Robert Heeter
BIOE 391 Numerical Methods – Due 22 April 2022

Problem Set 10 Solutions

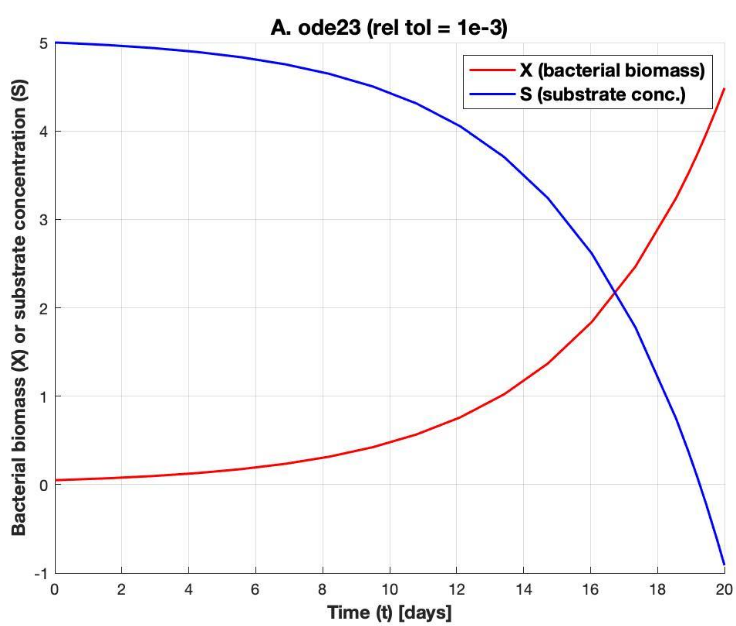
1.

a. Output

PART A: ode23: rel tol = 1e-3, incorrect (neg) results, tictoc = 0.007541s

t	X	S
0.0000	0.0500	5.0000
0.3556	0.0542	4.9944
1.6491	0.0724	4.9701
2.9566	0.0972	4.9371
4.2644	0.1304	4.8928
5.5723	0.1750	4.8333
6.8801	0.2348	4.7536
8.1879	0.3151	4.6466
9.4958	0.4228	4.5030
10.8036	0.5673	4.3103
12.1114	0.7611	4.0518
13.4193	1.0213	3.7049
14.7271	1.3704	3.2395
16.0350	1.8388	2.6150
17.3429	2.4673	1.7769
18.5546	3.2399	0.7468
18.9151	3.5136	0.3819
19.1785	3.7280	0.0960
19.4419	3.9558	-0.2078
19.7693	4.2583	-0.6110
20.0000	4.4852	-0.9136

Figure



Robert Heeter
BIOE 391 Numerical Methods – Due 22 April 2022

b. Output

PART B: ode23: rel tol = 1e-6, correct (pos) results, tictoc = 0.024118s

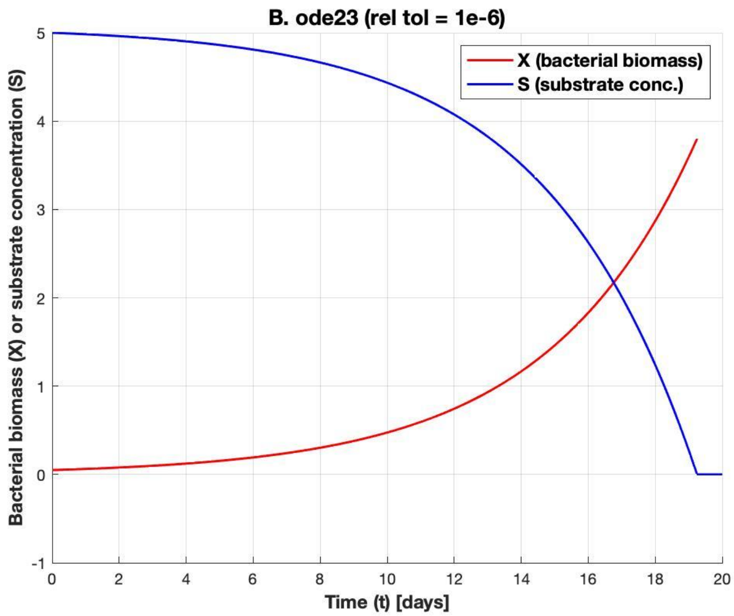
t	X	S
0.0000	0.0500	5.0000
0.3556	0.0542	4.9944
0.7111	0.0587	4.9884
1.0440	0.0632	4.9823
1.3686	0.0680	4.9760
1.6854	0.0731	4.9693
1.9948	0.0783	4.9622
2.2971	0.0838	4.9549
2.5925	0.0896	4.9472
2.8815	0.0956	4.9392
3.1643	0.1019	4.9308
3.4411	0.1084	4.9221
3.7123	0.1153	4.9130
3.9779	0.1224	4.9035
4.2384	0.1298	4.8937
4.4938	0.1374	4.8834
4.7443	0.1454	4.8728
4.9902	0.1537	4.8618
5.2315	0.1622	4.8503
5.4686	0.1711	4.8385
5.7014	0.1803	4.8262
5.9302	0.1899	4.8135
6.1552	0.1997	4.8004
6.3763	0.2099	4.7868
6.5939	0.2204	4.7728
6.8078	0.2313	4.7583
7.0184	0.2425	4.7433
7.2257	0.2541	4.7279
7.4298	0.2660	4.7119
7.6308	0.2784	4.6955
7.8288	0.2910	4.6786
8.0238	0.3041	4.6612
8.2160	0.3175	4.6433
8.4055	0.3314	4.6249
8.5923	0.3456	4.6059
8.7765	0.3602	4.5864
8.9582	0.3752	4.5664
9.1374	0.3907	4.5458
9.3142	0.4065	4.5246
9.4887	0.4228	4.5029
9.6609	0.4395	4.4807
9.8309	0.4566	4.4578
9.9987	0.4742	4.4344
10.1645	0.4922	4.4104
10.3281	0.5107	4.3857
10.4898	0.5296	4.3605

10.6496	0.5490	4.3347
10.8074	0.5689	4.3082
10.9634	0.5892	4.2811
11.1176	0.6100	4.2534
11.2700	0.6312	4.2250
11.4207	0.6530	4.1960
11.5696	0.6753	4.1663
11.7170	0.6980	4.1360
11.8627	0.7213	4.1049
12.0068	0.7451	4.0732
12.1493	0.7694	4.0409
12.2904	0.7942	4.0078
12.4299	0.8195	3.9740
12.5681	0.8454	3.9395
12.7047	0.8718	3.9043
12.8400	0.8987	3.8684
12.9740	0.9262	3.8317
13.1065	0.9542	3.7943
13.2378	0.9828	3.7562
13.3678	1.0120	3.7173
13.4971	1.0419	3.6775
13.6263	1.0726	3.6365
13.7555	1.1043	3.5943
13.8848	1.1368	3.5509
14.0140	1.1704	3.5062
14.1433	1.2049	3.4601
14.2725	1.2405	3.4127
14.4017	1.2771	3.3639
14.5310	1.3148	3.3137
14.6602	1.3535	3.2619
14.7895	1.3935	3.2087
14.9187	1.4346	3.1539
15.0480	1.4769	3.0974
15.1772	1.5205	3.0393
15.3064	1.5654	2.9795
15.4357	1.6116	2.9179
15.5649	1.6591	2.8545
15.6942	1.7081	2.7893
15.8234	1.7585	2.7221
15.9527	1.8103	2.6529
16.0819	1.8638	2.5817
16.2112	1.9188	2.5083
16.3404	1.9754	2.4328
16.4676	2.0327	2.3564
16.5923	2.0905	2.2793
16.7145	2.1488	2.2016
16.8343	2.2075	2.1233
16.9516	2.2666	2.0446
17.0665	2.3259	1.9654

17.1790	2.3855	1.8859
17.2891	2.4454	1.8062
17.3968	2.5054	1.7262
17.5021	2.5654	1.6461
17.6050	2.6255	1.5660
17.7056	2.6856	1.4859
17.8037	2.7455	1.4059
17.8995	2.8053	1.3262
17.9928	2.8649	1.2469
18.0837	2.9240	1.1679
18.1721	2.9828	1.0896
18.2581	3.0411	1.0119
18.3417	3.0988	0.9350
18.4227	3.1558	0.8590
18.5029	3.2132	0.7823
18.5827	3.2715	0.7047
18.6621	3.3304	0.6261
18.7412	3.3902	0.5464
18.8201	3.4509	0.4655
18.8989	3.5126	0.3832
18.9780	3.5756	0.2991
19.0585	3.6410	0.2120
19.1456	3.7130	0.1159
19.1723	3.7353	0.0862
19.1989	3.7578	0.0563
19.2144	3.7708	0.0389
19.2298	3.7840	0.0214
19.2361	3.7893	0.0143
19.2406	3.7930	0.0093
19.2434	3.7954	0.0061
19.2453	3.7970	0.0040
19.2466	3.7981	0.0026
19.2474	3.7988	0.0017
19.2480	3.7992	0.0011
19.2484	3.7995	0.0007
19.2487	3.7997	0.0004
19.2489	3.7998	0.0002
19.2491	3.7999	0.0001
19.2492	3.8000	0.0000
19.2494	3.8000	0.0000
19.2495	3.8000	0.0000
19.2498	3.8000	-0.0000
19.2503	3.8000	0.0000
19.2505	3.8000	-0.0000
19.2508	3.8000	0.0000
19.2510	3.8000	-0.0000
19.2513	3.8000	0.0000
19.2514	3.8000	-0.0000
19.2517	3.8000	0.0000

19.2519	3.8000	-0.0000
19.2522	3.8000	0.0000
19.2524	3.8000	-0.0000
19.2526	3.8000	0.0000
19.2528	3.8000	-0.0000
19.2530	3.8000	0.0000
...	...	...

Figure

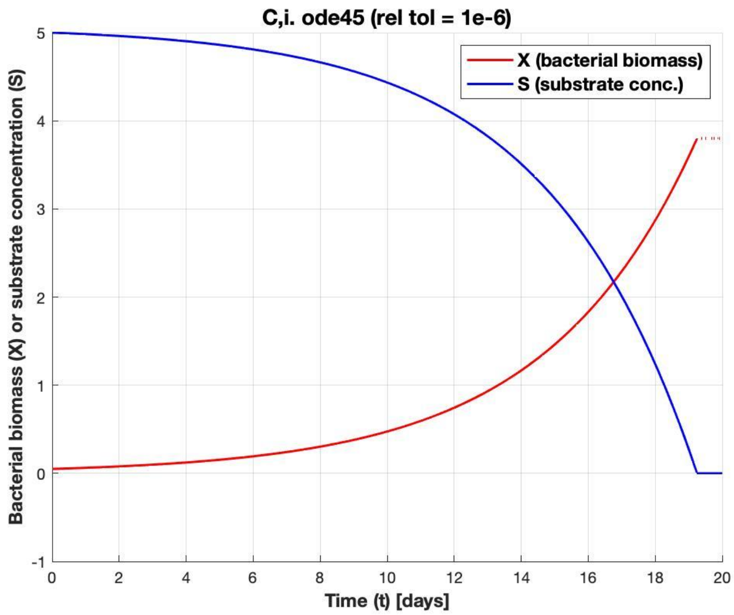


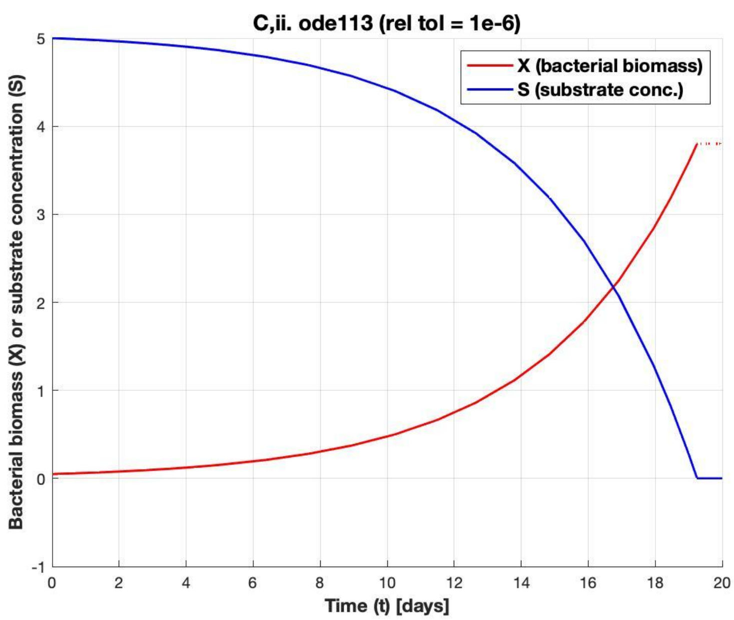
c. Output

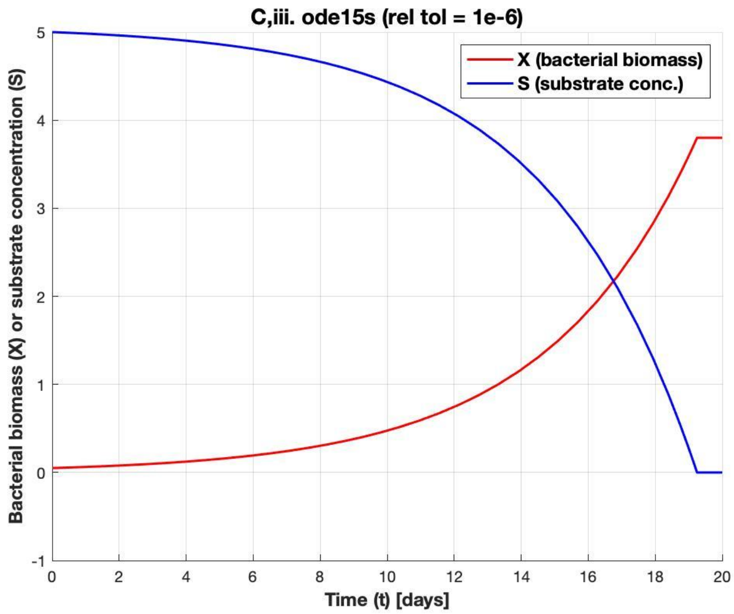
ode45: rel tol = 1e-6, correct (pos) results, tictoc = 0.038857s
ode113: rel tol = 1e-6, correct (pos) results, tictoc = 0.097805s
ode15s: rel tol = 1e-6, correct (pos) results, tictoc = 0.027030s
ode23s: rel tol = 1e-6, correct (pos) results, tictoc = 0.020104s
ode23t: rel tol = 1e-6, correct (pos) results, tictoc = 0.018544s
ode23tb: rel tol = 1e-6, correct (pos) results, tictoc = 0.016578s

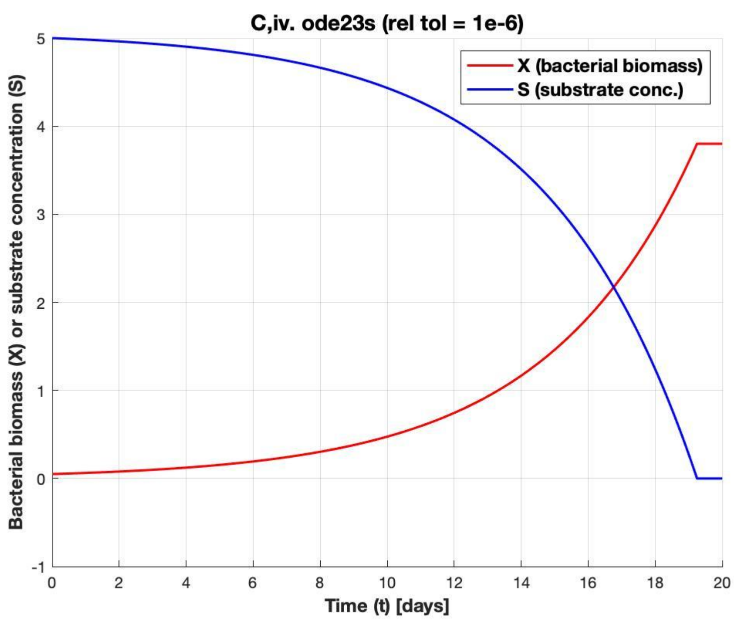
Robert Heeter
BIOE 391 Numerical Methods – Due 22 April 2022

Figures

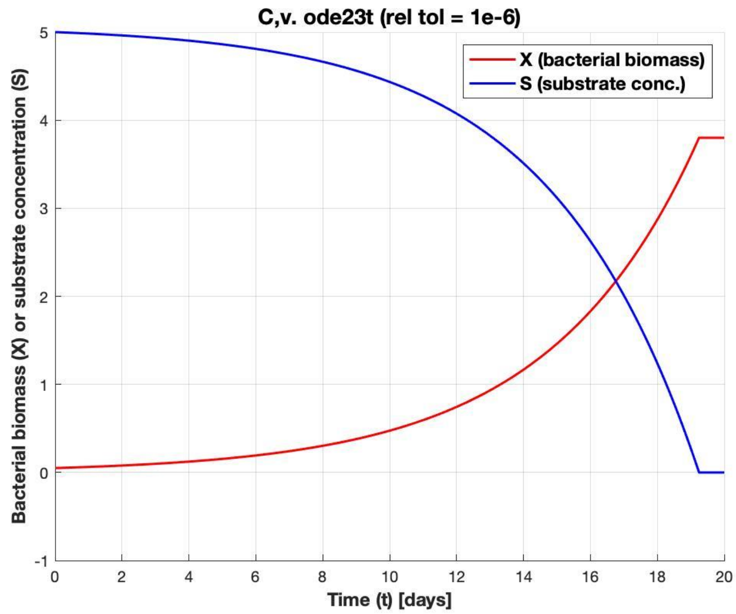


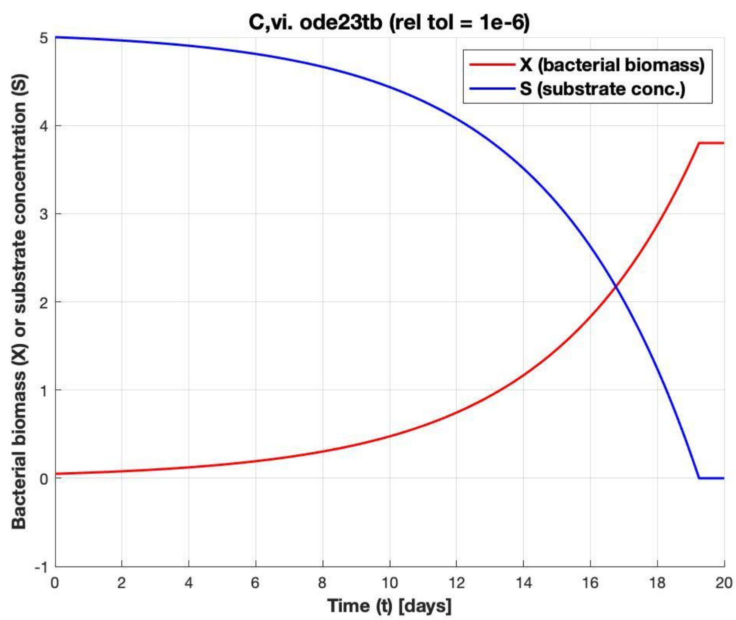



C,iii. ode15s (rel tol = 1e-6)
X (bacterial biomass)
S (substrate conc.)
Bacterial biomass (X) or substrate concentration (S)
Time (t) [days]
5
4
3
2
1
0
-1
0
2
4
6
8
10
12
14
16
18
20


C,iv. ode23s (rel tol = 1e-6)
X (bacterial biomass)
S (substrate conc.)
Bacterial biomass (X) or substrate concentration (S)
Time (t) [days]
5
4
3
2
1
0
-1
0
2
4
6
8
10
12
14
16
18
20

Robert Heeter
BIOE 391 Numerical Methods – Due 22 April 2022



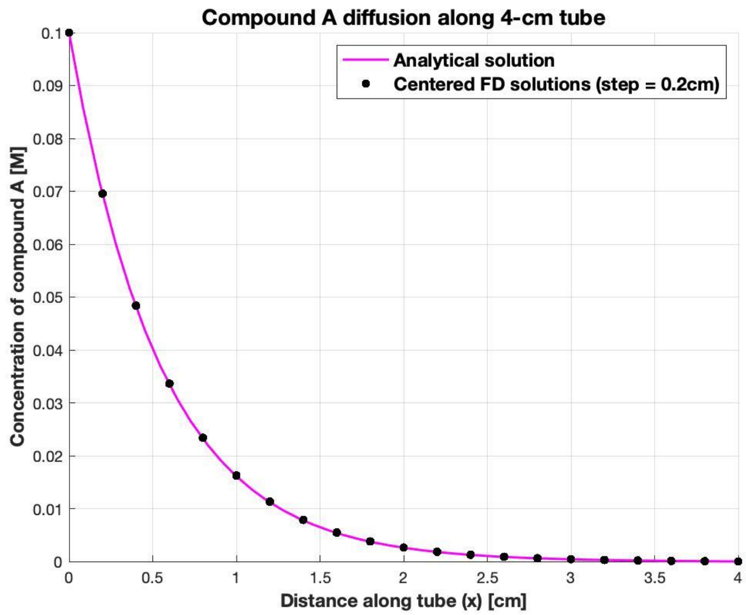


2. **Output**

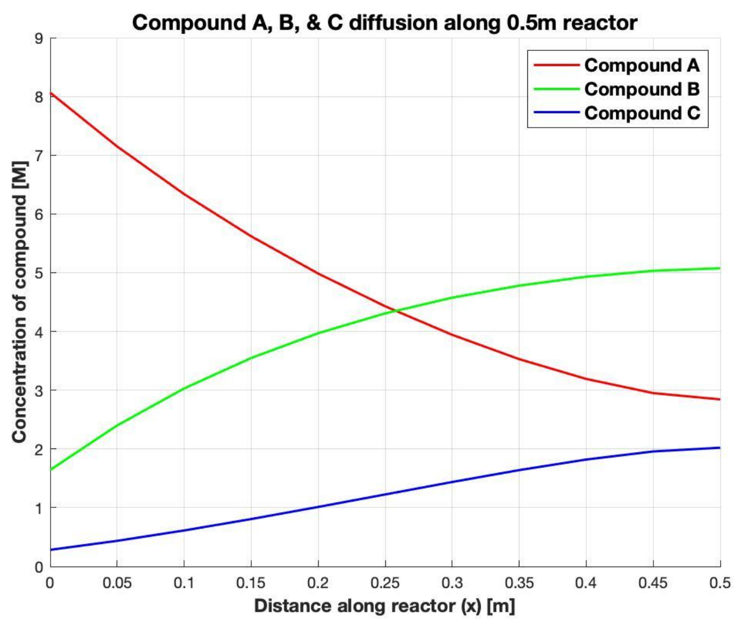
k constants:
k_12 = 0.576573
k_21 = 1.032396
k_31 = 0.105071
k_32 = 0.456061

Robert Heeter
BIOE 391 Numerical Methods – Due 22 April 2022

3. **Figure** *(see end of document for analytical solution)*



4. **Figure**

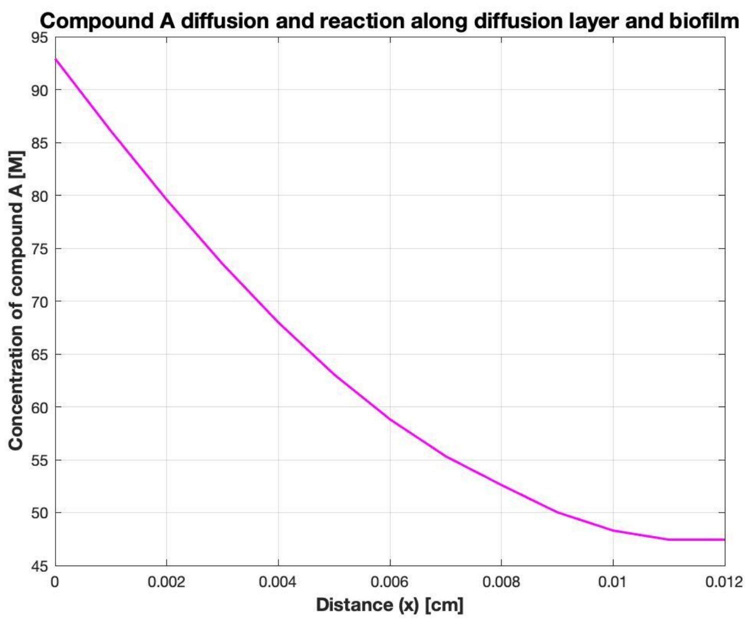


Robert Heeter
BIOE 391 Numerical Methods – Due 22 April 2022

5. **Output**

x (cm)	c_a (M)
0.0000	92.9211
0.0010	86.0754
0.0020	79.5709
0.0030	73.5080
0.0040	67.9772
0.0050	63.0588
0.0060	58.8215
0.0070	55.3216
0.0080	52.6026
0.0090	50.0261
0.0100	48.2992
0.0110	47.4357
0.0120	47.4357

Figure



6. *Attempted but did not finish.*

Robert Heeter
BIOE 391 Numerical Methods – Due 22 April 2022

Complete MATLAB Code

```
% Robert Heeter
% BIOE 391 Numerical Methods
% HOMEWORK 10 MATLAB SCRIPT

clc, clf, clear, close all

%% P1. PROBLEM 23.27
disp('P1. PROBLEM 23.27');

tint = [0 20]; % time interval (days)
ivals = [0.05 5]; % inital values X(0) = 0.05, S(0) = 5

% PART A
tic;
[t,I] = ode23(@bacteria,tint,ivals); % use @bacteria function below for system of differential
equations
time = toc; % execution timing
tXS = [t(:) I(:,1) I(:,2)]'; % combine results into one matrix

figure
hold on
plot(t,I(:,1),'-r','LineWidth',1.5);
plot(t,I(:,2),'-b','LineWidth',1.5);
hold off
grid on
xlabel('Time (t) [days]','FontSize',12,'FontWeight','bold');
ylabel('Bacterial biomass (X) or substrate concentration (S)','FontSize',12,'FontWeight','bold');
title('A. ode23 (rel tol = 1e-3)','FontSize',14,'FontWeight','bold');
legend('X (bacterial biomass)','S (substrate conc.)','FontSize',12,'FontWeight','bold');

fprintf('PART A: ode23: rel tol = 1e-3, incorrect (neg) results, tictoc = %fs\n  t         X
S\n',time);
fprintf('  %7.4f  %7.4f  %7.4f  \n',tXS);
fprintf('\n');

% PART B
opt = odeset('RelTol',1e-6); % set relative tolerance = 1e-6

tic;
[t,I] = ode23(@bacteria,tint,ivals,opt);
tXS = [t(:) I(:,1) I(:,2)]';
time = toc;

figure
hold on
plot(t,I(:,1),'-r','LineWidth',1.5);
plot(t,I(:,2),'-b','LineWidth',1.5);
hold off
grid on
xlabel('Time (t) [days]','FontSize',12,'FontWeight','bold');
ylabel('Bacterial biomass (X) or substrate concentration (S)','FontSize',12,'FontWeight','bold');
title('B. ode23 (rel tol = 1e-6)','FontSize',14,'FontWeight','bold');
legend('X (bacterial biomass)','S (substrate conc.)','FontSize',12,'FontWeight','bold');

fprintf('PART B: ode23: rel tol = 1e-6, correct (pos) results, tictoc = %fs\n  t         X
S\n',time);
fprintf('  %7.4f  %7.4f  %7.4f  \n',tXS(1:3,1:150));
fprintf('  ...       ...      ...\n\n');

% PART C
fprintf('PART C:\n');
tic;
[t,I] = ode45(@bacteria,tint,ivals,opt);
```

Robert Heeter
BIOE 391 Numerical Methods – Due 22 April 2022

```
time1 = toc;
tXS1 = [t(:) I(:,1) I(:,2)]';
fprintf('ode45: rel tol = 1e-6, correct (pos) results, tictoc = %fs\n',time1);

figure
hold on
plot(t,I(:,1),'-r','LineWidth',1.5);
plot(t,I(:,2),'-b','LineWidth',1.5);
hold off
grid on
xlabel('Time (t) [days]','FontSize',12,'FontWeight','bold');
ylabel('Bacterial biomass (X) or substrate concentration (S)','FontSize',12,'FontWeight','bold');
title('C,i. ode45 (rel tol = 1e-6)','FontSize',14,'FontWeight','bold');
legend('X (bacterial biomass)','S (substrate conc.)','FontSize',12,'FontWeight','bold');

tic;
[t,I] = ode113(@bacteria,tint,ivals,opt);
time2 = toc;
tXS2 = [t(:) I(:,1) I(:,2)]';
fprintf('ode113: rel tol = 1e-6, correct (pos) results, tictoc = %fs\n',time2);

figure
hold on
plot(t,I(:,1),'-r','LineWidth',1.5);
plot(t,I(:,2),'-b','LineWidth',1.5);
hold off
grid on
xlabel('Time (t) [days]','FontSize',12,'FontWeight','bold');
ylabel('Bacterial biomass (X) or substrate concentration (S)','FontSize',12,'FontWeight','bold');
title('C,ii. ode113 (rel tol = 1e-6)','FontSize',14,'FontWeight','bold');
legend('X (bacterial biomass)','S (substrate conc.)','FontSize',12,'FontWeight','bold');

tic;
[t,I] = ode15s(@bacteria,tint,ivals,opt);
time3 = toc;
tXS3 = [t(:) I(:,1) I(:,2)]';
fprintf('ode15s: rel tol = 1e-6, correct (pos) results, tictoc = %fs\n',time3);

figure
hold on
plot(t,I(:,1),'-r','LineWidth',1.5);
plot(t,I(:,2),'-b','LineWidth',1.5);
hold off
grid on
xlabel('Time (t) [days]','FontSize',12,'FontWeight','bold');
ylabel('Bacterial biomass (X) or substrate concentration (S)','FontSize',12,'FontWeight','bold');
title('C,iii. ode15s (rel tol = 1e-6)','FontSize',14,'FontWeight','bold');
legend('X (bacterial biomass)','S (substrate conc.)','FontSize',12,'FontWeight','bold');

tic;
[t,I] = ode23s(@bacteria,tint,ivals,opt);
time4 = toc;
tXS4 = [t(:) I(:,1) I(:,2)]';
fprintf('ode23s: rel tol = 1e-6, correct (pos) results, tictoc = %fs\n',time4);

figure
hold on
plot(t,I(:,1),'-r','LineWidth',1.5);
plot(t,I(:,2),'-b','LineWidth',1.5);
hold off
grid on
xlabel('Time (t) [days]','FontSize',12,'FontWeight','bold');
ylabel('Bacterial biomass (X) or substrate concentration (S)','FontSize',12,'FontWeight','bold');
title('C,iv. ode23s (rel tol = 1e-6)','FontSize',14,'FontWeight','bold');
legend('X (bacterial biomass)','S (substrate conc.)','FontSize',12,'FontWeight','bold');
```

Robert Heeter


```
tic;
[t,I] = ode23t(@bacteria,tint,ivals,opt);
time5 = toc;
tXS5 = [t(:) I(:,1) I(:,2)]';
fprintf('ode23t: rel tol = 1e-6, correct (pos) results, tictoc = %fs\n',time5);

figure
hold on
plot(t,I(:,1),'-r','LineWidth',1.5);
plot(t,I(:,2),'-b','LineWidth',1.5);
hold off
grid on
xlabel('Time (t) [days]','FontSize',12,'FontWeight','bold');
ylabel('Bacterial biomass (X) or substrate concentration (S)','FontSize',12,'FontWeight','bold');
title('C,v. ode23t (rel tol = 1e-6)','FontSize',14,'FontWeight','bold');
legend('X (bacterial biomass)','S (substrate conc.)','FontSize',12,'FontWeight','bold');

tic;
[t,I] = ode23tb(@bacteria,tint,ivals,opt);
time6 = toc;
tXS6 = [t(:) I(:,1) I(:,2)]';
fprintf('ode23tb: rel tol = 1e-6, correct (pos) results, tictoc = %fs\n\n',time6);

figure
hold on
plot(t,I(:,1),'-r','LineWidth',1.5);
plot(t,I(:,2),'-b','LineWidth',1.5);
hold off
grid on
xlabel('Time (t) [days]','FontSize',12,'FontWeight','bold');
ylabel('Bacterial biomass (X) or substrate concentration (S)','FontSize',12,'FontWeight','bold');
title('C,vi. ode23tb (rel tol = 1e-6)','FontSize',14,'FontWeight','bold');
legend('X (bacterial biomass)','S (substrate conc.)','FontSize',12,'FontWeight','bold');


%% P2. PROBLEM 23.32
disp('P2. PROBLEM 23.32');

cdata = [85.3 66.6 60.6 56.1 49.1 45.3 41.9 37.8 33.7 34.4 35.1;
    16.9 18.7 24.1 20.9 18.9 19.9 20.6 13.9 19.1 14.5 15.4;
    4.7 7.9 20.1 22.8 32.5 37.7 42.4 47 50.5 52.3 51.3]'; % concentration data

kguess = [0.15 0.15 0.15 0.15]; % initial guess for constants k
k = fminsearch(@conc_SSR,kguess,[],cdata); % use fminsearch to optimize values of k with additional
functions @conc_SSR and @concentration
fprintf('k constants:\nk_12 = %f\nk_21 = %f\nk_31 = %f\nk_32 = %f\n\n',k); % display results


%% P3. PROBLEM 24.11
disp('P3. PROBLEM 24.11');

D = 1.5e-6; % D coefficient (cm^2/s)
k = 5e-6; % k coefficient (s^-1)
L = 4; % length of tube (cm)

step = 0.2; % step size for finite difference
A_i = 0.1; % A(x=0) boundary concentration (M)
A_f = 0; % A(x=4) boundary concentration (M)

coefa = zeros(L/step - 1,L/step - 1); % preallocate matrix of coefficients
dim = size(coefa);
diag_1 = 1:(dim(1)+1):(dim(1)*dim(2)); % indices for middle diagonal
diag_2 = dim(1)+1:(dim(1)+1):(dim(1)*dim(2)); % indices for upper diagonal
diag_3 = 2:(dim(1)+1):(dim(1)*dim(2)); % indices for lower diagonal

coefa(diag_1) = 2 + ((k/D)*step^2); % set values for tridiagonal matrix
```

```
coefa(diag_2) = -1;
coefa(diag_3) = -1;

b = zeros(L/step - 1,1); % set values for constant column vector
b(1) = A_i;
b(end) = A_f;

A_int = gaussseidel(coefa,b); % use gaussseidel to solve coef*A_int = b for internal concentrations
A_comp = [A_i; A_int(:); A_f]; % complete concentration vector including boundary conditions
x = 0:step:L;

figure % plot results
hold on
fplot(@(x) (-4.5365247157e-8).*exp(1.82574185835.*x) +
0.100000045365.*exp(-1.82574185835.*x),[0,4],'-m','LineWidth',1.5); % analytical solution
plot(x,A_comp,'.k','MarkerSize',15);
hold off
grid on
xlabel('Distance along tube (x) [cm]','FontSize',12,'FontWeight','bold');
ylabel('Concentration of compound A [M]','FontSize',12,'FontWeight','bold');
title('Compound A diffusion along 4-cm tube','FontSize',14,'FontWeight','bold');
legend('Analytical solution','Centered FD solutions (step = 0.2cm)','FontSize',12,'FontWeight','bold');


%% P4. PROBLEM 24.13
disp('P4. PROBLEM 24.13');

D = 0.1; % constant (m^2/min)
U = 1; % constant (m/min)
k1 = 3; % constant (1/min)
k2 = 1; % constant (1/min)
L = 0.5; % length of reactor (m)

step = 0.05; % step size for finite difference
ca_i = 10; % input concentrations of A, B, & C
cb_i = 0;
cc_i = 0;

% PRODUCT A
coefa = zeros(L/step + 1,L/step + 1); % preallocate matrix of coefficients
dim = size(coefa);
diag_1 = 1:(dim(1)+1):(dim(1)*dim(2)); % indices for middle diagonal
diag_2 = dim(1)+1:(dim(1)+1):(dim(1)*dim(2)); % indices for upper diagonal
diag_3 = 2:(dim(1)+1):(dim(1)*dim(2)); % indices for lower diagonal

coefa(diag_1) = (-2*D/(step^2)) - k1; % set values for tridiagonal matrix
coefa(diag_2) = (D/(step^2)) - (U/(2*step));
coefa(diag_3) = (D/(step^2)) + (U/(2*step));

coefa(1,1:3) = [U+(3*D/(2*step)), -2*D/(step), D/(2*step)]; % boundary conditions
coefa(end,end-1:end) = [-2*D/(step^2), (2*D/(step^2))+k1];

b = zeros(L/step + 1,1); % set values for constant column vector
b(1) = U*ca_i; % boundary conditions
b(end) = 0;

ca_comp = gaussseidel(coefa,b); % use gaussseidel to solve coefa*ca_comp = b for all concentrations

% PRODUCT B
coefb = zeros(L/step + 1,L/step + 1); % preallocate matrix of coefficients
dim = size(coefb);
diag_1 = 1:(dim(1)+1):(dim(1)*dim(2)); % indices for middle diagonal
diag_2 = dim(1)+1:(dim(1)+1):(dim(1)*dim(2)); % indices for upper diagonal
diag_3 = 2:(dim(1)+1):(dim(1)*dim(2)); % indices for lower diagonal

coefb(diag_1) = (-2*D/(step^2)) - k2; % set values for tridiagonal matrix
```

```
coefb(diag_2) = (D/(step^2)) - (U/(2*step));
coefb(diag_3) = (D/(step^2)) + (U/(2*step));

coefb(1,1:3) = [U+(3*D/(2*step)), -2*D/(step), D/(2*step)]; % boundary conditions
coefb(end,end-1:end) = [-2*D/(step^2), (2*D/(step^2))+k2];

b = -1*k1*ca_comp; % set values for constant column vector
b(1) = U*cb_i; % boundary conditions
b(end) = k1*ca_comp(end);

cb_comp = gaussseidel(coefb,b); % use gaussseidel to solve coefb*cb_comp = b for all concentrations

% PRODUCT C
coefc = zeros(L/step + 1,L/step + 1); % preallocate matrix of coefficients
dim = size(coefc);
diag_1 = 1:(dim(1)+1):(dim(1)*dim(2)); % indices for middle diagonal
diag_2 = dim(1)+1:(dim(1)+1):(dim(1)*dim(2)); % indices for upper diagonal
diag_3 = 2:(dim(1)+1):(dim(1)*dim(2)); % indices for lower diagonal

coefc(diag_1) = (-2*D/(step^2)); % set values for tridiagonal matrix
coefc(diag_2) = (D/(step^2)) - (U/(2*step));
coefc(diag_3) = (D/(step^2)) + (U/(2*step));

coefc(1,1:3) = [U+(3*D/(2*step)), -2*D/(step), D/(2*step)]; % boundary conditions
coefc(end,end-1:end) = [-2*D/(step^2), (2*D/(step^2))];

b = -1*k2*cb_comp; % set values for constant column vector
b(1) = U*cc_i; % boundary conditions
b(end) = k2*cb_comp(end);

cc_comp = gaussseidel(coefc,b); % use gaussseidel to solve coefc*cc_comp = b for all concentrations
x = 0:step:L;

% PLOT RESULTS
figure
hold on
plot(x,ca_comp,'-r','LineWidth',1.5);
plot(x,cb_comp,'-g','LineWidth',1.5);
plot(x,cc_comp,'-b','LineWidth',1.5);
hold off
grid on
xlabel('Distance along reactor (x) [m]','FontSize',12,'FontWeight','bold');
ylabel('Concentration of compound [M]','FontSize',12,'FontWeight','bold');
title('Compound A, B, & C diffusion along 0.5m reactor','FontSize',14,'FontWeight','bold');
legend('Compound A','Compound B','Compound C','FontSize',12,'FontWeight','bold');


%% P5. PROBLEM 24.14
disp('P5. PROBLEM 24.14');

D = 0.8; % diffusion coefficient (cm^2/day)
Df = 0.64; % diffusion coefficient (cm^2/day)
k = 0.1; % first-order rate constant (1/day)
L = 0.008; % length of diffusion layer (cm)
Lf = 0.004; % length of biofilm (cm)

step = 0.001; % step size for finite difference (cm)
ca_i = 100; % bulk liquid concentrations for A (mol/L)

coefa = zeros((L+Lf)/step + 1,(L+Lf)/step + 1); % preallocate matrix of coefficients
dim = size(coefa);
diag_1 = 1:(dim(1)+1):(dim(1)*dim(2)); % indices for middle diagonal
diag_2 = dim(1)+1:(dim(1)+1):(dim(1)*dim(2)); % indices for upper diagonal
diag_3 = 2:(dim(1)+1):(dim(1)*dim(2)); % indices for lower diagonal

% 0 < X < L
```

Robert Heeter
BIOE 391 Numerical Methods – Due 22 April 2022

```
coefa(diag_1(1:(L/step))) = -2*D/(step^2); % set values for tridiagonal matrix
coefa(diag_2(1:(L/step))) = D/(step^2);
coefa(diag_3(1:((L/step)-1))) = D/(step^2);

% L < X < (L + Lf)
coefa(diag_1(((L/step)+2):end)) = (-2*Df/(step^2)) - k;
coefa(diag_2(((L/step)+2):end)) = Df/(step^2);
coefa(diag_3(((L/step)+1):end)) = Df/(step^2);

% X = 0, L, L+Lf
coefa(1,1:2) = [(2*D/(step^2)) (-1*D/(step^2))]; % boundary/transition conditions
coefa(9,8:10) = [(D/step) ((-1*(D+Df)/step) - (k*step/2)) (Df/step)];
coefa(end,end-1:end) = [(-2*Df/(step^2)) ((2*Df/(step^2)) + k)];

b = zeros((L+Lf)/step + 1,1); % set values for constant column vector
b(1) = D*ca_i/(step^2); % boundary conditions
b(end) = 0;

ca_comp = gaussseidel(coefa,b); % use gaussseidel to solve coefa*ca_comp = b for all concentrations
x = 0:step:(L+Lf);

figure
plot(x,ca_comp,'-m','LineWidth',1.5);
grid on
xlabel('Distance (x) [cm]','FontSize',12,'FontWeight','bold');
ylabel('Concentration of compound A [M]','FontSize',12,'FontWeight','bold');
title('Compound A diffusion and reaction along diffusion layer and
biofilm','FontSize',14,'FontWeight','bold');

xca_comp = [x(:) ca_comp(:)]'; % combine results into one matrix
fprintf(' x (cm)    c_a (M)\n');
fprintf('%7.4f  %8.4f\n',xca_comp);
fprintf('\n');


%% P6. PROBLEM 22.22
disp('P6. PROBLEM 24.22');

% ATTEMPTED BUT DID NOT FINISH
coefa = zeros(L/step - 1,L/step - 1); % preallocate matrix of coefficients
dim = size(coefa);
diag_1 = 1:(dim(1)+1):(dim(1)*dim(2)); % indices for middle diagonal
diag_2 = dim(1)+1:(dim(1)+1):(dim(1)*dim(2)); % indices for upper diagonal
diag_3 = 2:(dim(1)+1):(dim(1)*dim(2)); % indices for lower diagonal

coefa(diag_1) = 2 + ((k/D)*step^2); % set values for tridiagonal matrix
coefa(diag_2) = -1;
coefa(diag_3) = -1;

b = zeros(L/step - 1,1); % set values for constant column vector
b(1) = A_i;
b(end) = A_f;

A_int = gaussseidel(coefa,b); % use gaussseidel to solve coef*A_int = b for internal concentrations
A_comp = [A_i; A_int(:); A_f]; % complete concentration vector including boundary conditions
x = 0:step:L;


%% Additional Functions

function XS = bacteria(~,I)
% ABOUT: Differential equation system for P#1; X = I(1) = bacterial
% biomass, S = I(2) = substrate concentration

Y = 0.75; % yield coefficient
k_max = 0.3; % maximum baterial growth rate
```

Robert Heeter


```
k_s = 1e-4; % half saturation constant


XS = [Y*k_max*I(2)*I(1)/(k_s+I(2)); -1*k_max*I(2)*I(1)/(k_s+I(2))];

end


function Cs = concentrations(~,c,k)
% ABOUT: Differential equation system for P#2; c = vector of concentration
% of reactions 1-3, k = vector of constants for reactions 1-3

Cs = [-1*k(1)*c(1) + k(2)*c(2) + k(3)*c(3); k(1)*c(1) - k(2)*c(2) - k(4)*c(2); k(4)*c(2) - k(3)*c(3)];

end


function SSR = conc_SSR(k,cdata)
% ABOUT: Sums squares of discrepances between model predictions and data
% for P#2

tspan = 1:15; % time interval (days)
ivals = [100 0 0]; % inital values c1(0) = 100, c2(0) = c3(0) = 0

[~,c] = ode45(@concentrations,tspan,ivals,[],k); % use ode45 to solve system of differential equations
in function @concentration

R = (c([1:6,8:10,12,15],:)-cdata).^2; % calculate sum of squares of differences
SSR = sum(R,'all');

end


function x = gaussseidel(A,b,es,maxit)
% ABOUT: Gauss-Seidel method from textbook .m file.
% INPUTS: A = coefficient matrix; b = right side vector; es = relative
% error threshold (default = 0.00001%); maxit = max iterations (default =
% 50)
% OUTPUTS: x = solution vector

if nargin < 2
    error('At least 2 input arguments required.')
end
if nargin<4 || isempty(maxit)
    maxit=50;
end
if nargin<3 || isempty(es)
    es=0.00001;
end

[m,n] = size(A);
if m~=n
    error('Matrix A must be square.');
end

C = A;
x = zeros(1,n);
d = zeros(1,n);

for i = 1:n
  C(i,i) = 0;
end
x = x';
for i = 1:n
  C(i,1:n) = C(i,1:n)/A(i,i);
end
```

Robert Heeter
BIOE 391 Numerical Methods – Due 22 April 2022

```
for i = 1:n
  d(i) = b(i)/A(i,i);
end

iter = 0;
ea = 100;
while (1)
  xold = x;
  for i = 1:n
    x(i) = d(i)-C(i,:)*x;
    if x(i) ~= 0
      ea = abs((x(i) - xold(i))/x(i)) * 100;
    end
  end
  iter = iter+1;
  if max(ea)<=es || iter >= maxit
      break
  end
end

end
```