# The Alternating Direction Implicit (ADI) Method

ADI is actually a family of methods. In it's simplest form ADI consists of the following two equations, evaluated at each time step.

$$\frac{1}{\alpha}\frac{\widetilde{T}_{i,j}^{n+1}-T_{i,j}^{n}}{\Delta t}=\frac{T_{i,j+1}^{n}-2\,T_{i,j}^{n}+T_{i,j-1}^{n}}{\Delta y^2}+\frac{\widetilde{T}_{i+1,j}^{n+1}-2\widetilde{T}_{i,j}^{n+1}+\widetilde{T}_{i-1,j}^{n+1}}{\Delta x^2}+\dddot{q}$$

$$\frac{1}{\alpha}\frac{T_{i,j}^{n+1}-T_{i,j}^{n}}{\Delta t}=\frac{T_{i,j+1}^{n+1}-2\,T_{i,j}^{n+1}+T_{i,j-1}^{n+1}}{\Delta y^2}+\frac{\widetilde{T}_{i+1,j}^{n+1}-2\widetilde{T}_{i,j}^{n+1}+\widetilde{T}_{i-1,j}^{n+1}}{\Delta x^2}+\dddot{q}$$

To simplify notation define:

$$\beta_x=\frac{\alpha\Delta t}{\Delta x^2}\quad\text{and}\quad\beta_y=\frac{\alpha\Delta t}{\Delta y^2}$$

Gathering unknown terms on the left of the equations, we first must solve the linear system:

$$-\beta_x\widetilde{T}_{i-1,j}^{n+1}+\left(1+2\beta_x\right)\widetilde{T}_{i,j}^{n+1}-\beta_x\widetilde{T}_{i+1,j}^{n+1}=\beta_y T_{i,j-1}^{n}+\left(1-2\beta_y\right)T_{i,j}^{n}+\beta_y T_{i,j-1}^{n}+\dddot{q}\alpha\Delta t$$

For each value of j, this represents a tridiagonal linear system in the form:

$$\begin{pmatrix} 1+2\beta_x & -\beta_x & 0 & 0 & \cdots & & 0 & 0 \\ -\beta_x & 1+2\beta_x & -\beta_x & 0 & & & & \\ 0 & -\beta_x & 1+2\beta_x & -\beta_x & 0 & & & \\ . & & & & & & & . \\ . & & & & & & & . \\ . & & & & & & 0 & . \\ & & \cdots & & 0 & -\beta_x & 1+2\beta_x & -\beta_x & 0 \\ & & & & & 0 & -\beta_x & 1+2\beta_x & -\beta_x \\ & & & & & & 0 & -\beta_x & 1+2\beta_x \end{pmatrix} \begin{pmatrix} \widetilde{T}^{n+1}_{1,j} \\ \widetilde{T}^{n+1}_{2,j} \\ \widetilde{T}^{n+1}_{3,j} \\ . \\ . \\ . \\ \widetilde{T}^{n+1}_{nx-2,j} \\ \widetilde{T}^{n+1}_{nx-1,j} \\ \widetilde{T}^{n+1}_{nx,j} \end{pmatrix} = (1-2\beta_y) \begin{pmatrix} T^n_{1,j} \\ T^n_{2,j} \\ T^n_{3,j} \\ . \\ . \\ . \\ T^n_{nx-2,j} \\ T^n_{nx-1,j} \\ T^n_{nx,j} \end{pmatrix} + \beta_y \begin{pmatrix} T^n_{1,j+1} \\ T^n_{2,j+!} \\ T^n_{3,j+1} \\ . \\ . \\ . \\ T^n_{nx-2,j+1} \\ T^n_{nx-1,j+1} \\ T^n_{nx,j+1} \end{pmatrix} + \beta_y \begin{pmatrix} T^n_{1,j-1} \\ T^n_{2,j-1} \\ T^n_{3,j-1} \\ . \\ . \\ . \\ T^n_{nx-2,j-1} \\ T^n_{nx-1,j-1} \\ T^n_{nx,j-1} \end{pmatrix} + \begin{pmatrix} s \\ o \\ u \\ r \\ c \\ e \end{pmatrix}$$

From a computational perspective it is important to note that solution of equation above for any given j does not require previous solution of the equivalent equation for any other value of j. For ny grid points in the y direction, we can productively use ny processors on a parallel computer.

Now that we have known values for every $\widetilde{T}^{n+1}_{i,j}$, the second equation can be arranged for solution in the form:

$$-\beta_y T^{n+1}_{i,j-1} + (1+2\beta_y) T^{n+1}_{i,j} - \beta_y T^{n+1}_{i,j-1} = T^n_{i,j} + \beta_x \widetilde{T}^{n+1}_{i-1,j} - 2\beta_x \widetilde{T}^{n+1}_{i,j} + \beta_x \widetilde{T}^{n+1}_{i+1,j} + \ddot{q}\alpha\Delta t$$

For each value of i, this represents a tridiagonal linear system in the form:

$$
\begin{pmatrix}
1+2\beta_y & -\beta_y & 0 & 0 & \cdots & & & 0 & 0 \\
-\beta_y & 1+2\beta_y & -\beta_y & 0 & & & & & \\
0 & -\beta_y & 1+2\beta_y & -\beta_y & 0 & & & & \\
. & & & & & & & & . \\
. & & & & & & & & . \\
. & & & & & & & 0 & . \\
 & & \cdots & & 0 & -\beta_y & 1+2\beta_y & -\beta_y & 0 \\
 & & & & & 0 & -\beta_y & 1+2\beta_y & -\beta_y \\
 & & & & & & 0 & -\beta_y & 1+2\beta_y
\end{pmatrix}
\begin{pmatrix} T_{i,1}^{n+1} \\ T_{i,2}^{n+1} \\ T_{i,3}^{n+1} \\ . \\ . \\ . \\ T_{i,ny-2}^{n+1} \\ T_{i,ny-1}^{n+1} \\ T_{i,ny}^{n+1} \end{pmatrix}
=
\begin{pmatrix} T_{i,1}^{n} \\ T_{i,2}^{n} \\ T_{i,3}^{n} \\ . \\ . \\ . \\ T_{i,ny-2}^{n} \\ T_{i,ny-1}^{n} \\ T_{i,ny}^{n} \end{pmatrix}
-2\beta_x
\begin{pmatrix} \widetilde{T}_{i,1}^{n+1} \\ \widetilde{T}_{i,2}^{n+1} \\ \widetilde{T}_{i,3}^{n+1} \\ . \\ . \\ . \\ \widetilde{T}_{i,ny-2}^{n+1} \\ \widetilde{T}_{i,ny-1}^{n+1} \\ \widetilde{T}_{i,ny}^{n+1} \end{pmatrix}
+\beta_x
\begin{pmatrix} \widetilde{T}_{i+1,1}^{n+1} \\ \widetilde{T}_{i+1,2}^{n+1} \\ \widetilde{T}_{i+1,3}^{n+1} \\ . \\ . \\ . \\ \widetilde{T}_{i+1,ny-2}^{n+1} \\ \widetilde{T}_{i+1,ny-1}^{n+1} \\ \widetilde{T}_{i+1,ny}^{n+1} \end{pmatrix}
+\beta_x
\begin{pmatrix} \widetilde{T}_{i-1,1}^{n+1} \\ \widetilde{T}_{i-1,2}^{n+1} \\ \widetilde{T}_{i-1,3}^{n+1} \\ . \\ . \\ . \\ \widetilde{T}_{i-1,ny-2}^{n+1} \\ \widetilde{T}_{i-1,ny-1}^{n+1} \\ \widetilde{T}_{i-1,ny}^{n+1} \end{pmatrix}
+
\begin{pmatrix} s \\ o \\ u \\ r \\ c \\ e \\ \\ \\ \\ \end{pmatrix}
$$

The combination of the two equations is an analog to a fully implicit method. Like a fully implicit method it is unconditionally stable, and has accuracy $O(\Delta t, \Delta x^2, \Delta y^2)$. With a little creativity an analog to the Crank-Nicholson method can be created with accuracy $O(\Delta t^2, \Delta x^2, \Delta y^2)$.