Theoret. Comput. Fluid Dynamics (1998) 12: 109–129

Theoretical and Computational Fluid Dynamics

Direct Numerical Simulation of Turbulent Compression Ramp Flow

N.A. Adams

ETH Zürich, Institute of Fluid Dynamics,
ETH Zentrum, CH-8092 Zürich, Switzerland

Communicated by M.Y. Hussaini

Received 13 November 1997 and accepted 14 May 1998

Abstract. A numerical procedure for the direct numerical simulation of compressible turbulent flow and shock–turbulence interaction is detailed and analyzed. An upwind-biased finite-difference scheme with a compact centered stencil is used to discretize the convective part of the Navier–Stokes equations. The scheme has a uniformly high approximation order and allows for a spectral-like wave resolution while dissipating nonresolved wave numbers. When hybridized with an essentially nonoscillatory scheme near discontinuities, the scheme becomes shock–capturing and its resolution properties are preserved. Diffusive parts are discretized with symmetric compact finite differences and an explicit Runge–Kutta scheme is used for time-advancement. The peculiarities of efficient upwinding and coupling procedures are described and validation results are given. Using direct numerical simulation data, some aspects of turbulent supersonic compression ramp flow are studied to demonstrate the effectiveness of the simulation procedure.

1. Introduction

A major problem in modeling turbulent supersonic flows is the correct assessment of viscous–inviscid interaction problems. These encompass most of the critical issues of gas-turbine design and supersonic aircraft design. Of particular interest is the interaction of turbulent boundary layers with shocks. Here, all three ingredients, mean flow, turbulent fluctuation, and shock, interact strongly, a situation which poses a stringent requirement for accurate turbulence modeling when the Reynolds-averaged Navier–Stokes equations are solved.

A standard test case for turbulence models are compression corner flows. Generically they appear in many aeronautical configurations. They give rise to a particularly interesting combination of phenomena, all of which are more or less confined to a relatively narrow region about the corner. First, the turbulence in the oncoming boundary layer responds to a rapid distortion. Second, for large enough Mach numbers and deflection angles there is a shock-induced unsteady separation. The area of separated flow is contained by a detached, curved shear layer, and fluctuations in this shear layer, subject to high strain, are strongly amplified. Third, there is an unsteady shock motion related to the unsteady separation. Present turbulence models in most cases give unsatisfactory results in the region of rapid distortion and in the separation region, in particular with regard to mean flow profiles and turbulence quantities. A representative account of the prediction capability of present turbulence models for the supersonic compression ramp has been given, e.g., by Wilcox (1990).

The objective of this work is the direct numerical simulation (DNS) of a shock–boundary layer interaction along a supersonic compression corner. It is expected that DNS data will shed light on the validity of basic

modeling assumptions. Also of concern is the impact of three-dimensionality and unsteadiness on the solution of the Reynolds-averaged Navier–Stokes equations with turbulence models. This report summarizes the first phase of the work during which a numerical method suitable for this problem has been developed and a computer code has been written and tested.

In Section 2 we address the fundamental equations, boundary conditions, and grid generation. We focus in Section 3 on the discretization of the fundamental equations. Code validation is summarized in Section 4. Finally, in Section 5 several aspects of supersonic turbulent compression ramp flow are studied using DNS data in order to assess the effectiveness of the simulation procedure.

2. Mathematical Model

2.1. Governing Equations

The fundamental equations solved are the conservation equations for mass, momentum, and energy in generalized coordinates:

$$\frac{\partial}{\partial t}\frac{U}{J}+\frac{\partial}{\partial \xi}\frac{F_{\mathrm{E}}}{J}+\frac{\partial}{\partial y}\frac{G_{\mathrm{E}}}{J}+\frac{\partial}{\partial \zeta}\frac{H_{\mathrm{E}}}{J}=\frac{\partial}{\partial \xi}\frac{F_{\mathrm{S}}}{J}+\frac{\partial}{\partial y}\frac{G_{\mathrm{S}}}{J}+\frac{\partial}{\partial \zeta}\frac{H_{\mathrm{S}}}{J},$$

where the conservative variables are $U=\{\rho\rho u\rho v\rho wE\}$, with $E=p/(\kappa-1)+\rho(u^2+v^2+w^2)/2$. Considering only essentially two-dimensional configurations we limit the coordinate generalization to the (x,z)-plane. The physical space (x,y,z) is mapped onto the computational space (ξ,η,ζ) which is Cartesian and equispaced (y is mapped onto η linearly). The convective fluxes are then given by

$$F_{\mathrm{E}}=\begin{bmatrix}\rho(u\xi_x+w\xi_z)\\ \rho u(u\xi_x+w\xi_z)+p\xi_x\\ \rho v(u\xi_x+w\xi_z)\\ \rho w(u\xi_x+w\xi_z)+p\xi_z\\ (E+p)(u\xi_x+w\xi_z)\end{bmatrix},$$

and similarly for G_{E} and H_{E}. The viscous fluxes are given by

$$F_{\mathrm{S}}=\begin{bmatrix}\tau_{xx}\xi_x+\tau_{xz}\xi_z\\ \tau_{xy}\xi_x+\tau_{yz}\xi_z\\ \tau_{xz}\xi_x+\tau_{zz}\xi_z\\ -q_x\xi_x-q_z\xi_z+(u\tau_{xx}+v\tau_{xy}+w\tau_{xz})\xi_x+(u\tau_{xz}+v\tau_{yz}+w\tau_{zz})\xi_z\end{bmatrix},$$

and similarly for G_{S} and H_{S}. The Jacobian of the coordinate transformation is $J=\xi_x\zeta_z-\xi_z\zeta_x$. The stresses are defined as

$$\tau_{xx}=\frac{\mu}{Re}\left[\frac{4}{3}\left(\frac{\partial u}{\partial \xi}\xi_x+\frac{\partial u}{\partial \zeta}\zeta_x\right)-\frac{2}{3}\frac{\partial v}{\partial \eta}\eta_y-\frac{2}{3}\left(\frac{\partial w}{\partial \xi}\xi_z+\frac{\partial w}{\partial \zeta}\zeta_z\right)\right],$$

with analogous definitions for τ_{yy} and τ_{zz}, and

$$\tau_{xy}=\frac{\mu}{Re}\left[\left(\frac{\partial v}{\partial \xi}\xi_x+\frac{\partial v}{\partial \zeta}\zeta_x\right)+\frac{\partial u}{\partial \eta}\eta_y\right],$$

and similarly for τ_{xz}, τ_{yz}, and τ_{xz}. The heat fluxes are defined as

$$q_x=-\frac{\mu}{(\kappa-1)M_\infty^2 PrRe}\left(\frac{\partial T}{\partial \xi}\xi_x+\frac{\partial T}{\partial \zeta}\zeta_x\right),$$

q_y and q_z analogously. The viscosity is calculated according to Sutherland's law. We also assume the thermal equation of state for perfect gases to be valid, $\kappa M_\infty^2 p=\rho T$. The abbreviations ξ_x, ξ_z, η_y, ζ_x, ζ_z are used for the metric coefficients $\partial\xi/\partial x$, $\partial\xi/\partial z$, $\partial\eta/\partial y$, $\partial\zeta/\partial x$, $\partial\zeta/\partial z$.

2.2. Boundary Conditions

At the inflow all dependent variables are imposed. At the no-slip wall all velocities are set to zero and either the wall temperature or the wall-normal gradient of temperature is prescribed. Given a wall-normal temperature-gradient distribution $\partial T/\partial n$, a Neumann condition for the temperature can be imposed by setting

$$T_\zeta = \frac{\sqrt{\zeta_x^2 + \zeta_z^2}(\partial T/\partial n) - (\xi_x\zeta_x + \xi_z\zeta_z)(\partial T/\partial \xi)}{\zeta_x^2 + \zeta_z^2}$$

whenever it appears during the computation of heat flux and stress terms (due to the temperature dependence of the viscosity).

At the outer truncation plane, Dirichlet boundary conditions fixing all variables at their free-stream values are prescribed. A nonreflecting condition for the temporal derivative of incoming Riemann variables, as in Thompson (1987), has been found occasionally to give rise to viscous instability. These inviscid nonreflecting boundary conditions are well-posed only if viscous terms are negligible. This is not necessarily the case if the shock, for instance at the exit, reaches close to the truncation plane, where the mesh spacing is coarse.

As for all spatially developing flows the outflow boundary requires special treatment. A buffer domain approach has been used successfully by Pruett *et al.* (1995). The drawback of that approach is that the viscous terms of the basic equations need to be manipulated in the buffer region. Here, we resort to a simpler approach as proposed by Guo *et al.* (1996) and introduce a sponge layer at the outflow boundary, typically about one boundary-layer thickness long. In doing so a vector quantity $Z = -\sigma(x)(U - U_0)$ is added to the right-hand side of the Navier–Stokes equations. Herein U stands for the vector of conservative variables and U_0 for a given steady basic flow (e.g., laminar or turbulent mean flow). The damping function $\sigma(x)$ is set to zero outside of the sponge region and

$$\sigma(x) = A_s(N_s + 1)(N_s + 2)\frac{(x - x_s)^{N_s}(L_x - x)}{(L_x - x_s)^{N_s+2}} \tag{1}$$

within the sponge region, $x_s \le x \le L_x$, following Israeli and Orszag (1981), where L_x is the streamwise length of the computational domain. The parameters are typically chosen as $A_s = 4$ and $N_s = 3$. The damping effect is not very sensitive to the parameter choice, see Section 4.2. At the outflow plane, inviscid nonreflecting boundary conditions of Thompson (1987) are then applied to U. Guo *et al.* (1996) found that this approach is about as effective as a well-tuned buffer-domain approach, although significantly simpler to implement.

2.3. Grid Generation

For the generation of an analytic mapping of the computational domain onto the physical domain we follow a simple algebraic procedure. We restrict our interest to channel-like geometries where the lower and the upper boundary can be approximated by simple functions and inflow and outflow boundaries are straight. The mapping is nonconformal and the orthogonal partition of the computational domain in general will be mapped onto a nonorthogonal partition of the physical domain. The mapping consists of two steps: (1) the computational domain $(\xi, \zeta) \in [0, 1] \times [0, 1]$ with a uniformly spaced partitioning is mapped onto an intermediate space with nonuniform partitioning $(s, r) \in [0, 1] \times [0, 1]$; (2) the intermediate space (s, r) is mapped onto the physical space (x, z), see Figure 1.

Using a linear blending function between the lower and upper boundaries we define this latter mapping function by

$$x(\xi, \zeta) = (1 - r)\, x_\mathrm{l}(s) + r\, x_\mathrm{u}(s), \qquad z(\xi, \zeta) = (1 - r)\, z_\mathrm{l}(s) + r\, z_\mathrm{u}(s).$$

The indices l and u indicate that the functions are to be taken at the lower and upper boundaries, respectively. The metric coefficients ξ_x, ξ_z, ζ_x, ζ_z are the components of the inverse Jacobi matrix $\partial(\xi, \zeta)/\partial(x, z)$.

For the mesh-point distributions along the parameter lines $s(\xi)$ at the lower and upper boundaries we define

$$s(\xi) = a\xi + b + c_1 \sinh[g_3(\xi)].$$

The following abbreviations are used:

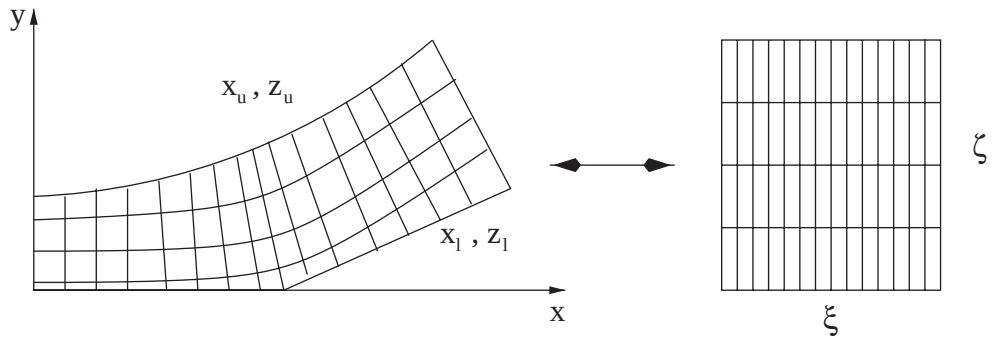

Figure 1. Sketch of the mapping between computational (ξ, ζ) and physical space (x, z).

$$g_3(\xi) = \frac{\xi - c_2}{c_3}, \qquad a = 1 - c_1 \left[\sinh\left(\frac{c_2}{c_3}\right) + \sinh\left(\frac{1 - c_2}{c_3}\right) \right], \qquad b = c_1 \sinh\left(\frac{c_2}{c_3}\right).$$

If we consider compression corner geometries, then c_1 and c_3 are given parameters which tune the grid-point condensation around the given corner point x_c. This point coincides with the zero of $\sinh[g_3(\xi)]$ which is the condition used to compute c_2 by solving $x_c - x(c_2) = 0$ for c_2. Given all parameters, we define the variation of x along the lower or upper boundary in terms of the parameter s as $x(s) = Ls$, where L is the maximum value which x can assume on the lower and upper boundaries, respectively. Having obtained $x(s)$ we get $z(s)$ by

$$z(x) = d_2 \left[x + \frac{1}{d_1} \ln\left(\cosh\left(d_1 x - x_c\right)\right) + d_3 \right]. \tag{2}$$

A corner singularity in the mapping is avoided by prescribing a finite curvature R_c at $(x_c, 0)$. The same is done for the upper boundary but using a much larger curvature radius. Doing so, the mapping between the computational and physical domains remains regular and an order degradation due to a corner singularity can be avoided. Also, the resolution of the highly curved mesh at the corner is facilitated by grid-point condensation. The ramp endpoint is given by $(L, \sin(\varphi)(L - x_c))$, where φ is the ramp angle in physical space (x, z). The parameter d_2 is computed from the condition that $z(L) = \sin(\varphi)(L - x_c)$, which defines d_2 implicitly. The symbols d_1 and d_2 in (2) are defined as

$$d_1 = \frac{(1 + d_2^2)^{3/2}}{R_c}, \qquad d_3 = -\frac{d_2}{d_1} \ln(\cosh(d_1 x_c)).$$

In the transversal direction we introduce the parameter function $r(\zeta)$,

$$r(\zeta) = \frac{z_{1/2}}{z_1} h_1(\zeta) \left[\left(1 - \frac{z_{1/2}}{z_1}\right) + h_1(\zeta) \left(\frac{2 z_{1/2}}{z_1} - 1\right) \right]^{-1},$$

which allows us to stretch the mesh in the transversal direction and to cluster mesh points around z_{mv}. The following abbreviations and parameters are used: e_1 and e_3 control the grid stretching at a point $(0, z_{mv})$, in a similar manner as do d_1 and d_3; about half of the grid points are located between $(0, 0)$ and $(0, z_{1/2})$. $(0, z_1)$ is the upper-left corner point. The auxiliary functions h_1 and h_2 are defined as

$$h_1(\zeta) = c\zeta + d + e_1 \sinh(h_2(\zeta)), \qquad h_2(\zeta) = \frac{\zeta - e_2}{e_3}.$$

The constants c and d are given by

$$c = 1 - e_1 \left[\sinh\left(\frac{1 - e_2}{e_3}\right) + \sinh\left(\frac{e_2}{e_3}\right) \right], \qquad d = e_1 \sinh\left(\frac{e_2}{e_3}\right).$$

Table 1. Typical grid mapping parameters.

Given parameter	Value
φ	18
x_c	50
L	$50 + 40\cos\varphi$
h_1	15
h_2	25
x_{mv}	50
c_1	3
c_2	0.8
$z_{1/2}$	3
z_{mv}	3
d_1	0
d_2	0.8
$R_{c,l}$	0.5
$R_{c,u}$	80
Computed parameter	**Value**
c_2	0.5452
d_2	0.5820

Given z_{mv}, the parameter e_2 is computed from the condition that the argument of $z(\zeta) = z_{mv}$ coincides with $h_2(\zeta) = 0$, i.e., e_2 is obtained by solving $z_{mv} - z_1 r(e_2) = 0$. Typical parameter values for the grid used in Section 5 are given in Table 1.

3. Discretization

3.1. Spatial Discretization

A family of symmetric compact finite-difference schemes with spectral-like resolution has been introduced by Lele (1992). They are being widely used for DNS of transitional and turbulent shear flows, e.g., Adams and Kleiser (1996), Pruett *et al.* (1995), and aeroacoustic problems. As with spectral schemes symmetric compact schemes are sensitive to boundary condition formulation and aliasing errors. The latter is of particular concern for the discretization of convective terms in the Euler or Navier–Stokes equations, where triple products appear in the conservative form of the momentum equations.

In this study we use compact upwind schemes for the discretization of convection terms, a class of which has been derived from a generalized formulation of compact schemes by Adams and Shariff (1996). The schemes have a centered stencil but become upwind biased due to nonsymmetric coefficients. The upwind biasing introduces a certain amount of numerical dissipation at nonresolved wave numbers which allows us to contain aliasing errors. Two particular schemes, called CUHD and CULD, out of that class of compact upwind schemes have been derived by Adams and Shariff (1996). Here we also introduce a slightly more dissipative scheme, obtained using the same procedure as in Adams and Shariff (1996), which we call CUVB. It additionally satisfies Assumption 2.4 of Cockburn and Shu (1994) and thus allows for total variation boundedness in the means by introducing appropriate limiters (a strategy which we do not pursue here for reasons detailed in Adams and Shariff (1996)).

3.2. Fundamentals of Compact Upwind Schemes

Defining $\mathbf{M}_\mathrm{L}$ as the *left-hand* matrix and $\mathbf{M}_\mathrm{R}$ as the *right-hand* matrix of the scheme, which is used to compute spatial derivatives in the convective terms, an approximation $\tilde{\mathbf{f}}'$ of the vector of derivative values of a function f on a grid is obtained from

$$\mathbf{M}_\mathrm{L}\tilde{\mathbf{f}}' = \mathbf{M}_\mathrm{R}\mathbf{f}, \tag{3}$$

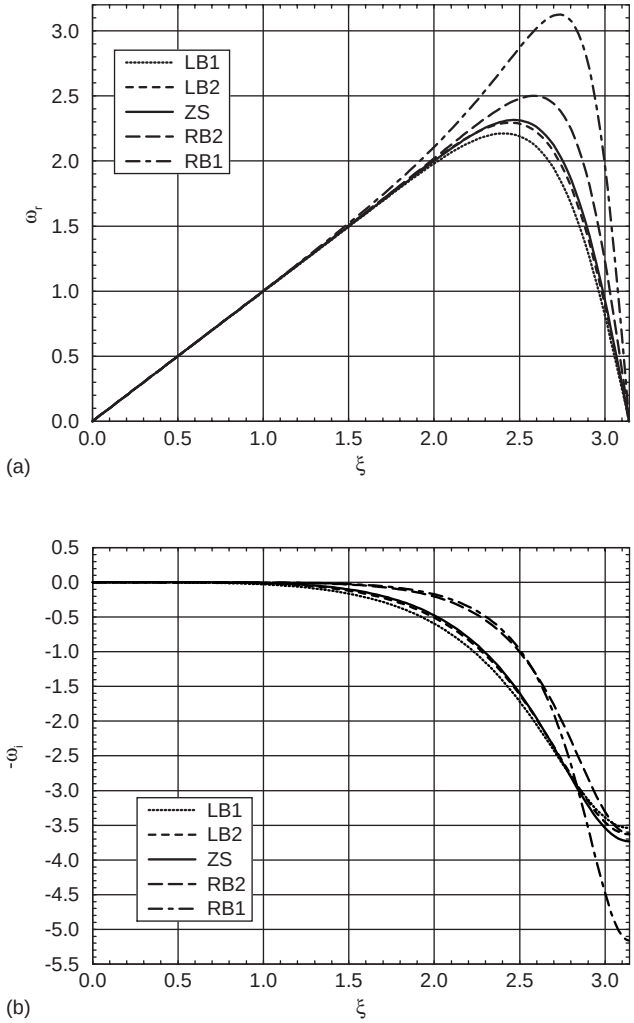

Figure 2. Dispersion (a) and dissipation (b) for scheme CUVB. LB1: scheme at the left boundary; RB1: scheme at the right boundary; LB2: scheme at the next-to-left boundary point; RB2: scheme at the next-to-right boundary point; ZS: scheme at interior points.

where **f** is the vector of function values at the grid points. By numerically minimizing a norm of the dispersive error, under the constraint of nonvanishing dissipation, we derive a scheme of consistently fifth order (fourth-order boundary closures) following the procedure described in Adams and Shariff (1996). The stencil width is five points. For an assessment of the resolution properties we resort to a scalar advection equation, the exact solution of which is $\exp(\mathrm{i}\omega t - \mathrm{i}\xi x)$. By replacing the spatial derivative in the advection equation by its approximation, the frequency $\omega = \xi$ is modified to $\tilde{\omega}(\xi)$, which is complex in general. The real part of $\tilde{\omega}$ represents the numerical dispersion, the imaginary part the numerical dissipation of the scheme. The scheme is designed to be about as dissipative at nonresolved wave numbers as the noncompact upwind scheme used in Rai and Moin (1993), while giving a significantly better representation of the dispersion relation for wave numbers (normalized with the grid-spacing h) up to $\xi \simeq 2$. Figure 2 shows the dispersion relation of scheme CUVB.

For a bounded nonperiodic domain the scheme's stencil needs to become one-sided at the boundaries and telescopes toward the stencil of the inner scheme. Accordingly the scheme's coefficients are different close to and at the boundaries. These boundary closures we call LB1 at the left boundary, LB2 next to the left-boundary, RB2 next to the right boundary, and RB1 at the right boundary, the inner scheme we call ZS. They do not include boundary conditions yet. The contribution of the boundary schemes to the overall resolution is small, they merely needed to be tuned to give stable global discretization. The coefficients of

scheme CUVB are given in the Appendix and more of its properties are discussed, following the guidelines of Adams and Shariff (1996).

3.3. Upwinding for Hyperbolic Conservation laws

Considering a hyperbolic conservation law

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0, \tag{4}$$

the flux-Jacobian $\partial F(U)/\partial U$ is similar to a diagonal matrix $\Lambda = S(\partial F(U)/\partial U)S^{-1}$. The matrices $\mathbf{M}_\mathrm{L}$ and $\mathbf{M}_\mathrm{R}$ in (3) of the upwind scheme, which is stable for positive eigenvalues λ of $\partial F(U)/\partial U$, are called $\mathbf{M}_\mathrm{L}^+$ and $\mathbf{M}_\mathrm{R}^+$. The ones which are stable for negative eigenvalues are called $\mathbf{M}_\mathrm{L}^-$ and $\mathbf{M}_\mathrm{R}^-$. $\mathbf{M}_\mathrm{L}^-$ is obtained from $\mathbf{M}_\mathrm{L}^+$ by multiplication with the unit permutation matrix and $\mathbf{M}_\mathrm{R}^-$ from $\mathbf{M}_\mathrm{R}^+$ by multiplication with the negative unit permutation matrix. A simple form of flux splitting is used. First we compute an approximation P_N of $\partial F/\partial x$ by each of the schemes:

$$P_N^+ = \mathbf{M}_\mathrm{L}^{+\,-1}\mathbf{M}_\mathrm{R}^+ F, \qquad P_N^- = \mathbf{M}_\mathrm{L}^{-\,-1}\mathbf{M}_\mathrm{R}^- F.$$

These approximate flux derivatives are then projected onto the local characteristic directions by

$$C_N^+ = S^{-1}P_N^+, \qquad C_N^- = S^{-1}P_N^-.$$

Defining the sign function to return the sign of λ and zero if $\lambda = 0$ we compute

$$C_N = \tfrac{1}{2}[(1 + \mathrm{sign}(\Lambda))C_N^+ + (1 - \mathrm{sign}(\Lambda))C_N^-]$$

which is then projected back onto the computational space to give

$$P_N = SC_N.$$

This procedure results in a stable upwind approximation for the flux derivative. The implicitly added numerical dissipation is apparent from an equivalent formulation:

$$P_N = \tfrac{1}{2}(P_N^+ + P_N^-) + \tfrac{1}{2}[S\ \mathrm{sign}(\Lambda)S^{-1}(P_N^+ - P_N^-)].$$

The last term is proportional to the scheme's truncation error, which is dominated for the present schemes by a sixth-order derivative term (Adams and Shariff, 1996, equation (8)) and resembles a hyperviscosity of sixth order.

3.4. Compact Essentially Nonoscillatory Coupling

Linear compact schemes are not suited for the computation of weak solutions of advection problems which contain discontinuities. Here, we couple the nonconservative compact upwind scheme with a conservative shock-capturing scheme around discontinuities. As has been pointed out in Adams and Shariff (1996), heuristic evidence for this type of hybridization shows that the main proposition of Hou and Le Floch (1994) holds for the compact upwind schemes, which are used here, as well. This proposition asserts that a certain class of nonconservative schemes gives solutions which converge to the viscosity solution of a conservation law if the schemes are corrected toward a conservative scheme in a neighborhood of discontinuities (see also the Appendix).

We use a finite-difference type essentially nonoscillatory (ENO) scheme as proposed by Shu and Osher (1989). In a first step, a discontinuity detection algorithm scans the instantaneous solution. A simple detection algorithm is employed: if a flux gradient is larger than a certain threshold and assumes a local maximum, the respective cell is marked to be treated by the ENO scheme. The marked regions are padded by a certain additional number of cells to allow the compact scheme on each side of the discontinuity to decouple and to allow the ENO scheme to expand its stencil fully. The fluxes at the centers of the marked cells are calculated

with the ENO procedure, and the flux derivatives at the corresponding cell faces are reconstructed. Scheme (3) is modified to return the ENO flux derivatives at the marked cell faces and the compact flux derivatives at all other points. For details see Adams and Shariff (1996).

It has been shown among others by Casper and Carpenter (1995) and Arora and Roe (1997) that shock-capturing schemes suffer from a post-shock reduction of accuracy, which only can be avoided by subcell resolution. Tests of the hybrid scheme (Adams and Shariff, 1996, Sections 6.2 and 6.3) and of a pure ENO scheme by Mahesh *et al.* (1996) demonstrate that the effect of the postshock oscillations on small-scale fluctuations is acceptably small, provided that the fluctuations are sufficiently well resolved.

The actual implementation in a DNS code with a reasonable capability of vector optimization is more involved. An efficient implementation of the compact ENO coupling is possible by using a correction formula for the inverse of $\mathbf{M}_\mathrm{L}$. We consider the one-dimensional, one-component problem. Given the flux F on the grid $\{x_j\}$, its derivative for x is approximated by

$$\frac{\partial F}{\partial x} \doteq P_N[F] = \mathbf{M}_\mathrm{L}^{-1}\mathbf{M}_\mathrm{R}F. \tag{5}$$

Assume that $\{x\}_\mathrm{E} = \{x_p,\ldots,x_q\}_1 \cup \cdots \cup \{x_r,\ldots,x_s\}_{n_\mathrm{E}}$ is the union of contiguous regions of points where the flux derivatives are approximated by the ENO scheme. If the shock-detection algorithm has detected a point x_i to be treated by the ENO scheme, we set the ith component of a mask Υ equal to 1. The other entries of Υ are 0. The vector Υ has n_E unity blocks with dimensions larger or equal to $(2N_\mathrm{sep}+1)$ each, where N_sep is the dimension of the padding on each side of an ENO region (Adams and Shariff, 1996). Whenever $\Upsilon_i = 1$ for a certain grid point, then $P_N[F]_i = P_N^\mathrm{ENO}[F]_i$ is computed by the ENO scheme.

The effect of the compact ENO coupling on (5) is that the ith component of $\mathbf{M}_\mathrm{R}F$ is replaced by the flux derivative at i calculated with the ENO scheme whenever $\Upsilon_i = 1$. The row i of $\mathbf{M}_\mathrm{L}$ has then to become unity so that the ENO flux derivative $P_N^\mathrm{ENO}[F]$ is returned exactly. We define a correction matrix from the product **BCD** of three matrices which changes the rows i of $\mathbf{M}_\mathrm{L}$ to unity whenever $\Upsilon_i = 1$. It can by decomposed into the matrices **B**, **D**, and **C**, the dimensions of which are given below. With this definition the fundamental equation for the computation of flux derivatives for the hybrid scheme can be written as

$$(\mathbf{M}_\mathrm{L} - \mathbf{BDC})P_N[F] = \mathbf{M}_\mathrm{R}F + \Upsilon(P_N^\mathrm{ENO}[F] - \mathbf{M}_\mathrm{R}F). \tag{6}$$

The rank of the correction matrix **BCD** is $\sum_{\nu=1}^{n_\mathrm{E}} N_\nu = m_\mathrm{E}$. It is evident that (6) returns the ENO flux derivatives exactly at points i whenever $\Upsilon_i = 1$.

To solve (6) efficiently we make use of an identity by Frobenius and Schur (Zurmühl and Falk, 1984, pp. 308, 312) which allows us to compute $(\mathbf{M}_\mathrm{L} - \mathbf{BDC})^{-1}$ from the inverse of $\mathbf{M}_\mathrm{L}$, corrected by the inverse of a rank m_E matrix **R**. Since the precomputed inverse of $\mathbf{M}_\mathrm{L}$ can be used this procedure is, for multidimensional problems, more efficient than inverting the left-hand side matrix of (6), if $m_\mathrm{E} \ll N$.

The matrices **B**, **D**, and **C** are defined as follows :

$$\underbrace{\mathbf{B}}_{N\times m_\mathrm{E}} = \sum_{\nu=1}^{m_\mathrm{E}} \Upsilon \mathbf{e}_\nu^T, \qquad \underbrace{\mathbf{D}}_{m_\mathrm{E}\times m_\mathrm{E}} = \underbrace{\mathbf{I}}_{m_\mathrm{E}\times m_\mathrm{E}} = \mathbf{I}_{m_\mathrm{E}}, \qquad \underbrace{\mathbf{C}}_{m_\mathrm{E}\times N} = \mathbf{B}^T(\mathbf{M}_\mathrm{L} - \mathbf{I}).$$

Here we define $\mathbf{e}_\nu$ as the m_E-component vector, the νth component of which is equal to unity, the rest being zero.

The solution algorithm for (6) is the following, according to Zurmühl and Falk (1984):

(0) calculate the uncorrected solution vector **y** from

$$\mathbf{M}_\mathrm{L}\mathbf{y} = \mathbf{M}_\mathrm{R}F + \Upsilon(P_N^\mathrm{ENO}[F] - \mathbf{M}_\mathrm{R}F)$$

by direct inversion using the precomputed LU-decomposition of $\mathbf{M}_\mathrm{L}$;

(1) compute matrix **V** from

$$\mathbf{M}_\mathrm{L}\mathbf{V} = \mathbf{B}$$

by direct inversion using the precomputed LU-decomposition of $\mathbf{M}_\mathrm{L}$;

(2) generate the rank m_E correction matrix **R** from

$$\mathbf{R} = \mathbf{I}_{m_\mathrm{E}} - \mathbf{CV} ;$$

(3) get the solution correction vector **z** from

$$\mathbf{Rz} = \mathbf{Cy}$$

(note that **R** is usually fully occupied so that this procedure is only efficient if $m_{\mathrm{E}} << N$);

(4) find the solution vector from

$$P_N[F] = \mathbf{y} + \mathbf{Vz}.$$

For a multidimensional problem all points in index planes normal to the derivative direction are gathered and a vector loop is spanned over these points.

4. Code Validation

DNS codes are required to resolve all scales appearing in a flow problem. So it seems reasonable to perform two different kinds of test, the first of which is to assess how well fluctuations about a mean flow are represented, the second is to see how well the mean flow itself can be computed. Concerning the first, we follow a procedure which has been used by Adams and Kleiser (1996), where linear stability theory is used as a reference. For the second, we compare with the steady-state computations and experimental results for laminar compression ramp flow of Simeonides *et al.* (1994), in this case the ENO scheme is active around the shock.

4.1. Temporally Growing Instability

To assess the effectiveness of the compact upwind schemes the code is validated by comparison with linear stability theory. The laminar mean flow is superimposed with an eigensolution of the linearized Navier–Stokes equations. For low amplitudes the instability should evolve in accordance with linear theory. We derive a growth rate from the kinetic energy content of the instability wave as was done by Adams and Kleiser (1996). The growth rate which is observed in the DNS is required to match linear theory.

We consider the case of a temporally growing (streamwise and spanwise periodic) three-dimensional instability in a boundary layer. The parameters are given in Table 2. The reference length is the laminar displacement thickness, α and β are the streamwise and spanwise wave numbers, respectively, and A is the initial maximum amplitude of the streamwise velocity fluctuation. The exact growth rate from linear stability theory (LST) is $\omega_i = 6.433554 \cdot 10^{-3}$. In Table 3 the growth rate after one time step (imaginary

Table 2. Parameters for temporally growing instability.

M_∞	3
Re_{δ_1}	7500
α	0.25
β	0.45
A	10^{-4}

Table 3. Growth rates for a linear instability wave, DNS compared with LST.

	Scheme	
Discretization	CUHD	CUVB
$10 \times 10 \times 50$	6.42030×10^{-3}	6.16112×10^{-3}
$20 \times 20 \times 50$	6.43817×10^{-3}	6.42900×10^{-3}
$10 \times 10 \times 100$	6.41529×10^{-3}	6.15629×10^{-3}
$20 \times 20 \times 100$	6.43314×10^{-3}	6.42412×10^{-3}
$40 \times 40 \times 100$	6.43363×10^{-3}	6.43328×10^{-3}
$40 \times 40 \times 200$	6.43352×10^{-3}	6.43320×10^{-3}
Linear theory	6.43355×10^{-3}	

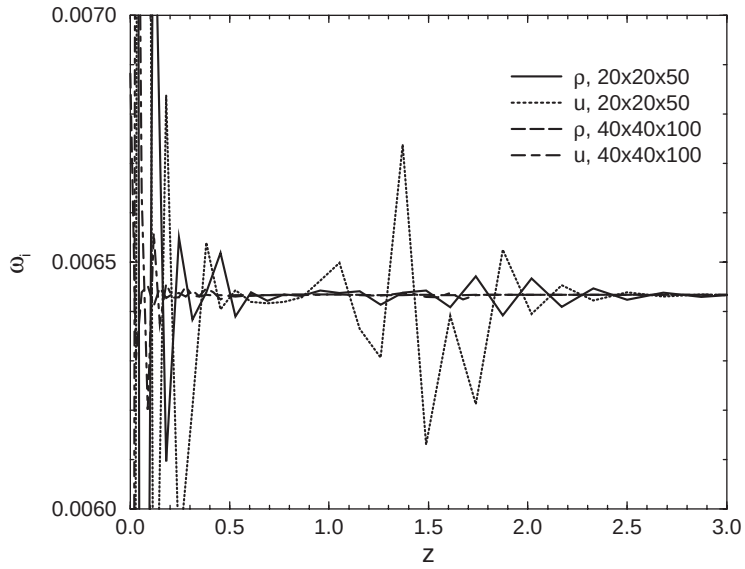

Figure 3. Local growth rate for scheme CUHD.

part of the complex temporal eigenvalue) with schemes CUHD and CUVB are shown (note that since the solution is shock-free the ENO scheme is inactive). Since here we are concerned only with the accuracy of the spatial discretization only one time step is performed. Integration over longer time intervals is considered in Section 4.2.

Figures 3, 4 and 5, 6 show the local growth rate and frequency distribution across the boundary layer for schemes CUHD and CUVB, respectively. We use the fact that the eigenfunctions maintain their shape in a parallel boundary layer. Note that the computation of growth rate and frequency from the eigenfunctions is ill-conditioned near the wall, where the amplitude becomes very small. This affects in particular cases with low wall-normal resolution. An increased resolution in z allows also for a better resolution of the boundary-layer edge where the mean-flow profile curvature is maximum. Strong near-wall oscillations are

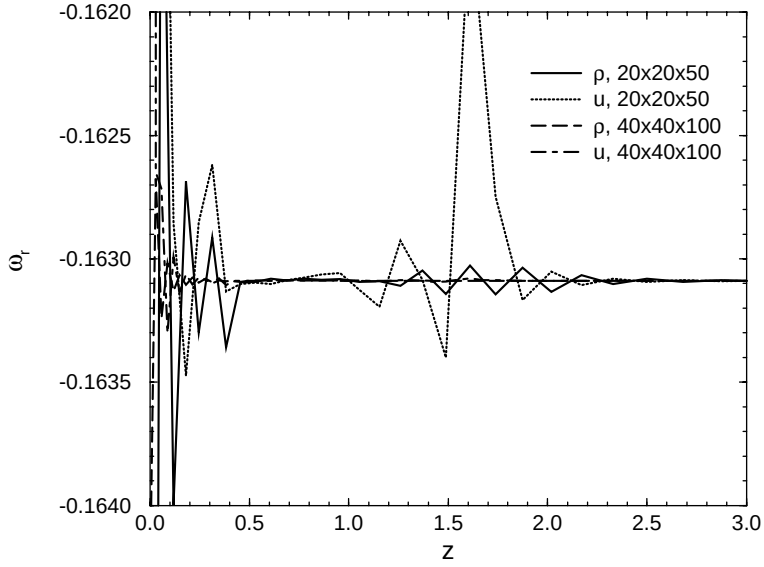

Figure 4. Local frequency for scheme CUHD.

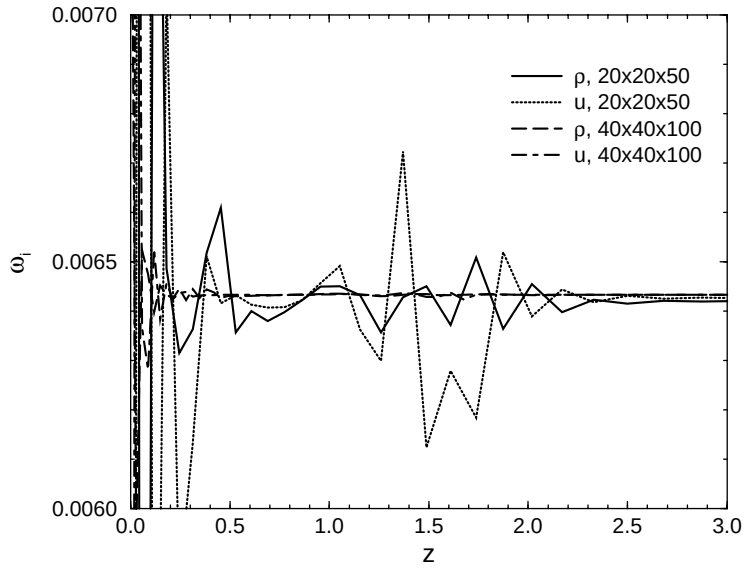

Figure 5. Local growth rate for scheme CUVB.

caused by numerical round-off due to small local perturbation amplitudes. From the figures it is evident that, for sufficient resolution, the numerical solution accurately resembles linear theory.

4.2. Spatially Growing Instability

A similar procedure as before is used to assess the effectiveness of the outflow boundary treatment and the accuracy of the time integration. A two-dimensional spatially evolving second mode instability wave, see Adams and Kleiser (1996), of a maximum streamwise velocity amplitude A is superimposed on a laminar boundary layer. The test-case parameters are given in Table 4: α_r is the streamwise wave number, $-\alpha_i$ is the spatial growth-rate, and ω is the frequency of the instability wave. The streamwise size of the computational

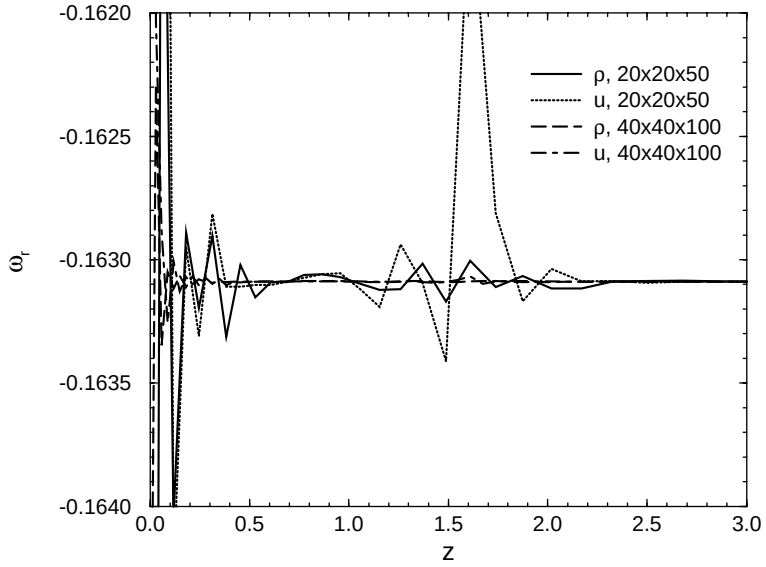

Figure 6. Local frequency for scheme CUVB.

Table 4. Parameters for spatially growing instability.

M_∞	4.5
Re_{δ_1}	10000
α_r	2.25
ω	2.05
$-\alpha_i$	0.030821
A	10^{-4}

domain is about eight primary-instability wavelengths. Since the efficiency of the spanwise discretization has been assessed by the previous test we consider here only two-dimensional flow and the spanwise wave number β is zero. A parallel basic flow is enforced by adding a forcing term to cancel its residual, see Adams and Kleiser (1996). The local growth rate at z = 0.5 is computed by a Fourier transform of 48 samples from data spanning over 8 primary-instability periods. Resolution is N_x = 160, N_z = 50 and N_x = 320, N_z = 100 cells, respectively. For these computations the sponge parameters, equation (1), are N_s = 3 and A = 4. A change of the sponge parameters to N_s = 6 or A = 10 has been found to have only a minor effect on the efficiency of the sponge layer. Figure 7 gives an illustration of the sponge-layer effect. It can be seen that the density wave is suppressed in a narrow region close to the outflow boundary without distorting the oncoming wave.

The amplitude of an instability wave is a much less sensitive measure of discretization accuracy than the local growth rate, which we use in the following. Figures 8 and 9 compare the effect of grid refinement for schemes CUHD and CUVB, respectively. For comparison, the growth rate as predicted by linear stability theory is given in the figure caption. The sponge layer at the end of the computational domain is indicated by gray shading. Note that the growth rate is ill-defined in the sponge region since the perturbation amplitude is forced toward zero. A small inflow transient region appears as a consequence of imposing the time evolution of the instability from an LST result. This transient diminishes with increasing resolution. The issue whether the outflow boundary treatment could cause spurious reflections which increase in time is addressed in Figure 10. Increasing the integration time from 1 period to 80 periods shows no increased error in the local growth rate so that the existence of spurious amplified waves can be excluded. Since scheme CUVB behaves essentially as scheme CUHD we restrict the following sponge-layer tests to scheme CUHD.

The error in the local growth rate also remains unchanged at a given position for an unchanged resolution if the computational domain size is doubled, as can be seen from Figure 11. This means that there is no artificial coupling between the inflow and outflow boundaries. The same holds when the sponge region is

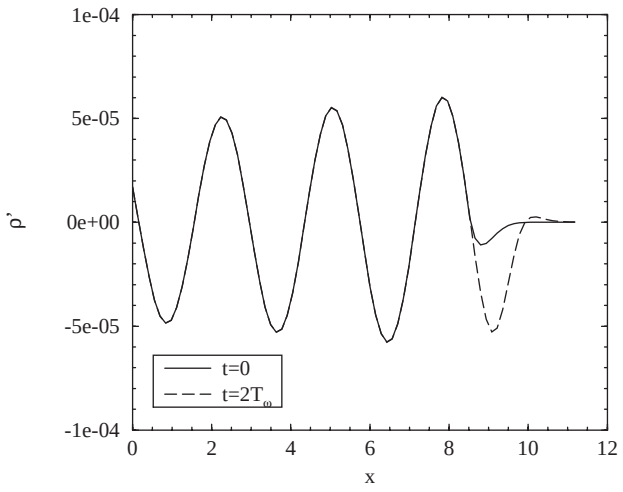

Figure 7. Density fluctuation at t = 0 and after two periods.

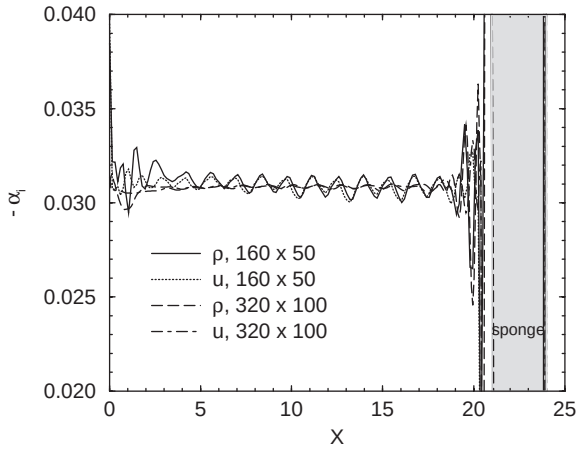

Figure 8. Local growth rate for scheme CUHD with N_x = 160, N_z = 50 and with N_x = 320, N_z = 100; LST: $-\alpha_i$ = 0.030821.

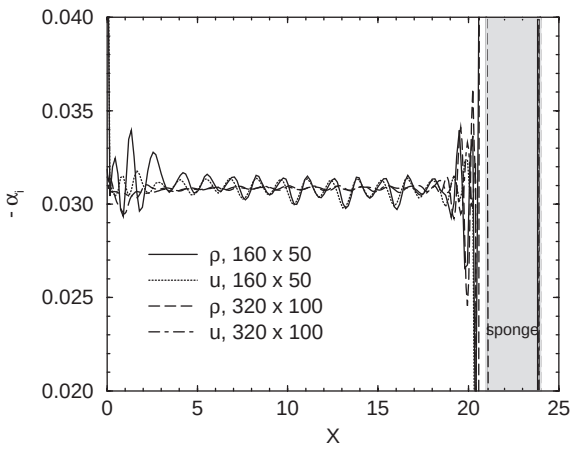

Figure 9. Local growth rate for scheme CUVB with N_x = 160, N_z = 50 and with N_x = 320, N_z = 100; LST: $-\alpha_i$ = 0.030821.

extended so that it starts at x = 18, the growth-rate distribution in the valid part of the domain remains essentially unchanged (Figure 11).

Without showing figures we note that if the fluctuation amplitude is increased by two orders of magnitude, then weak nonlinear effects cause larger excursions of the local growth rate from the linear value (maximum about 10% error) but there is no noticeable reflection increase from the sponge region.

4.3. Steady-State Solution

As an example for the second kind of test we compare our results with the experimental and computational results of Simeonides *et al.* (1994). As a test case we select their 7.5° ramp at M_∞ = 6. The flow is laminar over the whole extent of the ramp. Figure 12 shows the residual history (rms value of the density derivative). After 13 000 time steps with a third-order pure ENO scheme the computation is continued for an additional 16 000 time steps using the hybrid scheme (combining the fifth order compact scheme with a fourth-order local-Lax–Friedrichs-flux ENO scheme). About 1.5% of all points are, on average, treated by the ENO procedure when the flux derivatives are computed with the hybrid scheme. We emphasize that our scheme is time-accurate and of low overall dissipation. Only for validation purposes is it used to obtain a steady-state

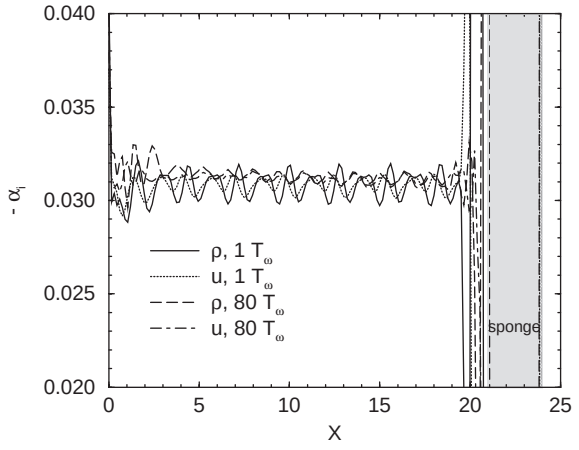

Figure 10. Local growth rate for scheme CUHD with N_x = 160, N_z = 50 after 1 period and after 80 periods; LST: $-\alpha_i$ = 0.030821.

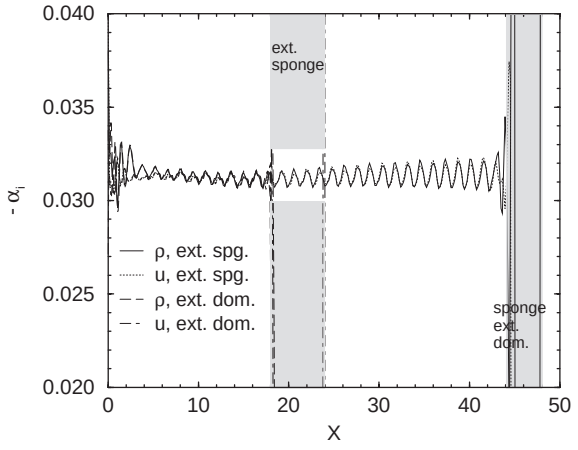

Figure 11. Local growth rate after 80 periods for scheme CUHD with N_x = 320, N_z = 50 and extended (doubled) domain and with N_x = 160, N_z = 50 and extended sponge region.

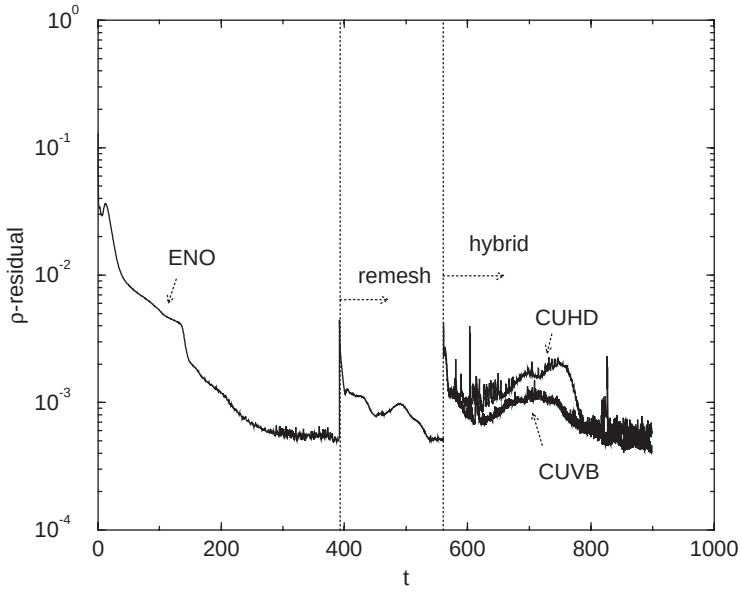

Figure 12. Residual history (rms of $\partial\rho/\partial t$) for schemes CUHD and CUVB.

solution. The low dissipation is reflected in the large number of time steps needed to reach a steady-state solution.

Note also that we do not include the leading edge within the computational domain, but prescribe a similarity boundary layer at the inflow, so we observe some inflow transient. This is done to match the procedure used in later DNS computations. The agreement of the results with the finest-mesh-level computational data and experimental data of Simeonides *et al.* (1994) is reasonable (Figure 13). No attempt has been made to improve the results further by continuing the computation or by grid refinement.

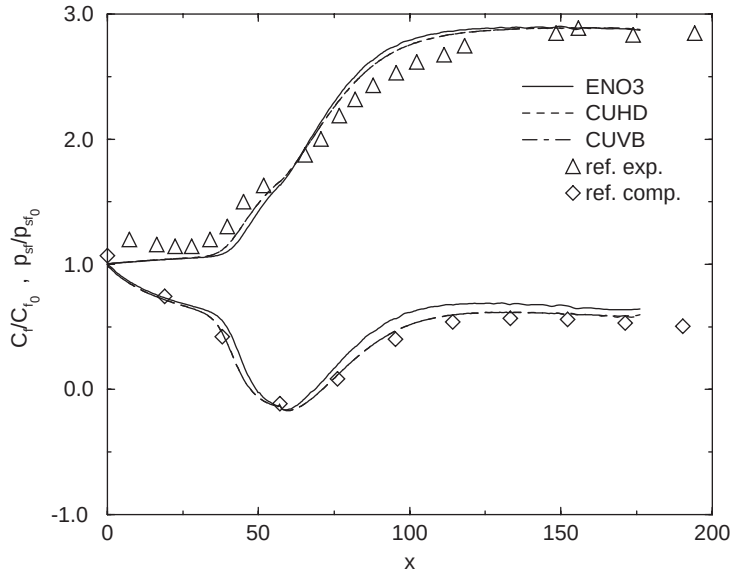

Figure 13. Skin friction and surface pressure compared with computational and experimental results of Simeonides *et al.* (1994).

5. Turbulent Supersonic Compression Ramp

In this section we discuss some technical issues concerning DNS of the turbulent boundary layer along a compression ramp and demonstrate the effectiveness of the simulation procedure using part of a DNS database.

We consider a plane compression ramp with a deflection angle of 18°. The free-stream Mach number of the flow along this configuration is M_∞ = 3 and the momentum-thickness Reynolds number at the inflow is Re_θ = 1685. As reference quantities the free-stream values of u, ρ, T, and μ and the displacement thickness of the mean boundary layer at the inflow are used. During a preliminary simulation its has been found that inflow data have a significant impact on the large-scale unsteady behavior (Adams, 1997). Here, we generate inflow data from a separate spatial boundary layer DNS, which in turn is fed with temporal DNS inflow data as described in Adams (1997). At a cross-flow plane, about six mean-boundary-layer thicknesses downstream of the inflow, the conservative variables are sampled 1000 times in an interval Δt = 167, corresponding to about 2.5 "flow-through" times (the time the free-stream flow needs to pass the ramp configuration). The sampled data are made periodic in time by smoothly blending the interval ends. Contrary to the procedure in Adams (1997), where the inflow data were sampled only over about Δt = 6 this procedure allows the flow in the interior to be sufficiently decorrelated (about 150 large-eddy turnover times) before the inflow data repeat.

Figure 14 shows a side view of the computational mesh (each 10th grid line is shown). The streamwise and spanwise mesh resolution in terms of wall units at the inflow is on the average $\Delta_x^+ \simeq 6$ and $\Delta_y^+ \simeq 3.3$, respectively. The first grid point off the wall in the wall-normal direction is at the inflow at $\Delta_z^+(z_1) = 1.4$. Within the mean boundary layer at the inflow there are 97 of 181 grid points which give on the average $\Delta_z^+ \simeq 2.3$.

An issue mentioned in Section 2.3 was the effect of the corner-rounding employed to avoid a coordinate-mapping singularity. Figure 15 shows a close up of the corner region with contours of spanwise vorticity ω_y (color coding stops to indicate the mean boundary-layer edge). It is apparent that the curved corner segment is small compared with the flow structures (note that the mesh as shown is half as fine than for the computation).

Figure 16 shows a visualization of an instantaneous shock surface (div(**u**) = -0.4) where the bottom and back of the computational domain are color coded according to the local density. It is evident that the shock forms essentially outside of the boundary layer where its tip fringes into the boundary layer. We choose an isothermal-wall boundary condition, since it appears more reasonable to expect the wall (as the boundary of a body with relatively high heat capacity) to be unable to react to fast temperature changes due to turbulent fluctuations. Experimental data of Zheltovodov *et al.*, see Settles and Dodson (1991), support the assumption

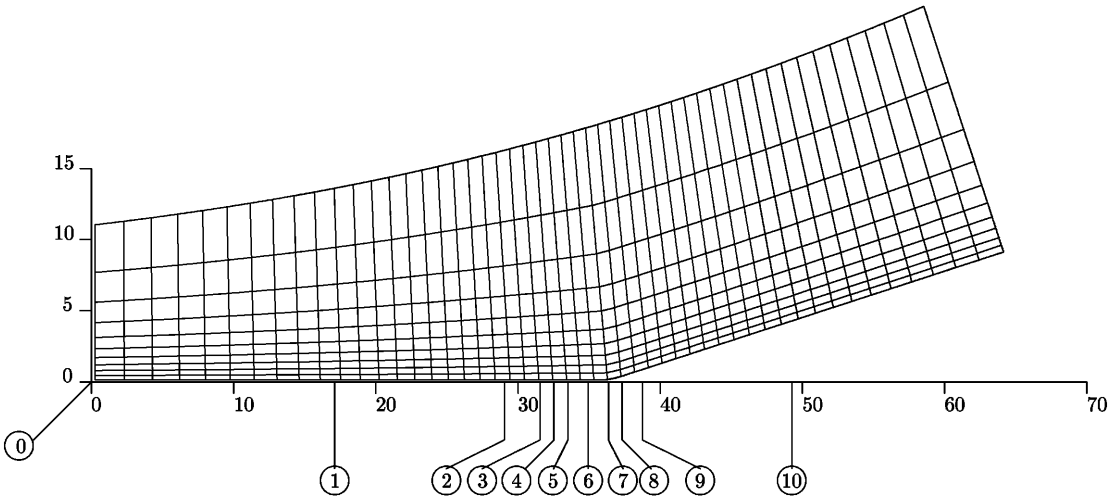

Figure 14. Computational mesh (each 10th grid line); numbers in circles indicate stations where profiles are shown in Figure 18.

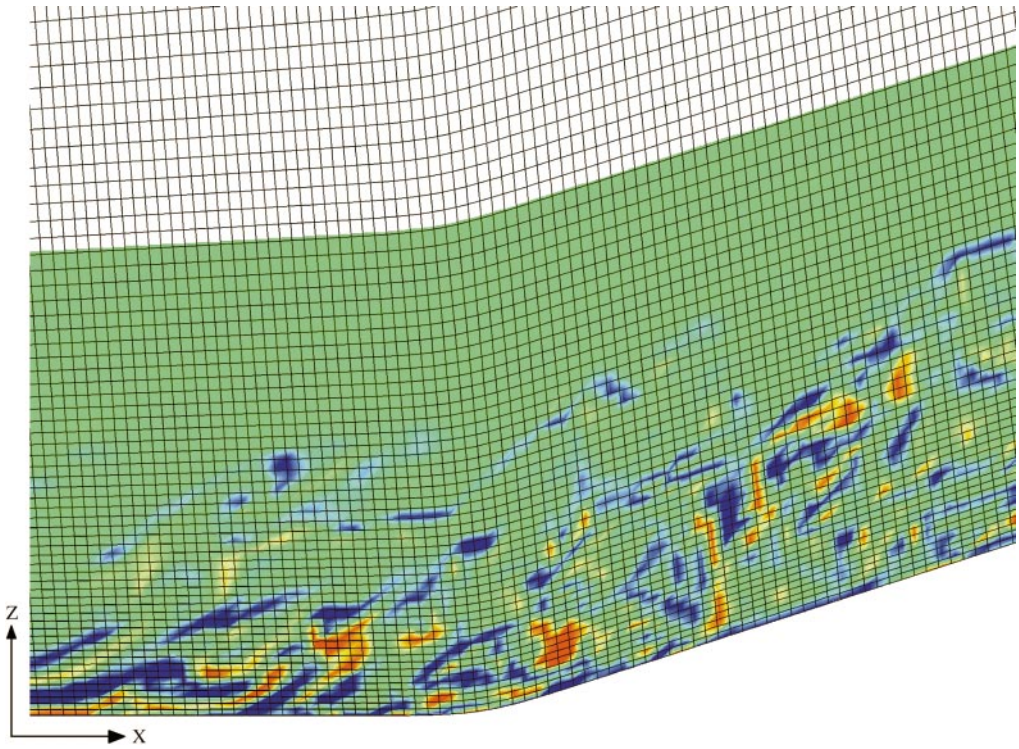

Figure 15. Close-up at the corner, contours of ω_y.

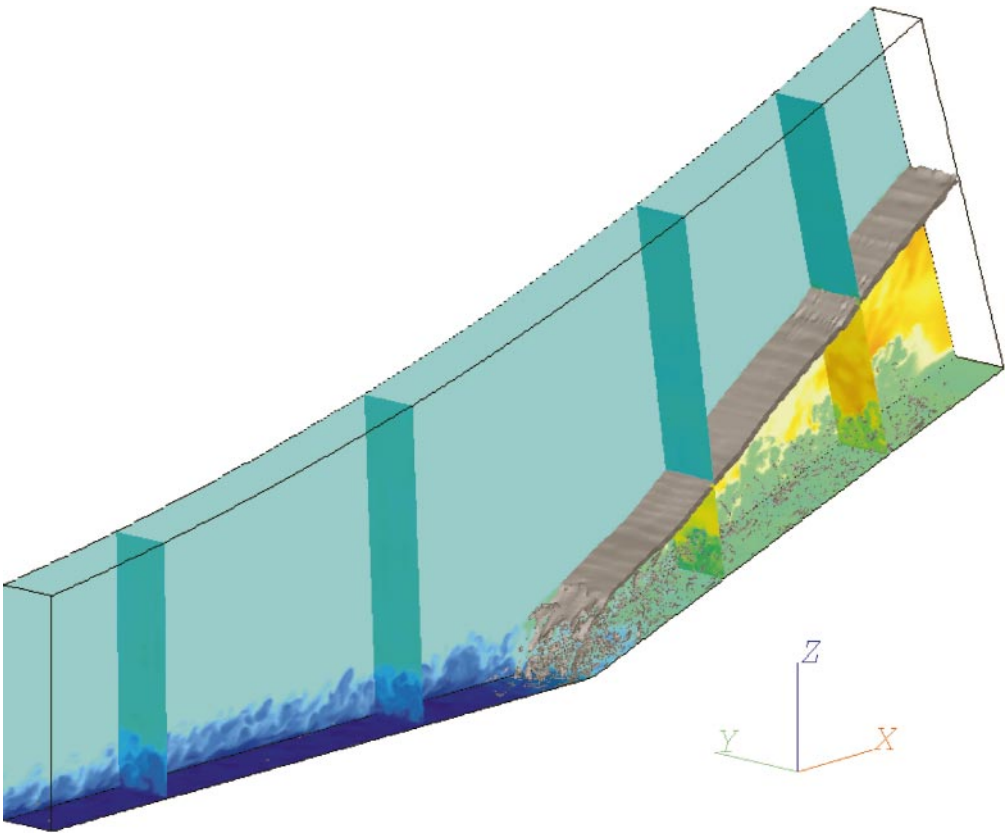

Figure 16. Density at the wall, in the plane $y = 2.9$ and shock-surface with div($\mathbf{u}$) = -0.4.

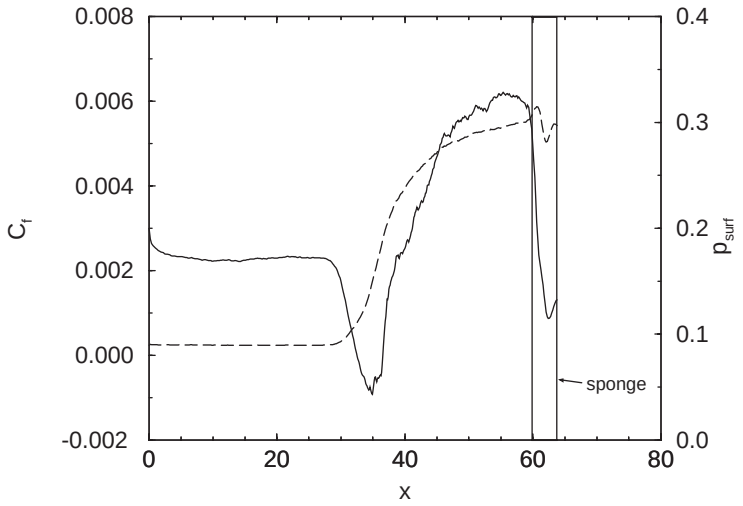

Figure 17. Skin friction and surface pressure for 150 samples.

of a constant wall temperature. Since the wall temperature is kept constant, density remains low close to the wall but increases behind the deflection shock.

From 150 time samples spanning a time interval Δt = 172 spanwise-averaged and time-sampled distributions of the skin friction coefficient C_f and the surface pressure p_surf have been obtained and are displayed in Figure 17. The C_f distribution indicates an incipient separation around the corner, which is located at x = 36.23. The evolution of Favre-averaged longitudinal velocity profiles is shown in Figure 18. The downstream positions of the profiles are indicated by the numbers 0–10 and correspond to the markers in Figure 14. Note that only the part of profile 10 which is beneath the shock is shown, so that it approaches the postshock value of the free-stream velocity.

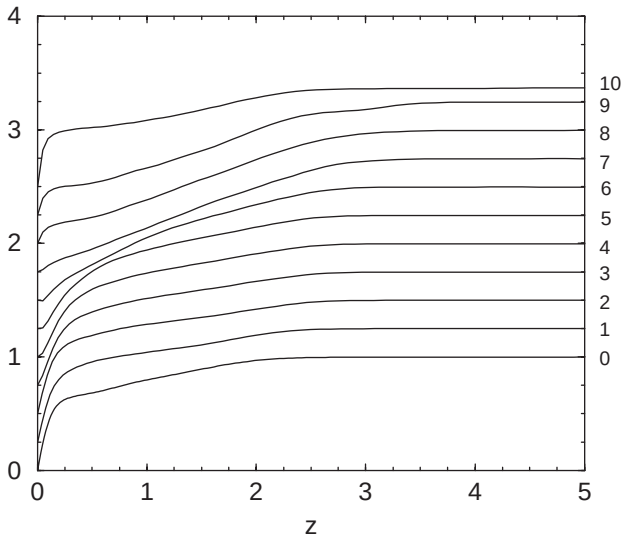

Figure 18. Favre-averaged (150 samples) longitudinal velocity profiles at different downstream stations as marked in Figure 14 (profiles are staggered by an offset of 0.25 for clarity).

6. Concluding Remarks

A novel hybrid shock-capturing scheme is used to discretize the compressible Navier–Stokes equations with an accuracy required by DNS for transitional and turbulent flow. The prominent feature of this procedure is a new compact upwind scheme of consistently high order (fifth order) which is coupled with an ENO scheme to capture shocks. The coupling makes the resulting hybrid scheme nonlinear. An efficient coupling algorithm has been devised which essentially works as a correction to the underlying linear compact upwind scheme. The hybrid scheme is used as a building block of a solution method for the unsteady compressible Navier–Stokes equations. The numerical method has been validated by comparison with linear stability theory and with steady-state solutions.

DNS data for the turbulent boundary layer along a compression ramp show reasonable results. A quantitative comparison with available experimental results is not possible for the foreseeable future due to the fact that experimental Reynolds numbers are significantly higher than those which can be achieved with DNS. Further DNS-data analysis is in progress.

Appendix

For the nomenclature of this appendix we refer to Adams and Shariff (1996). For completeness all relevant parameters, stability analysis, and validation results for scheme CUVB are given here, along the same line as in Adams and Shariff (1996) for schemes CUHD and CULD. Unlike schemes CUHD and CULD of Adams and Shariff (1996), the inner scheme of CUVB also satisfies Assumption 2.4 of Cockburn and Shu (1994).

The parameters of the target functional, equation (31) of Adams and Shariff (1996) for scheme CUVB are given in Table 5. Table 6 summarizes the initial guesses for optimization. Constrained numerical optimization as described in Adams and Shariff (1996) gives the coefficients of scheme CUVB (Table 7). The truncation

Table 5. Target functional parameters.

Scheme*	α	β	Ξ	γ	$\Pi(\xi_0)$
ZS	5	0	2.3	0.04	$\omega_i(2)^2$
LB1	1	0	2.1	0	—
LB2	1	0	2.3	0	—
RB2	1	0	2.3	0	—
RB1	1	0	2.3	0	—

* ZS stands for the scheme at the interior points, LB1 for the scheme at the left boundary, RB1 for the scheme at the right boundary, LB2 for the scheme at the next-to-left boundary point, and RB2 for the scheme at the next-to-right boundary point.

Table 6. Initial guesses for optimization. Abbreviations same as in Table 5.

Scheme	Coefficient	CUVB
ZS	a_{-2}	0.2
	a_0	0.2
	a_2	0.2
LB1	a_3	-1
	a_4	-1
LB2	a_2	0
	a_3	-0.2
RB2	a_{-3}	1.
	a_{-2}	$-1.$
RB1	a_{-4}	$-1.$
	a_{-3}	$-1.$

Table 7. Coefficients for scheme CUVB. Abbreviations same as in Table 5.

Scheme			$\alpha_{\mu_1},\ldots,\alpha_{\mu_r}$		
ZS	−0.152087544263	0.2375084218594	1	−0.104175088526	0.018754210929673
LB1	1	1.344856712172	−1.655143287828	—	—
LB2	−1.384173201496	1	6.620636755258	−2.122934969145	—
RB2	15.74839515424	27.04931646417	1	−0.3211206221354	—
RB1	4.850969558221	7.826957363797	1	—	—
Scheme			$a_{\nu_1},\ldots,a_{\nu_r}$		
ZS	0.354175088526	−1.550033687437	1.025050531156	0.1832996458959	−0.012491578140655
LB1	−2.557476118695	3.982714931742	−0.9827149317416	−0.4425238813046	0
LB2	3.362325146817	−11.84525146073	8.836922058206	0.2990810938894	−0.6530768381839
RB2	−1.846602572079	−30.2283794418	21.15923086495	12.27649588754	−1.36074473861
RB1	0.0020010162019952	−0.6538343569155	−8.740436045696	5.756444175312	3.635825211098

Table 8. Truncation error estimate. Abbreviations same as in Table 5.

Scheme	$\tilde{\mathcal{E}}_c$
ZS	0.104455
LB1	2.00482
LB2	3.35575
RB2	9.33802
RB1	7.16414

error estimate $\tilde{\mathcal{E}}_c$ from equation (10) of Adams and Shariff (1996) for the different schemes is given in Table 8.

Figure 19 shows the pseudo-eigenspectra for three different realizations of **E** for scheme CUVB with N = 40 and N = 160 cells. Here, **E** is a matrix of unit norm and with random elements. Since the maximum absolute value of the real parts of the pseudo-eigenvalues is bounded by a linear function, Figure 20, evidence for algebraic stability is given (Adams and Shariff, 1996).

If the solution develops discontinuities the hybrid scheme switches locally to an ENO scheme as described by Adams and Shariff (1996). For the hybrid scheme with CUVB as the underlying compact scheme the same tests as in Section 6 of Adams and Shariff (1996) have been performed and the results are similar to

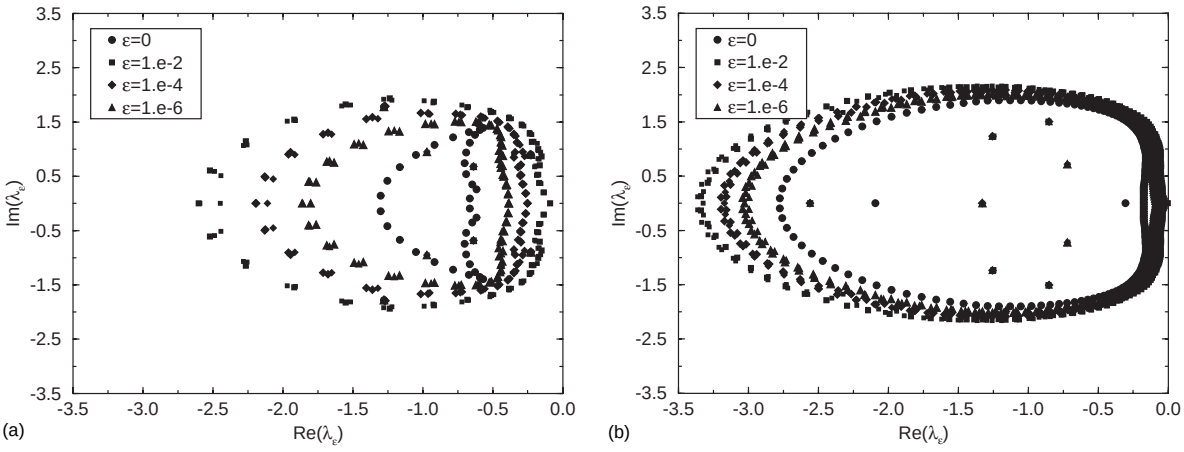

Figure 19. ε-pseudo-eigenspectra for three different realizations of **E** for each ε with N = 40 (a) and N = 160 (b).

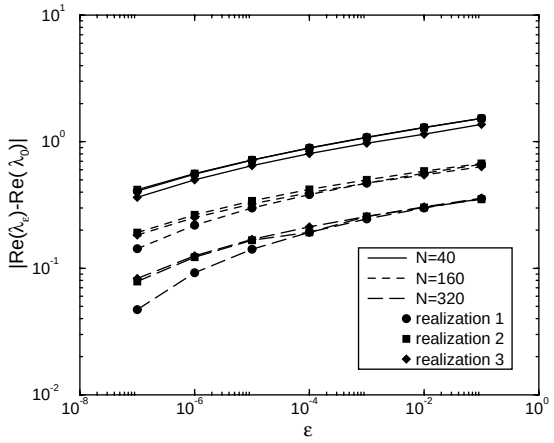

Figure 20. Bound of $|\Re(\lambda_\varepsilon)|$.

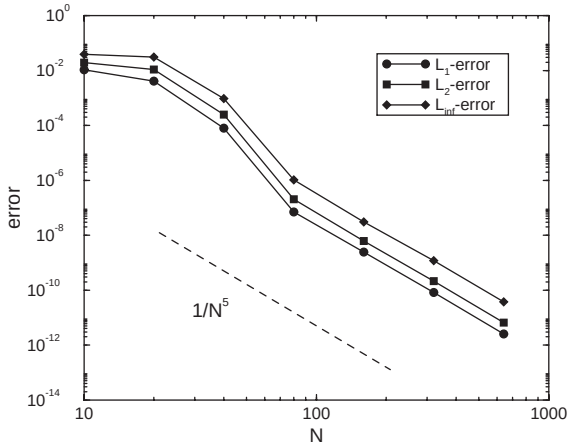

Figure 21. Error norms for Burgers' equation at $t = 2/\pi$.

those with CUHD. As an example the convergence rate of scheme CUVB for the test with Burgers' equation is shown in Figure 21.

The coefficients of schemes CUVB, CUHD, and CULD can be obtained as a Fortran source by anonymous ftp from spartakus.ifd.mavt.ethz.ch.

Acknowledgment

Helpful discussions with Dr. K. Mahesh, Center for Turbulence Research, Stanford University, and NASA Ames Research Center are greatly appreciated. Thanks are also due to Dr. L. Kleiser and Dr. C. Härtel, Institute of Fluid Dynamics, ETH Zürich, for critically reviewing a draft of this paper. The main part of this work has been conducted while I was a postdoctoral research fellow with the Center for Turbulence Research, Stanford University and NASA Ames Research Center, California; I am indebted to Prof. P. Moin for his support. Part of this work has been supported by the CRAY Research, Inc. Computations were performed at the Swiss Center for Scientific Computing.

References

Adams, N.A. (1997). DNS of shock boundary-layer interaction—preliminary results for compression ramp flow. In *CTR Annual Research Briefs 1997*. Center for Turbulence Research, Stanford University and NASA Ames Research Center, Stanford, California, 329–338.

Adams, N.A., and Kleiser, L. (1996). Subharmonic transition to turbulence in a flat plate boundary layer at Mach number 4.5. *J. Fluid Mech.*, **317**, 301–335.

Adams, N.A., and Shariff, K. (1996). A high-resolution hybrid compact-ENO scheme for shock–turbulence interaction problems. *J. Comput. Phys.*, **127**, 27–51.

Arora, M., and Roe, P.L. (1997). On postshock oscillations due to shock capturing schemes in unsteady flows. *J. Comput. Phys.*, **130**, 25–40.

Casper, J., and Carpenter, M.H. (1995). Computational considerations for the simulation of shock-induced sound. Technical Report 110222, NASA TM, NASA Langley Research Center, Hampton, Virginia.

Cockburn, B., and Shu, C.-W. (1994). Nonlinearly stable compact schemes for shock calculations. *SIAM J. Numer. Anal.*, **31**, 607–627.

Guo, Y., Adams, N.A., and Kleiser, L. (1996). A comparison study of an improved temporal DNS and spatial DNS of compressible boundary layer transition. *AIAA J.*, **34**, 683–690.

Hou, T.Y., and Le Flock, P.G. (1994). Why nonconservative schemes converge to wrong solutions: error analysis. *Math. Comput.*, **62**, 497–530.

Israeli, M., and Orszag, S.A. (1981). Approximation of radiation boundary conditions. *J. Comput. Phys.*, **41**, 115–135.

Lele, S.K. (1992). Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.*, **103**, 16–42.

Mahesh, K., Moin, P., and Lele, S.K. (1996). The interaction of a shock wave with a turbulent shear flow. Technical Report TF-69, Department of Mechanical Engineering, Stanford University, Stanford, California.

Pruett, C.D., Zang, T., Chang, C.-L., and Carpenter, M.H. (1995). Spatial direct numerical simulation of high-speed boundary-layers flows—part I: algorithmic considerations and validation. *Theoret. Comput. Fluid Dynamics*, **7**, 49–76.

Rai, M.M., and Moin, P. (1993). Direct numerical simulation of transition and turbulence in a spatially evolving boundary layer. *J. Comput. Phys.*, **93**, 169–192.

Settles, G.S., and Dodson, L.J. (1991). Hypersonic shock/boundary-layer interaction database. Technical Report NASA CR 177577, NASA Ames Research Center, Moffet Field, California.

Shu, C.-W., and Osher, S. (1989). Efficient implementation of essentially non-oscillatory shock-capturing schemes. II. *J. Comput. Phys.*, **83**, 32–78.

Simeonides, G., Haase, W., and Manna, M. (1994). Experimental, analytical and computational methods applied to hypersonic compression ramp flows. *AIAA J.*, **32**, 301–310.

Thompson, K.W. (1987). Time dependent boundary conditions for hyperbolic systems. *J. Comput. Phys.*, **68**, 1–24.

Wilcox, D.C. (1990). Supersonic compression-corner applications of a multiscale model for turbulent flows. *AIAA J.*, **28**, 1194–1198.

Zurmühl, R., and Falk, S. (1984). *Matrizen und ihre Anwendungen, Band I: Grundlagen*, 5th edition. Springer-Verlag, Berlin.