

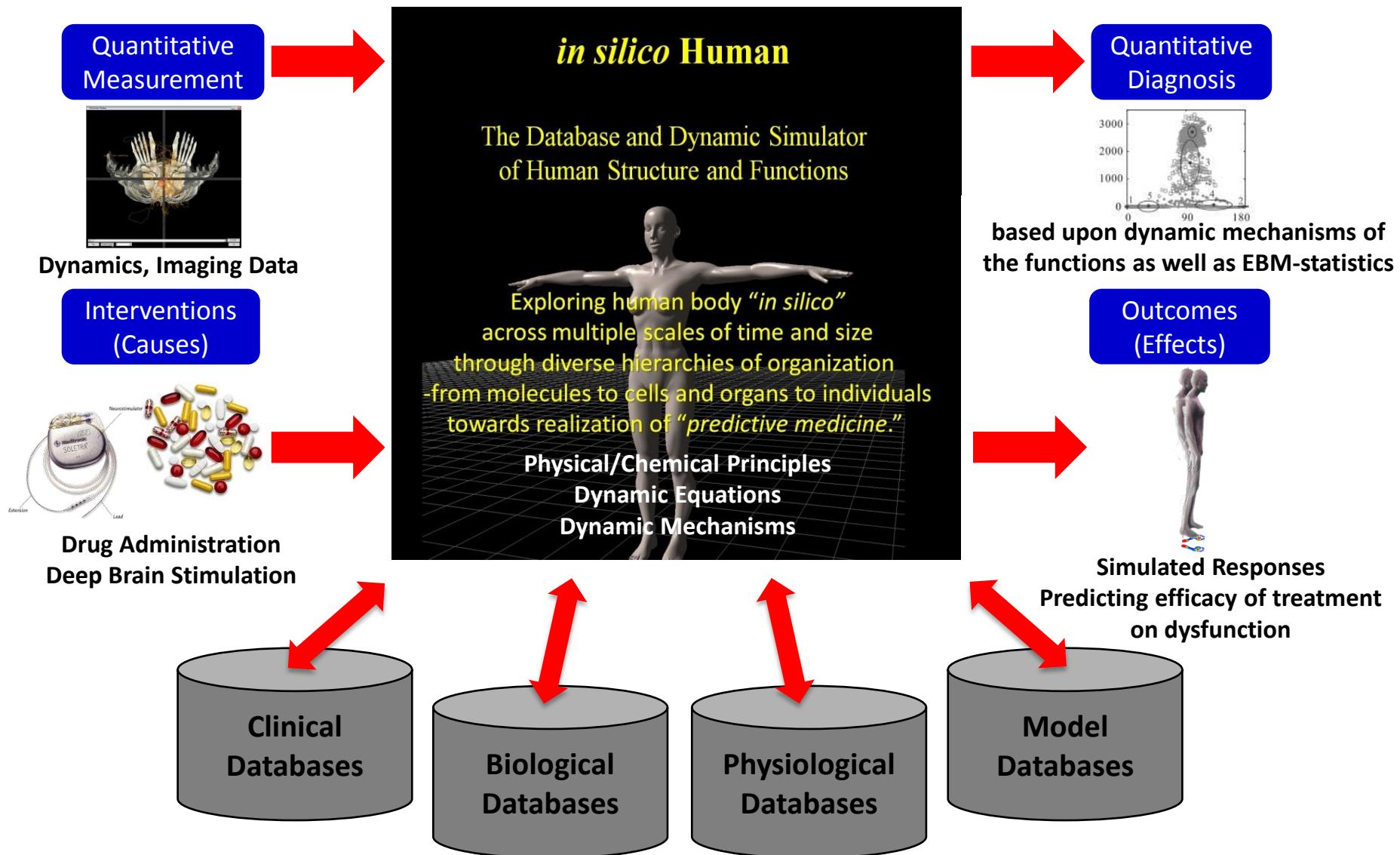
# *Application of FreeFem++ to physiome framework*

HideKi Oka

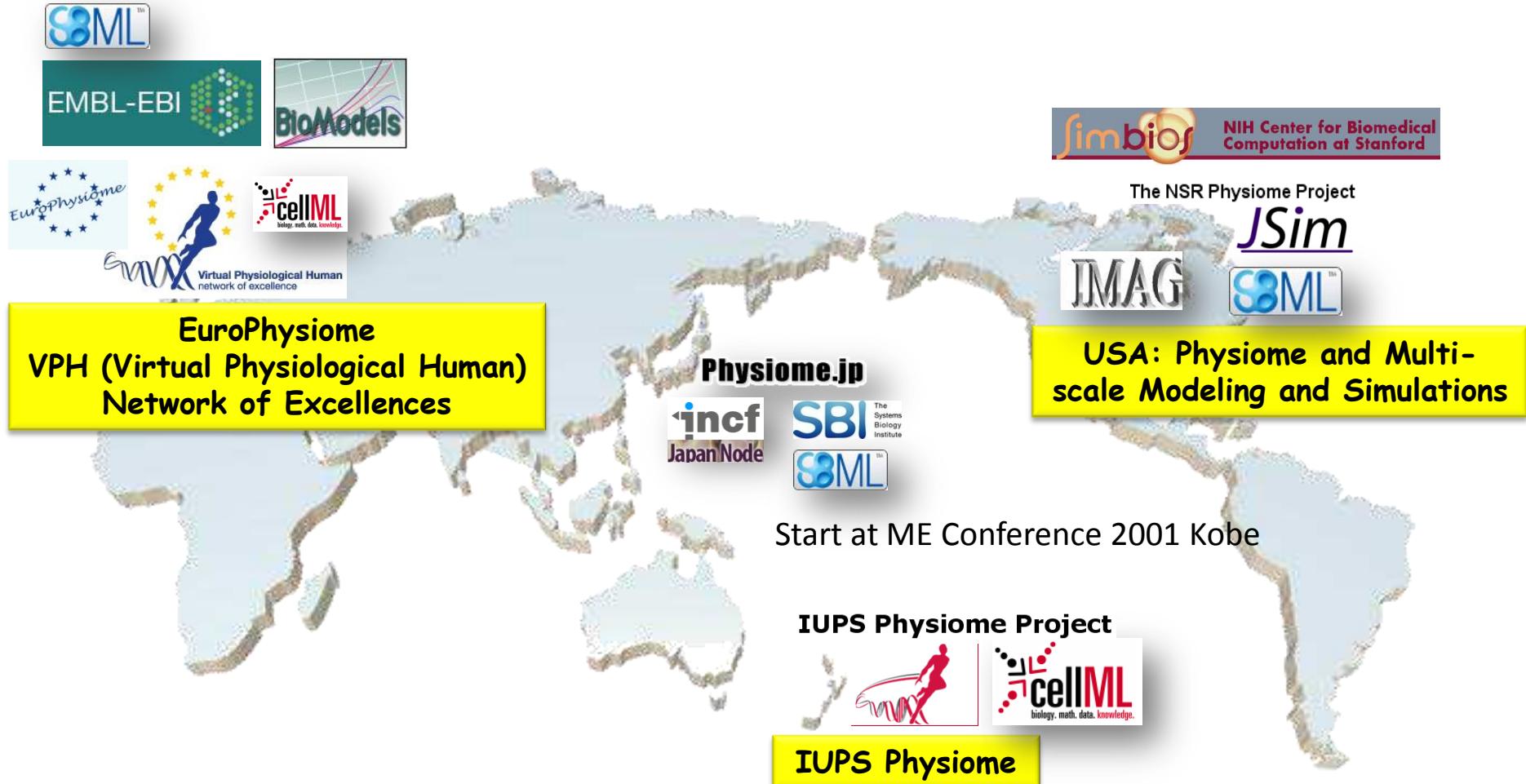
**RIKEN Brain Science Institute, Neuroinformatics Information Japan Center  
Tokai University Department of Medicine (Cardiology)**

E-mail: [czoka@brain.riken.jp](mailto:czoka@brain.riken.jp)  
[czoka@tokai-u.jp](mailto:czoka@tokai-u.jp)

# Toward Integrative & Predictive Biology and Medicine

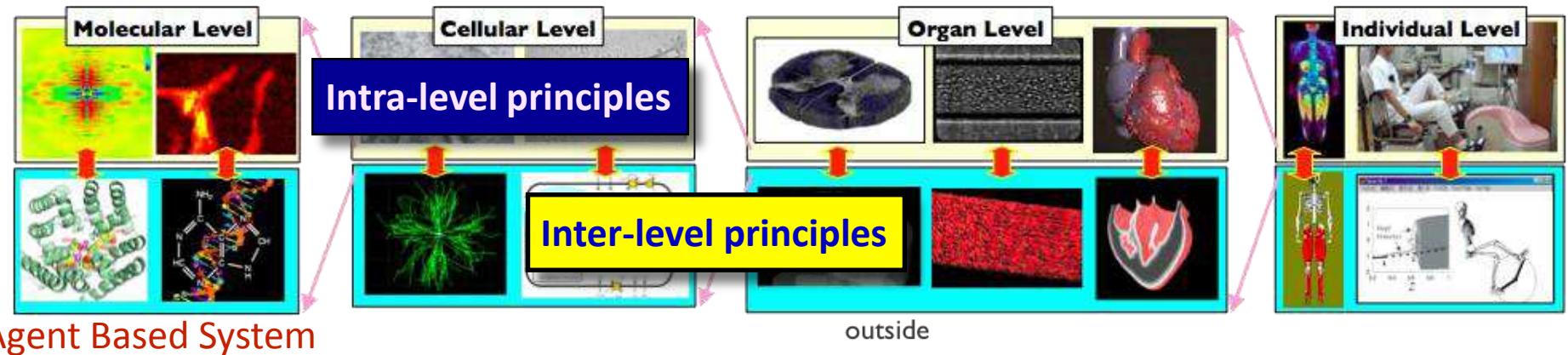


# Worldwide Effort for Physiome and Systems Biology

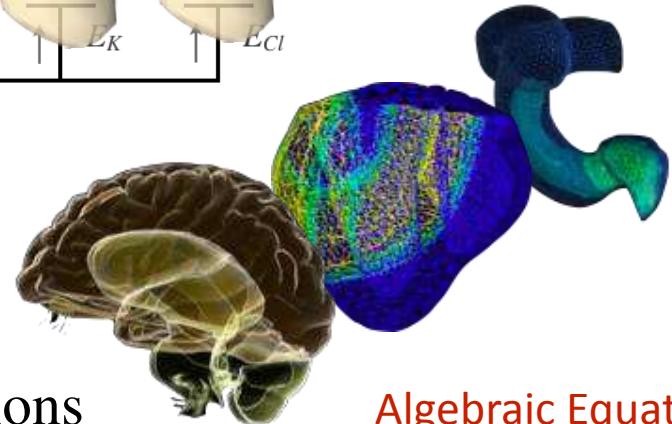
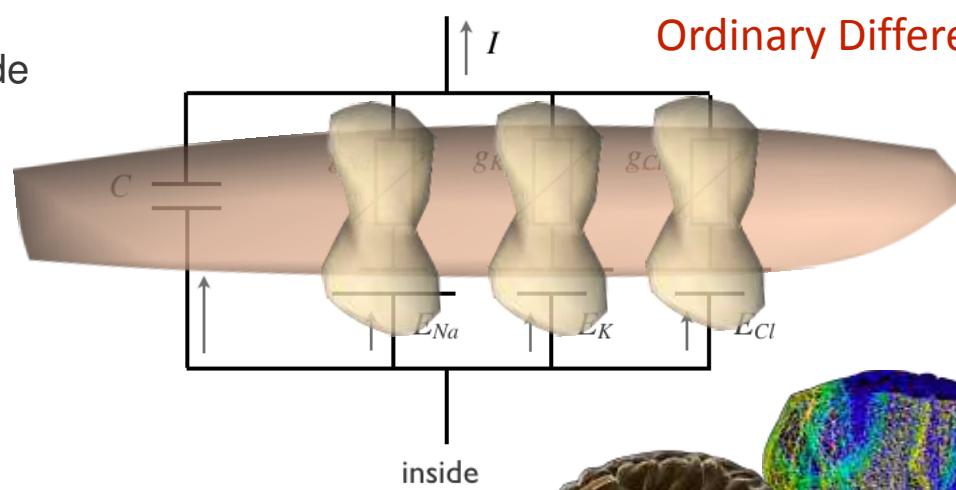
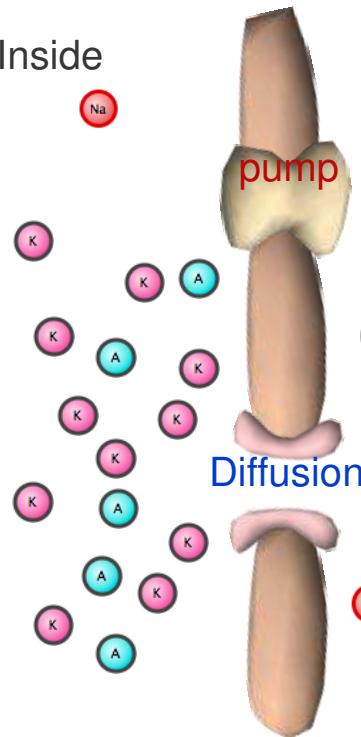


"physio-" (life) and "-ome" (as a whole).

# Several Model Types and Techniques to Simulate

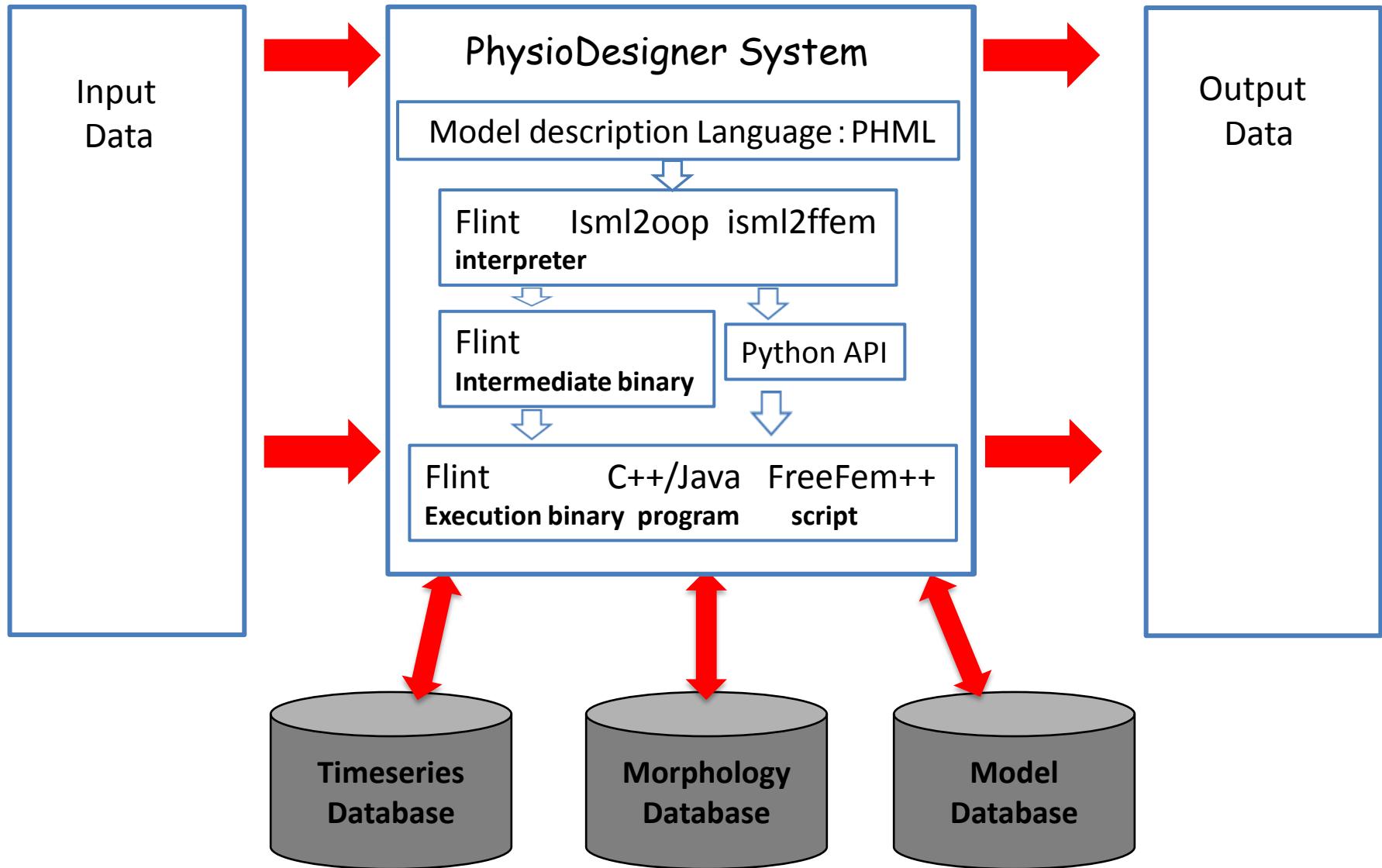


Agent Based System



A variety of model expressions  
A variety of solving methods

# PhysioDesigner System



# PhysioDesigner HP

The screenshot shows the PhysioDesigner HP website. The main header reads "PhysioDesigner A modeling tool for multilevel physiological systems". Below the header, there's a navigation bar with links to "WELCOME", "SCREEN SHOTS", "Download", "Simulation", "DOCUMENTS", "PHML", and "ABOUT US". The "WELCOME" section features a large image of the software's graphical user interface, which includes a hierarchical tree view of models and a detailed configuration dialog box. To the right of the interface, there's a "What's new" section listing release dates and descriptions for versions 1.0beta5, 1.0beta4, 1.0beta3, and 1.0beta2. Below this, there are sections for "Former project" (insilicoPlatform), "Development History", "Model database" (ModelDB in Physiome.jp), and a "Sitemap". At the bottom of the page, there's a note about the history of the project, mentioning its predecessor insilicoIDE (ISIDE) and its transition to PhysioDesigner.

<http://www.physiodesigner.org>

maintained and developed  
by OIST Kitano Unit, Dr. Asai

Originally developed as  
insilicoIDE in GCOE of  
Osaka University

- **PhysioDesigner 1.0 beta6.1**
- **Flint 1.0 beta6.1**
- freely download

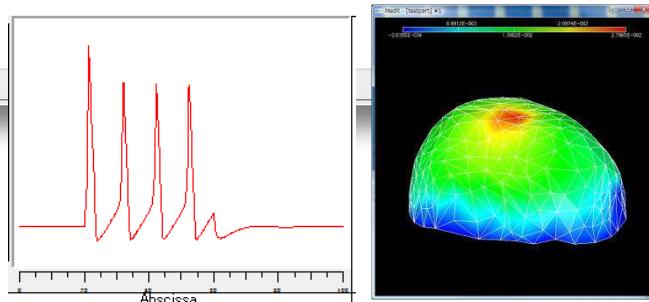
# PhysioDesigner System

## Morphology DB

The screenshot shows the Physiome.jp Morphology DB interface. On the left, there is a search results table with columns for ID, Name, Version, and Description. The table lists numerous entries, mostly starting with 'JON20...' or 'JON20...'. On the right, there is a detailed view of a specific morphology, showing a complex 3D structure with various components and connections.

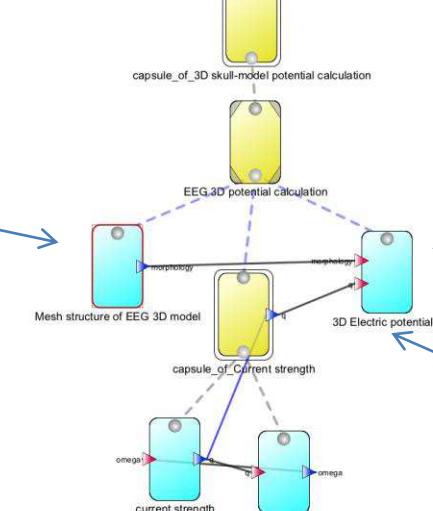
## Model DB

Automatically generate  
Simulation tools  
ISSim  
C++/Java  
FreeFem++



## PhysioDesigner Canvas

Create PHML model on canvas  
constructing simulation procedure  
visually



The screenshot shows the PhysioDesigner system integrated with other tools. At the top right is the 'Image Viewer' showing a brain scan. Below it is the 'Morphology Editor' showing a 3D cube geometry. To the right is the 'FreeFem++' interface, which includes a 3D plot of a mesh and a corresponding 2D plot below it.

## FreeFem++

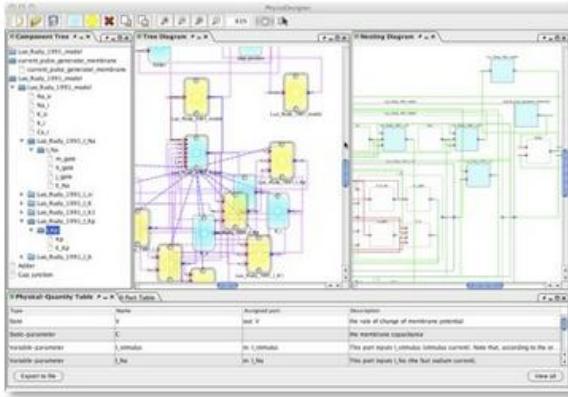
<http://www.freefem.org>

# History of insilicoIDE/PhysioDesigner

- 2000 Module based original model description language was developed.  
MATLAB base in Osaka University
- 2007 Global Center of Excellence start in Osaka University
- 2008 May IDE0.1 First version **ISML0.1**
- 2009 Oct IDE1.0 Introduce XML DB and APIs
- 2010 Feb IDE1.1 **Use FreeFem++ for PDE simulation**
- 2010 Sep IDE1.3 Add Morphology editor
- 2011 Apr IDE1.4 Strengthen Morphology editting function
- 2011 Sep IDE1.4.2 Combine Morphology Geometry and Segment functions
- 2011 Nov IDE1.4.4 Image/segmentation function, Morphology editor/meshing function
- 2012 April Rename as PhysioDesigner, move to development site to System Biology Institute/OIST
- PhysioDesigner1.6 beta
- PDE: partial differential equation



## What is PhysioDesigner?



### An open platform for multilevel modeling

PhysioDesigner is an open platform for supporting multilevel modeling of physiological systems in the field of integrated life-science and systems biology, including physiology, neuroscience and so on. Users can combine and build mathematical models of biological and physiological functions on PhysioDesigner. Users can also integrate morphometric data on a model, which is used, for example, to define a domain in which partial differential equations are solved.

The models developed by PhysioDesigner is stored in PHML(Physiological Hierarchy Markup Language) format which is an XML-based specification, to describe wide variety of models of biological and physiological functions with hierarchical structure. PHML fully inherits the specification from [insilicoML](#) (ISML). PhysioDesigner can act as an editor and browser of the models written in PHML and ISML. Besides that, PhysioDesigner can import models written in [CellML](#) format and export into it (some of model expressions in ISML cannot be exported into CellML).

One of distinguished feature of PhysioDesigner, it provides a function for users to create SBML-PHML hybrid models, which is a novel way to create multi-level physiological systems. Besides, it is possible to integrate morphological data in a model.

### What's new

2012.04.24

PhysioDesigner 1.0beta has been released.

2012.01.20

PhysioDesigner 1.0alpha has been released.

### Former project

[InsilicoPlatform](#)

[Development History](#)

### Model database

[ModelDB in Physiome.jp](#)

[Sitemap](#)

Supported by:



Collaborating with:



company  
logo

Access to <http://physiodesigner.org>

# Multilevel Biophysical Model Description Language PHML(Physiological Hierarchy Markup Language)

The screenshot shows a Windows Internet Explorer window displaying an XML document. The URL in the address bar is [http://www.physiome.jp/modeldb/get.php?rmid=478&version=1.0&rname=Chicago\\_2003\\_pancreatic\\_beta-cell](http://www.physiome.jp/modeldb/get.php?rmid=478&version=1.0&rname=Chicago_2003_pancreatic_beta-cell). The XML code is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <is:insilico-model xmlns:is="http://www.physiome.jp/ns/insilicoml" xmlns:m="http://www.w3.org/1998/Math/MathML"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" version="1.0"
  xsi:schemalocation="http://www.physiome.jp/ns/insilicoml-1.0 insilicoml.xsd">
- <is:header>
  - <is:model-name>Chicago_2003_pancreatic_beta-cell</is:model-name>
  - <is:meta-information>
    - <is:article-set>
      - <is:article type="model-proposal" article-id="1">
        - <is:journal>
          <is:journal-title>American journal of physiology. Endocrinology and metabolism</is:journal-title>
          <is:publisher-name />
          <is:volume>285</is:volume>
          <is:issue>1</is:issue>
        - <is:publication-date>
          <is:year>2003</is:year>
          <is:month>Jul</is:month>
        - <is:journal>
          <is:pubmed-id>12644446</is:pubmed-id>
          <is:article-title>Modeling of Ca2+ flux in pancreatic beta-cells: role of the plasma membrane and intracellular stores.</is:article-title>
          <is:vernacular-article-title lang="any" />
        - <is:pages>
          <is:first-page />
          <is:last-page />
        - <is:pages>
        - <is:target-model-set>
          <is:target-model db-id="478" />
        - <is:target-model-set>
        - <is:author-set>
          - <is:author>
            <is:first-name>Leonid E</is:first-name>
            <is:last-name>Fridlyand</is:last-name>
          - <is:author>
        - <is:author-set>
      - <is:article>
    - <is:meta-information>
  - <is:header>
</is:insilico-model>
```

XML file、specification is in PhysioDesigner HP、PHML models are stored in ModelDB (<http://www.physiome.jp>)

# PHML

- XML-based model descriptive specification.

PHML inherits ISML specification

PHML has partial compatibility with [CellML](#), and is developing cooperative frameworks with [SBML](#).

- PHML has some complementary features to describe models.

## Requirements

- ★ It can express hierarchy of biophysical functions and structures.
- ★ It can describe dynamics of biophysical functions
  - Ordinary / Partial differential equations, agent based system, boundary conditions, etc
- ★ It can describe structures related to biophysical functions
  - Surface information, Volume information
- ★ It can describe meta-information of the model and physiology.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<is:insilico-model xmlns:is="http://www.physiome.jp/ns/insilicoml" xmlns:m="http://www.w3.org/1998/Math/MathML" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" version="1.0" xsi:schemalocation="http://www.physiome.jp/ns/insilicoml-1.0 insilicoml.xsd">
  <is:module-set>
    <is:module module-id="00adeeb8-51c7-4828-979c-92888c58d897">
      <is:property>
        <is:name>m_u</is:name>
        <is:type>functional-unit</is:type>
        <is:capsulation state="true">
          <is:capsulated-by module-id="c2bf70d0-13af-426e-8e5c-fa8c78e91185"/>
        </is:capsulation>
        <is:template state="false"/>
        <is:time-unit name="second" prefix="m"/>
        <is:keyword-set/>
        <is:track>
          <is:involved db-id="this" date="2009-05-20T06:44:00+09:00"/>
        </is:track>
        <is:description/>
      </is:property>
      <is:port-set>
        <is:port port-id="1" direction="in">
          <is:name>v</is:name>
          <is:description/>
        </is:port>
        <is:port port-id="2" direction="out">
          <is:name>u</is:name>
          <is:reference physical-quantity-id="1"/>
        </is:port>
      </is:port-set>
      <is:physical-quantity>
        <is:physical-quantity physical-quantity-id="1" type="variable-parameter">
          <is:precision>double</is:precision>
          <is:unit unit-id="0"/>
          <is:dimension type="scalar"/>
        </is:physical-quantity>
        <is:argument-set>
          <is:initial-value/>
        </is:argument-set>
      </is:physical-quantity>
    </is:module>
  </is:module-set>
</is:insilico-model>
```

# How PHML describes model

## Header(model information)

- Description of ML contents
- Reference paper
- Time data

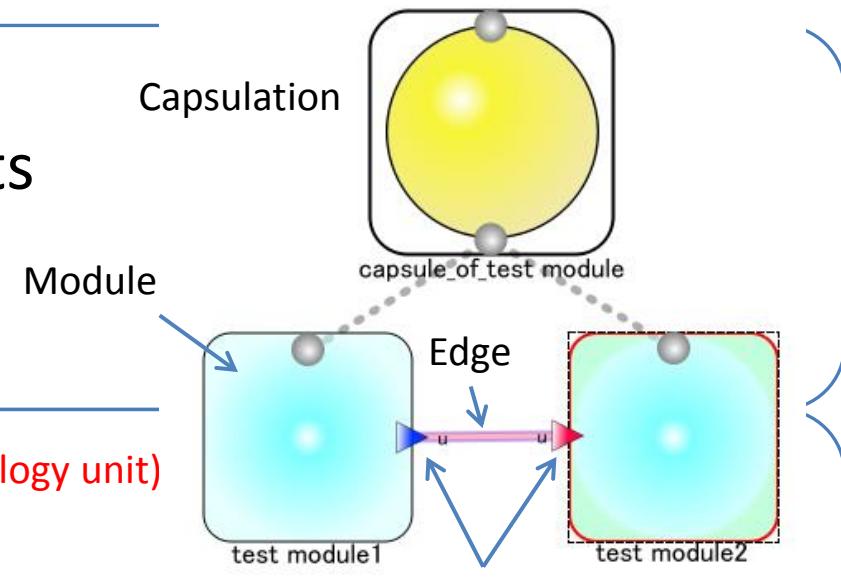
## Module set(phenomena and morphology unit)

- Port description
- Physical quantity description

Equation definition、Constant value、Function definition

## Edge(represent relationship among modules)

- Module connectivity information



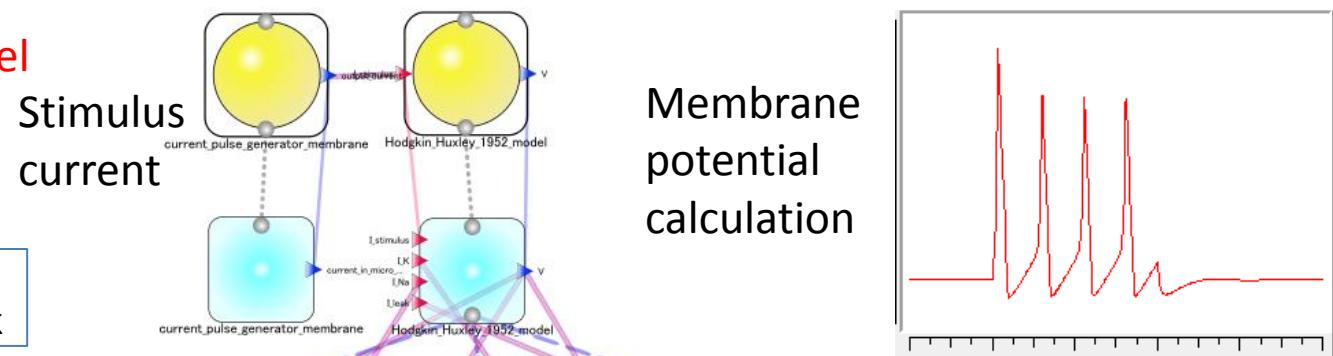
# PHML example: Hodgkin-Huxley Eq.

Cell membrane  
electrophysiological model

$$\begin{array}{l} I_{\text{stimulus}} \\ \downarrow \\ I_m = I_{\text{Na}} + I_K + I_{\text{leak}} \end{array}$$

$$V_m$$

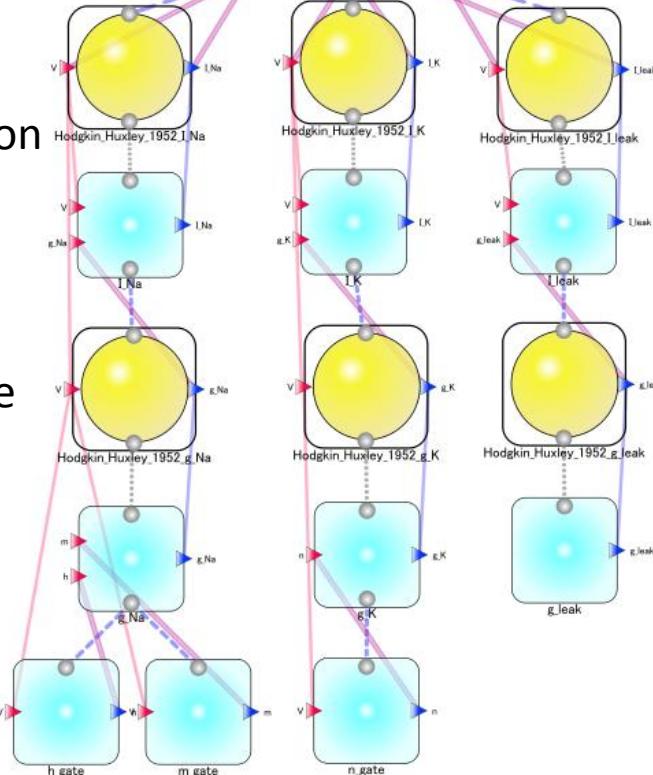
Represent  
Model hierarchy



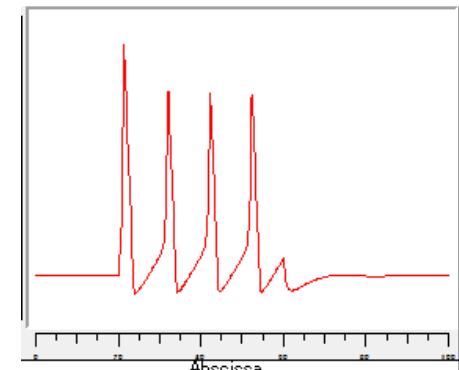
Current  
calculation

Conductance

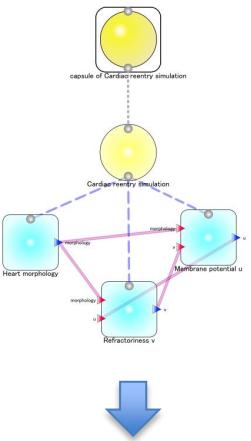
Gate variable



Membrane  
potential  
calculation



# PHML tags



Python API  
(set)

```
set_type(self, edge_id, value)
```



```
<is:insilico-model version="1.0">
  <is:header>
    ...
  </is:header>
  <is:unit-set>
    ...
  </is:unit-set>
  <is:module-set>
    ...
  </is:module-set>
  <is:controller-set>
    ...
  </is:controller-set>
  <is:template-set>
    ...
  </is:template-set>
  <is:instance-set>
    ...
  </is:instance-set>
  <is:timeseries-set>
    ...
  </is:timeseries-set>
  <is:edge-set>
    ...
  </is:edge-set>
</is:insilico-model>
```

## insilico-model Specifications

- [header](#)
- [unit-set](#)
- [edge-set](#)
- **[module-set](#)**
- [controller-set](#)
- [template-set](#)
- [instance-set](#)
- [timeseries-set](#)

## module-set Specifications

- [property](#)
- [port-set](#)
- **[physical-quantity-set](#)**
- [event-set](#)
- [morphology](#)
- [import](#)

## physical-quantity-set Specifications

- [name](#)
- [description](#)
- [value-type-set](#)
- [dimension](#)
- [max-delay](#)
- [initial-value](#)
- [argument-set](#)
- [domain](#)
- [implementation](#)
- [problem-condition-set](#)
- [Miscellaneous Notes](#)



Python API  
(get)

```
get_type(self, edge_id, value)
```



## Simulation tools

ISSim  
C++/Java  
FreeFem++

# Model DB contents

## (1) Cell membrane Electrophysiology model

(Cells are Cardiomyocyte, Nerve Cell, Pancreas  $\beta$ -cell etc.)

Nerve Cell :

various parts of the brain(such as the thalamus and hippocampus)

micro

Nerve cells in the spinal cord

Optic nerve cells in the retina

Model of research papers  
Sharing and reproduction

## (2) Cell network model that combines the plurality of cells

Neural network model using a detailed electrophysiological model of cell

Electrophysiological model of myocardial tissue

## (3) Organ continuum model

Partial differential equation + morphology

## (4) Motion control model of human(0.3 series)



macro

# Model DB

## Physiome.jp

The screenshot shows the Physiome.jp Model DB View Models page. At the top, there are links for Home, insilicoML, insilicoIDE, downloads, tutorial, FAQ, and insilicoDB. The insilicoDB section includes Morphology DB, Time-series DB, terms of use, analysis, MedMod, Publications, Forum, Wiki, Contact, and Links. A 'supported by' section lists various organizations like JNC, MEI, and FUJITSU. A 'taken over by' section lists PhysioDesigner and Dynamic Brain. A 'collaborating with:' section lists SB, Dynamic Brain, and JNC. The main content area is titled 'Model DB: View Models' and features a 'Model Search' form with fields for ISML Ver., Word:, Field: (Model Description), Order by: (DBID), and a search button. Below the search form is a table listing 373 models. The columns in the table are: ID, Type, DBID, Model Name, ISML Ver., ISODE 0.3, ISODE 1.0, PCODE 0.6, and Last Update. The table contains numerous entries, each with a small icon and a link to the model details.

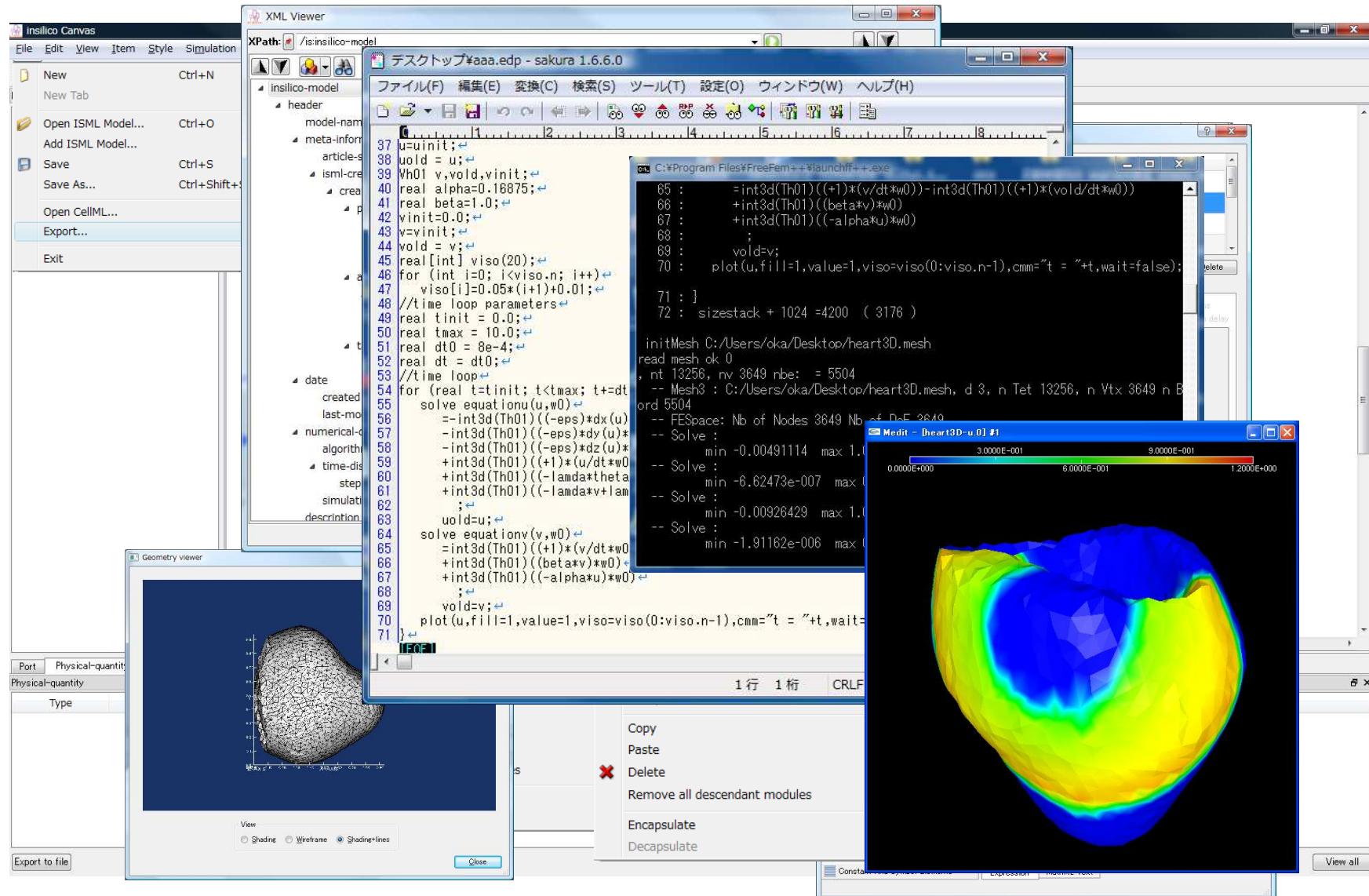
<http://www.physiome.jp/modeldb/index1.php>

## INCF Japan Node/Dynamic Brain Platform

The screenshot shows the INCF Japan Node/Dynamic Brain Platform Xoonips itemselect.php page. The title bar says 'Xoonips - DynamicBrain Platform'. The left sidebar has Japanese language options and a navigation menu with links to 'Dynamic Brain' (説明), '静的記憶' (静的記憶), '消費プロジェクト' (消費プロジェクト), '学術会議' (学術会議), '発表モデル' (発表モデル), and '更に見る...' (More). Below the sidebar is a 'ログイン' (Login) form with fields for 'ユーザー名' (User Name) and 'パスワード' (Password). The main content area is titled 'アイテム一覧' (Item List) and shows a list of 21-40 of 84 items. Each item has a thumbnail, a title, and a link. The titles include: Discrete-time Wilson-Cowan Neuron Model, Double pendulum, FHN model, Flip-flop oscillations network model with a Milnor attractor (single unit), Flip-flop oscillations network model with a Milnor attractor(two units), Hayashi Izhizuka 1992 pacemaker neuron model, Hodgkin Huxley 1952 g\_K, Hodgkin Huxley 1952 g\_leak, Hodgkin Huxley 1952 g\_Na, Hodgkin Huxley 1952 I\_K, Hodgkin Huxley 1952 I\_leak, Hodgkin Huxley 1952 I\_Na, Hodgkin Huxley 1952 model, Hodgkin Huxley 1952 model, Hodgkin Huxley 1952 model for AP Clamp, Humanoid HOAP 1 gait, Inrig Neuron, Integrate-and-Fire Neuron Model, and Inter Neuron.

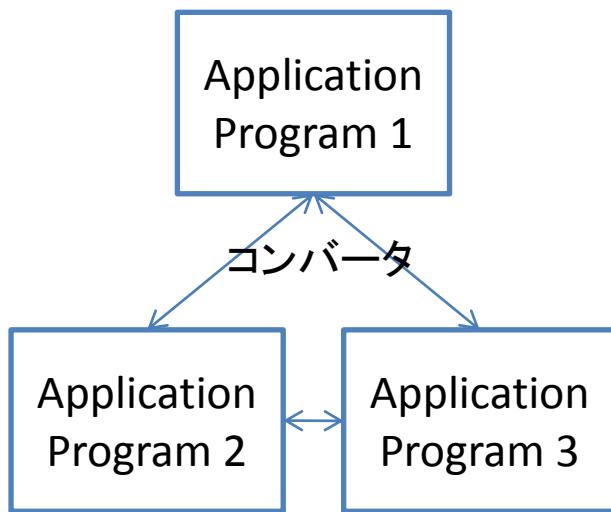
<http://dynamicbrain.neuroinf.jp/modules/xoonips/itemselect.php>

# Build PHML model and execute it



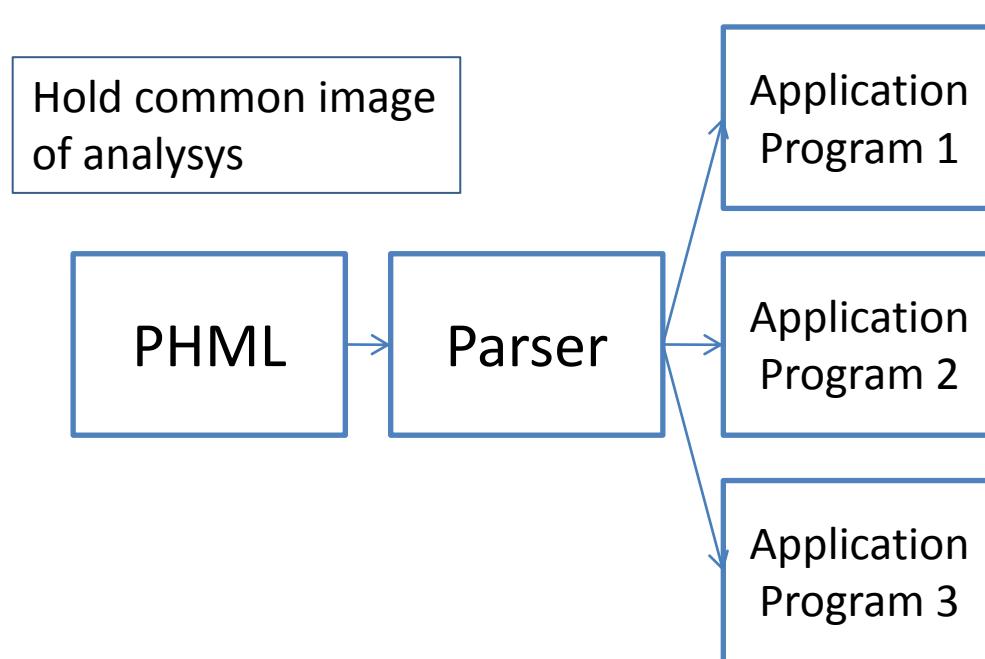
# Which is better ?

## Case 1



Input data format is different per each application

## Case2



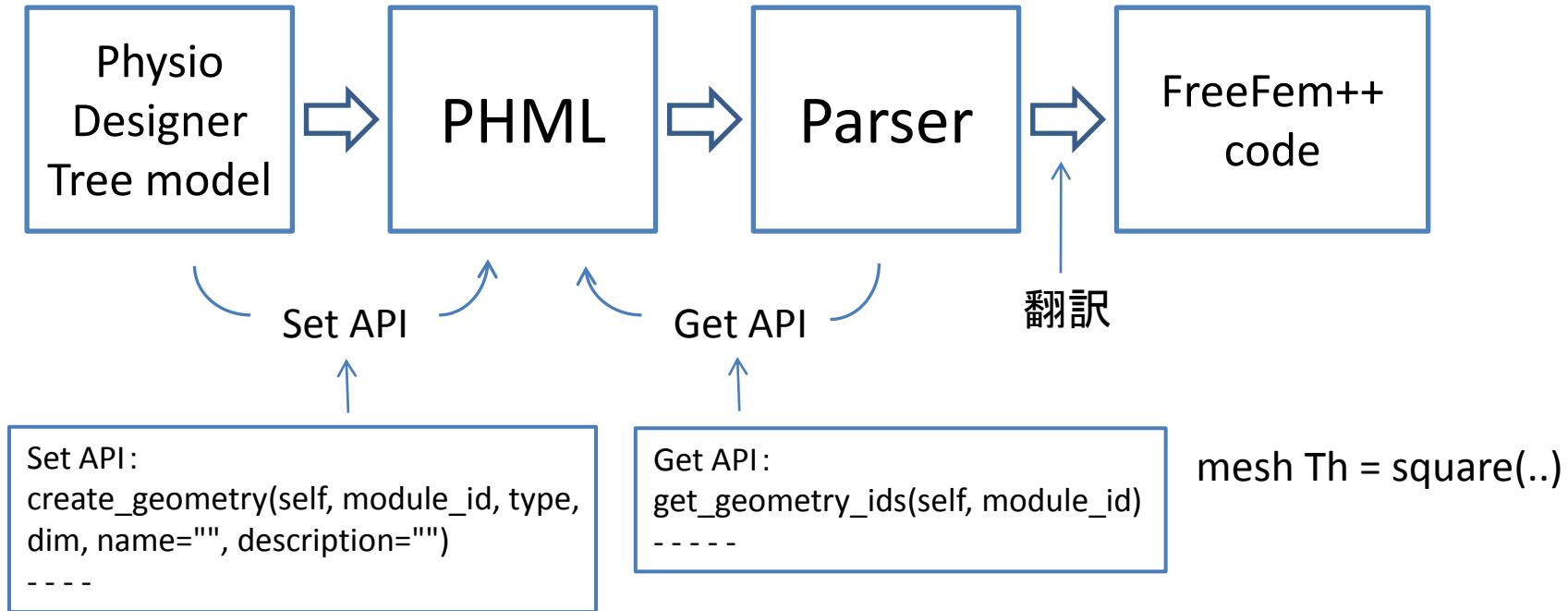
Different realization method

# Target simulation language

- Easy and anyone can do it
    - None scratch program, Small size program
    - Applicable to multi-purpose
    - Combination of morphology definition, simulation and visualization in a same language
    - Public and multi platform
  - Data Base
    - Reuse of simulation execution environment
- We tried OpenFEM, SciLab and FreeFem++



# Simulation Flow of PhysioDesigner



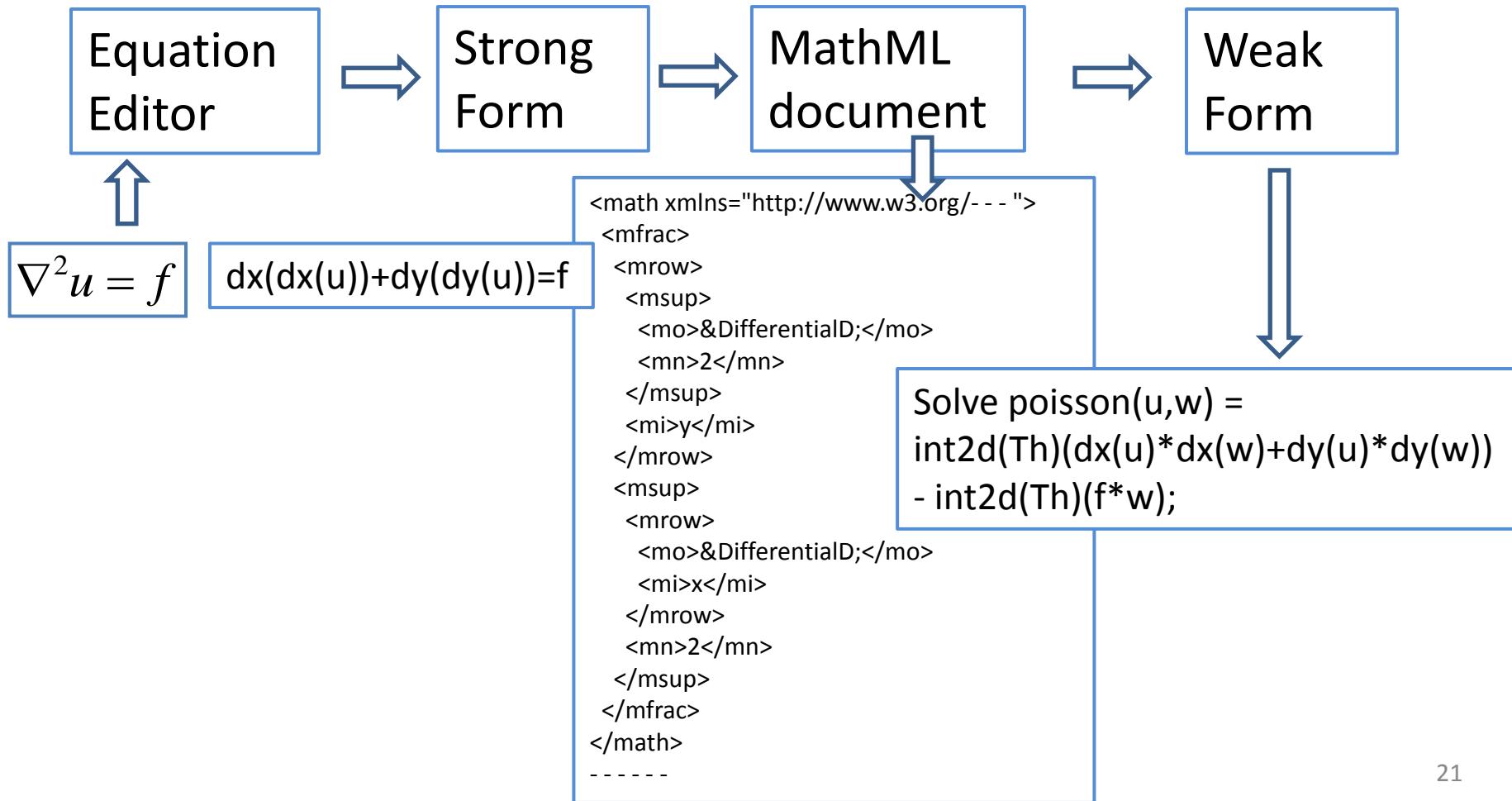
Python API: Use of Application Program Interface

Get ready for APIs for various operations:

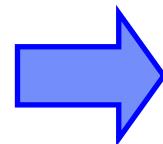
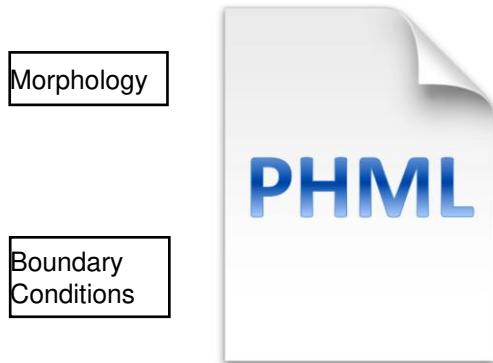
→ Increase Programmable efficiency

# Application Equation

- FreeFEM++ : various application equations



# Morphology and PDEs

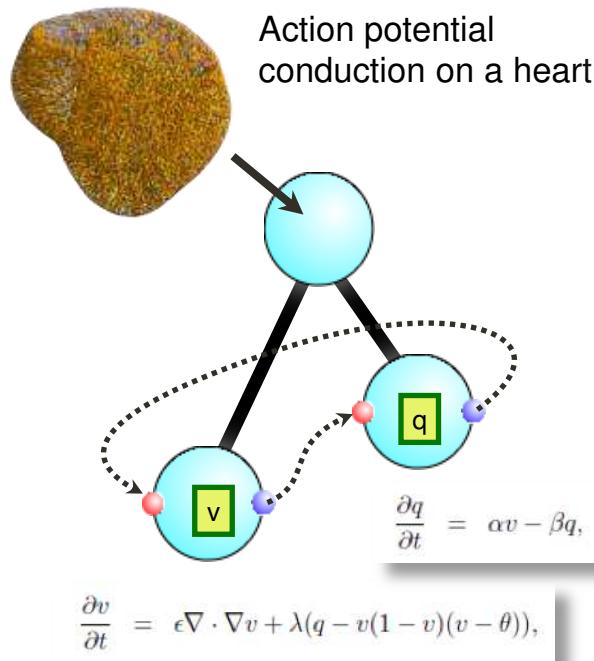


*FreeFEM*

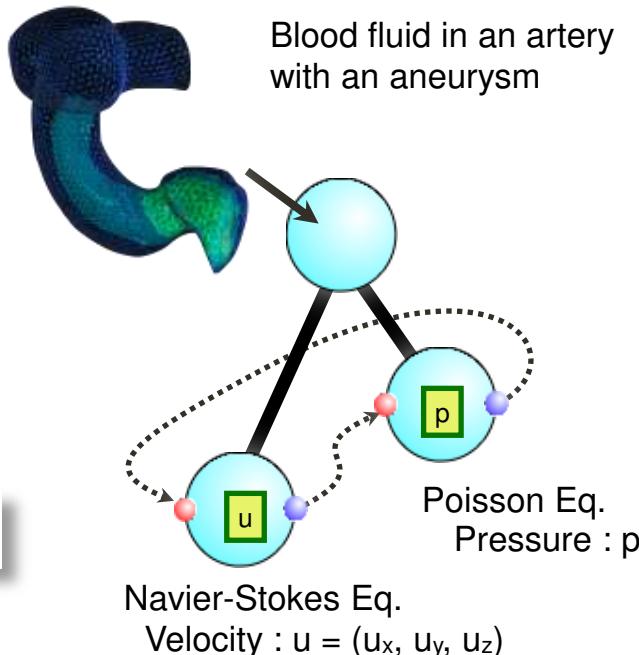
```
mesh Th=square(imax,jmax,[lxmin*imax*x,lymin*jmax*y]);  
for (real t=0;t<tmax; t+=dt){  
    Luo-Rudy current model  
    Bi-domain propagation model  
    plot(v,viso=viso(0:viso.n-1),fill=1,value=1);  
}
```

<http://www.freefem.org/>

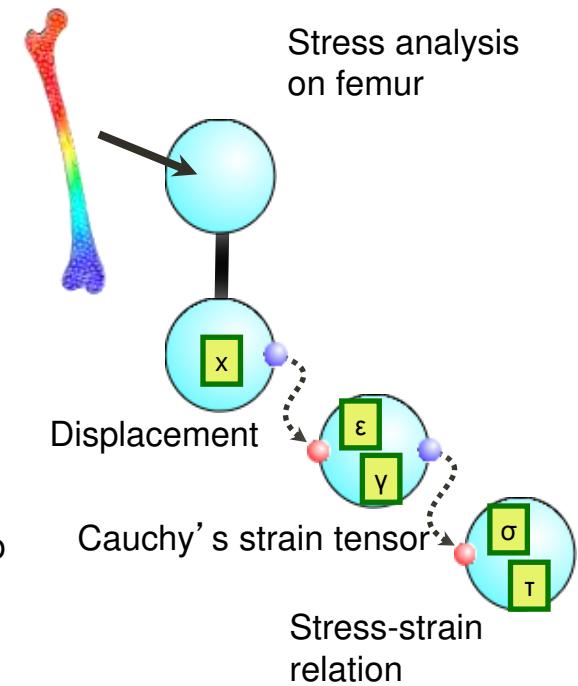
## Cardiac Electrophysiology



## Fluid Mechanics



## Structural Mechanics



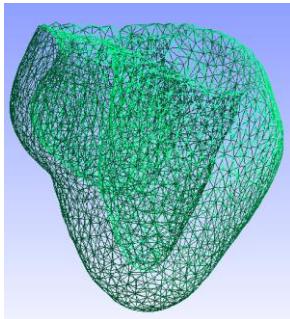
# Example of PHM: FHN3D

## Continuum Model

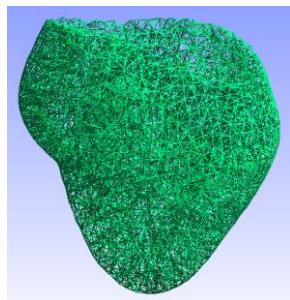
Fitzhugh-Nagumo  
equation

Time dependence of  
Membrane potential

Morphology  
data  
(surface)

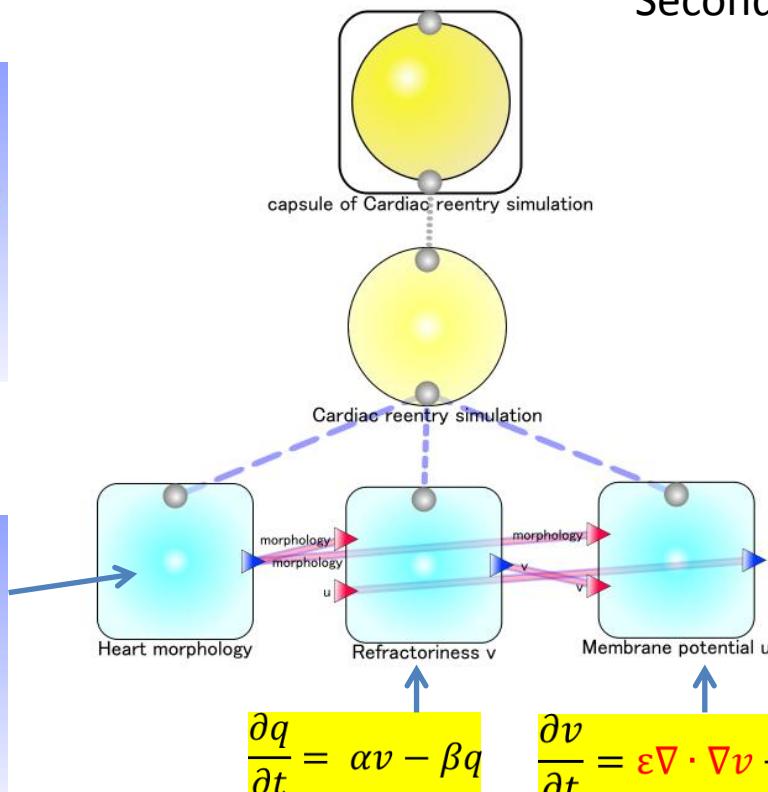
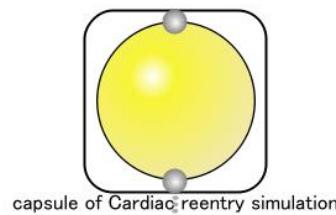


Finite Element Mesh



$$\begin{aligned}\partial v / \partial t &= \varepsilon \nabla \cdot \nabla v + \lambda(q - v(1-v))(v - \theta) \\ \partial q / \partial t &= \alpha v - \beta q\end{aligned}$$

Partial differential equation  
First derivative for the time  
Second derivative for the space



Morphology  
definition  
Is a key

Finite Element  
Method

# Flint

- ODE (Ordinary Differential Equation) Solver

Static instance , Parallel processing

Directly  
Callable from IDE

The screenshot displays three windows of the Flint software:

- Simulation Window - [HodgkinHuxley\_1952\_neuron\_model]**: Shows simulation parameters like Numerical Integration Method (Euler), Simulation Length (100), and Data Output (1 data per 1 step(s)).
- Variables list**: Shows a list of available variables with their module names and IDs. Available Variables include h\_gate, beta\_h, alpha\_h, V, n, g\_K, I\_K, I\_Na, I\_leak, e\_leak, e\_Na, h, I\_K, I\_leak, I\_Na, m, and n. Enabled Variables are I\_K, I\_Na, I\_leak, e\_leak, e\_Na, h, I\_K, I\_leak, I\_Na, m, and n.
- Plot**: A graph showing the membrane potential V over time. The plot has three ordinates: Ordinate(Left) for I\_K, Ordinate(Middle) for V, and Ordinate(Right) for I\_Na. The x-axis is labeled "Abscissa" and the y-axis is labeled "time".

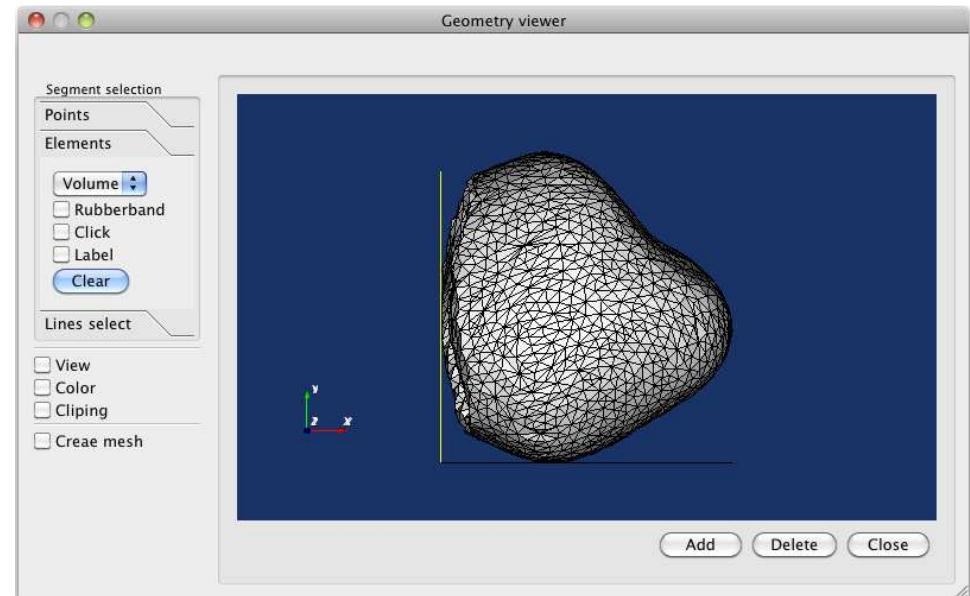
**Timestep**  
**Selection of Solver**  
(Euler,RungeKutta)

# Biological Morphology Visualization

Objective : Gives morphology information to partial differential eqaution



Image viewer (Medical Image)



Morphology editor (Morphology Definition)

# Imageviewer/Morphology editor

Use VTK library

- **Image viewer** Image view, material value and shape extraction  
Extract biological material parameters from medical image
- **Morphology editor (Geometry)** Shape (Solving space) definition  
Generate Surface model (surface mesh generation)  
Multiple methods → Primitive, External, Vertex, Parametric
- **Morphology editor (Segment)** Boundary condition's application  
area definition → Point, Line, Surface, Volume, Group

# View function of ImageViewer

## Morphology DB

**Physiome.jp**

Home  
insilicoML  
insilicoIDE  
downloads  
tutorial  
FAQ  
**Morphology DB**  
Time-series DB  
Term of use  
About  
Publications  
Forum  
Wiki  
Contact  
Links

**Supported by:**  
MEIcenter  
SBI  
DYNAMIC BRAIN  
FUJITSU  
JAPANESE GOVERNMENT

**Description:**

**Model Data**

MorphologyID: MPID000011  
File Type: ZIP > download  
File Name: MPID000011\_brain\_structure\_data.zip  
Last Update: 2011-08-22 20:26:57  
Published ID:

The following five files readable on MATLAB are included in the archive.  
MorphologyDB: 3D MRI structure image including all tissues.  
grey matt: White matter  
white matt: White matter  
skullBone matt: Skull  
csf matt: Other tissues mainly including cerebrospinal fluid (CSF)

Structural MRI (T1-weighted image) was acquired with a 1.5T MRI (Magnex Elosse, Shimadzu-Magnex Shimadzu). A total of a male subject (33 years-old) with following parameters: repetition time = 20ms, echo time = 2.38ms, field of view = 256mm, matrix = 256 x 256, voxel size = 1mm x 1mm x 1mm. The structural MRI image was normalized to a standard T1 structural image (T1.nii) by using SPM5 (Ashburner & Friston, 1999). After segmentation, the normalized structural MRI image (MPID000011\_brain\_structure.mat) was segmented into three components: grey matter (grey matt), white matter (white matt) and others. The skull was also peeled from the normalized structural MRI image by using BET software (Smith, 2002) (Brain Extraction Tool). Then, the skull was reconstructed from the normalized structural MRI image with the normalized structural MRI image with the skull strip MRI image, the skull data was constructed (skullBone matt). Furthermore, by combining the normalized structural MRI image with the skull, grey matter, and white matter images, the cerebrospinal fluid was obtained (csf matt). All structural images are saved as a Matlab™ mat format (Mathworks <http://www.mathworks.com>) with a variable img (3D matrix data).

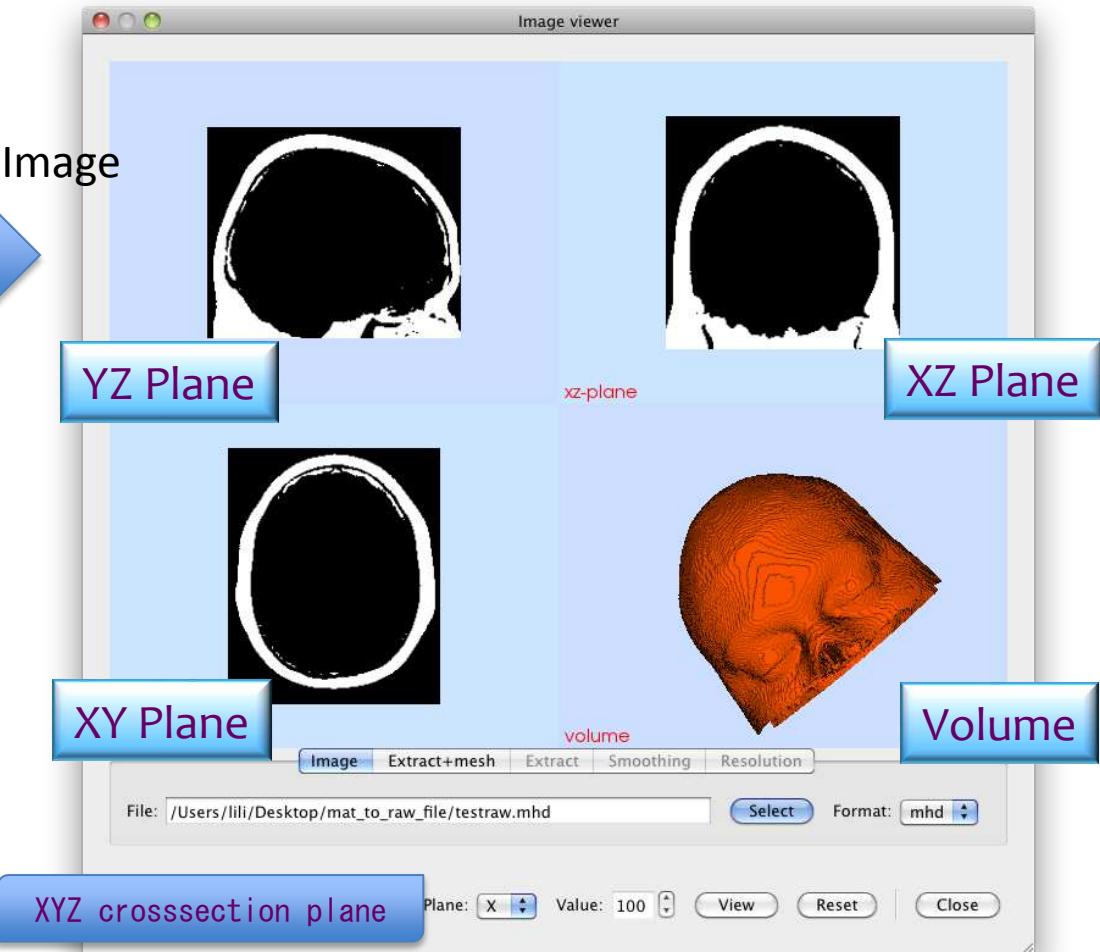
[References]  
Ashburner, J and Friston, K.J. (1999) Nonlinear spatial normalization using basis functions. Human Brain Mapping 7: 254-269.  
Ashburner, J and Friston, K.J. (2005) Unified segmentation. NeuroImage 26: 839-851.  
Smith, S.M. (2002) Fast robust automated brain extraction. Human Brain Mapping 17: 143-155.

**Author Data**

Name	Hiroaki Mizuhara
Organization	Graduate School of Informatics, Kyoto University, Japan

All contents copyright(c) 2008-2011 MEIcenter All rights reserved.

Medical Image

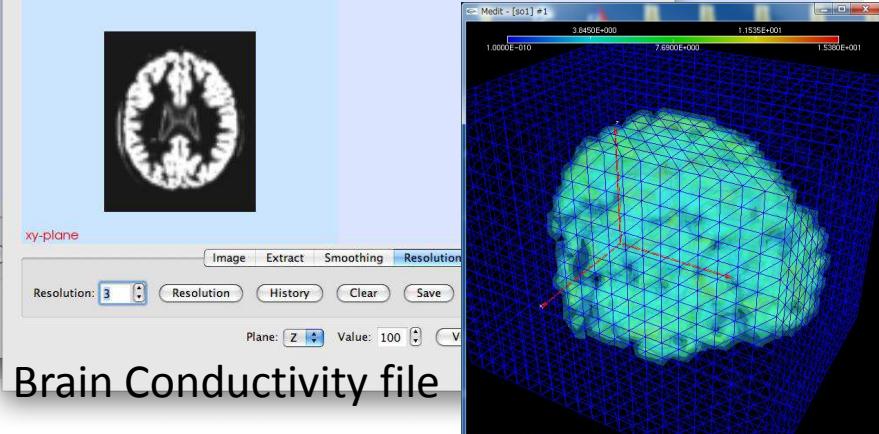
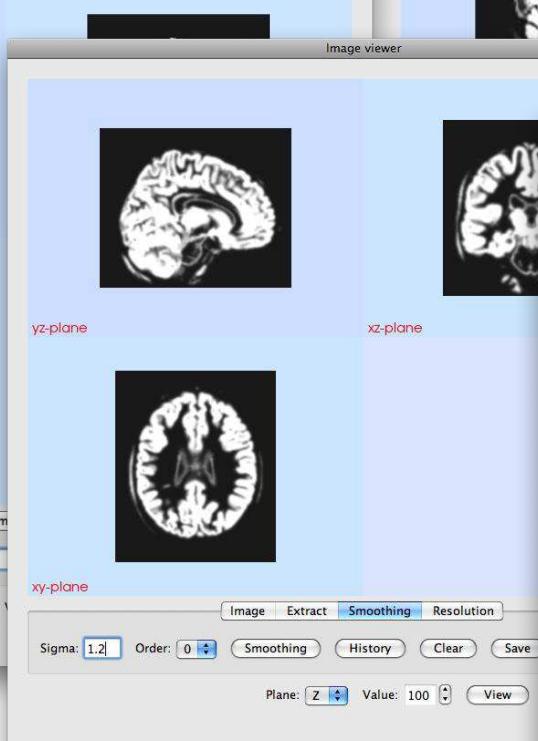
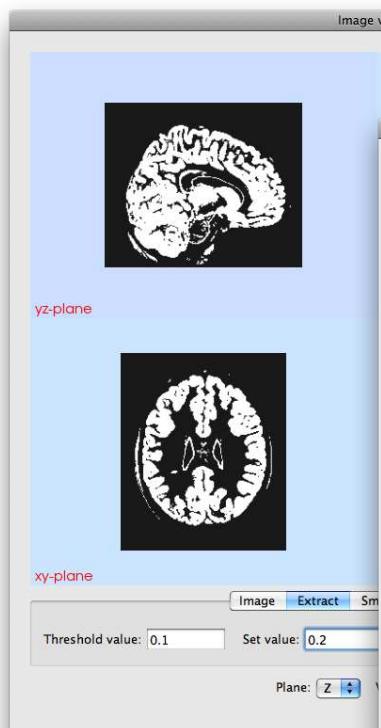
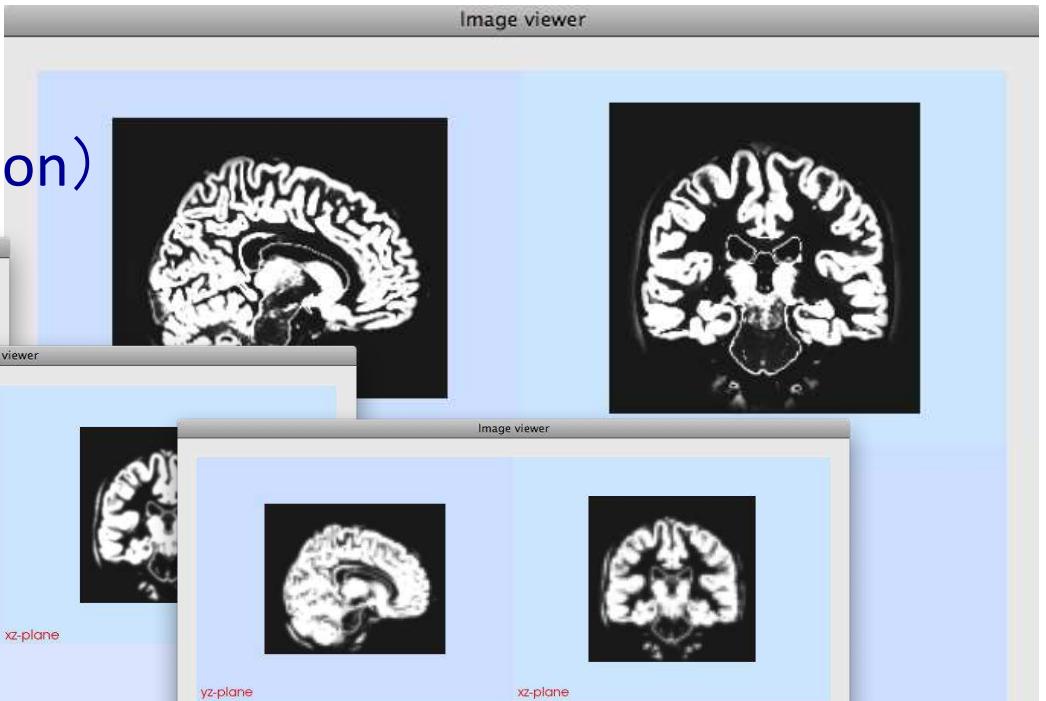


- ◆ Image Format: MATLAB , Metaimage , RAW , Analyze, Nifftiti
- ◆ View function : Cross section and 3D view

# Image Processing function of Image viewer

- ◆ Border extraction (Extract)
- ◆ Smoothing (Smoothing)
- ◆ Change Resolution (Resolution)

Original Image

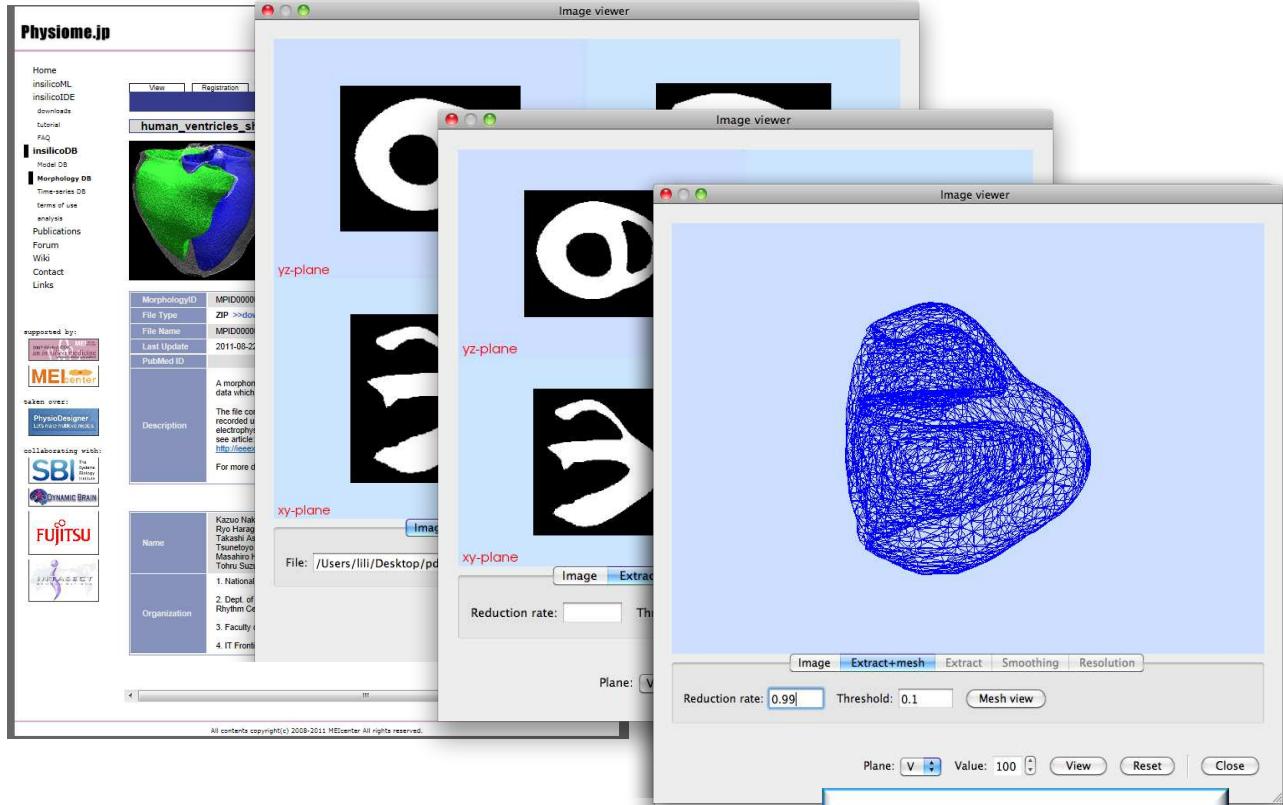


Brain Conductivity file

# Segmentation function of ImageViewer

- ◆ Shape extraction from a Raw Image and make Surface model

## Morphology DB



Make  
Surface model  
mesh

By Morphology editor  
make Volume mesh



Surface mesh file

# Function of Morphology editor

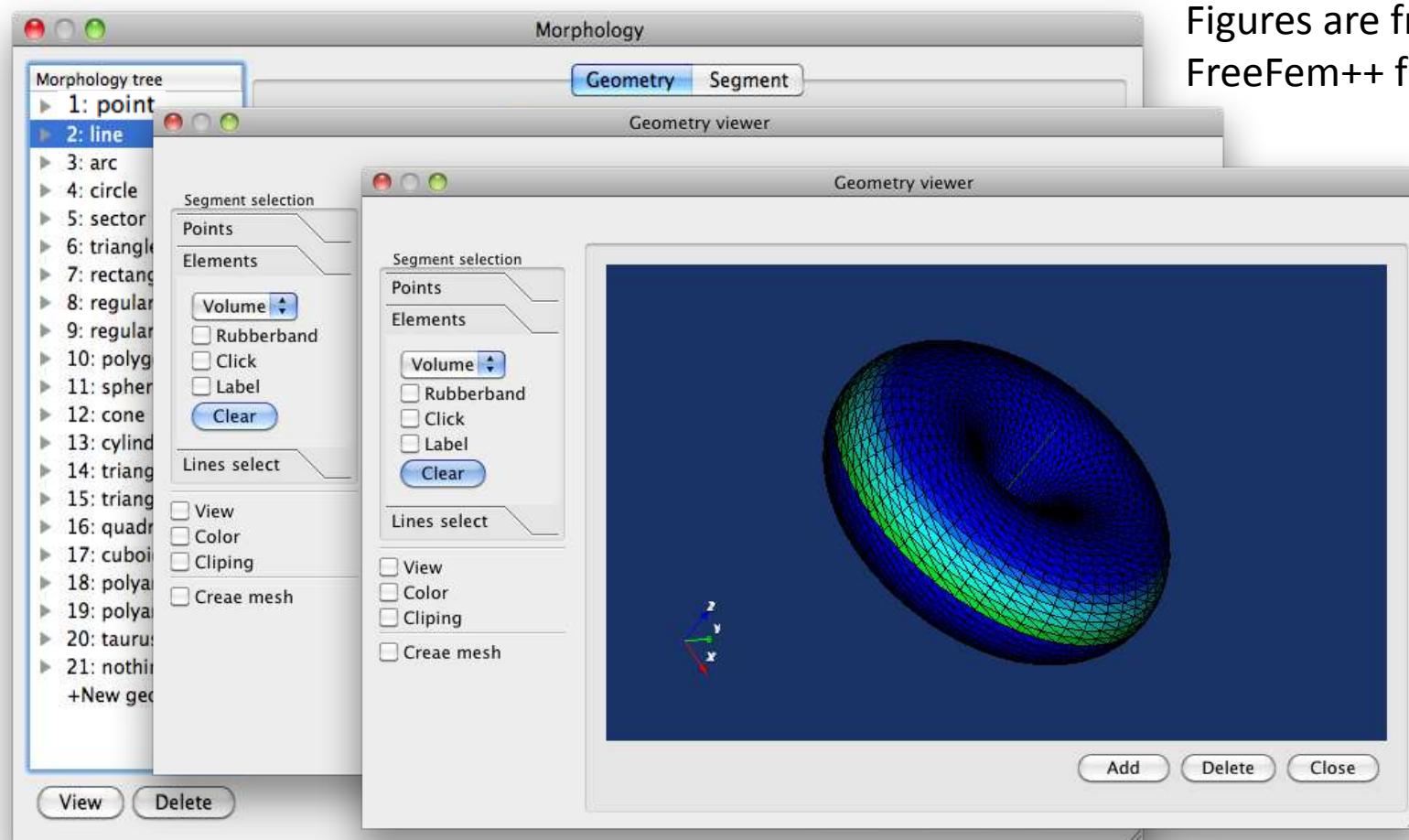
## Kind of morphology definition

- ✧ Primitive : Pre-build primitive morphologys
- ✧ External : External morphology file
- ✧ Vertex : From given node points
- ✧ Parametric : From mathematical equations

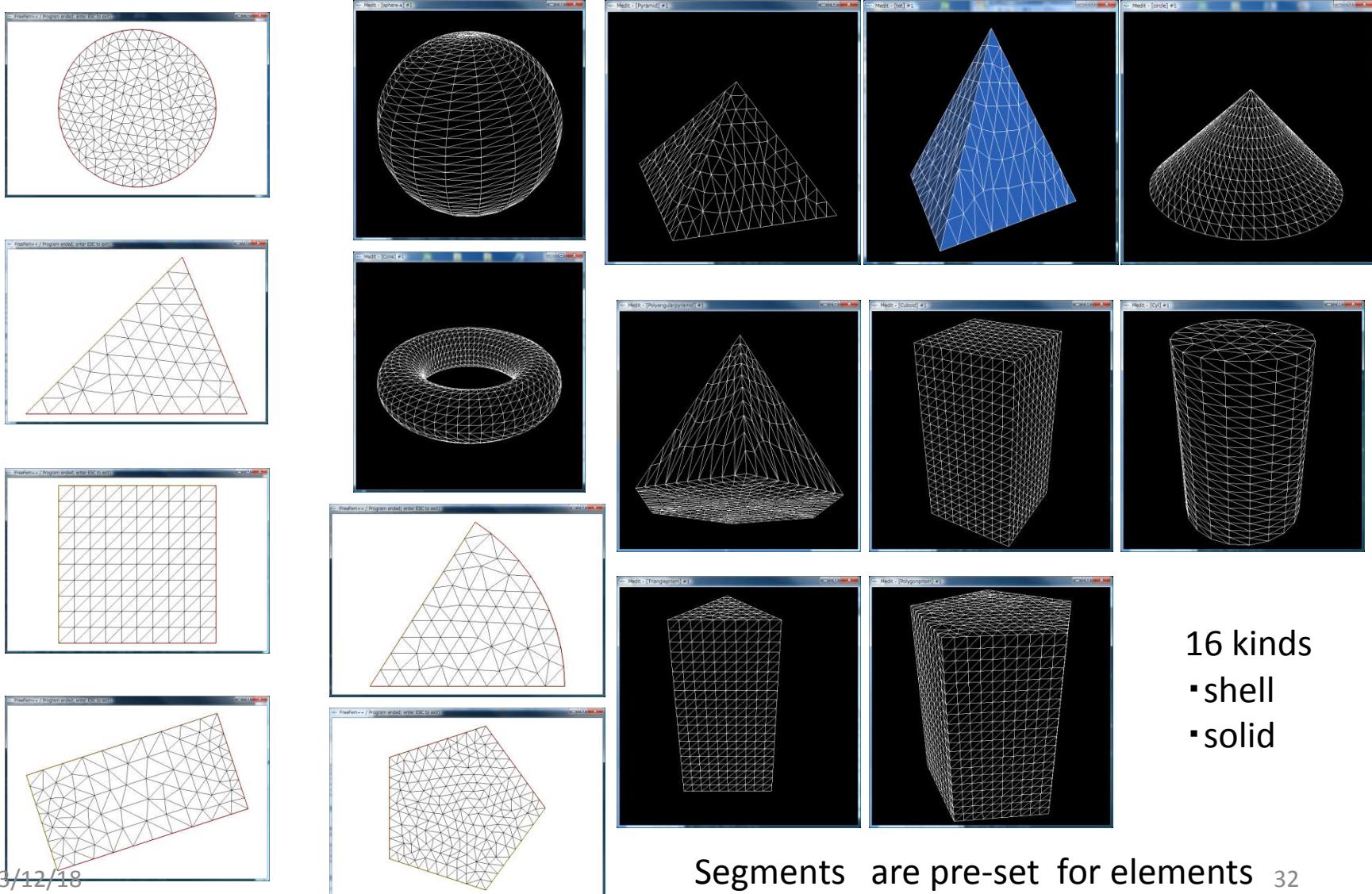
# Editor function of Morphology editor

## Primitive

- ◆ 20 pre-build primitive figures (Triangle, Polygon, cuboid etc.)  
Make morphology from Single or multi number of primitives



# FreeFem++ primitive

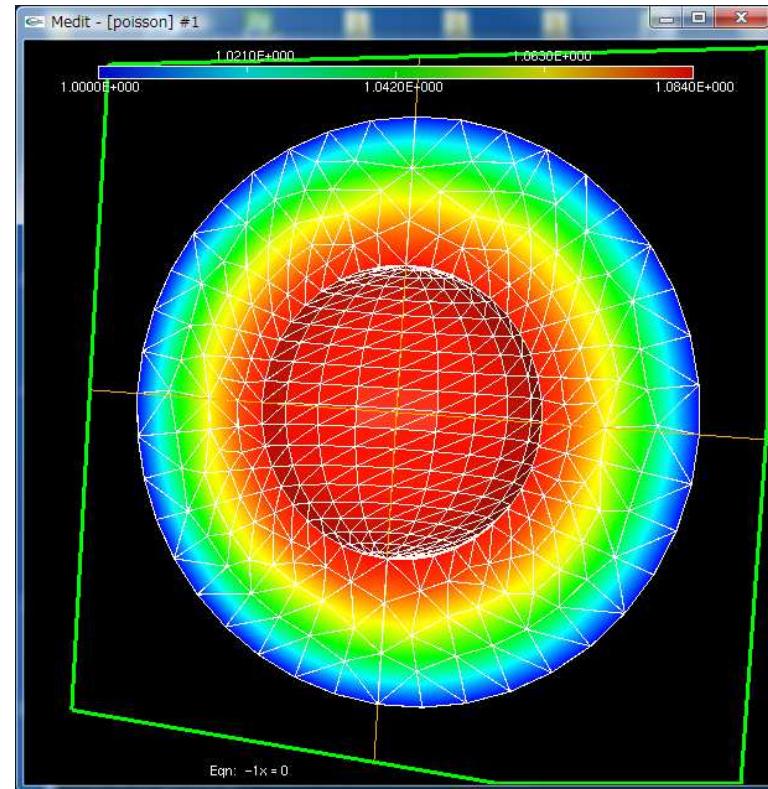
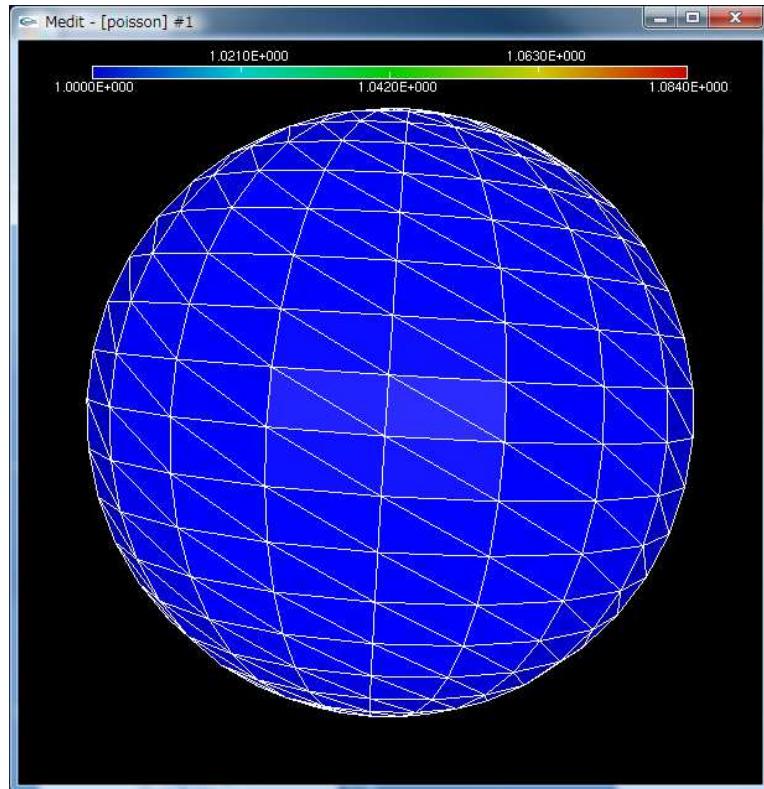


16 kinds  
▪ shell  
▪ solid

Segments are pre-set for elements 32

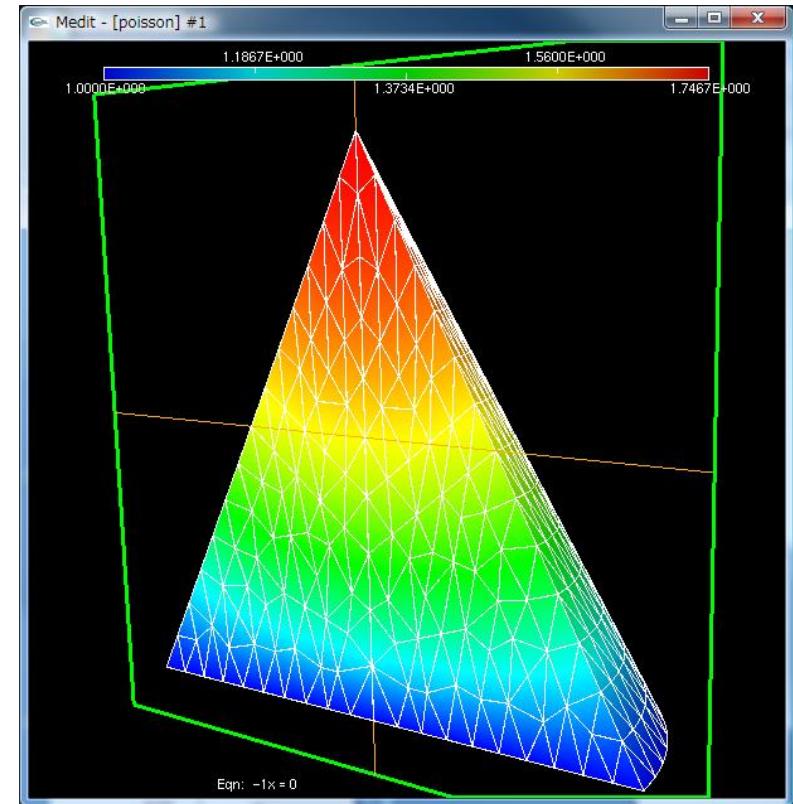
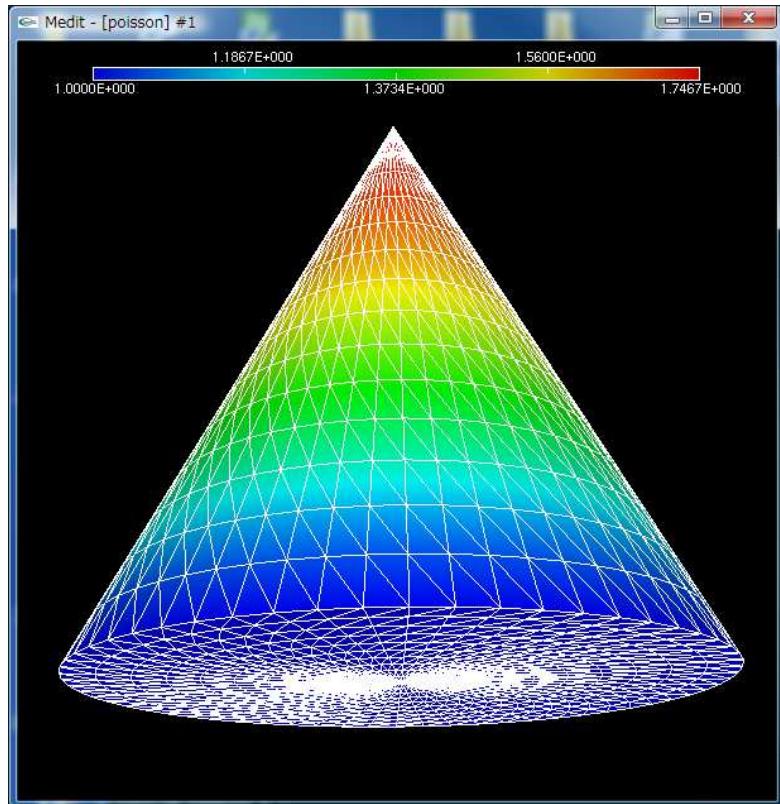
# Example 1(sphere)

$R_{\text{out}}=1.0$   $R_{\text{in}}=0.5$  boundary condition:  $u=1.0$  on outer surface

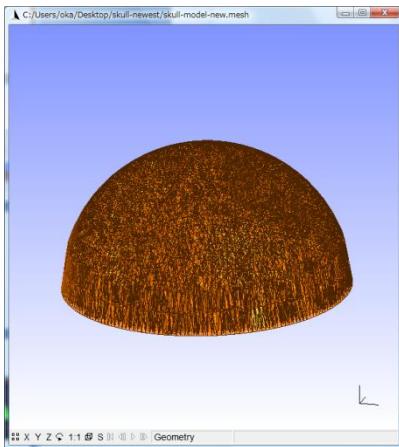


# Example2(Cone)

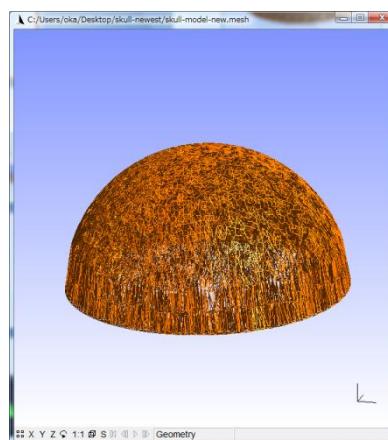
R=H=1.0 Boudary condition: u=1.0 on bottom surface



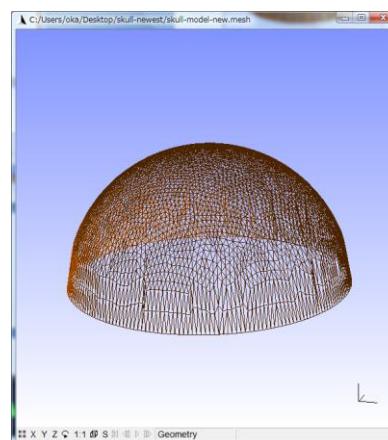
# All mesh



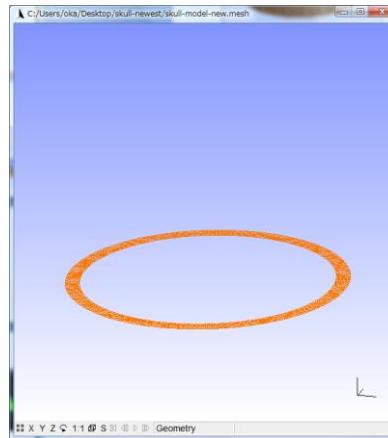
1



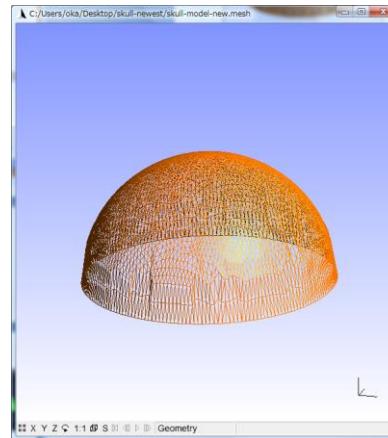
2



4



3



**Visibility**

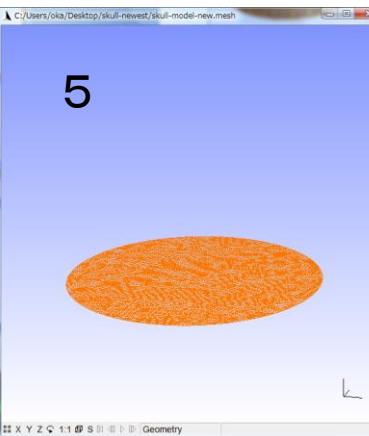
List browser Tree browser Numeric Interactive Per window

Type	Number	Name
Volume	1	
Volume	6	
Surface	2	
Surface	3	
Surface	4	
Surface	5	(label)
Surface	7	

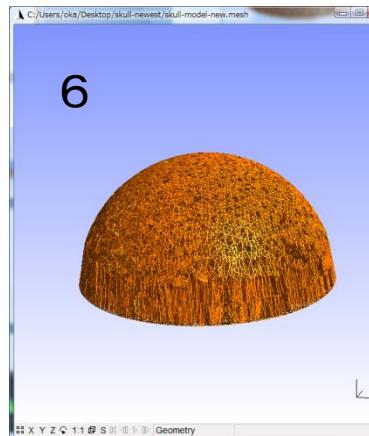
Elementary entities

Set visibility recursively

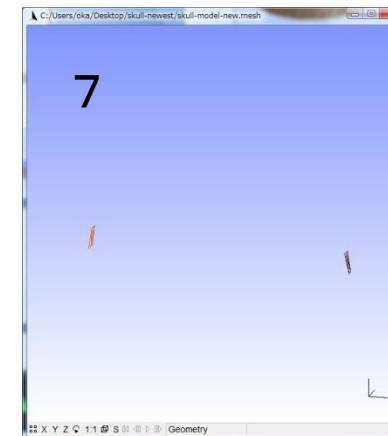
By gmsh



5

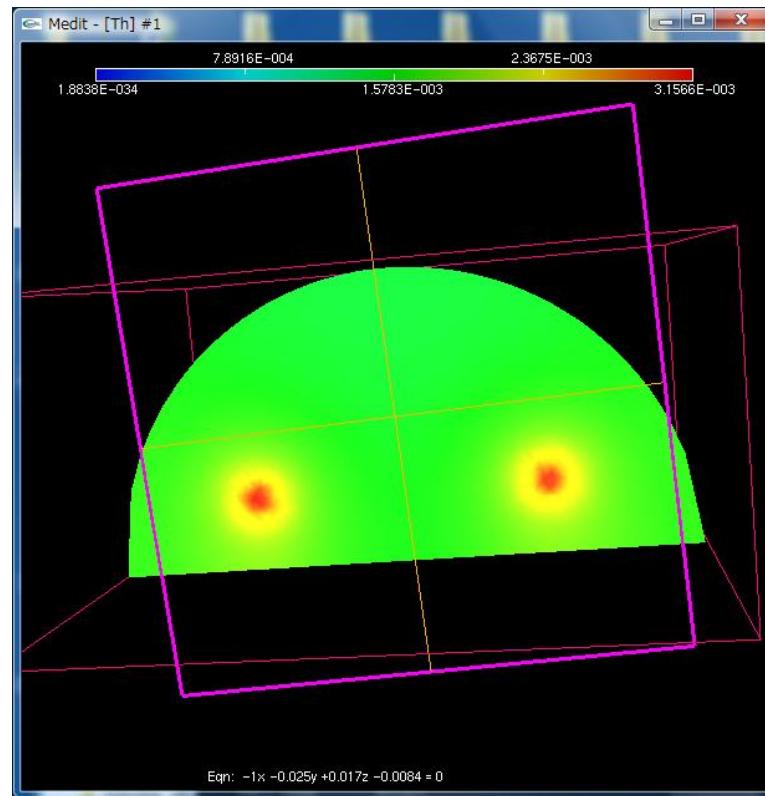
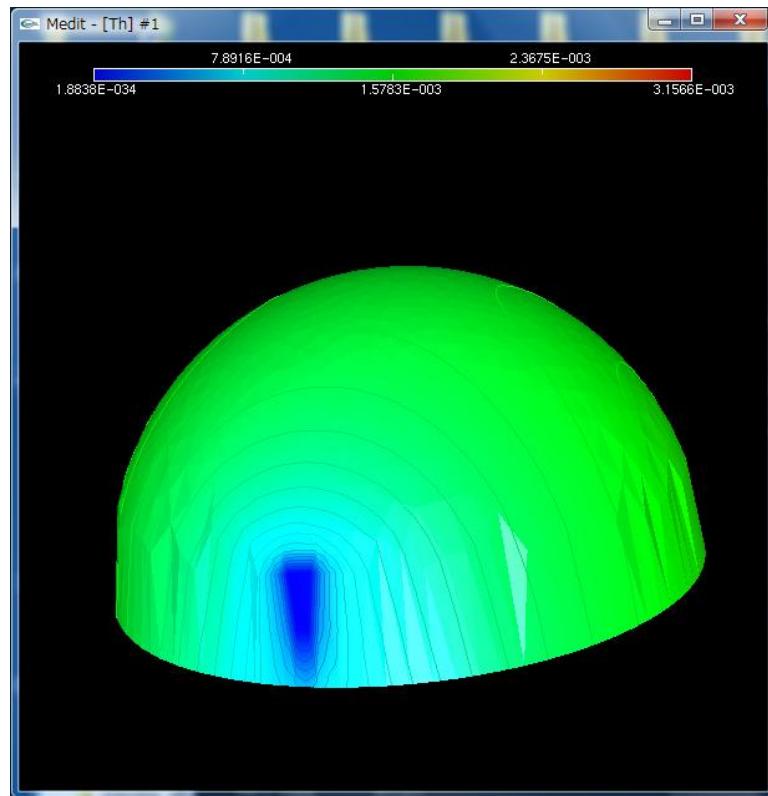


6

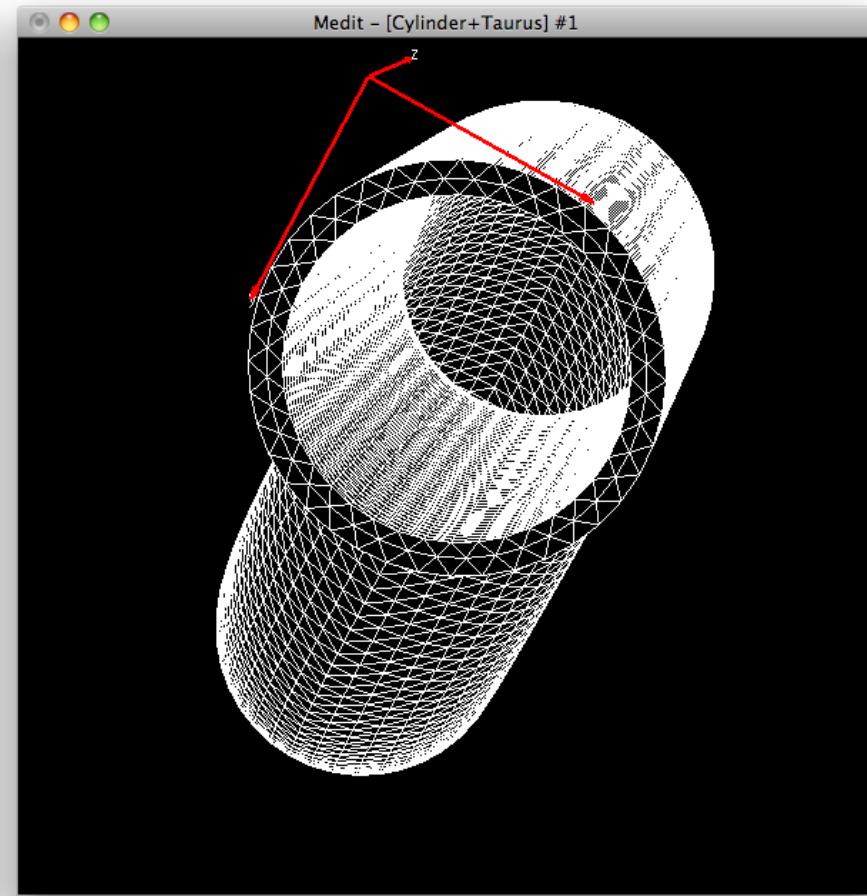
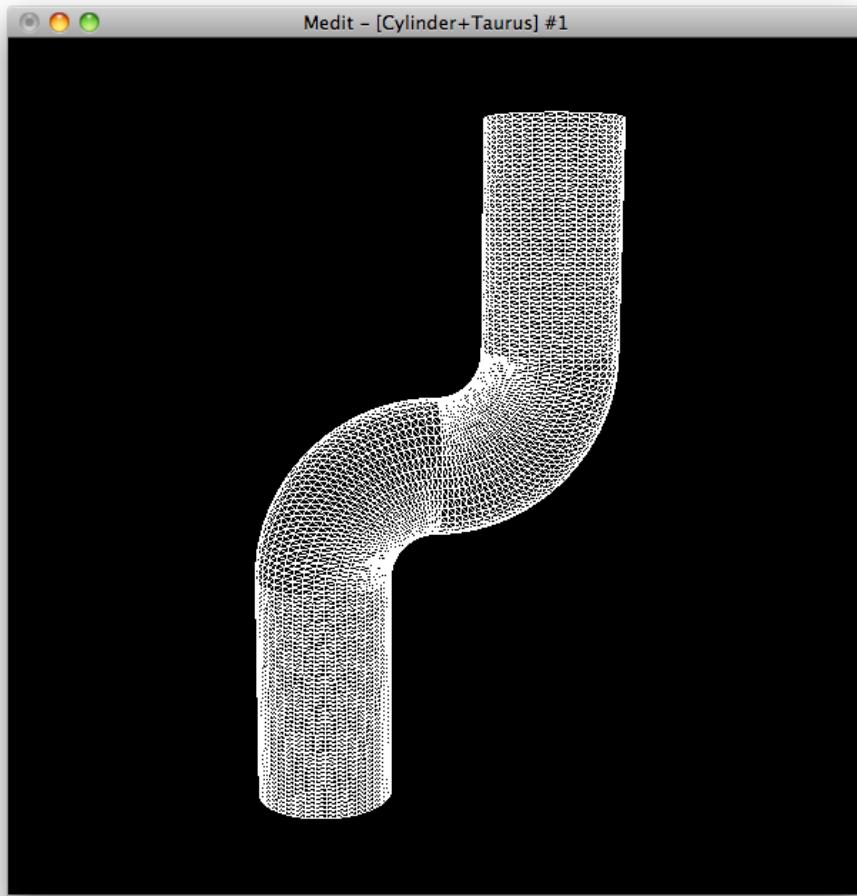


7

# EEG calculation for skull model



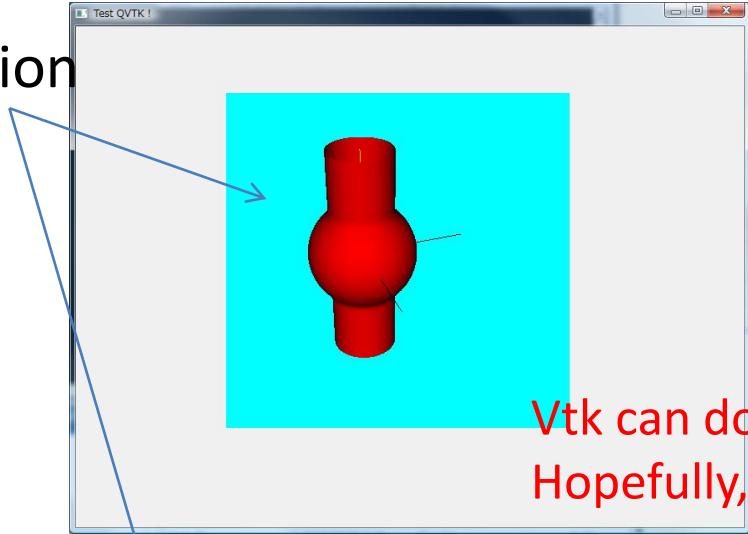
# Blood Tube model



# Boolean operations of primitives (Sphere and Cylinder)

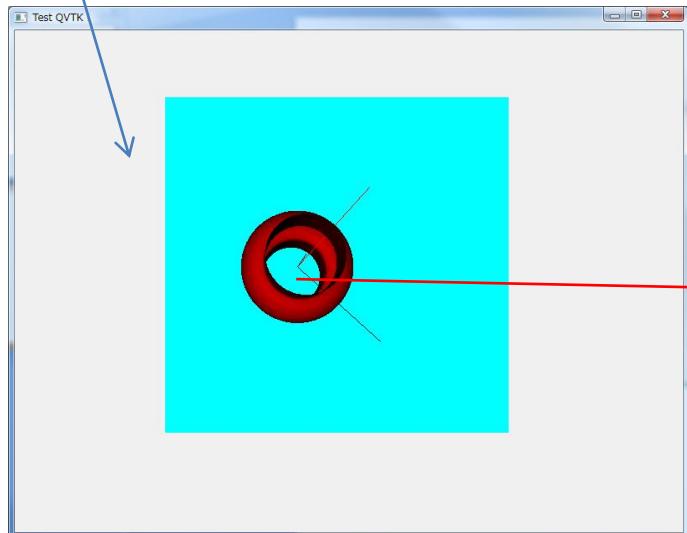
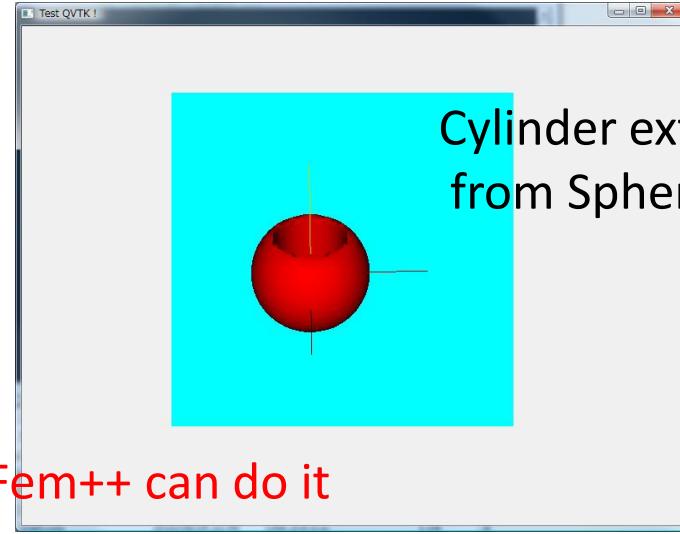
Difference

Union



Vtk can do it  
Hopefully, FreeFem++ can do it

Cylinder extracted from Sphere

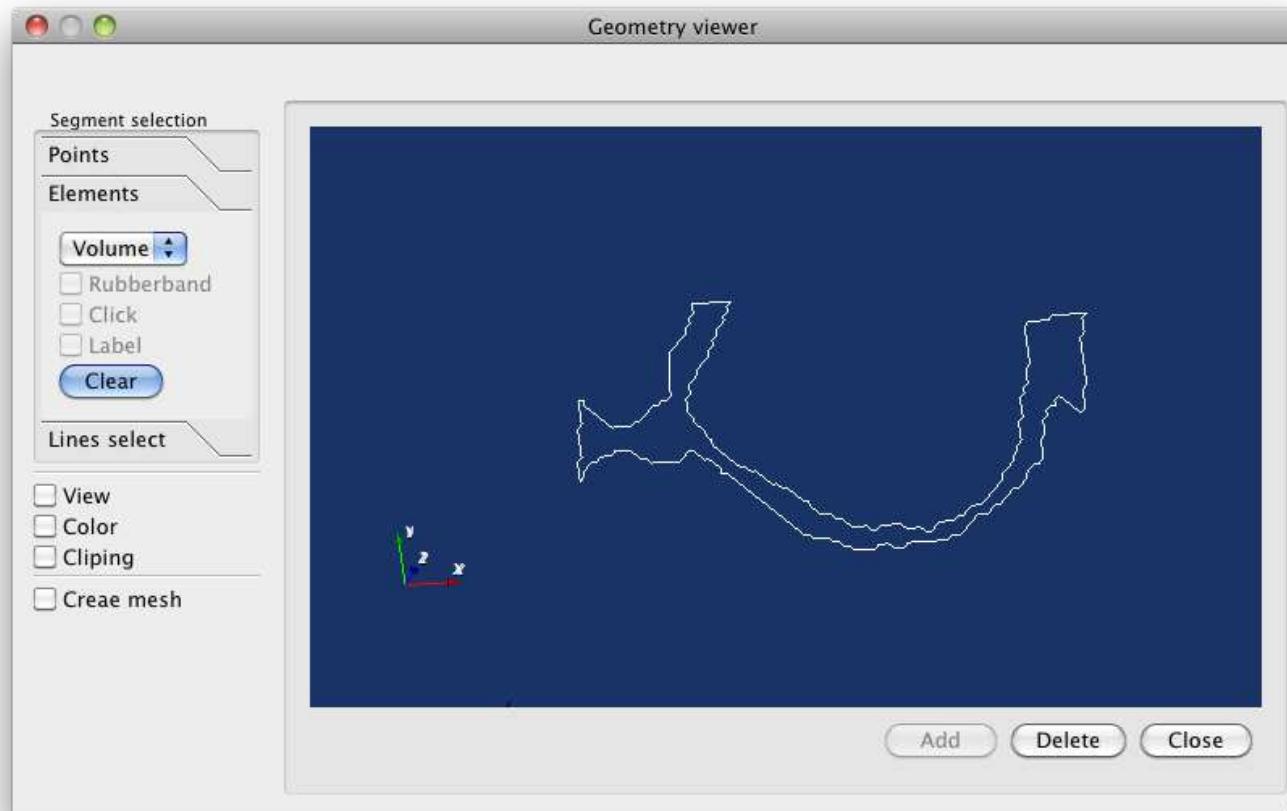


Intersection

# Morphology editor

## Vertex

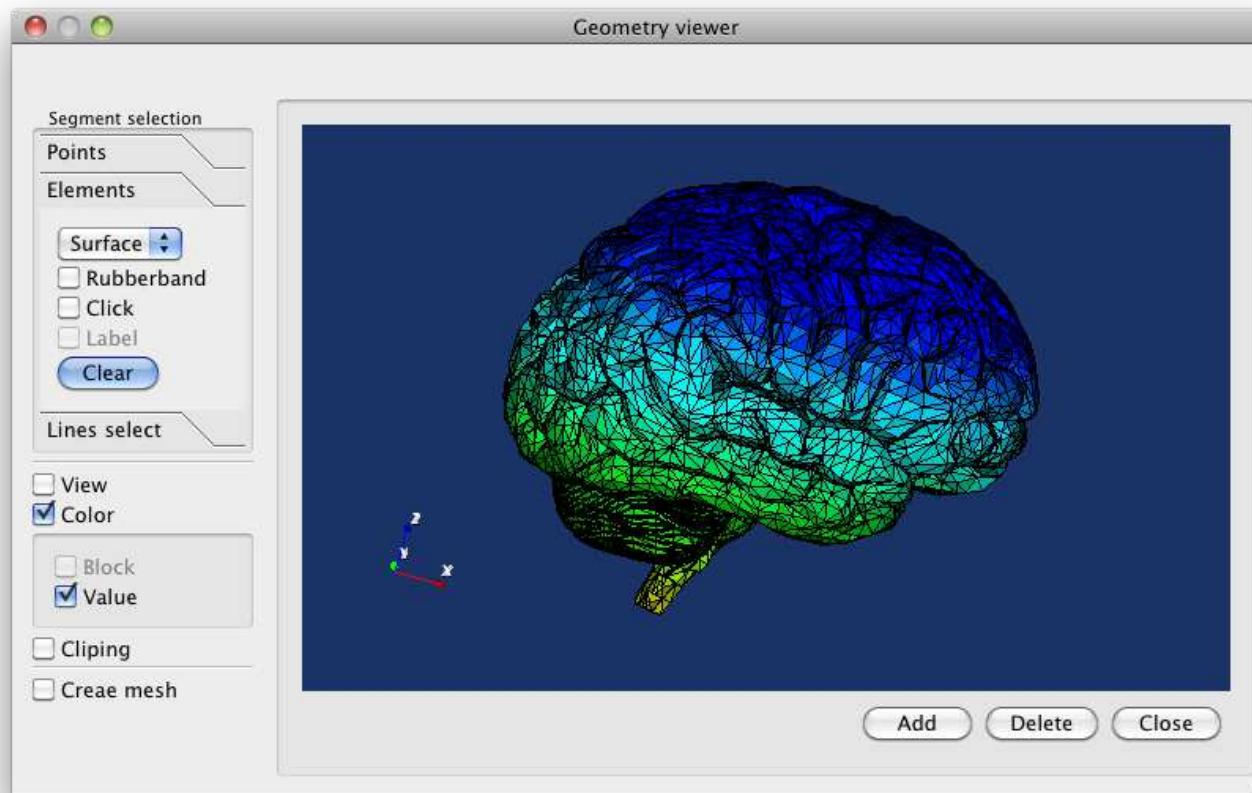
- ◆ Construct a morphology giving points and elements (node, triangle element and tetrahedral elements)



# Morphology editor

## External

- From external morphology file, visualization by Geometry Viewer, segment definition and Volume mesh generation can be done.



Support morphology file : VTK, VRML, STL, medit, gmsh

# Morphology definition in FreeFem++

- Primitive

include "cuboid.idp"

```
mesh3 Th = Cuboid(NX,NY,NZ,...)
```

- External

```
Th = readmesh3("cuboid.msh")
```

- Vertex

```
border Ci(t=0,1){x=xx[i]+t*(xx[i]-xx[i-1]); y=yy[i-1]+t*(yy[i]-yy[i-1]); label = i;}
```

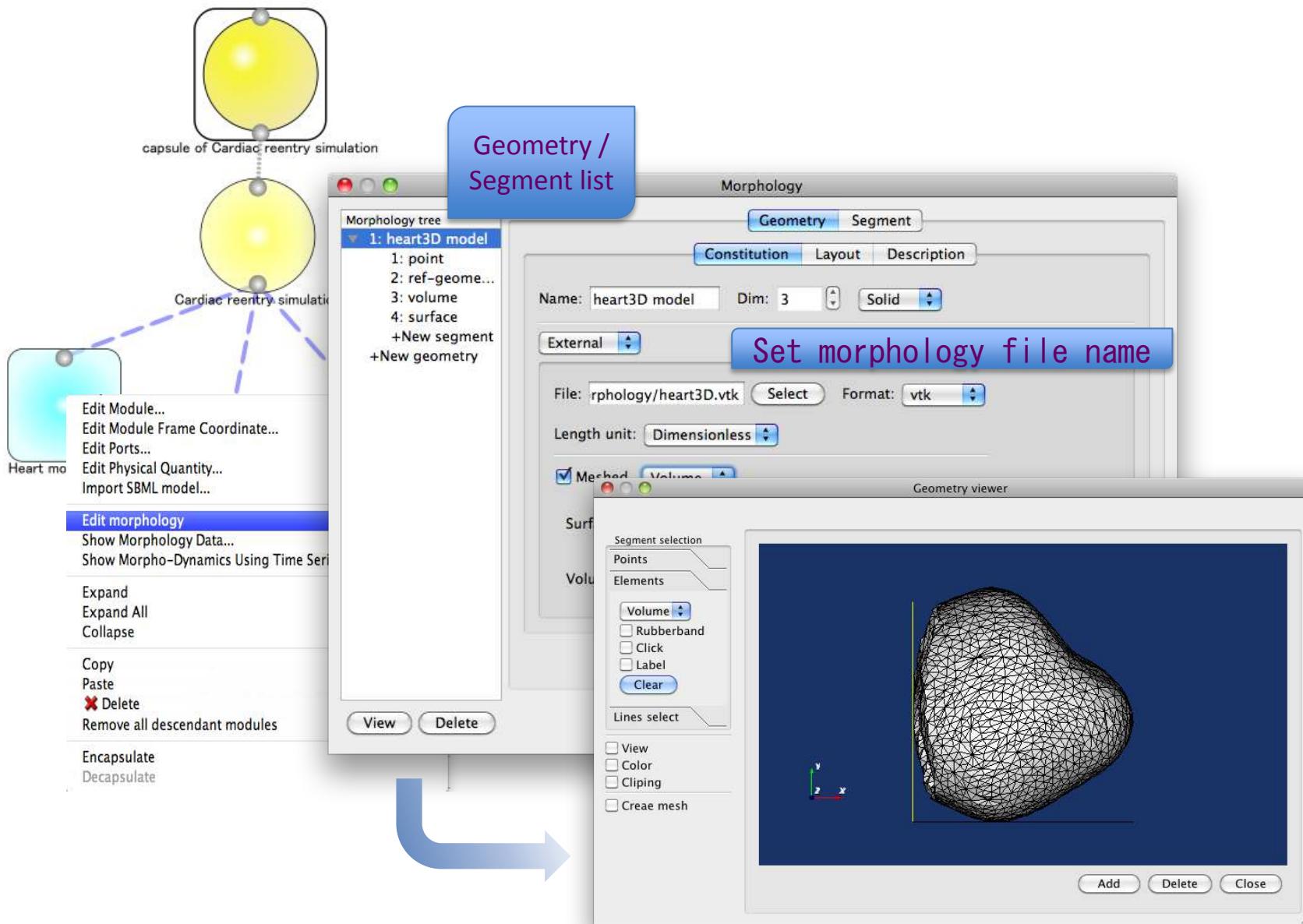
```
mesh Th = buildmesh(C0(1)+C1(2)+....+CN(1));
```

- Parametric

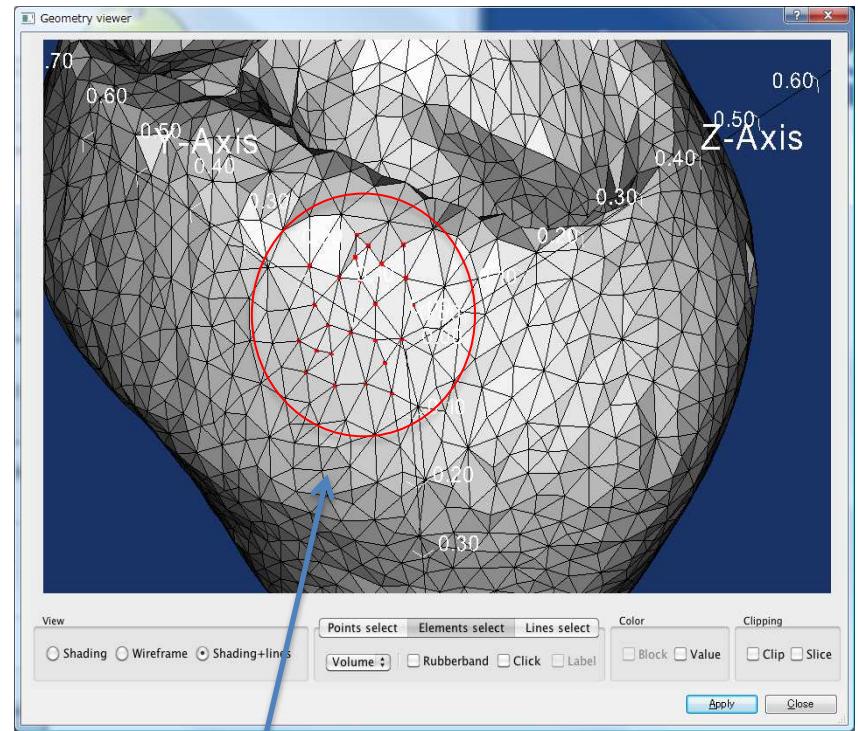
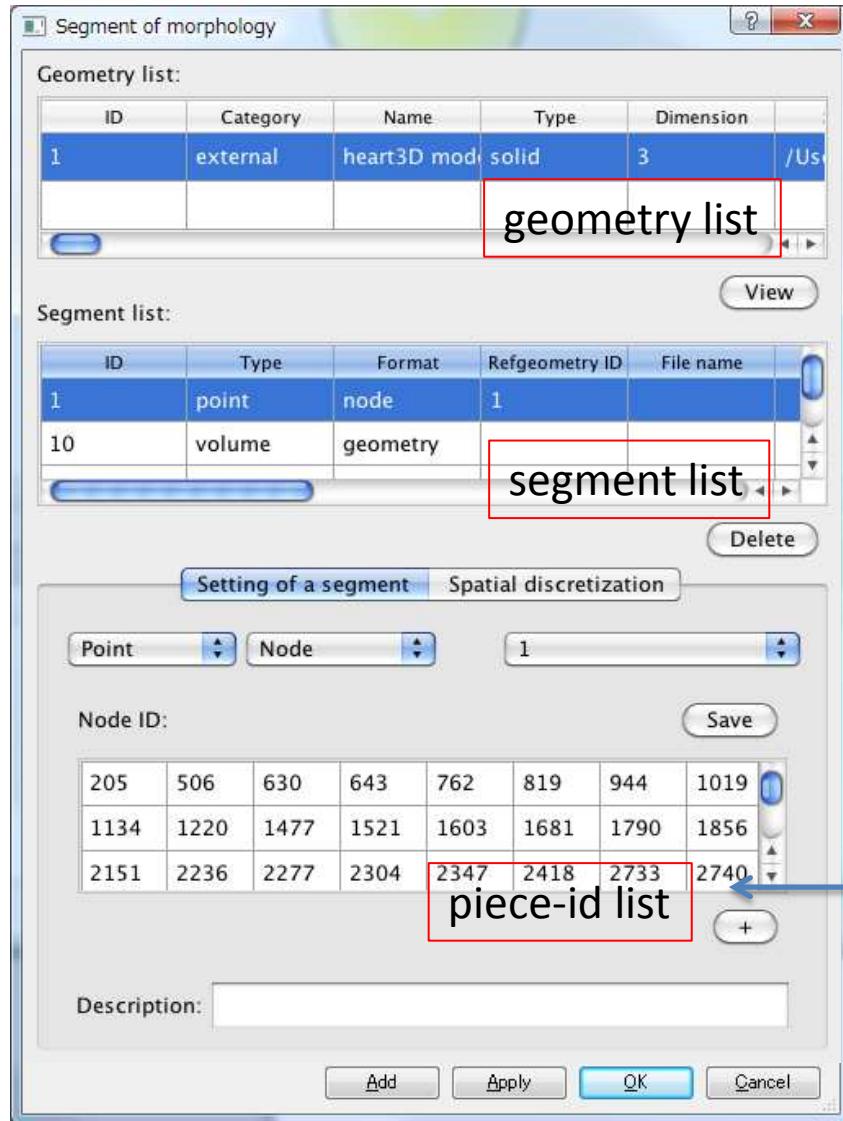
```
border gamma(t=0,pi) { x=cos(t); y=sin(t);}
```

```
mesh Th = buildmesh(gamma(20));
```

# Morphology definition window



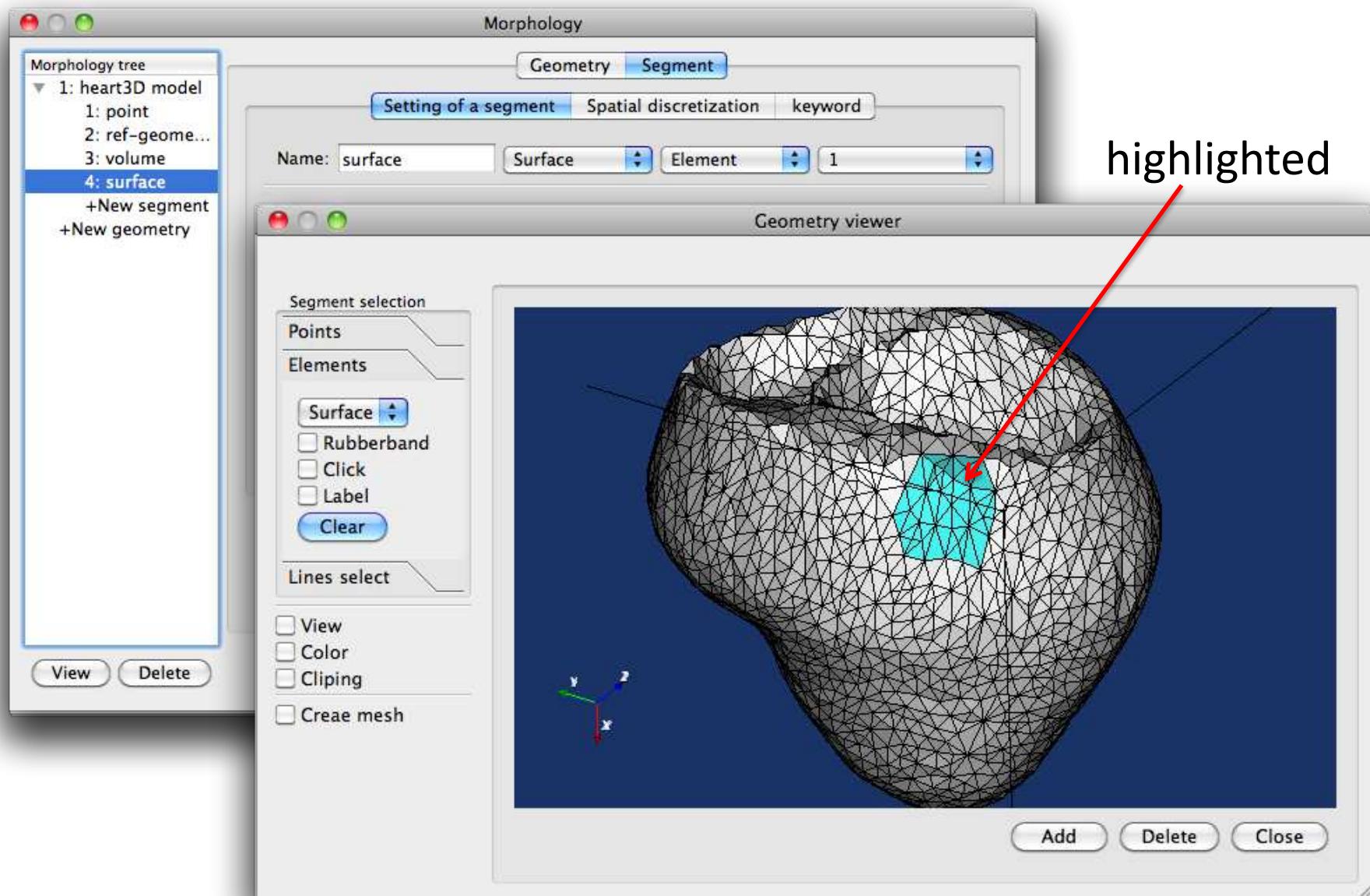
# Segment definition windows



Select node,element(triangle,tetradedra)  
by mouse operation → segmentation

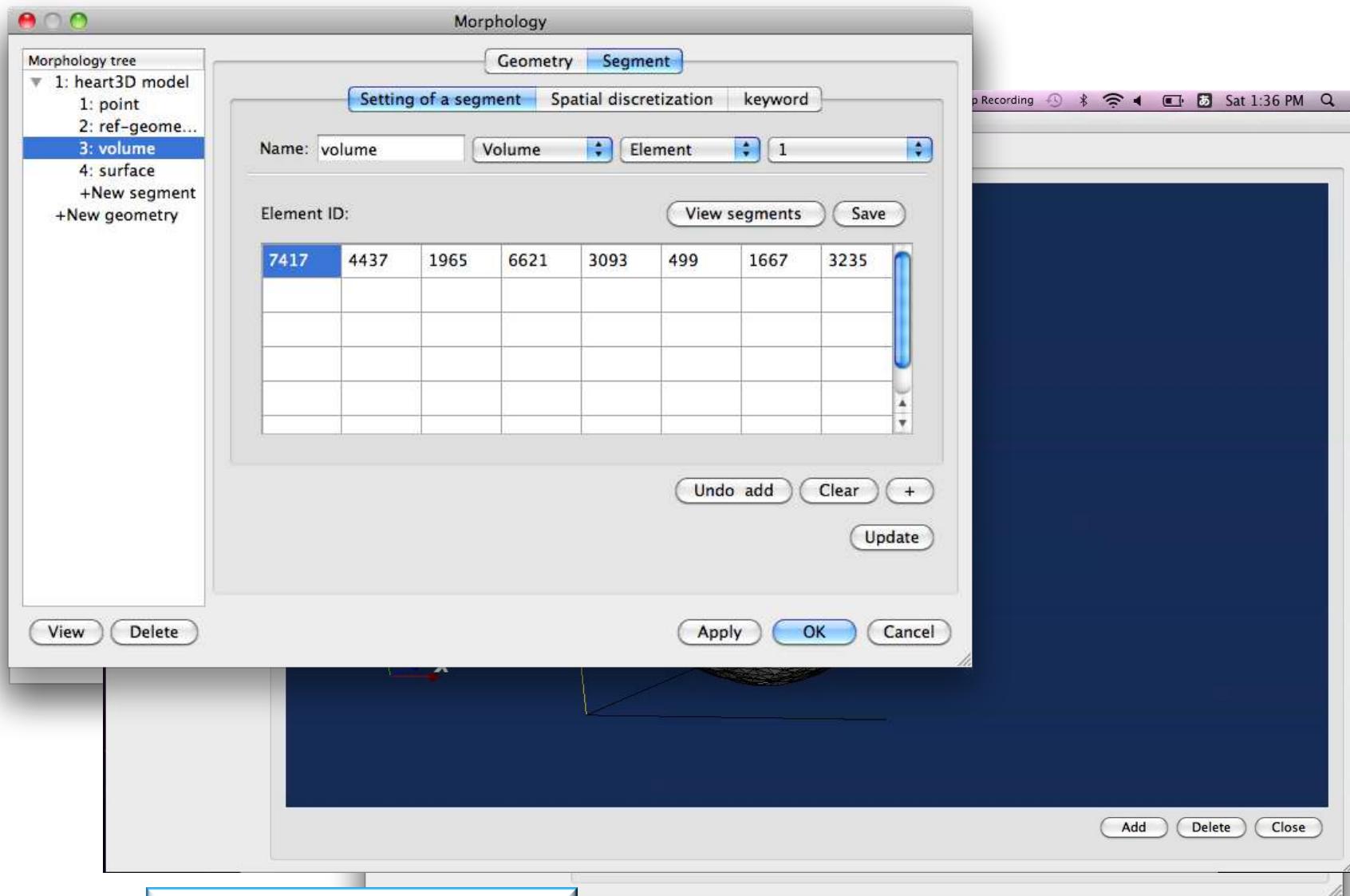
mouse pickup , rubberband operation

# Morphology Segment definition



highlighted

# Segment definition function

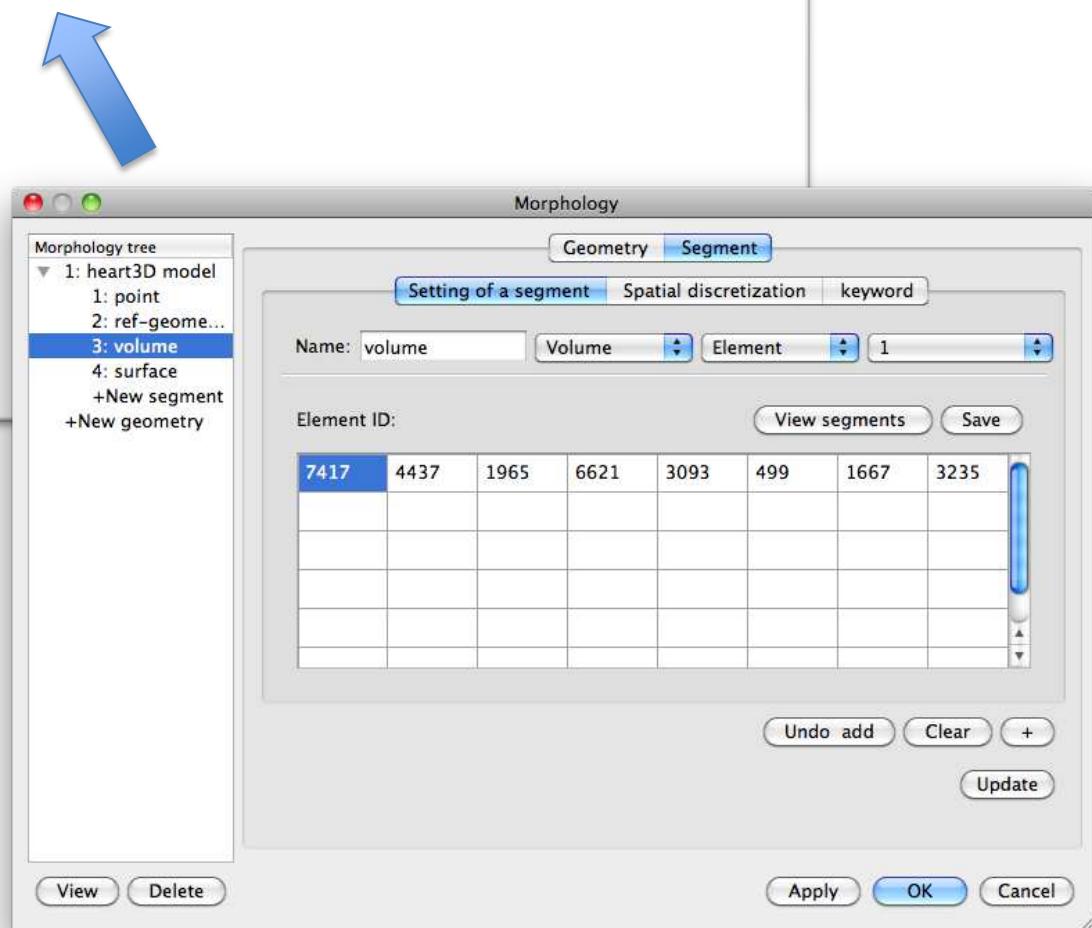


Apply/OK save to PHML

# Morphology Segment definition

Write to or Read from PHML model

```
- <is:segment segment-id="3" type="volume" format="element" ref-geometry-id="1">
  <is:description></is:description>
  <is:keyword></is:keyword>
  <is:name>volume</is:name>
  <is:keyword> </is:keyword>
  <is:name>volume</is:name>
- <is:piece-set>
  <is:piece id="7417"></is:piece>
  <is:piece id="4437"></is:piece>
  <is:piece id="1965"></is:piece>
  <is:piece id="6621"></is:piece>
  <is:piece id="3093"></is:piece>
  <is:piece id="499"></is:piece>
  <is:piece id="1667"></is:piece>
  <is:piece id="3235"></is:piece>
</is:piece-set>
<is:keyword> </is:keyword>
<is:name>volume</is:name>
</is:segment>
```



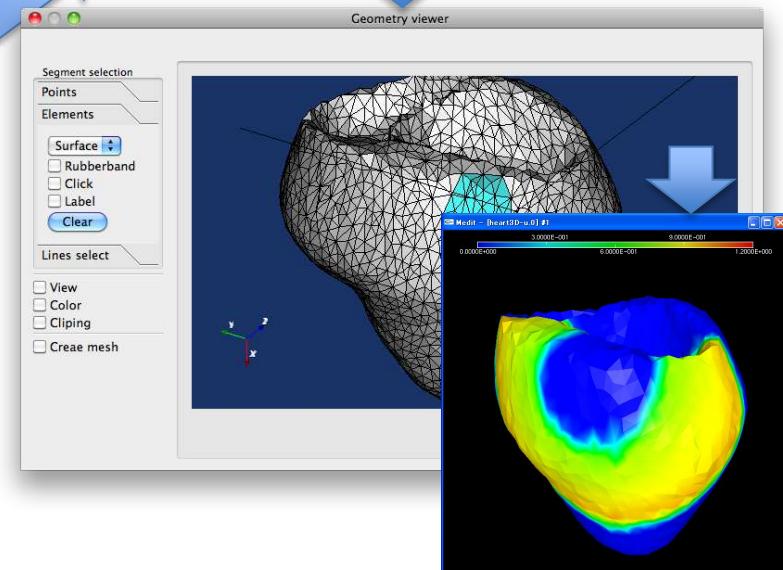
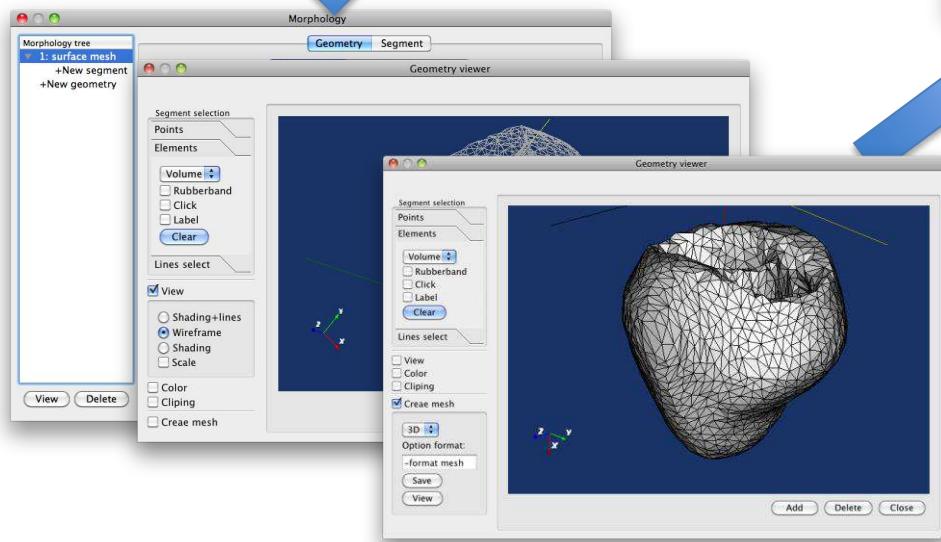
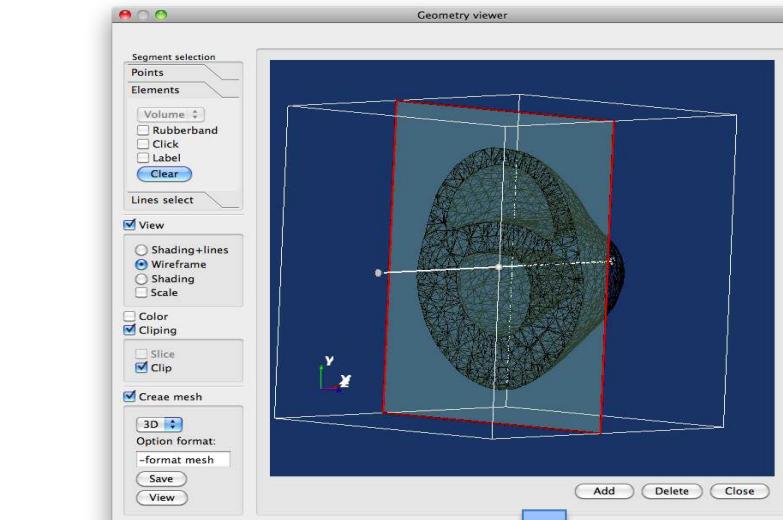
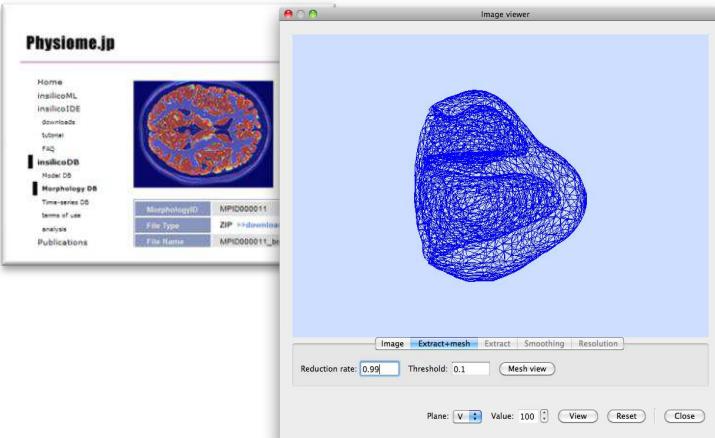
# Boundary condition point/surface

- If number of points and elements are small,  
boundary surface → collection of label → segment  
number 1  
→ `on(1,u=1.0)`
- If number of points and elements are large,  
boundary surface → collection of label → convert  
to mesh file  
`Th1 = readmesh3("boundary.msh");  
int3d(Th1)(1.0*w)`

# ImageViewer to Morphology editor

From raw image in MorphologyDB,  
make a surface model

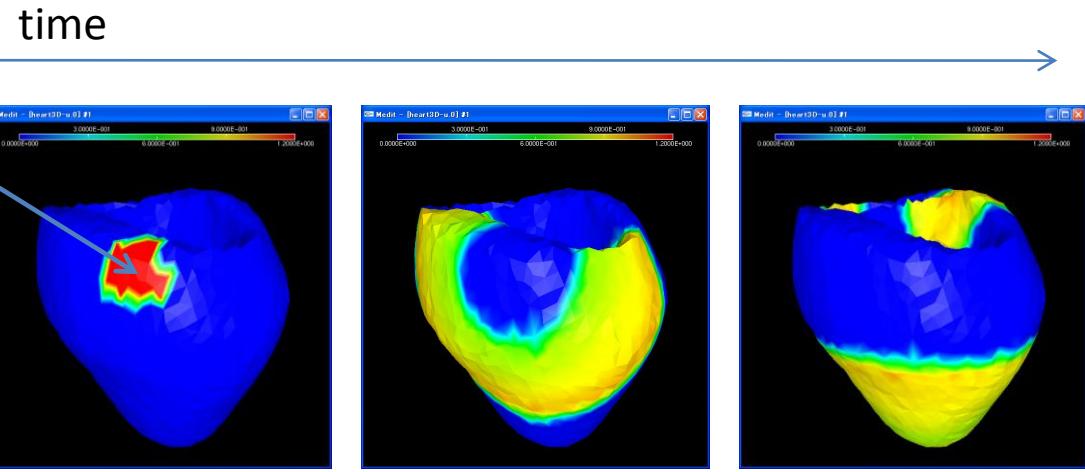
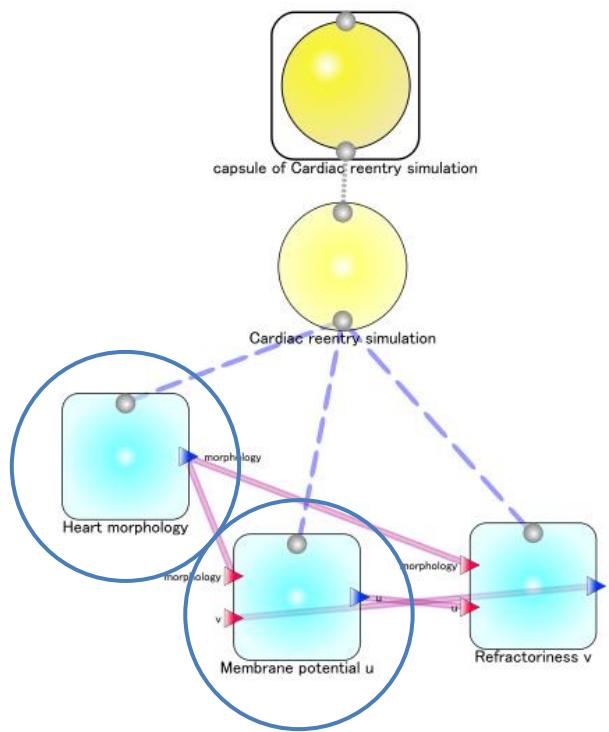
Morphology editor: volume mesh generation



# Example: Cardiac reentry simulation

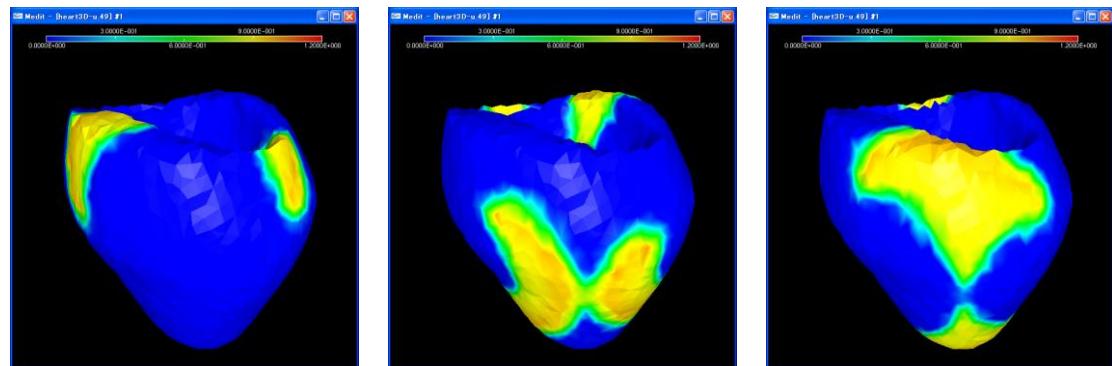
Initial condition : t=0 initial excitation position → assign segments

- Morphology editor
- Segment definition
- Physical Quantity
- Problem condition
- Set Initial condition

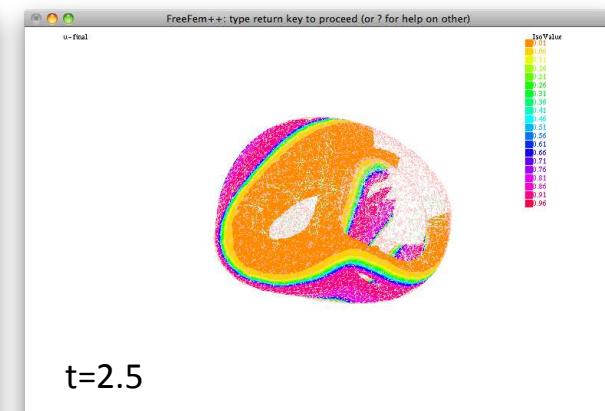
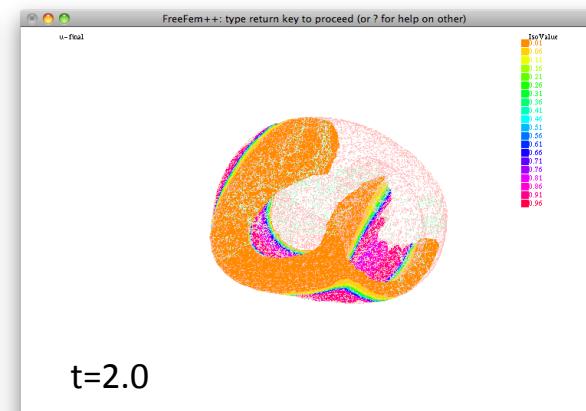
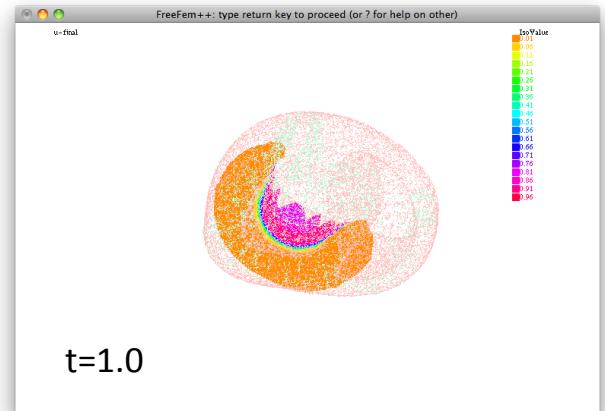
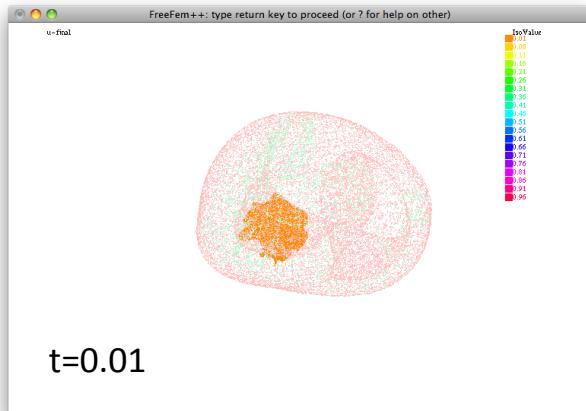


First stimulus

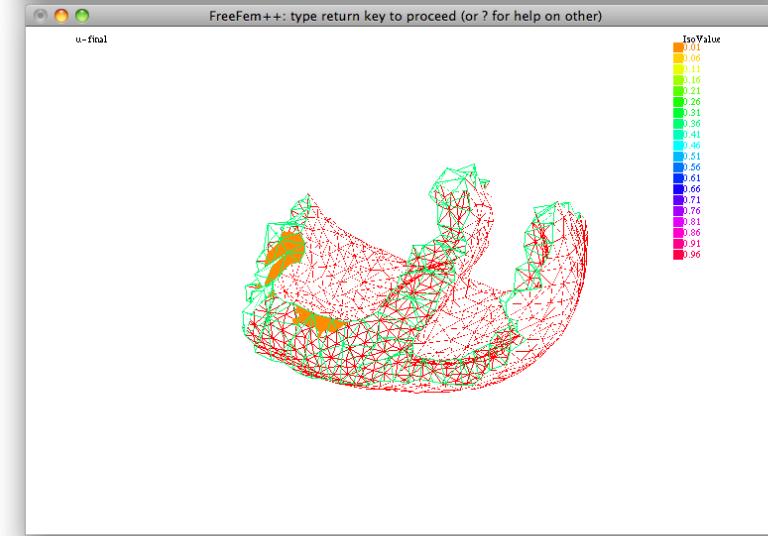
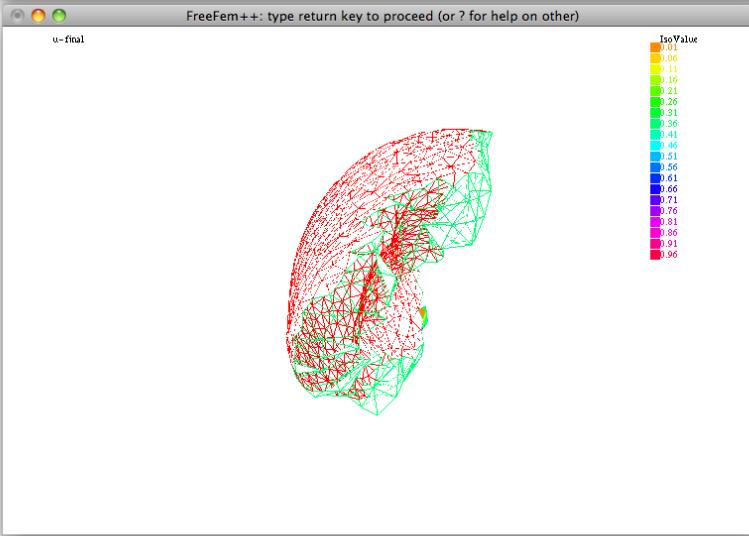
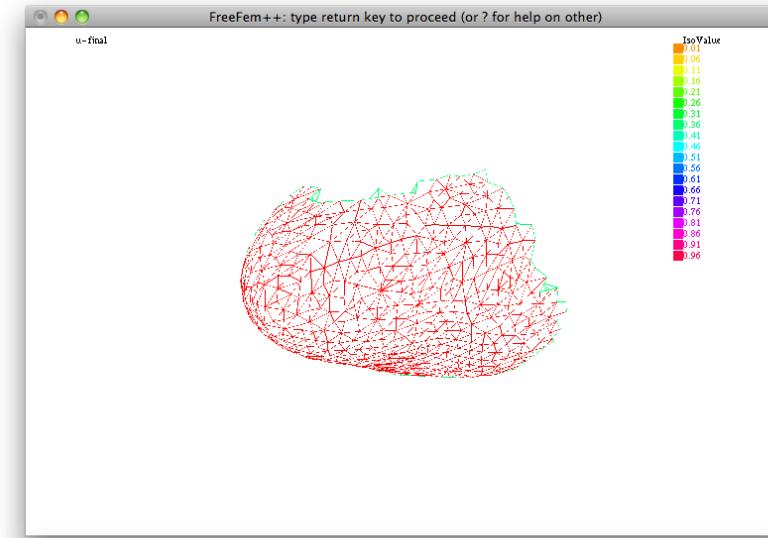
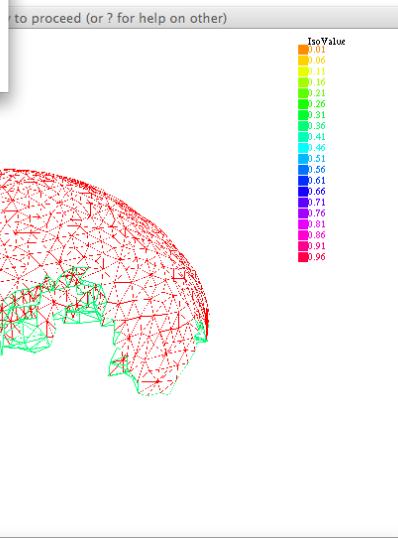
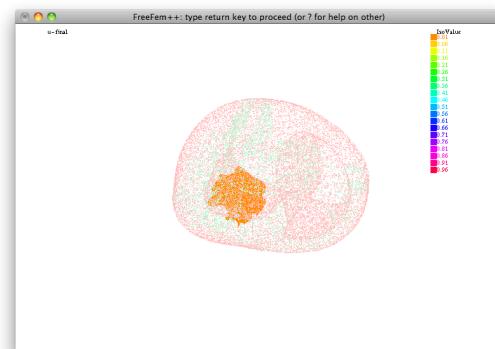
Second stimulus



# Excitation propagation (4 cores)



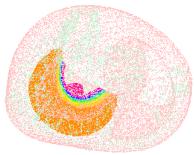
# $t=0.01$ (4cores)



FreeFem++: type return key to proceed (or ? for help on other)

IsoValue

- 0.01
- 0.06
- 0.11
- 0.16
- 0.21
- 0.26
- 0.31
- 0.36
- 0.41
- 0.46
- 0.51
- 0.56
- 0.61
- 0.66
- 0.71
- 0.76
- 0.81
- 0.86
- 0.91
- 0.96

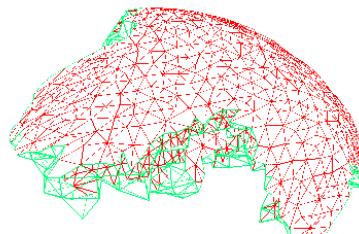


# $t=0.5$ (4cores)

proceed (or ? for help on other)

IsoValue

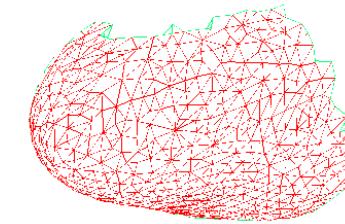
- 0.01
- 0.06
- 0.11
- 0.16
- 0.21
- 0.26
- 0.31
- 0.36
- 0.41
- 0.46
- 0.51
- 0.56
- 0.61
- 0.66
- 0.71
- 0.76
- 0.81
- 0.86
- 0.91
- 0.96



FreeFem++: type return key to proceed (or ? for help on other)

IsoValue

- 0.01
- 0.06
- 0.11
- 0.16
- 0.21
- 0.26
- 0.31
- 0.36
- 0.41
- 0.46
- 0.51
- 0.56
- 0.61
- 0.66
- 0.71
- 0.76
- 0.81
- 0.86
- 0.91
- 0.96

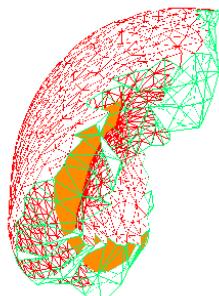


FreeFem++: type return key to proceed (or ? for help on other)

u-final

IsoValue

- 0.01
- 0.06
- 0.11
- 0.16
- 0.21
- 0.26
- 0.31
- 0.36
- 0.41
- 0.46
- 0.51
- 0.56
- 0.61
- 0.66
- 0.71
- 0.76
- 0.81
- 0.86
- 0.91
- 0.96

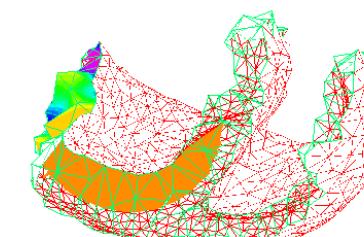


FreeFem++: type return key to proceed (or ? for help on other)

u-final

IsoValue

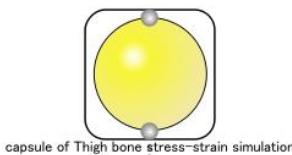
- 0.01
- 0.06
- 0.11
- 0.16
- 0.21
- 0.26
- 0.31
- 0.36
- 0.41
- 0.46
- 0.51
- 0.56
- 0.61
- 0.66
- 0.71
- 0.76
- 0.81
- 0.86
- 0.91
- 0.96



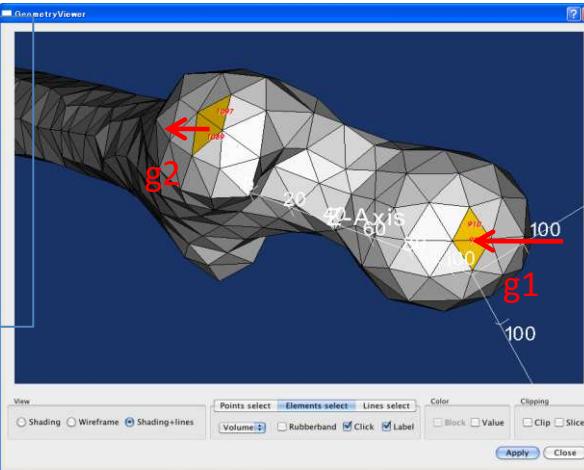
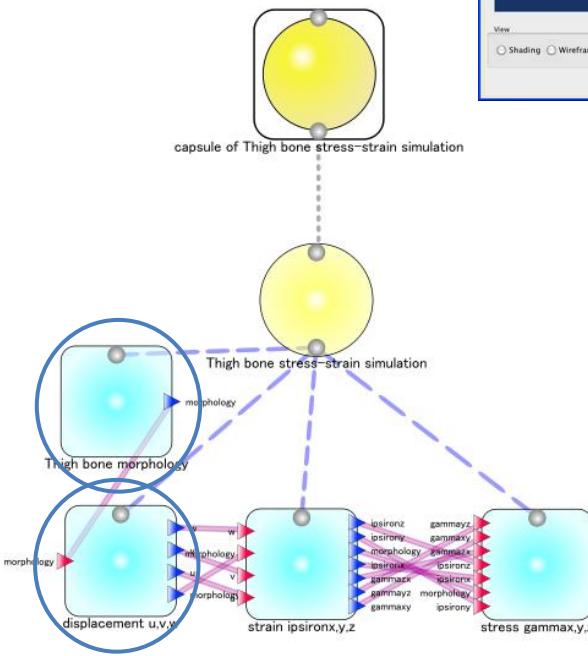
## Example: Thigh bone stress-strain simulation

Boundary condition: Apply stress to end of thigh bone (segment)

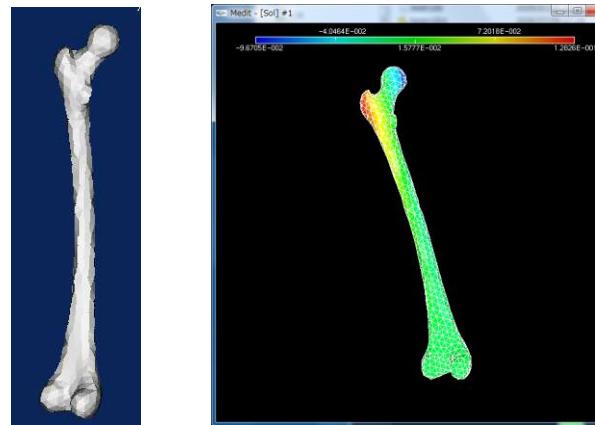
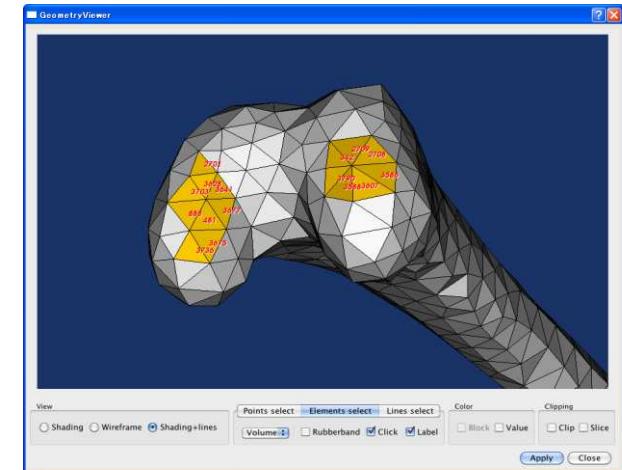
- Morphology editor
  - Assign Segment
  - Physical Quantity
  - Problem condition
  - Set Boundary condition



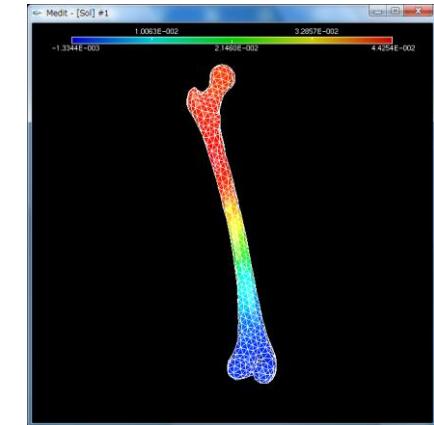
Load condition  
( $g_1 = -1500\text{N}$ ,  $g_2 = 100\text{N}$ )



## Fix (constrain) boundary



displacement  $u_7$

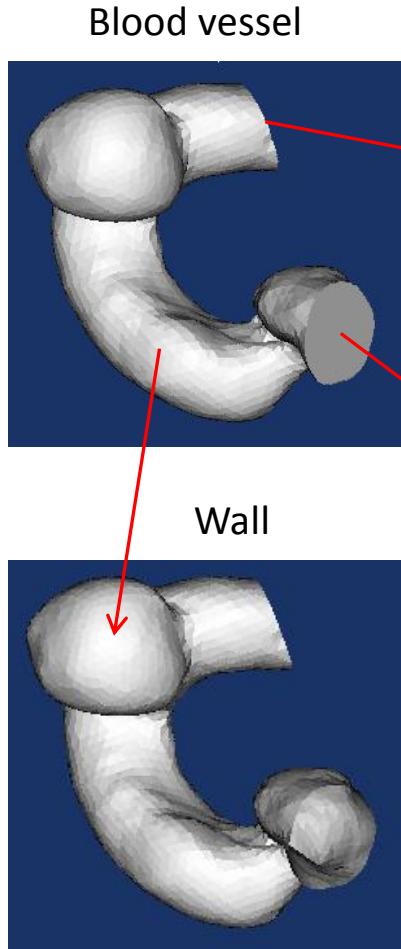
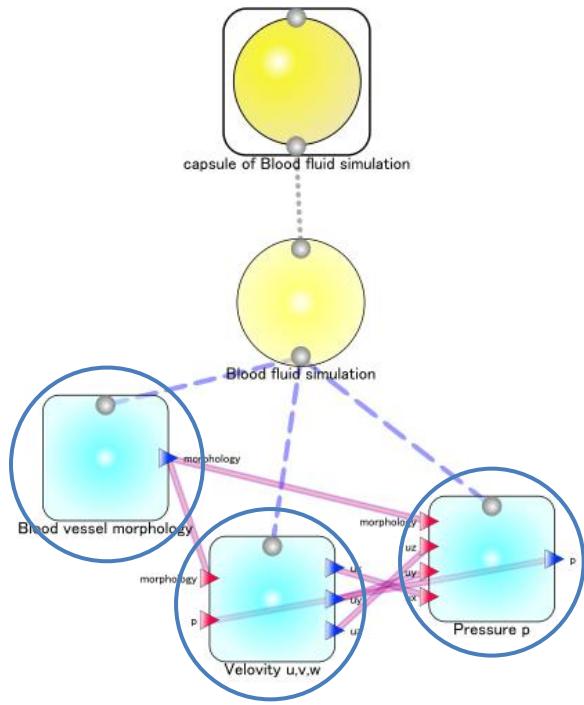


stress  $\sigma_z$

# Example: Blood vessel fluid simulation

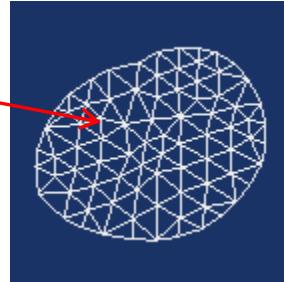
Boundary condition: Blood Flow Analysis, Boundary surface file

- Morphology editor
- Segment → vtk file
- Physical Quantity
- Problem condition
- Ste boundary condition



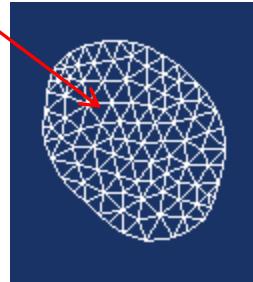
$$\mathbf{u}_N = \mathbf{0}, \nabla \mathbf{u}_T = \nabla p = 0$$

Outlet



$$\mathbf{u} = \text{convect}(\mathbf{u} - dt, \mathbf{u}_{\text{old}}) \\ P = 0$$

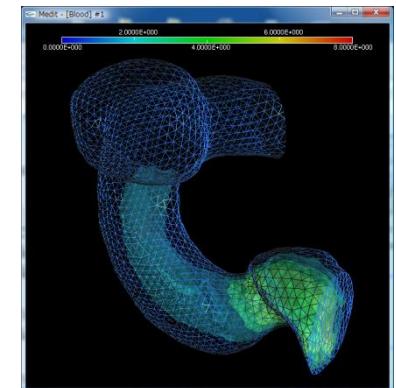
Inlet



In case of large  
Boundary condition  
surface

$$\mathbf{u} = u_0 * \mathbf{n}, \quad u_0 = 4 \\ \nabla p = 0$$

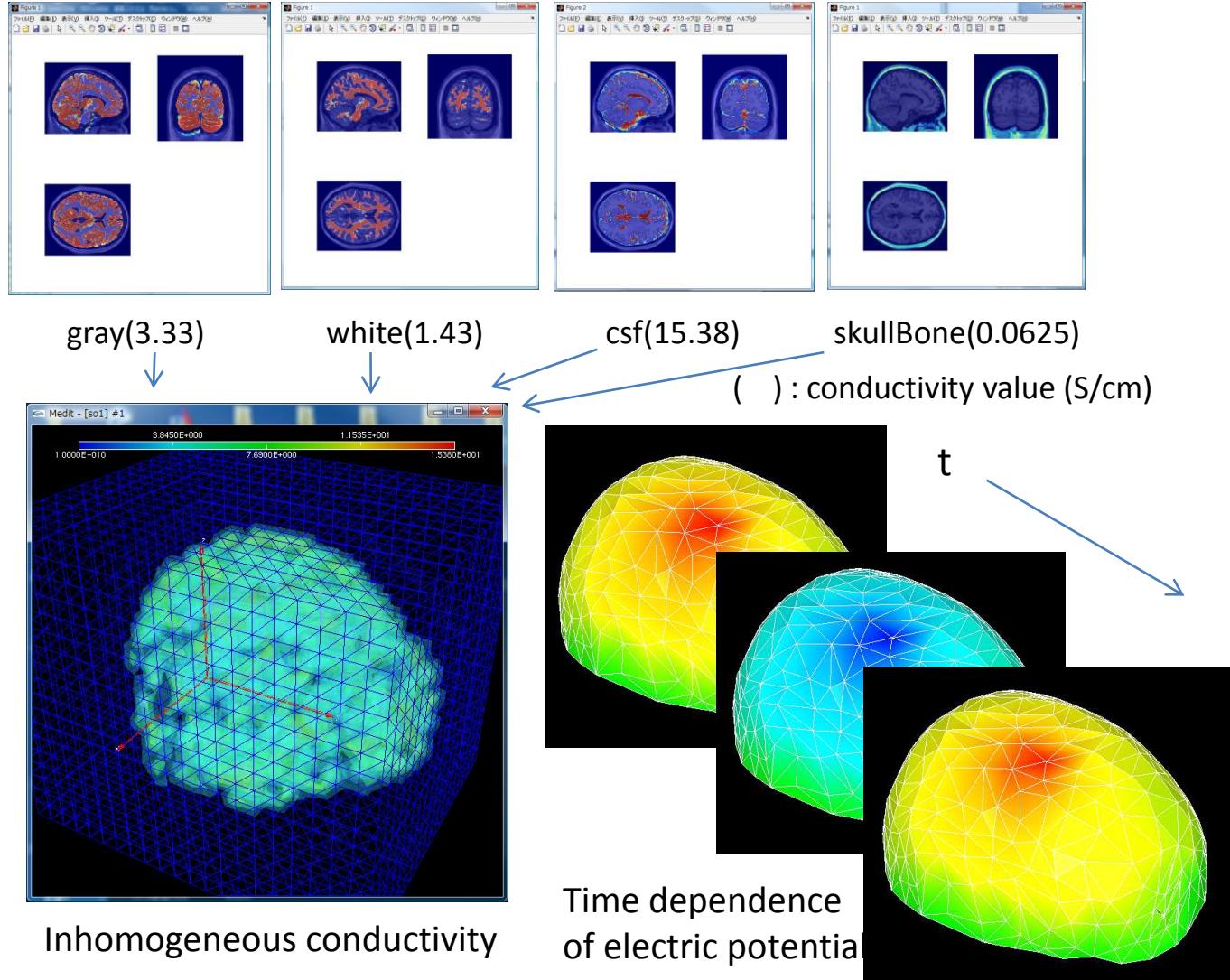
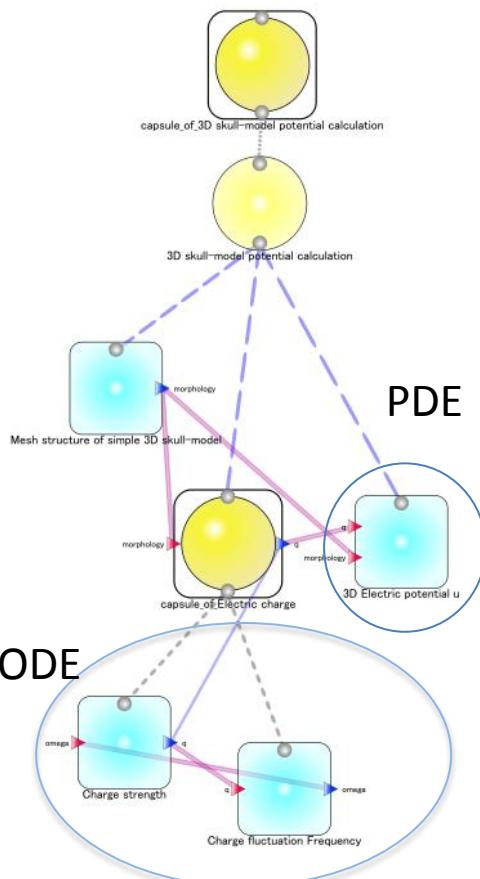
- Blood
- Newton fluid
  - Navier Stokes Equation



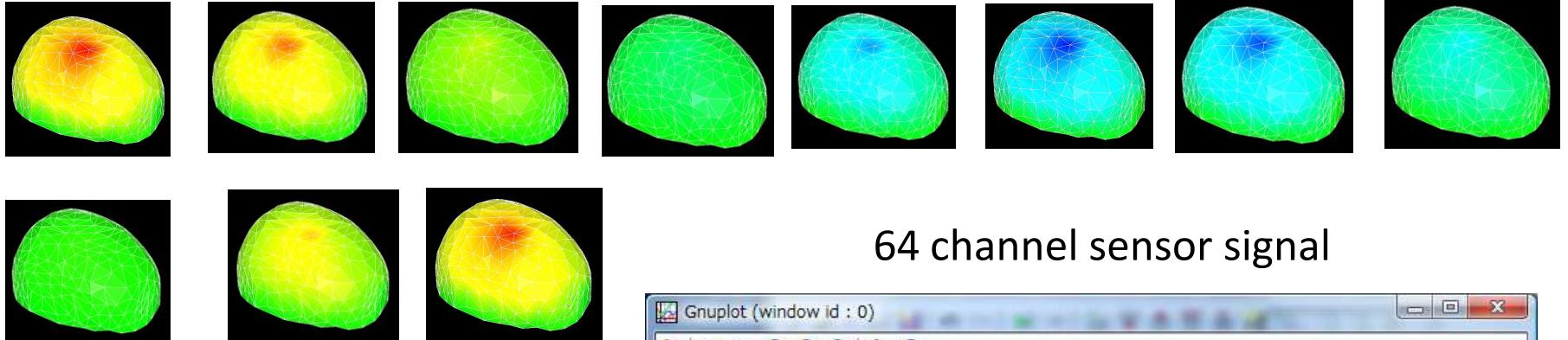
# Example: Electric field in brain (EEG)

Obtain EEG signal from brain activity sources

ImageVewer



# Time dependence of EEG signal

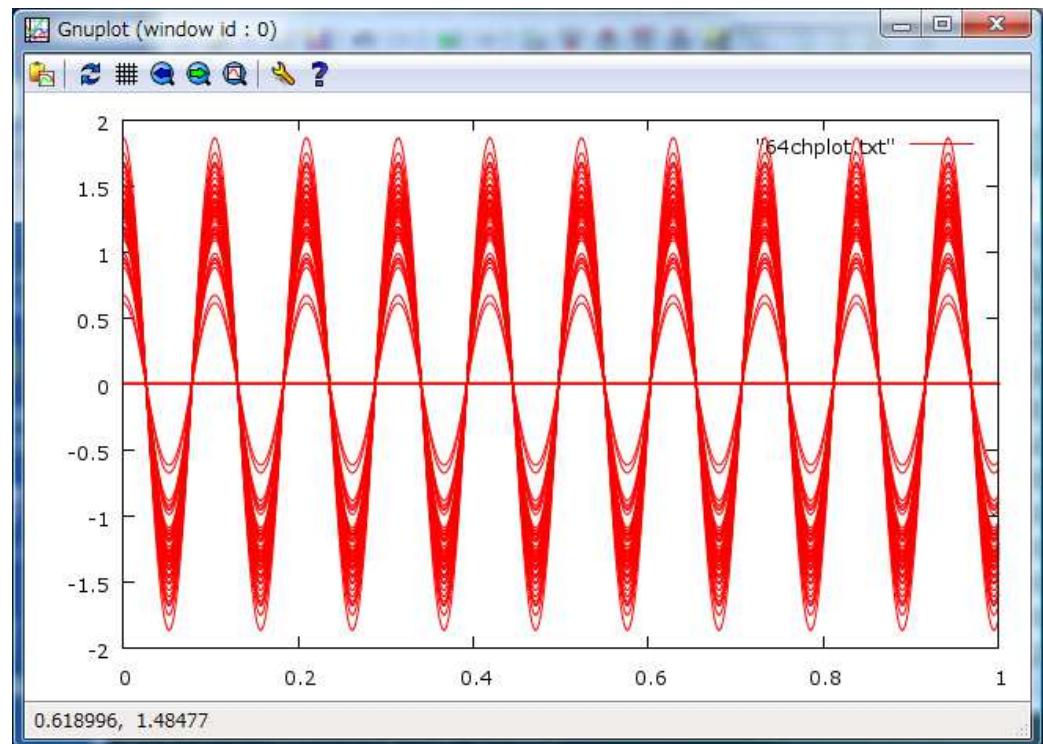


Simple Alpha  
rhythm model

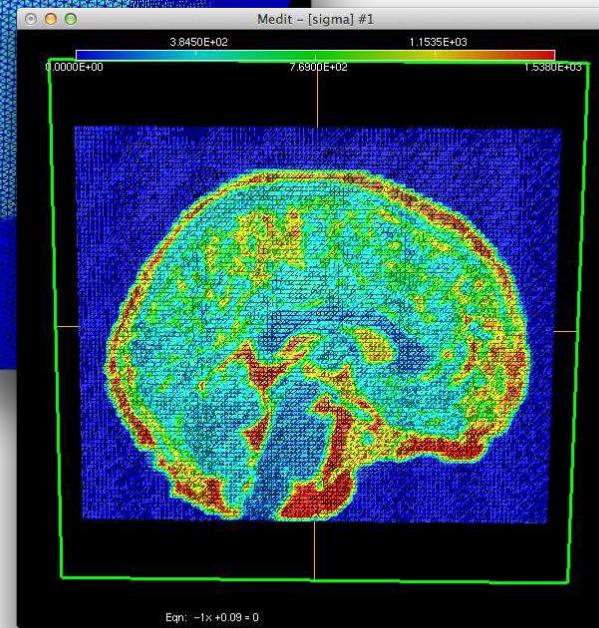
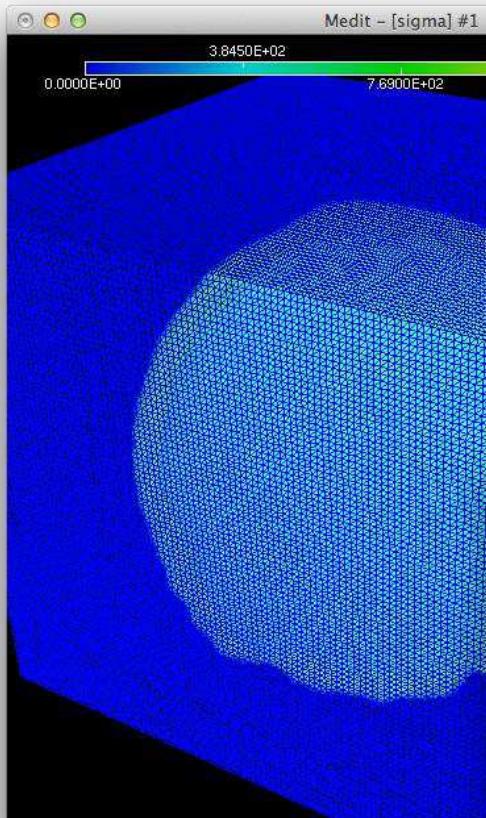
1 cycle  
0.1msec

y axis scale  
(potential)  
depends on  
strength of  
current source

64 channel sensor signal

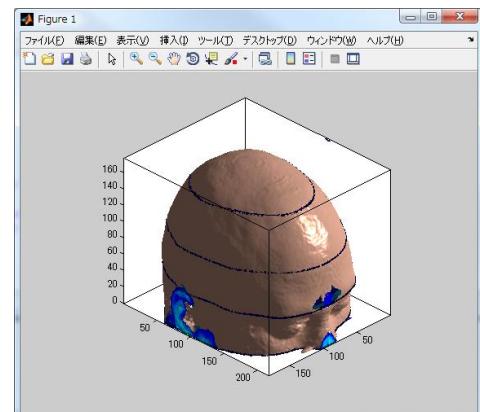
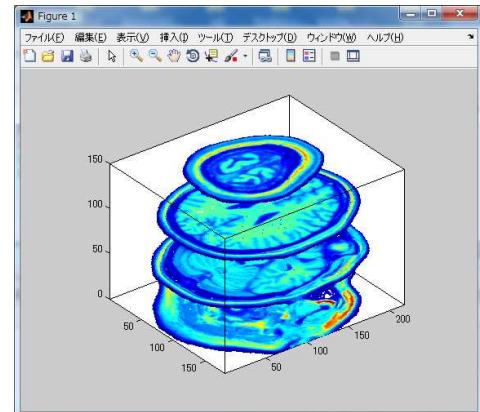


# Inhomogeneous conductivity $\sigma$

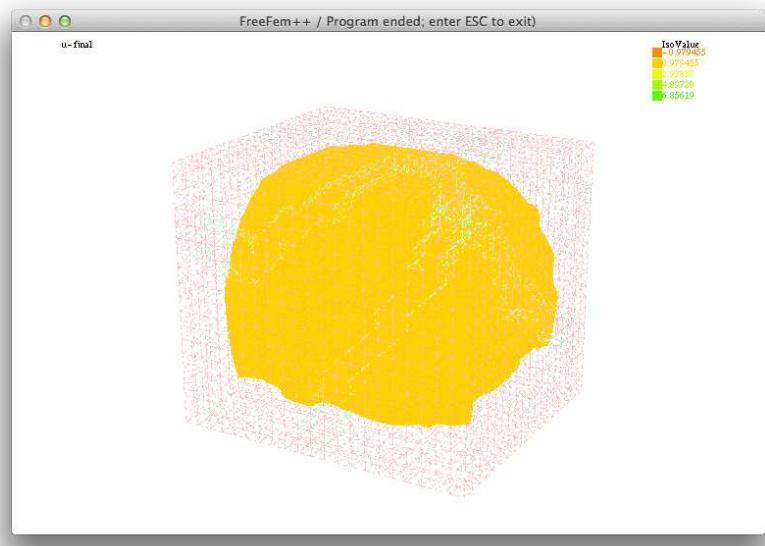


Max 1mm resolution  
= MRI image resolution

cf. MRI data for the same structure



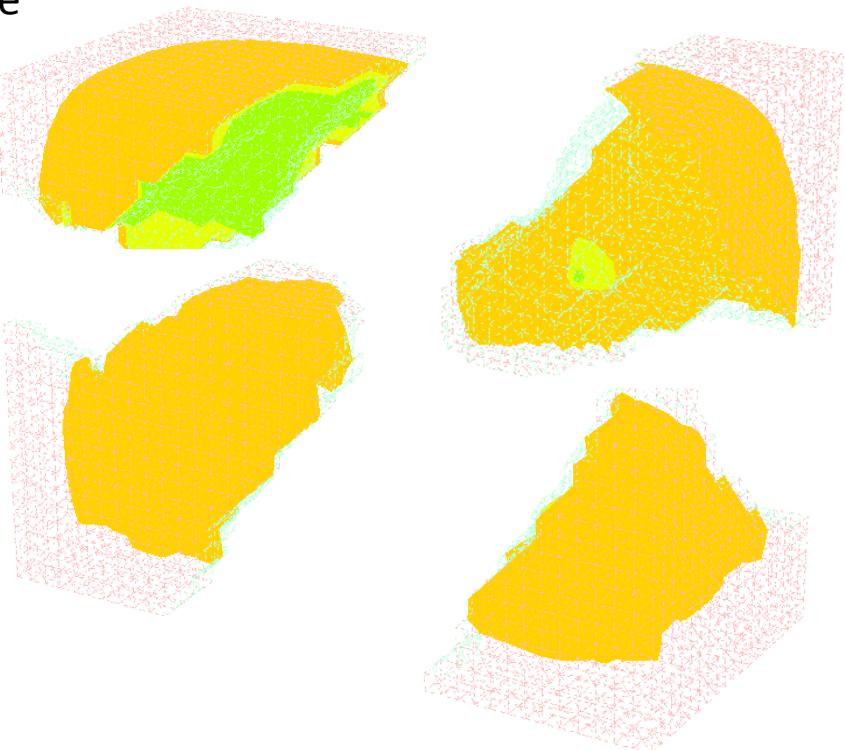
# Parallel processing for high resolution



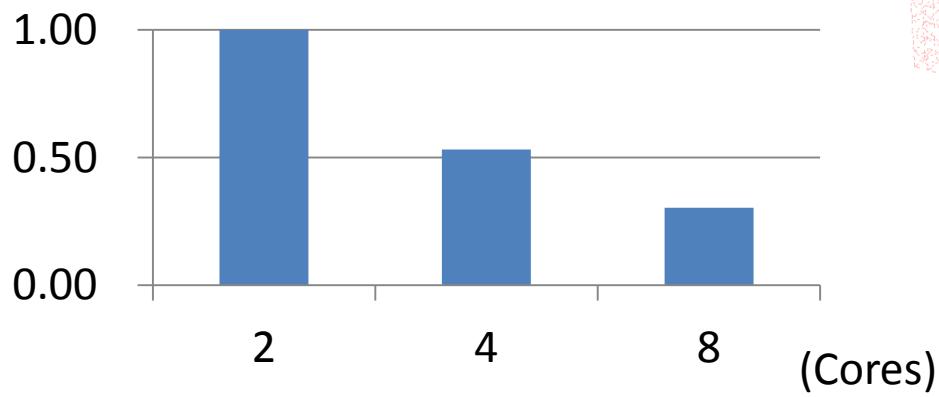
4 core  
example



Solve Poisson's equation  
with FreeFem++-mpi



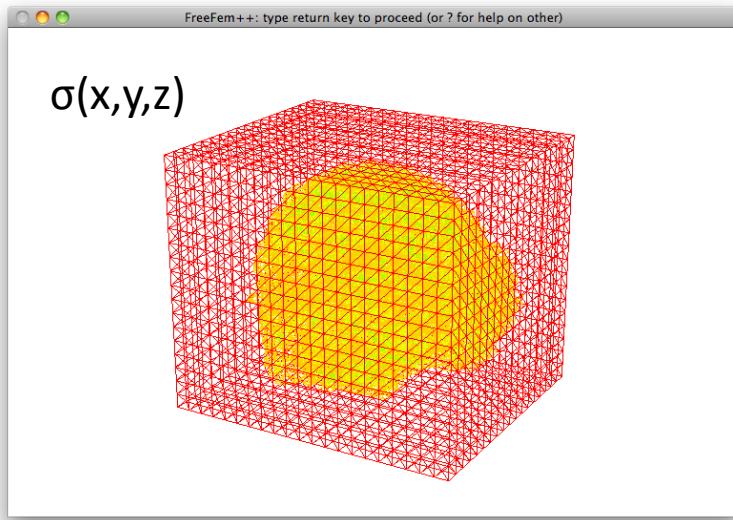
Execution time (relative)



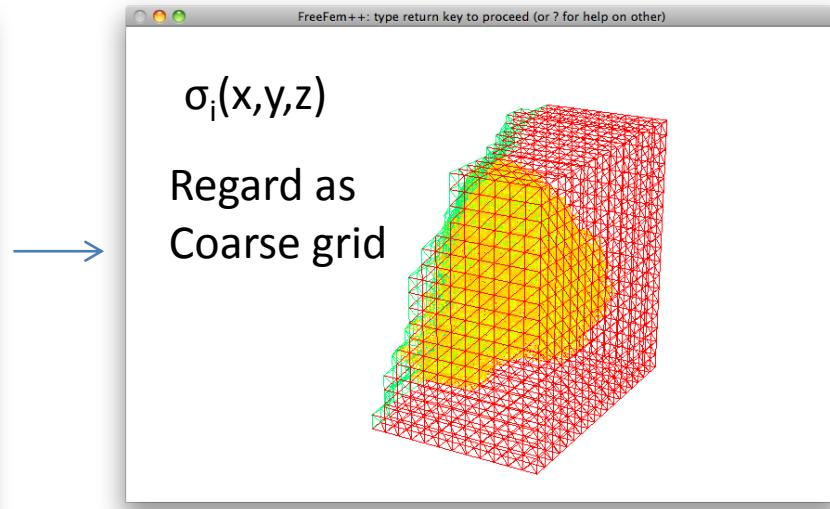
Domain Decomposition Method  
(Schwarz scheme) + MPI

# Interpolate function

- Generate Restriction、Prolongation matrix



```
mesh : Th  
Vh fespace(Th,P1)  
Vh sigma  
R = interpolate(Thi,Th)  
sigmai[] = R*sigma[]
```

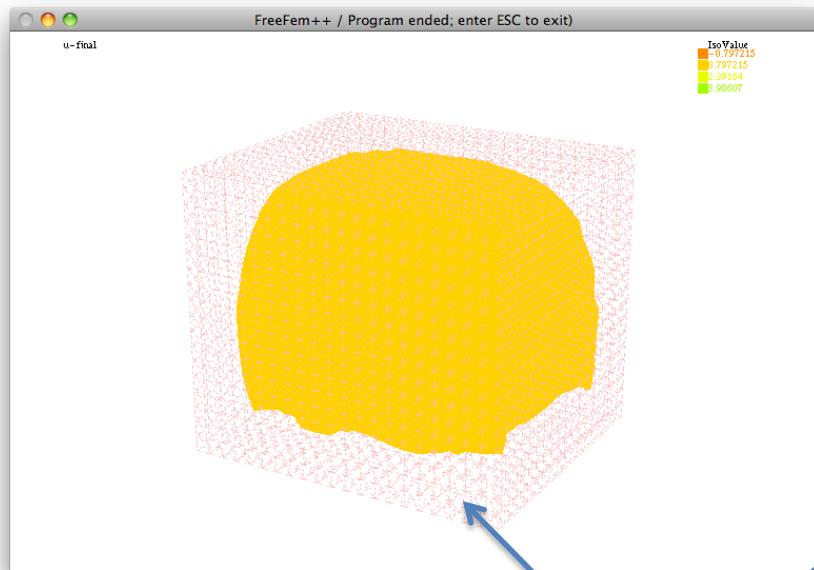


```
mesh : Thi  
Vhi fespace(Thi,P1)  
Vhi sigmai
```

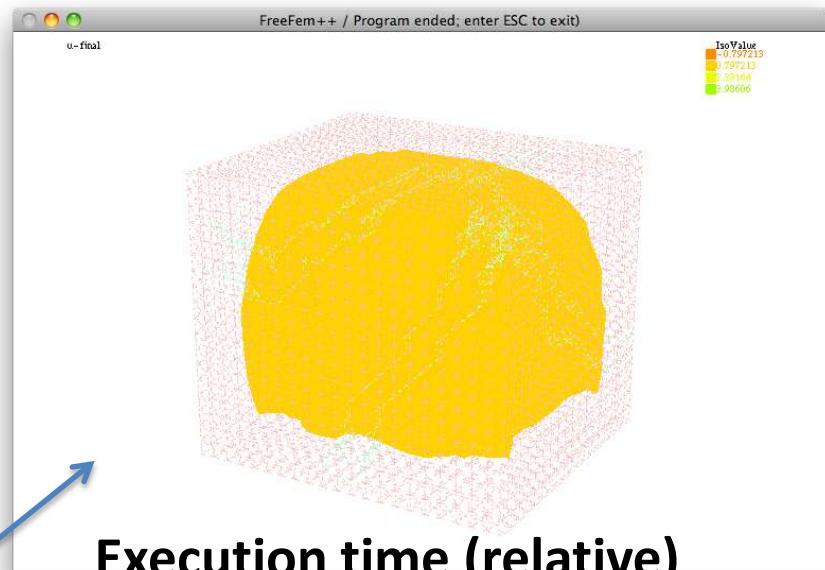
# Comparison of 1 and 4 cores results

Electric potential

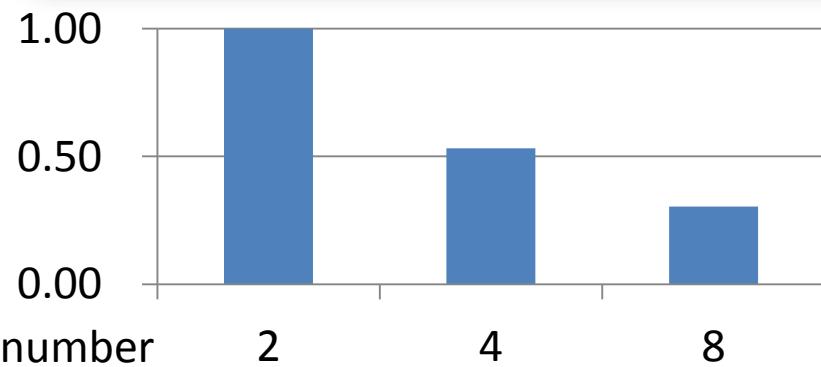
1 core



Combined 4core results



Execution time (relative)



Same result is obtained.

Linear scaling



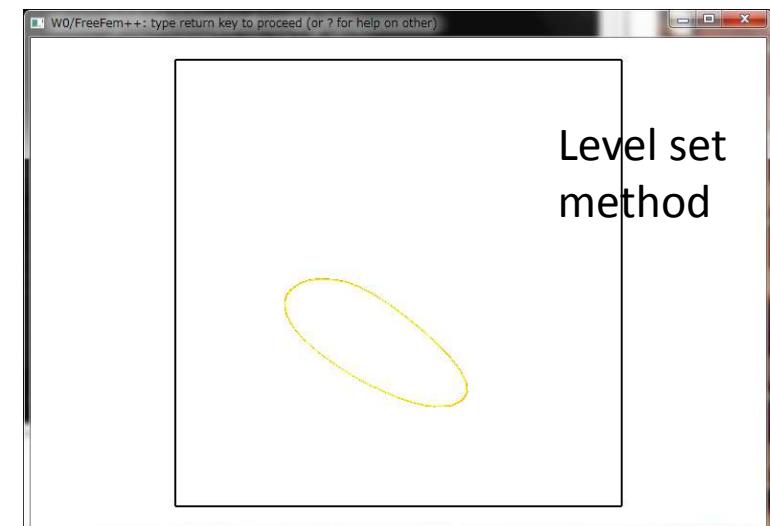
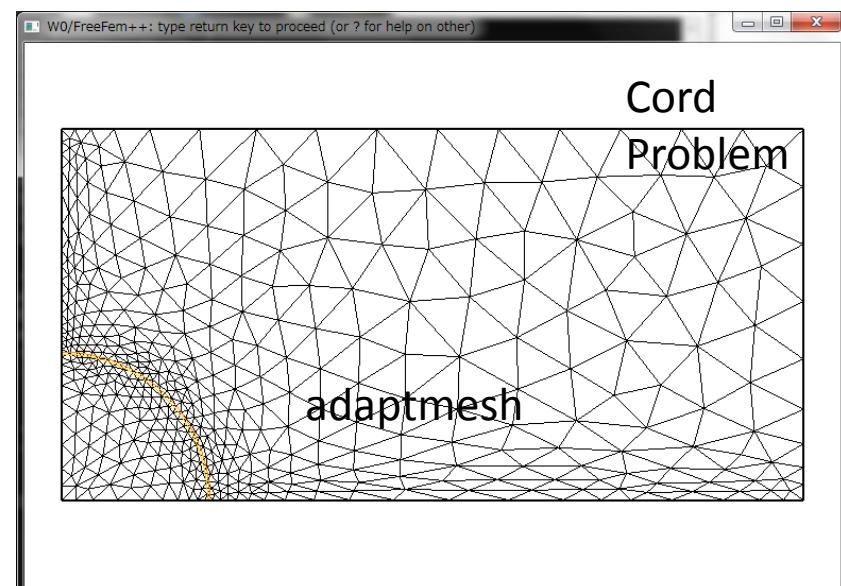
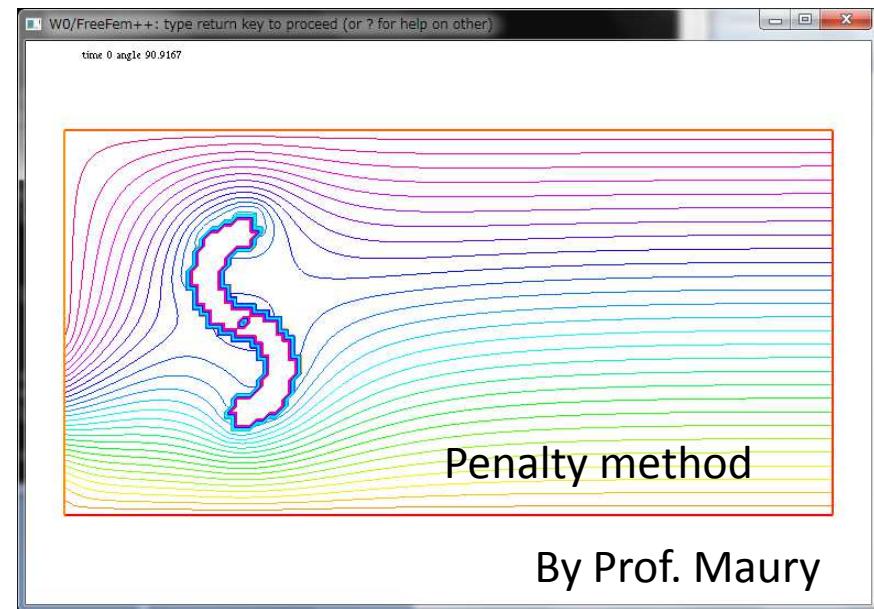
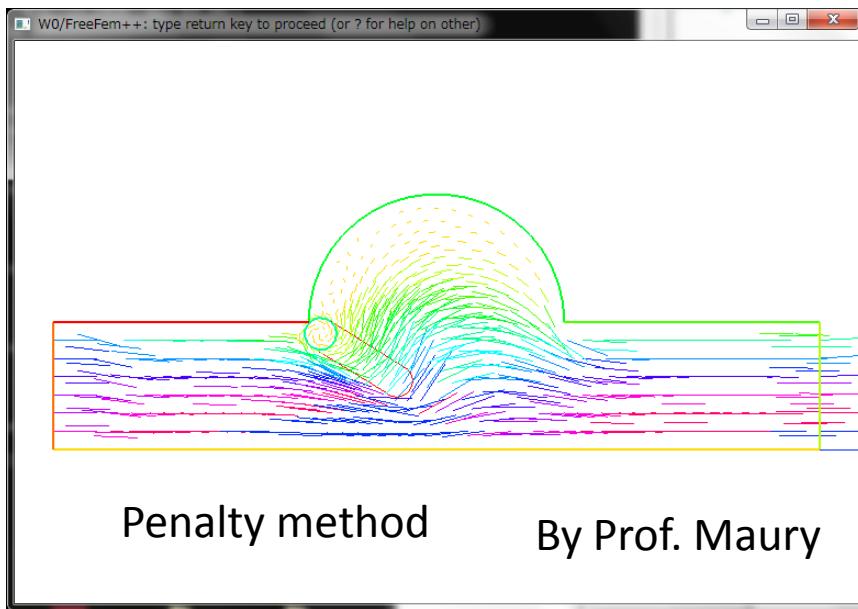
# Requirements for FreeFem++

- Boolean operation of primitives : +/- AND,OR,NOR,EXOR
- Pure 1D FEM solver (we have many ODEs + PDE)
- Full support of Matrix/Tensor operation
- Full support of LAPACK, dense matrix linear solver
- Easy inclusion of C++ code (more over GPU code) PDE+ODE etc.
- Automatic conversion from strong to weak form (freeFem3D admits a strong form)
- Easier parallel code (Current code is too complex and large overhead)
- Quad mesh and/or Mixed mesh
- More advanced examples ( ALE/IBM, BEM, up-to-date CFD solvers etc.)
- More functional and use-friendly 3D visualization
- 3D tools same as 2D ones
- Function call is very slow

# Current concerns

- Nonlinear continuum mechanics
- Fluid-Structure interaction problem
  - ALE, IBM, Level Set, Penalty, Fictitious domain
- Automatic control of Stability
  - Reaction-convection-diffusion equation
  - Navier Stokes equation
  - mesh generation (timestep vs. mesh length)
- Really applicable to real problem, speed, large scale and complex problems, guarantee of precision compared with commercial software ??

# Moving Boundary / FSI Problem FF++ examples

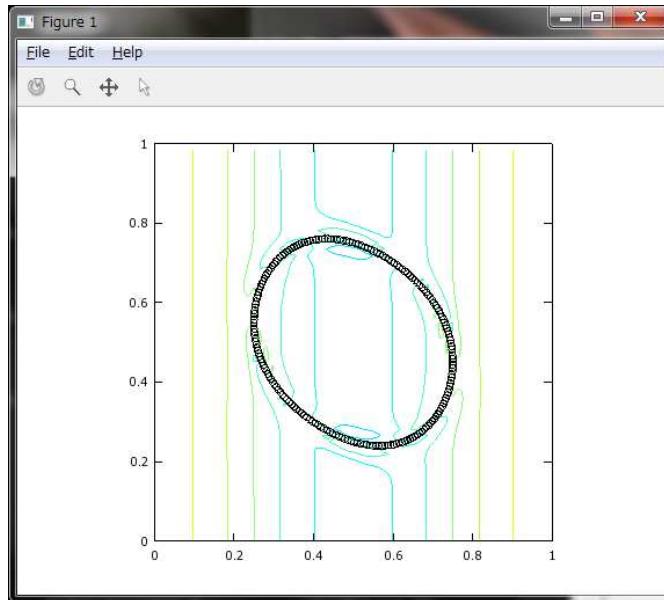


For ALE approach, movemesh is effective.

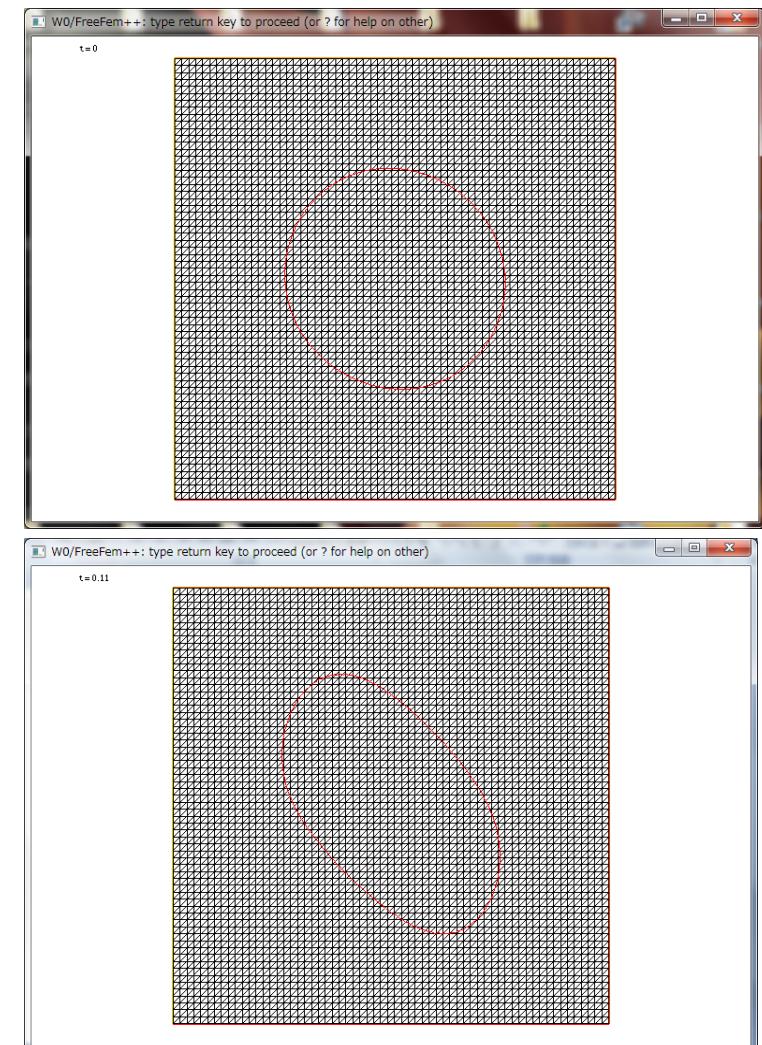
# Immersed Boundary Method

- Peskin's example

Fix mesh and trace boundary movement



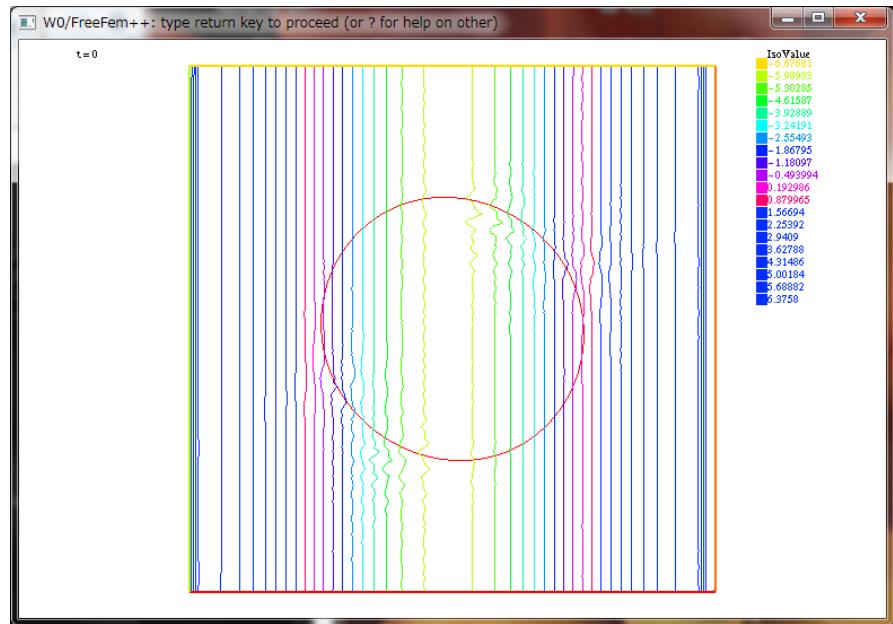
Matlab (FDM)



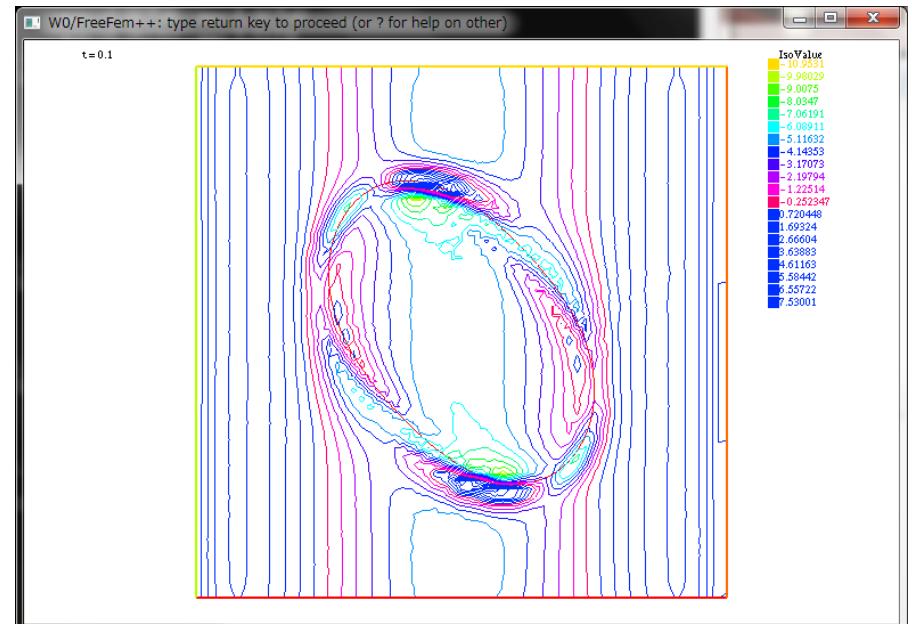
FreeFem++

# Boundary movement and vorticity

$t = 0.0$



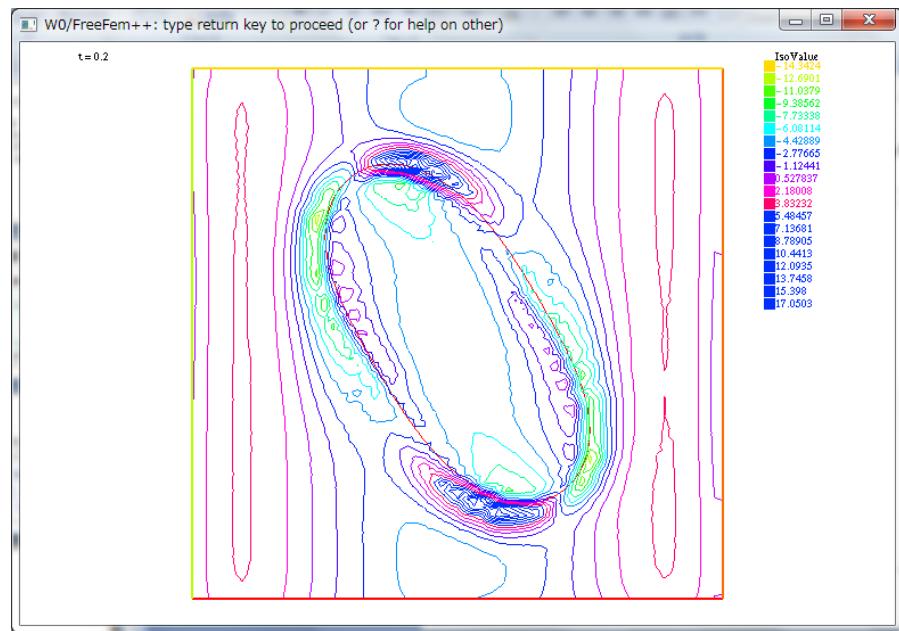
$t = 0.12$



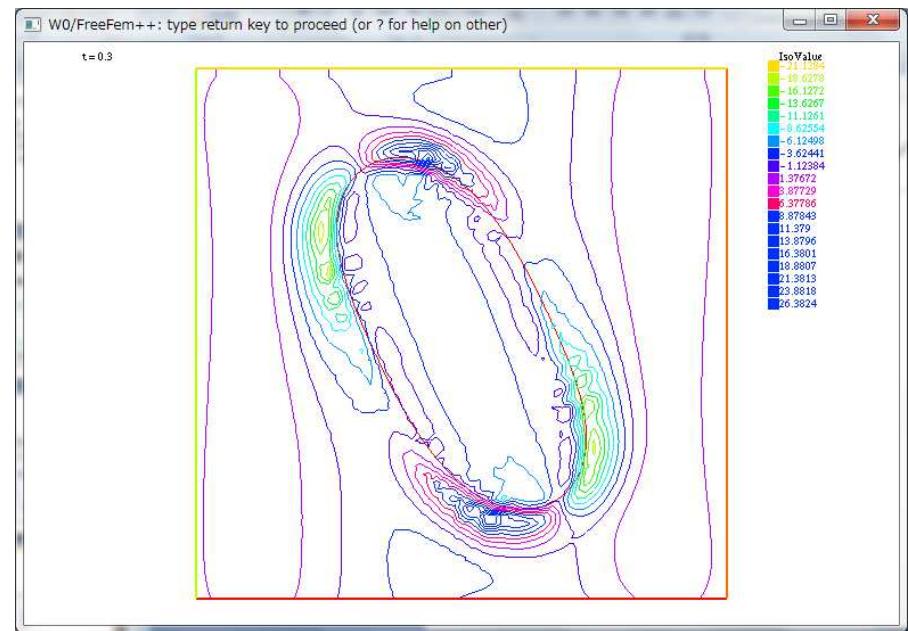
FreeFem++ implementation

# Boundary movement and vorticity

$t = 0.2$

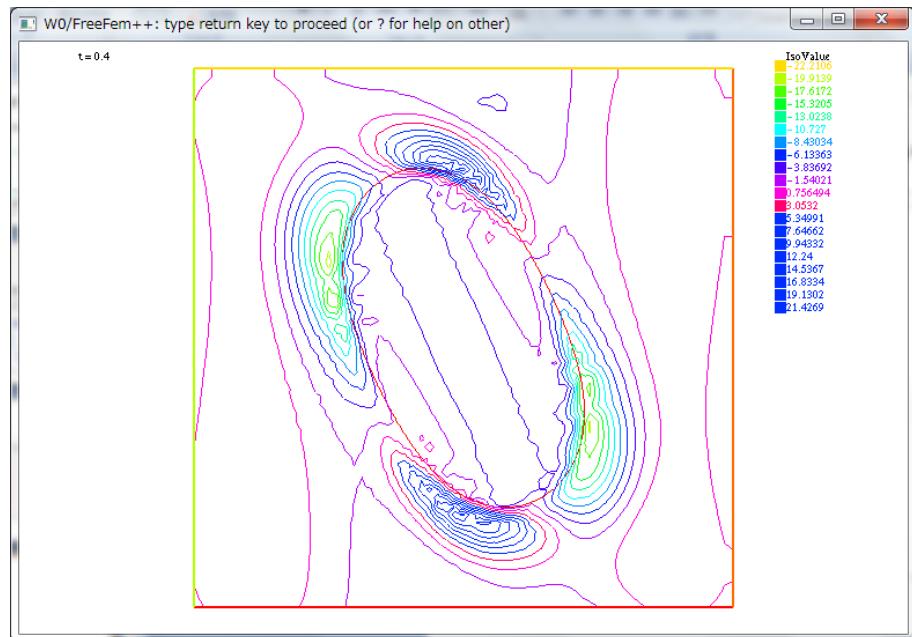


$t = 0.3$

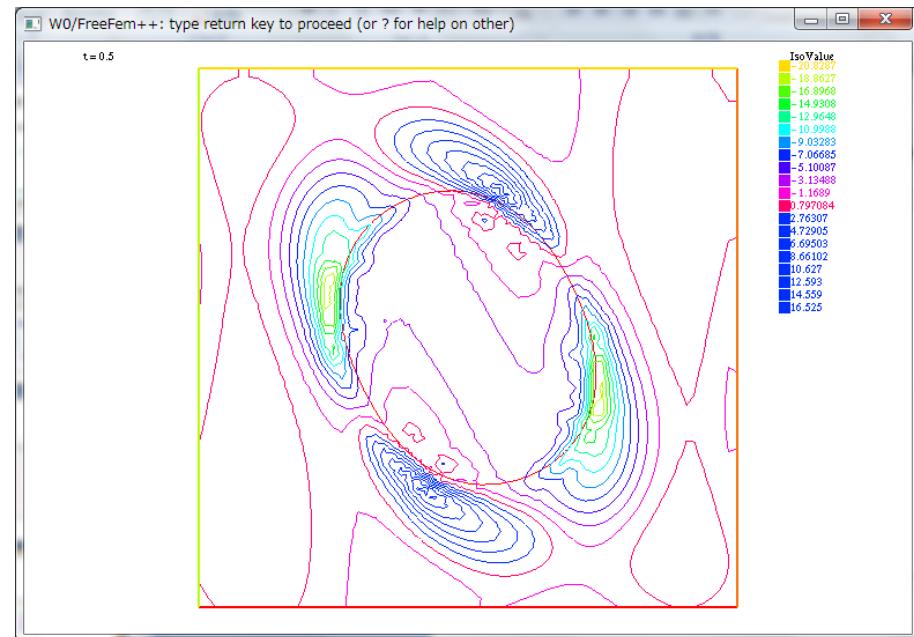


# Boundary movement and vorticity

$t = 0.4$

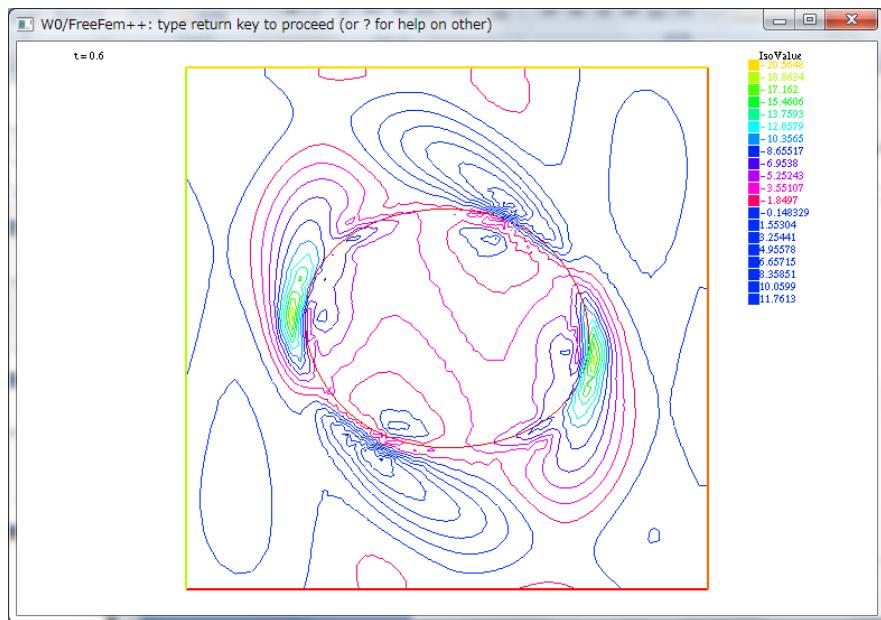


$t = 0.5$

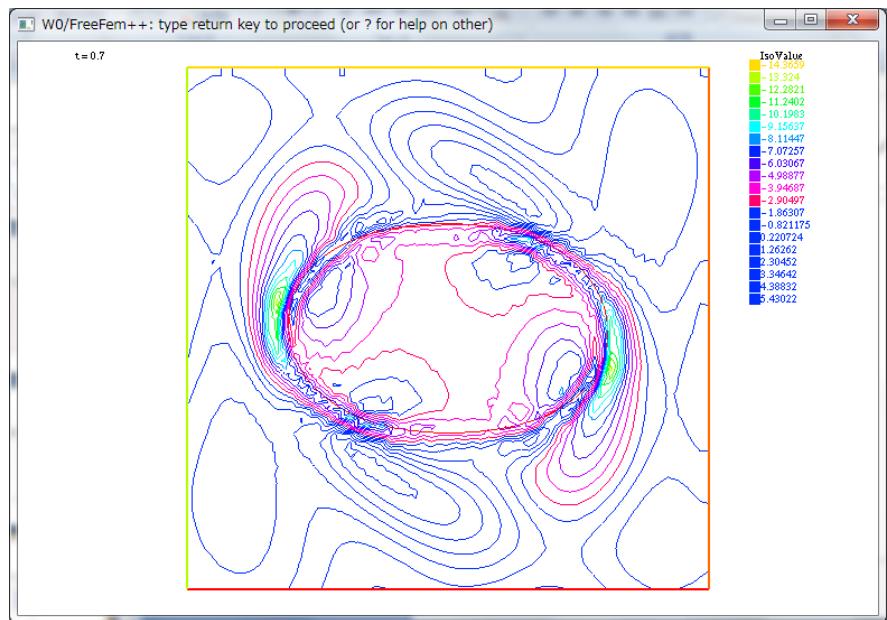


# Boundary movement and vorticity

$t = 0.6$

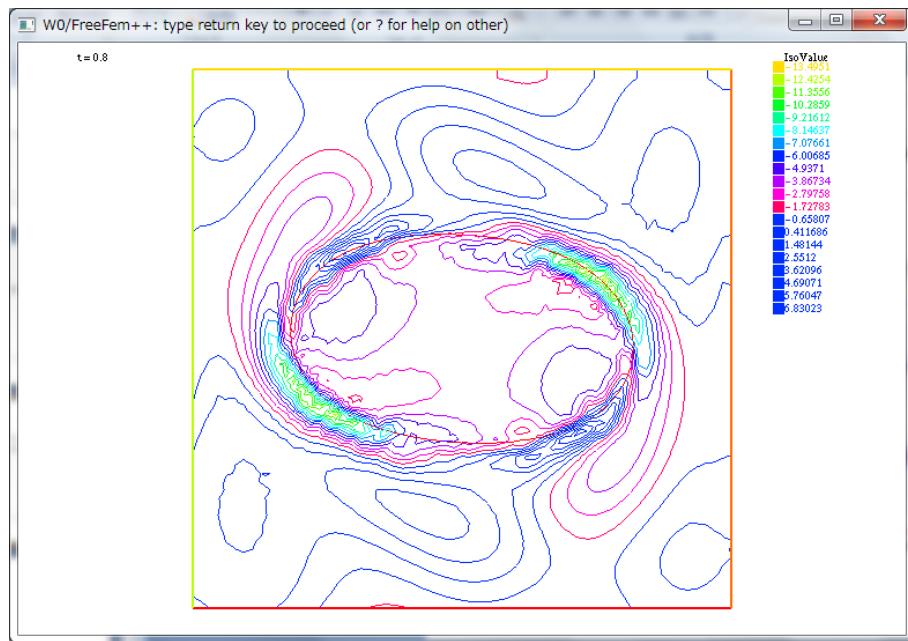


$t = 0.7$

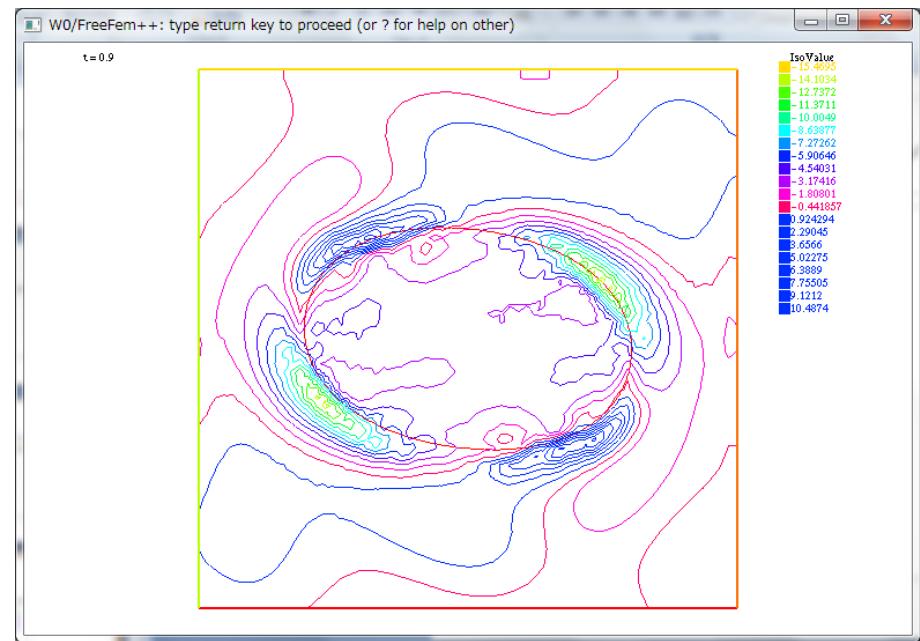


# Boundary movement and vorticity

$t = 0.8$



$t = 0.9$



# Multiscale and Multiphysics

- Any device or tool from FreeFem++ ?
- Need to communicate with other applications,  
key is different space and time scales !
- Many multiscale frameworks, but really  
helpful to multiscale simulation ?
- Multiphysics, only gathers physical problems !

# Example: Pancreatic beta-cell

## Cell Signaling and Electrophysiology of Cells coupled by ATP

Joint research with SBI

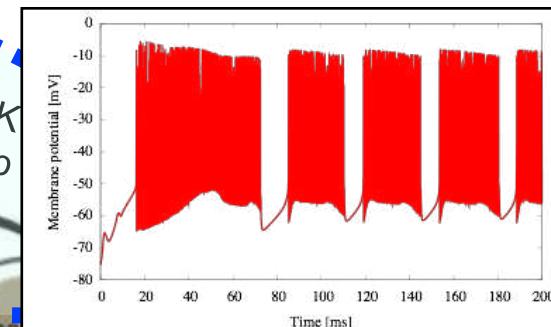
Pancreatic beta-cell model

The model which described  
the membrane potential Na  
dynamic state of the beta cell  
of the pancreas

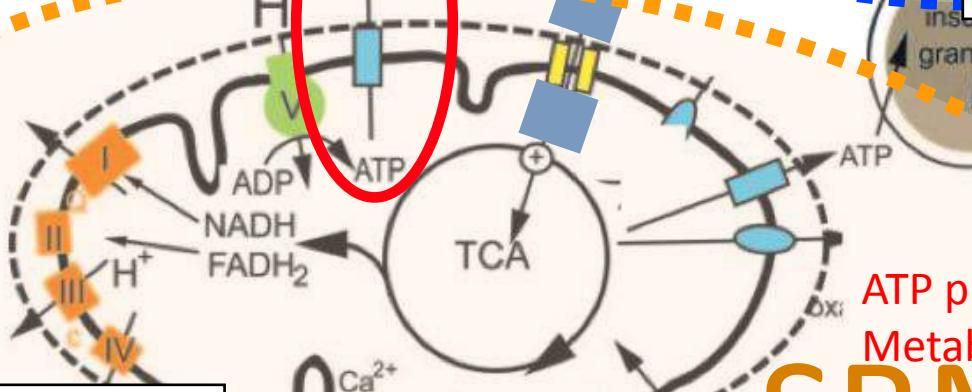
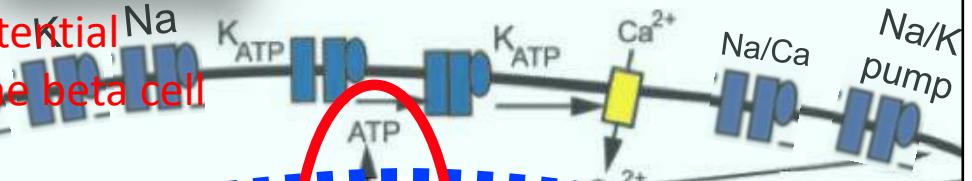
SBML+ISML mixed model construction and a simulation

Fridlyand et al. (2003) Am J Physiol

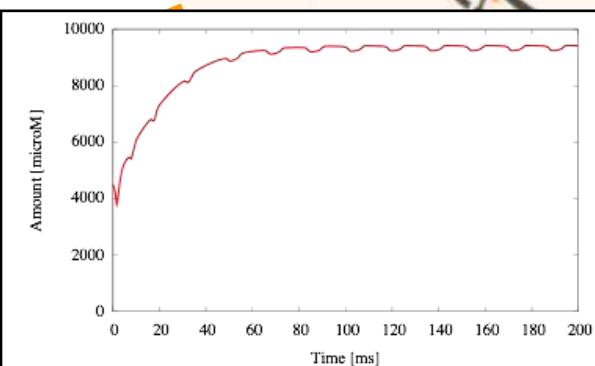
ISML



Glucose increase  
ATP production  
 $K_{ATP}$  channel close  
Ca channel open  
 $Ca^{2+}$  increase



SBML



Jiang et al. (2007) Mamm Genome

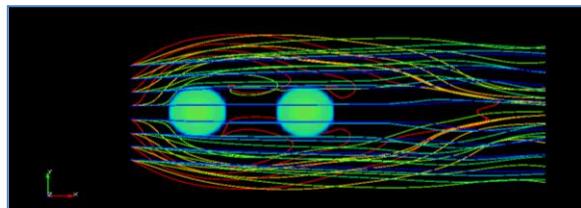
# Platelet simulation (continuum) model

Origin of Myocardial/Cerebral Infarction

Micro level

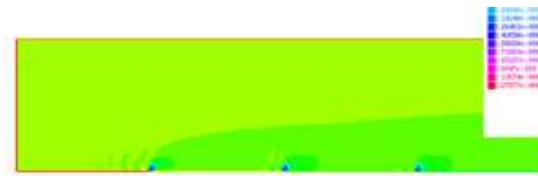
Cell simulator

RICS



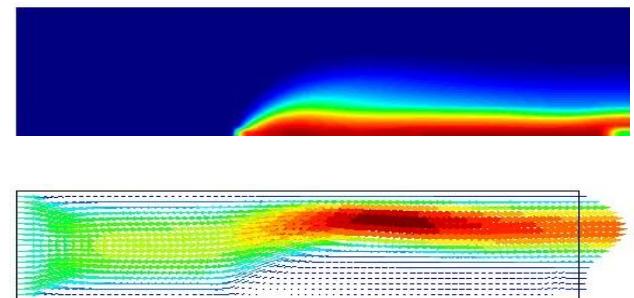
Semi Micro level

Continuum model



Macro level

Continuum model (Fogelson)

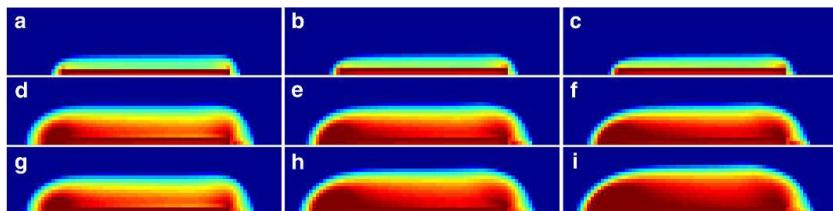


Metabolism reaction main  
Add diffusion and Fluid motion

Continuum model  
+metabolism reaction eq.

Run by FreeFem++

K. Leiderman, A L. Fogelson



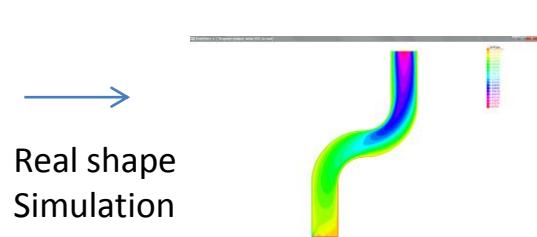
Platelet aggregates and makes a clot at injured position



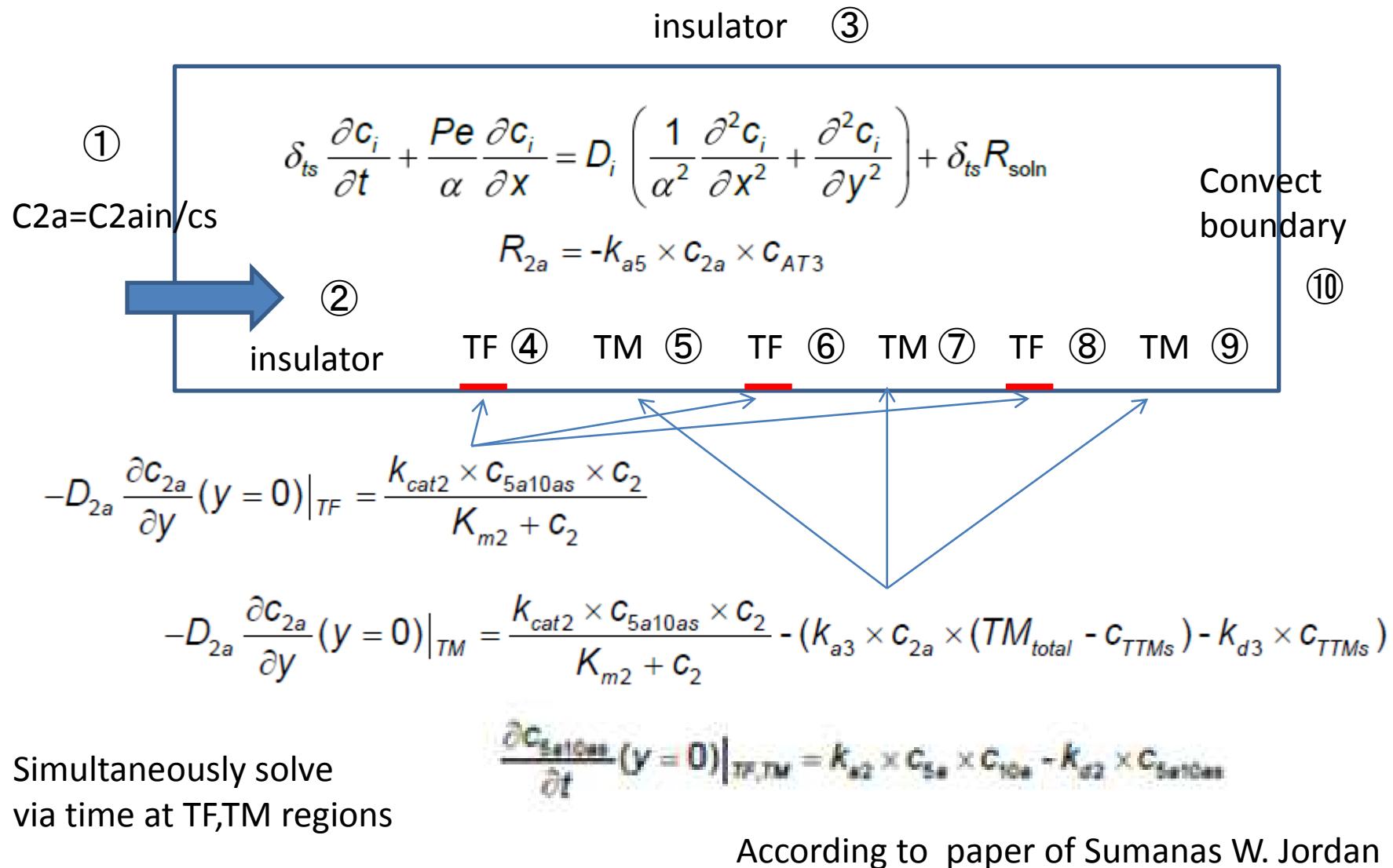
Growth simulation  
IBM approach

Contiuum model main  
+minimum meta. reaction eq.  
Wall interaction included

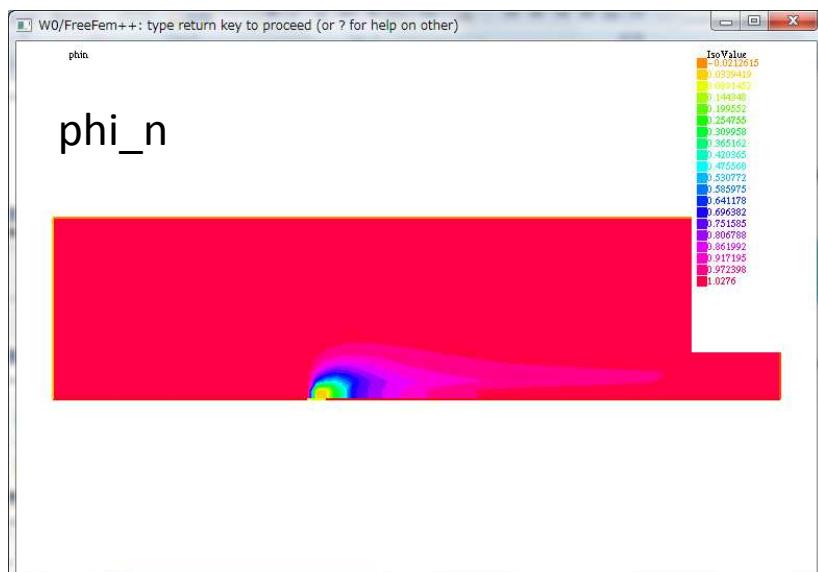
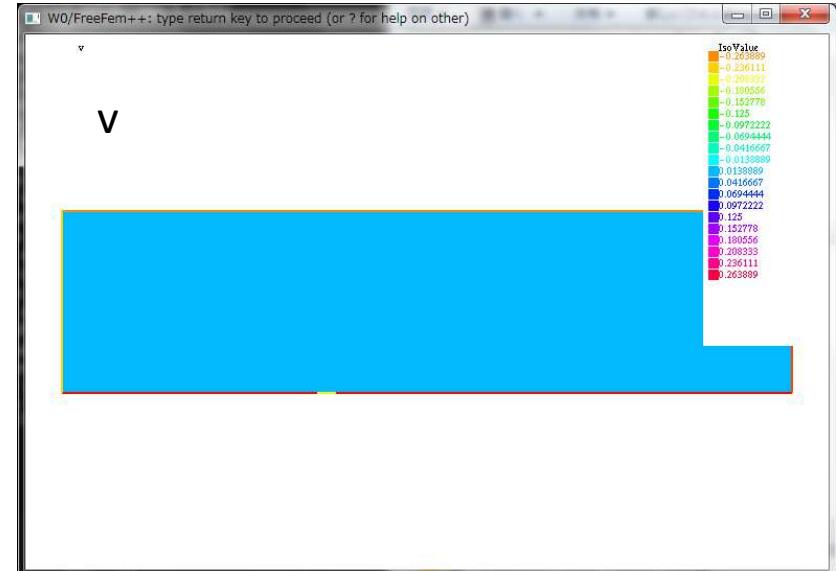
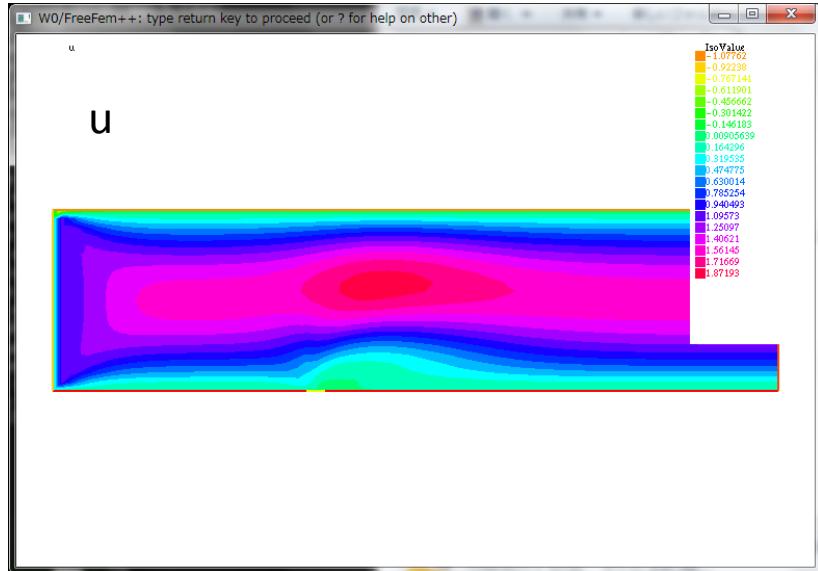
Convert to FreeFem++



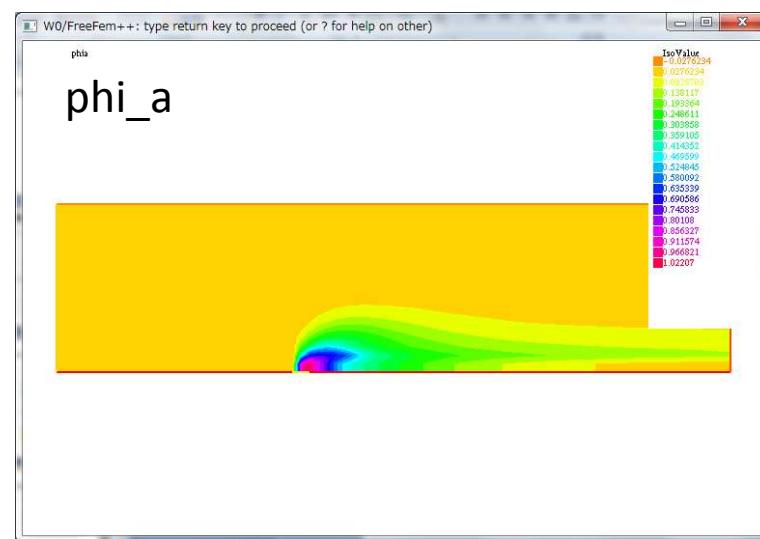
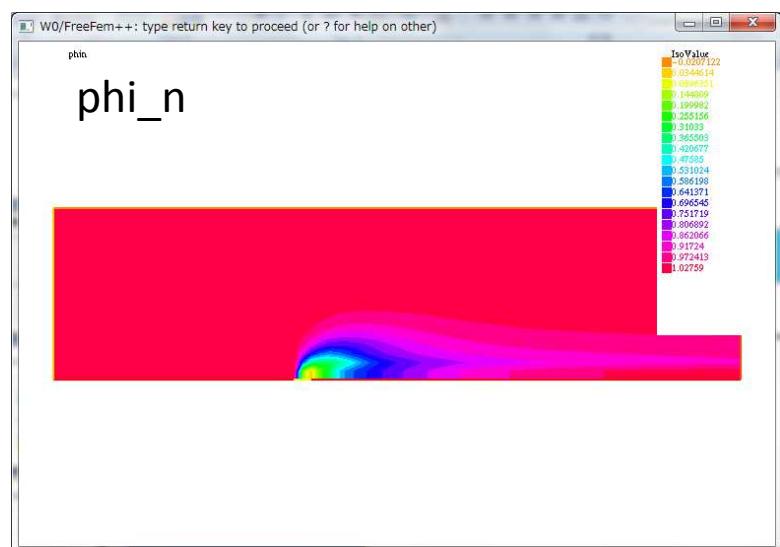
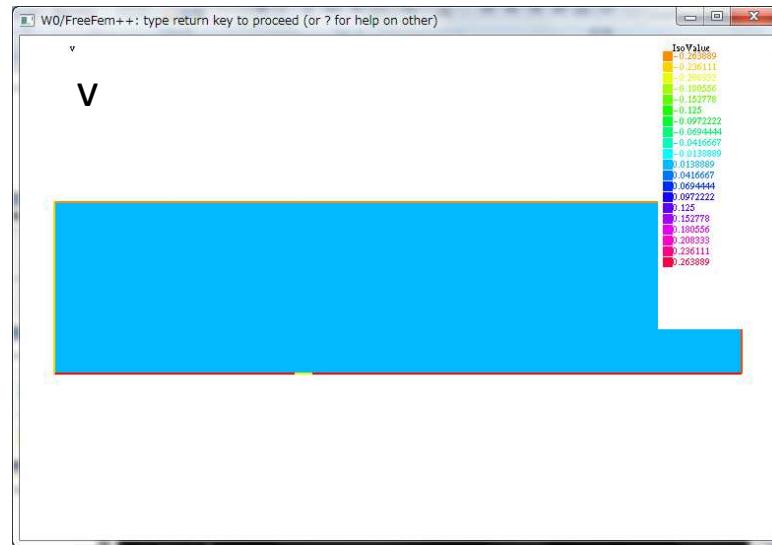
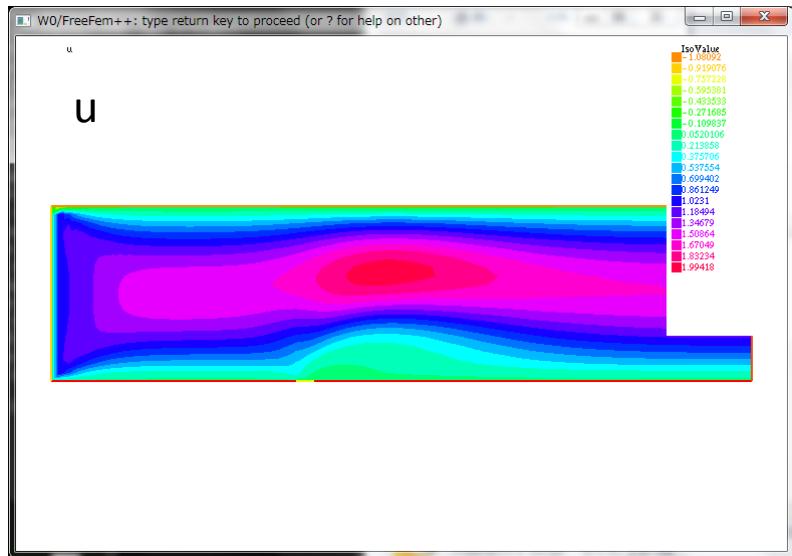
# Simulation scheme



(t=5)

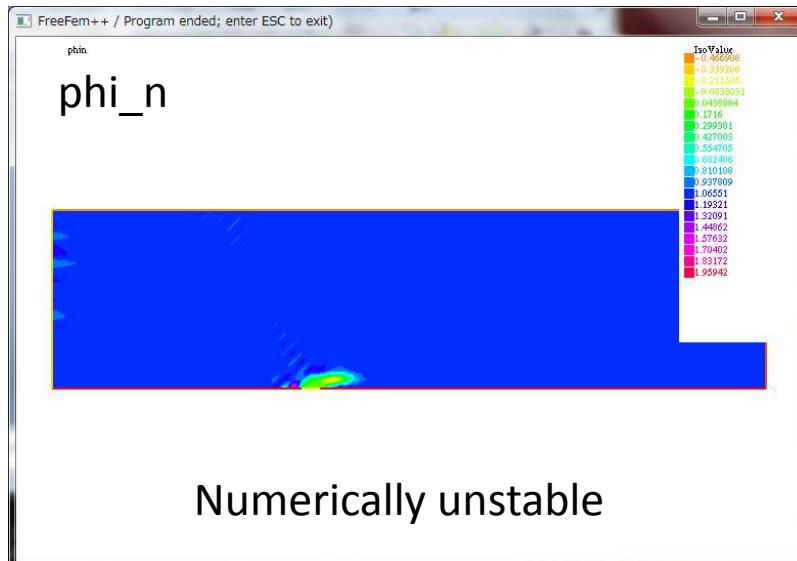
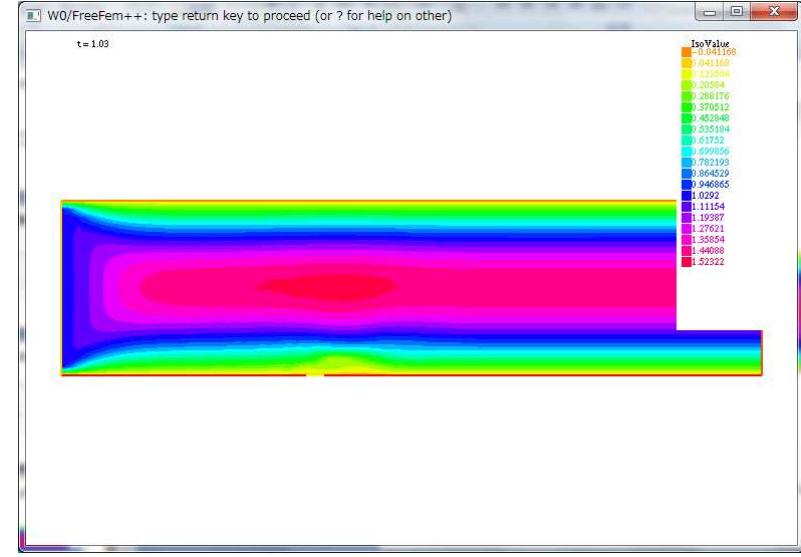
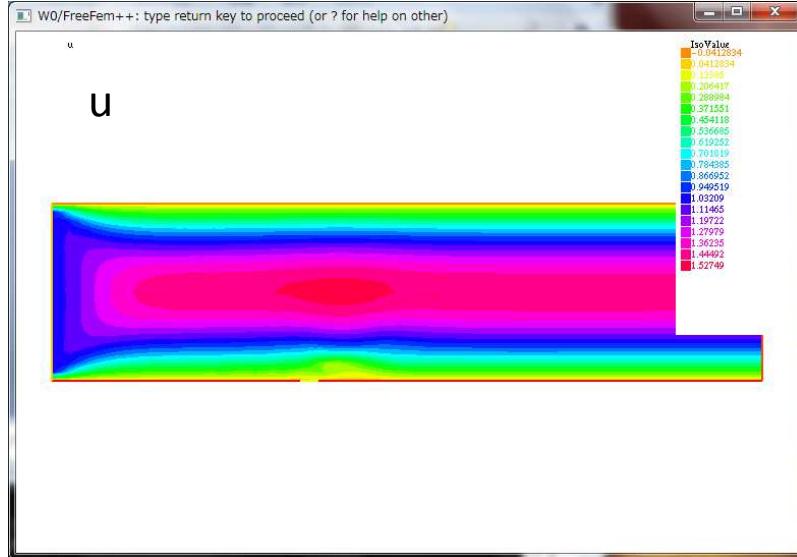


(t=10)

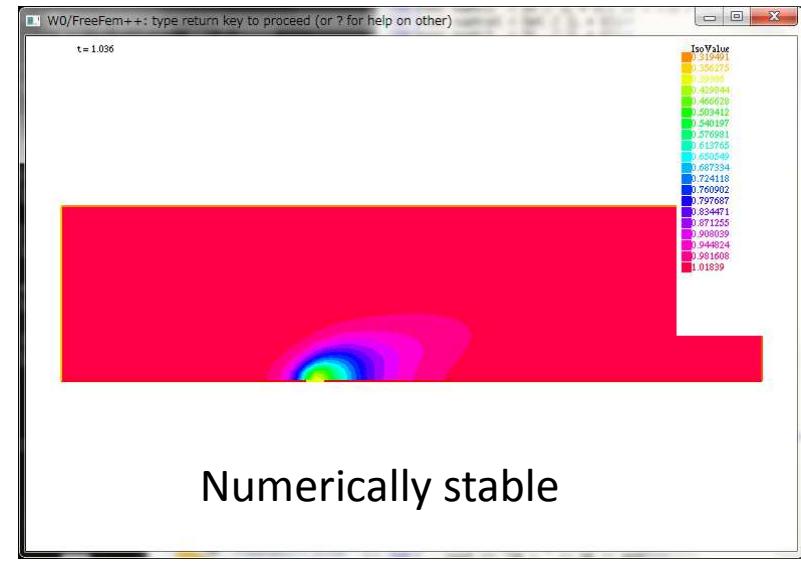


# FF++ solution ( $t=1$ )

Without artificial diffusivity



Numerically unstable



Numerically stable

# Next plan

- Target application level is Elmer(Mechanics) and OpenFORM(CFD)
- Elmer has many kinds of equations
- OpenFORM is popular in CFD

Fixed physical Equations



Pre-build FreeFem++ code  
Only gives parameter values

User defined equations



Current procedure  
Mathml + FreeFem++ + parser

PHML should be modified accordingly

# Summary

- We are developing Physiome Platform PhysioDesigner and it is opened to public by his Home Page <http://www.physiodesigner.org>.
- FreeFem++ excellent feature is utilized for PDE simulation.
- Multiscale model development is being developed and stored into Data Base in cooperation with related organizations.
- For standardization and application to medicine, accumulation of more application examples is necessary.