Optimizing Sleeping Environment

Jerome Andaya
Jeffrey Lin
Khai Phan

Problem Statement

Our Project was to effectively model a square room to visualize temperature changes in regards to how much the outside world could affect it.

Our goal was to find the optimal percentages to open two windows in a room in order for the room to converge to a specific temperature. A potential upgrade was to find the optimal percentages that would converge fastest.

Thermal Conductive Model

Method for Thermal Conduction: Successive Over Relaxation

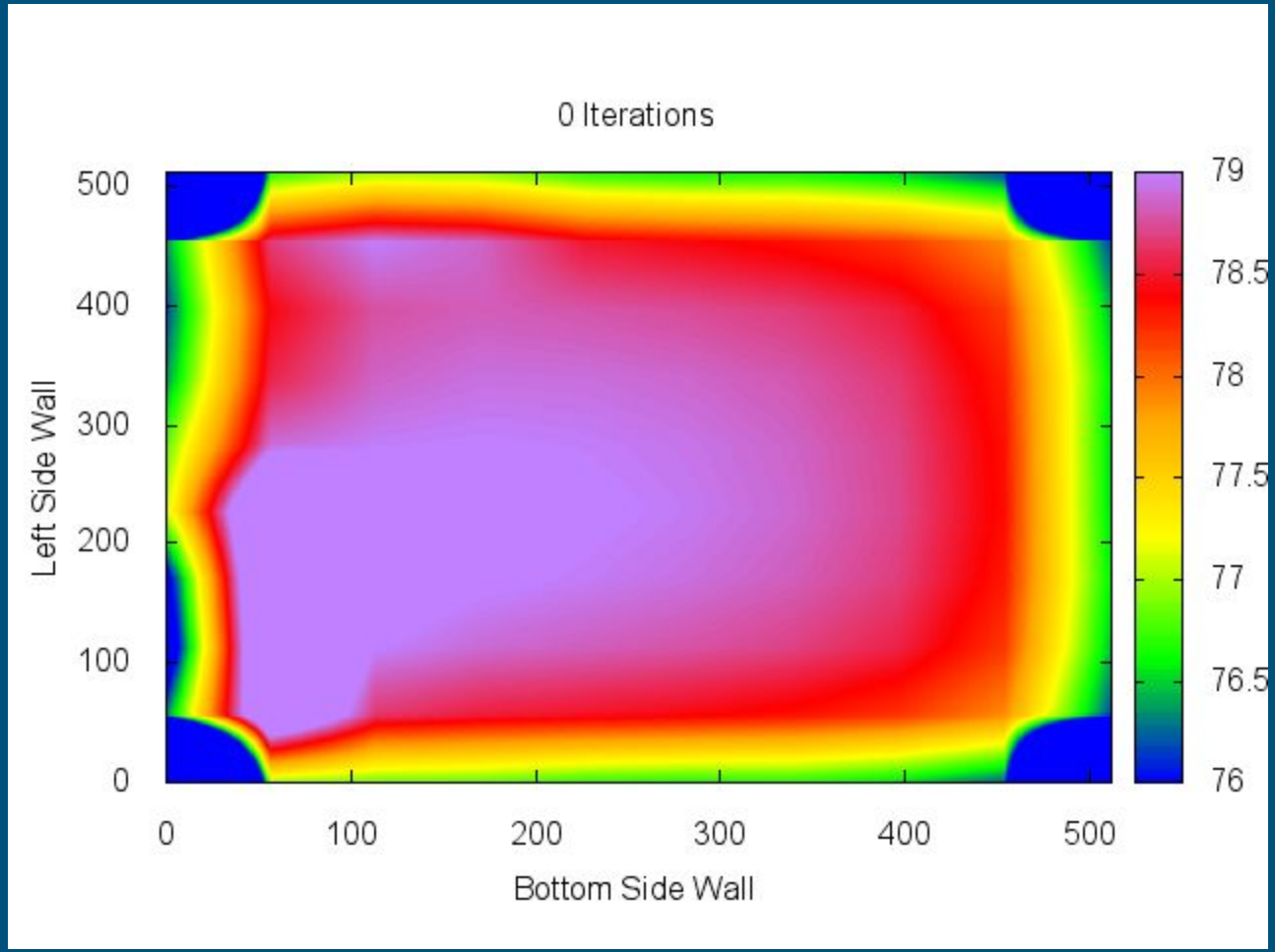
Method we've been using all semester

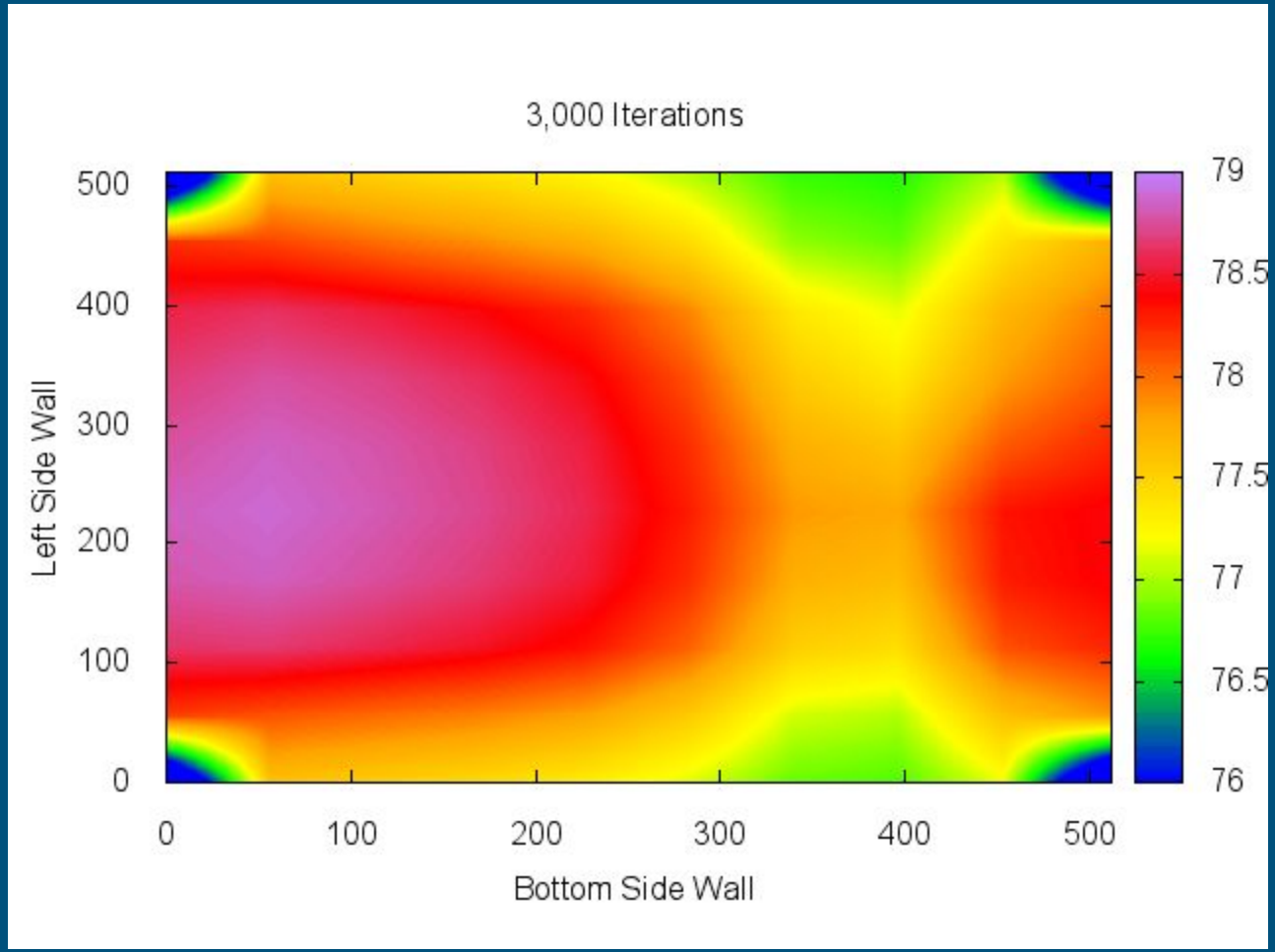
Using CUDA and nVidia GPU to perform block-wise SOR on a 512 x 512 matrix.

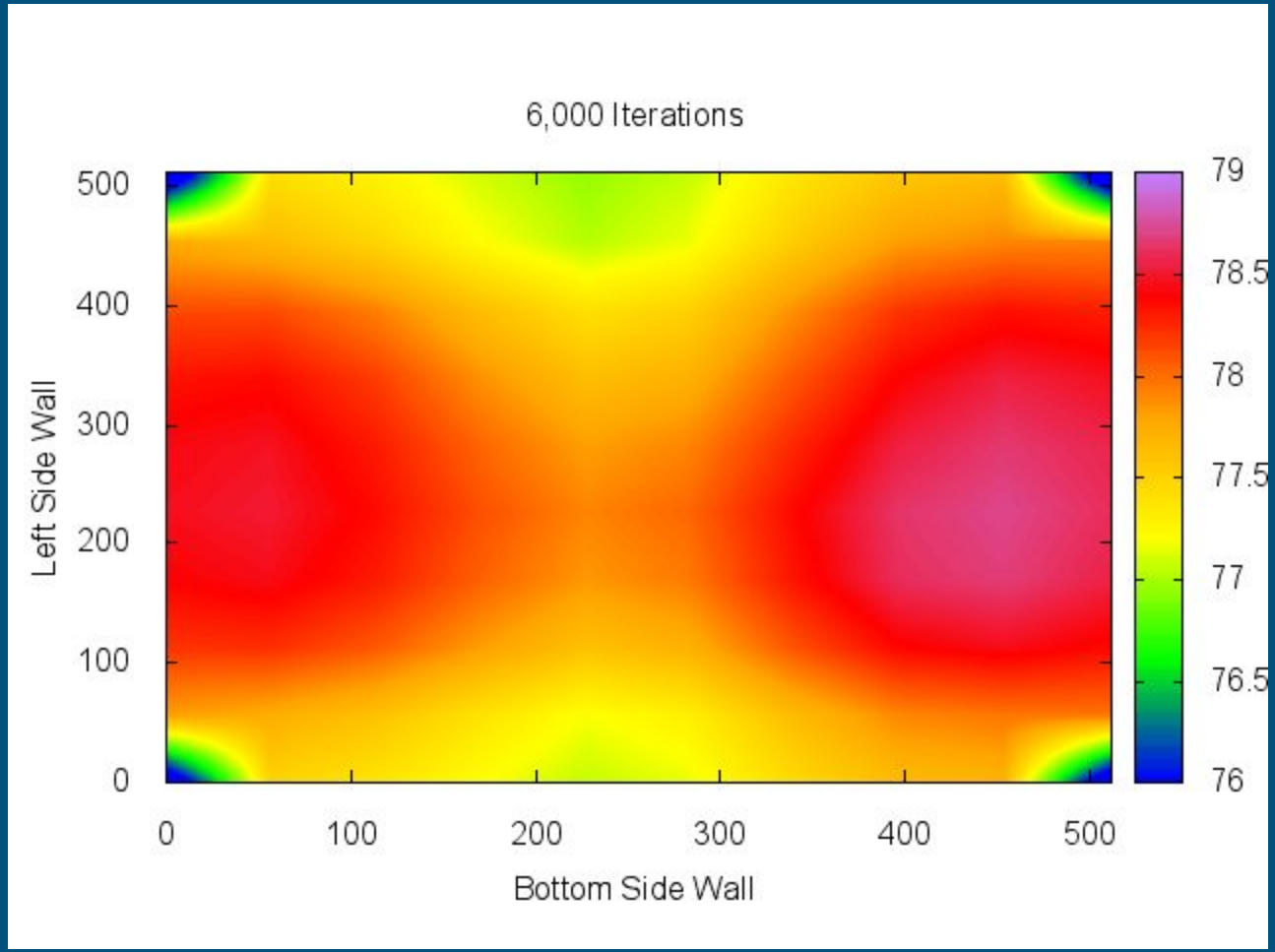
Jacobi style with 2 matrices. d_y > d_result > d_y

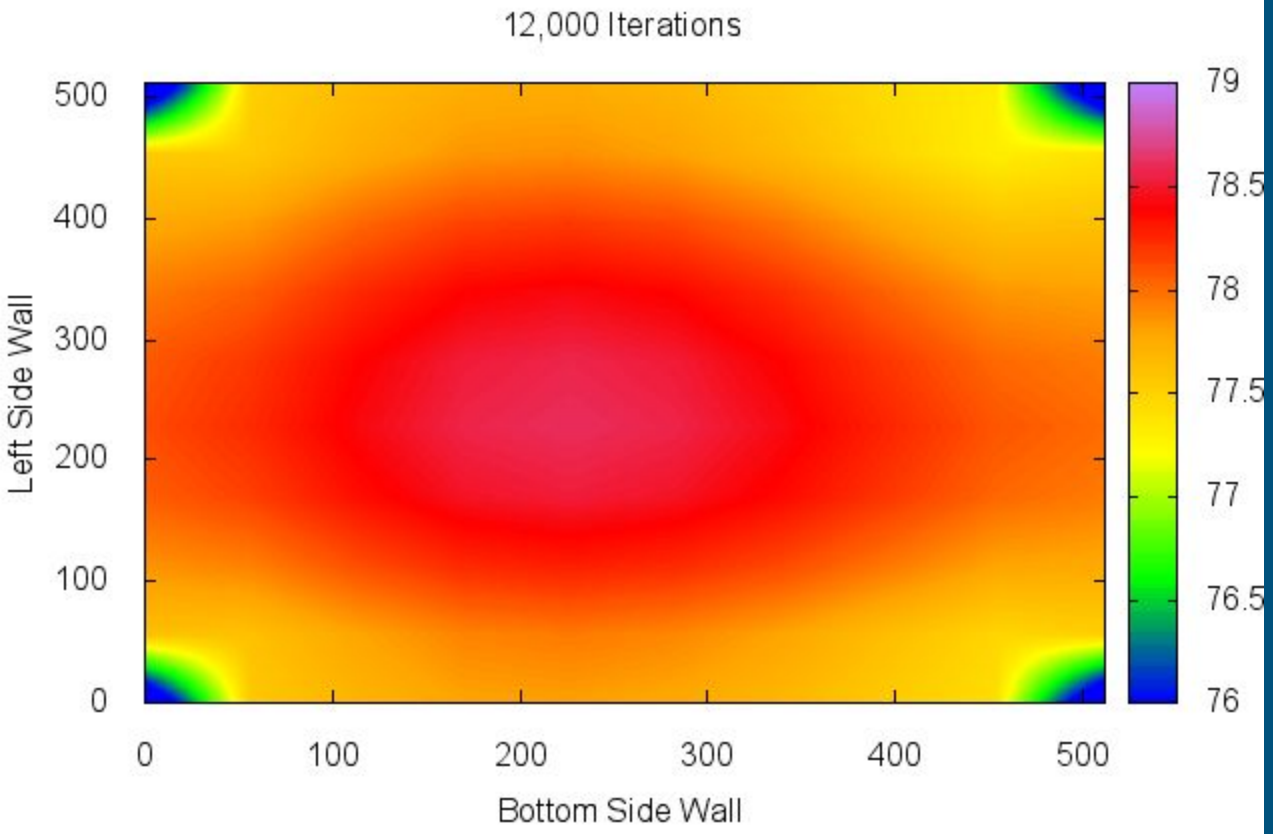
Successive Over Relaxation Logic

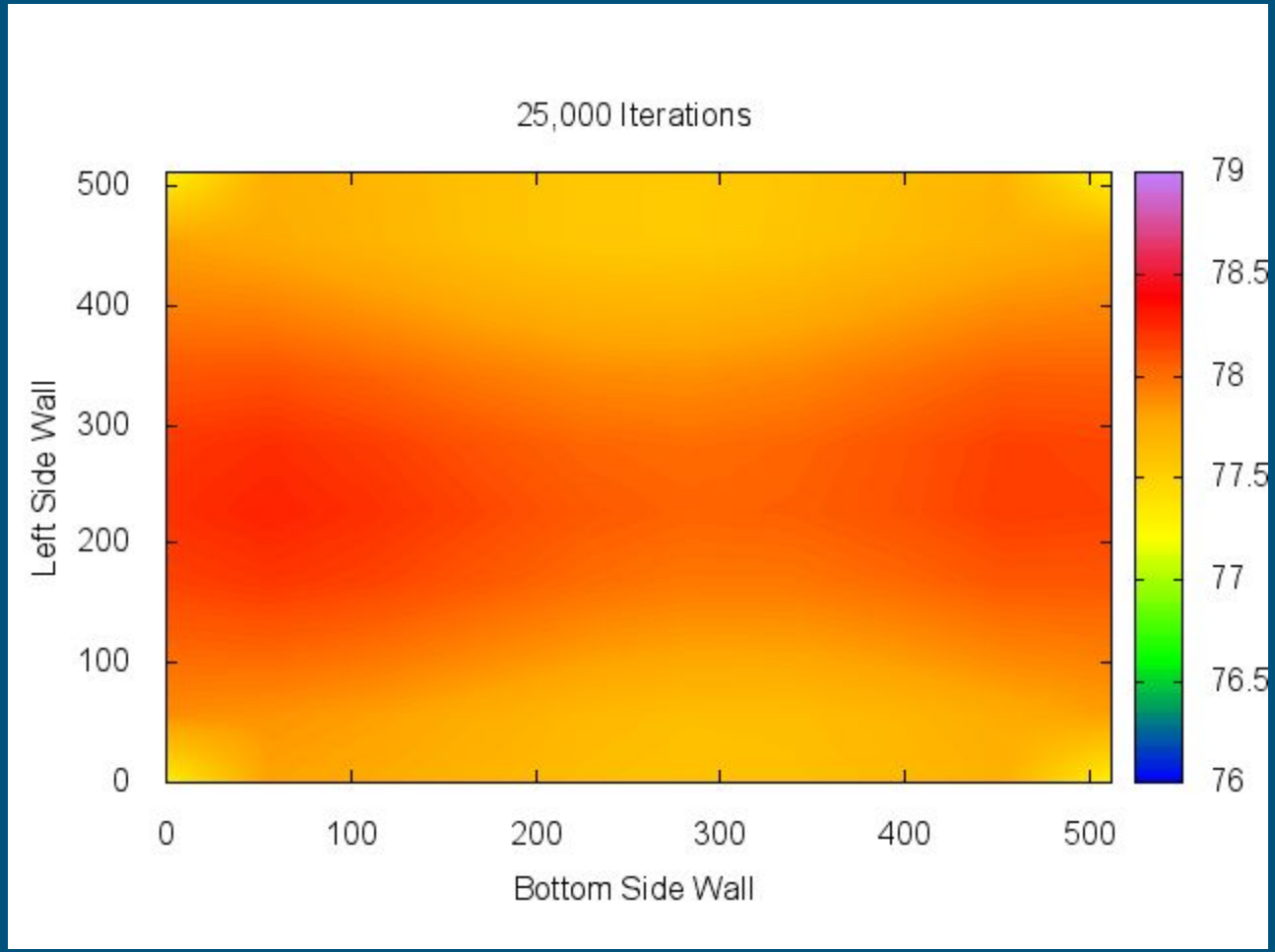
```
__syncthreads();
if (ignoreFlag[tid%256]) {
    result[i] = RHO * y[i] + (1.0-RHO) * (
            w_drift *(y[((((i+arrLen)%arrSize))+arrSize)%arrSize]) +     //top
            mw_drift*((y[(((i+1)%arrSize)+arrSize)%arrSize]) +           //right
                    (y[(((i-1)%arrSize)+arrSize)%arrSize]) +             //left
                    (y[((((i-arrLen)%arrSize))+arrSize)%arrSize]))       //bottom
    );
}
```

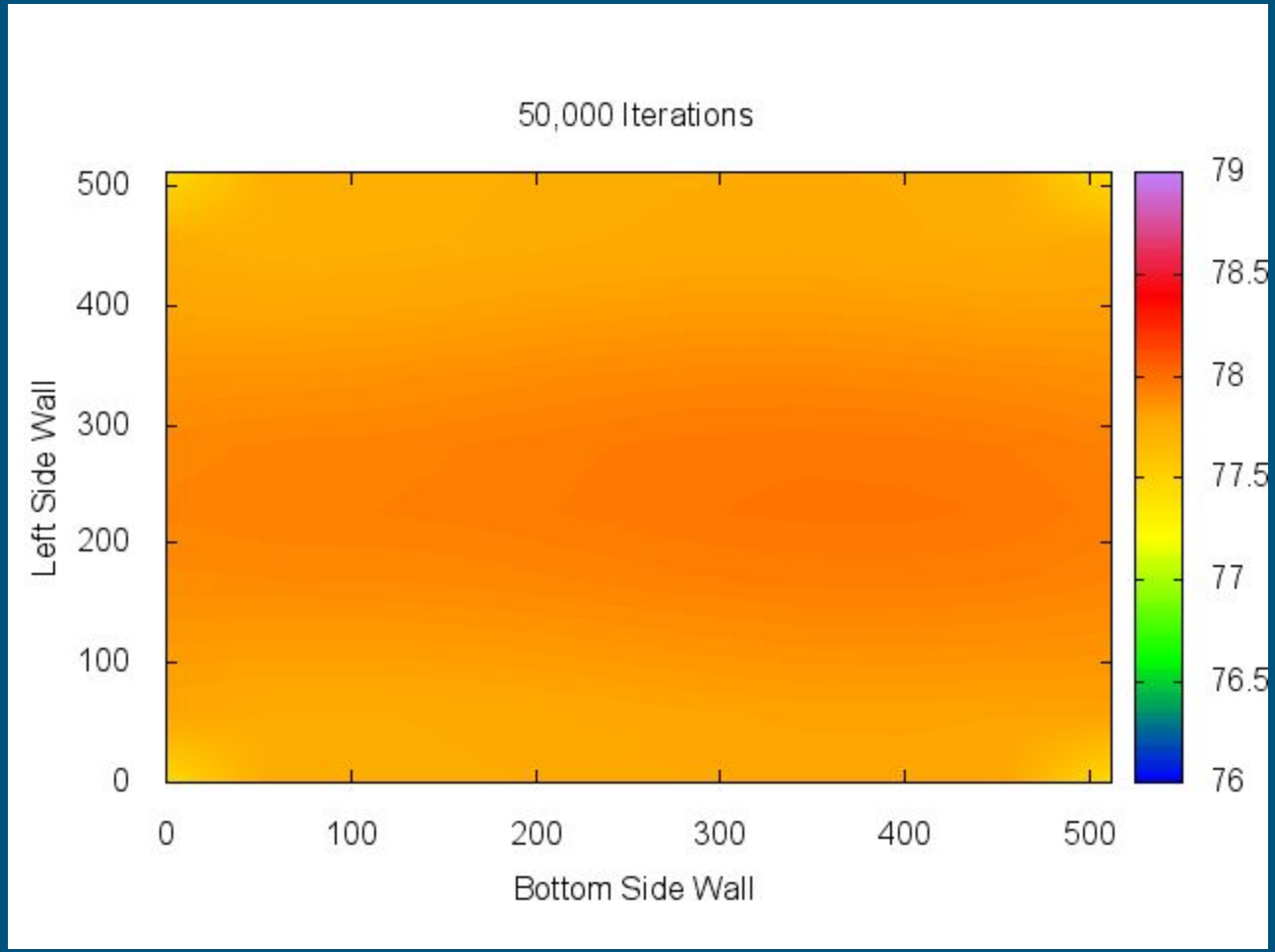

0 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
79
78.5
78
77.5
77
76.5
76

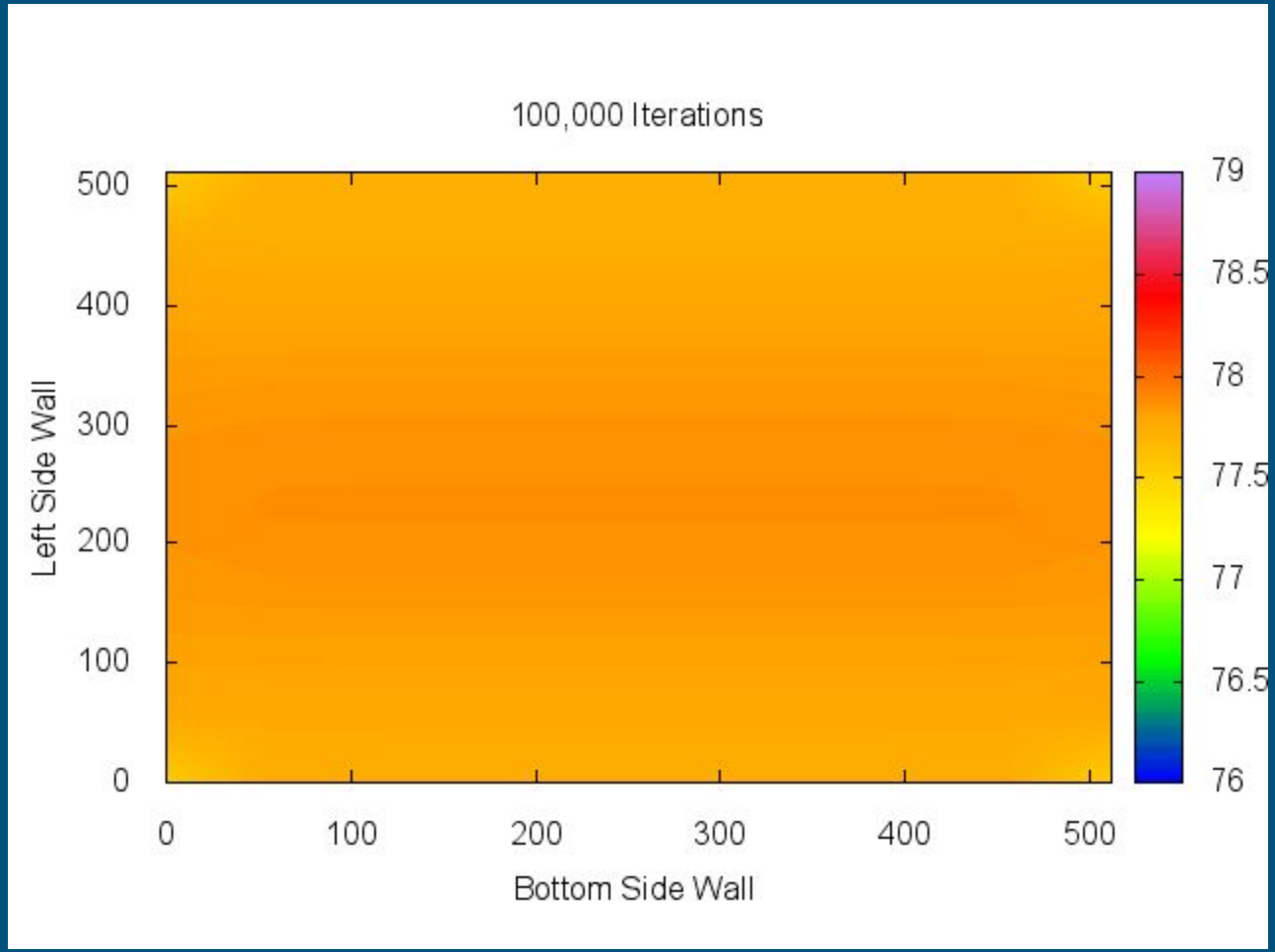

3,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
76
76.5
77
77.5
78
78.5
79


6,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
76
76.5
77
77.5
78
78.5
79


12,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
76
76.5
77
77.5
78
78.5
79


25,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
79
78.5
78
77.5
77
76.5
76


50,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
76
76.5
77
77.5
78
78.5
79


100,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
79
78.5
78
77.5
77
76.5
76

Problems with Naive Model

Our naive approach of not shifting the heat didn’t accurately model how heat would be moving across time. We believed by increasing the weight of the top source, we would be able to simulate wind by having the cold windows affect the area below them significantly.

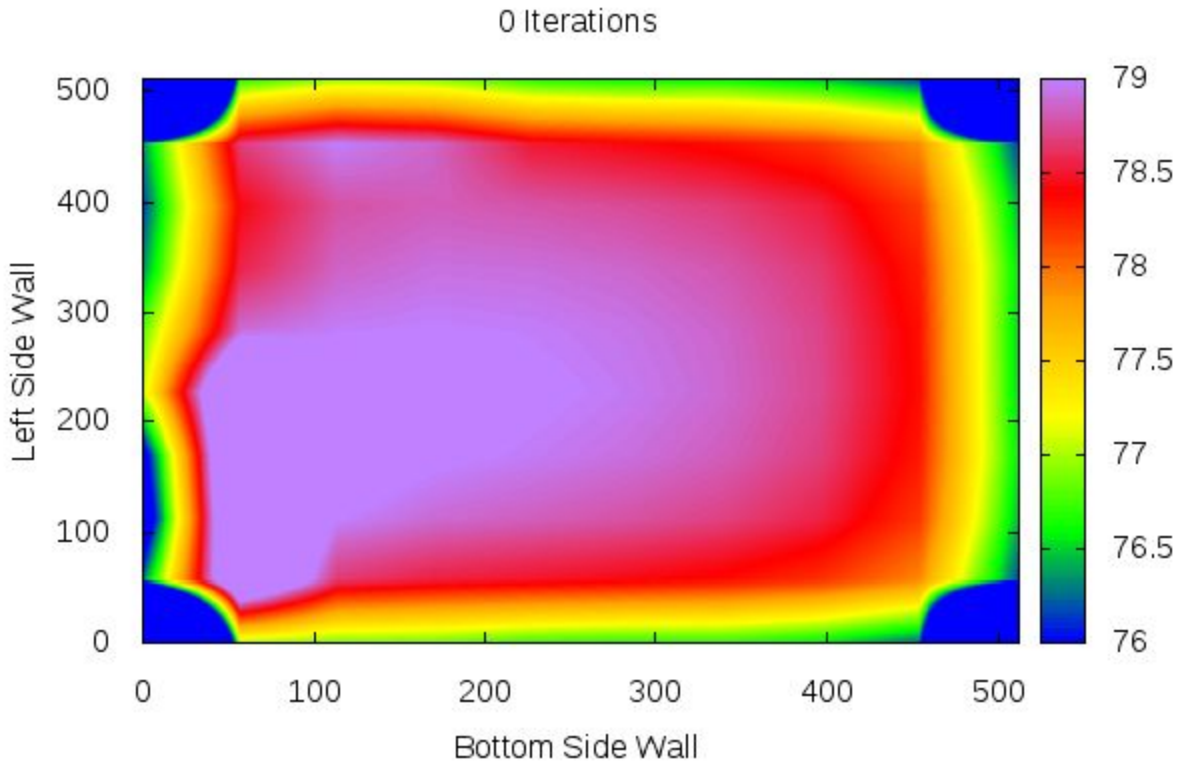
This hypothesis proved to be false as this actually changed heat conductivity. By changing the weight, we were only changing the rate of which heat transfers. In reality, convection is the movement of air; thus, we needed a model to move air throughout the room.

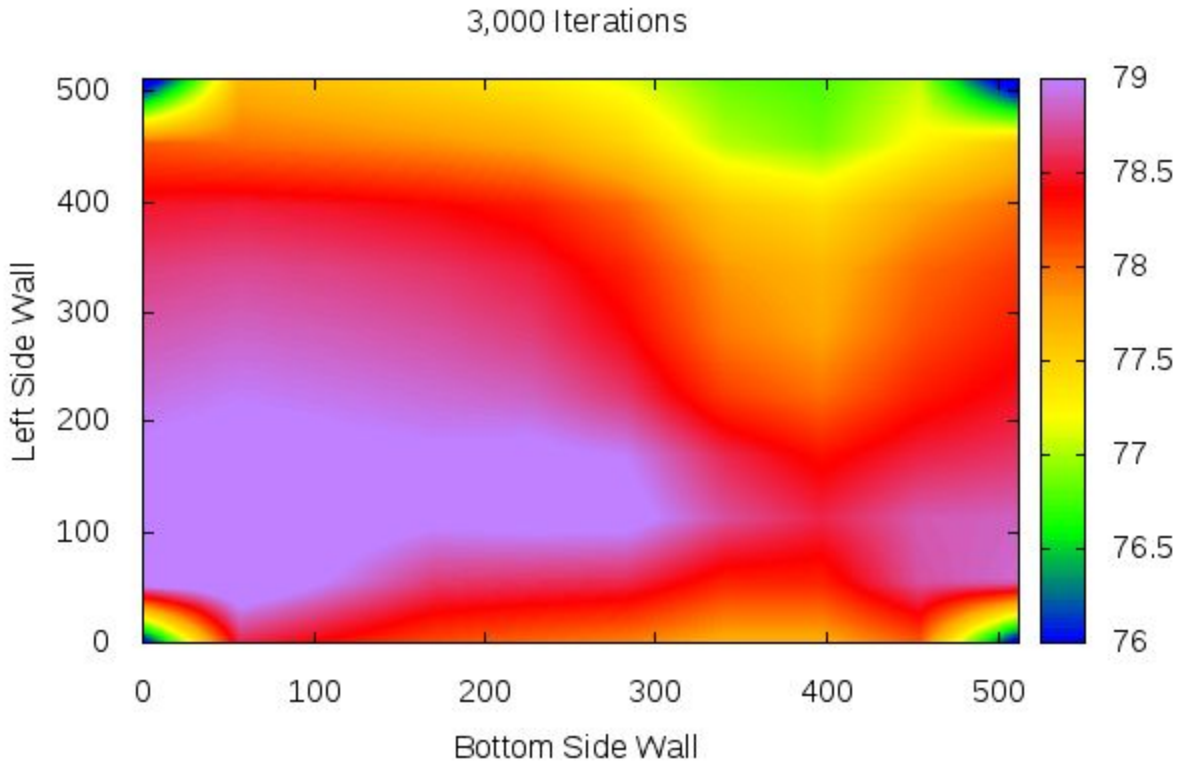
Improved Method - Shift Heat

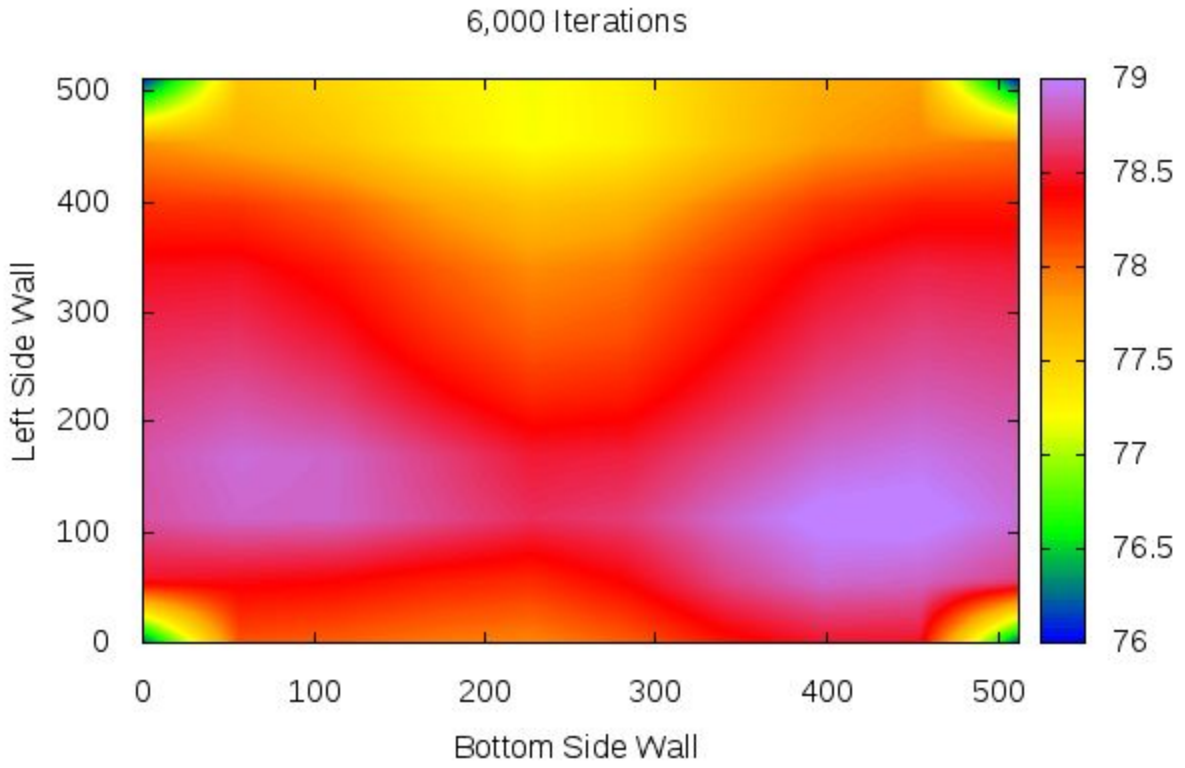
To solve this problem we were recommended shift the air and the heat it contains. We want to have the cold air from the windows come in and cool the room around it.

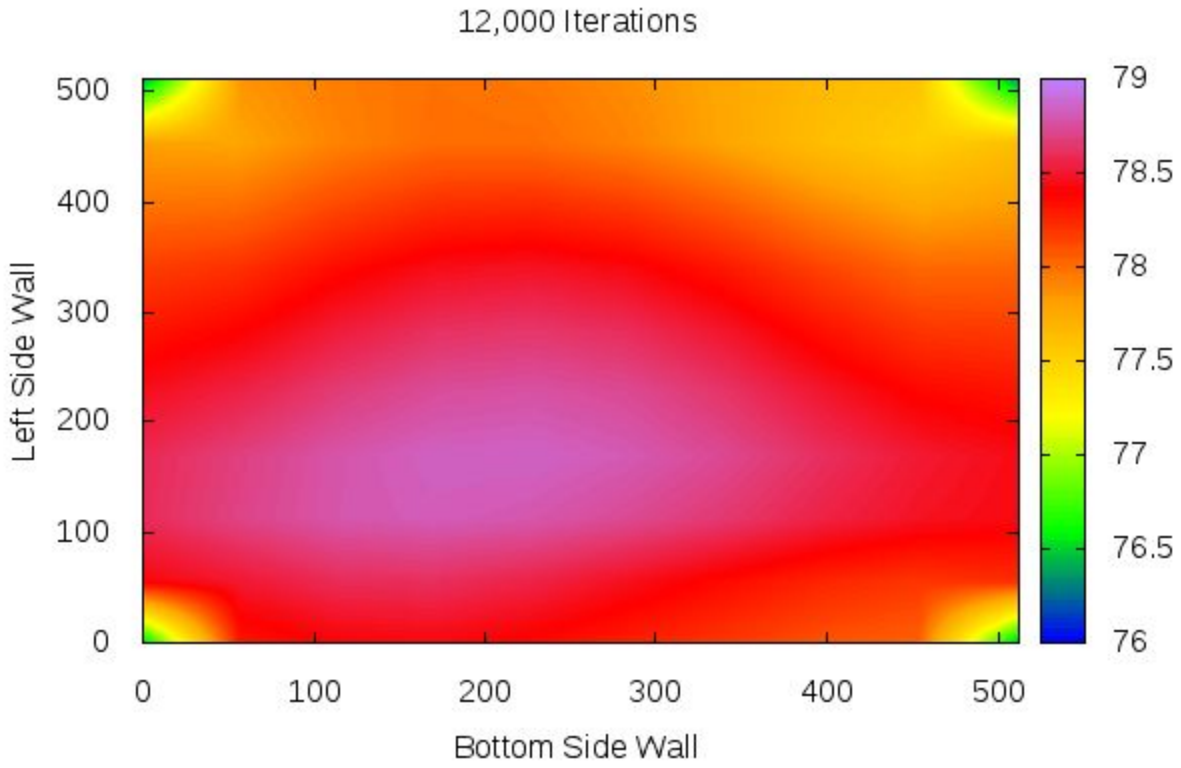
For CUDA blocks having blockIdx.x in the ranges [4,12) and [20,28), we shifted their values one block every 1000 iterations of SOR.

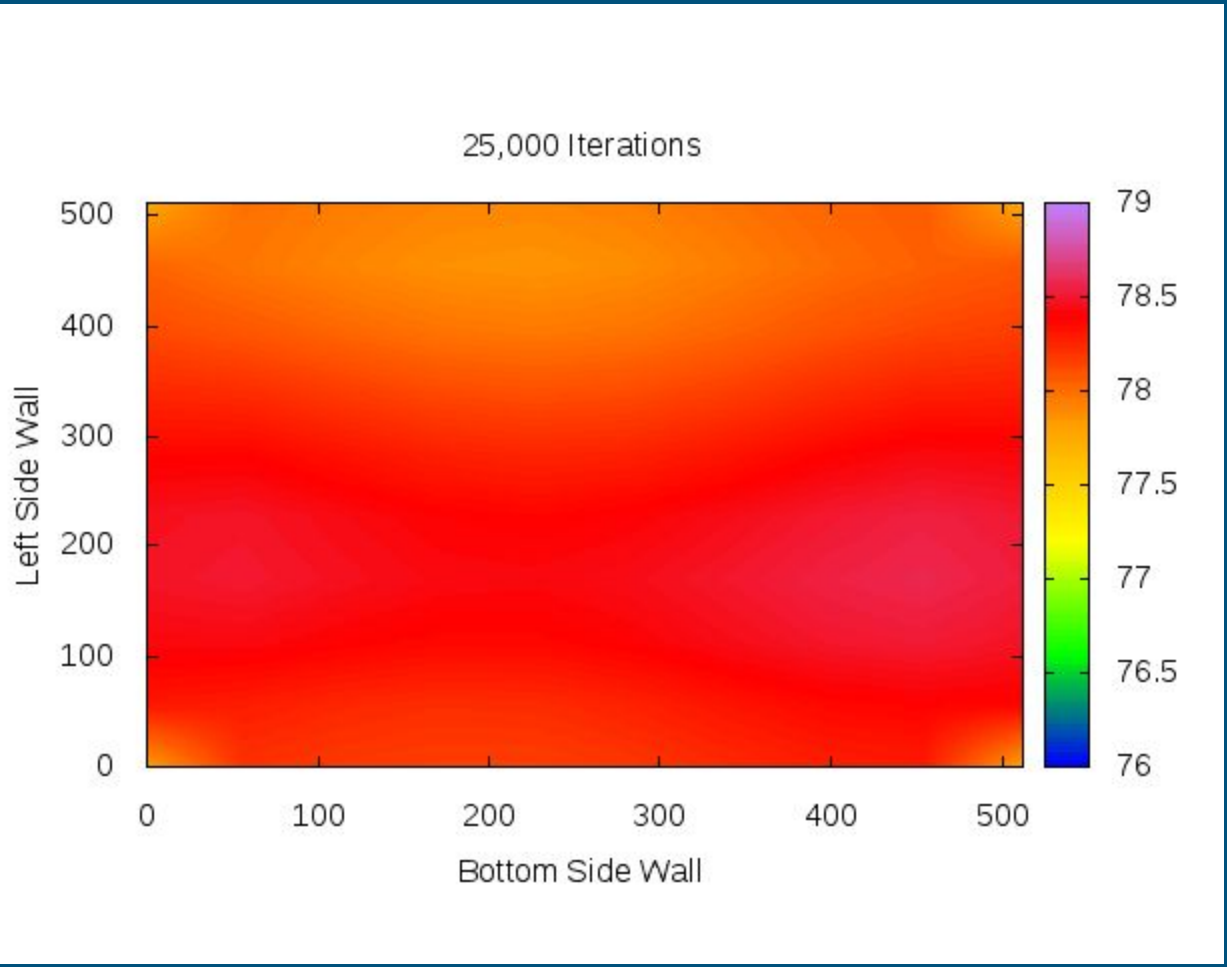
All other blocks are held at their positions.

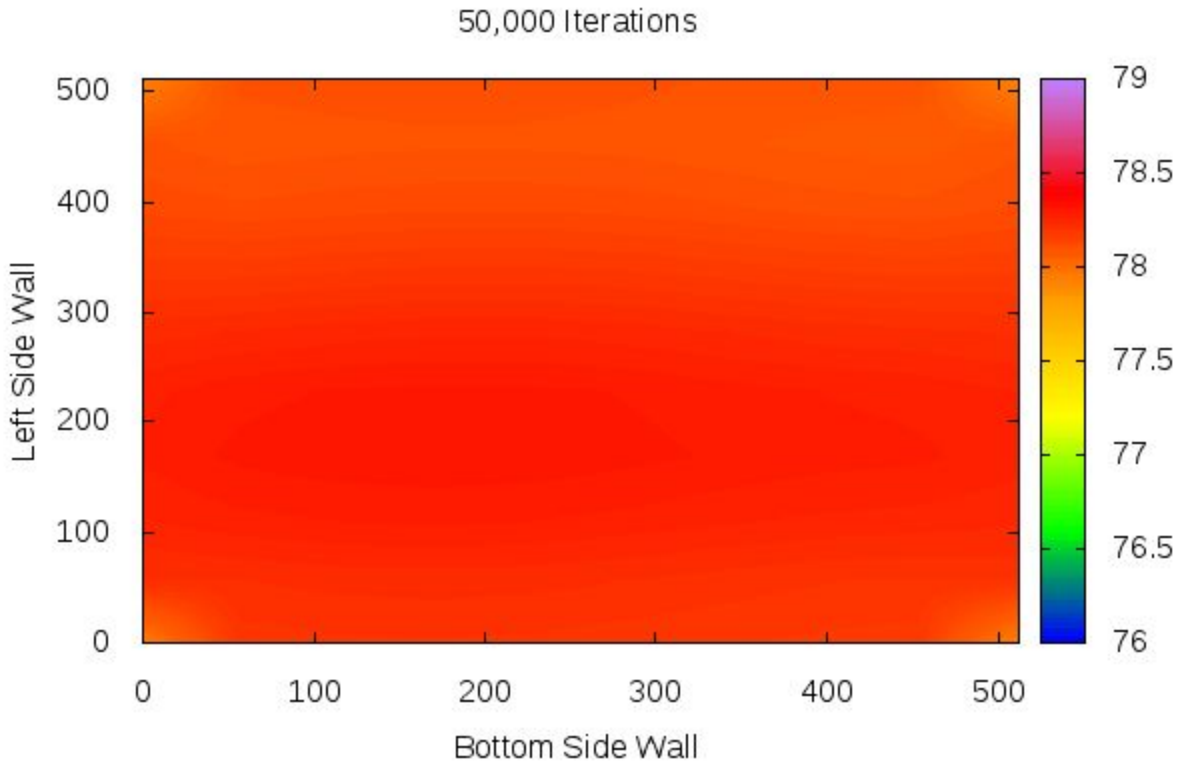

0 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
79
78.5
78
77.5
77
76.5
76

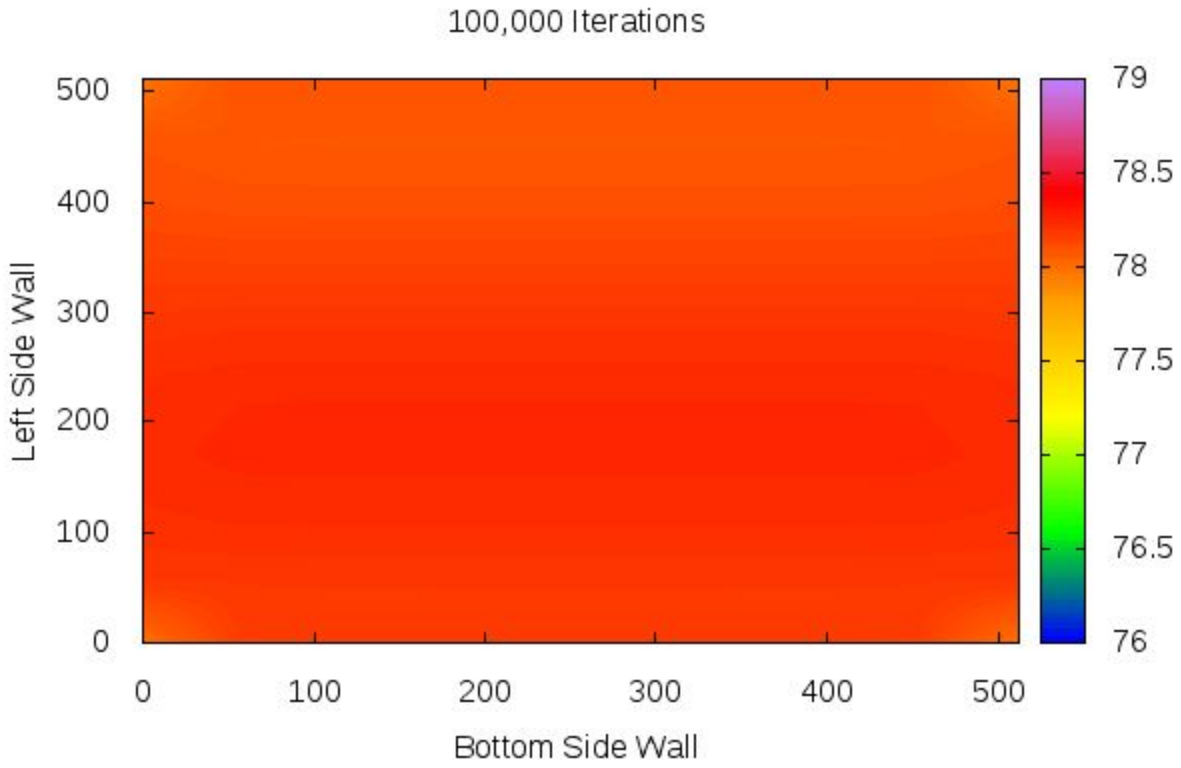

3,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
76
76.5
77
77.5
78
78.5
79


6,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
76
76.5
77
77.5
78
78.5
79


12,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
76
76.5
77
77.5
78
78.5
79


25,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
76
76.5
77
77.5
78
78.5
79


50,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
76
76.5
77
77.5
78
78.5
79


100,000 Iterations
Left Side Wall
Bottom Side Wall
0
100
200
300
400
500
79
78.5
78
77.5
77
76.5
76

For the Future...

- Fix model to remove temperature changing windows
- Add velocity of convection
- Add dictionary search optimization for function calls

Conclusion

- Things didn't work out as planned, but we learned!
- Careful when trying to model drift, could end up actually being conductivity!
- The simpler the better.

Thank you!