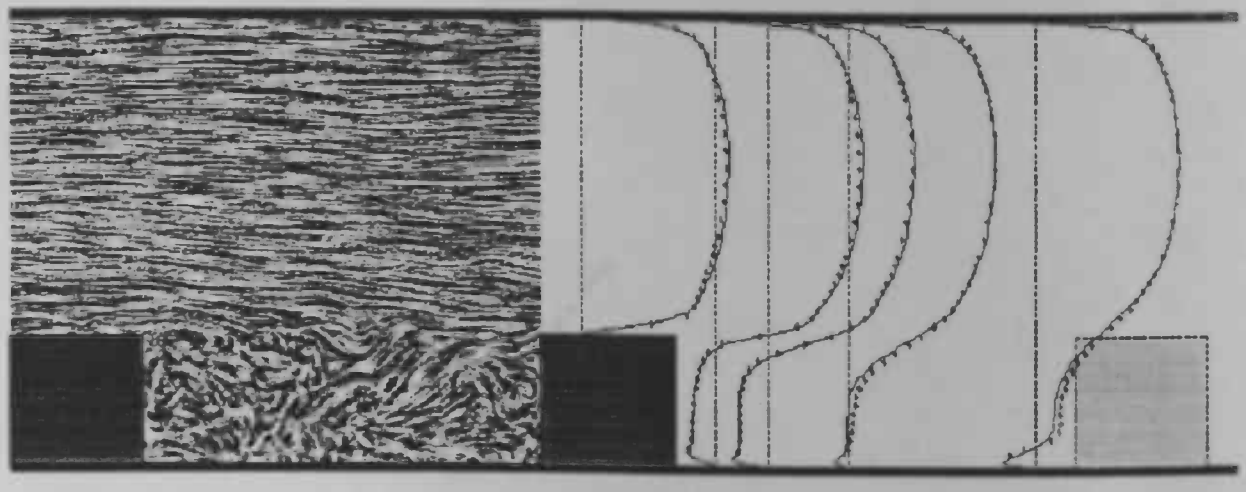
Turbulent flow and heat transfer in a channel with surface mounted cubical obstacles

Reanne van der Velde



Department of
Mathematics

RuG

Master's thesis

Turbulent flow and heat transfer in a channel with surface mounted cubical obstacles

Reanne van der Velde

University of Groningen
Department of Mathematics
P.O. Box 800
9700 AV Groningen

June 1999

This project has been carried out in cooperation with Gasunie Research.

Contents

Chapter 1

Introduction

In this report we consider the turbulent flow and heat transfer in a channel with an array of surface mounted cubical obstacles. The cubes are mounted in a regular pattern at one wall of the channel. Turbulent flow in a part of this channel is shown in Figure 1.1.
This problem has served as a test case at the last three ERCOFTAC/IAHR/COST Workshops on Refined Turbulence Modelling. See [1], [3] and [4]. From this series of workshops it may be concluded that this particular test case provides a major challenge to current turbulence models. At present, it poses a problem to which no turbulence model seems to have a satisfactory answer.

Both the flow and the heat transfer have experimentally been investigated by Meinders *et al.* [10], [11]. They have measured mean velocities and second-order moments of fluctuating velocities in the two planes that bisect the cubical obstacles. So far, the temperature has been measured at the surfaces of one heated cube only (by means of infrared thermography).

The turbulent flow and heat transfer in a channel with surface mounted cubical obstacles forms a generic example of a problem that occurs in many engineering applications, for instance in the design of cooling devices. We have performed a numerical simulation of it without using any turbulence models. This approach, called Direct Numerical Simulation (DNS), is the most accurate - but also the most expensive - way of computing complex turbulent flows with heat transfer, since all dynamically significant scales of motion are to be solved numerically from the unsteady, incompressible Navier-Stokes equations and the energy equation.
In view of the computational complexity, our first concern is to reduce the computational costs as far as we can get. This implies, among others, that the number of grid points has to be kept as small as possible. Lowering this number pays off. For instance, a reduction by a factor of two yields a saving of about one order of magnitude in both computing time and memory (in three spatial dimensions). To use the lowest possible number of grid points, spatial discretization methods for the Navier-Stokes equations need to be strained to their limit. On non-uniform grids various ways exist to discretize convective and diffusive operators. We propose to apply a 4th-order, finite-volume discretization method.
The objective of this study is two-fold. Firstly, we want to investigate how well our 4th-order discretization method performs in case of a complex turbulent flow with heat transfer. Thereto we will compare the results of the numerical simulation with the available experimental data. So far, only the experimental data of Meinders *et al* [10], [11] are available to

compare with. So, we also consider the turbulent flow and heat transfer in a channel with flat walls. For this channel a number of simulations have been performed by several research groups.
Secondly, we aim to add insight into the problem under consideration by discussing numerical results that have not been measured yet. For instance the temperature has only been measured at the surface of a cube, and not within the flow.

The report is organized as follows. The mathematical model is presented in Chapter 2. The numerical approach is concisely described in Chapter 3. Numerical results are discussed in Chapter 4.

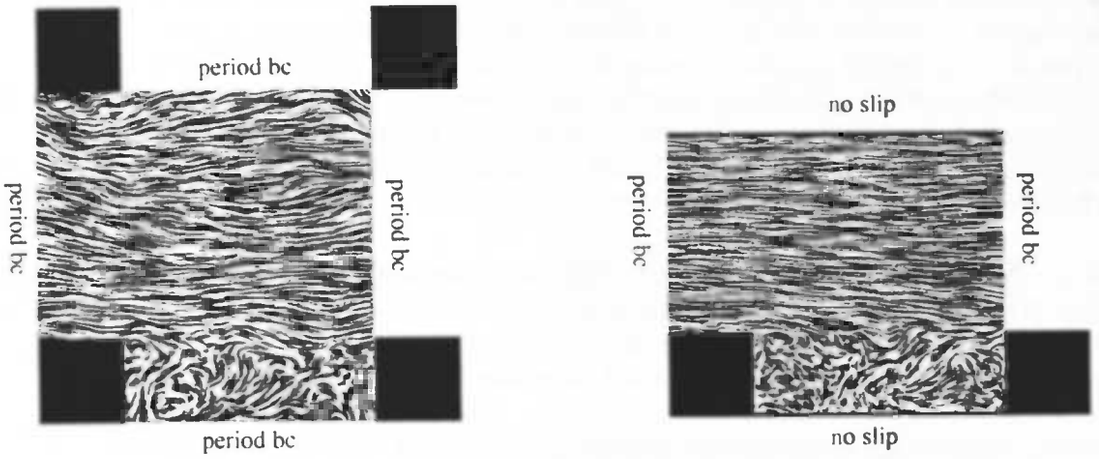


Figure 1.1: Top- and side-view of a sub-channel unit. Both pictures show an instantaneous flow field (taken from the DNS) at the plane that bisects the cubes.

Chapter 2

Mathematical Model

2.1 Navier-Stokes equations

The motion of the fluid is governed by conservation laws for mass, momentum and energy. We give these equations in conservation form for a Cartesian coordinate system. The partial derivative of a quantity ϕ with respect to the ith coordinate is denoted as $\partial_i \phi$. Furthermore the summation convection holds for the repeated indices.

- *conservation of mass*

$$\partial_t \rho + \partial_i(\rho u_i) = 0$$

- *conservation of momentum*

$$\partial_t(\rho u_i) + \partial_j(\rho u_i u_j) = \rho F_i + \partial_j \sigma_{ij}$$

  F_i denotes the ith component of an external force and $\sigma = (\sigma_{ij})$ is the stress tensor, which is given by

$$\sigma_{ij} = -p\delta_{ij} + \mu(\partial_j u_i + \partial_i u_j),$$

  with δ_{ij} the Kronecker symbol.

- *conservation of energy*

$$\partial_t(\rho E) + \partial_i(\rho E u_i) = \rho F_i u_i + \partial_j(u_i \sigma_{ij}) - \partial_i q_i$$

  $E = e + \frac{1}{2} u_i u_i$ is the total energy and q_i denotes the heat flux. According to Fourier's law, the heat flux is proportional to the temperature gradient

$$q_i = -\lambda \partial_i T.$$

The equations above need to be completed by a pair of equations of state, which describe the thermo-dynamics. For an ideal gas these are

$$p = \rho R T \text{ and } e = c_v T,$$

where $R = c_p - c_v$, with c_p and c_v the specific heats at constant pressure and constant volume, respectively. The specific heats are taken to be constant.
Then we can combine:

$$e = c_p T - \frac{p}{\rho}$$

We notice that $h = e + \frac{p}{\rho} = c_p T$ is the enthalpy.

Incompressible formulation

We consider air under conditions for which it can be assumed to be incompressible. Yet, we allow the density to depend on the temperature. In addition, we assume the viscosity μ to be constant. If we neglect all external forces, the continuity and momentum equations can be written as:

$$\text{div } \mathbf{u} = 0 \tag{2.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \text{grad})\ \mathbf{u} = -\frac{1}{\rho} \text{grad } p + \nu \text{ div grad } \mathbf{u} \tag{2.2}$$

Here is $\nu = \frac{\mu}{\rho}$ the kinematic viscosity. We can substitute $E = e + \frac{1}{2} u_i u_i$ in the energy equation, with e as explained above. Together with the relation for the stress tensor σ it is possible to rewrite the energy equation. After rearranging the terms and applying both the continuity equation (2.1) and the momentum equation (2.2) we get for a constant value of the density

$$\frac{\partial T}{\partial t} + \mathbf{u} \text{ grad } T = \frac{1}{\rho c_p} \text{ div } (\lambda \text{ grad } T) + \frac{1}{\rho c_p} \frac{\partial p}{\partial t} + \frac{1}{\rho c_p}\ \mathbf{u} \text{ grad } p + \frac{D}{\rho c_p},$$

where D is the viscous dissipation:

$$D = \mu \left[2 \left(\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2 \right) + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)^2 \right]$$

Both the pressure term and the viscous dissipation in the energy equation are neglected. We treat the coefficient λ as a constant. Under these assumptions the energy equation results into

$$\frac{\partial T}{\partial t} + \mathbf{u} \text{ grad } T = \frac{\nu}{Pr} \text{ div grad } T. \tag{2.3}$$

Here we have introduced the Prandtl number $Pr = \frac{\mu c_p}{\lambda}$.

Equations (2.1), (2.2) and (2.3) are rewritten in conservation form again, because in the numerical model the equations are discretized using conservation cells. If these changes are made and using div $\mathbf{u} = 0$ again we get:

$$\begin{aligned} \text{div } \mathbf{u} &= 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \text{ div}(\mathbf{u}\mathbf{u}^{\mathrm{T}}) &= -\frac{1}{\rho} \text{ div } (pI) + \nu \text{ div grad } \mathbf{u} \\ \frac{\partial T}{\partial t} + \text{ div}(\mathbf{u}\, T) &= \frac{\nu}{Pr} \text{ div grad } T \end{aligned} \tag{2.4}$$

I is the identity matrix and the divergence of a symmetric matrix is defined as:

$$\text{div} \begin{pmatrix} a_1 & a_2 & a_3 \\ a_2 & b_2 & b_3 \\ a_3 & b_3 & c_3 \end{pmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} \text{div}(a_1\ a_2\ a_3)^{\mathrm{T}} \\ \text{div}(a_2\ b_2\ b_3)^{\mathrm{T}} \\ \text{div}(a_3\ b_3\ c_3)^{\mathrm{T}} \end{pmatrix}$$

Equations (2.4) are made dimensionless by scaling them with a characteristic length L, a characteristic velocity U and a characteristic temperature T_c. This leads to:

$$\begin{aligned} \text{div } \mathbf{u} &= 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \text{div}(\mathbf{u}\mathbf{u}^{\mathrm{T}}) &= -\text{div}\,(pI) + \frac{1}{Re}\,\text{div grad } \mathbf{u} \\ \frac{\partial T}{\partial t} + \text{div}(\mathbf{u}\,T) &= \frac{1}{PrRe}\,\text{div grad } T \end{aligned} \tag{2.5}$$

This set of equations contains two parameters, the Prandtl number Pr and the Reynolds number Re. The Reynolds number measures the relative importance of the convective terms compared to the diffusive terms. The Reynolds number is defined as $Re = \dfrac{UL}{\nu}$.

Equations (2.5) are formulated as a conservation law, in which for any control volume Ω with boundary Γ applies:

$$\int_\Gamma \mathbf{u}\cdot\mathbf{n}\,d\Gamma = 0 \tag{2.6}$$

$$\int_\Omega \frac{\partial \mathbf{u}}{\partial t}\,d\Omega = -\int_\Gamma \mathbf{u}\,(\mathbf{u}\cdot\mathbf{n})\,d\Gamma - \int_\Gamma pI\cdot\mathbf{n}\,d\Gamma + \frac{1}{Re}\int_\Gamma \text{grad } \mathbf{u}\cdot\mathbf{n}\,d\Gamma \tag{2.7}$$

$$\int_\Omega \frac{\partial T}{\partial t}\,d\Omega = -\int_\Gamma \mathbf{u}T\cdot\mathbf{n}\,d\Gamma + \frac{1}{PrRe}\int_\Gamma \text{grad } T\cdot\mathbf{n}\,d\Gamma \tag{2.8}$$

The direction normal to the wall is denoted as $\mathbf{n}$. In the conservation law the solution is allowed to be less smooth than the solution of Equations (2.5). So, for turbulent flows, the integral formulation is prefered.

A channel with surface mounted cubical obstacles

Apart from boundary conditions, Equations (2.6), (2.7) and (2.8) are sufficient to describe the flow and heat transfer in the channel with flat walls. But in the channel with surface mounted cubical obstacles we need an additional equation which describes the heat transfer in the heated obstacle.

The cubical elements consist of an internal copper core covered by a thin epoxy layer. A schematic drawing of a composed element is shown in Figure 2.1. In the heated cube, the temperature decay across the copper core is negligible compared to that in the epoxy layer because of the high thermal conductivity of copper as compared to that of epoxy. This implies that we may take a uniform copper temperature. The temperature in the epoxy layer is obtained from the following version of the energy equation

$$\rho_e c_{p_e}\frac{\partial T_e}{\partial t} = \lambda_e\,\text{div grad } T_e, \tag{2.9}$$

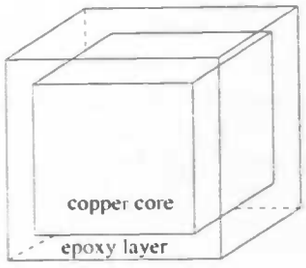


Figure 2.1: Schematic picture of a cubical obstacle

where λ_e is the thermal conductivity of the epoxy, ρ_e the density of the epoxy and c_{pe} the specific heat of the epoxy. These are all considered as constants. Equation (2.9) (the heat equation) is made dimensionless by scaling it with a characteristic length L, a characteristic velocity U and a characteristic temperature T_e. After that the heat equation becomes

$$\frac{\partial T_e}{\partial t} = c_{dif} \text{ div grad } T_e,$$

or for any volume Ω with boundary Γ

$$\int_\Omega \frac{\partial T_e}{\partial t} d\Omega = c_{dif} \int_\Gamma \text{ grad } T_e \cdot \text{n } d\Gamma, \tag{2.10}$$

with $c_{dif} = \dfrac{\lambda_e}{\rho_e c_{p_e}} \dfrac{1}{UL}$.

2.2 Boundary conditions

Equations (2.6), (2.7), (2.8) and (2.10) must be complemented by boundary conditions. Both test cases deal with a fully developed and symmetrical state, that is influences of the in- and outlet can be neglected. This justifies to confine the flow domain to a sub-channel unit with periodic boundary conditions in the streamwise and in the spanwise direction. At the solid walls of the flow domain the velocity must satisfy the no-slip conditions for a viscous fluid: $\mathbf{u} = 0$.
These guarantee that the normal component of the velocity is equal to zero (the fluid cannot flow through the wall) and that the tangential component of the velocity is equal to zero (the fluid sticks to the wall because of the viscosity).
The boundary conditions for the heat transfer differ for the two test cases.

A channel with flat walls

This test case deals with a fully developed and symmetrical thermal field. Therefore we have also applied periodic boundary conditions to the streamwise and spanwise direction for the temperature. The air temperature is prescribed at the solid channel walls: $T = T_0$ and $T = T_1$ at the lower and upper wall, respectively.

A channel with surface mounted cubical obstacles

The air temperature is prescribed at the inlet as well as at the channel walls and at the walls of the unheated cubes. Since just one cube in the array is heated we should not apply periodic boundary conditions at both the spanwise boundaries and the streamwise boundaries. Because of the high Reynolds number the air blows hard against the heated cube. So, the cube will hardly heat the air in spanwise direction. Therefore we have still put periodic boundary conditions in the spanwise direction. In the streamwise direction we have doubled the domain for the heat transfer since only one cube is heated. We have doubled the domain such that the heated cube is located at the middle of the domain. We have applied a Neumann condition at the outlet. This means that only convective heat transfer is possible at the outlet. This will not cause problems because the diffusive heat transfer is almost negligible compared to the convective heat transfer. Yet, a Neumann condition at the outlet will not fit entirely with the solution we look for. Therefore, non-physical waves may be reflected by the artificial outflow boundary. To suppress them, we have applied a buffer zone. This implies that we have enlarged the diffusion coefficient $\frac{1}{RePr}$. The Reynolds number influences the velocities also. So, we have adapted the Prandtl number. This is done by decreasing the Prandtl number linearly in a small area in front of the outlet.
To couple the air temperature with that of the heated cube we have computed the conduction of heat through the epoxy layer of the cube simultaneously with the convection and diffusion of heat in the air:

$$\underset{\text{(conduction)}}{-\lambda_e \frac{\partial T_e}{\partial n}} = \underset{\text{(convection)}}{-\lambda \frac{\partial T}{\partial n}} + \underset{\text{(radiation)}}{\epsilon\sigma \, (T_f^4 - T_{amb}^4)}. \tag{2.11}$$

At the wall of the cube the condition

$$T_e = T$$

applies, with T_e the epoxy temperature, T the air temperature and T_{amb} the ambient temperature. The radiation is modeled in terms of the average temperature on a face T_f. Hence, the radiation is constant per face. ϵ is the surface emissivity and σ the Stefan-Boltzmann constant. n denotes the outward unit normal (see Figure 2.2). Equation (2.11) is made dimensionless by scaling it with a characteristic length L and a characteristic temperature T_c

$$\frac{\partial T_e}{\partial n} = \frac{1}{c_{lab}} \frac{\partial T}{\partial n} - c_{rad}(T_f^4 - T_{amb}^4). \tag{2.12}$$

Here, $c_{lab} = \dfrac{\lambda_e}{\lambda}$ and $c_{rad} = \dfrac{\epsilon\sigma T_c^3 L}{\lambda_e}$.

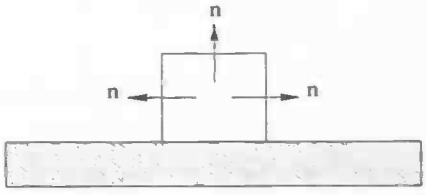


Figure 2.2: The normal n directs outwards

Equation (2.12) holds for the five faces of the cube that are cooled by the air flow. At the interface between the heated cubical obstacle and the channel wall a boundary condition for the temperature T_e is needed yet. The geometry of the interface between the heated cubical obstacle and the channel wall is rather complex in the experimental setting. It consists among others of screws, wires etc. We have disregarded all these elements. Instead, we have simply continued the epoxy layer near the interface. In addition, we have assumed that the temperature at the lower surface of the base plate equals the ambient temperature. Then we may obtain a boundary condition by performing a linear interpolation between the epoxy temperature in the cube nearest to the upper surface of the base plate and the ambient temperature at the lower surface of the base plate right under the cube (see Figure 2.3).

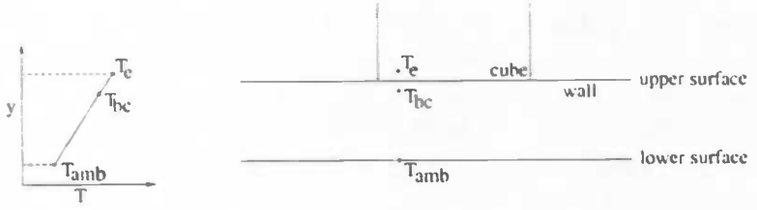


Figure 2.3: The boundary condition T_{bc} at the bottom of the cube is obtained by performing a linear interpolation between the epoxy temperature T_e in the cube and the ambient temperature T_{amb} at the lower surface of the base plate right under the cube.

With the boundary conditions prescribed in this section, Equations (2.6), (2.7), (2.8) and (2.10) are sufficient to describe the flow and heat transfer in both channels.

Chapter 3

Numerical Model

3.1 Spatial discretization

For the spatial discretization the sub-channel unit is covered by a staggered, Cartesian grid that can be stretched away from both the cubes and the channel walls. When a staggered grid is used the pressure and temperature are placed in the middle of the cell and the velocity normal to a cell face is placed in the middle of that cell face (see Figure 3.1). These cells are used in the air as well as in the cubical obstacles. In the air we must discretize Equations (2.6),

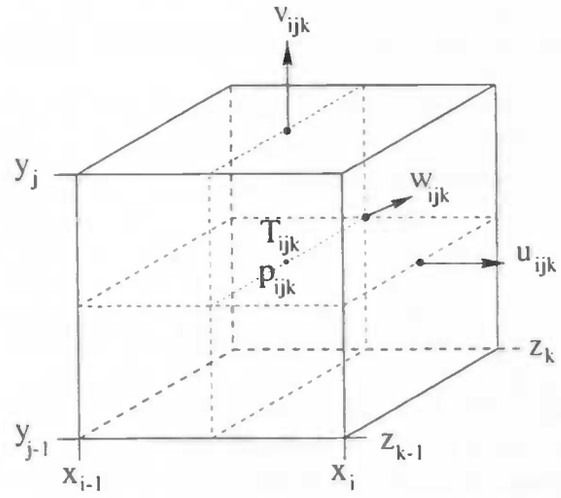


Figure 3.1: Placement of variables. Temperature and pressure are placed in the middle of the cell, the velocity normal to a cell face in the middle of that cell face.

(2.7) and (2.8). The discretization of Equations (2.6) and (2.7) has already been described in [12]. Here we will explain the discretization of Equation (2.8).
Define $\delta x_i = x_i - x_{i-1}$, $\delta y_j = y_j - y_{j-1}$ and $\delta z_k = z_k - z_{k-1}$, then Equation (2.8)

$$\int_\Omega \frac{\partial T}{\partial t}\, d\Omega \;=\; -\int_\Gamma \mathbf{u}T \cdot \mathbf{n}\, d\Gamma + \frac{1}{PrRe}\int_\Gamma \text{grad}\, T \cdot \mathbf{n}\, d\Gamma$$

is discretized as

$$\begin{aligned}
\frac{\partial T_{i,j,k}}{\partial t}\delta x_i\delta y_j\delta z_k &= (-uT|_e + \frac{1}{PrRe}\frac{\partial T}{\partial x}|_e)\delta y_j\delta z_k \\
&+ (-vT|_n + \frac{1}{PrRe}\frac{\partial T}{\partial y}|_n)\delta x_i\delta z_k \\
&+ (-wT|_d + \frac{1}{PrRe}\frac{\partial T}{\partial z}|_d)\delta x_i\delta y_j \\
&- (-uT|_w + \frac{1}{PrRe}\frac{\partial T}{\partial x}|_w)\delta y_j\delta z_k \\
&- (-vT|_s + \frac{1}{PrRe}\frac{\partial T}{\partial y}|_s)\delta x_i\delta z_k \\
&- (-wT|_u + \frac{1}{PrRe}\frac{\partial T}{\partial z}|_u)\delta x_i\delta y_j \ ,
\end{aligned}$$

where e,n,d,w,s and u denote the midpoint of the east, north, down, west, south and up face of the cell.

For the computation of $\mathbf{u}T$ on the surfaces of the cell the velocities u, v and w are available but the temperature is defined in the cell center. The temperature is averaged over the two neighbouring cells. For the computation of $\frac{\partial T}{\partial n}$ a central discretization is used. Then, we get:

$$\begin{aligned}
\frac{\partial T_{i,j,k}}{\partial t}\delta x_i\delta y_j\delta z_k &= \left(-\tfrac{1}{2}(T_{i+1,j,k}+T_{i,j,k})u_{i,j,k} + \frac{1}{PrRe}\frac{T_{i+1,j,k}-T_{i,j,k}}{\frac{1}{2}\delta x_{i+1}+\frac{1}{2}\delta x_i} \right)\delta y_j\delta z_k \\
&+ \left(-\tfrac{1}{2}(T_{i,j+1,k}+T_{i,j,k})v_{i,j,k} + \frac{1}{PrRe}\frac{T_{i,j+1,k}-T_{i,j,k}}{\frac{1}{2}\delta y_{j+1}+\frac{1}{2}\delta y_j} \right)\delta x_i\delta z_k \\
&+ \left(-\tfrac{1}{2}(T_{i,j,k+1}+T_{i,j,k})w_{i,j,k} + \frac{1}{PrRe}\frac{T_{i,j,k+1}-T_{i,j,k}}{\frac{1}{2}\delta z_{k+1}+\frac{1}{2}\delta z_k} \right)\delta x_i\delta y_j \qquad (3.1) \\
&- \left(-\tfrac{1}{2}(T_{i,j,k}+T_{i-1,j,k})u_{i-1,j,k} + \frac{1}{PrRe}\frac{T_{i,j,k}-T_{i-1,j,k}}{\frac{1}{2}\delta x_i+\frac{1}{2}\delta x_{i-1}} \right)\delta y_j\delta z_k \\
&- \left(-\tfrac{1}{2}(T_{i,j,k}+T_{i,j-1,k})v_{i,j-1,k} + \frac{1}{PrRe}\frac{T_{i,j,k}-T_{i,j-1,k}}{\frac{1}{2}\delta y_j+\frac{1}{2}\delta y_{j-1}} \right)\delta x_i\delta z_k \\
&- \left(-\tfrac{1}{2}(T_{i,j,k}+T_{i,j,k-1})w_{i,j,k-1} + \frac{1}{PrRe}\frac{T_{i,j,k}-T_{i,j,k-1}}{\frac{1}{2}\delta z_k+\frac{1}{2}\delta x_{k-1}} \right)\delta x_i\delta y_j
\end{aligned}$$

On non-uniform grids one would be tempted to tune the weights $\frac{1}{2}$ in the interpolation of T at a cell face to the actual mesh sizes, but we think that it is important that the flux through a surface of a control volume is computed independent of the control volume in which it is considered. This can only be achieved when the weights are taken independent of the grid location, and hence equal to the uniform weights. The advantage is that the coefficient matrix of the discrete convective operator is skew-symmetric in this manner.

On uniform grids, the local truncation error of the numerical integration of the energy equation over the control volume around $T_{i,j,k}$ is of the order of $(\delta x)^5, (\delta y)^5$ and $(\delta z)^5$. Per volume that is $(\delta x)^2, (\delta y)^2$ and $(\delta z)^2$. To eliminate the leading term of the truncation error, the energy equation is also integrated over a larger control volume around $T_{i,j,k}$ which is three times larger in every direction. Here, it may be noted that we can not blow up the 'original' volumes by a factor two (in all directions) since our grid is not collocated. On a staggered grid, three times larger volumes are the smallest ones possible for which the same discretization

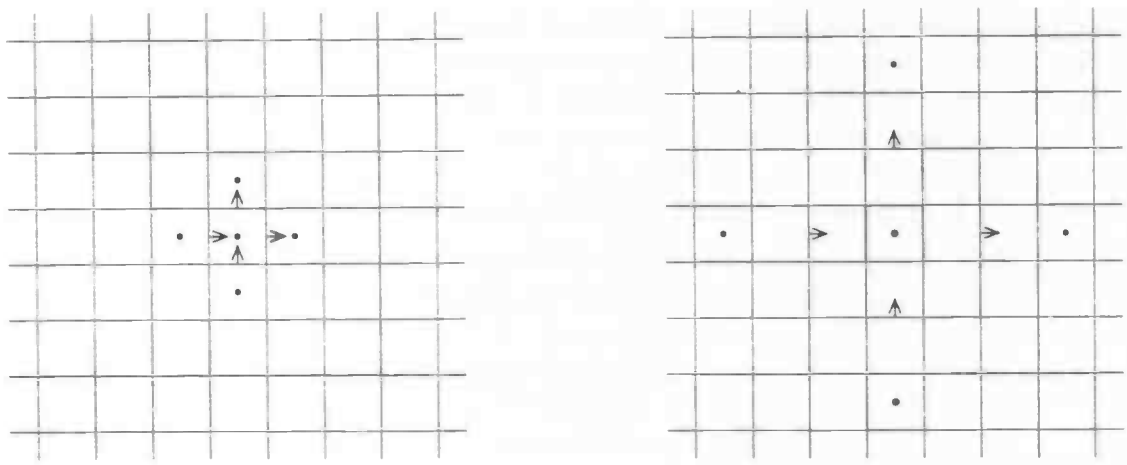


Figure 3.2: The left picture shows a control volume for the conservation of energy (in two spatial dimensions). The right picture shows a three-times larger control volume that is applied to eliminate the leading term of the truncation error. The arrows denote the components of the discrete velocity and the dots the components of the discrete temperature that are used to discretize the application of the conservation law of the control volume.

rules can be aplied as for the 'original' volumes. The three times larger cell consists of x, y, and z that satisfy (see Figure 3.2)

$$x_{i-2} \leq x \leq x_{i+1}, \quad y_{i-2} \leq y \leq y_{i+1} \quad \text{and} \quad z_{i-2} \leq z \leq z_{i+1}.$$

Since the same integration rule can be applied for both the original control volume and the three-times larger control volume, the leading term in the local truncation error of the latter is 3^5 times the leading term in the error in the former on a uniform grid. Hence, the leading term may be eliminated by taking:

$$\begin{aligned}
\frac{\partial T_{i,j,k}}{\partial t}\Big(\ 3^5\delta x_i\delta y_j\delta z_k - \delta X_i\delta Y_j\delta Z_k\ \Big) = \\
&+ 3^5 * \text{(right-hand side of Equation (3.1))} \\
&+\Big(\ -\tfrac{1}{2}(T_{i+3,j,k}+T_{i,j,k})u_{i+1,j,k} + \tfrac{1}{PrRe}\tfrac{T_{i+3,j,k}-T_{i,j,k}}{\frac{1}{2}\delta X_{i+1}+\frac{1}{2}\delta X_i}\ \Big)\delta Y_j\delta Z_k \\
&+\Big(\ -\tfrac{1}{2}(T_{i,j+3,k}+T_{i,j,k})v_{i,j+1,k} + \tfrac{1}{PrRe}\tfrac{T_{i,j+3,k}-T_{i,j,k}}{\frac{1}{2}\delta Y_{j+1}+\frac{1}{2}\delta Y_j}\ \Big)\delta X_i\delta Z_k \\
&+\Big(\ -\tfrac{1}{2}(T_{i,j,k+3}+T_{i,j,k})w_{i,j,k+1} + \tfrac{1}{PrRe}\tfrac{T_{i,j,k+3}-T_{i,j,k}}{\frac{1}{2}\delta Z_{k+1}+\frac{1}{2}\delta Z_k}\ \Big)\delta X_i\delta Y_j \\
&-\Big(\ -\tfrac{1}{2}(T_{i,j,k}+T_{i-3,j,k})u_{i-2,j,k} + \tfrac{1}{PrRe}\tfrac{T_{i,j,k}-T_{i-3,j,k}}{\frac{1}{2}\delta X_i+\frac{1}{2}\delta X_{i-1}}\ \Big)\delta Y_j\delta Z_k \\
&-\Big(\ -\tfrac{1}{2}(T_{i,j,k}+T_{i,j-3,k})v_{i,j-2,k} + \tfrac{1}{PrRe}\tfrac{T_{i,j,k}-T_{i,j-3,k}}{\frac{1}{2}\delta Y_j+\frac{1}{2}\delta Y_{j-1}}\ \Big)\delta X_i\delta Z_k \\
&-\Big(\ -\tfrac{1}{2}(T_{i,j,k}+T_{i,j,k-3})w_{i,j,k-2} + \tfrac{1}{PrRe}\tfrac{T_{i,j,k}-T_{i,j,k-3}}{\frac{1}{2}\delta Z_k+\frac{1}{2}\delta Z_{k-1}}\ \Big)\delta X_i\delta Y_j
\end{aligned}$$

Here, the volume of a three-times larger cell around $T_{i,j,k}$ is denoted by $\delta X_i\delta Y_j\delta Z_k$

Now, the leading term of the truncation error is of the order of $(\delta x_i)^7, (\delta y_j)^7$ and $(\delta z_k)^7$. Per volume the leading truncation error of $\frac{\partial T}{\partial t}$ will be of the order of $(\delta x_i)^4, (\delta y_j)^4$, and $(\delta z_k)^4$. It is clear that this holds on a uniform grid. Yet, it turns out that this method is also more accurate on a non-uniform grid [15], [17].

A channel with surface mounted cubical obstacles

In a channel with cubical obstacles we must discretize the temperature in the heated cube too. Inside the heated cube the temperature T_e is governed by the heat equation

$$\int_\Omega \frac{\partial T_e}{\partial t} d\Omega \;=\; c_{dif} \int_\Gamma \text{grad } T_e \cdot \mathbf{n}\, d\Gamma.$$

This equation is discretized as follows

$$\frac{\partial T_e}{\partial t}\delta x_i \delta y_j \delta z_k = c_{dif}\Big[\; \frac{\partial T_e}{\partial n}|_e \delta y_j \delta z_k + \frac{\partial T_e}{\partial n}|_n \delta x_i \delta z_k + \frac{\partial T_e}{\partial n}|_d \delta x_i \delta y_j \\ - \frac{\partial T_e}{\partial n}|_w \delta y_j \delta z_k - \frac{\partial T_e}{\partial n}|_s \delta x_i \delta z_k - \frac{\partial T_e}{\partial n}|_u \delta x_i \delta y_j \;\Big].$$

i.e. just like Equation (2.8) is discretized. As before e, n, d, w, s and u denote the east, north, down, west, south and up face of the cell. As in the discretization of Equation (2.8) we have used a central discretization for the approximation of $\frac{\partial T}{\partial n}$. Then we obtain an expression of the form:

$$\frac{\partial T_{e_{i,j,k}}}{\partial t}\delta x_i \delta y_j \delta z_k = c_{dif}\Big[\; c_{o_{i,j,k}} T_{e_{i+1,j,k}} + c_{n_{i,j,k}} T_{e_{i,j+1,k}} + c_{d_{i,j,k}} T_{e_{i,j,k+1}} + c_{w_{i,j,k}} T_{e_{i-1,j,k}} + \\ c_{s_{i,j,k}} T_{e_{i,j-1,k}} + c_{u_{i,j,k}} T_{e_{i,j,k-1}} - c_{diag_{i,j,k}} T_{e_{i,j,k}} \;\Big]$$

Here

$$c_{o_{i,j,k}} = \frac{\delta x_i \delta z_k}{\frac{1}{2}(\delta y_j + \delta y_{j+1})}\;,\quad c_{n_{i,j,k}} = \frac{\delta y_j \delta z_k}{\frac{1}{2}(\delta x_i + \delta x_{i+1})}\;,\quad c_{d_{i,j,k}} = \frac{\delta x_i \delta y_j}{\frac{1}{2}(\delta z_k + \delta z_{k+1})}\;,$$

$$c_{w_{i,j,k}} = \frac{\delta x_i \delta z_k}{\frac{1}{2}(\delta y_{j-1} + \delta y_j)}\;,\quad c_{s_{i,j,k}} = \frac{\delta y_j \delta z_k}{\frac{1}{2}(\delta x_{i-1} + \delta x_i)}\;,\quad c_{u_{i,j,k}} = \frac{\delta x_i \delta y_j}{\frac{1}{2}(\delta z_{k-1} + \delta z_k)}$$

and $c_{diag} = c_o + c_n + c_d + c_w + c_s + c_u$.
We only need to compute the temperature in the epoxy layer. This is a thin layer and it is represented by a few grid points. A few is five grid points. Therefore it is not useful to compute the epoxy temperature with the 4th-order discretization method.

3.2 Time discretization

In this section, we consider time-integration methods which are stable if the time step (at least) satisfies:

- $2\delta t < Re(\delta x)^2$ and the CFL-condition $\delta t < \frac{\delta x}{U_{max}}$ (Equation (2.7))
- $2\delta t < PrRe(\delta x)^2$ and the CFL-condition $\delta t < \frac{\delta x}{U_{max}}$ (Equation (2.8))
- $2\delta t < \frac{(\delta x)^2}{c_{dif}}$ (Equation (2.10))

Here, U_{max} denotes the maximum velocity. For the flows considered in this report the time step limitation is due to the CFL-condition. For example, for the flow in a channel with surface mounted cubical obstacles at $Re = 13,000$ and $Pr = 0.71$ ($\delta x = 6\ 10^-3, U_{max} = 3.86, c_{dif} = 6.13\ 10^{-7}$; see Section 4.2.1 and 4.2.2) the time step limitation $\delta t < \frac{\delta x}{U_{max}}$ is about two orders of magnitude stronger than $2\delta t < PrRe(\delta x)^2$ and about four orders of magnitude stronger than $2\delta t < \frac{(\delta x)^2}{c_{dif}}$.
The time-advancement of the convective and diffusive fluxes in the air flow is carried out by an explicit one-leg method that is tuned to get the largest possible interval of convective stability. This is explained in [16] for Equation (2.7). Here, we will explain the one-leg method applied to Equation (2.8). The pressure and the incompressibility constraint are treated implicitly in time. The resulting discrete Poisson equation for the pressure is solved iteratively (with the help of a modified incomplete Choleski Conjugate-Gradient method).
We denote the velocity and temperature at time $t = n\delta t$ by $\mathbf{u}^n$ and T^n, respectively. Starting from time $t = (n + \beta - \frac{1}{2})\delta t$, we integrate the energy equation over one time step δt using the midpoint rule

$$\frac{T^{n+\beta+\frac{1}{2}} - T^{n+\beta-\frac{1}{2}}}{\delta t} = \mathbf{f}(T^{n+\beta}, \mathbf{u}^{n+\beta}) \tag{3.2}$$

Here, the right-hand side $\mathbf{f}$ denotes the spatial discretization of convective and diffusive terms, as described in Section 3.1. Our aim is to determine β such that the corresponding method allows for the largest time step. We concluded that convective stability domains puts the most severe restriction on the time step. Thus, we look for stability domains which include eigenvalues $\lambda = x + iy$, where the real part is negative and the absolute value of the imaginary part y is much larger than the absolute value of the real part. Here, 'much' can range from one to two orders of magnitude. The real eigenvalues are of the form $\frac{\delta t}{2PrRe(\delta x)^2}$ and the imaginary eigenvalues of the form $\frac{\delta t U_{max}}{\delta x}$. Both the real part and the imaginary part of the eigenvalues depend linearly on δt. Hence, we can state our aim otherwise: We want to determine β such that the corresponding method possesses the largest region of convective stability.
The velocity and the temperature are defined on integer time levels only. We assume that the velocity and temperature are known up to and including level n. Then, the velocity $\mathbf{u}^{n+1}$ and the temperature T^{n+1} can be solved from Equation (3.2) if the velocities and temperatures at non-integer time levels in Equation (3.2) are approximated in terms of velocities and temperatures at integer levels. We approximate the off-step velocity $\mathbf{u}^{n+\beta}$ by a linear extrapolation of $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$ and denote the result by

$$\hat{\mathbf{u}} = (1 + \beta)\mathbf{u}^n - \beta\mathbf{u}^{n-1}.$$

The off-step temperature $T^{n+\beta}$ is approximated by

$$\hat{T} = (1 + \beta)T^n - \beta T^{n-1}.$$

The off-step temperature $T^{n+\beta+\frac{1}{2}}$ is approximated by a linear interpolation between T^{n+1} and T^n. Substituting these off-step approximations in Equation (3.2), we obtain the one-leg scheme

$$(\beta + \tfrac{1}{2})T^{n+1} - 2\beta T^n + (\beta - \tfrac{1}{2})T^{n-1} = \mathbf{f}(\hat{T}, \hat{\mathbf{u}}). \tag{3.3}$$

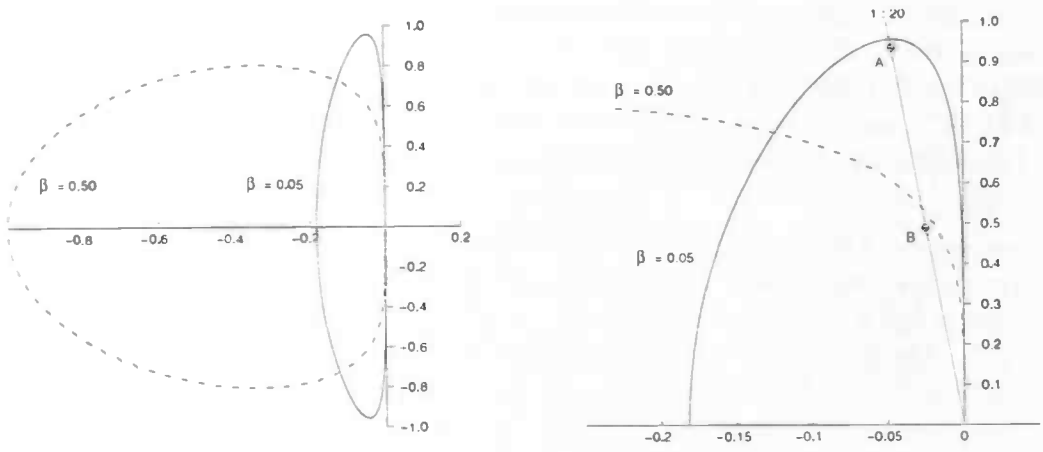


Figure 3.3: The left picture shows the stability domain of the one-leg method for $\beta = 0.05$ and $\beta = 0.5$. The horizontal axis corresponds with the real axis and the vertical axis with the imaginary axis. The right picture shows a blow up of the stability domains near the positive imaginary axis.

In Figure 3.3 the stability domain of the one-leg method is drawn for different values of β. We look for the one-leg method with the best linear stability properties. Figure 3.3 (left) shows the stability domain of the one-leg method for $\beta = 0.05$ and $\beta = 0.5$ (Adams-Bashforth). The stability domain is pressed against the imaginary axis when β goes to zero. In the limit $\beta = 0$ the stability domain is equal to the interval $[-i, i]$. Since only purely imaginary eigenvalues are allowed in this method, this method cannot be used to integrate a diffusive flux in time. We mentioned earlier that we look for a method that possesses the largest region of convective stability. That is, a method with a stability domain which includes eigenvalues with the absolute imaginary part about one to two orders of magnitude larger then the real part. Under these conditions, the one-leg method with $\beta = 0.05$ outperforms Adams-Bashforth. Figure 3.3 (right) shows a blow up of the stability domain of both methods near the positive imaginary axis. The points denoted by A and B lie on the line $|x| : |y| = 1 : 20$. The points A and B lie near to the boundary of the stability domains $\beta = 0.05$ and $\beta = 0.5$, respectively. A lies approximately two times as far form the origin as B. Thus, the time step of the one-leg method with $\beta = 0.05$ can be enlarged by a factor of two compared to Adams-Bashforth. For $|x| : |y| = 1 : 10$ this factor is about 1.5 and for $|x| : |y| = 1 : 100$ it is approximately 2.5. The one-leg method for $\beta = 0.05$ is not stable if an eigenvalue exists with imaginary part 0.99, i.e if $\frac{\delta t \delta x}{U_{max}} = 0.99$. As to be seen in Figure 3.3 the imaginary part may be 0.95 at most. Therefore we need to adapt the CFL-condition. The one-leg method with $\beta = 0.05$ is stable if the adapted CFL-condition holds: $\delta t < 0.95 \frac{\delta x}{U_{max}}$.

A channel with surface mounted cubical obstacles

In a channel with surface mounted cubical obstacles we have updated the temperature in the heated cube explicitly by means of Euler's method:

$$T^{n+1}_{e_{i,j,k}} = T^{n}_{e_{i,j,k}} + \delta t \, \frac{c_{dif}}{\delta x_i \delta y_j \delta z_k} \Big[\, c_{o_{i,j,k}} T^{n}_{e_{i,j+1,k}} + c_{w_{i,j,k}} T^{n}_{e_{i,j-1,k}} + c_{n_{i,j,k}} T^{n}_{e_{i+1,j,k}} + c_{s_{i,j,k}} T^{n}_{e_{i-1,j,k}}$$
$$+ c_{u_{i,j,k}} T^{n}_{e_{i,j,k+1}} + c_{d_{i,j,k}} T^{n}_{e_{i,j,k-1}} - c_{diag_{i,j,k}} T^{n}_{e_{i,j,k}} \, \Big]$$

3.3 Boundary conditions

A channel with flat walls

In both channels, we have taken the x-direction as the streamwise direction, the y-direction as the direction normal to the wall and the z-direction as the spanwise direction. The computational domain has $nx \times ny \times nz$ cells, where $i = 0$ and $i = nx$ correspond with the in- and outlet, $j = 0$ and $j = ny$ correspond with the solid walls and $k = 0$ and $k = nz$ correspond with the spanwise boundaries. For the 4th-order discretization method three ghost velocities and temperatures are needed in every direction. We thereby made use of mirror points in the solid walls. The mean of the value at the point in the domain and the value at its mirror point is equal to the prescribed value at the wall. The discretized boundary conditions are summarized below.

location	variable			
	u	v	w	T
$(0, j, k)$	$u_{nx,j,k}$	$v_{nx,j,k}$	$w_{nx,j,k}$	$T_{nx,j,k}$
$(-1, j, k)$	$u_{nx-1,j,k}$	$v_{nx-1,j,k}$	$w_{nx-1,j,k}$	$T_{nx-1,j,k}$
$(-2, j, k)$	$u_{nx-2,j,k}$	$v_{nx-2,j,k}$	$w_{nx-2,j,k}$	$T_{nx-2,j,k}$
$(nx+1, j, k)$	$u_{1,j,k}$	$v_{1,j,k}$	$w_{1,j,k}$	$T_{1,j,k}$
$(nx+2, j, k)$	$u_{2,j,k}$	$v_{2,j,k}$	$w_{2,j,k}$	$T_{2,j,k}$
$(nx+3, j, k)$	$u_{3,j,k}$	$v_{3,j,k}$	$w_{3,j,k}$	$T_{3,j,k}$

Table 3.1: Ghost velocities and temperatures in the x-direction. In this direction periodic conditions are applied.

location	variable			
	u	v	w	T
$(i, 0, k)$	$-u_{i,1,k}$	0	$-w_{i,1,k}$	$2T_0 - T_{i,1,k}$
$(i, -1, k)$	$-u_{i,2,k}$	$-v_{i,1,k}$	$-w_{i,2,k}$	$2T_0 - T_{i,2,k}$
$(i, -2, k)$	$-u_{i,3,k}$	$-v_{i,2,k}$	$-w_{i,3,k}$	$2T_0 - T_{i,3,k}$
(i, ny, k)		0		
$(i, ny+1, k)$	$-u_{i,ny,k}$	$-v_{i,ny-1,k}$	$-w_{i,ny,k}$	$2T_1 - T_{i,ny,k}$
$(i, ny+2, k)$	$-u_{i,ny-1,k}$	$-v_{i,ny-2,k}$	$-w_{i,ny-1,k}$	$2T_1 - T_{i,ny-1,k}$
$(i, ny+3, k)$	$-u_{i,ny-2,k}$		$-w_{i,ny-2,k}$	$2T_1 - T_{i,ny-2,k}$

Table 3.2: Ghost velocities and temperatures in the y-direction. No-slip conditions are imposed at the boundaries $j = 0$ and $j = ny$.

The discretization of the periodic boundary conditions in the z-direction proceeds similar to the discretization of the periodic boundary conditions in the x-direction.

A channel with surface mounted cubical obstacles

We have confined the flow domain to a sub-channel unit of $nx \times ny \times nz$ cells. For the heat transfer we have doubled the domain in the streamwise direction, since only one cube in the

array is heated. The discretization of the Dirichlet conditions and the periodic boundary conditions is equal to the discretization in the flat walled channel. The 'inlet' from the flow is located at $i = 0$. For the temperature, the inlet is located at $i = -nx$ and the outlet at $i = nx$. Near the in- and outlet and close to the heated cube we have applied the 2nd-order discretization method to the energy equation. We have applied the 2nd-order method at the inlet because we have prescribed the temperature there. In turbulent flows the temperature is everything but constant and therefore it is not necessary to compute the temperature accurately at the inlet. At the outlet we have prescribed a Neumann condition. This condition does not describe the physics in all details. Thus, at the outlet it is also not necessary to apply the 4th-order method. In the cube we have applied the 2nd-order method to compute T_e. The air temperature and the epoxy temperature are coupled with each other, therefore we have applied the 2nd-order method to compute the air temperature at the cube. Applying the 2nd-order method means that we need only one ghost point at the in- and outlet and at the faces of the heated cube. At the in- and outlet these are given by:

- $T_{-nx,j,k} = 2T_{in} - T_{1-nx,j,k}$ Dirichlet condition at the inlet
- $T_{-nx,j,k} = 2T_{cube} - T_{1-nx,j,k}$ Dirichlet condition at the face of the unheated cube at the inlet.
- $T_{nx,j,k} = T_{1-nx,j,k}$ Neumann condition at the outlet
- $T_{nx,j,k} = 2T_{cube} - T_{1-nx,j,k}$ Dirichlet condition at the face of the unheated cube at the outlet.

It will be clear that T_{in} is the prescribed temperature at the inlet and that T_{cube} is the prescribed temperature of the unheated cubes.
In Chapter 2 we have described two equations for a channel with cubical mounted obstacles from which the boundary conditions are obtained for both T and T_e at the faces of the heated cube. At the five faces of the heated cube that are cooled by the air flow these two equations were (at each time step):

$$\begin{aligned} \frac{\partial T_e}{\partial n} &= \frac{1}{c_{lab}} \frac{\partial T}{\partial n} - c_{rad}(T_f^4 - T_{amb}^4) \\ T_e &= T \end{aligned}$$

The radiation is computed by means of the average temperature (T_f) on a face; hence, it is constant per face. So, we have two equations and two unknown quantities T_e and T. For the approximation of $\frac{\partial T}{\partial n}$ and $\frac{\partial T_e}{\partial n}$ a central discretization is used, just like we did in the preceding discretizations of $\frac{\partial T}{\partial n}$. The temperatures in Equation (3.5) are averaged over the two neighbouring cells. Then we get, for example at the downstream face of the heated cube ($i = i_1$):

$$\frac{T_{e_{i1+1,j,k}} - T_{e_{i1,j,k}}}{\frac{1}{2}\delta x_{i1} + \frac{1}{2}\delta x_{i1+1}} = \frac{1}{c_{lab}} \frac{T_{i1+1,j,k} - T_{i1,j,k}}{\frac{1}{2}\delta x_{i1} + \frac{1}{2}\delta x_{i1+1}} - c_{rad}\left(T_f^4 - T_{amb}^4 \right) \tag{3.4}$$

$$\frac{1}{2}\left(T_{e_{i1+1,j,k}} + T_{e_{i1,j,k}} \right) = \frac{1}{2}\left(T_{i1+1,j,k} + T_{i1,j,k} \right) \tag{3.5}$$

$T_{i1,j,k}$ and $T_{e_{i1,j,k}}$ are located in the cube, so $T_{i1,j,k}$ is a ghost temperature. $T_{i1+1,j,k}$ and $T_{e_{i1+1,j,k}}$ are located in the air. Therefore, $T_{e_{i1+1,j,k}}$ is a ghost temperature. See also Figure 3.4. The ghost temperature $T_{e_{i1+1,j,k}}$ is solved from Equation (3.4). If we apply Equation (3.5) to

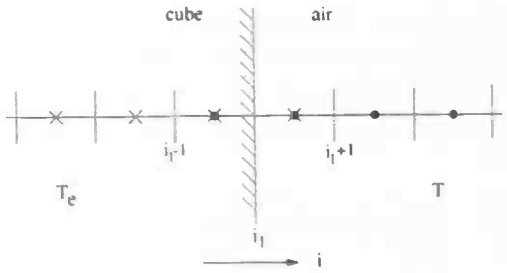


Figure 3.4: The coupling of the air temperature and the epoxy temperature. The air temperature is depicted by dots and the epoxy temperature corresponds to the crosses.

eliminate $T_{i1,j,k}$ this leads to:

$$\begin{aligned} T_{e_{i1+1,j,k}} &= T_{e_{i1,j,k}} + \frac{1}{c_{lab}}\left(T_{i1+1,j,k} - (T_{e_{i1+1,j,k}} + T_{e_{i1,j,k}} - T_{i1+1,j,k})\right) \\ &\quad -c_{rad}\ \tfrac{1}{2}(\delta x_{i1+1} + \delta x_{i1})\left(T_f^4 - T_{amb}^4\right) \end{aligned}$$

After rearranging terms we get:

$$T_{e_{i1+1,j,k}} = \frac{c_{lab}-1}{c_{lab}+1}T_{e_{i1,j,k}} + \frac{2}{c_{lab}+1}T_{i1+1,j,k} - \frac{c_{rad}\ c_{lab}}{c_{lab}+1}\ \tfrac{1}{2}(\delta x_{i1+1} + \delta x_{i1})\left(T_f^4 - T_{amb}^4\right) \quad (3.6)$$

Now it remains to solve $T_{i1,j,k}$ from Equation (3.5):

$$T_{i1,j,k} = T_{e_{i1,j,k}} + T_{e_{i1+1,j,k}} - T_{i1+1,j,k} \quad (3.7)$$

We will prove that the above discretizations are stable *i.e.* that the discretization error does not increase. Suppose that the error of $T_{e_{i,j,k}}$ on time $t = n\delta t$ is $\delta^n_{i,j,k}$ and suppose that the error of $T_{i,j,k}$ on time $t = n\delta t$ is $\epsilon^n_{i,j,k}$. Then the discretization error on time $t = n\delta t$ of Equation (3.6) becomes

$$\left|\delta^n_{i1+1,j,k}\right| = \left|\frac{c_{lab}-1}{c_{lab}+1}\right|\left|\delta^n_{i1,j,k}\right| + \left|\frac{2}{c_{lab}+1}\right|\left|\epsilon^n_{i1+1,j,k}\right|.$$

Note that $\frac{c_{rad}\ c_{lab}}{c_{lab}+1}\ \frac{1}{2}(\delta x_{i1+1} + \delta x_{i1})\left(T_f^4 - T_{amb}^4\right)$ is constant, thus it causes neither an increase nor a decrease in the error of the ghost temperature T_e. The coefficient c_{lab} equals approximately 9 (see Section 4.2.2). Therefore $\frac{c_{lab}-1}{c_{lab}+1} > 0$ and $\frac{2}{c_{lab}+1} > 0$ and thus:

$$\begin{aligned} \left|\delta^n_{i1+1,j,k}\right| &\leq \left(\frac{c_{lab}-1}{c_{lab}+1} + \frac{2}{c_{lab}+1}\right)\ \max\left\{\left|\delta^n_{i1,j,k}\right|, \left|\epsilon^n_{i1+1,j,k}\right|\right\} \\ &\leq 1\ \max\left\{\left|\delta^n_{i1,j,k}\right|, \left|\epsilon^n_{i1+1,j,k}\right|\right\} \end{aligned}$$

Hence, the discretization in Equation (3.6) is stable, but what about Equation (3.7). Well, we have

$$\begin{aligned} \left|\epsilon^n_{i1,j,k}\right| &= \left|\delta^n_{i1,j,k} + \delta^n_{i1,j,k} + \epsilon^n_{i1+1,j,k}\right| \quad (3.8) \\ &\leq 3\ \max\left\{\left|\delta^n_{i1,j,k}\right|, \left|\delta^n_{i1+1,j,k}\right|, \left|\epsilon^n_{i1+1,j,k}\right|\right\}, \end{aligned}$$

thus the discretization error $\epsilon^n_{i1,j,k}$ may blow up with a factor three. Ghost temperatures in the cube are computed to obtain the convective and diffusive flux close by the cube. In the first section of this chapter we have discussed how we discretize these fluxes. We have used the 2nd-order method in the neighbourhood of the cube. In one dimension, Equation (3.1) can then be written as:

$$\begin{aligned}\frac{\partial T_{i1+1}}{\partial t}\delta x_{i1+1} = & - \frac{1}{2}(T_{i1+2}+T_{i1+1})u_{i1+1} + \frac{1}{PrRe}\frac{T_{i1+2}-T_{i1+1}}{\frac{1}{2}\delta x_{i1+2}+\frac{1}{2}\delta x_{i1+1}} \\ & + \frac{1}{2}(T_{i1+1}+T_{i1})u_{i1} - \frac{1}{PrRe}\frac{T_{i1+1}-T_{i1}}{\frac{1}{2}\delta x_{i1+1}+\frac{1}{2}\delta x_{i1}} \end{aligned} \tag{3.9}$$

T_{i1} will not blow up the error in the convective flux because $u_{i1} = 0$. After rearranging terms in Equation (3.9) the diffusive flux becomes

$$\frac{1}{PrRe}\frac{T_{i1+2}-2T_{i1+1}+T_{i1}}{\frac{1}{2}\delta x_{i1+2}+\delta x_{i1+1}+\frac{1}{2}\delta x_{i1}}.$$

In this flux we have a constant denominator. Therefore, we may restrict ourselves to the discretization error ϵ in the numerator:

$$|\epsilon| = |\epsilon_{i1+2} - \epsilon_{i1+1} + \epsilon_{i1}| \tag{3.10}$$

If we apply Relation (3.8) to ϵ_{i1+1}, Equation (3.10) results into:

$$|\epsilon| = |\epsilon_{i1+2} - 3\epsilon_{i1+1} + \delta_{i1} + \delta_{i1+1}|$$

Here $i1+1$ denotes the central term. Since its absolute coefficient ($|-3|$) is equal to the sum of the three other coefficients, we can keep the time-integration stable by choosing the time-step small enough ($\delta t < 0.95\frac{\delta x}{U_{max}}$, see Section 3.2). Then the discretization errors of the ghost temperatures do not blow up the discretization errors of the diffusive and convective fluxes and thus they do not increase the error of the air temperatures. Hence, the coupling (3.6) and (3.7) is stable.

At the interface between the epoxy layer and the base plate we have obtained a boundary condition for T_e by performing a linear interpolation between the epoxy temperature in the cube nearest to the wall and the ambient temperature T_{amb} at the lower surface of the base plate right under the cube. If we define the thickness of the wall as d, we get for the boundary condition T_{e_0} (see Figure 3.5):

$$T_{e_{0,j,k}} = \frac{\left(x_{-\frac{1}{2}}+d\right)T_{e_{1,j,k}} + \left(x_{\frac{1}{2}}-x_{-\frac{1}{2}}\right)T_{in}}{x_{\frac{1}{2}}+d}$$

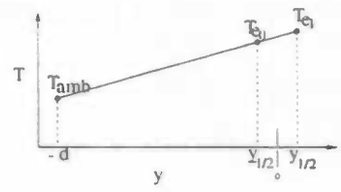


Figure 3.5: By performing a linear interpolation between T_{e_1} and T_{amb} we obtain T_{e_0}. Here $y = -d$ and $y = 0$ correspond to the lower surface and the upper-surface of the base plate, respectively.

Chapter 4

Results

In this chapter two cases will be considered.

1. A channel with flat walls
2. A channel with surface mounted cubical obstacles

The second case uses an extended computer program of the first case. So, the first case is considered to validate our simulation code.

4.1 A channel with flat walls

Before we discuss the results of this test case, we will demonstrate that the added numerical equations are implemented well.

4.1.1 Program verification

For this test case we added the following items to the existing DNS computer program:

- spatial discretization of the temperature
  - discretization of the convective part
  - discretization of the diffusive part
- time discretization of the temperature
- discretization of the boundary conditions of the temperature

By taking $\mathbf{u} = 0$, we have tested the discretization of the diffusive part. Then, the time discretization is also tested. With $\mathbf{u} = 0$ the energy equation can be written as

$$\frac{\partial T}{\partial t} = \frac{1}{PrRe}\left(T_{xx} + T_{yy} + T_{zz}\right) + ft(x, y, z). \tag{4.1}$$

Here, ft is a source term, which can be added to the equation without problems. For some exact functions which satisfy the equation we have determined the discretization error. In selecting these exact solutions we have taken the various types of boundary conditions into

account so that the discretized boundary conditions were also tested. We have confined the flow domain to a sub-channel unit of dimension $1 \times 1 \times 1$.
The test functions are:

1. periodic boundary conditions:
   $T_\alpha(y,t) = \cos(4\pi y)\ e^{ct}, \quad c = \frac{-(4\pi)^2}{PrRe}$
   $ft = 0$

2. $y = 0 : T = T_0 \quad , \quad y = 1 : \frac{\partial T}{\partial y} = 0 :$
   $T_\beta(y,t) = \sin(\frac{5}{2}\pi y)\ e^{ct} + T_0, \quad c = \frac{-(\frac{5}{2}\pi)^2}{PrRe}$
   $ft = 0$

3. $y = 0 : \frac{\partial T}{\partial y} = 0 \quad , \quad y = 1 : T = T_1 :$
   $T_\gamma(y,t) = \cos(\frac{5}{2}\pi y)\ e^{ct} + T_1 y^2, \quad c = \frac{(\frac{5}{2}\pi)^2}{PrRe}$
   $ft = \frac{-2T_1}{PrRe}$

We want the parameter $c \approx 1$. Therefore we have set the Prandtl number to $Pr = 1$ and we have varied the Reynolds number. We have choosen a stable time step δt and we let the program run for $n\ \delta t = 1$ second. We have tested both the 2nd-order and the 4th-order discretization method on uniform grids. Then the disretization error at a prescribed point should depend linearly on δy^2 and δy^4, respectively. The error of the time-integration method is δt^2. Therefore the error in the time-integration may be larger than the error in the discretization of the diffusive part in the 4th-order method. The linear dependence on δy^4 may then be disturbed. We note that the discretization error also depends on the derivative of the exact function T (Taylor series). Thus, with different functions we get different errors. The errors are summarized in Tables 3.1, 3.2 and 3.3. The error plots are shown in Figure 4.1. We got exactly the same discretization errors when T_α, T_β and T_γ depended on x or z. We may conclude that the discretization of the diffusive part is well implemented and that the error in the time-integration method does not dominate the error in the discretization of the diffusive part.
Now it remains for us to test the discretization of the convective part. The test functions must comply with the next version of the energy equation

$$\frac{\partial T}{\partial t} + \text{div}(\text{u}\ T) \quad = \quad \frac{1}{\text{Pr}Re}\ \text{div grad}\ T + ft$$

If the results of the simulation of the channel with flat walls agree well with reference data, we may conclude that the discretization of the convective part is also well implemented. We will discuss the results in the next section.

number of cells	Discretization error		j coordinate of the error $T_{i,j,k} - T_\alpha(\frac{1}{2}, 1)$
	2nd-order	4th-order	
16 x 15 x 16	$1.19463\ 10^{-2}$	$2.25964\ 10^{-3}$	8
16 x 19 x 16	$7.48985\ 10^{-3}$	$9.18813\ 10^{-4}$	10
16 x 25 x 16	$4.34392\ 10^{-3}$	$3.16608\ 10^{-4}$	13
16 x 35 x 16	$2.22236\ 10^{-3}$	$8.42381\ 10^{-5}$	18
16 x 45 x 16	$1.34590\ 10^{-3}$	$3.11087\ 10^{-5}$	23
16 x 55 x 16	$9.01488\ 10^{-4}$	$1.40063\ 10^{-5}$	28
16 x 75 x 16	$4.85059\ 10^{-4}$	$4.07022\ 10^{-6}$	38

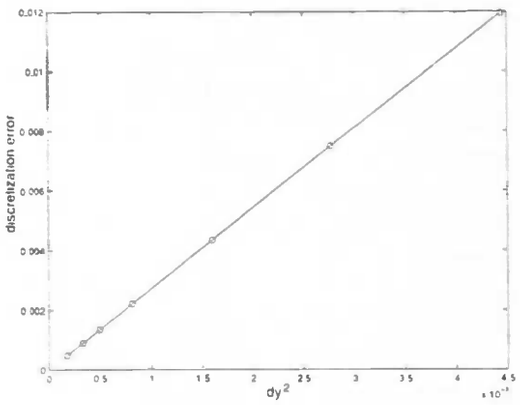
Table 4.1: Discretization errors $|T - T_\alpha|$ at different grids, where $T_\alpha = \cos(4\pi y)e^{ct}$.

number of cells	Discretization error		j coordinate of the error $T_{i,j,k} - T_\beta(\frac{1}{2}, 1)$
	2nd-order	4th-order	
16 x 15 x 16	$5.95627\ 10^{-3}$	$4.65807\ 10^{-4}$	8
16 x 19 x 16	$3.70890\ 10^{-3}$	$1.84198\ 10^{-4}$	10
16 x 25 x 16	$2.14084\ 10^{-3}$	$6.22335\ 10^{-5}$	13
16 x 35 x 16	$1.09177\ 10^{-3}$	$1.63401\ 10^{-5}$	18
16 x 45 x 16	$6.60331\ 10^{-4}$	$6.00215\ 10^{-6}$	23
16 x 55 x 16	$4.41998\ 10^{-4}$	$2.69569\ 10^{-6}$	28
16 x 75 x 16	$2.37677\ 10^{-4}$	$7.82413\ 10^{-7}$	38

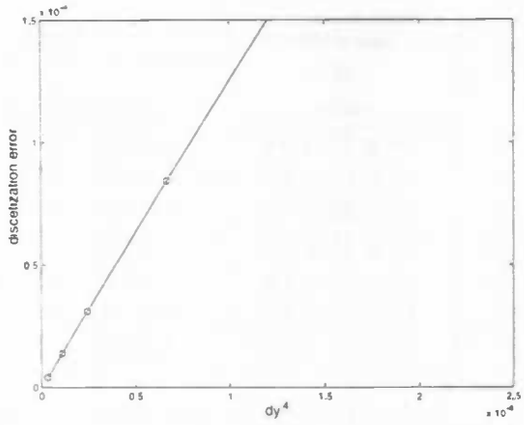
Table 4.2: Discretization errors $|T - T_\beta|$ at different grids, where $T_\beta = \sin(\frac{5}{2}\pi y)\ e^{ct} + T_0$.

number of cells	discretization error		j coordinate of the error $T_{i,j,k} - T_\gamma(\frac{1}{2}, 1)$
	2nd-order	4th- order	
16 x 15 x 16	$5.99580\ 10^{-3}$	$4.69745\ 10^{-4}$	8
16 x 19 x 16	$3.73235\ 10^{-3}$	$1.88502\ 10^{-4}$	10
16 x 25 x 16	$2.15389\ 10^{-3}$	$6.54287\ 10^{-5}$	13
16 x 35 x 16	$1.09825\ 10^{-3}$	$1.82039\ 10^{-5}$	18
16 x 45 x 16	$6.64209\ 10^{-4}$	$7.18647\ 10^{-6}$	23
16 x 55 x 16	$4.44579\ 10^{-4}$	$3.50760\ 10^{-6}$	28
16 x 75 x 16	$2.39057\ 10^{-4}$	$1.22859\ 10^{-6}$	38

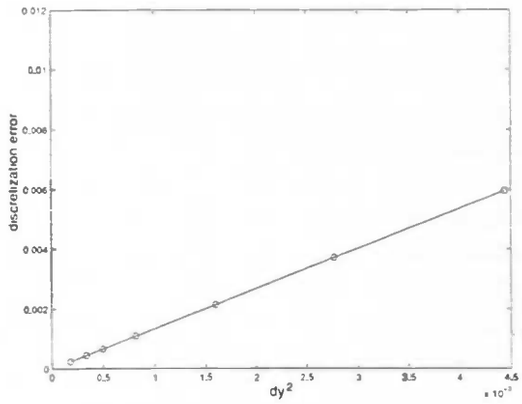
Table 4.3: Discretization errors $|T - T_\gamma|$ at different grids, where $T_\gamma = \cos(\frac{5}{2}\pi y)\ e^{ct} + T_1 y^2$.



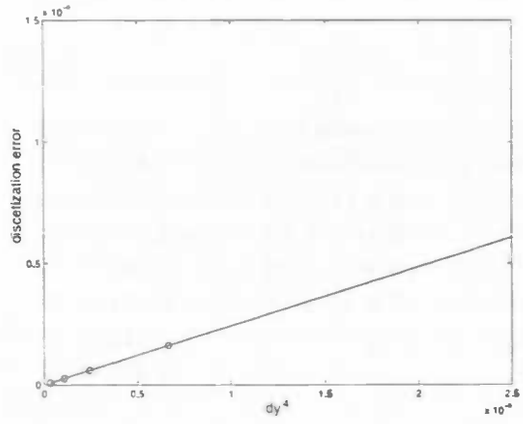
function 1, 2nd-order method.



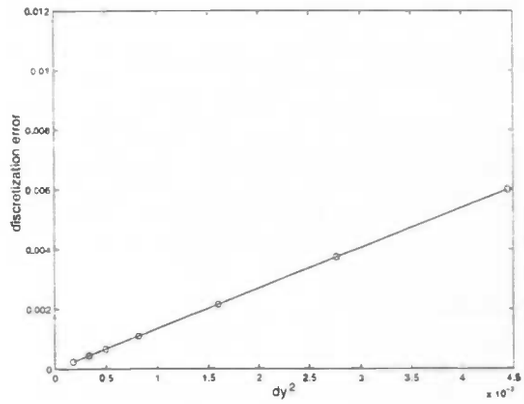
function 1, 4th-order method.



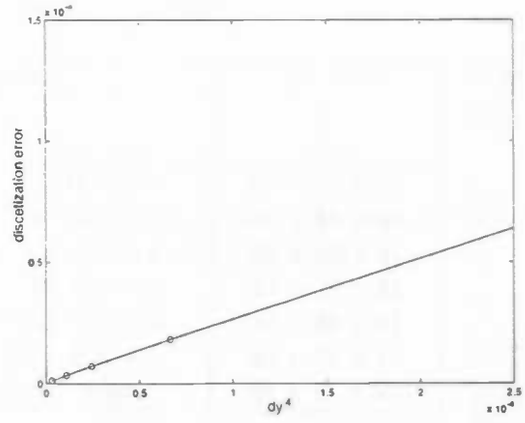
function 2, 2nd-order method.



function 2, 4th-order method.



function 3, 2nd-order method.



function 3, 4th-order method.

Figure 4.1: Discretization errors as functions of δy^2 (2nd-order) and δy^4 (4th-order).

4.1.2 Simulation

To validate our simulation code we have computed the velocity and the temperature of a fully developed flow in a channel with flat walls first. We have confined the flow domain to a sub-channel unit of dimension $2\pi \times 1 \times \pi$. The height of the channel is 1. The sub-channel unit is covered by a $64 \times 64 \times 32$ staggered grid that is stretched away from both the channel walls. The first grid point away from a wall is located at $4 \cdot 10^{-3}$. At both channel walls we have put the temperature equal to $T_0 = T_1 = 0$. The Reynolds number is equal to $Re = 5,600$ (based on the channel width and the bulk velocity) and the Prandtl number is $Pr = 0.71$. At this Reynolds and Prandtl number a number of simulations have been performed by a several research groups; see e.g. [2], [7], [8] and [9].
We have distinguished two areas in the turbulent boundary layer:

- An inner layer where turbulent mixture is the dominant phenomenon.
- A laminar sublayer close to the channel wall where turbulent friction is negligible as compared to the skin-friction at the channel wall

In Figures 4.2 and 4.3 mean velocity and temperature profiles are depicted as functions of the logarithmic distance to the wall, so that the laminair sublayer is shown much larger than it is in reality. The profiles are made dimensionless with the friction velocity and friction temperature, respectively

$$u_\tau \equiv \sqrt{\frac{1}{Re}\frac{\partial u}{\partial y}} \quad \text{and} \quad T_\tau \equiv \frac{1}{u_\tau Pr Re}\frac{\partial T}{\partial y}.$$

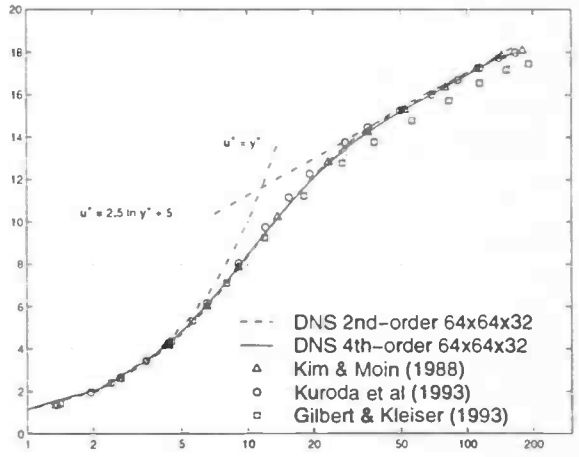


Figure 4.2: Comparison of the mean streamwise velocity u^+ as function of y^+ in a channel with flat walls. The dashed lines represent the law of the wall and the log law. The markers represent DNS-results that are taken from both the ERCOFTAC Database and the Japanese DNS Data Base of Turbulent Transport Phenomena.

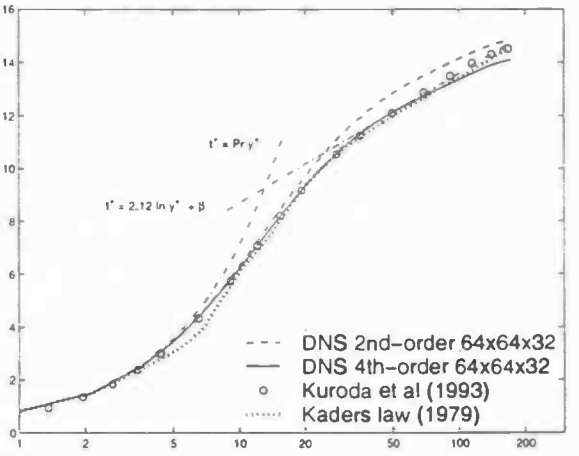


Figure 4.3: Comparison of the mean temperature T^+ in a plane channel. The dashed lines represent the law of the wall and the log law. Here, it may be noted that Kaders data provides the best fit through a large number of experiments (performed at a large range of Reynolds numbers).

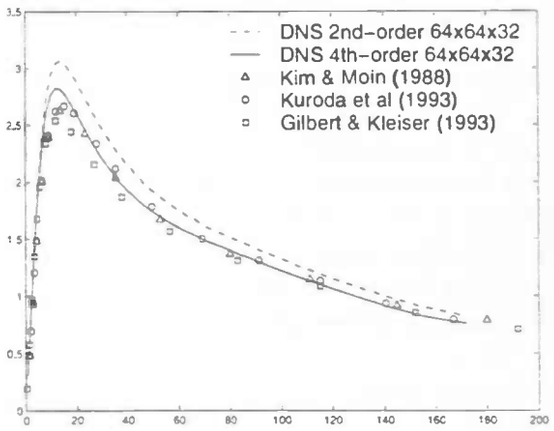


Figure 4.4: Comparison of u_{rms} in a fully developed channel.

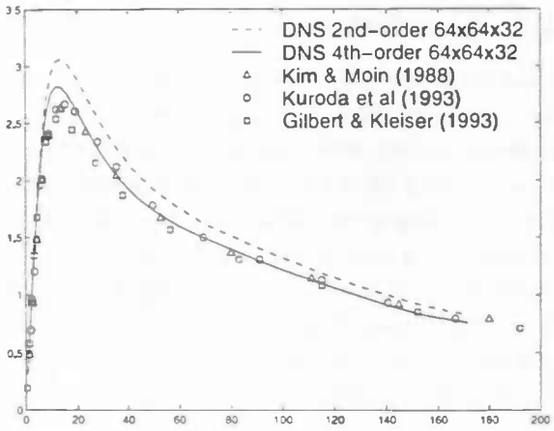


Figure 4.5: Comparison of T_{rms} in a fully developed channel,

Then, the dimensionless mean velocity $u^+ = \frac{\overline{u}}{u_\tau}$ is plotted versus $y^+ = u_\tau Re\ y$. The dimensionless mean temperature $T^+ = \frac{\overline{T}}{T_\tau}$ is also plotted versus $y^+ = u_\tau Re\ y$. The mean velocity profile is measured to compare it with those of other DNS's. Here it may be stressed that the grids used by the DNS's that we compare with have typically about 128^3 grid points, that is 16 times more grid points than our grid has. Nevertheless, the agreement is excellent. Figure 4.3 displays a comparison of the temperature profiles Here too the agreement is excellent. As shown in the figure, the agreement between the computed results and Kader's formula [6] (which is based on a large number of experimental results) is also good. Thus, we may conclude that a $64 \times 64 \times 32$ grid suffices to perform this DNS.
With Reynolds decomposition, the instantaneous quantities in a turbulent flow can be considered as the superposition of a mean, an average over a period which is much larger than the time scales of the turbulent fluctuations, and a fluctuating part:

$$u = \overline{u} + u' \tag{4.2}$$

$$T = \overline{T} + T' \tag{4.3}$$

The averages of the fluctuating parts are zero $\overline{u'} = \overline{T'} = 0$.
In Figures 4.4 and 4.5 u_{rms} and T_{rms} are plotted versus y^+ (linear scale) to compare with those of other DNS's. u_{rms} and T_{rms} are solved from Equations (4.2) and (4.3):

$$u_{rms}^2 = \overline{u'u'} = \overline{u^2} - (\overline{u})^2$$

$$T_{rms}^2 = \overline{T'T'} = \overline{T^2} - \left(\overline{T}\right)^2$$

In Figures 4.1 to 4.4 both the 2nd- and 4th-order results are shown. As can be seen, the results of the 2nd-order discretization method agree less with the reference data than those of the 4th-order method (except for the mean velocity profile; in the mean velocity profile no difference can be noticed). Therefore, we will use the fourth-order discretization to simulate the flow and heat transfer in the channel with the surface mounted cubes.

4.2 A channel with surface mounted cubical obstacles

In this section the computational domain is considered first. Next, all the values of the parameters and the prescribed boundary conditions are given. Then the results of the velocities are discussed and the last part of this section concerns the results of the heat transfer.

4.2.1 Computational domain and grid

A matrix of 25×10 cubes (each of size H^3) is mounted at one wall of the channel as sketched in Figure 4.2.1. The pitch of the cubes equals $4H$, both in the streamwise and in the spanwise direction. The height of the channel is $3.4H$. The flow domain is confined to a sub-channel unit of dimension $4H \times 3.4H \times 4H$.

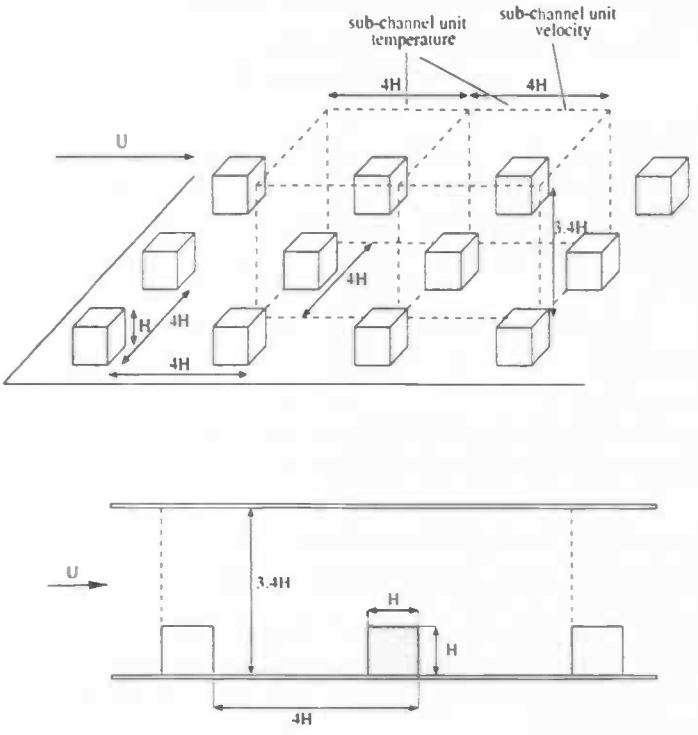


Figure 4.6: Three-dimensional view (upper plot) and side view (lower plot) of the channel with cubical mounted obstacles. In the lower figure is the middle cube the heated cube.

The sub-channel unit is covered by a 100^3 staggered grid that is stretched away from both the cubes and the channel walls. The first grid point away from a cube (or a wall) is located at $0.006H$. A cube is represented by 40 grid points in each direction. The grid is continued inside the heated cube. Therefore, the epoxy layer is represented by 5 grid points only.
For the heat transfer we have doubled the domain in the streamwise direction. The inlet is taken $4H$ upstream from the heated cube. The coordinate system originates from the channel wall at the centre of the windward face of the cube.

4.2.2 Parameters and boundary conditions

The parameters used in this simulation are:

Characteristic temperature	T_c	=	290.1 K	(ambient temparature)
Characteristic velocity	U	=	3.86 m/s	(bulk velocity)
Characteristic length	L	=	51 10^{-3} m	(channel height)
Reynolds number	$Re = \frac{UL}{\nu}$	=	13,000	
Prandtl number	$Pr = \frac{\mu c_p}{\lambda}$	=	0.71	
	$c_{dif} = \frac{\lambda_e}{\rho_e c_{p_e}} \frac{1}{UL}$	=	6.13 10^{-7}	
	$c_{lab} = \frac{\lambda_e}{\lambda}$	=	9.05	
	$c_{rad} = \frac{\epsilon \sigma T_c^3 L}{\lambda_e}$	=	0.28	
Thermal conductivity epoxy	λ_e	=	0.237 W/mK	
Thermal conductivity air	λ	=	0.0262 W/mK	
Thermal diffusivity epoxy	$a_e = \frac{\lambda_e}{\rho_e c_{p_e}}$	=	1.206 10^{-7} m^2/s	
Thermal diffusivity air	$a = \frac{\lambda}{\rho c_p}$	=	0.18 10^{-4} m^2/s	
Surface emissivity	ϵ	=	0.95	
Stefan-Boltzmann constant	σ	=	5.67 10^{-8} W/m^2K^4	
Thickness lower channel wall	d	=	8 10^{-3} m	
Copper temperature	T_{co}	=	348 K	
Ambient temperature	T_{amb}	=	290.1 K	

All these constants are taken from the Ph.D-thesis of Meinders [10].
At $4H$ downstream (measured from the windward face of the heated cube) the normal derivative of the temperature is set to zero. In addition, in a buffer zone (of length $0.4H$) the Prandtl number is decreased from 0.71 to 0.2 to suppress non-physical waves which may be reflected by the artificial outflow boundary.
A substantial difficulty appears in the coupling of air temperature with that of the heated cube; the diffusivity of air is approximately two orders of magnitude larger than that of epoxy. Consequently, the (diffusive) time scales in the air and in the epoxy layer differ significantly. Air reacts much faster upon temperature changes than epoxy does and it takes much longer to reach an equilibrium state in the epoxy layer than in the air. To shorten the time needed to reach an equilibrium the diffusivity in the epoxy layer is increased initially. It starts from a value that is slightly lower than the diffusivity of air and is then gradually decreased till it reaches its given value.
At $4H$ upstream from the heated cube the air temperature is put equal to $T = 17.1°C$. The surface temperature of the unheated cube upstream and that of the flat wall of the channel are also set to the ambient temperature. At the lower channel wall and at the surface of the windward face of the unheated cube downstream we have taken $T = 21°C$ to model the warm

up of the channel wall by the heated cube roughly.
To start with, we have taken the temperatures on a cube face from the Ph.D.-thesis of Meinders [10] to compute the radiation. After a while we have computed the radiation by means of the average measured temperature on a face. These average temperatures will be compared with the average temperatures on a cube face from the Ph.D.-thesis of Meinders [10] later on in this chapter.

4.2.3 Flow

In the preceding section on the channel with flat walls we have concluded that the Navier Stokes equations, the energy equation and the matching boundary conditions are well implemented. To obtain velocities in the channel with cubical mounted obstacles we only needed no-slip and periodic boundary conditions *i.e.* the same boundary conditions as in the channel with flat walls. This is not the case for the temperature. We have applied an extra equation (Equation (2.10)) and different boundary conditions (in streamwise direction and at the cube). Therefore we first discuss the velocity results.

An instantaneous flow field at two planes through the centre of the cubes is shown in Figure 1.1. This field resulted from the 100^3 DNS. In both pictures the flow is directed from left to right. The flow is characterized by the presence of distinct vortex structures in the vicinity of the obstacles. The vorticity ω is defined as the rotation of the instantaneous velocity field of a fluid element:

$$\omega = \nabla \times \mathbf{u}$$

It may be noted that large structures of recirculating flow behind the obstacles are not present in any of the snapshots of this flow. These regions can only be observed if the flow is averaged over a long period in time.
The statistics of the flow have been averaged over 540 seconds (40 shedding cycles). First- and second-order statistics of the velocity field obtained from the 100^3 DNS at the cross sec-

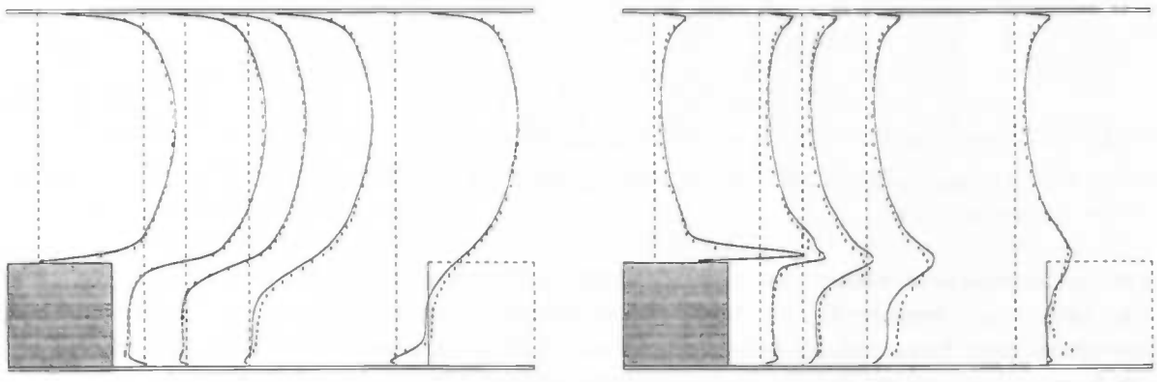


Figure 4.7: A comparison of first-order statistics (left picture) and second-order statistics (right picture) of the DNS with experimental data. Shown are the mean streamwise velocity $\overline{u}$ (left picture) and $\overline{u'u'}$ (right picture) in the plane parallel to the streamwise direction that bisects the cubes. The continuous lines correspond to the DNS; the experimental data is depicted by the dots.

tion of the channel that bisects a cube are compared to the available experimental data in Figure 4.7. The profiles of the mean streamwise velocity and the mean-square of the fluctuating streamwise velocity are in good agreement with the experiments, except in the front of a cube, where some discrepancies between the mean-squares of the fluctuating streamwise velocities exist. So, in conclusion, the 100^3 simulation reproduces the turbulent fluctuations reasonably well.

4.2.4 Heat transfer

At the cube-surface

Meinders *et al.* [10], [11] have measured the temperature at the surfaces of the heated cube by means of infrared thermography. At each face they have measured the surface temperature at a grid of 30^2 points. Liquid crystals were used to correct for the spatial image degradation of the infrared camera they used. In addition, they have computed the local (adiabatic) heat transfer coefficient

$$h_{ad} = \frac{-\lambda \frac{\partial T}{\partial n}|_{surface}}{T_{surface} - T_{amb}} \tag{4.4}$$

where the air temperature and its gradient are to be evaluated at the surface of the heated cube. Meinders *et al.* have computed the numerator of the heat transfer coefficient h_{ad} by writing it as the difference of the conductive and radiative heat fluxes at the surface, see Eq. (2.11). The radiative heat fluxes were calculated from the averaged temperature per face, the surface emissivity and the view factors of the faces. The conductive heat flux at the outer surface of the epoxy layer has been computed from the temperature distribution inside the heated cube. For that the Laplace equation was solved numerically on a 30^3 grid, where the temperature at the grid points on the faces was taken from the experiment.

	experiment				simulation			
face	q_{cond} W/m^2	q_{conv} W/m^2	q_{rad} W/m^2	T_f ^oC	q_{cond} W/m^2	q_{conv} W/m^2	q_{rad} W/m^2	T_f ^oC
windward	2976	2762	214.4	51.50	2775	2581	193.6	48.44
top	2592	2364	227.9	53.35	2687	2486	201.3	49.51
leeward	2084	1821	262.6	57.95	1717	1466	251.0	56.18
side I	2676	2445	230.6	53.74	2505	2295	210.1	50.73
side II	2597	2363	234.3	54.23	2505	2295	210.1	50.73

Table 4.4: Averaged heat fluxes and temperatures per cube face. A comparison of numerical results with experimental data.

Average temperatures and heat fluxes per face of the heated cube are compared in Table 4.4. Here, q_{cond} denotes the conductive heat flux through epoxy layer, q_{conv} represents the convective heat flux, and q_{rad} stands for the radiative heat flux. The average temperature per face is denoted by T_f. All quantities are given in the same physical dimensions as in the experiment. As can be seen the average surface temperatures of the simulation are lower than in the experiment. The largest difference occurs at the top face of the cube. There, the computed average surface temperature is approximately 7% lower than the measured temperature. The average temperatures at two side faces (side I and side II) of the cube

should be equal by symmetry. The same applies to the fluxes at the two side faces. The experimental results do not satisfy the symmetry perfectly, due to errors in the measurements. The difference between the experimental results at the two side faces gives an idea of the magnitude of the error in the experiment.

The integral of the total heat flux ($q_{cond} = q_{conv} + q_{rad}$) over the five faces of the cube is $2.67W$ according to the measurements and $2.74W$ in the numerical simulation. The total power dissipated in the cube during the experiment was $2.96W$. Thus, the loss through the base at which the cube is mounted ($2.96 - 2.67W$) is approximately four times the difference between the total heat flux in the experiment and in the numerical simulation.

The time-averaged temperatures along some paths at the surface of the heated cube are compared to those of the experiment in Figure 4.8. A number of measuring-points lie on the intersection of two paths. At these points two data-points are available. Both are shown in Figure 4.8 to illustrate the uncertainty in the experimental data. Given this uncertainty, we may conclude that the experimentally and numerically obtained mean temperature agree.

Yet, at the edges the numerical simulation predicts a temperature that is (much) lower than the measured temperature. The difference may become as large as five degrees Celcius, which is about 10%. This (relative) difference blows up if we look at the heat transfer coefficient h_{ad} (see Figure 4.9). Indeed, the denominator in the right-hand side of Eq. (4.4) may differ 20% (or even more) if the surface temperature differs 10%. The amplification of the difference between the measurements and the numerical simulation can be observed at the corners denoted by B and C in Figure 4.9.

In addition, it may be noted that the experimental results are somewhat questionable near the edges, because they do not satisfy the symmetry condition. This is illustrated in Figure 4.10. In this figure, we have encircled three pairs of data-points. Each couple is connected by a straight line. The slope of these line segments should be zero by symmetry. Obviously, this condition is not satisfied by the encircled experimental data-points, *i.e.* near the corners A, B, C and D. The middle-most picture in the left column of Figure 4.8 shows the corresponding, measured temperatures along the path ABCD are almost symmetrical. The difference between the temperature at two symmetric locations stays within one degree Celsius.

Near the base plate the surface temperature is predicted well. Yet, the heat transfer coefficient differs significantly from the measurements. This difference is a consequence of the simple model that we have used to describe the temperature of the base plate. In the simulation we have put the base plate at a uniform temperature of $21°C$ (except at the interface between the base plate and the cube), that is at the ambient temperature plus four degrees.

The h_{ad}-profiles at the line BC on the windward face display wiggles. These wiggles are likely due to the fact that the temperature gradient in the epoxy layer is not fully resolved (with only 5 grid points in the epoxy layer).

The distinct spatial gradients in the heat transfer coefficient at the surface of the heated cube are related to the complex flow patterns near the cube. Obviously, the heat transfer at the windward face is large compared to that at the other faces. At the leeward face, the heat transfer coefficient is low because a large fraction of the heat is detained in the recirculation bubbles in the wake of the cube. Heat convected away from the cube recirculates in the wake and thus causes an increase of the temperature at the rear side of the cube, which in its turn suppresses the transfer of heat from the rear side. This phenomenon can also be observed at a smaller scale. For example, the distance between the corner B and the location of the maxima of the heat transfer coefficient at the path AB in the pictures in the left-hand column

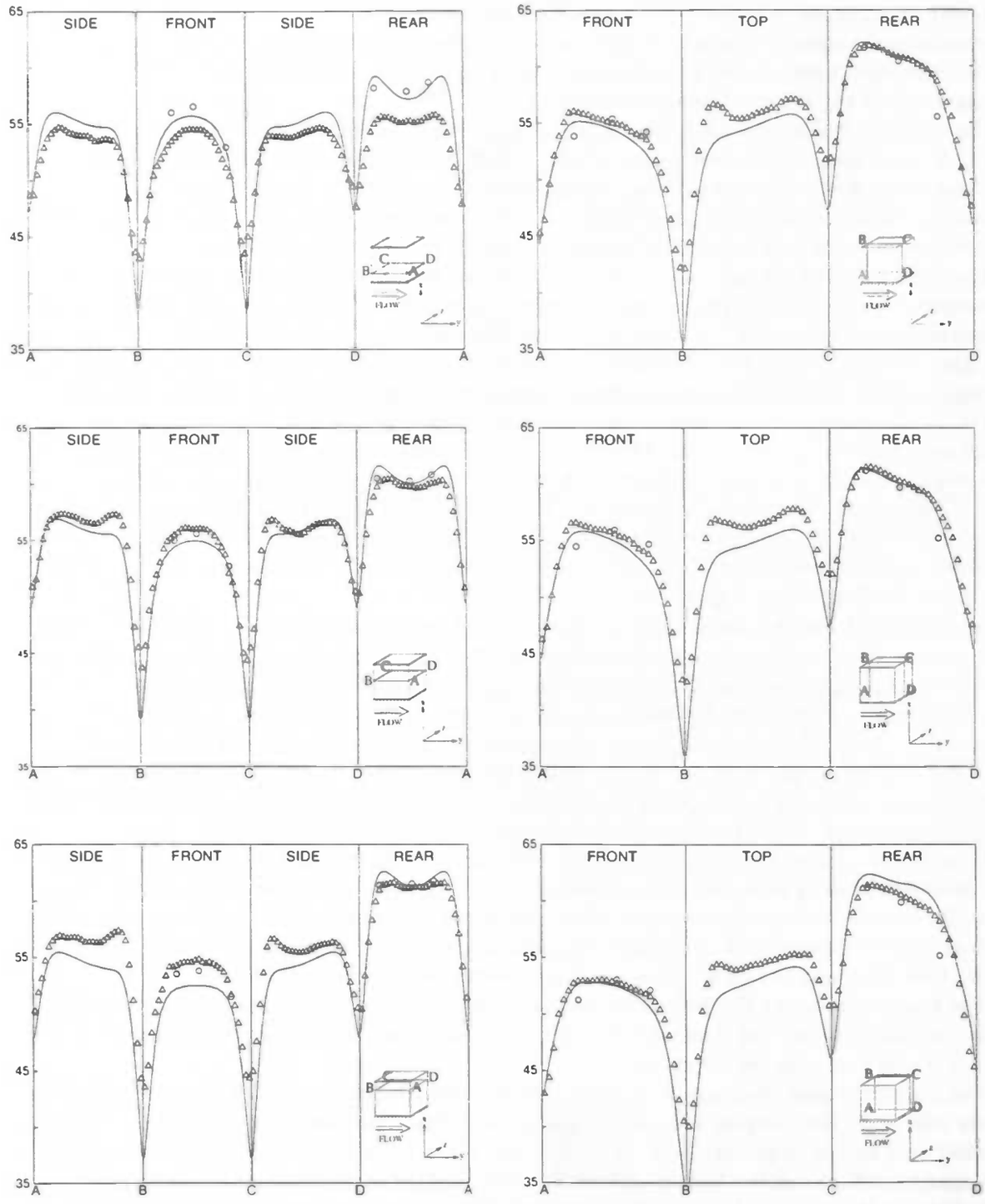


Figure 4.8: The mean temperature along different paths at the surface of the heated cube. The markers denote the experimental data by Meinders *et al.* Note that at three points along BC and DA (left-hand column) and at three points along AB and CD (right-hand column) two data-points are available. Both are shown.

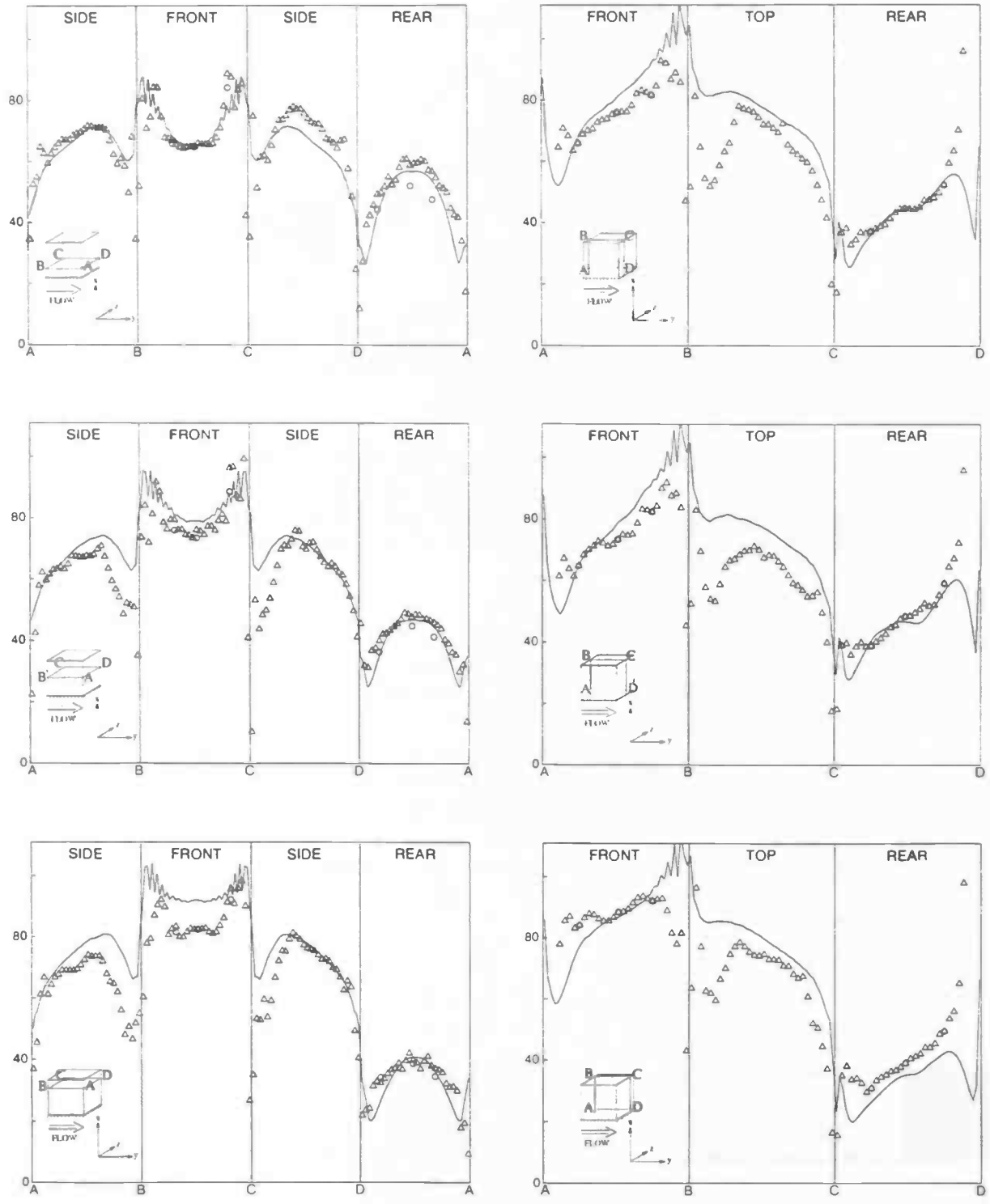


Figure 4.9: The mean heat transfer coefficient h_{ad} along different paths at the surface of the heated cube. The markers denote the experimental data by Meinders *et al.*

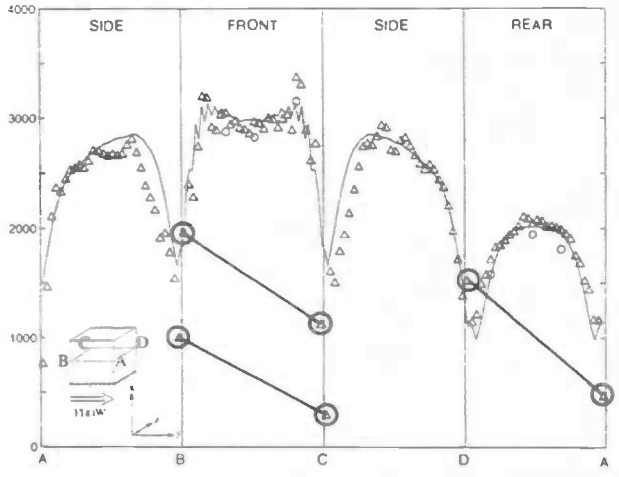


Figure 4.10: The convective heat flux along the path ABCDA. Three pairs of data-points are encircled to illustrate the asymmetry in the experimental data.

of Figure 4.9 is equal to the recirculation length along AB. In other words, the dip in the heat transfer at the side face near the corner B is caused by an eddy that reduces the removal of heat from this part of the surface. The same holds for path CD on the other side.

In the flow

Figure 4.11 and 4.12 depicts the mean and mean-square of the fluctuating temperature at four locations at/past the heated cube. Here, we have performed two computations. In addition to one described so far a temperature $\tilde{T}$ is computed from the energy equation (2.8). The temperature $\tilde{T}$ satisfies the boundary conditions that are imposed on T, except for the faces of the heated cube. There, $\tilde{T}$ is set equal to the averaged (in space and time) computed face temperature T_f (see Table 4.4). At the surface of the heated cube the difference between T and $\tilde{T}$ can be as large as 10 to 15 degrees. In the flow these differences disappear

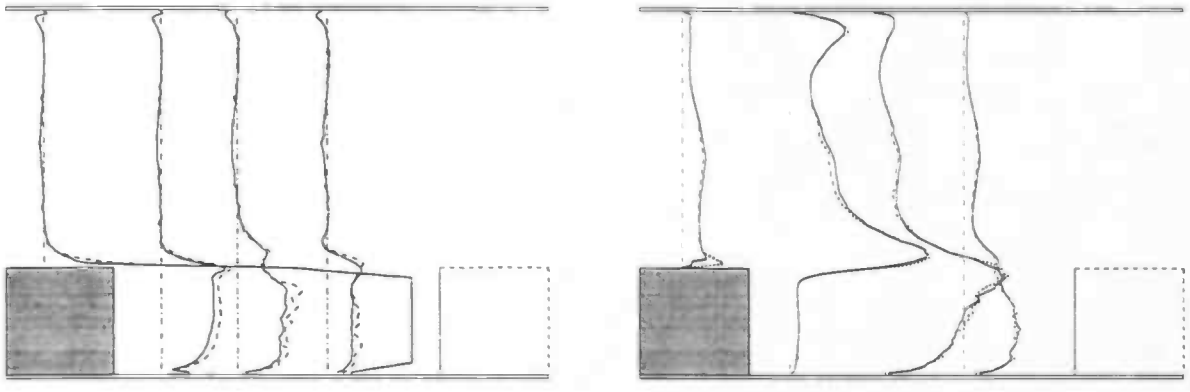


Figure 4.11: First-order statistics (left picture) and second-order statistics (right picture) of the temperature in the flow. Shown are the mean temperature $\overline{T}$ (left picture) and $\overline{T'T'}$ in the plane that bisects the cubes. The geometry is drawn to scale. In the left picture the difference between the temperature and the ambient temperature is shown. The continuous lines represent T; $\tilde{T}$ corresponds to the dashed lines.

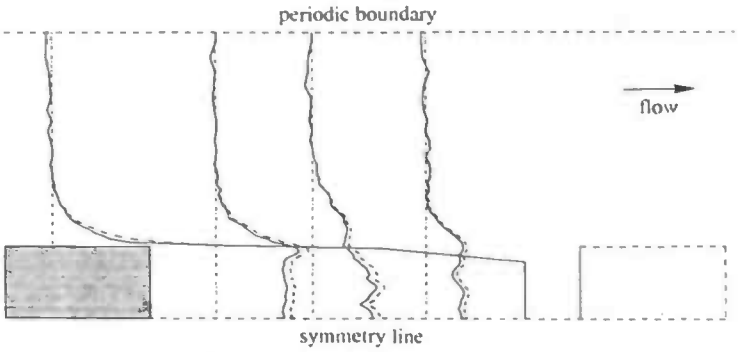


Figure 4.12: The mean temperature $\overline{T}$ at half cube height. The vertical corresponds to the spanwise direction. The continuous lines depicts T; $\tilde{T}$ is depicted by the dashed lines. The left-hand cube is heated. The geometry is drawn to scale. Shown is the difference between the temperature and the ambient temperature.

rapidly. Figure 4.11 and 4.12 show that this holds for the mean temperature (Fig 4.11 left picture and Fig 4.12) as well as for the mean-square fluctuating temperature (Fig 4.11 right picture). Therefore, we may conclude that the temperature distribution in the flow is relatively insensitive to variations of the surface temperature of the heated cube.

Chapter 5

Conclusions

In this report we have discussed the results of a 4th-order discretization method for computing the flow and heat transfer in a channel with surface mounted cubical obstacles. The results can be summarised as follows.

- For a channel with flat walls the results of the second-order discretisation method agree less with the reference data than those of the fourth-order method on the same computational grid. With the fourth-order method a $64 \times 64 \times 32$ grid suffices to perform a DNS at $Re = 5,600$ (based on channel height and bulk velocity).
- The turbulent flow profiles in the channel with the surface mounted cubical obstacles agree well with the available experimental data.
- The time-averaged surface temperatures also agree with the measurements, except for the edges of the cube where differences up to 10% exist. Differences between the measured and computed heat transfer coefficient are due to an amplification of the differences in the surface temperature, and to an asymmetry in the experimental data.
- The complex flow structures around the cubes yield strong variations in the local heat transfer coefficient.
- The temperature in the flow is relatively insensitive to variations of the surface temperature of the cubes.

Appendix A

Program Description

The numerical model in Chapter 3 has been implemented in a Fortran77 program. This appendix gives more information about the calling sequence, the subroutines, several variables used in the DNS-program and the in- and output files.

A.1 Calling sequence

The subroutines and functions are called in the following order:

setup	OPENIN		
	RDPARS		
	VERIFY		
	RDGRID		
	RDBCS		
	RDFLD		
	BNDCD0	BNDCDU	
		BNDCDV	
		BNDCDW	
	RDMNS		
	CELLS		
	RDGEOM		
	RDTCUB		
	CLOSIN		
	OPNOUT		
	PSOLVE0	POISSN	
		PRECON	
		PDROP0	ICCGP2
			BNDCDS
time integration	INTGRT	BNDCDU	
		BNDCDV	
		BNDCDW	
		UCELL	
		VCELL	
		WCELL	

	CNSTRNT	PSOLVE	ICCGP2
		BNDCDP	
		PDROP	
		BNDCDU	
		BNDCDV	
		BNDCDW	
	MAXDIV		
	MASFLW		
	INTTMP	TCELL	
		BNDCTC	
		TCUBSO	
		BNDCTC	
		BNDCDU	
		BNDCDV	
		BNDCDW	
		BNDCT	
	MEANS	MNXM	
		MNYM	
		MNZM	
	WRTFLD		
	TWALL		
	GRADT		
	WRTCUB		
	BULK	ENERGY	
post-processing	WRTMNS		
	CLSOUT		

The setup subroutines are executed once, while the integration in time is repeated until the maximum simulation time is expired. The individual subroutines/functions are discussed in the following section.

A.2 Subroutines

In this section a short description of the subroutines and functions is given.

BNDCD0 : Initializes boundary and ghost velocities.
BNDCDU : Computes boundary and ghost u-velocities.
BNDCDV : Computes boundary and ghost v-velocities.
BNDCDW : Computes boundary and ghost w-velocities.
BNDCDP : Computes boundary and ghost pressures.
BNDCDS : Computes boundary and ghost pressure drops.
BNDCT2 : Computes boundary and ghost temperatures.
BNDCTC : Computes virtual cube-temperatures.
BULK : Computes bulk quantities, such as the kinetic energy, output to **bulk.dat**.

CELLS : Defines the sizes of the control volumes.
CLOSIN : Closes the input files.
CLSOUT : Closes the output files.
CNSTRNT : Satisfies the constraint div $\mathbf{u} = 0$.
ENERGY : Computates the kinetic energy (function).
GRADT : Computes the average temperature gradients at the cube, output to **dtdnti.dat, dtdntj.dat** and **dtdntk.dat** .
ICCGP2 : Computes the conjugate gradient iteration.
INTGRT : Integrates the convective and diffusive terms in time.
INTTMP : Integrates the energy equation in time.
MASFLW : Computes the mass flow, output to **maxdiv.dat**.
MAXDIV : Computes the maximum of div **u**, output to **maxdiv.dat**.
MEANS : Computes and updates the averages.
MNXM : Computes and updates the averages at $x = x_m$.
MNYM : Computes and updates the averages at $y = y_m$.
MNZM : Computes and updates the averages at $z = z_m$.
OPENIN : Opens the input files.
OPNOUT : Opens the output files.
PDROP : Computes the pressure drop.
PDROP0 : Initializes the computation of the pressure drop.
POISSN : Discretizes div grad p.
PRECON : Computes preconditioner yz-periodic poisson.
PSOLVE : Poisson solver.
PSOLVE0 : Initializes the poisson solver.
RDBCS : Reads the boundary conditions from input file **bndconds**.
RDFLD : Reads the velocity, pressure and temperature from file **fld.dat** and **tmp.dat**.
RDGEOM : Reads the surfaces of the cells from input file **geometry.dat**.
RDGRID : Reads the coordinates of the grid-lines from input file **grid.dat**.
RDMNS : Reads the time-averages from files **mnx##.dat**, **mny##.dat** and **mnz##.dat**.
RDPARS : Reads the parameters from input file **parameters**.
RDTCUB : Reads the temperature in the cube from file **cub.dat**.
TCELL2 : Discretizes the convective and difusive flux through t-cells.
TCUBS0 : Computes the cube temperature.
TWALL : Computes the average wall temperatures at the cube, output to **twalli.dat, twallj.dat** and **twallk.dat**.
UCELL : Discretizes the convective and difusive flux through u-cells.
VCELL : Discretizes the convective and difusive flux through v-cells.
WCELL : Discretizes the convective and difusive flux through w-cells.
VERIFY : Verifies the values of the parameters.
WRTCUB : Writes the cube temperature to file **cub.dat**.
WRTFLD : Writes the velocity and pressure to file **fld.dat** and the temperature to file **tmp.dat**.
WRTMNS : Writes the averages to files **mnx##.dat, mny##.dat** and **mnz##.dat**.

A.3 Common block variables

Globally used variables are defined within *common blocks*. Here, the most important variables are given in alphabetical order.

- **/BNDCONDS/**

TX0, TX1	Temperature boundary conditions at the walls in x-direction, are not defined when periodic boundary conditions are used.
TY0, TY1	Temperature boundary conditions at the walls in y-direction, are not defined when periodic boundary conditions are used.
TZ0, TZ1	Temperature boundary conditions at the walls in z-direction, are not defined when periodic boundary conditions are used.
TX0GIV, TX1GIV	declares if TX0 and TX1 are Dirichlet or Neumann conditions.
TX0GIV, TX1GIV	declares if TY0 and TY1 are Dirichlet or Neumann conditions.
TX0GIV, TX1GIV	declares if TZ0 and TZ1 are Dirichlet or Neumann conditions.
TI0, TI1	(averaged) Temperatures at the walls of the heated cube in i-direction, used in the computation of the radiation.
TJ0, TJ1	(averaged) Temperatures at the walls of the heated cube in j-direction, used in the computation of the radiation.
TK0, TK1	(averaged) Temperatures at the walls of the heated cube in k-direction, used in the computation of the radiation.
TCO, TIN	Copper temperature and ambient/inlet temperature.

- **/CUBES/**

CL, CR, CDIF	$c_{lab}, c_{rad}, c_{dif}$
TCUBE(I,J,K)	Cube temperature at new time level at position (i, j, k)
TCEX(I,J,K)	Cube temperature at old time level at position (i, j, k)
CO(I,J,K), CW(I,J,K)	c_o and c_w at position (i, j, k)
CN(I,J,K), CS(I,J,K)	c_n and c_s at position (i, j, k)
CU(I,J,K), CD(I,J,K)	c_u and c_d at position (i, j, k)
CDIAG(I,J,K)	c_{diag} at position (i, j, k)

- **/DIMENSIONS/**

NX	Number of cells in x-direction.
NY	Number of cells in y-direction.
NZ	Number of cells in z-direction.

- **/FINITEVOL/**

ORDER	Order of the spatial discretization.
X(I), Y(J), Z(K)	x-, y- and z-coordinates of the grid points.
DX(i), DX3(I)	Grid size $x(i) - x(i-1)$ and large grid size $x(i+1) - x(i-2)$.
DY(J), DY3(J)	Grid size $y(j) - y(j-1)$ and large grid size $y(j+1) - y(j-2)$.
DZ(K), DZ3(K)	Grid size $z(k) - z(k-1)$ and large grid size $z(k+1) - z(k-2)$.

- **/PRANDTL/**

PR	Prandtl number.

- **/PRESSURE/**
  - P(I,J,K) Pressure at new time level at position (i, j, k).
  - PEX(I,J,K,N) Pressure at old time level n at position (i, j, k).
- **/REYNOLDS/**
  - RE Reynolds number.
- **/TEMPERATURE/**
  - T(I,J,K) Temperature at new time level at location (i, j, k).
  - TEX(I,J,K) Temperature at old time level at location (i, j, k).
- **/TIMEINTGRT/**
  - TIME Present time.
  - DT Time-step.
  - BETA β.
- **/VELOCITY/**
  - U(I,J,K), UEX(I,J,K) u-velocity at new and previous time-level at position (i, j, k).
  - V(I,J,K), VEX(I,J,K) v-velocity at new and previous time-level at position (i, j, k).
  - W(I,J,K), WEX(I,J,K) w- velocity at new and previous time-level at position (i, j, k).

Appendix B

List of symbols

c_p	specific heat of air at constant pressure
c_{p_e}	specific heat of epoxy at constant pressure
d	width of the channel wall
h	adiabatic heat transfer coefficient
$p(x,y,z,t)$	pressure
Pr	Prandtl number
Re	Reynolds number
t	time
$T(x,y,z,t)$	air temperature
T_{amb}	ambient temperature
T_{co}	copper temperature
$T_e(x,y,z,t)$	epoxy temperature
T_{in}	inlet temperature
T_f	average temperature on a cube face
$\mathbf{u} = \begin{pmatrix} u(x,y,z,t) \\ v(x,y,z,t) \\ w(x,y,z,t) \end{pmatrix}$	velocity
$\mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$	position
β	coefficient in one-leg method
ϵ	surface emissivity
λ	thermal conductivity of air
λ_e	thermal conductivity of epoxy
μ	dynamic viscosity
ν	kinematic viscosity
ρ	density
σ	Stefan-Boltzmann constant

Bibliography

[1] T. Craft (ed.), *Proceedings of the 7th ERCOFTAC/IAHR/COST Workshop on Refined Flow Modeling*, UMIST, Manchester, UK (1998).

[2] N. Gilbert and L. Kleiser, Turbulence model testing with the aid of direct numerical simulation results, *Proc. Turb. Shear Flows 8*, Paper 26-1, Munich (1991).

[3] K. Hanjalić and S. Obi (eds.) *Proceedings of the 6th ERCOFTAC/IAHR/COST Workshop on Refined Flow Modeling*, Delft University of Technology, The Netherlands (1997).

[4] A. Hellsten and P. Rautaheimo (eds.), *Proceedings of the 8th ERCOFTAC/IAHR/COST Workshop on Refined Flow Modeling*, Helsinki University of Technology, Finland (1999).

[5] H.W. Hoogstraten, *Stromingsleer*, RuG, lecture notes.

[6] B.A. Kader, Temperature and concentration profiles in fully turbulent boundary layers, *Int. J. Heat Mass Transfer* 24, 1541 (1981).

[7] N. Kasagi, Y. Tomita and A. Kuroda, Direct numerical simulation of passive scalar field in a turbulent channel flow, *ASME J. Heat Transfer* **114**, 598 (1992).

[8] J. Kim, P. Moin and R. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, *J. Fluid Mech.* **177**, 133 (1987).

[9] A. Kuroda, N. Kasagi and M. Hirata, Direct numerical simulation of turbulent plane Couette-Poisseuille flows: effect of mean shear rate on the near-wall turbulence structures, *Proc. Turb. Shear Flows 9*, F.Durst *et al.* (eds.), Berlin: Springer-Verlag 241-257 (1995).

[10] E.R. Meinders, *Experimental study of heat transfer in turbulent flows over wall-mounted cubes*, Ph.D-thesis, Delft University of Technology, The Netherlands (1998).

[11] E.R. Meinders, Th. van der Meer, K. Hanjalić and C.J.M. Lasance, Application of infrared thermography to the evaluation of local convective heat transfer on arrays of cubical protrusions, *Int J. Heat and Fluid Flow* **18**, 152 (1997).

[12] R.M. van der Velde, R.W.C.P. Verstappen, A.E.P. Veldman, Symmetry-preserving discretisation of turbulent flow and heat transfer, submitted to *J. Comp. Phys.* (1999).

[13] R.M. van der Velde, R.W.C.P. Verstappen, A.E.P. Veldman, Turbulent flow and heat transfer in a channel with surface mounted cubical obstacles, *Proc. ERCOFTAC/IAHR/COST Workshop on Refined Flow Modeling 8*, (1999).

[14] A.E.P. Veldman, *Numerieke Stromingsleer*, RuG, lecture notes.

[15] A.E.P. Veldman and K. Rinzema, Playing with nonuniform grids, *J. Engng. Math.* **26**, 119 (1991).

[16] R.W.C.P. Verstappen and A.E.P. Veldman, Direct numerical simulation of turbulence at lower costs, *J. Engng. Math.* **32**, 143 (1997).

[17] R.W.C.P. Verstappen and A.E.P. Veldman, Spectro-consistent discretization of Navier-Stokes: a challenge to RANS and LES, *J. Engng. Math.* **34**, 163 (1998).