conditions for the vorticity must be created at portions of the boundary where the velocity is specified. There are also problems with the accuracy of the discrete solution. See Guevremont *et al.* [1988]; Gunzburger, Mundt, and Peterson [1989]; and Osswald, Ghia, and Ghia [1988] and the references cited therein for discussions of the faults and virtues of the velocity–vorticity formulation and its numerical approximation.

Test Problems. Anytime one develops a numerical algorithm one would like to implement it in the form of a working code so that the algorithm's accuracy and efficiency can be demonstrated through computational experiments. In order to generate numbers, one must ultimately choose a particular physical problem for which the code provides a numerical simulation. There are a wide variety of such test problems being used. Here we discuss the relative merits of some of these.

Perhaps the most popular test problem is the driven cavity problem. Here the flow domain is a square (or cube in three dimensions). Along all walls except the top one, the velocity is required to vanish. Along the top wall the normal velocity component vanishes and the tangential components are prescribed constants. The Reynolds number for the flow, for a fluid of given dynamic viscosity, is determined by the size of the box and the magnitude of the nonvanishing velocity components at the boundary. A difficulty associated with the driven cavity problem is that the flow contains (at the corners) strong, nonphysical singularities. The effect of these singularities is mitigated in various ways by smoothing out the transition between the boundary conditions at the corners. In spite of these difficulties, the driven cavity problem is too easy a test for candidate algorithms for the numerical simulation of viscous incompressible flows. The reason for this is that it is rather easy to generate solutions that have the global features one would expect in such a flow. Furthermore, since the problem is for the most part not physically realizable, one cannot compare the numerical solution with meaningful experimental data.

The second most popular test problem is the flow over a backward facing step. The flow configuration is sketched in Figure 21.1, where the left boundary is an inflow and the right boundary is an outflow. The bottom boundary is a solid wall, while the top boundary may be a wall or a boundary at which the flow field is essentially inviscid.

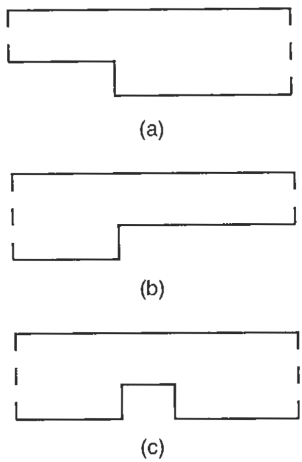

Figure 21.1
Three simple geometric configurations often used in test problems. (a) Backward facing step, (b) forward facing step, and (c) full step.

For the most part, the top boundary is chosen to be a solid wall. This problem is also too easy a test, mainly because certain benign features of the flow, e.g., the position of the reattachment point behind the step, scales with the Reynolds number; see Halim and Hafez [1984].

Perhaps the best test problems that retain the feature of geometrically simple flow domains are the forward-facing step and the full-step problems sketched in Figure 21.1. The distribution of boundary conditions is similar to that for the backward-facing step problem. These problems are realistic in the sense that they do not scale with the Reynolds number and that meaningful experimental data can be used for comparison purposes.

None of the problems discussed so far is solvable exactly and therefore one can ask how one should measure the quality of the numerical solution. Ideally, and where possible, one should compare with experimental data. However, this is not a foolproof measuring device since one seldom knows the accuracy of the experimental data itself. One popular method of determining the accuracy of a numerical solution is to use the "eyeball norm." A numerical simulation is good in the eyeball norm if one looks at a picture of the numerical flow field and concludes that it looks reasonable, e.g., there are no unexpected wiggles. Of course, this measure of the quality of a flow simulation, although popular, can be extremely misleading. For example, one can easily generate numerical flow fields whose gross features, e.g., recirculation regions, look reasonable but whose detailed features are wrong. A better way to judge the quality of the numerical solution is to compute using meshes of different sizes and show, using the type of norms introduced in our discussions, that convergence is apparent. Incidentally, the nonavailablity of exact solutions for most test problems points out the value of rigorous error estimates. After all, if we know *a priori* that the numerical solution converges to the (unknown) exact solution, and also know something about the rates of convergence, then we can have some confidence that we are producing meaningful numerical flow fields. Of course, the fly in the ointment is that error estimates are asymptotic in nature, i.e., they hold as the mesh size tends to zero, while actual computations are carried out using finite mesh sizes.

Ideally one would like to have exact solutions with which to compare one's numerical output. There are available many exact solutions of the Navier–Stokes equations, sometimes in the sense that they can be determined to arbitrary accuracy by solving nonlinear ordinary differential equations. Unfortunately, these solutions necessarily are ones that scale with the Reynolds number. Some, like fully developed Poiseuille flow, are exceedingly simple in that the nonlinear convection term vanishes. Others, such as Hiemenz and Hammel flow (Schlicting [1979]) are of more use. Although comparison with these solutions does not always provide definite information concerning the quality of a numerical simulation, a great amount of useful information can be so obtained. Typically, the flow domain for these solutions is unbounded, and exact solutions are obtained by solving nonlinear ordinary differential equations. A finite domain problem is obtained by truncating the flow domains. One is free to choose the shape of the artificial boundary so that, for example, one can also use these problems to test one's methods on flow domains having curved boundaries. Along the artificial boundaries created by the domain truncation process one can impose boundary conditions derived from the exact solution. However, one is free to choose any type of boundary condition one desires, e.g., the velocity, the stress, the vorticity, the pressure, or some combination of these. Thus, knowing an exact solution allows one to test how effective one's algorithms are in treating a variety of boundary conditions. In fact, along outflow portions of the artificial boundary one can test different (nonexact) outflow boundary conditions. Finally, the most obvious advantage of having an exact solution available is that one can precisely measure, using one's favorite norms, how good a numerical solution is.

Upwind, Petrov–Galerkin, and Streamwise Diffusion Methods. For the most part, the schemes we have discussed in the book fall into the category of "central difference schemes" in the sense that nowhere in the discretization processes is there any bias towards any particular direction. Thus, the finite difference realizations of our schemes would involve only central difference quotient approximations to derivatives. It is well known that such approximations yield results that are often hopelessly polluted by oscillations whenever one does not adequately resolve regions where the flow variables experience large variations, e.g., boundary layers. At the outset of this discussion, it should be pointed out that if one does resolve such regions by, e.g., mesh refinement, then finite element methods as described in the bulk of the book will yield meaningful approximations.

However, mesh refinement for the purpose of keeping a calculation stable is often thought of as wasteful, i.e., one wants accuracy considerations to govern the selection of grids. For this reason, computational fluid dynamicists have long advanced the idea of upwind differencing in the convection term of the momentum equation, i.e., biasing, in the direction opposite to the flow, the differencing of that term. Crude methods for accomplishing this stabilize computations by introducing large amounts of artificial viscosity. This is fine in many settings, e.g., inviscid flows, where one is not interested in the details of the flow that are seriously influenced by the effects of viscosity. However, if one is really interested in viscous phenomena such as skin friction, then the introduction of large amounts of artificial viscosity can be ruinous.

The finite difference community has spent much time and effort in developing more sophisticated upwind schemes, where we use that term to denote any scheme that, either explicitly or implicitly, introduces a bias into the discretization. The finite element community has likewise been involved in such schemes, often drawing from techniques developed for finite difference methods, but sometimes developing new approaches that stem inherently from finite element methodology.

There are a variety of ways to introduce a bias into the discretization. One way is to choose test functions that differ from the basis functions