Appendix A

Heat equation using finite differences

A.0.1 Energy conservation

The heat equation is a beautiful equation that can be derived from first principles. Stating the energy conservation law leads to,

$$\frac{d}{dt}\Big(\underbrace{\int_D \rho \cdot c_p \cdot T \cdot dv}_{\text{total energy}}\Big) = \underbrace{\int_{\partial D} k\nabla \cdot T \cdot n \cdot dS}_{\text{power through boundaries}} + \underbrace{\int_D q \cdot dv}_{\text{power source term}}$$

which becomes by introducing the divergence operator,

$$\frac{d}{dt}\Big(\int_D \rho \cdot c_p \cdot T \cdot dv\Big) = \int_D \nabla \cdot (k\nabla \cdot T)dv + \int_D q \cdot dv$$

which means for every point of the domain D,

$$\frac{\partial(\rho \cdot c_p \cdot T)}{\partial t} = \nabla \cdot (k \cdot \nabla T) + q$$

If we assume the density and specific heat of the D are time-independent,

$$\rho \cdot c_p \cdot \frac{\partial T}{\partial t} = \nabla \cdot (k \cdot \nabla T) + q$$

Now, if we add the assumption that isotropy (same conductivity in all directions) on D,

$$\rho \cdot c_p \cdot \frac{\partial T}{\partial t} = k \cdot \nabla^2 T + q$$

Finally, if we limit ourselves to 2 dimensions and suppose there is no energy input in our system,

$$\frac{\partial T}{\partial t} = \frac{k}{\rho \cdot c_p}\Big(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\Big) \tag{A.1}$$

A.0.2 Discretizing time and space

In its simplest form, finite differences consist in discretizing the *time* and *space* domains as follows,

$$\Big(\frac{\partial T}{\partial t}\Big)^n_{i,j} \approx \frac{T^{n+1}_{i,j} - T^n_{i,j}}{\Delta t} \tag{A.2}$$

$$\left(\frac{\partial^2 T}{\partial x^2} \right)_{i,j}^{n} \approx \frac{T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}}{\Delta x^2} \tag{A.3}$$

$$\left(\frac{\partial^2 T}{\partial y^2} \right)_{i,j}^{n} \approx \frac{T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}}{\Delta y^2} \tag{A.4}$$

where n is a component of the discretized time vector, (i, j) are the coordinates of a point in the xy plane, Δt, the time step and Δx and Δy the space between points on the desired computation domain. By combining the expressions above and the equation A.1, the temperature can be computed on every point of the computation domain as a function of the previous time step,

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \Delta t \frac{k}{\rho \cdot c_p} \left(\frac{T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}}{\Delta x^2} + \frac{T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}}{\Delta y^2} \right) \tag{A.5}$$

A.0.3 Numerically stable solutions

The previous equation can be recasted in the following matrix form,

$$T^{n+1} = \left(\frac{\alpha \Delta t}{h^2} \cdot A + I_n \right) \cdot T_n + \frac{\alpha \Delta t}{h^2} \cdot B \tag{A.6}$$

where B encapsulates static boundary conditions (as the ones used in this work), $h = \Delta x = \Delta y$ and $\alpha = \frac{k}{\rho \cdot c_p}$ is the *diffusion* coefficient. To reduce the computational cost, it is recommended when using this formulation to specify in MATLAB or equivalent that A is a sparse matrix.

The expression A.6 is a finite arithmetic progression, so to make sure it will converge, nonnegativity is required,

$$\frac{\alpha \Delta t}{h^2} \cdot A + I_n \geq 0 \tag{A.7}$$

By looking at the smallest terms, which are in the diagonal, we get the largest time step allowed for numerical stability,

$$\Delta t \leq \frac{h^2}{\gamma \cdot \alpha} \tag{A.8}$$

where $\gamma = 4$. All the previous steps can be generalized to the 3D case, one would find $\gamma = 6$.