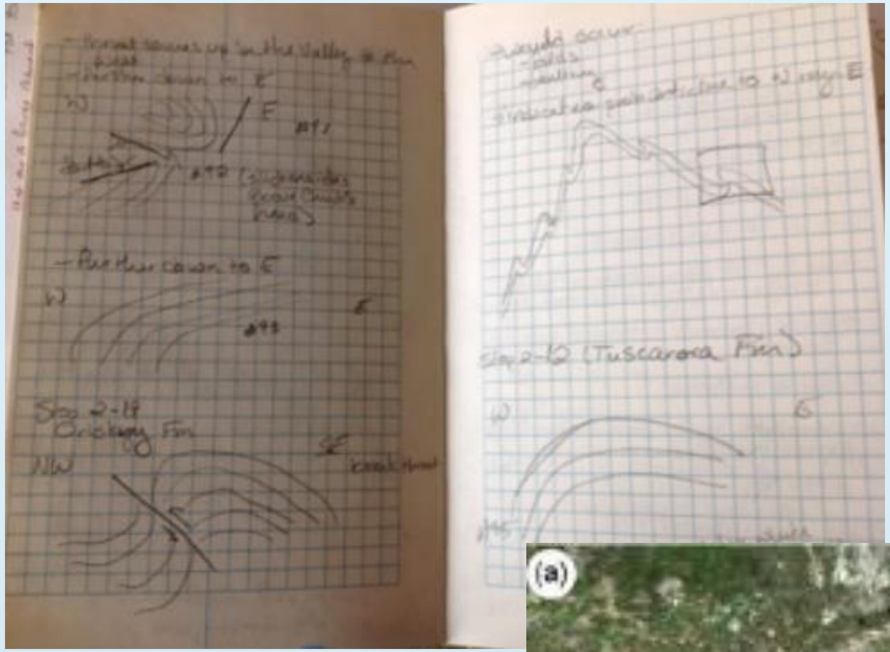
Towards an Open-Source Workflow for Geological Mapping

Kristiaan Joseph; Thierry Martin; Olivier Kaufmann

UMONS

24/10/19

MOTIVATION



Seamless workflow

Conceptual/Visual understanding

Address “What-if?” scenarios

Efficiency in the field

https://sp.lyellcollection.org/content/459/1/157/tab-figures-data

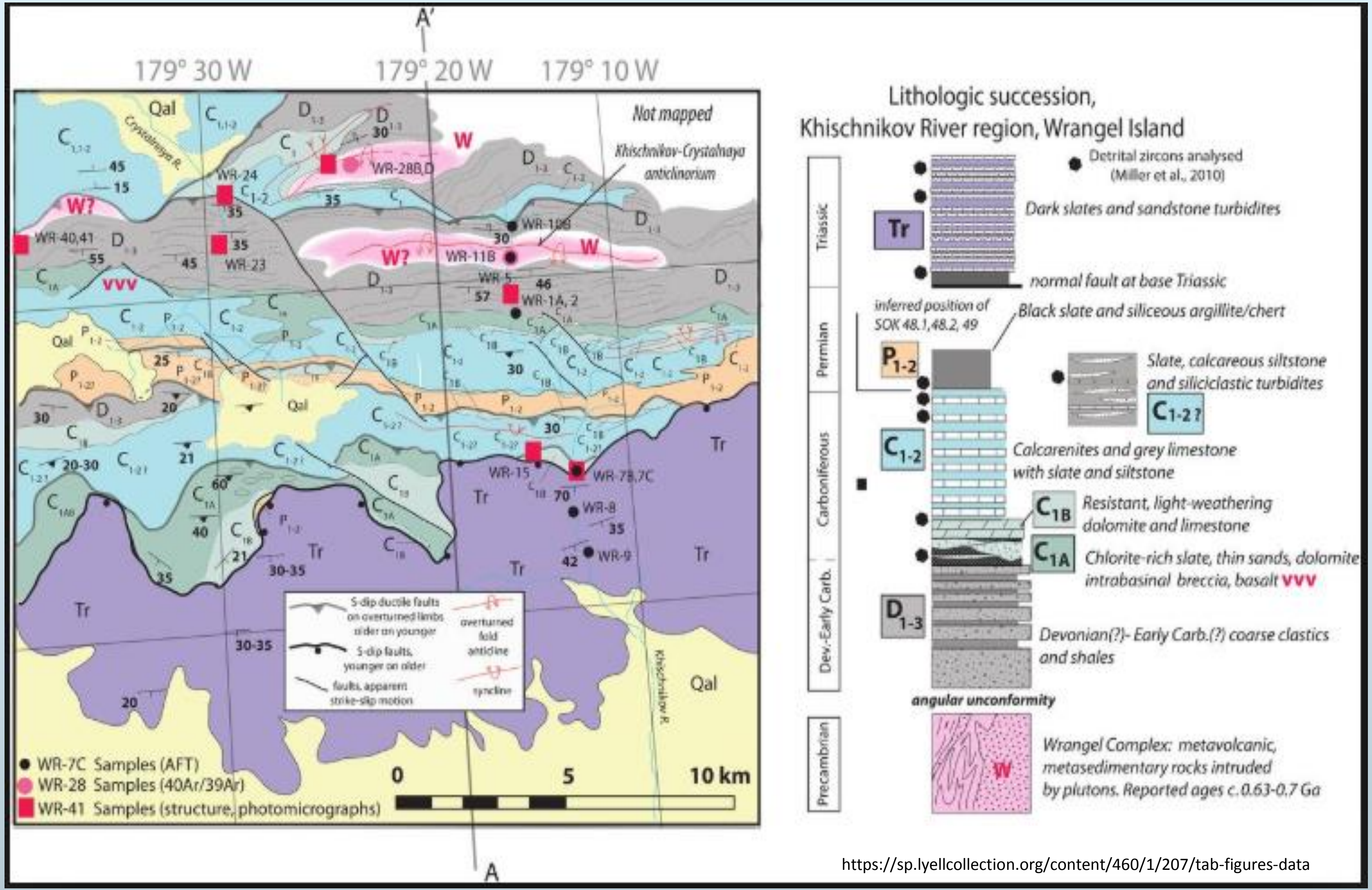
AIM

An ***open-source, integrated*** geological workflow to assess and quantify uncertainty utilizing new/updated inputs and decision factors to create models and maps

GOALS

Produce geologically credible 2D maps in a known setting

- Sedimentary environment
- Stratigraphic column
- Data at known points



https://sp.lyellcollection.org/content/460/1/207/tab-figures-data

Workflow

Data Collection

3D Modelling Inputs

Produce 3D model (Potential Field Method)

Derive 2D map

Output Map(s)

Tools

Jupyter notebooks

Matplotlib(2D)

GemPy

Paraview (VTK)

PyMesh, Gmsh, Meshio

3D Modelling Inputs

Plane orientation normals

Adherence to Sequence Pile

Scalar Field

Block Model

Foliations Dataframe:

In [11]:
```
gp.get_data(geo_data, 'orientations')
```

	X	Y	Z	G_x	G_y	G_z	dip	azimuth	polarity	formation	series	formation_number	order_series	isFault
0	621055.284454	2.476424e+06	386.818182	0.648785	-0.061502	0.758482	40.669425	95.415167	1	Unit3	Strat_Series	2	1	False
1	620946.810298	2.477202e+06	422.923077	0.466871	0.226770	0.854755	31.267317	64.093066	1	Unit3	Strat_Series	2	1	False
2	620653.973690	2.477877e+06	400.820896	0.234040	0.326523	0.915755	23.686789	35.631571	1	Unit3	Strat_Series	2	1	False
3	615165.009653	2.477880e+06	430.000000	-0.432540	0.210290	0.876748	28.747494	295.927864	1	Unit3	Strat_Series	2	1	False
4	615490.574975	2.478291e+06	435.000000	-0.431822	0.206362	0.878035	28.593829	295.542510	1	Unit3	Strat_Series	2	1	False
5	614562.705647	2.476466e+06	482.600000	-0.636122	0.285388	0.716870	44.203331	294.162820	1	Unit3	Strat_Series	2	1	False
6	614128.758702	2.475672e+06	475.375000	-0.480245	0.298546	0.824764	34.435457	301.867328	1	Unit3	Strat_Series	2	1	False
7	618954.185116	2.474840e+06	365.833333	0.292640	-0.279519	0.914456	23.871344	133.686261	1	Unit3	Strat_Series	2	1	False
8	619535.625878	2.476016e+06	491.125000	0.305975	-0.081305	0.948561	18.457026	104.880962	1	Unit3	Strat_Series	2	1	False
9	616406.887377	2.478867e+06	431.000000	-0.098850	0.441886	0.891608	26.923954	347.390550	1	Unit3	Strat_Series	2	1	False
0	616103.856828	2.477444e+06	625.333333	-0.406196	0.300498	0.862964	30.349005	306.493491	1	Unit3	Strat_Series	2	1	False

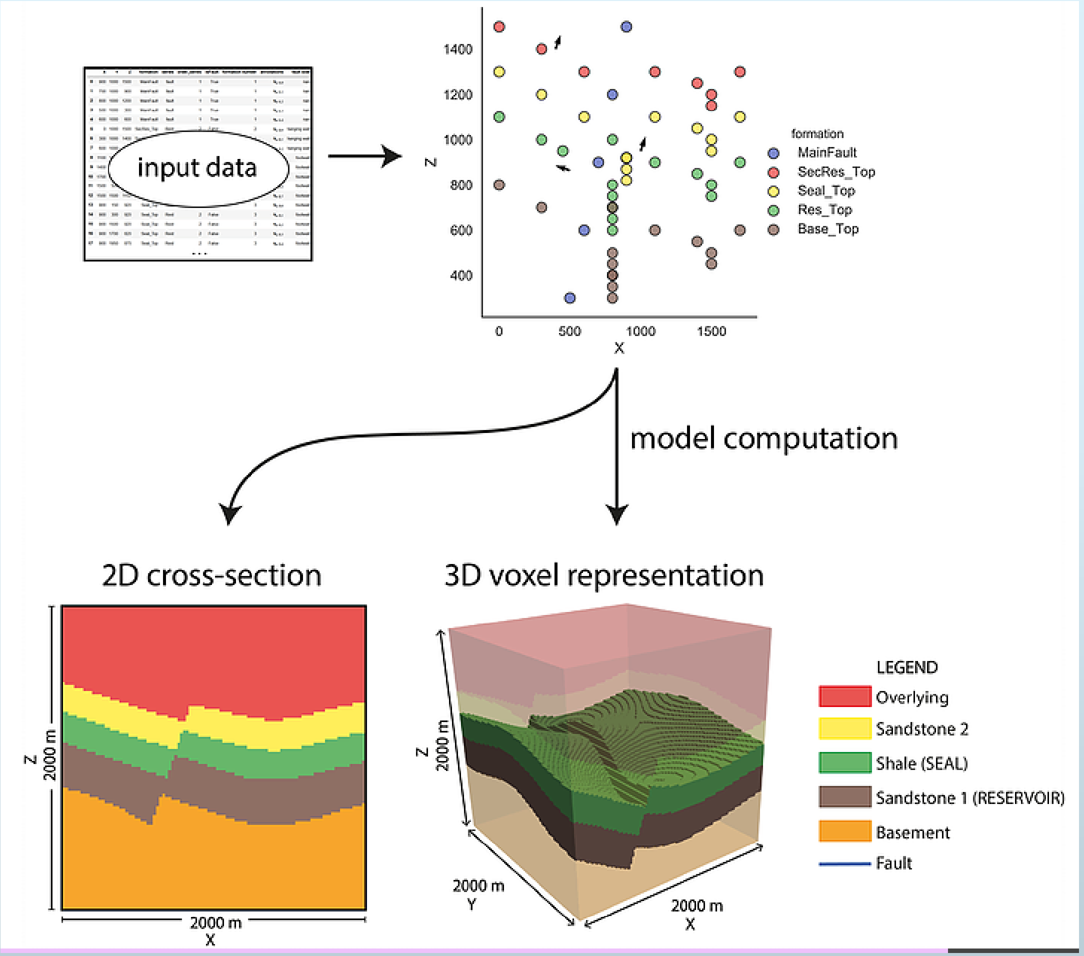
3D Geological Model

GemPy : Python-based 3-D structural geological modeling package

Implicit modelling based on the potential field method

Utilizes Theano for calculations

Topography can be incorporated



Gempy.org

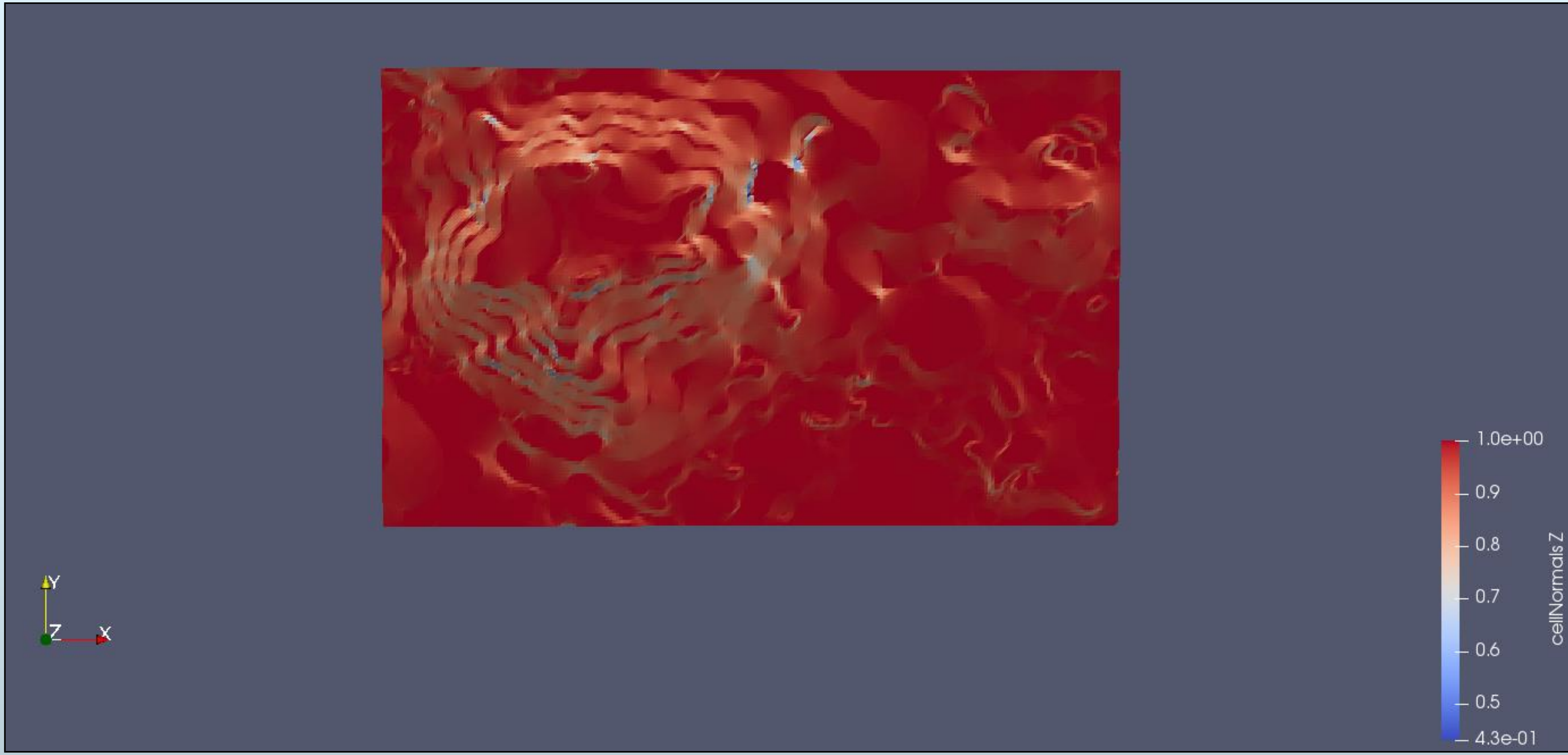
ParaView

- Open-source, data analysis and visualization tool (VTK)
- Build visualizations for qualitative and quantitative analysis
- Interactive 3D data manipulation
- Analyze extremely large datasets

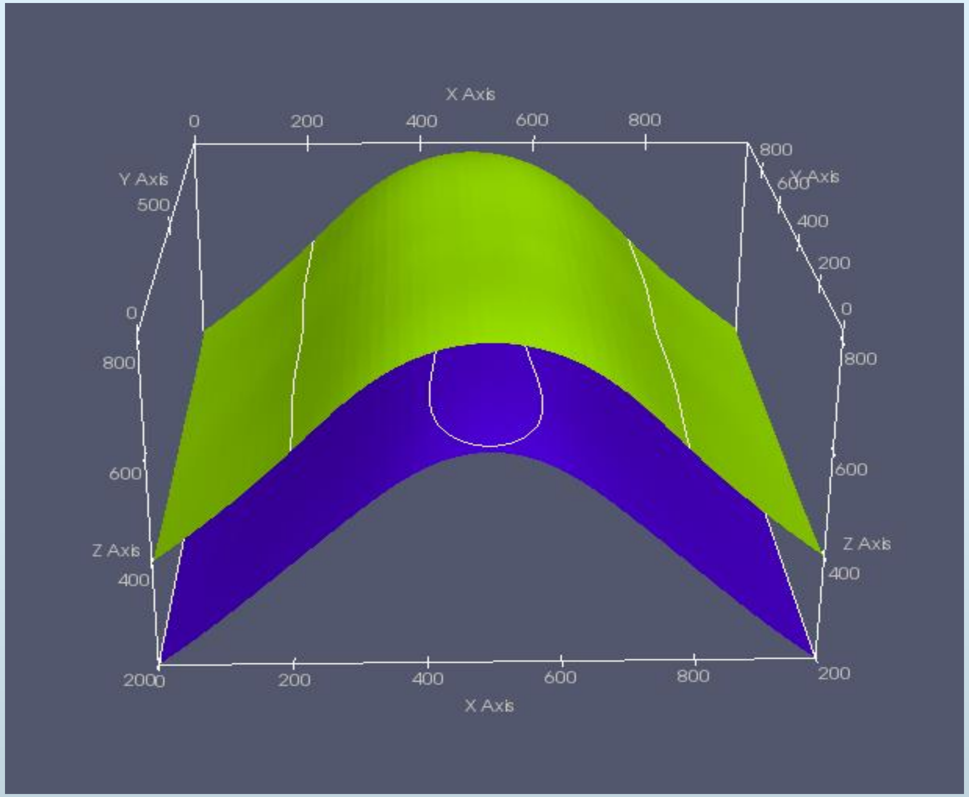
DEM

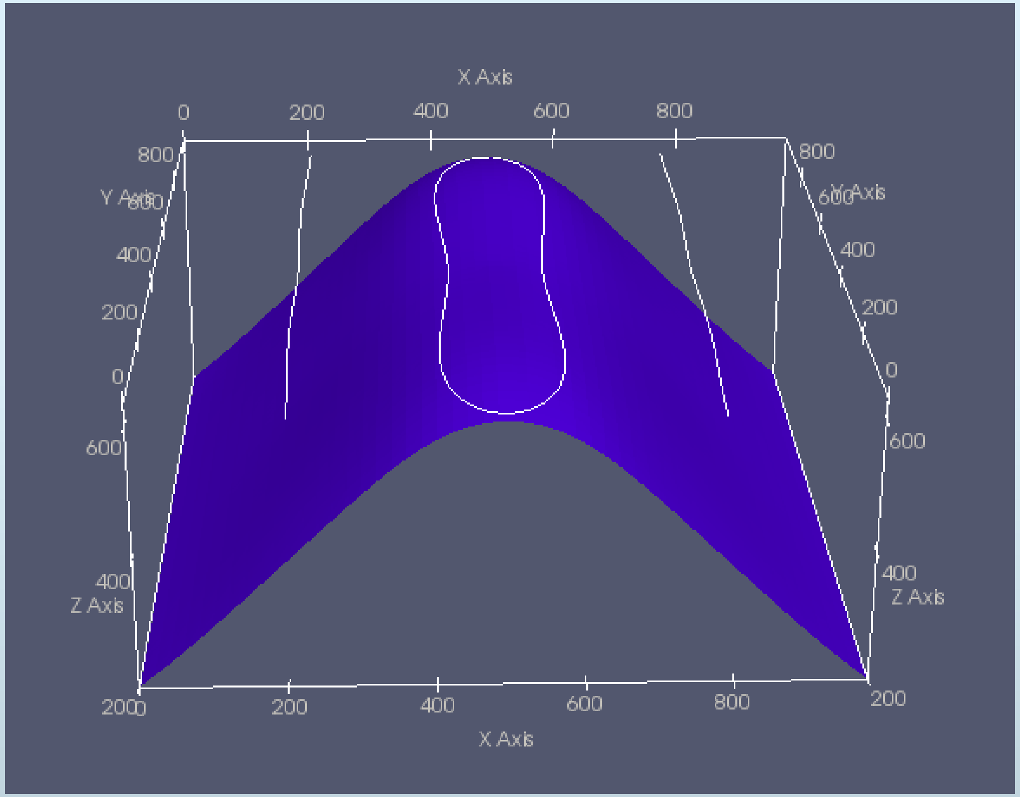
Point set → Closed surface with varying Z values (Gmsh)

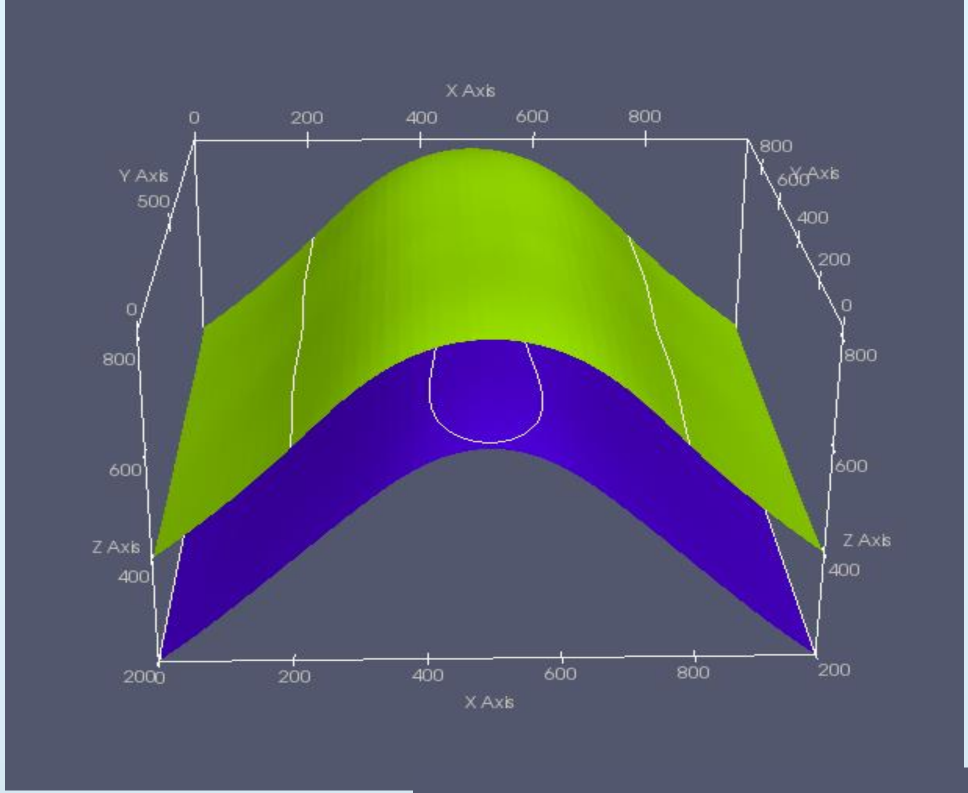
*Not yet included in GemPy Model

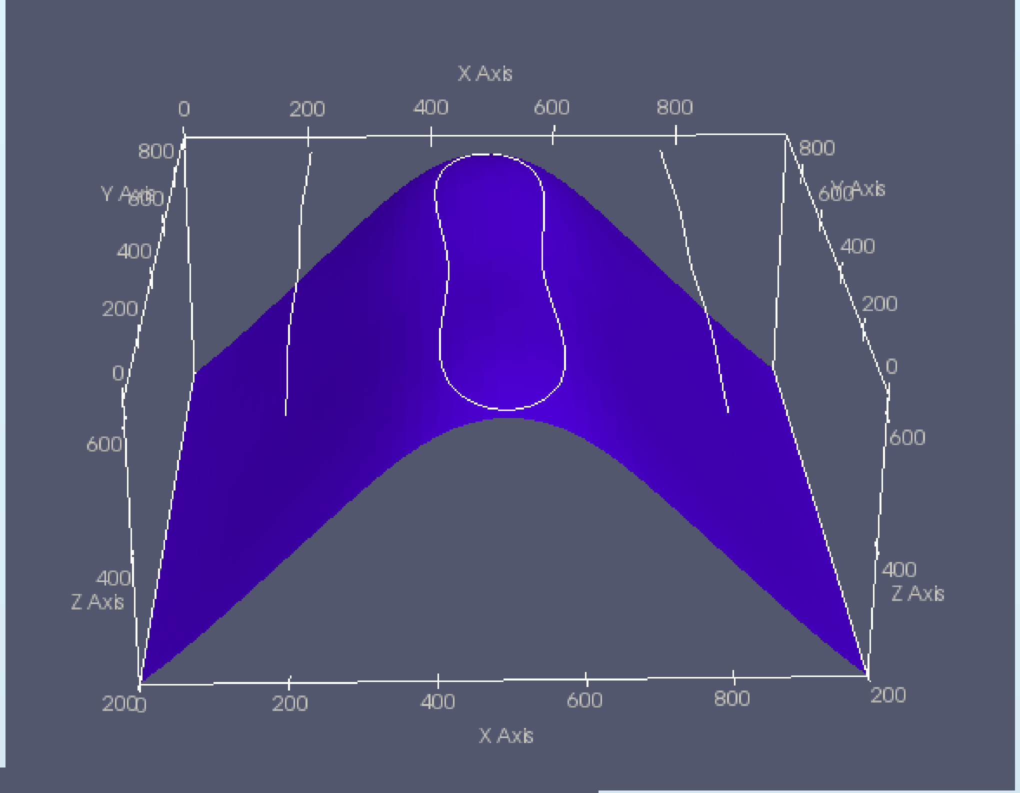


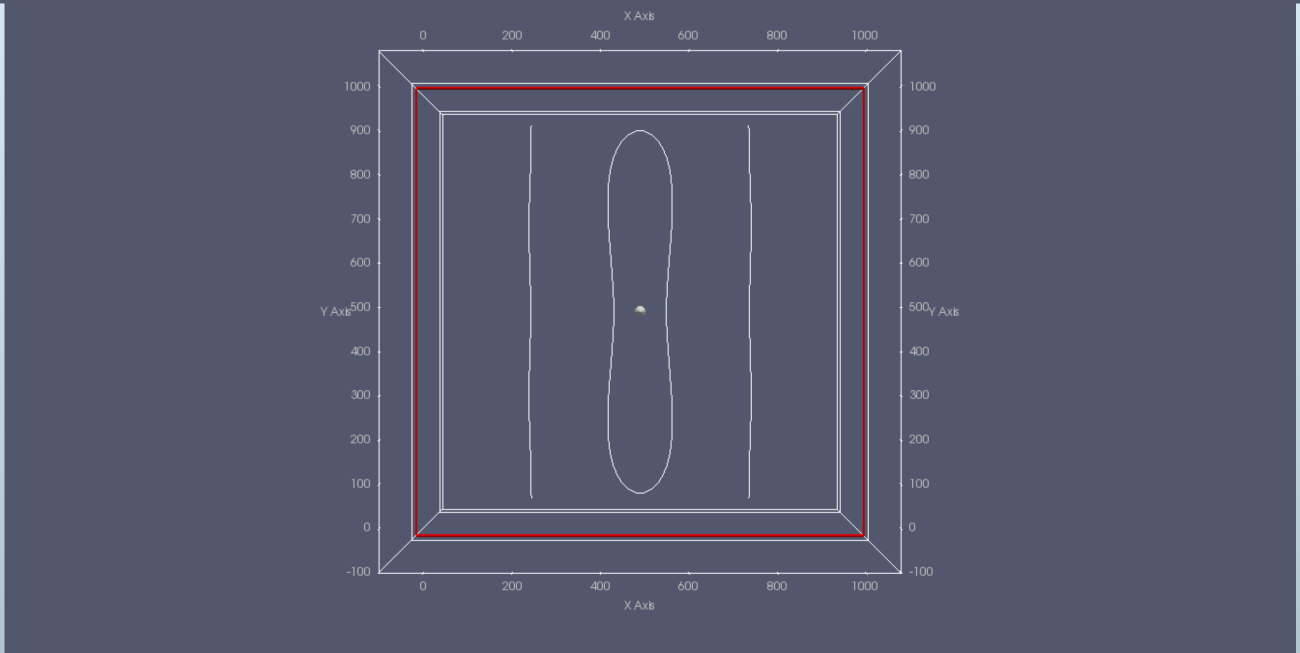
GemPy Model of a simple fold of 2 lithologies displayed in ParaView











White lines
are contours
at Z= 606

Outputs (in-progress)

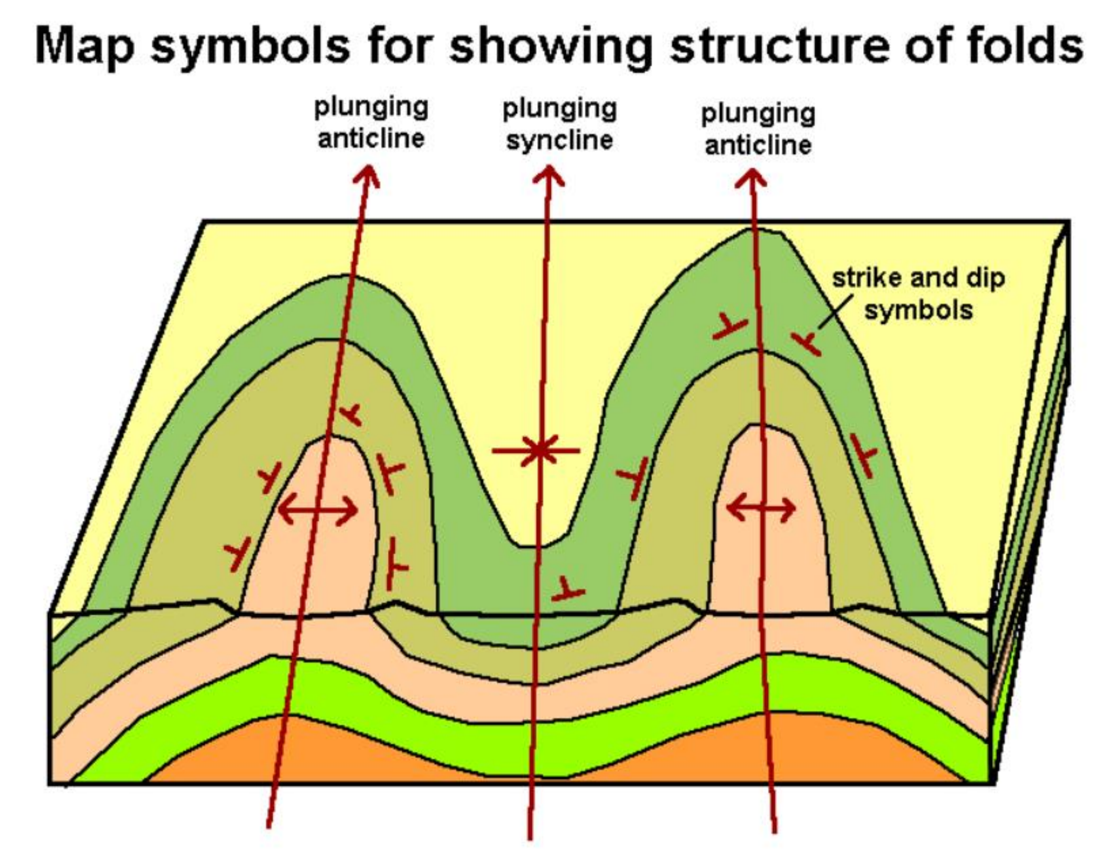
Geological Map

Structure Symbols

Contour Lines

2D Plane intersecting 3D Surface

- PyMesh vs VTK



```python
import meshio
```

```python
SURFACES_FOLD1= meshio.read('/home/kristiaan/SURFACES_FOLD1.vtk')
SURFACES_FOLD2= meshio.read('/home/kristiaan/SURFACES_FOLD2.vtk')
MAP_FOLD= meshio.read('/home/kristiaan/MAP_FOLD.vtk')
```

```python
meshio.write('/home/kristiaan/SURFACES_FOLD1.off', SURFACES_FOLD1)
meshio.write('/home/kristiaan/SURFACES_FOLD2.off', SURFACES_FOLD2)
meshio.write('/home/kristiaan/MAP_FOLD.off', MAP_FOLD)
```

2D Plane intersects 3D Surface

VTK → Meshio → PyMesh

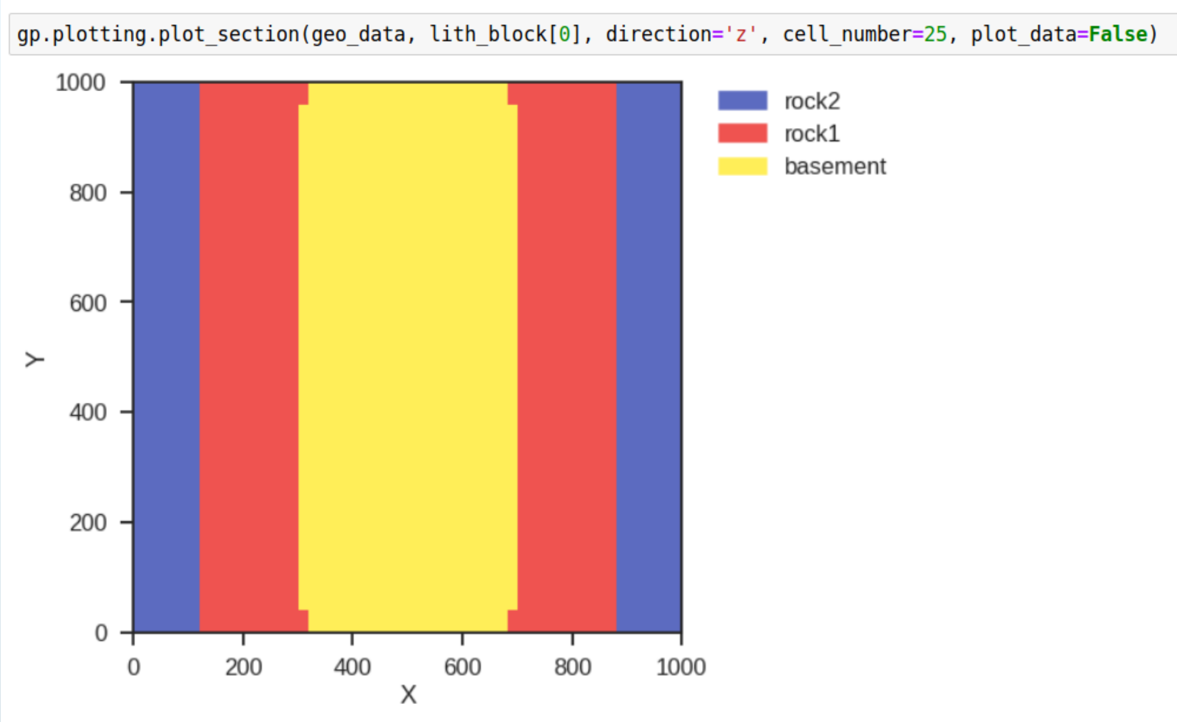
DOCKER

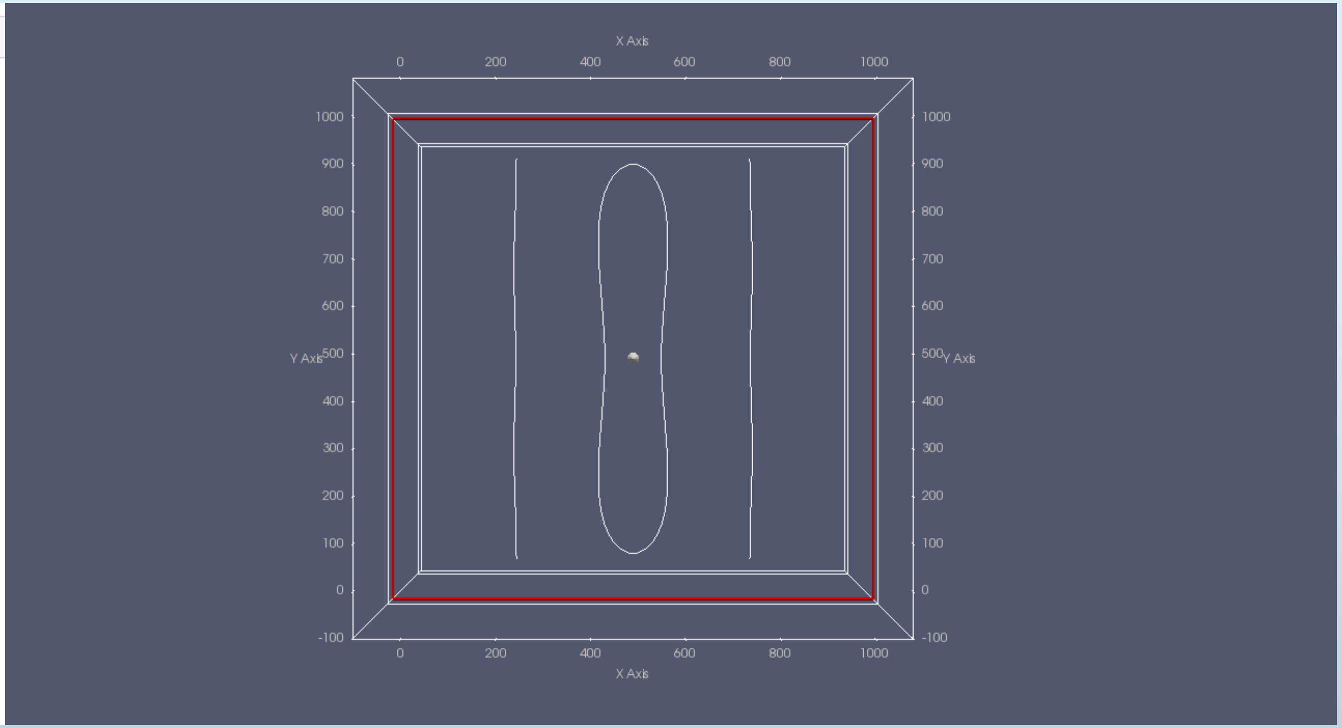
```python
>>> import os
>>> os.listdir('/')

>>> import pymesh
>>> S1 = pymesh.load_mesh('/SURFACES_FOLD1.off')
>>> MF = pymesh.load_mesh('/MAP_FOLD.off')
>>> output_mesh = pymesh.boolean(S1, MF operation = 'intersection', engine='igl')
>>> pymesh.save_mesh('/intersection.off', output_mesh)
```

Map View: GemPy

Map View: ParaView





Bigger Picture: The Iterative Workflow

- Model regeneration and Mapping
  - Streamlined process (eg, input formats)
  - Efficient computation times
- Uncertainty Analysis
  - Detect and reduce errors
  - Identify risks
- Select geological realistic scenarios

QUESTIONS?

Theano

Relies on symbolical graphs which represent mathematical expressions

Each individual method corresponds to part of this graph

Benefits:

- Direct execution on GPUs
- Capacity to efficiently compute gradients
- Provides optimized compiled code
- Capability to perform automatic differentiation (AD)

de la Varga *et al.*, 2019

Potential Field Method

Proposed by LaJaunie (1997); Basis for modelling in GemPy

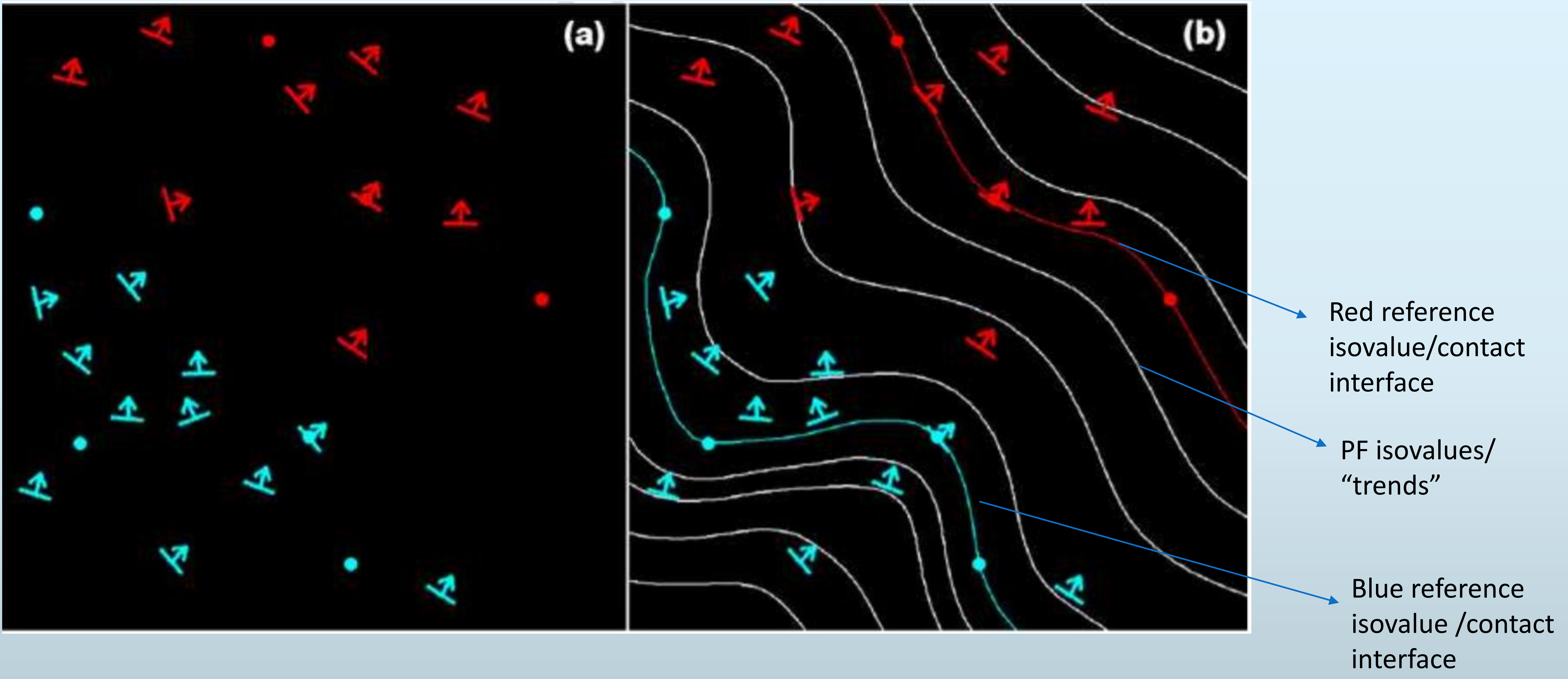
The Potential Field (PF) - A geological interface (isosurface) of a scalar field defined in the 3D space (Chiles *et al, 2007)*

PF Method's weighted interpolation function is based on Universal Cokriging

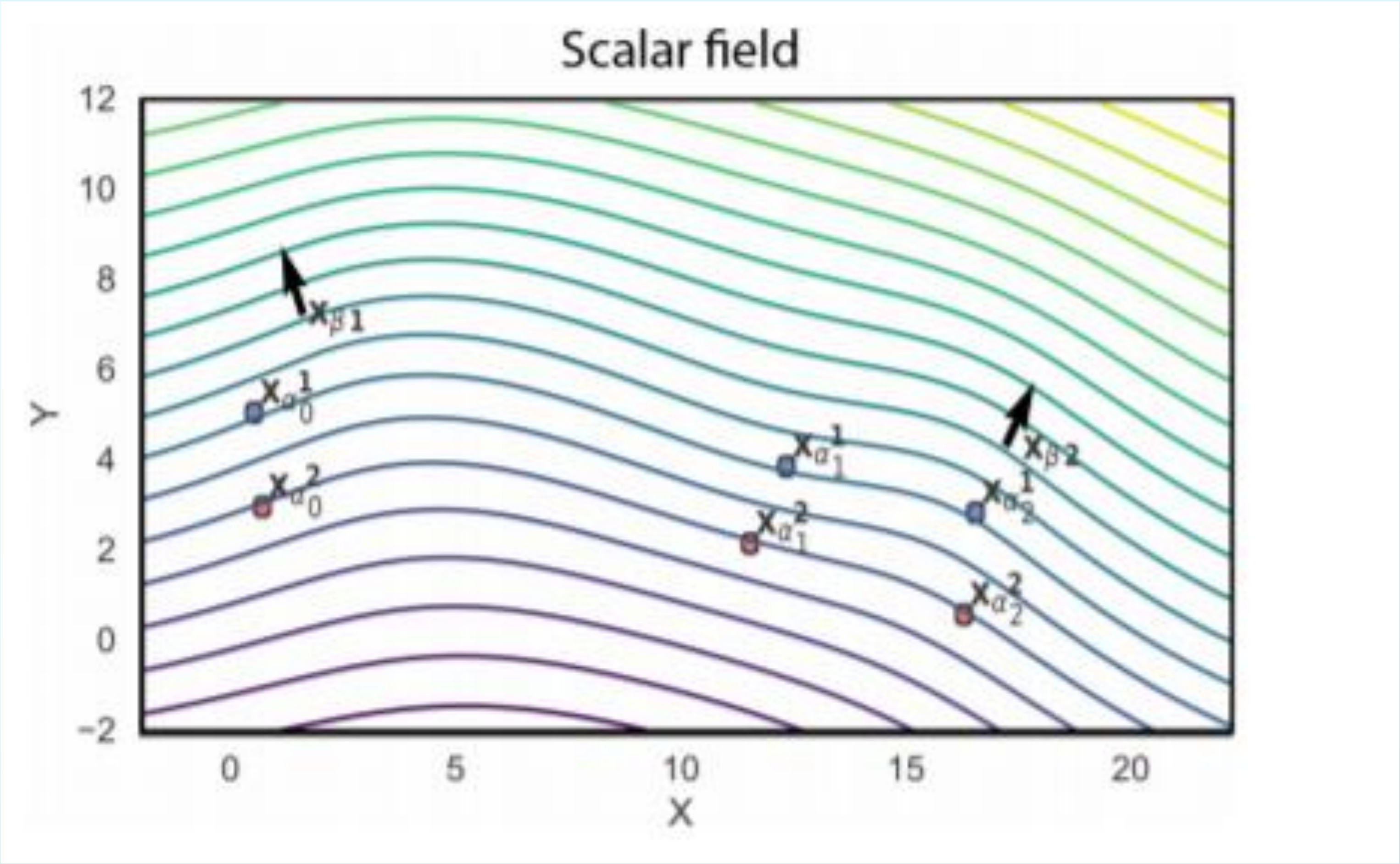
- (de la Varga *et al., 2019;* Chiles *et al., 2014;* LaJaunie *et al., 1997)*

Universal Kriging is appropriate for capturing trends using polynomial drift functions

Cokriging for combining potential and gradient values



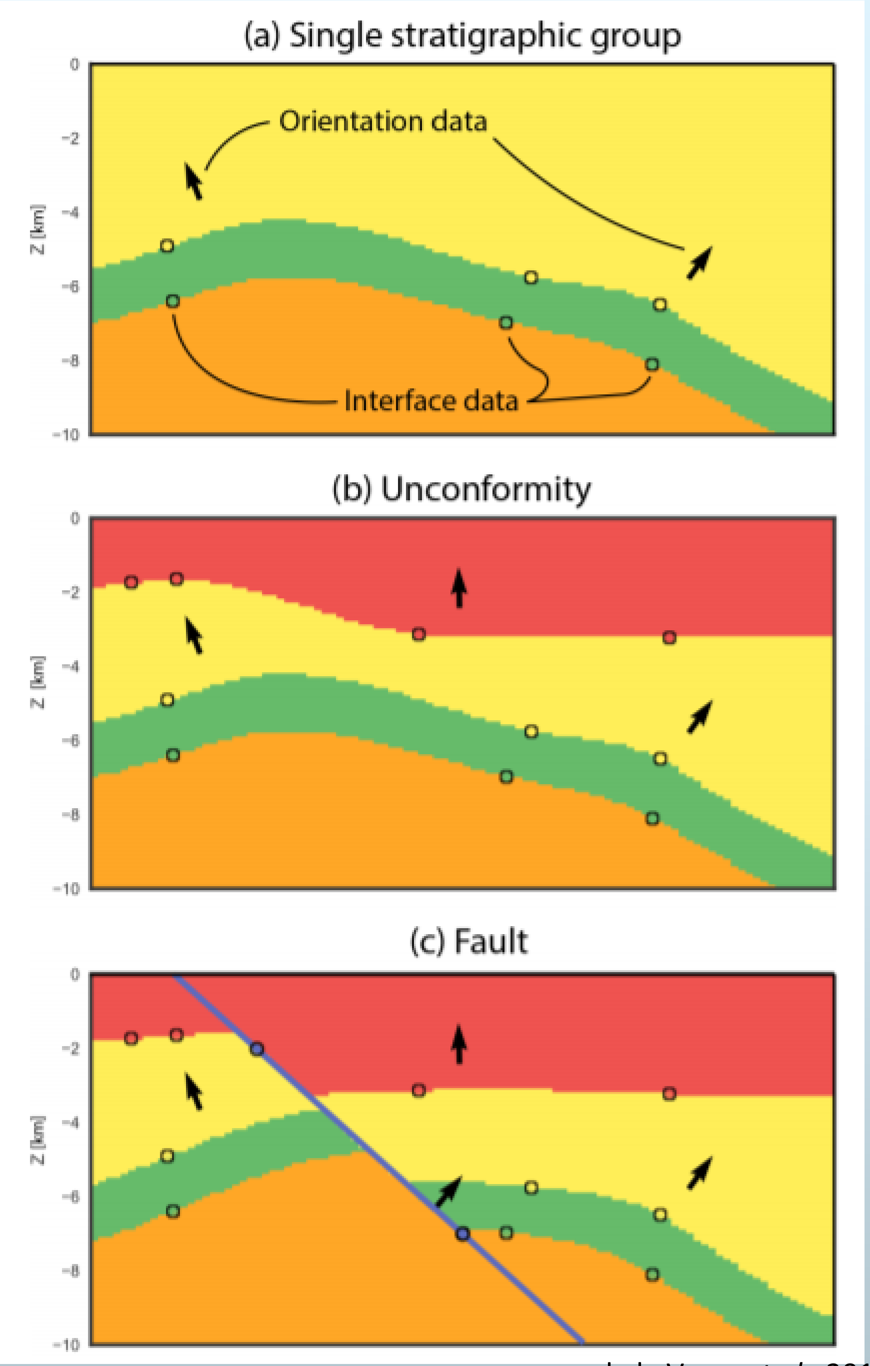
Potential Field Interpolation Method (Calcagno *et al., 2008)*

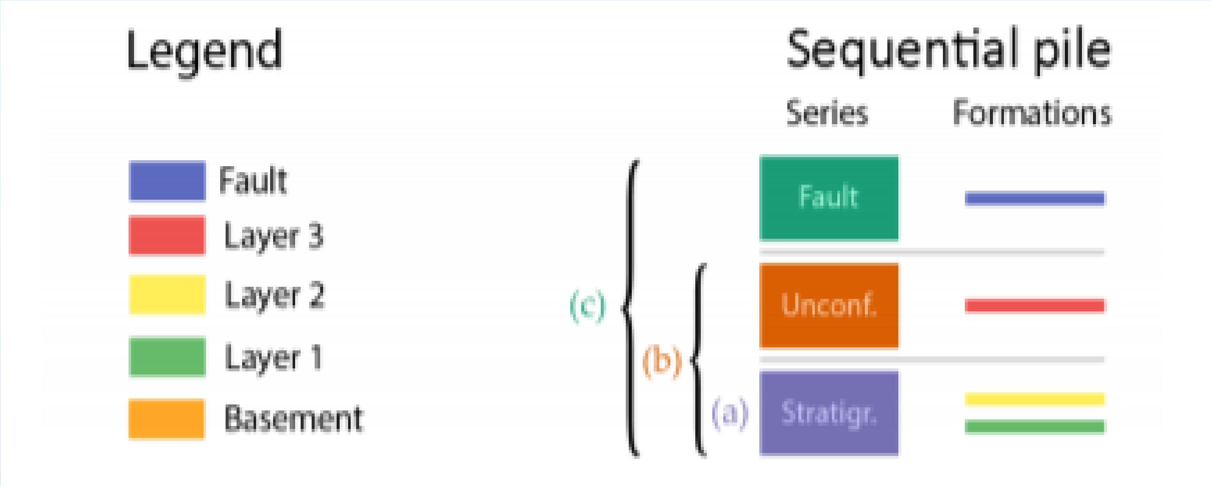


6 points – 3 in the red layer, 3 in the blue layer, Orientation indicated by arrows, Lines are isosurfaces, (de la Varga *et al., 2019)*

$$\boxed{T^{*}(\mathbf{x})-T^{*}(\mathbf{x}_0)}=\boxed{\sum_{\alpha=1}^{M}\mu_{\alpha}\left(T(\mathbf{x}_{\alpha})-T(\mathbf{x}'_{\alpha})\right)}+\boxed{\sum_{\beta=1}^{N}v_{\beta}\frac{\partial T}{\partial u_{\beta}}(\mathbf{x}_{\beta})}$$

Scalar fields | Weighted Spatial Parameters | Weighted Orientation Parameters





de la Varga *et al.*, 2019