StabFem, a Matlab/Octave interface to FreeFem++

Status and Future of the project

David Fabre (IMFT, Toulouse)

M. Pigou (IMFT), J. Sierra (IMFT/Univ. Salerno)

(and many other contributors)

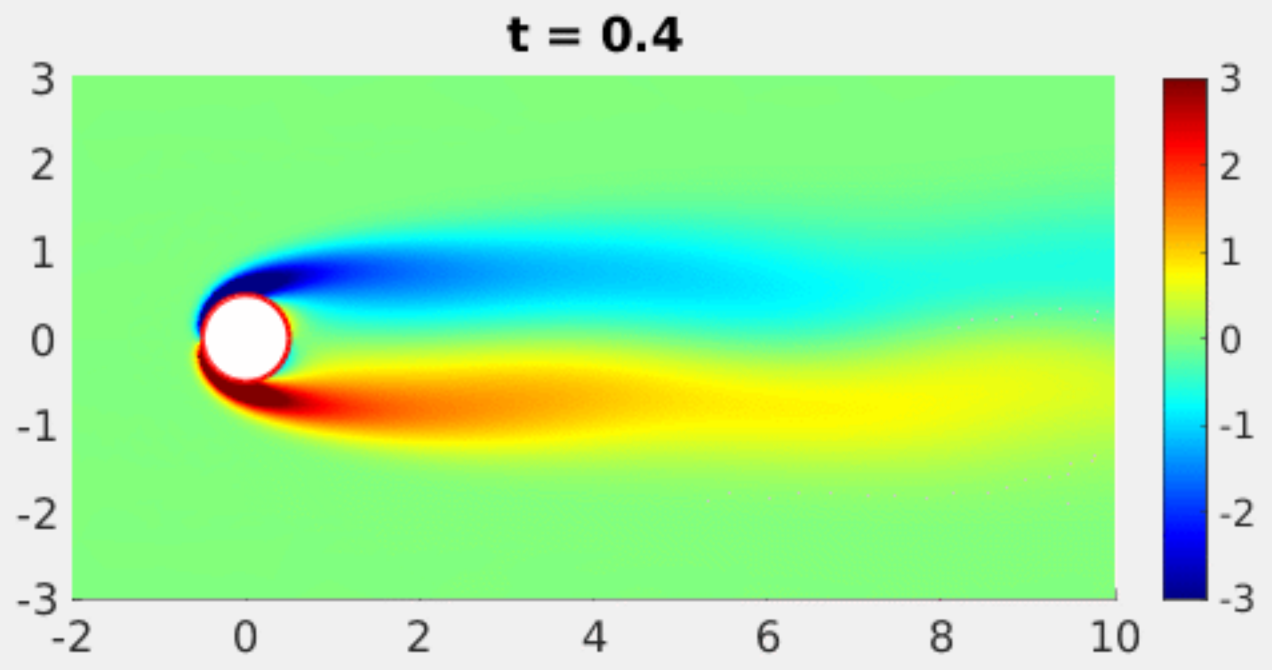
12th FreeFem Days, 11 dec. 2020.

Summary

- Who we are, what we do, why we designed this
- Main commands : `SF_Launch`, `SF_Mesh`, `SF_Plot`, `SF_Adapt`, `SF_Status`, `SF_Load`
- Demonstration : the L-shape problem
- How does it work ? File exchange formats
- Web facilities
- Achievements and future evolutions : towards a Python interface ?

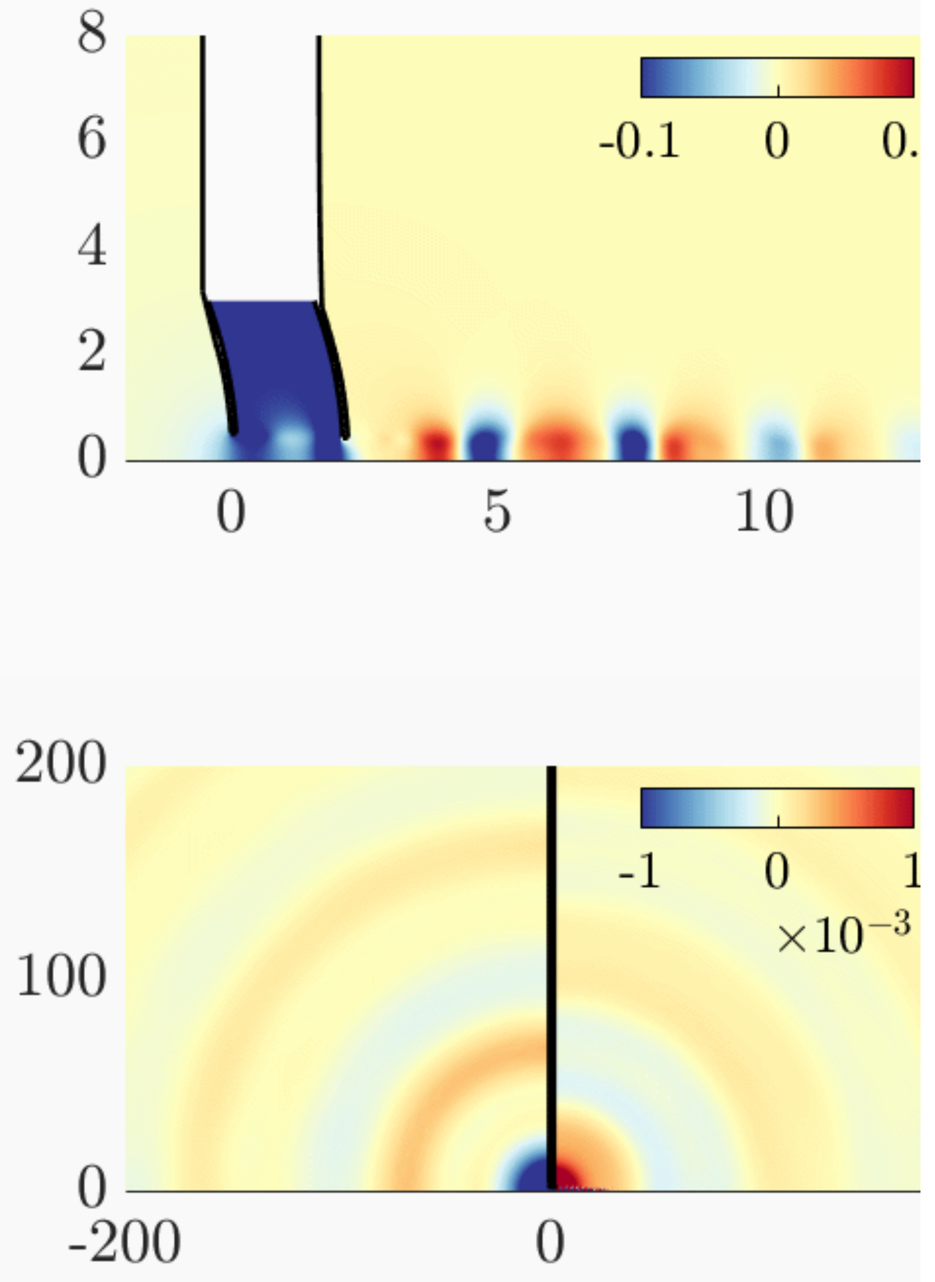
A few problems studied in IMFT and collaborating teams

(a) Instabilities and mode interactions in wake problems



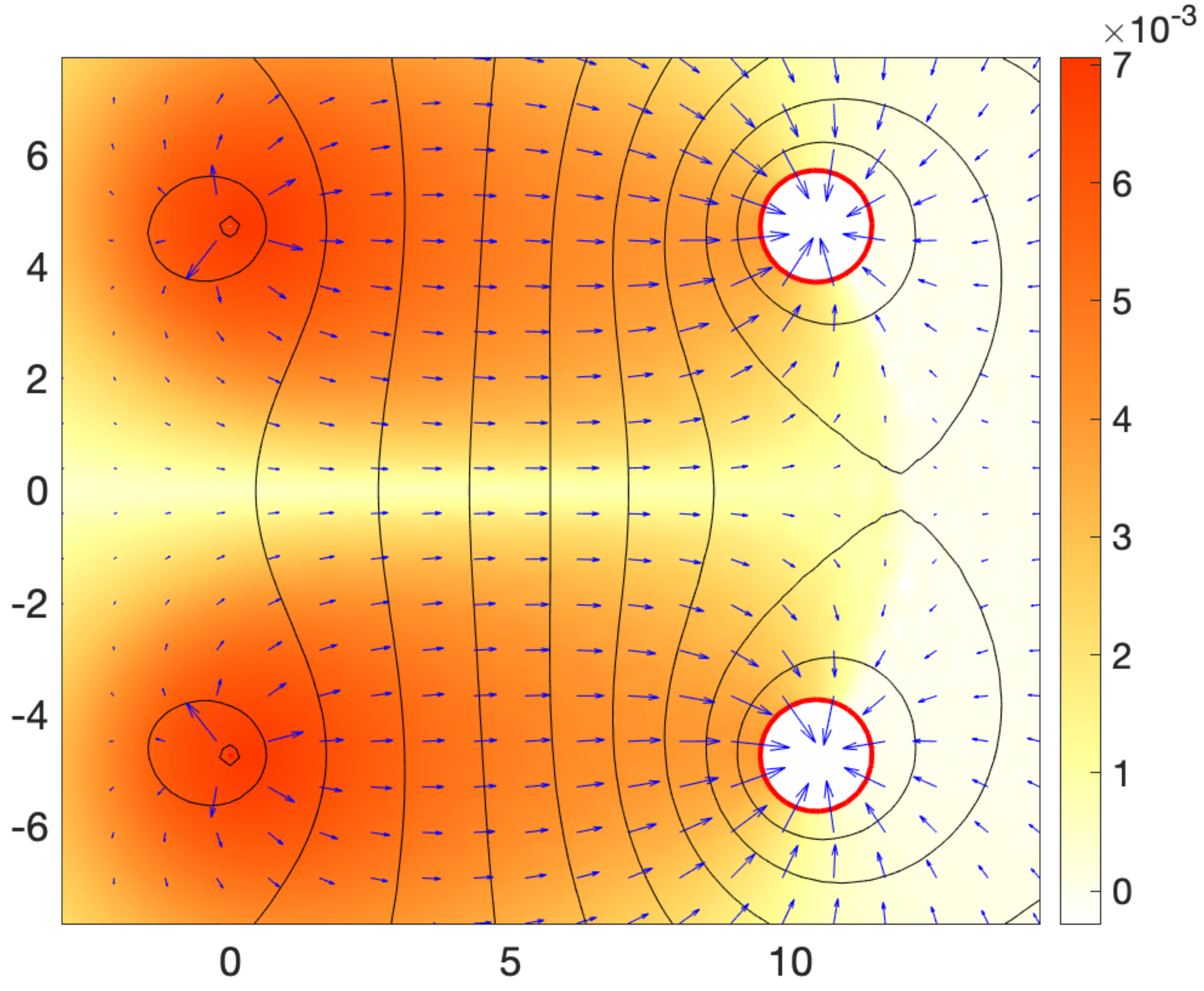
PhD. J. Sierra & D. Sabino ;
ONERA, Univ. Jaen

(b) Jet instabilities



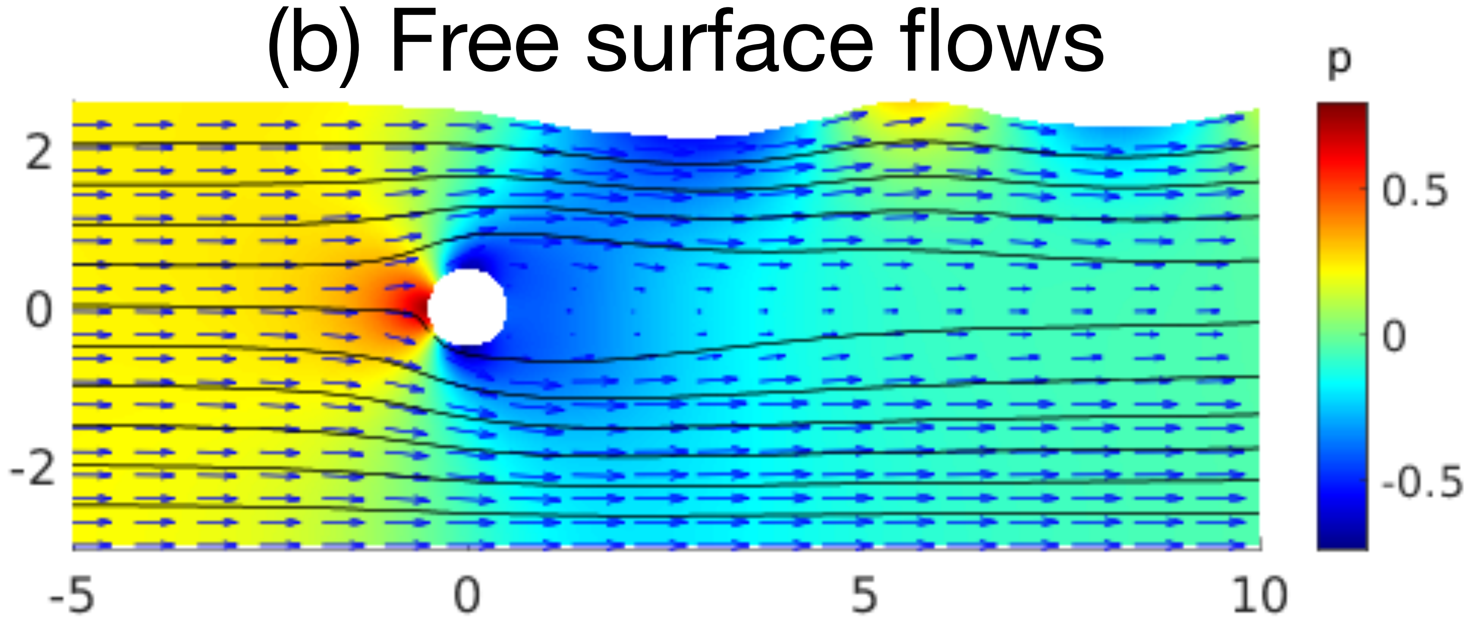
PhD. J. Sierra & R. Longobardi ;
Salerno, Berlin, Madrid

(d) Ionic wind propulsion



PhD. C. Marion

(b) Free surface flows

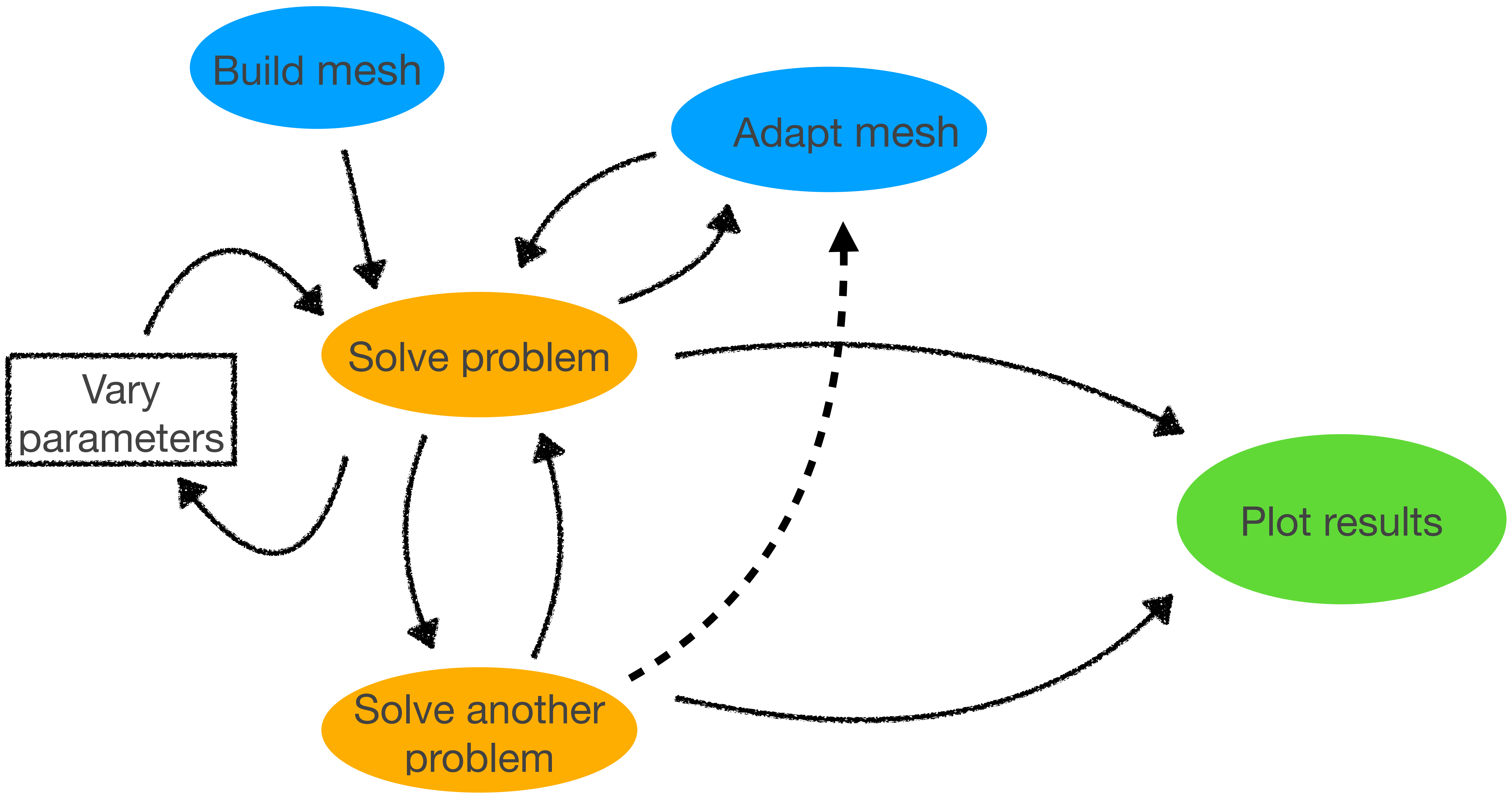


(PhD. P. Bonnefis & N. Achour)

Common approach: Linear and nonlinear instability analyses

Common tool: FreeFem++ !

A typical computation/postprocessing chain



Freefem++ is a very efficient tool to perform such studies
BUT : A whole parametric study can hardly be done entirely within a single FreeFem++ program

-> Freefem programs for « Elementary bricks » have to be successively launched (many people use shell scripts)
-> The graphical posprocessing is generally done outside FreeFem (Paraview, tecplot, gnuplot, Matlab,…)

-> Idea : a generic interface to manage the whole computation chain AND post processing chain

Specifications (« cahier des charges »)

- An interface using a high-level language, usable in command-line or script mode, containing a built-in plotting tool, and a generic database management system.
- An open repository gathering a collection of FreeFem codes (and maintaining them operational) as well as a solution to publish codes as « literate programs » (cf. Basilisk, JuPyter,…)
- Easy to plugin existing FreeFem programs (or inversely to take our FF programs and run them outside of the interface)
- Usable for education as well as research
- Multi-platform, open-source and freeware

=> the StabFem project based on the Octave/Matlab language and environment.

Started 2017, presented at FF-days in 2018, now reaching maturity.

Summary

- Who we are, what we do, why we designed this
- Main commands : `SF_Launch`, `SF_Mesh`, `SF_Plot`, `SF_Adapt`, `SF_Status`, `SF_Load`
- Demonstration : the L-shape problem
- How does it work ? File exchange formats
- Web facilities
- Achievements and future evolutions : towards a Python interface ?

SF_Launch : the generic FreeFem wrapper

Return object is a « handle » to the results
(Structure containing information on result file(s), metadata,…
Usable for plotting and/or usage by a subsequent FreeFem program)

```
handle = SF_Launch(
            ‘myprogram.edp’,
            ‘Mesh’, mymesh,
            ‘Options’,{‘Re’,100,’nit’,1000},
         [ ‘Init’, init_handle, ]
         [  ‘Datafile’, ‘Data.txt’, ]
         [  ‘Store’, ‘FolderName’, ]
         [ ‘Macros’,{’macro MyMacro 1 \\’,…},]
         [ ‘Params’, [100 1000] ]
         [  … ]
         )
```

FreeFem solver to use

Mesh to use (file name or handle)

List of options. Here the wrapper will launch

```
FreeFem++-mpi myprogram.edp -Re 100 -nit 1000
```

Name of datafile(s) to import

Name of Data Base folder to store results

A few lines to be included in the solver

Alternative method to transmit parameters.
Here the wrapper will launch

```
echo ’100 1000’ | FreeFem++-mpi myprogram.ed
```

NB : for a mesh generator use SF_Mesh instead (similar syntax)

SF_Plot : The generic plotting tool

Based (since 2019) on ffpdeplot library by M. Meitziger

- Allows to plot P2 (and P1b) data without downgrading to P1.
- Recognizes data defined on vectorial FE-spaces (e.g. [P2,P2,P1], etc…)
- A large number of options covering all needs of postprocessing for publications

Usage :

`SF_Plot(handle,’field’,[...])` -> color levels

`SF_Plot(handle,{’ux’,’uy’},[...])` -> quiver plots (vector field)

(…)

`SF_ExtractData` : To extract the value of fields at a « probe » or an array of probes
(Also based on M Meitziger’s library)

SF_Adapt : the mesh-adaptation wizard

Generic wrapper to Freefem’s powerful adaptmesh command;

Usage: [field1,field2,...] = SF_Adapt(field1,field2,... [,options...])

Works with up to 16 datasets (or 8 complex) and recognizes a large number of FE-spaces ([P2,P2,P1],[P2,P2,P2,P1],[P1b,P1b,P1] …)

NB : FreeFem code -> https://gitlab.com/stabfem/StabFem/-/blob/master/SOURCES_FREEFEM/AdaptMesh.edp

DatabaseTools

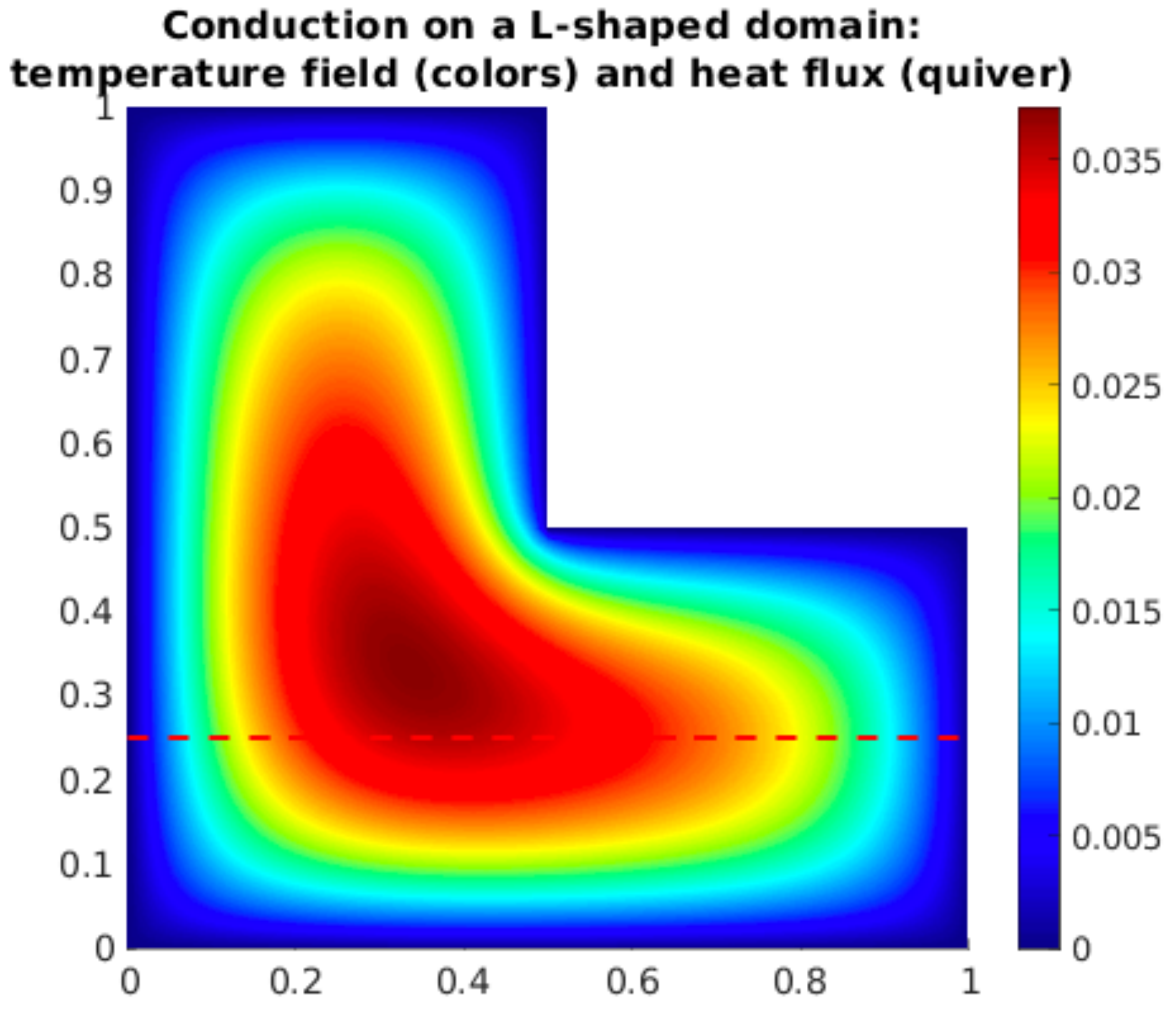
- SF_Status generates a summary of the database, indexed with all metadata identified in result files.
- SF_Load allows to import one dataset from the database, for postprocess or restarting the study.

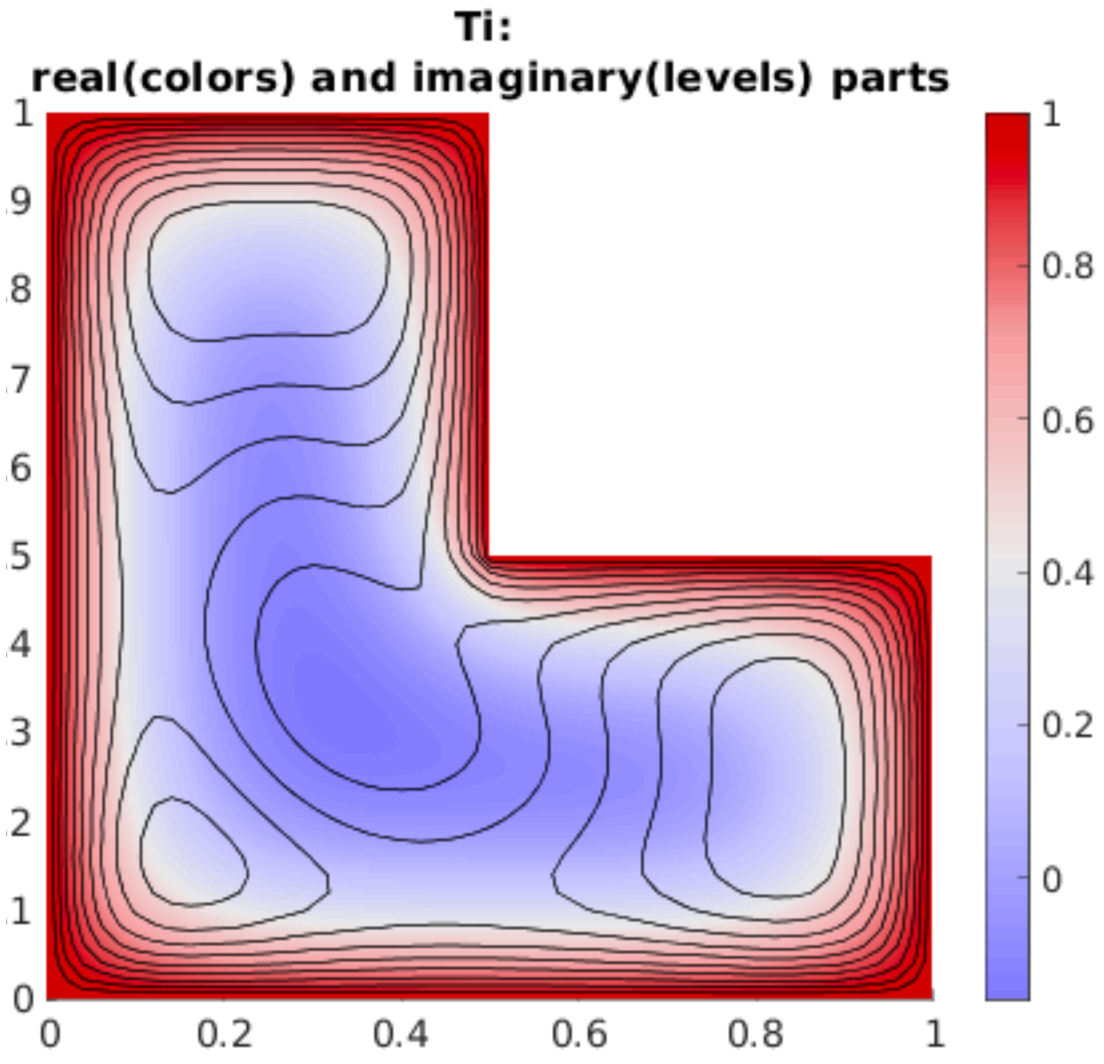
Summary

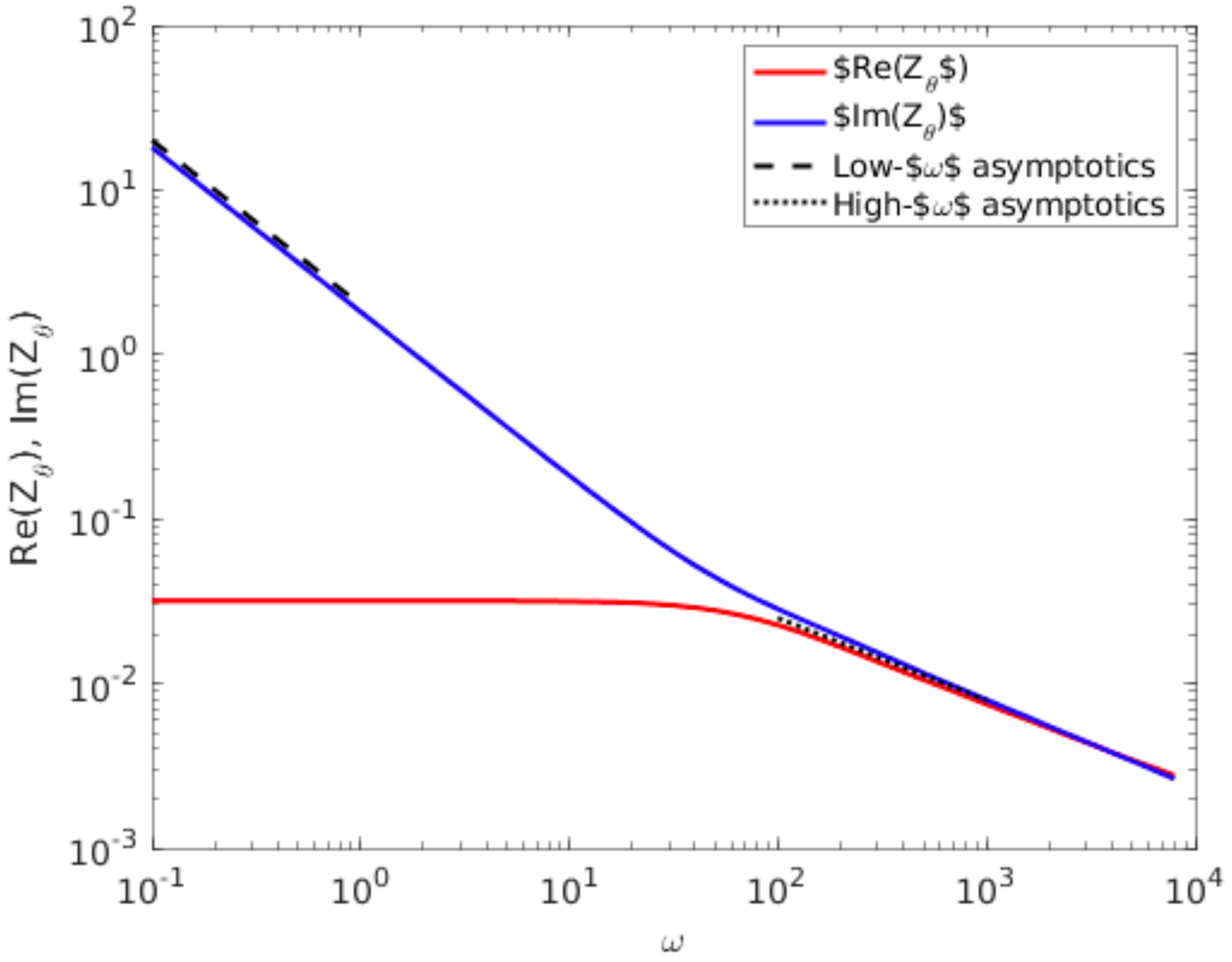
- Who we are, what we do, why we designed this
- Main commands : `SF_Launch`, `SF_Mesh`, `SF_Plot`, `SF_Adapt`, `SF_Status`, `SF_Load`
- Demonstration : the L-shape problem
- How does it work ? File exchange formats
- Web facilities
- Achievements and future evolutions : towards a Python interface ?

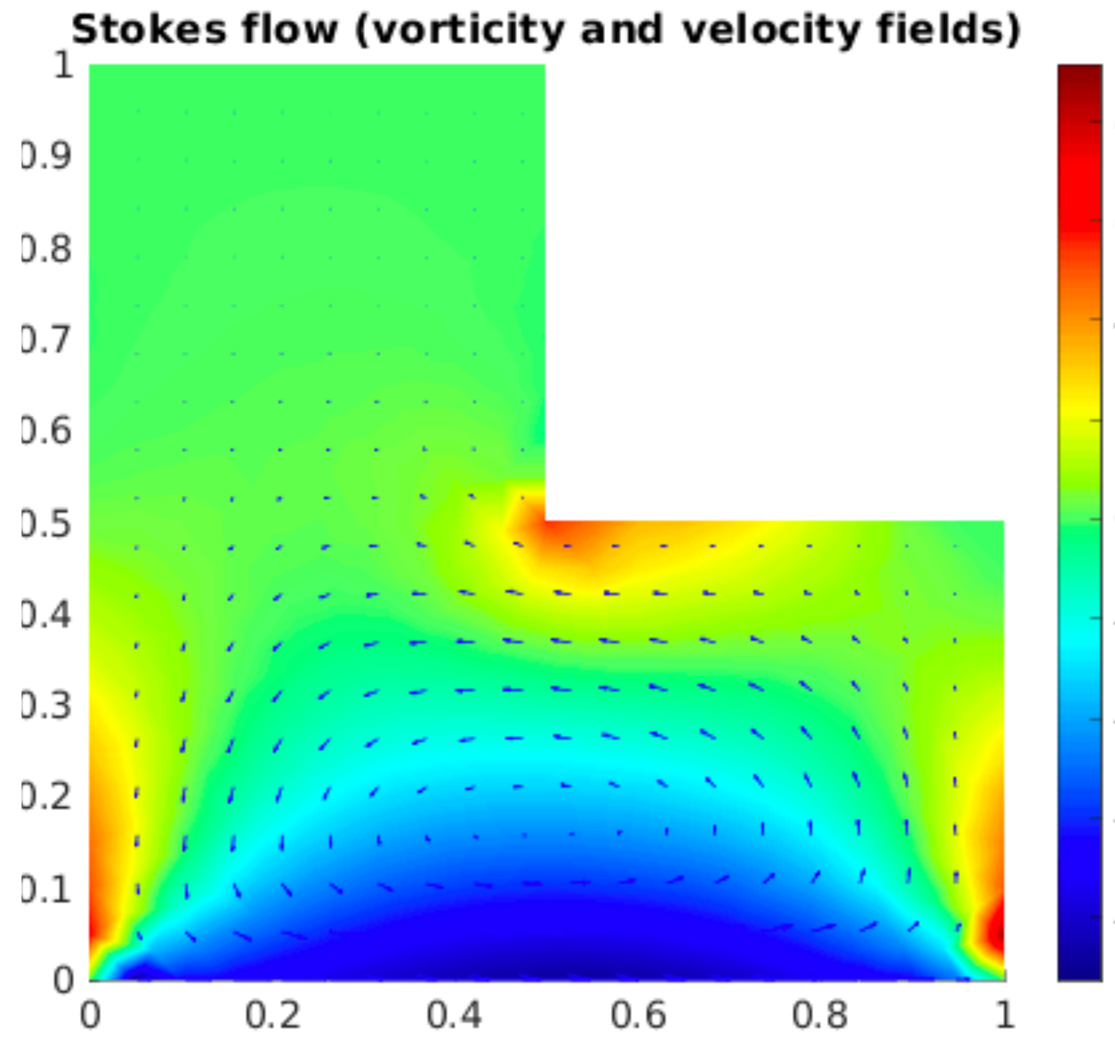
Demonstrations : the L-shape example

https://stabfem.gitlab.io/StabFem/STABLE_CASES/EXAMPLE_Lshape/SCRIPT_Lshape.html


Conduction on a L-shaped domain:
temperature field (colors) and heat flux (quiver)

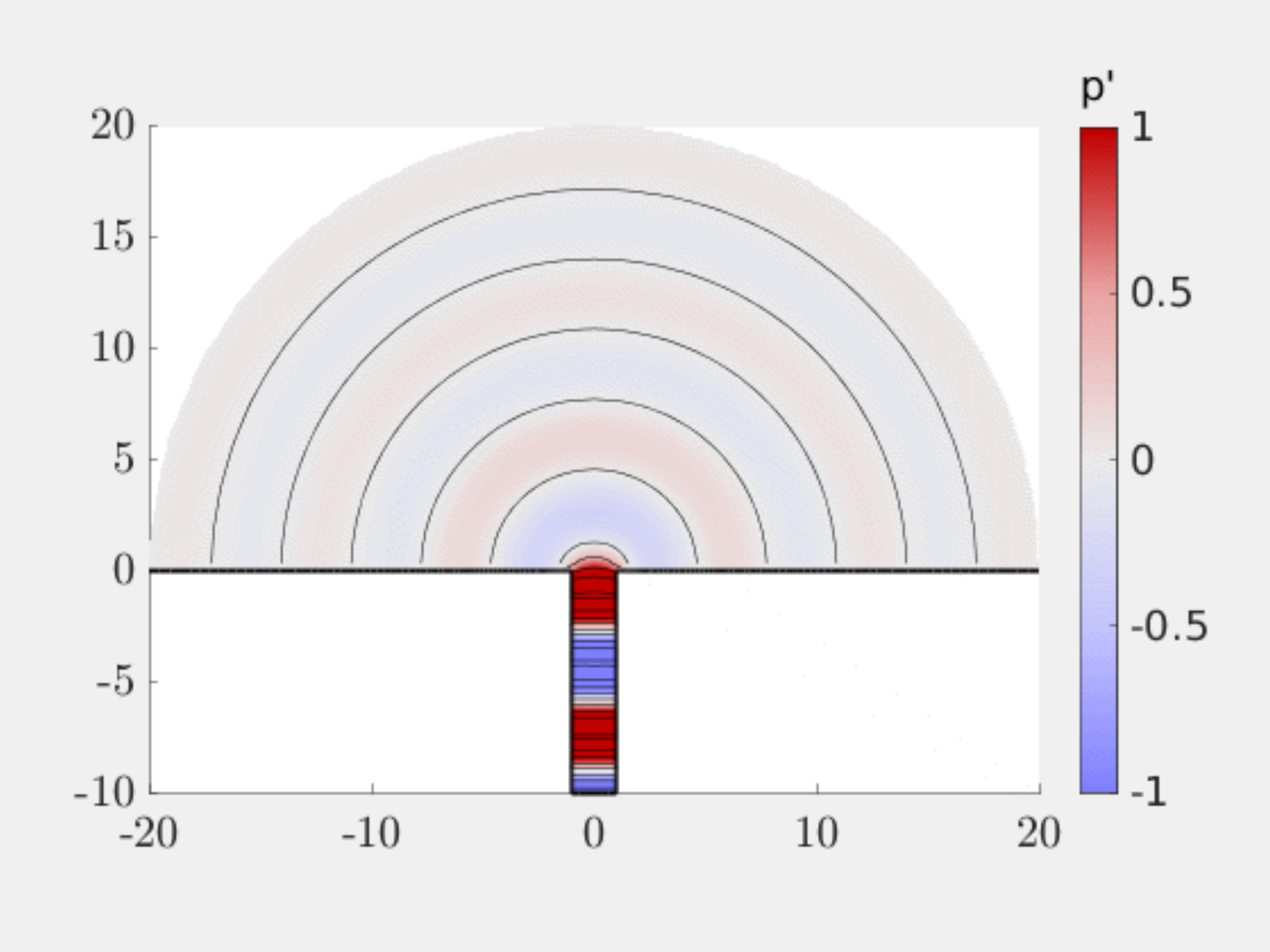

Ti:
real(colors) and imaginary(levels) parts




Stokes flow (vorticity and velocity fields)

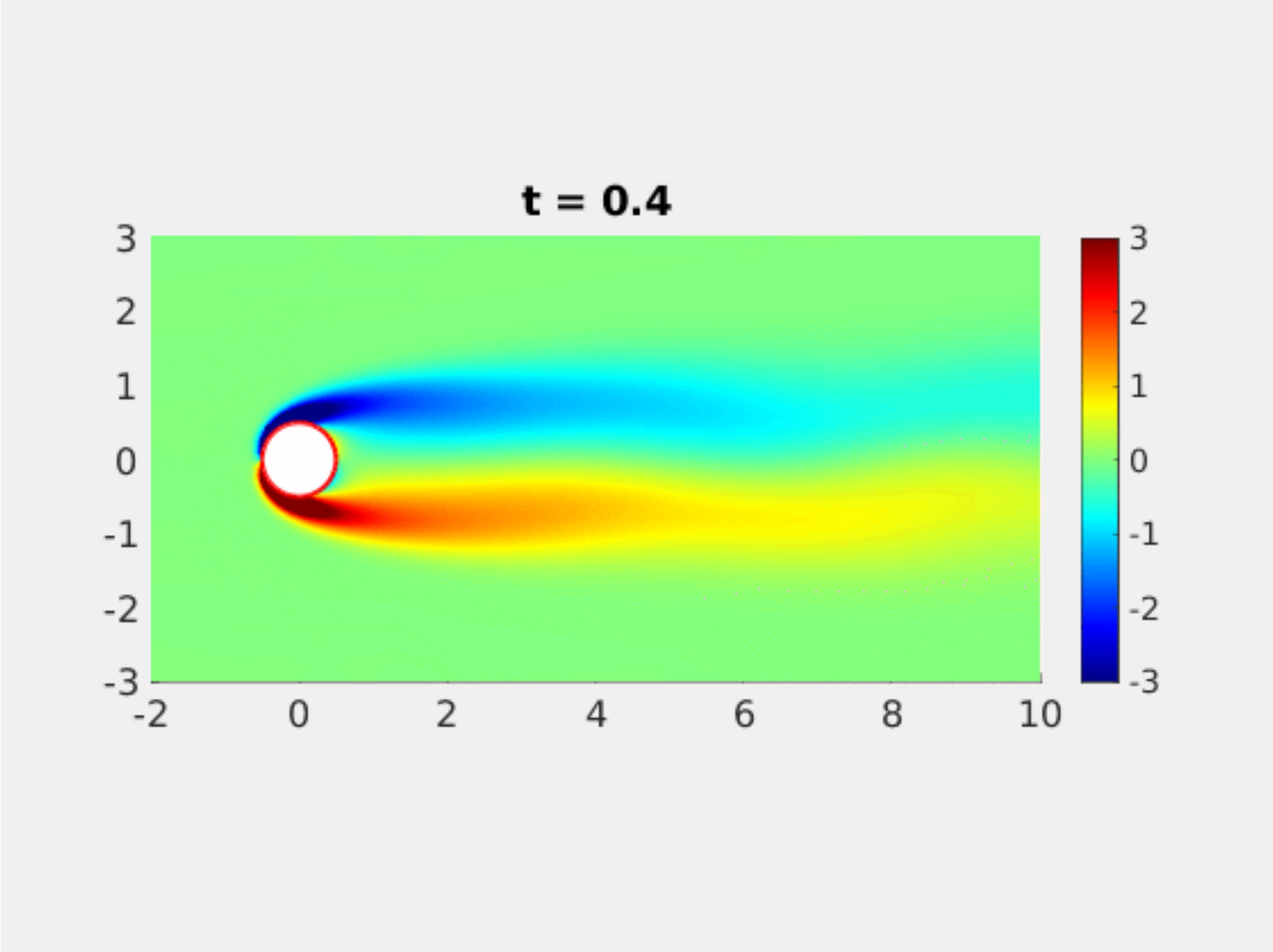
More examples :

Acoustic radiation from an open pipe



https://stabfem.gitlab.io/StabFem/STABLE_CASES/ACOUSTICS_PIPES/SCRIPT_DEMO_ACOUSTIQUE.html

Onset of Von-Karman vortex street in the wake of a cylinder



https://stabfem.gitlab.io/StabFem/STABLE_CASES/DNS_CYLINDER/SCRIPT_DNS_EXAMPLE.html

Summary

- Who we are, what we do, why we designed this
- Main commands : SF_Launch, SF_Mesh, SF_Plot, SF_Adapt, SF_Status, SF_Load
- Demonstration : the L-shape problem
- How does it work ? File exchange formats
- Web facilities
- Future evolutions : towards a Python interface ?

Exchange file formats

Most common (and previous) strategy :

- FreeFem native files (.txt or .bin) for subsequent FreeFem computations
- Graphical files (.vtk , etc…) for postprocessing

(Drawback: redundancy, no universal method for data indexing)

Adopted strategy :

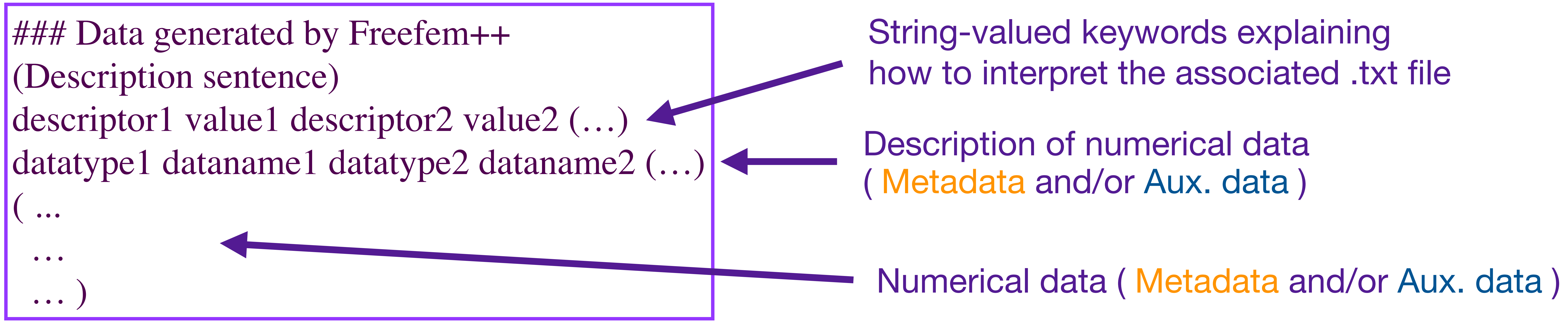
- Freefem native format (.txt) for BOTH FreeFem and posprocess.
- Auxiliary file with custom format (.ff2m) containing:

A. Instructions on how to interpret the data in .txt files

B. Auxiliary data useful for postprocessing (e.g. vorticity, …)

C. Metadata used to generate an index of the database.

Explanation of the .ff2m format.



```
### Data generated by Freefem++
(Description sentence)
descriptor1 value1 descriptor2 value2 (…)
datatype1 dataname1 datatype2 dataname2 (…)
( ...
  …
  … )
```

String-valued keywords explaining how to interpret the associated .txt file

Description of numerical data (Metadata and/or Aux. data)

Numerical data (Metadata and/or Aux. data)

Example 1
.txt file contains mesh-associated data
corresponding to a flow field [u,v,p]

```
### Data generated by Freefem++ ;
Navier-Stokes flow in a L-shaped bottom-driven cavity
datatype BaseFlow datastoragemode ReP2P2P1 datadescriptors ux,uy,p
real* Re real* Fx P1 vorticity
10
-1.2665
(… values of vorticity field …)
```

Example 2
.txt file contains three columns of data generated by a loop
(non-mesh-associated data)

```
### Data generated by Freefem++ ;
Solution of the unsteady thermal problem over a range of omega
datatype thermal datastoragemode columns datadescriptors omega,Flux_r,Flux_i
real* kappa
1.
```

Summary

- Who we are, what we do, why we designed this
- Main commands : `SF_Launch`, `SF_Mesh`, `SF_Plot`, `SF_Adapt`, `SF_Status`, `SF_Load`
- Demonstration : the L-shape problem
- How does it work ? File exchange formats
- Web facilities
- Achievements and future evolutions : towards a Python interface ?

Web facilities

- Sources are maintained on gitlab : https://gitlab.com/stabfem/StabFem/

  NB the repository contains ~100 FreeFem++ solvers (mostly oriented towards flow instability problems) which can be perfectly used outside of the interface !

  https://gitlab.com/stabfem/-/tree/master/SOURCES_FREEFEM

- A dynamically generated website where all users can publish « Literate programs » (inspired by Basilisk)

  https://stabfem.gitlab.io/StabFem/

  Howto ? Simply put a tag [[PUBLISH]] in your code and commit/push on gitlab !

- Website actually contains ~60 programs, including 5 tutorial examples and code for 6 published research papers.

- Other tools offered by gitlab : nonregression « autorun » tests, automatic generation of documentation, etc…

Summary

- Who we are, what we do, why we designed this
- Main commands : `SF_Launch`, `SF_Mesh`, `SF_Plot`, `SF_Adapt`, `SF_Status`, `SF_Load`
- Demonstration : the L-shape problem
- How does it work ? File exchange formats
- Web facilities
- Achievements and future evolutions : towards a Python interface ?

Achievements

- Stable version StabFem 3.8 does the job !

- A manual is available:

  https://gitlab.com/stabfem/StabFem/-/jobs/848781676/artifacts/file/99_Documentation/MANUAL/main.pdf

- Currently ~15 identified active users in research area (IMFT, Salerno, Berkeley, Jaen, Madrid, Berlin…), possibly much more more !

- Used for education at UPS, Toulouse

  (Master 1 « mécanique et énergétique », TPs de Mécanique des Fluides numériques et Ondes).

- 10+ research papers (6 papers with codes published on the website)

-> « Showcase paper » https://gitlab.com/stabfem/stabfem_publications/-/blob/master/Fabre_etal_AMR_2018.pdf

- Identified as « Projet Laboratoire » by IMFT ;

  Support from « Cosinus » code-development service
  Server for the website
  Expected funding for a forthcoming 6 month training course.

Current limitations :

- Limited to 2D meshes
- Code quality is not optimal (some parts deserve to be fully rewritten…)
- Not 100% freeware solution
- Choice of Matlab/Octave turns out to be a brake for development and diffusion of the software

Time for a translation towards Python ?

- Many advantages of Python vs. Matlab/Octave:
  Fully open-source and multi-platform language,
  Tools for code reliability (unitary functional tests,…) and notebook publishing (JuPyter),
  An active and growing base of developers and users, etc…

The future of StabFem

Now at the crossroads…

Keep on developing this on our side according to our needs

Or Continue with help and support from the FreeFem community ?

Keep the current Matlab/Octave choice

Or Rewrite everything in Python ?

Who is interested ? Who wants to contribute ?

To launch the debate …

- Exchange formats ? Is our solution (.txt/.ff2m files) satisfying for all potential users ? Should other formats be considered ?
- Ideas for full integration of 3D cases and parallel computing ???