8/03/20 **Calculating steady-state temperature:**

- Use the FDM + Eulers equation to develop a s.s equation.
- Implement into the script and run comparison with the matlab reference.

Steady-state in example matlab code.

$$T_{i,j} = \frac{1}{4}\cdot\left(T_{i+1,j} + T_{i,j+1} + T_{i-1,j} + T_{i,j-1}\right)$$

They are simply taking the average of the surrounding points until the difference from one iteration to the next becomes less than a pre-defined tolerance.

Original eqn.

$$T^{k+1}_{i,j} = T^{k}_{i,j} + \Delta t \cdot \alpha \cdot \left[\left(\frac{T^{k}_{i-1,j} - 2\cdot T^{k}_{i,j} + T^{k}_{i+1,j}}{\Delta y^2}\right) + \left(\frac{T^{k}_{i,j-1} - 2\cdot T^{k}_{i,j} + T^{k}_{i,j+1}}{\Delta x^2}\right)\right]$$

re-arrange for $\frac{\partial T}{\partial t}\Big|_t$ becomes...

$$\frac{T^{k+1}_{i,j} - T^{k}_{i,j}}{\Delta t} = \alpha \cdot \left[\left(\frac{T^{k}_{i-1,j} - 2\cdot T^{k}_{i,j} + T^{k}_{i+1,j}}{\Delta y^2}\right) + \left(\frac{T^{k}_{i,j-1} - 2\cdot T^{k}_{i,j} + T^{k}_{i,j+1}}{\Delta x^2}\right)\right]$$

(a) s.s $\frac{\partial T}{\partial t} = 0$ →

$$(a) = \alpha \cdot \left[\left(\frac{T^{k}_{i-1,j} - 2\cdot T^{k}_{i,j} + T^{k}_{i+1,j}}{\Delta y^2}\right) + \left(\frac{T^{k}_{i,j-1} - 2\cdot T^{k}_{i,j} + T^{k}_{i,j+1}}{\Delta x^2}\right)\right]$$

Re-arrange for $T_{i,j}$

$$0 = \alpha \cdot \left(\frac{T^k_{i-1,j} - 2\cdot T^k_{i,j} + T^k_{i+1,j}}{\Delta y^2} \right) + \alpha \cdot \left(\frac{T^k_{i,j-1} - 2\cdot T^k_{i,j} + T^k_{i,j+1}}{\Delta x^2} \right)$$

multiply by $\Delta y^2 \cdot \Delta x^2$

$$0 = \alpha \cdot \Delta x^2 \left(T^k_{i-1,j} - 2\cdot T^k_{i,j} + T^k_{i+1,j} \right) + \alpha \cdot \Delta y^2 \cdot \left(T^k_{i,j-1} - 2\cdot T^k_{i,j} + T^k_{i,j+1} \right)$$

Expand the brackets (removing k notation)

$$0 = \alpha \cdot \Delta x^2 \cdot T_{i-1,j} - 2\cdot\alpha\cdot\Delta x^2 \cdot T_{i,j} + \alpha \cdot \Delta x^2 \cdot T_{i+1,j} + \alpha \cdot \Delta y^2 \cdot T_{i,j-1} - 2\cdot\alpha\cdot\Delta y^2 \cdot T_{i,j} + \alpha \cdot \Delta y^2 \cdot T_{i,j+1}$$

Collect like terms of $T_{i,j}$ and re-arrange.

$$T_{i,j} \cdot 2\alpha \cdot (\Delta x^2 + \Delta y^2) = \alpha \cdot \Delta x^2 \cdot T_{i-1,j} + \alpha \cdot \Delta x^2 \cdot T_{i+1,j} + \alpha \cdot \Delta y^2 \cdot T_{i,j-1} + \alpha \cdot \Delta y^2 \cdot T_{i,j+1}$$

$$T_{i,j} = \frac{\alpha \cdot \Delta x^2 \cdot T_{i-1,j} + \alpha \cdot \Delta x^2 \cdot T_{i+1,j} + \alpha \cdot \Delta y^2 \cdot T_{i,j-1} + \alpha \cdot \Delta y^2 \cdot T_{i,j+1}}{2\alpha \cdot (\Delta x^2 + \Delta y^2)}$$

- Cancel out α from eqn. $\alpha = {}^{k_T}/_{c_p \cdot \rho}$, this may appear in-correct however if you allow $t \to \infty$ the final s-s conditions will be reached regardless of the physical conditions it occupies.

$$T_{i,j} = \frac{\Delta x^2 \cdot T_{i-1,j} + \Delta x^2 \cdot T_{i+1,j} + \Delta y^2 \cdot T_{i,j-1} + \Delta y^2 \cdot T_{i,j+1}}{2 \cdot (\Delta x^2 + \Delta y^2)}$$

$$T_{i,j} = \frac{\Delta x^2 \cdot (T_{i-1,j} + T_{i+1,j}) + \Delta y^2 \cdot (T_{i,j-1} + T_{i,j+1})}{2 \cdot (\Delta x^2 + \Delta y^2)}$$

Use this equation to solve for s-s

- This equation has worked and have produced the same result - albeit with slightly different time conditions in that my model claims to have solved it sooner,

- This difference is likely due to the slightly different dimensions?

  L my fly is 0.025, and in matlab reference fly is 0.0263

  L modify the matlab script to not treat boundary conditions as external?

Results: At this time I did not record results but here are some notes given what I came across in the tests.

- Before modifying the matlab script my model seemed to reach S.S sooner than ML
- After modification, my script would be slightly slower than the ML version.
- I should do a check between my S.S equation and the one used in the reference. and also double check my derivation.

However - in terms of accuracy both scripts are calculating closely the same values.