Multigrid, Semi-Refinement and Fluid Flow

P.W. Hemker

1. Introduction

One of the continuous, major challenges in numerical analysis is the fast solution of partial differential equations (PDEs). Many different types of such equations exist, appearing in many areas of science and technology, e.g., when fluid flow problems have to be computed.

When PDEs are solved numerically, they have to be discretized, i.e. their solution, which is a set of functions defined over an area, is characterized by a set of N numbers, and the original differential equations are transformed into a system of N algebraic equations in which these N numbers are the unknowns. The difficulty is that, for an accurate approximation of the solution, the number N should be chosen large, which results in a large size of the system of algebraic equations.

For problems in which the solution is a function over a two-dimensional domain the size of the system can already be very large, if the solution is a function of three space variables, the size can be enormous.

To solve these large systems of equations, special techniques have been developed. Among these, the *multigrid method* is optimal in the sense that it is the only known approach by which the amount of computational work to solve the algebraic system is only proportional to N, the number of unknowns. For all other methods the amount of work grows faster than directly proportional with N.

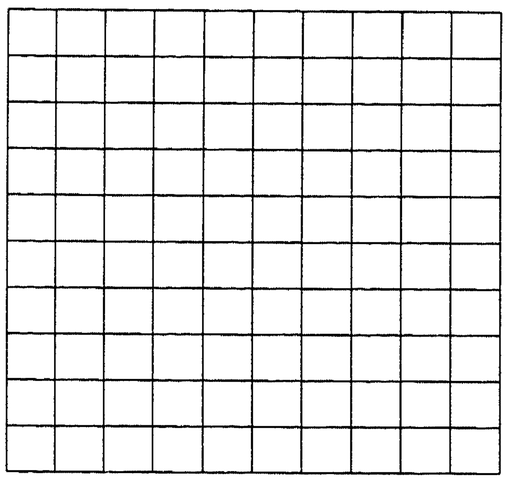

Figure 1. The domain Ω, discretized by a uniform mesh.

2. EXAMPLE

2.1. The problem

The basic model problem to demonstrate the value of multigrid has always been the Poisson equation on a square. This is a typical example for a general elliptic boundary value problem,

$$-\Delta u = f \quad \text{in} \quad \Omega = (0,1)^2\,; \quad u = 0 \quad \text{on} \quad \partial\Omega\,. \tag{2.1}$$

A uniform $n \times n$-mesh is placed over Ω, as shown in figure 1. This means that $n+1$ equidistant mesh-lines are drawn in the horizontal, and the same number in the vertical direction. The distance between the mesh-lines is called the mesh-width, $h = 1/n$.

The grid points are x_{ij}, where $0 \le i, j \le n$. If we want to approximate the solution of (2.1) numerically, discretisation is applied to (2.1) to get a set of linear equations:

$$AU = F\,, \tag{2.2}$$

where $F_{(i-1)(n+1)+j} = h^2 f(x_{ij})$, and A is a block tridiagonal matrix with a special structure. The system has $N = (n+1)^2$ equations and the same number of unknowns.

The element $U_{(i-1)(n+1)+j}$ of the solution vector U in the system of equations (2.2) represents the approximate solution of equation (2.1) at the point x_{ij}, i.e. $U_{(i-1)(n+1)+j} \approx u(x_{ij})$. The accuracy of this approximation depends on the type of discretisation and on h, the width of the mesh applied. This means that the approximation becomes more accurate if more mesh-lines are introduced. Typically, for a simple discretisation method, the error in the solution, $|U_{(i-1)(n+1)+j} - u(x_{ij})|$, is proportional to h^2. If higher accuracies are required, smaller values of h may be needed, i.e. a large number of

mesh-points may be necessary. Such large numbers of mesh-points give rise to very large systems (2.2), and the techniques used to solve such systems of moderate size (e.g., Gauss elimination) cannot be applied because the number of arithmetic operations (additions and multiplications) to compute the solution by these methods is proportional to N^3.

2.2. The solutions

For large systems of type (2.2), Gauss elimination would take too much time, even on present day's fastest computers, and different methods are used, that take advantage of the special properties of such equations. One such special property is, e.g., that although the system has $N = (n+1)^2$ unknowns, in each equation only a limited number of unknowns is involved, typically less than 10. This means that in the matrix A in (2.2) most entries are equal to zero, which reduces the amount of work considerably; but also other special properties of the matrix A can be used. All these special methods to solve discretized PDEs are iterative methods, where a first guess of the solution is improved step by step in an iteration process.

Until the sixties, simple relaxation methods were very popular. Here, all separate equations in (2.2) are scanned one by one, and each time when an equation is visited, the corresponding unknown is updated, based on the present information about the other unknowns. Such an approach was first mentioned by C.F. Gauss, in a letter to Chr.L. Gerling (26 December 1823), where he mentions: 'Das indirekte Verfahren lässt sich halb im Schlafe ausführen, oder man kann während desselben an andere Dinge denken'.

Later, in the seventies, more efficient iterative methods, based on the construction of Krylov spaces, appeared, such as the preconditioned conjugate gradient method, GMRES or—a more recent development—Bi-CG-Stab. Nowadays, these methods are the most popular ones to solve the very large systems. One reason is that these methods are relatively easy to implement in a computer program.

However, to restrict the amount of work to $\mathcal{O}(N)$, we have to resort to multigrid (MG) methods. These methods have a more complex structure. Invented in the sixties, they got the full attention of the numerical community not before 1980. A pioneering paper in the late seventies, [1], started the interest, and at present the multigrid method is well-accepted and in many fields it is successfully applied [3], see also figure 2.

3. MULTIGRID AND SEMI-COARSENING

The principle behind the MG technique is the fact that simple relaxation techniques only efficiently reduce the high-frequency errors on a mesh, and that the low-frequency errors can better be reduced by a discrete equation on a related coarser mesh with essentially less mesh points. (These errors may be compared to the disturbances of a quiet water surface, which in general

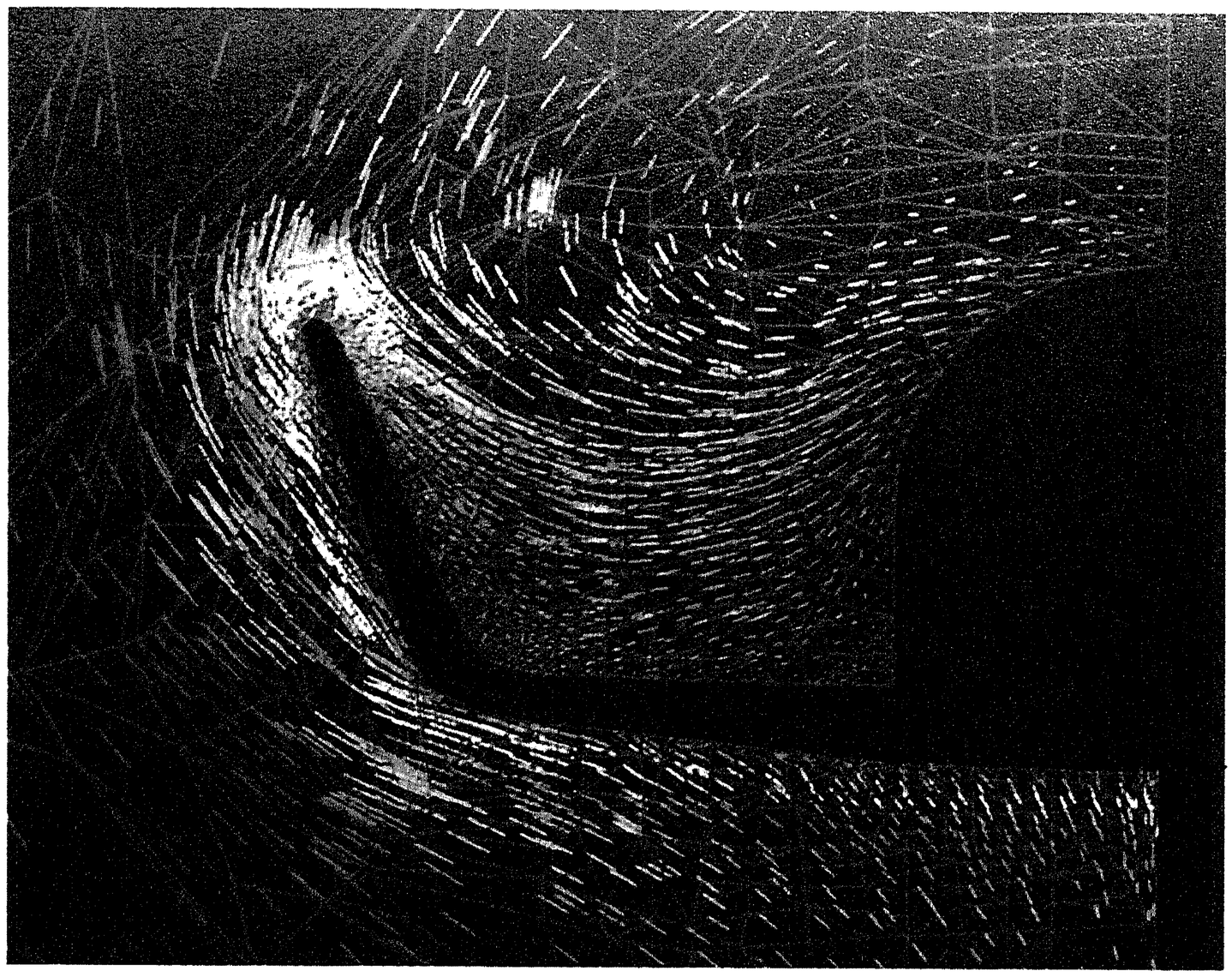

Figure 2. Streamline distribution around a half cross-section of the Hermes spaceplane. Applying multigrid methods, CWI studied numerical solutions of equations describing the flow around heavier-than-air craft as a part of the European space programme. Photo: Dassault.

consist of a spectrum ranging from short to long waves.) Now the MG method uses this principle recursively to solve the problem on the coarser meshes (see figure 3). All computational work together (on the coarse and

the fine meshes) to solve the differential problem as accurate as is possible on the finest mesh (with N mesh points) is still $\mathcal{O}(N)$.

It is well known how multigrid methods can be used for two-dimensional (2D) problems, and that the same techniques can be used for three-dimensional (3D) problems as well. One may even point to the fact that the total amount of work on the coarse grids is relatively smaller in the 3D-case, than in the 2D-case. However, the reverse side is that only a relatively small amount of error components can be annihilated by these coarse grid corrections. Still today, the consequence is that in the 3D-case powerful relaxation methods are required to reduce the total error with a sufficient efficiency.

One possible relaxation procedure is alternating plane-relaxation, in which all planes in the cube are visited by different orderings, and where for each

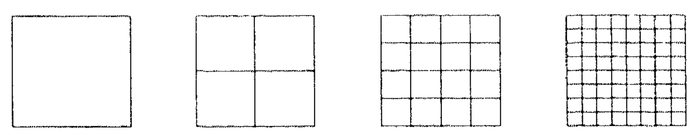

Figure 3. A classical sequence of grids in two dimensions.

plane a 2D sub-problem is solved (by a 2D MG method). This procedure is not very attractive, because there are many possibilities to order the planes in the cube, and a choice has to be made how these planes have to be visited. For a general problem such a choice is artificial, and the one choice may be better for the one problem while another choice can be advantageous in another situation. Such 3D-methods are also hard to vectorize or to parallelize so that we may have little advantage of new computer architectures.

However, there exists an alternative. Already for 2D fluid flow problems it had become clear that it is sometimes better to generate coarser grids, not by taking together a 2×2 set of four small cells to form one bigger cell, but to take together only 2 cells, so that a coarser mesh is obtained with a different mesh-size ratio. This is the principle of *semi-coarsening*. Here also we have the argument that the semi-coarsening is direction-dependent, and that there are more ways to assemble pairs of cells to form the coarse grids. But in the general, problem-independent case we may apply both semi-coarsenings at the same time. In that case the fine grid has two corresponding coarse grids. Now we have to study how the corrections from both coarse grids can cooperate to yield a good coarse grid correction for the solution on the finer grid.

We can approach the same technique from the other side. We may start with a coarse grid and make finer and finer grids, each time by halving grid cells into two finer cells (see figure 4). This principle of refinement can also be applied in three dimensions. In this case the number of possible grid refinements is even larger (see figure 5).

This approach of semi-refinement can be very powerful if it is combined with adaptive meshing, i.e., in all meshes only those cells are created that really contribute best to the reduction of the total error. Here the idea of hierarchical basis plays an important role, in order to combine the function approximations on the different grids into a single, unique representation.

4. Research at CWI

At CWI in recent years different applications have been studied, where MG was used to solve the discrete equations. Until 1994 all this work was concerned with problems in two space dimensions. Since 1984 fluid flow problems have been a major area of interest for MG investigations. First the

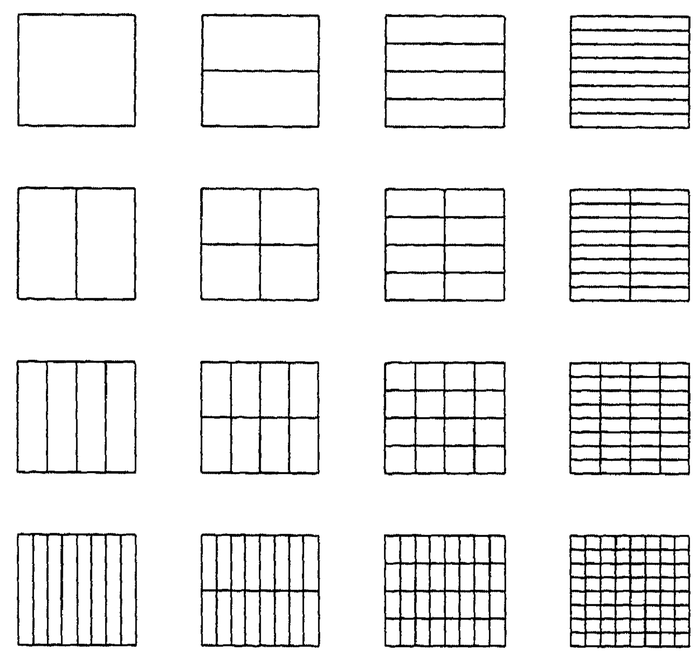

Figure 4. A family of semi-refined grids in two dimensions.

steady state Euler equations for inviscid compressible flow were considered. The equations differ essentially from (2.1) because the Euler equations are not elliptic, but—as time-dependent equations—hyperbolic. The approach to solve these is: to base the MG procedure on a finite-volume upwind discretisation. Later the same approach was extended for the compressible Navier-Stokes equations. In the steady case, these equations are elliptic, but for high Reynolds numbers (the interesting case in aerodynamical applications) they are singularly perturbed. This implies that thin layers may appear in the solution and that outside such layers the solution behaves very much like the Euler equations.

Steered by a contract with the European Space Agency, from 1988 to 1991, research was done for the special case of hypersonic flows. For a more recent contract, with a scientific board of the European Union, we studied structured adaptive methods for compressible flows. In these adaptive methods the refinement of the grids depends (automatically) on the features of the solution that is computed. Some examples will be given below. For more details we refer to [2].

At this moment the research is directed to the use of MG methods for 3D problems. A data structure has been developed to allow a well-structured coding of the self-adaptive semi-refinement algorithms. Procedures for self-adaptive representation of 3D-functions are now available, and the first experiments with 3D fluid flow problems have been carried out. In the mean time theoretical developments are continuing. Lately, Fourier analysis was used to compute the convergence rate of semi-coarsened MG algorithms. The computed rates were confirmed by numerical experiments.

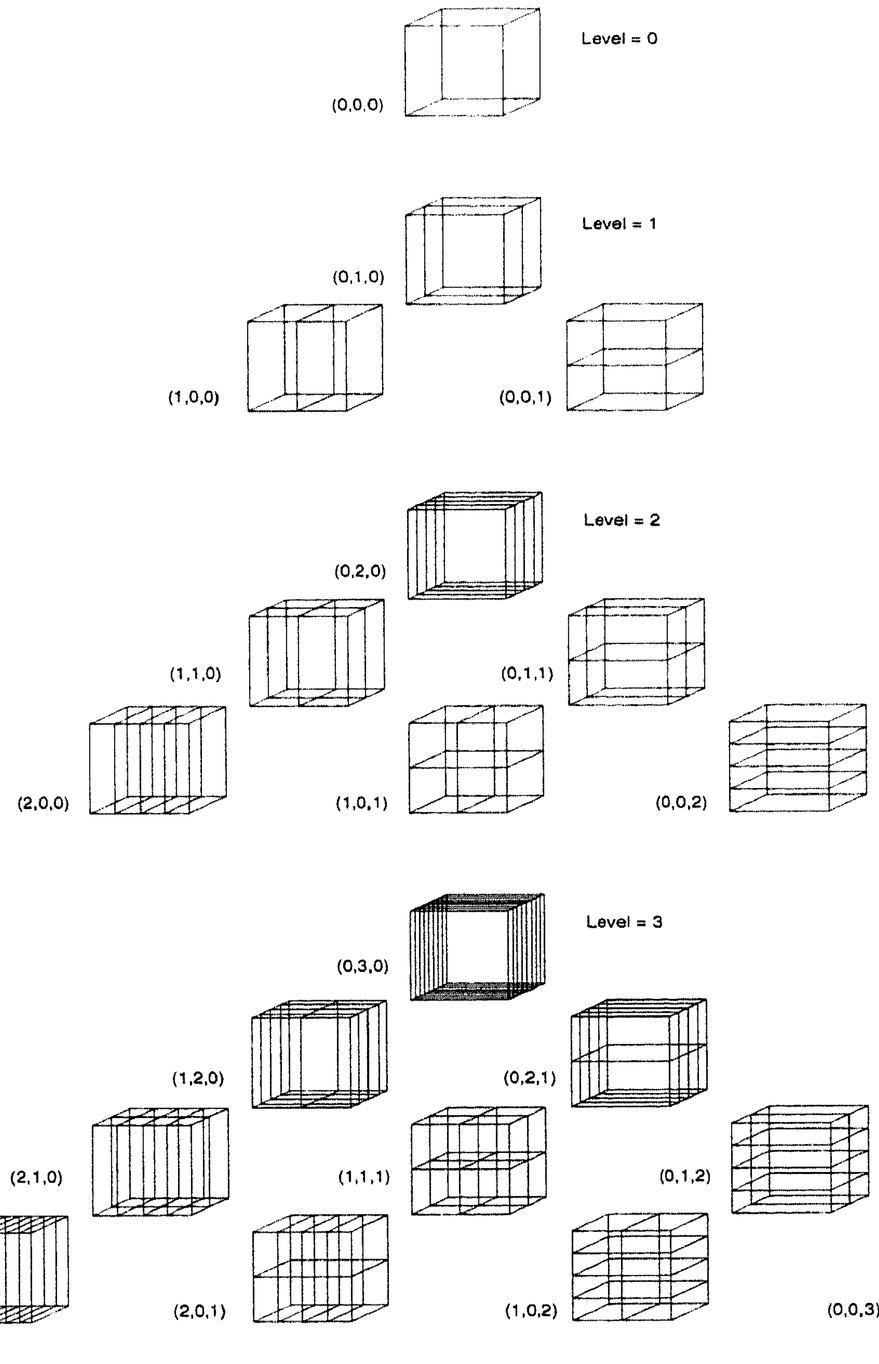

Figure 5. Semi-refinement of the cube. Grids on levels 0, 1, 2, and 3.

5. FLUID FLOW PROBLEMS

If the MG method is used, starting from a coarse mesh and constructing finer and finer meshes, this procedure can be continued until the approximate solution is sufficiently accurate. If the solution is not very regular, some parts of the domain will need more refinement than other parts. How far the refinement should be continued depends on the solution itself, and can be decided during the computation. In this way self-adaptive algorithms are developed, that minimize the number of gridpoints required, for a given precision. At CWI such algorithms have been studied in detail by H.T.M. van der Maarel. Algorithms have been developed for the Euler equations for compressible, inviscid flow, and also for the compressible

Navier-Stokes equations. An example of a solution of the Euler equations on a self-adaptive grid is given in figure 6, where the adapted grid and isolines for the Mach number distribution are shown for the flow over a NACA0012 airfoil.

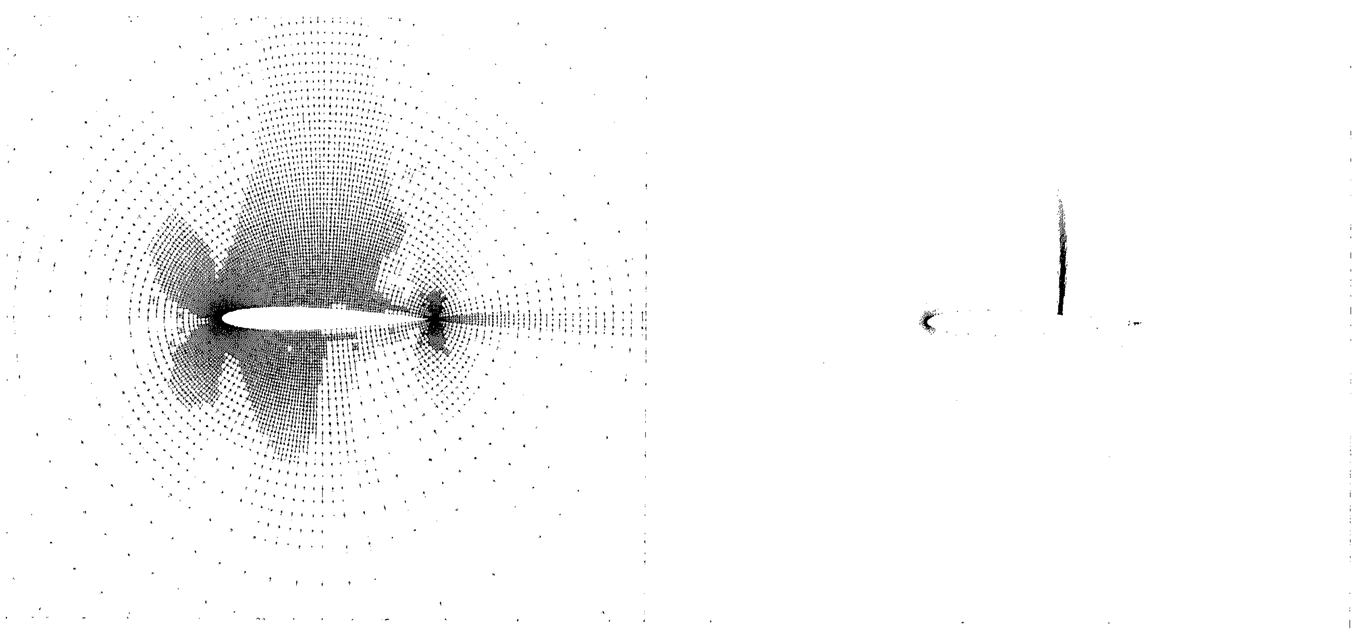

Figure 6. Euler solution for flow over an airfoil: the regular adaptive grid (left), and Mach number distribution (right).

The computation with the Navier-Stokes equations can be seen as an extension with diffusive terms of the same procedure for the Euler equations. Apart from the additional diffusive flux computations and boundary conditions, all techniques for the Navier-Stokes equations are the same as for the Euler flow computations. The convective part of the numerical flux is the same. For Navier-Stokes computations the flux is extended with a diffusive part, involving shear stresses and temperature gradients. The convective flux may be computed with either $\mathcal{O}(h)$ or $\mathcal{O}(h^2)$ accuracy. The diffusive part is always computed with $\mathcal{O}(h^2)$ accuracy.

Because the difference scheme for the diffusive part of the equations extends over a different set of grid cells compared to the convective part, much attention has to be paid to the different possible grid structures encountered in an adaptively refined Navier-Stokes grid.

The multigrid smoother approximately inverts the first-order accurate convective operator, extended with the second-order accurate diffusive terms. As with the Euler equations, the equations resulting from the second-order accurate discretisation are solved by defect correction.

An example is the computation of a flow along an adiabatic flat plate (see figure 7). The plate is located at $y = 0$, $-0.5 < x < 0.5$, and the Reynolds number (Re) equals 100. The result shown has been obtained with the first-order accurate discretisation and a refinement criterion based on the the cross-flow gradient of the velocity component times the mesh-width (the

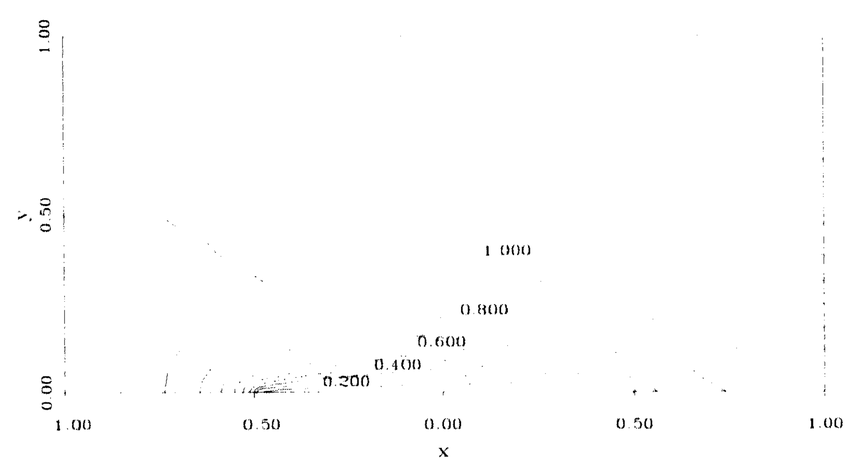

Figure 7. Navier-Stokes solution over a flat plate, Re=100, regular adaptive grid (above), and velocity in the x-direction (below).

undivided difference). Figure 7 shows the resulting grid when refinements are introduced where the undivided difference is larger than 0.1 ρ (above), and an iso-line plot of the computed velocity in the x-direction (below).

REFERENCES

1. A. BRANDT (1977). Multi-level adaptive solutions to boundary value problems. *Math. Comput. 31*, 333-390.
2. P.W. HEMKER, B. KOREN, W.M. LIOEN, M. NOOL, H.T.M. VAN DER MAAREL (1995). Review of an accurate and efficient multi-grid method for steady gas dynamics problems. *Computational Fluid Dynamics Review 1.*
3. P.W. HEMKER, P. WESSELING (eds.) (1993). *Multigrid methods IV*, volume 116 of *International Series of Numerical Mathematics*, Basel, 1994. Birkhäuser Verlag. Proceedings of the Fourth European Multigrid Conference, held in Amsterdam.