Turbulence Simulation on Sparse Grids using the Combination Method

M. Griebel [a] and W. Huber * [b]

[a] Institut für Informatik, Technische Universität München
Arcisstraße 21, D-80290 München, Germany

[b] Institut für Informatik, Technische Universität München
Arcisstraße 21, D-80290 München, Germany

In this paper, we study the parallel numerical solution of the Navier-Stokes equations with the sparse grid combination method. This algorithmic concept is based on the independent solution of many problems with reduced size and their linear combination. We describe the algorithm for three-dimensional problems and we report on its application to turbulence simulation. Furthermore, statistical results on a pipe flow for Reynolds number $Re_{cl} = 6950$ are presented and compared to results obtained from other numerical simulations and physical experiments.

1. Summary

Sparse grid techniques are very promising for the solution of linear PDEs. There, for three-dimensional problems, only $O(h_m^{-1}(\log(h_m^{-1}))^2)$ grid points are needed instead of $O(h_m^{-3})$ grid points like in the conventional full grid case. Here, $h_m = 2^{-m}$ denotes the mesh size in the unit cube. However, the accuracy of the sparse grid solution is of the order $O(h_m^2(\log(h_m^{-1}))^2)$ (with respect to the L_2- and $L_\infty-$norm) provided that the solution is sufficiently smooth (i.e. $\mid \frac{\delta^6 u(x,y,z)}{\delta x^2 \delta y^2 \delta z^2} \mid \leq \infty$). This is only slightly worse than the order $O(h_m^2)$ obtained for the usual full grid solution. For further details on sparse grids, see [4], [9].

For the solution of problems arising from the sparse grid discretization approach, two different methods have been developed in the last years. First, multilevel-type solvers for the system that results from hierarchical basis like finite element/finite volume or finite difference discretization (FE, FV, FD) and second, the so-called *combination method.* There, the solution is obtained on a sparse grid by a certain linear combination of discrete solutions on different meshes.

The multilevel-type solvers need hierarchical data structures. Thus, specially designed solvers are necessary (see [3]). In the case of the combination method, however, only simple data structures are needed. This method merely handles three-dimensional arrays. Furthermore, it is possible to use any PDE solver as a "black-box" solution method within the combination approach to get a solution on the different meshes involved.

*supported by Deutsche Forschungsgemeinschaft, AZ Ze 281/3-3,

Now, we intend to apply this method for turbulence simulation. Normally large-eddy- or k-ϵ-methods are used to solve the Navier-Stokes equations for turbulent flow problems. Both methods use models in order to cope with the necessary grid resolution and to make the problem tractable. Another way to deal with the Navier-Stokes equations is the direct numerical simulation method (DNS). The DNS is the most accurate approach to solve the Navier-Stokes equations because it does not use any kind of model. However, the disadvantage of this technique is that it requires an extremely high amount of memory and computing time, especially for problems with high Reynolds numbers. Therefore, it can normally be used for low Reynolds numbers only. Our aim is to use a DNS-code in combination with sparse grid discretization methods to compute problems with higher Reynolds numbers as well.

The remainder of this paper is organized as follows. In the next section, we briefly explain the theory of the combination method. Then, in section 3, we describe modifications of the combination formula necessary to apply it to turbulence simulation. In section 4, we explain, how the combination method can be applied within a given Navier-Stokes solver. Then, in section 5, we present statistical results of our numerical experiments for the application of the combination method to a turbulent pipe flow at a Reynolds number of 6950. We compare it with results from direct numerical simulations on a full grid as well as with results obtained from physical experiments and measurements with the laser-doppler anemometry (LDA), the hot-wire anemometry (HWA) and the particle image velocimetry (PIV), found in literature. In the last section, we shortly discuss the parallelization aspects of our method.

2. The combination method

For reasons of simplicity, we consider a partial differential equation

$$Lu = f$$

in the unit cube $\Omega =]0,1[^3 \subset \mathbb{R}^3$ with a linear, elliptic operator L of second order and appropriate boundary conditions. The usual approach is to discretize the problem by a FE, FV or FD method on an equidistant grid Ω_{h_m,h_m,h_m} with mesh size $h_m = 2^{-m}$ in x-, y-, and z-direction, and to solve the arising system of linear equations

$$L_{h_m,h_m,h_m} u_{h_m,h_m,h_m} = f_{h_m,h_m,h_m}. \tag{1}$$

We get a solution u_{h_m,h_m,h_m} with the error $e_{h_m,h_m,h_m} = u - u_{h_m,h_m,h_m}$ of the order $O(h_m^2)$, if u is sufficiently smooth. Here, we assume that u_{h_m,h_m,h_m} represents an appropriate interpolant defined by the values of the discrete solution on grid Ω_{h_m,h_m,h_m}. Extending this standard approach, we now study linear combinations of discrete solutions of the problem on different rectangular grids. Let Ω_{h_i,h_j,h_k} be a grid on $\overline{\Omega}$ with mesh sizes $h_i = 2^{-i}$ in x-direction, $h_j = 2^{-j}$ in y-direction, and $h_k = 2^{-k}$ in z-direction. In [9], the *combination method* has been introduced. We recall the definition of the combined solution $u^c_{h_m,h_m,h_m}$:

$$u^c_{h_m,h_m,h_m} := \sum_{i+j+k=m+2} u_{h_i,h_j,h_k} - 2 \sum_{i+j+k=m+1} u_{h_i,h_j,h_k} + \sum_{i+j+k=m} u_{h_i,h_j,h_k}. \tag{2}$$

Here, i, j, k range from 1 to m, where m is defined by the mesh size $h_m = 2^{-m}$ of the associated full grid. Thus, we have to solve $(m+1) \cdot m/2$ problems $L_{h_i,h_j,h_k} u_{h_i,h_j,h_k} = f_{h_i,h_j,h_k}$ with $i+j+k = m+2$, each with about 2^m unknowns, $m \cdot (m-1)/2$ problems $L_{h_i,h_j,h_k} u_{h_i,h_j,h_k} = f_{h_i,h_j,h_k}$ with $i+j+k = m+1$, each with about 2^{m-1} unknowns, and $(m-1) \cdot (m-2)/2$ problems $L_{h_i,h_j,h_k} u_{h_i,h_j,h_k} = f_{h_i,h_j,h_k}$ with $i+j+k = m$, each with about 2^{m-2} unknowns. Then, we combine their tri-linearly interpolated solutions. This leads to a solution defined on the sparse grid $\Omega^s_{h_m,h_m,h_m}$, see Figure 1. Definition (2) is illustrated in Figure 2.

Figure 1. The sparse grid $\Omega^s_{h_m,h_m,h_m}$ and the associated full grid Ω_{h_m,h_m,h_m} for the case $m = 4$.

The sparse grid $\Omega^s_{h_m,h_m,h_m}$ is a subset of the associated full grid Ω_{h_m,h_m,h_m}. Its grid points are defined to be the union of the grid points of the grids Ω_{h_i,h_j,h_k}, where $i+j+k = m+2$ and i, j and k range from 1 to m. For details on sparse grids, see [3], [15].

Altogether, the combination method involves $O(h_m^{-1}(\log_2(h_m^{-1}))^2)$ unknowns in contrast to $O(h_m^{-3})$ unknowns for the conventional full grid approach. Additionally, the combination solution $u^c_{h_m,h_m,h_m}$ is nearly as accurate as the solution u_{h_m,h_m,h_m} on the associated full grid. It can be proved (see [4], [9]) that the pointwise error satisfies

$$e^c_{h_m,h_m,h_m} := u - u^c_{h_m,h_m,h_m} = O(h_m^2(\log_2(h_m^{-1}))^2) \tag{3}$$

provided that the solution u is sufficiently smooth. This is only slightly worse than for the associated full grid, where the error is of the order $O(h_m^2)$.

For the proof we assume that for each u_{h_i,h_j,h_k} (interpolated from grid Ω_{h_i,h_j,h_k} to the domain Ω) an error splitting of the form

$$\begin{aligned} u_{h_i,h_j,h_k} \quad = \quad & u + C_1(h_i) \cdot h_i^2 + C_2(h_j) \cdot h_j^2 + C_3(h_k) \cdot h_k^2 + \\ & D_1(h_i,h_j) \cdot h_i^2 \cdot h_j^2 + D_2(h_i,h_k) \cdot h_i^2 \cdot h_k^2 + D_3(h_j,h_k) \cdot h_j^2 \cdot h_k^2 + \\ & E(h_i,h_j,h_k) \cdot h_i^2 \cdot h_j^2 \cdot h_k^2 \end{aligned} \tag{4}$$

holds for any fixed point $(x, y, z) \in \Omega$ with functions $C_1(h_i)$, $C_2(h_j)$, $C_3(h_k)$, $D_1(h_i, h_j)$, $D_2(h_i, h_k)$, $D_3(h_j, h_k)$, and $E(h_i, h_j, h_k)$ bounded by a constant C for all h_i, h_j, h_k. If we

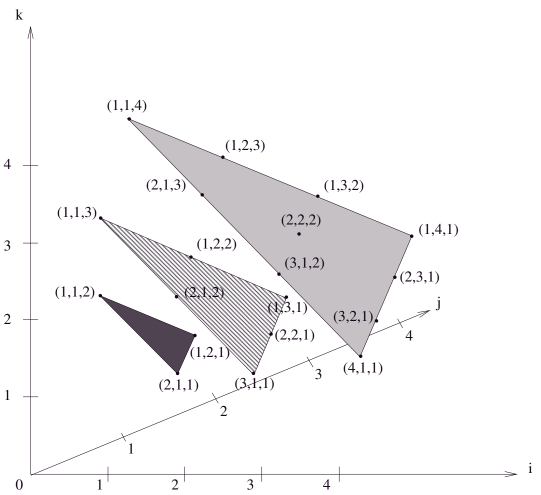

Figure 2. The indices (i, j, k) of all grids Ω_{h_i,h_j,h_k} arising in the combination process for $m = 4$.

plug (4) into (2), we obtain (3) after a short calculation. For details of the proof and its precise smoothness assumptions, see [4], [9]. Related techniques that are based on a direct application of the finite element method to sparse grids have been studied in [3].

3. Modifications of the combination method

In order to be able to apply the combination method within a solver for the simulation of turbulent flow in a pipe, definition (2) has to be modified in different directions. First, we have to switch from Cartesian to cylindrical coordinates. Second, we need a generalization of the combination formula to apply it also in the case of *different* numbers of grid points for the different coordinate directions. Third, to be able to control degeneration of the grids arising in the combination method and to avoid computations on grids with extreme aspect ratios of the mesh sizes, we want to limit the range of the combination formula somewhat.

3.1. Cylindrical coordinates

To apply the combination method to pipe flow computations, we have to allow for cylindrical coordinates. Now, the problem under consideration is defined on the unit cylinder $\Omega =]0,1[\times]0,2\pi[\times]0,1[\subset \mathbb{R}^3$ in the coordinate system (r, φ, z), where r denotes the radius, φ the angle and z the length. Compare also Figure 3.

For the discretization, again, we use a FD, FE or FV method on the grid Ω_{h_m,h_m,h_m} (in cylindrical coordinates), with mesh sizes 2^{-m} in r- and z-direction, respectively, and mesh size $2\pi \cdot 2^{-m}$ in φ-direction. Note that the discretization of a partial differential equation using cylindrical coordinates is more involved than for Cartesian coordinates, because a singularity appears on the middle axis of the cylinder. However, there exist methods that are able to deal with this problem. For examples, see [1], [12]. They result in a stable discretization of second order accuracy.

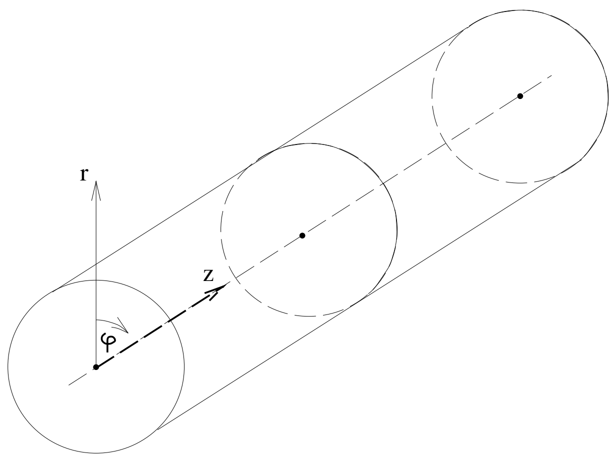

Figure 3. Cylindrical coordinates in the unit cylinder.

Similar to the case of Cartesian coordinates, sparse grids can now be defined for cylindrical coordinates, see Figure 4.

Figure 4. Front views ($r - \varphi-$direction) of the sparse grid $\Omega^s_{h_m,h_m,h_m}$ and the associated full grid Ω_{h_m,h_m,h_m} for the case $m = 4$.

Again, like in the case of Cartesian coordinates, a solution on the sparse grid $\Omega^s_{h_m,h_m,h_m}$ can be obtained by a linear combination of discrete solutions of the problem under consideration on different grids in cylindrical coordinates. To this end, let Ω_{h_i,h_j,h_k} denote a grid on the unit cylinder with mesh sizes $h_i = 2^{-i}$ in $r-$direction, $h_j = 2\pi \cdot 2^{-j}$ in φ-direction and $h_k = 2^{-k}$ in z-direction. Then, with formula (2), we obtain a *combination method in cylindrical coordinates* which produces a solution $u^c_{h_m,h_m,h_m}$ on the sparse grid $\Omega^s_{h_m,h_m,h_m}$ of the unit cylinder. Note, that, in the combination process, the tri-linear interpolation of the solutions u_{h_i,h_j,h_k} to some finer grids must be properly adjusted to cylindrical coordinates.

In principle, the advantages of sparse grids regarding the amount of grid points involved and the accuracy of the solution produced by the combination method carry over to the case of cylindrical coordinates. We confess, however, that the existence of an error splitting of the form as given in (3) has not yet been proved for cylindrical coordinates. But numerical experiments indicate that the combination method works nicely for cylindrical coordinates as well.

3.2. Different numbers of grid points for different directions

So far, the combination formula (2) produces a solution on a sparse grid which possesses the same amount of (boundary) grid points in each coordinate direction. However, to apply the combination method also in the case where *different* amounts of grid points for the different coordinate directions must be used, we have to generalize (2).

To this end, we have to change the notation somewhat. Now, let L, M, N denote the number of volume cells in the r-, φ- and z-direction, respectively, and let $\Omega_{(L,M,N)}$ denote the corresponding grid. Without loss of generality, let L, M, N be even and positive. Now, look for the *factorization* of L, M and N where

$$L =: 2^l \cdot \kappa_l, \quad M =: 2^m \cdot \kappa_m, \quad N =: 2^n \cdot \kappa_n, \tag{5}$$

with $\kappa_l, \kappa_m, \kappa_n, l, m, n \in \mathbb{N}, \kappa_l, \kappa_m, \kappa_n \geq 2$ and l, m, n maximal. The value p, which will be needed later on in (9), is determined by

$$p := min(l, m, n) + 1. \tag{6}$$

Now, we define the mesh sizes h_i, h_j, h_k of a grid Ω_{h_i,h_j,h_k} to be

$$h_i = k_L^{-1} \cdot 2^{-i}, \; h_j = k_M^{-1} \cdot 2^{-j} \cdot 2\pi, \; h_k = k_N^{-1} \cdot 2^{-k}, \tag{7}$$

with $k_L, k_M, k_N \in \mathbb{R}$. The mesh sizes h_i, h_j and h_k now take the values k_L, k_M and k_N into account which are defined via

$$L =: 2^p \cdot k_L, \; M =: 2^p \cdot k_M, \; N =: 2^p \cdot k_N. \tag{8}$$

The corresponding solutions are described by u_{h_i,h_j,h_k}. In this way, we are able to give a generalized combination formula for the case of different amounts of grid points/cells for the different coordinate directions by

$$u^c_{(L,M,N)} \;\; := \sum_{i+j+k=p+2} u_{h_i,h_j,h_k} - 2 \sum_{i+j+k=p+1} u_{h_i,h_j,h_k} + \sum_{i+j+k=p} u_{h_i,h_j,h_k}. \tag{9}$$

Now, i, j, k range from 1 to p, where p is given by (6). $u^c_{(L,M,N)}$ provides a solution on the *generalized* sparse grid $\Omega^s_{(L,M,N)}$ which is associated to $\Omega_{(L,M,N)}$ (by (7) and (9)). Its grid points are now defined to be the union of the points of the grids Ω_{h_i,h_j,h_k} according to (7) and (5), where $i + j + k = p + 2$ and each index i, j and k ranges from 1 to p.

3.3. Limitation of the range of the combination method

The use of extremely distorted grids with very high aspect ratio of the mesh sizes might cause certain problems. First, the numerical discretization scheme can be sensitive to it. Second, the solver on these grids might slow down and third, physical features of the solution might not be resolved properly on such grids, especially for turbulence simulation applications. We expect that "meaningless" solutions on such grids are canceled within the combination method and cause no harm. Nevertheless, we want to be able to control which grids and corresponding solutions will be involved in the linear combination process. Therefore, we must be able to limit the range of involved solutions and to use only a subset of all possible grids. We achieve this by introducing a new parameter $q \in \mathbb{N}$, $1 \leq q \leq p$, into the combination method. We define

$$u^{c,q}_{(L,M,N)} := \sum_{i+j+k=q+2} u_{h_i,h_j,h_k} - 2 \sum_{i+j+k=q+1} u_{h_i,h_j,h_k} + \sum_{i+j+k=q} u_{h_i,h_j,h_k} \tag{10}$$

where, each index i, j, k ranges now only from 1 to q. The mesh sizes h_i, h_j and h_k for grid Ω_{h_i,h_j,h_k} are now defined by $h_i = k_L^{-1} \cdot 2^{-i}$, $h_j = k_M^{-1} \cdot 2^{-j} \cdot 2\pi$ and $h_k = k_N^{-1} \cdot 2^{-k}$ where k_L, k_M and k_N are defined by

$$k_L = L/2^q, k_M = M/2^q, k_N = N/2^q. \tag{11}$$

For the case $q = p$, we obtain the old formula (9), but for values $q < p$ we limit the range of the combination method and thus exclude solutions u_{h_i,h_j,h_k} on extremely distorted grids. In the case $q = 1$, we just obtain the full grid solution $u_{(L,M,N)}$ and no combination takes place at all. The corresponding generalized sparse grid $\Omega^{s,q}_{(L,M,N)}$ consists of the union of the points of the grids Ω_{h_i,h_j,h_k} according to (7) and (11), where $i + j + k = q + 2$ and i, j, k range from 1 to q.

Now, in (10), we have to solve $q \cdot (q + 1)/2$ problems $L_{h_i,h_j,h_k} u_{h_i,h_j,h_k} = f_{h_i,h_j,h_k}$, $i + j + k = q + 2$, each with about $D := L \cdot M \cdot N \cdot 2^{2-2q}$ unknowns, $q \cdot (q - 1)/2$ problems $L_{h_i,h_j,h_k} u_{h_i,h_j,h_k} = f_{h_i,h_j,h_k}$, $i + j + k = q + 1$, each with about $D/2$ unknowns and $(q - 1) \cdot (q - 2)/2$ problems $L_{h_i,h_j,h_k} u_{h_i,h_j,h_k} = f_{h_i,h_j,h_k}$, $i + j + k = q$, each with about $D/4$ unknowns.

For different values of q, Table 1 shows the number of grids and the number of problems involved in the combination formula (10) and gives the fraction of main memory necessary to store the grids and corresponding solutions arising in (10) versus the main memory necessary for the full grid $\Omega_{(L,M,N)}$ and its corresponding solution $u_{(L,M,N)}$.

Table 1
Storage requirements.

q	Number of grids involved in the combination method	storage requirements in comparison to the full grid
1	1	1
2	4	7/8
3	10	31/64
4	19	55/256
5	31	86/1024
6	46	124/4096
7	64	169/16384
8	85	221/65536

Note that, if we would apply an optimal method for the solution of the different problems involving an amount of operations which is proportional to the number of grid points (i.e. multigrid), then the right column of Table 1 also gives more or less an impression of the run time improvement of the combination method in comparison to the full grid method.

4. Application to turbulence simulation

Laminar and turbulent flows are usually described by the Navier-Stokes equations. For their numerical solution, a lot of methods have been developed. Normally, turbulence models are used in order to make the problem tractable. However, turbulence models have problems with friction near the wall and difficulties in handling complex geometries. Thus, the best way to avoid such problems is to use a technique without any modeling like the DNS.

From the Department of Fluid Dynamics of the TU München, we obtained the DNS-code FLOWSI. The code FLOWSI is written in Fortran77. It uses a finite volume discretization method on staggered grids. For the discretization in space, a central-difference method is employed, and for the discretization in time, a Leap Frog method is used (see [11], [14]). The discretization error is of second order in space and time (for further details, see [10]). For a direct numerical simulation of a pipe flow with Reynolds number 6950, a full grid $\Omega_{(L,M,N)}$ with $L = 96, M = 128$ and $N = 256$ would be necessary.

Now, we combine the combination method and the FLOWSI-code to obtain a new, overall Navier-Stokes solver, that works on a sparse grid only. It possess all the advantages of the sparse grid approach, especially with respect to storage requirements and, as we will see in the following section, there is good hope, that the computed turbulent flow is statistically as accurate as a direct numerical simulation on the corresponding full grid.

To this end, we proceed as follows: At a time step t_i we use the Navier-Stokes code FLOWSI to treat the different discretized and unsteady problems that arise in the combination method (10) on the grids Ω_{h_i,h_j,h_k}, where $i+j+k = q+2$, $i+j+k = q+1$ and $i+j+k = q$. We stay on these grids for T successive time steps $\triangle t$ and perform with FLOWSI the necessary computations. Then, after T time steps, we assemble the combination solution $u^{c,q}_{(L,M,N)}$ for time step t_{i+T} from the different computed iterates $u^{t_{i+T}}_{h_i,h_j,h_k}$. After that, we project new iterates $u^{t_{i+T},new}_{h_i,h_j,h_k}$ for all the grids Ω_{h_i,h_j,h_k} involved in (10) from it and repeat this process with $t_i := t_{i+T}$. This is illustrated in Figure 5. The projection process is done by weighted restriction. Note that the weights must be proberly adjusted to cylindrical coordinates.

In this way, we obtain an outer time cycle with step size $T \cdot \triangle t$, where the actual iterates on the sparse grid $\Omega^{s,q}_{(L,M,N)}$ are assembled, and an inner time cycle where on every grid Ω_{h_i,h_j,h_k} an instance of the code FLOWSI operates concurrently with time steps $\triangle t$. Altogether, we obtain a new algorithm for the treatment of the Navier-Stokes equations based on the sparse grid approach.

Our new method has two parameters, q and T that directly and indirectly influence the performance of the solver with respect to the number of operations involved, the memory requirement and the accuracy of the obtained results of the turbulence simulation.

Recall, that for $q = 1$ only the full grid $\Omega_{(L,M,N)}$ is involved. In this case, we directly obtain the DNS simulation of FLOWSI on grid $\Omega_{(L,M,N)}$. However, for $q > 1$, we obtain, first, a sparse grid to work with, i.e. $\Omega^{s,q}_{(L,M,N)}$ in the outer iteration, and second, the

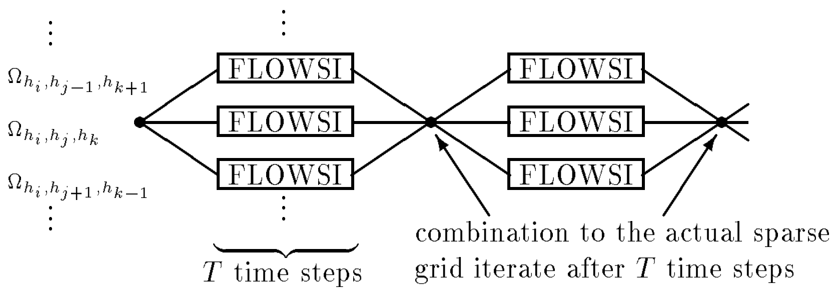

Figure 5. Integration of the combination method and FLOWSI.

smaller problems on the different grids Ω_{h_i,h_j,h_k} in the inner iteration. For $q = p$, we would exploit the full power of the combination method, but then, we must deal with quite distorted grids as well.

It can be noticed that, on such distorted grids, the iterates show laminar behaviour after a long simulation time, i.e. for large values of T. Therefore, it is absolutely necessary to combine the different iterates already after a few time steps, i.e. we have to choose the parameter T sufficiently small. The dependency of the accuracy of the simulation from q and T will be studied in the next section.

5. Turbulence statistics

Now, we report on the results of turbulence simulations obtained with the newly developed technique. We consider the flow through a pipe with a geometry as given in Figure 6. The normalized Reynolds number is $Re_{cl} = u_{cl} \cdot D/\nu = 6950$. Thus, for a direct numerical simulation, as it was performed in [5] and [13], a grid with $96 \times 128 \times 256$ cells is necessary. There, 41 samples were computed for the evaluation of different statistical moments.

Now, our aim is to evaluate the new algorithm described in the previous section for different values of its parameters q and T. The value of $\triangle t$ is chosen to be 1/50 of the Kolmogorov time scale. Regarding T, we performed experiments with $T = 10$, $T = 25$ and $T = 50$ so far. This corresponds to 1/5, 1/2 and 1 of the Kolmogorov time scale, respectively. Regarding q, we run our method for the values $q = 2, 3$ and 4. Then, in the combination method, $4, 10$ and 19 grids are involved. We started the simulations with a fully developed turbulent flow field as initial data. All computations were performed in single precision. In our experiments, we only computed 20 samples, until now.

Table 2 shows the mean flow property u_b/u_τ resulting from the DNS experiments in [5] and [13] and from the combination method algorithm (CM) as it was described in the previous section. Here, u_{cl} denotes the centerline velocity, u_b the mean velocity and u_τ the shear stress velocity. Recall, that the results of the DNS are equivalent to CM with $q = 1$. Note furthermore, that the combination method with $q = 3$ and the quite large $T = 50$ gave bad results. The flux in the main direction dropped down after a few

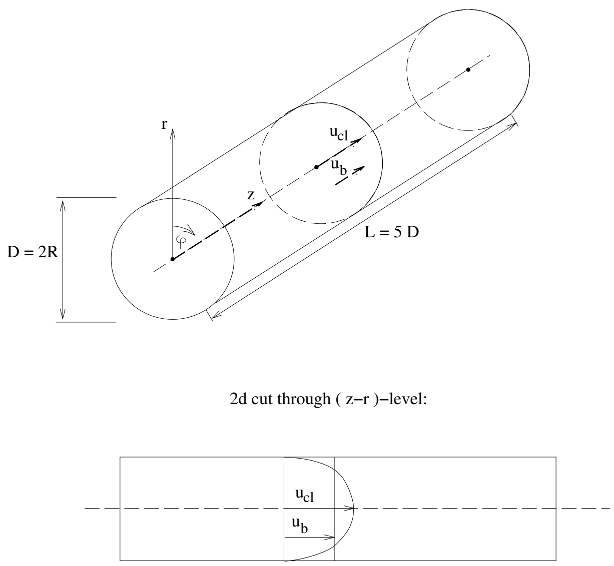

Figure 6. Pipe geometry.

samples. However, for more moderate values of T, we obtained values for u_b/u_τ that are in accordance with the results of the DNS. The same holds for $q = 4$ and $T = 10$.

Table 2
Numerical simulations: Mean flow property u_b/u_τ.

	DNS Unger	DNS Eggels	CM q=2,T=25	CM q=2,T=50	CM q=3,T=25	CM q=4,T=10
Re_{cl}	6950	6950	6950	6950	6950	6950
u_b/u_τ	14.74	14.73	14.72	14.79	14.76	14.84

For comparison purposes, we give the results from physical measurements of real turbulent pipe flows in Table 3. The figures are obtained from the literature, see [13].

Table 3
Physical experiments: Mean flow property u_b/u_τ.

	HWA Weiss	PIV Westerweel	LDA Westerweel	LDA Durst
Re_{cl}	7350	7100	7200	9600
u_b/u_τ	14.76	14.88	14.68	14.70

In Figure 7, we plot the mean velocity profile divided by the wall friction velocity u_τ in z-direction. It can be seen clearly that the profile obtained with the combination algorithm for $q = 2,\ T = 25$ and $T = 50$ coincide with the profile obtained from the DNS on the full grid. Also, the graphs for $q = 3,\ T = 25$ and $q = 4,\ T = 10$ are in really good agreement with the DNS results.

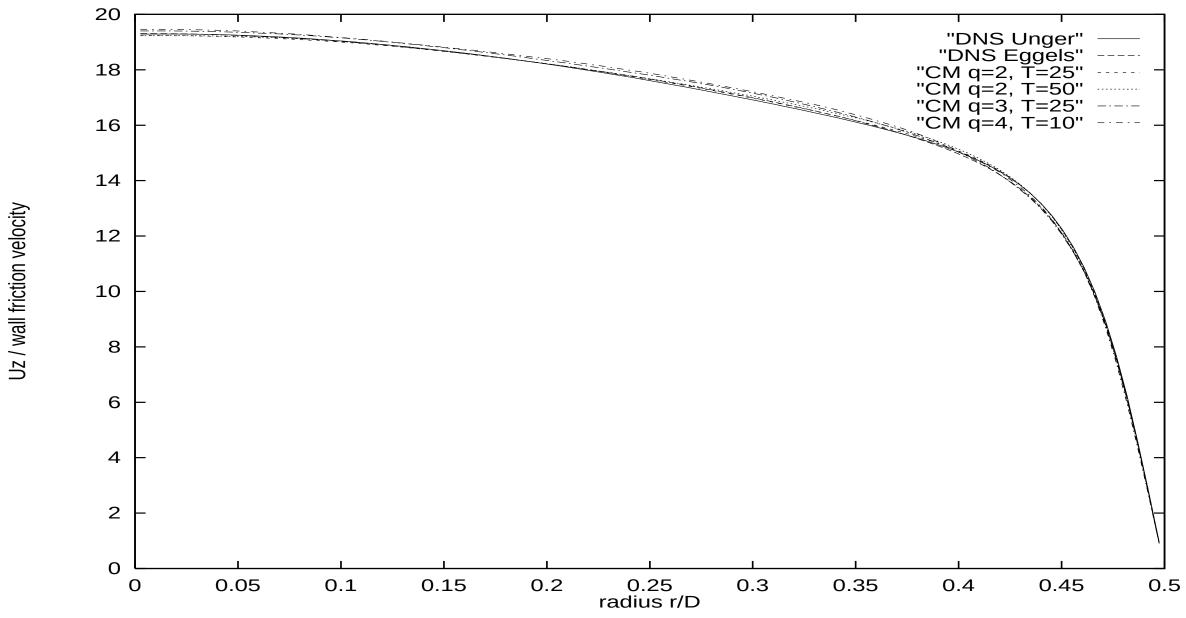

Figure 7. Numerical simulations: Axial mean velocity Uz divided by the wall friction velocity.

Figure 8 shows the mean velocity profiles divided by the wall friction velocity that had been obtained from physical measurements of real turbulent pipe flows, c.f. [13]. We see, that the numerically computed data and the measured data do not differ significantly.

Since the first statistical moments are normally not too critical for discretization errors, we also computed second moments of our simulation results. Here, we focus on the root-mean-square values (rms) of the velocity in z-direction. For the definition of rms, see for example [2]. In Figure 9, we plot the rms-profiles in axial direction divided by the wall friction velocity u_τ. The rms-profiles from [5] and [13] are equivalent. Thus, we only give the values from [13]. The rms-profiles that have been obtained with the combination method for $q = 2,\ 3,\ 4$ and $T = 25,\ 50,\ 10$ are in quite good agreement with that of the DNS. The little differences near the center of the cylinder $r = 0$ might stem from the influence of the numerical treatment of the middle axis singularity in the Navier-Stokes equations in cylindrical coordinates onto the combination method. Whether our reduced amount of gathered samples is the reason for the slight differences away from the middle axis, is an open question, yet. Figure 10 shows the rms-profiles obtained from measurements of physical experiments, c.f. [5], [13].

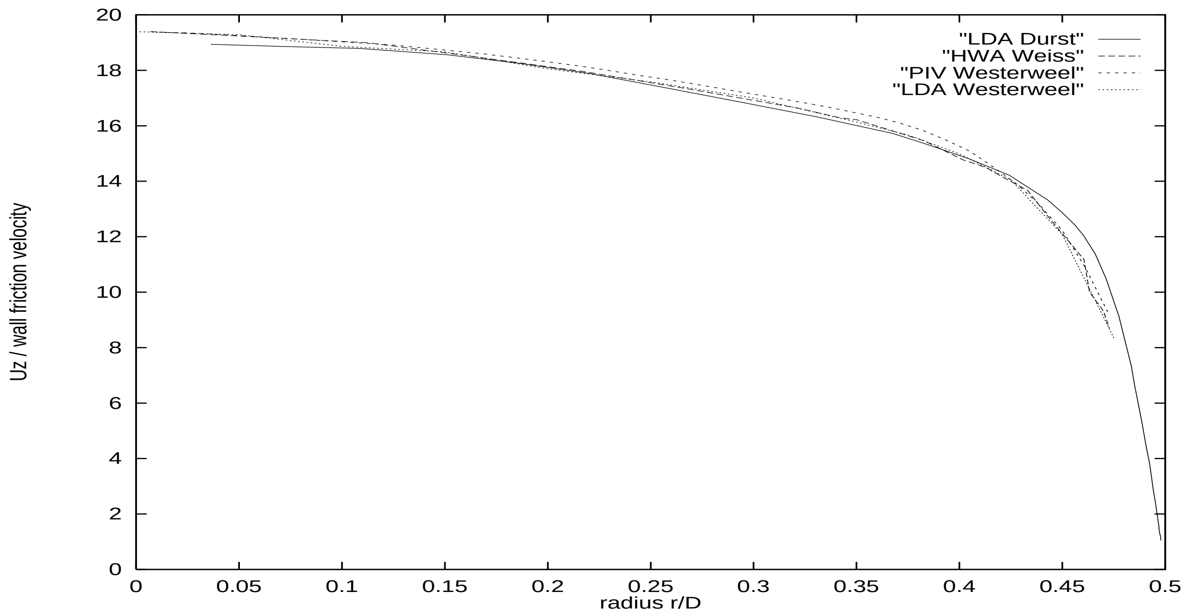

Figure 8. Physical experiments: Axial mean velocity Uz divided by the wall friction velocity.

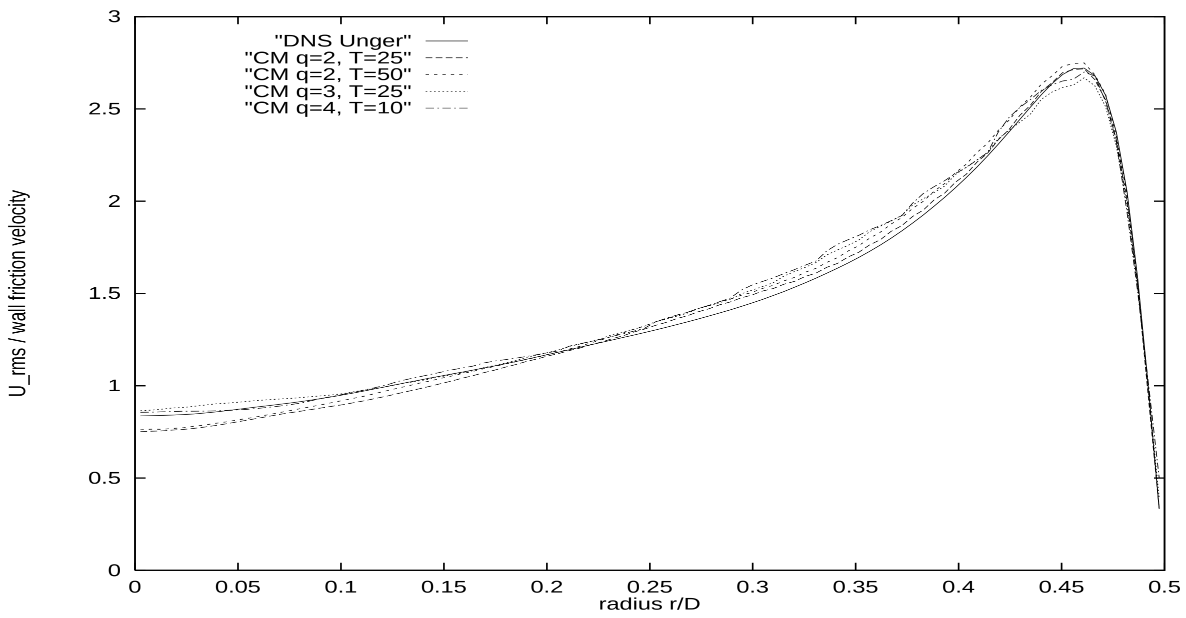

Figure 9. Numerical simulations: Axial rms-values divided by the wall friction velocity.

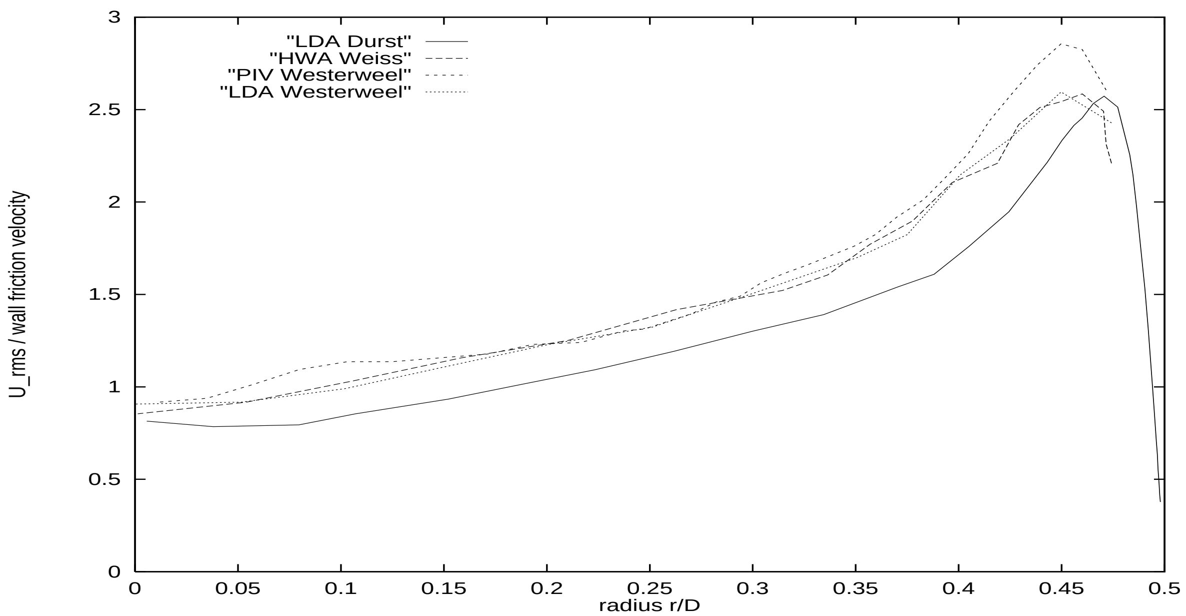

Figure 10. Physical experiments: Axial rms-values divided by the wall friction velocity.

We also tried to compute third statistical moments like the skewness. But up to now, we only gathered 20 samples, which is not sufficient for the sound evaluation of a third statistical moment. In a qualitative way, however, the present skewness results obtained from the combination method with $q = 2$ ($T = 25$, $T = 50$), $q = 3$ ($T = 25$) and $q = 4$ ($T = 10$) are in good agreement with the results given in [13].

6. Parallelization aspects

The parallelization of our combination algorithm can be achieved straightforwardly on a relative coarse grain level: The computations for the different subproblems that are associated to the grids Ω_{h_i,h_j,h_k} with $i+j+k=q+2$, $i+j+k=q+1$ and $i+j+k=q$ can be performed *fully in parallel*. Since the size of these grids and thus the amount of operations is known in advance, load balancing is simple and can be done statically.

However, in our combination algorithm, an iterate on the associated sparse grid must be assembled after T time steps, compare Figure 5. Thus, it seems, that data structures have to be set up for the sparse grid iterate (in one processor), and the data belonging to the grids Ω_{h_i,h_j,h_k} that are associated to the combination formula can be assigned to different processors. This results in a master-slave-type topology, see [6], [7]. Then, the combination of the sparse grid iterate (from the iterates on all the grids Ω_{h_i,h_j,h_k} involved in (10)) must be done in a synchronized $P:1$ communication, which might cause a bottleneck. Here, P denotes the number of processors. The same holds for the projection process to the new iterates $u_{h_i,h_j,h_k}^{t_i+T,new}$, compare Figure 5.

But the *explicit* storage and assemblage of the sparse grid iterate is not really necessary. In [8], it is described, how the explicit combination to the sparse grid and the reprojection to the different grids Ω_{h_i,h_j,h_k} with $i+j+k=q+2$, $i+j+k=q+1$ and $i+j+k=q$, can be avoided: The same effect is achieved by *directly* exchanging relevant data between the processors that store the data associated to the grids Ω_{h_i,h_j,h_k} ($i+j+k=q+2$, $i+j+k=q+1$ and $i+j+k=q$) and by updating the data on these grids. For details, see [8]. In this way, the master-slave-type topology with synchronization bottleneck can be overcome to some extent. The storage amount, necessary for the explicit assemblage of the sparse grid iterate is avoided and we obtain a parallel implementation with a distributed parallel storage concept. We implemented both parallel versions on a cluster of workstations using PVM 3.2.6.

For different values of q, Table 4 shows the fraction of the amount of memory necessary to store the data belonging to *one*, i.e. the largest grid arising in (10), versus the amount of memory necessary to store the data belonging to the full grid $\Omega_{(L,M,N)}$. This resembles the case that a sufficient amount of processors is available. Note that, since the amount of operations spent for the computations on each of the grids Ω_{h_i,h_j,h_k} with $i+j+k=q+2$, $i+j+k=q+1$ and $i+j+k=q$ is proportional to the number of its grid points, the right column of Table 4 gives more or less an impression of the run time improvement of the *parallized* combination algorithm in comparison to a sequential conventioned method operating on the full grid $\Omega_{(L,M,N)}$.

7. Concluding remarks

We presented a modification of the combination method that results, together with a given conventional Navier-Stokes solver like FLOWSI, in a solver for the Navier-Stokes equations on sparse grids. We applied this new method to the simulation of turbulent pipe flow with $Re_{cl}=6950$. It turned out that we obtained for first and second statistical moments practically the same results like for a DNS, but with far lesser computational costs and lower storage requirements.

Furthermore, our new algorithm can easily be parallelized on a relative coarse grain

Table 4
Requirements for distributed storing.

q	Number of grids involved in the combination method	storage requirements in comparison to the full grid
1	1	1
2	4	1/4
3	10	1/16
4	19	1/64
5	31	1/256
6	46	1/1024
7	64	1/4096
8	85	1/16384

level. For example, for $q = 4$ and $T = 10$, our parallel method was roughly 50 times faster than a solver on the corresponding full grid. This shows the potential and perspectives of our method for further computations with higher Reynolds numbers on finer grids that are presently not computable due to the huge memory requirements of the conventional full grid approach.

Acknowledgements: We like to thank Prof. R. Friedrich and his group, especially F. Unger, K. Wagner and E. Guentsch, for providing us with the code FLOWSI and the results of their DNS experiments.

REFERENCES

1. S. Barros, Optimierte Mehrgitterverfahren für zwei- und dreidimensionale elliptische Randwertaufgaben in Kugelkoordinaten, GMD Bericht Nr.178, R. Oldenbourg Verlag, München, Wien Nr.178, 1989.
2. P. Bradshaw, Turbulence and its Measurement, Pergamon Press, Oxford, 1971.
3. H. Bungartz, Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung, Dissertation, Institut für Informatik, TU München, 1992.
4. H. Bungartz, M. Griebel, D. Röschke, and C. Zenger, Pointwise convergence of the combination technique for Laplace's equation, in East-West Journal of Numerical Mathematics 2(1), pp. 21-45, 1994. Also available as SFB Bericht 342/16/93 A.
5. J. Eggels, F. Unger, M. Weiss, J. Westerweel, R. Adrian, R. Friedrich, and F. Nieuwstadt, Fully developed turbulent pipe flow: a comparison between direct numerical simulation and experiments, in Journal of Fluid Mechanics, 268, pp. 175-209, 1994.
6. M. Griebel, The combination technique for the sparse grid solution of PDE's on multiprocessor machines, in Parallel Proceedings Letters, 2(1), pp. 61-70, 1992. Also available as SFB Bericht 342/14/91 A.
7. M. Griebel, W. Huber, U. Rüde, and T. Störtkuhl, The combination tech-

nique for parallel sparse-grid-preconditioning and -solution of PDEs on multiprocessor machines and workstation networks, in Proceedings of the Second Joint International Conference on Vector and Parallel Processing CONPAR/VAPP V 92, 634, pp. 217-228, L. Bougé, M. Cosnard, Y. Robert, D. Trystram, ed., Springer Verlag, Berlin Heidelberg New York, 1992. Also available as SFB Bericht 342/11/92 A.
8. M. Griebel, W. Huber, T. Störtkuhl, and C. Zenger, On the parallel solution of 3D PDEs on a network of workstations and on vector computers, in Lecture Notes in Computer Science,732, pp. 276-291, Computer Architecture: Theory, Hardware, Software, Applications, A. Bode and M. Dal Cin, ed., Springer Verlag, Berlin Heidelberg New York, 1993.
9. M. Griebel, M. Schneider, and C. Zenger, A combination technique for the solution of sparse grid problems, in Proceedings of the IMACS International Symposium on Iterative Methods in Linear Algebra, pp. 263-281, P. de Groen and R. Beauwens, ed., Elsevier, Amsterdam, 1992.
10. L. Schmitt, Numerische Simulation Turbulenter Grenzschichten , Tech. Report 82/2, Lehrstuhl für Fluidmechanik, TU München, 1982.
11. ——, Grobstruktursimulation turbulenter Grenzschicht-, Kanal-, und Stufenströmungen, Dissertation, Lehrstuhl für Fluidmechanik, TU München, 1988.
12. G. Smith, Numerische Lösung von partiellen Differentialgleichungen, Vieweg Verlag, Braunschweig, 1970.
13. F. Unger, Numerische Simulation turbulenter Rohrströmungen, Dissertation, Lehrstuhl für Fluidmechanik, TU München, 1994.
14. H. Werner, Grobstruktursimulation der turbulenten Strömung über eine querliegende Rippe ein einem Plattenkanal, Dissertation, Lehrstuhl für Fluidmechanik, TU München, 1991.
15. C. Zenger, Sparse grids, in Parallel Algorithms for Partial Differential Equations, Proceedings of the Sixth GAMM-Seminar, Kiel, 1990, Notes on Numerical Fluid Mechanics, 31, W. Hackbusch, ed., Vieweg Verlag, Braunschweig, 1991.