Streamfunction Solution for 2D Flow

A Nek5000 Example

Kento Kaneko (kaneko@mit.edu)

May 15, 2024

Note: This is a casual document written to supplement the Nek5000 streamfunction example downloadable from kaneko.io/f/psi.tgz. Almost all of the introduced concepts are assumed to be common knowledge in the discipline of inviscid flows for which the author has no formal education in, and therefore do not claim any novelty of – nor provide proper attributions for the concepts.

Introduction

Streamfunction is a useful object in 2D flow because it is a scalar field that describes the flow field: allowing visualization of the flow through its isocontours. The streamfunction ψ satifies the relations

$$\partial_y \psi = u_1, \tag{1}$$

$$-\partial_x \psi = u_2; \tag{2}$$

u_1 and u_2 are the components of the velocity field $\boldsymbol{u}$. Furthermore, we can write it as a solution to the elliptic equation

$$\nabla^2 \psi = -\omega \text{ in } \Omega, \tag{3}$$

where $\omega = (\nabla \times \boldsymbol{u}) \cdot \hat{\boldsymbol{e}}_3$ and Ω is the problem domain. Thus if we have a velocity solution at a particular time, it appears trivial to solve for the streamfunction, but we are missing the boundary conditions. In the following, we consider types of domains (and boundary conditions) whose boundaries (if there are any) are coincident with a streamfunction isocontour. Other boundary types such as inflow / outflow can be considered with minor modifications, but are not the subject of this document.

Periodic Domain

For a purely periodic domain, boundaries with prescribed conditions do not exist, therefore the equation can be solved as-is with the weak formulation:

Find $\psi \in X \equiv H^1(\Omega)$ s.t.

$$a(v, \psi) = m(v, \omega), \ \forall v \text{ in } X, \tag{4}$$

where

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v, \ \ m(u, v) = \int_\Omega uv. \tag{5}$$

Note that constant functions are in the kernel of a: the solution is unique up to a constant.

Singly-Connected Boundary Domain

For a domain with a single connected boundary which itself is a streamline (e.g., wall-bounded domain, periodic domain with cylinder occlusions, etc.), we can apply an arbitrary constant essential boundary condition. We choose value 0 for simplicity. Then we solve the problem:

Find $\psi_0 \in X_0 \equiv H^1_0(\Omega)$ s.t.

$$a(v, \psi_0) = m(v, \omega), \ \forall v \text{ in } X_0, \tag{6}$$

where $H^1_0(\Omega)$ is a subset of $H^1(\Omega)$ whose members satisfy the homogeneous essential boundary condition.

Periodic Wall-Bounded Domain

For a periodic domain between two walls, the boundary conditions add complexity, but are still solvable for arbitrary flow conditions. For the formulation, we restrict to domain with periodic faces, Γ_p, whose normals $\hat{\boldsymbol{n}}_p$ satisfy the condition $|\hat{\boldsymbol{e}}_1 \cdot \hat{\boldsymbol{n}}_p| = 1$ *and* 'top' and 'bottom' walls whose respective normals $\hat{\boldsymbol{n}}_\pm$ satisfy $\pm\hat{\boldsymbol{n}}_\pm \cdot \hat{\boldsymbol{e}}_2 > 0$ (one-signed). The former restriction is not significant since the periodic cut is 'arbitrary' to a certain extent, and the latter restriction is due to implementation considerations. We note this restriction allows a simple definition of the domain length

$$L \equiv \sup_{\boldsymbol{x},\boldsymbol{y} \in \Omega} (\boldsymbol{x} - \boldsymbol{y}) \cdot \hat{\boldsymbol{e}}_1. \tag{7}$$

For periodic wall-bounded domains violating the restrictions, the following formulation may apply as-is or with minor modifications.

We first note that the two boundaries Γ_+ and Γ_- are associated with constant streamfunction values (due to $\nabla\psi = \mathbf{0}$ derived from the no-slip condition) ψ_+ and ψ_-, respectively. To find the difference between the two, we can integrate along an arbitrary curve Γ^*:

$$\begin{aligned}\psi_+ - \psi_- &= \int_{\Gamma^*} \nabla\psi \cdot d\boldsymbol{s} \\ &= \int_{\Gamma^*} (-u_2\,\hat{\boldsymbol{e}}_1 + u_1\,\hat{\boldsymbol{e}}_2) \cdot d\boldsymbol{s} \\ &= Q(\boldsymbol{u}; \Gamma^*) \end{aligned} \tag{8}$$

and note that difference in the streamfunction value is a functional evaluation of the flow solution. Since we have a choice in the integration path, we choose the path along of of the periodic faces (which are vertical by construction) we also note for this domain, the functional Q can be evaluated through a volume integral:

$$Q(\boldsymbol{u}) = \int_{\Gamma_p} u_1 \tag{9}$$

$$= m\left(L^{-1}, u_1\right). \tag{10}$$

We now turn to the problem of potential flow (i.e., $\omega = 0$) in the same domain Ω corresponding to $Q = 1$. The solution to the problem is $\phi = \phi_0 + \phi_D$, where $\phi_D \in \{w_{\Gamma_\pm} = \pm\frac{1}{2} : w \in X\}$ satisfies the essential boundary conditions. Thus, the problem statement for ϕ_0 is:

Find $\phi_0 \in X_0$ s.t.

$$a(v, \phi_0) = -a(v, \phi_D),\ \forall\, v \in X_0. \tag{11}$$

We note that this problem is solvable and now we can decompose $\psi = \psi_0 + Q\phi$ while noting that $Q\phi$ satisfies the essential boundary conditions for ψ thereby restricting the lifted solution to X_0: $\psi_0 \in X_0$. The weak form of this decomposition is

$$a(v, \psi_0) + Qa(v, \phi) = m(v, \omega); \tag{12}$$

however, we recognize from (11) that $a(v, \phi) = 0$ and we recover (6). This decomposition is completely independent of the domain specification with the only requirement being the computation of Q('s). For this domain, that computation is trivial; but as you can imagine, this approach is widely applicable in a numerical context.

As an example, for a periodic flow past a cylinder problem between bounded walls, we can solve for ϕ_+, where the solution is $-1/2$ on the bottom wall and the cylinder, and $1/2$ on the top wall; and ϕ_-, where the solution is $1/2$ on the top wall and $-1/2$ on the cylinder and bottom wall. A linear combination of these potential flow solutions can account for arbitrary flow conditions so that only (6) must be solved to find ψ once corresponding Q_+ and Q_- are evaluated.

1 Implementation in Nek5000

The Nek5000 example which starts with a rectangular domain, $\Omega = [0, 10] \times [0, 1]$, whose left and right boundaries are periodic, and whose top and bottom boundaries are walls. We apply a deformation to the domain such that the top wall is coincident with curve $x_2 = 1 + \cos(\pi x_1)/20$ (the bottom wall is $x_2 = 0$). The main entry point for the streamfunction example is the `comp_psi` subroutine called from `userchk` which reside in `util.f`. This subroutine computes the streamfunction ψ (the first argument) from the velocity specified in the second and third arguments, the rest being scalar parameters. `nproj` is the dimension of the residual projection space (see below), `itype` is the type of domain, and `nf` is the number of output fields (1 will output the streamfunction only, the rest are the components of the streamfunction and the vorticity). A streamfunction output from the example is shown in Figure 1 at $t = 100$. The internally, a general `poisson` wrapper subroutine around `hmholtz` and `cggo` is used to solved the homogeneous streamfunction problem for ψ_0. We show the temporal error behavior of the streamfunction-derived velocity in Figure 2. Residual Projection (discussed later) does not affect the error, as expected.

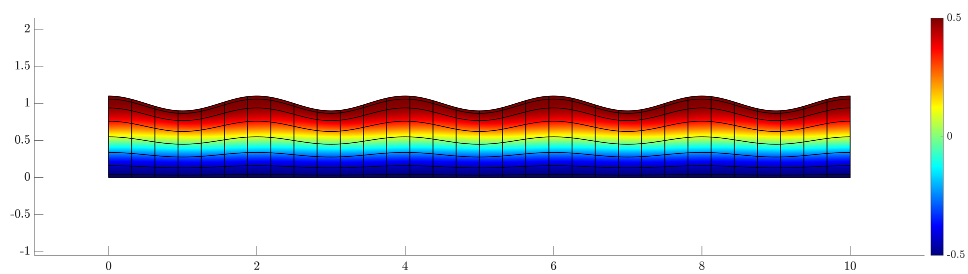

Figure 1: ψ at $t = 100$. Plotted using NekToolKit.

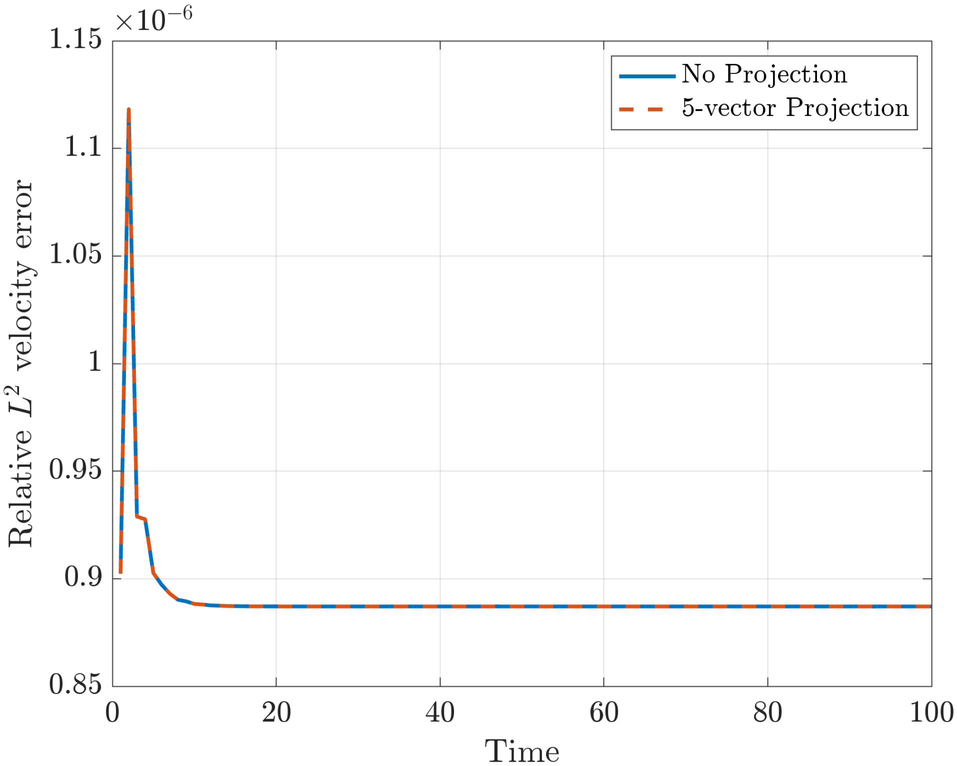

Figure 2: Relative L^2 error in the streamfunction-derived velocity vs. the velocity solution.

Residual Projection

We recognize that (6) has an equivalent minimization problem

$$\psi_0 = \arg\min_{z \in X_0} f(z;\omega), \tag{13}$$

$$f(z;\omega) \equiv \frac{1}{2}a(z,z) - m(z,\omega). \tag{14}$$

If we have an orthonormalized set of previous solutions $\{z_i\}_{i=1}^n$ (i.e., $a(z_i, z_j) = \delta_{ij}$), we would like a starting guess $z_0 = \sum_{i=1}^n c_i z_i$ that minimizes $f(z)$. We now follow standard procedure of substituting the linear combination and taking the derivative (product of two indexed variables imply summation from 1 to n):

$$f\left(c_i z_i;\omega\right) \equiv \frac{1}{2} a\left(c_i z_i, c_j z_j\right) - m\left(c_i z_i, \omega\right) \tag{15}$$

$$= \frac{1}{2} c_i^2 - c_i\, m\left(z_i, \omega\right) \tag{16}$$

$$\frac{\partial}{\partial c_j} f(c_i z_i;\omega) = c_j - m\left(z_j, \omega\right). \tag{17}$$

Thus, the constants are $c_j = m(z_j, \omega)$ and we can solve an easier (iterative) problem with the separation $\psi_0 = m(z_i, \omega) z_i + \tilde{\psi}$ for the problem:

$$a(v, \tilde{\psi}) = m(v, \omega) - a(v, z_i)\, m(z_i, \omega). \tag{18}$$

This residual projection is performed by Nek5000 for both the velocity and pressure solves, but this algorithm is implemented in the `initpproj`, `pproj`, and the `postpproj` subroutines in `util.f` for use in the `poisson` subroutine. The comparison of solving for ψ_0 with and without residual projection is shown in Figure 3. We see significant improvement in the iteration count, 75% overall and 94% for the last solve.

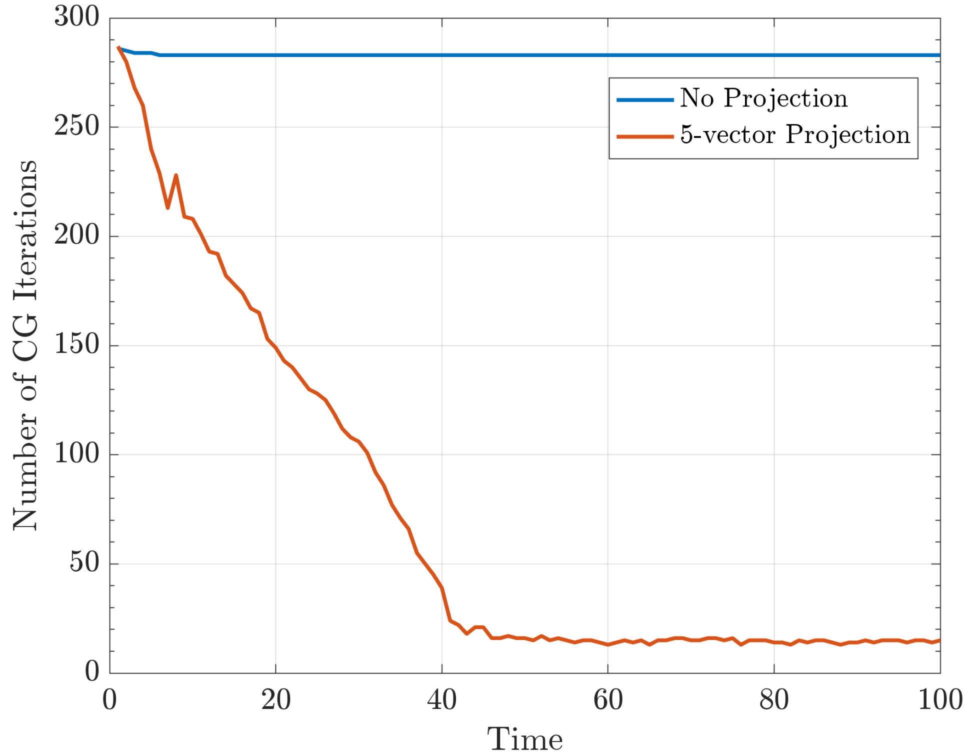

Figure 3: Residual projection comparison.