Table of Contents


```
% dftcs - Program to solve the diffusion equation
% using the Forward Time Centered Space (FTCS) scheme.
```

initialize parameters

```
%* Initialize parameters (time step, grid spacing, etc.).
tau = 1e-4;
N = 61;
L = 1.;  % The system extends from x=-L/2 to x=L/2
h = L/(N-1);  % Grid size
kappa = 1.;   % Diffusion coefficient
coeff = kappa*tau/h^2;
if( coeff < 0.5 )
  disp('Solution is expected to be stable');
else
  disp('WARNING: Solution is expected to be unstable');
end

%* Set initial and boundary conditions.
tt = zeros(N,1);          % Initialize temperature to zero at all
 points
tt(round(N/2)) = 1/h;     % Initial cond. is delta function in center
% The boundary conditions are tt(1) = tt(N) = 0
```

```
Solution is expected to be stable
```

numerical solution

```
%* Set up loop and plot variables.
xplot = (0:N-1)*h - L/2;   % Record the x scale for plots
iplot = 1;                 % Counter used to count plots
nstep = 150;               % Maximum number of iterations
nplots = 50;               % Number of snapshots (plots) to take
plot_step = nstep/nplots;  % Number of time steps between plots

%* Loop over the desired number of time steps.
for istep=1:nstep  %% MAIN LOOP %%

  %* Compute new temperature using FTCS scheme.
```

```
  tt(2:(N-1)) = tt(2:(N-1)) + ...
      coeff*(tt(3:N) + tt(1:(N-2)) - 2*tt(2:(N-1)));

  % conform to boundary conditions
  tt(1) = tt(2);
  tt(N) = tt(N-1);

  %* Periodically record temperature for plotting.
  if( rem(istep,plot_step) < 1 )   % Every plot_step steps
    ttplot(:,iplot) = tt(:);       % record tt(i) for plotting
    tplot(iplot) = istep*tau;      % Record time for plots
    iplot = iplot+1;
  end
end
```

moi sol'n at t = 0.015

```
x0 = 0;
t=[1e-8 0.001:0.001:1];
x = linspace(-0.5*L, 0.5*L, length(xplot));
sig=sqrt(2*kappa*t);
TG=zeros(length(x),length(t));
nimag = 30;
Timag=zeros([size(TG) nimag+1]);
Timag(:,:,1)=(ones(length(x),1)*(1./sig)).*exp(-x'.^2*(1./sig.^2)/2)/
sqrt(2*pi);
for ii=1:nimag
Timag(:,:,ii+1)=Timag(:,:,ii)+(ones(length(x),1)*(1./sig)).*exp(-(x
+(ii*L-(-1)^ii*x0))'.^2*(1./sig.^2)/2)/sqrt(2*pi)+...
    (ones(length(x),1)*(1./sig)).*exp(-(x-(ii*L+(-1)^ii*x0))'.^2*(1./
sig.^2)/2)/sqrt(2*pi);
end
```

find numerical solution at t = 0.015

query vector of times to find which index contains value

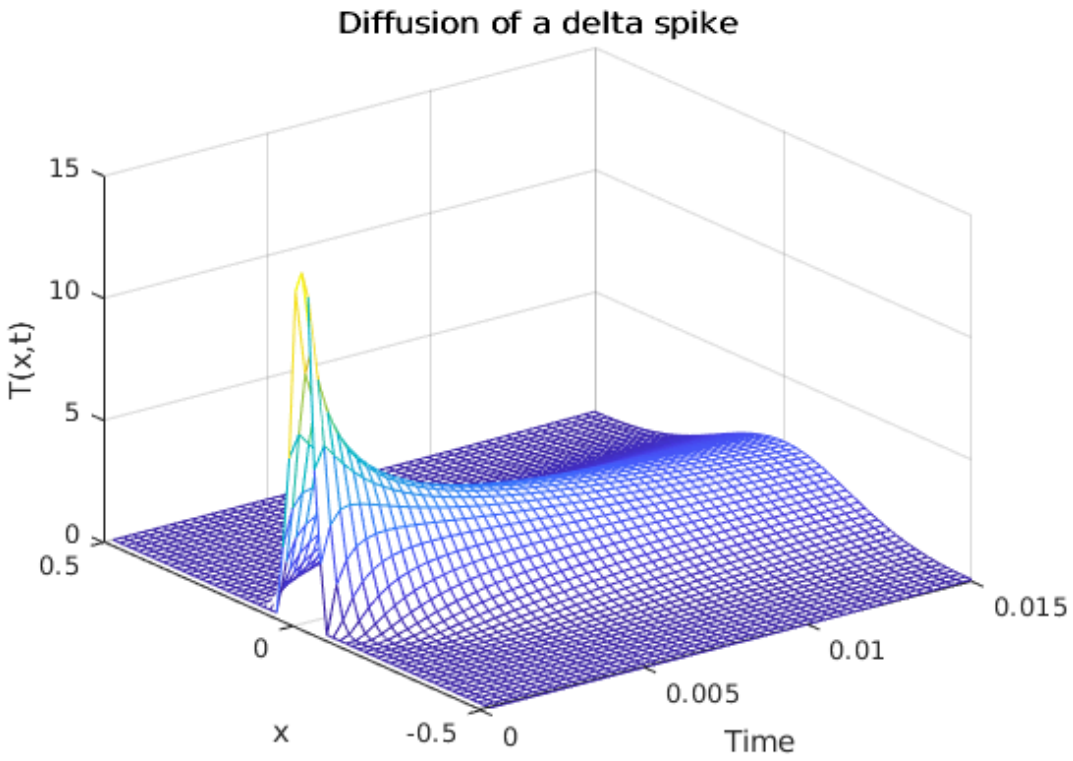
```
tplot(25)
idx = find(abs(tplot - 0.015) <= 1e-5);
tni = ttplot(:, idx);
```

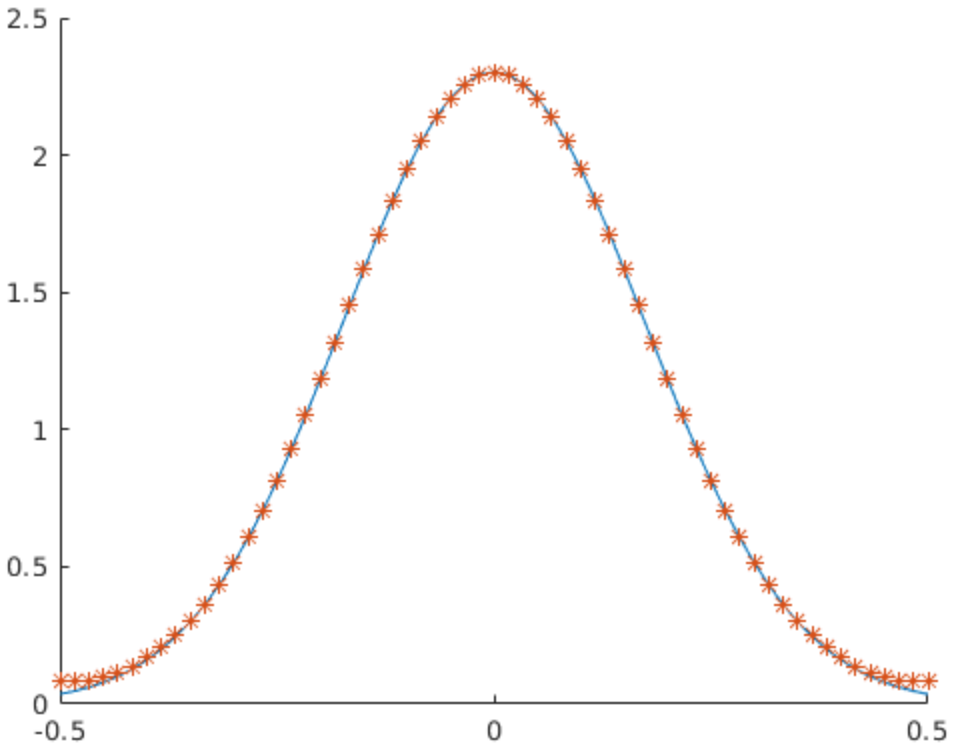
```
ans =

    0.0075
```

make plots

```
%* Plot temperature versus x and t as wire-mesh and contour plots.
figure(1); clf;
```

```
mesh(tplot,xplot,ttplot);  % Wire-mesh surface plot
xlabel('Time');  ylabel('x');  zlabel('T(x,t)');
title('Diffusion of a delta spike');
pause(1);

figure(2); clf;
hold on
plot(xplot, Timag(:,16,1), '-')
plot(xplot, tni, '*')
hold off
```





Published with MATLAB® R2019a