NEW TRENDS IN THE SOLUTION OF THE NAVIER-STOKES EQUATIONS FOR COMPRESSIBLE FLOWS

H. Deconinck, VKI
P.W. Hemker and B. Koren, CWI

Abstract

This paper reports on new trends in algorithm development for the numerical solution of the steady, 3-D, compressible Navier-Stokes equations. The principal goal of this project is to cure some of the weaknesses in existing state-of-the-art methods. Promising research results have been obtained already during the First Phase of the project[1]. This was achieved by developing:

- multi-D upwind discretization techniques for the convection terms, and
- solution-adaptive multigrid techniques for the solution of the discretized equations.

Research in these two areas is continued in the present Second Phase of the project. In the following two sections, we briefly discuss two specific techniques which are still in development in each of these two project areas: (i) multi-D. cell-vertex fluctuation-splitting techniques. and (ii) sparse-grid techniques (a new kind of multigrid techniques. well-suited for 3-D problems).

Multi-D, Cell-Vertex Fluctuation-Splitting Techniques

In many existing. compressible flow solvers. a basic building block is some solution method for the 1-D Riemann problem (the initial-boundary-value problem given by two constant gas states. separated by a discontinuity in one or more of the state variables). To solve the 1-D Riemann problem in a multi-D setting, a direc-. tion has to be specified along which the 1-D variations take place. (All gradients normal to this direction are neglected.) In standard upwind codes. in general. the interface-normal between two neighboring finite volumes is selected for this direction. Although this approach has shown to be very successful. it is not satisfactory from a physical point of view. Grid directions cannot determine upwinding directions in a physically proper manner. A consequence is that the interactions that characterize multi-D flows are bound to be misinterpreted. unless they are aligned with the grid. Features that are not grid-aligned are often poorly resolved. Therefore. main objective of this project was (and still is): to introduce multi-D physics for determining the upwinding directions and the corresponding propagation signals. By better mimicking physics. one may expect that the solution accuracy is improved. Of course. this should not be at the expense of computational efficiency and robustness. We proceed by briefly discussing a specific multi-D upwind approach.

Instead of concentrating on the time evolution of discontinuous solutions separated by finite-volume interfaces (as is done in the 1-D Riemann approach). one may

[1] An overview of this is given in a special journal issue: *CWI Quarterly*. Volume 6, Number 1 (CWI. Amsterdam. 1993).

consider the time evolution of the state in a complete cell. In the latter approach, the cell-vertex values are updated by the effect of linear waves evolving from the piecewise-linear initial data over the cells. By using a *continuous*. piecewise-linear initial-value distribution. the 1-D Riemann problem is completely avoided. Therefore. it is well-suited for multi-D extensions. Two distinct elements are needed for this approach:

- a decomposition in scalar contributions, of the flux residual (also called fluctuation) in each cell (each of these scalar fluctuations is associated with a well-defined propagation velocity), and
- a convection scheme which properly distributes these scalar fluctuations to the downstream cell vertices.

We start with a short discussion of the latter element. For this purpose, consider the 2-D, scalar convection equation:

$$\frac{\partial u}{\partial t} + \vec{\lambda} \cdot \nabla u = 0.$$

Suppose we have a mesh with quadrilateral cells, where each cell is subdivided into two triangular cells. Assuming a continuous, piecewise-linear representation of the solution over the vertices of the mesh (cell-vertex method), one can easily express the fluctuation in each triangular element, as a function of the solution values in the three corresponding vertices. A specific convection scheme is defined by the set of distribution weights, used to split the fluctuation over the downstream cell vertices. Herewith, different criteria can be imposed in order to achieve monotonicity. continuity and linearity preservation. It has been shown that only nonlinear schemes can ensure all of these desirable properties (precisely as with classical TVD-schemes). During the First Phase of our project, first-order as well as second-order accurate, and linear as well as nonlinear fluctuation-splitting schemes have been developed and applied[2].

We now proceed by discussing the first element of the specific multi-D upwind approach under consideration: the decomposition of the system fluctuation, which is a vector, in scalar contributions. Consider the 3-D system of conservation laws

$$\frac{\partial \vec{u}}{\partial t} + \nabla \cdot \vec{f}(\vec{u}) = 0,$$

where $\vec{u}$ represents the vector of unknowns (mass, momentum and energy) and $\vec{f}(\vec{u})$ the flux of these quantities. To apply the above fluctuation-splitting schemes to a hyperbolic system of conservation laws, a wave modeling step has to be added, which decomposes the fluctuation for the system in a number of scalar contributions (each of which can be handled by the fluctuation-splitting schemes discussed above). In 1-D, where waves can travel in one direction only, this decomposition is unique and forms the basis of the well-known flux-difference-splitting methods. However, in multi-D. waves can travel in infinitely many directions. hence allowing for many possible wave decomposition models. The design of optimal wave decompositions

[2] A second-order accurate, nonlinear, monotone scheme is applied in an industrial code of one of our endorsers.

(optimal in terms of both accuracy and efficiency), was and still is one the main tasks of our project.

Promising multi-D upwind results have been obtained already, not only through the above cell-vertex fluctuation-splitting approach [1], but also through the cell-centered finite-volume approach [4]. To solve the multi-D upwind discretized equations by a multigrid method, in the First Phase of the project, we also developed optimal multi-stage time-stepping schemes [1]. In Figure 1, we still give an example of the accuracy of the cell-vertex fluctuation-splitting approach, for a supersonic flow in a sine-shaped channel.

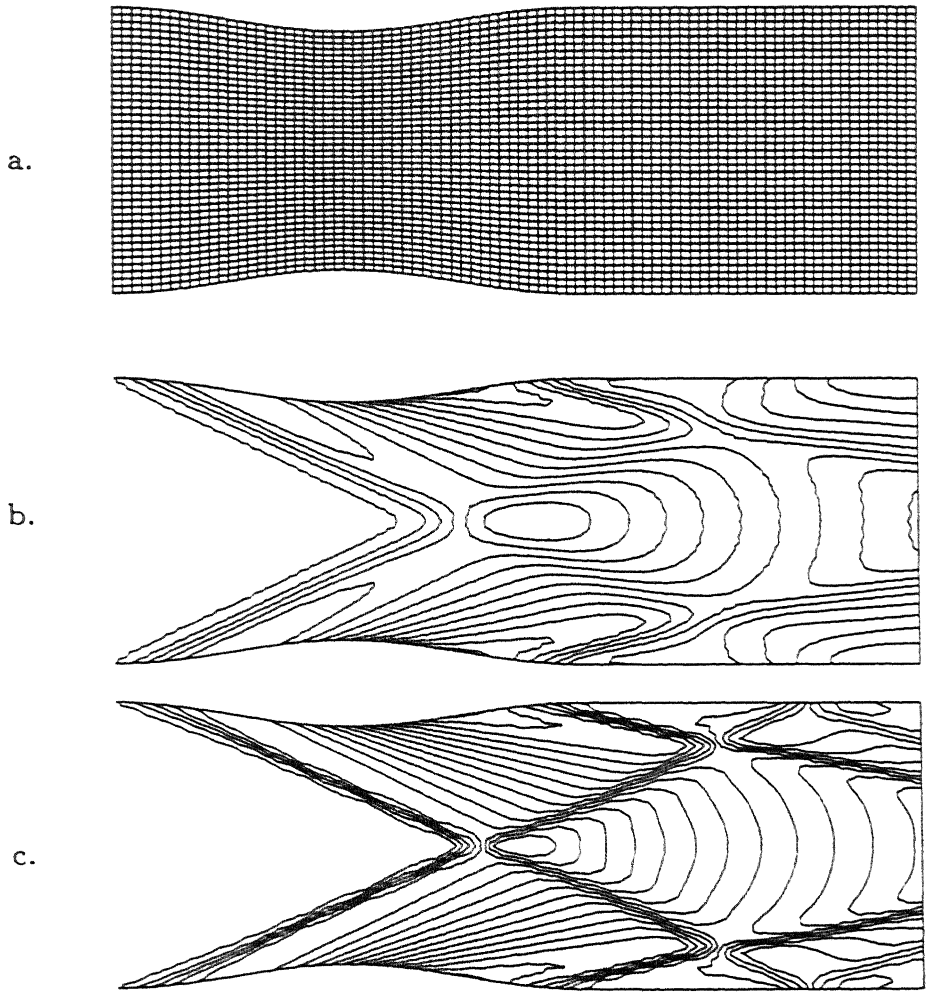

Figure 1: Sine-shaped channel flow ($M_{\text{inlet}} = 3.5$), computed by a cell-vertex fluctuation-splitting technique: **a.** 85×41-vertices grid. **b.** Mach-number distribution first-order accurate solution. **c.** Mach-number distribution second-order accurate solution.

Sparse-Grid Techniques

It is well-known how standard geometric multigrid methods can be applied to solve systems of algebraic equations, resulting from the discretization of 2-D partial differential equations. For 3-D partial differential equations, this is much less well-known. Cause is the greater difficulty in developing good 3-D smoothers. (A smoother is a numerical recipe to effectively reduce high-frequency components in the solution error during the iteration process.) An additional drawback of standard multigrid methods applied to 3-D problems is that they are hard to vectorize and parallelize. A remedy to all these problems of standard 3-D multigrid seems to be within reach now. For 2-D problems, it has appeared that it is not always necessary to create a finer grid by halving all cells in both coordinate directions (the standard way of refining). For some 2-D problems, it appears favorable to refine in one direction only: the *sparse-grid* way of refining [5]. Sparse-grid refinement can be applied to 3-D as well. In 3-D, with some care, it may completely remove aforementioned drawbacks of standard multigrid:

- For a 3-D sequence of sparse-grids, effective, and yet simple smoothers can be constructed quite easily.
- Thanks to this simplicity of smoothers, the vectorization properties may be better as well. (E.g., a *non-recursive* relaxation procedure as Jacobi's, can be applied as smoother.)
- The parallelization properties of sparse-grid methods are also better.
- Last but not least, to obtain the same solution accuracy as in a standard-grid computation, a sparse-grid computation needs less cells/points.

In fact, sparse-grid techniques can be interpreted as generalizations of standard geometric multigrid techniques. A crucial step in implementing sparse-grid techniques is the choice and construction of the data structure. We proceed by briefly describing the sparse-grid data structure, constructed just after the Second Phase's start. A detailed description is given in [2]. For a review of our standard (i.e. non-sparse) multigrid work, done during the project's First Phase, we refer to [3].

For the construction of discretization schemes for systems of partial differential equations, both physical and computational coordinates may be used. In the formal description of the data structure, only computational coordinates are used. We assume a Cartesian coordinate system and distinguish the x-, y- and z-coordinate directions. Further, we identify in this coordinate system an origin and a unit length. In a sparse-grid technique we need the simultaneous use of many different grids, cells, nodal points, and so on. However, there is only one *basic grid*, $\mathcal{R}_{0,0,0}$. This is the regular, rectangular grid consisting of all nodal points in $\Re^3$ that are located at points with integer coordinates in the computational space. Hence

$$\mathcal{R}_{0,0,0} = \{(i,j,k)\in\Re^3;\ i\in\mathbb{Z}, j\in\mathbb{Z}, k\in\mathbb{Z}\}\ . \tag{1}$$

Similarly, we introduce many (infinite) grids with nodal points at dyadic points in $\Re^3$. For any $(l,m,n)\in\mathbb{Z}^3$ we introduce a grid $\mathcal{R}_{l,m,n} \subset \Re^3$ as

$$\mathcal{R}_{l,m,n} = \{(i2^{-l}, j2^{-m}, k2^{-n})\in\Re^3:\ i\in\mathbb{Z}, j\in\mathbb{Z}, k\in\mathbb{Z}\}\ . \tag{2}$$

We call (l, m, n) the *level* of the grid. We also say that $\mathcal{R}_{l,m,n}$ is a grid *on the* $(l+m+n)$*-level*. The $\mathcal{R}_{l,m,n}$ in (2) are all possible grids. For an impression of the relation between these grids for $l+m+n \geq 0$, we refer to Figure 2. On these grids we may wish to handle all kinds of vertex- or box-centered methods, such as finite-element, mixed finite-element or finite-volume methods. I.e., we may wish to associate numerical values with any kind of cell, cell center, cell vertex, cell face or cell edge. Of course, in practice only finite parts, and a selection of all possibilities is used.

Let the discrete equations, that model the system of partial differential equations, be defined on a computational domain Ω. We assume that the computational domain Ω, an open set in $\mathbb{R}^3$, is not infinite, but that it consists of only a finite number of cells in the basic grid $\mathcal{R}_{0,0,0}$. Without loss of generality we may assume that the coordinates of all points in the closure $\overline{\Omega}$ of Ω are non-negative.

We pursue the construction of a data structure for solution-adaptive computations. This implies that we are interested in all the possible grids $\mathcal{R}_{l,m,n}$, with $l, m, n \geq 0$, as far as they cover the domain $\overline{\Omega}$. However, a priori we do not know what grids and what parts (what cell elements) of these grids will be needed in a computation. Therefore, we realize a data structure in which all cell elements that cover $\overline{\Omega}$ on the basic grid $\mathcal{R}_{0,0,0}$, will be represented. Further, all cells on the grid $\mathcal{R}_{l,m,n}$, $l, m, n \geq 0$, may exist in the data structure, provided that there exist also cells that cover the same space in the coarser grids $\mathcal{R}_{l-1,m,n}$, $\mathcal{R}_{l,m-1,n}$ and $\mathcal{R}_{l,m,n-1}$. Note that, for each of these three grids, these *mother cells* are uniquely determined. However, if any of the indices $l-1$, $m-1$ or $n-1$, is negative, we do *not* require corresponding mother cells to exist in the coarser grid. We notice that in all aspects the data structure is (and remains) symmetric with respect to the three coordinate directions.

The data structure has been implemented in FORTRAN. At present, pilot computations for a 3-D Poisson equation are being carried out. An extension to the steady, 3-D, compressible Navier-Stokes equations will be made later in the Second Phase.

Concluding Remarks

In our opinion, a major requirement to be fulfilled by tomorrow's numerical methods for computational aerodynamics is increased robustness, particularly in complex computations which are beyond the capabilities of users 'baby-sitting' computations with tuning parameters. It is mainly for their robustness that during the past decade upwind discretization methods have gained such a popularity. However, many flow problems exist for which today's upwind methods are not yet satisfactory. Essentially, these are problems characterized by genuinely multi-D flow features such as turbulence, separation, and so on. The upwind methods considered in this project try to respect as many multi-D flow features as possible, by extracting a maximum amount of physical information from a minimum amount of numerical data. The second important goal in developing tomorrow's numerical methods is the further improvement of computational efficiency per grid point. For this purpose, we think that solution-adaptive sparse-grid methods are best suited.

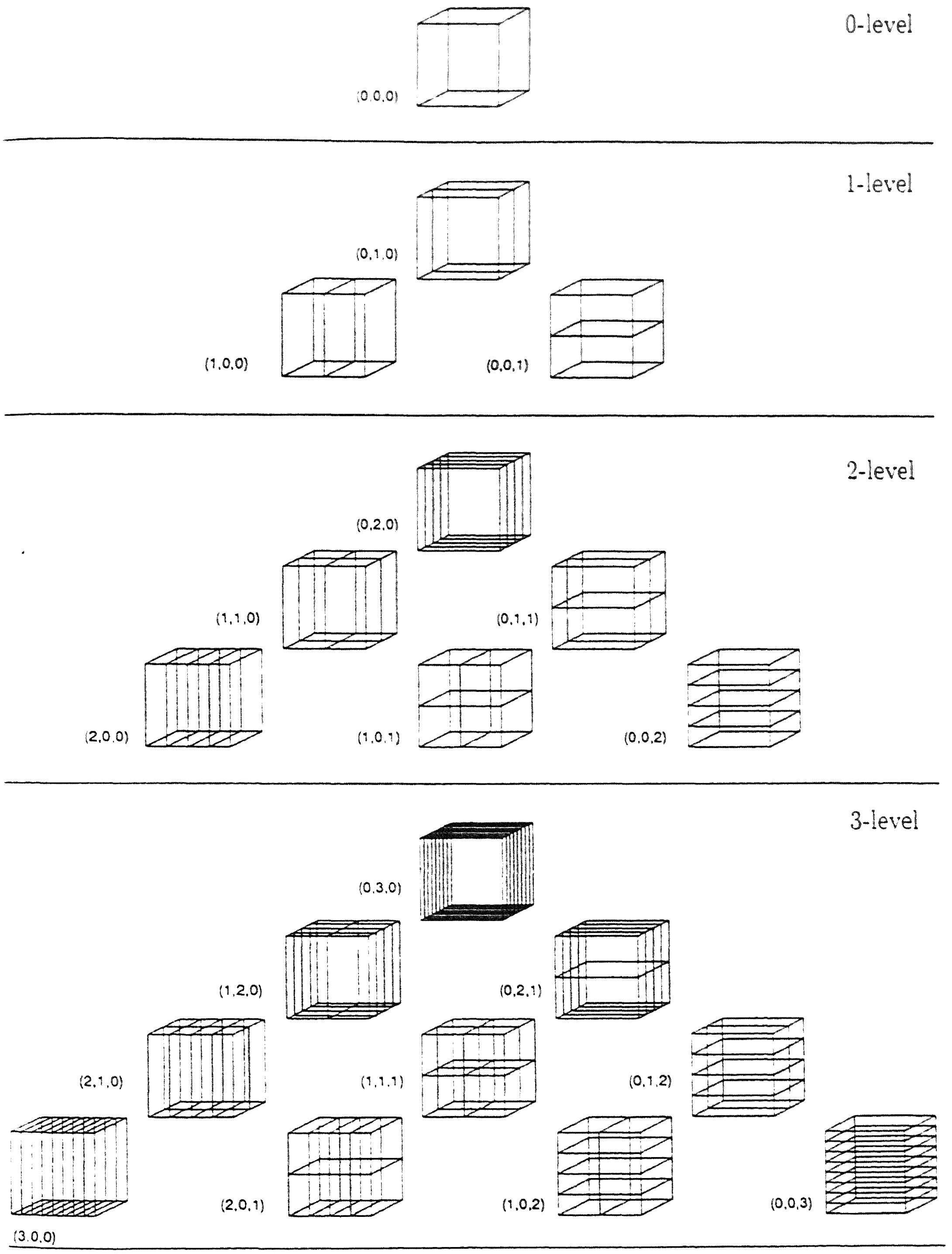

Figure 2: An impression of the grids $\mathcal{R}_{l,m,n}$ with $l,m,n \geq 0$. In this figure a cell on the basic grid $\mathcal{R}_{0,0,0}$ is shown, together with its refinements on the grids $\mathcal{R}_{l,m,n}$, $l+m+n=\ell$, on all the ℓ-levels, $\ell = 1,2,3$.

References

1. H. Deconinck. R. Struijs. H. Paillère, L.A. Catalano, P. De Palma, M. Napolitano and G. Pascazio. 'Development of Cell-Vertex Multi-Dimensional Upwind Solvers for the Compressible Flow Equations', *CWI Quarterly*, **6**, 1-28 (1993).

2. P.W. Hemker and P.M. de Zeeuw, 'BASIS3, a Data Structure for 3-Dimensional Sparse Grids', *CWI-Report* (to appear).

3. H.T.M. van der Maarel, P.W. Hemker and B. Koren. 'Application of a Solution-Adaptive Multigrid Method to the Euler Equations', *CWI Quarterly*, **6**, 49-75 (1993).

4. P. van Ransbeeck and Ch. Hirsch., 'Cell-Centered Genuinely Multidimensional Upwind Algorithms and Structured Meshes', *CWI Quarterly*, **6**, 29-48 (1993).

5. C. Zenger., 'Sparse grids', Technische Universität München, Institut für Informatik, TUM-I9037 (1990).