Lecture 2 Implementation HW#2 (a) 2.2

Algorithm (Explicit Scheme)

INPUT

begin

$N, J, T, \Delta t, \Delta x$ $\Delta t = \frac{T}{N}, \Delta x = \frac{1}{J}$

end

INITIALIZATION

begin

$\nu = \frac{\Delta t}{(\Delta x)^2}$

$t = 0$

for $j = 0, 1, \cdots, J$ do

$x_j = j\Delta x, \; U_j = u^0(x_j)$

end

end

IMPLEMENTATION

for $n = 0, 1, \cdots, N-1$ do

for $j = 1, 2, \cdots, J-1$ do

$V_j = U_j + \nu(U_{j+1} - 2U_j + U_{j-1})$

end,

$V_0 = 0,$

$V_J = 0,$

~~end~~
~~for~~
~~Output begin~~

for $j = 0, 1, 2, \cdots, J$ do

$U_j = V_j$

end;

$t = t + \Delta t,$

end

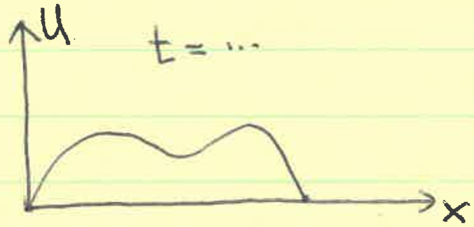
Output (if necessary)

begin t, U, x end

Note 1 Three arrays $x(0:J)$, $U(0:J)$, $V(0:J)$

Note 2 Fortran 77 : J and j, N and n are the same
Replace J and N by, say, JMAX and NMAX, respectively.

Note 3 Show the results in figures



Example $u^0(x) = \begin{cases} 2x & \text{if } 0 \le x \le \frac{1}{2} \\ 2-2x & \text{if } \frac{1}{2} \le x \le 1 \end{cases}$

$J = 20$ ($\Delta x = 0.05$) $\Delta t = 0.0012$ good results

$J = 20$ ($\Delta x = 0.05$) $\Delta t = 0.0013$ figures.

(Page 12).

```
function heat1d(J, N)

% initialization

    T = 0.2;
    t = 0.0;

    dx = 1/(J-1);
    dt = (T-t)/N;
    nu = dt/(dx*dx);

    x = linspace(0,1,J)';
    u = u_init(x);

    u1 = zeros(J,1);

% main loop for integration

    for n = 1:N
       for j = 2:J-1
          u1(j) = u(j) + nu*(u(j+1)-2*u(j)+u(j-1));
       end
       % for BCs
       u1(1) = 0.0;
       u1(J) = 0.0;
       % update time and solution
       t = t + dt;
       u = u1;
       % plot the solution
       figure(1)
       plot(x,u,'r-o');
       axis([0 1 0 1]);
       drawnow;
    end

% print parameters

    fprintf('\n%d %d %e %e %e\n', J, N, dt, dx, nu);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function u = u_init(x)

    u = sin(pi*x);
```

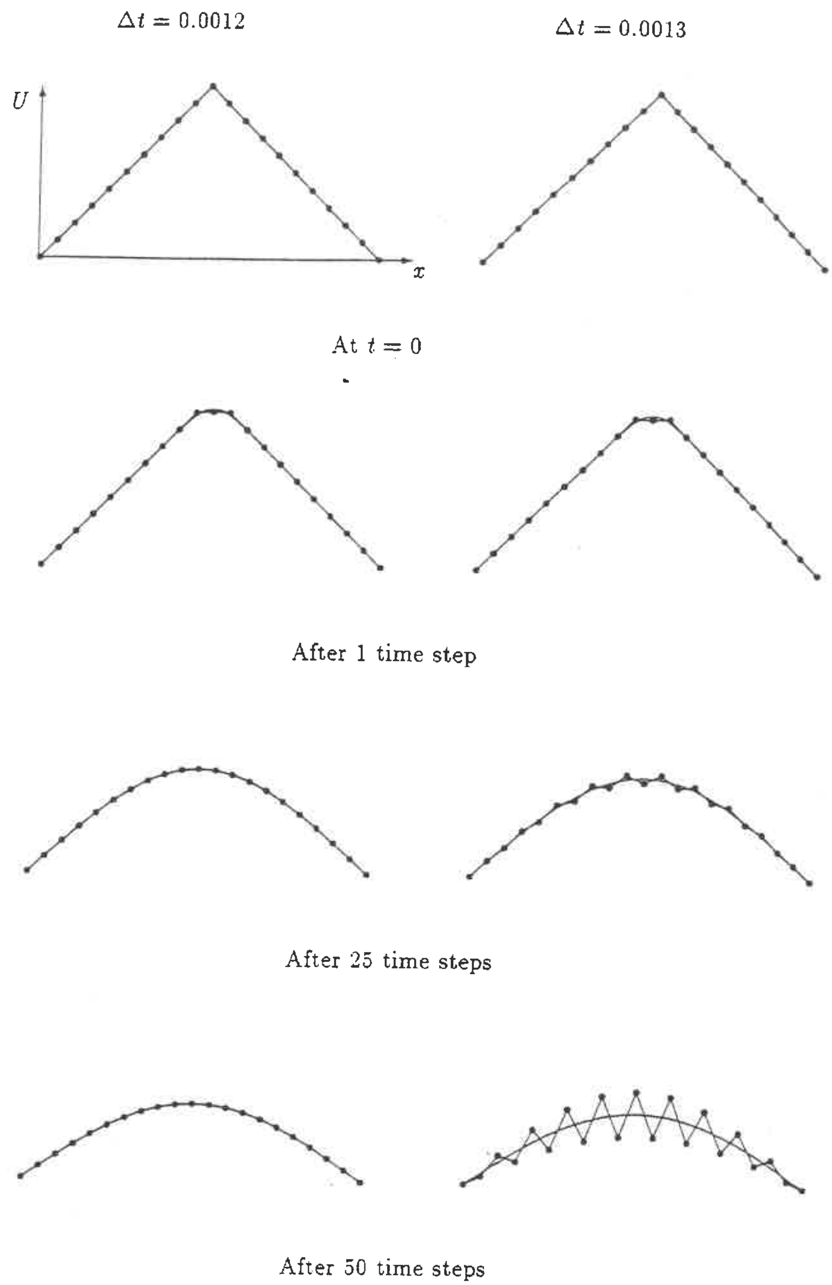


Fig. 2.2. Results obtained for the data of (2.24) with the explicit method; $J = 20$, $\Delta x = 0.05$. The exact solution is shown by the full curved line.

heat1d.cpp

```
// c++ driver for the explcit scheme for solving 1d heat equation

#include <math.h>
#include <stdio.h>
#include <assert.h>
#include <stdlib.h>
#include <time.h>

// prototype of defined functions

double u0(double x);
double b0(double t);
double b1(double t);

// main program

int main()
{

int j, J, n, N;
double t, T, dt, x, dx, nu;
double *u, *u1;
FILE *out;

out = fopen("soln.dat", "w");

   T = 0.2;
   t = 0.0;

// input J and N

   printf("J = "); scanf("%d", &J);
   dx = 1.0/(double) (J-1);
   dt = 0.5*dx*dx; N = (int) (T/dt) + 1;
   printf(" for nu = 0.5  N is about: %4d\n", N);
   printf("N = "); scanf("%d", &N);

// allocate space for u and u1

   u = new double[J]; assert(u != NULL);
   u1 = new double[J]; assert(u1 != NULL);

// calculate basic parameters

   dt = T/(double) (N-1);
   dx = 1.0/(double) (J-1);
   nu = dt/(dx*dx);

// compute the initial solution
```

```
   for (j = 0; j < J; ++j) {
      x = j*dx;
      u[j] = u0(x);
   }

// main loop -- integrate the PDE

   for (n = 0; n < N-1; ++n) {
      // for bcs
      u1[0] = b0(t+dt);
      u1[J-1] = b1(t+dt);
      // for interior points
      for (j = 1; j < J-1; ++j) {
         u1[j] = u[j] + nu*(u[j-1]-2.0*u[j]+u[j+1]);
      }
      // update time
      t = t + dt;
      // update the solution
      for (j = 0; j < J; ++j) u[j] = u1[j];
   }

// output the solution at t = T and other information

   fprintf(out, "\n# t = %10.3e\n", t);
   for (j = 0; j < J; ++j) {
      x = j*dx;
      fprintf(out, "%10.3e %10.3e\n", x, u[j]);
   }

   printf(" **** (J, N) = (%4d, %4d)  nu = %lf\n", J, N, nu);

return 0;
}

// define functions

double u0(double x)
{
double u, pi = acos(-1.0);
   u = sin(pi*x);
   return u;
}

double b0(double t)
{
double u;
   u = 0.0;
   return u;
}

double b1(double t)
```

```
{
double u;
   u = 0.0;
   return u;
}
```