Electronic Transactions on Numerical Analysis.
Volume 15, pp. 152-164, 2003.
Copyright © 2003, Kent State University.
ISSN 1068-9613.

LARGE-EDDY SIMULATION AND MULTIGRID METHODS[*]

SANDRA NÄGELE[†] AND GABRIEL WITTUM[‡]

Abstract. A method to simulate turbulent flows with Large-Eddy Simulation on unstructured grids is presented. Two kinds of dynamic models are used to model the unresolved scales of motion and are compared with each other on different grids. Thereby the behaviour of the models is shown and additionally the feature of adaptive grid refinement is investigated. Furthermore the parallelization aspect is addressed.

Key words. LES, turbulence, multigrid, parallelization

AMS subject classifications. 65N55, 65Y05, 76F65

1. Introduction.

1.1. The Problem.

Flows of incompressible fluids are modeled by the Navier-Stokes equations

$$\begin{aligned}&\frac{\partial u_i}{\partial t} - \frac{\partial}{\partial x_j}\left(\nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right) + \frac{\partial}{\partial x_j}(u_i u_j) + \frac{\partial p}{\partial x_i} = 0\\ &\frac{\partial u_j}{\partial x_j} = 0\end{aligned}$$

with viscosity ν, velocity $\vec{u} = (u_1, ..., u_d)^T$, d the spatial dimension and pressure p. Throughout this paper Einstein's summation convention is applied unless stated otherwise. For fluids subject to linear material laws, the viscosity ν is constant. However, for small viscosities and large velocities the flow develops unordered small scale fluctuations of velocity and pressure. Turbulence has a slowdown and a mixing effect for the flow. It occurs everywhere in nature as well as in technology. Since the first description of turbulence as a phenomenon by Reynolds in 1893, turbulence and its generation is still not fully understood.

The multi-scale character of turbulence makes simulation of turbulent flows a difficult business. To account for the full nonlinear multi-scale effect of turbulence, the Navier-Stokes equations must be solved resolving the micro-scale effects (see e.g. [9]). This is not possible for flows on technical scales. Thus, depending on the scale of interest, different modeling approaches exist.

A full simulation of turbulent flows resolving all scales involved, so-called Direct Numerical Simulation (DNS), is restricted to micro-scale or low-turbulence problems. DNS on a technologically interesting scale is not possible and will not be possible for many years. Statistical averaging models, so-called Reynolds-averaged Navier-Stokes-Equations (RANS), are derived and closed by some empiric equations for additional unknown quantities, like the turbulent kinetic energy and dissipation. Though widely used in practice, these models only give rough approximations of the flow. In particular there are critical important quantities of the flow like the Reynolds stresses, which are in many cases approximated poorly by RANS models. However, RANS models offer a cheap and simple way to approximate coarse scale behaviour of turbulent flows.

[*] This work was supported by the Deutsche Forschungsgemeinschaft, SFB 359 and Center of Applied Scientific Computing at Lawrence Livermore National Laboratories. Received May 22, 2001. Accepted for publication September 12, 2001. Recommended by Klaus Steuben.

[†] Technische Simulation, IfI, Universität Heidelberg, email: sandra.naegele@iwr.uni-heidelberg.de

[‡] Technische Simulation, IfI, Universität Heidelberg, email: wittum@iwr.uni-heidelberg.de

A third way to model turbulence is the Large-Eddy Simulation (LES). The idea is to resolve the large scales which can be represented by the computational grid and to model structures smaller than the resolution of the grid by subgrid-scale models, see e.g. for an overview [17]. Due to the resolution requirement LES computations need finer grids than RANS simulations do, however, LES gives much better accuracy for critical turbulence quantities and thus using LES gets more and more in the range of simulations on technical scales.

1.2. Numerical Methods and Tools for Turbulent Flows. The methods used differ according to the modeling approach. For DNS high-order explicit schemes are used to discretize in time and in spite of the non-smoothness of the solutions due to high small-scale fluctuations high-order or even spectral methods for the discretization in space. Since it is assumed that all scales are resolved in DNS, people use structured equidistant Cartesian grids, allowing discretizations like spectral methods etc. This assumption also gives rise for explicit schemes in time, avoiding the necessity of constructing sophisticated solvers, but requiring a huge number of time-steps. Mostly academic software is used for the simulations.

With the RANS approach standard methods for the Navier-Stokes equations are applied, like finite volume discretizations in space, implicit schemes in time, and sometimes multigrid methods. There is a bunch of commercial software around, providing simple RANS models. Advanced models, however, are typically implemented in academic software, since several numerical issues are connected with those models. Primary issues are adaptivity, discretization, solver, and parallelization. Advanced strategies are found only in recently developed academic software. Numerical methods for turbulence computation using RANS models are still highly developed compared with the relatively new field of Large-Eddy Simulation. Therefore mainly approaches from DNS are used, i.e. explicit discretization in time, since the idea is to apply filtering only in space and higher order methods on structured grids etc.

However, LES may also be viewed as an improvement of RANS models, since they are only modeling a few scales in contrast to modeling all scales as in RANS methods. Then it may be reasonable to apply a filter in time too, opening the opportunity to use implicit methods and larger time steps. This gives rise to the investigation of solvers for LES. Multigrid as numerical multi-scale approach matches LES quite well in that respect. Choosing multigrid as solver for LES systems, the coarse-grid operator needs a discussion, since the model now depends on the computational grid. Another interesting topic is the issue of adaptive local grid refinement in LES. Usually adaptive refinement is used to balance the numerical truncation error throughout the grid. In case of LES the model itself depends on the grid-size. Thus, adaptivity in LES computations should be used to equilibrate the modeling error as well. To that end, a tool to estimate modeling errors has to be developed.

In the present paper we discuss the use of multigrid methods for Large-Eddy Simulation in a parallel unstructured grid environment. First we describe our LES setting addressing some of the multi-scale issues. Thereafter we present a discretization for the incompressible Navier-Stokes equations on unstructured grids in 3 space dimensions, then multigrid methods suited for LES computations and the parallelization and software framework are described. Finally we show results of computations. Problems computed include the flow past a square cylinder benchmarking problem and the 2d mixing layer problem showing vortex pairing.

2. LES: Framework, Filtering, Subgrid-scale Models. The basic principle in a Large-Eddy Simulation is, that large scales are resolved and only the unresolved small scales are modeled. To realize this, one needs a scale separation decomposing the unknowns in large and small scales. Therefore each unknown is split in a local average $\overline{f}_i$ and a subgrid-scale component f_i' where $f_i = \overline{f}_i + f_i'$ stands for a velocity component or pressure. The local averages are generated by the application of a filter operator. This operator is a convolution integral of the form:

$$\overline{f}(x,t) = \int_\Omega G_\Delta(x,y) f(y,t)\, dV$$

Throughout the paper we use a volume-average box-filter with

$$G_\Delta(x,y) = \begin{cases} \frac{1}{|\Omega_\Delta(x)|} & y \in \Omega_\Delta(x) \\ 0 & else \end{cases} \quad \text{and filter width } \Delta := \sqrt[d]{|\Omega_\Delta(x)|}$$

where Ω_Δ denotes the support of the filter function. For the subgrid-scale components a model has to be defined which will be described later.

To transform the governing equations system into one only depending on local averages, the filter operator is applied to the incompressible Navier-Stokes equations. Under the assumption that integration and differentiation commute the resulting system is:

$$(1) \qquad \frac{\partial \overline{u}_i}{\partial t} - \frac{\partial}{\partial x_j}\Big(\nu\Big(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\Big)\Big) + \frac{\partial}{\partial x_j}(\overline{u_i u_j}) + \frac{\partial \overline{p}}{\partial x_i} = 0$$

$$(2) \qquad \frac{\partial \overline{u}_j}{\partial x_j} = 0$$

The unclosed convection term has to be replaced by $\frac{\partial}{\partial x_j}(\overline{u}_i \overline{u}_j)$ only depending on averaged quantities. In order not to change the equation the subgrid scale stress tensor $\tau_{ij} := \overline{u_i u_j} - \overline{u}_i \overline{u}_j$ is introduced. The momentum equation then becomes:

$$(3) \qquad \frac{\partial \overline{u}_i}{\partial t} - \frac{\partial}{\partial x_j}\left(\nu\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right)\right) + \frac{\partial}{\partial x_j}(\overline{u}_i \overline{u}_j) + \frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\tau_{ij} = 0$$

It remains to specify a model for τ_{ij}. This will be called "model part" throughout the paper. There are different models in use, for example one developed by Germano [8] and slightly modified by Lilly [12], then there is the first and oldest model introduced by Smagorinsky [15] and a model developed by Zang et.al. [18]. Except the Smagorinsky model all others are based on a locally varying model parameter C and a dynamic determination of this parameter. Partly the models are based on a pure eddy viscosity assumption [8], [12], [15], where it is assumed that the anisotropic part of the subgrid scale stress tensor is proportional to the shear stress tensor. In the model developed by Zang et.al. [18] a slightly different approach is introduced. Their model consists of a mixture between an eddy viscosity and a scale similarity model. The model terms of these models for the anisotropic part of the subgrid-scale stress tensor read:

$$(4) \qquad \text{eddy viscosity model:} \quad \tau_{ij} - \frac{1}{3}\delta_{ij}\tau_{kk} = -2C\Delta^2|\overline{S}|\overline{S}_{ij}$$

$$(5) \qquad \text{mixed model:} \ \tau_{ij} - \frac{1}{3}\delta_{ij}\tau_{kk} = -2C\Delta^2|\overline{S}|\overline{S}_{ij} + \mathcal{L}^m_{ij} - \frac{1}{3}\delta_{ij}\mathcal{L}^m_{kk}$$

with $\overline{S}_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right)$, $|\overline{S}| = \sqrt{2\overline{S}_{ij}\overline{S}_{ij}}$ and $\mathcal{L}^m_{ij} = \overline{\overline{u}_i \overline{u}_j} - \overline{\overline{u}}_i \overline{\overline{u}}_j$.

For the determination of the model parameter C a dynamic approach is applied, based on the assumption that the subgrid scales can be modeled by the smallest resolved scales. This range of scales is derived by filtering equation (1) with a second filter $G_{\hat{\Delta}}$ with $\hat{\Delta} > \Delta$ resulting in:

$$(6) \qquad \frac{\partial \hat{\overline{u}}_i}{\partial t} - \frac{\partial}{\partial x_j}\left(\nu\Big(\frac{\partial \hat{\overline{u}}_i}{\partial x_j} + \frac{\partial \hat{\overline{u}}_j}{\partial x_i}\Big)\right) + \frac{\partial}{\partial x_j}\left(\hat{\overline{u}}_i\, \hat{\overline{u}}_j\right) + \frac{\partial \hat{\overline{p}}}{\partial x_i} + \frac{\partial}{\partial x_j}T_{ij} = 0$$

The unclosed term is again replaced in the same way as before but the model term is now called testgrid-scale stress tensor $T_{ij} := \widehat{\overline{u_i u_j}} - \hat{\overline{u}}_i \hat{\overline{u}}_j$. The same model as for τ_{ij} is introduced for T_{ij} depending on twice filtered variables but the same parameter C. Then there are two different representations with different sizes of resolution or scale separation. To compare these two representations, equation (3) is filtered with $G_{\hat{\Delta}}$ and subtracted from equation (6). The result is then:

$$L_{ij} := \widehat{\overline{u}_i \overline{u}_j} - \hat{\overline{u}}_i \hat{\overline{u}}_j = T_{ij} - \widehat{\tau_{ij}}. \tag{7}$$

The Leonard term L_{ij} represents the resolved turbulent stress and relates the subgrid-scale and testgrid-scale stress tensors with each other independent of the precise models for them.

Insertion of model (4) for T_{ij} and τ_{ij}:

$$\begin{aligned} T_{ij} - \frac{1}{3}\delta_{ij} T_{kk} &= -\,2C\hat{\Delta}^2 |\hat{\overline{S}}| \hat{\overline{S}}_{ij} \\ \tau_{ij} - \frac{1}{3}\delta_{ij} \tau_{kk} &= -\,2C\Delta^2 |\overline{S}| \overline{S}_{ij} \end{aligned}$$

into relation (7) leads to the following equation:

$$L_{ij} - \frac{1}{3}\delta_{ij} L_{kk} = -2C\hat{\Delta}^2 |\hat{\overline{S}}| \hat{\overline{S}}_{ij} + 2C\Delta^2 \widehat{|\overline{S}| \overline{S}_{ij}} =: 2CM_{ij}$$

Since this is a tensorial equation for a single parameter, a least squares approach is used to determine C.

$$\begin{aligned} \text{minimize:} \quad & Q = (L_{ij} - \frac{1}{3}\delta_{ij} L_{kk} - 2CM_{ij})^2 \\ \Rightarrow \quad & C = \frac{L_{ij} M_{ij}}{M_{kl} M_{kl}} \end{aligned}$$

For model (5) the result is very similar except that a slightly more difficult expression has to be minimized. First the models for the stress tensors are:

$$\begin{aligned} T_{ij} - \frac{1}{3}\delta_{ij} T_{kk} &= -2C\hat{\Delta}^2 |\hat{\overline{S}}| \hat{\overline{S}}_{ij} + \mathcal{L}^M_{ij} - \frac{1}{3}\delta_{ij} \mathcal{L}^M_{kk} \\ \tau_{ij} - \frac{1}{3}\delta_{ij} \tau_{kk} &= -2C\Delta^2 |\overline{S}| \overline{S}_{ij} + \mathcal{L}^m_{ij} - \frac{1}{3}\delta_{ij} \mathcal{L}^m_{kk} \end{aligned}$$

with $\mathcal{L}^M_{ij} = \widehat{\overline{\overline{u}_i \overline{u}_j}} - \hat{\overline{\overline{u}}}_i \hat{\overline{\overline{u}}}_j$. To describe the expression for the model parameter C a few auxiliary tensors are defined:

$$\begin{aligned} H_{ij} &:= \widehat{\overline{\overline{u}}_i \overline{\overline{u}}_j} - \hat{\overline{\overline{u}}}_i \hat{\overline{\overline{u}}}_j \\ I_{kk} &:= L_{kk} + \widehat{\mathcal{L}^m_{kk}} - \mathcal{L}^M_{kk} \qquad\qquad \text{isotropic part} \\ M_{ij} &:= -\hat{\Delta}^2 |\hat{\overline{S}}| \hat{\overline{S}}_{ij} + \Delta^2 \widehat{|\overline{S}| \overline{S}_{ij}} \end{aligned}$$

In this case the insertion of the terms in expression (7) results in:

$$\begin{aligned} \text{minimize:} \quad & Q = (L_{ij} - H_{ij} - \frac{1}{3}\delta_{ij} I_{kk} - 2CM_{ij})^2 \\ \Rightarrow \quad & C = \frac{(L_{ij} - H_{ij}) M_{ij}}{M_{kl} M_{kl}} \end{aligned}$$

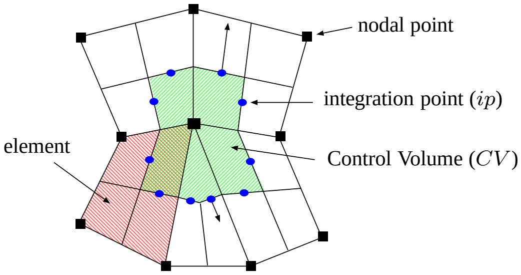


FIG. 1. *Sketch of a 2d control volume*

For this model the parameter is smaller than for the first model since the tensors L_{ij} and H_{ij} are approximately of the same size which can be easily seen by their definitions.

After determination of the model parameter C the insertion of the model term in equation (3) results in:

$$-\nu\left(\frac{\partial\overline{u}_i}{\partial x_j}+\frac{\partial\overline{u}_j}{\partial x_i}\right)+\delta_{ij}\overline{p}+\tau_{ij}=\begin{cases}-(\nu+\nu_t)\left(\frac{\partial\overline{u}_i}{\partial x_j}+\frac{\partial\overline{u}_j}{\partial x_i}\right)+\delta_{ij}\tilde{\overline{p}} & \text{eddy viscosity}\\[2ex] -\underbrace{(\nu+\nu_t)}_{:=\nu_{eff}}\left(\frac{\partial\overline{u}_i}{\partial x_j}+\frac{\partial\overline{u}_j}{\partial x_i}\right)+\mathcal{L}^m_{ij}+\delta_{ij}\tilde{\overline{p}} & \text{mixed}\end{cases}$$

with $\tilde{\overline{p}} := \overline{p} + \frac{1}{3}\tau^{eddy}_{kk}$ and τ^{eddy}_{kk} the trace of the eddy viscosity part of the model.

The Navier-Stokes equations are modified substituting viscosity ν by the effective viscosity ν_{eff} for the eddy viscosity type model and additionally for the mixed model a contribution from the scale similarity part. The isotropic contribution of τ_{ij} can not be represented by a eddy viscosity approach in an incompressible framework and is therefore added to the pressure term. Thus a modified pressure $\tilde{\overline{p}}$ is introduced in a Large-Eddy Simulation with a pure eddy viscosity model. For the mixed model the situation is different due to the scale similarity term. The isotropic part of the model coming from the shear stress tensor contribution is again added to the pressure term and also in this case the pressure is modified. But the contribution of the isotropic part to the pressure is reduced in comparison with model (4).

3. Discretization.

The equations are discretized with a finite volume method based on a vertex centered scheme where the control volumes are defined via dual boxes of the underlying finite element grid. A simple sketch of the resulting control volume in a 2d situation can be seen in figure 1. The construction, however, is general and applies to 3d as well.

After application of Gauss' theorem and splitting the integration over the whole control volume surface into a sum of integrations over subsurfaces the resulting system in discretized form reads:

$$\sum_{ip=ip_1}^{N_{ip}(CV)} (u_j\, n_j)\big|_{ip} = 0 \tag{8}$$

$$\frac{|CV|}{\Delta t}(U_i - U_i^{T-1}) + \sum_{ip=ip_1}^{N_{ip}(CV)} \left(u_i u_j n_j + p n_i - \nu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) n_j\right)\Big|_{ip} = 0. \tag{9}$$

$N_{ip}(CV)$ denotes the number of subsurfaces of the control volume surface and is therefore equal to the number of integration points of the control volume. In figure 1 an example

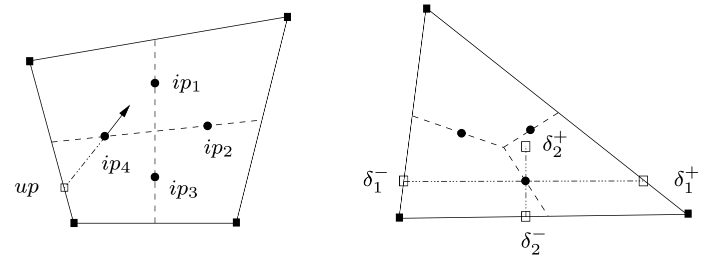


FIG. 2. *Sketch of elements with integration points and other positions*

for a control volume is shown with 10 integration points or subsurfaces respectively. Small letters correspond to integration point quantities and have to be further specified whereas capital letters denote nodal quantities and can be evaluated directly. The outer normal of each subsurface $\vec{n} = (n_1, ..., n_d)^T$ is scaled with the subsurface area to get a shorter notation. A quasi-Newton linearization of the convection term yields

$$\sum_{j=1}^{d} u_i u_j n_j = \sum_{j=1}^{d} u_i u_j^0 n_j$$

where u_j^0 stands for the last approximation of the integration point velocity u_j. The assembly of the discretized system can be done element-wise since mainly subsurface integrals have to be computed.

All unknowns are located in the nodes, thus the discretization would be unstable if the integration point quantities are interpolated via the Ansatz functions only, because the LBB-condition is not fulfilled in this case, [7]. To stabilize the system a special interpolation for the integration point velocities is constructed. The determination of this interpolation is based on the idea that the correct dependence of velocity and pressure is contained in the momentum equation itself. This kind of interpolation and stabilization was developed by Schneider and Raw [14] and was further modified by Schneider and Karimian [10]. To determine this kind of interpolation of u_i in each element and in each integration point the momentum equation is approximated with a very simple finite difference approach. The diffusion part is assumed to be a Laplacian for this finite difference approximation. In contrast to the above mentioned references, the Laplacian is then approximated with a standard 5-point stencil in 2d or 7-point stencil in 3d respectively. The convection term is linearized and afterwards discretized by an upwind method.

To explain the detailed form of the stabilization, in Figure 2 the position of all integration points is shown as well as the local flow direction at integration point ip_4. For the 5-point stencil in 2d, the corresponding positions are shown in the triangle to illustrate the application for the unstructured case. In 3d the procedure is straight forward and is applicable to hexahedra, tetrahedra, prisms and pyramids.

The finite difference approximation for one integration point reads:

$$\frac{u_i - u_i^{T-1}}{\Delta t} - \nu \sum_{j=1}^{d} \frac{u_i(\delta_j^+) - 2u_i + u_i(\delta_j^-)}{\delta_j^2} + |\vec{u}^0| \frac{u_i - u_i^{up}}{L_c} + \sum_{k=1}^{n_N} \frac{\partial N_k}{\partial x_i} P_k = 0 \,\Big|_{ip_l} \tag{10}$$

where $u_i(\delta_j^+)$ and $u_i(\delta_j^-)$ are the velocities interpolated at positions δ_j^+ and δ_j^- as indicated in Figure 2 and δ_j is the associated distance from the integration point. u_i^{up} denotes the upwind velocity, L_c is the distance between the integration point ip and the corresponding upwind position up and $\vec{u}^0$ denotes the last approximation of integration point velocity $\vec{u}$. N_k are the nodal Ansatz functions with $k = 1, ..., n_N$ and n_N the number of nodes of the element.

This leads to a system of equations depending on integration point velocities, nodal velocities and nodal pressures which can be solved directly in each element. The resulting integration point velocities are then inserted into the convection term of the momentum equation and first of all in the mass equation. By doing so, we introduce a pressure dependence in the mass equation in form of a Laplacian scaled with a constant times the mesh size squared.

Only the momentum equation was taken into account for the determination of the interpolation. But additionally the continuity equation can be considered to create an interpolation in the following way. The momentum equation minus u_i times the continuity equation will be discretized resulting in:

$$\begin{aligned}\frac{u_i - u_i^{T-1}}{\Delta t} - \nu \sum_{j=1}^{d} \frac{u_i(\delta_j^+) - 2u_i + u_i(\delta_j^-)}{\delta_j^2} + |\vec{u}^0| \frac{u_i - u_i^{up}}{L_c} + \sum_{k=1}^{n_N} \frac{\partial N_k}{\partial x_i} P_k \\ - u_i^0 \sum_{j=1}^{d} \sum_{k=1}^{n_N} \frac{\partial N_k}{\partial x_j} U_j(c_k) = 0 \,\Big|_{ip_l}\end{aligned}$$

For this interpolation the integration point velocity component u_i depends on $U_j(c_k)$ at the nodes c_k with $j = 1, ..., d$ and $k = 1, ..., n_N$, and on nodal pressures whereas in the first interpolation u_i only depends on $U_i(c_k)$, $k = 1, ..., n_N$, and nodal pressures. Therefore the second version is better suited for turbulent calculations with small viscosities where the dependence between the velocity components is stronger than in the laminar case.

4. Multigrid for LES. The discretized equation system is solved by a multigrid method with BiCGStab acceleration. The multigrid employs point-block ILU_β as smoother, standard restriction and prolongation and V(2,2)-cycle. By using multigrid in conjunction with LES one has to be aware that modeling and grid size are strongly coupled. Therefore on coarser meshes the large scales get larger and the portion of the modeled scales increases more and more. To prevent this effect, the model part is computed on the finest grid only and transfered to all coarser levels to insert the same model or more precisely to model the same scales. To realize that, the corresponding model terms have to be restricted to coarser levels. The scale similarity part contributes only to the right hand side of the equation system and therefore only to the defect. The multigrid cycle already restricts the defect to coarser levels and nothing has to be done for this part of the model. But the turbulent viscosity influences the Jacobi matrix. In order to allow assembling of the matrix on each grid level in a non-linear multigrid framework, the turbulent viscosity has to be restricted to coarser levels. This is done per simple injection. Therefore the effective viscosity $\nu_{eff} = \nu + \nu_t$ can be evaluated point-wise and is relatively smooth in comparison to the fine grid due to the coarser representation. The coarse grid operator is then assembled as described above in §3 where ν is replaced by ν_{eff}.

Unfortunately the model parameter C can oscillate strongly in space and time after the determination process. Numerical problems can arise due to these oscillations and should be smoothed. This can be achieved by averaging C in space and time. Anyway, C is determined in the least squares sense only, therefore the averaging or smoothing process does not damage the model but ensures numerical stability. Local spatial averaging over the testcell is applied to damp the spatial oscillations and a low pass filter of the form $C^{n+1} = \big(1-\varepsilon\big)C^{n+1} + \varepsilon C^n$ is used as temporal smoothing operator. Still the parameter varies strongly in space and time, but at least the high frequencies are damped by these modifications.

5. Implementation and Parallelization. The above methods have been implemented in the UG framework. UG is a software system for the simulation of PDE based models providing a lot of advanced numerical features. The main simulation strategies are the combination of adaptivity on locally refined unstructured grids, parallelism aiming at the use of massively parallel computers and multigrid methods, [4], [11]. Combining all these features in one software system is a challenging task for software engineering as well as algorithm development. It has been one of the main objectives of the Simulation in Technology Center at Heidelberg University, [2], [3], [4].

UG is based on a parallel programming model, called Dynamic Distributed Data (DDD), which has been developed by Birken, [5]. DDD is suited for graph based data structures and can be used independently of UG. DDD does the job of load migration and supports communication among distributed objects in a flexible and efficient way.

To ensure load balance the whole multigrid tree needs to be evenly distributed. To this end we form clusters of elements through the grid tree and distribute clusters, [1]. Methods used for static load balancing like recursive spectral bisection (RSB), recursive coordinate bisection (RCB) or others can be used to compute the graph partitioning, [11]. Most of the numerical part parallelizes well. The main difficulty is caused by parallelizing the smoother. This is done via a block-Jacobi approach. In particular if a problem needs strong smoothers like ILU this can mean a significant deterioration of the convergence for large numbers of processors especially on coarse grids.

To reduce the computational domain in a LES computation, periodic boundary conditions are introduced in directions where the flow is assumed to be periodic. This causes some problems concerning parallelization since the size of the discretization stencil at a periodic boundary should be the same as in the interior of the domain. Normally the boundary stencils are smaller than the interior ones, but in the periodic case this is no longer true. The connections to the nearest neighbours of each node are already provided by UG. But in the unstructured framework of UG, the additional matrix entries needed for periodicity have to be added by using geometrical informations. Since the load distribution is based on elements, the additional matrix entries can be easily found, if the corresponding elements and therewith their nodes are assigned to the same processor. Thus the load balancing strategy is adapted in such a way that the periodic boundary conditions can be realized similar to the sequential case despite the loss at processor boundaries as in the interior of the domain.

6. Numerical Results. As a 2d example the mixing layer problem at Re=500 based on the initial vorticity thickness δ and the free-stream velocity U_∞ is presented. The main feature of this flow is its temporal evolution resulting in pairing of eddies. The computational domain consists of a unit square with periodic boundary conditions in x-direction and prescribed velocity at the top boundary and in opposite direction at the bottom. The initial velocity distribution is given by

$$u(y) = U_\infty \ \tanh(\frac{y}{\delta}) \qquad \text{with } \delta = \frac{1}{56}$$

To enforce the formation of the fundamental eddies the initial velocity field is perturbed by superposing two divergence free functions of the form:

$$u' = \frac{\partial\Psi}{\partial y},\ v' = -\frac{\partial\Psi}{\partial x}, \qquad \text{with } \ \Psi = 0.001\, U_\infty e^{-(\frac{y}{\delta})^2} \cos(\phi\, x) \text{ and } \phi = 8\pi, 20\pi.$$

Three different grids are compared in this study to show that the large structures of the flow are well resolved on rather coarse grids and that adaptive refinement does not disturb these well resolved scales while still the main portion of the flow is fine enough to model the pairing effect. The first grid has got a grid size of $\frac{1}{256}$ in each direction and will be further referred to as ’fine’. It serves as a reference solution where the model part of the simulation is relatively small. The second one is a stretched grid with equidistant spacing in x-direction with $h = \frac{1}{256}$ and in y-direction the spacing is enlarged by a factor of 1.05 to the top and bottom boundary and is later called ’coarse’. The third one finally has got the same resolution as the ’coarse grid’ but only in a restricted area where it has been refined adaptively. This grid will be referred to as ’adap’. The adaption was performed by a simple indicator which compares the gradients on two levels to check if a refinement is necessary. The bottom left part of the ’adap’ grid is shown in Figure 5. The refinement region corresponds to the area of the domain where the interesting features of the flow develop. To compare these three grids with each other, the subgrid dissipation ϵ_{SGS} is calculated for all three realizations. The subgrid dissipation is a measure for the model contribution on the grid and will be integrated over the domain for comparison: $\int_\Omega \epsilon_{SGS}\, dV = \int_\Omega \tau_{ij} S_{ij}\, dV$.

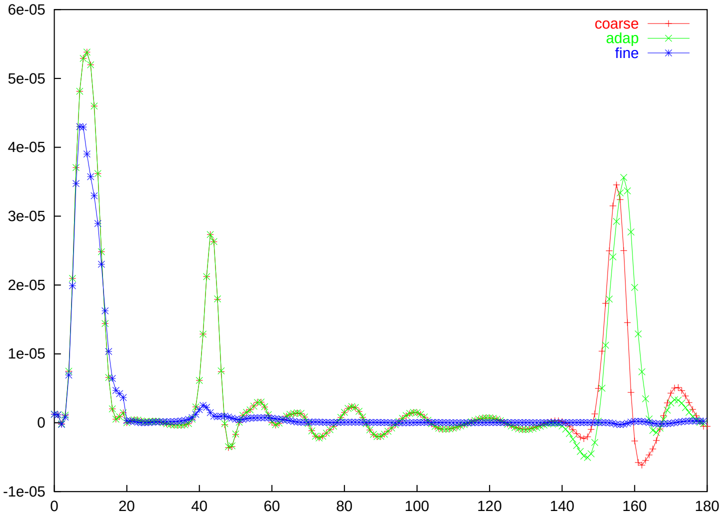


FIG. 3. *Subgrid dissipation for three different grids over time*

The evolution of this quantity is shown in Figure 3 for the mixed model on all grids. The first peak indicates the formation of the fundamental eddies, the second is the beginning of the first pairing and the last one belongs to the second pairing when only one eddy remains out of four fundamental ones. It can be seen that the model part slows down very strongly for the ’fine’ grid when less small structures are present through the pairing process. For the ’coarse’ and ’adap’ grid case the portion of small scales which can not be resolved and have to be modeled is much larger of course, due to the coarser resolution. In the beginning of the flow evolution, the ’coarse’ and ’adap’ case fit perfectly. They model the same portion of the flow, so that the adaption does not disturb the flow features during this period. At the end of the evolution they don’t fit any longer. But then the refined area of the grid is smaller than the size of the last remaining eddy which covers almost $\frac{2}{3}$ of the domain at that time and is very strongly influenced by the boundary conditions. Therefore the adaption did not deteriorate the result for the main period of the evolution process where no boundary influences are present. Since for model (4) the result is similar no curves are plotted for this case. Thus, the model used is not that important for this kind of flow. But the turbulence models do have a slightly different behaviour which can be seen in Figure 4 where the evolution of the vorticity

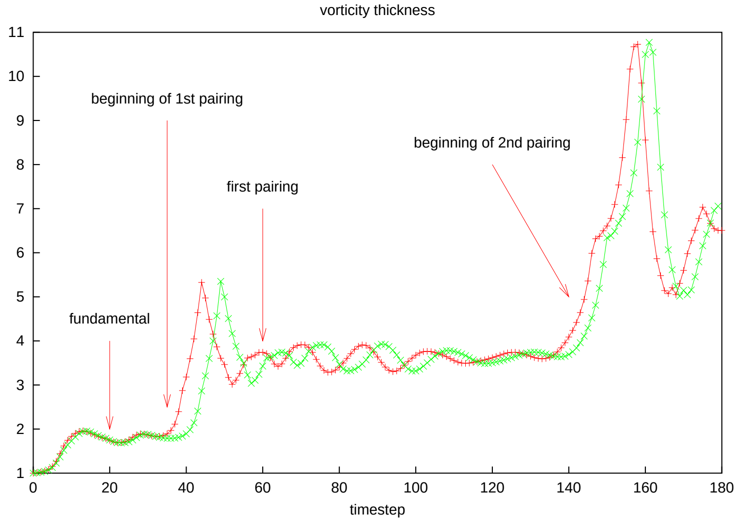


FIG. 4. *Vorticity thickness relative to δ_0 over time*

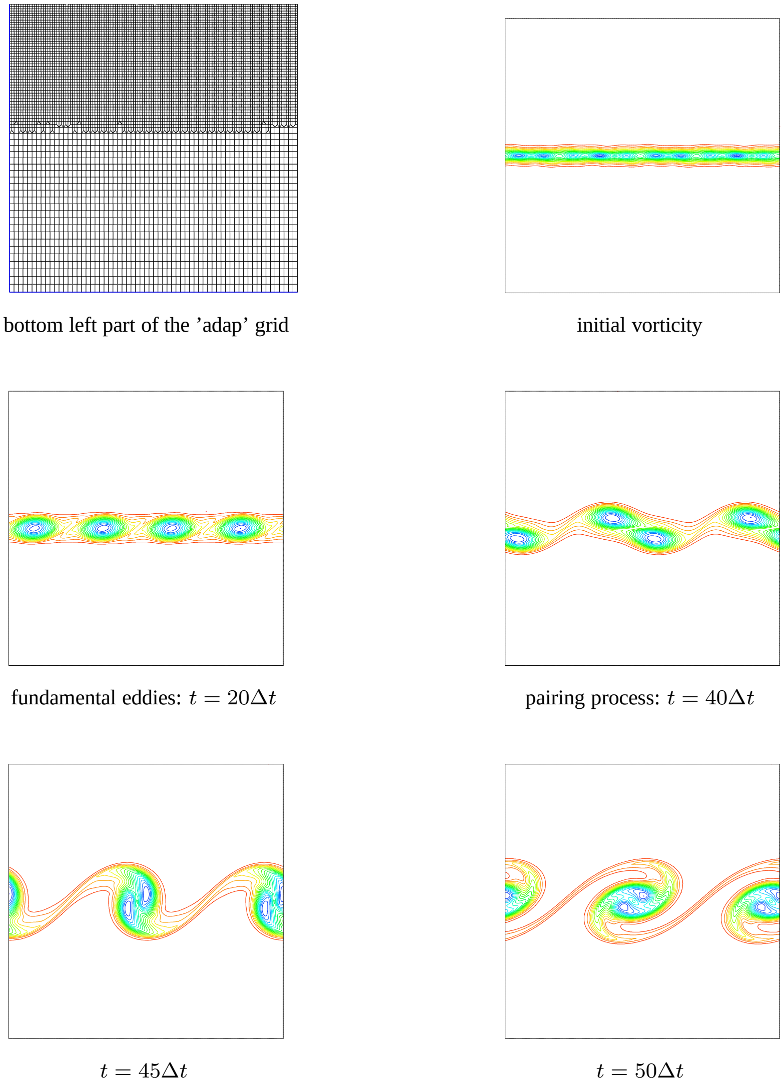
thickness over time is shown for models (4) and (5). The vorticity thickness is defined by $\delta := \frac{2U_\infty}{|\langle\omega_z\rangle|_{max}}$ where $\langle . \rangle$ denotes an average over the periodic direction and $\omega_z := \frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}$

The curve with the '+' symbol belongs to model (5) and the other one to model (4). Since for model (5) the pairing starts earlier than for model (4), this shows that backscatter, which means that energy flows from small to large scales, can not be represented very well by a pure eddy viscosity model. When pairing occurs, energy flows from small to large scales and thus two small eddies result in one large eddy. But since eddy viscosity models are dissipative, if C is positive, this phenomenon can not be modeled very well. Therefore the pairing starts later than in the mixed model case. Scale similarity models, however, are capable to model backscatter very well but not dissipation. The combination of them produces good results, since then the advantages of both models are combined. But still the evolution of the mixed and eddy viscosity model is very similar. At last, the evolution of the flow until the first pairing occurs is shown in Figure 5. The results compare very well to those of Boersma et.al., [6], who also made some investigations with adaptive refinement but using block-structured grids.

The solver for this 2d problem was a pure linear multigrid method where 4 levels and a V(2,2)-cycle with standard restriction and prolongation were used. The nonlinear reduction rate was set to 10^{-3}. Each time step needed approximately 5 nonlinear steps where at most 19 linear steps were needed for one nonlinear iteration. For both types of models the multigrid performed very similar and most of all very robust. Only in the beginning of the simulation more linear steps were needed. The main difficulty for the linear solver in the beginning is the zero starting solution for the pressure which doesn't fit to the prescribed velocity profile. As soon as the pressure solution was appropriate the solver worked well.

As a 3d example the flow around a long square cylinder is presented at a Reynolds number of 21400, [13], [16]. The domain and the boundary conditions are shown in Figure 6. In this case a BiCGStab method with linear multigrid as preconditioner was used to solve the linearized system in each nonlinear step. Since in 3d the flow phenomena are much more complicated, especially for flows around obstacles, the pure linear multigrid method with V(4,4)-cycle did not work. But in combination with BiCBStab the iteration converged well.

LES calculations need a relatively fine resolution, therefore the simulation was done in



bottom left part of the ’adap’ grid

initial vorticity

fundamental eddies: $t = 20\Delta t$

pairing process: $t = 40\Delta t$

$t = 45\Delta t$

$t = 50\Delta t$

FIG. 5. *Vorticity of the mixing layer problem*

parallel. The load balance for 16 processors can also be seen in Figure 6. This problem is very complicated because of steep gradients around the cylinder. As an example in Figure 7 the u_1 velocity component is plotted along the centerline of the cylinder in x-direction at subsequent instants. Around the cylinder the resolution has to be increased further to resolve these gradients. Uniform refinement would be too costly in this case. Bearing the 2d test with adaptivity in mind, which gave very good results, this approach will be applied also to the 3d case. Then only in the important regions of the domain the resolution will be increased and the portion of scales which have to be modeled reduces simultaneously. But a suitable criterion for the appropriate adaptive refinement has to be developed.

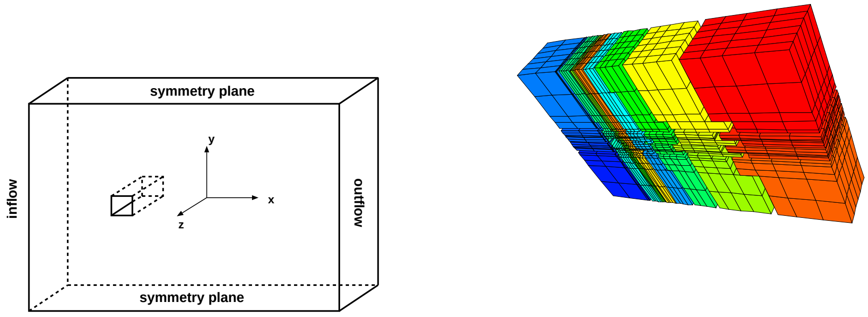


FIG. 6. *Cylinder domain with boundary conditions and load balance on 16 processors*

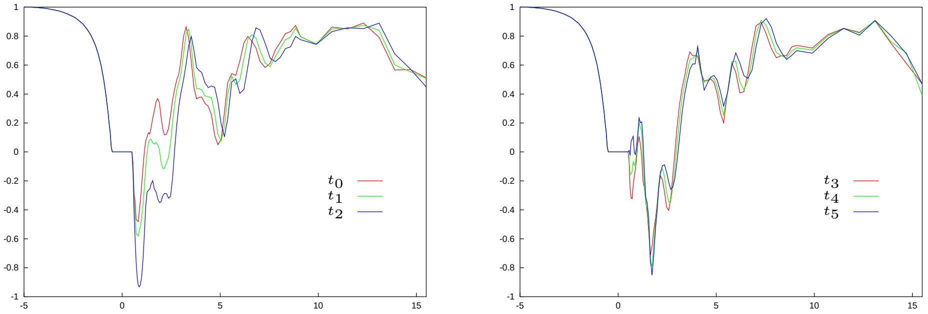


FIG. 7. *Line plots of the u_1 component in x-direction at times t_0 - t_5*

REFERENCES

[1] P. Bastian. Load balancing for adaptive multigrid methods. *SIAM J. Sci. Computing*, 19(4):1303–1321, 1998.

[2] P. Bastian, K. Birken, K. Johannsen, S. Lang, N. Neuss, H. Rentz-Reichert, and C. Wieners. Ug – a flexible software toolbox for solving partial differential equations. *Computation and Visualization in Science*, 1, 1997.

[3] P. Bastian, K. Birken, K. Johannsen, S. Lang, V. Reichenberger, C. Wieners, G. Wittum, and C. Wrobel. A parallel software-platform for solving problems of partial differential equations using unstr uctured grids and adaptive multigrid methods. In W. Jäger and E. Krause, editors, *High performance computing in science and engineering '98*, pages 326–339. Springer, 1998.

[4] P. Bastian, K. Johannsen, S. Lang, S. Nägele, C. Wieners, V. Reichenberger, G. Wittum, and C. Wrobel. Advances in high-performance computing: Multigrid methods for partial differential equations and its applications. In *High performance computing in science and engineering 2000*. Springer, 2000. submitted.

[5] K. Birken. *Ein Modell zur effizienten Parallelisierung von Algorithmen auf komplexen, dynamischen Datenstrukturen*. PhD thesis, Universität Stuttgart, 1998.

[6] B.J. Boersma, M.N. Kooper, F.T.M. Nieuwstadt, and P. Wesseling. Local grid refinement in large-eddy simulations. *Journal of Engineering Mathematics*, 32:161–175, 1997.

[7] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagranfian multipliers. *RAIRO Anal Numer*, 8:129–151, 1974.

[8] M. Germano. Turbulence: the filtering approach. *Journal of Fluid Mechanics*, 238:325–336, 1992.

[9] J.O. Hinze. *Turbulence*. McGraw Hill, 1975.

[10] S.M.H. Karimian and G. Schneider. Pressure-Based Control-Volume Finite Element Method for Flow at all Speeds. *AIAA Journal*, 33:1611–1618, 1995.

[11] S. Lang. *Parallele Numerische Simulation instationärer Probleme mit adaptiven Methoden auf unstrukturierten Gittern*. PhD thesis, Universität Stuttgart, 2001.

[12] D.K. Lilly. A proposed modification of the Germano subgrid-scale closure method. *Physics of Fluids A*, 4:633–634, 1992.

[13] W. Rodi, J.H. Ferziger, M. Breuer, and M. Pourquié. Status of Large Eddy Simulation: Results of a Workshop. *Journal of Fluids Engineering*, 119:248–262, 1997.

[14] G.E. Schneider and M.J. Raw. Control Volume Finite-Element Method for Heat Transfer and Fluid Flow Using Colocated Variables. *Numerical Heat Transfer*, 11:363–390, 1987.

[15] J. Smagorinsky. General circulation experiments with the primitive equations I. The basic experiment. *Monthly Weather Review*, 9:99–164, 1963.

[16] P.R. Voke. Flow Past a Square Cylinder: Test Case LES2. In P. Voke J. Chollet and L. Kleiser, editors, *Direct and Large-Eddy Simulation II: Proceedings of the Second ERCOFTAC Workshop*, pages 355–374. Kluwer Academic, 1997.

[17] D.C. Wilcox. *Turbulence Modeling for Computational Fluid Dynamics*. DCW Industries, 1993.

[18] Y. Zang, R.L. Street, and J.R. Koseff. A dynamic mixed subgrid-scale model and its application to turbulent recirculating flows. *Physics of Fluids A*, 12:3186–3195, 1993.