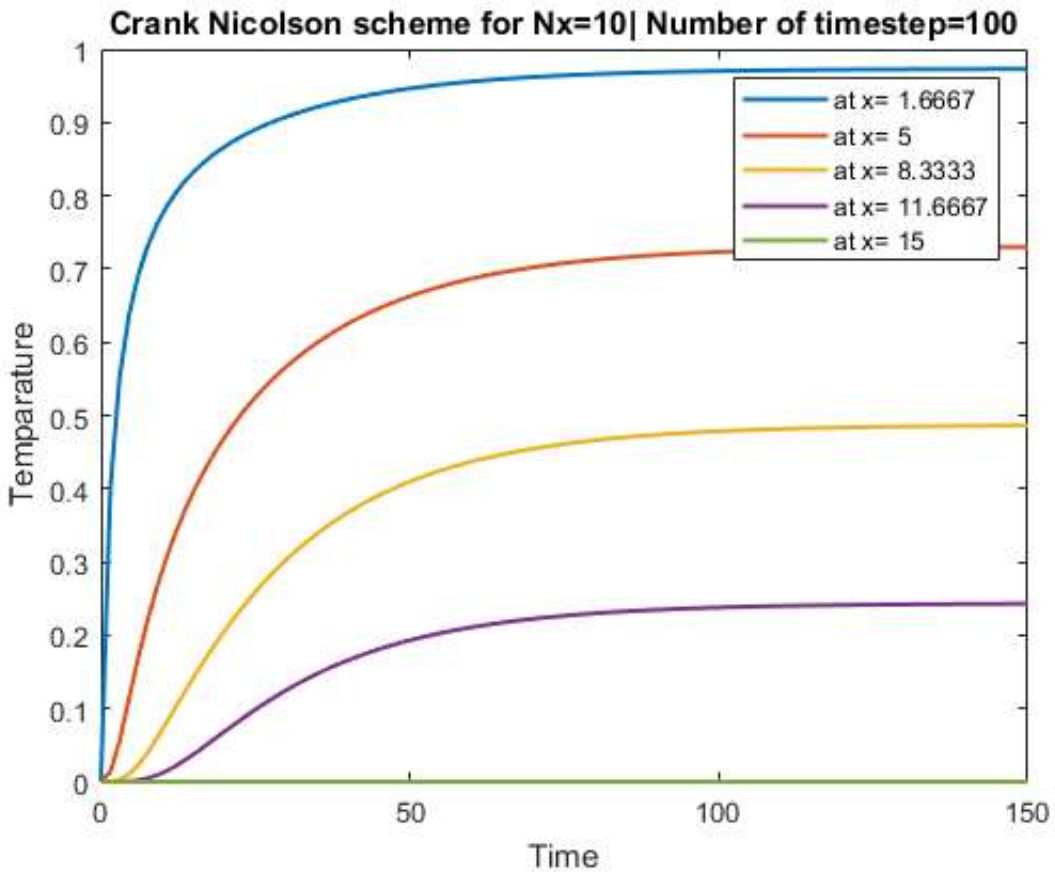
Contents

Crack Nicolson Method for Nx=10

```
clc; clear all
% Setup
close all;
N = 10;
L = 15;
alpha = 1;
dx = L/(N-1);
X = linspace(0,L,N)';

% initial condition
T = 0*X;

% Spatial derivative operator
A = gallery('tridiag',N-2,1,-2,1);

% inhomogeneous term
f = @(t) (-(X(2:N-1).^2-4*X(2:N-1).^2+2).*exp(-X(2:N-1).^2));

% Time advancement
% Must solve the system
%
% $$\left(I-\frac{\alpha dt}{2dx^2}A\right) T^{n+1} = \left(I+\frac{\alpha dt}{2dx^2}A\right)
 T^{n} + \frac{dt}{2}\left(f(t^{n+1})+f(t^n)\right)$$
%

g = @(t,dt,T) (speye(N-2)-alpha*dt/2/(dx^2)*A)\(T(2:N-1)+ ...
               alpha*dt/2/(dx^2)*A*T(2:N-1) + dt*(f(t)+f(t+dt))/2);

% Stable run
Nt = 100;            % time step
t_final = 150;         % final time
time = linspace(0,t_final,Nt);  % time array
dt=time(2)-time(1);
pt = time;   % desired plot times
pn = length(time);           % number of desired plots
pc = 1;                % plot counter
rt = zeros(1,pn);
T_s = zeros(N,pn);       % solution storage


for t = time
    % plot storage
    if ( t >= pt(pc) )
        T_s(:,pc) = T;
        rt(pc) = t;
        pc = pc + 1;
```

```
        if (pc > pn)
            break
        end

    end

    % time advancement
    T(2:N-1) = g(t,dt,T);
end

% Plot stable run
for m=2:2:length(X)
    txt=['at x= ',num2str(X(m))];
    plot(time,T_s(m,:),'LineWidth',1.5,'DisplayName',txt);
    hold on
end
xlabel('Time');
ylabel('Temparature')
legend show
title('Crank Nicolson scheme for Nx=10| Number of timestep=100')
```

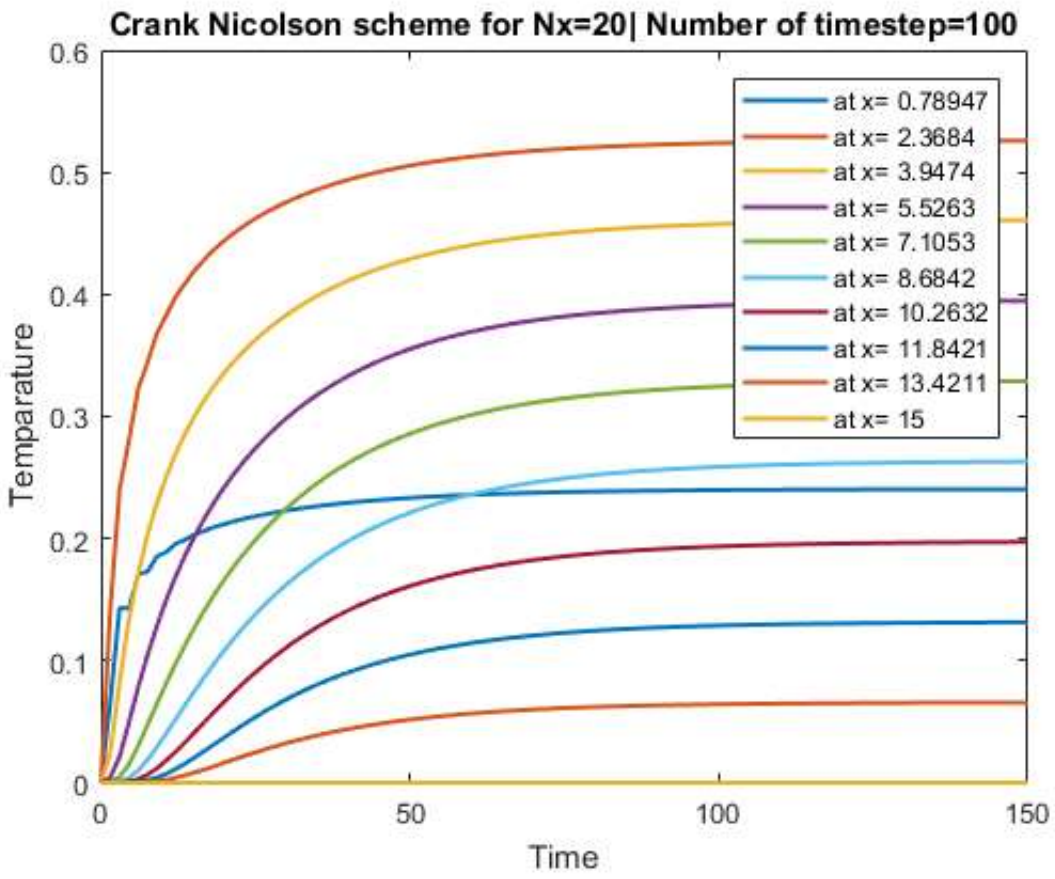


Crack Nicolson Method for Nx=20

```
clc; clear all
% Setup
close all;
N = 20;
L = 15;
```

```
alpha = 1;
dx = L/(N-1);
X = linspace(0,L,N)';

% initial condition
T = 0*X;

% Spatial derivative operator
A = gallery('tridiag',N-2,1,-2,1);

% inhomogeneous term
f = @(t) (-(X(2:N-1).^2-4*X(2:N-1).^2+2).*exp(-X(2:N-1).^2));

% Time advancement
% Must solve the system
%
% $$\left(I-\frac{\alpha dt}{2dx^2}A\right) T^{n+1} = \left(I+\frac{\alpha dt}{2dx^2}A\right)
 T^{n} + \frac{dt}{2}\left(f(t^{n+1})+f(t^n)\right)$$
%

g = @(t,dt,T) (speye(N-2)-alpha*dt/2/(dx^2)*A)\(T(2:N-1)+ ...
              alpha*dt/2/(dx^2)*A*T(2:N-1) + dt*(f(t)+f(t+dt))/2);

% Stable run
Nt = 100;           % time step
t_final = 150;        % final time
time = linspace(0,t_final,Nt);  % time array
dt=time(2)-time(1);
pt = time;   % desired plot times
pn = length(time);          % number of desired plots
pc = 1;               % plot counter
rt = zeros(1,pn);
T_s = zeros(N,pn);      % solution storage


for t = time
    % plot storage
    if ( t >= pt(pc) )
        T_s(:,pc) = T;
        rt(pc) = t;
        pc = pc + 1;
        if (pc > pn)
            break
        end

    end

    % time advancement
    T(2:N-1) = g(t,dt,T);
end

% Plot stable run
for m=2:2:length(X)
    txt=['at x= ',num2str(X(m))];
    plot(time,T_s(m,:),'LineWidth',1.5,'DisplayName',txt);
    hold on
end
```

```
xlabel('Time');
ylabel('Temparature')
legend show
title('Crank Nicolson scheme for Nx=20| Number of timestep=100')
```



Published with MATLAB® R2017a