Solving the heat equation

Charles Xie

The heat conduction for heterogeneous media is modeled using the following partial differential equation:

$$\rho c\left[\frac{\partial T}{\partial t}+\nabla\cdot\left(\mathbf{v}T\right)\right]=\nabla\cdot\left[k\nabla T\right]+q \qquad (1)$$

where k is the thermal conductivity, c is the specific heat capacity, ρ is the density, $\mathbf{v}$ is the velocity field, and q is the internal heat generation. The heat equation has two parts: the diffusion part characterized by the conductivity field function and the advection part characterized by the velocity field function. The internal heat generation term can be thought of as an external force.

The partial differential equation is discretized in both space and time domain in order to find approximate solutions. Sometimes, this numeric method is called the **finite-difference time-domain** (FDTD) method. An implicit FDTD method can be used to achieve better numeric stability:

$$\begin{aligned}
&\rho_{i,j,k}c_{i,j,k}\left\{\begin{array}{l}\dfrac{T_{i,j,k}^{n+1}-T_{i,j,k}^{n}}{\Delta t}+\dfrac{u_{i+1,j,k}^{n+1}T_{i+1,j,k}^{n+1}-u_{i-1,j,k}^{n+1}T_{i-1,j,k}^{n+1}}{2\Delta x}+\\ \dfrac{v_{i,j+1,k}^{n+1}T_{i,j+1,k}^{n+1}-v_{i,j-1,k}^{n+1}T_{i,j-1,k}^{n+1}}{2\Delta y}+\dfrac{w_{i,j,k+1}^{n+1}T_{i,j,k+1}^{n+1}-w_{i,j,k-1}^{n+1}T_{i,j,k-1}^{n+1}}{2\Delta z}\end{array}\right\}\\
&=\frac{1}{\Delta x}\left\{k_{i+1/2,j,k}\frac{T_{i+1,j,k}^{n+1}-T_{i,j,k}^{n+1}}{\Delta x}-k_{i-1/2,j,k}\frac{T_{i,j,k}^{n+1}-T_{i-1,j,k}^{n+1}}{\Delta x}\right\}\\
&+\frac{1}{\Delta y}\left\{k_{i,j+1/2,k}\frac{T_{i,j+1,k}^{n+1}-T_{i,j,k}^{n+1}}{\Delta y}-k_{i,j-1/2,k}\frac{T_{i,j,k}^{n+1}-T_{i,j-1,k}^{n+1}}{\Delta y}\right\}\\
&+\frac{1}{\Delta z}\left\{k_{i,j,k+1/2}\frac{T_{i,j,k+1}^{n+1}-T_{i,j,k}^{n+1}}{\Delta z}-k_{i,j,k-1/2}\frac{T_{i,j,k}^{n+1}-T_{i,j,k-1}^{n+1}}{\Delta z}\right\}+q_{i,j,k}^{n} \qquad (2)
\end{aligned}$$

where $M>i\geq 0$, $N>j\geq 0$, and $L>k\geq 0$, with M, N, and L representing the numbers of cubic voxels in x, y, and z directions, and Δx, Δy, and Δz are the lengths of the cubic voxels in the three directions, respectively. Δt is the time step. In theory, **an implicit method** is *unconditionally* stable, allowing any time step to be taken, and the simulation will not "blow" up[1]. Note that in Eq. (2) we also avoid directly calculating the spatial derivatives of the conductivity function, which would be very large at the interfaces between two distinct kinds of media and therefore would cause numeric instability.

Let:

[1] In practice, the advection part of the heat equation can cause numeric instability, especially when the velocity is high.

$$\Gamma_{i,j,k} = \frac{\rho_{i,j,k} c_{i,j,k}}{\Delta t}$$

$$A^x_{i,j,k} = \frac{k_{i-1/2,j,k}}{(\Delta x)^2} = \frac{k_{i-1,j,k} + k_{i,j,k}}{2(\Delta x)^2}, B^x_{i,j,k} = \frac{k_{i+1/2,j,k}}{(\Delta x)^2} = \frac{k_{i,j,k} + k_{i+1,j,k}}{2(\Delta x)^2}$$

$$A^y_{i,j,k} = \frac{k_{i,j-1/2,k}}{(\Delta y)^2} = \frac{k_{i,j-1,k} + k_{i,j,k}}{2(\Delta y)^2}, B^y_{i,j,k} = \frac{k_{i,j+1/2,k}}{(\Delta y)^2} = \frac{k_{i,j,k} + k_{i,j+1,k}}{2(\Delta y)^2}$$

$$A^z_{i,j,k} = \frac{k_{i,j,k-1/2}}{(\Delta z)^2} = \frac{k_{i,j,k-1} + k_{i,j,k}}{2(\Delta z)^2}, B^z_{i,j,k} = \frac{k_{i,j,k+1/2}}{(\Delta z)^2} = \frac{k_{i,j,k} + k_{i,j,k+1}}{2(\Delta z)^2}$$

$$C^x_{i,j,k} = A^x_{i,j,k} + \frac{\rho_{i,j,k} c_{i,j,k} u^{n+1}_{i-1,j,k}}{2\Delta x}, D^x_{i,j,k} = B^x_{i,j,k} - \frac{\rho_{i,j,k} c_{i,j,k} u^{n+1}_{i+1,j,k}}{2\Delta x}$$

$$C^y_{i,j,k} = A^y_{i,j,k} + \frac{\rho_{i,j,k} c_{i,j,k} v^{n+1}_{i,j-1,k}}{2\Delta y}, D^y_{i,j,k} = B^y_{i,j,k} - \frac{\rho_{i,j,k} c_{i,j,k} v^{n+1}_{i,j+1,k}}{2\Delta y}$$

$$C^z_{i,j,k} = A^z_{i,j,k} + \frac{\rho_{i,j,k} c_{i,j,k} w^{n+1}_{i,j,k-1}}{2\Delta z}, D^z_{i,j,k} = B^z_{i,j,k} - \frac{\rho_{i,j,k} c_{i,j,k} w^{n+1}_{i,j,k+1}}{2\Delta z} \qquad (3)$$

Eq. (2) can be rewritten as:

$$T^{n+1}_{i,j,k} = \frac{\Gamma_{i,j,k} T^n_{i,j,k} + q^n_{i,j,k} + C^x_{i,j,k} T^{n+1}_{i-1,j,k} + D^x_{i,j,k} T^{n+1}_{i+1,j,k} + C^y_{i,j,k} T^{n+1}_{i,j-1,k} + D^y_{i,j,k} T^{n+1}_{i,j+1,k} + C^z_{i,j,k} T^{n+1}_{i,j,k-1} + D^z_{i,j,k} T^{n+1}_{i,j,k+1}}{\Gamma_{i,j,k} + A^x_{i,j,k} + B^x_{i,j,k} + A^y_{i,j,k} + B^y_{i,j,k} + A^z_{i,j,k} + B^z_{i,j,k}} \qquad (4)$$

This is a very sparse matrix that can be solved efficiently using **a relaxation method**[2].

Two boundary conditions can be used. The first kind is the Dirichilet boundary condition, which maintains the temperature at the boundary. The second kind is the Neumann boundary condition, which maintains the flux at the boundary.

[2] http://en.wikipedia.org/wiki/Relaxation_method