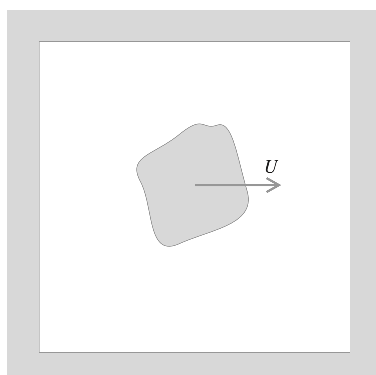
MAE 250H, Spring 2019
J. D. Eldredge
Homework 5, Due Tuesday, May 30

This homework is focused on solutions in the vicinity of complex geometries. You are to use the immersed boundary projection method, as presented in class, to enforce the presence of the geometry.

Solution of 2-d potential flow with immersed boundary method. Consider an object of some arbitrary shape accelerated impulsively from rest to speed U, as shown in the figure below. The shape is inside of a square box with impenetrable walls. Even if the fluid is viscous, the very first instant is purely a potential flow, because no vorticity has yet been created. In other words, we are interested in solving the flowfield before vorticity has been created and diffused to satisfy the no-slip condition; the only operable boundary condition is no-flow-through, both on the walls of the enclosure (which are stationary) and on the moving object. As we know, the no-flow-through condition on a stationary wall can be enforced by setting the streamfunction to a constant. For a moving object, the streamfunction is no longer constant, but it is locally equal to the value of streamfunction due to rigid body motion, $\psi_b(x, y) = Uy - Vx$.



The governing equation of this problem is thus

$$\nabla^2 \psi = 0,$$

subject to $\psi = \psi_0$ on the enclosure walls and $\psi(x, y) = \psi_b(x, y) = Uy - Vx$ on the surface of the moving object. The constant streamfunction ψ_0 might not be zero, and will be part of the exploration in this problem.

The discrete solution of this problem will make use of the usual discrete Laplacian, here evaluated on nodal data (where streamfunction lives). Since the walls of the enclosure are aligned with the grid, we will impose the Dirichlet condition $\psi = \psi_0$ directly, so that it effectively becomes a forcing function on the right-hand side: $Ls = r = -Ls_0$, where s is the discrete streamfunction and s_0 is the streamfunction with zeros in all interior nodes and ψ_0 in the boundary nodes. The body's behavior will be enforced with immersed boundary projection, via Lagrange multiplier forcing:

$$\begin{bmatrix} L & H \\ E & 0 \end{bmatrix} \begin{pmatrix} s \\ f \end{pmatrix} = \begin{pmatrix} r \\ s_b \end{pmatrix} \tag{1}$$

where H and E are, respectively, the regularization and interpolation operators of the body Lagrange points and s_b denotes the vector of streamfunction values evaluated on the body Lagrange points.

As we already know from our work with the fractional step method, this problem can be solved by LU decomposition:

$$\begin{bmatrix} L & 0 \\ E & -EL^{-1}H \end{bmatrix} \begin{bmatrix} I & L^{-1}H \\ 0 & I \end{bmatrix} \begin{pmatrix} s \\ f \end{pmatrix} = \begin{pmatrix} r \\ s_b \end{pmatrix} \tag{2}$$

or, as an algorithm,

Algorithm 1 Algorithm

Solve $Ls^* = r$
Solve $EL^{-1}Hf = -Es^* + s_b$
Compute $s = s^* - L^{-1}Hf$

What we need for carrying out this algorithm are:

1. Operations H and E, based on a choice of discrete Dirac delta function. Here, we will use the ddf of Roma et al. (1999):

$$\tilde{d}(r) = \begin{cases} \frac{1}{3}\left(1 + \sqrt{1 - 3r^2}\right), \ r < 0.5 \\ \frac{1}{6}\left(5 - 3r - \sqrt{1 - 3(1-r)^2}\right), \ 0.5 \le r < 1.5 \\ 0, \ r \ge 1.5 \end{cases} \tag{3}$$

The two-dimensional ddf is then

$$d_2(x, y) = \tilde{d}(x/\Delta x)\tilde{d}(y/\Delta x). \tag{4}$$

Note that we have scaled both the x and y coordinates by grid spacing Δx. The E operation is then given by

$$s(\boldsymbol{\xi}_k) = \sum_{i,j} s_{ij} d_2(x_i - \xi_k, y_j - \eta_k), \tag{5}$$

where $\boldsymbol{\xi}_k = (\xi_k, \eta_k)$ are the coordinates of the kth Lagrange point and x_i and y_j are the coordinates of node (i, j). The regularization operator H can be defined as the transpose of this operator, $H = E^T$, so that

$$f_{ij} = \sum_{k=1}^{N_b} f(\boldsymbol{\xi}_k) d_2(x_i - \xi_k, y_j - \eta_k). \tag{6}$$

where N_b is the number of Lagrange points. Keep in mind that the spacing between points on the body should be approximately the same as Δx.

2. A means of solving the Dirichlet Poisson equation on nodes on a uniform Cartesian grid. This is very easily done with discrete sine transforms. This constitutes L^{-1}.

3. A means of solving the system $EL^{-1}E^T f = -Es^* + s_b$. The operator on the left-hand side is symmetric and positive definite, so we can use the conjugate gradient method. In Julia, we can use the `IterativeSolvers` package, which has it implemented. Matlab has `pcg`. Each iteration of the cg method will call a routine that evaluates the operator acting on the current guess for f. This, in turn, will call operations that evaluate E^T, L^{-1}, and E.

What sorts of things should you do?

1. A circle is an obvious shape to start with. On an open domain (with no enclosure), with the cylinder centered at the origin, the exact solution is known:

$$\psi(x, y) = \frac{UR^2 \sin\theta}{\sqrt{x^2 + y^2}} \tag{7}$$

   In fact, you can pretend that you are doing this open domain problem by replacing the constant value ψ_0 on the domain boundary with this exact solution, evaluated on all boundary nodes. Use this for validation of your method.

2. Check what is happening inside the body. Plot the streamfunction to see this clearly.

3. Use other body shapes. Explore!