Flow Control on the basis of a FEATFLOW-MATLAB Coupling

Lars Henning[1], Dmitri Kuzmin[3], Volker Mehrmann[2], Michael Schmidt[2],
Andriy Sokolov[3], and Stefan Turek[3]

[1] Technische Universität Berlin, Fachgebiet Mess- und Regelungstechnik, 10623 Berlin, Germany, lars.henning@tu-berlin.de
[2] Technische Universität Berlin, Fachgebiet Numerische Mathematik, 10623 Berlin, Germany, {mehrmann,mschmidt}@math.tu-berlin.de
[3] Universität Dortmund, Fachbereich Mathematik, 44227 Dortmund, Germany, {kuzmin,asokolow,stefan.turek}@math.uni-dortmund.de

Summary

For the model-based active control of three-dimensional flows at high Reynolds numbers in real time, low-dimensional models of the flow dynamics and efficient actuator and sensor concepts are required. Numerous successful approaches to derive such models have been proposed in the literature.

We propose a software environment for a comfortable and performant testing of control, actuator and sensor concepts which may be based on such models. It is realized by providing an easily manageable MATLAB control interface for the k-ε-model from the FEATFLOW CFD package. Potentials and limitations of this tool are discussed by considering exemplarily the control of the recirculation bubble behind a backward facing step.

1 Introduction

Active control of fluid flow is one of the major challenges in many key technologies, e.g. in chemical process engineering or aeronautics. In this paper we will discuss a model problem for such applications, the active control of the length of a recirculation bubble behind a backward facing step which is very difficult to solve with current mathematical methods.

Active control methods require the interaction of the controller with either the physical system (via measurements) or with a numerical simulation (via an observer). Here we only discuss the latter interaction. A major problem in designing real time controllers is that the observations have to be provided fast enough and that the computational method needed in the controller design is efficient enough. These two requirements call for fast numerical simulation methods and small state dimensions in the dynamical system that describes the flow. On the other hand the observed data have to be good enough to allow the computation of a good control, which typically means that the simulation method has to produce sufficiently accurate approximations of the observed data.

In order to design such controllers, it is therefore very important to use the most suitable model assumptions for this interactive task and to coordinate the interaction of controller design and numerical simulation by creating an efficient interface between simulation and controller design. Such an efficient coupling of current commercial CFD packages and modern control software requires detailed knowledge about the CFD code and how it can be coupled with the control software.

For this reason we discuss the coupling of the CFD code FEATFLOW[3] with classical control techniques as they are available from MATLAB [1] or SLICOT [2] toolboxes.

In Section 2 we present as new feature a MATLAB-control interface, which has been developped for FEATFLOW's RANS solver with k-ε turbulence model and which is planned for FEATFLOW's DNS and LES solvers (see Fig. 1). We present in this paper as example just the interface for the RANS solver, though the discussion in Section 4 will show that DNS or LES may be a better choice for certain control configurations.

The active control of the length of the recirculation bubble by insufflation and suction at the edge of the backward facing step is described in Section 3. We will demonstrate how control strategies can be easily developed and implemented, only requiring minimal insight in the operation of the CFD code FEATFLOW.

Finally, in Section 4 we discuss the potentials and limitations of this tool.

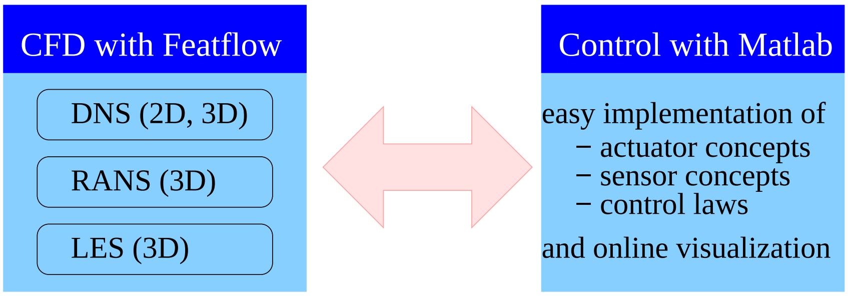

Figure 1 MATLAB control interface for FEATFLOW CFD codes, based on *Direct Numerical Simulation* (DNS), *Reynolds-averaged Navier-Stokes equations* (RANS) and *Large Eddy Simulation* (LES).

2 New features of FEATFLOW

2.1 A realization of the $k - \varepsilon$ turbulence model

As a finite element 3D solver for the incompressible Navier-Stokes equations we use the CFD code FEATFLOW. The underlying numerical algorithm is based on (nonconforming) FEM discretizations, adaptive implicit time-stepping and (geometric) multigrid solvers on general domains. FEATFLOW is an open-source software package built on the FEAT2D and FEAT3D libraries written in FORTRAN77 [3].

Dealing with three-dimensional flows at high Reynolds numbers, the numerical costs of DNS (Direct Numerical Simulation) are extremely high. For instance, a 3D simulation of the backward facing step with Reynolds number $Re = 30\,000$ requires about $\mathrm{Re}^{\frac{9}{4}} \geq 10^{10}$ nodes to resolve the smallest eddies [4]. Therefore, a k-ε turbulence model was added to FEATFLOW, in order to make the calculation of such flows on meshes of moderate size possible. The corresponding CFD code **PP3D-KE** was developed by D. Kuzmin building on the laminar FEATFLOW version (http://www.featflow.de). The mathematical foundations of the program can be described as follows.

We consider the following system of *Reynolds-averaged Navier-Stokes* (RANS) equations:

$$\begin{aligned} &\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \nabla \cdot ((\nu_0 + \nu_\mathrm{T}) \mathcal{D}(\mathbf{v}))\,, \\ &\nabla \cdot \mathbf{v} = 0. \end{aligned} \tag{1}$$

Here $\mathbf{v} = \mathbf{v}(t;\mathbf{x})$ with $\mathbf{v} = (\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z)^T \in \mathbb{R}^3$ is a time-averaged velocity and $p = p(t;\mathbf{x}) \in \mathbb{R}$ is a time-averaged pressure, both defined on a time-space domain $(0,T) \times \Omega$ with $T > 0$ and $\Omega \subset \mathbb{R}^3$. $\mathcal{D}(\mathbf{v}) = \nabla \mathbf{v} + (\nabla \mathbf{v})^T$ is the Newtonian strain tensor and $\nu_\mathrm{T} = C_\mu \frac{k^2}{\varepsilon}$ is the effective viscosity. The turbulent kinetic energy $k = k(t;\mathbf{x})$ and its dissipation rate $\varepsilon = \varepsilon(t;\mathbf{x})$ are modeled by the following scalar transport equations:

$$\begin{aligned} &\frac{\partial k}{\partial t} + \nabla \cdot \left(k\mathbf{v} - \frac{\nu_T}{\sigma_k} \nabla k \right) = P_k - \varepsilon, \\ &\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left(\varepsilon\mathbf{v} - \frac{\nu_T}{\sigma_\varepsilon} \nabla \varepsilon \right) = \frac{\varepsilon}{k}(C_1 P_k - C_2 \varepsilon), \end{aligned} \tag{2}$$

where $P_k = \frac{\nu_T}{2} |\nabla \mathbf{v} + \nabla \mathbf{v}^T|^2$. The default values of the involved empirical constants are $C_\mu = 0.09$, $C_1 = 1.44$, $C_2 = 1.92$, $\sigma_k = 1.0$, $\sigma_\varepsilon = 1.3$.

Additionally, appropriate boundary conditions for $\mathbf{v}$, k and ε have to be prescribed on $\partial\Omega = \Gamma_\text{in} \cup \Gamma_\text{out} \cup \Gamma_\text{wall} \cup \Gamma_{sym}$.

As usual, Dirichlet boundary conditions for $\mathbf{v}$, k and ε are prescribed on the inflow boundary Γ_{in},

$$\mathbf{v} = \mathbf{g}, \quad k = c_{ab} |\mathbf{v}|^2, \quad \varepsilon = C_\mu \frac{k^{3/2}}{l_0}, \tag{3}$$

where c_{ab} is an empirical constant [5], l_0 is a mixing length and $\mathbf{g}$ is a given inflow velocity profile.

Let $(\mathbf{t}, \mathbf{n})$ be the local orthogonal basis for a wall node, where $\mathbf{t}$ and $\mathbf{n}$ are the tangential and normal directions, respectively. At the outlet $\mathit{\Gamma}_{out}$ the following 'do-nothing' boundary conditions are prescribed:

$$\frac{\partial \mathbf{v}}{\partial \mathfrak{n}} = 0, \quad \frac{\partial k}{\partial \mathfrak{n}} = 0 \qquad \frac{\partial \varepsilon}{\partial \mathfrak{n}} = 0, \quad \mathbf{n} \cdot [pI - \nu_T \mathcal{D}(\mathbf{v})] = 0. \tag{4}$$

In the $k - \varepsilon$ model the behavior of a fluid near an impervious solid wall is modeled by wall functions. The computational wall boundary $\mathit{\Gamma}_{\text{wall}}$ is located at a distance δ from the real geometrical wall boundary. In our case we assume that the computational domain is already reduced by a layer of width δ, which is a user-defined parameter. We set the following boundary conditions on $\mathit{\Gamma}_{\text{wall}}$:

$$\mathbf{n} \cdot \mathbf{v} = 0, \quad (\mathcal{D}(\mathbf{v})\mathbf{n} \cdot \mathbf{t})\mathbf{t} = -\frac{u_\tau^2}{\nu_T} \frac{\mathbf{v}}{|\mathbf{v}|}, \quad k = \frac{u_\tau^3}{\sqrt{C_\mu}} \quad , \quad \varepsilon = \frac{u_f^\tau}{\kappa \delta}, \tag{5}$$

where $\kappa = 0.41$ is the von Kármán constant and u_τ is the friction velocity. Here u_τ is the solution of the logarithmic wall law equation:

$$|\mathbf{v}| = u_\tau \left(\frac{1}{\kappa} \log y^+ + 5.5\right), \tag{6}$$

where $y^+ = \frac{u_\tau \delta}{\nu}$ is the local Reynolds number.

In our numerical simulations of the backward facing step we try to avoid the influences of the side walls and the upper wall on the recirculation zone, therefore we impose symmetry boundary conditions on $\mathit{\Gamma}_{sym}$ [6]:

$$\mathbf{n} \cdot \mathbf{v} = 0, \quad \mathcal{D}(\mathbf{v}) \cdot \mathbf{n} = 0. \tag{7}$$

The discretization in space is performed by an unstructured grid finite element method [7]. A detailed description of the numerical algorithm for the $k - \varepsilon$ model can be found in [5], [8].

2.2 A FEATFLOW-MATLAB control interface

Dealing with flow control problems, we assume that we are able to influence the flow in $\mathit{\Omega}$ by manipulating the Dirichlet boundary conditions in a subset $\mathit{\Gamma}_{ctrl} \subset \mathit{\Gamma}_{wall}$, i.e.

$$\mathbf{v}(t; \mathbf{x}^{(c)}) = \mathbf{u}(t; \mathbf{x}^{(c)}) \quad \text{for } (t; \mathbf{x}^{(c)}) \in (0, T) \times \mathit{\Gamma}_{ctrl},$$

with a *control* or *input* function $\mathbf{u}(t; \mathbf{x})$. We assume further that we can observe or measure the velocity field and/or pressure field of the fluid in subsets $\mathit{\Omega}_{meas} \subset \mathit{\Omega}$ and/or $\mathit{\Gamma}_{meas} \subset \mathit{\Gamma}$, i.e. that we know the *observation* or *output* function

$$\mathbf{y} = \mathbf{y}(t, \mathbf{x}^{(m)}), \quad \mathbf{y} = (\mathbf{v}, p), \quad \text{where } (t, \mathbf{x}^{(m)}) \in (0, T) \times (\mathit{\Omega}_{meas} \cup \mathit{\Gamma}_{meas}).$$

A (feedback) controller $\mathbf{u}(t, \mathbf{x}^{(c)}) = f(t, \mathbf{y}(s \leq t, \mathbf{x}^{(m)}))$ can then be used for the calculation of appropriate controls $\mathbf{u}$, possibly on the basis of current and former observations $\mathbf{y}$, in order to achieve some control objective. Aiming to implement such controllers comfortably in MATLAB, but carrying out the flow calculations in a perfomant manner with FEATFLOW, a FEATFLOW-MATLAB control interface has been developed.

The interaction between the **PP3D-KE** program and MATLAB was realized with the help of the MATLAB Engine (function `engOpen`), which operates by running in the background as an independent process. This offers several advantages: under UNIX, the MATLAB engine can run on the user's machine, or on any other UNIX machine on the user's network, including those of a different architecture. Thus, the user can implement a user interface on his/her workstation and perform the computations on a faster machine located elsewhere on a network (see MATLAB Help).

The MATLAB-controller part is realized as an m-file *MatlabController.m*. During the simulation, the **PP3D-KE** code calls this MATLAB function at every time step. The transaction phase consists of three stages: receiving the required data (geometry, velocity, pressure) from the output domain Γ_{meas}, execution of *MatlabController.m* and calculation of a control $\mathbf{u} = (\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z)$, setting $\mathbf{v} = \mathbf{u}$ as a Dirichlet boundary condition for velocity in the input domain Γ_{ctrl}. So *MatlabController.m* has, in principle, the following interface:

$$\tilde{\mathbf{v}}(\mathbf{x}_i^{(c)}) = \text{MatlabController}(t_n, \mathbf{x}_j^{(m,v)}, \mathbf{x}_k^{(m,p)}, \tilde{\mathbf{v}}(\mathbf{x}_j^{(m,v)}), \tilde{p}(\mathbf{x}_k^{(m,p)})).$$

Here, t_n is the current time, $\mathbf{x}_j^{(m,v)}$, $\mathbf{x}_k^{(m,p)}$ are the coordinates of the velocity and pressure nodes lying in $\Gamma_{meas} \cup \Omega_{meas}$, and $\tilde{\mathbf{v}}(\mathbf{x}_j^{(m,v)})$ and $\tilde{p}(\mathbf{x}_k^{(m,p)})$ are the corresponding discrete velocity field and pressure field, respectively, all communicated by FEATFLOW to MATLAB. After the control is executed on the basis of this information (and possibly of similar data computed at previous time steps), the discrete velocity field $\tilde{\mathbf{v}}(\mathbf{x}_i^{(c)})$ with respect to $\mathbf{x}_i^{(c)} \in \mathbb{R}^3$, the coordinates of the velocity nodes lying in Γ_{ctrl}, is communicated by MATLAB to FEATFLOW.

3 Controlling the flow over a backward facing step

3.1 A benchmark configuration

We illustrate the facilities and limitations of the new FEATFLOW-MATLAB control interface by considering as example the flow over a backward facing step. One of the main features of such flows is a recirculation region just downstream of the step, and we aim to control its length by means of insufflation and suction at the edge of the step, see Fig. 2.

The uncontrolled flow over the step is a classical benchmark for the validation of CFD codes, since it is well-understood analytically and extensive experimental results are available (see e.g. [10–14]). FEATFLOW's **PP3D-KE** has also been validated for it [8].

The control of the recirculation length is the subject of numerous experimental, numerical and analytical studies, and has, e.g., been intensively investigated in the

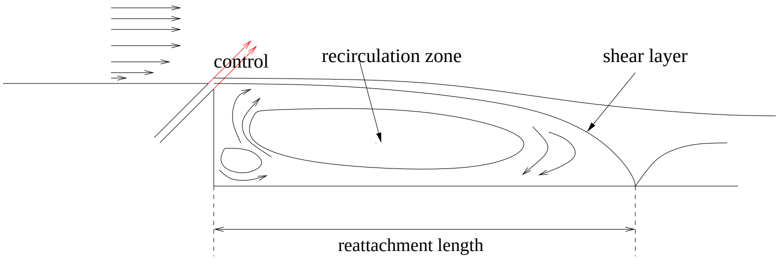

Figure 2 Flow over a backward facing step (see [9])

DFG-Collaborative Research Center ”Control of complex turbulent shear flows” (see e.g. [9, 15–20]). We mention some model-based active control concepts for the step flow from the literature, a more extensive survey can be found in [21].

Applying the theory of optimal control to the NSE, open loop controls for the step flow are investigated and numerically calculated in [22–24]. In order to reduce the numerical costs and to obtain a more robust feedback-like control, suboptimal control strategies like instantaneous control are applied in [24–27]. Based on the full NSE, these approaches will mostly be limited in the near future to low Reynolds number flows in two-dimensional domains.

For the real-time control of three-dimensional flows at high Reynolds numbers low-dimensional models are required. In [28–31], low dimensional Galerkin models are derived from the full NSE by means of the method of proper orthogonal decomposition (POD) on the basis of snapshots from a DNS. In [18, 19] low-dimensional Galerkin models are derived by combining classical POD modes and physically motivated transition modes in order to better capture the flow dynamics. In [20] the flow dynamics are described by low-dimensional vortex models without that the NSE have to come into play. Finally, in [9] black-box models are identified on the basis of experimental and numerical data and used for the design of robust controllers which then worked in real-time experiments and in LES simulations, respectively.

Whereas the approaches in [22–31] use general mathematical reduction methods and control concepts, the approaches in [18–20, 9] use flow-specific physical insight for the modeling and for the control concept. For instance, they make use of the so-called Kelvin-Helmholz instability of the shear layer above the recirculation zone with a characteristic frequency f_{shear}. A systematic excitation of these instabilities can lead to a shortening of the recirculation zone, and provides the basis for efficient and well-realizable controls via periodic suction and insufflation.

It is not the purpose of this paper to propose another control strategy for the backward facing step. We will show to what extent FEATFLOW in connection with the k-ε-model and the MATLAB control interface can provide a comfortable and performant software environment for *testing* such control concepts for three-dimensional and high Reynolds number flows. And we will see that it may even be used to *design* simple but efficient controls.

3.2 Implementation in FEATFLOW and MATLAB

As a specific test case, we consider the flow over a step of dimensionless height $h = 1$. The inlet section has the length $5h$ and the wake section the length $20h$. The height as well as the width of the domain are $3h$. We choose x, y and z as coordinates for the downstream, the vertical and the spanwise direction, respectively, see Fig. 3.

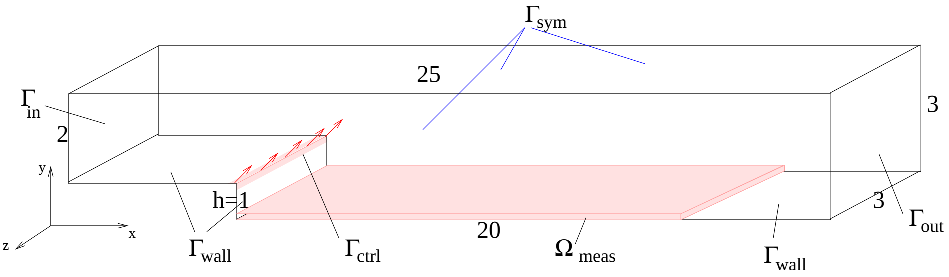

Figure 3 The computational domain

We assume that we can blow out and suck in fluid at an angle of 45 degree in positive x-y-direction at a slot at the edge of the step of width $0.05h$ in the x and y direction, i.e.

$$\mathbf{v}(t; x, y, z) = \frac{1}{\sqrt{2}}(u(t), u(t), 0)^T \quad \text{on } (0, T) \times \mathit{\Gamma}_{ctrl},$$

where $u(t)$ is a scalar control function that can be freely varied in time and $\mathit{\Gamma}_{ctrl} = \{(x, y, z) \in \mathit{\Gamma} : 4.95 \leq x \leq 5, 0.95 \leq y \leq 1, 0 < z < 3\}$. Note that the implementation of distributed vector-valued controls $\mathbf{v}(t; x, y, z) = \mathbf{u}(t; x, y, z)$ is also possible.

The length of the recirculation zone is defined via the reattachment position x_r of the shear layer detaching at the edge of step. For each z, $x_r(t, z)$ is defined by a zero wall shear stress $\tau_w(t; x, z) = 0$, with

$$\tau_w(t; x, z) = \eta \left(\frac{\partial \mathbf{v}_x}{\partial y} \right) |_{(t;x,y=0,z)}, \tag{8}$$

where η denotes the viscosity [9]. We will determine $\tau_w(t; x, z)$ by measuring $\mathbf{v}_x(t; x, y, z)$ in the domain $\mathit{\Omega}_{meas} = \{(x, y, z) : 5 < x < 20, \quad 0 < y < 0.125, \quad 0 < z < 3\}$ and define a scalar reattachment length $x_r(t)$ by averaging $\tau_w(t; x, z)$ in spanwise direction.

The flow is governed by the dimensionless NSE (1). Prescribing a constant block velocity profile $\mathbf{v}_\infty = (1, 0, 0)^T$ at the inflow boundary, we apply the boundary conditions

$$
\begin{aligned}
\mathbf{v}(t;x,y,z) &= \mathbf{v}_\infty && \text{on } \mathit{\Gamma}_{in} && \text{(inhom. Dirichlet)}, && (9a)\\
\partial_z \mathbf{v}(t;x,y,z) &= 0 && \text{on } \mathit{\Gamma}_{out} && \text{(hom. Neumann)}, && (9b)\\
\partial_n \mathbf{v}(t;x,y,z) &= 0 && \text{on } \mathit{\Gamma}_{sym}, && \text{(symmetry)}, && (9c)\\
\mathbf{v}(t;x,y,z) &= 0 && \text{on } \mathit{\Gamma}_{wall}, && \text{(no slip)}. && (9d)
\end{aligned}
$$

We now discuss the numerical implementation of the mathematical model. Dealing with nonstationary high Reynolds number flows in a three-dimensional domain, we use the RANS-solver **PP3D-KE** from FEATFLOW. We have to provide a coarse mesh as basis for FEATFLOW's multigrid solvers and use a mesh which is locally refined near the edge of the step and near the floor of the expected recirculation zone, see Fig. 4.

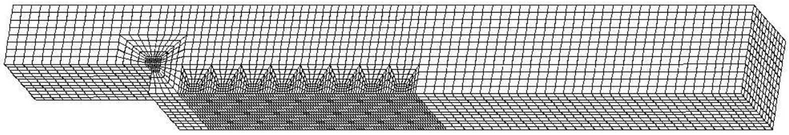

Figure 4 Mesh on multigrid level 3 (22848 elements)

The boundary conditions (9), actuator positions $\mathit{\Gamma}_{ctrl}$ and sensor positions $\mathit{\Omega}_{meas}$ are easily specified in a FEATFLOW data file. In each time step of the simulation, the MATLAB routine *MatlabController.m* receives from **PP3D-KE** the discrete velocity field $\tilde{\mathbf{v}}_x(t_n;x_i,y_j,z_k)$ with respect to the mesh nodes (x_i,y_j,z_k) lying in $\mathit{\Omega}_{meas}$. Corresponding to (8) we approximate the wall shear stress in $(x_i,0,z_k)$ by $\tilde{\tau}_w(t_n;x_i,z_k) = \tilde{\mathbf{v}}_x(t_n,x_i,y_j,z_k)/y_j$ and average for each x_i over all corresponding z_k. We then define $x_r(t_n)$ as a reasonable zero of the polynomial fitting of $\tilde{\tau}_w(x_i)$, see Fig. 5.

The implementation of open loop controls $u(t) = f(t)$ or closed loop controls $u(t) = f(t, x_r(s \le t))$ for given control laws f in the MATLAB function is straight forward by prescribing the corresponding values of $\tilde{\mathbf{v}}(t_n,x_i,y_j,z_k)$ in all mesh velocity nodes (x_i,y_j,z_k) lying on $\mathit{\Gamma}_{ctrl}$.

3.3 Numerical simulation and control results

First the uncontrolled flow is simulated with $Re_h = 30\,000$, which leads to a steady state solution. Here Re_h is the Reynolds number with respect to the height $h = 1$ of the step and the inflow velocity $|\mathbf{v}_\infty| \equiv 1$. As initial data, corresponding steady state solutions from calculations with lower Reynolds numbers are taken. Figures 6 and 7 show the $\mathbf{v}_x(x,z)$ and the $k(x,z)$ distributions in the $x-y-$cut-plane at $z = 1.5$.

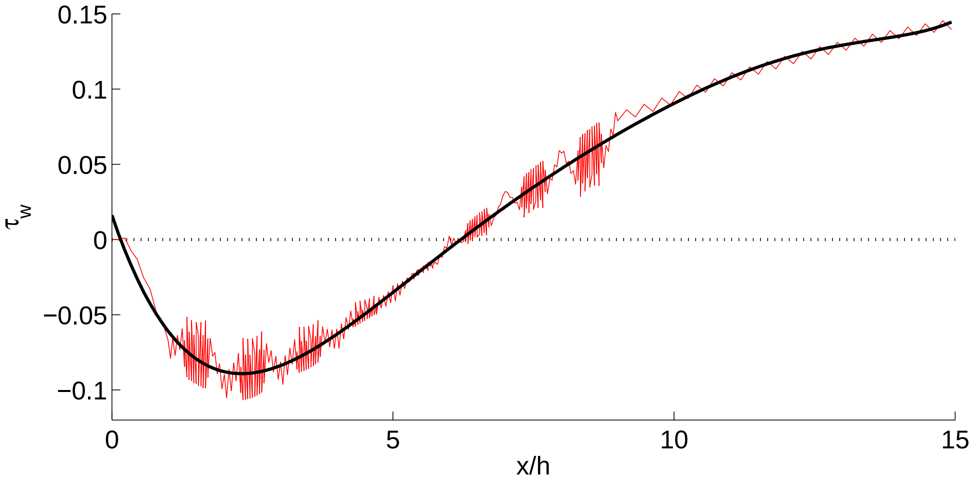

Figure 5 Approximated wall shear stress distribution $\tilde{\tau}_w(x)$ of the steady state and the fitting polynomial, with x denoting the distance to the step.

The typical recirculation zone can be observed, as well as the typical second vortex in the bottom corner of the step. The steady $\tilde{\tau}_w(x)$ distribution and the resulting reattachment length $x_r/h = 6.21$ are shown in Figure 5.

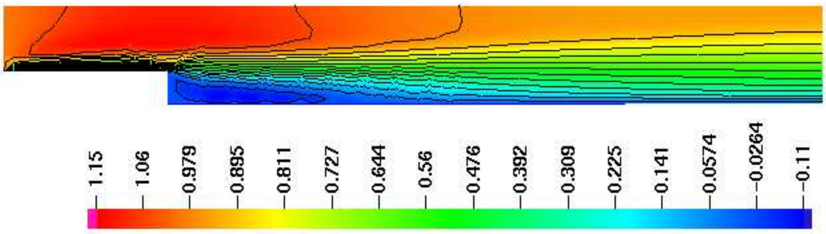

Figure 6 Downstream velocity component $\mathbf{v}_x$ of the steady state solution in x-y-cutplane at $z = 1.5$

However the typical shear layer instabilities with the characteristic frequency f_{shear} and thus a time-oscillation of the reattachment length cannot be observed. This is explained by the time-averaged character of the $k-\varepsilon$ turbulence model and by the fact, that the near wall zone at the step (where shear layer instabilities originate) is not calculated directly, but is modeled by logarithmic wall functions, which are derived from the boundary layer theory. The numerical results for the backward

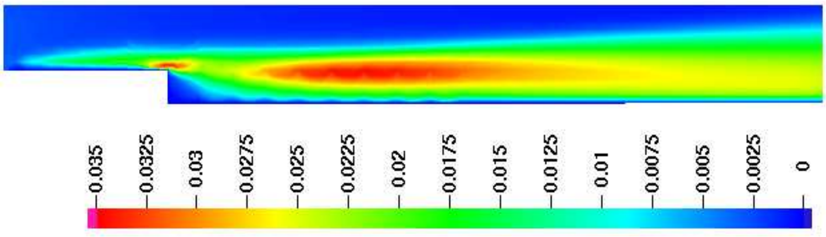

Figure 7 Turbulent kinetic energy k of the steady state solution in x-y-cutplane at $z = 1.5$

facing step and its comparison with others can be found in [8], [6] and [32]. We only observe an excessive turbulent kinetic energy in the shear layer zone.

Next the system response to a number of open loop controls is calculated, which provides sufficient information to identify black-box models of the system's input/output behavior and to design a robust closed-loop control for the reattachment length.

In fact, classical step experiments are performed. Thereby the actuation amplitude $u(t)$ is switched from zero to different levels a_0 to obtain different operating points of the system, see Fig. 8. A family of linear time-continuous models of 4th order is fitted to the measured data by application of subspace methods [33]. The corresponding bode responses of the identified transfer functions $G_i(s)$ are given in Fig. 9.

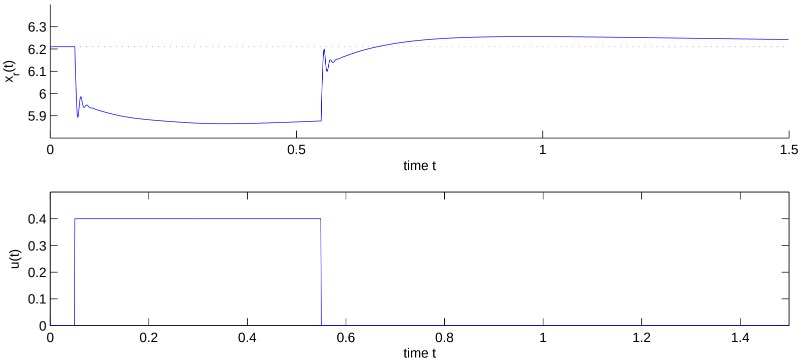

Figure 8 Open loop control and the corresponding system response

To synthesize a robust controller $C(s)$, a $\mathcal{H}_\infty$-controller design scheme is chosen. In $\mathcal{H}_\infty$-control, stability and/or performance of the 'worst' plant used to describe the process can be guaranteed. The goal is to find a good trade-off between the closed-loop sensitivity function $S(s)$, giving the performance, the restriction of the magnitude of the plant input signals, given by the transfer function $C(s)S(s)$, and robustness, given by the complementary sensitivity $T(s) = 1 - S(s)$. Here, the sensitivity transfer function is given by $S(s) = 1/(1 + C(s)G_N(s))$ with G_N being a nominal model showing minimal distance to all identified models over a certain frequency range. The trade-off is achieved by solving the mixed sensitivity problem, i.e. the closed-loop transfer functions are weighted with $W_T(s)$, $W_{CS}(s)$ and $W_S(s)$ depending on the frequency, and then the optimal controller $C(s)$ is obtained by minimizing the combined cost functional

$$\underset{\mathbf{C}}{min} \|N(C(s))\|_\infty, \quad N = \begin{bmatrix} W_T(s)\,T(s) \\ W_{CS}(s)\,C(s)S(s) \\ W_S(s)\,S(s) \end{bmatrix}.$$

For more details the reader is referred to standard textbooks, e.g. [34].

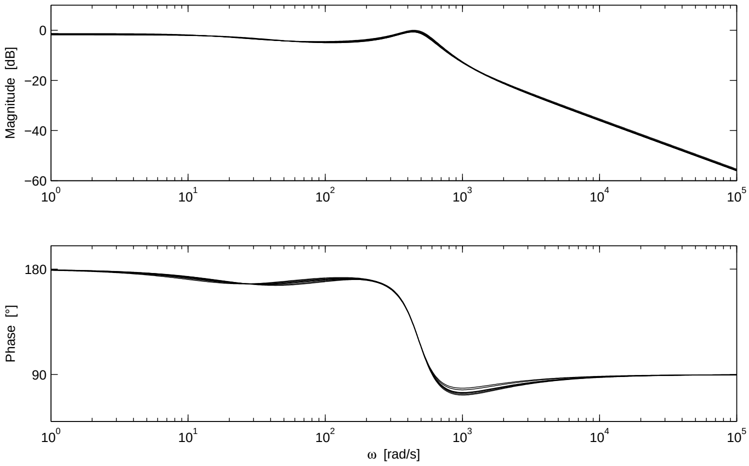

Figure 9 Bode responses of all identified models

The synthesized $\mathcal{H}_\infty$-controller is added in a classical control-loop as shown in Fig. 10. For the implementation into the FEATFLOW-MATLAB-interface the controller transfer function $C(s)$ is converted into a discrete state-space form.

The successful operation of the $\mathcal{H}_\infty$-controller is illustrated in Fig. 11, where a tracking of a reference trajectory $x_{ref}(t)$ by the reattachment length $x_r(t)$ is performed.

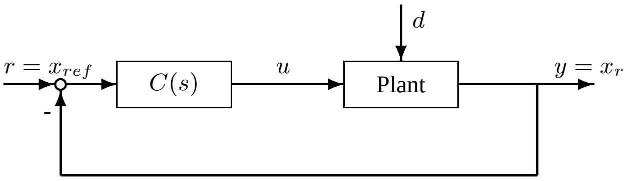

Figure 10 Control-loop with $\mathcal{H}_\infty$-controller $C(s)$ (x_{ref} - reference, u - manipulated variable, d - disturbances, $y = x_r$ - output)

4 Discussion

By using the newly developed **PP3D-KE** module from the CFD code FEATFLOW for the flow calculations, the simulation of nonstationary flows at high Reynolds numbers and in three-dimensional domains becomes feasible. The implementation of the flow and control configuration via easily manageable FEATFLOW data files and MATLAB m-files allows to easily test different actuator, sensor and control concepts and may even be a tool for the development of controllers. Only minimal insight into the CFD-code is required. In this sense, the presented FEATFLOW-MATLAB coupling can be considered as a general purpose tool for flow control purposes.

However, the performance of this tool is also subject to some substantial restrictions. Using the RANS approximation of the NSE with $k - \varepsilon$ turbulence model, some physically important flow phenomena may no longer be resolved. This is for instance often the case for flow configurations with detaching and separating flows, since the wall-layer models in the turbulence models are no longer valid. In our example of the backward facing step, the Kelvin-Helmholz shear layer instabilities cannot be observed in the numerical results. However, these instabilities may be the basis for very efficient control concepts, i.e. the control of the re-attachment length by enhancing the shear layer instabilities via a harmonic actuation with about the characteristic frequency of the instabilities, see [9]. Furthermore, the simulated flow is basically two-dimensional and the simulated transfer of inputs (insufflation amplitude) to outputs (reattachment length) is linear, though experiments and LES-simulations show distinct three-dimensional structures and a nonlinear input/output behavior. Here Large Eddy Simulations (LES) prove to be the better choice in order to numerically observe and simulate these physical phenomena [9].

The presented flow control environment based on FEATFLOW's k-ε model aims to present *one* tool to tackle nonstationary three-dimensional flow control problems. The example of the backward facing step clearly shows that the simultaneous use of different models and experiments for simulation, control design, test and validation purposes is recommended. Therefore the development of MATLAB-interfaces for the DNS and LES solvers in the FEATFLOW package is an important future task.

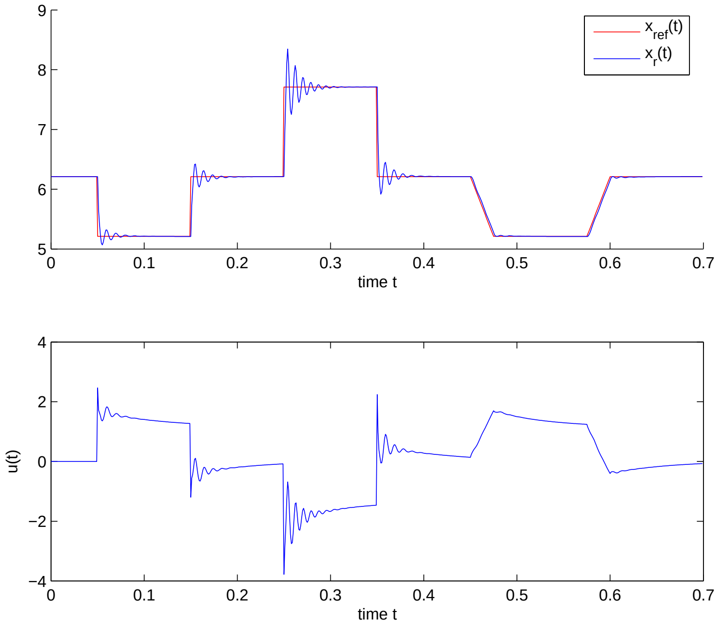

Figure 11 Reattachment length x_r following a reference curve by means of a robust controller

References

[1] The MathWorks, Inc., Cochituate Place, 24 Prime Park Way, Natick, Mass, 01760. MATLAB Version 7.0.4.352 (R14) , 2005.

[2] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel und A. Varga, ”SLICOT-A Subroutine Library in Systems and Control Theory”, Applied and Computational Control, Signals and Circuits. **1**, 1999, pp. 499-532.

[3] S. Turek et al.: ”FEATFLOW. Finite element software for the incompressible Navier-Stokes equations: User Manual, Release 1.2”, 1999. (www.featflow.de)

[4] M. Griebel, T. Dornseifer, T. Neunhoeffer: ”Numerical Simulation in Fluid Dynamics: A Practical Introduction”. SIAM Monographs on Mathematical Modeling and Computation, 1998.

[5] D. Kuzmin, S. Turek: ”Numerical simulation of turbulent bubbly flows”. 3rd International Symposium on Two-Phase Flow Modelling and Experimentation 2004. Editors: G.P. Celata, P.Di Marco, A. Mariani, R.K. Shah. Pisa, 2004.

[6] G. Medić, B. Mohammadi: ”NSIKE - an incompressible Navier-Stokes solver for unstructured meshes”. INRIA Research Report No 3644, 1999.

[7] S. Turek: ”Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach”. LNCSE 6, Springer, 1999.

[8] D. Kuzmin, R. Löhner and S. Turek (Eds.), ”Flux-Corrected Transport”. Springer, 2005.

[9] R. Becker, M. Garwon, C. Gutknecht, G. Bärwolff, R. King: ”Robust control of separated shear flows in simulation and experiment”. Journal of Process Control, **15**, 2005, pp. 691-700.

[10] B.F. Armaly, F. Durst, J.C.F. Pereira, B. Schonung: ”Experimental and theoretical investigation of backward-facing step flow”. Journal of Fluid Mechanics, **127**, 1983, pp. 473-496.

[11] K. Gartling: ”A Test Problem For Outflow Boundary Conditions - Flow Over a Backward Facing Step”. International Journal for Numerical Methods in Fluids, **11**, 1990, pp. 953-967.

[12] J. Kim, P. Moin: ”Application of a Fractional-Step Method to Incompressible Navier-Stokes Equation”. Journal of Computational Physics, **59**, 1985 pp. 308-314.

[13] H. Le, P. Moin, J. Kim: ”Direct numerical simulation of turbulent flow over a backward-facing step”. Journal of Fluid Mechanics, **330**, 1997, pp. 349-374.

[14] L. Kaiktsis, G. E. Karniadakis, S. A. Orszag: ”Onset of Three-Dimensionality, Equilibria, and Early Transition in Flow over a Backward facing Step”. Journal of Fluid Mechanics, **231**, 1991, pp. 501-528.

[15] T. Weinkauf, H.-C. Hege, B.R. Noack, M. Schlegel, A. Dillmann: ”Coherent Structures in a Transitional Flow around a Backward-Facing Step”. Physics of Fluids, **15(9)**, 2003.

[16] M. Garwon, R. King: ”A multivariable adaptive control strategy to regulate the separated flow behind a backward-facing step”. 16th IFAC World Congress, Prague, Czech Republic, July 2005.

[17] L. Henning, R. King: ”Multivariable closed-loop control of the reattachement length downstream of a backward-facing step”. 16th IFAC World Congress, Praha, Czech Republic, July 2005.

[18] J. Gerhard, M. Pastoor, R. King, B.R. Noack, A. Dillmann, M. Morzynski and G. Tadmor: ”Model-based control of vortex shedding using low-dimensional Galerkin models”. AIAA-Paper 2003-4262, 2003.

[19] O. Lehmann, D.M. Luchtenburg, B.R. Noack, R. King, M. Morzynski, G. Tadmor: ”Wake stabilization using POD Galerkin models with interpolated modes ”, accepted for publication.

[20] M. Pastoor, R. King, B.R. Noack, A. Dillmann and G. Tadmor: ”Model-based coherent-structure control of turbulent shear flows using low-dimensional vortex models”. AIAA-Paper 2003-4261, 2003.

[21] M. Hinze: ”Optimal and instantaneous control of the instationary Navier-Stokes equations”. Habilitation thesis, Fachbereich Mathematik, Technische Universität Berlin, 2000.

[22] R. Glowinski: ”Finite element methods for the numerical simulation of incompressible viscous flow; Introduction to the Control of the Navier-Stokes Equations”. Lectures in Applied Mathematics, **28**, 1991.

[23] M. D. Gunzburger, S.Manservisi: ”The velocity tracking problem for Navier-Stokes flows with boundary controls”. SIAM Journal Control and Optimization, **39**, 2000, pp. 594-634.

[24] M. Hinze, K. Kunisch: ”Control strategies for fluid flows - optimal versus suboptimal control”. In H.G. Bock et al., editor, ENUMATH 97, 1997, pp. 351-358.

[25] M. Hinze, K. Kunisch: ”Suboptimal Control Strategies for Backward Facing Step Flows”. Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, Editor A. Sydow, Vol. 3, 1997, pp. 53-58.

[26] H. Choi, M. Hinze, K. Kunisch: ”Instantaneous control of backward-facing step flows”. Applied Numerical Mathematics, **31**, 1999, pp. 133-158.

[27] T. R. Bewley, P. Moin, R. Temam: ”DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms”. Journal of Fluid Mechanics, **447**, 2001, pp. 179-225.

[28] K. Afanasiev, M. Hinze: ”Adaptive control of a wake flow using proper orthogonal decomposition”. In Shape Optimization & Optimal Design, Lecture Notes in Pure and Applied Mathematics 216. Marcel Dekker, 2001.

[29] M. Hinze, K. Kunisch: ”On suboptimal Control Strategies for the Navier-Stokes Equations”. ESAIM: Proceedings **4**, 1998, pp. 181-198.

[30] K. Ito, S. S. Ravindran: ”Reduced basis method for optimal control of unsteady viscous flow”. International Journal of Computational Fluid Dynamics, **15**, 2001, pp. 97-113.

[31] S. S. Ravindran: ”Control of flow separation over a forward-facing step by model reduction.” Computer Methods in Applied Mechanical Engineering. **191**, 2002, pp. 4599-4617.

[32] B. Mohammadi, O. Pironneau: ”Analysis of the k-epsilon turbulence model”. Wiley, 1994.

[33] L. Ljung: ”System identification - Theory for the user.” Prentice Hall PTR, 1999.

[34] S. Skogestad, I. Postlethwaite: ”Multivariable feedback control - Analysis and design”. 2 edn. John Wiley & Sons, Ltd, 2005.