Project Proposal

Cody Michael Morrin (UNI:cmm2360) cody.morrin@columbia.edu
Yung-Ching Lin (UNI:yl4337) yl4337@columbia.edu

Motivation

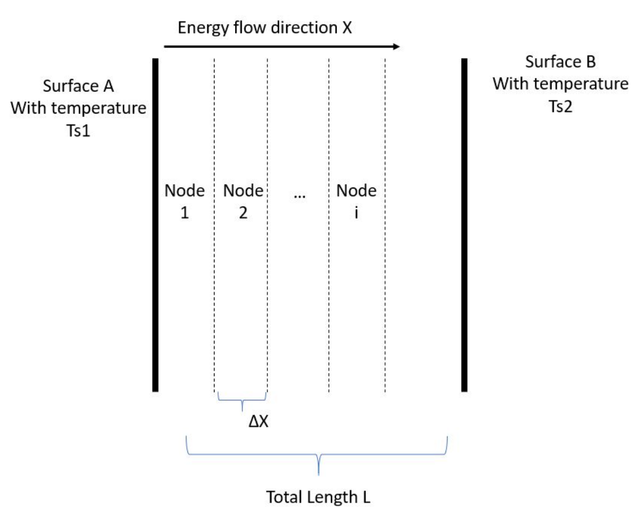
- Relaxation Iterative Method

Relaxation methods are methods of solving partial differential equations that involve splitting the matrix/vector that arises from finite differencing then iterating until a solution is found. These equations describe boundary-value problems, in which the solution-function's values are specified on the boundary of a domain. The main reason for choosing this as the final project topic is because it is fairly easy to increase the complexity by changing the edge conditions or increasing the pieces in space dimensions. In this project, we apply relaxation method on differential equations (diffusion). We plan to see the different results on different complexity of the equations with different parallelization methods.

Project Topic - Discretized 1D heat equation

- Problem description

Assuming we have two different surfaces temperatures (Ts1, Ts2). We want to describe how temperature changes according to the direction of X where X is starting from surface A to surface B using 1D heat equation. The length between two surfaces is L. We break L into several discrete sections with length Δx. We called nodes in each Δx unit. We can break L up into as many nodes as we want. See the following picture as reference.



We define u(x,t) as the temperature of some substance at node x and time t, where we can increase the total time steps as much as we want. The original diffusion equation will be like :

$$\frac{\partial}{\partial t}u(x_i, t_n) = \alpha\frac{\partial^2}{\partial x^2}u(x_i, t_n),$$

Next, we use Forward Euler Scheme to replace the derivatives with finite difference approximations. Forward Euler Scheme increments the solution through an interval while using derivative information from only the beginning of the interval. We got

$$(1)\quad \mathbf{u(x_i, t_{n+1}) = u(x_i, t_n) + F^* (u(x_{i+1}, t_n) - 2^* u(x_i, t_n) + u(x_{i-1}, t_n))}$$

where F is the parameter in discrete diffusion equation. The value of F depends on the thermal diffusivity parameter, Δx, and Δt.

- Algorithm steps

for timestep from **0** to total time **T**
  for nodes from **0** to total nodes **N**
    apply **equation (1)** on each node
    Set the boundary conditions (node 0 and node (N-1))

Strategies

- Maybe store values (in the spatial dimension) in an array instead of a list, faster to access, and we never need to extend the domain. In the time dimension, we want to be able to iterate forever, so I think it should be stored in a list, or could just not store it and iterate with a recursive function (I think a list is better though, because it makes the comparison of the solution at different time values easier).
- Possible parallelizations: the calculation of the value at each x point depends only on the values of the previous iteration, so they could all be calculated at the same time.

Reference

1. The 1D Heat Diffusion
   http://hplgit.github.io/num-methods-for-PDEs/doc/pub/diffu/sphinx/._main_diffu001.html
2. Relaxation Iterative Method
   https://en.wikipedia.org/wiki/Relaxation_(iterative_method)