Computational Fluid Dynamics Reading Group: Projection Methods

Fall 2009. 4th Session, 06/10/09
Dharshi Devendran & Eduardo Corona

1.1 "Primitive variable" formulation methods

In the literature, discretization methods for solving the Navier-Stokes equation which are written in terms of the velocity and pressure are known as "primitive (or primary) variable formulation" methods (we actually solve for $u$ and $P$, not for the vorticity, say). It is important to distinguish those which rely on a standard time discretization step, in which velocity and pressure terms evolve simultaneously, and those which are fractional-step projection methods. This last type of projection methods was first introduced by Chorin and Temam in 1969, and is characterized by a "fractional-step", followed by an orthogonal projection onto the space of solenoidal vector fields (that is, incompressible vector fields).

We will see that in both methods, it is often the case that a Poisson equation for the pressure is solved.

Key difference: In the standard discretization methods, this will not ensure incompressibility, and an extra boundary condition (really painful) has to be fulfilled. However, in fractional-step projection methods this equation will indeed replace the incompressibility condition, rendering it a significantly more attractive method.

1.1.1 Standard Discretization Method

We briefly introduce the standard discretization techniques, which will serve as a baseline for comparison with projection methods, and will introduce some helpful results. So, first we discretize the Navier-Stokes equation in time, discretizing the nonlinear term in an explicit manner, and the linear term implicitly. That is:

\[
\begin{align*}
\frac{\partial u}{\partial t} + u \cdot \nabla u &= -\nabla p + \nu \Delta u \quad \text{(N-S)} \\
0 &= \nabla \cdot u
\end{align*}
\]

\[
\frac{u^{n+1} - u^n}{\Delta t} + \nabla p = \nu \Delta u^{n+1} - u^n \cdot \nabla u^n
\]

\[
\begin{align*}
[-\Delta + \gamma I]u^{n+1} + \nabla P^{n+1} &= g(u^n) \\
\nabla \cdot u^{n+1} &= 0
\end{align*}
\]

(D N-S)

Where $\gamma = 1/(\nu \Delta t)$ and $g(u^n) = \gamma u^n - \nu^{-1} u^n \cdot \nabla u^n$. Taking the divergence of this discretized Navier-Stokes equation, we find that:

\[
[-\Delta + \gamma I] \nabla \cdot u^{n+1} + \Delta P^{n+1} = \nabla \cdot g(u^n)
\]

\[
\Delta P^{n+1} = \nabla \cdot g(u^n)
\]
That is, once we have $u^n$ we can solve this Poisson equation for $P^{n+1}$, and then use a Helmholtz solver to find $u^{n+1}$. Although this equation is implied by the incompressibility condition (with which we got rid of $u^{n+1}$), it does not imply it. Substituting in the momentum equation, it only implies that:

$$ \left[-\Delta + \gamma I \right] \nabla \cdot u^{n+1} = 0 $$

Which on its own does not ensure that $\nabla \cdot u^{n+1} = 0$ on the entire domain. However, if we also demand that $\nabla \cdot u^{n+1} \big|_{\partial \Omega} = 0$, then the resulting Dirichlet problem is unique, and so this extra condition ensures incompressibility.

Hence, the time-discretized Navier-Stokes problem can be written as a boundary value problem consisting of two elliptic equations:

$$ \left\{ \begin{array}{l}
\left[-\Delta + \gamma I \right] u^{n+1} + \nabla P^{n+1} = g(u^n) \\
\Delta P^{n+1} = \nabla \cdot g(u^n) \\
u^{n+1} \big|_{S} = b \\
\nabla \cdot u^{n+1} \big|_{S} = 0
\end{array} \right. $$

It is assumed that $\int b \cdot n dS = 0$.

The main problem with this formulation is that, since we impose 2 boundary conditions for $u$ and none for $P$, it is impossible to decouple them. We must solve for both simultaneously, in order to account for the redundancies in solving for the velocity term.

**Decoupling the Pressure** In order to derive conditions for the pressure term, we first observe that for homogeneous boundary conditions for $u$, the discretized equation tells us that $-\nabla P^{n+1} + g(u^n) \in R([-\Delta + \gamma I])$. By orthogonality, we have that $-\nabla P^{n+1} + g(u^n) \in N([-\Delta + \gamma I]^\perp)$, where $A^\perp$ denotes the adjoint operator of $A$. Using Green’s identity, with $\Delta y = -\Delta + \gamma I$, we have that:

$$ \int_{\Omega} u \cdot \Delta y v - v \cdot \Delta y u dV = \int_{\partial \Omega} n \times (u \cdot \nabla \times v - v \cdot \nabla \times u) + n \cdot (u \nabla \cdot v - v \nabla \cdot u) dS $$

Using the homogeneous boundary conditions,

$$ \int_{\Omega} u \cdot \Delta y v - v \cdot \Delta y u dV = - \int_{\partial \Omega} n \times v \cdot \nabla \times u dS $$

So, $\Delta y$ is $\Delta y$ with the boundary condition $n \times v \big|_{\partial \Omega} = 0$. Thus, $\chi \in N([-\Delta y]^\perp) \iff \Delta y \chi = 0$, $n \times \chi \big|_{\partial \Omega} = 0$ (and $n \times \chi \neq 0$ for non-trivial $\chi$). For all such $\chi$,

$$ \int_{\Omega} [-\nabla P^{n+1} + g(u^n)] \cdot \chi = 0 $$

For the non-homogeneous case, this condition becomes:

$$ \int_{\Omega} [-\nabla P^{n+1} + g(u^n)] \cdot \chi = \int_{\partial \Omega} n \times b \cdot (\nabla \times \chi) + (n \cdot b) \nabla \cdot \chi dS $$

We have to work very hard to deal with this boundary condition for the pressure term, in order to solve the uncoupled system of equations.
Decomposing the Pressure: Glowinski-Pironneau  The main idea of this method is to convert the integral condition into Dirichlet boundary data. In particular, we can come up with a decomposition scheme for $p$ on the boundary:

$$p(x) = p_0(x) + \int_{\partial\Omega} G(x, s') \lambda(s') dS(s'), \ x \in \partial\Omega$$

Where:

$$\begin{cases} 
\Delta p_0 = \nabla \cdot g & ; \quad p_0|_{\partial\Omega} = 0 \\
\Delta G(x, s) = 0 & ; \quad G|_{\partial\Omega} = \delta(x - s) 
\end{cases}$$

Determining $G(x, s)$ for a non-trivial domain is already a problem. Substituting the boundary conditions, we obtain:

$$\int_{\partial\Omega} B(s, s'; \chi) \lambda(s') dS(s') = \beta(s; \chi) \ \forall \chi \in N(\Delta_{\gamma})$$

Solving for all $\chi$ is too burdensome, and there is a way to exchange it for:

$$\int_{\partial\Omega} \tilde{B}(s, s'; v) \lambda(s') dS(s') = \tilde{\beta}(s; v) \ \forall v \text{ s.t. } n \times v |_{s} = 0$$

At the expense of solving yet another pair of elliptic equations:

$$\begin{cases} 
\Delta_{\gamma} u_0 = -\nabla p_0 + g & ; \quad u_0|_{\partial\Omega} = b \\
\Delta_{\gamma} U(x, s) = -\nabla G(x, s) & ; \quad U|_{\partial\Omega} = 0 
\end{cases}$$

Even if we overcome the technical burden of solving for $G$ and $U$, this still leaves the problem of discretizing the equation for $\lambda$. Since we only care about the boundary, we can use a subset of $v$’s such that their support is inside $\partial\Omega + B_{\epsilon}(0)$. Finally, on each time step we have to solve the following set of equations:

$$\begin{align*}
\tilde{B} \lambda^{n+1} & = \tilde{\beta} \\
Dp^{n+1} & = b(\lambda) \\
\Gamma u^{n+1} & = c(p)
\end{align*}$$

One possibility to solve these equations is to perform an approximation using finite elements (this is somewhat natural, since this might facilitate discretizing the integral conditions).

1.1.2 Fractional-step projection method

This method, introduced by Chorin and Temam, is what is usually known as the projection method for solving the Navier-Stokes equations. As we have seen, in the case of the standard discretization procedure, even if we obtain separate equations for velocity and pressure, quite a bit of work is needed to ensure the incompressibility condition.

The main idea of this fractional-step projection method can be explained as follows:
STEP 1: We compute an intermediate velocity \( u^* \), by ignoring the pressure term and incompressibility. We solve the time discretized equation.

STEP 2: We perform a step which involves a Poisson equation for the pressure, and is equivalent to a projection of \( u^* \) onto the space of solenoidal (incompressible) velocity fields. In this case, we directly obtain Neumann boundary conditions as a result of this projection (as opposed to the ordeal introduced by the integral conditions in the previous method).

Although both the standard discretization and the fractional-step methods can be interpreted as projection methods of sorts, the role that projection operators play in them is quite distinct. The method by Chorin and Temam hinges on the following two orthogonal decomposition theorems:

Theorem 1 (Ladyzhenskaya) Any vector field \( v \in V \) admits a unique orthogonal decomposition

\[
\begin{align*}
  v &= w + \nabla \phi \\
  w &= \text{solenoidal} (\nabla \cdot w = 0), \quad w \cdot n = 0 \text{ in } \partial \Omega, \quad \text{and of course } \nabla \phi \text{ is a potential (irrotational) field.}
\end{align*}
\]

Proof. (Sketch) The proof of this fact uses the vector identity \( \nabla \cdot (w \phi) = (\nabla \cdot w) \phi + w \cdot \nabla \phi \) to show that, if \( \nabla \cdot \omega = 0 \) and the divergence theorem is valid for \( v \in V \), then this and \( w \cdot n = 0 \) in \( \partial \Omega \) \( \iff \) \( w \) is orthogonal to any potential field. To prove existence of \( \phi \), we note that it is the solution of the Neumann problem:

\[
\begin{align*}
  \Delta \phi &= \nabla \cdot v \text{ in } \Omega \\
  \nabla \phi \cdot n &= v \cdot n \text{ in } \partial \Omega
\end{align*}
\]

In the case of nonhomogenous boundary data, we need this second decomposition theorem:

Theorem 2 (Temam) Any potential vector field \( v = \nabla \phi \) has a unique orthogonal decomposition

\[
\nabla \phi = \nabla \phi_0 + \nabla h
\]

in which \( \phi_0|_{\partial \Omega} = 0 \) and \( \Delta h = 0 \) (\( h \) is harmonic). Hence, \( \nabla h \) is solenoidal.

Proof. In a similar fashion as for the previous decomposition, \( \phi_0 \) and \( h \) are uniquely determined by:

\[
\begin{align*}
  \begin{cases}
    \Delta \phi_0 = \nabla \cdot v &; \phi_0|_S = 0 \\
    \Delta h = 0 &; n \cdot \nabla h|_S = n \cdot (v - \nabla \phi_0)|_S
  \end{cases}
\end{align*}
\]

We then write \( I - P = Q_0 + Q_h \), where \( Q_0 \) is a projection onto \( \{ \nabla \phi : \phi|_S = 0 \} \) and \( Q_h \) onto \( \{ \nabla \phi : \Delta h = 0 \} \).
**Projection Method Equations**

**STEP 1:** As promised, the intermediate velocity \( u^* \) is computed, is a discretization of the momentum equation in which the pressure term is omitted:

\[
\begin{align*}
\frac{u^*-u^n}{\Delta t} &= -(u^n \cdot \nabla)u^n + \nu \Delta u^n \\
 u^*|_{\partial \Omega} &= b^{n+1}
\end{align*}
\]

**STEP 2:** Using the decomposition theorems, \( u^* \) can be written \( u^{n+1} + \nabla \varphi \) where \( \varphi \) is proportional to the pressure. That is:

\[
\begin{align*}
\frac{u^{n+1}-u^*}{\Delta t} &= -\nabla P^{n+1} \\
\n \cdot u^{n+1} &= 0 \\
 n \cdot u^{n+1}|_{\partial \Omega} &= n \cdot b^{n+1}
\end{align*}
\]

**How exactly did I apply the theorems?** We consider the subspace of \( L^2(\Omega) \):

\[ J_0(\Omega) = \{ w \in L^2(\Omega) : \nabla \cdot w = 0 \ n \cdot w|_{S} = 0 \} \]

Then, our first decomposition theorem tells us that, if \( v \) is a vector field, it can be written as \( v = P_{h_0}[v] + (I - P_{h_0})[v] \).

Now, let's consider our intermediate velocity \( u^* \), \( w \) its solenoidal part. We want to find \( u^{n+1} \) with a prescribed boundary value \( n \cdot u^{n+1}|_{\partial \Omega} = n \cdot b^{n+1} \). Using our two decomposition theorems, we can write \( u^* \) as:

\[ u^* = w + \nabla \varphi_0 + \nabla h \]

I can then add and subtract \( \nabla h_{BC} \) where \( \Delta h_{BC} = 0 \) and \( n \cdot \nabla h_{BC} = n \cdot b \).

\[ u^* = [w + \nabla h_{BC}] + [\nabla (h - h_{BC}) + \nabla \varphi_0] \]

We can then define \( u^{n+1} = w + \nabla h_B \), and \( \Delta t \nabla P^{n+1} = \nabla (h - h_{BC}) + \nabla \varphi_0 \). Then:

\[ u^{n+1} = P_{\theta_0}[u^*] + \nabla h_B \]

1. We can then associate the projection out of the subspace \( \{ \nabla \varphi_0 \} \) with the incompressibility condition, since \( \nabla h \) and \( w \) are both solenoidal (since \( h \) is harmonic).

2. The space of harmonic flows \( \{ \nabla h \} \) gives us freedom to specify the normal component of \( u \) in the boundary. However, it is impossible to determine the tangential component (since the resulting Neumann problem for \( h_{BC} \) is uniquely solvable).

Hence, the boundary conditions for the velocity on the Navier-Stokes equation will not be adequate to work with this projection operator in one step. It is then necessary to break its treatment into two steps, in which the viscosity term and the incompressibility condition are respectively dealt with. This is precisely what the Chorin method does.
Poisson equation for the pressure   By taking the divergence of the first
equation in the second half-step, we obtain:

\[
\begin{align*}
-\Delta P^{n+1} &= -\frac{1}{\Delta t}\nabla \cdot u^* \\
n \cdot \nabla P^{n+1}|_S &= 0
\end{align*}
\]

Where the boundary condition is obtained by noting that \( u^*|_S = b^{n+1} \) and
\( n \cdot u^{n+1}|_S = n \cdot b^{n+1} \), and so \( n \cdot P^{n+1}|_S = -\frac{1}{\Delta t} n \cdot (u^{n+1} - u^*)|_S = 0 \).

Remark 3 The velocity \( u^{n+1} \) obtained after these two half-steps is only guar-
anteed to satisfy the normal component of the boundary condition. However,
since \( u^* \) does satisfy this condition, it is expected that the error in the tangential
components will be small.

1.1.3   Implementation of Chorin’s Method

• Temporal Discretization:

Reconsider the Navier-Stokes equations:

\[
\begin{align*}
\frac{\partial u}{\partial t} + u \cdot \nabla u &= -\frac{\partial p}{\partial x} + \frac{1}{Re} \Delta u \\
\frac{\partial v}{\partial t} + u \cdot \nabla v &= -\frac{\partial p}{\partial y} + \frac{1}{Re} \Delta v \\
\nabla \cdot u &= 0
\end{align*}
\]

Here it is written in terms of its components in order to make the next
step clear. We will rewrite the nonlinear terms in conservation form. By
the incompressibility condition we have

\[ u \cdot \nabla \phi = \nabla \cdot (u \phi) \]

Since

\[ u \cdot \nabla u = (u^2)_x + (uv)_y \]

and

\[ u \cdot \nabla v = (uv)_x + (v^2)_y \]

the Navier Stokes equations become

\[
\begin{align*}
\frac{\partial u}{\partial t} + (u^2)_x + (uv)_y &= -\frac{\partial p}{\partial x} + \frac{1}{Re} \Delta u \\
\frac{\partial v}{\partial t} + (uv)_x + (v^2)_y &= -\frac{\partial p}{\partial y} + \frac{1}{Re} \Delta v \\
\nabla \cdot u &= 0
\end{align*}
\]

Next, we apply Chorin’s projection method. In order to do this we need
to choose a time discretization. We choose to use Forward Euler on the
nonlinear terms and Backward Euler on the diffusion terms. Then Chorin’s
projection method gives the following equations:
First Step:

\[
\frac{U^* - U^n}{\Delta t} = -((U^n)^2)_x - (U^n V^n)_y + \frac{1}{Re} \Delta U^n \tag{7}
\]
\[
\frac{V^* - V^n}{\Delta t} = -(U^n V^n)_x - ((V^n)^2)_y + \frac{1}{Re} \Delta V^n \tag{8}
\]
\[
U^*|_{\partial \Omega} = b^{n+1} \tag{9}
\]

Second Step:

\[
-\Delta P^{n+1} = -\frac{1}{\Delta t} \nabla \cdot U^* \tag{10}
\]
\[
\nabla P^{n+1} \cdot n|_{\partial \Omega} = 0 \tag{11}
\]
\[
U^{n+1} = U^* - \Delta t \nabla P^{n+1} \tag{12}
\]

Remark 4 We have a CFL condition since we used an explicit discretization of the nonlinear terms. It takes the form \( h \leq C \Delta t \) where \( C \) depends on the magnitude of \( U^* \) and \( h \) is the mesh size in the spatial discretization.

- **Staggered Grid:**

  In order to solve the above equations we need to perform a spatial discretization. We would like to use central differences for the first derivatives as well as the Laplacian operator. Although these discretizations are well-defined on a regular grid, we obtain an instability called the odd-even decoupling or spurious pressure modes when we use regular grids.

  This instability is due to the relationship between the pressure and velocity. Notice that in the time-discretized equations, the velocity depends on the first derivatives of the pressure and the pressure depends on the first derivatives of the velocity. Hence, if we use central differences and we label the nodes odd or even (according to a checkerboard) then we find that the pressure at even nodes depends on the velocity at odd nodes and the pressure at odd nodes depends on the velocity at even nodes. Similarly for the velocity. Thus, it’s not surprising that this decoupling leads to a checkerboard effect.

  In order to avoid this instability we need to use a staggered grid. A staggered grid can be viewed in many ways, but the easiest is as follows. We divide the domain into cells. \( U \) is computed on the vertical edge midpoints, \( V \) on the horizontal edge midpoints, and \( P \) on the center of each cell (the stream function \( Q \) would be computed on the corners). It
is important, however, not to confuse this with an actual grid.

We note two important things:

1. The Poisson equation for the pressure is \( \Delta P = \nabla \cdot (U, V) = U_x + V_y \). We can then compute \( U_x \) and \( V_y \) with the second order centered difference formula, which means this divergence is computed at the center of the cell (where the pressure is defined).

2. The correction step is then of the form: \( (U, V)^{n+1} = (U, V)^n + \Delta t(P_x^{n+1}, P_y^{n+1}) \).
   Again, using the centered difference formula places \( P_x \) and \( P_y \) exactly on the same location as \( U \) and \( V \).

- In order to take care of the boundary conditions we need to add in extra boundary cells (shown in gray in the picture).

- **Spatial Discretization:** The centered differences are defined as:

\[
\phi_x = \frac{\phi_{i+\frac{1}{2},j} - \phi_{i-\frac{1}{2},j}}{h}
\]

and

\[
\phi_y = \frac{\phi_{i,j+\frac{1}{2}} - \phi_{i,j-\frac{1}{2}}}{h}
\]

When we discretize the first derivatives in this way then the linear terms involving first derivatives is defined at the correct positions. This is not
the case for the nonlinear terms; thus, we need to interpolate to the correct points in order to make sense of the centered differences. Details can be found in the documentation for the code.

The Laplacian is discretized as usual. Once we have discretized in space, we obtain equations of the form:

**First Step:**

\[
(1 - \frac{\Delta t}{Re})L U^* = g_1(U^n, V^n) \quad (13)
\]

\[
(1 - \frac{\Delta t}{Re})L V^* = g_2(U^n, V^n) \quad (14)
\]

where \( L \) is the discrete Laplacian operator and \( g_1, g_2 \) contain the nonlinear terms as well as boundary conditions.

**Second Step:**

\[
LP^{n+1} = g(D \cdot U^*) \quad (15)
\]

\[
U^{n+1} = U^* - \Delta t DP^{n+1} \quad (16)
\]

where \( D \) represents the discrete gradient.

Note: The boundary conditions for \( P^{n+1} \) is incorporated into the discrete equations. Also, the arbitrary constant is set so that the linear system in the second step has a unique solution.

**Implementation:**

Steps 1 and 2 both result in linear systems with sparse, symmetric positive definite matrices. We found a good example of the finite difference implementation in Gilbert Strang’s Computational Science for Engineering II course materials. This code uses Cholesky with minimum degree to solve these systems. The full documentation for this code and more can be found at http://www-math.mit.edu/cse/. We will go over some of this code’s salient features, and then present some numerical results produced by it.

**Solving the linear systems** The matrices for the Poisson equations are swiftly constructed using a tensor product trick: If \( L_x \) is the 1D discrete \(-\Delta\) or \(\Delta_y\) operator (in the case of \(-\Delta\), using the stencil \([1 -2 1]\)) in \([0, l_x]\), and \(L_y\) is the corresponding operator in \([0, l_y]\), then the discrete operator on the grid is obtained via a "Kronecker sum":

\[
L_{2D} = L_x \otimes L_y = L_x \otimes I_y + I_x \otimes L_y
\]

This generalizes to any discrete elliptic operator in tensor spaces (including graphs), and allows us to use powerful properties of the tensor product and sum. In particular, if

\[
L_x = \Phi \Lambda_x \Phi^T, \quad L_y = \Psi \Lambda_y \Psi^T
\]
are diagonalizations of $L_x$ and $L_y$, we know that

$$L_{2D} = (\Phi \otimes \Psi)(\Lambda_x \oplus \Lambda_y)(\Phi^T \otimes \Psi^T)$$

is a diagonalization for $L_{2D}$ (eigenvectors are tensor products and eigenvalues are sums).

Finally, the equations are solved using a direct solver, Cholesky factorization. It is well known that the Cholesky factor of a sparse matrix may not be sparse at all, a phenomena known as "fill-in" (while performing Gaussian elimination, zero entries are "filled"). The problem of finding a node ordering such that the fill-in is minimized arises: some methods to tackle this include nested dissection, Cuthill-McKee, column count and minimum degree.

In particular, the minimum degree algorithms exploit the fact that Cholesky factorization is equivalent to an undirected graph node elimination game. If we replace non-zero entries for 1’s on each step, this corresponds to eliminating one node at a time, and to join all its neighbors in a clique. Minimizing fill-in is then equivalent to minimizing the number of new edges created in this process, and requires us to always eliminate the node with minimum degree. Although this problem by itself is NP hard, heuristic approximations have astoundingly good performance in mitigating the fill-in.

Therefore, an approximate minimum degree algorithm (symamd.m command in matlab) to find a node ordering for which the fill-in is approximately minimized (the exact minimum degree / fill-in problem is an NP hard problem).

**Alternative Implementation methods**  We note that Chorin’s method, as presented before, can be discretized in space and implemented using a wide variety of methods:

1. A fast FMM Poisson solver can be used to solve the elliptic equations on each iteration
2. We can use Finite Element / Galerkin schemes for both $u$ and $p$.
3. Spectral Methods (using Chebyshev polynomials)
4. Finite Differences on staggered grids for pressure and velocities $u$ and $v$.

The resulting linear systems in the case of finite elements and finite differences are sparse, and so other methods like preconditioned CG, multigrid may be used.

**1.1.4 Higher Order Methods:**

In order to correct for the violation of the tangential boundary conditions, it may be desirable to introduce the diffusion term into the equations of the incompressibility step. For example, a Crank-Nicholson scheme would be:

\[
\begin{align*}
\frac{u^{n+1/2} - u^n}{\Delta t} &= -(u^n \cdot \nabla)u^n + \frac{1}{2} \nu \Delta u^n \quad ; \quad u^{n+1/2} |_{\partial \Omega} = b^{n+1} \\
\frac{v^{n+1/2} - v^n}{\Delta t} &= -\nabla P^{n+1} + \frac{1}{2} \nu \Delta v^{n+1} \quad ; \quad v^{n+1} |_{\partial \Omega} = b^{n+1}
\end{align*}
\]
This method provides a second-order accurate treatment of the viscous term, and the boundary condition is now entirely fulfilled. However, in a similar fashion to the standard (non-fractional) step method, the Poisson equation for the pressure now does not guarantee incompressibility of $u^{n+1}$, and we must again come up with complicated integral conditions for $P$ in order to ensure this. Hence, this method cannot be considered a bona fide projection method in the same sense as the first order version, although it retains much of its flavor.