ordering of points, each node is identified with an index l, which is also the relative storage location. In this notation the equation (3.44) can be written

$$A_{\mathrm{W}}\phi_{\mathrm{W}} + A_{\mathrm{S}}\phi_{\mathrm{S}} + A_{\mathrm{P}}\phi_{\mathrm{P}} + A_{\mathrm{N}}\phi_{\mathrm{N}} + A_{\mathrm{E}}\phi_{\mathrm{E}} = Q_{\mathrm{P}} \,, \tag{3.45}$$

where the index l, which indicated rows in Eq. (3.44), is understood, and the index indicating column or location in the vector has been replaced by the corresponding letter. We shall use this shorthand notation from now on. When necessary for clarity, the index will be inserted. A similar treatment applies to three-dimensional problems.

For block-structured and composite grids, this structure is preserved within each block, and the solvers for regular structured grids may be used. This is discussed further in Chap. 5.

For unstructured grids, the coefficient matrix remains sparse, but it no longer has banded structure. For a 2D grid of quadrilaterals and approximations that use only the four nearest neighbor nodes, there are only five non-zero coefficients in any column or row. The main diagonal is full and the other non-zero coefficients lie within a certain range of the main diagonal but not necessarily on definite diagonals. A different type of iterative solver must be used for such matrices; they will be discussed in Chap. 5. The storage scheme for unstructured grids will be introduced in Chap. 8, since such grids are used mostly in complex geometries with the FV method.

3.9 Discretization Errors

Since the discretized equations represent approximations to the differential equation, the exact solution of the latter, which we shall denote by Φ, does not satisfy the difference equation. The imbalance, which is due to truncation of the Taylor series, is called *truncation error*. For a grid with a reference spacing h, the truncation error τ_h is defined as:

$$\mathcal{L}(\Phi) = L_h(\Phi) + \tau_h = 0 \,, \tag{3.46}$$

where $\mathcal{L}$ is a symbolic operator representing the differential equation and L_h is a symbolic operator representing the algebraic equation system obtained by discretization on grid h, which is given by Eq. (3.43).

The exact solution of the discretized equations on grid h, ϕ_h, satisfies the following equation:

$$L_h(\phi_h) = (A\phi - Q)_h = 0 \,. \tag{3.47}$$

It differs from the exact solution of the partial differential equation by the *discretization error*, ϵ_h^{d}, i.e.:

$$\Phi = \phi_h + \epsilon_h^{\mathrm{d}} \,. \tag{3.48}$$

From Eqs. (3.46) and (3.47) one can show that the following relation holds for linear problems:

$$L_h(\epsilon_h^{\mathrm{d}}) = -\tau_h \ . \tag{3.49}$$

This equation states that the truncation error acts as a source of the discretization error, which is convected and diffused by the operator L_h. Exact analysis is not possible for non-linear equations, but we expect similar behavior; in any case, if the error is small enough, we can locally linearize about the exact solution and what we will say in this section is valid. Information about the magnitude and distribution of the truncation error can be used as a guide for grid refinement and can help achieve the goal of having the same level of the discretization error everywhere in the solution domain. However, as the exact solution Φ is not known, the truncation error cannot be calculated exactly. An approximation to it may be obtained by using a solution from another (finer or coarser) grid. The estimate of the truncation error thus obtained is not always accurate but it serves the purpose of pointing to regions that have large errors and need finer grids.

For sufficiently fine grids, the truncation error (and the discretization error as well) is proportional to the leading term in the Taylor series:

$$\epsilon_h^{\mathrm{d}} \approx \alpha h^p + H \ , \tag{3.50}$$

where H stands for higher-order terms and α depends on the derivatives at the given point but is independent of h. The discretization error can be estimated from the difference between solutions obtained on systematically refined (or coarsened) grids. Since the exact solution may be expressed as (see Eq. (3.48)):

$$\Phi = \phi_h + \alpha h^p + H = \phi_{2h} + \alpha (2h)^p + H \ , \tag{3.51}$$

the exponent p, which is the order of the scheme, may be estimated as follows:

$$p = \frac{\log\left(\dfrac{\phi_{2h} - \phi_{4h}}{\phi_h - \phi_{2h}}\right)}{\log 2} \ . \tag{3.52}$$

From Eq. (3.51) it also follows that the discretization error on grid h can be approximated by:

$$\epsilon_h^{\mathrm{d}} \approx \frac{\phi_h - \phi_{2h}}{2^p - 1} \ . \tag{3.53}$$

If the ratio of the grid sizes on successive grids is not two, the factor 2 in the last two equations needs to be replaced by that ratio (see Roache, 1994, for details on error estimates when the grid is not systematically refined or coarsened).

When solutions on several grids are available, one can obtain an approximation of Φ which is more accurate than the solution ϕ_h on the finest grid by adding the error estimate (3.53) to ϕ_h; this method is known as *Richardson extrapolation*, (Richardson, 1910). It is simple and, when the convergence is

monotonic, accurate. When a number of solutions are available, the process can be repeated to improve the accuracy further.

We have shown above that it is the rate at which the error is reduced when the grid is refined that matters, not the formal order of the scheme as defined by the leading term in the truncation error. Equation (3.52) takes this into account and returns the correct exponent p. This estimate of the order of a scheme is also a useful tool in code validation. If a method should be, say, second-order accurate but Eq. (3.52) finds that it is only first-order accurate, there is probably an error in the code.

The order of convergence estimated using Eq. (3.52) is valid only when the convergence is monotonic. Monotonic convergence can be expected only on sufficiently fine grids. We shall show in the examples that the error dependence on grid size may be irregular when the grid is coarse. Therefore, care should be taken when comparing solutions on two grids; when convergence is not monotonic, solutions on two consecutive grids may not differ much even though the errors are not small. A third grid is necessary to assure that the solution is really converged. Also, when the solution is not smooth, the error estimates obtained with Taylor series approximations may be misleading. For example, in simulations of turbulent flows, the solution varies on a wide range of scales and the order of the solution method may not be a good indicator of solution quality. In Sect. 3.10 it will be shown that the error of a fourth-order scheme may not be much smaller than of a second-order scheme for these types of simulations.

3.10 An Introduction to Spectral Methods

Spectral methods are a class of methods less suited for general purpose CFD codes than FV and FE methods but, as they are important in some applications (e.g. simulation of turbulence), they are briefly described here. For a more complete description of them, see the book by Canuto et al. (1987).

3.10.1 Basic Concept

In spectral methods, spatial derivatives are evaluated with the aid of Fourier series or one of their generalizations. The simplest spectral method deals with periodic functions specified by their values at a uniformly spaced set of points. It is possible to represent such a function by a *discrete* Fourier series:

$$f(x_i) = \sum_{q=-N/2}^{N/2-1} \hat{f}(k_q)\, \mathrm{e}^{\mathrm{i}k_q x_i} \,, \tag{3.54}$$

where $x_i = i\,\Delta x, \quad i = 1, 2, \ldots N$ and $k_q = 2\pi q/\Delta x\, N$. Equation (3.54) can be inverted in a surprisingly simple way: