III

Publication III

Majander P., Siikonen T. , A parallel multi-block Navier-Stokes solver for large-eddy simulation in complex flows, 8th Finnish Mechanics Days, Espoo, Finland, pp. 395-406, June 12-13, 2003.

A PARALLEL MULTI-BLOCK NAVIER-STOKES SOLVER FOR LARGE-EDDY SIMULATION IN COMPLEX FLOWS

P. MAJANDER AND T. SIIKONEN

Laboratory of Applied Thermodynamics, Helsinki University of Technology
P.O. Box 4400, FINLAND
e-mail: Petri.Majander@hut.fi

ABSTRACT

A parallel pressure-based Navier-Stokes solver. The spatial discretization is based on a co-located finite-volume tecnique and a body fitted co-ordinate system. The solution is based on the traditional SIMPLE method. Time-dependent problems are solved with an implicit three-level method. The parallelization is based on a division of a computational domain into separate blocks and assigning each block into a processor of its own. Several flow cases are studied to test the solver, a steady-state cavity flow, a partly turbulent cavity flow, a fully developed turbulent pipe flow and finally jet in cross flow that has been studied experimentally by Crabb et. al [3]. The results are compared with the measurements and the efficiency of the parallelization is shown.

1 INTRODUCTION

The time-dependent solution of turbulent flow is computationally extremely intensive. In a direct numerical simulation (DNS) essentially all scales present in the flow field are solved. Due to the need of huge computational resources, DNS is usually applied with a relatively low Reynolds number in basic flows such as a channel flow. Spectral methods are often utilized because they reduce the number of the grid points needed due to their accuracy. The application of spectral methods in complex domains is difficult which makes them less appealing as engineering tools.

In a large-eddy simulation (LES) the small-scale motion is not solved which reduces the computational cost. The effect of the small-scale motion is usually modelled with a subgrid-scale model (SGS -model). As the computational power has increased, the applications in engineering have become feasible. Therefore, low-order finite-difference solutions in LES have recently become popular as shown by publications [1] and [2] to name just two.

In this study a parallel multi-block solver for incompressible flow is described. A spatial discretization is based on a co-located and structured finite-volume method with Rhie and Chow pressure coupling. An iterative three-level imlicit time stepping is implemented. The parallelization is based on a division of a computational domain into separate blocks, which are assigned in separate processors. The Message-Passing Interface (MPI) is used for exchanging the boundary data.

In the next sections the numerical method is derived and the algorithm of the parallel solver is presented. In Section 4, test cases are described and results are shown. The efficiency of the parallelization is studied. Also, a high-Reynolds number large eddy simulation of a jet in crossflow (JICF) is performed and some preliminary results are shown. The JICF has been studied experimentally by Crabb et. al [3].

2 GOVERNING EQUATIONS

The Navier-Stokes equations for isothermal incompressible flows are written as

$$\begin{aligned} \frac{\partial u_i}{\partial x_i} &= 0 \\ \frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} &= -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}, \end{aligned} \tag{1}$$

where u_i is the velocity, p is the pressure and ν is the kinematic viscosity. The LES equations are formally derived by applying a filtering operation: a filtered variable is defined as

$$\bar{f}(\mathbf{r}, t) = \int_V G(|\mathbf{r} - \mathbf{r}'|) f(\mathbf{r}', t) d\mathbf{r}', \tag{2}$$

where V is the volume of filtering and G is the filter function. After filtering Eqs. (1) the LES equations take the following form

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0 \tag{3}$$

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} + \nu \frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_j}, \tag{4}$$

where the SGS stresses are

$$\tau_{ij} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j. \tag{5}$$

Boussinesq approximation

$$\tau_{ij} - \frac{\delta_{ij}}{3}\tau_{kk} = -2\nu_T \overline{S}_{ij} = -\nu_T \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i} \right) \tag{6}$$

relates subgrid-scale stresses τ_{ij} to the eddy viscosity ν_T the resolved-scale strain rate tensor $\overline{S}_{ij}$. The Smagorinsky model for the eddy viscosity is written as

$$\nu_T = (C_s \Delta)^2 |\overline{S}|, \tag{7}$$

where C_s is called a Smagorinsky constant, Δ is a length scale and $|\overline{S}| = \sqrt{2\overline{S}_{ij}\overline{S}_{ij}}$ [17]. The length scale represents the cell size and it is usually computed as $V^{1/3}$. A value 0.18 for C_s has been evaluated for the inertial range dynamics [10]. In the presence of shear flow a smaller value must be used, which was first discovered by Deardorff [4]. He studied a channel flow and found that a smaller value for $C_s = 0.1$ had to be used in the presence of a strong shear-driven turbulence. This observation has since been confirmed by many studies. Furthermore, in the vicinity of the walls, the eddy viscosity must be reduced. This is usually carried out with the van Driest damping function. Germano et al. [6] presented a dynamic SGS model, where the model coefficient is calculated during the simulation. This is based on the application of two different filters. In addition to the grid filter G, a test filter $\widehat{G}$ is applied. The test filter width $\widehat{\Delta}$ is larger than the grid filter width Δ, usually $\widehat{\Delta} = 2\Delta$. The term grid filter implies that usually discretization takes care of the filtering in numerical computations without any explicit procedure. The grid filter and the test filter are applied to the momentum equations (1) to obtain the following equation:

$$\frac{\partial \widehat{\overline{u}}_i}{\partial t} + \frac{\partial \widehat{\overline{u}}_i \widehat{\overline{u}}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \widehat{\overline{p}}}{\partial x_i} - \frac{\partial T_{ij}}{\partial x_i} + \nu \frac{\partial^2 \widehat{\overline{u}}_i}{\partial x_j \partial x_j}, \tag{8}$$

where subtest stresses are given by

$$T_{ij} = \widehat{\overline{u_i}\overline{u_j}} - \hat{\overline{u}}_i \hat{\overline{u}}_j. \tag{9}$$

The test filter is now applied to the grid-filtered equations (4) ;

$$\frac{\partial \hat{\overline{u}}_i}{\partial t} + \frac{\partial \hat{\overline{u}}_i \hat{\overline{u}}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \hat{\overline{p}}}{\partial x_i} - \frac{\partial \widehat{\tau_{ij}}}{\partial x_i} - \frac{\partial \mathcal{L}_{ij}}{\partial x_i} + \nu \frac{\partial^2 \hat{\overline{u}}_i}{\partial x_j \partial x_j}, \tag{10}$$

$$\mathcal{L}_{ij} = \widehat{\overline{u_i}\overline{u_j}} - \hat{\overline{u}}_i \hat{\overline{u}}_j. \tag{11}$$

Using Eqs. (8) and (10), the expression for $\mathcal{L}_{ij}$ can be written as

$$\mathcal{L}_{ij} = T_{ij} - \hat{\tau}_{ij} \tag{12}$$

Eqs. (9) represent the subtest-scale stresses whose length scale is less than the subtest filter width $\hat{\Delta}$. Eqs. (12) represent then the resolved turbulent stresses by the scales between the grid filter width Δ and the subtest filter width $\hat{\Delta}$. Let us model both τ_{ij} and T_{ij} by the same functional form of the Smagorinsky model:

$$\tau_{ij} - \frac{\delta_{ij}}{3}\tau_{kk} = -2C\Delta^2 |\overline{S}| \overline{S}_{ij} = -2C\beta_{ij}, \tag{13}$$

$$T_{ij} - \frac{\delta_{ij}}{3}T_{kk} = -2C\hat{\Delta}^2 |\hat{\overline{S}}| \hat{\overline{S}}_{ij} = -2C\alpha_{ij}. \tag{14}$$

In order to solve C from Eqs. 13 and 14 Piomelli and Liu [13] developed the following localized dynamic model

$$\mathcal{L}_{ij} - \frac{\delta_{ij}}{3}\mathcal{L}_{kk} = -2C\alpha_{ij} + 2\widehat{C\beta_{ij}}, \tag{15}$$

where the coefficient C under the filter is replaced with an estimate C^*, which is assumed to be known. Eqs. (15) can be solved by the contraction :

$$C = -\frac{1}{2}\frac{\left(\mathcal{L}_{ij} - 2\widehat{C^*\beta_{ij}}\right)\alpha_{ij}}{\alpha_{mn}\alpha_{mn}} \tag{16}$$

For C^* one can use the value of the previous time step or iteration C^{n-1}. An important feature is that the coefficient calculation is based on the local information. However, the values for C must usually be limited in order to prevent destabilizing negative viscocity [13].

3 FLOW SOLVER

3.1 Conservation Form

The continuity equation in a conservation form becomes

$$\int_S \rho \mathbf{V} \cdot d\mathbf{S} = 0, \tag{17}$$

where ρ is the density and $\mathbf{V} = u\mathbf{i} + v\mathbf{j} + w\mathbf{j}$ is the velocity of the fluid. The integration is taken over the control volume faces. This constraint simply states that at every moment the mass flow out from the control volume equals the mass flow in. The momentum equation in the x-direction is written as

$$\frac{\partial}{\partial t}\int_V \rho u dV + \int_S \rho u \mathbf{V} \cdot \vec{n} dS + \int_S p n_x dS - \int_V \rho g_x dV - \mu \int_S \left(n_x \frac{\partial u}{\partial x} + n_y \frac{\partial u}{\partial y} + n_z \frac{\partial u}{\partial z} \right) dS = 0. \tag{18}$$

Above, p is the pressure, $\rho g_x$is the body force and $\mathbf{n} = n_x\mathbf{i} + n_y\mathbf{j} + n_z\mathbf{k}$ is a unit normal of the control volume face. The equations in the y- and z-directions are obtained by replacing the velocity component u with v and w and the wall-normal component n_x in the pressure term with n_y and n_z, respectively.

3.2 Momentum Equations

Cartesian equations are solved sequentially. Within each Cartesian directions, a contribution from all curvilinear grid directions is computed. A following time-stepping method is used in solving Eq. (18), which, after discretization for a node (ijk) becomes

$$V_{ijk}\left(\frac{(1+\gamma)\rho u^{n+1} - (1+2\gamma)\rho u^n + \gamma\rho u^{n-1}}{\Delta t}\right)_{ijk} = -\sum_{\phi=1}^{faces}(\hat{F}_\phi^{inv} - \hat{F}_\phi^{visc})^{n+1}S_\phi, \quad (19)$$

where the parameter γ defines a first-order accurate implicit Euler method with value 0 and a second-order accurate three-level imlicit method (3-LI) with value 0.5. The inviscid flux in the x-direction at the cell face ϕ is

$$F_\phi^{inv} = \hat{F}_\phi^{inv}S_\phi = S_\phi\rho u_\phi\mathbf{V}_\phi\cdot\mathbf{n}_\phi + (Sn_xp)_\phi = \dot{m}_\phi u_\phi + (Sn_xp)_\phi. \quad (20)$$

Here, $\dot{m}_\phi = S_\phi\rho\mathbf{V}_\phi\cdot\mathbf{n}_\phi$ is a mass flow through the face ϕ. In linearization of the viscous fluxes, a thin-shear-layer approximation is used in each co-ordinate direction instead of calculating the derivatives by using the generalized divergence theorem of Gauss. This approximation is done to avoid a large molecule in the matrix. In a Cartesian grid, both approximations are the same. At the face $(i+\frac{1}{2}, jk)$ (i.e. in the i-direction), the simplified viscous flux for the u-momentum equation becomes

$$F_{i+\frac{1}{2},jk}^{visc} = \frac{S\mu}{\Delta\xi}\left(u_{i+1,jk} - u_{ijk}\right), \quad (21)$$

where $\Delta\xi$ is the distance between the nodes (ijk) and $(i+1, jk)$. After linerization, a Poisson type equation is obtained for the velocity increment Δu

$$\sum_{nb}A_{nb}\Delta u_{nb} = -\sum_{\phi=1}^{faces}(Sn_x)_\phi\Delta p_\phi - \sum_{\phi=1}^{faces}(\hat{F}_\phi^{inv} - \hat{F}_\phi^{visc})^k - D_{ijk}^k, \quad (22)$$

where the sum on the left-hand side is over the neighbour nodes. In $\Delta u = u_{ijk}^{n+1} - u_{ijk}^k$ the state k lies between the solved state n and the state $n+1$ to be computed. The first term on the right hand side includes the pressure at the time level $n+1$ and in an iterative solution the term is ignored. Term D_{ijk}^k includes the time derivative as a source term. The coefficients and a more accurate derivation can be found in [11].

3.3 Pressure Coupling

The solution of the momentum equation must be coupled with pressure. The last two terms of Eq. (22) cancel when the iteration is converged. The linearized increments Δu and Δp are replaced in the following by iterative corrections u' and p':

$$\sum_{nb}A_{nb}u'_{nb} = -\sum_{\phi=1}^{faces}S_\phi n_{\phi x}p'_\phi, \quad (23)$$

where the standard simplification utilized in a SIMPLE method is to drop the non-diagonal terms from Eq. (23) to derive a manageable equation. The continuity equation states that

$$\sum_{\phi=1}^{faces_{ij}} (\rho(\mathbf{V}^* + \mathbf{V}') \cdot \mathbf{S})_\phi = 0$$
$$\sum_{\phi=1}^{faces_{ij}} (u' n_x S)_\phi + (v' n_y S)_\phi + (w' n_z S)_\phi = - \sum_{\phi=1}^{faces_{ij}} (\rho \mathbf{V}^* \cdot \mathbf{S})_\phi, \quad (24)$$

where $\mathbf{V}^*$ is the velocity field that does not obey the mass balance and $\mathbf{V}'$ is the iterative correction.

After some assumptions and manipulations of the terms we finally obtain the Poisson equation for the pressure corrections

$$B_{P,ijk} p'_{ijk} + B_{W,ijk} p'_{i-1,jk} + B_{E,ijk} p'_{i+1,jk} + B_{S,ijk} p'_{i,j-1,k} + B_{N,ijk} p'_{i,j+1,k} + B_{B,ijk} p'_{ijk-1} + B_{T,ijk} p'_{ijk+1} = -\Delta \dot{m}_{ijk}, \quad (25)$$

where the coefficients can be found in [11].

3.4 Calculation of Cell-Face Velocities

In a co-located, orthogonal grid, the velocity does not depend on the pressure at the same node (ijk), which can lead to a non-physical converged solution, a phenomenom known as a checker-board solution. Therefore methods have been developed to couple the pressure and the velocity in the calculation of mass balance. Rhie and Chow [15] have presented a remedy for uncoupling, which is in wide use. In a calculation of mass balance, the cell-face velocities are computed as

$$\bar{u}_{i-\frac{1}{2},jk} = \frac{1}{2}(\bar{u}_{i-1,jk} + \bar{u}_{ijk}) + C \frac{S_{i-\frac{1}{2},jk}}{4 A_{P,i-\frac{1}{2},jk}} [p_{i-2,jk} - 3p_{i-1,jk} + 3p_{ijk} - p_{i+1,jk}]. \quad (26)$$

The pressure term in Eq. (26) adds numerical dissipation. In a smooth pressure field, the term will become negligible. The pressure term is multiplied by a constant C in the range $0.01 - 0.5$ to adjust an optimum coupling [12].

3.5 Block Structure and Parallelization

A computational grid may be difficult to define on complex geometries by utilizing a structured grid. A well-known cure for this is a multi-block structure, where the computational domain is divided between many structured blocks that are connected together [9], [16]. A layer of ghost cells are used to set the boundary condition. Two layers of cells are defined to ensure a second-order accuracy if upwind-biased discretizations are used and for the Rhie and Chow interpolation. The cell values at the two layers at the edge of the block are transferred to the ghost cells of the neighbouring block and vice versa. The neighbouring blocks may have a different orientation. The connective boundary condition applies an orientation matrix and an offset vector at each connective face. These relate the indeces in the boundary data transfer as illustrated in Fig. 1. In the current version of the solver, each face must connect completely to another face and the grid lines must be continuous. The parallelization is based on dividing the blocks among different processors although many blocks can be assigned in one processor. The parallelization is implemented with a standart called Message-Passing Interface (MPI) [8]. Each block reserves a memory space which is used to transmit the boundary data with MPI_SEND and MPI_RECV operations. These operations are blocking; that is, the control does not return to the user program until the message has been received. Therefore, the order of communication between the processes is solved before the computation is

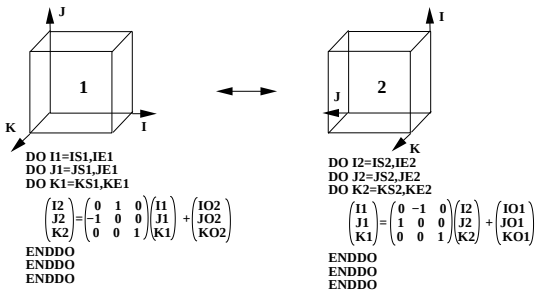

Figure 1: The connective boundary condition takes into account a different orientation between the blocks. An orientation matrix and an offset vector relates the indeces between the blocks.

started. Each sent message must be received in a proper order, otherwise the computation deadlocks. The grid should be divided as equally as possible to different processors in order to provide a good balancing. All processes compute equally the problem but only the master proces reads in an input file, a boundary data file and the grid. The master defines the boundary data needed and computes the order of the communication. The master transmits the data by MPI_BCAST command to all workers and the worker's part of the grid by MPI_SEND command. A flow chart of the solver is presented in Fig. (2). After reading the input data and the mesh each process computes the memory needed and allocates a one-dimensional table that contains all flow variables. No memory space is wasted even if the blocks would differ in size within one proces or between processes.

The computation is started with an initial guess, which is also copied at the previous time levels. The momentum equations (22) are solved in series with a multigrid solver. The mass balance is calculated with face velocities (26) that are coupled with the pressure. The mass flux error is used as a source in the pressure correction equation (25), which is solved with a multigrid (MG) solver. The MG algorithm solves the problem at the dense level first and moves the residual to the next coarsest level as a source. The coarse grid problem is approximated from the finer one. The process is called a Galerkin Coarse-Grid Approximation (GCA). After the coarsest grid level, the corrections are added to the unknown of each finer level, or the problem can be iterated also on the way up. The line Gauss Seidel (LGS) method is used at each level as a smoother. Reference [12] and Pensala (unpublished memorandum '3-D multigrid solver for Poisson-type equations' (in Finnish), HUT, 1996) give a detailed description of the MG solver. The pressure corrections p' and the velocity corrections are added to the pressures and the velocities multiplied by the underrelaxation coefficients α_p and α_u, respectively. Depending on the case, α_p usually varies from 0.1 to 0.8. In order for the velocities to satisfy the continuity equation, α_u should be 1 at this stage.

The Poisson equations for the momentum and pressure are solved independently in all blocks. During the iteration a Diriclet condition ($\Delta\mathbf{V}, p' = 0$) is set at the first ghost cell. After each subiteration cycle the boundary condition is updated for velocities and pressure ($\mathbf{V}, p$). The boundary value is lagging in time which is cured by iterating within a time step. In a similar vay to that of Bui [2], all the boundary values at the connecting boundary are copied in a single vector that is transmitted with a single message passing call. The criterion of convergence can be set in many different ways and it can depend on any primary variable or the balance of mass fluxes. Usually, an L_1- or L_2-norm of a residual of some variable must converge below a preset limit. As the convergence is reached, the solution of the new time level is started by updating the variables at the old and present time levels.

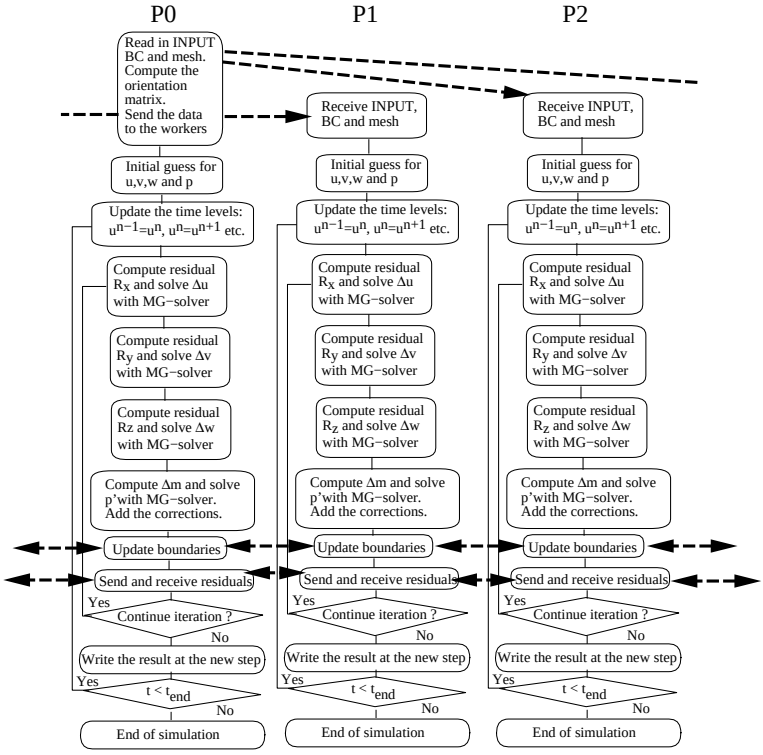

Figure 2: A flow diagram of the parallel code. The dashed arrows represent commmunication between the processors.

4 TEST CALCULATIONS

4.1 Steady State Cavity Flow

The first test case is a steady state cavity flow, where the Reynolds number $Re = \dfrac{U_w L}{\nu} = 400$, which means that the flow is stationary and two-dimensional. Here, U_w is the velocity of the upper lid of the cavity. A first-order Euler method is used to iterate a converged result and a second-order central discretization scheme is applied for the convective and diffusive terms. The grid points are equally distributed and the number of cells is $96 \times 96 \times 8$, where the cells in the k-direction are set for the MG solver. A no-slip condition is used at every solid wall and at the sliding wall. A zero-gradient condition is set for the pressure in the wall-normal direction. The domain is partitioned in equal sub-domains which are assigned in separate processors. The steady-state solution with the implicit Euler scheme was iterated until the L_2-norm of u-velocity was decresed below 10^{-10}. The computations were performed in an SGI-server equipped with eight 250 MHz R10000 processors. The u-velocity scaled by the wall velocity U_w along the vertical centerline in the cavity is shown in Fig. 3. With eight processors the number of iterations increased by 10 % compared to that needed by the single processor computation with a single domain. The CPU-time is decreased more than linearly as the number of processors is increased. Fig. 3 shows the speed-up of the computation per iteration number. The super-linear speed-up is probably due to the more efficient usage of a cache memory that the processor is capable of with the smaller domain assigned to it.

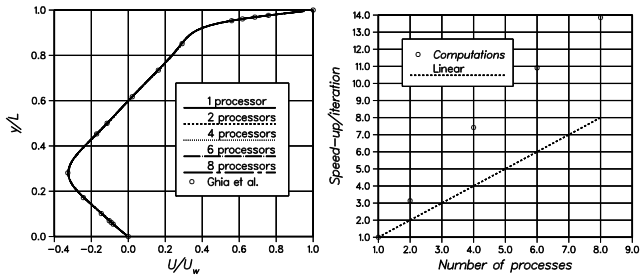

Figure 3: Cavity flow at Re=400. Converged u/U_w-velocity along the vertical centreline in the left and the speed-up per iteration. Reference result by Ghia et al. [7].

4.2 Turbulent Cavity Flow

The second test case is a cavity flow at a Reynolds number of 10000. The width of the cavity (W)is half of the length and the height (L) of the cavity. This flow has been studied experimentally by Prasad and Koseff [14].

The grid contains $64 \times 64 \times 32$ grid cells and it is geometrically streched from the wall. The domain is partitioned in equal 1, 2, 4, and 8 sub-domains which are assigned in separate processors. Large eddy simulations were calculated with and without a subgrid-scale model. The $L1$-norm of the mass balance is iterated to be smaller than 10^{-8}. With four processors a non-modelled computation requires 9.6 iterations and a modelled one 5.8 iterations per time step for this criterium. Fig. 4 represent the results along the vertical and horizontal centerline. The computations overpredict the thickness and the maximum velocity on the downstream and especially on the upstream wall. The insufficient resolution of the interior coarse grid might cause the discrepancies between the computed and the measured data. The dynamic model does not affect the results much on the vertical and horizontal centerlines. The model computes the largest viscosity near the downstream eddy, where the flow is turbulent. The speed-up is again superlinear although not as strongly as in the steady-state case where the mesh is smaller.

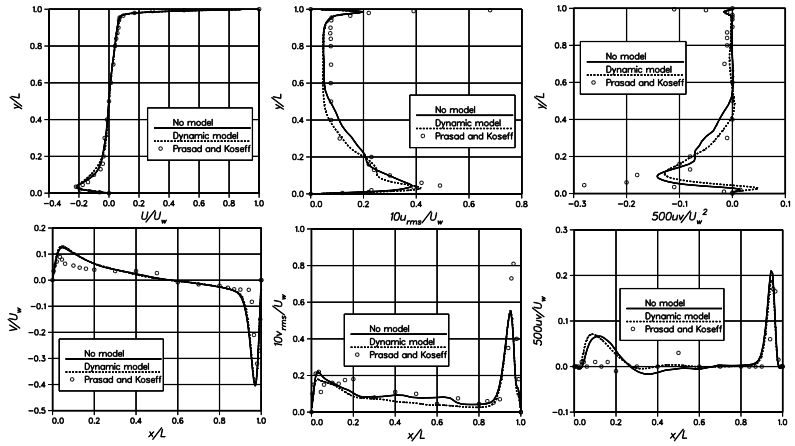

Figure 4: Cavity flow at Re=10000. Top : U/U_w-velocity, $10u_{rms}/U_w$ and $500uv/U_w^2$ along the vertical centreline. Bottom : V/U_w-velocity, $10v_{rms}/U_w$ and $500uv/U_w^2$ along the vertical centreline.

4.3 Fully Turbulent Pipe Flow

The third test case is a fully turbulent pipe flow at a friction velocity based Reynolds number $Re_\tau = \dfrac{u_\tau D}{\nu} = 621.4$, which was set by forcing the flow with a constant body force. This approximately equals $Re_b = \dfrac{U_b D}{\nu} \approx 10000$, where U_b is the bulk flow velocity and D is the diameter of the pipe. The shortest mesh is 1.2 D long and it consists of 5 blocks, each containing $32 \times 32 \times 32$ cells, altogether 163840 cells. The height of the two cells next to the wall is approximated to be around one in wall units ($r^+ = 1$) from the law of the wall. The streching ratio varies between 1.075-1.087 in the four outer blocks. The azimuthal mesh spacing $R\Delta\theta^+$ at the wall is 15 and the streamwise mesh-spacing Δz^+ is 24. This flow was first computed by using one Power4 -processor, after which all blocks were assigned to separate processors. The basic mesh was then multiplied and connected together, so that the computations were performed by using multiple pipe lenghts up to 30 processors. The pipe flow may be considered a simple flow physically, but computationally it possesses many complex features for the current Cartesian solver. Non-modelled large eddy simulations were calculated first. Within the time step of $0.05T$ ($T = D/U_b$) eight subiterations were taken which was considered to provide a sufficient convergence. The period during which the statistics was gathered varied between 200 T and 500 T depending on the mesh size. In the end the data was averaged also in the axial and radial directions. The velocity profile is slightly flatter than the measured one. The streamwise rms fluctuations are overpredicted whereas the radial rms fluctuations are underpredicted as seen in Fig. 5. The fluctuations obtained from the simulation using the shortest pipe lenght $1.2D$ deviate somewhat from the rest of the results. A localized SGS-model was

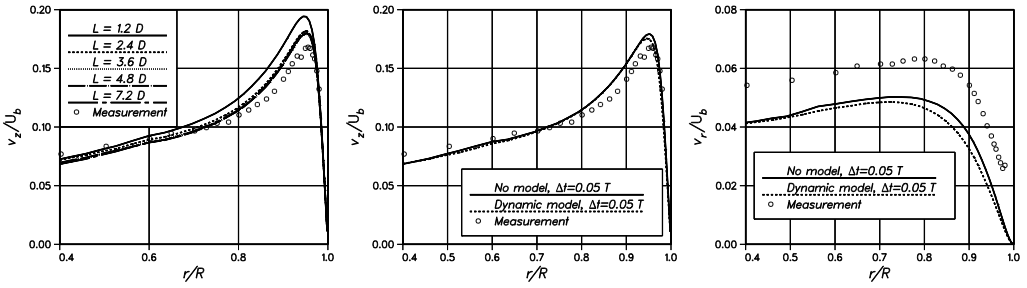

Figure 5: Results from the pipe flow at $Re_\tau = 621.4$. At left streamwise fluctuations with different mesh lengths. The effect of the SGS model on the streamwise and radial fluctuations at the centre and the right, respectively. The reference measurements by den Toonder and Nieuwstadt [5].

utilized with the pipe length of $3.6D$. The model dampens slightly the resolved turbulent intensities as shown in Fig. 5. Near the wall the model dampens itself excluding the need for the van Driest damping function, which requires a distance from the wall. In complex geometries the definition of the wall distance may be ambiguous. The parallelization was linear in these calculations.

4.4 Jet in crossflow

The setup of the jet in crossflow (JICF) is sketched in Fig. 6. The Reynolds number $Re_D = 46700$ referred to the pipe diameter is rather large for an LES of the whole jet. Therefore the computational domain has been reduced from that of the wind tunnel used by Crabb et al. [3], who have measured both average velocities and turbulent intensities in the case. In the LES a frictionless wall condition is applied at the lateral boundaries, and a no-slip condition at the lower wall. At the inlet a uniform velocity U_∞ is set and a zero-gradient condition is extrapolated at the outlet for all

variables. For the jet there is an inlet pipe one diameter D long. At the lower end, a fully developed average flow profile is applied. For the pressure the zero-gradient is set at all boundaries. The grid is geometrically streched in streamwise and jetwise directions. The smallest cell next to the wall is $0.002D$ high. The mesh is divided equally into 25 blocks consisting of 2 764 800 cells altogether. A constant Smagorinsky model was used in the computation because the dynamic model did not stabilize the computation. A rather large Smagorinsky constant $C_s = 0.17$ was used, and in the start of the calculation even higher value was needed. The flow was integrated during a time of 96 T, where $T = D/U_\infty$ and the statistics used to present the results here was gathered during the last 42 T. The recirculation zone seems to be located closer to the jet an closer to the wall than the experiment suggests (Figs. 7, 8). Turbulent fluctuations seem grow later in simulation, probably partly due to the Smagorinsky model and the steady inlet contitions. The jet profile peak right above the jet exit is higher than the measured one. The jet pipe profile might not be fully developed but flatter in the experiment. A closer examination of the results will be reported later elsewhere.

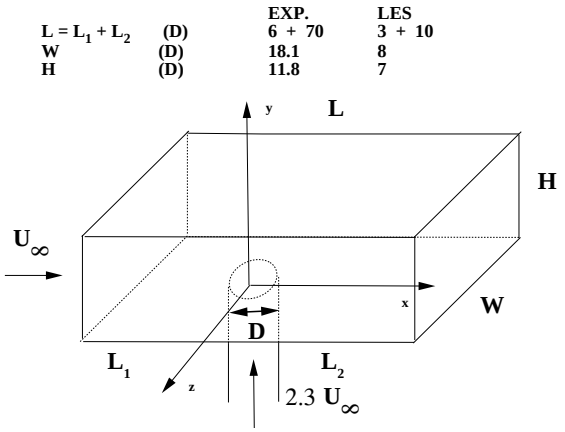

Figure 6: Schematics of the domain in the jet in crossflow. Experiment refers to the windtunnel used by Crabb et al. [3] whereas the LES refers to the computational domain size.

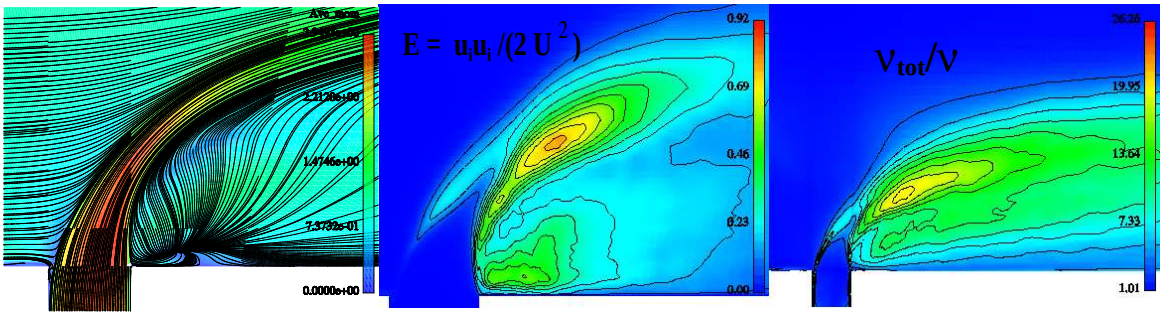

Figure 7: The averaged streamlines, the effective viscosity and the resolved turbulent energy shown from left to right, respectively.

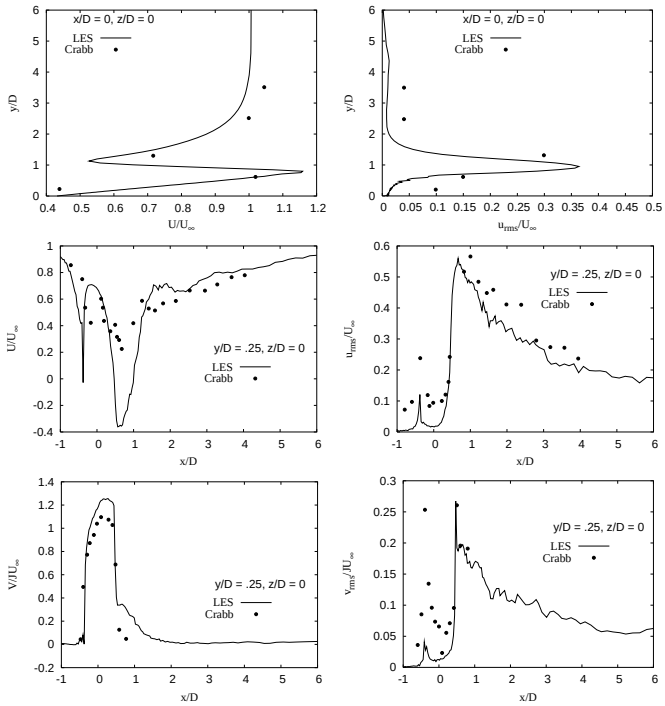

Figure 8: Top : The streamwise velocity and the rms velocity at x/D = 0 at the symmetry plane. Centre : The streamwise velocity and the rms velocity at y/D = 0.25 at the symmetry plane. Bottom : The vertical velocity and the rms velocity at y/D = 0.25 at the symmetry plane.

5 CONCLUSIONS

A parallel, finite-volume solver was developed for large-eddy simulation. The algorithm is based on a traditional SIMPLE technique and an implicit iterative time stepping is applied. The implicit part of the solution is solved independently in each block and proces and the boundary values are exchanged only after each outer iteration loop. This simplifies the code a lot without affecting much the convergence. Also, the amount of the message passing between the processes is reduced. The parallelization of the present solver shows good scaling in the present ideally balanced computations.

Several flows are simulated to test the solver. The dynamic model reduces the viscosity correctly in the vicinity of the wall without any ad hoc damping functions. Unfortunately, it does not always stabilize the computation. In the jet in crossflow a traditional Smagorinsky model was used.

ACKNOWLEDGEMENTS

This research project has been funded by the Graduate School of Computational Fluid Dynamics and IVO foundation, whose support is gratefully acknowledged. CSC, the Center of Scientific Computation, is acknowledged for providing us with the IBM cluster.

REFERENCES

[1] M. Breuer. Large eddy simulation of the subcritical flow past a circular cylinder: numerical and modeling aspects. *International Journal for Numerical Methods in Fluids*, 28:1281–1302,

1998.

[2] T.T. Bui. A parallel, finite-volume algorithm for large-eddy simulation of turbulent flows. *Computers & Fluids*, 29:877–915, 2000.

[3] D Crabb, D.F.G Durao, and J.H. Whitelaw. A round jet normal to a crossflow. *Transactions of the ASME: Journal of Fluids Engineering*, 103:568–580, 1981.

[4] J.W. Deardorff. A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *Journal of Fluid Mechanics*, 41:453–480, 1970.

[5] J.M.J. den Toonder and F.T.M. Nieuwstadt. Reynolds number effects in a turbulent pipe flow for low to moderate Re. *Physics of Fluids*, 9:3398–3409, November 1997.

[6] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A*, 7:1760–1765, 1991.

[7] U. Ghia, K. Ghia, and C. Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of Computational Physics*, 48:387–411, 1982.

[8] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface, Second Edition*. The MIT Press, Cambridge, Massachusetts, 1999. ISBN 0-262-57132-3.

[9] J. Hoffren. Time-accurate schemes for a multi-block Navier–Stokes solver. Report A-14, Helsinki University of Technology, Laboratory of Aerodynamics, 1992. ISBN 951–22–1350–8.

[10] M. Lesieur. *Turbulence in Fluids, Third Revised and Enlargened Version*. Kluwer Academic Publishers, Dordrecht, 1997. ISBN 0-7923-4416-2.

[11] P. Majander. Deveplopments in large eddy simulation. Report 128, Helsinki University of Technology, 2000. ISBN 951–22–4861–1.

[12] A. Miettinen. A study of the pressure correction approach in the colocated grid arrangement. Technical report no. 110, Helsinki University of Technology, Laboratory of Applied Thermodynamics, 1997. ISSN 1237–8372.

[13] U. Piomelli and J. Liu. Large-eddy simulation of rotating channel flows using a localized dynamic model. *Physics of Fluids*, 7(4):839–848, April 1995.

[14] A.K. Prasad and J.R. Koseff. Reynolds number and end-wall effects on a lid-driven cavity flow. *Physics of Fluids A*, 1(2):208–218, February 1989.

[15] C.M. Rhie and W.L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21(11):1525–1532, November 1983.

[16] A. Rizzi, P. Eliasson, I. Lindblad, C. Hirsch, C. Lacor, and J. Haeuser. The engineering of multiblock/multigrid software for navier-stokes flows on structured meshes. *Computers & Fluids*, 22(2):341–367, 1993.

[17] J. Smagorinsky. General circulation experiments with the primitive equations, part I: The basic experiment. *Montly Weather Review*, 91:99–152, 1963.