DIFFUSION EQUATION Bc1-4

FINAL REPORT

Professors:
Andrea Prosperetti
Amit Amritkar

By: Hung Vu
ID:1152030

MECE 5397: Scientific Computing in Mechanical Engineering
Department of Mechanical Engineering
University of Houston

Spring 2018
May 9, 2018

Abstract

A diffusion equation was provided by the professors of MECE 5397 at the University of Houston. Two methods that were elected are the explicit method and Alternating Direction Implicit scheme. The explicit method was selected because the method is fast. However, the stability of the explicit method requires that $\lambda = \frac{D\Delta t}{\Delta x^2} < \frac{1}{2}$. In other words, the pseudo code could blow up unless that condition is met. The explicit form has a second order error. In the other case, the ADI scheme, however, is unconditionally stable. The time and space will have a second order accuracy. This method uses two steps that each involve a tri-diagonal matrix. Each step is written explicitly and implicitly. This is the reason why it's called the "alternating direction." In this report, a description of the numerical method and results will be shown.

Mathematical statement of the problem

A diffusion equation was provided by the professors of MECE 5397 at the University of Houston. As shown in Figure 1, the diffusion is a 2D problem with 4 different boundary conditions. One of the four boundary conditions is a Neumann condition which ghost nodes will be used during MATLAB set-up.

MECE 5397

Project B – Diffusion Equation

Write a computer code to solve the two-dimensional diffusion equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t} \tag{19}$$

The domain of interest is the rectangle

$$a_x < x < b_x, \qquad a_y < y < b_y \tag{20}$$

and the boundary conditions

$$u(x,y,t=0)=0 \qquad u(x=a_x, y) = f_a(y), \qquad u(x=b_x, y) = g_a(y), \tag{21}$$

$$\left.\frac{\partial u}{\partial y}\right|_{y=a_y} = 0, \qquad u(x, y=b_y) = f_a(b_y) + \frac{x-a_x}{b_x-a_x}[g_a(b_y) - f_a(b_y)] \tag{22}$$

$$a_x = a_y = -\pi, \qquad b_x = b_y = \pi \tag{23}$$

$$f_a(y) = (y-a_y)^2 \cos\frac{\pi y}{a_y}, \qquad g_a(y) = y(y-a_y)^2 \tag{24}$$

Use ghost node(s) for Neumann condition(s).
Carry out the time integration to steady state, i.e., until the result becomes independent of time.

Figure 1: Diffusion Equation provided by professors

Discretized version of the equations

Discretization is the process of changing an equation into a form that can be suitable for computers to identify. As shown in Figure 2, an explicit discretization method was used to convert the diffusion equation that was given.

Explicit Method

$$\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2} = \frac{du}{dt}$$

$$\frac{u^n_{i-1,j} - 2u^n_{i,j} + u^n_{i+1,j}}{\Delta x^2} + \frac{u^n_{i,j-1} - 2u^n_{i,j} + u^n_{i,j+1}}{\Delta y^2} = \frac{u^{n+1}_{i,j} - u^n_{i,j}}{\Delta t}$$

$$u^{n+1}_{i,j} - u^n_{i,j} = \lambda\left[u^n_{i-1,j} + u^n_{i,j-1} - 4u^n_{i,j} + u^n_{i+1,j} + u^n_{i,j+1}\right]$$

$$\lambda = \frac{\Delta t}{\Delta x^2} \qquad \Delta x^2 = \Delta y^2$$

$$\boxed{u^{n+1}_{i,j} = \lambda\left[u^n_{i-1,j} + u^n_{i,j-1} - 4u^n_{i,j} + u^n_{i+1,j} + u^n_{i,j+1}\right] + u^n_{i,j}}$$

Figure 2:Explicit Discretization

The second method that was used to solve the diffusion equation was the Alternating Direction Implicit scheme. This method uses two steps, each involving a tri-diagonal matrix, to solve the problem. As shown in Figure 3, the first step is explicit in y and implicit in x. In figure 4 however, is the second step that is explicit in x and implicit in y. This is the reason why it’s called the “alternating direction.”

$$\frac{u^{n+1/2}_{i,j} - u^{n}_{i,j}}{\Delta t/2} = \frac{u^{n}_{i-1,j} - 2u^{n}_{i,j} + u^{n}_{i+1,j}}{\Delta x^2} + \frac{u^{n+1/2}_{i,j-1} - 2u^{n+1/2}_{i,j} + u^{n+1/2}_{i,j+1}}{\Delta y^2}$$

$$u^{n+1/2}_{i,j} - u^{n}_{i,j} = \frac{1}{2}\lambda\left[u^{n}_{i-1,j} - 2u^{n}_{i,j} + u^{n}_{i+1,j} + u^{n-1/2}_{i,j-1} - 2u^{n+1/2}_{i,j} + u^{n+1/2}_{i,j+1}\right]$$

$$\lambda = \frac{\Delta t}{\Delta x^2}$$

$$\Delta x^2 = \Delta y^2$$

$$\boxed{-\lambda u^{n+1/2}_{i,j-1} + 2(1+\lambda)u^{n+1/2}_{i,j} - \lambda u^{n+1/2}_{i,j+1} = \lambda u^{n}_{i-1,j} + 2(1-\lambda)u^{n}_{i,j} + \lambda T^{n}_{i+1,j}}$$

Figure 3: ADI Discretization (explicit part)

$$\frac{u^{n+1}_{i,j} - u^{n+1/2}_{i,j}}{\Delta t/2} = \frac{u^{n+1}_{i+1,j} - 2u^{n+1}_{i,j} + u^{n+1}_{i-1,j}}{\Delta x^2} + \frac{u^{n+1/2}_{i,j+1} - 2u^{n+1/2}_{i,j} + u^{n+1/2}_{i,j-1}}{\Delta y^2}$$

$$2u^{n+1}_{i,j} - 2u^{n+1/2}_{i,j} = \frac{\Delta t}{\Delta x^2}\left[u^{n+1}_{i+1,j} - 2u^{n+1}_{i,j} + u^{n+1}_{i-1,j} + u^{n+1/2}_{i,j+1} - 2u^{n+1/2}_{i,j} + u^{n+1/2}_{i,j-1}\right]$$

$$\lambda = \frac{\Delta t}{\Delta x^2} \qquad \Delta x^2 = \Delta y^2$$

$$\boxed{-\lambda u^{n+1}_{i-1,j} + 2(1+\lambda)u^{n+1}_{i,j} - \lambda u^{n+1}_{i+1,j} = \lambda^{n+1/2}_{i,j-1} + 2(1-\lambda)u^{n+1/2}_{i,j} + \lambda u^{n+1/2}_{i,j+1}}$$

Figure 4:ADI Discretization (implicit part)

Description of the numerical method

Technical specifications of the computer used

Majority of the code and experiment was done at the computer lab at the University of Houston. The technical specifications of the computer are listed below:

Number of sockets - 4

CPU model name - Intel® Xeon® CPU E5620 @2.40 GHz

Number of cores/CPU - 1

Current CPU clock frequency - CPU MHz: 2394.000

Max CPU clock frequency - 2660 MHz

L1, L2, and L3 cache sizes –

-L1i cache: 32K

- L1d cache: 32K

- L2 cache: 256K

- L3 cache: 12288K

Number of memory channels - 64

Size of each DIMM - 27 bytes

Total DRAM per CPU - 27 bytes

Results