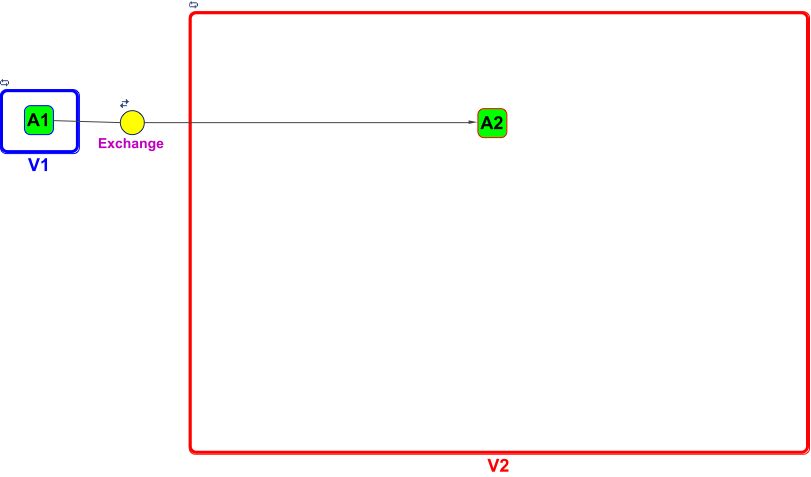
# **Dynamic simulation of a network of chemical reactions, Metabolic Control Analysis (MCA), Pharmacokinetics/Pharmacodynamics (PKPD).**

In this section we apply the methods we have learned to simulate a single chemical or enzymatic reaction, to the analysis of large networks of enzymatic or non-enzymatic reactions that belong to intracellular processes (e.g. metabolic pathways) or extracellular processes (e.g. exchange reaction between compartments, absorption and/or elimination).

Since we are going to simulate transport reactions of solutes between different compartments, it is important to understand how the *law of mass action* applies to these processes. As an example, we use a very simple model of two compartments, V1 = 1 liter, and V2 = 100 liters, containing the same species A (called A1 in compartment V1 and A2 in compartment V2) at the same concentration of 1M in both compartments. We use again the MATLAB Toolbox *Simbiology* to study the properties of this model. All the steps of this analysis can be replicated by running the cells in the *m*-file: ../TUTORIALS/PKPD/Two\_boxes.m. We start by loading the project and by getting some information about the model:

sbioloadproject('Two\_boxes');

sbioselect(m1,'Type','compartment')

sbioselect(m1,'Type','species')

sbioselect(m1,'Type','parameter')

sbioselect(m1,'Type','reaction')

We extract key parameters and species from the model and save them as variables in the Workspace.

V1 = sbioselect(m1,'Name','V1');

V2 = sbioselect(m1,'Name','V2');

A1 = sbioselect(m1,'Name','A1');

A2 = sbioselect(m1,'Name','A2');

kV1V2 = sbioselect(m1,'Name','kV1V2');

kV2V1 = sbioselect(m1,'Name','kV2V1');

We also store some initial values for concentrations and rate constants:

A1\_Init = A1.InitialAmount;

A2\_Init = A2.InitialAmount;

kV1V2.Value = 0.01;

kV2V1.Value = 0.01;

Initially the rate constants for the transfer of A from V1 to V2 and and for the transfer of A from V2 to V1 are set to the same value of 0.01 s-1. Before we start the simulation we also need to get some information on the configuration parameters of the ODE solver.

cs = getconfigset(m1, 'default');

cs

Stop = 300;

set(cs, 'SolverType', 'ode15s');

set(cs, 'StopTime', Stop);

set(cs, 'TimeUnit','second');

set(cs.RunTimeOptions, 'StatesToLog', 'All');

set(cs.CompileOptions, 'UnitConversion', true);

set(cs.CompileOptions, 'DimensionalAnalysis', true);

cs

We are now ready to follow the time evolution of the system:

Two\_boxes\_kinetics = sbiosimulate(m1, cs);

Two\_boxes\_kinetics.DataNames

m1.species

FIG\_1 = figure;set(FIG\_1,'Units','normalized','Position',[0.6 0.6 0.4 0.4]);clf

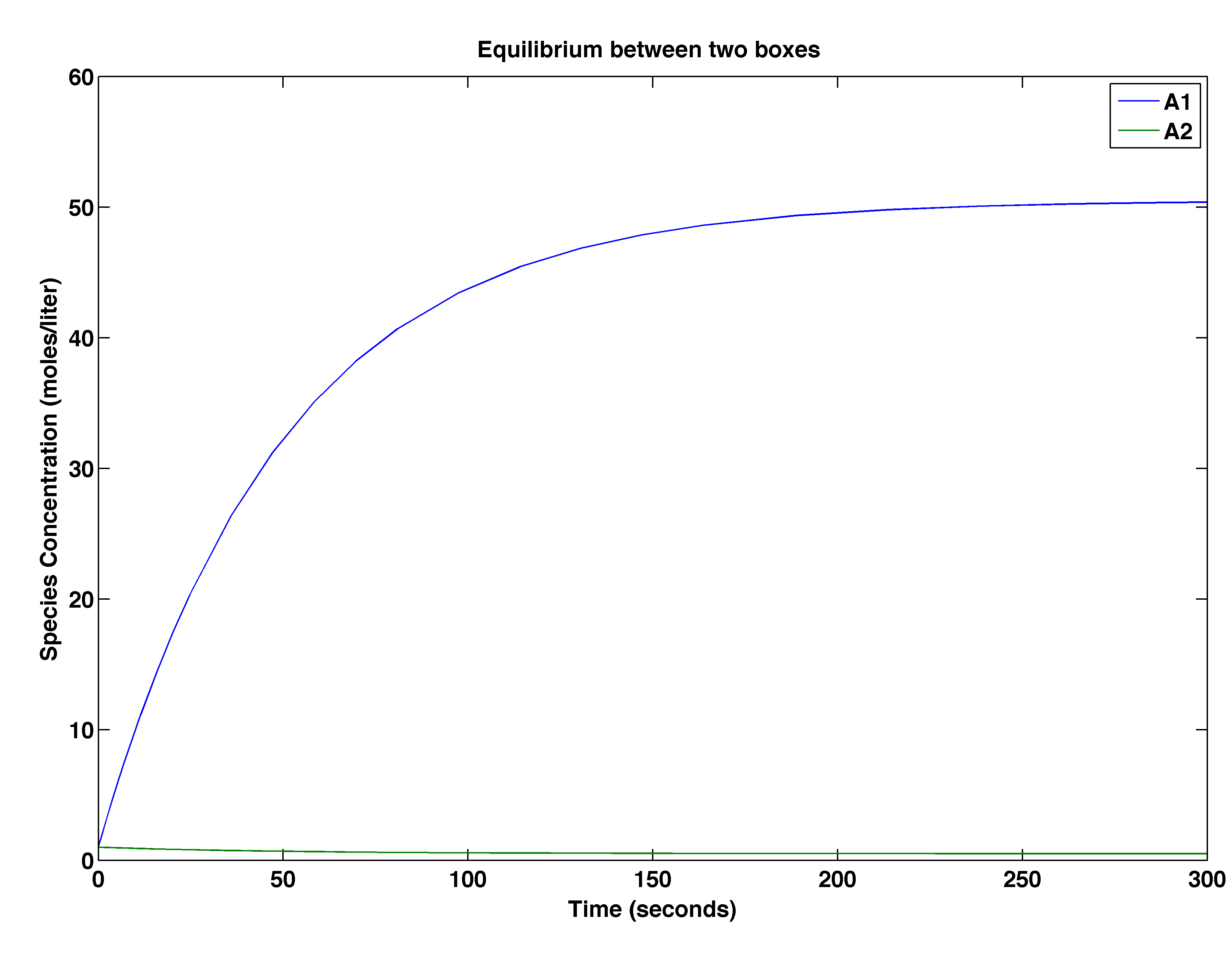
axes1 = axes('Parent',FIG\_1,'Position',[0.08 0.1 0.9 0.82]);

hold(axes1,'all');box(axes1,'on');

plot(Two\_boxes\_kinetics.Time,Two\_boxes\_kinetics.Data(:,:))

legend('A1','A2');ylabel('Species Concentration (moles/liter)');xlabel('Time (seconds)')

title('Equilibrium between two boxes ');



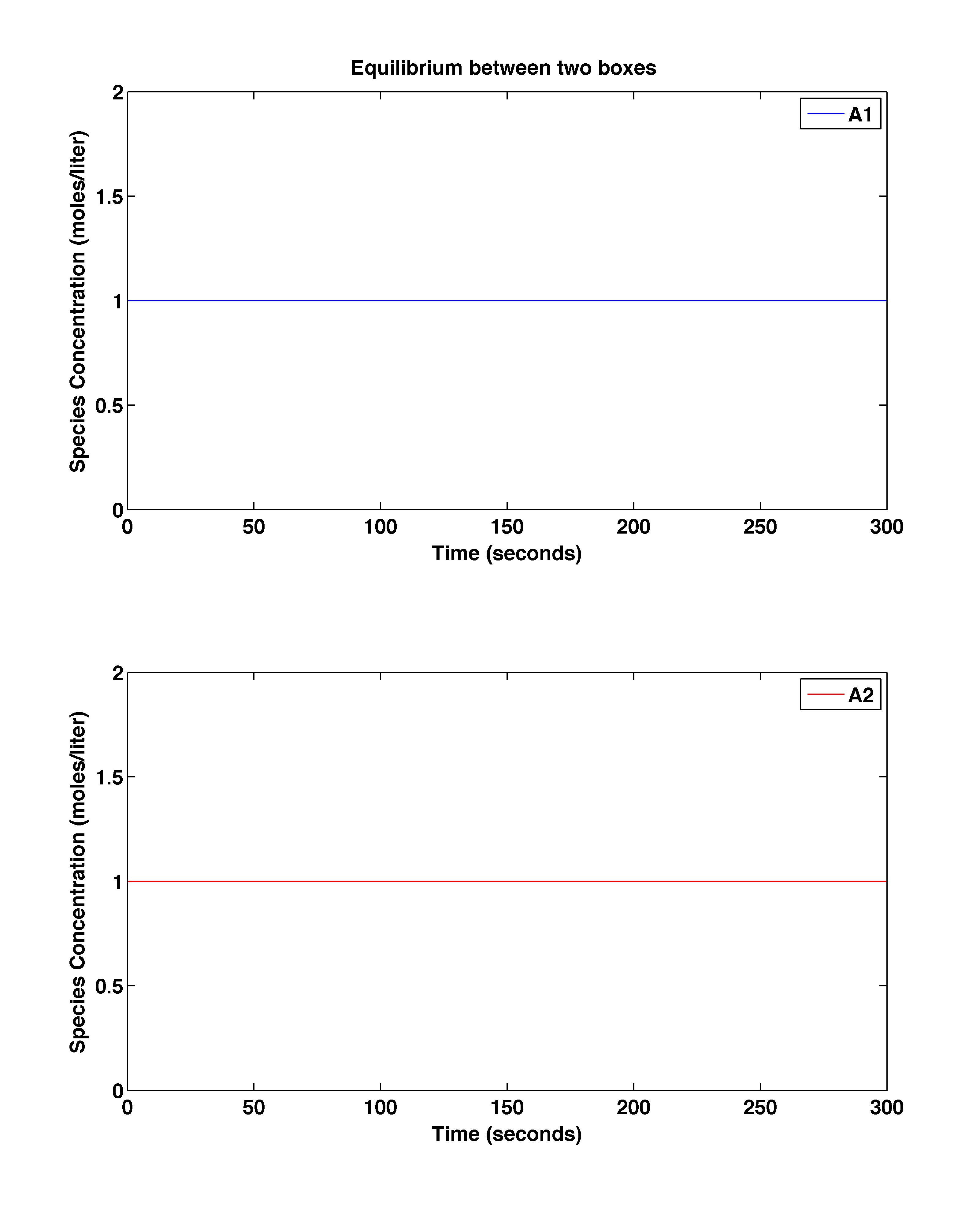
We see that very rapidly the concentration of A2 decreases from 1 M to ~0.5 M and that of A1 increases from 1 M to ~ 50 M. This result is somewhat surprising, because starting from the same concentrations of A in both compartments, and with equal rate constants for the transfer in both direction, intuitively we would expect the system to be already at equilibrium, and nothing to happen over time.

However the result can be understood by considering the exchange reaction as two independent exponential decay processes. For example, breaking up the process in 1 s steps, at 1 s we would have:

Based on this decay of both A1 and A2 at 1 s, the total number of molecules that left V1 and V2 is given by:

These molecules must be added to the other compartment:

We can see how the concentration of A increases very rapidly inside compartment V1 and decreases very slowly inside compartment V2.



If we want the system to be at equilibrium from the beginning, the solution is to *scale* (=divide) the rate constants by the volumes of the respective compartments. For example, we could set the rate constant for the transfer from V1 to V2 to 1 s-1 and the rate constant for the transfer from V2 to V1 to 0.01 s-1.

A1.InitialAmount = A1\_Init ;

A2.InitialAmount = A2\_Init;

kV1V2.Value = 1.0;

kV2V1.Value = 0.01;

Two\_boxes\_kinetics = ...

sbiosimulate(m1, cs);

FIG\_2 = figure;

set(FIG\_2,'Units','normalized',...

'Position',[0.6 0.6 0.4 0.8]);clf

subplot(2,1,1)

plot(Two\_boxes\_kinetics.Time,...

Two\_boxes\_kinetics.Data(:,1),'-b')

legend('A1');

ylabel('Species Concentration (moles/liter)');xlabel('Time (seconds)')

title('Equilibrium between two boxes ');

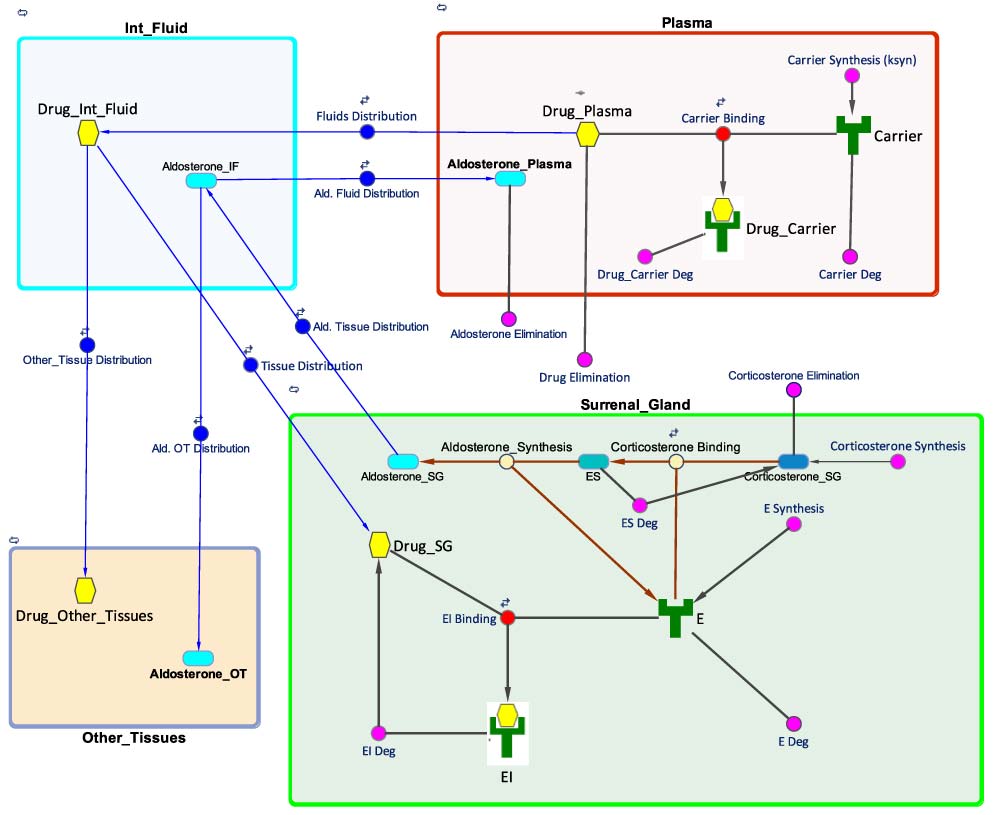
subplot(2,1,2)

plot(Two\_boxes\_kinetics.Time,Two\_boxes\_kinetics.Data(:,2),'-r')

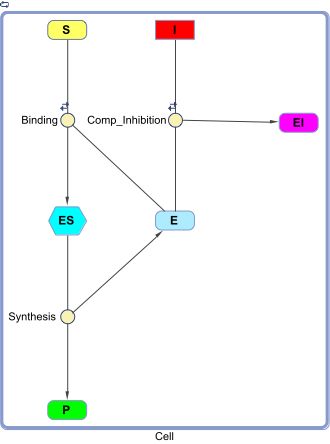
legend('A2');ylabel('Species Concentration (moles/liter)');xlabel('Time (seconds)')

With this new setting of the rate constants, we can see how the concentration of A in the two compartments remains stable over time. Thus, each time we construct a model that involves exchanges between compartments it is *very important* to remember to scale the *microscopic experimental rate constants* by the volumes of the respective compartments.

With this in mind, we are now ready to study complex networks of biological/biochemical reactions. As an example, we will analyze a pharmacokinetics-pharmacodynamics model in which a drug binds to a carrier protein in the plasma and to a target enzyme in a peripheral tissue. *Pharmacokinetics* is the study of the mechanisms of absorption and distribution of a drug, the chemical changes of the substance in the body, and the effects and routes of excretion of the metabolites of the drug. *Pharmacokinetics* is often studied in conjunction with pharmacodynamics, the study of a drug's pharmacological effect on the body. When both studies are combined to create a model of the drug action we refer to this model as a *Pharmacokinetics/Pharmacodynamics* model, or ***PKPD*** model.



As an example, we will examine a PKPD model consisting of several compartments of the human body including the plasma, the interstitial fluid, the surrenal gland, and all the other tissues. The model describes a state of *hyperaldosteronism* due to a surrenal gland tumor, or to intrinsic iperactivity of aldosterone synthetase, leading to a higher than normal plasma concentration of aldosterone and its correction by means of an inhibitor of this enzyme.



First, we consider a *competitive inhibitor*. We use again the MATLAB Toolbox *Simbiology* to carry out the simulations in the example. All the steps in the simulation can be replicated by running the cells in the *m*-file: ../TUTORIALS/PKPD/PKPD.m.

We start by loading the project and getting some information about the model:

sbioloadproject('PKPD');

sbioselect(m1,'Type','compartment')

sbioselect(m1,'Type','species')

sbioselect(m1,'Type','parameter')

sbioselect(m1,'Type','reaction')

getequations(m1)

The volumes of the different compartments, the concentrations of all the species, and the rate constants for the transfer of species between compartments have been set to realistic values for an individual of ~70 kg in body mass. The model contains also rate constants for the synthesis and degradation of the plasma protein that carries the drug and for aldosterone synthetase.

Then we extract key parameters and species from the model and save them as variables in the workspace.

PL = sbioselect(m1,'Name','Plasma');

SG = sbioselect(m1,'Name','Surrenal\_Gland');

IF = sbioselect(m1,'Name','Int\_Fluid');

OT = sbioselect(m1,'Name','Other\_Tissues');

E = sbioselect(m1,'Name','E');

ES = sbioselect(m1,'Name','ES');

EI = sbioselect(m1,'Name','EI');

ESI = sbioselect(m1,'Name','ESI');

Drug\_PL = sbioselect(m1,'Name','Drug\_Plasma');

Drug\_IF = sbioselect(m1,'Name','Drug\_Int\_fuid');

Drug\_SG = sbioselect(m1,'Name','Drug\_SG');

Drug\_OT = sbioselect(m1,'Name','Drug\_Other\_Tissues');

Aldosterone\_PL = sbioselect(m1,'Name','Aldosterone\_Plasma');

Aldosterone\_IF = sbioselect(m1,'Name','Aldosterone\_IF');

Aldosterone\_SG = sbioselect(m1,'Name','Aldosterone\_SG');

Aldosterone\_OT = sbioselect(m1,'Name','Aldosterone\_OT');

Corticosterone\_SG = sbioselect(m1,'Name','Corticosterone\_SG');

kon\_CI = sbioselect(m1,'Name','kon\_CI');

koff\_CI = sbioselect(m1,'Name','koff\_CI');

kon\_EI = sbioselect(m1,'Name','kon\_EI');

koff\_EI = sbioselect(m1,'Name','koff\_EI');

kon\_E = sbioselect(m1,'Name','kon\_E');

koff\_E = sbioselect(m1,'Name','koff\_E');

kcat\_E = sbioselect(m1,'Name','kcat\_E');

ksin\_cort = sbioselect(m1,'Name','ksin\_cort');

kel\_cort = sbioselect(m1,'Name','kel\_cort');

ksin\_E = sbioselect(m1,'Name','ksin\_E');

kdeg\_E = sbioselect(m1,'Name','kdeg\_E');

kdeg\_EI = sbioselect(m1,'Name','kdeg\_EI'); % degradation EI complex

kdeg\_ES = sbioselect(m1,'Name','kdeg\_ES'); % degradation ES complex

kti\_ald = sbioselect(m1,'Name','kti\_ald');

kit\_ald = sbioselect(m1,'Name','kit\_ald');

We also store some initial values:

Aldosterone\_PL\_Init = Aldosterone\_PL.InitialAmount;

Aldosterone\_SG\_Init = Aldosterone\_SG.InitialAmount;

Aldosterone\_OT\_Init = Aldosterone\_OT.InitialAmount;

Aldosterone\_IF\_Init = Aldosterone\_IF.InitialAmount;

We set the koff for the competitive inhibitor to 0.001 (1/hour).

koff\_EI.Value = 0.001;

Kd for the competitive inhibitor.

Drug\_Kd = koff\_EI.Value/kon\_EI.Value;

display(['Drug\_Kd = ' num2str(Drug\_Kd) ' micrograms/L']);

Before we start the simulation we will set all the necessary parameters and variable values. We also need to get some information on the configuration parameters of the simulation.

cs = getconfigset(m1, 'default');

cs

Here we configure the ODE solver:

set(cs, 'SolverType', 'sundials');

set(cs, 'TimeUnit','hour');

set(cs.RunTimeOptions, 'StatesToLog', 'All');

set(cs.CompileOptions, 'UnitConversion', true);

set(cs.CompileOptions, 'DimensionalAnalysis', true);

cs

Now we are ready to simulate the time evolution of this system. We will first set the initial stop time at 6 hours.

Stop = 6;

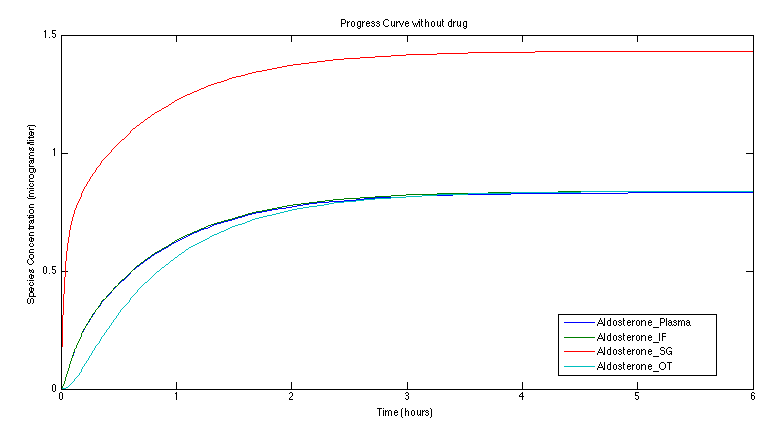
set(cs, 'StopTime', Stop);

In the standard simulation we don't want any model variation (first []) and no variations in dosing (second []). The use of these parameters will appear clear later on.

PKPD\_Kinetics = sbiosimulate(m1, cs, [], []);

PKPD\_Kinetics.DataNames

m1.species

We plot the aldosterone concentration in the plasma and the interstitial fluid, and in the tissues where aldosterone is not produced. We can see how 6 hours is the minimum amount of time required for the system to reach equilibrium starting from concentrations of aldosterone that are 0 everywhere. Notice how the concentration of aldosterone is rapidly equilibrating between plasma and interstitial fluid (IF), but more slowly between plasma/(IF) and the tissues. As expected, aldosterone concentration is higher in the surrenal gland (SG), where it's synthesized, than everywhere else.

**Sensitivity Analysis - Metabolic Control Analysis (MCA) and Control Coefficients**

So far we have been interested in the kinetic behavior of the model with respect to the plasma comcentration of aldosterone. A natural question to ask is: which parameters of the model affect aldosterone levels, and what are the magnitudes of those effects? To answer this question we can compute the *sensitivity* of plasma aldosterone with respect to various parameters in the model. Calculating sensitivities lets you determine which species or parameter in a model is most sensitive to a specific condition (for example, a drug), defined by a species or parameter. Thus, if a model has a species x, and two parameters y and z, the time-dependent sensitivities of x with respect to each parameter value are the time-dependent derivatives. If we are calculating the sensitivity of plasma aldosterone (x) with respect to two parameters (y) and (z) in the model, the time-dependent derivatives are:

Finally, normalizing the sensitivities of a species with respect to different parameters allows them to be compared with each other. The following lines show you how sensitivities of a species x with respect to a parameter k are calculated for each normalization type:

'None': no normalization. 'Half': normalization relative to the numerator 'Full': dimensionless

change of x *vs* change of k % change of x *vs* change of k % change of x *vs* % change of k

Related to the concept of *full* sensitivity is the concept of *control coefficient* (cc). A control coefficient is a relative measure of how much a perturbation on a 'local' variable affects a 'system' variable (e.g. fluxes or concentrations). It is defined as:

where A is the variable, i the step (enzyme) and v the rate of the perturbed step. The most common control coefficients are those for the entire pathway flux and the species concentrations, but any variable of the system can be analyzed with MCA and have control coefficients defined by equations analogous to the above. Although there is no need for the system to be in a steady state, traditionally we are interested in knowing how the flux ***J*** through an entire or part of a metabolic pathway (this is the 'system' variable) reacts to a small perturbation of the activity of a certain enzyme (this is the 'local' variable) in the pathway.

First, we configure the task specific stop time, so we are sure the system is at equilibrium. We simulate up to 30 hours.

set(cs, 'StopTime', 30.0);

Now we are ready to calculate the flux ***J*** and concentrations control coefficient for the model parameters. For example, we can calculate the flux ***J*** of aldosterone synthesis in the surrenal gland; this flux is the product of the rate of aldosterone synthesis from the ES (as concentration/s) times the surrenal gland volume:

***J***\_ald = ReactionFlux19 = (kcat\_E\*ES)\*Surrenal\_Gland\_Volume

We find out the column of the simulation array containing the ES concentrations:

PKPD\_sens = sbiosimulate(m1);

PKPD\_sens.DataNames

Species are logged in the following order:

1 Plasma Carrier microgram/liter

2 Plasma Drug\_Carrier microgram/liter

3 Plasma Drug\_Plasma microgram/liter

4 Plasma Aldosterone\_Plasma microgram/liter

5 Int\_Fluid Drug\_Int\_Fluid microgram/liter

6 Int\_Fluid Aldosterone\_IF microgram/liter

7 Surrenal\_Gland E microgram/liter

8 Surrenal\_Gland Drug\_SG microgram/liter

9 Surrenal\_Gland EI microgram/liter

10 Surrenal\_Gland Corticosterone\_SG microgram/liter

11 Surrenal\_Gland Aldosterone\_SG microgram/liter

12 Surrenal\_Gland ES microgram/liter

13 Surrenal\_Gland ESI microgram/liter

14 Other\_Tissues Drug\_Other\_Tissues microgram/liter

15 Other\_Tissues Aldosterone\_OT microgram/liter

Thus, ES is column 12 in the array Data. We can easily set up separate determinations of flux and concentration cc's for all parameters with a single loop:

nparams = size(params,1);

cc\_Jk = zeros(1,nparams);

cc\_Ck = zeros(1,nparams);

delta = eps^(1/4);

for n = 1:nparams

param\_no = n;

k\_ref = params(param\_no).Value;

% We can try to make the change infinitesimally small

k\_range = [(1-delta)\*k\_ref k\_ref (1+delta)\*k\_ref];

J\_range = zeros(1,3);

C\_range = zeros(1,3);

for i = 1:3

params(param\_no).Value = k\_range(i);

PKPD\_sens = sbiosimulate(m1);

ReactionFlux19 = (kcat\_E.Value\*PKPD\_sens.Data(end,12))\*SG.capacity;

J\_range(i) = ReactionFlux19; % Flux cc

C\_range(i) = PKPD\_sens.Data(end,4); % Concentration cc

end

Centered difference:

dJdk = ((J\_range(3)-J\_range(1))/(k\_range(3)-k\_range(1)))\*(k\_range(2)/J\_range(2));

dCdk = ((C\_range(3)-C\_range(1))/(k\_range(3)-k\_range(1)))\*(k\_range(2)/C\_range(2));

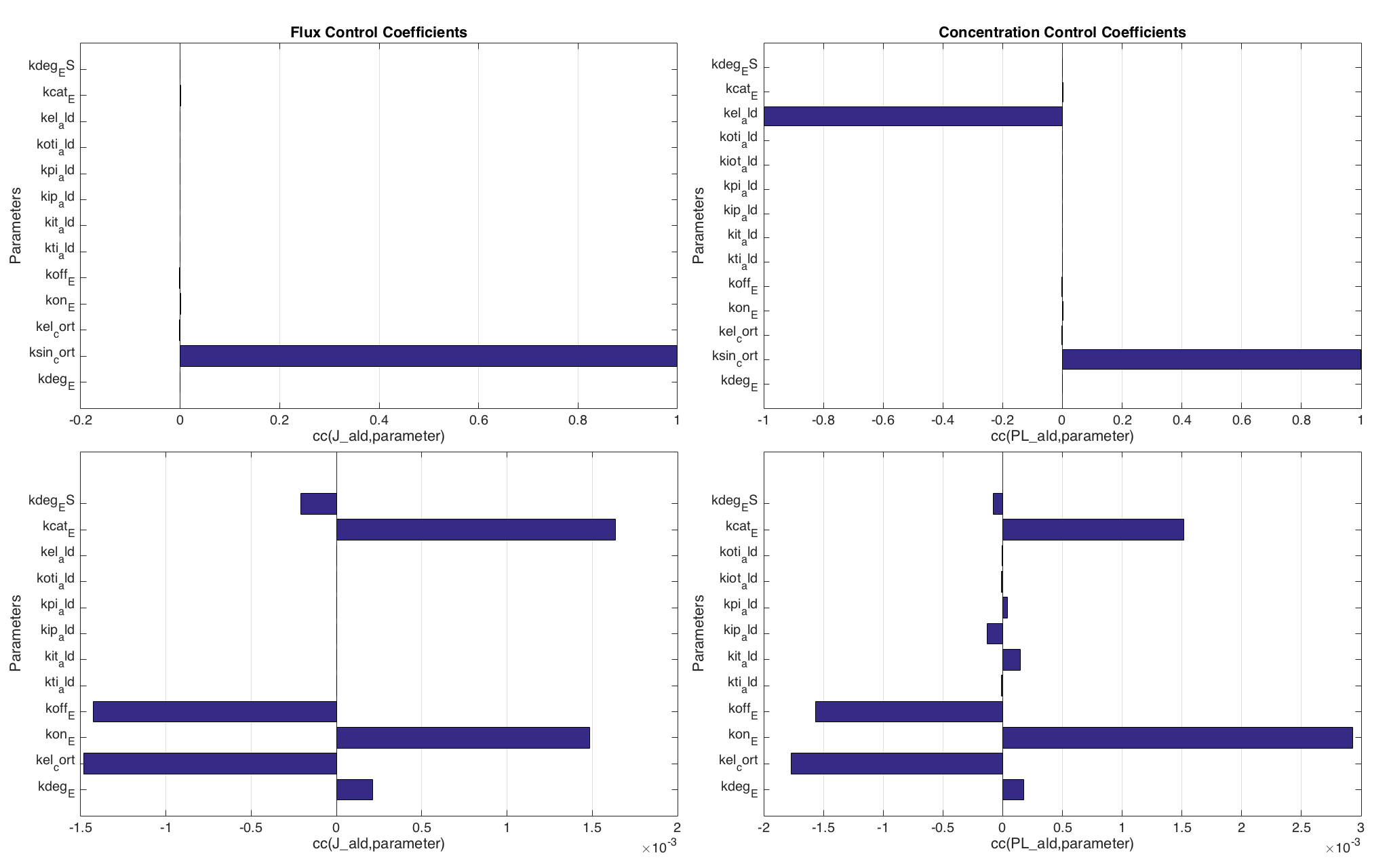
cc\_Jk(n) = dJdk;

cc\_Ck(n) = dCdk;

params(param\_no).Value = k\_ref;

end

We can plot out the results as bar graphs:

****

*Negative control coefficients* indicate that an *increase* in that parameter produces a *decrease* in flux or concentration. The upper two insets show that the synthesis of corticosterone is the factor the produces the largest increase in aldosterone synthesis; aldosterone clearance from the plasma is the factor the produces the largest decrease in aldosterone concentration. The lower two insets show an enlargement of the bars that count near 0 in the upper two insets. As expected, kcat\_E and kon\_E produce the largest increase in aldosterone synthesis or concentration. koff\_E and the clearance of corticosterone produce the largest decrease. Notice how the degradation of the enzyme (kdeg\_E) and of the enzyme:substrate complex (kdeg\_ES) affect the aldosterone synthesis and plasma concentration in opposite ways.

**Dose Response Curves**

It is clear that the rate constants for the synthesis of corticosterone (ksyn\_cort) and the renal clearance of aldosterone (kel\_ald) are the main factors affecting the plasma concentration of aldosterone. In this example we cannot increase the renal clearance and we do not have a drug that inhibits corticosterone synthesis. The third most important factor is the kcat (kcat\_E) of aldosterone synthase, but we only have a competitive inhibitor that does not affect the kcat, and in practive works by subtracting a certain amount of enzyme from the pool of active synthase. We will work with what we have.

The first step is to determine what level of inhibition we need to achieve in order to bring the plasma aldosterone level around its normal level of 3 micrograms/liter. We can set up a Dose-Response curve (percentage of maximal response against log of the drug concentration) to determine this level. In this case we really simulate the effect of the drug inhibition by changing the amount of enzyme in the surrenal gland.

E\_ref = E.InitialAmount;

High\_Conc = 1000;

ncons = 35;

conc\_vec = logspace(log10(1),log10(High\_Conc),ncons);

range = conc\_vec <= 1 + High\_Conc;

We simulate for 30 hours:

Stop = 30;

set(cs, 'StopTime', Stop);

Response = zeros(1,ncons);

Dose = conc\_vec;

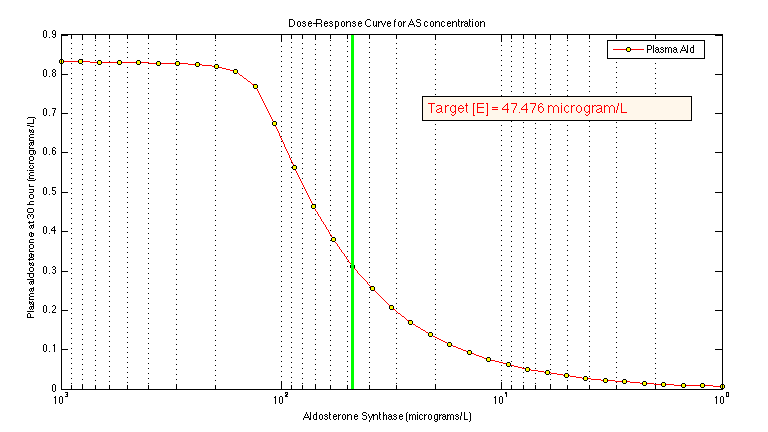
for i = 1:ncons

E.InitialAmount = conc\_vec(i);

PKPD\_Kinetics = sbiosimulate(m1, cs, [], []);

Response(i) = PKPD\_Kinetics.Data(end,4);

end



We are almost there: this means that we need to bring the activity of aldosterone synthase to ~4.7% of the initial amount in order to bring the concentration of plasma aldosterone into ~normal range.

Now we are ready to start experimenting with our drug. For example, we can develop a Dose-Response curve of the level of plasma aldosterone at 30 hours after a single dose at 6 hour.

First, we get some information about the dosing that is already stored in the model:

doses = get(m1.doses);

d1 = getdose(m1, 'Single Dose');

d2 = getdose(m1, 'Repeated Dose');

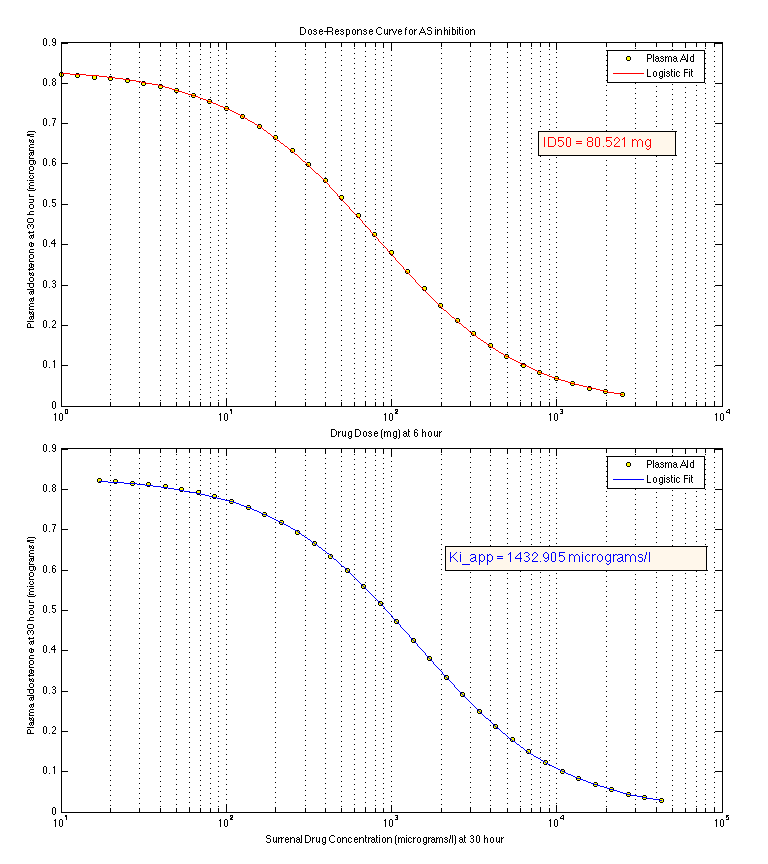
We loop through a vector of drug concentrations spaced logarithmically.

High\_Conc = 2500;

ncons = 35;

conc\_vec = logspace(log10(1),log10(High\_Conc),ncons);

range = conc\_vec <= 1 + High\_Conc;



We simulate for 30 hours

Stop = 30;

set(cs, 'StopTime', Stop);

Response = zeros(1,ncons);

Dose = conc\_vec;

Drug\_conc = zeros(1,ncons);

In the simulation we don't want any variation in the model (first []), but we are going to change the dosing parameters (d1).

for i = 1:ncons

set(d1, 'Amount', conc\_vec(i));

set(d1, 'Rate', conc\_vec(i));

PKPD\_Kinetics = ...

sbiosimulate(m1, cs, [], d1);

Response(i) = PKPD\_Kinetics.Data(end,4);

Drug\_conc(i) = PKPD\_Kinetics.Data(end,8);

end

The dose response curves suggest an ID50 ~ 80 mg (dose inhibiting 50% of the enzyme), and a drug apparent Ki ~ 1.4 mg/l. This is much higher than the expected Ki = Kd = 0.01 micrograms/l. An explanation of this unusual result is discussed in the example 'PKPDnc' (which can be run using the *m*-file: ../TUTORIALS/PKPD/PKPDnc.m)

**Developing a Dosing Schedule**

Based on the ID50(at 30 hours) of 80 mg and in consideration of the progressive accumulation of the drug with repeated doses, we can try a schedule of 60 mg/day. Let's see what the stored repeated dosage looks like (it is stored in the 'd2' variable):

get(d2)

Let's set the trial to 20 days and the amount to 60 mg/day, each time administered during the course of 1 hour (the absorption time):

d2.RepeatCount = 20; % doses

d2.Amount = 60; % milligrams

d2.Interval = 24; % hours

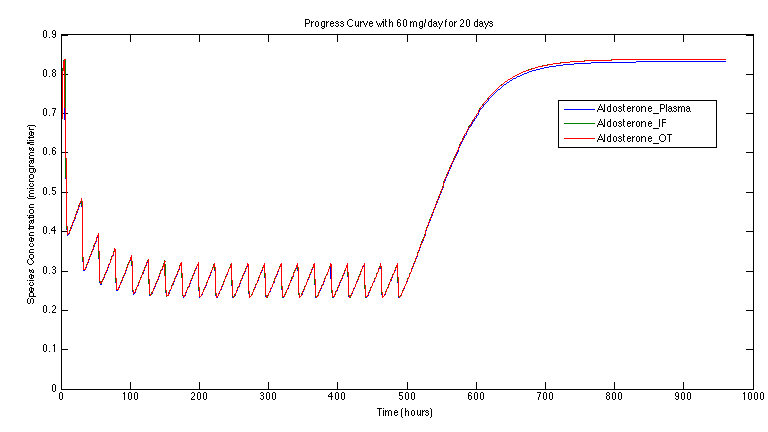
d2.Rate = 60; % milligrams/hour

d2.StartTime = 6; % hours

Now we can repeat the simulation with this dose schedule and simulate for 960 hours (40 days). After 20 days we will see the aldosterone level going up again, but we are primarily interested in seeing if during the administration of the drug we obtain a stable normal level of plasma aldosterone.

Stop = 960;

set(cs, 'StopTime', Stop);

PKPD\_Kinetics = sbiosimulate(m1, cs, [], d2);

We have achieved a stable normalization of the plasma aldosterone levels with only 1 daily dose and acceptably small daily fluctuations. We can expect that even smaller fluctuations could be achieved by splitting the daily dose in two doses. We decrease the amount to 25 mg and increase the repeat count to 40 with an interval of 12 hours:

d2.Amount = 25;

d2.Rate = 25;

d2.Interval = 12;

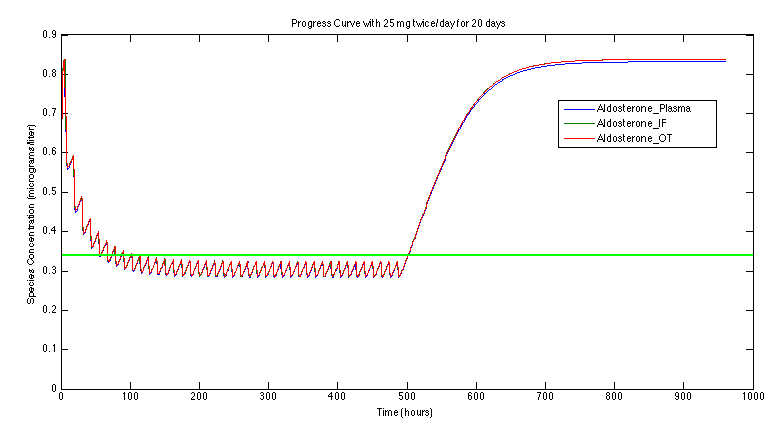
d2.RepeatCount = 40;

d2.StartTime = 6;

Stop = 960;

set(cs, 'StopTime', Stop);

PKPD\_Kinetics = sbiosimulate(m1, cs, [], d2);

The green line represents the physiological maximum value of plasma aldosterone during the day. A stable normalization of the plasma aldosterone levels with very small daily fluctuations is achieved with only 2 daily doses of 25 mg.

Finally, we are interested in knowing whether the cc's for the aldosterone synthesis flux have changed in this new steady state produced by the drug. Again we can easily determine the cc's for all parameters with a single calculation. A convenient time to stop the simulation is 398 hours (middle of a small fluctuation):

Stop = 397;

set(cs, 'StopTime', Stop);

nparams = size(params,1);

cc\_Jk = zeros(1,nparams);

cc\_Ck = zeros(1,nparams);

delta = eps^(1/4);

for n = 1:nparams

param\_no = n;

k\_ref = params(param\_no).Value;

k\_range = [(1-delta)\*k\_ref k\_ref (1+delta)\*k\_ref];

J\_range = zeros(1,3);

C\_range = zeros(1,3);

for i = 1:3

params(param\_no).Value = k\_range(i);

PKPD\_sens = sbiosimulate(m1, cs, [], d2);

ReactionFlux19 = (kcat\_E.Value\*PKPD\_sens.Data(end,12))\*SG.capacity;

J\_range(i) = ReactionFlux19; % Flux cc

C\_range(i) = PKPD\_sens.Data(end,4); % Concentration cc

end

dJdk = ((J\_range(3)-J\_range(1))/(k\_range(3)-k\_range(1)))\*(k\_range(2)/J\_range(2));

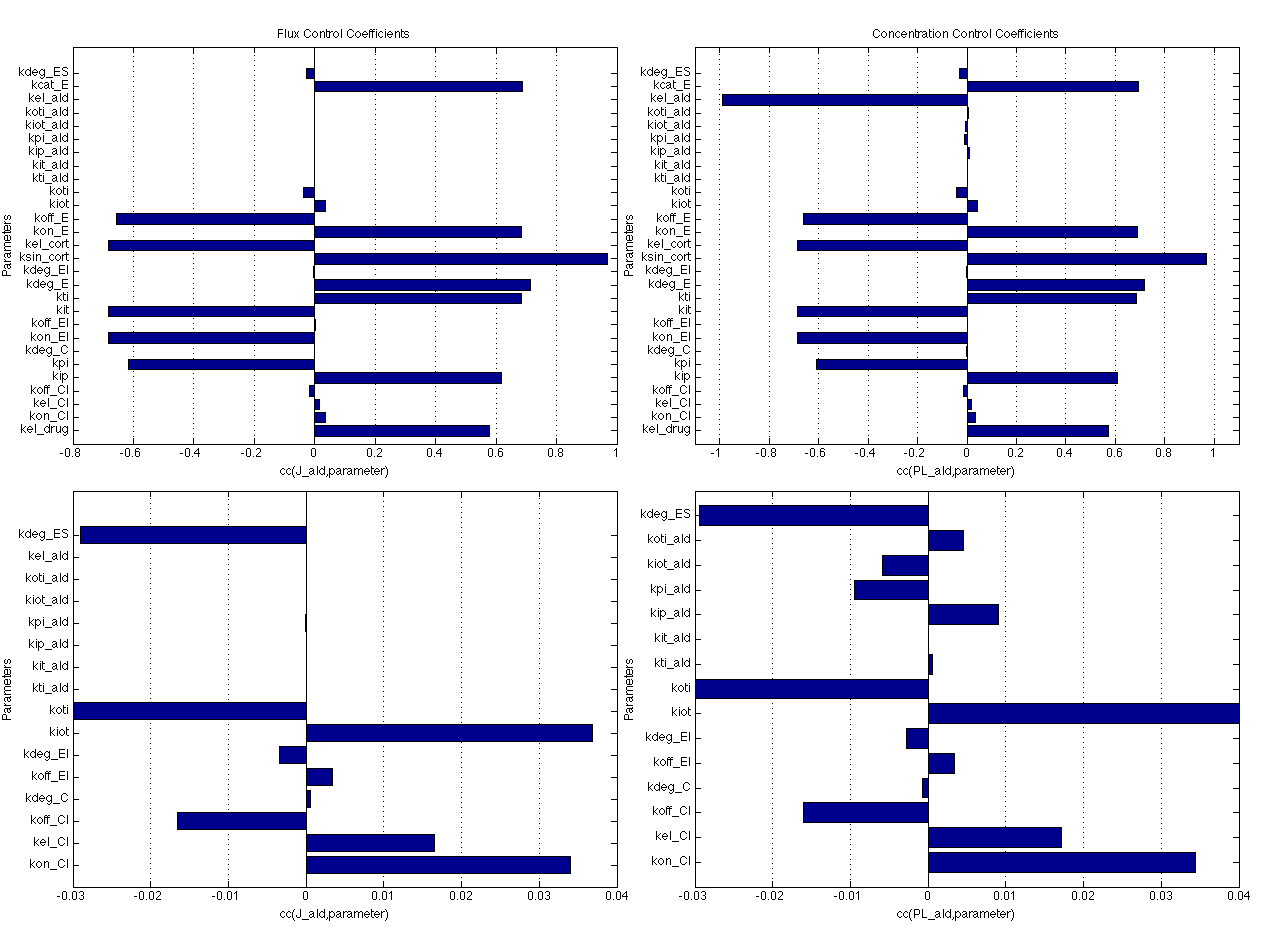
dCdk = ((C\_range(3)-C\_range(1))/(k\_range(3)-k\_range(1)))\*(k\_range(2)/C\_range(2));

cc\_Jk(n) = dJdk;

cc\_Ck(n) = dCdk;

params(param\_no).Value = k\_ref;

end



The upper two insets show that the synthesis of corticosterone is the factor the produces the largest increase in aldosterone synthesis; but now other factors are also very important (e.g., kcat\_E, kdeg\_E). Aldosterone clearance from the plasma is the factor the produces the largest decrease in aldosterone concentration. The lower two insets show an enlargement of the bars that count near 0 in the upper two insets.

In conclusion, the presence of the inhibitor dramatically affects the cc's for both the synthesis and concentration of aldosterone. Thus, other steps in the PKPD model become susceptible to further modulation.

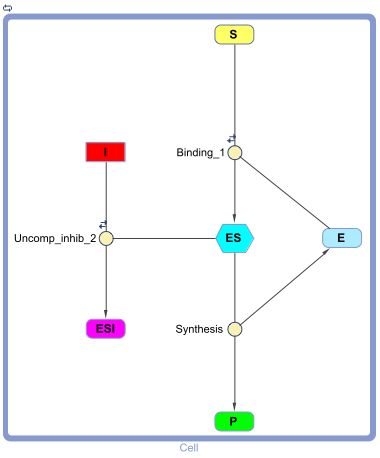
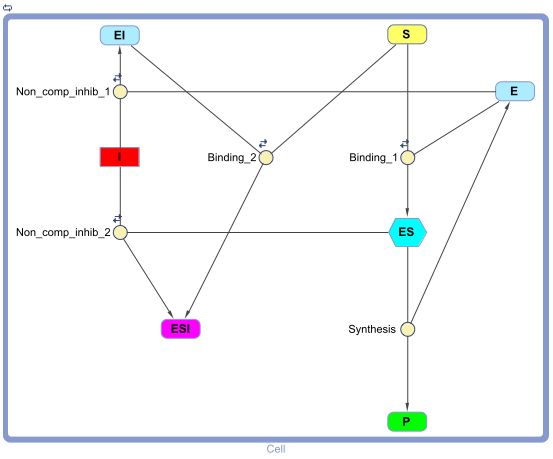
**PRACTICE**

Two additional MATLAB scripts are provided allowing the simulation of the same model of aldosterone synthesis and distribution in the body in the presence of a non-competive: (../TUTORIALS/PKPD/PKPDnc.m)

and an uncompetitive:

(../TUTORIALS/PKPD/PKPDunc.m)

inhibitor, respectively.



Carry out the simulations and explain the effects of these inhibitors in comparison with the effects of the competitive inhibitor used in the class example.