Chorin's Projection Method with Finite Difference for 2D Incompressible Navier Stokes

August 12, 2019

The reason why NSE is hard is a coupling between momentum and pressure via the continuity contraint. So we can just first pretend the pressure does not exist (e.g. Burger's equations) and solve that explicitly. Then we can project that intermediate velocity field onto a divergence free space. In other words, we split the NSE into solenoidal and irrotational components. Here we will use finite difference for everything (no staggered grid though).

We begin with writing down the NSE:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p &= \Delta \mathbf{u} \\ \nabla \mathbf{u} &= 0 \\ \mathbf{u} &= 0 \quad \text{on} \quad \partial\Omega \end{aligned}$$

where $\mathbf{u} = (u, v)$ is a two-dimensional velocity and p is a scalar pressure. Ω is the domain whereas $\partial\Omega$ represents the boundary. These boundary conditions are called "no-slip". This should generalize to cavity BC or general dirichlet.

Chorin's Method The first step of Chorin's is to ignore pressure.

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= \Delta \mathbf{u} \\ \mathbf{u} &= 0 \quad \text{on} \quad \partial\Omega \end{aligned}$$

Note this is just Burger's equation in 2D. We can discretize this in time in two ways. Define the solution to this "intermediary" system as $\mathbf{u}^*$.

Explicit Discretization Here, we can use Adams-Bashford or Runge-Kutta. We usually use Adams-Bashford since less evaluations.

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\triangle t} + (\mathbf{u}^{\frac{n+1}{2}} \cdot \nabla)\mathbf{u}^{\frac{n+1}{2}} = \Delta \mathbf{u}^{\frac{n+1}{2}} \tag{1}$$

We can estimate the half time steps as:

$$(\mathbf{u}^{\frac{n+1}{2}} \cdot \nabla)\mathbf{u}^{\frac{n+1}{2}} \approx \frac{3}{2}(\mathbf{u}^n \cdot \nabla)\mathbf{u}^n - \frac{1}{2}(\mathbf{u}^{n-1} \cdot \nabla)\mathbf{u}^{n-1}$$
$$\Delta\mathbf{u}^{\frac{n+1}{2}} \approx \frac{3}{2}(\Delta\mathbf{u}^n) - \frac{1}{2}(\Delta\mathbf{u}^{n-1})$$

Set $\mathbf{u}^{-1} = \mathbf{u}^0$ when initializing this technique. This first step is then a first-order explicit Euler method.

Semi-Implicit Discretization Adams-Bashford for advection and Crank-Nicholson for diffusion. The benefit of this is that CN is an implicit method which is more expensive but better. I think (but am not sure) that you can only do this with (spatial) finite difference since you get explicit representations for the Laplacian.

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\triangle t} + (\mathbf{u}^{\frac{n+1}{2}} \cdot \nabla)\mathbf{u}^{\frac{n+1}{2}} = \Delta(\frac{\mathbf{u}^* + \mathbf{u}^n}{2}) \tag{2}$$

We will revisit this later when we discretize space.

Back to Chorin's We can now take the remaining piece in NSE:

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla p \tag{3}$$

which can be discretized in space as:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\triangle t} = -\nabla p^{n+1} \tag{4}$$

Notice we are using the intermediate time step now! We can rearrange this to get an update rule to march forward:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \triangle t \nabla p^{n+1} \tag{5}$$

The remaining unknown is p^{n+1}. We can derive a form for it by taking the divergence of both sides of this new equation.

$$\nabla \cdot (\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\triangle t}) = \nabla \cdot (-\nabla p^{n+1})$$
$$\frac{1}{\triangle t}(\nabla \cdot \mathbf{u}^{n+1} - \nabla \cdot \mathbf{u}^*) = -\Delta p^{n+1}$$
$$\frac{\nabla \cdot \mathbf{u}^*}{\triangle t} = \Delta p^{n+1}$$

where the last step holds because $\nabla \cdot \mathbf{u}^{n+1} = 0$ by the continuity equation.

Spatial Discretization We will present central difference (with uniform grid) approximations for the pressure and burgers equation above.

Start with (sorry added ρ back in)

$$\frac{1}{\rho}\Delta p^{n+1} = \frac{\nabla \cdot \mathbf{u}^*}{\triangle t} \tag{6}$$

This will be discretized to

$$\frac{p^{n+1}_{i+1,j} - 2p^{n+1}_{i,j} + p^{n+1}_{i-1,j}}{\triangle x^2} + \frac{p^{n+1}_{i,j+1} - 2p^{n+1}_{i,j} + p^{n+1}_{i,j-1}}{\triangle y^2} = \frac{\rho}{\triangle t}(\frac{u^*_{i+1,j} - u^*_{i-1,j}}{\triangle x} + \frac{v^*_{i,j+1} - v^*_{i,j-1}}{\triangle y})$$
$$\frac{p^{n+1}_{i+1,j} - 2p^{n+1}_{i,j} + p^{n+1}_{i-1,j}}{\triangle x^2} + \frac{p^{n+1}_{i,j+1} - 2p^{n+1}_{i,j} + p^{n+1}_{i,j-1}}{\triangle y^2} = C_{i,j}$$

where we use $C_{i,j}$ as a short hand for the RHS. We now rely on numerical methods in elliptic PDEs to solve this:

$$\triangle y^2(p_{i+1,j} - 2p_{i,j} + p_{i-1,j}) + \triangle x^2(p_{i,j+1} - 2p_{i,j} + p_{i,j-1}) = \triangle x^2 \triangle y^2 C_{i,j} \tag{7}$$
$$\triangle y^2(p_{i+1,j} + p_{i-1,j}) + \triangle x^2(p_{i,j+1} + p_{i,j-1}) - \triangle x^2 \triangle y^2 C_{i,j} = (2\triangle x^2 + 2\triangle y^2)p_{i,j} \tag{8}$$
$$p_{i,j} = \frac{\triangle y^2 p_{i+1,j} + \triangle y^2 p_{i-1,j} + \triangle x^2 p_{i,j+1} + \triangle x^2 p_{i,j-1} - \triangle x^2 \triangle y^2 C_{i,j}}{2\triangle x^2 + 2\triangle y^2} \tag{9}$$

If we consider time discretization, then we get:

$$p^{n+1}_{i,j} = \frac{\triangle y^2 p^n_{i+1,j} + \triangle y^2 p^n_{i-1,j} + \triangle x^2 p^n_{i,j+1} + \triangle x^2 p^n_{i,j-1} - \triangle x^2 \triangle y^2 C_{i,j}}{2\triangle x^2 + 2\triangle y^2} \tag{10}$$

This is called Jacobi Iteration. We can do slightly better if we use new values as we update them i.e.

$$p^{n+1}_{i,j} = \frac{\triangle y^2 p^n_{i+1,j} + \triangle y^2 p^{n+1}_{i-1,j} + \triangle x^2 p^n_{i,j+1} + \triangle x^2 p^{n+1}_{i,j-1} - \triangle x^2 \triangle y^2 C_{i,j}}{2\triangle x^2 + 2\triangle y^2} \tag{11}$$

To implement this, we would need to loop. We can also use a simple acceleration technique called successive over-relaxation (SOR).

$$p^{n+1}_{i,j} = \beta * \{\frac{\triangle y^2 p^n_{i+1,j} + \triangle y^2 p^{n+1}_{i-1,j} + \triangle x^2 p^n_{i,j+1} + \triangle x^2 p^{n+1}_{i,j-1} - \triangle x^2 \triangle y^2 C_{i,j}}{2\triangle x^2 + 2\triangle y^2}\} + (1-\beta) * p^n_{i,j} \tag{12}$$

Now that we have computed p^{n+1}, we can estimate its gradient. Here we need to impose a boundary conditions!

$$\nabla p^{n+1}_{i,j} \approx (\frac{p^{n+1}_{i+1,j} - p^{n+1}_{i-1,j}}{\triangle x}, \frac{p^{n+1}_{i,j+1} - p^{n+1}_{i,j-1}}{\triangle y})$$
$$\frac{\partial p}{\partial x} = 0, \frac{\partial p}{\partial y} = 0$$

We add a note here that in fact this is not how many papers compute pressure. Several use spectral methods (which we intentionally avoid here) and others use more sophisticated iterative methods. We will settle for something simple for now. Secondly, pressure here is not the actual pressure. Kim and Morin's paper show to derive/estimate the true pressure. But for our purposes, it doesn't really matter. We can consider this the real pressure for the time being. When we get to real experiments where the magnitudes of values matter, we need to add an additional step.

Back to deriving!

Then we can plug this into the Burgers. Recall (with explicit discretization)

$$\mathbf{u}^* = \mathbf{u}^n - \triangle t(\frac{3}{2}(\mathbf{u}^n \cdot \nabla)\mathbf{u}^n - \frac{1}{2}(\mathbf{u}^{n-1} \cdot \nabla)\mathbf{u}^{n-1}) + \triangle t(\frac{3}{2}\Delta\mathbf{u}^n - \frac{1}{2}\Delta\mathbf{u}^{n-1}) \tag{13}$$

For exposition, we will write the discretization for the first dimension of $\mathbf{u}$: u.

$$\begin{aligned} u^* = u^n - \triangle t(\frac{3}{2}(u^n\frac{\partial u^n}{\partial x} + v^n\frac{\partial u^n}{\partial y}) - \frac{1}{2}(u^{n-1}\frac{\partial u^{n-1}}{\partial x} + v^{n-1}\frac{\partial u^{n-1}}{\partial y}) \\ + \triangle t(\frac{3}{2}(\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2}) - \frac{1}{2}(\frac{\partial^2 u^{n-1}}{\partial x^2} + \frac{\partial^2 u^{n-1}}{\partial y^2})) \end{aligned}$$

We can discretize this further with centered difference by the following substitutions:

$$\begin{aligned} \frac{\partial u^n}{\partial x} &= \frac{u^n_{i+1,j} - u^n_{i-1,j}}{\triangle x} \\ \frac{\partial^2 u^n}{\partial x^2} &= \frac{u^n_{i+1,j} - 2u^n_{i,j} + u^n_{i-1,j}}{\triangle x^2} \end{aligned}$$

Similar ones for v^n and u^{n-1}.

Then lastly, we can do the projection step:

$$\begin{aligned} u^{n+1}_{i,j} &= u^*_{i,j} - \triangle t(\frac{p^{n+1}_{i+1,j} - p^{n+1}_{i-1,j}}{\triangle x}) \\ v^{n+1}_{i,j} &= v^*_{i,j} - \triangle t(\frac{p^{n+1}_{i,j+1} - p^{n+1}_{i,j-1}}{\triangle y}) \end{aligned}$$

Crank-Nicholson People make a big deal about using Crank Nichoslon for the diffusion term, which results in some important error reduction. It is much more difficult to implement but we have walked through it here.

Recall the momentum equation,

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\triangle t} + (\mathbf{u}^{\frac{n+1}{2}} \cdot \nabla)\mathbf{u}^{\frac{n+1}{2}} = \Delta(\frac{\mathbf{u}^* + \mathbf{u}^n}{2}) \tag{14}$$

We are going to rewrite this in a different way. Add in the Adam-Bashford approximation for the advection term. Let $\mathbf{H}^n = (\mathbf{u}^n \cdot \nabla)\mathbf{u}^n$. Then,

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\triangle t} = \frac{1}{2}(3\mathbf{H}^n - \mathbf{H}^{n-1}) + \frac{1}{2Re}(\frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2})(\mathbf{u}^* + \mathbf{u}^n)$$

where $\frac{\delta}{\delta x}$ is a finite difference operator. Let $A = \frac{\triangle t}{2Re}(\frac{\delta^2}{\delta x^2})$ and $B = \frac{\triangle t}{2Re}(\frac{\delta^2}{\delta y^2})$ to get the following,

$$\begin{aligned}
\mathbf{u}^* - \mathbf{u}^n &= \frac{\triangle t}{2}(3\mathbf{H}^n - \mathbf{H}^{n-1}) + (A+B)(\mathbf{u}^* + \mathbf{u}^n) \\
(\mathbf{u}^* - \mathbf{u}^n) - (A+B)(\mathbf{u}^* + \mathbf{u}^n) &= \frac{\triangle t}{2}(3\mathbf{H}^n - \mathbf{H}^{n-1}) + 2(A+B)(\mathbf{u}^* + \mathbf{u}^n) \\
(1 - A - B)(\mathbf{u}^* + \mathbf{u}^n) &= \frac{\triangle t}{2}(3\mathbf{H}^n - \mathbf{H}^{n-1}) + 2(A+B)(\mathbf{u}^* + \mathbf{u}^n)
\end{aligned}$$

We might try to solve this but it will involve solving a dense linear system, which is expensive. Thus, we make the following assumption:

$$(1 - A - B)(\mathbf{u}^* + \mathbf{u}^n) \approx (1 - A)(1 - B)(\mathbf{u}^* + \mathbf{u}^n) \tag{15}$$

This allows us to use something called *alternating direction implicit method*. We solve the above in two steps!

$$(1 - A)\mathbf{w} = \frac{\triangle t}{2}(3\mathbf{H}^n - \mathbf{H}^{n-1}) + 2(A+B)(\mathbf{u}^* + \mathbf{u}^n) \tag{16}$$

$$(1 - B)(\mathbf{u}^* + \mathbf{u}^n) = \mathbf{w} \tag{17}$$

We will now explicitly write out the algorithm, starting with the first step.

$$(1 - A)\mathbf{w} = \frac{\triangle t}{2}(3\mathbf{H}^n - \mathbf{H}^{n-1}) + 2(A+B)(\mathbf{u}^* + \mathbf{u}^n)$$

Let $\mathbf{w} = (u^\ddagger, v^\ddagger)$. We can then explicitly write the finite difference part. We first focus on the first dimension.

$$u^\dagger_{i,j} - \frac{\triangle t}{2}\frac{u^\dagger_{i+1,j} - 2u^\dagger_{i,j} + u^\dagger_{i-1,j}}{\triangle x^2} = C_u \tag{18}$$

where

$$\begin{aligned}
C_u = \frac{\triangle t}{2}(3(u^n_{i,j}\frac{u^n_{i+1,j} - u^n_{i-1,j}}{\triangle x} + v^n_{i,j}\frac{u^n_{i,j+1} - u^n_{i,j-1}}{\triangle y}) - (u^{n-1}_{i,j}\frac{u^{n-1}_{i+1,j} - u^{n-1}_{i-1,j}}{\triangle x} + v^{n-1}_{i,j}\frac{u^{n-1}_{i,j+1} - u^{n-1}_{i,j-1}}{\triangle y})) \\
+ 2(\frac{\triangle t}{2Re}(\frac{u^n_{i+1,j} - 2u^n_{i,j} + u^n_{i-1,j}}{\triangle x^2} + \frac{u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}}{\triangle y^2}))
\end{aligned}$$

We can continue to massage this a bit.

$$2 \triangle x^2 u^{\ddagger}_{i,j} - \triangle t u^{\ddagger}_{i+1,j} + 2 \triangle t u^{\ddagger}_{i,j} - \triangle t u^{\ddagger}_{i-1,j} = 2 \triangle x^2 C_u$$
$$- \triangle t u^{\ddagger}_{i+1,j} + (2 \triangle x^2 + 2 \triangle t) u^{\ddagger}_{i,j} - \triangle t u^{\ddagger}_{i-1,j} = 2 \triangle x^2 C_u$$

We can solve all of these together (for all i, j) as a linear system. Let's now do the second dimension.

$$v^{\dagger}_{i,j} - \frac{\triangle t}{2} \frac{v^{\dagger}_{i+1,j} - 2v^{\dagger}_{i,j} + v^{\dagger}_{i-1,j}}{\triangle x^2} = C_v \tag{19}$$

where

$$C_v = \frac{\triangle t}{2}(3(u^n_{i,j} \frac{v^n_{i+1,j} - v^n_{i-1,j}}{\triangle x} + v^n_{i,j} \frac{v^n_{i,j+1} - v^n_{i,j-1}}{\triangle y}) - (u^{n-1}_{i,j} \frac{v^{n-1}_{i+1,j} - v^{n-1}_{i-1,j}}{\triangle x} + v^{n-1}_{i,j} \frac{v^{n-1}_{i,j+1} - v^{n-1}_{i,j-1}}{\triangle y}))$$
$$+ 2(\frac{\triangle t}{2Re}(\frac{v^n_{i+1,j} - 2v^n_{i,j} + v^n_{i-1,j}}{\triangle x^2} + \frac{v^n_{i,j+1} - 2v^n_{i,j} + v^n_{i,j-1}}{\triangle y^2}))$$

Then,

$$- \triangle t v^{\ddagger}_{i+1,j} + (2 \triangle x^2 + 2 \triangle t) v^{\ddagger}_{i,j} - \triangle t v^{\ddagger}_{i-1,j} = 2 \triangle x^2 C_v \tag{20}$$

Note that its tridiagonal which makes all this work worth it. Now we have $\mathbf{w}$. We move on to step 2.

Recall $(1 - B)(\mathbf{u}^* - \mathbf{u}^n) = \mathbf{w}$. We can smooth this a bit more.

$$(\mathbf{u}^* - \mathbf{u}^n) - \frac{\triangle t}{2}(\frac{\delta^2}{\delta y^2}\mathbf{u}^* - \frac{\delta^2}{\delta y^2}\mathbf{u}^n) = \mathbf{w} \tag{21}$$

Focus on the i-th and j-th element.

$$(u^*_{i,j} - u^n_{i,j}) - \frac{\triangle t}{2}(\frac{u^*_{i,j+1} - 2u^*_{i,j} + u^*_{i,j-1}}{\triangle y^2} - \frac{u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}}{\triangle y^2}) = u^{\ddagger}_{i,j}$$
$$2 \triangle y^2 u^*_{i,j} - 2 \triangle y^2 u^n_{i,j} - \triangle t(u^*_{i,j+1} - 2u^*_{i,j} + u^*_{i,j-1}) + \triangle t(u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}) = 2 \triangle y^2 u^{\ddagger}_{i,j}$$
$$2 \triangle y^2 u^*_{i,j} - \triangle t u^*_{i,j+1} + 2 \triangle t u^*_{i,j} - \triangle t u^*_{i,j-1} = 2 \triangle y^2 u^{\ddagger}_{i,j} + 2 \triangle y^2 u^n_{i,j} - \triangle t(u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}) = S$$
$$- \triangle t u^*_{i,j+1} + (2 \triangle y^2 + 2 \triangle t) u^*_{i,j} - \triangle t u^*_{i,j-1} = S$$

This is again another tridiagonal matrix solve. This will give us $\mathbf{u}^*$.

Similarly for the v-momentum, we get

$$S = 2 \triangle y^2 v^{\ddagger}_{i,j} + 2 \triangle y^2 v^n_{i,j} - \triangle t(v^n_{i,j+1} - 2v^n_{i,j} + v^n_{i,j-1}) \tag{22}$$
$$- \triangle t v^*_{i,j+1} + (2 \triangle y^2 + 2 \triangle t) v^*_{i,j} - \triangle t v^*_{i,j-1} = S \tag{23}$$