Group Project Part II - Paige Pellouchoud

Table of Contents

running teammates runge kutta 4th order code to get SIR population data for seasonal influenza

Givens

```
S(1) = 990;
I(1) = 10;
R(1) = 0;
h = 2; % time step 2 days
Time = 100; % T = 100 days

% Seasonal Influenza
% Beta = 0.3 Gamma = 0.1
N = @(S, I, R) S + I + R;

Beta = 0.3;
gamma = 0.1;

dSdt = @(N, S, I) (-Beta/N)*S*I;
dIdt = @(N, S, I) (Beta/N)*S*I - gamma*I;
dRdt = @(I) gamma*I;

Population(1) = S(1) + I(1);
Population2(1) = 0;
Population3(1) =  0;

for i = 1:50
    Population(i+1) = N(S(i),I(i),R(i));
    K1Susceptible = dSdt(Population(i), S(i), I(i));
    K1Infected = dIdt(Population(i), S(i), I(i));
    K1Recovered = dRdt(I(i));
```

```
    K2StepsizeS = S(i)+1/2*K1Susceptible*h; %K1 Seasonal Influenza
    K2StepsizeI = I(i)+1/2*K1Infected*h; %K1 Infected
    K2StepsizeR = R(i) + 1/2*K1Recovered*h; %K1 Recovery
    Population2(i+1) = N(K2StepsizeS,K2StepsizeI,K2StepsizeR);
    K2SusceptibleS = dSdt(Population2(i+1), K2StepsizeS, K2StepsizeI); %K2
Seasonal Influenza
    K2Infected = dIdt(Population2(i+1), K2StepsizeS, K2StepsizeI); %K2
Infected
    K2Recovered = dRdt(K2StepsizeI); %K2 Recovery
    K3StepsizeS = S(i)+1/2*K2SusceptibleS*h; %K3 Seasonal Influenza
    K3StepsizeI = I(i) + 1/2*K2Infected*h; %K3 Infected
    K3StepsizeR = R(i) + 1/2*K2Recovered*h; %K3 Recovery
    Population3(i+1) = N(K3StepsizeS,K3StepsizeI,K3StepsizeR);
    K3SusceptibleS = dSdt(Population3(i+1), K3StepsizeS, K3StepsizeI);
    K3Infected = dIdt(Population3(i+1), K3StepsizeS, K3StepsizeI);
    K3Recovered = dRdt(K3StepsizeI);
    K4StepsizeS = S(i) + K3SusceptibleS; %K4 Seasonal Influenza
    K4StepsizeI = I(i) + K3Infected; %K4 Infected
    K4StepsizeR = R(i) + K3Recovered; %K4 Recovery
    Population4(i+1) = N(K4StepsizeS,K4StepsizeI,K4StepsizeR);
    K4SusceptibleS = dSdt(Population4(i+1),K4StepsizeS, K4StepsizeI);
    K4Infected = dIdt(Population4(i+1), K4StepsizeS,K4StepsizeI);
    K4Recovered = dRdt(K4StepsizeI);
    % Adding all the K values up
    S(i+1) = S(i) + (1/3)*(K1Susceptible + 2*K2SusceptibleS +
2*K3SusceptibleS + K4SusceptibleS);  % 1/6 changed to 1/3 since h = 2 now
    I(i+1) = I(i) + (1/3)*(K1Infected + 2*K2Infected + 2*K3Infected +
K4Infected); % 1/6 changed to 1/3 since h = 2 now
    R(i+1) = R(i) + (1/3)*(K1Recovered + 2*K2Recovered + 2*K3Recovered +
K4Recovered); % 1/6 changed to 1/3 since h = 2 now
end

% as of now there is some small error for this 4th order runge kutta code I
can't
% find the mistake

% vectors for h = 2 for 4th order runge kutta 51 values per vector includes
% day zero aka position 1

S;
I;
R;
```

Part II - Interpolation - Paige Pellouchoud

```
Sint = S;
Iint = I;
Rint = R;
```

linear interpolation

```
for i = 1:50
```

```
    Sint(i) = Sint(i) + ((Sint(i+1)-Sint(i))./(2)); %newtons interpolation
to give output for odd days for susceptible population
    Iint(i) = Iint(i) + ((Iint(i+1)-Iint(i))./(2)); %newtons interpolation
to give output for odd days for the infected population
    Rint(i) = Rint(i) + ((Rint(i+1)-Rint(i))./(2)); %newtons interpolation
to give output for odd days for the recovered population

end

% h = 2 Susceptible interpolated odd day population, Infected interpolated
% odd day population, recovered interpolated odd day population

Sint;
Iint;
Rint;
```

Error for Linear Interpolation

```
Nint = 50; % number of interpolated points

%make null set for last values in interpolated sets because from 0 to 100
%there are 51 even values and 50 odd values
Sint(51) = [];
Iint(51) = [];
Rint(51) = [];

%running teammates runge kutta 4th order for seasonal influenza for h = 1 to
extract the odd
%values for the error calculation

% Givens
SS(1) = 990;
II(1) = 10;
RR(1) = 0;
h = 1; % time step 1 days
Time = 100; % T = 100 days

% Seasonal Influenza
% Beta = 0.3 Gamma = 0.1
N = @(SS, II, RR) SS + II + RR;

Beta = 0.3;
gamma = 0.1;

dSdt = @(N, SS, II) (-Beta/N)*SS*II;
dIdt = @(N, SS, II) (Beta/N)*SS*II - gamma*II;
dRdt = @(II) gamma*II;

Population(1) = SS(1) + II(1);
Population2(1) = 0;
Population3(1) = 0;

for i = 1:100
```

```
    Population(i+1) = N(SS(i),II(i),RR(i));
    K1Susceptible = dSdt(Population(i), SS(i), II(i));
    K1Infected = dIdt(Population(i), SS(i), II(i));
    K1Recovered = dRdt(II(i));
    K2StepsizeS = SS(i)+1/2*K1Susceptible*h; %K1 Seasonal Influenza
    K2StepsizeI = II(i)+1/2*K1Infected*h; %K1 Infected
    K2StepsizeR = RR(i) + 1/2*K1Recovered*h; %K1 Recovery
    Population2(i+1) = N(K2StepsizeS,K2StepsizeI,K2StepsizeR);
    K2SusceptibleS = dSdt(Population2(i+1), K2StepsizeS, K2StepsizeI); %K2
Seasonal Influenza
    K2Infected = dIdt(Population2(i+1), K2StepsizeS, K2StepsizeI); %K2
Infected
    K2Recovered = dRdt(K2StepsizeI); %K2 Recovery
    K3StepsizeS = SS(i)+1/2*K2SusceptibleS*h; %K3 Seasonal Influenza
    K3StepsizeI = II(i) + 1/2*K2Infected*h; %K3 Infected
    K3StepsizeR = RR(i) + 1/2*K2Recovered*h; %K3 Recovery
    Population3(i+1) = N(K3StepsizeS,K3StepsizeI,K3StepsizeR);
    K3SusceptibleS = dSdt(Population3(i+1), K3StepsizeS, K3StepsizeI);
    K3Infected = dIdt(Population3(i+1), K3StepsizeS, K3StepsizeI);
    K3Recovered = dRdt(K3StepsizeI);
    K4StepsizeS = SS(i) + K3SusceptibleS; %K4 Seasonal Influenza
    K4StepsizeI = II(i) + K3Infected; %K4 Infected
    K4StepsizeR = RR(i) + K3Recovered; %K4 Recovery
    Population4(i+1) = N(K4StepsizeS,K4StepsizeI,K4StepsizeR);
    K4SusceptibleS = dSdt(Population4(i+1),K4StepsizeS, K4StepsizeI);
    K4Infected = dIdt(Population4(i+1), K4StepsizeS,K4StepsizeI);
    K4Recovered = dRdt(K4StepsizeI);
    % Adding all the K values up
    SS(i+1) = SS(i) + (1/6)*(K1Susceptible + 2*K2SusceptibleS +
2*K3SusceptibleS + K4SusceptibleS);
    II(i+1) = II(i) + (1/6)*(K1Infected + 2*K2Infected + 2*K3Infected +
K4Infected);
    RR(i+1) = RR(i) + (1/6)*(K1Recovered + 2*K2Recovered + 2*K3Recovered +
K4Recovered);
end

% population results for h = 1 day

 SS;
 II;
 RR;

% null 101
SS(101) = [];
II(101) = [];
RR(101) = [];

%extracting odd days
Sodd = SS(2:2:end);
Iodd = II(2:2:end);
Rodd = RR(2:2:end);

Sodd;
Iodd;
```

```
Rodd;

%ran teammates runge kutta 4th order for seasonal influenza for h = 1 to
extract the odd
%values for the error calculation

for i = 1:50
 SEL2 = sqrt(sum((Sint(i)-Sodd(i)).^2)./Nint);
 IEL2 = sqrt(sum((Iint(i)-Iodd(i)).^2)./Nint);
 REL2 = sqrt(sum((Rint(i)-Rodd(i)).^2)./Nint);
end
```

print errors for Linear Interpolation

```
SEL2
IEL2
REL2


SEL2 =

    0.3099


IEL2 =

    0.0188


REL2 =

    0.3286
```

Quadratic Interpolation

```
% Begin newton quadratic interpolation
% Operate on S vector-51 values at even numbered days
% Interpolated values will be calculated at odd numbered days.

Time = [1:1:101];

%quadratic interpolation results for 49 values starting at t=1 day
% and ending at t=49 days

for i=1:49 %i=1 when Time=0
   %Begin quadratic interpolation for S
   M(i) = (S(i+1)-S(i))./(Time(i+1)-Time(i));
   P(i) = ((S(i+2)-S(i+1))./(Time(i+2)-Time(i+1)))-M(i);
   %Some values for I calculation are inserted as +1 for this interpolation
   %Some values for S  calc are inserted as -1 for this interpolation
   Squint(i)=S(i)+(M(i).*1)+((P(i)./(Time(i+2)-Time(i))).*(1).*(-1));
   %page 217(i)
```

```
   %Begin quadratic interpolation for I
   M(i) = (I(i+1)-I(i))./(Time(i+1)-Time(i));
   P(i) = ((I(i+2)-I(i+1))./((Time(i+2)-Time(i+1)))-M(i));
   Iquint(i)=I(i)+(M(i).*1)+((P(i)./(Time(i+2)-Time(i))).*(1).*(-1));
   %page 217

   %Begin quatratic interpolation for R
   M(i) = (R(i+1)-R(i))./(Time(i+1)-Time(i));
   P(i) = ((R(i+2)-R(i+1))./((Time(i+2)-Time(i+1)))-M(i));
   Rquint(i)=R(i)+(M(i).*1)+((P(i)./(Time(i+2)-Time(i))).*(1).*(-1));
   %page 217

   %equation 6.51 quadratic interpolation
end

% quadratic interpolation results for 50th odd day value manually
% calculated makes arrays even for 50 odd number days in the interval.
Squint(50) = 86.945;
Iquint(50) = 1.8280;
Rquint(50) = 938.9814;

%print quadratic interpolated values for h = 2 days

Squint;
Iquint;
Rquint;
```

Error for Quadratic Interpolation

```
for i = 1:50
 SEL2Q = sqrt(sum((Squint(i)-Sodd(i)).^2)./Nint);
 IEL2Q = sqrt(sum((Iquint(i)-Iodd(i)).^2)./Nint);
 REL2Q = sqrt(sum((Rquint(i)-Rodd(i)).^2)./Nint);
end
```

print errors for Quadratic Interpolation

```
SEL2Q
IEL2Q
REL2Q
```

```
SEL2Q =

    3.9295


IEL2Q =

    0.0203
```

```
REL2Q =

     0.0247
```

Published with MATLAB® R2023b

	Error in Odd-Day Values when Odd Days were estimated with **Linear Interpolation**	Error in Odd-Day Values when Odd Days were estimated with **Quadratic Interpolation**
Susceptible Population	0.3099 (SEL2)	3.9295 (SEL2Q)
Infected Population	0.0188 (IEL2)	0.0203 (IEL2Q)
Recovered Population	0.3286 (REL2)	0.0247 (REL2Q)