

Iterative Solutions of the 3D Transient Navier-Stokes Equations on Parallel Computers

Wouter Couzy *

Abstract

The three-dimensional, incompressible, unsteady Navier-Stokes equations are discretized by the spectral element method based on Legendre polynomials. High-order projection methods and a subcycling method are constructed in such a way that third-order advancement in time can be obtained. The preconditioning of the pressure operator is briefly discussed. The resulting iterative spectral element solver is implemented on the Cray T3D and on the Intel Paragon. Some issues related to parallelization are addressed. Finally, we will give some results for the numerical simulation of the three-dimensional backward-facing-step flow on the Cray T3D.

Key words: Navier-Stokes equations, spectral element discretization, projection methods, iterative methods, parallelization.

AMS subject classifications: 76D05, 65M70, 65F10, 65Y05.

1 Introduction

Over the last decade, computer codes based on spectral element discretizations (see e.g. Patera [16] and Maday and Patera [13]) have become an efficient tool for the numerical simulation of fluid-flow problems (Fischer and Rønquist [6], Karniadakis et al. [9]). Numerous important developments on the theoretical, algorithmical and computational level have made that the application of spectral methods is no longer limited to regular problems in simple geometries, but is extended to complex “real-life” flow problems.

*CESAME, Université Catholique de Louvain, Av. Lemaître 4, 1348 Louvain-la-Neuve, Belgium.

ICOSAHOM'95: Proceedings of the Third International Conference on Spectral and High Order Methods. ©1996 Houston Journal of Mathematics, University of Houston.

The popularity of the spectral element method is due to the fact that it combines the practical advantages of the well-established finite element method with the “spectral” ability to reduce the number of degrees of freedom to obtain a prescribed level of accuracy. Another argument in favour of this method is that it is a domain decomposition method, yielding a natural implementation on parallel computers.

In this paper we will be concerned with the simulation of unsteady, incompressible fluid flows in three dimensions (described by the Navier-Stokes equations) on the current generation of parallel, distributed-memory computers. As an example, we will investigate the three-dimensional flow over a backward-facing step. The main numerical difficulties of such a flow lie in the computation of the recirculation zones just behind the step and the representation of the boundary layers at the sidewalls (see Figure 3 for a schematic representation of the problem). The spectral element discretization technique is very suited for this kind of fluid-flow problems. Mesh refinement at places where accurate results are essential (in the boundary layers and behind the step) is obtained by taking many, small spectral elements. Larger elements can be used at the outflow and close to the plane of symmetry. The flow is steady in the range of Reynolds number that we have simulated ($Re < 350$ for the present paper, $Re < 648$ to be obtained soon), so time-accurate solutions are not essential. Nevertheless, in order to be capable to simulate unsteady flows, high-order time-integration schemes are an issue. We will propose an implicit scheme that is third-order accurate in time. To avoid the solution of unsymmetric, indefinite systems, the nonlinear terms are discretized by an explicit scheme, based on the operator-integration-factor splitting, proposed by Maday et al. [14]. This technique can also be interpreted as a subcycling method and introduces a time-splitting error that is also of order three. The favourable stability characteristics of this scheme are investigated for different ratios of convection to diffusion.

The spectral element solver is based on the decoupling

of the pressure from the velocity field. In order to avoid additional boundary conditions for the pressure, this decoupling is performed on the space- and time-discrete equations, by either the pressure-correction method [19], or a high-order projection method, as proposed by Blair Perot [2]. The advantage of the latter method is that the resulting pressure operator is consistent and relatively cheap to evaluate. Furthermore, this projection method is third-order accurate in time.

The resulting discrete systems are symmetric, (semi-) positive definite and are solved by the conjugate gradient method. Since the operators are ill-conditioned (especially the pressure operator associated with problems at high Reynolds numbers / small time steps), efficient preconditioners are essential. According to [4], we construct a preconditioner based on the local, elemental pressure operators. This block diagonal matrix is inverted independently on each element by fast diagonalization techniques [11]. This preconditioner applies to pressure operators resulting from the pressure-correction and high-order projection methods.

In order to perform large-scale simulations at a reasonable cost, the use of parallel, distributed-memory computers can not be avoided. The performance of this generation computers has overtaken that of the fastest single-processor machines. The spectral element code has been implemented on two of the leading machines: the Cray T3D at the EPF Lausanne and the Intel Paragon at the ETH Zürich. The inter-processor communication is accomplished by the PVM paradigm (T3D and Paragon) and by the NX communication library (Paragon). Some algorithmical improvements, like a “parallel” conjugate gradient method (CGM) (see Meurant [15]) lead to a reduction of the amount of communication. It is important to study the relation between the cpu time spent in computation and in communication. The ratio of these quantities determines the parallel efficiency.

2 Discretization in space

The spectral element method (SEM) allows either to look for very accurate solutions, or, if precision is less important or unrealistic to obtain, to reduce the number of gridpoints in comparison with more classical discretization methods. Exponential convergence with respect to the degree of the polynomial expansions is obtained, provided that the solution is smooth enough. This explains that spectral methods are especially suited for problems in which high regularity is common. Another argument to use this type of methods is that numerical dissipation and dispersion

errors are almost absent. The fact that the spectral grid points are clustered at the boundaries can be considered as another advantage when trying to solve flows that are dominated by boundary-layer dynamics.

The SEM is based on the decomposition of the computational domain Ω in a number, say K , of nonoverlapping subdomains Ω_k (spectral elements). On each of these elements the solution is expanded in tensor-product based polynomials of high degree, say N , with typically $4 \leq N \leq 15$. This decomposition allows for local refinements at places where the solution is (expected to be) rapidly changing. The variational formulation provides automatically continuity of the solution at the interfaces between adjacent elements and deformed geometries can be handled without difficulties. All these factors make the SEM highly flexible with respect to geometry, accuracy and parallelization.

The Navier-Stokes equations are given on a three-dimensional domain Ω with boundary $\partial\Omega$ by

$$\begin{aligned} (1) \quad & \frac{\partial \underline{u}}{\partial t} - Re^{-1} \Delta \underline{u} + \underline{u} \cdot \nabla \underline{u} + \nabla p = \underline{b}, \\ (2) \quad & -\operatorname{div} \underline{u} = 0 \end{aligned}$$

and are subjected to appropriate initial and boundary conditions. Let us consider homogeneous Dirichlet boundary conditions, $\underline{u} = 0$ on $\partial\Omega$. Here, \underline{u} is the velocity, p is the pressure and \underline{b} is a body force. The Reynolds number $Re = UL/\nu$ is based on a characteristic velocity U , a characteristic length L and the kinematic viscosity ν . The spectral element method (see Patera [16], Maday and Patera [13], Maday et al. [12], and Rønquist [17] for more details) leads to the following space-discrete formulation:

$$\begin{aligned} (3) \quad & B \frac{\partial \underline{u}}{\partial t} + Re^{-1} A \underline{u} - D^T p + C(\underline{u}) \underline{u} = B \underline{b} \\ (4) \quad & -D \underline{u} = 0. \end{aligned}$$

Here, B is the diagonal mass matrix, A is the discrete Laplace operator, D is the discrete divergence operator, and $C(\underline{u})$ the nonlinear convection operator. The velocities and pressure, (\underline{u}, p) , are sought in $X_N \times M_N$, defined by

$$\begin{aligned} (5) \quad & X_N = \mathcal{H}_0^1(\Omega) \cap \mathcal{P}_{N,K}(\Omega)^3 \\ (6) \quad & M_N = \mathcal{L}_0^2(\Omega) \cap \mathcal{P}_{N-2,K}(\Omega), \end{aligned}$$

with $\mathcal{P}_{N,K} = \{\phi \in \mathcal{L}^2(\Omega); \phi|_{\Omega_k} \text{ is a polynomial of degree less than or equal to } N\}$ and $\mathcal{H}_0^1(\Omega)$ is the space of all square integrable functions vanishing at $\partial\Omega$ with integrable first-order derivatives. $\mathcal{L}^2(\Omega)$ is the space of all square integrable functions over Ω and \mathcal{L}_0^2 of all the functions in $\mathcal{L}^2(\Omega)$ with zero average. In fact, in each spatial direction and

on each spectral element, the velocities are discretized on an $(N+1)$ -point Gauss-Lobatto-Legendre (GLL) grid and the pressure on an $(N-1)$ -point Gauss-Legendre (GL) grid. Consequently, the operator D does not only take a derivative, but also an interpolation from the GLL to the GL grid into account.

3 Time discretization

It is our goal to construct a spectral element solver that is third-order accurate in time. Because of considerations based on the condition number of the operators (see e.g. Rønquist [17]) and on the fact that we want to avoid indefinite, nonsymmetric matrices, all terms of the Navier-Stokes equations are discretized in time by an implicit scheme, except for the nonlinear convection terms. This implicit/explicit splitting can be performed in several ways and should not degenerate the order of the time scheme. Furthermore, the explicit treatment of the nonlinear terms leads to a stability condition on the time step. Karniadakis et al. [9] proposed to extrapolate the nonlinear terms in time. This approach leads to a zero splitting error, but the region of absolute stability is rather small (see Couzy [3]). We have adopted the operator-integration-factor splitting (see Maday et al. [14]) because of its favourable stability characteristics. Timmermans [18] has also used this method in the context of spectral element discretizations. Equation (3) is written in terms of an integrating factor $\mathcal{Q}_C^{(t^*,t)}$

$$(7) \quad \frac{\partial}{\partial t} \left(\mathcal{Q}_C^{(t^*,t)} B \underline{u} \right) = \mathcal{Q}_C^{(t^*,t)} (-Re^{-1} A \underline{u} + D^T p + B \underline{b}),$$

which is defined by

$$(8) \quad \frac{\partial}{\partial t} \mathcal{Q}_C^{(t^*,t)} = -\mathcal{Q}_C^{(t^*,t)} C(\underline{U}), \quad \mathcal{Q}_C^{(t^*,t^*)} = I,$$

with I the identity matrix and t^* an arbitrary fixed time. The convection operator is linearized, with $\underline{U} = \underline{U}(t)$, determined by extrapolation of previous approximations $\underline{u}^n, \underline{u}^{n-1}, \dots, \underline{u}^{n-q}$, for some q . The operator-integration-factor splitting proceeds by discretizing (7) by an appropriate time scheme. Here, we will apply a backward differentiation formula of order s (BDFs), $s \leq 4$, with coefficients β_0, \dots, β_s , see e.g. Gear [8]. Consequently, we set $q = s - 1$. A BDFs is an A-stable method, implying that there is no stability condition on the linear (Stokes) part of the Navier-Stokes equations. Equation (7) is then discretized as

$$\beta_s B \underline{u}^{n+1} + \sum_{i=1}^s \beta_{s-i} \mathcal{Q}_C^{(t^{n+1}, t^{n+1-i})} B \underline{u}^{n+1-i} =$$

$$(9) \quad -\frac{\Delta t}{Re} A \underline{u}^{n+1} + \Delta t D^T p^{n+1} + \Delta t B \underline{b}^{n+1}$$

and the continuity equation (2) as $-D \underline{u}^{n+1} = 0$. The next step consists of the evaluation of the terms involving the operator-integration factor, which is never constructed explicitly. Instead, we define

$$(10) \quad \mathcal{Q}_C^{(t^{n+1}, t^{n+1-i})} B \underline{u}^{n+1-i} = B \tilde{\underline{u}}_i^{n+1}, \quad i = 1, \dots, s,$$

where $\tilde{\underline{u}}_i^{n+1}$ is obtained by solving the following initial value problem:

$$(11) \quad B \frac{\partial \tilde{\underline{u}}_i}{\partial t} = C(\underline{U}) \tilde{\underline{u}}_i, \quad t^{n+1-i} < t \leq t^{n+1},$$

with initial condition $\tilde{\underline{u}}_i(t^{n+1-i}) = \underline{u}^{n+1-i}$. Problem (11) is solved with a stepsize $\Delta s = \Delta t/M$; M is the number of subcycles and has an important impact on the stability of the scheme. In this way, the stepsize for the expensive implicit part is decoupled from the cheap explicit part. Roughly speaking, the stability condition for the convective part is on Δs and not on Δt .

It is important to solve the initial value problem (11) by a time-integration scheme with a large stability region along the imaginary axis. The fourth-order, explicit Runge-Kuta method is such a scheme. Combined with a BDFs method, we obtain the BDFs/RK4 scheme, the order of which is $\min\{4, s\}$. The splitting error does not vanish when the problem is steady.

The stability of the operator-integration-factor-splitting method can be investigated by applying the scalar test equation

$$(12) \quad \frac{dy}{dt} = \lambda_1 y + \lambda_2 y,$$

where λ_1 and λ_2 represent the eigenvalues spectra of the diffusion and convection operators, respectively. According to Rønquist [17], we give the following asymptotic estimates for the parameters λ_1 and λ_2 :

$$(13) \quad \lambda_1 \sim -Re^{-1} K_1^2 N^4$$

$$(14) \quad \lambda_2 \sim K_1 N^2 i,$$

with K_1 the number of spectral elements in a typical space direction. In Figure 1, the maximum allowed time step is determined for BDF3/RK4 with $\Delta s = \Delta t$ and $\Delta s = \Delta t/2$. The parameters λ_1 and λ_2 are chosen corresponding to $K_1 = N = 5$. The third-order extrapolation scheme combined with the BDF3 (see Karniadakis et al. [9]) is denoted by BDF3/EX3. It is clear that, at least for large Reynolds numbers, the stability condition is on Δs . The operator-integration-factor-splitting schemes are more stable than the one based on extrapolation. For small

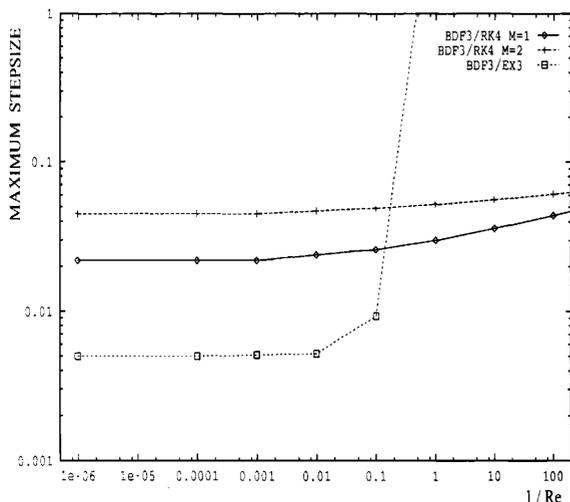


Figure 1: Log-log plot of the maximum allowed time step as a function of the Reynolds number; $\lambda_1 = -3125Re^{-1}$ and $\lambda_2 = 125i$, corresponding to $K_1 = 5$ and $N = 5$.

Reynolds numbers the influence of the implicit BDF causes the BDF3/EX3 scheme to become unconditionally stable. The effect of performing two subcycles on the stability becomes also less important. The results of this analysis have been confirmed by several numerical tests. More details can be found in Couzy [3].

4 Construction and preconditioning of the pressure operator

For reasons of convenience, we write the discrete Navier-Stokes equations in the following form

$$(15) \quad H\underline{u}^{n+1} - D^T p^{n+1} = B\underline{f}^{n+1},$$

$$(16) \quad -D\underline{u}^{n+1} = 0,$$

with

$$(17) \quad H = \frac{\beta_s}{\Delta t} B + Re^{-1} A.$$

H is the Helmholtz operator and all the explicit terms have been included in the new right-hand-side vector \underline{f} . We propose two methods to decouple the pressure from the velocity field. First, we give the pressure-correction method (see Van Kan [19]):

$$(18) \quad H\underline{u}^* = B\underline{f}^{n+1} + D^T p^n$$

$$(19) \quad S_{pc} \delta p = -D\underline{u}^*$$

$$(20) \quad \underline{u}^{n+1} = \underline{u}^* + \frac{\Delta t}{\beta_s} B^{-1} D^T \delta p.$$

Here, \underline{u}^* is an intermediate velocity field that is not necessarily divergence-free, and δp is defined by $\delta p = p^{n+1} - p^n$. The pressure operator S_{pc} is given by

$$(21) \quad S_{pc} = \frac{\Delta t}{\beta_s} D B^{-1} D^T.$$

The third step (20) can be seen as the projection of the non divergence-free \underline{u}^* on the divergence-free field \underline{u}^{n+1} . The decoupling operation introduces an error in time that reduces the order of the scheme to $\min\{2, s\}$.

Second, we propose a high-order projection method that is based on a recent paper by Blair Perot [2], following

$$(22) \quad H\underline{u}^* = B\underline{f}^{n+1}$$

$$(23) \quad S_{bp} p^{n+1} = -D\underline{u}^*$$

$$(24) \quad \underline{u}^{n+1} = \underline{u}^* + F D^T p^{n+1},$$

with

$$(25) \quad S_{bp} = D F D^T$$

and

$$(26) \quad F = \frac{\Delta t}{\beta_s} B^{-1} \left(I - \frac{\Delta t}{\beta_s Re} A B^{-1} + \left(\frac{\Delta t}{\beta_s Re} A B^{-1} \right)^2 \right).$$

Note that neither the evaluation of the pressure operator S_{pc} , nor S_{bp} requires expensive inversions (we recall that B is a diagonal matrix), as would be the case for the Uzawa method. Note also that the decoupling has been applied to the *space-discrete* operators, where boundary conditions have already been implemented. This avoids problems related to additional boundary conditions for the pressure [2]: The pressure operator is consistent. The pressure-correction method has a decoupling error proportional to $Re^{-1}(\Delta t)^2$ and the projection method to $(Re^{-1}\Delta t)^3$, which makes these methods only suited for problems with moderate to large Reynolds numbers and / or small time steps.

The Helmholtz operator H and the pressure operators S_{pc} and S_{bp} are (semi-)positive definite and symmetric. The corresponding problems are therefore solved by a preconditioned conjugate gradient method (PCGM). As a preconditioner for the Helmholtz problem, we used the diagonal of H . The preconditioning of the pressure operator is more complex. First, we remark that for $Re^{-1}\Delta t \ll 1$, the matrices S_{pc} and S_{bp} are almost identical. Their respective preconditioners can therefore be built along the same lines. We have adopted the strategy of Couzy and Deville [4], which consists of the construction of the local pressure operator on each spectral element. This elemental matrix is "identical" to the original matrix, except that it takes neither the interelemental coupling, nor the deformation into account, and can be inverted by a fast diagonalization technique (see Lynch et al. [11]).

5 Parallelization

The spectral element solver based on the aforementioned features has been implemented on the Cray T3D. The T3D is a massively parallel processor machine (MPP). This implies that the architecture is based on a high number of processors, each one of them having its own CPU, memory and cache. In the case of the T3D in Lausanne, these processors are 256 DEC chips 21064 (better known as DEC Alpha chips) with a peak performance of 150 Mflops each. They are connected by a fast, bidirectional 3D torus system interconnect network. Interprocessor communication can be managed by three so-called programming models. For our application we have chosen PVM (parallel virtual machine), because it is well-established and portable to other machines. The latter is only true to a certain extent: most MPP's, like the T3D, support a PVM "dialect", leading to non-negligible difficulties when porting a code to another computer.

Basically, parallelizing a spectral element code is the same as parallelizing a PCGM for the Helmholtz and pressure equations. When each spectral element is allocated to a processor, communication occurs only at two places (see e.g. Fischer and Patera [5]): the computation of scalar products (inherent in a PCGM) and the assembly phase (also called "direct stiffness", inherent in spectral element discretizations) that follows every operator-vector multiplication and takes care of the interface variables.

Several tests on the T3D showed that the cascade sum algorithm (see e.g. Hockney and Jesshope [7]) is the fastest way to compute scalar products: Each processor computes its own part of the product and after $2 \log(P)$ communication steps the global value is known on every processor. Two scalar products have to be computed at every iteration of the PCGM. The value of the first one is (implicitly) needed for the second one, so we speak of two synchronization points. Meurant [15] has proposed to modify the PCGM in such a way that three, instead of two, scalar products are required. They can, however, be computed at the same time. Hence, the number of synchronization points is reduced to one and better parallel efficiency can be obtained. In most practical situations, the modified PCGM shows the same convergence behaviour as the classical PCGM and is stable. In some cases, however, we encountered divergence. As an example, we mention the iterative solution of the Uzawa pressure operator [12] by two *nested* modified PCGM's. Classical PCGM converges correctly.

The parallel efficiency of the assembly phase depends highly on its implementation. Without going into the details, we confine ourselves to saying that the fastest com-

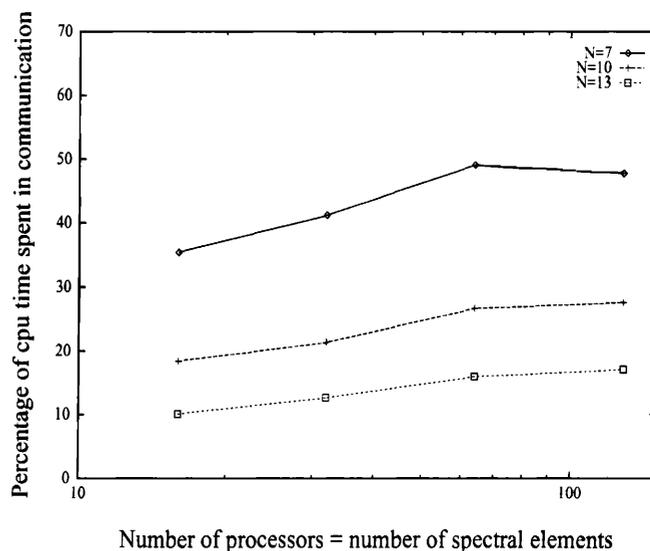


Figure 2: The percentage of cpu time spent in communication as a function of the number of spectral elements $16 \leq K = P \leq 128$, for $N = 7$, $N = 10$, and $N = 13$. Navier-Stokes flow ($Re \approx 30$) with analytical solution. Pressure correction and subcycling, BDF1/RK4, $M = 3$.

munication algorithms are obtained by avoiding blocking receives and by following the basic rules of "parallel common sense".

The parallel efficiency is investigated by comparing the cpu time spent in communication to the total cpu time. To this end, we have solved a Navier-Stokes flow ($Re \approx 30$) with an analytical solution by the pressure-correction method and subcycling (BDF1/RK4, $M = 3$) for different values of $K = P$ and N . The results are depicted in Figure 2. We see that the efficiency increases with N . This is not surprising since the number of operations on each spectral element (processor) scales like N^4 . For $N = 10$ and $N = 13$, the percentage of cpu time for communication grows slightly with K . This is mainly due to an increasing ratio between the number of interfaces (equals the number of messages during the assembly phase) and the number of spectral elements, and only partially to a degrading parallel performance. This drives us to the conclusion that we have almost obtained parallel scalability. We investigated in more detail the results for $N = 10$, $K = 128$: Only 5.6 percent of the total cpu time has been used for communicating the scalar products around the processors; 22.5 percent was taken by the assembly phase, which can be subdivided in 7.5 percent for the corners, 9.1 percent for the edges and 5.9 percent for the faces.

Besides fast communication procedures, single-processor

optimization is another issue. It is a well-known problem on modern MPP's to get high single-node performances. We have tried to solve this problem by writing all the cpu-intensive operations in BLAS, a highly optimized package of basic linear algebra subroutines. In this way, we have obtained a sustained performance (Navier-Stokes computations, $K = P = 128$) of 1.2 Gflops ($N = 7$), 2.6 Gflops ($N = 10$), up to 6 Gflops for the (unrealistic) value of $N = 20$.

The same program has also been tested on the Intel Paragon. We took the same test case that led to the results of Figure 2 and compared the performances of the two MPP's. First, we remarked that PVM on the Paragon produces disastrous results. At the time of the tests, PVM was implemented as an intermediate layer between the Fortran program and the Intel communication routines. The function calls to and fro the different layers slow down the performance of the program and lead to unacceptable cpu times. Therefore, we have resorted to NX, the Intel library of communication routines. The results are represented in Table 1. We conclude that the single-processor

	Cray T3D	Intel Paragon
N=7, K=16	25s; 35.4%	53s; 21.3%
N=10, K=16	66s; 18.4%	169s; 9.84%
N=13, K=16	148s; 10.1%	390s; 6.00%
N=7, K=32	29s; 41.1%	69s; 25.4%
N=10, K=32	85s; 21.3%	234s; 12.0%
N=13, K=32	215s; 12.6%	590s; 7.65%
N=7, K=64	33s; 49.2%	74s; 29.8%
N=10, K=64	88s; 26.6%	233s; 14.7%
N=13, K=64	207s; 15.9%	563s; 9.39%

Table 1: Cpu time in seconds (per processor), for a problem that ran on $K = P$ processors and percentage of cpu spent in communication. Ten time steps $\Delta t = 0.01$ of a Navier-Stokes flow $Re \approx 30$. Tolerance for convergence is set to 10^{-10} for the velocities and 10^{-8} for the pressure. Pressure correction method with subcycling, BDF1/RK4, $M = 3$.

performance of the T3D is considerably higher than the Paragon. The communication seems to be faster on the Paragon, but is in fact slightly slower when we take the relatively low single-node speed into account.

6 Simulation of a 3D backward-facing-step flow

The 3D flow over a backward-facing step is a good test case for various reasons. First, it exhibits several physical phenomena that are not easy to simulate, like recirculation zones just behind the step and thin boundary layers (see Figure 3 for a sketch of the basic flow properties). Another factor that complicates the numerical simulation is the large aspect ratio of the geometry (1:36). Second, experimental data by Armaly et al. [1] is available for a 1:1.94 expansion ratio. The Reynolds number is based on a characteristic velocity defined as the average inlet velocity, a characteristic length of two times the height of the inlet channel ($2 \cdot 0.5149$) and the kinematic viscosity ν . Armaly et al. predict that the flow is symmetric for all Reynolds numbers $Re < 8000$. Furthermore, away from the two sidewalls, the flow is two-dimensional for $Re < 400$ and $Re > 6600$. For $400 \leq Re \leq 6600$ three-dimensional effects have been observed. Each value of the Reynolds number is characterized by a certain length of the recirculation zone. In fact, there are three of them; The first one is located at the bottom half, directly downstream of the step. The second one was measured at the upper wall downstream of the expansion for $400 \leq Re \leq 6600$. A third recirculation zone occurs at the bottom wall, just downstream of the first one, for $1200 \leq Re \leq 2300$.

It is our final goal to simulate the backward-facing step flow at $Re = 648$. This situation is well documented in [1]. Here, we restrict ourselves to intermediate results at $Re = 172$ and $Re = 343$. The simulations have been performed by the BDF1/RK4 subcycling method with $M = 3$ and pressure correction for the first part of the transient. The discretization consists of 128 spectral elements of degree $9 \times 9 \times 9$. Once the solution becomes more or less stationary, we have changed to the third-order Blair-Perot projection method combined with BDF3/RK4 subcycling ($M = 3$). The results for $Re = 172$ are represented in Figures 4 and 5. In Figure 4 the velocity component in the flow direction is displayed together with a part of the spectral element grid. The aspect ratio of the plot has been modified to indicate more clearly the flow characteristics. Just behind the step we observe a zone with a negative velocity component; the recirculation zone. The length of this zone corresponds to the value obtained by the experiments of Armaly et al. [1], and to the two-dimensional computations of Kim and Moin [10], see Figure 6. In the corner we see a very small eddy, the existence of which is not addressed in the aforementioned paper. In Figure 5, we display the spanwise velocity component in the plane $x = 1$. We see clearly that three-dimensional phenomena

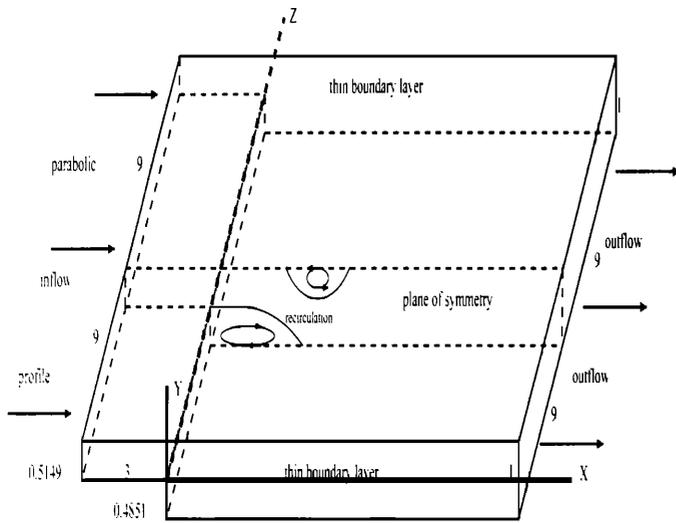


Figure 3: The geometry of the backward-facing step with a 1:1.94 expansion ratio. The inflow profile is given as the tensor product of a parabola, which is zero at the walls and one at the center, and a Blasius boundary-layer profile characterized by $\delta_{.99} = 0.50$. The size of the geometry behind the step is 19, which is large enough to ensure fully developed outflow. For the range of Reynolds numbers that we consider, two recirculation zones are of interest. Their locations are indicated in the plane of symmetry.

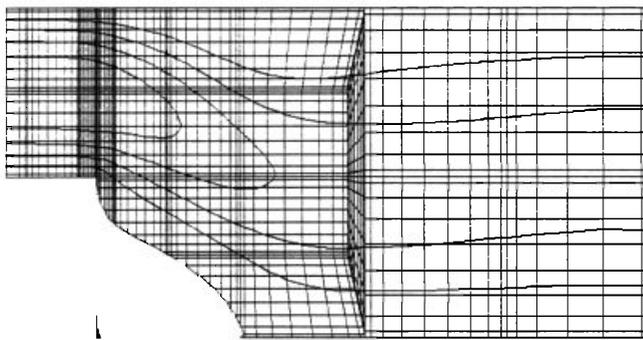


Figure 4: Contour lines of the streamwise velocity (symmetry plane). The spectral element grid is superimposed. The dashed lines just downstream of the step correspond to negative velocities. The axis in the vertical direction has been blown up by a factor 5. $Re = 172$.

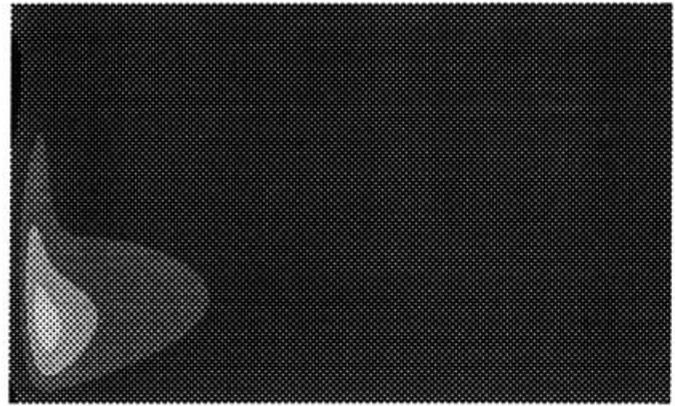


Figure 5: Spanwise velocity component in a spanwise (transversal) cut plane. $x = 1$, half-way the recirculation zone. The black zones represent negative values of the velocity; the consecutive shades of gray indicate a velocity increase of 0.02 each. The axis in the vertical direction has been blown up. $Re = 172$.

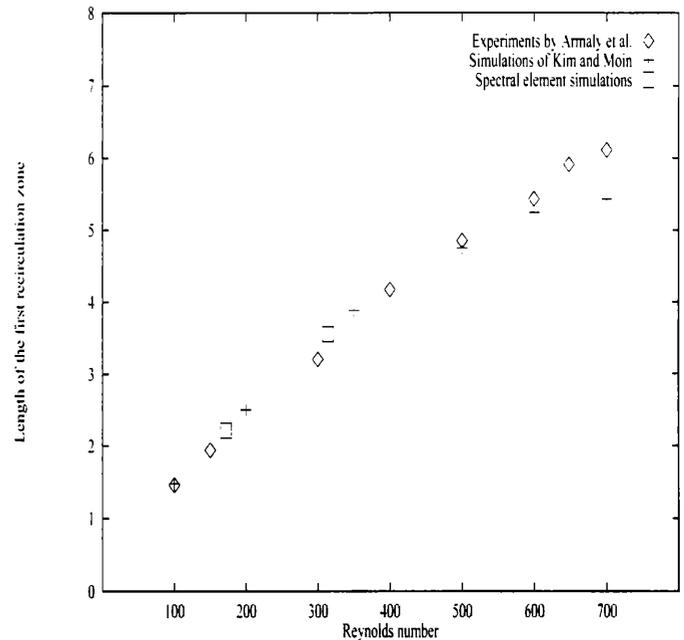


Figure 6: Comparison of the length of the first recirculation zone as a function of the Reynolds number.

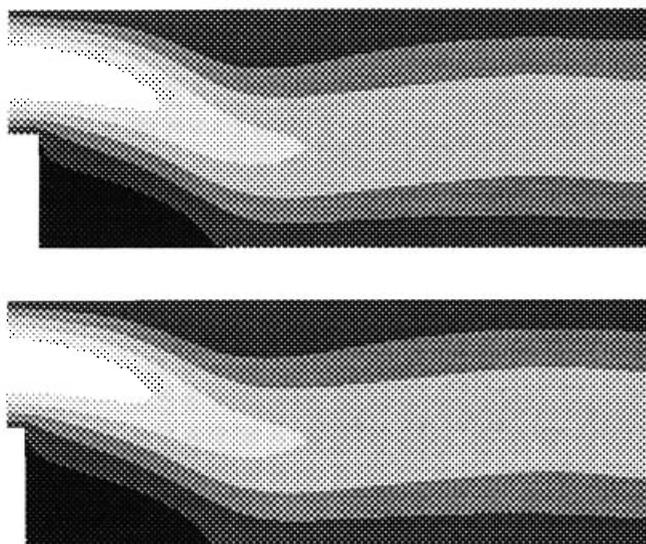


Figure 7: Streamwise velocity component. Two cross sections: At 5/18 ($z = 5$) and 9/18 ($z = 9$, plane of symmetry) of the backward-facing-step geometry. The black zones represent negative values of the velocity; the consecutive shades of gray indicate a velocity increase of 0.2 each. $Re = 343$.

are present, despite the predicted two-dimensionality by Armaly et al.

The second simulation took place at $Re = 343$. From Figure 6 we learn that the length of the separation zone still corresponds to the experimentally observed value. Figure 7 suggests that the flow remains two-dimensional: There is almost no difference between the results at the symmetry plane and the cross section at 5/18 of the geometry. We also notice that a second recirculation zone is about to appear at the upper wall. However, like for the $Re = 172$ case, three-dimensional effects are observed. They are present in almost the entire flow, but are relatively high near the boundary layer and in the recirculation zone. The three-dimensionality remains present after grid-refinement, by taking a longer inflow channel, and by changing the boundary-layer thickness.

7 Conclusions

A solution method for the three-dimensional, incompressible, unsteady Navier-Stokes has been proposed relying on a spectral element discretization in space. The time discretization and decoupling method for the pressure are such that third-order accuracy in time can be obtained. The stability has been investigated. Good parallel effi-

ciency has been demonstrated. A comparison between the Cray T3D and the Intel Paragon showed better performance for the former machine. The parallel solver has been applied to a three-dimensional flow over a backward-facing step. At $Re = 172$ and at $Re = 343$, good agreement has been obtained with experimental data, although three-dimensional phenomena have been observed. The length of the recirculation zone corresponds to the results of Armaly et al. [1]. The flow remains steady. The numerical simulations are currently extended to higher Reynolds numbers and to complex, unsteady flows.

8 Acknowledgment

The author is grateful for the advice of M.O. Deville. This paper presents research results of the Belgian incentive programme "Information Technology - Computer Science of the Future", initiated by the Belgian State, Prime Minister's Office, Science Policy Programming.

References

- [1] Armaly, B.F., Durst, F., Pereira, J.C.F., and Schönung, B. (1983). Experimental and Theoretical Investigation of Backward-Facing Step Flow. *J. Fluid Mech.*, Vol. 127, pp 473-496.
- [2] Blair Perot, J. (1993). An Analysis of the Fractional Step Method. *J. Comp. Phys.*, Vol 108, pp 51-58.
- [3] Couzy, W. (1995). *Spectral Element Discretization of the Unsteady Navier-Stokes Equations and its Iterative Solution on Parallel Computers*. PhD Thesis. Ecole Polytechnique Fédérale de Lausanne.
- [4] Couzy, W., and Deville, M.O. (1994). Iterative Solution Technique for Spectral-Element Pressure Operators at High Reynolds Numbers. *Proceedings of the Second European Computational Fluid Dynamics Conference*, (Ed. Wagner, S. et al.) Stuttgart, Germany, pp 613-618.
- [5] Fischer, P.F., and Patera, A.T. (1991). Parallel Spectral Element Solution of the Stokes Problem. *J. Comp. Phys.*, Vol. 92, pp 380-421.
- [6] Fischer, P.F., and Rønquist, E.M. (1994) Spectral Element Methods for Large Scale Parallel Navier-Stokes Calculations. *Comp. Meth. Appl. Mech. Engr.*, Vol. 116, pp 414-443.

- [7] Hockney, R.W., and Jesshope, C.R. (1981). *Parallel Computers. Architecture, Programming and Algorithms*. Adam Hilger Ltd., Bristol.
- [8] Gear, G.W. (1970). *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, New Jersey.
- [9] Karniadakis, G.E.M., Israeli, M., and Orszag, S.A. (1991). High-Order Splitting Methods for the Incompressible Navier-Stokes Equations. *J. Comp. Phys.*, Vol. 97, pp 414-443.
- [10] Kim, J., and Moin, P. (1985). Applications of a Fractional Step Method to Incompressible Navier-Stokes Equations. *J. Comp. Phys.*, Vol. 59, pp 308-323.
- [11] Lynch, R.E., Rice, J.R., and Thomas, D.H. (1964). Direct Solution of Partial Difference Equations by Tensor Product Methods. *Numerische Mathematik*, Vol. 6, pp 185-199.
- [12] Maday, Y., Meiron, D., Patera, A.T. and Rønquist, E.M. (1993). Analysis of Iterative Methods for the Steady and Unsteady Stokes Problem: Applications to Spectral Element Discretizations. *SIAM J. Sci. Comp.*, Vol. 14, No. 2, pp 310-337.
- [13] Maday, Y., and Patera, A.T. (1989). Spectral Element Methods for the Incompressible Navier-Stokes Equations. *State-of-the-Art Surveys on Computational Mechanics* (Ed. Noor, A.K. and Oden, J.T.), ASME, pp 71-143.
- [14] Maday, Y., Patera, A.T. and Rønquist, E.M. (1990). An Operator-Integration-Factor Splitting Method for Time-Dependent Problems: Application to Incompressible Fluid Flow. *J. Sci. Comp.*, Vol. 5, No. 4, pp 263-292.
- [15] Meurant, G. (1987). Multitasking the Conjugate Gradient Method on the Cray X-MP/48. *Parallel Computing*, Vol. 5, pp 267-280.
- [16] Patera, A.T. (1984). A Spectral Element Method for Fluid Dynamics; Laminar Flow in a Channel Expansion. *J. Comp. Phys.*, Vol. 54, pp 468-488.
- [17] Rønquist, E.M. (1991). *Spectral Element Methods for the Unsteady Navier-Stokes Equations*, Lecture Series 1991-01, Von Karman Institute for Fluid Dynamics, Belgium.
- [18] Timmermans, L.J.P. (1994). *Analysis of Spectral Element Methods with Application to Incompressible Flow*. PhD Thesis. Eindhoven University of Technology.
- [19] Van Kan, J. (1986). A Second-Order Accurate Pressure-Correction Scheme for Viscous Incompressible Flow. *SIAM J. Sci. Stat. Comp.*, Vol. 7, pp 870-891.

