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. The full repository of Matlab code can be found at

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 worth noting that via the von Neumann stability method the 2D explicit method is conditionally stable, therefore the magnitude of is dependent on , , and the Diffusivity constant D as seen in Figure 1 below.

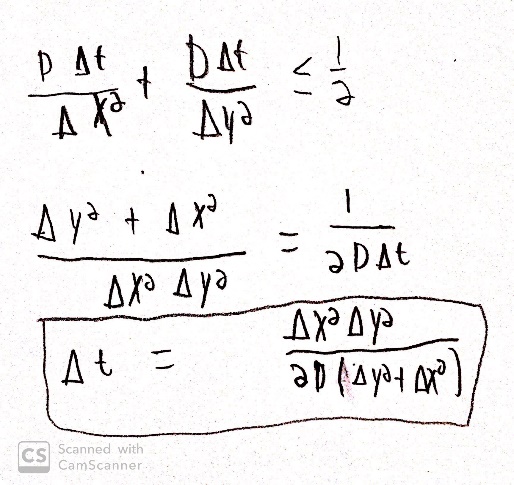


Figure 1 - Explicit Stability Condition

Moving to the implicit method, it cannot be solved for but can be solved for . The resulting equation being:

When and are converted into column vectors by stacking their rows.

= >

Now a pentagonal matrix, A, can be found such that

or

In pseudo Code where the formation of this matrix would be:

%Matrix of Coefficeints

A = zeros(Nxy,Nxy);

for i = 2:Nxy

for j = 2:Nxy-1

A(i,i) = -Lamby;

A(i,i+(Ny-1)) = -Lambx;

A(i,i+Ny) = 2\*Lambx+2\*Lamby+1;

A(i,i+(Ny+1)) = -Lambx;

A(i,i+2\*Ny) = -Lamby ;

end

end

Where Lamx = and Lamy = .

Finally, the inverse multiplication was iterated.

for k = 1:Nt

p = reshape(U,[Nxy,1])

q = A\p

UNEW = reshape(q,[Ny,Nx])

%BC

UNEW(1,:) = UTB;

UNEW(Ny,:) = UBB;

UNEW(:,1) = ULB;

UNEW(Nx,:) = UNEW(Nx-1,:)

U = UNEW

end

Technical Specifications of the Computer Used

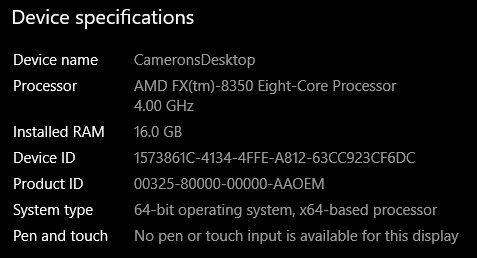


Figure 2- Technical Specifications

Results

In this study, a few assumptions were made. The first of these assumptions was that the initial condition were irrelevant to the steady state solution of the problem. As such, all nodes other than the boundary points were set to 0. The Second assumption was that the diffusivity constant, D, was equal to 1. This provided a baseline by which to compare the two methods as they are both valid for any D provided the explicit method’s stability condition is met. The number of time steps Nt was determined based on error analysis, while the spatial steps Nx and Ny were tested for a wide range of conditions, 2-50. The results of this spatial variation can be seen table x below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Explicit Method for Error = 10^-7 | | | | | |
| Grid Size(Nx,Ny) | (2,2) | (6,6) | (12,12) | (25,25) | (50,50) |
| # of Nodes | 4 | 36 | 144 | 625 | 2500 |
| Time Taken(sec) | 0.194263 | 3.145914 | 14.358919 | 61.721771 | 206.997145 |
| Iterations | 2 | 137 | 616 | 2658 | 9972 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Implicit Method for Error = 10^-7 | | | | | |
| Grid Size(Nx,Ny) | (2,2) | (6,6) | (12,12) | (25,25) | (50,50) |
| # of Nodes | 4 | 36 | 144 | 625 | 2500 |
| Time Taken(sec) | Inf | Inf | Inf | Inf | Inf |
| Iterations | Inf | Inf | Inf | Inf | Inf |

– Perform grid convergence study

– Evaluate the effect of diffusive CFL\*

As seen above the explicit scheme produced results in line with the expected theoretical behavior. In Figure X it can be seen that the boundary conditions are met and the steady state of the grid is a smooth transition between these boundaries. The implicit scheme produced very poor results in comparison to theoretical expectations. It diverges in all cases. This is attributed to an error or mistranslation of the discretized process to Matlab. The von Neumann stability method proves without doubt that this method is unconditionally stable.

– Verify the order of spatial accuracy of discretization