INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2001; **35**: 687–720

Implicit time accurate simulation of unsteady flow

René van Buuren[1], Hans Kuerten[2] and Bernard J. Geurts*[,3,4]

Twente Institute of Mechanics, J.M. Burgers Centre, Faculty of Mathematical Sciences, University of Twente, Enschede, Netherlands

SUMMARY

Implicit time integration was studied in the context of unsteady shock-boundary layer interaction flow. With an explicit second-order Runge–Kutta scheme, a reference solution to compare with the implicit second-order Crank–Nicolson scheme was determined. The time step in the explicit scheme is restricted by both temporal accuracy as well as stability requirements, whereas in the A-stable implicit scheme, the time step has to obey temporal resolution requirements and numerical convergence conditions. The non-linear discrete equations for each time step are solved iteratively by adding a pseudo-time derivative. The quasi-Newton approach is adopted and the linear systems that arise are approximately solved with a symmetric block Gauss–Seidel solver. As a guiding principle for properly setting numerical time integration parameters that yield an efficient time accurate capturing of the solution, the global error caused by the temporal integration is compared with the error resulting from the spatial discretization. Focus is on the sensitivity of properties of the solution in relation to the time step. Numerical simulations show that the time step needed for acceptable accuracy can be considerably larger than the explicit stability time step; typical ratios range from 20 to 80. At large time steps, convergence problems that are closely related to a highly complex structure of the basins of attraction of the iterative method may occur. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: chaos; computational fluid dynamics; direct numerical simulation; implicit time integration

1. INTRODUCTION

Present day computer capabilities are much too small to accurately resolve, for example, all turbulent scales in flows that occur in practically relevant applications [1]. Typically, such

* Correspondence to: Faculty of Mathematical Sciences, University of Twente, PO Box 217, 7500 AE Enschede, Netherlands. Tel.: +31 53 4894125; fax: +31 53 4894888.

[1] Current address: KPN Research, Drienerlolaan 5, 7522 NB Enschede, Netherlands.

[2] Current address: Faculty of Mechanical Engineering, Process Technology Group, Eindhoven University of Technology, Den Dolech 2, PO Box 513, 5600 MB Eindhoven, Netherlands.

[3] E-mail: b.j.geurts@math.utwente.nl

[4] Also at the Department of Engineering, Queen Mary and Westfield College, University of London, Mile End Road, London E1 4NS, U.K.

Received March 1999
Revised May 2000

flows are time-dependent and a direct numerical simulation (DNS) of the full Navier–Stokes equations or a large eddy simulation (LES) should be considered. The main emphasis has, however, been put on simulation of steady flows and a number of significant developments in the numerical methods have been reported, which enhance the efficiency of these steady flow calculations. Jameson [2] suggested that it is possible to apply the methods developed for steady state computations to unsteady flow problems as well. By adding a pseudo-time derivative to the set of algebraic equations that result after discretization of the temporal and spatial derivatives, the solution at the next time level corresponds to a steady solution of the pseudo-time problem.

In the past a lot of work has been done to improve numerical techniques for steady flow problems with respect to accuracy and computational efficiency. Examples of these developments are multi-grid methods for steady flows, total variation diminishing (TVD) schemes for flows with shocks, parallel computing, higher-order spatial discretization schemes, implicit methods, etc. In the context of unsteady flow simulations using a pseudo-time formulation, a number of these methods and approaches need to be re-evaluated, as the numerical features of the unsteady problem usually differ considerably from those desired in relation to steady state calculations. As an example, rapid multi-grid convergence is more easily achieved in combination with spatial discretization methods that add sufficient (artificial) dissipation. Such may be acceptable for steady flow calculations in the Reynolds-averaged Navier–Stokes setting (RANS) but it is not suitable for DNS or LES. In these cases, one is interested in retaining the properties of the smaller unsteady structures in the flow and correspondingly turns to spatial discretizations, which add only a minimal amount of dissipation [3]. A central role is played by the temporal integration method and in particular by the value of the numerical time step. On the one hand, one would like this parameter to be as large as possible to proceed quickly with the evolution of the solution and thereby enhance efficiency. On the other hand, the time accuracy of the solution demands a sufficiently small time step. A major aspect in any unsteady simulation is hence the sensitivity of the flow predictions on the size of the time step. This obviously depends on the specific flow phenomena that one is simulating and monitoring as well as on the properties of the numerical method adopted for the simulations. In particular, the global error propagation that arises in a flow problem during a longer time frame is relevant for actual studies. This illustrates the non-linear accumulation of sources of local error. We concentrate on this sensitivity issue of predictions for an unsteady fluid flow in detail.

In this paper we consider implicit time integration schemes, such as the Crank–Nicolson and Euler backward approaches, to solve the Navier–Stokes equations for an unsteady flow. We follow the approach of Jameson and reformulate the evolution problem into a steady flow problem for each time step. Specifically, we concentrate on a DNS of unsteady supersonic two-dimensional boundary layer flow. The unsteadiness of this flow is caused by the interaction of the boundary layer with shocks that arise in the flow [4]. Also, spontaneous vortex shedding is observed. These phenomena combine into a rather complex flow, which makes it an appropriate model for the present study. In addition, this study has relevance for more complex cases since its numerical methodology also applies to other flows. Simultaneously, this specific flow has some importance for various situations of practical significance, for example, transonic flight conditions or a combustion ramjet. The main findings in relation to sensitivity of the flow predictions to numerical time step changes that we can establish in relation to the unsteady shock–boundary layer flow can also be sketched in relation to other canonical flows, such as flow in a mixing layer and in a regular boundary layer.

The method of lines is adopted to separate the temporal from the spatial discretization. Straightforward integration in time can be done with explicit methods, such as Runge–Kutta methods. The main advantages of these methods are their simplicity and their low memory demands. However, the time step is bounded by numerical stability requirements. For flows where the dominant physical time scales are comparatively large, e.g. a slowly pitching airfoil, the stability time step is often too restrictive with respect to the necessary temporal accuracy. If an appropriate A-stable implicit time integration scheme is applied, there is no such stability restriction on the time step. As a result of the application of an implicit scheme, a large set of coupled non-linear equations has to be solved at each time step and the possibility exists to simulate an unsteady flow with a time step that is larger than the stability time step. In principle, this system can be solved directly with a Newton–Raphson-like approach, as is done in Reference [5]. Alternatively, a pseudo-time derivative can be added to create a new set of differential equations following the approach of Jameson [2]. The stationary solution of these differential equations corresponds to the desired Navier–Stokes solution at the next time step. This approach can be interpreted as a quasi-Newton method. In this way all convergence acceleration techniques developed for steady flow computations can be used like, for example, multigrid [6,7], in order to accelerate the iteration process in pseudo-time.

With the introduction of an A-stable implicit time integration scheme, the stability requirement on the time step vanishes. However, several new questions arise. How large can the time step actually chosen to be in order to resolve the unsteady solution accurately? How accurate should the non-linear system, which is related to each time step, be solved and how sensitive does the prediction of different flow quantities depend on this and other related numerical parameters? Is it possible to solve the non-linear set of equations at each time step with arbitrary precision and for all relevant step sizes? These type of questions motivated much of the research reported here and may help to identify numerical parameter settings that lead to proper accumulative error–behaviour in the simulation of complex unsteady phenomena with implicit methods. Likewise, it may contribute to identify criteria that dynamically determine bounds for the time steps that yield a time accurate simulation. We explicitly remark that in this paper the focus is on sensitivity of predictions in relation to variations in the time step. An empirical approach is adopted, which implies analysing a systematic set of numerical parameters, such as the time step. A promising approach for actually dynamically determining the time step during the course of a simulation, such that the temporal accuracy is maintained, is based on temporal correlation between the solution at different instants and will be published elsewhere [van Buuren R, IJzerman WL, Geurts BJ. A dynamic time-step method for accurate simulation of unsteady flow based on temporal correlation. *Journal of Computational Physics* 2000 (submitted)]. The temporal correlation criterion may also be interpreted as capturing a certain (sufficiently large) fraction of the total energy contained in the solution at every point in time. In this way, a close connection with a proper orthogonal decomposition [8] of the signal, which is local in time, can be established [9]. Part of judging the suitability of any specific strategy for determining the time step in a flow simulation is the amount of variation in certain flow properties due to variations in the time stepping method, e.g., the time step or the relaxation method. Since different flow properties have different responses to variations in time integration parameters we incorporate both mean flow aspects as well as information from local instantaneous solution components into the analysis.

In the past a lot of work has been done concerning the central questions raised above. In Reference [7] a two-dimensional turbulent flow around a pitching airfoil is solved. The time accuracy of these simulations was validated empirically 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 non-linear set of equations for each time step and the required convergence level needed at each pseudo-time step was investigated focusing on some typical flow quantities. In Reference [6] similar two- and three-dimensional unsteady flow around an airfoil is considered. An application that is more closely related to the present flow is the DNS performed in Reference [5] for a spatially evolving three-dimensional turbulent boundary layer. In this 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 was 600. A fixed number of iterations were performed for the non-linear system at each time step. The validation of this number of iterations was done by comparing the Navier–Stokes solution obtained with 2, 3 and 4 iterations per time step. All these examples have in common that the time step is related to an external forcing frequency. In the case of transitional and inhomogeneous turbulent flow smaller spatial flow structures with smaller characteristic time scales can arise within the flow domain. Since these features of the solution are dynamically relevant they should be captured by the time step determination approach. The flow studied here illustrates a case in which time scales that are characteristic of the 'internal' physics of the flow are the important factor for the time step in relation to global error accumulation.

In this paper we simulate a two-dimensional shock–boundary layer interaction flow. The flow is unsteady due to the specific properties of a steady blowing and suction profile [4] at the upper boundary of the computational domain, where we prescribe the normal velocity. Of special importance for this flow is the artificial outflow boundary, which is represented by characteristic non-reflecting boundary conditions [10] in combination with a buffer domain to eliminate numerical reflections due to the outflow boundary [4]. For this flow we do not prescribe any explicit external unsteady flow condition and so another criterion to identify a suitable time step is necessary. For this purpose we calculate the global error caused by the spatial discretization and use this to compare with the global error due to the time integration. The global time integration error not only depends on the magnitude of the time step but also on the accuracy with which the non-linear system arising at each time step is solved and the relaxation method that is used to obtain or accelerate convergence. The obvious demand that the global error decreases if the solution is determined more accurately each time step results in a set of requirements for these numerical parameters. The specific construction of a stopping criterion that determines the local accuracy turns out to be a key factor.

The contents of this paper is as follows. In Section 2 we state the governing equations and the numerical method using the explicit Runge–Kutta scheme. In Section 3 the numerical results obtained with the explicit method are presented, which are used as a reference solution for the implicit time integration method introduced in Section 4. Section 5 contains a discussion of the global error bounds for the implicit time integration scheme. 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 and a connection is made with the classical chaos theory. Finally, in Section 6 conclusions are summarized.

2. GOVERNING EQUATIONS AND NUMERICAL METHOD

In this section we state the equations governing viscous compressible flow and specify the numerical method incorporating the explicit time integration method.

2.1. Governing equations

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

$$\frac{\partial q}{\partial t} + \nabla\cdot(f_{\mathrm{c}} - f_{\mathrm{v}}) = 0 \tag{1}$$

with $q = [\rho, \rho u, \rho v, E]^T$ and where f_{c} and f_{v} denote the convective and viscous fluxes respectively, which are given by

$$f_{\mathrm{c}} = \begin{bmatrix} \rho u \mathbf{i}_x & \rho v \mathbf{i}_y \\ (\rho u^2 + p)\mathbf{i}_x & \rho u v \mathbf{i}_y \\ \rho u v \mathbf{i}_x & (\rho v^2 + p)\mathbf{i}_y \\ u(E+p)\mathbf{i}_x & v(E+p)\mathbf{i}_y \end{bmatrix},$$

$$f_{\mathrm{v}} = \begin{bmatrix} 0 & 0 \\ \sigma_{xx}\mathbf{i}_x & \sigma_{xy}\mathbf{i}_y \\ \sigma_{yx}\mathbf{i}_x & \sigma_{yy}\mathbf{i}_y \\ (u\sigma_{xx} + v\sigma_{xy} + kT_x)\mathbf{i}_x & (u\sigma_{yx} + v\sigma_{yy} + kT_y)\mathbf{i}_y \end{bmatrix} \tag{2}$$

where $\mathbf{i}_x$ and $\mathbf{i}_y$ are unit vectors in the x- and y-direction respectively, ρ is the density, E is the total energy density, and u and v are the velocity components in the x- and y-direction respectively. The heat conductivity k equals

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

where μ is the non-dimensional viscosity, Pr is the Prandtl number, γ is the adiabatic gas constant, M_∞ is the Mach number at infinity, and Re is the Reynolds number to which we return shortly. Here we use $Pr = 0.72$ and $\gamma = 1.4$ The constitutive equation for the pressure, p, is given by

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

The temperature T is related to the density ρ and the pressure p by the perfect gas law

$$T = \gamma M_\infty^2 \frac{p}{\rho} \tag{5}$$

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

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

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

$$\sigma_{xx} = \frac{\mu}{Re}\left(\frac{4}{3}u_x - \frac{2}{3}v_y\right), \quad \sigma_{yy} = \frac{\mu}{Re}\left(\frac{4}{3}v_y - \frac{2}{3}u_x\right), \quad \sigma_{xy} = \sigma_{yx} = \frac{\mu}{Re}(u_y + v_x) \tag{7}$$

Here $Re = (\rho_\infty u_\infty \delta_i^*)/\mu_\infty$ is the reference Reynolds number. The above variables 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_∞, pressure and energy density $\rho_\infty u_\infty^2$, viscosity μ_∞ and time δ_i^*/u_∞. The subscript ∞ refers to the free stream values.

2.2. Spatial discretization

In this section we specify the spatial discretization method. The Navier–Stokes equations (1) are written in conservation form, where the state vector q contains the densities of the conserved quantities. Integration of Equation (1) over an arbitrary volume in space shows that the components of q change only due to a flux through the boundaries of this volume. To solve Equation (1) we use a finite volume method on a structured grid that computes the flux over the control volume edges, as in Figure 1

$$\Omega_{i,j}\frac{\mathrm{d}q_{i,j}}{\mathrm{d}t} + h_{i+1/2,j} - h_{i-1/2,j} + h_{i,j+1/2} - h_{i,j-1/2} = 0 \tag{8}$$

where $\Omega_{i,j}$ is the area of the control volume and h the numerical flux vector that is evaluated at the four boundary segments $(i + 1/2, j)$, etc.

The numerical flux h contains a convective and a viscous part

$$h_{i+1/2,j} = c_{i+1/2,j} - d_{i+1/2,j} \tag{9}$$

where c is the numerical convective flux and d denotes the numerical viscous flux. The convective and viscous fluxes are approximated in different ways. For the convective term a

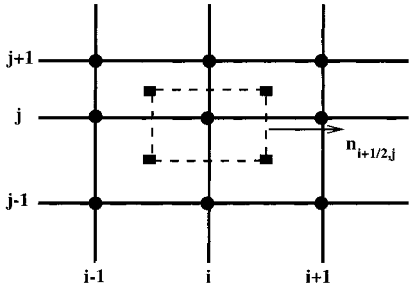

Figure 1. Control volume for grid point (i,j). The arrow denotes the normal on the edge.

higher-order TVD scheme is used in order to capture shocks occurring in this flow, while for the viscous flux we use a conservative second-order central scheme [11], which is suitable for our purposes. The spatial discretization is described in more detail next.

2.2.1. Convective terms. In order to capture shocks correctly we use a higher-order TVD scheme for the convective flux [12]. The convective flux on the control volume edges is approximated using the flux difference splitting method of Roe [13], e.g.

$$c_{i+1/2,j}=\frac{1}{2}l_{i+1/2,j}\begin{bmatrix} f_{c,x}(q_l)+f_{c,x}(q_r) \\ f_{c,y}(q_l)+f_{c,y}(q_r) \end{bmatrix}\cdot \mathbf{n}_{i+1/2,j}-\frac{1}{2}l_{i+1/2,j}|A_{i+1/2,j}|(q_r-q_l) \tag{10}$$

where $f_{c,x}$ and $f_{c,y}$ denote the x and y components of the convective flux f_c in Equation (1), $l_{i+1/2,j}$ is the length of the control volume edge, $\mathbf{n}_{i+1/2,j}$ is the unit normal vector on the edge in the outward direction and the dot denotes the inner product. The appropriate left and right state vectors [14] are denoted by q_l and q_r and $A_{i+1/2,j}$ is the combined flux Jacobi matrix obtained by taking the Jacobi matrix of the inner product of the normal with the flux vector on the control volume edge. The absolute value of the flux Jacobi matrix is defined as $|A|=R|\Lambda|L$, where R and L are right and left eigenvector matrices of A and $|\Lambda|$ is a diagonal matrix containing the absolute values of the eigenvalues of A. As a result, the flux Jacobi matrix becomes a function of q_l, q_r and the normal $\mathbf{n}_{i+1/2,j}$, i.e. $A_{i+1/2,j}=A(q_l, q_r, \mathbf{n}_{i+1/2,j})=A(q_{lr}, \mathbf{n}_{i+1/2,j})$, where $q_{lr}=q_{i+1/2,j}$ denotes Roe's average state vector, which is determined by calculating the primitive variables according to

$$u_{lr}=\frac{u_l+du_r}{1+d}, \quad v_{lr}=\frac{v_l+dv_r}{1+d}, \quad H_{lr}=\frac{H_l+dH_r}{1+d} \tag{11}$$

with $d=\sqrt{\rho_l/\rho_r}$ and H is the specific enthalpy. For a first-order approximation of the flux the left state vector q_l equals $q_{i,j}$ and the right state vector q_r equals $q_{i+1,j}$. To achieve a higher-order monotonic upwind method we interpolate the state vector with the MUSCL scheme [14] using the minmod limiter

$$q_l = q_{i,j} + \frac{1}{4}[(1-\eta)\,\mathrm{Lim}(\Delta q_{i-1/2}, \omega\Delta q_{i+1/2})(1+\eta)\,\mathrm{Lim}(\Delta q_{i+1/2}, \omega\Delta q_{i-1/2})] \tag{12}$$

$$q_r = q_{i+1,j} - \frac{1}{4}[(1-\eta)\,\mathrm{Lim}(\Delta q_{i+3/2}, \omega\Delta q_{i+1/2})(1+\eta)\,\mathrm{Lim}(\Delta q_{i+1/2}, \omega\Delta q_{i+3/2})] \tag{13}$$

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

$$\mathrm{Lim}(a, b) = \frac{1}{2}(\mathrm{sign}(a) + \mathrm{sign}(b))\min(|a|, |b|) \tag{14}$$

The parameters η and ω must satisfy

$$-1 \leq \eta \leq 1; \qquad 1 \leq \omega \leq \frac{3-\eta}{1-\eta} \tag{15}$$

to ensure monotonicity [15]. For the special choice $\eta = \frac{1}{3}$, the scheme is third-order accurate in smooth regions. In this paper we will use $\eta = \frac{1}{3}$ and $\omega = \frac{3}{2}$. This method has been applied successfully before both to steady flow around airfoils [12,16] and to time-dependent turbulent flow in a mixing layer at high Mach numbers, including unsteady shocks [17].

2.2.2. Viscous terms. The viscous flux is approximated with a second-order accurate conservative scheme as defined in Reference [11]. Since the viscous flux contains second-order spatial derivatives, the standard approach is to first find an appropriate approximation of the first-order derivatives on the control volume edges in Figure 1. The first-order derivative in a point in the middle of a grid cell can be approximated by applying Gauss' theorem to the surrounding grid cell. Consider, for example, u_x

$$u_x \approx \frac{1}{|\Omega|}\int_\Omega u_x \,\mathrm{d}\Omega = \frac{1}{|\Omega|}\int_S un_x \,\mathrm{d}S \tag{16}$$

where $|\Omega|$ is the volume of the grid cell, n_x is the x component of the normal on the grid cell edge and S is the boundary of the volume. Applying Gauss' theorem once more on the control volume with the above definition of the first derivatives results in a second-order accurate conservative approximation of the viscous flux.

2.3. Boundary conditions

Figure 2 presents a sketch of the computational domain. The solid wall is represented by an adiabatic no-slip boundary. At the inflow boundary we impose the Blasius solution of the compressible boundary layer equations as described in Reference [4]. 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. Therefore, one numerical boundary condition is necessary to obtain a well-posed problem. In total we extrapolate the density from the interior and impose the pressure and the streamwise and normal velocity components. The

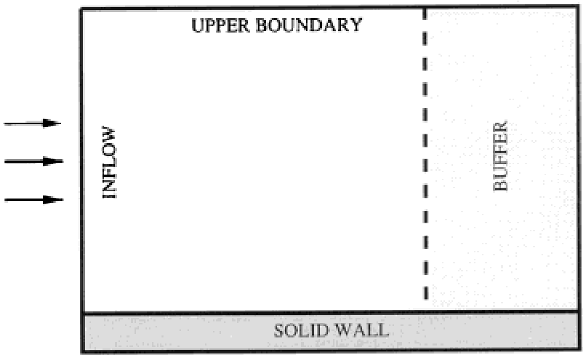

Figure 2. Computational domain.

flow in the second part of the inflow boundary, away from the solid wall, is supersonic and convection-dominated and all characteristic enter the computational domain so that all flow quantities can be imposed.

For the outflow boundary we use a special buffer technique as developed in Reference [4]. 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:

$$q = q_{\mathrm{ref}} + \tilde{\zeta}(x)(\tilde{q} - q_{\mathrm{ref}}) \tag{17}$$

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

$$\zeta = (1 - C_1 x_{\mathrm{b}}^2)\left(1 - \frac{1 - \mathrm{e}^{C_2 x_{\mathrm{b}}^2}}{1 - \mathrm{e}^{C_2}}\right) \tag{18}$$

where C_1 and C_2 are tuning parameters. The co-ordinate x_{b} is the buffer co-ordinate defined by

$$x_{\mathrm{b}} = \frac{x - x_{\mathrm{s}}}{x_{\mathrm{e}} - x_{\mathrm{s}}} \tag{19}$$

where x_{s} and x_{e} are the x-co-ordinates 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 buffer function is applied. Hence, if the grid is refined, the buffer function is 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 $\tilde{\zeta}$ as

$$\tilde{\zeta} = \zeta^{C_3 \Delta t} \tag{20}$$

where the constant C_3 is added as a tuning parameter. In Reference [4] an extensive investigation has been performed on the tuning of the constants in Equations (18) and (20), which leads to the choice of $C_1 = 0.005$, $C_2 = 20$ and $C_3 = 26.4$. The 'optimized' buffer as described has also been applied to free shear layers and performs equally well, without further adjustments in the tuning parameters.

The upper boundary acts as a free stream boundary, where a time-independent blowing and suction profile is prescribed by imposing the normal velocity. 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 (LODI) non-reflecting boundary condition as defined in References [4,10]. This method is based on the characteristics of inviscid flow. For outgoing characteristics, information can be obtained from the interior domain. For incoming characteristics, no physical conditions except for the normal velocity are known and therefore the amplitude of the incoming waves is kept constant in time, thus minimizing reflections at the boundary.

2.4. *Explicit time integration*

The Navier–Stokes equations (1) can be written in a semi-discrete form as

$$\frac{\mathrm{d}q_{i,j}}{\mathrm{d}t} + f_{i,j}(q) = 0 \tag{21}$$

where $f(q)$ denotes the total numerical flux defined in Section 2.2. For the explicit time integration we use a second-order accurate, explicit four-stage compact storage Runge–Kutta scheme, given by

$$q^{(k)} = q^{(0)} + \alpha_k \Delta t f(q^{(k-1)}) \quad (k = 1, 2, 3, 4) \tag{22}$$

with $q^{(0)} = q(t)$, $q^{(4)} = q(t + \Delta t)$ and the coefficients $\alpha_1 = \frac{1}{4}$, $\alpha_2 = \frac{1}{3}$, $\alpha_3 = \frac{1}{2}$ and $\alpha_4 = 1$. 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

$$\Delta t_{i,j} = \frac{1}{\dfrac{1}{\Delta t^{\mathrm{c}}_{i,j}} + \dfrac{1}{\Delta t^{\mathrm{v}}_{i,j}}} \tag{23}$$

where $\Delta t^{\mathrm{c}}_{i,j}$ is the local time step limitation related to the convective flux and analogously $\Delta t^{\mathrm{v}}_{i,j}$ is related to the viscous flux. The convective time step limitation is given by

$$\Delta t^{\mathrm{c}}_{i,j} = \frac{\sigma|\Omega_{i,j}|}{\max(l_{i+1/2,j}\lambda_{i+1/2,j}, l_{i-1/2,j}\lambda_{i-1/2,j}) + \max(l_{i,j+1/2}\lambda_{i,j+1/2}, l_{i,j-1/2}\lambda_{i,j-1/2})} \tag{24}$$

where σ is the Courant–Friedrich–Lewy (CFL) number and the λs are the maximal absolute values of the eigenvalues of the flux Jacobi matrix on the corresponding control volume edges. The local viscous time step limitation is given by

$$\Delta t^{\mathrm{v}}_{i,j} = \frac{\sigma|\Omega_{i,j}|^2}{\alpha(|l_{i+1/2,j}|^2 + |l_{i,j+1/2}|^2)} \tag{25}$$

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

3. EXPLICIT NUMERICAL REFERENCE RESULTS

In this section we present some simulation results for the reference test case described in Reference [4] using the Runge–Kutta scheme as specified above. As an initial condition we use the compressible Blasius boundary layer 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 $\Delta y_{\max}/\Delta y_{\min} = 8.8$. The blowing and suction profile described in the previous section is displayed in Figure 3. The specific choice of the parameters a, d, w and ϕ influence the shock strength and position as well as the temporal behaviour of the flow (for more details see Reference [4]). 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 that occur in the flow [4]. Also, spontaneous vortex shedding is observed. Data sampling starts after the flow becomes statistically stationary. The transient process that leads to this state is illustrated in Figure 4, where we plot the shock sensor defined by

$$S(t) = \max\left|\frac{\partial p}{\partial x}(t)\right|\frac{\Delta x}{p_\infty} \tag{26}$$

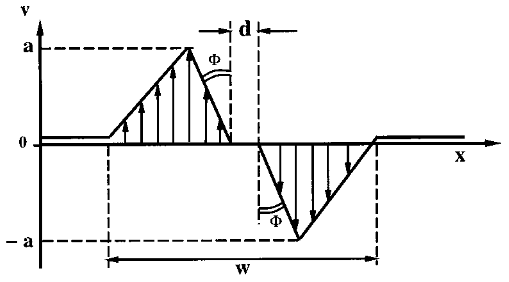

Figure 3. Blowing and suction profile of the normal velocity, v at the free-stream boundary.

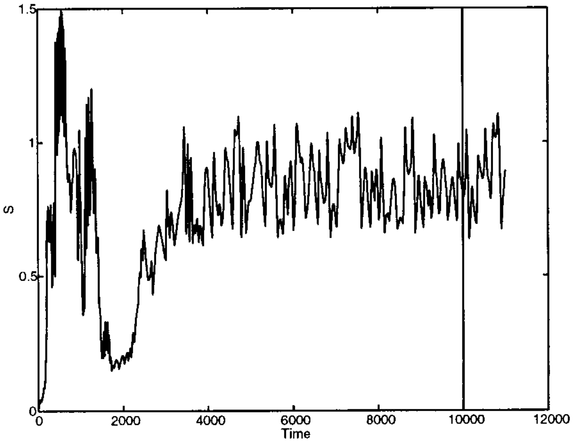

Figure 4. Shock sensor as a function of time. The vertical line represents the beginning of the data sampling interval which lies well within the statistically stationary region.

This is a measure for the maximum shock strength in the flow. The temporal behaviour of the shock sensor indicates that after a certain period the flow becomes statistically stationary. As an initial condition for the sampling period throughout this paper we take the solution obtained at $t = 10000$. The sampling is performed in the interval T from $t = 10000$ to $t = 11000$ in which period we determine time averaged and root-mean-square (r.m.s.) flow quantities. We used several CFL numbers, σ, 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 $t = 11000$ 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, which represents a good level of accuracy to define the point of reference. In Figure 5 the time-averaged Mach field is plotted. The presence of two shocks and a separation bubble can clearly be observed. The solid line in Figure 6 represents the mean

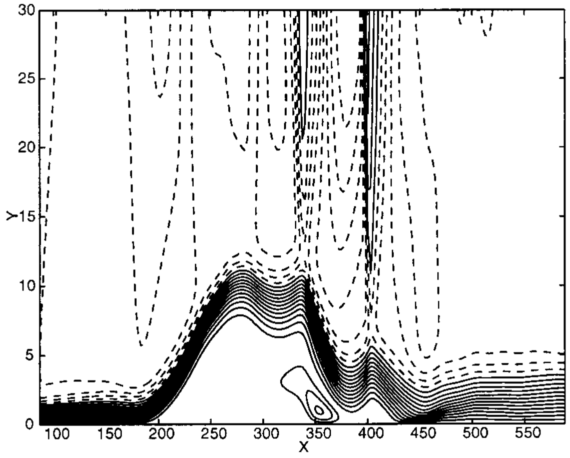

Figure 5. Time averaged Mach field over the sampling period T. Supersonic flow ($M > 1$) is shown with dotted contour lines and subsonic flow with solid contour lines.

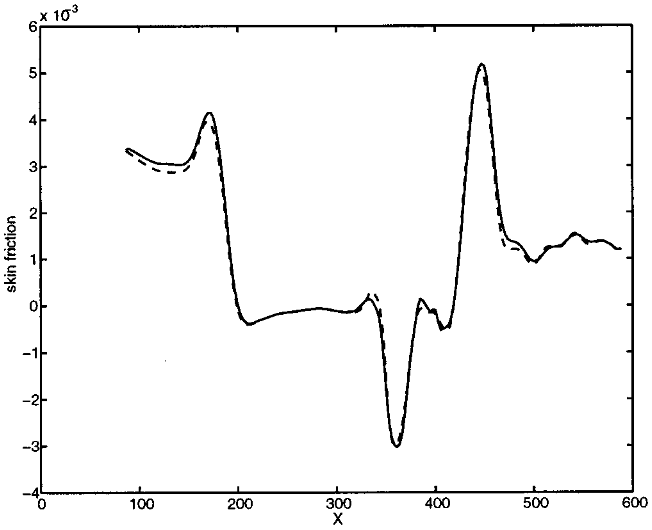

Figure 6. Mean skin friction for the explicit method. The solid line represents the skin friction on the original grid (193 × 65), the dashed line on the fine grid (385 × 129).

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 7 and 8 (solid lines). The first location ($U1$) is in the boundary layer just in front of

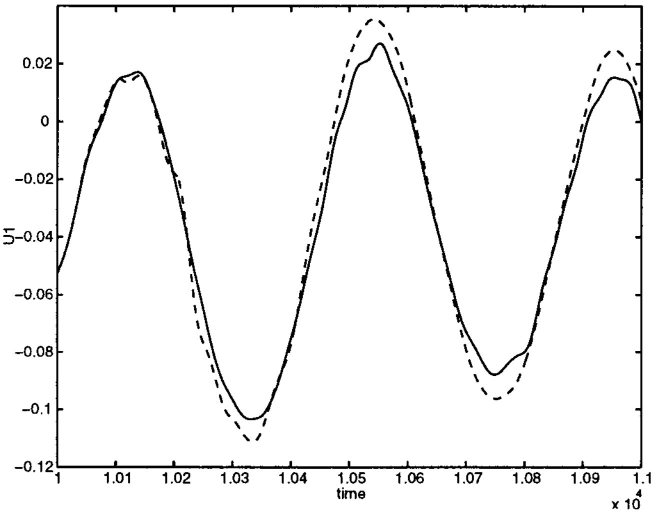

Figure 7. Instantaneous solution *U*1 obtained with the explicit time integration for the original grid (solid line) and the fine grid (dashed line).

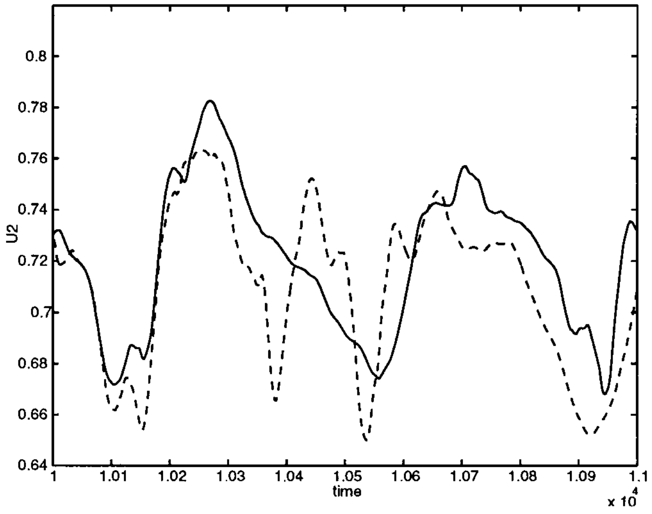

Figure 8. Instantaneous solution *U*2 obtained with the explicit time integration for the original grid (solid line), and the fine grid (dashed line).

the separation bubbled The second location ($U2$) is within the separation bubble itself. We observe that the temporal behaviour at the second location is more complex than that at the first location and it may be expected that the sensitivity of the solution at the two locations to changes in, for example, the time step also differs considerably. Therefore, we will use both locations to study the effect of large time steps in the next sections.

4. IMPLICIT TIME INTEGRATION METHOD

In the previous section, a second-order accurate explicit Runge–Kutta method was used for the time integration. The time step needed for a time accurate simulation may be significantly larger than the stability time step for explicit methods given by Equation (23). In order to circumvent the stability requirements on the time step, we adopt an implicit time integration method. In order to determine the accuracy of the solution, we compare the implicit solution with the explicit reference solution for different types of flow quantities, i.e. mean, r.m.s. and instantaneous solution properties, which display quite different dependencies on variations of numerical parameters. In this section, we define the implicit time integration method.

4.1. Time integration and implicit approximation of the flux

In Section 2.4 the Navier–Stokes equations were written in a semi-discrete form in Equation (21). Applying the second-order Crank–Nicolson scheme to the temporal derivative in Equation (21) yields

$$q_{i,j}^{n+1} + \frac{1}{2}\Delta t f_{i,j}(q^{n+1}) = q_{i,j}^{n} - \frac{1}{2}\Delta t f_{i,j}(q^{n}) \tag{27}$$

where the superscript denotes the time level and the time step is represented by Δt. Rewriting Equation (27) yields the following non-linear system:

$$F_{i,j}(q^{n+1}) = g_{i,j}(q^{n}) \tag{28}$$

where the right-hand side $g_{i,j}$ depends on quantities known from the previous time level and is therefore fixed during the calculation of q^{n+1}. Several aspects of this approach do not depend on the implicit Crank–Nicolson time integration used here. In fact, all implicit multi-step methods give rise to the general non-linear form $F(q^{n+1}) = g(q^{n}, q^{n-1}, \ldots)$. In principle, Equation (28) can be solved with Newton iteration but this is not always computationally efficient. We use a method introduced in Reference [2], where a pseudo-time is added to Equation (28), which yields

$$\frac{\mathrm{d}v_{i,j}}{\mathrm{d}\tau} + F_{i,j}(v) = g_{i,j} \tag{29}$$

The resulting steady state solution of Equation (29) corresponds to the desired solution q^{n+1} of Equation (28). To solve Equation (29) we use the Euler backward scheme in pseudo-time as was done in Reference [12], where it was shown that the use of Euler backward was computationally more efficient to obtain a steady state than the explicit Runge–Kutta scheme defined in Section 2.4. Although the Euler backward is only a first-order accurate scheme, this choice has no effect on the second-order accuracy of the solution q^{n+1} of Equation (28). Applying the Euler backward to Equation (29), the following system has to be solved for each pseudo-time level:

$$\left(\frac{I}{\Delta\tau_{i,j}}+\frac{\partial F}{\partial v}(v^k)\right)\Delta v_{i,j}=g_{i,j}-F_{i,j}(v^k) \tag{30}$$

where $\Delta v = v^{k+1}-v^k$, the superscript k denotes the pseudo-time level and $\partial F/\partial v$ is a symbolic representation of the numerical Jacobi matrix of F defined in Equation (28). For infinite $\Delta\tau$ and exact numerical representation of the Jacobi matrix, the iteration scheme (30) corresponds to the exact Newton iteration with quadratic convergence if the iterative solution is sufficiently close to the fixed point. However, it is not possible to obtain and invert the exact Jacobi matrix at reasonable cost. Therefore, we approximate the Jacobi matrix as in References [12,18]. 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 semi-Newton iteration.

The numerical system that has to be solved at each time step is defined in Equation (30). A computationally more efficient method is possible if the left-hand side of Equation (30) is evaluated only once per time step and not after every pseudo-time step as would be required from Equation (30). The number of pseudo-iterations necessary to obtain the steady state with a desired accuracy does not increase due to this adjustment [19]. This indicates that this additional approximation of the Jacobi matrix has no appreciable influence for unsteady simulations. Applying this approximation to the numerical system in Equation (30), we get

$$\mathscr{A}\Delta v_{i,j}=g_{i,j}-F_{i,j}(v^k) \tag{31}$$

where $\mathscr{A}$ is now a fixed matrix and only the right-hand side needs updating after every pseudo-time step. The speed-up factor obtained with this approximation compared with the case where the Jacobi matrix is updated every pseudo-time step is about 2.3.

Similar to the flux, we distinguish the convective part of the Jacobi matrix from its viscous part. The convective part of the Jacobi matrix is approximated using a first-order upwind approach. The viscous part is approximated with a second-order central scheme. Due to the five-point stencil of Roe's scheme in the two-dimensional case [13], we get a sparse Jacobi matrix with five bands of non-zero 4 × 4 matrices. The five blocks of the convective flux for a grid point (i,j) are given by

$$D^{\mathrm{c}}_{i,j}=A^{+}_{i-1/2,j}-A^{-}_{i+1/2,j}+A^{+}_{i,j-1/2}-A^{-}_{i,j+1/2}$$

$$N^{\mathrm{c}}_{i,j}=A^{-}_{i,j+1/2}$$

$$S^{c}_{i,j} = -A^{+}_{i,j-1/2}$$

$$E^{c}_{i,j} = A^{-}_{i+1/2,j}$$

$$W^{c}_{i,j} = -A^{+}_{i-1/2,j} \tag{32}$$

where D^c, N^c, S^c, E^c and W^c denote the diagonal, north, south, east and west contribution to the convective part of the flux. The matrices $A+$ and A^- are determined by the positive and negative eigenvalues of the flux Jacobi matrix [20].

The viscous flux defined in Section 2.2.2 uses a nine-point stencil. In order to stay within the five-point stencil used for the convective part of the numerical Jacobi matrix, the cross-derivatives are neglected. Consider a transformation from physical to computational space

$$(x, y) \rightarrow (\xi, \eta) \tag{33}$$

Applying this transformation to, for example, the stress term σ_{xx} in Equation (7) gives

$$\sigma_{xx} = \frac{4}{3}\frac{\mu}{Re}(\xi_x u_\xi + \eta_x u_\eta) - \frac{2}{3}\frac{\mu}{Re}(\xi_y u_\xi + \eta_y u_\eta) \tag{34}$$

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

$$\sigma_{xx} \approx \frac{4}{3}\frac{\mu}{Re}\xi_x\left(\frac{\rho u}{\rho}\right)_\xi - \frac{2}{3}\frac{\mu}{Re}\xi_y\left(\frac{\rho v}{\rho}\right)_\xi \tag{35}$$

Linearization around the conservative variables 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 convective and viscous blocks multiplied by a factor $\frac{1}{2}\Delta t$ arising from the time integration and the addition of the diagonal blocks $I/\Delta\tau$ due to the pseudo-time iteration.

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 ρ, ρu 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.

4.2.1. Buffer domain. The buffer technique described in Section 2.3 was developed originally for the explicit time integration scheme described in Section 2.4. 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 the buffer is applied. If we 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 Equation (21) followed by the application of the buffer results in a discretized evolution 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 Equation (1) in the buffer domain. These steps are considered in more detail next. Define the buffer by

$$\xi = \begin{cases} 1 & \text{if } x < x_{\mathrm{b}} \\ \tilde{\zeta}(x) & \text{if } x \geq x_{\mathrm{b}} \end{cases} \tag{36}$$

If we would use the Euler forward method we can write the solution on the next time step in two phases

$$\begin{aligned} v^{n+1} &= q^n_{i,j} - \Delta t f_{i,j}(q^n) \\ q^{n+1}_{i,j} &= q^*_{i,j} + \xi(v^{n+1} - q^*_{i,j}) = q^n_{i,j} - \Delta t\left\{\frac{1}{\Delta t}(\xi - 1)(q^*_{i,j} - q^n_{i,j}) + \xi f_{i,j}(q^n)\right\} \equiv q^n_{i,j} - \Delta t \tilde{f}_{i,j}(q^n) \end{aligned} \tag{37}$$

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

$$\tilde{f}_{i,j}(q) = \left\{\frac{1}{\Delta t}(\xi - 1)(q^*_{i,j} - q_{i,j}) + \xi f_{i,j}(q)\right\} \tag{38}$$

Taking the limit Δt to zero in Equation (37) shows that there exists a consistent differential equation for every region of the computational domain

$$\begin{cases} \dfrac{\mathrm{d}q}{\mathrm{d}t} + f(q) = 0, & x < x_{\mathrm{b}} \\ \dfrac{\mathrm{d}q}{\mathrm{d}t} + \alpha(q^* - q) + f(q) = 0, & x_{\mathrm{b}} \leq x < x_{\mathrm{e}} \\ q = q^*, & x = x_{\mathrm{e}} \end{cases} \tag{39}$$

where $\alpha = C_3 \log(\zeta)$ with ζ defined in Section 2.3. Notice that the value of α goes to infinity near the outflow boundary. Therefore, we do not treat Equation (39) directly but use the formulation of the new flux in Equation (38) 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 new Jacobi matrix is straightforward since the blocks in

Equation (32) 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 Equation (38).

4.2.2. 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

$$(\rho v)^{n+1}_{i,j*} = v_i \rho^{n+1}_{i,j*} = (\rho v)^{n}_{i,j*} - \Delta t v_i f^{(\rho)}_{i,j*}(q^n) \tag{40}$$

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 (1).

4.3. Linear solver

At each pseudo-time level a linear system has to be solved as a result of the discretization in Equation (30). Using the block notation as defined above, the system for each grid point (i, j) can be written as

$$\left(N + S + E + W + \frac{I}{\Delta\tau_{i,j}} + D\right)\Delta v_{i,j} = g_{i,j} - F_{i,j}(v^k) \tag{41}$$

where $\Delta\tau_{i,j}$ is the time step obtained by the same stability criterion as described in Section 2.4 but now applied on the pseudo-time level. In order to solve Equation (41) the same symmetrical Gauss–Seidel method is used as in Reference [12]. *A priori* it is not clear whether the linear system should be solved to machine accuracy. Here the residual is not reduced to machine accuracy but instead only two sweeps are performed at each pseudo-time level, which is found more efficient with respect to CPU time.

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 accuracy of the solution obtained with the implicit method and the choice of the implicit time step.

5.1. Error bounds

To empirically identify a suitable accuracy time step, Δt_{acc}, we need a criterion for the accuracy in order to determine whether a certain time step is acceptable or not. To this purpose we introduce $\mathscr{E}$, which represents a measure for the global accumulated error over the sampling period T, as described in Section 3. We will define norms to specify $\mathscr{E}$ in more detail shortly. We require the global errors to be comparable, i.e.

$$\mathscr{E}(\Delta t) \lessapprox \mathscr{E}(\Delta x) \tag{42}$$

which states that the global error due to the time integration should be smaller than or maximally comparable with the global error due to the spatial discretization errors. In this way, the total global error remains comparable with the spatial discretization error and is not essentially increased by the temporal integration. There is no *a priori* reason to work with simulations in which either one of the global error terms is much smaller than the other in this philosophy. The main guidance for selecting a proper Δt is to have an acceptable accumulation of local errors in the non-linear evolution of the flow. At least Δt should be sufficiently small to keep $\mathscr{E}(\Delta t)$ bounded. In addition, the total global error should remain at the same level as that due to the spatial discretization. In fact, errors arising from spatial discretization are usually much more difficult to reduce in view of memory requirements and computational effort. For this reason $\mathscr{E}(\Delta x)$ forms a direct and pragmatic bound for the global error. The reduction of $\mathscr{E}(\Delta t)$ by reducing Δt is computationally somewhat simpler and it is comparably straightforward to reduce accumulated global errors due to time integration below the level of global errors due to spatial discretization. Typically, we found in the present study that reducing $\mathscr{E}(\Delta t)$ to a level about five to ten times smaller than $\mathscr{E}(\Delta x)$ is suitable. For other flows the findings differ slightly to which we return momentarily.

The next step is to determine appropriate norms for the errors defined in Equation (42). For this purpose, the numerical results for the explicit run on the original grid are considered as the reference solutions. We define the following general formula for the norm of the error of the averaged and r.m.s. quantities

$$\|\psi\| = \frac{1}{\psi_{\text{norm}}} \left\{ \frac{1}{L_x L_y} \int_0^{L_x} \int_0^{L_y} (\psi - \psi_{\text{ref}})^2 \, \mathrm{d}x \, \mathrm{d}y \right\}^{1/2} \tag{43}$$

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. This is added in order to render all norms roughly of comparable magnitude which facilitates the comparison of norms of different quantities. For the instantaneous quantities that are monitored as the solution as a function of time at a fixed location, we define

$$\|\psi\| = \frac{1}{\psi_{\text{norm}}} \left\{ \frac{1}{T} \int_0^{T} (\psi - \psi_{\text{ref}})^2 \, \mathrm{d}t \right\}^{1/2} \tag{44}$$

where T is the total sampling time. For the specific case of the skin friction, the normalization value ψ_{norm} is defined as $\psi_{\text{norm}} = \max(\psi_{\text{ref}})$, which leads to

$$\|\psi\| = \frac{1}{\max(\psi_{\text{ref}})} \left\{ \int_0^{L_x} (\psi - \psi_{\text{ref}})^2 \, \mathrm{d}x \right\}^{1/2} \tag{45}$$

In this paper we will use the skin friction, the r.m.s. of U and the instantaneous velocity U measured at two different locations in the computation 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 presentation.

In order to determine the norms related to the global spatial discretization error $\mathscr{E}(\Delta x)$ 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 $t = 10000$ and the Blasius solution at the inflow and outflow boundary are obtained with the same interpolation method. In Figure 6 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 7 and 8 we have plotted the velocity U in two different locations in the computational domain. In the smooth region ($U1$) of the flow, the results compare well, the main difference is in the amplitude. For the velocity $U2$ only the large structures seem to agree. On the fine grid, additional higher frequencies in time are resolved and the reference grid appears somewhat too coarse to resolve all features. However, for calculations with the implicit method in which we focus on the sensitivity of the solution to changes in Δt, the reference setting is more than adequate.

5.2. Comparison with explicit results

In this section we study the effect of the magnitude of the time step on the global error norms $\mathscr{E}$. In the global errors we use the suggestive notations Δt and Δx to distinguish between the two types of error sources. However, the global error due to the time stepping method not only depends on the magnitude of the time step but also on the local tolerance level $\mathscr{E}$ that is a measure of the local accuracy with which the solution of the non-linear system in Equation (30) for each time step is obtained. Also the relaxation parameter $\Delta\tau$ may have some effect. The global error $\mathscr{E}$ on a given grid at least depends on three parameters: the time step Δt, the local tolerance level $\mathscr{E}$ and the relaxation parameter $\Delta\tau$. To complete the definition of the iteration scheme a definition of the stopping criterion is required. For this purpose we introduce $\phi(v)$, which yields the stopping criterion $\phi(v) < \epsilon$. Our total numerical method is now desired to satisfy the obvious requirement that the global accumulative errors should decrease if the solution is determined with a higher local accuracy at each time step. A more accurate solution can be obtained, for example, by decreasing the time step or decreasing ϵ. Shortly we will define the function $\phi(v)$ but first a more precise formulation of the requirements on the global error are given

(1) $\mathscr{E}(\Delta t_1, \epsilon, \Delta\tau) \leq \mathscr{E}(\Delta t_2, \epsilon, \Delta\tau), \quad \Delta t_1 \leq \Delta t_2, \quad \forall\Delta\tau$

(2) $\mathscr{E}(\Delta t, \epsilon_1, \Delta\tau) \leq \mathscr{E}(\Delta t, \epsilon_2, \Delta\tau), \quad \epsilon_1 \leq \epsilon_2, \quad \forall\Delta\tau$

(3) $\mathscr{E}(\Delta t, \epsilon, \Delta\tau_1) \approx \mathscr{E}(\Delta t, \epsilon, \Delta\tau_2), \quad \Delta\tau_1 \neq \Delta\tau_2$

(4) $\phi(v)$ sensitive to the initial solution (46)

Our eventual aim is to identify values for Δt, ϵ and $\Delta\tau$ in combination with a good stopping criterion ϕ, such that the numerical predictions respond to changes in these parameters in the above manner. In this way a 'predictable' error–behaviour can be realized though a proper setting of the numerical parameters. If Δt_1 is smaller than Δt_2, the truncation error for the time integration method should also be smaller. Therefore, requirement (1) states that the global error $\mathscr{E}$ should decrease if the time step decreases. Secondly, it appears reasonable that the global error should not increase if one puts more efforts in solving the solution each time step. So, requirement (2) says that if one solves the non-linear system more accurately, the obtained solution is also better in the sense that the global error is smaller. Requirement (3) is somewhat less transparent. Since we do not solve system (30) 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. multi-grid. Sensitivity on the relaxation method should be small. Finally, $\phi(v)$ should be sensitive to a better initial guess of the solution, which means that if an initial solution in pseudo-time is close to the numerical solution of Equation (28), fewer iterations should be needed compared with the case where the difference between the initial solution and the numerical solutions is larger. Although the global error demands in Equation (46) seem very reasonable, it is not trivial to find a $\phi(v)$ such that all four requirements are satisfied. A few examples of functions $\phi(v)$ that we considered are

$$\phi_1(v^k) = \frac{\|F(v^k) - g\|_2}{\|g\|_2}; \qquad \phi_2(v^k) = \frac{\|F(v^k) - g\|_2}{\|\Delta t\tilde{f}(q^n)\|_2}; \qquad \phi_3(v^k) = \frac{\|\Delta v^k\|_2}{\|\Delta v^1\|_2} \tag{47}$$

where the superscript k denotes the pseudo-time level. Computations and analysis indicate that ϕ_1 does not give rise to simulations with a global error that satisfies requirement (1). This may be explained by a Taylor expansion for small time steps. The denominator scales with Δt while the numerator scales with a constant plus Δt. To remove this Δt dependence ϕ_2 is introduced. Although the first requirement is satisfied when using ϕ_2, it turns out to be very sensitive to the relaxation parameter $\Delta\tau$ and is therefore not a suitable candidate. The third measure, ϕ_3, is not sensitive to a better initial estimate of the solution. However, computations indicate that requirements (1) and (2) are satisfied, and in the limit for $\Delta\tau \to 0$ and $\Delta\tau \to \infty$, requirement (3) is satisfied. For the intermediate range of $\Delta\tau$ it is found that requirement (3) is approximately satisfied but the results are not totally independent of $\Delta\tau$. In the sequel we will use ϕ_3.

The time step for the explicit reference run is fixed at $\Delta t = 0.2$. For the simulations with the implicit time integration scheme we perform calculations for a range of Δt values. In Figures 9–12 the norms defined in Equations (43) and (44) are plotted for various quantities as a function of the local accuracy $\epsilon = 10^{-k}$. The required global error relation stated in Equation (42) is represented in these figures by a solid horizontal line corresponding in this case to a suggestive value of $\mathscr{E}(\Delta x)/10$. Basically, we can now refer to a setting of the numerical time integration parameters as appropriate if $\mathscr{E}(\Delta t)$ lies below this line and the opposite applies if $\mathscr{E}(\Delta t)$ lies above this 'spatial error reference line'. As an example, the figures show that for almost all quantities and time steps $\epsilon = 10^{-1}$ is too large, whereas typically the accumulated time integration error responds acceptably if the solution at every time step is determined with a sufficiently small residue of $\epsilon = 10^{-2}$ or smaller. To illustrate this in more detail, in Figure 13 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

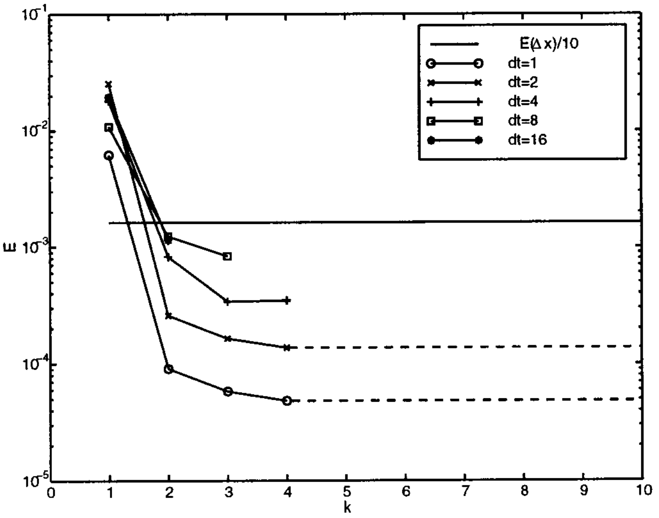

Figure 9. Global error for the skin friction as a function of the local error $\epsilon = 10^{-k}$ for various time steps.

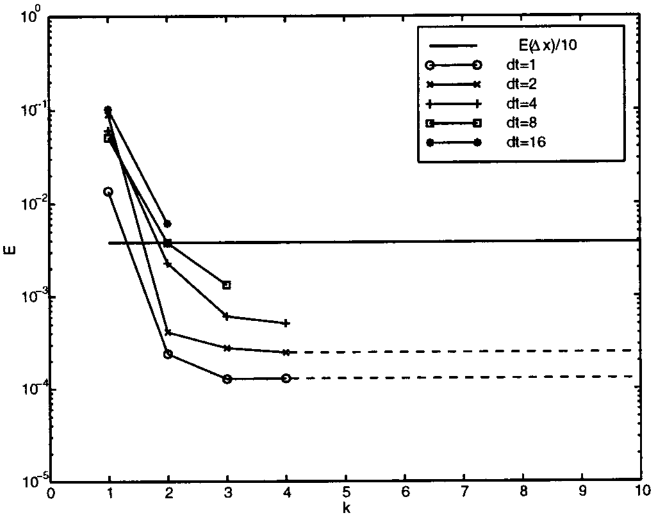

Figure 10. Global error for the r.m.s. U as a function of the local error $\epsilon = 10^{-k}$ for various time steps.

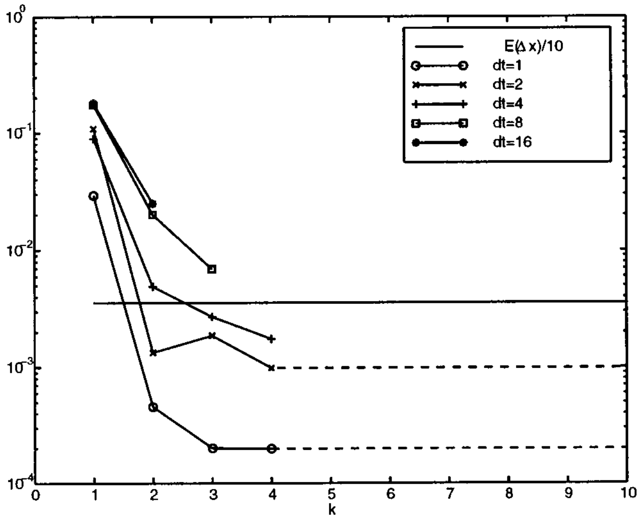

Figure 11. Global error for the instantaneous velocity $U1$ as a function of the local error $\epsilon = 10^{-k}$ for various time steps.

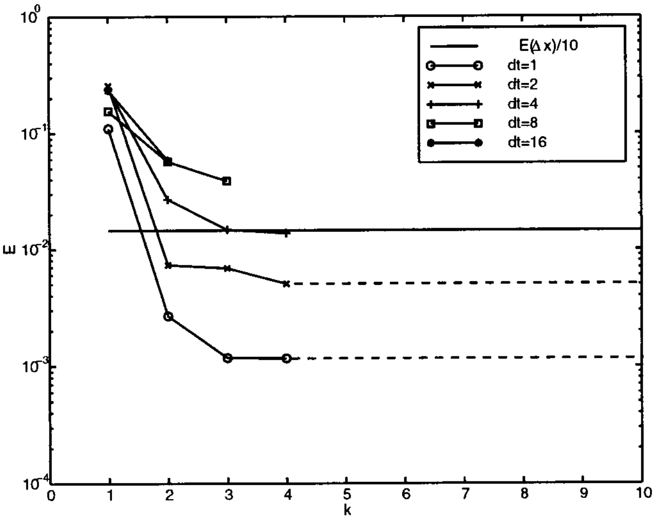

Figure 12. Global error for the instantaneous velocity $U2$ as a function of the local error $\epsilon = 10^{-\mathrm{k}}$ for various time steps.

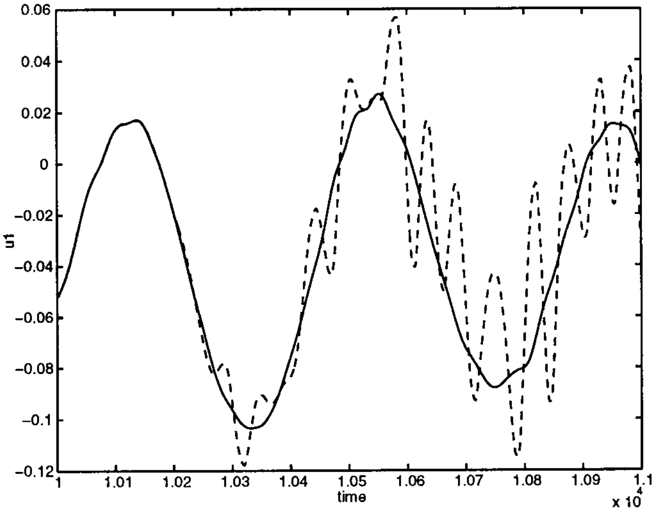

Figure 13. Instantaneous solution $U1$ as a function of time for the implicit scheme with $\Delta t = 2$ and $\epsilon = 10^{-1}$ (dashed) and the explicit reference solution (solid).

clearly indicates that ϵ is too large. If $\epsilon = 10^{-2}$ almost excellent agreement is obtained, which illustrates the significant drop in the norm shown in Figure 11. In fact, for $\epsilon \leq 10^{-2}$ the first three requirements in (46) were verified. The dashed lines in Figures 9–12 correspond to the solution with arbitrarily small ϵ. Clearly, the global error reaches an asymptotic value corresponding to solutions that only contain the truncation error of the time integration scheme. 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. This aspect will be addressed in more detail in the next section.

The sensitivity of mean flow properties is shown to be quite small. For the skin friction, Figure 9 shows that for all time steps the choice $\epsilon = 10^{-2}$ results in an error smaller than 0.1 per cent. We also monitored their mean quantities like, for example, 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. Hence, for mean quantities it seems that the implicit time step can be chosen up to about eighty times larger than the explicit stability times step.

The precise ratio between $\mathscr{E}(\Delta t)$ and $\mathscr{E}(\Delta x)$ that is still acceptable for global time accuracy is not sharply defined and is somewhat influenced by the precise norm that is used to identify a measure for the error. Consider, for instance, the quantity $U1$ at time step $\Delta t = 4$ and $\epsilon = 10^{-2}$. In Figure 11 the error lies slightly above the spatial error reference line. However, in Figure 14 the solution is seen to almost coincide with the solution found from application of the Runge–Kutta scheme. Finally, in Figures 15–18 we show the quantities $U1$ and $U2$ for the time steps $\Delta t = 8$ and $\Delta t = 16$ for the highest attainable accuracy ϵ in these cases. The figures show that for $\Delta t = 8$ the quantity $U1$ is resolved quite well. For $\Delta t = 16$ the solution

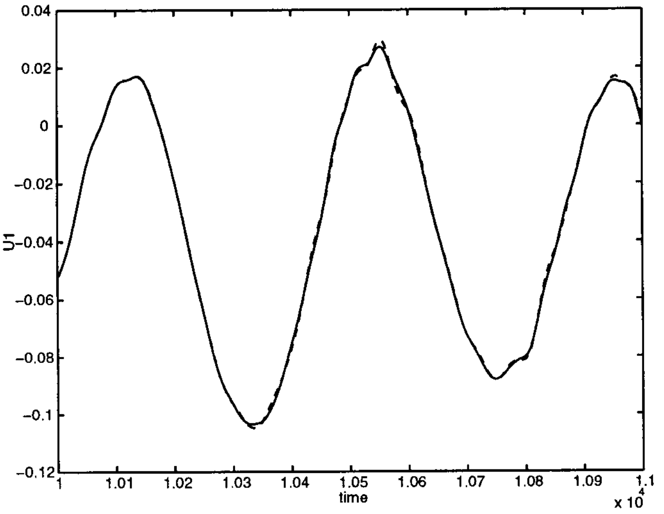

Figure 14. Instantaneous solution $U1$ as a function of time for the implicit scheme with $\Delta t = 4$ and $\epsilon = 10^{-2}$ (dashed) and the explicit reference solution (solid).

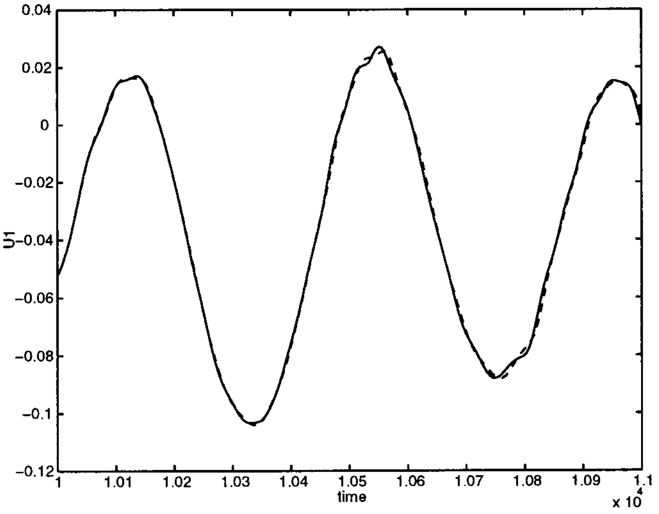

Figure 15. Instantaneous solution $U1$ as a function of time for the implicit scheme with $\Delta t = 8$ and $\epsilon = 10^{-3}$ (dashed) and the explicit reference solution (solid).

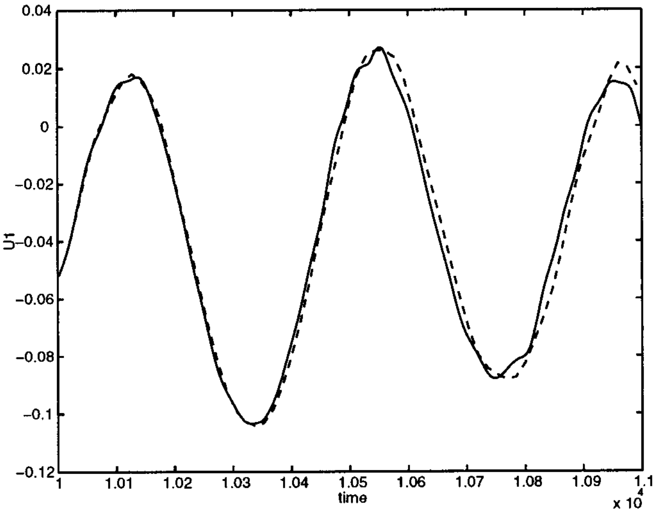

Figure 16. Instantaneous solution $U1$ as a function of time for the implicit scheme with $\Delta t = 16$ and $\epsilon = 10^{-2}$ (dashed) and the explicit reference solution (solid).

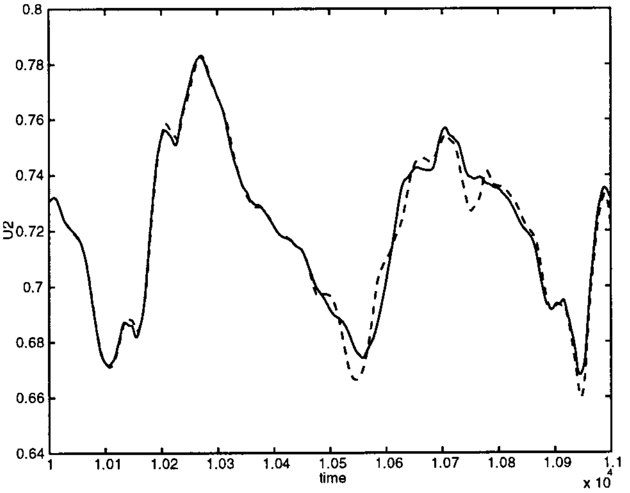

Figure 17. Instantaneous solution $U2$ as a function of time for the implicit scheme with $\Delta t = 8$ and $\epsilon = 10^{-3}$ (dashed) and the explicit reference solution (solid).

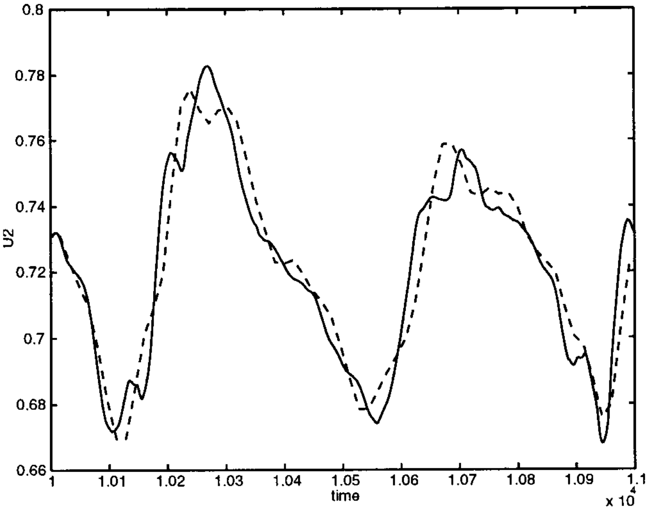

Figure 18. Instantaneous solution $U2$ as a function of time for the implicit scheme with $\Delta t = 16$ and $\epsilon = 10^{-2}$ (dashed) and the explicit reference solution (solid).

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 even more for $\Delta t = 16$, where the main trends are still captured but the deviation of the solution increased considerably. Hence, the global error criterion (42) represents a robust and useful guidance for identifying suitable time steps, even for instantaneous quantities. To be on the safe side, a choice of $\Delta t = 4$ is close to Δt_{acc} for all instantaneous quantities. For the instantaneous quantities the accuracy time step is therefore a factor four times smaller than for the mean quantities but compared with the stability time step it is still about a factor 20 times larger.

We also investigated the sensitivity of the flow predictions to changes in Δt, for conventional boundary layer flow, which does not involve far field pressure gradients to influence the flow [21]. In the case of two-dimensional boundary layer flow, the development of the flow is much less abrupt, both in space and time. The transition from laminar to complex unsteady behaviour is well described for Δt about 20–40 times larger than the explicit stability time step for instantaneous flow quantities. The situation is reversed if we turn to flow in a mixing layer [22]. In this flow the explicit stability time step is typically much closer to the accuracy time step and consequently the speed up arising from implicit time integration is much smaller. In both these flows we noticed that relation (42) is a useful way to classify suitable numerical parameter settings that give rise to 'predictable' error behaviour.

5.3. Dynamical behaviour 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 Figures 9–12). The convergence does not necessarily break down in the first time step for each Δt and ϵ but may occur somewhere within the sampling period. This indicates that the convergence breakdown depends not only on the time step and stopping criterion but also on the initial condition and accumulated effects.

To study the effect of the time step on the convergence behaviour we perform one time step for a range of Δt values with initial conditions taken as the solution obtained at $t = 10000$ 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 machine accuracy can be obtained. The convergence behaviour of the numerical system (31) in pseudo-time can be visualized with phase portraits by plotting the values of two characteristic quantities at every pseudo-time level. The value of Δv should converge to zero and therefore two locations are taken where the absolute value of Δv is m-aximum. In Figures 19–21 we have plotted such phase portraits of the density for $\Delta t = 6$, $\Delta t = 8$ and $\Delta t = 16.5$. In order to visualize the behaviour, the density at iteration level $k = 1000$ is subtracted.

The dynamical structure for these three time steps in quite different. In the case of $\Delta t = 6$ the dynamic convergence behaviour resembles that of a converging spiral associated with a stable stationary solution of Equation (29). Compared with the convergence behaviour at smaller time steps, which are all straight lines, a structural difference of the dynamics around the fixed

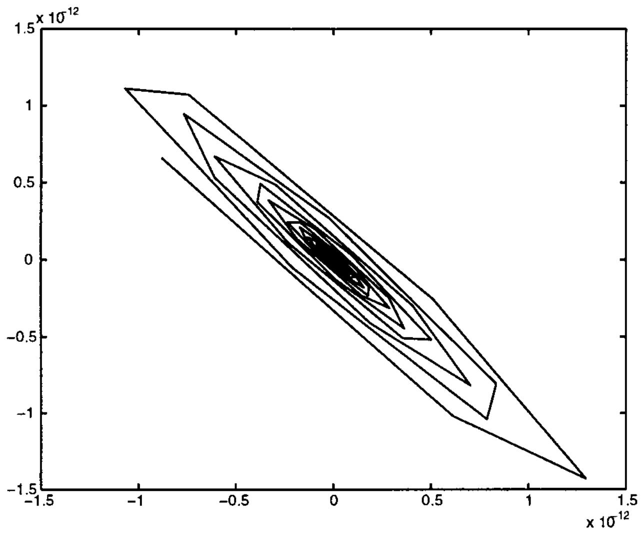

Figure 19. Phase portrait of two typical values of ρ for $\Delta t = 6$.

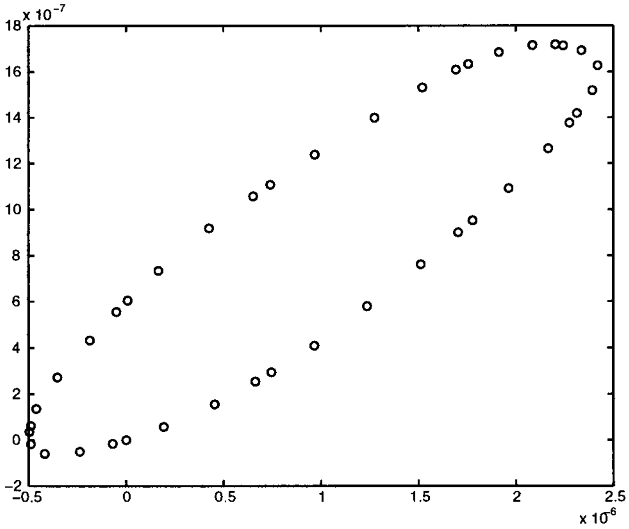

Figure 20. Phase portrait of two typical values of ρ for $\Delta t = 8$.

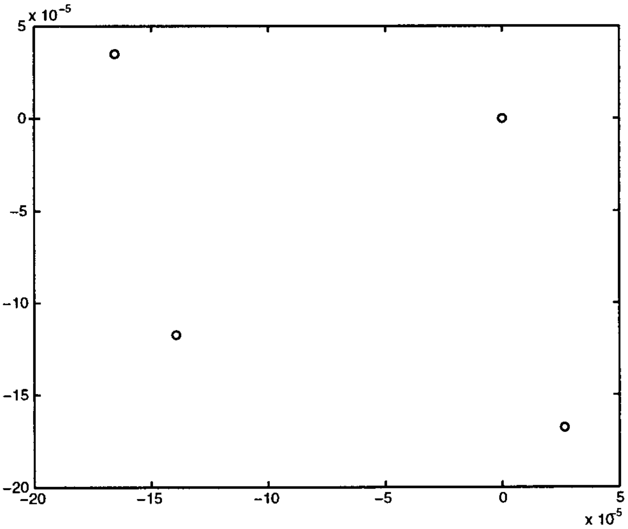

Figure 21. Phase portrait of two typical values of ρ for $\Delta t = 16.5$.

point already occurs even thought the fixed point of Equation (28) 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. We investigated numerically that this limit cycle is an attractor. Further investigations indicate 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 in pseudo-time 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 behaviour we define the winding number by

$$\Gamma = \frac{1}{2\pi(N+1-k^*)} \sum_{k=k^*}^{N} \psi_k \tag{48}$$

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 centre of the limit cycle. For a range of Δt between $\Delta t = 6$ and $\Delta t = 100$, we observe that $0.21 \leq \Gamma \leq 0.3$. 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.

There is a striking analogy between these findings and recent work of Yee and Sweby [23,24], who made a study of the asymptotic behaviour of time integration methods for steady state problems. In the present context we also use a time integration scheme to obtain a steady state, which connects the present work with the work of Yee and Sweby if we consider the pseudo-time step as the bifurcation parameter. To confirm this observation we repeat the simulation for $\Delta t = 16.5$ (period for solution) with $\text{CFL}_\tau = 1.2$. In Figure 22 it is shown that for $\text{CFL}_\tau = 1.2$ 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 be related to the presence of the basins of attraction for spurious and real solutions. Yee and Sweby observe that for certain time discretizations and fixed-point solutions of a differential equation fragmented basin of attraction exists. This agrees with the results for the simulations at $\Delta t = 8$.

Additionally we examined the influence of the temporal integration of Equation (21) on the convergence problems described above and adopted the Euler backward scheme which is a dissipative scheme in contrast to the Crank–Nicolson scheme, which is non-dissipative. 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.

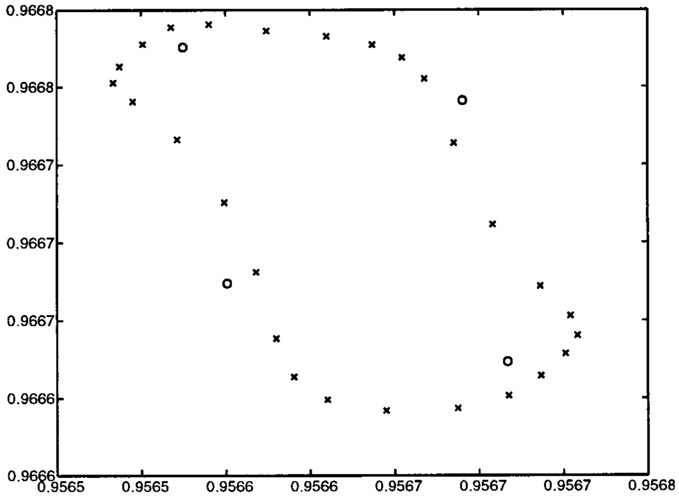

Figure 22. Phase portrait of two typical values of ρ for $\Delta t = 16.5$; for $CFL = 1$ (○) and $CFL = 1.2$ (×).

6. CONCLUSIONS

In this paper 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. In order to avoid numerical reflections at the outflow boundary we used a special buffer technique developed by Wasistho *et al.* [4]. By analogy with the treatment needed for the explicit Euler forward scheme we have shown that the application of the buffer technique is consistent with a modified differential equation in the buffer domain. This facilitates the implicit treatment of the buffer significantly and only requires a flux redefinition. For the implicit time integration scheme we used the second-order Crank–Nicolson scheme, which results in a large set of coupled non-linear algebraic equations that have to be solved each time step. To solve this system we added a pseudo-time derivative and used the Euler backward scheme in pseudo-time to obtain the stationary solution. In principle, Newton iteration prescribes that the Jacobi matrix has to be determined every pseudo-time step. However, we found that the Jacobi matrix during a time step can be kept constant, which results in a speed-up factor of about 2.3.

Contrary to explicit time integration methods, the time step for implicit schemes such as Crank–Nicolson is bounded by accuracy requirements alone. How to find acceptable Δt and even how to identify the suitability of a certain Δt is a complicated matter. As a guideline for the allowable global error due to the time integration scheme we found that an estimate of the global spatial discretization error is an appropriate measure. We formulated requirements for the global time integration errors that correlate the time step, local accuracy of each time step and the relaxation method. If the numerical parameters are properly specified the error–behaviour complies with these requirements and the global error decreases if the solution is determined more accurately each time step. Computations showed that if the requirements are obeyed proper solutions of the flow problem are obtained. It appeared that the stability time step is too restrictive with respect to accuracy and a considerable speed-up is, in principle, possible using implicit methods.

Convergence problems were encountered for large time steps. Yee and Sweby [23,24] 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 non-linear system of equations to obtain the solution at the next time step. Numerical simulations confirm the sensitive dependence on the pseudo-time step.

Finally, we observe that, although no special attention has been paid to convergence acceleration techniques, the current implementation of the implicit scheme is already competitive with the explicit scheme with respect to CPU time. Since the number of pseudo-iterations for large time steps is considerable we expect that the application of multi-grid techniques will result in a significant speed-up. This is a topic of future research, in particular in the context of transitional and turbulent flow in three spatial dimensions.

REFERENCES

1. Rodi W, Ferziger JH, Breuer M, Pourquié M. Status of large-eddy simulation: results of a workshop. *Transactions of the ASME* 1997; **119**: 284.
2. Jameson A. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. AIAA Paper No. 91-1596, 1991.
3. Broeze J, Geurts BJ, Kuerten JGM, Streng M. Multigrid acceleration of time-accurate DNS of compressible turbulent flow. In *Proceedings 7th Copper Mountain Conference on Multigrid Methods*, Melson ND (ed.). NASA, Conference Publications 3227, NASA: Washington, DC, 1995; 87.
4. Wasistho B, Geurts BJ, Kuerten JGM. Spatial simulation techniques for time-dependent compressible flow over a flat plate. *Computers and Fluids* 1997; **26**: 713–739.
5. Rai MM, Moin P. Direct numerical simulation of transition and turbulence in a spatially evolving boundary layer. *Journal of Computational Physics* 1993; **109**: 169–192.
6. Crumpton PI, Giles MB. Implicit time-accurate solutions on unstructured dynamic grids. *International Journal for Numerical Methods in Fluids* 1997; **25**: 1285–1300.
7. Pierce NA, Alonso JJ. Efficient computation of unsteady viscous flows by an implicit preconditioned multgrid method. *AIAA Journal* 1998; **36**: 3.
8. Aubry N, Guyonnet R, Lima R. Spatio-temporal analysis of complex signals: theory and applications. *Journal of Statistics in Physics* 1991; **64**: 683–739.
9. IJzerman WL. Signal representation and modeling of spatial structures in fluids. PhD thesis, University of Twente, Netherlands, 2000.
10. Poinsot TJ, Lele SK. Boundary conditions for direct simulations of compressible viscous flow. *Journal of Computational Physics* 1992; **101**: 104–129.
11. Vreman AW, Geurts BJ, Kuerten JGM. Comparison of numerical schemes in large eddy simulation of the temporal mixing layer. *International Journal for Numerical Methods in Fluids* 1996; **22**: 297–311.
12. van Buuren R, Kuerten JGM, Geurts BJ. Instabilities of stationary inviscid compressible flow around an airfoil. *Journal of Computational Physics* 1997; **138**(2): 520–539.
13. Roe PL. Characteristic-based schemes for the Euler equations. *Annual Review of Fluid Mechanics* 1986; **18**: 337–365.
14. van Leer B. Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second-order scheme. *Journal of Computational Physics* 1974; **14**: 361–370.
15. Yee HC. *Computational Fluid Dynamics*. von Karman Institute for Fluid Dynamics, Lecture Series *1989-04*, 1989.
16. van der Burg JW. Numerical methods for transonic flow calculations. PhD thesis, University of Twente, Netherlands, 1992.
17. Vreman AW, Kuerten JGM, Geurts BJ. Shocks in direct numerical simulation of the confined three-dimensional mixing layer. *Physics of Fluids* 1995; **7**: 2105.
18. Yee HC. Construction of explicit and implicit shock capturing methods. *Journal of Computational Physics* 1987; **68**: 151–179.

19. Heeg RS. Stability and transition of attachment-line flow. PhD thesis, University of Twente, Netherlands, 1998.
20. Roe PL. Characteristic-based schemes for the Euler equations. *Annual Review of Fluid Mechanics* 1986; **18**: 337–365.
21. Geurts BJ, Vreman AW, Kuerten JGM. Comparison of DNS and LES of transitional and turbulent compressible flow: flat plate and mixing layer. In *Proceedings of the 74th Fluid Dynamics Panel and Symposium on Application of DNS and LES to Transition and Turbulence*, Crete (AGARD Conference Proceedings 551), 1994; 51.
22. Vreman AW, Geurts BJ, Kuerten JGM. Large-eddy simulation of the turbulent mixing layer. *Journal of Fluid Mechanics* 1997; **339**: 357.
23. Yee HC, Sweby PK. Global asymptotic behavior of iterative implicit schemes. *International Journal of Bifurcation and Chaos* 1994; **4**(6): 1579–1611.
24. Yee HC, Sweby PK. Dynamical approach study of spurious steady state numerical solutions of nonlinear differential equations II. Global asymptotic behaviour of time discretizations. *Computational Fluid Dynamics* 1995; **4**: 219–283.