Direct Numerical Simulation of a Serrated Nozzle End for Jet-Noise Reduction

A. Babucke, M. Kloker, and U. Rist

Institut für Aerodynamik und Gasdynamik, Universität Stuttgart, Pfaffenwaldring 21 D-70550 Stuttgart, Germany, email: babucke@iag.uni-stuttgart.de

Summary. Sound generation of a subsonic laminar jet has been investigated using direct numerical simulation (DNS). The simulation includes the nozzle end, modelled by a finite flat splitter plate with Mach numbers of $Ma_I = 0.8$ above and $Ma_I = 0.2$ below the plate. Behind the nozzle end, a combination of wake and mixing layer develops. Due to its instability, roll up and pairing of spanwise vortices occur, with the vortex pairing being the major acoustic source. As a first approach for noise reduction, a rectangular notch at the trailing edge is investigated. It generates longitudinal vortices and a spanwise deformation of the flow downstream of the nozzle end. This leads to a an early breakdown of the large spanwise vortices and accumulations of small-scale structures. Compared to a two-dimensional simulation performed earlier [3], the emitted sound is reduced by 6dB.

1 Introduction

Noise reduction is of special interest for many technical problems, as high acoustic loads lead to a reduced quality of life and may cause stress for persons concerned permanently. The current investigation focuses on jet noise as it is a major noise source of aircrafts. As the major airports are typically located in highly populated areas, noise reduction would improve the situation of many people. Direct aeroacoustic simulations are a relatively new field in computational fluid dynamics, facing several difficulties due to largely different scales. The hydrodynamic fluctuations are small-scale structures containing high energy compared to the acoustics with relatively long wavelengths and small amplitudes. Therefore, high resolution is required to compute the noise sources accurately. On the other hand a large computational domain is necessary to obtain the relevant portions of the acoustic far-field. Due to the small amplitudes of the emitted noise, boundary conditions have to be chosen carefully, in order not to spoil the acoustic field with reflections.

Up to now, large-eddy or direct numerical simulations of jet noise have been focusing on either pure mixing layers [2, 5, 7] or low Reynolds number jets [9], where an S-shaped velocity profile is prescribed at the inflow. Our approach is to include the nozzle end, modelled by a thin finite flat plate with two different free-stream velocities above and below. Including the nozzle end shifts the problem to

a more realistic configuration, leading to a combination of wake and mixing layer behind the splitter plate. Additionally, wall-bounded actuators for noise reduction can be tested without the constraint to model them by artifical volume forces. In the current investigation, a passive 'actuator' is considered as a first realistic approach for noise reduction.

2 Numerical Method

2.1 Simulation Framework

Obtaining DNS results does not only require the simulation itself, it also includes pre- and postprocessing as well as stability analysis of the considered baseflow. By far, most computer resources are acquired by the DNS code. Nevertheless the user spends a lot of time in setup of the problem and analysis of the computed data. Therefore a complete framework of tools with a common structure of in- and outputfiles, based on the EAS3 framework [8], has been developed. This does not only lead to a more efficient work for the scientist himself, it also provides reproducibility of the complete set of results. The typical workflow for a DNS is illustrated in figure 1. The first step is the definition of the problem itself. Here, the flow conditions, the grid and the domain decomposition are specified, providing initial data for the DNS. This process strongly depends on the problem to be investigated, resulting in specific tools. A typical approach is to use the solution of the boundary-layer equations and interpolate them on the grid. If the problem allows to obtain a steady-state solution, the DNS code may be used here to obtain a converged solution of the Navier-Stokes equations. The initial condition or a converged solution obtained from the previous step can be used in linear stability theory providing amplification rates and eigenfunctions. This information is used to define disturbance generation for the actual simulation performed by the DNS code. The resulting output is raw binary data with the flow variables given on the computational grid for several time steps. To get a better understanding of the flow physics, postprocessing is used to compute e.g. spectra and amplitudes or vortex criteria.

As the initial condition requires relatively few computations, it is run interactively on the Itanium frontend machine of the NEC-SX8 vector computer. Linear stability theory and the DNS code are executed on the SX8 vector computer. For both codes, the usability has been improved by startscripts running on the frontend machine. These scripts gather all input files, compile the code with the required array sizes, create the jobfile and submit the actual job to the queueing system of the vector computer. Additionally the input files and the source code are archived in the output directory. Thus, the whole computation can be reproduced easily. Postprocessing is done on the frontend machine and consists mainly of a collection of shell-scripts which can be selected via a common input file. These scripts basically trigger the EAS3 command line interface [8]. If an operation needs to be done for multiple files, multiple entities of EAS3 are started, simply by adding an '&' to its call in the script. Due to the common file system of the frontend and the vector machine, no unnecessary copying of data is required.

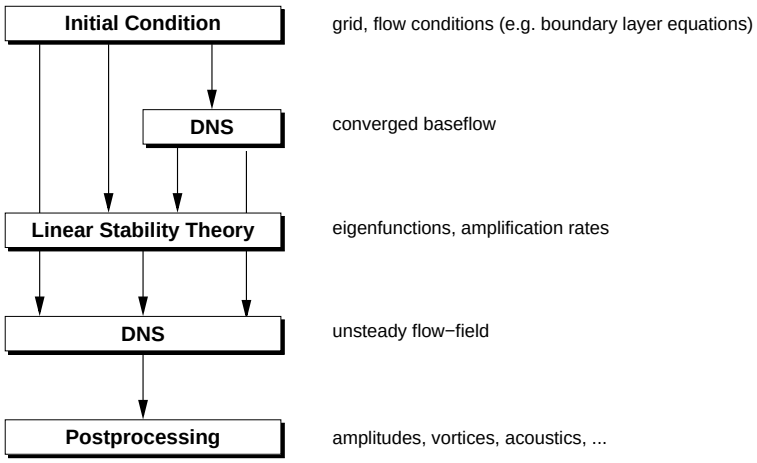

Fig. 1. Workflow for DNS including pre- and postprocessing as well as stability analysis.

2.2 DNS code

The main part of this framework is the DNS code NS3D, solving the full three-dimensional unsteady compressible Navier-Stokes equations on multiple domains. The purpose of domain decomposition is not only to increase computational performance. The combination with grid transformation and the concept of modular boundary conditions allows to compute a wide range of problems. Computation is done in non-dimensional quantities: velocities are normalised by the reference velocity $\overline{U}_\infty$ and all other quantities by their inflow values, marked with the subscript $_\infty$. Length scales are made dimensionless with a reference length $\overline{L}$ and the time t with $\overline{L}/\overline{U}_\infty$, where the overbar denotes dimensional values. Temperature dependence of viscosity μ is modelled using the Sutherland law:

$$\overline{\mu}(T) = \overline{\mu}(\overline{T}_\infty) \cdot T^{3/2} \cdot \frac{1+T_s}{T+T_s} \,, \tag{1}$$

where $T_s = 110.4K/\overline{T}_\infty$ and $\overline{\mu}(\overline{T}_\infty = 280K) = 1.735 \cdot 10^{-5} kg/(ms)$. 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 $Ma = U_\infty/c_\infty$ and the Reynolds number $Re = \rho_\infty U_\infty L/\mu_\infty$.

We use the conservative formulation of the Navier-Stokes equations 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

$$E = \rho \cdot c_v \cdot T + \frac{\rho}{2} \cdot \left(u^2 + v^2 + w^2\right) \,. \tag{2}$$

The complete set of equations is given in [4]. The simulation is carried out in a rectangular domain with x, y, z being the coordinates in streamwise, normal and spanwise direction, respectively. A typical setup for jet noise computation is shown in figure 2.

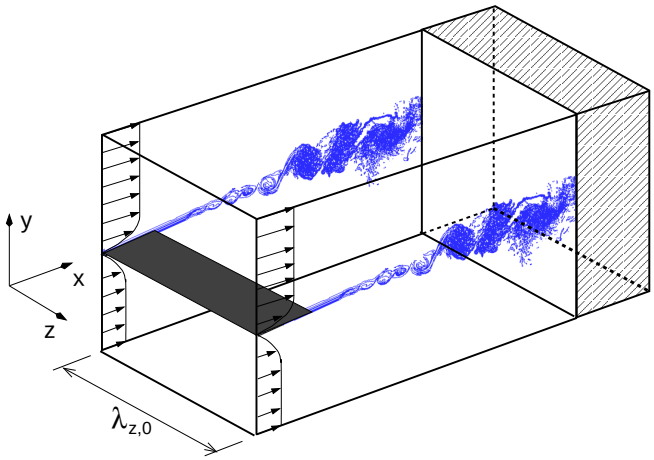

Fig. 2. Integration domain for jet noise computation with splitter plate and sponge zone.

Since the flow is assumed to be periodic in spanwise direction a spectral discretization in z-direction is used:

$$f(x,y,z,t) = \sum_{k=-K}^{K} \hat{F}_k(x,y,t) \cdot e^{i(k\gamma_0 z)} \; . \tag{3}$$

f denotes any flow variable, $\hat{F}_k$ its complex Fourier coefficient, K the number of spanwise modes and $i = \sqrt{-1}$. The fundamental spanwise wavenumber γ_0 is given by the fundamental wavelength $\lambda_{z,0}$ representing the width of the integration domain by $\gamma_0 = 2\pi/\lambda_{z,0}$.

Spanwise derivatives are computed by transforming the respective variable into Fourier space, multiplying its spectral components with their wavenumbers $(i \cdot k \cdot \gamma_0)$ for the first derivatives or square of their wavenumbers for the second derivatives and transforming them back into physical space. Due to the non-linear terms in the Navier-Stokes equations, higher harmonic spectral modes are generated at each time step. To suppress aliasing, only 2/3 of the maximum number of modes for a specific z-resolution are used [6]. 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 number of points in spanwise direction are needed ($0 \leq z \leq \lambda_z/2$) and equation (3) is transferred to

$$f(x,y,z,t) = F_{0r}(x,y,z,t) + 2 \cdot \sum_{k=1}^{K} F_{kr}(x,y,t) \cdot \cos(k\gamma_0 z) \tag{4}$$
$$\text{for } f \in [u,v,\rho,T,p]$$

$$f(x,y,z,t) = -2 \cdot \sum_{k=1}^{K} F_{ki}(x,y,t) \cdot \sin(k\gamma_0 z) \tag{5}$$
$$\text{for } f \in [w] \; .$$

The spatial discretization in streamwise (x) and normal (y) direction is done by 6^{th}-order compact finite differences. The tridiagonal equation systems of the compact finite differences are solved using the Thomas algorithm. To reduce the aliasing error, alternating up- and downwind-biased finite differences are used for convective terms as proposed by Kloker [11]. The second derivatives are evaluated directly which distinctly better resolves the second derivatives compared to applying the first derivative twice. The square of the modified wavenumber k^*_{mod} over k^* is shown for different schemes in figure 3. The exact solution for the second derivative is k^{*2}. While the relative error of the second derivatives is less than 1% up to a modified wavenumber of $k^* = 1.00$ or $k^* = 1.74$ for 4^{th} and 6^{th}order, respectively, the result of computing the first derivative twice is only good up to $k^* = 0.62$ and $k^* = 1.42$ accordingly. Due to that, our scheme provides a resolution advantage of factor 3 for the viscous terms compared to a standard scheme of 4^{th}order, often used for compressible flows. Moreover, another advantage of direct second-derivatives computation is the fact, that k^{*2}_{mod} does not vanish for the least resolved waves with $k^* = \pi$ providing better accuracy and stability of the code.

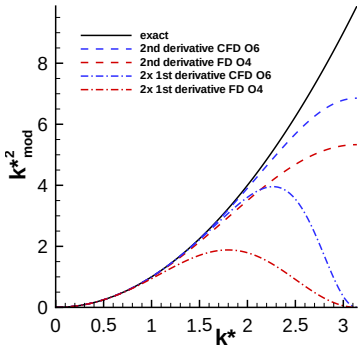

Fig. 3. Comparison of second derivative versus twice the first derivative for a wave with wave number $k^* = k \cdot \Delta x$.

Arbitrary grid transformation in the x-y plane is provided by mapping the physical grid on an equidistant computational ξ-η grid:

$$x = x(\xi, \eta) \ , \ y = y(\xi, \eta) \ . \tag{6}$$

According to [1], the first derivatives can be computed as:

$$\frac{\partial}{\partial x} = \frac{1}{J}\left[\left(\frac{\partial}{\partial \xi}\right)\left(\frac{\partial y}{\partial \eta}\right) - \left(\frac{\partial}{\partial \eta}\right)\left(\frac{\partial y}{\partial \xi}\right)\right] \tag{7}$$

$$\frac{\partial}{\partial y} = \frac{1}{J}\left[\left(\frac{\partial}{\partial \eta}\right)\left(\frac{\partial x}{\partial \xi}\right) - \left(\frac{\partial}{\partial \xi}\right)\left(\frac{\partial x}{\partial \eta}\right)\right] \tag{8}$$

$$J = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix} = \frac{\partial x}{\partial \xi} \cdot \frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi} \cdot \frac{\partial x}{\partial \eta} \tag{9}$$

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 spatial derivatives, equations (7) and (8) are applied twice. Here one has to take into account that the metric coefficients and by that also the Jacobi determinant are a function of ξ and η as well.

Time integration of the Navier-Stokes equations is done using the classical 4^{th}-order Runge-Kutta scheme as described in [11]. At each time step 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 $\mathbf{Q}$.

2.3 Parallelization

To use the full potential of the NEC-SX8 vector computer at HLRS, we have chosen a hybrid parallelization of both MPI and Microtasking. As shared memory parallelization, Microtasking is used along the spanwise 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 would have caused high communication overhead. Therefore domain decomposition is applied only in the ξ-η plane. At each boundary (left, right, top, bottom) of a domain, one can specify the neighbour or one of the implemented boundary conditions. The explicit formulation of the boundary conditions allows to easily implement new boundary conditions or modifications of them. As the domain decomposition must not influence the result, the compact finite differences are used in the whole computational domain. Solving the resulting tridiagonal equation system is the crucial part of the parallelization concept. The Thomas algorithm requires two recursive loops (forward and backward) [4]. Therefore each MPI process has to wait until the previous neighbour has finished its part of the recursive loop. An ad-hoc implementation would result in a serialisation of the Thomas algorithm. To avoid that, we make use of the fact that we have to compute not only one but up to 25 spatial derivatives 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 derivate 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 in figure 4, the backward loop works in the opposite direction, accordingly.

If communication time is neglected, the theoretical speedup for forward- and backward-loop is expressed by:

$$speedup = \frac{m \cdot n}{m + n - 1} \tag{10}$$

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 figure 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

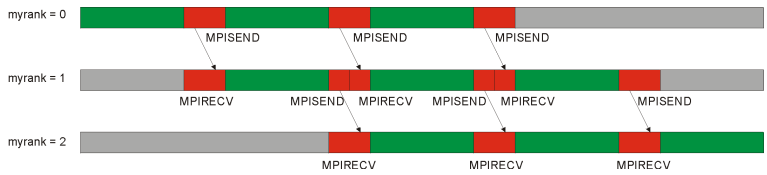

Fig. 4. Illustration of pipelining showing the forward-loop for three spatial derivatives on three domains. Green colour is denoted to computation, red to communication and grey colour shows dead time.

efficiency of the pipelined Thomas algorithm does not affect 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 would result 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.

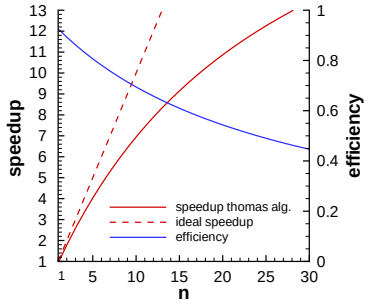

Fig. 5. Theoretical speedup and efficiency of the pipelined Thomas algorithm versus number of domains n for 25 equations.

2.4 Boundary conditions

At the borders of each domain where no neighbour exists, one can select a specific boundary condition. Up to now a variety of boundary conditions have been implemented. According to their properties, the code knows where time integration has to be done and where values are prescribed. This allows to easily implement new boundary conditions or modifications of existing ones, e.g. for disturbance generation.

For the jet-noise investigation, we use a one-dimensional characteristic boundary condition [10] at the freestream. This allows outward-propagating acoustic waves to leave the domain. An additional damping zone forces the flow variables smoothly to a steady state solution, avoiding reflections due to oblique waves. Having a subsonic flow, we also use a characteristic boundary condition at the inflow, allowing upstream propagating acoustic waves to leave the domain. Additionally amplitude and phase distributions from linear stability theory (see 2.6) can be prescribed to introduce defined disturbances. The outflow is the most crucial part as one has to avoid large structures passing the boundary and contaminating the acoustic field. Therefore, a combination of grid stretching and spatial low-pass filtering is applied in the sponge region. Disturbances become increasingly badly resolved as they propagate through the sponge region. As the spatial filter depends on the step size in x-direction, perturbations are smoothly dissipated before they reach the outflow boundary. This procedure shows very low reflections and has been already applied by Colonius et al. [7].

For the splitter plate representing the nozzle end, an isothermal boundary condition is used with the wall temperature being fixed to its value from the initial condition. The pressure is obtained by extrapolation from the interior gridpoints. An extension of the wall boundary condition is the modified trailing edge, where the end of the splitter plate is no more constant along the spanwise direction. As we have grid transformation only in the x-y plane and not in z-direction, the spanwise dependency of the trailing edge is achieved by modifying the connectivity of the affected domains. Instead of regularly prescribing the wall boundary condition along the whole border of the respective subdomain, we can also define a region without wall, now. At these gridpoints, the spatial derivatives in normal direction are recomputed, now using also values from the domain on the other side of the splitter plate. The spanwise derivatives are computed in the same manner as inside the flowfield with the Fourier-transformation being applied along the whole spanwise extent of the domain. The concept of modular boundary conditions, chosen because of flexibility and maintainability, requires explicit boundary conditions and by that a non-compact finite-difference scheme, here. Therefore explicit finite differences have been developed with properties quite similar to the compact scheme used in the rest of the domain. The numerical properties of the chosen 8^{th}-order scheme are compared with standard explicit 6^{th}-order finite differences and the compact scheme of 6^{th} order, regularly used in the flowfield. For the first derivative, the real and imaginary parts of the modified wavenumber k^*_{mod} are shown in figure 6: the increase from order six to eight does not fully reach the good dispersion relation of the 6^{th}-order compact scheme but at least increases the maximum of k^*_{mod} by 10% compared with an ad hoc explicit 6^{th}-order implementation. The imaginary part of the modified wavenumber, responsible for dissipation, shows similar characteristics as the compact scheme with the same maximum as for the rest of the domain. Also for the second derivative, shown by the square of the modified wavenumber k^{*2}_{mod} in figure 7, the increase of its order improves the properties of the explicit finite difference towards the compact scheme.

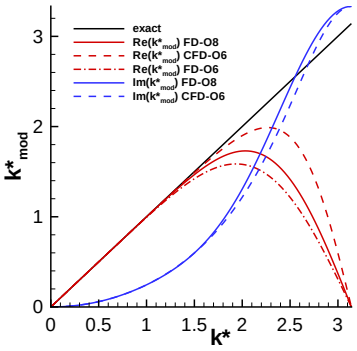

Fig. 6. Real and imaginary part of the modified wavenumber k^*_{mod} for the first derivative based on a wave with wave number $k^* = k \cdot \Delta x$. Comparison of 8^{th}-order explicit finite difference with 6^{th}-order explicit and compact scheme.

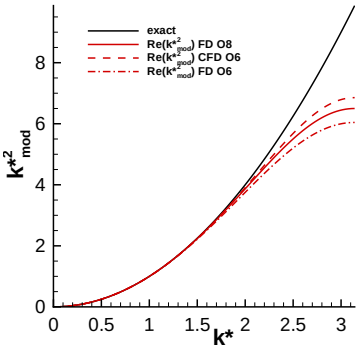

Fig. 7. Square of the modified wavenumber of the second derivative for a wave with wave number $k^* = k \cdot \Delta x$. Comparison of 8^{th}-order explicit finite difference with 6^{th}-order explicit and compact scheme.

2.5 Initial condition

For the current investigation, an isothermal laminar subsonic jet with the Mach numbers $Ma_I = 0.8$ for the upper and $Ma_{II} = 0.2$ for the lower stream has been selected. As both temperatures are equal ($T_1 = T_2 = 280K$), the ratio of the streamwise velocities is $U_I/U_{II} = 4$. This large factor leads to strong instabilities behind the nozzle end, so a moderate number of grid points in x-direction is sufficient to simulate the aeroacoustic source. The Reynolds number $Re = \rho_\infty U_1 \delta_{1,I}/\mu_\infty = 1000$ is based on the displacement thickness $\delta_{1,I}$ of the upper stream at the inflow. With $\delta_{1,I}(x_0) = 1$, length scales are normalized with the displacement thickness of the fast stream at the inflow. The boundary layer of the lower stream corresponds to the same origin of the flat plate.

The cartesian grid is decomposed into sixteen subdomains as illustrated in figure 8: eight in streamwise and two in normal direction. Each subdomain contains 325 x 425 x 65 points in x-, y- and z-direction, resulting in 42 spanwise modes (dealiased) and a total number of 143.6 million gridpoints. The mesh is uniform in streamwise direction with a step size of $\Delta x = 0.15$ up to the sponge region, where the grid is highly stretched. In normal direction, the finest step size is $\Delta y = 0.15$ in the middle of the domain with a continuous stretching up to a spacing of $\Delta y = 1.06$ at the upper and lower boundaries. In spanwise direction, the grid is uniform with a spacing of $\Delta z = 0.2454$ which is equivalent to a spanwise wavenumber $\gamma_0 = 0.2$, where $\lambda_z/2 = \pi/\gamma_0 = 15.708$ is the spanwise extent of the domain. The origin of the coordinate system ($x = 0$, $y = 0$) is located at the end of the nozzle. The nozzle end itself is modeled by a finite thin flat plate with a thickness of Δy. Due to the

vanishing thickness of the nozzle end, an isothermal boundary condition at the wall has been chosen. The temperature of the plate is $\overline{T}_{wall} = 296K$, being the mean value of the adiabatic wall temperatures of the two streams.

The initial condition along the flat plate is obtained from similarity solutions of the boundary-layer equations. Further downstream, the full boundary-layer equations are integrated downstream, providing a flow-field sufficient to serve as an initial condition and for linear stability theory. The resulting streamwise velocity profiles of the initial condition are shown in figures 9 and 10. Behind the nozzle end, the flow field keeps its wake-like shape for a long range. As high amplification rates occur here, the flow is already unsteady before a pure mixing layer has developed. This means that the pure mixing layer investigated earlier [2, 5, 7] has to be considered as a rather theoretical approach.

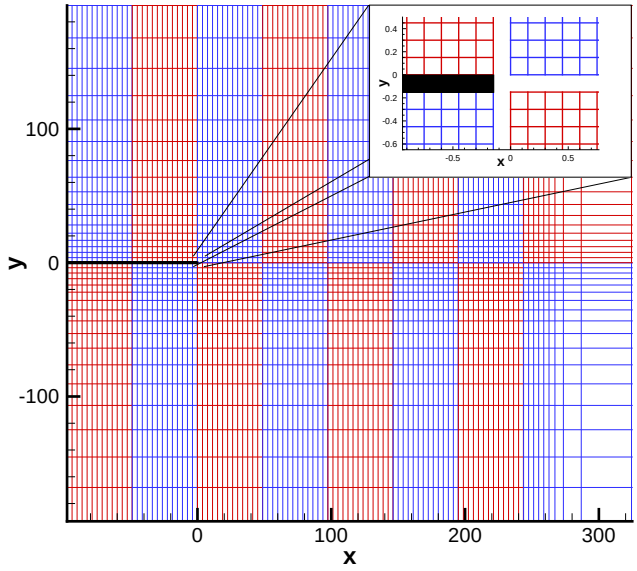

Fig. 8. Grid in x-y plane showing every 25^{th} gridline and illustrating the domain decomposition. Insert: implementation of the splitter plate at the borders of the corresponding domain.

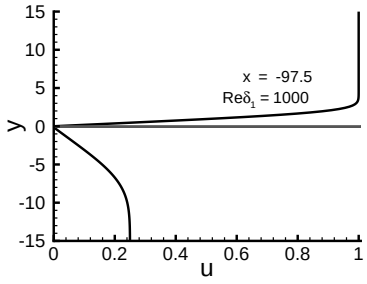

Fig. 9. Profile of the streamwise velocity u for the upper and lower boundary layer at the inflow.

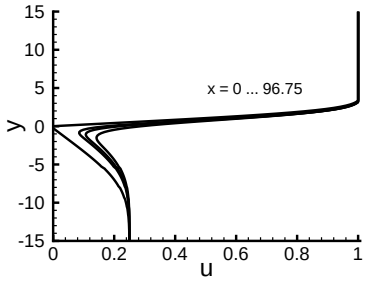

Fig. 10. Downstream evolution of the streamwise velocity profile behind the nozzle end.

2.6 Linear Stability Theory

Spatial linear stability theory (LST) [12] is based on the linearisation of the Navier-Stokes equations, split into a steady two-dimensional baseflow and wavelike disturbances

$$\Phi = \hat{\Phi}_{(y)} \cdot e^{i(\alpha x + \gamma z - \omega t)} + c.c. \tag{11}$$

with $\Phi = (u', v', w', \rho', T', p')$ representing the set of fluctuations of the primitive variables. As only first derivatives in time occur, the temporal problem, where the streamwise wavenumber $\alpha = \alpha_r$ is prescribed, is solved first by a 4^{th}-order matrix solver providing the complex eigenvalues (ω_r, ω_i), with ω_r being the frequency and ω_i the temporal amplification. Once an amplified eigenvalue is found, the Wielandt iteration iterates the temporal to the spatial problem by varying the spatial amplification $-(\alpha_i)$ such that $\omega_i = 0$. This can also be done for a range of streamwise wavenumbers α_r and x positions to obtain a stability diagram. A selected spatial eigenvalue (α_r, α_i) can be fed into the matrix solver to obatin the eigenfunction, being the amplitude and phase distribution of the primitive variables along y. The eigenfunctions can be used directly in the DNS-code for disturbance generation at the inflow.

As the flow is highly unsteady behind the nozzle end and enforcing an artificial steady state does not work properly, we use the initial condition derived from the boundary-layer equations to compute eigenvalues and eigenfunctions. According to figure 11, a fundamental angular frequency of $\omega = 0.0688$ was chosen for the upper boundary layer. The amplification keeps almost constant in downstream direction. As the two boundary layers emerge from the same position, the lower boundary layer is stable up to the nozzle end. Behind the edge of the splitter plate, amplification rates 50 times higher than in the upper boundary layer occur due to the inflection points of the streamwise-velocity profile. Maximum amplification in the mixing layer takes place for a frequency of roughly three to four times of the fundamental frequency of the boundary layer as illustrated in figure 12.

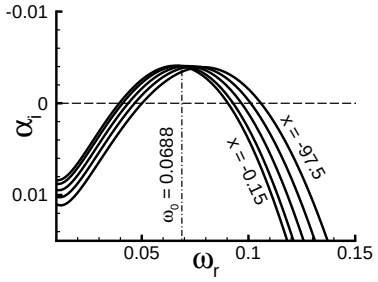

Fig. 11. Amplification rates of the upper (fast) boundary layer given by linear stability theory for various $x-$positions.

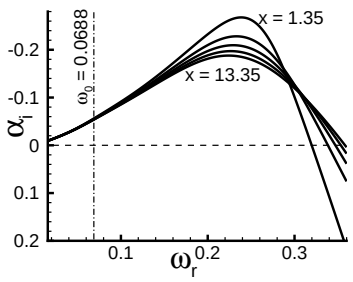

Fig. 12. Amplification rates for various $x-$positions behind the splitter plate predicted by linear stability theory.

3 Numerical Results

For the pure mixing layer without splitter plate [2], we already found that introducing a steady longitudinal vortex leads to a break-up of the big spanwise vortices and may reduce the emitted sound originating from vortex pairing. A variety of wall-mounted actuators are cogitable for the generation of streamwise vortices, our approach is to engrail the trailing edge of the splitter plate. Here, a rectangular spanwise profile of one notch per spanwise wavelength with a depth of 10 in x-direction has been chosen as a first choice. At the inflow of the upper boundary layer, the flow is disturbed with the Tollmien-Schlichting (TS) wave $(1,0)$ with the fundamental frequency and an amplitude of $\hat{u}_{max} = 0.005$, being the same as for the two-dimensional simulation, performed earlier [3]. The TS wave generates higher harmonics in the upper boundary layer, driving the roll-up of spanwise vortices (Kelvin-Helmholtz instability) and the subsequent vortex pairing behind the splitter plate. An additional oblique wave $(1,1)$ with a small amplitude of $\hat{u}_{max} = 0.0005$ is intended to provide a more realistic inflow disturbance than a purely two-dimensional forcing. A total number of 80000 time steps with $\Delta t = 0.018265$ has been computed, corresponding to an non-dimensional elapsed time of $t = 1461$, with the last four periods of the fundamental frequency used for analysis.

The instantaneous flowfield is illustrated in figure 13, showing the λ_2 vortex criterion. Small vortices emerge from the longitudinal edges, slightly deforming the first spanwise vortex of the Kelvin-Helmholtz instability. Further downstream, multiple streamwise vortices exist per $\lambda_{z,0}$, being twisted around the spanwise vortices. This vortex interaction leads to a breakdown of the big spanwise vortices. From $x \approx 120$ onwards, the Kelvin-Helmholtz vortices known from the two-dimensional investigations are now an accumulation of small-scale structures.

A spectral decomposition is shown in figures 14 and 15, based on the maximum of v along y. The normal velocity has been chosen as it is less associated with upstream propagating sound. The modes are denoted as (h,k) with h and k being the multiple of the fundamental frequency ω_0 and the spanwise wavenumber γ_0, respectively.

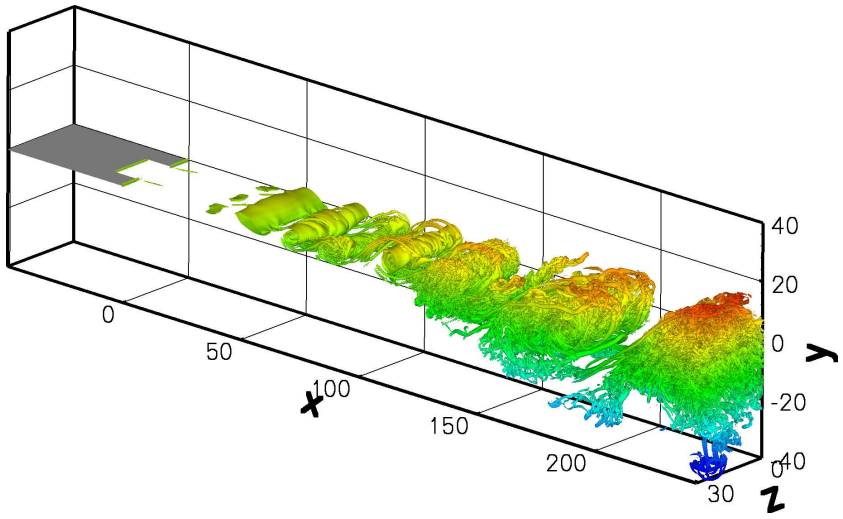

Fig. 13. Perspective view of the engrailed trailing edge and the vortical structures in the instantaneous flow field, visualised by the isosurface $\lambda_2 = -0.005$. The distance from the plane of the splitter plate ($y = 0$) is coloured from blue to red.

As figure 14 shows, the nonlinear interaction of the introduced disturbances $(1, 0)$ and $(1, 1)$ in the upper boundary layer generates nonlinearly the mode $(0, 1)$ up to an amplitude of $\hat{v} = 2 \cdot 10^{-5}$. From $x = -25$ onwards, the upstream effect of the notch at the end of the splitter plate prevails. The engrailment at the end of the splitter plate ($-10 \leq x \leq 0$) generates steady spanwise disturbances $(0, k)$ with peaks up to $\hat{v} = 8 \cdot 10^{-3}$ at the corners. In the notch ($7.8 \leq z \leq 23.6$), the combination of wake and mixing layer originates further upstream at $x = -10$ instead of $x = 0$. This results in a spanwise deformation, corresponding to the disturbance $(0, 1)$. Its amplitude decreases behind the splitter plate up to $x = 15$. Higher harmonics in spanwise direction $(0, 2)$ and $(0, 4)$ are generated at the notch as well, but only mode $(0, 2)$ shows a similar upstream effect as mode $(0, 1)$. Behind the splitter plate, the amplitudes of the first two higher harmonics in spanwise direction stay almost constant at an amplitude of $\hat{v} \approx 6 \cdot 10^{-4}$ and $\hat{v} \approx 4 \cdot 10^{-4}$, respectively. As two streamwise vortices per $\lambda_{z,0}$ emerge from the longitudinal edges, the steady, spanwise higher harmonics mainly correspond to these streamwise vortices. The similar amplitudes behind the splitter plate indicate that the engrailed trailing edge introduces a spanwise deformation due to the different origin of the mixing layer as well as longitudinal vortices. For $x > 40$, all steady modes grow due to non-linear interaction with the travelling waves, resulting in a spanwise deformation of the mixing layer.

The introduced two-dimensional TS wave grows slowly in the upper boundary layer. Figure 16 reveals the good agreement of its amplification rate with linear stability theory. Near the end of the splitter plate, the amplification rate differs from LST due to the discontinuity in geometry. With an amplitude of the driving TS

wave of $\hat{v} \approx 2 \cdot 10^{-3}$, shown in figure 15, the generated higher harmonic modes $(2,0)$, $(3,0)$ reach an amplitude of $\hat{v} \approx 3 \cdot 10^{-4}$ and $\hat{v} \approx 2 \cdot 10^{-5}$, respectively. According to the forcing at the inflow, only low-amplitude oblique disturbances $(2,1)$ and $(3,1)$ are generated in the upper boundary layer. Behind the splitter plate, the growth of two-dimensional disturbances $(h,0)$ is only weakly affected by the engrailed trailing edge. The growth rate of the fundamental frequency shows excellent agreement with linear stability theory. The higher the frequency of the disturbances, the more differs their amplification rate with a slightly lower mean amplification value compared to LST. The initially small three-dimensional disturbances $(h,1)$ grow instantaneously at the beginning of the notch ($x = -10$) by approximately one order of magnitude. Further downstream, they are driven by their two-dimensional counterparts $(h,0)$. Saturation of the first two higher harmonics $(2,0)$ and $(3,0)$ occurs at $x \approx 70$, the position of the first vortex roll up. The two-dimensional fundamental disturbance $(1,0)$ saturates at $x \approx 140$. This corresponds to the pairing of the accumulated small-scale structures.

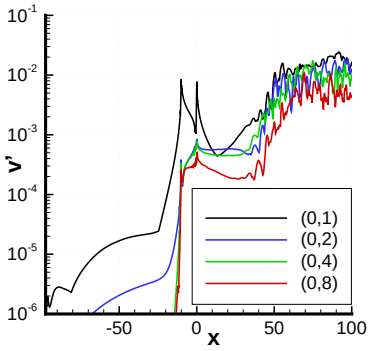

Fig. 14. Generation of the steady modes $(0,k)$ at the trailing edge, based on the maximum of v over y.

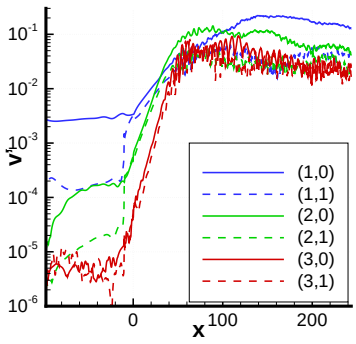

Fig. 15. Maximum amplitude of normal velocity v along y for unsteady modes (h,k).

In order to evaluate the effect of the modified trailing edge, the emitted sound is compared with a two-dimensional simulation with the same flow parameters, performed earlier [3]. The acoustic field, visualised by the dilatation $\nabla \mathbf{u}$, is given for the two cases in figures 18 and 19 for the two-dimensional simulation and the engrailed trailing edge, respectively. In both cases, no reflections from the boundaries are visible. For the two-dimensional simulation, the acoustic field is determined by long-wave sound, originating mainly from $x \approx 150$ and $x \approx 220$. This corresponds to the positions of vortex pairing [3]. The emitted sound for the engrailed trailing edge is mainly high-frequency noise with short wavelengths.

Despite being a two-dimensional simulation, the acoustic field in figure 18 is less clearer than for the pure mixing layer [7]. Nevertheless two main sources can be determined at $x \approx 150$ and $x \approx 220$, corresponding to the positions of vortex pairing

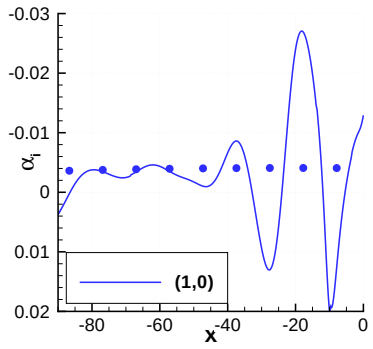

Fig. 16. Amplification rate of the Tollmien-Schlichting wave, compared with linear stability theory (marked with symbols).

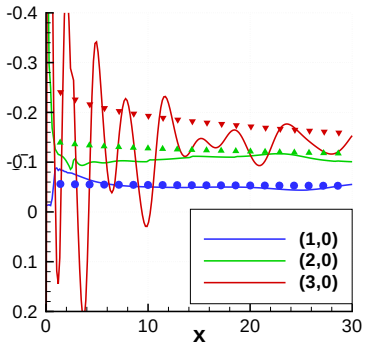

Fig. 17. Amplification rates of two-dimensional disturbances behind the splitter plate, compared with linear stability theory (marked with symbols).

[3]. The emitted sound for the engrailed trailing edge is mainly high-frequency noise with short wavelengths as shown in figure 19. The main sources are located at $x \approx 140$ and $x \approx 200$ which is equivalent to the pairing of the allocations of the small-scale structures. For both, the two-dimensional case and the modified trailing edge, sound generation takes place not directly at the edge of the splitter plate, but further downstream in the mixing layer.

The dilatation plots themselves do not show clearly whether the emitted sound is reduced. By placing an observer in the acoustic far-field ($x = 195, y = -121.8, z = 0$), marked by a cross in figure 19, the sound pressure level can be evaluated more precisely. The time-dependent pressure fluctuations are shown in figure 20 over four periods of the fundamental frequency. For both cases, the pressure fluctuations are almost random. The two-dimensional sample is dominated by low-frequency fluctuations compared to the engrailed-trailing-edge case. The pressure fluctuations of the two- and three-dimensional case are $p'_{2D} = 0.0139$ and $p'_{3D} = 0.00693$, respectively. This means that the engrailed nozzle end leads to a reduction by a factor two, corresponding to a decrease of the noise by -6dB.

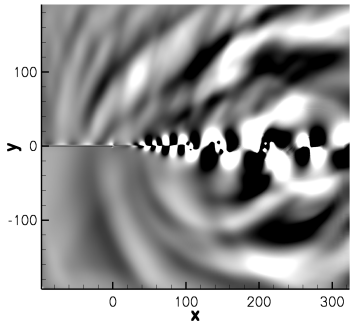

Fig. 18. Snapshot of the far-field sound for the two-dimensional simulation showing the dilatation $\nabla \mathbf{u}$ in a range of $\pm 3 \cdot 10^{-4}$.

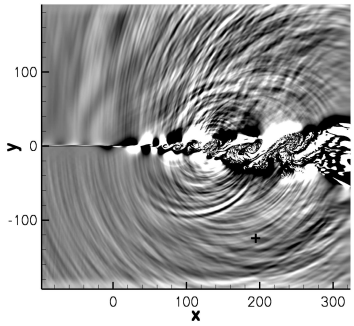

Fig. 19. Snapshot of the dilatation field $\nabla \mathbf{u}$ for the engrailed trailing edge at spanwise position $z = 0$. Contour levels are the same as in figure 18. The position of the acoustic observer is marked by a cross.

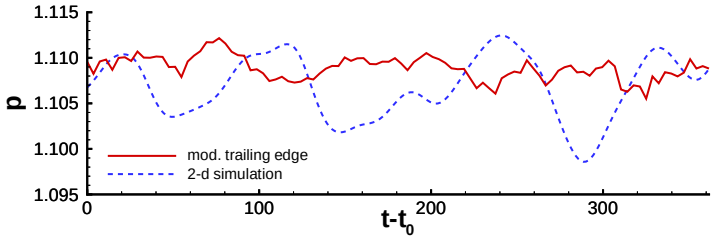

Fig. 20. Acoustic pressure fluctuations in the far-field at the observer's position $(x = 195, y = -121.8)$ for 2-d and 3-d trailing-edge simulation. The plotted time interval corresponds to four periods of the fundamental frequency.

4 Computational Aspects

The simulation was run on the NEC-SX8 Supercomputer of the hww GmbH, Stuttgart, using 16 nodes which corresponds to a total number of 128 processors. On each node, one MPI process was executed, each with shared-memory parallelization having eight tasks. The computation of 80000 time steps required 46 hours

wall-clock time. This leads to a total CPU-time of nearly 6000 hours and a specific computational time of $1.8\mu s$ per gridpoint and time step (including four Runge-Kutta subcycles), being only the double value of the incompressible N3D code of the IAG [13]. With a sustained performance of 694.7 GFLOP/s, 34% of the theoretical peak performance of the computer are reached. The code shows a vector operation ratio of 99.47% with an average vector length of 206 and a total memory size of 162 GB. As the array sizes of each domain are equal, only slight performance differences between the MPI processes exist. Profiling shows that the main computational time (40%) is spent in the Fourier-transformation, followed by the evaluation of the Navier-Stokes equations with 17.4% CPU-time. The computation of the spatial derivatives in streamwise and normal direction requires 10% and 16%, respectively. Time-integration is only a minor part with 1.7% CPU-time.

Having 21 derivatives in x- and 25 derivatives in y-direction, the theoretical speedup of the pipelined Thomas algorithm is, according to equation (10), 6 for the streamwise and 1.92 for the normal direction. This means that the efficiency of the corresponding parts of the code is 75% and 96%, respectively. Taking into account that the major parts of the code (FFT, Navier-Stokes equations) are local for each MPI process, the overall efficiency regarding MPI parallelization is around 96%. As mentioned in section 2.3, this does not include communication times. Nevertheless, profiling shows that the time spent for data transfer is negligible. A limiting factor for the shared-memory parallelization is the spanwise resolution. For a symmetric computation, the number of gridpoints in z-direction is $(2^n + 1)$, with the integer n depending on the number of spanwise modes (here $n = 6$). By that, the spanwise resolution is not a multiple of 8 (the number of shared-memory tasks). The spanwise resolution of 65 points used here means that seven processors pass a Microtasking-loop eight times and 1 processor nine times. By that the efficiency of a shared-memory parallelized loop decreases to 90%. This value corresponds well to the profiling results, showing that 10.5% of the CPU-time is spent in the barriers, framing a Microtasking-loop (e.g. subroutines ex_lpminit, ex_lpmterm). This problem does not exist for non-symmetric simulations, the spanwise resolution is then an exact multiple of eight. Including the losses of both types of parallelization, an overall efficiency of 86% on 128 processors was reached.

To achieve further gains in performance, we intend to optimise the code further, with the main focus on the Fourier-transformation as it requires the most of the computational time. By using the FFT from the SX8 internal library instead of our own subroutines [8], we hope to achieve improvements, since this increased the speed of the incompressible N3D code by 20-30% [13].

5 Conclusion

The sound generation of an isothermal subsonic jet with Mach numbers $Ma_I = 0.8$ and $Ma_{II} = 0.2$ has been simulated using spatial DNS. The nozzle end is modelled by a thin finite flat plate with spanwise engrailment at its trailing edge. This modification of the nozzle end serves as a first example of an actuator for noise reduction, generating streamwise vortices and a spanwise deformation of the flow. Further downstream, the induced longitudinal vortices are bended around the spanwise vortices of the Kelvin-Helmholtz instability, leading to a breakdown of the large coherent structures. By that, the spanwise vortices, known from two-dimensional

simulations are now an accumulation of small scale structures. The emitted sound is compared to a two-dimensional simulation with the same flow parameters. The engrailed trailing edge leads to higher-frequency noise, while the generated sound of the two-dimensional simulation is dominated by low-frequency noise. Despite the parameters of the notch were chosen arbitrarily, a noise reduction of 6dB could be achieved. Therefore, we are confident that further improvements in jet-noise reduction are possible. Besides finding the optimal parameters for the engrailment (shape and dimensions), we also intend to test different types of active and passive actuators.

The investigation was performed using our complete simulation framework of preprocessing, stability analysis, DNS and postprocessing. Due to the combination of frontend machine and supercomputer, a comfortable combination of our tools is possible. This does not only increase the usability, it also provides reproducibility of the complete set of results. The DNS code NS3D uses a hybrid parallelization of MPI and Microtasking. The achieved computational performance is 694.7 GFLOP/s on 16 nodes of the NEC-SX8 vector computer, corresponding to 34% of its theoretical peak performance. The parallel efficiency of the code was investigated by a combination of theoretical analysis and profiling. For this simulation, a scaling of 86% on the 128 processors was reached.

Acknowledgements

The authors would like to thank the Deutsche Forschungsgemeinschaft (DFG) for its financial support within the subproject SP5 in the DFG/CNRS research group FOR-508 *"Noise Generation in Turbulent Flows"*. The provision of supercomputing time and technical support by the Höchstleistungsrechenzentrum Stuttgart (HLRS) within the projects *"Lamtur"* and *"Teraflop Workbench"* is gratefully acknowledged.

References

1. J. D. Anderson. *Computational Fluid Dynamics*. McGraw-Hill, 1995.
2. A. Babucke, M. J. Kloker, and U. Rist. DNS of a plane mixing layer for the investigation of sound generation mechanisms. *to appear in Computers and Fluids*, 2007.
3. A. Babucke, M. J. Kloker, and U. Rist. Numerical investigation of flow-induced noise generation at the nozzle end of jet engines. In *to appear in: New Results in Numerical and Experimental Fluid Mechanics VI*, Contributions to the 15. STAB/DGLR Symposium Darmstadt, 2007.
4. A. Babucke, J. Linn, M. Kloker, and U. Rist. Direct numerical simulation of shear flow phenomena on parallel vector computers. In *High performance computing on vector systems: Proceedings of the High Performance Computing Center Stuttgart 2005*, pages 229–247. Springer Verlag Berlin, 2006.
5. C. Bogey, C. Bailly, and D. Juve. Numerical simulation of sound generated by vortex pairing in a mixing layer. *AIAA J.*, 38(12):2210–2218, 2000.
6. C. Canuto, M. Y. Hussaini, and A. Quarteroni. *Spectral methods in fluid dynamics*. Springer Series of Computational Physics. SpringerVerlag Berlin, 1988.

7. T. Colonius, S. K. Lele, and P. Moin. Sound generation in a mixing layer. *J. Fluid Mech.*, 330:375–409, 1997.
8. EAS3 project. http://sourceforge.net/projects/eas3.
9. J. B. Freund. Noise sources in a low-Reynolds-number turbulent jet at Mach 0.9. *J. Fluid Mech.*, 438:277–305, 2001.
10. M. B. Giles. Nonreflecting boundary conditions for Euler equation calculations. *AIAA J.*, 28(12):2050–2058, 1990.
11. M. J. Kloker. A robust high-resolution split-type compact FD scheme for spatial DNS of boundary-layer transition. *Appl. Sci. Res.*, 59:353–377, 1998.
12. L. Mack. Boundary-layer linear stability theory. In *AGARD Spec. Course on Stability and Transition of Laminar Flow*, volume R-709, 1984.
13. R. Messing, U. Rist, and F. Svenson. Control of turbulent boundary-layer flow using slot actuators. In *High performance computing on vector systems: Proceedings of the High Performance Computing Center Stuttgart 2006*. Springer Verlag Berlin, 2007.