Inverse Problems in Geophysics
Part 9: Newtons method

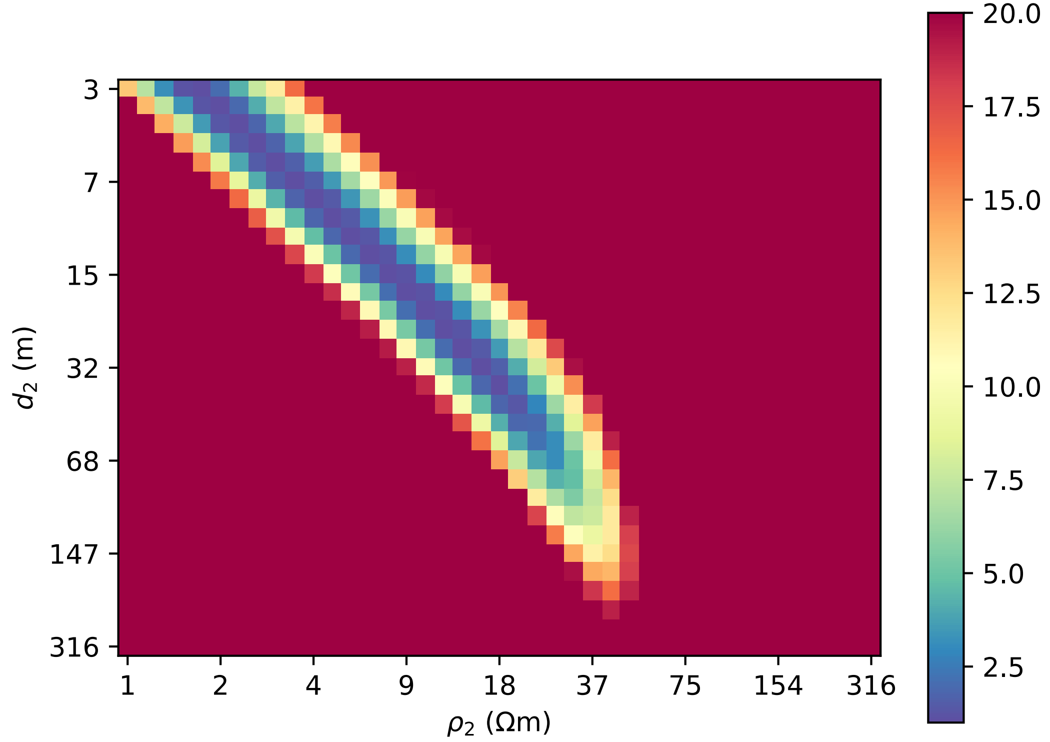
2. MGPY+MGIN

Thomas Günther
thomas.guenther@geophysik.tu-freiberg.de

TU BERGAKADEMIE FREIBERG
TUBAF
Die Ressourcenuniversität. Seit 1765.

Recap

- linear problems: least-squares solution of (regularized) problem
- non-linear problems: linearization of $f(m) \Rightarrow$ linear problem for $\mathbf{\Delta m}$ and $\mathbf{\Delta d} = \mathbf{d} - \mathbf{f(m)}$
- **grid search**: systematic search through model space



Objective function for VES

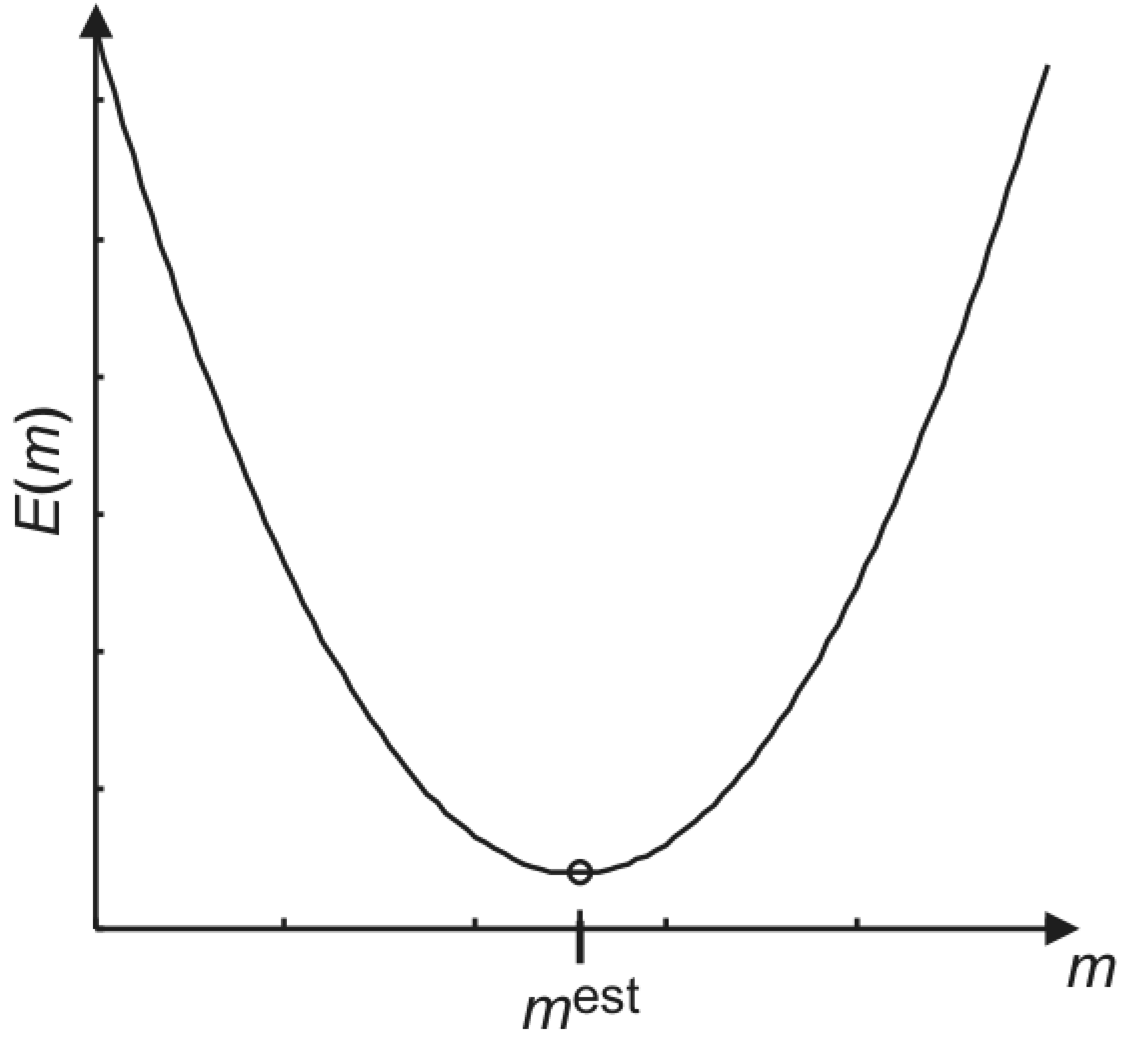
Objective function for linear problems

$$\Phi_d = (\mathbf{d} - \mathbf{Gm})^T(\mathbf{d} - \mathbf{Gm}) =$$

$$(\mathbf{d}^T - \mathbf{m}^T\mathbf{G}^T)(\mathbf{d} - \mathbf{Gm}) =$$

$$\mathbf{d}^T\mathbf{d} + \mathbf{m}^T\mathbf{G}^T\mathbf{Gm} - 2\mathbf{d}^T\mathbf{Gm}$$

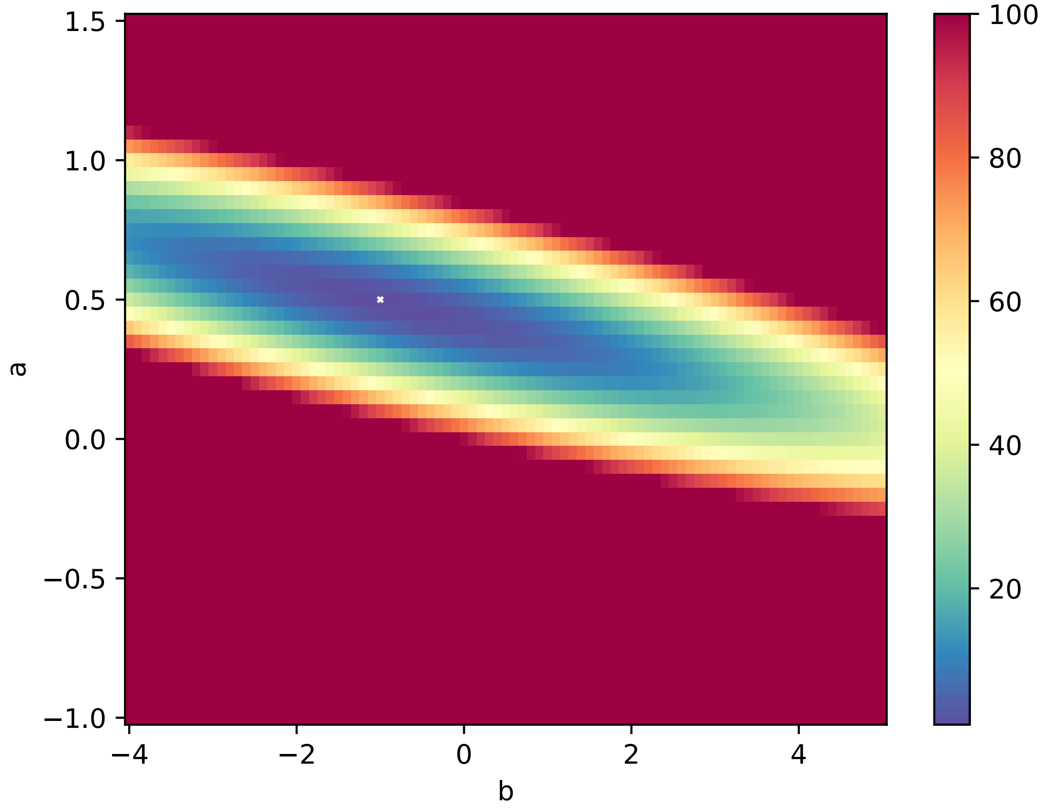
$\Rightarrow$ a parabola for all m_i

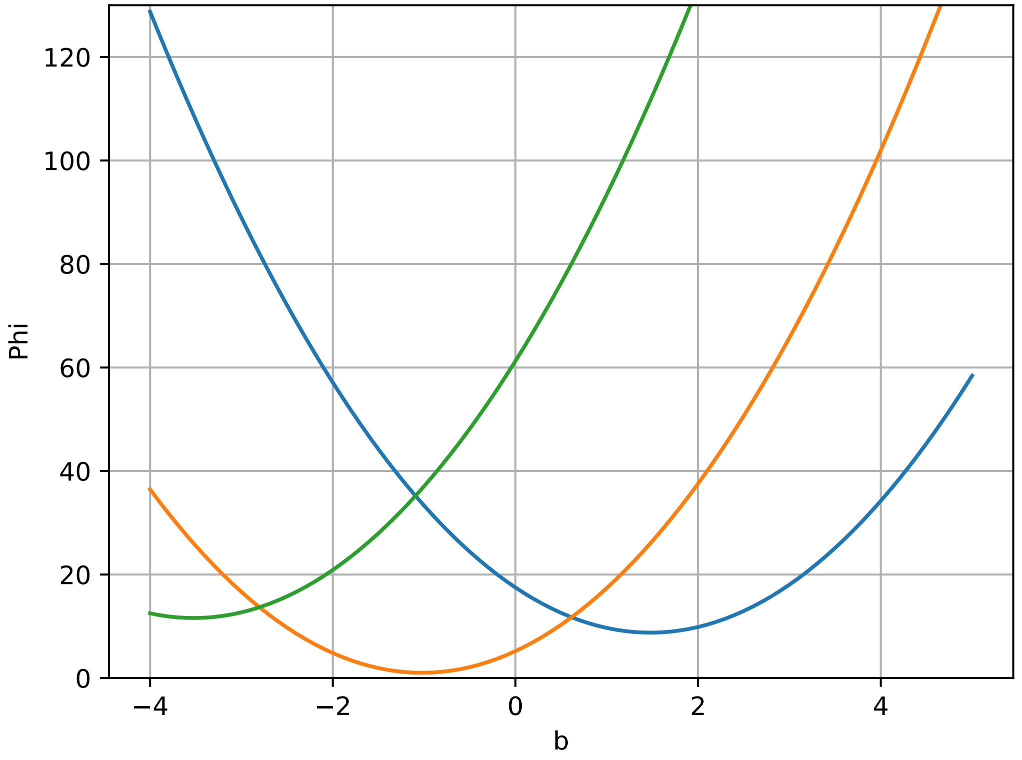


Menke (2012)

Example: linear regression

$y = ax + b$ (a=0.5, b=-1)





Newtons method

Taylor series expansion for the objective function:

$$\Phi_d(\mathbf{m}) = \Phi_d(\mathbf{m}^0) + \sum_i^M b_i(m_i - m_i^0) +$$

$$+\frac{1}{2}\sum_i^M\sum_j^M B_{ij}(m_i - m_i^0)(m_j - m_j^0)$$

with **gradient vector** $b_i = \frac{\partial \Phi_d}{\partial m_i}$ and **Hessian matrix** $B_{ij} = \frac{\partial^2 \Phi_d}{\partial m_i \partial m_j}$

Newtons method

$$\ldots \sum_i^M b_i(m_i - m_i^0) + \frac{1}{2}\sum_i^M \sum_j^M B_{ij}(m_i - m_i^0)(m_j - m_j^0)$$

Minimum: derivative zero

$$\frac{\partial Phi_d}{\partial m_k} = 0 = b_k + \sum_j B_{kj}(m_k - m_k^0) \Rightarrow \mathbf{m} - \mathbf{m}^0 = -\mathbf{B}^{-1}\mathbf{b}$$

Newtons method for linear problems

$$\Phi_d = (\mathbf{d} - \mathbf{Gm})^T(\mathbf{d} - \mathbf{Gm})$$

$$b_i = \frac{\partial \Phi_d}{\partial m_i} = -2\mathbf{G}^T(\mathbf{d} - \mathbf{Gm})$$

$$B_{ij} = \frac{\partial^2 \Phi_d}{\partial m_i \partial m_j} = 2\mathbf{G}^T\mathbf{G}$$

$\Rightarrow \mathbf{m} = -\mathbf{B}^{-1}\mathbf{b} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{d}$ (least-squares solution)

Newtons method for non-linear problems

$$\Phi_d = (\mathbf{d} - \mathbf{f}(\mathbf{m}))^T(\mathbf{d} - \mathbf{f}(\mathbf{m}))$$

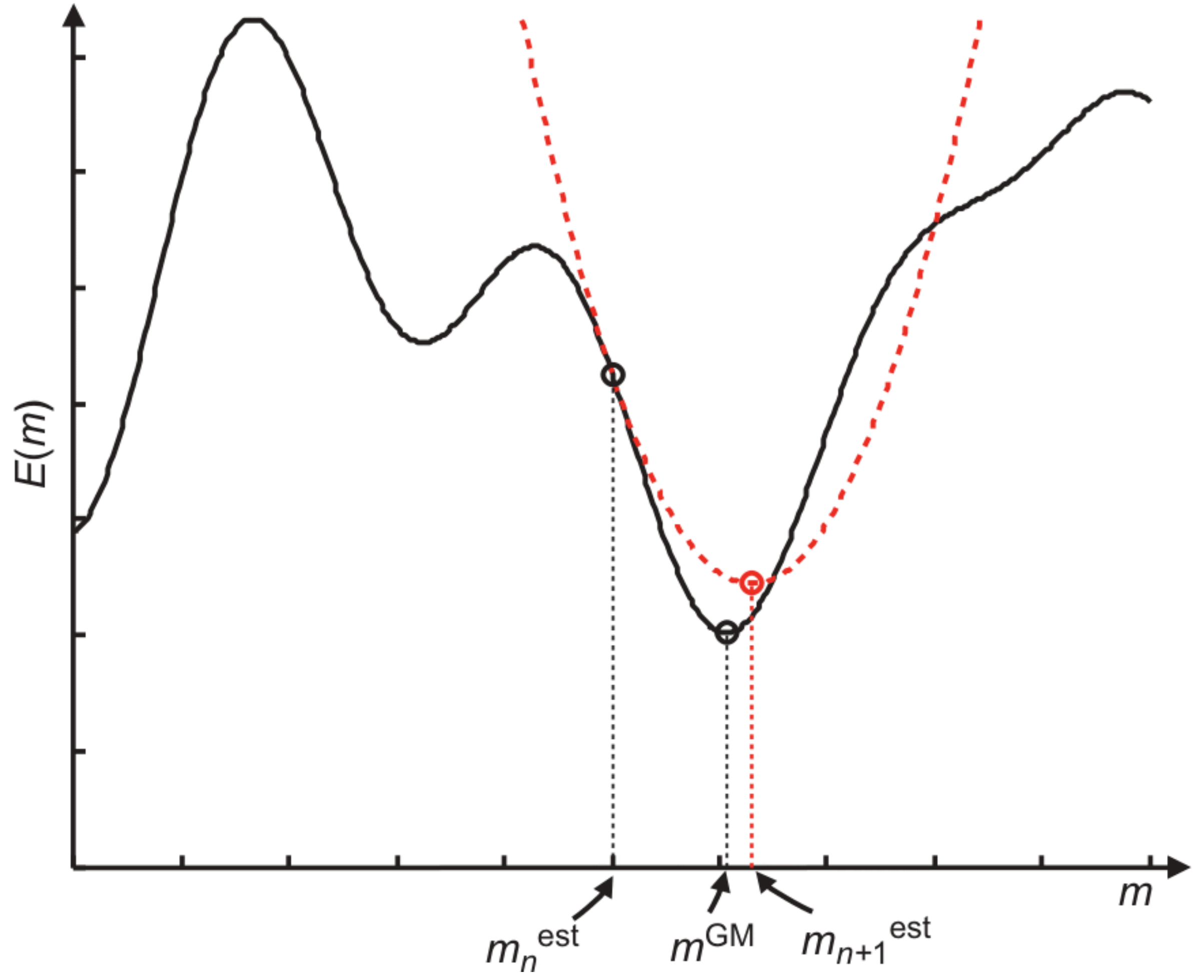
$$b_i = \frac{\partial \Phi_d}{\partial m_i} = -2\mathbf{S}^T(\mathbf{d} - \mathbf{f}(\mathbf{m})) \quad \text{with} \quad S_{ij} = \frac{\partial f_i(\mathbf{m}^0)}{\partial m_j}$$

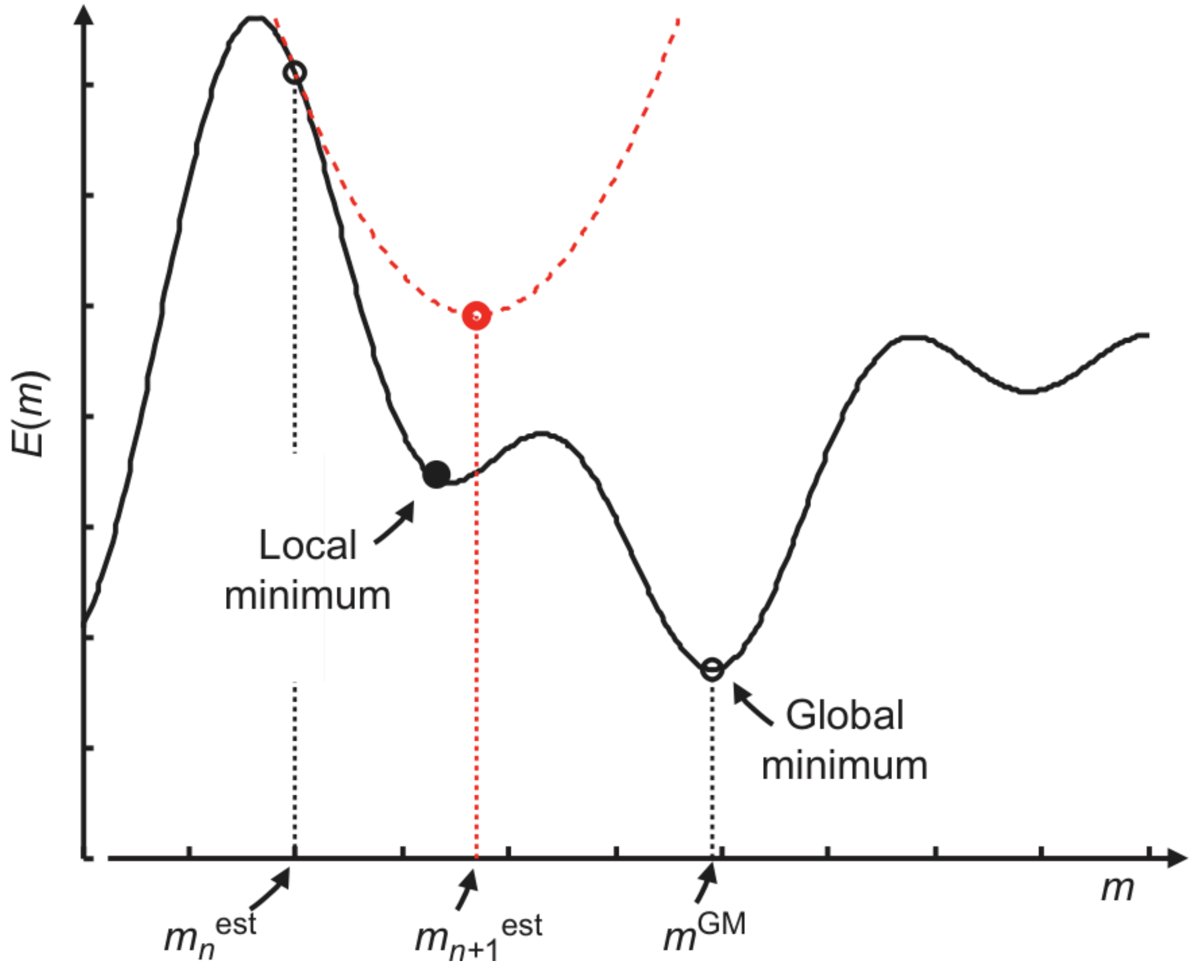
$$B_{ij} = \frac{\partial^2 \Phi_d}{\partial m_i \partial m_j} = \frac{\partial b_i}{\partial m_j} \approx 2\mathbf{S}^T\mathbf{S}$$

$$\Rightarrow \mathbf{m} - \mathbf{m}^0 = -\mathbf{B}^{-1}\mathbf{b} = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T(\mathbf{d} - \mathbf{f}(\mathbf{m}))$$

$\Rightarrow$ least-squares solution for $\Delta\mathbf{m}/\Delta\mathbf{d}$

Linearization (Menke, 2012)





Gauss-Newton minimization

1. Choose starting model $\mathbf{m}^0$ and set n=0
2. Compute model response $\mathbf{f}(\mathbf{m}^n)$
3. Compute sensitivity matrix $\mathbf{S}^n$
4. Solve linearized subproblem $\mathbf{S}^n \Delta\mathbf{m}^n = \Delta\mathbf{d} = (\mathbf{d} - \mathbf{f}(\mathbf{m}^n))$
5. Update model by $\mathbf{m}^{n+1} = \mathbf{m}^n + \Delta\mathbf{m}^n$
6. If convergence quit, otherwise $n \leftarrow n + 1$ & proceed with 2.

Line search for strong non-linearity

The Taylor approximation might change along $\mathbf{\Delta m}$, so we optimize the step length τ by “searching along the line”

$$\mathbf{m}^{n+1} = \mathbf{m}^n + \tau^n \mathbf{\Delta m}^n$$

- exact line search: testing $\mathbf{f}(\mathbf{m} + \tau \mathbf{\Delta m})$ for many τ
- interpolation of $\mathbf{f}(\mathbf{m}^n + \tau \mathbf{\Delta m})$ between $\mathbf{f}(\mathbf{m}^n)$ & $\mathbf{f}(\mathbf{m}^n + \mathbf{\Delta m})$
- fit parabola through points $\mathbf{f}(\mathbf{m}^n)$, $\mathbf{f}(\mathbf{m}^n + \mathbf{\Delta m}/2)$, $\mathbf{f}(\mathbf{m}^n + \mathbf{\Delta m})$

Gauss-Newton minimization

1. Choose starting model $\mathbf{m}^0$ and set n=0
2. Compute model response $\mathbf{f}(\mathbf{m}^n)$
3. Compute sensitivity matrix $\mathbf{S}^n$
4. Solve linearized subproblem $\mathbf{S}^n \Delta\mathbf{m}^n = \Delta\mathbf{d} = (\mathbf{d} - \mathbf{f}(\mathbf{m}^n))$
5. Update model by $\mathbf{m}^{n+1} = \mathbf{m}^n + \tau^n \Delta\mathbf{m}^n$
6. If convergence quit, otherwise $n \leftarrow n + 1$ & proceed with 2.

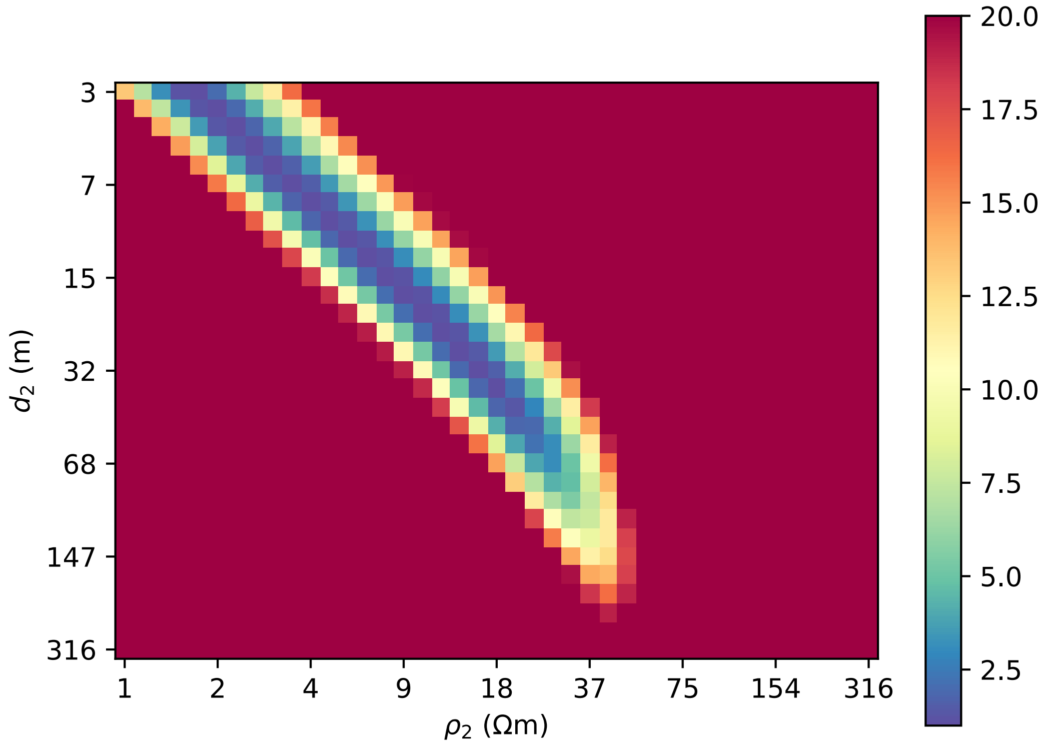
Computation of the sensitivity matrix

- analytically (derivation of the forward operator)
- transforming the PDE and its numerical solution
- perturbation method (brute force)

$$\frac{\partial f_i(\mathbf{m})}{\partial m_j} \approx \frac{f_i(\mathbf{m} + \delta_j \Delta m) - f_i(\mathbf{m})}{\Delta m}$$

with the Dirac vector $\delta_j = [0, \dots, 0, 1, 0, \dots, 0]^T$
$\Rightarrow$ one full forward computation for every model parameter

Model transformation



Objective function for VES

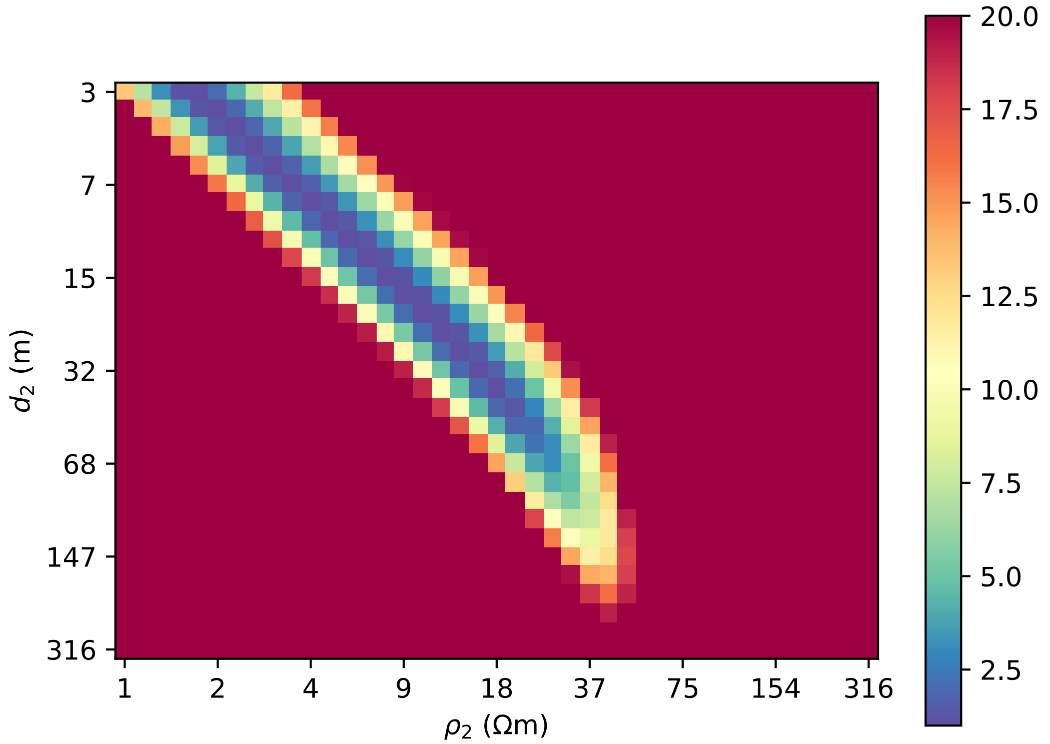
To keep the parameters positive, we often invert for the logarithms.

If we invert for $\hat{m}$ instead of m, we use the chain rule

$$\frac{\partial f}{\partial \hat{m}} = \frac{\partial f}{\partial m} \cdot \frac{\partial m}{\partial \hat{m}} = \frac{\partial f}{\partial m} \Big/ \frac{\partial \hat{m}}{\partial m}$$

E.g. $\partial \log \rho / \partial \rho = 1/\rho$

Data transformation



Objective function for VES

Often, measured data show a wide range so that we use the logarithm

If we invert $\hat{d}$ instead of d, we use the chain rule

$$\frac{\partial \hat{f}}{\partial m} = \frac{\partial f}{\partial m} \cdot \frac{\partial \hat{f}}{\partial f}$$

E.g. $\partial \log \rho^a / \partial \rho^a = 1/\rho_a$

Combined model and data transformation

Sensitivity $S_{ij} = \frac{\partial \rho_i^a}{\partial \rho_j}$

Data $d_i = \log \rho_i^a$, model parameter $m_i = \log \rho_j$

$\Rightarrow$ Jacobian matrix

$$J_{ij} = \frac{\partial \log \rho_i^a}{\partial \log \rho_j} = \frac{\partial \rho_i^a}{\partial \rho_j} \cdot \frac{\rho_j}{\rho_i^a}$$

Regularization

$$\Phi = (\mathbf{d} - \mathbf{f}(\mathbf{m}))^T(\mathbf{d} - \mathbf{f}(\mathbf{m})) + \lambda(\mathbf{c} - \mathbf{Cm})^T(\mathbf{c} - \mathbf{Cm})$$

$$b_i = \frac{\partial \Phi}{\partial m_i} = -2\mathbf{S}^T(\mathbf{d} - \mathbf{f}(\mathbf{m})) - 2\lambda\mathbf{C}^T(\mathbf{c} - \mathbf{Cm})$$

$$B_{ij} = \frac{\partial^2 \Phi}{\partial m_i \partial m_j} = \frac{\partial b_i}{\partial m_j} \approx 2\mathbf{S}^T\mathbf{S} + 2\mathbf{C}^T\mathbf{C}$$

$$\Rightarrow (\mathbf{S}^T\mathbf{S} + \lambda\mathbf{C}^T\mathbf{C})\Delta\mathbf{m} = \mathbf{S}^T(\mathbf{d} - \mathbf{f}(\mathbf{m}^n)) + \lambda\mathbf{C}^T(\mathbf{c} - \mathbf{Cm})$$