Workshop on Numerical Approximation of PDEs
Honoring 60th birthday of Frédéric Hecht
Málaga, 1-4 Septiembre 2014

Adding user interactivity to FreeFem++ with EJS.
From teaching to research

Presented by Eliseo Chacón Vera[a]

Joint work with:

Francisco Esquembre, María José Cano, Antoine LeHyaric

[a]Dpto. Matemáticas, Facultad de Matemáticas, Universidad de Murcia, España
email: eliseo@um.es

Outline (:-)) of my gratitude to Prof. Hecht:

- **The use of FreeFem++: research and teaching**
  - **research: Still remember reading a fem code in Fortran some years ago!**
  - **teaching: A crash course in numerical pdes from diffusion to Navier-Stokes is now possible and students like it. Makes pdes friendly!**

Outline (:-)) of my gratitude to Prof. Hecht:

- **The use of FreeFem++: research and teaching**
  - **research: Still remember reading a fem code in Fortran some years ago!**
  - **teaching: A crash course in numerical pdes from diffusion to Navier-Stokes is now possible and students like it. Makes pdes friendly!**
- **Prof. Hecht warm personality makes the above much more fun and important**

Outline (:-)) of my gratitude to Prof. Hecht:

- **The use of FreeFem++: research and teaching**
  - **research: Still remember reading a fem code in Fortran some years ago!**
  - **teaching: A crash course in numerical pdes from diffusion to Navier-Stokes is now possible and students like it. Makes pdes friendly!**
- **Prof. Hecht warm personality makes the above much more fun and important**

Big THANK YOU and HAPPY ANNIVERSARY!

Outline of the talk:

- A few words on the teaching of pdes
- Some examples of interactive simulations
  - Undergraduate level
  - Graduate level
- An introduction to Easy Java Simulations (EJS)
- How to create an applet with FreeFem++ and EJS

Why teaching pdes is a difficult task

Pdes are crucial in many science majors....but

- **Solutions are functions not numbers, see them as surfaces, no explicit solutions in general**
- **Even simple equations are tricky. Analytic solutions are not easy to get**. Students do not get the picture!
- **Technical aspects are very complicated** separation of variables, Fourier Series.
- **Model everyday simple physical phenomena (most of them)**
  - Diffusion, transport, reaction, combination of them
  - Wave motion
  - etc...

  still hard to get a visual idea
- **Domain of definition, different boundary conditions, etc,...**
- **Numerical methods: finite difference, elements, science computing,....**

Just blackboard is not enough....some methodology is outdated

Thesis: A picture is worth a thousand words

- **Computer simulations+ interactive engagement**
- **In blackboard session (3-4 h): basic conservation laws, meaning of differential operators, boundary conditions.**
- **Given the computer model...a 2h lab activity for: observation, prediction and testing. Real time interaction of the student with the pde: tunning parameters, making predictions, understand the role of terms in pde and parameters**
- **At least, they get the physical and visual meaning of: Diffusion, transport, reaction, and how are described in math terms**

BUT

- **Create a simulation demands a huge computational effort**
- **Existing softwares do not allow full interactivity**

OPEN SOURCE SOFT DO THE TRICK!: **Full control on the pde**

- **Interface: Easy Java Simulation** (by Francisco Esquembre)
- **Engine: FreeFem++** (by Frédéric Hecht)

Undergraduate level Example 1:

the Laplace operator

With sliders, field boxes or mouse we change

- **Computational domain, mesh size**
- **Boundary conditions**
- **Force term: concavity vs convexity**
- **2D or 3D views.**

i.e., we can play with any parameter

Download:
Item 15 on http://www.um.es/freefem/ff++/pmwiki.php

run: Laplace.jar **run:** CircleWithHoles.jar

Undergraduate level Example 2:

Non-steady Convection-Diffusion-Reaction

- **As before, play with**
  - **Difussion term**
  - **Reaction term**
  - **Transport term**
  - **Show numerical instability**
  - **Compute and recompute**
- **Description available makes it self contained**

run: NonSteadyConvectionDiffusionReaction.jar

Undergraduate level Example 3:

Wave equation

As before, play with

- **Time evolution step by step**
- **Different initial data given**
- **Or data given by user**

run: Wave.jar

Graduate level Example 1:

Steady Stokes driven lid test+convection

Play with mesh and convection at will

run: Stokes2D.jar

Graduate level Example 2:

Navier-Stokes driven lid test

Play with time evolution at will

run: NSdrivenlid.jar

Goals at undergraduate level...

Applets are given and students play with

- Main physical effects
- Real time feedback from changes
- No need of variational formulations
- No need of classical techniques
- See and understand the meaning of a pde

interactive visual learning

Goals at graduate level...

The source of the applets are given and students play with them

- Use variational formulations
- Deeper knowledge of the model
- Get experience with
  - EJS
  - FreeFem++
- Create new applets

Easy Java Simulations (EJS)

Modelling and authoring tool designed by Francisco Esquembre

- Generates simple interactive simulations in Java (platform independent)
- More content than building aspects
- Applets for teaching

Download at http://www.um.es/fem/EjsWiki/

Where EJS belongs

- To the Open Source Physics project http://www.opensourcephysics.org/
- Inside the network ComPADRE Digital Library http://www.compadre.org/
- Science SPORE Prize (November 2011)

Is a network of free online resources for students and teachers in Physics and Astronomy Education.
Idea is to engage students in physics, computation, and computer modeling.

A quick overview on how EJS works

Each simulation has three parts, named panels:

- **Description: to write info, homeworks, etc...**
- **Model: to add variables, equations, coding lines, etc...**
- **View: objects to create a (quite fancy) view**

run: SimplePendulumTest.ejs

Note that the basic equation is:

$$\theta''(t) = -\frac{g}{L}\sin(\theta(t)) - \frac{b}{mL^2}\theta'(t), \quad t > 0$$

$$\theta(0) = 0.4, \quad \theta'(0) = 0$$

More EJS examples

Use the EJS Digital library:

- The comPADRE OSP collection
- The Davidson College (North Carolina)
- The Eckerd College (Florida)
- Universidad de Murcia
- OPS Simulations by Loo Kang (Singapore)

From the column of icons on the rhs use

Read from a EJS digital library option

How to share your EJS example

Several options:

- EJS installed
- Package in a self-executable self-contained **jar** file
- Include into a web site in the form of an applet

From the column of icons on the rhs use

Package current simulation option

How to link FreeFem++ with EJS (1)

Have your FF++ script (toyStokes2d.edp) ready and open a new EJS simulation.

The basic steps

- In **Model panel** go to **Elements subpanel**
  1. select **External folder**
  2. **drag and drop** on the rhs frame an instance of the **FreeFem object**
  3. **open the instance** and paste your FF++ script inside. Name, let's say **toyproblem**
  4. **close the instance**

How to link FreeFem++ with EJS (2)

- In **Model panel** go to **Variables subpanel** and declare (any name possible)
  1. **toto** of type **ScriptOutput**
  2. **triang** of type **PDEData**
  3. **pressure** of type **PDEData**
  4. **velocity** of type **PDEData**

Write **org.colos.freefem.*** on the **Imports** of **Run options** (upper right corner icon)

The EJS vs. FF++ conection has been made!

How to link FreeFem++ with EJS (3)

- In **Model panel** go to **Fixed relations subpanel** and write

```
toto=toyproblem.runScript();
triang=toto.getData(0,0);
pressure=toto.getData(1,0);
velocity=toto.getData(2,0);
```

Our toy problem will be run and each plot is stored into toto.getData according to the appearance in the FF++ script

How to link FreeFem++ with EJS (4)

- In **View panel** we construct a view tree selecting containers. Click on any object and drag and drop on the right panel

  1. from the **Interface and Windows, containers and drawing panels** group get
     (a) a **Frame object**
     (b) a **PlottingPanel object**
  2. from the **2D Drawables and Fields and plots group** get a **mesh2D object** and add it to the PlottingPanel object
  3. double click on the **mesh2D object** select as **data field** any variable **triang, pressure or velocity**

  Save as **toyproblem.ejs** and click the **green triangle (play)**

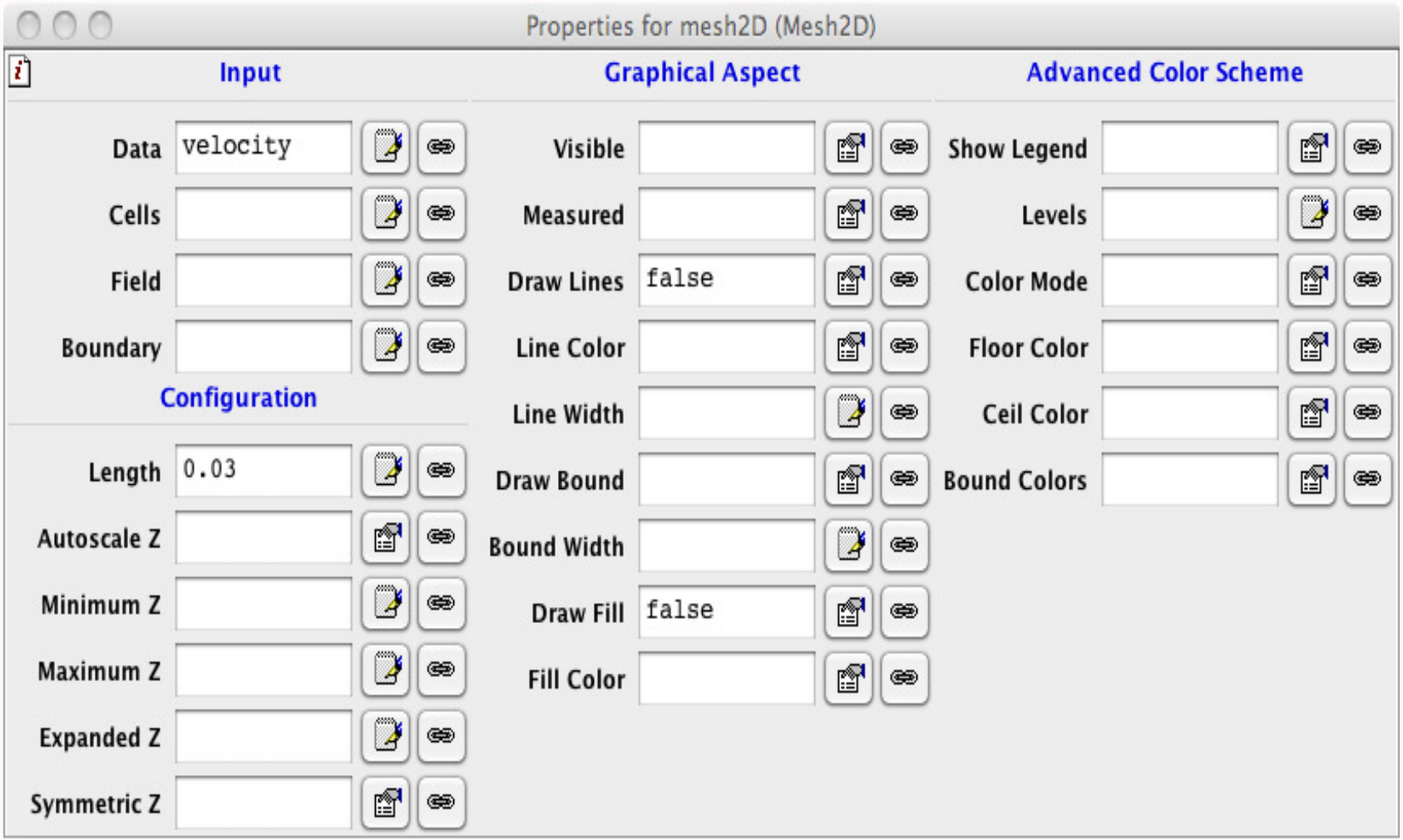
Improve the view

Check properties of objects in **View panel**.
Double click on

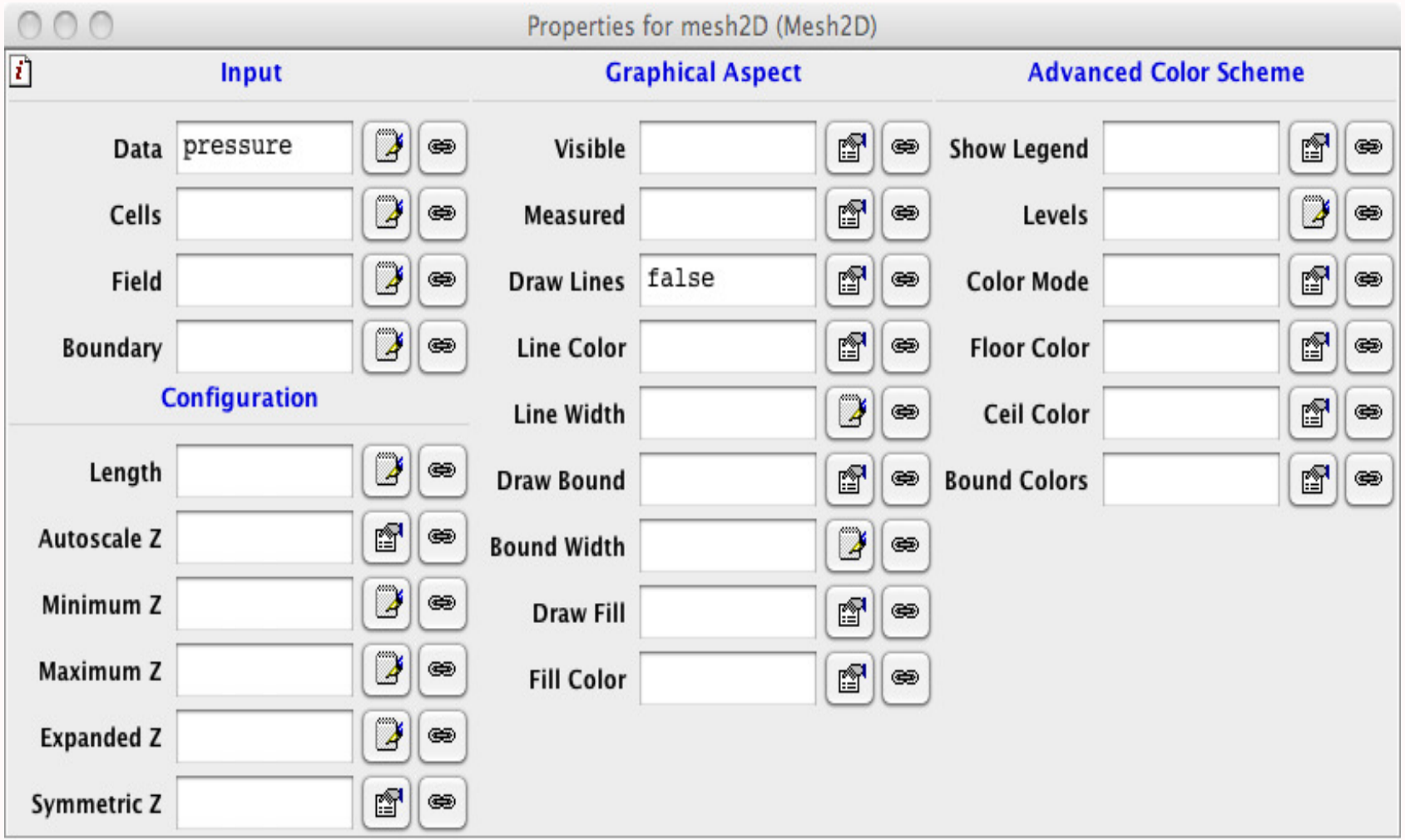
- **frame**: change position, size, title, etc...
- **plottingPanel**: change grid, etc...
- **mesh2D**: change draw lines to see the mesh, etc...

Updates by click the **red square** / **green triangle (play)**

Possible options for velocity



Possible options for pressure

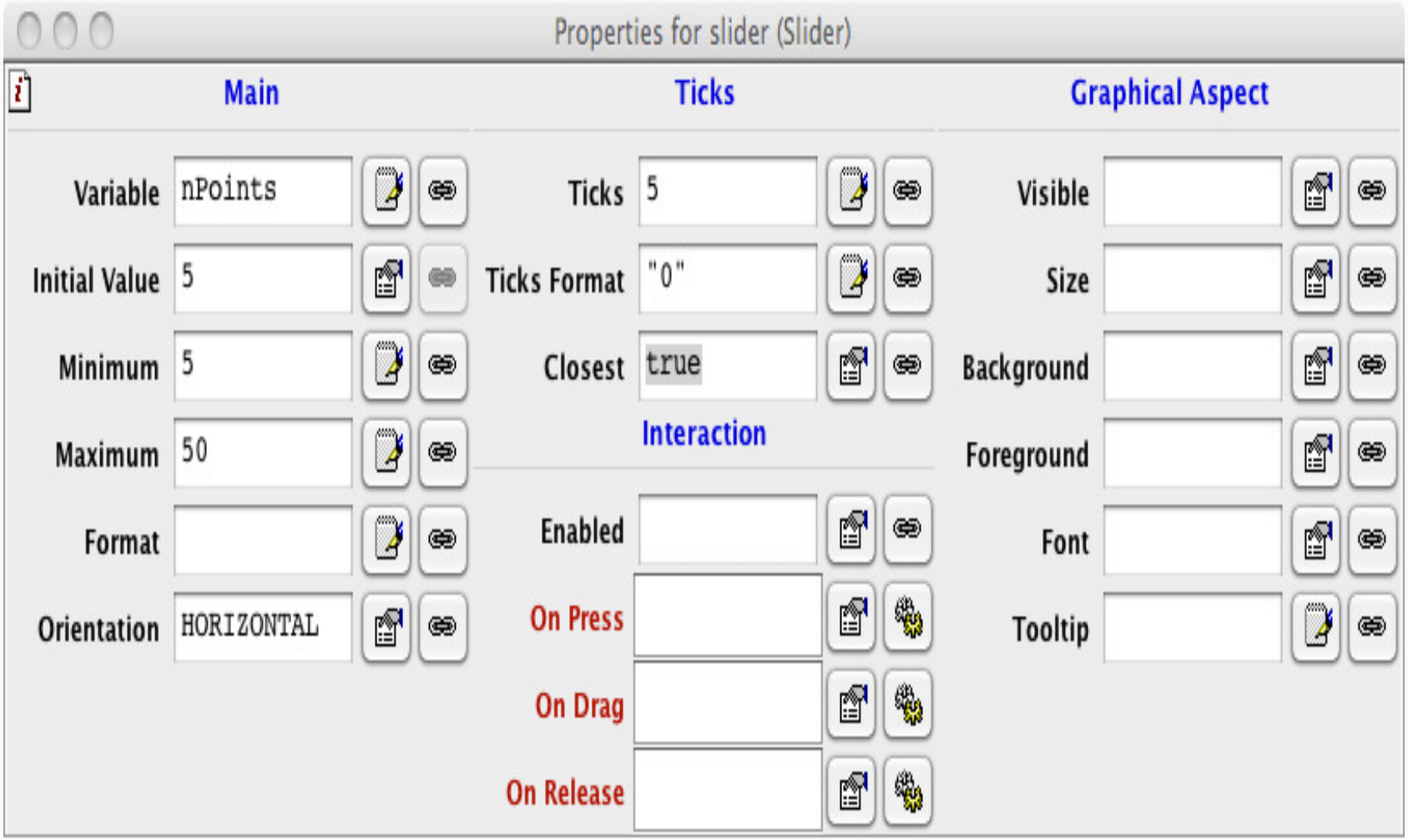


Interactivity

On **View panel** use the **Input and output group** .
Double click on

- **Slider**: adds a slider to the view, choose the Up position
- **$(varName)**: links the FF++ script to the slider via varName
- **Open the FF++ script from Model, Elements** and use **$(varName)**: for instance **int nn=$(nPoints);**
- **Declare** variable nPoints
- **Now slider links code** via variable nPoints

Possible options for slider



Conclusions on EJS–FF++ :

- **adds to teaching strategies on pdes**
- **creates an interface for any other applications**

More info:

- Bringing pdes to life for students, European Journal of Physics, 36 (2015)
- Simulations of PDE models in Java, submitted

Thank you for your attention