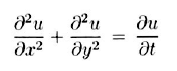
**Mathematical statement of the problem**

The problem to be solved takes the form of a two-dimensional diffusion equation. The general form of the diffusion equation in two space dimensions is:



(1)

In this form, x and y can be considered space dimensions and t and be considered time. The problem that will be solved in this report is given by:



(2)

On the domain of interest:



(4)

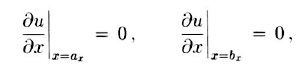
(3)



With the following boundary conditions:

(5)





(7)

(6)

The boundary conditions given by Eq. (5) are Dirichlet boundary conditions. The value of the function u at these two boundaries will not change with respect to time. The boundary conditions given by Eq. (7) are Neumann boundary conditions.

**Discretization Using the Explicit Method**

In order to compute the steady state solution of the equation across the domain, the equation must be discretized. Two methods will be used, the explicit method and the implicit method.

For the explicit method, take:

(8)

(9)

Where N and M are positive integers. Set:

(10)

(11)

(12)

(13)

The forward difference method for approximating a first derivative gives:

(14)

The central difference method for approximating a second derivative gives:

(15)

(16)

If:

(17)

(18)

The explicit method discretization of the equation is:

(19)

**Discretization Using the Implicit Method**

The implicit method works much the same way as the explicit method, however the second derivatives in the space dimensions are taken at i+1, therefore:

(20)

(21)

Using the same assumptions and notation as the explicit method, the implicit method discretization of the equation is:

(22)

**Model Algorithms**

In order to apply these two methods to the problem stated, attention must be paid to the Neumann boundary conditions. The values of the nodes at the boundaries is unknown. In order to solve for the values of u at these boundaries, ghost nodes are introduced at:

(23)

(24)

The forward difference method concludes that:

(27)

(26)

(25)

Now the values at the boundaries at i+1 can be calculated using the method previously described.

*Steady State Stopping Condition*

The steady state formula of U reduces to:

;

(28)

Using the central difference method (Eq. 15 and 16), this discretizes into Laplacian difference equation:

(29)

Given the boundary conditions, this discretization becomes a linear system of equations. Once the steady state values at all the unknown nodes are solved for, the steady state values of the function can be compared to the values of for each n until the function is determined to be sufficiently close to steady state. To do this, the L∞ error is calculated using the formula:

(30)

Once is within a certain tolerance is considered to have reached steady state.

*Explicit Algorithm and Pseudocode*

1. *Create the mesh, determine the time step*
   * *Set: N=number of nodes*
   * *Set: h=2\*pi/(N-1)*
   * *Set: dt=h^2/6*
   * *Set: x = -pi to pi, step h*
   * *Set y = -pi to pi, step h*
2. *Set the Initial Conditions and Boundary Conditions*
   * *Nm2=N-2*
   * *Number of interior nodes: intn=Nm2\*N*
   * *Uvec=zeros(intn,1)*
   * *Set phi to be the boundary condition at y = ay = -pi*
   * *Set psi to be the boundary condition at y = by = pi*
   * *BCvec=zeroes(intn,1)*
   * *BCvec(1:N)=phi(x(1:N))*
   * *BCvec(intn-N:intn)=psi(x(1:N))*
3. *Create E*
   * *lambda=dt/(dx^2)*
   * *E is a matrix such that Uvec(:,:,i+1)=E\*Uvec(:,:,i)+lambda\*BCvec*
4. *Solve to Steady State*
   * *Determine Usvec, the vector of the interior nodes at steady state*
   * *While MaxDif>0.001*
   * *i=i+1*
   * *Uvec(:,;,i)=E\*Uvec(:,:,i-1)+lambda\*BCvec*
   * *MaxDif=abs(max(Uvec(:,;,i)-Usvec))*

*Implicit Algorithm and Pseudocode*

1. *Create the mesh, determine the time step*
   * *Set: N=number of nodes*
   * *Set: h=2\*pi/(N-1)*
   * *Set: dt=h^2/6*
   * *Set: x = -pi to pi, step h*
   * *Set y = -pi to pi, step h*
2. *Set the Initial Conditions and Boundary Conditions*
   * *Nm2=N-2*
   * *Number of interior nodes: intn=Nm2\*N*
   * *Uvec=zeros(intn,1)*
   * *Set phi to be the boundary condition at y = ay = -pi*
   * *Set psi to be the boundary condition at y = by = pi*
   * *BCvec=zeroes(intn,1)*
   * *BCvec(1:N)=phi(x(1:N))*
   * *BCvec(intn-N:intn)=psi(x(1:N))*
3. *Create I*
   * *lambda=dt/(dx^2)*
   * *E is a matrix such that Uvec(:,:,i+1)=I/=\(Uvec(:,:,i)+lambda\*BCvec)*
4. *Solve to Steady State*
   * *Determine Usvec, the vector of the interior nodes at steady state*
   * *While MaxDif>0.001*
   * *i=i+1*
   * *Uvec(:,;,i)=I\(Uvec(:,:,i-1)+lambda\*BCvec)*
   * *MaxDif=abs(max(Uvec(:,;,i)-Usvec))*

**Verification and Grid Convergence Study**

*Manufactured Solutions*

In order to verify the models, the method of manufactured solutions was used. The code was modified to solve:

(31)

(32)

On the same domain of interest, where:

(33)

If the code is valid, the approximation of v will approach the exact solution of v. In addition, the codes were verified using a grid convergence study to determine if the models are sensitive to the choice of grid size. The codes were also verified by comparing the two methods. If both methods produce the same results, it is an indication that the models are valid.

For each method, the code was set to output the exact solution of v (Eq. 31) and the approximated solution of v at a given time t=2.0s, with the given time step of 0.01s for different mesh sizes. The maximum absolute difference between the exact solution and the approximated solution across all interior nodes was calculated.

Figure 1: Error between Exact Solution of v(x,y,t) and Model as N is increased for the method of manufactured solutions

From Figure 1, it is clear that the method of manufactured solutions verifies the models. Even with a very coarse mesh (N=5), the maximum difference between the exact solution of v and the model approximation for both methods was only 0.15, and steadily decreased as N was increased.

Figure 2: Maximum difference between exact solution of v(x,y,t) and model as time step is decreased from 0.1 to 0.01 for method of manufactured solutions, N=20.

It is useful to examine how the grid responds with respect to the time step using the method of manufactured solutions in order to better understand the results of the model. The implicit method converges unconditionally, however, as Figure 2 demonstrates, the explicit method requires a very small time step, otherwise, the error will be very large.

*Grid Convergence*

Another grid convergence study was also performed on the models themselves, first to evaluate how the approximated solutions change as the number of spatial nodes is changed, and then again to evaluate how the approximated solutions change as the time step is changed. An elegant way to observe these effects is to run the models until the error reaches a specified tolerance from the steady state solution, and then return the time in seconds at which the tolerance was reached. To do this, the error was calculated and the tolerance was set to 0.001. Because the time at which u(x,y,t) reaches steady state is fixed and the error is fixed by the tolerance, the steady state time returned by the models should be very close to the same for different grids.

Figure 3: Grid Convergence Study of the spatial mesh of the Explicit Method. N is the number of nodes in x, equal to the number of nodes in y. Steady State Time reflects the time it takes for the model to approximate u(x,y,t) such that is less than the tolerance of 0.001. The time step was taken to be 0.001s.

The steady state time reflects the time t at which all the nodes of the approximation are within a tolerance of 0.001 of the steady state solution. For 30 ≤ N ≤ 80, u(x,y,t) reaches the steady state solution in 37 s < t < 38 s. Similarly, the implicit model produced values of t= 37.96s and t=37.7s for N=30 and N=40 respectively.

Figure 4: Grid Convergence Study of the spatial mesh of the Implicit Method. N is the number of nodes in x, equal to the number of nodes in y. Steady State Time reflects the time it takes for the model to approximate u(x,y,t) such that is less than the tolerance of 0.001. The time step was taken to be 0.01s.

Figure 5: Grid Convergence Study of the effect of time step on steady state time. Steady state time reflects the time it takes for the model to approximate u(x,y,t) such that is less than the tolerance of 0.001. N was taken to be 20.

Both methods were within the specified tolerance of the steady state solution at or before t=38s independent of the grid size or time step, therefore it is concluded that:

(34)

As one final test to verify the models, the codes can be modified to output u(x,y,tf) for a specified mesh size, time step, and a specified time t=tf, chosen to be 39 s. The implicit and the explicit models can then be directly compared using the formula for the L∞ error (Eq 30). The simulation was run and the inputs and outputs are shown below.

Table 1: 𝜖∞ of Explicit and Implicit Model for the Given Inputs

|  |  |
| --- | --- |
| Input: | |
| N | 40 |
| dt | 0.001 |
| tf | 39 |
| Output: | |
| 𝜖∞ | 9.587E-06 |

The models are thus verified.

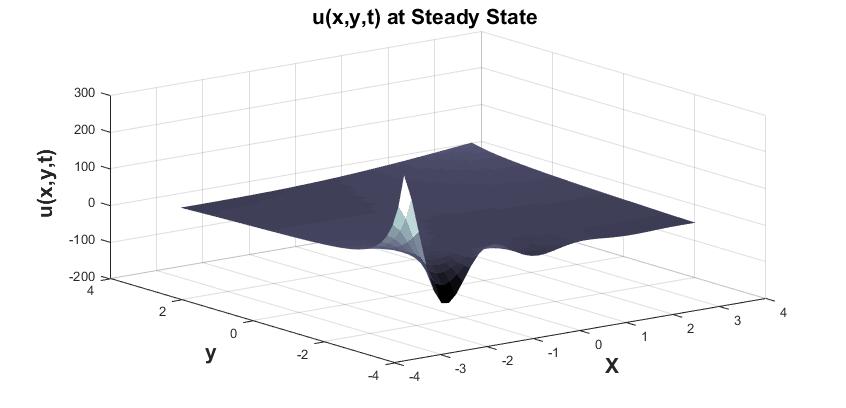
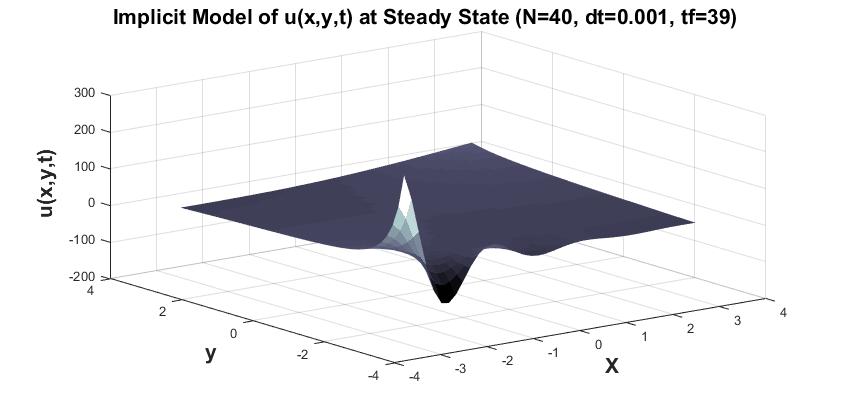
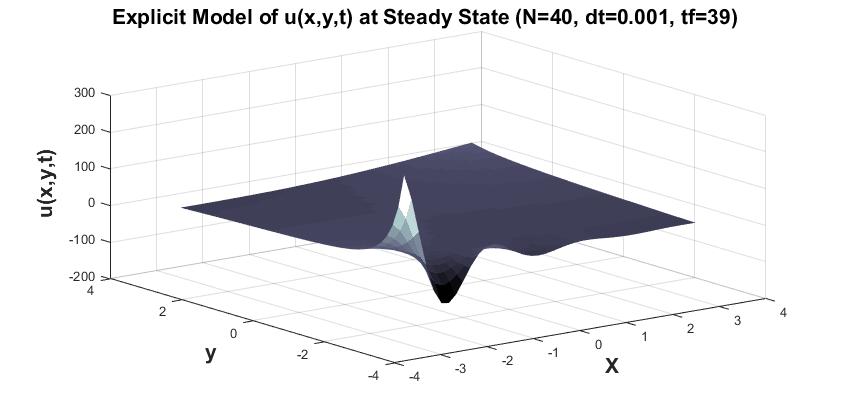
**Results: Explicit and Implicit Models of u(x,y,t) at Steady State**

Figure 6: Surface Plots of u(x,y,t) at Steady State.