Lecture 20

FEM and Navier-Stokes Equations: implicit method

(Lecture notes taken by Steven Burgess and Peter Gerakios)

• Implicit method and weak equations.

We have three equations of interest, the velocity in the “x” direction \(u\), the velocity in the “y” direction \(v\), and the incompressibility equation.

\[
\frac{u^{k+1} - u^k}{\Delta t} + (u^{k+1})_x^2 + (u^{k+1})y_0 - \mu \Delta u^{k+1} + p_x^{k+1} = f_x^{k+1}
\]

\[
\frac{v^{k+1} - v^k}{\Delta t} + (v^{k+1})_y^2 + (u^{k+1})_y_0 - \mu \Delta v^{k+1} + p_y^{k+1} = g_y^{k+1}
\]

\[
u^{x^{k+1}}_x + v^{y^{k+1}}_y = 0
\]

Initial condition can be given as:

\[u(x,y,0) = \text{given}\]

\[v(x,y,0) = \text{given}\]

And the boundary conditions can be given as:

\[u = u_1 \quad \text{on} \quad \partial\Omega_1\]

\[v = v_1 \quad \text{on} \quad \partial\Omega_1\]

\[\frac{dp}{dn} = 0 \quad \text{on} \quad ?\]

\[(u,v) = (0,0) \quad \text{on} \quad \partial\Omega_{\text{wall}}\]

(non-slip boundary condition)
Now multiply by a suitable test function \( \varphi \) (which equals zero on \( \partial \Omega \)) and integrate by parts to yield:

\[
\iint \left( \frac{u^{k+1} - u^k}{\Delta t} \right) \varphi - \iint (u^{k+1})^2 \varphi_x - \iint (u^{k+1} v^{k+1}) \varphi_y + \mu \iint (u_x^{k+1} \varphi_x + u_y^{k+1} \varphi_y)
\]
\[
\quad - \iint p^{k+1} \varphi_x = \iint f^{k+1} \varphi
\]

\[
\iint \left( \frac{v^{k+1} - v^k}{\Delta t} \right) \varphi - \iint (v^{k+1})^2 \varphi_y - \iint (u^{k+1} v^{k+1}) \varphi_x + \mu \iint (v_x^{k+1} \varphi_x + v_y^{k+1} \varphi_y)
\]
\[
\quad - \iint p^{k+1} \varphi_y = \iint g^{k+1} \varphi
\]
\[
\quad - \iint (u^{k+1} \varphi_x + v^{k+1} \varphi_y) = 0
\]

Now we want to discretize the weak forms.

- **FEM with mixed shape functions.**

Make the following replacements, where we will use higher order shape functions for the velocity components (quadratic shape functions, \( \varphi \)) and shape functions of one degree lower for the pressure component (linear shape functions, \( \overline{\varphi} \)):

\[
\begin{align*}
    u &\rightarrow \sum_j u_j \varphi_j \\
    v &\rightarrow \sum_j v_j \varphi_j \\
    \varphi &\rightarrow \varphi_i \\
    p &\rightarrow \sum_j p_j \overline{\varphi}_j
\end{align*}
\]
Substituting these expressions into the weak form equations, and with some work, yields the following matrices related to each component of the left hand side of the weak equations:

\[ B = \left[ \frac{1}{\Delta t} \iint \varphi_i \varphi_j \right] \]

\[ C_x(u) = \left[ \sum_i \iint \varphi_{ix} \varphi_j \varphi_i u_l \right] \]

\[ C_y(v) = \left[ \sum_i \iint \varphi_{iy} \varphi_j \varphi_i v_l \right] \]

\[ A = \left[ \mu \iint (\varphi_{ix} \varphi_{jx} + \varphi_{iy} \varphi_{jy}) \right] \]

\[ D_x = \left[ \iint \varphi_j \varphi_{ix} \right] \]

\[ D_y = \left[ \iint \varphi_j \varphi_{iy} \right] \]

And we get an expression for the “u” velocity component to be:

\[ B u^{k+1} - C_x(u^{k+1}) u^{k+1} - C_y(v^{k+1}) u^{k+1} + A u^{k+1} - D_x p^{k+1} = d_1 \]

Define the following:

\[ \beta(u^{k+1}, v^{k+1}) \equiv B - C_x(u^{k+1}) - C_y(v^{k+1}) + A , \]

We can write the velocity equations as follows:

**u equation:** \[ \beta(u^{k+1}, v^{k+1}) u^{k+1} - D_x p^{k+1} = d_1 \]

**v equation:** \[ \beta(u^{k+1}, v^{k+1}) v^{k+1} - D_y p^{k+1} = d_2 \]
The pressure equation requires slightly more work, but the process is the same as for the velocity equations. The pressure equation is derived as follows:

\[- \int \left( u^{k+1} \bar{\varphi}_x + v^{k+1} \bar{\varphi}_y \right) = 0 \]

\[- \int \sum_j u_j^{k+1} \varphi_j \bar{\varphi}_{lx} - \int \sum_j v_j^{k+1} \varphi_j \bar{\varphi}_{ly} = 0 \]

\[- \sum_j \int \varphi_j \bar{\varphi}_{lx} u_j^{k+1} - \sum_j \int \varphi_j \bar{\varphi}_{ly} v_j^{k+1} = 0 \]

Or, multiply by (-1) and integrate by parts to get

\[ p \text{ equation: } -D_x^T u^{k+1} - D_y^T v^{k+1} = 0 \]

And we can now form the following “equilibrium equations”, which are nonlinear

\[
\begin{bmatrix}
\beta(u^{k+1}, v^{k+1}) & 0 & -D_x \\
0 & \beta(u^{k+1}, v^{k+1}) & -D_y \\
-D_x^T & -D_y^T & 0
\end{bmatrix}
\begin{bmatrix}
u^{k+1} \\
v^{k+1} \\
p^{k+1}
\end{bmatrix} = \begin{bmatrix}d_1 \\
d_2 \\
0\end{bmatrix}
\]

- Equilibrium algebraic equations.

The general form of the equilibrium equations is given as follows:

\[
\begin{bmatrix}A & E^T \end{bmatrix} \begin{bmatrix}x \\
y\end{bmatrix} = \begin{bmatrix}g \end{bmatrix}
\]

Various applications of this “form” are given as, but not limited to:

1. Flow, Navier-Stokes \((x = \begin{bmatrix}u \\ v\end{bmatrix}, y = p)\) and “A” is a block diagonal matrix
2. Circuits (E is a conservation of currents at each node)
3. Structures ("A" is stiffness matrix)
• **Solution by block Gauss elimination method.**

“A” matrix is usually square and the “E” matrix is usually a skinny rectangular matrix.

From the general form of the equilibrium equations, we have:

\[ Ax + E^T y = g \]

and solving for “x” yields

\[ x = A^{-1}g - A^{-1}E^T y . \]

Plugging this expression for “x” into the “y” equation and simplifying yields

\[ Ex + 0y = f \]
\[ E(A^{-1}g - A^{-1}E^T y) = f \]

\[ EA^{-1}E^T y = -f + EA^{-1}g \]

Where we are assuming \( E^T \) is full column rank and “A” is a symmetric and positive definite (spd) matrix, making \( EA^{-1}E^T \) invertible.

Once “y” is obtained, solving for “x” proves to be trivial.

• **Solution by nullspace method.**

Assume the following:

\[ E = \begin{bmatrix} R_1 & R_2 \end{bmatrix} \quad and \quad EN = 0 \]
We then have:

$$\begin{bmatrix} R_1 & R_2 \\ R_1^{-1}R_2 & I \end{bmatrix} = 0$$

And “x” can be expressed in its most general form as follows:

$$x = Nx_0 + x_p.$$ 

We want the following:

$$Ex + 0 = f$$

$$E(Nx_0 + x_p) = f$$

$$ENx_0 + Exp = f$$

And from the equilibrium equations,

$$Ax + ETy = g$$

$$A(Nx_0 + xp) + ETy = g$$

$$N^T(ANx_0 + Ax_p) + N^TE^Ty = N^Tg$$

$$N^TANx_0 + N^TAx_p = N^Tg$$

$$N^TANx_0 = N^Tg - N^TAx_p.$$ 

Where we are assuming $N^TAN$ is invertible. Once “x” has been found, it remains to find “y” from

$$EAx + EE^Ty = Eg.$$ 

This is generally referred to as a dimension reduction scheme, with the name depending on the particular application.
• Nonlinear problems.

Example linear system:

\[ Ax = d \]

Example non-linear system:

\[ A(x)x = d(x) \]

One can solve the non-linear system sequentially using the following iterative equation:

\[ A(x^m)x^{m+1} = d(x^m) \]

\[ x^{m+1} = A(x^m)^{-1}d(x^m) = G(x^m) \]

This solution technique is referred to as Picard’s iterative scheme and a special case is Newton’s method.

One could also use Newton’s method to (in some instances) to obtain faster convergence to a solution. Simply find where the residuals equal zero to find the solution:

\[ F(x) = d(x) - A(x)x = 0 \]

Faster convergence with this method is only obtained if the initial guess is close to the correct solution.