

3rd MEI International Symposium
November 30 to December 2, 2008
San Francisco, California



Reusable Modular Code for Multi-Scale
Physiological Systems Modeling

Gary M. Raymond, Erik A. Butterworth, and
James B. Bassingthwaighe,
Department of Bioengineering,
University of Washington, Seattle



Advances In Modeling

- CellML and FieldML
 - Unambiguous specification of the mathematical definition of models
 - IDE, the Integrated Development Environment
- SBML for the sharing of computational models
- ISML multilevel biophysical models
 - Structural and logical relations between modules specified by edges leading to modular, hierarchical and/or network representations
- JSim supports CellML and SBML



Introduction

- Importance of Models and Modeling Systems
- Roadblocks to achieving useful systems
- Modularity
 - Name space and name space rules
 - Using syntactic replacement (an old idea) and merging equations
- Example of developing Modularity
 - Flow1, Exchange2, Convert2
 - Creating the new Model
- Future directions



Modeling Physiological Systems

- Multi-scalar: Composed of elements operating at different scales in space and time.
- Diverse processes: Example: blood flow in organ carrying chemical species metabolized in cells involves
 - Transport through multiple capillaries with different flow rates, Transport across cell membranes, Chemical Reactions, and Sequestration and/or Washout.
- Elements or modules in hierarchical arrangements.



Mathematical Models and Modeling Systems

- Key to understanding and gaining insight into the behavior, modification, and control of physiological systems.
- (Beard, Bassingthwaite, and Greene, 2005) models facilitate
 - “Organizing information.”
 - “Thinking logically about what components and interactions are important.”
 - “Simulating, predicting, and optimizing procedures, experiments, and therapies.”
 - “Disproving hypotheses and defining improved hypotheses.”



If Mathematical Models and Modeling are KEY, what are the LOCKS?

- So unavailable. “Show me the code!”
- So complex that only their authors can understand them.
- So unverified. “Hidden assumptions.”
- So undocumented.
- So incorrect sometimes.
- So un-modular. The “Stand Alone Model.”
- So time-consuming to generate and code new models
- So what (are)...



Potential Rewards of Modular Model Construction?

- Ease communication of models and concepts among teachers and investigators.
- Reproducibility of published information.
- Collaborations among investigators of like interests.
- Ease in modifying models to create alternative hypotheses.
- Common standards for models, the screws and bolts that hold modern science together into an integrated and testable form.



Building Systems for Modular Multi-scale Modeling

- CAVEATS:
 - Is modeling becoming too difficult?
 - No single system able to solve all possible kinds of problems.
 - A system that will do everything for everybody will be so complicated that no one will use it.



“One small step ...”

- We need modeling systems that avoid the locks we have already mentioned. Models and modeling systems need to
 - Be simple enough so that our colleagues can understand them.
 - Have verification standards for the mathematical algorithms utilized, and comparison with even simpler models.
 - Be well-documented.
 - Be available from the internet.
 - Be standardized.
 - Be modular.
 - Allow for rapid assembly of new models.

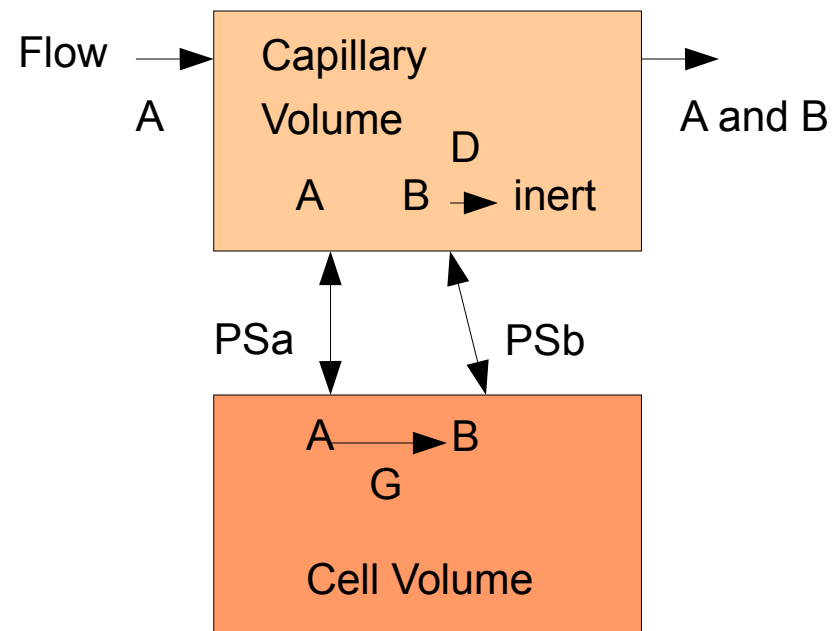


An Approach to Modularity

- Models consist of a **name space (the interface)** and **rules** governing the name space
- **Name space** consists of the following items:
 - Variables
 - Parameters
- **Rules** governing the name space consists of
 - Equations which govern and relate the name space items
- Converting models to modules (simple and old idea of variable name (string) replacement)

Reusable Modular Code with JSim: Example: Indicator-Dilution Model

- Simple example
 - Chemical species A
 - In flowing capillary region
 - Able to be transported back and forth between capillary region and cell region
 - Species A converts to chemical species B in the cell
 - Chemical species B able to be transported back and forth between cell region and capillary region
 - Chemical species B undergoes further degradation to inert substance.





Modules Required

- 2 one-compartment modules with flow.
- 2 two-compartment modules with exchange between the two compartments.
- 1 module for conversion of a chemical species into another chemical species.
- 1 module for degradation of a species.



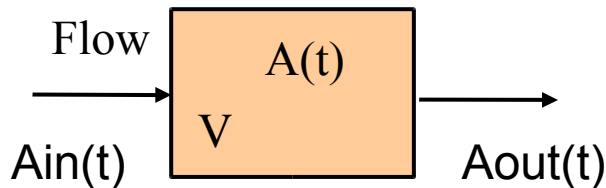
How a new model is generated

- For each use of every module:
 - Locate the code.
 - Copy the code.
 - Create new code using string replacement algorithms (old idea).
- For all modules:
 - Modify the order of the code if necessary.
 - Add in additional code as necessary.
 - Combine fluxes, i.e., merge differential equations.
- Write out new model/module.

Automated Combining of Modules: Flow Embedding directives as comments `//%`

Flow1.mod

One Compartment Model with Flow

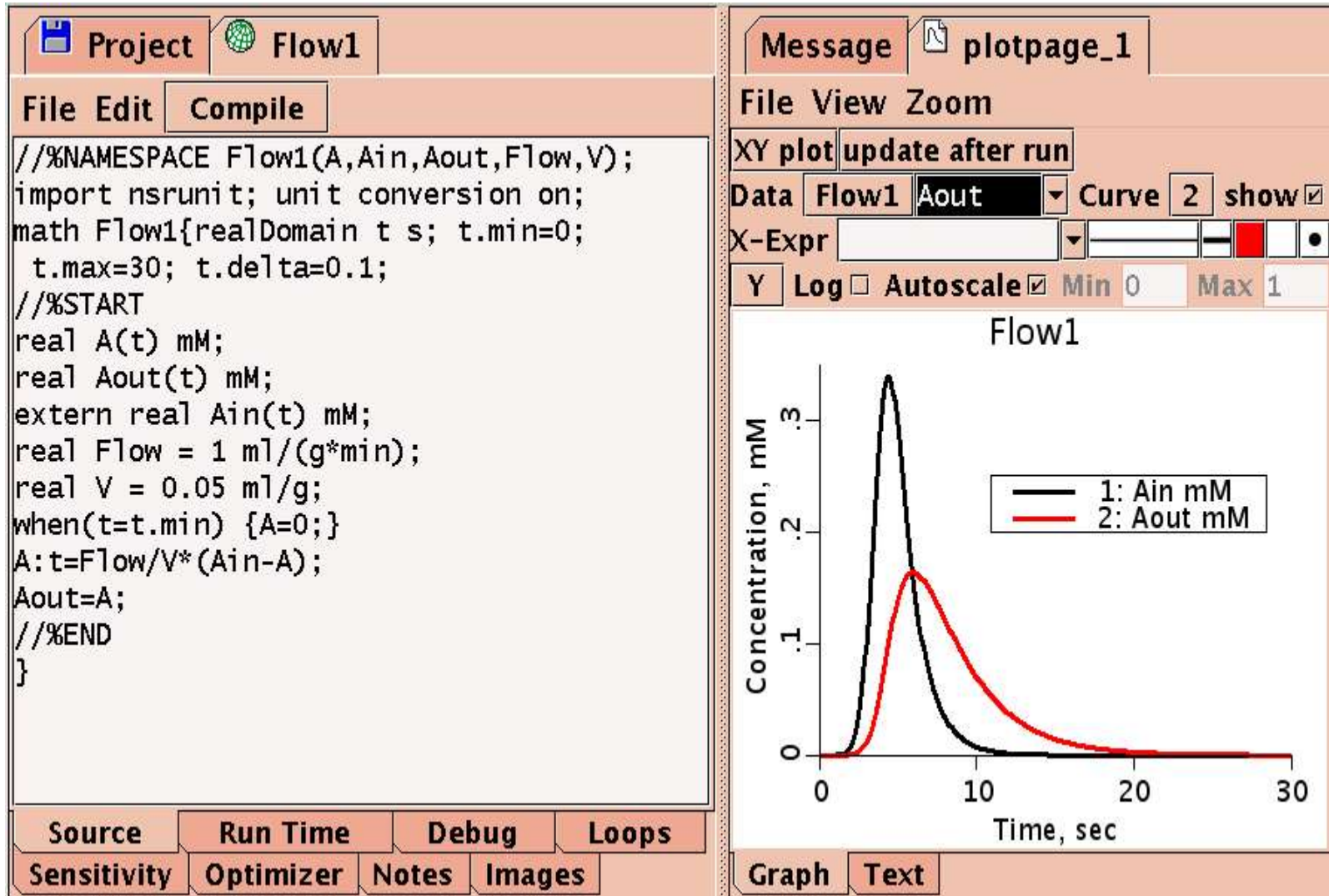


$$dA/dt = Flow/V * (A_{in} - A_{out})$$

```
//%NAMESPACE Flow1(A,Ain,Aout,Flow,V)
import nsrunit; unit conversion on;
math Flow1{realDomain t s; t.min=0;
  t.max=30; t.delta=0.1;
//%START
real A(t) mM;
real Aout(t) mM;
extern real Ain(t) mM;
real Flow = 1 ml/(g*min);
real V = 0.05 ml/g;
when(t=t.min) {A=0;}
A:t=Flow/V*(Ain-A);
Aout=A;
//%END
}
```

Flow Model (module)

Directives don't affect model



The screenshot displays a software interface for a flow model. The left pane shows the source code for a module named 'Flow1'. The code defines variables for concentration (A, Aout), flow rate (Flow), and volume (V), and includes a differential equation for the concentration of A over time. The right pane shows a plot titled 'Flow1' with 'Concentration, mM' on the y-axis and 'Time, sec' on the x-axis. The plot shows two curves: a black curve for '1: Ain mM' and a red curve for '2: Aout mM'. The black curve peaks at approximately 0.35 mM around 5 seconds, while the red curve peaks at approximately 0.15 mM around 7 seconds. The interface also includes a message window, file view, and various control buttons like 'Run Time', 'Debug', and 'Loops'.

```
//%NAMESPACE Flow1(A,Ain,Aout,Flow,V);
import nsrunit; unit conversion on;
math Flow1{realDomain t s; t.min=0;
  t.max=30; t.delta=0.1;
//%START
real A(t) mM;
real Aout(t) mM;
extern real Ain(t) mM;
real Flow = 1 ml/(g*min);
real V = 0.05 ml/g;
when(t=t.min) {A=0;}
A:t=Flow/V*(Ain-A);
Aout=A;
//%END
}
```

Message plotpage_1

File View Zoom

XY plot update after run

Data Flow1 Aout Curve 2 show

X-Expr

Y Log Autoscale Min 0 Max 1

Flow1

Concentration, mM

Time, sec

1: Ain mM
2: Aout mM

Source Run Time Debug Loops

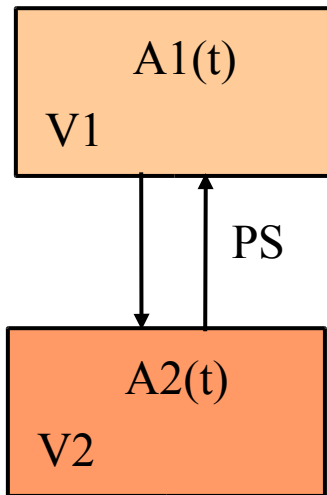
Sensitivity Optimizer Notes Images

Graph Text



Automated Combining of Modules: Permeability-Surface Area Conductance

Exchange2.mod



$$dA1/dt = PS/V1 * (A2 - A1)$$

$$dA2/dt = PS/V2 * (A1 - A2)$$

```
//%NAMESPACE Exchange2(A1,V1,A2,V2,PS)
import nsrunit; unit conversion on;
math Exchange2{ realDomain t s; t.min=0;
  t.max=30; t.delta=0.1;
//%START
real V1 = 0.05 ml/g;
real V2 = 0.6 ml/g;
real A1(t) mM;
real A2(t) mM;
real PS = 1 ml/(g*min);
when(t=t.min) {A1=0;}
when(t=t.min) {A2=0;}
A1:t=PS/V1*(A2-A1);
A2:t=PS/V2*(A1-A2);
//%END
}
```

Exchange Model (module)

Project Exchange2

File Edit Compile

```
//%NAMESPACE Exchange2(A1,V1,A2,V2,PS);
import nsrunit; unit conversion on;
math Exchange2{ realDomain t s; t.min=0;
  t.max=30; t.delta=0.1;
//%START
real V1 = 0.05 ml/g;
real V2 = 0.6 ml/g;
real A1(t) mM;
real A2(t) mM;
real PS = 1 ml/(g*min);
when(t=t.min) {A1=0;}
when(t=t.min) {A2=0;}
A1:t=PS/V1*(A2-A1);
A2:t=PS/V2*(A1-A2);
//%END
}
```

Message E2

File View Zoom

XY plot update after run

Data Exchange2 A2 Curve 2 show

X-Expr

Y Log Autoscale Min 0 Max 1

Exchange2

Time (sec)	A1 Concentration (mM)	A2 Concentration (mM)
0	1.0	0.0
5	0.4	0.1
10	0.15	0.25
15	0.08	0.35
20	0.05	0.4
25	0.04	0.45
30	0.03	0.48

Concentration, mM

Time, sec

1: A1 mM
2: A2 mM

Source Run Time Debug Loops

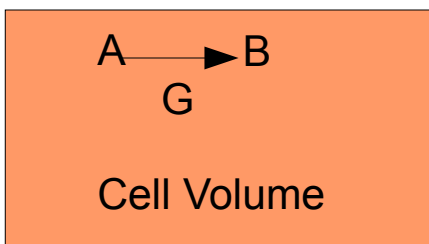
Sensitivity Optimizer Notes Images

Graph Text



Automated Combining of Modules: Conversion of Substance into Product

Convert2.mod

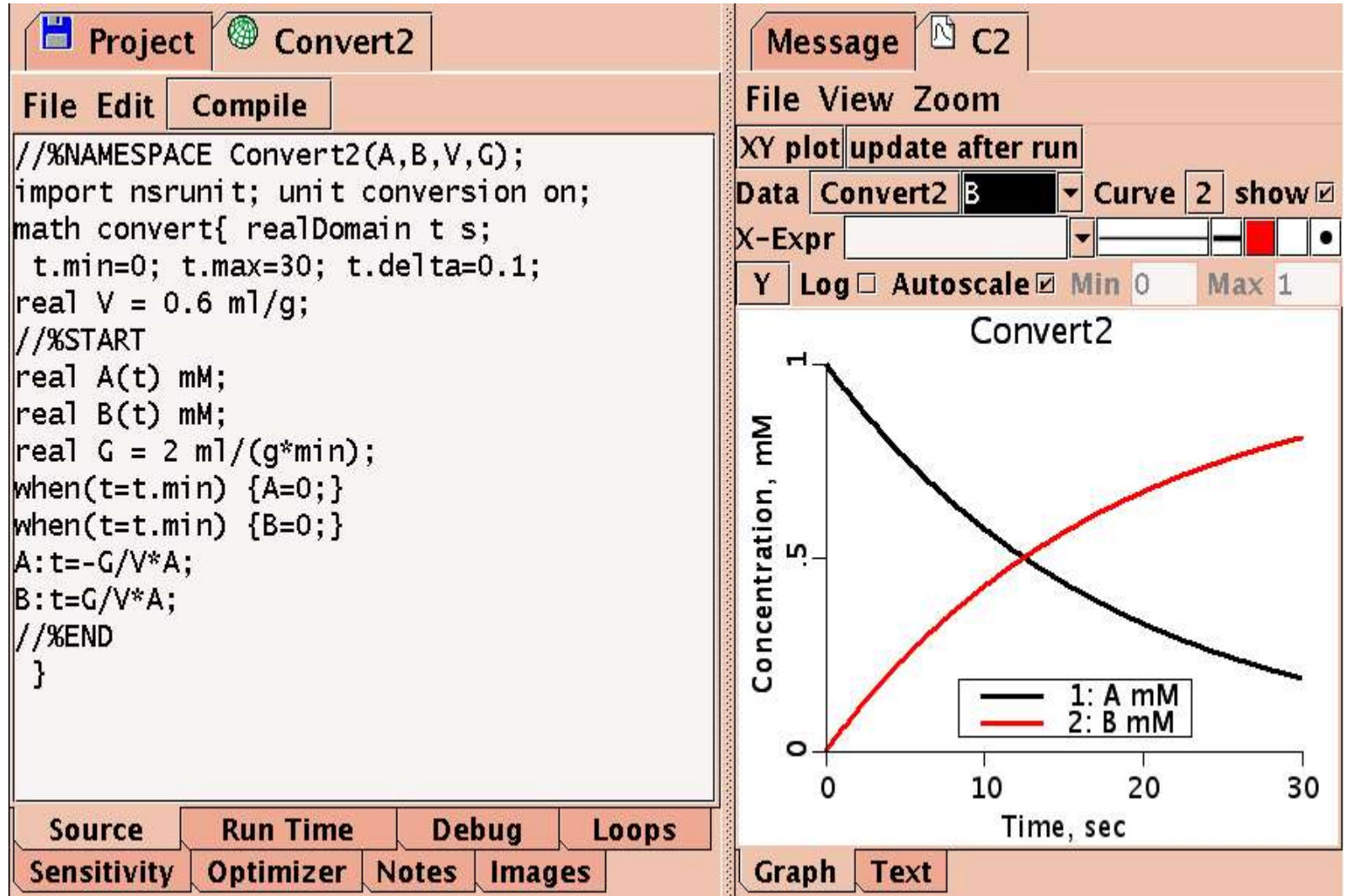


$$dA/dt = -G/V * A$$

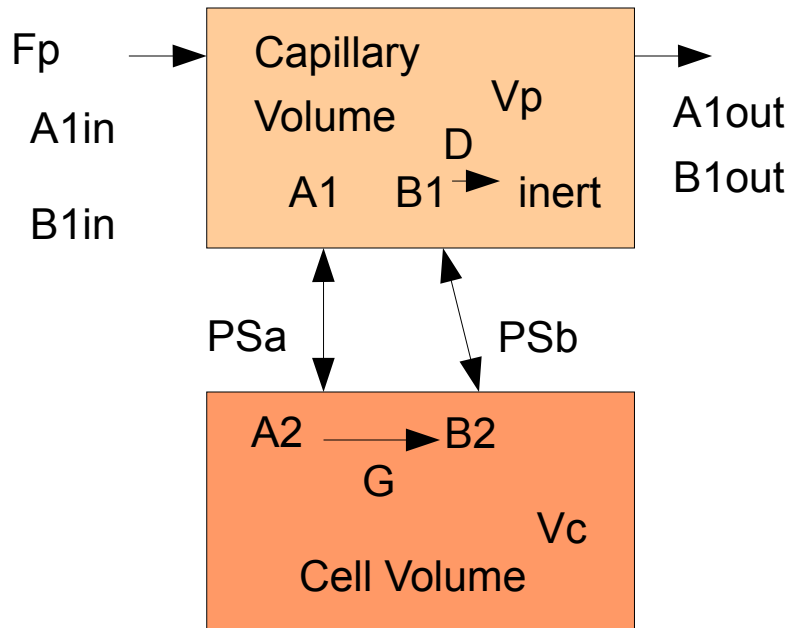
$$dB/dt = G/V * A$$

```
//%NAMESPACE Convert2(A,B,V,G)
import nsrunit; unit conversion on;
math convert{ realDomain t s;
  t.min=0; t.max=30; t.delta=0.1;
  real V = 0.6 ml/g;
  //%START
  real A(t) mM;
  real B(t) mM;
  real G = 2 ml/(g*min);
  when(t=t.min) {A=0;}
  when(t=t.min) {B=0;}
  A:t=-G/V*A;
  B:t=G/V*A;
  //%END
}
```

Conversion Model (module)



Automated Combining of Modules: A New Model



```

import nsrunit; unit conversion on;
math All {realDomain t sec; t.min = 0;
t.max = 30.0; t.delta = 0.1;
//%EXTERNAL Flow1 Flow1.mod
//%EXTERNAL Exchange2 Exchange2.mod
//%EXTERNAL Convert2 Convert2.mod
//%RENAME Flow1(A1,A1in,A1out,Fp,Vp)
//%RENAME Flow1(B1,B1in,B1out,Fp,Vp)
//%RENAME Exchange2(A1,Vp,A2,Vc,PSa)
//%RENAME Exchange2(B1,Vp,B2,Vc,PSb)
//%RENAME Convert2(A2,B2,Vc,G)
//%START
real D = 1 ml/(g*min);
B1:t=-D/Vp*B1;
//%END
//%PROCESS
}

```



Generated Code for new model (module)

```
import nsrunit; unit conversion on;
math All {realDomain t sec; t.min = 0;
  t.max = 30.0; t.delta = 0.1;
real A1(t) mM;
real A1out(t) mM;
extern real A1in(t) mM;
real Fp = 1 ml/(g*min);
real Vp = 0.05 ml/g;
real B1(t) mM;
real B1out(t) mM;
extern real B1in(t) mM;
real Vc = 0.6 ml/g;
real A2(t) mM;
real PSa = 1 ml/(g*min);
```

```
real B2(t) mM;
real PSb = 1 ml/(g*min);
real G = 2 ml/(g*min);
real D = 1 ml/(g*min);
when(t=t.min) {A1=0;}
A1:t=Fp/Vp*(A1in-A1)+(PSa/Vp*(A2-A1));
A1out=A1;
when(t=t.min) {B1=0;}
B1:t=Fp/Vp*(B1in-B1)+(PSb/Vp*(B2-B1))+(-D/Vp*B1);
B1out=B1;
when(t=t.min) {A2=0;}
A2:t=PSa/Vc*(A1-A2)+(-G/Vc*A2);
when(t=t.min) {B2=0;}
B2:t=PSb/Vc*(B1-B2)+(G/Vc*A2);
}
```



The Ordinary Differential Equations have been automatically combined

$$A1:t=Fp/Vp*(A1in-A1)+(PSa/Vp*(A2-A1));$$

$$B1:t=Fp/Vp*(B1in-B1)+(PSb/Vp*(B2-B1))+(-D/Vp*B1);$$

$$A2:t=PSa/Vc*(A1-A2)+(-G/Vc*A2);$$

$$B2:t=PSb/Vc*(B1-B2)+(G/Vc*A2);$$

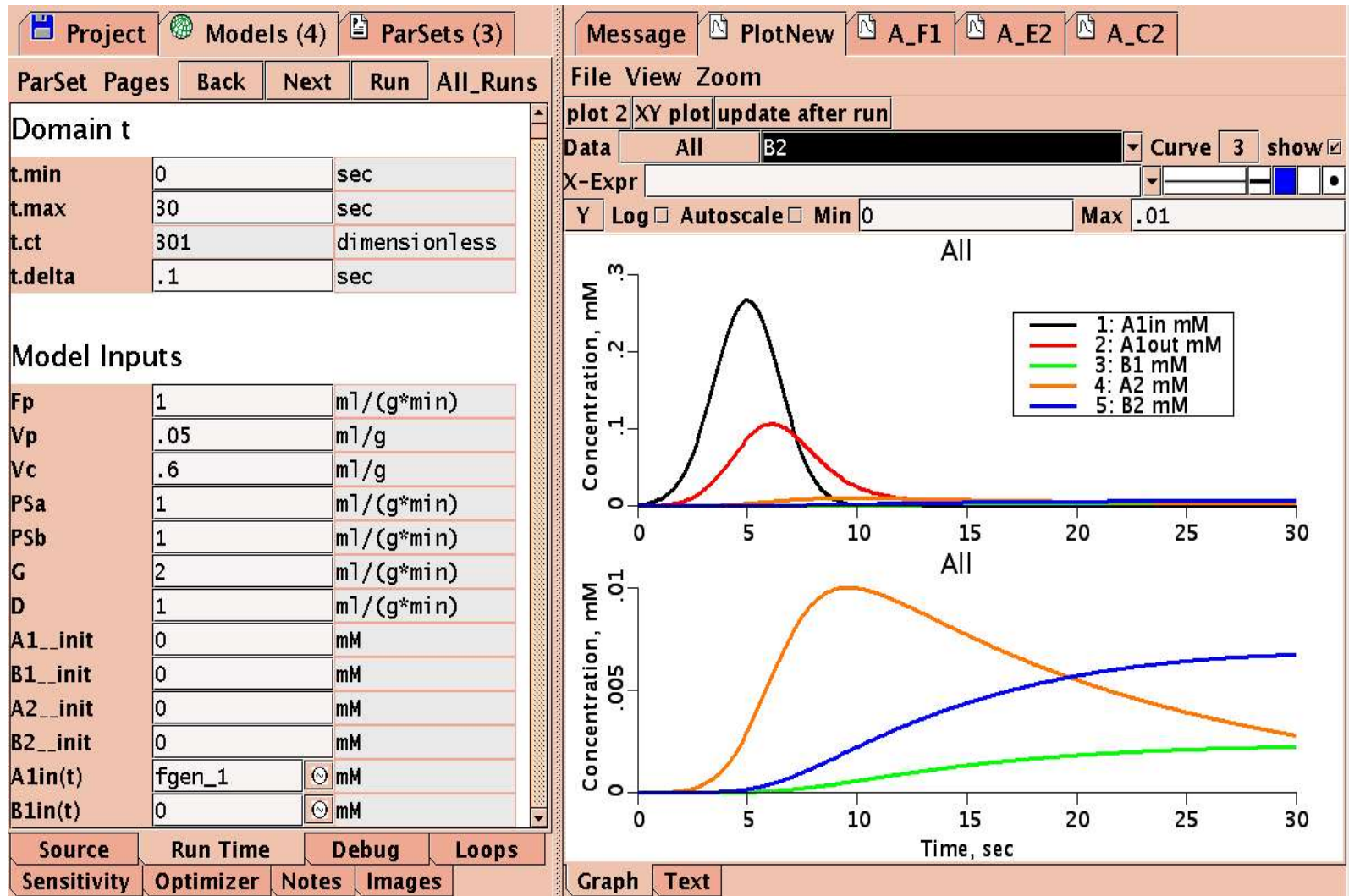
$$dA1/dt = Fp/Vp*(A1in - A1) + PSa/Vp*(A2 - A1)$$

$$dB1/dt = Fp/Vp*(B1in - B1) + PSb/Vp*(B2 - B1) - D/Vp*B1$$

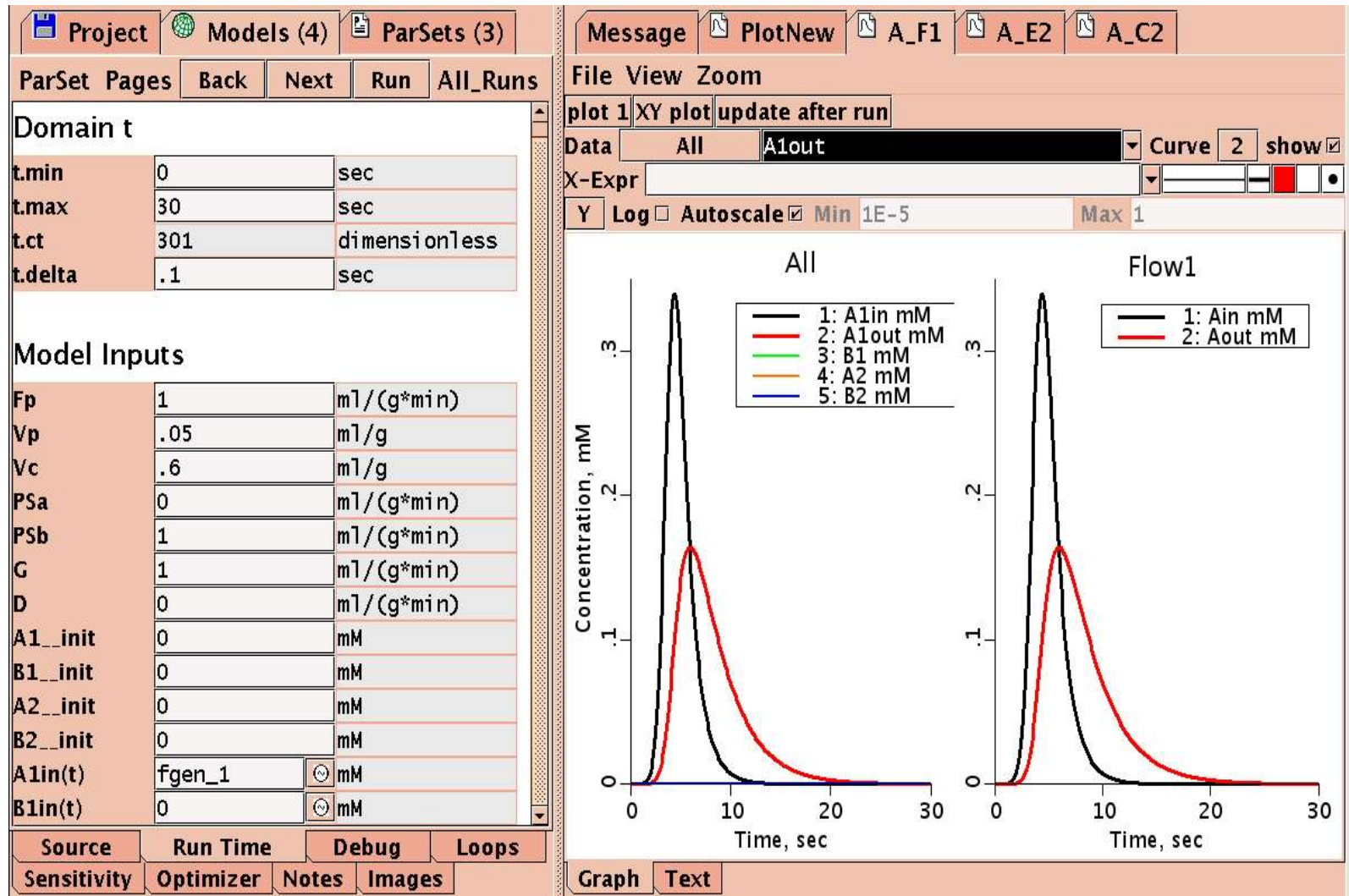
$$dA2/dt = PSa/Vc*(A1 - A2) - G/Vc*A2$$

$$dB2/dt = PSb/Vc*(B1 - B2) + G/Vc*A2$$

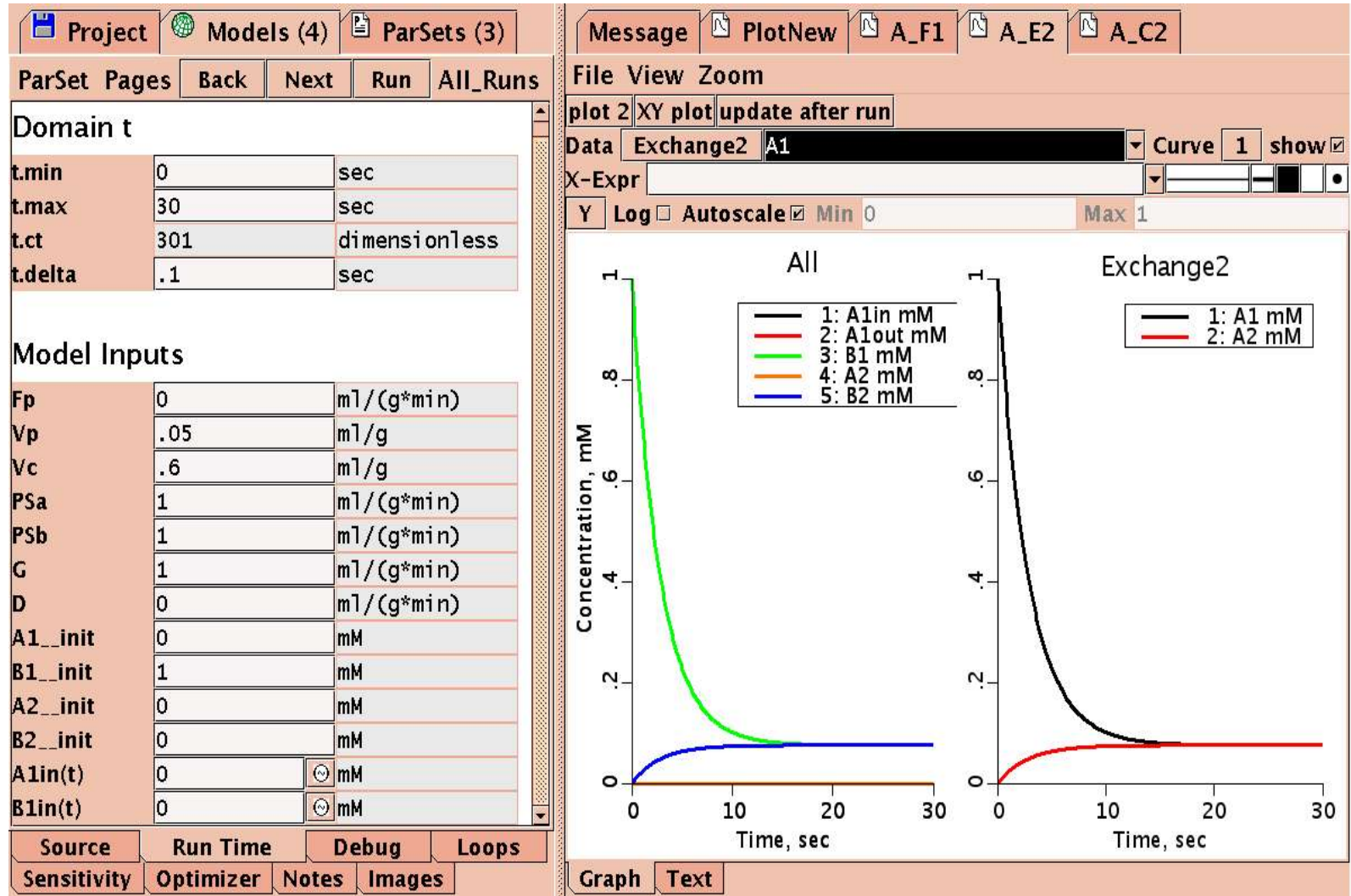
Did the Automatic Combining of Modules work? Yes.



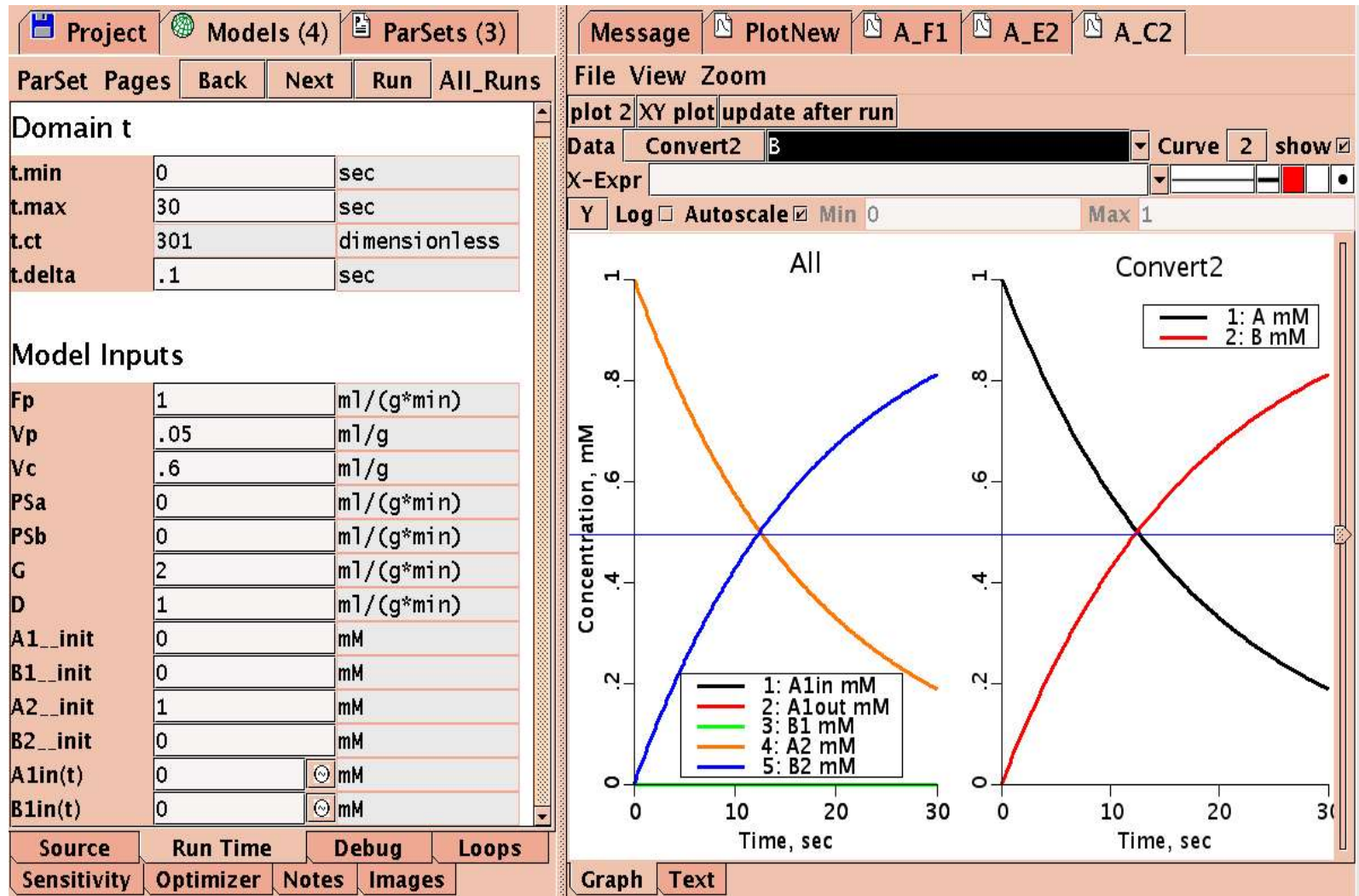
Validation of new model: Comparison with Flow model: Set conductance PSa to 0



Validation of new model: Comparison with Exchange Model: $F_p=0$, $D=0$, $B1_init=1$



Validation of new model: Comparison with Convert Model: $PSa=0$, $PSb=0$, $A2_init=1$.





Automated Combining of Modules: What has been done so far

- Multi-species, multi-compartmental models with flow, facilitated (competitive) transport across the membrane, and with three different types of metabolic models:
 - Direct conversion of one species to another
 - Michaelis-Menten conversion of one species to another
 - Enzyme conversion of one species to another
- Serial PDE models for O_2 and CO_2 transport and blood-tissue exchange.



Possible Future Directions

Build Modular Replacement Processing into Jsim.

Omit the NAMESPACE directive

```
//%NAMESPACE Flow1(A,Ain,Aout,Flow,V);
```

Replace with

```
//%RENAME Flow1(V=Vp,Flow=Fp,A=A1,Ain=A1in,Aout=A1out)
```

also allow

```
//%RENAME Flow1(V=Vp=0.07 ml/g, ... and
```

```
//%RENAME Flow1(V=Vp=
```

```
('DatabaseFileName',VariableNameInDatabase), ... .
```

Multiple `//%START(n)` and `//%STOP(n)` directives.

Allow for `//%//%DIRECTIVES` for embedding modules within modules.

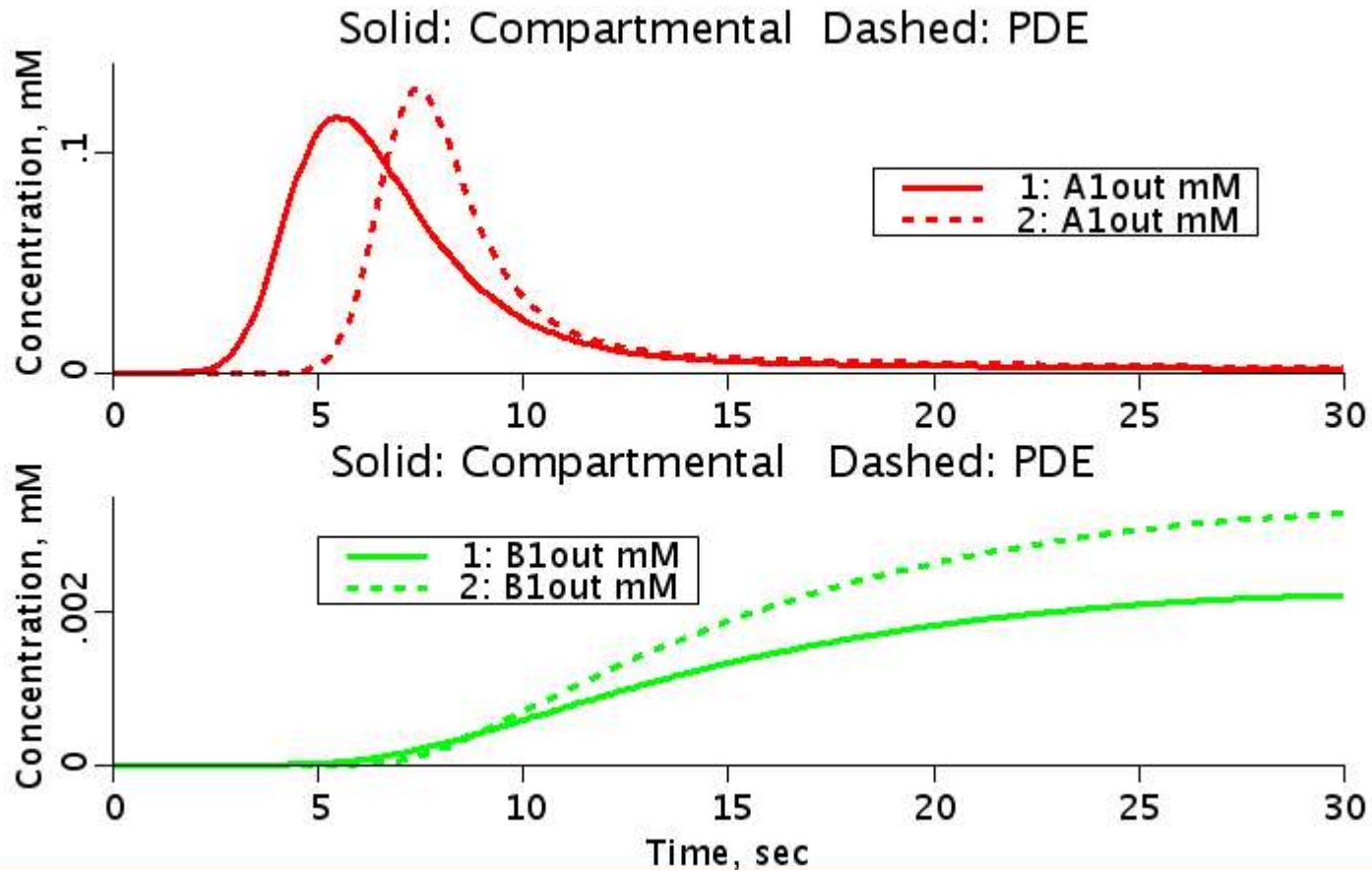


Final Point

Paraphrasing Norton Juster, “What if the model answer is right but the question is wrong?”

Take the new model we just wrote and make a companion model with partial differential equations. Compartment models have an assumption in instantaneous mixing. A capillary is much longer than wide and what comes out is not instantaneously equilibrated with what is flowing in. Diffusion is not infinite. How does finite diffusion affect the amount of species B, the product, that is washed out. Is it less, equal, or greater in models with limited diffusion compared with compartment models?

Comparison Compartmental and PDE Models: Instantaneous Mixing vs. Diffusion: More B produced in limited diffusion model





Thank you.

Questions
Comments
Criticisms