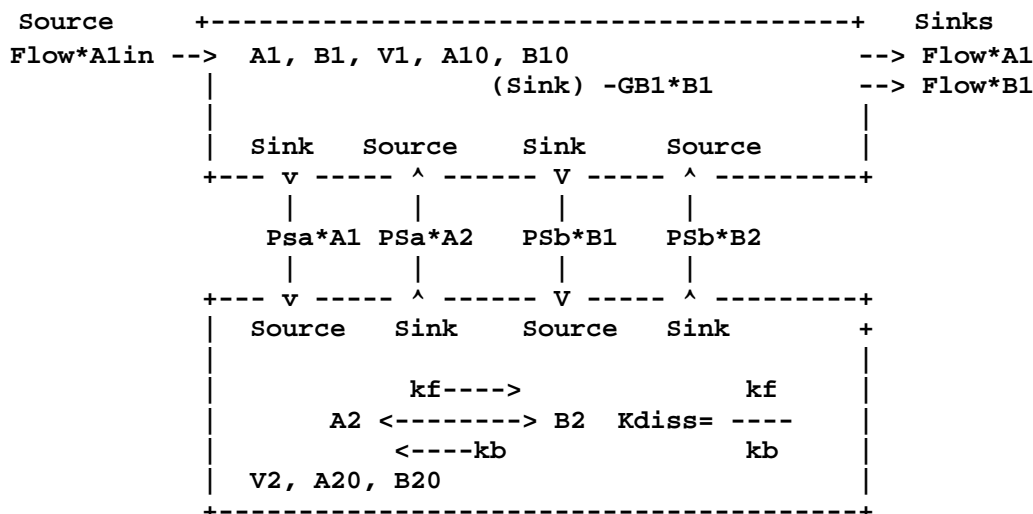


Creating New Compartmental Models

Lesson V: Creating a new model



We next construct a model for two species with inflow of species A, exchange of A between two compartments, conversion of A to B in non-flowing compartment, exchange of B with the flowing compartment, consumption of B in the flowing compartment and outflow of A and B.

The governing equations are

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

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

$$dA_1/dt = (Flow/V_1) * (A_{in} - A_1) - (PS_A/V_1) * (A_1 - A_2);$$

$$dA_2/dt = (PS_A/V_2) * (A_1 - A_2) - k_f * A_2 + k_b * B_2;$$

$$dB_1/dt = (Flow/V_1) * (-B_1) - (PS_B/V_1) * (B_1 - B_2) - (G_{B1}/V_1) * B_1;$$

$$dB_2/dt = (PS_B/V_2) * (B_1 - B_2) + k_f * A_2 - k_b * B_2;$$

2. Write this model for JSim, using the previous models as examples. Don't forget to include the initial conditions. A correct version of this model is included but do not use it unless all else fails. Learning how to write and debug models is part of the exercise. With A₁₀=A₂₀=A₃₀=A₄₀=0, A_{in} = 1 mM (constant), K_{diss}=2, k_b = 4/sec, G_{1B}=2 ml/sec, V₁=1 ml, V₂=2 ml, P_{sa}=P_{sb}=3 ml/sec, and F= 1 ml/sec, what is the value of the outflow of B₁ at 20 seconds? (Answer: B_{1out}(20 seconds) =.209 mM.)

Question: Identify the sources and sinks in all four equations. It may be necessary to multiply the terms out to separate sources and sinks. (Hint: Sources have plus signs, sinks have minus signs.)

Question: What are some parameter and initial condition combinations to make this model into the (1) Synthesis model, (2) the Exchange model, (3) the Conversion Model and (4) the Flow Model. Indicate what the output variables are for each.

```

import nsrunit; unit conversion on;
/*
Source      +-----+ Sinks
Flow*Alin --> A1, B1, V1, Al0, Bl0          --> Flow*A1
              |                               --> Flow*B1
              |   (Sink) -GB1/V1*B1           |
              |                               |
            Sink     Source       Sink    Source
            +---v----^-----V-----^-----+
                |         |        |        |
             PsA*A1 PsA*A2 PSB*B1 PSb*B2
                |         |        |        |
            +---v----^-----V-----^-----+
               Source   Sink    Source   Sink
               |         |        |        |
                   kf---->                  kf
                 A2 <------> B2 Kdiss= ----
                       <----kb                    kb
                V2, A20, B20
            +-----+

*/
math AllProcessesODE {realDomain t s; t.min=0; t.max=20; t.delta=0.1;

// Dependent Variable and constants
real A1(t) mM, A2(t) mM, B1(t) mM, B2(t) mM, Alout(t) mM, Blout(t) mM,
A10 = 0 mM, A20 = 0 mM, B10 = 0 mM, B20 = 0 mM,
V1 = 1 ml, V2 = 2 ml,
Flow = 1 ml/s, // Flow rate
PSa = 3 ml/s, // Passive exchange rate for A
PSb = 3 ml/s, // Passive exchange rate for B
GB1 = 2 ml/s, // Consumption rate for B in flowing compartment
Kdiss = 3 dimensionless, //Dissociation constant
kb = 4 sec(-1), // Backwards rate constant
kf sec(-1); // Forward rate constant
kf = Kdiss*kb; //
extern real Alin(t) mM; // Alin will be defined at run time using a
                        // function generator

// Initial Condition
when(t=t.min){A1=A10; A2=A20; B1=B10; B2=B20;}

// Ordinary Differential Equation
A1:t = (Flow/V1)*(Alin-A1) + (PSa/V1)*(A2-A1);
A2:t =                      (PSa/V2)*(A1-A2) -kf*A2+kb*B2;
Alout = A1;
B1:t = (Flow/V1)*(-B1) + (PSb/V1)*(B2-B1) -(GB1/V1)*B1;
B2:t =                      (PSb/V2)*(B1-B2) +kf*A2-kb*B2;
Blout=B1;

// No analytic solutions are given for the general case. Compare
// Analytic solutions for special cases where they exist
```

```
} // End of model
```

Model is in AllProcesses.proj.

Lesson VI: Equations for distributed regions (Partial Differential Equations)

Consider a flow carrying a concentration of a metabolite through a tube. In a small region of the tube, we write the equation for mass conservation in one dimension as:

$$d(V * C)/dt = \partial(V * C)/\partial t + U * \partial(V * C)/\partial x = +Sources - Sinks + V * Diffusion.$$

where V is the volume, C is the concentration, U is the velocity, t is the time and x is the spatial dimension.

$$d(V * C)/dt = \partial(V * C)/\partial t + U * \partial(V * C)/\partial x$$

means that the total derivative (also called the material derivative) is equal to the local rate of change plus the advection of a gradient. See any standard text of fluid dynamics for a fuller explanation. To the observer positioned at a particular place along the tube, what is observed is the local rate of change from sources and sinks and the advection of a gradient. The gradient term is moved to the right hand side, and we customarily write,

$$\partial(V * C)/\partial t = -U * \partial(V * C)/\partial x + Sources - Sinks + V * Diffusion.$$

Assuming the volume is constant, the volume can be placed outside the partial derivatives.

$$V * \partial C / \partial t = -U * V * \partial C / \partial x + Sources - Sinks + V * Diffusion.$$

Let $U * V$ the velocity multiplying the volume be rewritten as

$$-U * V = (-Flow / Area) * (Area * L) = -Flow * L,$$

where $Flow$ is the flow rate, commonly given in ml/unit time or (ml/gram of tissue)/unit time, and $Area * L = Volume$ of the capillary, and L is the length. This yields

$$V * \partial C / \partial t = -Flow * L * \partial C / \partial x + Sources - Sinks + V * Diffusion.$$

Dividing both sides by the capillary volume, we have

$$\partial C / \partial t = -(Flow * L / V) * \partial C / \partial x + Sources / V - Sinks / V + Diffusion.$$

Note that if a region has no flow,

$$dC/dt = \partial C / \partial t,$$

and there is no advective term.

How does the partial differential equation compare with the ordinary differential equation for a stirred tank? Consider the simple case with advection and no diffusion, sources or sinks. The ordinary differential equation is given as

$$dC/dt = Flow/V * (C_{in} - C_{out}).$$

With slight rearrangement, this is the same as

$$dC/dt = -Flow * L/V * (C_{out} - C_{in})/L.$$

But

$$(C_{out} - C_{in})/L = (C(L) - C(0))/(L - 0)$$

which is approximately equal to dC/dx .

Axial diffusion is neither a source nor a sink. It redistributes the molecules in the region. In a region with no flow, the diffusion equation is

$$\partial C / \partial t = D * \partial^2 C / \partial x^2$$

where D is the diffusion coefficient in cm^2/sec .

When advection is included, we have the classic advection-diffusion equation,

$$\partial C / \partial t = -(F * L/V) * \partial C / \partial x + D * \partial^2 C / \partial x^2.$$

Sources and Sinks:

We encounter the same sources and sinks in distributed models as in compartmental models: synthesis and decay, exchange with other regions, conversion of one chemical species to another.

Recall the complex problem of Section V:

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

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

$$dA_1/dt = (Flow/V_1) * (A_{in} - A_1) - (PS_A/V_1) * (A_1 - A_2);$$

$$dA_2/dt = (PS_A/V_2) * (A_1 - A_2) - k_f * A_2 + k_b * B_2;$$

$$dB_1/dt = (Flow/V_1) * (-B_1) - (PS_B/V_1) * (B_1 - B_2) - (G_{B1}/V_1) * B_1;$$

$$dB_2/dt = (PS_B/V_2) * (B_1 - B_2) + k_f * A_2 - k_b * B_2;$$

In a distributed model with axial diffusion, the last four equations are given by

$$\partial A_1 / \partial t = -(Flow * L / V_1) * \partial A_1 / \partial x - (PS_A / V_1) * (A_1 - A_2) + D_{A1} * \partial^2 A_1 / \partial x^2;$$

$$\partial A_2 / \partial t = (PS_A / V_2) * (A_1 - A_2) - k_f * A_2 + k_b * B_2 + D_{A2} * \partial^2 A_2 / \partial x^2;$$

$$\partial B_1 / \partial t = -(Flow * L / V_1) * \partial B_1 / \partial x - (PS_B / V_1) * (B_1 - B_2) - (G_{B1} / V_1) * B_1 + D_{B1} * \partial^2 B_1 / \partial x^2;$$

$$\partial B_2 / \partial t = (PS_B / V_2) * (B_1 - B_2) + k_f * A_2 - k_b * B_2 + D_{B2} * \partial^2 B_2 / \partial x^2;$$

With compartment Models we had initial conditions and inflow and outflow were incorporated in the ordinary differential equations. With partial differential equations, we have initial conditions, and inflows, outflows, and regions with neither incorporate this information in boundary conditions on the left and right hand side of the regions. It is customary to place the inflowing material at the left boundary and outflowing material at the right boundary.

For an inflowing boundary (such as for species A_1 and B_1 in the above example) the total flux boundary condition is given as

$$-(Flow * V_1 / L) * (A_1 - A_{in_1}) + D_{A1} * \partial A_1 / \partial x = 0, \text{ and}$$

$$-(Flow * V_1 / L) * (B_1 - 0) + D_{B1} * \partial B_1 / \partial x = 0 \text{ where } A_{in_1} \text{ is the inflowing concentration at}$$

$x = x.min.$. A no flux boundary condition is used at the exit end of the flowing regions and all ends of non-flowing regions. We have the follow,

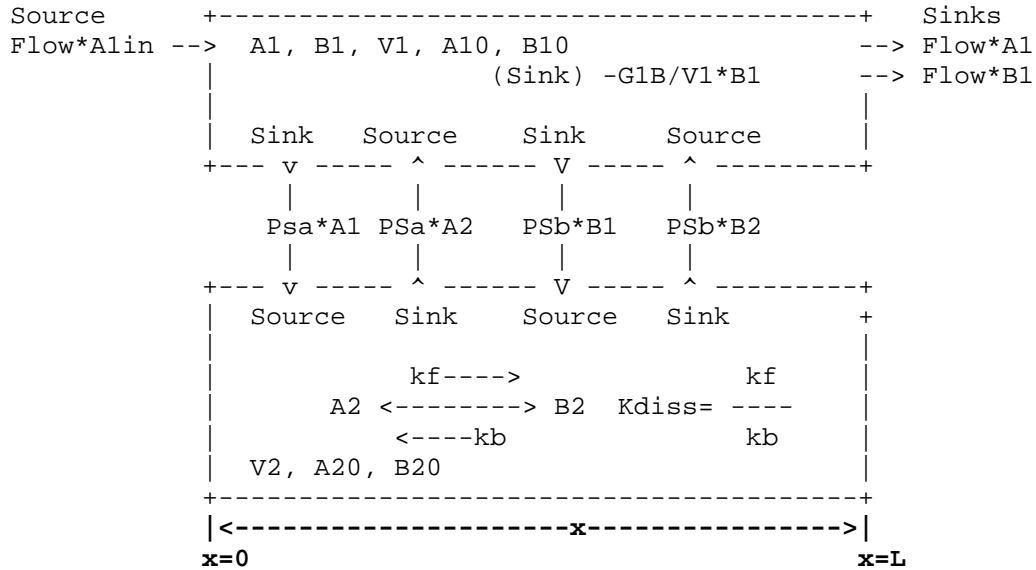
$$\partial A_1 / \partial x = 0, \quad \partial B_1 / \partial x = 0, \quad \partial A_2 / \partial x = 0, \text{ and } \partial B_2 / \partial x = 0 \text{ at } x = x.max; \text{ and also}$$

$$\partial A_2 / \partial x = 0 \text{ and } \partial B_2 / \partial x = 0 \text{ at } x = x.min.$$

The main differences in the code are highlighted in boldface. Code for the PDE version is given as

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

```
/*
```



```
*/
```

```
math AllProcessesPDE {
```

```
real L=0.1 cm, Ngrid=31;
```

```
realDomain x cm; x.min=0; x.max=L; x.ct=Ngrid;
```

```
realDomain t s; t.min=0; t.max=20; t.delta=0.1;
```

```
// Dependent Variable and constants
```

```
real A1(x,t) mM, A2(x,t) mM, B1(x,t) mM, B2(x,t) mM, Alout(t) mM, Blout(t) mM,
```

```
A10 = 0 mM, A20 = 0 mM, B10 = 0 mM, B20 = 0 mM,
```

```
V1 = 1 ml, V2 = 2 ml,
```

```
Flow = 1 ml/s, // Flow rate
```

```
PSa = 3 ml/s, // Passive exchange rate for A
```

```
PSb = 3 ml/s, // Passive exchange rate for B
```

```
G1B = 2 ml/s, // Consumption rate for B in flowing compartment
```

```
D1 = 1e-5 cm^2/s, // Axial diffusion in V1
```

```
D2 = 1e-5 cm^2/s, // Axial diffusion in V2
```

```
Kdiss = 3 dimensionless, //Dissociation constant
```

```
kb = 4 sec^(-1), // Backwards rate constant
```

```
kf sec^(-1); // Forward rate constant
```

```
kf = Kdiss*kb; //
```

```
extern real Alin(t) mM; // Alin will be defined at run time using a
```

```
// function generator
```

```
// Initial Condition
```

```
when(t=t.min) {A1=A10; A2=A20; B1=B10; B2=B20;}
```

```
// Boundary conditions
```

```
when (x=x.min) {-(Flow*L/V1)*(A1-Alin)+D1*A1:x=0;
```

```
-(Flow*L/V1)*(B1-0)+D2*B1:x=0;
```

```
A2:x=0; B2:x=0;}
```

```
when (x=x.max) {A1:x=0; B1:x=0;
```

```
Alout=A1; Blout=B1;
```

```
A2:x=0; B2:x=0; }
```

```
// Partial Differential Equation
```

```
A1:t = (-Flow*L/V1)*A1:x +D1*A1:x:x + (PSa/V1)*(A2-A1) ;
```

```

A2:t =                                     +D2*A2:x:x + (PSa/V2)*(A1-A2)      -kf*A2+kb*B2;

B1:t = (-Flow*L/V1)*B1:x +D1*B1:x:x + (PSb/V1)*(B2-B1)      -(G1B/V1)*B1;
B2:t =                                     +D2*B2:x:x + (PSb/V2)*(B1-B2)      +kf*A2-kb*B2;

// No analytic solutions are given for the general case. Compare
// Analytic solutions for special cases where they exist
// Making D1 and D2 large ( 0.2 cm^2/sec) will make this model behave like
// the compartmental version

} // End of model

```

The ODE and PDE versions are in the same project AllProcessesPDE.proj