8. National Conference on Computational Mechanics
MekIT'15
B. Skallerud and H I Andersson (Eds)

ON VALIDATION AND IMPLEMENTATION OF AN IMMERSED BOUNDARY METHOD IN A HIGH ORDER FINITE DIFFERENCE CODE FOR FLOW SIMULATIONS

JØRGEN R. AARNES[1], NILS E. L. HAUGEN[1,2] AND HELGE I. ANDERSSON[1]

[1]Department of Energy- and Process Engineering
Norwegian University of Science and Technology (NTNU)
Kolbjørn Hejes vei 1b, 7491 Trondheim, Norway
e-mail: jorgen.r.aarnes@ntnu.no, web page: http://www.ntnu.no/ept

[2] SINTEF Energy Research
Kolbjørn Hejes vei 1c, 7491 Trondheim, Norway
e-mail: nils.e.haugen@sintef.no - Web page: http://www.sintef.no/home/sintef-energy

Key words: Computational Methods, Immersed Boundary, Computational Fluid Dynamics, Flow Past Cylinder, Validation, Compressible Flow

Abstract. An open source code using high-order finite differences for fluid flow simulations has been validated for use for flows past a stationary cylinder. The implementation of the immersed boundary method in the code has been considered. A new implementation is suggested, replacing orthogonal mirror points by mirror points along grid lines.

1 INTRODUCTION

As the cost of high computing power is ever decreasing, the use of high accuracy computations of fluid flow is no longer restricted to academia, but widespread in both research and engineering applications. The extent one trusts the results from computational fluid dynamic simulations can be illustrated by the way direct numerical simulations (DNS) of turbulence simulations are often referred to as numerical experiments, rather than simulations or numerical computations. The firm belief in computational results is not unfounded. Indeed, numerical simulations does not only compare very well with experimental results, they also make a large range of problems that cannot easily be studied experimentally available for the ever diligent researchers.

A problem may, however, arise when developed software is applied to research for which is has not been thoroughly tested and validated. The value of numerical results that are seemingly trustworthy will quickly decrease if a small change of the case studied, e.g.

a small increase of the domain size, has a large, unexpected influence on the computed results.

It is common to mention some means of validation, most often a grid refinement study, in papers concerning computational fluid dynamics. If such a study is enough to regard the computed results as reliable is up to the researchers publishing the results, and the readers of said study. Should the researcher include more validation results one will often regard the methods as better validated. However, a vast amount of validation results is not particularly interesting in itself, and too much will leave little room for the scientific study which was the authors motivation in writing the paper.

The aim of this paper is to present a thorough and systematic validation of the open source high order finite difference code known as the Pencil code [1] used to compute compressible flow around a circular cylinder at Reynolds number $\mathrm{Re} = 100$, where

$$\mathrm{Re} = \frac{U_\infty D}{\nu}, \tag{1}$$

with free stream velocity U_∞, cylinder diameter D and kinematic viscosity ν. The flow problem is a well-known benchmarking case. The validation is presented in detail as the code used is open for use by other researchers, whom may easily compare their validation to the results presented here. Further, it is the intent of the authors to make use of this software for simulating turbulent two-phase flow and the importance of understanding how different parameters affect the results cannot be overstated.

The Pencil Code is not a black box routine, but an open source code where all routines can be inspected in detail. In the process of validation, a closer look is taken on the way the fluid-solid interface in the flow problem is represented. This is done by the immersed boundary method with discrete forcing and ghost cells as described in, e.g., [2, 3]. The code uses 6th order central difference on all fluid points, and a three point deep ghost zone is therefore necessary in representing the solid interface. More details on the implementation of the immersed boundary implementation and a suggested method of improvement can be found in Sec. 3.

2 VALIDATION

In conducting a thorough validation meant to lay the foundation for further work for a specific case, the effect of varying several different parameters has be studied. The parameters include both physical properties of the system (domain size, Mach number) and numerical properties (grid spacing, Courant number). For all validation runs the drag coefficient

$$C_D = \frac{2F_D}{\rho U_\infty^2 D}, \tag{2}$$

lift coefficient

$$C_L = \frac{2F_L}{\rho U_\infty^2 D}, \tag{3}$$

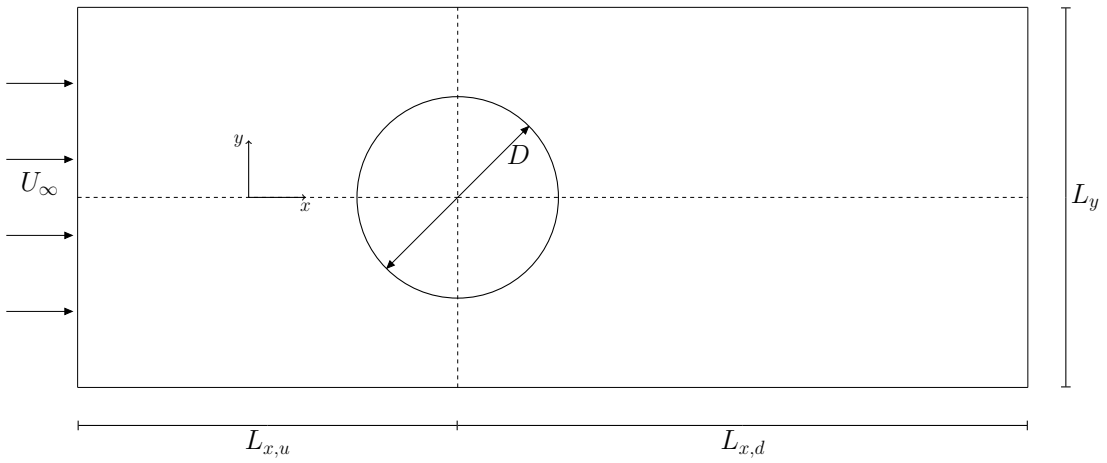

Figure 1: Definition of physical set-up. Cylinder is not to scale.

and Strouhal number

$$St = \frac{fD}{u}, \tag{4}$$

are computed, and used to compare the effect of tuning different parameters. The coefficients are expressed in terms of the free stream velocity U_∞, density ρ, cylinder diameter D, vortex shedding frequency f, drag force F_D and lift force F_L. The drag and lift forces are computed on a given number of forcepoints close to the cylinder surface. These forces vary with time, as the Reynolds number is sufficiently high to cause vortex shedding. Therefore, the mean drag coefficient and the root-mean-square lift coefficient (mean lift coefficient would be zero) are computed and used in the validation.

The physical set-up is a two-dimensional rectangular domain of length L_x and width L_y, as show in Fig. 1. The length can be split into the upstream length, $L_{x,u}$ and the downstream length, $L_{x,d}$, which need not be equal. Partially reflecting Navier-Stokes characteristic boundary conditions (NCSBC) (see [4]) are used at both the inlet and the outlet, with constant inflow velocity U_∞ at the inlet. In the spanwise direction periodic boundary conditions are used. One could argue that free-slip walls should also be tested, but as this is not appropriate for the author's further work with this set up it is not the focus of this study.

For all the validation runs the following properties are not altered: $D = 100 \cdot 10^{-3}$ m, $U_\infty = 1.0$ m/s and $\nu = 1.0 \cdot 10^{-3}$ m^2/s. All lengths in the system are non-dimensionalized in terms of the cylinder diameter D.

The results from the validations runs are plottet with error bars, such that the effect of the different parameters are more easily compared. The area spanned by an error bar is 1% (0.5% above and 0.5% below a point) of the "correct" value of the drag, lift and Strouhal number. Note that the correct value is not obtainable analytically, and the results found by Qu et al. [5] using body conformal grids and very large domains are used

for this purpose. For the largest domain used by Qu et al. (200D×200D) the values are $C_D = 1.310$, $C'_L = 0.2151$ and $St = 0.1647$ for incompressible flow. An alternative would be to use the asymptotical values at very large domains with body conformal grids, found by Posdziech and Grundermann [6]. They do, however, not include root-mean-square lift in their results, as Qu et al. do.

2.1 Grid refinement

The first parameter that is validated is the grid spacing. The grid is Cartesian, and the convective time step is computed for each time step by

$$\Delta t = C_{CFL} \frac{(\Delta x)_{\min}}{(|\mathbf{u}| + c_s)_{\max}}, \tag{5}$$

where C_{CFL} is the Courant number, $(\Delta x)_{min} = \min(\Delta x, \Delta y, \Delta z)$, $\mathbf{u}$ is the fluid velocity at a grid point and c_s is the sound speed. Hence, the time-step is proportional to the smallest grid spacing in the system. This motivates choosing the grid spacing equal in the x- and y-direction, $\Delta x = \Delta y$. The remaining parameters are chosen rather arbitrarily, some of them perhaps overly cautiously others not strict enough, as will be shown in the validation to come. These initial results are purely of qualitative interest, to find a sufficiently fine grid, not to find quantitative information about the drag, lift and Strouhal number. The upstream and downstream length are chosen equal, and the length and width of the system are set to $20D$ and $10D$, respectively. The flow is weakly compressible with a Mach number set to 0.05. The Courant number is set to 0.4.

In Fig. 2 it can be seen that the resolution has little effect on the Strouhal number. The effect on the rms lift and the drag coefficient are comparable for $D/\Delta x \geq 25$ (for $D/\Delta x < 25$, however, the rms lift has a large drop, making the similarity between C_D and C'_L less than obvious in the figure). The oscillatory behaviour for both these quantities for $D/\delta x < 64$ is an unwanted effect. Thus, the grid spacing should by this initial grid refinement study be chosen such that $D/\Delta x \geq 64$ to avoid this. Hence, quite a large number of grid points are necessary to compute accurate results, especially when the domain size is increased (assuming that the grid refinement is close to independent of the domain size). The use of a stretched grid should therefore be considered in order to reduce the resolution without using a coarser grid close to the cylinder surface.

2.2 Mach number

The Mach number, defined here as

$$\mathrm{Ma} = \frac{U_\infty}{c_s}, \tag{6}$$

has a large impact on the time-step in the simulations of compressible fluid flow. As the Mach number is reduced to approach the incompressible limit, the time step is also

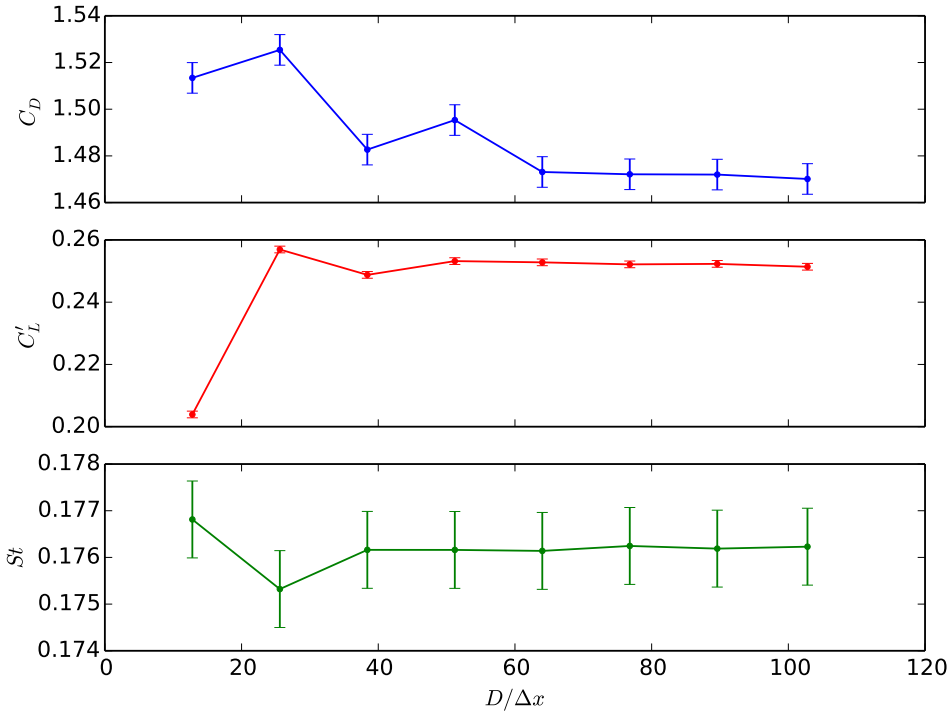

Figure 2: Computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number for different grid spacing.

reduced. It is therefore necessary to consider the qualitative impact the Mach number has on the results, such that the effects are known for Mach numbers that are convenient to use in the simulations. All Mach numbers considered here are for subsonic flow. The domain size and Courant number is equal to that in the grid refinement simulations. The grid spacing is set to $D/\Delta x = 64$.

Figure 3 depicts the results from runs with varying Mach number. It can be seen that once again the Strouhal number is much less affected by the parameter variation, than the drag and lift coefficients. In contrast to the grid refinement results the results in Fig. 3 vary in a smooth, non-oscillatory fashion, approaching the incompressible limit as the Mach number is decreased. This is as expected, as for most flows no important changes is observed when the Mach number is reduced from 0.2 to 0.01 [7] (not reduced below 0.025 in the present computations). From the results, one should expect less than 1% deviation from the incompressible limit, when the Mach number is set to 0.1. Thus, setting $\mathrm{Ma} = 0.1$ is a reasonable trade-off between accuracy and computational cost.

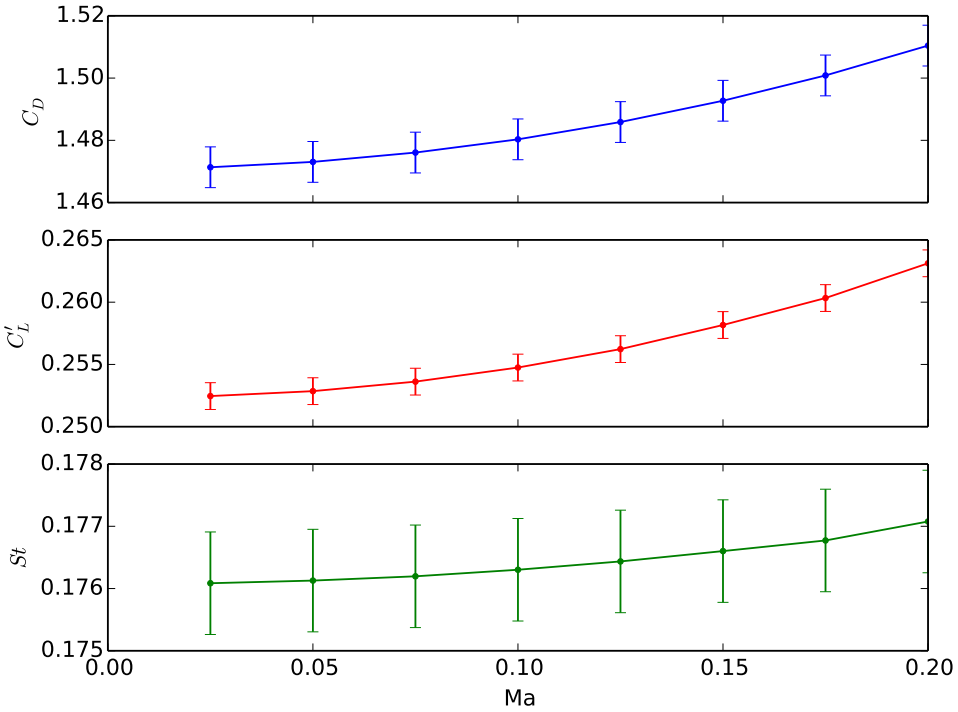

Figure 3: Computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number for varying Mach number.

2.3 Domain size

While both grid refinement and varying the Mach number is fairly straightforward, looking at the effect of changing the domain size requires some more thought. Not only should one expect that the three parameters $L_{x,u}$, $L_{x,d}$ and L_y affect the computed results, when varied one at the time, one might also expect that the variations are not independent (as assumed with grid spacing and Mach number variations). Hence, varying a single parameter at a time is not necessarily enough to fully understand how the domain size affect the computed quantities.

Consider first the variation of system width, where the results are depicted in Fig. 4. These results are computed with $D/\Delta x = 64$, $C_{CFL} = 0.4$, Ma $= 0.1$ and $L_{x,u} = L_{x,d} = 10D$. It is clear that the width of the system has a large impact on the computed coefficients, and it is the first parameter this far that has more than a modest impact on the Strouhal number. This far the system width has been set to $L_y = 10D$. The results in Fig 4 indicates that this is, by far, a too narrow domain to obtain results that quantitatively accurate. However, as the results – in a similar manner as the results from the Mach number variation – drop quite smoothly towards an asymptotic limit,

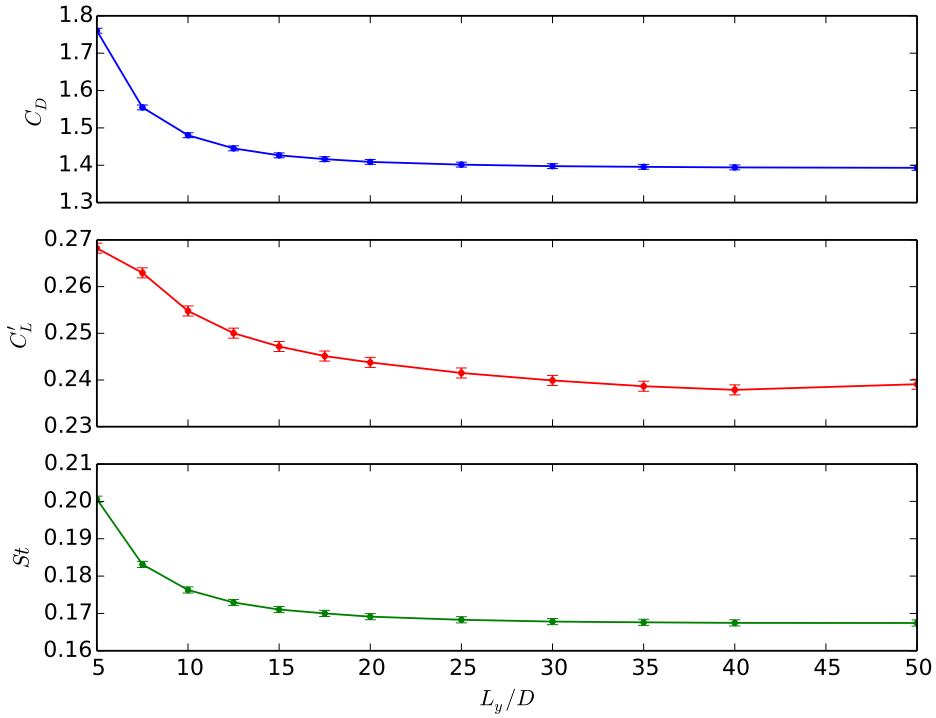

Figure 4: Computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number for varying upstream length (blue) and downstream length (red).

the qualitative behaviour in the system may be satisfyingly resolved with such a domain width.

The computed drag coefficient, lift coefficient and Strouhal number for varied streamwise lengths are depicted Fig. 5, where the blue and red curves represent the results for varied upstream and downstream lengths, respectively. The results are computed with $D/\Delta x = 64$, $C_{CFL} = 0.4$, Ma $= 0.1$ and $L_y = 10D$. When the upstream length is varied the downstream length is set to $10D$, and vice versa. Hence, the curves overlap exactly at the point where the varied parameter is $10D$. The curves also overlap to a large extent for the computed Strouhal number for $L_{x,u}, L_{x,d} \geq 10D$, which is affected very little by the upstream and downstream lengths.

The way the mean drag and rms lift coefficient vary, on the other hand, is not as expected. Both show a significant drop as the upstream and downstream lengths are increased from $5D$ to $10D$. This trend does not, however, extend towards an asymptotic limit, as was the case for increasingly large domain width. Consider the mean drag coefficient for increasingly large downstream lengths (red curve in top window of Fig. 5). The curve shows some oscillations for small values of $L_{x,d}$, before it decreases steadily. For $L_{x,d} > 35D$, however, there is an unexpected increase in the computed drag coefficient.

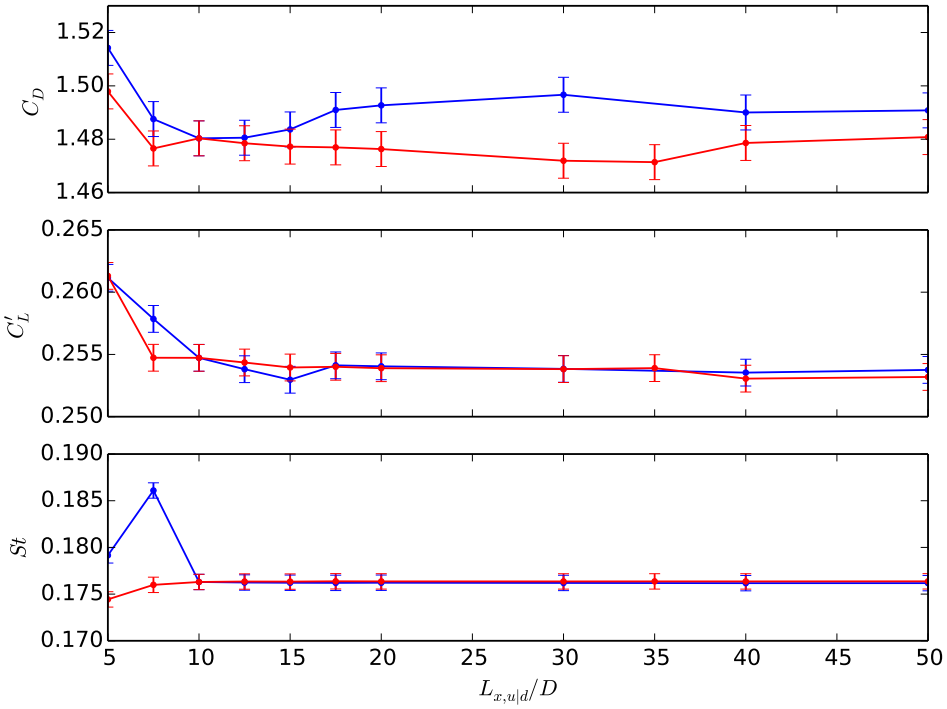

Figure 5: Computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number for varying domain width.

This increase is not only apparent in the results computed at $L_{x,d} = 40D$, there is a further increase for $L_{x,d} = 50D$. The reason for this behaviour, and the similar behaviour for the variations with upstream length, is not obvious. It is probable that it is related to some blocking effect due to the domain being too small for the flow problem, either in the spanwise or the streamwise direction, or both.

To understand the behaviour of the computed drag coefficient in Fig. 5 a few more validation runs for varying domain size are performed. The downstream length is varied for three different parameter configurations. One with $L_{x,u} = 10D$ and $L_y = 10D$ (the same as before), one with $L_{x,u} = 20D$ and $L_y = 10D$ and one with $L_{x,u} = 10D$ and $L_y = 20D$. The results are seen in Fig. 6.

Several aspects of the domains influence on the computed mean drag coefficient are apparent in the figure. Consider first the difference between the two red curves. The upper red curve is equal to the computed drag for varying downstream length in Fig. 5, while the bottom red curve depicts the results with doubled domain width. The two curves are, qualitatively, very similar; the bottom curve is only shifted downwards by approximately 0.075. This shift – towards a more accurate result for the computed drag coefficient – can also be seen in Fig. 4, when comparing the computed mean drag coefficient for $L_y = 10D$

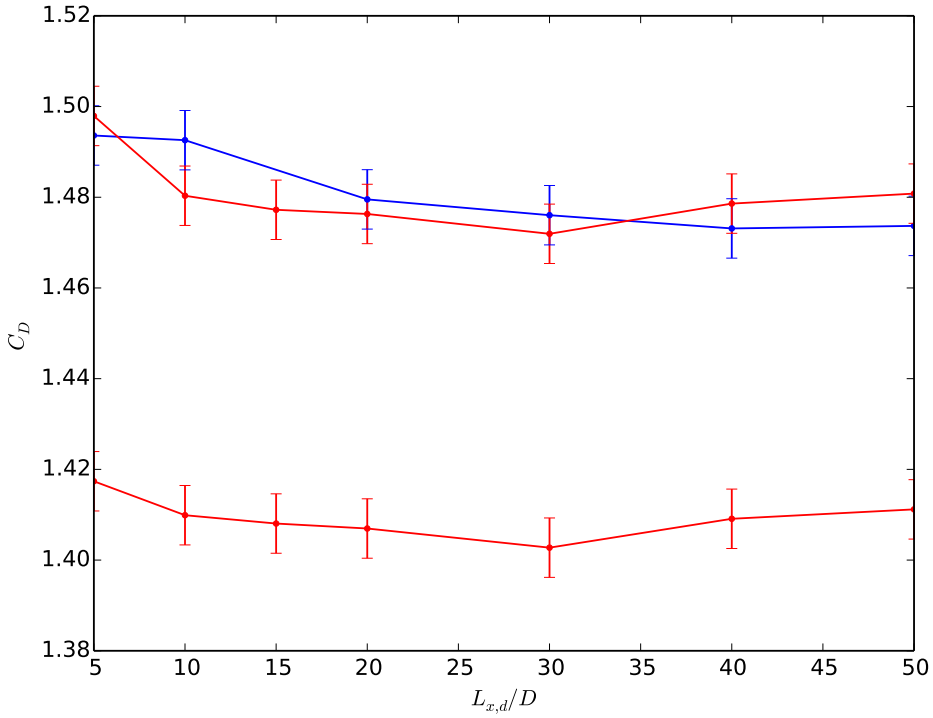

Figure 6: Computed mean drag coefficient for varying downstream length, for three parameter configurations: $L_{x,u} = 10D$ and $L_y = 10D$ (top red), $L_{x,u} = 20D$ and $L_y = 10D$ (blue), $L_{x,u} = 10D$ and $L_y = 20D$ (bottom red).

and $L_y = 20D$. This confirms that the blockage effects from a narrow width has a large impact on the quantitative results and a very modest impact on the qualitative results.

Comparing the top red curve with the blue curve, for which the upstream length has been doubled shows that a possible reason for the unexpected jagged form of the drag coefficient computed with $L_{u,x} = 10D$ is a blockage effect due to a too short upstream length. This is in accordance with [6] that advise against using domain with smaller upstream lengths than $20D$. The blue curve shows the expected behaviour of dropping towards an asymptotic value for increasingly large downstream lengths. Thus, to achieve the expected qualitative behaviour of increasing the domain in the downstream direction the upstream direction must be larger than some threshold value. It is expected that this statement also holds for the an increase in the upstream length, being dependent on a sufficiently large downstream length to show the expected asymptotic curve of the mean drag and rms lift coefficients.

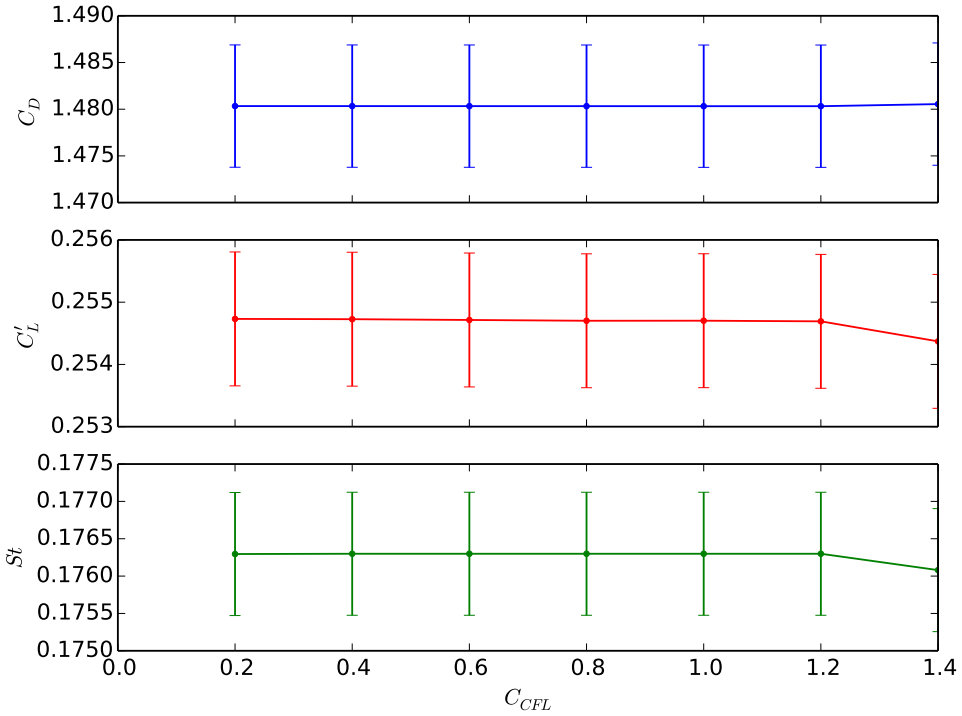

Figure 7: Computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number for different Courant numbers.

2.4 Courant number

The Courant number is a numerical constant that sets restrictions on the time step (Eq. (5)) to maintain stability of the explicit time discretization. One often tries to avoid a larger Courant number than one, which is related to the stability limit found by von Neumann analysis of the one-dimensional advection equation discretized by the upwind scheme. An even lower Courant number is often used chosen, to ensure that the simulations are well within the stability region. This is not necessarily a good idea, as the time step will become small, and a small Courant number may impact the computations of the flow at small length scales in a negative way. Indeed, for higher order discretization schemes a Courant number larger than unity may be permitted (as shown in, e.g., [8] where $C_{CLF} = 1.43$ is the stability criteria), though they may not be a good idea as certain scales of the solution may be affected by such a choice.

Figure 7 depict the computed drag and lift coefficients and the Strouhal number for different Courant numbers. The constant parameters are $D/\Delta x = 64$, Ma $= 0.1$, $L_{x,u} = L_{x,d} = 10D$ and $L_y = 10D$. It can be seen that the Courant number has no significant effect on the computed quantities for $C_{CLF} \leq 1.2$. This does not mean that $C_{CLF} = 1.2$ is the optimal choice of Courant number for all simulations with this set-up. If, e.g.,

forced turbulence is introduced or the Reynolds number is increased such that turbulence is generated in the flow, the small scale structures will have a significant impact on the result. This may influence the choice of Courant number as the Courant number influences the small scale error.

With $C_{CLF} = 1.4$ the computations are close to becoming unstable, and the computed lift and drag show some deviations from sinusoidal behaviour. A run with $C_{CLF} = 1.6$ was also initiated, but did not complete successfully, as instability produced NaNs in the solution.

2.5 Comparison to data sets from previous studies

The impact of variation of different parameters of interest has been studied in detail. To conclude the validation part of this paper a few runs for a large domain have been computed, and are compared to previous studies. Table 1 show computed values for mean drag coefficient, rms lift coefficient and Strouhal number, from several different previous studies as well as the present one. The extent the previous studies are comparable to the present one varies from study to study. All studies are assumed grid independent, and only the domain size is given for comparison. While some studies use the rms lift as a control parameter, others use the amplitude of the lift. This amplitude has been scaled by 0.707 to an approximate rms-value. This may be done as the lift coefficient is a smooth sinusoidal-like function with zero mean value.

The present results have been computed on a rectangular domain of width $L_y = 60D$, upstream length $L_{x,u} = 20D$ and downstream length $L_{x,d} = 40D$. The flow is compressible with Mach number Ma = 0.1 and the Courant number is set to $C_{CLF} = 1.0$. The grid is equidistant with grid spacing $\Delta x = D/60$. In addition, two numerical parameters have been altered, as compared to the previous runs. One is a scaling parameter used in the computations of the drag force and lift force, and one is a parameter related to the fluid points very close to the cylinder surface. The latter (*linear_close_interpolate*) will be discussed further in Sec. 3.

The studies in Tab. 1 represent a wide spread of computational fluid dynamic methods for computing the flow past a cylinder. The top seven and the present study use the immersed boundary method to represent the cylinder in the flow, yet they use quite different domain sizes. The eight remaining do not use the immersed boundary method, but different body conformal grids. Finite-volume, finite-difference, finite element, spectral element and latice-boltzmann methods are represented in the table. Only Haugen & Kragset, Lie et al. and the present study are for compressible flows, while the remaining are computations for incompressible flows. The study by Haugen & Kragset uses the Pencil Code, the same software that is used in the present study, but with a different domain size and resolution. The studies by Qu et al. and Posdziech & Grundmann are intended to compute the coefficients to high accuracy, for very large domains, meant to give qualitative results useful for, e.g., benchmarking purposes. Only the domains most similar to the domain used in the present study are included in Tab. 1. Two of the do-

Table 1: Comparison of data sets from previous studies. Root-mean-square lift coefficients marked by a superscript star ($^{(*)}$) denote lift coefficient amplitudes scaled to root-mean-square values. The non-rectangular grids are marked as circular inlet/C-type (◖) or circular/O-grid (○). Domains where the cylinder is not centred have both upstream and downstream length given.

	$[(L_{x_u} + L_{x_d}) \times L_y]/D^2$	C_D	C'_L	St
Lai & Peskin [9]	$(6.1 + 20.5) \times 26.6$	1.4473	$0.233^{(*)}$	0.165
Kim, Kim & Choi [10]	70×100	1.33	$0.22^{(*)}$	0.165
Su, Lai & Lin [11]	$(13.4 + 16.5) \times 16.7$	1.40	$0.240^{(*)}$	0.168
Pan [12]	60×60	1.32	$0.226^{(*)}$	0.16
Tseng & Ferziger [3]	32×16	1.42	0.29	0.164
Noor, Chern & Horng [13]	–	1.4	–	0.167
Haugen & Kragset [14]	70×35	1.328	–	0.166
Park et al. [15]	$(50 + 20) \times 100$, ◖	1.33	$0.235^{(*)}$	0.165
Shi et al. [16]	100, ○	1.331	–	0.1650
Mittal [17]	100×100	1.322	0.226	0.1644
Stålberg et al. [18]	160, ○	1.32	$0.233^{(*)}$	0.166
Posdziech & Grundmann [6]	$(20 + 50) \times 40$ ◖	1.3504	$0.234^{(*)}$	0.1667
Posdziech & Grundmann [6]	$(40 + 50) \times 80$ ◖	1.3321	$0.229^{(*)}$	0.1650
Li et al. [19]	100×100	1.336	–	0.164
Qu et al. [5]	60×60	1.326	0.2191	0.1660
Present	$(20 + 40) \times 60$	1.334	0.227	0.1658

mains used by Posdziech & Grundmann are close to the present domain size, and both are included in the table.

It is seen that the computed results are in very good agreement with previous studies. A few of the studies using the immersed boundary method are for smaller domains than the high accuracy runs in this study. These results compare well with the results for similar domain sizes, as depicted in Figs. 4–6. The validations performed does not only show how the different parameters affect the solution, but also that the solution is in good agreement with literature.

The results computed with the Pencil Code can be regarded as highly accurate. The results are, however, achieved by using a very fine grid, something that may prove troublesome if the computed case is to be generalized to three dimensions. The critical point when regarding the grid spacing in the set-up is the resolution of the points closest to the cylinder surface. Therefore, the way the fluid equations are solved here is studied in detail to find possible improvements to the boundary layer representation.

3 IMMERSED BOUNDARY METHOD

The immersed boundary method was introduced by Peskin in the 1970s to model flow around heart valves [20]. Today it is a class of methods that represent a boundary immersed in a flow with non-body conformal grids [2]. As the grid does not conform to

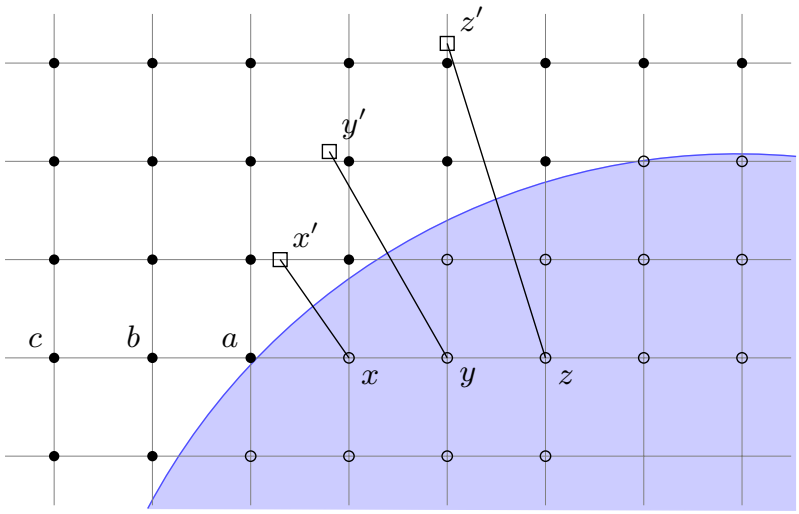

Figure 8: Ghost points set by corresponding mirror points at orthogonal distance from solid surface. Fluid points (a,b,c; ●), ghost points (x,y,z; ○) and a mirror points (x,y,z; □).

the solid boundary, incorporating the boundary conditions require a modification to the equations in the vicinity of the boundary. In the Pencil Code, this is done by what is called a discrete forcing approach. In this way of implementing the immersed boundaries the flow equations are first discretized on a Cartesian grid without regard to the immersed boundary, and then forcing terms that represent the boundaries' effect on the flow are introduced. For details on the grid representation close to the boundaries the reader is referred to [14]. Only the main outline will be repeated here.

3.1 Orthogonal mirror point method

To avoid special handling of the grid points close to the solid surface, a ghost-cell zone is constructed inside the solid surface. The ghost-cell zone is three grid points deep, contrary to the much used single ghost-point in discrete forcing immersed boundary methods to ensure that the boundary conditions are met (see, e.g., [3]). Three points are necessary to make the sixth order central differencing scheme applicable at the grid points close to the solid surface as well as far from the boundaries. The ghost-cells are given values by using mirror points outside the solid surface. Figure 8 illustrate the relation between mirror points and ghost points for three ghost points.

The way the mirror points are set up using lines orthogonal to the interface makes satisfying the boundary conditions fairly straightforward. However, the distance from a computed fluid point close to the immersed boundary to the mirror points used to generate ghost points necessary to compute the flow variables at the fluid point can be large. This may have a negative influence on the accuracy of the computations. This effect will be

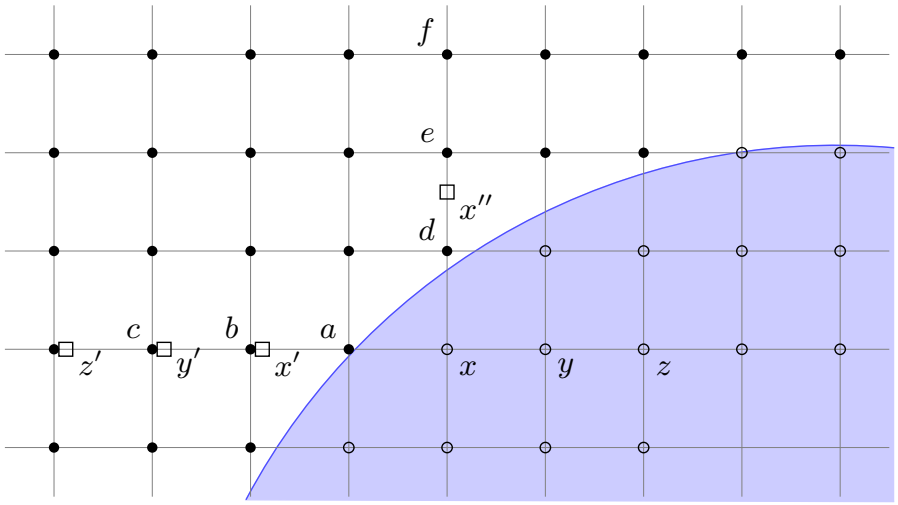

Figure 9: Ghost points set by corresponding mirror points at orthogonal distance from solid surface. Filled circles denote a fluid points, circles denote a ghost points and a squares denote mirror points.

greatest on the grid points nearest to the cylinder surface, as they make use of three grid points inside the solid geometry. This is illustrated in Fig 8, where the relation between the fluid point a close to the cylinder surface and the position of the mirror point used to set the value at the ghost points d can be seen. It is possible to handle these points with as exceptions. A parameter *limit_close_linear* in the Pencil Code lets the user choose the distance from the cylinder that qualifies for special handling – using interpolation between the surface and the neighbouring point to compute the flow variables, rather than the high order finite difference method (detail in [14]). The default distance is half a grid cell (thus fluid point a in Fig. 8 would be handled as an exception). This distance is used in all the validation runs in Sec. 2, except for the simulations compared with previous studies, where a larger distance has been used (this gave favourable results).

3.2 Cartesian mirror point method

A new implementation of the immersed boundary method in the Pencil Code is suggested. The method makes use of the Cartesian grid by setting local mirror points along the grid lines. This minimises the distance from a fluid point to the mirror points used when setting the ghost-cell values. The method is quite straightforward for the velocity components in the flow, but requires special handling of the density function to satisfy a zero gradient in the radial direction. In addition some ghost points will have to be computed several times – once for each direction for which they are to be used. The concept is illustrated in Fig. 9.

Consider a computation of the fluid point b. The ghost points x and y are needed (in

addition to three fluid points to the right and the left fluid point a) in the 6th order central differencing scheme. The mirror points used in the computation of the fluid velocities in the ghost-points x and y are x' and y', respectively. As the mirror points x' and y' are much closer to b than the corresponding mirror points in the orthogonal mirror point method (Fig. 8) it is expected that the error from the ghost point representation in the central differencing scheme will be reduced.

The fluid velocity vector in the mirror points is computed by interpolation between the two nearest grid points and the surface point on this grid line. Hence, when computing x' in Fig. 9 the points b and a are used, as well and the surface point to the left of a. When computing y' points c, b and the surface point to the left of a is used in the interpolation, etc. The surface points can be used for the velocity interpolation since non-slip and impearmeability conditions at the surface require all velocity components to be zero there. Such conditions do not apply to the density, for which all that is known is that the radial component of the density vector is zero at the surface. Due to this, the new immersed boundary method is more of a hybrid method than a pure Cartesian grid method. The orthogonal mirror points shown in Fig. 8 are constructed in this scheme as well, yet only to set the density in the ghost-points in the new method.

The mirror point x'' that is included in Fig. 9 is used when computing the velocity vector in x for computation of the y-components of d, e and f. This drawback, the necessity to compute the value in some ghost points several times, is not expected to have a large impact on the computational cost of the immersed boundary implementation.

This new immersed boundary implementation is expected to improve the accuracy of the computed flow variables, such that high accuracy results can be computed with a coarser grid than that used in the validation runs. If these expectations are met, this will not only negate the added cost of decomposition and additional ghost points computations, but reduce the overall simulation time significantly.

3.3 Results

Figure 10 depict mean drag coefficient, rms lift coefficient and Strouhal number for varying grid size computed with both two different implementations of the immersed boundary method in the Pencil Code. The blue and red curves show the results for the Cartesian mirror point method (new implementation) and the orthogonal mirror point method, respectively. The computations have been carried out with $L_y = 10D$, $L_{x,u} = L_{x,d} = 10D$, $Ma = 0.1$ and $C_{CFL} = 0.9$. The scaling parameter and the parameter related to the treatment of fluid points close to the solid surface (both mentioned in Sec. 2.5) have been set to the same values as those used in Sec. 2.1–2.4.

It is seen that performing a grid refinement study with the new implementation of the immersed boundary method yields less oscillatory behaviour when going from one grid size to the next. Further, it can be seen that all three computed quantities, C_D, C_L' and St reach a point where decreasing the grid spacing has only a small effect. As a matter of fact, the variations in the computed results, for all three quantities, is within

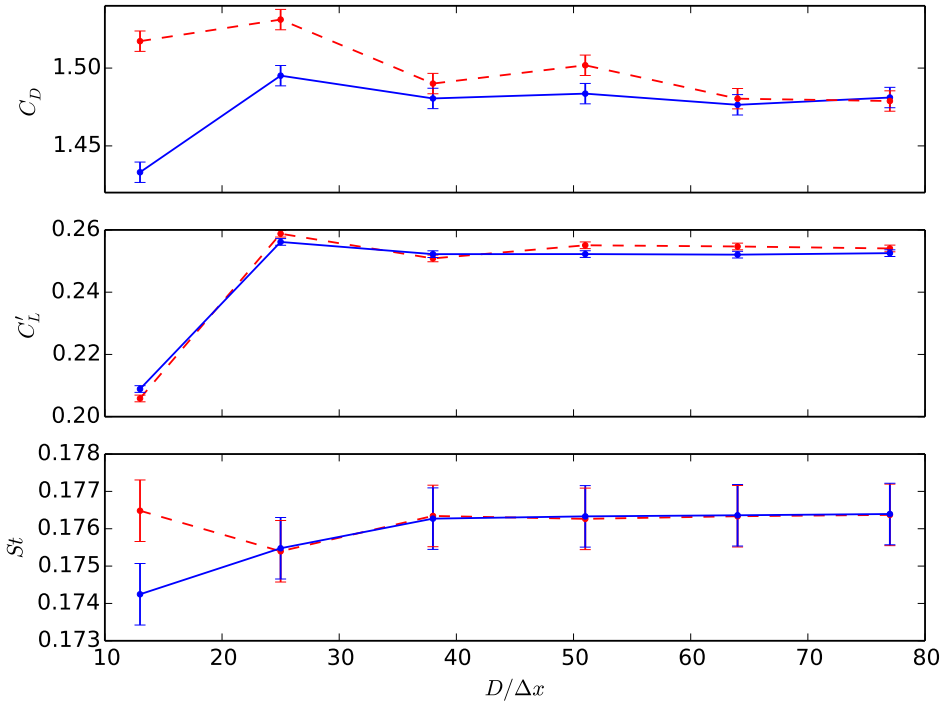

Figure 10: Computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number for different Courant numbers. Two different implementations of the immersed boundary method have been used to get the results. The Cartesian mirror point method (blue) and the orthogonal mirror point method (red, dashed).

the error bounds (0.5% in each direction, of reference value) for $D/\Delta x \geq 38$. This is a substantial improvement over the grid refinement performed in Sec. 2.1, indicating that the new immersed boundary method does indeed yield more accurate ghost points within the cylinder. The extra overhead in the computations, due to some ghost points being computed several times, is has a negligible impact on the computational time compared to increasing the grid spacing by a factor of 1.7, when going from $D/\Delta x = 64$ to $D/\Delta x = 38$ (recall that the grid spacing not only affects the spatial resolution, but also the time step used in the computations).

4 CONCLUSION

Thorough validation for the case of computing flow past a stationary cylinder has been conducted. In the validations the effect of the grid spacing, Mach number, domain size and Courant number on the computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number was studied.

Out of the different parameters tuned in during the validation, only the grid spacing

and the domain size behaved somewhat unexpected. The computed results showed an oscillatory behaviour for different grid sizes, resulting in the choice of a quite fine grid for the remaining validation runs ($D/\Delta x = 64$). When considering the domain size, the upstream and downstream lengths showed dependent behaviour. This indicated that a minimum length in the streamwise direction is not itself sufficient in computing high accuracy results, the cylinder placements in the domain must be such that the upstream and downstream lengths are both sufficiently large ($> 20D$). Of the remaining parameters, both the domain width and the Mach number indicated smooth decrease towards an asymptotic value for the computed quantities. The Courant number had very little effect on the computed values, for $C_{CFL} \leq 1.2$.

The results computed with the Pencil Code compared very well to previous studies. The computations of high accuracy results was, however, very computationally costly, on the fine grid used in the computations. This motivated a closer look at the flow computations closest to the solid cylinder, where the immersed boundary method with a ghost-zone is used in the Pencil Code.

A new implementation of the immersed boundary method was suggested. The method improves on the way the ghost-points inside the cylinder are computed, by making use of the grid lines in computing the fluid flow on corresponding mirror points. Such an improvement greatly reduced the oscillatory behaviour seen of the grid refinement results, indicating higher accuracy of boundary layer around the cylinder. The increased accuracy allows for an increase in the grid spacing of a factor of 1.7 with negligible impact on the accuracy.

The presented study lays a solid foundation for further work with flow past a circular cylinder using the high order finite difference Pencil code. As the impact of many central parameters is well understood, the choice of the these parameters – which is always a compromises between accuracy and computational cost – can be made without too much guesswork and hand-waving arguments. Further, the improved accuracy close to the solid surface will be beneficial in the authors' further study with this set-up, and for other users of the open source code that include immersed solid objects in their flow simulations.

REFERENCES

[1] The Pencil Code. http://www.nordita.org/software/pencil-code, 2015.

[2] R. Mittal and G. Iaccarino, "Immersed Boundary Methods," *Annual Review of Fluid Mechanics*, vol. 37, no. 1, pp. 239–261, 2005.

[3] Y.-H. Tseng and J. H. Ferziger, "A ghost-cell immersed boundary method for flow in complex geometry," *Journal of Computational Physics*, vol. 192, no. 2, pp. 593–623, 2003.

[4] T. J. Poinsot and S. Lele, "Boundary conditions for direct simulations of compressible viscous flows," *Journal of Computational Physics*, vol. 101, no. 1, pp. 104–129, 1992.

[5] L. Qu, C. Norberg, L. Davidson, S.-H. Peng, and F. Wang, "Quantitative numerical analysis of flow past a circular cylinder at Reynolds number between 50 and 200," *Journal of Fluids and Structures*, vol. 39, pp. 347–370, 2013.

[6] O. Posdziech and R. Grundmann, "A systematic approach to the numerical calculation of fundamental quantities of the two-dimensional flow over a circular cylinder," *Journal of Fluids and Structures*, vol. 23, no. 3, pp. 479–499, 2007.

[7] R. H. Pletcher, J. C. Tannehill, and D. Anderson, *Computational fluid mechanics and heat transfer*. CRC Press, 2012.

[8] P. Moin and K. Mahesh, "DIRECT NUMERICAL SIMULATION: A Tool in Turbulence Research," *Annual Review of Fluid Mechanics*, vol. 30, no. 1, pp. 539–578, 1998.

[9] M.-C. Lai and C. S. Peskin, "An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity," *Journal of Computational Physics*, vol. 160, no. 2, pp. 705–719, 2000.

[10] J. Kim, D. Kim, and H. Choi, "An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries," *Journal of Computational Physics*, vol. 171, pp. 132–150, July 2001.

[11] S.-W. Su, M.-C. Lai, and C.-A. Lin, "An immersed boundary technique for simulating complex flows with rigid boundary," *Computers & Fluids*, vol. 36, pp. 313–324, Feb. 2007.

[12] D. Pan, "An Immersed Boundary Method on Unstructured Cartesian Meshes for Incompressible Flows with Heat Transfer," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 49, no. 3, pp. 277–297, 2006.

[13] D. Z. Noor, M.-J. Chern, and T.-L. Horng, "An immersed boundary method to solve fluidsolid interaction problems," *Computational Mechanics*, vol. 44, no. 4, pp. 447–453, 2009.

[14] N. E. L. Haugen and S. Kragset, "Particle impaction on a cylinder in a crossflow as function of Stokes and Reynolds numbers," *Journal of Fluid Mechanics*, vol. 661, pp. 239–261, 2010.

[15] J. Park, K. Kwon, and H. Choi, "Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160," *KSME International Journal*, vol. 12, no. 6, pp. 1200–1205, 1998.

[16] J. M. Shi, D. Gerlach, M. Breuer, G. Biswas, and F. Durst, "Heating effect on steady and unsteady horizontal laminar flow of air past a circular cylinder," *Physics of Fluids*, vol. 16, no. 12, pp. 4331–4345, 2004.

[17] S. Mittal, "Excitation of shear layer instability in ow past a cylinder at low Reynolds number," *International journal for numerical methods in fluids*, vol. 49, no. August, pp. 1147–1167, 2005.

[18] E. Stålberg, A. Brüger, P. Lötstedt, A. V. Johansson, and D. S. Henningson, "High order accurate solution of flow past a circular cylinder," *Journal of Scientific Computing*, vol. 27, no. 1-3, pp. 431–441, 2006.

[19] Y. Li, R. Zhang, R. Shock, and H. Chen, "Prediction of vortex shedding from a circular cylinder using a volumetric Lattice-Boltzmann boundary approach," *European Physical Journal: Special Topics*, vol. 171, no. 1, pp. 91–97, 2009.

[20] C. S. Peskin, "Flow patterns around heart valves: A numerical method," *Journal of Computational Physics*, vol. 10, no. 2, pp. 252–271, 1972.