Part III

Applications to Fluid Dynamics

Starting with [Bra73] and [SB77], an ever-increasing number of works have applied multigrid techniques to solve *steady-state* flow problems. Early investigations include works on transonic potential flows, such as [Arl78] , [Boe82], [BK83], [DH82], [MS83] , [Jam79], [SB77], [SZH83], [Bro82], [Cau83], [MR82] , [SC82] and [vdWvdVM83], the latter five treating 3-dimensional flows; works on the Stokes and incompressible Navier-Stokes equations, like [Bra73], [Bra80a], [BD79], [Din79], [Fuc82], [TF81], [Ver83] , [Ver84b], [WS80], [Wes77]; works on the Euler equations [Bra82c] , [Jam83], [Jes83], [Ni82] and on the compressible Navier-Stokes equations [Bra82c]. A survey of all this is not attempted here. Our purpose here is to trace a line of development which gradually leads from very simple equations to the most complicated ones, adding the difficulties step by step, but always maintaining the full multigrid efficiency; i.e., insisting on *solving every problem to $O(h^2)$ accuracy in just few work units*, where the work unit is the minimal amount of computer operations needed to express a discretization of the problem on a grid with meshsize h, and where the operations used can be fully parallelized (or vectorized) over the entire grid. *Minimal computer storage* is also maintained, i.e., a storage just a fraction more than needed to store the solution itself on grid h. Moreover, to show how these goals are achieved for the more complicated systems of equations, our emphasis here is on the treatment of *systems* of differential equations, although the line of development starts of course with simple scalar equations. In particular, the work on the scalar convection-diffusion problem [Bra81a] is a crucial step in that line, as will become clear in §19.3, not to mention the extensive work on the Poisson equation and on more general diffusion problems.

Most works mentioned above lag far behind the ideal performance, for various reasons (see discussion in §1.7). To achieve the goals stated above, many of the principles delineated in the previous two parts of this Guide are, and perhaps must be, used. Other principles described above have not yet been used, but they are available, ready to be added and enhance the power of the flow solvers presented in this part. This includes: methods of flexible local refinements and local coordinate curving (see §9); higher-order techniques (§10: the double discretization scheme of §10.2 is already used to obtain the mentioned O(h2) approximation in cases of inviscid or small-viscosity problems, but still higher orders are obtainable, if desired, for small extra work); procedures for further reducing computer storage (§8.7); the general approach of multigridding directly the real "outer" problem (e.g., the optimization or design problem for which the flow equations are solved; cf. §13); and the methods for efficiently treating sequences of many boundary-value problems and solving time-dependent problems (§15 and 16).

The work described in this part has been done in collaboration with Nathan Dinar and Ruth Golubev. Much of it has appeared before in [BD79] and [Bra82c].