Solve the Stefan problem in two space dimensions

1. Question

The one phase Stefan problem in two space variables is given:

Where , and subject to the boundary conditions:

and the initial temperature distribution:

Here denotes the temperature of the water at a point (x, y) in the region at any time t. The region is assumed to be occupied by ice at . defines the moving boundary and g(x) is a given function such that 。

1. Analysis

The approximate method used to solve to this problem was discussed in detail in [1]. So, following [1], we assume that the temperature profile for constant *x* is given by

(7)

Where s and B are still to be found. From [1], we have these following two equations:

(8)

We solve the differential equations (9) and (8) in a step by step manner employing central difference in the space direction and forward difference in the time direction which are shown as follow:

(10)

(12)

(13)

Where and k = 1,2,…….Here . For the initial conditions we have:

and

1. Experiment

In order to test the proposed numerical scheme for Stefan problem, we take the same case of initial positions of the interface as considered by them, namely,

The time and space steps are taken to be and . I display the result the same as [1], which comparison of position of the moving boundary along x = 0 and x = 1 at various times.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (16) | 0.05 | 4136.5699 | (17) | 0.05 | 4250.8142 | (18) | 0.05 | 8987.7346 |
|  | 0.10 | 5009.0097 |  | 0.10 | 5199.7802 |  | 0.10 | 9666.1492 |
|  | 0.15 | 5749.933 |  | 0.15 | 5996.0705 |  | 0.15 | 10208.8899 |
|  | 0.20 | 6404.886 |  | 0.20 | 6692.7863 |  | 0.20 | 10679.3126 |
|  | 0.25 | 6997.7505 |  | 0.25 | 7317.7005 |  | 0.25 | 11104.4126 |
|  | 0.30 | 7542.977 |  | 0.30 | 7887.7052 |  | 0.30 | 11497.9706 |
|  | 0.35 | 8050.1388 |  | 0.35 | 8414.0537 |  | 0.35 | 11867.9369 |
|  | 0.40 | 8525.9888 |  | 0.40 | 8904.7144 |  | 0.40 | 12219.3195 |

Table1: x = 0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (16) | 0.05 | 5491.6013 | (17) | 0.05 | 7415.8463 | (18) | 0.05 | 10407.645 |
|  | 0.10 | 6033.4856 |  | 0.10 | 7784.9545 |  | 0.10 | 10781.3038 |
|  | 0.15 | 6563.8815 |  | 0.15 | 8146.7803 |  | 0.15 | 11131.1314 |
|  | 0.20 | 7070.4357 |  | 0.20 | 8506.2093 |  | 0.20 | 11464.0373 |
|  | 0.25 | 7552.3109 |  | 0.25 | 8862.382 |  | 0.25 | 11783.9352 |
|  | 0.30 | 8011.2902 |  | 0.30 | 9214.0848 |  | 0.30 | 12093.1826 |
|  | 0.35 | 8449.6287 |  | 0.35 | 9560.4662 |  | 0.35 | 12393.3833 |
|  | 0.40 | 8869.483 |  | 0.40 | 9901.021 |  | 0.40 | 12685.7335 |

Table2: x = 1

|  |
| --- |
|  |

Figure1: the moving boundary condition under equation (16), the curve from bottom to up is t = 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40

|  |
| --- |
|  |

Figure2: the moving boundary condition under equation (17), the curve from bottom to up is t = 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40

|  |
| --- |
|  |

Figure3: the moving boundary condition under equation (18), the curve from bottom to up is t = 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40

By using equation (7), we can get the final result of u(x, y, t), some rsults are shown below:

|  |
| --- |
|  |

Figure4:u(x,y,0.05) under equation (16)

|  |
| --- |
|  |

Figure5:u(x,y,0.05) under equation (17)

|  |
| --- |
|  |

Figure6:u(x,y,0.05) under equation (18)

We can see that the figures shown can fit the initial conditions and boundary conditions (2)-(6), which means the numerical scheme has a high accuracy on the two dimensional Stefan problem.

1. Limitations

We solve the differential equations (9) and (8) in a step by step manner employing central difference in the space direction and forward difference in the time direction, which are very sensitive to the step length of time domain and space domain. But I do not want to prove it theoretically here, I only give some numerical result to show the relations between stability of the scheme and step length of the two. Here we just take equation (16) for example:

|  |  |  |
| --- | --- | --- |
| Step length of space | Stability of the scheme | Speed of the cheme |
| 0.1 | Stable | Fast |
| 0.01 | Stable | Fast |
| 0.001 | Stable | Slow |
| 0.0001 | Stable | Slow |

Table3: stability along the x direction

|  |  |  |
| --- | --- | --- |
| Step length of time | Stability of the scheme | Speed of the cheme |
| 0.1 | Not Stable | Useless |
| 0.01 | Not Stable | Useless |
| 0.001 | Stable | Fast |
| 0.0001 | Stable | Slow |

Table4: stability along the t direction

1. Conclusion

This numerical scheme for two dimensional Stefan problem is easy and can be used for real-time application. I use a forward and central difference which are sensitive to the step length, but by choosing the step length appropriately, we still can get a high accuracy on this problem

Reference

[1] R.S GUPTA. Solution of a weakly two-dimensional melting problem by an approximate method.

Journal of Computational and Applied Mathematics.

[2] A. Kharab. Spreadsheet Solution of a Two-Dimensional Stefan Problem Using an Approximate Method. Heat Transfer Engineering

Appendix A:

function [S B] = stefan\_cal(deltaX, deltaT, Xstep, Tstep, S0, B0)

lambda = deltaT / (deltaX).^2;

%% i = 0,1,2...N, space domain

%% k = 0,1,2...M, time domain

%% initialize

M = Tstep;

N = Xstep;

B = zeros(Tstep+1, Xstep+1);

S = zeros(Tstep+1, Xstep+1);

B(1,:) = B0;

S(1,:) = S0;

%% iteration

for i = 2:M+1

for j = 1:N+1

%% S(k+1,0) and B(k+1,0)

if j == 1

S(i, 1) = S(i-1, 1) + 2 \* lambda / ( S(i-1, 1)\*(8 + B(i-1, 1))) \* ...

( S(i-1, 1) \* (2 + B(i-1, 1) ) \* (S(i-1, 2) - S(i-1, 1)) + ...

3 \* (deltaX).^2 \* (2 + B(i-1, 1)) );

B(i, 1) = B(i-1, 1) + 12 \* lambda / ( S(i-1, 1).^2 \* (8 + B(i-1, 1) ) ) \* ...

( 2 \* S(i-1, 1) \* (S(i-1, 2) - S(i-1, 1)) - deltaX.^2 \* (B(i-1, 1).^2 + 10 \* B(i-1, 1) + 10)) + 2 \* lambda \* (B(i-1, 2) - B(i-1, 1));

continue;

end

if j == N+1

S(i, N+1) = S(i-1, N+1) + 2 \* lambda / (S(i-1, N+1)\*(8 + B(i-1, N+1))) \* ...

( S(i-1, N+1) \* (2 + B(i-1, N+1) ) \* (S(i-1, N) - S(i-1, N+1)) + ...

3 \* (deltaX).^2 \* (2 + B(i-1, N+1)) );

B(i, N+1) = B(i-1, N+1) + 12 \* lambda / (S(i-1, N+1).^2 \* (8 + B(i-1, N+1) ) ) \* ...

( 2 \* S(i-1, N+1) \* (S(i-1, N) - S(i-1, N+1)) - deltaX.^2 \* ( B(i-1, N+1).^2 + 10 \* B(i-1, N+1) + 10) ) +...

2 \* lambda \* (B(i-1, N) - B(i-1, N+1));

continue;

end

%%------------------------------------------------------------------%%

S(i, j) = S(i-1, j) + lambda / (4 \* S(i-1, j) \* (8 + B(i-1, j))) \* ...

( 2 \* (3 + B(i-1, j)) \* (S(i-1, j+1) - S(i-1, j-1)).^2 + ...

4 \* S(i-1, j) \* (2 + B(i-1, j)) \* (S(i-1, j+1) - 2 \* S(i-1, j) + S(i-1, j-1)) + ...

24 \* deltaX.^2 \* (2 + B(i-1, j)) + 2 \* S(i-1, j) \* (S(i-1,j+1) - S(i-1,j-1)) \* (B(i-1, j+1) - B(i-1, j-1)) );

B(i, j) = B(i-1, j) + lambda / (4 \* S(i-1, j).^2 \* (8 + B(i-1, j)) ) \* ...

( -2 \* (30 + 13 \* B(i-1, j) + B(i-1, j).^2 ) \* (S(i-1, j+1) - S(i-1, j-1)).^2 + ...

48 \* S(i-1, j) \* ( S(i-1, j+1) - 2 \* S(i-1, j) + S(i-1, j-1) ) - ...

48 \* deltaX.^2 \* ( B(i-1, j).^2 + 10 \* B(i-1, j) + 10) - ...

4 \* S(i-1, j) \* ( S(i-1, j+1) - S(i-1, j-1)) \* ( B(i-1, j+1) - B(i-1, j-1)) ) + ...

lambda \* ( B(i-1, j+1) - 2 \* B(i-1, j) + B(i-1, j-1) );

end

end

end

Appendix B:

% S is the S(x,t)

% B is the B(x,t)

% t mean the time

% deltaT means the step length of T

% deltaX means the step length of X

% Ystep means the number of steps along Y

function [X Y Z] = cal\_U(B, S, t, deltaT, deltaX, Ystep)

Xstep = 1 / deltaX;

Tstep = t / deltaT + 1;

ymax = max(S(Tstep,:));

x = 0:deltaX:1;

y = 0:(ymax/Ystep):ymax;

[X Y] = meshgrid(x, y);

[rows cols] = size(X);

for i = 1:rows

for j = 1:cols

Xindex = int32(X(i,j)/deltaX + 1);

Z(i,j) = -(1 + B(Tstep, Xindex)) \* ( Y(i,j) / S(Tstep, Xindex)).^2 + B(Tstep,Xindex) \* (Y(i,j)/S(Tstep, Xindex)) + 1;

end

end

end

Appendix C

clear;

close all;

clc;

Xstep = 10;

Tstep = 1000;

deltaX = 1 / Xstep;

deltaT = 1 / Tstep;

tic;

%% S(x, 0) = g1(x)

b1 = -ones(1, Xstep+1);

s1 = g1((0:Xstep) \* deltaX);

[S1 B1] = stefan\_cal(deltaX, deltaT, Xstep, Tstep, s1, b1);

format short

disp(['.........................................................................................']);

disp(['.........................................................................................']);

disp(['s(0.05, 0) of g1:' num2str(1e4\*S1(51,1)) '|| s(0.05, 1) of g1:' num2str(1e4\*S1(51,11))]);

disp(['s(0.10, 0) of g1:' num2str(1e4\*S1(101,1)) '|| s(0.05, 1) of g1:' num2str(1e4\*S1(101,11))]);

disp(['s(0.15, 0) of g1:' num2str(1e4\*S1(151,1)) '|| s(0.05, 1) of g1:' num2str(1e4\*S1(151,11))]);

disp(['s(0.20, 0) of g1:' num2str(1e4\*S1(201,1)) '|| s(0.05, 1) of g1:' num2str(1e4\*S1(201,11))]);

disp(['s(0.25, 0) of g1:' num2str(1e4\*S1(251,1)) '|| s(0.05, 1) of g1:' num2str(1e4\*S1(251,11))]);

disp(['s(0.30, 0) of g1:' num2str(1e4\*S1(301,1)) '|| s(0.05, 1) of g1:' num2str(1e4\*S1(301,11))]);

disp(['s(0.35, 0) of g1:' num2str(1e4\*S1(351,1)) '|| s(0.05, 1) of g1:' num2str(1e4\*S1(351,11))]);

disp(['s(0.40, 0) of g1:' num2str(1e4\*S1(401,1)) '|| s(0.05, 1) of g1:' num2str(1e4\*S1(401,11))]);

disp(['.........................................................................................']);

figure

plot(S1(51,:) ,'r\*');hold on;plot(S1(51,:))

plot(S1(101,:),'ro');hold on;plot(S1(101,:))

plot(S1(151,:),'rx');hold on;plot(S1(151,:))

plot(S1(201,:),'rs');hold on;plot(S1(201,:))

plot(S1(251,:),'rd');hold on;plot(S1(251,:))

plot(S1(301,:),'rv');hold on;plot(S1(301,:))

plot(S1(351,:),'r<');hold on;plot(S1(351,:))

plot(S1(401,:),'r>');hold on;plot(S1(401,:))

title('g1(x)')

%% S(x, 0) = g2(x)

b2 = -ones(1, Xstep+1);

s2 = g2((0:Xstep) \* deltaX);

[S2 B2] = stefan\_cal(deltaX, deltaT, Xstep, Tstep, s2, b2);

disp(['.........................................................................................']);

disp(['s(0.05, 0) of g2:' num2str(1e4\*S2(51,1)) '|| s(0.05, 1) of g2:' num2str(1e4\*S2(51,11))]);

disp(['s(0.10, 0) of g2:' num2str(1e4\*S2(101,1)) '|| s(0.05, 1) of g2:' num2str(1e4\*S2(101,11))]);

disp(['s(0.15, 0) of g2:' num2str(1e4\*S2(151,1)) '|| s(0.05, 1) of g2:' num2str(1e4\*S2(151,11))]);

disp(['s(0.20, 0) of g2:' num2str(1e4\*S2(201,1)) '|| s(0.05, 1) of g2:' num2str(1e4\*S2(201,11))]);

disp(['s(0.25, 0) of g2:' num2str(1e4\*S2(251,1)) '|| s(0.05, 1) of g2:' num2str(1e4\*S2(251,11))]);

disp(['s(0.30, 0) of g2:' num2str(1e4\*S2(301,1)) '|| s(0.05, 1) of g2:' num2str(1e4\*S2(301,11))]);

disp(['s(0.35, 0) of g2:' num2str(1e4\*S2(351,1)) '|| s(0.05, 1) of g2:' num2str(1e4\*S2(351,11))]);

disp(['s(0.40, 0) of g2:' num2str(1e4\*S2(401,1)) '|| s(0.05, 1) of g2:' num2str(1e4\*S2(401,11))]);

disp(['.........................................................................................']);

figure

plot(S2(51,:) ,'r\*');hold on;plot(S2(51,:))

plot(S2(101,:),'ro');hold on;plot(S2(101,:))

plot(S2(151,:),'rx');hold on;plot(S2(151,:))

plot(S2(201,:),'rs');hold on;plot(S2(201,:))

plot(S2(251,:),'rd');hold on;plot(S2(251,:))

plot(S2(301,:),'rv');hold on;plot(S2(301,:))

plot(S2(351,:),'r<');hold on;plot(S2(351,:))

plot(S2(401,:),'r>');hold on;plot(S2(401,:))

title('g2(x)')

%% S(x, 0) = g3(x)

b3 = -ones(1, Xstep+1);

s3 = g3((0:Xstep) \* deltaX);

[S3 B3] = stefan\_cal(deltaX, deltaT, Xstep, Tstep, s3, b3);

disp(['.........................................................................................']);

disp(['s(0.05, 0) of g3:' num2str(1e4\*S3(51,1)) '|| s(0.05, 1) of g3:' num2str(1e4\*S3(51,11))]);

disp(['s(0.10, 0) of g3:' num2str(1e4\*S3(101,1)) '|| s(0.05, 1) of g3:' num2str(1e4\*S3(101,11))]);

disp(['s(0.15, 0) of g3:' num2str(1e4\*S3(151,1)) '|| s(0.05, 1) of g3:' num2str(1e4\*S3(151,11))]);

disp(['s(0.20, 0) of g3:' num2str(1e4\*S3(201,1)) '|| s(0.05, 1) of g3:' num2str(1e4\*S3(201,11))]);

disp(['s(0.25, 0) of g3:' num2str(1e4\*S3(251,1)) '|| s(0.05, 1) of g3:' num2str(1e4\*S3(251,11))]);

disp(['s(0.30, 0) of g3:' num2str(1e4\*S3(301,1)) '|| s(0.05, 1) of g3:' num2str(1e4\*S3(301,11))]);

disp(['s(0.35, 0) of g3:' num2str(1e4\*S3(351,1)) '|| s(0.05, 1) of g3:' num2str(1e4\*S3(351,11))]);

disp(['s(0.40, 0) of g3:' num2str(1e4\*S3(401,1)) '|| s(0.05, 1) of g3:' num2str(1e4\*S3(401,11))]);

disp(['.........................................................................................']);

disp(['.........................................................................................']);

figure

plot(S3(51,:) ,'r\*');hold on;plot(S3(51,:))

plot(S3(101,:),'ro');hold on;plot(S3(101,:))

plot(S3(151,:),'rx');hold on;plot(S3(151,:))

plot(S3(201,:),'rs');hold on;plot(S3(201,:))

plot(S3(251,:),'rd');hold on;plot(S3(251,:))

plot(S3(301,:),'rv');hold on;plot(S3(301,:))

plot(S3(351,:),'r<');hold on;plot(S3(351,:))

plot(S3(401,:),'r>');hold on;plot(S3(401,:))

title('g3(x)')

%--------------------------------------------------------------------------------------------------%

% calculate U(x,y,t) = -(1+B)\*(y/s)^2 + B \* (y/s)"2 + 1; %

%--------------------------------------------------------------------------------------------------%

figure;

Ystep = 20;

[X Y Z] = cal\_U(B1, S1, 0.05, deltaT, deltaX, Ystep);

mesh(X,Y,Z)

title('g1(x)')

figure;

Ystep = 20;

[X Y Z] = cal\_U(B2, S2, 0.05, deltaT, deltaX, Ystep);

mesh(X,Y,Z)

title('g2(x)')

figure;

Ystep = 20;

[X Y Z] = cal\_U(B2, S2, 0.05, deltaT, deltaX, Ystep);

mesh(X,Y,Z)

title('g3(x)')

toc;

Appendix D:

function result = g1(x)

result = 0.5 - 0.2 .\* cos(0.5 .\* pi .\* x);

end

function result = g2(x)

result = 0.5 - 0.2 \* cos(pi \* x);

end

function result = g3(x)

result = 1.0 - 0.2 \* exp(-25 \* x.^2);

end