CAAM 336 · DIFFERENTIAL EQUATIONS

Homework 50 · Solutions

Posted Wednesday 16 April 2014. Due 1pm Friday 25 April 2014.

50. [25 points]

Let $H_D^1(0,1) = \{v \in H^1(0,1) : v(0) = 0\}$. Let N be a positive integer, let $h = \dfrac{1}{N+1}$ and let $x_k = kh$ for $k = 0, 1, \ldots, N+1$. Let the continuous piecewise linear hat functions $\phi_j \in H_D^1(0,1)$ be such that

$$\phi_j(x) = \begin{cases} \dfrac{x - x_{j-1}}{h} & \text{if } x \in [x_{j-1}, x_j), \\ \dfrac{x_{j+1} - x}{h} & \text{if } x \in [x_j, x_{j+1}), \\ 0 & \text{otherwise,} \end{cases}$$

for $j = 1, \ldots, N$ and

$$\phi_{N+1}(x) = \begin{cases} \dfrac{x - x_N}{h} & \text{if } x \in [x_N, x_{N+1}], \\ 0 & \text{otherwise.} \end{cases}$$

Let the timestep $\Delta t \in \mathbb{R}$ be such that $\Delta t > 0$ and let $t_k = k\Delta t$ for $k = 0, 1, 2, \ldots$. Let $V_N = \operatorname{span}\{\phi_1, \ldots \phi_{N+1}\}$, let $u_0 \in H_D^1(0,1)$ be such that

$$u_0(x) = \begin{cases} 0 & \text{if } x \in [0, 1/4], \\ 4x - 1 & \text{if } x \in (1/4, 1/2], \\ 3 - 4x & \text{if } x \in (1/2, 3/4], \\ 0 & \text{if } x \in (3/4, 1], \end{cases}$$

and let

$$u_{0,N}(x) = \sum_{j=1}^{N+1} u_0(x_j)\phi_j(x).$$

Let $u(x,t)$ be the solution of

$$\begin{aligned} \frac{\partial u}{\partial t}(x,t) - \frac{\partial}{\partial x}\left((\sin(\pi x) + 1)\frac{\partial u}{\partial x}(x,t)\right) &= 0, \quad 0 < x < 1, \quad t > 0, \\ u(0,t) &= 0, \quad t \geq 0, \\ \frac{\partial u}{\partial x}(1,t) &= 0, \quad t \geq 0, \\ u(x,0) &= u_0(x), \quad 0 < x < 1. \end{aligned}$$

(a) We can obtain finite element approximations to u by finding u_N such that

$$u_N(x,t) = \sum_{j=1}^{N+1} \alpha_j(t)\phi_j(x)$$

where the $\alpha_j(t)$ are such that

$$\int_0^1 \frac{\partial u_N}{\partial t}(x,t)v(x)\,dx + \int_0^1 (\sin(\pi x) + 1)\frac{\partial u_N}{\partial x}(x,t)v'(x)\,dx = 0 \text{ for all } v \in V_N$$

and

$$u_N(x,0) = u_{0,N}(x), \quad 0 < x < 1.$$

Let

$$\boldsymbol{\alpha}(t) = \begin{bmatrix} \alpha_1(t) \\ \alpha_2(t) \\ \vdots \\ \alpha_{N+1}(t) \end{bmatrix}.$$

What system of ordinary differential equations and initial condition does $\boldsymbol{\alpha}(t)$ satisfy?

(b) For the remainder of the question, your answers and codes should not feature the inverse of any matrices. What system of equations has to be solved in order to use the forward Euler method to compute an approximation $\boldsymbol{\alpha}^{(k)}$ to $\boldsymbol{\alpha}(t_k)$ for $k = 1, 2, 3, \ldots$?

(c) Use the forward Euler method with a timestep of $\Delta t = 10^{-5}$ to obtain approximations to $u_N(x,t)$ (and hence $u(x,t)$) at $t = 0.001, 0.01, 0.1, 0.2$ for $N = 15, 31$. Produce a plot showing these approximations but use a different figure for each value of N. Also plot $u_N(x,0)$ on both of your figures.

(d) Repeat part (c) but take the timestep $\Delta t = 10^{-4}$.

(e) What system of equations has to be solved in order to use the backward Euler method to compute an approximation $\boldsymbol{\alpha}^{(k)}$ to $\boldsymbol{\alpha}(t_k)$ for $k = 1, 2, 3, \ldots$?

(f) Use the backward Euler method with a timestep of $\Delta t = 10^{-3}$ to obtain approximations to $u_N(x,t)$ (and hence $u(x,t)$) at $t = 0.001, 0.01, 0.1, 0.2$ for $N = 15, 31$. Produce a plot showing these approximations but use a different figure for each value of N. Also plot $u_N(x,0)$ on both of your figures.

Solution.

(a) [3 points] Let $\boldsymbol{\alpha}^{(0)} \in \mathbb{R}^{N+1}$ be the vector with entries $\alpha_j^{(0)} = u_0(x_j)$, let the mass matrix $\mathbf{M} \in \mathbb{R}^{(N+1)\times(N+1)}$ be the matrix with entries

$$M_{jk} = \int_0^1 \phi_k(x)\phi_j(x)\,dx$$

and let the stiffness matrix $\mathbf{K} \in \mathbb{R}^{(N+1)\times(N+1)}$ be the matrix with entries

$$K_{jk} = \int_0^1 (1 + \sin(\pi x))\phi_k'(x)\phi_j'(x)\,dx.$$

Then $\boldsymbol{\alpha}(t)$ is the solution to the system of ordinary differential equations

$$\mathbf{M}\boldsymbol{\alpha}'(t) = -\mathbf{K}\boldsymbol{\alpha}(t)$$

with initial condition

$$\boldsymbol{\alpha}(0) = \boldsymbol{\alpha}^{(0)}.$$

(b) [3 points] For $k = 1, 2, 3, \ldots$, we can use the forward Euler method to obtain approximations $\boldsymbol{\alpha}^{(k)}$ to $\boldsymbol{\alpha}(t_k)$. We can compute these approximations as

$$\boldsymbol{\alpha}^{(k)} = \boldsymbol{\alpha}^{(k-1)} + \Delta t \boldsymbol{\beta}^{(k-1)}$$

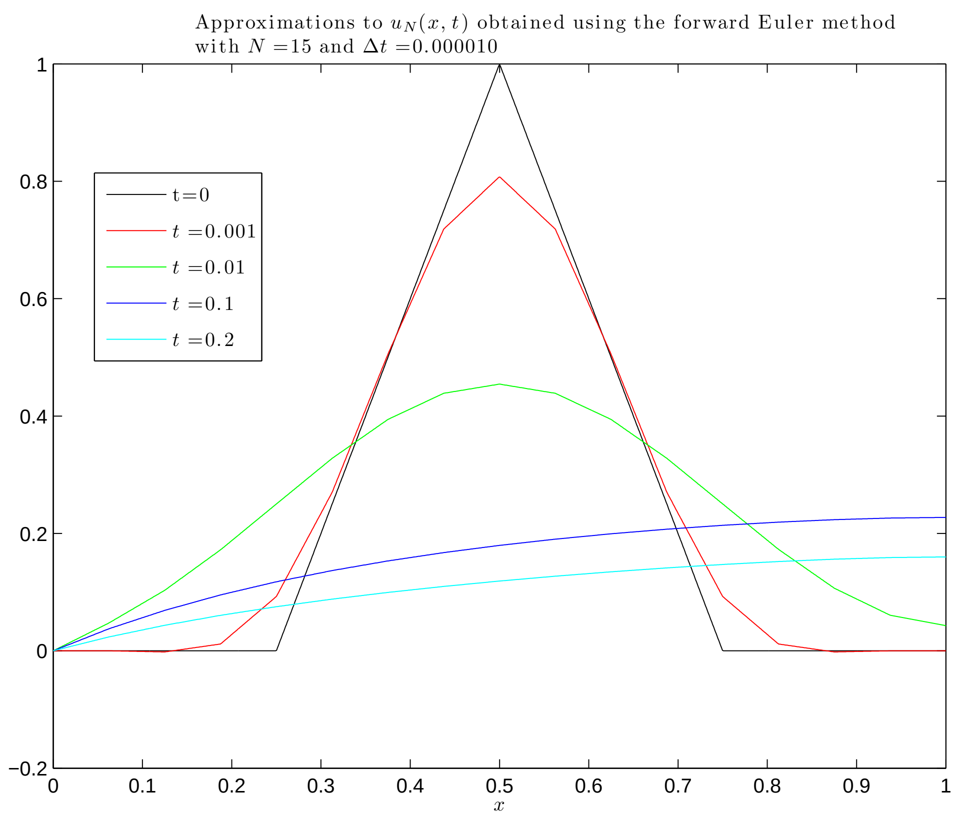
where $\boldsymbol{\beta}^{(k-1)}$ is the solution of

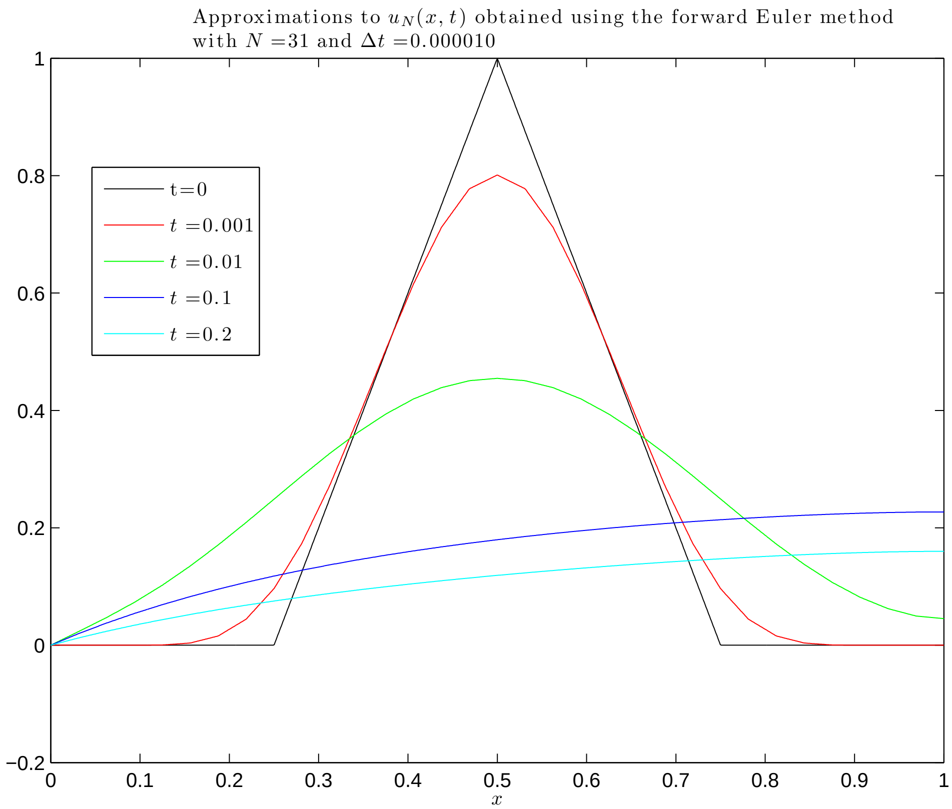
$$\mathbf{M}\boldsymbol{\beta}^{(k-1)} = -\mathbf{K}\boldsymbol{\alpha}^{(k-1)}.$$

It is also acceptable to say that we can compute $\boldsymbol{\alpha}^{(k)}$ by solving

$$\mathbf{M}\boldsymbol{\alpha}^{(k)} = (\mathbf{M} - \Delta t \mathbf{K})\, \boldsymbol{\alpha}^{(k-1)}.$$

(c) [6 points] The requested plots are below.





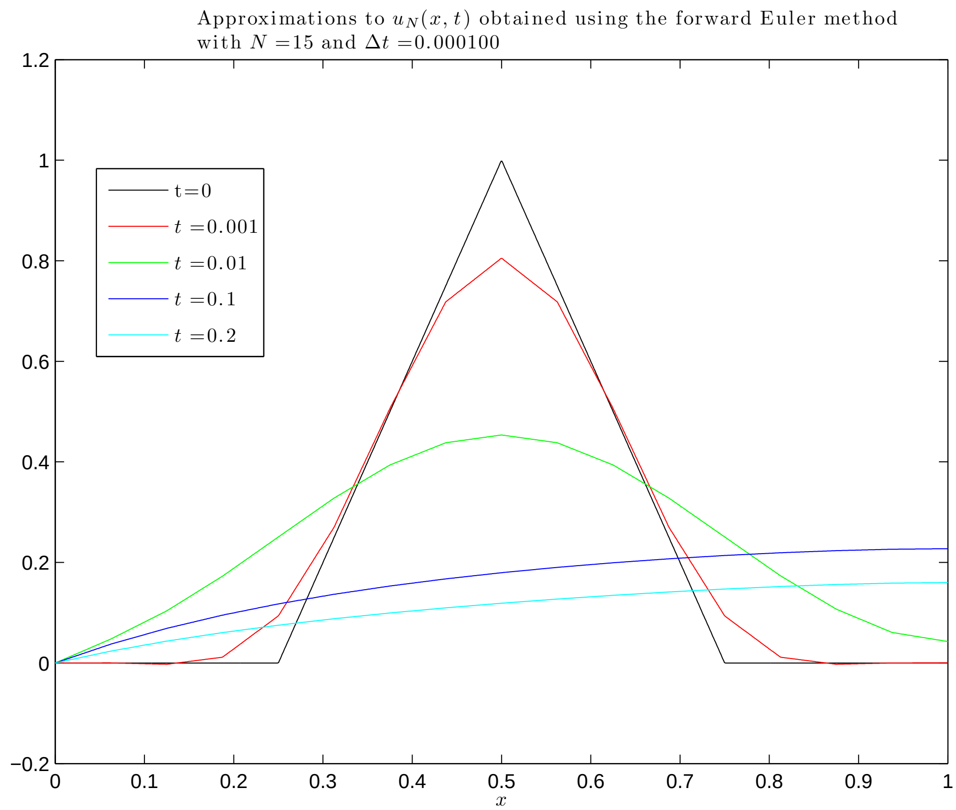
The code used to produce the results shown in this part and in parts (d) and (f) is below. Note that the below code uses the MATLAB functions which you had to write in Homework 2 and Homework 49.

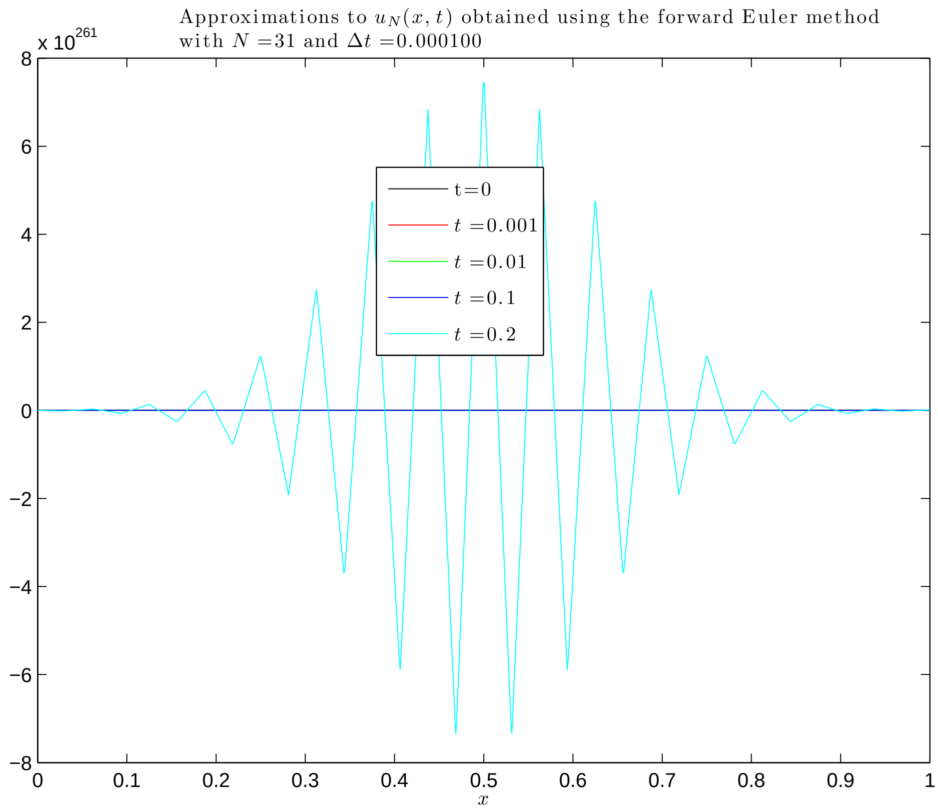
```
clear
clc

Nvec = [15 31];
DeltatvecFE=[1e-5 1e-4];
plotvalsFE=[100 1000 10000 20000];
plotvalsFE=[plotvalsFE; plotvalsFE/10];
DeltatvecBE=[1e-3];
plotvalsBE=[1 10 100 200];
colors='rgbc';
legendstr{1}='t=0';
figcount=1;

for j=1:length(Nvec)
   N = Nvec(j);
   h = 1/(N+1);

   K=sparse(N+1,N+1);
   K=K+sparse(1:N,2:N+1,-1/h-(cos(pi*(1:N)*h)-cos(pi*(2:N+1)*h))/(pi*h^2),N+1,N+1);
   K=K+K.';
   K=K+sparse(1:N,1:N,2/h+(cos(pi*(0:N-1)*h)-cos(pi*(2:N+1)*h))/(pi*h^2),N+1,N+1);
   K=K+sparse(N+1,N+1,1/h+(cos(pi*N*h)-cos(pi*(N+1)*h))/(pi*h^2),N+1,N+1);

   M=sparse(N+1,N+1);
   M=M+sparse(1:N,2:N+1,h/6,N+1,N+1);
   M=M+M.';
   M=M+sparse(1:N,1:N,2*h/3,N+1,N+1);
   M=M+sparse(N+1,N+1,h/3,N+1,N+1);
```

```
x=linspace(0,1,1000)';

for jj=1:length(DeltatvecFE)
    Deltat=DeltatvecFE(jj);
    plotcount=1;
    alpha=initialinterpolant((1:N+1)*h,N).';
    figure(figcount)
    clf
    uN = zeros(size(x));
    for l=1:N+1
        uN=uN+alpha(l)*hat(x,l,N);
    end
    plot(x,uN,'k')
    hold on
    for k=1:max(plotvalsFE(jj,:))
        s=M\(-K*alpha);
        alpha=alpha+Deltat*s;
        if k==plotvalsFE(jj,plotcount);
            uN = zeros(size(x));
            for l=1:N+1
                uN=uN+alpha(l)*hat(x,l,N);
            end
            plot(x, uN, colors(plotcount))
            plotcount=plotcount+1;
        end
    end
    for k=1:length(plotvalsFE(jj,:))
        legendstr{k+1}=['$t=$' num2str(plotvalsFE(jj,k)*Deltat)];
    end
    legend(legendstr,'interpreter','latex','location','best')
    xlabel('$x$','interpreter','latex')
    titlestr{1}='Approximations to $u_N(x,t)$ obtained using the forward Euler method
        ';
    titlestr{2}=['with $N=$' num2str(N) ' and $\Delta t=$' num2str(Deltat,'%f')];
    title({titlestr{1};titlestr{2}},'interpreter','latex')
    figcount=figcount+1;
end

for jj=1:length(DeltatvecBE)
    Deltat=DeltatvecBE(jj);
    plotcount=1;
    alpha=initialinterpolant((1:N+1)*h,N).';
    figure(figcount)
    clf
    uN = zeros(size(x));
    for l=1:N+1
        uN=uN+alpha(l)*hat(x,l,N);
    end
    plot(x,uN,'k')
    hold on
    for k=1:max(plotvalsBE(jj,:))
        alpha=(M+Deltat*K)\(M*alpha);
        if k==plotvalsBE(jj,plotcount);
            uN = zeros(size(x));
            for l=1:N+1
                uN=uN+alpha(l)*hat(x,l,N);
            end
            plot(x, uN, colors(plotcount))
            plotcount=plotcount+1;
        end
    end
    for k=1:length(plotvalsBE(jj,:))
        legendstr{k+1}=['$t=$' num2str(plotvalsBE(jj,k)*Deltat)];
    end
    legend(legendstr,'interpreter','latex','location','best')
    xlabel('$x$','interpreter','latex')
    titlestr{1}='Approximations to $u_N(x,t)$ obtained using the backward Euler
```

```
            method';
        titlestr{2}=['with $N=$' num2str(N) ' and $\Delta t=$' num2str(Deltat,'%f')];
        title({titlestr{1};titlestr{2}},'interpreter','latex')
        figcount=figcount+1;
    end

end
saveas(figure(1),'hw50c15.eps','epsc')
saveas(figure(2),'hw50d15.eps','epsc')
saveas(figure(3),'hw50f15.eps','epsc')
saveas(figure(4),'hw50c31.eps','epsc')
saveas(figure(5),'hw50d31.eps','epsc')
saveas(figure(6),'hw50f31.eps','epsc')
```

(d) [4 points] The requested plots are below.

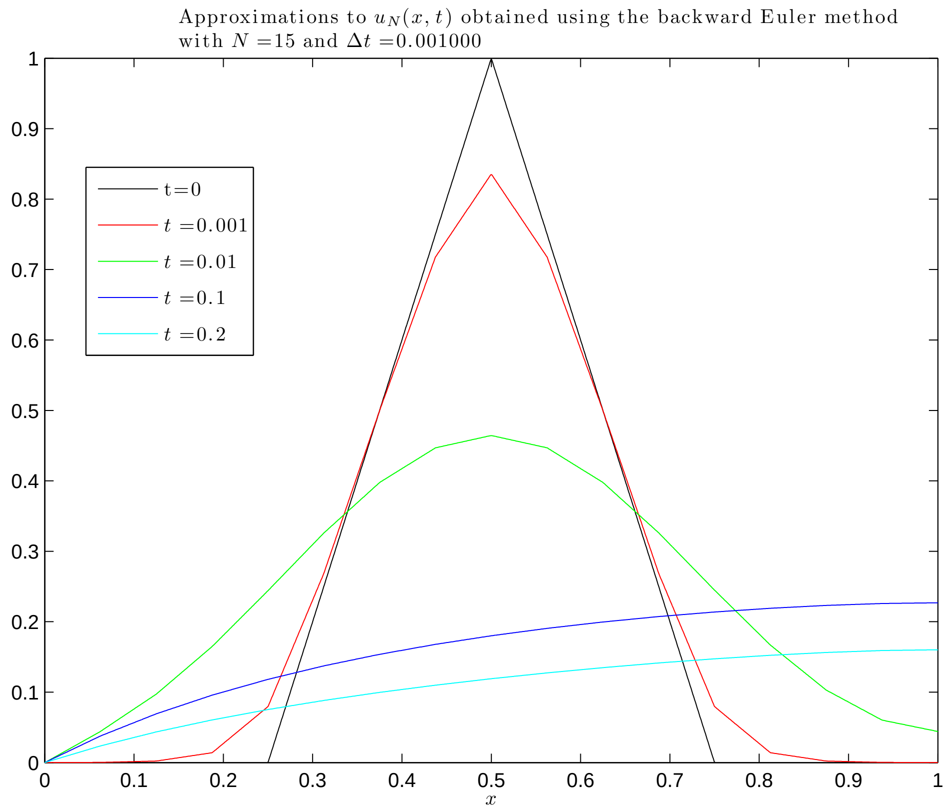


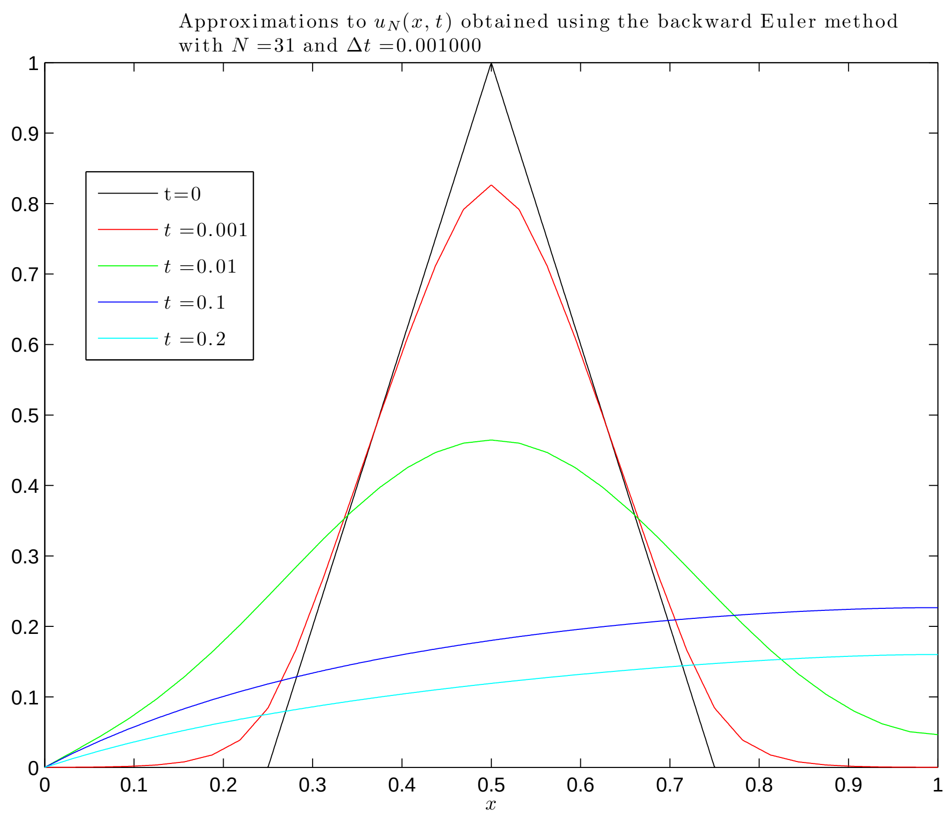


(e) [3 points] For $k = 1, 2, 3, \ldots$, we can use the backward Euler method to obtain approximations $\boldsymbol{\alpha}^{(k)}$ to $\boldsymbol{\alpha}(t_k)$. We can compute these approximations by solving

$$(\mathbf{M} + \Delta t \mathbf{K})\, \boldsymbol{\alpha}^{(k)} = \mathbf{M}\boldsymbol{\alpha}^{(k-1)}.$$

(f) [6 points] The requested plots are below.


Approximations to $u_N(x,t)$ obtained using the backward Euler method
with N =15 and Δt =0.001000
t=0
t =0.001
t =0.01
t =0.1
t =0.2
x


Approximations to $u_N(x,t)$ obtained using the backward Euler method
with N =31 and Δt =0.001000
t=0
t =0.001
t =0.01
t =0.1
t =0.2
x