Performance of Reynolds Averaged Navier-Stokes Models in Predicting Separated Flows: Study of the Hump Flow Model Problem

Daniele Cappelli *

Ecole Nationale Superieure de l'Aronautique et de l'Espace, Toulouse, France

Nagi N. Mansour †

NASA Ames Research Center, Moffett-Field, CA

Separation can be seen in most aerodynamic flows, but accurate prediction of separated flows is still a challenging problem for computational fluid dynamics (CFD) tools. The behavior of several Reynolds Averaged Navier-Stokes (RANS) models in predicting the separated flow over a wall-mounted hump is studied. The strengths and weaknesses of the most popular RANS models (*Spalart-Allmaras*, *k*-ϵ, *k*-ω, *k*-ω-*SST*) are evaluated using the open source software OpenFOAM. The hump flow modeled in this work has been documented in the 2004 CFD Validation Workshop on Synthetic Jets and Turbulent Separation Control. Only the baseline case is treated; the slot flow control cases are not considered in this paper. Particular attention is given to predicting the size of the recirculation bubble, the position of the reattachment point, and the velocity profiles downstream of the hump.

Nomenclature

c	Chord of the hump
C_1	Coefficient of the production term of ϵ for the $k-\epsilon$ model
C_2	Coefficient of the dissipation term of ϵ for the $k-\epsilon$ model
C_f	Skin friction coefficient
C_μ	Coefficient of eddy viscosity for the $k-\epsilon$ model
k	Turbulent kinetic energy
M	Mach number
Re_c	Reynolds number based on the chord of the hump
S	Outward-pointing face area vector
S_{ij}	Rate-of-strain tensor
t	Time
u	Scalar component of velocity
$\mathbf{u}$	Velocity vector
U_∞	Asymptotic velocity
$\mathbf{x}$	Space vector
α	Coefficient of the production term of ω for the $k-\omega$ model
α_ω	Coefficient of the viscous term of the ω equation
β	Coefficient of the dissipation term of ω in the $k-\omega$ model
ϵ	Rate of dissipation of turbulent kinetic energy
ω	Specific rate of dissipation of turbulent kinetic energy
σ_ϵ	Coefficient of the viscous term in the ϵ equation
ρ	Density
ν	Kinematic viscosity

*Student intern, STIEP, NASA Ames Research Center; Double Master degrees student, SUPAERO - Politecnico di Milano

†Chief Division Scientist, NASA Advanced Supercomputing Division, AIAA Associate Fellow

ν_t Turbulent kinematic viscosity
ν_{eff} Effective viscosity $(\nu + \nu_t)$

I. Introduction

A sketch of the experimental geometry for the "hump" flow is shown in figure 1.[3] Numerical simulations of this hump model were the object of study during the CFD Validation Workshop held in Williamsburg, Virginia in 2004. The goal of the workshop, fully described in Rumsey et al.'s paper[1] was to bring together an international group of computational fluid dynamics practitioners to assess the current capabilities of different classes of flow solution methodologies to predict flow fields induced by synthetic jets and separation control geometries.

For the base-line case (no flow control), most CFD results missed the pressure levels over the hump between $0.2 < x/c < 0.6$, and also predicted higher pressures in the separated region upstream of $x/c = 1$. This trend was in part due to the blockage of the side plates used in the experiments which caused a decrease of pressure, as compared to 2D simulations.

Rumsey et al.'s analysis[1] concluded that the most significant results concerned the separated flow downstream of the model: the separation location was predicted reasonably well by most CFD methods, while the reattachment location was over-predicted compared to the experimental location ($x/c = 1.1$). A possible reason for the reattachment point being over-predicted was that most of the methods under-predicted the magnitude of the turbulent shear stress in the separated region.

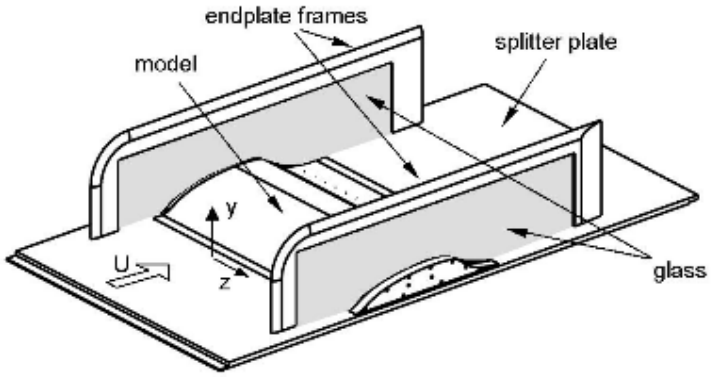

Figure 1. Isometric view showing the model mounted on the splitter plate with end plates in place.[3]

In 2008 an updated survey was conducted by Rumsey.[2] The conclusion of that study was that no major progress was made since the time of the first workshop in terms of RANS/URANS capabilities. Results were still under-predicting the eddy viscosity in the separated region and over-predicting the size of the reattachment bubble. He et al.[4] obtained reasonably good results by using the commercial software *Fluent* with a second order upwind scheme and the SIMPLE algorithm for pressure-velocity coupling. In this case, the k-ϵ model agreed well with the experimental reattachment location, but still under-predicted the magnitude of the turbulent shear-stress. These results brought into question why the same software yielded different results from Brettini and Cravero.[5] The answer may be that this flow is strongly sensitive to the resolution of the mesh, the position of the inlet and to the particular boundary conditions employed. Our work focuses on these aspects and discusses how these settings affect the solution.

After a brief introduction to the turbulence models and the SIMPLE algorithm implemented in *Open-*

FOAM,[a] we show our results for the baseline case[b] by using 4 different turbulence models (*Spalart-Allmaras*, *k*-ϵ, *k*-ω, *k*-ω-*SST*) and analyze the weaknesses and strengths of each RANS model. The effects that the position of the inlet, the boundary condition on the upper wall, and the resolution of the mesh have on the hump-flow will be discussed as well.

II. RANS methods in *OpenFOAM*

The complete implementation of the algorithm can be found in the source code of the *simpleFoam* solver provided with **OpenFOAM** (directory *simpleFoam*)[c]. In the appendix, we report some lines of code where the momentum equations and the equations for k, ϵ and ω are implemented. The equation for momentum, which is defined in the file *UEqn.H* , is:

$$\bar{u}_j \frac{\partial \bar{u}_i}{x_j} - \frac{\partial}{\partial x_j}\left[\nu_{eff}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)\right] = -\frac{\partial p}{\partial x_i} \tag{1}$$

The RANS equation for turbulent kinetic energy, k, is:

$$\frac{\partial k}{\partial t} + \bar{u}_j \frac{\partial k}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu_{eff}\right)\frac{\partial k}{\partial x_j}\right] = \nu_T \frac{\partial \bar{u}_i}{\partial x_j}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \epsilon \tag{2}$$

While the equation for dissipation, ϵ, is:

$$\frac{\partial \epsilon}{\partial t} + \bar{u}_j \frac{\partial \epsilon}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_T}{\sigma\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right] = C_1 \frac{\epsilon}{k}\nu_T \frac{\partial \bar{u}_i}{\partial x_j}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - C_2 \frac{\epsilon^2}{k} \tag{3}$$

By using the following definition of turbulent viscosity, $\nu_t = C_\mu \frac{k^2}{\epsilon}$, we have the $k-\epsilon$ model. The first transported variable is k, the second one is ϵ.
By defining the specific dissipation $\omega = \frac{\epsilon}{k}$, as second transported variable, we have the $k-\omega$ model. The equation used for k is the same implemented for the $k-\epsilon$ model, while the equation for ω becomes:

$$\frac{\partial \omega}{\partial t} + \bar{u}_j \frac{\partial \omega}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu + \alpha_\omega \nu_T\right)\frac{\partial \omega}{\partial x_j}\right] = \alpha \frac{\omega}{k}\nu_T \frac{\partial \bar{u}_i}{\partial x_j}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \beta\omega^2 \tag{4}$$

The eddy viscosity is then defined as $\nu_t = \frac{k}{\omega}$

Because of their complexity, we will not describe the equations for the models $Spalart - Allmaras$ and $k-\omega-SST$. Their implementation is available in the same directory where $k-\epsilon$ and $k-\omega$ are implemented[d]. For further information about *OpenFOAM* we reference the reader to the manual available on the Internet.[8] In table 1 we list the coefficients used in *OpenFOAM* for the $k-\epsilon$ and $k-\omega$ equations.

$k-\epsilon$		$k-\omega$	
σ_ϵ	1.3	α_ω	0.5
$C1$	1.44	α	0.52
$C2$	1.92	β	0.072
C_μ	0.09	C_μ	0.09

Table 1. Coefficients for the k-ϵ and $k-\omega$ models.

III. SIMPLE Algorithm as implemented in OpenFOAM

SimpleFOAM is a steady-state solver for incompressible, turbulent flow. We recall that the Navier-Stokes equations for a single-phase flow with a constant density and viscosity are the following:

$$\nabla \cdot \mathbf{u} = 0 \tag{5}$$

[a]The Open-FOAM version used is the 2.1.1
[b]The baseline does not include the plenum, since the problem of flow-control is not dealt with here.
[c]Directory:*openfoam211/applications/solvers/incompressible/simpleFoam*
[d]$openfoam211/src/turbulenceModels/RAS$

$$\nabla \cdot (\mathbf{uu}) - \nabla \cdot (\nu \nabla \mathbf{u}) = -\frac{1}{\rho}\nabla p \tag{6}$$

The solution of these equations is not straightforward because of the non-linear term $\nabla \cdot (\mathbf{uu})$ and because an explicit equation for the pressure is not available. The approach used in *OpenFOAM* is to derive an equation for the pressure by taking the divergence of the momentum equation and substituting it in the continuity equation. Temporal discretization is performed using some implicit temporal scheme, such as:[6]

$$\int_t^{t+\Delta t} f(t, \mathbf{U}(\mathbf{x}, t))dt = (1 - C)\Delta t f(t, \mathbf{U}(\mathbf{x}, t^0)) + C\Delta t f(t, \mathbf{U}(\mathbf{x}, t^n)) \tag{7}$$

where different values of C can recover temporal schemes defined in Juretic's thesis.[6] When the momentum equation is approximated by using the equation 7, the following relationship is derived:[6,7]

$$a_p \mathbf{u}_p = \mathbf{H(u)} - \nabla p \tag{8}$$

where the subscript P refers tot he center of cell P. Velocity can be rewritten as follows:

$$\mathbf{u}_p = \frac{\mathbf{H(u)}}{a_P} - \frac{\nabla p}{a_P} \tag{9}$$

The term $\mathbf{H(u)}$ includes all terms apart from the pressure gradient at the new time step and the diagonal term $a_P^n \mathbf{U}_P^n$, where the subscript n represents the new time level. The continuity equation is discretized as:

$$\nabla \cdot \mathbf{u} = \sum_f \mathbf{S}\mathbf{u}_f = 0 \tag{10}$$

where $\mathbf{S}$ is outward-pointing face area vector and $\mathbf{u}_f$ is the velocity on the face, which is obtained by interpolating the semi-discretized form of the momentum equation (9) as follows:

$$\mathbf{u}_f = \left(\frac{\mathbf{H(u)}}{a_P}\right)_f - \left(\frac{\nabla p}{a_P}\right)_f \tag{11}$$

By substituting this equation into the discretized continuity equation above, the pressure equation is derived:

$$\nabla \cdot \left(\frac{1}{a_P}\nabla p\right) = \nabla \cdot \left(\frac{\mathbf{H(u)}}{a_P}\right) \tag{12}$$

The mass flux through a cell face can be obtained by using equation 11 as follows:

$$F = \mathbf{S} \cdot \left[\left(\frac{\mathbf{H(u)}}{a_p}\right)_f - \left(\frac{1}{a_P}\right)(\nabla p)_f\right] \tag{13}$$

A. The *simpleFoam* application: Implementation

The SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm solves the Navier-Stokes equations with an iterative procedure, which can be summed up as follows:

1. Set the boundary conditions;
2. Solve the discretized momentum equation to compute the intermediate velocity field;
3. Compute the mass fluxes at the cells faces;
4. Solve the pressure equation and apply under-relaxation;
5. Correct the mass fluxes at the cell faces;
6. Correct the velocities on the basis of the new pressure field;
7. Update the boundary conditions;
8. Repeat till convergence.

IV. Results

In this section, the most significant results obtained by running the *hump flow* configuration in *OpenFOAM* are presented. In order to compare our results to experimental data, the freestream Reynolds number and Mach number were fixed according to the experiments.

A. Fluid dynamic similitude

In order to operate in the subsonic regime, the freestream velocity was chosen to be 34.0 m/s, so that $M = 0.1$ The chord of our model extends from 0.0 to 1.0 m (in order to easily compare our results with the dimensionless results provided by the other works: x/c, y/c, U/U_∞). This means that in order to use the same Reynolds number as the experimental one (the Reynolds number for the baseline case is ~ 930000[1, 3, 9]) we have to change the value of the kinematic viscosity in the *transportProperties* file, as follows:

$$\nu = \frac{U_\infty c}{Re_c} = \frac{34.0 \times 1.0}{930000} = 3.66 \times 10^{-5} \mathrm{m}^2/\mathrm{s} \tag{14}$$

In the following table, we summarize the basic properties of our set up. The Reynolds number is based on the chord length of the hump[e].

c	1 m
U_∞	34 m/s
ν	3.66×10^{-5} m^2/s
M	0.1
Re_c	930 000

Table 2. Properties of our "hump flow"

B. Mesh

For our simulations, we used two different meshes: one with the inlet set at -6x/c upstream of the leading edge of the hump, to reproduce the experiments as well as possible and a second with inlet located at -1x/c upstream of the hump, to estimate the influence of the boundary layer on the flow; the goal is to compare the effects of the development of the boundary layer along the lower wall of our mesh[f]. The boundary condition on the upper wall was changed in order to study the theoretical case (no upper wall blockage) and the experimental one: in this case, the effects of the upper wall of the wind tunnel on the flow have been taken into account using a no-slip condition on the upper wall.

Our mesh was built by using the *OpenFOAM* utility *blockMesh*[g]. It was set up to be fine in the proximity of the hump[h]. In figure 2, we show a simplified sketch of the mesh employed to run our simulations. The grading value defines the scale factor of our cells. A value of 100 in the y direction means that the cells near the lower wall are 100 times smaller than the cell close to the upper wall. In table 3 we summarize the characteristics of the mesh employed for the 2-D case, the number of iterations, and the computational time to get to convergence, using 100 processors in parallel.

For the inlet located at -6 x/c, the length of the splitter plate has been selected to match the measured velocity profiles at $x/c = -2.1$ (see figure 3), as suggested by Blakumar.[10] Only 2D simulations were run, i.e. the effects of the side plates were not taken into account. Since the mesh generated by *OpenFOAM*'s *blockMesh* utility is 3D by default, to switch from 3D to 2D, we had to specify *OpenFOAM*'s *empty* condition on the side walls.

[e] In the experiments the chord length was 420 mm.

[f] The lower wall of our mesh reproduce the splitter plate used in the experiments

[g] By default, OpenFOAM handles both unstructured and structured meshes as unstructured

[h] The points are scaled by using the option *simpleGrading*

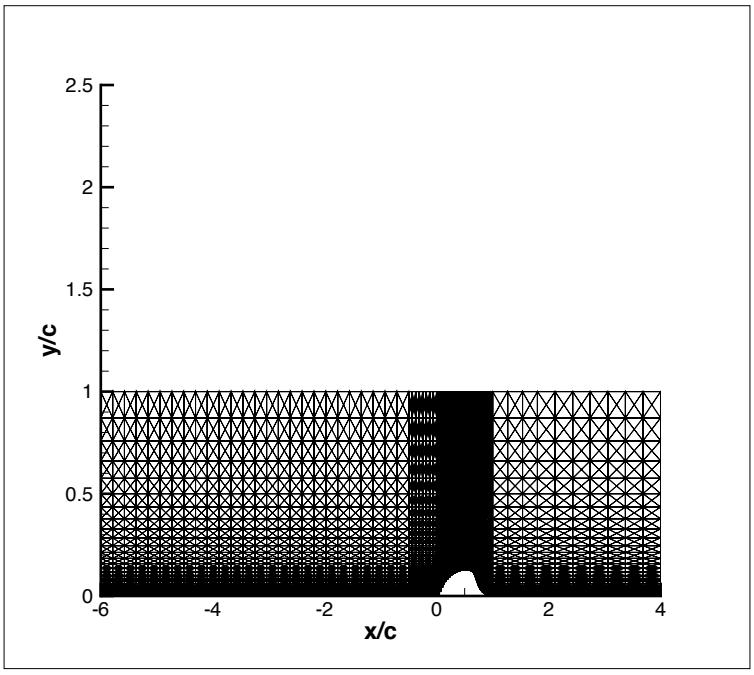

Figure 2. Sketch of our mesh. 1/10 lines plotted

	n cells	y cells	grading	num iter	n proc	Time
Spal-Allmar	336 000	300	500	30 000	100	2h30
k-ϵ	224 000	200	100	8000	100	2h
k-ω	336 000	500	300	30 000	100	2h30
k-ω-SST	336 0002	500	300	30 000	100	2h30

Table 3. Size of mesh and computational time for the 2D case

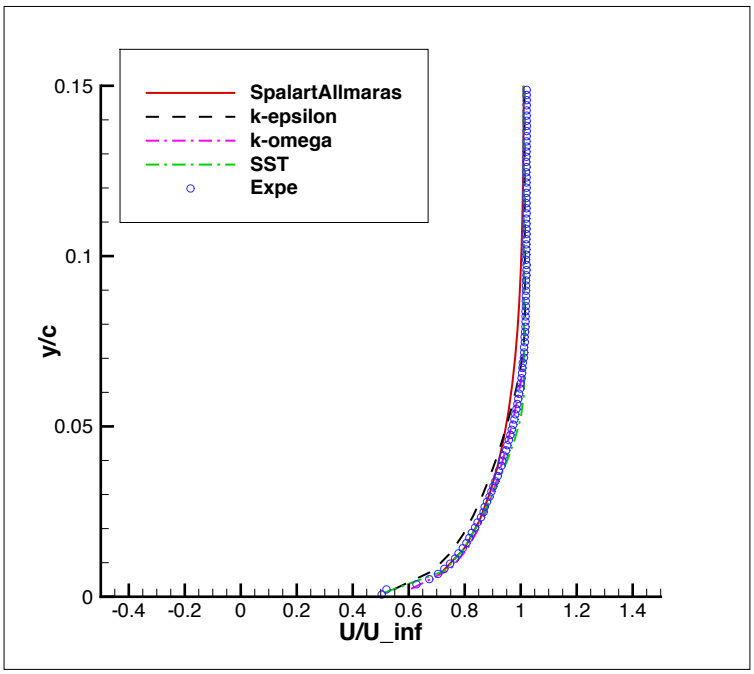

Figure 3. Velocity profiles at the experimental inlet, $x/c = -2.14$**.**

C. Boundary Conditions

An important characteristic of the "hump" flow reported by the experimental study was its insensitivity to the inlet conditions. However, from a numerical point of view, setting coherent values at the inlet is very important in order to prevent numerical problems in the solution convergence.

An important parameter to know is the turbulent intensity I, defined as $I = \frac{\sqrt{u^2}}{U_\infty}$, where u is the value of velocity fluctuation in the x-direction.

If we simplify the physics of our problem, assuming isotropic turbulence, we have the following relationship for turbulent kinetic energy:

$$k = \frac{3}{2}(IU_\infty)^2 \tag{15}$$

Turbulent viscosity, ν_t is the unknown of the problem and is calculated from the solution of the turbulence equations (k-ϵ, k-ω, *Spalart Allmaras*, $k-\omega-SST$). To a first approximation, the following relationship can be used:[11]

$$\frac{\nu_t}{\nu} \simeq 10 \text{ to } 100 \tag{16}$$

By considering the relationship between k, ϵ, ω and ν_t, we can estimate ϵ and ω, once k and ν_t are known. If we suppose a value of 1.5% for I, we have $k = \frac{3}{2}(IU_\infty)^2 = 0.39$.

ν is fixed in order to set the Reynolds number, and its value is 3.66×10^{-5}; this means that the turbulent kinematic viscosity can be chosen in the range $\nu_t \simeq 3.66 \times 10^{-4} - 3.66 \times 10^{-3}$. By using the relationships between k, ϵ, ω, and ν_t, we have:

$$\epsilon = C_\mu \frac{k}{\nu_t} \simeq 4 \text{ to } 40 \ m^2/s^3 \tag{17}$$

$$\omega \simeq \frac{k}{\nu_t} \simeq 100 \text{ to } 1000 \ s^{-1} \tag{18}$$

Velocity is determined from the Mach number to be 34 m/s at the inlet; for pressure, we use *zero gradient* condition. At the lower wall, the velocity and turbulent kinetic energy are set to 0, we assume the pressure does not change in the y-direction through the boundary layer (hypothesis of thin boundary layers), and have used a *zero gradient* condition. The turbulence dissipation rate, ϵ, and vorticity scale, ω, are calculated through the use of wall functions (*OpenFOAM*'s *epsilonWallFunction* and *omegaWallFunction*). Two

different conditions have been used for the upper wall: in one case, we assume the upper wall blockage is negligible: a *symmetry plane* condition is used. For the second case, in order to evaluate the effects of the walls on the flow, we imposed a viscous wall condition (the same used for the lower wall). At the exit, we chose a *zero gradient* condition for all the variables, except that the pressure is forced to equal the asymptotic pressure. Since in the simpleFoam solver, $p_{simpleFoam} = \frac{\Delta p}{\rho}$, we have $p = 0$ at the outlet. The boundary conditions that we have defined are summarized in tables 4 and 5.

	inlet	**lower wall**	**upper wall**	**outlet**
U	34 m/s	0 m/s	symmetry plane	zero gradient
p	zero gradient	zero gradient	symmetry plane	$p = p_\infty$
ν_t	3.66e-4 m^2/s	wall function	symmetry plane	zero gradient
k	0.39 m^2/s^2	0 m^2/s^2	symmetry plane	zero gradient
ϵ	37.4 m^2/s^3	wall function	symmetry plane	zero gradient
ω	500 1/s	wall function	symmetry plan	zero gradient

Table 4. Boundary conditions for the 2D case. Upper wall considered as a symmetry plane.

	inlet	**lower wall**	**upper wall**	**outlet**
U	34 m/s	0 m/s	0 m/s	zero gradient
p	zero gradient	zero gradient	zero gradient	$p = p_\infty$
ν_t	3.66e-4 m^2/s	wall function	wall function	zero gradient
k	0.39 m^2/s^2	0 m^2/s^2	0 m^2/s^2	zero gradient
ϵ	37.4 m^2/s^3	wall function	wall function	zero gradient
ω	500 1/s	wall function	wall function	zero gradient

Table 5. Boundary conditions for the 2D case. Upper wall considered as viscous wall.

D. Pressure

In this section the results for pressure in the 2D case are shown. In figures 4, 5, 6 and 7, the trend of pressure for the different turbulence methods is shown as contour plots covering the entire computational domain, while in figures 8 and 9, pressure coefficients are compared to experimental data. These simulations were run by using both 1-equation and 2-equation turbulence models, more specifically: *Spalart-Allmaras*[i], k-ϵ, k-ω, k-ω-SST[j]. Pressure shows how the flow is accelerated up to around the mid-chord of the hump, where a peak magnitude of C_p is observed. A sudden drop of pressure downstream leads to a separation at around x/c=0.65, which corresponds to the location of the cavity slot for the controlled cases[k]. The *exp-no-plates* data refer to the pressure coefficients measured in the absence of the end-side plates. In this case, the problem of blockage does not affect the measurements, so that the pressure is not under-predicted. In figure 8, our cases[l] are compared with experiments: all the methods are very close to the experimental results without plates, except k-ω-SST which over-predicts pressure. In figure 9, simulations are run by taking into account the effects of the upper wall, and are compared with experiments: in this case we have a slight improvement in the pressure prediction, especially for the k-ω-SST model that matches the experimental results very well. Downstream of the separation point, pressure is over predicted by *Spalart Allmaras* and k-ϵ, while k-ω and k-ω-SST provide accurate results.

[i] One equation model
[j] Two equation models
[k] In this study the controlled case will not be studied
[l] In this plot, the upper wall is treated as a symmetric plane

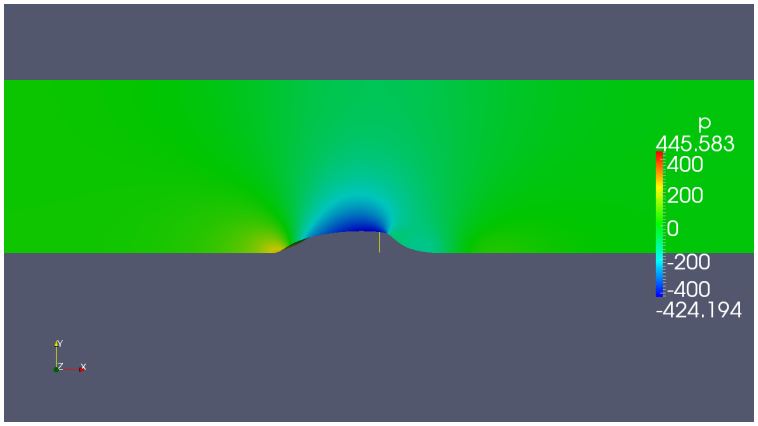

Figure 4. $\Delta p/\rho$ distribution over the plane. 2D case. Inlet at -6 x/c. Spalart-Allmaras model

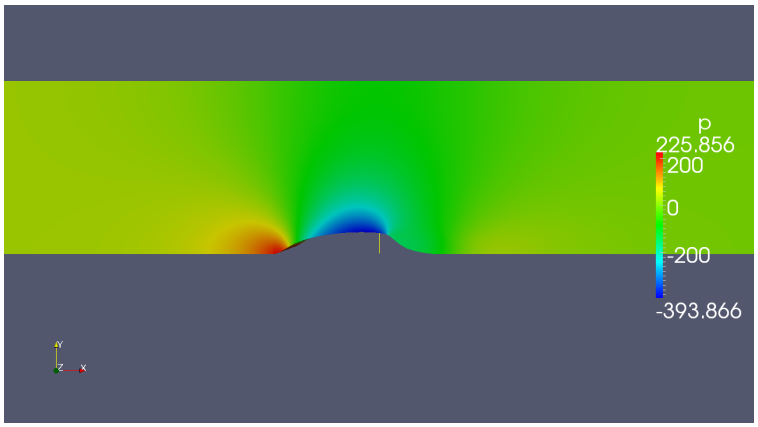

Figure 5. $\Delta p/\rho$ distribution over the plane. 2D case. Inlet at -6 x/c. k-ϵ model

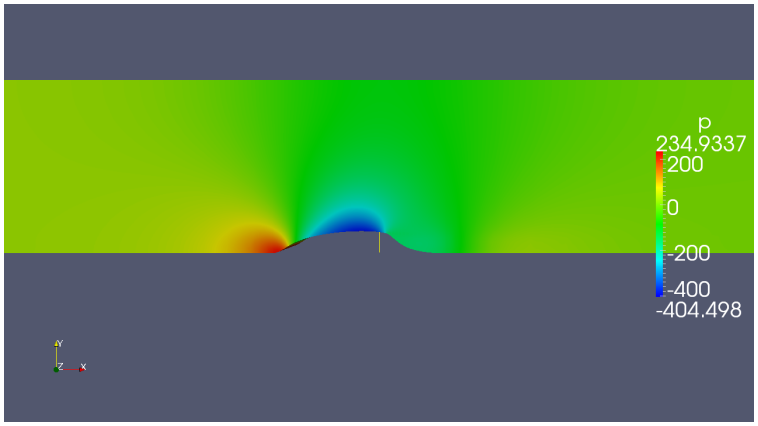

Figure 6. $\Delta p/\rho$ distribution over the plane. 2D case. Inlet at -6 x/c. k-ω model

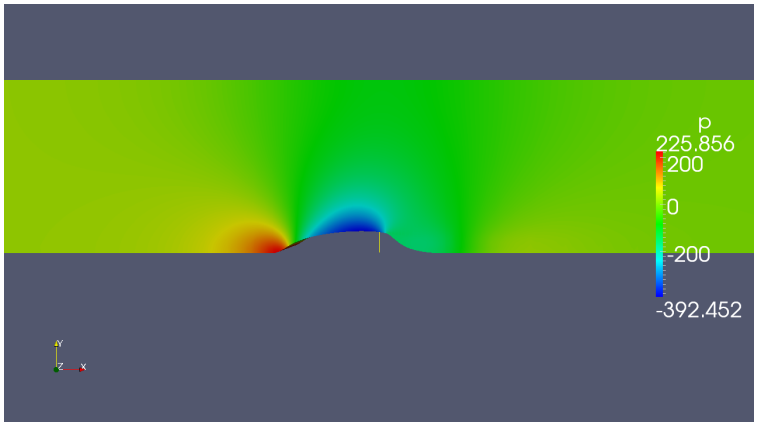

Figure 7. $\Delta p/\rho$ distribution over the plane. 2D case. Inlet at -6 x/c. k-ω-SST model

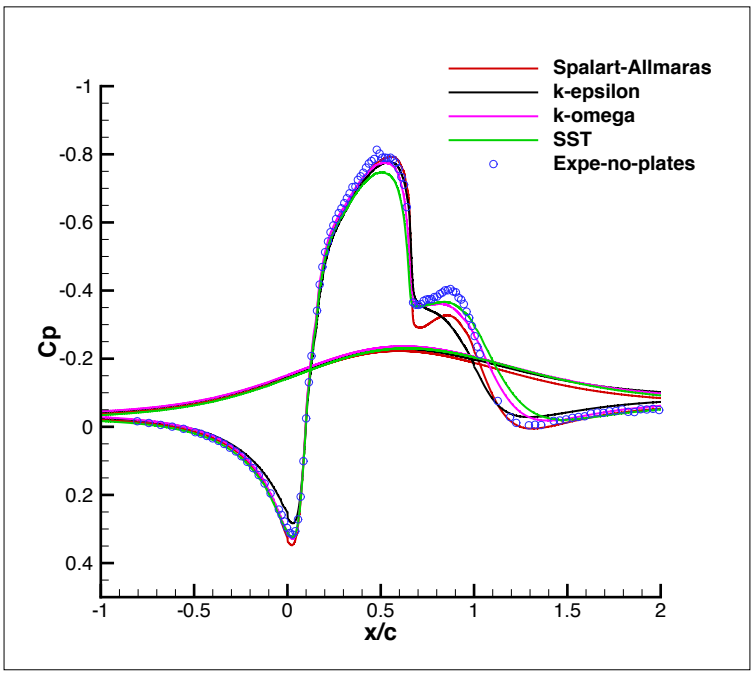

Figure 8. Comparison between numerical and experimental pressure coefficients. Theoretical case. Inlet at -6 x/c

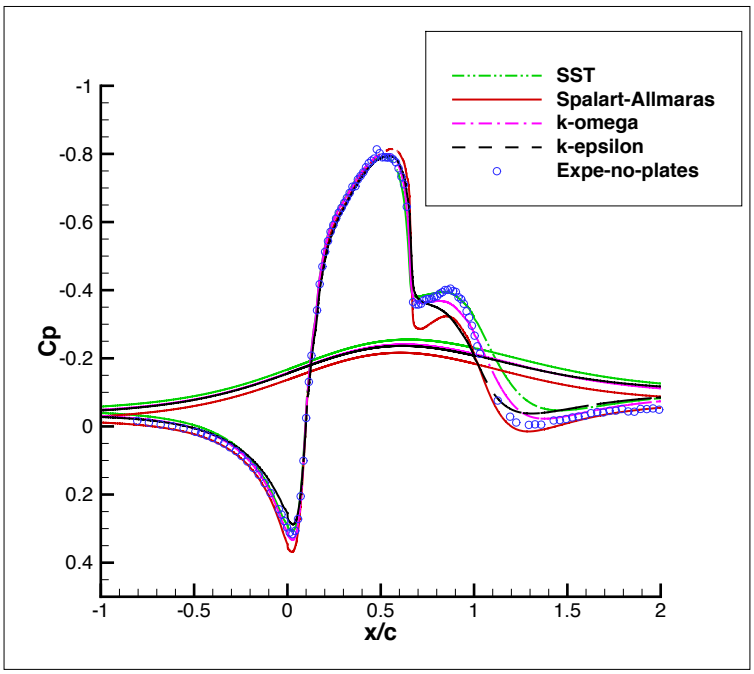

Figure 9. Comparison between numerical and experimental pressure coefficients, by taking into account the effects of the upper wall. 2D case. Inlet at -6 x/c

E. Velocity

In figures 10, 11, 12 and 13 the velocity field is shown for all turbulence models; these are 2D cases that do not take into account the blockage of the upper wall. In figures 14, 15, 16 and 17, velocity profiles at $x/c = 0.8$, $x/c = 1.0$, $x/c = 1.1$ and $x/c = 1.2$ are compared with experimental results (inlet located at -6 x/c). Our grid origin is the beginning of the hump, so that 1.0 x/c is the last point of the hump.
In the contour plots, the recirculation bubble (low speed flow marked by blue color) is visible for all of the models. The trend of velocity is the same for all models: the flow accelerates where the pressure drops and separates soon after. The size of the bubble may change for different models; we will discuss this aspect in the following section.

The most significant results are obtained by comparing the velocity profiles downstream of the hump. Velocity profiles are extracted in the bubble ($x/c = 0.8$ and $x/c = 1.0$), at the experimental reattachment point ($x/c = 1.1$), and outside the bubble ($x/c = 1.2$). The predicted profiles match the experimental results very well at all locations except $x/c = 1.2$ which is a critical location. As we will see in the next section, most numerical methods predict the reattachment point to be very close to $x/c = 1.2$, while experimental data suggest $x/c = 1.1$. *Spalart-Allmaras*, $k - \omega$, and *SST* fail to reproduce the velocity profile correctly at $x/c = 1.2$ close to the wall, while $k - \epsilon$ provides good results, probably because it is the most accurate method in predicting the position of the reattachment point[m]. In figure 18 the effects of the upper wall on velocity are shown. We can see the boundary layer developing all along the upper wall of our geometry. This is responsible for the decrease in pressure we show in figure 9. The presence of the upper wall does not affect the bubble or the velocity profiles downstream of the hump. In figure 19 we show the velocity profiles at the critical location $x/c = 1.2$. The trend is very similar to the one obtained by using a symmetry plane as the boundary condition for the upper wall.

To have an idea of the accuracy of our solutions we show the velocity profiles collected by Rumsey at $x/c = 1.2$ (figure 20). In this case, two methods predict the experimental results with a good accuracy: *AZ-cobalt-des-1-3d* and *META-cfd++lns-3D*. These are two hybrid LES/RANS methods. One of them implements the *DES* method (Detached Eddy Simulation), the other one implements the *LNS* method (Limited Numerical

[m]These aspects will be discussed deeper in the next section.

Scales). All the other curves, obtained by using RANS models, are quite far from the experimental results: an evidence is the fact that, in proximity of the wall, u-velocities are negative, while the experimental curve predicts positive values. This is a consequence of the fact that the RANS simulations used in the workshop over-predicted the length of the bubble, so that the location x/c=1.2 was still inside the recirculation zone. The good behavior of the k-ϵ method used in our simulations may be a consequence of predicting the size of the bubble properly.

F. Inlet located at $x/c = -1$

In this section we want to analyze the effects of the location of the inlet on the velocity profiles. So far, the inlet has been located at -6x/c upstream of the hump. In figures 21, 22, and 23 we show the behavior of velocity profiles for the inlet located at -1x/c upstream of the hump. In this case the boundary layer at the lower wall is thinner than that calculated for the inlet located -6 x/c upstream of the hump: when the flow separates, the wake remains thinner, so that if we move from the lower wall to the upper wall, we see that the velocity profile reaches the external flow velocity faster than the experimental results. This is very important, because it shows that, even if the "hump flow" is not sensitive to different inlet conditions (in terms of Reynolds number and Mach number as shown by Greenblatt[3] in experiments), the position of the inlet plays an important role in the accuracy of solution.

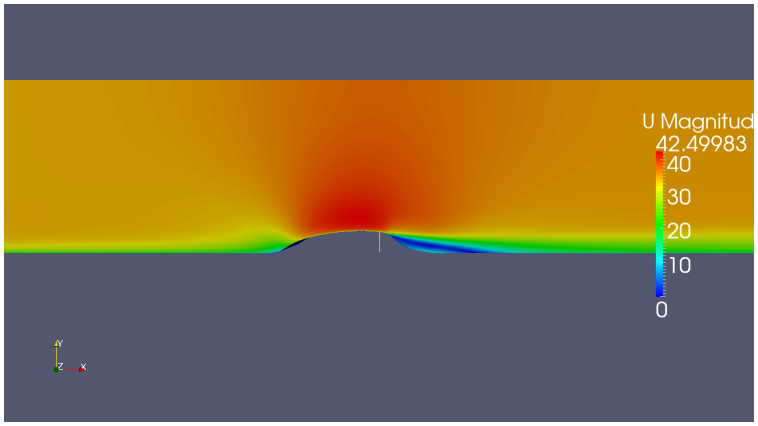

Figure 10. Zoom of velocity field in the $x-y$ plane. 2D case. Inlet at -6 x/c. Spalart-Allmaras model

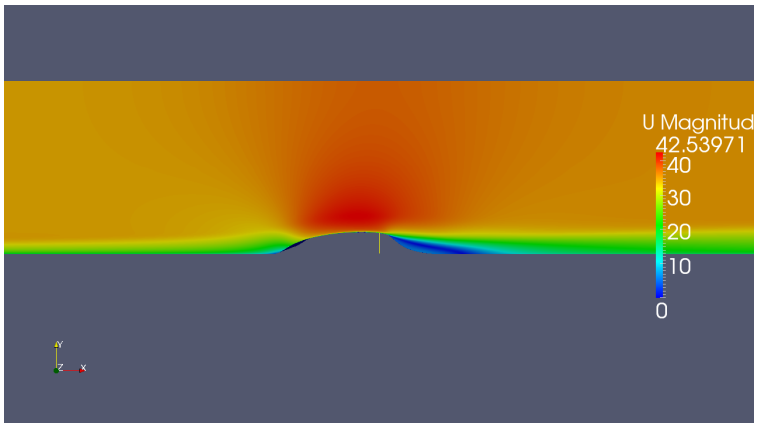

Figure 11. Zoom of velocity field in the $x-y$ plane. 2D case. Inlet at -6 x/c. k-ϵ model

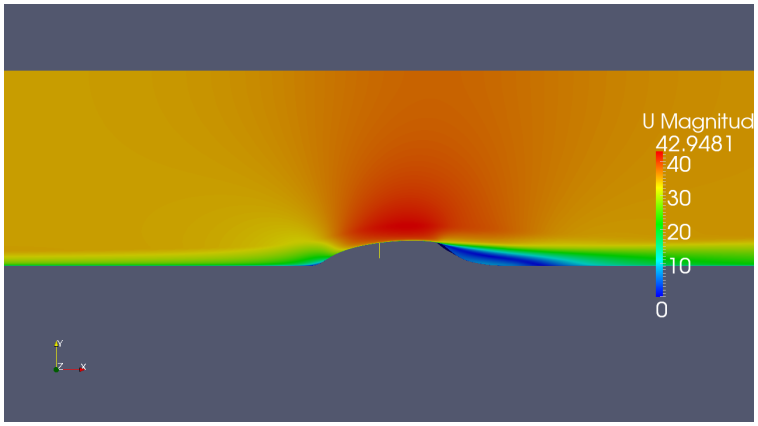

Figure 12. Zoom of velocity field in the $x-y$ plane. 2D case. Inlet at -6 x/c. k-ω model

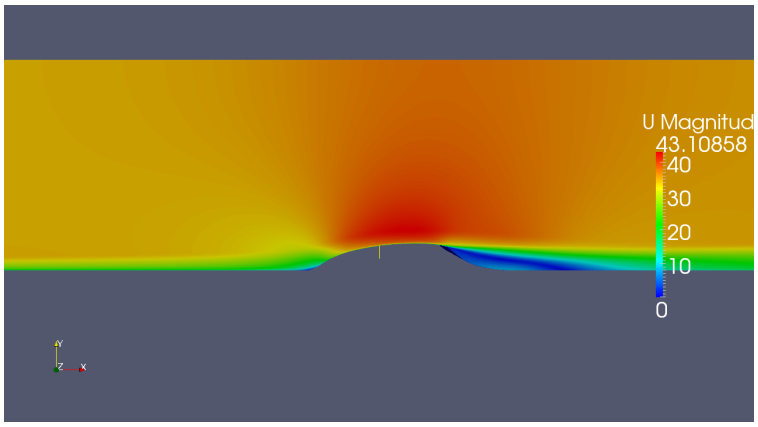

Figure 13. Zoom of velocity field in the $x-y$ plane. 2D case. Inlet at -6 x/c. k-ω-SST model

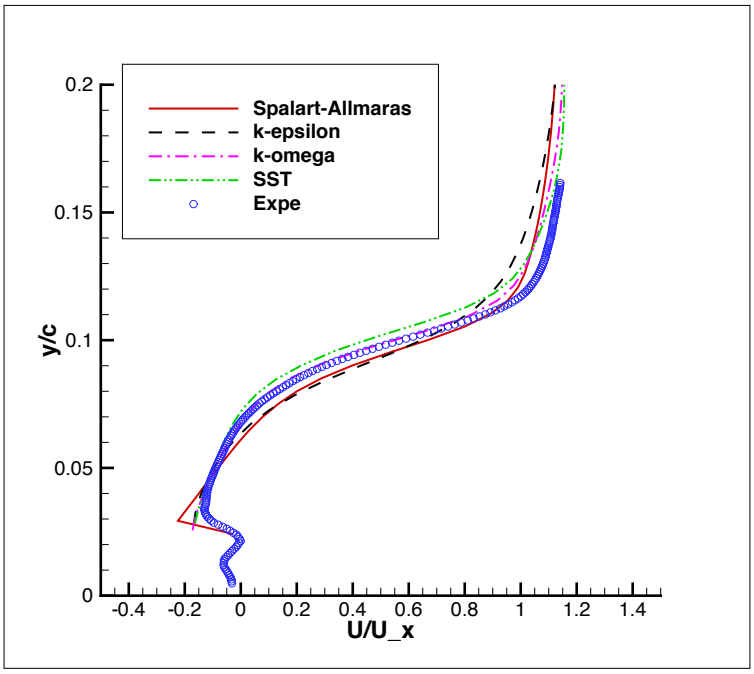

Figure 14. Simulated velocity profiles compared to experimental data. x/c=0.8. Inlet located at -6 x/c

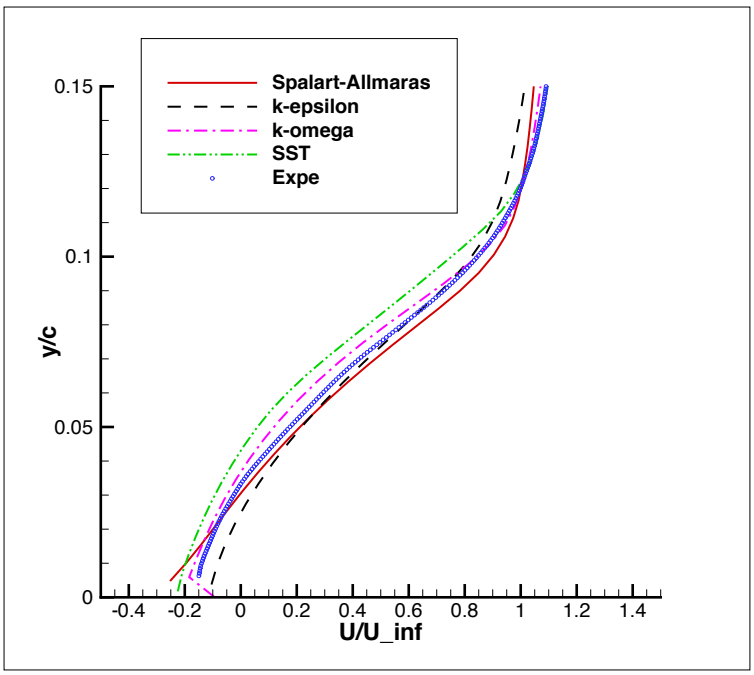

Figure 15. Simulated velocity profiles compared to experimental data. x/c=1.0. Inlet located at -6 x/c

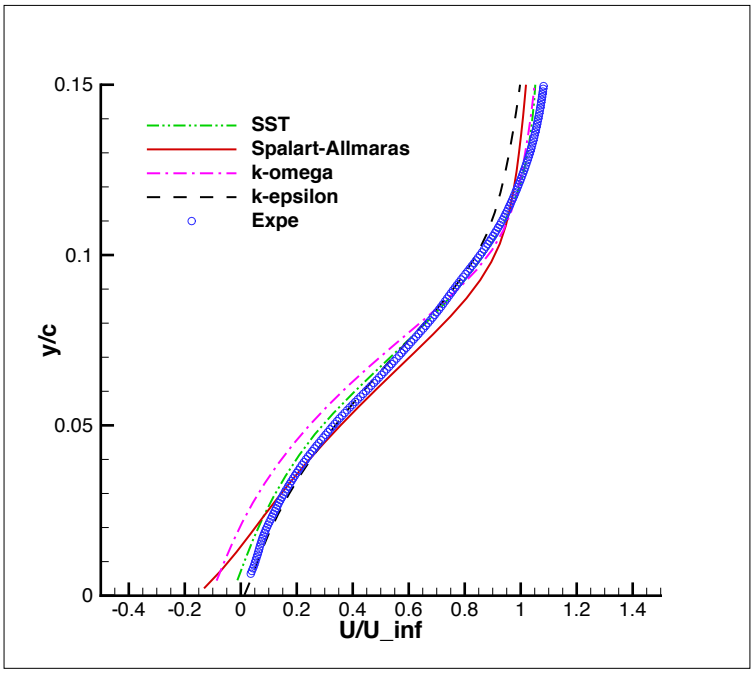

Figure 16. Simulated velocity profiles compared to experimental data. x/c=1.1. Inlet located at -6 x/c

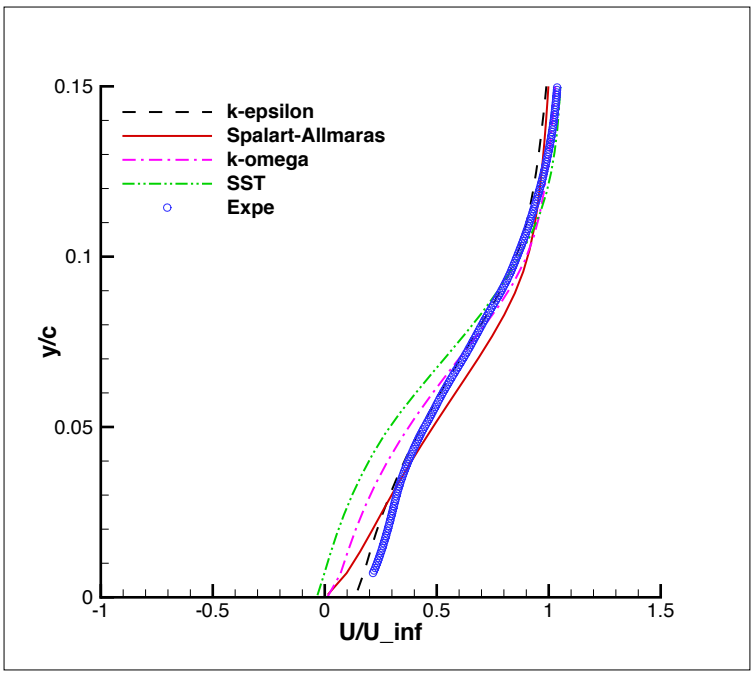

Figure 17. Simulated velocity profiles compared to experimental data. x/c=1.2. Inlet located at -6 x/c

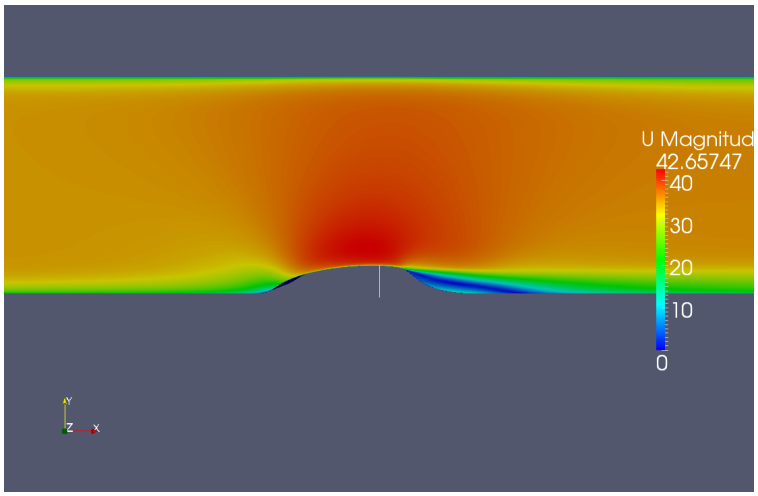

Figure 18. Velocity field in the x-y plane, by taking into account the effects of the upper wall. Inlet located at -6 x/c

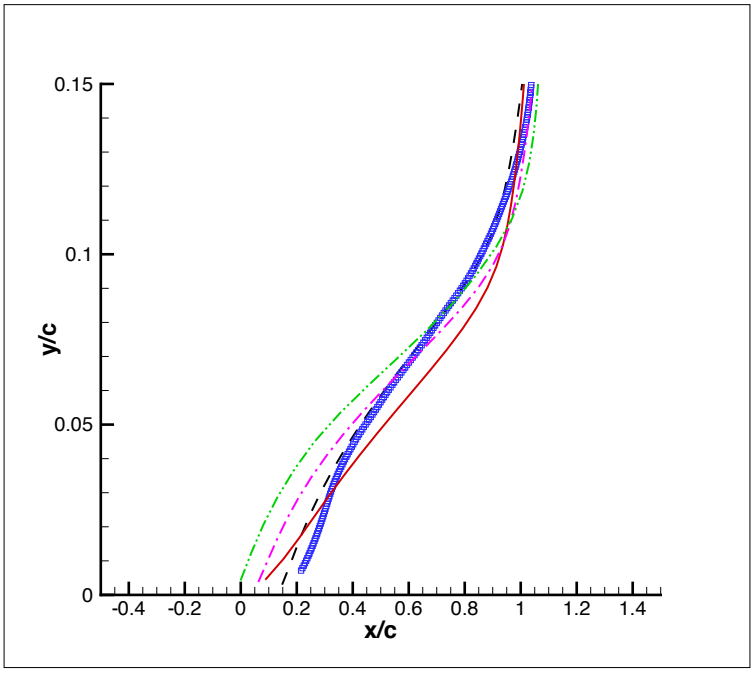

Figure 19. Simulated velocity profiles compared to experimental data, in the presence of the upper wall. x/c=1.2. Inlet located at -6 x/c

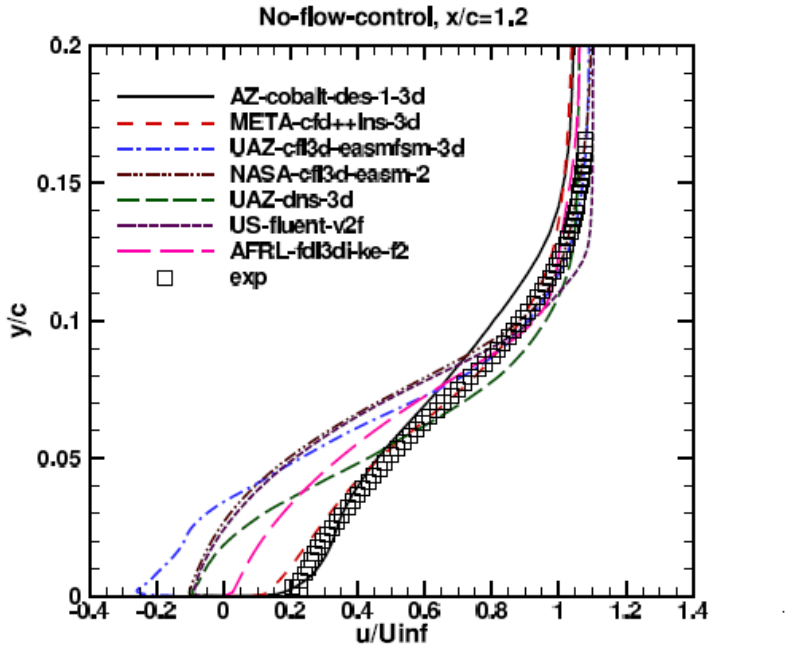

Figure 20. Velocity profiles collected by Rumsey. $x/c = 1.2$[1]

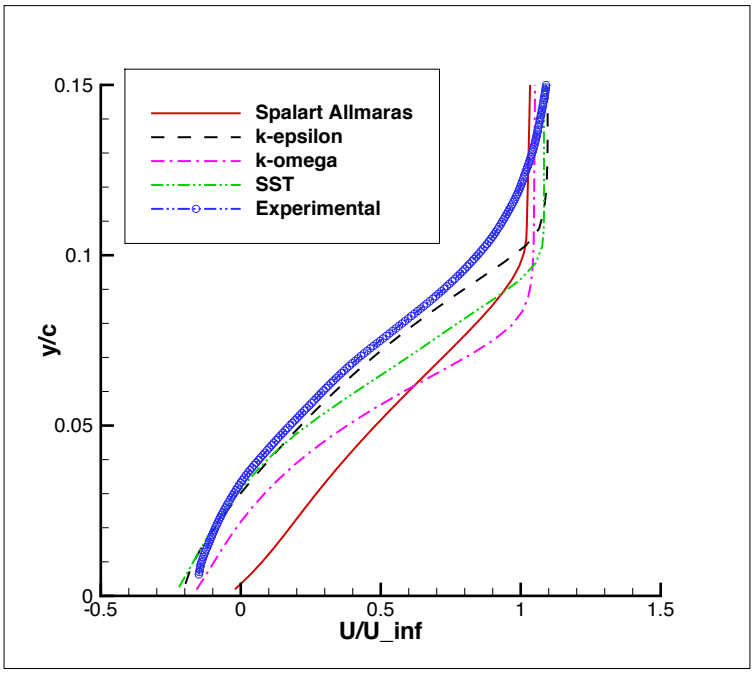

Figure 21. Simulated velocity profiles compared to experimental data. x/c=1.0. Inlet located at -1 x/c

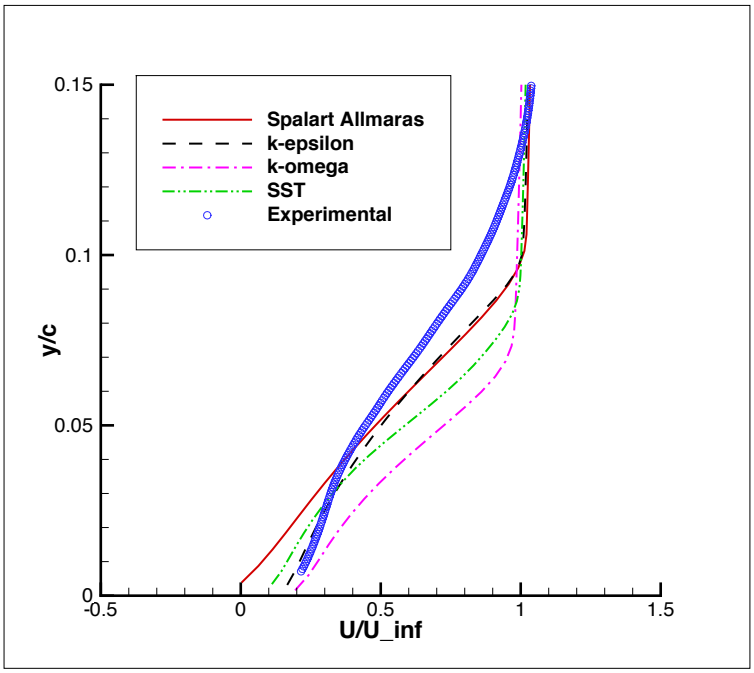

Figure 22. Simulated velocity profiles compared to experimental data. x/c=1.2. Inlet located at -1 x/c

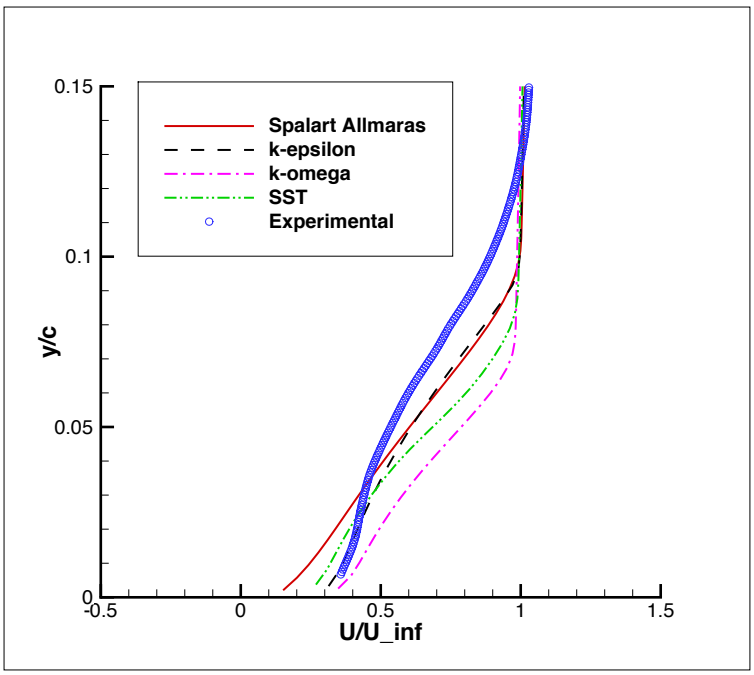

Figure 23. Simulated velocity profiles compared to experimental data. x/c=1.3. Inlet located at -1 x/c

G. Recirculation zone

The size of the bubble is an important parameter for the evaluation of the accuracy of our simulations. In order to estimate the length of the bubble, both the separation point and the reattachment point have to be evaluated. This is possible by examining the curve of the skin friction coefficient. When the curve cuts the zero axis, the term du/dn is zero. It means that the flow is separating or reattaching. In order to get the skin friction coefficient trend, we have to compute the derivative of velocity along n. This field is not one of the standard outputs in the software, so we need to modify the source code of OpenFOAM by defining this new parameter. Once du/dn is available, we can compute C_f as:

$$C_f = \frac{\tau}{\frac{1}{2}\rho U_\infty^2} = \frac{\mu \frac{\partial \bar{U}}{\partial y}}{\frac{1}{2}\rho U_\infty^2} \tag{19}$$

Figure 24 shows the skin friction coefficient over the hump for all the RANS models we considered. The upper wall blockage does not affect the value of the skin friction at the lower wall. Curves are shown for both cases (viscous wall and symmetry plane) with the inlet located at -6x/c. By identifying the points where the C_f curves cut the axis, it possible to locate the separation point and the reattachment point. As we can see in table 6 and in figure 24, the separation point is accurately predicted by all the RANS models, while the only model able to catch the exact position of reattachment is $k-\epsilon$. This explains why the velocity profiles downstream of the reattachment point are better predicted by the $k-\epsilon$ model as compared to the other models. Figures 25, 26, 27 and 28 show the size of the bubble computed with different *RANS* models. The smallest bubble is predicted by $k-\epsilon$, the biggest by $k-\omega-SST$ as summarized in table 6.

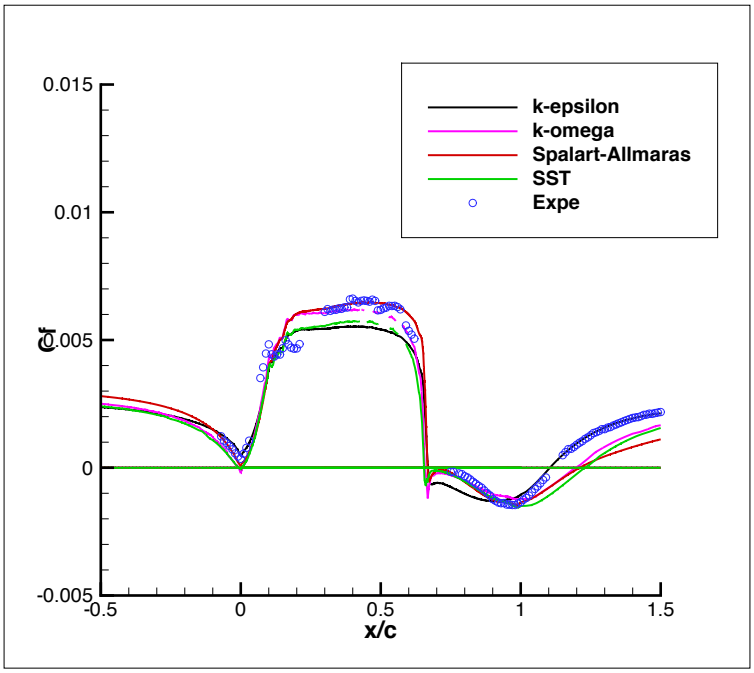

Figure 24. Skin friction coefficient. Inlet located at $x/c = -6$

H. Turbulent Shear Stress

Rumsey reported that one of the parameters that can affect the prediction of the size of the bubble is the turbulent shear stress. He concluded that as long as this parameter is not predicted properly, the turbulence in the bubble can be compromised and its length and size mispredicted. After having defined the derivatives of velocity in *OpenFOAM*, we compute the turbulent shear stress by using Boussinesq's hypothesis:

$$\overline{u'v'} = -2\nu_t \overline{S_{ij}} = -\nu_t \left(\frac{\partial \bar{u}}{\partial y} + \frac{\partial \bar{v}}{\partial x} \right) \tag{20}$$

	Separation point [x/c]	Reattachment point [x/c]
Spalart Allmaras	0.667	1.198
k-ϵ	0.669	1.104
k-ω	0.656	1.196
k-ω-SST	0.656	1.229
Experiments	**0.65-0.67**	**1.11**

Table 6. Location of the the separation point and of the reattachment point for the different RANS models.

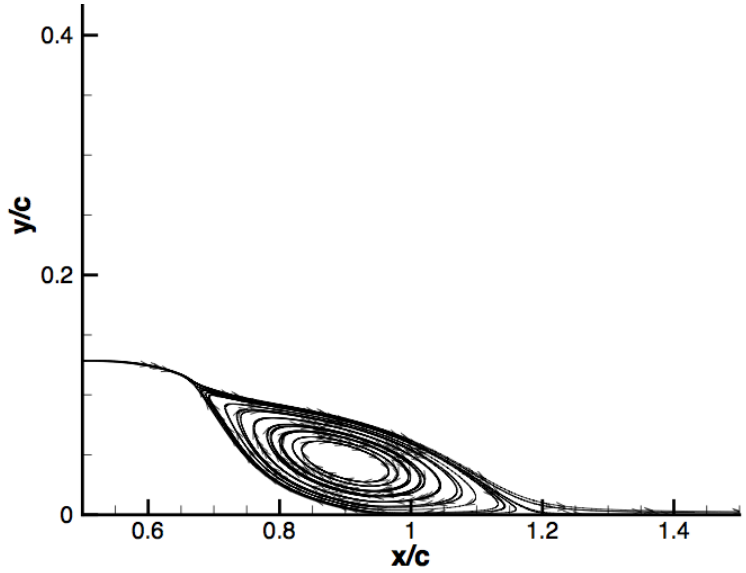

Figure 25. Visualization of the bubble in *TecPlot* for the *Spalart-Allmaras* model

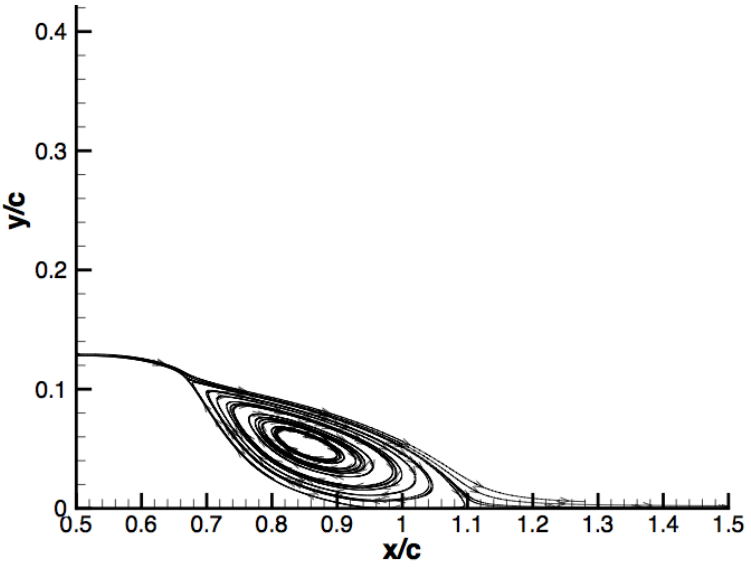

Figure 26. Visualization of the bubble in *TecPlot* for the $k-\epsilon$ model

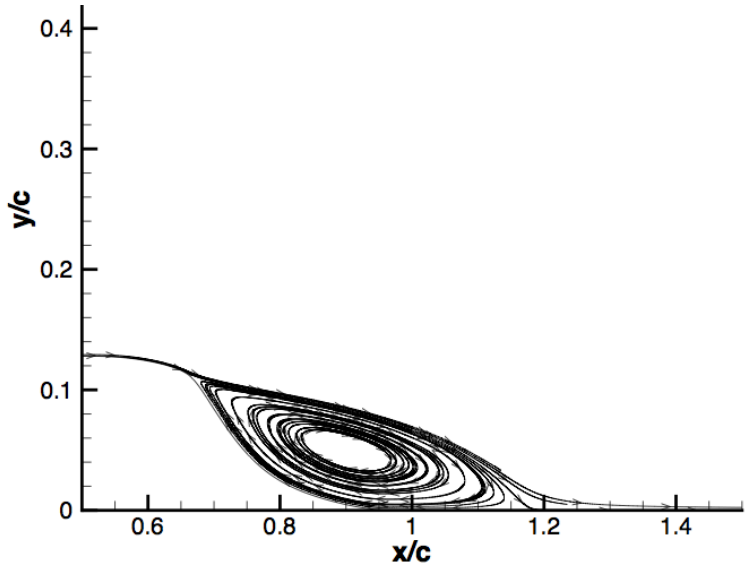

Figure 27. Visualization of the bubble in *TecPlot* for the $k-\omega$ model

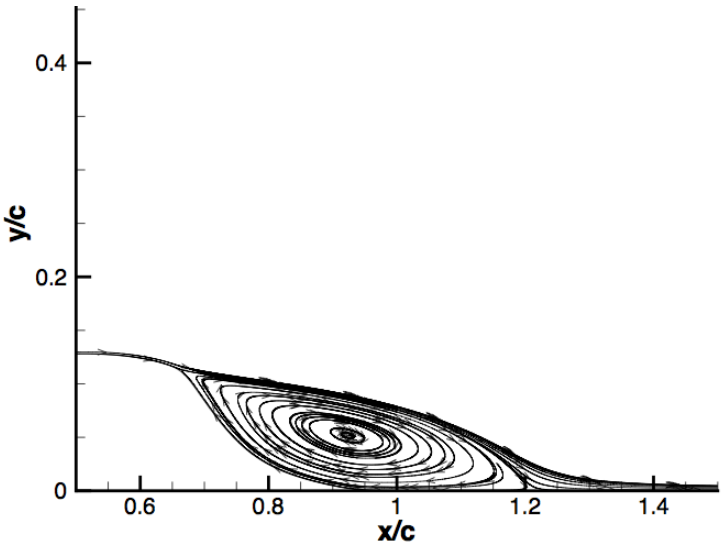

Figure 28. Visualization of the bubble in *TecPlot* for the SST model

In figure 29 we compare the turbulent shear stress computed by using the equation 20 with the experiments. We find that, for all of the models, the turbulent shear stress is under-predicted. These results are consistent with Rumsey's findings, but, as discussed in the previous section, the k-ϵ model results seem to contradict his conclusion. This in fact is not the case; Rumsey's conclusion holds: as we will see in the next section.

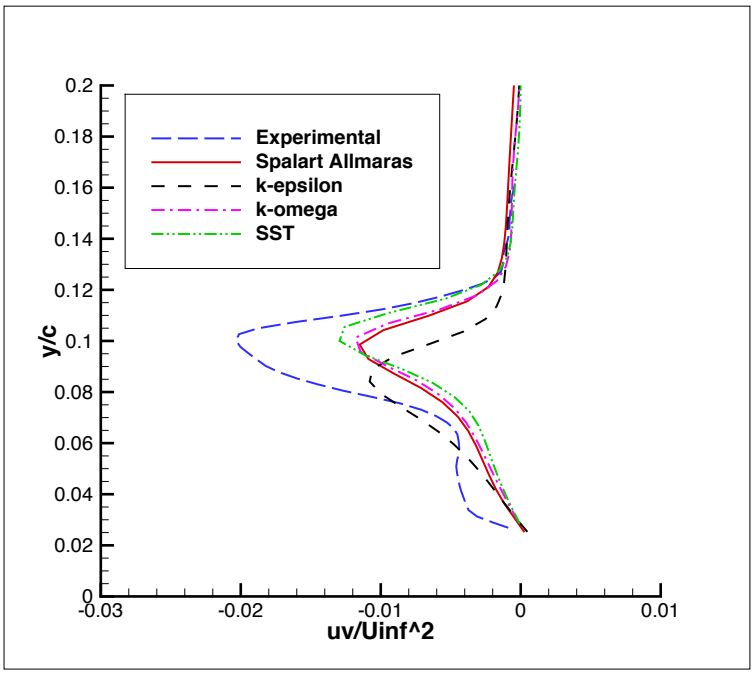

Figure 29. Turbulent shear stress at $x/c = 0.8$.

I. The problem of recovery

So far, we have shown the velocity profiles in the proximity of the bubble ($x/c = 0.8$, $x/c = 1.0$, $x/c = 1.1$ and $x/c = 1.2$). In this section we deal with the problem of recovery by analyzing velocity profiles downstream of the reattachment point (x/c=1.3). At this location, we observe that all of the models fail to reproduce the experimental velocity profile; this is directly the consequence of under-predicting the shear stress as observed by Rumsey. In all of the cases the mixing of the free-stream turbulence with the wall-generated turbulence during the boundary layer recovery is poorly modeled. We point to the fact that results obtained using a coarse mesh are closer to the experimental data than those obtained with a fine mesh; this is a clear indication that the eddy viscosity is under-predicted by the models. The numerical viscosity of a coarse mesh causes an increase in mixing, so that the results are closer to the experiments, for the wrong reason.

V. Conclusion

The results that we have obtained show that the k-ϵ model as implemented in *OpenFOAM* predicts the pressure coefficient over the hump and the velocity profiles upstream and inside the separation bubble better than the other models. However, the recovery of the mean profiles is not predicted downstream of the reattachment point.

The boundary conditions used for the upper wall (*symmetry plane* or *viscous wall*) do not significantly change the results, even though the *viscous wall* condition can help adjust and improve the predictions of the pressure coefficient distribution over the hump by the *SST* model.

Concerning the size of the bubble, the $k-\epsilon$ model predicts the separation point and the reattachment point with good accuracy, while $Spalart-Allmaras$, $k-\omega$ and $k-\omega-SST$ over predict the length of the recirculation region. This may be caused by the fact that RANS models under-predict the magnitude of the turbulent shear stress, as observed by Rumsey in the workshops of 2004 and 2008. The k-ϵ model happens

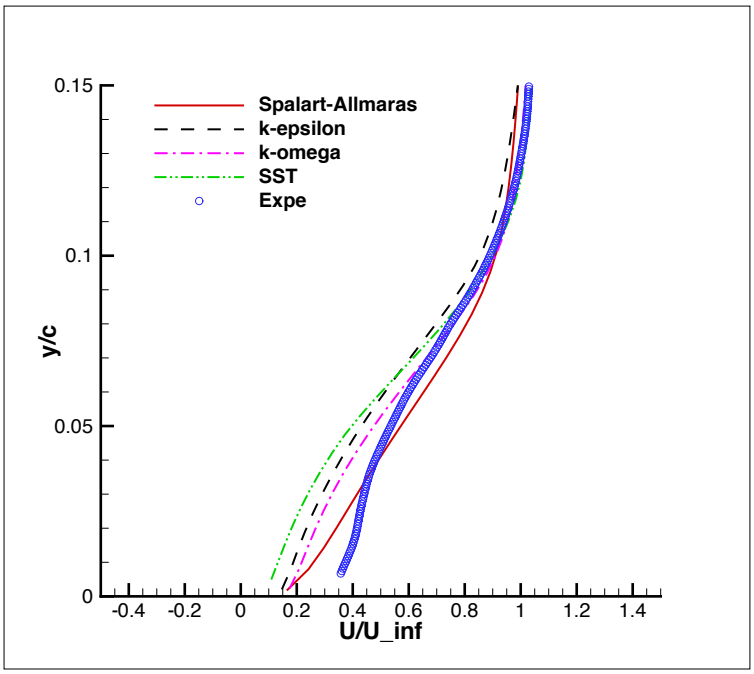

Figure 30. Simulated velocity profiles compared to experimental data. x/c=1.3. Inlet located at -6 x/c

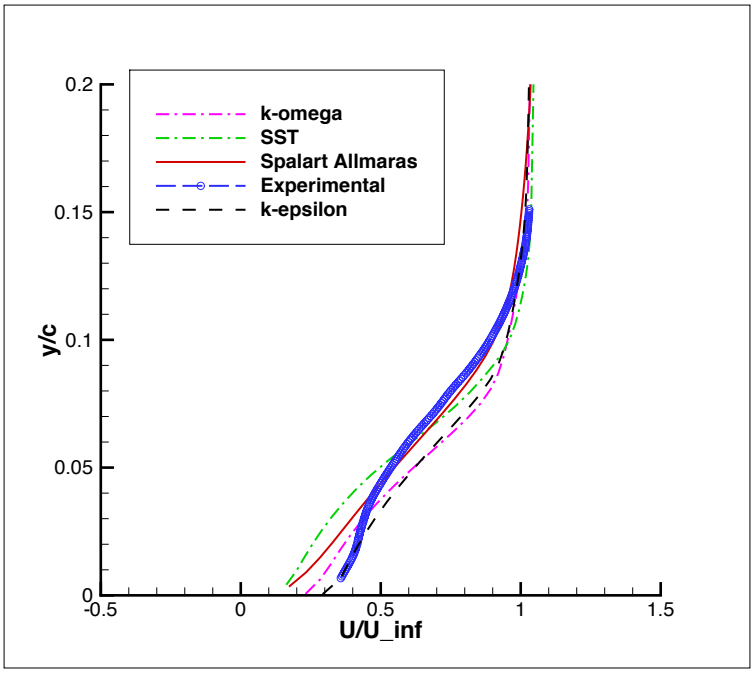

Figure 31. Simulated velocity profiles compared to experimental data. x/c=1.3. Inlet located at -6 x/c. Coarse mesh

to match the shear stress at the location that affects the size of the bubble.

Even if the hump-flow is considered insensitive to inlet conditions in terms of the Reynolds and Mach numbers, the development of the boundary layer upstream of the hump strongly affects the velocity profiles downstream. This is evidenced by the fact that by moving the *inlet* face of our mesh from $x/c = -6$ to $x/c = -1$, the field of velocity computed downstream of the model changes substantially.

Particular attention has to be paid to the resolution of the mesh, which plays a very important role in the accuracy of results. More specifically, since we are interested in the derivatives of velocity (skin friction coefficient) and in the pressure distribution over the hump, the mesh has to be very fine close to the lower wall. In our work, we have scaled the mesh so that the points in proximity of the hump are 300 or 500 times smaller than in the free stream. A mesh that does not satisfy these requirements does not provide accurate values of pressure and skin friction or can provide good results for the wrong reason (problem of recovery of velocity).

Finally, our results are consistent with those obtained by He et al,[4] who used the commercial software *Fluent*. But predictions could be different from other results, even when computed using the same software, because of the great sensitivity of the solution to all of the variables discussed in the paper.

Appendix

The equation for momentum is defined in the file *UEqn.H*

```
 tmp<fvVectorMatrix> UEqn
 (
     fvm::div(phi, U)
   + turbulence->divDevReff(U)
   ==
     sources(U)
 );
 UEqn().relax();
 sources.constrain(UEqn());
 solve(UEqn() == -fvc::grad(p));

 ...

  tmp<fvVectorMatrix> kEpsilon::divDevReff(volVectorField& U) const
{
 return
    (
   - fvm::laplacian(nuEff(), U)
   - fvc::div(nuEff()*dev(T(fvc::grad(U))))
     );
}
```

The source code for $k - \epsilon$ model is in the directory *openfoam211/src/turbulenceModels/RAS/kEpsilon*. Here we report the lines of code where the equations for k and ϵ are implemented:

```
// Turbulent kinetic energy equation
   tmp<fvScalarMatrix> kEqn
   (
       fvm::ddt(k_)
     + fvm::div(phi_, k_)
     - fvm::Sp(fvc::div(phi_), k_)
     - fvm::laplacian(DkEff(), k_)
    ==
       G
     - fvm::Sp(epsilon_/k_, k_)
   );
```

```
    kEqn().relax();
    solve(kEqn);
    bound(k_, kMin_);

      // Dissipation equation
   tmp<fvScalarMatrix> epsEqn
    (
        fvm::ddt(epsilon_)
      + fvm::div(phi_, epsilon_)
      - fvm::Sp(fvc::div(phi_), epsilon_)
      - fvm::laplacian(DepsilonEff(), epsilon_)
     ==
        C1_*G*epsilon_/k_
      - fvm::Sp(C2_*epsilon_/k_, epsilon_)
    );
    epsEqn().relax();
    epsEqn().boundaryManipulate(epsilon_.boundaryField());

    solve(epsEqn);
    bound(epsilon_, epsilonMin_);

     // Re-calculate viscosity
    nut_ = Cmu_*sqr(k_)/epsilon_;
    nut_.correctBoundaryConditions();
```

The source code for $k-\omega$ is available at the directory: *openfoam211/src/turbulenceModels/incompressible/RAS/kOmega*. The following lines show how the equation for k and the equation for ω are implemented in this case:

```
void kOmega::correct()
{
    RASModel::correct();

    if (!turbulence_)
    {
        return;
    }

    volScalarField G("RASModel::G", nut_*2*magSqr(symm(fvc::grad(U_))));

    // Update omega and G at the wall
    omega_.boundaryField().updateCoeffs();

    // Turbulence specific dissipation rate equation
    tmp<fvScalarMatrix> omegaEqn
    (
        fvm::ddt(omega_)
      + fvm::div(phi_, omega_)
      - fvm::Sp(fvc::div(phi_), omega_)
      - fvm::laplacian(DomegaEff(), omega_)
     ==
        alpha_*G*omega_/k_
      - fvm::Sp(beta_*omega_, omega_)
    );

    omegaEqn().relax();

    omegaEqn().boundaryManipulate(omega_.boundaryField());
```

```
    solve(omegaEqn);
    bound(omega_, omegaMin_);


    // Turbulent kinetic energy equation
    tmp<fvScalarMatrix> kEqn
    (
        fvm::ddt(k_)
      + fvm::div(phi_, k_)
      - fvm::Sp(fvc::div(phi_), k_)
      - fvm::laplacian(DkEff(), k_)
     ==
        G
      - fvm::Sp(Cmu_*omega_, k_)
    );

    kEqn().relax();
    solve(kEqn);
    bound(k_, kMin_);


    // Re-calculate viscosity
    nut_ = k_/omega_;
    nut_.correctBoundaryConditions();
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Acknowledgments

The authors would like to thank the following people for the help they provided: Susheel Sekhar NPP Fellow, NASA ARC, Jean Lachaud, ARC, University of California - Santa Cruz, and Julien De Muelenaere PhD student at Stanford University.

References

[1]C.L. Rumsey and T.B. Gatski and W.L. Sellers and V.N. Vatsa and S.A. Viken, *Summary of the 2004 CFD Validation Workshop on Synthetic Jets and Turbulent Separation Control* , Nasa Langley Research Center, AIAA 2005-1270, 2005.

[2]Christopher L. Rumsey *Successes and Challenges for Flow Control Simulations* , Nasa Langley Research Center - AIAA 2008-4311, 2008.

[3]David Greenblatt and Keith B. Paschal and Chung-Sheng Yao and Jerome Harris and Norman Schaeffler and Anthony E. Washburn, *A Separation Control CFD Validation Test Case. Part 1*, AIAA 2004-2220, 2004.

[4]Chuan He and Thomas C. Corke , *Numerical and Experimental Analysis of Plasma Flow Control Over a Hump Model* , University of Notre Drame, IN, 2007.

[5]Bettini, C. and Cravero, C, *Computational Analysis of Flow Separation Control for the Flow over a Wall-MountedHump Using a synthetic Jet*, AIAA Paper 2007-0516, 2007.

[6]Franjo Juretic, *Error analysis in Finite Volume CFD*, PhD Thesis, Imperial College, 2004.

[7]Jasak H., *Error analysis and estimation for the Finite Volume method with applications to fluid flows*, PhD Thesis, Imperial College, 1996.

[8]OpenFOAM foundation, *www.openfoam.org/doc* , User Guide, 2012.

[9]http://cfdval2004.larc.nasa.gov/ , *CFD Validation of Synthetic Jets and Turbulent Separation Control* , NASA Langley Research Center.

[10]P.Balakumar, *Computations of Flow over a Hump Model Using Higher Order Method with Turbulence Modeling* , Nasa Langley Research Center, AIAA 2004-2220, 2004.

[11]Platteeuw P.D.A, *Application of the Probabilistic Collocation method to uncertainty in turbulence models* , M. Sc. THesis, 2008