ON THE VALIDATION OF NUMERICAL MODEL FOR SIMULATION OF TURBULENT FREE-SURFACE FLOWS

T. Bodnár[1] **and K. Kozel**[1] **and J. Příhoda**[2]
1) Department of Technical Mathematics, Faculty of Mechanical Engineering, Czech Technical University in Prague
2) Institute of Thermomechanics, Academy of Sciences of Czech Republic, Prague

Introduction

Numerical simulation of turbulent flows has undergone rapid evolution in the past decade. Mathematical models describing this kind of flows became more complex and versatile. Numerical methods solving these mathematical models are today more and more accurate and robust than ever before. The wide variety of available mathematical models and numerical methods allows for solution of wide range of problems. However in most cases it is not a-priori known which combination of mathematical model and numerical solver should be used in a specific case to get the best possible results. Therefore the future evolution of numerical simulation of complex turbulent flows is strictly dependent on validation of numerical codes against experimental data.

There are not many available experiments investigating turbulent free-surface flows. One of such experiments was performed by Chára and Hořejší and published in [3]. The aim of this paper is to present the comparison of results of numerical simulations with experimental data for this case.

Mathematical Model

The approach used in our model is based on the assumptions for variable-density incompressible flow. It means the flow is treated as if the domain is filled by only one fluid, which density is variable. The discontinuity in density profile arises at the free-surface. The key point in the modeling of this kind of flows is the appropriate formulation of mass conservation law, i.e. in the choice of adequate form of continuity equation. In this case the full "compressible" version should be used.

Mass conservation

$$\frac{\partial \rho}{\partial t} + \mathrm{div}\,(\rho \boldsymbol{v}) = 0 \tag{1}$$

Using the chain rule, this could be rewritten as:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{v} \cdot \mathrm{grad}\,\rho = -\rho\,\mathrm{div}\boldsymbol{v} \tag{2}$$

Because of the incompressibility assumption, the right-hand side of equation (2) should be equal zero. From this directly follows the expression on the left-hand side is then also equal zero. This means that the mass conservation in variable-density incompressible flows requires two separate conditions to be fullfiled:

a) *Density transport equation*

$$\frac{\partial \rho}{\partial t} + \boldsymbol{v} \cdot \operatorname{grad} \rho = 0 \tag{3}$$

b) *Divergence-free constrain*

$$\operatorname{div}\boldsymbol{v} = 0 \tag{4}$$

The equation (3) is used to control the time-evolution of liquid relative mass-fraction. So it could be simply rewritten as

$$\frac{\partial r}{\partial t} + \boldsymbol{v} \cdot \operatorname{grad} r = 0 \tag{5}$$

Here $r = r_L$ for liquid and $r = r_G$ for gas. Because of the linearity of equation (5) we have some freedom in choice of values r_L and r_G. The interface between gas and liquid can thus be found as an isosurface of $r = r_S$, where $r_S = (r_L + r_G)/2$. Using the computed value of r the local volume fraction of liquid could be established as $c_L = (r - r_G)/(r_L - r_G)$ which is in the range from 0 to 1. Using this value the appropriate local material properties (density and viscosity) could be obtained by simple interpolation between the corresponding values of gas and liquid. E.g. for local density the following relation holds:

$$\rho = c_L \rho_L + (1 - c_L)\rho_G \tag{6}$$

This method can be seen as an elementary implementation of VOF method introduced in [7].

To enforce the divergence-free constrain and calculate pressure, the equation (4) is modified by adding the time-derivative of pressure properly scaled by the artificial speed of sound $\mathfrak{c}$:

$$\frac{1}{\mathfrak{c}^2}\frac{\partial p}{\partial t} + \operatorname{div}\boldsymbol{v} = 0 \tag{7}$$

This model is suitable for computation of steady flow, while the time-marching technique is used to get stationary solution. In such a case the non-physical additional term in the equation (7) disappears and the true solution of the steady problem is recovered.

Reynolds averaged Navier-Stokes equations

The conservation of Reynolds-averaged momentum components results in the following set of equations:

$$\rho\frac{\partial v_i}{\partial t} + \rho v_j\frac{\partial v_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\left(\mu + \mu_T\right)\frac{\partial v_i}{\partial x_j}\right] + \rho g_i \quad i = 1,2,3 \tag{8}$$

Here μ and μ_T denote the laminar and turbulent viscosities, while g_i stands for gravity acceleration. The components of velocity vector v_i could be computed directly from these equations. The pressure p and density ρ are updated from equations (7) and (5).

Turbulence Model

The VOF method allows for very straightforward implementation of turbulence model. In this case the SST k-ω model according to [6] was chosen. The full description of this model exceeds the scope of this paper and thus only the governing equations for the the turbulent kinetic energy k and for ω are mentioned here.

$$\rho\frac{\partial k}{\partial t} + \rho v_j \frac{\partial k}{\partial x_j} = P - \beta^* \rho k \omega + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_T}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \tag{9}$$

$$\rho\frac{\partial \omega}{\partial t} + \rho v_j \frac{\partial \omega}{\partial x_j} = \frac{\gamma\rho}{\mu_T}P - F_4\beta\rho\omega^2 + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_T}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_j}\right] + 2\rho\frac{1-F_1}{\sigma_{\omega 2}\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \tag{10}$$

The details of the model could be found in [6] and in [2].

Numerical Solution

Numerical solution of the above presented mathematical model is based on finite-volume cell-centered semi-discretization on structured mesh. The time-integration of the resulting system of ordinary differential equations is carried out using explicit Runge-Kutta multistage scheme. Because of the use of the central-differencing in spatial discretization, suitable stabilization technique is used to avoid non-physical oscillations in the solution.

Space Discretization

The computational mesh is structured, consisting of hexahedral primary control volumes. To evaluate the viscous fluxes also dual finite volumes are needed. These have octahedral shape and are centered around the corresponding primary cell faces. See the following figure 1 for the schematic view of such configuration.
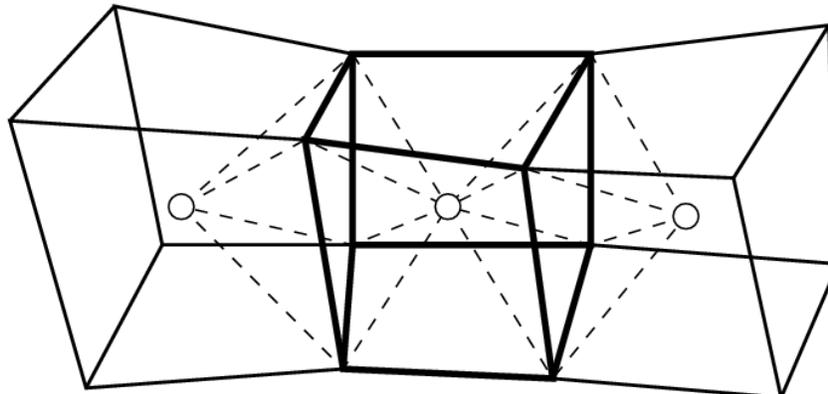

Figure 1: Finite-volume grid in 3D

The system of RANS equations (including the modified continuity equation) could be rewritten in the vector form. Here we use W to denote the vector of unknowns (including pressure). Vectors F, G and H denote the inviscid fluxes in x,y,z directions, while R, S and T stand for their viscous counterparts. Using this notation, the spatial finite-volume semi-discretization could be written in the following form:

$$\frac{\partial \mathsf{W}_{ijk}}{\partial t} = -\frac{1}{|D|} \oint_{\partial D} \left[(\mathsf{F} - \mathsf{R}), (\mathsf{G} - \mathsf{S}), (\mathsf{H} - \mathsf{T}) \right] \cdot \hat{\boldsymbol{\nu}} \, dS + \frac{1}{|D|} \int_D \mathsf{f}_\mathsf{w} \tag{11}$$

Here D denotes the computational cell, $\hat{\boldsymbol{\nu}}$ is the outer unit normal vector of the cell boundary, dS is the surface element of this boundary. The vector f_w contains the external body forces (e.g. gravity in our case). The equation (11) can be rewritten in operator form:

$$\frac{\partial \mathsf{W}_{ijk}}{\partial t} = -\mathcal{L}\, \mathsf{W}_{i,j,k} \tag{12}$$

Here $\mathcal{L}$ stands for the finite-volume discretization operator. This operator is still exact at this stage and it should be properly discretized to allow for numerical solution. This is done by the replacement of fluxes in it's formulation by their numerical (approximate) versions.

The inviscid flux integral can be approximated in a central manner, e.g. the value of the flux F on the cell face with index $\ell = 1$ is computed as average of cell-centered values from both sides of this face:
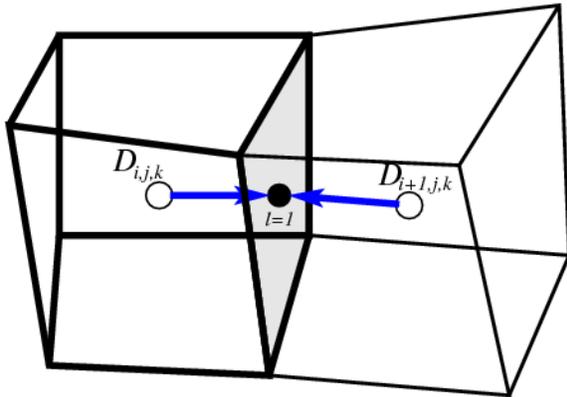

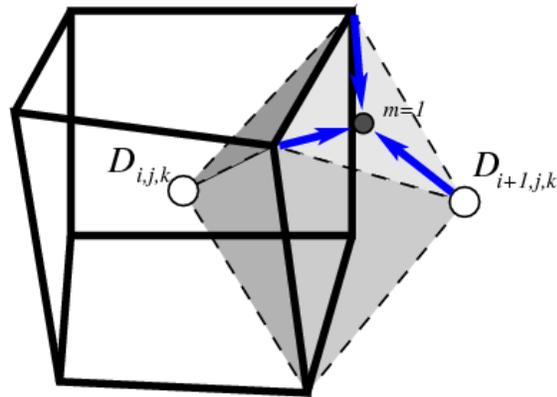Figure 2: Inviscid flux discretization   Figure 3: Viscous flux discretization

$$\mathsf{F}_1^n = \frac{1}{2}[\mathsf{F}(\mathsf{W}_{i,j,k}^n) + \mathsf{F}(\mathsf{W}_{i+1,j,k}^n)] \tag{13}$$

The contribution of inviscid fluxes is finally summed up over the cell faces $\ell = 1, \ldots, 6$. In this way we can write down the inviscid flux approximation:

$$\oint_{\partial D} \mathsf{F} \nu^x \, dy \, dz \approx \sum_{\ell=1}^{6} \mathsf{F}_\ell \nu_\ell^x S_\ell \tag{14}$$

The discretization of viscous fluxes is a little bit more complicated because the vectors $\mathsf{R}, \mathsf{S}, \mathsf{T}$ were defined using the derivatives of velocity components. So we need to approximate somehow these derivatives at cell faces. This can be done using the dual finite-volume grid that is centered around the corresponding faces (see Figure 1 and 3).

The evaluation of velocity gradient components is then replaced by the surface integral over the dual volume boundary. Finally this surface integral is approximated by a discrete sum over the dual cell faces (with indices $m = 1, \ldots, 8$). For example trying to evaluate the first component of the viscous flux R_1 (i.e. approximate u_x) at the cell face $l = 1$ we must proceed in the following way:

$$u_x \approx \oint_{\partial \tilde{D}} u \, \nu^x \, dy \, dz \approx \sum_{m=1}^{8} u_m \nu_m^x S_m \tag{15}$$

The outer normal of the dual cell faces should be properly approximated $\nu^x \approx \nu_m^x$. The values of velocity components in the middle nodes of these faces are taken as an average of the values in the corresponding vertices.

Time Integration

The problem is now in the semi-discrete form:

$$\frac{d\mathsf{W}_{ijk}}{dt} = -\tilde{\mathcal{L}} \, \mathsf{W}_{i,j,k} \tag{16}$$

This system of ordinary differential equations can be solved e.g. by the Runge-Kutta multistage method:

$$
\begin{aligned}
\mathsf{W}_{i,j,k}^{(0)} &= \mathsf{W}_{i,j,k}^n \\
\mathsf{W}_{i,j,k}^{(r+1)} &= \mathsf{W}_{i,j,k}^{(0)} - \alpha_{(r)} \Delta t \tilde{\mathcal{L}} \mathsf{W}_{i,j,k}^{(r)} \qquad r = 1, \ldots, m \\
\mathsf{W}_{i,j,k}^{n+1} &= \mathsf{W}_{i,j,k}^{(m)}
\end{aligned}
\tag{17}
$$

The three-stage explicit RK scheme has coefficients:
$\alpha_{(1)} = 1/2, \ \alpha_{(2)} = 1/2, \ \alpha_{(3)} = 1$. This scheme is second order only, however it provides extended stability region leading to CFL=2 which improves the overall efficiency of the whole scheme.

The efficiency and robustness of the method could further be increased by modification of the above algorithm. The modification used for simulations presented in this paper follows the Runge-Kutta time integration procedures outlined in [9] and further refined in [8]. The idea behind this modified approach lies in splitting of the space discretization operator into inviscid and viscous part. The inviscid operator is evaluated at each Runge-Kutta stage, while the viscous operator is evaluated just in few stages. This corresponds to the use of different Runge-Kutta coefficients for time integration of inviscid and viscous fluxes. The modified algorithm could thus be written in the following form:

$$
\begin{aligned}
\mathsf{W}_{i,j,k}^{(0)} &= \mathsf{W}_{i,j,k}^n \\
\mathsf{W}_{i,j,k}^{(r+1)} &= \mathsf{W}_{i,j,k}^{(0)} - \alpha_{(r)} \Delta t \left(\mathcal{Q}^{(r)} + \mathcal{D}^{(r)} \right) \\
\mathsf{W}_{i,j,k}^{n+1} &= \mathsf{W}_{i,j,k}^{(m)}
\end{aligned}
\tag{18}
$$

Here the space discretization operator at stage (r) is split as follows:

$$\mathcal{L}\mathsf{W}_{i,j,k}^{(r)} = \mathcal{Q}^{(r)} + \mathcal{D}^{(r)} \tag{19}$$

The inviscid flux $\mathcal{Q}$ is evaluated in usual way at each stage

$$\mathcal{Q}^{(r)} = \mathcal{Q}\mathsf{W}_{i,j,k}^{(r)} \qquad \text{with} \qquad \mathcal{Q}^{(0)} = \mathcal{Q}\mathsf{W}_{i,j,k}^{n} \tag{20}$$

The viscous flux $\mathcal{D}$ uses a blended value of from the previous stage and the actual stage according to the following rule:

$$\mathcal{D}^{(r)} = \beta_{(r)}\mathcal{D}\mathsf{W}_{i,j,k}^{(r)} + (1 - \beta_{(r)})\mathcal{D}^{(r-1)} \qquad \text{with} \qquad \mathcal{D}^{(0)} = \mathcal{D}\mathsf{W}_{i,j,k}^{n} \tag{21}$$

The coefficients $\alpha_{(r)}$ and $\beta_{(r)}$ are chosen to guarantee large enough stability region for the Runge-Kutta method. The following set of coefficients was used for this study:

$$\begin{array}{ll} \alpha_{(1)} = 1/3 & \beta_{(1)} = 1 \\ \alpha_{(1)} = 4/15 & \beta_{(1)} = 1/2 \\ \alpha_{(1)} = 5/9 & \beta_{(1)} = 0 \\ \alpha_{(4)} = 1 & \beta_{(4)} = 0 \end{array}$$

It is easy to see that for this four-stage method only two evaluations of dissipative terms are needed which saves significant amount of calculations while retaining the advantage of large stability region of the method. Further admissible sets of coefficients together with comment on the increase of the efficiency and robustness could be found in the original papers [9] and [8] and the references therein.

Numerical Stabilization

It is a well known property of central schemes, that in the presence of strong gradients they produce non-physical oscillations in the solution. There are many ways to avoid this phenomena. The method used here is based on pressure stabilization. This approach is long time used in finite-element community and has been used in finite-volume framework in *Vierendeels, Riemslagh, & Dick* [12]. The main principle of this method is to add a pressure dissipation term (Laplacian) into the modified continuity equation. This helps to prevent oscillations in pressure, which stabilizes the whole numerical method.

The pressure stabilization is introduced by the additional source term on the right-hand side of modified continuity equation. This term has the following form:

$$Q_{i,j,k} = \frac{1}{|D_{i,j,k}|} \sum_{\ell=1}^{2N} \frac{p_\ell - p_{i,j,k}}{b_\ell} S_\ell \tag{22}$$

Here ℓ denotes the control volume cell face index, p_ℓ is the pressure in the corresponding neighboring cell and S_ℓ is the cell face area. The value b_ℓ has the dimension of velocity and represents the maximal convective velocity in the domain and local diffusive velocity.

$$b_\ell = \max(\sqrt{v_1^2 + v_2^2 + v_3^2}) + \frac{2\nu}{L_\ell} \tag{23}$$

Symbol L_ℓ corresponds to a distance between the actual and neighboring cell centers.

On an uniform cartesian mesh with cells of size δx this term gives:

$$Q = \frac{\delta x}{2b}\Delta p \tag{24}$$

This type of numerical stabilization has some advantages over the classical artificial diffusion applied to the velocity components. First, its "artificial" effects are clearly separated from the physical viscosity included in RANS equations.

The second method applied to remove non-physical oscillation from the numerical solution uses a conservative *non-linear filter*. This technique was described in detail in [4] and [11]. The algorithm is based on idea of removing high frequency oscillations from the numerical solution. The oscillations are localized, measured and conservatively redistributed to neighbouring grid cells. On a structured grid this post-processing could be done quite easily in each grid direction.

The main advantage of this combined approach is that it allows for parameter fine-tuning and thus it is possible to apply just the minimal amount of stabilization without excessively influencing the solution.

Numerical simulations

Numerical tests were performed for segment of a 2D channel with two ribs of square cross-section. Channel is partially filled by water, while the remaining volume is occupied by air. The geometrical configuration can be seen in the Figure 4. The dimensions used in our simulations were: $b = 30\,mm$, $H = 125\,mm$, $L = 120\,mm$.
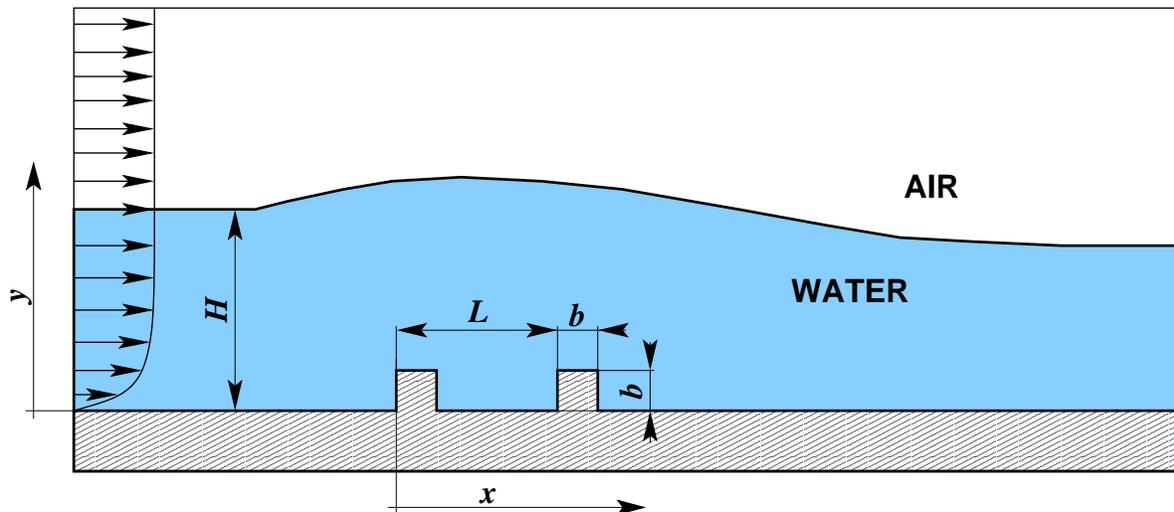

Figure 4: Computational geometry

The test calculations were performed on structured, cartesian, wall-fitted grid. Besides of the mesh refinement in the wall-normal direction some cell stretching is also applied in the horizontal direction (see Fig. 5 sketch of the grid).
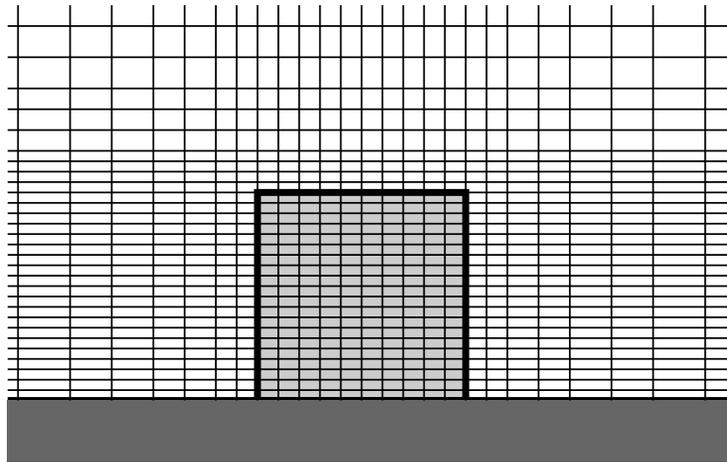
Figure 5: Cartesian grid structure in the proximity of the rib.

The ribs placed at the channel bottom were simulated using immersed boundary approach. This means the computational domain, including ribs, is divided into control volumes and is being treated as if it would contain fluid. The flow parameters are then computed in the whole domain according to the discretized governing equations, while specific values ($v = 0$, $k = 0$) are kept set in the grid cells located inside of the ribs. This approach simplifies significantly the computation in the case of multiple obstacles. The mesh structure detail can be seen in the Figure 5.

The boundary conditions were chosen to match as close as possible the experimental data from [3]. The power-law velocity profile was prescribed at the inlet. This velocity profile had an exponent 2/9 and characteristic velocity $0.35\ m/s$ at the distance $110\ mm$ from the bottom.

At the outlet the homogeneous Neumann conditions were used for all quantities to simulate the fully developed flow. For the density a piecewise constant profile was prescribed at the inlet with the jump across the water/air interface located at height $H = 110\ mm$ above the channel bottom. For the pressure a fixed value was prescribed at the top of the channel, while homogeneous Neumann conditions were used at the other parts of the boundary.

Numerical results

The first results of numerical simulations for the above described test case are presented in this section. The global view of the flow structure in the vicinity of ribs can be seen in the figure 6.
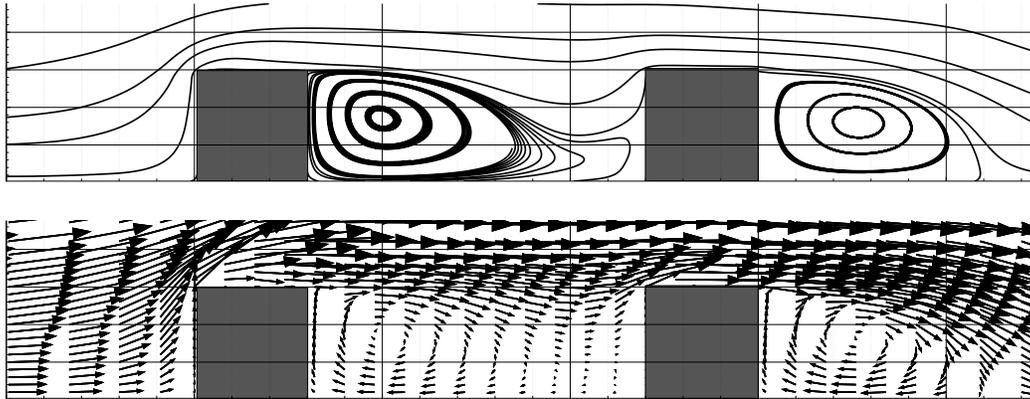
Figure 6: Flow structure in the proximity of ribs.

The recirculation zones behind the ribs (also observed in the experiments) are well captured. More detailed information about the agreement between numerical results and experimental data could be obtained from the figure 7 where the velocity profiles are compared in selected cross-sections.

Conclusions, remarks

The numerical results presented here were chosen to show the resolution of the flow in the proximity of the ribs. The numerical simulations have demonstrated the applicability of the mathematical model and numerical solver for selected problem. The results seem to be in a good agreement with experimental data. The test calculations haveshown relatively slow convergence of the method for this test case. This is mainly because of the slow evolution of the recirculation zones from the initial flow field. The moderate velocity (i.e. Reynolds number) in this case could be responsible for this. Such kind of behavior wasn't observed for higher inlet velocities.

This problem could be possibly solved by more appropriate choice of artificial compressibility parameter. Further model sensitivity test with respect to this parameter will be needed to increase the efficiency of the numerical solver. More test calculations will be needed to test the model sensitivity with respect to grid spacing which is of crucial importance especially in the case of immersed boundary approach used in this work.

More detailed comparison with experimental data for the above described test case as well as with other available experimental data sets is under preparation.
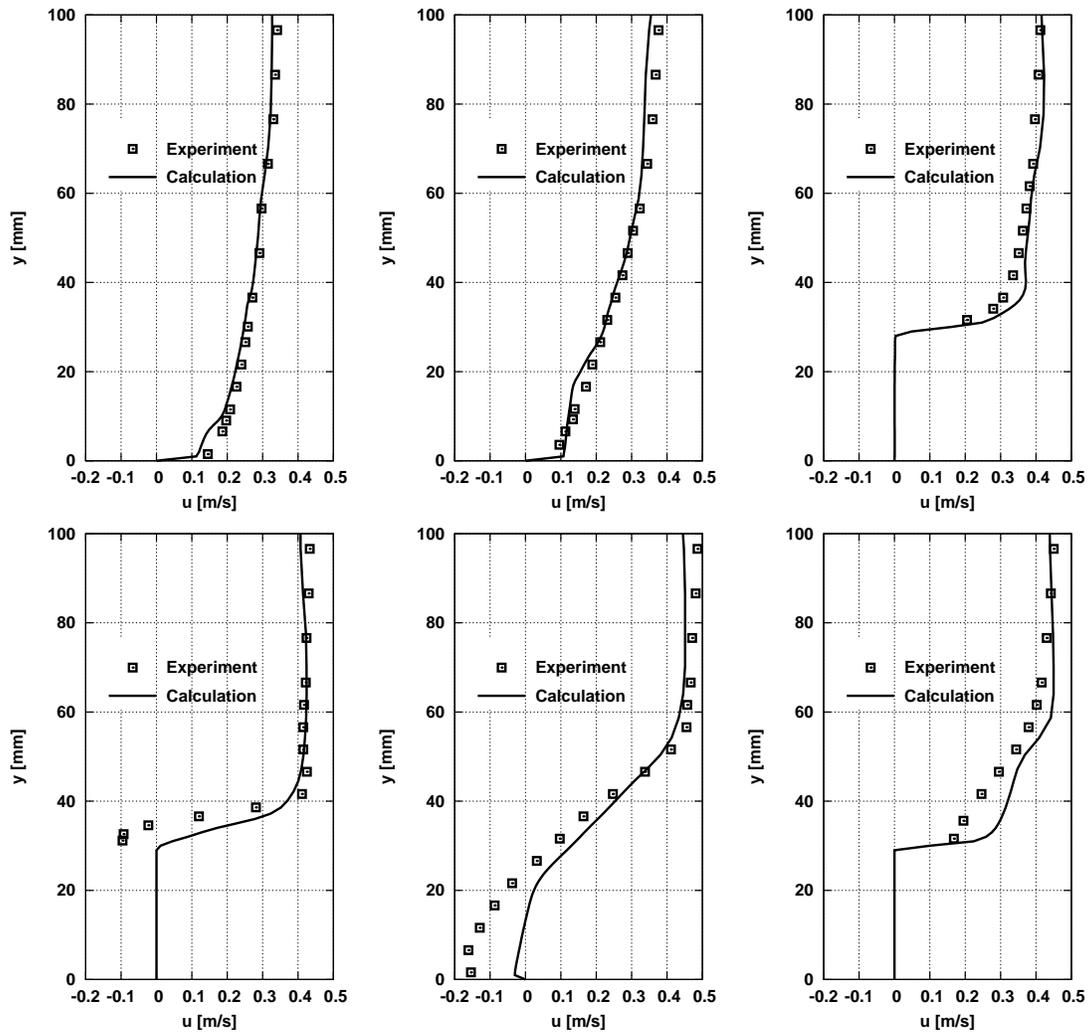
Figure 7: Comparison of calculated and measured horizontal velocity profiles in cutplanes located at $x = -100, -36, 0, 15, 75, 150\,mm$.

References

[1] BENEŠ, L. & KOZEL, K.: Numerical Simulation of 3D Atmospheric Boundary Layer Flow with Pollution Dispersion. In: *Proceedings of International Conference on Advanced Engineering Design*, (pp. 141–146). University of Glasgow, Glasgow (2001). ISBN 0-85261-731-3.

[2] BODNÁR, T. & PŘÍHODA, J.: Numerical simulation of turbulent free-surface flow in curved channel. *Journal of Flow, Turbulence and Combustion*, vol. 76, no. 4: (2006) pp. 429–442.

[3] CHÁRA, Z.: UVP a LDA měření rychlostních polí v okolí překážek umístěných na dně kanálu. In: *XIV. Medzinárodn vedecká konferencia Aplikácia experimentálních a numerických metód v mechanike tekutín, Rájecké Teplice*, (pp. 145–150) (2004).

[4] ENGQUIST, B., LÖTSTEDT, P., & SJÖGREEN, B.: Nonlinear Filters for Efficient Shock Computation. *Mathematics of Computation*, vol. 52, no. 186: (1989) pp. 509–537.

[5] FERZIGER, J. H. & PERIĆ, M.: *Computational Methods for Fluid Dynamics*. Springer Verlag (1997).

[6] HELSTEN, A.: Some improvements in Menter's $k - \omega$ SST turbulence model. Tech. rep., American Institute of Aeronautics and Astronautics (1998). AIAA-98-2554.

[7] HIRT, C. W. & NICHOLLS, B.: Volume of fluid (VOF) method for dynamics of free boundaries. *Journal of Computational Physics*, vol. 39: (1981) pp. 201–221.

[8] JAMESON, A.: Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings. In: *AIAA 10th Computational Fluid Dynamics Conference, Honolulu* (1991). AIAA Paper 91-1596.

[9] JAMESON, A., SCHMIDT, W., & TURKEL, E.: Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. In: *AIAA 14th Fluid and Plasma Dynamic Conference, Palo Alto* (1981). AIAA paper 81-1259.

[10] RANSAU, S. R.: Solution Methods for Incompressible Viscous Free Surface Flows: A Literature Review. Preprint 3/2002, Norwegian University of Science and Technology, Department of Mathematical Sciences, Norvegian University of Science and Technology, N-7491 Trondheim, Norway (2002).

[11] SHYY, W., CHEN, M.-H., MITTAL, R., & UDAYKUMAR, H. S.: On the Suppression of Numerical Oscillations Using a Non-Linear Filter. *Journal of Computational Physics*, vol. 102: (1992) pp. 49–62.

[12] VIERENDEELS, J., RIEMSLAGH, K., & DICK, E.: A Multigrid Semi-implicit Line-Method for Viscous Incompressible and Low-Mach-Number Flows on High Aspect Ratio Grids. *Journal of Computational Physics*, vol. 154: (1999) pp. 310–344.