Towards Combining Numerical and Symbolic Computing in Python

L. Gross, L. Gao, J. Fenwick, C. Altinay
The University of Queensland, School of Earth Sciences
Earth Systems Science Computational Center, St Lucia QLD 4072, Australia

Goal: Software toolbox to build and run joint inversions
First Steps: Symbolic generation and manipulation of misfit functions with linking to numerical solution techniques
Case Study: Minimization problems constraint by a partial differential equation (PDE)

Generic Inversion Problem

$$\min_{u,r} \int_\Omega H\left(u, u_{,i}, r, r_{,i}\right) dx$$

u solution for parameter *r*:

$$-X_{i,i} + Y = 0$$
$$X, Y = f\left(u, u_{,i}, r\right)$$

Variation and Linearization

for $\int_\Omega \overline{H}\,(U) dx$

with $U = (r, u, \lambda)$

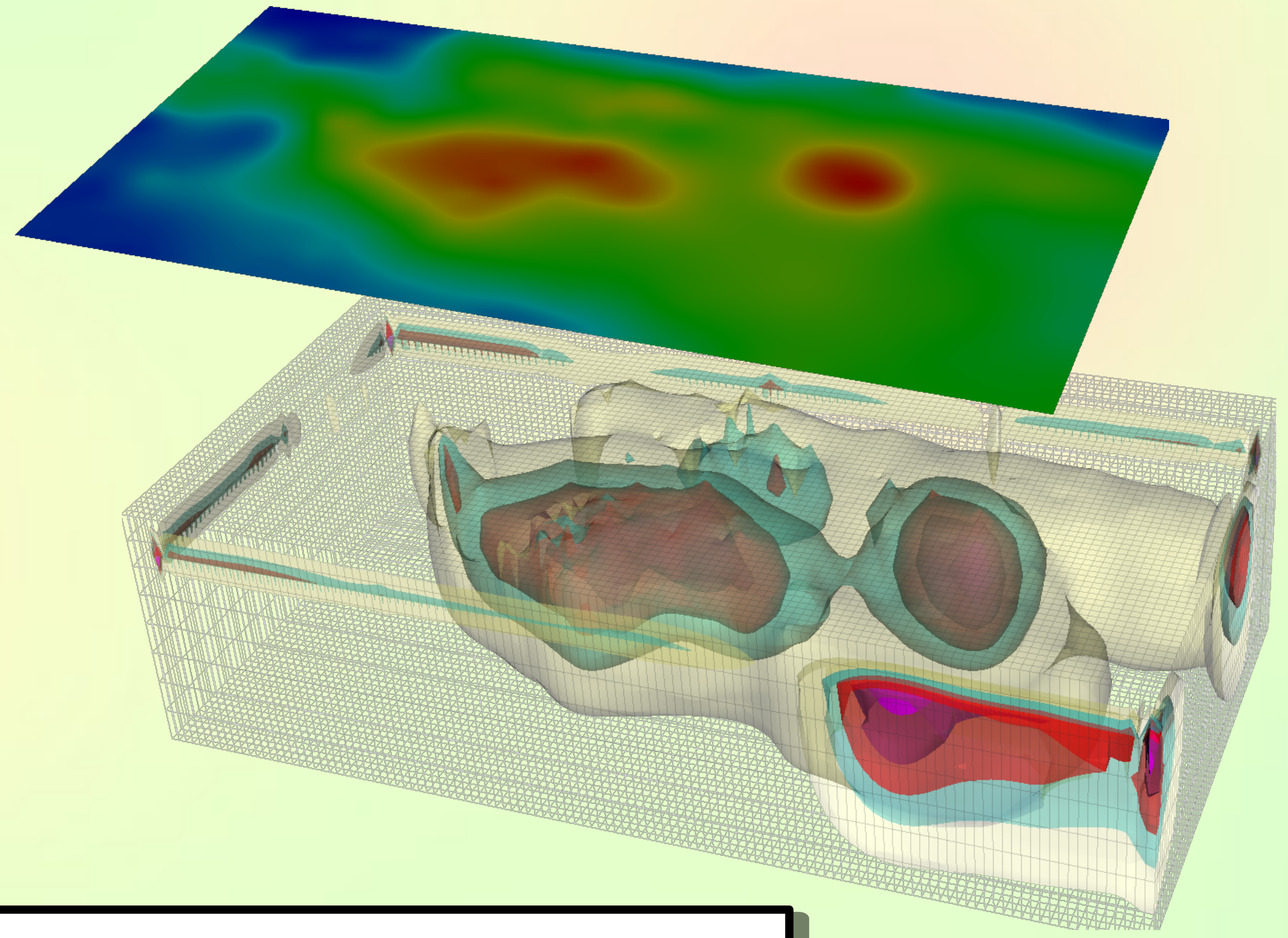
$$\overline{H} = H + \lambda_{,i} X_i + \lambda Y$$

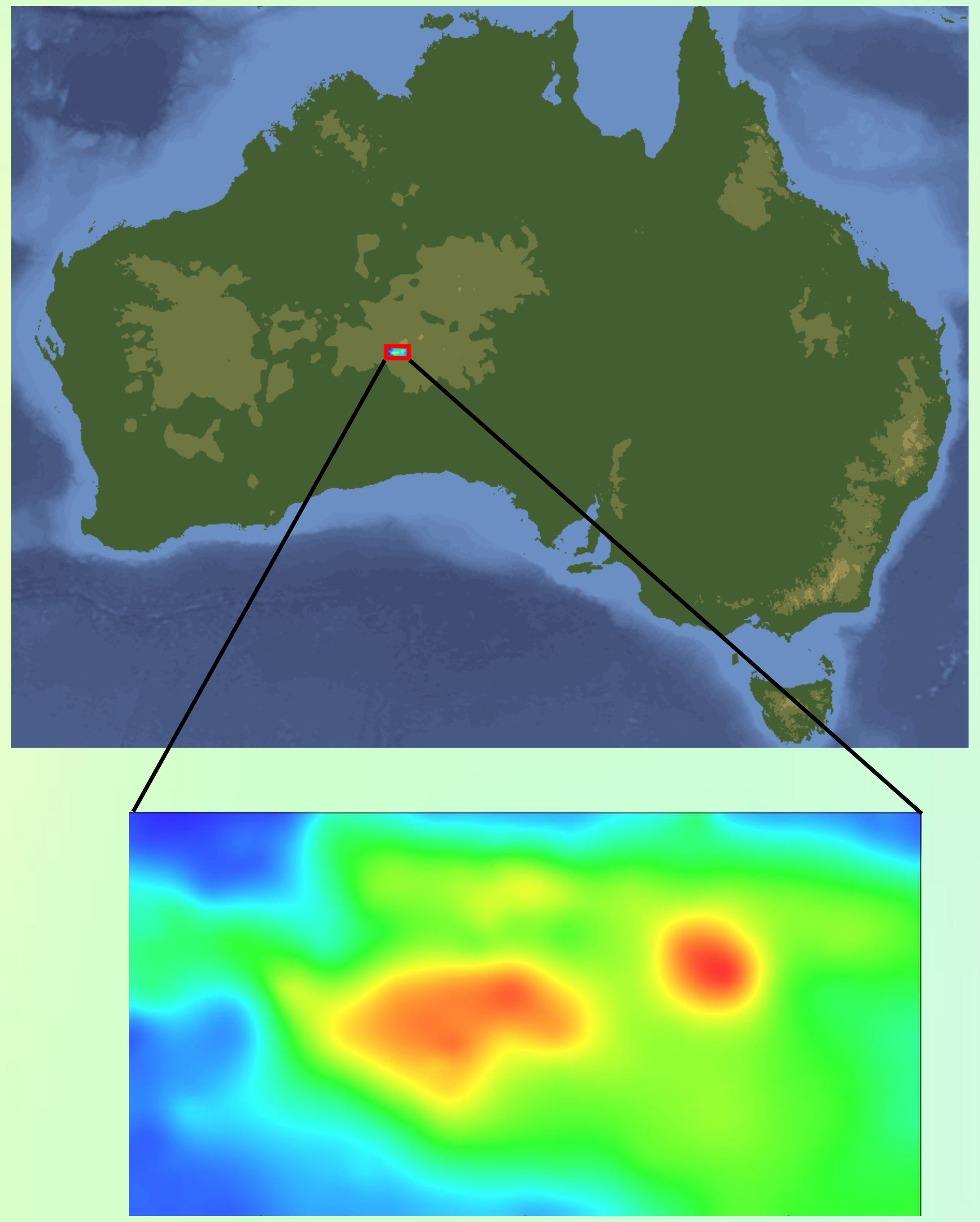
System of linear PDEs for *U*

$$-\left(\frac{\partial^2 \overline{H}}{\partial u_{i,j} \partial u_{k,l}} u_{k,l} + \frac{\partial \overline{H}}{\partial u_{i,j}}\right)_{,j} + \frac{\partial \overline{H}}{\partial u_i \partial u_k} u_k + \frac{\partial \overline{H}}{\partial u_i} = 0$$

Gravity Inversion:

$$\min_{\substack{u,r \\ u_{,ii}=4\pi r}} \int_\Omega h \cdot \left(u_{,2} - g\right)^2 + \|r_{,i}\|^a dx$$





```
dom=Brick(...) # generate FEM mesh
r,u=symbols("r,u")
g=Scalar(..., dom)  # gravity data
v=VariationalProblem(dom, u, r)
v.setValues(X=grad(u), \
    Y=2*pi*r, \
    H=h*(grad(u)[2]-g)**2 \
      +length(grad(r))**a)
r, u = v.getSolution()
```

- Symbolic generation of linear PDE for escript (with sympy)
- Substitute symbols → numerical representation

Numerical solution of linear PDEs using escript

- Finite Element Method
- Performs evaluation of PDE coefficients
- C/C++
- Python interface
- https://launchpad.net/escript-finley

returns numerical representation of solution *u* and parameter *r*

Acknowledgement: Supported by the Australian Geophysical Observing System EIF (AGOS)

EGU2012-3979
ESSI1.4, XL140