22.315 - Midterm Exam

Name: Lorenzo Mazzocco
Due Date: March 22 2023
Code: StarCCM+

Contents

1. PROBLEM 1

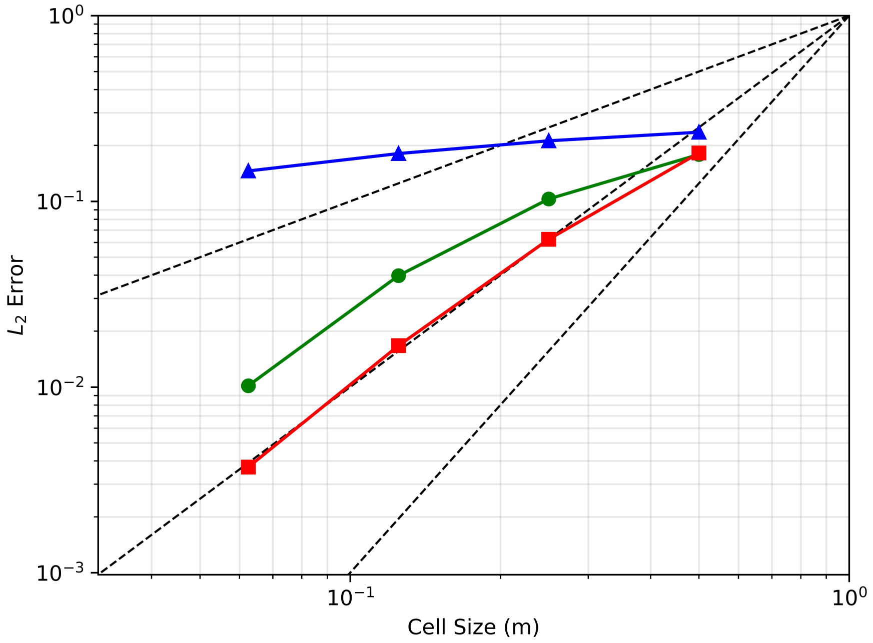


Figure 1.1: L2 error as a function of mesh size for three interpolation schemes: 1st order UD (blue), 2nd order UD (green) and MUSCL/CD (red). The dashed lines indicate different orders of accuracy p (1,2 and 3).

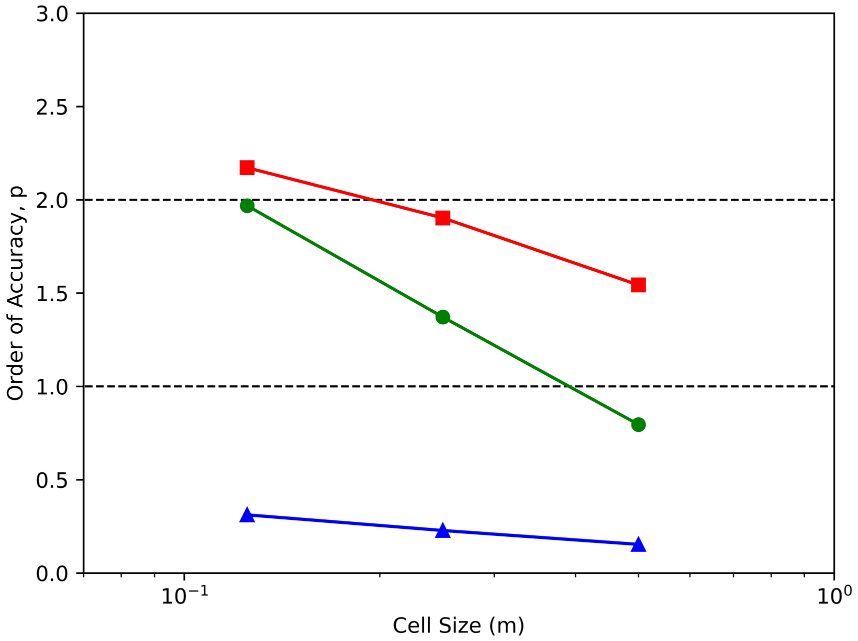


Figure 1.2: Order of accuracy of the simulation as a function of mesh size for three interpolation schemes: 1st order UD (blue), 2nd order UD (green) and MUSCL/CD (red).

1.1. Boundary Conditions

The flow field has two components: the advective field and the vortex. The advective field is parallel to the top and bottom boundaries. For inviscid flow this condition means that the flow does not interact with the boundary whatsoever, that is if we were to have the advective field only, we could choose either a wall condition or a symmetry plane condition for the top and bottom boundaries of our domain. The presence of the vortex field complicates the situation since it technically introduces a velocity component not parallel to the boundary at the top/bottom of our domain. My goal is to prove that the v component at the top and bottom boundaries is negligible. To do so we analyze the vortex field at initial conditions. The v component is:

$$v(x,y) = \frac{1}{0.3}\frac{x}{r}\frac{tanh(r)}{cosh^2(r)} \tag{1.1}$$

while costrained on the top boundary $y = 4$, the component depends only on the x coordinate:

$$r = \sqrt{x^2+16} \tag{1.2}$$

$$v(x) = \frac{1}{0.3}\frac{x}{\sqrt{x^2+16}}\frac{tanh(\sqrt{x^2+16})}{cosh^2(\sqrt{x^2+16})}, \qquad x \in [-4, 9] \tag{1.3}$$

we can plot the component:

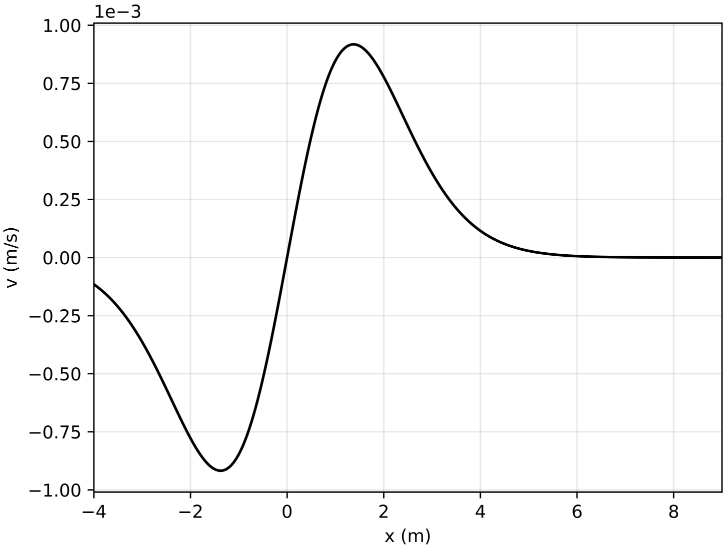


Figure 1.3: v component of the velocity field on the top boundary of the domain.

we can easily compute maximum and average values as well as the relative contribution to the total field which is given by:

$$c = \frac{v}{u_{adv} + v} \tag{1.4}$$

where $u_{adv} = 0.25\ m/s$:

- $max(v) = 9.18 \cdot 10^{-4} m/s \quad \longrightarrow \quad max(c) = 0.37\%$
- $mean(v) = 6.19 \cdot 10^{-6} m/s \quad \longrightarrow \quad mean(c) = 0.002\%$

we have thus proven that at initial conditions the vertical component of the velocity field is, for all practical purposes, negligible. It is true that during the evolution in time the numerical viscosity and diffusion will distort the field but the main effect of those numerical errors is to dissipate kinetic energy and reduce the overall magnitude of the vortex velocity field. Therefore we can consider the initial conditions our worst case scenario. For all those reasons I chose a wall BC for top and bottom but a symmetry plane BC would have been just as good.

As far as the inlet and outlet boundaries are concerned I selected a "Velocity Inlet" BC that let me impose a constant velocity field on the inlet surface equal to $\vec{v} = [0.25, 0.00] m/s$. The outlet is a simple "Outlet" BC. This set up allows us to have a constant advective field at all times.

1.2. Error Components

It is very important to underline that performing a mesh refinement study will not automatically yield the spatial accuracy order. We can only get the *total error* of the simulation, but the total error is composed of different components. That is, there are different mechanisms that introduce a divergence from the analytical solution, in our case we can identify the following:

1. floating point error $L_2^{(fp)}$
2. iteration error $L_2^{(it)}$
3. temporal discretization error $L_2^{(t)}$
4. spatial discretization error $L_2^{(s)}$

therefore we can express the total error as:

$$L_2 = L_2^{(fp)} + L_2^{(it)} + L_2^{(t)} + L_2^{(s)} \tag{1.5}$$

What is important to understand is that performing mesh refinement changes $L_2^{(s)}$ only, but will not affect the other kind of errors. For example, in a perfect environment where all the other errors are negligible for a set of spatial schemes that are second order accurate we would expect the error to become a quarter of the original for a refinement ratio of 2:

$$L_{2,2h} = 4 \cdot L_{2,h} \quad \longrightarrow \quad p = \frac{ln\left(\dfrac{L_{2,2h}}{L_{2,h}}\right)}{ln(r)} = 2 \tag{1.6}$$

But if the other errors are comparable to the spatial discretization error then the observed convergence order is not representative of the spatial discretization convergence order:

$$p = \frac{ln\left(\dfrac{L_{2,2h}^{(s)} + L_2^{(fp)} + L_2^{(it)} + L_2^{(t)}}{L_{2,h}^{(s)} + L_2^{(fp)} + L_2^{(it)} + L_2^{(t)}}\right)}{ln(r)} < 2 \tag{1.7}$$

Our focus should then be to reduce the errors we are not interested in as much as reasonably achievable. In the following sections I will try to prove that for my setup the other errors are negligible.

1.3. Floating Point Error

I call *floating point error* all the errors introduced by numerical operations and not directly ascribable to temporal discretization nor spatial discretization nor iteration truncation. To study this kind of error we can simulate the evolution of the advective field only, from 0 to 20s with a timestep of 0.01s. This way the flow is spatially constant and perpendicular to the mesh, this way even a first order advective scheme gives a perfect interpolation and $L_2^{(s)} = 0$. The fact that the flow field is constant in time assures us that $L_2^{(t)} = 0$ and $L_2^{(it)} = 0$. We impose a fixed number of inner iterations, equal to the average number of inner iterations for our vortex simulations (42), to obtain a realistic number of floating point operations over the 20s simulated. We then compute L_2 being confident that it is equal to $L_2^{(fp)}$.

The result is a mere $L_2^{(fp)} = 2.4 \cdot 10^{-11}$, that is, for all practical purposes, $L_2^{(fp)} \simeq 0$.

1.4. Iteration Error

To decrease the iteration error as much as possible I put stringent criteria on the iteration convergence. First of all I registered the velocity at 10 point probes across the domain: 4 around the initial position of the vortex, 4 around the final position of the vortex and 2 at random locations in the middle. A diagram showing the position of the probes can be found in figure 1.4:

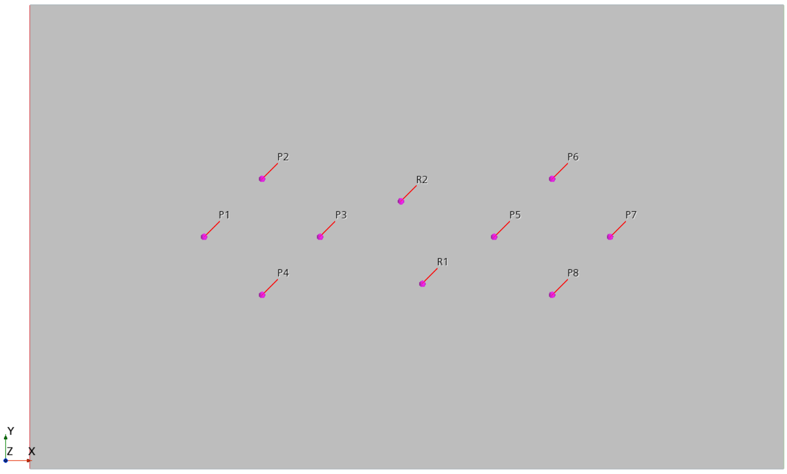


Figure 1.4: Positions of the different point probes on the domain

I then imposed the following stopping criteria for the inner iterations:

1. Asymptotic convergence of velocity on all 10 probes. |max - min| $<$ 1E-5 on a sample of 30 iterations
2. Asymptotic convergence of maximum and minimum velocity magnitudes over the whole domain. |max - min| $<$ 1E-5 on a sample of 30 iterations

Assuming that the probes are representative of the whole domain, we can presume the error for each cell is $E_{cell} < 10^{-5}$, therefore by definition $L_2^{(it)} \leq 10^{-5}$ over the whole domain. As we see from figure 1.1 the smallest total error we obtained is on the order of 10^{-3}, therefore iteration error is negligible.

1.5. Temporal Discretization Error

This error is the most difficult to estimate. I selected a 2nd order implicit solver, the StarCCM+ manual is not very clear on which integration schemes it selects between BDF2, BDF2Opt(4) and BDF2Opt(5). Furthermore the manual underlines that the code uses variable time-steps but does not disclose the algorithm of time-step refinment [7]. Theoretically the global truncation error for BDF2 with adaptable time-step should be $e_t = O(\Delta t^2)$, but it has been shown that in finite volume applications this scheme could yield a lower convergence rate [5]. The ideal way forward given these premises would be to perform a time-step refinement study with 3 simulations around $\Delta t = 0.01$ on a hyper-fine mesh to obtain the order of convergence of the temporal solver. In this case I don't possess a suitable combination of computational power and time to perform this study. I will assume a second order convergence of the global truncation error and estimate my L2 error to be:

$$L_2^{(t)} \simeq (\Delta t)^2 = 10^{-4} \tag{1.8}$$

A more precise evaluation can be obtained using the Lagrange form of the Taylor remainder for the local truncation error. We could in fact say that the global truncation error is bounded by:

$$e_t \leq |f^{(3)}(t^*)|(\Delta t)^2 \tag{1.9}$$

where $|f^{(3)}(t^*)|$ is the maximum value of the third derivative of the function we are integrating ($t^* \in [0, 20]s$). In our case $f = \vec{v}$ (in reality this is more complex than what I said since we are trying to understand the effect on L2 error but the solver integrates each velocity component separately, more on this on appendix A). I estimated this value to be about $|f^{(3)}(t^*)| \simeq 5.4 \cdot 10^{-6}$ (a full derivation can be found in appendix A). This would bring the value of the temporal discretization error to an acceptable value of

$$L_2^{(t)} \leq |f^{(3)}(t^*)|(\Delta t)^2 \simeq 10^{-10} \tag{1.10}$$

Given all the uncertainty around this estimation and given that our total errors are on the order of 10^{-3} - 10^{-2}, I would have selected a smaller timestep, like 0.005s, just to be completely sure. Unfortunately before running the problem I expected the total L2 errors for the simulations to be higher than the values I later obtained so I may have selected too big of a timestep. Thanks to the slowly-varying field this is good enough for our purposes.

1.6. Types of Errors Introduced by the Different Methods

The first consideration is that both the 2nd order UD scheme and the MUSCL/CD scheme show a good order of accuracy with respect to spatial discretization (both methods converge near 2nd order). Whereas the 1st order UD scheme is barely responsive to mesh refinement and achieves a convergence rate well below 1. Then there are a couple of observations which arise from observing the data produced by the simulation:

1. the error fields for 1st UD and 2nd UD (smaller for the latter) are rings whereas the MUSCL/CD error is mainly concentrated at the center of the vortex

2. in coarse meshes the 1st UD and MUSCL/CD scheme tend to distort the vortex whereas the 2nd UD scheme conserves the shape quite well

Evidence for the two considerations above is presented in Appendix B.
Furthermore it is possible to identify some non-physical oscillation in the MUSCL/CD scheme (typical of a higher order scheme), we can see an example in fig 1.5:

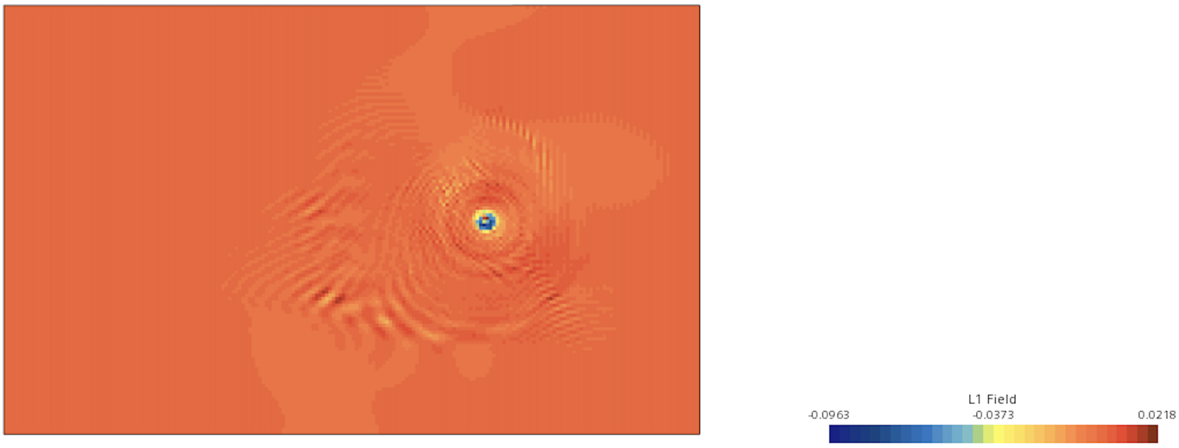


Figure 1.5: Difference between the velocity field and the analytical solution magnitudes for fine mesh and MUSCL/CD interpolator. Non-physical oscillations introduced by the interpolation scheme are present around the vortex.

My conclusions about the errors for the different schemes can be summarized in the following bullet points:

- **1st order UD**: highly diffusive scheme, tends to destroy kinetic energy in the middle range of the vortex ring because of the rotation of the velocity field (this is why we have higher error where the velocity of the vortex is greater).
- **MUSCL/CD**: this scheme does not introduce any type of numerical diffusion but suffers from overshooting, especially in the very center of the vortex (eye of the cyclone) where it undershoots the velocity field quite a bit (this can be seen in figure 1.5 quite well).
- **2nd order UD**: the behaviour is in between the other two schemes.

1.7. Order of Convergence of the Different Schemes

The order of spatial convergence is consistent with the accuracy of the different methods. This can be seen from figure 1.1 and 1.2. 1st order UD does not cross $p = 1$, 2nd order UD does not cross $p = 2$ and MUSCL/CD barely crosses $p = 2$. This has to be expected since the code is constructed to be globally of 2nd order.

From figure 1.2 we can see that the order of convergence appears to be a function of mesh size. Theoretically this shouldn't be the case: order of spatial convergence should be independent of mesh size (especially in structured meshes like ours). But we have to remember that 1.1 and 1.2 do not show the spatial order of convergence but show the *observed* order of convergence for the whole code. That is, there are other mechanisms at play that make the global order of convergence a function of grid size.

1.8. Unexpected Findings

There are no major unexpected findings, good schemes are consistently 2nd order as to be expected with a commercial code like StarCCM+. The only unexpected thing is how bad the convergence order for 1st order UD is $(p \simeq 0.3)$ compared to the expected value of $p = 1$. A possible explanation could be that the a highly diffusive scheme creates higher acceleration, jerk and snap in the distorted vortex field which get picked up as error by the temporal solver. This way a big component of the global error derives from $L_2^{(t)}$ which becomes comparable to $L_2^{(s)}$ severely reducing the observed order of convergence for mesh refinement. But this is a theory I did not verify.

2. PROBLEM 2

2.1.

In the case of isothermal incompressible flow the Navier-Stokes momentum equation is:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{g} \tag{2.1}$$

which can be adimensionalized to:

$$\frac{\partial \boldsymbol{u^*}}{\partial t^*} + \boldsymbol{u^*} \cdot \nabla^* \boldsymbol{u^*} = -\nabla^* p^* + \frac{1}{Re}\nabla^{*2} \boldsymbol{u^*} + \frac{1}{Fr^2}\hat{g} \tag{2.2}$$

where $Re = \frac{uL}{\nu}$ is Reynold's number and $Fr = \frac{u}{\sqrt{gL}}$ is Froude number. Supposing the high Froude limit of negliglible external field is valid the adimensionalized equation becomes:

$$\frac{\partial \boldsymbol{u^*}}{\partial t^*} + \boldsymbol{u^*} \cdot \nabla^* \boldsymbol{u^*} = -\nabla^* p^* + \frac{1}{Re}\nabla^{*2} \boldsymbol{u^*} \tag{2.3}$$

Under these conditions the adimensionalized equation depends only on Re, which we consider to be equal for both water (W) and mercury (Hg). Therefore the adimensionalized velocities are the same:

$$\boldsymbol{u^*} \equiv \frac{\boldsymbol{u}}{U} \tag{2.4}$$

$$\boldsymbol{u^*}_W = \boldsymbol{u^*}_{Hg} \tag{2.5}$$

But the characteristic velocities (U) are different since (assuming $L_{Hg} = L_W = L$):

$$Re_{Hg} = Re_W \tag{2.6}$$

$$\frac{U_{Hg}L}{\nu_{Hg}} = \frac{U_W L}{\nu_W} \tag{2.7}$$

$$U_{Hg} = U_W \frac{\nu_{Hg}}{\nu_W} \tag{2.8}$$

Given that $\nu_{Hg} = 1.15 \cdot 10^{-7} m^2/s$ and $\nu_W = 1.00 \cdot 10^{-6} m^2/s$ [1], $U_W \simeq 10 \cdot U_{Hg}$. Therefore the velocity field of water has a magnitude about 10 times larger than that of liquid mercury, in fact reordering eq. (2.4) and imposing eq. (2.5) one gets:

$$\boldsymbol{u}_W \simeq 10 \cdot \boldsymbol{u}_{Hg} \tag{2.9}$$

2.2.

As one can see from eq. (2.1) the equation depends on the gradient of the pressure field, not the absolute value. Therefore it's not necessary to input the latter, only the former is required.

2.3.

Given a series of cells like shown in Fig. 2.1:

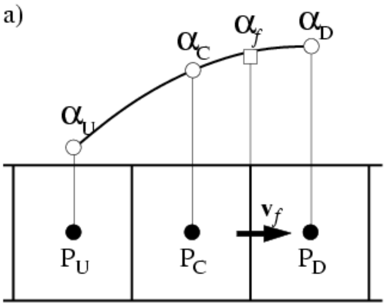


Figure 2.1: Three cells (Upwind, Centered, Downwind) and their respective value for the physical quantity α

an interpolation scheme is said to be bounded if the interpolation value at every point between U and D the interpolated value of α (like α_f) is bound by:

$$\alpha_U \leq \alpha_f \leq \alpha_D \qquad \text{or} \qquad \alpha_U \geq \alpha_f \geq \alpha_D \tag{2.10}$$

If a scheme is not bounded it means that it is introducing non-physical oscillations in the solution. In his PhD thesis in 1954 (published only in 1959) Sergei Godunov proved that numerical schemes for solving partial differential equations, having the property of not generating new extrema (i.e. non-physical oscillations), can be at most first-order accurate [2]. So the answer is no, a 2nd order scheme cannot be bounded by itself but the boundedness has to be enforced by other means (for example StarCCM+ switches to first orders schemes using a criterion based on the Normalized-Variable-Diagrams NVD adimensional variable ξ) .

2.4.

As introduced by [3], given a PDE of the type:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{2.11}$$

the total variation (TV) of the solution is defined as:

$$TV(u^n) = \sum_j |u^n_{j+1} - u^n_j| \tag{2.12}$$

where n is the time index and j is the position index. A numerical scheme is said to be Total Variation Diminishing (TVD), or more precisely Total Variation Nonincreasing (TVNI), if

$$TV(u^{n+1}) \leq TV(u^n) \tag{2.13}$$

2.5.

The first method historically used to avoid checkerboarding is that of the staggered grid arrangement, where the pressures are stored at the cell center and the velocities are stored at the face center. This solution requires more memory because it effectively introduces new grid points. The solution now used by commercial codes is the introduction of a Rhie-Chow-type dissipation factor [6] when computing the interpolated value of the velocity at the face.

2.6.

SIMPLE truncates the pressure correction term while PISO includes the second term using variables from the previous iteration. Therefore each PISO iteration has a more precise pressure correction than SIMPLE. It's for this reason that SIMPLE has historically suffered from overshooting and oscillations, leading to a slower convergence. Because of the fact that in unsteady simulations we need to converge the inner iterations at every timestep (outer iterations), we are in need of a fast algorithm more than in the case of a steady-state simulation. Nevertheless with the introduction of optimal under-relaxation parameters for SIMPLE-type algorithms their convergence speed is similar to that of PISO. On top of that I would argue that a optimally under-relaxed SIMPLE may be better for some unsteady problems since PISO suffers from splitting error proportional to $(\Delta t)^2$ and therefore requires smaller timesteps [4]. SIMPLE-type algorithms don't suffer from this kind of error and may be applied to larger timesteps.

3. PROBLEM 3

3.1.

Using the mid-point rule one can numerically integrate the volume flux by the following formula:

$$\iint_S \vec{v}\cdot\vec{n}dS \simeq \vec{v}_m \cdot \vec{n}_m A_S \tag{3.1}$$

where $\vec{v}_m$ and $\vec{n}_m$ are the velocity and normal unit vector at the midpoint of the surface while A_S is the surface area. The mid-point rule is accurate to the 2nd order. For brevity we omit the *m* subscript from now on.

Each face is represented by a vector that we call segment $\vec{s} = (s_x, s_y)$. The normal vector to the face can then be either $\vec{N} = (-s_y, s_x)$ or its opposite, the normal unit vector is then just the direction of $\vec{N}$, that is: $\vec{n} = \vec{N}/|\vec{N}|$. We always select the normal vector facing outwards of the cell. The area of the cell is then $A = |\vec{s}|$ because the height of the cell is $h = 1$ everywhere. With the velocities at the midpoint provided by the table the computation of the volume flux is trivial:

- **FACE W**: $\vec{s} = (0,4); |\vec{s}| = 4; \vec{n} = (-1,0); \vec{v} = (6,3); \vec{v}\cdot\vec{n} = -6 \longrightarrow \phi_W = -24$
- **FACE S**: $\vec{s} = (6,0); |\vec{s}| = 6; \vec{n} = (0,-1); \vec{v} = (4,4); \vec{v}\cdot\vec{n} = -4 \longrightarrow \phi_S = -24$
- **FACE E**: $\vec{s} = (0,4); |\vec{s}| = 4; \vec{n} = (1,0); \vec{v} = (3,7); \vec{v}\cdot\vec{n} = 3 \longrightarrow \phi_E = 12$
- **FACE NE**: $\vec{s} = (-4,2); |\vec{s}| = 2\sqrt{5}; \vec{n} = \frac{1}{\sqrt{5}}(1,2); \vec{v} = (2,3); \vec{v}\cdot\vec{n} = \frac{8}{\sqrt{5}} \longrightarrow \phi_{NE} = 16$
- **FACE NW**: $\vec{s} = (2,2); |\vec{s}| = 2\sqrt{2}; \vec{n} = \frac{1}{\sqrt{2}}(-1,1); \vec{v} = (1,7/2); \vec{v}\cdot\vec{n} = \frac{5}{2\sqrt{2}} \longrightarrow \phi_{NW} = 5$

where we have assumed that segments have dimensions of m, velocities have dimensions of m/s, normal vectors have no dimension and fluxes have dimension of m^3/s.

3.2.

The general form of the continuity equation for fluid flow is:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \tag{3.2}$$

for incompressible flows $\frac{\partial \rho}{\partial t} = 0$ therefore the continuity equation becomes:

$$\nabla \cdot \vec{u} = 0 \tag{3.3}$$

By Gauss' divergence theorem, if a fluid is incompressible:

$$\iiint_V (\nabla \cdot \vec{u}) dV = \oiint_{\partial V} \vec{u} \cdot \vec{n} dS = \sum_i \phi_i = 0 \tag{3.4}$$

where the last term is the sum of all the fluxes across each face of the cell. In our case $\sum_i \phi_i = -15.$ We have thus demonstrated that the fluid has to be compressible. Furthermore integrating eq. (3.2), knowing that $\rho = 1\ kg/m^3$ everywhere,

$$\iiint_V \frac{\partial \rho}{\partial t} dV = -\iiint_V \nabla \cdot (\rho \vec{v}) dV = -\rho \sum_i \phi_i = 15 \frac{kg}{s} \tag{3.5}$$

but in finite volume codes the fields inside the cell are represented by the cell-center value. Therefore:

$$\frac{\partial \rho}{\partial t} \simeq \frac{1}{V} \iiint_V \frac{\partial \rho}{\partial t} dV = -\frac{\rho}{V} \sum_i \phi_i = 0.5 \frac{kg}{m^3 s} \tag{3.6}$$

Appendices

A. Evaluation of Lagrange Taylor Reminder for Temporal Solver

The analytical solution to the problem is given by a combination of advective field and vortex field (moving in time), that is:

$$\vec{v}_{sol} = \vec{v}_{adv}(x, y) + \vec{v}_{vortex}(x, y, t) \tag{A.1}$$

Therefore only the vortex field contributes to the derivatives with respect to time. The general form of the vortex field is:

$$u(x, y, t) = -\frac{y_r}{r}\frac{f_t(r)}{f_{max}} \tag{A.2}$$

$$v(x, y, t) = \frac{x_r}{r}\frac{f_t(r)}{f_{max}} \tag{A.3}$$

where the following definitions are applied for relative position (x_r, y_r) and distance from vortex center r:

$$x_r = x - u_{adv}t \tag{A.4}$$

$$y_r = y \tag{A.5}$$

$$r = \sqrt{x_r^2 + y_r^2} = \sqrt{(x - u_{adv}t)^2 + y^2} \tag{A.6}$$

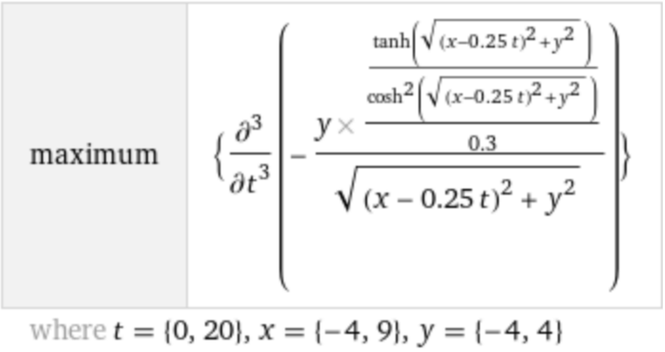
I input the following optimization problem in Wolfram Alpha:

$$\text{find max value of} \quad \frac{\partial^3}{\partial t^3}u(x, y, t) \quad \text{in the domain} \tag{A.7}$$

$$\text{find max value of} \quad \frac{\partial^3}{\partial t^3}v(x, y, t) \quad \text{in the domain} \tag{A.8}$$

$$\tag{A.9}$$

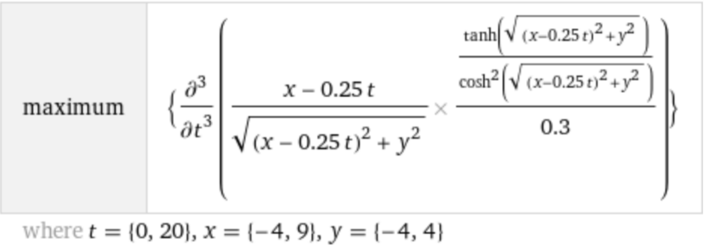
The inputs are shown in figure A.1 and A.2:



maximum $\left\{\frac{\partial^3}{\partial t^3}\left(-\frac{y \times \frac{\frac{\tanh\left(\sqrt{(x-0.25\,t)^2+y^2}\right)}{\cosh^2\left(\sqrt{(x-0.25\,t)^2+y^2}\right)}}{0.3}}{\sqrt{(x-0.25\,t)^2+y^2}}\right)\right\}$

where $t = \{0, 20\}$, $x = \{-4, 9\}$, $y = \{-4, 4\}$

Figure A.1: Wolfram Alpha input, optimization problem for u



maximum $\left\{\frac{\partial^3}{\partial t^3}\left(\frac{x-0.25\,t}{\sqrt{(x-0.25\,t)^2+y^2}}\times\frac{\frac{\tanh\left(\sqrt{(x-0.25\,t)^2+y^2}\right)}{\cosh^2\left(\sqrt{(x-0.25\,t)^2+y^2}\right)}}{0.3}\right)\right\}$

where $t = \{0, 20\}$, $x = \{-4, 9\}$, $y = \{-4, 4\}$

Figure A.2: Wolfram Alpha input, optimization problem for v

the square root of the sum of the results squared is $5.4 \cdot 10^{-6}$. This is the L2 error contribution in the worst case scenario where the biggest time derivatives happen at the same time for both velocity components at all cells at every time-step.

B. Error Fields and Final Vortices

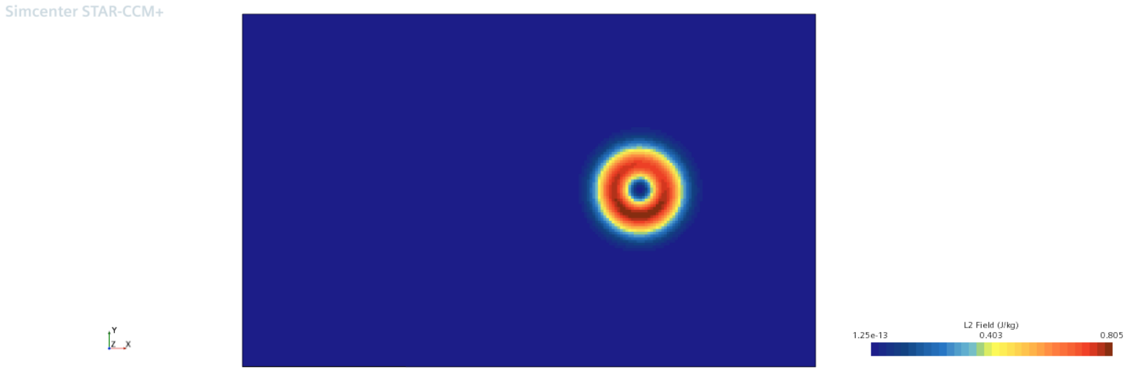


Figure B.1: L2 Error Field for 1st order UD on fine mesh

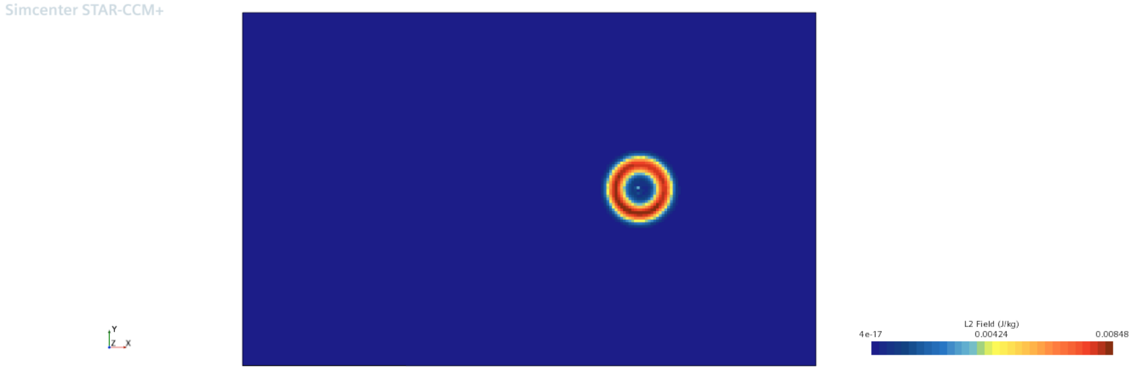


Figure B.2: L2 Error Field for 2nd order UD on fine mesh

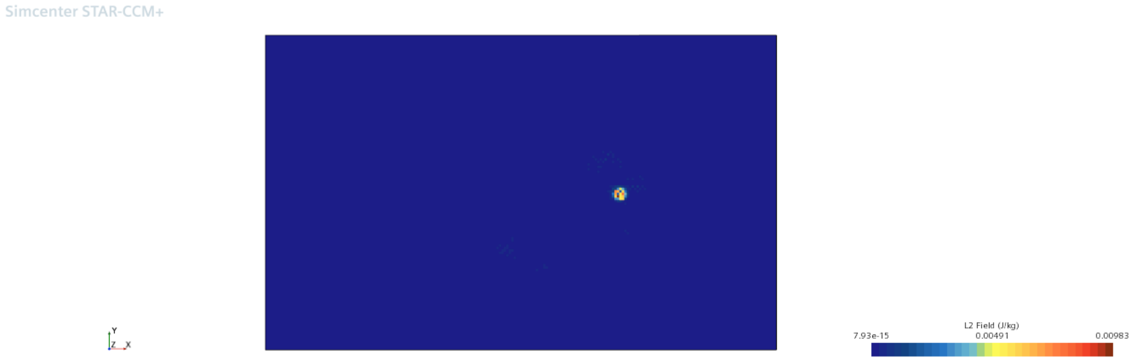


Figure B.3: L2 Error Field for MUSCL/CD on fine mesh

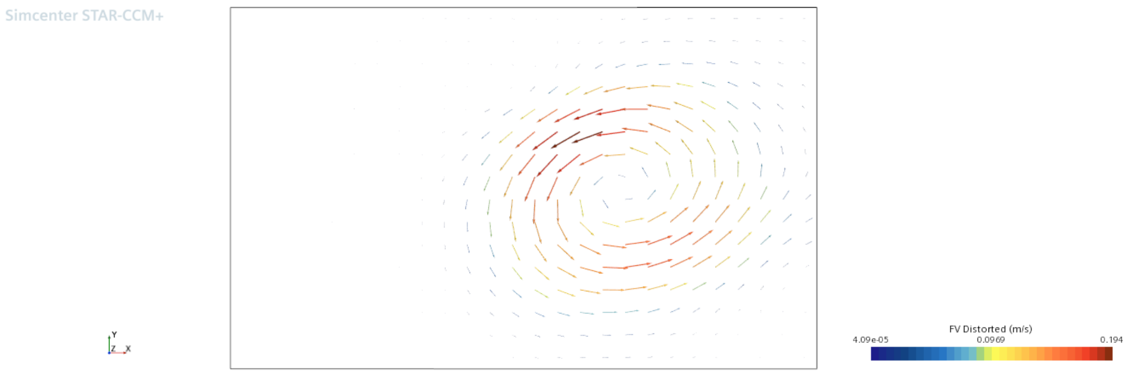


Figure B.4: Final vortex distortion for 1st order UD on mesh 1 ($\Delta x = \Delta y = 0.5$)

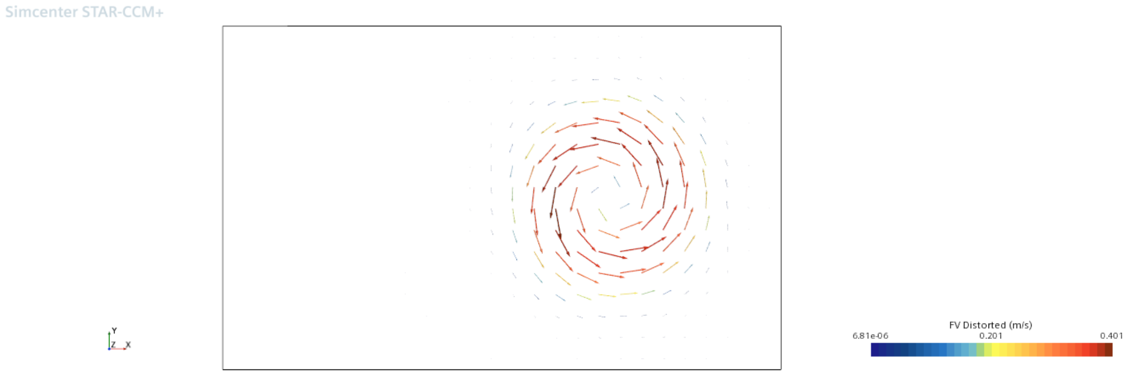


Figure B.5: Final vortex distortion for 2nd order UD on mesh 1 ($\Delta x = \Delta y = 0.5$)

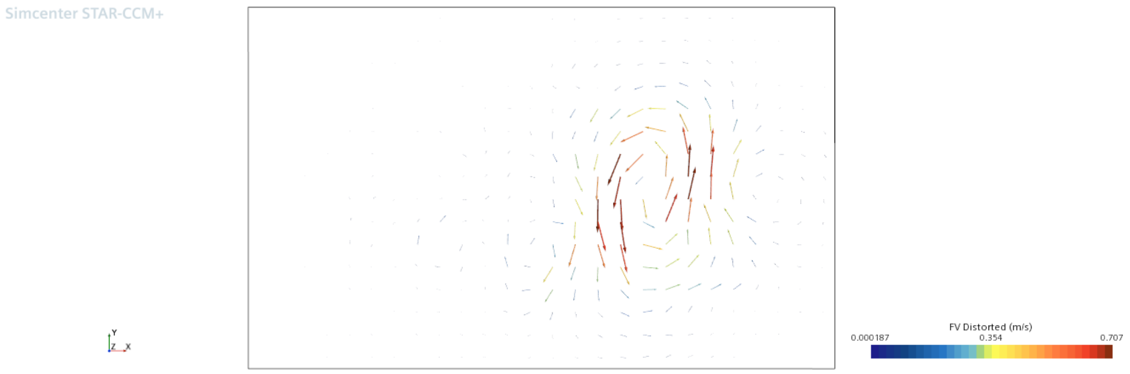


Figure B.6: Final vortex distortion for MUSCL/CD on mesh 1 ($\Delta x = \Delta y = 0.5$)

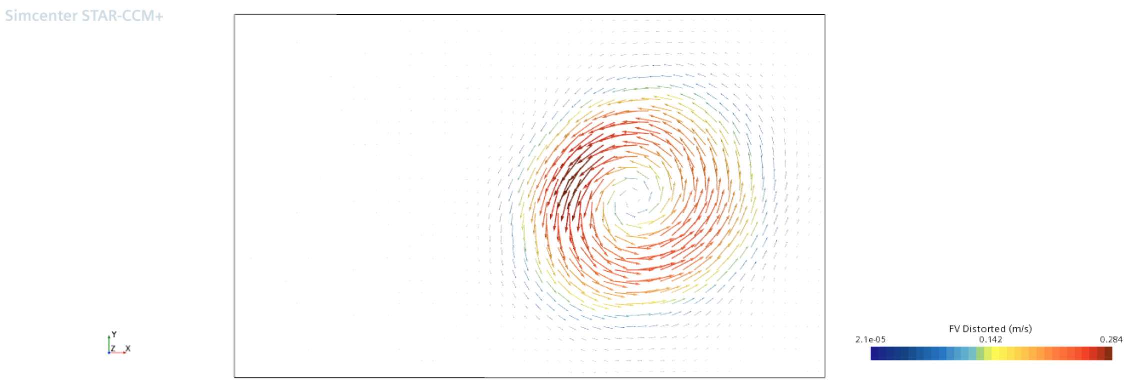


Figure B.7: Final vortex distortion for 1st order UD on mesh 2 ($\Delta x = \Delta y = 0.25$)

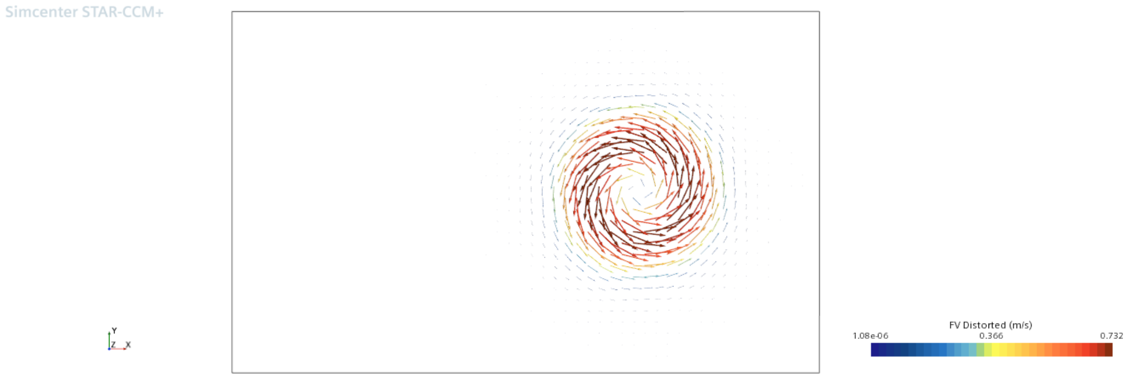


Figure B.8: Final vortex distortion for 2nd order UD on mesh 2 ($\Delta x = \Delta y = 0.25$)

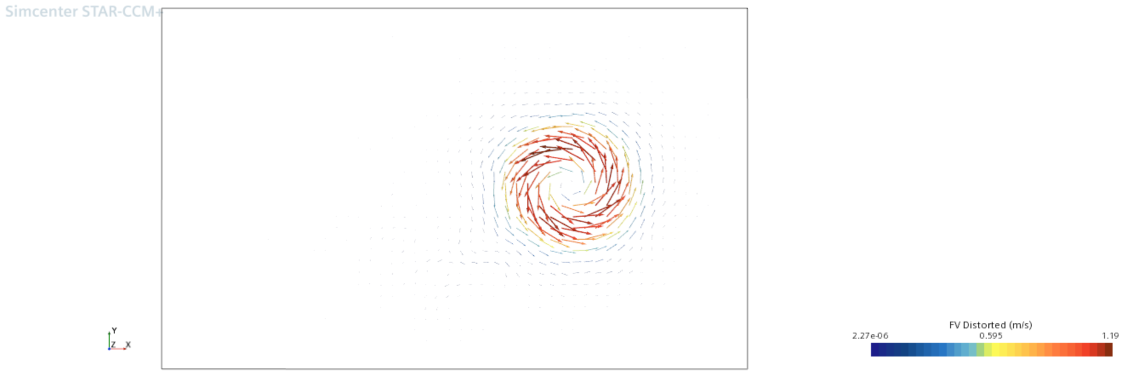


Figure B.9: Final vortex distortion for MUSCL/CD on mesh 2 ($\Delta x = \Delta y = 0.25$)

C. Code

All the code can be found in this public GitHub Repository.

Bibliography

[1] Engineers Edge. *Kinematic Viscosity Table Chart of Liquids and Calculator.* URL: `https://www.engineersedge.com/fluid_flow/kinematic-viscosity-table.htm`.

[2] Sergei K. Godunov and I. Bohachevsky. “Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics”. In: *Matematičeskij sbornik* 47(89).3 (1959), pp. 271–306. URL: `https://hal.science/hal-01620642`.

[3] Ami Harten. “High Resolution Schemes for Hyperbolic Conservation Laws”. In: *Journal of Computational Physics* 49.3 (Mar. 1983), pp. 357–393. DOI: `10.1016/0021-9991(83)90136-5`.

[4] M. Peric J.H. Ferziger. “Computational Methods for Fluid Dynamics”. In: Springer, 2002. Chap. 7, p. 181.

[5] C. Klaij. “Verification of BDF Time-Integration Method with Variable Step Size”. In: *WCCM-ECCOMAS2020* (2020).

[6] C. M. Rhie and W. L. Chow. “Numerical study of the turbulent flow past an airfoil with trailing edge separation”. In: *AIAA Journal* 21.11 (1983), pp. 1525–1532. DOI: `10.2514/3.8284`. eprint: `https://doi.org/10.2514/3.8284`. URL: `https://doi.org/10.2514/3.8284`.

[7] Siemens. *StarCCM+ 16.04.012-R8 User Manual.* URL: `/Siemens/16.04.012-R8/STAR-CCM+16.04.012-R8/doc/en/online/index.html#page/STARCCMP%5C%2FGUID-A806D14C-6833-458E-89B7-8B273EE048B7.html`.