A Finite-Element based Navier-Stokes Solver for LES

W. Wienken \textsuperscript{a}, J. Stiller \textsuperscript{b} and U. Fladrich \textsuperscript{c}.

\textsuperscript{a}Technische Universität Dresden, Institute of Fluid Mechanics (ISM)

\textsuperscript{b}Technische Universität Dresden, Institute for Aerospace Engineering (ILR)

\textsuperscript{c}Technische Universität Dresden, Center for High Performance Computing (ZHR),
D-01062 Dresden, Germany

A new Navier-Stokes solver for laminar and turbulent flow is presented. Special focus is laid on large-eddy simulation of turbulent flows in complex geometries. For discretisation, a streamline-upwind/Petrov-Galerkin (SUPG) finite element method is employed on an unstructured grid of tetrahedral cells. Temporal integration is carried out with an explicit Runge-Kutta scheme. To reduce computational time, parallelisation based on grid partitioning is used.

The new solver is validated for various laminar and turbulent flows, including turbulent channel flow and the flow around a square cylinder. The computations agree well with results from experiments and direct numerical simulations. Furthermore, due to extensive optimisation, the solver exhibits excellent scalability even on a large number of processors.

\textbf{Keywords:} large-eddy simulation, finite element method, parallelisation

\section{1. Introduction}

The importance of turbulent flows led to a continuous effort on more accurate turbulence models and cost-effective numerical algorithms. Additionally, the steady increase in power and the decreasing price of computers allowed to consider these rather expensive models for engineering purposes. An attractive approach is the large-eddy simulation (LES) which computes large scales of turbulence directly while modelling unresolved small structures.

Using LES for flows of engineering interest demands the development of numerical algorithms, which are able to deal with complex geometries without losing too much efficiency, compared to specialised methods. A promising approach is the application of unstructured meshes. The higher numerical complexity is justified perspectivly by the potential of adaptive grid refinement techniques. Furthermore, dynamic grid partitioning is regarded as an appropriate technique for speeding up calculations.

In the present work, an enhanced SUPG finite element method is adopted for computing compressible turbulent flows within the LES framework.
2. Basic Equations

The LES approach makes use of the finding that turbulent flows are dominated by large structures, which are long-living, high-energetic, non-isotropic and strongly depend on the geometry, initial and boundary conditions. In contrast, small scales are short-lived, low-energetic, universal, isotropic and in statistical average dissipative, and therefore, easier to model. For LES the large are separated from the small scales by an filtering operation (see e.g. Martin et al. [1]). The filtered Navier-Stokes equations read

\[ \partial_t \tilde{U} + \partial_x F_i = \partial_i D_i \]

where \( \tilde{U} \) are the resolved conservative variables, and

\[ \tilde{F}_i = \tilde{F}_i + F_{i \text{sgs}}, \quad \tilde{D}_i = \tilde{D}_i + D_{i \text{sgs}} \]

respectively represent the filtered advective and diffusive fluxes, which are both splitted in a resolved and a subgrid-scale (SGS) contribution. In general, the resolved part of any function \( f(U) \) is defined as \( \bar{f} = f(\tilde{U}) \). In particular, the resolved variables and fluxes can be written as

\[
\tilde{U} = \begin{bmatrix}
\tilde{\rho} \\
\tilde{\rho} \tilde{u}_1 \\
\tilde{\rho} \tilde{u}_2 \\
\tilde{\rho} \tilde{u}_3 \\
\tilde{\rho} \tilde{E}
\end{bmatrix}, \quad \tilde{F}_i = \tilde{u}_i \tilde{U} + \tilde{p} \begin{bmatrix}
\delta_{i1} \\
\delta_{i2} \\
\delta_{i3} \\
0
\end{bmatrix}, \quad \tilde{D}_i = \begin{bmatrix}
0 \\
\tilde{\tau}_{i1} \\
\tilde{\tau}_{i2} \\
\tilde{\tau}_{i3}
\end{bmatrix}
\]

\[
\tilde{E} = c_v \tilde{T} + \frac{1}{2} \tilde{u}^2, \quad \tilde{\tau}_{ij} = 2\eta (\tilde{\epsilon}_{ij} - \frac{1}{3} \delta_{ij} \tilde{\epsilon}_{kk}), \quad \tilde{\epsilon}_{ij} = \frac{1}{2}(\partial_j \tilde{u}_i + \partial_i \tilde{u}_j), \quad \tilde{q}_i = -\lambda \partial_i \tilde{T}
\]

where \( \tilde{\rho} \) is the density, \( \tilde{u} \) the velocity, \( \tilde{p} \) the pressure, \( \tilde{T} \) the temperature, \( \tilde{E} \) the specific total energy, \( \tilde{\tau} \) the viscous stress tensor, and \( \tilde{q} \) the heat flux. The SGS contribution to the advective fluxes can be summarized as

\[
F_{i \text{sgs}} \approx \begin{bmatrix}
0 \\
\tau_{i1 \text{sgs}} \\
\tau_{i2 \text{sgs}} \\
\tau_{i3 \text{sgs}} \\
\tilde{u}_j \tau_{ij \text{sgs}} + \tilde{q}_i \text{sgs}
\end{bmatrix}^T
\]

with

\[
\tau_{ij \text{sgs}} = \rho u_i u_j - \tilde{\rho} \tilde{u}_i \tilde{u}_j, \quad \tilde{q}_i \text{sgs} = \rho \tilde{u}_i \tilde{h} - \tilde{\rho} \tilde{u}_i \tilde{h}
\]

The SGS stresses \( \tau_{ij \text{sgs}} \) are computed using the Smagorinsky model with van Driest damping [3,4], while the Reynolds analogy is adopted for the SGS heat flux \( q_i \text{sgs} \). The subgrid contributions emerging from diffusive fluxes can be neglected according to [2,1].

3. Numerics

For spatial discretisation a SUPG finite element method [5–7] with linear shape functions is applied. The resulting integral formulation reads

\[
\int_{\Omega} \left[ W \cdot (\partial_t \tilde{U} - \partial_i W \cdot (\tilde{F}_i - \tilde{D}_i)) \right] dV = \int_{\Gamma} W \cdot (\tilde{F}_i - \tilde{D}_i) n_i dA
\]

\[
+ \sum_{e=1}^{N_e} \int_{\Omega^e} W \cdot A \cdot T \cdot (\partial_t \tilde{U} + \partial_i \tilde{F}_i - \partial_i \tilde{D}_i) dV = 0.
\]
The first two integrals constitute the usual Galerkin formulation while the last expression represents the SUPG operator. \( W \) is a piecewise linear weight function and \( T \) the stabilisation matrix which depends on the local element size \( \Delta x \) and the squares of the advective flux Jacobians \( A_i \) [5]. The spatial discretisation results in a time-dependent system of ordinary differential equations that is integrated using an explicit 4-stage Runge-Kutta scheme along with a damped Jacobi iteration for resolving the consistent mass matrix.

We remark, that the finite element formulation accommodates three different methods: Dropping the stabilisation results in the second order accurate but unstable Galerkin FEM. Omitting the time derivative and diffusive fluxes in the SUPG operator gives the first order streamline diffusion (SD) method. Otherwise we get the full SUPG FEM representing a second order upwind scheme. Here only the latter is used. A detailed comparison of these methods is subject of a forthcoming paper.

4. Implementation

The numerical model was implemented on top of the grid library \( MG \) (Multilevel Grids) [8]. \( MG \) provides a light-weight interface for parallel adaptive finite element solvers on tetrahedral grids. The kernel of \( MG \) has been shown to scale up to several hundred processors. Interprocessor communication is based on MPI. The grid adaptation starts with a single coarse grid and results in a dynamically distributed multilevel grid. Optionally, only the relevant grid portions are stored on each grid level. For illustration, Fig. 1 depicts the local grid refinement and partitioning of a simple configuration.

The current Navier-Stokes solver (MG-NS) can be run in adaptive mode but does only use the finest grid level. Besides LES various statistical models are available for simulation of turbulent flows. Additionally, interfaces for other transport equations are set up.

![Figure 1. Adapted and distributed multilevel grid.](image-url)
5. Results

5.1. Parallel Performance Analysis

Throughout code development, extensive performance analysis and optimisation were part of the implementation efforts. The analysis tool VAMPIR [11] was intensively used for this purpose. Fig. 2 shows the VAMPIR visualisation of a run on 16 CPUs of an SGI ORIGIN 3800 system.

The colours identify different stages of the program execution, e.g. red indicates calls to communication subroutines and other colours represent different states of calculation. Messages sent between processors are shown as black lines. In addition to the global view, Fig. 3 shows the behaviour of the program on one of the 16 processors. The different stages of a time-step (e.g. the Runge-Kutta sub-steps) can clearly be identified.

As both pictures suggest, the communication constitutes a relatively small amount of the execution time compared to the actual calculations. This proposition is supported by the determination of the parallel efficiency $E_n = \frac{T_1}{nT_n}$ ($T_n$ denotes the execution time on $n$ processors). The tests were conducted at the TU DRESDEN on a SGI ORIGIN 3800 system with 128 processors, 120 of which can be employed for user applications. The turbulent channel flow with a constant global mesh size served as a test problem. Figure 4(a) depicts the trend of the efficiency as dependent on the number of processors used. The measured efficiency is above 96% for all runs. The numbers higher than 1.0 (tests with 16 to 64 processors) are caused by cache effects as the local problem size becomes smaller. In consistence with the almost constant parallel efficiency, the speed-up $S_n = \frac{T_1}{T_n}$ is almost...
Figure 3. Time step analysis.

Figure 4. Scalability tests on the SGI ORIGIN 3800.
linear. This can be examined in Fig 4(b).

Earlier tests on a larger number of processors (up to 512) with an application different to MG-NS but also based on MG assure us that the efficiency is not expected to decrease significantly shortly beyond 120 processors.

5.2. Transsonic Flow past a Sphere

The supersonic flow past a sphere was chosen to test the implemented dynamic grid adaptation and the stabilisation. Figure 5 shows a comparison between a schlieren shadograph by A.C. Charters (from VAN DYKE [9]) at a computation using an adaptive grid consisting of ca. $6 \times 10^5$ nodes. Though the Reynolds number was considerably smaller in the computation ($Re_d = 2000$), the excellent qualitative agreement is evident.

![Flow past a sphere at a Mach number of 1.53](image)

Figure 5. Flow past a sphere at a Mach number of 1.53

5.3. Turbulent Channel Flow

The validation of the code for LES included turbulent flows such as homogeneous isotropic turbulence, turbulent channel flow and the flow around a square cylinder. Here, we report only some results obtained for the turbulent flow. Additional results are presented in a forthcoming paper.

Figure 6(a) depicts the configuration used for the LES. The flow is driven by a constant average pressure gradient in $x$ direction. For the homogeneous $x$ and $z$ directions periodic boundary conditions are assumed. The chosen Reynolds number ($Re_\tau = u_\tau \delta / \nu = 180$, where $u_\tau$ is the wall shear velocity) corresponds to the direct numerical simulations carried out by KIM ET AL. [10]. For the LES a regular tetrahedral grid consisting of ca. $3.65 \times 10^5$ nodes was used. The computed average velocity (Fig. 6(b)) is in good qualitative agreement with the DNS data. However, the maximum velocity is overestimated which
can be attributed to the fact that the grid was too coarse for resolving the wall layer accurately. The velocity fluctuations (Fig. 6(c)) agree well with the DNS. Finally, the resolved Reynolds stress is compared to asymptotic theory for high Reynolds numbers in Fig. 6(d).

Figure 6. Set-up and results for turbulent channel flow

6. Conclusion and Outlook

The newly developed solver proved to be a valuable tool for computing laminar and turbulent flows in complex geometries. The qualitative and quantitative agreement with experiments and DNS data is good. Due to sustained optimisation, the parallel efficiency is above 96% even for large processors numbers.
So far, only a part of MG’s features are used for LES-computations. In particular, the capability for dynamic grid adaption appears very attractive despite of considerable theoretical problems concerning subgrid-scale modeling.

REFERENCES