Blood-Tissue Exchange Models Tutorial

Introduction

We have already seen the simple compartment models for flow, exchange, reaction, and consumption. The compartment models are described mathematically with ordinary differential equations. These differential equations depend only on time. An assumption of compartment models is that their contents are instantaneously well mixed. For the one compartment model with flow, the concentration at the outflow is the same everywhere inside the compartment. There are no gradients. Diffusion in the compartment is assumed to be infinite.

The approximate dimensions of a capillary are 5 to 8 um in diameter and 1000 um in length. For simple molecules in water the diffusion coefficient is approximately 10^{-5} cm²/sec. It would take 1000 seconds for a simple molecule in water to diffuse from one end of the capillary to the other end. The compartmental assumption that diffusion is infinite is not valid on short time scales.

Deriving the governing partial differential equation for a region with flow

We introduce partial differential equations in time and one spatial dimension. We usually call the spatial dimension "*x*". What follows next is the development of the partial differential equations for multiple regions.

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

$$\frac{d(V \cdot C)}{dt} = +Sources - Sinks + \frac{\partial}{\partial x}\left(D \cdot \frac{\partial}{\partial x}(V \cdot C)\right) \quad ,$$

where *V* is the volume, assumed constant, and *D* is the diffusion coefficient. The partial differential equation is similar to the equation for the one compartment model with flow with the addition of the diffusion term. The derivative on the left hand side of this equation is known as the "material derivative" or "Lagrangian derivative" and is equal to the local rate of change plus the advection of a gradient (See any standard text on fluid dynamics for a full explanation):

$$V \cdot \frac{dC}{dt} = V \cdot \frac{\partial C}{\partial t} + U \cdot V \cdot \frac{\partial C}{\partial x} = +Sources - Sinks + V \cdot \frac{\partial}{\partial x}\left(D \cdot \frac{\partial}{\partial x} C\right)$$

where *U* is the velocity of the fluid. The advection of the gradient is subtracted from both sides giving

$$V \cdot \frac{\partial C}{\partial t} = -U \cdot V \cdot \frac{\partial C}{\partial x} + Sources - Sinks + V \cdot \frac{\partial}{\partial x}\left(D \cdot \frac{\partial}{\partial x} C\right) \quad .$$

Simplifying the velocity of the flow multiplying the volume as

$$U \cdot V = \left(\frac{Flow}{Area}\right) \cdot (Area \cdot L) = Flow \cdot L$$

where *Area* is the cross sectional area of the capillary and *L* is its length, and dividing both sides by the

volume, *V*, yields

$$\frac{\partial C}{\partial t} = -\frac{Flow \cdot L}{V} \cdot \frac{\partial C}{\partial x} + \frac{Sources - Sinks}{V} + \frac{\partial}{\partial x}\left(D \cdot \frac{\partial}{\partial x} C\right) \quad .$$

This equation in JSim's Mathematical Modeling Language (MML) is written as

```
C:t = -(Flow*L/V)*C:x + Sources/V — Sinks/V + (D*C:x):x;
```

Comparison with ODE model formulation

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

$$\frac{dC}{dt} = \frac{Flow}{V} \cdot (C_{in} - C_{out}).$$ With slight rearrangement, this is the same as

$$\frac{dC}{dt} = -\frac{Flow \cdot L}{V} \cdot \frac{(C_{out} - C_{in})}{L} \quad .$$ However, $\frac{(C_{out} - C_{in})}{L}$ in a distributed model would become

$$\frac{(C(L) - C(0))}{(L - 0)} \approx \frac{\partial C}{\partial x}$$ when *L* is small.

Partial differential equations for a two region model

The typical equations for a two region model with exchange and consumption are given by

$$\frac{\partial C_1}{\partial t} = -\frac{Flow \cdot L}{V_1} \cdot \frac{\partial C_1}{\partial x} + \frac{PS}{V_1} \cdot (C_2 - C_1) - \frac{G_1}{V_1} \cdot C_1 + D_1 \cdot \frac{\partial^2 C_1}{\partial x^2}$$

$$\frac{\partial C_2}{\partial t} = +\frac{PS}{V_2} \cdot (C_1 - C_2) - \frac{G_2}{V_2} \cdot C_2 + D_2 \cdot \frac{\partial^2 C_2}{\partial x^2} \quad .$$

In MML, they are written as
```
C1:t =-(Flow*L/V1)*C1:x+(PS/V1)*(C2-C1)-(G1/V1)*C1+D1*C1:x:x;
C2:t=                   (PS/V2)*(C1-C2)—(G2/V2)*C2+D2*C2:x:x;
```

NOTA BENE:
- 🔴 It was assumed above that the diffusion coefficients are constant. However, this is not necessary and the diffusion terms can be written as `(D*C:x):x`, where D is not constant.
- 🔴 The stagnant region does not have an advection term.
- 🔴 The exchange term is divided by V_1 in the first equation and V_2 in the second equation.
- 🔴 The diffusion term in the second equation is mandatory, even if the diffusion coefficient, D_2 is zero (e.g. the second region indicates stationary binding sites, stationary transporter sites, etc.) Omitting the diffusion term causes the system of the two partial differential equations to become uncoupled and the second equation is treated as a set of ordinary differential equations with unpredictable results.

Boundary Conditions

Left inflow total flux boundary condition (Robin condition) $-\dfrac{Flow \cdot L}{V} \cdot C + D \cdot \dfrac{\partial C}{\partial x} = -\dfrac{Flow \cdot L}{V} \cdot C_{\text{in}}$ is

written in MML as

```
when(x=x.min) -(Flow*L/V)*(C-Cin) + D*C:x=0.;   // Flow>0
```
Corresponding right outflow (Neumann) boundary condition is
```
when(x=x.max) C:x=0;
```
The PDE is given as
```
C:t = -(Flow*L/V)*C:x + D*C:x:x;
```

Right inflow total flux boundary condition is written in MML as
```
when(x=x.max) -(Flow*L/V)*(C-Cin) + D*C:x=0.;   // Flow<0
```
Corresponding left outflow (Neumann) boundary condition is
```
when(x=x.min) C:x=0;
```
The PDE is given as
```
C:t = -(Flow*L/V)*C:x + D*C:x:x;
```


No flux (Neumann) boundary conditions (usually for non-flowing regions)
```
when(x=x.min) C:x=0;
when(x=x.max) C:x=0;
```

Initial Conditions

Initial conditions are usually given as either
```
when(t=t.min) C=0;
```
or
```
when(t=t.min) C=C0(x);
```
where *C0* is either given as a constant or a function of *x*.

If the first form is used, the initial condition is assumed to be a function of x, and a parameter is generated, `C__init(x)` with the value of `0` (zero) which can be replaced by a function generator.

Oscillating Flows
An oscillating flow problem is modeled in BTEX10_OscillatingFlow.proj.

Putting it all together

The typical regions for these type of transport problems are plasma (p), endothelial cells (ec), the interstitial fluid region (isf),parenchymal cells (pc), and mitochondria (mito).

The exchange rates between the regions are given as PSg for the gap between the plasma and the interstitial fluid region; Psecl for the barrier between the endothelial cell and the plasma on the lumenal side (facing the lumen or capillary) of the endothelial cell; PSeca for the barrier between the endothelial cell and the interstitial fluid region on the ablumenal side (facing away from the lumen or capillary) of the endothelial cell; PSpc for the barrier between the interstitial fluid region and the parenchymal cell; and PSmito for the barrier between the parenchymal cell and the mitochondria.

Consumption rates in these regions are given as Gp, Gec, Gisf, Gpc, and Gmito.

Diffusion rates are given as Dp, Dec, Disf, Dpc, and Dmito.

The plasma volume is given as Vp. All the other volumes are considered to be volumes of distribution, that is they may be larger than the physical volumes associated with the regions to account for the affinity of a metabolite in a particular region, such as oxygen in the red blood cell. The volumes of distribution are given as V'isf, V'ec, V'pc, and V'mito, but usually spelledas Visfp, Vecp, Vpcp, and Vmitop, where the p stands for the prime.

BTEX50 Partial Differential Equations in JSim's Mathematical Modeling Language
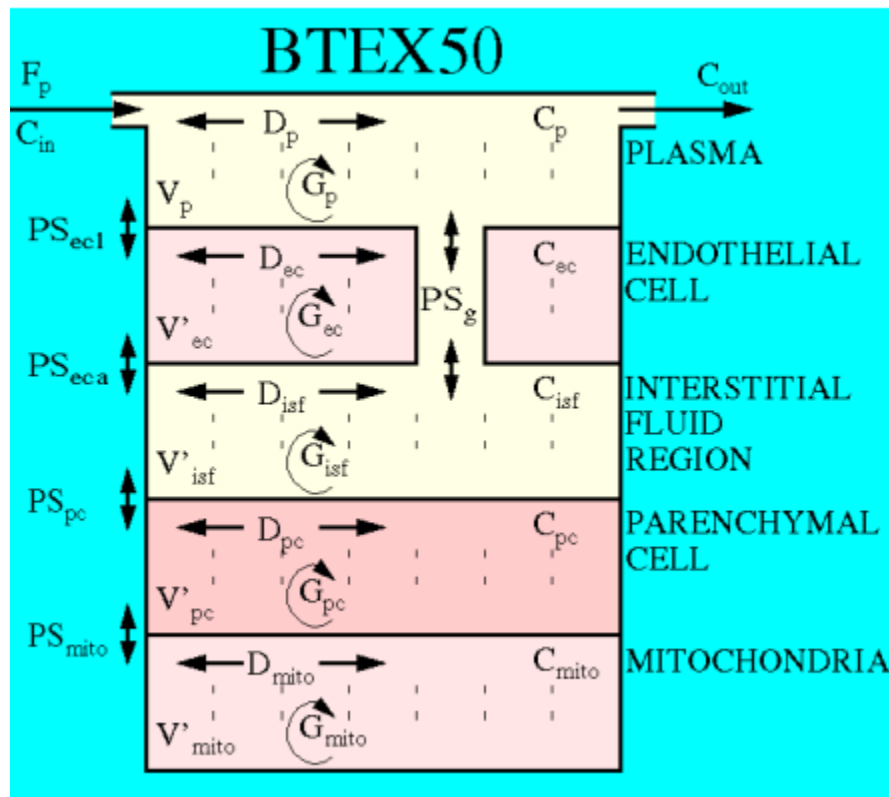
```
Cp:t = Fp*L/Vp*Cp:x -Gp/Vp*Cp + Dp*Cp:x:x          + Psg/Vp*(Cisf-Cp)
                                                   + Psecl/Vp*(Cec-Cp);

Cisf:t =      -Gisf/Visfp*Cisf + Disf*Cisf:x:x     + PSg/Visfp*(Cp-Cisf)
                                                   + Pseca/Visfp*(Cec-Cisf)
                                                   + PSpc/Visfp*(Cpc-Cisf);

Cec:t =            -Gec/Vecp*Cec + Dec*Cec:x:x     + Psecl/Vecp*(Cp-Cec)
                                                   + PSeca/Vecp*(Cisf-Cec);

Cpc:t =            -Gpc/Vpcp*Cpc + Dpc*Cpc:x:x     + Pspc/Vpcp*(Cisf-Cpc)
                                                   + Psmito/Vpcp*(Cmito-Cpc);

Cmito:t = -Gmito/Vmitop*Cmito + Dmito*Cmito:x:x + PSmito/Vmitop*(Cpc-Cmito);
```