



KUNGL  
TEKNISKA  
HÖGSKOLAN

TRITA-MEK  
Technical Report 2000:14  
ISSN 0348-467X  
ISRN KTH/MEK/TR--00/14--SE

# Direct numerical simulation of turbulent flow in plane and cylindrical geometries

Jukka Komminaho

Doctoral Thesis  
Stockholm, 2000

Royal Institute of Technology  
Department of Mechanics



Direct numerical simulation of turbulent flow in plane  
and cylindrical geometries

by

Jukka Komminaho

December 2000  
Technical Reports from  
Royal Institute of Technology  
Department of Mechanics  
S-100 44 Stockholm, Sweden

Typsatt i  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ .

Akademisk avhandling som med tillstånd av Kungliga Tekniska Högskolan i Stockholm framlägges till offentlig granskning för avläggande av teknologie doktorsexamen fredagen den 8:de december 2000 kl 10.15 i sal K2, Kungliga Tekniska Högskolan, Teknikringen 28, Stockholm.

©Jukka Komminaho 2000

Kopiecenter, Stockholm 2000

*Till Mia*



# Direct numerical simulation of turbulent flow in plane and cylindrical geometries

**Jukka Komminaho, 2000**

Department of Mechanics, KTH

SE-100 44 Stockholm, Sweden

## Abstract

This thesis deals with numerical simulation of turbulent flows in geometrically simple cases. Both plane and cylindrical geometries are used. The simplicity of the geometry allows the use of spectral methods which yield a very high accuracy using relatively few grid points. A spectral method for plane geometries is implemented on a parallel computer. The transitional Reynolds number for plane Couette flow is verified to be about 360, in accordance with earlier findings. Turbulent Couette flow at twice the transitional Reynolds number is studied and the findings of large scale structures in earlier studies of Couette flow are substantiated. These large structures are shown to be of limited extent and give an integral length scale of six half channel heights, or about eight times larger than in pressure-driven channel flow. Despite this, they contain only about 10 % of the turbulent energy. This is demonstrated by applying a very small stabilising rotation, which almost eliminates the large structures. A comparison of the Reynolds stress budget is made with a boundary layer flow, and it is shown that the near-wall values in Couette flow are comparable with high-Reynolds number boundary layer flow. A new spectrally accurate algorithm is developed and implemented for cylindrical geometries and verified by studying the evolution of eigenmodes for both pipe flow and annular pipe flow. This algorithm is a generalisation of the algorithm used in the plane channel geometry. It uses Fourier transforms in two homogeneous directions and Chebyshev polynomials in the third, wall-normal, direction. The Navier–Stokes equations are solved with a velocity-vorticity formulation, thereby avoiding the difficulty of solving for the pressure. The time advancement scheme used is a mixed implicit/explicit second order scheme. The coupling between two velocity components, arising from the cylindrical coordinates, is treated by introducing two new components and solving for them, instead of the original velocity components. The Chebyshev integration method and the Chebyshev tau method is both implemented and compared for the pipe flow case.

**Descriptors:** Turbulence, plane Couette flow, pipe flow, laminar-turbulent transition, direct numerical simulation, spectral methods.

## Preface

This thesis considers development, implementation and use of spectrally accurate algorithms for direct numerical simulation of geometrically simple flows. It is based on the following papers:

**Paper 1** KOMMINAHO, J., LUNDBLADH, A. & JOHANSSON A. V. 1996 Very large structures in plane turbulent Couette flow. *J. Fluid Mech.* **320**, 259–285.

**Paper 2** KOMMINAHO, J. & SKOTE, M. 2000 Reynolds stress budgets in Couette and boundary layer flows.

**Paper 3** KOMMINAHO, J., LUNDBLADH, A. & JOHANSSON A. V. 1997 Determination of the transitional Reynolds number in plane Couette flow through study of relaminarization. In *First AFOSR Int. Conf. on DNS/LES*. pp 233–240.

**Paper 4** KOMMINAHO, J. 1995 Numerical simulation of the Navier–Stokes equations on massively parallel computers. TRITA-MEK 1995:13.

**Paper 5** KOMMINAHO, J. & JOHANSSON A. V. 2000 Development of a spectrally accurate DNS code for cylindrical geometries.

The papers are here re-set in the present thesis-format. Some of them are published as indicated above.

# Contents

<b>Preface</b>	vi
<b>Chapter 1. Introduction</b>	1
<b>Chapter 2. Simulations of turbulent flows</b>	5
2.1. Spectral methods	5
2.1.1. Accuracy of spectral methods	7
2.1.2. Applicability of spectral methods	12
2.2. Algorithm used in the spectral code	12
<b>Chapter 3. Plane Couette flow</b>	15
3.1. Transitional Couette flow	17
3.2. Turbulent Couette flow	20
3.2.1. Large structures in Couette flow	21
3.2.2. Turbulent statistics and Reynolds stress budgets	26
<b>Chapter 4. Flows in cylindrical geometries</b>	30
4.1. Verification of pipe and annular pipe flow code	31
4.2. Future work	33
<b>Acknowledgement</b>	34
<b>Bibliography</b>	35
<b>Papers</b>	
<b>Paper 1: Very large structures in plane turbulent Couette flow</b>	43
<b>Paper 2: Reynolds stress budgets in Couette and boundary layer flows</b>	77
<b>Paper 3: Determination of the transition Reynolds number in plane Couette flow</b>	101

**Paper 4: Numerical simulation of the Navier–Stokes equations on massively para**

113

**Paper 5: Development of a spectrally accurate DNS code for cylindrical geometri**

157

## CHAPTER 1

# Introduction

Turbulence is all around us, virtually all the flows of interest are turbulent. Turbulence is the rule, not the exception, in fluid dynamics. Turbulent flows range from scales of a few  $\mu\text{m}$  in high-speed aircrafts up to astrophysical flows and interstellar turbulence where the scales are measured in parsec.

A turbulent flow differs from laminar flows, in that the mean motion of the turbulent flow does not satisfy the Navier–Stokes equations. It is a flow disordered in space and time, without the steady motion of laminar flow.

Although fluid turbulence is deterministic (the governing equations are deterministic) the evolution in time is very complicated, due to the nonlinear interactions. Thus, without precise knowledge of every fluid particle’s position and velocity, it cannot be predicted accurately for any length of time into the future. It seems impossible to consider theoretically for arbitrary times the deterministic evolution of a given turbulent field, starting with a given field of initial conditions. One has to resort to numerical methods to study the evolution of a turbulent field, in a ‘brute force’ attempt to gain insight into the properties of the turbulent flows. This approach has, in the last decades, started to gain popularity due to the significant progress in the speed and capacity of the computers, and also the development of efficient and accurate algorithms.

The development of the theory in fluid mechanics is closely coupled to experiments. It was not until the nineteenth century that mathematicians developed the fundamental equations that govern viscous flow, the Navier–Stokes equations, after the French and English researchers Claude Louis Marie Navier, (1785–1836) and Sir George Gabriel Stokes, (1819–1903).

Flows in cylindrical geometries were studied experimentally and analytically very early. Turbulence and transition in pipe flow was studied early, see [Reynolds \(1883\)](#). In this pioneering experiment he defined the non-dimensional parameter that is now called Reynolds number. He visualised the flow in the pipe with a streak of coloured water, and could thus see when the flow became turbulent. The coloured streak would break down and mix with the water some distance downstream the inlet of the pipe. He noted that the Reynolds number at which the breakdown occurred is sensitive to the disturbance levels.

In 1890 the French scientist M. [Couette](#) studied the flow between two rotating cylinders, now referred to as Taylor–Couette flow, in order to determine

the viscosity of fluids. He rotated the outer cylinder and for low rotation rates the torque is proportional to the product of the viscosity of the fluid and the rotation rate,  $\nu\Omega$ . For high enough rotation the flow becomes turbulent and the torque increases faster than linearly. A few years later Mallock (1896) designed a similar apparatus, but with a rotating inner cylinder. The observed flow was laminar only for rotation rates much lower than those used by Couette.

In 1923 Taylor performed experiments on the stability of Taylor–Couette flow. He developed a theory for calculating the stability of this flow, and also did a few numerical calculations based on that theory. The calculations, involving Bessel functions, were tedious and determining the numerical values was a major challenge. The agreement between his experiments and calculations is remarkable, and is the first example where a stability calculation matches an experiment.

Earlier attempts to calculate the stability of the flow between two moving parallel infinite planes, plane Couette flow, had been unsuccessful. The results indicated that the flow would be linearly stable for all Reynolds numbers. Romanov (1973) showed that plane Couette flow lack growing eigenmodes for all Reynolds numbers.

When the digital computer was developed in the 1930:ies and 1940:ies people began to think about simulating turbulent flows with them. Already Neumann and Emmons proposed numerical simulations of turbulence as early as 1949, but it was not until 1964 that the first simulations of turbulence were performed. This study was made by R. W. Bray at Cambridge and is described in Batchelor (1969). They used a spectral method with expansion in Fourier space, and the nonlinear terms were evaluated as convolutions. All calculations were performed on the EDSAC 2 computer with  $10 \times 10$  Fourier coefficients. In this study they developed a theory of the kinetic energy spectrum in two-dimensional turbulence and tried to verify it with numerical experiments. The resolution however was too low to draw any conclusions. In a recent study by Lindborg & Alvelius (2000) support for this theory was found, with a simulation using  $4096^2$  points.

Also Lilly (1969) made a simulation of two-dimensional turbulence, using finite differences on a  $64 \times 64$  grid. The computations was carried out on a CDC6600.

It is natural that the first simulations of turbulence concerned two-dimensional flow. The introduction of a third dimension complicates the calculations in two ways. For a two-dimensional flow one only needs one flow variable, the vorticity. In (incompressible) three-dimensional flow one needs to solve for at least two variables, with the third calculated from the continuity condition. If using primitive variables, the velocities and the pressure, one also needs to solve for the pressure, via the continuity equation. This adds to the complexity of the code. Furthermore one needs to resolve the third dimension, which dramatically increases the number of grid points needed for the calculation.

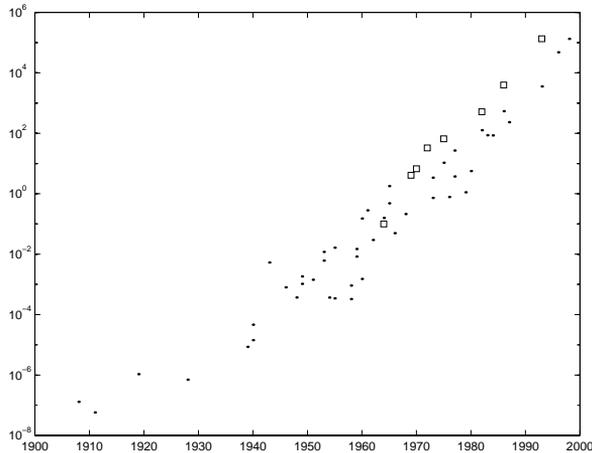


FIGURE 1.1. The amount of computer power (measured in additions per second) one can buy for a dollar as a function of time (●) (from Kurtzweil 1999), and number of thousand grid points in some simulations (□), from table 1.1.

Thus, it is not surprising that the first simulations used large-eddy simulations (LES) to reduce the number of grid points (by modelling the small scales).

Among the first simulations of three-dimensional turbulence was made by Deardorff (1970) who used a CDC6600, and the simulation code was a finite difference method with 6720 grid points. He did not fully resolve the turbulence but used a sub-grid scale model to describe the behaviour of the smallest scales. This use of sub-grid scale models made it possible to simulate turbulent channel flow at very high Reynolds number, despite the low resolution.

Schumann (1975) used LES in simulations of plane and annular flows, using 65536 grid points. In these calculations the mean-velocity profile, and second order statistics are in fair agreement with experimental data. Moin & Kim (1982) calculated the channel flow at Reynolds number 13800, using 516096 grid points, with an LES code, but unlike Schumann they did not model the wall-layer dynamics. Instead they extended the calculations down to the wall.

Thus the first simulations on three-dimensional turbulence used LES to avoid the very high demands put on computer resources needed to do a direct numerical simulation. In 1972, Orszag & Patterson (1972) performed the first direct numerical simulation of homogeneous isotropic turbulence on a  $32^3$  grid using a spectral Galerkin method.

Rogallo (1981) used spectral methods to calculate homogeneous three-dimensional turbulence subject to mean strain, examining effects of mean shear, irrotational strain and rotation on homogenous turbulence. This type of flows was also studied by Hallbäck (1993) who used DNS to obtain data for developing and testing of turbulence models.

Author	year	grid points
Bray <sup>1</sup>	1964	100
Lilly	1969	4096
Deardorff	1970	6720
Orszag & Patterson	1972	32768
Schumann	1975	65536
Moin & Kim	1982	516000
Kim <i>et al.</i>	1986	3962880
Jiménez <i>et al.</i>	1993	134217728

TABLE 1.1. Number of grid point in some simulations versus the year.

The computational capacity required for a direct numerical simulation of wall-bounded flows is so high that the first such simulation was done as late as 1987. These were studies of curved channel (Moser & Moin 1987) and plane channel (Kim *et al.* 1987). The computations of Moser & Moin (1987) were done on a CRAY-XMP, and used  $128^2 \times 65$  grid points. The simulation of Kim *et al.* (1987) used  $192 \times 160 \times 129$  grid points.

It is evident that the use of direct numerical simulations as a research tool has developed rapidly. In only three decades the number of grid points have increased by a factor of  $134 \times 10^6 / 100 = 1.34 \times 10^6$ . This development is of course closely related to the development of the computers. In figure 1.1 we have plotted the number of flops (floating point operations per second) one can buy for one USD, as a function of time. Included is also the number of grid points, measured in thousands, for various simulations. As we can see, from that figure and table 1.1 the development of simulations closely follows the development of computer power.

One may note also that the rapid development of computers has had a major impact also on experimental investigations of turbulence. This is reflected in terms of fast data acquisition and analysis, and the handling of very large data sets obtained from local time-series measurements with e.g. hot-wire anemometry or laser doppler velocimetry, and whole field measurements with e.g. particle image velocimetry.

---

<sup>1</sup>Simulation performed by Bray in 1964 and reported in Batchelor (1969).

## CHAPTER 2

# Simulations of turbulent flows

There are many different methods of solving partial differential equations numerically, such as finite elements, finite differences or spectral methods. In the finite element technique the domain is divided into small elements and a trial function is specified in each element. Finite elements have traditionally been used in structural mechanics and are, because of the localised trial function, well suited for use in complicated geometries. Finite difference methods approximate derivatives of a function by local expansion (see e.g. [Strikwerda 1989](#)). This approach is reasonable since the derivative is a local property. Spectral methods are based on global functions, and the function is expanded in a sum of basis functions, (see e.g. [Boyd 1989](#))

$$u(x) \approx \sum_{n=0}^N a_n \psi_n(x). \quad (2.1)$$

This series is substituted into the equation

$$Lu = f(x),$$

where  $L$  is an operator describing the differential equation. A residual function may be defined as

$$R(x, a_0, a_1, \dots, a_N) = Lu_N - f.$$

The goal is to minimise the residual function by choosing suitable base functions  $\psi_n$  and expansion coefficients  $a_n$ .

### 2.1. Spectral methods

There are a number of requirements on the base functions in a spectral method. They should

- be complete, meaning that any solution<sup>1</sup> can be described to arbitrary precision almost everywhere<sup>2</sup>,
- have rapid convergence<sup>3</sup>,
- be easy (fast) to compute.

---

<sup>1</sup>The solution does not have to be continuous, as long as it is continuous almost everywhere.

<sup>2</sup>Almost everywhere implies that the sequence does not have to converge pointwise.

<sup>3</sup>For rapid convergence the solution should belong to  $C^\infty(a, b)$ .

A more mathematical definition of completeness is given below.

Consider the space  $L^2(a, b)$ , i.e. the space of Lebesgue measurable functions on the interval  $[a, b]$

$$v : (a, b) \rightarrow \mathbb{C}$$

which are square integrable

$$\int_a^b |v(x)|^2 dx < \infty,$$

with pointwise operations and an inner product

$$(v_i, v_j) = \int_a^b v_i(x)v_j^*(x)dx.$$

$L^2(a, b)$  is a Hilbert space (see e.g. [Young 1990](#), for an introduction to Hilbert space). A Hilbert space is an inner product space which is a complete metric space with respect to the metric  $\|v\| = (v, v)^{1/2}$ , induced by its inner product.

To be able to expand arbitrary functions  $u \in L^2(a, b)$  as in [\(2.1\)](#) we need to find a basis in  $L^2(a, b)$ .

Define a weighted inner product

$$(v_i, v_j)_w = \int_a^b v_i(x)v_j^*(x)w(x)dx,$$

with associated norm

$$\|v_i\|_w = \left( \int |v_i(x)|^2 w(x) dx \right)^{1/2}$$

where  $w(x)$  is a weight function.

A sequence  $(v_i)_{i \in \mathbb{Z}}$  in  $L^2(a, b)$  is called an orthogonal sequence, with respect to  $w$ , if  $(v_i, v_j)_w = 0$ , for  $i \neq j$ . Normalise the sequence such that

$$(\psi_i, \psi_j)_w = \delta_{ij}, \quad \psi_i = \frac{1}{\|v_i\|} v_i.$$

The orthonormal sequence  $\psi_n$  is *complete* if the only member in  $L^2(a, b)$  which is orthogonal to every  $\psi_n$  is the zero vector. We can now write the expansion of an arbitrary function

$$u(x) = \sum_{n=0}^{\infty} (u, \psi_n)_w \psi_n.$$

Example of an orthonormal sequence in  $L^2(0, 2\pi)$  is the Fourier series

$$\psi_n(x) = \frac{1}{\sqrt{2\pi}} e^{inx}, \quad 0 < x < 2\pi.$$

The sequence  $(\psi_n)_{n=-\infty}^{\infty}$  form a complete sequence, and thus we can expand in Fourier series. The expansion coefficients  $a_n$  in [\(2.1\)](#) can be calculated from

the inner product of the function and the orthonormal basis functions

$$a_n = (u, \psi_n) = \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} u(x) e^{-inx} dx.$$

Weierstrass approximation theorem implies that any sequence of orthogonal polynomials,  $(P_n)_{n=0}^{\infty}$  (orthogonal with respect to a weighted norm), are complete in  $L^2(-1, 1)$ .

Thus, Chebyshev polynomials,  $T_i$ , on the interval  $[-1, 1]$ , with weight function  $1/\sqrt{1-x^2}$  form a complete sequence, since

$$\int_{-1}^1 \frac{T_i(x)T_j(x)}{\sqrt{1-x^2}} dx = c_{ij} \frac{\pi}{2} \quad (2.2)$$

where  $c_{ij} = 2$  for  $i = j = 0$ ,  $c_{ij} = 1$  for  $i = j \neq 0$  and  $c_{ij} = 0$  for  $i \neq j$ .

It can be shown that the  $k$ -th Fourier coefficient of a function which is infinitely differentiable and periodic with all its derivatives on  $[0, 2\pi]$  decays faster than any negative power of  $k$ . Thus Fourier expansion satisfies the two first requirements. By using a fast Fourier transform when evaluating the coefficients we also fulfil the third requirement.

Since Chebyshev polynomials  $T_n$  can be written as Fourier cosine expansion by the change of variable  $T_n(\cos \theta) \equiv \cos(n\theta)$ , we satisfy all above requirements also with Chebyshev polynomials.

### 2.1.1. Accuracy of spectral methods

One great advantage with spectral methods as compared with finite difference or finite volume is their minimal phase error. Let us analyse the differentiation error of finite difference schemes. Consider a single Fourier mode,  $f(x) = e^{ikx}$ , in one dimension. Discretize  $f$  on a domain of length  $2\pi$ , using a uniform mesh of  $N$  points. The mesh spacing is then  $h = 2\pi/N$ . The exact first derivative of  $f$  at node  $j$  is simply  $f'_j = ikf(x_j) = ik e^{ikx_j}$ . Consider the second order accurate scheme

$$\frac{\partial f}{\partial x} = \frac{f_{j+1} - f_{j-1}}{2h}. \quad (2.3)$$

For the Fourier mode this scheme yields  $f'_j = i\tilde{k}f(x_j)$  where

$$\tilde{k} = \frac{1}{h} \sin(kh),$$

is the modified wave number. The difference between  $\tilde{k}$  and  $k$  is the differentiation error as a function of the resolution of the wave. For higher order finite difference schemes this error becomes smaller (see e.g. [Kreiss & Olinger 1972](#); [Fornberg 1996](#); [Mansour et al. 1979](#)). In figure 2.1 the modified wave number is plotted for the above second order scheme, and for the fourth order schemes from [Mansour et al. \(1979\)](#) and compared with spectral method.

As we can see, a spectral method requires fewer grid points than a finite difference method. To resolve a wave with spectral methods we only need two

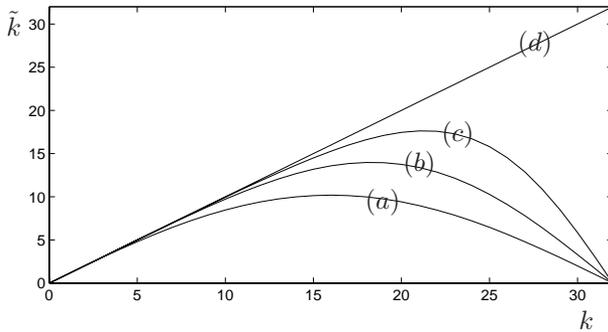


FIGURE 2.1. Comparison of modified wave numbers  $\tilde{k}$  vs  $k$  on a 64 grid. (a) Second order, (b) fourth order, (c) compact fourth order, (d) spectral.

points per wave length for a Fourier expansion (Kreiss & Oliger 1972), and  $\pi$  points per wave length for Chebyshev polynomials (Gottlieb & Orszag 1977). With second order finite difference methods we would need many more points per wave length to obtain acceptable accuracy, (see also e.g. Kreiss & Oliger 1972; Fornberg 1996; Camuto *et al.* 1988). Already fourth order FD schemes are much better in that respect. If we want a phase error,  $(k - \tilde{k})/k$ , to be smaller than 0.05 the second order FD would require about 12 points per wavelength, and the fourth order FD about 6. For a phase error of 0.01 the corresponding values would be 26 and 8 respectively.

The high phase accuracy is especially important in simulations of transition to turbulence because such simulations must follow the evolution of the flow and the nonlinear interactions between different waves for several characteristic periods. The breakdown of localized disturbances to turbulence was studied in Henningson *et al.* (1993). The disturbance energy initially grows algebraically in time and when it is strong enough for strong nonlinear effects to take place, the disturbance undergoes a rapid growth and breakdown to form a turbulent spot. Thus, to accurately follow the growth of the disturbance one needs to follow it for long times before it breaks down into a turbulent spot, and any numerical dispersion of the wave package that constitutes the initial disturbance could change the scenario.

We may illustrate the accuracy of spectral methods with an example. Solve the equation

$$\frac{d^2 u}{dx^2} - \lambda^2 u = -x, \quad u(1) = 1 + \frac{1}{\lambda^2}, \quad (2.4)$$

on the interval  $[-1, 1]$ . This type of equation is solved in the spectral DNS code used in this thesis. It arises when we Fourier transform the Navier–Stokes equations in two directions, and apply time discretization. The only remaining derivative is in the wall-normal direction, which may be evaluated

with a suitable method, e.g. finite differences or a spectral method. Typical values of the parameter  $\lambda$  is in the range 10–100.

The analytical solution may be written as

$$u(x) = e^{-\lambda} e^{\lambda x} + \frac{1}{\lambda^2} x.$$

If we now proceed to solve the equation numerically and compare it with the analytical solution, we may define an error

$$e = \left( \frac{1}{N} \sum_{i=0}^N (u(x_i)_{num} - u(x_i)_{analyt})^2 \right)$$

and compare this for different numerical methods. In figure 2.2 we compare two second order and two fourth order (central and compact) finite difference methods (FD) with two Chebyshev collocation methods.

The Chebyshev integration method (Lundbladh *et al.* 1992) and Chebyshev tau method (see e.g. Camuto *et al.* 1988) are two slightly different methods of solving the above equation. In the integration method one solves for the second derivative and integrates twice to obtain the solution. In the tau method one solves directly for the unknown function.

The integration method was originally proposed by Greengard (1988) as numerically more well-posed than the tau method. Both methods have now been implemented in the spectrally accurate boundary layer DNS code Lundbladh *et al.* (1999). Stellan Berlin (private communication) made some comparisons regarding the numerical stability of the boundary layer flow code, and showed that the tau method is more stable in most cases. This might be connected to the condition number of the resulting system matrices, which in the integration method is more ill-conditioned than in the tau method. This is discussed in section 4 of paper 5 in this thesis, where we have compared the condition numbers of the system matrices for both methods in both the pipe and channel flow codes.

In figure 2.2 the integration method seems to have a small advantage over the tau methods for low resolutions and large values of the parameter  $\lambda$ , but both Chebyshev methods show the expected spectral accuracy. For high resolutions the numerical errors in solving the system matrix is the dominant source of errors, and since the integration method matrix has a higher condition number it is also marginally less accurate compared to the tau method.

For low values of the parameter  $\lambda$  the Chebyshev methods are clearly superior to the second order FD methods for any resolution showed in the graph. For stiffer problems the Chebyshev methods are superior once the resolution is high enough.

Stretching the grid in the FD case to resolve the thin 'boundary layer' is only marginally effective to lower the error, mostly for  $\lambda = 100$ . The stretched grid is the same as in the Chebyshev methods, i.e. a cosine-distribution. The amount of stretching is not optimised to yield lowest possible errors.

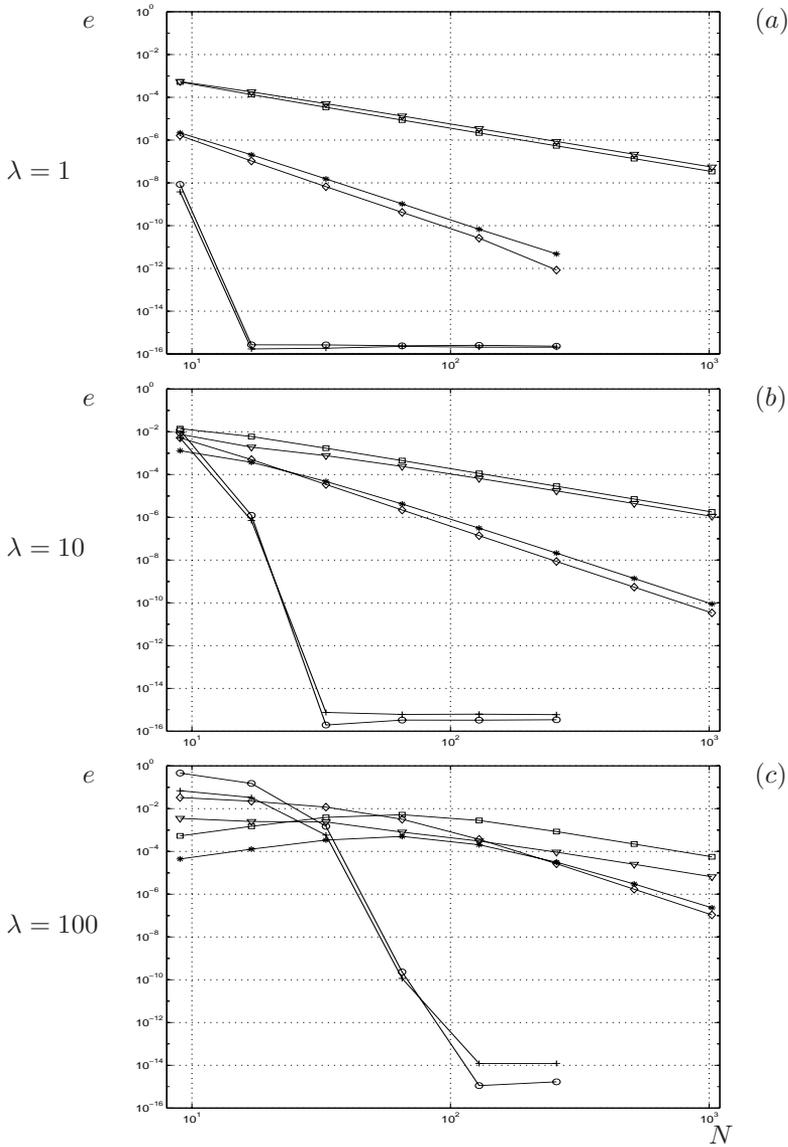


FIGURE 2.2. Comparison of error obtained with different numerical methods for (a)  $\lambda = 1$ , (b)  $\lambda = 10$  and (c)  $\lambda = 100$ . The symbols represent different numerical methods. (+) Chebyshev integration, (o) Chebyshev tau, (□) 2nd order central FD, (\*) 4th order central FD, (◇) 4th order compact FD scheme on an equidistant grid (Arnim Brüger, private communication), (∇) 2nd order central FD scheme on the same grid as the Chebyshev methods.

The fourth order central FD scheme uses a stencil which is 5 points wide. Thus it is straight-forward to implement in the interior, but near the boundaries we cannot use this stencil. Instead of using an asymmetric scheme we have here simply used the analytical value outside the boundary. In a real case it would be necessary to use at least a third order approximation at the boundary to retain the fourth order accuracy of the scheme. Thus, our implementation of the fourth order central scheme has the big advantage of being prescribed in four points at the boundaries and this becomes important for high values of  $\lambda$  and low resolutions. The fourth order central FD schemes are much more accurate than the second order scheme. For all values of  $\lambda$  it shows the expected fourth order accuracy. For a reasonable resolution the fourth order FD is an order of magnitude more accurate than the second order FD.

The fourth order compact scheme (Arnim Brüger, private communication) uses a stencil of 3 points, and can be used in all interior points. It shows the expected fourth order accuracy, and is in fact roughly a factor of two more accurate than the fourth order central scheme. The fact that the compact scheme has larger errors than the central scheme for low resolutions is related to the above mentioned treatment of the boundaries for the central scheme.

Thus, for this model problem the Chebyshev methods yield superior accuracy. For more 'real-life' problems this difference between spectral and FD may be smaller, and a higher order FD method may be comparable to spectral methods for use in DNS of turbulence. Rai & Moin (1991) investigated the accuracy of different FD schemes, and concluded that higher order upwind schemes are good candidates for direct numerical simulation of complex flows. Low order statistics were reasonably described even for second order schemes with resolution similar to spectral methods, but to obtain higher order statistics (skewness and flatness), they showed that schemes with higher (fourth or higher) order accuracy are needed.

It may be illustrative in this context to compare the statistics obtained from computations with a spectral and a second order FD scheme, by comparing the statistics from simulations using spectral methods and simulations using FD methods. Komminaho *et al.* (1996) and Bech *et al.* (1995) both simulated plane Couette flow, with spectral and FD methods respectively. The second order statistics are very similar despite a difference in Reynolds number. The third order statistics show some differences, especially near the wall. Experiments show a positive value of  $S_v$  near the wall, as should be expected. The wall-normal skewness,  $S_v$  in the spectral simulation is positive near the wall. This has also been observed in the channel flow simulation by Kim *et al.* (1987), whereas for the FD simulation the skewness is negative, and approaching zero at the wall. This difference between spectral and FD simulations using second order central difference schemes was also observed in Rai & Moin (1991).

### 2.1.2. *Applicability of spectral methods*

Spectral methods are not suitable for every problem/geometry. In fact they are suitable only to a small set of geometries. It is possible to formulate spectral methods with two inhomogenous directions, but it results in much more complexity than for one inhomogenous direction. Restricting ourselves to flows with only one inhomogenous direction, there are still a number of interesting flows that are possible to solve with spectral methods.

Perhaps the most obvious flow case is homogenous turbulence, for both two- and three-dimensional cases. Indeed, the first simulation of turbulence was made for two-dimensional turbulence with a spectral method (by Bray, reported in [Batchelor 1969](#)). Other simulations of homogenous two-dimensional turbulence include [Brachet \*et al.\* \(1988\)](#); [Lindborg & Alvelius \(2000\)](#). [Orszag & Patterson \(1972\)](#) performed the first three-dimensional simulation of homogenous turbulence, and since then it has been a popular case, ([Shumann & Patterson 1978](#); [Hallbäck 1993](#); [Jiménez \*et al.\* 1993](#), among others)

Flows with two homogenous directions that are possible to calculate with spectral methods include pressure driven plane channel ([Kim \*et al.\* 1987](#); [Moser \*et al.\* 1999](#)) and plane Couette flow ([Lee & Kim 1991](#); [Komminaho \*et al.\* 1996](#)).

Developing boundary layer flows are possible to compute, with and without pressure gradient, using a fringe region technique (see [Lundbladh \*et al.\* 1999](#), for a description of the fringe technique). The use of a fringe region is necessary due to the periodic boundary condition in the developing direction.

With a cylindrical coordinate system one can simulate pipe flow ([Zhang \*et al.\* 1994](#)), Taylor–Couette flow, curved channel flow ([Moser & Moin 1987](#)) and annular pipe flow.

## 2.2. **Algorithm used in the spectral code**

The incompressible Navier–Stokes equations can be formulated in several different ways. The one chosen here is a velocity-vorticity formulation. This appears to have been used for the first time for simulation of Navier–Stokes equations in [Dennis \*et al.\* \(1979\)](#). They used a second order finite difference method for a calculation of the unsteady flow in a cubical box with one moving wall.

The reason behind using velocity-vorticity formulation rather than the more usual (velocity and pressure) is that it is thereby possible to avoid the difficulty of solving for the pressure. Unlike the velocity, there is no evolution equation for the pressure and it is indirectly determined from the continuity equation. In incompressible flow the pressure adjusts itself to the velocity field instantaneously, implying infinite wave speed. This often causes problems, see [Canuto \*et al.\* \(1988\)](#) and e.g. [Gresho & Sani \(1987\)](#) for discussion on solution strategies for the pressure and pressure boundary condition for incompressible flow.

The incompressible Navier–Stokes equations can, in conservative form, in a rotating reference system in general coordinates, be written as

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \underbrace{\mathbf{u} \times (\nabla \times \mathbf{u} + 2\boldsymbol{\Omega})}_{=\mathbf{H}} - \nabla \left( \frac{1}{2} \mathbf{u} \cdot \mathbf{u} \right) + \frac{1}{Re} \nabla^2 \mathbf{u}, \quad (2.5a)$$

and the continuity equation for incompressible flow is given by

$$\nabla \cdot \mathbf{u} = 0, \quad (2.5b)$$

where  $\mathbf{u}$  is the velocity vector,  $\boldsymbol{\Omega}$  is the system angular rotation rate vector and  $Re$  is the Reynolds number.

We may take the divergence of the momentum equation and use the continuity equation to obtain a Poisson equation for the pressure. We apply the Laplace operator on the momentum equation, use the above equation for the pressure, to obtain a fourth order equation for the velocities with the pressure eliminated,

$$\frac{\partial \nabla^2 \mathbf{u}}{\partial t} = -\nabla (\nabla \cdot \mathbf{H}) + \nabla^2 \mathbf{H} + \frac{1}{Re} \nabla^4 \mathbf{u}. \quad (2.6)$$

The next step is to rewrite (2.6) into a system of second order equations by introducing an auxiliary vector  $\phi$ ,

$$\nabla^2 \mathbf{u} = \phi \quad (2.7a)$$

$$\frac{\partial \phi}{\partial t} = \underbrace{-\nabla (\nabla \cdot \mathbf{H}) + \nabla^2 \mathbf{H}}_{=-\nabla \times (\nabla \times \mathbf{H})} + \frac{1}{Re} \nabla^2 \phi. \quad (2.7b)$$

By taking the curl of the momentum equation we obtain a second order equation for the vorticities

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \nabla \times \mathbf{H} + \frac{1}{Re} (\nabla \times \phi). \quad (2.8)$$

Note that  $\nabla \times \phi \equiv \nabla^2 \boldsymbol{\omega}$ . We now have our velocity-vorticity formulation in a general coordinate system.

To obtain the algorithm used in the channel code we use Cartesian coordinates, and take the wall-normal components of equation (2.7a,b) and (2.8). The other velocities and vorticities are then obtained from the continuity equation and vorticity definition.

In cylindrical geometries (2.7a,b) constitutes a coupled system of equations and we need to use two components of each of these equations (the radial and azimuthal ones). By a substitution of the  $\phi$ -components into

$$\begin{aligned} \phi_A &= \phi_1 + i\phi_2 \\ \phi_B &= \phi_1 - i\phi_2 \end{aligned}$$

and similarly for the velocities and vorticities we can obtain an uncoupled set of equations. The treatment of the boundary conditions is here somewhat complex and requires the use also of the axial component of the vorticity equation (2.8).

It is possible to obtain essentially the same algorithm as for the cylindrical geometry in general coordinates (Arne Johansson, private communication).

## CHAPTER 3

### Plane Couette flow

Plane Couette flow is the flow between two infinitely large parallel planes, see figure 3.1. This can be seen as a limiting case of cylindrical Couette flow, where the parameter  $\gamma = 1 - r_o/r_i \rightarrow 0$ . It is one of the canonical flow cases. It has a monotonic velocity profile (see figure 3.1), both in the laminar and turbulent cases. This monotonic velocity profile means that when applying a system rotation (with the rotation vector aligned with the  $z$ -axis) to a plane Couette flow, both sides are stabilised or destabilised, depending on the sign of the rotation. This is in contrast to the pressure driven channel flow, where one side is stabilised and the other side is destabilised.

We write the Navier–Stokes equation (2.5a) in Cartesian coordinates in a rotating reference frame as,

$$\frac{\partial u'_i}{\partial t} + \frac{\partial}{\partial x_j}(u'_i u'_j) = -\frac{1}{\rho} \frac{\partial p'}{\partial x_i} + \nu \frac{\partial^2 u'_i}{\partial x_j \partial x_j} + 2\epsilon_{ijk} u'_j \Omega_k \quad (3.1a)$$

$$\frac{\partial u'_i}{\partial x_i} = 0. \quad (3.1b)$$

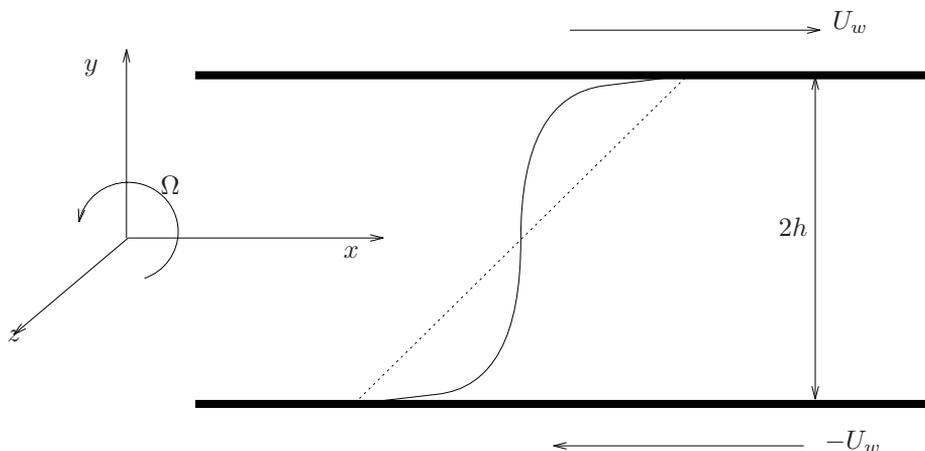


FIGURE 3.1. The flow geometry in plane Couette flow: (—), turbulent mean velocity profile, (···), laminar mean velocity profile.

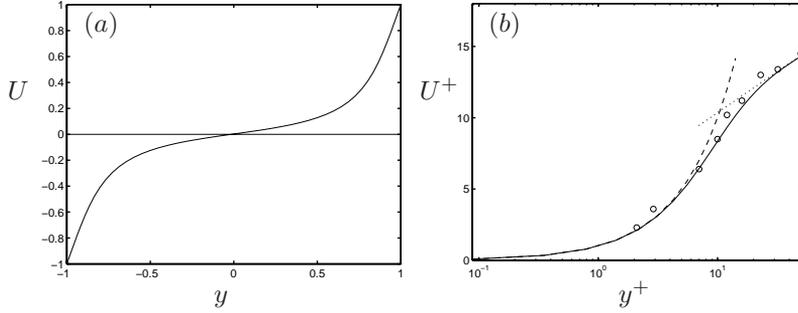


FIGURE 3.2. The mean velocity profile (a) in outer, and (b) in inner variables; (o) are from the experiments by [Bech \*et al.\* \(1995\)](#). Dashed curve:  $U^+ = y^+$ . Dotted line:  $U^+ = (1/0.4) \ln y^+ + 4.6$ .

The effect of the system rotation can be seen as a volume force in the fluid, also known as the Coriolis force and the centrifugal force. The Coriolis force is the last term in the momentum equation, (3.1a) and the centrifugal force has been included in the pressure.

Divide the flow into a mean and a fluctuating part,  $u' = U + u$ , where the mean part is defined as an ensemble average over  $N$  different times, and also an average over the homogeneous directions ( $x$  and  $z$  in the Couette flow and  $z$  in the boundary layer)

$$\bar{u}' \equiv U(y, t) = \frac{1}{NL_x L_z} \sum_{i=1}^N \int_0^{L_x} \int_0^{L_z} u'(x, y, z, t) dx dz. \quad (3.2)$$

The Reynolds equation for the mean flow is now obtained as

$$\frac{\partial U_i}{\partial t} + \frac{\partial}{\partial x_j} (U_i U_j) = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} - \frac{\partial}{\partial x_j} R_{ij} + \epsilon_{ijk} U_j \Omega_k \quad (3.3)$$

where  $R_{ij} = \overline{u_i u_j}$  is the velocity correlation tensor, and will here be referred to as the 'Reynolds stress tensor'.

A fully developed plane Couette flow has a constant total shear stress across the entire channel (see figure 3.3). This is easily obtained by integrating the streamwise component of (3.3)

$$\nu \frac{dU}{dy} - R_{ij} = \nu \frac{dU}{dy} \Big|_{wall} + \frac{1}{\rho} \frac{dP}{dx} (y - y_{wall}). \quad (3.4)$$

In Couette flow the pressure gradient  $dP/dx$  is zero. Now use the wall friction velocity,

$$u_\tau = \sqrt{\nu \left| \frac{dU}{dy} \right|_{wall}}, \quad (3.5)$$

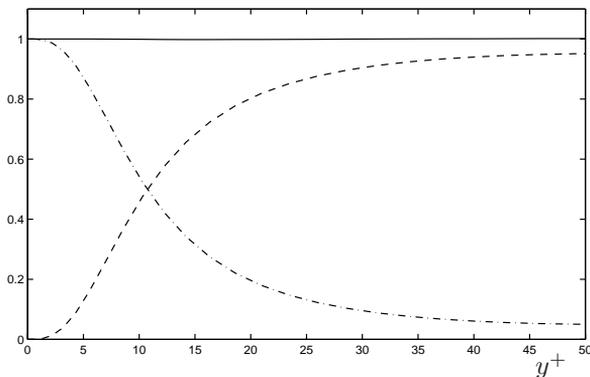


FIGURE 3.3. Total shear stress (—), viscous shear stress,  $\frac{dU}{dy}$  (---), turbulent shear stress,  $-R_{12}$  (- · -), all normalised by  $u_\tau^2$ .

to define the non-dimensional velocity,  $u_i^+ = u_i/u_\tau$ , and length  $x_i^+ = x_i/l_*$  where  $l_* = \nu/u_\tau$ . The above total shear stress (3.4) can be expressed in normalised form as

$$\frac{dU^+}{dy^+} - R_{12}^+ = 1. \quad (3.6)$$

The existence of a non-zero mean shear rate, and associated turbulence production at the centre of the channel, gives a significantly different character to the flow in this region, as compared to pressure-driven channel flow.

In comparison with zero-pressure-gradient boundary layers and pressure-driven channel flow, plane Couette flow has the unique feature of combining the parallel flow property with a zero pressure gradient. This suggests that one would expect similarities with boundary layer flow, but with a significantly simpler mean flow equation.

### 3.1. Transitional Couette flow

Analysing flow stability is difficult because of the non-linear nature of the Navier–Stokes equations. A standard approach is to analyse the stability of linearised Navier–Stokes equations subject to infinitesimal disturbances. The linear equation has limited validity, but as long as the disturbance is small compared to the base flow, the non-linear effects are negligible. By assuming disturbances of the form  $u(x, y, z, t) = \hat{u}(y)e^{i(\alpha x + \beta z - \omega t)}$  one may derive (see e.g. [Hallbäck et al. 1996](#)) the Orr–Sommerfeld equation

$$\left[ (U - c)(D^2 - k^2) - U'' - \frac{1}{i\alpha R}(D^2 - k^2)^2 \right] \hat{u} = 0,$$

where  $\omega = \alpha c$  is the frequency of the disturbance,  $D$  denotes derivative in the wall normal direction,  $k^2 = \alpha^2 + \beta^2$ ,  $\alpha$  denotes the streamwise wave number,

and  $\beta$  is the spanwise wave number. If there exist wavenumbers for which  $Im(\omega) > 0$  the flow is linearly unstable.

The critical Reynolds number,  $Re_c$ , is defined as the lowest value of  $Re$  at which there is any solution of the Orr-Sommerfeld equation with  $Im(\omega) = 0$ . For  $Re > Re_c$  linearly unstable solutions of the Orr-Sommerfeld equation may exist.

According to linear stability theory the plane Couette flow is stable (Drazin & Reid 1981) for all Reynolds numbers, i.e.  $Re_c = \infty$ . But from experiments we know that the flow becomes turbulent for sufficiently high Reynolds numbers.

The first experimental investigations of plane Couette flow was made by Reichardt (1956) in a running belt apparatus with both belts moving. In this study he used oil and water as fluids, and reported measurements of mean velocity profiles. He also tried to determine the transitional Reynolds number, i.e. the lowest Reynolds number for which turbulence is sustained. He arrived at a value of 750.

Numerical results by Orszag & Kells (1980) demonstrated that three-dimensional disturbances can drive the transition to turbulence. Nagata (1990) found three-dimensional finite-amplitude solutions in plane Couette flow.

Later studies of transition include Leutheusser & Chu (1971). They carried out the transition experiments in a facility where a free surface water flow was used as the moving wall. The other wall was a stationary flat plate located above and parallel with the water surface, and air was drawn into the channel formed between the water surface and the stationary plate. The spanwise aspect ratio was 12 and the water surface probably rough, as they state that the water flow was turbulent. They determined the transitional Reynolds number to 280, but is probably too low, since it is most possibly affected by the roughness of the water surface.

One way to determine the transitional Reynolds number is to introduce some kind of disturbance in the laminar flow and study the evolution of the disturbance. If it grows in time and develops into a turbulent spot, then the Reynolds number is higher than the transitional  $Re$ , otherwise it is lower.

In the direct numerical simulation by Lundbladh & Johansson (1991) of turbulent spot formation in plane Couette flow it was shown that if a localized disturbance with high enough amplitude was introduced it grew if the Reynolds number was 375 or higher. For a Reynolds number of 350 the disturbance eventually died out and no turbulent spot was formed, and therefore they concluded that the transitional Reynolds number is between 350 and 375. This is in agreement with the experimental study of turbulent spots in plane Couette flow by Tillmark & Alfredsson (1992) where they found the transitional  $Re$  to be  $360 \pm 10$ .

One could perhaps raise the question whether there exists other more 'optimal' disturbances for which transition occurs at lower Reynolds number. In order to avoid this difficulty one can start with a turbulent state which contains

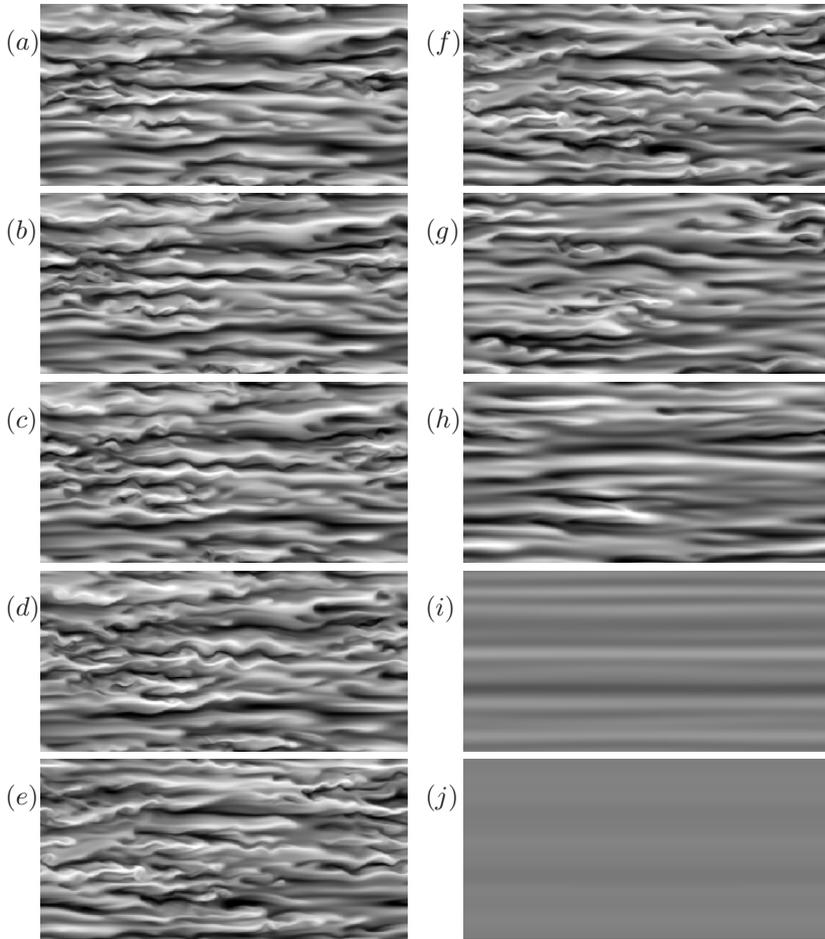


FIGURE 3.4. The instantaneous streamwise velocity field in an  $x$ - $z$  plane at  $y = 0$ , and at different times. in the large box. (a)  $T = 673.8$ , (b)  $T = 681.1$ , (c)  $T = 687.4$ , (d)  $T = 694.8$ , (e)  $T = 700.9$ , (f)  $T = 758.7$ , (g)  $T = 829.2$ , (h)  $T = 913.3$ , (i)  $T = 1075.1$ , and (j)  $T = 1259.9$ . In (a)–(e) the Reynolds number is 375 and in (f)–(j) 350.

many possible disturbances and then successively lower the Reynolds number until the flow relaminarizes. Both this and the above approach has been adapted by [Tillmark & Alfredsson \(1992\)](#) and both yielded the same transitional  $Re$ .

In the study by [Dauchot & Daviaud \(1995\)](#) they studied the transitional Reynolds number in plane Couette flow by studying the growth of turbulent spots. They introduced strong disturbances and investigated the dependence

of the growth on the amplitude of the disturbance. Their disturbance was in the form of a jet, with 1mm in diameter, introduced perpendicularly from one wall into the channel and out through the other wall. They introduced a critical amplitude,  $A_c = f(Re) = v/U_{wall}$  ( $v$  is the velocity of the jet), that was a function of the Reynolds number. Disturbances with amplitudes below  $A_c$  did not develop into turbulent spots, whereas disturbances with amplitude above  $A_c$  did. They found that for high enough disturbances they could produce turbulent spots for a Reynolds number of 335. In the same experimental apparatus [Daviaud \*et al.\* \(1992\)](#) found the transitional Reynolds number to be  $370 \pm 10$ . This is in agreement with the  $A_c$  curve reported in [Dauchot & Daviaud](#) the authors claim, since [Daviaud \*et al.\*](#) used a  $A_c = 8.5$ , and they also find  $Re_{trans} = 370$  for  $A_c = 8.5$ .

Another approach to determine the transitional Reynolds number was taken in [Komminaho \*et al.\* \(1997\)](#). The idea was to begin with a turbulent state and then successively lower the Reynolds number (by increasing the viscosity) until the flow relamenarized. In a sufficiently large computational domain the turbulence would be in 'a natural' state, and would therefore contain the least stable modes. This regardless of the way the turbulence started. Thereby it was hoped that the somewhat contradictory results of [Lundbladh & Johansson \(1991\)](#); [Tillmark & Alfredsson \(1992\)](#) and [Dauchot & Daviaud \(1995\)](#) would be clarified. The findings supported the  $Re_{trans}$  of about 360 reported by [Lundbladh & Johansson \(1991\)](#); [Tillmark & Alfredsson \(1992\)](#). The simulation started with a Reynolds number of 500, and was lowered in steps down to 375, where the turbulence was sustained for long times. When lowering the  $Re$  further to 350, the turbulence vanished rather quickly, see figure 3.4 where the process is depicted. The streamwise velocity at the centreplane is shown for different times, (a)–(e) for  $Re = 375$  and (f)–(j) for  $Re = 350$ .

### 3.2. Turbulent Couette flow

Fully developed turbulent plane Couette flow is difficult to study experimentally. If one uses a one-moving-wall setup, the channel needs to be very long to produce a fully developed flow, and one also needs to apply some pressure gradient to obtain the correct mass flux. For a two-moving-wall setup one has difficulty of measuring, since disturbances introduced by the measuring devices are not convected out from the measuring volume.

The first experiments on turbulent Couette flow appears to be made by [Reichardt \(1956, 1959\)](#), in his measurements of the mean velocity profile. He confirmed experimentally that the mean velocity profile has an S-shaped profile, and that it is quite linear in the central region. Later [Robertson & Johnson \(1970\)](#) measured both mean velocity and turbulent statistics using hot-wire anemometry. They also measured two-point velocity correlations.

More recent experimental studies include [Aydin & Leutheusser \(1991\)](#) and [Tillmark & Alfredsson \(1992, 1994\)](#).

In later years direct numerical simulations of plane Couette flow have been successfully carried out, and several investigations have been carried out.

### 3.2.1. Large structures in Couette flow

Large scale structures in the core region of the flow have been observed in plane Couette flow in several studies. It appears that the first to note the large scale structures as a peculiarity in plane Couette flow, were [Lee & Kim \(1991\)](#), although [Miyake \*et al.\* \(1987\)](#) made a brief comment on them. Since then these large structures have been observed in several direct numerical simulations, [Kristoffersen \*et al.\* \(1993\)](#); [Bech & Andersson \(1994\)](#); [Komminaho \*et al.\* \(1996\)](#), and experimentally in [Tillmark & Alfredsson \(1994\)](#); [Tillmark \(1995\)](#). For the different parameters of the simulations, see table 3.1. Ongoing activities with DNS of plane Couette flow are found also at Science Univ. of Tokyo (Kawamura, private communication) and Univ. of Southampton (Hu, private communication).

Authors	$Re$	$Re_\tau$	$L_x$	$L_z$	$\Delta x^+$	$\Delta z^+$
<a href="#">Miyake <i>et al.</i> (1987)</a>	1250	62	64	16	61.6	15.4
<a href="#">Lee &amp; Kim (1991)</a>	3000	170	$4\pi$	$\frac{8}{3}\pi$	11.1	4.95
<a href="#">Papavassiliou &amp; Hanratty (1997)</a>	2660	157	$4\pi$	$2\pi$	15.4	7.7
<a href="#">Kristoffersen <i>et al.</i> (1993)</a>	1300	83.2	$10\pi$	$4\pi$	10.9	8.2
	1300	83.2	$4\pi$	$2\pi$	10.9	8.2
<a href="#">Bech &amp; Andersson (1994)</a>	1300	82.6	$16\pi$	$2\pi$	10.8	8.1
	1300	82.2	$10\pi$	$4\pi$	10.1	4.0
	750	52.9	8	4	8.8	4.4
<a href="#">Komminaho <i>et al.</i> (1996)</a>	750	51.9	$10\pi$	$4\pi$	9.0	5.1
	750	52.0	$28\pi$	$8\pi$	8.9	5.1

TABLE 3.1. Some details for different Couette flow simulations. The Reynolds numbers are based on half channel height and half the relative wall velocity.  $L_x$  and  $L_z$  are the dimensions of the computational domain, measured in half channel height  $h$ , in  $x$  and  $z$  directions. The resolution (expressed in wall units) have been evaluated in physical space for all simulations, to make it possible to compare finite difference and spectral methods (when using dealiasing in spectral methods, the number of points are different in physical and Fourier space).

These very large structures in Couette flow appear to have no counterpart in other flows. Figure 3.5, which is taken from the simulations by [Komminaho \*et al.\* \(1996\)](#), shows the streamwise velocity in the centre plane of the channel. The large structures can clearly be seen as the long elongated streaks.

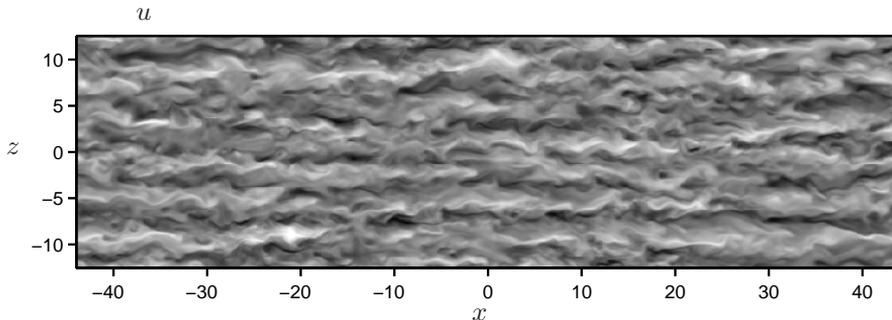


FIGURE 3.5. Instantaneous velocity fields at  $T = 2020$  in an  $(x, z)$  plane at  $y = 0$ . Streamwise velocity range  $-0.5$  to  $0.5$ .

Define the twopoint correlation as

$$R_{uu}(x) = \frac{\overline{u(x')u(x'+x)}}{\overline{u(x')^2}}$$

where only a streamwise separation  $x$  is considered, and an overbar denotes averaging over the  $x$ - $z$  plane in question, as well as over a period of time,  $T$ .

The correlation  $R_{uu}$  reveals the existence of the large structures, and is substantially larger than in channel flow and pipe flow. In figure 3.6 we show the two-point correlation from several different simulations of Couette flow as well as some experiments. Define the integral length scale as

$$\Lambda_{uu} = \int_0^{L_x/2} R_{uu}(x) dx. \quad (3.7)$$

In the channel flow simulation by Kim *et al.* (1987) it is about  $0.8h$ , and in the square duct flow by Gavrilakis (1992) it is about  $1.6h$ . In the present Couette flow it is  $6.1h$ .

The experiments by Tillmark & Alfredsson (1994) were carried out with a high degree of control of all experimental conditions. They used LDV measurements in their Couette flow apparatus with two moving walls for a Reynolds number of 1260. Robertson & Johnson (1970) used one moving wall and one stationary, and measured with hot-wires. Their measurements were carried out at  $Re = 11800$  and show much lower value of  $R_{uu}$ . This difference may be an effect of the difference of the experimental setup. In the two-moving wall the flow is stationary in the middle between the walls, whereas in the one-moving wall setup the flow in the middle of the channel is convected with half the wall speed. This means the one-moving wall apparatus must be sufficiently long to achieve fully developed flow.

In the simulations by Lee & Kim (1991) they reported that the very large scale structures contributed to about 30% of the turbulent kinetic energy, and that they are persistent counter-rotating streamwise vortices, with a spanwise spacing of about  $2h$ . They tried several different computational domains, the

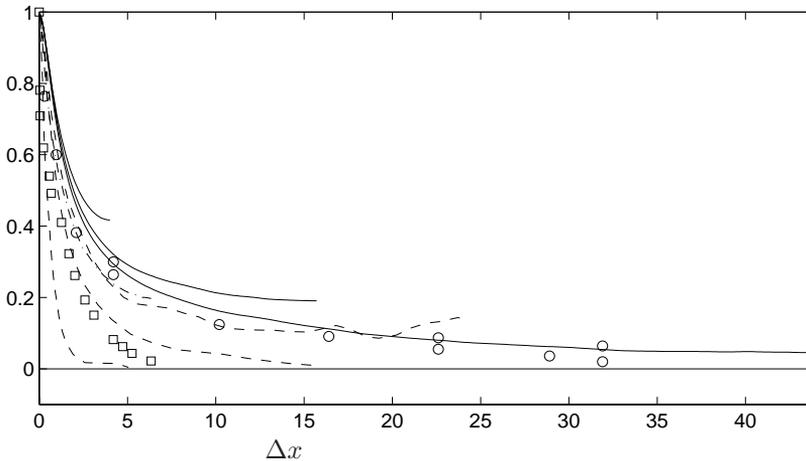


FIGURE 3.6. The two-point correlation for the streamwise velocity,  $R_{uu}$  at  $y = 0$  vs streamwise separation, from different experiments and simulations Komminaho *et al.* (1996) (—) (three different box sizes), Bech & Andersson (1994) (---) (three different box sizes), Papavassiliou & Hanratty (1997) (- · -), Symbols are from measurements by Bech *et al.* (1995) (o) and Robertson & Johnson (1970) (□).

largest being  $1024\pi$  in the streamwise direction and the large scales appeared to still be longer than the box used. They drew the conclusion that these structures are essentially infinitely long. However, the spatial resolution was very low for the large box sizes. In the simulations by Kristoffersen *et al.* (1993) they found similar structures, but much weaker and not stationary, as in the Lee & Kim simulation, neither in location nor in shape.

It seems quite possible that the differences in the large scales between the two simulations are connected to the size of the computational domain (and resolution, etc). In Lee & Kim the spanwise width ( $8/3\pi$ ) allows two pairs of the vortices and in Kristoffersen *et al.* the spanwise width ( $2\pi$ ) allows only one and a half pair, i.e. three vortices.

In Bech & Andersson (1994) they tried three different sizes of the computational domain, and found the large structures in only one of the domains,  $16\pi \times 2\pi$ , whereas in the other two the flow showed much lower correlations.

Papavassiliou & Hanratty (1997) used a pseudospectral code in their simulation of plane Couette flow. Their computational domain was  $4\pi \times 2\pi$  and the Reynolds number 2660. The resolution of their simulation is probably too low in the wall-normal direction with only 65 Chebyshev modes. They present the total shear, normalised with  $u_\tau$ , which should be constant and equal to one, but has a small bump in the near-wall region. They also fail to capture the right value of the peak of the turbulent energy production,  $\mathcal{P}^+$ , which in

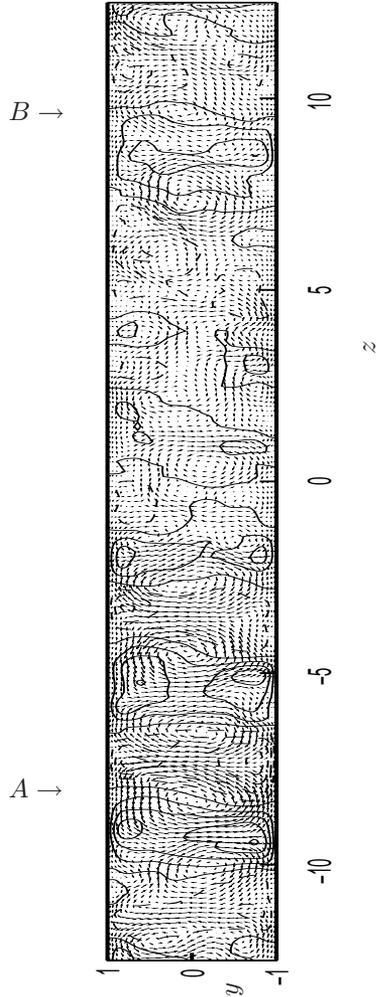


FIGURE 3.7. Velocity field, averaged over the  $x$ -axis, viewed in a  $(y,z)$ -plane. The solid contours denote  $u \geq 0$ , and dashed contours denote  $u < 0$ . The contour increment is 0.05. The  $v$  and  $w$  velocities are superimposed as vectors. For clarity every second point in the  $y$ -direction has been omitted,  $T = 2020$ . Note that the scales are not the same in  $y$  and  $z$  direction

Couette flow should be 0.25, (Komminaho *et al.* 1996). Their correlation curve is comparable to the  $16\pi \times 2\pi$  simulation by Bech & Andersson (1994).

In Komminaho *et al.* (1996) we used three different sizes of the computational domain, to investigate the influence of the size of the domain on the flow field. As can be seen in figure 3.6 there is a large influence on the large

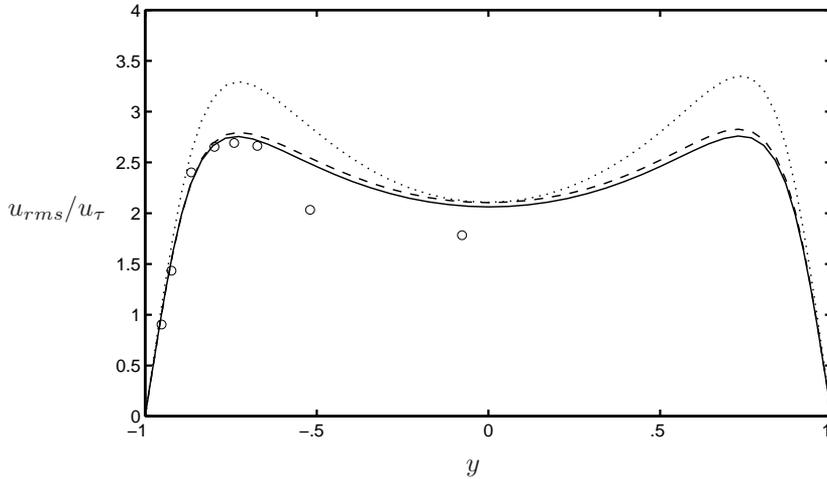


FIGURE 3.8. The  $u_{rms}/u_\tau$  values for the different box sizes as a function of  $y$ :  $28\pi \times 2 \times 8\pi$  (—),  $10\pi \times 2 \times 4\pi$  (---), and  $8 \times 2 \times 4$  ( $\cdots$ ), ( $\circ$ ) experiments by [Bech et al. \(1995\)](#)

structures, which is reflected in the correlation curve. In a narrow box one can get an artificial enhancement of the large structures, because of the periodic boundary condition. In the smallest box there was room for one pair of the structures and they became very strong. In the largest box there was room for about six pairs of the vortical structures. It is evident that the correlations are overestimated in the smaller boxes. In the largest box the correlation curve agrees well with the experimentally measured correlation by Tillmark from [Bech et al. \(1995\)](#) and it was concluded that the length was sufficient to contain the largest scales.

It is interesting to note that the second largest box, which is as large as in the simulations of Bech reported in [Bech et al. \(1995\)](#), does show strong large structures, whereas Bech report only weak structures, see also [Andersson et al. \(1998\)](#) where they discuss these large structures.

The structures, which fill the gap between the planes, are not stationary neither in time nor space. This is apparent from figure 3.7 which shows an instantaneous velocity field in a  $y$ - $z$ -plane averaged in the streamwise direction. A rather strong pair of eddies can be seen at  $A$ , whereas at  $B$  the eddies are weaker. This indicates that the structures may meander in space and be intermittent in time. It was also observed that when the integral length scale,  $\Lambda_{uux}$ , was large the  $x$ -averaged field showed stronger eddies than when the integral length scale was smaller.

The velocity statistics ([Komminaho et al. 1996](#)) are affected by the computational domain. In figure 3.8 the  $u_{rms}$ -value, averaged in  $x$  and  $z$  direction

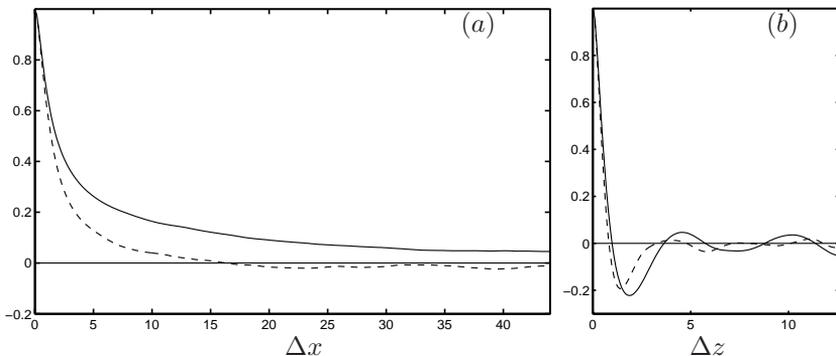


FIGURE 3.9. The two-point correlation for the streamwise velocity and different rotation rates at  $y = 0$ :  $\Omega = 0$  (—) and  $\Omega = -0.005$  (---): (a) for streamwise separation, (b) for spanwise separation.

as well as in time, is shown for three different sizes of the computational domain. The smallest box shows considerably higher values than the other two, indicating that this box is totally inadequate for obtaining accurate quantitative data. It is evident that the smallest box gives this artificially high value of  $u_{rms}$  because of the very strong large structures caused by the strong (resonant) coupling.

The difference between the simulations in the two larger boxes is much smaller, and the values for the largest box compare well with the values obtained from measurements by Tillmark, for the near-wall points. Further out there is a considerable difference in the turbulence intensities, at least partly due to the difference in Reynolds number.

The structures were found to be very sensitive to system rotation. We applied a weak spanwise rotation with a sign so as to stabilise the flow. In contrast to channel flow, both sides in Couette flow are stabilised, or destabilised if the sign of the rotation is switched.

For a rotation rate of  $\Omega = -0.005$  the correlation showed a drastic reduction, and the integral length scale dropped from  $6.1h$  to  $2.0h$ . Larger rotation rates,  $\Omega = -0.03$ , eventually caused a relaminarization of the flow. This relaminarization with high enough rotation rate has also been studied by Tillmark & Alfredsson (1996). They also found that for a Reynolds number of 750 the flow relaminarizes for a rotation rate of  $-0.03$ .

### 3.2.2. Turbulent statistics and Reynolds stress budgets

The maximum streamwise velocity fluctuation is  $u_{rms,max} = 2.76$  for the non-rotating case, a value which is slightly higher than what found in the survey by Mochizuki & Nieuwstadt (1996), where they report a value of 2.71 for a wide

Case $Re_{\delta^*}$	$u_{rms}^+/y^+$	$v_{rms}^+/y^{+2}$	$w_{rms}^+/y^+$	$-\langle uv \rangle^+/y^{+3}$	$\varepsilon^+$
ZPG 539	0.385	0.0112	0.232	0.00099	0.203
ZPG 920	0.398	0.0119	0.252	0.00102	0.223
Couette	0.414	0.0135	0.268	0.00121	0.246

TABLE 3.2. Limiting values for  $y^+ \rightarrow 0$ . Boundary layer data from Skote *et al.* (1998). Couette data from paper 1.

range of Reynolds numbers, for both boundary layer flows as well as pressure-driven channel and pipe flow. The boundary layer experiments by Österlund (1999) show an increasing  $u_{rms,max}$  with Reynolds number, with an estimated limiting value of at least 2.9.

The peak in the streamwise velocity fluctuation is at  $y^+ = 14.1$ , a location which is well reproduced in both boundary layer flows as well as pressure-driven channel and pipe flow. In Mochizuki & Nieuwstadt (1996) they found the location to be about  $y^+ = 15$  for all these flow cases, possibly with a very weak Reynolds number effect. The experiments by Österlund (1999) also show a maximum at about  $y^+ = 15$ .

An interesting quantity is the limiting value of  $u_{rms}/u_\tau$  at the wall. In simulations of channel and boundary layer flows one has found that it increases with increasing Reynolds number. The Couette flow should represent an approximation of the high Reynolds number limit for the near-wall region, since it has a perfectly constant total shear stress. It is interesting to note in table 3.2 that the Couette value is indeed higher than the boundary layer DNS results. Also, the value of 0.41 agrees closely with the recent high-Reynolds number results of Österlund (1999) for measurements in zero-pressure-gradient turbulent boundary layers. Österlund (1999) obtained about 0.41 for a  $Re_\theta = 10000$  and estimated an asymptotic value of about 0.43 for high  $Re$ .

The transport equations for the Reynolds stress tensor are obtained by multiplying (3.1a) (after subtracting the mean equation 3.3) with  $u_j$ , adding the corresponding equation with switched indices  $i, j$  and ensemble averaging. The resulting equations read

$$\frac{DR_{ij}}{Dt} \equiv \left( \frac{\partial}{\partial t} + U_j \frac{\partial}{\partial x_j} \right) R_{ij} = \mathcal{P}_{ij} - \varepsilon_{ij} + \Pi_{ij} + G_{ij} + D_{ij} + T_{ij} + C_{ij} \quad (3.8)$$

where

$$\mathcal{P}_{ij} \equiv -\overline{u_i u_k} \frac{\partial U_j}{\partial x_k} - \overline{u_j u_k} \frac{\partial U_i}{\partial x_k}, \quad (3.9a)$$

$$\varepsilon_{ij} \equiv 2\nu \overline{u_{i,k} u_{j,k}}, \quad (3.9b)$$

$$D_{ij} \equiv \frac{\partial}{\partial x_k} (\nu R_{ij,k}), \quad (3.9c)$$

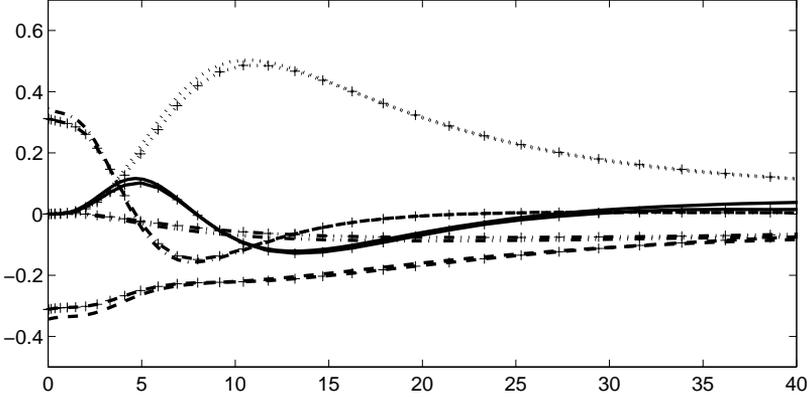


FIGURE 3.10. Terms in the  $R_{11}$  budget for non-rotating Couette and zero pressure gradient boundary layer case. The different terms are: ( $\cdots$ )  $\mathcal{P}_{11}$ , ( $--$ )  $-\varepsilon_{11}$ , ( $-\cdot-\cdot-$ )  $D_{11}$ , ( $-\cdot\cdot\cdot-$ )  $\Pi_{11}$ , ( $-$ )  $T_{11}$ , and the ZPG are marked with (+).

$$\Pi_{ij} \equiv \frac{1}{\rho} \left( p \frac{\partial u_i}{\partial x_j} + p \frac{\partial u_j}{\partial x_i} \right), \quad (3.9d)$$

$$G_{ij} \equiv -\frac{\partial}{\partial x_k} \left( \frac{1}{\rho} \overline{u_j p} \delta_{ik} + \frac{1}{\rho} \overline{u_i p} \delta_{jk} \right), \quad (3.9e)$$

$$T_{ij} \equiv -\frac{\partial}{\partial x_k} \overline{u_i u_j u_k} \quad (3.9f)$$

$$C_{ij} \equiv -2\Omega_k (R_{ij}\epsilon_{ikl} + R_{il}\epsilon_{jkl}). \quad (3.9g)$$

Here  $\mathcal{P}_{ij}$  is the production due to mean field gradients, whose trace ( $\mathcal{P}_{ii}$ ) represents twice the production of turbulent energy, the transfer of energy from the mean flow to the turbulent fluctuations.

$\varepsilon_{ij}$  is the dissipation rate tensor, and  $D_{ij}$  is the diffusion tensor. They both represent viscous effects, but whereas  $D_{ij}$  is a molecular diffusion term acting to even out the turbulent stresses by spatial redistribution,  $\varepsilon_{ij}$  acts as a destruction term of turbulent energy (and stresses).

$\Pi_{ij}$  is the pressure-strain rate correlation tensor, which is traceless and represents inter-component transfer between Reynolds stress terms.  $G_{ij}$  is the divergence of the pressure-velocity correlation, and represents transport driven by pressure fluctuations.

$T_{ij}$  is the divergence of the triple correlation tensor, acting as a spatial redistribution term.  $C_{ij}$  is the traceless Coriolis tensor, which acts as a redistributive term among the stress components.

The different terms in  $R_{11}$  budget are shown in figure 3.10 for both Couette flow and zero pressure-gradient (ZPG) boundary layer flow. It is evident that

the different terms are very similar in both cases. The only differences exist near the wall.

In the near-wall region there is a balance between dissipation and diffusion. The limiting values of the dissipation rate anisotropies,  $e_{ij} = \varepsilon_{ij}/\varepsilon - \frac{2}{3}\delta_{ij}$  (along with the stress anisotropies  $a_{ij} = R_{ij}/K - \frac{2}{3}\delta_{ij}$ ) were calculated in paper 2 and compared with the predictions obtained by the algebraic dissipation rate anisotropy models of [Hallbäck \*et al.\* \(1990\)](#) and [Sjögren & Johansson \(2000\)](#). The agreement is quite satisfactory for both models in the Couette case, while the [Hallbäck \*et al.\*](#) model is in better agreement with DNS data for the ZPG boundary layer.

The [Sjögren & Johansson](#) model gives  $e_{ij} = a_{ij}$  as limiting value in the two-component limit, such as on a solid wall. This describes the situation very accurately in the Couette case.

## CHAPTER 4

### Flows in cylindrical geometries

Turbulence in pipe flow was studied already by Reynolds (1883). Later experiments of fully developed turbulent pipe flow include, among many others, those by Laufer (1954); Towens *et al.* (1972); Lawn (1971); Perry & Abell (1975). More recent experiments include those by Reich & Beer (1989); Westerweel *et al.* (1992); Fontaine & Deutch (1995); Schwartz-van Manen & Nieuwstadt (1996); den Toonder & Nieuwstadt (1997); Zagarola & Smits (1998).

Experiments on annular pipe flow are scarce. One with a moving inner wall is the experiment by Shands *et al.* (1980). Another is that of Nouri *et al.* (1993) where they study Newtonian and non-Newtonian fluids in annular pipe flow.

The Taylor–Couette flow case has attracted many researchers. There are many articles on that subject, see e.g. Chossat & Iooss (1994) and references therein for a wealth of information. As examples of experimental studies see Andereck *et al.* (1986) for pure Taylor–Couette flow, and Takeuchi & Jankowski (1981) for Taylor–Couette flow with a pressure gradient in the axial direction. The latter case was also studied by Leuptow *et al.* (1992) and Wereley & Leuptow (1999).

There are today a few reported studies of turbulent pipe flow using numerical simulations. The early work concerned stability and transition. Fully developed laminar pipe flow shares with Couette flow the property that it is linearly stable over all Reynolds numbers, Salwen & Grosch (1972); Garg & Rouleau (1972).

This seemingly simple flow configuration hides two difficulties when solving Navier–Stokes equations in cylindrical coordinates. The first is purely mathematical and is related to the treatment of the coordinate singularity at the axis  $r = 0$ . The second comes from the numerical treatment of the time derivatives, and is related to the fact that when using cylindrical coordinates the azimuthal grid resolution is proportional to the radial distance from the centreline. Of these two difficulties, the former is most important. The only consequence of not taking the latter into consideration is a decrease in the size of the time-step. One possibility to remedy the time step problem is to filter the solution near the origin (Umsheid & Sankar-Rao 1971) and thereby reducing the resolution.

Another approach is taken by Akselvoll & Moin (1995). They decompose the computational domain into two regions. Within each region the derivatives

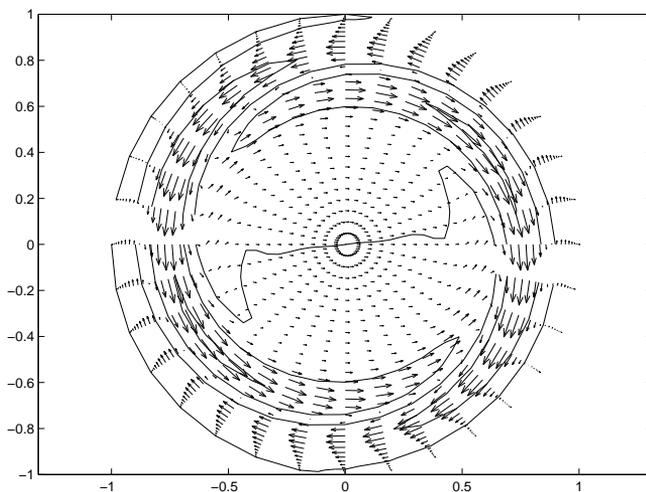


FIGURE 4.1. Velocity vectors in an  $r$ - $\varphi$ -plane, with the streamwise velocity as contours, showing the  $(1, 1)$  eigenmode for pipe flow, at  $Re = 2000$ .

in one direction, radial for the outer, and azimuthal for the inner region, is treated implicitly, thereby avoiding the time-step problem.

When excluding the origin from the computational domain, i.e. in annular pipe or Taylor–Couette flow the time-step problem is less of a concern.

Shapiro *et al.* (1999) recently investigated linear and nonlinear stability of annular pipe flow. The linear stability analysis was performed with Chebyshev collocation method. The nonlinear analysis was made with the same code used by Eggels *et al.* (1994), adapted to the annular domain. They obtained good agreement between the linear theory and their DNS simulation of the growth of the eigenfunctions with small initial amplitude. However, some of the eigenvalues in their eigenvalue map seem to be erroneous, see further section 5.2 in paper 5 where we present eigenvalue maps for the same case.

Moser *et al.* (1983) developed and implemented an algorithm for solving Navier–Stokes equations in a cylindrical geometry, i.e. for Taylor–Couette flow as well as curved channel flow. It is a spectral algorithm, based on Fourier transforms in axial and azimuthal directions and Chebyshev polynomials in the radial direction. The algorithm was evaluated for plane channel and Taylor–Couette flow with both axisymmetric and wavy vortices, with good agreement with theoretical and experimental results. This code was further used in Moser & Moin (1987) in a study of curved channel flow.

#### 4.1. Verification of pipe and annular pipe flow code

Initially the convergence of the Chebyshev expansion in the radial direction was studied, with good results, by solving Laplace equation on a circular disc

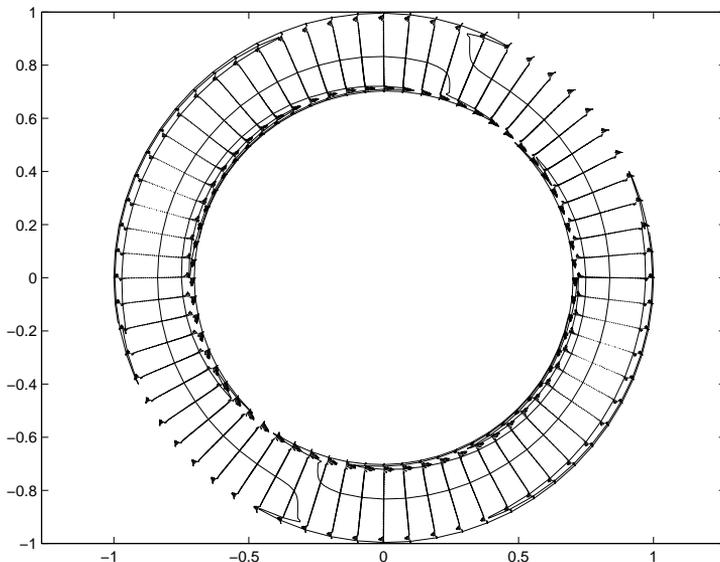


FIGURE 4.2. Velocity vectors in an  $r$ - $\varphi$ -plane, with the streamwise velocity as contours, showing the  $(1, 1)$  eigenmode for annular pipe flow at  $Re = 10000$ , with  $\delta = 0.3$ .

. We used two different boundary conditions, one with a discontinuous second derivative and one smooth function, and obtained the expected convergence with increasing number of modes.

A much more demanding test is to study the evolution of developing eigenmodes. Especially three-dimensional eigenmodes with flow through the origin was studied to verify that the code could correctly handle flow through the origin.

The verification of the pipe flow code was done by studying the evolution of eigenmodes, the Orr–Sommerfeld–Squire modes. The solver for the eigenmodes was graciously provided by Peter Schmid at University of Washington, and has been used in a study of pipe flow and growth of optimal disturbances by [Schmid & Henningson \(1993\)](#).

The streamwise base flow is of the form  $W = W(r)$ . Following [Schmid & Henningson](#) the eigenvalue problem can be written in radial velocity, radial vorticity form as:

$$(-i\omega + i\beta W)T\Phi - \frac{i\beta}{r} \left( \frac{W'}{k^2 r} \right)' \Phi = \frac{1}{Re} T(k^2 r^2 T)\Phi + \frac{2\alpha^2 \beta}{Re} T\Omega \quad (4.1)$$

$$k^2 r^2 (-i\omega + i\beta W)\Omega + \frac{iW'}{r} \Phi = \frac{1}{Re} S\Omega + \frac{2\beta}{Re} T\Phi \quad (4.2)$$

where

$$k^2 = \frac{\alpha^2}{r^2} + \beta^2 \quad (4.3)$$

$$T = \frac{1}{r^2} - \frac{1}{r} \frac{d}{dr} \left( \frac{1}{k^2 r} \frac{d}{dr} \right) \quad (4.4)$$

$$S = k^4 r^2 - \frac{1}{r} \frac{d}{dr} \left( k^2 r^3 \frac{d}{dr} \right) \quad (4.5)$$

$$\Phi \equiv -ir\hat{u} \quad (4.6)$$

$$\Omega \equiv \frac{\beta r \hat{v} - \alpha \hat{w}}{\alpha k^2 r^2}. \quad (4.7)$$

The velocities can now easily be calculated from the eigenmodes corresponding to the eigenvalue

$$\hat{u} = \frac{i\Phi}{r} \quad (4.8)$$

$$\hat{v} = -\frac{\alpha}{k^2 r^2} \frac{\partial \Phi}{\partial r} + \alpha \beta r \Omega \quad (4.9)$$

$$\hat{w} = -\frac{\beta}{k^2 r} \frac{\partial \Phi}{\partial r} - \alpha^2 \Omega. \quad (4.10)$$

In figures 4.1 and 4.2 we show the eigenmodes used in the verification of the pipe and annular pipe flow codes. The eigenvalues in the pipe flow case was  $(c_r, c_i) = (0.4064055, -0.06988367)$  and the Reynolds number based on radius and centre line velocity 2000. For the annular pipe flow case the eigenvalue was  $(c_r, c_i) = (0.23794, 0.0017622)$ , and the Reynolds number based on half the gap with between the cylinders and the centre line velocity was 10000. The evolution in time was studied and excellent agreement with linear theory was obtained. The time advancement scheme was also verified to be of second order.

## 4.2. Future work

There are some remaining issues with the algorithm, for the case of pipe flow. When including the origin in the computational domain one has to be careful with the behaviour of the velocities when approaching the origin, i.e. when  $r \rightarrow 0$ . There are some constraints that the velocities have to obey to get a regular, well behaved velocity field near the origin.

Implementing these constraints for the velocities near the origin remains to be done. One complication is that the expansion is from  $r = -1$  to  $r = 1$ , and we can not impose any boundary conditions at  $r = 0$ .

This is not an issue in the annular pipe flow case, since the origin is excluded from the computational domain.

## Acknowledgement

I would like to thank my supervisor Prof. Arne Johansson for introducing me to the field of turbulence, especially numerical simulation of turbulence. Arne has always studied my manuscripts carefully and made valuable comments. His great knowledge in mathematics in general and turbulence in particular has been very useful in achieving good scientific research.

I would like to thank all my colleges and friends at the department, especially those which I had the pleasure to share room with, for providing a very stimulating and professional atmosphere to work in. Working conditions at the department have always been excellent, especially after the introduction of the coffee machine. My colleagues have always had time to take a cup of coffee with me and relax from the (sometimes) frustrating chores of programming. Also the computer resources have been well provided for, an important factor in the field of numerical simulations.

Last but not least I would like to thank Mia for your love.

## Bibliography

- AKSELVOLL, K. & MOIN, P. 1995 An efficient method for temporal integration of the Navier–Stokes equations in confined axisymmetric geometries. *J. Comp. Phys.* **125**, 454–463.
- ANDERECK, C. D., LIU, S. S. & SWINNEY, H. L. 1986 Flow regimes in a circular Couette system with independently rotating cylinders. *J. Fluid Mech.* **164**, 155–183.
- ANDERSSON, H. I., LYGREN, M. & KRISTOFFERSEN, R. 1998 Roll cells in turbulent Couette flow: Reality or artifact? In *Sixteenth Int. Conf. on Num. Meth. in Fluid Dyn.* (ed. C.-H. Bruneau), pp. 117–122. Springer–Verlag.
- AYDIN, E. M. & LEUTHEUSSER, H. J. 1991 Plane-Couette flow between smooth and rough walls. *Exps. Fluids* **11**, 302–312.
- BATCHELOR, G. K. 1969 Computation of the energy spectrum in two-dimensional turbulence. *Phys. Fluids Suppl. II* **12**, 233–239.
- BECH, K. H. & ANDERSSON, H. I. 1994 Very-large-scale structures in DNS. In *Direct and Large-Eddy Simulations I* (ed. P. R. Voke, L. Kleiser & J.-P. Chollet), pp. 13–24. Dordrecht: Kluwer.
- BECH, K. H., TILLMARK, N., ALFREDSSON, P. H. & ANDERSSON, H. I. 1995 An investigation of turbulent Couette flow at low Reynolds numbers. *J. Fluid Mech.* **286**, 291–325.
- BOYD, J. P. 1989 *Chebyshev and Fourier spectral methods. Lecture Notes in Engineering* 49. Berlin New York: Springer–Verlag.
- BRACHET, M. E., MENEGUZZI, M., POLITANO, H. & SULEM, P. L. 1988 The dynamics of freely decaying two-dimensional turbulence. *J. Fluid Mech.* **194**, 333–349.
- CANUTO, C., HUSSAINI, M. Y., QUARTERNI, A. & ZANG, T. A. 1988 *Spectral Methods in Fluid Dynamics. Springer series in computational physics*. Berlin New York: Springer–Verlag.
- CHOSSAT, P. & IOOSS, G. 1994 *The Couette–Taylor problem. Applied mathematical sciences* 102. Berlin Heidelberg New York: Springer–Verlag.
- COUETTE, M. 1890 Etudes sur le frottement des liquides. *Ann. Chim. Phys.* **21**, 433–510.
- DAUCHOT, O. & DAVIAUD, F. 1995 Finite amplitude perturbation and spot growth mechanism in plane Couette flow. *Phys. Fluids* **7** (2), 335–343.
- DAVIAUD, F., HEGSETH, J. & BERGÉ, P. 1992 Subcritical transition to turbulence in plane Couette flow. *Phys. Lett. Rev.* **69**, 2511–2514.

- DEARDORFF, J. W. 1970 A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *J. Fluid Mech.* **41**, 453–480.
- DENNIS, S. C. R., INGHAM, D. B. & COOK, R. N. 1979 Finite-difference methods for calculating steady incompressible flows in three dimensions. *J. Comp. Phys.* **33** (3), 325–339.
- DRAZIN, P. G. & REID, W. R. 1981 *Hydrodynamic stability*. Cambridge: Cambridge University Press.
- EGGELS, J. G. M., UNGER, F., WEISS, M. H., WESTERWEEL, J., ADRIAN, R. J., FRIEDRICH, R. & NIEUWSTADT, F. T. M. 1994 Fully developed turbulent pipe flow: A comparison between direct numerical simulations and experiments. *J. Fluid Mech.* **268**, 175–209.
- FONTAINE, A. A. & DEUTCH, S. 1995 Three-component, time-resolved velocity statistics in the wall region of a turbulent pipe flow. *Exps. Fluids* **18** (3), 168–173.
- FORNBERG, B. 1996 *A practical guide to pseudospectral methods*. Cambridge Monographs on Applied and Computational Mathematics 1. Cambridge: Cambridge University Press.
- GARG, V. K. & ROULEAU, W. T. 1972 Linear stability of pipe Poiseuille flow. *J. Fluid Mech.* **54**, 113–127.
- GAVRILAKIS, S. 1992 Numerical simulation of low-Reynolds-number turbulent flow through a straight square duct. *J. Fluid Mech.* **244**, 101–129.
- GOTTLIEB, D. & ORSZAG, S. A. 1977 *Numerical analysis of spectral methods: Theory and applications*. Regional conference series in applied mathematics 26. Philadelphia: SIAM.
- GREENGARD, L. 1988 Spectral integration and two-point boundary value problems. Tech. Rep. YALEU/DCS/RR-646. Yale Univ. Research Center for Scientific Computation.
- GRESHO, P. M. & SANI, R. L. 1987 On pressure boundary conditions for the incompressible Navier–Stokes equations. *Int. J. Numer. Methods Fluids* **7**, 1111–1145.
- HALLBÄCK, M. 1993 Development of Reynolds stress closures of homogenous turbulence through physical and numerical experiments. Phd thesis, Dept. of Mechanics, Royal Institute of Technology, Stockholm.
- HALLBÄCK, M., GROTH, J. & JOHANSSON, A. V. 1990 An algebraic model for non-isotropic turbulent dissipation rate term in Reynolds stress closures. *Phys. Fluids A* **2** (10), 1859–1866.
- HALLBÄCK, M., HENNINGSON, D. S., JOHANSSON, A. V. & ALFREDSSON, P. H. 1996 *Turbulence and transition modelling*. ERCOFTAC series 2. Kluwer academic publishers.
- HENNINGSON, D. S., LUNDBLADH, A. & JOHANSSON, A. V. 1993 A mechanism for bypass transition from localized disturbances in wall bounded shear flows. *J. Fluid Mech.* **250**, 169–207.
- JIMÉNEZ, J., WRAY, A. A., SAFFMAN, P. G. & ROGALLO, R. S. 1993 The structure of intense vorticity in isotropic turbulence. *J. Fluid Mech.* **255**, 65–90.
- KIM, J., MOIN, P. & MOSER, R. 1987 Turbulence statistics in fully developed channel flow. *J. Fluid Mech.* **177**, 133–166.
- KOMMINAHO, J., LUNDBLADH, A. & JOHANSSON, A. V. 1996 Very large structures in plane turbulent Couette flow. *J. Fluid Mech.* **320**, 259–285.
- KOMMINAHO, J., LUNDBLADH, A. & JOHANSSON, A. V. 1997 Determination of the

- transition Reynolds number in plane Couette flow through study of relaminarization. In *First AFOSR Int. Conf. on DNS/LES* (ed. C. Liu & Z. Liu), pp. 233–240. Louisiana Tech University, Ruston, LA, USA: Grayden Press.
- KREISS, H.-O. & OLIGER, J. 1972 Comparison of accurate methods for the integration of hyperbolic equations. *Tellus* **24**, 199–215.
- KRISTOFFERSEN, R., BECH, K. H. & ANDERSSON, H. I. 1993 Numerical study of turbulent plane Couette flow at low Reynolds number. *Appl. Sci. Research.* **51**, 337–343.
- KURTZWEIL, R. 1999 *The age of spiritual machines*. Phoenix.
- LAUFER, C. J. 1954 The structure of turbulence in fully developed pipe flow. Technical Report NACA Reports 1158–1209. NACA.
- LAWN, C. J. 1971 The determination of the rate of dissipation in turbulent pipe flow. *J. Fluid Mech.* **48**, 477–505.
- LEE, M. J. & KIM, J. 1991 The structure of turbulence in a simulated plane Couette flow. In *Eighth Symp. on Turbulent Shear Flows*, pp. 5.3.1–5.3.6. Tech. University of Munich.
- LEUPTOW, R. M., DOCTER, A. & MIN, K. 1992 Stability of axial flow in an annulus with rotating inner cylinder. *Phys. Fluids A* **4** (11), 2446–2455.
- LEUTHEUSSER, H. J. & CHU, V. H. 1971 Experiments on plane Couette flow. *J. Hydr. Div., ASCE* **97**, 1269–1279.
- LILLY, D. K. 1969 Numerical simulation of two-dimensional turbulence. *Phys. Fluids Suppl. II* **12**, 240–249.
- LINDBORG, E. & ALVELIUS, K. 2000 The kinetic energy spectrum of the two-dimensional enstrophy turbulence cascade. *Phys. Fluids* **12** (5), 945–947.
- LUNDBLADH, A., BERLIN, S., SKOTE, M., HILDINGS, C., CHOI, J., KIM, J. & HENNINGSON, D. S. 1999 An efficient spectral method for simulation of incompressible flow over a flat plate. *Tech. Rep.* TRITA-MEK 1999:11. Royal Institute of Technology, Stockholm.
- LUNDBLADH, A., HENNINGSON, D. S. & JOHANSSON, A. V. 1992 An efficient spectral integration method for the solution of the Navier–Stokes equation. *Tech. Rep.* TN 1992-28. FFA, Stockholm.
- LUNDBLADH, A. & JOHANSSON, A. V. 1991 Direct simulation of turbulent spots in plane Couette flow. *J. Fluid Mech.* **229**, 499–516.
- MALLOCK, A. 1896 Experiments on fluid viscosity. *Phil. Trans. Roy. Soc. A* **187**, 41–56.
- SCHWARTZ-VAN MANEN, A. D. & NIEUWSTADT, F. T. M. 1996 Large-scale structures in a numerical and experimental low-Reynolds number turbulent pipe flow. *Eur. J. Mech. B/Fluids* **15** (6), 897–915.
- MANSOUR, N. N., MOIN, P., REYNOLDS, W. C. & FERZIGER, J. H. 1979 Improved methods for large eddy simulations of turbulence. In *Turbulent Shear Flows I* (ed. S. . W. Durst, Launder). Springer-Verlag.
- MIYAKE, Y., KAJISHIMA, T. & OBANA, S. 1987 Direct numerical simulation of plane Couette flow at transitional Reynolds number. *JSME Int. J.* **30** (259), 57–65.
- MOCHIZUKI, S. & NIEUWSTADT, F. T. M. 1996 Reynolds-number-dependence of the maximum in the streamwise velocity fluctuations in wall turbulence. *Exps. Fluids* **21**, 218–226.

- MOIN, P. & KIM, J. 1982 Numerical investigation of turbulent channel flow. *J. Fluid Mech.* **118**, 341–377.
- MOSER, R. D., KIM, J. & MANSOUR, N. N. 1999 Direct numerical simulation of turbulent channel flow up to  $Re_\tau = 590$ . *Phys. Fluids* **11** (4), 943–945.
- MOSER, R. D. & MOIN, P. 1987 The effect of curvature in wall-bounded turbulent flows. *J. Fluid Mech.* **175**, 479–510.
- MOSER, R. D., MOIN, P. & LEONARD, A. 1983 A spectral method for the Navier–Stokes equations with application to Taylor–Couette flow. *J. Comp. Phys.* **52**, 524–544.
- NAGATA, M. 1990 Three-dimensional finite-amplitude solutions in plane Couette flow: bifurcations from infinity. *J. Fluid Mech.* **217**, 519–527.
- NOURI, J. M., UMUR, H. & WHITELAW, J. H. 1993 Flow of Newtonian and non-Newtonian fluids in concentric and eccentric annuli. *J. Fluid Mech.* **253**, 617–641.
- ORSZAG, S. A. & KELLS, L. C. 1980 Transition to turbulence in plane Poiseuille flow and plane Couette flow. *J. Fluid Mech.* **96**, 159–205.
- ORSZAG, S. A. & PATTERSON, G. S. 1972 Numerical simulation of three-dimensional homogenous isotropic turbulence. *Phys. Lett. Rev.* **28**, 76–79.
- PAPAVASSILIOU, D. V. & HANRATTY, T. J. 1997 Interpretation of large-scale structures observed in a turbulent plane Couette flow. *Int. J. Heat and Fluid Flow* **18** (1), 55–69.
- PERRY, A. E. & ABELL, C. J. 1975 Scaling laws for pipe-flow turbulence. *J. Fluid Mech.* **67**, 257–271.
- RAI, M. M. & MOIN, P. 1991 Direct simulation of turbulent flow using finite-difference schemes. *J. Comp. Phys.* **96**, 15–53.
- REICH, G. & BEER, H. 1989 Fluid flow and heat transfer in an axially rotating pipe –I. Effect of rotation on turbulent pipe flow. *J. Heat Mass. Transfer* **32** (3), 551–562.
- REICHARDT, H. 1956 Über die Geschwindigkeitsverteilung in einer geradlinigen turbulenten Couetteströmung. *Z. Angew. Math. Mech.* **36**, 26–29.
- REICHARDT, H. 1959 Gezetzmässigkeiten der geradlinigen turbulenten Couetteströmung. *Mitt. Max-Planck-Istitut für Strömungsforschung* **22**, Göttingen.
- REYNOLDS, O. 1883 An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and the law of resistance in parallel channels. *Phil. Trans. Roy. Soc. Lond.* **174**, 935–982.
- ROBERTSON, J. M. & JOHNSON, H. F. 1970 Turbulence structures in plane Couette flow. *J. Eng. Mech. Div. ASCE* **96**, 1171–1172.
- ROGALLO, R. S. 1981 Numerical experiments in homogeneous turbulence. *Tech. Rep.* TM-81315. NASA.
- ROMANOV, V. A. 1973 Stability of plane-parallel Couette flow. *Funkcional Anal. i Proloženi* **7** (2), 62–73, Translated in *Functional Anal. & Its Applics* **7**, 137–146 (1973).
- SALWEN, H. & GROSCH, C. E. 1972 The stability of Poiseuille flow in a pipe of circular cross-section. *J. Fluid Mech.* **54**, 93–112.
- SCHMID, P. J. & HENNINGSON, D. S. 1993 Optimal energy density growth in Hagen–Poiseuille flow. *J. Fluid Mech.* **277**, 197–225.
- SCHUMANN, U. 1975 Subgrid scale models for finite difference simulations of turbulent flows in plane channel and annuli. *J. Comp. Phys.* **18** (4), 376–404.

- SHANDS, J., ALFREDSSON, H. & LINDGREN, E. R. 1980 Annular pipe flow subject to axial motion of the inner boundary. *Phys. Fluids* **23** (10), 2144–2145.
- SHAPIRO, I., SHTILMAN, L. & TUMIN, A. 1999 On stability of flow in an annular channel. *Phys. Fluids* **11** (10), 2084–2992.
- SHUMANN, U. & PATTERSON, G. S. 1978 Numerical study of pressure and velocity fluctuations in nearly isotropic turbulence. *J. Fluid Mech.* **88**, 685–709.
- SJÖGREN, T. & JOHANSSON, A. V. 2000 Development and calibration of algebraic nonlinear models for terms in the Reynolds stress transport equations. *Phys. Fluids* **12** (6), 1554–1572.
- SKOTE, M., HENKES, R. A. W. M. & HENNINGSON, D. S. 1998 Direct numerical simulation of self-similar turbulent boundary layers in adverse pressure gradients. *Flow, turbulence and combustion* **60**, 47–85.
- STRIKWERDA, J. C. 1989 *Finite difference schemes and partial differential equations. Mathematics series*. New York: Wadsworth & Brooks/Cole.
- TAKEUCHI, D. I. & JANKOWSKI, D. F. 1981 A numerical and experimental investigation of the stability of spiral Poiseuille flow. *J. Fluid Mech.* **102**, 101–126.
- TAYLOR, G. I. 1923 Stability of a viscous liquid contained between two rotating cylinders. *Phil. Trans. Roy. Soc. Lond. A* **223**, 289–343.
- TILLMARK, N. 1995 Experiments on transition and turbulence in plane Couette flow. PhD thesis, Department of Mechanics, Royal Institute of Technology, Stockholm.
- TILLMARK, N. & ALFREDSSON, P. H. 1992 Experiments on transition in plane Couette flow. *J. Fluid Mech.* **235**, 89–102.
- TILLMARK, N. & ALFREDSSON, P. H. 1994 Structures in turbulent plane Couette flow obtained from correlation measurements. In *Advances in turbulence V* (ed. R. Benzi), pp. 502–507. Kluwer, proceedings of the Fifth European Turbulence Conference, Sienna.
- TILLMARK, N. & ALFREDSSON, P. H. 1996 Experiments on rotating plane Couette flow. In *Advances in Turbulence VI* (ed. S. Gavrilakis, L. Machiels & P. A. Monkewitz), pp. 391–394. Kluwer, proceedings of the Sixth European Turbulence Conference, Lausanne, Switzerland, 2–5 July.
- DEN TOONDER, J. M. J. & NIEUWSTADT, F. T. M. 1997 Reynolds number effects in a turbulent pipe flow for low to moderate Re. *Phys. Fluids* **9** (11), 3398–3409.
- TOWENS, H. W., GOW, J. L., POWE, R. E. & WEBER, N. 1972 Turbulent flow in smooth and rough pipes. *Trans. ASME J. of Basic Engr.* **6**, 353–362.
- UMSHEID, L. & SANKAR-RAO, M. 1971 Further tests of a grid system for global numerical prediction. *Mon. Weather Rev.* **99** (9), 686–690.
- WERELEY, S. T. & LEUPTOW, R. M. 1999 Velocity field for Taylor–Couette flow with an axial flow. *Phys. Fluids* **11** (12), 3637–3649.
- WESTERWEEL, J., ADRIAN, R. J., EGGELS, J. G. M. & NIEUWSTADT, F. T. M. 1992 Measurements with particle image velocimetry of fully turbulent pipe flow at low Reynolds number. In *Proc. 6th Int. Symp. on Applications of Laser Tech. to Fluid Mechanics*. Lisbon, Portugal.
- YOUNG, N. 1990 *An introduction to Hilbert space. Cambridge mathematical textbooks* 1. Cambridge: Cambridge university press.
- ZAGAROLA, M. V. & SMITS, A. J. 1998 Mean-flow scaling of turbulent pipe flow. *J. Fluid Mech.* **373**, 33–79.
- ZHANG, Y., GANDHI, A., TOMBOULIDES, A. G. & ORSZAG, S. A. 1994 Simulation

of pipe flow. In *Application of direct and large eddy simulation to transition and turbulence*, pp. 17.1–17.9. AGARD Conf, Creta, Greece.

ÖSTERLUND, J. 1999 Experimental studies of zero pressure-gradient turbulent boundary layer flow. Phd thesis, Dept. of Mechanics, Royal Institute of Technology, Stockholm.

P1

# Paper 1



# Very large structures in plane turbulent Couette flow

By **Jukka Komminaho, Anders Lundbladh**  
and  
**Arne V. Johansson**

Royal Institute of Technology, Dept of Mechanics, Stockholm, Sweden

Published in *J. of Fluid Mech.*

A direct numerical simulation was carried out of plane turbulent Couette flow at a Reynolds number of 750, based on half the velocity difference between the walls and half the channel width. Particular attention was paid to choosing a computational box that is large enough to accommodate even the largest scales of the turbulence. In the central region of the channel very large elongated structures were observed, in accordance with earlier findings. The study is focused on the properties of these structures, but is also aimed at obtaining accurate turbulence statistics. Terms in the energy budget were evaluated and discussed. Also, the limiting values of various quantities were determined and their relevance in high Reynolds number flows discussed. The large structures were shown to be very sensitive to an imposed system rotation. They could be essentially eliminated with a stabilizing system rotation (around the spanwise axis) small enough for only a minor damping of the rest of the scales. Despite the fact that the large structures dominate the appearance of the flow field their energy content was shown to be relatively small, on the order of 10% of the total turbulent kinetic energy.

---

Due to copyright restrictions, the following article has been deleted. See *Journal of Fluid Mechanics*, vol. 320, pp. 259–285 for the original article.



Paper 2

P2



# Reynolds stress budgets in Couette and boundary layer flows

By **Jukka Komminaho and Martin Skote**

Dept. of Mechanics, KTH, SE-100 44, Stockholm, Sweden

Reynolds stress budgets for both Couette and boundary layer flows are evaluated and presented. Data are taken from direct numerical simulations of rotating and non-rotating plane turbulent Couette flow and turbulent boundary layer with and without adverse pressure gradient. Comparison of the total shear stress for the two flows suggests that the Couette case may be regarded as the high Reynolds number limit for the boundary layer flow close to the wall. The direction of rotation is chosen so that it has a stabilizing effect, whereas the adverse pressure gradient is destabilizing. The pressure strain-rate tensor is in the Couette flow case presented for a split into slow, rapid and Stokes terms.

---

## 1. Introduction

The development of cheap, powerful, computers has lead to wide use of CFD codes for the prediction of turbulent flows. These codes almost always use turbulence models to try to capture the characteristics of the turbulent flow, and the prediction is no better than the weakest link in computational chain. Often the weakest link is the turbulence model. But to develop better turbulence models one must have data to compare them against. In the early days of turbulence modelling one had to rely on indirect methods to test the various closure models. Experimental difficulties in measuring pressure and velocity with sufficient resolution did not make direct comparisons possible.

With the development of high-speed supercomputers, and new algorithms, Orszag (1969, 1970); Kreiss & Oliger (1972); Basdevant (1983), it became possible to simulate turbulent flows directly without resorting to large eddy simulations or turbulence models. Now it became possible to evaluate any desirable quantity and use them to test turbulence models. The channel flow simulation by Kim *et al.* (1987) was the first fully resolved simulation of a pressure-driven channel flow, and the database from the simulation has been used extensively to evaluate various turbulence models, Mansour *et al.* (1988).

There are few experimental studies of Couette flow with reports of turbulence statistics. In the study of Couette flow at a Reynolds number of 1300, Bech *et al.* (1995), report both second and higher order statistics from both

experiments and simulations. The agreement between the experiments and the simulation is good for the statistics, but their simulations do not fully capture the very large scale structures of the experiments. This is e.g. seen from the two-point correlations which are lower in the simulation than in the experiment. In [Bech & Andersson \(1994\)](#) they used three different sizes of computational domain and observed large structures in one box, but not in the other two. The reason behind this is unclear.

In [Bech \(1995\)](#) they present Reynolds stress budgets from the simulation in [Bech \*et al.\* \(1995\)](#), and they look very similar to the ones presented here, despite the higher Reynolds number in their simulation.

In the present paper the budget data for the Reynolds stresses in the Couette flow case are evaluated from the flow fields of the plane Couette flow simulation by [Komminaho \*et al.\* \(1996\)](#).

Data are also presented from three different turbulent boundary layers. One is a zero pressure gradient (ZPG) boundary layer, and two are boundary layers subject to an adverse pressure gradient (APG). Data from the ZPG boundary layer have not previously been presented. The simulation with a moderate APG (APG1) has been analyzed in [Skote \*et al.\* \(1998\)](#), while the strong APG case (APG2) has been presented in [Skote & Henningson \(2000\)](#).

The ZPG turbulent boundary layer flow has been studied in a large number of investigations, see e.g. the assessment of data by [Fernholz & Finley \(1996\)](#). Turbulent statistics close to the wall were obtained through DNS by [Spalart \(1988\)](#), and were confirmed later in the experiment of a low Reynolds number ZPG turbulent boundary layer by [Ching \*et al.\* \(1995\)](#). Various Reynolds stress budgets from DNS of both ZPG and APG boundary layers were presented by [Na & Moin \(1996\)](#). Near-wall limit values of an APG boundary layer were also investigated in the DNS of [Spalart & Watmuff \(1993\)](#) and in the experiment of [Nagano \*et al.\* \(1992\)](#).

The results from the simulations of [Komminaho \*et al.\* \(1996\)](#), [Skote \*et al.\* \(1998\)](#) and [Skote & Henningson \(2000\)](#) are documented here for future use in turbulence model development, in particular for near-wall modelling. The present plane Couette flow data are well suited for this purpose since the condition of a constant total shear is, unlike the situation in the boundary layer, fulfilled for all Reynolds numbers. The boundary layer data can be used for the development of low Reynolds number turbulence models.

## 2. Data analysis

One can write the Navier–Stokes and continuity equations in a rotating reference frame as,

$$\frac{\partial u'_i}{\partial t} + \frac{\partial}{\partial x_j} (u'_i u'_j) = -\frac{1}{\rho} \frac{\partial p'}{\partial x_i} + \nu \frac{\partial^2 u'_i}{\partial x_j \partial x_j} + 2\epsilon_{ijk} u'_j \Omega_k \quad (1a)$$

$$\frac{\partial u'_i}{\partial x_i} = 0. \quad (1b)$$

The effect of the system rotation can be seen as a volume force in the fluid, also known as the Coriolis force and the centrifugal force. The Coriolis force is the last term in the momentum equation, and the centrifugal force has been included in the pressure.

Divide the flow into a mean and a fluctuating part,  $u' = U + u$ , where the mean part is defined as an ensemble average over  $N$  different times, and also an average over the homogeneous directions ( $x$  and  $z$  in the Couette flow and  $z$  in the boundary layer)

$$\overline{u'} \equiv U(y, t) = \frac{1}{NL_xL_z} \sum_{i=1}^N \int_0^{L_x} \int_0^{L_z} u'(x, y, z, t) dx dz. \quad (2)$$

The Reynolds equation for the mean flow is now obtained as

$$\frac{\partial U_i}{\partial t} + \frac{\partial}{\partial x_j} (U_i U_j) = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} - \frac{\partial}{\partial x_j} R_{ij} + \epsilon_{ijk} U_j \Omega_k \quad (3)$$

where  $R_{ij} = \overline{u_i u_j}$  is the velocity correlation tensor, and will here be referred to as the 'Reynolds stress tensor'.

### 2.1. Couette data

Plane Couette flow is the flow between two parallel planes, moving in opposite directions with velocity  $\pm U_w$  in the  $x$ -direction, at a distance  $2h$ . The wall-normal direction is denoted  $y$ . The system rotation  $\Omega$  applied in the present work is around the  $z$  axis.

The various statistical quantities have been evaluated and averaged from 12 different velocity fields, and the average was taken in both  $x$  and  $z$  direction. The time between the samples was  $T = 40$ , and they are statistically independent for all but the very largest scales, see [Komminaho \*et al.\* \(1996\)](#) where the time scale for the integral length scale ( $\Lambda_{uu}$  defined as  $\int R_{uu}(\Delta x) dx$ ,  $R_{uu}$  being the two-point velocity correlation) was found to be more than 50.

### 2.2. Boundary layer data

The statistics have been produced in the same manner as in the Couette case, except for the important difference that the flow is not homogeneous in the streamwise ( $x$ ) direction. The boundary layer is growing and developing in the  $x$ -direction due to the increasing Reynolds number. Thus, the statistics are unique for each streamwise position. However, here we are only dealing with the near-wall statistics, which in the viscous scaling should be invariant under the Reynolds number. But in the low Reynolds number flows simulated with DNS, there is a small influence of the increasing Reynolds number. This effect is confined to the part very close to the wall ( $y^+ < 3$ ). In the ZPG simulation e.g., the boundary layer undergoes a doubling of the Reynolds number, but the budgets fall on top of each other for different streamwise positions, except for the small increase of the values at the wall. The statistics are therefore shown for one streamwise position in all three cases.

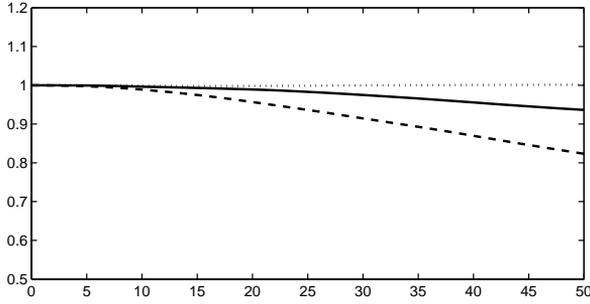


FIGURE 1. Total shear stress. Couette ( $\cdots$ ). Boundary layer: (- -)  $Re_{\delta_*} = 539$ , (—)  $Re_{\delta_*} = 920$ .

The simulations APG1 and APG2 were performed with a pressure distribution leading to a self-similar boundary layer at high Reynolds numbers. The pressure gradient parameter  $\beta$ ,

$$\beta \equiv \frac{\delta_*}{\tau_w} \frac{dP}{dx}, \quad (4)$$

defines the APG in these two simulations.

The Reynolds number at the position where the budgets have been evaluated is shown in table 1, together with the local value of the friction velocity, freestream velocity and pressure gradient parameter.

Case	$Re_{\delta_*}$	$Re_{\Theta}$	$u_\tau$	$U$	$\beta$
ZPG	920	606	0.048	1.0	0.0
APG1	1064	655	0.036	0.76	0.65
APG2	2573	1309	0.020	0.60	5.0

TABLE 1. Reynolds number, friction velocity, freestream velocity and pressure gradient at the streamwise position where the Reynolds stress budgets have been evaluated.

Another effect of the Reynolds number is the increasing length of the region with constant shear stress ( $\tau^+$ ). This is illustrated in figure 1, which shows the total shear stress at two Reynolds numbers for the ZPG case, as well as for Couette flow. From figure 1 it is clear that the total shear stress for the boundary layer becomes more constant when the Reynolds number is increased. Since  $\tau^+$  is constant for the Couette flow, it might be argued that this flow approximates a high Reynolds number boundary layer close to the wall.

### 2.3. Reynolds stress budget

The transport equations for the Reynolds stress tensor are obtained by multiplying (1a) (after subtracting the mean equation 3) with  $u_j$ , adding the corresponding equation with switched indices  $i, j$  and ensemble averaging. The resulting equations read

$$\frac{DR_{ij}}{Dt} \equiv \left( \frac{\partial}{\partial t} + U_j \frac{\partial}{\partial x_j} \right) R_{ij} = \mathcal{P}_{ij} - \varepsilon_{ij} + \Pi_{ij} + G_{ij} + D_{ij} + T_{ij} + C_{ij} \quad (5)$$

where

$$\mathcal{P}_{ij} \equiv -\overline{u_i u_k} \frac{\partial U_j}{\partial x_k} - \overline{u_j u_k} \frac{\partial U_i}{\partial x_k}, \quad (6a)$$

$$\varepsilon_{ij} \equiv 2\nu \overline{u_{i,k} u_{j,k}}, \quad (6b)$$

$$D_{ij} \equiv \frac{\partial}{\partial x_k} (\nu R_{ij,k}), \quad (6c)$$

$$\Pi_{ij} \equiv \frac{1}{\rho} \left( p \frac{\partial u_i}{\partial x_j} + p \frac{\partial u_j}{\partial x_i} \right), \quad (6d)$$

$$G_{ij} \equiv -\frac{\partial}{\partial x_k} \left( \frac{1}{\rho} \overline{u_j p} \delta_{ik} + \frac{1}{\rho} \overline{u_i p} \delta_{jk} \right), \quad (6e)$$

$$T_{ij} \equiv -\frac{\partial}{\partial x_k} \overline{u_i u_j u_k}, \quad (6f)$$

$$C_{ij} \equiv -2\Omega_k (R_{ij} \epsilon_{ikl} + R_{il} \epsilon_{jkl}). \quad (6g)$$

Here  $\mathcal{P}_{ij}$  is the production due to mean field gradients, whose trace ( $\mathcal{P}_{ii}$ ) represents twice the production of turbulent energy, the transfer of energy from the mean flow to the turbulent fluctuations.

$\varepsilon_{ij}$  is the dissipation rate tensor, and  $D_{ij}$  is the diffusion tensor. They both represent viscous effects, but whereas  $D_{ij}$  is a molecular diffusion term acting to even out the turbulent stresses by spatial redistribution,  $\varepsilon_{ij}$  act as a destruction term of turbulent energy (and stresses).

$\Pi_{ij}$  is the pressure-strain rate correlation tensor, which is traceless and represents inter-component transfer between Reynolds stress terms.  $G_{ij}$  is the divergence of the pressure-velocity correlation, and represents transport driven by pressure fluctuations. This split in the above two terms is not unique, there are several different ways in which one may separate the pressure-velocity term when deriving the RST equations, but as the investigation in Groth (1991) shows the above separation seems to make most physical sense.

$T_{ij}$  is the divergence of the triple correlation tensor, acting as a spatial redistribution term.

$C_{ij}$  is the traceless Coriolis tensor, which acts as a redistributive term among the stress components.

The transport equation for the kinetic energy,  $K \equiv \frac{1}{2}\mathcal{P}_{ii}$  is

$$\frac{DK}{Dt} = \mathcal{P} - \varepsilon + \mathcal{D} \quad (7)$$

where  $\mathcal{P} = \frac{1}{2}\mathcal{P}_{ii}$  is the turbulent energy production,  $\varepsilon = \frac{1}{2}\varepsilon_{ii}$  is the viscous dissipation, and  $\mathcal{D} = \frac{1}{2}(T_{ii} + G_{ii} + D_{ii})$  is the sum of the molecular and turbulent diffusion of  $K$ . This term acts as a spatial redistribution of  $K$ .

In a fully developed plane Couette flow, the flow is homogeneous in the  $x$  and  $z$  directions, and the relevant non-zero stresses are  $R_{11}$ ,  $R_{12}$ ,  $R_{22}$  and  $R_{33}$ . Figures 2–8 show the terms in the budget of these stresses, as functions of the wall-normal distance  $y^+ = yu_\tau/\nu$ , where  $u_\tau = \sqrt{\tau_w/\rho}$  is the friction velocity. Note that in the non-rotating case the Coriolis term,  $C_{ij}$ , is zero. All quantities are shown in  $^+$ -units, non-dimensionalized with  $u_\tau^4/\nu$ . The simulation flow fields represent a plane Couette flow at a Reynolds number  $Re_\tau = u_\tau h/\nu = 52$  ( $Re_\tau = 48$  for the rotating case) based on friction velocity  $u_\tau$  and channel half-height  $h$ . This corresponds to a Reynolds number 750 based on wall-velocity and  $h$ . Despite this very low Reynolds number it is twice that of the transition Reynolds number of 360, Lundbladh & Johansson (1991); Tillmark & Alfredsson (1992); Komminaho *et al.* (1997). For the rotating case the rotation is as low as  $\Omega = -0.005$ , corresponding to a Rossby number of 200.

The budgets for the Reynolds stresses in the ZPG case are essentially the same as in Spalart (1988). The moderate APG case, APG1, show very similar profiles in the Reynolds stress budgets as the APG simulation of Na & Moin (1996). The effects of the APG will be stronger in the APG2 case, which has a skin friction approximately 60 % of that in APG1. In this work, in contrast to the budgets in Spalart (1988) and Na & Moin (1996), the pressure term is divided into pressure-strain and pressure-velocity diffusion, for comparison with the Couette data.

In figures 2 to 9 the budgets for the Reynolds stresses are shown. The figures include both non-rotating and rotating Couette flow as well as all three boundary layer cases and the profiles from the ZPG case can be compared with the Couette case with zero rotation.

### 2.3.1. Longitudinal Reynolds stress

One may note that the maximum of the production term  $\mathcal{P}_{11}$  is 0.5. This is easily obtained by integrating the stream-wise momentum equation once, and multiplying with  $\frac{dU^+}{dy^+}$ . The advection term is zero in the Couette flow case negligible in the near-wall region for boundary layers. By neglecting the advection term and assuming wall similarity, we obtain the following relation for the turbulence production:

$$\mathcal{P}_{11} \equiv -2\frac{\overline{uv}}{u_\tau^2} \frac{dU^+}{dy^+} = 2\frac{dU^+}{dy^+} \left( 1 - \frac{dU^+}{dy^+} + \frac{\nu}{\rho u_\tau^3} \frac{dP}{dx} y^+ \right), \quad (8)$$

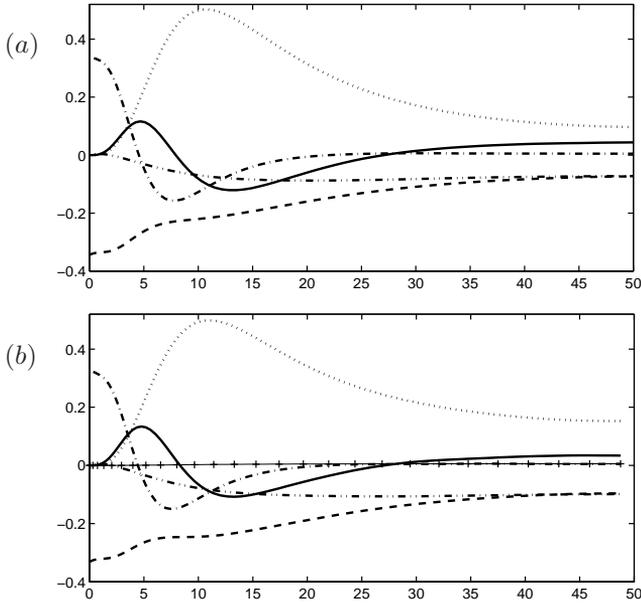


FIGURE 2. Terms in the Couette flow  $R_{11}$ -budget for (a) the non-rotating case and (b) the rotating case,  $\Omega = -0.005$ . The different terms are: ( $\cdots$ )  $\mathcal{P}_{11}$ , ( $--$ )  $-\varepsilon_{11}$ , ( $-\cdot-$ )  $D_{11}$ , ( $-\cdot\cdot-$ )  $\Pi_{11}$ , ( $-$ )  $T_{11}$ , ( $+$ )  $C_{11}$ .

where the pressure gradient term is non-zero only in the adverse pressure gradient (APG) cases. The last term within the parenthesis can be rewritten as  $\beta y^+ / \delta_*^+$ . From the above relation it follows that the maximum of  $\mathcal{P}_{11}$  is 0.5 occurring at a position where  $dU^+ / dy^+ = 0.5$  for Couette flow and ZPG boundary layer. This holds irrespective of the value of the Reynolds number and the system rotation and was shown to accurately describe also the low-Reynolds number plane Couette flow simulation of [Komminaho \*et al.\* \(1997\)](#) where the Reynolds number was as low as 375.

The overall character of the different terms in the Reynolds stress budget for  $R_{ij}$  is the same as for the channel flow in [Mansour \*et al.\* \(1988\)](#). Figure 2 shows that the production term  $\mathcal{P}_{11}$  is the dominant positive term in the range  $y^+ > 5$ , and has a maximum of 0.5 in the buffer region, at  $y^+ = 11$ , falling to 0.10 in the centre of the channel. The location of the peak production can be found to be  $y^+ \approx 11$  also in channel and pipe flow, [Sahay & Sreenivasan \(1999\)](#). The non-zero production in the central region is a consequence of the non-zero mean shear in this region.

$\Pi_{11}$  is negative throughout the channel, thereby transferring energy from  $R_{11}$  to  $R_{22}$  and  $R_{33}$ .

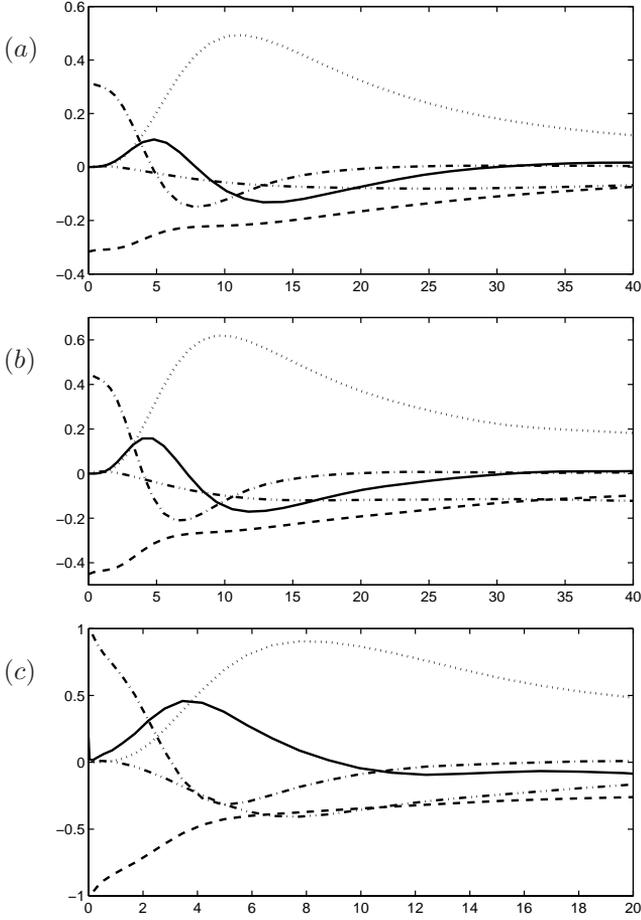


FIGURE 3. Terms in the  $R_{11}$ -budget for boundary layer flow (a) ZPG. (b) APG1. (c) APG2. The different terms are: ( $\cdots$ )  $\mathcal{P}_{11}$ , ( $--$ )  $-\varepsilon_{11}$ , ( $-\cdot-$ )  $D_{11}$ , ( $-\cdot\cdot-$ )  $\Pi_{11}$ , ( $-$ )  $T_{11}$ .

Despite the very low rotation rate for the Couette flow case the effects on some terms in the budgets are significant, away from the wall. The production  $\mathcal{P}_{11}$  is about 60% larger in the centre of the channel for the rotating case. The dissipation  $\varepsilon_{11}$  and the pressure-strain-rate  $\Pi_{11}$  are both 30% larger for the rotating case, whereas the redistributive term  $T_{11}$  is about 20% smaller. Near the walls the non-rotating and rotating cases are very similar, as can be expected since the maximum production is 0.5 in both cases.

In figure 3a the budget for the longitudinal Reynolds stress is shown for the ZPG case. The maximum of the production term  $\mathcal{P}_{11}$  is 0.5 as in the Couette case. The other terms in the budget for  $R_{11}$  corresponds very closely to those in the Couette case. The adverse pressure gradient increases the production  $\mathcal{P}_{11}$

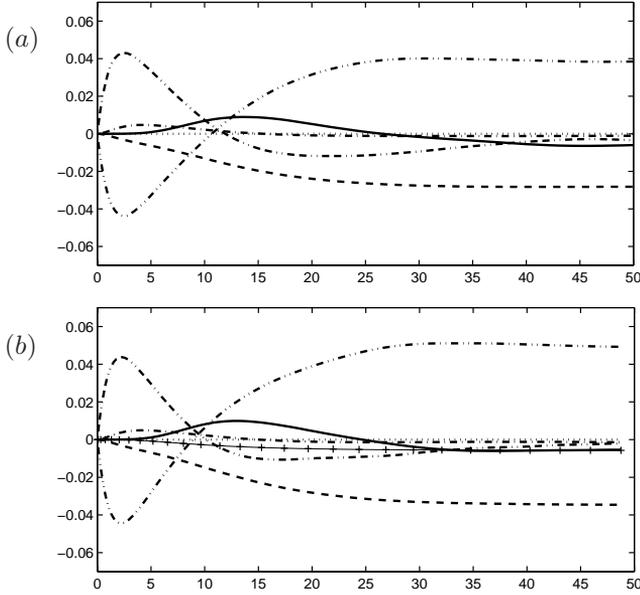


FIGURE 4. Terms in the Couette flow  $R_{22}$ -budget for (a) the non-rotating case and (b) the rotating case,  $\Omega = -0.005$ . The different terms are: ( - - )  $-\varepsilon_{22}$ , ( - · - )  $D_{22}$ , ( - · · - )  $\Pi_{22}$ , ( - · · · )  $G_{22}$ , ( — )  $T_{22}$ , ( + )  $C_{22}$ .

as seen in figures 3b and c. For APG1 it is 0.6 and APG2 0.9. The increase of the maximum is not explained by the contribution from the streamwise velocity gradient since that part of the production term is negligible close to the wall.

The increased value of  $\mathcal{P}_{11}$  is thus explained from the contribution from the pressure gradient in equation (8). For the case APG2 we have a  $\delta_*^+$  of 86 so that the last term within the paranthesis in equation (8)  $\beta y^+ / \delta_*^+$  is about 0.58 at  $y^+ = 10$ , i.e. near the maximum in production. It can, hence, be seen to be of the order one influence. Since  $\beta y^+ / \delta_*^+ = \beta y^+ \frac{U_\infty}{u_\tau} / Re_{\delta_*}$  we can see that the effect of the pressure gradient term decreases with increasing Reynolds number.

The position of the maximum is shifted towards the wall, most notably in the APG2 case (figure 3c).

Also the rest of terms show more extreme values in the APG cases, even though the shape of the profiles remain roughly the same. The enhanced values in the near-wall region are partly due to the decrease in the friction velocity (which all the terms in the budget are scaled with). The lower value of  $u_\tau$  is a consequence of the adverse pressure gradient. One might argue that  $u_\tau$  is not the correct scaling in an APG flow, since the total shear stress is not constant in this scaling. Alternative scalings, including a velocity scale dependent on

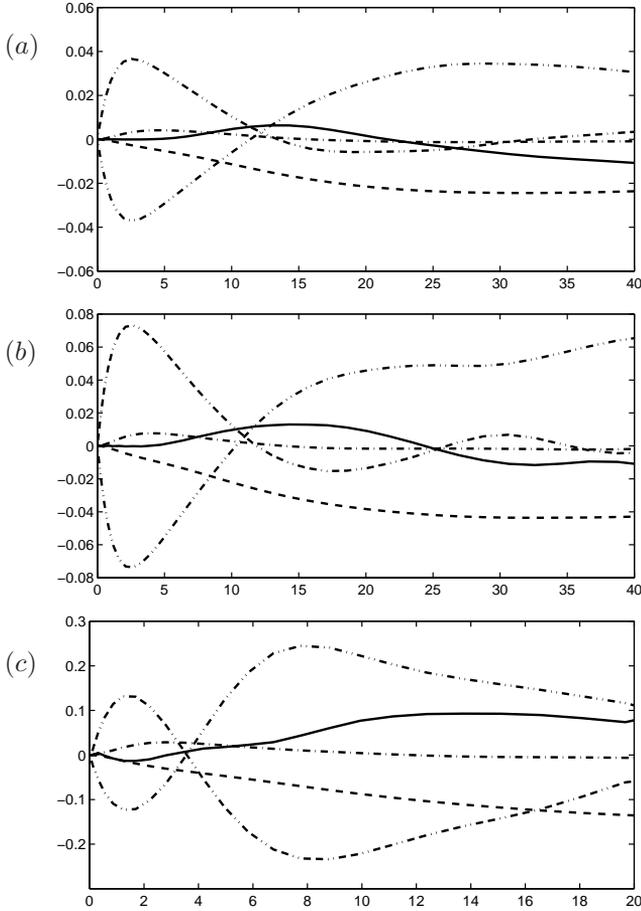


FIGURE 5. Terms in the  $R_{22}$ -budget for boundary layer flow (a) ZPG. (b) APG1. (c) APG2. The different terms are: (---)  $-\varepsilon_{22}$ , (-·-)  $D_{22}$ , (-·-·-)  $\Pi_{22}$ , (- - -)  $G_{22}$ , (—)  $T_{22}$ .

the wall normal distance that produce a constant shear stress, are discussed in [Skote & Henningson \(1999\)](#) and [Skote & Henningson \(2000\)](#).

### 2.3.2. Normal Reynolds stress

In figure 4 the budget for  $R_{22}$  in the Couette flow case is shown.  $\Pi_{22}$  is negative close to the wall, and positive towards the centre. Thus it transfers energy from the wall-normal components to the horizontal components near the wall. This reversal of the sign was attributed to the splatting effect in the LES study of turbulent channel flow by [Moin & Kim \(1982\)](#) (see also [Hunt & Graham 1978](#)). In the turbulence modelling context this effect is normally referred to as the wall-reflection contribution to the pressure strain. The attempts to

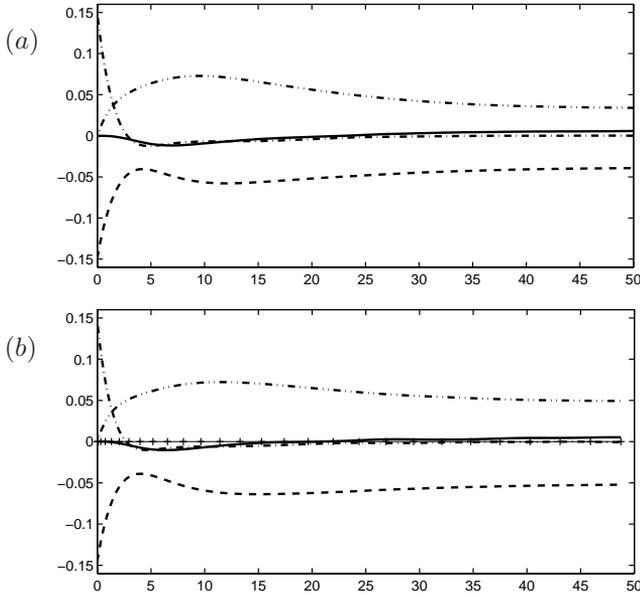


FIGURE 6. Terms in the Couette flow  $R_{33}$ -budget for (a) the non-rotating case and (b) the rotating case,  $\Omega = -0.005$ . The different terms are: (---)  $-\varepsilon_{33}$ , (-·-·)  $D_{33}$ , (-·-·-·)  $\Pi_{33}$ , (—)  $T_{33}$ , (+)  $C_{33}$ .

model this (see [Gibson & Launder 1978](#)) typically assumes a variation on a length-scale of the order of the macro-scale. The present results and those of [Aronson \*et al.\* \(1997\)](#) and [Perot & Moin \(1995\)](#) however show that the effect is confined to a thin region near the wall. In some recent model development (see e.g. [Sjögren & Johansson 2000](#)) this effect is only indirectly accounted for through realizable models.

The same trend regarding the dissipation and the pressure-strain rate can also be seen in figures 4 and 6 for  $R_{22}$  and  $R_{33}$  budgets.

In figure 5a the budget for  $R_{22}$  in the ZPG case is shown. All the terms show slightly lower values than in the Couette case, while the shapes of the profiles are similar. As the pressure gradient increases, all the terms become larger, as seen from figures 5b and c. A peak in the pressure-strain term has developed in the APG2 case at the position  $y^+ = 8$ , and exceeds the maximum value of the pressure diffusion. The formation of a peak is not observed in the ZPG and APG1 cases, where a plateau is developed in the pressure-strain, and the value is lower than the pressure-velocity gradient.

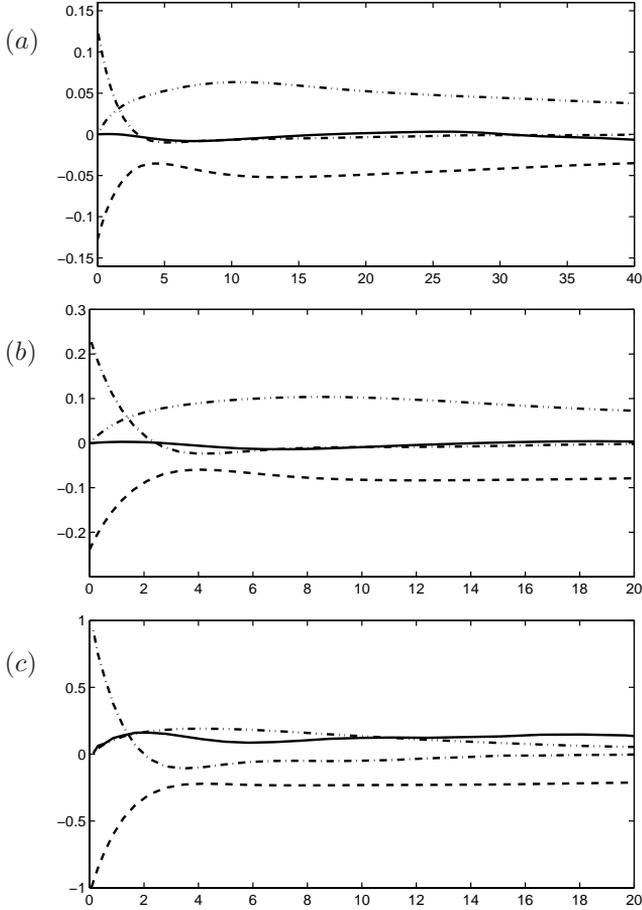


FIGURE 7. Terms in the  $R_{33}$ -budget for boundary layer flow (a) ZPG. (b) APG1. (c) APG2. The different terms are:  $(- -)$   $-\varepsilon_{33}$ ,  $(- \cdot - \cdot)$   $D_{33}$ ,  $(- \cdot \cdot \cdot -)$   $\Pi_{33}$ ,  $(—)$   $T_{33}$ .

### 2.3.3. Spanwise Reynolds stress

In the ZPG budget for the spanwise Reynolds stress, shown in figure 7a, the values of the different terms are, as in the  $R_{22}$  budget, lower than in the Couette flow. The shapes of the profiles are similar to those in the Couette case. The pressure gradient enhances the values, but nothing else seems to be affected in the APG1 case. In APG2 however, the turbulent transport is of the same magnitude as the pressure-strain.

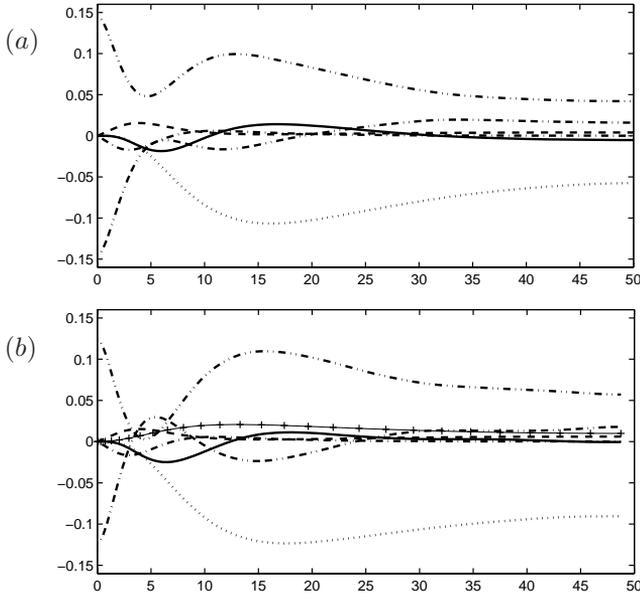


FIGURE 8. Terms in the Couette flow  $R_{12}$ -budget for (a) the non-rotating case and (b) the rotating case,  $\Omega = -0.005$ . The different terms are: ( $\cdots$ )  $\mathcal{P}_{12}$ , ( $--$ )  $-\varepsilon_{12}$ , ( $-\cdot-$ )  $D_{12}$ , ( $-\cdot\cdot-$ )  $\Pi_{12}$ , ( $-\cdot\cdot\cdot$ )  $G_{12}$ , ( $-$ )  $T_{12}$ , ( $+$ )  $C_{12}$ .

#### 2.3.4. Reynolds shear stress

The budget for the Reynolds shear stress in Couette flow is presented in figure 8. The pressure strain ( $\Pi_{12}$ ) and pressure diffusion ( $G_{12}$ ) balance each other at the wall. This is also the case in Mansour *et al.* (1988). The value of  $\Pi_{12}$  at the wall in Couette flow is more than twice the value found in the channel flow simulation Mansour *et al.*, and also for ZPG flow it is higher.

The budget for the Reynolds shear stress in boundary layer flow is presented in figure 9. The profiles are approximately the same as in the Couette case, except for the pressure-strain and pressure diffusion at the wall which shows larger values in the Couette case. The outer ( $y^+ > 5$ ) values are however the same in the two flows.

The outer peak (at  $y^+ = 6$ ) of the pressure-strain equals the value at the wall in ZPG and APG2 cases. In the weaker APG boundary layer, APG1, the outer peak has a lower value than at the wall.

#### 2.4. Near-wall behavior

There is a balance between dissipation and viscous diffusion on the wall. From the data in figures 2–6 we may also compute the dissipation rate anisotropies,

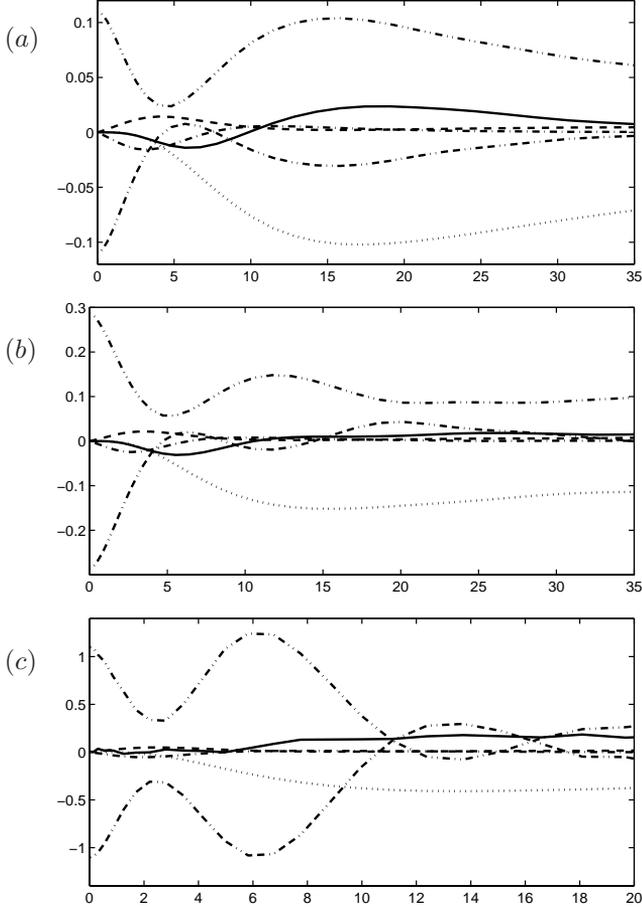


FIGURE 9. Terms in the  $R_{12}$ -budget for boundary layer flow (a) ZPG. (b) APG1. (c) APG2. The different terms are: ( $\cdots$ )  $\mathcal{P}_{12}$ , ( $--$ )  $-\varepsilon_{12}$ , ( $- \cdot - \cdot$ )  $D_{12}$ , ( $- \cdot \cdot \cdot -$ )  $\Pi_{12}$ , ( $- \cdot \cdot \cdot \cdot$ )  $G_{12}$ , ( $-$ )  $T_{12}$ .

$e_{ij} = \varepsilon_{ij}/\varepsilon - \frac{2}{3}\delta_{ij}$ . The limiting values of these (along with the stress anisotropies  $a_{ij} = R_{ij}/K - \frac{2}{3}\delta_{ij}$ ) are given in table 2 and compared with the predictions obtained by the algebraic dissipation rate anisotropy models of Hallbäck *et al.* (1990) and Sjögren & Johansson (2000). The agreement is quite satisfactory for both models in the Couette case, while the Hallbäck *et al.* model is in better agreement with DNS data for the ZPG boundary layer. In the Hallbäck *et al.* model  $e_{ij}$  is given by

$$e_{ij} = \left[ 1 + \alpha \left( \frac{1}{2} \Pi_a - \frac{2}{3} \right) \right] a_{ij} - \alpha (a_{ik} a_{kj} - \frac{1}{3} \Pi_a \delta_{ij}), \quad \alpha = \frac{3}{4}, \quad (9)$$

component	1,1	2,2	3,3
$a_{ij}$	0.72	$-\frac{2}{3}$	-0.05
$e_{ij}$	0.73	$-\frac{2}{3}$	-0.06
$(e_{ij})_{\text{Hallbäck}}$	0.67	$-\frac{2}{3}$	0.00
$(e_{ij})_{\text{Sjögren}}$	0.72	$-\frac{2}{3}$	-0.05

TABLE 2. Couette data: Limiting values for the stress anisotropies  $a_{ij}$  and dissipation rate anisotropies  $e_{ij}$ , and comparison with models.

component	1,1	2,2	3,3
$a_{ij}$	0.76	$-\frac{2}{3}$	-0.09
$e_{ij}$	0.45	$-\frac{2}{3}$	-0.38
$(e_{ij})_{\text{Hallbäck}}$	0.49	$-\frac{2}{3}$	0.16
$(e_{ij})_{\text{Sjögren}}$	0.76	$-\frac{2}{3}$	-0.09

TABLE 3. Boundary layer data: Limiting values for the stress anisotropies  $a_{ij}$  and dissipation rate anisotropies  $e_{ij}$ , and comparison with models.

whereas in the [Sjögren & Johansson](#) model we have

$$e_{ij} = \left(1 - \frac{1}{2}F\right)a_{ij}, \quad F = 1 - \frac{9}{8}(\text{II}_a - \text{III}_a). \quad (10)$$

In the above expressions we have introduced the two nonzero invariants of the anisotropy tensor,

$$\text{II}_a = a_{ij}a_{ji}, \quad (11)$$

$$\text{III}_a = a_{ij}a_{jk}a_{ki}. \quad (12)$$

The latter model gives  $e_{ij} = a_{ij}$  as limiting value in the two-component limit, such as on a solid wall. This describes the situation very accurately in the Couette case. One may note that for this extremely low Reynolds number the dissipation rate is highly anisotropic also at the centreline.

Some important limiting values at the wall are given in [table 4](#) and [5](#). The dependence of the Reynolds number in the boundary layer is strong as seen in [table 4](#). All the values increase for higher Reynolds number, but they do not reach the values of the Couette flow. Hence, one might argue that the Couette data constitute a high Reynolds number limit for the boundary layer.

The effect of the APG on the boundary layer is quite severe as seen from [table 5](#). All limit values are increased when the boundary layer is subject to an APG. The rotation in the Couette case has the opposite effect; all limit values decreases.

Case $Re_{\delta_*}$	$u_{rms}^+/y^+$	$v_{rms}^+/y^{+2}$	$w_{rms}^+/y^+$	$-\langle uv \rangle^+/y^{+3}$	$\varepsilon^+$
ZPG 539	0.385	0.0112	0.232	0.00099	0.203
ZPG 920	0.398	0.0119	0.252	0.00102	0.223
Couette	0.414	0.0135	0.268	0.00121	0.246

TABLE 4. Limit values for  $y^+ \rightarrow 0$ 

Case	$u_{rms}^+/y^+$	$v_{rms}^+/y^{+2}$	$w_{rms}^+/y^+$	$-\langle uv \rangle^+/y^{+3}$	$\varepsilon^+$
APG1 $\beta = 0.65$	0.476	0.0177	0.344	0.00181	0.346
APG2 $\beta = 5.0$	0.728	0.0470	0.764	0.00598	1.35
Couette $\Omega = -0.005$	0.387	0.0124	0.243	0.00093	0.238

TABLE 5. Limit values for  $y^+ \rightarrow 0$ 

### 2.5. Anisotropy tensor

The Reynolds stress anisotropy tensor  $a_{ij}$  has, as already mentioned above, two nonzero invariants,  $\text{II}_a$  and  $\text{III}_a$ . All anisotropic states can be represented in the anisotropy invariant map (Lumley & Newman 1977) which are bounded by the lines  $8/9 + \text{III}_a = \text{II}_a$  and  $6\text{III}_a^2 = \text{II}_a^3$ . They represent two-component and axisymmetric turbulence, respectively.

In figure 10a the AIM paths for both the non-rotating and rotating Couette cases are shown. Their main characteristics are the same as for the channel flow simulations of Moser *et al.* (1999). Close to the wall the turbulence is very near the two-component limit, approaching the one-component limit near the edge of the viscous sublayer. At  $y^+ \approx 8$  the AIM path turns towards the isotropic state. For the present cases the  $Re_\tau$  is so low that there is nearly no real log-layer in the profiles with corresponding agglomeration of points in the AIM, as observed in the higher- $Re$  channel flow simulations.

The AIM paths for the boundary layer flows are shown in figure 10b. The ZPG case is very similar to the Couette flow. There is some agglomeration of points at the end of the path which is  $y^+ \approx 150$  (for ZPG). The path for APG1 starts at a lower value of  $\text{III}_a$  and represents a lower degree of anisotropy than in the ZPG case. The end of the APG1-path is at  $y^+ \approx 100$ . The differences between ZPG and APG1 are not so large in comparison with the APG2 case, where the path starts in the lower left corner and represents much lower degrees of anisotropy than in the other cases. This is explained by the less structured turbulence in a strong APG boundary layer. The path for APG3 was terminated at  $y^+ \approx 50$ , and is similar to the anisotropy states from a backward-facing step, see Le & Moin (1992).

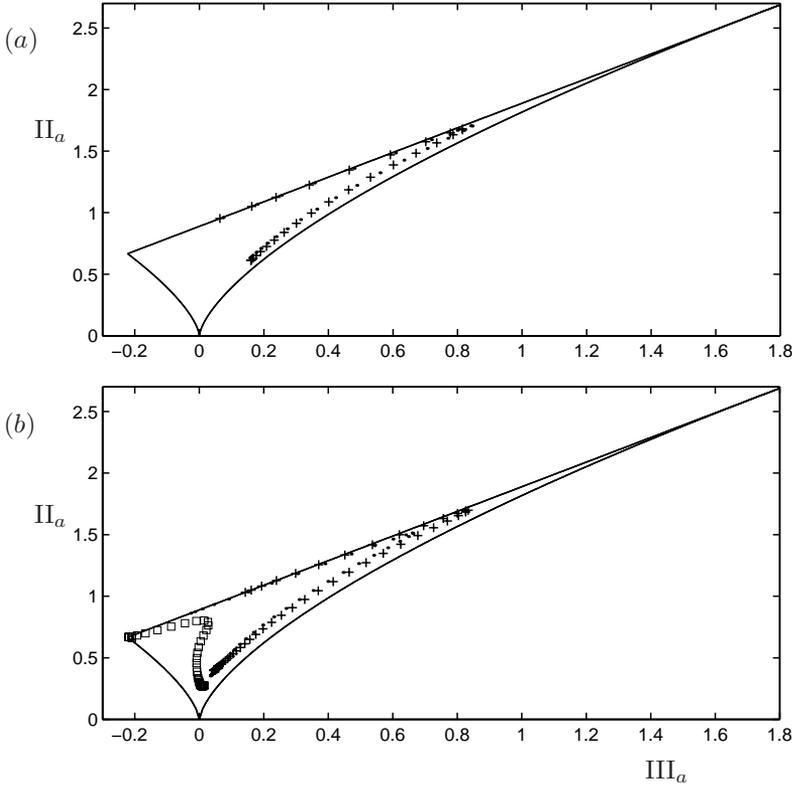


FIGURE 10. The anisotropy invariant map. a) AIM paths for the nonrotating (+) and rotating (·) case. b) AIM paths for ZPG (+) ; APG1 (·) ; APG2 (□).

### 2.6. Pressure-strain rate split

The results from a split of the pressure-strain rate is here presented for the Couette flow. The result from taking the divergence of the Navier–Stokes equation is a Poisson equation for the pressure,

$$\frac{\partial^2 p}{\partial x_i \partial x_i} = -\frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} (u'_i u'_j) - 2\epsilon_{ijk} \Omega_j^s \frac{\partial u'_k}{\partial x_i} \quad (13)$$

with the wall boundary condition,

$$\frac{\partial p}{\partial y} = \frac{1}{Re} \frac{\partial^2 v'}{\partial y^2} - 2U\Omega^s. \quad (14)$$

By splitting the source term in the Poisson equation into one part containing the mean velocity gradient and one part containing only gradients of the fluctuating part, we may derive equations for the rapid, slow and Stokes

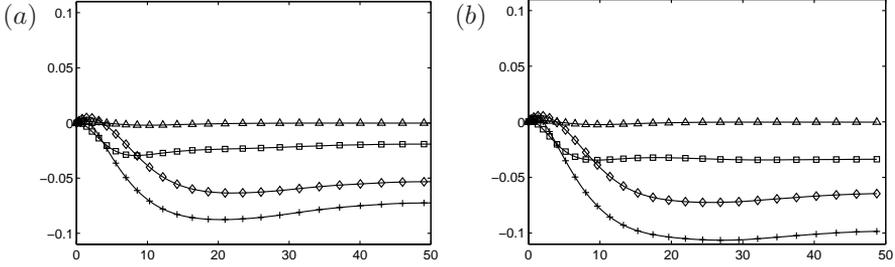


FIGURE 11. The  $\Pi_{11}$ -split for (a) the non-rotating case and (b) the rotating case,  $\Omega = -0.005$ . The different terms are: (+)  $\Pi_{11}^{(\text{tot})}$ , ( $\diamond$ )  $\Pi_{11}^{(s)}$ , ( $\square$ )  $\Pi_{11}^{(r)}$ , ( $\triangle$ )  $\Pi_{11}^{(\text{St})}$ .

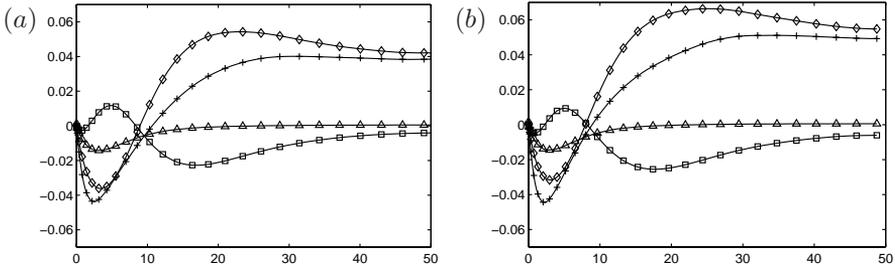


FIGURE 12. The  $\Pi_{22}$ -split for (a) the non-rotating case and (b) the rotating case,  $\Omega = -0.005$ . The different terms are: (+)  $\Pi_{22}^{(\text{tot})}$ , ( $\diamond$ )  $\Pi_{22}^{(s)}$ , ( $\square$ )  $\Pi_{22}^{(r)}$ , ( $\triangle$ )  $\Pi_{22}^{(\text{St})}$ .

pressure, respectively.

$$\nabla^2 p^{(r)} = -2 \left( \frac{\partial U_i}{\partial x_k} + \epsilon_{ijk} \Omega_j^s \right) \frac{\partial u_k}{\partial x_i}, \quad \frac{\partial p}{\partial y} = 0 \quad (15)$$

$$\nabla^2 p^{(s)} = -\frac{\partial u_i}{\partial x_j} \frac{\partial u_j}{\partial x_i}, \quad \frac{\partial p}{\partial y} = 0 \quad (16)$$

$$\nabla^2 p^{(\text{St})} = 0, \quad \frac{\partial p}{\partial y} = \frac{1}{Re} \frac{\partial^2 v}{\partial y^2} - 2U\Omega^s. \quad (17)$$

The Stokes pressure is solely due to the inhomogeneous boundary condition, and may be added to either the rapid or the slow pressure. Note that the last term in the boundary condition for the Stokes pressure is non-zero only for a moving wall, e.g. Couette flow. Restricting ourself to the present case of a channel with two homogeneous directions the rapid part simplifies further,

$$\nabla^2 p^{(r)} = -2 \frac{dU}{dy} \frac{\partial v}{\partial x} - 2\Omega^s \omega_3. \quad (18)$$

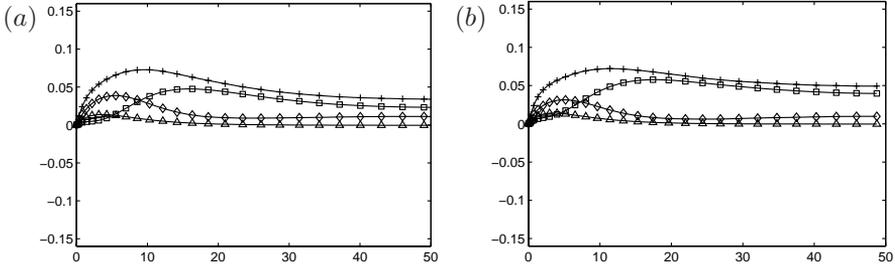


FIGURE 13. The  $\Pi_{33}$ -split for (a) the non-rotating case and (b) the rotating case,  $\Omega = -0.005$ . The different terms are: (+)  $\Pi_{33}^{(\text{tot})}$ , ( $\diamond$ )  $\Pi_{33}^{(\text{s})}$ , ( $\square$ )  $\Pi_{33}^{(\text{r})}$ , ( $\triangle$ )  $\Pi_{33}^{(\text{St})}$ .

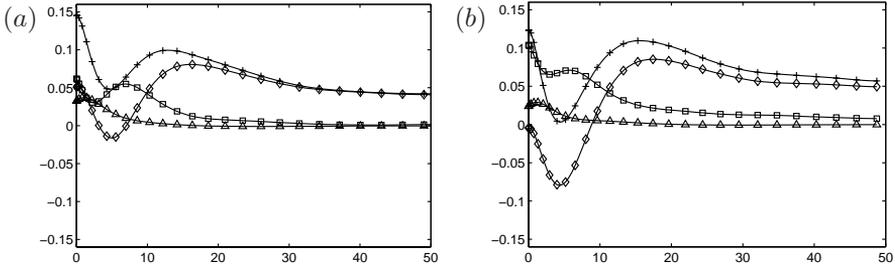


FIGURE 14. The  $\Pi_{12}$ -split for (a) the non-rotating case and (b) the rotating case,  $\Omega = -0.005$ . The different terms are: (+)  $\Pi_{12}^{(\text{tot})}$ , ( $\diamond$ )  $\Pi_{12}^{(\text{s})}$ , ( $\square$ )  $\Pi_{12}^{(\text{r})}$ , ( $\triangle$ )  $\Pi_{12}^{(\text{St})}$ .

The split into rapid, slow and Stokes pressure strain-rate can be seen in figure 11–14 for  $\Pi_{11}$ – $\Pi_{12}$ . The slow part of  $\Pi_{11}$  is larger than the rapid except near the wall,  $y^+ < 10$ , where the mean velocity gradient is large. The rapid part is more affected by the rotation than the slow part.

Also for the  $\Pi_{22}$ -term the slow part is larger than the rapid part, and contribute most to the pressure strain-rate. Here the slow part is more affected by the rotation.

For the  $\Pi_{33}$ -terms the rapid part contributes most, except for  $y^+ < 10$ , and is also most affected by the rotation.

The Stokes part for  $\Pi_{22}$ ,  $\Pi_{33}$  and  $\Pi_{12}$  is significant only in the region  $y^+ < 10$ , and for  $\Pi_{11}$  it is negligible throughout the channel.

The general character and amplitude of the various pressure strain rate terms are almost identical even for  $Re = 375$ , despite the low Reynolds number.

### 3. Summary

We have used the Couette flow simulation data of Komminaho *et al.* (1996) and the boundary layer data of Skote *et al.* (1998); Skote & Henningson (2000) to

compute terms in the transport equation for the Reynolds stresses. For the Couette flow we have also presented data for a split of the pressure strain rate term in rapid, slow and Stokes. Data was presented for both rotating (slow stabilizing rotation) and non-rotating Couette flow. One can see a small effect of the rotation on the limiting values at the wall in the Couette flow, but it is small as could be expected, since it is a very slow rotation. In the centre of the channel the budgets were strongly influenced by the rotation.

Boundary layer data were presented for one zero pressure gradient flow and two adverse pressure gradient flows. Strong influence on the budgets from the adverse pressure gradient were detected. The near-wall limits of turbulence statistics were shown to increase with Reynolds number in the zero pressure gradient boundary layer, but they did not reach the values obtained from the Couette flow.

### Acknowledgements

The authors would like to thank Arne Johansson for many valuable comments on the manuscript.

### References

- ARONSON, D., JOHANSSON, A. V. & LÖFDAHL, L. 1997 Shear-free turbulence near a wall. *J. Fluid Mech.* **338**, 363–385.
- BASDEVANT, C. 1983 Technical improvement for direct numerical simulation of homogeneous three-dimensional turbulence. *J. Comp. Phys.* **50** (2), 209–214.
- BECH, K. H. 1995 Simulation of rotating and non-rotating turbulent plane Couette flow. PhD thesis, Department of Applied Mechanics, Thermodynamics and Fluid dynamics, Norwegian Institute of Technology, Trondheim.
- BECH, K. H. & ANDERSSON, H. I. 1994 Very-large-scale structures in DNS. In *Direct and Large-Eddy Simulations I* (ed. P. R. Voke, L. Kleiser & J.-P. Chollet), pp. 13–24. Dordrecht: Kluwer.
- BECH, K. H., TILLMARK, N., ALFREDSSON, P. H. & ANDERSSON, H. I. 1995 An investigation of turbulent Couette flow at low Reynolds numbers. *J. Fluid Mech.* **286**, 291–325.
- CHING, C. Y., DJENIDI, L. & ANTONIA, R. A. 1995 Low-Reynolds-number effects in a turbulent boundary layer. *Exps. Fluids* **19**, 61–68.
- FERNHOLZ, H. H. & FINLEY, P. J. 1996 The incompressible zero-pressure-gradient turbulent boundary layer: An assessment of the data. *Prog. Aerospace Sci.* **32**, 245–311.
- GIBSON, M. M. & LAUNDER, B. E. 1978 Ground effects on pressure fluctuations in the atmospheric boundary layer. *J. Fluid Mech.* **86**, 491–511.
- GROTH, J. 1991 Description of the pressure effects in the Reynolds stress transport equations. *Phys. Fluids A* **3** (9), 2276–2277.
- HALLBÄCK, M., GROTH, J. & JOHANSSON, A. V. 1990 An algebraic model for non-isotropic turbulent dissipation rate term in Reynolds stress closures. *Phys. Fluids A* **2** (10), 1859–1866.

- HUNT, J. C. R. & GRAHAM, J. M. R. 1978 Free-stream turbulence near plane boundaries. *J. Fluid Mech.* **84**, 209–235.
- KIM, J., MOIN, P. & MOSER, R. 1987 Turbulence statistics in fully developed channel flow. *J. Fluid Mech.* **177**, 133–166.
- KOMMINAHO, J., LUNDBLADH, A. & JOHANSSON, A. V. 1996 Very large structures in plane turbulent Couette flow. *J. Fluid Mech.* **320**, 259–285.
- KOMMINAHO, J., LUNDBLADH, A. & JOHANSSON, A. V. 1997 Determination of the transition Reynolds number in plane Couette flow through study of relaminarization. In *First AFOSR Int. Conf. on DNS/LES* (ed. C. Liu & Z. Liu), pp. 233–240. Louisiana Tech University, Ruston, LA, USA: Grayden Press.
- KREISS, H.-O. & OLIGER, J. 1972 Comparison of accurate methods for the integration of hyperbolic equations. *Tellus* **24**, 199–215.
- LE, H. & MOIN, P. 1992 Direct numerical simulation of turbulent flow over a backward facing step. In *Annual Research Briefs*, pp. 161–173. Stanford Univ. Center for Turbulence Research.
- LUMLEY, J. L. & NEWMAN, G. R. 1977 The return to isotropy of homogenous turbulence. *J. Fluid Mech.* **82**, 161–178.
- LUNDBLADH, A. & JOHANSSON, A. V. 1991 Direct simulation of turbulent spots in plane Couette flow. *J. Fluid Mech.* **229**, 499–516.
- MANSOUR, N. N., KIM, J. & MOIN, P. 1988 Reynolds-stress and dissipation-rate budgets in a turbulent channel flow. *J. Fluid Mech.* **194**, 15–44.
- MOIN, P. & KIM, J. 1982 Numerical investigation of turbulent channel flow. *J. Fluid Mech.* **118**, 341–377.
- MOSER, R. D., KIM, J. & MANSOUR, N. N. 1999 Direct numerical simulation of turbulent channel flow up to  $Re_\tau = 590$ . *Phys. Fluids* **11** (4), 943–945.
- NA, Y. & MOIN, P. 1996 Direct numerical simulation of studies of turbulent boundary layers with adverse pressure gradient and separation. *Tech. Rep.* TF-68. Thermosciences Division, Department of Mechanical Engineering, Stanford University.
- NAGANO, Y., TAGAWA, M. & TSUJI, T. 1992 Effects of adverse pressure gradients on mean flows and turbulence statistics in a boundary layer. In *Turbulent Shear Flows 8*, pp. 7–21. Springer.
- ORSZAG, S. A. 1969 Numerical methods for the simulation of turbulence. *Phys. Fluids Suppl. II* **12**, 250–257.
- ORSZAG, S. A. 1970 Transform method for the calculation of vector-coupled sums: Application to the spectral form of the vorticity equation. *J. Atmosph. Sci.* **27**, 890–895.
- PEROT, J. B. & MOIN, P. 1995 Shear-free turbulent boundary layers. Part 1. Physical insights into near-wall turbulence. *J. Fluid Mech.* **295**, 199–227.
- SAHAY, A. & SREENIVASAN, K. 1999 The wall-normal position in pipe and channel flow at which viscous and turbulent shear stresses are equal. *Phys. Fluids* **11** (10), 3186–3188.
- SJÖGREN, T. & JOHANSSON, A. V. 2000 Development and calibration of algebraic nonlinear models for terms in the Reynolds stress transport equations. *Phys. Fluids* **12** (6), 1554–1572.
- SKOTE, M., HENKES, R. A. W. M. & HENNINGSON, D. S. 1998 Direct numerical

- simulation of self-similar turbulent boundary layers in adverse pressure gradients. *Flow, turbulence and combustion* **60**, 47–85.
- SKOTE, M. & HENNINGSON, D. S. 1999 Analysis of the data base from a DNS of a separating turbulent boundary layer. Center for Turbulence Research, Annual Research Briefs, 225–237.
- SKOTE, M. & HENNINGSON, D. S. 2000 Direct numerical simulation of a separating turbulent boundary layer. *J. Fluid Mech.* Submitted.
- SPALART, P. R. 1988 Direct simulation of a turbulent boundary layer up to  $Re_\theta = 1410$ . *J. Fluid Mech.* **187**, 61–98.
- SPALART, P. R. & WATMUFF, J. H. 1993 Experimental and numerical study of a turbulent boundary layer with pressure gradients. *J. Fluid Mech.* **249**, 337–371.
- TILLMARK, N. & ALFREDSSON, P. H. 1992 Experiments on transition in plane Couette flow. *J. Fluid Mech.* **235**, 89–102.

# Paper 3

P3



# Determination of the transition Reynolds number in plane Couette flow through study of relaminarization

By **Jukka Komminaho, Anders Lundbladh**  
and  
**Arne V. Johansson**

Royal Institute of Technology, Dept. of Mechanics, Stockholm, Sweden

Published in proceedings of the first AFOSR Int. Conf. on DNS/LES

By numerical simulation of relaminarization we have determined the transition Reynolds number of plane Couette flow in a manner that complements earlier studies. The approach adopted here was to start with a fully developed turbulent state and then decrease the Reynolds number until the turbulence vanished. We used a computational domain of two different sizes, and found that it must be very large in order to avoid artificial effects of the periodic boundary condition. It was hereby possible to establish, in accordance with recent transition studies, that the transitional Reynolds number in plane Couette flow is between 350 and 375, based on half the velocity difference and half the distance between the two surfaces.

---

## 1. Introduction

Plane Couette flow (PCF) represents one of the fundamental, canonical flow situations. It is linearly stable (see [Drazin & Reid \(1981\)](#)) for all Reynolds numbers, but from experiments we know that plane Couette flow becomes turbulent for sufficiently high Reynolds numbers. The transition from laminar flow to turbulence is thus subcritical. With this type of transition there exists both laminar and turbulent regions in the flow simultaneously, as the experiments in PCF by [Tillmark & Alfredsson \(1992, 1994\)](#) shows. Further investigations of turbulent spots in PCF that has recently been performed in numerical and physical experiments by [Lundbladh & Johansson \(1991\)](#) and [Daviaud \*et al.\* \(1992\)](#). All these investigations reveal that there is a transitional Reynolds number above which self-sustained turbulent spots can exist.

In the experiments by [Tillmark & Alfredsson \(1992, 1994\)](#) they used an apparatus with two moving walls, two running belts. They used two different approaches to disturb the flow, in their study. By introducing a large air bubble at the bottom of the channel they created a disturbance that developed to a turbulent spot for Reynolds numbers larger than 360. The other approach was

to start with a turbulent flow and gradually decrease  $Re$  until all turbulence vanishes. Also this approach gave a transitional  $Re$  of 360.

In the numerical experiments by Lundbladh & Johansson (1991) they studied the development of localized disturbances introduced in laminar Couette flow. They used several different Reynolds numbers and found that the disturbance died out for  $Re$  lower than 350, and for  $Re$  larger than 375 the disturbance developed to a turbulent spot.

Also Daviaud *et al.* (1992) found a transitional  $Re$  of about 370 in their experiment on PCF using a similar apparatus as Tillmark & Alfredsson.

In all the above investigations the authors have found a transitional  $Re$  of about 360–370. Some results by Dauchot & Daviaud (1995*a*) are somewhat contradictory to the above. Dauchot & Daviaud found that for very strong disturbances they could trigger a growing turbulent spot at  $Re$  of about 330. In the report by Dauchot & Daviaud (1995*b*) they found that by introducing a wire in the core region of the flow they generated streamwise vorticities at low Reynolds numbers (about 160). These streamwise vorticities vanished when they removed the wire. With the wire in the flow they found a transitional Reynolds number of about 340. This turbulent state did not die when they removed the wire.

Also, some earlier experiments of transition to turbulence have been carried out by Reichardt (1956) in an oil channel, and by Leutheusser & Chu (1971) in an air flow between a stationary wall and a moving water surface. Reichardt observed turbulence for Reynolds numbers above 750. Leutheusser and Chu observed turbulence above 280. Their apparatus had a small spanwise aspect ratio of about 12. This is rather small compared to the apparatus used by Tillmark & Alfredsson (1992). They reported that the moving water was turbulent and it is possible that roughness on the water surface may have an effect in lowering the value of the transitional Reynolds number.

Hamilton *et al.* (1995) made numerical experiments in a study of minimal channel Couette flow. They reduced the Reynolds number and the size of the computational domain to barely sustain turbulence. Thereby they could observe the breakdown and regeneration of the near-wall turbulent structures. This minimization of the computational domain will probably effect the transitional Reynolds number, since this minimal domain puts large constraints on the turbulence. Hamilton *et al.* (1995) found that for sustained turbulence they needed a Reynolds number of 400.

In the present study direct numerical simulation of the Navier–Stokes equations is used to study reverse transition, by successively lowering the Reynolds number, by increasing the viscosity, in a manner similar to Hamilton *et al.*, but with significantly larger channel. The lowest value for which turbulence can be sustained determines the transitional Reynolds number in a manner that complements and substantiates the results first obtained numerically by Lundbladh & Johansson (1991) and experimentally by Tillmark & Alfredsson (1992).

Simulation	Time	Reynolds number	Number of spectral modes
A	0–200	400	$85 \times 33 \times 85$
B	200–340	400	$170 \times 49 \times 170$
C	340–760	375	$170 \times 49 \times 170$
D	760–1300	375	$170 \times 49 \times 170$
E	760–1300	350	$170 \times 49 \times 170$

TABLE 1. The various simulations. The computational domain measures  $60 \times 2 \times 30$  half channel heights.

## 2. Numerical method

The simulation code in the present study uses spectral methods to solve the Navier-Stokes equations, with Fourier representation in the streamwise ( $x$ ) and spanwise ( $z$ ) directions, and Chebyshev polynomials in the wall-normal ( $y$ ) direction. The non-linear terms are treated pseudo-spectrally using FFT's, in a manner similar to that of Kim *et al.* (1987). The program was originally written in Fortran 77 to run on vector machines by Lundbladh *et al.* (1992), but has later been ported to run on a massively parallel machine, the Thinking Machines CM-200. In the present simulations aliasing errors from the evaluation of the nonlinear term were removed by the 2/3-rule. The time stepping scheme used is semi-implicit, with a third-order Runge–Kutta for the nonlinear term and a second order Crank–Nicolson for the linear term. The time step was dynamically kept at 90 % of the theoretical CFL limit. Periodic boundary conditions are used in  $x$  and  $z$  directions, and the no slip condition at the walls.

The initial field was chosen to be the sum of the laminar solution for Couette flow and random noise. The initially unphysical velocity field was allowed to relax to a turbulent state, and when the flow reached a statistically steady state the resolution was increased. The largest simulation used  $170 \times 49 \times 170$  spectral modes, or  $256 \times 49 \times 256$  physical grid points.

### 2.1. Computational domain and resolution

We used two different sizes of the computational domain. The larger computational domain was  $60 \times 2 \times 30$ , measured in half channel heights, in  $x$ ,  $y$ , and  $z$  directions, respectively, and the smaller was  $30 \times 2 \times 15$ . As we will see later the smaller box is probably too small to make a reliable investigation of the transitional Reynolds number. As a comparison the dimensions of the experimental apparatus in the study of Tillmark & Alfredsson (1992) was  $300 \times 2 \times 70$  half channel heights.

The periodic boundary conditions in the streamwise and spanwise direction are only justified, for this case where we want to approximate the turbulent state between infinitely large plates, if the two-point velocity correlation vanishes for the largest separations allowed by the computational domain. In the study of Hamilton *et al.* (1995) they used the periodic boundary condition in a

minimal channel at a low Reynolds number to obtain a less complex turbulent flow.

Because of the periodicity and the symmetry of the  $R_{uu}$  function the largest separation is half the box-length. The two-point correlation for the streamwise velocity is given by

$$R_{uu}(x) = \frac{\overline{u(x')u(x'+x)}}{\overline{u(x')^2}}$$

where only a streamwise separation  $x$  is considered, and an overbar denotes averaging over the  $x$ - $z$  plane in question, as well as over a period of time,  $T$ . If this correlation is significantly non-zero for half box-length separation, the largest eddies will be affected. For a more thorough discussion of the effects of a small box on the two-point correlation and turbulence statistics, see Komminaho *et al.* (1996).

The grid spacing is the same in both simulations. In wall units, the spacing in the streamwise and spanwise directions are,  $\Delta x^+ \approx 10$ , and  $\Delta z^+ \approx 5$ . In the wall normal direction the mean spacing is,  $\Delta y_{mean}^+ \approx 1.1$ . This is somewhat better than the well resolved simulation by Kim, Moin & Moser Kim *et al.* (1987). Note, that we use the convention that grid spacing is measured as  $\frac{x_l}{n_x}$ , where  $x_l$  is the length and  $n_x$  the number of spectral modes in one direction.

The resolution of the computation can be assessed by examining the one-dimensional velocity spectra. The spectra depicted in figure 1 show no evidence of energy pile-up at high wave numbers. The energy shows a decay of about 7 decades for the streamwise velocity. This altogether indicates that the resolution is adequate.

### 3. Results

The Reynolds number was initially 400 and the resolution  $85 \times 33 \times 85$  spectral modes. When the flow had reached a turbulent state, the resolution was increased to  $170 \times 49 \times 170$  spectral modes. The integration in time proceeded ( $tu_w/h$ ) 140 time units before the Reynolds number was lowered to 375. At this Reynolds number the flow remains turbulent, and the simulation of the flow continued to  $T = 760$  before the Reynolds number was further lowered to 350. At this Reynolds number the turbulence disappeared within 400 time units, as illustrated in figure 2(a) where the mean shear rate at the wall ( $S_w = \frac{dU}{dy}|_{wall}$ ) is shown as a function of time. In a large enough computational domain the mean shear rate should be constant for a constant Reynolds number. Because of the coexistence of relaminarized and turbulent regions the mean shear rate is not constant over time, even in the larger domain. Here the value varies between 1.7 and 2.1, or about 20 % for a Reynolds number of 375.

For comparison the laminar value of  $S_w = 1$  is shown in the figure 2. The value for the turbulent flow approaches the laminar value monotonically after the change of Reynolds number from 375 to 350. Four hundred time units later the two curves join and the flow has relaminarized completely.

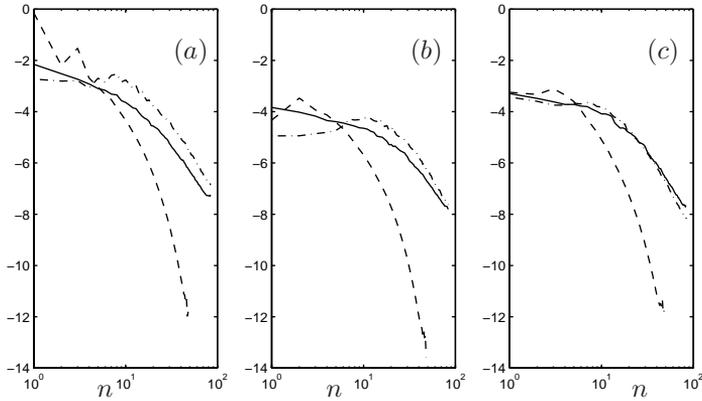


FIGURE 1. The one-dimensional velocity spectra for (a) streamwise velocity  $u$ , (b) wall-normal velocity  $v$  and (c) spanwise velocity  $w$ , at  $T = 1300$  and  $Re = 375$ . The solid line is in the streamwise direction, dashed line is the wall-normal direction and the chain-dashed line is the spanwise direction.

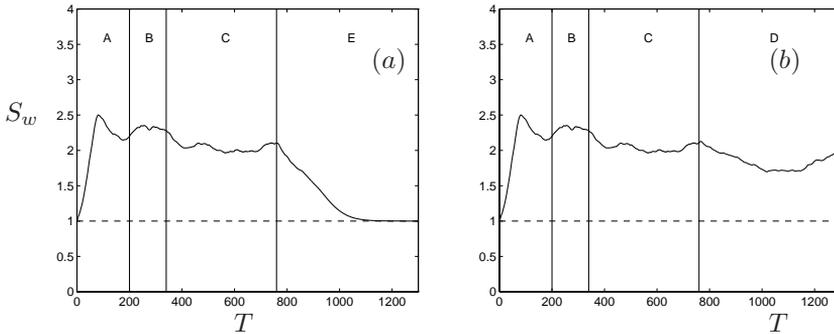


FIGURE 2. Mean shear rate, in the large box, at the wall ( $S_w$ ) versus time, for the turbulent flow (solid) and for a laminar flow (dashed). (a) shows relaminarization at  $Re = 350$ , and (b) shows sustained turbulence at  $Re = 375$ .  $S_w$  is taken from the simulations A, B, C, and E, respectively A, B, C and D. As described in table 1.

The relaminarization process is illustrated in figure 3(a)–(j). The viscous damping/dissipation rapidly kills the small scales. We are left at large times with longitudinal streaks, as we could expect because of the low damping for Orr–Sommerfeld modes with low streamwise wave-number. Locally we may also observe a dramatic, almost explosive, instability of the longitudinal streaks resulting in new small scale turbulence. This instability of the streaks is essentially identical to that observed in the study of bypass transition from

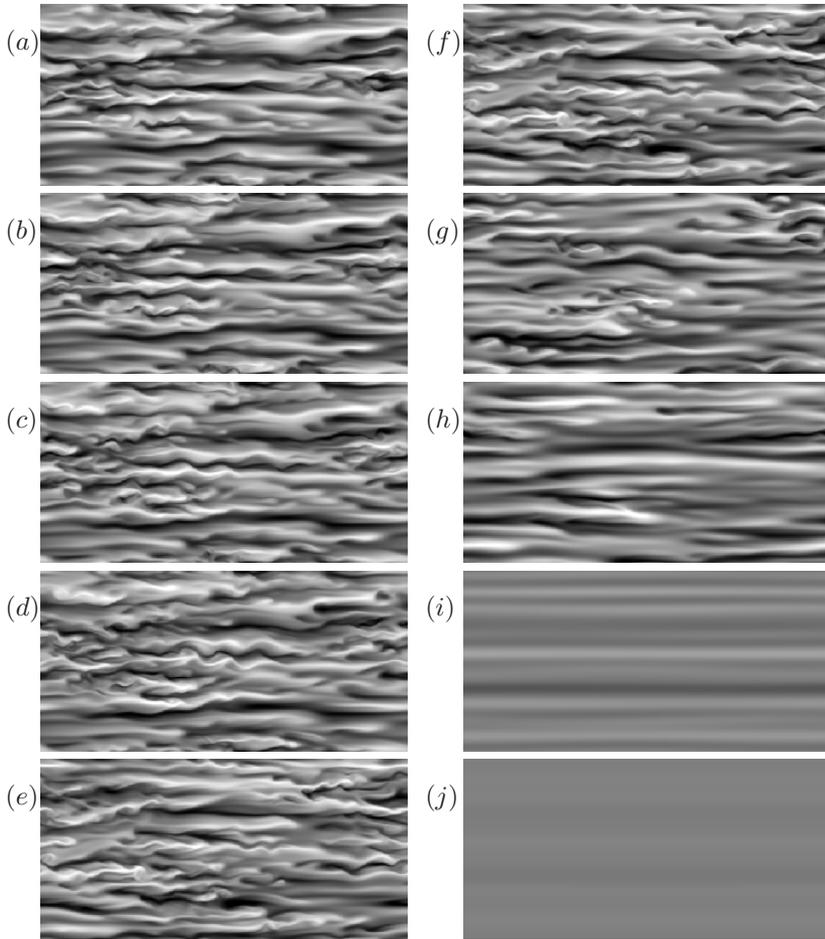


FIGURE 3. The instantaneous streamwise velocity field in an  $x$ - $z$  plane at  $y = 0$ , and at different times. in the large box. (a)  $T = 673.8$ , (b)  $T = 681.1$ , (c)  $T = 687.4$ , (d)  $T = 694.8$ , (e)  $T = 700.9$ , (f)  $T = 758.7$ , (g)  $T = 829.2$ , (h)  $T = 913.3$ , (i)  $T = 1075.1$ , and (j)  $T = 1259.9$ . In (a)–(e) the Reynolds number is 375 and in (f)–(j) 350.

localized disturbances by [Henningson \*et al.\* \(1993\)](#), and by [Kreiss \*et al.\* \(1994\)](#). It is also worth mentioning in this context that [Johansson \*et al.\* \(1991\)](#) observed intense local turbulence production in channel flow turbulence in connection with streaks that exhibited a development very similar to that seen here.

Also, we continued the simulation from  $T = 760$  with a Reynolds number of 375 to make sure that the flow remained turbulent at this Reynolds number. The simulation proceeded for the same amount of time as the simulation with

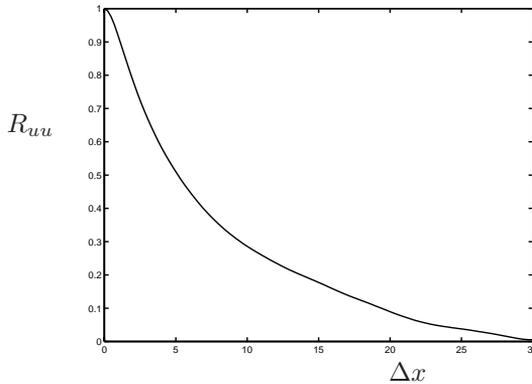


FIGURE 4. The two-point streamwise velocity correlation ( $R_{uu}$ ), in the large box, as a function of point separation  $\Delta x$ . The Reynolds number is 375 and the correlation curve is time averaged from  $T = 500$  to  $T = 1300$

the lower Reynolds number, or to  $T = 1300$  (see figure 2(b)). In this figure the curve is the same as in figure 2(a) until  $T = 760$ . The flow remained turbulent for the time simulated. The fact that the flow relaminarized at  $Re = 350$ , but not at 375 indicates that the transitional Reynolds number lies between 350 and 375, substantiating earlier investigations by Lundbladh & Johansson (1991), and Tillmark & Alfredsson (1992).

The two-point velocity correlation  $R_{uu}(x)$  is shown in figure 4 for the simulation in the large domain. The remaining correlation at half the box-length is sufficiently small for all turbulence statistics to be very well converged. For instance, based on the observations made by Komminaho *et al.* the deviations in  $u_{rms}$ -values from those of an infinite box can be judged to be less than 1 %.

The two-point velocity correlation for turbulent flow is one for zero separation and decreases with the separation, ultimately approaching zero as the separation goes to infinity. The maximum separation allowed by the computational domain is half the box length. The integral length scale  $\Lambda = \int_0^\infty R_{uu}(x)dx$  should be finite for turbulent flow and is found to be about 8 half channel heights in the present simulation for Reynolds number 375. Komminaho *et al.* reported a value of 6.1 for Reynolds number 750.

The two-point correlation in the small domain was considerably higher than in the large. At a separation of half the box length the two-point velocity correlation  $R_{uu}(\frac{x_l}{2} = 15)$  is as high as 0.3 for the small domain at a Reynolds number of 375. Note that this is considerably higher than the correct value at a separation of 15 (cf fig 4). Note also that the box needed here to avoid such effects is many times larger than that needed for a plane turbulent channel flow (see Kim *et al.* 1987).

#### 4. Conclusions

By numerical simulation of relaminarization we have determined the transition Reynolds number of plane Couette flow to be between 350 and 375 in good agreement with earlier results obtained by other means, (Lundbladh & Johansson 1991; Tillmark & Alfredsson 1992; Daviaud *et al.* 1992). The approach adopted here was to start with a fully developed turbulent state and then decrease the Reynolds number until the turbulence vanished. We decreased  $Re$  by increasing the viscosity, instead of decreasing the wall velocity. We thereby avoided the problem of the 'rolls' that would be created at the walls if the wall velocity is momentarily changed, Tillmark & Alfredsson (1994).

We made simulations in computational domains of two different sizes, viz.  $60 \times 2 \times 30$ , and  $30 \times 2 \times 15$  half channel heights. It was shown that the small box simulation suffers from substantial artificial effects of the periodic boundary condition, on the largest scales, giving a high value of the two-point velocity correlation at a separation of half the box-length. In the larger box this problem is minimal with a correlation at half box-length of 0.005–0.01. At Reynolds numbers just above the transitional one viscous damping of small scales is strong and regeneration of turbulence is seen to be coupled to an instability of the long streak structures. This instability is 'explosive' in character.

The observations of the turbulence regeneration process in these simulations are quite similar to those of Hamilton *et al.* (1995). It may also be noted that the streak instability is quite similar in character to the process involved in bypass scenarios and bears resemblance to the turbulence production processes associated with wall streaks in general.

#### References

- DAUCHOT, O. & DAVIAUD, F. 1995*a* Finite amplitude perturbation and spot growth mechanism in plane Couette flow. *Phys. Fluids* **7** (2), 335–343.
- DAUCHOT, O. & DAVIAUD, F. 1995*b* Streamwise vortices in plane Couette flow. *Phys. Fluids* **7** (5), 901–903.
- DAVIAUD, F., HEGSETH, J. & BERGÉ, P. 1992 Subcritical transition to turbulence in plane Couette flow. *Phys. Lett. Rev.* **69**, 2511–2514.
- DRAZIN, P. G. & REID, W. R. 1981 *Hydrodynamic stability*. Cambridge: Cambridge University Press.
- HAMILTON, J. H., KIM, J. & WALEFFE, F. 1995 Regeneration of near-wall turbulence structures. *J. Fluid Mech.* **287**, 317–348.
- HENNINGSON, D. S., LUNDBLADH, A. & JOHANSSON, A. V. 1993 A mechanism for bypass transition from localized disturbances in wall bounded shear flows. *J. Fluid Mech.* **250**, 169–207.
- JOHANSSON, A. V., ALFREDSSON, P. H. & KIM, J. 1991 Evolution and dynamics of shear-layer structures in near-wall turbulence. *J. Fluid Mech.* **224**, 579–599.
- KIM, J., MOIN, P. & MOSER, R. 1987 Turbulence statistics in fully developed channel flow. *J. Fluid Mech.* **177**, 133–166.

- KOMMINAHO, J., LUNDBLADH, A. & JOHANSSON, A. V. 1996 Very large structures in plane turbulent Couette flow. *J. Fluid Mech.* **320**, 259–285.
- KREISS, G., LUNDBLADH, A. & HENNINGSON, D. S. 1994 Bounds for threshold amplitudes in subcritical shear flows. *J. Fluid Mech.* **270**, 175–198.
- LEUTHEUSSER, H. J. & CHU, V. H. 1971 Experiments on plane Couette flow. *J. Hydr. Div., ASCE* **97**, 1269–1279.
- LUNDBLADH, A., HENNINGSON, D. S. & JOHANSSON, A. V. 1992 An efficient spectral integration method for the solution of the Navier–Stokes equation. *Tech. Rep. TN 1992-28*. FFA, Stockholm.
- LUNDBLADH, A. & JOHANSSON, A. V. 1991 Direct simulation of turbulent spots in plane Couette flow. *J. Fluid Mech.* **229**, 499–516.
- REICHARDT, H. 1956 Über die Geschwindigkeitsverteilung in einer geradlinigen turbulenten Couetteströmung. *Z. Angew. Math. Mech.* **36**, 26–29.
- TILLMARK, N. & ALFREDSSON, P. H. 1992 Experiments on transition in plane Couette flow. *J. Fluid Mech.* **235**, 89–102.
- TILLMARK, N. & ALFREDSSON, P. H. 1994 On Rayleigh instability in decaying plane Couette flow. *Appl. Sci. Res.* **53**, 187–196.



# Paper 4

P4



# Numerical simulation of the Navier–Stokes equations on massively parallel computers

By **Jukka Komminaho**

Royal Institute of Technology, Dept of Mechanics, Stockholm, Sweden

TRITA-MEK, Technical Report 1995:13

The efficient use of a parallel computer for the solution of the incompressible Navier–Stokes equation in the channel flow geometry is reported. The method used Fourier transforms in the two horizontal directions and a Chebyshev transform in the wall normal direction. The time stepping scheme is an explicit low storage third order Runge–Kutta or a second order Adam–Bashforth for the non-linear term and the implicit second order Crank–Nicolson for the linear term. Some issues concerning the performance of the code is discussed. Implementation of the code, as well as the performance on the Connection Machine model 200 is described in detail. A maximum performance of the code, on a CM200 with 16k processors, of about 600 Mflops in 64 bit arithmetics, and about 1 Gflop in 32 bit arithmetic was obtained. Some preliminary results from the implementation of the code on the SP2 from IBM is also discussed in some detail. The process of setting up an initial velocity field and the post processing of the simulation results is explained.

---

## 1. Introduction

Numerical simulations of turbulent fluid flow put heavy demands on the speed and memory of the computers running the simulation, and the requirements increase dramatically with the Reynolds number. The computational work increases as the Reynolds number to the power of three, which makes simulations of flows with high Reynolds numbers very time consuming.

In the last twenty years the computational capacity of the fastest available machines, the supercomputers, has increased roughly three thousand times, from the 100 Mflops of the one processor computer Cray-1 to the 280 Gflops of the multi processor computer Intel Paragon XP/S MP (6768 processors) [Dongarra \(1995\)](#). It is evident that this stunning development of computational power has, by no means, come to an end. There will in the future be more and more powerful computers available. The most powerful machines of today are parallel computers with many processors ( $> 100$ ) and we believe that this will be the case also, at least in the near future.

In order to use the computational power of these machines one often has to rewrite existing programs. In this report we will describe the rewriting of

a direct numerical simulation program initially running efficiently on vector computers such as Cray (Lundbladh *et al.* 1992), to run efficiently on parallel computers. We wrote the code initially for the Connection Machine 200, from Thinking Machines, and have recently started to investigate the possibility to port it to the SP2, from IBM. The code allows direct numerical simulation of turbulent flows in channel flow geometry, with or without moving walls and with possibility to superimpose a system rotation.

## 2. A short history of parallel computers

In the beginning of the 17th century, in 1614, Napier introduced the logarithm, Shurkin (1984). For the first time it became easy to do multiplication and division. All one had to do was to add or subtract logarithms, and this was easy done on instruments such as an abacus. In 1621 William Oughtred produced a device with two flat pieces of wood fixed so that they could slide against each other. On each piece there was a scale on which the numbers were arranged at a distance from the end relative to their logarithm. It was the slide ruler. This is an example of an analog computer. This also illustrates that people throughout the history have sought for ways to increase the computing capability of their minds.

The first to try to build a machine of a more general nature, i.e. one that could be controlled by a program, was Charles Babbage in 1840's, with his Analytical Engine. The design of the Analytical Engine is very much alike the computers of today. But, for several reasons, he never succeeded in completing the work. The most prominent being his inability to stop redesigning and begin manufacturing the machine.

In 1930 an engineer at MIT, Vannevar Bush, built an analog computer, a differential analyzer. It was a collection of shafts, gears, and wires, and measured movements and distances and performed computations with these measurements. He built a larger machine in 1935 with improved design. This machine weighed 100 tons and included 2000 vacuum tubes, several thousand relays, 150 motors, and about 200 miles of wires. This machine worked well and several duplicates of the machine were set up at different laboratories. But these machines were still analog computers.

IBM built a digital computer, an electro-mechanical machine called Automatic Sequence Controlled Calculator (ASCC), or Mark I, in 1944. It could multiply two twenty-three digits number in four seconds, by todays standards a very low speed. People realized that in order to press the performance to higher levels one must use electronics instead.

### 2.1. *The first electronic computers*

In Moore School, Mauchly and Eckert built the first digital electronic computer, ENIAC. It was finished in 1946, and consisted of over 17000 vacuum tubes, weighed over 30 tons, and consumed 140 kilowatts. It could perform a multiplication of two eight digit numbers in 2.6 milliseconds. Actually this

was a highly parallel machine (Burkes & Burkes 1981). It had 25 independent computing units (20 accumulators, 1 multiplier, 1 divider/square rooter, and 3 table lookup units). These units could each follow their own sequence of operations and cooperate towards the solution of a single problem. The architecture of the ENIAC was not fixed as in most modern computers, but could be rearranged for each problem by rewiring the connections between the different units. The algorithm was hardwired into the computer. But the programmers found the programming of the ENIAC in a parallel fashion difficult. In 1948 ENIAC was rewired to to work as a serial computer, to simplify the programming (Burkes & Burkes 1981). The time was clearly not ripe for this parallel architecture.

The first stored program machine was EDSAC. This machine had an architecture similar to many modern machines. This architecture is known as the von Neumann architecture. It consists of a memory unit for storage of data and instructions, a control unit, and an arithmetic unit. The path between the memory and the arithmetic and control unit is often referred to as the von Neumann bottleneck.

## 2.2. *The development of high performance computers*

Performance of a computer can be increased in essentially two ways, decrease the clock period, or increase the amount of work done in each clock period, or do work in parallel. Parallelism can be introduced in principally four different ways (Hockney 1985):

- Pipelining, which is the explicit segmentation of an arithmetic unit into different parts, each of which performs a sub-function on a pair of operands. For example, if a floating point multiplication is subdivided into five sections, each of which does one part of the multiplication, then five pairs of operands can be in the pipeline at a given time. Thereby increasing the performance fivefold, compared to an arithmetic units which accepts only one pair of operands and computes the result before accepting the next pair (Ortega 1989).
- Functional, providing several independent units for performing different functions, logic, addition, division, or multiplication, and allowing these to operate simultaneously on different data. Thereby increasing the performance as many times as there are functional units.
- Arrays, providing an array of identical processing elements under common control, all performing the same operation simultaneously on different data (this is called SIMD, see section 6.1).
- Multiprocessing, providing several processors, each under the control of their own instructions, operating on their own data (this is called MIMD, see section 6.1).

These can be, and have been, combined in the design of a computer. For instance, most modern high performing RISC processors are using the first two of the above. They have both pipelining and several functional units. It is

also possible to build SIMD or MIMD computers with these RISC processors, thereby combining three of the above four features.

In the following sections we will call SIMD or MIMD computer parallel computers, i.e. restricting the concept to machines with several processing units. Pipelining of arithmetic operations and several functional units can be thought of as parallelism within a single processing unit. The ENIAC was therefore not a parallel computer, but it was a computer with parallelism within the processing unit.

The first computers aimed for government agency and scientific laboratory, with their need of high performance number crunchers, came from Control Data Corp., or CDC. IBM had actually attempted to build such a machine, a tube operated machine called STRETCH, but the machine was a disaster, and IBM took a \$20 million loss, [Shurkin \(1984\)](#), when pulling the plug from the project.

The CDC6600 machine was the second machine that Seymore Cray built. It became a big success, being 20 times faster than any computer in the world. CDC 6600 was the first computer to employ functional parallelism. It was first delivered in 1964. It had 10 separate functional units for multiplication, division, long addition, shift, boolean, branch, and increment. It also had 10 independent peripheral processors, forming a link with slow input/output devices. In 1968 the CDC6600 was replaced by the CDC7600. It was four times faster because the clock cycle was reduced from 100 ns to 27.5 ns. The 10 serially organized functional units were replaced by 8 pipelined functional units and one serial unit which could not be pipelined.

After the fiasco with the STRETCH, IBM announced a series of computers, the 360, in 1964. but no machine was delivered until 1967. This computer, the IBM 360/91, had a performance of about twice that of CDC6600. It had a look-ahead facility, and separate execution units for floating-point and integer address calculations each of which was pipelined.

### 2.2.1. *Vector computers*

In 1964, a computer named STAR 100 was conceived at CDC. It was to be a processor for vectors, that would be pipelined and able to sustain performance of 100 Mflops on long vectors. The actual design started in 1967, but the computer was not delivered until 1973. The STAR 100 suffered from its old technology, it had magnetic-core memory, whereas other computers in 1973 had semiconductor memory, and the clock was slow with an 80 ns period. It did not become a success. It was, however, completely reengineered to use semiconductor memory. This new machine, the CYBER 205, was released in 1980.

The CYBER 205 consists of a vector unit and a scalar unit. The vector unit comprises up to four floating-point pipelines, each of which comprises five separate pipelined functional units. The peak performance of the four pipe-machine is 800 Mflops. The memory is divided into 8 sections, each section into

8 memory stacks and each memory stack into 8 memory banks. This machine works on vectors directly in the memory, not in special vector registers.

In 1983 the CDC founded a company, the ETA System Inc, and in 1986 they announced their machine ETA<sup>10</sup>. This machine has 8 processors with an architecture similar to the CYBER 205 but three times faster. It also has up to 18 I/O units, and a shared memory. Each processor is architecturally similar to a two-pipe CYBER 205. One ETA<sup>10</sup>-Q processor, with a clock period of 19 ns, has a peak performance of 210 Mflops. The company goal was a clock period of 5 ns and a peak performance for one processor of 800 Mflops, and this is almost met in the ETA<sup>10</sup>-G, with a clock period of 7 ns, and a peak performance of 570 Mflops.

In 1976 the first CRAY-1 computer was announced. It contained 12 functional units, all pipelined, a fast clock, 12.5 ns, and a 16-bank memory. The novel feature was the eight vector registers, capable of holding 64 64-bit floating-point numbers, and a set of machine instructions for manipulating and performing arithmetic on these vectors. It was the first pipelined vector computer to become a commercial success. It had a peak performance of about 160 Mflops, [Dongarra \(1995\)](#). In 1982 the CRAY X-MP was announced. It was a multiprocessor system of pipelined vector machines, with a shared memory (MIMD). The model X-MP/48 has a peak performance of 840 Mflops.

IBM did not enter the vector computer market until 1985, when the multiprocessor IBM 3090 (a part of the System/370 series) was announced. It had a scalar performance of about 5 Mflops and a vector performance about four times higher. Each processor of the 3090 can support one vector unit, and the maximum number of processors are six. The theoretical performance of the latest model, the IBM 3090/600J VF, with 6 processors and a clock period of 14.5 ns, is 828 Mflops.

### 2.2.2. *Parallel computers*

The performance level achievable by pipelining operations is limited. One can only subdivide an operation into a few number of subtasks. To achieve higher performance one must either use a faster clock, use more functional units or use several processors, performing work in parallel. The number of functional units that can be used in parallel is also limited, mainly because it is difficult to make programs that can use many units simultaneously in an efficient manner. To reach the Tera-flop limit, one must most certainly build parallel machines. Indeed, the most powerful supercomputers of today are also parallel machines.

When building a parallel computer one may either interconnect several computers each of which is capable of independent of each other perform computations (MIMD), or one may interconnect several simple processors, which are in turn are controlled by a master controller (SIMD). The latter machines are also called computer array. Example of the former is CRAY Y/MP (shared memory MIMD), and SP2 (local memory MIMD), whereas the CM200 is an example of the latter category.

In the above section on vector computers we have already seen examples of shared memory MIMD computers. We will here discuss a few other MIMD computers, as well as SIMD. The first MIMD machine was the Burroughs D-825, built in 1960, see the survey of multiprocessors by [Enslow \(1977\)](#). It was a shared memory machine, with up to four processors, 16 memory modules, and 64 I/O devices. The connection between the processors and the I/O devices was a crossbar switch matrix.

In the 60s a few other other multiprocessor machines were built, including Burroughs B-5000 with two processors and CDC 6500, a dual CDC 6400s system. In the 70s, though, the technology, and the integrated circuit had evolved to a stage where it became possible to build more complex multiprocessors, including computer arrays. The first machines were essentially research projects, but later commercial versions of several of the machines were built. The first idea for computers arrays appeared in a paper by [Unger \(1958\)](#). This machine was a two dimensional array with interconnected neighbors, all controlled by a single master controller. The idea of grid-connected computer was also pushed in a paper by [Slotnick \*et al.\* \(1962\)](#).

An early example of MIMD computer was the Carnegie–Mellons C.mmp which consisted of 16 DEC PDP-11 connected to 16 memory modules by a  $16 \times 16$  crossbar switch. Along the same lines another computer was projected, the S1. The complete design of the S1 would have consisted of 16 Cray-1 vector computers connected to 16 memory banks by a full cross-bar switch.

The first technically successful array processor was the ILLIAC IV, built by Burroughs. It consisted of 64 processors, arranged in a two-dimensional grid. The clock period was 80 ns, and the peak performance about 100 Mflops. Although it never reached the envisioned 1 Gflops for a machine with 256 processors, it still had a major influence on the development of algorithms for parallel machines and computer languages, [Hockney \(1985\)](#).

Burroughs went on building parallel machines. After the ILLIAC, they built PEPE (Parallel Element Processor Ensemble). PEPE consisted of loosely coupled system of 288 processing elements. Each processing element (PE) consisted of three different processors. The PEPE was designed to control a ballistic missile defense system, and the three processors in one PE had different task. One for input of radar signals, one for processing of data, and one for output of control signals. The three different types of processors were controlled by three master control units, and they in turn were controlled by a CDC7600. Since each of the PEs was to follow its own target the need for communication among the PEs was little. When necessary, the communication took place via the memories of the control units. Deliveries began in 1976.

ICL DAP (Distributed Array Processor) consists of 4096 1-bit processors arranged in a two-dimensional array ( $64 \times 64$ ). There is a single master controller unit which processed a single instruction stream and broadcasts a single command to all 4096 PEs. In addition to referencing its own memory each PE can reference data stored in the memory of its immediate NEWS (North, East

West and South) neighbors. Since the processors are 1-bit processors the speed of the arithmetic is very dependent of the word length, and the performance for 32-bit floating point multiply is about 16 Mflops.

The Delencor Heterogeneous Element Processor (HEP) is a computer that offers the facility of programming with multiple instruction streams. A full system consists of 16 Process Execution Modules (PEM) connected to 128 Data Memory Modules via a multi-stage packet switching network. Each PEM may process up to 50 instruction streams. These streams share an 8-stage instruction execution pipeline, and the execution of the streams are switched every clock cycle. Only one instruction from every instruction stream can be in the pipeline, so in order to reach the maximum performance the pipeline must be full i.e. at least eight instruction streams must be initiated. The peak performance of a full system is about 80 Mflops.

One computer with a more unusual network is the Erlangen General Purpose Array (EGPA) (see [Hockney \(1985\)](#) survey of MIMD computers in 1985). The nodes consists of AEG 80-60 minicomputers, and the interconnection network is topologically in the form of a pyramid. The control C-computer, at the top of the pyramid, controls four B-computers at the corner of its base. This configuration with five computers was built in 1981, but the idea is expandable to more levels.

Hypercubes as the interconnection network became popular after a paper by [Pease \(1977\)](#). The first hypercube built was the Cosmic Cube, at Cal Tech. They made a commercial derivate called Intel iPSC/1, (in 1985), which is a SIMD computer with a maximum of 128 nodes. Each node consists of an 80286 plus an arithmetic coprocessor, 80287, and 512 kbytes of memory, and the host computer is an Intel 310 minicomputer. The maximum peak performance of a full system is 8 Mflops.

Another company, N-cube, marketed a machine called Ncube/10. This machine could be expanded to a maximum of 1024 nodes, each node consisting of a custom VLSI chip containing CPU, FPU and the necessary communication channels. Each node also has 512 kbytes of memory, and the peak performance of the machine is 100 Mflops. The host that controls the machine is based on a 80286 processor.

The Connection Machine/2 is a machine with hypercube architecture, with up to 4096 nodes. Each node consists of 16 single-bit processors connected in a crossbar. Later models have an additional floating point unit per every 32 processor, see section [6.3](#) for more details.

### 3. More modern designs.

Experience has shown that the bottleneck in massively parallel computers is the communication. One must add additional communication hardware to each node in order to obtain a better balance between speed of calculation and communication. There are several examples of modern designs where this issue is addressed, examples are Meiko, which has an additional processor per node for

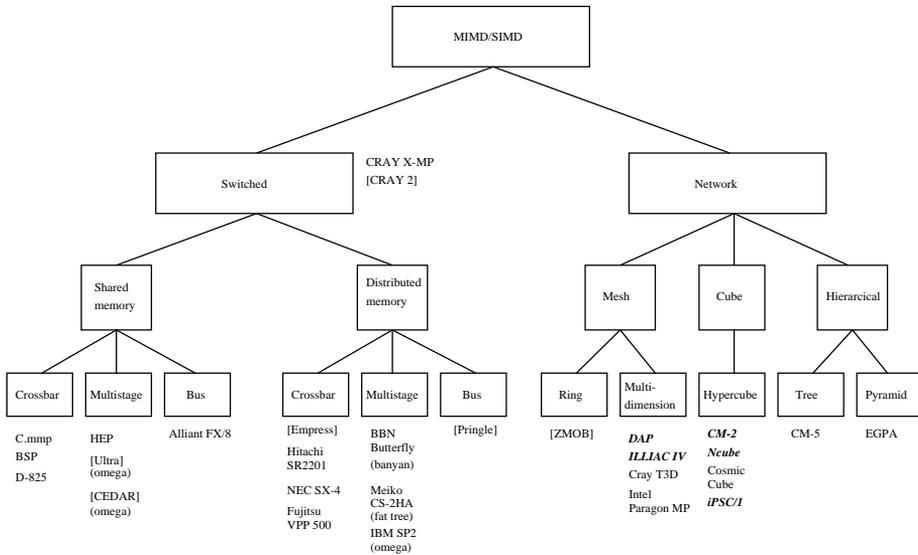


FIGURE 1. An overview of the different computers described in the text. SIMD machines are marked with bold italic. The type of multistage network is given in parenthesis. A description of computers within brackets can be found in [Hockney \(1985\)](#).

the communication, and Cray T3D, which has a direct memory access controller (BLT) on each node that works independently of the processor. Many of the modern parallel machines have fewer than 1000 processors, but the processors in turn are in some cases the most powerful processors that exist today.

Meiko CS-2HA is a MIMD machine with a multi-stage packet switch, fat tree network, in which the bandwidth between stages remains constant. The maximum number of nodes is 256, where each node consists of a SPARC scalar unit, a communication processor and two Fujitsu  $\mu$ VP vector units. This gives a performance of 200 Mflops per node, with a 20 ns clock cycle, and 51 Gflops for a full system.

Intel Paragon MP has three Intel i860 XP processors per node, one for the communication, and two for computations, giving a performance per node of 250 Mflops. The interconnecting network is a two-dimensional mesh.

Hitachi SR2201 is a massively parallel computer, with up to 1024 processors in a three-dimensional crossbar switch network, capable of 300 Gflops.

Cray T3D is a MIMD machine with a three dimensional torus network. Each node consists of two processors (from DEC), one block transfer engine (BLT) and a network interface. A full system consists of 2048 nodes, and has a peak performance of about 300 Gflops.

The CM500 is a more modern design than the model 200. CM500 has a fat tree interconnecting network with a maximum of 2048 nodes. Each node consists of a 40 MHz SuperSPARC microprocessor and four vector unit floating point and integer arithmetic accelerators, for a total per node performance of 160 Mflops and a total performance for a full system of 330 Gflops

NEC SX-4 is a machine with very powerful nodes, each node is capable of achieving a peak vector performance of 2 Gflops. Each node consists of a vector unit with eight pipelines, and a scalar unit capable of issuing two instructions per clock period. A full system comprises 512 nodes in a crossbar network and have a peak performance of 1 Tflops.

Fujitsu VPP500 is also a parallel machine with very powerful processing nodes, each node is in itself a supercomputer with a performance of 1.6 Gflops, and since a total of 222 nodes can be interconnected in a crossbar network, the full machine has a performance of 350 Gflops. Prior to VPP500 Fujitsu built a computer for National Aerospace Laboratory in Japan. This computer, the Numerical Wind Tunnel consists of 140 processors, each capable of 1.7 Gflops. We may here mention the performance of 60–70 Gflops obtained for a CFD code on the Numerical Wind Tunnel.

As we have seen, there are principally two design philosophies, either design a machine with many, not so powerful, processors, or a machine with few but very powerful processors. The early parallel computers had a few processing nodes, mainly because with the technology of that day one could not build extensive networks requiring tens of thousands of logical gates. But when the technology evolved it became possible to build more processors and more elaborate networks, and with the VLSI technique it became possible in the 80s to build thousands of processors at low costs. But these massively parallel computers with thousands of processors is very difficult to program efficiently, a typical program reaches only 5–20% of peak performance. Compared to the more conventional vector processors such as Cray X-MP, where it is not unusual to get about 40–50% of peak performance, this utilization is rather low. Also, the necessary system software (compilers etc) tend to become very complicated for the massively parallel machines. This tends to delay the software development and also sometimes makes them less user-friendly. To quote [Lincoln \(1977\)](#)

The resource and development time requirements for essentially simple software systems can exceed estimates by astronomical amounts when a radically new architecture is at hand.

Therefore it seems that most of the modern designs tend to use a moderate number (100–1000) of very powerful processors interconnected with a simple network such as two- or three-dimensional mesh. A further advantage with few processors is that the communication/calculation ratio is lower, thereby decreasing the demand on the speed of the interconnecting network.

#### 4. The governing equations

The equations governing fluid flow is the Navier–Stokes equations. For an incompressible fluid they can be written as:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} \quad \text{in } \Omega \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \quad (2)$$

$$\mathbf{u} = 0 \quad \text{at } \partial\Omega \quad (3)$$

where boldface is used to denote a vector, and  $\Omega$  is the domain of interest. The boundary conditions for the velocity is  $\mathbf{u} = 0$  at solid stationary boundaries. These are the momentum and continuity equations. If we introduce non-dimensional variables we can write the momentum equation as:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{R} \nabla^2 \mathbf{u} \quad (4)$$

where  $R$  is the Reynolds number,  $R = \nu/UL$ ,  $U$  is a typical velocity scale,  $L$  is a typical length scale and  $\nu$  is the viscosity. If we introduce a rotating coordinate system we can write the velocity in an inertial system as:

$$\mathbf{u}_I = \mathbf{u}_R + \Omega \times \mathbf{r}$$

and the corresponding material derivative becomes:

$$\left( \frac{D\mathbf{u}}{Dt} \right)_I = \left( \frac{D\mathbf{u}}{Dt} \right)_R + \Omega \times (\Omega \times \mathbf{r}) + 2\Omega \times \mathbf{u}_R$$

where the second term on the right hand side is the centrifugal term and the third term is the Coriolis term. Altogether, the non-dimensional momentum equation in a rotating system can be written as (dropping the subscript R):

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p - \Omega \times (\Omega \times \mathbf{r}) - 2\Omega \times \mathbf{u} + \frac{1}{R} \nabla^2 \mathbf{u}. \quad (5)$$

Since the centrifugal force acts as an extra pressure term we can include this term in the pressure term and introduce a 'reduced' pressure  $p'$ . Also, if we rewrite the convective term as:

$$(\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \left( \frac{1}{2} \mathbf{u} \cdot \mathbf{u} \right) + (\nabla \times \mathbf{u}) \times \mathbf{u}$$

we can write the Navier–Stokes equation as:

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla p' - \nabla \left( \frac{1}{2} \mathbf{u} \cdot \mathbf{u} \right) + \mathbf{u} \times (\nabla \times \mathbf{u} + 2\Omega) + \frac{1}{R} \nabla^2 \mathbf{u} \quad \text{in } \Omega \quad (6)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \quad (7)$$

$$\mathbf{u} = 0 \quad \text{at } \partial\Omega. \quad (8)$$

The unknowns in the above equation system are the velocities and the pressure. But the treatment of the pressure is difficult, since, unlike the velocities, there is no evolution equation for the pressure. Instead, it is determined by the continuity equation. We will avoid this difficulty by rewriting the pressure-velocity formulation into a velocity-vorticity formulation where the pressure

does not appear explicitly. By taking the divergence of the momentum equation (6) we obtain a Poisson equation for the pressure:

$$\nabla^2 p = \nabla \cdot \mathbf{H} - \nabla^2 \left( \frac{1}{2} \mathbf{u} \cdot \mathbf{u} \right) \quad (9)$$

where  $\mathbf{H} = -(\nabla \times \mathbf{u}) \times \mathbf{u} - 2\boldsymbol{\Omega} \times \mathbf{u}$ . Applying the Laplace operator on the momentum equation (6), and substituting the pressure with equation (8) and using the continuity equation (7) one obtains an equation for the velocities without the pressure:

$$\frac{\partial \nabla^2 \mathbf{u}}{\partial t} = \nabla^2 \mathbf{H} - \nabla(\nabla \cdot \mathbf{H}) + \frac{1}{R} \nabla^4 \mathbf{u}. \quad (10)$$

This equation can be split into a system of two equations:

$$\frac{\partial \phi}{\partial t} = \mathbf{h}_v + \frac{1}{R} \nabla^2 \phi \quad (11)$$

$$\nabla^2 \mathbf{u} = \phi \quad (12)$$

where

$$\mathbf{h}_v = \nabla^2 \mathbf{H} - \nabla(\nabla \cdot \mathbf{H}). \quad (13)$$

An equation similar to the the one above can be obtained for the vorticity by taking the curl of the momentum equation:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \mathbf{h}_\omega + \frac{1}{R} \nabla^2 \boldsymbol{\omega} \quad (14)$$

where

$$\mathbf{h}_\omega = \nabla \times \mathbf{H}. \quad (15)$$

So far we have written the equations in general coordinates, but in the numerical algorithm we use Cartesian coordinates. Therefore we will write down the equations in Cartesian coordinates. Also, we can define the flow studied to be channel flow, i.e. flow between two plates of infinite extent in the  $x$ - and  $z$ -direction. The domain  $\Omega = (-\infty, \infty) \times (-1, 1) \times (-\infty, \infty)$ , and with boundary  $\partial\Omega = (y = \pm 1)$ .

It is sufficient to calculate the normal velocity and the normal vorticity, because the other velocity components can be obtained from the incompressibility constraint and the definition of the normal vorticity. The reason for choosing to solve the equations for the normal velocity and vorticity is that continuity is automatically satisfied and pressure is eliminated from the problem. The equations to be solved are for the normal velocity:

$$\frac{\partial \phi}{\partial t} = h_v + \frac{1}{R} \nabla^2 \phi \quad (16a)$$

$$\nabla^2 u = \phi \quad (16b)$$

$$v(\pm 1) = \frac{\partial v}{\partial y}(\pm 1) = 0 \quad (16c)$$

where

$$h_v = \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} \right) H_2 - \frac{\partial}{\partial y} \left( \frac{\partial H_1}{\partial x} + \frac{\partial H_3}{\partial z} \right) \quad (17)$$

and for the normal vorticity:

$$\frac{\partial \omega}{\partial t} = h_\omega + \frac{1}{R} \nabla^2 \omega \quad (18a)$$

$$\omega(\pm 1) = 0, \quad (18b)$$

where

$$h_\omega = \frac{\partial H_1}{\partial z} - \frac{\partial H_3}{\partial x}. \quad (19)$$

## 5. Spatial and temporal discretization

The spatial discretization is a spectral collocation method with Fourier series expansion in the homogeneous directions and Chebyshev series in the inhomogeneous  $y$ -direction. The use of Fourier series assumes periodicity which may be justified provided that the length scales of the flow are substantially smaller than the extent of the computational domain (Canuto *et al.* 1988). The computational domain here is  $\Omega = (-x_L/2, x_L/2) \times (-1, 1) \times (-z_L/2, z_L/2)$ . The collocation points in the  $x$  and  $z$ -direction are:

$$x_l = lx_L/N_x, \quad l = -N_x/2, \dots, N_x/2 - 1 \quad (20)$$

$$z_m = mz_L/N_z, \quad m = -N_z/2, \dots, N_z/2 - 1 \quad (21)$$

where  $N_x$  and  $N_z$  are the number of collocation points in the  $x$  and  $z$ -direction respectively. The independent variables have discrete Fourier expansions of the form:

$$\mathbf{u}(x, y, z, t) = \sum_{l=-N_x/2}^{N_x/2-1} \sum_{m=-N_z/2}^{N_z/2-1} \hat{\mathbf{u}}_{l,m}(y, t) e^{\alpha_l x + \beta_m z} \quad (22)$$

where  $\alpha_l = 2\pi l/x_L$ , and  $\beta_m = 2\pi m/z_L$ , and  $N_x - 1$  and  $N_z - 1$  are the number of Fourier modes in each direction. The collocation points in the  $y$ -direction are:

$$y_n = \cos \frac{\pi n}{N_y}, \quad n = 0, 1, \dots, N_y. \quad (23)$$

In the  $y$ -direction the velocity has the following discrete Chebyshev polynomial representation:

$$\hat{\mathbf{u}}_{l,m}(y, t) = \sum_{n=0}^{N_y} \tilde{\mathbf{u}}_{l,m,n}(t) T_n(y). \quad (24)$$

The time discretization is chosen as a semi-implicit scheme. The linear part of equation (16a) is discretized implicitly using the second order accurate Crank–Nicolson scheme and the nonlinear (advective and rotation) terms explicitly by either the second order Adam–Bashforth or a low storage third order

	$a_n/\Delta t^n$	$b_n/\Delta t^n$
Euler	1	0
AB2	$1 + \Delta t^n/2\Delta t^{n-1}$	$-\Delta t^n/2\Delta t^{n-1}$
RK3	8/15	0
3-stage	5/12	-17/60
	3/4	-5/12
RK3	8/17	0
4-stage	17/60	-15/68
	5/12	-17/60
	3/4	-5/12

TABLE 1. Time-stepping coefficients.

three or four stage Runge–Kutta scheme. The C–N scheme is A-stable, i.e. it is absolutely stable in the entire left-half plane, [Strikwerda \(1989\)](#). The time discretization will take the following form (applied on equation [16a](#))

$$\phi^{n+1} = \phi^n + a_n h_v^n + b_n h_v^{n-1} + \frac{(a_n + b_n)}{2R}(\phi^{n+1} + \phi^n) \quad (25)$$

where  $a_n$  and  $b_n$  are dependent on the explicit scheme used. The different coefficients are given in table [1](#). For a discussion on the stability properties of the time stepping see [Lundbladh et al. \(1992\)](#).

If we now expand the dependent variables in equations [\(16a\)–\(18a\)](#) in Fourier series and apply the above time stepping we obtain the following equations for the normal velocity and vorticity:

$$\left(1 - \frac{a_n + b_n}{2R}(D^2 - k^2)\right) \hat{\phi}^{n+1} = \left(1 + \frac{a_n + b_n}{2R}(D^2 - k^2)\right) \hat{\phi}^n + a_n \hat{h}_v^n + b_n \hat{h}_v^{n-1} \quad (26)$$

$$(D^2 - k^2)\hat{v}^{n+1} = \hat{\phi}^{n+1} \quad (27)$$

$$\hat{v}(\pm 1) = D\hat{v}(\pm 1) = 0 \quad (28)$$

$$\left(1 - \frac{a_n + b_n}{2R}(D^2 - k^2)\right) \hat{\omega}^{n+1} = \left(1 + \frac{a_n + b_n}{2R}(D^2 - k^2)\right) \hat{\omega}^n + a_n \hat{h}_\omega^n + b_n \hat{h}_\omega^{n-1} \quad (29)$$

$$\hat{\omega}(\pm 1) = 0, \quad (30)$$

where D denotes a derivative in the normal direction, and  $k^2 = \alpha^2 + \beta^2$ . This can be written in a more compact form as

$$(D^2 - \lambda^2)\hat{\phi}^{n+1} = \hat{f}_v^n \quad (31)$$

$$(D^2 - k^2)\hat{v}^{n+1} = \hat{\phi}^{n+1} \quad (32)$$

$$(D^2 - \lambda^2)\hat{\omega}^{n+1} = \hat{f}_\omega^n \quad (33)$$

where

$$\lambda^2 = k^2 + 2R/(a_n + b_n) \quad (34)$$

$$\hat{f}_\psi^n = \hat{p}_\psi^n - \frac{2Ra_n}{a_n + b_n} \hat{h}_\psi^n \quad (35)$$

and

$$\begin{aligned} \hat{p}_\psi^n &= - \left[ D^2 - \lambda^2 + \frac{4R}{a_n + b_n} \right] \hat{\psi}^n - \frac{2Rb_n}{a_n + b_n} \hat{h}_\psi^{n-1} \\ &= - \hat{f}_\psi^{n-1} - \frac{4R}{a_n + b_n} \hat{\psi}^n - \frac{2Rb_n}{a_n + b_n} \hat{h}_\psi^{n-1}. \end{aligned} \quad (36)$$

The quantity  $\hat{p}_\psi^n$  denotes the partial right hand side of the equations, where  $\psi$  denotes  $v$  or  $\phi$ .

This gives us the normal velocity and vorticity. From these and the continuity equation we can find the remaining velocity and vorticity components. One finds that

$$\hat{u} = \frac{i}{k^2} (\alpha D \hat{v} - \beta \hat{\omega}) \quad (37)$$

$$\hat{w} = \frac{i}{k^2} (\alpha \hat{\omega} + \beta D \hat{v}) \quad (38)$$

for the velocities, and

$$\hat{\chi} = \frac{i}{k^2} (\alpha D \hat{\omega} - \beta \hat{\phi}) \quad (39)$$

$$\hat{\vartheta} = \frac{i}{k^2} (\alpha \hat{\phi} + \beta D \hat{\omega}) \quad (40)$$

for the vorticities. This gives the velocity and vorticity components for all wave numbers except for wave number zero. This wave number needs special treatment, see [Lundbladh \*et al.\* \(1992\)](#).

The remaining direction to be expressed in discretized form is the  $y$ -direction. Using equation (24) to expand, for instance, the equation (31), and furthermore using the orthogonality property of the Chebyshev polynomials, we find the following relation between the coefficients

$$\tilde{\phi}_j^{(2)} - \lambda^2 \tilde{\phi}_j = \tilde{f}_j \quad j = 0, \dots, N_y \quad (41)$$

where  $\tilde{\phi}_j^{(2)}$  denotes the second derivative. We will solve for the coefficients of the Chebyshev series for the second derivative. One can relate the Chebyshev coefficients of the derivatives to the coefficients of the function. Using this relation one can rewrite the equation (41) to yield a pentadiagonal equation system with the two top rows filled by the boundary condition ([Lundbladh \*et al.\* 1992](#); [Canuto \*et al.\* 1988](#)). This equation system decouples into two three-diagonal systems with the top row filled.

Thus, the solution of the wall normal velocity,  $v$ , in each time step, consist of the following steps:

1. Calculate the nonlinear term  $H$ , by inverse transformation of the velocity and vorticity components, find  $H$  in physical space from equation  $H_i = \epsilon_{ijk} u_j (\omega_k + 2\Omega_k)$ ,  $i = 1, 2, 3$ , and transform  $H$  back to spectral space.
2. Calculate  $(D^2 - \lambda^2)\hat{\phi}^{n+1} = \hat{f}^n$ :
  - (a) Chebyshev transform  $\hat{H}_i$  and calculate  $\hat{h}_v$  according to equation (17).
  - (b) Calculate  $\hat{f}$  according to equation (35).
  - (c) Set up the matrix and solve the equation system for  $\hat{\phi}^{n+1}$ .
3. Finally, calculate  $(D^2 - k^2)\hat{v}^{n+1} = \hat{\phi}^{n+1}$ :
  - (a) Set up the matrix and solve the equation system for  $\hat{v}^{n+1}$ .

In addition, the wall normal vorticity is found by a similar procedure. When the normal velocity and vorticity is found it is easy to find the other velocity and vorticity components from equations (37)–(40).

### 5.1. A remark on aliasing

The equations (31)–(33) are all in the Fourier space, where the non-linear terms become convolution sums. These sums can be evaluated in an efficient way by transforming the velocities and the vorticities to physical space, where they become pointwise operations. This transformation is done with the fast Fourier transform, FFT.

When evaluating the convolution sum by Fourier transforming to the physical space and then back to the spectral space one must be careful not to introduce aliasing errors. We may illustrate the problem by introducing the above described discrete transform on two variables

$$U_j = \sum_{k=-N/2}^{N/2-1} \hat{u}_k e^{ikx_j} \quad j = 0, 1, \dots, N-1 \quad (42)$$

$$V_j = \sum_{k=-N/2}^{N/2-1} \hat{v}_k e^{ikx_j} \quad j = 0, 1, \dots, N-1 \quad (43)$$

and define the pointwise operation in physical space

$$W_j = U_j V_j \quad j = 0, 1, \dots, N-1$$

and the inverse transform

$$\widehat{W}_k = \frac{1}{N} \sum_{j=0}^{N-1} W_j e^{-ikx_j} \quad k = -\frac{N}{2}, \dots, \frac{N}{2} - 1 \quad (44)$$

where

$$x_j = 2\pi j/N. \quad (45)$$

Use the discrete transform orthogonality property to obtain

$$\widehat{W}_k = \sum_{m+n=k} \widehat{u}_m \widehat{v}_n + \sum_{m+n=k \pm N} \widehat{u}_m \widehat{v}_n = \widehat{w}_k + \sum_{m+n=k \pm N} \widehat{u}_m \widehat{v}_n. \quad (46)$$

The second term on the right hand side is the aliasing error. In other words, evaluating the convolution sum by Fourier transforming the variables  $u$  and  $v$  to physical space, do a pointwise multiplication, and the transform back to spectral space leads to an extra term not present when doing the convolution directly. To remove this extra term there are essentially two different techniques, cf. [Canuto \*et al.\* \(1988\)](#). One is based on the fact that if one uses  $M$  rather than  $N$  points in the discrete transform when evaluating the convolution sum, and  $M > \frac{3N}{2} - 1$ , then the error in the convolution sum will be zero.

There are also another technique for the dealiasing. This aliasing removal is based on phase shifts. One transforms on a grid shifted by a factor  $\Delta$  in physical space. If one evaluates the convolution sum twice, with one evaluation on a grid shifted by half a grid cell ( $\Delta = N/\pi$ ), then the aliasing error will be totally eliminated, [Canuto \*et al.\* \(1988\)](#). This at the cost of two evaluations of the convolution sum. There is another method of greatly reducing the aliasing error by using random phase shifts instead of the above  $N/\pi$ , and only evaluate the convolution sum once. This will reduce the aliasing error to  $(\Delta t)^2$  times the aliasing error. This was observed by [Rogallo \(1977\)](#), and is also described in [Canuto \*et al.\* \(1988\)](#).

But it is not necessary to use dealiasing. If one wants to reduce the aliasing error one could increase the resolution. According to [Canuto \*et al.\*](#) an aliased calculation will yield as acceptable an answer as a dealiased one, if the resolution is sufficient.

When comparing the number of flops required to obtain a certain error, with and without dealiasing [Lundbladh \(1993\)](#) finds that for high resolutions it may even be more efficient not to use dealiasing in  $x$ - and  $z$ -directions. On a massively parallel computer, the truncation of 1/3 of the grid points can be done in two ways: either just zeroing 1/3 of the matrix, which implies a performance loss of 1/3 (the operations on the zero elements is never used), or reshaping the matrix to a smaller one, which implies communication, and thereby a reduction in performance. Thus, both these methods yields a performance loss. The method described by [Rogallo](#) could be used as well, and without performance loss, but because of the simplicity of the 2/3 dealiasing method we decided to implement the 2/3 dealiasing method, rather than the method by [Rogallo \(1977\)](#).

## 6. Implementation of the numerical algorithm

### 6.1. Parallel computers in general

There are a number of things that differ from one parallel computer to another, control of the processors, local or shared memory and the interconnection scheme. Flynn (1966) proposed a taxonomy based on instruction and data stream. This yields the following four categories:

- SISD, or Single-Instruction-Single-Data defines a serial computer.
- MISD, a Multiple-Instruction-Single-Data would involve multiple processors applying different instructions on a single datum, a hypothetical possibility.
- SIMD, in a Single-Instruction-Multiple-Data all processors are under the control of a master processor, and all the individual processors carry out the same operation. There is a single instruction stream operating on multiple data. An example of a SIMD type machine is the Connection Machine 200.
- MIMD, in a Multiple-Instruction-Multiple-Data the individual processors run their own program. This allows for more flexible ways of programming, but also introduces the problem of synchronization. In a SIMD system this synchronization is carried out by the master processor, but in a MIMD system other mechanisms must be used to make sure that the processors are working in the correct order with the correct data. Examples of MIMD type machines are the SP2 from IBM, and the Power Challenge, from Silicon Graphics.

See also the survey of parallel computer architectures by Duncan (1988), where the authors discuss not only the above taxonomy, but also other important aspects of parallel architectures.

Another important distinction in parallel computers is shared or local memory. In a shared memory system all processors have access to a common memory, whereas in a local memory system all processors have their own memory. In shared memory systems all communication between individual processors is through the memory. This gives potentially very rapid communication of data between processors. But if more than one processor wants to access the common memory simultaneously, one processor must wait until the memory is free again, the access is delayed. This delay, which is called congestion time, depends on the memory bandwidth. With this type of machines there is an upper limit to how many processors the system can have. Usually the memory bandwidth puts an upper limit of 16–32 processors. An example of a MIMD with shared memory is Power Challenge from Silicon Graphics.

In local memory system each processor can address only its own memory. The machines can be classified according to different programming models. One is message-passing, where communication between processors take place by sending messages between processors, in which data or code are transferred. The communication is like package-switching, every processor in the system

has an address, and routing information in a message guides it through the network to its destination. This routing of messages is generally worked out by the programmer as the program is written. This need for explicitly coding the communication into the program tends to complicate the use of these machines, and some vendors have tried some to hide the communication from the user. The effect is that the user essentially sees only one global memory as in shared memory machines. This approach is sometimes called data parallel programming model.

The advantage of this approach is that the user does not have to bother with the details of communication and thus the programs tend to become less complex. One disadvantage though is that the difficult task of coding high performing programs on parallel machines may become even more difficult since the user does not have explicit control over the communication.

How the individual processors communicate with each other is another important aspect. There are a number of interconnection schemes, but here we will mention only two, the hypercube and the switching network.

The processors in an  $n$ -dimensional hypercube can be visualized as the  $2^n$  vertices of the cube. Each processor is connected to its  $n$  adjacent vertices, along the edges of the cube. The hypercube has a number of interesting theoretical properties, of which one is that it is well suited for implementing fast Fourier transforms (see Leighton (1992)), by embedding a butterfly network on the hyper cube. The communication pattern is well suited for the hyper cube architecture. One drawback with the architecture is that the number of connections to each processor grows logarithmically with the size of the network, which can present some difficulties for large machines. One example on a machine with a hypercube network is Connection Machine 200

In a switching network the processors are interconnected through a series of switches, and the switches are connected in some network, a popular one being the omega network.

## 6.2. Parallel work and speedup

The degree of parallelism of a numerical algorithm is the number of operations that can be done in parallel. In the present problem this is typically very high. Almost all of the operations in the algorithm lies in the various transforms and in the three diagonal matrix solvers and they can all be done in parallel. The number of transforms that can be done in parallel is  $n_1 n_2$  if the transform is done in the  $n_3$  direction. Furthermore, there is a certain degree of parallelism in each transform. For an  $N$ -dimensional transform performed on an  $\log_2 N$ -dimensional butterfly network with  $N$  processors, this parallelism is equal to  $N$ , Leighton (1992). Thus, the transform takes only  $\log_2 N$  steps, as opposed to the transform on one processor which takes  $N \log_2 N$ . The solving of the three diagonal matrices, one for each  $x$ - $z$  point can also be done

in parallel. The only operations that cannot be parallelized are some operations performed on wave number zero (the wave number zero needs special treatment, see [Lundbladh \*et al.\* \(1992\)](#) for further details).

One can define the speedup of a parallel algorithm as

$$S_p = \frac{\text{execution time for a single processor}}{\text{execution time using } p \text{ processors}}.$$

With this definition the speedup of the above transform would be  $N$ . But in reality it is not that simple, the need for communicating the data across the different processors adds time to the parallel algorithm. Also, the amount of serial work as opposed to parallel work slows the algorithm down.

Taking the above effects into account, we can write the speedup as

$$S_p = \frac{T_1}{(\alpha + (1 - \alpha)/p)T_1 + t_d}$$

where  $T_1$  is the time for the algorithm on a single processor,  $\alpha$  is the amount of serial work,  $(1 - \alpha)$  is the amount of parallel work with average parallelism  $p$ , and  $t_d$  is the total time for communication. The total communication time depends on the amount of data being transferred and on the bandwidth of the network. It also depends on the latency, the time it takes for the network and the processors to get ready to transmit data. A crude model for the total communication time can then be written as

$$t_d = \tau_n + m/b_n$$

where  $\tau_n$  is the latency,  $m$  the total number of bytes transferred and  $b_n$  is the network's bandwidth.

We are now able to estimate the possible speedup of the present algorithm on a parallel machine with  $p$  processing units. The total number of operations that can be done with average parallelism  $p$  in each iteration of the algorithm is approximately  $110n_1n_2n_3 + 9 \cdot 2.5n_1n_2n_3 \cdot \log_2(n_1n_3) + 9 \cdot 2.5n_1n_2n_3 \cdot \log_2(n_2 - 1)$ . The amount of serial work done on one single processing unit per iteration is about  $48n_2$ .

As an example we can take the parallel machine to be the CM200 with 128 processing units (FPU's). This machine has a network bandwidth  $b_n$  of 700 Mbytes/s for simple communication patterns (circular shift, see [Helin \(1992\)](#) for a discussion of communication and communication models for the CM200). But the communication in the FFT's are not as fast as this. Measurements made by [Johnsson \*et al.\* \(1992\)](#) suggest that the performance of an entirely local FFT on a CM200 with 2048 FPU's is about 10 Gflops, or about half the peak performance. But the speed of a distributed three dimensional FFT on a  $512^3$  array is merely 1.2 Gflops. The difference between the two operations are the communication time. This can be estimated to be about 12 seconds for the three dimensional FFT. The amount data transferred in an FFT is about  $n^3 \log_2 n$ . This gives a bandwidth of about 800 Mbytes/s for the CM200 with

2048 FPU's. If we repeat this calculation to obtain the bandwidth on a CM200 with 128 FPU's we end up with a bandwidth  $b_n$  of about 420 Mbyte/s.

The total amount of bytes being transferred in each iteration of the algorithm is about  $m = 9 \cdot 8 \cdot n_1 n_2 n_3 \log_2 n_3$  for 9 variables in double precision (8 bytes) with the  $n_3$ -axis as the distributed axis.

The time for the algorithm on a single processing unit must also be estimated in some manner. One simple estimate is to take the total number of operations for one iteration and divide by 20 % of the peak performance of one processing unit, and thereby obtain the time for one iteration on a single processing unit. The figure 20 % is somewhat randomly chosen, but tests made by [Lundblad et al.](#) suggests that it is a typical performance value on workstations for the algorithm, it is also a typical value obtained on the CM200 when running Fortran code, see [Helin \(1992\)](#). But, since the used FFT's has a performance of about half the peak performance, when entirely local, we must take that into account. The local three diagonal matrix solvers and the local Chebyshev transform requires about 45 % of the total work and the FFT's about 55 %. This gives a value of  $T_1 = 410$  s/iteration for a problem with dimensions  $128 \times 129 \times 128$  in the  $n_1$ ,  $n_1$  and  $n_3$  directions respectively. The total communication time  $t_d = m(p-1)/(p \cdot b_n) = 2.5$  s/iteration if we ignore latency. We can now proceed to calculate the speedup (neglecting the amount of operations done serially), as

$$S_p = \frac{410}{(1/128) \cdot 410 + 2.5} = 70.$$

As we shall see this estimated speedup has the right order of magnitude although it's not entirely correct.

### 6.3. Connection Machine model 200

The Connection Machine model 200, (in the following denoted by CM200), is a SIMD (Single Instruction Multiple Data), ([Connection machine technical manual 1991](#)). It has implemented data parallelism. Objects in data parallel languages are represented by high level data types such as the array object of Fortran 90. Parallel data, that is, arrays, are allocated to processing elements (PE). Since many data sets are larger than the number of PEs available, the system uses a virtual processing mechanism, whereby each physical processor simulates some number of virtual processors. The ratio between physical and virtual processors is called the virtual processor ratio. The virtual processor concept makes the size of the physical machine transparent to the programmer.

#### 6.3.1. The physical processor and FPU

The basic unit of the CM200 is an integrated circuit consisting of 16 small processors and a routing device for communication. The 16 processors and the router are etched on a single chip, which are arranged in pairs. Each pair shares a group of memory chips with 4 Mbit memory and a optional floating-point unit, see figure 2. There are 2048 pairs of such chips in a fully configured

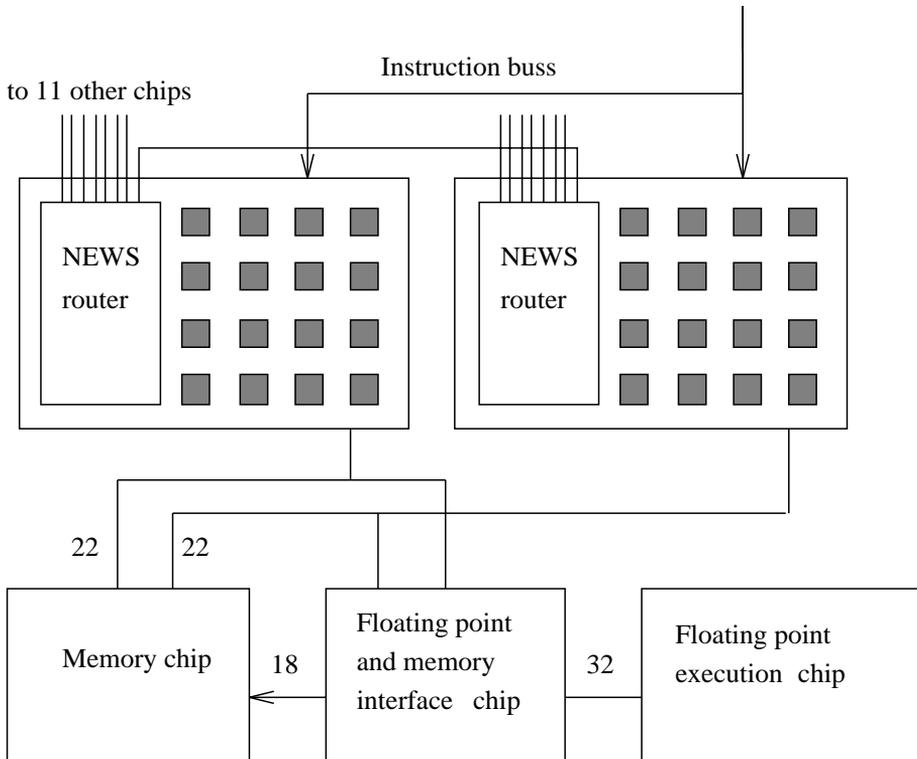


FIGURE 2. The processor chip

CM200, making a total of 65536 processing units. Thus the CM200 has a total memory of 8 gigabyte. The 16 processors on each chip are interconnected by a switch making it possible to create a direct connection between any pair of processors. The router on each chip is connected to 12 other routers in a 12-dimensional hyper cube, allowing communication with any processor in only 12 communication steps. Each pair of chips with its FPU is a CM200 processing node. The CM200 operates at 10 MHz. Theoretically, the FPU is able to produce two 32-bit results per cycle giving it a peak performance of 20 Mflops. With a full machine with 2k FPUs, or 64k processors, the peak performance is about 40 gigaflops in 32-bit arithmetic, or 20 gigaflops in 64-bit arithmetic.

The CM200 needs a host computer to control the instruction stream, the so called front-end. When running a program on the CM200 the front-end computer executes the program, performs any serial operations on scalar data stored in its own memory, and sending any instruction for parallel operations on parallel data to the CM200. The peak performance of the front end is typically on the order of a few thousandth of the CM200. This means that even a small proportion of serial operation, in a program, can reduce the speed of the program severely. But the front end is capable of doing serial work

while the CM200 is busy doing parallel work, so if the serial work is spread out evenly among the parallel, the impact on the performance does not have to be too severe.

### 6.3.2. Address field and array mapping

The address field of the CM200 is divided into three parts: (off-chip | on-chip | memory). The off-chip field encode the CM200 processor chips, the on-chip field encodes the 16 processors on each chip, and the lowest order encodes the memory address local to a processor. The standard data allocation scheme is consecutive storage, where the CM200 first determines how many data elements there are to each processor, then stores that many successive elements at each processor. This can be illustrated as:

$$\underbrace{(x_{m-1}x_{m-2} \cdots x_{m-n})}_{rp} \underbrace{(x_{m-n-1}x_{m-n-2} \cdots x_0)}_{vp}$$

where the field denoted  $rp$  encodes real processor addresses and the field  $vp$  encodes memory addresses.

There are three different ways of allocating data on the CM200:

1. `:news` order, maps array elements via the use of gray codes to allow fast NEWS (North-East-West-South), or grid, communication.
2. `:serial` order, in which data is allocated to a single processor.
3. `:send` order, maps the array elements on to the processing elements according to the binary representation of the array element's integer label.

For multidimensional arrays, each axis is encoded separately according to the description above, but first ignoring any serial axis, cf. Sabot (1991). Serial axes are encoded last, by simply making them local on every processor. The compiler tries to allocate the arrays so that each axis will have a segment of each address field, although this can also be controlled through the use of the compiler directive `cmf$ layout`. To modify this default allocation, weights can be specified to each axis. The weights are used to control the swapping of off-chip hyper-cube address bits between different parallel axes. The weights are interpreted in a linear fashion, that is, if axis  $x$  has a weight two times that of  $z$ , the compiler will try remove twice as many off-chip bits from the former axis, assuming that there are other axes that can accept these bits. This means that a heavily weighted axis may reside entirely on-chip.

### 6.3.3. Data layout in memory

In order to get good performance of the program the data allocation in memory is crucial. On-chip communication is much faster than inter-chip communication, because on-chip communication is a local memory reference, whereas inter-chip communication is slowed down by the limited bandwidth at the chip boundary. Obviously, one should try to allocate data in such a manner as to minimize inter-chip communication. However, the transforms used accesses

	serial axis first Mflops	serial axis last Mflops
a=b+c	120	120
forall a=b+c	10	10
doloop a=b+c	120	0.5
d=e+f	20	20
forall d=e+f	10	10
doloop d=e+f	120	0.5
a=e+f	4.0	4.0
forall a=e+f	3.7	3.7
doloop a=e+f	7.5	0.08

TABLE 2. The measured performance of adding two double precision (128,128,128) arrays on a 4k CM200.

data from positions far apart along the axis of transformation, and since there are Fourier transforms in two directions, and a Chebyshev transform in the remaining direction, essentially all the data must be communicated in each three dimensional transform.

#### 6.3.4. FFT performance tests and array layout

The CMSSL documentation stated that the FFT would yield best perform if the data was allocated according to `:send` or `:serial` ordering. Due to memory considerations, only one axis can be completely local on a processor. To find out more about what impact the data allocation has on performance, we tried several different ways of allocating data and measured the performance of each.

First, we decided to make the  $y$ -axis serially ordered. The reason for this is that we have a lot of code with loops over the  $y$ -axis, i.e. the Chebyshev transforms and the three-diagonal matrix solvers, and CM Fortran performs best when no communication is necessary. The communication pattern in an FFT is complicated, but regular, and the CMSSL routines exploit the hyper-cube architecture fully, making them very efficient. The allocation of the axes is now clear, `:send` for  $x$ - and  $z$ -axes, and `:serial` for  $y$ -axis.

To determine the impact on performance of the order of the axes, and if there are any performance differences when writing the loops with the HPF (High Performance Fortran) extension `forall` compared to Fortran 90 and Fortran 77 syntax, we timed addition of two  $(128 \times 128 \times 128)$  double precision arrays with one serial axis and two send axes and with the above mentioned different ways of expressing a do-loop. Furthermore, to check the compilers ability to handle automatic and explicitly declared arrays, we timed the loops with both static and automatic arrays. The arrays `d`, `e` and `f` in table 2 are automatic arrays, and `a`, `b` and `c` are explicitly declared arrays. The `forall` in the table reads:

```
forall (y=1:128) a(y, :, :) = b(y, :, :) + c(y, :, :)
```

array size (128,128,128) array layout	Single precision		Double precision	
	[s]	Mflops	[s]	Mflops
<code>:serial, :send, :send</code>	0.936	78	1.62	45
<code>:serial, 10:send, :send</code>	0.395	185	0.653	112
<code>:serial, 100:send, :send</code>	0.395	185	0.650	113

TABLE 3. Performance of real to half-complex FFT, with the second axis as the conjugate symmetric axis, on a 4k CM200

array size (128,128,128) array layout	Single precision		Double precision	
	[s]	Mflops	[s]	Mflops
<code>:serial, :send, :send</code>	0.976	75	1.67	44
<code>:serial, 10:send, :send</code>	1.20	61	2.13	34
<code>:serial, 100:send, :send</code>	1.20	61	2.13	34

TABLE 4. Performance of real to half-complex FFT, with the third axis as the conjugate symmetric axis, on a 4k CM200

and the do loop reads:

```
do y=1,128; a(y,::)=b(y,::)+c(y,::); enddo.
```

The operations in the table are all equivalent and should take about the same time. From the table we can deduce that the compiler, cmf 2.0, does not yield high performance when the serial axis is last, or automatic arrays are used in conjunction with explicitly dimensioned ones. In addition it was found that `forall` loops over serial dimensions are undesirable.

According to Sabot (1991), and our own tests, all serial dimensions must precede all parallel dimension for best performance. But the issue of different weights on the axes is still not solved. Table 3 gives performance for real to half-complex FFT, with the second axis as the real axis packed into a complex, and table 4 gives the performance with the third axis as the conjugate symmetric axis. The second and third axes are transformed. The arithmetic operation count is calculated from  $2.5n_1n_2n_3 \log_2(n_2n_3)$ , where  $n_1$ ,  $n_2$  and  $n_3$  are the lengths of the respective axes. Notice that the performance for single precision data is about 60 % higher than for double precision data. According to Johnson *et al.* (1992) this difference is due to the fact that the data path between each floating point unit and its memory is 32-bits wide. Data paths internal to the floating point unit are 64-bits wide.

From these tables it is clear that the preferable layout is `10:send, or 100:send`, for the conjugate symmetric axis, to obtain the highest possible performance of the FFTs.

According to our tests we should use `:serial, 10:send, :send` in the program. This means that for a sufficiently large problem (see further section 6.5) the data will be allocated on the CM200 according to figure 3, where the  $y$ -axis is local for one processor, the  $x$ -axis is entirely on-node, and the only

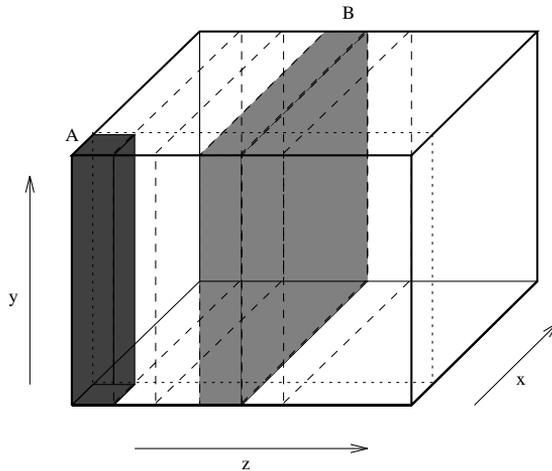


FIGURE 3. The data allocation on the CM200

axis off-node is the  $z$ -axis. A in the figure shows the data allocated on a single processor, and the slice B shows the amount of data allocated on one processor node. There are 128 such slices in a 128 cube problem, one for each processing unit in a 4k CM200. Compare figure 2, where one processor is a gray box, and the whole figure represents a processor node.

#### 6.4. The IBM SP2

The SP2 from IBM is a MIMD type of machine with distributed memory. The machine is built on a 66 MHz POWER2 architecture processor node capable of up to 266 Mflops. The nodes are of two types, 'thin' or 'wide'. The thin nodes is similar to a RISC System/6000 Model 390, and the wide node to a Model 590, Tengwall (1994). The main difference between the two types of nodes is the memory bandwidth, which on the wide node is 2 Gbytes/s. Another difference is the data cache size, thin nodes have data cache of 32 kbytes, and wide nodes have 256 kbytes. This means that for most programs the wide node has a better performance. The POWER2 nodes have double FXUs (Fixed eXecution Units) and FPUs (Floating Point Units), see also figure 4. Up to six instructions, or eight operations may be carried out simultaneously. Four of which are floating point operations. This means that for the processors running at 66.6 MHz, a peak performance of 266 Mflops is possible. The interconnection network is a multistage switched omega network providing a minimum of four path between any pair of nodes.

We have made a preliminary investigation of how our code would be implemented to run efficiently on the SP2. Our first thought was that we would only have to take the code, written in `cmf`, from the CM200, and essentially recompile it with the beta version of IBMs HPF compiler. Naturally we would have to replace the `cmf`, and CMSSL, specific code, such as the `layout` directives, and

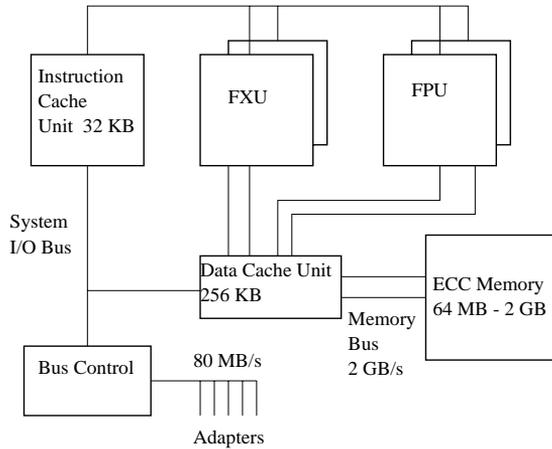


FIGURE 4. SP2 wide node

the FFT routines, with HPFs corresponding directive `distribute`, and IBMs ESSL (Engineering and Scientific Subroutine Library) FFT routines. Unfortunately the beta version of the HPF compiler was performing badly, and the ESSL did not have any parallel FFT routines. We have therefore concentrated on running the code efficiently on one node, using IBMs Fortran 90 compiler (`x1f`), and using the serial ESSL FFT routines.

In order to get reasonable performance for the final parallel version of the program it is of utmost importance to get good performance on one node. We tested a few routines, the Chebyshev transform, on the SP2, and realized that the memory access pattern was totally wrong for this type of cache based machine. The problem was the following: we had explicit do-loops over the  $y$ -axis, and array syntax over the  $x$  and  $z$  axes, and on the CM200 the direction with the explicit do-loop must be the first dimension in the array (since it is the serially ordered direction), this means that the arrays was declared as `(ny,nx,nz)` to suit the Connection Machine. The `x1f` compiler essentially expands expressions like:

```
do y=1,ny
  a(y, :, :)=b(y, :, :)*c(y, :, :)
enddo

to

do y=1,ny
  do z=1,nz
    do x=1,nx
      a(y,x,z)=b(y,x,z)*c(y,x,z)
    enddo
  enddo
enddo
```

and we see that the memory access pattern is wrong on a conventional machine, the  $y$ -direction varies fastest in memory, and should be the innermost loop. We therefore redeclared all the arrays, so that the  $y$ -direction would lie last in the array, and then it is correct to have it as the outer loop. It was also found that the use of compiler flags is important to get good performance. We achieved a performance of the code of about 55 Mflops on one wide node, and about 20 Mflops on a thin node, in 64 bit arithmetic. The difference in performance between a wide and a thin node is understandable since the wide node has a higher memory bandwidth and a larger data cache.

Recently IBM has delivered a parallel version of the ESSL, called Parallel ESSL. We have therefore begun to work on a parallel version of the SP2 code. We will use Fortran 90 with MPL, Message Passing Library, or MPI, Message Passing Interface, for the communication, and a parallel FFT from PESSL. The distributed axis will be the  $z$ -axis, and  $x$  and  $y$  axes are local on each processor. Almost all communication will take place in the FFT, the only communication we have to explicitly state are in the input routines (`rdisc.f` `rparam.f`), the output routines (`wdisc.f` `wpl.f` `wamp.f` and `wcfl.f`), and a few routines that collect statistics (`boxamp.f` `boxcfl.f` `hghamp.f`).

Since there is no concept of a global address space in Fortran 90 (as in CM Fortran or HPF), we must declare all arrays to be the size of the local arrays. Example, if we run our code on four nodes, and have a 64 cube problem, we must declare the extent of the  $z$ -axis to be  $64/4 = 16$ . This is done in the include file `par.f` where one declares the number of processors to run on.

It is difficult to predict what kind of performance levels we could expect from the code when running on several nodes. The code on local data would yield a perfect speed-up, and the only part of the algorithm suffering from communication would be the FFT. Measurements made by IBM on a eight wide node SP2 suggests a speedup of about 5 for an complex two-dimensional FFT. If this would hold also in our code the expected overall speedup would be about 5–6, thus giving a performance of about 250–300 Mflops on eight wide nodes.

### 6.5. Performance of *chapar* on CM200

Predicting performance of a program on the CM200 is a difficult task. It depends on the size of the machine, and on the size of the problem. It also depends on how the compiler decides to distribute the arrays on the machine, and on the communication pattern that results from this distribution. This distribution is not explicitly controllable by the programmer (at least not in Fortran), but one has to rely on the compiler directives.

In our measurements of the performance of the program, we have found that the number of instances along the  $z$ -axis should match the number of FPU's on the machine for best performance. See for instance table 5 where we have listed the performance of a  $256 \times 257 \times 256$  problem, with various length of the  $x$  and  $z$  axes. The times are for one iteration and the flop count are

problem size	Single precision			Double precision		
	[s]	$\mu\text{s}/\text{gp}$	Mflops	[s]	$\mu\text{s}/\text{gp}$	Mflops
$1024 \times 257 \times 64$	21.7	1.3	510	37.3	2.2	300
$512 \times 257 \times 128$	21.6	1.3	510	37.2	2.2	300
$256 \times 257 \times 256$	21.6	1.3	510	37.1	2.2	300
$128 \times 257 \times 512$	11.0	0.65	1000	17.6	1.0	630
$64 \times 257 \times 1024$	12.0	0.71	920	18.8	1.1	590

TABLE 5. Performance of the program for the largest problem fitting in the memory, and various length of the  $z$ -axis on a 16k CM200. gp in the table stands for grid points.

CM200 size	problem size	Single precision			Double precision		
		[s]	$\mu\text{s}/\text{gp}$	Mflops	[s]	$\mu\text{s}/\text{gp}$	Mflops
4k	$32 \times 33 \times 32$	0.19	5.6	80	0.30	9.1	50
4k	$64 \times 65 \times 64$	1.30	4.9	100	2.13	8.1	65
4k	$128 \times 129 \times 128$	5.47	2.6	230	8.81	4.2	140
8k	$32 \times 33 \times 32$	0.18	5.3	85	0.28	8.3	55
8k	$64 \times 65 \times 64$	0.66	2.5	210	1.09	4.1	130
8k	$128 \times 129 \times 128$	5.37	2.5	230	9.12	4.3	140
16k	$32 \times 33 \times 32$	0.18	5.3	85	0.28	8.3	55
16k	$64 \times 65 \times 64$	0.36	1.3	380	0.56	2.1	250
16k	$128 \times 129 \times 128$	2.85	1.3	440	4.74	2.2	260

TABLE 6. Performance of the program for various sizes of the problem and various sizes of the CM200. gp in the table stands for grid points.

about  $110n_1n_2n_3 + 9 \cdot 5/2n_1n_2n_3 \cdot \log_2(n_1n_3) + 9 \cdot 5/2n_1n_2n_3 \cdot \log_2(n_2 - 1)$ . The length of axis  $x$  is halved and the length of axis  $z$  is doubled in each row, but the total number of spectral modes remains constant (about 17 million). The tests are run on a 16k CM200, i.e. with 512 FPU's. Notice the doubling in performance for a problem with  $z$ -axis of length equal to the number of FPU's.

For performance data on smaller problem, see table 6, where we have tabulated three different problem sizes, and tested each on three different machine sizes. When the size of the problem is doubled in each dimension, the time required do not scale accordingly, but only about 2–7 times, meaning that the time per iteration and grid-point decreases, and thus the performance increases. We also see from the table that running a small problem on larger machines does not yield higher performance. For instance the 32 cube problem yield about 50 Mflops independent of the size of the machine. But when running sufficiently large problems the performance usually doubles when moving to a larger machine, see the 64 cube problem, which runs at 65 Mflops on a 4k

CM200 and doubles to 130 Mflops on a 8k CM200 and doubles again to 250 Mflops on a 16k CM200.

On a larger machine we can expect an increase in performance proportional to the size of the machine if the problem size is scaled similarly and the increase in size is put on the off-chip ( $z$ ) axis. However, the length of the two other axes cannot be increased on-chip. The added off-chip address bits may affect performance, causing a sub-linear speed-up. However, the length of one on-chip axis can be increased if the length of the other on-chip axis is decreased with the same factor, without performance loss.

## 7. Performing a simulation

A simulation consists of several steps, first one has to create an initial velocity field, then perform the actual simulation, and last one has to analyze the data. These steps are performed with the aid of the programs `sta`, `chapar` and various programs for analyzing the data. These are `rit` for whole velocity fields, `rps` for plane files, `pamp1` and `pamp2` for box averaged amplitude files, and `pstat` for  $y$ -dependent  $x$ - $z$  averaged amplitude files. These programs are further described below.

### 7.1. Initializing velocity fields

The necessary initial velocity field is generated by the program `sta`, which is the same program used for the serial version of the channel code, i.e. `cha`, see further [Lundbladh \*et al.\* \(1992\)](#). The same options that can be used for the serial version of the code can also be used in the parallel version, the only exception is the symmetry option, this should not be used in the parallel version: `nfzsym` should be 0.

The initial velocity field resides in the front end file system, therefore the start up of `chapar` is rather slow, due to the limited speed of data from the front end to the CM200.

### 7.2. Execution of `chapar`

#### 7.2.1. Configuring `chapar`

The main configuration is done in `par.f`, which is an include file that contains the size of the problem, i.e. the number of grid points in each direction. For an example of a `par.f` file see appendix [Appendix A.3.1](#).

The configuration that can be done at run time is set in the input file `cha.i`. The options can be divided in numerical options and options for data input and output. Among the numerical is options for variable or fixed time stepping, time stepping method, Adam–Bashforth or 3 or 4-stage third order Runge–Kutta, and dealiasing in  $x$  and/or  $z$ -direction. The options for data input includes the name of the input file, if the input field can be of different size than what `chapar` is configured for. The options for output of data

includes possibilities of writing various data, i.e. whole velocity fields for different times, plane files and amplitude files, both  $y$ -dependent  $x$ - $z$ -averaged and box-averaged. The plane files can also be written in compressed format to save disk space. For a complete description of the different options see section [Appendix A.1](#), and for an example of a `cha.i` file see section [Appendix A.3.3](#).

### 7.2.2. *Memory requirements and problem size considerations*

Due to the performance loss associated with automatic arrays, almost all of the programs need to be recompiled for each size of problem, excluding only some startup routines, which are size independent. The size of the problem (i.e. the number of grid points) is set before compilation in the file `par.f`, by the parameters `nx`, `ny` and `nz`. There are restrictions on the parameters: `nx` and `nz` must be powers of two, due to the FFT routines used, and `ny-1` must be even and factorable by 2,3 and/or 5. The code is written in single precision, whereas in most cases there is a need to run in double precision. The CM200 Fortran compiler, `cmf` has a compiler flag, `-double-precision` to be used for that purpose. One problem is that the `cmf` flag `-double-precision` only promotes real and complex to double length (8 bytes), but not integers. The `f77` flag `-r8` on the other hand promotes both real and integer to double length. Therefore we could not use the latter flag for compiling the `VECFIT` package or the `sta` program. Instead we run all sun-fortran program through a filter which converts single precision program to double precision.

The parallel program uses more memory than the serial one, although some effort to minimize the memory usage was made. There are more three dimensional arrays in the parallel code than in the serial code, due to the fact that the serial code traversed the data plane by plane, whereas in the parallel program the whole domain is transformed as a whole. There are 14 three dimensional variables, and the memory requirement is  $14 \cdot nx \cdot ny \cdot nz \cdot nprec$  bytes, where `nprec` is 4 bytes for single precision and 8 for double. In addition to this there are a few two-dimensional arrays, but since they only require  $nx \cdot nz \cdot nprec$  bytes each they do not contribute significantly to the total memory usage. This means that the program requires about 230 megabytes of memory for a 128 cube problem, and about 1800 megabytes for a 256 cube problem. This means that the 128 cube problem can run on a 4k CM200 using about half the memory, and the 256 cube problem can run on a 16k CM200 using about 90 % of the memory.

## 7.3. *Post processing*

The post processing programs are used to analyze the various output files. They can generate plots in Postscript, and some in other more special formats.

### 7.3.1. *Post processing the velocity field with rit*

The program `rit` is used to read an entire velocity field and generate various plots. These can be generated in either Tektronix 4014 format or Postscript.

There is also a possibility to generate plots of the velocity field in one plane in PGM format, a grey scale coding of the velocity. The program `rit` is able to read compressed format. The program is intended to be fairly self explanatory.

### 7.3.2. Post processing the plane files with `rps`

The program `rps` reads plane files, both compressed and uncompressed, and is used to generate various plots. The program handles  $x$ - $z$  planes and  $x$ - $y$  planes but not  $y$ - $z$  planes. The program is also used for generating the input files for the utility `xmovie` to enable the visualization of the flow. This data format of these files is binary, with one pixel corresponding to one byte. The program is intended to be fairly self explanatory.

### 7.3.3. Post processing the amplitude files with `pamp1`, `pamp2` and `pstat`

The programs `pamp1` and `pamp2` reads the box averaged amplitude files. `pamp1` is used to plot data from one file and `pamp2` is used to plot data from several files. `pstat` reads the  $y$ -dependent  $x$ - $z$ -plane averaged amplitude file. It can be used to generate plots of various  $y$ -dependent quantities, both time averaged and instantaneous. There is also a possibility to plot the same data as `pamp1` does, i.e. box averaged data. It can also be used for plotting running time averaged data together with instantaneous data. The programs are intended to be fairly self explanatory.

## 7.4. Utilities

There are a few utilities to the above programs. One is the already mentioned `xmovie`, which takes as input the files generated by `rps` and displays them on the display as a movie. Other utilities are the compression programs, `dpc` and `dfc`. `dpc` compresses and uncompresses the plane files and `dfc` compresses and uncompresses the velocity field files.

## References

- BURKES, A. W. & BURKES, A. R. 1981 The ENIAC: first general-purpose electronic computer. *Ann Hist Comput.* **3** (4), 310–399.
- CANUTO, C., HUSSAINI, M. Y., QUARTERNI, A. & ZANG, T. A. 1988 *Spectral Methods in Fluid Dynamics*. Springer series in computational physics. Berlin New York: Springer-Verlag.
- Connection machine technical manual 1991 CM-200. *Tech. Rep.*. Thinking Machine Corporation, Cambridge.
- DONGARRA, J. 1995 Performance of various computers using standard linear equation software. available from netlib.gov in electronic form.
- DUNCAN, R. 1988 A survey of parallel computer architectures. *Computer* **23** (2), 9–21.
- ENSLAW, P. H. 1977 Multiprocessor organization – A survey. *Computing surveys* **9** (1), 103–129.

- FLYNN, M. J. 1966 Very high speed computing systems. *Proc. IEEE* **54**, 1901–1909.
- HELIN, J. 1992 Performance analysis of the CM-2, a massively parallel SIMD computer. In *ACM International conference on Supercomputing*. Washington.
- HOCKNEY, R. W. 1985 Parallel computers: Architecture and performance. In *Parallel computing 85* (ed. M. Feilmeier), pp. 33–62. North-Holland.
- JOHANSSON, S. L., JACQUEMIN, M. & KRAWITZ, R. L. 1992 Communication efficient multi-processor FFT. *J. Comp. Phys.* **102**, 381–397.
- LEIGHTON, F. T. 1992 *Introduction to parallel algorithms and architectures: Arrays, trees, hypercubes*. San Mateo, California: Morgan Kaufmann Publishers.
- LINCOLN, N. R. 1977 It's really not as much fun building a supercomputer as it is simply inventing one. In *Proceedings of the symposium on high speed computer and algorithm organization* (ed. D. J. Kuck), pp. 3–11. University of Illinois.
- LUNDBLADH, A. 1993 Simulation of bypass transition to turbulence in wall bounded shear flows. Phd thesis, Royal Institute of Technology, Department of Mechanics, Stockholm.
- LUNDBLADH, A., HENNINGSON, D. S. & JOHANSSON, A. V. 1992 An efficient spectral integration method for the solution of the Navier–Stokes equation. *Tech. Rep. TN 1992-28*. FFA, Stockholm.
- ORTEGA, J. M. 1989 *Introduction to parallel and vector solutions of linear systems. Frontiers of computer science 1*. New York: Plenum press.
- PEASE, M. C. 1977 The indirect binary  $n$ -cube microprocessor array. *IEEE Trans. Comput.* **C-26**, 474–482.
- ROGALLO, R. S. 1977 An ILLIAC program for the numerical simulation of homogeneous incompressible turbulence. *Tech. Rep. TM-73203*. NASA.
- SABOT, G. 1991 Efficiency hints for slicewise CM fortran 1.0. *Tech. Rep.*. Thinking Machine Corporation, Cambridge.
- SHURKIN, J. 1984 *Engines of the mind, A history of the computer*, 1st edn. New York: W. W. Norton & Company.
- SLOTNICK, D. L., BORK, W. C. & MCREYNOLDS, R. C. 1962 The SOLOMON computer. *AFIPS Conf. Proc.* **22**, 97–107.
- STRIKWERDA, J. C. 1989 *Finite difference schemes and partial differential equations. Mathematics series*. New York: Wadsworth & Brooks/Cole.
- TENGWALL, C. G. 1994 SP2 architecture and performance. In *Parallel scientific computing Proceedings PARA '94*. Lyngby, Denmark.
- UNGER, S. H. 1958 A computer oriented towards spatial problems. *Proc. Inst. Radio Eng. (USA)* **46**, 1744–1750.

## Appendix A. Appendix, description of parameter files and subroutines

### Appendix A.1. *cha.i* file

*cha.i* is formatted and sequential. Comments can be put after data on lines not containing character input, but no tabbing are allowed. Contents line by line :

1. **namnin**; character\*80. Input velocity file name.
2. **namnut**; character\*80. Output velocity file name.
3. **tmax**; real. The final time to which to simulate, determines the scope of the simulation.
4. **imax**; integer. The maximum number of iterations to simulate. The simulation will stop after this number of iterations, if the final time is not reached.
5. **dt**; real. The time step length,  $\leq 0$  for variable time step, in which case it will be regulated to keep the CFL number close to CFL max for the time stepping method chosen.
6. **nst**; integer. The time stepping method. (1 Euler/AB2, 3 three Stage Runge-Kutta, 4 four stage Runge-Kutta)
7. **nfxd**; integer. Dealiasing flag for x-direction, 1 for dealiasing, 0 for no dealiasing
8. **nfzd**; integer. Dealiasing flag for z-direction, 1 for dealiasing, 0 for no dealiasing
9. **varsize**; logical. Flag to allow read of a file of different size than the code is compiled for. The spectral coefficients are padded with zeros or truncated to achieve a spectral accurate interpolation. The resolution can not be reduced in wall normal direction.
10. **cflux**; logical. Flag for constant mass-flux.
11. **rot**; real. The dimension less rotation rate around the spanwise axis, 0 for no rotation.
12. **spat**; logical. Flag for spatial simulation. With this version of **chapar** it is currently not possible to do a spatial simulation. This flag should be set to **.false..**
13. **icfl**; integer. The interval in number of iterations for calculating the CFL number.
14. **iamp**; integer. The interval in number of iterations for calculating, and saving, the various amplitudes. If equal to zero no amplitudes will be calculated.
15. **namamp**; character\*80. The name of the amplitude file if the previous item is nonzero, otherwise left out.
16. **longli**; logical. Flag to generate long (*y*-dependent) amplitude statistics .
17. **namlng**; character\*80. Name of the file to save the *y*-dependent statistics if the previous item is true, otherwise left out.

18. `ixt`; integer. The interval in number of iterations for extremum amplitudes. Currently unused. Should be 0 in the current version.
19. `namext`; character\*80. The name of the extremum file if the previous item is nonzero, otherwise left out. Currently unused.
20. `msave`; integer. The number of complete intermediate velocity fields to be saved. If nonzero item 21 and 22 are repeated for each file.
21. `tsave`; real. The time for which to save an intermediate field. If variable time stepping has been chosen the program automatically adjusts the time step to reach exactly the desired time. For fixed time stepping the save is done at the nearest time.
22. `nmsave`; character\*80. The name of the intermediate velocity file.
23. `mwave`; integer. The number of wavenumbers to save amplitudes for. If nonzero, item 25 is repeated for each wavenumber. Currently unused. Should be zero in the current version.
24. `namwav`; character\*80. The name of the wavenumber file.
25. `kx kz`; both integers. The wavenumber  $\alpha$  as multiples of the fundamental  $2\pi/x_L$ , the wavenumber  $\beta$  as multiple of the fundamental  $2\pi/z_L$ .
26. `np1`; integer. The number of planes to be continuously saved during the simulation ( $< 15$ ). If nonzero followed by item 27 and then for each plane item 28 through 31, and possibly 32.
27. `ip1`; integer. The saving interval for planes in number of iteration.
28. `tpl(:,1)`; integer. The type of plane to be saved. 1 for  $x$ - $y$  plane, 2 for  $x$ - $z$  plane. Saving of  $y$ - $z$  plane currently not implemented.
29. `p1(:,2)`; integer. The variable to be saved, i.e. 1 for  $u$ , 2 for  $v$ , 3 for  $w$ .
30. `cpl`; real. The coordinate for which to save the plane.
31. `nampl`; character\*80. The name of the file in which to save the planes. If the name has a suffix of `.pc`, `.pcl`, `.pch` or `.pcv` the output is compressed. The compression rate depends on the desired accuracy of the output plane, and this is chosen with the suffix, `.pc` for an accuracy of  $10^{-8}$ , `pcl` for an accuracy of  $10^{-4}$ , `.pch` for an accuracy of  $10^{-12}$ . The suffix `.pcv` lets the user determine the accuracy. In this case the next item is the desired accuracy.
32. `eps`; real. The desired accuracy of the compressed plane.

## Appendix A.2. *Description of the various variables and subroutines*

Here we give the various subroutines, their names and what they do.

### Appendix A.2.1. *chapar and main variables*

This is the main code segment which contains declarations of all the global arrays. The local code takes care of the time stepping. General naming convention is that an array whose name ends with an 'r' is the real part of the three dimensional array, and an array whose name ends with an 'i' is the imaginary part.

Variable names in *italics* are old variables that is unused in the current version (1.5).

*global variables*

*main storage*

<i>ur, ui</i>	before <b>nonlin</b> the velocity $\hat{u}$ , after the nonlinear product <i>hatH1</i>
<i>vvr, vvi</i>	before <b>nonlin</b> the velocity $\hat{v}$ , after the nonlinear product <i>hatH2</i>
<i>wr, wi</i>	before <b>nonlin</b> the velocity $\hat{w}$ , after the nonlinear product <i>hatH3</i>
<i>omxr, omxi</i>	vorticity $\hat{\chi}$
<i>omyr, omyi</i>	vorticity $\hat{\omega}$
<i>omzr, omzi</i>	vorticity $\hat{\vartheta}$
<i>fomyr, fomyi</i>	right hand side for normal vorticity equation $\hat{p}_\omega$
<i>homyr, homyi</i>	nonlinear term in the normal vorticity equation $\hat{h}_\omega$
<i>phr, phi</i>	particular solution for the $\phi$ equation
<i>fvr, fvi</i>	right hand side for $\phi$ equation $\hat{p}_v$
<i>q</i>	the top rows in the tridiagonal matrices
<i>w3</i>	work space array for FFTs and integration
<i>tmprx, tmpix</i>	work space array used in <b>prhs</b>
<i>tmpry, tmpiy</i>	work space array used in <b>prhs</b>
<i>tmprz, tmpiz</i>	work space array used in <b>prhs</b>

*other global variables*

<i>boxr, boxi</i>	two dimensional work arrays
<i>k2, k2i</i>	two dimensional arrays containing $k^2$ and $1/k^2$
<i>alfa</i>	the wavenumbers $\alpha$ in the $x$ -direction
<i>beta</i>	the wavenumbers $\beta$ in the $z$ -direction
<i>pww</i>	partial right hand sides for the $u, w$ equations, wavenumber zero
<i>var</i>	flag for variable time step length
<i>t</i>	the simulation time for the present time step
<i>tc</i>	the simulation time for the present stage
<i>tmax</i>	the maximum (final) time to which to simulate
<i>tleft</i>	the amount of time left until next save of a velocity field
<i>dtn</i>	the time step length for the current step
<i>dtnp1</i>	the time step length for the next step
<i>an, bn</i>	time stepping coefficient $a^n, b^n$ for the current step
<i>anp1, bnp1</i>	time stepping coefficient $a^{n+1}, b^{n+1}$ for the current step
<i>anrk, bnrk</i>	table for the time-stepping coefficients for RK3
<i>it</i>	the number of the current iteration
<i>maxit</i>	the maximum number of iterations
<i>nst</i>	the number of stages in the time discretization
<i>nsave</i>	the maximum number of saved velocity field
<i>varsiz</i>	flag for allowing read of a velocity field of another size than the program is compiled for
<i>namnin</i>	the name of the input velocity field file
<i>namnut</i>	the name of the final output velocity field file
<i>nmsave</i>	the names of the intermediate velocity fields files
<i>tsave</i>	the times for which to save intermediate velocity fields
<i>isave</i>	the number for the next intermediate velocity field to save

urx	temporary variable to read or write one record from an external file
udr	<i>temporary variable to read or write one record from the external ur file</i>
mpl	the maximum number of plane files
nampl	the names of the plan files
tpl(,1)	the type of plane 1 for <i>xy</i> , 2 for <i>xz</i>
tpl(,2)	the variable 1 for <i>u</i> , 2 <i>v</i> and 3 <i>w</i>
tpl(,3)	the coordinate index for the plane, i.e. <i>z</i> for <i>xy</i> -planes, <i>y</i> for <i>xz</i>
npl	the number of plane files
ipl	the number of time iterations between writing planes
cpl	the coordinate value for the plane, i.e. <i>z</i> for <i>xy</i> -planes, <i>y</i> for <i>xz</i>
longli	flag to output <i>y</i> -dependent statistics
icfl	the calculation interval for the CFL number
iamp	the calculation interval for the amplitudes
nwave	<i>the maximum number of waves for which to save amplitudes</i>
mwave	<i>the number of waves to save amplitudes for</i>
namamp	the name of the amplitude file
namlng	the name of the <i>y</i> -dependent amplitude file
namwav	<i>the name of the wave amplitude file</i>
namext	<i>the name of the extremum amplitude file</i>
cfl	the CFL number
cflmax	the value to which the CFL number is regulated
cflp	the partial CFL values in each box = $CFL/(\pi\Delta t)$
kx, kz	<i>the wave numbers for which to save amplitudes</i>
amp	the total amplitudes for each <i>y</i> -plane
campw	<i>the complex wave amplitude for selected wavenumbers for each y-plane</i>
iext	<i>the calculation interval for the extremum amplitudes</i>
vext	<i>the extremum amplitudes for each y-plane</i>
cext	<i>the coordinates for the extremum amplitudes</i>
pou	Poiseuille flow flag, T/F Poiseuille/Couette flow
cflux	flag, .true. to enforce constant massflux
re	the Reynolds number
px	the mean pressure gradient
mflux	the initial field massflux
rot	the rotation number $\Omega$
u0low, u0upp	the mean <i>u</i> -velocities on the lower and upper plate
w0low, w0upp	the mean <i>w</i> -velocities on the lower and upper plate
eta	the coordinates for the collocation points
deta	the distances between the collocation points
wint	the integration weights for each collocation point
xl	$x_L$ , the length of the box in the <i>x</i> -direction
zl	$z_L$ , the length of the box in the <i>z</i> -direction
xs	the distance the box has been shifted to the right since time zero at the current time-step
xsc	the distance the box has been shifted at the current stage
prex	preprocessing data for the <i>x</i> -direction FFT
prey	preprocessing data for the <i>y</i> -direction Chebyshev transform
prez	preprocessing data for the <i>z</i> -direction FFT
ifac	preprocessing data for the <i>y</i> -direction parallel Chebyshev transform
wsave1	preprocessing data for the <i>y</i> -direction parallel Chebyshev transform

<b>wsave2</b>	preprocessing data for the $y$ -direction parallel Chebyshev transform
<b>fftid1</b>	preprocessing data for CMSSL FFT
<b>fftid2</b>	preprocessing data for CMSSL FFT
<b>nfxd</b>	dealiasing flag in $x$ -direction
<b>nfzd</b>	dealiasing flag in $z$ -direction
<b>i</b>	loop index
<b>dabyte</b>	# bytes in a record, plane compressing
<b>darec</b>	record, used to save plane compressed data
<b>eps</b>	the chosen accuracy for plane compressing
<b>epsc</b>	all stored values in the plane compressing are multiples of epsc
<b>mbit</b>	plane compressing stuff
<b>mcbt</b>	plane compressing stuff
<b>darecl</b>	record length necessary to host dabyte bytes
<b>dapt</b>	pointer, pointing to the right position in darec

### Appendix A.2.2. *The other subroutines*

#### **boxamp** and **boxcfl**

**boxamp** accumulates the square of the amplitudes for 12 statistical quantities by  $y$ -plane. Both total amplitudes and those of selected wavenumbers are calculated. The accumulation is done in Fourier-physical space by use of the Parseval identity.

**boxcfl** finds the maximum partial CFL number in one box.

#### **dcheb** and **icheb**

**dcheb** and **icheb** perform Chebyshev differentiation and integration respectively on wave number zero. In case of integration the zeroth coefficient in the integrated variable must be supplied.

#### **dealias**

**dealias** performs dealiasing in the  $x$  and/or  $z$  direction by truncating the top 1/3 spectral modes.

#### **fft2db** and **fft2df**

The routines perform a 2-dimensional Fourier transform in the  $x$  and  $z$  directions of one variable by calling the appropriate routines in CMSSL. **fft2df** works from physical to Fourier space and **fft2db** in the opposite direction. The transforms are not normalized.

#### **getdt**

**getdt** calculates the timestep corresponding to the desired value of the CFL-number,  $cflmax$ .

#### **hghamp**

**hghamp** accumulates the amplitudes to the power of three and four for the velocities as well as the amplitude of the Reynolds stress  $uv$  by  $y$ -plane. This is done in physical space. A total of 23 statistical quantities are accumulated by  $y$ -plane.

**infft**

**infft** initiates the various FFT routines as well as the parallel Chebyshev routine.

**linear**

**linear** advances all velocities one iteration with the nonlinear terms calculated in **nonlin** as driving terms in the linear evolution equations.

**nonlin**

Calculates the quadratic advective terms, the cfl number and accumulates statistics.

**par.f**

**par.f** is an include file in which compile time options can be set. The size of the problem is set here.

**pchbb** and **pchbf**

The routines perform a Chebyshev transform in the  $y$  direction of one variable. **pchbf** works from physical to spectral space and **pchbb** from spectral to physical space. The transforms are not normalized.

**pcmpr**

**pcmpr** performs preprocessing for the plane compression subroutines.

**pcosti**

**pcosti** initializes the preprocessing arrays for the parallel Chebyshev routines.

**pdcheb** and **picheb**

**pdcheb** and **picheb** are parallel versions of **dcheb** and **icheb**. They perform Chebyshev differentiation and integration respectively on the parallel arrays. In case of integration the zeroth coefficient in the integrated variable must be supplied.

**ppar**

**ppar** prints all compile time parameters.

**prad2**, **prad3**, **prad4** and **prad5**

**prad2** performs a radix 2 Chebyshev transform of one variable. The others perform a radix 3, 4, or 5 Chebyshev transform. They are used by **pchbb** and **pchbf**.

**prepr**

**prepr** calculates certain preprocessing information to increase the speed of the subsequent computation.

**prhs**

**prhs** is called once in the initialization phase to calculate partial right hand sides and the vorticities  $\chi, \vartheta$  necessary to start the main time iteration loop. These quantities are later generated as a byproduct when advancing the evolution equations in linear.

**psetmat**, **trid3** and **trid5**

**psetmat** and **trid3**, **trid5** are the parallel versions of **setmat** and **trid**. They together solve Helmholtz equations arising from the implicit discretization of diffusion and pressure in the  $y$ -direction. **Psetmat** generates the system matrix which is pentadiagonal

with the two first rows filled but with odd and the even coefficients uncoupled so that the system splits into two tridiagonal systems (with their first row filled), which are then solved by `trid3` or `trid5`. In fact several systems each with multiple right hand sides are generated and solved simultaneously, `trid3` is called with three right hand sides, and `trid5` with five right hand sides.

#### **rdisc and wdisc**

**rdisc** and **wdisc** reads/writes the contents of the main storage from/to a sequential file. **Rdisc** can optionally read files of a different size than the program is compiled for. In that case the upper frequencies are truncated or padded with zeroes as appropriate.

#### **rparam**

**rparam** reads in the execution parameters from the file **cha.i**, and sets the regulation allowed `cfi` value.

#### **setmat and trid**

**setmat** and **trid** together solve Helmholtz equations arising from the implicit discretization of diffusion and pressure in the  $y$ -direction. **Setmat** generates the system matrix which is pentadiagonal with the two first rows filled but with odd and the even coefficients uncoupled so that the system splits into two tridiagonal systems (with their first row filled), which are then solved by `trid`. In fact several systems each with multiple right hand sides are generated and solved simultaneously. They are here used only for wave number zero.

#### **wamp, wcfi**

**wamp** accumulates the statistics generated by **boxamp** and writes them to the logfile and an `ampfile`. If `longli` is true it also writes the statistics generated by **boxamp** and **hghamp** by plane to a separate `ampfile`. **wcfi** similarly accumulates the `cfi` number generated by **boxcfi** and writes it to the logfile.

#### **wpl**

**wpl** writes selected planes to output files at iteration intervals set by the parameter `ipl`. The planes are transformed to physical space by Fourier transforms. At present only  $x$ - $y$  and  $x$ - $z$  planes can be generated.

### Appendix A.3. *Examples, user created files*

#### Appendix A.3.1. *Example par.f file*

Below is an example of the adjustable part of a **par.f** include file. It is set up for a  $32 \times 33 \times 32$  spectral mode simulation.

```
c par.f contains size of problem
.
.
.
c adjustable parameters
c number of spectral modes
      integer nx,ny,nz
      parameter (nx=32,ny=33,nz=32)
```

Appendix A.3.2. *Example sta.i file*

Below is an example of a simple **sta.i** file to generate a velocity field with random disturbances in a file named **c0.u**. Note that comments are allowed on lines with non-character data.

```
c0.u
750    re
60     xl
30     zl
.false. Couette flow
0.0    amplitude (.1399 for 6%)
0.0    theta
1.     scale in x direction
1.     scale in z direction
1      type of distribution in the normal direction
.true.  noise
.005   energy density
16     number of x-modes (<=nx/2)
12     number of y-modes (even,<ny*2/3)
31     number of z-modes (odd, <nz)
-1     seed -700000 - -1
```

Appendix A.3.3. *Example cha.i file*

Below is an example of a simple **cha.i** file to run initial data in file **c0.u** to time 10 and output the result to file **c10.u**. An amplitude list is written to **c10amp.d** and a  $y$ -dependent amplitude file is written to **c10amp.dl**. Two plane files are written, both compressed, the first with default accuracy  $10^{-8}$  and the other with user chosen accuracy  $10^{-7}$ . Dealiasing is turned on in both  $x$  and  $z$  directions, and we allow read of field of other size than the compiled size of **chapar**. Cfl calculation interval is set to 4 iterations, i.e. every physical time step since the selected time stepping scheme (four stage Runge-Kutta) requires four iteration per time step. Amplitude calculation interval is set to 20 iterations, and also the plane saving interval is set to 20. One intermediate velocity field is saved at time 5, under the name **c5.u**.

```
c0.u
c10.u
10.    simulation end time
300    max number of iterations
0.0    time step, 0 for automatic variation
4      number of time-discretization stages (1/3/4)
1      dealiasing flag in x direction (1=dealiasing)
1      dealiasing flag in z direction (1=dealiasing)
.true.  allow read of field of other than compiled size
.false. constant massflux
```

```

0.0          rotation rate
.false.     spatial simulation
.false.     linearized equations
4           cfl calc interval
20          amplitude calc interval
c10amp.d    y-dependent amp-file
.true.
c10amp.dl
0           extremum calc interval, not available
1           number of saved fields
5           the time to save the complete field
c5.u
0           number of saved wavenumbers, not available
2           number of saved planes
20          plane saving interval
2           type of plane 2=xz-plane
1           variable to be saved 1=u
0.9        the coordinate for which to save the plane
cm9u10.pc
2           type of plane 2=xz-plane
2           variable to be saved 2=v
0.9        the coordinate for which to save the plane
cm9v10.pcv
1e-7       the desired accuracy for the above compr. plane

```



# Paper 5

P5



# Development of a spectrally accurate DNS code for cylindrical geometries

By Jukka Komminaho and Arne V. Johansson

Dept of Mechanics, KTH, SE-100 44 Stockholm, Sweden

Unpublished

A spectrally accurate simulation code for cylindrical geometries is developed, and the algorithm is described in detail. The algorithm is based on Fourier transforms in the azimuthal and axial directions, and Chebyshev polynomials are used in the radial direction. Verification of the algorithm was done by comparing analytical and numerical growth rates of eigenmodes, and showed excellent agreement.

---

## 1. Introduction

The use of direct numerical simulation (DNS) is today a well established technique to study different turbulent flows and increase our understanding of fundamental phenomena in turbulent flows.

Turbulence in pipe flow was studied as early as 1883 by Reynolds. Later experiments of fully developed turbulent pipe flow include, among many others, those by Laufer (1954); Towens *et al.* (1972); Lawn (1971); Perry & Abell (1975). More recent experiments include those by Reich & Beer (1989); Westerweel *et al.* (1992); Fontaine & Deutch (1995); Schwartz-van Manen & Nieuwstadt (1996); den Toonder & Nieuwstadt (1997); Zagarola & Smits (1998).

Reich & Beer (1989) investigated rotating pipe flow at low to moderate Reynolds number and different rotation rates. They report the effect of the rotation on the mean flow, flow resistance and heat transfer. They found that rotation suppresses the turbulent fluctuations, thus considerably reducing the friction coefficient and the heat transfer coefficient.

Fontaine & Deutch (1995) measured velocity statistics in the wall region of a turbulent pipe flow with the use of LDV. The Reynolds number based on bulk velocity,  $Re_b$ , was 5050. They report both second and higher order statistics very close to the wall, down to  $y^+ \approx 2$ , and compare the experimental findings with the ones from the boundary layer experiment of Karlsson & Johansson (1988) and the direct simulation of a channel flow of Antonia *et al.* (1992) with good agreement of the mean flow and second order statistics.

den Toonder & Nieuwstadt (1997) also made measurements with LDV in turbulent pipe flow for several different Reynold numbers. The  $Re_b$  varied from

5000 to 25000, and they investigated the Reynolds number effect on second order statistics. They concluded that the axial rms value does not show any Reynolds number effect for  $y^+ < 30$  whereas for the radial rms a clear  $Re$ -dependance was observed. The Reynolds-number-independence of axial rms velocity is consistent with the findings in the survey of several different flow cases, boundary layer flow channel flow and pipe flow, with a large range of Reynolds number by Mochizuki & Nieuwstadt (1996). The  $Re_\tau$  varied from 100 to 4300, and the peak axial rms value was almost constant. They compared these rms profiles with the DNS of Eggels *et al.* (1994) and found good agreement for the axial rms value, but a discrepancy for the radial rms, with higher values than in the simulation. The reason for this was unclear.

Zagarola & Smits (1998) made experiments in a pressurised pipe, and measured the mean velocity for Reynolds number from  $31 \times 10^3$  to  $35 \times 10^6$ .

There are today a few reported studies of turbulent pipe flow using numerical simulations. The early work concerned stability and transition. Fully developed laminar pipe flow shares with Couette flow the property that it is linearly stable over all Reynolds numbers, Salwen & Grosch (1972); Garg & Rouleau (1972).

Patera & Orszag (1981) and Orszag & Patera (1983) investigated the non-linear stability of pipe flow guided by their theory of secondary instability. They found that all axi-symmetric disturbances decay, but non-axisymmetric disturbances can be strongly unstable (non-linearly). Schmid & Henningson (1993) examined optimal energy density growth in pipe flow. They showed that, even in the linear region, certain optimal disturbances exhibit large transient growth before they eventually decay.

Leonard & Wray (1982) developed a spectral algorithm using Fourier transforms in azimuthal and axial directions and Jacobi polynomials in the radial direction. The disadvantage with Jacobi polynomials is that they lack fast transform methods, i.e. the number of operations for the algorithm scales as  $\mathcal{O}(N^2)$  whereas for a spectral method based on fast transforms it scales as  $\mathcal{O}(N \log N)$ , where  $N$  is the number of polynomials in the expansion.

Unger *et al.* (1993) used a second-order accurate finite difference method for DNS of fully developed turbulent pipe flow, with Reynolds number,  $Re_b$  up to 5300. They obtained good agreement with experiments by Westerweel *et al.* (1992) for the second order statistics and fair agreement for higher order statistics. See also Eggels *et al.* (1994) where the results are presented in more detail.

Zhang *et al.* (1994) used a spectral method in their DNS of pipe flow with Reynold number up to 4000. Their code uses Fourier transforms in axial and azimuthal directions, and spectral elements with Jacobi polynomials in the radial direction.

Loulou (1996) developed and implemented a spectral method for pipe flow simulations, using fourier transformation in azimuthal and axial direction, and B-splines in the radial direction. In their simulation of rotating pipe flow,

Orlandi & Fatica (1997) used a finite difference scheme, second-order accurate in space. The algorithm is described in more detail in Verzicco & Orlandi (1996). They present both mean flow and both second and higher order statistics for both rotating and non-rotating pipe flow. As in the experiments by Reich & Beer (1989) they find that rotation gives a reduced drag. For the non-rotating case they obtain good agreement with Eggels *et al.* (1994). The data base produced was further used in Orlandi (1997) to study the change of turbulence production and dissipation with rotation.

This seemingly simple flow configuration hides two difficulties when solving Navier–Stokes equations in cylindrical coordinates. The first is purely mathematical and is related to the treatment of the coordinate singularity at the axis  $r = 0$ . The second comes from the numerical treatment of the time derivatives, and is related to the fact that when using cylindrical coordinates the azimuthal grid resolution is proportional to the radial distance from the centreline. Of these two difficulties, the former is most important. The only consequence of not taking the latter into consideration is a decrease in the size of the time-step. One possibility to remedy the time step problem is to filter the solution near the origin (Umsheid & Sankar-Rao 1971) and thereby reducing the resolution. Another approach is made by Akselvoll & Moin (1995). They decompose the computational domain into two regions. Within each region the derivatives in one direction, radial for the outer, and azimuthal for the inner region, is treated implicitly, thereby avoiding the time-step problem.

Loulou (1996) also reduced the number of grid points near the origin to alleviate the time-step problem. Unger *et al.* (1993) used a method similar to Akselvoll & Moin, they treated the diffusive and advective terms implicitly in the azimuthal direction, thereby allowing much larger time steps compared to the original explicit treatment.

When excluding the origin from the computational domain, i.e. in annular pipe or Taylor–Couette flow the time-step problem is less of a concern. For annular pipe flow there are relatively few experiments. One with a moving inner wall is the experiment by Shands *et al.* (1980)

Schumann (1975) developed a finite difference method for simulations of turbulent flows in plane channels and annular pipe flow, and obtained results that agreed rather well with experimental values.

Shapiro *et al.* (1999) recently investigated linear and nonlinear stability of annular pipe flow. The linear stability analysis was performed with Chebyshev collocation method. The nonlinear analysis was made with the same code used by Eggels *et al.* (1994), adapted to the annular domain. They obtained good agreement between the linear theory and their DNS simulation of the growth of the eigenfunctions with small initial amplitude. However, some of the eigenvalues in their eigenvalue map seems to be wrong, see further section 5.2 where we present eigenvalue maps for the same case. They further studied non-linear stability by introducing finite-amplitude eigenfunctions which were unstable, and followed their evolution in time, down to a point where their

resolution was insufficient. They concluded that the mechanism of transition consists of two counter-rotating modes with streamwise rolls, originating in the simulation from non-linear interaction between azimuthal modes.

The Taylor–Couette flow case has attracted many researchers. There are many articles on that subject, see e.g. [Chossat & Iooss \(1994\)](#) and references therein for a wealth of information. As examples of experimental studies see [Andereck \*et al.\* \(1986\)](#) for pure Taylor–Couette flow, and [Takeuchi & Jankowski \(1981\)](#) for Taylor–Couette flow with a pressure gradient in the axial direction. The latter case was also studied by [Leuptow \*et al.\* \(1992\)](#) and [Wereley & Leuptow \(1999\)](#).

[Moser \*et al.\* \(1983\)](#) developed and implemented an algorithm for solving Navier–Stokes equations in a cylindrical geometry, i.e. for Taylor–Couette flow as well as curved channel flow. It is a spectral algorithm, based on fourier transforms in axial and azimuthal directions and Chebyshev polynomials in the radial direction. The algorithm was evaluated for plane channel and Taylor–Couette flow with both axisymmetric and wavy vorticities, with good agreement with theoretical and experimental results.

Other flow configurations in cylindrical geometries that are possible to study with the present code are Taylor–Couette flow, annular pipe flow and curved channel flow.

This report describes the development of a spectral method for use in cylindrical coordinates. There are some differences in the algorithm for the cases pipe flow and annular pipe flow, mainly concerning the Chebyshev expansion in the radial direction.

## 2. Governing equations and the numerical method

The incompressible Navier–Stokes equations can be formulated in several different ways. The standard is based on the primitive-variables (velocity and pressure), but formulations can also involve streamfunctions and/or vorticity, see [Aziz & Hellums \(1967\)](#) for a streamfunction-vorticity formulation, and [Dennis \*et al.\* \(1979\)](#) for a velocity-vorticity formulation.

### 2.1. Derivations of the velocity-vorticity formulation

We will here solve the equations with a velocity-vorticity formulation, thereby avoiding the difficulty of solving for the pressure. Unlike the velocity, there is no evolution equation for the pressure and it is determined from the continuity equation. This often causes problems, see [Canuto \*et al.\* \(1988\)](#); [Gresho & Sani \(1987\)](#) where the authors discuss this.

The incompressible Navier–Stokes equations can, in conservative form, in a rotating reference system in general coordinates, be written as

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \underbrace{\mathbf{u} \times (\nabla \times \mathbf{u} + 2\boldsymbol{\Omega})}_{=\mathbf{H}} - \nabla \left( \frac{1}{2} \mathbf{u} \cdot \mathbf{u} \right) + \frac{1}{R} \nabla^2 \mathbf{u}, \quad (1a)$$

and the continuity equation for incompressible flow is given by

$$\nabla \cdot \mathbf{u} = 0, \quad (1b)$$

where  $\mathbf{u}$  is the velocity vector,  $\boldsymbol{\Omega}$  is the system angular rotation rate vector,  $R$  is the Reynolds number based on the pipe radius ( $\delta$ ) in case of pipe flow, or half the gap width in case of annular pipe flow, centreline (or maximum) velocity ( $U_{CL}$ ) and the kinematic viscosity ( $\nu$ ),  $R = U_{CL}\delta/\nu$ .

We rewrite in velocity-vorticity formulation to eliminate pressure. For this purpose we first take the divergence of the momentum equation and use the continuity equation to obtain a Poisson equation for the pressure,

$$\nabla^2 p = \nabla \cdot \mathbf{H} - \nabla^2 \left( \frac{1}{2} \mathbf{u} \cdot \mathbf{u} \right). \quad (2)$$

We apply the Laplace operator on the momentum equation, use the above equation for the pressure, to obtain a fourth order equation for the velocities with the pressure eliminated,

$$\frac{\partial \nabla^2 \mathbf{u}}{\partial t} = -\nabla(\nabla \cdot \mathbf{H}) + \nabla^2 \mathbf{H} + \frac{1}{R} \nabla^4 \mathbf{u}. \quad (3)$$

The next step is to rewrite (3) into a system of second order equations by introducing an auxiliary vector  $\phi$ ,

$$\nabla^2 \mathbf{u} = \phi \quad (4a)$$

$$\frac{\partial \phi}{\partial t} = \underbrace{-\nabla(\nabla \cdot \mathbf{H}) + \nabla^2 \mathbf{H}}_{=-\nabla \times (\nabla \times \mathbf{H})} + \frac{1}{R} \nabla^2 \phi. \quad (4b)$$

By taking the curl of the momentum equation we obtain a second order equation for the vorticities

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \nabla \times \mathbf{H} + \frac{1}{R} (\nabla \times \phi). \quad (5)$$

Note that  $\nabla \times \phi \equiv \nabla^2 \boldsymbol{\omega}$ . We now have our velocity-vorticity formulation in a general coordinate system.

## 2.2. Express in cylindrical coordinates

Introduce a cylindrical coordinate system  $(r, \varphi, z)$  with velocities

$$[u(r, \varphi, z, t), v(r, \varphi, z, t), w(r, \varphi, z, t)],$$

and vorticities

$$[\omega_1(r, \varphi, z, t), \omega_2(r, \varphi, z, t), \omega_3(r, \varphi, z, t)].$$

In the following we will not write the dependence of  $r, \varphi, z, t$ . Let the system angular rotation rate be around the  $z$ -axis,  $\boldsymbol{\Omega} = (0, 0, \Omega_3)$ . The continuity equation for the velocities can here be written as

$$\frac{1}{r} \frac{\partial ru}{\partial r} + \frac{1}{r} \frac{\partial v}{\partial \varphi} + \frac{\partial w}{\partial z} = 0. \quad (6)$$

The radial and azimuthal components of equations (4a), (4b) and (5) expressed in cylindrical coordinates read

$$\frac{\partial \phi_1}{\partial t} = h_{u_1} + \frac{1}{R} \left[ \left( \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \varphi^2} + \frac{\partial^2}{\partial z^2} \right) \phi_1 - \frac{\phi_1}{r^2} - \frac{2}{r^2} \frac{\partial \phi_2}{\partial \varphi} \right] \quad (7a)$$

$$\frac{\partial \phi_2}{\partial t} = h_{u_2} + \frac{1}{R} \left[ \left( \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \varphi^2} + \frac{\partial^2}{\partial z^2} \right) \phi_2 - \frac{\phi_2}{r^2} + \frac{2}{r^2} \frac{\partial \phi_1}{\partial \varphi} \right] \quad (7b)$$

$$\phi_1 = \left( \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \varphi^2} + \frac{\partial^2}{\partial z^2} \right) u - \frac{u}{r^2} - \frac{2}{r^2} \frac{\partial v}{\partial \varphi} \quad (7c)$$

$$\phi_2 = \left( \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \varphi^2} + \frac{\partial^2}{\partial z^2} \right) v - \frac{v}{r^2} + \frac{2}{r^2} \frac{\partial u}{\partial \varphi} \quad (7d)$$

$$\frac{\partial \omega_3}{\partial t} = h_{\omega_3} + \frac{1}{R} \left[ \frac{1}{r} \frac{\partial r \phi_2}{\partial r} - \frac{1}{r} \frac{\partial \phi_1}{\partial \varphi} \right], \quad (7e)$$

where

$$h_{u_1} = \frac{1}{r^2} \frac{\partial^2 H_1}{\partial \varphi^2} + \frac{\partial^2 H_1}{\partial z^2} - \frac{1}{r^2} \frac{\partial H_2}{\partial \varphi} - \frac{1}{r} \frac{\partial^2 H_2}{\partial r \partial \varphi} - \frac{\partial^2 H_3}{\partial r \partial z} \quad (7f)$$

$$h_{u_2} = \frac{1}{r^2} \frac{\partial H_1}{\partial \varphi} - \frac{1}{r} \frac{\partial^2 H_1}{\partial r \partial \varphi} - \frac{H_2}{r^2} + \frac{1}{r} \frac{\partial H_2}{\partial r} + \frac{\partial^2 H_2}{\partial r^2} + \frac{\partial^2 H_2}{\partial z^2} - \frac{1}{r} \frac{\partial^2 H_3}{\partial \varphi \partial z} \quad (7g)$$

$$h_{\omega_3} = \frac{1}{r} \frac{\partial r H_2}{\partial r} - \frac{1}{r} \frac{\partial H_1}{\partial \varphi}, \quad (7h)$$

and the nonlinear terms are

$$H_1 = v(\omega_3 + 2\Omega_3) - w\omega_2 \quad (7i)$$

$$H_2 = -u(\omega_3 + 2\Omega_3) + w\omega_1 \quad (7j)$$

$$H_3 = u\omega_2 - v\omega_1. \quad (7k)$$

The equations are coupled, the equation for  $\phi_1$ , (7a) also contains  $\phi_2$ , but as is shown in the next section a simple change of variables decouples the equations.

### 2.3. Fourier transformation

We apply the Fourier transform in the  $\varphi$  and  $z$  direction, with wave numbers  $\alpha$  and  $\beta$ ,  $u(r, \varphi, z, t) \rightarrow \hat{u}(r, \alpha, \beta, t)$ . Introduce the new variables

$$\hat{\phi}_A = \hat{\phi}_1 + i\hat{\phi}_2 \quad (8a)$$

$$\hat{\phi}_B = \hat{\phi}_1 - i\hat{\phi}_2 \quad (8b)$$

$$\hat{u}_A = \hat{u} + i\hat{v} \quad (8c)$$

$$\hat{u}_B = \hat{u} - i\hat{v}. \quad (8d)$$

This leads to a set of equations in  $\alpha$  and  $\beta$ , where the only coupling comes from the nonlinear terms. This technique has also been used by [Garg & Rouleau \(1972\)](#); [Orszag & Patera \(1983\)](#) and [Zhang \*et al.\* \(1994\)](#) to decouple the velocity equations.

The equations can be divided into one general and three special cases, depending on the wave number. This decomposition into four cases facilitates the treatment of the equations when dealing with the boundary conditions, see section 2.7. The categories are

- the general case,  $\alpha, \beta > 0$ ,
- $\alpha = \beta = 0$ , the mean flow, which is solved directly from the momentum equation (1a),
- $\alpha = 0$ , axi-symmetric case, for which (7a) is uncoupled, and we solve directly for  $\hat{\phi}_1$  and  $\hat{u}$ , and for  $\hat{v}$  through the momentum equation (1a),
- $\beta = 0$ , where we need to solve for  $\hat{w}$  from equation (1a), and solve for  $\hat{\omega}_3$  from (5) to obtain  $\hat{u}$  and  $\hat{v}$ .

One difficulty with the resulting equations is that we have no *a priori* boundary condition for  $\hat{\phi}_1$  and  $\hat{\phi}_2$ . The boundary conditions apply to the velocities  $\hat{u}$ ,  $\hat{v}$  and  $\hat{w}$ . This difficulty can be handled by solving for both a particular solution with a boundary condition that equals the value of the solution at the previous time level, and a homogeneous solution with non-zero boundary conditions. We then reconstruct the solution as a linear combination of the homogeneous and particular solutions, see also section 2.7.

### 2.3.1. Equations for the general case, $\alpha, \beta > 0$

The equations (7a) to (7e) can now be written in Fourier space as

$$\frac{\partial \hat{\phi}_A}{\partial t} = \hat{h}_{u_A} + \frac{1}{R} \left( \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{\alpha^2 + 2\alpha + 1}{r^2} - \beta^2 \right) \hat{\phi}_A \quad (9a)$$

$$\frac{\partial \hat{\phi}_B}{\partial t} = \hat{h}_{u_B} + \frac{1}{R} \left( \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{\alpha^2 - 2\alpha + 1}{r^2} - \beta^2 \right) \hat{\phi}_B \quad (9b)$$

$$\hat{\phi}_A = \left( \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{\alpha^2 + 2\alpha + 1}{r^2} - \beta^2 \right) \hat{u}_A \quad (9c)$$

$$\hat{\phi}_B = \left( \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{\alpha^2 - 2\alpha + 1}{r^2} - \beta^2 \right) \hat{u}_B \quad (9d)$$

$$\frac{\partial \hat{\omega}_3}{\partial t} = \hat{h}_{\omega_3} + \frac{1}{R} \left[ \frac{1}{r} \frac{\partial r \hat{\phi}_2}{\partial r} - \frac{i\alpha}{r} \hat{\phi}_1 \right], \quad (9e)$$

where the nonlinear terms are

$$\hat{h}_{u_1} = -\frac{\alpha^2}{r^2} \hat{H}_1 - \beta^2 \hat{H}_1 - \frac{i\alpha}{r^2} \hat{H}_2 - \frac{i\alpha}{r} \frac{\partial \hat{H}_2}{\partial r} - i\beta \frac{\partial \hat{H}_3}{\partial r} \quad (9f)$$

$$\hat{h}_{u_2} = \frac{i\alpha}{r^2} \hat{H}_1 - \frac{i\alpha}{r} \frac{\partial \hat{H}_1}{\partial r} - \frac{\hat{H}_2}{r^2} + \frac{1}{r} \frac{\partial \hat{H}_2}{\partial r} + \frac{\partial^2 \hat{H}_2}{\partial r^2} - \beta^2 \hat{H}_2 + \frac{\alpha\beta}{r} \hat{H}_3 \quad (9g)$$

$$\hat{h}_{\omega_3} = \frac{1}{r} \hat{H}_2 + \frac{\partial \hat{H}_2}{\partial r} - \frac{i\alpha}{r} \hat{H}_1 \quad (9h)$$

and

$$\hat{h}_{u_A} = \hat{h}_{u_1} + i\hat{h}_{u_2} \quad (9i)$$

$$\hat{h}_{u_B} = \hat{h}_{u_1} - i\hat{h}_{u_2}. \quad (9j)$$

The original variables  $\hat{u}$ ,  $\hat{v}$  can easily be extracted from the new  $\hat{u}_A$  and  $\hat{u}_B$  by

$$\hat{u} = \frac{1}{2}(\hat{u}_A + \hat{u}_B) \quad (10)$$

$$\hat{v} = \frac{i}{2}(\hat{u}_B - \hat{u}_A), \quad (11)$$

with similar equations for the other variables. The remaining variables that have to be calculated are the vorticities and the axial velocity. The vorticities are calculated through their definition,

$$\hat{\omega}_1 = \frac{i\alpha}{r}\hat{w} - i\beta\hat{v} \quad (12a)$$

$$\hat{\omega}_2 = i\beta\hat{u} - \frac{\partial\hat{w}}{\partial r} \quad (12b)$$

$$\hat{\omega}_3 = \frac{1}{r}\hat{v} + \frac{\partial\hat{v}}{\partial r} - \frac{i\alpha}{r}\hat{u}. \quad (12c)$$

The remaining velocity component can be obtained from the continuity equation for the velocities, equation (6),

$$\hat{w} = \frac{1}{\beta} \left( \frac{i}{r}\hat{u} + i\frac{\partial\hat{u}}{\partial r} - \frac{\alpha}{r}\hat{v} \right). \quad (13)$$

The equations (9e) and (12c) are two independent ways of calculating  $\hat{\omega}_3$ , and in section 2.7 we show that we need to use both equations in order to obtain expressions for the constants used in the evaluation of the boundary conditions.

### 2.3.2. Equations for $\alpha = 0$

For the case  $\alpha = 0$ , i.e. axi-symmetric case, the equations (7a)–(7b) are uncoupled and in this case we solve for  $\hat{\phi}_1$  and  $\hat{u}$ . Thus equations (9a) and (9c) are replaced by

$$\frac{\partial\hat{\phi}_1}{\partial t} = \hat{h}_u + \frac{1}{R} \left( \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{1}{r^2} - \beta^2 \right) \hat{\phi}_1 \quad (14a)$$

$$\hat{\phi}_1 = \left( \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{1}{r^2} - \beta^2 \right) \hat{u}, \quad (14b)$$

and we go back to the original momentum equation for  $v$ ,

$$\frac{\partial\hat{v}}{\partial t} = \hat{H}_2 + \frac{1}{R} \left( \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{1}{r^2} - \beta^2 \right) \hat{v}. \quad (14c)$$

The gradient terms (of pressure and dynamic pressure) on the right hand side of (1a) here vanish since  $\alpha = 0$ .

The equations for the axial velocity, (13) and  $\hat{\omega}_1$ , (12a) and  $\hat{\omega}_3$ , (12c) remain unchanged. For  $\hat{\omega}_2$  we can use the equations (13) and (14b) to obtain

$$\hat{\omega}_2 = -\frac{i}{\beta}\hat{\phi}_1. \quad (15)$$

The reason for this special treatment of  $\alpha = 0$  is of a numerical nature. When solving directly for  $\hat{\phi}_1$  and  $\hat{u}$  the expression for the constant necessary to obtain the correct boundary condition at the wall for  $\hat{u}$  (see section 2.7) is very simple, and actually the equations to solve become very similar to those of the channel flow code. Observe that it is only in the case  $\alpha = 0$  that the equations are similar, because of the decoupling of  $\hat{u}$  and  $\hat{v}$ .

### 2.3.3. Equations for $\beta = 0$

In the case  $\beta = 0$  we clearly need another equation for the axial velocity  $\hat{w}$ . This we obtain from the original momentum equation,

$$\frac{\partial \hat{w}}{\partial t} = \hat{H}_3 + \frac{1}{R} \left[ \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{\alpha^2}{r^2} \right] \hat{w} - P_z \delta_{\alpha 0}, \quad (16)$$

where  $P_z$  is the pressure gradient driving the flow,  $\delta_{\alpha 0} = 1$  for  $\alpha = 0$ , and zero otherwise. The gradient of the fluctuating pressure and dynamic pressure vanish here since  $\beta = 0$ .

For this case we may make use of the continuity equation in a more direct way and replace the equations for  $\hat{\phi}_A$  and  $\hat{\phi}_B$  by an evolution equation for  $\hat{\omega}_3$ ,

$$\frac{\partial \hat{\omega}_3}{\partial t} = \hat{h}_{\omega_3} + \frac{1}{R} \left[ \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{\alpha^2}{r^2} \right] \hat{\omega}_3. \quad (17)$$

It is now possible to obtain relations between  $\hat{u}$ ,  $\hat{v}$  and  $\hat{\omega}_3$ . Start by defining a stream function  $\Xi$  such that  $\mathbf{u} = \nabla \times \Xi$ . Now

$$\boldsymbol{\omega} = \nabla \times \mathbf{u} = \nabla \times (\nabla \times \Xi) \equiv -\nabla^2 \Xi,$$

and we are able to compute  $\hat{\Xi}_3$  from (keeping in mind that  $\beta = 0$ )

$$\left[ \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{\alpha^2}{r^2} \right] \hat{\Xi}_3 = -\hat{\omega}_3 \quad (18)$$

and  $\hat{u}$  and  $\hat{v}$  from

$$\hat{u} = \frac{i\alpha}{r} \hat{\Xi}_3, \quad \hat{v} = -\frac{\partial}{\partial r} \hat{\Xi}_3. \quad (19)$$

The vorticities  $\hat{\omega}_1$  and  $\hat{\omega}_2$  are computed from (12a) and (12b).

### 2.3.4. Equations for $\alpha = \beta = 0$

For the case  $\alpha = \beta = 0$  we compute  $\hat{v}$  and  $\hat{w}$  from equations (14c) and (16). The radial velocity  $\hat{u}$  is zero, and the vorticities are computed from their defining equations (12b) and (12c) (with  $\hat{\omega}_1 = 0$ ).

#### 2.4. Time stepping

The time stepping chosen is a semi-implicit one, with explicit treatment of the nonlinear term in equation (9a), (9b) and implicit treatment of the viscous term.

The model equation is

$$\frac{\partial f}{\partial t} = G(f) + \frac{1}{R} (D^2 f - \lambda^2 f) \quad (20)$$

where  $G(f)$  is a nonlinear term which we will discretize with explicit time stepping, and  $D$  denotes a derivative.

##### 2.4.1. The implicit part

The time derivative in the following model equation

$$\frac{\partial f}{\partial t} = \frac{1}{R} (D^2 - \lambda^2 f) \quad (21)$$

can be discretized with the following three time level implicit scheme, (Fletcher 1991)

$$\frac{(1 + \gamma_n)(f^{n+1} - f^n)}{\Delta t^{n+1}} - \frac{\gamma_n(f^n - f^{n-1})}{\Delta t^n} = \frac{1}{R} (D^2 - \lambda^2) ((1 - \beta_i)f^n + \beta_i f^{n+1}). \quad (22)$$

With this scheme we can get Backward Euler, Crank–Nicolson and the three-level fully implicit scheme (3LFI). Table 1 summarises the different properties of these schemes.

	$\beta_i$	$\gamma_n$	order in time
B Euler	1	0	1
CN	1/2	0	2, A-stable
3LFI	1	1/2	2, A-stable

TABLE 1. Time-stepping coefficients for the implicit part.

##### 2.4.2. The explicit part

The explicit part can be discretized with the following three time level explicit scheme, (Fletcher 1991)

$$\frac{(1 + \gamma_n)(f^{n+1} - f^n)}{\Delta t^{n+1}} - \frac{\gamma_n(f^n - f^{n-1})}{\Delta t^n} = (1 - \beta_e)G(f^n) + \beta_e G(f^{n-1}). \quad (23)$$

With this scheme we can get Forward Euler, second order accurate Adam–Bashforth and three-level fully explicit scheme, see also table 2.

	$\beta_e \frac{\Delta t^{n+1}}{\Delta t^n}$	$\gamma_n$	order in time
F Euler	0	0	1
AB2	-1/2	0	2
3LFE	-1	1/2	2

TABLE 2. Time-stepping coefficients for the explicit part.

### 2.4.3. Combining implicit and explicit parts

When combining both explicit and implicit parts we must use the same parameter  $\gamma_n$  for both parts, but the parameter  $\beta_e$ ,  $\beta_i$  does not have to be the same. For example, take  $\gamma_n = 0$ ,  $\beta_i = 1/2$  and  $\beta_e = -1/2$  yields CN for the implicit part and AB2 for the explicit part. See also table 3.

	$\beta_i$	$\beta_e \frac{\Delta t^{n+1}}{\Delta t^n}$	$\gamma_n$	order in time
BE/FE	1	0	0	(1,1)
BE/AB2	1	-1/2	0	(1,2)
CN/AB2	1/2	-1/2	0	(2,2)
BDF2	1	-1	1/2	(2,2)

TABLE 3. Time-stepping coefficients to obtain the different schemes.

We write the complete equation as

$$\frac{(1 + \gamma_n)(f^{n+1} - f^n)}{\Delta t^{n+1}} - \frac{\gamma_n(f^n - f^{n-1})}{\Delta t^n} = \frac{1}{R} (D^2 - \lambda^2) [(1 - \beta_i)f^n + \beta_i f^{n+1}] + [(1 - \beta_e^n)G(f^n) + \beta_e^n G(f^{n-1})]. \quad (24)$$

Divide by  $\beta_i$ , multiply with  $R$  and collect terms,

$$\begin{aligned} \left[ D^2 - \lambda^2 - R \frac{1 + \gamma_n}{\beta_i \Delta t^{n+1}} \right] f^{n+1} = & -\frac{1 - \beta_i}{\beta_i} [D^2 - \lambda^2] f^n - \\ & R \left( \frac{1 + \gamma_n}{\beta_i \Delta t^{n+1}} + \frac{\gamma_n}{\beta_i \Delta t^n} \right) f^n + R \frac{\gamma_n}{\beta_i \Delta t^n} f^{n-1} - \\ & R \frac{1 - \beta_e^n}{\beta_i} G(f^n) - R \frac{\beta_e^n}{\beta_i} G(f^{n-1}). \end{aligned} \quad (25)$$

Before we proceed to apply the above time advancement scheme to the general equations, we need to consider the effect of the variable time-step on the accuracy of the solution. When using Crank–Nicolson there is no problem with variable time-step since this scheme only involves two different time levels, but for BDF2 one has to consider the effects.

2.5. *The influence of the variable time step on the accuracy*

In this section  $h_n$  denotes the time step at time level  $n$ . For the case of BDF2 we approximate the time derivative of  $y$  with the following scheme:

$$\frac{\partial y}{\partial t} \approx \frac{3(y^{n+1} - y^n)}{2h_n} - \frac{y^n - y^{n-1}}{2h_{n-1}}. \quad (26)$$

Taylor expansion of  $y^n$  and  $y^{n-1}$  around time  $t^{n+1}$  yields

$$y^n = y^{n+1} - h_n y'(n+1) + \frac{h_n^2}{2} y''(n+1) - \frac{h_n^3}{6} y'''(n+1) + \mathcal{O}(h_n^4) \quad (27)$$

$$\begin{aligned} y^{n-1} = & y^{n+1} - (h_n + h_{n-1})y'(n+1) + \\ & \frac{(h_n + h_{n-1})^2}{2} y''(n+1) - \frac{(h_n + h_{n-1})^3}{6} y'''(n+1) + \mathcal{O}((h_n + h_{n-1})^4) \end{aligned} \quad (28)$$

where  $h_n$  is the time step at time level  $n$  and a prime denotes a time derivative. Putting these into the equation (26) gives

$$\begin{aligned} \frac{\partial y}{\partial t} \approx & y'(n+1) + y'' \left( -\frac{h_n}{4} + \frac{h_{n-1}}{4} \right) + \mathcal{O}(h_n^2) \\ = & y'(n+1) + y'' \frac{\epsilon h_n}{4} + \mathcal{O}(h_n^2) \end{aligned} \quad (29)$$

where we have assumed  $h_{n-1} = h_n + \epsilon h_n$ . The order in time is thus 2 if  $\epsilon$  is zero, i.e. for a constant time-step, and close to 2 if the variation is reasonably slow.

2.6. *Applied to the general case,  $\alpha, \beta > 0$* 

Here we describe time advancement scheme applied to the general case  $\alpha, \beta > 0$ , equations (9a) and (9b). For the other cases it is done in the same manner.

$$\begin{aligned} (D^2 - \lambda_A^2) \hat{\phi}_A^{n+1} = & -C_1 [D^2 - \lambda_A^2] \hat{\phi}_A^n - C_2 R r^2 \hat{\phi}_A^n + \\ & C_3 R r^2 \hat{\phi}_A^{n-1} - C_6 R r^2 \hat{h}_{u_A}^n - C_4 R r^2 \hat{h}_{u_A}^{n-1} \end{aligned} \quad (30a)$$

$$\begin{aligned} (D^2 - \lambda_B^2) \hat{\phi}_B^{n+1} = & -C_1 [D^2 - \lambda_B^2] \hat{\phi}_B^n - C_2 R r^2 \hat{\phi}_B^n + \\ & C_3 R r^2 \hat{\phi}_B^{n-1} - C_6 R r^2 \hat{h}_{u_B}^n - C_4 R r^2 \hat{h}_{u_B}^{n-1} \end{aligned} \quad (30b)$$

where

$$D^2 = r \frac{\partial}{\partial r} \left( r \frac{\partial}{\partial r} \right) = r^2 \frac{\partial^2}{\partial r^2} + r \frac{\partial}{\partial r} \quad (31a)$$

$$\lambda_A^2 = (\alpha + 1)^2 + r^2 (\beta^2 + RC_5) \quad (31b)$$

$$\lambda_B^2 = (\alpha - 1)^2 + r^2 (\beta^2 + RC_5). \quad (31c)$$

Define

$$\hat{f}_{\phi_i}^n = \hat{p}_{\phi_i}^n - C_6 r^2 R \hat{h}_{u_i}^n, \quad i = A, B \quad (32)$$

where

$$\begin{aligned}\hat{p}_{\phi_i}^n &= -\mathcal{C}_1 [D^2 - \lambda_i^2] \hat{\phi}_i^n - \mathcal{C}_2 Rr^2 \hat{\phi}_i^n + \mathcal{C}_3 r^2 R \hat{\phi}_i^{n-1} - \mathcal{C}_4 r^2 R \hat{h}_{u_i}^{n-1} = \\ &= -\mathcal{C}_1 \hat{f}_{\phi_i}^{n-1} - \mathcal{C}_2 Rr^2 \hat{\phi}_i^n + \mathcal{C}_3 Rr^2 \hat{\phi}_i^{n-1} - \mathcal{C}_4 Rr^2 \hat{h}_{u_i}^{n-1}, \quad i = A, B\end{aligned}\quad (33)$$

and

$$\begin{aligned}\mathcal{C}_1 &= \frac{1-\beta_i}{\beta_i} & \mathcal{C}_2 &= \frac{1+\gamma_n}{\Delta t^{n+1} \beta_i^2} + \frac{\gamma_n}{\Delta t^n \beta_i} \\ \mathcal{C}_3 &= \frac{\gamma_n}{\Delta t^n \beta_i} & \mathcal{C}_4 &= \frac{\beta_i^n}{\beta_i} \\ \mathcal{C}_5 &= \frac{1+\gamma_n}{\Delta t^{n+1} \beta_i} & \mathcal{C}_6 &= \frac{1-\beta_i^n}{\beta_i} \\ \mathcal{C}_7 &= \frac{1+\gamma_n}{\Delta t^{n+1} \beta_i} + \frac{\gamma_n}{\Delta t^n \beta_i}.\end{aligned}$$

We can then write equations (30a), (30b) as

$$(D^2 - \lambda_i^2) \hat{\phi}_i^{n+1} = \hat{f}_{\phi_i}^n, \quad i = A, B. \quad (34a)$$

These two equations are ordinary differential equations in  $r$  and can be solved, e.g. by expanding the functions involved in Chebyshev polynomials and solve the resulting equation system (see section 3).

From  $\hat{\phi}_A$  and  $\hat{\phi}_B$  we are able to calculate the  $\hat{u}_A$  and  $\hat{u}_B$  velocities. We multiply equation (9c) and (9d) with  $r^2$  to get

$$(D^2 - \gamma_i^2) \hat{u}_i^{n+1} = r^2 \hat{\phi}_i^{n+1}, \quad i = A, B \quad (34b)$$

where  $D^2$  is defined by (31a), and

$$\gamma_A^2 = (\alpha + 1)^2 + r^2 \beta^2 \quad (35)$$

$$\gamma_B^2 = (\alpha - 1)^2 + r^2 \beta^2. \quad (36)$$

Notice that the equations (34a) and (34b) are similar in structure and can be solved with the same numerical subroutine.

### 2.7. Boundary conditions and constants

A problem with the above equations (34a) <sub>$i=A,B$</sub>  and (34b) <sub>$i=A,B$</sub>  is that the boundary condition do not apply to the quantities for which we have differential equations. To remedy this, each of the quantities can be solved for a particular solution with the value at the preceding time level as boundary conditions. Then we solve for a homogeneous solution with non-zero boundary conditions. Finally the boundary conditions are fulfilled by a suitable linear combination of the particular and homogeneous solutions (see also Boyd (1989), p 188). For the case of annular pipe flow we solve for twice as many homogenous solutions. This is because of the coupling between the odd and even Chebyshev coefficients in the annular pipe flow case.

This approach of computing the constants is slightly different from the one chosen in the channel flow code, where the boundary condition for the particular solution is zero. Both approaches have been tried in the pipe flow code and the present approach allows at least a factor of ten smaller time steps

compared to the channel flow code approach. The time step is limited by the resolution near the wall. With a small time step the numerical boundary layers near the wall, arising from the solution of equation (37b) and (38b), become thin and for a small enough time step the resolution will be inadequate to resolve these thin layers.

## 2.8. Pipe flow

### 2.8.1. The general case, $\alpha, \beta > 0$

For all  $\alpha$  and  $\beta > 0$  we solve for a particular solution, fulfilling the correct right hand side with a boundary condition equal to the value at the preceding time level, and a homogenous solution with zero right hand side and boundary condition of unity. We then combine these two solutions to obtain a solution with correct boundary condition. Note that it is sufficient to prescribe the boundary condition of the solutions at  $r = 1$  without loss of generality, since the odd and even Chebyshev coefficients decouple (the boundary condition at  $r = -1$  follow automatically).

$$(D^2 - \lambda_A^2)\hat{\phi}_{A,p}^{n+1} = \hat{f}_{\phi_A}^n \quad \hat{\phi}_{A,p}^{n+1}(1) = \hat{\phi}_A^n(1) \quad (37a)$$

$$(D^2 - \lambda_A^2)\hat{\phi}_{A,h}^{n+1} = 0 \quad \hat{\phi}_{A,h}^{n+1}(1) = 1 \quad (37b)$$

$$(D^2 - \gamma_A^2)\hat{u}_{A,p}^{n+1} = r^2\hat{\phi}_{A,p}^{n+1} \quad \hat{u}_{A,p}^{n+1}(1) = 0 \quad (37c)$$

$$(D^2 - \gamma_A^2)\hat{u}_{A,h}^{n+1} = r^2\hat{\phi}_{A,h}^{n+1} \quad \hat{u}_{A,h}^{n+1}(1) = 0 \quad (37d)$$

$$(D^2 - \lambda_B^2)\hat{\phi}_{B,p}^{n+1} = \hat{f}_{\phi_B}^n \quad \hat{\phi}_{B,p}^{n+1}(1) = \hat{\phi}_B^n(1) \quad (38a)$$

$$(D^2 - \lambda_B^2)\hat{\phi}_{B,h}^{n+1} = 0 \quad \hat{\phi}_{B,h}^{n+1}(1) = 1 \quad (38b)$$

$$(D^2 - \gamma_B^2)\hat{u}_{B,p}^{n+1} = r^2\hat{\phi}_{B,p}^{n+1} \quad \hat{u}_{B,p}^{n+1}(1) = 0 \quad (38c)$$

$$(D^2 - \gamma_B^2)\hat{u}_{B,h}^{n+1} = r^2\hat{\phi}_{B,h}^{n+1} \quad \hat{u}_{B,h}^{n+1}(1) = 0 \quad (38d)$$

where subscripts  $p$  and  $h$  denote the particular and homogeneous parts, respectively.

From no-slip at the wall and continuity we obtain the following boundary condition

$$\frac{\partial \hat{u}}{\partial r} = 0, \text{ at } r = 1. \quad (39)$$

The constants are defined from

$$\hat{u}_A = \hat{u}_{A,p} + C_1\hat{u}_{A,h} \quad (40)$$

$$\hat{u}_B = \hat{u}_{B,p} + C_2\hat{u}_{B,h} \quad (41)$$

$$\hat{\phi}_A = \hat{\phi}_{A,p} + C_1\hat{\phi}_{A,h} \quad (42)$$

$$\hat{\phi}_B = \hat{\phi}_{B,p} + C_2\hat{\phi}_{B,h}. \quad (43)$$

At  $r = 1$  we have

$$\frac{\partial}{\partial r}(\hat{u}) = \frac{1}{2} \frac{\partial}{\partial r}(\hat{u}_A + \hat{u}_B) \Rightarrow \frac{\partial}{\partial r}(\hat{u}_{A,p} + \hat{u}_{B,p}) + C_1 \frac{\partial}{\partial r} \hat{u}_{A,h} + C_2 \frac{\partial}{\partial r} \hat{u}_{B,h} = 0. \quad (44)$$

This yields a relation between  $C_1$  and  $C_2$ . To get another relation we take the definition of  $\hat{\omega}_3$ , (12c)

$$\begin{aligned} \hat{\omega}_3^{n+1} &= \frac{i}{2r} \frac{\partial}{\partial r} [r(\hat{u}_B^{n+1} - \hat{u}_A^{n+1})] - \frac{i\alpha}{2r} (\hat{u}_A^{n+1} + \hat{u}_B^{n+1}) = \\ &= \frac{i}{2r} \frac{\partial}{\partial r} [r(\hat{u}_{B,p}^{n+1} + C_2 \hat{u}_{B,h}^{n+1} - \hat{u}_{A,p}^{n+1} - C_1 \hat{u}_{A,h}^{n+1})] - \\ &\quad \frac{i\alpha}{2r} (\hat{u}_{A,p}^{n+1} + C_1 \hat{u}_{A,h}^{n+1} + \hat{u}_{B,p}^{n+1} + C_2 \hat{u}_{B,h}^{n+1}) = \\ &= \{r = 1\} = \frac{i}{2} \frac{\partial}{\partial r} [\hat{u}_{B,p}^{n+1} + C_2 \hat{u}_{B,h}^{n+1} - \hat{u}_{A,p}^{n+1} - C_1 \hat{u}_{A,h}^{n+1}]. \end{aligned} \quad (45)$$

Now we have a relation between  $C_1$  and  $\hat{\omega}_3^{n+1}$  at  $r = 1$ , and we need another relation for  $\hat{\omega}_3^{n+1}$  at  $r = 1$ . This we get from equation (9e) by applying our time advancement scheme

$$\begin{aligned} \hat{\omega}_3^{n+1} &= \hat{\omega}_3^n + \frac{\Delta t^{n+1}}{1 + \gamma_n} \left[ \frac{\gamma_n}{\Delta t^n} (\hat{\omega}_3^n - \hat{\omega}_3^{n-1}) + \frac{1 - \beta_i}{R} \hat{L}_\omega^n + \right. \\ &\quad \frac{i\beta_i}{2R} (C_2(1 - \alpha + \hat{\phi}_{B,h}^{n+1}) - C_1(1 + \alpha + \hat{\phi}_{A,h}^{n+1}) - \\ &\quad \left. \hat{\phi}_A^n(1 + \alpha) + \hat{\phi}_B^n(1 - \alpha) + \hat{\phi}'_{B,p}{}^{n+1} - \hat{\phi}'_{A,p}{}^{n+1}) + \right. \\ &\quad \left. (1 - \beta_e^n) \hat{h}_{\omega_3}^n + \beta_e^n \hat{h}_{\omega_3}^{n-1} \right] \end{aligned} \quad (46)$$

where

$$\begin{aligned} \hat{L}_{\omega_3}^n &= \frac{1}{r} \frac{\partial r \hat{\phi}_2^n}{\partial r} - \frac{i\alpha}{r} \hat{\phi}_1^n \\ \hat{h}_{\omega_3}^n &= \frac{1}{r} \frac{\partial r \hat{H}_2^n}{\partial r} - \frac{i\alpha}{r} \hat{H}_1^n. \end{aligned}$$

This will finally yield a  $2 \times 2$  linear equation system,

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \quad (47)$$

where

$$\begin{aligned} a_{11} &= \hat{u}'_{A,h} & a_{12} &= \hat{u}'_{B,h} \\ a_{21} &= 1 + \alpha + \hat{\phi}'_{A,h} - RC_5 \hat{u}'_{A,h} & a_{22} &= -1 + \alpha - \hat{\phi}'_{B,h} + RC_5 \hat{u}'_{B,h} \\ r_1 &= -\hat{u}'_{A,p} - \hat{u}'_{B,p} & r_2 &= -2i\hat{p}_{\omega_3}^{n+1} - \hat{\phi}'_{A,p}{}^{n+1} + \hat{\phi}'_{B,p}{}^{n+1} - \hat{\phi}_A^n(1 + \alpha) + \\ & & & \hat{\phi}_B^n(1 - \alpha) + RC_5 \hat{u}'_{A,p}{}^{n+1} - RC_5 \hat{u}'_{B,p}{}^{n+1} \end{aligned}$$

and

$$\begin{aligned} \hat{p}_{\omega_3}^{n+1} = & R \left( \frac{1 + \gamma_n}{\Delta t^{n+1} \beta_i} + \frac{\gamma_n}{\beta_i \Delta t^n} \right) \hat{\omega}_3^n - R \frac{\gamma_n}{\beta_i \Delta t^n} \hat{\omega}_3^{n-1} + \\ & \frac{1 - \beta_i}{\beta_i} \hat{L}_\omega^n + R \frac{1 - \beta_e^n}{\beta_i} \hat{h}_{\omega_3}^n + R \frac{\beta_e^n}{\beta_i} \hat{h}_{\omega_3}^{n-1}. \end{aligned} \quad (48)$$

We are now able to calculate the required constants at the new time level  $n + 1$ , and update the velocities. We have to store the value of  $\hat{\omega}_3^{n-1}$ ,  $\hat{\omega}_3^n$ ,  $\hat{L}_\omega^n$ ,  $\hat{h}_{\omega_3}^n$ ,  $\hat{h}_{\omega_3}^{n-1}$ ,  $\hat{\phi}_A^n$  and  $\hat{\phi}_B^n$  at  $r = 1$  for every wave number.

### 2.8.2. The case $\alpha = 0$

In the axi-symmetric case we will solve for  $\hat{\phi}_1$ ,  $\hat{u}$  and  $\hat{v}$

$$(\mathbf{D}^2 - \lambda_A^2) \hat{\phi}_{1,p}^{n+1} = \hat{f}_{\phi_1}^n \quad \hat{\phi}_{1,p}^{n+1}(1) = \hat{\phi}_1^n(1), \quad (49a)$$

$$(\mathbf{D}^2 - \lambda_A^2) \hat{\phi}_{1,h}^{n+1} = 0 \quad \hat{\phi}_{1,h}^{n+1}(1) = 1, \quad (49b)$$

$$(\mathbf{D}^2 - \gamma_A^2) \hat{u}_p^{n+1} = r^2 \hat{\phi}_{1,p}^{n+1} \quad \hat{u}_p^{n+1}(1) = 0, \quad (49c)$$

$$(\mathbf{D}^2 - \gamma_A^2) \hat{u}_h^{n+1} = r^2 \hat{\phi}_{1,h}^{n+1} \quad \hat{u}_h^{n+1}(1) = 0, \quad (49d)$$

$$(\mathbf{D}^2 - \lambda_A^2) \hat{v}^{n+1} = \hat{f}_v^{n+1} \quad \hat{v}^{n+1}(1) = 0. \quad (49e)$$

The main difference here is that the equations for the constants  $C_1$   $C_2$  are different. Since we solve for  $\hat{v}$  directly we only need  $C_1$ , which we get directly from the boundary condition (39)

$$C_1 = -\frac{\hat{u}'_p}{\hat{u}'_h}. \quad (50)$$

### 2.8.3. The case $\beta = 0$

In this case we solve time evolution equations for  $\hat{w}$  and  $\hat{\omega}_3$ . By using the relations between  $\hat{\Xi}_3$  and  $\hat{\omega}_3$  we obtain  $\hat{u}$  and  $\hat{v}$ .

$$(\mathbf{D}^2 - \lambda_1^2) \hat{\omega}_{3,p}^{n+1} = \hat{f}_{\omega_3}^n \quad \hat{\omega}_{3,p}^{n+1}(1) = \hat{\omega}_3^n(1), \quad (51a)$$

$$(\mathbf{D}^2 - \lambda_1^2) \hat{\omega}_{3,h}^{n+1} = 0 \quad \hat{\omega}_{3,h}^{n+1}(1) = 1, \quad (51b)$$

$$(\mathbf{D}^2 - \gamma_1^2) \hat{\Xi}_{3,p}^{n+1} = -r^2 \hat{\omega}_{3,p}^{n+1} \quad \hat{\Xi}_{3,p}^{n+1}(1) = 0, \quad (51c)$$

$$(\mathbf{D}^2 - \gamma_1^2) \hat{\Xi}_{3,h}^{n+1} = -r^2 \hat{\omega}_{3,h}^{n+1} \quad \hat{\Xi}_{3,h}^{n+1}(1) = 0, \quad (51d)$$

$$(\mathbf{D}^2 - \lambda_1^2) \hat{w}^{n+1} = \hat{f}_w^n, \quad \hat{w}(1) = 0, \quad (51e)$$

and to obtain  $\hat{u}$  and  $\hat{v}$

$$\hat{u} = \frac{i\alpha}{r} \hat{\Xi}_3 = \frac{i\alpha}{r} \left( \hat{\Xi}_{3,p} + C_5 \hat{\Xi}_{3,h} \right) \quad (52)$$

$$\hat{v} = -\frac{\partial}{\partial r} \hat{\Xi}_3 = -\left( \frac{\partial}{\partial r} \hat{\Xi}_{3,p} + C_5 \frac{\partial}{\partial r} \hat{\Xi}_{3,h} \right) \quad (53)$$

and the constant  $C_5$  is easily obtained from (39) as

$$C_5 = -\frac{\hat{\Xi}'_{3,p}}{\hat{\Xi}'_{3,h}}. \quad (54)$$

#### 2.8.4. The case $\alpha = \beta = 0$

In this case we simply use the evolution equations for both  $\hat{v}$  and  $\hat{w}$ , (49e) and (51e).

### 2.9. Cylindrical geometry with finite inner radius

Flows that may be computed in this geometry include annular pipe flow, Taylor–Couette flow and curved channel flow. A change of variables from  $r$  to  $y = 2\frac{r-\mu}{\delta}$  transforms the flow domain from  $[r_i, 1]$  to  $[-1, 1]$ , to suit the use of Chebyshev polynomials.

#### 2.9.1. The general case, $\alpha, \beta > 0$

Here we solve for particular solutions in the same way as for regular pipe flow, but for the homogenous solution it is not enough to prescribe the boundary condition at one wall only, since the Chebyshev coefficients do not decouple. Hence we are forced to use two different homogenous solutions instead of one.

$$(D^2 - \lambda_A^2)\hat{\phi}_{A,p}^{n+1} = \hat{f}_{\phi_A}^n \quad \hat{\phi}_{A,p}^{n+1}(\pm 1) = \hat{\phi}_A^{n+1}(\pm 1) \quad (55a)$$

$$(D^2 - \lambda_A^2)\hat{\phi}_{A,h1}^{n+1} = 0 \quad \hat{\phi}_{A,h1}^{n+1}(-1) = 0, \hat{\phi}_{A,h1}^{n+1}(1) = 1 \quad (55b)$$

$$(D^2 - \lambda_A^2)\hat{\phi}_{A,h2}^{n+1} = 0 \quad \hat{\phi}_{A,h2}^{n+1}(-1) = 1, \hat{\phi}_{A,h2}^{n+1}(1) = 0 \quad (55c)$$

$$(D^2 - \gamma_A^2)\hat{u}_{A,p}^{n+1} = r^2\hat{\phi}_{A,p}^{n+1} \quad \hat{u}_{A,p}^{n+1}(\pm 1) = 0 \quad (55d)$$

$$(D^2 - \gamma_A^2)\hat{u}_{A,h1}^{n+1} = r^2\hat{\phi}_{A,h1}^{n+1} \quad \hat{u}_{A,h1}^{n+1}(\pm 1) = 0 \quad (55e)$$

$$(D^2 - \gamma_A^2)\hat{u}_{A,h2}^{n+1} = r^2\hat{\phi}_{A,h2}^{n+1} \quad \hat{u}_{A,h2}^{n+1}(\pm 1) = 0 \quad (55f)$$

with identical equations for  $B$ . The necessary constants are defined from

$$\hat{u}_A = \hat{u}_{A,p} + C_1\hat{u}_{A,h1} + C_2\hat{u}_{A,h2} \quad (56)$$

$$\hat{u}_B = \hat{u}_{B,p} + C_3\hat{u}_{B,h1} + C_4\hat{u}_{B,h2} \quad (57)$$

$$\hat{\phi}_A = \hat{\phi}_{A,p} + C_1\hat{\phi}_{A,h1} + C_2\hat{\phi}_{A,h2} \quad (58)$$

$$\hat{\phi}_B = \hat{\phi}_{B,p} + C_3\hat{\phi}_{B,h1} + C_4\hat{\phi}_{B,h2}. \quad (59)$$

$$(60)$$

The procedure for obtaining the equations necessary to compute the constants are the same as above, the only difference is that we now obtain a  $4 \times 4$  equation

system,

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} C_2 \\ C_4 \\ C_1 \\ C_3 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} \quad (61)$$

where the different coefficients are

$$\begin{aligned} a_{11} &= \hat{u}'_{A,h,2}{}^{n+1}(-1), & a_{21} &= -1 - r_0 \hat{\phi}'_{A,h,2}{}^{n+1}(-1) - \alpha + RC_5 r_0 \hat{u}'_{A,h,2}{}^{n+1}(-1) \\ a_{12} &= \hat{u}'_{B,h,2}{}^{n+1}(-1), & a_{22} &= 1 + r_0 \hat{\phi}'_{B,h,2}{}^{n+1}(-1) - \alpha - RC_5 r_0 \hat{u}'_{B,h,2}{}^{n+1}(-1) \\ a_{13} &= \hat{u}'_{A,h,1}{}^{n+1}(-1), & a_{23} &= -\hat{\phi}'_{A,h,1}{}^{n+1}(-1) + RC_5 \hat{u}'_{A,h,1}{}^{n+1}(-1) \\ a_{14} &= \hat{u}'_{B,h,1}{}^{n+1}(-1), & a_{24} &= \hat{\phi}'_{B,h,1}{}^{n+1}(-1) - RC_5 \hat{u}'_{B,h,1}{}^{n+1}(-1) \\ a_{31} &= \hat{u}'_{A,h,2}{}^{n+1}(1), & a_{41} &= -1 - \hat{\phi}'_{A,h,2}{}^{n+1}(1) - \alpha + RC_5 \hat{u}'_{A,h,2}{}^{n+1}(1) \\ a_{32} &= \hat{u}'_{B,h,2}{}^{n+1}(1), & a_{42} &= 1 + \hat{\phi}'_{B,h,2}{}^{n+1}(1) - \alpha - RC_5 \hat{u}'_{B,h,2}{}^{n+1}(1) \\ a_{33} &= \hat{u}'_{A,h,1}{}^{n+1}(1), & a_{43} &= -\hat{\phi}'_{A,h,1}{}^{n+1}(1) + RC_5 \hat{u}'_{A,h,1}{}^{n+1}(1) \\ a_{34} &= \hat{u}'_{B,h,1}{}^{n+1}(1), & a_{44} &= \hat{\phi}'_{B,h,1}{}^{n+1}(1) - RC_5 \hat{u}'_{B,h,1}{}^{n+1}(1) \end{aligned}$$

and the right hand side

$$\begin{aligned} r_1 &= -\hat{u}'_{A,p}{}^{n+1}(-1) - \hat{u}'_{B,p}{}^{n+1}(-1) \\ r_2 &= 2ir_0 p_{\omega_3}^{n+1}(-1) - r_0 \hat{\phi}'_{B,p}{}^{n+1}(-1) + r_0 \hat{\phi}'_{A,p}{}^{n+1}(-1) + \\ &\quad (1 + \alpha) \hat{\phi}'_A{}^n(-1) + (\alpha - 1) \hat{\phi}'_B{}^n(-1) + RC_5 r_0 \left[ \hat{u}'_{B,p}{}^{n+1}(-1) - \hat{u}'_{A,p}{}^{n+1}(-1) \right] \\ r_3 &= -\hat{u}'_{A,p}{}^{n+1}(1) - \hat{u}'_{B,p}{}^{n+1}(1) \\ r_4 &= 2ip_{\omega_3}^{n+1}(1) - \hat{\phi}'_{B,p}{}^{n+1}(1) + \hat{\phi}'_{A,p}{}^{n+1}(1) + \\ &\quad (1 + \alpha) \hat{\phi}'_A{}^n(1) + (\alpha - 1) \hat{\phi}'_B{}^n(1) + RC_5 \left[ \hat{u}'_{B,p}{}^{n+1}(1) - \hat{u}'_{A,p}{}^{n+1}(1) \right]. \end{aligned}$$

We are now able to calculate the required constants at the new time level  $n+1$  and update the velocities. As before we have to store the value of  $\hat{\omega}_3^{n-1}$ ,  $\hat{\omega}_3^n$ ,  $\hat{L}_\omega^n$ ,  $\hat{h}_{\omega_3}^n$ ,  $\hat{h}_{\omega_3}^{n-1}$ ,  $\hat{\phi}_A^n$  and  $\hat{\phi}_B^n$  but now at both walls,  $y = \pm 1$ , for every wave number.

The special cases  $\alpha = 0$  and  $\beta = 0$  are handled in the same manner as in the pipe flow case, taking into account the need for two homogenous solutions.

### 3. Chebyshev expansion

We will expand the variables in Chebyshev polynomials in the radial direction with  $r = [-1, 1]$ . In the case of annular pipe flow, we need to transform this range to  $[r_i, 1]$ , where  $r_i$  is the radius of the inner cylinder.

### 3.1. Pipe flow

The typical equation systems to solve are given by

$$(D^2 - \lambda_i^2)\hat{\psi} = \hat{f}_i, \quad i = A, B \quad (62)$$

or if we insert  $D^2$  from (31a) and  $\lambda^2$  from (31b)

$$\left( r^2 \frac{\partial^2}{\partial r^2} + r \frac{\partial}{\partial r} - \gamma_1 - r^2 \gamma_2 \right) \hat{\psi}_i = \hat{f}_i \quad (63)$$

where  $\gamma_1 = (\alpha \pm 1)^2$  and  $\gamma_2 = \beta^2 + RC_5$ . This is a coupled system of ordinary differential equations with known right hand sides.

We will here present two ways at solving this system. The first is to expand the second derivative of the unknown function, and integrate twice to get the function itself. This method is called the integration method, and was initially used in the channel flow code of [Lundbladh \*et al.\* \(1992\)](#). The main advantage with this approach is that there is no need to differentiate to get the first or second derivate, thereby avoiding an operation that can be shown to amplify numerical errors in the solution.

The second way to solve the system of ODE's is to expand for the function itself, and differentiate to get the derivatives needed. This method is referred to as the Chebyshev-Tau method and was used in the channel flow code of [Kim \*et al.\* \(1987\)](#).

#### 3.1.1. The integration method

In the integration method we expand the second derivative of the unknown function  $\hat{\psi}$  in a series of Chebyshev polynomials. We can write the expansion as

$$\frac{\partial^2 \hat{\psi}_n}{\partial r^2} = \sum_{j=0}^N \psi_{jn}^{(2)} T_j. \quad (64)$$

In the analysis we need the following two basic relations:

$$r T_j = \frac{1}{2} (T_{j+1} + T_{j-1}) \quad (65)$$

$$\int T_j dr = \frac{1}{2} \left( \frac{T_{j+1}}{j+1} - \frac{T_{j-1}}{j-1} \right) + \text{const.} \quad (66)$$

For further useful relation see [Appendix A](#). Deriving the relation between the Chebyshev coefficients is now straightforward. Use the relations (66) and (65) twice to expand the first term in (63), use both relations once to obtain the second term, and so on. A more thorough derivation of the equation system can be found in [Appendix B](#). For brevity we will here only show the general





3.2. *Cylindrical geometry with finite inner radius*

The integration method seems to be less suitable than the Chebyshev-Tau method for the cylindrical geometry, and therefore we will only present the Chebyshev-Tau method for the annular pipe flow case.

In order to be able to use the algorithm developed above for annular pipe flow, we need to change the radial coordinate from  $[r_i, 1]$  to  $[-1, 1]$ . Let us introduce the following change of variable

$$y = 2 \frac{r - \mu}{\delta} \tag{71}$$

where  $\mu = .5(r_i + 1)$  is the mean radius, and  $\delta = 1 - r_i$  is the gap width between the cylinders. Now we need to change all occurrences of  $r$  to  $y$  and  $\frac{\partial}{\partial r}$  to  $\frac{\partial}{\partial y}$ . It is possible, with considerable amount of algebra, see [Delabre & Trespé \(2000\)](#), to derive expressions similar to (68) and obtain a 9-diagonal system matrix. However, in this case the odd and even modes do not decouple, and we have to handle the full 9-diagonal matrix. This is easily realized by considering the relation for multiplying with  $r$ , (97), which involves both odd and even coefficients. Here we present the general form of the equations,

$$\begin{aligned} & -\psi_{n-4} \frac{1}{32} \frac{\gamma_2 \delta^2}{n-1} - \psi_{n-3} \frac{1}{4} \frac{\gamma_2 \delta \mu}{n-1} - \\ & \psi_{n-2} \frac{1}{16} \frac{\delta^2 \gamma_2 + 8\mu^2 \gamma_2 (n+1) + 8\gamma_1 (n+1) - 8(n^3 - 3n^2 + 4)}{(n-1)(n+1)} + \\ & \psi_{n-1} \frac{1}{4} \frac{\mu(16n^2 - 8n - 24 + \gamma_2 \delta^2)}{\delta(n+1)} + \\ & \psi_n \frac{1}{16} \frac{\gamma_2 \delta^4 n + 16\gamma_2 \delta^2 \mu^2 n + 16n^3 \delta^2 + 16\gamma_1 n \delta^2 - 32n\delta^2 - 128n\mu^2 + 128n^3 \mu^2}{\delta^2(n+1)(n-1)} + \\ & \psi_{n+1} \frac{1}{4} \frac{\mu(16n^2 + 8n - 24 + \gamma_2 \delta^2)}{\delta(n-1)} + \\ & \psi_{n+2} \frac{1}{16} \frac{\delta^2 \gamma_2 - 8\mu^2 \gamma_2 (n-1) - 8\gamma_1 (n-1) + 8(n^3 + 3n^2 - 4)}{(n-1)(n+1)} - \\ & \psi_{n+3} \frac{1}{4} \frac{\gamma_2 \delta \mu}{n+1} - \psi_{n+4} \frac{1}{32} \frac{\gamma_2 \delta^2}{(n+1)} = \\ & \frac{1}{2} \frac{1}{n-1} g_{n-2} - \frac{n}{(n+1)(n-1)} g_n + \frac{1}{2} \frac{1}{n+2} g_{n+3}. \tag{72} \end{aligned}$$

The above expression simplifies, in the case of  $r_i = -1$  (i.e.  $\mu = 0, \delta = 2$ ) to the same general expression as for the pipe flow.

The above described method is somewhat more efficient than the method described in [Moser \*et al.\* \(1983\)](#). Their expansion in the Chebyshev polynomials result in a banded matrix with 14 super- and 8 sub-diagonals, and a few offdiagonal elements. They report that the operation count for solving this matrix is  $235N$ . In our method the operation count is  $2(pq + 2)N =$

$2(4 \cdot 4 + 2)N = 36N$ , (Golub & van Loan 1989), for the above 9-diagonal matrix, and  $2(2 \cdot 2 + 1)N = 10N$  for the 5-diagonal matrix in the pipe flow case, where  $p$  is the number of sub-diagonals, and  $q$  is the number of superdiagonal. The extra  $2N$  ( $1N$ ) comes from the treatment of the boundary condition.

### 3.3. Boundary conditions

The boundary conditions can be expressed in terms of the coefficients in the Chebyshev series. The boundary condition at  $r = 1$  can be expressed as (see equation 99 in Appendix B) keeping in mind that  $T_j(1) = 1, \forall j$ :

$$\hat{\psi}_{r=1} = \psi_0^{(0)} + \psi_0^{(1)} + \psi_0^{(2)} \frac{1}{4} - \psi_1^{(2)} \frac{1}{12} - \psi_2^{(2)} \frac{7}{48} + \sum_{n=3}^{N-2} \psi_n^{(2)} \frac{3}{(n^2-1)(n^2-4)} - \frac{1}{2(N-3)(N-2)(N-1)} - \frac{1}{2(N-2)(N-1)N}, \quad (73)$$

for the integration method. For the Chebyshev-Tau method the boundary condition simply reads

$$\hat{\psi}_{r=1} = \sum_{n=0}^N \psi_n. \quad (74)$$

## 4. Numerical aspects concerning the system matrix

At first glance the integration method seems superior to the Chebyshev-Tau method, because in the integration method there is no need to take derivatives of the function, only to integrate. Differentiation is more sensitive to numerical errors than integration. The algorithm would, hence, seem to be less sensitive to numerical errors for the integration method than for the Chebyshev-Tau method. As it turns out the choice is not that simple. There are some subtle numerical aspects involved in the choice of Chebyshev method. One such aspect is the condition number of the resulting system matrix.

The condition number of a matrix is defined as (Golub & van Loan 1989),

$$\text{cond}(\mathbf{A}) \equiv \frac{\|\mathbf{A}\|}{\|\mathbf{A}^{-1}\|}$$

and the condition number goes to infinity if  $\mathbf{A}$  is singular. The relative error in the solution  $\mathbf{x}$  to the linear equation system  $\mathbf{A}\mathbf{x} = \mathbf{b}$  can be  $\text{cond}(\mathbf{A})$  times the relative error in  $\mathbf{A}$  and  $\mathbf{b}$ . The condition number of the matrix hence quantifies the sensitivity of the equation system.

The condition number of the system matrix for both the integration method and the Chebyshev-Tau method have been calculated for the present application, pipe flow, and is also compared with the channel flow case. The result is tabulated in table 4 for a Reynolds number of 4000 and 65 Chebyshev modes. A much lower condition number for the Chebyshev-tau method can be noted for both cases.

	integration	Chebyshev-tau
pipe	$3 \times 10^{10}$	$6 \times 10^4$
channel	$4 \times 10^{10}$	$5 \times 10^4$

TABLE 4. Condition number of the system matrix. pipe in the table refers to the present work, and channel refers to the code described in [Lundbladh \*et al.\* \(1992\)](#)

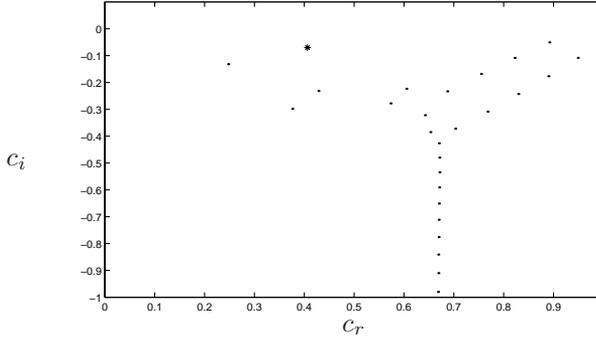


FIGURE 1. Eigenvalues for  $\alpha = 1$ ,  $\beta = 1$  and  $Re = 2000$ . The eigenvalue marked \* was used in the evaluation.

## 5. Verification

In order to verify the pipe flow code, we have studied the evolution of eigenmodes, the so called Orr–Sommerfeld–Squire modes. The solver for the eigenmodes was provided by Peter Schmid at University of Washington, and has been used in a study of pipe flow and growth of optimal disturbances by [Schmid & Henningson \(1993\)](#).

The flow variables are assumed to be of the following form

$$(u, v, w, p) = (\hat{u}, \hat{v}, \hat{w}, \hat{p})e^{i\alpha\varphi + i\beta z - i\omega t} \quad (75)$$

where  $\alpha$  is the azimuthal wave number and  $\beta$  is the streamwise wave number. The streamwise base flow is of the form  $W = W(r)$ . Following [Schmid & Henningson](#) the eigenvalue problem can be written in radial velocity, radial vorticity form as:

$$(-i\omega + i\beta W)T\Phi - \frac{i\beta}{r} \left( \frac{W'}{k^2 r} \right)' \Phi = \frac{1}{Re} T(k^2 r^2 T)\Phi + \frac{2\alpha^2 \beta}{Re} T\Omega \quad (76)$$

$$k^2 r^2 (-i\omega + i\beta W)\Omega + \frac{iW'}{r} \Phi = \frac{1}{Re} S\Omega + \frac{2\beta}{Re} T\Phi \quad (77)$$

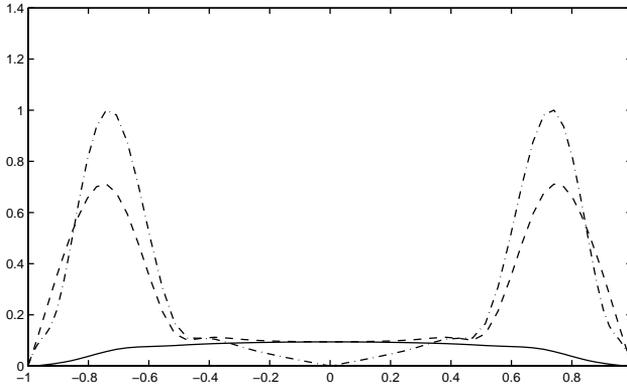


FIGURE 2. Velocities associated with the wall mode  $(c_r, c_i) = (0.4064, -0.06988)$ ,  $\alpha = 1$ ,  $\beta = 1$  and  $Re = 2000$ , the radial (—) ( $u$ ), the azimuthal ( $v$ ) (---) and the streamwise ( $w$ ) (- · -) velocity.

where

$$k^2 = \frac{\alpha^2}{r^2} + \beta^2 \quad (78)$$

$$T = \frac{1}{r^2} - \frac{1}{r} \frac{d}{dr} \left( \frac{1}{k^2 r} \frac{d}{dr} \right) \quad (79)$$

$$S = k^4 r^2 - \frac{1}{r} \frac{d}{dr} \left( k^2 r^3 \frac{d}{dr} \right) \quad (80)$$

$$\Phi \equiv -ir\hat{u} \quad (81)$$

$$\Omega \equiv \frac{\beta r \hat{v} - \alpha \hat{w}}{\alpha k^2 r^2}. \quad (82)$$

The velocities can now easily be calculated from the eigenmodes corresponding to the eigenvalue

$$\hat{u} = \frac{i\Phi}{r} \quad (83)$$

$$\hat{v} = -\frac{\alpha}{k^2 r^2} \frac{\partial \Phi}{\partial r} + \alpha \beta r \Omega \quad (84)$$

$$\hat{w} = -\frac{\beta}{k^2 r} \frac{\partial \Phi}{\partial r} - \alpha^2 \Omega. \quad (85)$$

### 5.1. Pipe flow

Before calculating the eigenmodes we specify the mean flow in the axial direction  $W(r) = 1 - r^2$ . In pipe flow all eigenmodes are damped and we have studied the least damped wall mode. The Reynolds number was 2000, and the azimuthal ( $\alpha$ ) and streamwise ( $\beta$ ) wavenumber for the eigenmodes was  $(1, 1)$ .

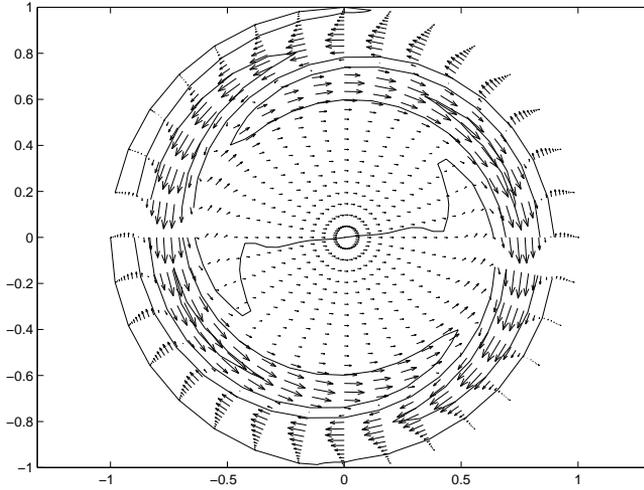


FIGURE 3. Velocity vectors in an  $r$ - $\varphi$ -plane, with the streamwise velocity as contours, showing the eigenmode.

The eigenvalues are depicted in figure 1. The eigenvalue for the wall mode we chose is marked with \*,  $(c_r, c_i) = (0.4064055, -0.06988367)$ .

The corresponding velocities are shown in figure 2. Figure 3 shows the radial and azimuthal velocities as velocity vectors in an  $r$ - $\varphi$ -plane, and contours of the streamwise velocity.

#### 5.1.1. The numerical simulation

We have used the pipe flow code to simulate the time evolution of the above shown eigenmode. The computational domain was  $(r, \varphi, z) = (1, 2\pi, 2\pi)$ , and the numerical resolution  $(nr, nt, nz) = (65, 32, 16)$ . The time advancement scheme used was the mixed explicit/implicit BDF2.

The initial amplitude of the eigenmode was set to a low value,  $10^{-6}$ , to avoid nonlinear effects. The energy of the eigenmode was monitored, and is shown in figure 4 for a time step of 0.07. This gives a CFL number of about 0.05. The decay of the energy follows the theoretical value within the resolution of the graphics.

When starting the time evolution we do not have access to more than the first time level, and must therefore use a first order time advancement scheme for the explicit part. We use a first order time advancement scheme for the first implicit time step also. The first time step is therefore Forward Euler/Backward Euler.

To verify the order of the combined explicit/implicit time advancement scheme, we did several different simulations and varied the time step, and calculated the damping rate of the eigenmode, see table 5 for the different

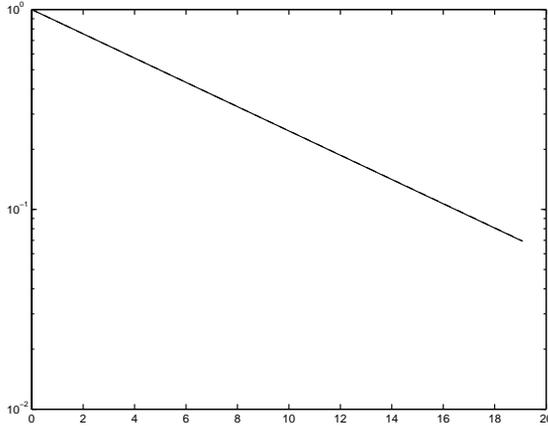


FIGURE 4. The energy of the eigenmode, as a function of time, where (—) is the numerical result, for a time step of 0.07, and (---) is the theoretical exponential decay  $-0.069883667$ .

simulations. We could then calculate the overall order in time. In order to avoid the errors introduced by the first time step being of only first order, we skipped the first few time steps in the evaluation of all damping rates.

Define the error  $\epsilon$  of the damping rate

$$\epsilon = \frac{|c_{i,num} - c_{i,theory}|}{c_{i,theory}},$$

and make the assumption that  $\epsilon = k(\Delta t)^p$ , where  $k$  is a constant, and  $p$  is the order of the time advancement scheme. We can now evaluate the order in time by

$$p = \ln\left(\frac{\Delta t_2}{\Delta t_1}\right) \ln\left(\frac{\epsilon_1}{\epsilon_2}\right).$$

For the resulting orders in time see table 6.

$\Delta t$	$c_i$	$\epsilon \times 10^3$
0.08	-0.0700120	1.838
0.07	-0.0699785	1.368
0.06	-0.06995377	1.003

TABLE 5. The error in the damping rate for various time steps, and the second order scheme.  $\epsilon$  is the normalised error in the damping rate,  $\epsilon = |c_{i,num} - c_{i,theory}|/c_{i,theory}$ .

$\Delta t_1$	$\Delta t_2$	$p$
0.08	0.07	2.2
0.07	0.06	2.01
0.08	0.06	2.1

TABLE 6. The order in time for various time steps.

### 5.2. Annular pipe flow

For the case with finite inner radius the eigenvalue solver had to be rewritten to an annular geometry. Since the solver uses a Chebyshev transformation in the radial direction, one only needs to change the definition of Chebyshev polynomials, and change the boundary condition at  $r_i$ . The mean axial velocity is now

$$W(r) = \frac{1}{4} \left( 1 - r^2 + \frac{r_i^2 - 1}{\ln(r_i)} \ln(r) \right).$$

There is one more parameter in the annular geometry, compared to pipe flow, namely the size of the gap between the inner and outer cylinders, or equivalently the ratio between the outer and the inner radius. The growth rate of the eigenmodes in annular pipe flow depends on this ratio, as well as the Reynolds number. In a recent study by [Shapiro \*et al.\* \(1999\)](#) the stability of annular pipe flow was investigated. They did both a linear stability analysis and calculated eigenvalue maps, and a non-linear analysis where they introduced a finite amplitude eigenfunction and studied the evolution in time with a DNS code.

#### 5.2.1. The numerical simulation

We have used the annular pipe flow code to simulate the time evolution of the above shown eigenmode. The computational domain, measured in radius of the outer cylinder, was  $(r, \varphi, z) = (0.3, 2\pi, 0.6\pi)$ , and the numerical resolution  $(nr, nt, nz) = (97, 32, 16)$ . The time advancement scheme used was the mixed explicit/implicit BDF2.

The Reynolds number based on half the gap width,  $\delta$ , was 10000, with  $\delta = 0.3$ . The azimuthal ( $\alpha$ ) and streamwise ( $\beta$ ) wavenumber for the eigenmodes was (1, 1). The eigenvalues are depicted in figure 5, and the wall mode we chose is marked with \*,  $(c_r, c_i) = (0.23794, 0.0017622)$ , and since  $c_i$  is positive it represents an unstable mode.

For this Reynolds number the flow becomes linearly unstable for  $r_i > 0.595$ , corresponding to  $\delta < 0.404$ .

The initial amplitude of the eigenmode was set to a low value,  $10^{-6}$ , to avoid nonlinear effects. The corresponding velocities are shown in figure 6, and figure 7 shows the radial and azimuthal velocities as velocity vectors in an  $r$ - $\varphi$ -plane, complemented with contours of the streamwise velocity.

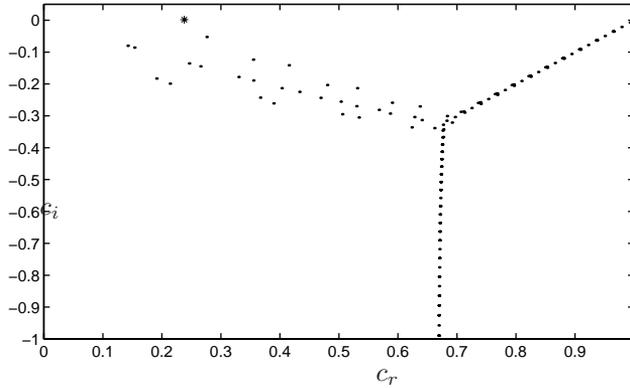


FIGURE 5. Eigenvalues for  $\alpha = 1$ ,  $\beta = 1$  and  $Re = 10000$ . The eigenvalue marked \* was used in the evaluation.

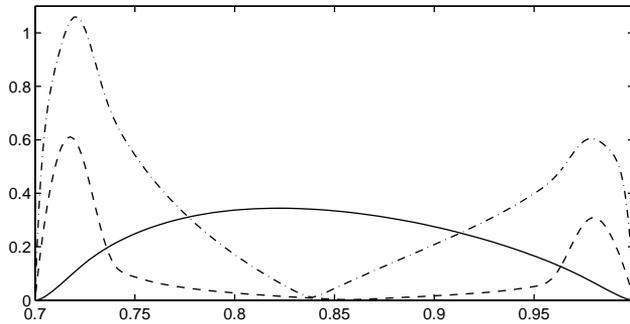


FIGURE 6. Velocities associated with the wall mode  $(c_r, c_i) =$ ,  $\alpha = 1$ ,  $\beta = 1$  and  $Re = 10000$ , the radial ( $—$ ) ( $u$ ), the azimuthal ( $- -$ ) ( $v$ ) and the streamwise ( $- \cdot -$ ) velocity.

In figure 8 we can see the theoretical and numerically obtained solution at a time  $T = 6.7$ , and they show excellent agreement.

## 6. Conclusions

We have developed an algorithm for solving the Navier–Stokes equations in cylindrical coordinates. Based on that algorithm we have implemented both a pipe flow code and an annular pipe flow code.

We have verified both the pipe flow code and the annular pipe flow code, by simulating the time evolution of an Orr–Sommerfeld–Squire mode, and compared the damping rate with the theoretical value. They are in excellent agreement. Furthermore we have verified that the mixed explicit/implicit time advancement scheme we used is second order in time.

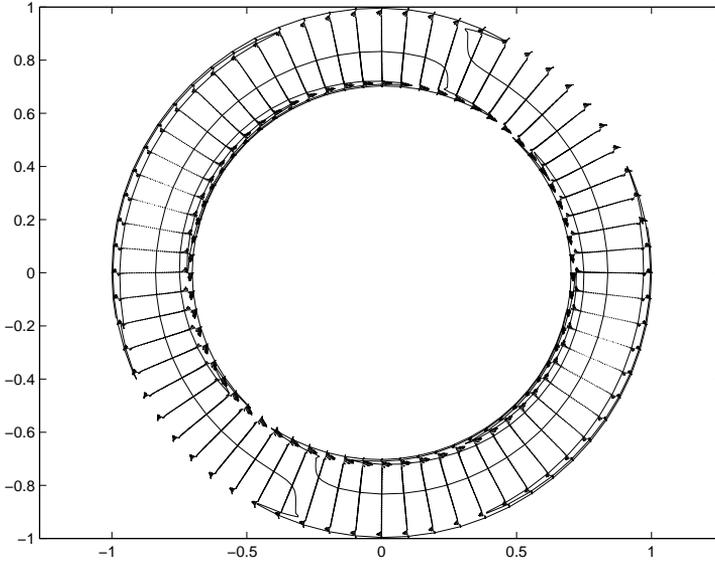


FIGURE 7. Velocity vectors in an  $r$ - $\varphi$ -plane, with the streamwise velocity as contours, showing the eigenmode.

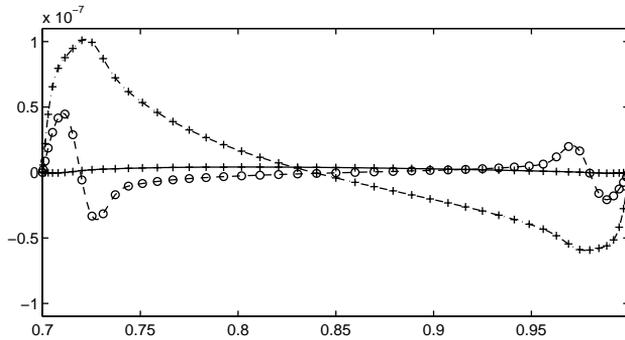


FIGURE 8. Velocities associated with the wall mode  $(c_r, c_i) =$ ,  $\alpha = 1$ ,  $\beta = 1$  and  $Re = 10000$ , the radial ( $u$ ) (—), the azimuthal ( $v$ ) (---) and the streamwise ( $w$ ) (- · -) velocity.

## 7. Future work

There are some remaining issues with the algorithm, for the case of pipe flow. When including the origin in the computational domain one has to be careful with the behaviour of the velocities when approaching the origin, i.e. when  $r \rightarrow 0$ . There are some constraints that the velocities have to obey to get a regular, well behaved velocity field near the origin, see [Appendix D](#). While the

initial velocity field is well behaved, numerical errors inevitably destroys this property if not imposed on the solution.

The condition are automatically fulfilled for  $\alpha = 0$  and 1, but for higher  $\alpha$  values we should implement some constraint on the velocities in order to fulfil the regularity condition. As an example, for  $\alpha = 3$ ,  $u_A$  is even and zero at  $r = 0$  and  $\frac{\partial^2}{\partial r^2}u_A = 0$ . With the algorithm we are only guaranteed to get  $u_A = 0$ , but not to have a zero second derivative.

In this report we have not tried to avoid the time stepping problem originating from the dense grid in the azimuthal direction, near the origin. As we remarked in the introduction it is possible to filter the solution near the origin. Since we are using spectral methods it is easy to filter in the azimuthal direction. However, care must be taken in the radial direction, where we have the Chebyshev expansion. The filter function must be a smooth function in the radial direction in order to retain the spectral accuracy. These issues need to be addressed in a future study.

These issues are not of concern in the annular pipe flow code, since the origin is excluded from the computational domain. In our tests we have noticed that the annular pipe flow code is less sensitive to errors in the implementation, and the present code can be used in future studies of annular pipe flow.

## 8. Acknowledgement

The authors would like to thank Peter Schmid at the University of Washington for providing us with the eigenvalue solver used in the verification. It proved to be very valuable in the verification of the code. We would also like to thank François Delabre and Fabien Treppe at École Polytechnique in Paris for their work with the introduction of the inner cylinder.

## References

- AKSELVOLL, K. & MOIN, P. 1995 An efficient method for temporal integration of the Navier–Stokes equations in confined axisymmetric geometries. *J. Comp. Phys.* **125**, 454–463.
- ANDERECK, C. D., LIU, S. S. & SWINNEY, H. L. 1986 Flow regimes in a circular Couette system with independently rotating cylinders. *J. Fluid Mech.* **164**, 155–183.
- ANTONIA, R. A., TEITEL, M. K. J. & BROWN, L. W. B. 1992 Low Reynolds number effects in a fully developed turbulent channel flow. *J. Fluid Mech.* **236**, 579–605.
- AZIZ, K. & HELLUMS, J. D. 1967 Numerical solution of the three-dimensional equations of motion for laminar natural convection. *Phys. Fluids* **10** (2), 314–324.
- BATCHELOR, G. K. & GILL, A. E. 1962 Analysis of the stability of axisymmetric jets. *J. Fluid Mech.* **14**, 529–550.
- BOYD, J. P. 1989 *Chebyshev and Fourier spectral methods. Lecture Notes in Engineering* 49. Berlin New York: Springer-Verlag.

- CANUTO, C., HUSSAINI, M. Y., QUARTERNI, A. & ZANG, T. A. 1988 *Spectral Methods in Fluid Dynamics*. Springer series in computational physics. Berlin New York: Springer-Verlag.
- CHOSSAT, P. & IOOSS, G. 1994 *The Couette–Taylor problem*. Applied mathematical sciences 102. Berlin Heidelberg New York: Springer-Verlag.
- DELABRE, F. & TREMPÉ, F. 2000 Development of a DNS code for annular geometries. Master's thesis, Dept. of Mechanics, Royal Institute of Technology, Stockholm, TRITA-MEK 2000:09.
- DENNIS, S. C. R., INGHAM, D. B. & COOK, R. N. 1979 Finite-difference methods for calculating steady incompressible flows in three dimensions. *J. Comp. Phys.* **33** (3), 325–339.
- EGGELS, J. G. M., UNGER, F., WEISS, M. H., WESTERWEEL, J., ADRIAN, R. J., FRIEDRICH, R. & NIEUWSTADT, F. T. M. 1994 Fully developed turbulent pipe flow: A comparison between direct numerical simulations and experiments. *J. Fluid Mech.* **268**, 175–209.
- FLETCHER, C. A. J. 1991 *Computational Techniques for fluid dynamics, Volume I*, 2nd edn., chap. 7. Springer series in computational physics. Springer-Verlag.
- FONTAINE, A. A. & DEUTCH, S. 1995 Three-component, time-resolved velocity statistics in the wall region of a turbulent pipe flow. *Exps. Fluids* **18** (3), 168–173.
- GARG, V. K. & ROULEAU, W. T. 1972 Linear stability of pipe Poiseuille flow. *J. Fluid Mech.* **54**, 113–127.
- GOLUB, G. H. & VAN LOAN, C. F. 1989 *Matrix computations*, 2nd edn. Johns Hopkins series in the mathematical sciences 3. Baltimore: The John Hopkins University Press.
- GRESHO, P. M. & SANI, R. L. 1987 On pressure boundary conditions for the incompressible Navier–Stokes equations. *Int. J. Numer. Methods Fluids* **7**, 1111–1145.
- KARLSSON, R. I. & JOHANSSON, T. G. 1988 LDV measurements of higher order moments in a turbulent boundary layer. In *Laser anemometry in fluid mechanics*. Lisbon, Ladoan-Instituto Superior Tecnico.
- KIM, J., MOIN, P. & MOSER, R. 1987 Turbulence statistics in fully developed channel flow. *J. Fluid Mech.* **177**, 133–166.
- LAUFER, C. J. 1954 The structure of turbulence in fully developed pipe flow. Technical Report NACA Reports 1158–1209. NACA.
- LAWN, C. J. 1971 The determination of the rate of dissipation in turbulent pipe flow. *J. Fluid Mech.* **48**, 477–505.
- LEONARD, A. & WRAY, A. 1982 A new numerical method for the simulation of three-dimensional flow in a pipe. In *Proc. Int. Conf. Numer. Methods in Fluid Dyn.*, pp. 335–341. Berlin: Springer-Verlag.
- LEUPTOW, R. M., DOCTER, A. & MIN, K. 1992 Stability of axial flow in an annulus with rotating inner cylinder. *Phys. Fluids A* **4** (11), 2446–2455.
- LOULOU, P. 1996 Direct numerical simulation of incompressible pipe flow using B-spline spectral method. Phd thesis, Stanford University.
- LUNDBLADH, A., HENNINGSON, D. S. & JOHANSSON, A. V. 1992 An efficient spectral integration method for the solution of the Navier–Stokes equation. *Tech. Rep.* TN 1992-28. FFA, Stockholm.
- SCHWARTZ-VAN MANEN, A. D. & NIEUWSTADT, F. T. M. 1996 Large-scale structures

- in a numerical and experimental low-Reynolds number turbulent pipe flow. *Eur. J. Mech. B/Fluids* **15** (6), 897–915.
- MOCHIZUKI, S. & NIEUWSTADT, F. T. M. 1996 Reynolds-number-dependence of the maximum in the streamwise velocity fluctuations in wall turbulence. *Exps. Fluids* **21**, 218–226.
- MOSER, R. D., MOIN, P. & LEONARD, A. 1983 A spectral method for the Navier–Stokes equations with application to Taylor–Couette flow. *J. Comp. Phys.* **52**, 524–544.
- ORLANDI, P. 1997 Helicity fluctuations and turbulent energy production in rotating and non-rotating pipes. *Phys. Fluids* **9** (7), 2045–2056.
- ORLANDI, P. & FATICA, M. 1997 Direct simulation of turbulent flow in a pipe rotating about its axis. *J. Fluid Mech.* **343**, 43–72.
- ORSZAG, S. A. & PATERA, A. T. 1983 Secondary instability of wall-bounded shear flow. *J. Fluid Mech.* **128**, 347–385.
- PATERA, A. T. & ORSZAG, S. A. 1981 Finite-amplitude stability of axisymmetric pipe flow. *J. Fluid Mech.* **112**, 467–474.
- PERRY, A. E. & ABELL, C. J. 1975 Scaling laws for pipe-flow turbulence. *J. Fluid Mech.* **67**, 257–271.
- REICH, G. & BEER, H. 1989 Fluid flow and heat transfer in an axially rotating pipe –I. Effect of rotation on turbulent pipe flow. *J. Heat Mass. Transfer* **32** (3), 551–562.
- REYNOLDS, O. 1883 An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and the law of resistance in parallel channels. *Phil. Trans. Roy. Soc. Lond.* **174**, 935–982.
- SALWEN, H. & GROSCH, C. E. 1972 The stability of Poiseuille flow in a pipe of circular cross-section. *J. Fluid Mech.* **54**, 93–112.
- SCHMID, P. J. & HENNINGSON, D. S. 1993 Optimal energy density growth in Hagen–Poiseuille flow. *J. Fluid Mech.* **277**, 197–225.
- SCHUMANN, U. 1975 Subgrid scale models for finite difference simulations of turbulent flows in plane channel and annuli. *J. Comp. Phys.* **18** (4), 376–404.
- SHANDS, J., ALFREDSSON, H. & LINDGREN, E. R. 1980 Annular pipe flow subject to axial motion of the inner boundary. *Phys. Fluids* **23** (10), 2144–2145.
- SHAPIRO, I., SHTILMAN, L. & TUMIN, A. 1999 On stability of flow in an annular channel. *Phys. Fluids* **11** (10), 2084–2992.
- TAKEUCHI, D. I. & JANKOWSKI, D. F. 1981 A numerical and experimental investigation of the stability of spiral Poiseuille flow. *J. Fluid Mech.* **102**, 101–126.
- DEN TOONDER, J. M. J. & NIEUWSTADT, F. T. M. 1997 Reynolds number effects in a turbulent pipe flow for low to moderate Re. *Phys. Fluids* **9** (11), 3398–3409.
- TOWENS, H. W., GOW, J. L., POWE, R. E. & WEBER, N. 1972 Turbulent flow in smooth and rough pipes. *Trans. ASME J. of Basic Engr.* **6**, 353–362.
- UMSHEID, L. & SANKAR-RAO, M. 1971 Further tests of a grid system for global numerical prediction. *Mon. Weather Rev.* **99** (9), 686–690.
- UNGER, F., EGGELS, J. G. M., FRIEDRICH, R. & NIEUWSTADT, F. T. M. 1993 On second and higher order statistics in fully developed turbulent pipe flow. In *Ninth symposium on turbulent shear flows*, pp. 2.1.1–2.1.6. Kyoto, Japan.
- VERZICCO, R. & ORLANDI, P. 1996 A finite difference scheme for direct simulation in cylindrical coordinates. *J. Comp. Phys.* **123**, 402–413.

- WERELEY, S. T. & LEUPTOW, R. M. 1999 Velocity field for Taylor–Couette flow with an axial flow. *Phys. Fluids* **11** (12), 3637–3649.
- WESTERWEEL, J., ADRIAN, R. J., EGGELS, J. G. M. & NIEUWSTADT, F. T. M. 1992 Measurements with particle image velocimetry of fully turbulent pipe flow at low Reynolds number. In *Proc. 6th Int. Symp. on Applications of Laser Tech. to Fluid Mechanics*. Lisbon, Portugal.
- ZAGAROLA, M. V. & SMITS, A. J. 1998 Mean-flow scaling of turbulent pipe flow. *J. Fluid Mech.* **373**, 33–79.
- ZHANG, Y., GANDHI, A., TOMBOULIDES, A. G. & ORSZAG, S. A. 1994 Simulation of pipe flow. In *Application of direct and large eddy simulation to transition and turbulence*, pp. 17.1–17.9. AGARD Conf, Creta, Greece.

## Appendix A. Useful relations for Chebyshev polynomials

The Chebyshev polynomials of the first kind represent the eigenfunctions of a Sturm–Liouville problem, [Camuto \*et al.\* \(1988\)](#). They can be defined through a recursion formula:

$$2xT_j(x) = T_{j-1}(x) + T_{j+1}(x). \quad (86)$$

The recursion is started by  $T_0(x) = 0$  and  $T_1(x) = x$ . With a change of variables one can write the Chebyshev polynomials as

$$T_j(x) = \cos(k\theta), \quad \theta = \arccos x.$$

Thus, the Chebyshev polynomials are cosine functions after a change of variables, and one can use Fast Cosine Transforms for transformations between physical and Chebyshev space.

A regular function can be expanded as a sum of Chebyshev polynomials between  $x = 1$  and  $x = -1$ ,

$$f(x) = \sum_{n=0}^{+\infty} a_n T_n(x), \quad (87)$$

and the values of  $a_n$  can be computed from

$$a_n = \frac{2}{\pi c_n} \int_{-1}^1 \frac{f(x) T_n(x)}{\sqrt{1-x^2}} dx, \quad (88)$$

where

$$\begin{cases} c_0 = 2 \\ c_n = 1 \text{ if } n > 0. \end{cases} \quad (89)$$

The boundary condition at  $-1$  or  $1$  can be written as a relation between the different Chebyshev coefficients:

$$f(-1) = \sum_{n=0}^{+\infty} (-1)^n a_n = BC_{-1}, \quad f(1) = \sum_{n=0}^{+\infty} a_n = BC_1. \quad (90)$$

It is possible to find the Chebyshev coefficients of a function multiplied by  $x$  if we know the coefficients of the original function,

$$g(x) = xf(x) = \sum_{n=0}^{+\infty} b_n T_n(x) \quad (91)$$

with

$$b_n = \frac{1}{2} (c_{n-1} a_{n-1} + c_{n+1} a_{n+1}). \quad (92)$$

To integrate in Chebyshev space:

$$g(x) = \int f(x) dx = \sum_{n=0}^{+\infty} b_n T_n(x) \quad (93)$$

where

$$b_n = \frac{1}{2n}(c_{n-1}a_{n-1} - c_{n+1}a_{n+1}). \quad (94)$$

#### Appendix A.1. Change of domain

In the annular pipe flow case we make a change of variables from  $r$  to  $y$ :

$$r = \mu + \frac{\delta}{2}y \quad (95)$$

and expand

$$f(y) = \sum_{n=0}^{+\infty} a_n T_n(y). \quad (96)$$

By using equation (92), we can establish how the Chebyshev coefficients are changed when a function is multiplied by  $r$ , i.e. the coefficients for  $rf(y)$  are given by

$$b_n = \mu a_n + \frac{\delta}{4}(c_{n-1}a_{n-1} + c_{n+1}a_{n+1}). \quad (97)$$

## Appendix B. Chebyshev integration method

Using the relation (94) to integrate equation (64) we get:

$$\frac{\partial \hat{\psi}_n}{\partial r} = \psi_0^{(1)} T_0 + \psi_0^{(2)} T_1 + \psi_1^{(2)} \frac{1}{4} T_2 + \sum_{j=2}^N \psi_j^{(2)} \frac{1}{2} \left( \frac{T_{j+1}}{j+1} - \frac{T_{j-1}}{j-1} \right). \quad (98)$$

Integrate one more time to obtain  $\hat{\psi}_n$ :

$$\begin{aligned} \hat{\psi}_n = & \psi_0^{(0)} T_0 + \psi_0^{(1)} T_1 + \frac{\psi_0^{(2)}}{4} T_2 + \frac{\psi_1^{(2)}}{8} \left( \frac{1}{3} T_3 - T_1 \right) + \frac{\psi_2^{(2)}}{8} \left( \frac{1}{6} T_4 - \frac{4}{3} T_2 \right) + \\ & \sum_{j=3}^N \psi_j^{(2)} \frac{1}{4} \left( \frac{T_{j+2}}{(j+2)(j+1)} - \frac{2T_j}{j^2-1} + \frac{T_{j-2}}{(j-1)(j-2)} \right). \end{aligned} \quad (99)$$

Multiply equation (99) with  $r^2$ , and use relation (65) twice:

$$\begin{aligned} r^2 \hat{\psi}_n = & \psi_0^{(0)} \frac{1}{2} (T_2 + T_0) + \psi_0^{(1)} \frac{1}{4} (T_3 + 3T_1) + \psi_0^{(2)} \frac{1}{16} (T_4 + 2T_2 + T_0) + \\ & \psi_1^{(2)} \frac{1}{96} (T_5 - T_3 - 8T_1) + \psi_2^{(2)} \frac{1}{192} (T_6 - 6T_4 - 15T_2 - 8T_0) + \\ & \psi_3^{(2)} \frac{1}{320} (T_7 - 3T_5 + T_3 + 25T_1) + \sum_{j=4}^N \psi_j^{(2)} \frac{1}{16} \left( \frac{T_{j+4}}{(j+1)(j+2)} - \right. \\ & \left. \frac{6T_{j+2}}{(j+2)(j^2-1)} - \frac{2(j^2-10)T_j}{(j^2-1)(j^2-4)} + \frac{6T_{j-2}}{(j^2-1)(j-2)} + \frac{T_{j-4}}{(j-1)(j-2)} \right). \end{aligned} \quad (100)$$

Multiply equation (98) with  $r$ :

$$r \frac{\partial \hat{\psi}_n}{\partial r} = \psi_0^{(1)} T_1 + \psi_0^{(2)} \frac{1}{2} (T_2 + T_0) + \psi_1^{(2)} \frac{1}{8} (T_3 + T_1) + \sum_{j=2}^N \psi_j^{(2)} \frac{1}{4} \left( \frac{T_{j+2}}{j+1} - \frac{2T_j}{j^2-1} - \frac{T_{j-2}}{j-1} \right). \quad (101)$$

Multiply equation (64) with  $r^2$ :

$$r^2 \frac{\partial^2 \hat{\psi}_n}{\partial r^2} = \frac{\psi_0^{(2)}}{4} (2T_2 + 2T_0) + \frac{\psi_1^{(2)}}{4} (T_3 + 3T_1) + \sum_{j=2}^N \psi_j^{(2)} \frac{1}{4} (T_{j+2} + 2T_j + T_{j-2}). \quad (102)$$

Putting it all together and collecting terms results in the following system of equations for the Chebyshev polynomials  $T_0$ - $T_N$ , with coefficients  $\psi_j^{(2)}$  as unknowns:

$$\begin{aligned} n = 0 : & -\psi_0^{(0)} \left( \frac{\gamma_2}{2} + \gamma_1 \right) + \psi_0^{(2)} \left( \frac{1}{2} + \frac{1}{2} - \frac{\gamma_2}{16} \right) + \psi_2^{(2)} \left( \frac{1}{4} - \frac{1}{4} + \frac{\gamma_2}{24} \right) = g_0 \\ n = 1 : & \psi_0^{(1)} \left( 1 - \frac{3\gamma_2}{4} - \gamma_1 \right) + \psi_1^{(2)} \left( \frac{3}{4} + \frac{1}{8} + \frac{\gamma_2}{12} + \frac{\gamma_1}{8} \right) + \\ & \psi_3^{(2)} \left( \frac{1}{4} - \frac{1}{8} - \frac{5\gamma_2}{64} - \frac{\gamma_1}{8} \right) - \psi_5^{(2)} \frac{\gamma_2}{192} = g_1 \\ n = 2 : & -\psi_0^{(0)} \frac{\gamma_2}{2} + \psi_0^{(2)} \left( \frac{1}{2} + \frac{1}{2} - \frac{\gamma_2}{8} - \frac{\gamma_1}{4} \right) + \psi_2^{(2)} \left( \frac{1}{2} - \frac{1}{6} + \frac{5\gamma_2}{64} + \frac{\gamma_1}{6} \right) + \\ & \psi_4^{(2)} \left( \frac{1}{4} - \frac{1}{12} - \frac{\gamma_2}{80} - \frac{\gamma_1}{24} \right) - \psi_6^{(2)} \frac{\gamma_2}{320} = g_2 \\ n = 3 : & -\psi_0^{(1)} \frac{\gamma_2}{4} + \psi_1^{(2)} \left( \frac{1}{4} + \frac{1}{8} + \frac{\gamma_2}{96} - \frac{\gamma_1}{24} \right) + \psi_3^{(2)} \left( \frac{1}{2} - \frac{1}{16} - \frac{\gamma_2}{320} + \frac{\gamma_1}{16} \right) + \\ & \psi_5^{(2)} \left( \frac{1}{4} - \frac{1}{16} - \frac{\gamma_2}{192} - \frac{\gamma_1}{48} \right) - \psi_7^{(2)} \frac{\gamma_2}{480} = g_3 \\ n = 4 : & -\psi_0^{(2)} \frac{\gamma_2}{16} + \psi_2^{(2)} \left( \frac{1}{4} + \frac{1}{12} + \frac{\gamma_2}{32} - \frac{\gamma_1}{48} \right) + \psi_4^{(2)} \left( \frac{1}{2} - \frac{1}{30} + \frac{\gamma_2}{240} + \frac{\gamma_1}{30} \right) + \\ & \psi_6^{(2)} \left( \frac{1}{4} - \frac{1}{20} - \frac{3\gamma_2}{1120} - \frac{\gamma_1}{80} \right) - \psi_8^{(2)} \frac{\gamma_2}{672} = g_4 \end{aligned}$$

$$\begin{aligned}
5 \leq n \leq N : & -\psi_{n-4}^{(2)} \frac{\gamma_2}{16(n-3)(n-2)} + \\
& \psi_{n-2}^{(2)} \left( \frac{1}{4} + \frac{1}{4(n-1)} + \frac{3\gamma_2}{8n(n-3)(n-1)} - \frac{\gamma_1}{4(n-1)n} \right) + \\
& \psi_n^{(2)} \left( \frac{1}{2} - \frac{1}{2(n^2-1)} + \frac{\gamma_2(n^2-10)}{8(n^2-1)(n^2-4)} + \frac{\gamma_1}{2(n^2-1)} \right) + \\
& \psi_{n+2}^{(2)} \left( \frac{1}{4} - \frac{1}{4(n+1)} - \frac{3\gamma_2}{8n(n+1)(n+3)} - \frac{\gamma_1}{4(n+1)n} \right) - \\
& \psi_{n+4}^{(2)} \frac{\gamma_2}{16(n+3)(n+2)} = g_n.
\end{aligned}$$

Actually, for the highest values of  $n$  the above equation is modified in a way that depends on the specific method of truncation of the series.

### Appendix C. Chebyshev-Tau method

Using the recursion relation (69) twice to obtain a relation between  $\psi_{n-2}$ ,  $\psi_n$  and  $\psi_{n+2}$ , and using relation (91) twice to handle the  $r^2$  multiplication yields a relation between  $\psi_{n-4}$ ,  $\psi_{n-2}$ ,  $\psi_n$ ,  $\psi_{n+2}$  and  $\psi_{n+4}$ . (Notice that there is an error in the derivation of the equation for  $n = 2$  in [Canuto \*et al.\* \(1988\)](#).)

$$\begin{aligned}
n = 2 : & \psi_0 \left[ -\gamma_1 + \gamma_2 \frac{1}{6} \right] + \psi_2 \left[ \frac{8}{6} + \gamma_1 \frac{4}{6} + \gamma_2 \frac{1}{24} \right] + \\
& \psi_4 \left[ \frac{16}{6} - \gamma_1 \frac{1}{6} + \gamma_2 \frac{1}{12} \right] - \psi_6 \gamma_2 \frac{1}{24} = g_0 - \frac{2}{3}g_2 + \frac{1}{6}g_4 \\
n = 3 : & \psi_1 \left[ \frac{1}{4} - \gamma_1 \frac{1}{4} - \gamma_2 \frac{3}{32} \right] + \psi_3 \left[ \frac{21}{8} + \gamma_1 \frac{3}{8} + \gamma_2 \frac{3}{32} \right] + \\
& \psi_5 \left[ \frac{25}{8} - \gamma_1 \frac{1}{8} + \gamma_2 \frac{1}{32} \right] - \psi_7 \gamma_2 \frac{1}{32} = \frac{1}{4}g_1 - \frac{3}{8}g_3 + \frac{1}{8}g_5 \\
n = 4 : & -\psi_0 \gamma_2 \frac{1}{12} + \psi_2 \left[ \frac{4}{6} - \gamma_1 \frac{1}{6} - \gamma_2 \frac{1}{60} \right] + \psi_4 \left[ 4 \frac{14}{15} + \gamma_1 \frac{4}{15} + \gamma_2 \frac{1}{15} \right] + \\
& \psi_6 \left[ \frac{36}{10} - \gamma_1 \frac{1}{10} + \gamma_2 \frac{1}{60} \right] - \psi_8 \gamma_2 \frac{1}{40} = \frac{1}{6}g_2 - \frac{4}{15}g_4 + \frac{1}{10}g_6
\end{aligned}$$

$$\begin{aligned}
5 \leq n \leq N : & -\psi_{n-4} \gamma_2 \frac{1}{8(n-1)} + \psi_{n-2} \left[ \frac{(n-2)^2 - \gamma_1}{2(n-1)} - \gamma_2 \frac{1}{4(n^2-1)} \right] + \\
& \psi_n \left[ \frac{n(n^2-2+\gamma_1)}{n^2-1} + \gamma_2 \frac{n}{4(n^2-1)} \right] + \\
& \psi_{n+2} \left[ \frac{(n+2)^2 - \gamma_1}{2(n+1)} + \gamma_2 \frac{1}{4(n^2-1)} \right] - \\
& \psi_{n+4} \gamma_2 \frac{1}{8(n+1)} = \frac{1}{2(n-1)}g_{n-2} - \frac{n}{n^2-1}g_n + \frac{1}{2(n+1)}g_{n+2}.
\end{aligned}$$

## Appendix D. Behaviour of the velocities near the origin

A short remark may be made on the apparent singularities in the above equations (9a)–(9d) as we approach  $r = 0$ . Since we are dealing with physical quantities, representing the velocity field in a pipe, we know that the quantities involved are finite, and bounded near the axis. It can be shown, (Batchelor & Gill 1962; Loulou 1996), that near the axis the Fourier components behaves as

$$\begin{aligned} (\hat{u}_A, \hat{u}_B, \hat{w}) &\approx (c_1 r^{\alpha+1}, c_2 r^{\alpha-1}, c_3 r^\alpha), & \alpha = 1, 2, 3, \dots \\ (\hat{u}_A, \hat{u}_B, \hat{w}) &\approx (c_1 r^1, c_2 r^1, c_3 r^0), & \alpha = 0 \end{aligned}$$

and it is easy to verify that the equations (9c), (9d) are well-behaved as  $r \rightarrow 0$ .