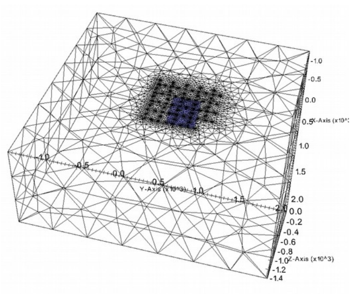
The University of Queensland
School of Earth and Environmental Sciences

Geophysical Modeling using esys-escript

Lutz Gross
E-mail : l.gross@uq.edu.au

esys-escript

Geometry



Mathematical Model

$$F=-A\nabla u-Bu+X$$
$$\nabla^t F+C\nabla u+Du=Y$$

esys-escript: scripted Model Implementation

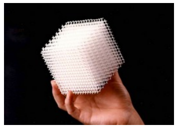
```
from esys.escript import *
import esys.escript.unitsSI as U
Q=10*U.W/U.m**3; K=1.7*U.W/(U.m*U.K); rhocp=1.5*U.Mega*U.J/(U.m**3*U.K)
# … time step size:
dt=1.*U.year; n_end=100
# … 40 x 20 grid on 10km x 5km
mydomain=Rectangle(40, 20,l0=10*U.km, l1=5*U.km)
x=mydomain.getX()
# … create PDE and set coefficients:
mypde=LinearPDE(domain)
mypde.setValue(A=K*kronecker(mydomain), D=rhocp/dt, q=whereZero(x[1]-
5*U.km))
# … initial temperature is a vertical, linear profile:
T=0*U.Celsius+30*U.K/U.km*(5*U.km-x[1])
n=0
while n<n_end :
   mypde.setValue(Y=Q+rhocp/dt*T, r=T)
   T=myPDE.getSolution()
    n+=1
    saveVTK("u%s"%n, temperature=T, flux=-K*grad(T) )
    print("Time ",n*dt,": min/max temperature =",inf(T), sup(T))
```

python powered

Desktop

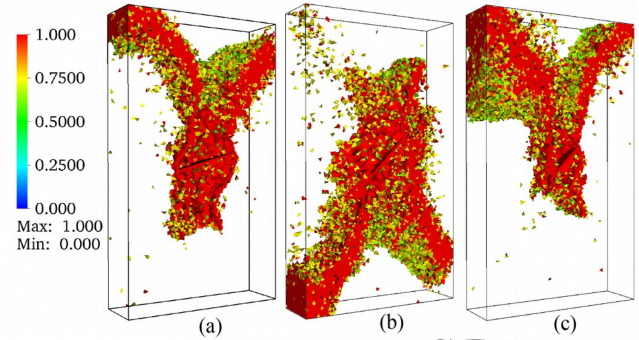
Parallel Supercomputers

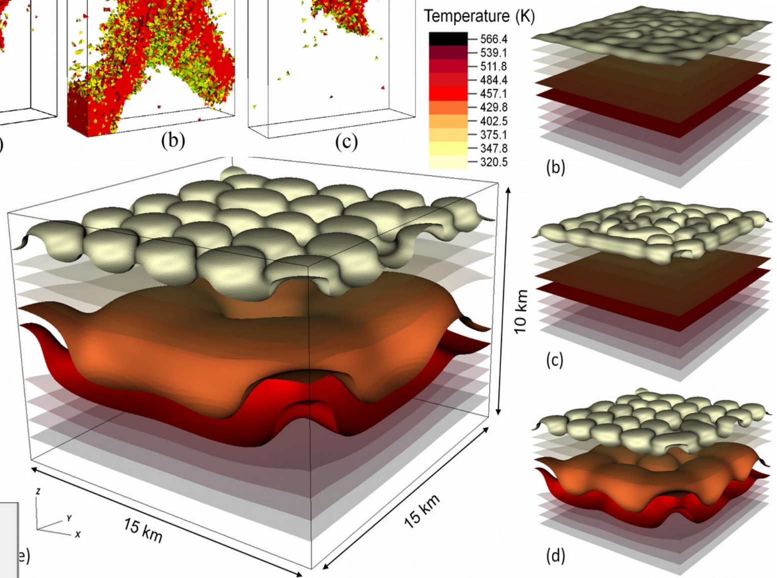
esys-escript (cont.)

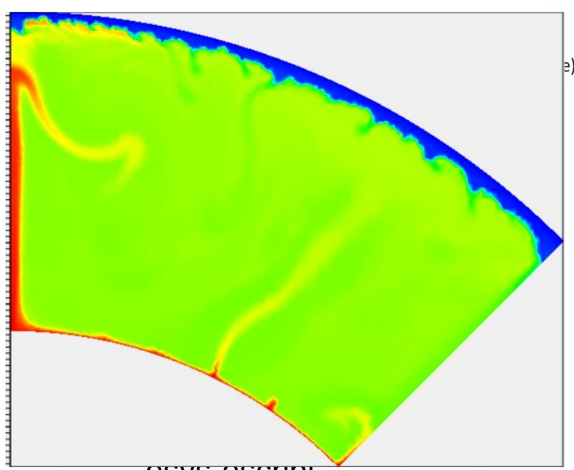
- Modeling with PDEs $\rho c_p \frac{\partial T}{\partial t} - \nabla^t K \nabla T = Q$
- Rapid prototyping of new models 
- Easy of Use 
- “Supercomputing for the masses” 
- Open Source
- Programming in python

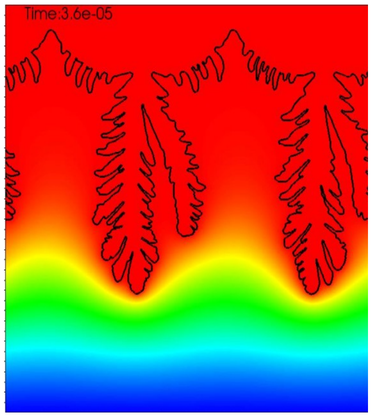
Some applications

- Mantel Convection
- Pores Media Flow
- Geomechanics
- Geophysical inversion
- Earthquakes
- Volcanoes
- Tsunamis
- ...

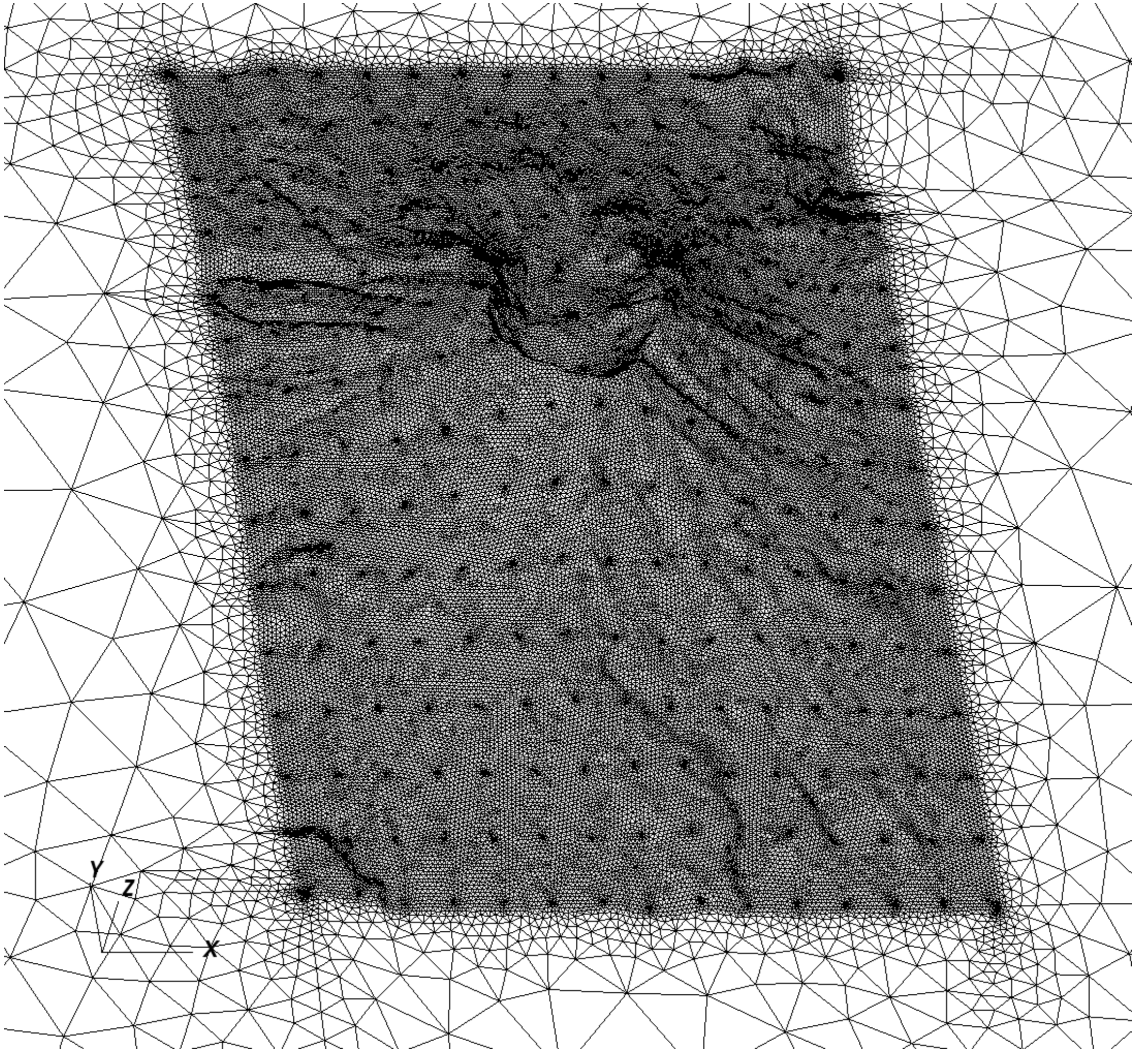




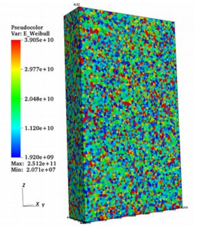




In 3D with topography



Discretization in esys-escript



- Finite Element Method (FEM)
  - Grids and unstructured meshes in 2D and 3D
- approximate solution PDEs in variational form
  - by continuous, piecewise linear approximation
  - Solve discrete problem with sparse matrix
    - With direct, iterative, multi-grid solver
- Parallelization on large number of cores (>25000)
  - Use Domain decomposition
  - Hidden from then user

Downloads

- Program: https://launchpad.net/escript-finley
  - Installation from source
- API Documentation: https://esys-escript.readthedocs.io
- Anaconda: https://anaconda.org/conda-forge/esys-escript
- Flatpack store

In this course

- Gravity Anomalies
- Magnetic Field
- Magnetotellurics
- Seismic Waves
  - in frequency domain
  - In time domain

Python Script Structure

(a) Definition of the domain

(b) Definition of PDE coefficients

(c) Obtain Solution

(d) Post-processing

- Get solution at points → recordings
- Write to file → visualization

Escript's PDEs

general linear PDE for unknown solution *u* :

'generalized' flux: $\boldsymbol{F} = -\boldsymbol{A}\nabla u + \boldsymbol{X}$

conservation equation:

$$\nabla^t \boldsymbol{F} + D u = \underbrace{Y}_{\text{volume source}} + \underbrace{y_{dirac}}_{\text{point source}}$$

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Escript's PDEs in 2D

$$\boldsymbol{F} = \begin{bmatrix} F_0 \\ F_1 \end{bmatrix} = -\begin{bmatrix} A_{00}\dfrac{\partial u}{\partial x_0} + A_{01}\dfrac{\partial u}{\partial x_1} + X_0 \\ A_{10}\dfrac{\partial u}{\partial x_0} + A_{11}\dfrac{\partial u}{\partial x_1} + X_1 \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix}$$

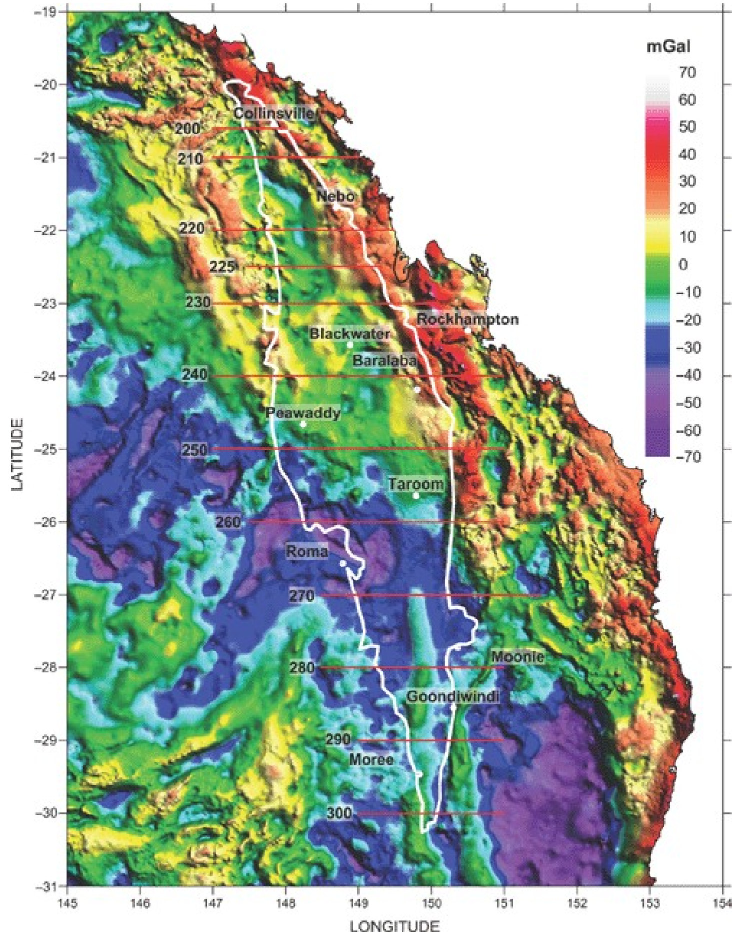
$$\frac{\partial F_0}{\partial x_0} + \frac{\partial F_1}{\partial x_1} + D u = Y + y_{dirac}$$

Escript's PDEs in Python

```
from esys.escript.linearPDEs import LinearSinglePDE
domain = ...
mypde=LinearSinglePDE(domain)
mypde.setValue(A=?, X=?,  D=?, Y=?, y_dirac=? )
u=mypde.getSolution()
```

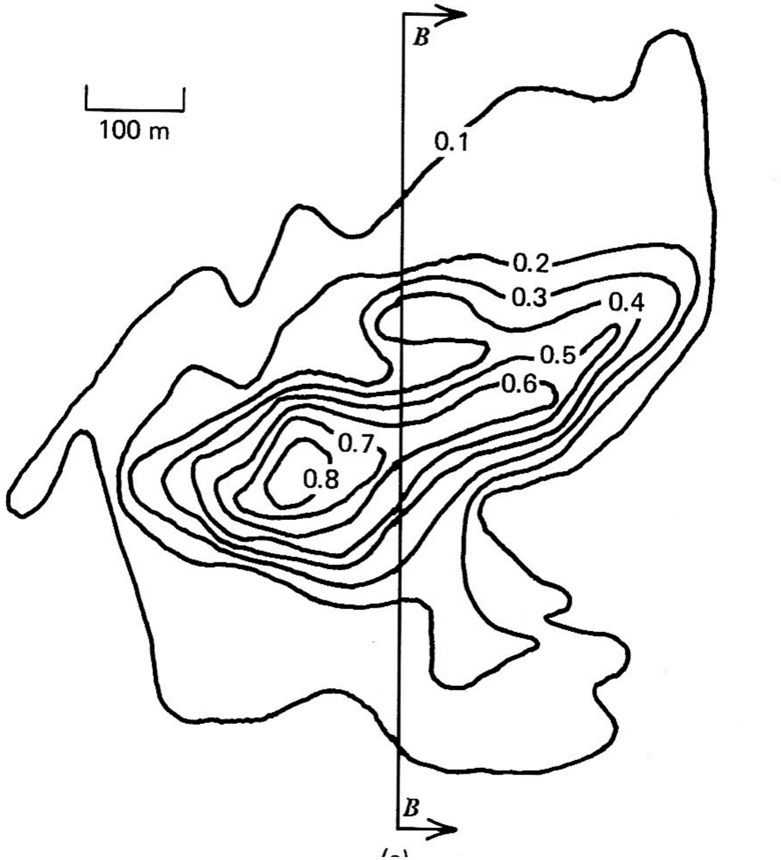
Identify PDE coefficients!?

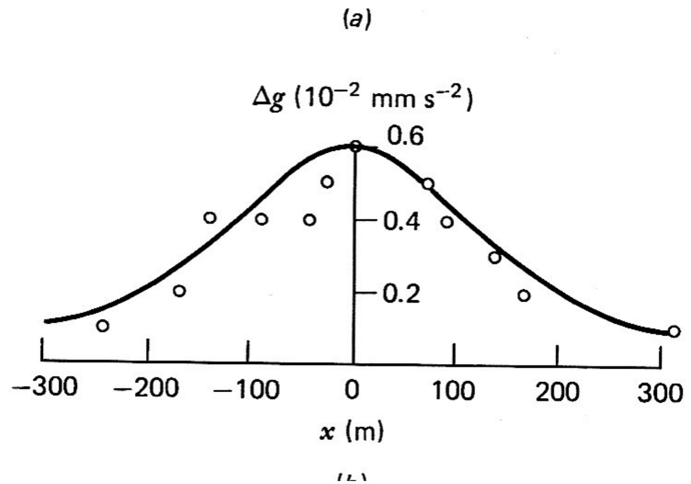
Modeling Gravity Anomaly



Gravity Anomalies (cont.)

Contour plot of vertical gravity anomaly g_z



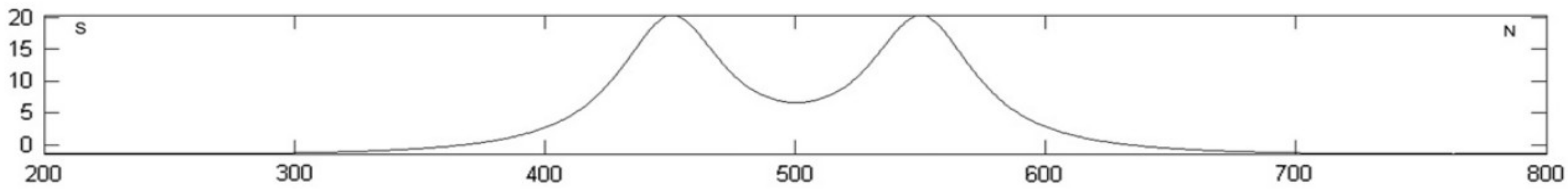


2D Interpolation of point observations

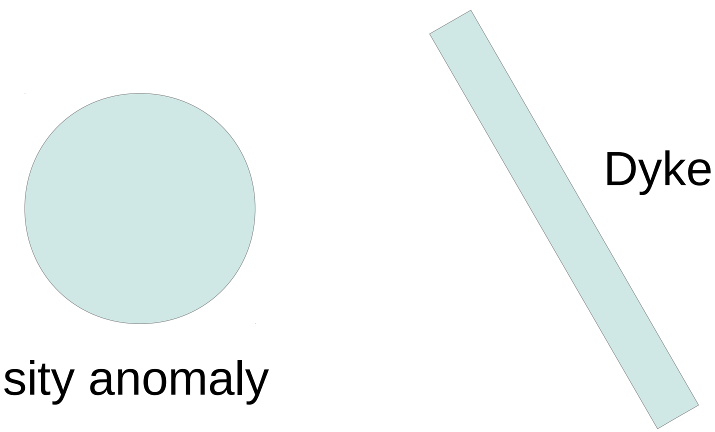
From Turcotte & Schubert Geodynamics

Gravity anomaly

Deviation of g_z from background with constant density



Subsurface:



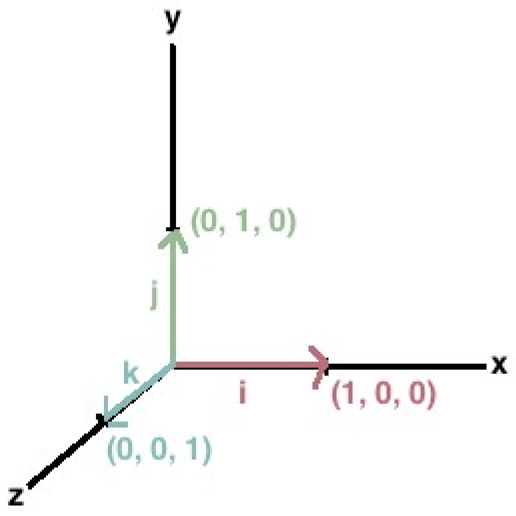
Gravity Force **g** as Vector Field

$$\boldsymbol{g}=\boldsymbol{i}\,g_x+\boldsymbol{j}\,g_y+\boldsymbol{k}\,g_z$$

unit vector x,y,z-direction : $\boldsymbol{i},\boldsymbol{j},\boldsymbol{k}$

x,y,z components of $\boldsymbol{g}: g_x, g_y, g_z$

$$\boldsymbol{g}=(g_x, g_y, g_z)$$



Gauss's law for gravity

- Generalized version of Newton's law

$$\frac{\partial g_x}{\partial x}+\frac{\partial g_y}{\partial y}+\frac{\partial g_z}{\partial z}=-4\pi G\rho$$

$G=$ universal gravitational constant

$\rho=$ density (anomaly)

The Field Potential

- The gravity field **g** is expressed as the gradient of a the gravity potential U_G
  - Work with a scalar rather then vector quantity

$$\boldsymbol{g} = -\nabla U_G = -\left(\boldsymbol{i}\frac{\partial U_G}{\partial x} + \boldsymbol{j}\frac{\partial U_G}{\partial y} + \boldsymbol{k}\frac{\partial U_G}{\partial z}\right)$$

$$g_x = -\frac{\partial U_G}{\partial x}, g_y = -\frac{\partial U_G}{\partial y}, g_z = -\frac{\partial U_G}{\partial z}$$

Compact Form

$$\boldsymbol{g} = -\nabla U_G$$

$$\nabla^t \cdot \boldsymbol{g} = -4\pi G\rho$$

$$\nabla = \text{gradient operator}$$

$$\nabla^t = \text{divergence operator}$$

In Escript?

$$\nabla^t \cdot \boldsymbol{g} = -4\pi G\rho \quad \Longleftrightarrow \quad \boldsymbol{\nabla}^t \boldsymbol{F} + D u = Y + y_{dirac}$$

$$\boldsymbol{g} = \boldsymbol{F} \qquad Y = -4\pi G\rho$$

$$y_{dirac} = 0 \quad D = 0$$

$$U_G = u \qquad \boldsymbol{A} = \boldsymbol{I}$$

$$\boldsymbol{X} = \boldsymbol{0}$$

$$\boldsymbol{g} = -\nabla U_G \quad \Longleftrightarrow \quad \boldsymbol{F} = -\boldsymbol{A}\boldsymbol{\nabla} u + \boldsymbol{X}$$

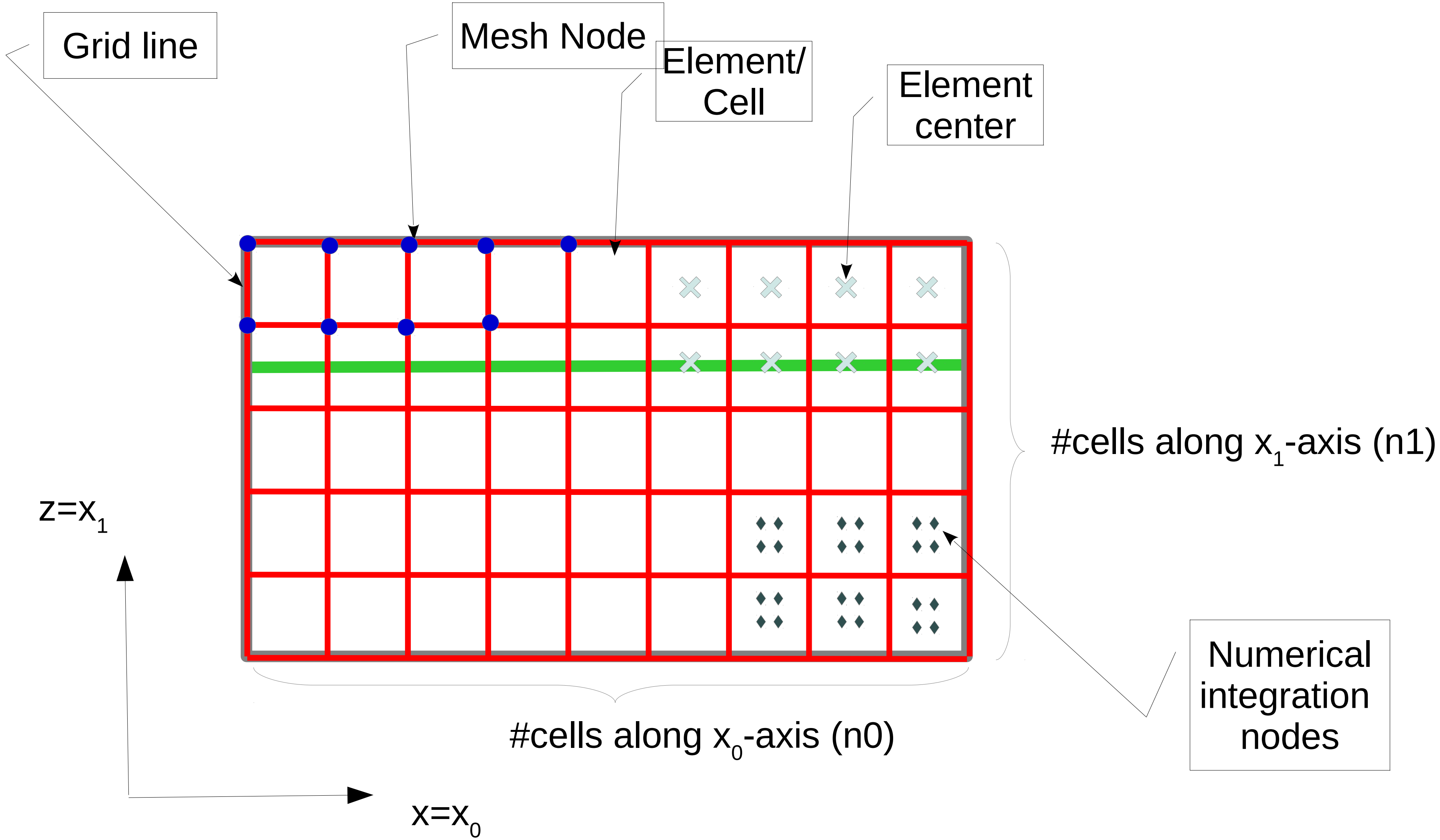
Escript's PDEs in 2D

$$u = U_G, x_0 = y, x_1 = z$$

$$\boldsymbol{F} = \begin{bmatrix} F_0 \\ F_0 \end{bmatrix} = \begin{bmatrix} g_y \\ g_z \end{bmatrix} = -\begin{bmatrix} A_{00}\frac{\partial u}{\partial x_0} + A_{01}\frac{\partial u}{\partial x_1} \\ A_{10}\frac{\partial u}{\partial x_0} + A_{11}\frac{\partial u}{\partial x_1} \end{bmatrix} = -\begin{bmatrix} \frac{\partial u}{\partial x_0} \\ \frac{\partial u}{\partial x_1} \end{bmatrix} \qquad \boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\frac{\partial F_0}{\partial x_0} + \frac{\partial F_1}{\partial x_1} = \frac{\partial g_y}{\partial x_1} + \frac{\partial g_z}{\partial x_1} = Y = -4\pi G \rho$$

Grid/Mesh Representation



Grids/meshes

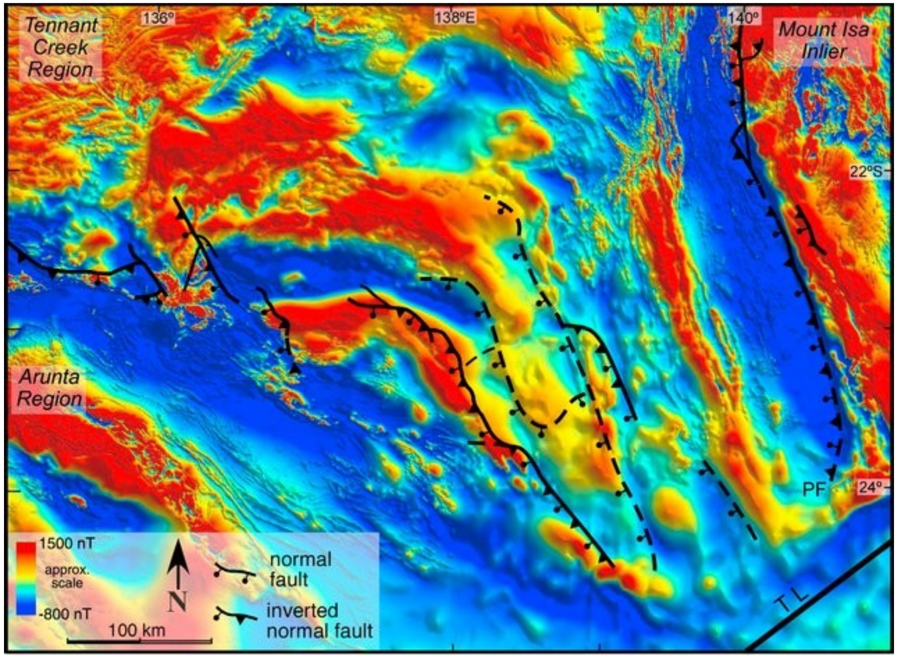
- Terminology: n0 x n1 grid
  - Example before: 9x5 grid
- Quantities are represented by values
  - At Nodes or
  - At Cell centers or
  - At Integration nodes

Mesh/Grid Generation

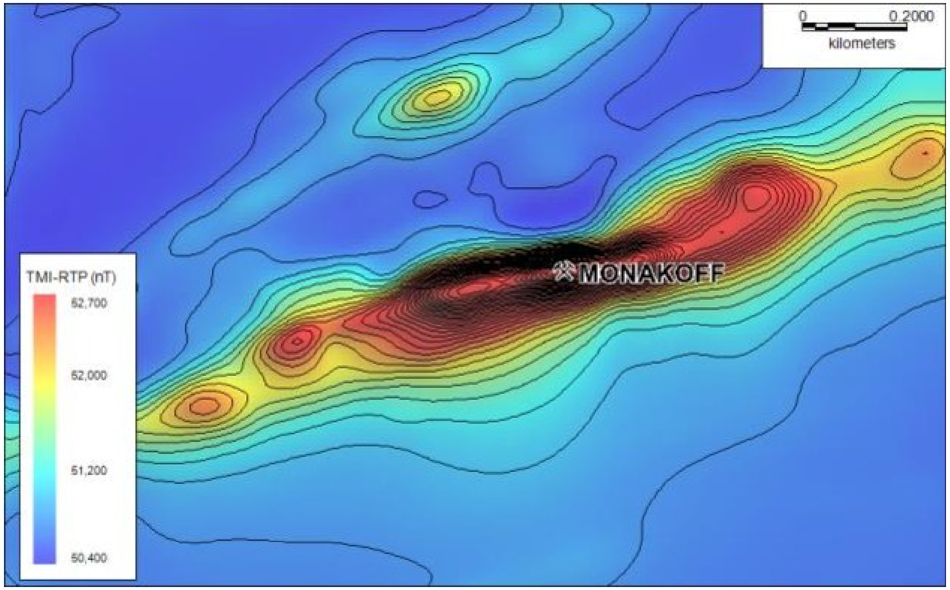
- Here we use 2D rectangular Domain with a rectangular grid:

```
from esys.finley import Rectangle
dx=10
n0, n1=100, 100
Lx=Nx*dx
Ly=Ny*dx
domain=Rectangle(n0=Nx, n1=Ny, l0=Lx, l1=Ly)
```

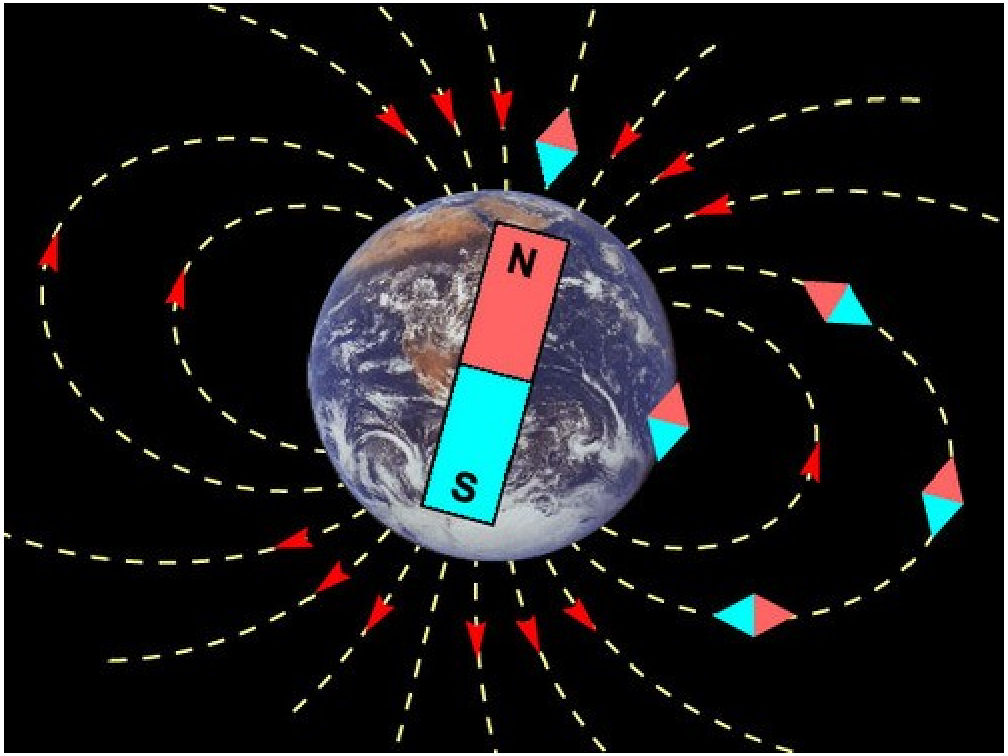
Magnetic Anomaly Mapping



Here: magnetic field anomaly
=
difference of total and background field intensity



Background Magnetic Field



Magnetic Field

- Magnetic Induction $\mathbf{B_I}$ is sum of magnetic field $\mathbf{B_t}$ and magnetization $\mathbf{M}$:

$$\boldsymbol{B}_I = \boldsymbol{M} + \boldsymbol{B}_t$$

- Magnetic Induction is divergence free:

$$\nabla^t \boldsymbol{B}_I = 0$$

Susceptibility

Magnetization **M** is proportional to background magnetic field $\mathbf{B}_h$

$$\boldsymbol{M} = k\, \boldsymbol{B}_b$$

- With susceptibility k,
  - 0<k<1
  - Causes magnetic anomalies
  - Is depending on location in the subsurface

The Field Potential

- The magnetic field **B** is expressed as the gradient of a the gravity potential U_M
  - Work with a scalar rather then vector quantity

$$\boldsymbol{B} = -\nabla U_M = -\left(\boldsymbol{i}\frac{\partial U_M}{\partial x} + \boldsymbol{j}\frac{\partial U_M}{\partial y} + \boldsymbol{k}\frac{\partial U_M}{\partial z}\right)$$

$$B_x = -\frac{\partial U_G}{\partial x}, B_y = -\frac{\partial U_G}{\partial y}, B_z = -\frac{\partial U_G}{\partial z}$$

Governing Equation

set : $\boldsymbol{B} = \boldsymbol{B_t} - \boldsymbol{B_h}$ assume : $\nabla^t \boldsymbol{B}_b = 0$

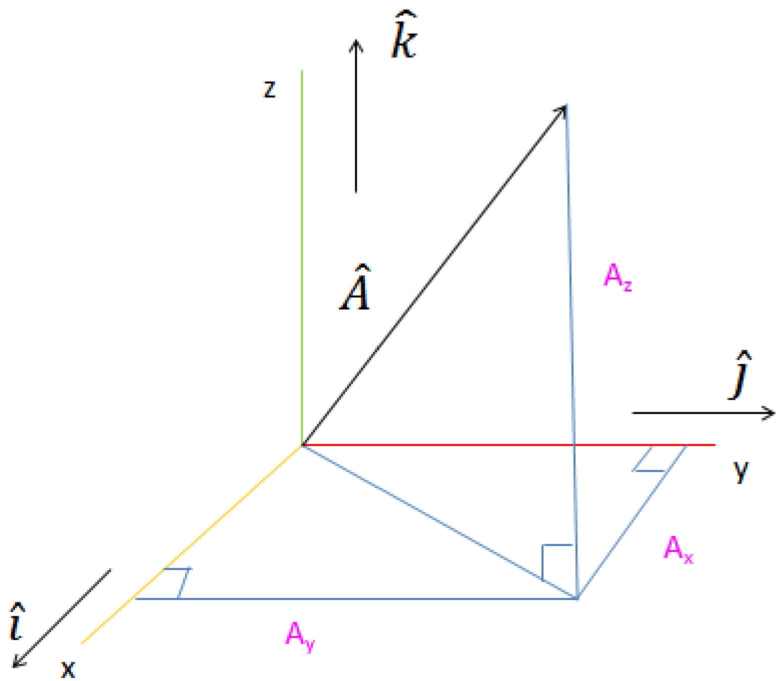
$$\nabla^t \left(k\, \boldsymbol{B}_h + \boldsymbol{B} \right) = 0$$

$$\boldsymbol{B} = -\nabla U_M$$

Intensity of a Field

Intensity, length, norm:

$$b = |\boldsymbol{B}| = \sqrt{B_x^2 + B_y^2 + B_z^2}$$



Magnetic Field Anomaly

=Difference of the intensity of the total magnetic field $\mathbf{B_t}$ and the intensity of the background magnetic field $\mathbf{B_b}$

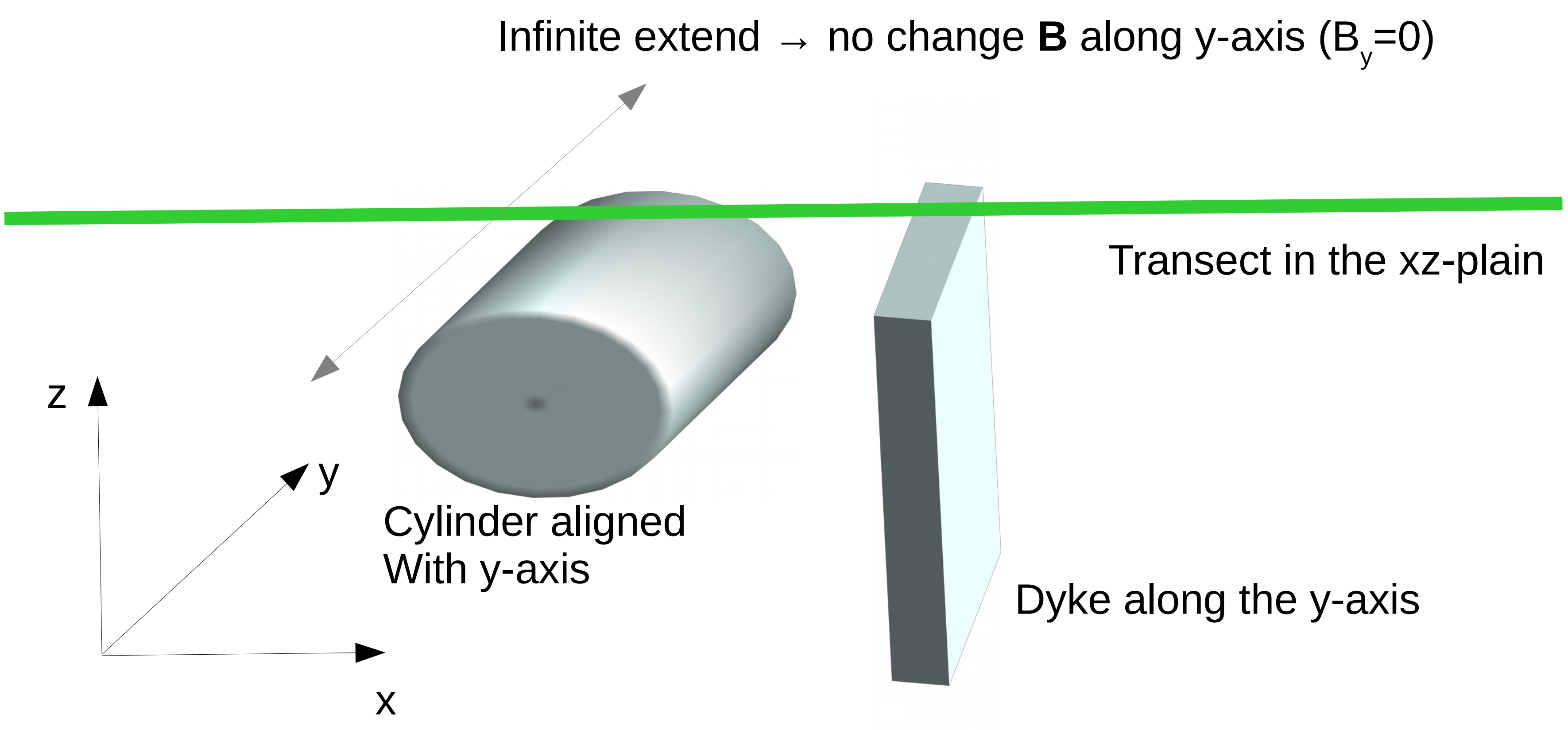
$$b_a = |\boldsymbol{B_t}| - |\boldsymbol{B_b}|$$

$$\boldsymbol{B}_t = \boldsymbol{B} + \boldsymbol{B_b}$$

Modeling magnetic response?

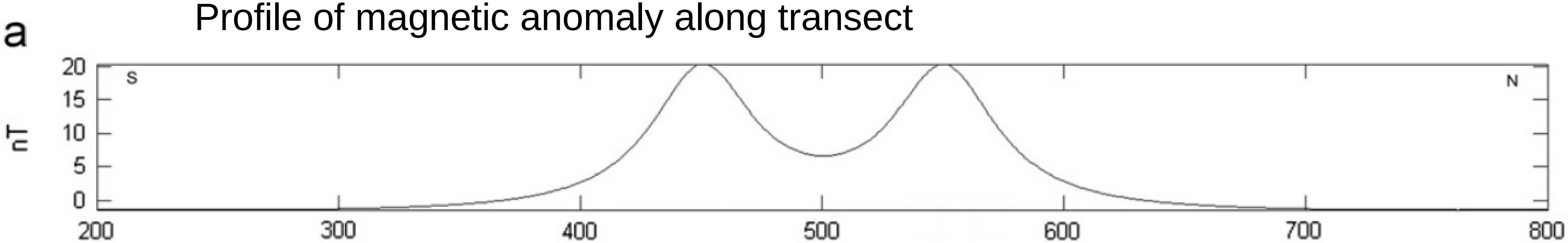
- Given an anomaly in the subsurface as
  - Susceptibility distribution/anomaly k
  - Background magnetic field $\mathbf{B}_H$
- What is the total total magnetic field anomaly in the subsurface?

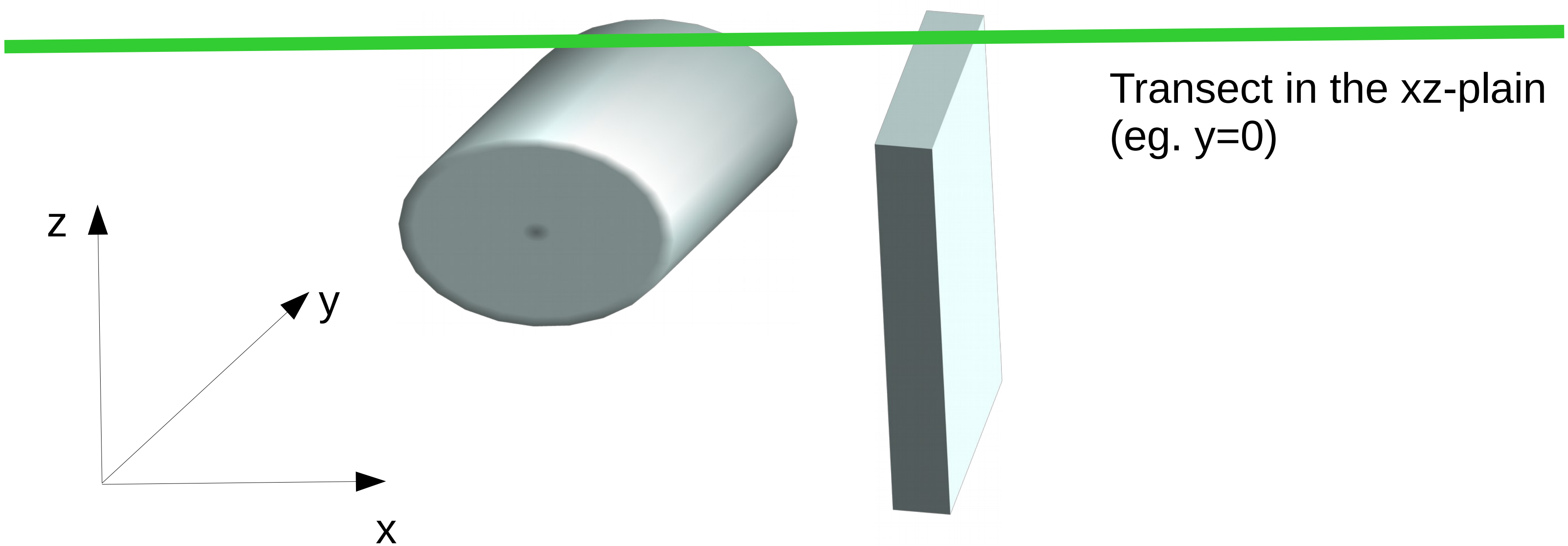
2D Modeling Setup



2D Modeling Setup

Profile of magnetic anomaly along transect





In Escript?

$$\nabla^t \left(k\, \boldsymbol{B}_h + \boldsymbol{B}\right) = 0 \quad \Longleftrightarrow \quad \nabla^t \boldsymbol{F} + D\, u = Y + y_{dirac}$$

$$\boldsymbol{B} + k\, \boldsymbol{B}_h = \boldsymbol{F}$$

$$D = 0$$

$$y_{dirac} = 0$$

$$Y = 0$$

$$U_M = u$$

$$\boldsymbol{A} = \boldsymbol{I}$$

$$\boldsymbol{X} = k\, \boldsymbol{B}_h$$

$$\boldsymbol{F} = -\nabla U_M + k\, \boldsymbol{B}_h \quad \Longleftrightarrow \quad \boldsymbol{F} = -\boldsymbol{A} \nabla u + \boldsymbol{X}$$

Escript's PDEs in 2D

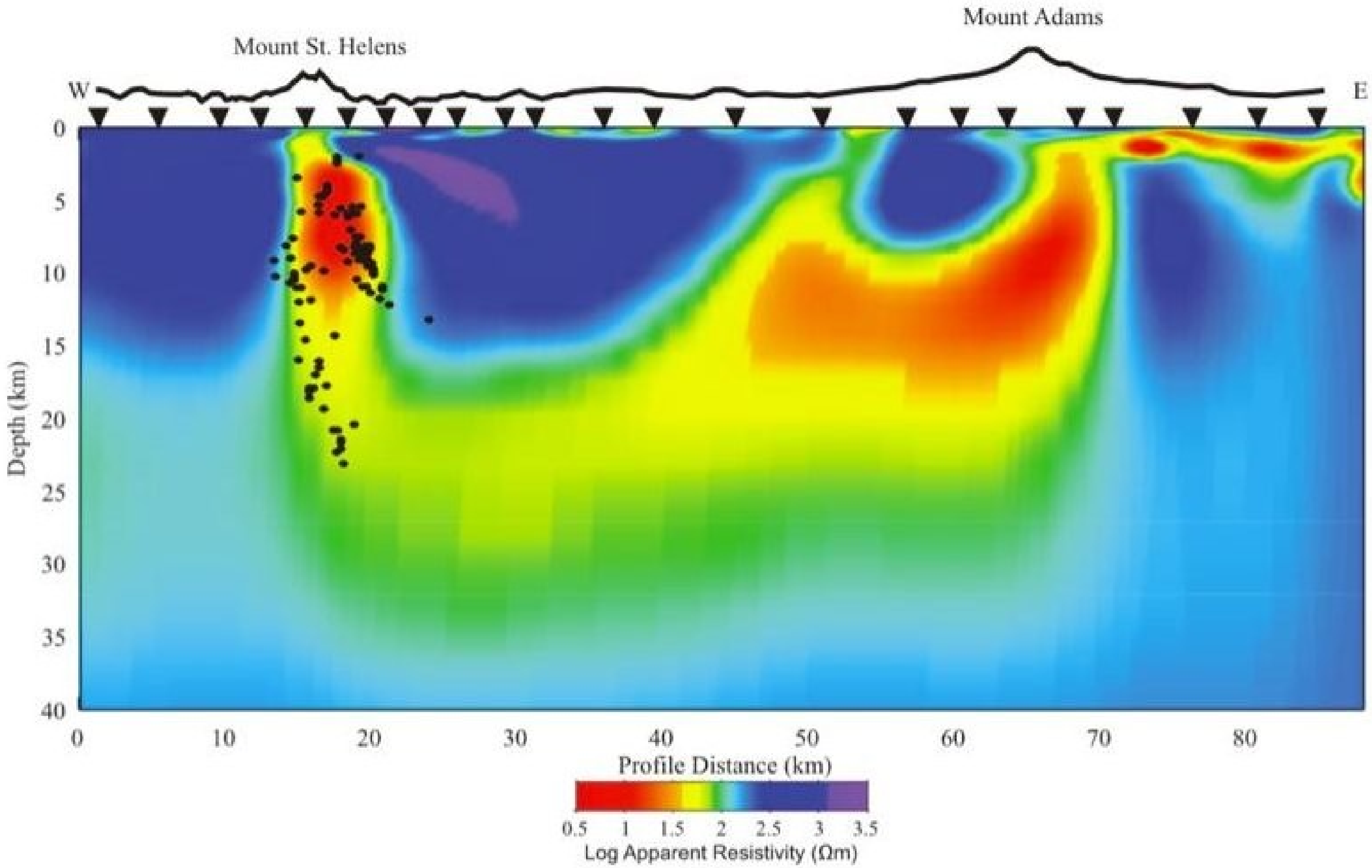
$$u=U_M\,,x_0=y\,,x_1=z$$

$$\boldsymbol{F}=\begin{bmatrix}F_0\\F_1\end{bmatrix}=\begin{bmatrix}B_y+k\,B_{by}\\B_z+k\,B_{bz}\end{bmatrix}=-\begin{bmatrix}\frac{\partial u}{\partial x_0}+X_0\\\frac{\partial u}{\partial x_1}+X_1\end{bmatrix}=-\begin{bmatrix}A_{00}\frac{\partial u}{\partial x_0}+A_{01}\frac{\partial u}{\partial x_1}+X_0\\A_{10}\frac{\partial u}{\partial x_0}+A_{11}\frac{\partial u}{\partial x_1}+X_1\end{bmatrix}$$

$$\begin{bmatrix}X_0\\X_1\end{bmatrix}=\begin{bmatrix}k\,B_{by}\\k\,B_{bz}\end{bmatrix}\quad\boldsymbol{A}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$$

$$\frac{\partial F_0}{\partial x_0}+\frac{\partial F_1}{\partial x_1}=\frac{\partial}{\partial x_1}\left(B_y+k\,B_{by}\right)+\frac{\partial}{\partial x_1}\left(B_z+k\,B_{bz}\right)=Y=0$$

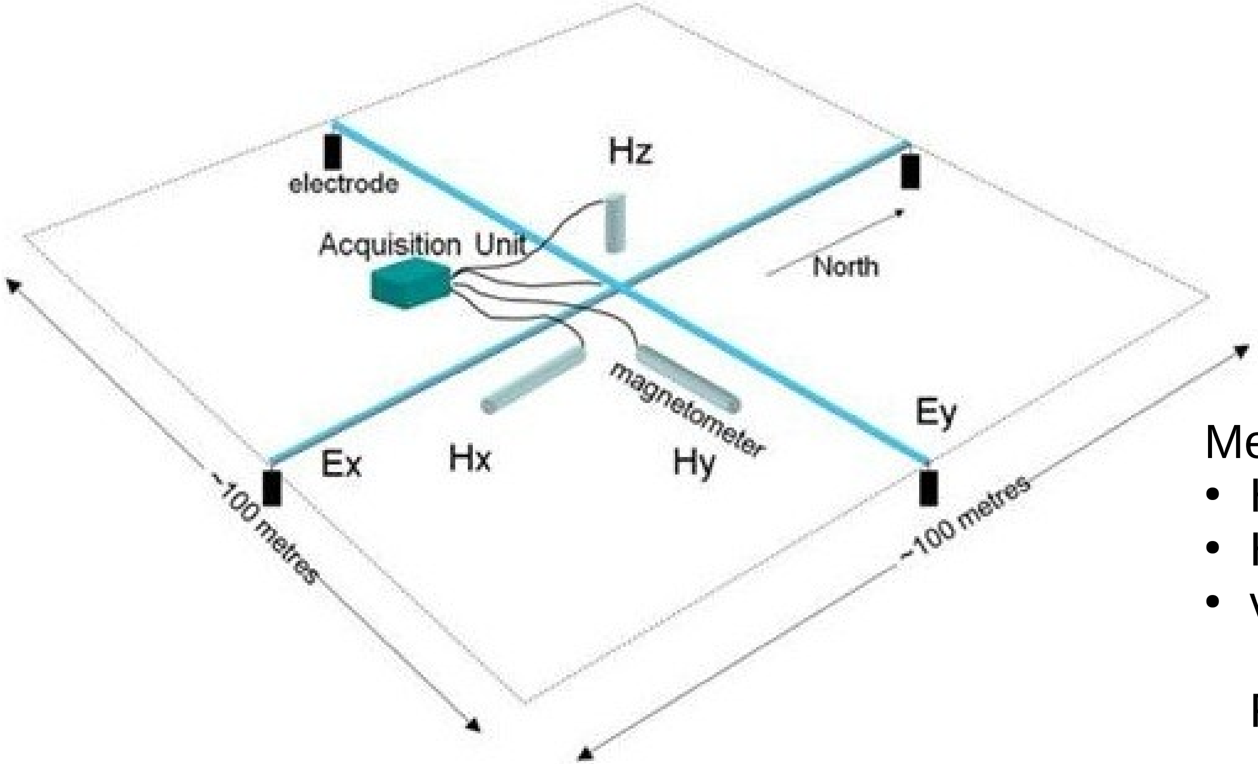
Magnetotellurics (MT)



The Method of Magnetotellurics

- passive electromagnetic (EM) method
  - Target: subsurface resistivity distribution
- naturally generated source field by variations in Earth’s magnetic field
  - EM field from the sum (<1Hz)
  - Lightnings (>1Hz)

Survey Design

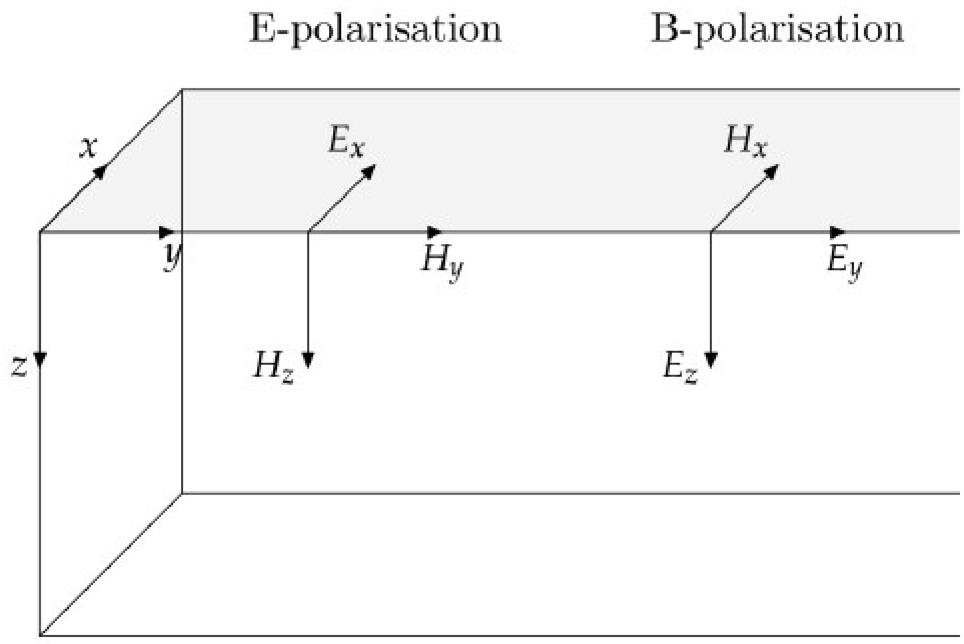


Measured Fields

- Horizontal electric Ex, Ey
- Horizontal magnetic Hx, Hy
- vertical magnetic Hz

Recorded over several hours

TM-mode: H–Polarisation



Measurements provide Impedance defined as

$$Z_{yx} = \frac{E_y}{H_x}$$

Depending of frequency f of incoming wave!

Quantities For Interpretation

- Apparent Resistivity for angular frequency $\omega = 2\pi f$
- $$\rho_a = \frac{1}{\omega \mu_0} |Z_{yx}(\omega)|^2$$
- Phase of impedance:

$$\mu_0 = 1.26 \cdot 10^{-6} \frac{mkg}{s^2 A^2}$$

$$\phi = \arctan\left(\frac{Z_{yx}(\omega).imag}{Z_{yx}(\omega).real}\right)$$

- To obtain information on subsurface conductivity σ
  - look at dependency on ω
  - Use 2D or 3D profiles at surface

Modeling equations: TM mode

$$\frac{\partial H_x}{\partial y} = -\sigma E_z$$

$$\frac{\partial H_x}{\partial z} = \sigma E_y$$

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\boldsymbol{j}\,\omega \mu_0 H_x$$

Recall Esys-escript PDE template

$$\text{flux}: \boldsymbol{F} - \boldsymbol{A}\nabla u + \boldsymbol{X} = \begin{bmatrix} -A_{00}\dfrac{\partial u}{\partial x_0} - A_{01}\dfrac{\partial u}{\partial x_0} - X_0 \\ -A_{10}\dfrac{\partial u}{\partial x_0} - A_{11}\dfrac{\partial u}{\partial x_1} - X_1 \end{bmatrix}$$

conservation equation:

$$Y = \nabla^t \boldsymbol{F} + Du = \frac{\partial F_0}{\partial x_0} + \frac{\partial F_1}{\partial x_1} + Du$$

The University of Queensland Australia

TM in esys.escript (cont.)

conservation equation:

$$Y = \frac{\partial F_0}{\partial x_0} + \frac{\partial F_1}{\partial x_1} + D u$$

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\boldsymbol{j}\,\omega \mu_0 H_x$$

$$\text{flux}: \boldsymbol{F} = \begin{bmatrix} F_0 \\ F_1 \end{bmatrix} = \begin{bmatrix} E_z \\ -E_y \end{bmatrix}$$

$$D = \boldsymbol{j}\,\omega \mu_0 \quad Y = 0$$

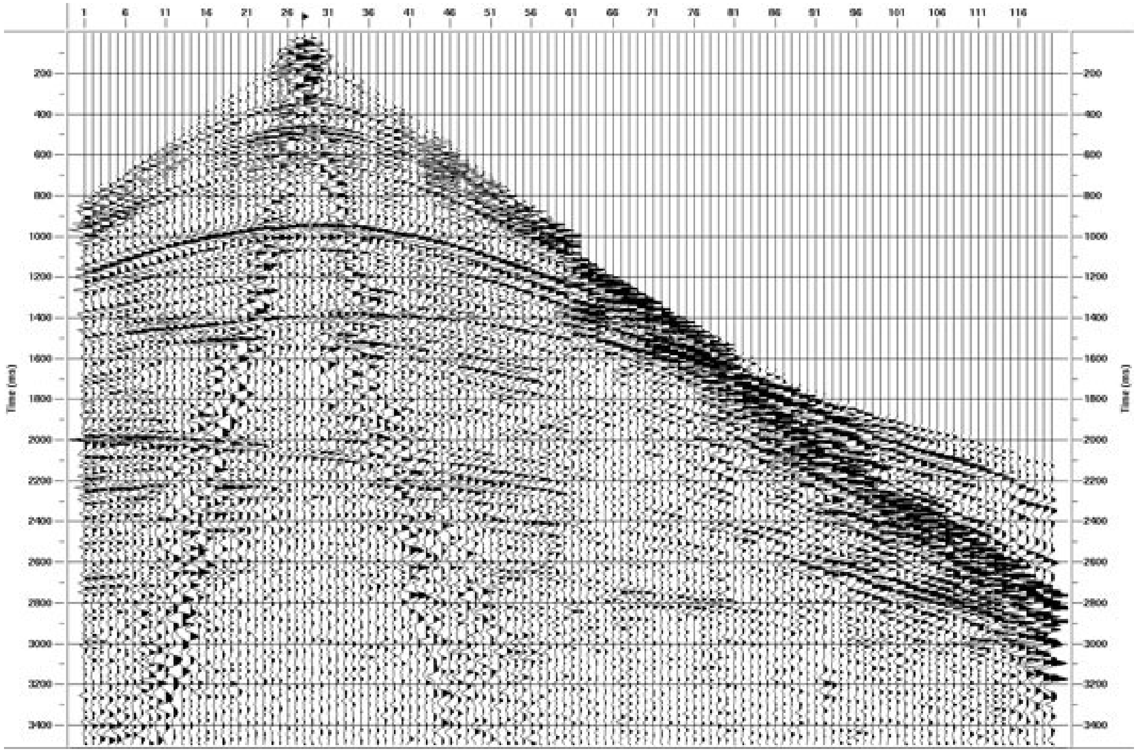
TM in esys.escript (cont.)

$$\frac{\partial H_x}{\partial y} = -\sigma E_z \qquad \frac{\partial H_x}{\partial z} = \sigma E_y$$

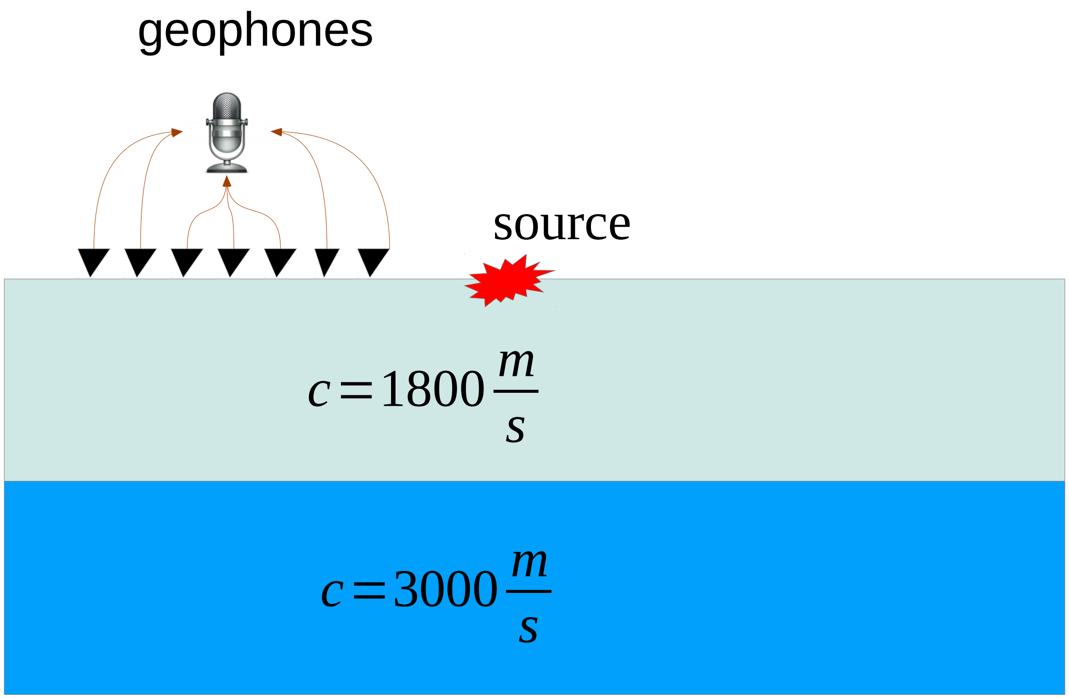
$$\boldsymbol{F} = \begin{bmatrix} E_z \\ -E_y \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sigma}\frac{\partial H_x}{\partial y} \\ -\frac{1}{\sigma}\frac{\partial H_x}{\partial z} \end{bmatrix} = -\begin{bmatrix} \frac{1}{\sigma}\frac{\partial u}{\partial x_0} + 0\frac{\partial u}{\partial x_1} \\ 0\frac{\partial u}{\partial x_0} + \frac{1}{\sigma}\frac{\partial u}{\partial x_1} \end{bmatrix}$$

$$u = H_x \qquad \boldsymbol{X} = 0 \qquad \boldsymbol{A} = \begin{bmatrix} \frac{1}{\sigma} & 0 \\ 0 & \frac{1}{\sigma} \end{bmatrix}$$

Wave propagation



Synthetic seismograms



Wave Propagation model

- Sonic wave equation in time domain

$$\frac{1}{\rho c^2}\frac{\partial^2 p}{\partial t^2} = \nabla^t \frac{1}{\rho} \nabla p + \dot{w}(t) \cdot \delta_{\mathbf{x}_s}$$

  - w(t) – source wavelet
  - Point source @ $\mathbf{x}_s$ by Dirac δ-Function
  - p(**x**,t) – pressure function of t and **x**
  - Density ρ
  - p-wave propagation speed

$$c = v_p = \sqrt{\frac{K + \frac{4}{3}G}{\rho}}$$

In Frequency Domain

- Apply Fourier Transformation on time:

$$\hat{p}(\boldsymbol{x},\omega)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} p(\boldsymbol{x},t)\,e^{\boldsymbol{i}\omega t}\,dt$$

- Angular frequency ω
- $\hat{p}$ is complex!
- As p is real: $\hat{p}(\boldsymbol{x},\omega)=\overline{\hat{p}(\boldsymbol{x},-\omega)}$
  - We need to look at ω≥0 only.

Wave Equation in Frequency Domain

$$-\nabla^t \nabla p - k^2 p = P_{s\omega} \cdot \delta_{\mathbf{x}_s}$$

$$k = \frac{\omega}{c}$$

- Solution $p=p(\mathbf{x};\omega,s)$
- $P_{s\omega}$ – power amplitude for point source at $\mathbf{x}_s$
- Assume constant density

Solution rescaling

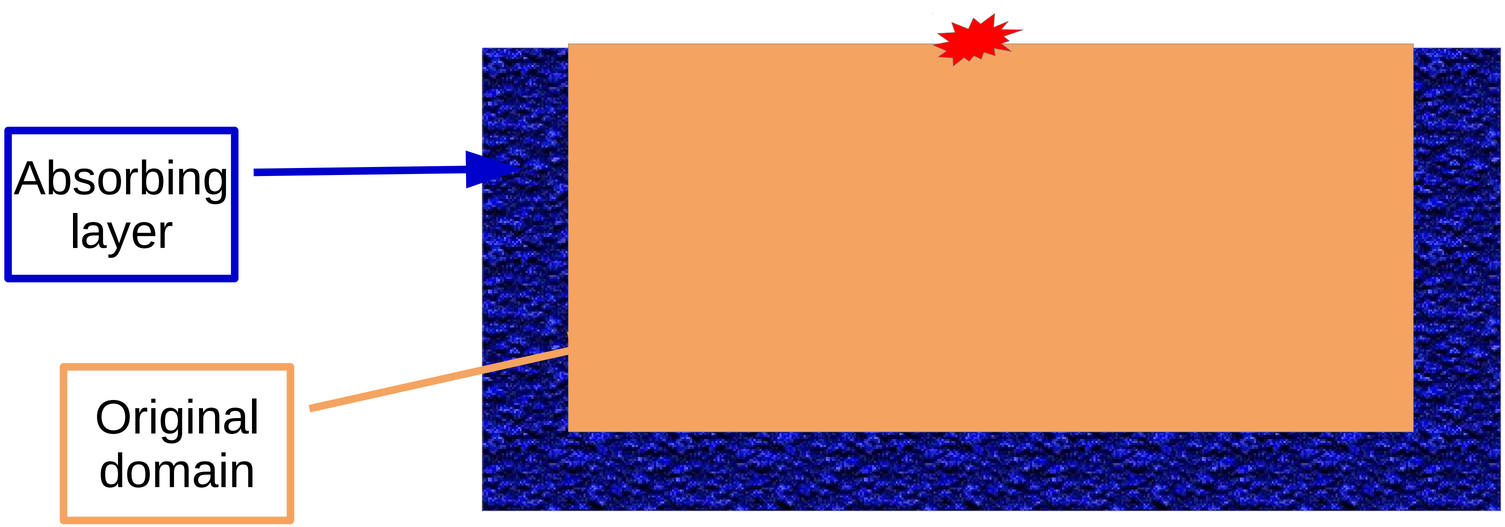
$$-\nabla^t \nabla u - k^2 u = \delta_{\boldsymbol{x}_s}$$

$$p(x;\omega,s) = P_{\omega s} \cdot u(x;\omega,s)$$

$$-\frac{\partial^2 u}{\partial x_0^2} - \frac{\partial^2 u}{\partial x_1^2} + k^2 u = \delta_{\boldsymbol{x}_s}$$

Boundary Reflection

- Undesirable wave front reflected from numerical domain boundary
- Introduce wave absorbing layer near boundary
  - → Perfect Matching Layers (PML)

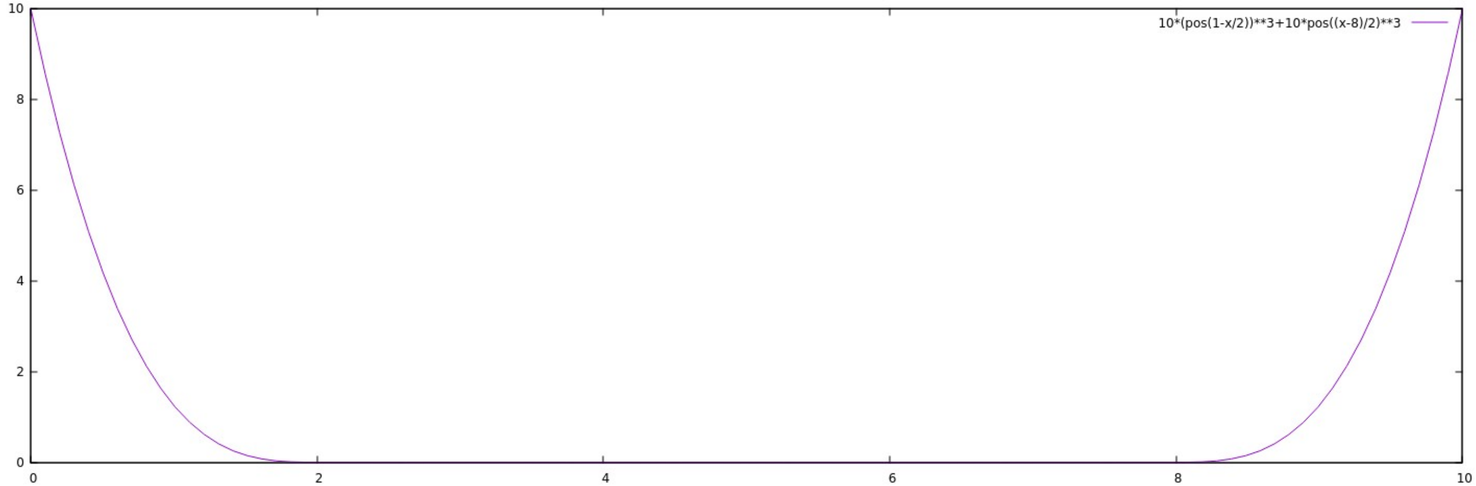


Perfect Matching Layers (PML)

$$\text{coordinate trafo}: x_i \rightarrow \gamma_i \cdot x_i$$

$$\gamma_i = \left(1 - \boldsymbol{j}\frac{S}{k} Q_i\right)$$

$$Q_i(x_i) = 1 \text{ near the boundries } x_i = 0 \,;\, x_i = L_i$$



PML (cont.)

$$-\frac{1}{\gamma_0}\frac{\partial}{\partial x_0}\frac{1}{\gamma_0}\frac{\partial u}{\partial x_0}-\frac{1}{\gamma_1}\frac{\partial}{\partial x_1}\frac{1}{\gamma_1}\frac{\partial u}{\partial x_1}-k^2 u=\delta_{\mathbf{x}_s}$$

multiply by: $\gamma_0\gamma_1$, use γ_i depends on x_i only!

$$-\frac{\partial}{\partial x_0}\frac{\gamma_1}{\gamma_0}\frac{\partial u}{\partial x_0}-\frac{\partial}{\partial x_1}\frac{\gamma_0}{\gamma_1}\frac{\partial u}{\partial x_1}-k^2\gamma_0\gamma_1 u=\delta_{\mathbf{x}_s}$$

In Esys-escript PDE template

conservation equation :

$$y_{dirac}+Y=\frac{\partial F_0}{\partial x_0}+\frac{\partial F_1}{\partial x_1}+Du$$

$$\text{flux}:\boldsymbol{F}=\begin{bmatrix}-\frac{\gamma_1}{\gamma_0}\frac{\partial u}{\partial x_0}\\ -\frac{\gamma_1}{\gamma_0}\frac{\partial u}{\partial x_0}\end{bmatrix}$$

$$D=-k^2\gamma_0\gamma_1$$

$$y_{dirac}=\delta_{\boldsymbol{x}_s}$$

$$Y=0$$

$$-\frac{\partial}{\partial x_0}\frac{\gamma_1}{\gamma_0}\frac{\partial u}{\partial x_0}-\frac{\partial}{\partial x_1}\frac{\gamma_0}{\gamma_1}\frac{\partial u}{\partial x_1}-k^2\gamma_0\gamma_1 u=\delta_{\boldsymbol{x}_s}$$

Recall Esys-escript PDE template

$$\boldsymbol{F} = \begin{bmatrix} -\dfrac{\gamma_1}{\gamma_0}\dfrac{\partial u}{\partial x_0} \\ -\dfrac{\gamma_1}{\gamma_0}\dfrac{\partial u}{\partial x_0} \end{bmatrix} = \begin{bmatrix} -A_{00}\dfrac{\partial u}{\partial x_0} - A_{01}\dfrac{\partial u}{\partial x_0} - X_0 \\ -A_{10}\dfrac{\partial u}{\partial x_0} - A_{11}\dfrac{\partial u}{\partial x_1} - X_1 \end{bmatrix}$$

$$\boldsymbol{X} = 0 \qquad \boldsymbol{A} = \begin{bmatrix} \dfrac{\gamma_1}{\gamma_0} & 0 \\ 0 & \dfrac{\gamma_0}{\gamma_1} \end{bmatrix}$$

Wave Equation in Time Domain

$$\frac{1}{\rho c^2}\frac{\partial^2 p}{\partial t^2} = \nabla^t \frac{1}{\rho} \nabla p + \dot{w}(t) \cdot \delta_{\boldsymbol{x}_s}$$

Wave Equation in Time Domain

$$\frac{1}{\rho c^2}\ddot{p} = \nabla^t \frac{1}{\rho} \nabla p + \dot{w}(t) \cdot \delta_{\boldsymbol{x}_s}$$

Translate to system of first order problems:

$$\dot{\boldsymbol{V}} = \frac{1}{\rho} \nabla p$$

$$\frac{1}{\rho c^2}\dot{p} = \nabla^t \boldsymbol{V} + w(t) \cdot \delta_{\boldsymbol{x}_s}$$

Time Integration

- Heun scheme solving :

$$\dot{U} = f(U, t)$$

for $U^{(n)} = U(t^{(n)})$ with $t^{(n)} + n \cdot h$

$$\hat{U}^{(n+1)} = U^{(n)} + h f(U^{(n)}, t^{(n)})$$
$$\bar{U}^{(n+1)} = \hat{U}^{(n+1)} + h f(\hat{U}^{(n+1)}, t^{(n+1)})$$
$$U^{(n+1)} = \frac{1}{2}(U^{(n)} + \bar{U}^{(n+1)})$$
$$n \leftarrow n+1$$

Application to Wave

$$\boldsymbol{U}=\begin{bmatrix}\boldsymbol{V}\\ p\end{bmatrix} \qquad f(U,t)=\begin{bmatrix}\boldsymbol{V}'\\ p'\end{bmatrix}$$

With

$$\boldsymbol{V}'=\frac{1}{\rho}\nabla p$$

$$\frac{1}{\rho c^2}p'=\nabla^t \boldsymbol{V}+w(t)\cdot\delta_{\boldsymbol{x}_s}$$

Solve PDE for u=p'

$$\frac{1}{\rho c^2} p' = \nabla^t \boldsymbol{V} + w(t) \cdot \delta_{\boldsymbol{x}_s}$$

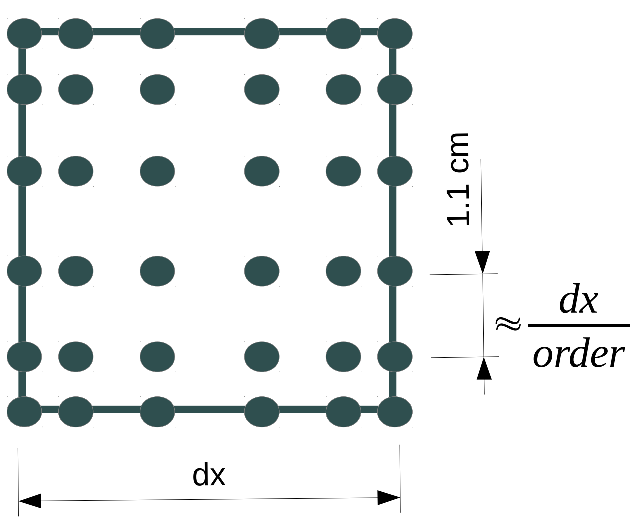
PDE template: $$\underbrace{D}_{\frac{1}{\rho c^2}} u = -\nabla^t \underbrace{\boldsymbol{F}}_{-\boldsymbol{V}} + Y + \underbrace{y_{dirac}}_{w(t) \cdot \delta_{\boldsymbol{x}_s}}$$

$$\boldsymbol{F} = -\underbrace{\boldsymbol{A}}_{=0} \nabla u + \boldsymbol{X} = -\boldsymbol{V}$$

Spectral Element Method (SEM)

- Implemented in the esys.speckley module
- Higher polynomial order polynomial
  - Non-equidistant nodes
  - Very good for waves

A grid cell:



Stability & Accuracy

- Small step size for stability:
  - In a time step wave travels never more than a node spacing:

$$h < \frac{dx}{order} \cdot \frac{1}{max_{\mathbf{x}} c(\mathbf{x})}$$

- Small element size for accuracy:
  - Wave length is never shorter than the node spacing:

$$dx < \frac{order}{2\pi f_{max}} \cdot min_{\mathbf{x}} c(\mathbf{x})$$

f_{max} maximum frequency in wavelet