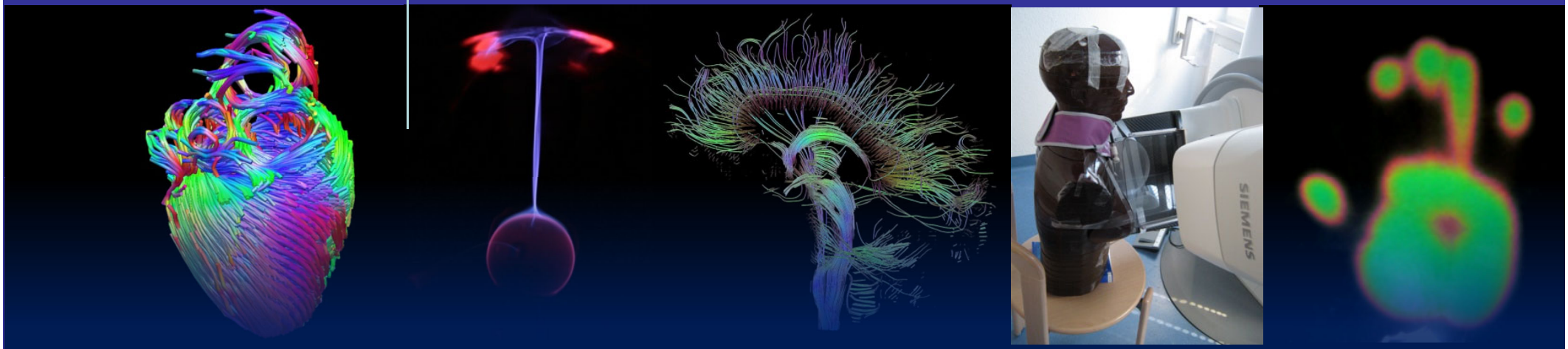
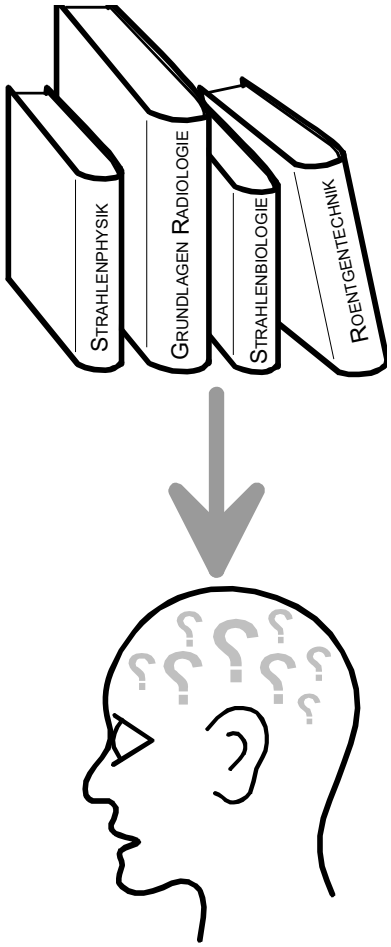
Using Graphical Model Editors

Hyperboost Training Course Model-based Data Analysis for Clinical Applications

Stephan Scheidegger
Medical Biophysics Group ZHAW
2024



CONTENT MBDA



Model-based data analysis for clinical application – Modelling and Biological Systems:

Day 1

0920-1100: Modelling and Biological Systems

1320-1400: Using Graphical Model Editors

1400-1450: Using Python for model fitting

Day 2

1110-1200: Biokinetic / Biodynamic Modelling (→ Lab2: Model-based Data Analysis of PET Images)

Day3

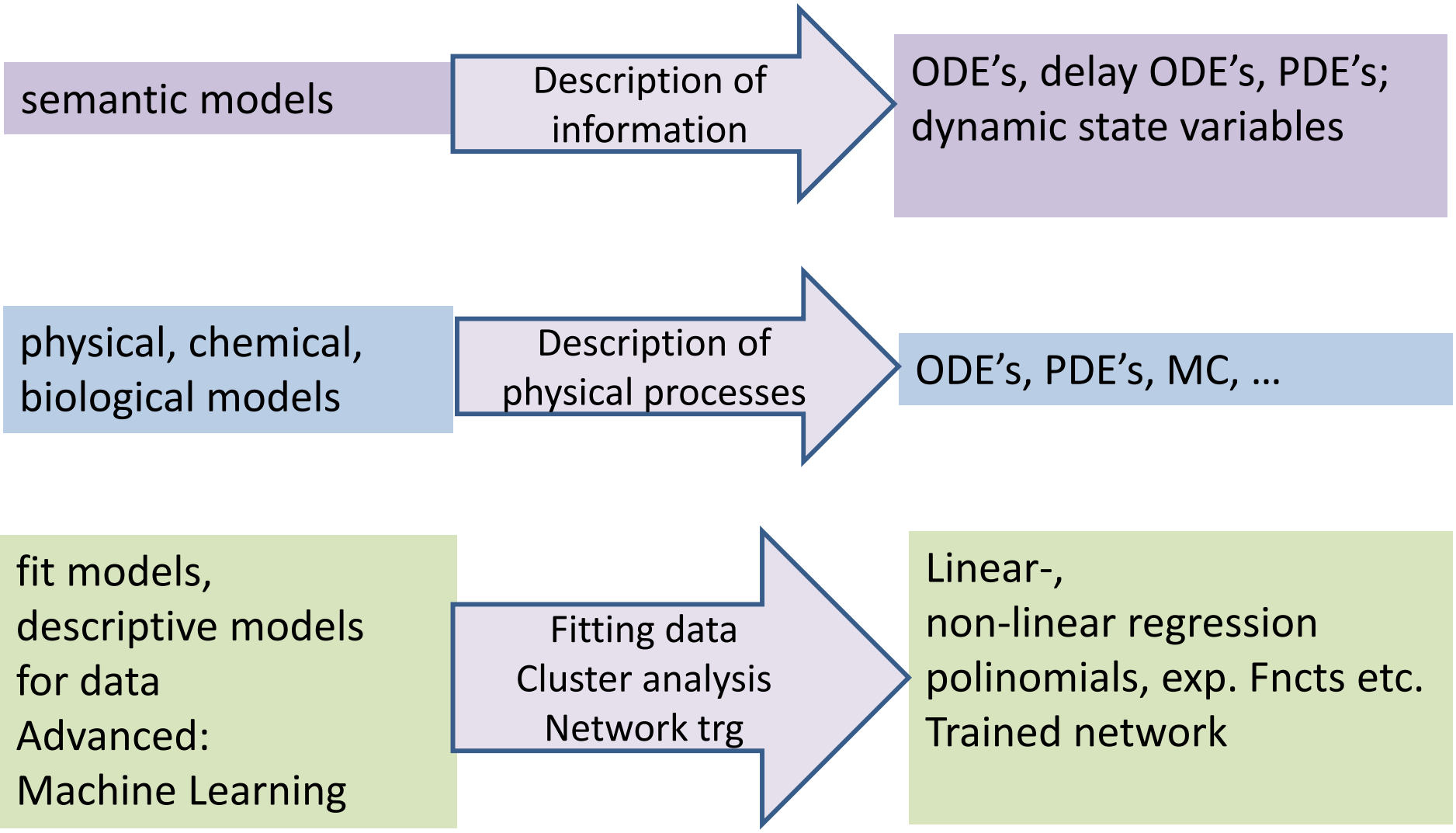
0900-1100: Radiobiological Models

Learning Objectives

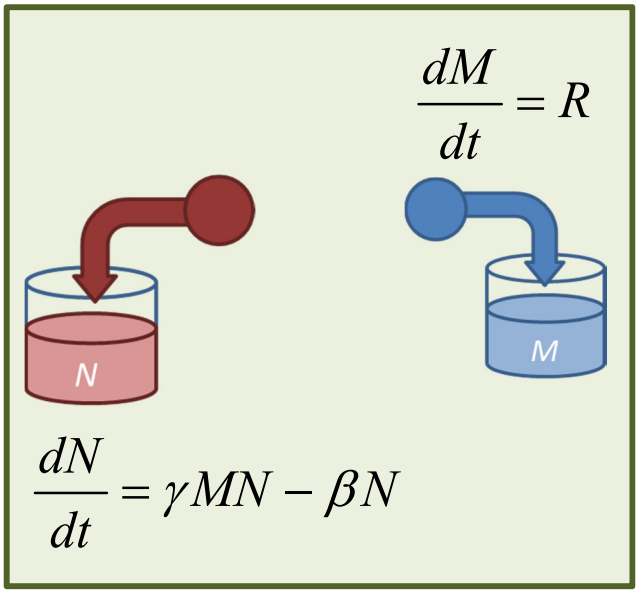
Students are able

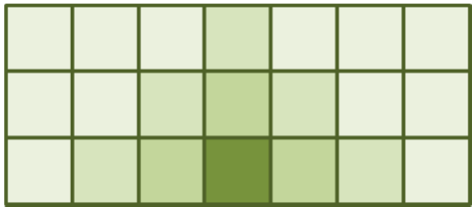
- To be aware of the different purposes of modelling
- to explain the assumption for compartmental models
- to model compartmental biological systems and explore them by using computer simulations
- to use models for biological data analysis
- To use modelling and computer simulation as in silico lab tools

Catching the Real World in Models



Modelling Approaches and Techniques





$$\frac{dn}{dt} = \gamma mn - \beta n + \kappa \cdot \left(\frac{\partial^2 n}{\partial x^2} + \frac{\partial^2 n}{\partial y^2} \right) \pm \ldots$$

Depending on the purpose, different modelling approaches can be useful:

- Compartmental models: Mathematical description by ordinary differential equations (ODE) or delay differential equations (DDE); simulation using finite difference methods
- Spatio-temporal models: Mathematical description by partial differential equations (PDE), simulation using time-domain finite difference (TD-FD) methods or finite elements methods (FEM)

Modelling and Computer Simulation

$$\frac{dN}{dt} = f(N,t)$$

To perform "biological experiment" in silico, numerical solving of the model equations is needed (in case of DE-based models):

$$\frac{\Delta N}{\Delta t} = f(N,t)$$

$$\Delta N = f(N,t) \cdot \Delta t$$

- numerical integration using Euler's or Runge-Kutta Method for ODE
- Time-Domain Finite Difference (TD-FD) Method for PDE
- Finite Element Method (FEM) for PDE
- Deep Learning

$$N(t) = N(t-\Delta t) + \Delta N(t-\Delta t) =$$
$$N(t-\Delta t) + f(N(t-\Delta t), t-\Delta t) \cdot \Delta t$$

Modelling and Computer Simulation

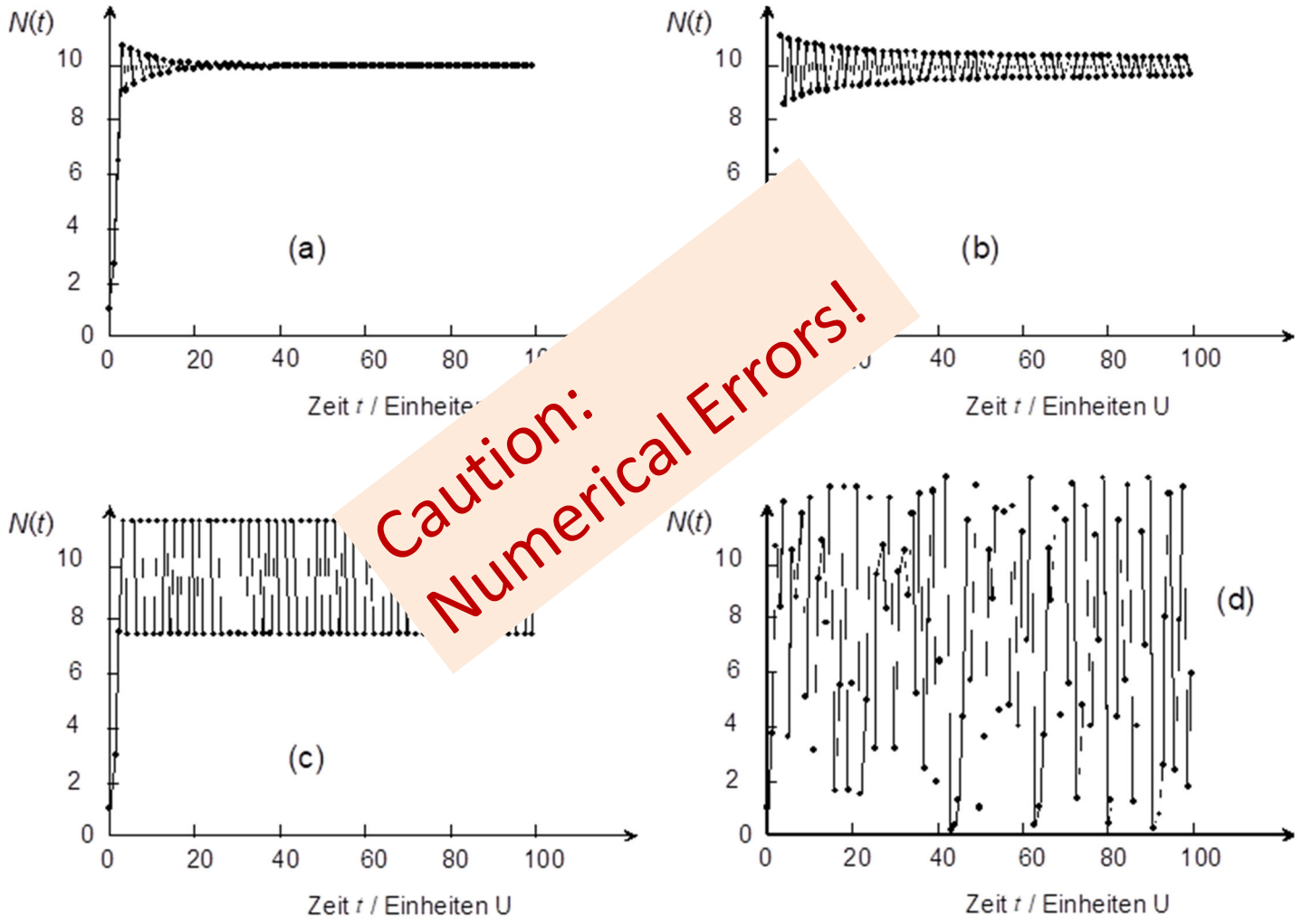
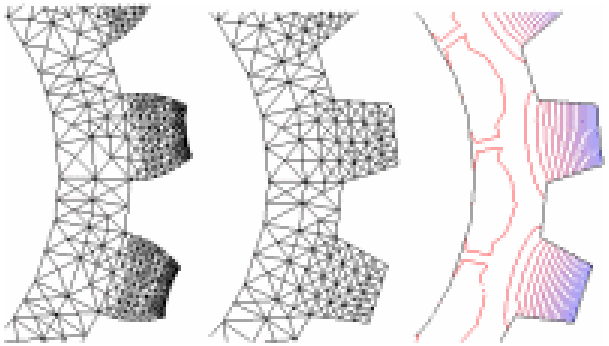
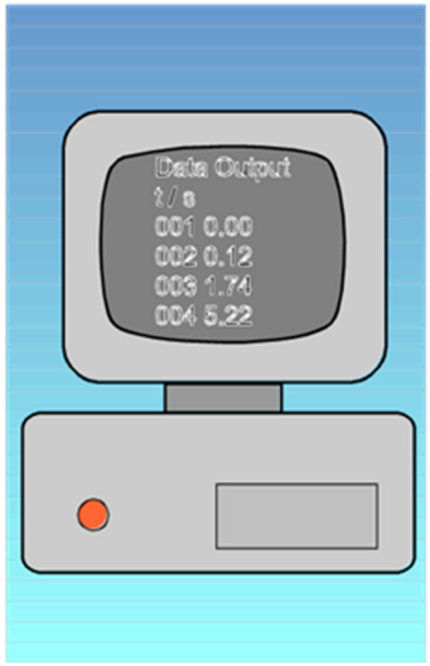


Fig.1. Numerical oscillations in the system $\frac{dN}{dt} = \alpha N - \beta N^2$

Modelling and Computer Simulation

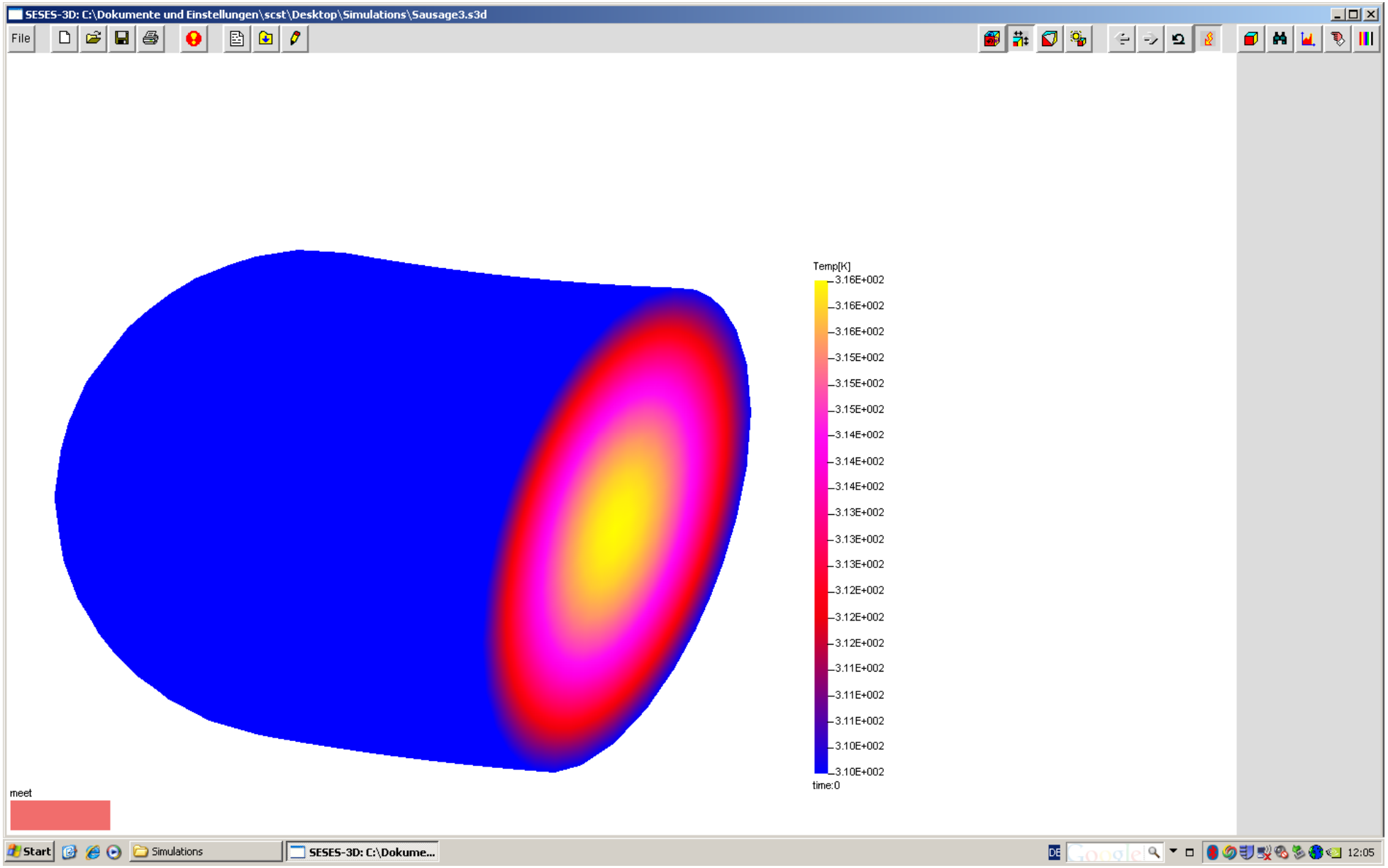




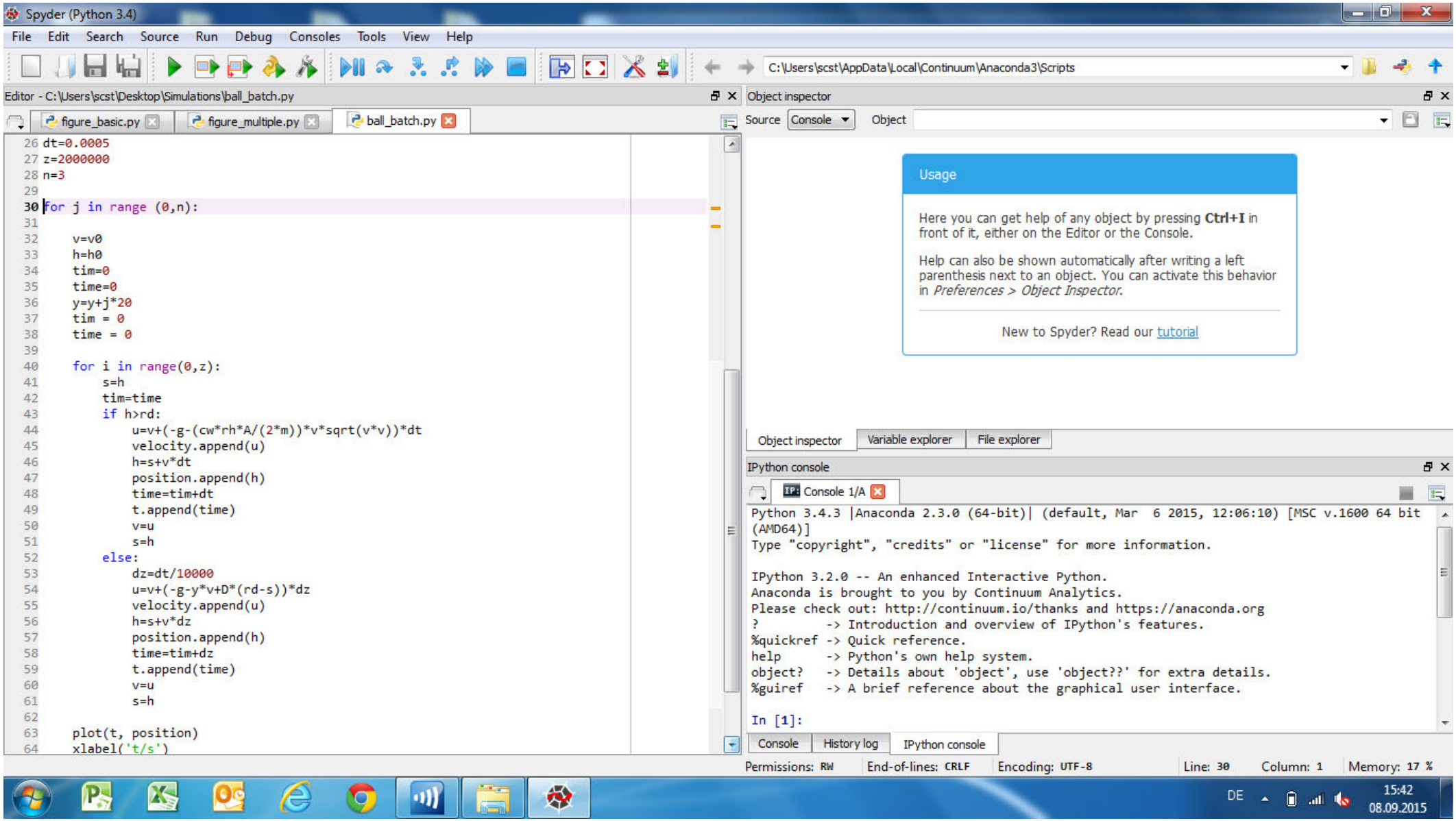
Different tools for implementation are available:

- (high level) programming languages without or with libraries (e.g. Python, Matlab, Octave, …)
- Graphical modelling tools; generation of a mark-up language code which can be feeded into a solver (e.g. Systems Biology Mark-up Language SBML; M. Hucka et al.: The Systems Biology Markup Language (SBML: A medium for representation and exchange of biochemical network models. In: *Bioinformatics*. Vol. 19, no. 4, 2003, S. 524–531)

Modelling and Computer Simulation: FEM Multi-Physics Tools



Modelling and Computer Simulation: Spyder (Python Editor)



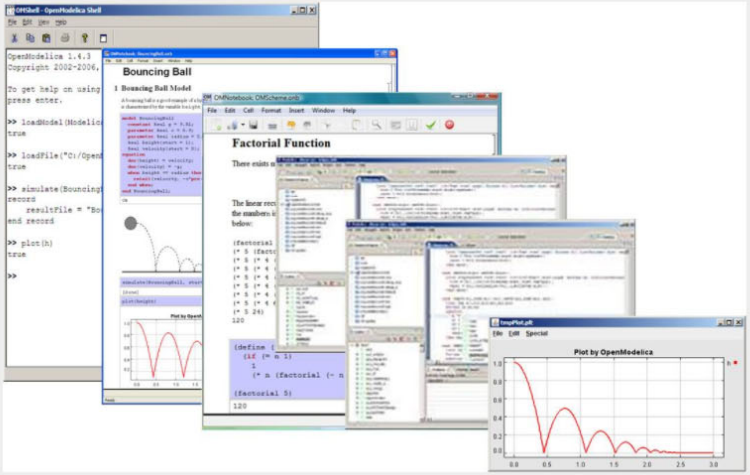
OpenModelica

Home Download Users Developers Events Research

Introduction

OPENMODELICA is an open-source Modelica-based[1] modeling and simulation environment intended for industrial and academic usage. Its long-term development is supported by a non-profit organization – the Open Source Modelica Consortium (OSMC). An overview journal paper is available and slides about Modelica and OpenModelica.

The goal with the OpenModelica effort is to create a comprehensive Open Source Modelica modeling, compilation and simulation environment based on free software distributed in binary and source code form for research, teaching, and industrial usage. We invite researchers and students, or any interested developer to participate in the project and cooperate around OpenModelica, tools, and applications.



Join the OpenModelicaInterest mailing list to get information about new releases.

Help us: get the latest source code or nightly-build and report bugs.

To learn about Modelica, read a book or a tutorial about Modelica.

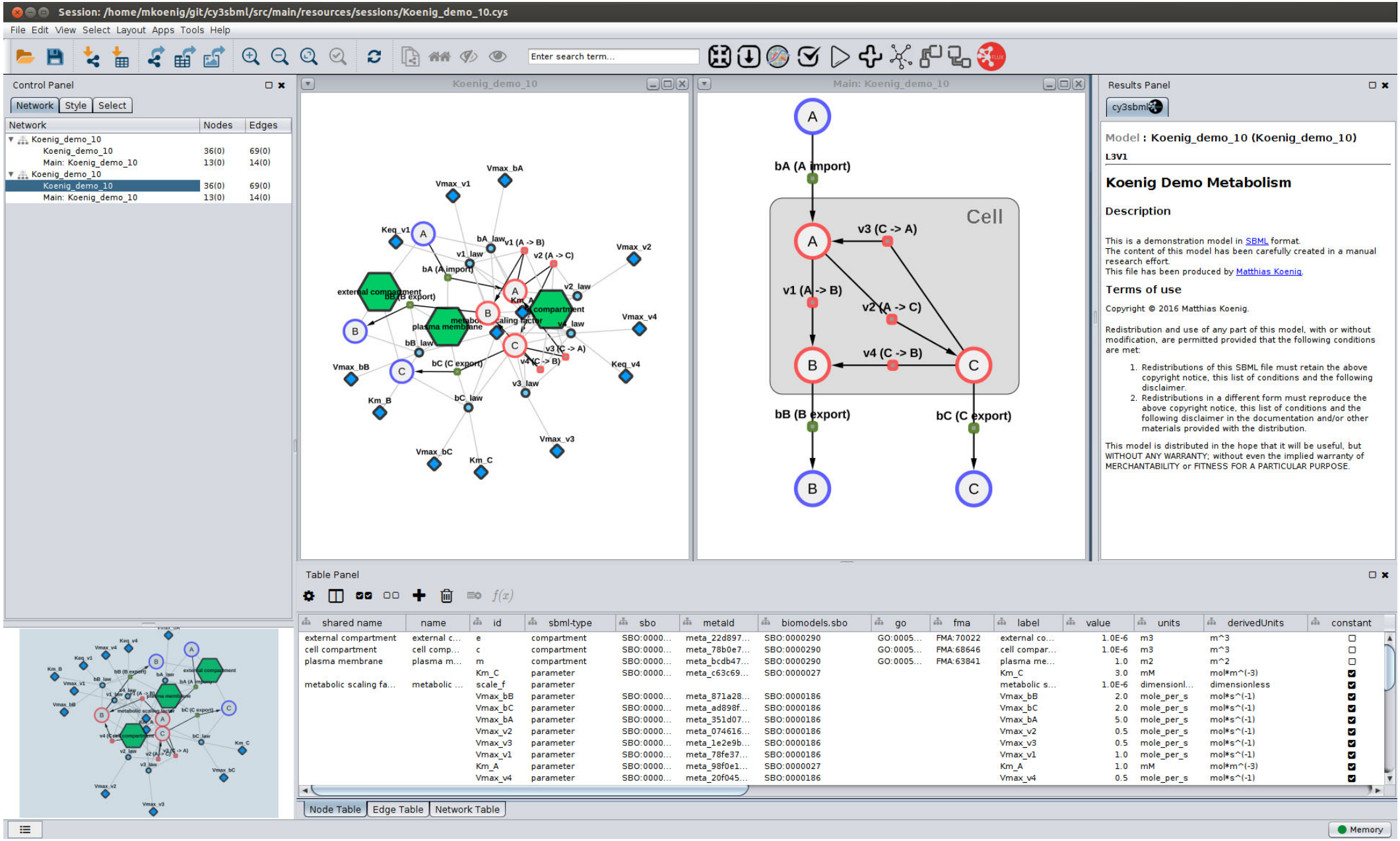
Interactive step-by-step beginners Modelica on-line spoken tutorials Interactive OMWebbook with examples of Modelica textual modeling and textbook companions with application OpenModelica exercises. A Jupyter notebook Modelica mode, available in OpenModelica.

To get advice how to make existing Modelica libraries work in OpenModelica, see Porting.

For systems engineering with requirement traceability and verification, see ModelicaML.

OpenModelica provides library coverage reports of open-source Modelica libraries showing which libraries work well with OpenModelica and how the support improved over time.

Systems Biology Markup Language



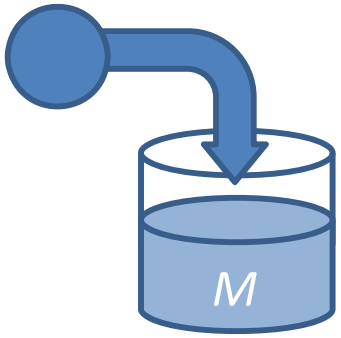
Modelling and Computer Simulation: Graphical Model-Builder for Compartmental Systems

Semantics

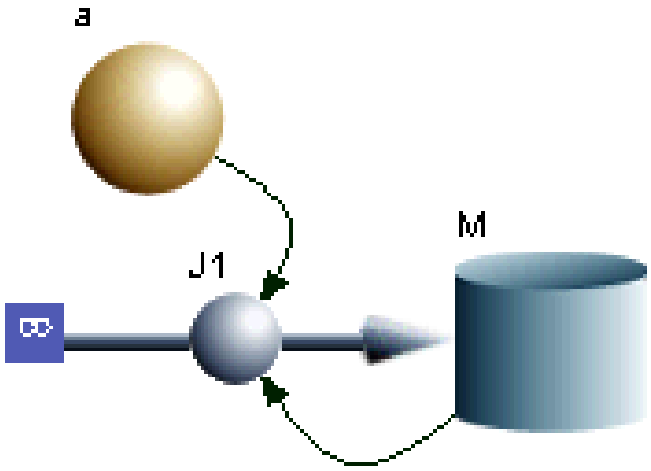
Graphical Representation (Flow Chart)

Model Equations (Mark-up language)

$$\frac{dM}{dt} = R$$



$$M(t) = \int R dt$$




```
{Top model}

  {Reservoirs}
  d/dt (M) = + J1
    INIT M = 0

  {Flows}
  J1 = a*M

  {Functions}
  a = 1
{Globals}
{End Globals}
```

Graphical modelling editors for compartmental system simulation:

Berkeley Madonna
https://berkeley-madonna.myshopify.com/

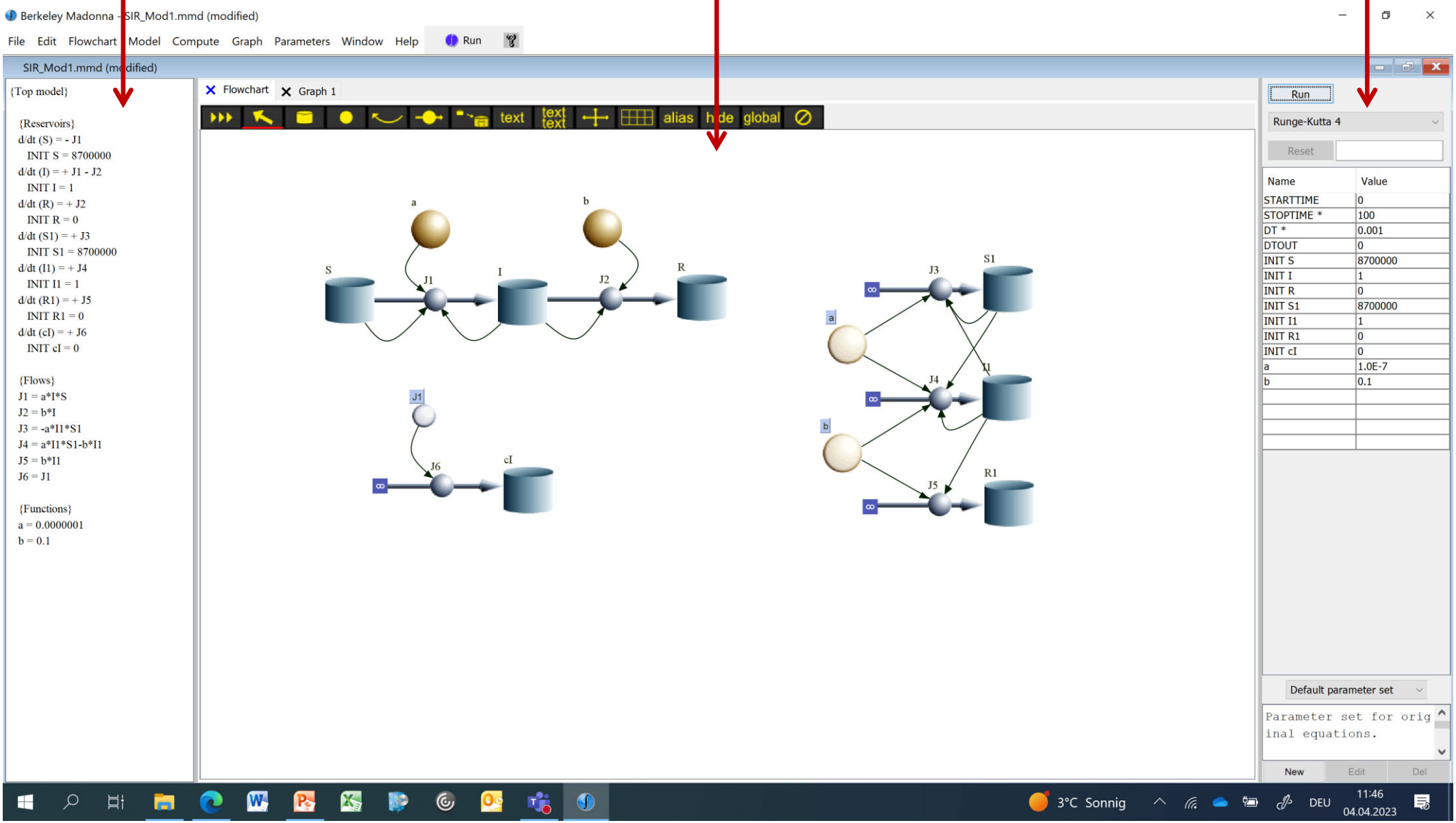
Vensim
https://vensim.com/vensim-personal-learning-edition/

Stella
https://www.iseesystems.com/store/products/stella-architect.aspx

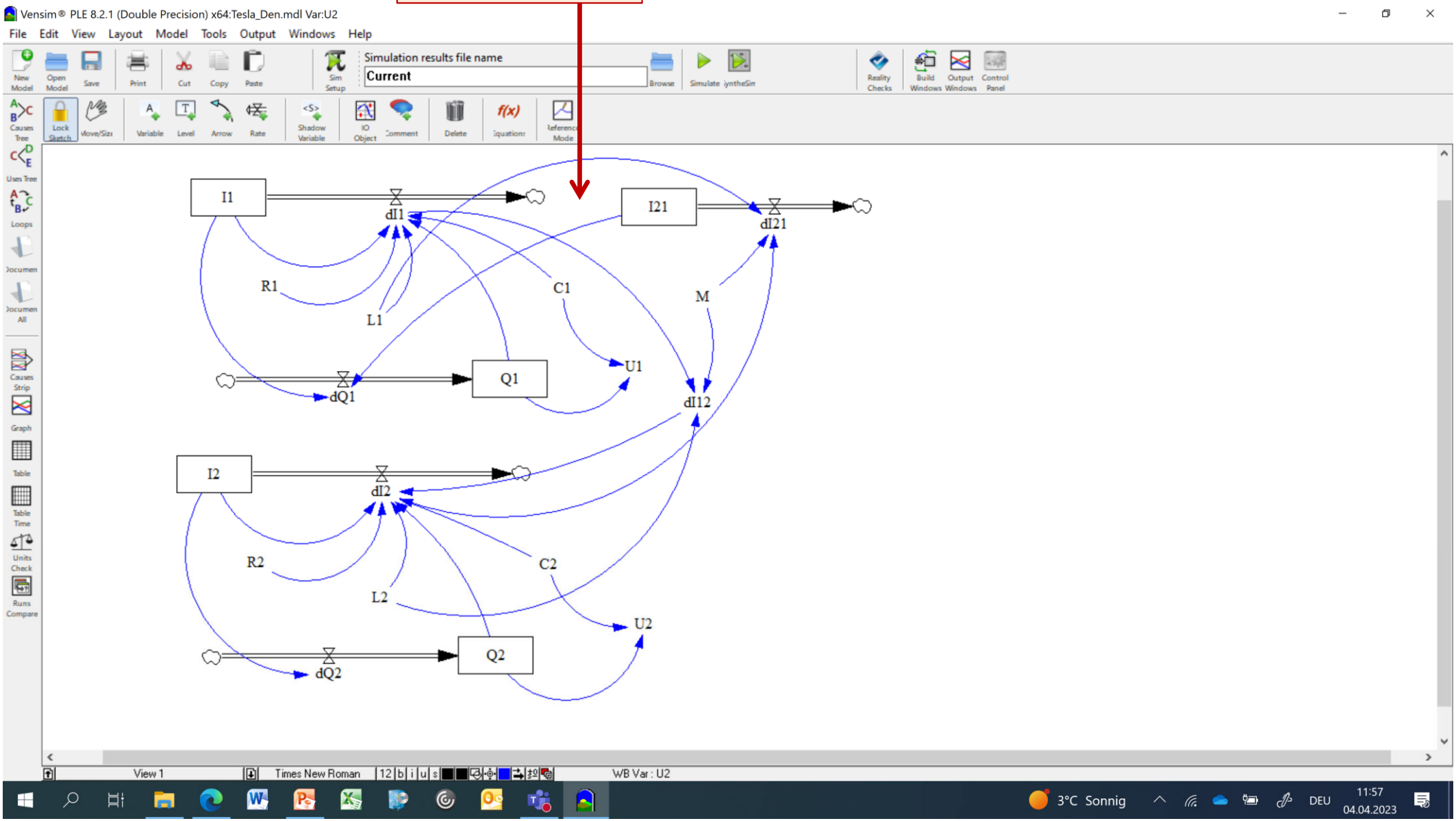
Markup code:
Model equations

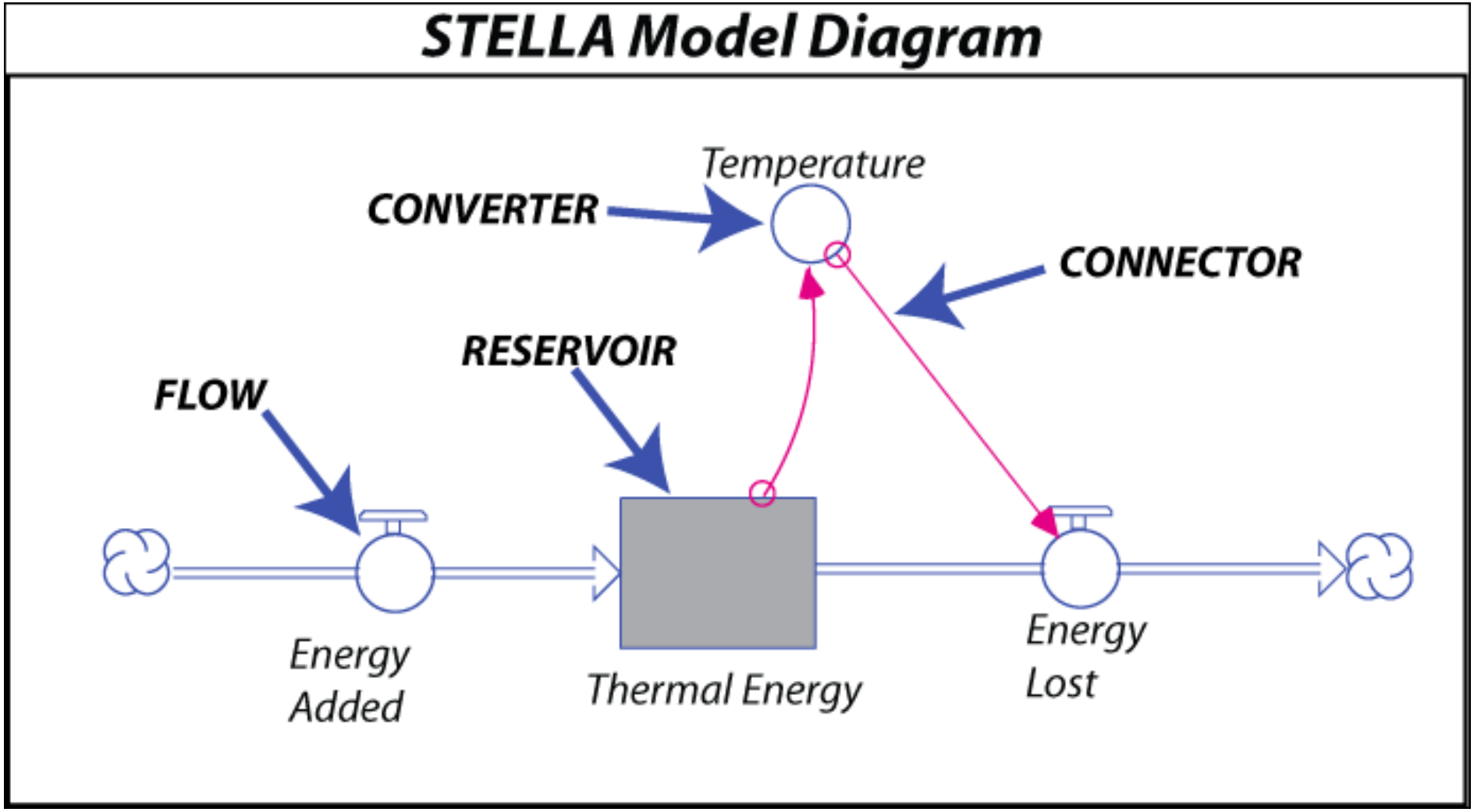
Flow chart:
Graphical model representation

Simualtion “cockpit”:
Numerical integration procedure
Time increment Δt
Parameter values



Flow chart:
Graphical model representation




STELLA Model Diagram
Temperature
CONVERTER
CONNECTOR
RESERVOIR
FLOW
Energy
Added
Thermal Energy
Energy
Lost

moving_car - Simulink

SIMULATION DEBUG MODELING FORMAT APPS

New Open Save Print FILE

Library Browser LIBRARY

Log Signals PREPARE

Stop Time 10.0

Normal

Fast Restart

Step Back Run Step Forward Stop SIMULATE

Data Inspector REVIEW RESULTS

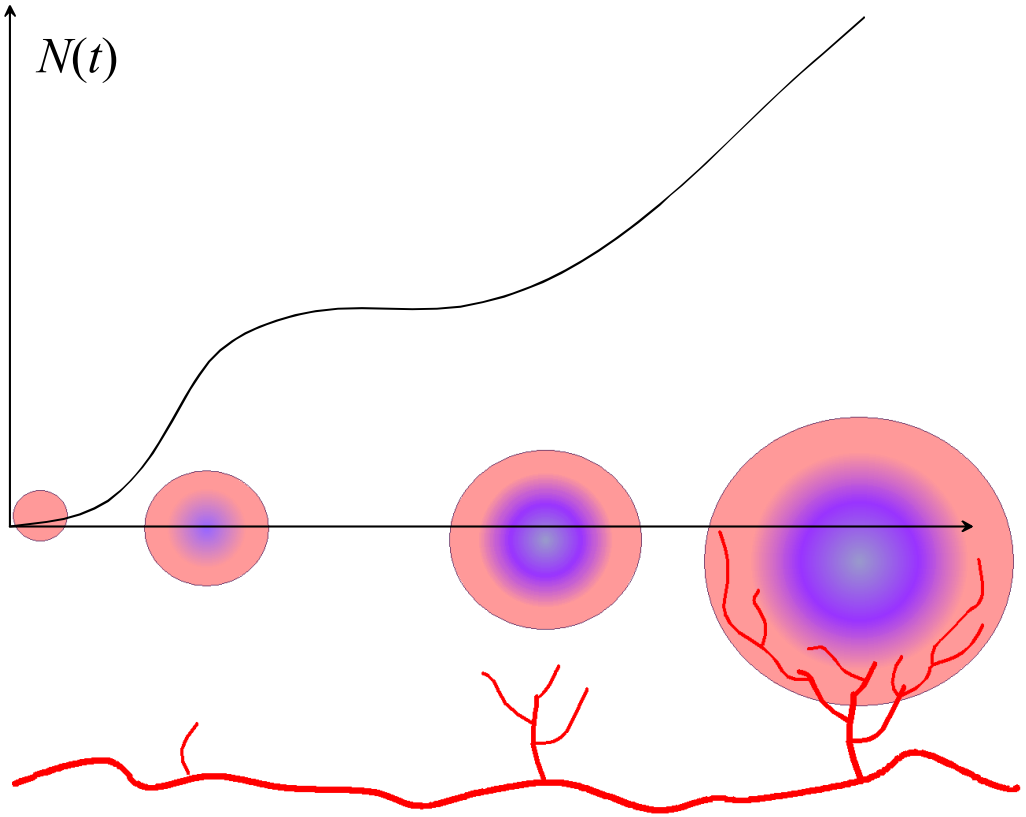
moving_car

moving_car

Copyright 2017-2018 The MathWorks, Inc.

Ready 100% VariableStepAuto

Control of Gene Expression: Oxygenation and VEGF

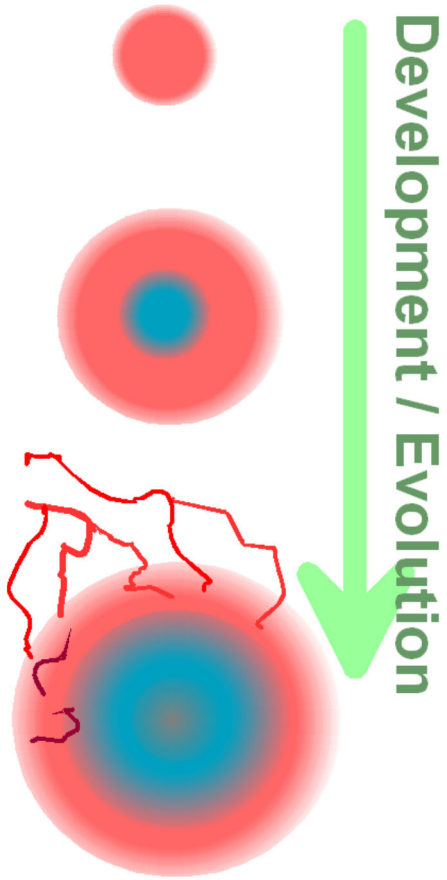


Vascular Endothelial Growth Factor (VEGF) is expressed in hypoxic tissues.

- VEGF expression and secretion lead to activation of vascular endothelial cells
- Result: vessels sprout in direction of the VEGF gradient.

Fig.1. Chromosomes during mitosis

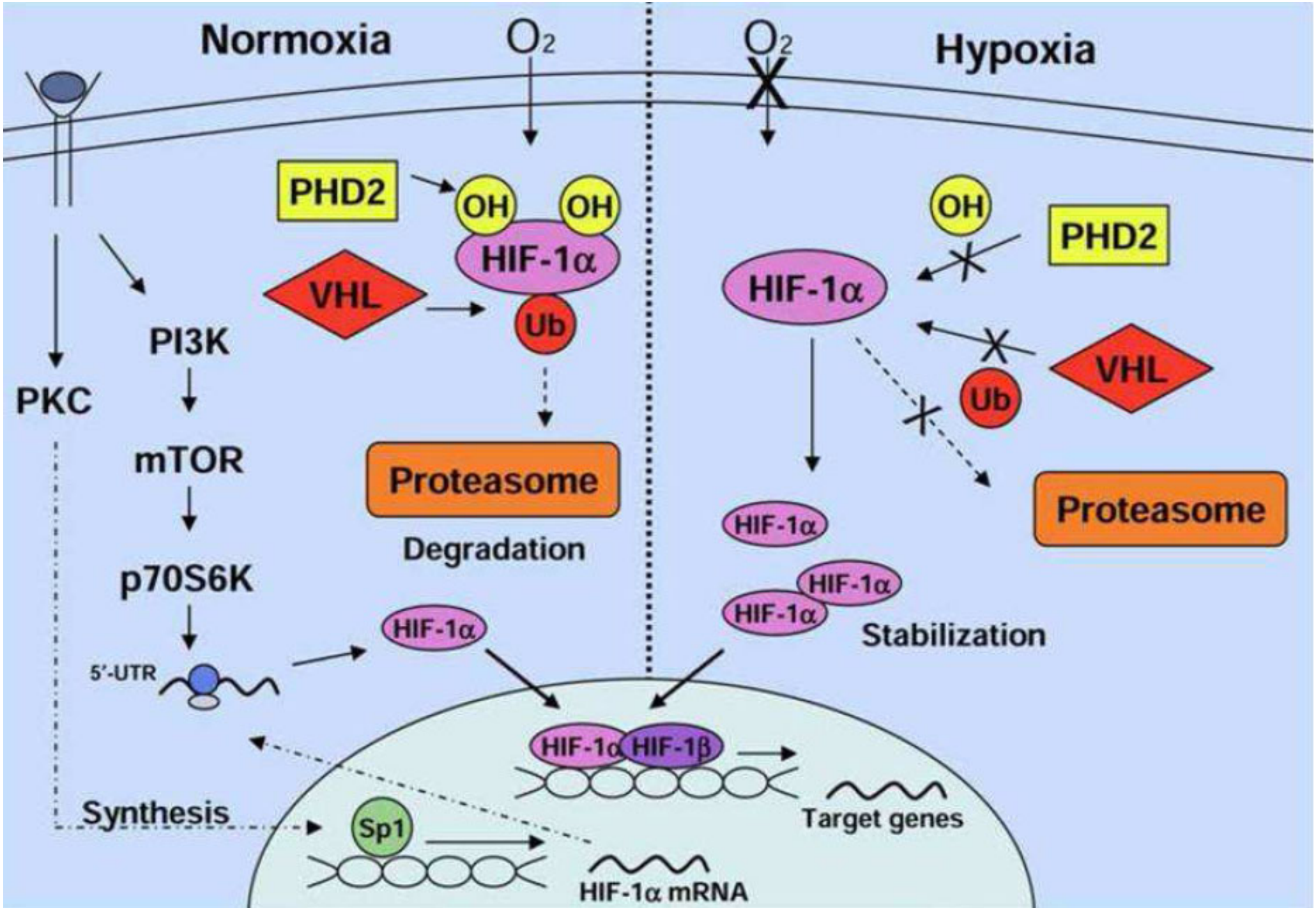
Control of Gene Expression: Oxygenation and VEGF



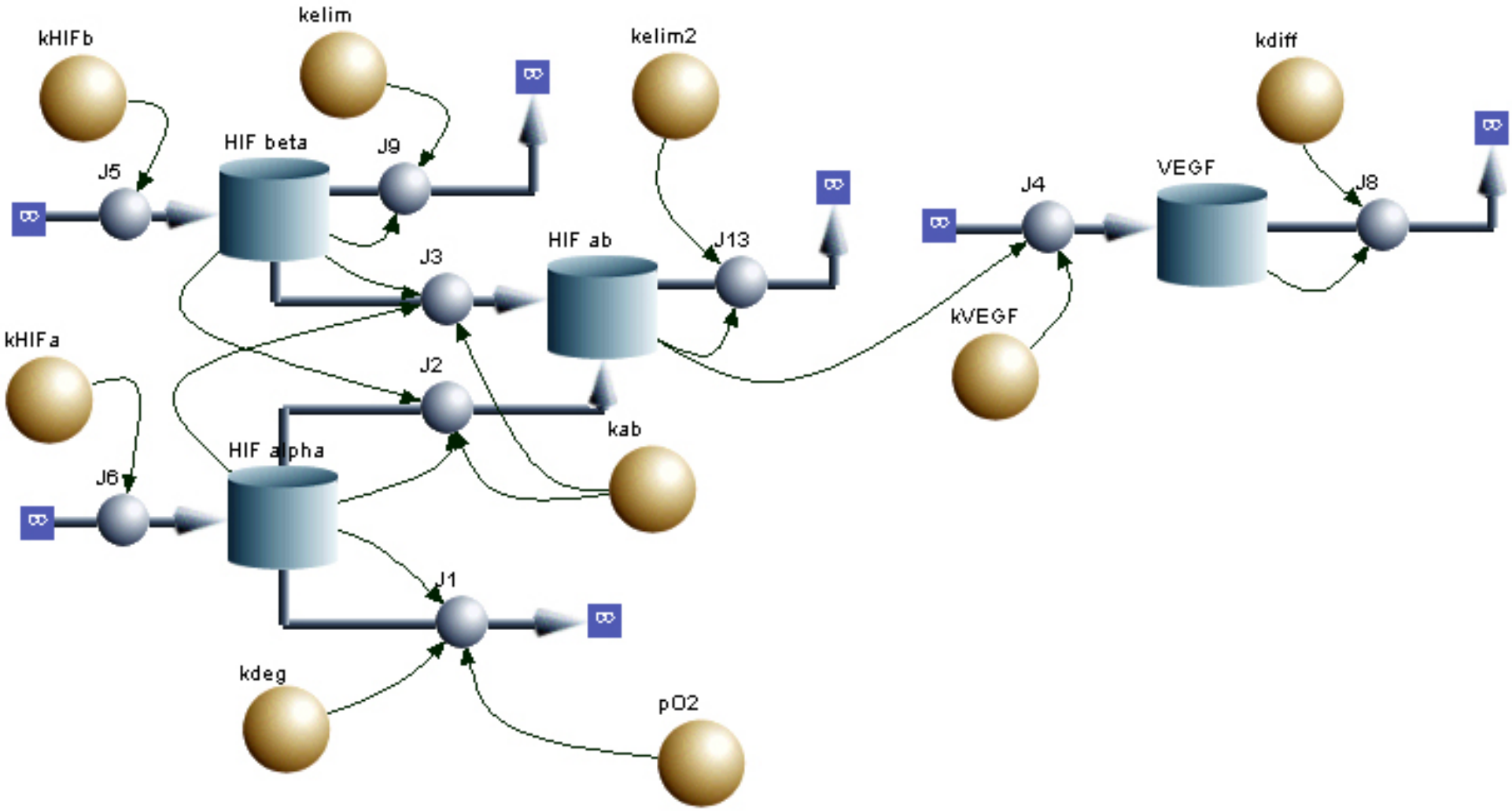
During growth, tumours develop a hypoxic (and subsequently a necrotic) core:

- VEGF release leads to ingrowth of external vessels (angiogenesis)
- Some tumours acquire a vascularized rim
- As a result of the fast growing tumour cell population and lack of tissue organization, tumour vessels often are non-function or leaky.

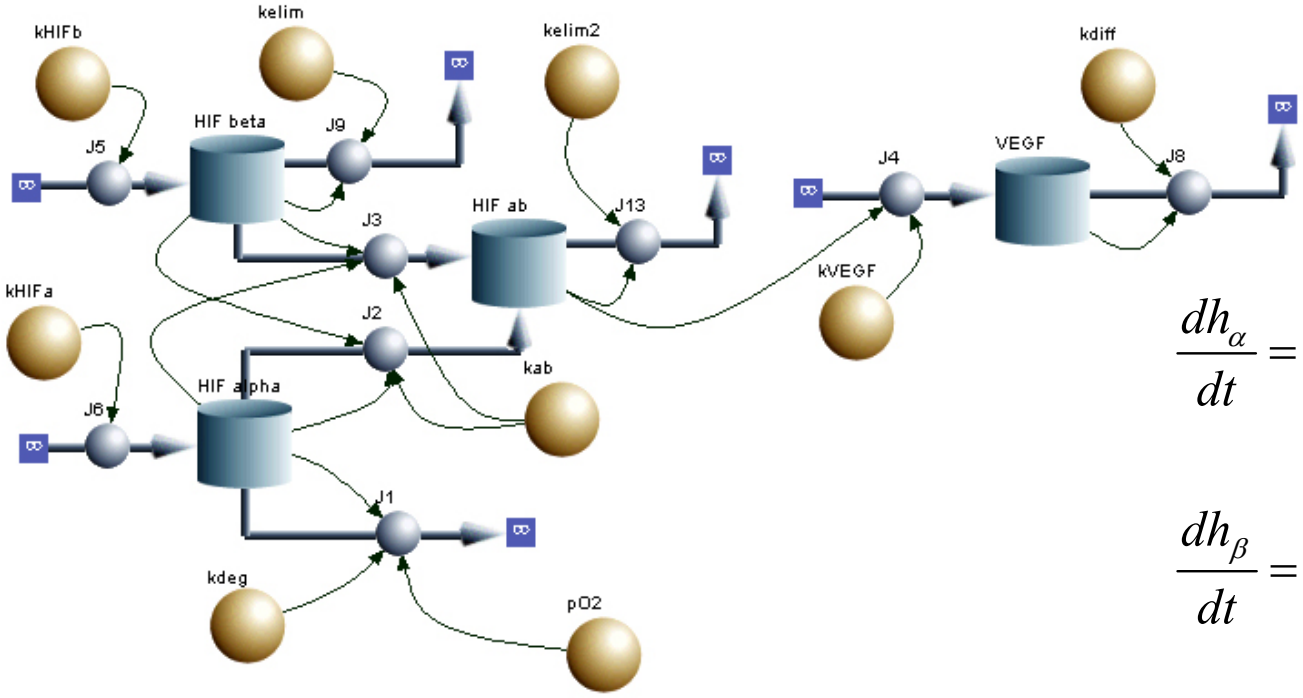
Fig.1. Development of hypoxic and necrotic core during tumour growth


Normoxia
O2
O2
Hypoxia
PHD2
OH
OH
HIF-1α
VHL
Ub
PI3K
PKC
mTOR
p70S6K
Proteasome
Degradation
5'-UTR
HIF-1α
HIF-1α
HIF-1α
HIF-1α
Stabilization
HIF-1α
HIF-1β
Target genes
Synthesis
Sp1
HIF-1α mRNA
OH
PHD2
HIF-1α
VHL
Ub
Proteasome

Control of Gene Expression: VEGF



Control of Gene Expression: VEGF



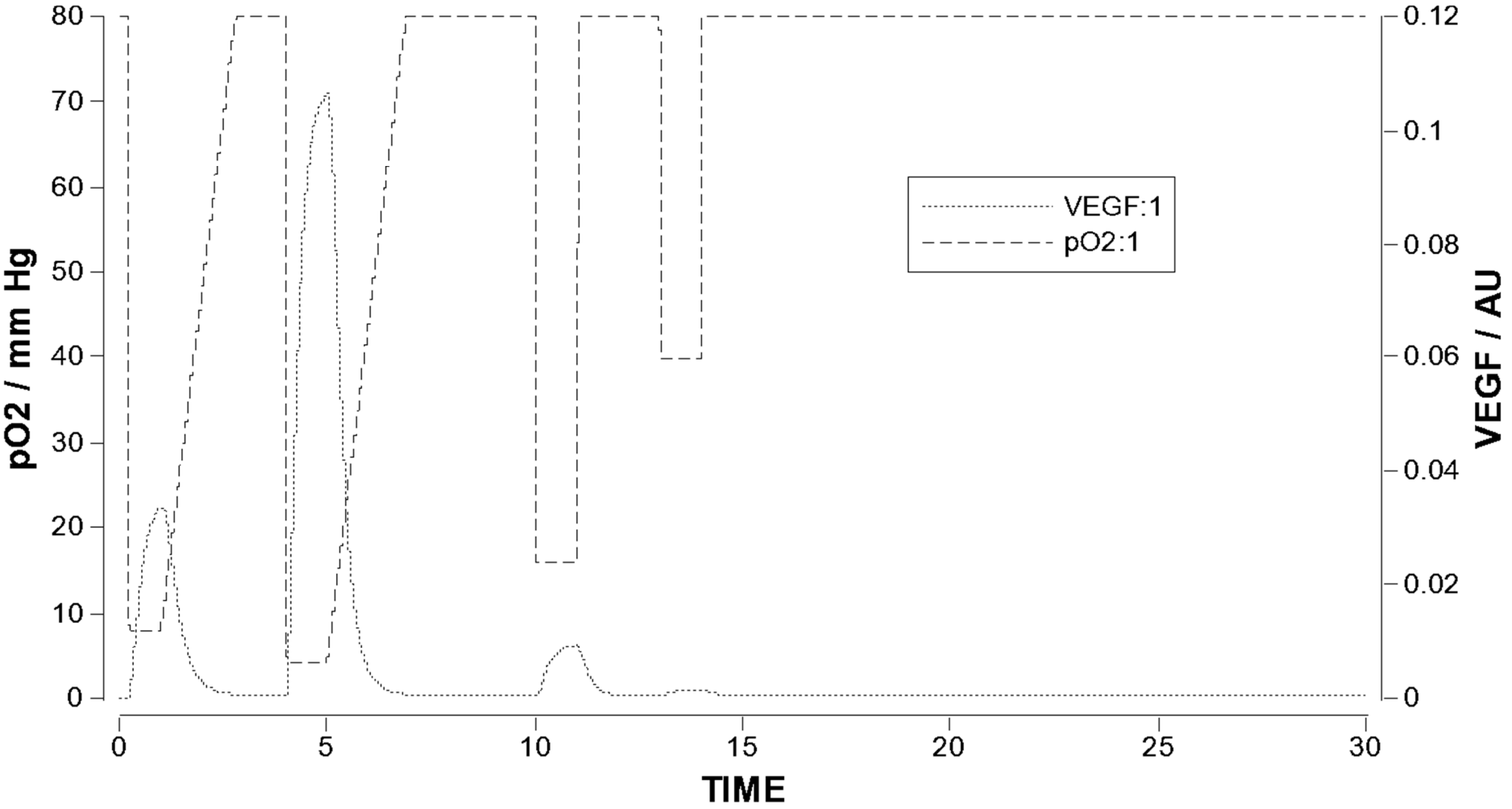
$$\frac{dh_\alpha}{dt} = k_\alpha - k_{O_2} p_{O_2}^2 h_\alpha - k_{\alpha\beta} h_\alpha h_\beta$$

$$\frac{dh_\beta}{dt} = k_\beta - k_{\alpha\beta} h_\alpha h_\beta - k_{\beta e} h_\beta$$

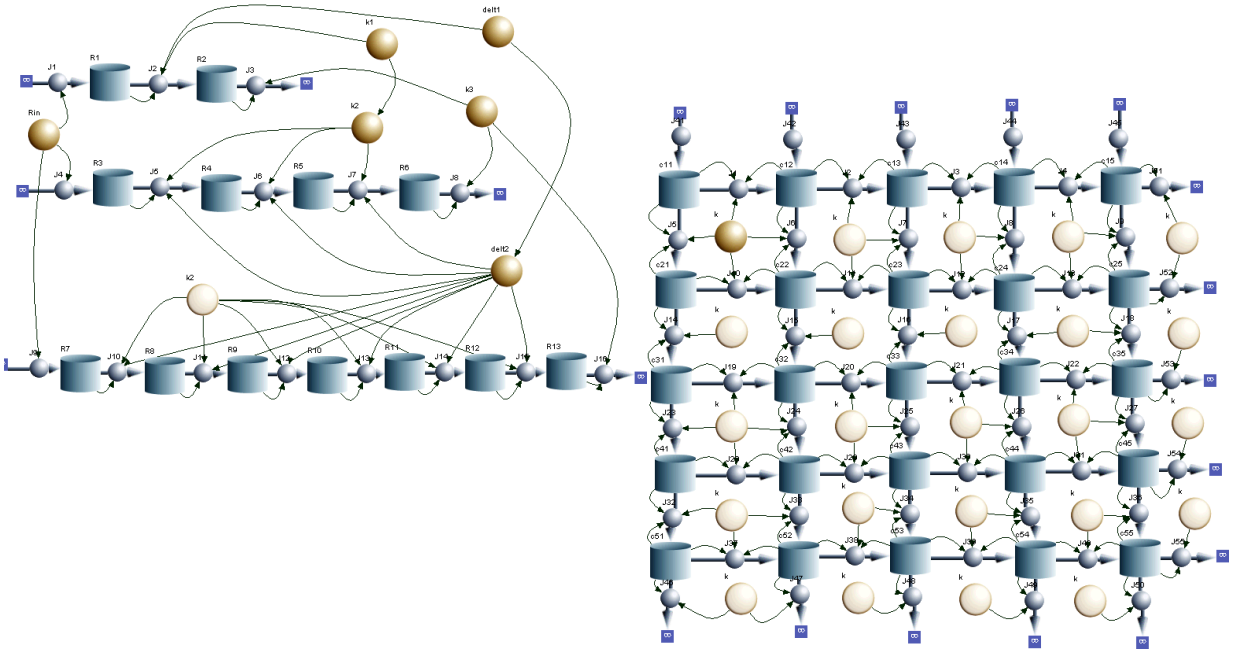
$$\frac{dh_{\alpha\beta}}{dt} = 2k_{\alpha\beta} h_\alpha h_\beta - k_{\alpha\beta e} h_{\alpha\beta}$$

$$\frac{dv}{dt} = k_v h_{\alpha\beta} - k_{ve} v$$

Control of Gene Expression: VEGF

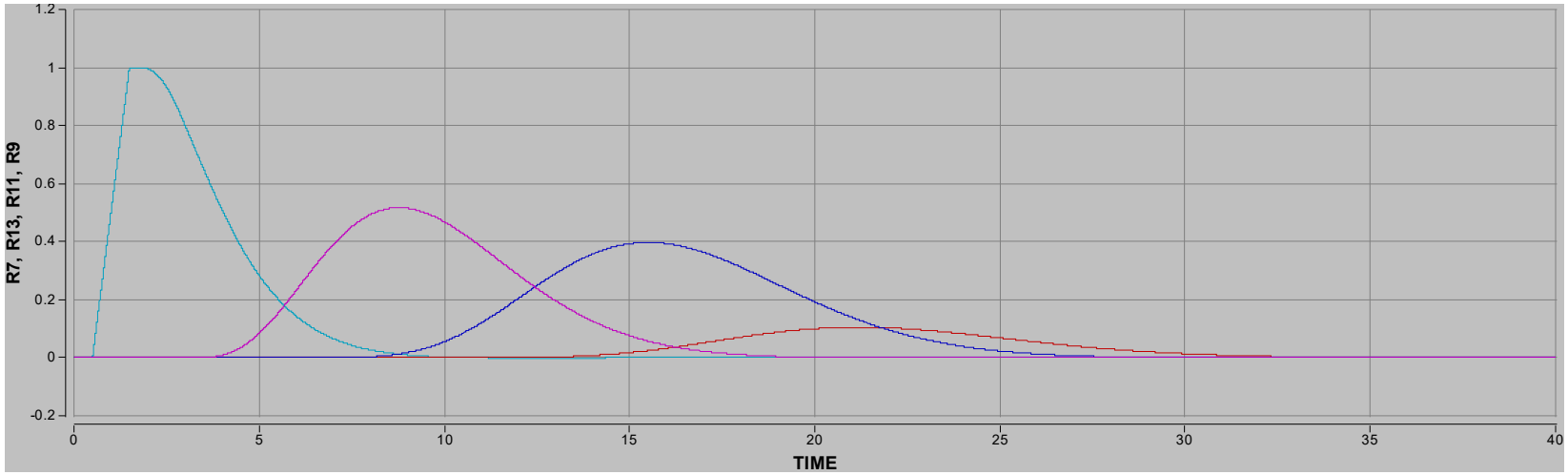


Modelling Diffusion Networks



Reaction chains

- Lead to signal dispersion
- Some aspects of complex biological systems may be covered by reaction-chain – diffusion networks



Compartmental Graphical Model Editors: Pro & Con's

Pro

- Fast programming
- System visualization via flow chart
- Graphically supported modelling process
- Powerful and stable solvers
- Batch run options
- Implemented fitting methods

Desadvantages:

- Dedicated for compartmental models, therefore limited to models that can be described by ODE's or signaling chains
- Graphical representation for large systems or diffusion chains / waveguides difficult → generative algorithms for writing markup code useful