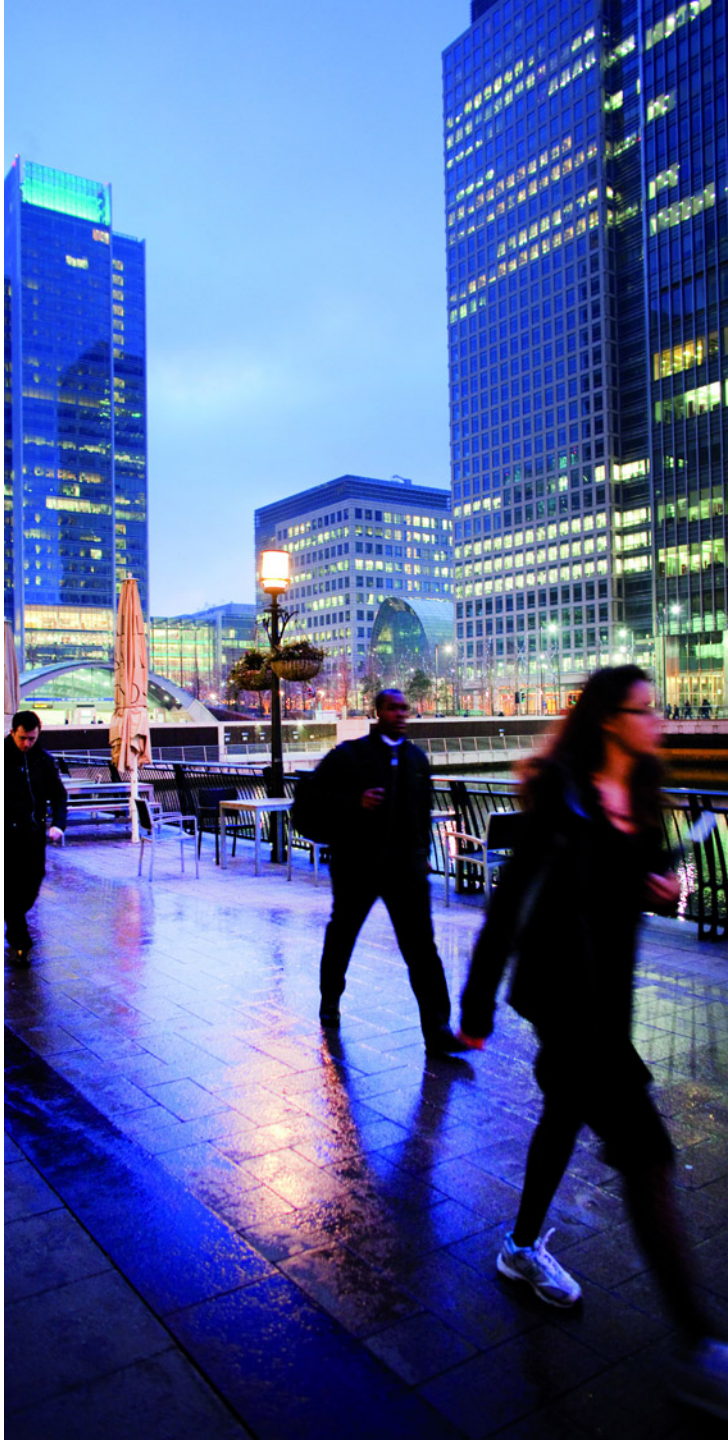
Post-processing: coming next in SALOME

Cédric Aguerre – EDF Lab Paris-Saclay
Adrien Bruneton – CEA Saclay
Clarisse Genrault – CEA Saclay

SALOME User Day 2016
EDF Lab Paris-Saclay

December 9th 2016



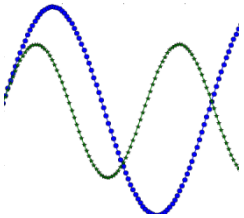
AGENDA

1. MEDCALC

A FIRST LEVEL ACCESS TO PROCESSING AND VISUALIZATION FEATURES OF MESHES AND FIELDS IN SALOME

2. CURVEPLOT

HIGH LEVEL MATPLOTLIB ACCESS INTO SALOME



MEDCALC

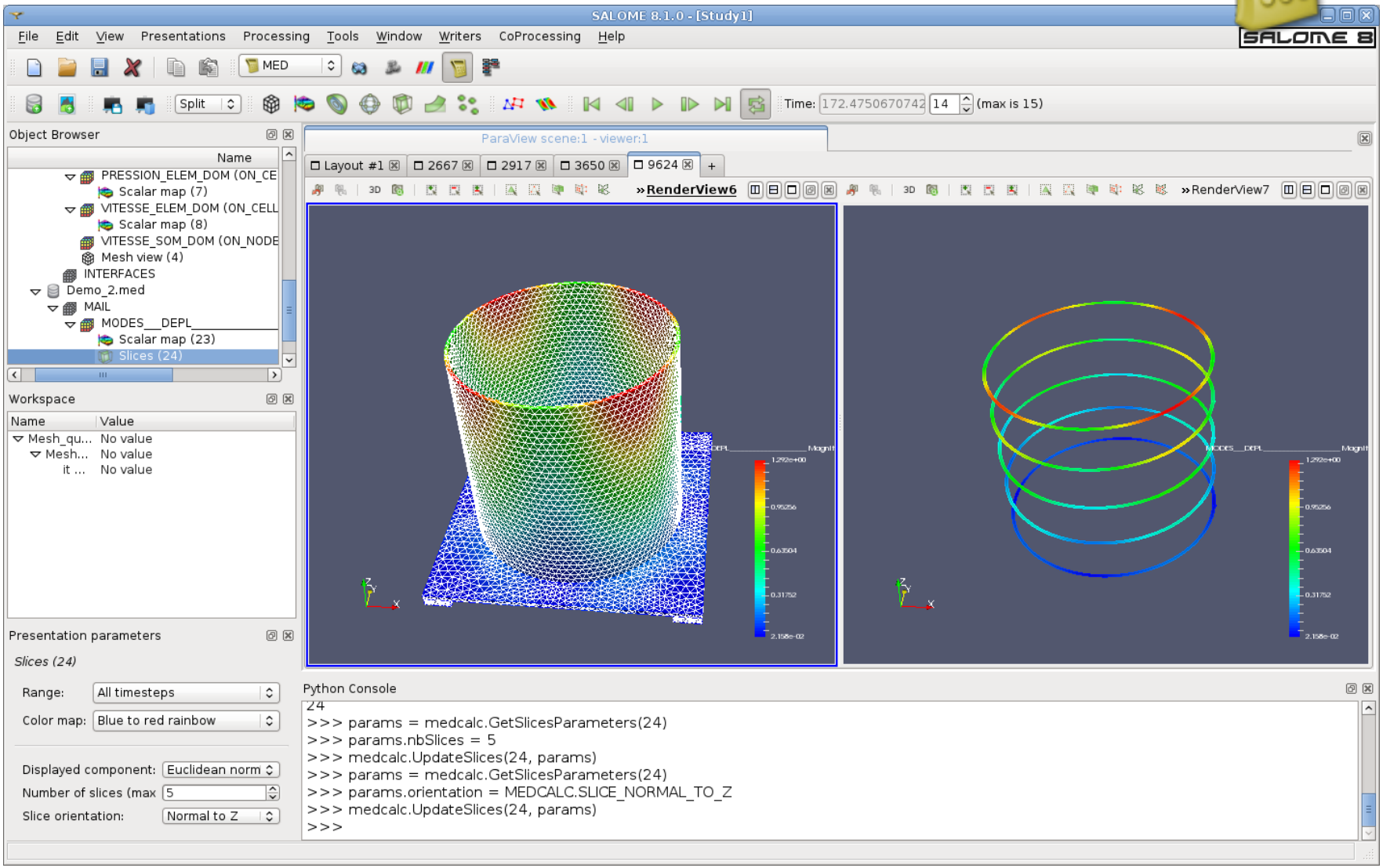
MEDCALC? A WORK IN PROGRESS

- **Load, process and visualize meshes and fields from the same SALOME module**
  - → Integrate PARAVIS viewers w/o exposing the full ParaView GUI
  - → Take advantage of the power of MEDCoupling and PARAVIS combination
  - → Provides arithmetic functions on fields

- **« Easy to use »**
  - GUI exposing essential functions with PARAVIS look and feel
  - A dedicated Python API: you can dump!

- **Finally, need advanced operations?**
  - Switch to PARAVIS in a second!
  - The pipeline is already there!

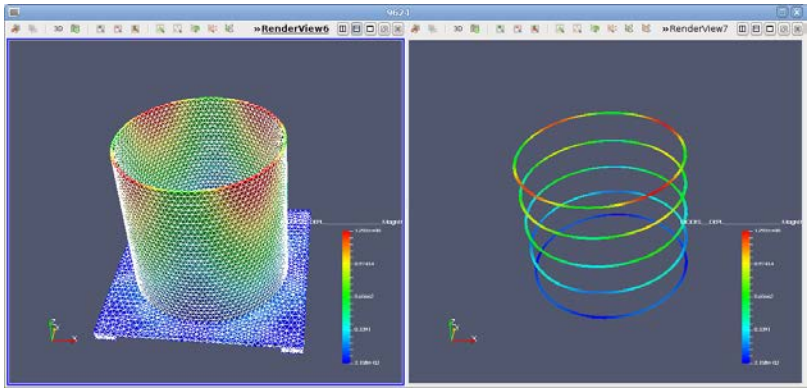
MEDCALC: GUI OVERVIEW



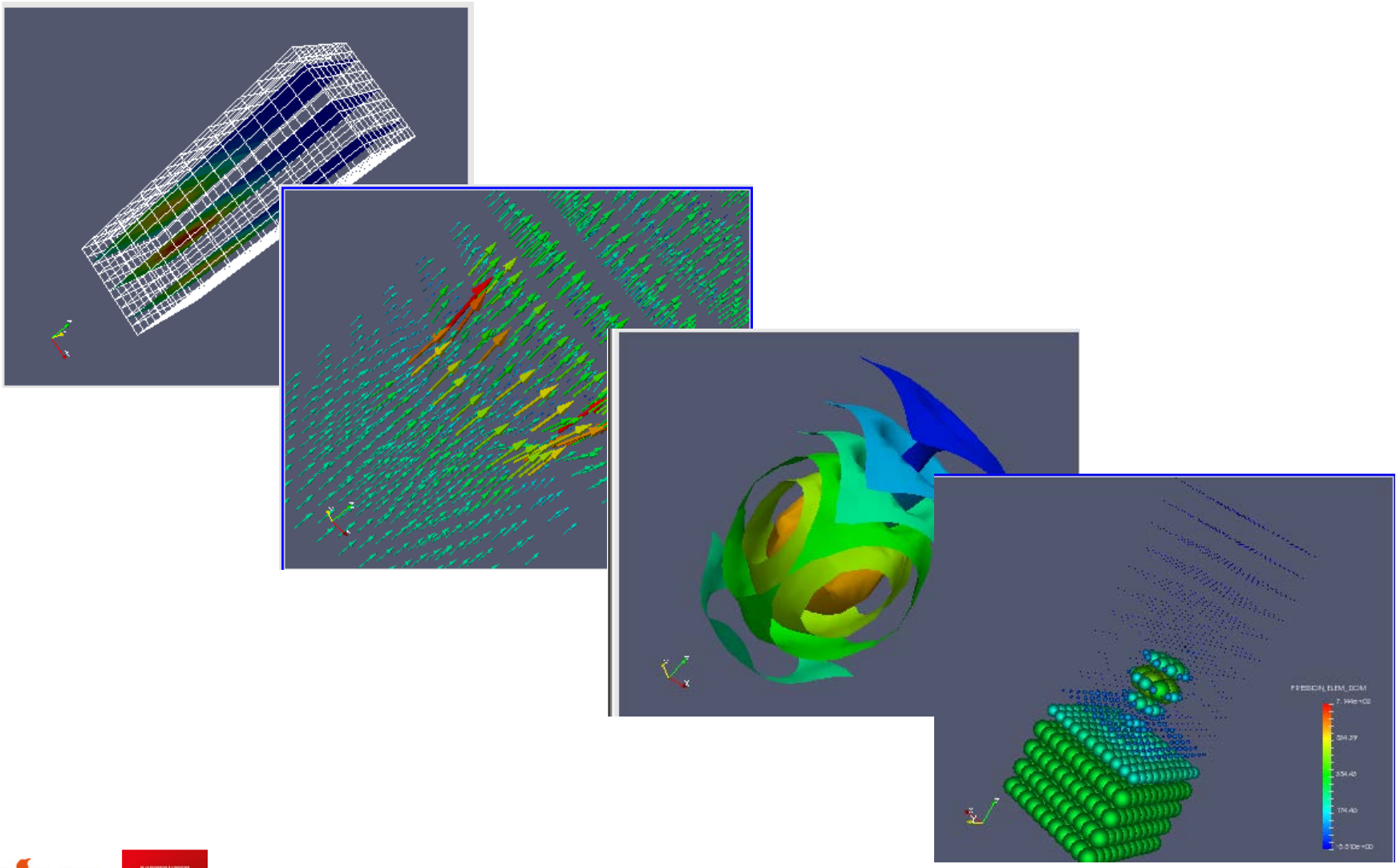
MEDCALC: PYTHON SAMPLE

```
>>> source_id = medcalc.LoadDataSource('/local00/home/F62173/testfiles/Demo_2.med')

>>> presentation_id = medcalc.MakeMeshView(medcalc.GetFirstMeshFromDataSource(source_id),
                                           viewMode=MEDCALC.VIEW_MODE_NEW_LAYOUT)

>>> presentation_id = medcalc.MakeScalarMap(accessField(80),
                                           viewMode=MEDCALC.VIEW_MODE_OVERLAP)

>>> presentation_id = medcalc.MakeSlices(accessField(80),
                                         viewMode=MEDCALC.VIEW_MODE_SPLIT_VIEW)

>>> params = medcalc.GetSlicesParameters(presentation_id)
>>> params.orientation = MEDCALC.SLICE_NORMAL_TO_Z
>>> params.nbSlices = 5
>>> medcalc.UpdateSlices(22, params)
```

Each GUI action triggers
a python function call



MEDCALC: PRESENTATIONS

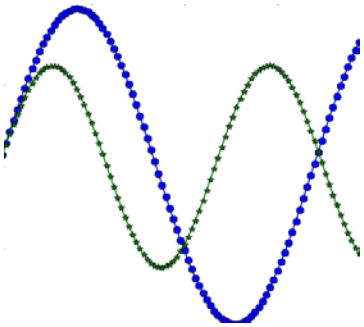


MEDCALC! THERE IS WORK TO BE DONE!

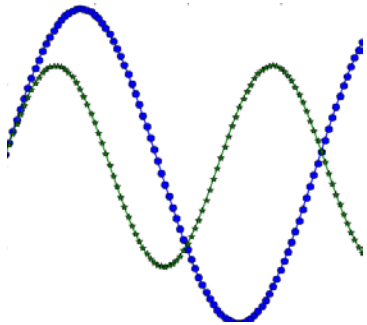
- **Still a prototype to be beta-tested (SALOME 8.2)**
  - Presentations with minimal settings
  - Modes and animations
  - Memory-passed objects
  - ParaView pipeline – switch to PARAVIS
  - GUI and scripting

- **Roadmap 2017**
  - Concept validation thanks to user feedback
  - Dataspace / workspace interaction
  - MEDCoupling visibility enhancement

CURVEPLOT



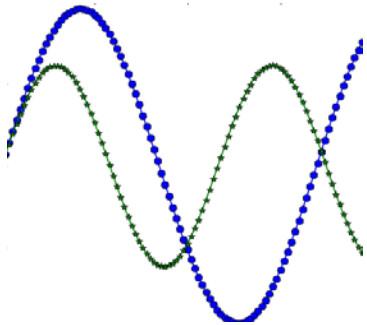
CURVEPLOT



Some history and background

- At CEA, several business projects use the “old” C++ Plot2D, based on Qwt
- Problem: some divergence/forks in the code!
  - New/modified functionalities/API
  - Not reversed in the main stream
- In general, current tool is not obvious to maintain and to change
- Hazardous mix between Qwt API and pure PLOT2D API
- From recent SALOME versions, we benefit from the *de-facto* integration of *matplotlib*
  - *matplotlib* provides an easy to manipulate API and most wanted services

SPECIFICATIONS

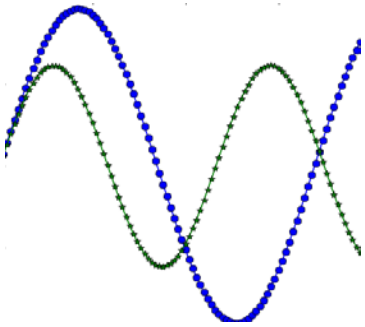


Main needs around a plotting tool

- Obviously, plotting 2D curves!
- No 3D: ParaVis is already there! And MED visualization is almost ready too.
- Curve related functionalities
  - Overlapping curves
  - Various representation modes (log, color, etc …)
- Why not use Paraview's capabilities ?
  - Not user-friendly enough for business module developer
  - Too heavy: requires ParaView loading!
  - Harder to interface with custom file formats (XML, .data, …)

Start minimal and grow only if needed! (similar to MEDCalc)

IMPLEMENTATION & API



Implementation

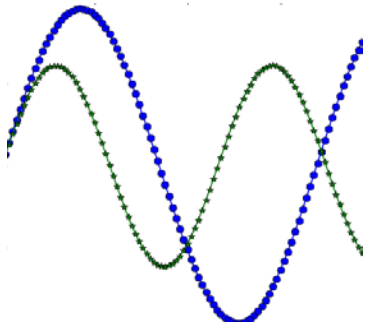
- Full Python (like *matplotlib*), with PyQt4 backend
  - With a C++ wrapping (using internal SALOME interpreter)

- Public API does not expose MVC design. Only simple commands
  - `AddCurve, DeleteCurve`
  - `AddPlotSet, DeletePlotSet`
  - `SetCurveMarker, SetXLog, SetCurveLabel`

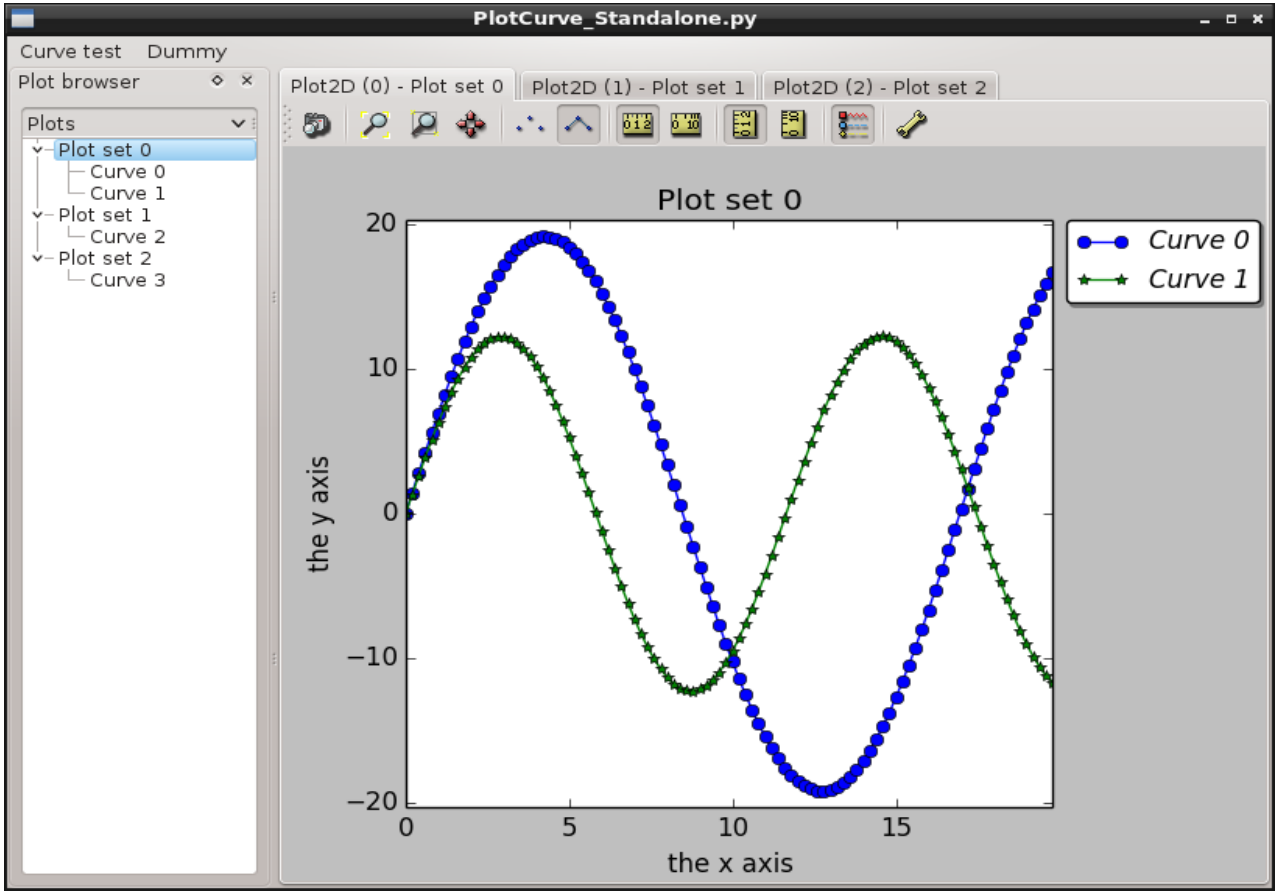
- Main parameters: unique curve identifier (=an integer), and a unique plot set identifier

- Testing part: ~45 unit tests with screenshot comparison

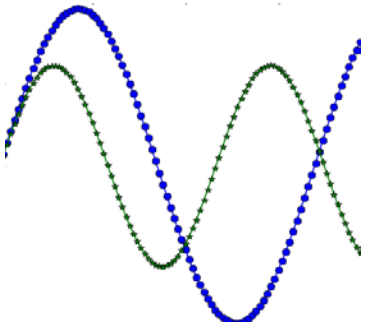
SOME SCREENSHOTS (1/2)

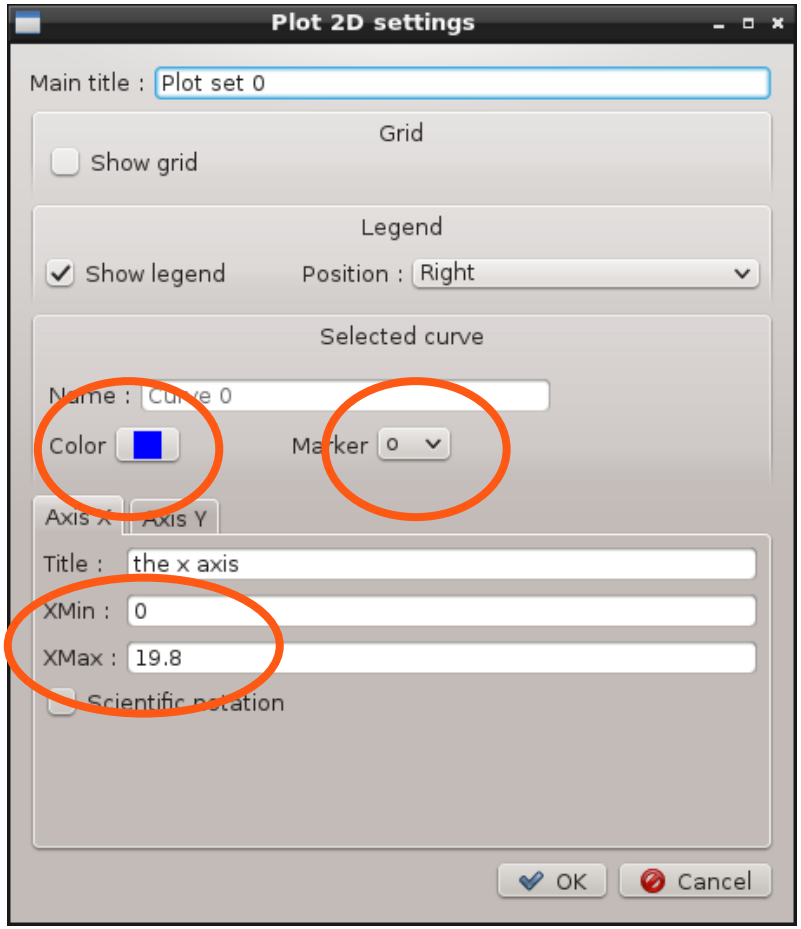


Illustrative standalone application

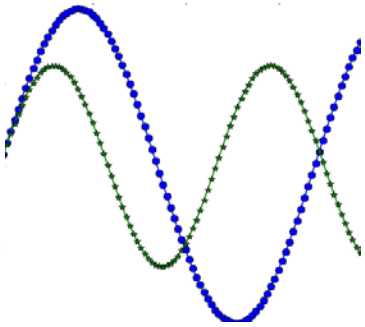


SOME SCREENSHOTS (2/2)





CONCLUSION & TODO LIST



Conclusion

- Simple interface to most commonly requested plotting facilities
- *matplotlib* completely encapsulated
- New functionalities: finer handling of curves (changing colors, markers, etc …)

On-going work

- Have a full dedicated SALOME module using the tool
  - Loading files in a table, etc …
  - Potentially create/modify columns, using PANDAS for example
  - And obviously plot the data loaded this way

- Major expected improvements
  - Double Y axis (was in former Plot2D and is needed at CEA)
  - Performance optimization: not as efficient as pure *matplotlib* yet