
Direct Numerical Simulation of Shear Flow Phenomena on Parallel Vector Computers

Andreas Babucke, Jens Linn, Markus Kloker, and Ulrich Rist

Institute of Aerodynamics and Gasdynamics, University of Stuttgart,
Pfaffenwaldring 21, D-70569 Stuttgart, Germany,
babucke@iag.uni-stuttgart.de

Abstract A new code for direct numerical simulations solving the complete compressible 3-D Navier-Stokes equations is presented. The scheme is based on 6th-order compact finite differences and a spectral ansatz in spanwise direction. A hybrid MPI/shared-memory parallelization is implemented to utilize modern parallel vector computers as provided by HLRS. Domain decomposition and modular boundary conditions allow the application to various problems while keeping a high vectorization for favourable computing performance. The flow chosen for first computations is a mixing layer which may serve as a model flow for the initial part of a jet. The aim of the project is to learn more on the mechanisms of sound generation.

1 Introduction

The parallel vector computers NEC SX-6 and NEC SX-8 recently installed at HLRS led to the development of a new code for spatial direct numerical simulations (DNS) of the unsteady compressible three-dimensional Navier-Stokes equations. DNS simulations require high order schemes in space and time to resolve all relevant scales while keeping an acceptable number of grid-points. The numerical scheme of the code is based on the previous compressible code at the Institut für Aero- und Gasdynamik (IAG) and has been further improved by using fully 6th-order compact finite differences in both streamwise (x) and normal (y) direction. Computing the second derivatives directly leads to a better resolution of the viscous terms. By the means of grid transformation in the x - y plane one can go beyond an equidistant cartesian grid to arbitrary two-dimensional geometries. The parallelization concept of both MPI and shared memory parallelization allows to use parallel vector machines efficiently. Combining domain decomposition and grid transformation enhances the range of applications furthermore. Different boundary conditions can be applied easily due to their modular design.

The verified code is applied to a plane subsonic mixing layer consisting of two streams with unequal velocities. The intention is to model the ini-

tial part of a high Reynolds number jet and to investigate the process of sound generation inside a mixing layer. By understanding its mechanisms, we want to influence the flow in order to reduce the emitted noise. Aeroacoustic computations face the problems of i) the large extent of the acoustic field compared to the flow field and ii) the low amplitudes of the emitted sound relative to the instability waves' pressure fluctuations inside the shear region. Therefore, a high-order accurate numerical scheme and appropriate boundary conditions have to be used to minimize spurious numerical sound.

2 Computational Scheme

2.1 Governing Equations

The DNS code is based on the Navier-Stokes equations for 3-d unsteady compressible flows. In what follows, velocities are normalized by the inflow velocity U_∞ and all other quantities by their inflow values, marked with the subscript ∞ . Length scales are made dimensionless with a reference length L and time t with L/U_∞ . Symbols are defined as follows: x , y and z are the spatial coordinates in streamwise, normal and spanwise direction, respectively. The three velocity components in these directions are described by u , v , w . ρ , T and p represent density, temperature and pressure. The specific heats c_p and c_v are assumed to be constant and therefore their ratio $\kappa = c_p/c_v$ is constant as well. Temperature dependance of viscosity μ is modelled using the Sutherland law:

$$\mu(T) = T^{3/2} \cdot \frac{T_\infty + T_s}{T + T_s} \quad (1)$$

with $T_s = 110.4 \text{ K}$. Thermal conductivity ϑ is obtained by assuming a constant Prandtl number $Pr = c_p \mu / \vartheta$. The most characteristic parameters describing a compressible viscous flow field are the Mach number Ma and the Reynolds number $Re = \rho_\infty U_\infty L / \mu_\infty$.

We use the conservative formulation described in [13] which results in the solution vector $\mathbf{Q} = [\rho, \rho u, \rho v, \rho w, E]$ containing the density, the three momentum densities and the total energy per volume

$$E = \rho \cdot c_v \cdot T + \frac{\rho}{2} \cdot (u^2 + v^2 + w^2) \quad (2)$$

The continuity equation, the three momentum equations and the energy equation can be written in vector notation

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = 0 \quad (3)$$

with the flux vectors \mathbf{F} , \mathbf{G} and \mathbf{H} :

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ u(E + p) + q_x - u\tau_{xx} - v\tau_{xy} - w\tau_{xz} \end{bmatrix} \quad (4)$$

$$\mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho v^2 + p - \tau_{yy} \\ \rho vw - \tau_{yz} \\ v(E + p) + q_y - u\tau_{xy} - v\tau_{yy} - w\tau_{yz} \end{bmatrix} \quad (5)$$

$$\mathbf{H} = \begin{bmatrix} \rho w \\ \rho uw - \tau_{xz} \\ \rho vw - \tau_{yz} \\ \rho w^2 + p - \tau_{zz} \\ w(E + p) + q_z - u\tau_{xz} - v\tau_{yz} - w\tau_{zz} \end{bmatrix} \quad (6)$$

containing normal stresses

$$\tau_{xx} = \frac{\mu}{Re} \left(\frac{4}{3} \frac{\partial u}{\partial x} - \frac{2}{3} \frac{\partial v}{\partial y} - \frac{2}{3} \frac{\partial w}{\partial z} \right) \quad (7)$$

$$\tau_{yy} = \frac{\mu}{Re} \left(\frac{4}{3} \frac{\partial v}{\partial y} - \frac{2}{3} \frac{\partial u}{\partial x} - \frac{2}{3} \frac{\partial w}{\partial z} \right) \quad (8)$$

$$\tau_{zz} = \frac{\mu}{Re} \left(\frac{4}{3} \frac{\partial w}{\partial z} - \frac{2}{3} \frac{\partial u}{\partial x} - \frac{2}{3} \frac{\partial v}{\partial y} \right), \quad (9)$$

shear stresses

$$\tau_{xy} = \frac{\mu}{Re} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad (10)$$

$$\tau_{xz} = \frac{\mu}{Re} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \quad (11)$$

$$\tau_{yz} = \frac{\mu}{Re} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \quad (12)$$

and the heat flux

$$q_x = -\frac{\vartheta}{(\kappa - 1)RePrMa^2} \frac{\partial T}{\partial x} \quad (13)$$

$$q_y = -\frac{\vartheta}{(\kappa - 1)RePrMa^2} \frac{\partial T}{\partial y} \quad (14)$$

$$q_z = -\frac{\vartheta}{(\kappa - 1)RePrMa^2} \frac{\partial T}{\partial z}. \quad (15)$$

Closure of the equation system is provided by the ideal gas law:

$$p = \frac{1}{\kappa Ma^2} \cdot \rho T. \quad (16)$$

2.2 Grid Transformation

To be able to compute complex geometries, a grid transformation in the x - y plane as described by Anderson [1] is applied. This means that the physical x - y plane is mapped onto an equidistant computational ξ - η grid:

$$x = x(\xi, \eta), \quad y = y(\xi, \eta). \tag{17}$$

The occurring x and y derivatives need to be transformed into derivations with respect to ξ and η

$$\frac{\partial}{\partial x} = \frac{1}{J} \left[\left(\frac{\partial}{\partial \xi} \right) \left(\frac{\partial y}{\partial \eta} \right) - \left(\frac{\partial}{\partial \eta} \right) \left(\frac{\partial y}{\partial \xi} \right) \right] \tag{18}$$

$$\frac{\partial}{\partial y} = \frac{1}{J} \left[\left(\frac{\partial}{\partial \eta} \right) \left(\frac{\partial x}{\partial \xi} \right) - \left(\frac{\partial}{\partial \xi} \right) \left(\frac{\partial x}{\partial \eta} \right) \right] \tag{19}$$

$$J = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix} = \frac{\partial x}{\partial \xi} \cdot \frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi} \cdot \frac{\partial x}{\partial \eta} \tag{20}$$

with the metric coefficients $(\partial x/\partial \xi)$, $(\partial y/\partial \xi)$, $(\partial x/\partial \eta)$, $(\partial y/\partial \eta)$ and J being the determinant of the Jacobi matrix. To compute second derivatives resulting from viscous terms in the Navier-Stokes equations, Eqs. (18) and (19) are applied twice taking into account that the metric coefficients and by that also the determinant of the Jacobi matrix can be a function of ξ and η . It is possible to compute the metric coefficients and their derivatives analytically if a specific grid transformation is recognized – if not, they will be computed using 4th-order central finite differences.

2.3 Spatial Discretization

As we use a conservative formulation, convective terms are discretized as one term to better restrain conservation equations. Viscous terms are expanded because computing the second derivative results in double accuracy compared to applying the first derivative twice. The Navier-Stokes equations combined with grid transformation lead to enormous terms, e.g. plotting the energy equation requires more than ten pages. Due to that, code generation had to be done using computer algebra software like Maple [11].

The flow is assumed to be periodic in spanwise direction. Therefore we apply a spectral ansatz to compute the derivatives in z direction

$$f(x, y, z, t) = \sum_{k=-K}^K \hat{F}_k(x, y, t) \cdot e^{i(k\gamma z)} \tag{21}$$

with f being a flow variable, \hat{F}_k its complex Fourier coefficient, K the number of spanwise modes and $i = \sqrt{-1}$. The basic spanwise wavenumber γ is given by the basic wavelength λ_z which is the width of the integration domain.

$$\gamma = \frac{2\pi}{\lambda_z} \tag{22}$$

Spanwise derivatives are computed by transforming the respective variable into Fourier space, multiplying its spectral components with the their wavenumber ($i \cdot k \cdot \gamma$) (or square of their wavenumber for second derivatives) and transforming it back into physical space. Due to products in the Navier-Stokes equations, higher harmonic spectral modes are generated at each timestep. To suppress aliasing, only 2/3 of the maximum number of modes for a specific z -resolution is used [2].

If a two dimensional baseflow is used and disturbances of u, v, ρ, T, p are symmetric and disturbances of w are antisymmetric, flow variables are symmetric/antisymmetric with respect to $z = 0$. Therefore only half the points in spanwise direction are needed and Eq. (21) is transferred to

$$f(x, y, z, t) = F_{0r}(x, y, z, t) + 2 \cdot \sum_{k=1}^K F_{kr}(x, y, t) \cdot \cos(k\gamma z) \quad (23)$$

for $f \in [u, v, \rho, T, p]$

$$f(x, y, z, t) = -2 \cdot \sum_{k=1}^K F_{ki}(x, y, t) \cdot \sin(k\gamma z) \quad (24)$$

for $f \in [w]$.

The spatial derivatives in x - and y -direction are computed using 6th order compact finite differences. Up- and downwind biased differences are applied to the convective terms which have a non-zero imaginary part of the modified wavenumber k_{mod}^* . Its alternating usage leads to carefully designed damping and by that allows the reduction of aliasing errors while keeping the favorable dispersion characteristics of a central scheme [8]. Different schemes can be chosen with respect to the current problem. The real and imaginary parts of the modified wavenumber k_{mod}^* are shown as a function of the nondimensional wavenum-

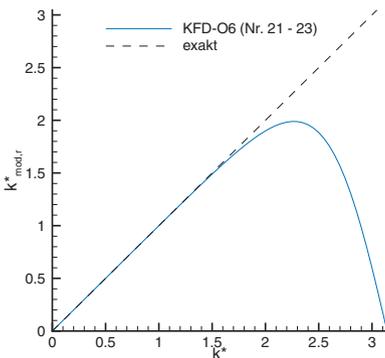


Fig. 1. Real part of the modified wavenumber $k_{mod,r}^*$, equal to k_{mod}^* of central 6th order compact finite difference

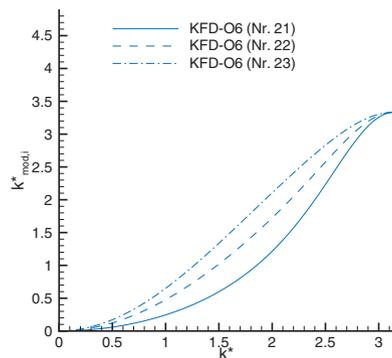


Fig. 2. Imaginary part of the modified wavenumber $k_{mod,i}^*$ for downwind biased finite differences

ber k^* in Fig. 1 and 2 for the implemented schemes. First derivatives resulting from viscous terms, caused by grid transformation and temperature dependence of viscosity, as well as second derivatives are evaluated by standard central compact FD's of 6th order. The resulting tridiagonal equation system is solved using the Thomas algorithm. The algorithm and its solution on multiple domains is discussed detailed in Sect. 2.6.

2.4 Time Integration

The time integration of the Navier-Stokes equations is done using the classical 4th order Runge-Kutta scheme as described in [8]. At each timestep and each intermediate level the biasing of the finite differences for the convective terms is changed. The ability to perform computations not only in total value but also in disturbance formulation is provided by subtracting the spatial operator of the baseflow from the time derivatives of the conservative variables \mathbf{Q} .

2.5 Boundary Conditions

The modular concept for boundary conditions allows the application of the code to a variety of compressible flows. Each boundary condition can either determine the primitive flow variables (u, v, w, ρ, T, p) or provide the time-derivatives of the conservative variables \mathbf{Q} . The spatial regime for time integration is adapted automatically. To keep the code as flexible as possible, boundary-specific parameters, such as the introduction of disturbances, are handled by the boundary conditions themselves. Up to now a variety of boundary conditions is implemented, e.g. isothermal or adiabatic walls containing a disturbance strip if specified, several outflow conditions including different damping zones or a characteristic inflow for subsonic flows having the ability to force the flow with its eigenfunctions obtained from linear stability theory.

2.6 Parallelization

To use the full potential of the new vector computer at HLRS, we have chosen a hybrid parallelization of both MPI [12] and Microtasking. As shared memory parallelization, Microtasking is used along the z direction. The second branch of the parallelization is domain decomposition using MPI. Due to the fact that the Fourier transformation requires data over the whole spanwise direction, a domain decomposition in z direction would have caused high communication costs. Therefore domain decomposition takes place in the ξ - η plane. The arbitrary domain configuration in combination with grid transformation, allows computations for a wide range of problems, e.g. the simulation of a flow over a cavity as sketched in Fig. 3.

The evaluation of the compact finite differences, described in Sect. 2.3 requires to solve a tridiagonal equation system of the form

$$a_k \cdot x_{k-1} + b_k \cdot x_k + c_k \cdot x_{k+1} = f_k \quad (25)$$

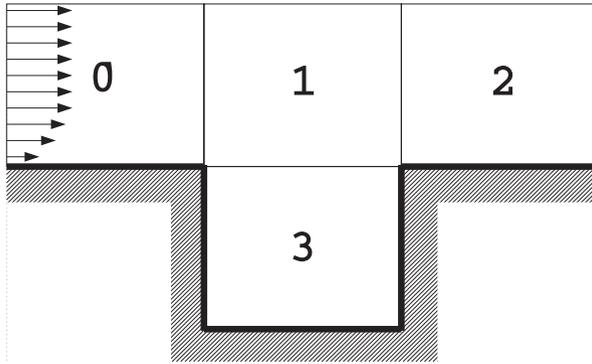


Fig. 3. Exemplary domain configuration for computation of flow over a cavity consisting of four domains. Hatched areas mark noslip wall boundary conditions

for both ξ and η direction with a, b, c being its coefficients. The computation of the RHS f is based on non-blocking MPIISEND/MPIIRECV communication [12]. The standard procedure for the solution of Eq. (25) is the Thomas algorithm consisting of three steps:

1. Forward-loop of LHS:

$$\begin{aligned}
 d_1 &= b_1 \\
 d_k &= b_k - a_k \cdot \frac{c_{k-1}}{d_{k-1}} \quad , \quad (k = 2, \dots, K)
 \end{aligned}
 \tag{26}$$

2. Forward-loop of RHS:

$$\begin{aligned}
 g_1 &= \frac{f_1}{d_1} \\
 g_k &= \frac{-a_k \cdot g_{k-1} + f_k}{d_k} \quad , \quad (k = 2, \dots, K)
 \end{aligned}
 \tag{27}$$

3. Backward-loop of RHS:

$$\begin{aligned}
 x_K &= g_K \\
 x_k &= g_k - x_{k+1} \cdot \frac{c_k}{d_k} \quad , \quad (k = (K - 1), \dots, 1)
 \end{aligned}
 \tag{28}$$

The forward-loop of the LHS requires only the coefficients of the equation system. This has to be done only once at the initialization of the simulation. As Eqs. (27) and (28) contain the RHS f changing at every intermediate Runge-Kutta step, the computation of forward- and backward-loop of the RHS requires a special implementation to achieve acceptable computational performance. The inherent problem regarding parallel implementation is that both loops require values from the previous step, g_{k-1} for the forward-loop and x_{k+1} for the backward-loop of the RHS (note that equation (28) goes from $(K - 1)$ to 1). An ad-hoc implementation would lead to large dead times because each process has to wait

until the previous one has finished. To avoid that, we make use of the fact that we have to compute not only one but up to 25 spatial derivative depending on the spatial direction. The procedure is implemented as follows: the first domain starts with the forward-loop of derivative one. After its completion, the second domain continues the computation of derivative one while the first domain starts to evaluate derivative number two simultaneously. For the following steps, the algorithm continues accordingly. The resulting pipelining is shown exemplary for the forward-loop of the RHS in Fig. 4. If communication time is neglected, the theoretical speedup for forward- and backward-loop of the RHS is expressed by:

$$speedup = \frac{m \cdot n}{m + n - 1} \tag{29}$$

with n being the number of domains in a row or column respectively and m the number of equations to be solved. Theoretical speedup and efficiency of the pipelined Thomas algorithm are shown in Fig. 5 for 25 equations as a function of the number of domains. For 30 domains, efficiency of the algorithm decreases to less than 50 percent. Note that all other computations, e.g. Fourier transformation, Navier-Stokes equations and time integration, are local for each MPI process. Therefore the efficiency of the pipelined Thomas algorithm does not af-

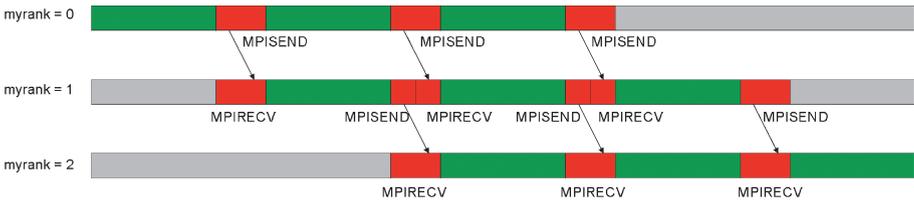


Fig. 4. Illustration of pipelining showing the forward-loop of the RHS for three spatial derivatives on three domains. Green color is denoted to computation, red to communication and grey colour shows dead time

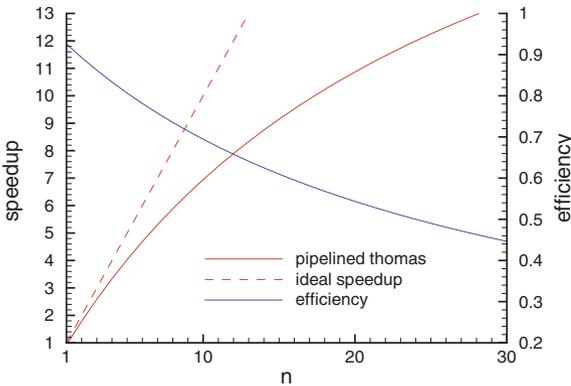


Fig. 5. Theoretical speedup and efficiency of the pipelined Thomas algorithm versus number of domains n for 25 equations

fect the speedup of the entire code that severely. The alternative to the current scheme would be an iterative solution of the equation system. The advantage would be to have no dead times, but quite a number of iterations would be necessary for a converged solution. This results in higher CPU time up to a moderate number of domains. As shared memory parallelization is implemented additionally, the number of domains corresponds to the number of nodes and therefore only a moderate number of domains will be used.

3 Verification of the Code

To verify the code, simulations of a supersonic boundary layer have been performed and in this chapter two cases of this simulations are presented. The results from DNS are compared with *Linear Stability Theory* (LST) and with a previous results from a DNS. In the first case the linear development of a 3-d wave in a boundary layer is compared with the results from LST and in the second case results for a subharmonic resonance case are shown and compared with the work done by Thumm [13].

In both cases the Mach number is $Ma = 1.6$ and freestream temperature is $T_\infty^* = 300$ K. A global Reynolds number of $Re = 10^5$ is chosen, which leads to a reference length scale of $L^* = 2.963$ mm. At the lower boundary (Fig. 6) an adiabatic wall is modeled ($\frac{\partial T}{\partial y} = 0$) and at the upper boundary an exponential decay condition is used (see [13] for further details). The integration domain ends with a buffer domain, in which the disturbances are smoothly ramped to zero. Disturbances are introduced by a disturbance strip at the wall (x_{DS}) into the boundary layer. The grid resolution for both cases is the same as applied by Thumm [13]. A streamwise wave number is dissolved with 16 points, leading to a step size in x -direction of $\Delta x = 0.037$. The step size in y -direction is $\Delta y =$

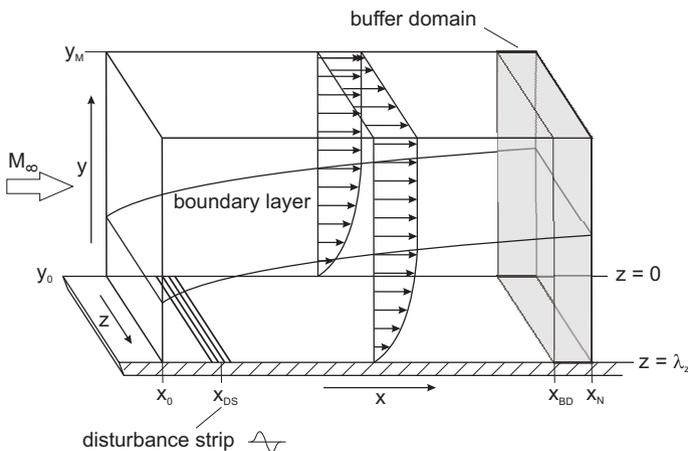


Fig. 6. Computational domain

0.00125 and two Fourier modes ($K_{max} = 2$) are employed in the z -direction. The integration domain starts at $x_0 = 0.225$ and ends at $x_N = 9.64$. The height of the domain includes approximately 2 boundary layer heights δ_{x_N} at the outflow ($y_M = 0.1$).

For a detailed investigation the flow properties are decomposed using a Fourier-decomposition with respect to the frequency and the spanwise wave number

$$\phi'(x, y, z, t) = \sum_{h=-H}^H \sum_{k=-K}^K \hat{\phi}_{(h,k)}(x, y) \cdot e^{i(h\omega_0 t + k\gamma_0 z)} \quad , \quad (30)$$

where ω_0 is the fundamental frequency at the disturbances spectrum, and $\gamma_0 = \frac{2\pi}{\lambda_z}$ the basic spanwise wave number.

3.1 Linear Stage of Transition

In this section, a 3-d wave ($\Psi = \arctan(\gamma/\alpha_r) \simeq 55^\circ \Rightarrow \gamma = 15.2$) with a small amplitude ($A_{(1,1)} = 5 \cdot 10^{-5}$) is generated at the disturbance strip. The development of the disturbance is linear, so the results can be compared with LST. The frequency parameter ($F = \frac{\omega}{Re} = \frac{2\pi f^* L^*}{u_\infty^* Re}$) is chosen to $F_{(1,1)} = 5.0025 \cdot 10^{-5}$. In Fig. 7 the amplification rate α_i for the u' -velocity from DNS and LST are plotted over the x -coordinate. A gap is found between the results, the amplification rates from DNS is higher then those obtained from LST. This gap is also in the simulation of Thumm [13] and Eißler [5], they shove it to non-parallel effects. Maybe this is the reason for the discrepancies in the amplification rates of the u' -velocity.

In Figs. 8–10 the eigenfunctions of u' , v' and p' -disturbance profiles at $x = 4.56$ from DNS and LST are shown. The agreement between DNS and LST is much better for the eigenfunctions of the 3-d wave then for the amplification rates.

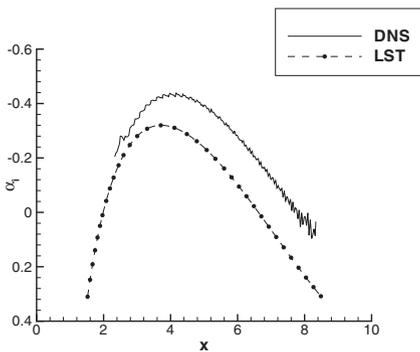


Fig. 7. Downstream development of the amplification rate of u' for the DNS and LST

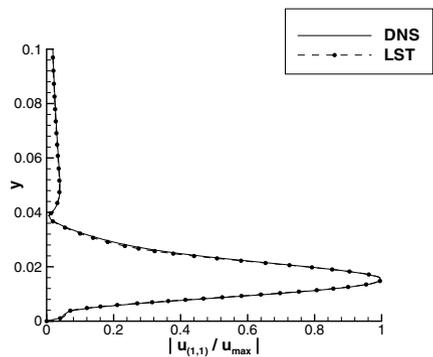


Fig. 8. Comparison of the u' -eigenfunctions of 3-d wave at $x = 4.56$

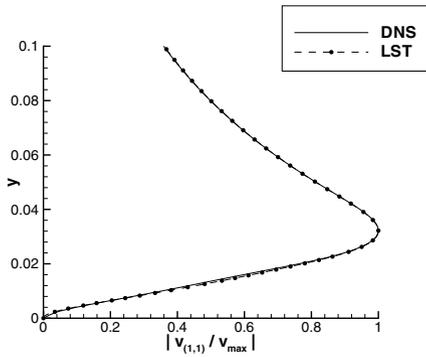


Fig. 9. Comparison of the v' -eigenfunctions of 3-d wave at $x = 4.56$

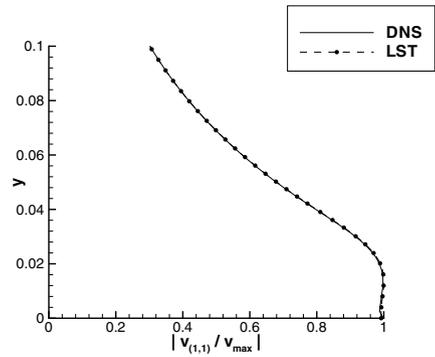


Fig. 10. Comparison of the p' -eigenfunctions of 3-d wave at $x = 4.56$

3.2 Nonlinear Stage of Transition

For the validation of the scheme in the nonlinear stage of transition a subharmonic resonance case from Thumm [13] has been simulated. The two disturbances, a 2-d and a 3-d wave ($\Psi \simeq 45^\circ \Rightarrow \gamma = 5.3$), are now introduced into the integration domain. The frequency parameter for the 2-d wave is $F_{(1,0)} = 5.0025 \cdot 10^{-5}$ and the 3-d wave is $F_{(1/2,1)} = 2.5012 \cdot 10^{-5}$. The amplitudes are $A_{(1,0)} = 0.003$ and $A_{(1/2,1)} = 10^{-5}$. When the amplitude of the 2-d wave reaches 3–4% of the freestream velocity u_∞ , the damped 3-d wave interacts non-linear with the 2-d wave and subharmonic resonance occurs (see Figs. 11–12). This means that the phase speed c_{ph} of the small disturbance adjusts to the phase speed of the high amplitude disturbance. Due to that the 3-d wave grows strongly non-linear.

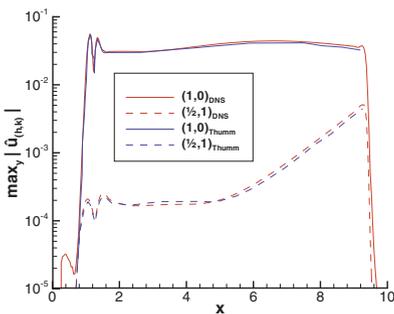


Fig. 11. Amplitude development of the u' -velocity downstream for the subharmonic resonance case

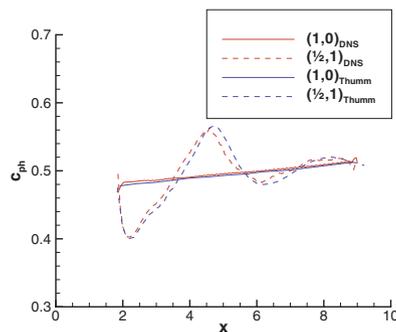


Fig. 12. Phase speed c_{ph} of the v' -velocity for the subharmonic resonance case at $y = 0.0625$

The downstream development of the u' -disturbances obtained from DNS is shown in Fig. 11, the results from Thumm [13] for this case are plotted as well. Thumms results differ only slightly. A reason for the small discrepancies is the different disturbance generation method. Thumm disturbs only v' while in the simulations here, $(\rho v)'$ is disturbed at the wall.

In Fig. 12 the phase speed of the v' -velocity over x at $y = 0.0625$ for the 2-d and 3-d wave is shown for the DNS and the results of Thumm. The phase speed of the 3-d wave approach to the 2-d wave further downstream. Although it is unknown at which y -coordinate, Thumm has determined the phase speed, the results show a good agreement.

4 Simulation of a Subsonic Mixing Layer

The current investigation is part of the DFG-CNRS project “Noise Generation in Turbulent Flows” [15]. Our motivation is to simulate both the compressible mixing layer itself as well as parts of the surrounding acoustic field. The term mixing layer describes a flow field composed of two streams with unequal velocities and serves as a model flow for the initial part of a jet as illustrated by Fig. 13. Even with increasing computational power, one is limited to jets with low Reynolds numbers [6].

4.1 Flow Parameters

The flow configuration is closely matched to the simulation of Colonius, Lele and Moin [4]. The Mach numbers are $Ma_1 = 0.5$ and $Ma_2 = 0.25$, with the subscripts 1 and 2 denoting the inflow values of the upper and the lower stream respectively. As both stream temperatures are equal ($T_1 = T_2 = 280K$), the ratio of the streamwise velocities is $U_2/U_1 = 0.5$. The Reynolds number $Re = \rho_1 U_1 \delta / \mu = 500$ is based on the flow parameters of the upper stream and the vorticity thickness δ at the inflow x_0

$$\delta(x_0) = \left(\frac{\Delta U}{|\partial u / \partial y|_{max}} \right)_{x_0} . \tag{31}$$

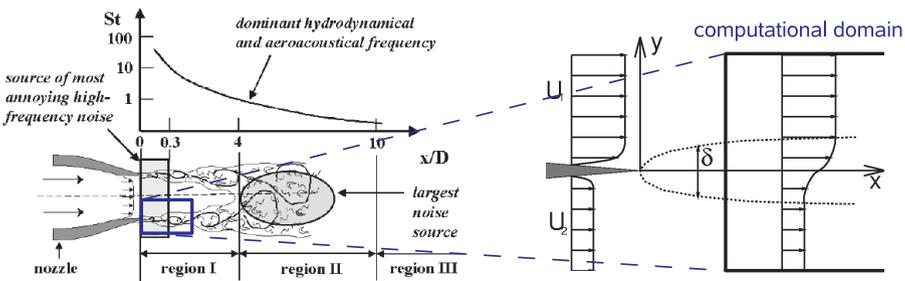


Fig. 13. Location of the computational domain showing the mixing layer as an initial part of a jet

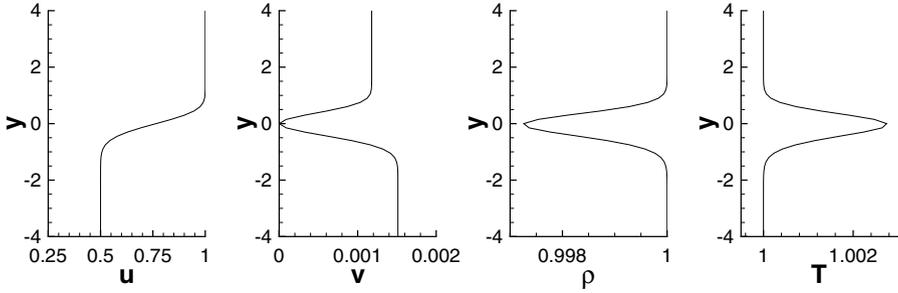


Fig. 14. Initial condition of the primitive variables u , v , ρ and T at the inflow $x_0 = 30$

The initial condition of the mixing layer is provided by solving the steady compressible two-dimensional boundary-layer equations. The initial coordinate $x_0 = 30$ is chosen in a way that the vorticity thickness at the inflow is 1. By that length scales are made dimensionless with δ . The spatial development of the vorticity thickness of the boundary layer solution is shown in Fig. 13. Velocities are normalized by $U_\infty = U_1$ and all other quantities by their values in the upper stream. Figure 14 shows the initial values at $x_0 = 30$.

A cartesian grid of 2300×850 points in x - and y -direction is used. In stream-wise direction the grid is uniform with spacing $\Delta x = 0.157$ up to the sponge region where the grid is highly stretched. In normal direction the grid is continuously stretched with the smallest stepsize $\Delta y = 0.15$ inside the mixing layer ($y = 0$) and the largest spacing $\Delta y = 1.06$ at the upper and lower boundaries. In both directions smooth analytical functions are used to map the physical grid on the computational equidistant grid. The grid and its decomposition into 8 domains is illustrated in Fig. 15.

4.2 Boundary Conditions

Non-reflective boundary conditions as described by Giles [7] are implemented at the inflow and the freestream boundaries. To excite defined disturbances, the flow is forced at the inflow using eigenfunctions from linear stability theory (see Sect. 4.3) in accordance with the characteristic boundary condition. One-dimensional characteristic boundary conditions possess low reflection coefficients for low-amplitude waves as long as they impinge normal to the boundary. To minimize reflections caused by oblique acoustic waves, a damping zone is applied at the upper and lower boundary. It draws the flow variables \mathbf{Q} to a steady state solution \mathbf{Q}_0 by modifying the time derivative obtained from the Navier-Stokes Eqs. (3):

$$\frac{\partial \mathbf{Q}}{\partial t} = \frac{\partial \mathbf{Q}}{\partial t}_{\text{Navier-Stokes}} - \sigma(y) \cdot (\mathbf{Q} - \mathbf{Q}_0) \quad (32)$$

The spatial dependence of the damping term σ allows a smooth change from no damping inside the flow field to maximum damping σ_{max} at the boundaries.

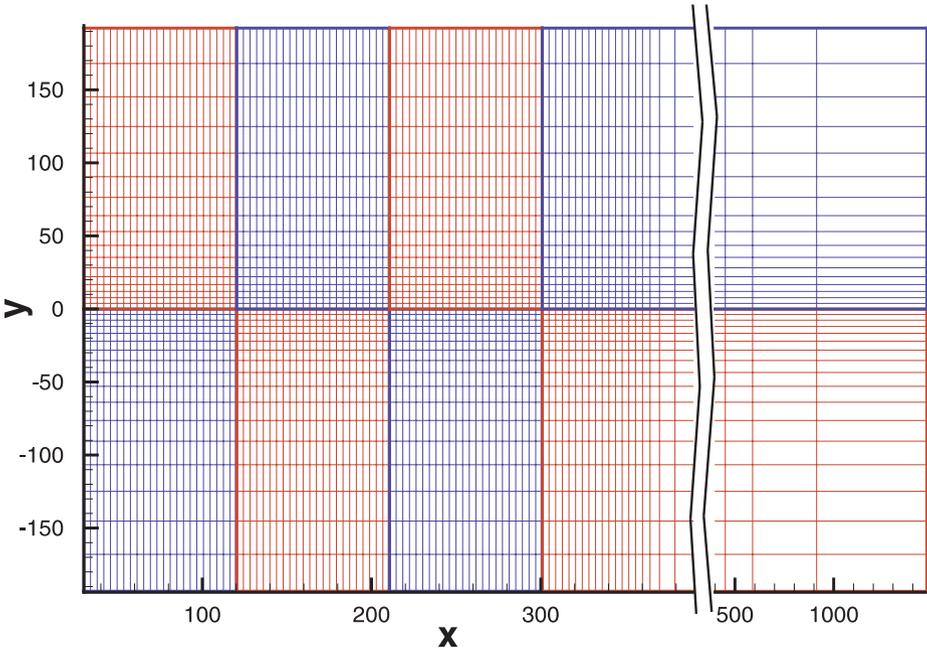


Fig. 15. Grid in physical space showing every 25th gridline. Domain decomposition in 8 subdomains is indicated by red and blue colours

To avoid large structures passing the outflow, a combination of grid stretching and low-pass filtering [9] is used as proposed by Colonius, Lele and Moin [3]. Disturbances become increasingly badly resolved as they propagate through the sponge region and by applying a spatial filter, the perturbations are substantially dissipated before they reach the outflow boundary. The filter is necessary to avoid negative group velocities which occur when the non-dimensional modified wavenumber k_{mod}^* is decreasing (see Fig. 1).

4.3 Linear Stability Theory

Viscous linear stability theory [10] describes the evolution of small amplitude disturbances in a steady baseflow. It is used for forcing of the flow at the inflow boundary. The disturbances have the form

$$\Phi = \hat{\Phi}(y) \cdot e^{i(\alpha x + \gamma z - \omega t)} + c.c. \tag{33}$$

with $\Phi = (u', v', w', \rho', T', p')$ representing the set of disturbances of the primitive variables. The eigenfunctions are computed from the initial condition by combining a matrix-solver and Wielandt iteration. The stability diagram in Fig. 16 shows the amplification rates at several x positions as a function of the frequency ω . Note that negative values of $-\alpha_i$ correspond to amplification while positive values denote damping. Figure 16 shows that the highest amplification

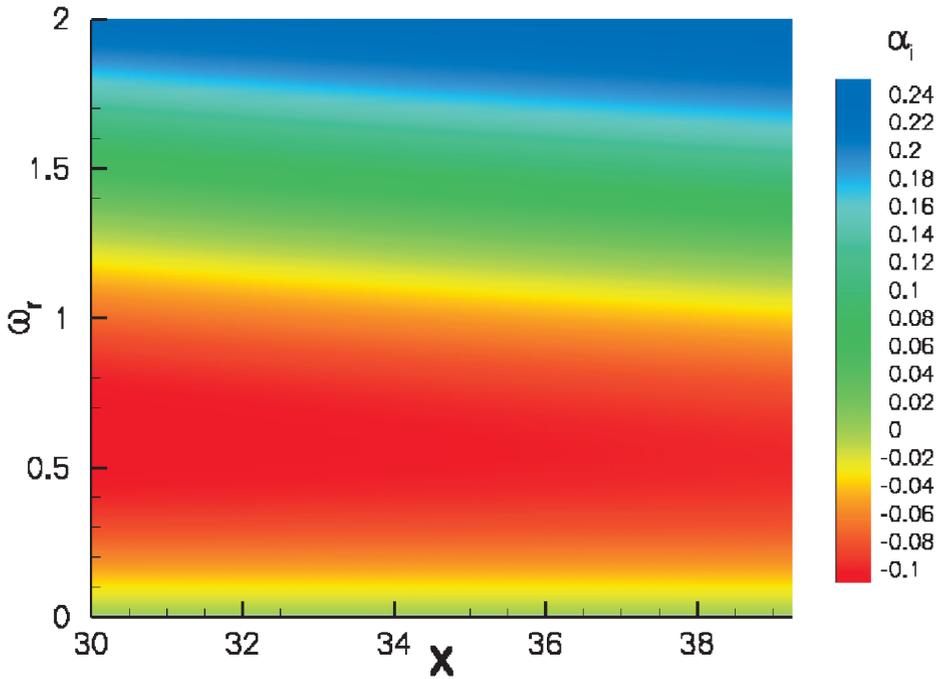


Fig. 16. Stability diagram for 2d disturbances of the mixing layer showing the amplification rate $-\alpha_i$ as a function of frequency ω and x -position

$\alpha_i = -0.1122$ is given for the fundamental frequency $\omega_0 = 0.6296$. Forcing at the inflow is done using the eigenfunctions of the fundamental frequency ω_0 and its subharmonics $\omega_0/2$, $\omega_0/4$ and $\omega_0/8$.

4.4 DNS Results

The high amplification rate as predicted by linear stability theory in the previous Sect. 4.3 leads to a soon roll-up of the mixing layer. Further downstream vortex pairing takes place. Figure 17 illustrates the spatial development of the subsonic mixing layer by showing the spanwise vorticity. In the center of Fig. 18 ($-20 \leq y \leq 20$) the spanwise vorticity is displayed. Above and below, the dilatation $\nabla \mathbf{u}$ gives an impression of the emitted sound. At the right side, the initial part of the sponge zone is included. From the dilatation field, one can determine three major sources of sound:

- in the initial part of the mixing layer ($x = 50$)
- in the area where vortex pairing takes place ($x = 270$)
- at the beginning of the sponge region

The first source corresponds to the fundamental frequency and is the strongest source inside the flow field. Its position is upstream of the saturation of the

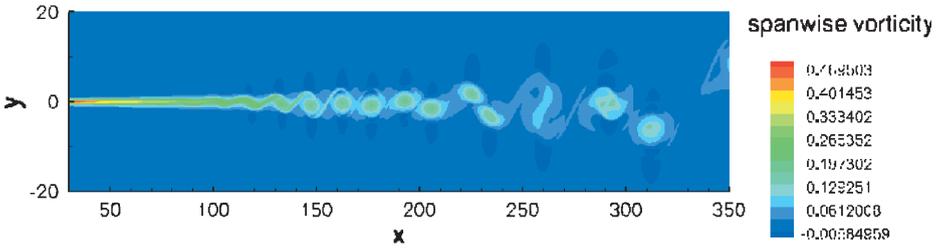


Fig. 17. Instantaneous view of the mixing layer showing roll-up of the vortices and vortex pairing by plotting spanwise vorticity

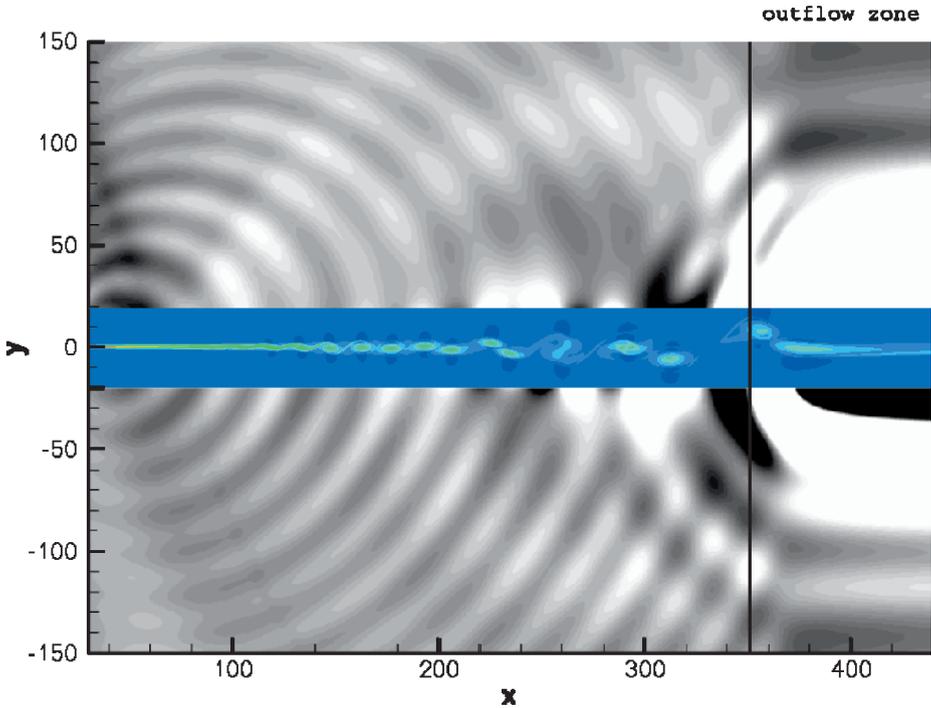


Fig. 18. Instantaneous view of the mixing layer showing spanwise vorticity in the center ($-20 \leq y \leq 20$) and dilatation to visualize the emitted sound. The beginning of the outflow zone consisting of grid-stretching and filtering is indicated by a vertical line

fundamental frequency which corresponds to the results of Colonius, Lele and Moin [4]. The second source is less intensive and therefore can only be seen by shading of the dilatation field. Source number three is directly related to the sponge zone which indicates that dissipation of the vortices occurs too fast. Due to that there is still the necessity to improve the combination of grid-stretching and filtering. As dissipation inside the outflow region is depending on the timestep Δt ,

choosing the appropriate combination of filter- and grid-stretching-parameters is nontrivial.

5 Performance

Good computational performance of a parallel code is first of all based on its single processor performance. As the NEC SX-8 is a vector computer we use its characteristic values for evaluation: the vector operation ratio is 99.75% and the length of the vector pipe is 240 for a 2-d computation on a grid having 575×425 points. Due to the fact that array sizes are already fixed at compilation, optimized memory allocation is possible which reduces the bank conflict to 2% of the total user time. All this results in a computational performance of 9548.6 MFLOP/s which corresponds to 60% of the peak performance of the NEC SX-8 [14]. Computing 30000 timesteps required a user time of 5725 seconds, so one timestep takes roughly $0.78 \mu\text{s}$ per grid-point.

To evaluate the quality of the parallelization, speedup and efficiency are taken into account. Again 30000 timesteps are computed and the grid size of each domain is those mentioned above. Figure 19 shows the dependance of speedup and efficiency on the number of MPI processes. The efficiency decreases to 83% for 8 processes. A somehow strange behaviour is the fact that the efficiency of the single processor run is less than one. Therefore efficiency is based on the maximum performance per processor. The reason for that is the non-exclusive usage of a node for runs with less than 8 processors. So computational performance can be affected by applications of other users. Comparing the achieved efficiency of 89.3% for four processors with the theoretical value of 78.1% according to Eq. (29) shows that even for 2-d computations, solving the tridiagonal equation system is not the major part of computation.

If we extend the simulation to the three-dimensional case, Microtasking, the second branch of the parallelization, is applied. We still use eight domains and by

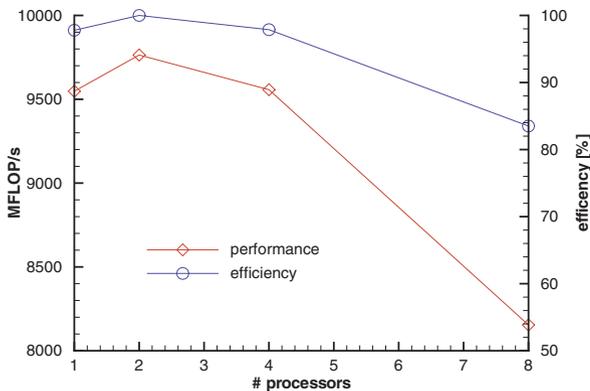


Fig. 19. Computational performance per processor (red) and efficiency (blue) as a function of MPI processes for 2-d computations

that eight MPI-processes with the same grid-size in x - and y -direction but now the spanwise direction is resolved with 33 points corresponding to 22 spanwise modes in the symmetrical case. Each MPI-process runs on its own node having 8 tasks. Computing again 30000 timesteps gives a performance of 380 GFLOP/s and by that an efficiency of 60%. One reason for the decrease in performance is the small number of spanwise modes. Best load-balancing can be achieved for a high number of spanwise modes because the z -resolution in physical space has to be of the form $2^{(kexp+1)}$ with $kexp$ depending on the number of spanwise modes. But the main reason is the poor performance of the FFT routines. Therefore we plan to implement the machine-specific MathKeisan routines. They already showed large improvements in the incompressible code N3D of IAG.

6 Outlook

A new DNS code for the unsteady three-dimensional compressible Navier-Stokes equations has been developed. An improved numerical scheme, based on the previous compressible IAG code, as well as a hybrid parallelization, consisting of MPI and shared memory parallelization, has been implemented. This allows its application to a variety of problems in compressible fluid dynamics while achieving at the same time high computational performance (≈ 9 GFLOP/s per CPU). The main characteristics of the code are the following:

- solution of the full compressible three-dimensional Navier-Stokes equations
- 6th-order accurate compact finite differences in x - and y -direction
- spectral ansatz in spanwise direction (symmetric and non-symmetric)
- direct computation of the second derivatives resulting in better resolved viscous terms
- 4th-order Runge-Kutta time integration
- computation in total value or disturbance formulation
- arbitrary grid transformation in the x - y plane
- hybrid parallelization consisting of MPI and shared memory parallelization
- applicable to a wide range of problems: sub-, trans- and supersonic

To increase the performance for three-dimensional simulations, we plan to implement the FFT routines installed on the NEC SX-8 machine. As communication is not depending on spanwise resolution, we hope that performance in 3-d computations will be as good as in the 2-d case.

The code has been tested and verified for both linear and non-linear disturbances. Comparing the results with reference cases for transitional flows showed excellent agreement. The computation of a subsonic mixing layer is intended to model the initial part of a high Reynolds number jet. By choosing appropriate boundary conditions, it is possible to compute both the flow and the surrounding acoustic field. These simulations will be extended in the future to gain more details on the mechanisms of sound generation with the intention to control jet-induced noise.

Acknowledgements

The authors would like to thank the Deutsche Forschungsgemeinschaft (DFG) for its financial support within the cooperation “Noise Generation in Turbulent Flows” and the HLRS for access to computer resources and support inside the Teraflop project.

References

1. Anderson, J. D.: Computational Fluid Dynamics. McGraw-Hill, 1995
2. Canuto, C., Hussaini, M. Y., Quarteroni, A., Zang, T. A.: Spectral methods in fluid dynamics, Springer Series of Computational Physics (1988), Springer Verlag, Berlin
3. Colonius T., Lele S. K., Moin P.: Boundary Conditions for Direct Computation of Aerodynamic Sound Generation. AIAA-Journal 31, no. 9 (1993), 1574–1582
4. Colonius T., Lele S. K., Moin P.: Sound Generation in a mixing layer. J. Fluid Mech. 330 (1997), 375–409
5. Eißler, W.: Numerische Untersuchung zum laminar-turbulenten Strömungsumschlag in Überschallgrenzschichten. Dissertation, Universität Stuttgart, 1995
6. Freund J. B.: Noise Sources in a low-Reynolds-number turbulent jet at Mach 0.9. J. Fluid Mech. 438 (2001), 277–305
7. Giles M. B.: Non-reflecting boundary conditions for Euler equation calculations. AIAA-Journal 28, no. 12 (1990), 2050–2058
8. Kloker, M. J.: A robust high-resolution split type compact FD scheme for spatial direct numerical simulation of boundary-layer transition. Applied Scientific Research 59 (1998), 353–377
9. Lele, S. K.: Compact Finite Difference Schemes with Spectral-like Resolution. J. Comp. Physics 103 (1992), 16–42
10. Mack L. M.: Boundary-layer linear stability theory. AGARD-Report 709 (1984), 3.1–3.81
11. Kofler, M.: Maple V Release 2. Addison-Wesley, 1994
12. MPI Forum Mpi A message-passing interface standard. Technical Report CS-94-230, University of Tennessee, Knoxville, 1994
13. Thumm, A.: Numerische Untersuchung zum laminar-turbulenten Grenzschichtumschlag in transsonischen Grenzschichtströmungen. Dissertation, Universität Stuttgart, 1991
14. <http://www.hlrs.de/hw-access/platforms/sx8/>
15. <http://www.iag.uni-stuttgart.de/DFG-CNRS/>