

Readme (FHN2Dfiber) for ISIDE release 1.4.3

Electrical excitation propagation simulation using Fitzhugh-Nagumo model was already installed as a model of FHN2D.isml in sample directory.

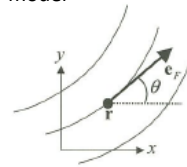
This time, imaging myocardial tissue, fiber structure is included in FHN2D model.

2D fiber model is explained in a following figure.

## Conductivity tensor considering fiber direction

$\sigma_L$ : conductivity along fiber direction  
 $\sigma_T$ : conductivity perpendicular to fiber direction  
 $\theta$ : angle from x axis  
 $\mathbf{e}_F$ : unit vector along fiber direction

2D model



Conductivity tensor:  $\sigma(\mathbf{r}) = \begin{pmatrix} \sigma^x(\mathbf{r}) & \sigma^y(\mathbf{r}) \\ \sigma^y(\mathbf{r}) & \sigma^z(\mathbf{r}) \end{pmatrix} = \sigma_T(\mathbf{r})I_2 + (\sigma_L(\mathbf{r}) - \sigma_T(\mathbf{r}))\mathbf{e}_F(\mathbf{r})\mathbf{e}_F^T(\mathbf{r})$

$\mathbf{e}_F = (\cos\theta, \sin\theta)^T$

$$\mathbf{e}_F \mathbf{e}_F^T = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \end{pmatrix} = \begin{pmatrix} \cos^2\theta & \sin\theta\cos\theta \\ \sin\theta\cos\theta & \sin^2\theta \end{pmatrix}$$

$$\sigma = \begin{pmatrix} \sigma^x & \sigma^y \\ \sigma^y & \sigma^z \end{pmatrix} = \begin{pmatrix} \sigma_T + (\sigma_L - \sigma_T)\cos^2\theta & (\sigma_L - \sigma_T)\sin\theta\cos\theta \\ (\sigma_L - \sigma_T)\sin\theta\cos\theta & \sigma_T + (\sigma_L - \sigma_T)\sin^2\theta \end{pmatrix}$$

myocardial tissue fiber direction

Reference: 「心臓のフィジオロジー」

This fiber model is applied to conductivity epsiron of FHN2D model.

## Applying fiber model to FHN2D

- Apply fiber model to epsiron of FHN2D model

$$\frac{\partial v}{\partial t} = \epsilon \nabla \cdot \nabla v + \lambda(q - v(1 - v)(v - \theta)),$$

$$\frac{\partial q}{\partial t} = \alpha v - \beta q,$$

$$\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$$

$$\text{Assuming } \sigma_T / \sigma_L = \text{eps}_T / \text{eps}_L = 5$$

(along fiber direction is 5 times larger than perpendicular direction)

This is a first trial to use matrix-vector formation in insilicoML and conversion to FreeFem++ is also.

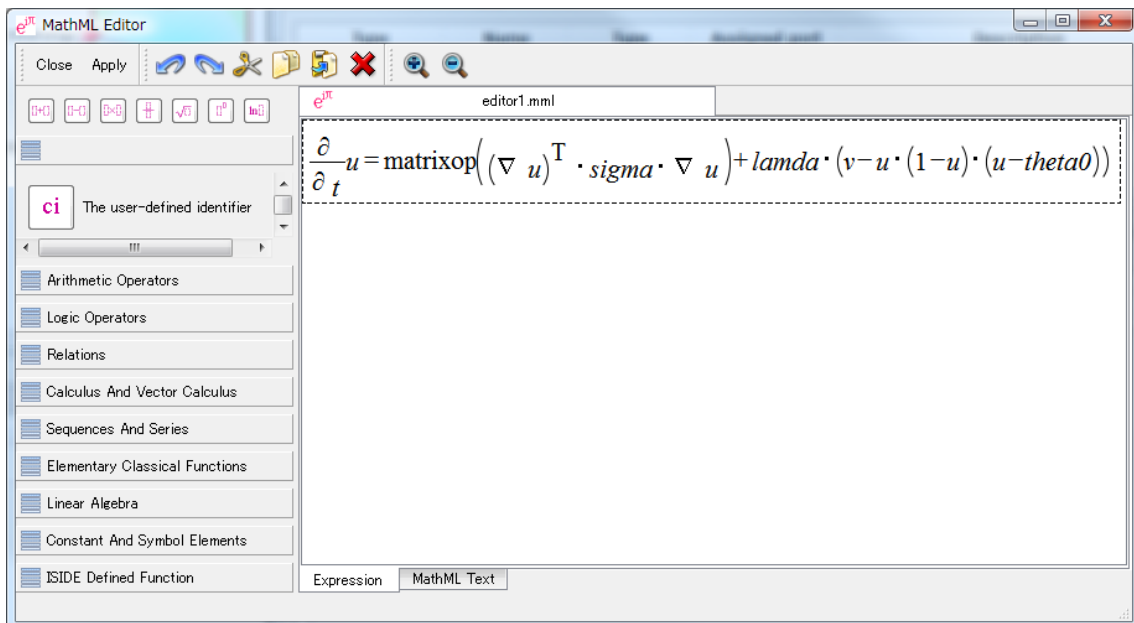
## Matrix-vector formalization for FreeFem++

```
func sigma = [ [ epsT+(epsL-epsT)*cos(theta1)^2,
               (epsL-epsT)*sin(theta1)*cos(theta1) ],
              [ (epsL-epsT)*sin(theta1)*cos(theta1),
                epsT+(epsL-epsT)*sin(theta1)^2 ] ];

macro grad(u) [dx(u),dy(u)]//;

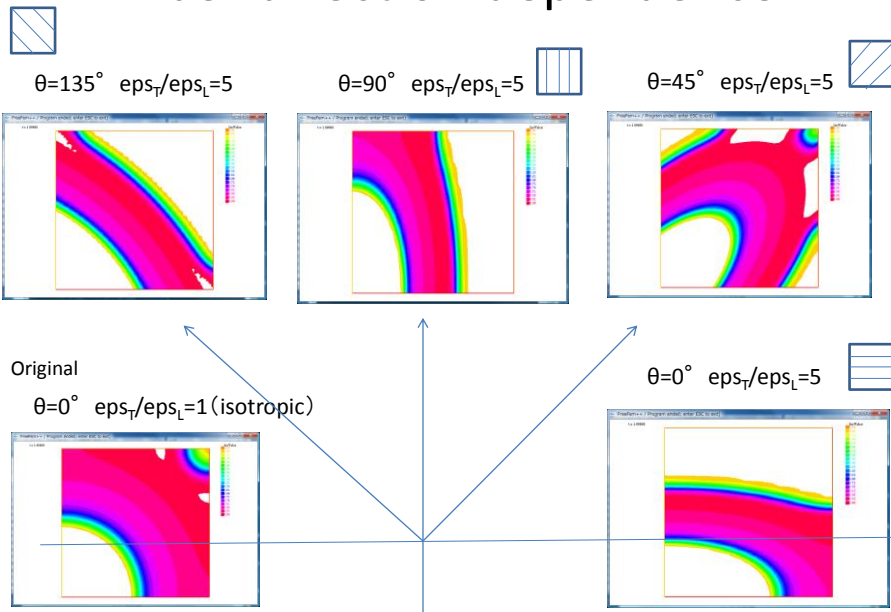
solve fumodelu(u,w)
= .....
+int2d(Th1) (grad(u)'*sigma*grad(w))   ( ' : transpose )
vector x matrix x vector operation
```

Then, we introduce “matrixop” indicator to specify the matrix-vector operation in equation implementation of a “state” physical quantity.



Changing “theta” value, you can see how fiber direction affects excitation propagation.

# Fiber direction dependence



If you change physical quantity “theta” value, you can reproduce this result.