22S-CSB150-EEB159 Lab6

LISA WANG

TOTAL POINTS

94 / 100

QUESTION 1

1 Ex.1 **19 / 25**

- **- 0 pts** Correct
- **- 6 pts** `dsdt = -K1 * e * s + K1MINUS * c;`
- **- 6 pts** `dedt = -K1 * e * s + (K1MINUS + K2) * c;`
- ✓ **- 6 pts** *`dcdt = K1 * e * s - (K1MINUS + K2) * c;`*
- **- 6 pts** `dpdt = K2 * c;`

there is still an equation for d[ES]/dt

QUESTION 2

2 Ex.2 **25 / 25**

- ✓ **- 0 pts** *Correct*
- **- 10 pts** Correctly complete the `QSSA_func.m` code (equations):

```matlab
dsdt = -K1 * e * s + K1MINUS * c;
dedt = -K1 * e * s + (K1MINUS + K2) * c;
dcdt = K1 * e * s - (K1MINUS + K2) * c;
dpdt = K2 * c;
```

- **- 5 pts** Correctly complete `enzykmekin.m` code (ICs and parameters):

```matlab
K1 = 600;
K1MINUS = 120;
K2 = 1.2;
% Initial conditions x0.
s0 = 1;
e0 = 1;
c0 = 0;
p0 = 0;
```

- **- 10 pts** Generate the correct plot
- **- 2 pts** Plot aesthetics (graph/axes titles)
- **- 2 pts** You have the correct plot, but you need to show your modified functions

QUESTION 3

3 Ex.3 **25 / 25**

- ✓ **- 0 pts** *Correct*
- **- 5 pts** Correctly modify `QSSA_func` file and ICs and plot attributes in `enzymekin` file to accommodate the 2 extra variables

```matlab
% ----sQSSA----
% Constraints
% e(t) + c(t) = e0
% Assumption
% dcdt ~ 0
s_qssa = x(5); p_qssa = x(6);
Km = (K1MINUS + K2)/K1;
c_qssa = s_qssa*e0/(s_qssa + Km);
ds_qssadt = -K2*c_qssa;
dp_qssadt = K2*c_qssa;
```

% Pack output

dxdt=[dsdt;dedt;dcdt;dpdt;ds_qssadt;dp_qssadt];

return

```

- **- 5 pts** Correct plot
- **- 2 pts** Plot aesthetics (graph/axes titles)
- **- 25 pts** Not attempted

QUESTION 4

## *4* Ex.4 **25 / 25**

✓ **- 0 pts** *Correct*

- **- 5 pts** Identify and discuss one condition for which QSSA is valid
- **- 5 pts** Identify and discuss one condition for which QSSA is not valid
- **- 5 pts** Plot for valid case
- **- 5 pts** Plot for invalid case
- **- 25 pts** not attempted
```

```
% Lisa Wang Lab 6
% UID: 105502901
%
% Exercise 1 answers:
% dS/dt = -K1*e*s + K1MINUS * c
% dE/dt = -K1*e*s + K1MINUS * c + K2 * c
% dP/dt = K2* c
%
% The ES complex is an intermediate, not a product or an reactant
```

```
%Exercise 2
% THIS SCRIPT SIMULATES MICHAELIS-MENTEN ENZYME DYNAMICS
% Standard Quasi Steady State Approximation sQSSA
% Calls QSSA_func.m which implements the differential eqns
% Reactions
%     R1: S + E -- k1 --> ES
%     R2: ES    -- kminus1 --> E + S
%     R3: ES    -- k2 --> E + P
%
% Also, solves for set of equations with sQSSA applied.
% Constraints
%     e(t) + c(t) = e0
% Assumption
%     dcdt ~ 0

% Converted to minutes
K1      = 600;
K1MINUS = 120;
K2      = 1.2;

% Initial conditions x0.
s0 = 1;
e0 = 1;
c0 = 0;
p0 = 0;

% First 4 elements are for the original set of equations
% Last 2 elements are for sQSSA.
% [s_ori e_ori c_ori p_ori s_qssa p_qssa]
x0 = [s0; e0; c0; p0];  % ICs

% Calculate QSSA metrics
Km = (K1MINUS+K2)./K1;
tc = 1./(K1*(s0+Km));
ts = (s0+Km)./(e0*K2);
epsilon = e0/(Km+s0);

disp(sprintf('\n----------Running MM Script-----------\n'));
```

1 Ex.1 **19 / 25**

- **- 0 pts** Correct
- **- 6 pts** `dsdt = -K1 * e * s + K1MINUS * c;`
- **- 6 pts** `dedt = -K1 * e * s + (K1MINUS + K2) * c;`
- ✓ **- 6 pts** *`dcdt = K1 * e * s - (K1MINUS + K2) * c;`*
- **- 6 pts** `dpdt = K2 * c;`

there is still an equation for d[ES]/dt

```
% Lisa Wang Lab 6
% UID: 105502901
%
% Exercise 1 answers:
% dS/dt = -K1*e*s + K1MINUS * c
% dE/dt = -K1*e*s + K1MINUS * c + K2 * c
% dP/dt = K2* c
%
% The ES complex is an intermediate, not a product or an reactant
```

```
%Exercise 2
% THIS SCRIPT SIMULATES MICHAELIS-MENTEN ENZYME DYNAMICS
% Standard Quasi Steady State Approximation sQSSA
% Calls QSSA_func.m which implements the differential eqns
% Reactions
%     R1: S + E -- k1 --> ES
%     R2: ES    -- kminus1 --> E + S
%     R3: ES    -- k2 --> E + P
%
% Also, solves for set of equations with sQSSA applied.
% Constraints
%     e(t) + c(t) = e0
% Assumption
%     dcdt ~ 0

% Converted to minutes
K1      = 600;
K1MINUS = 120;
K2      = 1.2;

% Initial conditions x0.
s0 = 1;
e0 = 1;
c0 = 0;
p0 = 0;

% First 4 elements are for the original set of equations
% Last 2 elements are for sQSSA.
% [s_ori e_ori c_ori p_ori s_qssa p_qssa]
x0 = [s0; e0; c0; p0];  % ICs

% Calculate QSSA metrics
Km = (K1MINUS+K2)./K1;
tc = 1./(K1*(s0+Km));
ts = (s0+Km)./(e0*K2);
epsilon = e0/(Km+s0);

disp(sprintf('\n----------Running MM Script-----------\n'));
```

```
----------Running MM Script-----------
```

```
disp(sprintf('Intial substrate: %f\nInitial Enzyme: %f\n',s0,e0));
```

```
Intial substrate: 1.000000
Initial Enzyme: 1.000000
```

```
disp(sprintf('KM: %f\nFast time tc: %f\nSlow time ts: %f\n',Km,tc,ts));
```

```
KM: 0.202000
Fast time tc: 0.001387
Slow time ts: 1.001667
```

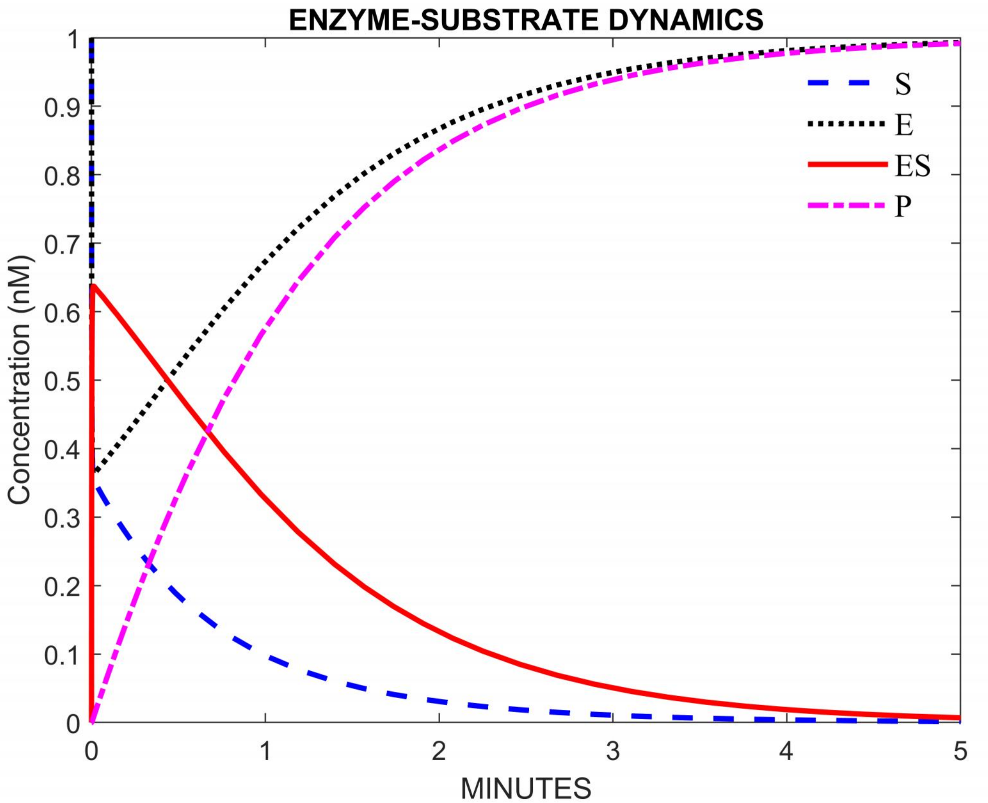
```
disp(sprintf('tc/ts= %f\n',tc/ts));
```

```
tc/ts= 0.001384
```

```
disp(sprintf('epsilon=e0/(KM+s0): %f\n',epsilon));
```

```
epsilon=e0/(KM+s0): 0.831947
```

```

% Call Matlab Numerical Solver on our function f(t,x)
tspan = [0 floor(5*ts)]; %scale simulation time to slow time
%tspan = [0 0.05]; % trying to use a different time span
[t,x]=ode15s(@(t,x) QSSA_func(t,x,K1,K1MINUS,K2,e0), tspan, x0);

% Post Process
h = plot(t,x);

set(h,{'LineWidth'},{2;2;2;2},{'LineStyle'},...
    {'--';':';'-';'-.'},{'Color'},{'b';'k';'r';'m'});
title(sprintf('ENZYME-SUBSTRATE DYNAMICS'));
xlabel('MINUTES');
ylabel('Concentration (nM)');
legend({'S','E','ES','P'},'FontName','Times','FontSize',12);
legend('boxoff')
```




```
% Converted to minutes
K1      = 600;
K1MINUS = 120;
K2      = 1.2;

% Initial conditions x0.
s0 = 1;
e0 = 1;
c0 = 0;
p0 = 0;
s_QSSA0 = 1;
p_QSSA0 = 0;


% First 4 elements are for the original set of equations
% Last 2 elements are for sQSSA.
% [s_ori e_ori c_ori p_ori s_qssa p_qssa]
x0 = [s0; e0; c0; p0; s_QSSA0; p_QSSA0];  % ICs

% Calculate QSSA metrics
Km = (K1MINUS+K2)./K1;
tc = 1./(K1*(s0+Km));
ts = (s0+Km)./(e0*K2);
epsilon = e0/(Km+s0);
```

```
disp(sprintf('\n----------Running MM Script-----------\n'));
```

```
----------Running MM Script-----------
```

```
disp(sprintf('Intial substrate: %f\nInitial Enzyme: %f\n',s0,e0));
```

```
Intial substrate: 1.000000
Initial Enzyme: 1.000000
```

```
disp(sprintf('KM: %f\nFast time tc: %f\nSlow time ts: %f\n',Km,tc,ts));
```

```
KM: 0.202000
Fast time tc: 0.001387
Slow time ts: 1.001667
```

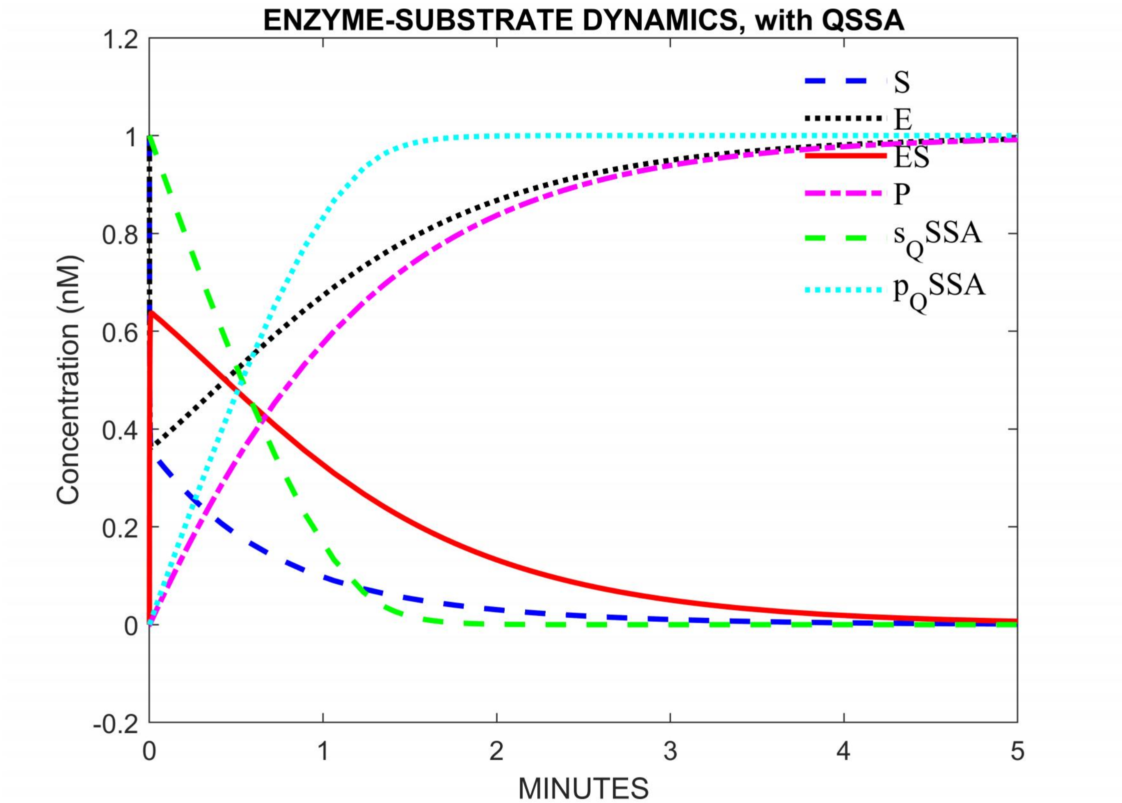
```
disp(sprintf('tc/ts= %f\n',tc/ts));
```

```
tc/ts= 0.001384
```

```
disp(sprintf('epsilon=e0/(KM+s0): %f\n',epsilon));
```

```
epsilon=e0/(KM+s0): 0.831947
```

```
% Call Matlab Numerical Solver on our function f(t,x)
tspan = [0 floor(5*ts)]; %scale simulation time to slow time
%tspan = [0 0.05]; % trying to use a different time span
[t,x]=ode15s(@(t,x) sQSSA_func(t,x,K1,K1MINUS,K2,e0,Km), tspan, x0);

%Post Process
h = plot(t,x);

set(h,{'LineWidth'},{2;2;2;2;2;2},{'LineStyle'},...
    {'--';':';'-';'-.'; '--'; ':'},{'Color'},{'b';'k';'r';'m'; 'g'; 'c'});
title(sprintf('ENZYME-SUBSTRATE DYNAMICS, with QSSA'));
xlabel('MINUTES');
ylabel('Concentration (nM)');
legend({'S','E','ES','P', 's_QSSA', 'p_QSSA'},'FontName','Times','FontSize',12);
legend('boxoff')
```




```
%%Exercise 3 Question:
%% Implement the new equations
% in addition to the four original equations,
%% don't forget the update the initial condition vector appropriately.
% Plot all 6 solutions (s, e, c, p, sqssa, pqssa).

% lecture 8 dSqssa and dpqssa are derived from assumption that dP_qssa/dt
% >> d[ES]/dt, and d[ES]/dt is approximately zero. Then we can say that
% based on this assumption, we say that Km = Kd and that Km = (K1MINUS+K2)./K1;
% We can also say that Kd is the concentration at half maximum rate
% the concentration of the complex is [ES] = (e0* s_qssa) / (Kd + s_qssa)
% then d[P]/dt =dp_qssadt = K2* ((e0 *s_qssa)/(Km + s_qssa))
% andd d[S]/dt = ds_qssadt = -ds_qssadt = -K2* ((e0 *s_qssa)/(Km +
% s_qssa));
% We can also write (Vmax * s_qssa) / (Km + s_qssa), when Vmax is given, in which
% case it is not given here.
% we know that this corresponding relationship is modeled correctly
% because as the s_qssa goes up, the p_ssa goes down.
```

Exercise 4

```
% One condition QSSA is valid is when K1MINUS is large
% for example, K1MINUS is 200 and E0 = S0 = 1. We can see that this affects the equation so tha
% since QSSA is valid when total enzyme is much smaller than the sum of Kd
% and original substrate level. It captures the trajectory of the MM
```

2 Ex.2 **25 / 25**

✓ **- 0 pts** *Correct*

- 10 pts Correctly complete the `QSSA_func.m` code (equations):

```matlab
dsdt = -K1 * e * s + K1MINUS * c;
dedt = -K1 * e * s + (K1MINUS + K2) * c;
dcdt = K1 * e * s - (K1MINUS + K2) * c;
dpdt = K2 * c;
```

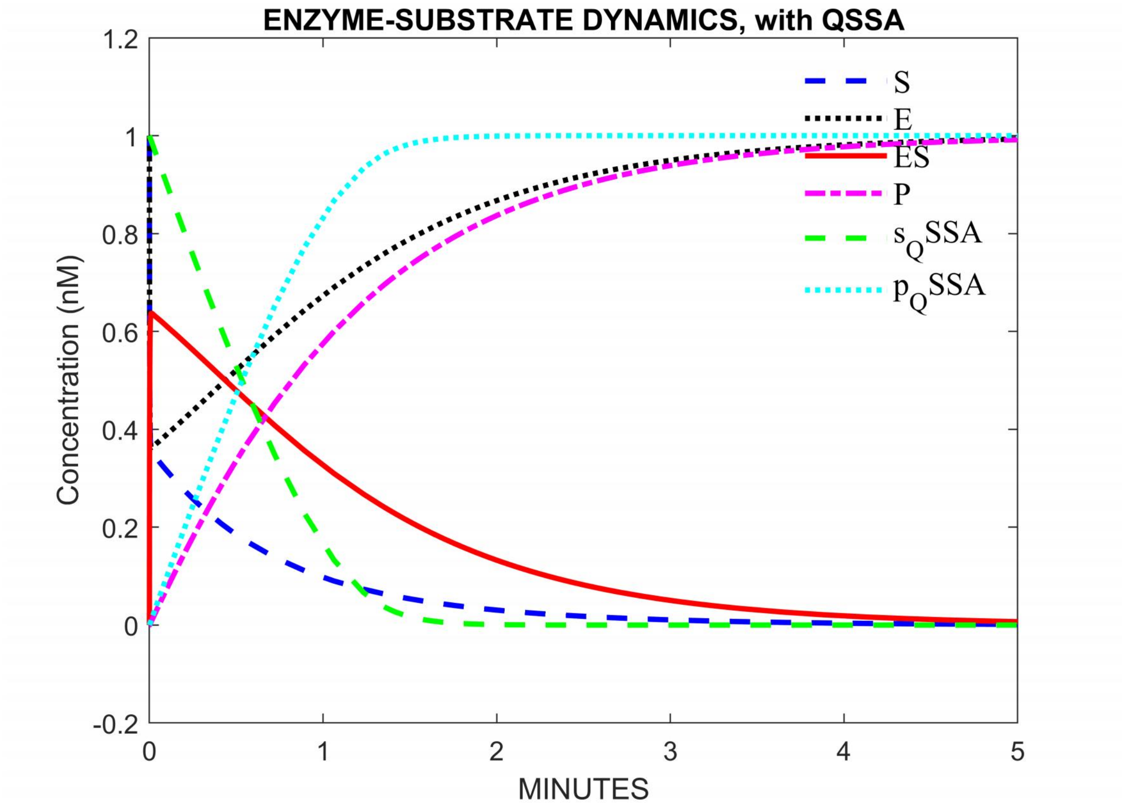
- 5 pts Correctly complete `enzykmekin.m` code (ICs and parameters):

```matlab
K1 = 600;
K1MINUS = 120;
K2 = 1.2;

% Initial conditions x0.
s0 = 1;
e0 = 1;
c0 = 0;
p0 = 0;
```

- 10 pts Generate the correct plot

- 2 pts Plot aesthetics (graph/axes titles)

- 2 pts You have the correct plot, but you need to show your modified functions




```
%%Exercise 3 Question:
%% Implement the new equations
% in addition to the four original equations,
%% don't forget the update the initial condition vector appropriately.
% Plot all 6 solutions (s, e, c, p, sqssa, pqssa).

% lecture 8 dSqssa and dpqssa are derived from assumption that dP_qssa/dt
% >> d[ES]/dt, and d[ES]/dt is approximately zero. Then we can say that
% based on this assumption, we say that Km = Kd and that Km = (K1MINUS+K2)./K1;
% We can also say that Kd is the concentration at half maximum rate
% the concentration of the complex is [ES] = (e0* s_qssa) / (Kd + s_qssa)
% then d[P]/dt =dp_qssadt = K2* ((e0 *s_qssa)/(Km + s_qssa))
% andd d[S]/dt = ds_qssadt = -ds_qssadt = -K2* ((e0 *s_qssa)/(Km +
% s_qssa));
% We can also write (Vmax * s_qssa) / (Km + s_qssa), when Vmax is given, in which
% case it is not given here.
% we know that this corresponding relationship is modeled correctly
% because as the s_qssa goes up, the p_ssa goes down.
```

Exercise 4

```
% One condition QSSA is valid is when K1MINUS is large
% for example, K1MINUS is 200 and E0 = S0 = 1. We can see that this affects the equation so tha
% since QSSA is valid when total enzyme is much smaller than the sum of Kd
% and original substrate level. It captures the trajectory of the MM
```

3 Ex.3 **25 / 25**

✓ **- 0 pts** *Correct*

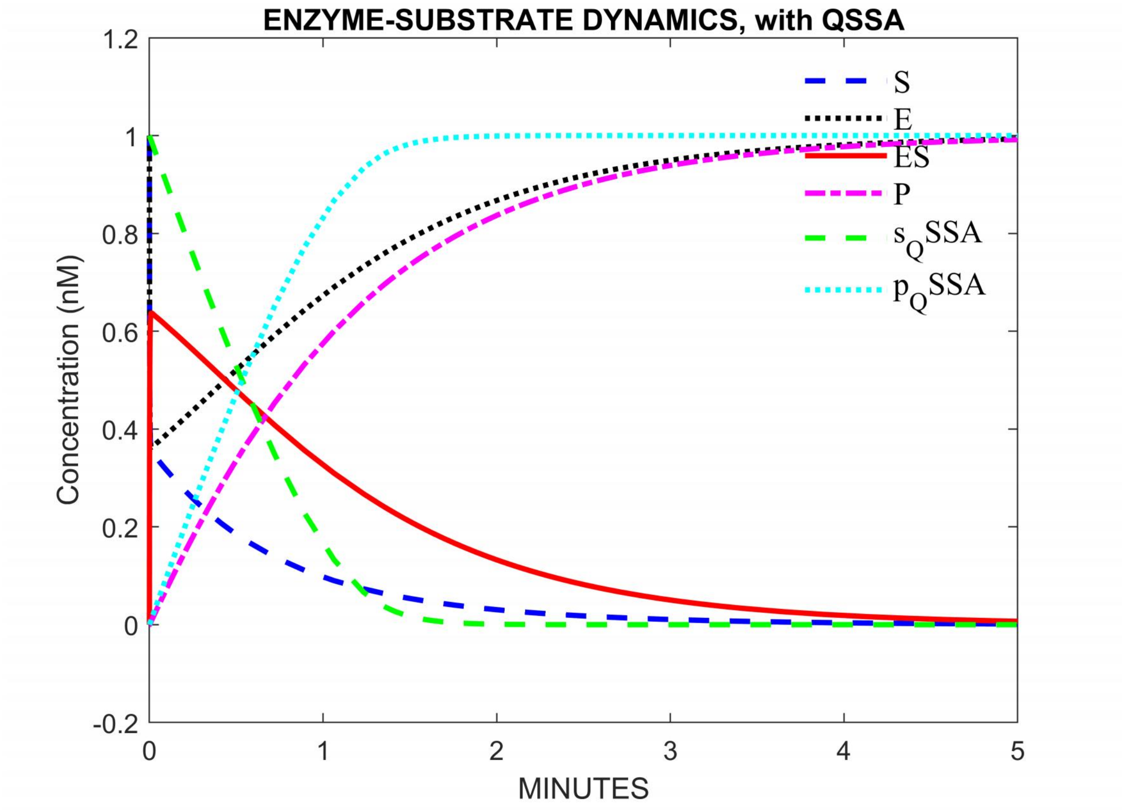
- 5 pts Correctly modify `QSSA_func` file and ICs and plot attributes in `enzymekin` file to accommodate the 2 extra variables

```matlab
% ----sQSSA----
% Constraints
% e(t) + c(t) = e0
% Assumption
% dcdt ~ 0
s_qssa = x(5); p_qssa = x(6);
Km = (K1MINUS + K2)/K1;
c_qssa = s_qssa*e0/(s_qssa + Km);
ds_qssadt = -K2*c_qssa;
dp_qssadt = K2*c_qssa;
% Pack output
dxdt=[dsdt;dedt;dcdt;dpdt;ds_qssadt;dp_qssadt];
return
```

- 5 pts Correct plot

- 2 pts Plot aesthetics (graph/axes titles)

- 25 pts Not attempted




```
%%Exercise 3 Question:
%% Implement the new equations
% in addition to the four original equations,
%% don't forget the update the initial condition vector appropriately.
% Plot all 6 solutions (s, e, c, p, sqssa, pqssa).

% lecture 8 dSqssa and dpqssa are derived from assumption that dP_qssa/dt
% >> d[ES]/dt, and d[ES]/dt is approximately zero. Then we can say that
% based on this assumption, we say that Km = Kd and that Km = (K1MINUS+K2)./K1;
% We can also say that Kd is the concentration at half maximum rate
% the concentration of the complex is [ES] = (e0* s_qssa) / (Kd + s_qssa)
% then d[P]/dt =dp_qssadt = K2* ((e0 *s_qssa)/(Km + s_qssa))
% andd d[S]/dt = ds_qssadt = -ds_qssadt = -K2* ((e0 *s_qssa)/(Km +
% s_qssa));
% We can also write (Vmax * s_qssa) / (Km + s_qssa), when Vmax is given, in which
% case it is not given here.
% we know that this corresponding relationship is modeled correctly
% because as the s_qssa goes up, the p_ssa goes down.
```

Exercise 4

```
% One condition QSSA is valid is when K1MINUS is large
% for example, K1MINUS is 200 and E0 = S0 = 1. We can see that this affects the equation so tha
% since QSSA is valid when total enzyme is much smaller than the sum of Kd
% and original substrate level. It captures the trajectory of the MM
```

```
% enzyme-substrate dynamics well.
%
%
% One condition that QSSA is invalid is when K1MINUS is small or of
% comparable magnitude as the total enzyme. For examplem, when K1MINUS is
% 2, E0 = S0 = 1, the sum of K1MINUS and S0 is not much greater than E0.
% The approximation is not as accurate because we do not meet our
% assumption E0 << Kd + s0
```

```
% poor approximation graph

% Converted to minutes
K1      = 600;
K1MINUS = 2;
K2      = 1.2;

% Initial conditions x0.
s0 = 1;
e0 = 1;
c0 = 0;
p0 = 0;
s_QSSA0 = 1;
p_QSSA0 = 0;

disp(sprintf('\n----------Running MM Script-----------\n'));
```

```
----------Running MM Script-----------
```

```
disp(sprintf('Intial substrate: %f\nInitial Enzyme: %f\n',s0,e0));
```

```
Intial substrate: 1.000000
Initial Enzyme: 1.000000
```

```
disp(sprintf('KM: %f\nFast time tc: %f\nSlow time ts: %f\n',Km,tc,ts));
```

```
KM: 0.202000
Fast time tc: 0.001387
Slow time ts: 1.001667
```

```
disp(sprintf('tc/ts= %f\n',tc/ts));
```

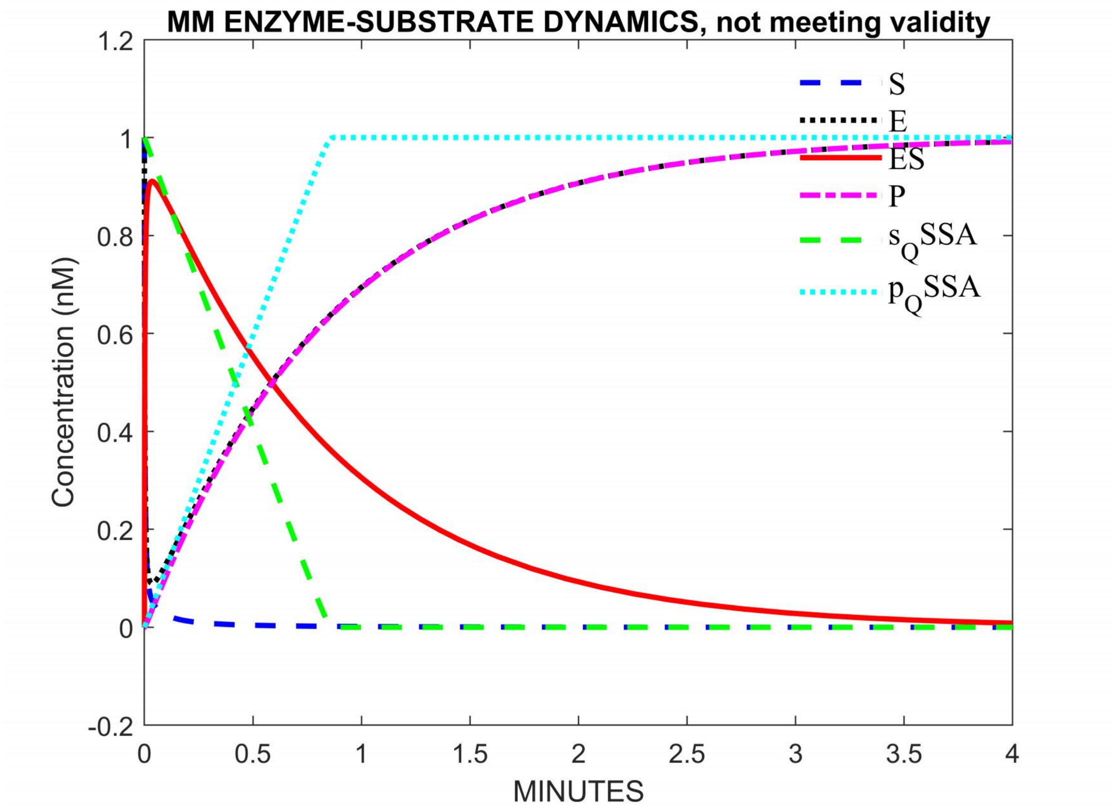
```
tc/ts= 0.001384
```

```
disp(sprintf('epsilon=e0/(KM+s0): %f\n',epsilon));
```

```
epsilon=e0/(KM+s0): 0.831947
```

```


% First 4 elements are for the original set of equations
% Last 2 elements are for sQSSA.
% [s_ori e_ori c_ori p_ori s_qssa p_qssa]
x0 = [s0; e0; c0; p0; s_QSSA0; p_QSSA0];  % ICs
```

```
% Calculate QSSA metrics
Km = (K1MINUS+K2)./K1;
tc = 1./(K1*(s0+Km));
ts = (s0+Km)./(e0*K2);
epsilon = e0/(Km+s0);

% Call Matlab Numerical Solver on our function f(t,x)
tspan = [0 floor(5*ts)]; %scale simulation time to slow time
%tspan = [0 0.05]; % trying to use a different time span
[t,x]=ode15s(@(t,x) sQSSA_func(t,x,K1,K1MINUS,K2,e0,Km), tspan, x0);

%Post Process
h = plot(t,x);

set(h,{'LineWidth'},{2;2;2;2;2;2},{'LineStyle'},...
    {'--';':';'-';'-.'; '--'; ':'},{'Color'},{'b';'k';'r';'m'; 'g'; 'c'});
title(sprintf('MM ENZYME-SUBSTRATE DYNAMICS, not meeting validity'));
xlabel('MINUTES');
ylabel('Concentration (nM)');
legend({'S','E','ES','P', 's_QSSA', 'p_QSSA'},'FontName','Times','FontSize',12);
legend('boxoff')
```




```
% good approximation graph

% Converted to minutes
```

```
K1       = 600;
K1MINUS = 8000;
K2       = 1.2;

% Initial conditions x0.
s0 = 1;
e0 = 1;
c0 = 0;
p0 = 0;
s_QSSA0 = 1;
p_QSSA0 = 0;

disp(sprintf('\n----------Running MM Script-----------\n'));
```

```
----------Running MM Script-----------
```

```
disp(sprintf('Intial substrate: %f\nInitial Enzyme: %f\n',s0,e0));
```

```
Intial substrate: 1.000000
Initial Enzyme: 1.000000
```

```
disp(sprintf('KM: %f\nFast time tc: %f\nSlow time ts: %f\n',Km,tc,ts));
```

```
KM: 0.005333
Fast time tc: 0.001658
Slow time ts: 0.837778
```

```
disp(sprintf('tc/ts= %f\n',tc/ts));
```

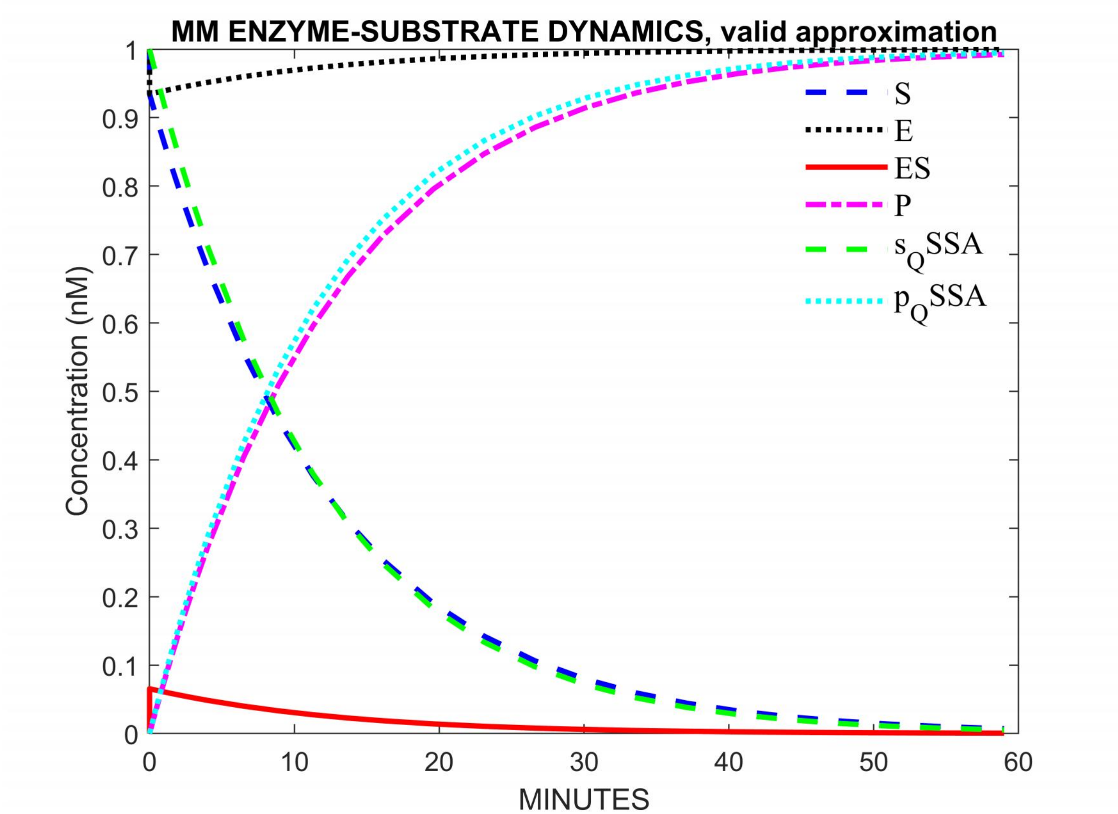
```
tc/ts= 0.001979
```

```
disp(sprintf('epsilon=e0/(KM+s0): %f\n',epsilon));
```

```
epsilon=e0/(KM+s0): 0.994695
```

```

% First 4 elements are for the original set of equations
% Last 2 elements are for sQSSA.
% [s_ori e_ori c_ori p_ori s_qssa p_qssa]
x0 = [s0; e0; c0; p0; s_QSSA0; p_QSSA0];  % ICs

% Calculate QSSA metrics
Km = (K1MINUS+K2)./K1;
tc = 1./(K1*(s0+Km));
ts = (s0+Km)./(e0*K2);
epsilon = e0/(Km+s0);

% Call Matlab Numerical Solver on our function f(t,x)
tspan = [0 floor(5*ts)]; %scale simulation time to slow time
%tspan = [0 0.05]; % trying to use a different time span
[t,x]=ode15s(@(t,x) sQSSA_func(t,x,K1,K1MINUS,K2,e0,Km), tspan, x0);

%Post Process
h = plot(t,x);
```

```
set(h,{'LineWidth'},{2;2;2;2;2;2},{'LineStyle'},...
    {'--';':';'-';'-.'; '--'; ':'},{'Color'},{'b';'k';'r';'m'; 'g'; 'c'});
title(sprintf('MM ENZYME-SUBSTRATE DYNAMICS, valid approximation'));
xlabel('MINUTES');
ylabel('Concentration (nM)');
legend({'S','E','ES','P', 's_QSSA', 'p_QSSA'},'FontName','Times','FontSize',12);
legend('boxoff')
```




```
function [dxdt] = sQSSA_func(t, x, K1, K1MINUS, K2, e0, Km)
s = x(1); e = x(2); c = x(3); p = x(4); s_qssa = x(5); p_qssa = x(6);
dsdt = -K1*e*s + K1MINUS*c;
dedt = -K1*e*s + (K1MINUS*c) + K2*c;
dcdt = K1*e*s - (K1MINUS*c) - K2*c;
%dcdt = 0;
dpdt = K2*c;
ds_qssadt = -K2* ((e0 *s_qssa)/(Km + s_qssa)); % ds goes up when dp goes down
dp_qssadt = K2* ((e0 *s_qssa)/(Km + s_qssa));

dxdt = [dsdt; dedt; dcdt; dpdt; ds_qssadt; dp_qssadt];
return
end

function [dxdt] = QSSA_func(t,x,K1,K1MINUS,K2,e0)
```

```
% Single Substrate-Enzyme Reaction
% R1: S + E -- k1 --> ES
% R2: ES    -- k1minus --> E + S
% R3: ES    -- k2 --> E + P

% ----Original----
s=x(1); e=x(2); c=x(3); p=x(4);
dsdt = -K1*e*s + K1MINUS*c;
dedt = -K1*e*s + (K1MINUS*c) + K2*c;
dcdt = K1*e*s - (K1MINUS*c) - K2*c;
%dcdt = 0;
dpdt = K2*c;

% ----sQSSA----
% Constraints
%     e(t) + c(t) = e0
% Assumption
%     dcdt ~ 0

%s_qssa = x(5); p_qssa = x(6);

% Pack output
dxdt=[dsdt;dedt;dcdt;dpdt];
return
end
```

4 Ex.4 **25 / 25**

✓ **- 0 pts** *Correct*

- 5 pts Identify and discuss one condition for which QSSA is valid

- 5 pts Identify and discuss one condition for which QSSA is not valid

- 5 pts Plot for valid case

- 5 pts Plot for invalid case

- 25 pts not attempted