Time-resolved incompressible turbulent channel flow at low Reynolds number

Davide Gatti (davide.gatti@kit.edu) and Bettina Frohnapfel
Karlsruhe Institute of Technology, Germany

1 Introduction

This benchmark dataset is designed to provide all researcher within the SPP time-resolved turbulent velocity fields of a channel flow at low Reynolds number and is intended for testing of postprocessing and visualization tools developed by the researcher within the SPP. At the present state, the database consists of:

- three-dimensional, three-component turbulent velocity fields
- two-dimensional views of the same turbulent channel flow, obtained by extracting:
  - a cross-sectional slice of the turbulent fields at a given streamwise position
  - a cross-sectional slice of the turbulent fields at a streamwise position that moves downstream with the bulk velocity of the flow
- trajectories of Lagrangian trackers

2 Governing equations

A turbulent channel flow is the flow between two indefinitely wide parallel plates placed at a distance $2h$ from one another. We indicate with x, y and z the streamwise, spanwise and wall-normal directions. The streamwise direction is the direction of the mean flow. u, v and w are the three velocity components in the three direction x, y and z respectively. The velocity vector $\mathbf{u} = \{u, v, w\}$ is indicated in bold font. The turbulent channel flow is governed by the incompressible Navier-Stokes equations:

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} = -\nabla p + \frac{1}{Re_\tau}\nabla^2 \mathbf{u} \tag{2}$$

where all variables have been nondimensionalized by the channel semi-height h^* and the friction velocity $u_\tau = \sqrt{\tau_w^*/\rho^*}$. Here the asterisk denotes dimensional quantities, ρ^* is the density of the fluid and τ_w^* is the time- and space-averaged wall shear stress, defined as $\tau_w^* = \rho^* \nu^* \langle \frac{\mathrm{d}u^*}{\mathrm{d}y^*} \rangle$. The

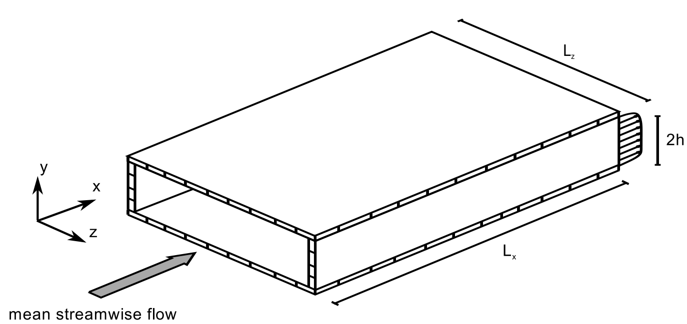

Figure 1: Sketch of a channel flow domain

operator $\langle\cdot\rangle$ denotes the average in both time and wall-parallel spatial directions and ν^* is the kinematic viscosity of the fluid. We consider the bottom wall to be located at $y = 0$ and the top wall at $y^* = 2$

The statistical properties of the turbulent flow depend on the Reynolds number Re, the nondimensional parameters appearing in equation 2. With the present nondimensionalization the Reynolds number is referred to as friction Reynolds number Re_τ and is defined as:

$$Re_\tau = \frac{u_\tau h}{\nu} \tag{3}$$

In the following the velocity $\mathbf{u}$ may be decomposed, whenever convenient, into two components according to the so-called Reynolds decomposition, defined as:

$$\mathbf{u} = \mathbf{U} + \mathbf{u}' \tag{4}$$

where the velocity $\mathbf{U}$ is the mean velocity and is defined as $\mathbf{U} = \langle\mathbf{u}\rangle$, whereas $\mathbf{u}'$ is the velocity fluctuation from the mean, defined as $\mathbf{u}' = \mathbf{u} - \mathbf{U}$.

3 Boundary conditions and flow pumping

The boundary conditions required by the problem are no-slip condition $\mathbf{u} = \mathbf{0}$ at the two channel walls, i.e. for each x and z at $y = 0$ and $y = 2$. At all other non-solid boundaries of the computational domain periodic boundary conditions are applied, i.e. $\mathbf{u}\,(x = 0, y, z) = \mathbf{u}\,(x = L_x, y, z)$ and $\mathbf{u}\,(x, y, z = 0) = \mathbf{u}\,(x, y, z = L_z)$. Here L_x and L_z are the streamwise and spanwise extent of the computational domain.

In order to drive the flow throughout the channel, a forcing term which mimic pumping needs to be provided. A constant pressure gradient $\mathrm{d}p/\mathrm{d}x = -1$ is enforced to this aim. This choice will result in a time- and space-averaged wall shear stress of $\tau_w = -h\mathrm{d}p/\mathrm{d}x = 1$. Since the mean pressure gradient is kept constant in time, the flow rate and wall friction will both fluctuate in time around their mean value.

4 Numerical method

The governing equation are solved with a pseudospectral solver which adopts mixed discretization [2]. Discrete Fourier series is used in the streamwise- and spanwise periodic directions, while fourth-order compact finite differences are adopted in the wall-normal direction. The velocity is therefore described as follows:

$$\mathbf{u}\,(x, y_i, z, t) = \sum_{m=-n_x}^{n_x} \sum_{n=-n_z}^{n_z} \hat{\mathbf{u}}_{m,n}\,(y_i, t)\, e^{2\pi \mathrm{i} m \kappa_x x} e^{2\pi \mathrm{i} n \kappa_z z} \tag{5}$$

where the hat denotes the complex Fourier coefficient, $\kappa_x = 2\pi/L_x$ and $\kappa_z = 2\pi/L_z$ are the streamwise and spanwise base wavenumbers. y_i is the wall-normal coordinate of the i-th collocation point of the compact finite differences. The position of such collocation points is inhomogeneous in the wall-normal direction and given by the following expression

$$y_i = \tanh(a(2(i-1)/n_y - 1))/\tanh(a) + 1 \qquad \text{for } i = 0, 1, ..n_y \tag{6}$$

here n_y is the total number of collocation points in the wall-normal direction, n_x and n_z are the number of Fourier modes in the respective directions, a is a stretch factor that determines how much grid refinement is performed in the vicinity of the wall. In the present numerical experiment $n_x = n_z = 48$, $n_y = 193$ and $a = 1.6$. The base wavenumber, which determine the streamwise and spanwise domain size are set to $\kappa_x = 0.8$ and $\kappa_z = 2$. The value of the friction Reynolds number is $Re_\tau = 150$. Equations are integrated in time with a mixed explicit, low-storage, third-order Runge-Kutta for the nonlinear convective term of the Navier-Stokes equations and Crank-Nicholson for

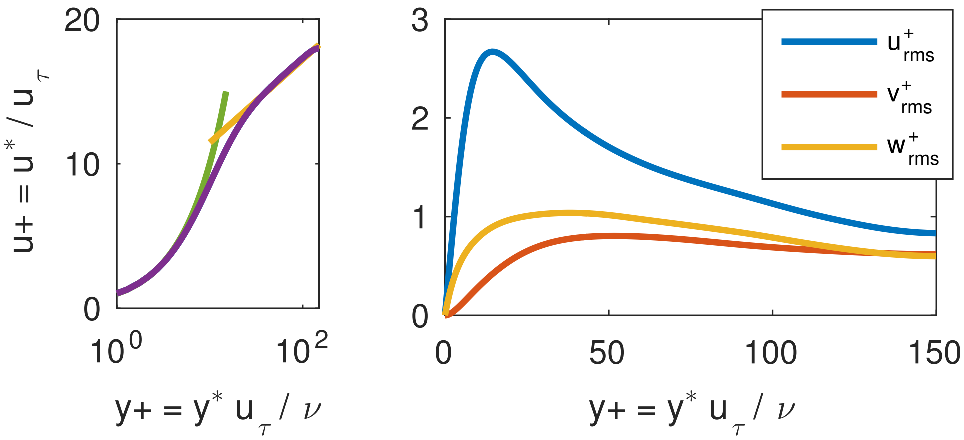

Figure 2: Left: mean flow velocity $U^+ = \langle u^+ \rangle$ as function of the wall-normal distance y^+ in viscous units. The linear ($u^+ = y^+$) and logarithmic ($u^+ = 2.5 \ln y^+ + 5.6$) behaviour expected from theory are also reported. Right: r.m.s. of the turbulent fluctuations as function of the wall-normal distance in wall units.

the linear part of the equations. Continuity is enforced exactly by projecting the equations in a divergence-free manifold. Dealiasing is performed according to the 2/3 rule. 10000 velocity fields are stored at each simulation timestep every $\Delta t = 0.001$ once the flow is observed to reach a statistically stationary state for a total of $10h/u_\tau$ time units, which correspond to 1500 viscous time units.

5 Basic flow characteristics

In the following, the superscript + indicates quantities which have been nondimensionalized via the friction velocity u_τ and the kinematic viscosity ν of the fluid. The mean bulk velocity, defined as $U_b^+ = \frac{1}{2}\int_0^2 U^+(y)\,\mathrm{d}y$, is $U_b^+ = 15.365$, which translates into a bulk Reynolds number of $Re_b = \frac{U_b h}{\nu} = 2305$. The friction Reynolds number is fixed by imposing a constant pressure gradient and is equal to $Re_\tau = 150$.

Figure 2(left) shows the mean velocity profile $U^+ = \langle u \rangle^+$, which follows the expected linear behaviour for $y^+ \leq 5$ and a "logarithmic" behaviour in a certain range of wall-normal distances [3]. The r.m.s. of the velocity fluctuations, defined as $u^+_{rms} = \sqrt{\langle u'u' \rangle^+}$, are reported in figure 2(right) The results are largely in agreement with those reported by Kim, Moser & Moin [1]. Further statistics of interest can be computed directly from the three-dimensional field or can be provided upon request.

The Reynolds shear stress $\langle u'v' \rangle^+$ is reported in figure 3, together with the viscous shear stress $\mathrm{d}U^+/\mathrm{d}y^+$. Their sum is expected to provide a linear total shear stress across the channel.

6 The 3D dataset

Three-dimensional velocity fields are written in unformatted binary format. Fields are numbered sequentially with integer number as "`pField1.fld`", "`pField2.fld`", ..., "`pField10000.fld`". Matlab and Python scripts are provided to load such fields and automatically define streamwise, spanwise and wall-normal coordinates. Such scripts are included in the database and comprise a guide.

Due to its sheer size, which tantalizes 1.6TB for all 10000 fields, the database is not uploaded to the SPP1881 central server at the moment but will be made available for direct download upon request (please write to davide.gatti@kit.edu).

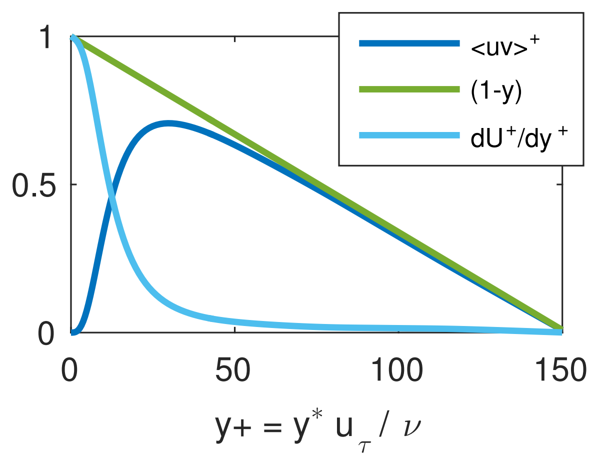

Figure 3: Reynolds shear stress $\langle u'v' \rangle^+$ and viscous stress $\mathrm{d}U^+/\mathrm{d}y^+$ as function of wall normal distance in wall units.

The data are saved in native u,v,w variables and in physical space. The grid size is $(n_x, n_y, n_z) = (192 \times 196 \times 192)$. The first and the last collocations points in the wall-normal direction are ghost-cells (i.e. points that lie outside the flow domain, below the bottom wall and above the top wall) which can be used to differentiate the velocity field when velocity gradients are needed, but can be otherwise disregarded.

7 The 2D datasets

The previous three-dimensional dataset is used to produce two-dimensional, cross-sectional slices showing a (z, y)-plane located at the streamwise position x_p. All three velocity components are stored on this plane. Two different cases are provided at the moment. In the first case, located in the directory "`2DSlice/konst_xp`", the streamwise position x_p is constant in time. In the directory "`2DSlice/variable_xp_Ub`" you can find the second case, in which the streamwise position of the cut plane moves downstream in time with a velocity equal to the mean bulk velocity of the flow, defined as $U_b = \frac{1}{2} \int_0^2 U(y)\, \mathrm{d}y$.

In both cases the grid size is $(n_y, n_z) = (196 \times 192)$. Again, the first and the last collocations points in the wall-normal direction are ghost-cells, which can be used to differentiate the velocity field when velocity gradients are needed, but can be otherwise disregarded. In each case the fields are named sequentially with integer number as "`2DField1.fld`", "`2DField2.fld`", ..., "`2DField10000.fld`".

7.1 Postprocessing of data

The data can be loaded and postprocessed with any program of your choice. For convenience, we provide here in the following examples with matlab or python3.

7.1.1 using Matlab

```
% This script loads a 2D Field with field number "iF" from the
% current working directory.
%
% In the velocity field matrix U=(1:3,1:nz,1:ny+3) the indeces have
% following meaning:
%
```

```
% 1st - velocity component (1=u,2=v,3=w)
% 2nd - spanwise position
% 3rd - wall-normal position

% Field number
iF = 1;

% Discretization parameters (no need to change)
% ---------------------------------------------
nz=192;  % n. of points in the spanwise direction
ny=193;  % n. of points in the fluid domain in the
         % wall-normal direction
a=1.6;   % near-wall grid-refinement parameter
kz=2;    % spanwise base wavenumber
% ---------------------------------------------

% Define y-coordinates
y=zeros(ny+3,1);
for i=1:ny+3
    y(i)=tanh(a*(2*(i-2)/ny-1))/tanh(a)+1;
end

% Define z-coordinates
z=linspace(0,2*pi/kz,nz);

% Open field
f=fopen(strcat('2DField',num2str(iF),'.fld'));
% Read velocity field
U=reshape(fread(f,'double'),[3,nz,ny+3]);
% Close field
fclose(f);

% Remove ghost cells when not needed
U = U(:,:,2:end-1); y = y(2:end-1);

% Do your postprocessing...
```

7.1.2 using Python3

For using the following script the python module Numpy is required.

```
import numpy as     np
from   numpy import pi

# Field number
iF = 1;

# Discretization parameters (no need to change)
# ---------------------------------------------
nz=192;  # n. of points in the spanwise direction
ny=193;  # n. of points in the fluid domain in the
         # wall-normal direction
a=1.6;   # near-wall grid-refinement parameter
```

```
kz=2;      # spanwise base wavenumber
# ––––––––––––––––––––––––––––––––

# Define y- and z-coordinates
y = np.tanh(a*(2*np.arange(-1,ny+2)/ny-1))/np.tanh(a)+1
z = np.linspace(0,2*pi/kz,nz)

# Load data
U = np.fromfile('2DField'+str(iF)+'.fld',np.float64).reshape(ny+3,nz
    ,3)

# Remove ghost cells if not needed
U = U[1:-1,...]; y = y[1:-1]

# Do your postprocessing ...
```

8 Lagrangian tracker trajectories

A database of Lagrangian particle trackers trajectories has also been produced. $n_{\text{particles}} = 10000$ particles have been seeded into the flow at random initial positions in the streamwise and spanwise directions. The wall-normal positions of the particles is not random and has been chosen so that 1/10 of the total number of particles is located at the following wall-normal initial location $y/h = \{0.0055, 0.0178, 0.0245, 0.0316, 0.0391, 0.0471, 0.0556, 0.0646, 0.0741, 0.0842\}$. This means that particles have been seeded preferentially very close to the bottom wall of the channel and their evolution is tracked in time. This choice of wall-normal particle locations is arbitrary and may be changed in the future or upon request. This database serves at the moment as benchmark for testing postprocessing tools of other members within the SPP1881.

The position of the particles is integrated forward in time together with the Navier-Stokes equation, i.e. for 10000 timesteps. At every timestep and for every particle, the three spatial coordinates and the three velocity components are saved. Access to this database can be obtained by writing an email to davide.gatti@kit.edu.

The data can be read via the following Matlab script:

```
% This script loads the particle tracking data

% In the matrix data=(1:6,1:nparticles,1:nsteps) the indeces have
% following meaning:
%
% 1st - variable of interest (1=x,2=y,3=z,4=u,5=v,6=w)
% 2nd - particle number tag
% 3rd - timestep


% Discretization parameters (no need to change)
% ––––––––––––––––––––––––––––––––
nvars=6;
nparticles=10000;
nsteps=10000;
% ––––––––––––––––––––––––––––––––

f = fopen('particles.bin');
data = reshape(fread(f,'double'),[nvars,nparticles,nsteps]);
fclose(f);
```

```

% Do your postprocessing...
```

References

[1] J. Kim, P. Moin, and R. Moser. Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.*, 177:133–166, 1987.

[2] P. Luchini and M. Quadrio. A low-cost parallel implementation of direct numerical simulation of wall turbulence. *J. Comp. Phys.*, 211(2):551–571, 2006.

[3] S.B. Pope. *Turbulent Flows*. Cambridge University Press, Cambridge, 2000.