Example- Second-order ODE with step input

This example is meant to illustrate why using a backward-looking Euler approximation to derivatives is better (i.e., stable) than using a forward-looking Euler approximation (can be unstable) to derivatives [e.g., equations (2.69) and (2.71) of the text].

First, consider the general second-order ODE

$$\frac{d^2y(t)}{dt^2} + (s_1 + s_2)\frac{dy(t)}{dt} + s_1 s_2\, y(t) = s_1 s_2\, x(t)$$

where

$$y(0) = 0,\ \left.\frac{dy(t)}{dt}\right|_{t=0} = 0, \text{ and } \ x(t) = u(t).$$

Analytic Solution

Assume $y(t) = A e^{st}$, then the characteristic equation for the homogeneous or unforced solution

$$\frac{d^2y_u(t)}{dt^2} + (s_1 + s_2)\frac{dy_u(t)}{dt} + s_1 s_2\, y_u(t) = 0,$$

is

$$s^2 + (s_1 + s_2)s + s_1 s_2 = (s + s_1)(s + s_2) = 0$$

with roots $-s_1$ and $-s_2$. This yields the unforced solution

$$y_u(t) = A_1 e^{-s_1 t} + A_2 e^{-s_2 t}.$$

The inhomogeneous or forced solution is $y_f(t) = y_{ss} = y(t \to \infty)$. At steady-state derivatives go to zero and the ODE becomes

$$0 + (s_1 + s_2)0 + s_1 s_2\, y(t \to \infty) = s_1 s_2\, u(t \to \infty) = s_1 s_2 \text{ and } y(t \to \infty) = 1.$$

The total solution is the sum of the unforced and forced solutions

$$y(t) = y_u(t) + y_f(t) = A_1 e^{-s_1 t} + A_2 e^{-s_2 t} + 1.$$

Next, apply the initial conditions to solve for the constants A_1 and A_2.

From $y(0)=0$, we get the equation $y(0)=0=A_1+A_2+1 \quad \Rightarrow \quad A_1+A_2=-1$.

From $\left.\frac{dy(t)}{dt}\right|_{t=0}=0$, we get the equation

$$\left.\frac{dy(t)}{dt}\right|_{t=0}=0=\left(-s_1A_1\,e^{-s_1t}-s_2A_2\,e^{-s_2t}+0\right)\Big|_{t=0} \quad \Rightarrow \quad 0=-s_1A_1-s_2A_2.$$

Solving the two equations for the two unknowns yields

$$A_1=\frac{-s_2}{s_2-s_1} \text{ and } A_2=\frac{s_1}{s_2-s_1}.$$

Therefore, the final analytic solution is

$$\underline{y(t)=\left(\frac{-s_2}{s_2-s_1}\right)e^{-s_1t}+\left(\frac{s_1}{s_2-s_1}\right)e^{-s_2t}+1 \quad t\geq 0.}$$

Forward-looking Euler Numerical Solution (Text method)

Applying equations (2.69) and (2.71) of the text to the general second-order ODE yields the difference equation

$$\frac{y[n+2]-2y[n+1]+y[n]}{T^2}+(s_1+s_2)\frac{y[n+1]-y[n]}{T}+s_1s_2y[n]=s_1s_2x[n]=s_1s_2u[n]$$

which simplifies to

$$y[n+2]+\left[-2+(s_1+s_2)T\right]y[n+1]+\left[1-(s_1+s_2)T+s_1s_2T^2\right]y[n]=s_1s_2T^2u[n].$$

Exploiting time-invariance (i.e., let $n\rightarrow n-2$) yields

$$\underline{y[n]+\left[-2+(s_1+s_2)T\right]y[n-1]+\left[1-(s_1+s_2)T+s_1s_2T^2\right]y[n-2]=s_1s_2T^2u[n-2]}$$

for $n\geq 1$. The initial conditions are $y[0]=y[-1]=x[-1]=0$ and $x[0]=1$.

Backward-looking Euler Numerical Solution

Applying the backward-looking Euler approximations for derivatives

$$\left.\frac{dy(t)}{dt}\right|_{t=nT} = \frac{y[n]-y[n-1]}{T} \text{ and } \left.\frac{d^2y(t)}{dt^2}\right|_{t=nT} = \frac{y[n]-2y[n-1]+y[n-2]}{T^2}$$

to the general second-order ODE, yields the difference equation

$$\frac{y[n]-2y[n-1]+y[n-2]}{T^2}+(s_1+s_2)\frac{y[n]-y[n-1]}{T}+s_1s_2y[n]=s_1s_2u[n].$$

This can be simplified first to

$$\left[1+(s_1+s_2)T+s_1s_2T^2\right]y[n]+\left[-2-(s_1+s_2)T\right]y[n-1]+y[n-2]=s_1s_2T^2u[n].$$

Then, the difference equation can be put in the standard form

$$y[n]+\left(\frac{-2-(s_1+s_2)T}{1+(s_1+s_2)T+s_1s_2T^2}\right)y[n-1]+\left(\frac{1}{1+(s_1+s_2)T+s_1s_2T^2}\right)y[n-2]=$$
$$\left(\frac{s_1s_2T^2}{1+(s_1+s_2)T+s_1s_2T^2}\right)u[n]$$

valid for $n \geq 1$.

The necessary initial conditions for $y[n]$ are

$$y(0) \quad \Rightarrow \quad y[0]=0$$

and

$$\left.\frac{dy(t)}{dt}\right|_{t=0}=0 \quad \Rightarrow \quad \frac{y[0]-y[0-1]}{T}=\frac{0-y[-1]}{T}=0 \quad \Rightarrow \quad y[-1]=0.$$

Since there are no past terms for $x[n]$, it does not require initial conditions.

Specific example- Second-order ODE with step input

When $s_1 = 1$ and $s_2 = 21$ are selected, the second-order ODE is

$$\frac{d^2 y(t)}{dt^2} + 22\frac{dy(t)}{dt} + 21\,y(t) = 21\,x(t)$$

with $y(0) = 0$, $\left.\frac{dy(t)}{dt}\right|_{t=0} = 0$, and $x(t) = u(t)$.

The **analytic** solution is then

$$y(t) = -1.05\,e^{-1t} + 0.05\,e^{-21t} + 1 \quad t \geq 0.$$

With $T = 0.1$ s, the **forward**-looking Euler numerical solution (text method)

$$y[n] + \left[-2 + (s_1 + s_2)T\right] y[n-1] + \left[1 - (s_1 + s_2)T + s_1 s_2 T^2\right] y[n-2] = s_1 s_2 T^2 u[n]$$

becomes

$$y[n] + 0.2\,y[n-1] - 0.99\,y[n-2] = 0.21\,u[n-2]$$

for $n \geq 1$. The initial conditions are $y[0] = y[-1] = x[-1] = 0$ and $x[0] = 1$.

With $T = 0.1$ s, the **backward**-looking Euler numerical solution

$$y[n] + \left(\frac{-2 - (s_1 + s_2)T}{1 + (s_1 + s_2)T + s_1 s_2 T^2}\right) y[n-1] + \left(\frac{1}{1 + (s_1 + s_2)T + s_1 s_2 T^2}\right) y[n-2] =$$

$$\left(\frac{s_1 s_2 T^2}{1 + (s_1 + s_2)T + s_1 s_2 T^2}\right) u[n]$$

becomes

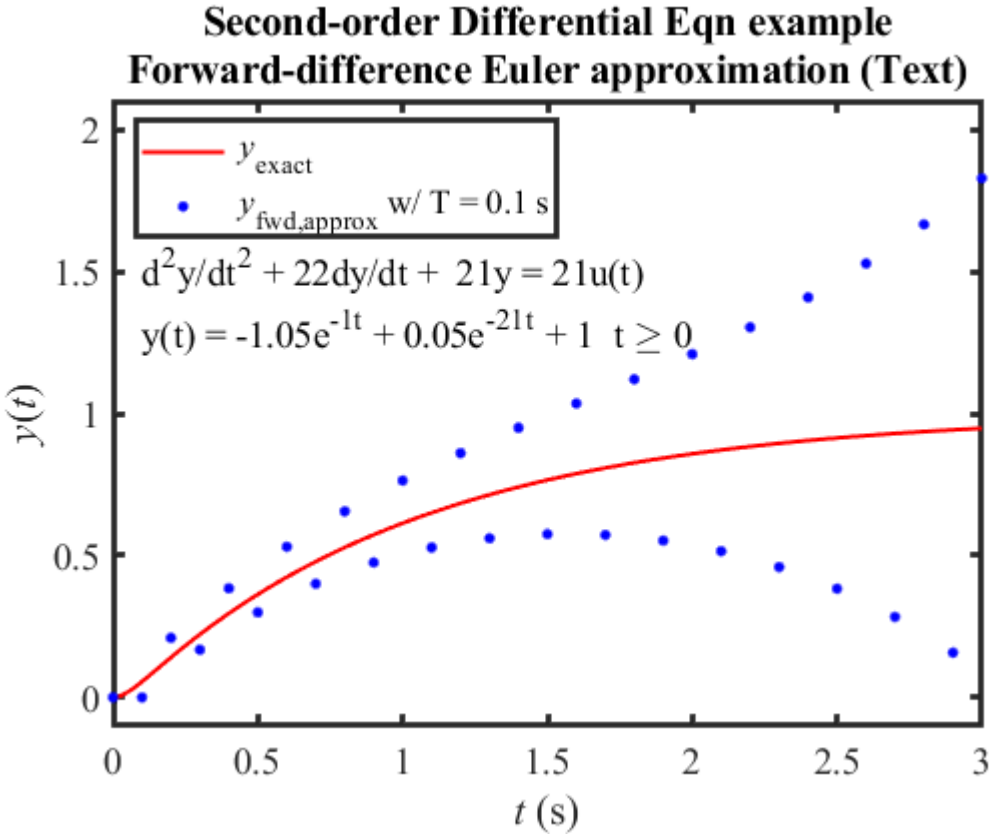
$$y[n] - 1.23167\,y[n-1] + 0.293255\,y[n-2] = 0.0615836\,u[n] \text{ for } n \geq 1.$$

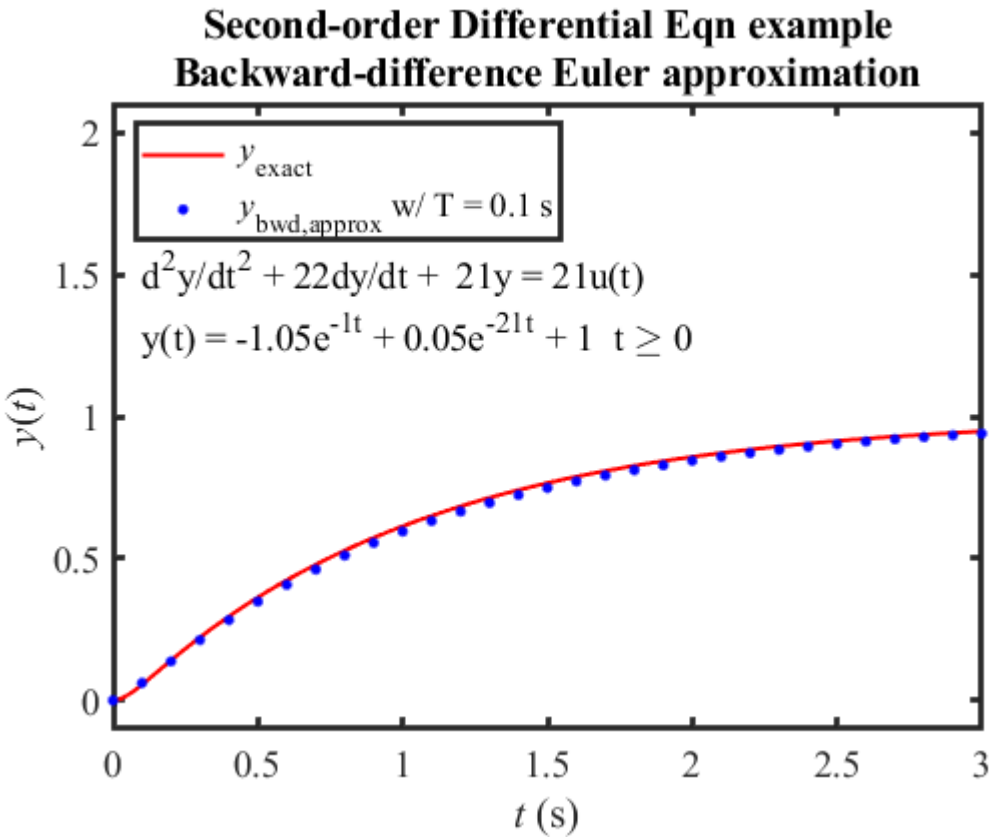
The initial conditions for $y[n]$ are $y[-1] = y[0] = 0$. Since there are no past terms for $x[n]$, it does not require initial conditions.

The corresponding MATLAB m-files to implement these solutions and the corresponding results are attached on the following pages.

```
% Numerical ODE Solution Example (chap2_2ODE_euler_soln_fwd.m)
%
% For a second-order ordinary differential equation (ODE)
%                 d2y/dt2 + (s1+s2)dy/dt + (s1s2) = u(t)/(s1s2)
% where y(0) = dy(0)/dt = 0 which has an analytic solution-
%                 y(t) = A1*e^(-s1*t) + A2*e^(-s2*t) + 1
% where A1 = -s2/(s2-s1) and A2 = s1/(s2-s1),
% find approximate numerical solution by using a forward-difference
% Euler's approximation for derivatives to change it into a
% second-order difference equation which can be solved
% recursively.  Compare numerical results with exact solution.
close all; clear; clc;
% *** define ODE variables ***
s1 = 1; s2 = 21;
tstop = 3*max(1/s1,1/s2); % How far to go in time in seconds
% *** Forward-difference Euler approximation ***
T = 0.1;                    % Time step for numerical approximation
a1 = -2+(s1+s2)*T; a2 = 1-(s1+s2)*T+s1*s2*T*T;
a = [a1, a2];                % a = [a1 a2] coeff. vector
b = [0,0,s1*s2*T*T];         % b = [b0,b1,b2] coeff. vector
n = 1:1:round(tstop/T);      % Define index vector for recur()
x = ones(1,length(n));       % unit step input
x0 = [0,1]; y0 = [0,0];      % initial conditions
y = recur(a,b,n,x,x0,y0);    % yields output for n=1,2,3,...
yapprox = [0,y]; n=[0,n];    % tack on values at t=nT=0
% *** Analytic solution ****
A1 = -s2/(s2-s1);
A2 = s1/(s2-s1);
t = 0:0.01:tstop;       % Define time steps for analytic sol'n
yexact = 1+A1*exp(-s1*t)+A2*exp(-s2*t);
plot(t,yexact,'r',n*T,yapprox,'b.')
legend(' {\ity}_{exact}',[' {\ity}_{fwd,approx} w/ T = ',...
    num2str(T),' s'],'Location','NW'),
axis([0 tstop -0.1 2.1])
ylabel('{\ity}({\itt})','fontsize',16,'fontname','times')
xlabel('{\itt} (s)','fontsize',16,'fontname','times')
title({'Second-order Differential Eqn example';...
    'Forward-difference Euler approximation (Text)'},'fontsize',...
    16,'fontname','times')
text(0.1,1.52,['d^{2}y/dt^2 + ',num2str(s1+s2),'dy/dt +  ',...
    num2str(s1*s2),'y = ',num2str(s1*s2),'u(t)'],'fontsize',...
    14,'fontname','times')
text(0.1,1.3,['y(t) = ',num2str(A1),'e^{-',num2str(s1),'t} + ',...
    num2str(A2),'e^{-',num2str(s2),'t} + 1  t \geq 0'],...
    'fontsize',14,'fontname','times')
set(findobj('type','line'),'linewidth',1.5,'markersize',12)
set(findobj('type','axes'),'linewidth',2,'fontsize',14,'fontname','times')
```

```
% Numerical ODE Solution Example (chap2_2ODE_euler_soln_bwd.m)
%
% For a second-order ordinary differential equation (ODE)
%                 d2y/dt2 + (s1+s2)dy/dt + (s1s2) = u(t)/(s1s2)
% where y(0) = dy(0)/dt = 0 which has an analytic solution-
%                 y(t) = A1*e^(-s1*t) + A2*e^(-s2*t) + 1
% where A1 = -s2/(s2-s1) and A2 = s1/(s2-s1),
% find approximate numerical solution by using a backward-difference
% Euler's approximation for derivatives to change it into a
% second-order difference equation which can be solved
% recursively.  Compare numerical results with exact solution.
close all; clear; clc;
% *** define ODE variables ***
s1 = 1; s2 = 21;
tstop = 3*max(1/s1,1/s2); % How far to go in time in seconds
% *** Backward-difference Euler approximation ***
T = 0.1;                  % Time step for numerical approximation
a1 = (-2-(s1+s2)*T)/(1 + (s1+s2)*T + s1*s2*T*T);
a2 = 1/(1 + (s1+s2)*T + s1*s2*T*T);
a = [a1, a2];                % coefficient vector for y[] terms
b = [(s1*s2*T*T)/(1 + (s1+s2)*T + s1*s2*T*T)]; % [b0] coeff. for x[n]
n = 1:1:round(tstop/T);      % Define index vector for recur()
x = ones(1,length(n));       % unit step input
x0=[]; y0=[0,0];             % initial conditions
y = recur(a,b,n,x,x0,y0);    % yields output for n=1,2,3,...
yapprox = [0,y]; n = [0,n];  % tack on values at t=nT=0
% *** Analytic solution ****
A1 = -s2/(s2-s1);  A2 = s1/(s2-s1);
t = 0:0.01:tstop;                % Define time steps for analytic sol'n
yexact = 1+A1*exp(-s1*t)+A2*exp(-s2*t);
plot(t,yexact,'r',n*T,yapprox,'b.')
legend(' {\ity}_{exact}',[' {\ity}_{bwd,approx} w/ T = ',...
    num2str(T),' s'],'Location','NW'),
axis([0 tstop -0.1 2.1])
ylabel('{\ity}({\itt})','fontsize',16,'fontname','times')
xlabel('{\itt} (s)','fontsize',16,'fontname','times')
title({'Second-order Differential Eqn example';...
    'Backward-difference Euler approximation'},'fontsize',...
    16,'fontname','times')
text(0.1,1.52,['d^{2}y/dt^2 + ',num2str(s1+s2),'dy/dt +  ',...
    num2str(s1*s2),'y = ',num2str(s1*s2),'u(t)'],'fontsize',...
    14,'fontname','times')
text(0.1,1.3,['y(t) = ',num2str(A1),'e^{-',num2str(s1),'t} + ',...
    num2str(A2),'e^{-',num2str(s2),'t} + 1  t \geq 0'],...
    'fontsize',14,'fontname','times')
set(findobj('type','line'),'linewidth',1.5,'markersize',12)
set(findobj('type','axes'),'linewidth',2,'fontsize',14,'fontname','times')
```





- The **forward**-looking Euler numerical solution (text method) is **unstable**. However, the **backward**-looking Euler numerical solution is **stable**.