *Journal of Scientific Computing*. **3**(3), 233–260.

Jameson, A., W. Schmidt, and E. Turkel (1981). *Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes*, AIAA Paper 81–1259.

Johnson, C. and P. Hansbo (1992). Adaptive finite element methods in computational mechanics, *Computer Methods in Applied Mechanics and Engineering*. **101**(1–3), 143–181.

Kamm, J. R., W. J. Rider, and J. S. Brock (2003). *Combined Space and Time Convergence Analyses of a Compressible Flow Algorithm*, AIAA Paper 2003–4241.

King, M. L., M. J. Fisher, and C. G. Jensen (2006). A CAD-centric approach to CFD analysis with discrete features, *Computer-Aided Design & Applications*. **3**(1–4), 279–288.

Knight, D. D. (2006). *Elements of Numerical Methods for Compressible Flows*, New York, Cambridge University Press.

Moin, P. (2007). Application of high fidelity numerical simulations for vehicle aerodynamics, *The Aerodynamics of Heavy Vehicles II: Trucks, Buses and Trains*, Tahoe City, California, August 26–31, 2007.

Morton, K. W. and D. F. Mayers (2005). *Numerical Solution of Partial Differential Equations: an Introduction*, 2nd edn., New York, Cambridge University Press.

Oden, J. T. and J. N. Reddy (1976). *An Introduction to the Mathematical Theory of Finite Elements*, New York, Wiley.

Paraschivoiu, M., J. Peraire, and A. T. Patera (1997). A posteriori finite element bounds for linear functional outputs of elliptic partial differential equations, *Computer Methods in Applied Mechanics and Engineering*. **150**(1–4), 289–312.

Pelletier, D. and P. J. Roache (2006). Chapter 13: Verification and validation of computational heat transfer, in *Handbook of Numerical Heat Transfer*, 2nd edn., W. J. Minkowycz, E. M. Sparrow, and J. Y. Murthy, eds., Hoboken, NJ, Wiley.

Pierce, N. A. and M. B. Giles (2000). Adjoint recovery of superconvergent functionals from PDE approximations, *SIAM Review*. **42**(2), 247–264.

Potter, D. L., F. G. Blottner, A. R. Black, C. J. Roy, and B. L. Bainbridge (2005). *Visualization of Instrumental Verification Information Details (VIVID): Code Development, Description, and Usage*, SAND2005–1485, Albuquerque, NM, Sandia National Laboratories.

Rannacher, R. and F. T. Suttmeier (1997). A feed-back approach to error control in finite element methods: application to linear elasticity, *Computational Mechanics*. **19**(5), 434–446.

Richards, S. A. (1997). Completed Richardson extrapolation in space and time, *Communications in Numerical Methods in Engineering*. **13**, 1997, 573–582.

Richardson, L. F. (1911). The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam, *Philosophical Transactions of the Royal Society of London*. Series A, Containing Papers of a Mathematical or Physical Character. **210**, 307–357.

Richardson, L. F. (1927). The deferred approach to the limit. Part I. Single lattice, *Philosophical Transaction of the Royal Society of London*. Series A, Containing Papers of a Mathematical or Physical Character. **226**, 299–349.

Richtmyer, R. and K. Morton (1967). *Difference Methods for Initial-Value Problems*, 2nd edn., New York, Interscience Publishers.

Roache, P. J. (1994). Perspective: a method for uniform reporting of grid refinement studies, *Journal of Fluids Engineering*. **116**, 405–413.

Roache, P. J. (1998). *Verification and Validation in Computational Science and Engineering*, Albuquerque, NM, Hermosa Publishers.

Roache, P. J. (2003a). Conservatism of the grid convergence index in finite volume computations on steady-state fluid flow and heat transfer, *Journal of Fluids Engineering*. **125**(4), 731–732.

Roache, P. J. (2003b). Criticisms of the ''correction factor'' verification method, *Journal of Fluids Engineering*. **125**(4), 732–733.

Roache, P. J. (2009). Private communication, July 13, 2009.

Roache, P. J. and P. M. Knupp (1993). Completed Richardson extrapolation, *Communications in Numerical Methods in Engineering*. **9**(5), 365–374.

Roy, C. J. (2001). *Grid Convergence Error Analysis for Mixed-Order Numerical Schemes*, AIAA Paper 2001–2606.

Roy, C. J. (2003). Grid convergence error analysis for mixed-order numerical schemes, *AIAA Journal*. **41**(4), 595–604.

Roy, C. J. (2005). Review of code and solution verification procedures for computational simulation, *Journal of Computational Physics*. **205**(1), 131–156.

Roy, C. J. (2009). *Strategies for Driving Mesh Adaptation in CFD*, AIAA Paper 2009–1302.

Roy, C. J. (2010). *Review of Discretization Error Estimators in Scientific Computing*, AIAA Paper 2010–126.

Roy, C. J. and F. G. Blottner (2003). Methodology for turbulence model validation: application to hypersonic transitional flows, *Journal of Spacecraft and Rockets*. **40**(3), 313–325.

Roy, C. J., C. J. Heintzelman, and S. J. Roberts (2007). *Estimation of Numerical Error for 3D Inviscid Flows on Cartesian Grids*, AIAA Paper 2007–0102.

Salas, M. D. (2006). Some observations on grid convergence, *Computers and Fluids*. **35**, 688–692.

Salas, M. D. and H. L. Atkins (2009). On problems associated with grid convergence of functionals, *Computers and Fluids*. **38**, 1445–1454.

Shih, T. I.-P. and Y. C. Qin (2007). *A Posteriori Method for Estimating and Correcting Grid-Induced Errors in CFD Solutions Part 1: Theory and Method*, AIAA Paper 2007–100.

Shih, T. I.-P., and B. R. Williams (2009). *Development and Evaluation of an A Posteriori Method for Estimating and Correcting Grid-Induced Errors in Solutions of the Navier-Stokes Equations*, AIAA Paper 2009–1499.

Sonar, T. (1993). Strong and weak norm refinement indicators based on the finite element residual for compressible flow computation: I. The steady case, *Impact of Computing in Science and Engineering*. **5**(2), 111–127.

Stewart, J. R. and T. J. R. Hughes (1998). A tutorial in elementary finite element error analysis: a systematic presentation of a priori and a posteriori error estimates, *Computer Methods in Applied Mechanics and Engineering*. **158**(1–2), 1–22.

Stern, F., R. V. Wilson, H. W. Coleman, and E. G. Paterson (2001). Comprehensive approach to verification and validation of CFD simulations – Part I: Methodology and procedures, *ASME Journal of Fluids Engineering*. **123**(4), 793–802.

Szabo, B. A. and I. Babuska (1991). *Finite Element Analysis*, New York, Wiley.

Tannehill, J. C., D. A. Anderson, and R. H. Pletcher (1997). *Computational Fluid Mechanics and Heat Transfer*, 2nd edn., Philadelphia, Taylor and Francis.

Thompson, J. F., Z. U. A. Warsi, and C. W. Mastin (1985). *Numerical Grid Generation: Foundations and Applications*, New York, Elsevier. (www.erc.msstate.edu/publications/gridbook)

Venditti, D. A. and D. L. Darmofal (2000). Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one dimensional flow, *Journal of Computational Physics*. **164**, 204–227.

Venditti, D. A. and D. L. Darmofal (2002). Grid adaptation for functional outputs: application to two-dimensional inviscid flows, *Journal of Computational Physics*. **176**, 40–69.

Venditti, D. A. and D. L. Darmofal (2003). Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows, *Journal of Computational Physics*. **187**, 22–46.

Wahlbin, L. B. (1995). *Superconvergence in Galerkin Finite Element Methods*, Volume 1605 of Lecture Notes in Mathematics, Springer-Verlag, Berlin.

Whiteman, J. R. (1994). *The Mathematics of Finite Element and Applications: Highlights 1993*, New York, Wiley.

Xing, T. and F. Stern (2009). *Factors of Safety for Richardson Extrapolation*, IIHR Hydroscience and Engineering Technical Report No. 469, March 2009.

Zhang, X. D., J.-Y. Trepanier, and R. Camarero (2000). A posteriori error estimation for finite-volume solutions of hyperbolic conservation laws, *Computer Methods in Applied Mechanics and Engineering*. **185**(1), 1–19.

Zhang, Z. and A. Naga (2005). A new finite element gradient recovery method: superconvergence property, *SIAM Journal of Scientific Computing*. **26**(4), 1192–1213.

Zienkiewicz, O. C., R. L. Taylor, and J. Z. Zhu (2005). *The Finite Element Method : Its Basis and Fundamentals*, 6th edn., Oxford, Elsevier.

Zienkiewicz, O. C. and J. Z. Zhu (1987). A simple error estimator and adaptive procedure for practical engineering analysis, *International Journal for Numerical Methods in Engineering*. **24**, 337–357.

Zienkiewicz, O. C. and J. Z. Zhu (1992). The superconvergent patch recovery and a posteriori error estimates, Part 2: Error estimates and adaptivity, *International Journal for Numerical Methods in Engineering*. **33**, 1365–1382.