Euler Method on Bacterial Growth

Table of Contents

Document Info

```
% Written by: Drew Christner
% Class: MAE 215 Final Project Part 1
% Date: 14 April 2021
```

Variables

```
dt = .5 ; % base change in time
dt_mod = 3 ; % modifier for run 2

r = .8 ; % growth rate per hour

t_start = 0 ; % beggioning of time vector
t_end = 4 ; % end of time vector
ta = t_start:dt:t_end ; % time for run 1
tb = t_start:(dt):t_end ; % time for run 2

m = (t_end-t_start)/dt ;

Na = zeros(1,t_end+1) ; % concentration for run 1
Nb = zeros(1,t_end+1) ; % concentration for run 2
Nc = zeros(1,t_end+1) ; % concentration for predictor run 1
Nd = zeros(1,t_end+1) ; % concentration for predictor run 2
Na(1) = 1000 ; % starting value for run 1
Nb(1) = 1000 ; % starting value for run 2
Nc(1) = 1000 ; % starting value for predictor run 1
Nd(1) = 1000 ; % starting value for predictor run 1
```

Concentration from Euler Loop

```
for i = 1:m
    Na(i+1) = Na(i) + r*dt*Na(i) ;
end
```

```
for i = 1:m
    Nb(i+1) = Nb(i) + r*(dt*dt_mod)*Nb(i) ;
end
```

Concentration from Analytical Evaluation

```
Nexa = Na(1)*exp(r*ta) ;

Nexb = Nb(1)*exp(r*tb) ;
```

Concentration from Predictor-Corrector

```
for i = 1:m % loop that takes a the slope of the future and current
 term, averages them, then calculates
    mx1 = r*Nc(i) ;
    Nc(i+1) = Nc(i) + r*dt*Nc(i) ;
    mx2 = r*Nc(i+1) ;
    Nc(i+1) = Nc(i) + dt*.5*(mx1+mx2) ;
end

for i = 1:m % repeats above but for modified dt
    mx1 = r*Nd(i) ;
    Nd(i+1) = Nd(i) + r*(dt*dt_mod)*Nd(i) ;
    mx2 = r*Nd(i+1) ;
    Nd(i+1) = Nd(i) + (dt*dt_mod)*.5*(mx1+mx2) ;
end
```

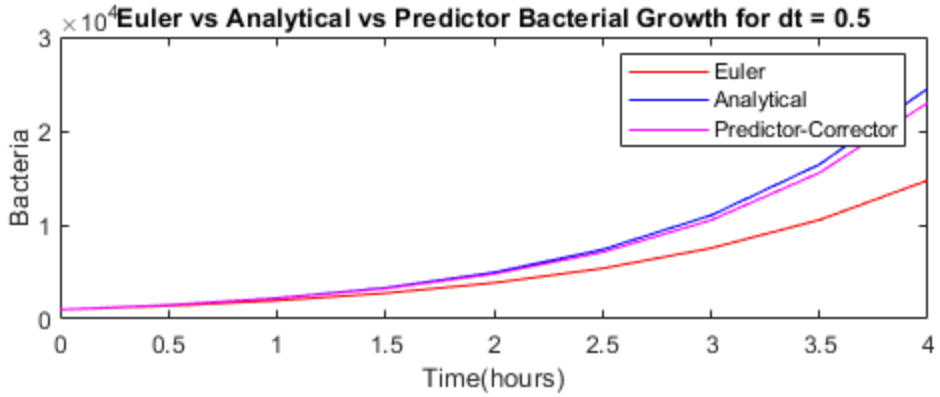
Calculating Percent Error

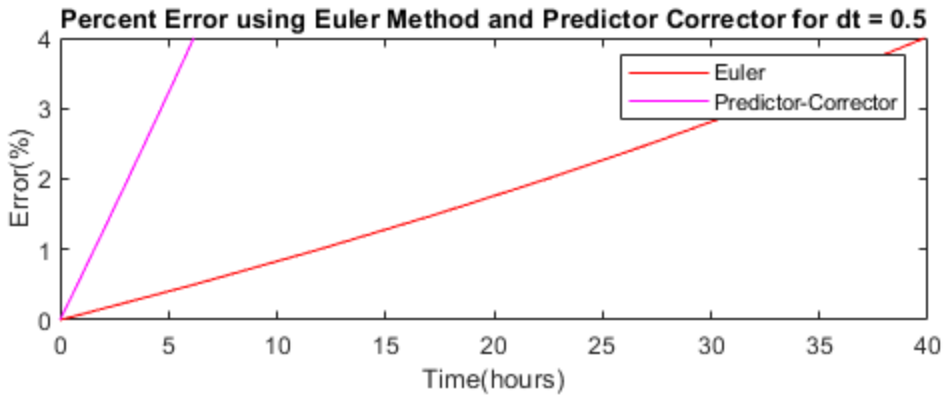
```
errorA = (abs(Na-Nexa)./Nexa).*100 ;
errorB = (abs(Nb-Nexb)./Nexb).*100 ;
errorC = (abs(Nc-Nexa)./Nexa).*100 ;
errorD = (abs(Nd-Nexb)./Nexb).*100 ;
```

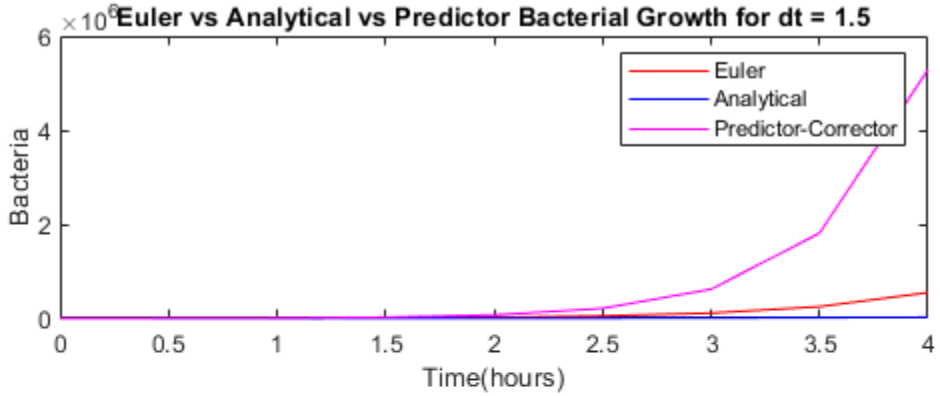
Plotting

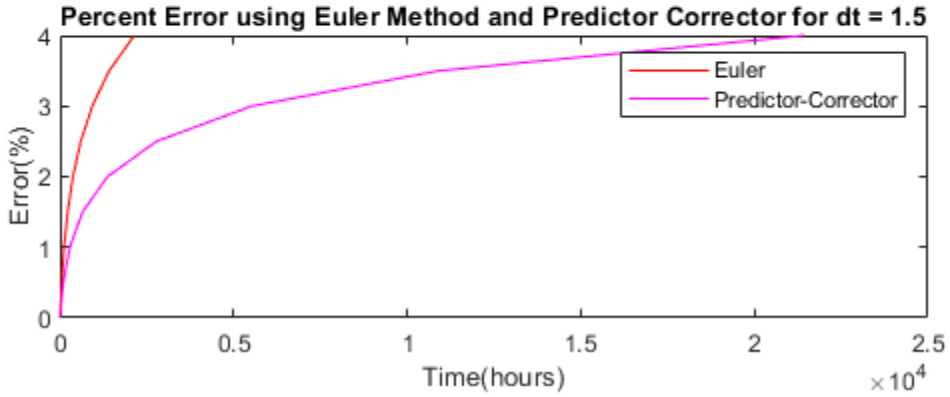
```
% Run 1
subplot(2,1,1)
plot(ta,Na,'r')
xlabel('Time(hours)')
ylabel('Bacteria')
title(['Euler vs Analytical vs Predictor Bacterial Growth for dt = '
 num2str(dt)])

hold on
plot(ta,Nexa,'b')

hold on
plot(ta,Nc,'m')
legend('Euler','Analytical','Predictor-Corrector')

subplot(2,1,2)
```

```
plot(errorA,ta,'r')

hold on
plot(errorC,ta,'m')
xlabel('Time(hours)')
ylabel('Error(%)')
title(['Percent Error using Euler Method and Predictor Corrector for
 dt = ' num2str(dt)])
legend('Euler','Predictor-Corrector')

figure

% Run 2

subplot(2,1,1)
plot(tb,Nb,'r')
xlabel('Time(hours)')
ylabel('Bacteria')
title(['Euler vs Analytical vs Predictor Bacterial Growth for dt = '
 num2str(dt*dt_mod)])

hold on

plot(tb,Nexb,'b')

hold on
plot(tb,Nd,'m')
legend('Euler','Analytical','Predictor-Corrector')

subplot(2,1,2)
plot(errorB,tb,'r')

hold on
plot(errorD,tb,'m')
xlabel('Time(hours)')
ylabel('Error(%)')
title(['Percent Error using Euler Method and Predictor Corrector for
 dt = ' num2str(dt*dt_mod)])
legend('Euler','Predictor-Corrector')
```


×10^4
Euler vs Analytical vs Predictor Bacterial Growth for dt = 0.5
Bacteria
Time(hours)
Euler
Analytical
Predictor-Corrector
0
1
2
3
0
0.5
1
1.5
2
2.5
3
3.5
4


Percent Error using Euler Method and Predictor Corrector for dt = 0.5
Error(%)
Time(hours)
Euler
Predictor-Corrector
0
1
2
3
4
0
5
10
15
20
25
30
35
40


×10^6
Euler vs Analytical vs Predictor Bacterial Growth for dt = 1.5
Bacteria
Time(hours)
Euler
Analytical
Predictor-Corrector
0
2
4
6
0
0.5
1
1.5
2
2.5
3
3.5
4


Percent Error using Euler Method and Predictor Corrector for dt = 1.5
Error(%)
Time(hours)
×10^4
Euler
Predictor-Corrector
0
1
2
3
4
0
0.5
1
1.5
2
2.5

Analysis

```
% The Euler Method error graphs showed that larger and smaller dt have
% different levels of error at different times.  The smaller dt has
 less
% error the closer to 0 the function is evaluated.  As time goes to
% infinity, a larger dt will have less error for the larger values.

% The Predictor-Corrector method follows the same pattern as the euler
% method.  When comparing the Euler and predictor methods, the Euler
 method
% has less error on a smaller dt.  Contrastly Predictor has a lower
 error
% than Euler as the dt gets larger.
```

Published with MATLAB® R2020a