CS205b/CME306

Lecture 18

The full Navier-Stokes equations are

$$\begin{aligned}
\rho_t + \nabla \cdot (\rho \mathbf{u}) &= 0 \\
(\rho \mathbf{u})_t + \nabla \cdot (\rho \mathbf{u}\mathbf{u}^T + p\mathbf{I}) &= \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} \\
E_t + \nabla \cdot ((E+p)\mathbf{u}) &= \nabla \cdot (\boldsymbol{\tau}\mathbf{u}) + \nabla \cdot (k \nabla T)
\end{aligned}$$

where T is the temperature, k is the thermal conductivity, and

$$\begin{aligned}
\boldsymbol{\tau} &= \begin{pmatrix} 2\mu u_x + \lambda(u_x + v_y) & \mu(u_y + v_x) \\ \mu(u_y + v_x) & 2\mu v_y + \lambda(u_x + v_y) \end{pmatrix} \\
&= \mu \begin{pmatrix} \nabla u \\ \nabla v \end{pmatrix} + \mu \begin{pmatrix} \nabla u \\ \nabla v \end{pmatrix}^T + \lambda(u_x + v_y)\mathbf{I} \\
&= \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \lambda(\nabla \cdot \mathbf{u})\mathbf{I}.
\end{aligned}$$

The parameter λ is often chosen to make the $\nabla \cdot \boldsymbol{\tau} = 0$. The latter criterion is called Stokes Hypothesis and results in $\lambda = -\frac{2}{3}\mu$ in 3D and $\lambda = -\mu$ in 2D. The Navier-Stokes equations simplify under the incompressibility assumption to

$$\begin{aligned}
\nabla \cdot \mathbf{u} &= 0 \\
\rho_t + \mathbf{u} \cdot \nabla \rho &= 0 \\
\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{\nabla p}{\rho} &= \frac{\nabla \cdot \boldsymbol{\tau}}{\rho} + \mathbf{g} \\
e_t + \mathbf{u} \cdot \nabla e &= \frac{\operatorname{tr}(\boldsymbol{\tau} \nabla \mathbf{u})}{\rho} + \frac{\nabla \cdot (k \nabla T)}{\rho} - \mathbf{u} \cdot \mathbf{g} \qquad (1)
\end{aligned}$$

where $\boldsymbol{\tau}$ simplifies to

$$\boldsymbol{\tau} = \mu \begin{pmatrix} 2u_x & u_y + v_x \\ u_y + v_x & 2v_y \end{pmatrix} = \mu \begin{pmatrix} \nabla u \\ \nabla v \end{pmatrix} + \mu \begin{pmatrix} \nabla u \\ \nabla v \end{pmatrix}^T = \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T).$$

1 Heat Equation

By removing the viscosity and forcing terms from equation 1 one has

$$e_t + \mathbf{u} \cdot \nabla e = \frac{\nabla \cdot (k \nabla T)}{\rho}. \qquad (2)$$

The assumptions that e and T satisfy the relationship

$$de = c_v dT$$

simplifies equation 2 to

$$c_v T_t + c_v \mathbf{u} \cdot \nabla T = \frac{\nabla \cdot (k \nabla T)}{\rho},$$

which can be further simplified to the standard heat equation

$$T_t = \frac{1}{\rho c_v} \nabla \cdot (k \nabla T) \tag{3}$$

by ignoring the effects of convection, i.e. setting $\mathbf{u} = 0$. (Note that the assumption $\mathbf{u} = 0$ will also eliminate the viscosity and forcing terms from the energy equation.) If k is constant, this can be written as

$$T_t = \frac{k}{\rho c_v} \Delta T.$$

Applying explicit Euler time discretization to equation 3 results in

$$\frac{T^{n+1} - T^n}{\Delta t} = \frac{1}{\rho c_v} \nabla \cdot (k \nabla T^n)$$

where either Dirichlet or Neumann boundary conditions can be applied on the boundaries of the computational domain. Assuming that ρ and c_v are constants allows us to rewrite this equation as

$$\frac{T^{n+1} - T^n}{\Delta t} = \nabla \cdot (\hat{k} \nabla T^n)$$

with $\hat{k} = \frac{k}{\rho c_v}$. Standard central differencing (second order accurate) can be used for the spatial derivatives as in

$$\frac{\hat{k}_{i+\frac{1}{2},j} \left(\frac{T_{i+1,j} - T_{i,j}}{\Delta x} \right) - \hat{k}_{i-\frac{1}{2},j} \left(\frac{T_{i,j} - T_{i-1,j}}{\Delta x} \right)}{\Delta x}$$

A time step restriction of

$$\Delta t \hat{k} \left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} + \frac{2}{\Delta z^2} \right) \leq 1$$

is needed for stability. If we $\Delta x = \Delta y$, then this is

$$2n \frac{\Delta t}{\Delta x^2} \hat{k} \leq 1,$$

where n is the dimension ($n = 2$ in 2D and $n = 3$ in 3D).

Implicit Euler time discretization

$$\frac{T^{n+1} - T^n}{\Delta t} = \nabla \cdot (\hat{k} \nabla T^{n+1}) \tag{4}$$

avoids this time step stability restriction. This equation can be rewritten as

$$T^{n+1} - \Delta t \nabla \cdot (\hat{k} \nabla T^{n+1}) = T^n \tag{5}$$

discretizing the $\nabla \cdot (\hat{k}\nabla T^{n+1})$ term using central differencing. For each unknown, T_i^{n+1}, equation 5 is used to fill in one row of a matrix creating a linear system of equations. Since the resulting matrix is symmetric, a number of fast linear solvers can be used (e.g. a PCG method with an incomplete Choleski preconditioner, see Golub and Van Loan [1]). Equation 4 is first order accurate in time and second order accurate in space, and Δt needs to be chosen proportional to Δx^2 in order to obtain an overall asymptotic accuracy of $O(\Delta x^2)$. However, the stability of the implicit Euler method allows one to chose Δt proportional to Δx saving dramatically on CPU time. The Crank-Nicolson scheme

$$\frac{T^{n+1} - T^n}{\Delta t} = \frac{1}{2}\nabla \cdot (\hat{k}\nabla T^{n+1}) + \frac{1}{2}\nabla \cdot (\hat{k}\nabla T^n)$$

can be used to achieve second order accuracy in both space and time with Δt proportional to Δx. For the Crank-Nicolson scheme,

$$T^{n+1} - \frac{\Delta t}{2}\nabla \cdot (\hat{k}\nabla T^{n+1}) = T^n + \frac{\Delta t}{2}\nabla \cdot (\hat{k}\nabla T^n)$$

is used to create a symmetric linear system of equations for the unknowns T_i^{n+1}. Again, all spatial derivatives are computed using standard central differencing.

Why not always use Crank-Nicholson, as it gives second order accuracy and no time step restriction? Let us look at the solution as $\Delta t \to \infty$. Backward Euler gives

$$\Delta T^n = 0,$$

which is the correct steady state solution. Crank-Nicholson gives

$$\Delta T^{n+1} = -\Delta T^n.$$

In $1D$ this is

$$T_{xx}^{n+1} = -T_{xx}^n$$

This shows that the curvature is changing sign at each time step. So the problem with Crank-Nicholson is that as Δt gets very large, you get oscillations, whereas with backward Euler, you get the steady-state solution.

References

[1] Golub, G. and Van Loan, C., *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1989.