

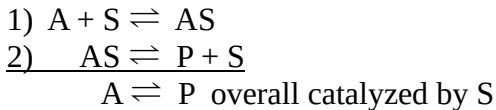
## The Steady-State Approximation (SSA) applied to a surface-catalyzed reaction

Richard K. Herz, rherz@ucsd.edu, May, 2015

There are two major approximations that can be made when simplifying reaction kinetic equations: the partial equilibrium approximation (PEA) applied to an elementary step, and the steady-state approximation (SSA) applied to a reactive species present in low concentration. For homogeneous reactions, the SSA is often applied to radicals, for example, to OH radicals in combustion reactions.

Here we show the SSA can be applied to surface species in a surface-catalyzed reaction.

Consider the following reaction mechanism over a catalyst surface. A and P are fluid-phase components.



Steady-state reactors are at steady-state by specification, so the SSA would be applied to an unsteady-state reactor. Here we write equations for an isothermal batch reactor.

There are four components (A, S, AS, P) so we could write four balance equations. Since the total number of sites is constant, we can use the site balance equation to eliminate one of the balances on surface sites.

When there is significant capacity of the surface to adsorb and accumulate A, we must write separate balances for A and P. When the capacity of the surface is relatively small, we can write a balance only for A and then get P from a stoichiometric table.

Here are the three ODEs for A, P and AS.

$$\frac{d\psi_A}{dt} = \alpha(-k'_1\psi_A(1-\theta_A) + k_{-1}\theta_A)$$

$$\frac{d\psi_P}{dt} = \alpha(k_2\theta_A - k'_{-2}\psi_P(1-\theta_A))$$

$$\frac{d\theta_A}{dt} = k'_1\psi_A(1-\theta_A) - k_{-1}\theta_A - k_2\theta_A + k'_{-2}\psi_P(1-\theta_A)$$

where

$$\psi_A = \frac{C_A}{C_{A,max}} \quad ; \quad \psi_P = \frac{C_P}{C_{A,max}} \quad ; \quad \theta_A = \frac{C_{AS}^s}{C_{tot}^s} \quad ; \quad k'_1 = C_{A,max} k_1 \quad ; \quad k'_{-2} = C_{A,max} k_{-2}$$

and where the surface-to-fluid capacity ratio is

$$\alpha = \frac{W(\text{kg}) C_{\text{tot}}^{\text{s}}(\text{mol surf sites/kg})}{V(\text{m}^3) C_{A,\text{max}}(\text{mol/m}^3)}$$

After specifying three initial conditions, the three ODEs can be integrated.

Note that there is no rate equation for the overall reaction. We can only write rate equations for the individual steps at this stage.

For this surface reaction, we can apply the SSA to adsorbed species when  $\alpha$ , the surface-to-fluid capacity ratio, is small. The SSA can also be justified by comparison of time constants but the justifications are equivalent.

See the blackboard notes for a detailed justification of the SSA: divide the  $(d\theta_A/dt)$  equation by the  $(1/\alpha)(d\psi_A/dt)$  equation and do an order-of-magnitude analysis.

The "quick way" is to set the time derivative of the third ODE above to zero when  $\alpha \ll 1$

$$\frac{d\theta_A}{dt} \approx 0 = k'_1 \psi_A (1 - \theta_A) - k_{-1} \theta_A - k_2 \theta_A + k'_{-2} \psi_P (1 - \theta_A)$$

This doesn't mean that  $\theta_A$  doesn't change with time. In fact, the SSA means that  $\theta_A$  changes very fast relative to the other variables such that its value can be computed from the current values of the other variables as if it instantaneously reaches a steady state relative to those other variables.

This eliminates this ODE and allows us to solve for  $\theta_A$  in terms of the dimensionless fluid-phase concentrations.

In addition, since the amount of A that can be adsorbed and accumulated over the surface is small compared to the amount of A and P in the fluid phase, we can additionally eliminate the ODE for P and solve for P using a stoichiometric table.

For the initial conditions  $\psi_A(t=0)=1; \psi_P(t=0)=0$  we get, after a lot of algebra:

$$\frac{d\psi_A}{dt} = r_A$$

where

$$r_A = \frac{-k(\psi_A - \psi_{A,EQ})}{1 + K\psi_A} \ll \text{an algebraic "rate equation"}$$

$$k = \frac{\alpha(k'_1 k_2 + k_{-1} k'_{-2})}{k_{-1} + k_2 + k'_{-2}} \quad ; \quad K = \frac{k'_1 - k'_{-2}}{k_{-1} + k_2 + k'_{-2}}$$

$$K_{EQ} = \frac{k_1 k_2}{k_{-1} k_{-2}} = \frac{k'_1 k_2}{k_{-1} k'_{-2}} \ll \text{equilibrium constant for overall reaction stoichiometry}$$

$$\psi_{A, EQ} = \frac{1}{1 + K_{EQ}} \ll \text{equilibrium concentration for these initial conditions}$$

We are back to a familiar situation. We have one ODE and a rate equation for a single reaction in a batch reactor.

We can only get this situation for a surface-catalyzed reaction when the surface-to-fluid capacity ratio is small. For larger surface capacity, we have to go back to the original set of ODEs.

The SSA and PEA can be applied separately or together.

You can look at the definitions for  $k$ ,  $K$  and  $\psi_{A, EQ}$  and see how additional approximations would affect the rate equation. For example, consider a case where step 1 is in close approach to equilibrium, i.e., apply the PEA to step 1. Or consider a case when step 2 is far from equilibrium, i.e., an "essentially irreversible" reaction.

I've spared you the algebra. That is, I've spared myself the effort of typing the algebra here!

Here are two recommendations if you need to write equations and do algebra yourself.

First recommendation: Write out SI units for all variables and parameters at the start, and check at the end result, at least. Checking units will identify when you have forgotten something or included something extra.

Second recommendation: Write a computer program to check your results against other methods of solution, where checking against known and verified analytical solutions is preferred.

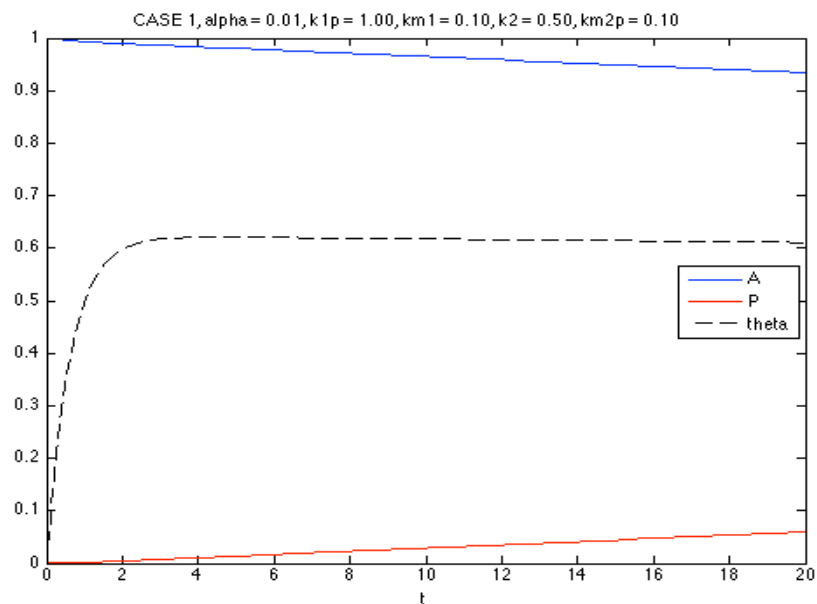
Below are results of a numerical solution of the reaction for several cases. The Matlab code is listed at the bottom of this document.

Case 1 is where the original set of three ODEs are integrated with no approximations applied.

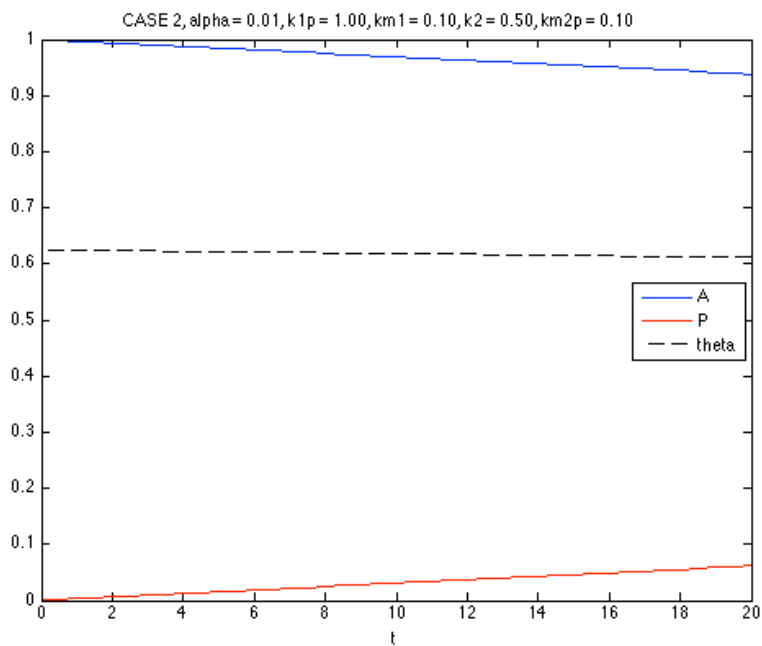
Cases 2 and 3 are where the SSA has been applied. Case 2 and 3 give the same result. Case 2 is for an earlier stage of the algebra. Case 3 is for the final stage of the algebra where the final algebraic rate equation was derived.

ALPHA = 0.01

CASE 1 - no approximations - 3 ODEs - to 20 seconds



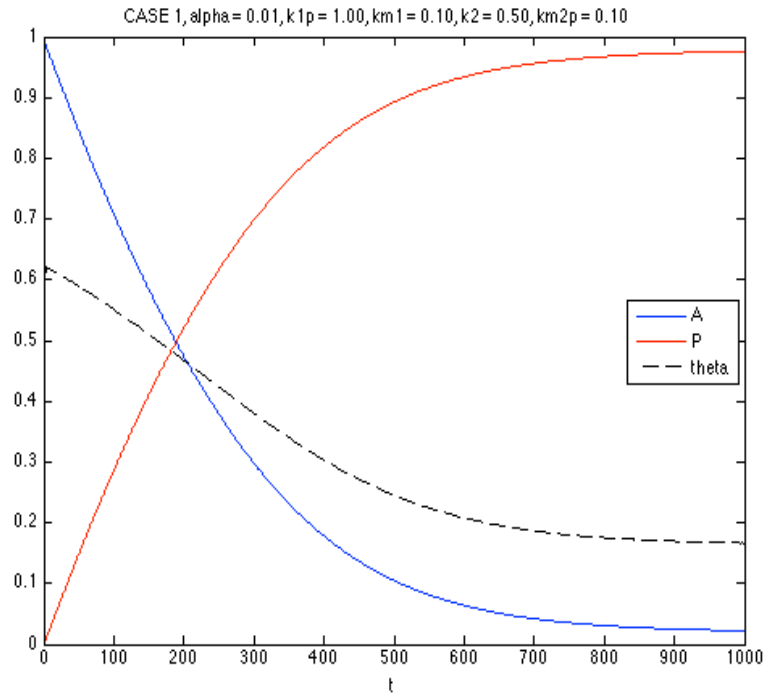
CASE 2 - apply SSA to adsorbed A - rate equation - to 20 seconds



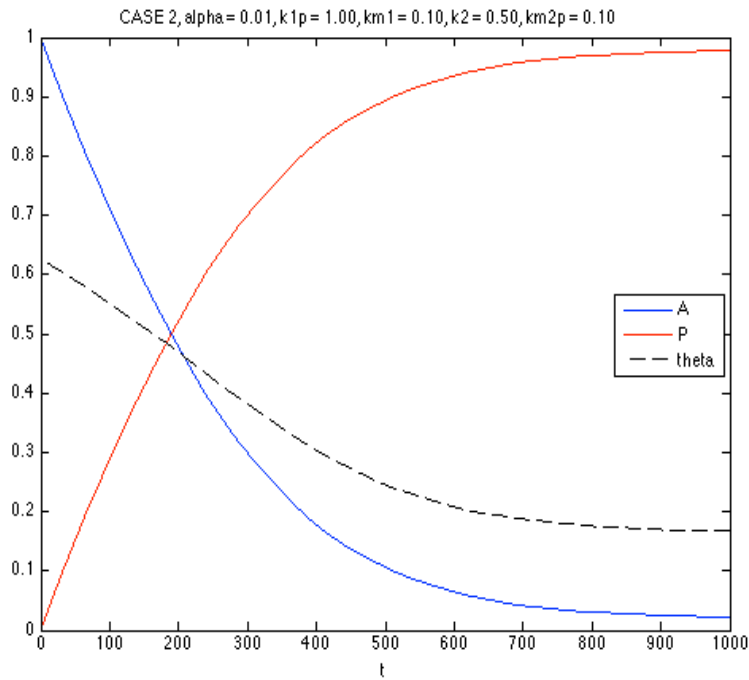
The SSA gives good results except for the surface coverage at early time.

ALPHA = 0.01

CASE 1 - no approximations - 3 ODEs - to 1000 seconds



CASE 2 - apply SSA to adsorbed A - rate equation - to 1000 seconds



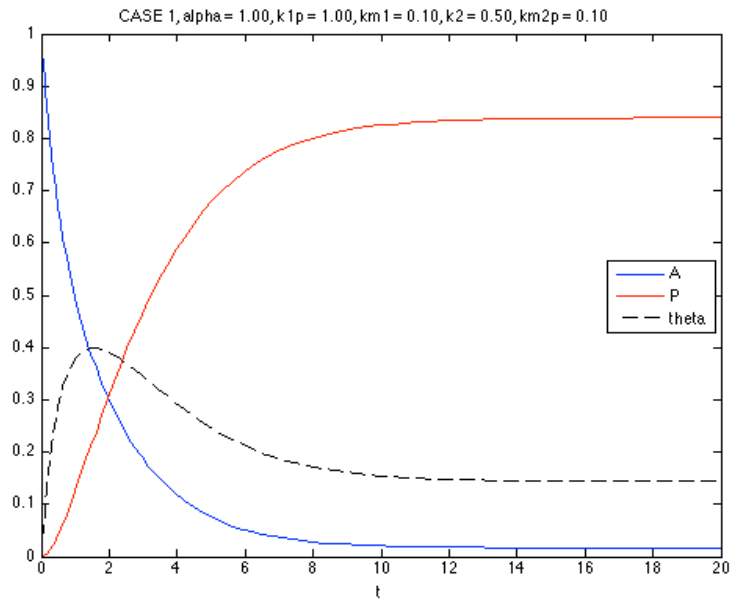
We conclude that SSA gives good results for alpha  $\ll$  1 over most of the time span.

Now, look what happens with alpha is NOT  $\ll 1$ .

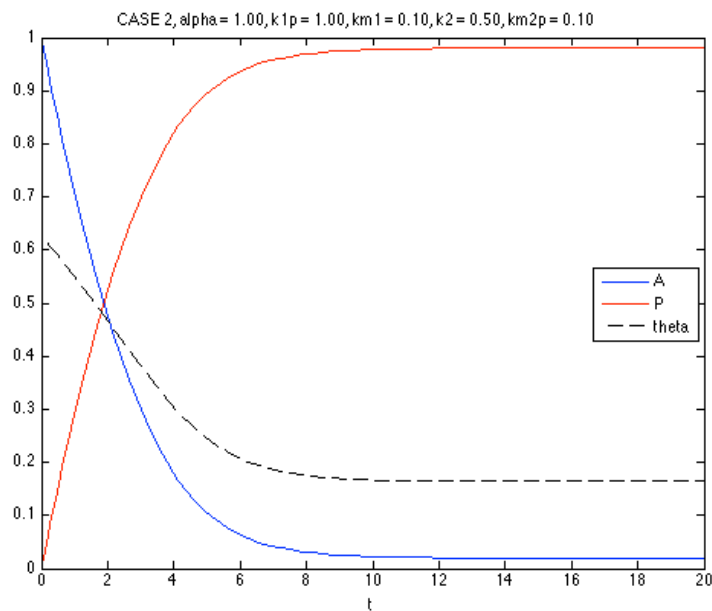
ALPHA = 1

CASE 1 - no approximations - 3 ODEs

Note that there is a significant quantity of A adsorbed over the large capacity surface at long time as equilibrium is approached. The sum of  $\psi_p + \psi_A$  is less than 1 because of storage of adsorbed A over the surface but the ratio ( $\psi_p / \psi_A$ ) is the equilibrium ratio, since a catalyst does not affect equilibrium.



CASE 2 - apply SSA to adsorbed A - rate equation - BAD HERE BECAUSE alpha not  $\ll 1$



## Listing of Matlab scripts

```
%% batch reactor with simple mechanism

% (1) A + S = AS
% (2) AS = P + S
% overall A = P catalyzed by S

clear all
close all

% alpha = (CtotS * W) / (Cmax * V), where CtotS (mol/kg) total surface sites,
% W (kg) mass cat, Cmax (mol/m3) max fluid conc, V (m3) volume fluid
alpha = 0.01; % d'less, surface-to-gas capacity ratio

k1 = 1; % (m3/mol/s), forward rxn coeffic for step 1
km1 = 0.1; % (1/s), reverse rxn coeffic for step 1 (k-1)
k2 = 0.5; % (1/s)
km2 = 0.1; % (m3/mol/s)
Cmax = 1; % (mol/m3), max fluid-phase conc
k1p = k1*Cmax; % (1/s)
km2p = km2*Cmax; % (1/s)

% Keq = (k1*k2)/(km1*km2) is equilibrium constant for overall reaction
% equilibrium conversion should not change with Cmax since no change in total
% moles in overall reaction stoichiometry for this reaction

% set params array to send to user-written function in ode45 calls
params = [alpha k1p km1 k2 km2p];

% y1 = psi-A, y2 = psi-P, y3 =theta-A

%% CASE 1

% no approximations applied, we have 3 ODEs
% and have no overall rate equation = function only of fluid-phase conc
% dy1/dt = alpha*(-k1p*y1*(1-y3) + km1*y3)
% dy2/dt = alpha*(k2*y3 - km2p*y2*(1-y3))
% dy3/dt = k1p*y1*(1-y3) - km1*y3 - k2*y3 + km2p*y2*(1-y3)

% integrate
y0 = [1 0 0]; % initial conditions
tspan = [0 1000]; % time span
[t y] = ode45('batchSimpleF1',tspan,y0,[],params);

y1 = y(:,1);
y2 = y(:,2);
y3 = y(:,3);

plot(t,y1,'b',t,y2,'r',t,y3,'k--')
tt = ...
    sprintf('CASE 1, alpha =%4.2f, k1p =%4.2f, km1 =%4.2f, k2 =%4.2f, km2p =
%4.2f',...
    alpha, k1p, km1, k2, km2p);
title(tt)
xlabel('t')
legend('A','P','theta','location','East')

%% CASE 2 - apply SSA to adsorbed reactant A

% when alpha << 1 we can apply SSA to theta-A
% now we get an overall rate equaton, r = function only of fluid-phase conc
% dy1/dt = r = alpha*(-k1p*y1*(1-y3) + km1*y3)
% get theta-A from y3 = (k1p*y1 + km2p*y2) / (k1p*y1 + km2p*y2 + km1 + k2)
```

```

% get y2 (psi-P) from stoichiometric table
% since, for alpha << 1, dy2/dt approx. = -dy1/dt
% that is, negligible accumulation of A adsorbed over catalyst
% stoichiometric table:
% y1 = y1(0) - extent
% y2 = y2(0) + extent
% y2 = y2(0) + y1(0) - y1
% for y2(0) = 0, y1(0) = 1, y2 = (1-y1)

% integrate
y0 = [1]; % initial conditions
% use same time span
[t y] = ode45('batchSimpleF2',tspan,y0,[],params);

y1 = y(:,1);
y2 = (1-y1); % for y1(t=0) = 1, y2(t=0) = 0
y3 = (k1p*y1 + km2p*y2) ./ (k1p*y1 + km2p*y2 + km1 + k2);

if alpha > 0.05
    fprintf('Case 2 does not apply since alpha NOT << 1 \n')
end

figure
plot(t,y1,'b',t,y2,'r',t,y3,'k--')
tt = ...
sprintf('CASE 2, alpha =%4.2f, k1p =%4.2f, km1 =%4.2f, k2 =%4.2f, km2p =%4.2f',...
    alpha, k1p, km1, k2, km2p);
title(tt)
xlabel('t')
legend('A','P','theta','location','East')

%% CASE 3 - SSA rate equation

% Cases 2 and 3 should give exactly the same results
% Case 3 is a further reduction of the algebraic results in Case 2

% group params for case of psi-A(t=0) = 1, psi-P(t=0) = 0
% y1(t=0) = 1, y2(t=0) = 0
C = km1 + k2 + km2p;
k = alpha*(k1p*k2+km1*km2p)/C;
K = (k1p-km2p)/C;
% KEQ = equilibrium constant of overall stoichiometry
KEQ = (k1*k2)/(km1*km2); % = (k1p*k2) / (km1*km2p) since p's cancel (*Cmax)
yEQ = 1/(1 + KEQ); % = (km1*km2p) / (k1p*k2 + km1*km2p) where p's cancel

% now use algebraic rate equation in batchSimpleF3 function for ode45
% for psi-A(t=0) = 1, psi-P(t=0) = 0
% rate = -k * (y - yEQ) / (1 + K*y);

% set params array to send to user-written function in ode45 calls
params = [k K yEQ];

% integrate
y0 = [1]; % initial conditions
% use same time span
[t y] = ode45('batchSimpleF3',tspan,y0,[],params);

y1 = y(:,1); % psi-A
y2 = (1-y1); % psi-P for y1(t=0) = 1, y2(t=0) = 0
y3 = (k1p*y1 + km2p*y2) ./ (k1p*y1 + km2p*y2 + km1 + k2); % theta-A

if alpha > 0.05
    fprintf('Case 3 does not apply since alpha NOT << 1 \n')
end

```



```

figure
plot(t,y1,'b',t,y2,'r',t,y3,'k--')
tt = ...
sprintf('CASE 3, alpha =%4.2f, k1p =%4.2f, km1 =%4.2f, k2 =%4.2f, km2p =%4.2f',...
alpha, k1p, km1, k2, km2p);
title(tt)
xlabel('t')
legend('A','P','theta','location','East')

```

### Listings of function files follows:

```

function dydt = batchSimpleF1(t,y,flags,params)

% params = [alpha k1p km1 k2 km2p];
alpha = params(1);
k1p = params(2);
km1 = params(3);
k2 = params(4);
km2p = params(5);

% y(1) = psi-A, y(2) = psi-P, y(3) = theta-A

dy1dt = alpha*(-k1p*y(1)*(1-y(3)) + km1*y(3));
dy2dt = alpha*(k2*y(3) - km2p*y(2)*(1-y(3)));
dy3dt = k1p*y(1)*(1-y(3)) - km1*y(3) - k2*y(3) + km2p*y(2)*(1-y(3));

dydt = [dy1dt ; dy2dt ; dy3dt]; % must return as column vector

```

```

function dydt = batchSimpleF2(t,y,flags,params)

% params = [alpha k1p km1 k2 km2p];
alpha = params(1);
k1p = params(2);
km1 = params(3);
k2 = params(4);
km2p = params(5);

% only one input y here
% y = y1 = psi-A, y2 = psi-P, y3 = theta-A

% THIS RATE FOR psi-A(t=0) = 1, psi-P(t=0) = 0

y2 = (1-y); % for y(t=0) = 1, y2(t=0) = 0
y3 = (k1p*y + km2p*y2) / (k1p*y + km2p*y2 + km1 + k2);

dydt = alpha*(-k1p*y*(1-y3) + km1*y3);

```

```

function dydt = batchSimpleF3(t,y,flags,params)

k = params(1);
K = params(2);
yEQ = params(3);

% only one input y here
% y = y1 = psi-A

% THIS RATE FOR psi-A(t=0) = 1, psi-P(t=0) = 0

rate = -k * (y - yEQ) / (1 + K*y);

dydt = rate;

```