

Understanding Compartmental Modeling:

Synthesis Model

Conversion Model

Flow Model

Exchange Model

All Processes Model

Compartmental Modeling:

The following five short lessons will teach you about compartmental modeling. At the end of the first four lessons you should be able to do the following:

- (1) Identify the sources and sinks for each substance in a complex reaction.
- (2) Write correct mass balance equations governing the system.
- (3) Code these equations in JSim's Mathematical Modeling Language (MML) along with other necessary information.
- (4) Compile them into a JSim model, run the model, and save it.

To test your skill, Lesson five asks you to write a new model which combines everything you learn in the previous four lessons.

Lesson I: Introduction to compartmental modeling and using JSim

1. A restricted compartmental definition which covers a few simple cases:

A compartment represents a physical volume, V . In the volume, there is an amount of material, Q . The concentration, C , is given by

$$C = Q/V.$$

A compartment is assumed to be instantaneously well mixed. In physical terms, this means that the diffusional process is infinitely fast.

Sources are processes which increase Q . Sinks are processes which decrease Q . The rate of change of Q with respect to time, the change in the amount is given by

$$dQ/dt = +Sources - Sinks.$$

This is called an ordinary differential equation (ODE).

In the simpler models, the volume is usually constant, and the equation is written as either

$$V * dC/dt = +Sources - Sinks$$

or

$$dC/dt = Sources/V - Sinks/V.$$

The complete specification of an ODE requires an initial condition (IC). The initial condition specifies the concentration at time equals 0. This is written as

$$C(0)=C_0.$$

2. Specifying units:

The units for the amount of material are usually moles (*mole*), millimoles (*mmol*), micromoles (*umol*), nanomoles (*nmol*) or picomoles (*pmol*). The units for concentration are usually molar (*M*), millimolar (*mM*), micromolar (*uM*), or nanomolar (*nM*), or picomolar (*pM*). The units for volume include liter (both *liter* and *L*), milliliter (ml), and cubic centimeters (cm^3). The units for time include seconds (both *s* and *sec*), millisecond (*ms* and *msec*), minute (*min*) and hour (*hr*). A 1 molar concentration is 1 mole of substance per liter. Water has a concentration of 55 moles/liter or 55 Molar.

3. A compartmental model for Synthesis, Constant Value, or Decay

The ODE and IC are given by

$$dC/dt=S*C/V, C(0)=C_0.$$

The analytic solution is

$$C(t)=C_0*\exp(S*t/V).$$

If $S>0$, the right hand side of the equation is a source and the solution is exponentially growing. If $S=0$, the solution is a constant (see previous example). If $S<0$, the right hand side of the equation is a sink and the solution is exponentially decaying.

4. Coding the model in JSim:

```
/* Note: Once something has been explained in the model code, the
comment will be deleted from future models, e.g., a comment can be in
block form beginning with forward slash asterisk and terminated with
asterisk forward slash. Other comments can be in the form two forward
slashes followed by the comment. Please note that the demonstration projects
have added detail and comments which are not necessary for the model to run. */

import nsrunit;          // enables units in the model
unit conversion on; /* Checks equations for
                      dimensional compatibility. */
math SynthesisModel {// Constant model
// INDEPENDENT VARIABLE
realDomain t s; t.min=0; t.max=3; t.delta=0.1;

// PARAMETERS
real C0 = 1 mM,          // Initial Concentration
      V = 2 ml,           // Volume
      S = 1 ml/s;        // Synthesis rate
```

```
// DEPENDENT VARIABLES
real C(t) mM,           // Numeric Solution concentration
      analyticC(t);      // Analytic Solution concentration

// INITIAL CONDITION
when(t=t.min) C=C0;

// ORDINARY DIFFERENTIAL EQUATION
C:t=S*C/V;
// ANALYTIC SOLUTION
analyticC=C0*exp(S*t/V);
}
```

Model is in SynthesisModel.proj.

This model is accessible from

<http://nsr.bioeng.washington.edu/jsim/models/webmodel/NSR/SynthesisModel/>

You can run this model as an applet. If you have downloaded JSim to your own computer, you can download the code. This model has a "visual" interface. You can go to the regular interface by clicking on the Input/Output button on the visual interface.

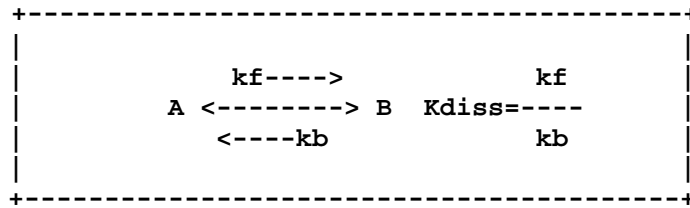
Save your projects by going to the Project tab and use the File tab.

Read model Notes (tab at bottom of run time page).

Also note that the title of the plot mentions "Run LOOPS".

Lesson II. Multiple substances in a compartment

1. Compartmental Model with conversion:



This is a model where substance A can become substance B and substance B can become substance A. The conversion is governed by a single rate constant and a dissociation constant,

K_{diss} . The dissociation constant is the inverse of the equilibrium constant.

$$K_{diss} = k_f / k_b.$$

$$k_f = K_{diss} * k_b.$$

$$dA/dt = -k_f * A + k_b * B.$$

$$dB/dt = k_f * A - k_b * B.$$

Note that the volumes are not used in this equation because the volumes would appear on both sides of the equations!!! The three exclamation points indicate that you need to really pay attention to what you have just read--it will be important later. In each ODE equation identify the sources and sinks.

2. Coding the model in JSim

```
import nsrunit;
unit conversion on;

math ConversionModel {
  // INDEPENDENT VARIABLE
  realDomain t s; t.min=0; t.max=3; t.delta=0.1;

  // PARAMETERS
  real A0      = 1 mM,           // Initial concentration of A
        B0      = 0 mM,           // Initial concentration of B
        Kdiss    = 3 dimensionless, // Dissociation constant
        kb       = 1 sec^(-1),    // Backward rate constant B->A
        kf       = 1 sec^(-1);    // Forward rate constant A->B
        kf       = Kdiss*kb;      // Calculate forward rate constant

  // VARIABLES
  real A(t) mM,                  // Concentration of A
        B(t) mM;                 // Concentration of B

  // INITIAL CONDITIONS
  when(t=t.min) {A=A0; B=B0;}    // Note the use of brackets
```

```
// ORDINARY DIFFERENTIAL EQUATIONS
A:t = -kf*A + kb*B;
B:t = kf*A - kb*B;

/* ANALYTIC SOLUTIONS from Maple(TM) */
real analyticA(t) mM, analyticB(t) mM;
private real      Exponential(t) dimensionless;
Exponential = exp(-(Kdiss+1)*kb*t);
analyticA = (1/(Kdiss+1))*(B0+A0 + (Kdiss*A0-B0)*Exponential);
analyticB = (1/(Kdiss+1))*(Kdiss*(B0+A0)-(Kdiss*A0-B0)*Exponential);

} /* End of Model*/
```

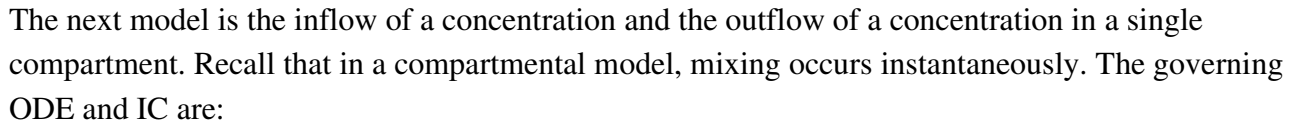
This model is accessible from

<http://nsr.bioeng.washington.edu/jsim/models/webmodel/NSR/ConversionModel/>

Explore the model by changing values for the various parameters and running solutions. You may have noticed that the volume is missing from this model. When everything happens in the same volume, and flow is not involved, the volume is usually excluded from the model.

Locate the ODE solvers page (Pages Tab pull down menu on Run Time Page.) Try the various solvers and also vary t.delta with them. How does the error in Euler-1 step change with changes in time step? What about RK4 (Runge-Kutta fourth order)?

1. Inflow and Outflow



where $C_{in}(t)$ is the inflowing concentration and $C_{out}(t)$ is the outflowing concentration, and $C(0)=0$; What are the source? What is the sink?

$$dC/dt = Flow/Volume * (C_{in}(t) - C).$$
$$C(t) = C_{in} - (C_{in} - C_0) * \exp(-Flow * t / Volume).$$

2. Coding the model in JSim:

```
import nsrunit;
import nsrunit; unit conversion on;

math FlowModel {

// INDEPENDENT VARIABLE
realDomain t sec; t.min=0; t.max=10; t.delta=0.1;

// PARAMETERS
real C0      = 1 mM,                // Initial Concentration
    Volume   = 0.05 ml/g,          // Volume of compartment per gram of tissue
    Flow     = 1 ml/(g*min) ;      // Flow rate (volume per gram of tissue per minute)
extern real Cin(t) mM;              // Inflowing concentration (defined with function generator)

// VARIABLES
```

```

real C(t) mM,                // Concentration in compartment
    Cout(t) mM;              // Outflowing concentration NOTE that Cout=C
because                        // the compartment is instantaneously well
mixed.
// INITIAL CONDITION
when(t = t.min) C=C0;

// ORDINARY DIFFERENTIAL EQUATION
C:t = (Flow/Volume)*(Cin-Cout);
Cout = C;

// ANALYTIC SOLUTION only works when Cin is a constant
real analyticC(t) mM;
real tau sec;
tau = Volume/Flow;
analyticC=Cin-(Cin-C0)*exp(-t/tau);

} // END OF MODEL

```

This model is accessible from

<http://nsr.bioeng.washington.edu/jsim/models/webmodel/NSR/FlowModel/>

Vary **C0** and **Cin**.

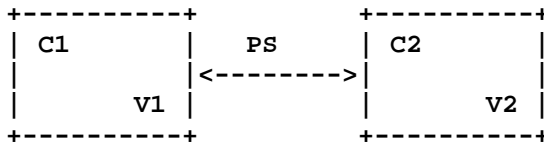
Follow the instructions in the JSim Notes (tab at bottom of Run Time Page).

3. The function generator

The function generator for **Cin(t)** can be accessed by the small sine wave enclosed in a circle (~) button to the right of **Cin(t)** . Selecting this button will bring up a menu for creating a function generator. You can accept the default name fgen_1 or create a new function generator with a different name. Select "Use existing fgen_1." Selecting the Pulse 1 button gives you access to all of the predefined functions. Note that the analytic solution based on constant input will no longer be valid as a comparison. You may wish to increase t.max to 30. Compartment flow models will be revisited when flow compartment in series are considered.

Lesson IV. Multiple compartments

1. Compartmental Model with exchange:



This is a model with exchange between two compartments. The exchange process is governed by an exchange coefficient. For reasons which will be explained in a later lesson, the exchange coefficient is designated PS . The governing equations are

$$V_1 * dC_1 / dt = PS * C_2 - PS * C_1$$

$$V_2 * dC_2 / dt = PS * C_1 - PS * C_2 ,$$

which emphasizes the mass balance. They are usually written in the form

$$dC_1 / dt = (PS / V_1) * (C_2 - C_1)$$

$$dC_2 / dt = -(PS / V_2) * (C_2 - C_1) .$$

Note that the right hand side (*RHS*) of the first equation is divided by V_1 and the *RHS* of the 2nd equation is divided by V_2 . The two volumes are not necessarily equal. Identify all sources and sinks.

2. Coding the model in JSim:

```
import nsrunit; unit conversion on;

math ExchangeModel {
// INDEPENDENT VARIABLE
realDomain t s; t.min=0; t.max=4; t.delta=0.1;

// PARAMETERS
real C10 = 1 mM,           // Initial concentration in first compartment
    C20 = 0 mM,           // Initial concentration in second compartment
    V1 = 0.05 ml/g,        // Volume of first compartment
                                // units are milliliters per gram of tissue
    V2 = 0.15 ml/g,        // Volume of second compartment
    PS = 5 ml/(g*min);     // Exchange rate

// VARIABLES
real C1(t) mM,             // Time dependent concentration in first compartment
    C2(t) mM;             // Time dependent concentration in second compartment

// INITIAL CONDITIONS
when(t=t.min) {C1 = C10;
               C2 = C20;}
```

```

// ORDINARY DIFFERENTIAL EQUATIONS
C1:t= (PS/V1)*(C2-C1);
C2:t=-(PS/V2)*(C2-C1);

/* ANALYTIC SOLUTIONS from Maple(TM) */
real analyticC1(t) mM,
    analyticC2(t) mM,
    Exponential(t) dimensionless;

Exponential    = exp(-PS*(V1+V2)*t/(V1*V2));
analyticC1(t) = (1/(V1+V2)) * (C20*V2+C10*V1
                             + V2*(C10-C20)*Exponential);
analyticC2(t) = (1/(V1+V2)) * (C20*V2+C10*V1
                             - V1*(C10-C20)*Exponential);

} // END OF MODEL

```

This model is accessible from

<http://nsr.bioeng.washington.edu/jsim/models/webmodel/NSR/ExchangeModel/>

Model is in ExchangeModel.proj.

Question: Ordinary Differential Equations are solved numerically by many different methods.

Access the ODE solvers from the **Run Time** GUI by using the **Pages**

tab and selecting **Solvers**. Change the ODE method to **Euler** and set **nstep** to 1.

Estimate the maximum error as a function of t.delta by changing t.delta (Run Time GUI) from 0.1 to 0.01 and 0.001. Do the same for RK2 and RK4. Interpret your results.

How many species will you need? The exchange coefficients for A and B are different. You should be able to use the units from the previous problem.