

Discrete stream function method for the incompressible Navier-Stokes equations with simple boundary conditions

Rauan

March 14, 2024

Abstract

The goal of these notes is to present the detailed overview of discrete stream function method for solving incompressible Navier-Stokes equations with simple boundary conditions. We will discuss in detail the scheme formulation, transient and spatial discretizations. Special attention will be paid to the change of unknown variables. After studying these notes one must get a coherent picture of the application of discrete stream function method to incompressible flows with simple Boundary Conditions (BCs) and be able to implement the scheme in code.

Contents

1	Introduction	1
2	Discretization	2
3	Interpolation and regularization operators	3
4	Nullspace method	3
5	Fast method for uniform grid and simple boundary conditions	5
6	Multi-domain approach	6
7	Summary	8
A	Appendix	10
A.1	Transient schemes	10

1 Introduction

In these notes we will be concerned with the discretization of the Navier-Stokes equations describing the flow of an incompressible fluid past the 2-dimensional cylinder with a singular boundary force \mathbf{f} added to the momentum equation as a continuous analog of the immersed boundary formulation:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \epsilon \nabla \cdot \nabla \mathbf{v} + \int_s \mathbf{f}(\xi(s, t)) \delta(\xi - x) ds \quad (1.1a)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (1.1b)$$

$$\mathbf{v}(\xi(s, t)) = \int_x \mathbf{v}(x) \delta(x - \xi) dx = \mathbf{v}_B(\xi(x, t)), \quad (1.1c)$$

which are written here in the non-dimensional form, i.e. $\epsilon \equiv Re^{-1}$ for brevity; also \mathbf{v} is the velocity and p pressure fields. The boundary conditions are assumed to be simple, i.e. Dirichlet and Neumann for the normal and tangential velocities, respectively. Spatial variable x represents position in the flow field, \mathcal{D} , and ξ denotes coordinates along the immersed boundary, $\partial \mathcal{B}$ having a velocity of \mathbf{v}_B . The geometry of the immersed object \mathcal{B} is considered to be of arbitrary shape.

2 Discretization

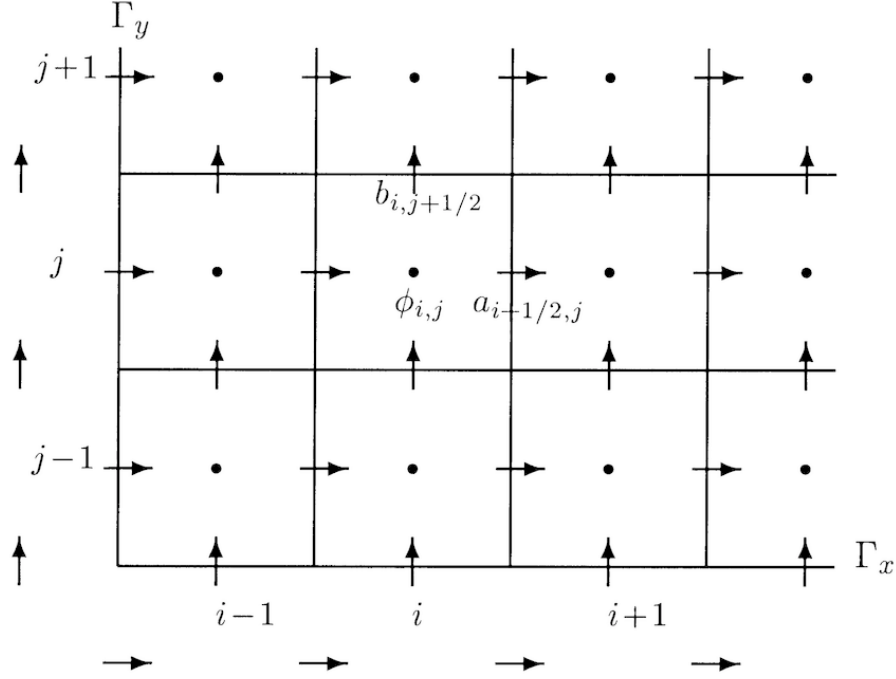


Figure 1: Staggered grid discretization of a two-dimensional computational domain \mathcal{D} and immersed boundary formulation for a body \mathcal{B} depicted by a shaded object. The horizontal and vertical arrows (\rightarrow , \uparrow) represent the discrete u_i and v_i velocity components, respectively. Pressure p_i is positioned at the center of each cell (\times). Lagrangian points, $\xi_k = (\xi_k, \eta_k)$, along $\partial\mathcal{B}$ are shown with filled squares (\blacksquare) where boundary forces $\mathbf{f}_k = (f_{x,k}, f_{y,k})$ are applied (\Rightarrow , \Uparrow).

The above system is discretized with a standard staggered Cartesian grid finite volume method. The mesh and variable locations are depicted in Fig. 1. The computational domain, \mathcal{D} , is represented by a Cartesian grid, (x_i, y_i) , and the immersed boundary, \mathcal{B} is described by a set of Lagrangian points, (ξ_k, η_k) , which can be a function of time.

We can rewrite Eqs. (1.1) using discrete differential operators as

$$\begin{aligned} M \frac{dq}{dt} + Gp - Hf &= \mathbf{N}(q) + Lq + bc_1 && \text{(momentum),} \\ Dq &= 0 + bc_2 && \text{(continuity),} \\ Eq &= \mathbf{v}_B^{n+1} && \text{(no-slip condition),} \end{aligned} \quad (2.1)$$

where $q = (u\Delta y, v\Delta x)$, p and f are the discrete velocity flux vector, pressure, and boundary force. Discretized non-linear advective term $\mathbf{v} \cdot \nabla \mathbf{v}$ is denoted as $\mathbf{N}(q)$, whereas operators M , L are mass matrix and discrete Laplacian respectively. Operators G and D are discrete gradient and divergence matrices s.t. $D = -G^T$ and [1, 4]. The rest operators E and H are the interpolation and regularization operators resulting from the regularization of the Dirac delta functions in momentum and no-slip equations, which are constructed in such a way that $E = -H^T$. No-slip constraint is enforced by equating the boundary velocity, \mathbf{v}_B , to the velocity value along $\partial\mathcal{B}$ interpolated by E from the neighbouring cells. Regularization operator diffuses the singular boundary force along $\partial\mathcal{B}$ to the Cartesian grid [2].

For computational convenience Eqs. (2.1) can be represented as system of linear equations

$$\begin{bmatrix} A & G & -H \\ D & 0 & 0 \\ E & 0 & 0 \end{bmatrix} \begin{pmatrix} q \\ p \\ f \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \\ \mathbf{v}_B^{n+1} \end{pmatrix} + \begin{pmatrix} bc_1 \\ bc_2 \\ 0 \end{pmatrix}, \quad (2.2)$$

where $A = \frac{1}{\Delta t}M - \frac{1}{2}L$ comes from implicit treatment using trapezoid method of viscous terms. Explicit term $r^n = \left(\frac{1}{\Delta t}M + \frac{1}{2}L\right)q^n + \frac{3}{2}\mathbf{N}(q^n) - \frac{1}{2}\mathbf{N}(q^{n-1})$ is also obtained from applying Adams-Bashforth scheme

to advective term. We can use the properties of operators discussed above to rewrite [Sys. \(2.2\)](#) as

$$\begin{bmatrix} A & G & E^T \\ G^T & 0 & 0 \\ E & 0 & 0 \end{bmatrix} \begin{pmatrix} q \\ p \\ \tilde{f} \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \\ v_B^{n+1} \end{pmatrix} + \begin{pmatrix} bc_1 \\ -bc_2 \\ 0 \end{pmatrix}, \quad (2.3)$$

making the above system of linear algebraic equations symmetric. Here, \tilde{f} is the boundary force with an incorporated scaling factor. This form of the equation is known as the Karush-Kahn-Tucker (KKT) system where $(p, \tilde{f})^T$ appear as a set of Lagrange multiplier to satisfy a set of kinematic constraints. In the discretized set of equations, the constraints are purely numerical and it is no longer necessary to distinguish the pressure and boundary force. Instead we can define a combined variable $\lambda = (p, \tilde{f})^T$ for the Lagrange multipliers and group the submatrices as $Q = [G, E^T]$. Note that by removing the boundary force and no-slip condition along $\partial\mathcal{B}$, the traditional discretization of the incompressible Navier-Stokes equations can be retrieved, i.e. being equivalent to applying divergence to momentum and obtaining pressure-Poisson.

Next, we apply the classical projection (fractional-step) algorithm to [Sys. \(2.3\)](#), which can be expressed as an approximate LU decomposition of the left-hand side matrix, to produce the algorithm of immersed boundary projection method [\[2\]](#):

1. $Aq^* = r_1$, (Solve for intermediate velocity).
2. $Q^T A_j^\dagger Q \lambda = Q^T q^* - r_2$, (Solve a modified Poisson equation).
3. $q^{n+1} = q^* - A_j^\dagger Q \lambda$, (Projection step).

Here A_j^\dagger denotes the j -th order Taylor series expansion of A^{-1} with respect to Δt . The explicit terms on the right-hand side have been grouped into r_1 and r_2 . Matrices A and $Q^T A_j^\dagger Q$ are constructed to be symmetric positive definite operators in order to use the conjugate-gradient method to efficiently solve for the intermediate velocity and the Lagrange multipliers. All boundary conditions are set to uniform flow $(U_\infty, 0, 0)$ in the streamwise direction (x -direction) except for the outflow boundary where a convective boundary condition:

$$\frac{\partial \mathbf{v}}{\partial t} + U_\infty \frac{\partial \mathbf{v}}{\partial x} = 0, \quad (2.4)$$

where backward first order difference scheme has to be used (without ghost velocity components) to preserve the symmetric property of matrices A and $Q^T A_j^\dagger Q$.

3 Interpolation and regularization operators

In Colonius 2007 [\[5\]](#).

4 Nullspace method

The nullspace or discrete streamfunction approach [\[1, 3\]](#) is a method for solving [Sys. \(2.3\)](#) without the immersed boundary formulation. In this case, the flow only needs to satisfy the incompressibility constraint, which leads us to the use of discrete streamfunction, s , such that

$$q = C\psi, \quad (4.1)$$

where C is a discrete curl operator. This operator is constructed with column vectors corresponding to the basis of the nullspace of divergence matrix D .

In this section we will show how unknown pressure variables can be eliminated from [Sys. \(2.3\)](#). Publications of Chang [\[1\]](#) and Hall [\[3\]](#) use the idea that matrix D is wider than tall for grids larger than 2×2 , hence it defines a nullspace. The nullspace of matrix D is the set of all solutions to the homogeneous linear system $Dx = 0$, where x is a vector in the null space of D . Let C be the nullspace matrix containing such vectors x .

The number of rows in the nullspace C is equal to the number of faces with unknown velocities (N_f). In two dimensions C has N_n columns, which is equal to the number of nodes in the grid, whereas in three dimensions the nullspace has N_e columns being the number of edges.

In the two-dimensional case, the matrix C has two non-zero elements in each row, which are +1 and -1. The +1 value corresponds to the node 90° from the normal velocity vector, whereas -1 corresponds to the node -90° from the normal velocity vector. For three dimensional case see Chang [1].

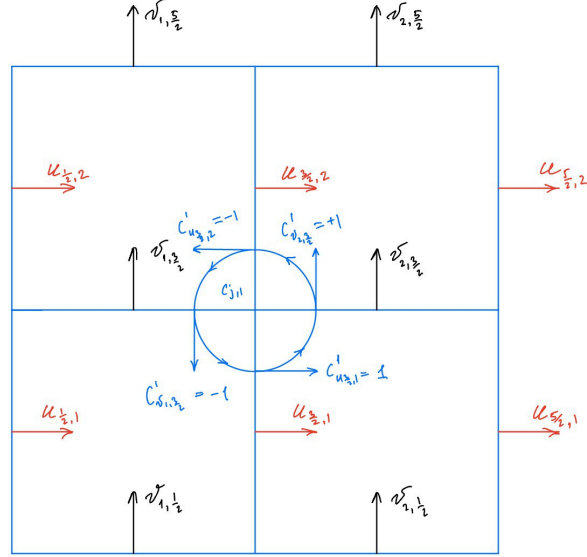


Figure 2: 2×2 example for C matrix.

A more intuitive way of constructing the matrix C relies on the utilization of counterclockwise vorticity around the nodes within the domain (Fig. 2). If the direction of the velocity vector on the adjacent face aligns with the vorticity's direction, +1 is assigned to the corresponding row; conversely, -1 is assigned in the case of opposite directions of velocity and vorticity. After applying the above procedure we obtain

$$C = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}.$$

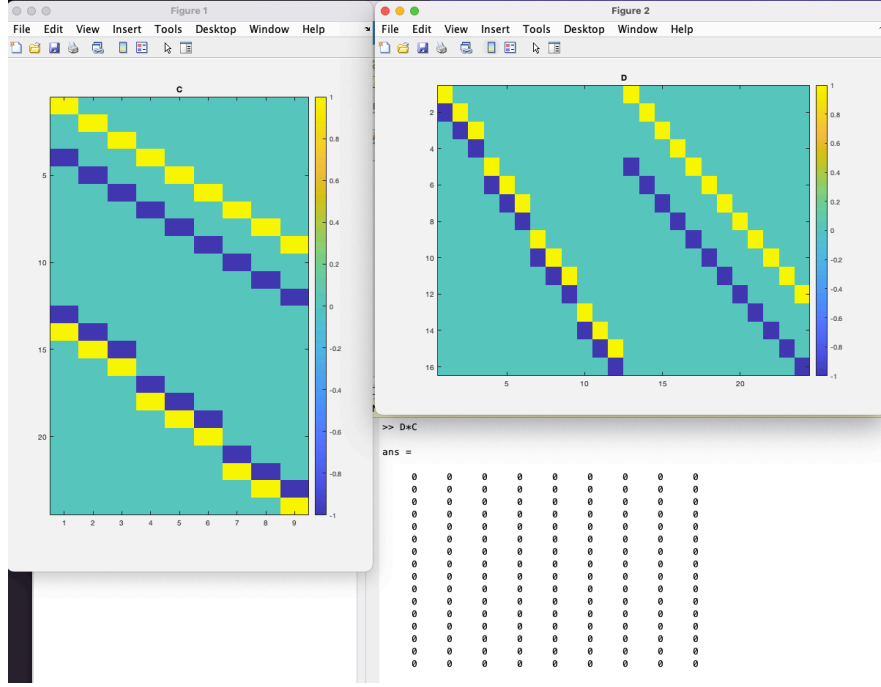


Figure 3: Divergence and Curl matrices.

The matrix C has dimensions corresponding to unknown velocities times the number of nodes around which these velocities revolve. Figure 3 illustrates matrices D and C for a 4×4 grid with an open boundary on the right side of the domain. The desired product then becomes $DC = 0$. $D = -G^T$ (from Section 2) leads to important property $(DC)^T = C^T D^T = -C^T G = 0$. Premultiplying momentum equation in Sys. (2.3) by C^T creates $C^T G p = 0$ term, which completely eliminates the pressure from our system.

In order to make use of efficient solvers, the matrix $C^T A$ can be made symmetric if multiplied by C from the right. We use the property of product $C^T C = -L$ being symmetric Laplacian and $C^T L C = -L^2$ being symmetric biharmonic operator with Dirichlet boundary conditions. Matrix $C^T C = -L^2$ has zero boundary conditions for the vector it is applied to. Hence, the final chord of this method is the introduction of new variable called discrete streamfunction $\psi : q = C\psi$ as discussed at the beginning of this section.

Sys. (2.3)

$$\begin{bmatrix} C^T A C & C^T E^T \\ EC & 0 \end{bmatrix} \begin{pmatrix} \psi^{n+1} \\ \tilde{f} \end{pmatrix} = \begin{pmatrix} C^T r_1^n \\ \mathbf{v}_B^{n+1} \end{pmatrix}. \quad (4.2)$$

The left-hand side matrix is symmetric but in general indefinite, making a direct solution less efficient. The projection (fractional step) approach mimics Eqs. (9)-(11), and we obtain

$$\begin{aligned} C^T A C \psi^* &= C^T r_1^n, \\ EC (C^T A C)^{-1} (EC)^T \tilde{f} &= EC s^* - \mathbf{v}_B^{n+1}, \\ \psi^{n+1} &= \psi^* - (C^T A C)^{-1} (EC)^T \tilde{f}, \end{aligned} \quad (4.3)$$

5 Fast method for uniform grid and simple boundary conditions

In this section we will show that a similar system to Eqs. (2.1) can be solved using fast sine transforms, this method is, however, restricted to uniform grids. Mass matrix M in this case becomes diagonal identity matrix. It is assumed that the values of the velocity are known in the region outside the computational domain, thus, we will apply simple Dirichlet BC's for normal and Neumann to tangential

velocity components at the boundaries. Let us operate with C^T on momentum equation from Eqs. (2.1) in order to eliminate the pressure. We will obtain

$$\frac{\partial \gamma}{\partial t} + C^T E^T \tilde{f} = -\frac{1}{\text{Re} \Delta^2} C^T C \gamma + C^T \mathbf{N}(q) + bc_\gamma, \quad (5.1)$$

where $\gamma = C^T q$ is circulation, Δ stands for constant spacing in uniform grid and we used the fact that $Lq = -\frac{1}{\text{Re} \Delta^2} C C^T q = -\frac{1}{\text{Re} \Delta^2} C \gamma$, since $Dq = 0$. The last part is equivalent to $\nabla^2 \mathbf{v} = \nabla(\nabla \cdot \mathbf{v}) - \nabla \times \nabla \times \mathbf{v} = -\nabla \times \nabla \times \mathbf{v}$ taking into account continuity equation $\nabla \cdot \mathbf{v} = 0$

BC's we apply result in zero Dirichlet BC's for γ . The discrete Laplacian is diagonalized by a sine transform that is computed in $O(N \log_2 N)$ operations (here N stands for the number of unknown γ components). The sine transform pair is denoted as

$$\hat{\gamma} = S \gamma \leftrightarrow \gamma = S \hat{\gamma}, \quad (5.2)$$

where the operator S is equal to inverse of itself due to the possible normalization process. For convenience the combination of fast sine transform with Laplace operator we denote

$$\Lambda = S C^T C S, \quad (5.3)$$

which is a diagonal matrix with eigenvalues of $C^T C$ with integer coefficients.

Using the same time marching procedure as in Section 2 we obtain

$$S \left(I + \frac{\beta \Delta t}{2} \Lambda \right) S \gamma^* = \left(I - \frac{\beta \Delta t}{2} C^T C \right) \gamma^n + \frac{\Delta t}{2} (3 C^T \mathbf{N}(q^n) - C^T \mathbf{N}(q^{n-1})) + \Delta t bc_\gamma, \quad (5.4)$$

$$EC \left(S \Lambda^{-1} \left(I + \frac{\beta \Delta t}{2} \Lambda \right)^{-1} S \right) (EC)^T \tilde{f} = EC S \Lambda^{-1} S \gamma^* - \mathbf{v}_B^{n+1}, \quad (5.5)$$

$$\gamma^{n+1} = \gamma^* - S \left(I + \frac{\beta \Delta t}{2} \Lambda \right)^{-1} S (EC)^T \tilde{f}. \quad (5.6)$$

We can obtain vector \mathbf{v} for all previous time steps using

$$q^n = C \psi^n + bc_q, \psi^n = S \Lambda^{-1} S \gamma^n + bc_\psi. \quad (5.7)$$

In the new system of equations only one linear system (Eq. (5.5)) needs to be solved. This linear system has positive definite left-hand side operator. This system requires

$$O(N(2 \log_2 N + N_{bw} + 4d)) \quad (5.8)$$

against

$$O(N(N_{bw} + (2d + 1)j + 4d)) \quad (5.9)$$

in step 2 of Algorithm (2), where N_{bw} is the bandwidth of the body-force regularization or interpolation operators, d is the dimension of the problem (2 or 3 dimensions) and j is the order of approximate Taylor-series inverse of matrix A . The speedup for 128×128 grid in 2D is about 10, and 30 for $128 \times 128 \times 128$ in 3D. It should be mentioned that there is no iterative error in solving the equation, only round-off error remains. In the case of stationery body the Poisson-alike equation for body forces can be efficiently solved using Cholesky decomposition.

In conclusion of this section it is required to say that this fast method can only be applied to simple boundary conditions, which take only a small portion of all possible scenarios. Nevertheless, in the next section we will discuss the workaround for implementing boundary conditions with a small trade-off.

6 Multi-domain approach

The fast sine transform method relies on simple boundary conditions (Dirichlet for normal and Neumann for tangential velocities at the boundary). Stating this for the small domain will result in two sources of errors

1. Extensive, algebraically decaying potential flow induced by the body.
2. Vorticity may advect or diffuse through the boundary.

These errors are minimized by using a large non-uniform domain as well as using convective boundary conditions discussed in Section 2 which are incompatible with fast sine method in Section 5. In this section we present the way of keeping these error minimizing strategies as well as applying fast sine method together.

The basic idea is to consider nested domains as in Fig. 4. The circulation γ on the inner mesh is interpolated onto the coarser (larger) mesh couple of times, then the Poisson equation is solved on the outer domain. This solution is then interpolated along the boundary of the inner mesh and the Poisson equation is solved with the "corrected" boundary values on smaller meshes. This technique is used recursively a number of times. The domain coarsening factor is 2 at each grid level.

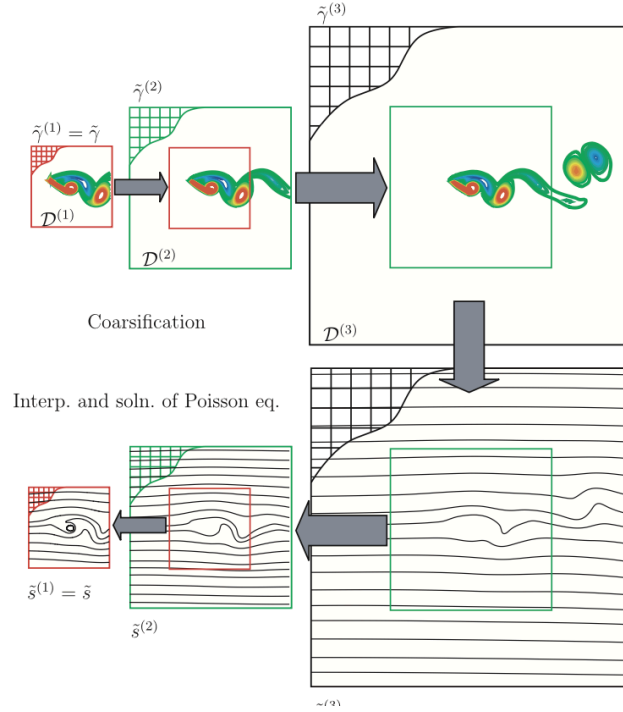


Figure 4: Three layer multi-domain.

The method is described as follows. Let us defined the domain of each grid as $\mathcal{D}^{(k)}$, $k = 1, 2, 3, \dots, N_g$, where $k = 1$ and $k = N_g$ stand for the smallest and largest domains respectively. We then define a multi-domain inverse Laplacian

$$\tilde{\psi} = S\bar{\Lambda}^{-1}S\tilde{\gamma}, \quad (6.1)$$

where $\tilde{\gamma}$ is an arbitrary input vector, $\tilde{\psi}$ is streamfunction solution, and the operator $S\bar{\Lambda}^{-1}S$ implies the following operations:

$$\tilde{\gamma}^{(1)} = \tilde{\gamma} \quad (6.2)$$

$$\tilde{\gamma}^{(k)} = \begin{cases} \tilde{\gamma}^{(k)} & \text{for } x \in \mathcal{D}^{(k)} \setminus \mathcal{D}^{(k-1)}, \\ P^{(k-1) \rightarrow (k)}(\tilde{\gamma}^{(k-1)}) & \text{for } x \in \mathcal{D}^{(k-1)}, \\ k = 2, 3, \dots, N_g, \end{cases} \quad (6.3)$$

$$\tilde{\psi}^{(N_g+1)} = 0, \quad (6.4)$$

$$\tilde{\psi}^{(k)} = S\bar{\Lambda}^{-1}S\tilde{\gamma}^{(k)} + bc_\psi [P^{(k+1) \rightarrow (k)}(\tilde{\psi}^{(k+1)})], k = N_g, N_g - 1, \dots, 1, \quad (6.5)$$

$$\tilde{\psi} = S\bar{\Lambda}^{-1}S\tilde{\gamma} = \tilde{\psi}^{(1)}. \quad (6.6)$$

In the process above $P^{(k-1) \rightarrow (k)}$ and $P^{(k+1) \rightarrow (k)}$ are fine-to-coarse and coarse-to-fine interpolation operators. In 2D case the fine-to-coarse interpolation uses 9-point stencil, that is

$$\begin{aligned} P^{(k-1) \rightarrow (k)}(\tilde{\gamma}^{(k-1)})_{2i,2j} = & \tilde{\gamma}_{i,j}^{(k-1)} + \frac{1}{2}\tilde{\gamma}_{i-1,j}^{(k-1)} + \frac{1}{2}\tilde{\gamma}_{i+1,j}^{(k-1)} + \frac{1}{2}\tilde{\gamma}_{i,j-1}^{(k-1)} + \frac{1}{2}\tilde{\gamma}_{i,j+1}^{(k-1)} + \\ & + \frac{1}{4}\tilde{\gamma}_{i-1,j-1}^{(k-1)} + \frac{1}{4}\tilde{\gamma}_{i+1,j-1}^{(k-1)} + \frac{1}{4}\tilde{\gamma}_{i-1,j+1}^{(k-1)} + \frac{1}{4}\tilde{\gamma}_{i+1,j+1}^{(k-1)}. \end{aligned} \quad (6.7)$$

The coefficients in Eq. (6.7) add up to 4, since the circulation $\tilde{\gamma}$ is the vorticity multiplied by the area and coarsification of the grid by factor of two leads to increase in a factor of 4 increase in cell area. The reverse coarse-to-fine interpolation is performed linearly in case of a mid-point and by taking the value from the coarser mesh if points coincide, note, that this only has to be performed along the boundary of the finer mesh (i.e. four straight segments). Using the multi-domain approach for circulation and solving the Poisson equation, we can finally write the system of equations to be solved for each time step.

$$\begin{aligned} & S \left(I + \frac{\beta \Delta t}{2} \Lambda \right) S \gamma^{(k)*} \\ &= \left(I - \frac{\beta \Delta t}{2} C^T C \right) \gamma^{(k)n} + \frac{\Delta t}{2} \left(3C^T \mathcal{N}(q^{(k)n}) - C^T \mathcal{N}(q^{(k)n-1}) \right) \\ &+ \frac{\Delta t}{2} b_{C_\gamma} \left(\left[P^{(k+1) \rightarrow (k)}(\gamma^{(k+1)*}) \right] + \left[P^{(k+1) \rightarrow (k)}(\gamma^{(k+1)n}) \right] \right) \\ &k = N_g, N_{g-1}, \dots, 1, \\ & EC \left(S \overline{\Lambda}^{-1} \left(I + \frac{\beta \Delta t}{2} \Lambda \right)^{-1} S \right) (EC)^T \tilde{f} = EC S \overline{\Lambda}^{-1} S \gamma^{(1)*} - \mathbf{v}_B^{n+1}, \\ &\gamma^{n+1} = \gamma^{(1)*} - S \left(I + \frac{\beta \Delta t}{2} \Lambda \right)^{-1} S (EC)^T \tilde{f} \\ &\psi^{n+1} = S \overline{\Lambda}^{-1} S \gamma^{n+1}. \end{aligned} \quad (6.8)$$

When the vorticity crosses the boundary of a grid level, $\gamma^{(j)}$ fields are not smooth across the interfaces, especially at the coarsest levels. Results show the propagation of a vortex through mesh levels and slight reflections of the circulation near the boundary, however, the errors remain confined to a small region near the boundary and diffused over time by the physical viscosity.

7 Summary

References

- [1] Wang Chang, Francis Giraldo, and Blair Perot. Analysis of an exact fractional step method. *Journal of Computational Physics*, 180(1):183–199, 2002.
- [2] Tim Colonius and Kunihiro Taira. A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 197:2131–2146, April 2008.
- [3] C. Hall, T. Porsching, R. Dougall, R. Amit, A. Cha, C. Cullen, George Mesina, and Samir Moujaes. Numerical methods for thermally expandable two-phase flow-computational techniques for steam generator modeling. *NASA STI/Recon Technical Report N*, 1980.
- [4] J. B. Perot. An analysis of the fractional step method. *Journal of Computational Physics*, 108:249, 1993.
- [5] Kunihiro Taira and Tim Colonius. The immersed boundary method: A projection approach. *Journal of Computational Physics*, 225(2):2118–2137, August 2007.

A Appendix

A.1 Transient schemes