Chapter 16

Time Dependent Problems

Several multigrid applications to evolution problems can be mentioned, including fast solvers to implicit equations, coarse-grid time steps, highly adaptable structures, high-order techniques and global conservation facilities.

One obvious application is to use fast multigrid solvers for solving the set of algebraic equations arising at each time step when implicit time differencing is employed. Such differencing is normally needed whenever a physical signal speed is considerably higher than the propagation speed of substantial changes in the solution. The latter speed determines the size of time steps δt we need for approximating the solution accurately, but with such δt and explicit differencing the numerical signal speed will be slower than the physical one, causing numerical instability. Using implicit equations and solving them by multigrid can be viewed as a way to inexpensively obtain high signal speeds by propagating information on coarse grids. Indeed, with a multigrid solver working on the correction problem (see §15), the cost of an implicit time step is comparable to that of an explicit one [BG91].

In many cases one can even do much better using techniques as in §15 above. For second-order parabolic problems, for example, significant changes in high-frequency components, whose wavelength is $O(h)$, occur only in very particular places such as

(A) Initially, for a short time interval, $0 \le t \le O(h^2)$;

(B) At distance $O(h)$ and time interval $O(h^2)$ from points where significant *changes* occur in boundary conditions or in forcing terms (source terms) of the equation.

At all other places significant high-frequency changes are induced by comparably significant low-frequency changes. Hence the frozen-τ technique,

with a special control for time-dependent problems [Bra79a, §3.9], can give us a solution with the fine-grid accuracy but where most of the time in most of the domain we use coarse grids only. The cost of an average step may then be far smaller than the cost of an explicit time-step.

For the heat equation in the infinite space and steady forcing terms, for example, one can show by Fourier analysis that marching from initial state to 90% steady-state, following the solution throughout with close to finest-grid accuracy, can in this way cost computational work equivalent to just 10 explicit time steps! The finest grid needs to be activated only in the first few time steps, and very rarely later [Gre92].

Notice that when we march (calculating smooth changes in the solution) on coarse grids we can also use large time steps with *explicit* differencing. When fully adapted grids are used there is no need for implicit differencing, because each range of components is in effect handled by a meshsize comparable to the wavelength and by a time-step corresponding to the propagation speed, so that no conflict arises between different characteristic speeds.

For problems with small parabolicity (e.g., parabolic singular perturbation to a "reduced" hyperbolic system), the above, technique can be superposed on an integrator of the reduced system (which may itself be based on a method of characteristics).

The multi-level techniques can also be applied to a parabolic *part* of the system, such as the implicit pressure equation in integrating Navier-Stokes equations [Bro82]. Here too, the techniques of §15 can further save a lot of fine-grid processing.

Whether the fast solver is used or not, the multi-level procedures can also give **highly flexible discretization structures**. Patches of finer grids with correspondingly finer time steps can be used in any part of the space-time domain, in a manner similar to §9.1. Anisotropic refinements, local coordinate transformations and rotated cartesian grids can be used as in §9.2, 9.3 and 9.4, all controlled by exchange-rate criteria (cf. 9.5, 9.6); but instead of criteria based on the τ_h^H of the approximate solution, criteria here will be based on recently accumulated changes in τ_h^H [Bra79a, §3.9].

In some problems, especially when integrating over long time periods, certain quantities, such as total mass, must strictly be conserved, otherwise the physics of the system would fundamentally change. Imposing such **global constraints**, with the corresponding freeing of some accuracy parameters in the difference equations, can easily be incorporated when a multigrid solver is used at each time step (see §5.7).

Finally, independently of the above techniques, one could use a multigrid procedure similar to §10.2 above to efficiently increase the approximation order of a stable discretization

$$L_0^h u_0^h(x,t) = 0, \qquad (t > 0) \tag{16.1}$$

of a time-dependent system, not necessarily linear, by using **coarse-grid**

defect corrections. Typically L_0^h is a simple low-order implicit operator, allowing simple integration. One wants to use a simple (e.g., central in time) higher-order operator L_1^h, which may be unstable, to raise the approximation order. This can be done by integrating the defect equation

$$L_0^H v^H = -I_h^H L_1^h u_0^h, \tag{16.2}$$

and then correcting

$$u_1^h = u_0^h + I_H^h v^H, \tag{16.3}$$

where H may either be coarser than h (coarse-grid defect correction) or $H = h$ (defect correction on the same grid). If the order of consistency of L_j^h is p_j, $(j = 0, 1)$, then for low-frequency components

$$|u_1^h - u| = O(h^{p_1} t + h^{p_0} H^{p_0} t^2), \tag{16.4}$$

where u is the differential solution and t is the time interval over which (16.1)–(16.3) is integrated from initial conditions $u_1^h(x,0) = u_0^h(x,0) = u(x,0)$.

The scheme (16.1)–(16.3) is always stable, since only the stable operators L_0^h and L_0^H are integrated. Notice that this is true only if u_0^h is integrated independently of u_1^h. The seemingly similar scheme, in which after each time step u_0^h is re-initialized by being replaced by the more accurate u_1^h, may well he unstable. On the other hand it does pay to re-initialize every $O(1)$ time interval, short enough to make the second term in (16.4) smaller than the first one.

The independent integration of u_0^h and v^h requires extra storage. By taking $H = 2h$ this extra storage becomes only a fraction of the basic storage (one time level of u_0^h), and the computational work is also just a fraction more than the work of integrating (16.1). These two-level schemes can be extended to more levels and more approximation orders. As in §10.2, such multigrid algorithms exploit the fact that the higher-order approximation L_1^h is desired only for sufficiently low frequencies; for the highest frequencies (where numerical instability occurs) L_0^h is in fact a better approximation than L_1^h.