Abstract:

Traditionaly the problem of simulating Heat Transfer has been one of complex mathematical models that can only accommodate a homogeneous material. This is of course a problem since you don't run into homogenous materials in real life. About all you could do is tell how long it would take for x amount of watts to travel through a substance, but with computers that substance can be broken down into many small pieces and those pieces can individually be solved for allowing for a heat dispersion pattern to be found in a substance.

Introduction:

The threory of modeling heat transfer has been around for a rather long time, but without computers to preform complex, redundant, and accurate computations quickley scientists and matematiticians have come up with many ways to simplify things so they can be finished in a series of equations instead of doing the same thing over and over again. Our approach is based off of Cellular Automata since it takes a set of variables gathered from a location's neighbors and then inputs those values into a formula and then it moves onto the next location. This method requires for the same set of equations to be used for each and every cell each time that the grid updates. Thus a 30 X 30 grid, which is considered so small or so rough you can only use it for validatation and estimates, suddenly requires you to preform the same equations 900 times over for a single timestep.

Background:

In the past all the simulations relied on the finite difference method almost exclusivley. While the finite difference method is a very viable solution when there is no need to simulate convection. However, when you want to be able to simulate convection the finite difference method is not computationaly feasable since it is almost excessivley intensive in a computational manner.

Lattice Gas Automata have been well documented for their potential to simulate hydrodynamic systems. In the process of modeling Heat Transfer the material will eventually melt into a very viscous material when that happens you need a way to simulate the flow of the liquid, the conduction heat transfer, and the convection heat transfer.

Theoretical Approach:

To find the heat conduction values a finite difference method is used. The primary equation that is used is:

$$k\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)=\rho C\frac{\partial T}{\partial \tau},$$

where T is the temperature at time τ and k, ρ, and C are the conductivity, density, and specific heat, respectively. The second partial derivatives in Equation 1 can be approximated based on the finite difference technique:

$$\frac{\partial^2 T}{\partial x^2}\approx\frac{1}{\Delta x^2}\left(T_{m+1,n}+T_{m-1,n}-2T_{m,n}\right),$$

$$\frac{\partial^2 T}{\partial y^2} \approx \frac{1}{\Delta y^2}\left(T_{m,n+1} + T_{m,n-1} - 2T_{m,n}\right),$$

and

$$\frac{\partial T}{\partial \tau} \approx \frac{T_{m,n}^{\tau+1} - T_{m,n}^{\tau}}{\Delta \tau}.$$

Figure 1 illustrates the two-dimensional transient conduction problem and nomenclature. Substituting the three finite difference equations above into Equation 1 yields

$$k\left(\frac{T_{m+1,n}^{\tau} + T_{m-1,n}^{\tau} - 2T_{m,n}^{\tau}}{\Delta x^2} + \frac{T_{m,n+1}^{\tau} + T_{m,n-1}^{\tau} - 2T_{m,n}^{\tau}}{\Delta y^2}\right) = \rho C\left(\frac{T_{m,n}^{\tau+1} - T_{m,n}^{\tau}}{\Delta \tau}\right).$$

Rearranging Equation 5 yields

$$T_{m,n}^{\tau+1} = T_{m,n}^{\tau} + \frac{\Delta \tau}{\rho C} k\left(\frac{T_{m+1,n}^{\tau} + T_{m-1,n}^{\tau} - 2T_{m,n}^{\tau}}{\Delta x^2} + \frac{T_{m,n+1}^{\tau} + T_{m,n-1}^{\tau} - 2T_{m,n}^{\tau}}{\Delta y^2}\right).$$

Equation 6 represents the transient numerical solution for the new temperature at node *(m,n)* with constant material properties.

Temperature Dependent Material Properties
Thermal Conductivity

The heat flow across two neighboring cells of different conductivities k_1 and k_2 is accomplished by giving an interface conductivity k_i equal to the harmonic mean of k_1 and k_2:

$$k_i = \frac{2k_1 k_2}{(k_1 + k_2)}.$$

Therefore, the thermal conductivities are defined as follows:

- $k_{m-1,n}^{x}$ and $k_{m+1,n}^{x}$ are the thermal conductivities for heat flow in the *x*-direction between elements *(m-1,n)* and *(m,n)* and *(m+1,n)* and *(m,n)*, respectively.
- $k_{m,n-1}^{y}$ and $k_{m,n+1}^{y}$ are the thermal conductivities for heat flow in the *y*-direction between elements *(m,n-1)* and *(m,n)* and *(m,n+1)* and *(m,n)*, respectively.

From Equation 7, the following equations arise:

$$k_{m-1,n}^{x} = \frac{2k_{m-1,n}^{x} k_{m,n}^{x}}{k_{m-1,n}^{x} + k_{m,n}^{x}},$$

$$k^x_{m+1,n} = \frac{2k^x_{m+1,n} k^x_{m,n}}{k^x_{m+1,n} + k^x_{m,n}},$$

$$k^y_{m,n-1} = \frac{2k^y_{m,n-1} k^y_{m,n}}{k^y_{m,n-1} + k^y_{m,n}},$$

and

$$k^y_{m,n+1} = \frac{2k^y_{m,n+1} k^y_{m,n}}{k^y_{m,n+1} + k^y_{m,n}}.$$

The thermal conductivities for $k^x_{m,n}$ and $k^y_{m,n}$, however, are defined as the arithmetical average of their neighboring cells in order to ensure conservation of heat flow in cell *(m,n)* in the *x*- and *y*-direction, respectively. They are defined as follows:

$$k^x_{m,n} = \frac{k^x_{m-1,n} + k^x_{m+1,n}}{2}$$

and

$$k^y_{m,n} = \frac{k^y_{m,n-1} + k^y_{m,n}}{2}.$$

Heat Capacity

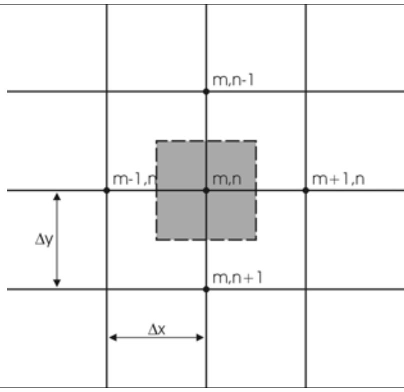
Density

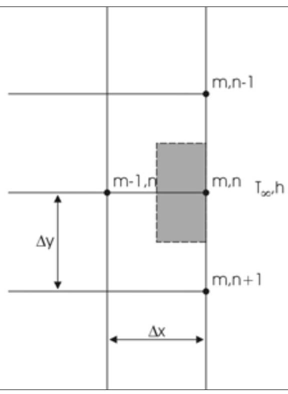
Nodal Equations

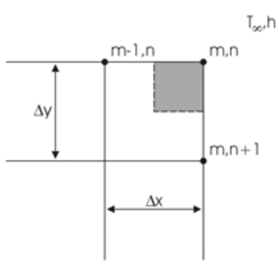
Letting the thermal conductivity vary with temperature yields the transient numerical solution for the temperature of an internal node *(m,n)*:

$$T^{\tau+1}_{m,n} = T^{\tau}_{m,n} + \frac{\Delta\tau}{\rho C}\left(\frac{k^x_{m+1,n}T^{\tau}_{m+1,n} + k^x_{m-1,n}T^{\tau}_{m-1,n} - 2k^x_{m,n}T^{\tau}_{m,n}}{\Delta x^2} + \frac{k^y_{m,n+1}T^{\tau}_{m,n+1} + k^y_{m,n-1}T^{\tau}_{m,n-1} - 2k^y_{m,n}T^{\tau}_{m,n}}{\Delta y^2}\right)$$

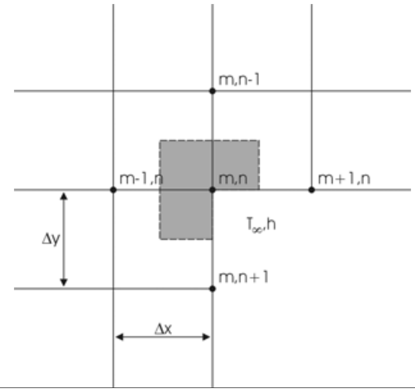
.

<table>
<tr><td colspan="2">Case 1: Interior Node</td></tr>
<tr><td>
</td><td>Figure of energy balance</td></tr>
<tr><td colspan="2">Insert energy balance here.

$$T_{m,n}^{\tau+1} = T_{m,n}^{\tau} + \frac{\Delta\tau}{\rho C}\left(\frac{k_{m+1,n}^{x}T_{m+1,n}^{\tau} + k_{m-1,n}^{x}T_{m-1,n}^{\tau} - 2k_{m,n}^{x}T_{m,n}^{\tau}}{\Delta x^2} + \frac{k_{m,n+1}^{y}T_{m,n+1}^{\tau} + k_{m,n-1}^{y}T_{m,n-1}^{\tau} - 2k_{m,n}^{y}T}{\Delta y^2}\right.$$</td></tr>
<tr><td colspan="2">Case 2: Plane with Convection</td></tr>
<tr><td>
</td><td>

$$T_{m,n} = \frac{T_{m-1,n} + \frac{T_{m,n+1} + T_{m,n-1}}{2} + BiT_{\infty}}{2 + Bi}¿$$</td></tr>
<tr><td colspan="2">Case 3: Exterior Corner with Convection</td></tr>
<tr><td>
</td><td>

$$T_{m,n} = \frac{\frac{(T_{m-1,n} + T_{m.n-1})}{2} + BiT_{\infty}}{1 + Bi}$$</td></tr>
<tr><td colspan="2">Case 4: Interior Corner with Convection</td></tr>
</table>



$$T_{m,n}=\frac{BiT_{\infty}+T_{m,n+1}T_{m-1,n}+\frac{T_{m+1,n}+T_{m,n-1}}{2}}{3+Bi}$$

Experimental Results:

As of yet Conway's Game of Life has been implemented and it has been validated. After that a finite difference model of heat transfer has been implemented. The finite difference model works remarkably fast considering that running it on a laptop with a 30X30 grid say 5000000 takes less then six hours. The heat transfer has been compaired with other studies done in the field and the results match.

Conclusion:

Designing a heat transfer model on a Cellular Automata framework is benificial since adding convection already seems a lot easier and streamlined then using finite difference to handle convection. A lattice Boltzman model still hasn't been implemented properly however.

Future Work:

The biggest thing that I can see that needs to be done is implement Convection using Lattice Boltzman. I would also eventually like for the simulation to be able to handle radiation heating since it is so very important if you are trying to model anything outside in the day time.