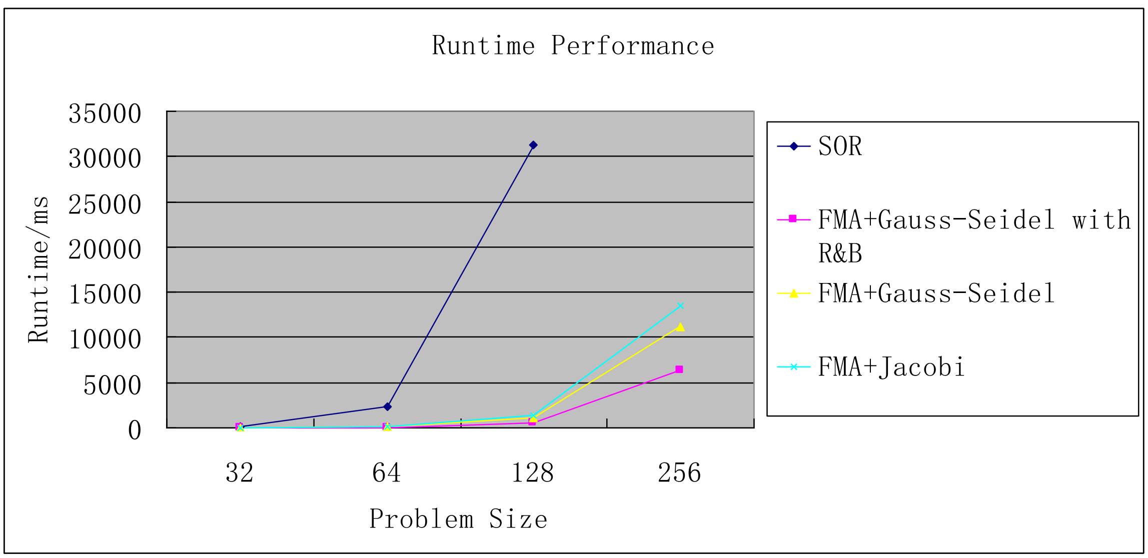
1. Algorithm Implementation

I first change the code from two dimensions to three dimensions, and then add the time step factor, i.e. coefficient C is added to the heat equation.
In relax.c, three ways are written to reduce high-wave number error component, Gauss-Seidel with black and red ordering, Gauss-Seidel, and Jacobi.
The result after one time step is written into the output file named soln.txt. Use Matlab to read the matrix in, and then draw the plot to demonstrate the correctness of the algorithm.

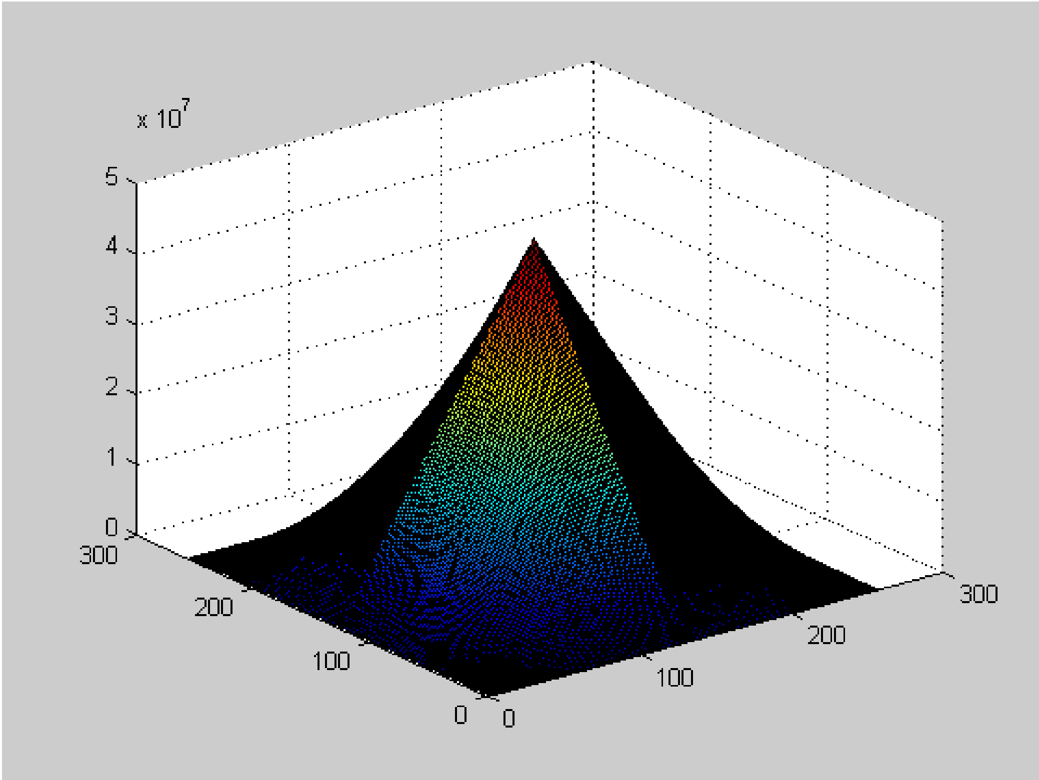
2. Runtime Performance Comparison

SOR is clearly much slower than FMA. And among the three different choices of relaxation schemes, Gauss-Seidel with R&B is the fastest, about one time faster than Jacobi.

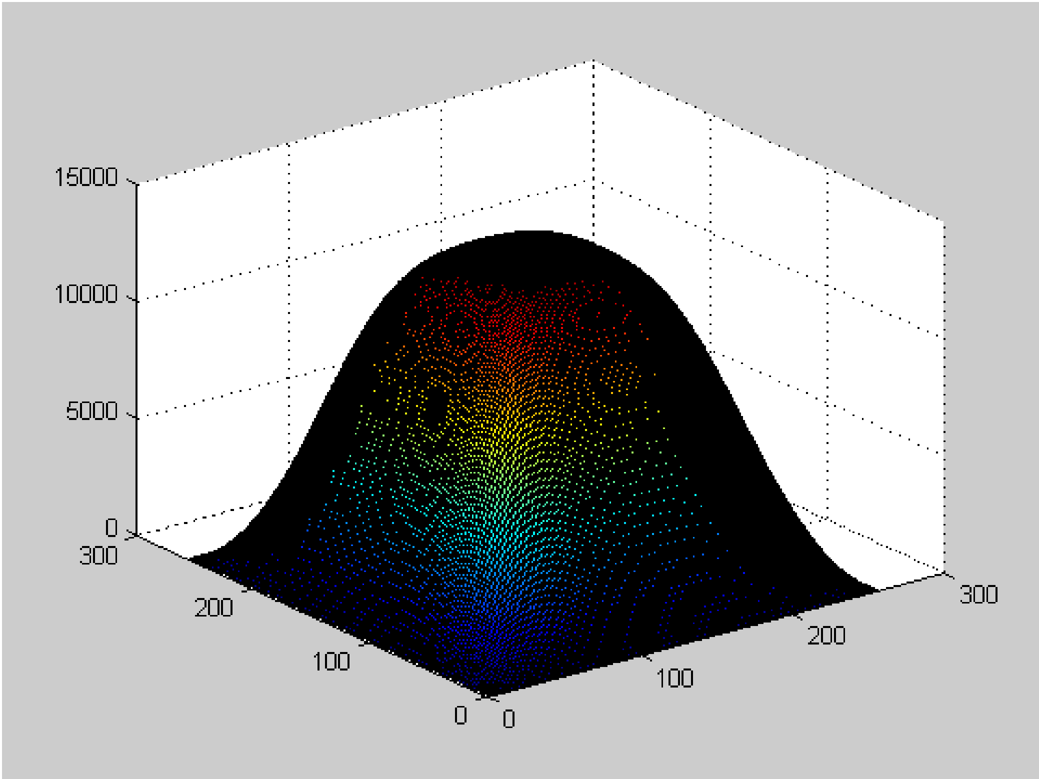


3. Relaxation Scheme Comparison

Besides the runtime performance difference, the appearance of the plots of the heat diffusion process is also different among these three. Gauss-Seidel and Jacobi are smoother than Gauss-Seidel with R&B. The plots are as follows in the next page.

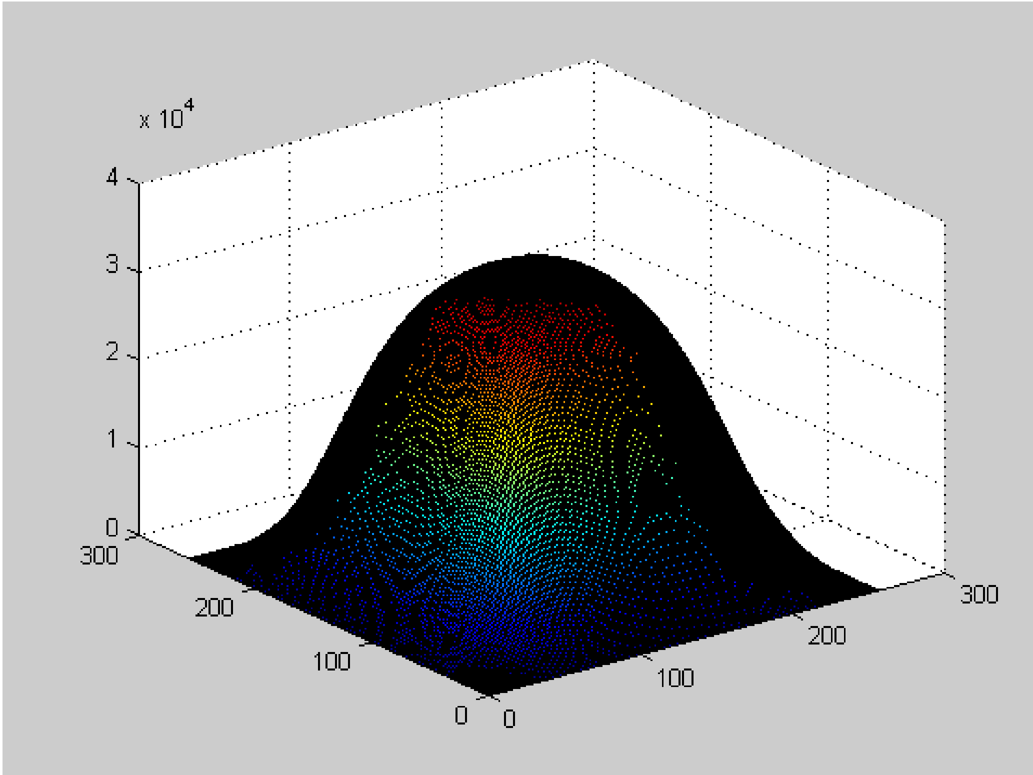


Gauss-Seidel with Red black ordering



Gauss-Seidel

darenc@uchicago.edu



Jacobi

4. Crank-Nicholson

The whole program is pretty much the same as when we are using Backward Euler except for several slight changes in the slvsml.c, relax.c and resid.c. Extra code is provided in another folder named Crank-Nicholson.

MORE INFORMATION ABOUT RUNNING THE PROGRAM COULD BE FOUND IN README.TXT.

Thank you!