Penalization of the Spalart-Allmaras turbulence model without and with a wall function: methodology for a Vortex In Cell scheme

H. Beaugendre[a,b], F. Morency[c]

[a] *Bordeaux INP, IMB, UMR 5251, F-33400,Talence, France*
[b] *Inria Bordeaux Sud-Ouest, Team CARDAMOM,*
200 Avenue de la Vieille Tour, 33405 Talence, France
corresponding author: heloise.beaugendre@math.u-bordeaux.fr
[c] *TFT Laboratory, Mechanical Engineering Department*
École de Technologie Supérieure, Montréal, Canada.
francois.morency@etsmtl.ca

Abstract

Immersed boundary methods (IBM) are alternative methods to simulate fluid flows around complex geometries. The grid generation is fast as it does not need to conform to the fluid-solid interface. However, special treatments are needed in the flow equations to properly take into account the wall proximity. The penalization method is a particular case of the IBM in which the wall boundary conditions are imposed via continuous forcing terms into the governing equations. Reynolds Averaged Navier-Stokes (RANS) equations completed with a turbulence model are still the most common way to model turbulence in engineering applications. However, RANS turbulence model implementation with penalization into a vortex formulation is not straight forward, in part because of the variable turbulent viscosity and partly because of the boundary conditions. This paper extends the penalization technique to turbulent flows. The objective of this paper is to validate the use of the Spalart-Allmaras turbulence model in the context of penalization and vortex formulation. Details of the resolution using a Vortex In Cell (VIC) numerical scheme are given. The proposed scheme is based on the advection of particles of vorticity and particles of turbulent

Email address: heloise.beaugendre@math.u-bordeaux.fr (H. Beaugendre)

Authors' accepted manuscript -- Article published in *Computers & Fluids*, vol. 170 (July 15, 2018), p. 313-323.
https://doi.org/10.1016/j.compfluid.2018.05.012

viscosity. A Lagrangian framework is chosen to solve the advection part. The remaining parts of the system of equations are solved with an Eulerian framework using a Cartesian uniform grid. To avoid fine meshes near the wall, a wall function compatible with the penalization method and the vortex formulation is proposed. The formulation and the coding are validated against the well-known periodic channel flow. Velocity profiles are computed without and with the wall function. Results agree with analytic law of the wall solutions, showing that RANS simulations can be conducted with VIC schemes and penalization.
Keywords: Turbulence, SA model, Vortex Formulation, VIC scheme, Brinkman Penalization, Wall function, IBM

1. Introduction

Immersed boundary methods (IBM) are alternative methods to simulate fluid flows around complex geometries submitted to arbitrarily large deformation or motion [1]. The grid generation is fast as it does not need to conform to the fluid-structure interface. However, a special treatment is needed in the flow equations to properly take into account the wall proximity. LESCAPE is an immersed boundary code in which the wall boundary conditions are imposed via continuous forcing terms added to the equations of conservation. To take into account wall boundary conditions, LESCAPE uses a porous media approach called the Brinkman penalization [2]. The discretization is done using a Vortex In Cell (VIC) scheme [3]. This penalization technique is extended, in this paper, to turbulent flows. LESCAPE code uses level set functions to localize solids and track motions [4, 5, 6]. The velocity-vorticity formulation of the incompressible Navier-Stokes equations are solved on a uniform Cartesian grid [5].

The objectives of this paper are to formulate and validate the Spalart-Allmaras turbulence [7] model using a vortex formulation in the framework of penalization methods. In this context, the Spalart-Allmaras turbulence model has been implemented into LESCAPE code to allow flow simulations at high Reynolds numbers. Although extensible to 3D, the developments are presented

for 2D flows simulations. The model implementation follows the methodology associated to a VIC scheme [3, 8]. A time splitting algorithm is used to solve the system of equations and a penalty term is added to each equation to enforce boundary conditions on the solid [9]. The global strategy used by LESCAPE is similar to the work performed by Lee [8] except in the treatment of the turbulence. In this work, we propose to solve the RANS equations and use in addition the Spalart-Allmaras turbulence model for closure. In this sense, we remove high frequencies linked to the turbulence in our solution. In Lee's work [8] the VIC scheme is directly used as a filter making a kind of LES simulations as discussed by Cottet [3]. The plan of this article is as follows: first, the specific vortex formulation of the Spalart-Allmaras turbulence model used is presented. Then, the numerical discretization of the equations, using a VIC scheme, is detailed. In a third part, to avoid excessive fine meshes, a penalized wall function is proposed and details on the implementation are given. Finally, the periodic channel flows is used to validate the model and the implementation technique.

2. Penalized RANS-Vortex formulation

2.1. Physical and mathematical model

We consider an incompressible turbulent flow. The governing equations, mass and momentum conservations, are

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

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nabla \cdot \mathbf{\Pi} \tag{2}$$

where ρ is the fluid density, $\mathbf{u}$ the mean velocity vector and p the pressure. The viscous stress tensor, $\mathbf{\Pi}$, is $(\mu+\mu_t)\left[\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right]$ for incompressible flows. The symbols μ and μ_t are respectively the laminar and turbulent dynamic viscosity.

The turbulent viscosity $\mu_t = \nu_t \rho$ in this equation is computed from the Spalart-Allmaras turbulence model. As there exist several forms of the model, the equation used in this work is detailed below [7]. The one equation model is:

$$\frac{\partial \tilde{\nu}}{\partial t} + (\mathbf{u} \cdot \nabla)\tilde{\nu} = P - D + \frac{1}{\sigma}\left[\nabla \cdot ((\nu + \tilde{\nu})\nabla\tilde{\nu}) + c_{b2}(\nabla\tilde{\nu})^2\right] \tag{3}$$

where the production and wall destruction are:

$$P = c_{b1}\,(1 - f_{t2})\,\tilde{S}\tilde{\nu} \quad ; \quad D = \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right] \left(\frac{\tilde{\nu}}{d}\right)^2 . \tag{4}$$

The turbulent viscosity is computed from

$$\mu_t = \rho \tilde{\nu} f_{v1}\,.$$

where the value of f_{v1} is given by

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \quad \text{and} \quad \chi = \frac{\tilde{\nu}}{\nu}\,.$$

To avoid negative values of the modified vorticity $\tilde{S}$ in the production term [7], we define $\bar{S}$ such that $\bar{S} = \dfrac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}$. Then, $\tilde{S}$ is computed by the following method

$$\tilde{S} = \Omega + \bar{S} \quad \text{if} \quad \bar{S} \geq -c_2 \Omega$$
$$\tilde{S} = \Omega + \frac{\Omega\left(c_2^2 \Omega + c_3 \bar{S}\right)}{\left(c_3 - 2c_2\right)\Omega - \bar{S}} \quad \text{if} \quad \bar{S} < -c_2 \Omega$$

where $c_2 = 0.7$, $c_3 = 0.9$, and Ω is the magnitude of the vorticity.

The distance from the wall is d, and

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}\,, \qquad f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6},$$
$$g = r + c_{w2}\left(r^6 - r\right)\,, \qquad r = \min\left(\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10\right)\,,$$
$$f_{t2} = c_{t3} \exp\left(-c_{t4}\chi^2\right)$$

The constants are: $c_{b1} = 0.1355$, $\sigma = 2/3$, $c_{b2} = 0.622$, $\kappa = 0.41$, $c_{w2} = 0.3$, $c_{w3} = 2$ $c_{v1} = 7.1$, $c_{t3} = 1.2$, $c_{t4} = 0.5$ $c_{w1} = c_{b1}/\kappa^2 + (1 + c_{b2})/\sigma$.

In a vortex formulation, the computed variable is the flow vorticity defined by

$$\omega = \nabla \wedge \mathbf{u} = \begin{pmatrix} \dfrac{\partial u_z}{\partial y} - \dfrac{\partial u_y}{\partial z} \\ \dfrac{\partial u_x}{\partial z} - \dfrac{\partial u_z}{\partial x} \\ \dfrac{\partial u_y}{\partial x} - \dfrac{\partial u_x}{\partial y} \end{pmatrix}.$$

In 2D, only ω_z remains and ω is a scalar. In this case, after the application of the curl operator to the momentum equations, the elongation term $(\omega \cdot \nabla)\mathbf{u}$ vanishes and the equation to solve is the following

$$\begin{aligned}\rho\frac{\partial\omega}{\partial t}+\rho(\mathbf{u}\cdot\nabla)\omega &= (\mu+\mu_t)\Delta\omega+2\frac{\partial(\mu+\mu_t)}{\partial x}\frac{\partial\omega}{\partial x}+2\frac{\partial(\mu+\mu_t)}{\partial y}\frac{\partial\omega}{\partial y}\\ &+ \frac{\partial^2(\mu+\mu_t)}{\partial x^2}\left(\frac{\partial u_y}{\partial x}+\frac{\partial u_x}{\partial y}\right)-\frac{\partial^2(\mu+\mu_t)}{\partial y^2}\left(\frac{\partial u_y}{\partial x}+\frac{\partial u_x}{\partial y}\right)\\ &+ 2\frac{\partial^2(\mu+\mu_t)}{\partial y\partial x}\left(\frac{\partial u_y}{\partial y}-\frac{\partial u_x}{\partial x}\right) \qquad (5)\end{aligned}$$

The variable turbulent viscosity introduces additional terms in the vorticity equation. The relative importance of each term is discussed briefly in [10].

After some algebra, using the incompressibility relation (1) and the definition of $\omega = \omega_z$, equation (5) can be rewritten as follows

$$\begin{aligned}\frac{\partial\omega}{\partial t}+(\mathbf{u}\cdot\nabla)\omega &= (\nu+\nu_t)\Delta\omega+\omega\Delta(\nu+\nu_t)+S_\omega\\ &+ 2\left[\frac{\partial(\nu+\nu_t)}{\partial x}\frac{\partial\omega}{\partial x}+\frac{\partial(\nu+\nu_t)}{\partial y}\frac{\partial\omega}{\partial y}\right]. \qquad (6)\end{aligned}$$

with S_ω defined by

$$S_\omega = 2\left[\left(\frac{\partial u_x}{\partial y}\frac{\partial^2(\nu+\nu_t)}{\partial x^2}-\frac{\partial u_y}{\partial x}\frac{\partial^2(\nu+\nu_t)}{\partial y^2}\right)+\frac{\partial^2(\nu+\nu_t)}{\partial y\partial x}\left(\frac{\partial u_y}{\partial y}-\frac{\partial u_x}{\partial x}\right)\right] \qquad (7)$$

We define by ν_{eff} the effective viscosity, $\nu+\nu_t$ and the final system of equations to solve in LESCAPE becomes:

$$\begin{aligned}\frac{\partial\omega}{\partial t}+(\mathbf{u}\cdot\nabla)\omega &= \nu_{eff}\Delta\omega+\omega\Delta\nu_{eff}+2\left[\frac{\partial\nu_{eff}}{\partial x}\frac{\partial\omega}{\partial x}+\frac{\partial\nu_{eff}}{\partial y}\frac{\partial\omega}{\partial y}\right]+S_\omega\\ \frac{\partial\tilde{\nu}}{\partial t}+(\mathbf{u}\cdot\nabla)\tilde{\nu} &= P-D+\frac{1}{\sigma}\left[\nabla\cdot\left((\nu+\tilde{\nu})\nabla\tilde{\nu}\right)+c_{b2}\left(\nabla\tilde{\nu}\right)^2\right]. \qquad (8)\end{aligned}$$

2.2. Immersed boundary method via penalization

The solid body, si, inside the domain is localized via a level set function, $\Phi(si)$, which in our case is the signed distance function d (negative inside the solid, positive outside). The level 0 of the signed distance function represents the fluid-solid interface. We define by $H(\Phi_{si})$ the Heaviside function applied

to Φ_{si} (or characteristic function of si) which equals one inside the solid and 0 outside. To impose the rigid motion of our solid, if $\mathbf{u}_{si}$ is the solid velocity vector, we should impose the following conditions in the solid: $\mathbf{u} = \mathbf{u}_{si}$ and $\tilde{\nu} = 0$. The extension of the Brinkman penalization to our system of equations is then the following:

$$\begin{aligned} \frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla)\omega &= \nu_{eff}\Delta\omega + \omega\Delta\nu_{eff} + 2\left[\frac{\partial \nu_{eff}}{\partial x}\frac{\partial \omega}{\partial x} + \frac{\partial \nu_{eff}}{\partial y}\frac{\partial \omega}{\partial y}\right] + S_\omega \\ &+ \boldsymbol{\lambda\nabla} \wedge \left[\mathbf{H}(\mathbf{\Phi(si)})\left(\mathbf{u_{si}} - \mathbf{u}\right)\right] \end{aligned} \tag{9}$$

$$\begin{aligned} \frac{\partial \tilde{\nu}}{\partial t} + (\mathbf{u} \cdot \nabla)\tilde{\nu} &= P - D + \frac{1}{\sigma}\left[\nabla \cdot \left(\left(\nu + \tilde{\nu}\right)\nabla\tilde{\nu}\right) + c_{b2}\left(\nabla\tilde{\nu}\right)^2\right] \\ &+ \boldsymbol{\lambda}\mathbf{H}(\mathbf{\Phi(si)})\left(\mathbf{0} - \boldsymbol{\tilde{\nu}}\right). \end{aligned} \tag{10}$$

The forcing terms appear in bold, and $\lambda \gg 1$ is the penalty parameter. In Brinkman *et al.* [2], $\lambda = \dfrac{\nu}{K_{si}}$, where K_{si} is the permeability of the solid.

3. Numerical scheme details

LESCAPE is formalized on regular Cartesian grids. The VIC scheme consists in using a viscous splitting algorithm in which the advective parts of the equations are solved in a Lagrangian framework while the other parts of the equations are solved on the grid with an Eulerian approach.

-Advection:
$$\begin{aligned} \frac{\partial \omega}{\partial t} + \left(\mathbf{u} \cdot \nabla\right)\omega = 0. \\ \frac{\partial \tilde{\nu}}{\partial t} + (\mathbf{u} \cdot \nabla)\tilde{\nu} = 0. \end{aligned}$$

-Source terms and diffusion:
$$\begin{aligned} \frac{\partial \omega}{\partial t} &= \nu_{eff}\Delta\omega + \omega\Delta\nu_{eff} + 2\left[\frac{\partial \nu_{eff}}{\partial x}\frac{\partial \omega}{\partial x} + \frac{\partial \nu_{eff}}{\partial y}\frac{\partial \omega}{\partial y}\right] + S_\omega. \\ \frac{\partial \tilde{\nu}}{\partial t} &= P - D + \frac{1}{\sigma}\left[\nabla \cdot \left(\left(\nu + \tilde{\nu}\right)\nabla\tilde{\nu}\right) + c_{b2}\left(\nabla\tilde{\nu}\right)^2\right]. \end{aligned}$$

-Penalization:
$$\begin{aligned} \frac{\partial \omega}{\partial t} &= \lambda\nabla \times \left(H\left(\Phi_{si}\right)\left(\mathbf{u}_{si} - \mathbf{u}\right)\right). \\ \frac{\partial \tilde{\nu}}{\partial t} &= \lambda H(\Phi(si))\left(0 - \tilde{\nu}\right). \end{aligned}$$

-Advection:

For the advective parts, in the region of the domain where the vorticity is above a threshold, the vorticity and $(\tilde{\nu}-\tilde{\nu}_\infty)$ are modeled by discrete particles localized at grid points. $\tilde{\nu}_\infty$ is related to the turbulent viscosity at free stream. The particles, containing the discrete vorticity and $(\tilde{\nu}-\tilde{\nu}_\infty)$ are then convected with a fourth order Runge-Kutta time-stepping scheme. The third order interpolation kernel M_4' [11] is used to interpolate the vorticity and the rate of change of turbulent viscosity back on the grid. Both vorticity and turbulent viscosity are transported by the same particle, reducing the computational cost, and the same interpolation kernel can be used.

To impose periodic boundary conditions, the particles that get out of the computational domain on one side are reinjected on the other side. Because of the stencil of the M_4' interpolation kernel, a band of three virtual nodes must be added on both sides of the computational domain. A periodic computational domain that extends in x direction from x_{start} to x_{end} is illustrated on the figure 1. The computational domain is discretized in x direction by n nodes spaced by dx. The nodes $n-2$ to n are duplicated on the left side and the nodes 1 to 3 are duplicated on the right side to impose periodicity in x direction. These virtual regions are illustrated with dashed lines. The content of a particle between node 1 and node 2, illustrated by a black dot on the left, will be distributed on 16 nodes, illustrated by circles. If the particle is located between x_{end} and $x_{end}+dx$, as the particle between node n and 1 on the right, the content of the particle is distributed on the 16 nearest nodes, including virtual nodes 1 and 2. Finally, if a particle is located at x_p such that $x_p > x_{end}+dx$ or $x_p < x_{start}$, then the particle is relocated according to the following rules

$$x_p = (x_p - (x_{end}+dx)) + x_{start} \quad \text{if } x_p > (x_{end}+dx) \tag{11}$$

$$x_p = x_{end} - (x_{start} - x_p) + dx \quad \text{if } x_p < (x_{start}) \tag{12}$$

-Source terms and diffusion:

A first order Euler implicit scheme is used for the time derivative. Second order

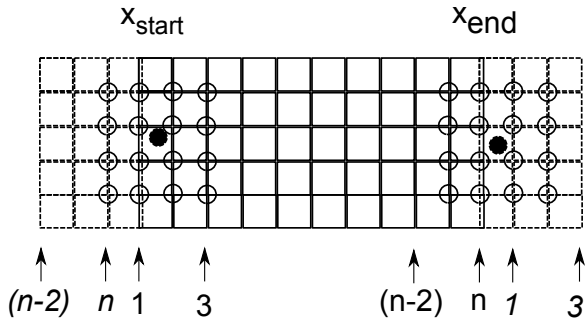


Figure 1: **Periodic boundary conditions in x direction for the advection part**

finite differences are used for spatial derivatives. A linearization of the source terms in equation (9) is proposed to prevent too much constraint on the time step.

$$\begin{aligned}\frac{\omega^{n+1}-\omega^n}{\Delta t} &= \nu_{eff}^n \Delta\omega^{n+1} + \omega^{n+1}\Delta\nu_{eff}^n + 2\left[\frac{\partial\nu_{eff}^n}{\partial x}\frac{\partial\omega^{n+1}}{\partial x} + \frac{\partial\nu_{eff}^n}{\partial y}\frac{\partial\omega^{n+1}}{\partial y}\right] \\ &+ S_\omega^n .\end{aligned}$$

$$S_\omega^n = 2\left[\left(\frac{\partial u_x^n}{\partial y}\frac{\partial^2\nu_{eff}^n}{\partial x^2} - \frac{\partial u_y^n}{\partial x}\frac{\partial^2\nu_{eff}^n}{\partial y^2}\right) + \frac{\partial^2\nu_{eff}^n}{\partial y\partial x}\left(\frac{\partial u_y^n}{\partial y} - \frac{\partial u_x^n}{\partial x}\right)\right]. \qquad (13)$$

where ω^n is the value at the previous time step, and ω^{n+1} is the value at the actual time step. The discretized equations are solved using an Alternate Direction Implicit (ADI) method [12]. The ADI method does not have the time step stability limitation of the explicit method and has a tridiagonal matrix structure that can be solved efficiently with the Tridiagonal Matrix Algorithm (TDMA).

The same strategy is used for the turbulent source terms with the following linearization:

$$\begin{aligned}\frac{\tilde\nu^{n+1}-\tilde\nu^n}{\Delta t} &= S_p\tilde\nu^{n+1} + S_c \\ &+ \frac{1}{\sigma}\left[\frac{\partial\left(\nu+\tilde\nu^n\right)}{\partial x_j}\left(\frac{\partial\tilde\nu^{n+1}}{\partial x_j}\right) + \left(\nu+\tilde\nu^n\right)\frac{\partial}{\partial x_j}\frac{\partial\tilde\nu^{n+1}}{\partial x_j} + c_{b2}\frac{\partial\tilde\nu^n}{\partial x_i}\frac{\partial\tilde\nu^{n+1}}{\partial x_i}\right]\end{aligned}$$

with

$$S_p = -\frac{2}{d}\left[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right]\left(\frac{\tilde{\nu}^n}{d}\right)$$

$$S_c = c_{b1}\left(1 - f_{t2}\right)\tilde{S}\tilde{\nu}^n + \left[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right]\left(\frac{\tilde{\nu}^n}{d}\right)^2$$

where $\tilde{\nu}^n$ is the value at the previous time step, and $\tilde{\nu}^{n+1}$ is the value at the actual time step. An ADI method is also used to solve the turbulent equation.

-Penalization:

An implicit Euler time discretization is used to approximate $\mathbf{u}^{n+1}$ and $\tilde{\nu}^{n+1}$ in the penalty source terms to avoid time step contraints due to a large penalty parameter. Details related to the implicit treatment of the penalty term can be found in [6]. The vorticity field at t^{n+1} is then evaluated on the grid by taking the **curl** of the velocity near the penalized area.

4. Wall function

The numerical wall function follows an idea proposed by [13], further extended by [14] and used with the Spalart-Allmaras turbulent model [15]. Consider a Cartesian grid with a fluid region and a penalized solid region as shown in figure 2. The solid region, outlined in light gray, does not follow the Cartesian grid. For the test cases presented in this paper, the solid velocity is zero. The velocity vector U is defined at each node located at grid line intersection. A local coordinate system aligned with the wall normal $\mathbf{n}$ is defined at each node located nearer than a normal distance δ away from the solid. On the figure, the local coordinate system $(\mathbf{n}, \mathbf{s})$ is defined at node A. Node A is located at a distance $y_n < \delta$ from the surface. The local coordinate system is rotated at an angle ϕ from the Cartesian coordinate system.

The velocity vector U is decomposed either in component (u_s, u_n) or (u_x, u_y), such that

$$u_x = u_s cos\phi - u_n sin\phi \tag{14}$$

$$u_y = u_s sin\phi + u_n cos\phi \tag{15}$$

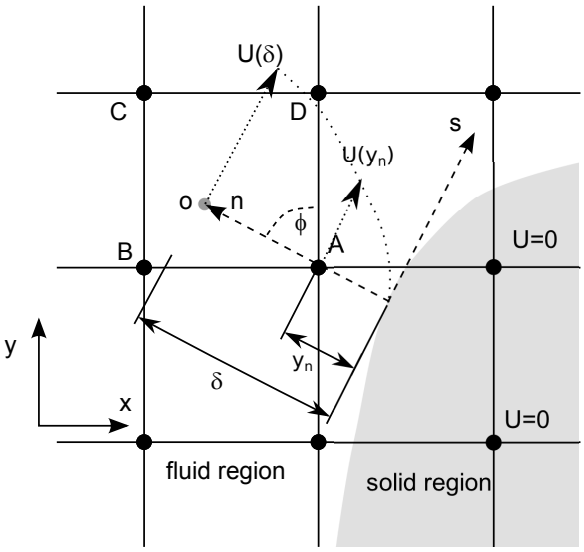


Figure 2: **Cartesian grid and local coordinate system**

The velocity $U(y_n)$ imposed at the node A is obtained from a wall function model. The wall function links the tangential velocity to the normal distance. For known velocities at nodes o and at the fluid-solid interface, the velocity at the node A is computed with the following assumptions:

- The normal velocity component u_n is zero.
- The velocity varies linearly in x and y direction around the node o.

4.1. Wall function mathematical model

For a node A located at a distance y_n from the wall, the location of the node o in the Cartesian coordinate system is obtained by

$$x_o = x_A - (\delta - y_n) sin\phi \tag{16}$$

$$y_o = y_A + (\delta - y_n) cos\phi \tag{17}$$

Then, the velocity at (x_o, y_o) is approximated by bilinear interpolations based on the cartesian velocities U_A, U_B, U_C, and U_D at nodes A, B, C, and D.

$$u_{x,o} = (1-f)(1-g)u_{x,B} + f(1-g)u_{x,A} + fgu_{x,D} + (1-f)gu_{x,C} \quad (18)$$

$$u_{y,o} = (1-f)(1-g)u_{y,B} + f(1-g)u_{y,A} + fgu_{y,D} + (1-f)gu_{y,C} \quad (19)$$

where

$$f = (x_0 - x_B)/(x_A - x_B) \quad (20)$$

$$g = (y_0 - y_B)/(y_C - y_B) \quad (21)$$

The tangential velocity component, needed for the wall function, is obtained from the Cartesian velocity components

$$u_{s,o} = u_{x,o}cos\phi + u_{y,0}sin\phi \quad (22)$$

$$U(\delta) = u_{s,o} \quad (23)$$

The wall model is of the form [13]

$$\frac{u_s}{u_\tau} = \frac{1}{\kappa}ln(1+\kappa y^+) + c(1 - e^{-y^+/d+} - \frac{y^+}{d^+}e^{-by^+}) \quad (24)$$

$$b = \frac{1}{2}\left(\frac{d^+\kappa}{c} + \frac{1}{d^+}\right) \quad (25)$$

$$c = \frac{1}{\kappa}ln\left(\frac{E}{\kappa}\right) \quad (26)$$

$$y^+ = \frac{u_\tau y}{\nu} \quad (27)$$

$$u_\tau = \sqrt{\frac{\tau}{\rho}} \quad (28)$$

where κ, E and d^+ assume the values of 0.4187, 9.793 and 11 respectively. This model is valid even for y^+ values below 10. For $y^+ \leq 1$, the RANS solution should be identical to the one obtained with Spalart-Allmaras turbulent model without wall function. The u_τ value is obtained from $u_s = u_{s,o}$ in equation (24) and δ values. Once u_τ is known, the tangential velocity component can

be computed at any point along the normal direction. At the node A, the tangential velocity is

$$y_A^+ = \frac{u_\tau y_n}{\nu} \tag{29}$$

$$U(y_n) = \left(\frac{1}{\kappa}\ln(1+\kappa y_A^+) + c(1-e^{-y_A^+/d+} - \frac{y_A^+}{d^+}e^{-by_A^+})\right)u_\tau \tag{30}$$

and the normal velocity is 0.

4.2. Numerical method

The normal distance from the wall is obtained from the level set function. The n_w near-wall nodes are identified as the nodes for which the level set values are above 0 but below the threshold value $\delta = 2 \times dy$. For every near-wall nodes A_i, the gradient of the level set gives the normal direction.

$$\mathbf{n} = \nabla\Phi(si)/|\nabla\Phi(si)| \tag{31}$$

Then, from the normal direction, the corresponding node position $(x_o, y_o)_i$ is found from equations (16) and (17).

The four nodes around $(x_o, y_o)_i$ are found from the horizontal and vertical distances between nodes, dx and dy. The four nodes around $(x_o, y_o)_i$ are obtained with integer divisions.

$$i_B = i_C = \text{int}\left(\frac{(x_o - x_{ini})}{dx}\right) \tag{32}$$

$$j_B = j_A = \text{int}\left(\frac{(y_o - y_{ini})}{dy}\right) \tag{33}$$

The value of u_τ is obtained iteratively from equation (24) using a Newton-Raphson method. The real valued function h is

$$\kappa^* = \kappa\frac{\delta}{\nu} \tag{34}$$

$$b^* = b\frac{\delta}{\nu} \tag{35}$$

$$h(u_\tau) = \frac{u_s}{u_\tau} - \frac{1}{\kappa}\ln(1+\kappa^* u_\tau) - c\left(1 - e^{-\frac{\delta}{\nu d^+}u_\tau} - \frac{\delta}{\nu d^+}u_\tau e^{-b^* u_\tau}\right) \tag{36}$$

The derivative of the function h is

$$\begin{aligned} h' = \frac{dh}{du_\tau} &= -u_s/u_\tau^2 - \frac{\kappa^*}{\kappa}\left(\frac{1}{1+\kappa^* u_\tau}\right) \\ &\quad -c\frac{\delta}{\nu d^+}\left(e^{-\frac{\delta}{\nu d^+}u_\tau} - e^{-b^* u_\tau} + b^* u_\tau e^{-b^* u_\tau}\right) \end{aligned} \tag{37}$$

At $t = 0$, the initial shear velocity value $u_{\tau,0}$ for the iterative process is obtained by using a linear velocity profile approximation

$$u_{\tau,0} = \sqrt{\frac{\nu u_s}{\delta}}\,. \tag{38}$$

Starting from the initial value, the shear velocity approximation is iteratively improved

$$u_{\tau,i} = u_{\tau,i-1} - h/h'\,. \tag{39}$$

Usually, within less than 10 iterations, the residual $|u_{\tau,i} - u_{\tau,i-1}|/u_{\tau,i}$ falls below 10^{-6}.

With the u_τ value and the wall distance y_n, the wall model equation (24) is used to compute the tangential velocity at node A. This velocity, once projected onto the Cartesian directions, becomes the penalized velocity at node A. Instead of imposing the velocity only inside the solid, the velocity is also imposed on nodes close enough to the wall.

5. Results

A periodic channel flow is selected to validate the methodology. The corresponding problem setup is depicted in the figure 3. In our computations, the

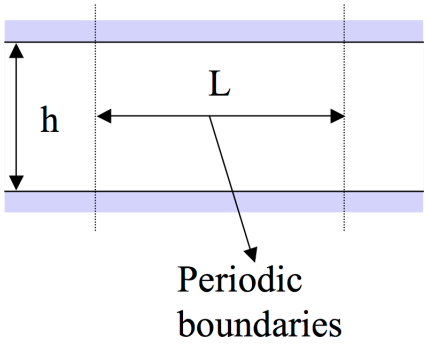


Figure 3: **Channel setup**

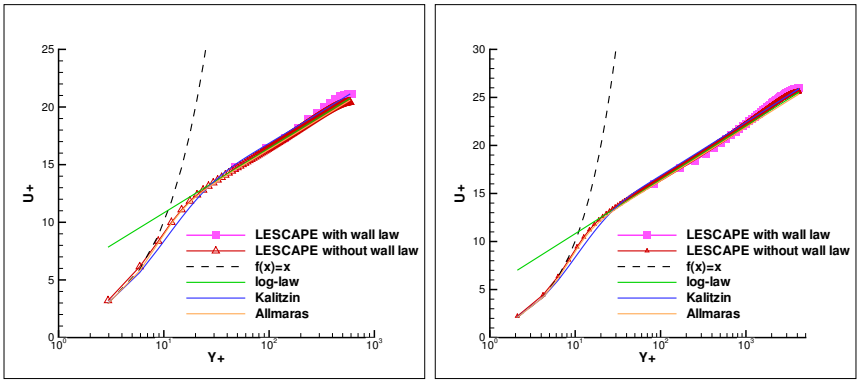


Figure 4: **Comparison between LESCAPE and theoretical log law, Left** $Re_\tau = 590$**, Right** $Re_\tau = 4200$

height of the channel is set to $h = 2$ m and the length is set to $L = 1$ m. On the left and right boundaries, periodic boundary conditions are imposed. The no-slip top/bottom walls are imposed via penalization, 10 nodes are penalized inside the wall with a zero velocity, consequently the computational domain corresponds to $[0\,;\,1] \times [-10\,dy\,;\,2 + 10\,dy]$. Numerical simulations corresponding to $Re_\tau = \dfrac{\rho u_\tau h/2}{\mu} = 590$ and $Re_\tau = 4200$ are presented in figure 4. LESCAPE numerical results are in very good agreement with the theoretical log law and with two wall functions from literature: the Kalitzin model [13] described in this paper and used into LESCAPE and the Allmaras one from [15].

For $Re_\tau = 590$ the numerical parameters are set to $\mu = 0.0017$ kg/ms,

$\rho = 1$ kg/m^3, with a mean velocity $u_{mean} = 18.45$ m/s and a corresponding Reynolds numbers $Re_{mean} = \dfrac{\rho u_{mean} h}{\mu} = 21\,705$. The mass flow rate inside the channel is imposed with the help of the stream function while the Poisson equation is solved: a differential of hu_{mean} is imposed between the top and bottom boundary conditions, ensuring the imposition of the proper mass flow rate through the channel. Without the wall function, the size of the mesh is set to $dx = dy = 5.e-3$ and the time step is fixed to $dt = 5.e-3$. With the wall function activated, the mesh is coarser $dx = 4.e-2$, $dy = 8.e-2$, and the time step is still $dt = 5.e-3$.

The results obtained with LESCAPE are compared to DNS results [16]. The velocity with wall function is closer to the DNS results than without wall function figures 5c and d. The vorticity distribution is well predicted by the simulation without wall function, figure 5b. With the wall function, as expected, the coarse mesh does not allow the vorticity to reach the appropriate maximum value, figures 5a and b. However, the wall function computes the proper shear velocity $u_\tau = 1$ as it should be for the imposed mean velocity and viscosity. If $dx = dy = 5.e-3$ is used with the wall function described by equation (24), the results are identical to those without wall function, although not shown here. Without the wall function, the SA turbulence model underestimates the maximum of the velocity inside the channel, figure 5c. This observation is not specific to LESCAPE, since a similar solution is observed using SA with OpenFOAM, figure 5d. The OpenFOAM case setup made available on Github by [17] is used to further investigate the SA model velocity prediction. The available case setup is modified to use the same mean velocity and viscosity as in the LESCAPE simulation. The OpenFOAM distance from the first node to the wall is $y+ = 0.26$. The velocity predicted in the center of the channel by OpenFOAM is nearly identical to the one of LESCAPE without wall function, as seen on figure 5d.

To complete the study and the verification of the numerical scheme, a numerical order study is performed on this test case at $Re_\tau = 590$. Four different meshes from $dx = dy = 2.e-2$ to $dx = dy = 2.5e-3$ have been used to

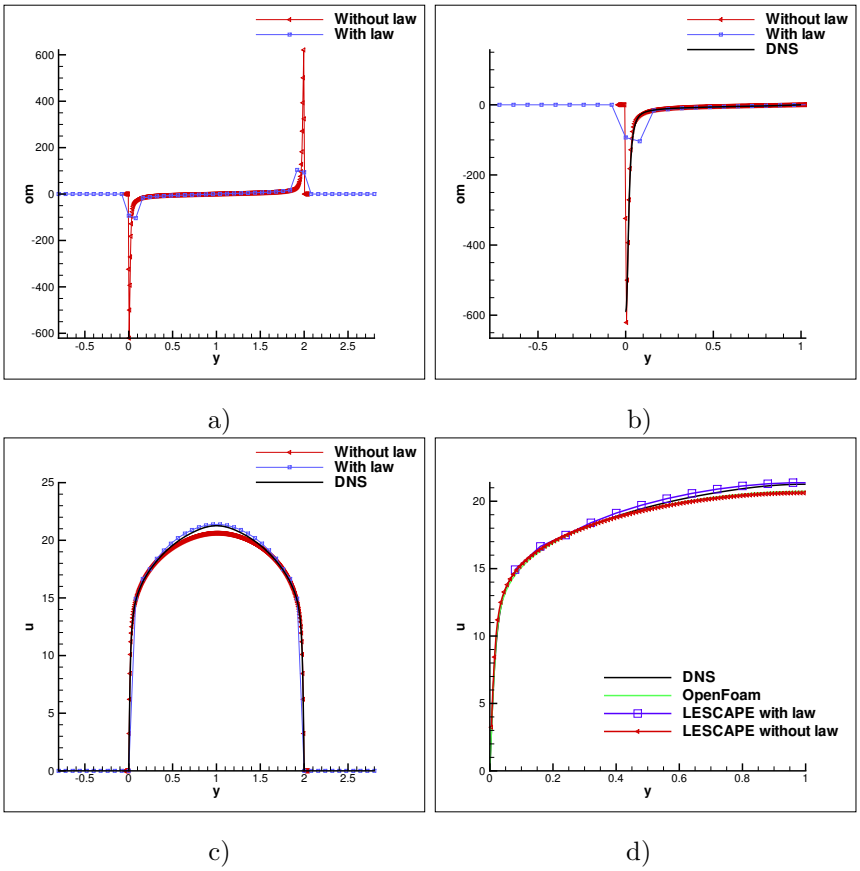


Figure 5: **Comparison between LESCAPE with and without wall function** $Re_\tau = 590$**; a) vorticity; b) zoom on vorticity; c) velocity profile compared to DNS results d) velocity profile compared to DNS and OpenFoam results [16]**

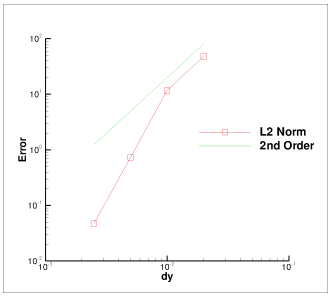


Figure 6: **Order study for a channel flow at** $Re_\tau = 590$ **without wall law**

perform the study. The reference solution has been obtained on a fifth mesh with $dx = dy = 1.125e-3$. No wall law has been used for this study and the same time step $dt = 1.e-4$ has been selected for all simulations. Figure 6 shows the evolution of the L2 norm of the error on the velocity u. The L2 norm evolution with mesh size is compared to the second order law. As expected, by construction, our numerical scheme is second order.

For $Re_\tau = 4200$ the numerical parameters are set to $\mu = 2.4e-4$ kg/ms, $\rho = 1$ kg/m^3, with a mean velocity $u_{mean} = 22.94$ m/s and a corresponding Reynolds number $Re_{mean} = 191\,167$. Without the wall function, the size of the mesh is set to $dx = dy = 5.e-4$ and the time step is fixed to $dt = 1.e-4$. With the wall function activated, the mesh is coarser $dx = dy = 4.e-2$ and the time step is set to $dt = 5.e-3$. The velocity results agree to DNS results [18], as presented in figure 7a. The vorticity distribution is well predicted by the simulation without wall function. With the wall function, as previously, the coarse mesh does not allow the vorticity to reach the appropriate maximum value. However, the wall function computes the appropriate shear velocity, still $u_\tau = 1$ for this case. If $dx = dy = 5.e-4$ is used with the wall function, the results are identical to those without wall function, although not shown here.

Finally, the wall function should be able to handle the case when the mesh is not aligned with the wall boundary. This is verified by dividing the computational domain in 44 intervals in the y direction, starting at $y = -0.5$ and ending at $y = 2.5$. This way, the level set 0 (dashed lines) that indicates the wall location is not aligned with the grid (solid lines), as shown on the figure 8. The resulting velocity profile for $Re_\tau = 590$ computed on this unaligned grid is compared, on the figure 9, with the one previously obtained (left part of figure 4) using a grid aligned with the wall. Both solutions are obtained using the wall function. The two solutions are almost identical, although the first point is not located at the same position. The DNS solution is shown for reference.

To check the extension of the methodology to any orientation of the wall with respect to the Cartesian grid, the flow through a channel bump has been performed. Figure 10a shows a sketch of the simulated geometry, a channel

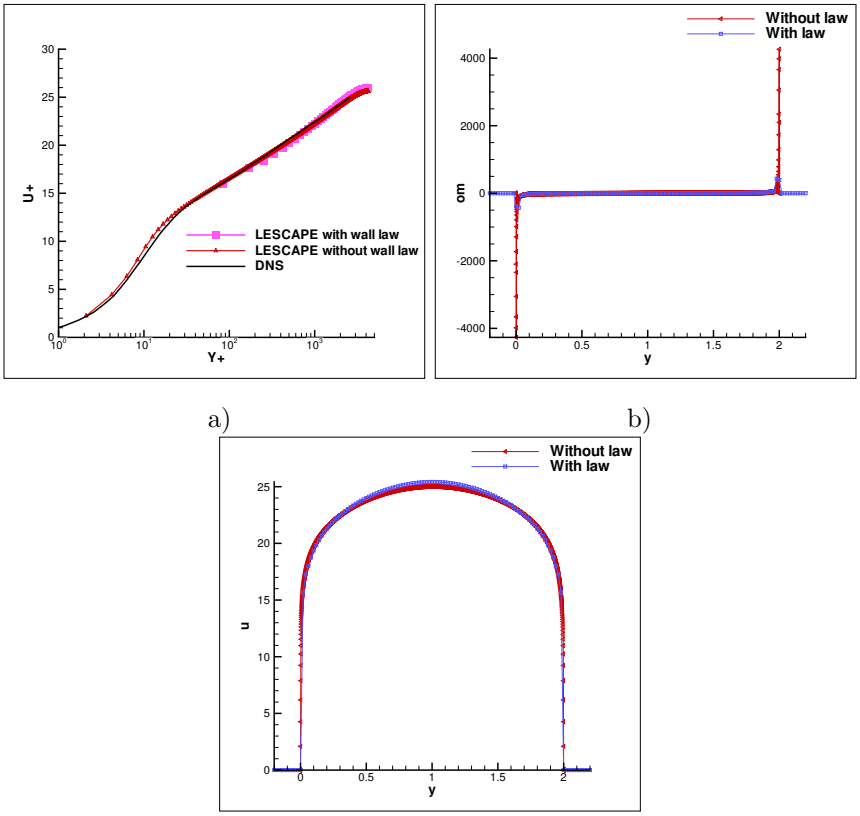


Figure 7: **a) comparison between DNS [18] and LESCAPE with and without wall function for $Re_\tau = 4200$, b) comparison of the vorticity with and without wall function; c) comparison of the velocity profile with and without wall function**

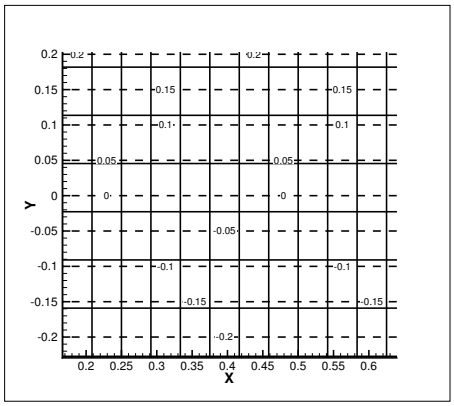


Figure 8: **Wall location, level set 0 not fitted with the mesh, (iso-line of the level set function in dashed lines, mesh lines in solid lines)**

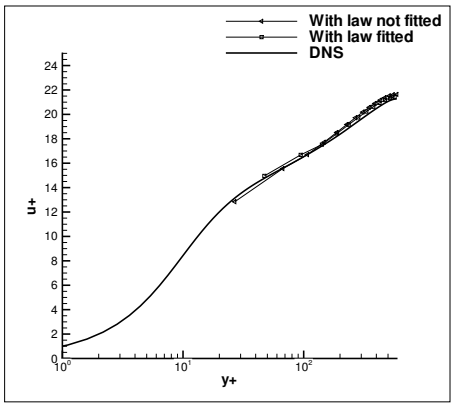
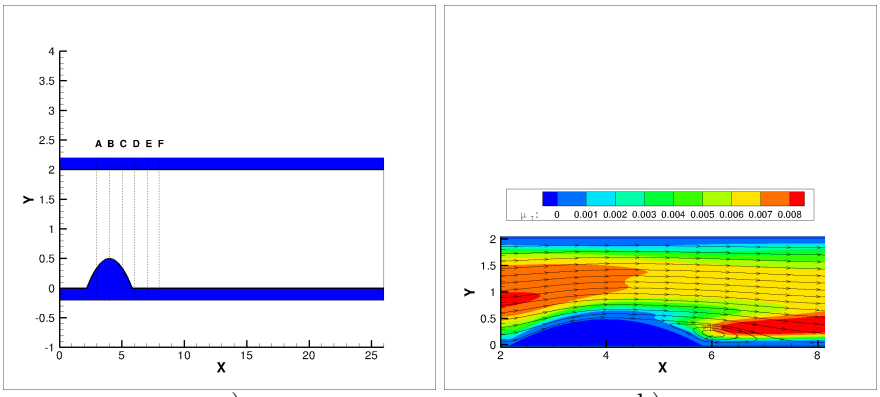


Figure 9: **Comparison between LESCAPE with wall function results for fitted and not fitted mesh**

with a curved lower wall referred to as bump, along with the 6 stations where comparisons with DNS results from Mollicone *et al.* [19] are performed. In Figure 10a, the penalization zone in the top and bottom appears colored, the black solid line represents the wall of the channel (iso-line 0 of the level set function). The channel dimensions are $(L_x \times L_y) = (26 \times 2)$ and the the bump shape is defined by $y = -0.15(x-4)^2 + 0.5$. The flow is from left to right and periodic boundary conditions are imposed in the streamwise direction. No slip



a) b)

Figure 10: **a) Sketch of the channel bump out of scale for clarity purpose. The penalization zone is in blue, the top and bottom channel walls correspond to the iso-line 0: the solid black line in this figure, dashed lines correspond to the 1D cut stations where comparison with DNS results are performed. b) Turbulent viscosity contours μ_t and streamlines for the channel bump flow.**

boundary conditions are imposed at the top and bottom walls. Compared to the bump width, the channel length is sufficient to allow a stabilization of the flow downstream and then to recover the typical turbulent channel flow behavior before re-entering the domain upstream due to periodicity. The bulk Reynolds number based on half the channel height is $Re_b = 2500$. Three different meshes have been used to perform the simulation $dx = dy = 2.e-2$ referred as 100 in Figure 11 (100 points inside the physical channel, we recall that for all our simulations in this paper 10 points are added on the top and bottom to define the penalization zone); $dx = dy = 1.e-2$ referred as 200; and $dx = dy = 5.e-3$ referred as 400. The time step is set to $dt = 1.e-3$ for all simulations, and the wall law has been used for the three simulations. Figure 10b shows the turbulent viscosity contours along with some streamlines of the flow close to the bump. Figure 11 presents the comparison between LESCAPE results and DNS results from Mollicone *et al.* [19]. The velocity results agree to DNS results [19] at each station, Figure 11. Mesh independence on the results is observed starting from a mesh spacing of $dx = dy = 1.e-2$ (200 points inside the physical channel),

Figure 11. As expected, the discrepancy between LESCAPE results and DNS results increases in the separation zone, since turbulence models and wall law are not suitable to handle perfectly this part of the flow [20]. Nevertheless, the curvature of the wall is correctly taken into account by the proposed method, and LESCAPE solution agrees with literature results.

6. Conclusions

The use of Spalart-Allmaras model with the penalized incompressible RANS equations has been successfully validated. The turbulence model has been used with the RANS equation recast as a vorticity transport equation. The space variation of the turbulent viscosity creates additional terms in the vorticity equation. These terms are obtained by taking the curl of the RANS equation with the turbulent shear stress expressed as a turbulent viscosity times the mean velocity derivatives. A VIC scheme is used to solve the resulting system of equations. Both the vorticity and the turbulent viscosity are transported by the same particle using the convective time step. A wall function is used near the wall to avoid fine meshes near the wall. The wall function is imposed through penalization of the near wall nodes. The model implementation is validated against periodic channel flow at various Reynolds numbers. The velocity and vorticity distribution into the channel agree well with log law relations and DNS results. Future work will consist in taking into account moving walls and derive a methodology to compute aerodynamic forces exerted on a solid.

7. Ackowledgement

- The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/20072013 under grant agreement no. 605180.

- Numerical experiments presented in this paper were carried out using MCIA (Mésocentre de Calcul Intensif Aquitain) facilities, the PLaFRIM

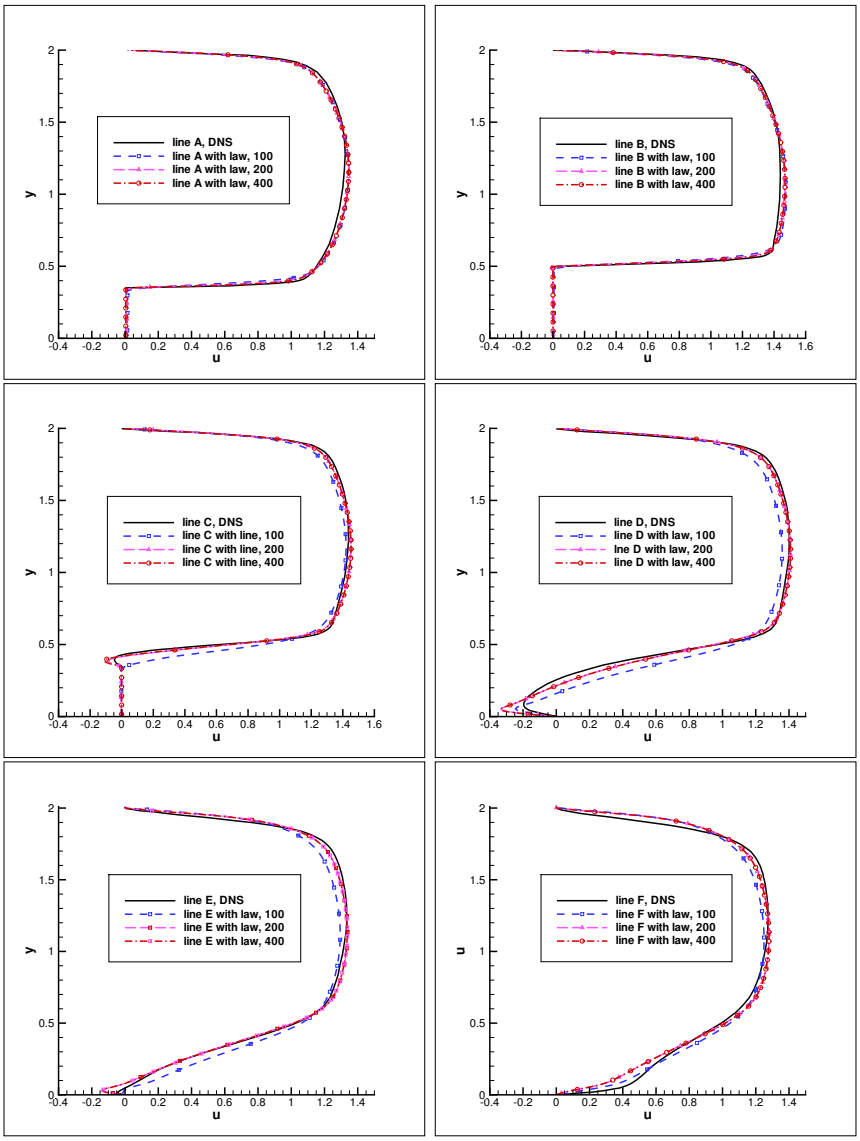


Figure 11: **1D cut of the velocity compared to the DNS solution from Mollicone** *et al.* **[19]**

experimental testbed, and the supercomputer Mammouth Parallèle 2 from Sherbrooke University, managed by Calcul Québec and Compute Canada. The operation of this supercomputer is funded by the Canada Foundation for Innovation (CFI), the ministère de l'Economie, de la science et de l'innovation du Québec (MESI) and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT). PLaFRIM cluster being developed under the Inria PlaFRIM development action with support from LABRI and IMB and other entities: Conseil Régional d Aquitaine, FeDER, Université de Bordeaux and CNRS, see https://plafrim.bordeaux.inria.fr/).

- This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the "Investments for the future" Programme IdEx Bordeaux - CPU (ANR-10-IDEX-03-02).

References

[1] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid-structure interaction, Progress in Aerospace Sciences 65 (2014) 1–21.

[2] H. Brinkman, A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particle, Appl Sci Res.

[3] G. Cottet, P. Koumoutsakos, Vortex methods: theory and practice, Cambridge Univ Pr, 2000.

[4] H. Beaugendre, F. Morency, F. Gallizio, S. Laurens, Computation of ice shedding trajectories using cartesian grids, penalization, and level sets, Modelling and Simulation in Engineering 2011 (2011) 1–15.

[5] F. Morency, H. Beaugendre, F. Gallizio, Aerodynamic force evaluation for ice shedding phenomenon using vortex in cell scheme, penalisation and level set approaches, International Journal of Computational Fluid Dynamics 26 (9-10) (2012) 435–450.

[6] H. Beaugendre, F. Morency, Innovative model for flow governed solid motion based on penalization and aerodynamic forces and moments, Report, INRIA Bordeaux, équipe CARDAMOM IMB (2015-04 2015).
URL https://hal.inria.fr/hal-01144855

[7] S. Allmaras, F. Johnson, P. Spalart, Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model, in: B. I. Edited (Ed.), proceedings ICCFD7-1902, 7th International Conference on Computational Fluid Dynamics, 2012.

[8] S.-J. Lee, Numerical Simulation of Vortex-Dominated Flows Using the Penalized VIC Method, Vortex Dynamics and Optical Vortices, InTech, 2017. doi:10.5772/65371.

[9] M. Coquerelle, G. Cottet, A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies, Journal of Computational Physics 227 (21) (2008) 9121–9137.

[10] G. J. Hokenson, Vorticity with variable viscosity, AIAA Journal 24 (6) (1986) 1039–1040. doi:10.2514/3.9384.
URL http://dx.doi.org/10.2514/3.9384

[11] J. Monaghan, Extrapolating B splines for interpolation, Journal of Computational Physics 60 (2) (1985) 253–262.

[12] T. J. Chung, Computational Fluid Dynamics, first edition Edition, Cambridge University Press, New York, NY, 2002.

[13] G. Kalitzin, G. Iaccarino, Turbulence modeling in an immersed-boundary RANS method, Report Annual Research Briefs 2002, Center for Turbulence Research (2002).

[14] F. Capizzano, Turbulent wall model for immersed boundary methods, AIAA Journal 49 (11) (2011) 2367–2381. doi:10.2514/1.56318.
URL http://dx.doi.org/10.2514/1.56318

[15] M. Berger, M. J. Aftosmis, S. R. Allmaras, Progress towards a cartesian cut-cell method for viscous compressible flow, in: 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2012, pp. AIAA 2012–1301.

[16] R. D. Moser, J. Kim, N. N. Mansour, DNS of turbulent channel flow up to $Re_\tau = 590$, Phys. Fluids 11 (1998) 943–945.

[17] A. Hedlund, Evaluation of RANS turbulence models for the simulation of channel flow, Master thesis (2014).
URL http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-238649

[18] A. Lozano-Duran, J. Jimenez, Effect of the computational domain on direct simulations of turbulent channels up to $Re_\tau = 4200$, Phys. Fluids 26.

[19] J.-P. Mollicone, F. Battista, P. Gualtieri, C. Casciola, Effect of geometry and reynolds number on the turbulent separated flow behind a bulge in a channel, J. Fluid Mech. 283 (2017) 100–133. doi:10.1017/jfm.2017.255.

[20] T. Knopp, F. Spallek, O. Frederich, G. Rapin, Application of numerical wall functions for boundary layer flows with separation and reattachment, New Results in Numerical and Experimental Fluid Mechanics X 132 (2016) 145–155.
URL http://elib.dlr.de/100181/