**Introduction**

Heat Equation, applications in ice sheet, the larger picture

**Problem Physics**

Heat Equation, introduce the equation, the physics of it, possible short derivation of equation??

**Algorithms**

Introduce and derive the algorithms and how they will be solved

* Simple Explicit
* Simple Implicit
* Crank-Nicolson Implicit

The Crank-Nicolson scheme was developed by Crank and Nicolson in 1947 as an implicit scheme to solve the heat equation. They wrote:

Again, is the central difference operator. Expanding this out and setting , we see

Our unknowns are all values at the next timestep. Just like the implicit method, the solution at the next timestep depends not only on the nodes already known at time , but at the next timestep . We can rearrange to get unknowns on one side and knowns on the other:

For each timestep, this is solved for each of the interior points (for nodes, going from to , there are interior points going from to ) – this gives us a system of equations. We write out the first few iterations to get a sense of the system that will need to be solved every time step.

Now that we have a sense of the pattern for these linear equations, we can rewrite this in matrix form. First, we write out the left-hand side (the unknowns) as :

Now the right-hand side (which is pretty similar) as , where :

We can further reduce this right-hand side to just one matrix, realizing that we are essentially multiplying by :

Things to add: do we solve this using LU decomposition (thomas algorithm?), perhaps we can solve this different ways. We can compare this matrix with the implicit scheme matrix. Amplification factors. Mention it is unconditionally stable and why (because implicit, I think) and how makes use of the trapezoidal method (perhaps get the original paper and have some kind of verification here).

Changing boundary conditions to reflect the changing surface temperature

#different ways to explore

#-represent space derivative (higher order, 4th order FD, look at scaling)

#-loglogplots see if error scales for h

#spectrally

#time stability

#runtime

#

#

#

**Code??**

**Results**

**Roadblocks**

When you have changing boundary conditions, it is tough to verify the different methods against analytical solutions – but we can do it assuming static boundary conditions

**Conclusion**

**Appendix?? Perhaps we can fit all necessary figures in the Results section though**