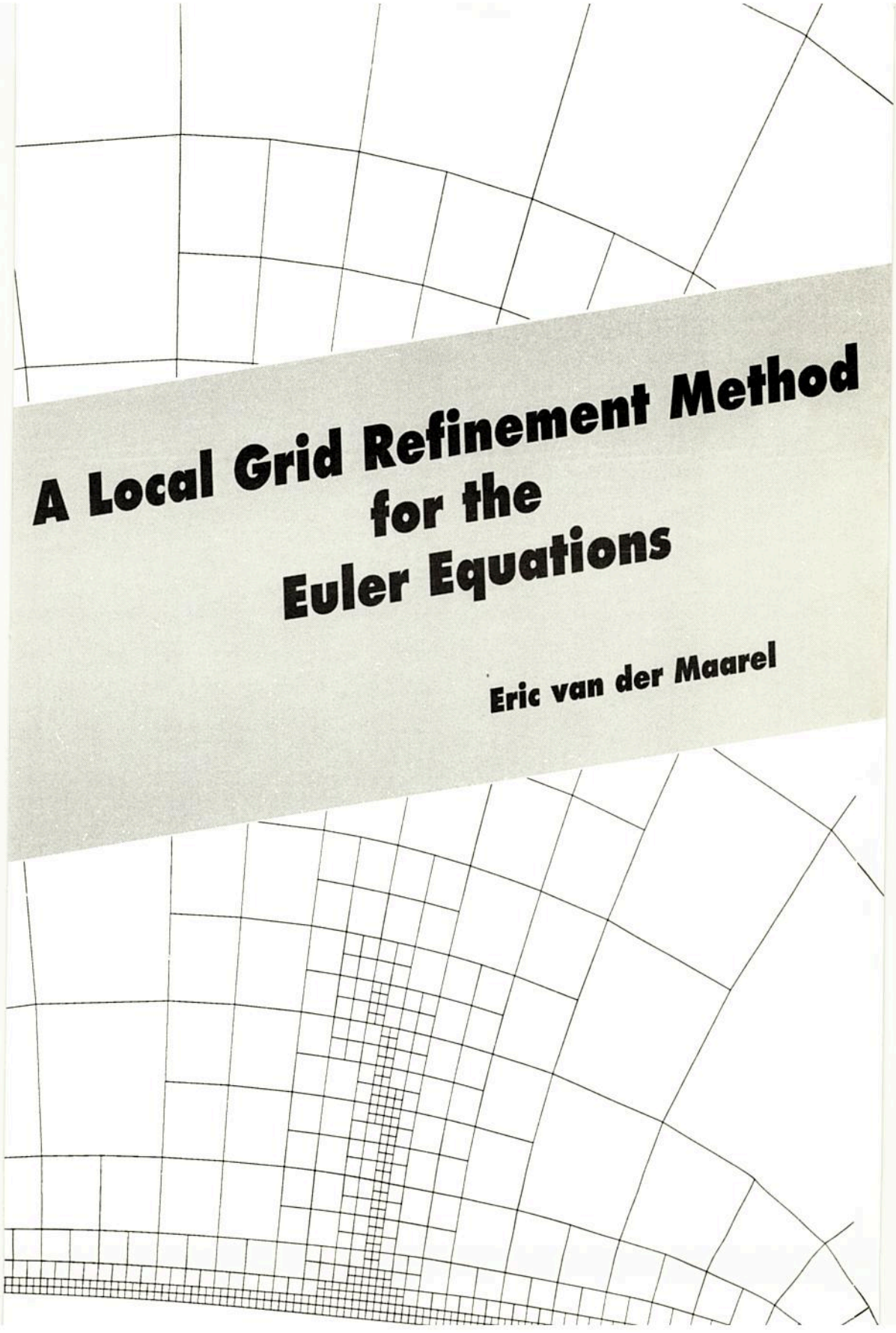


A Local Grid Refinement Method for the Euler Equations

Eric van der Maarel

A Local Grid Refinement Method for the Euler Equations

A Local Grid Refinement Method for the Euler Equations

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam,
op gezag van de Rector Magnificus,
prof.dr. P.W.M. de Meijer
in het openbaar te verdedigen in de Aula der Universiteit
(Oude Lutherse Kerk, ingang Singel 411, hoek Spui),
op dinsdag 23 februari 1993 te 15.00 uur

door

Hendricus Theodorus Maria van der Maarel

geboren te Amsterdam

CWI
Amsterdam
1993

Een Lokale Roosterverfijningsmethode voor de Euler Vergelijkingen

Promotor: Prof.dr. P.W. Hemker
Faculteit: Wiskunde en Informatica

This work has been supported by the European Communities, through a BRITE/EURAM Area 5 project under contract no. Aero-0003/1094.

To the memory of
my father and
my father-in-law

Acknowledgments

This thesis is the result of the effort of more than one person alone, but rather of a number of people, all of whom supported me in writing up our common achievements.

With pleasure I would like to express gratitudes and admiration to my supervisor Prof.Dr. P.W. Hemker. He is a wise and patient teacher, who provided me with all freedom I could ever have wished for.

Special thanks are due to Dr. B. Koren. Always ready to discuss details or provide references from his amazing on-line database (sheer memory), he has been very helpful in doing research and finishing this thesis.

Another very helpful person has been Mr. C.T.H. Everaars, to whom I owe my sincere appreciations for his programming skills and his flexibility during our cooperation.

Furthermore, I would like to express my gratitude to Prof.Dr.Ir. P. Wesseling, Drs. P.M. de Zeeuw, Dr. J. Molenaar and Dr. R.R.P. van Nooyen for their patience and comments during our 'NW2-werkbespreking' sessions.

CWI has provided me with a very pleasant environment to do research. Anybody who contributed to this, in whatever amount, is greatly acknowledged.

With the greatest pleasure I express my warmest appreciations to my wife, Jeanne van den Bosch, for her patience and confidence all those years and for creating the proper base to start from and to return to, over and over again.

Contents

CHAPTER 1

Introduction

1.1. General introduction

Tools to approximately predict the behaviour of a fluid in various situations, is of great interest to many technical disciplines. For example there is an interest from petro-chemical industry for the prediction of the recovery and for transport systems of oil and gas; in geophysical science for the prediction of the flow of water and pollutants in seas and rivers and for making weather-forecasts; in glass processing industry, for the prediction of the flow of liquid glass; in aerospace industry, for the prediction of loads on the structure of an airplane and the dynamic behaviour of an airplane under various conditions. Often, numerical predictions or simulations are used in an iterative process such as the design of a technical device or system. The number of iteration cycles in a design process is usually large. Hence, efficient computational tools are of major importance to these design processes.

In each of the disciplines mentioned above, a model of reality is considered that puts emphasis on the physical phenomena important to that discipline or to a particular problem under consideration. In this thesis we consider the Euler equations of fluid dynamics, for a perfect gas. Our main interest is the numerical approximation of solutions of the steady, two-dimensional Euler equations, by a solution-adaptive method, which uses local grid refinement (enrichment) and a multigrid method to solve the system of discrete equations. The Euler equations model the flow of a compressible, inviscid fluid, without taking into account the conduction of heat. Aerospace aerodynamics is an area of application interested in a physical model that neglects viscosity and heat conduction effects in the flow of a perfect gas. For an object moving through a gas, viscosity and heat conduction tend to play an important role only in the close vicinity of that object. Hence, in the 'outer' region (away from the object), a realistic description can often be made with the Euler equations. For an important part of the duration of a flight, an airplane is in a so-called quasi-steady situation, where the flight conditions change very slowly. Furthermore, air can be considered as an ideal gas with constant specific heats over a very wide range of conditions. These considerations motivate the study of the steady Euler equations. We restrict ourselves to

steady flows in two space dimensions.

1.2. The Euler equations of fluid dynamics

1.2.1. Euler equations. The Euler equations may be derived from the Boltzmann equation. The Boltzmann equation describes the probability density of a system of N gas particles, having position $x \in \mathbb{R}^d$ and velocity $v \in \mathbb{R}^d$, at time t, [22]. In the limit as $N \to \infty$, the gas behaves like a fluid. The Euler equations are valid for this limit situation (continuum hypothesis). Writing the solution of the Boltzmann equation in a formal power series of the mean free path $\epsilon \propto N^{-1}$ and applying his theory of integral equations, Hilbert found that for the solvability of the equation for the lowest-order term (i.e. existence of the solution of the equation for the lowest-order term of this expansion), a set of nonlinear conservation equations must be satisfied [11]. These equations are known as the Euler equations. Similarly, the Navier-Stokes equations are obtained, if the solution of the Boltzmann equation is written as the so-called Chapman-Enskog expansion [4]. This expansion adds higher-order correction terms to the Hilbert expansion. The compressible Euler equations can also be derived from the compressible Navier-Stokes equations, by assuming an inviscid and non-heat conducting fluid.

1.2.2. Conservation laws. Usually the Euler (and Navier-Stokes) equations are derived from the continuum hypothesis, Newton's second law of motion and the first law of thermodynamics. These principles can be considered as physical laws that describe the conservation of mass, momentum and energy for an arbitrary domain, contained in the domain of definition of the problem.

A *conservation law* states that the rate of change (in time) of the *amount* of a substance contained in a domain $\Omega^* \subset \Omega$ is equal to the sum of the *flux* of that substance across the boundary $\partial\Omega^*$ of Ω^*, and the amount of substance per unit volume produced per unit time in Ω^*. For the *density* of the substance (i.e., substance per unit volume), denoted by q, a flux $f(q)$ and source s per unit volume, a conservation law is given by

$$\frac{d}{dt}\int_{\Omega^*} q(x)\,d\Omega = -\oint_{\partial\Omega^*} f(q)\,n\,ds + \int_{\Omega^*} s(x)\,d\Omega.$$

Here, n denotes the outward unit normal at $\partial\Omega^*$ and x and t denote space and time coordinates respectively. In this thesis we are concerned with steady (i.e. vanishing dq/dt) conservation laws in two space dimensions, with fluxes $f(q)$ and $g(q)$. These conservation laws can be written as

$$\oint_{\partial\Omega^*} f n_x + g n_y\,ds = \int_{\Omega^*} s\,d\Omega, \quad \forall\Omega^* \subset \Omega, \tag{1.2.1}$$

and $\Omega \subset \mathbb{R}^2$. Here, the unit normal is $n = (n_x, n_y)^T$ with n_x and n_y components in the direction of the x and y axis respectively. The fluxes f and

g are d-dimensional vector functions. The components of fn_x and gn_y form the flux in the direction of the normal n.

Usually, conservation laws are given in differential form. This gives

$$\frac{\partial f(q)}{\partial x} + \frac{\partial g(q)}{\partial y} = s(x,y), \quad \forall (x,y) \in \Omega. \tag{1.2.2}$$

1.2.3. The steady Euler equations. The steady Euler equations defined in an open domain $\Omega \subset \mathbb{R}^2$, can be written in differential form as given by (1.2.2), with q, f and g defined by

$$q = \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix}, \qquad f(q) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{pmatrix}, \qquad g(q) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vH \end{pmatrix}. \tag{1.2.3}$$

For the Euler equations the source term is $s = 0$. These equations denote respectively the conservation of mass, of momentum in the direction of the x and y axis respectively, and of energy. Here, ρ denotes the density of the gas, u and v are the velocity in the direction of the x and y axis respectively, p the pressure and $H = E + p/\rho$ the specific total enthalpy, with $E = e + \frac{1}{2}(u^2 + v^2)$ the specific total internal energy and e the specific internal energy. For a perfect gas the internal energy is given by

$$e = \frac{1}{\gamma - 1}\frac{p}{\rho}, \tag{1.2.4}$$

where γ is the constant ratio of specific heats, $\gamma = c_p/c_v$, at constant pressure and volume respectively.

For a set of conservation laws such as the Euler equations, we can often write

$$f(q)n_x + g(q)n_y = T^{-1} f(Tq), \tag{1.2.5}$$

where T is a rotation matrix. Equation (1.2.5) signifies the rotational invariance of the conservation laws. However, often we will not use this notation, since many of our results are applicable to a wider range of problems than those which have the rotational invariance property. Usually we write for the flux in the direction of some unit vector n

$$f(q)n_x + g(q)n_y = f^*(q,n).$$

1.3. The numerical tools

In order to provide an efficient numerical tool for the computation of Euler flows, in this thesis an adaptive technique is studied. The adaptation lies in the construction of a solution-dependent, locally refined grid. For a numerical method, this is by no means the only possible way to adapt a technique to the evolving solution. Without being exhaustive, we consider a number

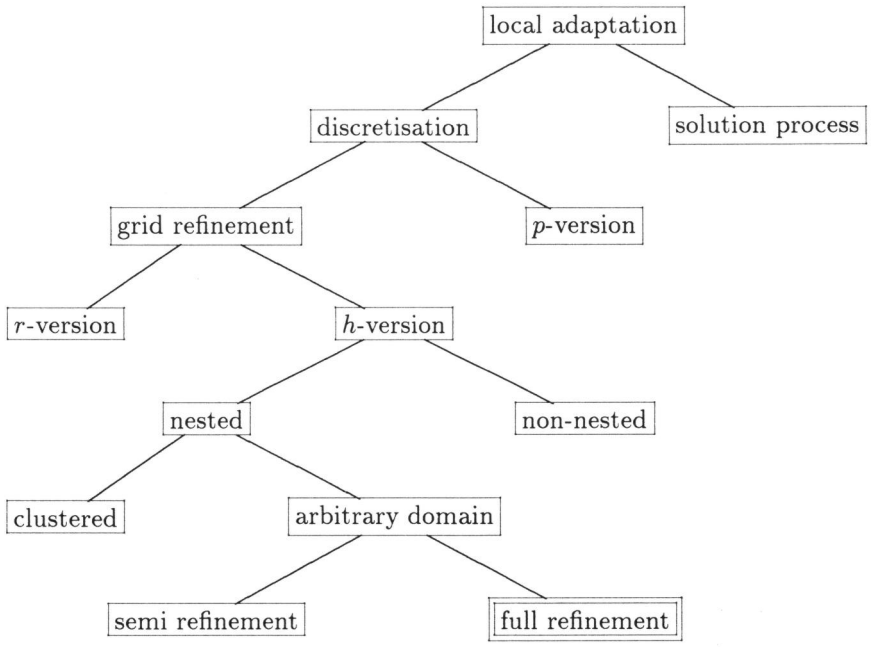


FIGURE 1.3.1. *Classification of adaptive techniques in numerical mathematics.*

adaptation techniques and we place the adaptive, local grid refinement method considered in the present thesis, in a wider perspective.

The system of equations obtained by discretisation of the continuous problem, is solved by a multigrid technique. Some considerations concerning multigrid and local grid refinement are presented in this section.

1.3.1. Classification of adaptive numerical techniques. If we study the literature on adaptive techniques in numerical mathematics, a classification as shown in Fig. 1.3.1 can be made. This classification is not complete, but it may place the solution-adaptive, local grid refinement method considered in this thesis, in a proper perspective. Each element of a subdivision in Fig. 1.3.1 may be further subdivided in order to distinguish between different methods. We only consider the subdivision which leads to the method considered in the present thesis. Note that the subdivisions made in this classification, are not mutually exclusive; a combination of techniques is often possible. Further, we are aware that we present only one of several possible classifications.

Adaptation in numerical computations may involve both the discretisation and the process of solving the system of discrete equations. Adaptation of

the solution process is very common and is used almost everywhere. For example one can think of methods which control local CFL number in time stepping schemes, or, in multigrid context, the adaptation of smoothing and multigrid cycling to the local convergence rate. Also the choice of a stopping criterion in an iterative process embedded in yet another iterative process, may be regarded as adaptation of the solution process, since it may control the number of iterations performed.

Adaptation of the discretisation has been extensively considered, e.g. in [5], and [18] gives a review of application of these techniques. In local adaptation of the discretisation, two basic and distinctly different techniques exist. First, there are discretisations which (locally) use higher order accurate approximations of the continuous equations. Methods based on this technique are said to use the p-version of adaptation. Second, there are discretisations which use (local) refinement of the grid. The common idea behind the two adaptation techniques is controlling the error of the discretisation through the reduction or possibly equidistribution of the local discretisation error.

The use of grid adaptation techniques may be further subdivided into methods which obtain refinement of the grid by *moving* grid points across the domain of definition, and methods which *add* points to the grid: grid enrichment. Moving-grid techniques usually keep the number of unknowns constant when adapting, but possibly with a high penalty on grid skewness. Methods using this technique are often called r-version methods. Grid enrichment techniques ar often referred to as applications of the h-version of refinement. Methods which use a combinations of these techniques exist: the so-called h-p-version and h-r-version (cf. [5]). The adaptation of the solution process may be combined with any of the h, p, h-p or h-r-version.

Local grid refinements by enrichment may be constructed in a nested or non-nested fashion. In non-multigrid context, non-nested refinement amounts to a reshifting of grid points after refinements have been introduced (h-r-version), or to the use of non-aligned, partially overlapping grids. Non-aligned grids complicate the discretisation of the equations drastically. Nevertheless, in multigrid context, a sequence of locally non-nested grids is considered by some authors (cf. [15]). These non-nested grids make grid transfers quite complicated .

Nested local grid refinement can extend over the domain or refinements can be clustered. This is done in such a way that locally refined subdomains satisfy a-priori posed requirements, such as e.g., a rectangular shape of a subdomain.

Finally, enrichment may be obtained by subdivision of a grid-cell in a full set of fine-grid cells, (i.e., subdivision in more coordinate directions) or by subdivision only in one prevailing direction. The latter is called semi-refinement.

In this thesis we consider a local grid enrichment method, which uses a sequence of locally nested grids consisting of non-clustered collections of cells in a generally irregularly shaped subdomain. Refinements are obtained by full

refinement of grid cells (refinement in two directions).

1.3.2. Local refinement and multigrid. The refinement procedure considered in this thesis results in a composite grid with locally refined regions. Adaptive, local grid refinement is combined with a multigrid solution technique. Examples of these approaches have been given through MLAT [3], PLTMG [2] and FAC [16], for vertex-centered discretisations of (mainly) elliptic problems. A theoretical justification for using multigrid on locally refined grids is given in [23]. There, a local grid refinement technique for elliptic problems in a finite element setting and with application of multigrid, is shown to be of optimal complexity. This is also found for the finite difference discretisation of a model problem and experimentally verified in [1].

The grid is formed by a regular partition of the computational domain into quadrilateral cells. These cells are refined by subdividing each cell into a set of 2×2 smaller cells, which form the grid on the next finer level. Generation of such local refinements results in a locally nested sequence of grids, in which the finer grids possibly cover only subdomains of the domain covered by the coarser grids.

On this sequence of grids, the conservation laws are discretised by an upwind finite volume method. The set of equations which result from the first-order accurate discretisation, is solved by nonlinear multigrid iteration. The equations from the second-order accurate discretisation are solved by defect correction, where the first-order discretisation is used as the less accurate operator and the multigrid method is employed to invert this operator. A computation is started on some basic grid. After a number of nonlinear multigrid iterations (FAS-cycles) a new level of refinements is generated. The initial approximation in all newly generated cells is found by interpolation from the next coarser grid. Then again the multigrid iteration is started with the new refinements as the top level. After one or more cycles, again new cells may be generated, both on the already existing levels and on an even finer level. Cells that have no refinements and have become superfluous with respect to the refinement criteria, may be discarded. These steps may be repeated until a satisfactory solution has been obtained. The basic principle of such an adaptive technique was already described in [7].

Local refinements are introduced in the grid, mainly for two reasons. First, the grid may be refined to provide the required level of *resolution*. Second, local refinements may be introduced to obtain a certain level of *accuracy* of the system of discrete equations. The resolution requirement of the grid is directly dictated by gradients of the true solution. Hence, gradients or undivided differences of computed solution components may serve as refinement criteria. The accuracy of the numerical approximation is controlled by the error in each discrete equation (i.e., the local discretisation error). In general, the influence of the local discretisation error on the accuracy of the solution (or the global discretisation error) is not simple, and unknown. However, usually when the

local discretisation error vanishes, then so does the error in the solution.

The multigrid and defect correction algorithms used in this thesis, are a extension of the work of Hemker and Spekreijse [9], [10], Hemker [8], Spekreijse [19], [20], [21], Koren and Spekreijse [14] and Koren [12], [13]. The main difference between the work mentioned above and the present extension is concerned with the discrete equations obtained from the discretisation in the neighbourhood of the coarse and fine grid interfaces. Another difference is that now a coarse-grid solution has an influence on the solution on the final, locally refined grid (i.e., composite grid), whereas in the non-adaptive case a coarse grid is only used to accelerate convergence. In the present method, the coarse grid 'holds' a part of the solution. This aspect is also reflected in the right-hand side of the equations defined for the grid on some level. For the non-refined cells the discrete equations have a right-hand side which is the source term of the problem. For the refined cells, the right-hand sides of the discrete equations approximate the local discretisation error.

1.4. Outline of the thesis

Chapter 2 of this thesis introduces the geometric structure, and the discretisation of conservation laws, with in mind the steady Euler equations. The geometric structure enables the definition of a system of discrete equations, which approximates the continuous problem. In this chapter the local discretisation error is studied and requirements are given which should be satisfied by the discretisation, to obtain a consistent discretisation in a specific sense, of first or second order. Emphasis is on the discretisation in the neighbourhood of interfaces between a coarse and a fine grid.

Chapter 3 presents results of numerical studies, obtained with the adaptive multigrid method described in Chap. 2. The problems have been selected to present accuracy, efficiency and flexibility of the method (i.e., its ability to be effectively used in non-standard situations).

Chapter 4 is concerned with the a-posteriori estimation of the local discretisation error, in order to use it in a local grid refinement criterion. For a one-dimensional model problem, the local and global discretisation errors are studied. Next, a method is introduced and studied which enables the a-posteriori estimation of the local discretisation error. Finally, this method is extended to the two-dimensional situation and used in a criterion for the introducton of local grid refinements, for an actual nonlinear model problem.

References

[1] D. Bai and A. Brandt, *Local mesh refinement multilevel techniques*, SIAM J. Sci. Statist. Comput. **8** (1987), no. 2, 109–134.

[2] R.E. Bank, *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations, Users' Guide 6.0*, Frontiers in Applied Mathematics, vol. 7, Society for Industrial and Applied Mathematics, Philadelphia, 1990.

[3] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp. **31** (1977), no. 138, 333–390.

[4] R.E. Caflisch, *Fluid dynamics and the Boltzmann equation*, Nonequilibrium Phenomena I, The Boltzmann equation (J.L. Lebowitz and E.W. Montroll, eds.), North-Holland, 1983, pp. 193–223.

[5] J.E. Flaherty, P.J. Paslow, M.S. Shephard and J.D. Vasilakis (eds.), *Adaptive methods for partial differential equations*, Troy, 1988, Rensselaer Polytechnique Institute, Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[6] W. Hackbusch and U. Trottenberg (eds.), *Multigrid Methods* II, Proc. of the 2nd European Conference on Multigrid Methods, held in Cologne, 1985, Lecture Notes in Mathematics, vol. 1228, Springer-Verlag, 1986.

[7] P.W. Hemker, *On the stucture of an adaptive multi-level algorithm*, BIT **20** (1980), 289–301.

[8] ———, *Defect correction and higher order schemes for the multi grid solution of the steady Euler equations*, In Hackbusch and Trottenberg [6], pp. 149–165.

[9] P.W. Hemker and S.P. Spekreijse, *Multigrid solutions of the steady Euler equations*, Advances in Multi-Grid Methods, Proc. conference held in Oberwolfach, Notes on Numerical Fluid Mechanics, vol. 11 (Oberwolfach, 1984) (D. Braess, W. Hackbusch and U. Trottenberg, eds.), Vieweg Braunschweig, 1985, pp. 33–44.

[10] ———, *Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations*, Appl. Num. Math. **2** (1986), 475–493.

[11] D. Hilbert, *Grundzüge einer Allgemeinen Theorie der Linearen Integralgleichungen*, Chelsea Publishing Company, New York., 1953.

[12] B. Koren, *Defect correction and multigrid for the efficient and accurate computation of airfoil flows*, J. Comput. Phys. **77** (1988), 183–206.

[13] ———, *Multigrid and Defect Correction for the Steady Navier-Stokes Equations, Application to Aerodynamics*, CWI, Amsterdam, 1990, CWI-tract 74.

[14] B. Koren and S.P. Spekreijse, *Solution of the steady Euler equations by a multigrid method*, In McCormick [17], pp. 323–336.

[15] D. Mavriplis and A. Jameson, *Multigrid solution of the Euler equations on unstructured and adaptive meshes*, In McCormick [17], pp. 413–429.

[16] S. McCormick, *The Fast Adaptive Composite grid (FAC) method for elliptic equations*, Math. Comp. **46** (1986), no. 174, 439–456.

[17] S.F. McCormick (ed.), *Multigrid Methods, Theory, Applications and Supercomputing, Proc. of the Third Copper Mountain Conference on Multigrid Methods, Copper Mountain, 1987, Lecture Notes in Pure and Applied Mathematics, vol. 110*, Marcel Dekker Inc., New York, 1988.

[18] J.T. Oden, *Progress in adaptive methods in Computational Fluid Dynamics*, In Flaherty et al. [5], pp. 206–252.

[19] S.P. Spekreijse, *Multigrid solution of monotone second-order discretizations of hyperbolic conservation laws*, Math. Comp. **49** (1986), no. 179, 135–155.

[20] ———, *Second order accurate upwind solutions of the 2D steady Euler equations by the use of a defect correction method*, In Hackbusch and Trottenberg [6], pp. 285–300.

[21] ———, *Multigrid Solution of the Steady Euler Equations*, CWI, Amsterdam, 1988, CWI-tract 46.

[22] S. Ukai, *Solutions of the Boltzmann equation*, Patterns and Waves, Qualitative Analysis of Nonlinear Differential Equations (T. Nishida, M. Mimura and H. Fujii, eds.), Studies in Mathematics and its Applications, vol. 18, North-Holland, 1986, pp. 37–96.

[23] J.R. Van Rosendale, *Rapid solution of finite element equations on locally refined grids by multi-level methods*, Ph.D. thesis, University of Illinois at Urbana-Champaign, May 1980.

CHAPTER 2

Discretisation of a system of conservation laws on a locally refined grid

2.1. Introduction

In this chapter we introduce a discretisation of a system of steady conservation laws, in two space dimensions, using a locally refined partitioning of the domain of definition, and we study discretisation errors.

We consider a system of d conservation laws, defined on an open domain $\Omega \subset \mathbb{R}^2$, and with $q : \overline{\Omega} \to \mathbb{R}^d$, $s : \Omega \to \mathbb{R}^d$, $f, g : \mathbb{R}^d \to \mathbb{R}^d$ given by

$$\frac{\partial f(q(x,y))}{\partial x} + \frac{\partial g(q(x,y))}{\partial y} = s(x,y),\ (x,y) \in \Omega. \tag{2.1.1a}$$

Boundary conditions of the problem are given by

$$B(q(x,y),x,y) = 0,\ (x,y) \in \partial\Omega. \tag{2.1.1b}$$

Here, $\partial\Omega \subset \overline{\Omega}$ denotes the boundary of the domain Ω.

For the discretisation we introduce a partitioning of the domain Ω. The partitioning forms, possibly after a smooth coordinate transformation, a set of regular quadrilaterals, called the grid. In transformed coordinates, the locally refined grid is composed of a sequence of locally nested grids, where each grid is a regular partitioning of a subdomain of the domain of definition. In non-transformed coordinates, the union of all quadrilaterals forms an approximation of the domain of definition.

Each quadrilateral of the grid is used as a control volume on which the system of conservation laws integrated over the control volume, is approximately satisfied. This leads to a discretisation of a weak formulation of problem (2.1.1).

The error in the approximation of the weak formulation consists of contributions from the various steps in the discretisation. The first step in the discretisation is the approximation of the domain by the partitioning. This has consequences for the accuracy of the discrete equations defined for quadrilaterals along the boundary of the domain and for the 'coarse-grid' approximation. The approximation of the weak formulation for each quadrilateral involves the

approximation of the mean flux per unit time and 'area' across each side of each quadrilateral. The next step is the approximation of the mean flux across the sides of the quadrilaterals by the flux at the mean state across each side. Finally the mean state itself is approximated from the discrete data (i.e., the available numerical solution, a cell-wise constant function). With this approximated mean state the flux is evaluated. This is done in an upwind fashion, where each flux depends in an upwind biased sense on the discrete data.

The numerical solution itself approximates the mean of the exact solution of (2.1.1) on each quadrilateral. Hence, each quadrilateral is associated with an approximation of the mean value on the quadrilateral of the solution of the continuous problem. This is the so-called *cell-centered* approach.

In this chapter, after formally introducing the geometric structure and notations, we discuss the discretisation and we study the various contributions to the local discretisation error. Emphasis is put on the discretisation involving the interfaces between a coarse part of the partitioning and fine part. Our analysis leads to a small set of requirements, to be satisfied in order to attain a discretisation which is first-order or second-order accurate (in a sense that will be specified) with respect to the mesh size of the partitioning. Numerical results of two test problems illustrate the discrete convergence of the proposed discretisation.

2.2. The geometric structure

2.2.1. Relations in the structure. We consider a system of conservation laws, defined on a bounded, open domain $\Omega \subset \mathbb{R}$, with boundary conditions on $\partial\Omega \subset \overline{\Omega}$, the boundary of Ω, where $\overline{\Omega}$ denotes the closure of Ω. We assume that a rectangular domain $\widetilde{\Omega} \subset \mathbb{R}^2$ exists and a sufficiently smooth surjective mapping $M : \overline{\widetilde{\Omega}} \to \overline{\Omega}$, which is also injective in the interior $\widetilde{\Omega}$. The mapping is a transformation of the Cartesian coordinates in $\overline{\widetilde{\Omega}}$, the *computational* space, into Cartesian coordinates in $\overline{\Omega}$, the *physical* space.

The system of conservation laws is discretised by a finite volume method. For this purpose, a regular rectangular partitioning of $\widetilde{\Omega}$ is introduced, consisting of disjoint rectangles. This set of rectangles also defines a partitioning of Ω, through the mapping M. A rectangle in the computational space $\widetilde{\Omega}$, as well as its image in Ω, is called a *cell*. The partitioning is called the *grid*. A grid is a collection of cells in the computational or the physical space.

We consider grids on different *levels of refinement*. A level of refinement l, with $l \in \mathbb{Z}$, is a regular partitioning, denoted by $\widetilde{\Omega}^l$, of a subdomain of $\widetilde{\Omega}$, and a surjective mapping $M^l : \overline{\widetilde{\Omega}} \to \overline{\widetilde{\Omega}^l}$, injective in the interior $\widetilde{\Omega}^l$. We use the notation

$$\Omega^l = M^l(\widetilde{\Omega}^l).$$

where $M^l(\widetilde{\Omega}^*)$ denotes the image of $\widetilde{\Omega}^*$ under the mapping M^l. For the image

$\widehat{\Omega}^l$ of $\widetilde{\Omega}$ under M^l, and the hull $\widehat{\Omega}$ of all images of $\widetilde{\Omega}$ under the sequence of mappings $\{M^l\}_{l\in\mathbb{Z}}$, we use the notations

$$\widehat{\Omega}^l = M^l(\widetilde{\Omega}), \tag{2.2.1}$$

$$\widehat{\Omega} = \bigcup_l \widehat{\Omega}^l. \tag{2.2.2}$$

A mapping M^l is associated with level of refinement l and is an approximation of M. Generally we take M^l such that it maps a cell vertex from the partitioning $\widetilde{\Omega^l}$ to the same point in $\mathbb{R}^2$ as the original mapping M does. Then in the sequence of mappings $\{M^l\}_{l\in\mathbb{Z}}$, M^l approaches M as $l \to \infty$. Depending on M^l, the grid Ω^l approximates $M(\widetilde{\Omega}^l)$. Hence, if $\overline{\widetilde{\Omega}^l} = \overline{\widetilde{\Omega}}$, then $\overline{\Omega^l}$ is an approximation of $\overline{\Omega}$.

Since the partitioning of Ω on a level l is regular quadrilateral, a cell on level l can be denoted by $\Omega^l_{i,j} \subset \widehat{\Omega}$. The set I of indices is defined as

$$I = \left\{(i,j,l) \in \mathbb{Z}^3 \mid \exists \Omega^l_{i,j} \subset \widehat{\Omega}\right\}.$$

The grid on level l is

$$\Omega^l = \left\{\Omega^l_{i,j} \mid (i,j,l) \in I\right\}.$$

Each cell on Ω^l has either *none* or only *one* neighbouring cell at each side, residing on the same level. A cell $\Omega^l_{i,j}$ is the northern neighbour of $\Omega^l_{i,j-1}$ and the eastern neighbour of $\Omega^l_{i-1,j}$, provided $(i,j-1,l),(i-1,j,l) \in I$.

The boundary $\partial\Omega^l_{i,j} \subset \overline{\Omega^l_{i,j}}$ consists of the four *cell faces* of the cell, identified through their relative locations by $\partial\Omega^l_{i,j,k}$, $k \in D$, and

$$\partial\Omega^l_{i,j} = \bigcup_{k\in D} \partial\Omega^l_{i,j,k},$$

where $D = \{N, E, S, W\}$ is the set of wind directions. We also have

$$\begin{aligned} \partial\Omega^l_{i,j,E} &= \partial\Omega^l_{i+1,j,W}, \quad \forall (i,j,l),(i+1,j,l) \in I, \\ \partial\Omega^l_{i,j,N} &= \partial\Omega^l_{i,j+1,S}, \quad \forall (i,j,l),(i,j+1,l) \in I. \end{aligned}$$

Refinements of a cell $\Omega^l_{i,j}$ are the cells obtained by subdivision of the corresponding cell $\widetilde{\Omega}^l_{i,j}$ in the computational domain into 2×2 smaller cells of equal size. By applying the mapping M^{l+1} to these refinements in the computational domain, we construct the refinements in the partitioning of the physical domain. Except for cells on the coarsest grid, each cell is one of the four *descendants* of a cell on the coarser grid. The coarse-grid cells on Ω^l and the fine-grid cells on Ω^{l+1} are coexistent (i.e., when being refined, giving cells on Ω^{l+1}, a coarse-grid cell remains part of Ω^l). A cell on the coarser grids is

called *parent* and its descendants are called its *kids*. In this way all cells in the geometric structure belong to a *quad-tree* structure.

Without loss of generality we take for the coarsest level $l = 0$, and for finer levels $l > 0$. For the smallest integer coordinates on the coarsest grid, Ω^0, without loss of generality we take $i = 0$ and $j = 0$. The integer coordinates of a cell on Ω^{l+1} are such that the kids of $\Omega^l_{i,j}$ are denoted by $\Omega^{l+1}_{2i,2j}$, $\Omega^{l+1}_{2i+1,2j}$, $\Omega^{l+1}_{2i,2j+1}$ and $\Omega^{l+1}_{2i+1,2j+1}$. A cell vertex in the physical domain is $P^l_{i,j} = M^l(\xi^l_{i,j}, \eta^l_{i,j})$, where, without loss of generality, $\widetilde{P}^l_{i,j} = (\xi^l_{i,j}, \eta^l_{i,j}) = (2^{-l}i, 2^{-l}j)$. In this way, implicitly we defined $P^l_{i,j}$, $(i,j,l) \in I$, as the south-west vertex of cell $\Omega^l_{i,j}$:

$$P^l_{i,j} = \partial\Omega^l_{i,j,S} \cap \partial\Omega^l_{i,j,W}.$$

Functions, function spaces and subdomains defined for the computational domain are identified by a tilde on the same symbol used for the physical domain.

2.2.2. The sequence of grids. The geometric structure described so far, is used for multigrid computations on a locally refined grid. The grid Ω^{l_b} on some basic level $l_b \geq 0$ covers $M^{l_b}(\widetilde{\Omega})$. The grids Ω^l, $l > l_b$, are adaptively constructed during the computation, when it is decided that cells should be refined or refinements should be deleted, depending on the computed solution. At some stage in the computation, a sequence of grids $\{\Omega^l\}_{l=0,\ldots,L}$ has been generated, where L is the highest level present. Thus, the cells on grid Ω^l, $l > l_b$ typically do *not* cover all of the domain Ω. Therefore, the grid $\Omega^l = \Omega^l_f \cup \Omega^l_c$ consists of a part Ω^l_f of which the cells have been refined (for which kids exist on level $l+1$) and a part Ω^l_c, with cells that have not been refined (without kids). Obviously we have

$$\begin{aligned} \Omega^l_f \cap \Omega^l_c &= \emptyset, \quad \forall l \in \{0, \ldots, L\}, \\ \Omega^l_c &= \emptyset, \quad \forall l \in \{0, \ldots, l_b - 1\}, \\ \Omega^L_f &= \emptyset. \end{aligned}$$

The set of all non-refined cells is called the *composite grid* Ω_c, defined by

$$\Omega_c = \bigcup_{l=l_b}^{L} \Omega^l_c.$$

Further, we define sets of indices associated with the different grids and

parts of grids by

$$
\begin{aligned}
I^l &= \{(i,j) \in \mathbb{Z}^2 \mid (i,j,l) \in I\},\\
I_c^l &= \left\{(i,j) \in I^l \mid \Omega_{i,j}^l \subset \Omega_c^l\right\},\\
I_c &= \{(i,j,l) \in I \mid (i,j) \in I_c^l\},\\
I_f^l &= \left\{(i,j) \in I^l \mid \Omega_{i,j}^l \subset \Omega_f^l\right\},\\
I_f &= \{(i,j,l) \in I \mid (i,j) \in I_f^l\}.
\end{aligned}
$$

For practical purposes we introduce $K : \mathbb{Z}^2 \to (\mathbb{Z}^2)^4$, defined by

$$K(i,j) = \{(2i,2j),(2i+1,2j),(2i,2j+1),(2i+1,2j+1)\}. \tag{2.2.3}$$

The boundary $\partial\Omega^l$ on level l is defined by

$$\partial\Omega^l = \text{ boundary of } \bigcup_{(i,j)\in I^l} \overline{\Omega_{i,j}^l}.$$

The part of this boundary coinciding with the boundary of the physical domain of definition, is denoted by $\partial\Omega_B^l$ and defined through

$$\partial\Omega_B^l = M^l(\partial\widetilde{\Omega}^l \cap \partial\widetilde{\Omega}).$$

Following [1] and [16], the part of the boundary of the subdomain Ω^l, which does not coincide with $\partial\Omega_B^l$, is called a *green* boundary denoted by $\partial\Omega_g^l$, and defined as

$$\partial\Omega_g^l = \partial\Omega^l \setminus \partial\Omega_B^l.$$

Cells on Ω^l that border a green boundary are called *green cells*.

A grid is called *uniform* if, in the computational domain, it covers all of $\widetilde{\Omega}$ and if it is not refined anywhere; it is called *locally uniform* with respect to a discrete operator in a cell, if no green boundary is involved in the definition of the operator for that cell. Otherwise the grid is called *locally non-uniform* in that cell, for that discrete operator. A cell is not necessarily a green cell, if a grid is locally non-uniform for the discretisation, in that cell. A grid that contains locally non-uniform cells, is called a *locally refined* grid. A composite grid that consists of cells from more than one level, is a locally refined grid.

2.3. Finite volume discretisation on a locally refined grid

2.3.1. The grid. The grid Ω^l on level l consists of cells $\Omega_{i,j}^l$, $(i,j) \in I^l$. A cell $\Omega_{i,j}^l$ in the physical domain is the result of $M^l(\widetilde{\Omega}_{i,j}^l)$. The mapping M^l is an approximation of M. We assume that M^l is a continuous mapping, and piecewise affine on each cell face $\partial\widetilde{\Omega}_{i,j,k}^l$. Then the grid Ω^l in the physical space is a collection of disjoint *quadrilaterals*. In that situation, M^l can

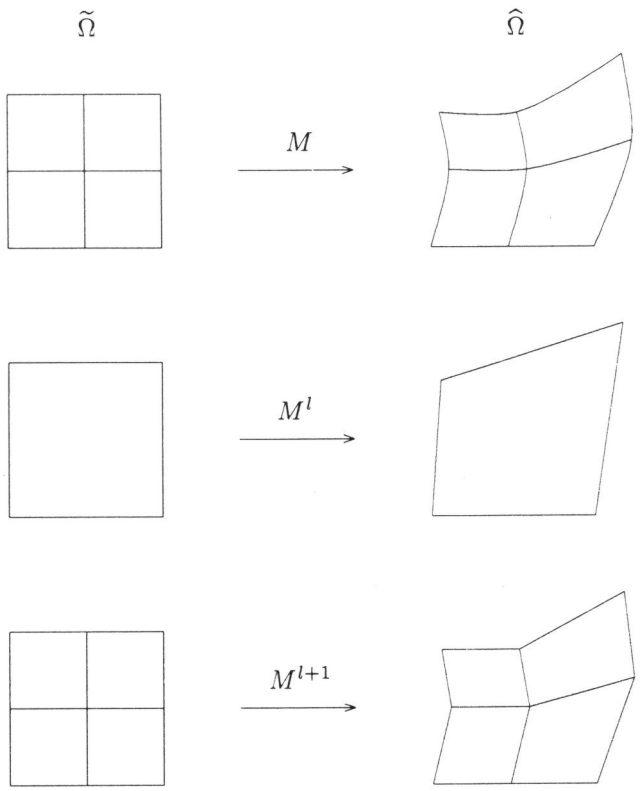


FIGURE 2.3.1. *Example of discretisation of a domain on different levels of refinement.*

be represented as a vector of two continuous functions, which are piecewise bilinear in each $\widetilde{\Omega}^l_{i,j}$. Furthermore we assume

$$(P^l_{i,j})^T = M^l(\widetilde{P}^l_{i,j}) = M(\widetilde{P}^l_{i,j}), \tag{2.3.1}$$

i.e., M^l is exact in the vertices $\widetilde{P}^l_{i,j}$. An example of this discretisation of a domain is shown in Fig. 2.3.1. Here, a set of 2×2 cells in $\widetilde{\Omega}^{l+1} \subset \widetilde{\Omega}$ is mapped to its image in $\widehat{\Omega}$ by applying M. This gives the general result as shown in the top figure of Fig. 2.3.1. The coarse-grid cell of $\widetilde{\Omega}^l$ results in the quadrilateral in Ω^l, when subjected to the mapping M^l, as shown in the middle figure. Finally, M^{l+1} applied to the set of 2×2 cells gives the set of quadrilaterals shown in the bottom figure. Notice that in general $M^l(\widetilde{\Omega}^*) \neq M^{l+1}(\widetilde{\Omega}^*)$, $l < \infty$.

The boundary $\partial\Omega^l_{i,j}$ consists of the four faces $\partial\Omega^l_{i,j,k}$, $k = N, E, S, W$. A cell face $\partial\Omega^l_{i,j,k}$ has a length denoted by $s^l_{i,j,k}$. The area of a cell $\Omega^l_{i,j}$ is

denoted by $A^l_{i,j}$.

2.3.2. Grid functions. Here we introduce function spaces for the various grids. If a function $u : \Omega^l \to \mathbb{R}^d$ is defined then also a function $\widetilde{u} : \widetilde{\Omega}^l \to \mathbb{R}^d$ is defined through

$$\widetilde{u}(\xi, \eta) = u(x^l(\xi, \eta), y^l(\xi, \eta)),$$

where x^l and y^l are given by

$$\begin{pmatrix} x^l(\xi, \eta) \\ y^l(\xi, \eta) \end{pmatrix} = M^l(\xi, \eta). \tag{2.3.2}$$

The set $X_s \subset \mathbb{R}^d$ is a d-dimensional vector space of *intrinsic quantities*. The function value of $q(x', y')$, e.g., an (approximate) solution of the conservation law and appropriate boundary conditions (2.1.1), at some point $(x', y') \in \overline{\Omega}$, is called the *state* at (x', y'). The state $q(x', y')$ denotes an intrinsic physical quantity. We define the function space $\overline{X}(\Omega^*)$ of such intrinsic-quantity functions defined on $\Omega^* \subset \widehat{\overline{\Omega}}$ by

$$\overline{X}(\Omega^*) = \{q : \Omega^* \to X_s \mid \Omega^* \subset \widehat{\overline{\Omega}}\}.$$

To each component of $q(x, y)$, a physical meaning can be assigned. From a physical point of view it is often clear that $q(x, y)$ can only take values from a restricted set of X_s. This is called the set of *admissible states*, and denoted by $X_a \subset X_s$. We introduce the function space $X(\Omega^*)$, defined by

$$X(\Omega^*) = \{q \in \overline{X}(\Omega^*) \mid q(x', y') \in X_a, \quad \forall (x', y') \in \Omega^*\}.$$

It is clear that solutions of (2.1.1) are sought in $X(\overline{\Omega})$. We also introduce a d-dimensional vector space of *extrinsic quantities*, denoted by $Y_s \subset \mathbb{R}^d$, and a corresponding function space

$$\overline{Y}(\Omega^*) = \{r : \Omega^* \to Y_s \mid \Omega^* \subset \widehat{\Omega}\}.$$

As opposed to an intrinsic quantity, an extrinsic quantity is a physical quantity that depends on the amount of matter considered. For example, density is an intrinsic quantity and mass an extrinsic quantity. Extrinsic and intrinsic quantities are related through the volume occupied by the matter considered. A function $q \in \overline{X}(\overline{\Omega^*})$ is related to a function $r \in \overline{Y}(\Omega^*)$ through the integral over $\omega \subset \Omega^*$

$$r(\omega) = \int_\omega q \, d\Omega.$$

For discretisation purposes we distinguish subspaces for a level l, given by

$$\overline{X}^l(\Omega^*) = \{q \in \overline{X}(\Omega^*) \mid q \text{ is constant in each } \Omega^l_{i,j} \cap \Omega^*, (i, j) \in I^l\},$$

and

$$X^l(\Omega^*) = \{q \in \overline{X}^l(\Omega^*) \mid q(x', y') \in X_a, \forall (x', y') \in \Omega^*\}.$$

Similarly, we introduce

$$\overline{Y}^l(\Omega^*) = \{r \in \overline{Y}(\Omega^*) \mid r \text{ is constant in each } \Omega^l_{i,j} \cap \Omega^*,\ (i,j) \in I^l\}$$

Notice that we have the nested sequence

$$\overline{X}^l(\Omega^*) \subset \overline{X}^{l+1}(\Omega^*) \subset \overline{X}(\Omega^*),$$

if $\Omega^* \subset \Omega^{l+1} \cap \Omega^l$. A similar nested sequence exists for $\overline{Y}$, $\overline{Y}^l$, $l = 0, \ldots, L$.

A function in $\overline{X}^l$ or $\overline{Y}^l$ is identified by the superscript l. For example, a solution of the set of discrete equations on a level l is a piecewise constant function $q^l \in X^l(\Omega^*)$. Furthermore, the value of q^l in a cell $\Omega^l_{i,j}$ is denoted by $q^l_{i,j}$, hence

$$q^l_{i,j} = q^l(x', y'), \quad (x', y') \in \Omega^l_{i,j} \cap \Omega^*.$$

The function value $q^l_{i,j}$ is called the *state* in cell $\Omega^l_{i,j}$.

Finally, we introduce the spaces of functions that are piecewise constant on the composite grid, which we denote by $\overline{X}_c(\Omega^*)$, $X_c(\Omega^*)$ and $\overline{Y}_c(\Omega^*)$ respectively. E.g., $\overline{X}_c(\Omega^*)$ is defined as

$$\overline{X}_c(\Omega^*) = \{q \in \overline{X}(\Omega^*) \mid q \text{ is constant in each } \Omega^l_{i,j} \cap \Omega^*,\ (i,j,l) \in I_c\},$$

and similar definitions for the other spaces X_c and $\overline{Y}_c$. A function from $\overline{X}_c(\Omega^*)$ is identified by the subscript c, and for $q_c \in \overline{X}_c(\Omega^*)$ we have

$$q_c(x', y') = q^l_{i,j}, \quad (x', y') \in \Omega^l_{i,j} \cap \Omega^*, \quad (i,j,l) \in I_c.$$

2.3.3. Restrictions. In order to define the relations between the different function spaces, it is appropriate to define a number of *restrictions*. The first three relate intrinsic vector functions and are identified by an overbar on the operator. The first is a projection and denoted by $\overline{R}^l : \overline{X}(\Omega^*) \to \overline{X}^l(\Omega^l)$, for any $\Omega^* \supset \Omega^l$, and given by the integral mean

$$\{\overline{R}^l u\}^l_{i,j} = \frac{1}{A^l_{i,j}} \int_{\Omega^l_{i,j}} u \, d\Omega. \tag{2.3.3}$$

A projection closely related to (2.3.3) is $\overline{R}^{l,l+1} : \overline{X}(\Omega^*) \to \overline{X}^l(\Omega^l_f)$, which restricts (2.3.3) to the refined part Ω^l_f of the grid:

$$\{\overline{R}^{l,l+1} u\}^l_{i,j} = \{\overline{R}^l u\}^l_{i,j}, \quad \forall (i,j,l) \in I_f. \tag{2.3.4}$$

Next, we define the restriction $\overline{R}^l_{l+1} : \overline{X}(\Omega^*) \to \overline{X}^l(\Omega^l_f)$, giving a cell-wise constant function, which in each $\Omega^l_{i,j} \subset \Omega^l_f$ delivers the integral mean of the operand over its kids. Let the collection of kids of $\Omega^l_{i,j}$ be

$$\Sigma^l_{i,j} = \bigcup_{m \in K(i,j)} \Omega^{l+1}_m,$$

then the restriction $\overline{R}^l_{l+1}$ is defined by

$$\{\overline{R}^l_{l+1} u\}^l_{i,j} = \frac{\int_{\Sigma^l_{i,j}} u \, d\Omega}{\int_{\Sigma^l_{i,j}} d\Omega}. \tag{2.3.5}$$

Note that $\overline{R}^{l,l+1} = \overline{R}^l_{l+1}$, if the grid is obtained by a piecewise bilinear mapping M^l and $M^{l+1} = M^l$.

Another set of three restrictions (denoted without the overbar) gives the relations between the extrinsic vector functions. The first restriction is the projection $R^l : \overline{Y}(\Omega^l) \to \overline{Y}^l(\Omega^l)$, defined by

$$\{R^l r\}^l_{i,j} = \int_{\Omega^l_{i,j}} r \, d\Omega.$$

The second restriction is the projection $R^{l,l+1} : \overline{Y}(\Omega^l) \to \overline{Y}^l(\Omega^l_f)$, defined by

$$\{R^{l,l+1} r\}^l_{i,j} = \{R^l r\}^l_{i,j}, \quad \forall (i,j,l) \in I^l_f.$$

Finally we define a restriction $R^l_{l+1} : \overline{Y}(\Omega^{l+1}) \to \overline{Y}^l(\Omega^l)$, which is related to $\overline{R}^l_{l+1}$ in (2.3.5) through the operators $A^l : \overline{X}(\Omega^l) \to \overline{Y}(\Omega^l)$ and $A^{l,l+1} : \overline{X}(\Omega^l_f) \to \overline{Y}(\Omega^l_f)$. These are defined as

$$A^l u(x', y') = A^l_{i,j} u(x', y'), \quad \forall (x', y') \in \Omega^l_{i,j},$$

and

$$A^{l,l+1} u(x', y') = A^l u(x', y'), \quad \forall (x', y') \in \Omega^l_{i,j} \text{ and } (i,j,l) \in I_f.$$

With these definitions, we define the last restriction by

$$R^l_{l+1} = A^{l,l+1} \overline{R}^l_{l+1} (A^{l+1})^{-1}. \tag{2.3.6}$$

Summary of notations. In this section we formally introduced a number of spaces of d-dimensional vector functions and appropriate relations between them. Function spaces of intrinsic quantities, such as the (numerical) solution of the problem, are denoted by X, X^l or X_c, for general functions, piecewise constant functions on level of refinement l or the composite grid respectively. The functions in these spaces can only take admissible values. With bars, $\overline{X}$, $\overline{X}^l$ and $\overline{X}_c$, we denote the spaces of functions which can take any value from $\mathbb{R}^d$. The spaces $\overline{Y}$, $\overline{Y}^l$ and $\overline{Y}_c$ are similar spaces for extrinsic quantities. Each of these is related to the space of intrinsic quantities with the same sub or superscripts through scaling with appropriate volume (area).

We introduced the restrictions $\overline{R}^l$, $\overline{R}^{l,l+1}$ and $\overline{R}^l_{l+1}$, which relate the function spaces of intrinsic quantities. The restrictions denoted without bars, R^l, $R^{l,l+1}$ and R^l_{l+1} relate the function spaces of extrinsic quantities. The operators denoted with bars deliver mean values of their operands. The operators denoted without bars deliver integrated values.

2.3.4. The system of discrete equations. In this section we describe the system of algebraic equations obtained by the discretisation. We distinguish between equations obtained for cells on the composite grid and equations obtained for refined cells.

Equations for a cell on a composite grid. A discretisation of the set of conservation laws (2.1.1a) on a composite grid is obtained by considering the weak formulation of the problem:
Find $q \in X(\overline{\Omega})$, which satisfies the boundary conditions and such that for all $\Omega^* \subset \Omega$

$$\int_{\Omega^*} \frac{\partial f(q(x,y))}{\partial x} + \frac{\partial g(q(x,y))}{\partial y}\, d\Omega = \int_{\Omega^*} s\, d\Omega. \tag{2.3.7}$$

We assume that q and s are defined on $\widehat{\Omega}$. In case $\widehat{\Omega} \setminus \Omega \neq \emptyset$, we assume that q and s, defined on Ω, are extended to $\widehat{\Omega}$ in a sufficiently regular way. Then (2.3.7) is approximated by

$$N(q) = r, \tag{2.3.8}$$

where $N : X(\widehat{\Omega}) \to \overline{Y}(\widehat{\Omega})$ and $r \in \overline{Y}(\widehat{\Omega})$ are given by

$$N(q) = \int_{\Omega^*} \frac{\partial f(q)}{\partial x} + \frac{\partial g(q)}{\partial y}\, d\Omega = \oint_{\partial\Omega^*} f(q)n_x + g(q)n_y\, ds, \tag{2.3.9a}$$

$$r = \int_{\Omega^*} s\, d\Omega \tag{2.3.9b}$$

for all $\Omega^* \subset \widehat{\Omega}$. In (2.3.9a) n_x and n_y are the components of the outward unit normal n on the boundary $\partial\Omega^*$, in x and y direction respectively. For

$\Omega^l_{i,j} \subset \widehat{\Omega}$ we define $R^l N : X(\widehat{\Omega}) \to \overline{Y}^l(\Omega^l)$ by

$$\{R^l N(q)\}^l_{i,j} \equiv \oint_{\partial\Omega^l_{i,j}} f(q)n_x + g(q)n_y \, ds.$$

The discretisation of the equations is obtained by requiring an approximation of (2.3.8) to hold for each cell on the composite grid. We first assume that for the discretisation the source term s is exactly integrated. In our notations this implies

$$r^l = R^l s. \tag{2.3.10}$$

The mean value of the flux across the kth cell face $\partial\Omega^l_{i,j,k} \subset \partial\Omega^l_{i,j}$ of cell $\Omega^l_{i,j}$ is

$$f^l_{i,j,k}(q) = \frac{1}{s^l_{i,j,k}} \int_{\partial\Omega^l_{i,j,k}} f(q)n_x + g(q)n_y \, ds.$$

Hence, a solution of (2.3.8) exactly satisfies

$$\{R^l N(q)\}^l_{i,j} = \sum_{k \in D} f^l_{i,j,k}(q) s^l_{i,j,k} = r^l_{i,j}, \quad \forall (i,j,l) \in I. \tag{2.3.11}$$

The equations (2.3.11) are approximated by approximating the mean fluxes $f^l_{i,j,k}$ across each cell face $\partial\Omega^l_{i,j,k}$ by a *numerical flux*, denoted by $F^l_{i,j,k}$. This numerical flux in general depends on the functions q^m, $m = l_b, \ldots, l$. On a level l the equations read for all $(i,j) \in I^l_c$

$$\sum_{k \in D} F^l_{i,j,k}(q^l; q^{l-1}, \ldots, q^{l_b}) s^l_{i,j,k} = r^l_{i,j}, \tag{2.3.12}$$

or in operator form

$$N^l(q^l; q^{l-1}, \ldots, q^{l_b}) = r^l,$$

where $N^l : X^l(\Omega^l) \to \overline{Y}^l(\Omega^l)$ is defined by

$$\{N^l(q^l; q^{l-1}, \ldots, q^{l_b})\}^l_{i,j} = \sum_{k \in D} F^l_{i,j,k}(q^l; q^{l-1}, \ldots, q^{l_b}) s^l_{i,j,k}.$$

Here, $q^{l-1}, \ldots, q^{l_b}$ act as parameters to N^l. These formulae define our discretisation on level l.

The numerical flux function. The numerical flux $F^l_{i,j,k}$ depends on the sequence $\{q^m\}_{m=l_b,\ldots,l}$. In this thesis we assume that the numerical flux can be written as

$$F^l_{i,j,k}(q^l; q^{l-1}, \ldots, q^{l_b}) = F((q^L)^l_{i,j,k}, (q^R)^l_{i,j,k}, n^l_{i,j,k}).$$

The arguments $(q^L)^l_{i,j,k}$ and $(q^R)^l_{i,j,k}$ denote estimates of the mean of q along $\partial\Omega^l_{i,j,k}$, dependent on $\{q^m\}_{m=l_b,\ldots,l}$, with a bias to the left and right side of $\partial\Omega^l_{i,j,k}$, respectively. The entry $n^l_{i,j,k} \in E$ denotes the unit normal on $\partial\Omega^l_{i,j,k}$, pointing outward from $\Omega^l_{i,j}$, where $E \subset \mathbb{R}^2$ is the unit circle in $\mathbb{R}^2$. The function $F : X_a \times X_a \times E \to \mathbb{R}^d$, given by $F(q^L, q^R, n)$, is an approximation of the flux $f(q)n_x + g(q)n_y$, with q^L and q^R in the neighbourhood of q.

There are various ways to define the states q^L and q^R and the numerical flux F. In fact, the choice of F and the states q^L and q^R determine the discretisation method and its accuracy. The left and right states are usually obtained by piecewise polynomial *reconstruction*, using discrete data, i.e., using $q^m \in X^l(\Omega^l)$, $m = l_b, \ldots, l$ (cf. Sec. 2.3.5). For example, for the flux we could use

$$F(q^L, q^R, n) = f(\tfrac{1}{2}(q^L + q^R))n_x + g(\tfrac{1}{2}(q^L + q^R))n_y.$$

Dependent on the choices for q^L and q^R, this usually gives rise to the second-order accurate difference scheme called the *central* scheme. This scheme is known to be unstable and requires additional dissipative terms in order to regain stability.

For our hyperbolic set of conservation laws (see [13]), we can take for F an approximate Riemann solver. This means that F approximates the flux $\widetilde{f}(\widetilde{q}_0)$, where $\widetilde{q}_0 = \widetilde{q}(0,t)$ is the solution at $x = 0$, $t > 0$ of the one-dimensional, hyperbolic initial value problem

$$\frac{\partial \widetilde{q}}{\partial t} + \frac{\partial \widetilde{f}(\widetilde{q})}{\partial x} = 0, \tag{2.3.13a}$$

$$\widetilde{q}(x,0) = \begin{cases} \widetilde{q}^L, & x < 0, \\ \widetilde{q}^R, & x > 0. \end{cases} \tag{2.3.13b}$$

Here $\widetilde{q} = \widetilde{q}(q,n)$ denotes the state q in a coordinate system along the normal n. The flux functions and states are related through

$$\widetilde{f}(\widetilde{q}) = \widetilde{f}(\widetilde{q}(q,n),n) = f(q)n_x + g(q)n_y$$

Problem (2.3.13) is called a one-dimensional Riemann problem. The exact solution of (2.3.13), with the numerical flux $F = \widetilde{f}(\widetilde{q}_0)$, was first used by Godunov [5], to obtain numerical solutions of one-dimensional hyperbolic problems. Godunov's scheme leads to an upwind discretisation of the system of partial differential equations. However, the exact solution of (2.3.13) is not differentiable. Differentiability is a prerequisite for the solution method, which

uses exact Newton iteration. A number of *approximate* Riemann solvers have been proposed. Discretisation schemes which use an approximate solution of a Riemann problem, associated with the system of hyperbolic conservation laws, are called Godunov-type schemes. The best-known approximate Riemann solvers are introduced in [7, 20, 15, 23, 14]. We use Osher's flux-difference splitting scheme as an approximate Riemann solver [14], unless stated otherwise. Osher's numerical flux function leads to a high resolution of discontinuities in the solution (both shocks and contact discontinuities). A system of nonlinear conservation laws generally does not have a unique solution. Osher's numerical flux only admits solutions which are entropy satisfying, i.e., the scheme only allows numerical solutions which for vanishing mesh width converge to the solution of the hyperbolic system, obtained in the limit of vanishing diffusion. This solution is the physically relevant solution. For a survey of flux splitting and flux-difference splitting schemes, we refer to [8], [24] and [11].

Equations for a refined cell. Discrete equations (2.3.12) are approximations of the conservation equation (2.3.7) for each cell that has not been refined. The left and right state for the computation of a numerical flux depend on the states in neighbouring cells, possibly on different levels. By definition, for a locally non-uniform grid cell, the left or right state for at least one cell face depends on coarse-grid states.

The set of equations obtained by applying the discretisation as described at the beginning of this subsection (2.3.4) are under-determined for a locally refined grid. If a neighbouring cell has been refined (has kids), that neighbour is not part of the composite grid, and there has not been defined an equation like (2.3.12) for the extra unknown. Additional equations, however, are easily obtained by

$$q^l_{i,j} = \{\overline{R}^l_{l+1} q^{l+1}\}^l_{i,j}, \quad \forall (i,j,l) \in I_f. \tag{2.3.14}$$

We use the equations (2.3.12) together with (2.3.14) to obtain approximate solutions on a locally refined (i.e. composite) grid.

2.3.5. Left and right states. Here we describe the computation of the left and right state used in the numerical flux function. We consider first-order and second-order accurate discretisations, both for a locally uniform and a locally non-uniform situation. We partly use the concept of reconstruction of piecewise C^∞ functions from the cell-wise constant data, $q^l_{i,j}$, associated with each cell. This concept was introduced in [22] and [26] for one-dimensional convection and extended and applied in [2] and [3] for unstructured grids in two space dimensions. Contrary to the work in [2] and [3], we do not reconstruct a single, unique (vector) function in each cell. However, we take care that in a locally non-uniform grid situation, the computation of the left and right state is such that the resulting scheme is consistent of the required

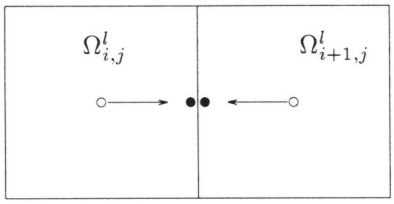


FIGURE 2.3.2. *Reconstruction for first-order consistency on a locally uniform grid.* ○: *available mean state;* ●: *left or right cell-face state.*

order, at least in some *weak* sense (cf. Sec. 2.4). This is done by making a different reconstruction for each side of each cell face.

Locally uniform composite grid. The computation of the states for the first-order as well as for the second-order consistent discretisation is a function of the mean states. On a locally uniform grid, first-order consistency is obtained by applying an $\mathcal{O}(h_l)$ accurate reconstruction. Consider for example the eastern cell face $\partial\Omega^l_{i,j,E}$ of a cell $\Omega^l_{i,j}$ on a locally uniform composite grid, where $(i+1,j) \in I^l$. For this situation we take for the states, as usual in first-order Godunov-type schemes, ([5], [10], [20]),

$$(q^L)^l_{i,j,E} = q^l_{i,j}, \tag{2.3.15a}$$

$$(q^R)^l_{i,j,E} = q^l_{i+1,j}. \tag{2.3.15b}$$

In Fig. 2.3.2 this is clarified. There, a mean state associated with a cell is denoted by a '○' in the centre of the cell.

For second-order consistency on a locally uniform grid, the states are based on $\mathcal{O}(h_l^2)$ accurate reconstructions of the state functions. This reconstruction can be done with a limiter, to suppress spurious wiggling of the solution (as proposed in e.g. [21] and [17] and applied in e.g. [12]), or without a limiter, like the κ-schemes [25] (as in e.g. [9], [18] and [12]). Again, for the eastern cell face of a cell $\Omega^l_{i,j}$ on a locally uniform grid, where $(i-1,j),(i+1,j),(i+2,j) \in I^l$, the limiter and κ-schemes are given by

$$(q^L)^l_{i,j,E} = C(q^l_{i-1,j},\, q^l_{i,j},\, q^l_{i+1,j}), \tag{2.3.16a}$$

$$(q^R)^l_{i,j,E} = C(q^l_{i+2,j},\, q^l_{i+1,j},\, q^l_{i,j}). \tag{2.3.16b}$$

This is schematically presented in Fig. 2.3.3. Here, $C : X_a \times X_a \times X_a \to X_a$ implements a κ-scheme or a limiter scheme. Notice that the κ-schemes are recovered by applying certain linear 'limiter' functions. However a κ-scheme does not satisfy monotonicity conditions (cf. [19]).

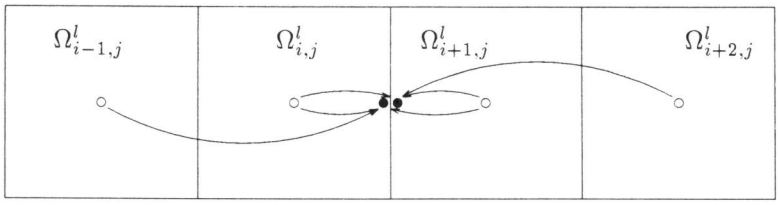


FIGURE 2.3.3. *Reconstruction for second-order consistency on a locally uniform grid.* ○*:available mean state;* ●*:left or right cell-face state.*

Virtual cells and states. On a locally refined grid, one or more of the mean states in (2.3.15) or (2.3.16) are not available, because the cells with which the states should be associated, do not exist. For this, we introduce the concept of *virtual cell* and associated *virtual state*.

With each integer coordinate pair $(2^n i + r, 2^n j + s) \notin I^{l+n}$, $0 \leq r, s < 2^n$, $n > 1$ and $(i,j) \in I^l$, $l_b \leq l \leq L - n$, we associate the virtual cell $\widetilde{\omega}^{l+}_{2^n i+r,2^n j+s} \subset \widetilde{\Omega}$, given by

$$\begin{aligned}\widetilde{\omega}^{l+n}_{2^n i+r,2^n j+s} =& 2^{-(l+n)}\,(2^n i + r, 2^n i + r + 1)\\ &\times 2^{-(l+n)}\,(2^n j + s, 2^n j + s + 1)\,.\end{aligned}$$

In the physical space the virtual cell $\omega^{l+n}_{2^n i+r,2^n j+s} \subset \widehat{\Omega}$ is defined as

$$\omega^{l+n}_{2^n i+r,2^n j+s} = M^{l+n}(\widetilde{\omega}^{l+n}_{2^n i+r,2^n j+s}). \tag{2.3.17}$$

Note that $\omega^{l+n}_{r,s}$ is exactly $\Omega^{l+n}_{r,s}$, if the grid would be sufficiently refined.

With each virtual cell $\omega^l_{i,j}$ we associate a virtual state $v^l_{i,j} \in X_a$, which can be interpreted as an approximation of the mean of the state vector function on $\omega^l_{i,j}$. In general a virtual state depends on the sequence $\{q^m\}_{m=l_b,\ldots,l}$.

Virtual cells and virtual states are used in the discretisation in the neighbourhood of green boundaries. To a large extent the virtual states determine the accuracy of the algebraic equations associated with the locally non-uniform grid.

Locally non-uniform grid. The concept of virtual states allows us to compute left and right states in a locally non-uniform grid situation, in a way similar to (2.3.15) and (2.3.16). The requirements to be satisfied for the proper computation of virtual states, are discussed in Sec. 2.4. Irrespective of the way to compute the virtual states, for first-order consistency we take for the eastern cell face of $\Omega^l_{i,j}$, similarly to (2.3.15), and $(i+1,j) \notin I^l$

$$(q^L)^l_{i,j,E} = q^l_{i,j}, \tag{2.3.18a}$$

$$(q^R)^l_{i,j,E} = v^l_{i+1,j}. \tag{2.3.18b}$$

Similar to (2.3.16), we take for second-order consistency, if $(i+1,j),(i+2,j) \notin I^l$

$$(q^L)^l_{i,j,E} = C\left(q^l_{i-1,j},\, q^l_{i,j},\, v^l_{i+1,j}\right), \tag{2.3.19a}$$

$$(q^R)^l_{i,j,E} = C\left(v^l_{i+2,j},\, v^l_{i+1,j},\, q^l_{i,j}\right), \tag{2.3.19b}$$

and if $(i+1,j) \in I^l$, but $(i+2,j) \notin I^l$

$$(q^L)^l_{i,j,E} = C\left(q^l_{i-1,j},\, q^l_{i,j},\, q^l_{i+1,j}\right), \tag{2.3.20a}$$

$$(q^R)^l_{i,j,E} = C\left(v^l_{i+2,j},\, q^l_{i+1,j},\, q^l_{i,j}\right), \tag{2.3.20b}$$

Formulae similar to (2.3.18)–(2.3.20) are used for the cell faces, $\partial\Omega^l_{i,j,k}$, $k = N, S, W$.

Boundary conditions. In the situation where a cell face coincides with the boundary $\partial\Omega^l$ of the domain Ω^l, a cell face state is determined by the boundary condition. We have for $\partial\Omega^l_{i,j,E} \subset \partial\Omega^l_B$

$$B^l_{i,j,E}((q^B)^l_{i,j,E}; q^l) = 0, \tag{2.3.21a}$$

$$(q^R)^l_{i,j,E} = (q^B)^l_{i,j,E}, \tag{2.3.21b}$$

where $B^l_{i,j,k}((q^B)^l_{i,j,k}; q^l) = 0$ is a discretisation of (2.1.1b) and determines a boundary state $(q_B)^l_{i,j,k}$ for $\partial\Omega^l_{i,j,k}$. In [10] approximations of boundary conditions are given by $B^l_{i,j,k}$, consistent with Osher's approximate Riemann solver. We assume that the discretisation of the boundary conditions, (2.3.21), has been incorporated in the discrete finite volume operators N^l, $l = 0, \ldots, L$.

2.4. Error analysis of the discretisation

2.4.1. Introductory remarks. In this section we study the local discretisation error and consistency of the discretisation described in Sec. 2.3. In the discretisation we distinguish three approximations, each of which have a contribution to the local discretisation error. These contribution are:

- approximation of the mapping from the computational space into the physical space; in equations this error is denoted by $\tau^l_m(q)$;
- approximation of the mean flux on a cell face by the flux evaluated at the mean state along the cell face (quadrature rule); in equations this error is denoted by $\tau^l_q(q)$;
- approximation of the mean state on a cell face by biased reconstructions; in equations this error is denoted by $\tau^l_r(q)$.

Often the approximation of M by M^l is not essential, since the change from M to M^l, merely changes the partitioning in the physical domain slightly, without affecting the accuracy of the resulting set of algebraic equations. However, it does affect the approximation of the domain of definition of the problem. Hence, for the interior cells of $\widetilde{\Omega}$ we can assume $M = M^l = M^{l_0}$,

for some constant l_0 and $l_0 \leq l \leq L$, without affecting the accuracy. At the domain-boundary, $\partial\widetilde{\Omega}$, the error of this approximation M^l can be important. Here, using $M = M^l = M^{l_0}$ is not possible in general, because it results in an approximation of the boundary, independent of the level l. Hence, the geometry of the discrete problem does not converge to the geometry of the continuous problem under consideration with increasing level of refinement. A more important reason to study this approximation is related to a-posteriori estimation of the local discretisation error, applying τ-extrapolation, a truncation error extrapolation technique, similar to Richardson's extrapolation. This uses the so-called relative local discretisation errors of two consecutive levels of refinement (cf. [4]). In Chap. 4, the relative local discretisation error is used for the a-posteriori estimation of the local discretisation error. In short, the relative local discretisation error is the difference in error of the discretisations on two consecutive levels. When a-posteriori estimating the relative local truncation error, the difference between the discretisations on two consecutive levels of refinement is estimated. Hence, the asymptotic relation between the discretisations of the two levels—including the difference caused by the difference of mappings on the two levels—is taken into account. Therefore, we study this relation here.

The two errors $\tau_q^l(q)$ and $\tau_r^l(q)$ together make up the local truncation error. For the fluxes approximated with the projection of the solution of the continuous problem, i.e. $q = \bar{q}$, the exact solution of the problem, they form the error in the system of algebraic equations, for a given particular partitioning.

The analysis presented in this section leads to requirements to be imposed on the reconstruction of cell-wise smooth functions from the cell-wise constant numerical data. These requirements, depend on the goal set out to be reached in a particular discretisation. We distinguish between the goals

- obtain a given order of consistency;
- obtain a given order of discrete convergence.

This distinction is made, since a certain order of local discretisation error for *all* equations may not be essential to obtain a given order of discrete convergence. Assume a total number of equations of $\mathcal{O}(h^{-2})$. An $\mathcal{O}(h^{-1})$ number of equations with low-order accuracy may not affect the rate of discrete convergence, not even in supremum norm. To study this in detail, we have to redefine the notion of consistency for a non-uniform mesh. In fact, we define a slightly *weaker* form of consistency. In absolute value, this weaker form is the discrete L_1-norm of the local truncation error for a collection of discrete equations. The requirements for consistency in both the weak and the usual sense are studied. Consider the situation where a given order of discrete convergence is to be obtained and local grid refinement is applied with the local discretisation error used in the refinement criterion. Then, an equally high-order local discretisation error is required for all algebraic equations, since a refinement decision based on the local discretisation error is based on the im-

plication that by replacing coarse-grid equations by fine-grid equations, this error in some norm is reduced at a sufficient rate. If there would be no reduction, or no sufficient reduction of local error, local grid refinement would become inefficient or even ineffective. Hence, consistency in the usual sense is important when refinement based on local discretisation error is used. From this point we use the term 'consistency' for consistency in the usual sense and 'weak consistency' for consistency in the weak sense.

2.4.2. Approximation of the mapping. In order to be able to make an a-posteriori estimation of the local discretisation error, with sufficient accuracy, we first study the consequences of approximating M by M^l. Actually, we are only interested in the relation between the mappings for two consecutive levels of refinement, since we want to study the use of two consecutive levels of refinement in the estimation of the local discretisation error. The relation between the mappings of two consecutive levels can be established through their relation to M. This relation also allows us to study the accuracy of the restriction to the coarse grid (cf. Sec. 2.4.3).

We consider a surjective mapping $M : \overline{\widetilde{\Omega}} \to \overline{\Omega}$, which is injective on $\widetilde{\Omega}$. We also consider its continuous, piecewise bilinear approximation associated with level of refinement l, $M^l : \overline{\widetilde{\Omega}} \to \overline{\widetilde{\Omega}^l}$, which is exact in the vertices $\widetilde{P}^l_{i,j,k}$. To simplify notations, in the present local analysis, we drop the indices which are constant throughout this part of the analysis. We consider a cell on level of refinement l. In the computational space, the corresponding cell is denoted by $\widetilde{\Omega}$. Its images in $\widehat{\Omega}$ are

$$\Omega = M(\widetilde{\Omega}),$$
$$\Omega^l = M^l(\widetilde{\Omega}),$$

where we dropped the subscripts. We use a local Cartesian coordinate system (ξ, η) in the computational space, with its origin in the center of $\widetilde{\Omega}$. Similar to (2.3.2), we use the Cartesian coordinates in the physical space obtained by M

$$\begin{pmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{pmatrix} = M(\xi,\eta),$$

where the origin in the physical space is

$$(0,0)^T = M(0,0). \tag{2.4.1}$$

We assume $\widetilde{\Omega}$ is the square with sides of length $h_l = 2h$, $\widetilde{\Omega} = (-h,h)^2$. Hence the area of the cell in the computational space is $4h^2$. In the physical space the area is denoted by A for Ω and A^l for Ω^l. The Jacobians of M and M^l

are denoted by J and J^l respectively:

$$J(\xi,\eta) = x_\xi y_\eta - x_\eta y_\xi,$$
$$J^l(\xi,\eta) = x^l_\xi y^l_\eta - x^l_\eta y^l_\xi,$$

where the subscripts denote differentiation with respect to ξ and η respectively. Furthermore, we assume that M is sufficiently smooth and for $(\xi,\eta) \in \widetilde{\Omega}$ can be written as

$$\begin{aligned} M = {} & m_0 + m_1\xi + m_2\eta + m_3\xi^2 + m_4\xi\eta + m_5\eta^2 \\ & + m_6\xi^3 + m_7\xi^2\eta + m_8\xi\eta^2 + m_9\eta^3 + \dots . \end{aligned} \tag{2.4.2}$$

Note that according to (2.4.1) $m_0 = (0,0)^T$. The mapping M^l is piecewise bilinear, hence it can be written as

$$M^l = m^l_0 + m^l_1\xi + m^l_2\eta + m^l_4\xi\eta.$$

We define $(x_i, y_i)^T$, $(x^l_i, y^l_i)^T$ by

$$m_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}, \quad i = 0,1,2,\dots, \tag{2.4.3}$$

$$m^l_i = \begin{pmatrix} x^l_i \\ y^l_i \end{pmatrix}, \quad i = 0,1,2,4. \tag{2.4.4}$$

From the exactness of M^l at the cell vertices, given by (2.3.1), we can express m^l_i, $i = 0,1,2,4$ in terms of m_i, $i = 0,1,\dots$. This gives

$$m^l_0 = m_0 + m_3h^2 + m_5h^2 + \mathcal{O}(h^4), \tag{2.4.5a}$$
$$m^l_1 = m_1 + m_6h^2 + m_8h^2 + \mathcal{O}(h^4), \tag{2.4.5b}$$
$$m^l_2 = m_2 + m_7h^2 + m_9h^2 + \mathcal{O}(h^4), \tag{2.4.5c}$$
$$m^l_4 = m_4 + \mathcal{O}(h^4). \tag{2.4.5d}$$

As a result, we have M^l expressed in terms of M, for $|\xi|, |\eta| \le h$, given by

$$\begin{aligned} M^l = {} & m_0 + m_1\xi + m_2\eta + m_3h^2 + m_4\xi\eta + m_5h^2 \\ & + m_6h^2\xi + m_7h^2\eta + m_8h^2\xi + m_9h^2\eta + \mathcal{O}(h^4). \end{aligned} \tag{2.4.6}$$

We find from (2.4.2) and (2.4.6) that

$$M^l - M = m_3(h^2 - \xi^2) + m_5(h^2 - \eta^2) + \mathcal{O}(h^3).$$

We are now interested in the difference between the weak form (2.3.9), for $\Omega = M(\widetilde{\Omega})$ and $\Omega^l = M^l(\widetilde{\Omega})$, divided by the respective areas A and A^l. This error is denoted by $\tau_m(q)$.

Let a sufficiently smooth, integrable function $w : \Omega \cup \Omega^l \to \mathbb{R}^d$ be defined, and let its Taylor series expansion around the origin $(x, y) = (0, 0)$ be given by

$$w(x, y) = w_0 + w_1 x + w_2 y + w_3 x^2 + \dots . \tag{2.4.7}$$

For the error of the weak form on Ω^l with respect to the weak form on Ω, it suffices to consider the difference

$$t(w) \equiv \frac{1}{A^l} \int_{\Omega^l} w \, d\Omega - \frac{1}{A} \int_{\Omega} w \, d\Omega, \tag{2.4.8}$$

for any sufficiently smooth and integrable function w. For M and M^l, assuming from this point that $J, J^l > 0$, this is equal to

$$t(w) = \frac{1}{A^l} \int_{\widetilde{\Omega}} w^l J^l \, d\widetilde{\Omega} - \frac{1}{A} \int_{\widetilde{\Omega}} w J \, d\widetilde{\Omega},$$

where $w^l = w(x^l, y^l)$. With

$$\delta_w = w^l - w,$$
$$\delta_J = J^l - J,$$

the integrand $w^l J^l$ can be written as

$$w^l J^l = wJ + \delta_w J + w\delta_J + \delta_w \delta_J .$$

An expression for δ_w in terms of m_i and w_i can be obtained by using (2.4.7) for (x^l, y^l) and (x, y), and by subtraction and substitution of (2.4.5). From this exercise it appears that $\delta_w = \mathcal{O}(h^2)$. Multiplication with J and integration over $\widetilde{\Omega}$ yields

$$\int_{\widetilde{\Omega}} \delta_w J \, d\widetilde{\Omega} = \frac{8}{3} J_0 h^4 \{w_1(x_3 + x_5) + w_2(y_3 + y_5)\} + \mathcal{O}(h^5), \tag{2.4.9}$$

where J_0 is $J(0, 0)$.

Similarly, based on (2.4.5), we can also find an expression for δ_J. It appears that generally $\delta_J = \mathcal{O}(h)$, with the $\mathcal{O}(h)$ terms linear in ξ or η. A straightforward calculation yields for the integral of $w\delta_J$

(2.4.10)

$$\begin{aligned}
&\int_{\widetilde{\Omega}} w\delta_J \, d\widetilde{\Omega} = \\
&\quad \frac{8}{3} h^4 \{ w_0(x_1 y_7 - x_7 y_1 + x_8 y_2 - x_2 y_8 - x_3 y_4 + x_4 y_3 - x_4 y_5 + x_5 y_4) \\
&\qquad + w_1(-x_1 x_3 y_2 + x_1 x_2 y_3 - x_1 x_2 y_5 + x_2 y_5 y_1) \\
&\qquad + w_2(-x_3 y_1 y_2 + x_2 y_1 y_3 - x_1 y_2 y_5 + x_5 y_1 y_2)\} + \mathcal{O}(h^5).
\end{aligned}$$

The term $\delta_w \delta_J$ gives only an $\mathcal{O}(h^5)$ contribution to the integral over $\widetilde{\Omega}$.

By (2.4.9) and (2.4.10) we can define $C(w)$ as the coefficient in the first term of an asymptotic expansion, given by the sum of (2.4.9) and (2.4.10), i.e.,

$$\int_{\widetilde{\Omega}} (w^l J^l - wJ)\, d\widetilde{\Omega} = h^4 C(w) + \mathcal{O}(h^5),$$

with C independent of h. Now we can write

$$A^l - A = h^4 C(1) + \mathcal{O}(h^5). \tag{2.4.11}$$

For the area A we have

$$A = 4h^2 J_0 + \mathcal{O}(h^4), \tag{2.4.12}$$

combination of (2.4.11) and (2.4.12) yields after some manipulation

$$\frac{1}{A^l} - \frac{1}{A} = -\frac{C(1)}{16 J_0^2} + \mathcal{O}(h).$$

Using this, it can be easily shown that the difference $t(w)$ satisfies

$$t(w) = \frac{h^2}{4J_0}(C(w) - w_0 C(1)) + \mathcal{O}(h^3). \tag{2.4.13}$$

With

$$\tau_m^l(q) = t\left(\frac{\partial f(q)}{\partial x} + \frac{\partial g(q)}{\partial y}\right),$$

we establish the asymptotic relation between the approximation of the weak form on two consecutive levels of refinement, caused by the approximation of the grid geometry. Although the difference between the partitioning obtained with M and the partitioning obtained with M^l itself is not essential for the convergence of the local discretisation error, we use the result (2.4.13) when we show that an a-posteriori error estimation is sufficiently accurate (cf. Chap. 4). Furthermore, we use (2.4.13) to show that on a composite grid, the equations for a refined cell (2.3.14), in general give at best second-order accuracy on the coarse grid. Assume for (2.4.13) that w is the differential operator, applied to the solution of the continuous problem. According to the result in (2.4.13), in general the discrete equations derived for boundary cells are at most second-order accurate, because (in general) we cannot use $M = M^l = M^{l_0}$, l_0 a constant, $l_0 \leq l \leq L$, in cells along the boundary of the domain. A piecewise bilinear mapping $M = M^l$ would yield $\tau_m^l(q) = 0$.

The above conclusions do not hold if a grid describes the geometry exactly, i.e., if the sequence of grids is exactly a (locally) nested sequence. E.g., this is the case if $M = M^l$, $\forall l \in \{0, \ldots, L\}$.

Note that for a homogeneous problem the difference between discretisations caused by the difference in mappings on the two levels, l and $l+1$ is zero. For

a homogeneous problem both $\tau_m^l(q)$ and $\tau_m^{l+1}(q)$ are zero. The error due to the mappings M^l, approximating M, is of minor importance in that situation.

2.4.3. Accuracy of the coarse-grid restriction. We consider the restriction $\overline{R}_{l+1}^l$, as defined by (2.3.5). We study the difference between the restrictions $\overline{R}_{l+1}^l \overline{R}^{l+1} q$ (cf. (2.3.3) and (2.3.5)) and $\overline{R}^{l,l+1} q$ (cf. (2.3.4)). In the computation of the virtual states we use as the coarse-grid discrete function on Ω_f^l, the restricted function $q_{i,j}^l = \{\overline{R}_{l+1}^l q^{l+1}\}_{i,j}^l$, $(i,j) \in I_f^l$, where $q^{l+1} = \overline{R}^{l+1} q$, while we assume that $q_{i,j}^l$ is the mean of q on $\Omega_{i,j}^l$, given by $\overline{R}^{l,l+1} q$. The error we make with this assumption is studied here.

In our analysis we again drop the unnecessary indices. We consider a cell Ω^l on level l, which has refinements on level $l+1$. A kid of Ω^l is identified by the subscript $m \in K$, where K is the set of indices (cf. (2.2.3)), associated with the cell Ω^l. For level l we have the approximation M^l of M and for level $l+1$ we have M^{l+1}. Similar to the notations in the previous subsection, we use the superscripts in our notations to distinguish between variables for $\Omega = M(\widetilde{\Omega})$ (e.g., A, no superscript) and $\Omega^l = M^l(\widetilde{\Omega})$ (e.g., A^l). Hence e.g., A_m^{l+1}, $m \in K$, denotes the area of the mth kid $M^{l+1}(\widetilde{\Omega}_m^{l+1})$.

Consider $\overline{R}_{l+1}^l q^{l+1}$, where $q^{l+1} = \overline{R}^{l+1} q$. By definition of $\overline{R}_{l+1}^l$ we have

$$\overline{R}_{l+1}^l \overline{R}^{l+1} q = \frac{\sum_{m \in K} A_m^{l+1} q_m^{l+1}}{\sum_{m \in K} A_m^{l+1}}. \tag{2.4.14}$$

With the results from the previous subsection we find

$$\frac{1}{\sum_{m \in K} A_m^{l+1}} = \frac{1}{A} - \frac{h_{l+1}^4 C(1)}{4A^2} + \mathcal{O}(h_{l+1}). \tag{2.4.15}$$

For the numerator in (2.4.14) we find

$$\sum_{m \in K} A_m^{l+1} q_m^{l+1} = \int_{\widetilde{\Omega}} qJ \, d\widetilde{\Omega} + \frac{1}{4} h_{l+1}^4 C(q) + \mathcal{O}(h_{l+1}^5). \tag{2.4.16}$$

Multiplication of (2.4.15) and (2.4.16) gives for (2.4.14):

$$\overline{R}^{l,l+1} \overline{R}^{l+1} q = \frac{1}{A} \int_{\widetilde{\Omega}} qJ \, d\widetilde{\Omega} + \frac{h_{l+1}^2}{16 J_0} (C(q) - q_0 C(1)) + \mathcal{O}(h_{l+1}^3).$$

With (2.4.8) and (2.4.13) we also have

$$\overline{R}^l q = \frac{1}{A} \int_{\widetilde{\Omega}} qJ \, d\widetilde{\Omega} + \frac{h_{l+1}^2}{4 J_0} (C(q) - q_0 C(1)) + \mathcal{O}(h_l^3).$$

Hence, the difference between the two restrictions is

$$\overline{R}^{l,l+1} q - \overline{R}_{l+1}^l \overline{R}^{l+1} q = \frac{3}{16} \frac{h_{l+1}^2}{J_0} (C(q) - q_0 C(1)) + \mathcal{O}(h_l^3).$$

2.4.4. Consistency and weak consistency. Here we consider the approximation of the weak form on a given partitioning by the set of algebraic equations, resulting from the discretisation. Hence, we assume that $M = M^l = M^{l_0}$, where $l_0 < l$. With (2.2.1) this implies $\Omega = M^l(\widetilde{\Omega})$. This in turn implies that we do not consider the differences between the discretisation on different levels, due to the differences in mappings from the computational into the physical space, on each level.

Local discretisation error and consistency. Consider the function spaces and sets of indices as defined in Sec. 2.2 and 2.3. Let $N(q) = r$ denote the continuous equations to be satisfied by the solution of the problem in weak formulation, for some partitioning of the domain. We denote the solution of the continuous problem by $\bar{q}$:

$$N(\bar{q}) = r.$$

Let the weak form N be approximated by the discrete finite volume operator N^l, on a level of refinement $l \in \{0, \ldots, L\}$, of a sufficiently smooth grid, consisting of quadrilaterals. Furthermore, let projections R^l and $\overline{R}^l$ be defined properly, for example by the definitions in Sec. 2.3.3, and related to each other through A^l, as in (2.3.6). Assume that the right-hand side of (2.1.1a) is exactly integrated, as denoted by (2.3.10). The local discretisation error for $N^l(q^l; q^{l-1}) = r^l$ is $\tau^l(\bar{q})$, where $\tau^l : X(\Omega^l) \to \overline{X}^l(\Omega^l)$ is given by

$$\tau^l(q) = (A^l)^{-1} \left(N^l(\overline{R}^l q; \overline{R}^{l-1} q) - R^l N(q) \right).$$

We define $\tau^l_{i,j}(q) \equiv \{\tau^l(q)\}^l_{i,j}$. The discretisation N^l is called an approximation to N of order of consistency p, if for all $(i, j, l) \in I$

$$\tau^l_{i,j}(\bar{q}) = \mathcal{O}(h_l^p),$$

for $h_l \to 0$. Notice that $h_l \to 0$, if $l \to \infty$.

Weak consistency. We also introduce a weak form of consistency, related to consistency in the usual sense. The new definition of consistency is weaker than the usual definition, because it considers the collective behaviour of the local discretisation error for a set of equations, rather than the behaviour of each equation separately. Consider a partitioning of the domain Ω, obtained by refining n times a previously obtained locally refined composite grid. Let a discretisation as described in Sec. 2.3 be defined for this new system. Then the *collective local truncation error*, $T^l_n(q)$, for an n times completely refined system is defined by

$$T^l_n(q) = A^{-1} R^l_{l+1} \ldots R^{l+n-1}_{l+n} \left(N^{l+n}(\overline{R}^q_{l+n}) - R^{l+n} N(q) \right).$$

In absolute value, this is the discrete L_1-norm of the local truncation error for the set of equations for all descendants of each cell $\Omega_{i,j}^l$. Note that

$$T_0^l(q) = \tau^l(q).$$

A discretisation N^l is called *weakly consistent of order p* if

$$\left\{T_n^l(\bar{q})\right\}_{i,j}^l = \mathcal{O}(h_{l+n}^p), \quad \forall (i,j,l) \in I,$$

for $n \to \infty$.

2.4.5. Analysis of local truncation error. In this section we study the local truncation error of the discretisation as introduced in Sec. 2.3. We use the same assumptions as formulated at the beginning of Sec. 2.4.4. From the analysis we obtain requirements for the computation of the virtual states, both for a consistent and weakly consistent discretisation.

As described at the beginning of Sec. 2.4, the local discretisation error consists of two contributions. The first contribution, which is denoted by $\tau_q^l(q)$, is a result of the quadrature rule that is used to approximate the mean value along a cell face, of the flux across that cell face. This mean value is approximated by the flux evaluated at the mean value of the state function along that face. Furthermore, the mean value of the state function q along a cell face is approximated by reconstruction of piecewise polynomial functions from a cell-wise constant function which represents the average of q in each cell. The reconstruction is done twice for each cell face, with a bias in both opposite directions. These reconstructed approximations of the mean value along a cell face is then used in an approximate Riemann solver. If q is a solution of the continuous equation (2.1.1a), this procedure gives a contribution to the local discretisation error, denoted by $\tau_r^l(q)$

Quadrature rule. Similar to the local analysis in Sec. 2.4.2, we drop the indices which are superfluous for this local analysis. As noted, the mapping M is assumed to be bilinear on $\widetilde{\Omega}$. It is given by

$$M(\xi,\eta) = m_0 + m_1\xi + m_2\eta + m_4\xi\eta, \tag{2.4.17}$$

where m_0, m_1, m_2 and m_4 are defined by (2.4.3). Note that the cell-wise constant parameters m_i are fully determined by the coordinates of the vertices of that cell. The area of cell Ω is

$$A = \int_\Omega d\Omega = \int_{\widetilde{\Omega}} J\, d\widetilde{\Omega} = 4h^2 J_0,$$

where $J_0 = x_1y_2 - x_2y_1 > 0$, the Jacobian of M at the origin.

We denote the mean value of the flux across the kth cell face by f_k and since M is linear at $\partial\widetilde{\Omega}_k$, we have

$$
\begin{aligned}
f_k(q) &= \frac{1}{s_k}\int_{\partial\Omega_k} f(q)n_x + g(q)n_y\, ds \\
&= \frac{1}{2h}\int_{\partial\widetilde{\Omega}_k} f(q)n_x + g(q)n_y\, d\widetilde{s}.
\end{aligned}
\tag{2.4.18}
$$

The mean of $q(x,y)$ along $\partial\Omega_k$ is denoted by

$$
q_k = \frac{1}{s_k}\int_{\partial\Omega_k} q\, ds.
$$

The unit normal along a cell face is constant, and we use the notation

$$
(n_x)_k = n_x|_{\partial\Omega_k}, \qquad (n_y)_k = n_y|_{\partial\Omega_k}.
$$

By q^* and q_k^* we denote

$$
\begin{aligned}
q^* &= (q, n_x, n_y)^T, \\
q_k^* &= (q_k, (n_x)_k, (n_y)_k)^T,
\end{aligned}
$$

and by f^* we denote the flux in the direction of $n = (n_x, n_y)^T$, given by

$$
f^*(q^*) = f(q)n_x + g(q)n_y.
$$

The contribution $\tau_q(q)$ of the quadrature rule to the local truncation error is given by

$$
\tau_q(q) = \frac{1}{A}\sum_{k\in D}\left(f^*(q_k^*) - f_k(q)\right)s_k.
$$

An expansion of $f^*(q^*)$ around q_k^* yields

$$
\begin{aligned}
f^*(q^*) = f^*(q_k^*) &+ \left.\frac{\partial f^*}{\partial q^{*\alpha}}\right|_{q_k^*}(q^* - q_k^*)^\alpha \\
&+ \frac{1}{2}\left.\frac{\partial^2 f^*}{\partial q^{*\alpha}\partial q^{*\beta}}\right|_{q_k^*}(q^* - q_k^*)^\alpha(q^* - q_k^*)^\beta + \dots,
\end{aligned}
\tag{2.4.19}
$$

where the superscripts denote the components of q^* and q_k^*, and the summation convention is used for $\alpha, \beta = 1, \dots, d+2$. We assume that the function $q(x(\xi,\eta), y(\xi,\eta))$ can be written as a Taylor series expansion around the origin of the local Cartesian coordinate system in the *computational* space, i.e., it is written as

$$
\begin{aligned}
q = q_0 &+ q_1\xi + q_2\eta + q_3\xi^2 + q_4\xi\eta + q_5\eta^2 \\
&+ q_6\xi^3 + q_7\xi^2\eta + q_8\xi\eta^2 + q_9\eta^3 + \dots.
\end{aligned}
\tag{2.4.20}
$$

For $k = E, W$, we use for ξ at $\partial\widetilde{\Omega}_k$ the constant $\xi_k = \xi|_{\partial\widetilde{\Omega}_k}$. If we now define

$$\begin{aligned} A_1 &= q_2 + q_4\xi_k + q_7\xi_k^2, \\ A_2 &= q_5 + q_8\xi_k, \\ A_3 &= q_9, \end{aligned}$$

and

$$\begin{aligned} Y_1 &= \eta, \\ Y_2 &= \eta^2 - \frac{1}{3}h^2, \\ Y_3 &= \eta^3, \end{aligned}$$

and use (2.4.18), (2.4.19) and (2.4.20) we can write the mean flux across the kth cell face as

(2.4.21)

$$\begin{aligned} f_k(q) &= \frac{1}{2h}\int_{-h}^{h} f^*(q^*|_{\xi_k})\,d\eta \\ &= \frac{1}{2h}\int_{-h}^{h}\left\{ f^*(q_k^*) + \left.\frac{\partial f^*}{\partial q^\alpha}\right|_{q_k^-} A_m^\alpha Y_m + \frac{1}{2}\left.\frac{\partial^2 f^*}{\partial q^\alpha \partial q^\beta}\right|_{q_k^-} \prod_{\nu=\alpha,\beta} A_m^\nu Y_m \right. \\ &\quad \left. + \frac{1}{6}\left.\frac{\partial^3 f^*}{\partial q^\alpha \partial q^\beta \partial q^\gamma}\right|_{q_k^-} \prod_{\nu=\alpha,\beta,\gamma} A_m^\nu Y_m + \mathcal{O}\left(\prod_{\alpha,\beta,\gamma\,\delta} A_m^\nu Y_m\right)\right\} d\eta. \end{aligned}$$

Integration of products of Y_m yields

(2.4.22a)
$$\frac{1}{2h}\int_{-h}^{h} Y_i\,d\eta = 0, \quad i = 1,2,3,$$

(2.4.22b)
$$\frac{1}{2h}\int_{-h}^{h} Y_iY_j\,d\eta = \begin{cases} \frac{1}{3}h^2 & \text{for } i = j = 1, \\ \mathcal{O}(h^4) & \text{otherwise,} \end{cases}$$

(2.4.22c)
$$\frac{1}{2h}\int_{-h}^{h} Y_iY_jY_k\,d\eta = \mathcal{O}(h^4), \quad i,j,k = 1,2,3.$$

Substitution of (2.4.22) into (2.4.21) and rearrangement of terms gives for the approximation of $f_k(q)$, $k = E, W$,

(2.4.23)
$$f^*(q_k^*) - f_k(q) = -\frac{1}{6}h^2 \left.\frac{\partial^2 f^*}{\partial q^\alpha \partial q^\beta}\right|_{q_k^-} (q_2^\alpha q_2^\beta + 2q_2^\alpha q_4^\beta \xi_k) + \mathcal{O}(h^4).$$

Since $f^* = fn_x + gn_y$, for the right-hand side of (2.4.23) we are interested in expressions like $(n_x)_k\, s_k\, \frac{\partial^2 f}{\partial q^\alpha \partial q^\beta}\Big|_{q_k}$. For $(n_x)_E\, s_E$ we find with (2.4.17)

$$(n_x)_E\, s_E = 2hy_\eta|_{\xi=h} = 2h(y_2 + y_4 h). \tag{2.4.24}$$

With (2.4.24), and with the aid of (2.4.19) and (2.4.20) we find

$$\begin{aligned}(n_x)_E\, s_E \left.\frac{\partial^2 f}{\partial q^\alpha \partial q^\beta}\right|_{q_E} &= \left.\frac{\partial^2 f}{\partial q^\alpha \partial q^\beta}\right|_{q_0} 2h(y_2 + y_4 h) \\ &+ \left.\frac{\partial^3 f}{\partial q^\alpha \partial q^\beta \partial q^\gamma}\right|_{q_0} 2h^2 \left((y_2 + y_4 h)q_1^\gamma + hy_2(q_3 + \frac{1}{3}q_5)^\gamma\right) \\ &+ \left.\frac{\partial^4 f}{\partial q^\alpha \partial q^\beta \partial q^\gamma \partial q^\delta}\right|_{q_0} 2h^3 y_2 q_1^\gamma q_1^\delta + \mathcal{O}(h^4).\end{aligned} \tag{2.4.25}$$

For $\partial\Omega_W$ we have

$$(n_x)_W\, s_W = -2h(y_2 - y_4 h).$$

Applying the changes in (2.4.25) given by

$$\begin{aligned} q_1 &\to -q_1, & q_E &\to q_W, \\ y_2 &\to -y_2, & (n_x)_E &\to (n_x)_W, \end{aligned} \tag{2.4.26}$$

yields an expression similar to (2.4.25) for $\partial\Omega_W$. We also have

$$(n_y)_E\, s_E = -2h(x_2 + x_4 h).$$

Hence, the changes

$$\begin{aligned} y_2 &\to -x_2, & y_4 &\to -x_4, \\ f &\to g, & (n_x)_E &\to (n_y)_E, \end{aligned}$$

applied to (2.4.25) and to the result of the changes (2.4.26) applied to (2.4.25), yield expressions for the other terms in (2.4.23). Substitution into (2.4.23), finally yields an expression for the collective contribution of the quadrature rule at $\partial\Omega_E$ and $\partial\Omega_W$ to the local truncation error. Similarly, for the contributions of $\partial\Omega_N$ and $\partial\Omega_S$, we apply the changes

$$\begin{aligned} x_2 &\to -x_1, & q_2 &\to q_1, \\ x_4 &\to -x_4, & (n_x)_E &\to (n_x)_N, \\ y_2 &\to -y_1, & (n_y)_E &\to (n_y)_N, \\ y_4 &\to -y_4, & (n_x)_W &\to (n_x)_S, \\ q_1 &\to q_2, & (n_y)_W &\to (n_y)_S. \end{aligned}$$

Then, combining results gives

$$\tau_q(q) = -\frac{h^2}{6J_0}\left[\left.\frac{\partial^2}{\partial q^\alpha \partial q^\beta}\right|_{q_0}\Big\{(y_4 f - x_4 g)(q_2^\alpha q_2^\beta - q_1^\alpha q_1^\beta) + (y_2 f - x_2 g)2q_2^\alpha q_4^\beta - (y_1 f - x_1 g)2q_1^\alpha q_4^\beta\Big\} + \left.\frac{\partial^3}{\partial q^\alpha \partial q^\beta \partial q^\gamma}\right|_{q_0}\Big\{((y_2 f - x_2 g)q_2^\gamma - (y_1 f - x_1 g)q_1^\gamma)\, q_1^\alpha q_2^\beta\Big\}\right] + \mathcal{O}(h^3).$$

Hence, the contribution of the quadrature rule to the local discretisation error is $\mathcal{O}(h^2)$.

Reconstruction. Here we study the role of the reconstruction step, in which the left and right states are computed. This gives us requirements which should be satisfied by the reconstruction, in order to obtain either a consistent or a weakly consistent discretisation. In the previous subsections we found that the local truncation error is limited to second-order by the choice of the mapping and by the choice of quadrature rule. Hence we are interested in first-order and second-order consistency only.

In our notations we will again drop indices if convenient. Assume we are interested in a local discretisation error, which is $\mathcal{O}(h^p)$, $p = 1, 2$. We consider the equation for a cell Ω on some level in the geometric structure. We assume that the solution of the problem is sufficiently smooth and that the numerical flux function is sufficiently differentiable.

The outward pointing unit normals on $\partial\Omega_k$ are given by

$$n_E = \begin{pmatrix} y_\eta \\ -x_\eta \end{pmatrix} \left.\frac{1}{\sqrt{x_\eta^2 + y_\eta^2}}\right|_{\partial\Omega_E}, \qquad n_S = \begin{pmatrix} y_\xi \\ -x_\xi \end{pmatrix} \left.\frac{1}{\sqrt{x_\xi^2 + y_\xi^2}}\right|_{\partial\Omega_S},$$
$$n_W = \begin{pmatrix} -y_\eta \\ x_\eta \end{pmatrix} \left.\frac{1}{\sqrt{x_\eta^2 + y_\eta^2}}\right|_{\partial\Omega_W}, \qquad n_N = \begin{pmatrix} -y_\xi \\ x_\xi \end{pmatrix} \left.\frac{1}{\sqrt{x_\xi^2 + y_\xi^2}}\right|_{\partial\Omega_N}.$$

We introduce the unit normal $\overline{n}_k$ in the physical space, on the kth cell face and $\overline{s}_k$, in absolute value equal to the length of $\partial\Omega_k$. The unit normals $\overline{n}_k$, $k \in D$ are shown in Fig. 2.4.1. The $\overline{n}_k$ and $\overline{s}_k$ are defined by

$$(2.4.27)\quad \overline{n}_k = \begin{cases} n_k, & k = E, N, \\ -n_k, & k = W, S, \end{cases} \qquad \overline{s}_k = \begin{cases} s_k, & k = E, N, \\ -s_k, & k = W, S. \end{cases}$$

Note that this gives $\overline{n}_k \overline{s}_k = n_k s_k$, $\forall k \in D$. We also introduce the vector $w \in X_a^2 \times E$, of length $2d + 2$, given by

$$w(q, q', n) = (q, q', \overline{n}),$$

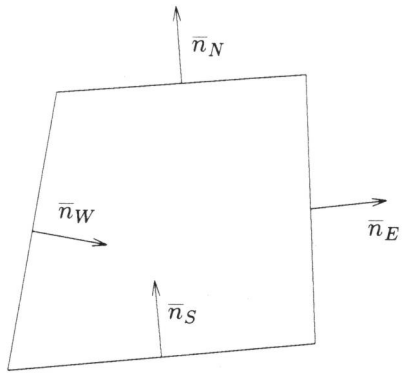


FIGURE 2.4.1. *Definition of* $\overline{n}_k$, $k \in D$.

where q, $q' \in X$ and $\overline{n} \in E$. For the kth cell face we define

$$w_k = (q_k, q_k, \overline{n}_k), \tag{2.4.28}$$

$$w_k^{LR} = (q_k^L, q_k^R, \overline{n}_k). \tag{2.4.29}$$

Assume that the following conditions hold

$$F(q, q', -n) = -F(q, q', n), \qquad \forall q, q' \in X_a, \quad n \in E, \tag{2.4.30}$$

$$F(q, q, n) = f^*(q^*), \qquad \forall q \in X_a, \quad n \in E. \tag{2.4.31}$$

The local discretisation error due to the reconstruction can now be written as

$$\tau_r = \frac{1}{A} \sum_{k \in D} \left(F(q_k^L, q_k^R, n_k) - f^*(q_k^*) \right) s_k. \tag{2.4.32}$$

With (2.4.28) and (2.4.30) we now have

$$F(w_k) = \begin{cases} f^*(q_k^*), & k = E, N, \\ -f^*(q_k^*), & k = W, S, \end{cases}$$

and with (2.4.27), (2.4.32) can be written as

$$\tau_r = \frac{1}{A} \sum_{k \in D} \left(F(w_k^{LR}) \bar{s}_k - f^*(q_k^*) s_k \right). \tag{2.4.33}$$

We denote the *reconstruction error* r_k, for the kth cell face by

$$r_k = w_k^{LR} - w_k. \tag{2.4.34a}$$

We denote the difference between $(w_0)_k$ and the mean w_k on $\partial\Omega_k$ by

$$\Delta_k = w_k - (w_0)_k, \tag{2.4.34b}$$

where $(w_0)_k$ is defined by

$$(w_0)_k = (q_0, q_0, (\overline{n}_0)_k),$$

and $(\overline{n}_0)_k$ defined by

$$(\overline{n}_0)_k = \begin{cases} \begin{pmatrix} y_2 \\ -x_2 \end{pmatrix} \dfrac{1}{\sqrt{x_2^2 + y_2^2}}, & k = E, W, \\ \begin{pmatrix} -y_1 \\ x_1 \end{pmatrix} \dfrac{1}{\sqrt{x_1^2 + y_1^2}}, & k = N, S. \end{cases}$$

Now, first making a Taylor series expansion of $F(w)$ around the mean w_k along cell face $\partial\Omega_k$ and substitution of it in (2.4.33) and next, substitution of an expansion of $F(w)$ around $(w_0)_k$, gives

(2.4.35)

$$\tau_r = \frac{1}{A} \sum_{k \in D} \left\{ \left. \frac{\partial F}{\partial w^\alpha} \right|_{(w_0)_k} r_k^\alpha + \left. \frac{\partial^2 F}{\partial w^\alpha \partial w^\beta} \right|_{(w_0)_k} r_k^\alpha (\Delta_k^\beta + \frac{1}{2} r_k^\beta) \right.$$
$$\left. + \frac{1}{2} \left. \frac{\partial^3 F}{\partial w^\alpha \partial w^\beta \partial w^\gamma} \right|_{(w_0)_k} r_k^\alpha (\Delta_k^\beta \Delta_k^\gamma + r_k^\beta \Delta_k^\gamma + \frac{1}{3} r_k^\beta r_k^\gamma + \dots) + \dots \right\} \bar{s}_k.$$

Here again the summation convention is used for α, β and γ. We use (2.4.35) to study the requirements for consistency.

2.4.6. Consistency requirements. Now we can formulate the requirements to be satisfied in the reconstruction phase in order to obtain a pth-order consistent or weakly consistent discretisation.

Consistency. We consider the contributions due to the cell faces $\partial\Omega_k$, $k = E, W$. Assume that the following asymptotic relations hold for $p, q, s = 1, 2$, and for r_k and δ_k as defined by (2.4.34):

(2.4.36a) $$r_E = \mathcal{O}(h^q),$$

(2.4.36b) $$r_W = r_E + \mathcal{O}(h^r), \quad r \geq q,$$

(2.4.36c) $$\Delta_E = \mathcal{O}(h^s),$$

(2.4.36d) $$\Delta_W = \Delta_E + \mathcal{O}(h^t), \quad t \geq s.$$

For the mapping M we also have

(2.4.37a) $$s_E = \mathcal{O}(h),$$

(2.4.37b) $$s_W = s_E + \mathcal{O}(h^2).$$

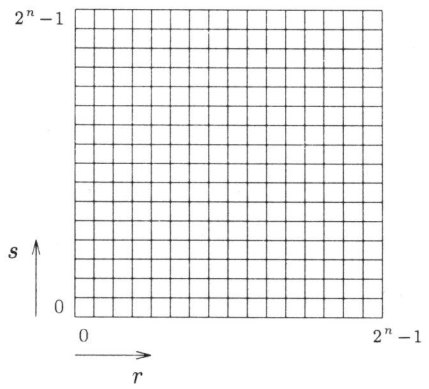


FIGURE 2.4.2. *An n times locally refined cell and indices used on the fine level.*

With (2.4.36) and (2.4.37), and by changing the order of summation, we find for the first term of (2.4.35),

$$\begin{aligned} r_E\,\bar{s}_E + r_W\,\bar{s}_W &= r_E\,s_E + (r_E + \mathcal{O}(h^r))(-\bar{s}_E + \mathcal{O}(h^2)) \\ &= \mathcal{O}(h^{q+2}) + \mathcal{O}(h^{r+1}). \end{aligned}$$

Hence, a pth-order consistent discretisation requires

$$q \geq p, \tag{2.4.38a}$$

$$r - 1 \geq p, \tag{2.4.38b}$$

It can easily be shown that all other terms in (2.4.35) give $\mathcal{O}(h^p)$ or smaller contributions to τ_r, provided $p, q, s \geq 1$ and (2.4.38) are satisfied.

Weak consistency. For weak consistency and its requirements with respect to reconstruction, we consider a cell $\Omega^l_{i,j}$ and all its descendants when the composite grid is n times refined. The definitions of the previous subsections hold, but a superscript specifying the level and a subscript for the cell number are added. The superscript n denotes level $l + n$. As shown in Fig. 2.4.2, subscript r denotes cell index $2^n i + r$ and subscript s denotes index $2^n j + s$. The collective local discretisation error is now given by

$$\{T^l_n(q)\}^l_{i,j} = \frac{1}{A^l_{i,j}} \sum_{r,s=0}^{2^n-1} A^n_{r,s}\,\tau^n_{r,s}(q). \tag{2.4.39}$$

With $\tau^l(q) = \tau^l_q(q) + \tau^l_r(q)$, we have

$$A^n_{r,s}\,\tau^n_{r,s}(q) = \sum_{k \in D} \left(F((w^{LR})^n_{r,s,k})\bar{s}^n_{r,s,k} - f^{*n}_{r,s,k}(q) s^n_{r,s,k} \right). \tag{2.4.40}$$

Assume the numerical flux $F((w^{LR})^n_{r,s,k})$ across cell face $\partial\Omega^n_{r,s,k}$ to be an $\mathcal{O}(h^q_{l+n})$ accurate approximation of the mean flux $f^{*n}_{r,s,k}(q)$. We have

$$F((w^{LR})^n_{r,s,k})\bar{s}^n_{r,s,k} = f^{*n}_{r,s,k}(q)s^n_{r,s,k} + \mathcal{O}(h^{q+1}_{l+n}). \tag{2.4.41}$$

Substitution of (2.4.41) and (2.4.40) into (2.4.39) yields a summation over all cell faces on the 'outer' boundary $\partial\Omega^n_{r,s,k} \subset \partial\Omega^l_{i,j,k}$, $\forall k \in D$. We consider $k = W$ and $n \to \infty$. For the contribution of the approximations on these cell faces to $\{T^l_n\}^l_{i,j}$ we find with (2.4.41)

$$\begin{aligned}
&\frac{1}{A^l_{i,j}} \sum_{s=0}^{2^n-1} \left(F((w^{LR})^n_{0,s,W})\bar{s}^n_{0,s,W} - f^{*n}_{0,s,W}(q)s^n_{0,s,W}\right) \\
&\quad = \frac{1}{A^l_{i,j}} \sum_{s=0}^{2^n-1} \left(C_1 h^{q+1}_{l+n} + \mathcal{O}(h^{q+2}_{l+n})\right) \\
&\quad = C_2 h^q_{l+n} + \mathcal{O}(h^{q+1}_{l+n}),
\end{aligned}$$

where C_1 and C_2 are constants independent of the level n. Similar results hold for $k = N, E, S$. Hence, if we have an $\mathcal{O}(h^q_{l+n})$ accurate approximation of the mean value along a cell face of the flux across the cell face, then the collective local discretisation error is $\mathcal{O}(h^q_{l+n})$ and hence, the approximation is qth-order weakly consistent. So, a weakly consistent approximation of order $p = 1, 2$, is obtained by pth-order accurate approximation of the mean flux. Let the reconstruction error be defined by (2.4.34). By the fact that the quadrature rule yields a second-order contribution, and by expansion of $F(w)$ around the mean $w^{l+n}_{r,s,k}$, it follows that a reconstruction error of order $p = 1, 2$, yields a pth-order weakly consistent discretisation.

Note that consistency of order $p = 1, 2$, implies weak consistency, since a necessary but not sufficient condition for consistency is a reconstruction error of order p, as given by (2.4.38a).

2.5. Numerical results

In this section we present numerical results of local discretisation errors and global discretisation errors on a locally refined grid. We consider consistent and weakly consistent discretisations. First we define the reconstruction and virtual states in order to obtain consistency or weak consistency. Next for a problem with a known solution we present results obtained on a composite grid, showing the relation between consistency and global error. Then we elaborate on this for the set of Euler equations with smooth solution, where one component of the solution is known.

2.5.1. Virtual states. We introduced the concepts of consistency and weak consistency. Here we describe virtual states to obtain discretisations which are first-order or second-order consistent in the weak or usual sense. Their analysis is deferred to Chap. 4 of this thesis.

Weak consistency. For a weakly consistent first-order or second-order discretisation we require a reconstruction of first-order or second-order accuracy respectively. This is simply achieved by first-order or second-order accurate computation of virtual states. First-order weak consistency for the algebraic equations defined for $\Omega^{l+1}_{2i+1,2j+1}$, where $\partial\Omega^{l+1}_{2i+1,2j+1,E} \subset \partial\Omega^{l}_{g}$, is attained by using a virtual state defined by

$$v^{l+1}_{2i+2,2j+1} = q^{l}_{i+1,j} \tag{2.5.1}$$

Second-order weak consistency is attained for the same situation by the virtual states defined as

$$v^{l+1}_{2i+2,2j+1} = \frac{3}{4} q^{l+1}_{i+1,j} + \frac{1}{4} q^{l+1}_{i,j+1}, \tag{2.5.2a}$$

$$v^{l+1}_{2i+3,2j+1} = \frac{3}{4} q^{l+1}_{i+1,j} + \frac{1}{4} q^{l+1}_{i+2,j+1}. \tag{2.5.2b}$$

For other virtual states we define similar formulae.

Consistency. Consistency of order p in the usual sense requires in addition a difference in reconstruction error of two opposite cell faces of order $p+1$. Again, consider the equations for $\Omega^{l+1}_{2i+1,2j+1}$, and $\partial\Omega^{l+1}_{2i+1,2j+1,E} \subset \partial\Omega^{l}_{g}$. First-order consistency, is attained by the virtual state

$$v^{l+1}_{2i+2,2j+1} = \frac{3}{4} q^{l+1}_{i+1,j} + \frac{1}{4} q^{l+1}_{i,j+1}. \tag{2.5.3}$$

Compare this to (2.5.2a), for *second-order weak* consistency. Second-order consistency is attained for the same situation by defining the virtual states

$$\begin{aligned} v^{l+1}_{2i+2,2j+1} = {} & \frac{17}{16} q^{l}_{i+1,j} + \frac{1}{16} \left(q^{l}_{i,j} + q^{l}_{i,j+1} + q^{l}_{i+1,j+1} \right) \\ & - \frac{2}{16} \left(q^{l}_{i+2,j} + q^{l}_{i+1,j-1} \right), \end{aligned} \tag{2.5.4a}$$

$$\begin{aligned} v^{l+1}_{2i+3,2j+1} = {} & \frac{17}{16} q^{l}_{i+1,j} + \frac{1}{16} \left(q^{l}_{i+2,j} + q^{l}_{i+2,j+1} + q^{l}_{i+1,j+1} \right) \\ & - \frac{2}{16} \left(q^{l}_{i,j} + q^{l}_{i+1,j-1} \right). \end{aligned} \tag{2.5.4b}$$

2.5.2. Discretisation errors for a nonlinear test problem. Now we apply the finite volume discretisations described so far to a model problem. For the model equations we are able to study in detail both local and global discretisation errors, on a uniform and a locally refined grid. The problem is of the form (2.1.1a) defined on $\Omega \subset \mathbb{R}^2$, with q, $f(q)$ and $g(q)$ given by

$$q = \begin{pmatrix} u \\ v \end{pmatrix}, \quad f(q) = \begin{pmatrix} u^2 \\ uv \end{pmatrix}, \quad g(q) = \begin{pmatrix} uv \\ v^2 \end{pmatrix}.$$

We consider the homogeneous case, $s(x, y) = 0$. These nonlinear equations have solutions that are constant in the characteristic direction v/u, when v/u itself is constant. For any boundary condition which has constant v/u, the solution is known in closed form.

The numerical flux function $F(q^L, q^R, n)$ used, is the upwind flux given by

$$F(q^L, q^R, n) = \begin{cases} T^{-1}(n) f(T(n) q^L), & \text{if } (n \cdot q^L), (n \cdot q^R) > 0, \\ T^{-1}(n) f(T(n) q^R), & \text{if } (n \cdot q^L), n(\cdot q^R) < 0, \\ 0, & \text{if } (n \cdot q^L)(n \cdot q^R) \leq 0, \end{cases}$$

where $T(n)$ is a *rotation matrix*, given by

$$T(n) = \begin{pmatrix} n_x & n_y \\ -n_y & n_x \end{pmatrix}.$$

We consider a smooth problem on the unit square with exact solution

$$u = \widetilde{u}(t(x, y)),$$
$$v = u \tan \varphi,$$

where $t = -x \sin \varphi + y \cos \varphi$ is a coordinate perpendicular to the characteristic direction and φ is a constant angle of a characteristic direction with respect to the positive x direction.

We study the discrete maximum norm of the αth component $\|(.)^\alpha\|_{\infty, X_c(\widetilde{\Omega}_c)}$ on the composite grid, of the local discretisation error $\tau_c(\bar{q})$, and of the error of the solution. On the composite grid the solution error is given by

$$\epsilon_c(x, y) = q^l_{i,j} - \{\overline{R}^l \bar{q}\}^l_{i,j}, \quad (x, y) \in \Omega^l_{i,j}, \quad \forall (i, j, l) \in I_c.$$

Here, $\bar{q}$ is the exact solution of the continuous problem. We take $\widetilde{u}(t) = 1 + \cos(\sqrt{2}\,\pi t)$, $\varphi = 45°$. The exact solution for this problem is shown in Fig. 2.5.1, and is given by

$$u = v = 1 + \cos(\pi(-x + y)).$$

The grid subdivides the domain on the finest level L in equal parts with size h_L in the direction of the x axis and $\frac{1}{2}h_L$ in the direction of the y axis. The discretisation errors for uniform grids are given in Tab. 2.5.1, both for the first-order discretisation ($p = 1$) and for the κ-scheme with $\kappa = 0$ ($p = 2$). The solution and error components are the same. Both the discretisation

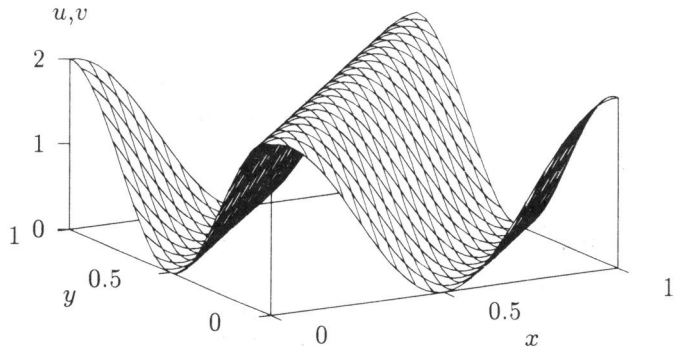


FIGURE 2.5.1. *Exact solution of the model problem.*

TABLE 2.5.1. *Discretisation error for the model problem on a uniform grid.*

	$p = 1$		$p = 2,\ \kappa = 0$	
h_L	$\|\tau_c(q)\|_{\infty,X_c(\widetilde{\Omega}_c)}$	$\|\epsilon_c\|_{\infty,X_c(\widetilde{\Omega}_c)}$	$\|\tau_c(q)\|_{\infty,X_c(\widetilde{\Omega}_c)}$	$\|\epsilon_c\|_{\infty,X_c(\widetilde{\Omega}_c)}$
$\frac{1}{4}$	0.6514e + 01	0.7214e + 00	0.7220e + 00	0.1488e + 00
$\frac{1}{8}$	0.3584e + 01	0.4937e + 00	0.1197e + 00	0.3162e − 01
$\frac{1}{16}$	0.1836e + 01	0.3090e + 00	0.2270e − 01	0.6067e − 02
$\frac{1}{32}$	0.9234e + 00	0.1800e + 00	0.5109e − 02	0.1380e − 02
$\frac{1}{64}$	0.4624e + 00	0.9924e − 01	0.1241e − 02	0.3376e − 03
$\frac{1}{128}$	0.2313e + 00	0.5270e − 01	0.3080e − 03	0.8429e − 04

and the solution of the discrete problem tend to pth-order convergence with decreasing mesh width, as predicted.

We now introduce a rather arbitrarily chosen locally refined grid by refining all cells to the right of $x = 0.5$. This gives a composite grid as shown in Fig. 2.5.2. On this composite grid we perform the same test with the first-order ($p = 1$) and second-order ($\kappa = 0$, $p = 2$) weakly consistent discretisation of the equations involving a green boundary. The maximum local truncation error and solution error are given in Tab. 2.5.2. This table shows that the local truncation error is zeroth-order (for $p = 1$) and first-order (for $p = 2$), although they are first-order and second-order weakly consistent respectively. The maximum local discretisation errors occur for the equations involving a green boundary. For the first-order accurate case the limiting value for $h_L \to 0$ of the maximum local truncation error can be easily computed and is

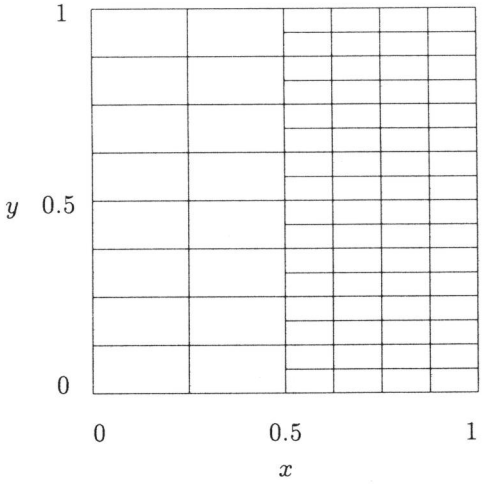


FIGURE 2.5.2. *Non-uniform composite grid for the model problem;* $h_L = \frac{1}{8}$

TABLE 2.5.2. *Discretisation error for the model problem on locally refined grid. Only weak consistency of the specified order for equations along green boundaries.*

	$p = 1$		$p = 2,\ \kappa = 0$	
h_L	$\|\tau_c(q)\|_{\infty,X_c(\widetilde{\Omega}_c)}$	$\|\epsilon_c\|_{\infty,X_c(\widetilde{\Omega}_c)}$	$\|\tau_c(q)\|_{\infty,X_c(\widetilde{\Omega}_c)}$	$\|\epsilon_c\|_{\infty,X_c(\widetilde{\Omega}_c)}$
$\frac{1}{8}$	0.9086e + 01	0.6720e + 00	0.5484e + 01	0.2575e + 00
$\frac{1}{16}$	0.7089e + 01	0.4213e + 00	0.2882e + 01	0.6101e − 01
$\frac{1}{32}$	0.6400e + 01	0.2492e + 00	0.1459e + 01	0.1342e − 01
$\frac{1}{64}$	0.6211e + 01	0.1406e + 00	0.7317e + 00	0.3043e − 02
$\frac{1}{128}$	0.6152e + 01	0.7608e − 01	0.3662e + 00	0.7176e − 03
$\frac{1}{256}$	0.6134e + 01	0.3999e − 01	0.1831e + 00	0.1738e − 03

TABLE 2.5.3. *Discretisation error for the model problem on locally refined grid. Consistency of the specified order for all equations.*

	$p = 1$		$p = 2,\ \kappa = 0$	
h_L	$\|\tau_c(q)\|_{\infty,X_c(\widetilde{\Omega}_c)}$	$\|\epsilon_c\|_{\infty,X_c(\widetilde{\Omega}_c)}$	$\|\tau(q)_c\|_{\infty,X_c(\widetilde{\Omega}_c)}$	$\|\epsilon_c\|_{\infty,X_c(\widetilde{\Omega}_c)}$
$\frac{1}{8}$	0.7916e + 01	0.6954e + 00	0.1295e + 01	0.1352e + 00
$\frac{1}{16}$	0.4224e + 01	0.4305e + 00	0.3079e + 00	0.2532e − 01
$\frac{1}{32}$	0.2147e + 01	0.2522e + 00	0.7402e − 01	0.5098e − 02
$\frac{1}{64}$	0.1078e + 01	0.1411e + 00	0.1802e − 01	0.1177e − 02
$\frac{1}{128}$	0.5396e + 00	0.7604e − 01	0.4444e − 02	0.2910e − 03
$\frac{1}{256}$	0.2699e + 00	0.3991e − 01	0.1103e − 02	0.7421e − 04

$9/8\pi\sqrt{3} = 6.12157\ldots$. The global error is first-order ($p = 1$) and second-order ($p = 2$) respectively. Note that the error in the second-order accurate solution computed with the non-uniform composite grid (Tab. 2.5.2) is larger than the error of the second-order accurate solution computed on a uniform grid with the same mesh size as the *coarse* part of the composite grid (Tab. 2.5.1). Apparently, the local discretisation errors for the equations on the locally refined grid are so large that the solution of the discrete problem is worse than the solution obtained without refinements. The first-order solution is less sensitive to such large local discretisation errors.

An improvement of this discretisation is obtained by the equations which are all consistent in the usual sense. This discretisation is also tested on locally refined grids as shown in Fig. 2.5.2. Discretisation errors are given in Tab. 2.5.3. This table shows that the local discretisation error is of the specified order. The maximum error of the solution for both first-order and second-order discretisation is smaller than on the uniform grid with the coarse mesh size. We find that both for a pth-order consistent and pth-order weakly consistent discretisation the order of discrete convergence in maximum norm is p. The consistent discretisation appears to give more accurate results for $p = 2$. Applying the weakly consistent discretisation, one has to be careful about the errors introduced by the relatively large local truncation errors of equations involving a green boundary. These errors may spoil the accuracy in infinity norm of the resulting numerical solution.

2.5.3. Euler equations with smooth solution. Next we show the results of a numerical experiment, for the steady Euler equations. The experiment is made to study the discretisation error for a smooth problem. Better than the model problem, this problem may represent problems encountered in practice. Again, the discretisations are consistent or weakly consistent of order $p = 1, 2$. The flow is described by the Euler equations, given by (1.2.2)–

(1.2.4). The unknown vector function $q \in X(\overline{\Omega})$ is

$$q = (c, u, v, z)^T,$$

where c is the speed of sound, u and v are Cartesian velocity components in x and y direction respectively, and z represents the physical entropy.

The numerical flux is Osher's approximate Riemann solver [14] in the so-called P-variant as introduced in [10]. It is written as

$$F(q^L, q^R, n) = T^{-1}(n)\widetilde{F}(T(n)q^L, T(n)q^R),$$

where $\widetilde{F}$ is the approximate Riemann solver and $T(n)$ is a rotation matrix, defined by

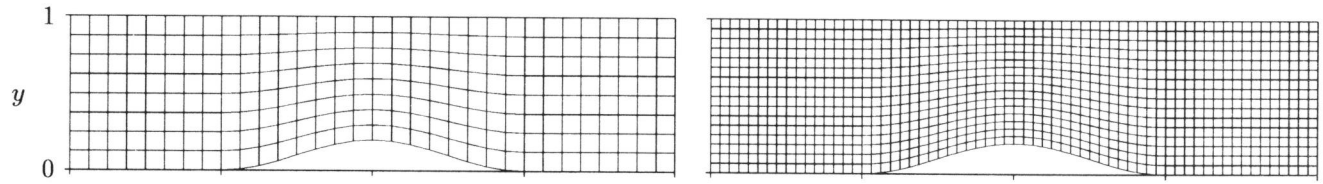
$$T(n) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & n_x & n_y & 0 \\ 0 & -n_y & n_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

We consider the subsonic flow through a channel with a 'sine-form' bump, defined by the shape $y_w(x)$ of the lower wall of the channel, given by

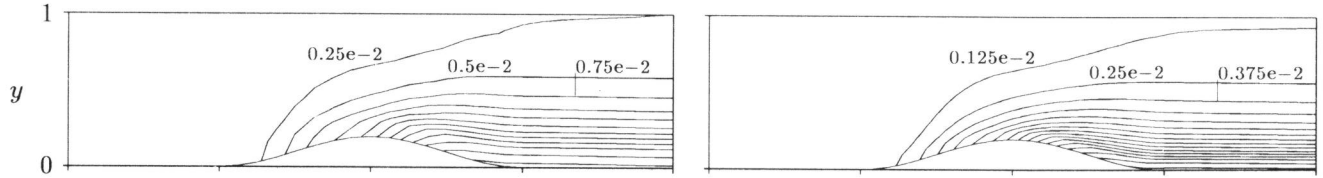
$$y_w(x) = \begin{cases} 0, & x \leq 1, \quad x \geq 3, \\ 0.1(1 - \cos((x-1)\pi)), & 1 < x < 3. \end{cases}$$

The upper wall is a straight line at $y = 1$. In the direction of the x axis, the computational domain extends from $x = 0$ to $x = 4$. The boundary conditions are 'overspecified' [9]. At $x = 0$ the Mach number is $M = 0.5$, the density $\rho = 1$ and the velocity $u = 1$, $v = 0$. On the upper and lower wall, impermeability is required, $un_x + vn_y = 0$ and at the outlet boundary, $x = 4$, the boundary conditions are specified to have the same values as the inlet conditions. For this problem the exact solution has constant entropy, i.e., constant $z = p\rho^{-\gamma}$. In Fig. 2.5.3–2.5.5 we show iso-line plots of the error in the entropy.

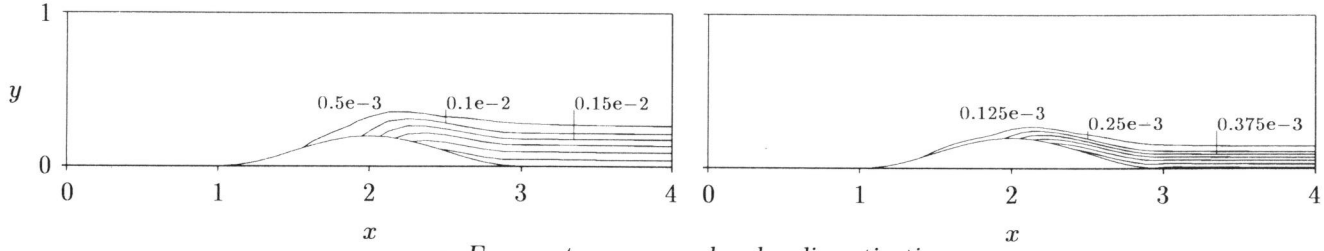
In Fig. 2.5.3a the grids used are shown. Here uniform grids are used. In Fig. 2.5.3b–c the entropy error is shown for first-order and second-order consistency respectively. The choice of iso-line values is such that for a first-order and second-order error, the pictures for the two grids should be the same asymptotically. In Fig. 2.5.4 similar results are shown for a locally refined grid and a weakly consistent discretisation for the equations involving a green boundary. Fig. 2.5.4b is for first-order weak consistency and in Fig. 2.5.4c results are shown for second-order weak consistency. Although the local discretisation error in a cell bordering a green boundary is $\mathcal{O}(h_l^{p-1})$, $p = 1, 2$, the global error is found to be first-order and second-order respectively. Finally, in Fig. 2.5.5 results are shown for consistent discretisations on the locally refined grid.



a. *Uniform grids.*

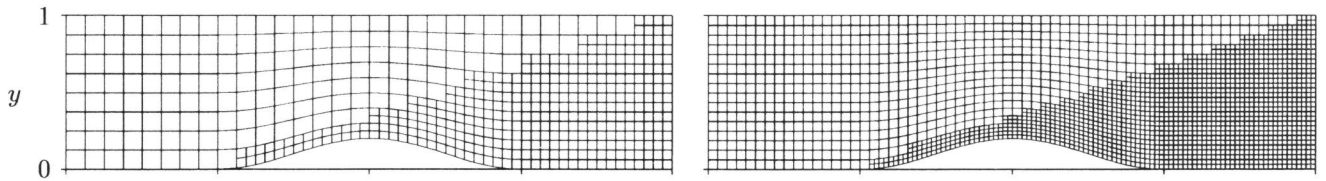


b. *Error entropy, first-order discretisation.*

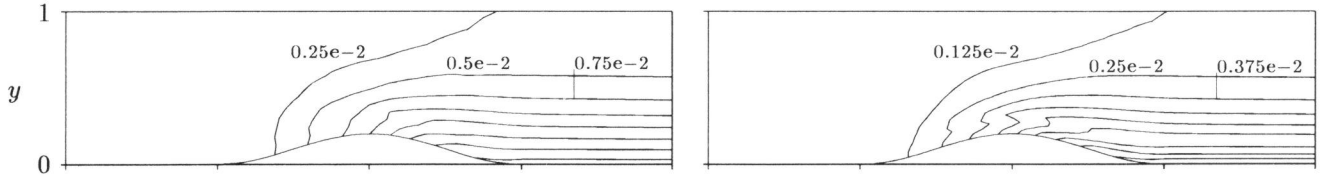


c. *Error entropy, second-order discretisation.*

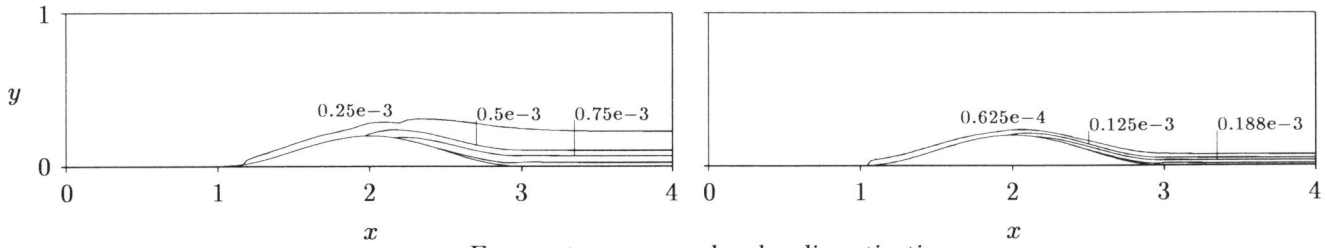
FIGURE 2.5.3. *Entropy error for the Euler equations with a smooth solution; uniform grids.*



a. *Locally refined grids.*

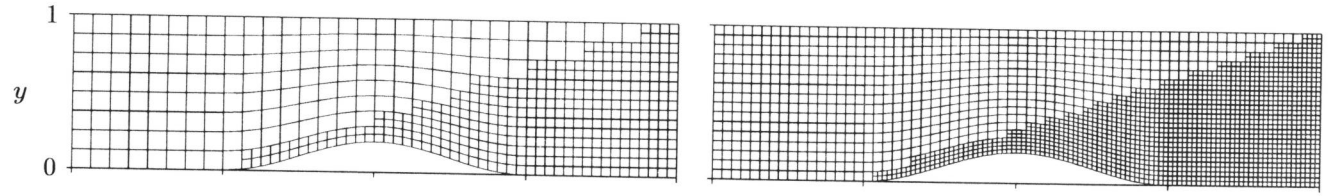


b. *Error entropy, first-order discretisation.*

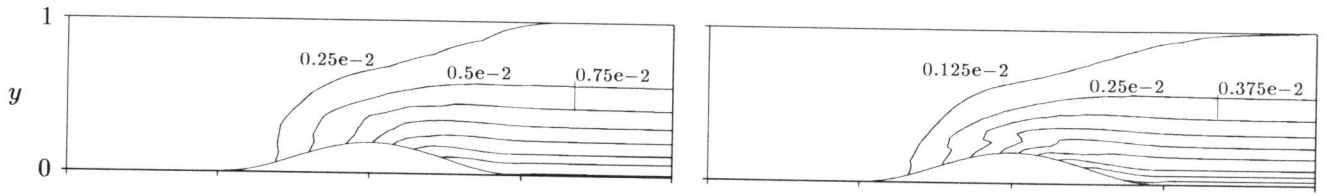


c. *Error entropy, second-order discretisation.*

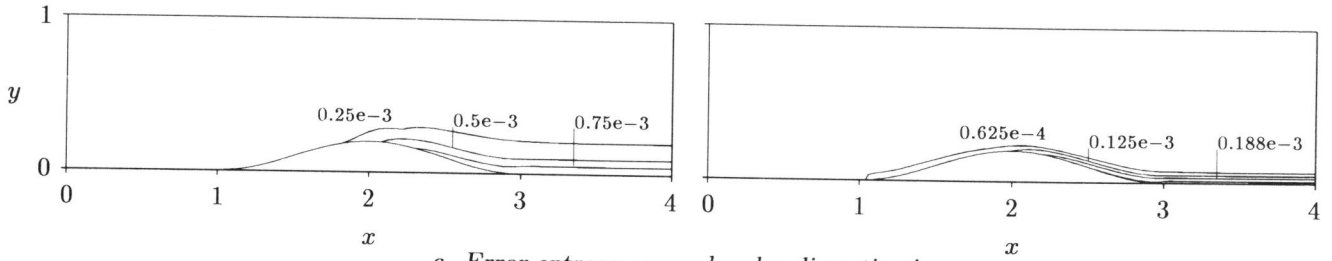
FIGURE 2.5.4. *Entropy error for the Euler equations with a smooth solution; locally refined grid and weak consistency of order* $p = 1, 2$.



a. *Locally refined grids.*



b. *Error entropy, first-order discretisation.*



c. *Error entropy, second-order discretisation.*

FIGURE 2.5.5. *Entropy error for the Euler equations with a smooth solution; locally refined grid and consistency of order* $p = 1, 2$. *Notice the large similarity with Fig. 2.5.4.*

In the results shown for the Euler equations the difference between the consistent and weakly consistent discretisations are very small.

2.6. Concluding remarks

We introduced a finite volume discretisation for a system of steady conservation laws, using a regular, non-uniform, locally refined grid. Analysis of the local discretisation error shows that the approximation of the curvilinear cells by quadrilateral cells yields at most a second-order contribution to the local error. This error is only essential for equations derived for boundary cells.

The restriction which transfers a fine-grid function to the refined part of the coarse grid, is shown to introduce an error of at most $\mathcal{O}(h_l^2)$. Hence, on the coarse grid at most second-order accuracy is obtained.

We studied the contributions of the various steps of the discretisation to the local discretisation error. The quadrature rule, used to approximate the mean flux across a cell face, a mid-point rule, yields an $\mathcal{O}(h_l^2)$ contribution to the local error. Besides the usual consistency, we introduced a new, weak form of consistency. We found requirements to be satisfied by the reconstruction phase of the discretisation process, to obtain a first-order or a second-order, consistent or weakly consistent discretisation. A weakly consistent discretisation of order $p = 1, 2$ is obtained by pth-order accurate reconstruction of the mean flux across a cell face. For the mid-point quadrature rule this requires pth-order accuracy in the reconstruction phase of the discretisation. Consistency of order p is obtained if, in addition the pth-order errors of the reconstruction for two opposite cell faces have a difference of order $p + 1$. Consistency implies weak consistency.

Numerical results show that the first-order and second-order, weakly consistent discretisations yield a first-order and second-order global error respectively. However, the second-order discretisation may be quite sensitive to the large local discretisation errors for the equations involving a green boundary. Application of the weakly consistent discretisation for green boundaries is not recommended, especially not for the second-order discretisation. The first-order weakly consistent discretisation appears to be less sensitive (almost insensitive) to these relatively large local discretisation errors (i.e., inconsistency errors). This is shown in Fig. 2.5.4 and 2.5.5.

Experiments for a model problem as well as for the Euler equations show that the error bounds which are theoretically derived, agree with the errors found in practice.

References

[1] R.E. Bank, *The efficient implementation of local mesh refinement algorithms*, Adaptive Computational Methods for Partial Differential Equations (Baltimore, 1983) (I. Babuška, J. Chandra and J.E. Flaherty, eds.), University of Maryland, Society for Industrial and Applied Mathematics, Philadelphia, 1984, pp. 74–81.

[2] T.J. Barth and P.O. Frederickson, *Higher order solution of the Euler equations on unstructered grids, using quadratic reconstruction*, AIAA-90-0013, 1990.

[3] T.J. Barth and D.C. Jespersen, *The design and application of upwind schemes on unstructured meshes*, AIAA-89-0366, 1989.

[4] A. Brandt, *Multi-level adaptive techniques (MLAT) for singular perturbation-problems*, Numerical Analysis of Singular Perturbation Problems (P.W. Hemker and J.J.H. Miller, eds.), Academic Press, 1979, pp. 53–142.

[5] S.K. Godunov, *Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics*, Math. Sbornik **47** (1959), 271–306, Translated from Russian: Cornell Aeronautical Laboratory.

[6] W. Hackbusch and U. Trottenberg (eds.), *Multigrid Methods* II, Proc. of the 2nd European Conference on Multigrid Methods, held in Cologne, 1985, Lecture Notes in Mathematics, vol. 1228, Springer-Verlag, 1986.

[7] A. Harten and P.D. Lax, *A random choice finite-difference scheme for hyperbolic conservation laws*, SIAM J. Numer. Anal. **18** (1981), 289–315.

[8] A Harten, P.D. Lax and B. van Leer, *On upstream differencing and Godunov-type schemes for hyperbolic conservation laws*, SIAM Rev. **25** (1983), 35–61.

[9] P.W. Hemker, *Defect correction and higher order schemes for the multi grid solution of the steady Euler equations*, In Hackbusch and Trottenberg [6], pp. 149–165.

[10] P.W. Hemker and S.P. Spekreijse, *Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations*, Appl. Num. Math. **2** (1986), 475–493.

[11] C. Hirsch, *Numerical Computation of Internal and External Flows*, vol. 2, John Wiley and Sons, Ltd., Chichester, 1990.

[12] B. Koren, *Defect correction and multigrid for the efficient and accurate computation of airfoil flows*, J. Comput. Phys. **77** (1988), 183–206.

[13] P.D. Lax, *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, Regional Conference Series in Applied Mathematics, vol. 11, Society for Industrial and Applied Mathematics, Philadelphia, 1973.

[14] S. Osher and F. Solomon, *Upwind difference schemes for hyperbolic systems of conservation laws*, Math. Comp. **38** (1982), no. 158, 339–374.

[15] P.L. Roe, *Approximate Riemann solvers, parameter vectors and difference schemes*, J. Comput. Phys. **43** (1981), 357–372.

[16] G.H. Schmidt and F.J. Jacobs, *Adaptive local grid refinement and multi-grid in numerical reservoir simulation*, J. Comput. Phys. **77** (1988), 140–165.

[17] S.P. Spekreijse, *Multigrid solution of monotone second-order discretizations of hyperbolic conservation laws*, Math. Comp. **49** (1986), no. 179, 135–155.

[18] ———, *Second order accurate upwind solutions of the 2D steady Euler equations by the use of a defect correction method*, In Hackbusch and Trottenberg [6], pp. 285–300.

[19] ———, *Multigrid Solution of the Steady Euler Equations*, CWI, Amsterdam, 1988, CWI-tract 46.

[20] J.L. Steger and R.F. Warming, *Flux vector splitting in the inviscid gasdynamic equations with application to finite-difference methods*, J. Comput. Phys. **40** (1981), 263–293.

[21] P.K. Sweby, *High resolution schemes using flux limiters for hyperbolic conservation laws*, SIAM J. Numer. Anal. **21** (1984), 995–1011.

[22] B. van Leer, *Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection*, J. Comput. Phys. **23** (1977), 276–299.

[23] ———, *Flux-vector splitting for the Euler equations*, Proceedings of the Eight International Conference on Numerical Methods in Fluid Dynamics, Aachen, 1982, Lecture Notes in Physics, vol. 170 (E. Krause, ed.), Springer-Verlag, 1982, pp. 507–512.

[24] ———, *On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher and Roe*, SIAM J. Sci. Statist. Comput. **5** (1984), 1–20.

[25] ———, *Upwind difference methods for aerodynamic problems governed by the Euler equations*, Large Scale Computations, Lectures in Applied Mathematics part 2 (B.E. Engquist, S.J. Osher and R.C.J. Sommerville, eds.), AMS, 1985, pp. 327–336.

[26] ———, *Towards the ultimate conservative difference scheme. V. A second order sequel to Godunov's method.*, J. Comput. Phys. **32** (79), 101–136.

CHAPTER 3

Application of the solution-adaptive multigrid method to the Euler equations

3.1. Introduction

In this section the discretisation introduced in Chap. 2 is applied to the Euler equations of fluid dynamics. The discretisations on the various levels of refinement introduce a set of nonlinear algebraic equations. The method to solve the set of nonlinear algebraic equations is an application of the nonlinear multigrid scheme, called *full approximation storage* (FAS), possibly embedded in the *full multigrid* (FMG) algorithm or an *iterative defect correction* (IDeC) process.

First, these algorithms are described for their use in the present context of locally refined grids. After that, a strategy to introduce local refinements is described. In the experiments to follow, this strategy is used in the grid-refinement cycles. Next, some aspects of the implementation of a local refinement, multigrid method in a computer code are discussed. Finally, a number of examples are presented. The problems chosen for the experiments give an example of the possible range of application, where this method may be used as a tool for analysis of fluid dynamics problems. The problems considered are:

- shock reflection on a flat surface;
- transonic flow around an airfoil;
- spurious entropy in the subsonic flow along a compression corner;
- shock wave along a continuously curved, convex surface.

First we consider two standard test cases from numerical fluid dynamics to validate the method and to get an idea of possible gain in efficiency of the local grid refinement method with respect to the uniform grid cases. Then, a problem is considered where the method is used to locally introduce a singular grid, in order to approximately solve a problem which has a singular solution, with sufficient accuracy. Finally, the method is used as a tool to study properties of the solution of the Euler equations in detail.

In this chapter we also give CPU execution times for the specific implementation of the solution-adaptive, local grid refinement code, run on a typical

present-day workstation. We compare these execution times with the execution times for an implementation for uniform grids only, that uses the same multigrid and defect correction algorithms as the adaptive code (cf. 3.5).

Refinement criteria used in all of these experiments are solely based on requirements for the grid in order to provide 'sufficient' resolution for the solution. However, sufficient resolution for the solution does not necessarily imply sufficiently small errors for the discrete approximation of the equations. If one is only interested in the components of the *solution* itself of some problem and not in any of its derivatives, then sufficient resolution depends solely on first-order derivatives of solution components. Apart from first-order derivatives, the local discretisation error usually also depends on higher-order derivatives. Therefore, using only gradients of solution components in the refinement criterion may not be sufficient. The subject of local discretisation errors and their a-posteriori estimation are considered in Chap. 4 of this thesis.

3.2. Multigrid and defect correction

3.2.1. Introductory remarks. The set of algebraic equations obtained by the discretisation introduced in Chap. 2 is solved by point Gauss-Seidel relaxation, with multigrid convergence acceleration. This particular multigrid procedure is an application of the nonlinear multigrid scheme, called full approximation storage [3]. For the second-order discretisation this process is embedded in an iterative defect correction process [1], [10]. The implementation of the multigrid scheme is directly based on the methods described in [15], [16], [31], [32] and [33], extensively applied in [33], [18] and [20]. Iterative defect correction is described in [1] and [10] and applied in [13], [32], [33], [18] and [20]. The basic method inside the iterative defect correction method, which is used to (approximately) invert the inaccurate discrete operator, is the nonlinear multigrid method.

In this section we give a brief description of the methods, and the slight modifications to our application. The description is a summary of the description presented in [36].

3.2.2. A locally nested sequence of discretisations. In order to use multigrid we have to specify grid transfer operators. The restriction operators $\overline{R}^l_{l+1}$ and R^l_{l+1} and the prolongation operator P^{l+1}_l are defined such that (*i*) a sequence of locally nested discretisations on the sequence of locally refined grids is obtained and such that (*ii*) the coarse-grid equations (2.3.14) are satisfied implicitly. A locally nested sequence of discretisations $\{N^l\}_{l=0,\ldots,L}$ of the differential operator N is obtained, by definition, if a coarse-grid discrete operator N^l, restricted to the refined cells, is a Galerkin approximation of the fine-grid discretisation. By definition, the restriction $R^{l,l+1}N^l$, as an approximation to N^{l+1}, is called a Galerkin approximation if

$$\left\{N^l(q^l;q^{l-1})\right\}^l_{i,j} = \left\{R^l_{l+1}N^{l+1}(P^{l+1}_l q^l;q^l)\right\}^l_{i,j}, \quad \forall (i,j,l) \in I_f.$$

The restriction operator for the solution, $\overline{R}^l_{l+1} : X^{l+1}(\Omega^{l+1}) \rightarrow X^l(\Omega^l_f)$, is defined by the operator which approximately takes the integral mean value (cf. (2.3.5))

$$\left\{\overline{R}^l_{l+1} q^{l+1}\right\}^l_{i,j} = \frac{1}{4} \sum_{m \in K(i,j)} q^{l+1}_m, \quad \forall (i,j,l) \in I_f.$$

As noted in Sec. 2.4, this restriction is second-order accurate. For the right-hand side a restriction operator, $R^l_{l+1} : \overline{Y}^{l+1}(\Omega^{l+1}) \rightarrow \overline{Y}^l(\Omega^l_f)$, is defined by

$$\left\{R^l_{l+1} r^{l+1}\right\}^l_{i,j} = \sum_{m \in K(i,j)} r^{l+1}_m, \quad \forall (i,j,l) \in I^l_f. \tag{3.2.1}$$

The operator for the prolongation of a correction for the solution, $P^{l+1}_l : \overline{X}^l(\Omega^l) \rightarrow \overline{X}^{l+1}(\Omega^{l+1})$, is defined by

$$\left\{P^{l+1}_l q^l\right\}^{l+1}_m = q^l_{i,j}, \quad \forall m \in K(i,j) \text{ and } \forall (i,j) \in I^l_f. \tag{3.2.2}$$

As shown in [33], [18] and [20], these restrictions and prolongation appear to give very good multigrid performance (together with the point Guass-Seidel relaxation). The prolongation (3.2.2) and restriction (3.2.1) satisfy the multigrid rule (cf. [10], [14], [42])

$$m_p + m_r > 2m,$$

where m_p is the order of accuracy of the interpolation used in the prolongation, (for P^{l+1}_l this is $\mathcal{O}(h_l)$), m_r the order of accuracy the restriction (for R^l_{l+1} this is $\mathcal{O}(h^2_l)$, taking into account that R^l_{l+1} is a restriction in $\overline{Y}^l$) and $2m$ the order of the differential equation ($2m = 1$ for the Euler equations). For the given definitions of restrictions and prolongation, the set of first-order accurate discrete equations, exclusive the equations involving a green boundary, form a locally nested sequence of discretisations, (i.e., the coarse-grid discretisation is a Galerkin approximation of the fine-grid discretisation). The first-order accurate reconstruction which uses first-order accurate computation of virtual states (hence first-order weak consistency), yields a locally nested sequence.

3.2.3. The FAS and FMG scheme. In the nonlinear multigrid algorithm FAS the equations for the first-order accurate discretisation are solved. We identify the discrete operator with first-order accuracy by N^l_{I}, and with second-order accuracy by N^l_{II}. The set of equations to be solved is then given by

$$N^l_{\mathrm{I}}(q^l; q^{l-1}) = s^l, \tag{3.2.3a}$$

where the right-hand side s^l is given by

(3.2.3b)

$$s^l_{i,j} = \begin{cases} r^l_{i,j}, & (i,j,l) \in I_c, \\ \left\{ N^l_{\mathrm{I}}(\overline{R}^l_{l+1} q^l; q^{l-1}) \right\}^l_{i,j} \\ \quad - \left\{ R^l_{l+1} \left(N^{l+1}_{\mathrm{I}}(q^{l+1}; q^l) - s^{l+1} \right) \right\}^l_{i,j}, & (i,j,l) \in I_f, \end{cases}$$

and where $r^l_{i,j}$ is defined by (2.3.10). Upon convergence of the nonlinear multigrid scheme, the solution of (3.2.3) satisfies (2.3.12), with appropriate definition of the numerical flux $F^l_{i,j,l}$ and it satisfies (2.3.14).

The collective, symmetric point Gauss-Seidel relaxation on each level of refinement acts as a smoother in the FAS scheme. For each cell $\Omega^l_{i,j}$ visited, the state $q^l_{i,j}$ is updated, by iterating on the local system $\{N^l_{\mathrm{I}}(q^l; q^{l-1})\}^l_{i,j} = r^l_{i,j}$, solving for $q^l_{i,j}$ using Newton iteration. The residual tolerance for this Newton iteration is taken such that in all but exceptional cases only one or two iteration steps are performed. The cells on each level are visited in an order which is equivalent to the usual lexicographical order. After a first relaxation sweep has been done, another sweep is done in the reversed direction. This smoother is shown to be very efficient [15], in both subsonic and supersonic Euler flow computations. A FAS cycle, where all q^l, $l = 0, \ldots, L$ are improved, is a recursive algorithm defined by the following steps:

(1) improve the solution q^l by applying p pre-relaxations to (3.2.3a) for level l, resulting in the approximate solution $(q^l)_0$;
(2) compute the right-hand side s^{l-1}, determined by (3.2.3b) for level l;
(3) improve the solution q^{l-1} by applying σ FAS cycles to the equations (3.2.3a) for level $l-1$;
(4) compute the correction of the solution, given by the difference of the present coarse-grid solution and the coarse-grid restriction, $d^{l-1} = q^{l-1} - \overline{R}^{l-1}_l (q^l)_0$;
(5) improve the solution q^l by adding the prolongation of the coarse grid correction, $q^l = (q^l)_0 + P^l_{l-1} d^{l-1}$;
(6) improve the solution q^l by applying q post-relaxation sweeps to the system (3.2.3a) for level l.

The steps (2)–(5) together are called the coarse grid correction. These steps are skipped for level 0.

The initial solution on the finest level is obtained by application of nested iteration (FMG) [2], [3], [10]. For a level $l \geq 0$, a cycle of the FMG scheme is recursively defined as follows:

(1) if $l = 0$ initialise the solution q_0 with some 'arbitrarily' chosen solution; if $l > 0$ initialise the solution on level l with a prolongation $\overline{P}^l_{l-1} q^{l-1}$;

(2) improve the solution on level l by application of γ FAS cycles with level l as highest level;
(3) if level l is not the highest level, then apply the FMG iteration cycle with a finest level $l+1$;

Throughout the experiments presented in this chapter we use $\sigma = 1$ (V-cycles), $p = q = 1$ (a single pre-relaxation and a single post-relaxation) and $\gamma = 1$ (a single V-cycle, before starting on a higher level). The prolongation $\overline{P}_l^{l+1}$ used in the FMG algorithm is bilinear interpolation.

3.2.4. Defect correction. The set of equations for second-order accuracy is solved, using iterative defect correction [10], [1]. The set of higher-order discretised equations on a level l, are given by

$$N_{\mathrm{II}}^l(q^l; q^{l-1}) = r^l. \tag{3.2.4}$$

The IDeC algorithm solves these equations, by iteratively solving

$$N_{\mathrm{I}}^l(q^l; q^{l-1}) = s^l,$$

applying the FAS scheme, with a modification to the right-hand side s^l for the equations for a cell of the composite grid, i.e. step (2) of the FAS algorithm. An initial solution for the IDeC process is obtained by application of the FAS algorithm to (3.2.3). In the IDeC iteration the right-hand side s^l depends on the defect of the higher-order accurate equations through

$$(3.2.5)\; s_{i,j}^l = \begin{cases} \left\{N_{\mathrm{I}}^l(q^l; q^{l-1})\right\}_{i,j}^l - \left\{N_{\mathrm{II}}^l(q^l; q^{l-1})\right\}_{i,j}^l, & (i,j,l) \in I_c, \\ \left\{N_{\mathrm{I}}^l(\overline{R}_{l+1}^l q^{l+1}; q^{l-1})\right\}_{i,j}^l \\ \quad - \left\{R_{l+1}^l \left(N_{\mathrm{I}}^{l+1}(q^l, q^{l+1}) - s^{l+1}\right)\right\}_{i,j}^l, & (i,j,l) \in \mathrm{I}_f. \end{cases}$$

In step (2) of the FAS algorithm the right-hand side is computed by (3.2.5). Upon convergence of the IDeC scheme (3.2.4) is satisfied.

In [18] it is shown that one nonlinear multigrid cycle per defect correction cycle is sufficient and most efficient. In all our experiments we do the same and use a single nonlinear multigrid cycle per defect correction cycle.

Before any local grid refinement is introduced, the solution on a basic level l_b is approximately computed. This is done by application of the nested iteration FMG, one or two FAS cycles to approximately solve the first-order accurate discretisation and then a sufficient number of IDeC cycles, for second-order accuracy.

3.3. Refinement cycles

Solution-adaptive grid refinement involves the grid to be refined at some stage in the solution process. Based on an a-posteriori estimation of relevant quantities appearing in the refinement criterion, the grid is refined where these

quantities exceed a pre-set or solution-dependent threshold value, (cf. [5], [29], [26]).

A computation with use of local grid refinement starts with applying the FMG algorithm and possibly subsequent iterative defect correction, so that an approximate solution is obtained for the uniform grid on some basic level l_b. Introduction of local grid refinements is accomplished by the following refinement algorithm, for l the highest level present:

(1) determine which cells on level l should be refined, or may be deleted from the system, based on the refinement criterion and an a-posteriori estimation of the relevant quantities used in the refinement criterion and based on the requirement that a virtual state $v_{i,j}^l$ only depends on q^l and q^{l-1};
(2) decide whether a grid on level $l+1$ should be created, call the (new) highest level, level L;
(3) refine the grid and delete obsolete cells on all levels, from l_b up to and including level $L-1$;
(4) initialise the approximate solution of the newly created refinements by application of the prolongation $\widehat{P}_m^{m+1}$, for $m = l_b, \ldots, L-1$ (similar to the FMG algorithm);
(5) improve the solution on all levels by application of ρ FAS (first-order discretisation) or IDeC (second-order discretisation) iterations on the composite grid;
(6) either apply a refinement cycle on the new system, or solve the present system of equations by a sufficient number of iterations;

The decision in step (2) of the refinement algorithm may be determined by the answer to the question whether the grids on all currently present levels have sufficiently converged, or whether the highest level allowed has already been reached. Notice that for newly created cells, the refinement cycle actually is an application of the nested iteration algorithm FMG, introduced in the previous section. For the prolongation $\widetilde{P}_l^{l+1}$ a bilinear interpolation is used for all newly created cells. In second-order computations, after initialisation of the solution for newly created cells, defect correction is continued, without applying the nonlinear multigrid scheme to the first-order accurate system (3.2.3) first. The number of iterations ρ before a new refinement cycle is started, step (5) of the refinement algorithm, determines to a large extent the efficiency of the adaptive grid refinement method. However, using an insufficient number of iterations in step (5) may yield a grid too much distorted by the insufficiently converged numerical solution, as compared with the grid that would be obtained with a converged solution. In practice, $\rho = 1$ or $\rho = 2$ for a first-order discretisation appears to yield a grid virtually the same as the grid obtained by using a fully converged solution. For a second-order discretisation $\rho = 4$ or $\rho = 5$ is sufficient.

3.4. Some aspects of implementation

In order to perform multigrid accelerated Euler flow computations with solution-dependent local grid refinement, a computer code has been developed in portable FORTRAN 77. This code consists of two modules. One module is called BASIS, and is entirely devoted to set up and do maintenance on the data structure. It is described in [17]. The second part, called EULER, consists of all routines related to the adaptive multigrid Euler flow computations. This module is described in [37]. Recently some work has been done on vectorisation of this code for a CRAY Y-MP. This resulted in an additional module, called EUVEL, which is presented in [25].

The data structure reflects the quad-tree relations of the cells in the geometric structure. The grid, made up by the geometric structure, is composed of so-called *patches*. A patch consists of a corner point, and possibly a horizontal wall, a vertical wall and a cell. If a patch contains a cell, sufficient neighbour patches exist to provide the necessary edges and vertices. On the other hand, each corner point, each edge and each cell of the geometric structure belongs to some patch. All data in the structure are stored and referenced through these patches. The patches in the data structure are also related in a quad-tree structure. As a matter of fact, the tree of cells is a subtree of the tree of patches.

In the linked list that implements the quad-tree structure, nine *pointers* are used for each patch. One pointer to the parent of a patch, four pointers to the kid patches and four pointers to the neighbours of a patch. In the FORTRAN implementation the use of pointers in the linked list is emulated by a large, two dimensional array of type *integer*. Each patch has a unique number. For each patch the patch numbers of its parent, its possible kids and its possible neighbours are stored in a row of the integer array. Furthermore, for each patch a set of properties is kept, which identify the *type* of the patch: whether the patch contains a cell, whether the horizontal wall or vertical wall is part of the green boundary or the boundary of the domain, whether the cell contained in the patch may be refined upon the earliest possible occasion, etc.. These properties are stored in a two dimensional array of type *logical*, the column identified by the unique number of a patch and the row identified by a specific property. Finally, all real data for the numerical problem are kept in another two dimensional array of type *real*. These data are also addressed through the unique patch number. For each patch a total of 18 real numbers are stored. The data structure handled by BASIS has a much wider range of applications than Euler flow computations and than cell-centred discretisation schemes.

All actions on the data in the data structure are performed through a depth-first traversal of the tree. The subroutines performing the necessary numerical actions work by application of this tree traversal algorithm.

For each patch visited through this algorithm, a subroutine is called to per-

form some action on the data in the data structure. The quad-tree structure and the use of such a traversal algorithm to perform any task, is very well suited for the implementation of a multigrid algorithm with adaptive mesh refinement. However, as noted in [25], automatic vectorisation (i.e., vectorisation by the compiler) of a code of this nature does not gain any performance. Therefore, in the vector extension library presented in [25], subroutines are provided that collect pointers to patches containing the geometric structure of a single level of refinement, and place them in an appropriate order in a separate array of pointers. The order of pointers in this array makes the smoothing suited for vectorised processing. Furthermore, in the vectorisable extension the original subroutines that are called for each single patch to perform some numerical action on the data, have been modified so they work on multiple patches. If instead of Osher's numerical flux, Van Leer's numerical flux function is used, (in both the vectorised and non-vectorised code) an overall speed-up of approximately four is obtained.

3.5. Shock reflection

3.5.1. The problem. In this section we consider a shock reflection problem. This is a gas-dynamics problem of a supersonic flow along a flat solid surface. The domain of definition is $\Omega = \{(x, y) \mid 0 < x < 4, 0 < y < 1\}$, and the flat surface is located at $y = 0$. A shock is impinging from the point $(0, 1)$, at an angle of 29° with the positive x direction. The boundary conditions for this solution are, at the inflow boundary $x = 0$ given by

$$\begin{aligned} u(0, y) &= 1, \\ v(0, y) &= 0, \\ M(0, y) &= 2.9, \\ \rho(0, y) &= 1. \end{aligned}$$

At the inflow boundary $y = 1$, the flow perpendicular to the horizontal boundary is subsonic, and we impose three conditions. These are approximately given by

$$\begin{aligned} u(x, 1) &= 0.90322141, \\ v(x, 1) &= -0.17459319, \\ M(x, 1) &= 2.37807192. \end{aligned}$$

The boundary $y = 0$ is the solid wall, and we impose impermeability, given by

$$(w \cdot n)|_{y=0} = 0,$$

where n is a normal on the boundary $\partial\Omega$ and $w = (u, v)^T$ the velocity vector. The shock is reflected at the solid wall, at an angle of about 23.279°. The exact solution is known from simple gas-dynamics shock relations. It is a

piecewise constant function. The impinging and reflected shocks form the discontinuities of this function.

3.5.2. Refinement. The domain Ω is rectangular. The coarsest grid used, level zero, is a 6×2 grid. The basic level is $l_b = 1$. Since away from the shock, the exact solution is a constant function, for both, a first-order discretisation and a second-order discretisation, the local discretisation error is zero, away from the shock. For an adaptive computation, it is sufficient *for this problem* to use only the variation of the solution as the refinement criterion. Hence, grids are refined based on the first undivided difference of a component of the solution. According to research on the use of undivided differences as a general refinement criterion, it is found that of any component of the solution, the first undivided difference of density gives best results ([5], [6]).

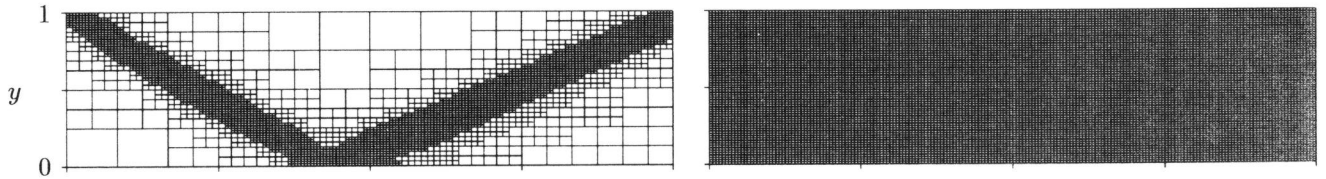
3.5.3. Results. For this problem, away from the shock, the discretisation yields equations with local discretisation error equal to zero. The accuracy of the results will be determined to a large extent by the *resolution* provided by the grid used.

First-order discretisation. The equations resulting from the first-order discretisation, are solved on an adaptively refined grid. For the highest level L we take $L = 4$, $L = 5$ and $L = 6$ respectively, to study the convergence behaviour. The number of multigrid FAS iterations (V-cycles) for each refinement cycle is two. A cell is refined if the absolute value of the first undivided difference in either x direction or y direction exceeds 0.05. We consider refinements to have become obsolete if the absolute value of the first undivided difference of density drops under 0.025. In Tab. 3.5.1 the number of cells used

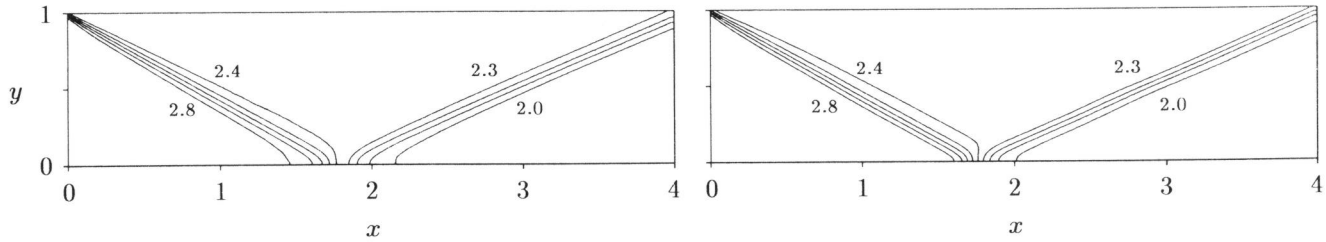
TABLE 3.5.1. *Final number of cells used for shock reflection problem; first-order discretisation.*

	locally refined		uniform	
L	composite	total	composite	total
4	1533	2040	3072	4092
5	3582	4772	12288	16380
6	7797	10392	49152	65532

are given for both the locally refined and the uniform grids with a highest level $L = 4, 5, 6$ respectively. Notice that the number of cells approximately doubles, going from finest level $L = L^*$ to finest level $L = L^* + 1$. Fig. 3.5.1 shows for $L = 5$ the grid obtained by local refinement and the corresponding uniform grid, with iso-plots of the Mach number on both the adaptive and non-adaptive grid respectively. In Fig. 3.5.2 the convergence histories of both the adaptive method and the method using uniform grids are given. These



a. *Grids.*



b. *Iso-line plots Mach number.*

FIGURE 3.4.1. *Iso-lines of Mach number for shock reflection problem on locally refined and uniform grids; first-order discretisation;* $L = 5$.

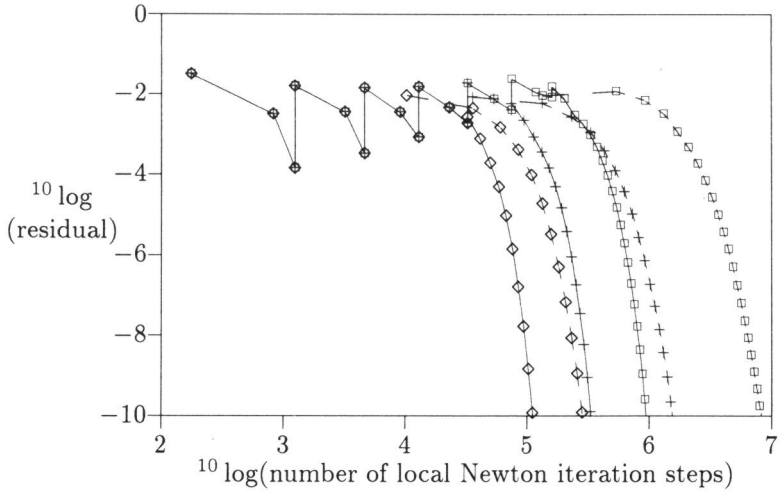
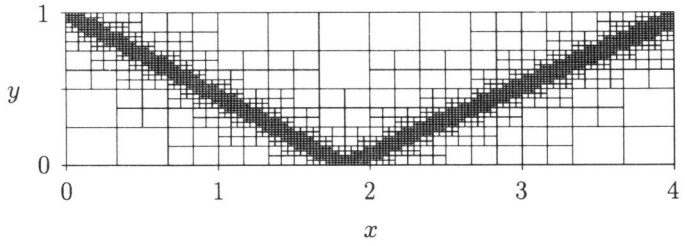


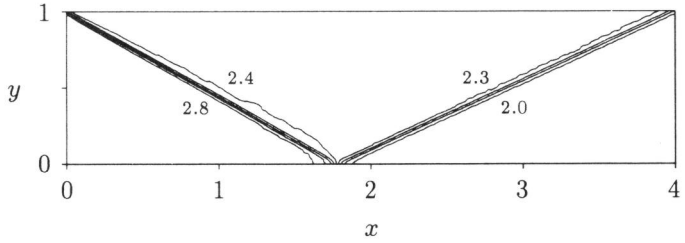
FIGURE 3.5.2. *Residual vs. amount of work: convergence histories for adaptively refined and uniform grids; first-order discretisation;* $\diamond$*:* $L = 4$*;* $+$*:* $L = 5$*;* $\square$*:* $L = 6$*; ———: locally refined; — —: uniform.*

figures show the logarithm of the mean of the four discrete L_1 norms of the residual of the first-order discretisation, on the components of $\overline{X}_c(\Omega_c)$, for all $(i,j,l) \in I_c$ defined by $(A^l_{i,j})^{-1}\{N^l_{\mathrm{I}}(q^l; q^{l-1}) - r^l\}^l_{i,j}$, vs. the logarithm of number of Newton iteration steps performed (i.e., the iteration used in the point relaxation to relax the nonlinear system for each cell). Each of the four norms is the discrete L_1 norm of a residual of the discretisation of one of the conservation laws. For $L = 6$ the number of Newton iteration steps to convergence up to machine precision for the adaptive method is about nine times less than the number of iterations needed when a uniform grid is used, while virtually the same solution is obtained (cf. Fig. 3.5.1 and Fig. 3.5.2). For $L = 5$ the number of iterations for the adaptive method is about five times less and for $L = 3$ this is about 2.5 times less.

Second-order discretisation. We use the second-order discretisation N^l_{II}, with the Van Albada limiter (cf. [35]), and virtual states as defined by (2.5.4). The refinement decision is the same as for the first-order discretisation. The number of defect correction iterations in each refinement cycle is five. It appears that after five defect correction cycles possible wiggles in the 'initial' solution have vanished. The final locally refined grid and iso-lines of Mach number for $L = 5$ are shown in Fig. 3.5.3 The number of cells for local refinement with this second-order discretisation is given in Tab. 3.5.2. Notice that the number of cells for all levels $L = 4, 5, 6$ is much smaller for the adaptive computation with the second-order discretisation than for the computation with



a. *Grid.*



b. *Iso-line plot of Mach number.*

FIGURE 3.5.3. *Grid and iso-lines of the Mach number for the shock reflection problem on a locally refined grid; second-order discretisation;* $L = 5$.

TABLE 3.5.2. *Final number of cells used for shock reflection problem; second-order discretisation.*

	locally refined		uniform	
L	composite	total	composite	total
4	924	1228	3072	4092
5	2004	2668	12288	16380
6	4707	6272	49152	65532

the first-order discretisation. For the second-order discretisation some extra refinements may be introduced, apart from the refinements introduced by the refinement criterion itself. These are introduced in order to let virtual states for the discretisation on level l, depend only on the solution on levels l and $l-1$.

Convergence histories for locally refined and uniform grids are given in Fig. 3.5.4. This figure shows the logarithm of the mean of the four discrete

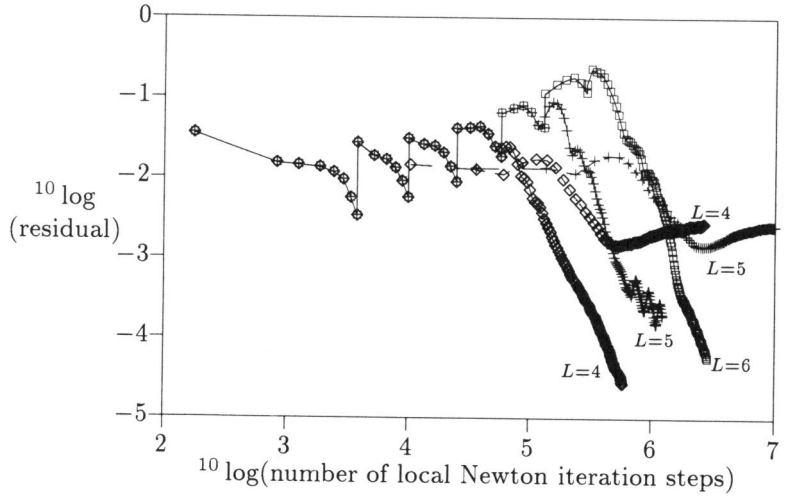


FIGURE 3.5.4. *Residual vs. amount of work: convergence histories for defect correction and second-order discretisation on uniform and locally refined grids;* $\diamond$*:* $L = 4$*;* $+$*:* $L = 5$*;* $\square$*:* $L = 6$*; ———: locally refined; — —: uniform.*

L_1 norms of the second-order discretisation on $\overline{X}_c(\Omega_c)$, vs. the logarithm of the number of Newton iteration steps. We did not consider $L = 6$ and a uniform grid. This problem is so large that it causes the computer to start swapping pieces of memory to disk, resulting in a very large processing time. Apparently, the defect correction process does not converge for uniform grids. The reason for this is possibly the following. On a uniform grid, with finest level L, many more Fourier modes can be represented than on the refined grid with finest level L. Especially low-frequency modes can be represented very well on the uniform grid, better than on the locally refined grid. In [8] an amplification factor $g \approx 1$ for low-frequency Fourier modes is found, in case of the linear convection problem in two space dimensions. However, it should be stressed that for this linear convection problem this high amplification factor corresponds to functions that are constant in the characteristic direction of the problem.

TABLE 3.5.3. *CPU-time required per Newton iteration step for the shock reflection problem.*

	adaptive-grid code		uniform-grid code
	locally refined	uniform	
FAS	0.96 ms/iter.	0.93 ms/iter.	0.84 ms/iter.
IDeC/FAS	1.14 ms/iter.	1.03 ms/iter.	0.84 ms/iter.

The defect correction algorithm for the locally refined grids does converge. For the second-order discretisation and defect correction, the discretisation on a locally refined grid yields a more robust algorithm for this problem.

3.5.4. Execution time. In order to get some idea of execution time, for this problem we give CPU-times of our FORTRAN research code on an SGI IRIS INDIGO XS workstation. All optimisation was done by the compiler. We give the average CPU-time it takes for all operations of the local refinement computation, per Newton iteration step. Note that the Newton iterations referenced here are *local* Newton iterations, used in the point relaxations. The results are given in Tab. 3.5.3. This table also shows the average CPU-time for another multigrid code, developed to work with uniform grids only. This particular code, called EULER7, implements the same multigrid and defect correction algorithms as used in the code for adaptive computations (cf. [18]). The FAS algorithm on a locally refined grid appears to be only three percent more expensive per Newton iteration step than on a uniform grid with the adaptive code. The iterative defect correction for a locally refined grid appears to be about 18% more expensive per Newton iteration step than iterative defect correction on a uniform grid. For the FAS algorithm, the adaptive code *with* local grid refinement, appears to be about 14% more expensive per Newton iteration step, than the non-adaptive code EULER7. For iterative defect correction, the adaptive-grid code is about 34% more expensive per Newton iteration step than EULER7.

3.6. Airfoil flow

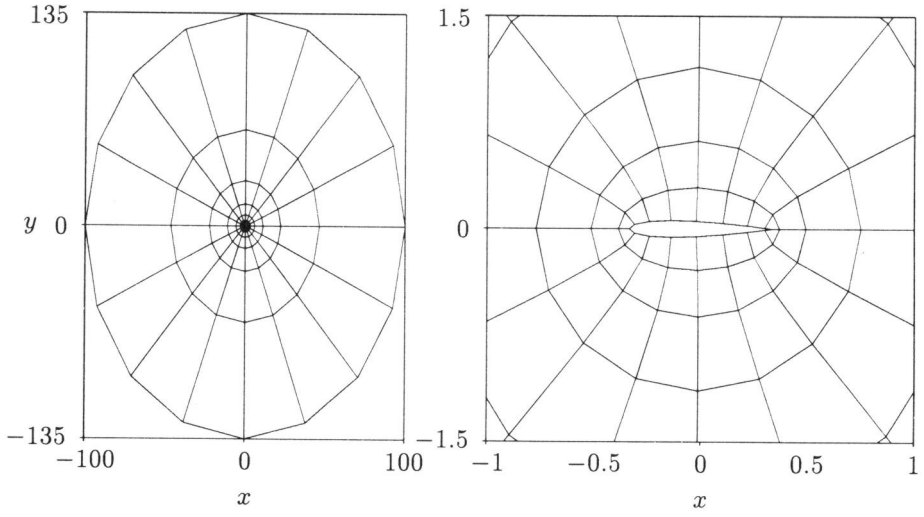
In this section we consider the transonic flow around the NACA0012-airfoil. The flow conditions at the far-field boundary are: $M_\infty = 0.8$, angle of attack $\alpha = 1.25°$, $\rho_\infty = 1$ and $(u_\infty^2 + v_\infty^2) = 1$. The computational domain extends to about 100 chords to all sides.

As the second-order operator N_{II}^l, we use the Van Albada limiter scheme [35], [31] and, again, third-order accurate computation of virtual states, as given by (2.5.4). The limiter scheme is used, since spurious wiggles in the solution are expected if a second-order non-limiter scheme is used.

In the refinement criterion we use first undivided differences of the density, in both the streamwise direction and perpendicular to the streamwise direction. Two thresholds are used, one for each direction. This prevents the

algorithm from refining in the neighbourhood of a shock only. It allows the algorithm also to find the contact discontinuity, and to resolve the expansion region. Then, we not only get a good resolution of the shock, but also a good resolution of the expansion, and this in turn is important for the accurate computation of the lift and drag coefficients. The use of only one threshold value (i.e., the same threshold for both criteria) would be inefficient for a small threshold value (too many refinements) On the other hand, a larger threshold value only refines at strong discontinuities.

The grid used is an O-type grid. The coarsest grid is a 5×8 grid. The highest level is $L = 5$. The uniform grid for level $l = 1$ is shown in full and in detail in Fig. 3.6.1. A cell is refined if the first undivided difference of density



a. *In full.* b. *In detail.*

FIGURE 3.6.1. *Uniform grid of level $l = 1$, around NACA-0012 airfoil.*

in flow direction is larger than 0.02, or when this difference in the direction perpendicular to the flow is larger than 0.004. The final adaptively refined grid, with $L = 5$, is shown in Fig. 3.6.2. In Fig. 3.6.3 the Mach number distributions are shown both for an adaptively refined and for a uniform grid. The pressure distribution for the uniform grid and for the locally refined grid are shown in Fig. 3.6.4. For the lift and drag coefficient on the adaptively generated composite grid we find $c_l = 0.3480$, $c_d = 0.0235$. On the non-adaptive grid we find $c_l = 0.3512$ and $c_d = 0.0235$. The difference between these values is less than 10% of the scatter found between different reference results listed in [41]. This reference gives $c_l = 0.3632$ and $c_d = 0.0230$ obtained

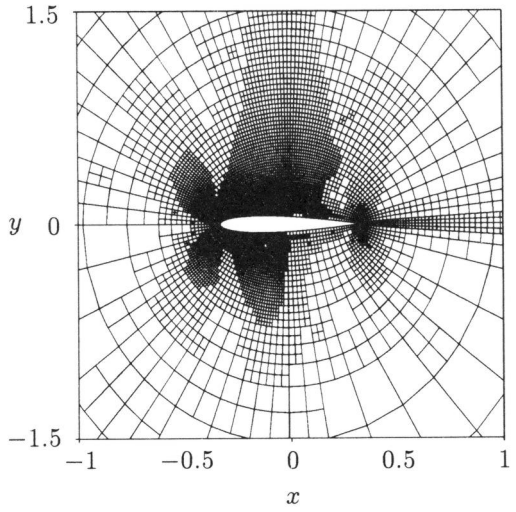
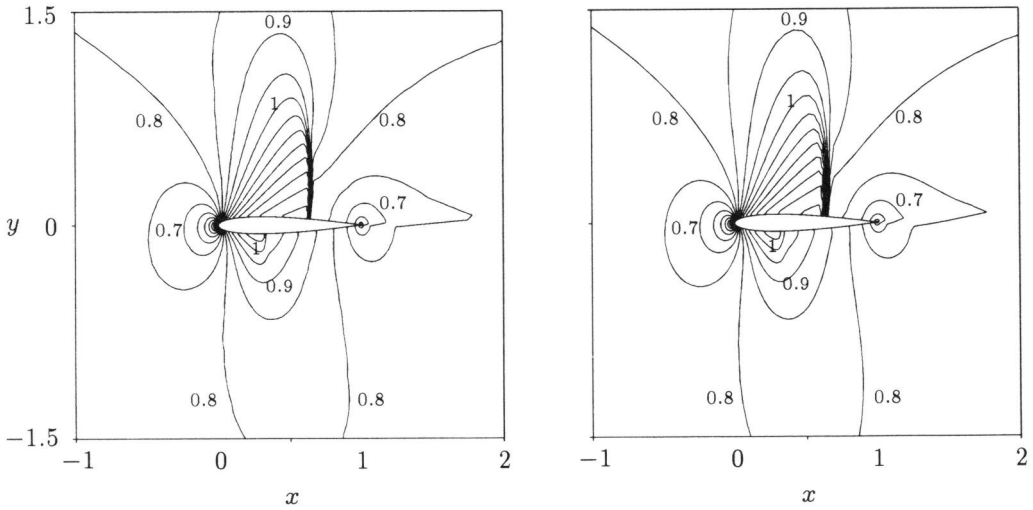


FIGURE 3.6.2. *Locally refined grid with $L = 5$, around NACA0012 airfoil.*



a. *Adaptively refined grid; $L = 5$.* b. *Uniform grid; $L = 4$.*

FIGURE 3.6.3. *Iso-line plots of the Mach number for the transonic flow around a NACA0012 airfoil; $\alpha = 1.25°$; $M_\infty = 0.8$; locally refined grid: $L = 5$; uniform grid: $L = 4$.*

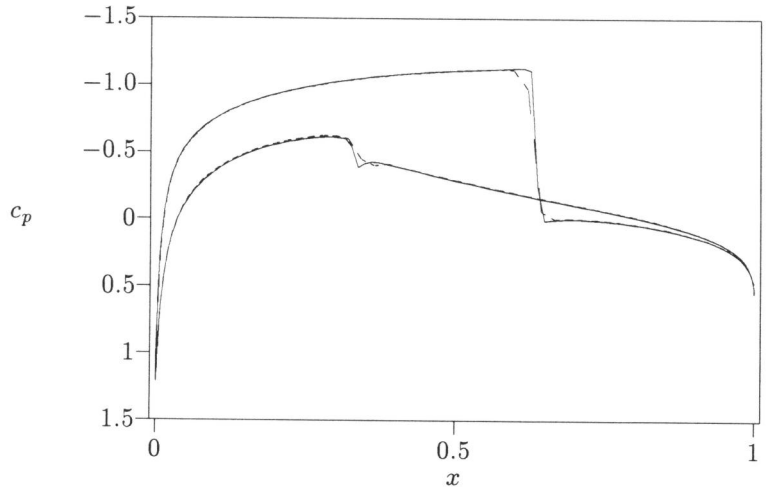


FIGURE 3.6.4. *Pressure distribution for adaptively refined and uniform grids for transonic flow around a NACA0012 airfoil;* $\alpha = 1.25°$; $M_\infty = 0.8$; ———: *locally refined grid,* $L = 5$; — —: *uniform grid,* $L = 4$.

on a grid of 20480 cells, by Schmidt and Jameson [41]. The number of cells on the adaptively generated composite grid is 7876 and a total number of 10488 cells was used in the computation. The non-adaptive grid uses 10240 cells on the finest grid and a total number of cells of 13640. The convergence histories of both the adaptive and non-adaptive case are shown in Fig. 3.6.5. The adaptive computation takes about three times less work than the computation on the non-adaptive grid.

3.7. Spurious entropy generation for subsonic flow past a compression corner

3.7.1. Introduction. In this section we study the numerical entropy generation for the steady, two dimensional Euler equations and a perfect gas. Numerical approximations of the subsonic Euler flow along a compression corner show spurious entropy generation, which is virtually independent of the mesh size of the computational grid. Sometimes, simple incompressible flow models are used to describe the flow in the vicinity of geometric singularities, such as the compression corner shown in Fig. 3.7.1. The presumption that the velocity near the corner is small, is then used to justify incompressible wedge flow as a model. The compressibility effect is often accounted for by a correction, such as the Prandtl-Glauert rule. However, in [40] it is found analytically, using the hodograph transformation, that even in a subsonic case the flow does not have to stagnate in the corner. Hence, the use of an incompressible model to ap-

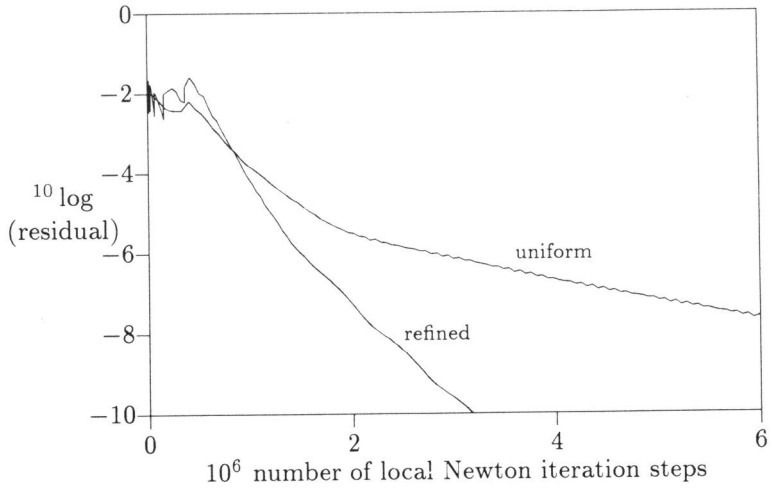


FIGURE 3.6.5. *Residual vs. amount of work: convergence histories of defect correction and second-order discretisation for NACA0012 airfoil flow;* $\alpha = 1.25°$; $M_\infty = 0.8$; *locally refined grid:* $L = 5$; *uniform grid:* $L = 4$.

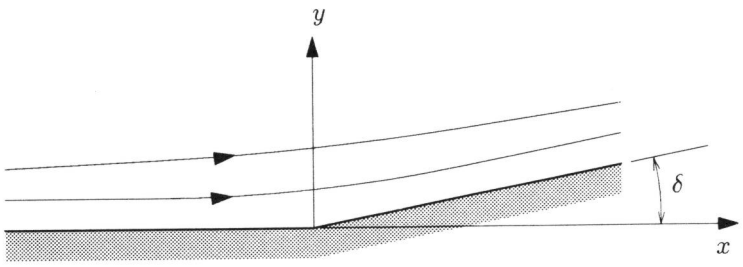


FIGURE 3.7.1. *Subsonic flow along a kinked wall.*

proximate the subsonic compressible flow along a compression corner may be unrealistic, since for incompressible models the corner is a stagnation point. Furthermore, the singularity in the solution at the corner for compressible flow appears to be considerably more complex than that for incompressible flow. Trying to remove the singularity by postulating the same singularity as for incompressible potential flow (i.e., an inverse power of the distance to the corner) in the Euler model, has proved to be unsuccessful (cf. [39] and [38]).

In a numerical method, discrete convergence of the solution may be obtained by considering the compression corner as the limit of a smooth surface, which is parametrised by the mesh width of the grid. However, convergence is slow. This problem shows a way to use a local grid refinement technique as a tool to approximate a singular solution. For the approximation of a singular solution, the grid becomes singular too, for mesh width $h_l \to 0$. We use the solution-adaptive grid refinement method to obtain reasonable discrete convergence, without excessive computational cost.

3.7.2. The problem. A typical layout of the geometry of the problem at hand is shown in Fig. 3.7.1, and the corner with angle δ. We consider a flow problem which is known to be *exactly* homentropic. As is well-known, in steady subsonic flow, the entropy along a streamline is constant. Hence, a first requirement for obtaining a homentropic flow is to impose a constant entropy at inflow in the domain. Furthermore, at the inflow boundary the velocity vector is imposed. This velocity corresponds with that of an incompressible potential flow (i.e. irrotational) along the surface, which is analytically known by conformal mapping. At outflow the corresponding pressure is imposed.

Boundary conditions are incorporated into the discretisation scheme in a way which is consistent with the discretisation in the interior of the computational domain (cf. [16]). In subsonic flows, this requires *three* boundary conditions at inflow and *one* boundary condition at outflow. Notice that by just obeying these numbers in subsonic flow computations, mathematical well-posedness is not yet guaranteed. For a study of the mathematical well-posedness of some typical subsonic outflow boundary conditions, we refer to [19].

The domain of definition is the area covered by the grid in Fig. 3.7.2. With the grid shown in Fig. 3.7.2 and similar 16×16 and 8×8 grids, a straightforward application of the discretisation gives for the entropy a result as shown in Fig. 3.7.3. This clearly shows that the entropy error is virtually independent of the refinement of the grid.

3.7.3. Nature of the error. In [38] a number of possible causes for this behaviour are considered, in order to make the nature of the zeroth-order error plausible.

First the local discretisation error is studied. In the discretisation an inconsistency is encountered, which is due to the kink in a grid as in Fig. 3.7.2. This inconsistency in the discretisation appears in equations derived for the

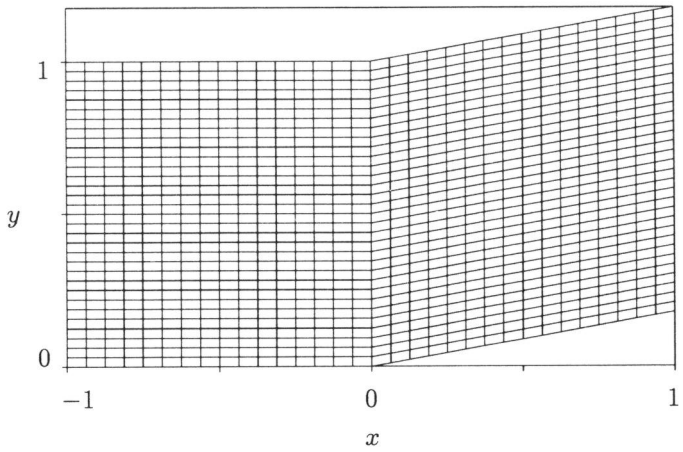


FIGURE 3.7.2. *The uniform grid along a compression corner;* 32×32.

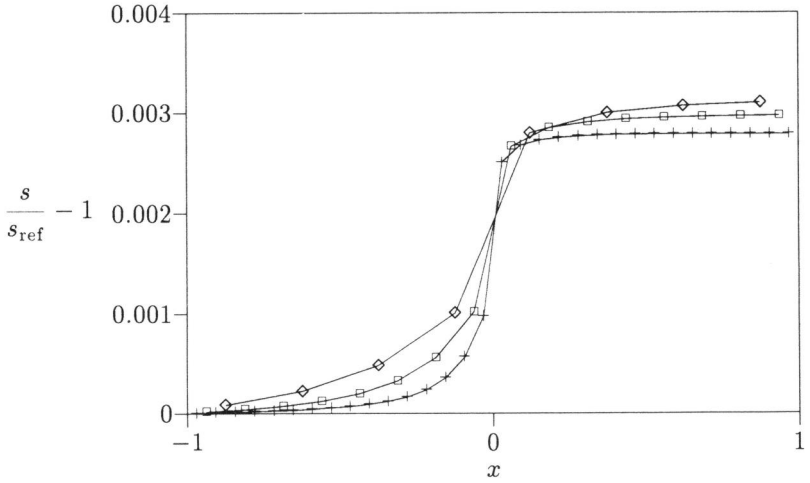


FIGURE 3.7.3. *The entropy error for the computation of the flow past a compression corner;* $\delta = 10°$; $\diamond$: 8×8; $\square$: 16×16; $+$: 32×32.

cells directly bordering the grid line emerging from the corner. It appears from numerical experiments with the kinked grid and a smooth flow, that this inconsistency has no adverse effect on the entropy error.

Secondly, the discretisation of the solid-wall boundary conditions are studied, again by numerical experiment. For this purpose, the wedge-shaped wall is replaced by a continuously curved wall. The boundary conditions are discretised identically as in the case of the wedge shaped wall. It appears that the discretisation of the solid wall boundary conditions also has no adverse effect on the entropy error.

From the experiments it becomes plausible that the singularity in the exact solution itself causes the bad convergence behaviour.

3.7.4. Parametrised smooth wall. In this subsection we study the entropy error in the flow along a continuously curved wall. The shape of the wall that we use, is given by

$$y_w(x) = \begin{cases} 0, & x \le -\frac{1}{2}l, \\ \left(\frac{(x+\frac{1}{2}l)^3}{l^2} - \frac{1}{2}\frac{(x+\frac{1}{2}l)^4}{l^3} \right) \tan\delta, & -\frac{1}{2}l < x \le \frac{1}{2}l, \\ x\tan\delta, & \frac{1}{2}l < x. \end{cases} \tag{3.7.1}$$

Here, l is the length of the curved part of the wall and δ the angle between the positive x direction and the uncurved part of the wall at $x > \frac{1}{2}l$. The geometry is shown in Fig. 3.7.4. The wall is defined in such a way that

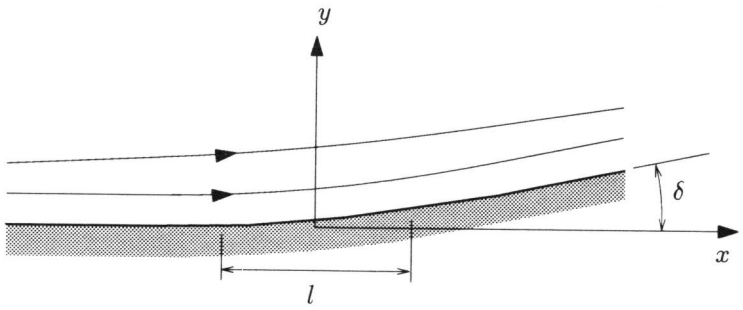


FIGURE 3.7.4. *The flow along a continuously curved wall.*

$y_w \in C^2[-1, \cos\delta]$. The grid used is shown in Fig 3.7.5. In Fig. 3.7.6, for $\delta = 10°$ and $l = 1$ the entropy error along the wall is shown for the 32×32, 16×16 and 8×8 grid. We find that the entropy error is first-order in mesh size as already mentioned in Sec. 3.7.3.

In these results the length l of the continuously curved wall segment is the same for all grids considered. Now we re-compute the flow along a continuously curved wall (3.7.1), but we let l depend on the mesh size. We use

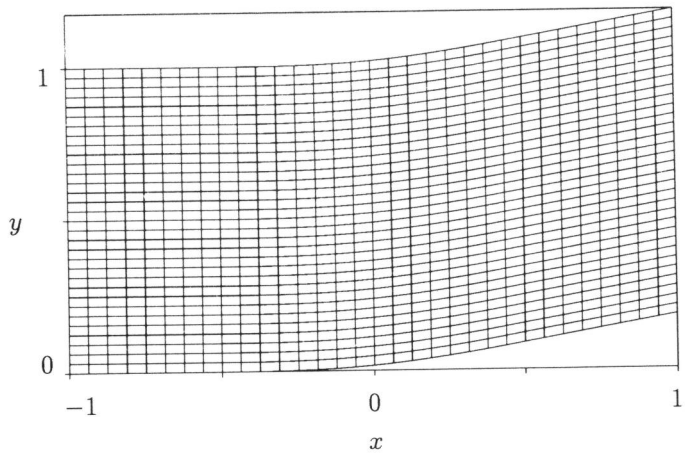


FIGURE 3.7.5. *A grid along a continuously curved wall;* 32×32

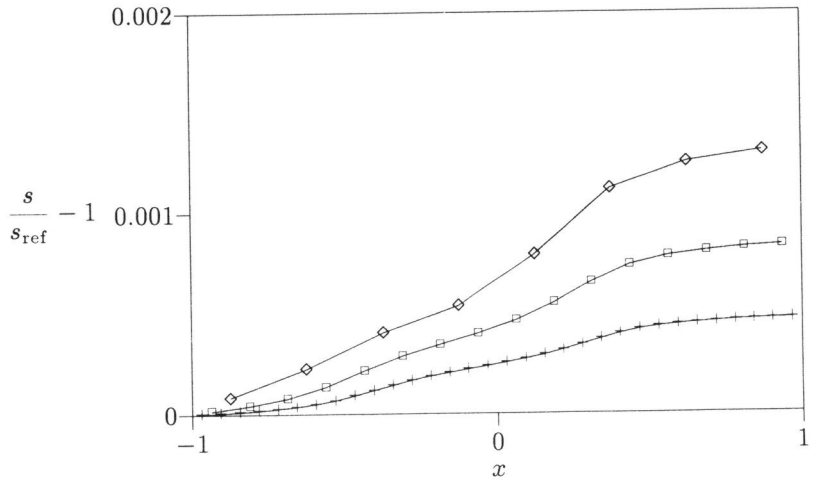


FIGURE 3.7.6. *The entropy error for the computation of the flow past a continuously curved wall;* $\delta = 10°$; $\diamond$: 8×8; $\square$: 16×16; $+$: 32×32.

$l = \mathcal{O}(h^p)$, $p > 0$, where h is the mesh width in the direction of the x axis. If $p < \infty$, it is clear that for $h \to 0$ the curved wall degenerates into the wedge-shaped wall. The results in Fig. 3.7.3 and in Fig. 3.7.6 can be considered as those for the limit $p = \infty$ and $p = 0$, respectively. The number of cells N along the curved part of the wall is

$$N = \frac{l}{h}.$$

Hence, with $l = \mathcal{O}(h^p)$, we have

$$N = \mathcal{O}(h^{p-1}).$$

For $p \geq 1$ and in the limit $h = 0$, there would be only one grid line emerging from the corner, and we arrive at a similar situation as for $p = \infty$ (i.e., zeroth-order entropy error). Thus, looking for decreasing entropy errors for decreasing mesh width, solving the problem of subsonic flow past a compression corner, we must take $0 < p < 1$. In Fig. 3.7.7 the behaviour of the entropy

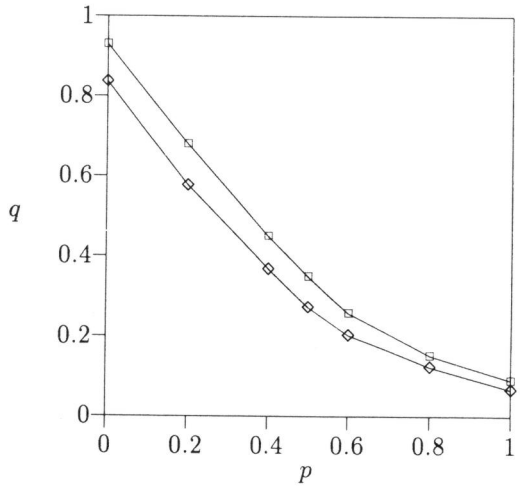


FIGURE 3.7.7. *The order of discrete convergence;* $l = h^p$; $\delta = 10°$; ◇: *computed from* 16×16 *and* 32×32 *grids;* □: *computed from* 32×32 *and* 64×64 *grids.*

error when l decreases as a function of h. Here, for l we take $l = c_1 h^p$, with c_1 constant, and for the entropy error we assume the form

$$\left\| \frac{s}{s_{\text{ref}}} - 1 \right\|_\infty = c_2 h^q,$$

c_2 constant, and $\|.\|_\infty$ a discrete supremum norm. From numerical experiments with $c_1 = 1$, on a 16×16, a 32×32 and a 64×64 grid, q has been determined as a function of p. As already shown, for $p = 0$ we find q approaches 1, as h approaches 0.

We find that for a subsonic flow, the flow along a curved wall can be used to successfully compute the flow along a compression corner. For $p \in (0,1)$, and $h \to 0$, the curved wall becomes kinked and the entropy error vanishes, because for any $p \in (0,1)$ it appears that $\|s/s_{\text{ref}} - 1\|_\infty = \mathcal{O}(h^q)$, and $q > 0$. If we want to have the entropy error disappear at the same rate as l, then according to Fig. 3.7.7, we should take $p \approx 0.4$. In Fig. 3.7.8 we

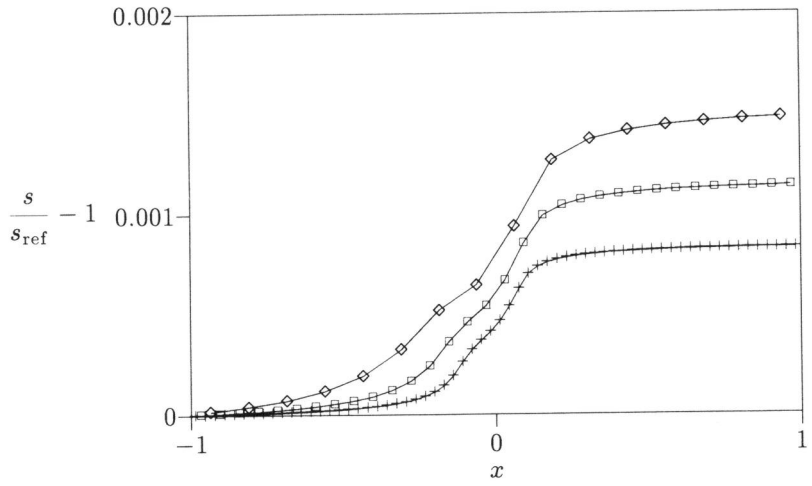


FIGURE 3.7.8. *The entropy error for the computatrion of the flow past a continuously curved wall;* $l = h^{0.4}$; $\delta = 10°$; $\diamond$: 16×16; $\square$: 32×32; $+$: 64×64.

give the entropy error distributions along the wall, as obtained for $p = 0.4$ on a 16×16, a 32×32 and a 64×64 grid. Given the rather low order of accuracy, $q(p = 0.4) \approx 0.4$, reduction of the entropy error below some required tolerance level may become expensive when applying uniform grid refinement. As an example of adaptive local grid refinement, in Fig. 3.7.9 we give results obtained on a 32×32 grid and local refinements, with $p = 0.4$. The entropy error is used in the refinement criterion. The criterion is based on the discrete gradient of the entropy error in streamline direction $(v \cdot \nabla s)/|v|$, multiplied by the square root of the area of a cell, and with v the velocity vector, s the entropy and ∇ a discrete gradient operator. Fig. 3.7.10 gives the locally adapted grid for the result of Fig. 3.7.9: the 32×32-grid with four additional levels of local refinement is shown. Notice that for decreasing mesh width,

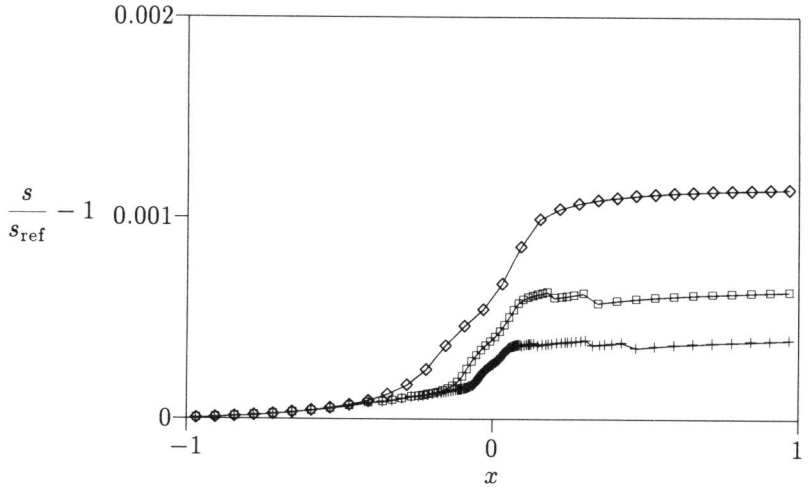


FIGURE 3.7.9. *The entropy error for computation of flow past a continuously curved wall with adaptively refined grids;* $l = h_l^{0.4}$; ◇: 32×32; □: 32×32 *with two levels of refinement;* +: 32×32 *with four levels of refinement.*

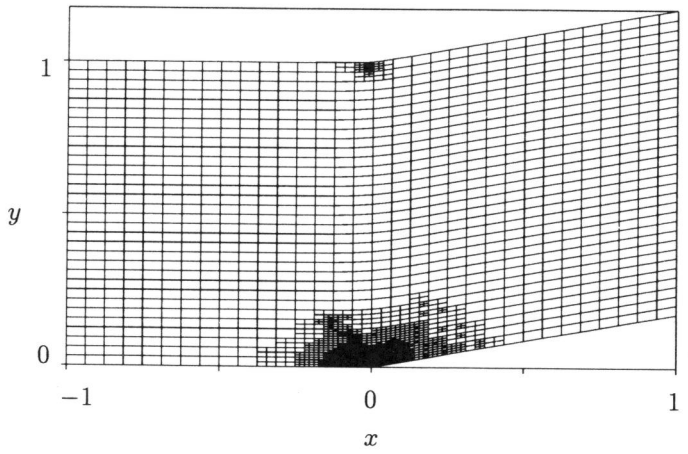


FIGURE 3.7.10. *Adaptively refined grid;* $l = h_l^{0.4}$; $\delta = 10°$; 32×32 *grid, with four levels of local refinement.*

the refined regions get smaller and also get closer to the corner. Finally, in Tab. 3.7.1, for the three grids considered in Fig. 3.7.9 we give an impression

TABLE 3.7.1. *Geometrical data for the adaptively refined grids, with* $l = h_l^{0.4}$, $\delta = 10°$.

grid	l	N
32×32 without local refinement	0.3299	5
32×32 with two levels of local refinement	0.1895	12
32×32 with four levels of local refinement	0.1088	28

how the rounded corner converges to the kink for decreasing mesh width.

3.7.5. Concluding remarks. It is possible to remove the zeroth-order global discretisation error from the numerical approximation of the subsonic Euler flow past a compression corner. Poor computational efficiency due to the rather low order of accuracy, may be effectively circumvented by application of local, solution-adaptive grid refinement. Numerical results indicate that local refinement (combined with the smooth discretisations of the kinked wall) *is* an alternative to reduce the error, without increasing the computational effort excessively .

We found the paradoxical result that by making a sequence of geometrically less accurate discretisations of the compression ramp, a numerical solution of the flow with better error behaviour can be obtained. The less accurate discretisations of the kinked wall employ discrete smooth versions of the exact kinked wall. By making the discretisation of the geometry dependent on the mesh size h, an $\mathcal{O}(h^q)$, $0 < q < 1$, entropy error can be obtained. For this result, the grid has a singularity at the compression corner.

3.8. Shock wave on a continuously curved, convex surface

The main subject of this section is the flow near both ends of a shock wave, as it appears in transonic flows. Generally, a local supersonic flow suddenly becomes subsonic, after being expanded along a convex surface. We say that the supersonic region is fenced off by a shock. In such a steady, inviscid flow along a continuously curved, convex wall (cf. Fig. 3.8.1), the following two intriguing flow regions exist:

- the flow region where the shock wave abuts the continuously curved, convex wall (the shock-foot region);
- the flow region near the other end of the shock wave (the shock-tip region).

We start by reviewing some studies of the local flow in both regions. Next, a detailed numerical study is presented of the flows in these regions. The

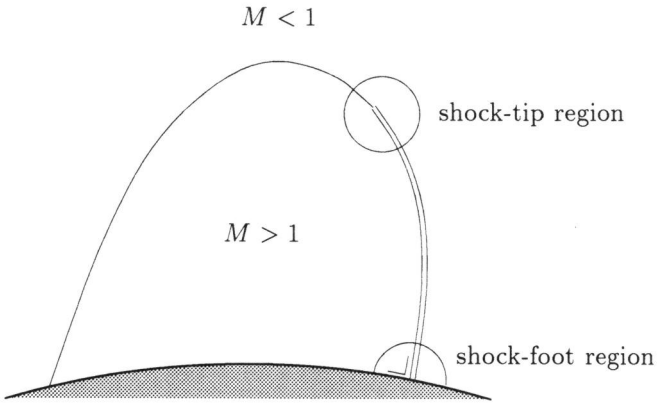


FIGURE 3.8.1. *Shock wave fencing off local supersonic zone along continuously curved, convex wall.*

solution-adaptive method is used to 'zoom in' to interesting phenomena. Emphasis is on the surface pressure, on the downstream side of the shock, and the way in which the shock and the smooth part of the flow merge, at the tip of the shock.

In the following, the shock foot and shock tip are defined as the lower and upper end point of the shock wave, respectively.

3.8.1. Shock-foot region. We introduce a Cartesian coordinate system (x, y), with its origin at the foot of the shock and with the direction of the positive y axis perpendicular to the surface, pointing into the flow. We also introduce a curvilinear system (ξ, η), with its origin at $x = 0$, $y = 0$, and positive η along the shock. We assume that the flow is homentropic upstream of the shock, as is the case for the steady flow around an object in uniform free stream. We consider two situations at the shock foot; one called a *normal extension* and the other an *oblique extension*.

Normal shock with normal extension. At the foot of a shock, the following physically interesting situation occurs. Since we consider a continuously curved surface, the flow does not change direction when passing through the foot of the shock. Hence, the shock is necessarily normal to the surface and, passing through the shock foot, the gas motion satisfies the normal-shock relations. However, the gas motion also has to satisfy the equations of curvilinear motion.

$$\frac{\partial p}{\partial y} = \frac{\gamma p M^2}{R}. \tag{3.8.1}$$

Here, M is the local Mach number, p the pressure, $R < \infty$ the local radius of curvature of the surface, at $(x, y) = (0, 0)$, and γ the (constant) ratio of

specific heats.

We consider a shock which is a normal shock at $\eta = 0$, and which is still a normal shock within some distance $\epsilon > 0$ away from the foot. Such a situation is shown in Fig. 3.8.2. Hence the normal-shock relations are satisfied for all

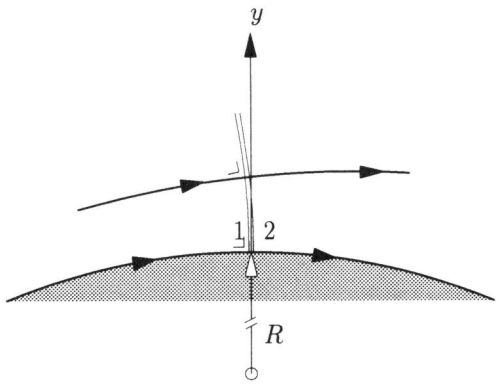


FIGURE 3.8.2. *Normal shock with normal extension.*

$\eta < \epsilon$. We refer to such a shock as a normal shock with normal extension.

The relations for a normal shock, and for curvilinear motion may be contradictory. Satisfying shock relations for a normal shock with normal extension, does not necessarily mean satisfying equation (3.8.1). We show that this is the case, by showing that a contradiction arises for a normal shock wave with normal extension, unless the flow directly upstream of the shock foot is a flow with a particular Mach number. We introduce the subscripts 1 and 2 for the upstream and downstream side of the shock, respectively (Fig. 3.8.2). From (3.8.1) we have at the shock foot, downstream of the shock

$$\frac{\partial p_2}{\partial y} = \frac{\gamma p_2 M_2^2}{R}. \tag{3.8.2}$$

We may also write for $\partial p_2/\partial y$

$$\frac{\partial p_2}{\partial y} = \frac{p_2}{p_1}\frac{\partial p_1}{\partial y} + p_1 \frac{\partial}{\partial y}\left(\frac{p_2}{p_1}\right). \tag{3.8.3}$$

According to (3.8.2) and (3.8.3), $\partial p_2/\partial y$ may be expressed in terms of the flow quantities upstream of the shock. The normal-shock relations express M_2 and

p_2 in terms of M_1 and p_1. These standard relations are given by (cf. [23])

$$p_2 = \frac{2\gamma M_1^2 - (\gamma - 1)}{\gamma + 1} p_1, \tag{3.8.4}$$

$$M_2^2 = \frac{2 + (\gamma - 1)M_1^2}{2\gamma M_1^2 - (\gamma - 1)}. \tag{3.8.5}$$

For the pressure gradient $\partial p_1/\partial y$ at the shock foot, at the upstream side of the shock, we already have

$$\frac{\partial p_1}{\partial y} = \frac{\gamma p_1 M_1^2}{R}. \tag{3.8.6}$$

The gradient $\frac{\partial}{\partial y}\left(\frac{p_2}{p_1}\right)$ of the pressure ratio across a normal shock can be found from (3.8.4). The homentropic condition upstream of the shock implies

$$\partial (p_t)_1/\partial y = 0, \tag{3.8.7}$$

where $(p_t)_1$ is the total pressure p_t upstream of the shock, given by

$$\frac{p_t}{p} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma - 1}}.$$

With the homentropic relation (3.8.7) and (3.8.6), it can be easily derived from (3.8.4) that

$$\frac{\partial}{\partial y}\left(\frac{p_2}{p_1}\right) = \frac{-4\gamma}{\gamma + 1}\left(1 + \frac{\gamma - 1}{2} M_1^2\right)\frac{M_1^2}{R}. \tag{3.8.8}$$

Setting (3.8.2) equal to (3.8.3) and substituting (3.8.4)–(3.8.8), yields a quadratic equation for M_1^2:

$$M_1{}^4 - (\gamma + 1)M_1^2 - 1 = 0. \tag{3.8.9}$$

From (3.8.9), with $M \in \mathbb{R}$, $M \geq 0$, it follows that the only particular Mach number M_1 which allows a normal shock wave with normal extension, is $M_1 = M_1^*$, defined by

$$M_1^* = \sqrt{\frac{\gamma + 1}{2} + \sqrt{\left(\frac{\gamma + 1}{2}\right)^2 + 1}}. \tag{3.8.10}$$

By a more cumbersome derivation, this particular Mach number was also derived in [34], [24] and [44].

Normal shock with oblique extension. For the flow in the shock-foot region and Mach number $M_1 \neq M_1^*$ further analytical results are presented in [44]. The main result presented in [44] is that at the downstream side of the shock foot, for $M_1 \neq M_1^*$, in general a singular solution must occur, with the property that at the foot the shock wave has an infinitely large curvature. Similar as for the normal shock with normal extension as depicted in Fig. 3.8.2, this shock can be illustrated as has been done in Fig. 3.8.3. At an arbitrarily small distance

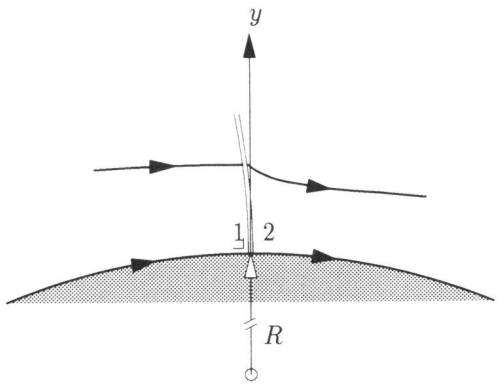
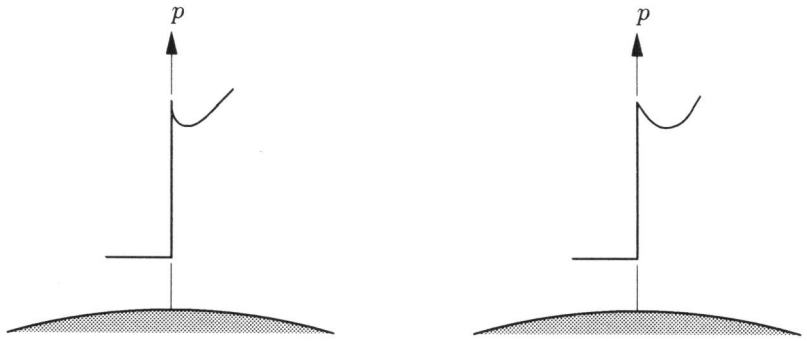


FIGURE 3.8.3. *Normal shock with oblique extension.*

$\eta = \epsilon > 0$ from the surface, the shock is an oblique shock. This singular change from normal shock wave at $\eta = 0$ to an oblique shock wave for $\eta > 0$, implies that at the downstream side of the shock at the foot an infinitely steep decrease in the surface pressure occurs. This singular behaviour prohibits a contradiction between the curvilinear motion at the downstream side of the shock, and the normal-shock relations. In the following we refer to shocks of this type as normal shocks with oblique extension.

According to [44] support for this singular solution seems to come from the fact that post-shock expansions (though not infinitely steep, see Fig. 3.8.4b) are observed in the results of wind tunnel experiments. However, in our opinion the physics in these wind tunnel experiments is too different from inviscid physics to let them be of much support to an inviscid analysis. Instead of support from wind tunnel experiments, support for at least the singular behaviour of the post-shock surface pressure comes from gas-dynamics. If the post-shock pressure correction would have a finite steepness (Fig. 3.8.4b), there would have to be a region of size > 0, in which the equation of curvilinear motion and the normal-shock relations are contradictory, which in steady flow is physically impossible.

Apart from this result, in [44] an analytical result is presented which renders a range of Mach numbers $(M_1^*, M_1^{**}]$ for M_1, for which a normal shock on a



a. *Infinitely steep expansion.* b. *Finitely steep expansion.*

FIGURE 3.8.4. *Post-shock surface pressure corrections.*

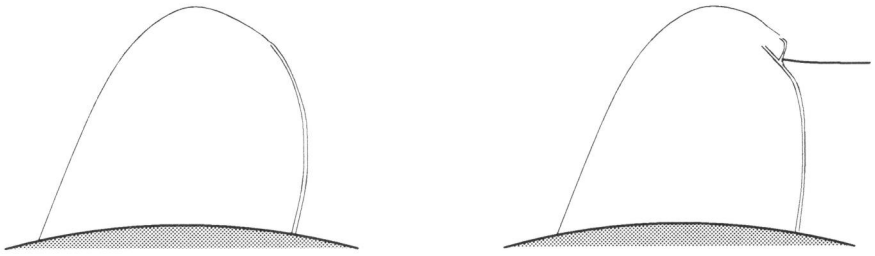
convex surface is impossible. Here, M_1^{**} is defined by

$$M_1^{**} = \sqrt{\frac{2 + \frac{\gamma+1}{2}}{2 - \frac{\gamma+1}{2}}}.$$

Note that $M_1^* < M_1^{**}$ for any admissible γ, i.e., $\gamma \in [1, 3]$. Though we do not know of any examples of a steady shock wave normal to a convex surface, with $M_1 \in (M_1^*, M_1^{**}]$, an intuitive understanding of this 'forbidden' range is still missing.

Present study of a shock-foot region. We see that some reason exist which motivates a numerical study of steady Euler flow in the shock-foot region. In doing so, we look at the convergence of the steepness for decreasing mesh width. To study the expansion's steepness for decreasing mesh width, without making excessive computational costs, we apply the solution-adaptive technique to zoom in at local flow features.

3.8.2. Shock-tip region. For the shock-tip region some uncertainty exists about how the shock wave merges into the smooth part of the flow. Some authors claim that the shock wave becomes of vanishing strength into the direction of the sonic line (cf. Fig. 3.8.5a). Others claim that it bifurcates such that a triple-shock configuration (or upside-down λ-configuration) occurs (cf. Fig. 3.8.5b). Notice that a triple-shock configuration must contain at least one additional discontinuity. In [4] it is proved that it is not possible for *three* zones of different continuously varying states to exist and meet in one singular point. Therefore, in Fig. 3.8.5b an extra zone has been added by introduction of a contact discontinuity, indicating that the fluid which just passed two shocks (above the contact discontinuity), is in a different state than the fluid which just passed only one shock.



a. *Single-shock configuration.* b. *Triple-shock configuration.*

FIGURE 3.8.5. *Two types of shock-tip configurations.*

Some profound analytical studies exist which support either the configuration in Fig. 3.8.5a or that in Fig. 3.8.5b. Representative examples of these studies are those presented in [9] and [22], respectively. In [9] the authors consider the transonic small perturbation equation and find through a hodograph method that the connection of the sonic line to the shock wave is in general tangent but distinguishable, i.e., generally the shock tip is a point of inflection (i.e. at their connection, shock wave and sonic line have curvature of opposite sign). It is found that the entire shock wave is of strong type (transition from supersonic to subsonic speed across the shock), and they conclude that other types of shock-tip configurations are possible, but only in exceptional cases. Opposed to this, the author of [22] finds that the triple-shock-tip configuration sketched in Fig 3.8.5b is the commonly occurring configuration. The single-shock configuration in Fig. 3.8.5a may be regarded as a special case of the triple-shock configuration, i.e., with the shock-triple-point in the highest possible position.

Of all analytical studies on shock-tip configurations, shortcomings are that they are local and use rather simplified equations, such as the transonic small perturbation equation. Though still with simplified equations, investigations of the local shock-tip flow as an integral part of a much more global flow have been made numerically. Examples of such numerical studies are those presented in [27], [12], and [43]. Each of these numerical works is concerned with the transonic small perturbation equation. Whereas the local tip flow certainly can be described by good approximation as a potential flow, the global flow generally cannot. Not only for studying the shock-foot flow, but also for studying the shock-tip flow, it makes sense to consider the full Euler equations. The above mentioned numerical results are limited in accuracy by the use of both the simplified equations and by the use of relatively coarse grids.

3.8.3. Numerical results. Numerical computations are performed for flows of a di-atomic, perfect gas ($\gamma = 7/5$) around a NACA0012 airfoil. The

computations are started in a non-adaptive way on the uniform 8×5, O-type grid Ω^0. For level 1, this grid is shown in Fig. 3.6.1. The far-field boundary is located at about 100 chord lengths away from the airfoil. The computations are continued, through a nested iteration, up to and including level $l_b = 3$, the 64×40-grid. For $l = 3$, local refinements are introduced. Refinements are introduced for cells for which at least one of its cell faces, in the evaluation of Osher's numerical flux across that cell face, a sonic point is encountered along the wave path. We may also refine cells in a 'buffer zone' next to the cells refined by the above criterion. Details about the refinement strategy are found in [21].

In the nested iteration from Ω^0 up to and including Ω^{l_b}, the discrete equations solved are first-order accurate only. Starting from Ω^{l_b} we begin to solve higher-order discretised equations; those corresponding to the limited Fromm scheme ($\kappa = 0$), as introduced in [35]. The computation of a virtual state is second-order in the first-order discretisation ($p = 1$) and the virtual states are computed third-order accurate in the second-order accurate discretisation ($p = 2$). Hence, consistency is obtained for all equations.

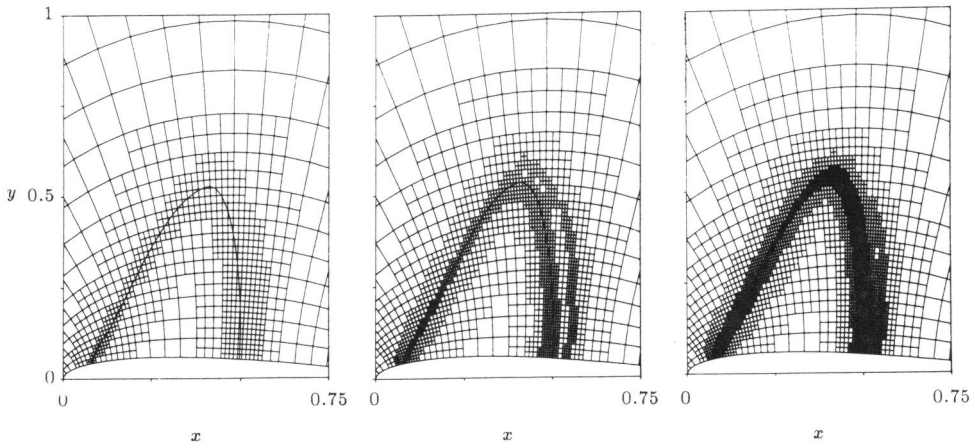
The following cases of far-field boundary conditions are considered

- $M_\infty = 0.8$, $\alpha = 0$,
- $M_\infty = 0.8$, $\alpha = \alpha_{M_1^*}$.

The special angle $\alpha_{M_1^*}$ is the angle at which $M_1 = M_1^*$. The first test case is arbitrarily chosen. However, from other computations its is known that it yields a flow with a shock wave for which $M_1 < M_1^* \approx 1.662$ for $\gamma = 7/5$. Hence, for the first test case we may expect a post-shock pressure correction. Using local grid refinement, we study the corresponding shock-foot flow, as well as the corresponding shock-tip flow.

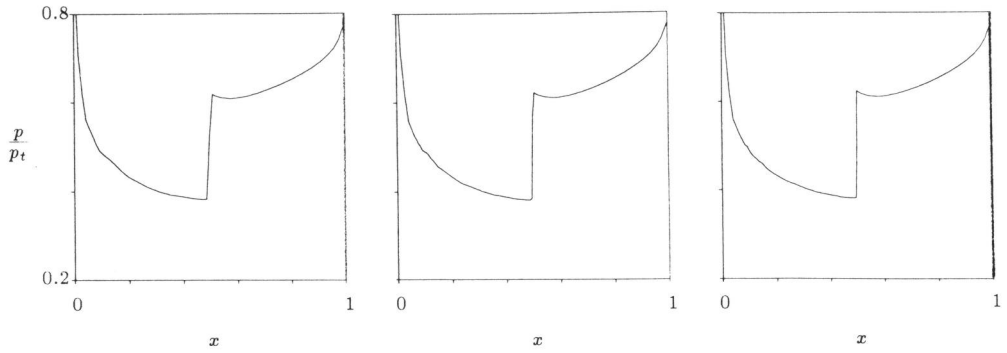
The second test case, with $M_1 = M_1^*$, is studied to verify whether, in agreement with analytical results, the numerical shock-foot flow is without any post-shock pressure correction. The value $M_1 = M_1^*$ may be obtained in an iteration, by varying the free stream Mach number for a fixed angle of attack or by varying the angle of attack α. This iteration has to allow the capturing of a nice shock wave on the convex surface, with M_1 converging to M_1^* in the limit of the iteration process. It appears that for the NACA0012 airfoil, variation of free stream Mach number does not allow $M_1 = M_1^*$. Hence, in the iteration we choose variation of angle of attack α. The iteration for α is based on an approximate Newton iteration using the numerical approximation of the derivative $\partial M_1 / \partial \alpha$. For a detailed description of the algorithm we refer to [21]. Note that according to theory (cf. [44]), this situation is at the limit of existence, with $M_1 = M_1^*$ the lower bound of the forbidden range.

Results for $\alpha = 0$. For $\alpha = 0$ we consider $L = 5, 6, 7$ and we take no buffer zone of cells to be refined. In Fig. 3.8.6 a detail of the various converged composite grids is given, together with the sonic lines found during the computation and used in the refinement criterion. In Fig. 3.8.7 the corresponding upper



a. $L = 5$. b. $L = 6$. c. $L = 7$.

FIGURE 3.8.6. *Converged composite grids with sonic lines for* $\alpha = 0$.



a. $L = 5$. b. $L = 6$. c. $L = 7$.

FIGURE 3.8.7. *Surface pressure distributions for* $\alpha = 0$.

surface pressure distributions are shown. Clearly visible in the latter figures is the occurrence of an expansion downstream of the shock. We study the convergence of the steepness of this expansion for decreasing mesh width. Assuming that in the exact solution, in conflict with theory as described in Sec. 3.8.1, a post-shock expansion occurs which is not infinitely steep (i.e., $|\partial p_2/\partial x| < \infty$), we introduce the error δ^L of the pressure gradient

$$\delta^L \equiv \left(\frac{\Delta p_2}{\Delta x}\right)^L - \frac{\partial p_2}{\partial x}. \tag{3.8.11}$$

In our notations we assume this to be positive for each finest level L. The discretisation is consistent. Then, given the finite gradient of the exact post-shock expansion, and hence regularity of the solution, we may assume discrete convergence for $L \to \infty$ of order $p > 0$. Hence for sufficiently large L we may assume

$$\delta^{L+1} = \left(\frac{1}{2}\right)^p \delta^L, \quad p > 0. \tag{3.8.12}$$

From definition (3.8.11) and the assumption that the solution is regular (and hence (3.8.12)) we should find

$$\left(\frac{\Delta p_2}{\Delta x}\right)^{L-1} - \left(\frac{\Delta p_2}{\Delta x}\right)^L > \left(\frac{\Delta p_2}{\Delta x}\right)^L - \left(\frac{\Delta p_2}{\Delta x}\right)^{L+1}. \tag{3.8.13}$$

In Tab. 3.8.1 we give the values of $\left(\frac{\Delta p_2}{\Delta x}\right)^L$ that are actually found in the

TABLE 3.8.1. *Post-shock expansions,* $\alpha = 0$.

L	5	6	7
$\left(\frac{\Delta p_2}{\Delta x}\right)^L$	−0.28	−0.36	−0.50

numerical experiments, for $L = 5, 6, 7$. Relation (3.8.13) appears not to be satisfied by the present numerical results. Hence, one of our assumptions (i.e., regular solution, L sufficiently large) is not satisfied. In the case that $L = 5, 6, 7$ *is* sufficiently large, the result is in agreement with theory in [44], which predicts a singular solution with $|\partial p_2/\partial x| = \infty$ at the shock foot for $M_1 < M_1^*$.

At the shock-tip, for all values of L considered, we did not observe a triple-shock-configuration as advocated in [22]. The configurations found are comparable to one which is supported by the results of [9]. In all cases considered, the shock does not bifurcate (cf. Fig. 3.8.8 for one detailed view of this). The subtle inflection point is not (yet) observed.

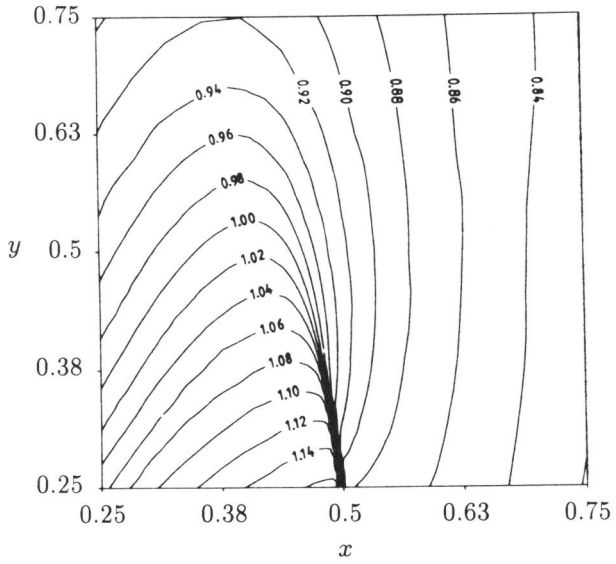
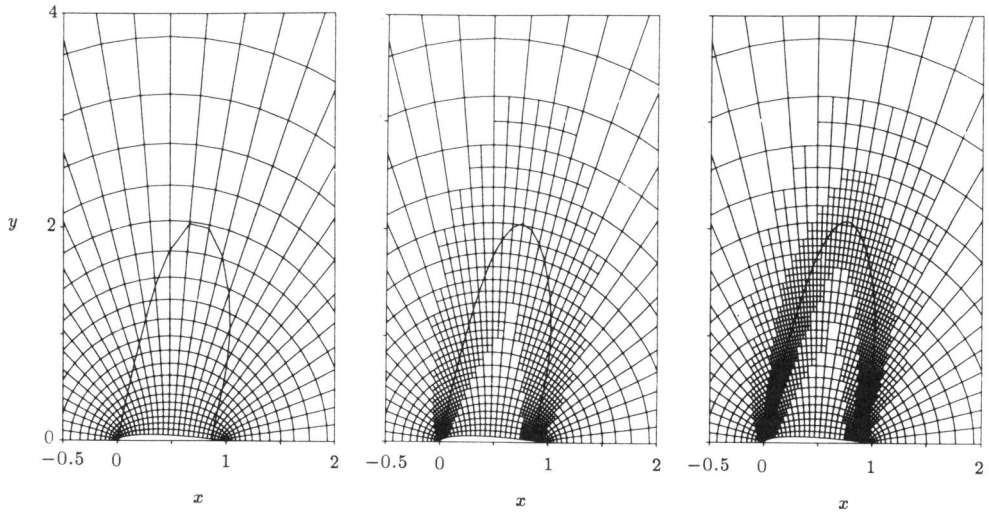


FIGURE 3.8.8. *Mach number distribution around shock tip;* $L = 7$, $\alpha = 0$.

Results for $\alpha = \alpha_{M_1^*}$. For $\alpha = \alpha_{M_1^*}$ we consider $L = 3, 4, 5$ and for the width of the buffer zone of cells to be refined we take two cells. Similar to the previous results, in Fig. 3.8.9 we give the converged composite grids and corresponding sonic lines and in Fig. 3.8.10 the corresponding surface pressure distributions. The main result is that the surface pressure distributions do not show post-shock expansions. In agreement with theory, downstream of the shock wave, all surface pressure distributions in Fig 3.8.10 nicely show a pressure increase only. A remarkable phenomenon that we observed in these experiments is that for all grids the defect correction iteration did not converge (nor did it diverge), as long as the intermediate value of $(M_1)^L$ was in the forbidden range $(M_1^*, M_1^{**}]$ during the iteration for α. The reason why we show results for $L = 3, 4, 5$ only (instead of for $L = 5, 6, 7$ as we did for the previous test case) is that the solution method diverges for $L \geq 6$. With a buffer zone of less than two cells, the iteration diverges, also for $L = 5$.

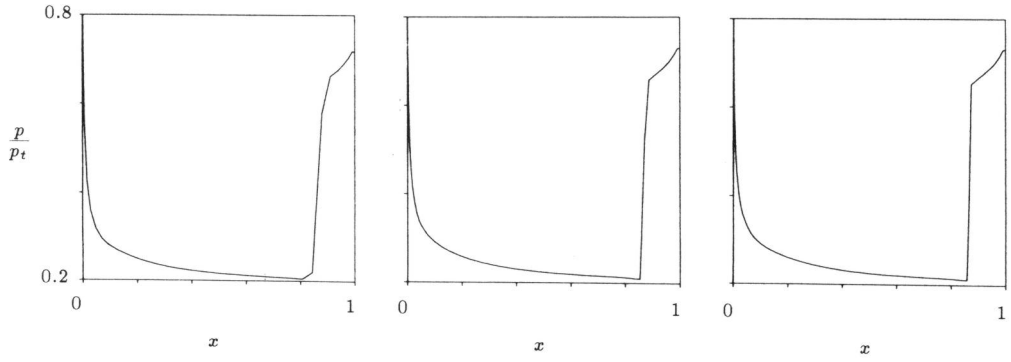
For the shock-tip flow, similar to the results for $\alpha = 0$, the configurations found agree best with those in [9].

The convergence problems which arose for this test case seem to indicate that no steady shock wave perpendicular to the wall can exist in some range with M_1^* as the lower bound. Interesting results can be found in [7] where for a circular cylinder at $M_\infty = 0.5$, Euler flow results are presented which have been obtained by various numerical methods. It is interesting to see that when



a. $L = 3$. b. $L = 4$. c. $L = 5$.

FIGURE 3.8.9. *Converged composite grids with sonic lines for* $\alpha = \alpha_{M_1^*}$.



a. $L = 3$. b. $L = 4$. c. $L = 5$.

FIGURE 3.8.10. *Surface pressure distributions for* $\alpha = \alpha_{M_1^*}$.

using unsteady numerical methods for this circular cylinder test case, fully subsonic initial solutions first seem to evolve to a specific quasi-steady solution, which becomes unsteady when continuing the time integration. Inspecting all circular cylinder results in [7], our conjecture is that the aforementioned, specific quasi-steady solution is at $M_1 = M_1^*$. It seems that the on-set of unsteady flow is when in the time integration the situation is reached where $M_1 = M_1^*$. Results which make this particularly plausible are those presented in [28] and [30].

3.8.4. Concluding remarks. The method for solution-adaptive grid refinement has been applied to study some phenomena of compressible fluid dynamics, related to normal shocks in transonic flows. From the physics point of view, some questions concerning shock-foot and shock-tip region are still open. Analytical results are available, but they are based on simplified models for inviscid flow. Wind tunnel experiments suffer from diffusion, not appearing in the Euler flow model.

In our numerical modeling, we find that at the shock tip the shock merges smoothly with the smooth part of the flow, without a bifurcation of the shock. However, our results do not show a point of inflection where the sonic line is attached to the shock.

We find that at the shock foot, with the Mach number $M_1 < M_1^*$ at the foot upstream of the shock smaller, for the particular number M_1^*, an expansion is present immediately behind the shock, with a pressure gradient which does not seem to converge to a fixed value. This supports the idea that the normal shock has infinite curvature at the foot, unless $M_1 = M_1^*$. Our results for M_1 converging to M_1^*, show that the expansion behind the shock disappears. This implies that at M_1^* the normal-shock relations and the equation for curvilinear motion are satisfied by a non-singular solution.

No Mach numbers M_1 larger than M_1^* are observed. For $M_1 \approx M_1^*$ we observed some convergence problems. According to theory, M_1^* is the infimum of a range of 'forbidden' Mach numbers at the upstream side of a normal shock, in a stationary flow. The convergence problems are most likely related to this forbidden range, since the numerical solutions obtained by methods for unsteady problems seem to develop into an unsteady solution once M_1 reaches M_1^*.

The adaptive local refinement technique shows to be a flexible and efficient tool for detailed studies of such physical phenomena.

References

[1] K. Böhmer, P.W. Hemker and H.J. Stetter, *The defect correction approach*, Defect Correction Methods, Computing, Suppl. 5 (K. Böhmer and H.J. Stetter, eds.), Springer Verlag, Wien, 1984, pp. 1–32.

[2] A. Brandt, *Multilevel adaptive computations in fluid dynamics*, AIAA J. **18** (1980), 1165–1172.

[3] ———, *Guide to multigrid development*, Multigrid Methods, Lecture Notes in Mathematics (W. Hackbusch and U. Trottenberg, eds.), Springer-Verlag, 1982, pp. 220–312.

[4] R. Courant and K.O. Friedrichs, *Supersonic flow and shock waves*, Springer, New York, 1976.

[5] J.F. Dannenhoffer, III, *Grid adaptation for complex two-dimensional transonic flows*, Ph.D. thesis, Massachusetts Institute of Technology, 1987.

[6] ———, *Adaptive grid embedding for complex two-dimensional flows*, Adaptive Methods for Partial Differential Equations (Troy, 1988) (J.E. Flaherty, P.J. Paslow, M.S. Shephard and J.D. Vasilakis, eds.), Rensselaer Polytechnique Institute, Society for Industrial and Applied Mathematics, Philadelphia, 1989, pp. 68–82.

[7] A. Dervieux, B. Van Leer, J. Périaux and A. Rizzi (eds.), *Numerical simulation of compressible Euler flows*, Vieweg, Braunschweig, 1989.

[8] J.-A. Desideri, *Analysis of the convergence of iterative implicit and defect-correction algorithms for hyperbolic problems*, Report NM-R9004, CWI, 1990.

[9] P. Germain and G. Gillon, *Ecoulements transsoniques au voisinage d'un point de rencontre d'une onde de choc et d'une ligne sonique*, Tech. Report ONERA Publication 102, ONERA, 1961.

[10] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer Series in Computational Mathematics, vol. 4, Springer-Verlag, Berlin, 1985.

[11] W. Hackbusch and U. Trottenberg (eds.), *Multigrid Methods* II, Proc. of the 2nd European Conference on Multigrid Methods, held in Cologne, 1985, Lecture Notes in Mathematics, vol. 1228, Springer-Verlag, 1986.

[12] M.M. Hafez and H.K. Cheng, *Convergence acceleration and shock fitting for transonic aerodynamics computations*, AIAA Paper 75–51, 1975.

[13] P.W. Hemker, *Defect correction and higher order schemes for the multi grid solution of the steady Euler equations*, In Hackbusch and Trottenberg [11], pp. 149–165.

[14] ———, *On the order of prolongations and restrictions in multigrid procedures*, J. Comput. Appl. Math. **32** (1990), 423–429.

[15] P.W. Hemker and S.P. Spekreijse, *Multigrid solutions of the steady Euler equations*, Advances in Multi-Grid Methods, Proc. conference held in Oberwolfach, Notes on Numerical Fluid Mechanics, vol. 11 (Oberwolfach, 1984) (D. Braess, W. Hackbusch and U. Trottenberg, eds.), Vieweg Braunschweig, 1985, pp. 33–44.

[16] ———, *Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations*, Appl. Num. Math. **2** (1986), 475–493.

[17] P.W. Hemker, H.T.M. van der Maarel and C.T.H. Everaars, *BASIS: A data structure for adaptive multigrid computations*, Report NM-R9014, CWI, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands, 1990.

[18] B. Koren, *Defect correction and multigrid for the efficient and accurate computation of airfoil flows*, J. Comput. Phys. **77** (1988), 183–206.

[19] ———, *Euler flow solutions for transonic shock wave – boundary layer interaction*, Internat. J. Numer. Methods Fluids **9** (1989), 59–73.

[20] ———, *Multigrid and Defect Correction for the Steady Navier-Stokes Equations, Application to Aerodynamics*, CWI, Amsterdam, 1990, CWI-tract 74.

[21] B. Koren and H.T.M. van der Maarel, *On steady, inviscid shock waves at continuously curved, convex surfaces*, Report NM-R9202, CWI, 1992, submitted to Theoret. Comput. Fluid Dynamics.

[22] A.N. Kraiko, *On the configuration of shock waves enclosing a local supersonic zone*, P.M.M. USSR **49** (1985), 179–183.

[23] H.W. Liepmann and A. Roshko, *Elements of Gasdynamics*, John Wiley and Sons, Ltd., 1957.

[24] C.C. Lin, *On the flow behind curved shocks*, J. Math. Phys. **27** (1948), 105–129.

[25] W.M. Lioen and M. Louter-Nool, *EUVEL: An EULER vector extension library*, 1993, In preparation.

[26] J.A. Michelsen, *Investigation of solution-derivative based adaption criteria for inviscid supersonic flow over bump*, Report Technical University of Denmark AFM Aero 0003-01-TUD, 1991.

[27] E.M. Murman, *Analysis of embedded shock waves calculated by relaxation methods*, AIAA J. **12** (1974), 626–633.

[28] M. Pandolfi and F. Larocca, *A contribution to the numerical prediction of transonic flows*, In Dervieux et al. [7], pp. 292–308.

[29] K.G. Powell, M.A. Beer and G.W. Law, *An adaptive embedded mesh procedure for leading-edge vortex flows*, AIAA Paper 89–0080, 1989.

[30] N. Satofuka and K. Morinishi, *Solution of compressible Euler flows, using rational Runge-Kutta time stepping schemes*, In Dervieux et al. [7], pp. 309–330.

[31] S.P. Spekreijse, *Multigrid solution of monotone second-order discretizations of hyperbolic conservation laws*, Math. Comp. **49** (1986), no. 179, 135–155.

[32] ———, *Second order accurate upwind solutions of the 2D steady Euler equations by the use of a defect correction method*, In Hackbusch and Trottenberg [11], pp. 285–300.

[33] ———, *Multigrid Solution of the Steady Euler Equations*, CWI, Amsterdam, 1988, CWI-tract 46.

[34] H.S. Tsien, *Flow conditions near the intersection of a shock wave with solid wall boundary*, J. Math. Phys. **26** (1947), 69–75.

[35] G.D. van Albada, B. van Leer and W.W. Roberts, *A comparative study of computational methods in cosmic gas dynamics*, Astron. Astrophys. **108** (1982), 76–84.

[36] H.T.M. van der Maarel, *Adaptive multigrid for the steady Euler equations*, Comm. Appl. Numer. Methods **8** (1992), no. 10, 749–760.

[37] H.T.M. van der Maarel, P.W. Hemker and C.T.H. Everaars, *EULER: An adaptive Euler code*, CWI report NM-R9015, 1990.

[38] H.T.M. van der Maarel and B. Koren, *Spurious, zeroth-order entropy generation along a kinked wall*, Internat. J. Numer. Methods Fluids **13** (1991), 1113–1129.

[39] A. Verhoff, *Private communication*, 1991.

[40] A. Verhoff, D. Stookesberry and T. Michal, *Hodograph solution for compressible flow past a corner and comparison with Euler numerical predictions*, Tech. Rep. MCAIR91-005, McDonnell Aircraft Company, Saint Louis, MO, 1991.

[41] H. Viviand, *Numerical solutions of two-dimensional reference test cases*, AGARD-AR-211, AGARD, 1985.

[42] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley and Sons, Ltd., Chichester, 1991.

[43] N.J. Yu and A.R. Seebass, *Inviscid transonic flow computations with shock fitting*, Symposium Transonicum II (K. Oswatitsch and D. Rues, eds.), Springer, Berlin, 1975, pp. 449–456.

[44] J. Zierep, *Der senkrechte Verdichtungsstoß am gekrümmten Profil*, ZAMP **IXb** (1958), 764–776.

CHAPTER 4

A-posteriori estimation of the local discretisation error

4.1. Introduction

The decision where to refine or derefine a given composite grid, ideally may be based on the local discretisation error. Therefore we study the a-posteriori estimation of the local discretisation error. For discontinuous solutions, estimation of the local discretisation error is not possible. However, refinement should not be based solely on the local discretisation error. Apart from sufficiently accurate equations, an accurate solution requires that the grid provides sufficient *resolution*. Resolution of the grid is measured by the derivatives of the function approximated by the numerical solution, the exact solution. The grid should therefore also be refined, on the basis of solution gradients. For the equations associated with cells near discontinuities in the solution, an estimate of the local discretisation error is superfluous, since the grid will be refined due to the approximations of the large gradient in the solution. For the smooth part of the solution an estimate of the local discretisation error can be obtained with sufficient accuracy and may be used in a refinement strategy.

In this chapter we study the local discretisation error and its a-posteriori estimation on a locally refined grid, for the upwind discretisation of a one-dimensional, scalar conservation law. We consider local refinement in a nonlinear multigrid context and emphasise estimating in the neighbourhood of the interface between a coarse and a fine grid.

In the neighbourhood of such a grid interface the discretisation scheme used is different from the one used elsewhere. Estimating the local truncation error in such a situation by means of extrapolation techniques, requires a different treatment of the various contributions to the local truncation error. We introduce two discretisations at grid interfaces and a procedure for the estimation of the local truncation error. We show for a model problem that under sufficient smoothness conditions, the estimate is sufficiently accurate.

Later, the procedure for estimation of the local truncation error is extended to two space dimensions. This requires expressions for the leading terms of the

error in the virtual states. This error is analysed for first-order and second-order accurate schemes. The first-order result is applied in the actual grid refinement criterion for a two-dimensional model problem.

4.1.1. Description of the 1D problem. To start the discussion, we consider a steady state solution for a one-dimensional, scalar conservation law defined on a bounded, open domain Ω, with appropriate boundary conditions

$$\frac{df(u(x))}{dx} = g(x), \quad x \in \Omega \subset \mathbb{R}, \tag{4.1.1a}$$

$$u(x_0) = u_0, \text{ on a proper part of } \partial\Omega, \tag{4.1.1b}$$

where $\partial\Omega \subset \overline{\Omega}$ is the boundary of the domain Ω. Smooth data $g(x)$ is assumed. We study the fully one-sided upwind discretisation of (4.1.1a), augmented with an approximation of (4.1.1b) and we study the solution of the resulting set of equations for an example problem. This one-dimensional conservation law may be considered as a model for fluid dynamics problems. We aim at applying the results of our analysis in a method for steady state fluid dynamics computations in more space dimensions and in a refinement criterion based on an a-posteriori estimate of the local discretisation error. Therefore, we analyse the situation where the partitioning of the domain has a sudden change in mesh size.

We estimate the local discretisation error (a-posteriori) by an extrapolation of the relative truncation error of the difference operator. In general it is only possible to use a standard extrapolation technique if a regular discretisation scheme is used everywhere [3]. Such a regular discretisation scheme admits a simple extrapolation, since asymptotic expansions for the local truncation error can be readily used. For an irregular discretisation scheme, used at boundary interfaces, we propose to split the local truncation error of the discrete operator into two parts. For one part we assume the existence of an asymptotic expansion which can be estimated by extrapolation (the regular or uniform part). The other part (the irregular or non-uniform part) is estimated via estimates of the derivatives which appear in the expression for this part, obtained by Taylor series expansion.

We consider first-order accurate discretisations only. Extension to second-order discretisations is straightforward. For a locally non-uniform grid we consider two discretisations: an inaccurate and an accurate one. Analyses are performed for both a *locally uniform* grid (hence application of a regular discretisation scheme) and a *locally non-uniform* grid (where an irregular discretisation scheme is used). The definitions for locally uniform grid and locally non-uniform grid are given in Sec. 2.2.2. They are introduced to facilitate the treatment of the sudden change in coarseness of the grid, due to the presence of local refinements.

First we present definitions and notations and we introduce the discretisation. Then we investigate the local discretisation error, the global discretisation error for an example problem and a method to estimate the local discretisation error of the discretisation of (4.1.1a) on a locally refined grid. All investigations are supported with results for an example problem. Finally the extension to two dimensions is made.

4.2. Preliminaries for the one-dimensional case

In this section we give the definitions and notations used in the part of this chapter concerned with the analysis of a one-dimensional problem and we describe the one-dimensional problem which serves as an example problem. The notations used here are consistent with the notations used for two dimensions, but for clarity we give a precise description.

Given a partitioning of the domain in cells, we distinguish between function values defined in a cell and function values defined at cell end points. A value in a cell Ω_i^l is denoted by $(.)_i^l$. A function value at the left end point of a cell is denoted by $(.)_{i,L}^l$ and at the right end point by $(.)_{i,R}^l$.

4.2.1. Definitions and notations. With a multiple grid solution method in mind, we consider different levels of refinement in the partitioning of the open domain Ω and associate with a level of refinement $l \in \mathbb{Z}$, the partitioning Ω^l. We call the partitioning Ω^l the grid on level l. The grid on level l is determined by the grid points $x_i^l \in \overline{\Omega^l}$, $i \in \mathbb{Z}$ and it consists of the intervals $\Omega_i^l = (x_i^l, x_{i+1}^l)$, called cells. The grid Ω^l on level l is defined by

$$\Omega^l = \bigcup_i \Omega_i^l.$$

The set I is the set of index pairs, defined as

$$I = \{(i,l) \in \mathbb{Z}^2 \mid \exists \Omega_i^l \subset \Omega\}.$$

We also define sets of indices I^l by

$$I^l = \{i \in \mathbb{Z} \mid (i,l) \in I\}, \quad \forall l \in \{0, \ldots, L\}, \tag{4.2.1}$$

where L is the finest level present. Without loss of generality, the sets of indices I^l can be chosen so that on the coarsest level, $l = 0$, the smallest index is zero.

The grid points x_i^l are the boundaries of the cells on level l. A grid point which has only one neighbouring cell and which does not coincide with a physical boundary, is called a *green* boundary. We denote the left and right

end points of a cell Ω_i^l also by $x_{i,L}^l$ and $x_{i,R}^l$ respectively. Hence we have the notation

$$x_{i,L}^l = x_i^l,$$
$$x_{i,R}^l = x_{i+1}^l.$$

The size s_i^l of a cell Ω_i^l is defined by

$$s_i^l = x_{i,R}^l - x_{i,L}^l = x_{i+1}^l - x_i^l,$$

where we assume $x_{i+1}^l > x_i^l$. We consider the (typical multigrid) situation where cells on a grid are obtained by splitting the cell on the next coarser grid, and the size of a cell on level $l+1$ is approximately half the size of a cell on level l. The grid points do not have to be uniformly distributed over Ω. For all l, we denote by h_l the maximum

$$h_l = \max_{i \in I^l} s_i^l.$$

Furthermore, we consider a grid which is sufficiently smooth, i.e.

$$s_{i+1}^l = s_i^l + \mathcal{O}(h_l^p),$$

with $p > 1$ sufficiently large. Consequently we have for all $l \in \{0, \ldots, L-n\}$

$$h_{l+n} = 2^{-n} h_l + \mathcal{O}(h_l^p).$$

The refinements of a cell Ω_i^l are denoted by Ω_{2i}^{l+1} and Ω_{2i+1}^{l+1}. The part Ω_f^l of the grid Ω^l consists of all cells on level l which are refined. Thus, Ω^{l+1} consists of all refinements of the cells in Ω_f^l. We denote the non-refined part of Ω^l by Ω_c^l. The collection of all non-refined cells on all levels is called the *composite* grid and is denoted by Ω_c. Thus, we have for all l

$$\Omega^l = \Omega_f^l \cup \Omega_c^l,$$

and the composite grid is

$$\Omega_c = \bigcup_{l=0}^{L} \Omega_c^l.$$

Similar to (4.2.1), the sets of indices associated with Ω_f^l and Ω_c^l are denoted by I_f^l and I_c^l respectively. The set I_c of index pairs associated with the composite grid is defined as

$$I_c = \{(i,l) \in I \mid i \in I_c^l, \quad \forall l \in \{0, \ldots, L\}\}.$$

In Fig. 4.2.1 an example is given of different levels of refinement and the associated composite grid.

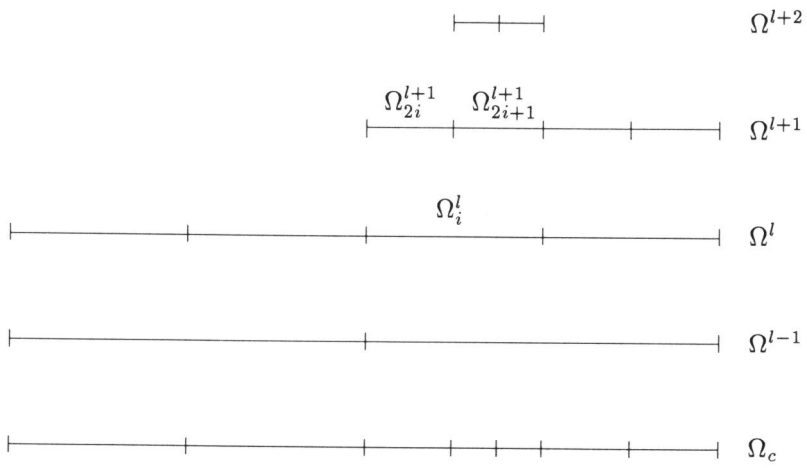


FIGURE 4.2.1. *Example of grids on different levels which result in a locally refined composite grid.*

For all $l \in \{0, \ldots, L\}$ we define two sets of piecewise constant functions on an arbitrary subdomain $\Omega^* \subset \Omega$ by

$$X^l(\Omega^*) = Y^l(\Omega^*) = \{u : \Omega^* \to \mathbb{R} \mid u(x) \text{ is constant in each } \Omega_i^l \cap \Omega^*\}.$$

We also define on Ω^* the sets $X(\Omega^*)$ and $Y(\Omega^*)$, which contain $X^l(\Omega^*)$ or $Y^l(\Omega^*)$ as well as all sufficiently smooth functions, for $l \in \{0, \ldots, L\}$. Hence, we have

$$X(\Omega^*) = C^k(\Omega^*) \cup \bigcup_{l=0}^{L} X^l(\Omega^*),$$

with k sufficiently large, and similar for $Y(\Omega^*)$. Notice that we have a nested sequence

$$X^l(\Omega^*) \subset X^{l+1}(\Omega^*) \subset X(\Omega^*),$$

if $\Omega^* \subset \Omega^{l+1} \subset \Omega^l$. The same applies to Y^l.

By u_i^l we denote the value of the cellwise constant function $u^l \in X^l(\Omega^*)$, in $\Omega_i^l \cap \Omega^*$. To simplify notation, we introduce restriction operators $\overline{R}^{l,l+n}$ and $R^{l,l+n}$, $n = 0, 1$. With $\Omega^{l+n} \subset \Omega^*$, the restriction operator $\overline{R}^{l,l+n} : X(\Omega^*) \to X^l(\Omega^{l+n})$ maps quite general functions to functions that are piecewise constant on cells in Ω^l, but only defined on Ω^{l+n}. This restriction is defined for

$l \in \{0, \ldots, L-n\}$ by

$$\{\overline{R}^{l,l+n} u\}_i^l = \frac{1}{s_i^l} \int_{x_{i,L}^l}^{x_{i,R}^l} u(x)\, dx, \quad \begin{cases} \forall i \in I^l, \text{ if } n = 0, \\ \forall i \in I_f^l, \text{ if } n = 1. \end{cases} \tag{4.2.2}$$

The other restriction operator, $R^{l,l+n}$, is related to (4.2.2) through the operator $S^{l,l+n} : X(\Omega^{l+n}) \to Y(\Omega^{l+n})$, for $n = 0, 1$, which is defined as

$$S^{l,l+n} u(x) = s_i^l u(x), \tag{4.2.3}$$

for all $u(x) \in X(\Omega_i^l \cap \Omega^{l+n})$, i.e. $i \in I^l$ if $n = 0$, or $i \in I_f^l$ if $n = 1$. We denote $S^{l,l}$ also by S^l. The restriction operator $R^{l,l+n} : Y(\Omega^{l+n}) \to Y^l(\Omega^{l+n})$, for $n = 0, 1$, is defined by

$$R^{l,l+n} = S^{l,l+n} \overline{R}^{l,l+n} (S^{l+n})^{-1}. \tag{4.2.4}$$

For example, we define the efflux function of cells on Ω^{l+1} by $F^{l+1} : X(\overline{\Omega}) \to Y^{l+1}(\Omega^{l+1})$, with for Ω_j^{l+1}

$$F_j^{l+1} = f(u(x_{j,R}^{l+1})) - f(u(x_{j,L}^{l+1})), \tag{4.2.5}$$

where $u(x)$ is continuous in the grid points x_j^{l+1}. Then, we have for all $i \in I_f^l$,

$$\begin{aligned} \{R^{l,l+1} F^{l+1}\}_i^l = F_{2i}^{l+1} + F_{2i+1}^{l+1} &= f(u(x_{2i+1,R}^{l+1})) - f(u(x_{2i,L}^{l+1})), \\ &= f(u(x_{i,R}^l)) - f(u(x_{i,L}^l)), \end{aligned}$$

which is the cell efflux function for the coarser cell of level l. We denote $\overline{R}^{l,l}$ also by $\overline{R}^l$ and $R^{l,l}$ by R^l.

We denote the restriction (mean value) of the exact solution $\overline{u} \in X(\Omega)$ of (4.1.1) by the cellwise constant function $\overline{u}^l \in X^l(\Omega^l)$:

$$\overline{u}^l = \overline{R}^l \overline{u}.$$

Approximations of $\overline{u}^l$ are denoted by u^l. Such an approximation can be considered as the restriction of some (possibly continuous) approximation u of $\overline{u}$. The $\overline{u}_i^l$ and u_i^l are so-called *cell states* of cell Ω_i^l. The error ϵ^l of an approximation u^l is defined by

$$\epsilon^l = \overline{u}^l - u^l. \tag{4.2.6}$$

Finally we define the piecewise constant functions on the composite grid, denoted by the subscript c. For example, the piecewise constant $u_c \in X_c(\Omega)$ is defined by

$$u_c(x) = u_i^l, \quad \forall x \in \Omega_i^l, \quad \forall (i,l) \in I_c. \tag{4.2.7}$$

We consider a finite volume discretisation of the steady conservation law of the form (4.1.1a). Integration of (4.1.1a) over a cell $\Omega_i^l \subset \Omega$, gives

$$\int_{\Omega_i^l} \frac{df(u(x))}{dx}\, dx = f(u(x_{i,R}^l)) - f(u(x_{i,L}^l)) = g_i^l s_i^l.$$

In the discretisation, the flux $f(u(x_{i,R}^l)) \equiv f_{i,R}^l$ is approximated through a numerical flux $F : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ by

$$F_{i,R}^l = F\left((u^L)_{i,R}^l, (u^R)_{i,R}^l\right).$$

The arguments $(u^L)_{i,R}^l$ and $(u^R)_{i,R}^l$ of the numerical flux function F represent the result of a left and right biased interpolation from u^l, as denoted by the superscripts L and R respectively. Similarly, the approximation of $f_{i,L}^l$ is denoted by $F_{i,L}^l$ and is given by

$$F_{i,L}^l = F\left((u^L)_{i,L}^l, (u^R)_{i,L}^l\right).$$

The function F may be considered to be e.g. an approximate Riemann solver. This may describe sophisticated upwind discretisations of the conservation law. Such upwind discretisations for conservation laws are considered in e.g. [6], [9] and [5].

In a local refinement context, some of the values u_i^l required in the computation of the left and right states, $(u^L)_{i,k}^l$ and $(u^R)_{i,k}^l$, $k = L, R$, may not be available. Therefore, we introduce the concept of *virtual states*. A virtual state v_i^l is an interpolant, generally computed from $u^{l-1}, \ldots, u^0$, which is used when Ω_i^l does not exist due to (the lack of) refinement of the grid. This occurs in the neighbourhood of a green boundary. For simplicity, we restrict ourselves to the situation where a virtual state v_i^l depends on u^{l-1} only. We introduce a *virtual cell*, $\omega_i^l \subset \Omega$, $i \notin I^l$, which is defined as the part of Ω which would be exactly Ω_i^l, if the grid would have been sufficiently refined. With the virtual cell ω_i^l we associate the virtual state v_i^l.

The discretisation on level l, defined on the subdomain Ω^{l+n}, is defined by the finite volume operator $N^{l,l+n} : X^l(\Omega^{l+n}) \to Y^l(\Omega^{l+n})$, $n = 0, 1$, given by

$$(4.2.8)\quad \{N^{l,l+n}(\overline{R}^{l,l+n} u^l; u^{l-1})\}_i^l = F_{i,R}^l - F_{i,L}^l, \quad \begin{cases} \forall i \in I^l, \text{ if } n = 0, \\ \forall i \in I_f^l, \text{ if } n = 1. \end{cases}$$

Here, u^{l-1} acts as a parameter in $N^{l,l+n}$, which determines the possible virtual states. We denote $N^{l,l}$ also by N^l. Following these notations, the discretisation for each cell of level $l+1$, as given by (4.2.5), is denoted by $R^{l+1}N$, hence

$$\{R^l N(u)\}_i^l = f(u(x_{i,R}^l)) - f(u(x_{i,L}^l)).$$

The set of equations on level l is written as

$$(4.2.9)\qquad N^l(u^l; u^{l-1}) = r^l,$$

where the right-hand side represents the source term.

For simplicity we restrict ourselves to the fully one-sided upwind computation of the flux and first-order accuracy. However, the framework described above can be used for more general numerical flux functions and higher-order accuracy. In fact it can be easily extended to a general second-order upwind discretisation of a conservation law in more space dimensions.

For the discretisation of our scalar problem (4.1.1a) we use the fully one-sided upwind computation of the flux, defined by the numerical flux function

$$F(u^L, u^R) = \begin{cases} f(u^L), & u^L, u^R \geq 0, \\ f(u^R), & u^L, u^R \leq 0, \\ 0, & \text{otherwise.} \end{cases}$$

For simplicity and without loss of generality, we consider a problem with $u^L, u^R \geq 0$ only. Since in this case $F(u^L, u^R)$ is a function of u^L only, we may redefine F by $F(u) = f(u)$ to shorten notation. The left state $(u^L)^l_{i,L}$ at the left boundary $x^l_{i,L}$ of Ω^l_i is given by

$$(u^L)^l_{i,L} = \begin{cases} u^l_{i-1}, & \text{if } (i-1) \in I^l, \\ v^l_{i-1}, & \text{if } x^l_{i,L} \text{ is a green boundary.} \end{cases} \tag{4.2.10}$$

This gives a first-order accurate discretisation if $(i-1) \in I^l$. We consider two interpolations for the virtual state v^{l+1}_{2i-1}: the first-order accurate interpolation $I1$, given by

$$v^{l+1}_{2i-1} = u^l_{i-1},$$

and the second-order accurate interpolation $I2$, given by

$$v^{l+1}_{2i-1} = \frac{3}{4} u^l_{i-1} + \frac{1}{4} u^l_i.$$

In Sec. 4.3 we show that the interpolation $I1$ leads to an $\mathcal{O}(1)$ and $I2$ to an $\mathcal{O}(h_l)$ local discretisation error.

4.2.2. The one-dimensional example problem. In our example problem, we choose the domain of definition $\Omega = (0,1)$, partitioned by cells of constant size $s^l_i = h_l$. For the flux function we take $f(u) = u^2$, and we choose a source $g(x) = 4x^3$. With a boundary condition $u(x_0) = 0$ at $x_0 = 0$, problem (4.1.1) does not have a unique solution. However, by imposing the extra condition

$$u(x) \geq 0, \quad \forall x \in \overline{\Omega},$$

we have the unique solution

$$u(x) = x^2, \quad \forall x \in \overline{\Omega}.$$

This example problem and its solution are sufficiently simple to allow a detailed analysis. Yet, the problem is sufficiently non-trivial to emulate the

properties of a fluid dynamics model such as the Euler equations, with respect to the aspects we focus on.

4.3. Analysis of accuracy

For the discretisation described above, we investigate the local truncation error and for the example problem, we study the error in the solution, both on a uniform grid (no local refinements) and on a locally refined grid. The local discretisation error is $\tau^l(\overline{u})$, where τ^l is the local truncation operator for N^l, defined by

$$\tau^l(u) = (S^l)^{-1}\left(N^l(\overline{R}^l u; \overline{R}^{l-1} u) - R^l N(u)\right), \tag{4.3.1}$$

where

$$\{R^l N(u)\}_i^l \equiv f(u(x_{i,R}^l)) - f(u(x_{i,L}^l)).$$

Application of the definitions of S^l and N^l in (4.2.3) and (4.2.8), gives for all $(i,l) \in I$,

$$\tau_i^l(u) = \frac{1}{s_i^l}(F_{i,R}^l - F_{i,L}^l - f_{i,R}^l + f_{i,L}^l), \tag{4.3.2}$$

with $F_{i,R}^l$, $F_{i,L}^l$ evaluated using $u^l = \overline{R}^l u$ and $f_{i,R}^l$ and $f_{i,L}^l$ evaluated using u. For our simple upwind discretisation, with

$$f_{i,L}^l = F(u(x_{i,L}^l)),$$
$$F_{i,L}^l = F(u_{i-1}^l),$$

we have for a sufficiently smooth $F(u)$

(4.3.3)

$$f_{i,L}^l = F_{i,L}^l + \left.\frac{\partial F}{\partial u}\right|_{u(x_{i,L}^l)} (u(x_{i,L}^l) - u_{i-1}^l) + \frac{1}{2}\left.\frac{\partial^2 F}{\partial u^2}\right|_{u(x_{i,L}^l)} (u(x_{i,L}^l) - u_{i-1}^l)^2 + \frac{1}{6}\left.\frac{\partial^3 F}{\partial u^3}\right|_{u(x_{i,L}^l)} (u(x_{i,L}^l) - u_{i-1}^l)^3 + \mathcal{O}\left((u(x_{i,L}^l) - u_{i-1}^l)^4\right),$$

and similar expressions for other fluxes. If u_{i-1}^l is not available due to missing refinements, v_{i-1}^l is used instead. Expression (4.3.3), possibly with virtual states, is used in the following subsections. We also use a Taylor series expansion of a sufficiently smooth $u(x)$ around $x_{i,L}^l$. This expansion is written as

$$u(x) = u_0 + u_1\xi + u_2\xi^2 + u_3\xi^3 + \mathcal{O}(\xi^4), \tag{4.3.4}$$

where $\xi = x - x_{i,L}^l$.

4.3.1. Locally uniform grid. Here we investigate the local truncation error and the global error for the example problem, defined in the previous section, on a locally uniform grid. A locally uniform grid is a collection of grid cells, for which no equation of the discretisation is involved with a virtual state. By definition, such an equation is obtained by application of the regular discretisation scheme.

Local truncation error. Here we assume that *all* equations are obtained by applying the regular discretisation scheme. On a sufficiently smooth grid, with a maximum size h_l of the cells, using (4.3.2), (4.3.4) and (4.3.3) we derive for $\tau^l(u)$ in Ω_i^l

$$\begin{aligned}\tau_i^l(u) = \frac{1}{s_i^l}\Bigg\{ & \left.\frac{\partial F}{\partial u}\right|_{u_0}(-u_2(s_i^l)^2 - \frac{1}{2}u_3(s_i^l)^3) \\ & + \frac{1}{2}\left.\frac{\partial^2 F}{\partial u^2}\right|_{u_0}(-u_1^2(s_i^l)^2 - \frac{4}{3}u_1u_2(s_i^l)^3) \\ & + \frac{1}{6}\left.\frac{\partial^3 F}{\partial u^3}\right|_{u_0}(-\frac{3}{4}u_1^3(s_i^l)^3)\Bigg\} + \mathcal{O}(h_l^3).\end{aligned} \tag{4.3.5}$$

Hence, for a differentiable solution of the continuous problem, the discretisation on a locally uniform grid is first-order accurate.

For the example problem, if $i > 0$, (4.3.2) gives

$$\tau_i^l(\bar{u})s_i^l = (\bar{u}_i^l)^2 - (\bar{u}_{i-1}^l)^2 - \bar{u}^2(x_{i,R}^l) + \bar{u}^2(x_{i,L}^l).$$

Because of the uniform grid, $s_i^l = h_l$ and hence $x_{i,L}^l = ih_l$. The restriction of the solution of the continuous problem is

$$\bar{u}_i^l = \frac{1}{h_l}\int_{ih_l}^{(i+1)h_l} x^2\,dx = h_l^2(i^2 + i + \frac{1}{3}), \tag{4.3.6}$$

and hence,

$$(\bar{u}_i^l)^2 = h_l^4(i^4 + 2i^3 + \frac{5}{3}i^2 + \frac{2}{3}i + \frac{1}{9}), \tag{4.3.7a}$$

$$(\bar{u}_{i-1}^l)^2 = h_l^4(i^4 - 2i^3 + \frac{5}{3}i^2 - \frac{2}{3}i + \frac{1}{9}). \tag{4.3.7b}$$

The square of the solution $\bar{u}(x)$ at the left and right end points of Ω_i^l respectively, is

$$\bar{u}^2(x_{i,L}^l) = h_l^4 i^4, \tag{4.3.8a}$$

$$\bar{u}^2(x_{i,R}^l) = h_l^4(i+1)^4. \tag{4.3.8b}$$

Hence, the local discretisation error is

$$\tau_i^l(\bar{u}) = -h_l^3(6i^2 + \frac{8}{3}i + 1), \quad i > 0. \tag{4.3.9}$$

A similar result may be obtained from (4.3.5), which for the example problem reduces to

(4.3.10)

$$\tau_i^l(u) = -\frac{1}{2}h_l \left\{ \left.\frac{\partial F}{\partial u}\right|_{u_0} \left.\frac{\partial^2 u}{\partial x^2}\right|_{x_{i,L}^l} + \left.\frac{\partial^2 F}{\partial u^2}\right|_{u_0} \left(\left.\frac{\partial u}{\partial x}\right|_{x_{i,L}^l} \right)^2 \right\} + \mathcal{O}(h_l^2).$$

For $i = 0$, the boundary condition at $x_0 = 0$ is implemented by defining the flux $F_{0,L}^l$ to be equal to the exact flux, $F_{0,L}^l = f_{0,L}^l = 0$. This gives the local truncation error

$$\tau_0^l(\bar{u}) = -\frac{8}{9}h_l^3. \tag{4.3.11}$$

Note that $\tau_0^l(u)$ is third-order if $u(0) = 0$, but generally it is zeroth-order. We define $0 < \alpha < 1$ through the relation $x = (i + \alpha)h_l$, for a given x and $i \to \infty$, while $h_l \to 0$. Then, the cellwise constant function $\tau^l(\bar{u}(x))$, defined by $\tau^l(\bar{u}(x)) = \tau_i^l(\bar{u})$, for $x \in \Omega_i^l$, is given by

$$\tau_i^l(\bar{u}(x)) = \begin{cases} -\frac{8}{9}h_l^3, & x \in \Omega_0^l, \\ -6x^2 h_l + (12\alpha x - \frac{8}{3}x)h_l^2 \\ \quad +(-6\alpha^2 + \frac{8}{3}\alpha - 1)h_l^3, & x \in \Omega^l \setminus \Omega_0^l. \end{cases}$$

Since α is bounded, the local discretisation error is first-order in the mesh width.

Error in the solution of example problem. The global discretisation error is defined by $\epsilon^l = \bar{u}^l - u^l$. Here u^l is the solution of (4.2.9), with $r^l = S^l g^l$, where $g^l = \overline{R}^l g$, with $F_{0,L}^l = 0$ and the extra condition $u_i^l > 0$. The discrete equation for a cell Ω_i^l is given by

$$\begin{aligned} (u_0^l)^2 &= g_0^l s_0^l, \\ (u_i^l)^2 - (u_{i-1}^l)^2 &= g_i^l s_i^l, \quad i > 0. \end{aligned}$$

By induction, it easily follows that

$$(u_i^l)^2 = \sum_{k=0}^{i} g_k^l s_k^l. \tag{4.3.12}$$

Because we consider a grid which is locally uniform everywhere, N^l is independent of its parameter u^{l-1}. By definition, $\bar{u}^l$ satisfies

$$N^l(\bar{u}^l) = S^l\left(g^l + \tau^l(\bar{u})\right),$$

with the conditions $F^l_{0,L} = 0$ and $\overline{u}^l_i > 0$. Similar to (4.3.12), the restriction $\overline{u}^l$ of the exact solution satisfies

$$(\overline{u}^l_i)^2 = \sum_{k=0}^{i} \left(g^l_k + \tau^l_k(\overline{u})\right) s^l_k. \tag{4.3.13}$$

Subtraction of (4.3.12) and (4.3.13) gives

$$(\overline{u}^l_i)^2 - (u^l_i)^2 = \sum_{k=0}^{i} \tau^l_k(\overline{u}) s^l_k. \tag{4.3.14}$$

With $s^l_i = h_l$, $\overline{u}(x) = x^2$, (4.3.9) and (4.3.11), this gives the cumulative truncation error up to and including cell Ω^l_i, given by

$$T^l_i \equiv \sum_{k=0}^{i} \tau^l_k(\overline{u}) s^l_k = h^4_l \left(\frac{1}{9} - \sum_{k=0}^{i} (6k^2 + \frac{8}{3}k + 1) \right). \tag{4.3.15}$$

Since

$$\sum_{k=0}^{i} k = \frac{1}{2}i^2 + \frac{1}{2}i, \qquad \sum_{k=0}^{i} k^2 = \frac{1}{3}i^3 + \frac{1}{2}i^2 + \frac{1}{6}i,$$

T^l_i, as defined by (4.3.15), is

$$T^l_i = -h^4_l(2i^3 + \frac{13}{3}i^2 + \frac{10}{3}i + \frac{8}{9}). \tag{4.3.16}$$

Expression (4.3.14) and the definition of ϵ^l (4.2.6) give

$$(\overline{u}^l_i)^2 - (u^l_i)^2 = (\overline{u}^l_i)^2 - (\overline{u}^l_i - \epsilon^l_i)^2 = T^l_i,$$

from which we obtain

$$\epsilon^l_i = \overline{u}^l_i \pm \overline{u}^l_i \sqrt{1 - T^l_i/(\overline{u}^l_i)^2}.$$

Since $T^l_i = \mathcal{O}(h_l)$ and both $u^l_i, \overline{u}^l_i > 0$, the minus sign applies. For small $T^l_i/(\overline{u}^l_i)^2$, the global discretisation error may be written as

$$\epsilon^l_i = \frac{1}{2}\frac{T^l_i}{\overline{u}^l_i} + \frac{1}{8}\frac{(T^l_i)^2}{(\overline{u}^l_i)^3} + \frac{1}{16}\frac{(T^l_i)^3}{(\overline{u}^l_i)^5} + \mathcal{O}\left(\frac{(T^l_i)^4}{(\overline{u}^l_i)^7}\right). \tag{4.3.17}$$

Substitution of the expressions (4.3.6) and (4.3.16) for $\overline{u}^l_i$ and T^l_i respectively, gives for the example problem, and small $T^l_i/(\overline{u}^l_i)^2$,

$$\epsilon^l_i = -h^2_l(i + \frac{2}{3} + \mathcal{O}(i^{-2})) + \mathcal{O}\left(\frac{(T^l_i)^4}{(\overline{u}^l_i)^7}\right).$$

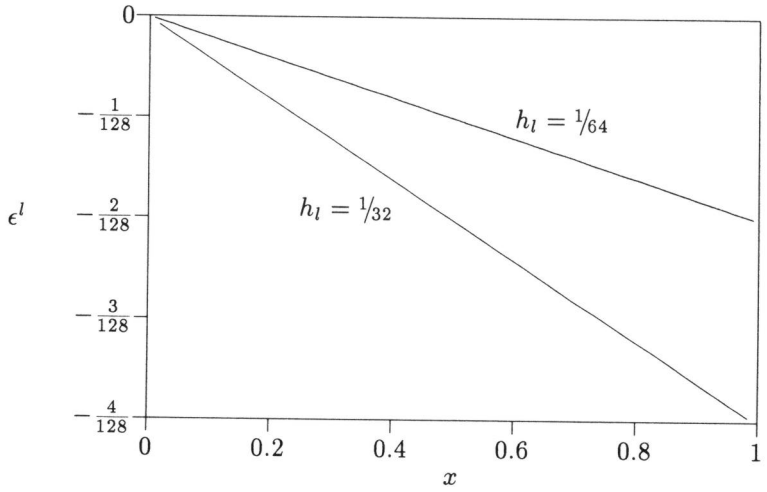


FIGURE 4.3.1. *Error of the solution, for the example problem on uniform grid.*

The cellwise constant error $\epsilon^l(x)$ is given by

$$\epsilon^l(x) = -xh_l + (-\frac{2}{3} + \alpha)h_l^2 + \mathcal{O}(h_l^4), \tag{4.3.18}$$

where again α is defined by $ih_l = x - \alpha h_l$, $0 < \alpha < 1$. In Fig. 4.3.1 a linear interpolation of the error ϵ^l as obtained by numerically computing the solution of the discretisation, is shown for two grids, one with $h_l = 1/32$ and one with $h_l = 1/64$. Equation (4.3.18) is in agreement with the results in Fig. 4.3.1. As we expected, the global discretisation error is first-order.

4.3.2. Locally non-uniform grid. A grid is called locally non-uniform in a cell Ω_i^l, if the discrete equation for Ω_i^l involves a virtual state. This situation results in an irregular discretisation scheme. First we look at the local and global error of the irregular discretisation, using the first-order accurate interpolation $I1$ to compute a virtual state. After this we study interpolation $I2$.

Local truncation error for interpolation $I1$. We consider a coarse grid, denoted by Ω^l and a fine one, denoted by Ω^{l+1}. The grid Ω^l covers the domain Ω completely and the grid Ω^{l+1} covers only a subdomain. Without loss of generality, we take the green boundary of Ω^{l+1} at x_g and $\overline{\Omega^{l+1}} = [x_g, 1]$. Furthermore, we define for the green boundary

$$x_g = x_m^l = x_{2m}^{l+1}.$$

The discrete equation for the cell on the locally non-uniform grid involves the virtual state v_{2m-1}^{l+1}. By interpolation $I1$ this virtual state is simply given by

$$v_{2m-1}^{l+1} = u_{m-1}^{l}.$$

With (4.3.3) and (4.3.4) we find the expansion of the local truncation error

$$\begin{aligned}\tau_{2m}^{l+1}(u) = \frac{1}{s_{2m}^{l+1}} \Bigg\{ & \left.\frac{\partial F}{\partial u}\right|_{u_0} \left(\frac{1}{2}u_1 s_{2m}^{l+1} - 2u_2 (s_{2m}^{l+1})^2 + \frac{5}{4}u_3 (s_{2m}^{l+1})^3\right) \\ & + \frac{1}{2}\left.\frac{\partial^2 F}{\partial u^2}\right|_{u_0} \left(-\frac{7}{4}u_1^2 (s_{2m}^{l+1})^2 + u_1 u_2 (s_{2m}^{l+1})^3\right) \\ & + \frac{1}{6}\left.\frac{\partial^3 F}{\partial u^3}\right|_{u_0} \left(\frac{1}{8}u_1^3 (s_{2m}^{l+1})^3\right) \Bigg\} + \mathcal{O}(h_l^3).\end{aligned}$$

This shows that interpolation $I1$ to compute a virtual state gives an $\mathcal{O}(1)$ local truncation error in the locally non-uniform grid cell. For our example problem the local discretisation error $\tau_{2m}^{l+1}(\bar{u})$ in Ω_{2m}^{l+1} is

$$\tau_{2m}^{l+1}(\bar{u}) s_{2m}^{l+1} = (\bar{u}_{2m}^{l+1})^2 - (\bar{u}_{m-1}^{l})^2 - \bar{u}^2(x_{2m,R}^{l+1}) + \bar{u}^2(x_{2m,L}^{l+1}). \tag{4.3.19}$$

If we substitute expressions similar to (4.3.7) and (4.3.8), into (4.3.19), we find for the example problem that the local discretisation error of the equation involving the virtual state, is also $\mathcal{O}(1)$, viz.

$$\tau_{2m}^{l+1}(\bar{u}) = 2x_g^3 - 11x_g^2 h_{l+1} + 2x_g h_{l+1}^2 - \frac{8}{3}h_{l+1}^3. \tag{4.3.20}$$

Error in solution of example problem for interpolation $I1$. The global error on the non-refined part Ω_c^l is the same as for the uniform grid situation. For the example problem the global error in the green-boundary cell Ω_{2m}^{l+1} can be found from the equation

$$(\bar{u}_{2m}^{l+1})^2 - (u_{2m}^{l+1})^2 = T_{m-1}^{l} + \tau_{2m}^{l+1}(\bar{u}) s_{2m}^{l+1} = T_{2m}^{l+1}, \tag{4.3.21}$$

where T_{2m}^{l+1} is the cumulative truncation error as in (4.3.15). For the coarse-grid contribution T_{m-1}^{l} to the cumulative local error, we have from (4.3.15)

$$T_{m-1}^{l} = -4x_g^3 h_{l+1} + \frac{20}{3}x_g^2 h_{l+1}^2 - \frac{16}{3}x_g h_{l+1}^3 + \frac{16}{9}h_{l+1}^4. \tag{4.3.22}$$

Furthermore, (4.3.20), (4.3.21) and (4.3.22) give

$$T_{2m}^{l+1} = -2x_g^3 h_{l+1} - \frac{13}{3}x_g^2 h_{l+1}^2 - \frac{10}{3}x_g h_{l+1}^3 - \frac{8}{9}h_{l+1}^4.$$

With expression (4.3.17) for the error, and proceeding along the same lines as for the uniform grid, we find the global error in Ω_{2m}^{l+1}

$$\epsilon_{2m}^{l+1} = -x_g h_{l+1} - \frac{2}{3}h_{l+1}^2 + \mathcal{O}(h_{l+1}^4). \tag{4.3.23}$$

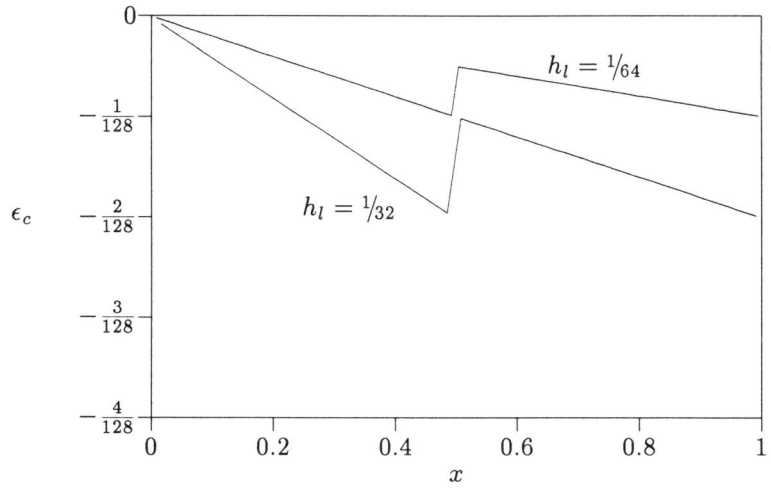


FIGURE 4.3.2. *Error of the solution for the example problem on the composite grid, using interpolation $I1$ for the virtual state and with green boundary $x_g = 1/2$.*

It appears that the solution in Ω_{2m}^{l+1} is first-order accurate, although the discrete equation for Ω_{2m}^{l+1} is zeroth-order consistent. For $x_g = 1/2$, a linear interpolation of the global error ϵ_c on the composite grid is shown in Fig. 4.3.2, for two values of the mesh size: $h_l = 1/32$ and $h_l = 1/64$. Equation (4.3.23) is in agreement with Fig. 4.3.2. Notice that for interpolation $I1$ we have

$$\epsilon_{2m}^{l+1} - \epsilon_{m-1}^{l} = \mathcal{O}(h_l). \tag{4.3.24}$$

We assume that for $k = l, l+1$ the error can be written as the asymptotic expansion

$$\epsilon^k = h_L \overline{R}^k \epsilon(x) + \mathcal{O}(h_l^2),$$

with h_L the maximum mesh width at the highest level and $\epsilon(x) \in X(\Omega)$, independent of h_L. From (4.3.24) it appears that $\epsilon(x)$ is discontinuous at $x = x_g$, which is also observed in Fig. 4.3.2.

Local truncation error for interpolation $I2$. In this subsection we consider the more accurate interpolation $I2$. The introduction of this higher-order accurate interpolation is not primarily required to obtain a higher-order error, as was demonstrated in the previous subsection (see also Sec. 2.5.2 and [8]). $I2$ is necessary to obtain a discretisation which has the same order of consistency everywhere. This could make local refinement an effective tool to reduce the local truncation errors. A technique for local refinement based on the local discretisation error, requires that the local truncation error of the equations

for the finer cells is smaller than for the corresponding coarse-grid cells. This is generally not the case when these equations are inconsistent. In practice, when using adaptive refinement, such inconsistency may result in small, highly refined regions (islands) in Ω. (see Sec. 4.5.4).

The more accurate interpolation used to obtain the virtual state v_{2m-1}^{l+1}, is given by the interpolation

$$v_{2m-1}^{l+1} = \frac{3}{4}u_{m-1}^{l} + \frac{1}{4}u_{m}^{l},$$

where u_m^l is the restriction of the fine-grid solution

$$u_m^l = \{\overline{R}^{l,l+1}u^{l+1}\}_m^l.$$

Following the same lines as in the previous section, we find the expansion of the local truncation error of N^{l+1}

$$\begin{aligned}\tau_{2m}^{l+1}(u) = \frac{1}{s_{2m}^{l+1}} \Bigg\{ & \left.\frac{\partial F}{\partial u}\right|_{u_0} (-2u_2(s_{2m}^{l+1})^2 + \frac{1}{4}u_3(s_{2m}^{l+1})^3) \\ & + \frac{1}{2}\left.\frac{\partial^2 F}{\partial u^2}\right|_{u_0} (-u_1^2(s_{2m}^{l+1})^2 - \frac{1}{3}u_1u_2(s_{2m}^{l+1})^3) \\ & + \frac{1}{6}\left.\frac{\partial^3 F}{\partial u^3}\right|_{u_0} (-\frac{3}{4}u_1^3(s_{2m}^{l+1})^3) \Bigg\} + \mathcal{O}(h_l^3).\end{aligned}$$

Since $s_{2m}^{l+1} = \mathcal{O}(h_{l+1}^2)$, application of interpolation $I2$ to obtain a virtual state gives an $\mathcal{O}(h_{l+1})$ local truncation error for the equation derived for the locally non-uniform grid cell. For the example problem with solution $\overline{u}(x) = x^2$, we find the local discretisation error

$$\tau_{2m}^{l+1}(\overline{u}) = -8x_g^2h_{l+1} - \frac{2}{3}x_gh_{l+1}^2 - \frac{8}{3}h_{l+1}^3.$$

Error in solution of example problem for interpolation I2. Again, on Ω_c^l the global discretisation error is the same as for the uniform grid. Only the computation of the global error in Ω_{2m}^{l+1} is a little more laborious. The state u_{2m}^{l+1} in the cell adjacent to the green boundary, is part of the solution of the nonlinear system

$$\begin{cases} (u_{2m}^{l+1})^2 - (\frac{3}{4}u_{m-1}^l + \frac{1}{4}u_m^l)^2 = g_{2m}^{l+1}s_{2m}^{l+1}, \\ u_m^l = \frac{1}{2}(u_{2m}^{l+1} + u_{2m+1}^{l+1}), \\ (u_{2m+1}^{l+1})^2 - (u_{2m}^{l+1})^2 = g_{2m+1}^{l+1}s_{2m+1}^{l+1}. \end{cases}$$

Since u_{m-1}^l and the right-hand sides $g_{2m}^{l+1}s_{2m}^{l+1}$ and $g_{2m+1}^{l+1}s_{2m+1}^{l+1}$ are known, it follows that u_{2m}^{l+1} can be considered as a function of the independent variables

u^l_{m-1}, $g^{l+1}_{2m}s^{l+1}_{2m}$ and $g^{l+1}_{2m+1}s^{l+1}_{2m+1}$:

$$u^{l+1}_{2m} = U(u^l_{m-1},\, g^{l+1}_{2m}s^{l+1}_{2m},\, g^{l+1}_{2m+1}s^{l+1}_{2m+1}), \tag{4.3.25}$$

where the function $U = U(a, r_1, r_2)$ satisfies

$$\begin{cases} U^2 - (\frac{3}{4}a + \frac{1}{4}b)^2 = r_1, \\ b = \frac{1}{2}(U + c), \\ c^2 - U^2 = r_2. \end{cases} \tag{4.3.26}$$

The solution of this system is not given in closed form, but we can use (4.3.26) to estimate the error ϵ^{l+1}_{2m}. Using the definition of the local truncation error, we can write the restriction of the exact solution as

$$\bar{u}^{l+1}_{2m} = U\left(\bar{u}^l_{m-1},\, (g^{l+1}_{2m} + \tau^{l+1}_{2m}(\bar{u}))s^{l+1}_{2m},\, (g^{l+1}_{2m+1} + \tau^{l+1}_{2m+1}(\bar{u}))s^{l+1}_{2m+1}\right). \tag{4.3.27}$$

Assuming that U is sufficiently differentiable and bounded, with bounded derivatives as $h_l \to 0$, it follows from (4.3.25) and (4.3.27) that the error in Ω^{l+1}_{2m} can be written as

$$\begin{aligned} \epsilon^{l+1}_{2m} = \bar{u}^{l+1}_{2m} - u^{l+1}_{2m} = {} & \frac{\partial U}{\partial a}\,\epsilon^l_{m-1} + \frac{\partial U}{\partial r_1}\,\tau^{l+1}_{2m}(\bar{u})s^{l+1}_{2m} \\ & + \frac{\partial U}{\partial r_2}\,\tau^{l+1}_{2m+1}(\bar{u})s^{l+1}_{2m+1} + \frac{1}{2}\,\frac{\partial^2 U}{\partial a^2}\,(\epsilon^l_{m-1})^2 + \mathcal{O}(h_l^3), \end{aligned} \tag{4.3.28}$$

where $\frac{\partial U}{\partial a}$, $\frac{\partial U}{\partial i}r_1$ and $\frac{\partial U}{\partial r_2}$ are evaluated at

$$a = \bar{u}^l_{m-1}, \tag{4.3.29a}$$

$$r_1 = (g^{l+1}_{2m} + \tau^{l+1}_{2m}(\bar{u}))s^{l+1}_{2m}, \tag{4.3.29b}$$

$$r_2 = (g^{l+1}_{2m+1} + \tau^{l+1}_{2m+1}(\bar{u}))s^{l+1}_{2m+1}. \tag{4.3.29c}$$

The local discretisation error is $\mathcal{O}(h_l)$ and for twice differentiable U, we find for $h_l \to 0$

$$\epsilon^{l+1}_{2m} = \frac{\partial U}{\partial a}\,\epsilon^l_{m-1} + \mathcal{O}(h_l^2).$$

Since $\epsilon^l_{m-1} = \mathcal{O}(h_l)$, the approximation u^{l+1}_{2m} of $\bar{u}^{l+1}_{2m}$ is first-order accurate. In Fig. 4.3.3 the derivatives of $\bar{u}^{l+1}_{2m} = U(a, r_1, r_2)$, with arguments given by (4.3.29), are given as a function of h_l for the example problem, i.e. with solution $\bar{u}(x) = x^2$ and with a green boundary at $x_g = 1/2$.

Assume an asymptotic expansion of U

$$U = U_0 + U_1 h_l + \mathcal{O}(h_l^2),$$

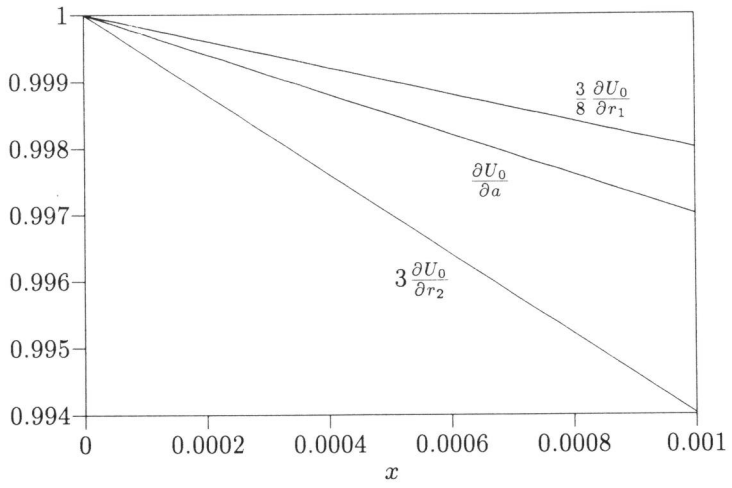


FIGURE 4.3.3. *Derivatives of* $U = \bar{u}_{2m}^{l+1}$ *of the example problem, at green boundary* $x_g = 1/2$.

with U_0 and U_1 independent of h_l. Then for $h_l \to 0$ the error ϵ_{2m}^{l+1} in (4.3.28) can be written as

$$\begin{aligned}\epsilon_{2m}^{l+1} = &\left(\frac{\partial U_0}{\partial a} + \frac{\partial U_1}{\partial a}\, h_l\right)\epsilon_{m-1}^{l} + \frac{\partial U_0}{\partial r_1}\,\tau_{2m}^{l+1}(\bar{u})s_{2m}^{l+1} \\ &+ \frac{\partial U_0}{\partial r_2}\,\tau_{2m+1}^{l+1}(\bar{u})s_{2m+1}^{l+1} + \frac{1}{2}\,\frac{\partial^2 U_0}{\partial a^2}\,(\epsilon_{m-1}^{l})^2 + \mathcal{O}(h_l^3),\end{aligned} \tag{4.3.30}$$

with all derivatives evaluated at (4.3.29). These derivatives are found by solving (4.3.26), differentiated with respect to a, r_1 or r_2. In the limit $h_l \to 0$, these derivatives are

$$\begin{aligned}&\frac{\partial U_0}{\partial a} = 1, \qquad \frac{\partial^2 U_0}{\partial a^2} = 0, \qquad \frac{\partial U_0}{\partial r_1} = \frac{2}{3}\frac{1}{x_g^2}, \\ &\frac{\partial U_1}{\partial a} = -\frac{3}{2}\frac{1}{x_g}, \qquad \frac{\partial U_0}{\partial r_2} = \frac{1}{12}\frac{1}{x_g^2}.\end{aligned} \tag{4.3.31}$$

These results are in agreement with Fig. 4.3.3, which shows the first derivatives of U_0, for $x_g = 1/2$ and small h_l. Substitution of (4.3.31) into (4.3.30) gives for the example problem

$$\epsilon_{2m}^{l+1} = -2x_g h_{l+1} + \frac{3}{2}h_{l+1}^2 + \mathcal{O}(h_l^3).$$

In Fig. 4.3.4 the error is given for two grids, one with $h_l = 1/32$ and one with $h_l = 1/64$, and $x_g = 1/2$. Note that in this particular case the discretisation with interpolation $I1$ (cf. Fig. 4.3.2), yields a more accurate solution than $I2$.

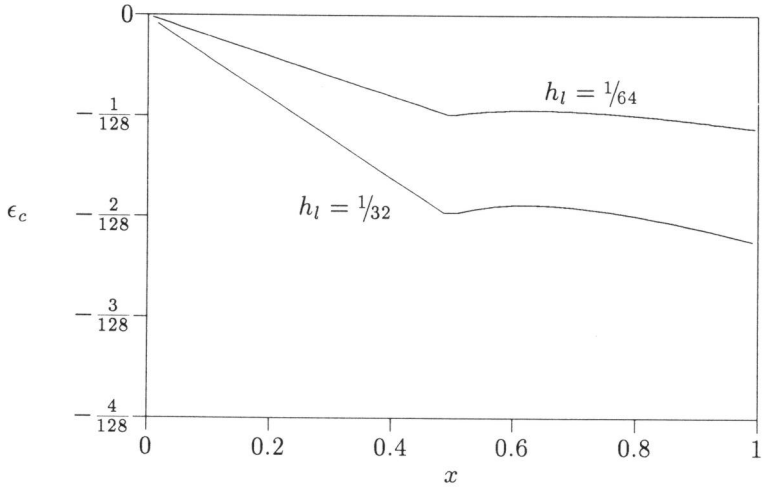


FIGURE 4.3.4. *Error of the solution for the example problem on the composite grid, using interpolation $I2$ for the virtual state and with green boundary $x_g = 1/2$.*

For both interpolations the numerical solution is first-order accurate. Opposed to interpolation $I1$, interpolation $I2$ gives

$$\epsilon_{2m}^{l+1} - \epsilon_{m-1}^{l} = \mathcal{O}(h_l^2).$$

This means that $\epsilon(x)$ is continuous at $x = x_g$. Interpolation $I2$ to compute the virtual state apparently gives a smoother error and hence a smoother approximation of the exact (smooth) solution. A further investigation however, reveals that $\epsilon(x)$ is not differentiable in $x = x_g$, which is also observed in Fig. 4.3.4. We remind that the use of $I2$ is primarily meant to obtain a higher-order accurate local truncation error, which can be used in constructing a solution-adaptive, locally refined grid.

4.4. A-posteriori estimation of the one-dimensional local discretisation error

In the previous example the exact solution and hence the exact local truncation error are known. In practice we want to estimate the local discretisation error, while we only have the solution of the discrete problem as an approximation of the exact solution. Now we consider estimating the local truncation error for a uniform grid and for a locally refined grid. The procedure described in this section is intended to be used (and is actually used) in a multigrid context. It is based on a classical truncation error extrapolation, [2], [3]. Instead of estimating the local truncation error by extrapolation, it may also be estimated by estimating expression (4.3.3) for each grid point. For a

first-order accurate discretisation this would imply estimating $\partial u/\partial x$, $\partial F/\partial u$ and $\partial^2 F/\partial u^2$ at each grid point. However, we aim at application of our a-posteriori estimation of the local discretisation error to a set of simultaneous conservation laws, defined in two or even three space dimensions. For these kinds of problems, estimating the local truncation error by approximating these derivatives would be very expensive, and even more so for higher-order accurate discretisations. Therefore, we propose to estimate the local discretisation error by application of an extrapolation technique that will be studied in Sec. 4.4.2 and 4.4.3.

Since our goal is the use of (an estimate of) the local discretisation error in a refinement criterion, we focus on the situation where the local truncation error is of the same order everywhere. Hence, where an irregular discretisation scheme is applied, we focus on the interpolation $I2$.

4.4.1. Preliminaries. On the sequence of grids, Ω^l, $l = 0, \ldots, L$, we consider a sequence of locally nested discretisations, denoted by

$$N^l(u^l; u^{l-1}) = r^l, \quad \forall l \in \{0, \ldots, L\}. \tag{4.4.1}$$

Again, u^{l-1} merely acts as a parameter, since solving the equations for a level l will leave u^{l-1} unchanged. This parameter is superfluous on level $l = 0$, and on any level for which the grid covers Ω completely. As usual for the multigrid FAS procedure (see [1]), the right-hand side of (4.4.1) is defined as

$$r^l = \begin{cases} S^l g^l, \text{ on } \Omega_c^l, \\ N^{l,l+1}(\overline{R}^{l,l+1} u^{l+1}; u^{l-1}) - R^{l,l+1}\left(N^{l+1}(u^{l+1}; u^l) - r^{l+1}\right), \text{ on } \Omega_f^l. \end{cases}$$

The solution of (4.4.1) is u_c, defined by (4.2.7) and is associated with the cells on the composite grid Ω_c. The approximation on Ω^l is $\overline{R}^l u$ and on Ω_f^l it is $\overline{R}^{l,l+1} u$. The sequence of (locally) nested discretisations allows us to define the so-called relative discretisation error, similar to the local discretisation error, defined by (4.3.1). The relative discretisation error is denoted by $\tau_{l+1}^l(\overline{u}^{l+1})$, where $\tau_{l+1}^l : X^{l+1}(\Omega^{l+1}) \to X^l(\Omega^{l+1})$ is defined by

$$\tau_{l+1}^l(u^{l+1}) = (S^{l,l+1})^{-1}\left(N^{l,l+1}(\overline{R}^{l,l+1} u^{l+1}; u^{l-1}) - R^{l,l+1} N^{l+1}(u^{l+1}; u^l)\right).$$

With the definition of the local truncation error (4.3.1) and $R^{l,l+1}$ defined by (4.2.4), the relative truncation error can be written as

(4.4.2)

$$\begin{aligned}\tau_{l+1}^{l}(\overline{R}^{l+1}u) &= (S^{l,l+1})^{-1}N^{l,l+1}(\overline{R}^{l,l+1}u;\overline{R}^{l-1,l}u)\\ &\quad - \overline{R}^{l,l+1}(S^{l+1})^{-1}N^{l+1}(\overline{R}^{l+1}u;\overline{R}^{l}u)\\ &= (S^{l,l+1})^{-1}\left(N^{l,l+1}(\overline{R}^{l,l+1}u;\overline{R}^{l-1,l}u) - R^{l,l+1}N^{,l+1}(u)\right)\\ &\quad - \overline{R}^{l,l+1}(S^{l+1})^{-1}N^{l+1}(\overline{R}^{l+1}u;\overline{R}^{l}u) + \overline{R}^{l,l+1}(S^{l+1})^{-1}N^{,l+1}(u)\\ &= \overline{R}^{l,l+1}\tau^{l}(u) - \overline{R}^{l,l+1}\tau^{l+1}(u).\end{aligned}$$

We assume that the global error and the local error can both be written as an asymptotic expansion in the mesh width h_l. By $\epsilon = \epsilon(x) \in X(\Omega)$ we denote a function independent of h_L, so that for a sufficiently smooth grid, the $\mathcal{O}(h_L^q)$ global error on a level l can be written as

$$\epsilon^l = \overline{R}^l\left(h_L^q\epsilon + \mathcal{O}(h_L^{q+1})\right),$$

where h_L is the mesh width at the finest level present. The solution of the system $N^l(u^l;u^{l-1}) = r^l$ is written as

$$u^l = \overline{u}^l - \epsilon^l = \overline{R}^l(\overline{u} - h_L^q\epsilon) + \mathcal{O}(h_L^{q+1}).$$

For the asymptotic expansion of the local truncation error, we introduce $\tau : X(\Omega) \to X(\Omega)$ by

$$\tau(u,u',\dots) = \tau(u(x),\, u'(x),\dots),$$

where the primes denote differentiation with respect to x. We assume that the local truncation error τ^l of N^l can be written as the expansion

$$\tau^l(u) = h_l^p\overline{R}^l\tau(u,u',\dots) + \mathcal{O}(h_l^{p+1}).$$

For example, for the model problem (4.1.1) discretised with the fully one-sided upwind flux, according to (4.3.10), τ is given by

$$\tau(u,u',u'') = -\frac{1}{2}\left\{\frac{\partial f}{\partial u}\,\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 f}{\partial u^2}\left(\frac{\partial u}{\partial x}\right)^2\right\} = -\frac{1}{2}\,\frac{\partial^2 f}{\partial x^2}\,. \tag{4.4.3}$$

For derivatives we use the notation

$$\delta_u\tau(u,u',\dots) = \frac{\partial\tau}{\partial u}\,(u,u',\dots).$$

For $u = \bar{u} - h_L^q \epsilon + \mathcal{O}(h_L^{q+1})$ and sufficiently smooth $\bar{u}(x)$ and $\epsilon(x)$, the error $\tau(u, u', \dots)$ is related to $\tau(\bar{u}, \bar{u}', \dots)$ through the asymptotic expansion, by

$$\begin{aligned} \tau(u, u', \dots) = \tau(\bar{u}, \bar{u}', \dots) &- h_L^q \epsilon \delta_u \tau(\bar{u}, \bar{u}', \dots) \\ &- h_L^q \epsilon' \delta_{u'} \tau(\bar{u}, \bar{u}', \dots) - \cdots + \mathcal{O}(h_L^{q+1}) + \mathcal{O}(h_L^{2q}). \end{aligned} \tag{4.4.4}$$

The preliminaries introduced here will be used to show that our estimates of the local discretisation error are sufficiently accurate, both in a locally uniform and locally non-uniform situation, provided that some smoothness conditions are satisfied. Both situations are described in detail in the following two subsections. Results for the example problem are given in section 4.4.4.

4.4.2. Estimating on a uniform grid. The restriction of the local truncation error $\tau^{l+1}(u)$ of $N^{l+1}(\bar{R}^{l+1}u)$ on a uniform grid is

$$\begin{aligned} \bar{R}^l \tau^{l+1}(u) &= \bar{R}^l \left(h_{l+1}^p \tau(u) + \mathcal{O}(h_l^{p+1}) \right) \\ &= 2^{-p} h_l^p \bar{R}^l \tau(u) + \mathcal{O}(h_l^{p+1}) \\ &= 2^{-p} \tau^l(u) + \mathcal{O}(h_l^{p+1}). \end{aligned} \tag{4.4.5}$$

Hence, we find

$$\tau^l(u) = 2^p \bar{R}^l \tau^{l+1}(u) + \mathcal{O}(h_l^{p+1}). \tag{4.4.6}$$

Substitution of (4.4.6) into (4.4.2) gives

$$\begin{aligned} \tau_{l+1}^l(\bar{R}^{l+1}u) &= \tau^l(u) - \bar{R}^l \tau^{l+1}(u) \\ &= (2^p - 1)\bar{R}^l \tau^{l+1}(u) + \mathcal{O}(h_l^{p+1}). \end{aligned}$$

Assuming the existence of an interpolation operator $P^{l+1} : X(\Omega^{l+1}) \to X^{l+1}(\Omega^{l+1})$, to interpolate a coarse-grid function, which for any sufficiently smooth $u \in X(\Omega^{l+1})$ satisfies

$$P^{l+1} \bar{R}^{l,l+1} \bar{R}^{l+1} u = \bar{R}^{l+1} u + \mathcal{O}(h_l). \tag{4.4.7}$$

With this interpolation and with (4.4.4), where $u = \bar{u} - h_L^q \epsilon + \mathcal{O}(h_L^{q+1})$, we find

$$\begin{aligned} \frac{1}{2^p - 1} P^{l+1} \tau_{l+1}^l(\bar{R}^{l+1}u) &= P^{l+1} \bar{R}^l \tau^{l+1}(u) + \mathcal{O}(h_l^{p+1}) \\ &= \tau^{l+1}(u) + \mathcal{O}(h_l^{p+1}) \\ &= \tau^{l+1}(\bar{u}) - h_l^p h_L^q \bar{R}^{l+1} \left(\epsilon \delta_u \tau(\bar{u}, \bar{u}', \dots) \right. \\ &\quad \left. + \epsilon' \delta_{u'} \tau(\bar{u}, \bar{u}', \dots) + \cdots \right) \\ &\quad + \mathcal{O}(h_l^p h_L^{q+1}) + \mathcal{O}(h_l^p h_L^{2q}) + \mathcal{O}(h_l^{p+1}). \end{aligned} \tag{4.4.8}$$

Hence, with an interpolation operator which satisfies (4.4.7) and provided ϵ and $\bar{u}$ are sufficiently smooth, the estimate

$$\tilde{\tau}^{l+1}(u^{l+1}) \equiv \frac{1}{2^p - 1} P^{l+1} \tau_{l+1}^l(u^{l+1}) = \tau^l(\bar{u}) + \mathcal{O}(h_l^t),$$

holds with $t = \min\{p + q, p + 1\}$. According to (4.4.3) and (4.4.4), for the one-sided upwind discretisation of (4.1.1) we investigate here, it suffices to have differentiable $\bar{u}(x)$ and $\epsilon(x)$.

4.4.3. Estimating on a locally non-uniform grid. For a locally non-uniform grid, we wish to be able to estimate the local truncation error in a similar way as described in the previous section. As noticed earlier, this is difficult because we have to find asymptotic expansions for the local error of an irregular discretisation scheme. In this section we describe how the local error may be split into two parts. This is done so that one part can be estimated by extrapolation. The other part is approximated by approximating the differences that appear in the expression for this local error.

In a locally non-uniform grid situation, (4.4.5) does not hold, since the discretisation in the neighbourhood of a green boundary differs from the regular discretisation. We split the local discretisation error into two parts. One part, $\tau_u^l(u)$, is the local truncation error in the case where each cell is part of a locally uniform grid (all neighbours exist). The other part, $\tau_n^l(u)$, is a perturbation due to the local refinements and virtual states (e.g., fine-grid cells have been deleted and hence some neighbours do not exist any more). This splitting gives

$$\tau^l(u) = \tau_u^l(u) + \tau_n^l(u). \tag{4.4.9}$$

The part $\tau_n^l(u)$ is zero in cells which are part of a locally uniform grid (if the regular discretisation scheme is employed). For the first-order accurate interpolation in (4.2.10) and fully one-sided upwind computation of the numerical flux, $\tau_n^l(u)$ is unequal zero, only when $x_{i,L}^l$ is a green boundary. We describe estimates for both τ_u^l and τ_n^l in detail, in the next subsections. We consider a locally refined composite grid, so that a coarse grid is locally uniform.

The non-uniform part. The non-uniform part of the local truncation error may be considered as a perturbation of the local truncation error on a locally uniform grid. In the locally non-uniform grid cells, $\tau_n^{l+1}(u)$ results from the difference between a virtual state v_{2m-1}^{l+1} in ω_{2m-1}^{l+1} and the corresponding state u_{2m-1}^{l+1}, if Ω_{2m-1}^{l+1} would exist. We have for $\tau_n^{l+1}(u)$

$$\{\tau_n^{l+1}(u)\}_{2m}^{l+1} = \tau_{2m}^{l+1}(u) - \{\tau_u^{l+1}(u)\}_{2m}^{l+1}.$$

By (4.3.3) and (4.3.2) we have, for $x_{2m,L}^{l+1}$ a green boundary,

$$\{\tau_n^{l+1}\}_{2m}^{l+1} = \frac{1}{s_{2m}^{l+1}} \left\{ \left.\frac{\partial F}{\partial u}\right|_{u(x_{2m,L}^{l+1})} (u_{2m-1}^{l+1} - v_{2m-1}^{l+1}) \right.$$
$$\left. + \mathcal{O}\left(\left(u(x_{2m,L}^{l+1}) - v_{2m-1}^{l+1} \right)^2 - \left(u(x_{2m,L}^{l+1}) - u_{2m-1}^{l+1} \right)^2 \right) \right\}.$$

The second-order accurate interpolation $I2$ to obtain the virtual state v_{2m-1}^{l+1} gives for the non-uniform part $\{\tau_n^{l+1}(u)\}_{2m}^{l+1}$ in Ω_{2m}^{l+1}

$$\{\tau_n^{l+1}(u)\}_{2m}^{l+1} = -\frac{1}{2} s_{2m}^{l+1} \left.\frac{\partial F}{\partial u}\right|_{u(x_{2m,L}^{l+1})} \left.\frac{\partial^2 u}{\partial x^2}\right|_{x_{2m,L}^{l+1}} + \mathcal{O}(h_l^2).$$

For a second-order accurate approximation of $\tau_n^{l+1}(u)$, it suffices to approximate $\partial F/\partial u$ at $u(x_{i,L}^l)$ and $\partial^2 u/\partial x^2$ at $x_{i,L}^l$ with first-order accuracy. Let v_{2m-1}^{l+1} be the virtual state obtained by interpolation $I2$ of the restriction of $u(x)$. In addition, let a function $\tilde{u}(x)$ satisfy

$$\frac{\int_{\omega_{2m-1}^{l+1}} \tilde{u}(x)\, dx}{\int_{\omega_{2m-1}^{l+1}} dx} = v_{2m-1}^{l+1}, \tag{4.4.10a}$$

$$\tilde{u}(x) = u(x), \quad x \geq x_{2m,L}^{l+1}, \tag{4.4.10b}$$

where ω_{2m-1}^{l+1} is a virtual cell. If the solution for a regular discretisation scheme is the restriction of a differentiable function, then there exists a $\tilde{u}(x)$ which satisfies (4.4.10) and which is also differentiable. Differentiability is ensured, since for a given v_{2m-1}^{l+1} the scheme may be considered a regular scheme. The regular discretisation is shown to have a solution, which is the restriction of a differentiable function. For this $\tilde{u}(x)$ we have

$$\frac{u_{2m+1}^{l+1} - 2u_{2m}^{l+1} + v_{2m-1}^{l+1}}{(s_{2m}^{l+1})^2} = \left.\frac{\partial^2 \tilde{u}}{\partial x^2}\right|_{x_{2m,L}^{l+1}} + \mathcal{O}(h_l) = \lim_{\xi \downarrow x_{2m,L}^{l+1}} \left.\frac{\partial^2 u}{\partial x^2}\right|_{\xi} + \mathcal{O}(h_l).$$

Furthermore, it can be easily shown that

$$\left.\frac{\partial F}{\partial u}\right|_{u_{2m}^{l+1}} = \left.\frac{\partial F}{\partial u}\right|_{u(x_{2m,L}^{l+1})} + \mathcal{O}(h_l),$$

for $u(x)$ continuous on $\overline{\Omega_{2m}^{l+1}}$.

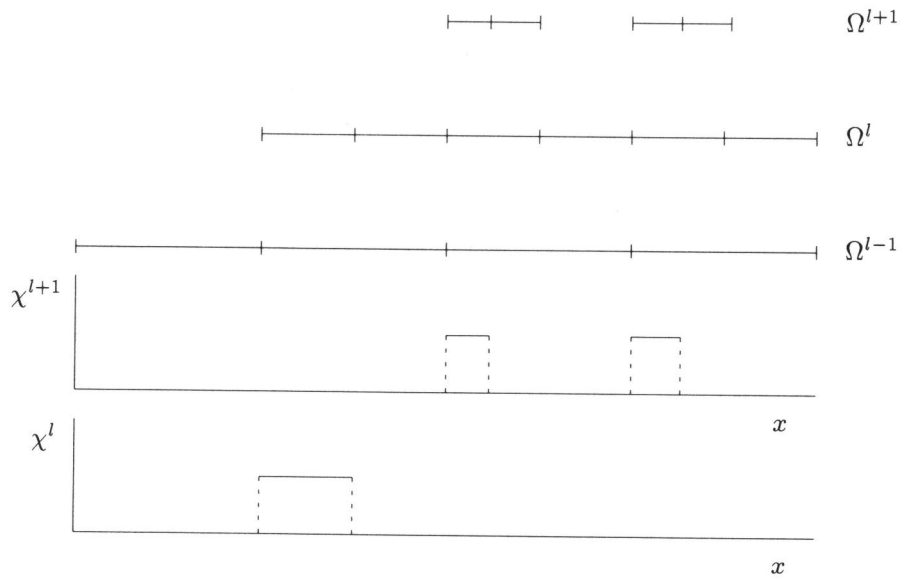


FIGURE 4.4.1. *Example of characteristic function on different levels of refinement.*

For $x^{l+1}_{2m,L}$ a green boundary, we have an a-posteriori estimate for $\tau^{l+1}_n(u)$, denoted by $\tilde{\tau}^{l+1}_n(u^{l+1}; u^l)$, given by

(4.4.11)

$$\begin{aligned}\{\tilde{\tau}^{l+1}_n(u^{l+1}; u^l)\}^{l+1}_{2m} &\equiv -\frac{1}{2s^{l+1}_{2m}} \left.\frac{\partial F}{\partial u}\right|_{u^{l+1}_{2m}} (u^{l+1}_{2m+1} - 2u^{l+1}_{2m} + v^{l+1}_{2m-1}) \\ &= \{\tau^{l+1}_n(u)\}^{l+1}_{2m} + \mathcal{O}(h_l^2).\end{aligned}$$

Otherwise, $\tilde{\tau}^{l+1}_n$ is zero.

The uniform part. Let the characteristic function $\chi^l \in X^l(\Omega^l)$ be defined by

(4.4.12) $$\chi^l(x) = \begin{cases} 1, & \text{if } x \in \Omega^l_i \text{ and } x^l_{i,L} \text{ is a green boundary,} \\ 0, & \text{everywhere else and } x \in \Omega^l. \end{cases}$$

An example of characteristic functions associated with the locally refined grid, is shown in Fig. 4.4.1. Similar to the locally uniform case, we introduce functions to be used in asymptotic expansions in terms of h_l, for both τ^l_u and τ^l_n, given by

$$\begin{aligned}\tau_u(u, u', \dots) &= \tau_u(u(x), u'(x), \dots) \\ \tau_n(u, u', \dots) &= \tau_n(u(x), u'(x), \dots).\end{aligned}$$

An asymptotic expansion of the local truncation error is then written as

$$\tau^l(u) = \overline{R}^l \Big\{ h_l^p \tau_u(u, u', \dots) + \mathcal{O}(h_l^{p+1}) + \chi^l \left(h_l^r \tau_n(u, u', \dots) + \mathcal{O}(h_l^{r+1}) \right) \Big\}, \tag{4.4.13}$$

where τ_u and τ_n are independent of h_l. The assumption that the coarse grid Ω^l is locally uniform, implies

$$\overline{R}^{l,l+1} \chi^l \tau_n(u, u', \dots) = 0.$$

The local truncation error on the refined part Ω_f^l of the coarse grid is then given by

$$\begin{aligned}
\hat{\tau}^l(u) &\equiv \overline{R}^{l,l+1} \tau^l(u) \\
&= \overline{R}^{l,l+1} \Big\{ h_l^p \tau_u(u, u', \dots) + \mathcal{O}(h_l^{p+1}) + \chi^l \left(h_l^r \tau_n(u, u', \dots) + \mathcal{O}(h_l^{r+1}) \right) \Big\} \\
&= h_l^p \overline{R}^{l,l+1} \tau_u(u, u', \dots) + \mathcal{O}(h_l^{p+1}).
\end{aligned}$$

The restriction of the local truncation error on the fine grid, can be written as

$$\begin{aligned}
\overline{R}^{l,l+1} \tau^{l+1}(u) &= h_{l+1}^p \overline{R}^{l,l+1} \tau_u(u, u', \dots) + \overline{R}^{l,l+1} \tau_n^{l+1}(u) + \mathcal{O}(h_l^{p+1}) \\
&= 2^{-p} \hat{\tau}^l(u) + \overline{R}^{l,l+1} \tau_n^{l+1}(u) + \mathcal{O}(h_l^{p+1}).
\end{aligned}$$

From this we see that the local truncation error on the refined part of the coarse grid then may be written as

$$\hat{\tau}^l(u) = 2^p \overline{R}^{l,l+1} \tau^{l+1}(u) - 2^p \overline{R}^{l,l+1} \tau_n^{l+1}(u) + \mathcal{O}(h_l^{p+1}).$$

The relative truncation error $\tau_{l+1}^l(u)$ as given by (4.4.2) can be written as

$$\begin{aligned}
\tau_{l+1}^l(\overline{R}^{l+1} u) &= \hat{\tau}^l(u) - \overline{R}^{l,l+1} \tau^{l+1}(u) \\
&= 2^p \overline{R}^{l,l+1} \tau^{l+1}(u) - 2^p \overline{R}^{l,l+1} \tau_n^{l+1}(u) - \overline{R}^{l,l+1} \tau^{l+1}(u) + \mathcal{O}(h_l^{p+1}) \\
&= (2^p - 1) \overline{R}^{l,l+1} \tau^{l+1}(u) - 2^p \overline{R}^{l,l+1} \tau_n^{l+1}(u) + \mathcal{O}(h_l^{p+1}).
\end{aligned} \tag{4.4.14}$$

From (4.4.13) and the definition of χ^l in (4.4.12), it is easily seen that $\tau^{l+1}(u)$ cannot be considered as the restriction of some smooth function. An interpolation operator which satisfies (4.4.7) for a smooth function, cannot be applied effectively to approximate $\tau^{l+1}(u)$ from (4.4.14). This would result in errors of $\mathcal{O}(h_l^p)$ in the approximation of the local truncation error, which itself is $\mathcal{O}(h_l^p)$. The part τ_u^{l+1} may be considered as the restriction of a smooth

function $h_{l+1}^p \tau_u(u(x)) + \mathcal{O}(h_l^{p+1})$. For the upwind discretisation of the conservation law (4.1.1a) τ_u is given by (4.4.3). Introduction of the splitting (4.4.9) in (4.4.14), gives

$$\tau_{l+1}^l(\overline{R}^{l+1}u) = (2^p - 1)\overline{R}^{l,l+1}\tau_u^{l+1}(u) - \overline{R}^{l,l+1}\tau_n^{l+1}(u) + \mathcal{O}(h_l^{p+1}).$$

Interpolation by using P^{l+1}, which satisfies (4.4.7), division by $2^p - 1$ and reordering gives

$$\frac{1}{2^p - 1} P^{l+1}\left(\tau_{l+1}^l(\overline{R}^{l+1}u) + \overline{R}^{l,l+1}\tau_n^{l+1}(u)\right) = \tau_u^{l+1}(u) + \mathcal{O}(h_l^{p+1}).$$

If $x_{2m,L}^{l+1}$ is a green boundary and the estimate of the non-uniform part is given by (4.4.11) with sufficiently smooth $\epsilon(x)$ and $\overline{u}(x)$, then we have, similar to (4.4.8), an estimate of the non-uniform part of the local truncation error given by

(4.4.15)

$$\begin{aligned}\tilde{\tau}_u^{l+1}(u^{l+1}; u^l) &\equiv \frac{1}{2^p - 1} P^{l+1}\left(\tau_{l+1}^l(u^{l+1}) + \overline{R}^{l,l+1}\tilde{\tau}_n^{l+1}(u^{l+1}; u^l)\right) \\ &= \tau_u^{l+1}(\overline{u}) + \mathcal{O}(h_l^t),\end{aligned}$$

where $t = \min\{p+1, p+q, 2\}$. From (4.4.3) and (4.4.4) it follows that $\overline{u}(x)$ and $\epsilon(x)$ have to be differentiable to ensure that (4.4.15) is true.

4.4.4. Estimating the local discretisation error for an example problem. In this section we investigate the a-posteriori estimation of the local error, for the example problem introduced in Sec. 4.2.2.

Locally uniform grid. In the model problem we use a first-order discretisation and we have a first-order accurate solution, $p = 1$. With the expression (4.3.9) for the exact local discretisation error of the example problem, we find

$$\tau_{2i}^{l+1}(\overline{u}) = -h_{l+1}^3(24i^2 + \frac{16}{3}i + 1), \tag{4.4.16a}$$

$$\tau_{2i+1}^{l+1}(\overline{u}) = -h_{l+1}^3(24i^2 + \frac{88}{3}i + \frac{29}{3}). \tag{4.4.16b}$$

For the problem and discretisation considered here, it suffices to take for P^{l+1} the cellwise constant interpolation

$$\{P^{l+1}(u^l)\}_{2i}^{l+1} = \{P^{l+1}(u^l)\}_{2i+1}^{l+1} = u_i^l,$$

for $i \in I_f^l$. This interpolation satisfies (4.4.7). For the example problem, with solution $\overline{u}(x) = x^2$, the coarse-grid operator N^l, acting on the restriction of the fine-grid solution of the discrete equations, gives

$$\{N^l(\overline{R}^l u^{l+1})\}_i^l = h_{l+1}^4(64i^3 + 48i^2 + 32i + \mathcal{O}(i^0)).$$

For the restriction of the fine-grid operator, acting on the fine-grid solution we have

$$\{R^l N^{l+1}(u^{l+1})\}_i^l = h_{l+1}^4(64i^3 + 96i^2 + 64i + \mathcal{O}(i^0)).$$

Hence, the relative truncation error is

$$\{\tau_{l+1}^l(u^{l+1})\}_i^l = -h_{l+1}^3(24i^2 + 16i + \mathcal{O}(i^0)).$$

The cellwise constant interpolation gives the estimate on the fine grid

$$\tilde{\tau}_{2i}^{l+1}(u^{l+1}) = \tilde{\tau}_{2i+1}^{l+1}(u^{l+1}) = -h_{l+1}^3(24i^2 + 16i + \mathcal{O}(i^0)).$$

As expected, this differs $\mathcal{O}(h_l^2)$ from the estimate (4.4.16).

Locally non-uniform grid. The local discretisation error in the locally non-uniform grid cell Ω_{2m}^{l+1} is

$$\tau_{2m}^{l+1}(\bar{u}) = -8x_g^2 h_{l+1} - \frac{2}{3}x_g h_{l+1}^2 - \frac{8}{3}h_{l+1}^3. \tag{4.4.17a}$$

For its neighbour we have

$$\tau_{2m+1}^{l+1}(\bar{u}) = -6x_g^2 h_{l+1} - \frac{44}{3}x_g h_{l+1}^2 - \frac{29}{3}h_{l+1}^3. \tag{4.4.17b}$$

In this locally non-uniform case, the local discretisation error is estimated by estimating τ_u^l and τ_n^l. By application of (4.4.11), we find for the non-uniform part

$$\{\tilde{\tau}_n^{l+1}(u^{l+1}; u^l)\}_{2m}^{l+1} = -2x_g^2 + \mathcal{O}(h_l^2).$$

Furthermore, we find

$$\{N^l(u^l; u^{l-1})\}_m^l = 8x_g^3 h_{l+1} + \frac{44}{3}x_g^2 h_{l+1}^2 + \mathcal{O}(h_l^3),$$

and

$$\{R^{l,l+1}N^{l+1}(u^{l+1}; u^l)\}_m^l = 8x_g^3 h_{l+1} + 24x_g^2 h_{l+1}^2 + \mathcal{O}(h_l^3).$$

The estimate of the relative truncation error is

$$\{\tau_{l+1}^l(u^{l+1})\}_m^l = -\frac{14}{3}x_g^2 h_{l+1} + \mathcal{O}(h_l^2).$$

Then we have with the cellwise interpolation P^{l+1} and $p = 1$

$$\begin{aligned}\{\tilde{\tau}_u^{l+1}(u^{l+1}; u^l)\}_{2m}^{l+1} &= \{\tilde{\tau}_u^{l+1}(u^{l+1}; u^l)\}_{2m+1}^{l+1} \\ &= \{\tau_{l+1}^l(u^{l+1})\}_m^l + \{\bar{R}^{l,l+1}\tilde{\tau}_n^{l+1}(u^{l+1}; u^l)\}_m^l \\ &= -\frac{17}{3}x_g^2 h_{l+1} + \mathcal{O}(h_l^2),\end{aligned}$$

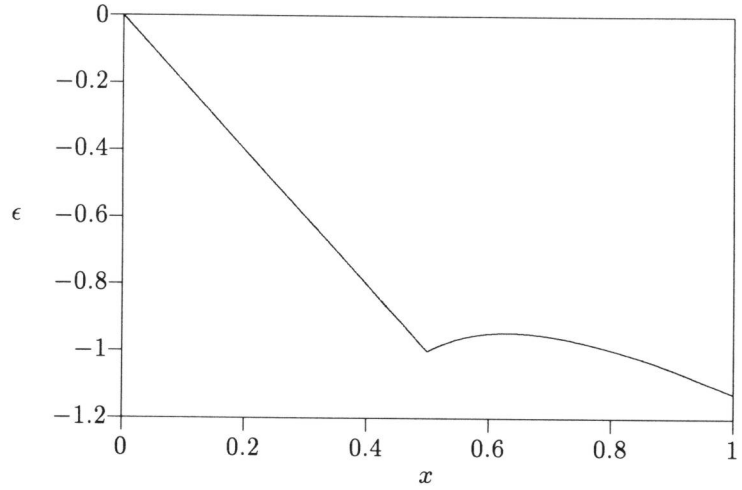


FIGURE 4.4.2. *The function $\epsilon(x)$ for the example problem and green boundary $x_g = 1/2$.*

and the estimation of the local truncation error is

$$\{\tilde{\tau}_u^{l+1} + \tilde{\tau}_n^{l+1}\}_{2m}^{l+1} = -\frac{23}{3}x_g^2 h_{l+1} + \mathcal{O}(h_l^2),$$
$$\{\tilde{\tau}_u^{l+1} + \tilde{\tau}_n^{l+1}\}_{2m+1}^{l+1} = -\frac{17}{3}x_g^2 h_{l+1} + \mathcal{O}(h_l^2).$$

Clearly, these are $\mathcal{O}(h_l)$ accurate approximations of the exact local truncation errors, given by (4.4.17), which itself is $\mathcal{O}(h_l)$. This inaccuracy is caused by the fact that ϵ, which is used in the asymptotic expansion of the error, is not sufficiently smooth. In Fig. 4.4.2 ϵ is shown for the green boundary at $x_g = 1/2$. As already noted, ϵ is not differentiable in x_g. The inaccuracy of this estimate is in full agreement with the theory developed in Sec. 4.4.3.

An accurate estimate of the local discretisation error can be made, by replacing ϵ with a sufficiently smooth $\tilde{\epsilon}$, and use this in the estimation of the uniform part $\tilde{\tau}_u^{l+1}$. Replacing ϵ is the same as replacing the function u used in the asymptotic expansion of u^l, with a sufficiently smooth and first-order accurate $\tilde{u}$, which satisfies (4.4.10). Then, the restriction of $\tilde{u}$ may be used in the computation of τ_{l+1}^l.

This restriction of $\tilde{u}$ is easily found by extrapolation of the solution u^{l+1}. We use (4.4.10) for the mean of $\tilde{u}$ at ω_{2m-1}^{l+1}. The mean at ω_{2m-2}^{l+1} is found from linear extrapolation by

$$\tilde{u}_{2m-2}^{l+1} = \frac{\int_{\omega_{2m-2}^{l+1}} \tilde{u}\,dx}{\int_{\omega_{2m-2}^{l+1}} dx} = 2v_{2m-1}^{l+1} - u_{2m}^{l+1}.$$

In Ω^l_{m-1} we use, instead of u^l_{m-1}, the value of $\frac{1}{2}(\tilde{u}^{l+1}_{2m-2} + \tilde{u}^{l+1}_{2m-1})$. For $\tilde{u}^{l+1}$ we find

$$\tilde{u}^{l+1}_{2m-1} = v^{l+1}_{2m-1} = x_g^2 + x_g h_{l+1} - \frac{1}{6}h^2_{l+1} + \mathcal{O}(h_l^3),$$
$$\tilde{u}^{l+1}_{2m-2} = x_g^2 - x_g h_{l+1} + \frac{5}{6}h^2_{l+1} + \mathcal{O}(h_l^3).$$

Then, we also have

$$\tilde{u}^l_{m-1} = \frac{1}{2}(\tilde{u}^{l+1}_{2m-2} + \tilde{u}^{l+1}_{2m-1}) = x_g^2 + \frac{1}{3}h^2_{l+1} + \mathcal{O}(h_l^3).$$

The restriction of the fine-grid operator, acting on $\tilde{u}^{l+1}$, is

$$\begin{aligned}\{R^l N^{l+1}(\tilde{u}^{l+1}; \tilde{u}^l)\}^l_m &= (u^{l+1}_{2m+1})^2 - (v^{l+1}_{2m-1})^2 \\ &= 8x_g^3 h_{l+1} + 24x_g^2 h^2_{l+1} + \mathcal{O}(h_l^3).\end{aligned}$$

Furthermore, we have for the coarse-grid operator, acting on the restriction of $\tilde{u}^{l+1}$

$$\begin{aligned}\{N^l(\overline{R}^{l,l+1}\tilde{u}^{l+1}; \tilde{u}^{l+1})\}^l_m &= (\tilde{u}^l_m)^2 - (\tilde{u}^l_{m-1})^2 \\ &= \frac{1}{4}(u^{l+1}_{2m} + u^{l+1}_{2m+1})^2 - (\tilde{u}^l_{m-1})^2 \\ &= 8x_g^3 h_{l+1} + 14x_g^2 h^2_{l+1} + \mathcal{O}(h_l^3).\end{aligned}$$

Then, the relative local discretisation error is

$$\{\tau^l_{l+1}(\tilde{u}^{l+1})\}^l_m = -5x_g^2 h_{l+1} + \mathcal{O}(h_l^2).$$

From this and with P^{l+1} the cellwise constant interpolation and $p = 1$, we find

$$\begin{aligned}\{\tilde{\tau}^{l+1}_u(\tilde{u}^{l+1}; \tilde{u}^l)\}^{l+1}_{2m} &= \{\tilde{\tau}^{l+1}_u(\tilde{u}^{l+1}; \tilde{u}^l)\}^{l+1}_{2m+1} \\ &= \{\tau^l_{l+1}(\tilde{u}^{l+1})\}^l_m + \{\overline{R}^{l,l+1}\tilde{\tau}^{l+1}_n(\tilde{u}^{l+1}; \tilde{u}^l)\}^l_m \\ &= -6x_g^2 h_{l+1} + \mathcal{O}(h_l^2).\end{aligned}$$

Then, the estimates of the local discretisation errors on Ω^{l+1} are

$$\begin{aligned}\{\tilde{\tau}^{l+1}_u + \tilde{\tau}^{l+1}_n\}^{l+1}_{2m} &= -6x_g^2 h_{l+1} - 2x_g^2 h_{l+1} + \mathcal{O}(h_l^2) \\ &= -8x_g^2 h_{l+1} + \mathcal{O}(h_l^2), \\ \{\tilde{\tau}^{l+1}_u + \tilde{\tau}^{l+1}_n\}^{l+1}_{2m+1} &= -6x_g^2 h_{l+1} + \mathcal{O}(h_l^2).\end{aligned}$$

These clearly are $\mathcal{O}(h_l^2)$ accurate approximations of the local discretisation errors given by (4.4.17).

4.5. Extension of the estimation to two space dimensions

In this section we consider the extension of the procedure developed in the previous section to two space dimensions. We consider a grid in physical domain obtained by an affine mapping $M^l = M$. An analysis for more general mappings would probably require an invariant description in general coordinates, similar as e.g. in [4].

For the present analysis we need expressions of the error of the discretisation due to the computation of the virtual states. For an affine mapping M and a function $q \in \overline{X}(\Omega)$ we have

$$q^l_{i,j} = \{\overline{R}^l q\}^l_{i,j} = \frac{1}{h_l^2} \int_{\widetilde{\Omega}^l_{i,j}} q \, d\widetilde{\Omega}.$$

This greatly simplifies the analysis, since we can do it completely in computational (ξ, η) coordinates. We first establish the accuracy of the virtual states for first-order and second-order accurate discretisations, for the virtual states defined by (2.5.1)–(2.5.4).

We assume a sufficiently smooth function q, and a Taylor series expansion, given by

$$\begin{aligned} q = {} & q_0 + q_1\xi + q_2\eta + q_3\xi^2 + q_4\xi\eta + q_5\eta^2 \\ & + q_6\xi^3 + q_7\xi^2\eta + q_8\xi\eta^2 + q_9\eta^3 + \dots, \end{aligned}$$

around the centre of the cell $\Omega^l_{i,j}$.

We consider the equations for a cell $\Omega^{l+1}_{2i+1,2j+1}$ where $\partial\Omega^{l+1}_{2i+1,2j+1,E} \subset \partial\Omega^{l+1}_g$. The virtual state $v^{l+1}_{2i+2,2j+1}$ required for these equations, is an approximation of the mean of q on $\omega^{l+1}_{2i+2,2j+1}$. This mean can be expressed as

$$\begin{aligned} \frac{1}{h^2_{l+1}} \int_{\tilde{\omega}^{l+1}_{2i+2,2j+1}} q \, d\widetilde{\Omega} = {} & q_0 + \frac{3}{2}h_{l+1}q_1 + \frac{1}{2}h_{l+1}q_2 \\ & + \frac{7}{3}h^2_{l+1}q_3 + \frac{3}{4}h^2_{l+1}q_4 + \frac{1}{3}h^2_{l+1}q_5 + \frac{15}{4}h^3_{l+1}q_6 \\ & + \frac{7}{6}h^3_{l+1}q_7 + \frac{1}{2}h^3_{l+1}q_8 + \frac{1}{4}h^3_{l+1}q_9 + \mathcal{O}(h^4_{l+1}). \end{aligned} \tag{4.5.1}$$

The mean state in (4.5.1) equals $q^{l+1}_{2i+2,2j+1}$ in the situation $(2i+2, 2j+1) \in I^l$, since $\omega^{l+1}_{2i+2,2j+1}$ is the part of $\widehat{\Omega}$ that would be $\Omega^{l+1}_{2i+2,2j+1}$ if the grid would be sufficiently refined (cf. (2.3.17)). In [7] it is shown that $q^{l+1}_{2i+2,2j+1}$ (and hence (4.5.1)) satisfies the requirements for consistency (2.4.38), for the reconstructions discussed in Sec. 2.3.5. For any virtual state $v^{l+1}_{2i+2,2j+1}$ that differs $\mathcal{O}(h^p_{l+1})$, $p = 1, 2$ from (4.5.1), the reconstructions discussed in Sec. 2.3.5 do not satisfy the consistency requirements and introduce an $\mathcal{O}(h^{p-1}_{l+1})$ error

in the equations for $\Omega^{l+1}_{2i+1,2j+1}$. However, the numerical flux will be a pth-order accurate approximation of the mean value of the exact flux, and hence pth-order consistency in the weak sense is obtained (cf. 2.4.6).

We study the accuracy of the virtual states, for both weak consistency and consistency in the usual sense, and for both first-order and second-order discretisations.

4.5.1. Virtual states for weak consistency. For pth-order weak consistency we only require a pth-order accurate approximation of the mean flux across a cell face. We consider virtual states for both first-order and second-order weak consistency.

First-order. The formula for a first-order accurate virtual state $v^{l+1}_{2i+2,2j+1}$ is given by (2.5.1), i.e.

$$v^{l+1}_{2i+2,2j+1} = q^l_{i+1,j}, \tag{2.5.1}$$

and schematically represented in Fig. 4.5.1. For this virtual state we find with

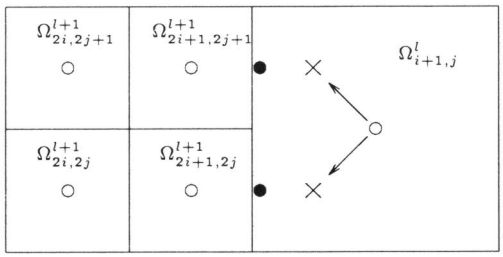


FIGURE 4.5.1. *Virtual state for first-order weak consistency on a locally refined grid;* ○*: available state;* ●*: left or right state;* ×*: virtual state.*

the Taylor series expansion

$$v^{l+1}_{2i+2,2j+1} = q_0 + 2h_{l+1}q_1 + \mathcal{O}(h^2_{l+1}). \tag{4.5.2}$$

This virtual state differs $\mathcal{O}(h_{l+1})$ from (4.5.1). Hence, it yields a zeroth-order error for the equations. However, since the virtual state *is* $\mathcal{O}(h_l)$ accurate, the reconstruction gives a first-order accurate virtual state, which yields a flux computation which is first-order accurate. This implies first-order weak consistency.

Second-order. A similar situation for second-order weak consistency is found, if the virtual states are computed with the second-order accurate formulae (2.5.2), i.e.

$$v^{l+1}_{2i+2,2j+1} = \frac{3}{4} q^{l+1}_{i+1,j} + \frac{1}{4} q^{l+1}_{i,j+1}, \tag{2.5.2a}$$

$$v^{l+1}_{2i+3,2j+1} = \frac{3}{4} q^{l+1}_{i+1,j} + \frac{1}{4} q^{l+1}_{i+2,j+1}. \tag{2.5.2b}$$

This is schematically presented in Fig. 4.5.2. Again, the virtual state $v^{l+1}_{2i+2,2j+1}$

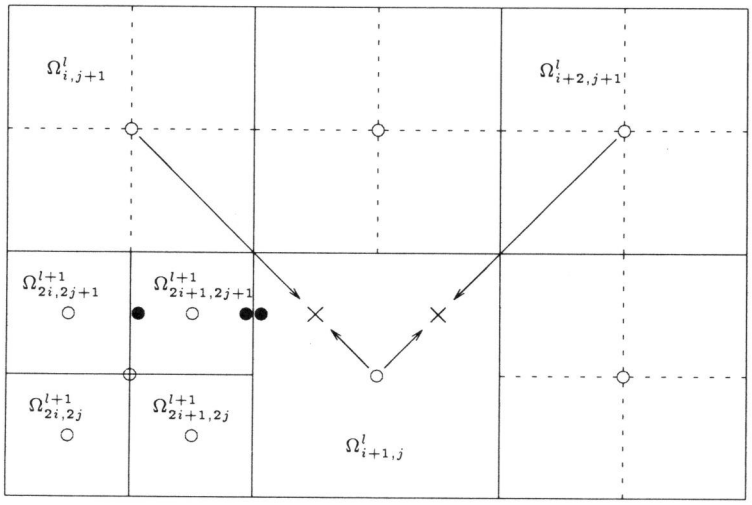


FIGURE 4.5.2. *Virtual states for second-order weak consistency and first-order consistency on a locally refined grid;* ○*: available state;* ●*: left or right state;* ×*: virtual state.*

is an approximation of the mean of q on $\omega^{l+1}_{2i+2,2j+1}$, given in (4.5.1). For the virtual state computed in this way, we have

$$\begin{aligned} v^{l+1}_{2i+2,2j+1} = q_0 + \frac{3}{2} h_{l+1} q_1 + \frac{1}{2} h_{l+1} q_2 + \frac{10}{3} h^2_{l+1} q_3 \\ + \frac{4}{3} h^2_{l+1} q_5 + \mathcal{O}(h^3_{l+1}). \end{aligned} \tag{4.5.3}$$

A similar result is obtained for $v^{l+1}_{2i+3,2j+1}$. Apparently, this is a second-order accurate approximation of (4.5.1). Similar to the first-order weakly consistent situation, the consistency requirements (2.4.38) are not satisfied for $p = 2$. However, the flux is second-order accurate, hence (2.5.2) yield second-order weak consistency. Formulae (2.5.2) are two of a number of possible choices for

second-order accurate virtual states. We have chosen these for their relative *compactness* and their *symmetry*. They are symmetric with respect to the diagonal through the centres of $\Omega^l_{i,j+1}$ and $\Omega^l_{i+1,j}$. A virtual state required for e.g. $\Omega^{l+1}_{2i+2,2j+2}$, with $\partial\Omega_{2i+2,2j+2,S} \subset \partial\Omega^{l+1}_g$, exactly results in (2.5.2a).

4.5.2. Virtual states for consistency. The requirements to be satisfied for a consistent discretisation, are given by (2.4.38). We consider both first-order and second-order consistency. A pth-order consistent discretisation for equations for cells near green boundaries requires a $(p+1)$st-order accurate computation of virtual states.

First-order. First-order consistency can be obtained by second-order accurate computation of virtual states. For this formulae (2.5.2) are applied. From (4.5.1) and (4.5.3) it is clear that the requirements (2.4.38) are satisfied for $p = 1$, if they are satisfied by $q^{l+1}_{2i+2,2j+1}$ when $\Omega^{l+1}_{2i+2,2j+1}$ would exist. This is shown to be the case in [7].

Second-order. Similarly a third-order accurate computation of the virtual state is required for second-order consistency. This causes no extra second-order error with respect to the situation where $\Omega^{l+1}_{2i+2,2j+1}$ would exist. As shown in [7], in that situation (i.e., $\Omega^{l+1}_{2i+2,2j+1}$ exists) a second-order accurate discretisation is obtained.

A third-order accurate computation of virtual states is given by (2.5.4), i.e.

(2.5.4a)
$$v^{l+1}_{2i+2,2j+1} = \frac{17}{16} q^l_{i+1,j} + \frac{1}{16}\left(q^l_{i,j} + q^l_{i,j+1} + q^l_{i+1,j+1}\right) - \frac{2}{16}\left(q^l_{i+2,j} + q^l_{i+1,j-1}\right),$$

(2.5.4b)
$$v^{l+1}_{2i+3,2j+1} = \frac{17}{16} q^l_{i+1,j} + \frac{1}{16}\left(q^l_{i+2,j} + q^l_{i+2,j+1} + q^l_{i+1,j+1}\right) - \frac{2}{16}\left(q^l_{i,j} + q^l_{i+1,j-1}\right).$$

This is schematically represented in Fig. 4.5.3. These are also chosen from a number of possible alternatives. Apart from compactness and symmetry, this choice is based on the size of the in absolute value largest negative coefficients. For the present choice, the negative coefficients are smaller in absolute value than for possible alternatives with similar compactness and symmetry. From the Taylor series expansion of q it can be shown that the virtual state $v^{l+1}_{2i+2,2j+1}$, obtained by (2.5.4a) can be expressed as

$$v^{l+1}_{2i+2,2j+1} = q_0 + \frac{3}{2}h_{l+1}q_1 + \frac{1}{2}h_{l+1}q_2 + \frac{7}{3}h^2_{l+1}q_3 + \frac{3}{4}h^2_{l+1}q_4 + \frac{1}{3}h^2_{l+1}q_5 + \frac{3}{2}h^3_{l+1}q_6 + \frac{5}{3}h^3_{l+1}q_7 + 3h^3_{l+1}q_9 + \mathcal{O}(h^4_{l+1}).$$

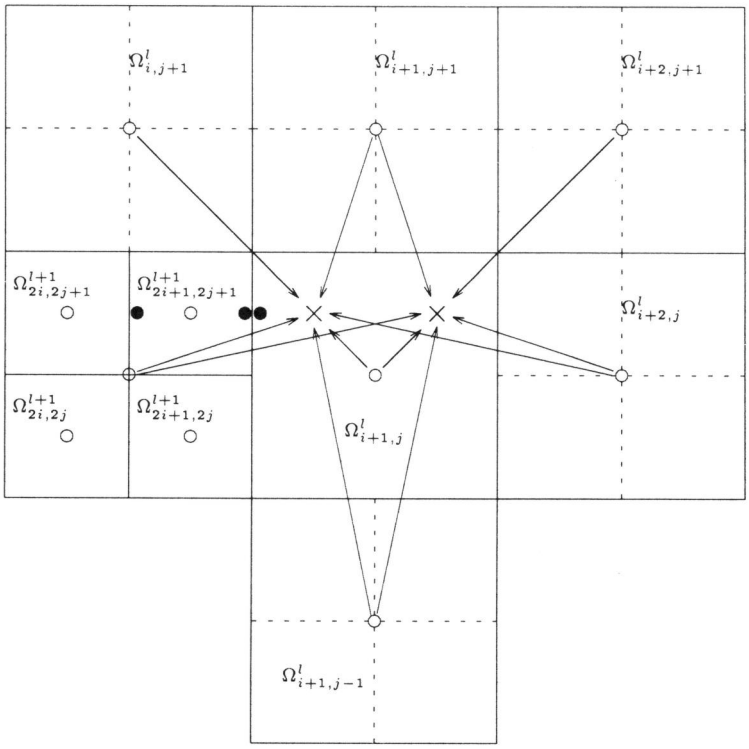


FIGURE 4.5.3. *Virtual states for second-order consistency on a locally refined grid;* ○*: available state;* ●*: left or right state;* ×*: virtual state.*

Clearly this is a third-order accurate approximation of the mean value on $\omega^{l+1}_{2i+2,2j+1}$. Hence, since this virtual state does not introduce extra second-order errors in the reconstruction, the requirements for consistency (2.4.38) are satisfied for $p = 2$ and the discretisation is second-order consistent.

Suggestions. Apart from accuracy considerations, the way to compute virtual states is chosen arbitrarily. Some suggestions for proper choices of virtual states are given here. These suggestions maintain the accuracy of the virtual states, while they may have other implications for the final numerical solution.

Virtual states may be computed with sufficient accuracy in several different ways. A way to do this for both first-order and second-order accuracy, is given in the previous section. Consider the situation where the four virtual states v^{l+1}_m, $\forall m \in K(i,j)$ are required, and for each of these we have two alternative ways to compute the virtual state: $(v^a)^{l+1}_m$ and $(v^b)^{l+1}_m$, which are both pth-order accurate. We can now choose the final virtual state as a linear

combination of these two alternatives

$$v_m^{l+1} = \lambda (v^a)_m^{l+1} + (1-\lambda)(v^b)_m^{l+1}.$$

The choice of λ is still undecided. E.g., λ may be chosen so that the mean value of the four virtual states v_m^{l+1}, $\forall m \in K(i,j)$ equals $q_{i,j}^l$ (as in the restriction of the fine-grid solution). This gives for λ

$$\lambda = \frac{q_{i,j}^l - \frac{1}{4}\sum_{m\in K(i,j)}(v^b)_m^{l+1}}{\frac{1}{4}\sum_{m\in K(i,j)}((v^a)_m^{l+1} - (v^b)_m^{l+1})}.$$

Another choice may be given e.g., by requiring that for the eastern cell faces, $\partial\Omega_{2i+1,2j,E}^{l+1} \cup \partial\Omega_{2i+1,2j+1,E}^{l+1} \subset \partial\Omega_g^{l+1}$, the numerical fluxes satisfy

$$F_{i,j,k}^l s_{i,j,k}^l = F_{2i+1,2j,E}^{l+1} s_{2i+1,2j,E}^{l+1} + F_{2i+1,2j+1,E}^{l+1} s_{2i+1,2j+1,E}^{l+1}.$$

From this requirement λ may be solved numerically. The implications of these choices require further investigation.

4.5.3. A-posteriori estimation of the local discretisation error in two space dimensions. Here we extend the method for estimation of the local discretisation error to the case with two space dimensions. We only consider first-order accuracy. In this case we only have to consider the contribution to the local discretisation error due to the reconstruction phase of the discretisation. The other contributions are all second-order. Extension to second-order accuracy is straightforward, although it involves more tedious calculations.

Similar to the one-dimensional case, we define the relative local truncation error by

$$\{\tau_{l+1}^l(q^{l+1})\}_{i,j}^l = \frac{1}{A_{i,j}^l}\Big\{\{N^l(\overline{R}^{l,l+1}q^{l+1}; q^{l-1})\}_{i,j}^l - \{R_{l+1}^l N^{l+1}(q^{l+1}; q^l)\}_{i,j}^l\Big\}, \quad \forall (i,j,l) \in I_f^l.$$

For $q^m = \overline{R}^m q$, $m = l-1, l, l+1$, and assuming exact approximation of the right-hand side, the relative truncation error is (cf. (4.4.2))

$$\tau_{l+1}^l(\overline{R}^{l+1}q) = \overline{R}^{l,l+1}\tau^l(q) - \overline{R}_{l+1}^l \tau^{l+1}(q) + \mathcal{O}(h_l^{p+1}).$$

Also for the two-dimensional case we split the local discretisation error $\tau^l(q)$ in a part $\tau_u^l(q)$, the local truncation error if the cell would be part of a locally uniform grid, and a perturbation, $\tau_n^l(q)$ due to the fact that the cell is a locally non-uniform grid cell, i.e.

$$\tau^l(q) = \tau_u^l(q) + \tau_n^l(q).$$

The perturbation $\tau_n^l(q)$ vanishes if the cell is part of a locally uniform grid. We denote estimates of the local discretisation error by a tilde over the symbols. Then, the uniform and non-uniform part of the local discretisation error are denoted by $\tilde{\tau}_u^l(q^l; q^{l-1})$ and $\tilde{\tau}_n^l(q^l; q^{l-1})$, respectively. Hence, we use the estimate

$$\tilde{\tau}^l(q^l) = \tilde{\tau}_u^l(q^l; q^{l-1}) + \tilde{\tau}_n^l(q^l; q^{l-1}). \tag{4.5.4}$$

The uniform part. The uniform part is estimated by extrapolation. This is similar to the extrapolation given in (4.4.15) for the one-dimensional case. We have for the uniform part

$$\tilde{\tau}_u^{l+1}(q^{l+1}; q^l) = \frac{1}{2^p - 1} P^{l+1} \left(\tau_{l+1}^l(q^{l+1}) + \overline{R}_{l+1}^l \tilde{\tau}_n^{l+1}(q^{l+1}; q^l) \right). \tag{4.5.5}$$

For the operator $P^l : \overline{X}(\Omega^*) \to \overline{X}^l(\Omega^l)$ the piecewise constant interpolation can be chosen, which for a sufficiently smooth $q \in \overline{X}(\Omega^{l+1})$ satisfies (cf. (4.4.7))

$$P^{l+1} \overline{R}_{l+1}^l \overline{R}^{l+1} q = \overline{R}^{l+1} q + \mathcal{O}(h_l).$$

The non-uniform part. The non-uniform part $\tau_n^l(q)$ of the local discretisation error is estimated by considering the terms appearing in the expressions for the local discretisation error, as found with the Taylor series expansions. An expression for the contribution of the reconstruction to the local discretisation error is given by (2.4.35). We are only interested in the reconstruction error $r_{i,j,k}^l$, appearing in the first term of (2.4.35), because all other terms give higher-order contributions to τ_n^l (assuming a consistent discretisation and a reconstruction that is at least first-order accurate). The left and right states $(q^L)_{i,j,k}^l$ and $(q^R)_{i,j,k}^l$ can also be written as the sum of a part from a uniform-grid situation and a perturbation due to the locally non-uniform grid. We define

$$\begin{aligned} (q^L)_{i,j,k}^l &= (q_u^L)_{i,j,k}^l + (q_n^L)_{i,j,k}^l, \\ (q^R)_{i,j,k}^l &= (q_u^R)_{i,j,k}^l + (q_n^R)_{i,j,k}^l, \end{aligned}$$

With this and with (2.4.35) we have for the non-uniform part of the local discretisation error

$$\begin{aligned} (\tau_n)_{i,j}^l &= (\tau)_{i,j}^l - (\tau_u)_{i,j}^l \\ &= \frac{1}{A_{i,j}^l} \sum_{k \in D} \left\{ \left. \frac{\partial F}{\partial (q^L)^\alpha} \right|_{q_{i,j}^l} \left((q_n^L)_{i,j,k}^l \right)^\alpha \right. \\ &\qquad \left. + \left. \frac{\partial F}{\partial (q^R)^\alpha} \right|_{q_{i,j}^l} \left((q_n^R)_{i,j,k}^l \right)^\alpha \right\} \bar{s}_{i,j,k}^l + \mathcal{O}(h_l^2), \end{aligned} \tag{4.5.6}$$

where the superscript α denotes the components of the state vectors, and where summation convention is used for $\alpha = 1, \ldots, d$. Here, we expanded the notations in (2.4.35) to its original components, using $(q^L)^\alpha$ and $(q^R)^\alpha$ instead of the components $(w^{LR})^\beta$, and $w^{LR} = (q^L, q^R)^T$. Note that $(q_n^L)_{i,j,k}^l$ and $(q_n^R)_{i,j,k}^l$ both vanish if the numerical flux $F_{i,j,k}^l$, across $\partial\Omega_{i,j,k}^l$ is independent of virtual states. To be more specific: $(q_n^L)_{i,j,k}^l$ equals zero when the left state $(q^L)_{i,j,k}^l$ does not depend on a virtual state. An analogous statement holds for the non-uniform part of the right state. Note that contributions from two opposite cell faces never cancel, since two opposite cell faces never are *both* part of the green boundary.

We now consider the eastern cell face $\partial\Omega_{2i+1,2j+1,E}^{l+1} \subset \partial\Omega_g^{l+1}$. For the first-order consistent discretisation we use the reconstruction given by (2.3.18), i.e.

$$
\begin{aligned}
(q^L)_{2i+1,2j+1,E}^{l+1} &= q_{2i+1,2j+1}^l, \\
(q^R)_{2i+1,2j+1,E}^{l+1} &= v_{2i+2,2j+1}^l.
\end{aligned}
$$

Hence, for this situation we have

(4.5.7a)
$$
(q_n^L)_{2i+1,2j+1,E}^{l+1} = 0,
$$

(4.5.7b)
$$
(q_n^R)_{2i+1,2j+1,E}^{l+1} = v_{2i+2,2j+1}^{l+1} - \frac{1}{h_{l+1}^2} \int_{\tilde{\omega}_{2i+2,2j+1}^{l+1}} q \, d\tilde{\Omega}.
$$

We assume that $F_{2i+1,2j+1,E}^{l+1}$ is the only numerical flux for this cell that depends on a virtual state. Substitution of (4.5.1) and (4.5.3) in (4.5.7), and subsequently substitution of (4.5.7) in (4.5.6) for cell $\Omega_{2i+1,2j+1}^{l+1}$, gives

(4.5.8)
$$
(\tau_n)_{2i+1,2j+1}^{l+1} = \frac{1}{A_{2i+1,2j+1}^{l+1}} \left. \frac{\partial F}{\partial (q^R)^\alpha} \right|_{q_{2i+1,2j+1}^{l+1}} (q_3 - \frac{3}{4} q_4 + q_5)^\alpha h_{l+1}^2 s_{2i+1,2j+1,E}^{l+1} + \mathcal{O}(h_{l+1}^2).
$$

We now have to estimate the derivatives appearing in (4.5.8). The estimates have to be first-order accurate only. We assume similar smoothness conditions as for the one-dimensional case in 4.4.3 and assume this condition to be satisfied. For the present situation we use the following approximations

(4.5.9a)

$$q_3 h_{l+1}^2 = \frac{1}{2}\left.\frac{\partial^2 q}{\partial \xi^2}\right|_{q_{i,j}^l} h_{l+1}^2 = \frac{3}{16}(q_{2i,2j}^{l+1} + q_{2i,2j+1}^{l+1}) - \frac{5}{16}(q_{2i+1,2j}^{l+1} + q_{2i+1,2j+1}^{l+1}) + \frac{1}{4}q_{i+1,j}^l + \mathcal{O}(h_{l+1}^3),$$

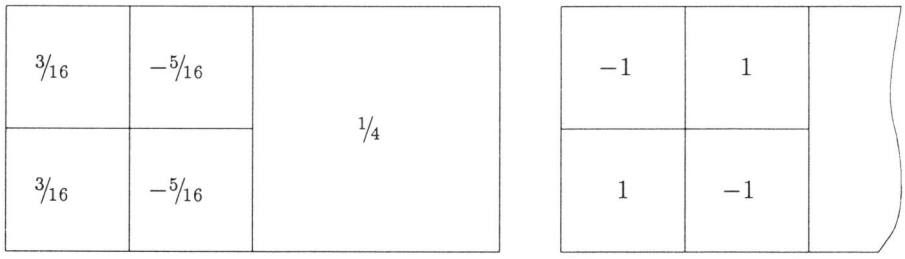
(4.5.9b)

$$q_4 h_{l+1}^2 = \left.\frac{\partial^2 q}{\partial \xi \partial \eta}\right|_{q_{2i+1,2j+1}^{l+1}} h_{l+1}^2 = q_{2i,2j}^{l+1} - q_{2i+1,2j}^{l+1} - q_{2i,2j+1}^{l+1} + q_{2i+1,2j+1}^{l+1} + \mathcal{O}(h_{l+1}^3),$$

(4.5.9c)

$$q_5 h_{l+1}^2 = \frac{1}{2}\left.\frac{\partial^2 q}{\partial \eta^2}\right|_{q_{i,j}^l} h_{l+1}^2 = \frac{3}{16}(q_{2i,2j}^{l+1} + q_{2i+1,2j}^{l+1}) - \frac{5}{16}(q_{2i,2j+1}^{l+1} + q_{2i+1,2j+1}^{l+1}) + \frac{1}{4}q_{i,j+1}^l + \mathcal{O}(h_{l+1}^3).$$

These contributions are schematically represented in Fig. 4.5.4. With the



a. *Estimation of* $q_3 h_{l+1}^2$. b. *Estimation of* $q_4 h_{l+1}^2$.

FIGURE 4.5.4. *First-order accurate approximations of the terms* $q_3 h_{l+1}^2$ *and* $q_4 h_{l+1}^2$; $q_5 h_{l+1}^2$ *is estimated similar to* $q_3 h_{l+1}^2$.

derivatives estimated by (4.5.9), ignoring higher-order terms, we can estimate τ_n^l, which itself is first-order, with second-order accuracy. The estimate of τ_n^l is denoted by $\tilde{\tau}_n^l$.

4.5.4. Local grid refinement based on a-posteriori estimation of the local discretisation error. With the estimate of the uniform part defined by (4.5.5) and the non-uniform part, defined by (4.5.8) and (4.5.9), and

with

$$\tilde{\tau}^l = \tilde{\tau}^l_u + \tilde{\tau}^l_n \tag{4.5.10}$$

we found a sufficiently accurate estimate $\tilde{\tau}^l$ of τ^l.

Numerical example of the estimate at an arbitrary green boundary. We study the estimation of the local discretisation error for a model problem with an arbitrary green boundary. In order to numerically establish the accuracy of the estimate of the local discretisation error given above, we consider the same two-dimensional, nonlinear model problem, already discussed in Sec. 2.5.2, with exact solution shown in Fig. 2.5.1, given by (2.5.5). We use a grid with a green boundary at $x = 0.5$, as shown Fig. 2.5.2. For this configuration, the local discretisation error is given in Tab. 2.5.3. In Tab. 4.5.1 norms of the

TABLE 4.5.1. *Error in the estimate of the first-order local discretisation error for the situation with an arbitrary green boundary; results for a model problem.*

h_L	$\|\tilde{\tau}^L - \tau^L\|_{\infty,\overline{X}^L(\Omega_g^L)}$	$\|\tilde{\tau}_c - \tau_c\|_{L_1,\overline{X}_c(\Omega)}$
$\frac{1}{8}$	0.1114e + 01	0.1323e + 01
$\frac{1}{16}$	0.2841e + 00	0.3763e + 00
$\frac{1}{32}$	0.7222e − 01	0.9709e − 01
$\frac{1}{64}$	0.1809e − 01	0.2427e − 01
$\frac{1}{128}$	0.4519e − 01	0.6068e − 02
$\frac{1}{256}$	0.1129e − 02	0.1518e − 02

error in the estimate of the local discretisation error are given. Here, $\tilde{\tau}^L$ is the estimate (4.5.10) of the local discretisation error τ^L, on the finest level L. The local discretisation error on the composite grid is τ_c and its estimate is denoted by $\tilde{\tau}_c$.

Note that the maximum norm is on $\overline{X}^L(\Omega_g^L)$, where $\Omega_g^l \subset \Omega^l$ denotes the collection of green cells of the grid on level l. Clearly, the estimate of the first-order local discretisation error is second-order accurate.

Estimate of local discretisation error used for local refinement. We can use the estimate of the local discretisation error in a refinement criterion. Therefore, as an example, we numerically solve the model problem discussed in Sec. 2.5.2, with exact solution given by (2.5.5).

First we apply in the refinement criterion the straightforward estimation of the local discretisation error, by extrapolation of the relative local discretisation error, assuming that the discretisations on two consecutive levels of

refinement are similar. This gives the (inaccurate) estimate

$$\hat{\tau}^l = \frac{1}{2^p - 1} P^{l+1} \tau_{l+1}^l (q^{l+1}, q^l), \tag{4.5.11}$$

where P^{l+1} is piecewise constant interpolation and p the order of consistency of the equations. This is an inaccurate estimate, since the assumption that the discretisations on two consecutive levels of refinement are similar, is incorrect. The grid is obtained by applying the refinement strategy given in Sec. 3.3. A cell is refined if the estimate (4.5.11) is larger than 1.0. The grid obtained with this strategy is shown in Fig. 4.5.5. We observe that it does not reflect

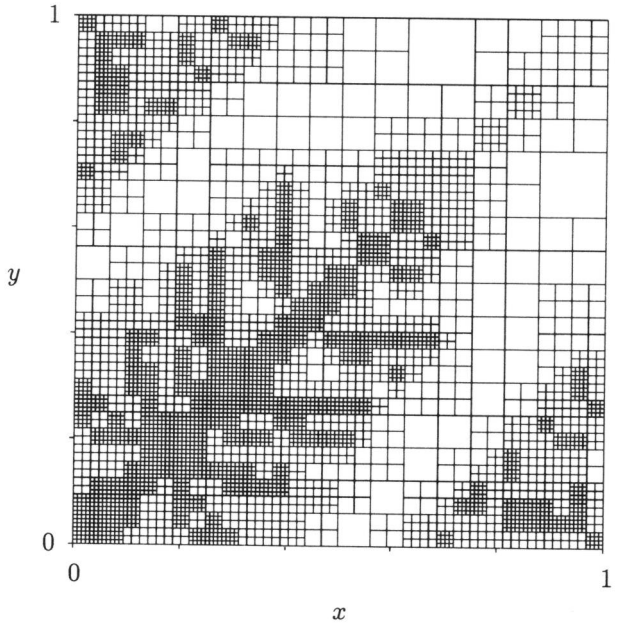


FIGURE 4.5.5. *Locally refined grid obtained with inaccurate estimate of the local discretisation error.*

the local discretisation error. Hence, this primitive refinement by application of (4.5.11) is not effective.

As a contrast we now apply the accurate estimate given by (4.5.4), (4.5.5) and (4.5.8), in the refinement criterion. We use the same refinement strategy, with cells refined when the estimate exceeds 1.0. The grid obtained is shown in Fig. 4.5.6 This grid reflects the absolute value of the local discretisation error for the first-order discretisation. For the first-order discretisation, the lowest-order term of the local discretisation error is given by τ_r^l in (2.4.35)

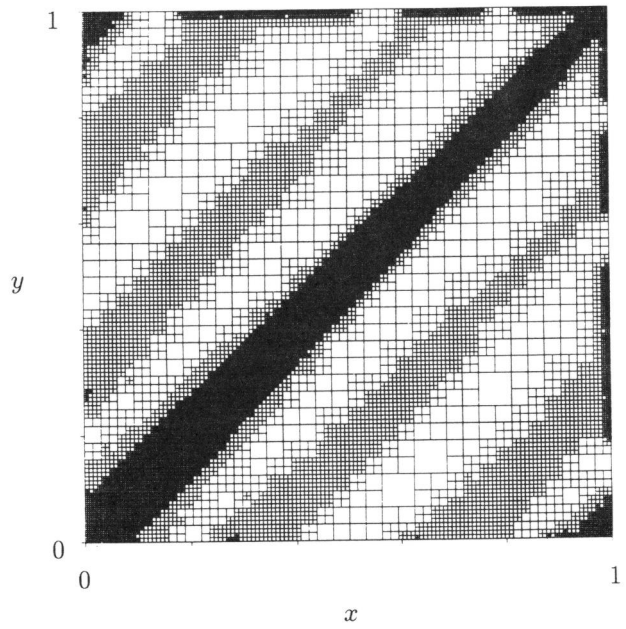


FIGURE 4.5.6. *Locally refined grid obtained with accurate estimate of the local discretisation error.*

and the local discretisation error can be written as

$$\{\tau^l\}^l_{i,j} = h_l \left\{ C_{\gamma,\delta} \frac{\partial^2 f}{\partial q^\alpha \partial q^\beta} \frac{\partial q^\alpha}{\partial x_\gamma} \frac{\partial q^\beta}{\partial x_\delta} + C'_{\gamma,\delta} \frac{\partial f}{\partial q^\alpha} \frac{\partial^2 q^\alpha}{\partial x_\gamma \partial x_\delta} \right\}^l_{i,j} + \mathcal{O}(h_l^2),$$

where $x_1 = x$ and $x_2 = y$. Here, summation convention for α, β, γ and δ and $C_{\gamma,\delta}$ and $C'_{\gamma,\delta}$ independent of h_l. There is refinement where first and second derivatives of the solution are large (in absolute value) except when the derivative of the flux vanishes. For the model problem, the flux derivatives are linear functions of the solution components. They vanish when the solution components vanish.

We observe that the more accurate error estimate generates correct local grid refinements (cf. Fig. 4.5.6), whereas the estimate (4.5.11) introduces spurious refinements.

4.6. Concluding remarks

For a locally refined grid we have studied in detail a-posteriori estimation of the local discretisation error for a finite volume, upwind discretisation of a

one-dimensional conservation law. We have shown how the same estimation technique can be applied to the two-dimensional case. We consider an estimate of an $\mathcal{O}(h_l^p)$ accurate local discretisation error to be sufficiently accurate, if it approximates the local discretisation error with $\mathcal{O}(h_L^{p+1})$ accuracy.

In the discretisation, for the one-dimensional case we considered two interpolations for the computation of virtual states at fine-coarse grid interfaces (so-called green boundaries). It is shown that, to obtain a first-order accurate solution, it suffices to use a first-order accurate interpolation to compute the virtual states. However, this gives zeroth-order local discretisation errors in the cells neighbouring a green boundary. A second-order accurate interpolation gives a first-order local discretisation error. For both discretisations the global error is first-order. However, the second-order accurate interpolation may yield a smoother solution.

It is also shown that extrapolation of the relative local truncation error (as in τ-extrapolation) can be used for the a-posteriori estimation of the local discretisation error, provided that the exact solution and the global discretisation error satisfy a certain smoothness condition. In the neighbourhood of a green boundary, the estimate is based on a splitting of the local discretisation error. We distinguish a part equal to the local discretisation error of the regular scheme (the uniform part) and a perturbation, due to the irregularity introduced by the lack of refinements in neighbouring cells (the non-uniform part). The uniform part is estimated by extrapolation of the relative local truncation error. The non-uniform part is approximated by estimating the expression for this term obtained by Taylor series expansion. This estimation technique can be applied if the solution of the continuous problem is differentiable. To derive this result, it is assumed that the global error can be written as an asymptotic expansion in the mesh width, with a differentiable function in the lowest-order term. For an example problem it is shown that, away from the green boundary, the global error can be considered as the restriction of a differentiable function.

The global error on a locally non-uniform grid may not satisfy the smoothness condition mentioned above. For a one-dimensional example problem it is shown how to accurately estimate the local discretisation error, if the smoothness condition is not satisfied. In that situation, the estimate is based on extrapolation of the numerical solution across the green boundary. The extrapolant obtained is an approximation of the solution, sufficiently accurate to use it in the estimation of both the uniform and the non-uniform part of the local truncation error. The extrapolated discrete function also satisfies the smoothness condition.

Estimation of the local discretisation error has been extended to the discretisation on a locally refined grid, for a system of conservation laws in two space dimensions. For the first-order accurate discretisation, the uniform part and the non-uniform part of the local discretisation error can be estimated with sufficient accuracy (i.e. with $\mathcal{O}(h_l^2)$ accuracy for the first-order discretisa-

tion), assuming that a similar smoothness condition as for the one-dimensional case is satisfied. An example of local refinement based on the accurate approximation of the local discretisation error for a two-dimensional model problem shows that the estimate can be successfully applied indeed.

References

[1] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp. **31** (1977), no. 138, 333–390.

[2] ———, *Multi-level adaptive techniques (MLAT) for singular perturbation-problems*, Numerical Analysis of Singular Perturbation Problems (P.W. Hemker and J.J.H. Miller, eds.), Academic Press, 1979, pp. 53–142.

[3] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer Series in Computational Mathematics, vol. 4, Springer-Verlag, Berlin, 1985.

[4] C.W. Oosterlee and P. Wesseling, *A multigrid method for an invariant formulation of the incompressible Navier-Stokes equations in general coordinates*, Comm. Appl. Numer. Methods **8** (1992), 721–734.

[5] S. Osher and F. Solomon, *Upwind difference schemes for hyperbolic systems of conservation laws*, Math. Comp. **38** (1982), no. 158, 339–374.

[6] P.L. Roe, *Approximate Riemann solvers, parameter vectors and difference schemes*, J. Comput. Phys. **43** (1981), 357–372.

[7] S.P. Spekreijse, *Multigrid Solution of the Steady Euler Equations*, CWI, Amsterdam, 1988, CWI-tract 46.

[8] H.T.M. van der Maarel and B. Koren, *Spurious, zeroth-order entropy generation along a kinked wall*, Internat. J. Numer. Methods Fluids **13** (1991), 1113–1129.

[9] B. van Leer, *Flux-vector splitting for the Euler equations*, Proceedings of the Eight International Conference on Numerical Methods in Fluid Dynamics, Aachen, 1982, Lecture Notes in Physics, vol. 170 (E. Krause, ed.), Springer-Verlag, 1982, pp. 507–512.

Samenvatting (*abstract in Dutch*)

Tegenwoordig worden in tal van takken de industrie numerieke stromingssimulaties toegepast. Een belangrijke toepassingsgebied vormt de vliegtuigbouw. Om te fungeren als gereedschap in een ontwerpproces, dient een numerieke simulatie nauwkeurig, maar ook goedkoop te zijn. Enerzijds maakt een goedkope en nauwkeurige methode simulaties mogelijk van complexere problemen. Anderzijds stelt het de industrie in staat een in economisch opzicht concurrerend produkt te ontwerpen in een aanvaardbare tijdspanne.

Multiroostermethoden behoren tot de meest efficiënte methoden om een stelsel discrete vergelijkingen op te lossen. Om een efficiënte numerieke simulatiemethode te maken, is het interessant om niet alleen gebruik te maken van een multiroostertechniek, maar ook om het aantal onbekenden in het numeriek-wiskundige probleem, dat ten grondslag ligt aan de simulatie, te beperken. Met dit doel wordt in dit proefschrift een multiroostermethode gepresenteerd, die gebruik maakt van automatische, oplossingsafhankelijke aanpassingen van het rekenrooster. In deze methode wordt een gegeven rooster alleen lokaal verfijnd, waar dat voor de gewenste nauwkeurigheid noodzakelijk is. Deze methode kan worden toegepast op fysiche modellen die worden beschreven met behulp van behoudswetten. Er wordt in het bijzonder een lokale roosterverfijningsmethode beschouwd voor een speciaal stelsel niet-lineaire behoudswetten, n.l. de stationaire Euler-vergelijkingen in twee ruimtedimensies. Dit stelsel van vier partiële differentiaalvergelijkingen beschrijft de twee-dimensionale stroming van een compressibel, ideaal gas. De Euler-vergelijkingen worden als wiskundig model vaak gebruikt in de vliegtuigbouw.

De discretisatie van het stelsel differentiaalvergelijkingen zoals gebruikt in dit proefschrift, is een eindige volume, upwind-discretisatie van de stationaire Euler-vergelijkingen in behoudsvorm. Het rekenrooster is een discretisatie van het definitiegebied van de differentiaalvergelijkingen. Een lokaal verfijnd rooster wordt in dit proefschrift opgevat als een samenstelling van (delen van) roosters met verschillende verfijningsgraad. Ieder verfijningsniveau heeft een rooster dat het definitiegebied geheel of gedeeltelijk overdekt. Het rooster op een eerstvolgend niveau van verfijning ontstaat door een aantal cellen van een rooster te verdelen in vier kleinere cellen van ongeveer gelijke grootte. Het rooster dat uiteindelijke de numerieke benadering van de oplossing bepaalt, is de verzameling van alle niet-verfijnde cellen.

De methode om het stelsel discrete (eerste orde nauwkeurige) vergelijkingen op te lossen, is een niet-lineaire multiroostermethode. Een tweede orde nauw-

keurig stelsel vergelijkingen wordt opgelost met behulp van defect correctie iteratie, waarbinnen de multiroostermethode voor de eerste orde vergelijkingen gebruikt wordt.

Zowel deze discretisatie van de vergelijkingen als de multiroostermethode en de defect correctie methode is in de jaren 1984–1989 ontwikkeld en geïmplementeerd op het CWI, door Hemker, Spekreijse en Koren. Het werk beschreven in dit proefschrift is een directe uitbreiding van het werk van deze auteurs.

Een belangrijk resultaat van deze uitbreiding is een numerieke methode die –door de automatisch adaptieve roosterverfijning– afhankelijk van het beschouwde probleem en de gebruikte discretisatie, voor de twee-dimensionale Euler-vergelijkingen een factor vier tot tien aan efficiëntie wint.

In hoofdstuk 2 van dit proefschrift wordt de geometrische structuur en de discretisatie voor stationaire behoudswetten beschreven. De geometrische structuur maakt het mogelijk een stelsel van discrete vergelijkingen te definiëren, dat het continue probleem benadert. In dit hoofdstuk wordt ook de lokale discretisatiefout bestudeerd en worden eisen geformuleerd waaraan de discrete vergelijkingen moeten voldoen om een –in een bepaalde zin– eerste of tweede orde consistente discretisatie te leveren. De aandacht is speciaal gericht op de randen die de overgang vormen tussen een locale verfijning en het aansluitende grove rooster.

In hoofdstuk 3 worden resultaten gepresenteerd van numerieke experimenten, verkregen met de methode beschreven in hoofdstuk 2. De beschouwde problemen zijn gekozen om een idee te krijgen van de nauwkeurigheid, de efficiency en de flexibiliteit van de methode (d.w.z. in hoeverre de methode de mogelijkheid biedt om effectief gebruikt te worden in bijzondere situaties).

Hoofdstuk 4 gaat over een a-posteriori schatting van de lokale discretisatiefout, met het doel deze te gebruiken in een criterium voor de verfijning van het rooster. De lokale en globale discretisatiefouten voor een één-dimensionaal modelprobleem worden in detail bestudeerd. Vervolgens wordt een methode geïntroduceerd en bestudeerd, die het mogelijk maakt de lokale discretisatiefout te schatten. Uiteindelijk wordt deze methode uitgebreid naar twee ruimtedimensies en wordt deze gebruikt als roosterverfijningscriterium voor een niet-lineair modelprobleem.