

Oct 26, 08 1:41

dftcs.cpp

Page 1/2

```

// dftcs - Program to solve the diffusion equation
// using the Forward Time Centered Space (FTCS) scheme.
#include "NumMeth.h"
using namespace std;
int main() {

    /* Initialize parameters (time step, grid spacing, etc.).
    cout << "Enter time step: "; double tau; cin >> tau;
    cout << "Enter the number of grid points: "; int N; cin >> N;
    double L = 1.; // The system extends from x=-L/2 to x=L/2
    double h = L/(N-1); // Grid size
    double kappa = 1.; // Diffusion coefficient
    double coeff = kappa*tau/(h*h);
    if( coeff < 0.5 )
        cout << "Solution is expected to be stable" << endl;
    else
        cout << "WARNING: Solution is expected to be unstable" << endl;

    /* Set initial and boundary conditions.
    Matrix tt(N), tt_new(N);
    tt.set(0.0); // Initialize temperature to zero at all points
    tt(N/2) = 1/h; // Initial cond. is delta function in center
    /// The boundary conditions are tt(1) = tt(N) = 0
    tt_new.set(0.0); // End points are unchanged during iteration

    /* Set up loop and plot variables.
    int iplot = 1; // Counter used to count plots
    int nStep = 300; // Maximum number of iterations
    int plot_step = 6; // Number of time steps between plots
    int nplots = nStep/plot_step + 1; // Number of snapshots (plots)
    Matrix xplot(N), tplot(nplots), ttplot(N,nplots);
    int i,j;
    for( i=1; i<=N; i++ )
        xplot(i) = (i-1)*h - L/2; // Record the x scale for plots

    /* Loop over the desired number of time steps.
    int iStep;
    for( iStep=1; iStep<=nStep; iStep++ ) {

        /* Compute new temperature using FTCS scheme.
        for( i=2; i<=(N-1); i++ )
            tt_new(i) = tt(i) + coeff*(tt(i+1) + tt(i-1) - 2*tt(i));

        tt = tt_new; // Reset temperature to new values

        /* Periodically record temperature for plotting.
        if( (iStep%plot_step) < 1 ) { // Every plot_step steps
            for( i=1; i<=N; i++ ) // record tt(i) for plotting
                ttplot(i,iplot) = tt(i);
            tplot(iplot) = iStep*tau; // Record time for plots
            iplot++;
        }
    }
    nplots = iplot-1; // Number of plots actually recorded

    /* Print out the plotting variables: tplot, xplot, ttplot
    ofstream ttplotOut("dftcs.txt");
    for( i=1; i<=N; i++ ) {
        for( j=1; j<=nplots; j++ )

```

Oct 26, 08 1:41

dftcs.cpp

Page 2/2

```
        ttplotOut <<xplot(i)<<" "<<tplot(j) <<" "<< ttplot(i,j) << endl;
    ttplotOut << endl;
}
}
/***** To plot in gnuplot *****/
splot "dftcs.txt" with lines
*****/
```