TIME INTEGRATION METHODS FOR COMPRESSIBLE FLOW

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 24 september 1999 te 15.00 uur

door

René van Buuren

geboren op 14 april 1972
te Leeuwarden

Dit proefschrift is goedgekeurd door
de promotor

prof. dr. ir. P.J. Zandbergen

en de assistent-promotor

dr. J.G.M. Kuerten

aan mijn ouders

Buuren, René van

Time integration methods for compressible flow
Thesis Universiteit Twente, Enschede.
- With ref. - With summary in Dutch.
ISBN 90-36513243

Faculty of Mathematical Sciences
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Contents

Chapter 1

Introduction

I am an old man now, and when I die and go to heaven there are two matters on which I hope for enlightenment. One is quantum electrodynamics, and the other is the turbulent motion of fluids. And about the former I am rather optimistic [1].

1.1 Turbulence

In both nature and technological applications the flow of fluids plays an important role. Consider e.g. an airplane: important properties as lift and drag are completely determined by the flow of air around the airplane. As a second example the mixing of fuel and oxygen in an internal combustion engine determines much of the efficiency of the engine and the emission of gases into the environment. In our bodies blood runs through our veins, and we breath air in and out of our lungs. Or consider the motion of gases in the atmosphere and the flow of water in lakes and seas and their relevance to weather prediction [30, 44, 64]. In all these cases turbulence plays an important role and depending on the situation turbulence has beneficial or detrimental effects. Obviously, a detailed investigation of the dominant flow mechanisms is essential in order to understand these systems or to optimize certain specific properties.

Roughly speaking there are two characteristic types of flow, i.e. laminar flow and turbulent flow. Laminar flow displays smooth and gradual changes in time and space, whereas turbulent flow behaves chaotic with rapid changes in time and space. An every-day example of these two types of flow can be found if one observes the flow from a water tap [64]. If the tap is opened slightly the

[1] Reportedly said by Sir Horace Lamb in an address to the British Association for the Advancement of Physics in 1932 [64].

water comes out in a smooth current, characterizing laminar flow. However, if the tap is fully opened the water flow becomes more vivid, characterizing turbulent flow.

In general a flow of a Newtonian fluid is mathematically described by the Navier-Stokes equations which are a set of coupled nonlinear partial differential equations combining the physical laws of conservation of mass, momentum and energy (see Eq. 6-1). By non-dimensionalising the Navier-Stokes equations it appears that a flow of a calorically perfect gas can be compactly characterized by four dimensionless numbers: (γ, Pr, Re, M). The adiabatic gas constant γ and the Prandtl number, Pr, depend on the type of fluid. For many fluids these are experimentally determined which leaves the Mach number, M, and the Reynolds number, Re still to be specified. The Mach number, which is the ratio of the speed of the fluid and the speed of sound, is related to compressibility, i.e. density variations, of the fluid. For low Mach number the fluid is nearly incompressible (e.g. constant density) like e.g. water in about all applications, while for high Mach number flow of e.g. gases typical compressible phenomena such as shock waves can occur. The Reynolds number is the ratio of inertia and viscous effects and is related to the two characteristic types of flow described earlier. For low Reynolds number the flow is laminar whereas it becomes turbulent if a critical (flow dependent) Reynolds number is exceeded.

In this century a lot of effort has been put into understanding turbulence and the phenomena which trigger turbulence. Next to theoretical developments and experiments, computer capacity has increased enormously in the past two decades which enables simulations of flows of increasing complexity. However, on the whole only a few substantial advances appear to have been made. For a short and comprehensible review we refer to [30]. Here we only focus on some aspects related to computational effort associated with flow simulations and for more details we refer to Refs. [30, 44, 85] and references therein. One of the assumptions on turbulence is that turbulent flows are hierarchical and involve a broad spectrum of so-called eddies. The largest eddies are produced by the driving forces of the flow, which in turn break into smaller eddies and this process continues until the resulting eddies reach a size on which molecular viscosity converts the eddy motion into heat. The property of this energy cascade from large to small eddies is predicted by Kolmogorov [51] and is expected to hold universally. In the Kolmogorov theory of turbulent flow the ratio of the lengths of the largest and smallest eddies is of the order $Re^{3/4}$ which in three dimensions results in a number of degrees of freedom in the order of $Re^{9/4}$.

With this prediction of the number of degrees of freedom it is possible to roughly estimate the computational cost for simulating turbulent flow. If

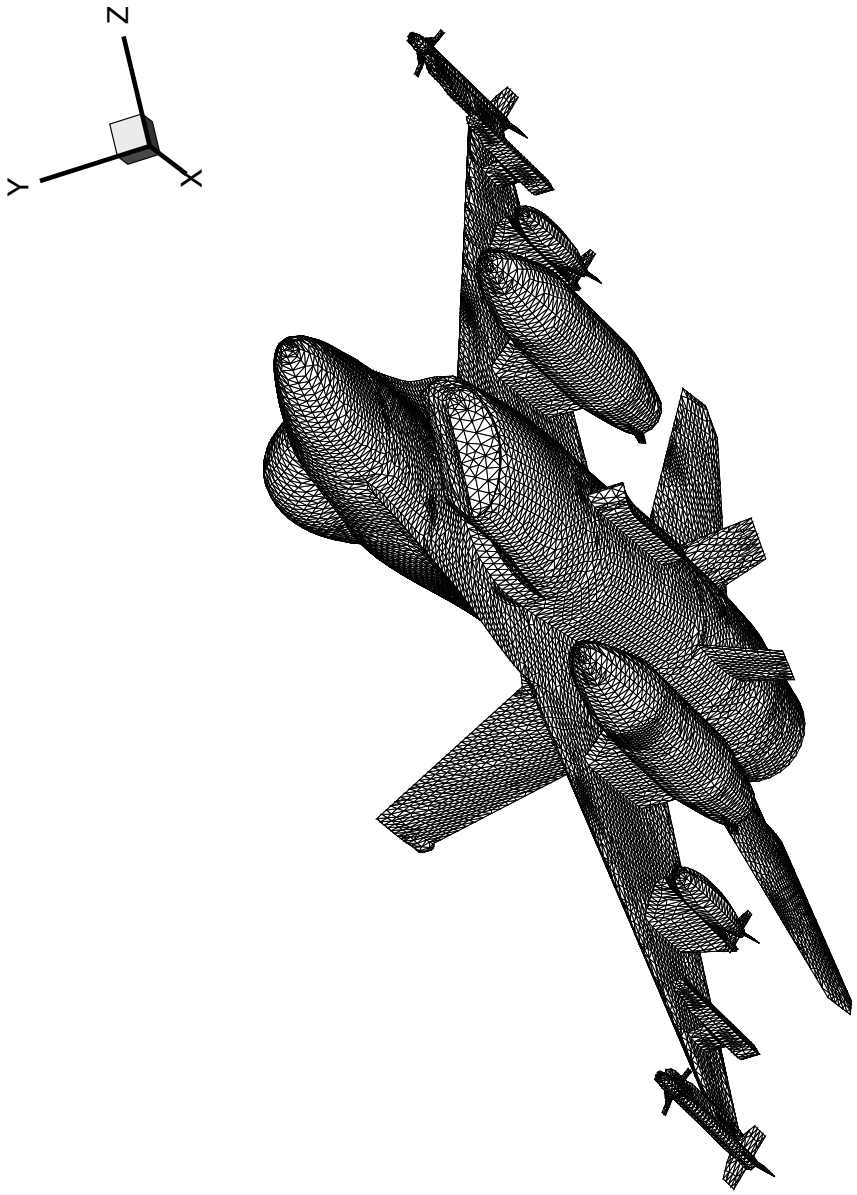


Figure 1-1: Surface triangulation of a generic fighter configuration; Technology demonstration of the FASTFLO CFD system (courtesy of NLR, Amsterdam).

a turbulent flow is to be simulated with a computer the continuous space is approximated by a discrete grid on which the flow quantities are determined. The grid size has to be fine enough to resolve all relevant eddies. This means that on the order of $Re^{9/4}$ grid points are required. If we assume that the amount of work is proportional to the number of grid points this means that the required CPU (central processor unit) time is also of the same order. For turbulent flows which typically have large Reynolds numbers the problem is obvious. Although flows with simple geometries such as turbulent flow over a flat plate can be simulated with increasingly high Reynolds number, see e.g. [71], the actual simulation of flows over complicated geometries in all its details is still far away. In Ref. [64] a striking example of the computational costs for a complete aircraft in flight conditions is presented. It is estimated that for an aircraft cruising at 250 meters per second at an altitude of 10.000 meters the Reynolds number is about 10^8 and together with the enormous complexity of the geometry of the entire airplane about 10^{16} grid points are required. With a computer that performs 10^{12} floating points operations per second, which did not yet exist at the time (1997), it would take several thousand years to compute only one second of flight time. A typical application of practical relevance is illustrated in figure 1.1 [17].

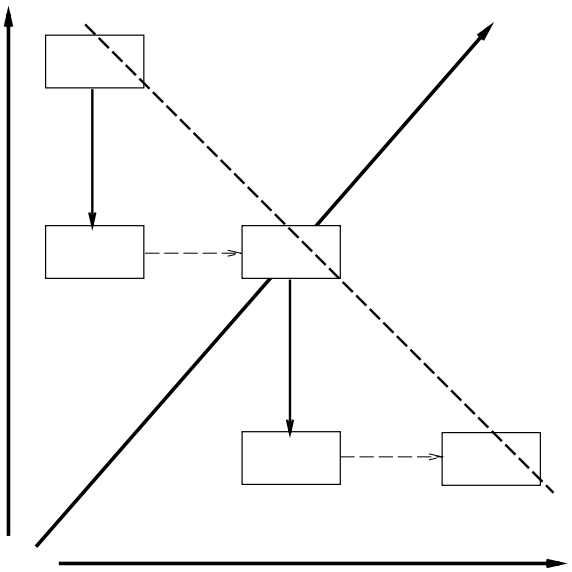


Figure 1-2: The relation between DNS, LES and RaNS with respect to complexity of flow, required resolution and computer power.

This example shows that a DNS (direct numerical simulation) of a flow in

a complex geometry described by the Navier-Stokes equations is not possible in the near future. Therefore, simplifications of the Navier-Stokes equations aimed at reducing the number of degrees of freedom such as RaNS (Reynolds averaged Navier-Stokes) and LES (large-eddy simulation) are commonly used (see e.g. Refs. [93]). In both methods the Navier-Stokes equations are filtered with respect to time (RaNS) or space (LES) such that the corresponding small structures in time or space do not have to be resolved by the numerical method. In this way coarser grids can be used which greatly reduces the required CPU time. However, the influence of these small structures, such as in the energy cascade described above, has to be modelled in order to close the Navier-Stokes equations. This closure is generally referred to as "turbulence modelling" in case of the RaNS equations and "sub-grid modelling" for LES. Although many models have been proposed for numerous types of flow and applications thus far no model has been developed which is generally valid. The relation between DNS, RaNS and LES with respect to computer power and required resolution (number of grid points) is represented in figure 1.1 (see also [81]). The computational hierarchy between DNS, LES and RaNS is clearly depicted on the dashed line which symbolically represents the present computational power. Simple flows with respect to Reynolds number and geometry can be resolved by a DNS whereas for more complicated flows a LES or RaNS approximation is required. However, the data obtained from a DNS of a simple flow can be used to validate the sub-grid model of a LES of the same flow. In turn this "approved" model can be incorporated into LES of a more complicated flow for which DNS is no longer feasible. A similar validation approach can be used for LES and the RaNS equations.

1.2 Time integration

Thus far we have only focused on the physical description of a flow by means of the full or filtered Navier-Stokes equations. We have argued that one way of enabling or accelerating a simulation is the modelling of small scale structures. Obviously computational methods play a crucial role as well. The increased use of CFD (computational fluid dynamics) and its application to more complicated flows impose severe demands on both computer hardware and numerical methods. Not only do the numerical methods have to resolve the physical phenomena occurring in a flow as accurately as possible, their performance on modern computer platforms, like e.g. massively parallel computers, is equally important. In order to identify the numerical aspects we write the Navier-Stokes equations in conservation form given by

$$\frac{\partial q}{\partial t} + \nabla \cdot f = 0 \tag{1-1}$$

where q is the unknown vector quantity and f denotes the flux vector. For more specific details see (6-1). In a numerical simulation both the temporal and spatial derivatives in (1-1) have to be discretised. A commonly used approach is the so-called method of lines which separates the temporal from the spatial discretisation. In this thesis we focus on the time integration of the Navier-Stokes equations. Assume that the spatial discretisation of $\nabla \cdot f$ is performed on an appropriate grid, then (1-1) becomes a set of coupled ordinary differential equation in time given by

$$\frac{dq_i}{dt} + F_i(q) = 0 \tag{1-2}$$

where the index i represents the grid point, q_i is the vector quantity in grid point i and F_i represents the numerical approximation of $\nabla \cdot f$ in grid point i.

Basically two types of time integration schemes can be applied to (1-2), i.e. explicit and implicit schemes. The former uses only information from the past to advance the solution q in time whereas the latter also requires information of the solution q which still has to be determined. To further clarify this point we present the Euler forward and backward schemes which are typical examples of the explicit and implicit approach respectively and are given by

$$\begin{array}{rcl} q_i^{n+1} & = & q_i^n - \Delta t F_i(q^n) \\ q_i^{n+1} & = & q_i^n - \Delta t F_i(q^{n+1}) \end{array} \tag{1-3}$$

where the superscript n denotes the time level and Δt the time step. Clearly, with the Euler forward scheme the solution q^{n+1} is easily computed if q^n is known. In comparison, for the Euler backward scheme the computation of q^{n+1} is more complicated and computationally intensive since F is a nonlinear function of q which means that a set of coupled nonlinear equations has to be solved. At first hand the explicit schemes appear to have the most favorable properties being simple and efficient. However, the main drawback of explicit schemes lies in the numerical stability limit on Δt (see section 2.4.1). If the time step during the simulation exceeds this stability limit the numerical scheme becomes unstable and the numerical error in the solution q can become unbounded. Many implicit schemes, on the other hand, in theory do not have this stability restriction on the time step and thus the time step can be related to the required numerical or physical accuracy alone. For applications in which the accuracy time step is much larger than the explicit stability time step an implicit scheme may constitute an attractive alternative. In figure 1-3 this is presented schematically. In this figure Δt_{stab} denotes the explicit stability time step, Δt_{imp} the implicit time step and Δt_{acc} the accuracy time. The accuracy time step represents a time level which gives rise to an acceptably accurate prediction of the time dependent solution. Ideally, the implicit time

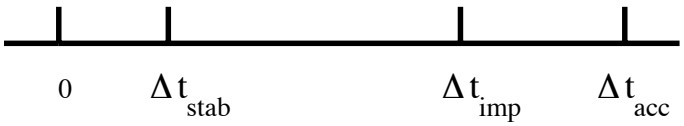


Figure 1-3: A case for which an implicit scheme may be more efficient than an explicit scheme.

step can be taken equal to the accuracy time step. The central theme in this thesis is related to Δt_{acc} and the key question that we try to answer is: What is a suitable criterion to dynamically determine the accuracy time step during a simulation in order to efficiently integrate the solution in time?

1.2.1 Numerical example

To illustrate the fact that an explicit stability time step may be too restrictive with respect to required accuracy we solve the linear convection equation with the two schemes in (1-3) for a specific choice of the initial and boundary conditions which is given by

$$\begin{cases} u_t + u_x &= 0 \\ u(x,0) &= \sin(x/100) \quad x \in [0,1] \\ u(0,t) &= \sin(-t/100) \quad t \geq 0 \end{cases} \tag{1-4}$$

where the subscripts denote the corresponding derivatives. The analytical solution to this problem is $u(x,t) = \sin((x-t)/100)$, which has a characteristic time scale in the order of 100. The spatial domain is discretised with a uniform grid with grid size $h = 0.01$ and for the spatial discretisation we use a first order upwind scheme (see section 2.3). For the Euler forward scheme the numerical stability restriction yields $\Delta t \leq h$. The numerical simulations are performed with $\Delta t = 0.01$ and $\Delta t = 1$ for the Euler forward and backward scheme respectively. The solutions $u(1/2,t)$ obtained with both time integration methods are presented as a function of time over a long time interval in figure 1-4. Visually, both solutions are indistinguishable. Although this is a very simple example it clearly shows that an explicit stability time step may be far too restrictive with respect to the required temporal accuracy if the characteristic time scale is much larger.

1.2.2 Relevant types of flow for implicit methods

Before we go into more detail concerning implicit methods let us first discuss the types of flow for which an implicit time integration method could be an

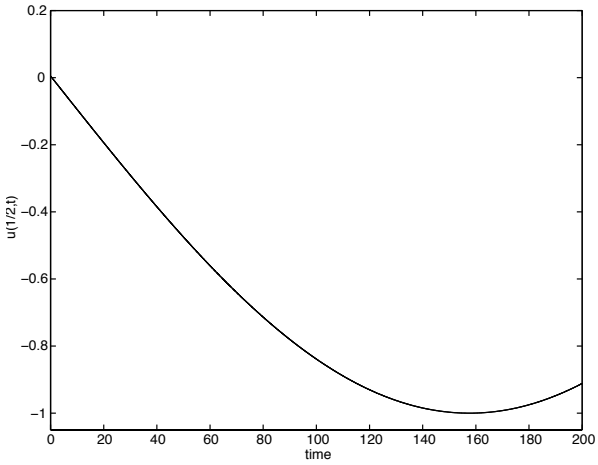


Figure 1-4: The solution $u(1/2, t)$ obtained with the Euler forward scheme with $\Delta t = 0.01$ and the Euler backward scheme with $\Delta t = 1$.

attractive alternative to explicit methods. First of all implicit methods can be used efficiently for steady flow problems. For these flows a time integration scheme is commonly used to advance the solution towards the steady state. The time in this case has no physical significance and is often denoted by pseudo time. Therefore, the accuracy of the time integration scheme has no influence on the accuracy with which the steady state solution is obtained. If an explicit scheme is used to advance the solution in pseudo time, the time step is still restricted for numerical stability reasons which means that an unnecessary small time step is used. So, for steady flows the benefits of implicit schemes are obvious and the application of these schemes may lead to a considerable speed up as will be shown in chapter 4.

For time dependent flows the break even point between explicit and implicit schemes is less transparent. In case of fully developed turbulent flows small scales in both time and space are present. The length and time scales of the smallest eddies scale as $Re^{-3/4}$ and $Re^{-1/2}$ [85] which appears to leave room for an implicit time integration scheme for large Reynolds number. However, it is argued in Ref. [9] that the smallest eddies are advected by the most energetic eddies. This dimensional argument yields an accuracy time scale similar to the CFL criterion which indicates that the explicit stability time step is also close to the accuracy time step and thus implicit schemes appear not efficient for turbulent flows.

The numerical example in (1-4) reveals that in case the physical time scale is large in comparison to the ratio of the grid size and local speed explicit

methods may not be the most efficient methods since the stability time step is related to the grid size (see section 2.4.1). Take for instance the turbulent viscous flow over an oscillating airfoil within the RaNS framework. Due to the Reynolds averaging the small time scales related to the turbulence are modelled leaving roughly the time scale related to the pitching frequency. In order to resolve the boundary layer a very fine grid is required near the airfoil. By rule of thumb one can argue that for accuracy reasons at least in the order of 50 to 100 time steps are required per period of the pitching frequency. Thus for high frequencies small time steps are required and explicit methods at first hand appear to be the best choice. However, for low frequencies the accuracy time step is considerably larger and the explicit stability time step, which is determined by the grid cell sizes near the solid boundary, may be far too restrictive. In this case implicit schemes have already proven to be an attractive alternative.

On the whole no general criterion can be given to decide whether an implicit scheme may be more efficient than an explicit scheme but some heuristic arguments can be given. For a DNS or LES at sufficiently small Reynolds number and in a simple geometry with a high aspect ratio of the grid the stability time step can be too restrictive. Also, for flows modelled with the RaNS equations with relatively large time scales implicit methods may be more efficient.

1.2.3 Efficiency of implicit schemes

Now that we have some feeling for which types of flow implicit schemes may be useful the following question arises: When is an implicit scheme more efficient than an explicit scheme? If we denote the amount of work units per time step of the explicit and implicit scheme by W_{exp} and W_{imp} respectively it easily follows that

$$\Delta t_{imp} \geq \frac{W_{imp}}{W_{exp}} \Delta t_{stab} \tag{1-5}$$

For a specific numerical scheme the amount of work W_{exp} can be easily computed. However, the amount of work for an implicit scheme depends on several parameters which in general are not known and therefore W_{imp} cannot be determined a priori. In the following we elaborate somewhat more on these parameters and discuss the computational complexity of implicit schemes using the Euler backward scheme as a typical example.

Consider the Euler backward scheme in (1-3) and assume we solve the resulting nonlinear set of equations with a Newton method. Then we obtain

the following iterative scheme:

$$\left(I + \Delta t \frac{\partial F}{\partial q}(q^k)\right) \Delta q_i^k = -\Delta t \; F_i(q^k) \tag{1-6}$$

where k denotes the iteration level, $\partial F/\partial q$ is a symbolic representation of the Jacobi matrix and $\Delta q_i^k = q_i^{k+1} - q_i^k$. If the iteration process converges, $\Delta q^\infty = 0$ and the solution at the next time level yields $q^{n+1} = q^\infty$. The numerical process in (1-6) consists of a linear and a nonlinear part. The linear part concerns the linear system arising within every iteration level of the nonlinear system which we denote by $A\Delta q = b$ where A in general is a sparse matrix. Depending on the properties of A such as e.g. diagonal dominance etc., an appropriate choice for an efficient linear solver can be made (see section 2.5). Next to the choice of the linear solver, which in most practical applications is an iterative scheme, we will show in this thesis that the accuracy with which the linear system is solved has a significant impact on the overall efficiency of the total method. The accuracy with which the linear system is solved can be defined by

$$\frac{||A\Delta q^p - b||}{||b||} < \epsilon_{lin} \tag{1-7}$$

where ϵ_{lin} is the required accuracy, the superscript p denotes the iteration level within the linear solution process and $||\cdot||$ denotes an appropriate norm.

The nonlinear part concerns the construction of the flux Jacobi matrix and the accuracy with which the nonlinear system is solved every time step i.e.

$$||F(q^k)|| < \epsilon_{nonlin} \tag{1-8}$$

where ϵ_{nonlin} is the required accuracy. In general the numerical techniques used to discretize the spatial derivatives for the flows discussed in this thesis use a local stencil, which means that the flux F_i in a grid point depends only on values of flow quantities in a neighborhood of the grid point i. As a result the flux Jacobi matrix is a sparse matrix as mentioned above. Consider e.g. a second order central spatial discretisation in one-dimension of a first order partial differential equation. Then $F_i = F_i(q_{i-1}, q_i, q_{i+1})$ and the resulting Jacobi matrix is a tridiagonal matrix. For higher order schemes in two or three dimensions larger stencils are required and the determination, storage and inversion of the Jacobi matrix becomes too expensive. This can be circumvented by an approximation of the flux Jacobi matrix using e.g. a lower order method which requires a smaller stencil. If the resulting system (1-6) converges this extra approximation has no effect on the accuracy of the solution q^{n+1} which is determined only by the order of the discretisation method used for the flux F. However, an approximation of the flux Jacobi matrix can influence the

required number of iterations to obtain a desired accuracy ϵ_{nonlin}. By and large we conclude that the amount of work W_{imp} is determined by the order of the approximation of the Jacobi matrix, the type of linear solver and the required accuracies ϵ_{lin} and ϵ_{nonlin}.

1.3 Outline of this thesis

The issues concerning the implicit time integration of the time dependent flows described in section 1.2.2 are the primary focus in this thesis. The key question that we would like to answer is: How can we dynamically determine the accuracy time step Δt_{acc}? With such a criterion it would be possible to efficiently integrate the solution in time by switching between an explicit and implicit scheme during a simulation. However, before developing such a criterion the efficiency of implicit schemes has to be studied first (see (1-5)).

In general it is hard to give estimates of the parameters mentioned in section 1.2.3 that influence the amount of work for an implicit method. Also, the type of discretisation required to resolve the physical phenomena may have a significant influence on the performance of an implicit scheme. Flows containing shock waves e.g. impose certain requirements on the numerical schemes. Without a proper numerical scheme shock waves are not captured correctly or numerical oscillations in the solution near shock waves are introduced. For more details see section 2.2. Higher order schemes suitable for shock capturing such as e.g. the MUSCL scheme [56] form a critical test case for our implicit schemes since these MUSCL type schemes add little numerical dissipation and incorporate a limiting function to avoid numerical oscillations (see section 2.3.1) which both influence the convergence characteristics of the nonlinear process. Therefore, in this thesis we simulate flows containing shock waves such as the one dimensional shock tube problem in chapter 3, the steady inviscid transonic flow around an airfoil in chapters 4 and 5 and the supersonic viscous flow over a flat plate in chapters 6 to 9.

The use of explicit schemes is widespread in the CFD community. The convergence towards the steady state for steady flows using explicit schemes is significantly improved by acceleration techniques such as local time stepping and multigrid [13, 38]. Also, the performance of explicit schemes on parallel platforms is excellent due to the locality of the data. Next to the key question how to dynamically determine the accuracy time step for unsteady flows we want to study the performance of the above mentioned acceleration techniques using implicit methods. Summarizing, the main research aims in this thesis can be formulated as follows:

1. What is a suitable implicit numerical scheme for steady and unsteady compressible flows containing shock waves?

2. The convergence towards a steady state of the transonic flow around an airfoil using an explicit time integration method as a smoother can be accelerated by the multigrid technique [15]. Does the application of multigrid yield convergence acceleration if an implicit scheme is used as a smoother?

3. Explicit schemes perform well on parallel platforms due to the locality of the required data. However, for implicit schemes the data locality is lost. Therefore, the question arises whether the implicit scheme mentioned in the first item is suitable for efficient parallel execution?

4. Are the multigrid results obtained for steady flows transferable to unsteady flows in general?

5. What is a suitable criterion to determine the accuracy time step for unsteady flow simulations?

6. What is the relation between Δt, ϵ_{lin} and ϵ_{nonlin} in view of overall efficiency?

If the latter two points are answered for a specific unsteady flow the behavior of Δt over a certain sampling period is known as well as the convergence characteristics of the implicit scheme. Possible ideas on the criterion for the dynamic determination of the accuracy time step can subsequently be tested by their ability to yield the above mentioned temporal behavior. In this way we come to the key question of this thesis:

7. What is a suitable criterion to dynamically determine the accuracy time step during a simulation in order to efficiently integrate the solution in time?

The contents of this thesis is as follows. In chapter 2 we discuss the numerical techniques which are used throughout this work. We show in chapter 3 that a basic linear solver suffices for the present flow applications. Additionally, we provide further indication of the relation between TVD (total variation diminishing) schemes and entropy satisfying solutions. The superior performance of an implicit scheme compared to an explicit scheme for a steady inviscid transonic flow around an airfoil is shown in chapter 4. With the implicit scheme a machine accurate steady state solution is obtained whereas for the explicit scheme the convergence stalls. The difference between these two solutions is studied and appears to be caused by a physical instability in the wake region which is not captured by the implicit scheme. In chapter 5 we study the performance of multigrid using the implicit scheme as a smoother applied to the steady flow in chapter 4. In general explicit methods for compressible flows

are highly parallelizable due to the locality of the data. By means of a domain decomposition we show in chapter 5 that the present implicit scheme is equally suitable for parallelization. In chapter 6 we apply an implicit scheme to simulate an unsteady shock boundary-layer flow. A criterion is developed in order to determine the accuracy time step for several types of flow quantities. Convergence problems occur for large time steps which appear to be related to the complex dynamical behavior of the numerical scheme. A multigrid analysis in chapter 7 reveals that no multigrid acceleration can be obtained for the present shock boundary-layer flow and more generally for unsteady flow in which the temporal terms are dominant in the smoothing operator. For the shock boundary-layer flow there appears to be a difference in the smoothing properties of the numerical scheme in x- and y-direction. Therefore, in chapter 8 we develop a new class of partially implicit time integration schemes which treat the fluxes in x-direction explicitly and the fluxes in y-direction implicitly. In chapter 9 some preliminary results of a dynamical time step criterion are presented. Finally, the conclusions are summarized in chapter 10.

Chapter 2

Numerical methods

2.1 Introduction

The focus in this thesis lies on the use and study of implicit methods for the time integration in CFD. Although a variety of different problems concerning numerical methods are discussed in other chapters, several aspects of the numerical methods are more general. In this chapter we outline the numerical methods used in this thesis. Additionally some properties of implicit time integration related to monotonicity preservation are presented. Due to the specific nature of the flow problems to be simulated in the next chapters shocks occur in the flow. These phenomena set certain demands on the spatial discretisation. In section 2.2 a brief description of the required properties of the numerical methods will be given. In section 2.3 the spatial discretisation used throughout this thesis will be described. Section 2.4.1 contains the description of the explicit time integration scheme which is used as a reference method in this thesis. In section 2.4.2 implicit time integration schemes are described. The application of implicit schemes results in a large set of coupled nonlinear algebraic equations which are solved with iterative schemes such as e.g. a Newton or approximate Newton method. During every iteration the solution to a linear system of equations has be to determined or approximated and two typical families of linear solution methods will be discussed in section 2.5. In section 2.6 the multigrid convergence acceleration technique is outlined. The description of boundary conditions and the numerical treatment of the boundary conditions will not be given in this chapter. Due to the diversity of the boundary conditions used for the different types of flows that are simulated in this thesis we postpone the description of specific details to the corresponding chapters.

2.2 Numerical requirements

The equations governing general inviscid or viscous flow are the Euler equations or Navier-Stokes equations respectively, which will be described in more detail in chapter 4 and chapter 6 in two dimensions. In general these equations can be written as a set of evolution equations in conservation form:

$$\frac{\partial q}{\partial t} + \nabla \cdot f(q, \nabla q) = 0 \tag{2-1}$$

where q represents the vector of unknown quantities, f represents the flux vector and the $\nabla\cdot$ operator and ∇ operator denote the multi-dimensional equivalent of the divergence and gradient operators respectively. For the Navier-Stokes equations the flux vector f can be split into a so-called inviscid and a viscous flux [86]. Later on this splitting will be used to discretize the two fluxes with different methods. Integration of (2-1) over an arbitrary volume, V, in space and application of Gauss' theorem yields

$$\frac{d}{dt}\int_V q dV + \int_S f(q, \nabla q) \cdot n dS = 0 \tag{2-2}$$

where S is the boundary of V and n is the outward unit normal on this surface. It is clear that the conserved quantity $\int_V q dV$ changes only due to the flux through the surface S. For numerical calculations the continuous equation (2-2) needs to be discretized. For illustration purposes we assume that equation (2-2) represents a one-dimensional differential equation. The volume V is now an interval that is divided into N sub-intervals through the introduction of $N+1$ grid points x_j with $j = 0, \ldots, N$ and $x_j < x_{j+1}$. The temporal and spatial discretisation of (2-2) can be performed separately which results in a semi-discretisation

$$\frac{dq_j}{dt} + F_j(q) = 0 \tag{2-3}$$

where q_j denotes the solution or solution vector in grid point x_j and F_j represents the corresponding numerical approximation of the flux. A numerical scheme which retains the conservation property in (2-2) will have the general form [57]

$$\frac{dq_j}{dt} + [F(q_{j-p}, q_{j-p+1}, \ldots, q_{j+k}) - F(q_{j-p-1}, q_{j-p}, \ldots, q_{j+k-1})] = 0 \tag{2-4}$$

where the term between square brackets represents the numerical flux F_j.

Conservative methods

A problem arises with numerical methods when discontinuities such as e.g. shocks are present in the solution. Analytically this means that there does not exist a solution which satisfies equation (2-1) in the classical sense but only equation (2-2) is satisfied. Such a solution is called a weak solution of equation (2-1) [105]. To illustrate some of the difficulties that may arise if discontinuities occur, a simple numerical experiment can be performed as in [56]. Consider the one-dimensional inviscid Burgers equation given by

$$\frac{\partial q}{\partial t} + \frac{\partial f(q)}{\partial x} = 0 \tag{2-5}$$

with $f(q) = \frac{1}{2}q^2$. For continuously differentiable solutions, the Burgers equation can be written as

$$\frac{\partial q}{\partial t} + q\frac{\partial q}{\partial x} = 0 \tag{2-6}$$

For $q \geq 0$ we consider the following two first-order upwind discretisations on a uniform grid with grid-spacing Δx and time step Δt given by

$$q_j^{n+1} = q_j^n - \alpha \Delta^n_{j-\frac{1}{2}} f \tag{2-7}$$

and

$$q_j^{n+1} = q_j^n - \alpha q_j^n \Delta^n_{j-\frac{1}{2}} q \tag{2-8}$$

with $\alpha = \Delta t/\Delta x$, $\Delta^n_{j-\frac{1}{2}} f = f(q_j^n) - f(q_{j-1}^n)$ and $\Delta^n_{j-\frac{1}{2}} q = q_j^n - q_{j-1}^n$ and the superscript n denotes the time level at t^n. As an initial condition we take

$$q(x) = \begin{cases} 1 & \text{if} \quad 0 \leq x < \frac{1}{2} \\ \frac{3}{4} & \text{if} \quad x = \frac{1}{2} \\ \frac{1}{2} & \text{if} \quad \frac{1}{2} < x \leq 1 \end{cases} \tag{2-9}$$

Furthermore $\Delta x = 1/100$ and the time step Δt is chosen such that the shock wave moves exactly one mesh width every two time steps. In figure 2-1 the solution q after 50 time steps is plotted in computational domain for the two discretisations in equations (2-7) and (2-8). Clearly, the shock wave does not propagate with the same shock speed for the two discretisations. From literature [56] it is known that the solution corresponding to the discretisation in equation (2-7), i.e. Godunov's scheme [36], is the proper one. It is straightforward to verify that the first discretisation is conservative as defined in equation (2-4) and the second discretisation is not conservative. The theorem of Lax-Wendroff [55] states that if the numerical solution converges as the grid is refined a proper weak solution of the underlying differential equation (2-2) is obtained if a conservative scheme is used.

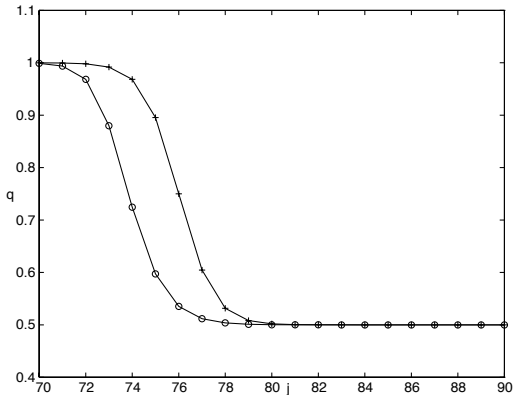


Figure 2-1: The solution q after 50 time steps in computational space. The +'s indicate the solution obtained with (2-7) and the ∘'s indicate the solution obtained with (2-8).

Numerical oscillations

Another problem related to the numerical treatment of shock waves is the occurrence of spurious numerical oscillations near the shock if an improper numerical scheme is used. Consider e.g. the second order conservative Fromm scheme [31] applied to (2-5) which results in the following discretisation

$$\begin{aligned} q_j^{n+1} &= q_j^n - \alpha\Delta_{j-\frac{1}{2}} f \\ &\quad -\frac{\alpha}{4}\left((1-\alpha q_{j+\frac{1}{2}})\Delta_{j+\frac{1}{2}} f - (1-\alpha q_{j-\frac{3}{2}})\Delta_{j-\frac{3}{2}} f\right) \end{aligned} \tag{2-10}$$

with $q_{j+\frac{1}{2}} = (q_{j+1} + q_j)/2$. In figure 2-2 the exact solution to (2-5) with initial condition (2-9) is compared to the numerical solution obtained with the schemes in equations (2-7) and (2-10). Although Fromm's scheme is attractive for smooth solutions due to its conservation property and second-order accuracy in time and space, it leads to unwanted overshoots and undershoots near shocks. Godunov's scheme on the other hand displays the proper monotonic behavior but is only of first order. The essential difference between these two methods can be traced back to the fact that Godunov's scheme is a total-variation diminishing (TVD) scheme in the sense of Harten [39] which means that

$$TV(q^{n+1}) \leq TV(q^n) \tag{2-11}$$

with $TV(q^n) = \sum_j |q_{j+1}^n - q_j^n|$. In Ref. [39] it has been proven that TVD-schemes are monotonicity preserving which ensures that no numerical oscillations will occur near the shock.

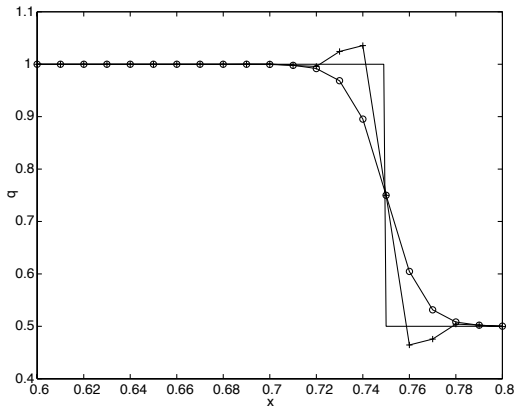


Figure 2-2: The solution q after 50 time steps. The $\circ$'s indicate the solution obtained with (2-7), the $+$'s indicate the solution obtained with (2-10) and the solid line represents the exact solution with initial condition (2-9).

The previous examples show that even for the simple one-dimensional Burgers equation the numerical schemes should possess certain features in order to correctly capture a shock. Although Godunov's scheme is TVD it is only of first order in time and space. For computations which demand a higher accuracy such as e.g. the viscous flow around an airfoil where an accurate estimation of e.g. the drag coefficient is desirable, one would like higher order schemes with the TVD property. Godunov proved [36] that it is not possible to construct a linear monotonicity preserving difference scheme which is of higher order. A lot of work has been performed on the construction of higher order monotonicity preserving schemes which necessarily have to be nonlinear. Examples of such schemes which are commonly used are ENO (Essentially Non-Oscillatory) schemes [40] and MUSCL (Monotonic Upwind Scheme for Conservation Laws) schemes [56]. Both types of methods incorporate a specific interpolation technique on the state vector which limits the gradient of the solution between adjacent cells (see section 2.3.1). In Ref. [15] no fully converged steady state for the two-dimensional inviscid Burgers equation was obtained with a typical ENO-scheme and hence the MUSCL scheme, which did not exhibit this problem, is incorporated in this thesis. Details of the MUSCL scheme will be described in the next section.

2.3 Spatial Discretisation

In this section we will define the numerical methods used for the spatial discretisations. The Navier-Stokes equations contain first and second order spatial derivatives which correspond to inviscid and viscous terms respectively. In the following sections the discretisation of these derivatives will be discussed by considering a scalar one dimensional differential equation. The extension of the discretisation to a system of equations or more dimensions is straightforward and will be outlined in the corresponding chapters.

2.3.1 Inviscid terms

As described in the previous section special properties of the numerical scheme are required if shocks are present in inviscid flow or if large gradients in the solution arise in case of viscous flow. The construction of a conservative numerical scheme is quite natural if one starts with the integral formulation of the conservation law applied to a certain control volume as in equation (2-2). This type of method is generally referred to as a finite volume method. Consider a grid with points $x_0, \ldots, x_N$ such that x_0 and x_N lie on the boundary and define the control volume as the intervals $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ with $x_{j+\frac{1}{2}} = (x_j + x_{j+1})/2$. For any control volume the integral formulation yields

$$\frac{d}{dt}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} q dx + f(q(x_{j+\frac{1}{2}}, t)) - f(q(x_{j-\frac{1}{2}}, t)) = 0 \tag{2-12}$$

where the flux function f contains only the inviscid flux. The integral can be approximated by $q_j \Omega_j$, with the conventional choice $\Omega_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$. In this way the spatial discretisation can be written as

$$\Omega_j \frac{dq_j}{dt} + h_{j+\frac{1}{2}} - h_{j-\frac{1}{2}} = 0 \tag{2-13}$$

where $h_{j+\frac{1}{2}}$ is a numerical approximation of the flux in $x_{j+\frac{1}{2}}$. Traditionally, the discrete flux is split into two parts, the central part and the dissipative part (see e.g. [15]), i.e.

$$h_{j+\frac{1}{2}} = c_{j+\frac{1}{2}} - d_{j+\frac{1}{2}} \tag{2-14}$$

where the central part is defined as

$$c_{j+\frac{1}{2}} = \frac{1}{2}\left(f(q_{j+1}) + f(q_j)\right). \tag{2-15}$$

The difference between several types of discretisations is now determined by the dissipative part $d_{j+\frac{1}{2}}$ [15].

Roe's dissipation model

For accuracy and efficiency reasons we use a higher order scheme. The discretisation method we adopt is based on the first order upwind scheme of Roe [73] which belongs to the class of approximate Riemann solvers [57]. Since Roe's scheme forms the basis of our spatial discretisation we first discuss this scheme in some detail after which we elaborate on the MUSCL (Monotonic Upwind Scheme for Conservation Laws) interpolation technique to obtain a higher order method. One of the main features of Roe's method is the incorporation of the characteristic wave velocities. Consider the general nonlinear equation given by

$$\frac{\partial q}{\partial t} + \frac{\partial f(q)}{\partial x} = 0. \tag{2-16}$$

Application of Roe's scheme gives:

$$\Omega_j \frac{dq_j}{dt} + a^-_{j+\frac{1}{2}}(q_{j+1} - q_j) + a^+_{j-\frac{1}{2}}(q_j - q_{j-1}) = 0 \tag{2-17}$$

where $a_{j+\frac{1}{2}}$ is an approximation of the characteristic wave speed $f'(q)$ at $x_{j+\frac{1}{2}}$, $a^- = \min(a, 0)$ and $a^+ = \max(a, 0)$. Roe's scheme can be written in the form of equation (2-13) by defining the dissipative flux as:

$$d_{j+\frac{1}{2}} = \frac{1}{2}\psi(a_{j+\frac{1}{2}})(q_{j+1} - q_j) \tag{2-18}$$

where $\psi(a) = |a|$ which is called the numerical viscosity coefficient. The term numerical viscosity can be clarified if we apply Roe's scheme to the linear advection equation,

$$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = 0 \tag{2-19}$$

where a is a real number. For an equidistant grid this results in:

$$\frac{dq_j}{dt} + \frac{a(q_{j+1} - q_{j-1})}{2\Delta x} = \psi(a)\frac{q_{j+1} - 2q_j + q_{j-1}}{2\Delta x} \tag{2-20}$$

with $\Delta x = \Omega_j$. Using Taylor expansions this can be rewritten as:

$$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = \frac{1}{2}\Delta x\psi(a)\frac{\partial^2 q}{\partial x^2} + \mathcal{O}\left((\Delta x)^2\right) \tag{2-21}$$

The leading order term on the right-hand side contains the characteristic second order spatial derivative of q with a positive sign, which is a dissipative term. This viscous contribution arises due to the numerical discretisation

and since $\psi(a) = |a|$ is positive, $\frac{1}{2}\psi(a)$ is referred to as numerical viscosity coefficient.

A priori there remain a large number of possibilities for $a_{j+\frac{1}{2}}$ such as e.g. $a_{j+\frac{1}{2}} = \frac{1}{2}(a_j + a_{j+1})$. Let $a_{j+\frac{1}{2}}$ be an approximation of the characteristic wave speed. Then a Roe type linearisation exists if [73]:

1. $a_{j+\frac{1}{2}}(q, q) = a(q)$,
2. $a_{j+\frac{1}{2}}(q_j, q_{j+1})(q_{j+1} - q_j) = f(q_{j+1}) - f(q_j)$

The first condition ensures consistency with the underlying differential equation. The second condition states that the Rankine-Hugeniot relation is obeyed which ensures a proper treatment of shock waves. From these conditions it follows that the approximation of the characteristic wave speed at $x_{j+\frac{1}{2}}$ equals [35]

$$a_{j+\frac{1}{2}} = \begin{cases} \dfrac{f(q_{j+1}) - f(q_j)}{q_{j+1} - q_j} & \text{if} \quad q_{j+1} - q_j \neq 0 \\ a(q_j) & \text{if} \quad q_{j+1} - q_j = 0 \end{cases} \tag{2-22}$$

However, it is well known that Roe's scheme does not result in a proper physical solution if the approximation of the characteristic wave speed equals zero [73, 86]. In this case the scheme permits an expansion shock wave as a steady solution which is physically not correct since this corresponds to a decrease of the total entropy. As suggested in [73] this problem can be avoided if a certain amount of numerical dissipation is added where a sonic point occurs. Within the present definition of the numerical viscosity coefficient ψ this can easily be defined as

$$\psi(a) = \begin{cases} |a| & \text{if} \quad |a| > \delta \\ \delta & \text{if} \quad |a| \leq \delta \end{cases} \tag{2-23}$$

where δ is a positive parameter. The main drawback of this approach is that a suitable value of δ depends on the flow problem.

MUSCL interpolation

Although Roe's scheme is TVD and conservative, it is only first order accurate in space. To arrive at a higher order monotonic upwind method we interpolate the state values with the MUSCL interpolation technique [56] using the minmod-limiter. In this way the characteristic wave speed becomes a function of these new interpolated state values which we denote by q_l and q_r and are

given by:

$$q_l = q_j + \frac{1}{4}\left[(1-\eta)\mathrm{Lim}(\Delta q_{j-\frac{1}{2}}, \omega\Delta q_{j+\frac{1}{2}}) + (1+\eta)\mathrm{Lim}(\Delta q_{j+\frac{1}{2}}, \omega\Delta q_{j-\frac{1}{2}})\right] \tag{2-24}$$

$$q_r = q_{j+1} - \frac{1}{4}\left[(1-\eta)\mathrm{Lim}(\Delta q_{j+\frac{3}{2}}, \omega\Delta q_{j+\frac{1}{2}}) + (1+\eta)\mathrm{Lim}(\Delta q_{j+\frac{1}{2}}, \omega\Delta q_{j+\frac{3}{2}})\right] \tag{2-25}$$

with $\Delta q_{j+\frac{1}{2}} = (q_{j+1} - q_j)$ and the limiter is defined as

$$\mathrm{Lim}(a,b) = \frac{1}{2}\left(\mathrm{sign}(a) + \mathrm{sign}(b)\right)\min(|a|,|b|) \tag{2-26}$$

The parameters η and ω must obey

$$\begin{array}{rl} -1 & \leq \eta \leq \quad 1, \\ 1 & \leq \omega \leq \quad \frac{3-\eta}{1-\eta} \end{array} \tag{2-27}$$

to ensure monotonicity [99]. For the special choice $\eta = \frac{1}{3}$ the scheme is third order accurate in smooth regions. In this thesis we will use: $\eta = \frac{1}{3}$ and $\omega = \frac{3}{2}$, unless stated otherwise. Note that at extrema the limiter function equals zero and thus the scheme becomes first order accurate in space. The central part $c_{j+\frac{1}{2}}$ and the dissipative part $d_{j+\frac{1}{2}}$ in (2-15) and (2-18) are evaluated with the new interpolated state vectors q_l and q_r which correspond to q_j and q_{j+1} respectively.

Properties of the limiter function

In order to illustrate the properties of the limiter function in the MUSCL scheme we once more consider the linear convection equation in (2-19) with $a > 0$. If we apply the MUSCL scheme to the convective part, $a\dfrac{\partial q}{\partial x}$, of the one-dimensional convection equation (2-19) we obtain $a(q_{l,j+\frac{1}{2}} - q_{l,j-\frac{1}{2}})$ after some algebraic manipulations. Substitution of (2-24) gives for this term

$$\frac{a}{\Delta x}\left[1 + \frac{1}{2}\chi(R_j) - \frac{1}{2}\frac{\chi(R_{j-1})}{R_{j-1}}\right](q_j - q_{j-1}) \tag{2-28}$$

with Δx the grid size,

$$\chi(R_j) = \frac{1}{2}[(1-\eta)\mathrm{Lim}(1, \omega R_j) + (1+\eta)\mathrm{Lim}(R_j, \omega)], \tag{2-29}$$

and $R_j = \dfrac{q_{j+1} - q_j}{q_j - q_{j-1}}$. In the following we will investigate for which functions $\chi(R)$ the MUSCL scheme results in a higher order TVD scheme. Clearly, since

$a > 0$, the term between brackets in (2-28) must be positive in order for the scheme to be TVD. This results in the constraint

$$0 \leq \chi(R) \leq 2R \tag{2-30}$$

which gives rise to the upper bound for ω in (2-27). Second order accuracy in space is obtained if $\chi(1) = 1$. It is easily verified that the limiter (2-29)

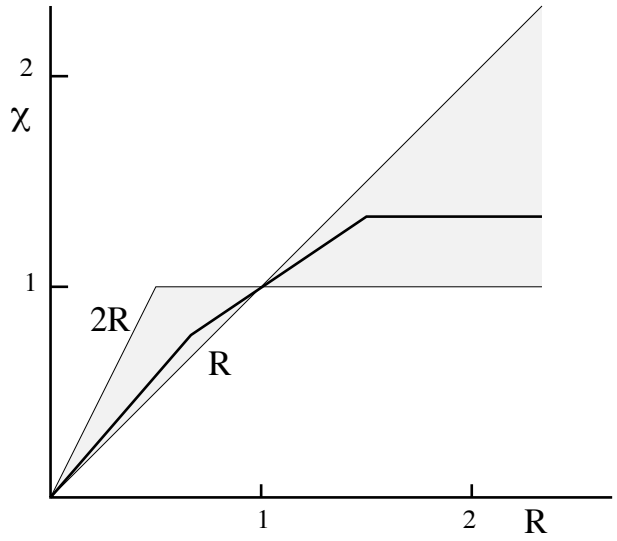


Figure 2-3: Region for the limiter in order to obtain a second-order TVD method (shaded) and the asymmetric minmod limiter with parameters $\omega = \frac{3}{2}$ and $\eta = \frac{1}{3}$ (solid line).

satisfies this condition for all η and ω obeying (2-27). Moreover, Sweby [83] shows that the limiter, $\chi(R)$, must lie between 1 and R in order to comply with the fact that every second order scheme with the present stencil has to be an average of the Lax-Wendroff scheme and the Beam and Warming scheme. Together with the previous constraint (2-30) this yields the shaded region in figure 2-3. This is the well-known second order TVD region [57] in which the limiter must lie. The application of a time stepping scheme may result in an additional constraint on the limiter which will be discussed in more detail in the next section. Taylor series expansion around x_j in (2-28) for the specific choice of $\omega = \frac{3}{2}$ and $\eta = \frac{1}{3}$ yields

$$a\frac{\partial q_j}{\partial x} + \frac{a}{12}\frac{\partial^4 q_j}{\partial x^4}\Delta x^3 \tag{2-31}$$

which shows that the truncation error is of third order in space and has a dissipative nature since the truncation error consists of a fourth order derivative with positive sign.

For a smooth part of a steady state solution, where smooth is defined as sufficiently slowly varying on the scale of the mesh size, the quotient R will be close to one. The lower boundary of the TVD region in figure 2-3 is the original minmod limiter, which corresponds to the choice $\omega = 1$ for all η [83]. The original minmod limiter is non-differentiable only for $R = 1$. If we choose $\omega \neq 1$ the limiter (2-26) becomes asymmetric in the sense that $\mathrm{Lim}(a, \omega b) \neq \mathrm{Lim}(b, \omega a)$. We have plotted the graph of the limiter for the specific choice of $\omega = \frac{3}{2}$ and $\eta = \frac{1}{3}$. It shows that the limiter is a smooth linear function in a neighborhood of $R = 1$. This smoothness property seems to have a positive effect on the convergence property of the numerical scheme for the flow simulated in chapter 4.

2.3.2 Viscous terms

As was shown in the previous section we approximate the first order derivatives with a conservative finite volume method. In the Navier-Stokes equations also viscous terms are present which contain second order derivatives with respect to the spatial variables. In order to obtain a second order accurate approximation of these derivatives within a finite volume context we use the approach as defined in [93]. Consider for instance the one-dimensional viscous Burgers equation given by:

$$\frac{\partial q}{\partial t} + q\frac{\partial q}{\partial x} = \nu \frac{\partial^2 q}{\partial x^2} \tag{2-32}$$

where ν stands for the viscosity. Integration over the control volume $V_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ of the term on the right-hand side yields

$$\int_{V_j} \frac{\partial^2 q}{\partial x^2} dx = \frac{\partial q}{\partial x}\Big|_{j+\frac{1}{2}} - \frac{\partial q}{\partial x}\Big|_{j-\frac{1}{2}} \tag{2-33}$$

The first order derivatives on the right-hand side are approximated by introducing a second control volume e.g. $V_{j+\frac{1}{2}} = [x_j, x_{j+1}]$ for which

$$\frac{\partial q}{\partial x}\Big|_{j+\frac{1}{2}} \approx \frac{1}{|V_{j+\frac{1}{2}}|} \int_{V_{j+\frac{1}{2}}} \frac{\partial q}{\partial x} dx = \frac{1}{|V_{j+\frac{1}{2}}|}(q_{j+1} - q_j) \tag{2-34}$$

with $|V_{j+\frac{1}{2}}| = x_{j+1} - x_j$. This approach results in a second order accurate approximation of the second order derivative as can be inferred from a Taylor

expansion. The generalization to higher dimensions is straightforward by applying Gauss's theorem to an appropriate second control volume in order to obtain the first order derivatives at the vertices of the original control volume (see [93]).

2.4 Time integration

In this section we define the time integration methods that are used and analyzed in later chapters of this thesis. In the previous section we outlined the spatial discretisation method. However, the main focus in this thesis lies on the properties of time integration schemes and we consider the spatial discretisation as fixed. By using the method of lines we can separate the spatial discretisation from the time integration. In this way we obtain from the general evolution equation (2-1) a large set of coupled ordinary differential equations

$$\frac{dq_j}{dt} + F_j(q) = 0 \tag{2-35}$$

where j represents a labelling of the grid nodes, q_j represents a vector of unknown quantities in the specific grid point j, F_j stands for the numerical flux in grid point j described in the previous section and q without a subscript symbolically represents all the state vectors required by the numerical stencil. The common time integration schemes can roughly be divided into two groups: explicit and implicit schemes. Aspects and examples of both types will be outlined in more detail in the next subsections.

2.4.1 Explicit time integration

The use of explicit time integration methods in CFD is wide-spread. The properties of these methods are quite well understood and fairly efficient schemes have been developed with respect to damping and accuracy properties as well as efficiency and memory usage, see e.g. [50, 52]. Also it has been shown that explicit schemes perform very well on parallel platforms, see e.g. [33, 82]. The main disadvantage of explicit schemes lies in the well-known fact that there exists a numerical stability bound on the time step which depends strongly on the grid size.

In this thesis we use an explicit time integration scheme as a reference method. In particular we use the four stage second order compact storage Runge-Kutta scheme defined by

$$\begin{aligned} q_j^{(0)} &= q_j^n \\ q_j^{(k)} &= q_j^{(0)} - \alpha_k \Delta t F_j\left(q^{(k-1)}\right), \quad (k = 1, 2, 3, 4) \\ q_j^{n+1} &= q_j^{(4)} \end{aligned} \tag{2-36}$$

with $q_j^n = q_j(t)$, $q_j^{n+1} = q_j(t + \Delta t)$ and the coefficients $\alpha_1 = 1/4$, $\alpha_2 = 1/3$, $\alpha_3 = 1/2$ and $\alpha_4 = 1$. Although there is a wide family of suitable Runge-Kutta methods that could have been used as a useful reference method we have chosen this Runge-Kutta scheme for its low memory overhead and favorable stability region.

The time step Δt is bounded for stability reasons and depends on the grid size and the eigenvalues of the flux Jacobi matrix of the numerical flux F. For the linear convection equation with $a > 0$ it can be derived with Von Neumann stability analysis that the time step is bounded by

$$\Delta t \leq \frac{\sigma \Delta x}{a} \tag{2-37}$$

where σ stands for the CFL (Courant-Friedrichs-Levy) number which depends on the specific choice of the explicit time integration method and the spatial discretisation. The extension to multiple dimensions and systems of equations is straightforward and will be described in the corresponding chapters. In the case of a steady flow the accuracy of the time stepping scheme is not important because the time derivative vanishes when the steady state is obtained and the quantities in each node can be advanced in time according to their own local stability time step Δt_j. This technique is called local time stepping and a considerable acceleration with respect to CPU-time may be obtained compared to the case where all quantities are advanced with a fixed global stability time step, especially on highly non-uniform grids. For time dependent problems this acceleration technique is not possible and the quantities in each grid point have to be advanced in time with the same time step $\Delta t \leq \min_j \Delta t_j$.

2.4.2 Implicit time integration

For flows where the dominant physical time scales are relatively large, the stability time step may be much smaller than the time step necessary for accuracy. An example of such a flow is e.g. the viscous flow around a slowly pitching airfoil where the turbulence is modelled within the Reynolds-averaged Navier-Stokes framework. In order to resolve the boundary layer around the airfoil correctly very small grid cells have to be used which results in a very small stability time step for explicit methods. It is possible to overcome this stability restriction if one uses an implicit scheme. The difference between implicit and explicit schemes is that the numerical flux in (2-35) is evaluated at the new time level which explains the designation implicit since the desired new solution is only specified as the solution to a system of equations.

A large class of implicit schemes can be written as:

$$q_j^{n+1} = \sum_{i=n-k}^{n} \alpha_i q_j^i - \Delta t \sum_{i=n-k}^{n+1} \beta_i F_j(q^i) \tag{2-38}$$

where the superscript indicates the time level and α_i and β_i are constants with $\beta_{n+1} \neq 0$. For convenience we take the time step Δt constant. It is possible to derive similar multi-step schemes which use a variable time step but then the coefficients α and β become functions of the time steps. Examples of commonly used implicit schemes are the first order Euler Backward scheme given by

$$q_j^{n+1} - q_j^n + \Delta t F_j(q^{n+1}) = 0 \tag{2-39}$$

the second order Crank-Nicolson scheme given by

$$q_j^{n+1} - q_j^n + \frac{1}{2}\Delta t F_j(q^n) + \frac{1}{2}\Delta t F_j(q^{n+1}) = 0 \tag{2-40}$$

and the second order BDF2 (backward differentiation formula) scheme given by

$$\frac{3}{2}q_j^{n+1} - 2q_j^n + \frac{1}{2}q_j^{n-1} + \Delta t F_j(q^{n+1}) = 0 \tag{2-41}$$

The main advantage of schemes such as (2-39)-(2-41) is that they are A-stable (for linear F) which means that their stability region contains the entire complex half plane with negative real part. Therefore, theoretically the time step is not bounded for numerical stability reasons. The difference between the Crank-Nicolson scheme and the BDF2 scheme is that the latter is a multi-step scheme whereas the first is a single-step scheme. A disadvantage of multi-step schemes is the occurrence of parasites [4]. For an A-stable multi-step method these parasites can be present but their norm remains bounded.

All implicit schemes of the type (2-38) can be written in the form

$$H_j(q^{n+1}) = g_j \tag{2-42}$$

with $H_j = q_j^{n+1} + \Delta t \beta_j F_j(q^{n+1})$ and $g_j = \sum_{i=n-k}^{n} \alpha_i q_j^i - \Delta t \sum_{i=n-k}^{n} \beta_i F_j(q^i)$. The right-hand side of (2-42) depends on previous solutions only and is therefore fixed during one time step. In general (2-42) represents a large set of coupled nonlinear algebraic equations. The desired solution of (2-42) corresponds to a fixed point of (2-38). For practical problems the number of equations is very large and direct solution methods cannot be used. Therefore iterative methods such as e.g. Newton iteration have to be used to solve (2-42). Newton

iteration involves the Jacobi matrix of H. For large problems which arise in CFD the computational cost of determining, storing and inverting the numerical Jacobi matrix of H is too high because large stencils are required to obtain an accurate solution at an acceptable number of grid points. A possible way to avoid some of these problems is to approximate the Jacobi matrix of H see e.g. [102] which is referred to as quasi-Newton iteration. Although the computational cost per iteration decreases significantly, the quadratic convergence of exact Newton iteration in the vicinity of the fixed point is lost. Additionally, convergence problems with the Newton and quasi-Newton method may occur if the changes in the solution q^n and q^{n+1} are large which typically arise if Δt is relatively large. An alternative way to tackle this problem follows the ideas of Jameson [48] where a pseudo time is added to (2-42) which yields

$$\frac{dv_j}{d\tau} + H_j(v) = g_j \tag{2-43}$$

The resulting steady state solution of (2-43) corresponds to the desired solution q^{n+1} of (2-42). To solve (2-43) we use the Euler backward scheme in pseudo time. Although Euler backward is only a first order accurate scheme this choice has no effect on the temporal accuracy of the solution q^{n+1} of (2-42) since the steady state solution of (2-43) is a solution of (2-42) for which the temporal accuracy is determined by the choice of coefficients α and β in (2-38). Applying Euler backward to (2-43), the following system has to be solved for each pseudo time level

$$\left(\frac{I}{\Delta\tau} + \frac{\partial H}{\partial v}(v^k)\right)\Delta v = g - H(v^k) \tag{2-44}$$

where $\Delta v = v^{k+1} - v^k$, the superscript k indicates the pseudo time level and $\frac{\partial H}{\partial v}$ is a symbolic representation of the numerical Jacobi matrix of H defined in (2-42). The iteration scheme in (2-44) corresponds to exact Newton iteration for infinite $\Delta\tau$ and exact numerical representation of the Jacobi matrix. The pseudo time step $\Delta\tau$ can be viewed as a relaxation parameter which for small values of $\Delta\tau$ increases the convergence range of the quasi-Newton iteration. Note that due to the so-called delta formulation used in (2-44) it is possible to approximate the Jacobi matrix without affecting the solution because the proper solution of (2-42) is obtained if the norm of Δv converges to zero.

2.4.3 TVD-property

In section 2.3.1 we remarked that the time stepping scheme may give rise to an additional constraint on the limiter used in the MUSCL scheme for the spatial discretisation. In Refs. [57, 83] the time stepping method results in an upper bound of $\chi \leq 2$ arising from stability requirements. Harten showed

in Ref. [39] that there appears to be an analogy between stability and TVD for explicit time integration schemes. In order for the scheme to be TVD also a CFL-like criterion has to be satisfied. The question which arises is whether such an analogy also exists for implicit time integration schemes. To study this we discretize the linear convection equation in (2-19) for $a > 0$ with the Euler backward scheme in time. In combination with the MUSCL-scheme for the spatial discretisation the overall discretisation results in

$$q_j^{n+1} + \xi_j(q_j^{n+1} - q_{j-1}^{n+1}) = q_j^n \tag{2-45}$$

with

$$\xi_j = \frac{a\Delta t}{\Delta x}\left[1 + \frac{1}{2}\chi(R_j) - \frac{1}{2}\frac{\chi(R_{j-1})}{R_{j-1}}\right]. \tag{2-46}$$

If we evaluate (2-45) for j and $j-1$ and subtract these two expressions this yields

$$(1 + \xi_j^{n+1})(q_j^{n+1} - q_{j-1}^{n+1}) = (q_j^n - q_{j-1}^n) + \xi_{j-1}^{n+1}(q_{j-1}^{n+1} - q_{j-2}^{n+1}) \tag{2-47}$$

It is not hard to show that the scheme is TVD if ξ_j is positive, which corresponds to the positivity demand on the coefficients in [39]. Summing (2-47) over $-\infty < j < \infty$ and shifting the index of the second term on the right-hand side gives

$$TV(q^{n+1}) \leq TV(q^n). \tag{2-48}$$

So, if the limiter function lies within the TVD-region (see figure 2-3) the magnitude of the time step does not affect the TVD-property. This seems to agree with the fact that the Euler backward scheme is A-stable. However, if we perform the same analysis for the Crank-Nicolson scheme, which is also A-stable, we obtain

$$\begin{aligned}(1 + \tfrac{1}{2}\xi_j^{n+1})(q_j^{n+1} - q_{j-1}^{n+1}) \quad &= \quad (1 - \tfrac{1}{2}\xi_j^n)(q_j^n - q_{j-1}^n) + \\ &\quad \tfrac{1}{2}\xi_{j-1}^{n+1}(q_{j-1}^{n+1} - q_{j-2}^{n+1})\end{aligned} \tag{2-49}$$

If we again assume that all coefficients are positive it is possible to show (as performed above) that the Crank-Nicolson scheme is TVD if $0 \leq \xi_j \leq 2$, which leads to a restriction on the time step given by

$$\Delta t \leq \frac{2\Delta x}{a}\frac{1}{1 + \max(\chi(R))} \tag{2-50}$$

It follows from (2-50) that the restriction on the time step becomes more severe in case the maximum value of the limiter increases. The least severe restriction

occurs if the limiter is zero which corresponds to the first order Roe scheme. However, we observe that the positivity requirement on the coefficients is only a sufficient condition for TVD and the analogy between stability and TVD for the Crank-Nicolson scheme is not yet clear.

To further study the TVD-property we consider a numerical simulation of the one-dimensional Burgers equation described in section 2.3.1 and compare the Euler backward and the Crank-Nicolson scheme for a time step which is five times larger than the time step used in section 2.2 and therefore violates the inequality in (2-50). For the spatial discretisation we use Roe's first order scheme which has the least severe restriction on the time step according to (2-50). In figure 2-4 the solution corresponding to the Euler backward scheme

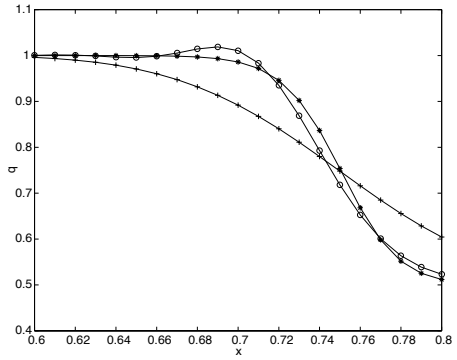


Figure 2-4: The solution q after 10 time steps with $\Delta t = \frac{10}{3}\Delta x$. The $\circ$'s indicate the solution obtained with the Crank-Nicolson scheme and the $+$'s indicate the solution obtained with the Euler backward scheme. The *'s represent the solution with the Crank-Nicolson scheme at the same time level but obtained with $\Delta t = \frac{2}{3}\Delta x$.

is shown to remain monotonic. The solution obtained with the Crank-Nicolson scheme shows wiggles near the shock which illustrates that the Crank-Nicolson scheme affects the TVD-property of the spatial discretisation if the time step is too large. A similar calculation with the BDF2 scheme reveals that the TVD-property is violated for large time steps as well. Therefore, there does not appear to exist an analogy between A-stability and TVD for general A-stable methods.

2.5 Linear solvers

The application of a Newton or quasi-Newton iteration method such as (2-44) to solve equation (2-42) results in a large linear system to be solved in every

pseudo time step. In the following the linear system will be denoted by:

$$Ax = b \tag{2-51}$$

where $A \in \mathbb{R}^{n \times n}$ and n denotes the dimension of A. For finite volume methods such as defined in section 2.3 the Jacobi matrix of the numerical flux results in a sparse matrix for which the sparsity pattern is determined by the stencil of the discretisation method. The magnitude of n is the number of equations in the underlying differential equation multiplied by the number of grid points. The inversion of such a large sparse system with a classical Gauss elimination procedure, which is an $\mathcal{O}(n^3)$ method, is too expensive with respect to CPU time. Also extra memory allocation is necessary since Gauss elimination generates a complete fill-in of the original sparse matrix. Iterative methods in general require less CPU time and often require no or little extra storage of matrix elements. In this section we will outline two classes of iterative solvers. Their effect on the convergence of the nonlinear problem (2-42) in the application of an unsteady flow problem will be studied in chapter 3.

2.5.1 Basic iterative methods

The general idea behind iterative methods is, given an approximation x^k of the solution to (2-51) at iteration level k, to obtain a better approximation x^{k+1} in an efficient manner. A specific class of iterative methods can be obtained by splitting the matrix A as

$$A = M - N, \tag{2-52}$$

where M is a non-singular matrix and M^{-1} can be determined efficiently. Corresponding to this splitting the iterative scheme is given by

$$Mx^{k+1} = Nx^k + b. \tag{2-53}$$

It is easily verified that the solution x^∞ corresponds to the solution of (2-51) if the scheme converges. Whether or not the method converges depends on the eigenvalues of $M^{-1}N$. Define the spectral radius of a matrix A by

$$\rho(A) = \max\{|\lambda| \ : \lambda \in \text{ spectrum of } A\}. \tag{2-54}$$

Convergence of the iterative method is assured if the spectral radius of $M^{-1}N$ is less than one [37]. A condition which guarantees $\rho(M^{-1}N) < 1$ (see [37]) is strict diagonal dominance which means that

$$\rho(M^{-1}N) \leq \max_{1 \leq i \leq n} \sum_{j=1, j \neq i}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| < 1 \tag{2-55}$$

where a_{ij} is a matrix element of A in row i and column j. Some classical splittings can easily be defined if we rewrite the matrix A by

$$A = L + U + D \tag{2-56}$$

where L is the lower triangular part, U is the upper triangular part and D is the diagonal of A. The Jacobi-method for example is obtained with $M = D$ and $N = -(L+U)$ and the Gauss-Seidel method is obtained with $M = D+L$ or $M = D+U$ and $N = -L$ or $N = -U$. The convergence speed of these methods depends on $\rho(M^{-1}N)$ and becomes very low when the spectral radius is close to unity. In this thesis we will only use the basic forms because it turns out that for the simulations performed in this thesis these methods compare favorably, with respect to CPU time, to more advanced methods.

2.5.2 Krylov methods

Another class of iterative methods is based on the concept of a Krylov space. In contrast to the basic iterative solvers of the previous section for which the convergence rate is constant Krylov methods may exhibit super-linear convergence [92]. Therefore Krylov methods may become very attractive if the linear system has to be solved accurately. There exists an enormous amount of literature on Krylov methods and applications of Krylov methods. Here we briefly describe the ideas of Krylov methods and refer to literature for more details [76]. Consider the basic iterative scheme in equation (2-53). If the residual at iteration level k is defined as

$$r_k = b - Ax^k, \tag{2-57}$$

the solution x^k lies in the following space

$$x^k = x^0 + \langle M^{-1}r_0, (M^{-1}A)M^{-1}r_0, \ldots, (M^{-1}A)^{k-1}M^{-1}r_0 \rangle \tag{2-58}$$

which can be derived by recursive substitution of (2-53) and the use of (2-52). The linear subspace in (2-58) is called the Krylov-space of dimension k and is denoted by $K^k(M^{-1}A; M^{-1}r_0)$. It is clear from (2-53) that the memory overhead in storing the Krylov space becomes significant if many iterations are required to solve the linear system (2-51). Therefore the main practical question is whether it is possible to design a method such that a minimal number of vectors have to be stored to construct the new solution x^k. Additionally, one would like to know the quality of the solution x^k with respect to the solution x of (2-51) i.e. is it possible to construct a new approximation x^k such that $||x^k - x||$ is minimal in some norm.

If one assumes that the matrix A is symmetric and positive definite it is possible to construct a Krylov method with the above mentioned optimality

conditions. This method is the so called CG (Conjugate gradient) method. It has been shown in [37] that the residuals $M^{-1}r_0 \dots M^{-1}r_{k-1}$ span the Krylov space. There are two features that make the CG method special. First, the new solution x^k obeys an optimality condition. Second, only $M^{-1}r_{k-1}$ has to be stored. However, for the applications of interest in this work the matrix A is not symmetric and positive definite. For general non-singular matrices A it is shown in [29] that it is impossible to obtain a Krylov method which has both properties of the CG method. So either the method satisfies an optimality condition and long recurrences (large storage) or no optimality condition and short recurrences. Here we use only one type of Krylov method for a general matrix A. Details of this method will be outlined next.

The GMRES method

In chapter 3 we use the GMRES (General Minimized Residual) [76] method which satisfies an optimality condition and necessarily possesses a long recurrence which implies that the amount of work per iteration and the required memory grows with the number of iterations. In this method, Arnoldi's method is used to construct an orthonormal basis $\{v_1, \dots, v_k\}$ for the Krylov subspace $K^k(M^{-1}A; M^{-1}r_0)$. The Arnoldi method is described by

1. Choose x^0 and compute $r_0 = M^{-1}(b - Ax^0)$ and $v_1 = r_0/||r_0||_2$
2. for $j = 1, \dots, k$ do
   $v_{j+1} = M^{-1}Av_j$
   for $i = 1, \dots, j$ do
   $h_{i,j} = v_{j+1}^T v_i,\ v_{j+1} = v_{j+1} - h_{i,j}v_i$
   end
   $h_{j+1,j} = ||v_{j+1}||_2,\ v_{j+1} = v_{j+1}/h_{j+1,j}$
   end

The elements $h_{i,j}$ form a $(k+1) \times k$ Hessenberg matrix which we denote by H_k. The approximate solution $x^k = x^0 + z^k$ is determined by solving the minimization problem given by

$$||r_k||_2 = ||b - Ax^k||_2 = \min_{z \in K(M^{-1}A; M^{-1}r_0)} ||r_0 - Az||_2 \tag{2-59}$$

It has been shown in [76] that $z^k = V_k y^k$ where V_k is the matrix with columns v_j and y^k is the solution of the following least squares problem

$$||\beta e_1 - H_k y^k||_2 = \min_{y \in \mathbb{R}^k} ||\beta e_1 - H_k y||_2 \tag{2-60}$$

where $\beta = ||r_0||_2$ and e_1 is the first unit vector in $\mathbb{R}^{k+1}$. The GMRES method cannot break down since the solution x of (2-51) is obtained if $h_{j+1,j} = 0$. For

more details on the GMRES-method we refer to [76]. Although there is a wide variety of Krylov methods for general matrices we have selected the GMRES method because it cannot break down and is very robust.

2.6 Multigrid

In order to perform a time step implicitly a large set of coupled nonlinear equations has to be solved which implies a considerable cost. It was shown in section 2.4.2 that the problem to find the desired root of (2-42) can be equivalently expressed as the differential equation (2-43) of which the steady state solution corresponds to the desired solution of (2-42). A lot of research has already been directed toward convergence acceleration techniques for steady flow problems. One of the more elegant acceleration techniques is the so called multigrid method for which Brandt [13] and Hackbusch [38] are known for their pioneering work.

If one uses e.g. a Runge-Kutta scheme to solve (2-43) it is well known that high frequency components of the error are reduced rapidly whereas low frequency components of the error are reduced only very slowly. Hence, the general idea of multigrid is to transfer the error, which remains after some iterations with the relaxation method, onto a coarser grid. The dominant low frequencies of the error on the fine grid appear as higher frequency contributions on the coarse grid which in turn can be damped more efficiently by applying the relaxation method on the coarse grid. In the next section the way in which solutions, error and operators can be transferred onto a coarser grid is described very briefly for a two-grid model. For a more elaborate discussion on multigrid we refer to the work of Hackbusch [38] and Wesseling [97].

2.6.1 Two-grid model

In this section the general ideas of multigrid methods will be explained using a two-grid model. Consider the general discrete set of equations

$$N^h(q^h) = g^h \tag{2-61}$$

where h represents the mesh width, N^h in general represents a nonlinear discrete operator (here the discrete operator H in (2-42)) in which also the boundary conditions are incorporated, g^h is the representation of the right-hand side in (2-42) on the present grid and q is the solution vector. In order to solve (2-61) a wide variety of relaxation methods can be used such as e.g. a time stepping scheme or Newton iteration. After a few relaxations we then would get

$$\widetilde{q}^h = q^h - v^h \tag{2-62}$$

where $\widetilde{q}^h$ is an approximation of the numerical solution q^h and v^h is the error. Note that since the exact solution is unknown the error is unknown as well. The residual of the solution $\widetilde{q}^h$ can be defined as

$$r^h = g^h - N^h(\widetilde{q}^h) \tag{2-63}$$

Substitution of (2-62) and (2-63) into (2-61) yields

$$N^h(\widetilde{q}^h + v^h) = N^h(\widetilde{q}^h) + r^h \tag{2-64}$$

In this equation v^h is the only unknown and if the relaxation method damps only the high frequencies efficiently, v^h contains only low frequency components. In order to find v^h this equation is next transferred onto a coarser grid with mesh size H and the basic equation is represented by

$$N^H(q^H) = g^H \tag{2-65}$$

where N^H is the coarse grid approximation to N^h and g^H the approximation of the right-hand side of (2-64). The right-hand side is calculated by

$$g^H = N^H(I_h^H \widetilde{q}^h) + I_h^H(r^h) \tag{2-66}$$

where I_h^H denotes a restriction operator from the fine to the coarse grid. The restriction operator may be defined differently for the solution and the residual. The representation of the right-hand side of (2-64) is not unique and our specific choice will be discussed shortly. The initial guess for the solution of (2-65) is obtained by restriction of $\widetilde{q}^h$ onto the coarse grid. In the two grid model it is assumed that equation (2-65) is solved exactly and we denote this solution by $\widetilde{q}^H$. A new approximation of q^h on the fine grid is then obtained by prolongation of the correction term obtained on the coarse grid to the fine grid as follows

$$\widetilde{\widetilde{q}}^h = \widetilde{q}^h + I_H^h(\widetilde{q}^H - I_h^H(\widetilde{q}^h)) \tag{2-67}$$

where I_H^h denotes the prolongation operator from the coarse grid to the fine grid. The advantage of the current approximation of the right-hand side of (2-64) is associated with the coarse grid correction in (2-67). If the solution of (2-61) is obtained the prolonged correction equals zero.

In Ref. [38] it has been shown that two properties have to be satisfied to ensure convergence of the two-grid model: the smoothness property and the approximation property. The smoothness property expresses the requirement that the residual r^h in (2-63) and the error v^h in (2-62) should contain only frequencies which can be represented sufficiently accurate on the coarser grid and thus the high frequencies have to be damped sufficiently by the relaxation

method. The approximation property requires that equation (2-61) has to be represented sufficiently accurate on the coarse grid, which results in the following constraint on the restriction and prolongation operators. In Refs. [38, 97] it is shown that the operators involved should satisfy

$$m_r + m_p > 2m \tag{2-68}$$

where m_r and m_p denote the order of accuracy of the restriction and prolongation operators and $2m$ denotes the order of the underlying differential operator.

The generalization to multiple grids is straightforward. The nonlinear system in equation (2-65) on the coarse grid can be transferred onto yet a coarser grid in a similar way. In this way a sequence of grids can be traversed in order to solve (2-61) where the amount of computational work obviously decreases considerably per coarsening. For more details we refer to [97].

Chapter 3

The shock tube

3.1 Introduction

The main purpose of this chapter is to study the performance of the linear solvers described in the previous chapter and their influence on the overall convergence of the nonlinear system involved in the implicit time integration scheme during one time step for a nonlinear unsteady test case. As a model we numerically solve the one-dimensional Euler equations for the well known unsteady shock tube problem. This flow contains several physical phenomena which also occur in more complicated flows such as a shock wave and a contact discontinuity. For the time integration we use the second order implicit Crank-Nicolson scheme. In section 3.2 we present the one-dimensional Euler equations and describe the shock tube problem. The spatial discretisation is discussed in section 3.3. In section 3.4 the numerical approximation of the flux Jacobi matrix is outlined. Numerical results concerning the performance of the linear solvers are presented in section 3.5. As mentioned in the previous chapter large time steps may affect the TVD-property of the spatial discretisation. However, for systems of equations the TVD property is not well defined. Therefore in section 3.6 we discuss the entropy production of a numerical method which in some sense is equivalent to the TVD property for systems of equations. In section 3.7 the conclusions are summarized.

3.2 Governing equations and problem definition

The equations governing one-dimensional inviscid flow are the one-dimensional Euler equations. In conservative variables these read

$$\frac{\partial q}{\partial t} + \frac{\partial f(q)}{\partial x} = 0 \tag{3-1}$$

with

$$q = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix} \quad \text{and} \quad f = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{bmatrix} \tag{3-2}$$

where ρ represents the density, E the total energy density, p the pressure and u the velocity. The constitutive equation for the pressure is given by

$$p = (\gamma - 1)(E - \frac{1}{2}\rho u^2) \tag{3-3}$$

where γ is the adiabatic gas constant which we take $\gamma = 1.4$.

The shock tube problem [77] can be described by analogy with the following physical setting. Consider a tube filled with gas and divided into two compartments by a membrane in the middle of the tube. On both sides of the membrane the gas is in equilibrium with a higher density and pressure in one half of the tube. Then the membrane is removed instantaneously and both states start to mix towards a new equilibrium. The temporal behavior during the initial process turns out to exhibit special features which also occur in more complicated aerodynamical configurations and make the shock tube problem an interesting study object. The flow possesses three distinct waves separating regions in which the state variables are constant, see e.g. figures 3-1 and 3-2. Across two of these regions discontinuities occur in some of the state variables. First a shock wave across which the density and the pressure are discontinuous propagates into the region of lower pressure. This is followed by a contact discontinuity across which the density is discontinuous. The third wave propagates in the opposite direction and there is a continuous transition between the constant states on either side of it. This wave is called a rarefaction wave since the density of the gas decreases as the wave passes through. Another advantage of the present shock tube problem is that there is a known analytical solution used for reference.

3.3 Spatial discretisation

In section 2.3.1 the flux splitting scheme of Roe was described for a scalar equation which we now extend to a system of equations. In relation to equation (2-22) two requirements had to be satisfied by the approximation of the characteristic wave speed. For a scalar equation this wave speed was defined as the derivative of the flux function. While for a system of equations the derivative of the flux vector with respect to the state vector q is a matrix and an additional requirement has to be satisfied which originates from the hyperbolicity of the Euler equations. Let $A(q) = \nabla_q(f(q))$, then a Roe type

linearisation of the Euler equations exists if the following requirements are met [35, 73]

1. $A_{j+\frac{1}{2}}(q,q) = A(q)$
2. $A_{j+\frac{1}{2}}(q_l,q_r)(q_r - q_l) = f(q_r) - f(q_l)$
3. $A_{j+\frac{1}{2}}$ is not defect, and has real eigenvalues and eigenvectors

where $A_{j+\frac{1}{2}}$ is the numerical flux Jacobi matrix. The third requirement preserves the hyperbolicity of the Euler equations and the state vectors q_l and q_r represent the left and right state vectors as described in section 2.3.1. The numerical flux $h_{j+\frac{1}{2}}$ in (2-13) on the control volume edge is approximated by

$$h_{j+\frac{1}{2}} = \frac{1}{2}(f(q_l) + f(q_r)) - \frac{1}{2}|A_{j+\frac{1}{2}}|(q_r - q_l) \tag{3-4}$$

The absolute value of the flux Jacobi matrix A is defined as $|A| = R|\Lambda|L$, where R and $L = R^{-1}$ are right and left eigenvector matrices of A and $|\Lambda|$ is a diagonal matrix containing the absolute values of the eigenvalues of the matrix A. The matrix $A_{j+\frac{1}{2}}$ is evaluated using the so-called Roe state variables which are determined by calculating the primitive variables according to

$$\begin{aligned} \rho_{lr} &= \frac{\rho_l + d\rho_r}{1+d} \\ u_{lr} &= \frac{u_l + du_r}{1+d} \\ H_{lr} &= \frac{H_l + dH_r}{1+d} \end{aligned} \tag{3-5}$$

with $d = \sqrt{\rho_r/\rho_l}$ and H the specific enthalpy defined by $H = (E+p)/\rho$ and thus the flux Jacobi matrix becomes a function of q_{lr} which is the approximation of $q_{j+\frac{1}{2}}$. For the first order Roe scheme q_l and q_r equal q_j and q_{j+1} respectively. To obtain a higher order accuracy the MUSCL technique (see section 2.3.1) is applied to the conservative state variables separately to obtain an improved approximation of q_l and q_r to be used in (3-5).

3.4 Approximation of the Jacobi matrix

In this section we define the numerical approximation of the Jacobi matrix. For the time integration we use the Crank-Nicolson scheme defined in section 2.4.2 where it was shown that a nonlinear system $H(q^{n+1}) = g(q^n)$ has to be solved every time step and H and g are given by

$$\begin{aligned} H_j(q^{n+1}) &= q_j^{n+1} + \tfrac{1}{2}\Delta t F_j(q^{n+1}) \\ g_j(q^n) &= q_j^n - \tfrac{1}{2}\Delta t F_j(q^n) \end{aligned} \tag{3-6}$$

where F represents the numerical flux. The Jacobi matrix of H is symbolically written as

$$\frac{\partial H}{\partial q} = I + \frac{1}{2}\Delta t \frac{\partial F}{\partial q} \tag{3-7}$$

where I denotes the identity matrix and $\frac{\partial F}{\partial q}$ represents the flux Jacobi matrix of the spatial discretisation F. Due to the size of the computational stencil of the higher order MUSCL scheme for the two-dimensional flows considered in chapters 4 and 6 the computational cost of determining, storing and inverting the flux Jacobi matrix of F is very high. For the present one-dimensional test case the MUSCL scheme results in a five-point stencil and a sparse matrix with five bands of 3×3 matrices is obtained. Although the computational costs for the exact inversion of this Jacobi matrix are quite acceptable, we will use the same first order approximation of the flux Jacobi matrix as will be used for the two-dimensional flow problem simulated in chapter 4. The details of this first order approximation are discussed below. Additionally, an exact numerical flux Jacobi matrix is determined for the first order scheme in section 3.4.2. In this way the effect of the first order approximation of the Jacobi matrix can be studied.

3.4.1 Numerical approximation of the flux Jacobi matrix

The approximation of the flux Jacobi matrix of the numerical flux F is based on the first order Roe scheme which has only a three-point stencil compared to the five-point stencil of the MUSCL scheme. The numerical flux, F, for Roe's scheme in a grid point j can be written as:

$$F_j = A^-_{j+\frac{1}{2}}(q_{j+1}, q_j)(q_{j+1} - q_j) + A^+_{j-\frac{1}{2}}(q_j, q_{j-1})(q_j - q_{j-1}) \tag{3-8}$$

The matrices A^+ and A^- are determined by the positive and negative eigenvalues of the matrix A by

$$A = R\Lambda L = R(\Lambda^+ + \Lambda^-)L = R\Lambda^+ L + R\Lambda^- L = A^+ + A^- \tag{3-9}$$

with R and L the right and left eigenvectors of the matrix A and Λ, Λ^+ and Λ^- diagonal matrices containing all the eigenvalues, only the positive eigenvalues and only the negative eigenvalues of A respectively. In order to obtain the flux Jacobi matrix of the numerical flux F, the derivatives of the matrices A^+ and A^- to q_{j-1}, q_j and q_{j+1} have to be determined. Since we already use an approximation of the flux Jacobi matrix for the higher order stencil we additionally assume that the matrices A^+ and A^- are independent of q. In total this results in the following block-tridiagonal approximation of the

Jacobi matrix of H in (3-6), where the diagonal and sub-diagonals contain 3×3 matrices given by:

$$\begin{aligned} D_j &= I + \tfrac{1}{2}\Delta t \left(A^+_{j-\frac{1}{2}} - A^-_{j+\frac{1}{2}} \right) \\ E_j &= \tfrac{1}{2}\Delta t A^-_{j+\frac{1}{2}} \\ W_j &= -\tfrac{1}{2}\Delta t A^+_{j-\frac{1}{2}} \end{aligned} \tag{3-10}$$

where D_j, E_j and W_j stand for the diagonal, east and west contribution to the Jacobi matrix of H corresponding to the grid point x_j.

3.4.2 Numerical flux Jacobi matrix

In order to examine the effect of the first order approximation used for the flux Jacobi matrix in relation to different spatial discretisation scheme, simulations are performed with the third order MUSCL scheme as well as with the first order Roe scheme. It is also possible for the first order scheme to determine a numerical flux Jacobi matrix using the numerical derivative defined by

$$\frac{\partial F_j}{\partial q^i_k} \approx \frac{F_j(q|q^i_k + \epsilon) - F_j(q)}{\epsilon}, \tag{3-11}$$

where j and k represent the grid node points, q represents the state vector, the superscript i denotes the i-th component of the state vector, $q|q^i_k + \epsilon$ denotes that only the i-th component with index k is perturbed, and ϵ is a sufficiently small real number. With this approach we can obtain a full Newton method for the first order scheme and in the vicinity of a root of $H = g$ the nonlinear system should converge quadratically. Using the two approximations of the flux Jacobi matrix it is possible to assess the effect of the approximation of the flux Jacobi matrix for the first order scheme on the number of required Newton iterations in order to solve the system $H = g$.

3.5 Numerical results

In this section we present the numerical results for Sod's shock tube problem [77] for which the analytical solution is known. First we present the numerical solution obtained with the Crank-Nicolson scheme for the time integration in combination with the MUSCL scheme for the spatial discretisation showing that this results in an accurate approximation of the solution to the shock tube problem. Second we study the effect of different linear solution methods on the number of required quasi-Newton iterations associated with the desired accuracy of the solution for one time step if the corresponding linear system is not solved to machine accuracy. We use this information in order to select an appropriate linear solver for the flow applications in the next chapters.

3.5.1 Numerical solution

The initial condition for Sod's shock tube problems is given by

$$q(x) = \begin{cases} q_l & \text{if} \quad 0 \leq x \leq \frac{1}{2} \\ q_r & \text{if} \quad \frac{1}{2} < x \leq 1 \end{cases} \tag{3-12}$$

with $(\rho_l, u_l, p_l) = (1, 0, 1)$ and $(\rho_r, u_r, p_r) = (0.125, 0, 0.1)$. The values of numerical parameters in the numerical method correspond to the values used in [15]. The simulation starts at $t = 0$ and we will compare the numerical solution at $t = 0.15$ with the analytical solution. For accuracy reasons a small time step is needed because the time scale of the physical process is very small. The time step is determined with (2-37) and we take a CFL number of $\sigma = 0.5$. For the one-dimensional Euler equations the wave propagation speed a in (2-37) is approximated by the absolute value of the maximum eigenvalue of the flux Jacobi matrix which corresponds to $|u| + c$, where c denotes the speed of sound. The physical domain is discretized by 160 grid points. Each time step we solve the resulting nonlinear equation $H = g$ with an accuracy of $\epsilon = 10^{-10}$. The specific form of the stopping criterion related to the accuracy is given in (3-15). The entropy fix parameter defined in section 2.3.1 is set to $\delta = 0.02$.

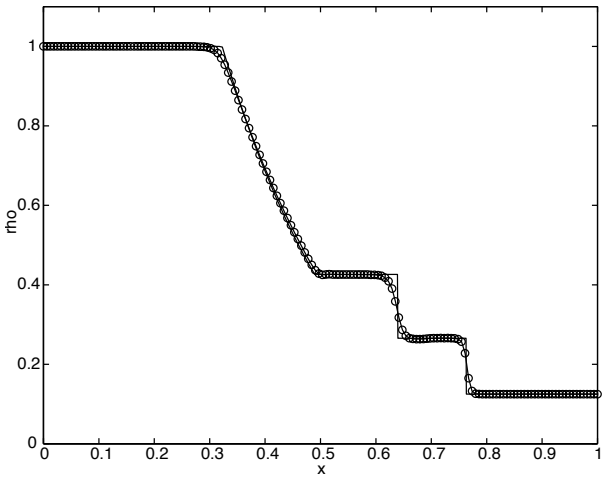


Figure 3-1: The density distribution at $t = 0.15$. The solid line represents the analytical solution and the o's the numerical solution obtained with the Crank-Nicolson scheme in combination with the MUSCL scheme.

In figure 3-1 the numerical density distribution is shown. Clearly, four different constant states for the density can be distinguished as discussed in

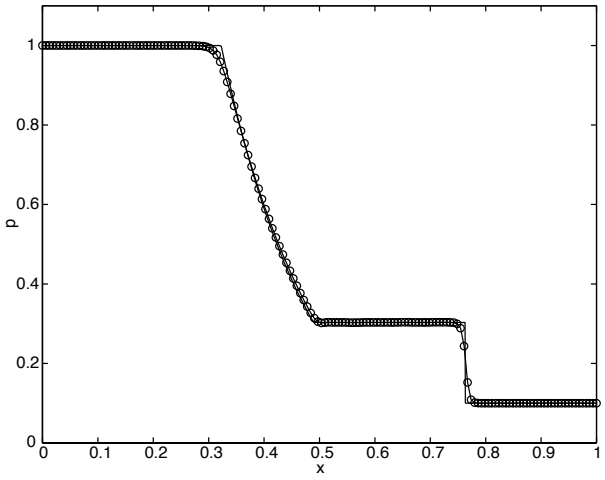


Figure 3-2: The pressure distribution at $t = 0.15$. The solid line represents the analytical solution and the o's the numerical solution obtained with the Crank-Nicolson scheme in combination with the MUSCL scheme.

section 3.2. From left to right, the rarefaction wave displays a smooth transition between two states whereas the contact discontinuity and the shock wave show a discontinuous transition respectively. The shock wave is captured with only two interior grid points. The numerical approximation of the constant state of the density in front of the contact discontinuity appears to contain small numerical oscillations. This artifact does not disappear after grid refinement although the distribution of the small oscillations concentrates more in a region just in front of the contact discontinuity. The small oscillations do not vanish if the time step is decreased even further which indicates that the present CFL number is sufficient for temporal accuracy. These oscillations seem to contradict the TVD property of the MUSCL scheme and requires some further attention. The TVD property is only defined for scalar one dimensional equations and the extension to systems of equations or multiple dimensions is not straightforward. In Ref. [35] a definition for a system of equations is presented where it is observed that for a typical time integration scheme and a linear flux function the TVD property holds under the same CFL-like restriction as in the scalar case. However, we will follow a different approach used in [15] and study the discrete entropy production which will be discussed in section 3.6. In addition to the density distribution we also present the pressure and entropy distribution in figures 3-2 and 3-3 respectively where

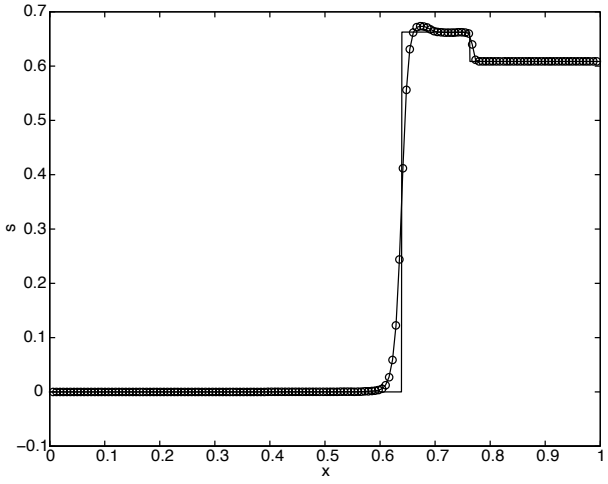


Figure 3-3: The entropy distribution at $t = 0.15$. The solid line represents the analytical solution and the o's the numerical solution obtained with the Crank-Nicolson scheme in combination with the MUSCL scheme.

the entropy is defined as

$$s = \log(\frac{\rho_\infty^\gamma}{p_\infty}\frac{p}{\rho^\gamma}). \tag{3-13}$$

where ρ_∞ and p_∞ are reference values for the density and the pressure. From figure 3-2 it is clear that the pressure is constant over the contact discontinuity and no numerical oscillations are present. Figure 3-3 reveals that the contact discontinuity as well as the shock wave cause an increase of entropy. The overshoot in entropy in front of the contact discontinuity is related to the small oscillations in the density by (3-13).

We conclude that the Crank-Nicolson scheme in combination with the MUSCL scheme results in an accurate approximation of the solution to the shock tube problem for the present numerical parameters, comparable to the results found in [15]. Although for the present magnitude of the time step oscillations occur in the density component of the solution, which seems to contradict the TVD-property, we show in section 3.6 that this solution is still acceptable with respect to the total entropy production.

3.5.2 Comparison of linear solvers

In this section we study the effect of several linear solvers on the required CPU time and the convergence of,

$$H(q^{n+1}) = g(q^n) \tag{3-14}$$

in the case the linear system is not solved to machine accuracy for one time-step. Because we are interested in large time steps for future applications of the implicit time integration scheme we take a CFL number $\sigma = 1.5$ for all simulations in this section. To study the influence of the approximation of the flux Jacobi matrix for the MUSCL scheme we also perform simulations for the first order scheme where the numerical flux Jacobi matrix and the numerical flux are more closely related. Also, the effect of the initial discontinuity at $t = 0$ is studied by performing simulations with a smooth initial condition which is obtained by smoothing the two states in (3-12) with a third order polynomial interpolation. In the following the discontinuous initial condition is denoted by case I and the smooth initial condition is denoted by case II.

The quasi-Newton iteration method used to obtain the solution of (3-14) can be written in the form of (2-44) with $\Delta\tau = \infty$. The iteration is stopped if

$$\frac{||\Delta v^k||_2}{||\Delta v^0||_2} < 10^{-10} \tag{3-15}$$

where the superscript k stands for the iteration level of the quasi-Newton process. The specific form of this stopping criterion will be discussed in chapter 6. The linear system, in the following denoted by $Ax = b$, for each quasi-Newton iteration level is solved or approximately solved with the linear solvers described in section 2.5:

1. Block-tridiagonal solver, which is a direct solver
2. Point Jacobi.
3. Point Gauss-Seidel
4. CG (applied to $A^T Ax = A^T b$).
5. GMRES.

For the first order scheme we additionally use the numerical flux Jacobi matrix defined in section 3.4.2 in combination with the block-tridiagonal solver to obtain the full Newton scheme which should result in quadratic convergence towards a fixed point of (3-14). In the following the number of iterations for solving the nonlinear system is denoted by outer iterations whereas the

	# outer	# inner	CPU (s)
Full Newton	6	–	4.04
Block-tridiagonal	13	–	0.78
Jacobi	13	8	0.94
Gauss-Seidel	13	6	0.90
CG(AA^T)	13	15	13.66
GMRES	13	4	1.02

Table 3-1: Minimal number of outer iterations needed for convergence and corresponding minimal number of inner iterations for the different linear solvers with Roe's scheme for case I.

	# outer	# inner	CPU (s)
Block-tridiagonal	33	–	1.96
Jacobi	33	9	2.42
Gauss-Seidel	33	7	2.37
CG(AA^T)	32	17	33.78
GMRES	32	8	2.91

Table 3-2: Minimal number of outer iterations needed for convergence and corresponding minimal number of inner iterations for the different linear solvers with the MUSCL scheme for case I.

number of iterations used in the approximation of the solution to the linear system is denoted by inner iterations.

In tables 3-1 to 3-4 the results are presented for the different test cases described above. The tables concerning the first order Roe discretisation have one additional row corresponding to the full-Newton scheme for which the expected quadratic convergence was obtained. The number of outer iterations with the full-Newton method is less than for the quasi-Newton iteration. However, the CPU time which is the quantity of most practical interest is approximately a factor four to five times larger than needed for the quasi-Newton iteration which indicates that an approximation of the flux Jacobi matrix may have a positive influence on the required CPU time for convergence.

	# outer	# inner	CPU (s)
Full Newton	4	–	2.66
Block-tridiagonal	7	–	0.38
Jacobi	7	6	0.43
Gauss-Seidel	7	5	0.43
CG(AA^T)	7	10	7.24
GMRES	7	5	0.55

Table 3-3: Minimal number of outer iterations needed for convergence and corresponding minimal number of inner iterations for the different linear solvers with the Roe scheme for case II.

	# outer	# inner	CPU (s)
Block-tridiagonal	29	–	1.71
Jacobi	29	9	2.14
Gauss-Seidel	29	9	2.24
CG(AA^T)	29	14	30.47
GMRES	28	9	2.67

Table 3-4: Minimal number of outer iterations needed for convergence and corresponding minimal number of inner iterations for the different linear solvers with the MUSCL scheme for case II.

Define the measure of accuracy with which the linear system is solved by

$$\frac{||A\widetilde{x} - b||_2}{||b||_2} \tag{3-16}$$

where $\widetilde{x}$ is an approximation of the solution to the linear system $Ax = b$. With the block-tridiagonal solver the linear system is solved up to machine accuracy each inner iteration level. Empirically we found that the number of outer iterations is minimal if the block-tridiagonal solver is used. Although this seems to indicate that it is necessary to solve the linear system very accurately it appears that it is sufficient to solve the linear system up to a certain accuracy depending on the spatial discretisation. It was found that for the first order Roe scheme and for the MUSCL scheme the accuracy with which the linear system has to solved is approximately 10^{-3} and 10^{-4} respectively.

In tables 3-1 to 3-4 the minimal number of inner iterations corresponding

to the minimal number of outer iterations are given. If the number of inner iterations is decreased this leads to an increase of the number of outer iterations in this case which explains the naming minimal number of inner iterations. The number of iterations needed for the Gauss-Seidel method is alway less or equal to the number of iterations needed with the Jacobi method. Since the accuracy with which the linear system has to be solved does not depend on the linear solver this can be explained by the fact that the convergence rate of the Gauss-Seidel method is about a factor two larger than the convergence rate of the Jacobi method [37]. The convergence rate of the CG method is related to the condition number $\kappa(A) = ||A||_2||A^{-1}||_2$ of the matrix A and hence the convergence for $A^TAx = b$ is related to $\kappa(A)^2$ which decreases the convergence rate [37] and thus explains the large number of inner iterations that are needed. The required CPU time for the CG method is high compared to the other methods. Partly this is due to the higher number of iterations but it is also due to the construction of AA^T and A^Tb. Although we recognize that an optimal use of Krylov methods is only possible if a proper preconditioning (see e.g. [91]) is applied to accelerate the convergence of the linear system, we observe that the required accuracy with which the linear system needs to be solved is quite low and only a few linear iterations are necessary. Therefore, no significant decrease of CPU time is expected here if a preconditioning is applied to the linear system. In tables (3-2) and (3-4) it appears that fewer outer iterations are needed if GMRES is used. However, if the accuracy in (3-15) is changed this observation does not hold and appears to be related to round-off effects. Although one fewer outer iteration is required this method is not efficient with respect to CPU time.

Additionally, it can be observed that for all simulations the MUSCL scheme required more outer iterations than the first order scheme which indicates that the accuracy of the approximation of the flux Jacobi matrix has a significant influence on the required number of outer iterations. The presence of a shock also affects the number of required outer iterations because for test case II the number of outer iterations is less than that for test case I for both discretisation schemes. This effect is most pronounced for the first order scheme. For the MUSCL scheme it is less significant which may be related to the approximation of the flux Jacobi matrix described above.

3.6 Entropy production

In this section we study the effect of a comparatively large time step on the quality of the numerical solution. In section 2.4.3 it was shown that in spite of the A-stability of the Crank-Nicolson scheme, the numerical scheme was not TVD for large time steps. The TVD-property in (2-11) is defined for

scalar one-dimensional problems. The extension to systems of equations or multi-dimensional problems is not straightforward. In Ref. [35] a TVD-norm is defined for a system of equations in one dimension which, in the linear case and in combination with a particular choice of time integration method, leads to the same restriction on the time step as in the scalar case. However, for the Crank-Nicolson scheme we were not able to formulate a necessary condition on the time step for the scalar case let alone for the present application. To circumvent this problem we turn to the concept of entropy satisfying solutions (see e.g. [57]). Although no one-to-one correspondence between TVD and entropy is proved for numerical schemes in general, it is made plausible in [15, 62, 66, 78] that a certain similarity does exist.

3.6.1 Entropy satisfying solutions

To introduce the definition of entropy satisfying solutions we turn to the viscous Burgers equation given by

$$\frac{\partial q}{\partial t} + \frac{\partial \frac{1}{2} q^2}{\partial x} = \epsilon \frac{\partial^2 q}{\partial x^2} \tag{3-17}$$

with $\epsilon \geq 0$. As the initial condition we take a piecewise constant initial condition

$$q(x,0) = \begin{cases} q_l & \text{if} \quad x < 0 \\ q_r & \text{if} \quad x > 0 \end{cases} \tag{3-18}$$

For $\epsilon = 0$ (inviscid case) and $q_l > q_r$ there exists a unique weak solution to (3-17) corresponding to a shock wave which propagates with shock speed $s = (q_l + q_r)/2$ and is given by

$$q(x,t) = \begin{cases} q_l & \text{if} \quad x < st \\ q_r & \text{if} \quad x > st \end{cases} \tag{3-19}$$

The characteristic curves of this solution in each region where q is constant run into the shock as time advances. However for $\epsilon = 0$ and $q_l < q_r$ there exist infinitely many weak solutions to (3-17). One of these solutions is equal to the solution (3-19) but now the characteristics run out of the shock. This solution is not stable to perturbations. Another weak solution is the so-called rarefaction wave given by

$$q(x,t) = \begin{cases} q_l & \text{if} \quad x < q_l t \\ x/t & \text{if} \quad q_l t < x < q_r t \\ q_r & \text{if} \quad x > q_r t \end{cases} \tag{3-20}$$

which is stable to perturbations. In fact this solution corresponds to the vanishing viscosity solution of (3-17) defined by

$$q(x,t) = \lim_{\epsilon \downarrow 0} q^{\epsilon}(x,t) \tag{3-21}$$

Since the concept of vanishing viscosity is not easy to work with for more complicated differential equations one would like a more practical condition. In Ref. [57] different criteria are described in order to arrive at the physically correct solution for the inviscid Burgers equation which corresponds to the vanishing viscosity solution of (3-17). Here we use the concept of the so-called entropy function since this also provides a tool to test a numerical scheme on its capability to yield a physically correct solution. Consider a general scalar conservation law given by

$$q_t + f(q)_x = 0 \tag{3-22}$$

where the subscript denotes the corresponding derivative. Assume that there exists a convex function $\eta(q)$ that satisfies another conservation law for a certain entropy flux ψ given by

$$\eta(q)_t + \psi(q)_x = 0 \tag{3-23}$$

Then it is not hard to show that if q is a solution to (3-22) then

$$\eta'(q) f'(q) = \psi'(q) \tag{3-24}$$

which is called the compatibility condition. In Ref. [57] it is shown that the vanishing viscosity solution $q(x,t)$ of the general differential equation

$$q_t + f(q)_x = \epsilon q_{xx} \tag{3-25}$$

also known as the entropy solution of (3-22), satisfies the inequality

$$\eta(q)_t + \psi(q)_x \leq 0 \tag{3-26}$$

in the weak sense. Clearly, this provides a tool that is useful in analyzing numerical methods. If a discrete form of the entropy inequality (3-26) holds for a numerical method, it can be shown that the solution converges to the entropy solution [57].

3.6.2 Application to the Euler equations

We now apply the concept of entropy satisfying solutions to the numerical solution of the one-dimensional Euler equations. In order to arrive at a proper

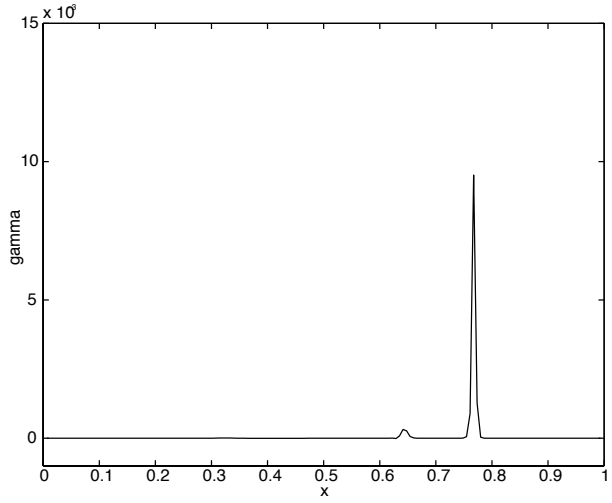


Figure 3-4: The entropy production of the numerical scheme obtained with the Crank-Nicolson scheme in combination with the MUSCL scheme for $\sigma = 0.5$.

entropy inequality, as in (3-26), we invoke the Second Law of Thermodynamics. For smooth solutions q the Euler equations in (3-1) can be transformed to

$$\frac{\partial Q}{\partial t} + \frac{\partial k(Q)}{\partial x} = 0 \tag{3-27}$$

with

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho s \end{bmatrix} \quad \text{and} \quad k = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u s \end{bmatrix} \tag{3-28}$$

where s denotes the specific entropy. For a weak solution q of the Euler equations which contains a discontinuity the equal sign in (3-27) does not hold for the entropy equation which according to the Second Law of Thermodynamics must satisfy

$$\frac{\partial \rho s}{\partial t} + \frac{\partial \rho u s}{\partial x} \geq 0 \tag{3-29}$$

If we discretize both (3-1) and (3-27) with a finite volume method this yields

$$\Omega_j \frac{dq_j}{dt} + h_{j+\frac{1}{2}} - h_{j-\frac{1}{2}} = 0 \tag{3-30}$$

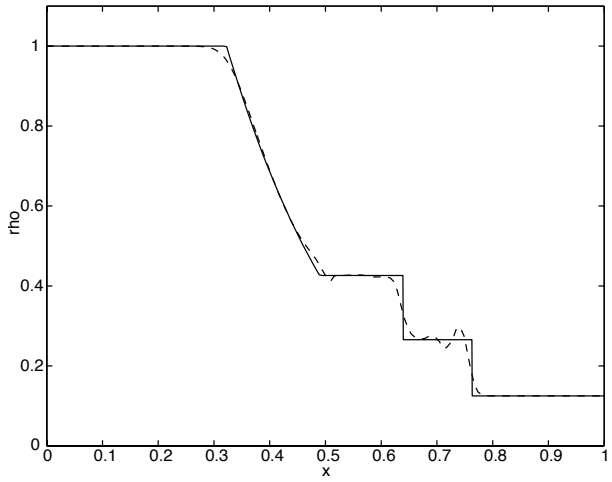


Figure 3-5: The density distribution at $t = 0.15$. The solid line represents the analytical solution and the dashed line represents the numerical solution obtained with the Crank-Nicolson scheme in combination with the MUSCL scheme for $\sigma = 2.5$.

and

$$\Omega_j \frac{dQ_j}{dt} + k_{j+\frac{1}{2}} - k_{j-\frac{1}{2}} \geq 0 \tag{3-31}$$

respectively, where $k_{j+\frac{1}{2}}$ is the numerical approximation of the flux k at the cell interface and the inequality applies only to the third component of Q. Substitution of (3-30) in (3-31) gives

$$-Q_q \left(h_{j+\frac{1}{2}} - h_{j-\frac{1}{2}} \right) + k_{j+\frac{1}{2}} - k_{j-\frac{1}{2}} \geq 0 \tag{3-32}$$

where Q_q denotes the Jacobi matrix of the transformation from the vector space q to the vector space Q. This inequality is the discrete equivalence of the entropy inequality (3-26) for the Euler equations. The discretisation of the entropy flux $k(Q)$ is usually performed with the same discretisation as for the flux $f(q)$. In this way the entropy flux is determined in a so-called Lax-c-consistent manner as in [15, 78] which means that the compatibility requirement in (3-24) applies to the discrete flux functions $h_{j+\frac{1}{2}}$ and $k_{j+\frac{1}{2}}$.

In the following we illustrate that for a comparatively large time step the numerical solution does not satisfy the discrete entropy inequality (3-26) at $t = 0.15$. For all simulations the nonlinear system at every time step is solved with an accuracy of 10^{-10}, as defined in (3-15). First we show that the numerical solution obtained in section 3.5.1 with $\sigma = 0.5$ results in a solution

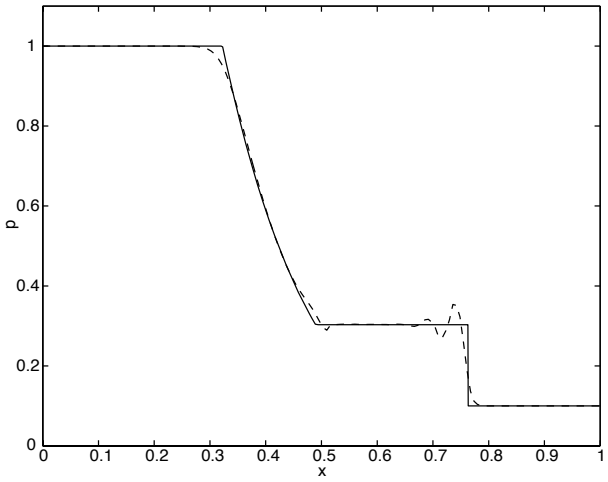


Figure 3-6: The pressure distribution at $t = 0.15$. The solid line represents the analytical solution and the dashed line represents the numerical solution obtained with the Crank-Nicolson scheme in combination with the MUSCL scheme for $\sigma = 2.5$.

that satisfies the discrete entropy inequality (3-32). In figure 3-4 the discrete entropy production is presented. The large peak of the entropy production occurs at the shock, whereas the small peak corresponds to an entropy production at the contact discontinuity which agrees well with the results found in [15]. The entropy production in the entire domain is non-negative which implies that a physically correct solution is obtained. We may conclude that the small oscillations observed in the density in figure (3-1) do not imply a physically incorrect solution in the entropy sense. Second, the solution at $t = 0.15$ was determined with $\sigma = 2.5$. In figures 3-5 to 3-6 the density distribution and the pressure distribution are presented. The amplitude of the oscillations in the density before the shock has increased significantly compared to the oscillations obtained with $\sigma = 0.5$. Also, oscillations in the pressure distribution are present near the shock which were not observed in the solution obtained with $\sigma = 0.5$. In figure 3-7 the entropy production is plotted. Clearly, there is a negative entropy production between the contact discontinuity and the shock wave which indicates that a non-physical solution is obtained. The magnitude of the entropy production peak has decreased considerably compared to the entropy production peak for the solution obtained with $\sigma = 0.5$. This may be explained by the number of grid points contained in the shock. The shock contains four points for the solution obtained with $\sigma = 2.5$ compared to two points for the solution obtained with $\sigma = 0.5$ and a lower entropy production

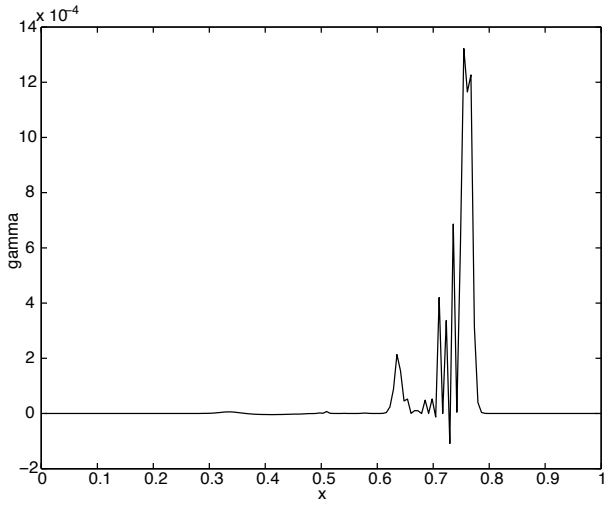


Figure 3-7: The entropy production of the numerical scheme obtained with the Crank-Nicolson scheme in combination with the MUSCL scheme for $\sigma = 2.5$.

may be expected. This result illustrates that similar to the numerical results for the one-dimensional Burgers equations in section 2.4.3 a relatively large time step using the Crank-Nicolson scheme may result in a physically incorrect solution.

3.7 Conclusions

In the previous sections we numerically solved the one-dimensional Euler equations for Sod's shock tube problem with the first order Roe scheme and the third order MUSCL scheme for the spatial discretisation combined with the Crank-Nicolson scheme for the temporal integration.

The influence of the numerical approximation of the flux Jacobi matrix was studied for the first order scheme by comparing the required number of outer iterations for the quasi-Newton and full-Newton method corresponding to the approximation of the flux Jacobi matrix in section 3.4.1 and section 3.4.2 respectively. For the full-Newton method the number of required outer iterations for one time step is minimal. Although the quasi-Newton method requires more outer iterations during one time step this method is computationally more efficient and thus constitutes a more promising method for practical applications. It is illustrated that for the MUSCL scheme the number of required outer iterations is considerably larger than for the first order

scheme. An extension of the stencil used in the numerical flux Jacobi matrix is a possible way to improve the quality of the numerical approximation and fewer outer iterations may be expected. However, given the above mentioned results for the first order scheme with respect to CPU time we expect that the additional computational costs of determining, storing and inverting the flux Jacobi matrix do not outweigh the additional cost of more outer iterations.

It turns out that the required accuracy with which the linear system has to be solved is only about three or four decades depending on the spatial discretisation corresponding to only a few inner iterations. All iterative methods that have been used achieve the required accuracy with very few iterations. In general the Gauss-Seidel method leads to the lowest overall CPU time in comparison to the other iterative methods. Therefore, in the remainder of this thesis we use the Gauss-Seidel method for the inner iteration.

Finally, we observe that similar to the results for the one-dimensional Burgers equation in section 2.4.3 a large time step results in a numerical solution which is physically not correct. Although no proof of the relation between TVD and entropy satisfying solutions for the MUSCL scheme is given we have provided further indication that these two concepts are strongly related as was also observed by Merriam [62].

Chapter 4

Inviscid flow around an airfoil

4.1 Introduction

In this chapter we numerically solve a steady transonic inviscid flow problem around an airfoil. Using the second-order explicit Runge-Kutta method in combination with the MUSCL-scheme and the limiter defined in section 2.3 it is not possible to obtain a machine accurate solution. In contrast to this result a machine accurate solution can be obtained with an appropriate implicit time integration scheme. Two points will be addressed in this chapter. First we consider the computational efficiency of the implicit method compared to the explicit scheme. Second we study the difference between the solution obtained with the implicit and explicit method and we give some evidence that the convergence stall observed when using the explicit scheme is in fact caused by a small physical instability in the wake.

The introduction of higher order TVD schemes for transonic flow applications enables the combination of two desirable features of a numerical scheme: a minimal amount of artificial dissipation and monotonicity preservation which eliminates numerical oscillations near shocks. Examples of such TVD schemes which are commonly used are MUSCL schemes [56] which incorporate a nonlinear function which limits gradient differences of the solution between adjacent cells. As is described in section 2.3.1 the graph of this limiter should satisfy specific constraints in order to be of higher order and monotonicity preserving as was shown in figure 2-3. This still leaves a certain freedom in the choice of the limiter. Commonly used limiters in literature are the superbee of Roe, Van Albada's limiter, the minmod limiter and Van Leer's limiter [83]. These limiters can be divided into two groups, differentiable and non-differentiable limiters.

In spite of the favorable properties of TVD schemes, it is often remarked in literature that the limiter may inhibit convergence to steady state. In Refs. [89, 90] it is remarked that a non-differentiable limiter inhibits convergence to

steady state independent of the time integration method. To obtain a steady state solution in Refs. [89, 90] a differentiable limiter was used. Conversely, employing a multi-grid technique and an optimized explicit time integration scheme Park and Kwon [67] obtain a steady state solution for a large class of limiters, including non-differentiable limiters. In this chapter we show that with a proper implicit scheme it is also possible to obtain a machine accurate steady state solution for the Euler equations with the non-differentiable minmod limiter.

Eriksson and Rizzi [28] have developed an approximate eigenvalue analysis for a semi-discrete centered finite volume scheme and adopted this to a transonic inviscid flow around an airfoil. They observed that there are modes which are not damped well and correspondingly dominate the asymptotic convergence to steady state. These modes appeared to be centered around the shock, the wake and the stagnation point of the airfoil and have eigenvalues with a very small negative real part. They conclude that the addition of artificial dissipation or the application of local time stepping is beneficial for convergence because it decreases the real part of the eigenvalues of these modes, thus damping them better.

We show, conversely, that methods which do obtain a machine accurate steady state solution in fact may add too much artificial dissipation. By studying the differences in the solution obtained with an explicit Runge-Kutta time stepping scheme and an implicit method there appears to be a weak instability in a region behind the trailing edge. On a coarse grid with a resolution comparable to the finest grid in [28] additional instabilities show up near the shock and stagnation point but these regions of instabilities disappear when refining the grid. A grid refinement study shows that the main region of instability in the wake does not vanish on a very fine grid. These instabilities are also confirmed by linear stability theory (LST).

The contents of this chapter is as follows. In section 4.2 we state the governing equations and the numerical method for the explicit Runge-Kutta scheme. In section 4.3 numerical results for the explicit scheme are presented and the convergence level of the residual will be compared with results obtained with other numerical methods in literature. In section 4.4 an implicit scheme is introduced with which a machine accurate solution with the non-differentiable minmod limiter can be obtained. Section 4.5 contains a discussion of the numerical results for the implicit scheme and a comparison study of the final residual levels obtained by the implicit and explicit methods. Finally, in section 4.6 the conclusions are summarized.

4.2 Governing equations and explicit numerical method

In this section we state the equations governing inviscid compressible flow and specify the time explicit numerical method. Although we are interested in the steady solution for the flow around an airfoil we do not solve the time-independent equations directly. Instead, we start from an initial condition and use a time stepping scheme to obtain the steady solution.

4.2.1 Governing equations

The equations governing inviscid compressible flow are the Euler equations. In Cartesian coordinates in two dimensions they read:

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0 \tag{4-1}$$

with

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad f = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{bmatrix} \text{ and } \quad g = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{bmatrix}, \tag{4-2}$$

where ρ is the density, E the total energy density and u and v are the velocity components in x and y direction respectively. The constitutive equation for the pressure, p, is given by

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho(u^2 + v^2)\right), \tag{4-3}$$

where γ is the adiabatic gas constant, which we take $\gamma = 1.4$.

4.2.2 Spatial discretisation

In section 2.3.1 the flux splitting scheme of Roe was described for a scalar equation and extended to a system of equations in one dimension in section 3.3 by taking the hyperbolicity of the Euler equations into account. Here we further extend the numerical method to the Euler equations in two dimensions. To solve (4-1) we use a finite volume method on a structured grid that computes the flux over the control volume edges as in figure 4-1

$$\Omega_{i,j}\frac{dq_{i,j}}{dt} + h_{i+\frac{1}{2},j} - h_{i-\frac{1}{2},j} + h_{i,j+\frac{1}{2}} - h_{i,j-\frac{1}{2}} = 0, \tag{4-4}$$

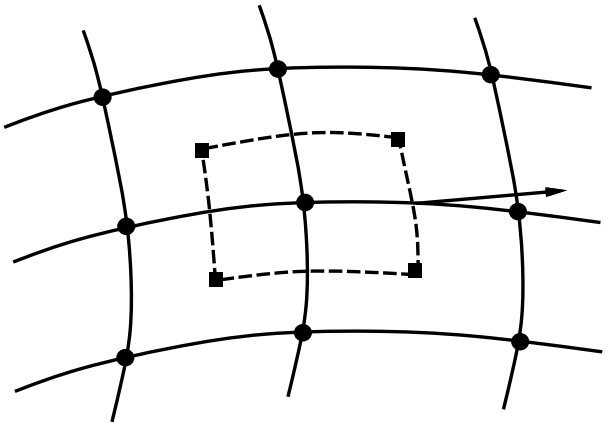


Figure 4-1: Control volume for grid point (i, j). The arrows denote the normals on the edges.

where $\Omega_{i,j}$ is the volume of the control volume and h is the numerical flux vector on the four boundary segments $(i + \frac{1}{2}, j)$ etc. The problem with the two-dimensional Euler equation is that the flux Jacobi matrices corresponding to the fluxes f and g cannot be diagonalized simultaneously. However, it can be shown that every linear combination of the two Jacobi matrices is diagonalizable [43]. In our simulations we consider the flux Jacobi matrix in the normal direction which is defined as

$$A(q, \mathbf{n}) = \nabla_q(f(q))n_x + \nabla_q(g(q))n_y \tag{4-5}$$

with $\mathbf{n} = [n_x, n_y]$. In order to have a Roe type linearisation the same requirements as stated in section 3.3 have to be satisfied by $A(q, \mathbf{n})$. The numerical flux on the control volume edges is approximated by

$$h_{i+\frac{1}{2},j} = \frac{1}{2} l_{i+\frac{1}{2},j} \begin{bmatrix} f(q_l) + f(q_r) \\ g(q_l) + g(q_r) \end{bmatrix} \cdot \mathbf{n}_{i+\frac{1}{2},j} - \frac{1}{2} l_{i+\frac{1}{2},j} \, |A_{i+\frac{1}{2},j}| \, (q_r - q_l), \tag{4-6}$$

where $l_{i+\frac{1}{2},j}$ is the length of the control volume edge, $\mathbf{n}_{i+\frac{1}{2},j}$ is the outward normal on the edge, q_l and q_r are appropriate left and right state vectors, the dot denotes the inner product and $A_{i+\frac{1}{2},j}$ is the combined flux Jacobi matrix defined in (4-5). Similar to the one-dimensional case the flux in (4-6) is evaluated with the so-called Roe average state-vector as defined in (3-5) supplemented with the definition of v_{lr}. To obtain a higher order accuracy the MUSCL technique (see section 2.3.1) is applied to the conservative state variables separately to obtain an improved approximation of q_l and q_r.

4.2.3 Explicit time integration

For the time integration we use the second order accurate explicit four stage compact storage Runge-Kutta scheme defined in section 2.4.1. The time step of the explicit scheme is bounded for stability reasons and for the control volume $\Omega_{i,j}$ it is chosen according to

$$\Delta t_{i,j} = \frac{\sigma \Omega_{i,j}}{\max\left(l_{i+\frac{1}{2},j}\lambda_{i+\frac{1}{2},j}, l_{i-\frac{1}{2},j}\lambda_{i-\frac{1}{2},j}\right) + \max\left(l_{i,j+\frac{1}{2}}\lambda_{i,j+\frac{1}{2}}, l_{i,j-\frac{1}{2}}\lambda_{i,j-\frac{1}{2}}\right)} \tag{4-7}$$

where σ is the CFL number and the λ's are the maximum absolute values of the eigenvalues of the flux Jacobi matrix on the corresponding control volume edges.

To increase the rate of convergence of the explicit scheme we apply local time stepping, i.e. each point is advanced according to its own stability time step. In some calculations presented in this chapter we apply global time stepping where a uniform time step is chosen equal to the minimum of all local time steps.

4.2.4 Boundary conditions

For the flow around an airfoil there are two types of boundaries. The far field boundary due to the finite extent of the computational domain and the solid wall. In the far field we permit subsonic inflow or outflow. We use a method which takes the incoming and outgoing characteristics into account. Depending on whether the boundary is an inflow or outflow boundary we extrapolate one or three Riemann invariants from the inner field and set the remaining Riemann invariants to their values at infinity [74].

The only physical condition for inviscid flow over a solid wall is the impermeability of the solid wall which is equivalent to the normal velocity at the solid wall being equal to zero. As numerical boundary conditions we extrapolate the density, the tangential velocity and the pressure. Due to the use of a C-grid the trailing edge becomes multi-valued. We therefore average the values of the state vectors at the trailing edge after every stage of the Runge-Kutta scheme.

To initialize the flow field, we set all dependent variables equal to their values at infinity determined by the Mach number, M_∞, and the angle of attack, α.

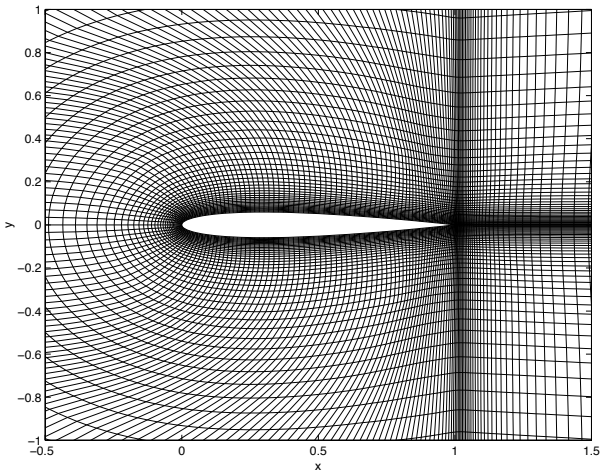


Figure 4-2: Enlargement of the C-grid around the NACA0012 airfoil.

4.3 Numerical results

In this section we present the numerical results obtained with the explicit scheme for the well-known test case of inviscid flow around a NACA0012 airfoil at $M_\infty = 0.8$ and $\alpha = 1.25^o$ [103]. With this combination of free-stream Mach number and angle of attack the flow is transonic. The solution has a strong shock on the upper surface of the airfoil, a weak shock on the lower surface and a weak contact discontinuity in the wake. The grid we use is a C-grid with 289×65 grid points where 160 points are located on the airfoil and 65 points in the wake, see figure 4-2. The far field boundary is located at approximately 20 chord lengths from the airfoil. The CFL number for the time step used here is CFL=0.6. In figure 4-3 a contourplot of the pressure shows a strong shock on the upper side of the airfoil and a weak shock on the lower side of the airfoil which is in agreement with the results in [15, 90]. The strong shock on the upper side is captured with only one grid point in the shock which can be seen in figure 4-4 where the pressure coefficient c_p, defined by

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2}, \tag{4-8}$$

is plotted. The lift and drag, which are defined by

$$c_l = \frac{1}{\frac{1}{2}\rho_\infty U_\infty^2 L} \oint -p[-n_x \sin\alpha + n_y \cos\alpha] ds \tag{4-9}$$

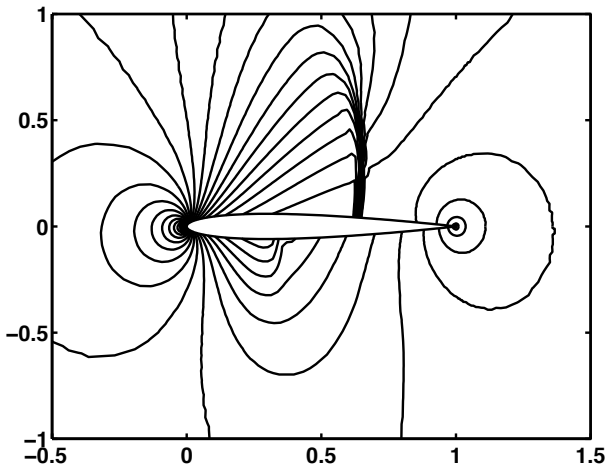


Figure 4-3: Contourplot of the pressure for $M_\infty = 0.8$ and $\alpha = 1.25^o$ using the MUSCL scheme.

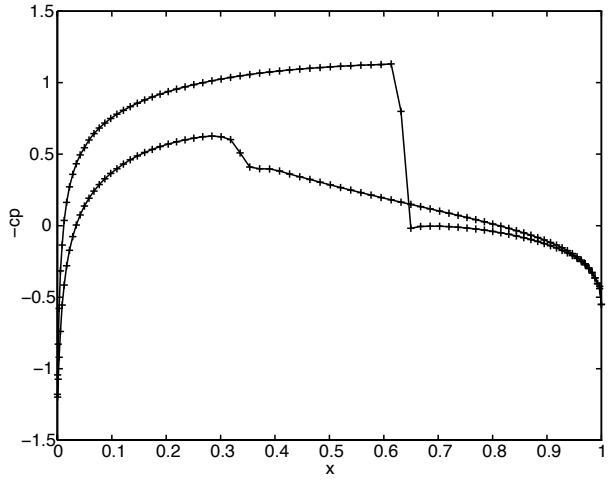


Figure 4-4: The pressure coefficient c_p along the surface of the airfoil for $M_\infty = 0.8$ and $\alpha = 1.25^o$ using the MUSCL scheme.

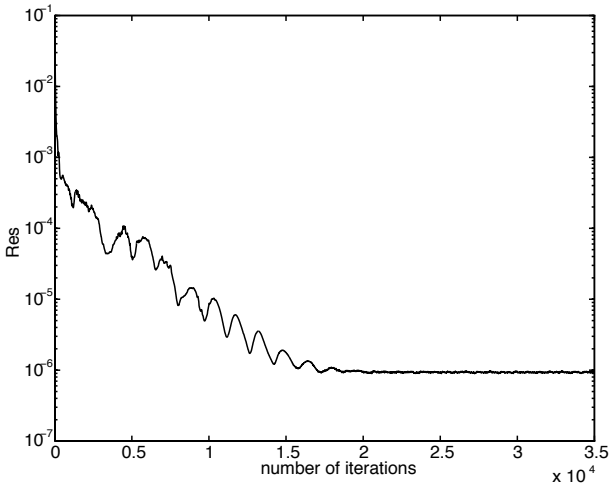


Figure 4-5: Convergence history of the residual of the density with the explicit method at $M_\infty = 0.8$ and $\alpha = 1.25^o$ with CFL=0.6.

and

$$c_d = \frac{1}{\frac{1}{2}\rho_\infty U_\infty^2 L} \oint -p[n_x \cos\alpha + n_y \sin\alpha] ds \tag{4-10}$$

respectively where $U_\infty = \sqrt{u_\infty^2 + v_\infty^2}$, L denotes the chord length of the airfoil and α represents the angle of attack. We find for the values of the lift and drag: $c_l = 0.348657$ and $c_d = 0.021980$. As can be seen in figure 4-5 the solution does not converge to machine accuracy. This result is in agreement with literature (see [89]). If we change the spatial discretisation from the third order MUSCL scheme to the first order scheme of Roe or to the second order scheme of Jameson [47] we do get a machine accurate solution, consistent with the findings in [28]. Therefore, it appears that the lower amount of dissipation in the MUSCL scheme (2-24) compared with Roe's and Jameson's scheme [15], prevents the solution to converge to machine accuracy.

Another explanation for the fact that the convergence of the MUSCL scheme stalls is the non-differentiability of the limiter function (2-26) [89, 90]. According to Ref. [90] this convergence stall is independent of the time stepping method. In the next section we, conversely, show that with a proper implicit scheme we can achieve a machine accurate solution with the minmod limiter.

4.4 Implicit method

In the previous section we established that it is not possible to obtain a machine accurate solution with the explicit Runge-Kutta scheme in combination with the MUSCL scheme and the minmod limiter. In this section we formulate an implicit factorization method which, as will be shown in section 4.5, enables a decrease of the residual to machine accuracy.

The discrete version of (4-1) for the Euler backward scheme can be written as

$$q_{i,j}^{n+1} = q_{i,j}^{n} - \Delta t F_{i,j}(q^{n+1}), \tag{4-11}$$

where the superscript n labels the time level and $F_{i,j}$ is the total numerical flux in the grid point (i,j). The Euler backward scheme is first order accurate in time, but this is not a concern here since we are only interested in the steady state solution. First order Taylor expansion of F around q^n yields

$$\left(\frac{I}{\Delta t} + \frac{\partial F}{\partial q}(q^n)\right) \Delta q_{i,j} = -F_{i,j}(q^n) \tag{4-12}$$

where $\frac{\partial F}{\partial q}$ is the symbolic representation of the Jacobi matrix of F and $\Delta q_{i,j} = q_{i,j}^{n+1} - q_{i,j}^{n}$. For infinite Δt and an exact flux Jacobi matrix this scheme is equal to Newton iteration for the problem $F(q) = 0$. However, it is not possible to obtain the exact flux Jacobi matrix at a reasonable cost. Therefore, we approximate the flux Jacobi matrix as in [98]. In view of the five point stencil of Roe's scheme in 2D we get a flux Jacobi matrix with five bands of 4×4-matrices. The five blocks for a grid point (i,j) are given by

$$\begin{aligned}
D_{i,j} &= A^{+}_{i-\frac{1}{2},j} - A^{-}_{i+\frac{1}{2},j} + A^{+}_{i,j-\frac{1}{2}} - A^{-}_{i,j+\frac{1}{2}} \\
N_{i,j} &= A^{-}_{i,j+\frac{1}{2}} \\
S_{i,j} &= -A^{+}_{i,j-\frac{1}{2}} \\
E_{i,j} &= A^{-}_{i+\frac{1}{2},j} \\
W_{i,j} &= -A^{+}_{i-\frac{1}{2},j},
\end{aligned} \tag{4-13}$$

where D, N, S, E and W stand for diagonal, north, south, east and west contribution. The delta formulation in (4-12) allows the use of an approximation of the flux Jacobi matrix without changing the steady state solution. If the iteration process converges it follows from (4-12) that the flux equals zero in all grid points and hence the solution satisfies the time-independent discrete

equations. The matrix on the left hand side in (4-12) can be rewritten as

$$\begin{aligned}\frac{I}{\Delta t}+\frac{\partial F}{\partial q} &= \left(\frac{I}{\Delta t}+D+N+S+E+W\right)\\ &= \left(\frac{I}{\Delta t}+D+N+E\right)\left(\frac{I}{\Delta t}+D\right)^{-1}\left(\frac{I}{\Delta t}+D+S+W\right)\\ &\quad -(N+E)\left(\frac{I}{\Delta t}+D\right)^{-1}(S+W).\end{aligned} \tag{4-14}$$

in which D, N, S, E and W are the contribution to $\frac{\partial F}{\partial q}$ from the corresponding parts in (4-13). Neglecting the last term, which is $\mathcal{O}((\Delta t)^2)$ compared to the first term, we obtain the following implicit factorization scheme,

$$\left(\frac{I}{\Delta t_{i,j}}+D+N+E\right)\left(\frac{I}{\Delta t_{i,j}}+D\right)^{-1}\left(\frac{I}{\Delta t_{i,j}}+D+S+W\right)\Delta q_{i,j} = -F(q^n_{i,j}) \tag{4-15}$$

which is similar to the scheme used in Ref. [102], however, with a different approximation of the flux Jacobi matrix: This system consists of a lower, upper and diagonal matrix and can be solved in two steps.

$$\begin{aligned}\left(\tfrac{I}{\Delta t_{i,j}}+D+N+E\right)\Delta q^*_{i,j} &= -F(q^n_{i,j})\\ \left(\tfrac{I}{\Delta t_{i,j}}+D+S+W\right)\Delta q_{i,j} &= \left(\tfrac{I}{\Delta t_{i,j}}+D\right)\Delta q^*_{i,j}\\ &= -F(q^n_{i,j})-N\Delta q^*_{i,j+1}-E\Delta q^*_{i+1,j}\end{aligned} \tag{4-16}$$

The last line in (4-16) shows that this factorization method can be regarded as two sweeps of the point Gauss-Seidel method where the direction in which the variables are solved changes in the second sweep but no intermediate update of the flux or the numerical flux Jacobi matrix is computed. Because of the topology of the C-grid, the grid is cut into two parts separated by the wake centerline and the stagnation line to retain symmetry for zero angle of attack and a symmetric grid.

The boundary condition at the solid wall is treated explicitly in the same way as for the explicit scheme. The far field boundary condition is treated implicitly.

4.5 Numerical results for the implicit scheme

In this section we show results for the same test case as considered in section 4.3 but now obtained with the implicit method defined in the previous section.

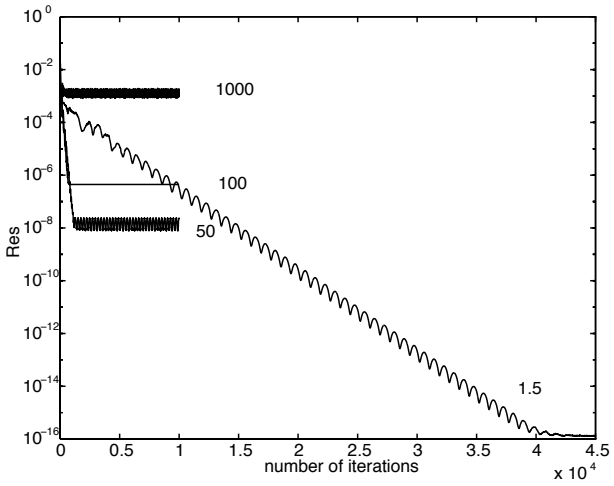


Figure 4-6: Convergence history of the residual of the density with the factorization method and several CFL number for $M_\infty = 0.8$ and $\alpha = 1.25^o$.

A study of the difference of the residual between the explicit method and the implicit method is also presented.

The time step for the implicit scheme is determined in the same way as the stability time step for the explicit scheme. Local time stepping is used to accelerate convergence. Since the Euler Backward scheme is A-stable we increase the CFL number as much as possible. In theory it should be possible to choose the CFL number infinite. We find that it is possible to increase the CFL number considerably compared with the CFL number for the explicit scheme. A threshold value of the CFL number of about 1000 is encountered, above which no convergence takes place. The occurrence of such a threshold in the CFL number can be due to the truncation error in (4-15) which has a more pronounced effect if the time step increases. Another reason may be that we do not treat the solid wall implicitly or the fact that the stability analysis is based on a linearized equation. Also, the sensitivity of the dynamical system on the time step may play an important role as is discussed in more detail in section 6.6.

As can be seen in figure 4-6 the final residual level of the factorization method depends on the CFL number. For large CFL numbers the convergence is fast but a stall occurs. If we decrease the CFL number the convergence rate and the final residual level decrease. Although a different method was used this effect has also been observed in ref. [46]. For the choice CFL=1.5 we obtain a machine accurate solution with the minmod limiter. Fast convergence to

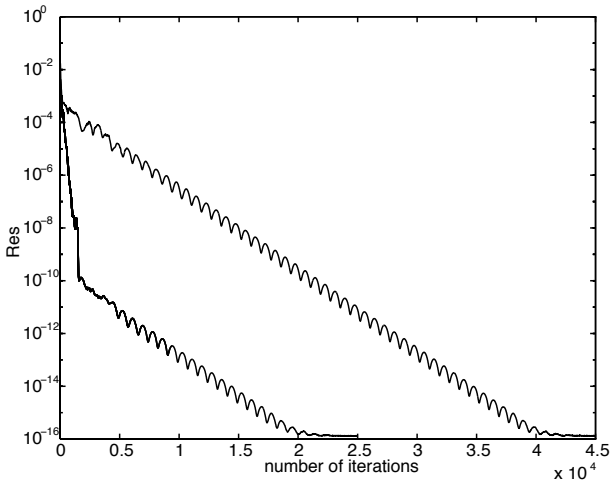


Figure 4-7: Convergence history of the residual of the density with the factorization method for $\alpha = 1.25^o$ and $M_\infty = 0.8$. The upper line is calculated with CFL=1.5 and the lower with a combination of CFL=50 and CFL=1.5

machine accuracy can be obtained if we start with CFL=50 and switch to CFL=1.5 when the solution reaches the minimum residual level for CFL=50. In figure 4-7 it is shown that after the switch to the lower CFL number a sharp drop of the residual occurs after which the evolution of the residual becomes similar to a run with fixed CFL=1.5. The number of iterations needed for machine accuracy is decreased by a factor 2 compared to the run with CFL=1.5.

In (4-15) we use the NE-SW combination of the factorization. In a method where the NE-SW factorization and the NW-SE version are applied alternatingly the convergence rate increases for large CFL number but the final residual level remains the same. The threshold value for the CFL number does not change either. For the choice of CFL=1.5 no convergence acceleration occurs over the pure NE-SW method. Another version of the factorization method is the NS-EW combination which is equivalent to the DD-ADI method described in [7] . However, we do not obtain a machine accurate solution with this method either.

When comparing the solution obtained with the explicit method and the factorization method (4-15) with CFL=1.5 there is no difference in the lift and drag up to at least six digits. The question that arises is what causes the difference in the final residual level between the two methods? Is it due to a physical instability or is it a numerical phenomenon? To study the difference between

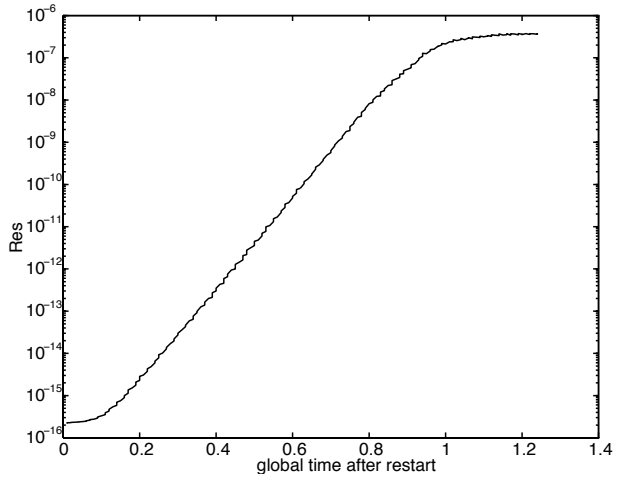


Figure 4-8: Residual as a function of time after a restart with the explicit scheme with CFL=0.6, $\alpha = 1.25^o$, $M_\infty = 0.8$ and global time stepping.

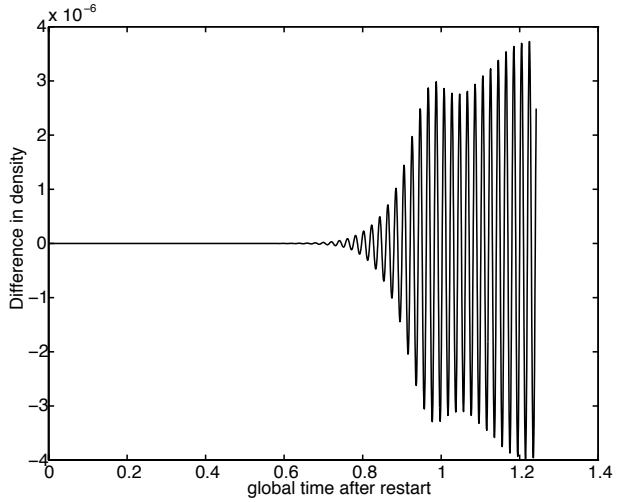


Figure 4-9: Density in a grid point in the wake after restart with the explicit scheme with CFL=0.6, $\alpha = 1.25^o$, $M_\infty = 0.8$ and global time stepping.

the solutions we proceed from the machine accurate solution obtained with the factorization method. We use this solution as initial state and continue with the explicit method using global time stepping and CFL=0.6. Figure 4-8 shows that the residual after the restart has an exponential growth until it reaches a level which is slightly lower than the final value of the residual in figure 4-5 which is obtained with local time stepping.

In figure 4-9 the evolution of the difference in density in a specific grid point in the wake region is plotted. Initially the solution appears almost constant in time but as the dominant instability has sufficiently grown the solution starts to oscillate with an exponential increase of amplitude until the amplitude saturates. The oscillation appears to contain several different frequency modes. The effects shown in figure 4-8 and figure 4-9 are very similar to shear layer instabilities as studied in Ref. [12] and results found from computations of a flow along a flat plate and in mixing layers as described in Ref. [93]. This may indicate that we are dealing with a physical phenomenon. The region where the amplitude of the oscillations is significantly different from zero is bounded and located just behind the trailing edge where a contact discontinuity exists due to a difference in tangential velocity above and below the airfoil.

To exclude the effect of this discontinuity we have performed similar calculations for $\alpha = 0^o$ and $M_\infty = 0.8$. The final value of the residual for the explicit method is three decades lower than for $\alpha = 1.25^o$ but again a stall in the convergence arises. With the factorization method, however, a machine accurate solution is obtained. A restart from the machine accurate solution with the explicit method and global time stepping shows no increase of the residual. Adding a random disturbance to the machine accurate solution of the order 10^{-12} and restarting with the explicit scheme using global time stepping, however, we find an increase in the residual approaching the same level as with uniform flow as initial condition.

In section 2.3.1 several parameters in the limiter function were introduced. Changing the parameters results in a different spatial discretisation. Therefore the dependence of the instability on the discretisation can be studied for a class of discretisation schemes by varying the parameter ω. We study the effects on the different limiter choices next. We have calculated the steady state solution at $\alpha = 1.25^o$ with the implicit scheme for various values of ω between 1 and 2.5 using a CFL number of 1.5. For $\omega = 1$ we obtain a residual of the order 10^{-6}, for $\omega = 2.5$ we obtain a residual of the order 10^{-9} and for $\omega = 1.25, \omega = 1.5$ and $\omega = 2.0$ a machine accurate solution is obtained. The case $\omega = 1$ corresponds to the original minmod limiter where the discontinuity of the derivative of the limiter is situated in the special point $(1, 1)$. The maximum value of the limiter (2-29) for $\omega = 2.5$ is equal to two, which is equal to the maximum

value of the superbee limiter of Roe. It is known that the superbee adds a minimum amount of dissipation compared to the other symmetrical limiters making it less robust, which may explain why we do not obtain a machine accurate solution for this choice of ω . For all other choices of ω the non-differentiability is removed from a neighborhood of $R = 1$ (see section 2.3.1) which seems to be a sufficient condition to obtain a machine accurate solution with the implicit scheme.

Additional calculations for $\alpha = 1.25^o$ and the Van Leer limiter are performed to see whether the instability depends on the type of limiter. The Van Leer limiter is defined as

$$\text{Lim}(a,b) = \begin{cases} \dfrac{|ab| + ab}{a+b} & \text{if} \quad a+b \neq 0 \\ 0 & \text{if} \quad a+b = 0. \end{cases} \tag{4-17}$$

To obtain the original Van Leer limiter the parameter ω is set to one in (2-24). For a CFL number of 1.5 we obtain a machine accurate solution with the implicit scheme. If we use this solution as initial data for the explicit scheme the residual increases with the same growth rate as when the minmod limiter is used with $\omega = \frac{3}{2}$ (see figure 4-8).

Finally, the implicit time integration is changed to the second order Crank-Nicolson scheme, defined by

$$q_{i,j}^{n+1} - q_{i,j}^{n} = \frac{\Delta t}{2}\left(F_{i,j}(q^{n+1}) + F_{i,j}(q^{n})\right) \tag{4-18}$$

for which the leading term in the truncation error is dispersive as is the case for the Runge-Kutta scheme. The factorization scheme is used here as well to solve the linear systems as in (4-14). The convergence behavior is similar to the convergence behavior of the Euler Backward scheme with CFL=1.5 (figure 4-6), but the convergence stalls at the same level as the explicit scheme (figure 4-5).

Summarizing the results, we can conclude that we obtain a machine accurate solution if too much numerical dissipation is added through the spatial discretisation or time integration. For the schemes with minimal numerical dissipation the convergence stalls at the final residual level of the explicit scheme.

4.5.1 Grid refinement

Although there is a slight difference between $\alpha = 1.25^o$ and $\alpha = 0^o$, both cases show that the steady solution found with the implicit method is unstable. This phenomenon may be caused by irregularities or coarseness of the grid. Therefore we repeat the calculation for $\alpha = 1.25^o$ on a coarser and a finer

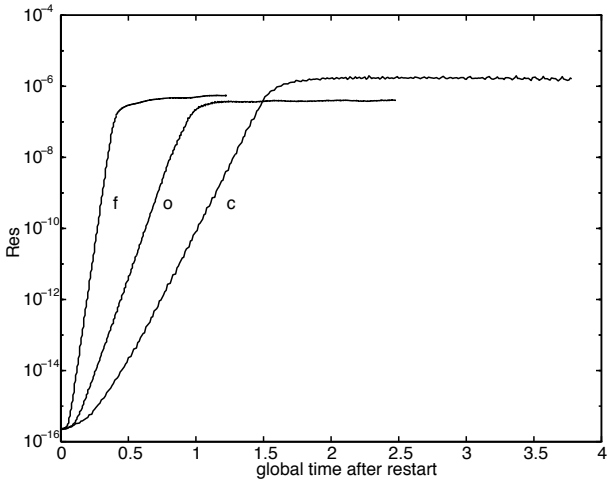


Figure 4-10: Residual as a function of time after a restart with the explicit scheme with CFL=0.6, $\alpha = 1.25^o$, $M_\infty = 0.8$ and global time stepping for the three different grids where o is the original grid, f the fine grid and c the coarse grid.

grid. The coarse grid, 145×33, is obtained by deleting every other grid line in i and j direction. The fine grid, 577×129, is obtained with a fourth order two dimensional interpolation of the grid. For both grids we obtain a machine accurate solution with the factorization method.

Restarting with the explicit method using global time stepping we find a similar increase of the residual as on the original grid. The maximum value of the amplitude and its location are on the fine grid close to the results on the original grid. The instability phenomenon is also seen on the coarse grid. However, the region where instabilities are prominent is almost the entire region behind the airfoil which indicates that the grid is far too coarse. In figure 4-10 the residual for the three grids after a restart is plotted. It shows that the increase of the residual is maximal for the finest grid. This might be due to the fact that the dissipation on a finer grid is lower and therefore the strength of the physical instability is represented better.

4.5.2 Linear Stability Theory

In order to further substantiate these observations the stability of the solution obtained with the factorization method is studied in the framework of linear stability theory (LST) [61]. To this purpose the solution along a vertical line through the wake is inferred from the available solution on the non-orthogonal

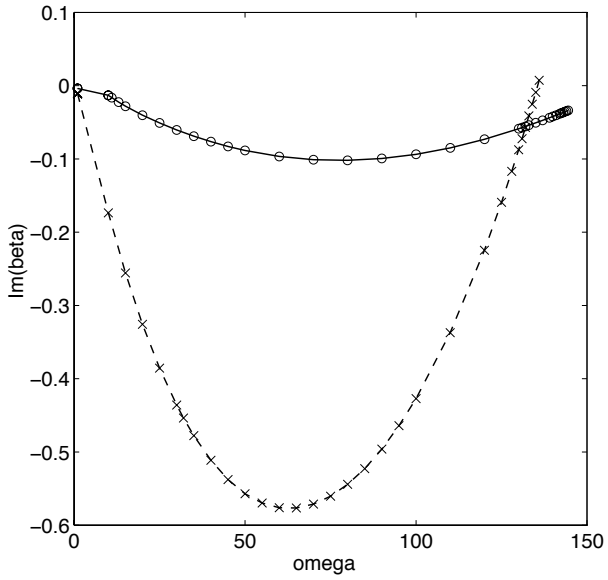


Figure 4-11: β_i as a function of $\widetilde{\omega}$ for a typical point in the wake for the cases $\alpha = 0.0^o$ (solid) and $\alpha = 1.25^o$ (dashed)

grid using fifth order accurate interpolation. Consistent with LST the parallel flow assumption is invoked and a perturbation q' of the form

$$q' = \text{Real}[\hat{\phi}(y)\exp[i(\beta x - \widetilde{\omega}t)]], \tag{4-19}$$

is determined where $\hat{\phi} = [\hat{\rho}, \widehat{\rho u}, \widehat{\rho v}, \hat{T}]$. Here we introduce the wave number β, the frequency $\widetilde{\omega}$ and the perturbation eigenfunction $\hat{\phi}(y)$ which are related to each other through a generalized eigenvalue problem with parameter β and eigenvalue $\widetilde{\omega}$. In general both β and $\widetilde{\omega}$ can be complex. In literature it is common practice to distinguish between temporal instability in which β is real and $\widetilde{\omega} = \widetilde{\omega}_r + i\widetilde{\omega}_i$ is complex and a spatial instability with $\beta = \beta_r + i\beta_i$ and real frequency $\widetilde{\omega}$. Here the spatial setting is most appropriate and a small perturbation with frequency $\widetilde{\omega}$ is predicted to grow exponentially with x, if $\beta_i(\widetilde{\omega}) < 0$. Since there is no a priori known natural frequency $\widetilde{\omega}$ for the instabilities in the wake we determine $\beta_i(\widetilde{\omega})$ for a wide range of frequencies at several x locations for the angles of attack $\alpha = 1.25^o$ and $\alpha = 0^o$. We observe that LST predicts both the $\alpha = 1.25^o$ and $\alpha = 0^o$ solutions to be unstable for a wide range of frequencies in the region behind the airfoil, extending several chord lengths.

In figure 4-11, the dependence of β_i on $\widetilde{\omega}$ for a typical line through the

wake is plotted for the cases $\alpha = 0.0^o$ and $\alpha = 1.25^o$. The spatial growth rate, β_i, is about 8 times larger in the $\alpha = 1.25^o$ case, which is consistent with the strongly increased residual level which arises from a restart with the explicit Runge-Kutta scheme compared to the much lower level found if $\alpha = 0^o$.

4.6 Conclusions

In this chapter we have shown that it is possible to obtain a machine accurate steady state solution to the Euler equations with the minmod limiter and an implicit time stepping scheme. We have applied an asymmetric version of the minmod limiter which removes the non-differentiability in a region where the quotient of the gradients of the solution of adjacent cells is near one. We argue that the stall of convergence of e.g. a time explicit method for the specific flow problem studied in this chapter is caused by a physical instability rather than by the minmod limiter.

The final values of the residual obtained with the factorization scheme depend on the CFL number. A threshold of CFL=1000 is encountered above which the solution does not converge. Fast convergence to machine accuracy can be obtained by starting with a high CFL number and then decreasing the CFL number.

Comparison of the solution obtained with the explicit scheme and the implicit scheme for two different angles of attack, $\alpha = 1.25^o$ and $\alpha = 0^o$, shows that the machine accurate solution of the implicit scheme is unstable. If we change the limiter to Van Leer's limiter (4-17), thus changing the spatial discretisation, we observe the same instabilities. Calculations performed with the second order implicit Crank-Nicolson time integration show a stall in convergence similar to the Runge-Kutta scheme which indicates that a machine accurate solution can only be obtained if too much numerical dissipation is added through either the spatial discretisation or time integration. For all schemes with a small amount of numerical dissipation the convergence stalls. A grid refinement study shows that the region of instability does not vanish on a very fine grid. We have shown that this instability is also predicted by linear stability theory for both angles of attack.

Although there is an instability in the mean flow, it is very weak and relevant quantities such as the drag and lift are not affected within engineering accuracy. The lift and drag for the explicit method and for the implicit method, with CFL number up to 100, are the same up to at least 6 digits. Still, the results show that a stall of convergence is not always due to the numerical scheme but may be caused by a small scale physical phenomenon that cannot be observed if the numerical scheme is too dissipative.

Finally, large scale applications, e.g. three dimensional viscous computa-

tions, with the factorization method require high memory usage and large computation time. Therefore the performance on parallel platforms is of major importance and this will be discussed in the next chapter.

Chapter 5

Acceleration techniques for steady flow computations

5.1 Introduction

Typical computational fluid dynamics problems require large numbers of floating-point operations, and the wall-clock simulation time of a flow problem is an important aspect in particular in relation to design optimization and various other engineering applications. Therefore, efficient acceleration techniques are an important topic of research in CFD. Different computer architectures, for instance based on vector or scalar processors, favor different algorithms, and consequently the choice of algorithm depends to some extent on the available hardware.

In this chapter the properties of the implicit approximate factorization method studied in chapter 4 are investigated further with respect to two types of acceleration techniques: multigrid and parallelization. It is difficult to understand or even address the interplay of the physical instability observed in the previous chapter and the solution process, especially when multigrid is involved, because the effect of the instability on the convergence behavior may depend very sensitively on the grid level. Also for the parallelization, where we use a spatial domain decomposition, the interplay between the number of domains, local accuracy and the instability is not clear. Therefore, these topics form key items of interest in this chapter.

This chapter is organized as follows. In section 5.2 we briefly describe the results of multigrid acceleration. In section 5.3 we discuss the parallelizability of the implicit inviscid flow solver. Because the prospects of a solver for the Navier-Stokes or Euler equations are determined by its actual performance on current or future computers, the competition between implicit and explicit methods is influenced by the increased availability of parallel comput-

ers. So far, explicit methods have benefited most from the developments in computer hardware, because the repetition of many, simple instructions without recurrence is well suited for vector processing, and the sole dependence on data from the previous time step is easily handled on parallel computers by supplying copies where needed. The parallelization experiments described in section 5.3 indicate that a good parallel performance can be expected for the implicit solver described in this chapter as well, because the solver is not too sensitive to approximations made to enable parallel execution. In this research we will focus on spatial domain-decomposition techniques as in Ref. [33]. The reason for this is that implementation on both shared memory systems and distributed memory systems is fairly easy, and it should be possible to achieve comparable parallel performance on both types of parallel architectures. Because the domain-decomposition approach described here affects the convergence rate towards the steady state, in contrast to most explicit multiblock algorithms, (theoretically) perfect scaling with the number of processors is not achieved. The parallel efficiencies, however, seem sufficient for practical purposes.

The simulations described here concern the well-known test case of inviscid flow around a NACA0012 airfoil at a free-stream Mach number of $M_\infty = 0.8$ and an angle of attack of $\alpha = 1.25^o$, which is discussed in more detail in section 4.3. Several test computations for different free-stream conditions were performed which confirm the conclusions from the results for the specific test case selected in this chapter. The case $M_\infty = 0.85$, $\alpha = 1.0$ with stronger shocks, the subsonic case $M_\infty = 0.63$, $\alpha = 2.0$ without shocks and the supersonic case $M_\infty = 1.2$, $\alpha = 7.0$ all produced convergence histories with the same characteristics as found for the case $M_\infty = 0.8$, $\alpha = 1.25$. Therefore, in the main part of this chapter we will restrict the presentation of the results to this case.

In all simulations described in this chapter, the coefficients of drag C_d and lift C_l are found to be equal up to at least six digits, so we will only consider the residuals and the convergence histories in this investigation. We will occasionally refer to the time-stepping towards the steady state as 'convergence of the outer iteration', which should be distinguished from any ('inner') iterative method used to solve the linear system of equations at each time step as defined in (4-15). In general, the convergence towards the steady state can be rated according to the number of time steps required, or according to the overall computing time.

5.2 Multigrid acceleration

In chapter 4 we showed the increased convergence rate towards the steady state of the implicit method compared to the explicit time-integration method, the latter being limited to small CFL numbers for numerical stability reasons. A popular approach to accelerate the convergence towards the steady state for explicit Runge-Kutta time-stepping is the application of a multigrid technique (see e.g. [16, 52]). A first and obvious question is how the convergence of the present implicit solver compares to these accelerated explicit Runge-Kutta methods. Therefore we briefly address the application of implicit and explicit time-stepping in combination with multigrid.

We applied a multigrid technique as described in section 2.6 which was already used in Refs. [16, 52] in conjunction with an explicit time integration method. In this method, nonlinear multigrid is applied to solve q from the steady Euler equations in (4-1), which can be written as a system of coupled nonlinear equations as in (2-42) by

$$H(q) = 0 \tag{5-1}$$

As a relaxation mechanism on the successive grids we use (pseudo) time-stepping for the unsteady Euler equations, as this will push the solution towards the solution of (5-1). Both explicit Runge-Kutta schemes and the Euler Backward time-stepping method described in chapter 4 can be used as time-stepping techniques.

This multigrid method is different from a multigrid technique where the approach is solely used to accelerate the solution of the linear system (4-15) in each time step, as described in [65]. In our implicit Euler Backward scheme, application of a multigrid method to solve the linear system is not attractive, because we only have an approximate Jacobi matrix, and accurate solution of the linear system will not pay off (see section 5.3.2). The multigrid method used in this chapter is closely related to the work reported in [42]. However, in [42] only first order spatial discretisation is applied, and the relaxation method is not based on (pseudo) time-stepping.

In our multigrid method the solution is restricted to coarser grids by injection, and the defect vector by full weighting. The correction to the solution is prolonged to the finer grid by bilinear interpolation. With this choice of the intergrid operators the approximation property is satisfied [38]. The initial solution is improved with Full Multigrid, and each multigrid W-cycle follows the Full Approximation Scheme.

It is known from the literature (see Van der Burg [15]) that the multigrid method does not converge very well for the Euler equations with the MUSCL scheme and several explicit compact storage Runge-Kutta time-stepping schemes.

This is confirmed by our findings. In figure 5-1 a result is shown for the four-stage explicit Runge-Kutta scheme (2-36) and CFL number of 0.6. The speedup in convergence rate compared to the single-grid computation is considerable, but the final residual level is not as low as in the single-grid explicit computation, and convergence itself is only attained for carefully chosen numbers of coarse-grid, pre- and post-relaxations. On the other hand, the implicit

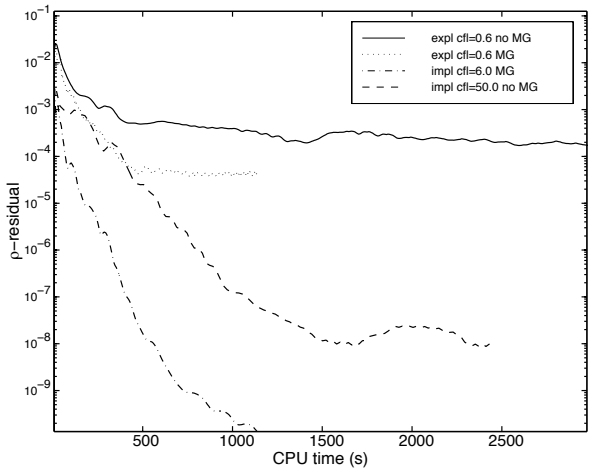


Figure 5-1: Convergence of the multigrid technique applied to attain the steady state of the Euler equations.

method (e.g. at a CFL number of 50) is also accelerated by the multigrid process, but is more robust and convergence arises for a wide range of multigrid parameters and CFL numbers. Again, the final level of the residual depends on the CFL number, but the level found here is well below the final level of the explicit Runge-Kutta scheme of figure 5-1, which is sufficient for engineering accuracy. By choosing different grid levels for the finest grid, we also found the convergence rate to be grid-independent. We observe that the multigrid acceleration is only about a factor two to four depending on the CFL number whereas for these types of problems a speed up factor of 10 is typically found (see e.g [52]). However, this appears to be related to the spatial discretisation. Here we use the MUSCL scheme where in most cases the more dissipative Jameson scheme is used as in [52]. With the Jameson scheme we also found a higher speed up factor comparable to the speed up factor mentioned above.

These results were confirmed by test computations for different free-stream conditions. The case $M_\infty = 0.85$, $\alpha = 1.0$ with stronger shocks, the subsonic case $M_\infty = 0.63$, $\alpha = 2.0$ without shocks and the supersonic case $M_\infty = 1.2$,

$\alpha = 7.0$ all gave convergent multigrid processes when implicit time-stepping was used. The final residual levels and the convergence rates were found to be dependent on the CFL number as in the single-grid computations.

These results show that the implicit scheme has better smoothing properties than the explicit Runge-Kutta scheme [97]. The dissipation in the implicit scheme is sufficient to ensure a convergent multigrid process. For higher-order TVD schemes and explicit time-stepping, the special treatment of defect correction may be required to reduce the residuals at all, as found by Van der Burg [15]. (For a discussion of the application of defect correction, see e.g. [54].) It is noted that the problems with multigrid are typical for the use of a higher-order TVD scheme such as the MUSCL scheme employed in this chapter. For more dissipative spatial discretisations, such as the popular second-order Jameson scheme, rapidly convergent multigrid methods are more readily obtained, also with explicit time-stepping schemes.

For a proper comparison of the implicit and explicit time-stepping methods as smoothers for multigrid, explicit Runge-Kutta schemes other than the compact storage scheme of (2-36) should be tested as well. Tests with the five-stage scheme proposed by Jameson [49] gave no significant improvement of the convergence behavior, which can be understood because the damping of high frequency components is similar for this scheme [52]. This chapter aims at demonstrating the power and simplicity of the implicit scheme. The apparent robustness of the multigrid process without multigrid parameter study or defect correction, is an appealing feature of the implicit scheme.

5.3 Parallelization

All previous simulations were performed on a single processor. Convergence acceleration compared to an explicit method was achieved first (chapter 4) by using an implicit method whereas in section 5.2 the multigrid method was applied to accelerate the convergence for both the explicit and implicit method. In this section we focus on parallelization as acceleration technique. The concept of parallelization is very simple, dividing the work over multiple processors decreases the wall clock simulation time. However, in practice this is not so simple and a good parallel performance depends on a lot of parameters such as e.g. computer architecture and parallelizability of the algorithms.

According to the parallel version of Amdahl's law [79], the parallel performance is limited by the fraction of the program that is necessarily sequential. This can be shown as follows. Assume that a fraction β of the computational work is parallelizable and that a fraction $(1-\beta)$ is essentially sequential. Further assume that all CPU's achieve the same Mflop rate, then the total CPU

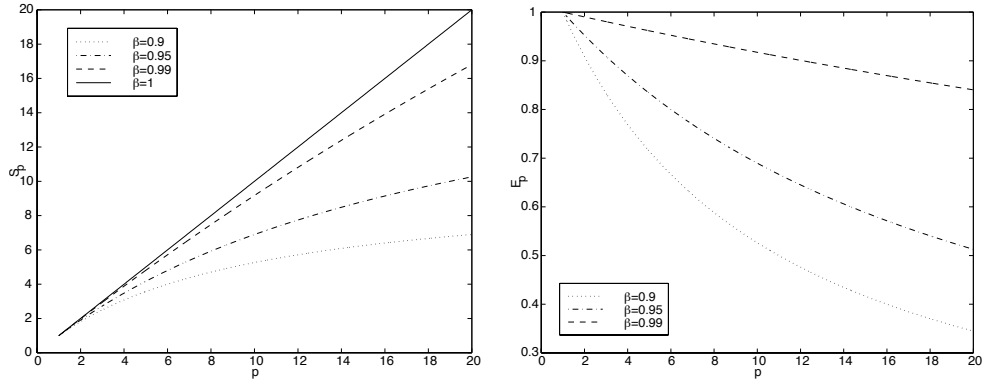
time is given by

$$T_p = (1-\beta)T_1 + \frac{\beta T_1}{p} \tag{5-2}$$

where T_1 stands for the execution time with one CPU, p is the number of processors involved in the parallel part of the computational work and T_p is the total execution time on p processors. For the speed-up factor we find

$$S_p = \frac{T_1}{T_p} = \frac{1}{1-\beta+\frac{\beta}{p}} \leq \frac{1}{1-\beta} \tag{5-3}$$

Clearly the maximum speed-up is bounded by the part of the computational work that is essentially sequential and not by the number of participating CPU's. The sensitive dependence of the speedup factor S_p on the paralleliza-



(a) S_p (b) E_p

Figure 5-2: Speed-up factor, S_p, and efficiency, E_p, as a function of the number of processors for various parallelization fractions β.

tion parameter β is illustrated in figure 5-2. In figure (a) the speed-up factor (5-3) is plotted as a function of the number of processors for various parallelization factors β. The solid line represents the perfect scaling with respect to the number of CPU's, corresponding to $\beta = 1$. Clearly, if 99% of the computational work is parallelizable the deviation from the perfect scaling is already significant for large p. If the fraction β decreases this deviation rapidly becomes larger. Defining the efficiency, E_p, of a CPU by $E_p = S_p/p$ this effect is clarified further in figure 5-2 (b). A small decrease in β results in a considerable decrease of the efficiency per CPU. The above results show

that parallelization is not as trivial as it looks at fore hand and good parallel performance is only achieved for values of β very close to 1.

In order to establish the theoretical parallel performance of an implicit solver as compared to the parallel performance of an explicit solver, it is sufficient to consider only that part of the implicit program that is essentially different from its explicit counterpart. The implicit program has three stages: the calculation of the Jacobi matrix and the numerical flux corresponding to the old solution, the matrix inversion, and the calculation of the new solution. Because the calculation of the Jacobi matrix is very similar to the calculation of the numerical flux, the first and third stages in the implicit algorithm do not differ in nature from any flux calculations in explicit programs. These stages can be done in parallel as in every explicit method with spatial domain decomposition. Therefore, for our present goal it is sufficient to consider only the solving phase (4-16) of the implicit factorization method to establish the theoretical parallelizability. On a single processor and in the present implementation this stage takes about thirty percent of the total computation time for each time step. If this portion is not parallelizable, this would lead, according to Amdahl's law, to a maximum speedup of 3 on any parallel computer, which would be a severe limitation on the performance of the solver.

5.3.1 Domain decomposition

Domain-decomposition methods have gained wide acceptance in solving the compressible Navier-Stokes or Euler equations. The suitability for parallelization is clear because of the presence of coarse grained loops over the subdomains in the program. These loops over the subdomains are usually limited in number and contain sufficient work to make parallel overhead relatively unimportant. Also, the possibility of local data storage is attractive, especially on distributed systems. Locality of data in this context means that the required data reside in the local memory associated with a processing element for the duration of the computation, with only exchange of data at the 'boundaries' of the subdomains.

The calculation of the Jacobi matrix and the fluxes can be done with domain decomposition in a familiar way, by introducing dummy points. It seems worthwhile if the matrix inversion part can be treated similarly, because then all the important subroutines of the program can compute with local data, with only boundary data exchange at appropriate stages.

In the sequential version of the matrix inversion, we cover the grid typically by starting at a far-field point, e.g. a far-field corner point or the far-field point in the wake, and traversing the field systematically, taking at each point the values of the neighboring points into account. In this way at each point, two out of four neighboring points have been updated already. In the second

stage of (4-16), the field is crossed in the opposite direction. This has no advantage for parallel execution, because, if standard domain decomposition is applied, each processor has to wait until the processor of the neighboring domain has completed its computations. Nevertheless, domain decomposition is attractive, especially when other parts of the program, for instance the flux calculation, already use domain decomposition for parallel execution. This is in contrast to a parallel computation in which the grid points are renumbered, such as the 'wavefront method', which is designed to solve tridiagonal systems similarly to the method used to solve (4-16) in this chapter. The wavefront method does give the correct results, but the method is not scalable, because at the beginning and at the end of the sweeps some processors are idle. Other disadvantages are that the data must be divided over the processors in a different way, taking less advantage of the structured grid, and that synchronization takes place after each wavefront line calculation.

Instead we choose the simplest form of domain decomposition, by ignoring the errors made at the subdomain boundaries, and start the iteration in each subdomain independently of the other subdomains. This does not affect the quality of the obtained steady state solution. If the iteration process converges to a sufficient level the accuracy is determined by the spatial discretisation and not by the iteration process. To obtain the closest resemblance with the performance of the sequential code, we seek the best way of crossing the grid when solving (4-16). We found that the best results with respect to convergence rate and final residual level are obtained when the domain-decomposed field is crossed by starting from opposite corners in each pair of adjacent subdomains in the first stage, exchanging boundary data with the neighboring subdomains after completing the first sweep, and then proceeding in the opposite direction, returning to the starting point. This is depicted in figure 5-3. Thus the field is traversed in a 'continuous' manner, in the sense that after each sweep each block starts with new values, either because of a flux and Jacobi matrix update after a whole time step, or because new values have been 'supplied' by the neighboring block.

However, if the number of subdomains is larger than two, the two sweeps of the iterative process do not cover the entire grid, as the information from the boundaries travels only across two adjacent blocks after two stages. This is likely to influence the overall convergence of the method. Because the gain of parallel computation may outweigh this loss of convergence, it is necessary to study the convergence rate for several domain divisions.

5.3.2 Numerical results

In this section the convergence behavior for several domain divisions is investigated. Since we are mainly interested in the convergence behaviour the

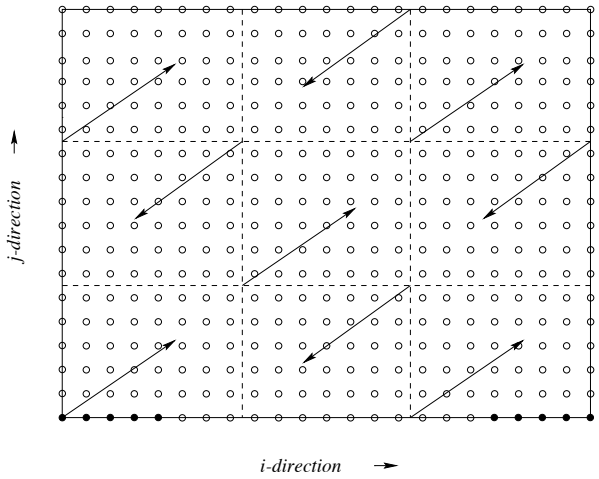


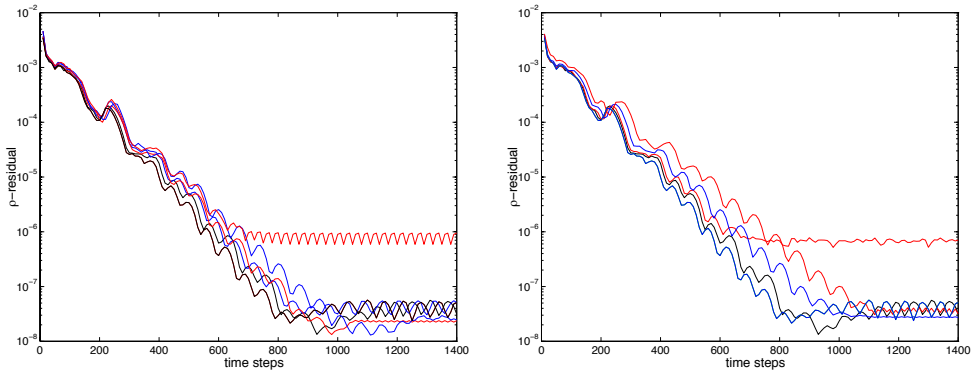
Figure 5-3: First stage in the traversal of the C grid in the factorization method with domain decomposition. In the second stage the arrows are reversed. The grid is shown in the computational domain. The dark grid points on the domain boundary are joint points in the wake behind the airfoil.

simulations are performed on a single CPU. Finally, the actual parallel performance is demonstrated on a Cray T3E distributed memory computer. We studied the convergence for several divisions up to a total of sixteen subdomains, where the number of subdomains is varied in both the i- and the j-direction. Possible block lines in the wake region were chosen to be continuous across the wake line. The block divisions are indicated as $n_i \times n_j$, n_i being the number of divisions in the i-direction and n_j the number in the j-direction. The division in figure 5-3 is 3×3.

In figure 5-4 convergence histories are plotted for various subdomain divisions. The fastest convergence is achieved with the 1×1 single-block division as expected. The convergence deterioration for different subdomain divisions is, however, rather small.

The domain decomposition is seen to have two effects. Firstly, the convergence rate is somewhat smaller if more subdomains are used. Moreover, faster convergence is obtained with more divisions in the i direction than in the j direction. This suggests that the convergence-rate deterioration is related to the number of interface points created by the subdomain division: because of the 289×65 grid, a single bisection in the j-direction creates a larger boundary portion than a bisection in the i-direction.

Secondly, the final convergence level may occasionally deviate considerably from the expected value. In figure 5-4 the simulations with 5×1 and 2×2



(a) domains in i-direction (b) domains in j-direction

Figure 5-4: Convergence behavior of the approximate factorization method at a CFL number of 50 a) for domain divisions 1×1, 2×1, 4×1, 5×1, 8×1 and 16×1 and b) for domain divisions 1×1, 2×1, 2×2, 2×4, and 2×8. The fastest convergence is obtained with the 1×1 division (most left curve in both figures a and b). Note the two simulations 5×1 and 2×2 with different final residual level.

result in a higher residual level. It is not well understood why this occurs for these specific domain decompositions. However, it is undesirable that the final results are influenced by the block division and thus this problem should be eliminated. The solution to this problem is obtained from the fact that the approximate factorization method gives a solution of the linear system only approximately. Our computations (we will come to this point shortly) suggest that the solution of the linear system with approximate factorization is just accurate enough to retain convergence. The difference between the single-block grid and the domain-decomposed grid is that for the latter internal block boundary points are treated with Jacobi relaxation instead of Gauss-Seidel relaxation in the first sweep, so that the accuracy required for fast convergence behavior is no longer attained. This suggests that we should solve the linear system more accurately when more blocks are used. To obtain a more accurate solution we first experimented with four Gauss-Seidel sweeps rather than two in the standard approximate factorization method (4-16). The results have been plotted in figure 5-5a, where the residual of the density has been plotted as a function of the number of outer iterations for several domain divisions.

At the cost of more CPU time per outer iteration step, the simulations in

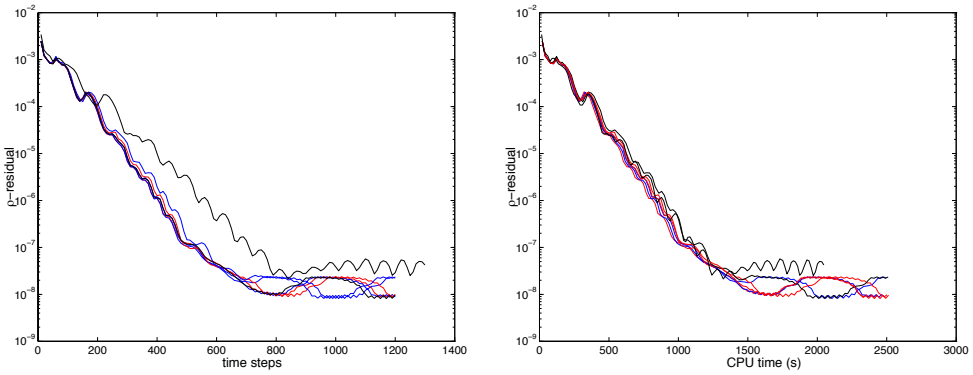


Figure 5-5: Density residual a) as a function of the number of outer iterations (time steps) and b) as a function of CPU time, for various domain divisions, but calculated with four Gauss-Seidel sweeps per outer iteration. The solid line (most right in figure a) is the single-block line of the approximate factorization method with two Gauss-Seidel sweeps. The different subdomain divisions can hardly be discerned.

figure 5-5a display a more favorable convergence behavior than the simulations in figure 5-4a with the same block divisions. Firstly, the convergence stall is eliminated, which was the principal goal. Secondly, the convergence towards the steady state requires fewer time steps, and the convergence rate (in terms of time steps) is also less influenced by the domain decomposition. If the residual level is plotted as function of the CPU time instead of time steps, we even see that the improved convergence behavior comes with no extra computational costs (figure 5-5b).

The convergence rate of the residual is dependent on the accuracy of the solution of the linear system. To quantify this, we introduce the relative error ε_{rel} in the solution Δq as

$$\varepsilon_{\text{rel}} = \frac{||\mathcal{M} \cdot \Delta q + H||}{||H||}, \tag{5-4}$$

where $||\cdot||$ is the discrete L_2 norm of the state vector over the entire grid and $\mathcal{M}\Delta q$ represents the left-hand side of (4-15). We have repeated the computations with increasing number of Gauss-Seidel sweeps until a prescribed relative error ε_{rel} is reached at each time step. In figure 5-6a the convergence history is plotted as a function of outer iteration steps and in figure 5-6b as a function of the CPU time for a single CPU. Figure 5-6a shows that we cannot improve the convergence rate indefinitely by solving the linear system

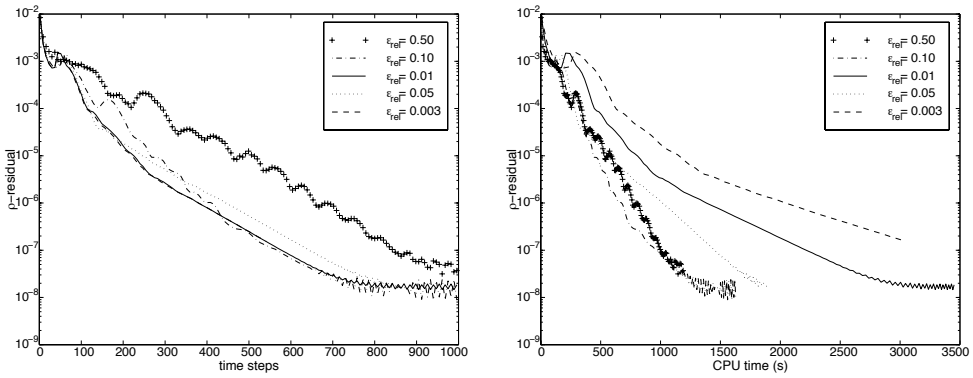


Figure 5-6: Density residual a) as a function of the number of outer iterations (time steps) and b) as a function of CPU time, for increasing accuracy of the solution of the linear system.

of equations more accurately, partly because the linear system contains only an approximation of the exact Jacobi matrix (see, e.g., Venkatakrishnan [89] or Tan [84], and references quoted therein). An optimal accuracy regarding computational performance is found for ε_{rel} between 0·1 and 0·5. Between these numbers the actual CPU time is of the same order of magnitude (figure 5-6b). This relative error range corresponds to roughly one to six Gauss-Seidel sweeps. Because, apparently, a large relative error in the solution is allowed, the use of sophisticated linear solvers like GMRES or Conjugate Gradient methods [8] will probably not be very rewarding in view of CPU costs.

To demonstrate the effectiveness of the Gauss-Seidel relaxation (see section 2.5) we plotted the relative error ε_{rel} as a function of the number of sweeps or "iterations" in figure 5-7 for Red-Black relaxation (see e.g. [79]) and for point Gauss-Seidel methods with different sweep directions. From this figure it is clear that the alternating or symmetric Gauss-Seidel method is most efficient in reducing the error in the solution of the linear system, although the asymptotic convergence rates of the displayed methods are not very different. This shows that, because of the modest accuracy that is required for the implicit method described in this chapter, the efficiency of linear solvers in this context is determined by their performance in the initial sweeps, and not by their asymptotic convergence rate.

We verified that, as long as the relative error as defined by (5-4) is prescribed, the outer convergence rate, i.e. the required number of time steps towards the steady state, is retained, even if different approximate linear solvers, like for instance Red-Black relaxation, are used. Moreover, it was

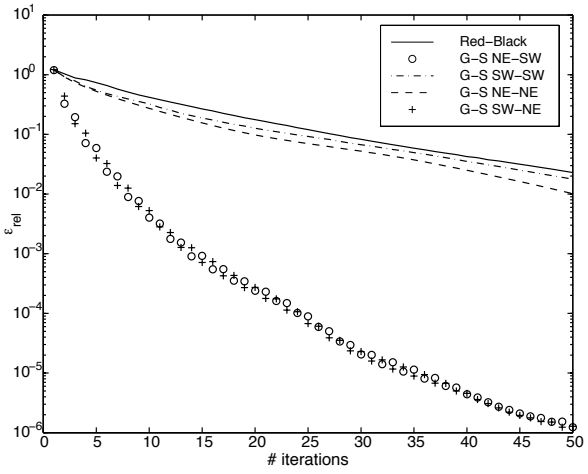


Figure 5-7: Relative error $\varepsilon_{\rm rel}$ as a function of the number of iterations for Red-Black relaxation and for point Gauss-Seidel relaxations with different sweep directions. Note that one Red-Black relaxation is defined as the treatment of both the "red" and the "black" points. In this plot the alternating (or symmetric) Gauss-Seidel relaxation is most efficient. Note that two iterations of Symmetric Gauss-Seidel correspond to the original "approximate factorization" approach.

checked that, if domain decomposition is applied and $\varepsilon_{\rm rel}$ is prescribed rather than the number of Gauss-Seidel relaxations, both the outer convergence rate and final convergence level are preserved for all subdomain divisions, although possibly now at the price of more Gauss-Seidel relaxations.

We remark that the results of this section indicate that the use of square subdomains in computational space is favorable for the total computation time. We have seen that the loss of convergence rate is mainly due to a less accurately solved linear system. Because domain decomposition introduces internal boundary points which are treated with Jacobi rather than Gauss-Seidel relaxation, on average more relaxation sweeps are required to obtain the same solution-vector accuracy. Therefore, subdomain divisions with the least number of internal boundary points, i.e. square subdomains, are favorable (this was already mentioned to explain the preference for bisections in the i-direction rather than in the j-directions). Although it is not inconceivable that the flow geometry introduces some preference direction for subdomain divisions, the present results suggest that the numerical error caused by the simplified treatment of internal boundary points is proportional to the number of internal boundary points, and outweighs any physical preference direction.

To demonstrate the parallel performance, parallel computations were done on a Cray T3E distributed memory computer. As an example we show the performance for domain decomposition with subdomain divisions only in the i-direction, the number of subdomains matching the number of processors exactly. In the case of 32 CPUs this is hardly optimal, because the subdomains are far from being square, but the performance is still quite good for this relatively small problem, as shown in figure 5-8.

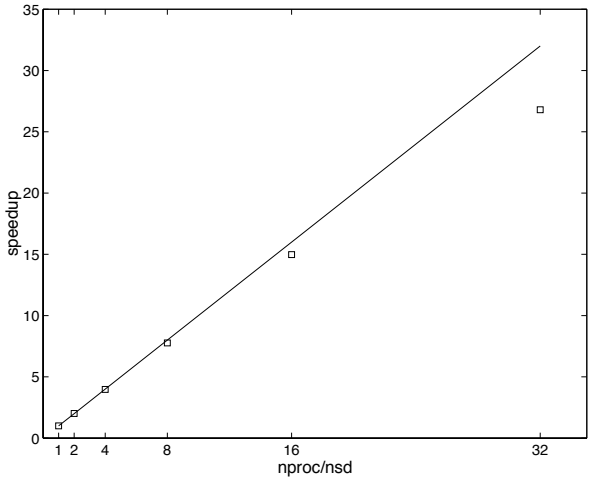


Figure 5-8: Parallel speedup as a function of the number of CPUs on the the Cray T3E. The domain decomposition was only done in the i-direction, and the number of subdomains was chosen equal to the number of processing elements.

5.4 Conclusions

In this chapter we discussed the multigrid and parallel performance of an implicit solver for the compressible Euler equations based on a quasi Newton iteration and an approximate factorization method. It is found that the convergence of the outer iteration is determined by the accuracy of the solution of the linear system that results from the quasi-Newton iteration, and that "approximate factorization" is a cheap, two-stroke point Gauss-Seidel method to solve the linear system of equations with sufficient accuracy. The relative error ε_{rel} defined by (5-4) is a suitable measure for outer iteration control when different linear solvers are used. Because the accuracy required to achieve convergence is low, the high efficiency of the initial sweeps of the alternating point

Gauss-Seidel relaxation pays off. It was found that Red-Black relaxation for this problem is between two or three times less efficient than Gauss-Seidel.

With straightforward domain decomposition a parallelizable multiblock code is obtained with roughly the same convergence rates as the single-domain problem. As discussed in the introduction, the convergence rate and final level of residuals depend on the CFL number. The domain-decomposition method affects the accuracy of solving the linear system of equations resulting from the quasi Newton method. By prescribing a fixed accuracy of the solution of the linear system, we retain the convergence rate and the final residual level of the Euler Backward time-stepping for the single block code. Although not mentioned in section 5.3.2 divisions up to 50 subdomains and more were checked to have no serious effect on the overall computation time. This does not only increase the theoretical performance of the solver, but the apparent insensitivity to the domain decomposition is also promising for cases where the geometry is less simple.

In the introduction it was mentioned that perfect scaling with the number of processors is not achieved. Should this cause any problems, it is possible to turn to different iterative solution methods of the linear system, such as the Red-Black method. With this method, if communication costs are neglected, perfect scaling is possible, because for each subdomain division the same numerical problem is solved. Experiments with different relaxation methods show that, as long as the linear system is solved with the same accuracy, similar convergence behavior is found with respect to the number of time steps required and the final residual level.

The implicit time-stepping method described here is also very suitable when it is used as a multigrid smoother. Convergent multigrid results are found quite robust and can be obtained without any careful multigrid parameter study, in contrast to multigrid with explicit Runge-Kutta time-stepping, for which convergence depends sensitively on the multigrid parameters. The overall speed up factor is not spectacular which appears to be related to the spatial discretisation.

Although in this chapter we studied inviscid flow with the Euler equations, we expect similar results for the compressible Navier-Stokes equations with respect to parallelization. In chapter 6 a fully implicit scheme is applied to an unsteady two-dimensional shock boundary-layer interaction flow over a flat plate. For this implicit method the main difference with the present implicit method applied to the inviscid flow lies in the contribution of the viscous flux. In [82] it was shown by Streng et al. that the calculation of the viscous flux is highly parallelizable so we expect a good parallel performance of the fully implicit method as well. Also, extensions to three-dimensional problems will not pose problems regarding the algorithm. In [71] a similar numerical method

was successfully used for 3D Navier-Stokes involving implicit time-stepping, quasi-Newton's method and factorization.

With respect to the multigrid method the extension to the Navier-Stokes equations is not so straightforward. In chapter 7 we show that no multigrid acceleration may be expected due to the natural unsteadiness of the flow.

Chapter 6

Shock boundary-layer interaction flow

6.1 Introduction

Due to the recent development in computer capacity, numerical methods and turbulence models, numerical simulation of complex unsteady flow has come within reach. Examples are the simulation of unsteady viscous flow modelled by the Reynolds-averaged Navier-Stokes equations and direct numerical simulation (DNS) and large-eddy simulation (LES) of turbulent flow. In the past efficient numerical methods have been developed for the calculation of steady flows. Usually the flow equations are advanced in time starting from an arbitrary initial condition using an implicit or explicit time integration method supplemented with convergence acceleration techniques. As only the steady solution is required the accuracy with which the time integration is performed is not an issue. However, if unsteady flow is simulated the accuracy of the time integration has to be considered and many efficient methods for steady flow simulation can no longer be applied in a straightforward way. Explicit methods, in which the time step is restricted by numerical stability requirements, may become practically useless in particular if a large resolution is required in directions normal to solid walls. On the other hand, in A-stable implicit methods, where the time step is not restricted by stability, accuracy considerations should determine the magnitude of the time step. Depending upon the "accuracy time step" an implicit method may or may not be more efficient than an explicit method. Ultimately, one would like to determine the accuracy time step dynamically during a simulation and possibly switch between explicit and implicit time integration schemes depending on the necessary accuracy and efficiency, as the flow develops.

As a first step towards such a dynamical time step determination we study

in this chapter a DNS of a two-dimensional unsteady shock boundary-layer interaction flow over a flat plate under an adverse pressure gradient. We use both an explicit Runge-Kutta method and the implicit Crank-Nicolson scheme and study the following questions. How large can the time step be chosen in order to resolve the unsteady solution accurately? How accurate should the nonlinear system of equations in each time step be solved and how sensitive does the prediction of different flow quantities, such as mean quantities and instantaneous quantities, depend on these numerical parameters? Is it possible to solve the nonlinear set of equations at each time step and for all relevant step sizes? What implicit method is computationally efficient?

In the selected test case the unsteadiness is caused by the interaction of the boundary layer with shocks that arise in the flow [14]. These phenomena combine into a rather complex unsteady flow which makes it an appropriate model for the present study. Simultaneously, it has some importance for various situations of practical interest like transonic flight conditions or a combustion ramjet. Furthermore, in this flow the time step required for temporal accuracy is relatively large compared to the stability time step of an explicit scheme so that an implicit scheme could become computationally competitive. However, we emphasize that the particular flow and numerical method studied here are only examples. The approach proposed can be applied to more general unsteady flows and is particularly useful for unsteady RaNS simulations where the stability time step of explicit methods is very restrictive (see e.g. [6]).

In the past a lot of work has been done concerning the questions which were mentioned above. In Ref. [69] a 2D turbulent flow around a pitching airfoil was considered. The time accuracy of these simulations was validated by varying the number of time steps per period of the pitching frequency. A pseudo time derivative was added to facilitate the solution of the nonlinear set of equations to be found each time step. The required convergence level needed at each pseudo time step was investigated by focussing on some typical flow quantities. In Ref. [24] a similar two- and three-dimensional unsteady flow around a 2D airfoil and 3D wing was considered. An application which is more closely related to the present flow is the DNS performed in Ref. [71] concerning a spatially evolving 3D turbulent boundary layer. In that study an implicit time integration scheme was used to follow the evolution of the flow in time. The number of time steps per period of the disturbances introduced at the inflow boundary used by these authors was 600. A fixed number of iterations was performed for the nonlinear system at each time step. The validation of this number of iterations was done by comparing the solution obtained with 2, 3 and 4 iterations per time step. In Ref. [6] the thin-layer Navier-Stokes equations are solved for a shock-induced oscillatory flow over

an airfoil. This flow is unsteady without any external unsteady driving effect which is also the case for the flow simulated in this chapter. The temporal accuracy is studied by varying the number of time steps per period of the natural oscillation of the shock.

The flow studied in this chapter is unsteady due to the specific properties of a steady blowing and suction profile [96] at the upper boundary of the computational domain where we prescribe the normal velocity. In contrast to Ref. [6] we cannot relate the time accuracy to the temporal behavior of the external flow condition, as in the examples described above, since we do not prescribe any explicit external flow unsteadiness. Therefore we need another criterion to determine a suitable time step. For this purpose we determine the global error caused by the spatial discretisation and use this as an upper bound for the global error due to the time integration. The global error not only depends on the magnitude of the time step but also on the accuracy with which the nonlinear system arising at each time step is solved and to some extent also on the relaxation method that is used to obtain or accelerate convergence. The obvious demand that the global error should decrease if at each time step the solution is determined more accurately, results in a set of requirements that relate the above mentioned numerical parameters. The specific construction of a stopping criterion that determines the local accuracy turns out to be a key factor in order to satisfy this demand on the global error.

The contents of this chapter is as follows. In section 6.2 we state the governing equations and the numerical method using the explicit Runge-Kutta scheme. In section 6.3 we present the numerical results obtained with the explicit method which are used as a reference solution for the implicit time integration method which is introduced in section 6.4 . Section 6.5 contains a discussion on the global error bounds for the implicit time integration scheme and the stopping criterion. Numerical simulations for the implicit scheme are presented and the resulting accuracy is evaluated with respect to the explicit reference solution. The convergence problems arising at large time steps are addressed in section 6.6 and a connection is made with classical chaos theory. Finally, in section 6.7 the conclusions are summarized.

6.2 Governing equations and explicit numerical method

In this section we state the equations governing viscous compressible flow and include the viscous contribution to the explicit stability time step defined in section 4.2.3.

6.2.1 Governing equations

The governing equations for two-dimensional compressible viscous flow are the Navier-Stokes equations. In conservation form and Cartesian coordinates they read

$$\frac{\partial q}{\partial t} + \nabla \cdot (f_c - f_v) = 0 \tag{6-1}$$

with $q = [\rho, \rho u, \rho v, E]^T$ and where f_c and f_v denote the inviscid and viscous fluxes respectively. The inviscid flux is given by $f_c = [f, g]$ where f and g are defined in (4-2) and the viscous flux is given by

$$f_v = \begin{bmatrix} 0 & 0 \\ \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \\ (u\sigma_{xx} + v\sigma_{xy} + kT_x) & (u\sigma_{yx} + v\sigma_{yy} + kT_y) \end{bmatrix} \tag{6-2}$$

The heat conductivity k equals

$$k = \frac{1}{Pr(\gamma - 1)M_\infty^2} \frac{\mu}{Re} \tag{6-3}$$

where μ is the non-dimensional viscosity, Pr the Prandtl number, γ the adiabatic gas constant, M_∞ the Mach number at infinity and Re the Reynolds number to which we return shortly. Here we use $Pr = 0.72$ and $\gamma = 1.4$. The temperature T is related to the density ρ and the pressure p by the ideal gas law

$$T = \gamma M_\infty^2 \frac{p}{\rho} \tag{6-4}$$

The dimensionless viscosity μ is related to the temperature T by Sutherland's law,

$$\mu(T) = \frac{1 + C}{T + C} T^{\frac{3}{2}} \tag{6-5}$$

where we use $C = 0.4$ which corresponds to a reference temperature $T_\infty = 276K$. For a Newtonian fluid such as air the elements of the shear stress are given by

$$\begin{aligned} \sigma_{xx} &= \frac{\mu}{Re}\left(\frac{4}{3}u_x - \frac{2}{3}v_y\right) \\ \sigma_{yy} &= \frac{\mu}{Re}\left(\frac{4}{3}v_y - \frac{2}{3}u_x\right) \\ \sigma_{xy} &= \sigma_{yx} = \frac{\mu}{Re}(u_y + v_x) \end{aligned} \tag{6-6}$$

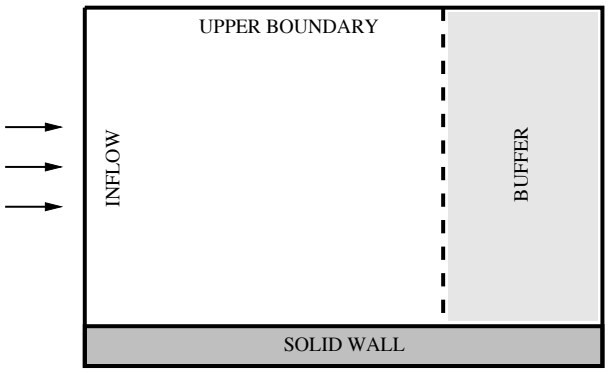


Figure 6-1: Computational domain

Here $Re = (\rho_\infty u_\infty \delta_i^*)/\mu_\infty$ is the reference Reynolds number. The above variables and equations have been made dimensionless using reference scales, i.e. a reference length δ_i^* which is taken as the displacement thickness at the inflow, density ρ_∞, velocity u_∞, temperature T_∞ and viscosity μ_∞. The pressure and energy density are in units $\rho_\infty u_\infty^2$ and time is scaled by δ_i^*/u_∞. The subscript ∞ refers to the free stream values.

6.2.2 Spatial discretisation

To solve (6-1) we use a finite volume method on a structured grid that computes the flux over the control volume edges as described in section 4.2.2. The corresponding numerical flux h in (4-4) now consists of a inviscid and viscous flux which are approximated in different ways. For the inviscid terms we use the higher order TVD scheme described in section 4.2.2. The MUSCL technique with this specific limiter has already been successfully used in [94] for an unsteady 3D flow. The viscous flux is approximated with a second order accurate conservative scheme as defined in [93]. Since the viscous flux contains second order spatial derivatives the standard approach is to find an appropriate approximation of the first order derivatives on the control volume edges which has been discussed in more detail in section 2.3.2.

6.2.3 Boundary conditions

In figure 6-1 the computational domain is sketched. The solid wall is represented by an adiabatic no slip boundary. At the inflow boundary we impose the Blasius solution to the compressible boundary layer equations as described in [96]. For the supersonic flow problem studied in this chapter the inflow boundary can be split into two parts. The first part is the region near the solid wall where the flow is subsonic due to the no-slip boundary condition. Based on the

characteristics for inviscid flow (see [95]) one numerical boundary condition is necessary. In total we extrapolate the density from the interior and impose the pressure and the streamwise and normal velocity components. The flow in the second part of the inflow boundary, away from the solid wall, is supersonic and convection dominated. Here all characteristics enter the computational domain and all flow quantities can be imposed. For the outflow boundary we use a special buffer technique as developed in [96]. With this approach the disturbances in all the solution components are gradually reduced to zero within the buffer domain. The buffer acts on all components of the state vector and can be described by the following formula

$$q = q_{ref} + \widetilde{\zeta}(x)(\widetilde{q} - q_{ref}) \tag{6-7}$$

where $\widetilde{\zeta}$ is the effective buffer function that contains a specific buffer function ζ which will be outlined below, $\widetilde{q}$ is the solution after the flux update and before applying the buffer technique and q_{ref} is the reference solution with respect to which the fluctuations in the solution are defined and damped in the buffer. In this chapter we use the similarity Blasius solution at the outflow position as the reference solution. The buffer function, ζ, is specified by

$$\zeta = (1 - C_1 x_b^2)\left(1 - \frac{1 - e^{C_2 x_b^2}}{1 - e^{C_2}}\right) \tag{6-8}$$

where C_1 and C_2 are tuning parameters. The buffer coordinate x_b is defined by

$$x_b = \frac{x - x_s}{x_e - x_s} \tag{6-9}$$

where x_s and x_e are the x-coordinates at the start and end of the buffer domain respectively. The effect of the buffer region for a certain flow configuration depends on the number of times that the buffer function is applied. Hence, if the grid would be refined the buffer function would be applied more frequently if one uses an explicit time integration scheme. In order to make the buffer procedure independent of the number of time steps we take the effective buffer value $\widetilde{\zeta}$ as

$$\widetilde{\zeta} = \zeta^{C_3 \Delta t} \tag{6-10}$$

where the constant C_3 is added as a tuning parameter. In ref. [96] an extensive investigation has been performed fixing suitable values of the constants in (6-8) and (6-10) which leads to $C_1 = 0.005, C_2 = 20$ and $C_3 = 26.4$ for the present flow.

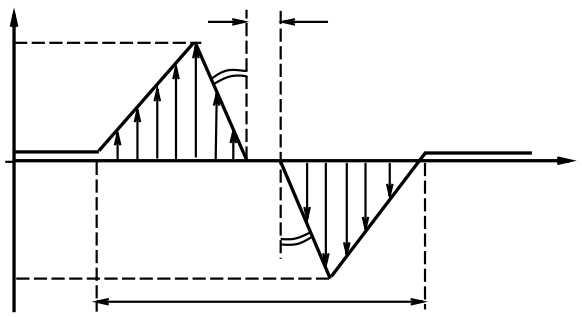


Figure 6-2: Blowing and suction profile of the normal velocity, v at the free stream boundary.

The upper boundary acts as a free stream boundary where a time independent blowing and suction profile is prescribed by imposing the normal velocity, see figure 6-2. Along this artificial boundary not all dependent variables are specified by physical boundary conditions and hence a numerical boundary condition is added. In the region where suction is applied three characteristics leave the computational domain and one characteristic enters the domain. Therefore three variables are determined by extrapolation from the interior domain and the fourth variable is determined by the prescribed normal velocity. In the region where blowing is applied three characteristics enter the computational domain and one characteristic leaves the domain. Again, the normal velocity is prescribed and the remaining variables are determined by coupling the incoming and outgoing characteristics with a locally one dimensional non-reflecting boundary condition as defined in [70, 96].

6.2.4 Explicit time integration

For the explicit time integration we use the second order accurate explicit four stage compact storage Runge-Kutta scheme defined in section 2.4.1. In section 4.2.3 the stability time step for the inviscid flow was derived. Here we include the contribution of the viscous flux into the stability time step. The time step of the explicit scheme is bounded for stability reasons and for the control volume $\Omega_{i,j}$ the local stability time step equals

$$\Delta t_{i,j} = \frac{1}{\dfrac{1}{\Delta t^c_{i,j}} + \dfrac{1}{\Delta t^v_{i,j}}} \tag{6-11}$$

where $\Delta t^c_{i,j}$ is the local time step limitation related to the inviscid flux and analogously $\Delta t^v_{i,j}$ is related to the viscous flux. The inviscid time step limitation is given by (4-7) and the local viscous time step limitation is given by

$$\Delta t^v_{i,j} = \frac{\sigma \Omega^2_{i,j}}{\alpha \left(\mid l_{i+\frac{1}{2},j} \mid^2 + \mid l_{i,j+\frac{1}{2}} \mid^2 \right)} \tag{6-12}$$

where $\alpha = \frac{4}{3}\mu$. In this chapter we will consider a time dependent flow and the time step, Δt, is taken equal to the minimum of all local stability time steps $\Delta t_{i,j}$.

6.3 Explicit numerical reference results

In this section we present some simulation results for the reference test case described in [96] using the Runge-Kutta scheme as specified above. As an initial condition we use the compressible Blasius boundary layer similarity solution. For the present test case we take the Mach number equal to $M_\infty = 1.3$ and the Reynolds number equal to $Re = 500$ based on the inflow displacement thickness. The length and height of the computational domain and the length of the buffer domain are respectively $L_x = 500$, $L_y = 30$ and $L_b = 50$. We use an orthogonal grid with 193×65 points in the streamwise and normal direction respectively. The grid is uniform in the x direction while it is stretched in the y direction using a rational stretching function with a maximal stretching ratio of $\Delta y_{max}/\Delta y_{min} = 8.6$. The blowing and suction profile described in the previous section is displayed in figure 6-2. The blowing and suction profile starts at a distance of 49.6591 from the inflow boundary. The specific choice of the parameters a, d, w and ϕ influences the shock strength and position as well as the temporal behavior of the flow (for more details see [96]). To obtain an unsteady flow with sufficiently strong shocks suitable for the present study we take $a = 0.12, d = 36, w = 300$ and $\tan(\Phi) = 0.0033$. With these settings the flow shows a strong interaction between the boundary layer and the shocks which occur in the flow [96]. Data sampling starts after the flow becomes statistically stationary. The transient process which leads to this state is illustrated in figure 6-3 where we plot the shock sensor defined by

$$S(t) = \max \left| \frac{\partial p}{\partial x}(t) \right| \frac{\Delta x}{p_\infty}. \tag{6-13}$$

This is a measure for the maximum shock strength in the flow on a given grid.

The temporal behavior of the shock sensor indicates that after a certain period the flow becomes statistically stationary. As an initial condition for

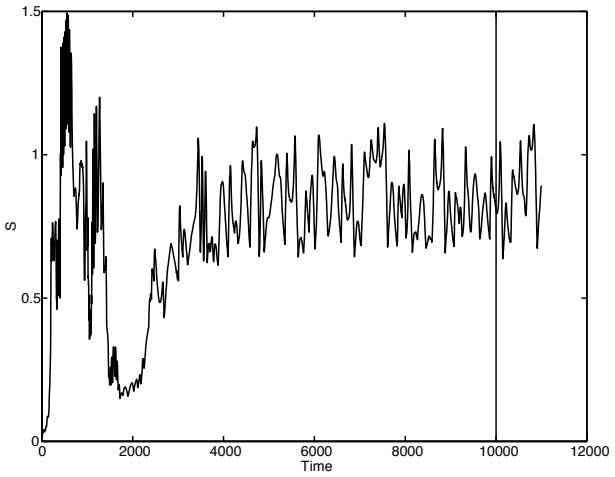


Figure 6-3: Shock sensor as a function of time. The vertical line represents the beginning of the data sampling interval which lies well within the statistically stationary region.

the sampling period throughout this chapter we take the solution obtained at $t = 10,000$. The sampling is performed in the interval T from $t = 10,000$ to $t = 11,000$ in which period we determine time averaged and r.m.s. (root mean square) values of fluctuating flow quantities. We used several CFL numbers, varying from $\sigma = 0.375$ to $\sigma = 2.0$. For the latter the flow becomes unstable in accordance with the stability limit of the Runge-Kutta scheme. We observed that the L_2-norm of the difference in the solution at $t = 11,000$ obtained with values of $\sigma = 1.5$ and smaller is of the order 10^{-4} and therefore we may safely take $\sigma = 1.5$ for the explicit reference simulation. In figure 6-4 the time averaged Mach field is plotted with increment $\Delta M = \max(M)/20$ with $\max(M) = 1.6255$. The presence of two shocks and a separation bubble can clearly be observed. The solid line in figure 6-5 represents the mean skin friction which displays the existence of a region of separated flow. Additionally, the instantaneous streamwise velocity components at two locations in the flow are plotted in figures 6-6 and 6-7 (solid lines). The significance of the dashed lines is outlined in section 6.5.1. The first location ($U1$) is in the boundary layer just in front of the separation bubble. The second location ($U2$) is within the separation bubble itself. We observe that the temporal behavior at the second location is more complex than at the first location and it may be expected that the sensitivity of the solution at the two locations to changes in e.g. the time step will be quite different as well. Therefore, we will use both locations to study the effect of large time steps in the next sections.

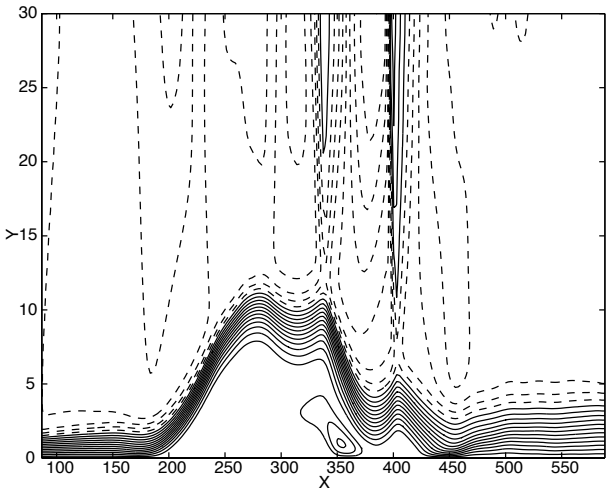


Figure 6-4: Time averaged Mach field over the sampling period T with $M < 1$ (solid) and $M \geq 1$ (dashed).

6.4 Implicit time integration method

In the previous section a second order accurate explicit Runge-Kutta method was used for the time integration. The disadvantage of explicit methods is the unavoidable restriction of the time step due to numerical stability requirements. The time step needed for a time accurate simulation may be significantly larger than the stability time step for explicit methods given by (6-11). For obvious reasons one would like to simulate with this accuracy time step. In order to circumvent the stability requirements on the time step we adopt an implicit time integration method. In this section we define the implicit time integration method. The resulting nonlinear system of equations will be solved using a pseudo time step method which was successfully applied in transonic flow computations in [20]. Although there exist numerous other methods that would be suitable to solve the resulting set of equations such as methods based on local linearisation or ADI type schemes we use the pseudo time stepping approach since our main focus is on the determination of the accuracy time step for different types of flow quantities. In the latter context the solution method determines only the computational effort and not the accuracy time step and hence the distinction is not relevant here.

The main focus will be on the relation between the global accuracy of the solution obtained with this implicit method and the choice of the time step and other numerical parameters which will be defined in detail below. In order to determine the accuracy of the solution we compare the implicit solution with

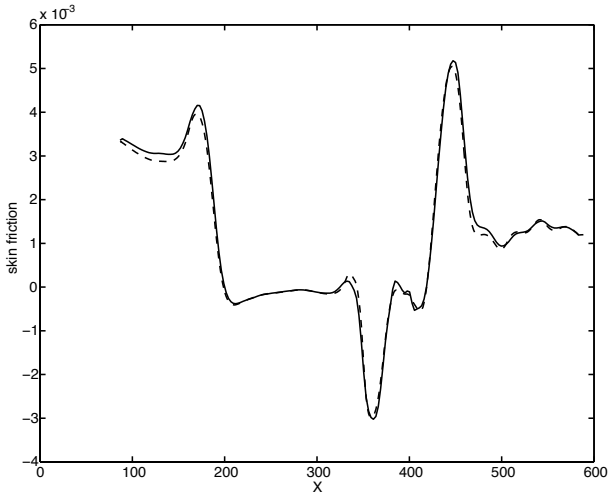


Figure 6-5: Mean skin friction for the explicit method. The solid line represents the skin friction on the original grid (193×65), the dashed line on the fine grid (385×129).

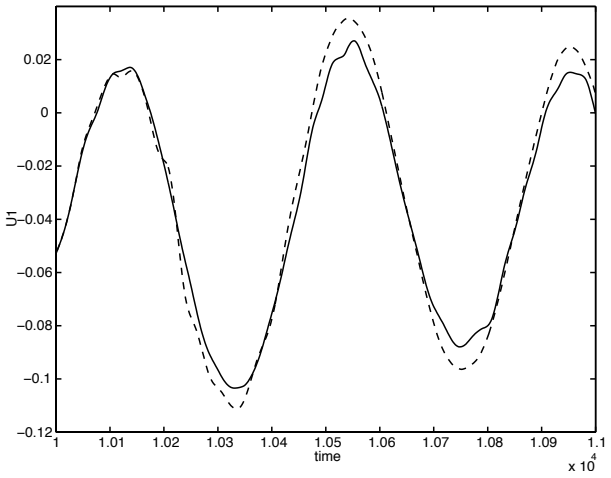


Figure 6-6: Instantaneous solution U1 obtained with the explicit time integration for the original grid (solid line) and the fine grid (dashed line).

the explicit reference solution for various different types of flow quantities, i.e. mean, r.m.s. and instantaneous. Also, we investigate and formulate a proper stopping criterion for the iteration process used to determine the steady state

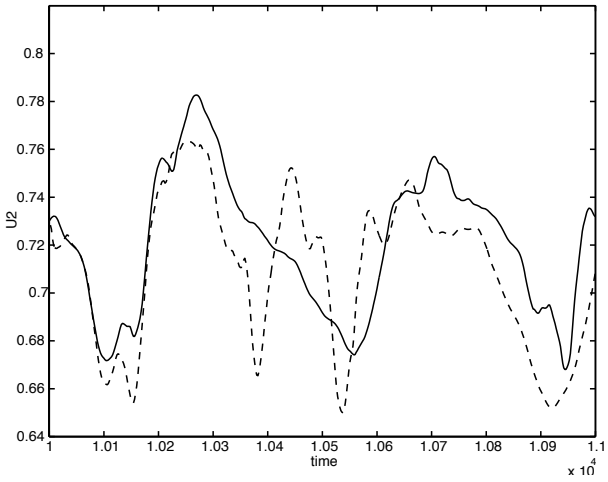


Figure 6-7: Instantaneous solution U2 obtained with the explicit time integration for the original grid (solid line), and the fine grid (dashed line).

solution in pseudo time. Moreover, we define a set of criteria which we require to be satisfied by the global error caused by the time integration and which relate the time step, the local accuracy and the relaxation method. In practice one would also be interested in the computational effort of the implicit method compared to the explicit method. It turns out that the basic implementation of our implicit method is quite competitive with the explicit code with respect to the required CPU time for relevant choices of the numerical parameters.

6.4.1 Time integration and implicit approximation of the flux

In chapter 4 the goal was to determine a steady state solution and it sufficed to use the first order Euler backward scheme. Here we simulate an unsteady flow and we want to determine the largest possible time step such that the solution is resolved with acceptable accuracy. Of course, it is possible to use a first order scheme for an unsteady simulation as well but it is our experience that in that case the time step has to be chosen much smaller than the allowed accuracy time step associated with a higher order time integration scheme. Therefore we use the second order Crank-Nicolson scheme (2-40) in the rest of this chapter. Each time step a nonlinear set of equations has to be solved as a result of the spatial discretisation and temporal integration given by (2-42). As mentioned already in section 2.4.2 it may be necessary to introduce a pseudo time derivative in order to facilitate or enable the convergence towards the

solution of the nonlinear system (2-42) in every time step. The resulting set of algebraic equations for every pseudo time step is given in (2-44) and a more detailed description of the iteration process can be found in section 2.4.2.

Similar to the flux, we distinguish the inviscid from the viscous part of the flux Jacobi matrix. The inviscid part of the flux Jacobi matrix is approximated using a first order upwind approach as described in section 4.4 which results in a sparse matrix with 5 bands of 4×4-matrices. The viscous flux defined in section 6.2.2 uses a nine point stencil. In order to stay within the five point stencil used for the inviscid part of the numerical flux Jacobi matrix we allow another approach in which the cross derivatives are neglected. Consider a transformation from physical to computational space

$$(x, y) \longrightarrow (\xi, \eta) \tag{6-14}$$

Applying this transformation to e.g. the stress term σ_{xx} in (6-6) gives

$$\sigma_{xx} = \frac{4}{3}\frac{\mu}{Re}\left(\xi_x u_\xi + \eta_x u_\eta\right) - \frac{2}{3}\frac{\mu}{Re}\left(\xi_y v_\xi + \eta_y v_\eta\right) \tag{6-15}$$

where ξ_x etc. are geometric terms arising from the transformation of the derivatives. Switching to conservative variables and neglecting the terms that would create cross-derivatives of ξ and η after additional differentiation with respect to x in (6-1) yields

$$\sigma_{xx} \approx \frac{4}{3}\frac{\mu}{Re}\xi_x \left(\frac{\rho u}{\rho}\right)_\xi - \frac{2}{3}\frac{\mu}{Re}\xi_y \left(\frac{\rho v}{\rho}\right)_\xi \tag{6-16}$$

Linearisation around the conservative variables and old time level and summing the results over all four edges of the control volume gives the corresponding viscous contribution to the 4×4 flux Jacobi matrices. The total Jacobi matrix can be obtained by the summation of the inviscid and viscous blocks multiplied by a factor $\frac{1}{2}\Delta t$ arising from the time integration and the addition of the diagonal blocks $\frac{I}{\Delta\tau}$ due to the pseudo time iteration.

The numerical system that has to be solved each time step is defined in (2-44). A computationally more efficient method is possible if the left-handside of (2-44) is evaluated only once in pseudo-time. It appears that, for the present test case, the number of pseudo iterations necessary to obtain a desired accuracy is unchanged in that case [41]. This indicates that either the solution does not vary much during the pseudo iterations or that the extra error introduced by the approximation of the flux Jacobi matrix has no appreciable influence. Applying this approximation to the numerical system in (2-44) we get

$$\mathcal{A}\Delta v_{i,j} = g_{i,j} - H_{i,j}(v^k) \tag{6-17}$$

where the matrix $\mathcal{A}$ is fixed during one time step and H corresponds to the left-handside in (2-42). It is clear that the solution of (2-42) is obtained when (6-17) converges. The speedup factor obtained with this approximation compared to the case where the flux Jacobi matrix is updated every pseudo time step is about 2.3.

6.4.2 Implicit treatment of boundary conditions

All boundaries of the computational domain are treated implicitly. For the solid wall and the inflow boundary the implicit approach is straightforward. At the free-stream boundary the implicit treatment of the conservative quantities $\rho, \rho u$ and E is also straightforward. However, the implicit treatment of the buffer domain and the blowing and suction is less trivial and will be outlined next.

Buffer domain

The buffer technique described in section 6.2.3 was developed originally in conjunction with the explicit time integration scheme described in section 2.4.1. For this explicit scheme the buffer is applied after every stage in the Runge-Kutta scheme. As mentioned before, the solution in the buffer depends on the number of times that the buffer is applied. If we would apply the buffer explicitly in the implicit scheme it is not clear whether this should be done after every time step, after every pseudo time step or even within the solution process of the linear system. To circumvent the difficulties and arbitrariness arising from an explicit application of the buffer function in the implicit scheme we base the implicit treatment of the buffer on a generalization and redefinition of the total flux in the buffer domain. Applying the Euler forward scheme to (2-35) followed by the application of the buffer, results in a discretized equation. Taking the limit of Δt to zero of this set of algebraic equations yields an equivalent differential equation in which an additional term is added to the flux in (6-1) in the buffer domain. These steps are considered in more detail next. Define the buffer by

$$\xi = \begin{cases} 1 & \text{if } x < x_b \\ \widetilde{\zeta}(x) & \text{if } x \geq x_b \end{cases} \tag{6-18}$$

where x_b denotes the x-coordinate at the beginning of the buffer domain. If we would use the Euler forward method we can write the solution on the next

time step in two phases

$$\begin{array}{rcl} v^{n+1} & = & q^n_{i,j} - \Delta t f_{i,j}(q^n) \\ q^{n+1}_{i,j} & = & q^*_{i,j} + \xi(v^{n+1} - q^*_{i,j}) \\ & = & q^n_{i,j} - \Delta t \left\{ \frac{1}{\Delta t}(\xi - 1)(q^*_{i,j} - q^n_{i,j}) + \xi f_{i,j}(q^n) \right\} \\ & = & q^n_{i,j} - \Delta t \widetilde{f}_{i,j}(q^n) \end{array} \tag{6-19}$$

where $q^*_{i,j}$ is the reference solution and $\widetilde{f}_{i,j}(q)$ is defined by

$$\widetilde{f}_{i,j}(q) = \left\{ \frac{1}{\Delta t}(\xi - 1)(q^*_{i,j} - q_{i,j}) + \xi f_{i,j}(q) \right\} \tag{6-20}$$

Taking the limit Δt to zero in (6-19) shows that there exists a consistent differential equation for every region of the computational domain

$$\left\{ \begin{array}{lcll} \dfrac{dq}{dt} + f(q) & = & 0, & x < x_b \\ \dfrac{dq}{dt} + \alpha(q^* - q) + f(q) & = & 0, & x_b \le x < x_e \\ q = q^* & & & x = x_e \end{array} \right. \tag{6-21}$$

where x_e stands for the x-coordinate at the end of the computational domain and $\alpha = C_3 \log(\zeta)$ with ζ defined in section 6.2.3. Due to the construction of ζ the value of α goes to infinity near the outflow boundary. Therefore we do not treat (6-21) directly but use the formulation of the new flux in (6-20) instead. This constitutes no limitation on the overall accuracy since the buffer domain is only used to damp the reflections near the outflow boundary. The construction of the corresponding new Jacobian matrix is straightforward since the blocks in (4-13) corresponding to the buffer domain only have to be multiplied with ξ and a new diagonal contribution has to be added according to the new flux definition in (6-20).

Blowing and suction

Implicit treatment of the blowing and suction boundary is done in a way similar to the treatment of the buffer domain. The blowing and suction specifies the vertical velocity. Applying the Euler forward method for ρv gives

$$(\rho v)^{n+1}_{i,j^*} = v_i \rho^{n+1}_{i,j^*} = (\rho v)^n_{i,j^*} - \Delta t v_i f^{(\rho)}_{i,j^*}(q^n) \tag{6-22}$$

where v_i is the imposed velocity which is independent of time, j^* is the j-index at the upper boundary and the superscript ρ denotes the flux vector of the mass equation in (6-1).

6.4.3 Linear solver

At each pseudo time level a linear system has to be solved or approximately solved as a result of the discretisation in (2-44). To this purpose we use a symmetric Gauss-Seidel method which was described in more detail in section 4.4. Although the number of required iterations of the nonlinear system is influenced by the accuracy of the solution to the linear system to a certain degree, the required total CPU time for one time step is lowest at very few linear iterations. In this chapter we use only one iteration of the symmetric Gauss-Seidel solver. Since only very few iterations are necessary to obtain an optimal efficiency, more advanced numerical methods such as e.g. Krylov methods are not required.

6.5 Implicit time integration results

In this section we will present the numerical results obtained with the implicit time integration method defined in the previous section. The main focus will be on the relation between the global accuracy of the solution obtained with the implicit method and the magnitude of the time step. In case of flow around a pitching airfoil the investigation on the accuracy time step can be related to an external time scale e.g. the pitching frequency. Although we do not apply an external unsteady flow condition the flow becomes unsteady and another measure for the time accuracy is desirable. Therefore we relate the global error induced by the temporal integration to the spatial discretisation error. In the numerical time integration scheme defined previously three parameters remain that influence this global error: the time step Δt , the local accuracy with which the nonlinear system (2-42) is solved each time step, ϵ and the relaxation parameter $\Delta\tau$. Obviously one would like the global error to decrease if the solution is determined more accurately at each time step. In order to satisfy this quite general requirement it appears that the specific definition of the measure for the local accuracy and the associated stopping criterion play an important role. These elements will be discussed in more detail later on.

6.5.1 Error bounds

In order to determine whether a certain time step is acceptable or not we need a measure for the accuracy of a certain flow quantity. To this purpose we introduce $\mathcal{E}$ which represents a measure for the global error over the sampling period T as described in section 6.3. In a moment we will define norms to specify $\mathcal{E}$ in more detail for different types of quantities. It appears quite

natural to require that over the total sampling period

$$\mathcal{E}(t) < \beta \mathcal{E}(\Delta x), \quad 0 \leq \beta < 1, \tag{6-23}$$

which states that the global error due to the time integration should be smaller than the global error due to the spatial discretisation. In this way the total error remains of the same order as the inherent spatial discretisation error. The value of $\mathcal{E}(\Delta x)$ depends on the flow quantity. If the error $\mathcal{E}(\Delta x)$ is large for a certain flow quantity compared to errors in other quantities this quantity is not well resolved on the reference grid. A smaller value of β leads to a more conservative upper bound for the temporal error. However, no general value for β can be given. The only practical requirement is that it should be small enough to ensure that the time integration error remains small compared to the spatial discretisation error for various quantities. In this chapter we take $\beta = 0.1$. In the following, we first focus on the global error due to the time integration after which we discuss the determination of the global error due to the spatial discretisation.

A measure for the spatial discretisation error

The next step is to determine appropriate norms for the errors defined in (6-23). Depending on the flow quantities that one considers more or less strict limits on Δt_{acc} may have to be set. Therefore, we look at mean, r.m.s. and instantaneous flow quantities. The numerical results for the explicit run on the original grid are defined as the reference solutions. We define the following general formula for the norm of the error of the averaged and r.m.s. quantities

$$\mathcal{E}(\psi) \quad = \quad \frac{1}{\psi_{norm}} \left\{ \frac{1}{L_x L_y} \int_0^{L_x} \int_0^{L_y} (\psi - \psi_{ref})^2 \, dxdy \right\}^{\frac{1}{2}} \tag{6-24}$$

where L_x and L_y are the lengths of the computational domain, ψ is a general notation for the quantity that one wants to observe, the subscript ref stands for reference value obtained with the explicit solver and the subscript $norm$ stands for a normalization value. For the instantaneous quantities we define

$$\mathcal{E}(\psi) \quad = \quad \frac{1}{\psi_{norm}} \left\{ \frac{1}{T} \int_0^{T} (\psi - \psi_{ref})^2 \, dt \right\}^{\frac{1}{2}} \tag{6-25}$$

where T is the total sampling time. In this chapter we will use the skin friction, the r.m.s. of U and the instantaneous velocity U measured at two different locations in the computational domain to monitor the accuracy of the simulation. Other quantities have been considered as well but do not lead to different conclusions and will not be incorporated in this thesis.

In order to determine the norms related to the global spatial discretisation error we perform a simulation on a refined grid with the explicit method. We use a fourth order interpolation method to obtain the fine grid with 385×129 points from the reference grid with 193×65 points. The initial condition at $t = 10,000$ and the Blasius solution at the inflow and outflow boundary are obtained with the same interpolation method. In figure 6-5 we have plotted the mean skin friction coefficient corresponding to the sampling period T for the explicit method on the original and the fine grid. The results are in good agreement, which indicates that the mean quantities are sufficiently well resolved on the original grid. In figures 6-6 and 6-7 we have plotted the velocity U at two different locations in the computational domain. In the smooth region ($U1$) of the flow the results on the original and fine grid are in good agreement, the main difference is in the amplitude, whereas the phase corresponds well. For the velocity $U2$ only the large structures seem to agree. On the fine grid additional higher frequencies in time are resolved. Considering the results for $U2$, the reference grid appears too coarse to resolve all instantaneous quantities. However, calculations for large time steps with the implicit method on the reference grid in the next section show very good agreement with the reference solution for $U2$.

Requirements on global error

Here we study the effect of the magnitude of the time step on the global norms $\mathcal{E}$ introduced in the previous section. The global error due to the time stepping method does not only depend on the magnitude of the time step but also on the local tolerance level ϵ which is a measure of the local accuracy of the solution to the nonlinear system in (2-44) for each time step. Also the relaxation parameter $\Delta\tau$ may have a considerable effect, although it only represents a way to obtain or accelerate convergence of the iterative process in (2-44). Therefore the global error $\mathcal{E}$ on a given grid now depends on three parameters: the time step Δt, the local tolerance level ϵ and the relaxation parameter $\Delta\tau$. The obvious requirement that the global error should decrease if the solution is determined more accurately at each time step, which can be achieved e.g. by decreasing the time step or using a higher local accuracy at each time step, is more precisely formulated as:

$$\begin{array}{lll} 1) & \mathcal{E}(\Delta t_1, \epsilon, \Delta\tau) \lesssim \mathcal{E}(\Delta t_2, \epsilon, \Delta\tau), & \Delta t_1 \leq \Delta t_2, \quad \forall \Delta\tau \\ 2) & \mathcal{E}(\Delta t, \epsilon_1, \Delta\tau) \lesssim \mathcal{E}(\Delta t, \epsilon_2, \Delta\tau), & \epsilon_1 \leq \epsilon_2, \quad \forall \Delta\tau \\ 3) & \mathcal{E}(\Delta t, \epsilon, \Delta\tau_1) \approx \mathcal{E}(\Delta t, \epsilon, \Delta\tau_2), & \Delta\tau_1 \neq \Delta\tau_2, \end{array} \tag{6-26}$$

Requirement one states that the global error $\mathcal{E}$ should decrease if the time step decreases. Secondly, the global error should not increase if one puts

more effort into solving the nonlinear system each time step. This is stated in requirement two. Requirement three is somewhat less transparent. Since we do not solve system (2-44) to machine accuracy the solution may still depend on the relaxation method. For our specific method the relaxation parameter is $\Delta\tau$, but one can also think of other relaxation methods like e.g. multigrid. Sensitivity on the relaxation method should be small. The requirements on the global error $\mathcal{E}$ in (6-26) need the specification of a measure for the local accuracy. However, the definition of the measure for the local accuracy, which we denote by $\phi(v)$, is not unique. The specific definition of $\phi(v)$, which yields the stopping criterion $\phi(v) < \epsilon$, appears to be a key factor in order to satisfy the obvious requirement that the global error should decrease if the solution is determined more accurately at each time step. Although the global error demands in (6-26) seem very reasonable, it is not trivial to find a local measure $\phi(v)$ such that all three requirements are satisfied. With

$$\phi(v^k) \quad = \quad \frac{\|\Delta v^k\|_2}{\|\Delta v^1\|_2} \tag{6-27}$$

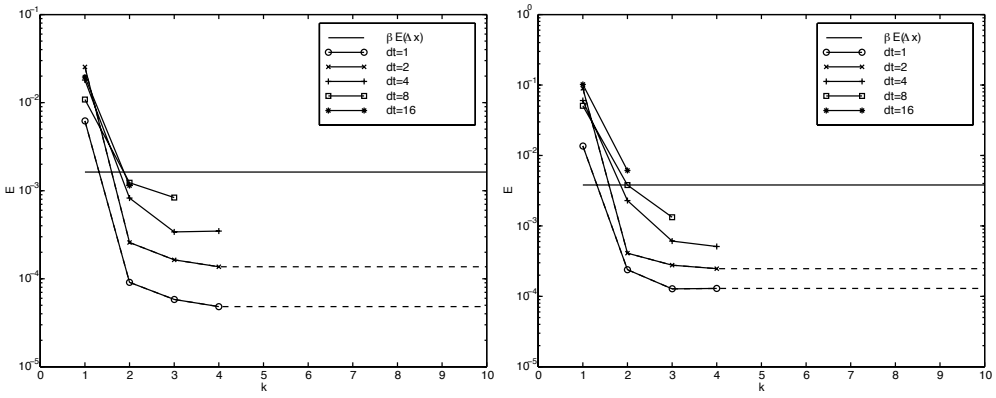
where the superscript k denotes for the pseudo time level, computations indicate that requirements one and two of (6-26) are satisfied. Also, one can verify that in the limit for $\Delta\tau \to 0$ and $\Delta\tau \to \infty$ the third requirement is satisfied.

6.5.2 Comparison with explicit results

In this section we perform numerical simulations with the implicit scheme for various time steps and local accuracies. First we illustrate the dependence of the global temporal error on the time step and local accuracy after which we elaborate on the optimal time step for mean and instantaneous flow quantities.

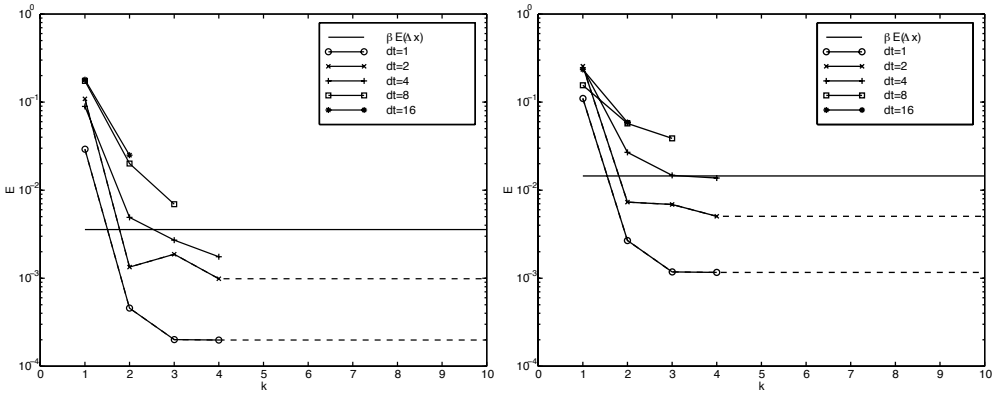
Global error versus time step and local accuracy

The time step for the explicit reference run is fixed at $\Delta t = 0.2$ which is slightly lower than the stability time step defined in section 6.2.4. For the simulations with the implicit time integration scheme we perform calculations for a range of Δt values. During simulation the time step is kept constant. In figure 6-8 the norms defined in (6-24) and (6-25) are plotted for various quantities and different time steps as a function of the local accuracy $\epsilon = 10^{-k}$. The norms are plotted for three types of quantities: mean, r.m.s. and instantaneous. At fore hand one would expect that quantities which vary more rapidly are also more sensitive to accumulated errors. This is illustrated by the results that the mean skin friction has the lowest global error for every Δt and ϵ considered, whereas the instantaneous quantities $U1$ and $U2$ have the largest global errors.



(a) Mean skin friction

(b) R.m.s. U



(c) Velocity $U1$

(d) Velocity $U2$

Figure 6-8: Global error for different quantities as a function of the local error $\epsilon = 10^{-k}$ for various time steps.

Given the required accuracy stated in (6-23) i.e. $\mathcal{E}(\Delta x)/10$ which is shown as a solid horizontal line, the figures indicate that for almost all quantities and time steps $\epsilon = 10^{-1}$ is too large. This is further illustrated in figure 6-9 where the instantaneous solution $U1$ is plotted for $\Delta t = 2$ and $\epsilon = 10^{-1}$. Although the main trend in time is captured correctly, some additional incorrect frequencies are introduced which clearly indicates that ϵ is too large. This correlates well with the fact that for $\epsilon = 10^{-1}$ the global error norms do not satisfy requirement one in (6-26). For the case $\epsilon = 10^{-2}$ almost excellent agreement

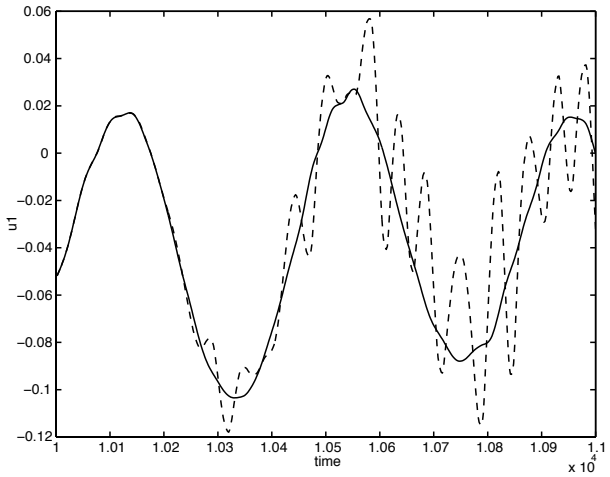


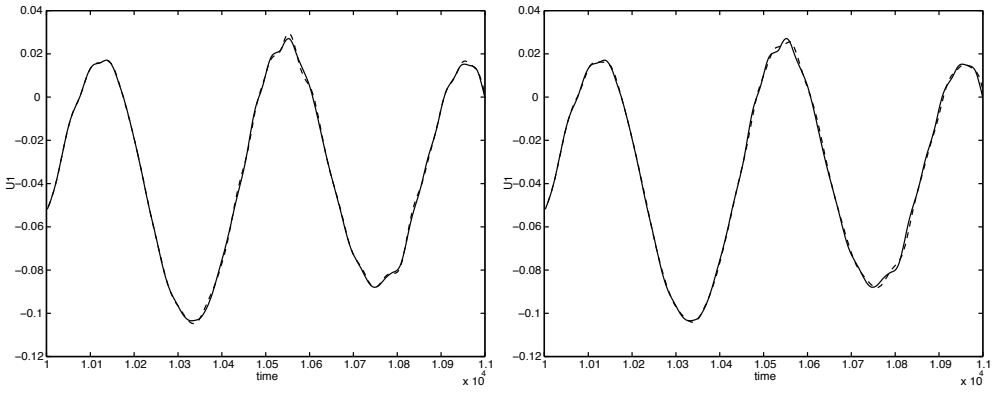
Figure 6-9: Instantaneous solution $U1$ as a function of time for the implicit scheme with $\Delta t = 2$ and $\epsilon = 10^{-1}$ (dashed) and the explicit reference solution (solid).

is obtained, which agrees with the significant drop in the norm shown in figure 6-8. For all smaller ϵ the three requirements in (6-26) are satisfied. The dashed lines in figure 6-8 correspond to a solution which would be obtained as $\epsilon \to 0$. Although the local accuracy is decreased significantly, the global error reaches an asymptotic value which indicates that the solutions converge quite rapidly as a function of ϵ and only contain the effect of the truncation error of the time integration. For larger Δt it is not possible to obtain a converged solution for arbitrary ϵ each time step. The threshold values of ϵ for $\Delta t = 4, \Delta t = 8$ and $\Delta t = 16$ are in the order of $\epsilon = 10^{-4}, \epsilon = 10^{-3}$ and $\epsilon = 10^{-2}$ respectively. The convergence does not necessarily break down in the first time step for each Δt and ϵ pair but may occur somewhere within the sampling period. This indicates that the convergence break-down depends not only on the time step and stopping criterion but also on the initial condition and accumulated effects. The convergence problem appears to be related to the pseudo time stepping method which is discussed in more detail in section 6.6.

Optimal time step for mean quantities

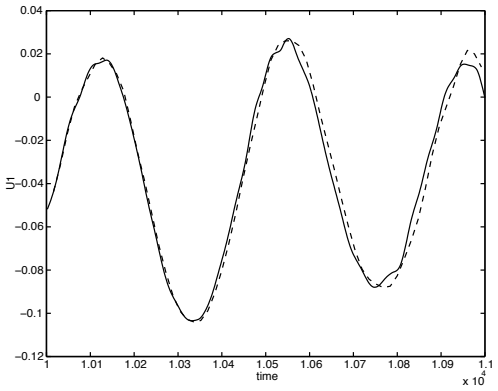
Having categorized the global error as a function of Δt and ϵ we now turn to the optimal choice of Δt. The horizontal lines in figure 6-8 represent the error bounds defined in (6-23) with $\beta = 0.1$. For the skin friction figure 6-8 shows that for all time steps the choice $\epsilon = 10^{-2}$ results in an error smaller

than 0.1%. We have also monitored other mean quantities like e.g. the mean Mach field or the mean pressure on the solid wall. The global errors for these quantities are even smaller than the global error for the skin friction, which is a spatial derivative of a mean quantity. So, for mean quantities it seems that the implicit time step can be chosen about eighty times larger than the explicit stability time step for this specific problem.

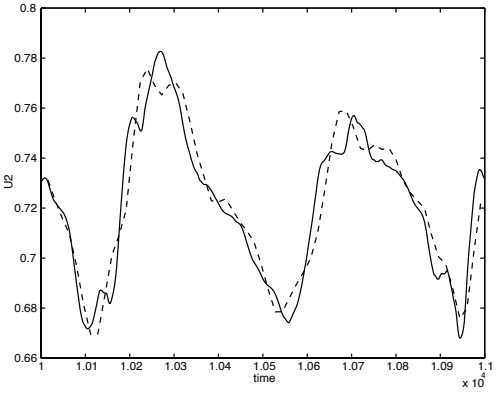


(a) $U1$ with $\Delta t = 4$ and $\epsilon = 10^{-2}$

(b) $U1$ with $\Delta t = 8$ and $\epsilon = 10^{-3}$



(c) $U1$ with $\Delta t = 16$ and $\epsilon = 10^{-2}$



(d) $U2$ with $\Delta t = 16$ and $\epsilon = 10^{-2}$

Figure 6-10: Instantaneous solution $U1$ and $U2$ as a function of time for the implicit scheme for various time steps and local accuracies (dashed) and the explicit reference solution (solid).

Optimal time step for instantaneous quantities

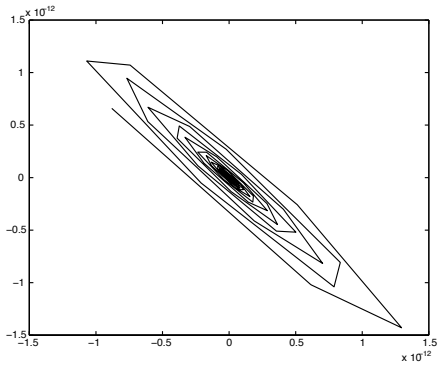
It was remarked before that no general value for β can be given at fore hand. We proposed a value of β that yields a conservative estimate for the allowable global error in time. However, for some quantities the choice $\beta = 0.1$ may be too restrictive. Consider for instance the quantity $U1$ at time step $\Delta t = 4$ and $\epsilon = 10^{-2}$. In figure 6-8 the error lies slightly above the error threshold. However, in figure 6-10 the solutions almost coincide which indicates that this value of β is too small for this specific quantity. Additionally, in figure 6-10 we show the quantities $U1$ and $U2$ for the time steps $\Delta t = 8$ and $\Delta t = 16$ for the highest attainable accuracy in these cases. For smaller Δt all quantities are resolved very well which can be concluded from the values of the norms in figure 6-8. The figures show that for $\Delta t = 8$ the quantity $U1$ is resolved quite well. For $\Delta t = 16$ the solution starts to deviate more but is still quite similar to the explicit solution. The difference between the explicit solution and the implicit solutions is larger for the quantity $U2$. At $\Delta t = 8$ the solutions start to deviate and even more for $\Delta t = 16$ where the main trend is still captured but the deviation from the reference solution has increased considerably. Hence the global error criterion (6-23) at $\beta = 0.1$ represents a robust and conservative approach even for instantaneous quantities. To be on the safe side, a choice of $\beta \leq 0.1$ for the quantities $U1$ and $U2$ is recommended which means that $\Delta t = 8$ and $\Delta t = 16$ are too large and $\Delta t = 4$ is close to Δt_{acc} for these quantities. For the instantaneous quantities the accuracy time step is therefore a factor 4 smaller than for the mean quantities. However, compared to the stability time step for the explicit method this is still about a factor 20 larger.

6.6 Dynamical behavior for large time steps

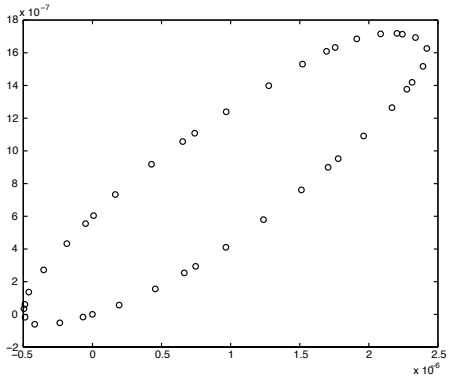
In the previous subsection it was noted that for large time steps it is not possible to obtain a solution over the total sampling period for arbitrary small ϵ, see figure 6-8. The convergence does not necessarily break-down in the first time step but may occur somewhere within the sampling period.

To study the effect of the time step on the convergence behavior we perform one time step for a range of Δt values with the initial condition taken as the solution obtained at $t = 10,000$ and the CFL_τ number for the pseudo time step set to 1.0, which is slightly smaller than the CFL number used for the explicit reference simulation. For time steps up to $\Delta t = 6$ no convergence problems are encountered and arbitrary accuracy can be obtained.

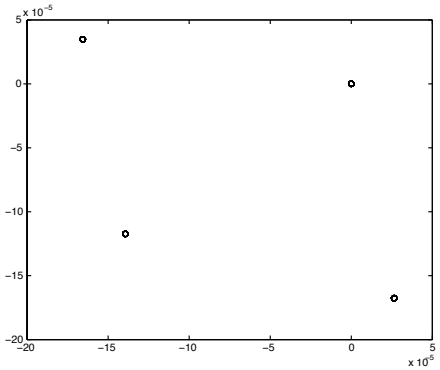
The convergence behavior of the numerical system (6-17) in pseudo time can be visualized with phase portraits by plotting the values of two typical quantities at every pseudo time level. The location of these quantities is determined by the value of Δv in (6-17) at a certain moment in pseudo time.



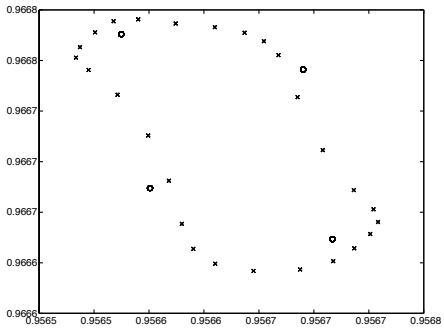
(a) $\Delta t = 6$ with $CFL_\tau = 1$



(b) $\Delta t = 8$ with $CFL_\tau = 1$



(c) $\Delta t = 16.5$ with $CFL_\tau = 1$



(d) $\Delta t = 16.5$ with $CFL_\tau = 1(o)$ and $CFL_\tau = 1.2(\times)$

Figure 6-11: Phase portrait of two typical values of ρ for various time steps and pseudo time steps.

The value of Δv should converge to zero and therefore two locations are taken where the absolute value of Δv is maximal. In figure 6-11 we have plotted such phase portraits in pseudo time using two typical values of the density for $\Delta t = 6, \Delta t = 8$ and $\Delta t = 16.5$. In order to visualize the behavior, the density is adjusted by subtracting its value at iteration level $k = 1000$ if necessary. The dynamical structure for these three time steps is quite different. In case $\Delta t = 6$ the dynamic convergence behavior resembles that of a converging spiral associated with a stable steady solution of (2-43). Compared to the

convergence behavior at smaller time steps, which are all straight lines, a structural difference of the dynamics around the fixed point already occurs even though the fixed point of (2-42) is still an attractor and the proper solution is obtained. For $\Delta t = 8$ the convergence of the pseudo time iterations stalls and no fixed point solution is obtained but there appears to exist some kind of limit cycle. To determine whether the limit cycle is an attractor we take a new initial condition such that the values of the observed densities lie near the center of the cycle. To obtain this initial condition we start with the solution at $t = 10,000$ and perform a number of pseudo iterations for which we determine the average of all flow quantities. The value of ϕ in the center is about two decades lower than its value on the limit cycle, indicating the existence of a true root of (2-42). However, after restarting the pseudo time iteration the solution returns to the limit cycle, which indicates that the limit cycle is indeed an attractor. If we start with $CFL_\tau = 1.0$ and switch to $CFL_\tau = 10^{-1}$ after a certain number of pseudo iterations the solution does converge to the required solution of (2-42). However, if we use this small CFL_τ number during the total pseudo time iteration the solution does not converge which indicates that the convergence depends sensitively on the initial condition in a rather complex way to which we return in a moment.

For $\Delta t = 16.5$ an exact period four solution is obtained. There seems to be a connection with classical chaos theory where e.g. a fixed point solution may become unstable and bifurcate into a period two solution which again may bifurcate into a period four solution etc, as a function of some bifurcation parameter. In our case we first investigate whether Δt is a proper bifurcation parameter and more systematically study the various types of possible convergence stall. To examine the long τ behavior we define the winding number by

$$\Gamma = \frac{1}{2\pi(N+1-k^*)} \sum_{k=k^*}^{N} \psi_k \tag{6-28}$$

where $k^* (\gg 1)$ is an iteration level at which the solution, roughly speaking, lies on the attractor, N $(\gg k^*)$ is a sufficiently large iteration level and ψ_k is the angle between two consecutive points in the phase portrait with respect to the center of the limit cycle. For a range of Δt between $\Delta t = 6$ and $\Delta t = 100$, we observe a complicated behavior and $0.21 \leq \Gamma \leq 0.3$. For some Δt we find almost exactly $\Gamma = 0.25$ which corresponds to a period four solution as in the case of $\Delta t = 16.5$. However, no period two solution ($\Gamma = 0.5$) is found, which indicates that no direct link to classical chaos theory is apparent or that Δt is not the proper bifurcation parameter.

In the work of Yee and Sweby [100, 101] a study is made of the asymptotic behavior of time integration methods for steady state problems which shares

a number of features with our results. It is shown that depending on the time integration method spurious solutions may occur as a function of the time step. A spurious solution is defined as a steady or periodic solution of the numerical system but not of the underlying differential equation. Additionally, they show that bifurcations can occur for both the real and spurious solutions and complex basins of attraction exist as a function of the time step. Depending on the initial solution and time step the iteration process may converge to a spurious or real solution.

In the present context we solve equation (2-42) to obtain the solution at the next time step. By adding a pseudo time derivative to this set of nonlinear equations, the required solution is the steady state solution of (2-43). So, we use a time integration scheme to obtain a steady state which connects the present work to the work of Yee and Sweby if we consider the pseudo time step as the bifurcation parameter as described in [100, 101]. To confirm this we repeat the simulation for $\Delta t = 16.5$ (period four solution) with $CFL_\tau = 1.2$. In figure 6-11 it is shown that for $CFL_\tau = 1.2$ the iteration process does not converge to the period four solution but to a higher periodic orbit which resembles the bifurcation path to chaos as in Yee and Sweby. The sensitivity on the CFL_τ number observed for the case of $\Delta t = 8$ may now be explained by the presence of the basins of attraction for the spurious and real solutions. Yee and Sweby observe that for certain time discretisations and fixed point solutions of a differential equation fragmented basins of attraction exist. This agrees with the results for the simulations at $\Delta t = 8$. If we start with the same initial condition, the iteration process does not converge to the fixed point solution for both $CFL_\tau = 1.0$ and $CFL_\tau = 10^{-1}$. However, when switching to $CFL_\tau = 10^{-1}$ after a certain number of iterations at $CFL_\tau = 1.0$ we do find the fixed point solution. This shows that the basin of attraction for the true fixed point solution has a complex structure. Finally, we note that for $\Delta t = 1$ and $\Delta t = 2$ we used $CFL_\tau = 1.0$ for $\epsilon \leq 10^{-3}$ which agrees with a remark of Yee and Sweby in [100] where they state that for practical computations the scheme has a higher chance of obtaining the correct physical solution if one uses a $\Delta\tau$ restriction of the same order as the stability limit of an explicit method.

Additionally we examined the influence of the temporal integration of (2-35) on the convergence problems described above and adopted the Euler backward scheme which is dissipative in contrast to the Crank-Nicolson scheme. For large time steps the same convergence problems occurred which indicates that this phenomenon is not caused by the specific properties of the Crank-Nicolson scheme but is quite likely to be more general.

6.7 Conclusions

In this chapter we have performed a DNS of a complex two dimensional unsteady flow over a flat plate with an explicit and an implicit time integration scheme. An explicit four stage compact storage Runge-Kutta method was used as a reference method. For the implicit time integration scheme we used the second order Crank-Nicolson scheme which results in a large set of coupled nonlinear algebraic equations that have to be solved each time step.

The main disadvantage of explicit schemes arises from the fact that the time step is bounded due to numerical stability requirements. A-stable implicit schemes such as Crank-Nicolson, do not exhibit this problem. Consequently, the time step is now bounded for accuracy reasons alone. As an upper bound for the global error over the sampling period due to the time integration scheme we used an estimate for the global spatial discretisation error. The spatial discretisation error is determined through comparing simulations on the original grid and a refined grid. Given the upper bound of the global error we formulated requirements for the global time integration error which involve the time step, local accuracy of each time step and the relaxation method such that the global error decreases if the local solution is determined more accurately each time step. It turned out that the choice of a proper function to measure the local accuracy plays a significant role in order to fulfill these requirements. Computations showed that with suitable numerical parameters the requirements are obeyed and proper solutions of the flow problem are obtained. The corresponding accuracy time step depends on the flow quantity that one wishes to resolve. Therefore we find that the global spatial discretisation error seems to be a good measure of reference for the allowed temporal error.

For instantaneous quantities it is possible to choose a time step about twenty times larger than the stability time step for the explicit scheme, while for mean quantities even a factor of eighty is attainable. This shows that the stability time step is too restrictive with respect to accuracy and a considerable speedup is in principle possible using implicit methods. With respect to the CPU time we observe that depending on the flow quantity we were able to achieve a speedup factor of 9.2 and 3.7 for mean quantities and instantaneous flow quantities respectively without any code optimization.

Convergence problems were encountered for large time steps. Yee and Sweby [100, 101] have shown that the steady state solution can bifurcate into chaos if a time integration scheme is used to obtain the steady state. In our case this corresponds to the pseudo time integration that we used to solve the nonlinear system of equations to obtain the solution at the next time step. Numerical simulations confirm the sensitive dependence on the pseudo time step.

Chapter 7

Analysis of multigrid performance for unsteady flow

7.1 Introduction

The discussion on the magnitude of the time step in chapter 6 constitutes a first step in achieving a criterion to dynamically determine the accuracy time step during a simulation. It appeared that for a relevant choice of the time step the implicit method could not outperform the explicit method with respect to CPU time because a lot of pseudo time steps were required to achieve a sufficient convergence level. Therefore, in this chapter we analyze the convergence acceleration properties of the multigrid method described in section 2.6 applied to the unsteady shock boundary-layer interaction flow.

Originally, multigrid was developed in order to accelerate the convergence of elliptic problems [13, 38]. Later, multigrid was also used to accelerate the convergence towards the steady state of hyperbolic systems which are typically present in aerodynamic applications. Although multigrid has been quite successful in these areas the "text-book" [14] acceleration of multigrid as when applied to elliptic problems is not found. A typical example can be found in Ref. [52] where a speedup factor of about ten was found. In chapter 5, however, only a modest acceleration was obtained which appeared to be related to the specific choice of the spatial discretisation.

Gradually multigrid has also found its way in unsteady flow applications, as introduced by Jameson in [48] by means of a pseudo time stepping approach as described in section 2.4.2. Often a time stepping scheme is used as a smoother ([97]) in the multigrid process. This would limit its application to compressible flows because for incompressible flows the continuity equation does not contain a time derivative. A solution to this problem is the introduction of artificial compressibility methods as proposed by Chorin in Ref. [23] and later improved

by Turkel in Ref. [87] in which a pseudo time derivative of the pressure is added to the continuity equation. This approach has successfully been adopted in e.g. [10, 58]. In both cases the second order BDF2 scheme was incorporated to advance the solution in physical time whereas an explicit Runge-Kutta scheme was used as a smoother in pseudo time. Depending on the flow application typically 50-300 V-cycles per time step were necessary for a decrease of the residual of about 3-4 decades. However, no details about the convergence behavior (which may depend on the magnitude of the time step), optimal number of grids etc. were presented.

For compressible flows quite some research has been performed on the use of multigrid for unsteady flows (see e.g. Refs. [2, 3, 25, 59, 69, 75]). In Refs. [2, 3] BDF2 is used for the temporal integration and a Runge-Kutta scheme is applied as a smoother in pseudo time. In these references two important elements arise. First the magnitude of the pseudo time step is not only limited by stability requirements due to the explicit time marching in pseudo time but also by the number of grids used in the multigrid process and the magnitude of the physical time step. In this chapter we use the implicit Euler backward scheme in pseudo time and thus no stability requirements have to be met. However, the magnitude of the pseudo time step may influence the convergence as was observed in chapter 6. Second, the simulations in Ref. [3] show that no multigrid acceleration is obtained if the physical time step is too small. In addition to this observation which relates to the total CPU time we show in this chapter that for unsteady flow problems good smoothing properties required for a multigrid acceleration can only be obtained if the pseudo time step and the physical time step obey certain criteria.

Finally, we mention the results presented in Ref. [25] in which the convergence properties of multigrid for an explicit Runge-Kutta scheme and the implicit Euler backward scheme in combination with the LU-SGS method in pseudo time are studied for various unsteady flow problems. It is found that for all cases the implicit multigrid scheme outperforms the explicit multigrid scheme. This is in agreement with our findings in chapter 5 where the accelerated implicit method (LU-SGS) outperforms the accelerated explicit method as well.

In this chapter we analyze the multigrid performance in relation to the numerical method described in chapter 6 for a relevant choice of the numerical parameters. By means of a Fourier analysis we study the damping characteristics of our numerical scheme which reveals that high and low frequencies of the error are damped equally. We argue that this is caused by the additional terms in the Jacobi matrix related to the unsteadiness of the flow and thus clarifies the failure of multigrid to accelerate the convergence here.

The contents of this chapter is as follows. In section 7.2, a test case is

identified which is suitable for multigrid acceleration. This is followed by a presentation of numerical results in section 7.3. It appears that the order of magnitude of the time step and pseudo time step prohibits multigrid acceleration which is substantiated by a model equation study in section 7.4. Finally, the conclusions are summarized in section 7.5.

7.2 Identification of test case

The development of a dynamical time stepping method for unsteady flow simulations is a major motivation for much of the research in this work. This is, however, only useful if a competitive implicit method for large time steps can be found. In chapter 6 it appeared that for the test case considered $\Delta t = 4$ is the maximum time step for the accurate prediction of instantaneous flow quantities of the shock boundary-layer interaction flow.

In this section we discuss some convergence characteristics of the numerical scheme. We take the solution at $t = 10,000$ as an initial condition and perform one step for various magnitudes of the time step and pseudo time step. The

$\epsilon = 10^{-2}$		$\Delta t = 1$	$\Delta t = 2$	$\Delta t = 4$	$\Delta t = 8$	$\Delta t = 16$
	$\Delta\tau = 1$	35	35	35	36	39
	$\Delta\tau = 10$	7	7	8	10	14
	$\Delta\tau = 100$	4	5	6	8	14,*
	$\Delta\tau = 1000$	4	4	6	7	–
$\epsilon = 10^{-3}$						
	$\Delta\tau = 1$	51	52	52	54	–
	$\Delta\tau = 10$	10	12,*	13,*	16,*	–
	$\Delta\tau = 100$	9	11,*	12,*	–	–
	$\Delta\tau = 1000$	9	11,*	12,*	–	–

Table 7-1: The number of required pseudo time steps in order to obtain a local accuracy of $\epsilon = 10^{-2}$ or $\epsilon = 10^{-3}$ as a function of the time step and pseudo time step. The – indicates that no convergence was possible for the first time step and * indicates that convergence was not possible over the total sampling period T.

results are presented in table 7-1 for two relevant values of the local accuracy ϵ (see section 6.5.1). It appears that the required number of pseudo time steps is rather insensitive to changes in the time step but is influenced more by the pseudo time step.

For $\Delta t = 1$ and $\Delta t = 2$ it is sufficient to simulate with a local accuracy of $\epsilon = 10^{-2}$ in order to resolve all types of quantities with an acceptable accuracy

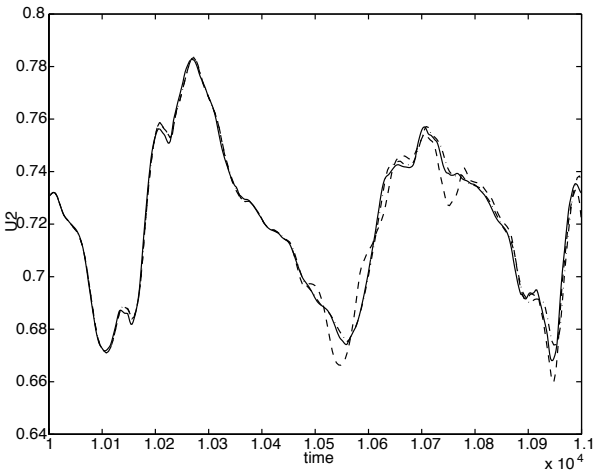


Figure 7-1: Instantaneous solution U2 as a function of time obtained with the implicit scheme for $\epsilon = 10^{-2}$ (dashed) and $\epsilon = 10^{-3}$ (dashed dotted) for $\Delta t = 4$ and the explicit reference solution (solid).

(see figure 6-8). It turns out that it is possible to simulate with infinitely large pseudo time steps during the complete sampling period in these cases which shows that no multigrid acceleration is needed since only 4 pseudo time steps are required each time step. For large time step such as $\Delta t = 16$ only mean quantities are resolved. Empirically we found that it is possible to use a pseudo time step of $\Delta\tau = 10$ during each time step and the speedup factor compared to the explicit method is already considerable. In section 6.5.2 it was shown that the optimal time step to resolve instantaneous flow quantities corresponded to $\Delta t = 4$. From figure 6-10 we see that for U1 a local accuracy of $\epsilon = 10^{-2}$ suffices. However, for the more sensitive quantity U2 (chapter 6) a local accuracy of 10^{-3} is necessary which is shown in figure 7-1. Simulations reveal that it is necessary to use a pseudo time step of $\Delta\tau = 1$ in order to obtain a converged solution during the total sampling period which means that approximately 52 pseudo time steps are required each time step. Therefore, the parameter setting $\Delta t = 4$, $\epsilon = 10^{-3}$ constitutes a good test case where multigrid acceleration is desired. We use this test case in the rest of this chapter.

7.3 Multigrid applied to unsteady flow

The use of multigrid was originally developed for elliptic problems and was later applied to steady hyperbolic problems as mentioned before. For these steady flow cases multigrid appeared to be very successful as well. The general consensus in literature is that if no "text book" acceleration is obtained there is something wrong e.g. in the implementation or choice of the boundary conditions, the intergrid operators or coarse grid operators [14]. Because an unsteady flow problem can be reformulated into a steady flow problem for each time step it is often assumed that multigrid will accelerate the convergence in pseudo time as well. In this chapter, however, we illustrate by means of a Fourier analysis that straightforward generalization of multigrid results for steady flow to unsteady flows is not always possible. On the basis of an order of magnitude analysis in section 7.3.1 we argue that no smoothing may be expected which is further substantiated by a model equation study in section 7.4.

The test case is described in section 7.2. The simulations are performed on the same grid as described in section 6.3. The multigrid process is described in section 2.6 and we use the same intergrid operators and coarse grid operator as defined in section 5.2. The next coarser grid is obtained by deleting every other grid point in both x- and y-direction. In the present setting we can alter the type of multigrid cycle, the number of grid levels, the magnitude of the pseudo time step and the number of pre- and post-relaxations as well as the number of coarse grid relaxations. A wide variety of settings of the multigrid parameters has been explored. Summarizing the results we conclude that within the current framework no multigrid acceleration could be obtained. In many cases the CPU time even increased compared to a single grid simulation.

In order to understand the fact that no multigrid acceleration is obtained we study the damping characteristics of the numerical scheme by means of a Fourier analysis which is explained next. For convenience we repeat our numerical system:

$$\left(\frac{I}{\Delta\tau} + I + \frac{1}{2}\Delta t\frac{\partial F}{\partial q}\right)\Delta q = g - F(q^k) \tag{7-1}$$

where F represents the numerical flux and the superscript denotes the pseudo time level. If q^∞ corresponds to the solution to (7-1) the error, w, in the solution at iteration level k can be defined as

$$w^k = q^k - q^\infty \tag{7-2}$$

For the present purpose it is not required to perform a complete Fourier decomposition of this error in two spatial dimensions. It is sufficient to use cross

sections that correspond to lines in the physical domain with constant x- or y-coordinate. If we assume periodic boundary conditions the error in e.g. the x-direction for constant y can be decomposed as

$$w^k(x) = \sum_{j=-N/2}^{j=N/2} c_j^k e^{I\phi_j} \tag{7-3}$$

where c_j^k denotes the j-th Fourier coefficient, $\phi_j = (\pi jx)/L$ and L represents the streamwise length of the computational domain. The damping characteristics of the numerical scheme in (7-1) can be studied by the amplification factor of subsequent Fourier coefficients i.e.

$$\frac{|c_j^{k+p}|}{|c_j^k|} \tag{7-4}$$

where $p \geq 1$. Good smoothing properties are obtained if high frequencies (large j) are damped faster than low frequencies (small j). By plotting the damping coefficient in (7-4) as a function of j a possible preference of the numerical scheme for damping high frequencies can be visualized. If this does not arise multigrid acceleration is not possible.

In the next section we illustrate that based on the damping properties of the numerical scheme no multigrid acceleration may indeed be expected for the current test case independent of the specific choice of the multigrid parameters.

7.3.1 Damping characteristics flat plate

It is common experience that multigrid performs better if a certain amount of numerical dissipation is added which can be achieved e.g. by using a more dissipative spatial discretisation. In order to simplify the analysis, in the following we use the first order Roe scheme for the spatial discretisation. If no multigrid acceleration can be obtained with this scheme then certainly for the MUSCL scheme no acceleration may be expected. Therefore we perform one time step with the Roe scheme on the original grid with $\Delta t = 4$ and $\Delta\tau = 1$. For q^∞ we take the solution which is obtained without multigrid with a local accuracy of $\epsilon = 10^{-12}$. In this way the error, w^k at pseudo time level k can be determined with (7-2). In figure 7-2 the error $w^k(x)$ in the density at the solid wall is plotted for three pseudo time levels. The errors are scaled by their range such that they can be presented in the same figure. Clearly, the initial error contains high as well as low frequencies. For good smoothing properties the high frequencies should vanish at subsequent pseudo time levels. However, as can be observed from figure 7-2 this is not the case and roughly a

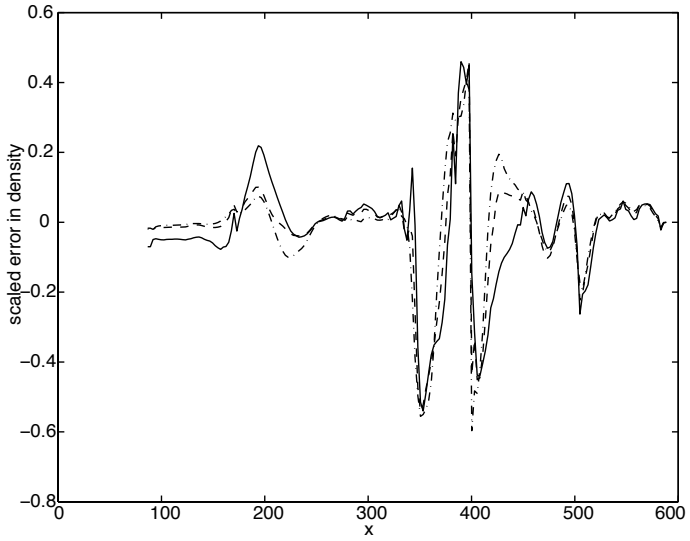


Figure 7-2: The scaled error $w^k(x)$ in the density at the solid wall for $k = 1$(solid), $k = 5$ (dashed) and $k = 10$ (dashed dotted). The errors are scaled by $0.01, 0.0081$ and 0.0045 respectively.

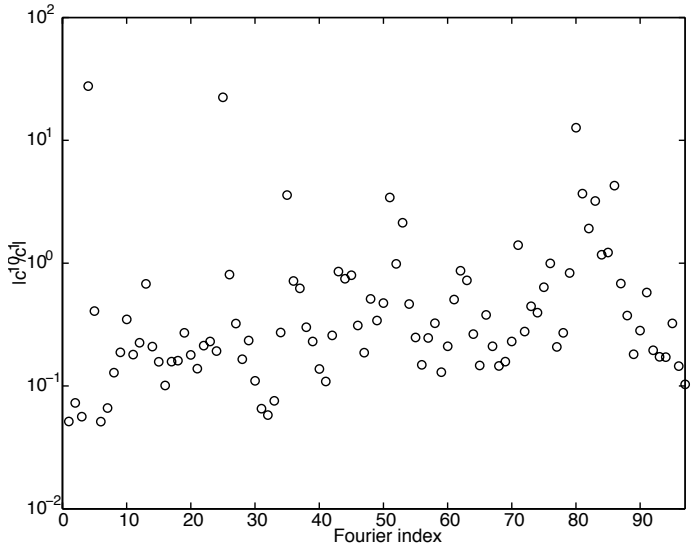


Figure 7-3: Damping coefficient c^{10}/c^1 for the error in the density at the solid wall.

uniform error reduction arises. Additionally, the damping coefficient c_j^{10}/c_j^1 is plotted in figure 7-3 which reveals that there is no damping preference for high frequencies and therefore no multigrid acceleration can be expected. Similar results are found for other cross sections parallel to the x-axis which indicates that there is no smoothing in x-direction. The smoothing properties of our numerical scheme in y-direction are studied next. In figure 7-4 the scaled error $w^k(y)$ is plotted for two different cross sections parallel to the y-axis for $x = 267$ and $x = 319.27$. If the two errors at the different cross sections are compared

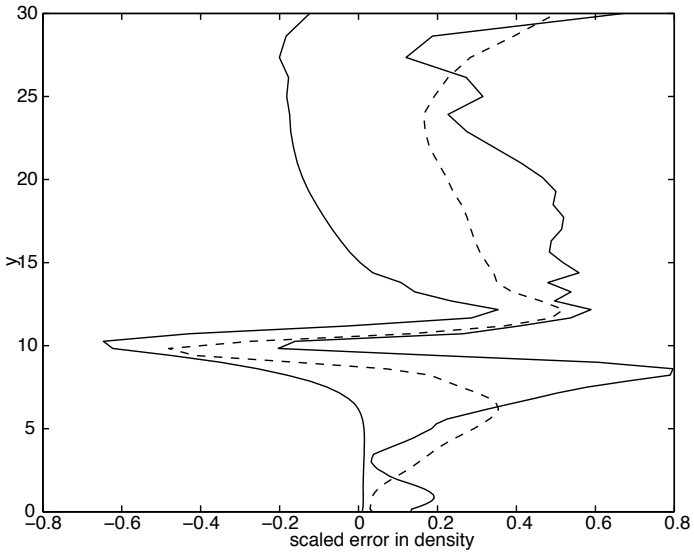


Figure 7-4: The scaled error $w^k(y)$ in the density for two cross sections parallel to the y-axis. The two solid lines represent the error at $k = 1$ the dashed line the error at $k = 10$ taken at the same cross section as the right solid line.

it is clear that the error at $x = 267.00$ is smoother than the error at $x = 319.27$ and no high frequencies seem to be present. Therefore for this particular error no further smoothing is necessary. However, for the error at $x = 319.27$ it is clear that the error profile at $k = 10$ contains less high frequencies which is also confirmed by a Fourier analysis. A smoothing analysis for other cross sections parallel to the y-axis does not alter our conclusions.

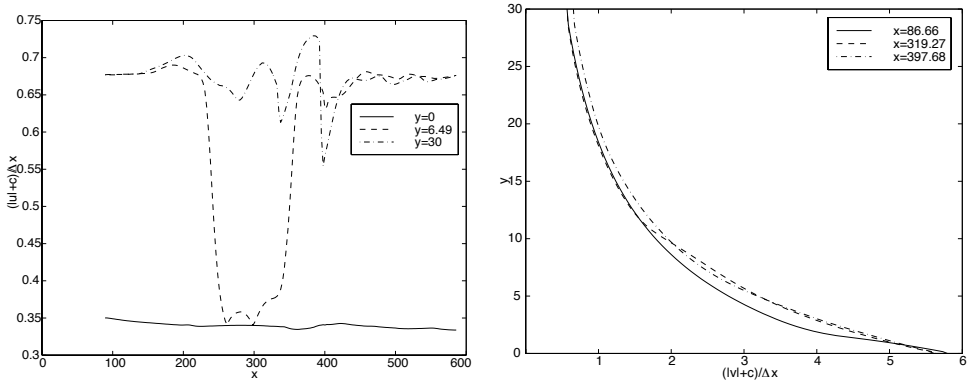
Although there is considerable smoothing in y-direction, the lack of smoothing in x-direction of our numerical scheme appears to be the bottleneck and explains the bad multigrid performance observed earlier. The peaks in the residual in x-direction are related to the shocks present in the flow. One might argue that this is the reason that no smoothing is obtained. However,

we observe that a Fourier-analysis of the error for the transonic inviscid flow in chapter 4 reveals that high frequencies are damped more efficiently than low frequencies which is in agreement with the fact that multigrid does accelerate the convergence towards the steady state (see chapter 5).

The preference in smoothing direction may be explained by a simple order of magnitude analysis. Consider the numerical scheme in (7-1). The flux Jacobi matrix can be written as

$$\frac{\partial F}{\partial q} = \frac{\partial F_x}{\partial q} + \frac{\partial F_y}{\partial q} \tag{7-5}$$

where F_x and F_y represent the flux in x- and y-direction respectively. In order to determine which term in 7-5 dominates the total flux Jacobi matrix we need a proper norm. The dependence of the smoothing on the spatial direction is related to the eigenvalues of the flux Jacobi matrix. Because the matrices are large an eigenvalue analysis is costly. However, a rough estimate is possible if we take into account that the building blocks of the flux Jacobi matrix are determined by the local eigenvalues of the flux Jacobi matrices of the Navier-Stokes equations as described in section 6.4. Because the present test case is convection dominated we approximate the eigenvalues by the maximum inviscid eigenvalues which in x- and y-direction are given by $(|u| + c)/\Delta x$ and $(|v| + c)/\Delta y$ respectively. In figure 7-5 these eigenvalues



(a) $(|u| + c)/\Delta x$ (b) $(|v| + c)/\Delta y$

Figure 7-5: Approximation of the inviscid eigenvalues for several cross sections in x- and y-direction.

are plotted for several cross sections parallel to the x- and y-axis. From the magnitude of the eigenvalues it follows that $\partial F_y/\partial q$ dominates the flux Jacobi

matrix in (7-5), especially in the region near the solid wall, which follows from the construction of the Jacobi matrix described in section 4.4. As observed earlier a sufficient spatial coupling (related to the superior performance for elliptic problems) must exist in order to obtain good smoothing properties. This may explain the fact that a certain amount of smoothing is obtained in y-direction whereas no smoothing at all is found in x-direction. However, next to the flux Jacobi matrix two additional terms are present in the smoothing operator (7-1) related to the implicit time integration in physical and pseudo time. These terms dominate the left-hand side of (7-1) if the time step and pseudo time step are relatively small. These results are in agreement with the results presented in table 7-1 where the required number of iterations for fixed CFL number in pseudo time was quite insensitive to changes in the magnitude of the physical time step which indicates that the term $I/\Delta\tau$ dominates the left-hand side of (7-1) especially for small $\Delta\tau$.

For large time steps and large pseudo time steps it is clear that the flux Jacobi matrix dominates the matrix in the left-hand side in (7-1) and better smoothing properties are expected. However, such simulations for this test case are not possible because of convergence problems related to the dynamical behavior as outlined in section 6.6 and because time-accuracy is desired. Therefore, in section 7.4 we consider a model equation and determine criteria for the magnitude of the time step and pseudo time step such that good smoothing properties are expected.

7.3.2 Changing the smoother

The previous results showed that for the present numerical method no smoothing is obtained. One may argue that we have used a bad smoother and multigrid acceleration can be obtained if more effort is put into finding a proper numerical method. In this section we show that significant changes in the numerical method do not change the smoothing properties which confirms our estimates about the desirable order of magnitude of the matrices in (7-1).

First we change the approximation of the flux Jacobi matrix and use the numerical Jacobi approximation as described in section 3.4.2. The simulation described above is repeated and no significant changes are observed. Similar to the previous case no smoothing is obtained.

Second we apply residual averaging to Δq in (7-1). Residual averaging is frequently used to increase the stability region of explicit time integration schemes see e.g. Ref. [27]. In this way larger time steps can be taken which improves the overall smoothing properties of the numerical scheme. The present residual averaging operator is given by

$$(1 - \xi\nabla_i\Delta_i)(1 - \xi\nabla_j\Delta_j)\Delta\widetilde{q}_{i,j} = \Delta q_{i,j} \tag{7-6}$$

where ξ is a constant, (i, j) denotes the coordinates in the computational domain and ∇ and Δ represent a forward and backward differencing operator respectively. The new solution in pseudo time is now updated by $\Delta\widetilde{q}$ instead of by Δq. We find that for $\xi = \frac{1}{2}$, the speedup factor on a single grid is about a factor 2 which is in agreement with results presented in [27]. However, the smoothing properties are not improved and no multigrid acceleration is obtained.

The above results once more indicate that the terms in (7-1) related to the time integration destroy the desired smoothing properties if Δt and $\Delta\tau$ are too small. In the next section a stronger relation between the eigenvalues of the flux Jacobi matrix, the time step and the pseudo time step will be derived for a model equation in order to analyze multigrid performance.

7.4 Model equation

In section 7.3.1 we showed that no multigrid acceleration can be expected for the present test case because low and high frequencies of the error in the solution during the pseudo time iteration were equally damped. Here we study the damping characteristics of our numerical scheme (see chapter 6) applied to the one-dimensional inviscid Burgers equation in (2-6). It is well-known that proper multigrid performance is only obtained if there is a strong coupling in space of the numerical operator which coincides with the fact that the multigrid algorithm was originally developed for elliptic operators [13, 38, 97]. The left-hand side of (7-1) is the smoothing operator where the partial derivative $\partial F/\partial q$ represent the coupling in space since F corresponds to the spatial flux. Also, one may argue that the terms in the smoothing operator related to the temporal integrations are multiples of the identity matrix which damps all Fourier-components equally. So, in order to obtain a sufficient spatial coupling the terms on the left-hand side of (7-1) must at least be of the same order and preferably $\partial F/\partial q$ should dominate. For explicit methods the stability time step is related to the eigenvalues of the flux Jacobi matrix (see e.g. section 4.2.3). Here we also relate the time steps in physical and pseudo time to the derivative of the flux function in a similar way. We will show that appropriate damping characteristics for multigrid acceleration are obtained if the CFL-numbers corresponding to the physical and pseudo time obey certain criteria.

Assume a uniform grid with grid spacing Δx on the interval $[0, 1]$ with

$N+1$ points ($i = 0 \ldots N$), define the minimum and maximum eigenvalue by

$$\begin{aligned} \widehat{q} &= \min_i \left|\frac{\partial F}{\partial q}\right| = \frac{1}{\Delta x} \min_i \frac{1}{2}|q_i + q_{i-1}| \\ \widetilde{q} &= \max_i \left|\frac{\partial F}{\partial q}\right| = \frac{1}{\Delta x} \max_i \frac{1}{2}|q_i + q_{i-1}| \end{aligned} \tag{7-7}$$

and define the physical and pseudo time step as in (2-37) by

$$\Delta t = \frac{\sigma_t \Delta x}{\max_i(q_i)} \approx \frac{\sigma_t}{\widetilde{q}} \quad \text{and} \quad \Delta\tau = \frac{\sigma_\tau \Delta x}{\max_i(q_i)} \approx \frac{\sigma_\tau}{\widetilde{q}} \tag{7-8}$$

where σ_t and σ_τ are the CFL-numbers in physical and pseudo time respectively. For the 1D Burgers equation the terms in the left-hand side of (7-1) are of the same order if

$$\begin{aligned} \frac{1}{2}\Delta t \frac{\partial F}{\partial q} &\gtrsim 1 \quad \Rightarrow \quad \sigma_t \gtrsim 2\xi \\ \text{and} \\ \frac{1}{2}\Delta t \frac{\partial F}{\partial q} &\gtrsim \frac{1}{\Delta\tau} \quad \Rightarrow \quad \sigma_t \sigma_\tau \gtrsim 2\xi\widetilde{q} \end{aligned} \tag{7-9}$$

with $\xi = \widetilde{q}/\widehat{q}$. Note that due to the construction in (7-1) the pseudo time step is dimensionless whereas σ_τ has dimension one over time. In figure 7-

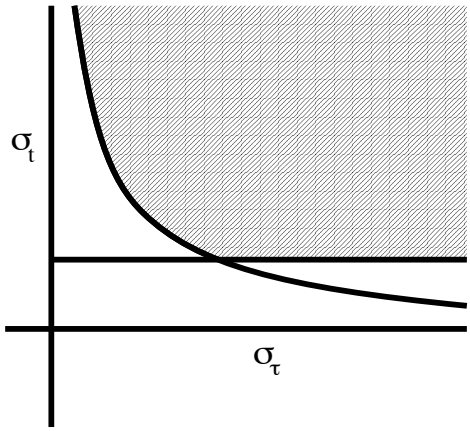


Figure 7-6: Area for which multigrid acceleration is expected

6 the inequalities in (7-9) are represented graphically and the shaded region corresponds to the region where both inequalities are satisfied. Clearly, four different cases can be distinguished: both inequalities are satisfied (shaded region), none are satisfied and only one is satisfied.

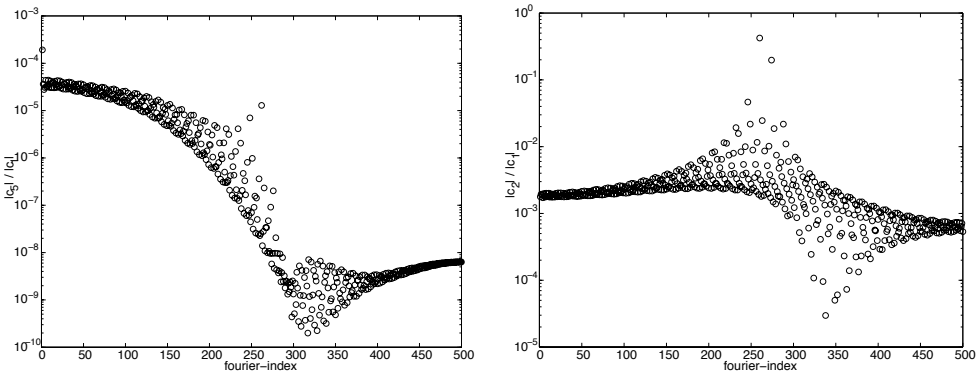
In order to study the damping characteristics corresponding to these four cases we perform a numerical simulation of the 1D inviscid Burgers equation with boundary conditions given by

$$q(x,0) = 1 \text{ and } q(0,t) = 1 + \frac{1}{2}\sin(\omega t) \tag{7-10}$$

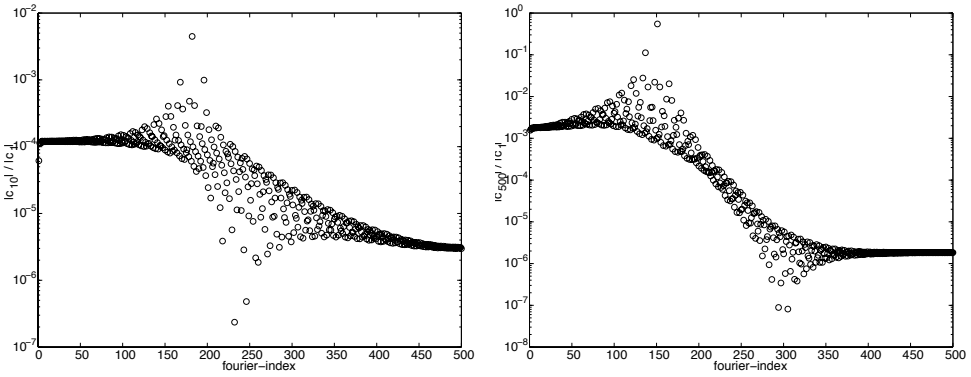
where ω represents the frequency of the disturbance at the inflow boundary condition. For the spatial discretisation we take the first order Roe scheme as in section 7.3. The solution after 1000 time steps, obtained with $\sigma_t = 1.0$, is determined. Then we perform one time step for four (σ_t, σ_τ) pairs corresponding to the four regions described above. As a test case we take $\omega = 1.0$ and $L = 10$, where L denotes the length of the physical domain. Computations are performed on a uniform grid with grid spacing $\Delta x = 1/100$. For this typical example $\xi = 2.44$.

The error in the solution during the pseudo time iteration is determined as in section 7.3 and q^∞ is determined with local accuracy $\epsilon = 10^{-12}$ as defined in (6-27). In figure 7-7 the damping characteristics are presented for four pairs of (σ_t, σ_τ). In figure (a) both requirements in (7-9) are satisfied, in (b) only the second requirement is satisfied, in (c) none of the requirements are satisfied and in (d) only the first requirement is satisfied. Because σ_τ determines the convergence rate, the damping characteristics are determined for an appropriate choice of the pseudo time level in each case. Only $(\sigma_t, \sigma_\tau) = (10, 1000)$ satisfies both requirements in (7-9). As can be observed from figure 7-7(a) the higher frequency components of the error are damped more rapidly than the lower frequencies, which indicates desired smoothing properties for multigrid acceleration. Clearly, no smoothing is obtained for the choices of σ_t and σ_τ in figures 7-7(b) and 7-7(c). Although the damping characteristic in figure 7-7d is not as good as in figure 7-7(a) (especially the bump between Fourier-index 100 and 200), one might argue that a little smoothing is obtained. However, as observed earlier good smoothing is only obtained if a strong spatial coupling exists in the numerical operator. For the present simulations the Jacobi matrix is determined only once. The number of required outer iterations to obtain the local accuracy 10^{-12} decreases significantly for $(\sigma_t, \sigma_\tau) = (10, 1000)$ if the Jacobi matrix is updated every pseudo time step whereas for the other three choices of σ_t and σ_τ the number of required outer iterations remains approximately the same. This further shows that the left-hand side of (7-1) is dominated by the spatial terms for $(\sigma_t, \sigma_\tau) = (10, 1000)$ and no smoothing may be expected for $(\sigma_t, \sigma_\tau) = (10, 1)$. Additionally we have performed simulations for higher frequencies ω which confirmed the findings for $\omega = 1$.

The above results indicate that the model presented in (7-9) predicts the smoothing rate rather well. Smoothing is only obtained for the choice of (σ_t, σ_τ) that lies in the shaded region in figure 7-6. Additionally, we observe



(a) $(\sigma_t, \sigma_\tau) = (10, 1000)$ (b) $(\sigma_t, \sigma_\tau) = (1, 10^{10})$



(c) $(\sigma_t, \sigma_\tau) = (1, 100)$ (d) $(\sigma_t, \sigma_\tau) = (10, 1)$

Figure 7-7: Damping characteristics for various combinations of σ_t and σ_τ.

that the frequency ω does not influence the smoothing properties for the different choices of (σ_t, σ_τ). This agrees well with the fact that the present approach relies on an estimation of the order of magnitude of the terms in (7-1). However, we observe that no attention is paid to the accuracy of the solution after one time step, only the smoothing properties of the numerical scheme are studied. For high frequencies a small σ_t is necessary in order to resolve the solution accurately in time. In this way the first inequality in (7-9) is not satisfied and no smoothing may be expected.

7.5 Conclusions

In this chapter an analysis of multigrid performance for unsteady flows was given. As a test case the unsteady flow described in chapter 6 is used. It turns out that independent of the multigrid settings such as pre and post relaxations, number of grid levels and type of cycles, no convergence acceleration is obtained. Also changes in the smoothing operator led to the same conclusion.

A Fourier analysis of the error on cross sections of the computational domain indicates that there is a difference in smoothing properties in x- and y-direction. In x-direction high and low frequencies in the error are damped equally, thus no smoothing is obtained. However, reasonable smoothing in y-direction is present. By means of a simple order of magnitude analysis we argue that this preference is related to the magnitude of the eigenvalues of the flux Jacobi matrix in both directions which is influenced greatly by the aspect ratio of the grid.

Similarly, we argue that the terms related to the temporal integration in both physical and pseudo time dominate the smoothing operator. Since for good multigrid performance a sufficient spatial coupling in the smoothing operator must be present, this explains the bad multigrid performance. These observations are further substantiated by a model equation study. In this study criteria on the CFL-numbers in physical and pseudo time are derived such that the smoothing operator is dominated by the flux Jacobi matrix. Numerical experiments show that proper smoothing properties for multigrid application are obtained if these criteria are obeyed. However, in the boundary layer application of chapter 6 these criteria do not comply with the demand of sufficient small Δt for time-accuracy.

Since the bottleneck for fast convergence is related to the behavior in x-direction the question arises whether it is possible to decouple the temporal treatment of the fluxes in the two spatial directions. Therefore, in the next chapter a new family of second order time integration schemes will be discussed which treat the flux in x-direction explicitly and the flux in y-direction implicitly.

Chapter 8

Partially implicit time integration schemes

8.1 Introduction

For flow problems where grids with high aspect ratios are required e.g due to complex geometries or thin boundary layers the stability time step of an explicit time integration scheme may be much smaller than the time step which would be required for accuracy reasons. Also, different convergence characteristics may be present in the corresponding coordinate directions as was observed in the previous chapter. In this chapter we develop new schemes which are especially adapted to such problems. In these schemes the flux in one direction is treated explicitly, whereas the flux in the other direction is taken in an implicit way. We call these schemes partially implicit. We consider the supersonic flow over a flat plate, described in chapter 6, as the test case. With respect to CPU time no spectacular gain is expected compared to the required CPU time for an explicit scheme since the aspect ratio for this test case is not very large. However, for the development of new schemes this constitutes no problem and more deserving applications may be found such as e.g. turbulent flows over a pitching wing modelled by the RaNS equations or a turbulent 3D flow over a flat plate as in [71]. In order to substantiate the applicability of the schemes for these cases we artificially increase the aspect ratio of the standard grid for the current test case. In this way the stability time step for an explicit scheme becomes more restrictive and the benefits of a partially implicit scheme are obvious.

In the previous chapter it was illustrated that for a supersonic 2D flow over a flat plate no multigrid acceleration could be obtained due to bad smoothing properties in the streamwise direction of the flow. Because multigrid did not accelerate the convergence the fully implicit method in chapter 7 could not

compete with the explicit reference method with respect to CPU time for a relevant choice of the numerical parameters. As the accuracy time step corresponds approximately to the numerical stability time step in streamwise direction in this test case, we suggest to distinguish between the flux in the streamwise and normal direction and to treat the streamwise flux explicitly and the normal flux implicitly in the time integration method. In this way the stiffness problem due to the fine grid in normal direction near the solid wall is removed. In Ref. [6] this idea has already been applied to an unsteady shock-induced oscillatory flow over an airfoil modelled by the thin-layer Navier-Stokes equations where the main motivation for this approach was related to memory requirements for a fully implicit method. There the following time integration scheme was suggested

$$\left(\frac{I}{\Delta t}+\frac{\partial F_y}{\partial q}\right)(q^{n+1}-q^n)=-F_x(q^n)-F_y(q^n) \tag{8-1}$$

where F_x and F_y denote the numerical fluxes in x- and y-direction respectively and $\partial F_y/\partial q$ symbolically represent the Jacobi matrix of the numerical flux F_y. This scheme can be viewed as a mix of the linearized Euler backward scheme and the Euler forward scheme applied to the fluxes F_y and F_x respectively. It is only first order accurate in time as can be shown using a Taylor expansion. Because (8-1) is a mixture of an explicit and an implicit scheme it cannot be expected that the numerical method is stable for all time steps which was confirmed by numerical experiments in [6]. It was observed that the time step for the partially implicit scheme was bounded by stability rather than by accuracy reasons but the time step that could be used was still about a factor 1000 larger than the time step for the strictly explicit scheme.

We found that for our test case a first order implicit time integration scheme does not yield an accurate solution for $\Delta t = 4$, which is the accuracy time step found in chapter 6 for second order schemes. The amplitude of the solution of the instantaneous quantities $U1$ and $U2$ was damped considerably which may be related to the dissipative truncation error of the Euler Backward scheme. In order to obtain an accurate solution the time step had to be decreased, which leads to a considerable increase of CPU time. In particular 400 time steps per period of the dominant oscillation were required in [6] for the fully implicit scheme to obtain an accurate solution. Therefore in this chapter we develop partially implicit schemes of higher order in time. It will be shown that a special combination of the second order Crank-Nicolson scheme with a class of second order Runge-Kutta schemes results in a second order consistent time integration scheme. Next to the consistency requirement of the numerical scheme we also address other numerical aspects such as stability and steady-state-consistency. It turns out that optimal stability properties and steady-state-consistency cannot be combined. Numerical simulations are

performed in order to establish the appropriate hierarchy of requirements on the numerical method.

The contents of the chapter is as follows. In section 8.2 a class of second order mixed schemes is derived. In section 8.3 a subclass of schemes is presented which are steady-state-consistent. The numerical stability of the mixed second order schemes is studied in section 8.4. In section 8.5 numerical results are presented for some specific mixed schemes. Finally, in section 8.6 the conclusions are summarized.

8.2 Mixed multi-stage schemes

In this section we develop a class of second order partially implicit schemes in time by combining the implicit Crank-Nicolson scheme with a class of explicit Runge-Kutta schemes. To this purpose consider the general differential equation

$$\frac{dq}{dt} + h(q) = \frac{dq}{dt} + f(q) + g(q) = 0 \tag{8-2}$$

where h is a sufficiently smooth function of q and f and g are a general decomposition of h. In the following we will treat the function f explicitly and the function g implicitly. For incompressible flows which are convectively dominated a commonly used splitting is to treat the nonlinear inviscid terms explicitly whereas the linear viscous terms are treated implicitly, see e.g. [63]. To this purpose the Adams-Bashforth Crank-Nicolson scheme is often incorporated [68] which is given by

$$q^{n+1} - q^n + \frac{1}{2}\Delta t(3f(q^n) - f(q^{n-1})) + \Delta t\frac{1}{2}(g(q^{n+1}) + g(q^n)) = 0 \tag{8-3}$$

It is known that in case the viscosity is small the scheme is unstable if the strict Von Neumann criterion is required [68]. This instability is related to the stability region of the explicit Adams-Bashforth scheme [4] which does not contain a part of the imaginary axis. Also, the scheme requires three time levels which may introduce parasites [4]. In this chapter we develop a class of single step partially implicit schemes by combining the Crank-Nicolson scheme with a certain class of Runge-Kutta schemes which in general have more desirable stability properties than the Adams-Bashforth scheme.

8.2.1 Second order multi-stage schemes

In this section we give necessary conditions for second order accuracy of a typical class of multi-stage schemes. To obtain second order accuracy we mix

the Crank-Nicolson scheme with a class of p-stage Runge-Kutta schemes given by

$$\left\{ \begin{array}{lcl} q^{(0)} & = & q^n \\ q^{(k)} & = & q^{(0)} - \alpha_k \Delta t f(q^{(k-1)}), \quad k = 1, \ldots, p \\ q^{n+1} & = & q^{(p)} \end{array} \right. \tag{8-4}$$

where α_k is a real number and $p \geq 2$. For $p = 4$ we obtain the explicit Runge-Kutta scheme defined in section 2.4.1. Using a Taylor expansion it is not hard to show that, in order to obtain a second order consistent explicit scheme of the type defined above, the coefficients in the final two stages must satisfy $\alpha_{p-1} = 1/2$ and $\alpha_p = 1$. The choice of the coefficients, α_k, for $0 < k < p-1$ and $p > 2$ does not affect the second order accuracy but may change the stability region considerably. In fact the schemes in (8-4) are at best second order accurate. To clarify this we first consider the three-stage scheme. For second order accuracy the values of α_p and α_{p-1} are fixed as mentioned above. A Taylor expansion around q^n yields for the local truncation error $E(\Delta t)$ as defined later on in (8-7)

$$E(\Delta t) = \left(\left[\frac{1}{2}\alpha_1 - \frac{1}{6} \right] (f_q)^2 f - \frac{1}{24} f_{qq} f^2 \right) (\Delta t)^3 + \mathcal{O}((\Delta t)^4) \tag{8-5}$$

where the subscript denotes the corresponding derivative and all functions are evaluated in q^n. Obviously, for general functions f it is not possible to obtain a globally third order accurate scheme. For a general p-stage scheme, with $p > 3$, the statement that the accuracy is at best of second order follows immediately from the results for the three-stage scheme because in every additional stage a factor Δt arises and therefore the k-th stage contributes to the $\mathcal{O}((\Delta t)^{p-k+1})$ term and higher order terms of the local truncation error.

The question arises whether it is possible to combine a second order explicit scheme with a second order implicit scheme resulting in an overall second order scheme in time. The requirement that the scheme should be a mixture of two second order schemes is quite natural because the numerical scheme should retain its second order accuracy in case one of the functions f or g vanishes. Similar to the accuracy results of the explicit schemes a specific class of second order accurate schemes is obtained if and only if the last two stages are fixed as follows:

$$\left\{ \begin{array}{lcl} q^{(0)} & = & q^n \\ q^{(k)} & = & q^{(0)} - \alpha_k \Delta t \; f(q^{(k-1)}) - \beta_k \Delta t \; g(q^{(k)}), \quad k = 1, \ldots, p-2 \\ q^{(p-1)} & = & q^{(0)} - \frac{1}{2}\Delta t \; f(q^{(p-2)}) - \frac{1}{2}\Delta t \; g(q^{(p-1)}) \\ q^{(p)} & = & q^{(0)} - \Delta t \; f(q^{(p-1)}) - \frac{1}{2}\Delta t \; g(q^{(p)}) - \frac{1}{2}\Delta t \; g(q^{(0)}) \\ q^{n+1} & = & q^{(p)} \end{array} \right. \tag{8-6}$$

where α_k and β_k are real numbers and $p \geq 2$. This class of schemes is a combination of the Runge-Kutta schemes in (8-4) with the Crank-Nicolson scheme. The stages corresponding to $k < p-1$ do not affect the accuracy but may affect the stability as will be shown in section 8.4.

The accuracy of these schemes can be derived by considering the two-stage scheme ($p = 2$) first. The local truncation error E of (8-6) is defined as in Ref. [4] by

$$E(\Delta t) = q^{n+1} - q^n + \Delta t(f(q^{(1)}) - \frac{1}{2}g(q^{(2)}) - \frac{1}{2}g(q^{(0)})) \tag{8-7}$$

A Taylor expansion around q^n yields $E(\Delta t) = \mathcal{O}((\Delta t)^3)$ which shows that the two-stage scheme in (8-6) is a locally third order consistent scheme and therefore a globally second order consistent scheme is obtained [4]. In the following discussions on the accuracy of the numerical schemes the order relates to the global truncation error unless stated otherwise.

For the schemes in (8-6) the function g is treated implicitly everywhere which implies that a nonlinear system has to be solved every stage. This is in contrast with the fully implicit scheme where the nonlinear system has to be solved only once. The second order consistency is not affected if the function g is treated explicitly in the stages with $k < p$. However, for our present flow application the function g corresponds to the flux in the normal direction as was outlined in the introduction. The stability time step in the normal direction was found to be too small compared to the accuracy time step due to the fine grid near the solid wall. If the function g is treated explicitly in the first stage, the first stage is fully explicit which results in a severe stability restriction on the time step. For this reason we treat the function g implicitly in all stages.

8.3 Steady-state-consistency

Next to the order of consistency the ability of a numerical scheme to yield a steady state solution independent of the time step is required. For the well-known second order scheme of MacCormack [60], Jameson showed in Ref. [47] that the steady-state solution depends on the time step and for this reason MacCormacks scheme was rejected in the selection of an explicit solver in [104]. In the following the ability of a numerical scheme to yield a steady state solution independent of the time step is denoted by steady-state-consistency. First we show that the schemes in (8-6) are not steady-state-consistent in general and subsequently a subclass of schemes is presented which are steady-state-consistent.

8.3.1 Model equation

In this section we show that the four stage scheme in (8-6) with $\beta_1 = 0$ and $\beta_2 = 0$ is not steady-state-consistent. The choice for this scheme appears to be sensible because of its computational efficiency (no implicit part in the first two stages) and its stability properties (see section 8.4). However, the numerical results presented in section 8.5.2 indicate that the steady-state-consistency is an important requirement for the choice of a proper time integration scheme. The steady-state-consistency of this scheme is studied for a model equation next.

Consider the one-dimensional viscous Burgers equation given by

$$q_t + (\frac{1}{2}q^2)_x = q_{xx} \tag{8-8}$$

In Ref. [11] a table of solutions to (8-8) is given and a specific time dependent solution is given by

$$q(x,t) = -\frac{2\sinh x}{\cosh x + e^{-t}} \tag{8-9}$$

Clearly, this solution evolves into the steady solution $q(x) = -2\tanh x$. The error in the numerical solution after one time step can be written as

$$\widetilde{E}(\Delta t) = \widetilde{q}(x, t+\Delta t) - q(x, t+\Delta t) \tag{8-10}$$

where $\widetilde{E}$ is the truncation error of the numerical scheme, $\widetilde{q}$ corresponds to the solution obtained with a numerical scheme and q denotes the exact solution given by (8-9). In order for the numerical scheme to recognize a steady state solution we demand that

$$\lim_{t\to\infty} \widetilde{E}(\Delta t) = 0 \tag{8-11}$$

independent of Δt. For the Runge-Kutta schemes and the Crank-Nicolson scheme the local truncation error can be written as

$$\widetilde{E}(\Delta t) = \sum_{k=3}^{\infty} \gamma_k(q)(\Delta t)^k \tag{8-12}$$

where the function $\gamma_k(q)$ is a polynomial in terms of derivatives of the function h in (8-2) with respect to q as can be derived from a Taylor expansion. In case a stable steady state exists, which corresponds to $h(q) = 0$, the error E vanishes due to the specific form of the truncation error. For the mixed four-stage scheme the general structure in the truncation error is less transparent.

However, it is possible to calculate the limit in (8-11) for the model equation (8-8). After some elementary and tedious calculations it is found that

$$\lim_{t\to\infty} \widetilde{E}(\Delta t) = \frac{32\left(6\left(\cosh(x)\right)^2 - 6 - \left(\cosh(x)\right)^4\right)\sinh(x)}{3\left(\cosh(x)\right)^7}(\Delta t)^3 + \mathcal{O}((\Delta t)^4) \tag{8-13}$$

which shows that the error does not vanish and thus the obtained steady state solution depends on Δt for this particular scheme.

8.3.2 Steady-state-consistent schemes

The model equation studied above for the four-stage scheme reveals that not all schemes (8-6) are steady-state-consistent. However, we will show that there exists a subclass of second order steady-state-consistent schemes. Consider the two-stage scheme in (8-6) and assume that the differential equation (8-2) has a steady state solution denoted by q^*. If we assume that the Navier-Stokes equations have a unique solution we show that this solution will also be found by a specific class of mixed schemes. Consider the first stage in (8-6)

$$q^{(1)} = q^* - \alpha_1 \Delta t\ f(q^*) - \beta_1 \Delta t\ g(q^{(1)}) \tag{8-14}$$

with $q^n = q^*$. Clearly, the solution q^* satisfies this equation if and only if $\alpha_1 = \beta_1$, if we disregard the trivial case where f and g are separately zero.

The numerical schemes which we use have the form (2-44) and the first stage of (8-6) is given by

$$(1 + \alpha_1 \Delta t g_q)\,\Delta q = -\alpha_1 \Delta t (f(q^{(0)}) + g(q^{(0)})) \tag{8-15}$$

where $q^{(0)} = q^*$ and $\Delta q = q^{(1)} - q^{(0)}$. The right-hand side of (8-15) vanishes because $q^{(0)}$ is the steady state solution yielding $q^{(1)} = q^{(0)}$. Generalizing this approach to a p-stage mixed scheme in (8-6) the following class of second order steady-state-consistent mixed schemes is obtained

$$\begin{array}{lcll} q^{(0)} & = & q^n & \\ q^{(k)} & = & q^{(0)} - \alpha_k \Delta t \left(f(q^{(k-1)}) + g(q^{(k)})\right), & k = 1, \ldots, p-2 \\ q^{(p-1)} & = & q^{(0)} - \frac{1}{2}\Delta t \left(f(q^{(p-2)}) + g(q^{(p-1)})\right) & \\ q^{(p)} & = & q^{(0)} - \Delta t f(q^{(p-1)}) - \frac{1}{2}\Delta t g(q^{(p)}) - \frac{1}{2}\Delta t g(q^{(0)}) & \\ q^{n+1} & = & q^{(p)} & \end{array} \tag{8-16}$$

such that $q^{n+1} = q^n$ independent of the time step. Apart from the accuracy and steady-state-consistency of a numerical scheme the numerical stability plays an important role as well and will we addressed next.

8.4 Stability

For the mixed schemes in (8-6) the best attainable stability properties are coupled to the stability of the explicit scheme since the function g, which is treated implicitly, may vanish and the mixed scheme reduces to the explicit scheme. It is not clear how the mixing affects the explicit stability region. At best the mixed scheme remains A-stable with respect to the complete explicit stability region which we denote in the sequel by M-stable (Mixed). However, the mixed schemes in general are not M-stable. First we show that the four-stage scheme of section 8.3.1, which is not steady-state-consistent, is M-stable. Second we study the stability properties of the steady-state-consistent schemes in (8-16).

8.4.1 M-stable schemes

In this section we show that a subclass of the mixed schemes defined in (8-6) is M-stable. As an illustration we take the four-stage scheme defined in 8.3.1. In order to study the stability properties of the four-stage scheme we perform a Von Neumann analysis [72] on the linear scalar equation

$$q_t + aq_x + bq_y = 0 \tag{8-17}$$

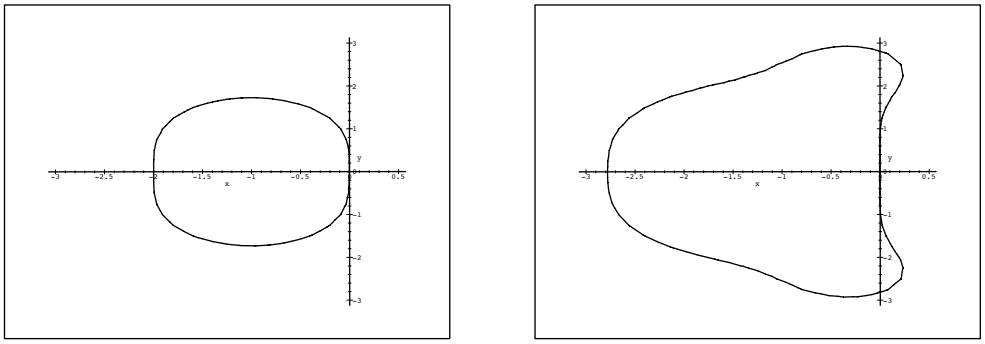
which represents a model of (8-2) and a and b are constants. Introducing an appropriate grid and application of the method of lines to separate the temporal integration from the spatial discretisation, as outlined in section 2.4, a coupled set of ordinary differential equations is obtained for the solution q in every grid point. If we assume that a steady state solution exists, the error in the solution with respect to this steady state solution obeys the same set of ordinary differential equations. Assuming periodic boundary conditions the error can be represented by a Fourier-series and the Fourier-coefficients obey

$$\frac{dc_k}{dt} = \widehat{z}c_k + \widehat{\xi}c_k \tag{8-18}$$

where c_k corresponds to the k-th Fourier-coefficient of the error and $\widehat{\xi}$ and $\widehat{z}$ are complex numbers which depend on the grid and spatial discretisation. For the present flow application $\widehat{z}$ and $\widehat{\xi}$ are related to the fluxes f and g in the partially implicit schemes respectively. Applying the four-stage scheme to (8-18) and after some algebraic manipulations we get

$$\zeta(z,\xi) = \left|\frac{c_k^{n+1}}{c_k^n}\right| = \left|\frac{1 + p(z) - \frac{1}{4}\xi^2}{(1-\frac{1}{2}\xi)^2}\right| \tag{8-19}$$

where $z = \Delta t\widehat{z}$, $\xi = \Delta t\widehat{\xi}$, ζ is the so-called stability function, $p(z) = z(1 + \frac{1}{2}z(1 + \frac{1}{3}z(1 + \frac{1}{4}z)))$, and $|1 + p(z)|$ corresponds to the stability function of the four-stage Runge-Kutta scheme. The stability region of a scheme is the set of complex numbers z and ξ for which $\zeta \leq 1$. We will only consider those z and ξ with negative or zero real part, since that corresponds to the dissipative character of the Navier-Stokes equations. We now return to the aspect of M-stability. For the four-stage scheme the following theorem holds.



(a) Two-stage Runge-Kutta (b) Four-stage Runge-Kutta

Figure 8-1: Stability region in the complex plane z-plane of the two- and four-stage Runge-Kutta schemes in (8-4).

Theorem 8.1 *The four-stage scheme in (8-6) with* $\beta_1 = 0$ *and* $\beta_2 = 0$ *is M-stable.*

Proof Assume $\xi = x + Iy$ and $\Re(\xi) \leq 0$. Let Γ denote the stability region of the four-stage Runge-Kutta scheme shown in figure 8-1. Because $|1 + p(z)|$ is the stability function for the purely four-stage explicit scheme, $|1 + p(z)| \leq 1, \quad \forall z \in \Gamma$ which results in the following bound on the stability function in (8-19)

$$\zeta(z, \xi) \leq \frac{1 + \frac{1}{4}|\xi|^2}{|1 - \frac{1}{2}\xi|^2} = \frac{1 + \frac{1}{4}x^2 + \frac{1}{4}y^2}{1 + \frac{1}{4}x^2 + \frac{1}{4}y^2 - x} \tag{8-20}$$

This expression is smaller than one in view of the assumption that $\Re(\xi) \leq 0$ which proves that the scheme is M-stable. □

This theorem can be extended to a class of Mixed-schemes which yields

Theorem 8.2 *The schemes in (8-6) with $\beta_j = 0$ for $j = 1, \ldots, p-2$ are M-stable.*

Proof Similar to the proof of theorem 8.1, where $p(z)$ corresponds to the underlying Runge-Kutta scheme in (8-4).

□

Although all these schemes are M-stable their stability region differs significantly as is shown in figure 8-1 for the special cases of the two- and four-stage schemes. Compared to the stability region of the two-stage explicit scheme, which does not contain a part of the imaginary axis, the stability region of the four-stage scheme contains a significant part of the imaginary axis.

8.4.2 Stability of steady-state-consistent schemes

In this section we study the stability of the three- and four-stage schemes specified in (8-16). As in the previous section we can derive the stability functions which are now given by

$$\begin{aligned} \zeta_3(z,\xi) &= \left| \frac{12 - 4\,\xi - 3\,\xi^2 + 12\,z - 4\,z\xi + 6\,z^2 + 2\,z^3 + \xi^3}{(-3+\xi)\,(-2+\xi)^2} \right| \\ \zeta_4(z,\xi) &= \left| \frac{48 - 28\,\xi - 8\,\xi^2 + 7\,\xi^3 - \xi^4 + p_1(\xi)z + p_2(\xi)z^2 + 8z^3 + 2z^4}{(-4+\xi)\,(-3+\xi)\,(-2+\xi)^2} \right| \end{aligned} \tag{8-21}$$

with $p_1(\xi) = 48 - 28\xi + 4\xi^2$ and $p_2(\xi) = 24 - 6\xi$ and the subscript on ζ refers to the three- or four-stage scheme. For any z in the stability region of the explicit three- or four-stage scheme we get

$$\lim_{|\xi|\to\infty} \zeta_3(z,\xi) = 1 \quad \text{and} \quad \lim_{|\xi|\to\infty} \zeta_4(z,\xi) = 1 \tag{8-22}$$

respectively which agrees with the fact that the Crank-Nicolson scheme is A-stable. In contrast to the four-stage scheme discussed in the previous section it is straightforward to show that e.g. $\zeta_4(\frac{3}{2}I, -\frac{7}{5}I) > 1$, which implies that the present four-stage scheme is not M-stable. However, there may exist an area in the complex z-plane which does contain a significant interval of the imaginary axis of the form $[-dI, dI]$, with d a constant, and with respect to which the scheme is stable for all ξ with $\Re(\xi) \leq 0$. In the following we denote this property of the scheme by PM-stability (Partially Mixed). For the stability analysis we are only interested in the case where $\Re(\xi) \leq 0$ and thus the functions ζ_3 and ζ_4 are analytic. If we can find a closed curve in the complex z-plane (as e.g. the boundary in figure 8-1) on which the scheme is

stable and which contains a part of the imaginary axis as described above, it follows from the maximum principle [1] that in the region enclosed by this curve the scheme is stable as well and a PM-stable method is obtained.

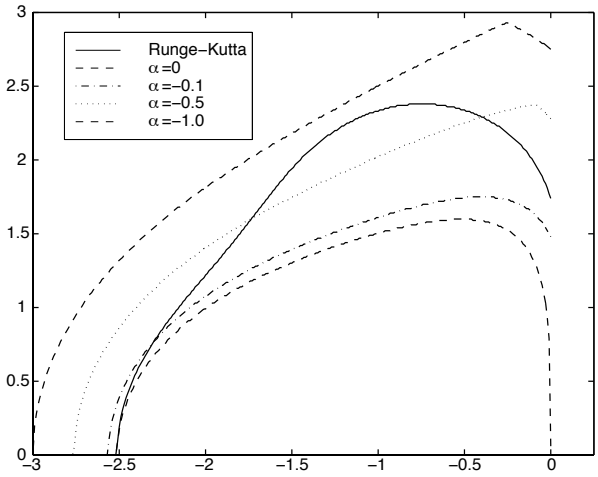


Figure 8-2: Stability curves for ζ_3 in (8-21) in the complex z-plane for $\xi = \alpha + Iy$ with $y \in [-100, 100]$. The upper dashed lines corresponds to $\alpha = -1.0$. The solid line represent the stability boundary of the three-stage Runge-Kutta scheme (8-4).

To study the PM-stability we numerically determine the boundary in the complex z plane for which $\zeta = 1$. To this purpose we introduce a uniform grid in the z-plane by

$$z_{i,j} = -ih + Ijh, \quad i = 0, \ldots, 300, j = 0, \ldots, 300 \tag{8-23}$$

where h is the cell-width which we take 0.01. The influence of the complex number ξ on the stability is determined for

$$\xi_j = \alpha + Ijk, \quad j = -1000, \ldots, 1000 \tag{8-24}$$

with $\alpha \in \{0, -0.1, -0.5, -1\}$ and k denotes the cell width which we take 0.1. Only a small family of α values is considered which will become clear in the following. The range of the imaginary part of ξ is quite large and in view of the result in (8-22) this represents an approximation for $|\xi| \to \infty$ for fixed α. Because $\zeta(z, \bar{\xi}) = \zeta(\bar{z}, \xi)$ only the complex numbers with $\Im(z) \geq 0$ have to be considered. The procedure is as follows. The value of the stability function in (8-21) is determined for every z in (8-23) and every ξ in (8-24). If the

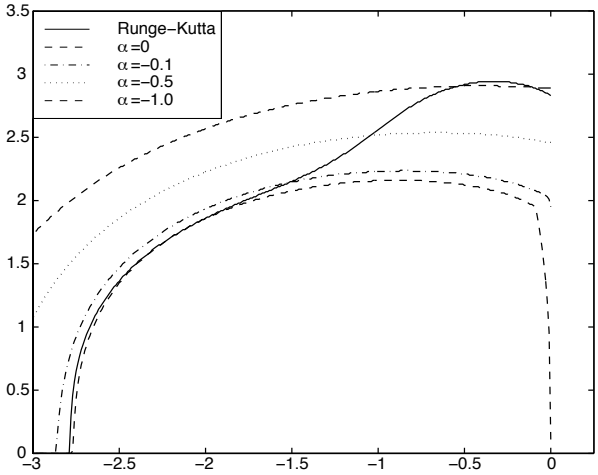


Figure 8-3: Stability curves for ζ_4 in (8-21) in the complex z-plane for $\xi = \alpha + Iy$ with $y \in [-100, 100]$. The upper dashed lines corresponds to $\alpha = -1.0$. The solid line represent the stability boundary of the four-stage Runge-Kutta scheme (8-4).

stability function is less than or equal to one for all ξ at a fixed α and z, the corresponding z-value is identified as stable for this particular value of α. In this way a stability region in the complex z-plane is obtained.

In figure 8-2 we have plotted the stability boundaries for the three-stage scheme for $\alpha \in \{0, -0.1, -0.5, -1\}$. If the z-plane is traversed in a vertical way starting from $\Im(z) = 0$, the boundaries correspond to the first time that $\zeta_3 > 1$ for fixed $\Re(z)$. In this way the different stability behavior on the imaginary axis in the complex z-plane is emphasized for different choices of α. The solid line represents the stability boundary of the explicit Runge-Kutta scheme which corresponds to $\xi = 0$ and clearly contains a part of the imaginary axis. For $\alpha = 0$ two observations can be made: the stability region is considerably smaller than the explicit stability region and the scheme is not stable on a part of the imaginary axis. By decreasing the value of α the stability region is increased and the scheme is stable on a part of the imaginary axis. For $\alpha = -1.0$ the explicit stability region is fully contained in the stability region. In figure 8-3 a similar plot is presented for the four-stage scheme. Again for $\alpha = 0$ the scheme is not stable on the imaginary axis and the stability region is reduced with respect to the explicit stability region. If the value of α is decreased, the scheme becomes stable on a part of the imaginary axis. It is clear that the three- and four-stage schemes are not PM-stable because no

part of the imaginary axis is contained in the stability region for $\alpha = 0$.

Although both the three- and four-stage scheme are not stable on the imaginary axis for $\alpha = 0$, the practical stability of the schemes (see section 8.5.2), which means that a stable and accurate simulation is possible, can be explained by a viscosity argument. The upwind spatial discretisation defined in section 2.3, which is also used for the flux in y-direction, results in an artificial viscosity term as was shown in (2-20). This viscosity term contributes to the negative real part of the Fourier footprint [72]. Also, physical viscosity is present which contributes to the negative real part of the Fourier footprint as well. Therefore, in practice the real part of ξ is never exactly zero and the numerical scheme is in fact stable on a part of the imaginary axis in the complex z-plane. This viscosity argument also applies to the flux which is treated explicitly. From figures 8-2 and 8-3 it follows that the numerical method is stable for a significant region in the complex z-plane (even for $\Re(\xi) = 0$) when $\Re(z)$ is sufficiently small. This further emphasizes the practical use of partially implicit schemes.

At this point some basic theoretical properties such as consistency, stability and steady-state-consistency have been categorized for several families of partially implicit schemes. With respect to stability and steady-state-consistency, however, it is not clear which one should be preferred. Therefore numerical simulations are performed next in order to determine which property is more important.

8.5 Numerical results and implementation

8.5.1 Implementation

In this section we perform a simulation for the boundary layer test case defined in chapter 5 and we take the solution at $t = 10,000$ as the initial condition. Every stage a nonlinear system of equations has to be solved corresponding to the numerical system described in (2-42) and we introduce a pseudo-time derivative in each stage. A priori there is no relation between the magnitude of the pseudo-time step for every stage and for convenience we use the same CFL-number to determine the pseudo-time step.

In chapter 6 we found that it sufficed to calculate the Jacobi matrix only once during one time step. For the partially implicit scheme we encountered convergence problems if the Jacobi matrix was frozen during all stages. Therefore, we recalculate the Jacobi matrix every stage but freeze it during a stage.

For the fully implicit scheme we used the Gauss-Seidel method to approximate the solution to the linear system which arises at every pseudo time step (6-17). Because only the flux in y-direction is treated implicitly the resulting Jacobi matrix consists of three bands of 4×4 matrices instead of five bands for

the fully implicit scheme. In order to solve the corresponding linear systems we use a block tridiagonal solver [88] which is a direct solver.

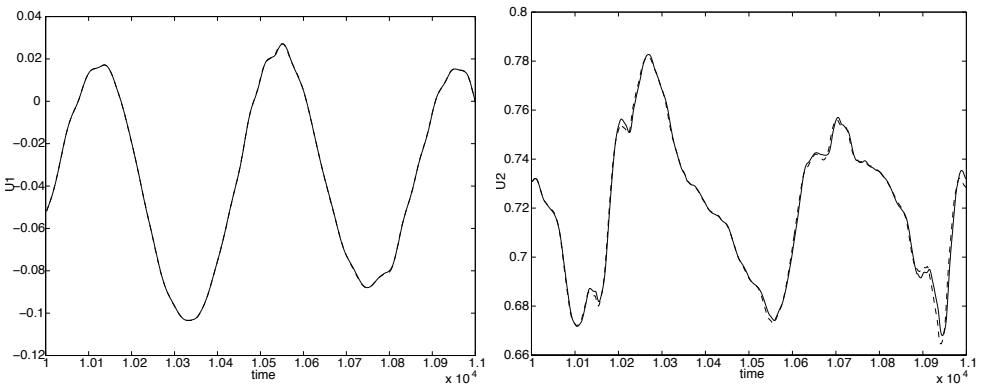
The accuracy time step for instantaneous flow quantities is $\Delta t = 4$ as was arrived at in chapter 6. However, as observed before, the total stability of the numerical scheme is determined primarily by the stability of the explicit part because the flux which is treated implicitly may vanish. Therefore we calculate the stability time step in x-direction as in section 6.2.4 to obtain a reference magnitude of the time step. To this purpose we take the solution at $t = 10,000$ which yields for the stability time step in x-direction, $\Delta t_{stab,x} \approx 2$ which is obtained with $CFL = 1.5$, and which is well below Δt_{acc}.

8.5.2 Numerical results

In this subsection numerical results are presented for the M-stable four-stage scheme and the steady-state-consistent two-, three- and four-stage schemes.

Steady-state-consistent schemes

In this subsection we present the numerical results for the two-, three- and four-stage steady-state-consistent schemes. Similar to the results presented in [6] for the scheme (8-1) we find that the two-stage scheme is not stable for all time steps. For $\Delta t = 2$ the numerical simulation was not stable. However,

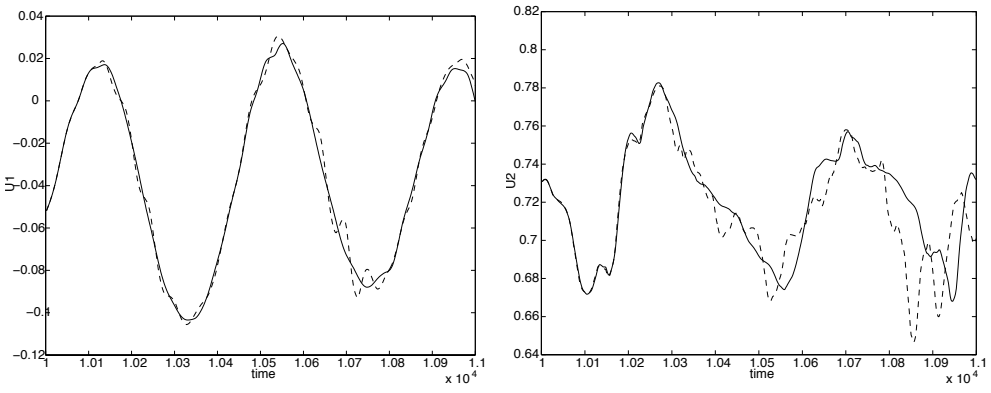


(a) Velocity $U1$ (b) Velocity $U2$

Figure 8-4: Instantaneous solutions obtained with the two-stage partially implicit scheme (8-16) (dashed line) and the reference solution (solid line) for $\Delta t = 1$.

for $\Delta t = 1$ the numerical simulation proved to be stable and we were able to use $\Delta\tau = 10^3$. With this setting of the numerical parameters a speed up of 1.5 is achieved. compared to the explicit case. In figure 8-4 we have plotted the solution to the instantaneous quantities $U1$ and $U2$ defined in chapter 6 obtained with a time step $\Delta t = 1$ and a local accuracy $\epsilon = 10^{-2}$. The quantity $U1$ is resolved accurately and the more sensitive quantity $U2$ is resolved fairly well. The fact that it is not possible to simulate with $\Delta t_{x,stab}$ may be related to the stability region of the explicit part of the two-stage scheme which is shown in figure 8-1. Although the total scheme is M-stable the explicit stability region does not contain a part of the imaginary axis.

The three- and four-stage schemes in (8-16) also do not contain a part of the imaginary axis in the complex z-plane in case of zero viscosity. However, as mentioned above we think that in practice sufficient viscosity is present. We do, however, expect better stability properties compared to the two-stage scheme since the explicit stability region of these schemes contains a part of the imaginary axis. For the three-stage scheme a stable simulation with $\Delta t = 2$ was not possible and the time step had to be decreased. A simulation with $\Delta t = 1$ results in a solution with comparable accuracy as obtained with the two-stage scheme.



(a) Velocity $U1$ (b) Velocity $U2$

Figure 8-5: Instantaneous solutions obtained with the four-stage partially implicit scheme (8-16) (dashed line) and the reference solution (solid line) for $\Delta t = 2$.

For the four-stage scheme a stable simulation with $\Delta t = 2$ is possible. The local accuracy is taken $\epsilon = 10^{-3}$ and in order to obtain a converged solution

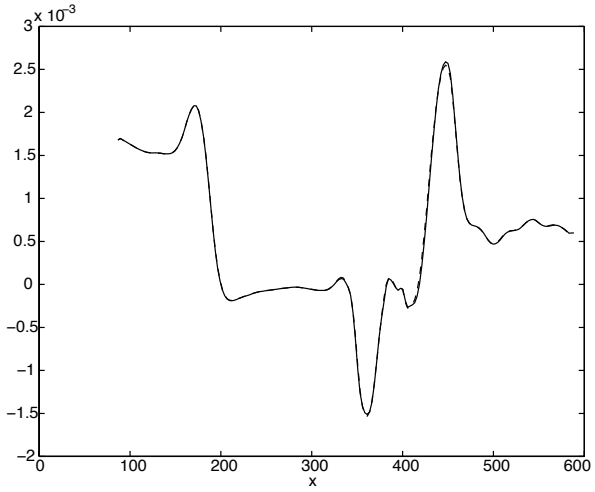


Figure 8-6: The mean skin friction obtained with the four-stage partially implicit scheme (8-16) (dashed line) and the reference solution (solid line) for $\Delta t = 2$.

over the total sampling period a pseudo-time step of $\Delta\tau = 1$ was necessary. In figure 8-5 the instantaneous $U1$ is plotted. Although the main trend is captured reasonably well the two-stage scheme with $\Delta t = 1$ results in a more accurate solution as can be seen in figure 8-4. The deviation in $U2$ is even more pronounced. At first $U2$ is captured rather well, see figure 8-5, but after approximately 200 time units the solution starts to deviate more and additional temporal frequencies arise. Increasing the local accuracy resulted in a similar behavior. The mean skin friction is shown in figure 8-6. Clearly, a simulation with $\Delta t = 2$ yields an accurate solution of the mean skin friction. Similar to the results of the two- and three-stage schemes the quantities $U1$ an $U2$ are resolved accurately for a simulation with $\Delta t = 1$.

M-stable four-stage scheme

In this section we present the numerical results obtained with the four-stage M-stable scheme defined in the previous section. Using the two-stage scheme (8-6) with a time step of $\Delta t = 1$ resulted in a solution with acceptable accuracy. In order to compare the solution obtained with the two-stage and four-stage schemes we first simulate with a time step $\Delta t = 1$. To rule out the influence of the local accuracy we take $\epsilon = 10^{-4}$. It turns out that it is not possible to obtain this accuracy over the total sampling period with an infinite pseudo-time step which was also observed in chapter 6 and we take $\Delta\tau = 10$

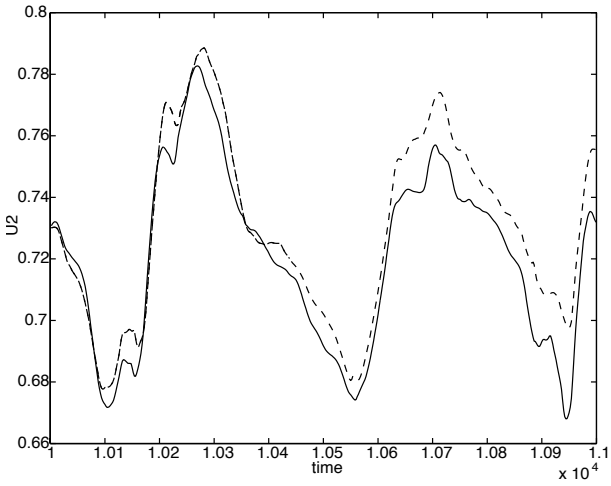


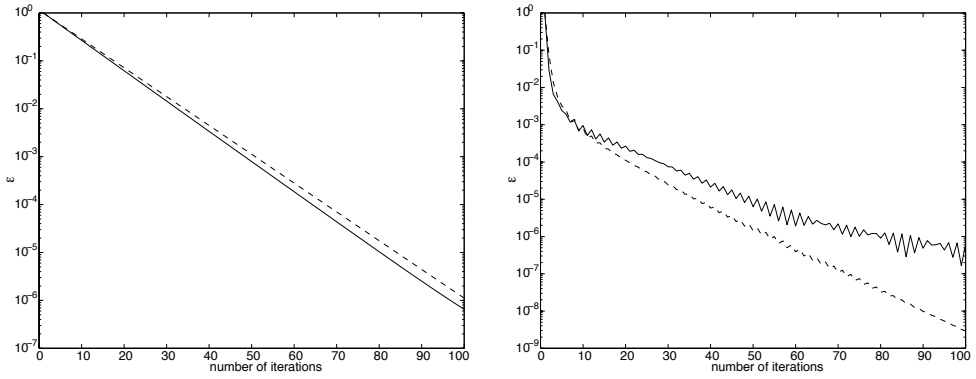
Figure 8-7: Instantaneous solution U2 obtained with the M-stable four-stage partially implicit scheme (dashed line) and the reference solution (solid line) for $\Delta t = 1$.

instead. The result for $U2$ is shown in figure 8-7. Although the main trend is captured the solution differs significantly from the reference solution whereas the solution obtained with the two-stage scheme for the same magnitude of the time step does not show such a large deviation (see figure 8-4). This appears to contradict the better stability properties of the four-stage scheme. Simulation with a time step of $\Delta t = 0.2$, which is comparable to the explicit time step, results in an accurate solution which is in agreement with the second order consistency of the four-stage scheme. This inadequacy of the four-stage scheme is caused by the fact that it is not steady-state-consistent as was shown in section 8.3.1.

8.5.3 Convergence characteristics

With the fully implicit Crank-Nicolson scheme used in chapter 6 an accurate simulation with $\Delta t = 4$ could be performed, which is a factor four larger than the allowable time step obtained for the partially implicit methods in the previous section. In order to compare the performance of the partially implicit schemes to the Crank Nicolson scheme we make some observations concerning the convergence properties and the required CPU time. To this end we perform one time step starting from the solution at $t = 10,000$ with $\Delta t = 1$. For the fully implicit scheme it was shown in chapters 4 and 6 that the number of iterations which is required to obtain a certain accuracy

depends on the magnitude of the pseudo-time step. Therefore the comparison is made for a range of CFL-numbers in pseudo-time. As a representative of



(a) $\Delta\tau = 1$ (b) $\Delta\tau = 1000$

Figure 8-8: Convergence history of the second stage of the two-stage scheme in (8-16) (solid) and the Crank-Nicolson scheme (dashed) as a function of the number of outer iterations for two pseudo time steps.

the partially implicit schemes we take the two-stage scheme in (8-16). We find that it is possible to take $\Delta\tau = \infty$ in the first stage which results in a local accuracy, defined in (6-27), in the order of 10^{-4}. For the second stage we vary the pseudo time step as described above. In figure 8-8 the convergence history for $\Delta\tau = 1$ and $\Delta\tau = 1000$ are presented as a function of the number of outer iterations. Additional simulation were performed for $\Delta\tau = 10$ and $\Delta\tau = 100$ with comparable results to $\Delta\tau = 1000$. In order to explain the results we study the total Jacobi matrix for the Crank-Nicolson scheme as in (2-44) which can symbolically be written as

$$\frac{\partial H}{\partial q} = \left(\frac{I}{\Delta\tau} + I + \frac{1}{2}\Delta t \frac{\partial F_x}{\partial q} + \frac{1}{2}\Delta t \frac{\partial F_y}{\partial q} \right) \tag{8-25}$$

where F_x and F_y denote the numerical fluxes in x- and y-direction respectively. For the first three decades the convergence rate of the fully and partially implicit scheme is very similar for all pseudo time steps which may be explained by looking at the dominant terms in (8-25). In chapter 7 it was shown that $\partial F_x/\partial q$ was the smallest term. Except for $\partial F_x/\partial q$ all other terms in (8-25) occur in the Jacobi matrix for both schemes. Because the Jacobi matrix which we use is already an approximation it appears that a additional small change in the Jacobi matrix, in this case neglecting $\partial F_x/\partial q$, does not have a

significant influence on the initial convergence rate. On the other hand the final residual level, which depends very sensitively on $\Delta\tau$ (section 6.6), is quite different for both methods. In [100, 101] not only the dependence of the steady state solution on the magnitude of the time step was studied but also a large set of different time integration schemes was studied. It was shown that the convergence properties differ significantly for each scheme. The underlying numerical system is very different for the partially implicit and fully implicit schemes which may explain the difference of the final residual levels in figure 8-8. Additionally, it is observed in [100, 101] that it is most likely to find the steady state solution of the underlying differential equation if the time step is approximately equal to the stability time step of an explicit scheme. This is confirmed by the numerical results for $\Delta\tau = 1$ where the final residual levels are quite similar for both methods.

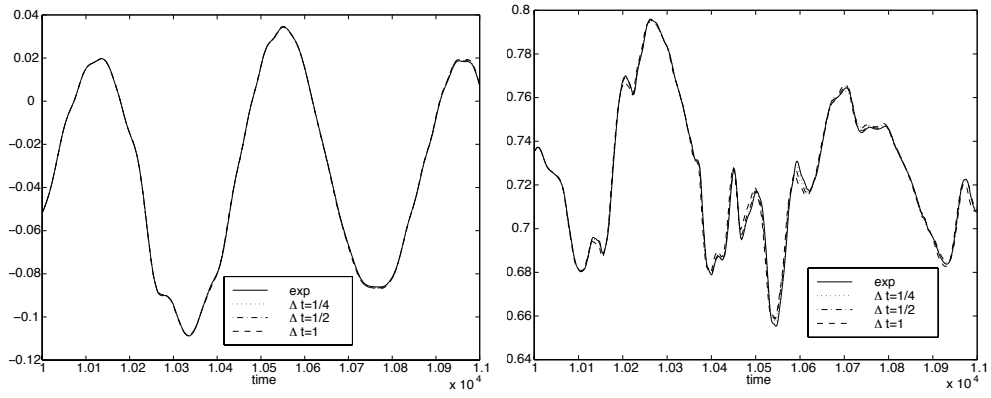
If we look at the CPU time per iteration the following observations can be made. For the fully implicit scheme in chapter 6 we use the Gauss-Seidel method for the linear process in (2-44). However, for the partially implicit scheme we use a block tridiagonal solver, which is a direct solver. Because the Jacobi matrix is frozen, see e.g.(6-17), the linear system is inverted only once. For the Gauss-Seidel method this implies that the diagonal blocks are inverted only once (see 4-16) whereas a LU-decomposition is performed once for the block tridiagonal matrix which is considerably more expensive. Therefore, the total CPU time which includes the storage, determination and inversion of the Jacobi matrix decreases only with a factor of about 1.25 for the partially implicit scheme. The computational costs of one iteration in (6-17) consists of two Gauss-Seidel sweeps for the fully implicit scheme and a forward and backward iteration in case of the block tridiagonal solver for the partially implicit scheme. We find that the speedup factor of one pseudo time step, without taking the storing, determination and inversion of the Jacobi matrix into account, is about a factor 1.86 per iteration in favor of the partially implicit scheme which can be understood as follows. In order to solve the linear system for the partially implicit scheme after the LU-decomposition a forward and backward iteration sweep is necessary which are quite similar to one Gauss-Seidel sweep. Therefore a speedup factor of about $\frac{5}{3}$ is expected due to the smaller computational stencil of the Jacobi matrix.

Summarizing we may conclude that for the present flow application the Crank-Nicolson scheme is about a factor two more efficient than the partially implicit scheme. However, compared to an explicit code a significant speed up factor can be obtained for a grid with high aspect ratio as will be shown next. Therefore, in case the memory requirements become a bottleneck the partially implicit methods constitute an attractive alternative to fully implicit methods.

8.5.4 High aspect ratio

For the present test case the partially implicit schemes yield an accurate solution using a time step of $\Delta t = 1$ which corresponds approximately to the stability time step in x-direction (see section 8.5.2). However, this time step is only about a factor 5 larger than the stability time step for the reference explicit scheme. The reason is the fact that the current aspect ratio is relatively small. Because of this the full potential of these partially implicit schemes is not exploited. In order to further investigate the performance of the partially implicit schemes without turning to another test case we artificially increase the aspect ratio of the grid. To this purpose we increase the stretching ratio of the grid, defined by $\Delta y_{max}/\Delta y_{min}$, to 89.5 compared to 8.6 for the original grid (see section 6.3), where Δy_{max} is kept approximately the same such that the aspect ratio increases mainly near the solid wall. The solution at $t = 10,000$ on the original grid is interpolated onto the refined grid using a fourth order interpolation technique. Because the time step limitation for the partially implicit schemes is caused by the stability time step in x-direction an accurate simulation with approximately this time step should be possible on the refined grid as well since the grid is unchanged in x-direction and the physical properties of the flow are unaltered. Due to the specific stretching of the grid (see section 6.3) the locations where the quantities $U1$ and $U2$ are measured on the original grid do not correspond to a grid node on the refined grid. Therefore two new locations on the refined grid are taken nearest to the corresponding locations on the original grid.

As a representative of the partially implicit schemes we take the steady-state-consistent two-stage scheme. Simulations are performed for several time steps including $\Delta t = 1$ which was used for simulations on the original grid. The solution is determined with a local accuracy of $\epsilon = 10^{-2}$ each stage and it was possible to take a pseudo time step $\Delta\tau = 10^3$. In figure 8-9 we have plotted the quantities $U1$ and $U2$ for various time steps where solid lines represent the reference solution obtained with the explicit scheme on the refined grid. The quantity $U1$ is resolved very well for all time steps and the quantity $U2$ is resolved quite accurately as well even for $\Delta t = 1$ indicating that the time step is determined by the stability time step in x-direction. Additional simulations were performed with the three-stage scheme with similar results. Due to the grid refinement in y-direction the stability time step for the explicit scheme decreased with approximately a factor 16 whereas the time step could remain the same for the partially implicit scheme. The total speedup for the two-stage scheme, using $\Delta t = 1$, is approximately a factor 20. The above results show that for simulations with a high aspect ratio the partially implicit schemes perform very well. For the current test case on the refined grid the aspect ratio at the solid wall is approximately 160. For other type of flows where the



(a) U1 (b) U2

Figure 8-9: The instantaneous quantities U1 and U2 obtained with the two-stage scheme on the refined grid for various Δt with local accuracy $\epsilon = 10^{-2}$. The solid line represents the reference solid obtained with the explicit scheme.

aspect ratio is much higher, e.g. the viscous turbulent flow around an airfoil, the partially implicit schemes therefore constitute an attractive alternative to explicit schemes with respect to CPU time.

8.6 Conclusions

In this chapter we focused on the development of partially implicit second order time integration schemes. A first order partially implicit scheme was already successfully used in Ref. [6]. However, due to the first order accuracy of the scheme a large number of time steps per period of the dominant unsteady flow quantity was needed. Therefore, a higher order accuracy is required.

In order to obtain a class of second order partially implicit schemes we mixed the second order Crank-Nicolson scheme with a class of explicit Runge-Kutta schemes with an arbitrary number of stages. It was shown that the last two stages of the mixed scheme determine the second order accuracy and that preceding stages only influence the stability of the scheme. For our present flow application we split the total flux into a contribution of the flux in x- and y-direction. However, in the derivation of the partially implicit scheme no assumptions on the splitting into an explicit and implicit part are made and the application of these schemes is not restricted to the current splitting.

Next to stability properties of the numerical scheme we found that the ability of a scheme to yield a steady state solution which is independent of the time step is an important requirement. Although the four-stage scheme in section 8.4.1 is M-stable, an accurate solution can only be obtained if a very small time step is used because the scheme is not steady-state-consistent. Hence, a class of steady-state-consistent second order schemes is presented in section 8.3.2. Numerical simulations were performed with the two-, three- and four-stage scheme. A simulation with the four-stage scheme and time step $\Delta t_{stab,x}$, corresponding to the explicit stability time step in x-direction, did not yield an accurate solution of the instantaneous flow quantities although the mean-skin friction is resolved quite accurately. For the two- and three stage schemes a stable simulation with $\Delta t_{stab,x}$ was not possible which can be explained by the stability properties of these schemes outlined in section 8.4.2. All schemes yield an accurate solution for $\Delta t = 1$ which is smaller than $\Delta t_{stab,x}$ which is approximately 2.0 at a CFL-number of 1.5. To our experience this CFL-number is the upper limit for a stable numerical simulation with the present spatial discretisation. In this light we see that these schemes perform rather well and that the stability is indeed determined by the stability in the x-direction. In section 8.4.2 we observed that these schemes are not stable on the imaginary axis in the complex z-plane for $\xi = 0$. However, this need not be a restriction for practical use because the presence of viscosity (numerical or physical) contributes to the negative real part of ξ. Hence, the partially implicit schemes are stable on a part of the imaginary axis in the complex z-plane.

For the present flow application the aspect ratio of the grid was too small for the partially implicit scheme to outperform the explicit solver with respect to CPU time. However, for flows where a higher aspect ratio is required such as e.g. viscous turbulent flow around a pitching airfoil, the partially implicit schemes become interesting as was already shown in Ref. [6] with a first order partially implicit scheme. This was confirmed by the numerical results in section 8.5.4 where the aspect ratio of the grid was increased and the speed up factor compared to the explicit code was about 20. For large scale applications such as e.g. a 3D flow over a flat plate the memory capacity may become a bottleneck for a fully implicit scheme and the benefits of the partially implicit schemes with respect to the memory storage are clear. With respect to the difference in smoothing properties in the multi-grid method for the x- and y-direction in the previous chapter we observe that in case a large number of nonlinear iterations is necessary in the stages of the partially implicit scheme, the multi-grid method may accelerate the convergence process because good damping properties were observed in the y-direction in the previous chapter. Therefore we expect that for large scale flow applications with high aspect

ratio the partially implicit schemes may constitute an attractive alternative for explicit or fully implicit schemes.

Chapter 9

A dynamical time step criterion

9.1 Introduction

The development of a dynamical time stepping method for unsteady flow is a major motivation for much of the work in this thesis. With such a criterion it would be possible to efficiently integrate the Navier-Stokes equations in time. If one would adopt a hybrid method involving an explicit as well as an implicit scheme then the accuracy time step could control the selection of either one during the simulation. Given the magnitude of the accuracy time step such a switch is possible if specific convergence characteristics, as studied in chapter 6, and the efficiency of the implicit scheme are known.

In order to measure the temporal accuracy two instantaneous flow quantities at different locations were monitored. Relating the temporal error to the spatial discretisation error as in (6-23) proved to be a reliable criterion to determine the accuracy time step. However, this does not directly give us a dynamical time step criterion. A first suggestion would be to use Fourier analysis on the time signal of the instantaneous quantities possibly supplemented by monitoring more quantities. The dominant modes could then be used as a measure for the required time step. Although this appears to be an efficient and simple method the main drawback lies in the fact that a rather lengthy time signal must already be available for a reliable Fourier decomposition. Consider e.g. the time signal $U1$ in figure 6-6. The dominant mode has a period of about 400 time units which means that the proposed Fourier decomposition method can become effective only after quite a long simulation time. For flows that vary rapidly in time this argument is not valid but in these cases an explicit time integration method is probably the best choice.

In Ref. [45] a study was performed of an extended POD (Proper Orthog-

onal Decomposition) applied to a 2D temporal mixing layer. For convenience the extended POD method is referred to as POD method in the following. The temporal behavior of this flow can roughly be described as follows: initially the perturbation grows exponentially in time and vortices are formed. The changes in the time signal are relatively small in this regime. When the mixing starts the rollers undergo pairing and merge and the solution changes more rapidly in time whereas in the final stages the signal decays due to viscous effects. The time intervals in which one eigenfunction could capture $p\%$ of the energy was determined. Exactly, this feature can be extended as a method to predict the accuracy time step. Application of this idea to the temporal mixing layer resulted in relatively large time intervals in the initial phase after which the length of the time intervals decreased considerably during the strong mixing process and increased again in the viscous phase. Clearly, this agrees well with the physical behavior in time as described above and will be pursued further in this chapter.

In this chapter we show some preliminary results of the POD method applied to the boundary-layer flow described in chapter 6 which is a considerably more complicated flow than the temporal mixing layer. In addition the flow develops both in time and space. Although the temporal behavior of this flow is quite smooth and a constant time step is possible the question remains whether the POD method predicts the same smooth behavior. The contents of this chapter is as follows. We start with a brief discussion of POD in section 9.2 after which the numerical results are presented in section 9.3. Finally, conclusions are drawn and recommendations made in section 9.4.

9.2 Proper orthogonal decomposition

The POD method is used to analyze and model large data sets in a wide variety of engineering disciplines. Especially in fluid dynamics large data sets occur during numerical simulations. In this section we discuss some elementary aspects of the POD method [1]. For more details we refer to literature e.g. Refs [5, 44, 45]. The POD method used here extracts the most important spatial structures from the data. These spatial structures, which are ordered by their energy contents (see below), form a set of optimal base functions.

Consider a signal $S(x,t)$ for $x \in \mathcal{D} \subset R^n, n = 1, 2, \ldots$ and $t \in \mathcal{T} = [0, T]$. In the POD analysis the spatial structures are constructed such that the time-averaged projection of the spatial structures on the signal S is as large as

[1] The text in this section closely follows the work of W.L. IJzerman from Ref. [45].

possible. The first structure, ϕ_1, is a solution of the maximization problem:

$$\max_{\phi} \left\{ \frac{\int_{\mathcal{T}} \left[\int_{\mathcal{D}} S(x,t)\phi(x)\,dx \right]^2 dt}{T \int_{\mathcal{D}} \phi(x)^2\,dx} \right\}. \tag{9-1}$$

The second structure, ϕ_2, is taken orthogonal to the first and is characterized by the constrained optimization problem:

$$\max_{\phi} \left\{ \frac{\int_{\mathcal{T}} \left[\int_{\mathcal{D}} S(x,t)\phi(x)\,dx \right]^2 dt}{T \int_{\mathcal{D}} \phi(x)^2\,dx} \text{ s.t. } \int_{\mathcal{D}} \phi_1(x)\phi(x)\,dx = 0 \right\}. \tag{9-2}$$

Other structures are found successively in an analogous manner.

Optimization problems such as (9-1) and (9-2) result in a symmetric eigenvalue problem, in which the general structures ϕ_k are the eigenfunctions:

$$\int_{\mathcal{D}} R_s(x,x';T)\phi(x')\,dx' = \mu^2\phi(x). \tag{9-3}$$

with the spatial correlation function, R_s, defined by

$$R_s(x,x';T) = \frac{1}{T}\int_0^T S(x,t)S(x',t)\;dt \tag{9-4}$$

The square root of the eigenvalues of this problem are called the singular values and numbered in descending order: $\mu_1 \geq \mu_2 \geq \cdots > 0$. The space spanned by the eigenfunctions $\{\phi_k\}_{k=1}^{\infty}$ contains the signal $S(x,t)$, i.e:

$$S(x,t) = \sum_{k=1}^{\infty} \mu_k \psi_k(t)\phi_k(x), \tag{9-5}$$

where the temporal coefficients are given by:

$$\psi_k(t) = \frac{1}{\mu_k}\int_{\mathcal{D}} S(x,t)\phi_k(x)\,dx, \tag{9-6}$$

for $k = 1, 2, \ldots$, see [44]. Due to the factor $1/\mu_k$ these temporal coefficients are normalized, which can be shown using the eigenvalue problem (9-3).

The square of the k^{th} singular value can be interpreted as the energy represented by the k^{th} structure:

$$\mu_k^2 = \frac{1}{T}\int_{\mathcal{T}} \left[\int_{\mathcal{D}} S(x,t)\phi_k(x)\,dx \right]^2 dt \tag{9-7}$$

The sum of the squares of the first N singular values gives the amount of energy captured by the first N structures. The number of structures required

to capture a part p, for example 99%, of the total energy gives an indication of the complexity of the signal and is interpreted as the dimension, d_p, of the signal, [5]:

$$d_p = \left\{ N \mid \sum_{k=1}^{N} \mu_k^2 \geq p \int_{\mathcal{T}} \int_{\mathcal{D}} S^2(x,t) dx dt = p \sum_{k=1}^{\infty} \mu_k^2 = p E_{tot} \right\}. \tag{9-8}$$

where E_{tot} stands for the total energy.

In practice (9-3) cannot be solved numerically since the matrix for the discrete eigenvalue problem is too large; it is an $N \times N$ matrix where N equals the number of grid points used in the spatial discretisation which in our application equals 193×65. Instead of (9-3) the temporal equivalent is solved numerically:

$$\begin{aligned} &\int_{\mathcal{T}} R_t(t,t')\psi(t')dt' = \mu^2 \psi(t), \qquad \text{with} \\ &R_t(t,t') = \int_{\mathcal{D}} S(x,t)S(x,t')dx, \end{aligned} \tag{9-9}$$

where R_t denotes the temporal correlation function. The eigenfunctions, spatial structures, of (9-3) are related to the eigenfunctions ψ_k by:

$$\phi_k(x) = \frac{1}{\mu_k} \int_{\mathcal{T}} S(x,t)\psi_k(t)dt. \tag{9-10}$$

The size of the matrix for the discrete version of (9-9) depends on the number of snapshots which is required for a suitable accuracy in μ_k and in our application is less than 5000. The numerical integrations in the above formulas, (9-9) and (9-10), are performed using the trapezoidal rule.

9.3 Numerical results

The results presented in chapter 6 for the shock boundary-layer flow indicate that a constant time step of $\Delta t = 4$ resolves the instantaneous quantities accurately whereas $\Delta t = 8$ is clearly too large. In this section we explore the possibilities to use the POD method as a tool in the dynamical time step choice. The POD method should predict a nearly constant time step in between $\Delta t = 4$ and $\Delta t = 8$. Since we are interested in a time accurate simulation the small temporal structures have to be resolved as well. With respect to this point it would be better to perform a POD on the field $P(x,t) = S(x,t) - \overline{S}(x)$ where $\overline{S}$ represents the mean flow field over the sampling period T, because the current flow is statistically stationary. However, in general the mean flow field is not known which would limit the applicability of the current

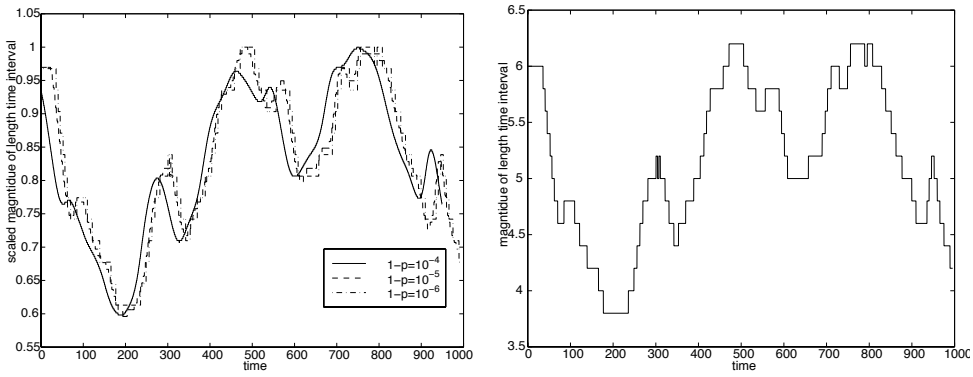
approach. Therefore, we perform the POD analysis on the flow field $S(x,t)$. Since, the mean flow contains a large fraction of the energy a larger fraction p than recommended in Ref. [45] i.e. $p > 0.99$, may be necessary.

For the data sampling we use the explicit time integration method described in section 4.2.3 with $\Delta t = 0.2$ and we store the total flow field every time step during the sampling period from $t = 10,000$ to $t = 11,000$. Although in practice, if the time step is dynamically determined, not so many fields are available, it is convenient for this exploratory research to have an abundance of samples. The determination of the interval in which the signal can be represented by one structure proceeds as follows. Let us assume that we want to resolve a fraction p of the total energy in the quantity $S(x,t)$. We start with the interval $[t_0, t_1]$, where t_0 and t_1 denote the subsequent time levels, and determine the energy contained in the first eigenfunction as in (9-7). If this is larger than the required fraction p we enlarge the interval with the next time step and again determine the energy contained in the first eigenfunction. This is repeated for $[t_0, t_j],\ j = 1, 2 \ldots$ until the signal can no longer be represented by one eigenfunction with the desired energy contents. The length of the interval so obtained may be used as a measure for the accuracy time step. If the signal varies rapidly then already after a short period the energy constraint in (9-8) will be violated and for mildly varying signals this violation will take much longer. This is reflected directly in the lengths of the intervals and thus can be used to govern the values of Δt. As we will show this length depends sensitively on the value of p.

First we take the pressure field as the signal $S(x,t)$ to be used in the POD analysis. In order to obtain a smoother graph the length of the interval in which only one eigenfunction captures a fraction p of the energy is determined for every sample field, which means that this field is considered as the initial field at t_0. In figure 9-1(a) the results are presented for three different values of $(1-p)$: $10^{-4}, 10^{-5}$ and 10^{-6}. Although the magnitudes of the sub-intervals for different fractions p differ significantly it appears that there is a large similarity between the POD signals for different values of p. Taking the quotient of the scaling factors we obtain $66.2/19.8 \approx 3.3$ and $19.8/6.2 \approx 3.2$ which indicates some kind of scaling law. With some tedious work involving a Taylor expansion around $t = t_0$ of $S(x,t)$, μ and ϕ in Δt one can show that

$$(1 - p) \propto (\Delta t)^2 \tag{9-11}$$

The derivation of this property can be found in [18] but exceeds the current purpose. With respect to the current choices of $(1-p)$ a factor $\sqrt{10}$ is predicted by (9-11) which agrees well with the numerical results, i.e. the values 3.3 and 3.2, indicating that the time steps are still within the linear region. This scaling property can also be used to make the POD method more efficient.



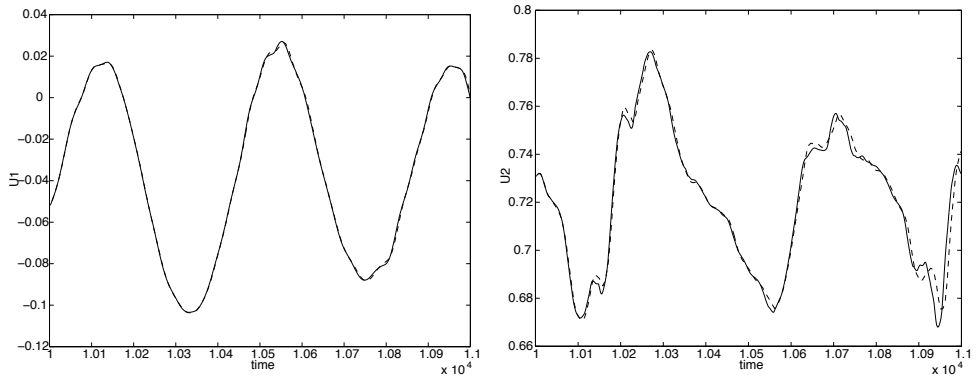
(a) Scaled intervals (b) $1 - p = 10^{-6}$

Figure 9-1: The length of the sub-intervals in (a) for three values of p scaled by 66.2, 19.8 and 6.2 respectively and the unscaled length for $1 - p = 10^{-6}$ in (b).

Assume we know the desired value of p say p^*. For small $(1 - p)$ less work is required to calculate the length of the time interval since it involves less data. The length of this interval can then be scaled to the corresponding length of the interval for $(1 - p^*)$ using the scaling law in (9-11).

Clearly, the value of p plays an important role in the final choice of the time step (sub-interval length). As mentioned above the predicted value of the time step should be somewhere between $\Delta t = 4$ and $\Delta t = 8$. In figure 9-1(b) the choice of $(1 - p) = 10^{-6}$ yields a proper estimate of the magnitude of the time step. In general, however, it is not obvious how to choose the value of p and this is a topic of current research. If the velocity field is used as the signal $S(x, t)$ this leads to the same conclusions.

The time step sequence following from $(1 - p) = 10^{-6}$ depicted in figure 9-1(b) is used in a simulation with the fully implicit Crank-Nicolson scheme. The numerical parameters are set equal to the test case in section 7.2, i.e. $\Delta\tau = 1.0$ and $\epsilon = 10^{-3}$. We observe that the results of the mean skin friction are indistinguishable from the reference solution which is in agreement with the results in chapter 6 and chapter 8. The results of this simulation with non-uniform time step for $U1$ and $U2$ are shown in figure 9-2. Clearly, $U1$ is captured very well and the non-uniformity in the time step appears to have no effect on the accuracy. The largest deviations in $U2$ in time are similar to the results shown in figure 7-1 with a uniform time step of $\Delta t = 4$. In total the temporal behavior is captured rather well and we observe that the time



(a) U1 (b) U2

Figure 9-2: The instantaneous solutions U1 and U2 as a function of time obtained with the Crank-Nicolson scheme with non-uniform time step (dashed) and explicit reference solution (solid).

sequence is based on a "non-optimal" value of p. Increasing the value of p, thus decreasing the time step, will result in a higher accuracy. However, these first results are very promising and further research of the use of POD for a dynamic time step criterion seems worthwhile.

Recently we have found a way to efficiently determine the accuracy time step with a minimum of memory storage and computational effort. Previously, we determined the length of the time interval in which only one eigenfunction contains a fraction p of the total energy. To achieve this we used a sufficiently large sequence of the signal field $S(x,t)$ in time. This requires a large computational overhead for complex three dimensional applications as explained above. However, there appears to be an attractive alternative to the suggested extended POD method which avoids this large overhead. In this method we define a new measure for the total energy given by the scaled correlation

$$\widehat{p} = \frac{|\left\langle S^n, S^{n+1} \right\rangle|}{||S^n||\ ||S^{n+1}||} \tag{9-12}$$

where $\widehat{p}$ denotes the energy, the superscript denotes the time level and the innerproduct is defined as

$$\langle f, g \rangle = \int_D f g \ dx \tag{9-13}$$

From Schwarz' inequality it follows that $0 \leq \widehat{p} \leq 1$. By means of a Taylor

expansion around time level t^n we obtain again

$$(1 - \widehat{p}\,) \propto (\Delta t)^2 \tag{9-14}$$

similar to the scaling law of the extended POD method in (9-11). A comparison of the Taylor expansions of the two scaling formulas reveals that the factor in front of Δt^2 differs only by a factor of 6. In this way we can relate the new method to the extended POD method by taking this factor into account. To test the new method we dynamically determine the time step whereas earlier we determined the time series a posteriori with a large data set (see figure 9-1). All numerical parameters are chosen the same as in the previous case which means that for the desired value of $(1 - p)$ we take 10^{-6}. Since we need only two signal fields to determine $\widehat{p}$ we use the scaling law in (9-14) to determine the time step corresponding to the desired energy level. In figure

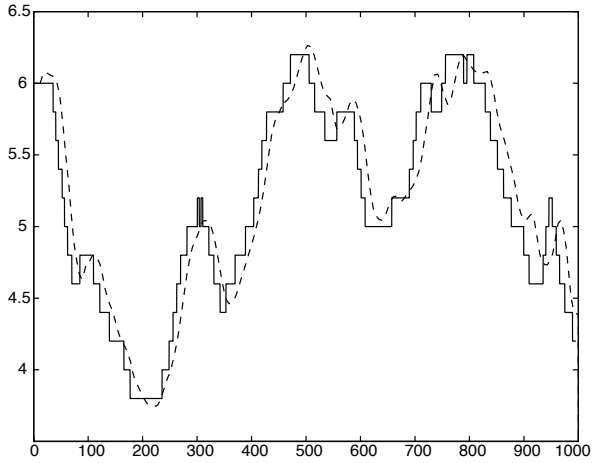


Figure 9-3: Magnitude of the accuracy time step determined a posteriori by the extended POD method (solid) and a priori by the correlation method (dashed).

9-3 the time step sequences for the extend POD method and the correlation method are plotted. Clearly, the dynamic time step sequence captures the dynamical behavior predicted by the extended POD method. There appears to be a small time lag for the dynamically determined time sequence. This can be explained by the fact that the time step in the dynamic case is determined by the history of the time signal. However, we observe that the time lag is minimal since only the last two signals in time are used together with the scaling law. The solution obtained with this dynamic approach has an acceptable accuracy comparable to the solution presented in figure 9-2. We

conclude that this new approach avoids the large overhead of the extended POD method. The computational costs are negligible and no extra memory storage is required.

9.4 Conclusions

In this chapter some preliminary results were presented showing the use of the POD method to dynamically determine the accuracy time step. The shock boundary layer flow of chapter 6 is used as a test case. In order for the POD method to be useful in flows with a more complicated temporal behavior the predicted dynamic time step should lie in the vicinity of the accuracy time step determined in chapter 6. The amount of energy to be contained in the first eigenfunction plays an important role. For the specific choice of $(1 - p) = 10^{-6}$ the predicted time step lies well within the desired region. Further numerical experiments confirm the scaling property between $(1-p)$ and Δt in (9-11) which indicates that the time steps are still within the linear region. A numerical simulation with the Crank-Nicolson scheme using a time sequence following from $(1-p) = 10^{-6}$ yields solutions with acceptable accuracy which emphasizes the promising use of POD for dynamically determining an accuracy time step.

However, at this point we would like to make the following remarks on the current approach. First we note that the value of p has a strong influence on the length of the sub-intervals in which a fraction p of the energy is contained in only one eigenfunction. In general it is not yet clear how to determine an appropriate value of p. Second the POD method was applied to data which was already created which a sufficiently small time step. The dynamic use of the POD during a simulation with a relatively large time step has not yet been performed.

With respect to the computational work we remark that the amount of work required to calculate the principle eigenvalue is about TNM where T stands for the number of samples in a time interval and N, M represent the number of grid points in x and y direction respectively. For large N and M this may become very costly. An obvious solution to this problem is to reduce the number of data points used in the POD method. In general this can be achieved e.g. using only the data from every other grid point or to take a specific region of the computational domain into account if at fore hand some flow characteristics are known. A POD analysis on a data set reduced up to a factor 8 in both directions still gave results rather similar to the results for the total data set indicating that this approach is quite useful and promising.

Finally, we emphasize a feature of the scaling property in (9-11). Assume we know the desired value of p say p^*. Computationally it is more efficient to

calculate the length of the time interval for small $(1-p)$ since it involves less data. The length of this interval can then be scaled to the corresponding length of the interval for $(1-p^*)$ using the scaling law in (9-11). The significance of this scaling property is emphasized in the use of the highly efficient correlation method (9-12). This method does not have the overhead of the extended POD method as described above. The computational costs are negligible and no additional memory storage is required because it only uses the signal fields of the last two time steps. However, in general the corresponding energy level of $\widetilde{p}$ is not equal to the desired energy level p^*. Application of the scaling law yields the accuracy time step corresponding to p^*.

Chapter 10

Conclusions

In this chapter we discuss to what extent the research aims defined in section 1.3 are achieved. Each chapter finishes with extensive conclusions and we refer to the corresponding sections for more details. The numbers of the following items correspond to the research aims described in section 1.3.

1. We find that for steady flows containing shock waves the Euler backward scheme in combination with the first order Roe approximation of the flux Jacobi matrix method and a basic linear solver such as the Gauss-Seidel method form a suitable implicit method. The Crank-Nicolson scheme in combination with the previously mentioned implicit scheme in pseudo time yields a suitable numerical method for unsteady flow applications.

2. A robust multigrid method is obtained if an implicit scheme is used as a smoother to obtain the steady state of a transonic flow around an airfoil.

3. By means of a domain decomposition an acceptable parallelization level is obtained for the implicit scheme.

4. In chapter 7 we showed that no multigrid acceleration can be obtained if the terms related to the time integration dominate the smoothing operator. Therefore, we conclude that multigrid results obtained for steady flow are not directly transferable to unsteady flow in general.

5. For unsteady flow without an external unsteady flow condition we find that the global spatial discretisation error is a good measure of reference for the allowed temporal error.

6. Simulations are performed in section 6.5 for various magnitudes of the time step and local accuracies which illustrate the relation between Δt and ϵ_{nonlin}.

7. It appears that an extended POD analysis is an appropriate tool to dynamically determine the accuracy time step. With this method the maximum length of a time interval such that a fraction p of the total energy is contained in only one spatial structure is determined. The length of such an interval is a measure for the temporal behavior of the flow and can be used to predict the accuracy time step.

In addition to item 7 we observe the following. In chapter 9 some preliminary results were presented on the use of POD as a tool to determine the accuracy time step. However, further research is required. One of the remaining questions is the determination of a proper value of p. The value of p has a significant influence on the resulting time step. However, at this point it is not clear how to determine an appropriate value of p. With respect to efficiency we remark that the computational costs of the POD analysis can be significantly decreased in two ways. First, a reduction of the data set in the POD analysis, leaving only 2% of the data, still yielded quite similar results which gives confidence on the practical use of this method for 3D simulations. Second, the scaling property between $(1-p)$ and Δt can be exploited to increase the efficiency.

Finally, we observe that the moderate improvements with respect to required CPU time of the implicit and partially implicit schemes compared to the explicit reference method are related to the present unsteady test case. The stability time step in x-direction is almost equal to the accuracy time step for instantaneous quantities determined in chapter 6. Since the aspect ratio is rather small, i.e. 16.5, no sensational speedup can be expected especially because of the higher computational costs of implicit schemes. Also, the possibilities of the partially implicit schemes described in chapter 8, which are typically suited for flow problems with high aspect ratio, do not come out well. In this sense the present test case constitutes a worst case and for more suitable flow problems significant improvements with respect to CPU can be expected.

Bibliography

[1] L.V. Ahlfors. *Complex Analysis*. Mathematical Series. McGraw-Hill, third edition, 1979.

[2] A. Arnone, M. Liou, and L.A. Povinelli. Multigrid time-accurate integration of Navier-Stokes equations. AIAA-paper 93-3361-CP, 1993.

[3] A. Arnone, Pacciani R., and A. Sestini. Multigrid computations of unsteady rotor-stator interaction using the Navier-Stokes equations. *J.Fluids.Eng.*, 117:647–652, 1995.

[4] K.E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley and Sons, second edition, 1988.

[5] N. Aubry, R. Guyonnet, and R. Lima. Spatiotemporal analysis of complex signals: theory and application. *J.Stat.Phys.*, 64(3/4):683–739, 1991.

[6] K.J. Badcock. A partially implicit method for simulating viscous aerofoil flows. *Int.J.Num.Meth.Fluids*, 19:259–268, 1994.

[7] J. Bardina and C.K. Lombard. Three dimensional hypersonic flow simulations with the CSCM implicit upwind Navier-Stokes method. *AIAA paper*, 87-1114-CP, 1987.

[8] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994.

[9] P. Bartello and S.J. Thomas. The cost-effectiveness of semi-Lagrangian advection. *Monthly Weather review*, 124(12):2883–2897, 1996.

[10] A. Belov, L. Martinelli, and A. Jameson. A new implicit algorithm with multigrid for unsteady incompressible flow calculations. *AIAA paper*, 95-0049, 1995.

[11] E.R. Benton and G.W. Platzman. A table of solutions of the one-dimensional Burgers equation. *Quarterly of Applied Mathematics*, pages 195–212, 1972.

[12] W. Blumen. Shear layer instability of an inviscid compressible fluid. *J.Fluid.Mech.*, 40:769–781, 1970.

[13] A. Brandt. Multi-level adaptive technique for fast numerical solution to the boundary value problem. In H. Cabannes and R. Temam, editors, *Proceedings 3rd International Conference on Numerical Methods in Fluid Mechanics*, volume 1 of *Lecture Notes in Physics*, pages 82–89. Springer Verlag, 1973.

[14] A. Brandt. Barriers to achieving textbook multigrid efficiency in CFD. Technical Report NASA/CR-1998-207647, ICASE, 1998.

[15] J.W. van der Burg. *Numerical methods for transonic flow calculations*. PhD thesis, University of Twente, The Netherlands, 1992.

[16] J.W. van der Burg, J.G.M. Kuerten, and P.J. Zandbergen. Multigrid and Runge-Kutta time stepping applied to the uniformly non-oscillatory scheme for conservation laws. *J.Eng.Math.*, 14:361–370, 1995.

[17] J.W. van der Burg and B. Oskam. FASTFLO - automatic CFD system for three-dimensional flow simulations. In *7th Conference on Industrial Technologies and 3rd Aero Days, Toulouse*, 1997.

[18] R. van Buuren, W.L. IJzerman, and B.J. Geurts. A dynamic time-step method for accuracte simulation of unsteady flow based on temporal correlation. *In preparation*, 1999.

[19] R. van Buuren, J.G.M. Kuerten, and B.J. Geurts. Instabilities of stationary flow. In *EUROMECH 3rd European Fluid Mechanics Conference, Göttingen, Germany, Sept. 1997*, 1997.

[20] R. van Buuren, J.G.M Kuerten, and B.J. Geurts. Instabilities of stationary inviscid compressible flow around an airfoil. *J.Comput.Phys.*, 138:520–539, 1997.

[21] R. van Buuren, B. Wasistho, J.G.M. Kuerten, B.J. Geurts, and P.J. Zandbergen. Time accurate simulations of supersonic flow. In Charles-Henri Bruneau, editor, *Sixteenth International Conference on Numerical Methods in Fluid Dynamics, Proceedings, Arcachon, France*, Lecture Notes in Physics, pages 326–331. Springer, 1998.

[22] R.van Buuren, J.G.M Kuerten, and B.J. Geurts. Time accurate simulation of unsteady flow. *submitted*, 1999.

[23] A.J. Chorin. A numerical method for solving incompressible viscous flow problems. *J.Comput.Phys.*, 2:275, 1967.

[24] P.I. Crumpton and M.B. Giles. Implicit time-accurate solutions on unstructured dynamic grids. *Int.J.Num.Meth.Fluids*, 25:1285–1300, 1997.

[25] L.D. Dailey and H. Pletcher. Evaluation of multigrid acceleration for preconditioned time-accurate Navier-Stokes algorithms. *Computers & Fluids*, 25(8):791–811, 1996.

[26] C.H. Driesen. *Simulation of convection-driven wet-chemical etching.* PhD thesis, University of Twente, The Netherlands, 1999.

[27] R. Enander. Implicit explicit residual smoothing for the multidimensional Euler and Navier-Stokes equations. *SIAM J.Sci.Comput.*, 18(5):1243–1254, 1997.

[28] L.E. Eriksson and A. Rizzi. Computer-aided analysis of the convergence to steady state of discrete approximations to the Euler equations. *J.Comput.Phys.*, 57:90–128, 1985.

[29] V. Faber and T. Manteuffel. Necessary and sufficient conditions for the existence of a conjugate gradient method. *SIAM J.Numer.Anal.*, 21:356–362, 1984.

[30] U. Frisch and S.A. Orszag. Turbulence: Challenges for theory and experiment. *Physics Today*, pages 24–32, January 1990.

[31] J.E. Fromm. A method for reducing dispersion in convective difference schemes. *J.Comput.Phys.*, 3:176–189, 1968.

[32] B.J. Geurts, R. van Buuren, and H. Lu. Application of polynomial preconditioners to conservation laws. *submitted*, 1999.

[33] B.J. Geurts and H. Kuerten. Numerical aspects of a block structured compressible flow solver. *J.Eng.Math.*, 27:293–307, 1993.

[34] B.J. Geurts, A.W. Vreman, J.G.M. Kuerten, and R. van Buuren. Noncommuting filters and dynamic modeling for LES of turbulent compressible flow in 3d shear layers. In *Direct and Large-Eddy simulation II, Grenoble*, pages 47–56, 1997.

[35] E. Godlewski and P.A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservations Laws*, volume 118 of *Applied Mathematical Sciences*. Springer, 1996.

[36] S.K. Godunov. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Math. Sbornik*, 47:357–393, 1959.

[37] G.H. Golub and C.F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, second edition, 1989.

[38] W. Hackbusch. *Multi-grid methods and Applications*. Springer series in computational mathematics. Springer-Verlag, Heidelberg, 1985.

[39] A. Harten. High resolution schemes for hyperbolic conservations laws. *J.Comput.Phys*, 49:357–393, 1983.

[40] A. Harten and S. Osher. Uniformly high-order accurate non-oscillatory schemes I. *SIAM J.Numer.Anal.*, 24(2):279–309, 1987.

[41] R.S. Heeg. *Stability and transition of attachement-line flow*. PhD thesis, Universisty of Twente, The Netherlands, 1998.

[42] P.W. Hemker and S.P. Spekreijse. Multiple grids and Osher's scheme for the efficient solution of the steady Euler equations. *Applied Numerical Mathematics*, 2:475–493, 1986.

[43] C. Hirsch. *Computational Methods for Inviscid and Viscous Flows*, volume 2 of *Numerical Computation of Internal and External Flows*. John Wiley and Sons, 1990.

[44] P. Holmes, J.L. Lumley, and G. Berkooz. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, 1996.

[45] W.L. IJzerman, E. van Groessen, and B.J. Geurts. Modified POD for flows with non-stationary field. *Submitted, January*, 1999.

[46] E. Issman, G. Degrez, and H. Deconinck. Implicit upwind residual-distribution Euler and Navier-Stokes solver on unstructured meshes. *AIAA Journal*, 34(10):2021–2028, 1996.

[47] A. Jameson. Transonic flow calculation. MAE Report 1651, Princeton University, 1983.

[48] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA-paper 91-1596*, 1991.

[49] A. Jameson and T.J. Baker. Multigrid solution of the Euler equations for aircraft configurations. Technical report, AAIA-paper 84-0093, 1984.

[50] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods with Runge-Kutta time stepping schemes. *AIAA-paper*, 81-1259, 1981.

[51] A.N. Kolmogorov. *Dokl.Akad.Nauk. SSSR*, 30:299, 1941.

[52] H. Kuerten and B.J. Geurts. Multigrid acceleration of a block structured compressible flow solver. *J.Eng.Math.*, 29:11–32, 1995.

[53] J.G.M. Kuerten, B.J. Geurts, and R. van Buuren. Dynamic inverse modelling in LES of the temporal mixing layer,. In *Second AFOSR International Conference on Direct Numerical Simulation and Large Eddy Simulation, Newark USA*, 1999.

[54] M.-H. Lallemand and B. Koren. Iterative defect correction and multigrid accelerated explicit time stepping schemes for the steady state euler equations. *SIAM Journal of Sci.Comp.*, 14(4):953–970, 1993.

[55] P.D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math.*, 13:217–237, 1960.

[56] B. van Leer. Towards the ultimate conservative difference scheme II. monotonicity and conservation combined in a second-order scheme. *J.Comput.Phys.*, 29:11–31, 1974.

[57] R. LeVeque. *Numerical Methods for Conservation Laws*. Lectures in Mathematics, ETH Zürich. Birkhäuser Verlag, 1992.

[58] C. Liu, X. Zheng, and C.H. Sung. Preconditioned multigrid methods for unsteady incompressible flows. *J.Comput.Phys.*, 139:35–57, 1998.

[59] F. Liu and S. Ji. Unsteady flow calculations with a multigrid Navier-Stokes method. *AIAA Journal*, 34(10):2047–2053, 1996.

[60] R.W. MacCormack. The effect of viscosity in hypervelocity impact cratering. *AIAA paper*, page 69, 1969.

[61] L.M. Mack. Boundary layer stability theory. Special Course on Stability and Transition of Laminar Flow, AGARD CP-551, 1984.

[62] M.L. Merriam. An entropy based approach to nonlinear stability. NASA Technical Memorandum 101086, NASA Ames, 1989.

[63] P. Moin and J. Kim. Numerical investigation of turbulent channel flow. *J.Fluid.Mech.*, 118:341–377, 1982.

[64] P. Moin and J. Kim. Tackling turbulence with supercomputers. *Scientific American*, 276(1):62–68, 1997.

[65] W.A. Mulder. Multigrid relaxation for the Euler equations. *J.Comput.Phys.*, 60:235–252, 1985.

[66] S. Osher and Tadmor E. On the convergence of difference approximations to scalar conservation laws. *Math.Comput.*, 50:19–51, 1988.

[67] T.S. Park and J.H. Kwon. An improved multistage time stepping for second order upwind TVD schemes. *Computers & Fluids*, 25(7):629–645, 1996.

[68] R. Peyret and T.D. Taylor. *Computational methods for fluid flow*. Computational Physics. Springer-Verlag, 1986.

[69] N.A. Pierce and J.J. Alonso. Efficient computation of unsteady viscous flows by an implicit preconditioned multgrid method. *AIAA Journal*, 36(3), 1998.

[70] T.J. Poinsot and S.K. Lele. Boundary conditions for direct simulations of compressible viscous flow. *J.Comput.Phys.*, 101:104–129, 1992.

[71] M.M. Rai and P. Moin. Direct numerical simulation of transition and turbulence in a spatially evolving boundary layer. *J.Comput.Phys.*, 109:169–192, 1993.

[72] R.D. Richtmyer and K.W. Morton. *Difference Methods for Initial Value Problems*. Wiley-Interscience, New-York, second edition, 1967 (repr.1994 Kreiger).

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

[74] P.L. Roe. Characteristic-based schemes for the Euler equations. *Ann.Rev.Fluid Mech.*, 18:337–365, 1986.

[75] C.L. Rumsey, M.D. Sanetrik, R.T. Biedron, N.D. Melson, and E.B. Parlette. Efficiency and accuracy of time-accurate turbulent Navier-Stokes computations. *Computers & Fluids*, 25(2):217–236, 1996.

[76] Y. Saad and M.H. Schulz. GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J.Sci.Stat.Comp.*, 7:856–869, 1985.

[77] G.A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservations laws. *J.Comput.Phys.*, 27:1–31, 1978.

[78] T. Sonar. Entropy production in second-order three-point schemes. *Numer.Math.*, 62:371–390, 1992.

[79] A.J. van der Steen. *Aspects of computational science.* National Computing Facilities Foundation, NCF, first edition, 1995.

[80] P. Strating and R. van Buuren. Fast computation of stationary inviscid flow around an airfoil. *J.Eng.Math*, 35(4):437–453, 1999.

[81] M. Streng, J. Broeze, H. Kuerten, and B. Geurts. De (on)berekenbaarheid van turbulentie. *Nederlands tijdschrift voor Natuurkunde*, 62:23–26, 1996.

[82] M. Streng, H. Kuerten, and B. Geurts. Parallel algorithms for DNS of compressible flows. In *Progress and Challenges in CFD Methods and Algorithms*, AGARD-CP-578, 1995.

[83] P.K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J.Numer.Anal.*, 21(5):995–1011, 1984.

[84] K.H. Tan. *Local Coupling in Domain Decomposition.* PhD thesis, University of Utrecht, The Netherlands, 1995.

[85] H. Tennekes and J.L. Lumley. *A first course in turbulence.* MIT Press, 1972.

[86] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics, a practical introduction.* Springer-Verlag, 1997.

[87] E. Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. *J.Comput.Phys.*, 72:277, 1987.

[88] J.M. Varah. On the solution of block-tridiagonal systems arising from certain finite-difference equations. *Math.of Comp.*, 26:859–868, 1972.

[89] V. Venkatakrishnan. Preconditioned conjugate gradient methods for the compressible Navier-Stokes equations. *AIAA Journal*, 29(7):1092–1100, 1991.

[90] V. Venkatakrishnan. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *J.Comput.Phys.*, 118:120–130, 1995.

[91] H.A. van der Vorst and K. Dekker. Conjugate gradient type methods and preconditioning. *J.Comput.Appl.Math.*, 24:73–87, 1988.

[92] H.A. van der Vorst and C. Vuik. The superlinear convergence behaviour of GMRES. *J.Comput.Appl.Math.*, 48:327–341, 1993.

[93] A.W. Vreman, B.J. Geurts, and J.G.M. Kuerten. Comparison of numerical schemes in large eddy simulation of the temporal mixing layer. *Int.J.Num.Meth. in Fluids*, 22:297–311, 1996.

[94] B. Vreman, H. Kuerten, and B. Geurts. Shocks in direct numerical simulation of the confined three-dimensional mixing layer. *Phys. Fluids*, 7:2105–2107, 1995.

[95] B. Wasistho. *Spatial direct numerical simulation of compressible boundary layer flow.* PhD thesis, University of Twente, The Netherlands, 1997.

[96] B. Wasistho, B.J. Geurts, and J.G.M Kuerten. Spatial simulation techniques for time-dependent compressible flow over a flat plate. *Computers & Fluids*, 26:713–739, 1997.

[97] P. Wesseling. *An Introduction to Multigrid Methods.* John Wiley & Sons, 1992.

[98] H.C. Yee. Construction of explicit and implicit shock capturing methods. *Journal of Computational Physics*, 68:151–179, 1987.

[99] H.C. Yee. *Computational Fluid Dynamics.* Lecture Series 1989-04. Von Karman Institute for Fluid Dynamics, 1989.

[100] H.C. Yee and P.K. Sweby. Global asymptotic behavior of iterative implicit schemes. *Int.J.Bif.Chaos.*, 4(6):1579–1611, 1994.

[101] H.C. Yee and P.K. Sweby. Dynamical approach study of spurious steady state numerical solutions of nonlinear differential equations II. global asymptotic behaviour of time discretizations. *Comp.Fluid Dyn.*, 4:219–283, 1995.

[102] S. Yoon and A. Jameson. Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations. *AIAA Journal*, 26(9):1025–1026, 1988.

[103] H. Yoshihara and P. Sacher. Test cases for inviscid flow field methods. Technical report, AGARD-AR-211, 1985.

[104] P.J. Zandbergen and D. Dijkstra. The selection of an explicit Navier-Stokes solver for compressible flows. Technical Report ISNaS 88.09.030, University of Twente, 1988.

[105] E. Zauderer. *Partial Differential equations of Applied Mathematics*. John Wiley & Sons Inc., second edition, 1989.

Summary

This thesis deals with implicit time integration methods for compressible flow containing shock waves. Next to the question about a suitable implicit time integration scheme for steady and unsteady flow the key questions concern suitable accuracy criteria on the time step and a dynamical determination of the accuracy time step during a simulation.

In order to study the influence of the linear solver and the required degree of approximation of the flux Jacobi matrix the unsteady shock tube problem was simulated in chapter 3 using the Crank-Nicolson scheme for the time integration. Simulations revealed that the required accuracy with which the linear system arising at every pseudo time had to be solved in order to obtain a minimum number of outer iterations was only about three or four decades corresponding to very few inner iterations. Therefore, we conclude that for these types of flow basic iterative methods such as the Gauss-Seidel method are the most efficient. The influence of the approximation of the flux Jacobi matrix was also studied. We found that the additional costs of determining, storing and inverting the full flux Jacobi matrix do not outweigh the additional costs of more outer iterations in case the flux Jacobi matrix is approximated.

These results were incorporated into the implicit numerical method to simulate a transonic inviscid flow around an airfoil in chapter 4. The speedup factor obtained with the implicit scheme compared to the explicit reference method is considerable. However, there is a difference in the steady state solution obtained with both methods. With the implicit scheme we obtain a machine accurate steady state solution whereas with the explicit scheme the convergence stalls at a finite residual level. This difference was studied and we concluded that there does not exist a steady state solution. It appears that a small physical instability in the wake region is present which is not captured by the implicit scheme due to the dissipative nature of this scheme.

In chapter 5 we discussed the multigrid and parallel performance of the implicit scheme. The multigrid performance of an explicit scheme for a transonic test case is very sensitive to the multigrid parameter setting [15]. The implicit scheme proved to be a very robust smoother and multigrid acceleration was obtained without careful parameter tuning. We showed that acceptable

parallel performance is achieved by means of a domain decomposition. For some subdomain divisions convergence problems were encountered which are related to the accuracy with which the linear system arising at every pseudo time step is solved at the subdomain boundaries. By increasing the accuracy convergence is obtained for all subdomain divisions.

A two-dimensional unsteady shock boundary-layer flow was simulated in chapter 6. The second order implicit Crank-Nicolson scheme was used for the time integration. In order to solve the nonlinear system of equations arising at every time step a pseudo time derivative was introduced and the implicit method of chapters 4 and 5 was used to advance the solution towards the desired solution at the next time level. Because no external unsteady flow condition was prescribed the global error induced by the time integration over a sampling period was related to the global spatial discretisation error. A set of requirements on the global error was formulated which relate the time step, the local accuracy and the relaxation parameter. Computations show that with suitable numerical parameters the requirements are obeyed and proper solutions of the flow problem are obtained. The corresponding accuracy time step depends on the type of flow quantity. Without any code optimization a speedup factor with respect to the explicit reference method of 9.2 and 3.7 is obtained for mean and instantaneous flow quantities respectively. For large time steps convergence problems were encountered which are related to a sensitive dependence on the relaxation parameter in pseudo time. The pseudo time step appears to be a bifurcation parameter which agrees with results presented by Yee and Sweby [100, 101].

For a relevant choice of the numerical parameters the implicit code cannot outperform the explicit reference code. In order to try to accelerate the convergence in pseudo time multigrid is applied. It turns out that no convergence acceleration can be obtained. An order of magnitude analysis revealed that the terms related to the implicit time integration in physical time dominate the smoothing operator. As a result high and low frequencies in the error are damped equally and no smoothing is obtained. In order to obtain proper smoothing the terms related to the spatial derivatives should dominate the smoothing operator. This is confirmed by a study of a model equation where criteria on the time step and the pseudo time step are formulated such that the spatial derivatives dominate the smoothing operator. For the shock boundary-layer flow these criteria do not comply with the demand of sufficiently small Δt for time accuracy which explains the bad multigrid performance.

Although the overall smoothing properties are not good there appears to be sufficient smoothing in y direction. Therefore, we suggest to treat the flux in x-direction explicitly and the flux in y-direction implicitly. To this purpose a new class of second order partially implicit schemes, which are a

mixture of the second order Crank-Nicolson scheme and the compact storage Runge-Kutta schemes, is developed in chapter 8. Next to consistency and stability the ability of the partially implicit schemes to yield a steady state independent of the time step plays an important role in order to obtain an accurate solution. However, for these schemes optimal stability properties and steady-state-consistency do not go together. Simulations indicate that the latter is the most stringent requirement. Suitable stability properties are obtained if a small amount of numerical dissipation is added. For large scale applications the memory capacity may become a bottleneck and we expect that for flow applications with high aspect ratio the partially implicit schemes constitute an attractive alternative to explicit or fully implicit schemes.

Some preliminary results on a dynamical time step criterion were presented in chapter 9. An extended POD analysis is used to determine the maximum length of a time interval such that a fraction p of the total energy is contained in only one spatial structure. The length of such an interval is a measure for the temporal behavior of the flow and can be used to predict the accuracy time step. The POD method is applied to the shock boundary-layer flow which for a specific value of p resulted in a non-uniform sequence of time steps within the time accuracy bounds determined in chapter 6. Simulation with this specific time sequence yields a solution with acceptable accuracy which indicates the promising use of POD as a tool to dynamically determine the accuracy time step.

Samenvatting

Het hoofdonderwerp van dit proefschrift is de studie van impliciete tijdsintegratiemethoden voor compressibele stromingen met schokgolven. Naast de voor de hand liggende vraag wat een geschikte impliciete methode is voor stationaire en instationaire stromingen ligt de nadruk op het bepalen van geschikte nauwkeurigheidscriteria en de dynamische bepaling van de nauwkeurigheidstijdstap.

De specifieke keuze van een lineaire oplosmethode en de orde van nauwkeurigheid waarmee de Jacobimatrix van de flux wordt bepaald hebben een grote invloed op het convergentiegedrag van de numerieke methoden. Om deze invloed te bestuderen simuleren we in hoofdstuk 3 het tijdsafhankelijke schokbuis probleem met de Crank-Nicolsonmethode. Uit de berekeningen volgt dat het lineaire stelsel dat ontstaat bij elke pseudotijdstap slechts tot drie of vier decaden nauwkeurig opgelost hoeft te worden om een minimaal aantal buiteniteraties te verkrijgen. Hierdoor is slechts een beperkt aantal binneniteraties nodig zodat eenvoudige iteratieve methoden zoals de Gauss-Seidelmethode het meest efficiënt zijn. Tevens hebben we de invloed van de orde van nauwkeurigheid van de Jacobimatrix van de flux bestudeerd. De extra kosten voor het opslaan, bepalen en inverteren van een nauwkeurige Jacobimatrix staan niet in verhouding tot de extra rekenkosten van extra buiteniteraties bij een minder nauwkeurige Jacobimatrix.

Deze resultaten zijn verwerkt in een impliciete numerieke methode om een transsone wrijvingsloze stroming rond een vleugelprofiel te simuleren. Hoewel de verkregen winst in rekentijd in vergelijking met de expliciete referentiemethode aanzienlijk is, blijken de beide stationaire oplossingen van elkaar te verschillen. Met de impliciete methode wordt een oplossing tot op machineprecisie verkregen terwijl de convergentie blijft steken als de expliciete methode wordt gebruikt. De reden voor dit verschil is dat de stationaire oplossing instabiel is. Er blijkt een zwakke fysische instabiliteit in het zog aanwezig te zijn die niet wordt opgemerkt door de impliciete methode doordat deze methode een enigszins dissipatief karakter heeft.

In hoofdstuk 5 bespreken we de mogelijkheden van de impliciete methode voor versnelling met multigrid en parallellisatie. Uit eerder onderzoek is geble-

ken dat multigrid in combinatie met een expliciete methode voor het oplossen van het transsone stromingsprobleem erg gevoelig is voor de juiste multigrid instellingen. Daarentegen blijkt de combinatie van multigrid met een impliciete methode een robuuste smoother op te leveren en wordt convergentieversnelling verkregen zonder een uitgebreide parameterafstelling. We hebben laten zien dat met behulp van domeindecompositie een acceptabel parallellisatieniveau wordt bereikt. Voor sommige domeindecomposities treden er convergentieproblemen op die samenhangen met de nauwkeurigheid waarmee het lineaire stelsel iedere pseudotijdstap wordt opgelost op de domeinranden. Door de nauwkeurigheid te verhogen wordt convergentie van de oplossing verkregen voor alle mogelijke domeindecomposities.

Daarna wordt in hoofdstuk 6 een simulatie van een tweedimensionale tijdsafhankelijke schok-grenslaag-interactie stroming uitgevoerd, waarbij we het tweede orde nauwkeurige Crank-Nicolsonschema voor de tijdsintegratie gebruiken. Iedere tijdstap moet een groot gekoppeld niet-lineair stelsel opgelost worden. Hiertoe wordt een pseudotijdsafgeleide geïntroduceerd zodat met de impliciete methode uit hoofdstuk 4 en 5 de oplossing in pseudotijd convergeert naar de oplossing op het volgende tijdstip. Omdat er geen externe tijdsafhankelijke stromingscondities worden voorgeschreven relateren we de globale fout, veroorzaakt door de tijdsintegratie, aan de globale plaatsdiscretisatiefout. We formuleren een aantal nauwkeurigheidscriteria die een verband leggen tussen de tijdstap, de lokale nauwkeurigheid en de relaxatieparameter. Simulaties met parameterinstellingen die aan deze criteria voldoen geven voldoende nauwkeurige oplossingen. De nauwkeurigheidstijdstap hangt af van de stromingsgrootheid. Zonder enige vorm van code-optimalisatie wordt een versnellingsfactor van 9.2 en 3.7 verkregen in vergelijking met de expliciete referentiemethode voor respectievelijk tijdsafhankelijke en gemiddelde stromingsgrootheden. Voor grote tijdstappen treden convergentieproblemen op die samenhangen met een gevoelige afhankelijkheid van de relaxatieparameter. Deze parameter lijkt een bifurcatieparameter te zijn overeenkomstig de bevindingen van Yee en Sweby [100, 101].

Voor een relevante keuze van de numerieke parameters levert de impliciete methode geen rekentijdwinst op. Om de convergentie in pseudotijd te versnellen passen we multigrid toe. Het blijkt dat geen versnelling verkregen kan worden. Door middel van een orde van grootte beschouwing wordt duidelijk dat de tijdsintegratietermen de smoothing-operator domineren. Om voldoende smoothing te verkrijgen zouden de fluxtermen de smoothing-operator moeten domineren. Dit wordt bevestigd door een modelstudie waarin criteria voor de grootte van tijdstap en de relaxatieparameter worden geformuleerd zodanig dat de fluxtermen de smoothing-operator domineren. Door de opgelegde nauwkeurigheidseis bij de schok-grenslaagstroming is de nauwkeurigheidstijd-

stap zo klein dat niet aan deze criteria wordt voldaan. Dit verklaart het negatieve multigridresultaat.

Hoewel de totale smoothingseigenschap van de numerieke methode slecht is, blijkt er wel enige smoothing in de y-richting te zijn. Daarom stellen we voor om voor de flux in x-richting een expliciete methode te gebruiken en voor de flux in y-richting een impliciete methode. Hiervoor hebben we in hoofdstuk 8 een nieuwe klasse van partieel-impliciete methoden ontwikkeld die een combinatie zijn van het tweede orde Crank-Nicolsonschema met de compact-storage Runge-Kuttaschema's. Het feit dat een numeriek schema een stationaire oplossing moet kunnen genereren die onafhankelijk is van de tijdstap is een belangrijke eis naast de traditionele eisen van consistentie en stabiliteit. Deze eis kan echter niet gecombineerd worden met optimale stabiliteitseigenschappen van het numerieke schema. Simulaties laten zien dat de eis van stationariteit het meest stringent is. Door een klein beetje numerieke dissipatie toe te voegen hebben de partieel- impliciete methoden toch geschikte stabiliteitseigenschappen. Voor grootschalige stromingsproblemen waarin het geheugen een bottleneck vormt of voor rekenroosters met hoge aspect ratio verwachten we dat de partieel-impliciete methoden een goed alternatief zijn voor expliciete of volledig impliciete methoden.

Tenslotte worden in hoofdstuk 9 enkele voorlopige resultaten gepresenteerd van de dynamische tijdstapbepaling. Hiertoe gebruiken we een aangepaste POD-analyse. In deze analyse bepalen we het maximale tijdsinterval waarin een fractie p van de energie bevat is in slechts één ruimtelijke structuur. Dit interval is een maat voor het tijdsgedrag van de stroming en kan daarom gebruikt worden als nauwkeurigheidstijdstap. De methode is toegepast op de tijdsafhankelijke schok-grenslaagstroming. Voor een specifieke waarde van p resulteert dit in een niet uniforme tijdstap binnen de nauwkeurigheidsgrenzen bepaald in hoofdstuk 6. Een simulatie met deze tijdreeks resulteert in een voldoende nauwkeurige oplossing wat de mogelijkheden van de POD-methode om dynamisch de nauwkeurigheidstijdstap te bepalen illustreert.

Dankwoord

Na vier turbulente jaren op zowel promotie als persoonlijk gebied is er een einde gekomen aan mijn periode als AIO. Het is dan ook een goed moment om enkele mensen te bedanken voor hun directe of indirecte bijdrage aan dit proefschrift.

Allereerst wil ik natuurlijk Prof. Zandbergen bedanken voor de mogelijkheid die hij mij heeft geboden om aan deze promotie te kunnen beginnen. In het bijzonder wil ik mijn dagelijkse begeleiders, Hans Kuerten en Bernard Geurts, bedanken zonder wie dit proefschrift niet tot stand had kunnen komen. Als klein detail zou ik jullie willen bedanken voor het feit dat concepten altijd met een blauwe pen zijn gecorrigeerd. Gelukkig heeft ons contact zich niet beperkt tot werk alleen. Naast de gezellige momenten buiten het werk om wil ik jullie vooral bedanken voor de ruimte die jullie me hebben gegeven om even op adem te komen. Hans, je enorme geduld en positieve houding die je altijd hebt gehad ten opzichte van mijn werk zijn op de kritieke momenten van doorslaggevend belang geweest. Jouw continue aandringen op details heeft mij een stuk kritischer gemaakt. Bernard, regelmatig hebben we lange tijd op je kamer doorgebracht met discussies over dit werk maar ook over andere zaken. Ik heb veel plezier beleeft aan onze 'ziener'-discussies, die zeker hebben geleid tot een bredere kijk op zaken. Verder wil ik graag Prof. Dick bedanken voor zijn belangrijke suggesties, die mij meerdere malen uit een impasse heeft gehaald. Tevens wil ik Patrick Strating bedanken voor zijn niet geringe bijdrage aan hoofdstuk 5 en Wilbert IJzerman voor zijn samenwerking in de totstandkoming van hoofdstuk 9. Als laatste wil ik Ruud van Damme en Brian Gilding bedanken voor het feit dat ik altijd bij hun terecht kon met detaillistische vragen over wiskunde.

Naast de inhoudelijke ondersteuning heb ik de afgelopen vier jaren veel steun gehad van familie en vrienden. Het zou te ver gaan om iedereen persoonlijk te bedanken en ik hoop niet dat de mensen, die ik niet bij name noem, het gevoel hebben dat hun vriendschap of steun wordt onderschat. Als eerste wil ik graag mijn ouders, mijn broer, Mirjam en Duco vooral bedanken voor hun steun in de afgelopen twee jaar. Tevens wil ik Paul en Wijnand bedanken voor het feit dat ze op het goede moment aanwezig waren. Verder zou ik graag Robert willen bedanken voor de nodige restauratie-uren. Als laatste noem ik mij kamergenoot Carlo. We hebben veel plezier gehad de afgelopen jaren op conferenties en zeker op onze kamer. Er komt een dag dan is het antwoord:....

Over de schrijver

De auteur van dit proefschrift werd geboren op 14 april 1972 te Leeuwarden. Hij groeide op in Grouw, en in 1990 deed hij eindexamen VWO aan de Stedelijke Scholen Gemeenschap te Leeuwarden. Van 1990 tot 1995 studeerde hij Toegepaste Wiskunde aan de Universiteit Twente. Tijdens zijn studie liep hij drie maanden stage bij het Institut Laue Langevin te Grenoble, Frankrijk, waarna hij onder supervisie van Prof.dr.ir. R. Martini afstudeerde bij de vakgroep ADAM (Algebra Discrete wiskunde Analyse en Meetkunde) op het onderwerp: 'An algorithmic approach to conservation laws'. Van 1995 tot 1999 was hij aangesteld als Assistent in Opleiding aan de Universiteit Twente onder supervisie van Prof.dr.ir. Zandbergen bij de vakgroep Toegepaste Analyse. Tijdens deze aanstelling deed hij onderwijservaring op door het geven van werkcolleges op het gebied van numerieke wiskunde en analyse.