Date: May 4, 2020
Author:

- Rajib Roy
- roy.rajib@live.com
- PhD candidate at the University of Wyoming, Laramie, USA

Crank Nicolson scheme derivation

The Crank-Nicolson temporal discretization scheme can be viewed as summation of implicit and explicit Euler discretization schemes.

For any variable ϕ, let assume the linear system for the partial differential equation is written as:

$$\frac{\partial \phi}{\partial t} = \mathbf{S}(\phi), \tag{1}$$

where $\mathbf{S}$ is the spatial discretization operator.

The temporal derivatives can discretized as :

Implicit Euler

$$\frac{\partial \phi^{t+\frac{1}{2}}}{\partial t} = \mathbf{S}(\phi^{t+\frac{1}{2}})$$

$$\frac{\phi^{t+\frac{1}{2}} - \phi^t}{(\Delta t/2)} = \mathbf{S}(\phi^{t+\frac{1}{2}}) \tag{2}$$

Explicit Euler

$$\frac{\partial \phi^{t+1}}{\partial t} = \mathbf{S}(\phi^{t+\frac{1}{2}})$$

$$\frac{\phi^{t+1} - \phi^{t+\frac{1}{2}}}{(\Delta t/2)} = \mathbf{S}(\phi^{t+\frac{1}{2}}) \tag{3}$$

Adding equations 2 and 3, we get

$$\frac{\phi^{t+\frac{1}{2}} - \phi^t}{(\Delta t/2)} + \frac{\phi^{t+1} - \phi^{t+\frac{1}{2}}}{(\Delta t/2)} = 2 * \mathbf{S}(\phi^{t+\frac{1}{2}})$$

$$\frac{\phi^{t+1} - \phi^t}{(\Delta t)} = \mathbf{S}(\phi^{t+\frac{1}{2}})$$

Final form of the discretization becomes:

$$\frac{\phi^{t+1} - \phi^t}{\Delta t} = \frac{1}{2}\left[\mathbf{S}(\phi^{t+1}) + \mathbf{S}(\phi^t)\right] \tag{4}$$

The temporal term is similar to the implicit Euler scheme, and key difference is in semi implicit treatment of the spatial operators.

Now, let's assume that an unsteady simulation is performed where is time-step the linear system is solved to tight tolerance at each time step before proceeding to next time-step. Thus, equation 1 at any time step$t$ can be approximated as,

$$\frac{\partial \phi^t}{\partial t} = \mathbf{S}(\phi^t) + \mathbf{R}(t), \tag{5}$$

where $\mathbf{R}(t)$ is the initial residual of the linear system at time-step t. We can also express the spatial operators in terms of the temporal derivative and residual. For example, the equation 5 can be reformulated as

$$\mathbf{S}(\phi^t) = \frac{\partial \phi^t}{\partial t} - \mathbf{R}(t). \tag{6}$$

Substituting, $\mathbf{S}(\phi^t)$ value in the Crank-Nicolson equation 2, we get

$$\frac{\phi^{t+1} - \phi^t}{\Delta t} = \frac{1}{2}\left[\mathbf{S}(\phi^{t+1}) + \frac{\partial \phi^t}{\partial t} - \mathbf{R}(t)\right],$$

$$2\frac{\phi^{t+1} - \phi^t}{\Delta t} - \frac{\partial \phi^t}{\partial t} + \mathbf{R}(t) = \mathbf{S}(\phi^{t+1}). \tag{7}$$

OpenFOAM applies this form of the CrankNicolson scheme with assumption that the linear system residual are driven to machine zero $\mathbf{R}(t) \rightarrow 0$ or atleast insignificantly small.

Usually in PISO like segregated algorithm, driving the linear system residual to machine zero is very expensive and computationally inefficient. It might be the reason the CrankNicolson scheme is recommended to be used with small blending with implicit Euler scheme.

I tested the Crank-Nicolson scheme in a projection algorithm (one outer loop only) where the initial residual are small but higher than PISO type algorithm multiple outer loops. The Crank-Nicolson scheme in the format of equation7 is unstable, but original format (equation 4 is stable.