Final Project

Numerical Solution to The Diffusion Equation

Cameron Olsen

1358873

Scientific Computing for Mechanical Engineers

Dr. Prosperetti and Dr. Amritkar

MECE 5397

Spring 2019

University of Houston

Cullen College of Engineering

Abstract

The diffusion equation was solved numerically in Matlab using the explicit and implicit method. The equation was discretized over a rectangular area with given boundary conditions and no initial conditions. The code was written modularly in order to accommodate adjustments should the mathematical statement change. The results of these two methods were compared using timing, grid convergence theory, and error analysis. Finally, visual representations of the data were created.

Mathematical Statement of Problem

Solve:

Over:

Given:

Discretized Version of the Equations

Explicit:

Implicit:

Description of the Numerical Method

The explicit method can be rearranged to solve for , shown below.

Three of the boundaries are fixed, at Ax, Ay, and By, therefore there is no need to include these in our formulation. In fact, it would not only be wasteful to do so but impossible as we lack the needed information of some of their neighboring points. The fourth boundary is a Neumann boundary condition. The Neumann condition can be understood to be having a rate of change of zero with respect to X on the Bx boundary. The discretization to third order accuracy for this condition can be seen below.

or

This leads to an alternate version of the discretized version of at X = Bx.

These two versions can be reconciled with a branching if statement as seen in the pseudo code below.

%Initializing Time Step

for k = 0:ht:Bt

v=v+1

%Space Step X

for j = 2:Nx

%Space Step Y

for i = 2:Ny-1

if j == Nx

UnE(i,j,v+1) = (UnE(i+1,j,v)-2\*UnE(i,j,v)+UnE(i-1,j,v))\*((ht\*D)/(hx^2)) + (-2\*UnE(i,j,v)+2\*UnE(i,j-1,v))\*((ht\*D)/(hy^2))+ UnE(i,j,v);

else

UnE(i,j,v+1) = (UnE(i+1,j,v)-2\*UnE(i,j,v)+UnE(i-1,j,v))\*((ht\*D)/(hx^2)) + (UnE(i,j+1,v)-2\*UnE(i,j,v)+UnE(i,j-1,v))\*((ht\*D)/(hy^2))+ UnE(i,j,v);

end

end

end

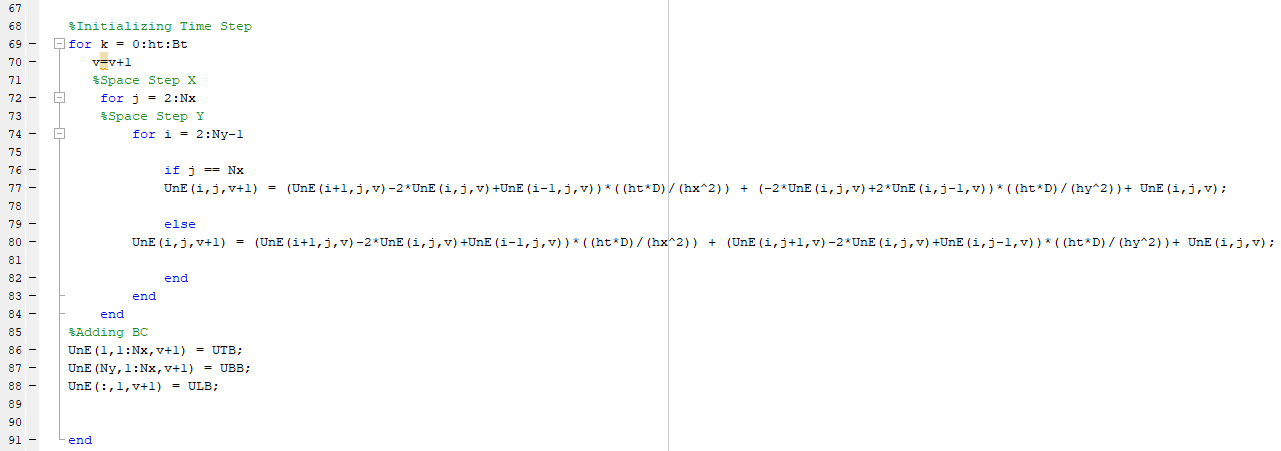
%Adding BC

UnE(1,1:Nx,v+1) = UTB;

UnE(Ny,1:Nx,v+1) = UBB;

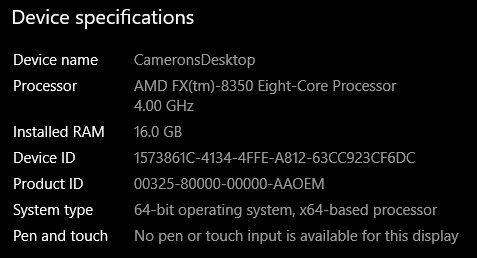
UnE(:,1,v+1) = ULB;

end



It is also worth noting that this method is conditionally stable, therefore the magnitude of is dependent on , , and the Diffusivity constant D as seen below.

Technical Specifications of the Computer Used



Results

– Specifications of parameters used in simulations

– Evaluate the effect of number of points used for discretization

– Perform grid convergence study

– Evaluate the effect of diffusive CFL\*

– Comparison of results with expected theoretical behavior

– Verify the order of spatial accuracy of discretization