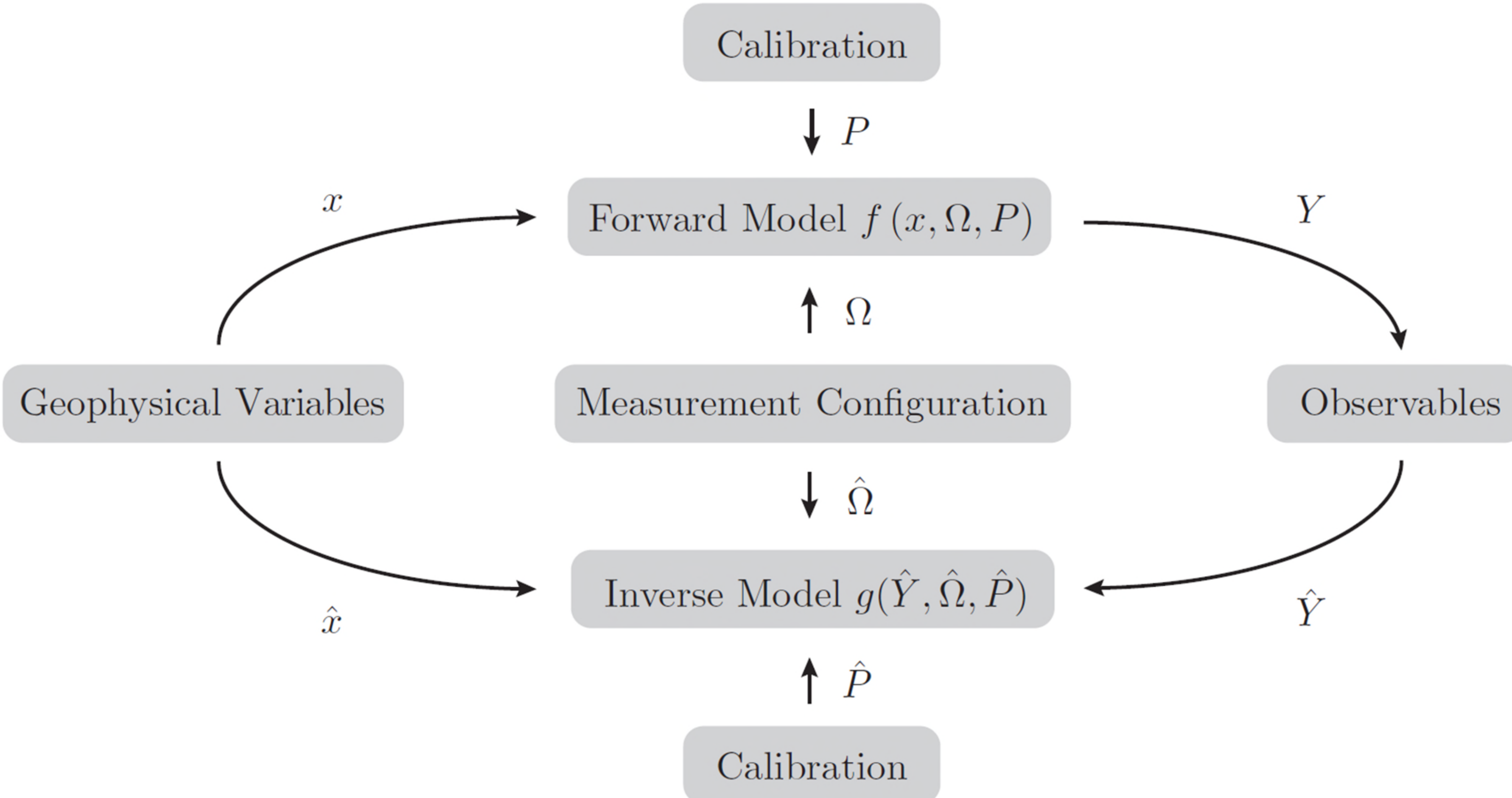
Chapter 5
Model Inversion

5.1 Approaches to Model Inversion

5.2 Direct Inversion

5.3 Iterative Nonlinear Optimization

5.4 Neural Networks and Lookup Tables


Calibration
P
x
Forward Model $f(x, \Omega, P)$
Y
Ω
Geophysical Variables
Measurement Configuration
Observables
$\hat{\Omega}$
Inverse Model $g(\hat{Y}, \hat{\Omega}, \hat{P})$
$\hat{x}$
$\hat{Y}$
$\hat{P}$
Calibration

5.1 Approaches to Model Inversion

- **Direct Inversion**
  - When $f(\cdot)$ is simple, with unambiguous relationships between y and x, then it is possible to analytically derive $g(\cdot) = f^{-1}(\cdot)$

- **Iterative nonlinear optimization**
  - When $f(\cdot)$ becomes more complex, with ambiguous relationships between y and x, then all input values into $f(\cdot)$ are iteratively adjusted until the forward modelled values of $\hat{y} = f\left(\hat{x}, \widehat{\Omega}, \hat{p}\right)$ are “close” to the observed values $\hat{y}$.
  - $g(\cdot)$ is not derived

- **Approximation**
  - $f(\cdot)$ becomes too complex even for an iterative optimization approach
  - Introduce simplifying assumptions and/or find approximate functions of $f(\cdot)$
    - E.g. train a neural network with $f(\cdot)$ and run it inverse to obtain $g(\cdot)$

5.2 Direct Inversion

- When $y = f(x)$ can be inverted without ambiguities then

$$\hat{x} = f^{-1}(\hat{y})$$

- An example of a direct inversion is the linear bare soil backscatter model

$$\sigma^0_{soil}[dB] = A + Bm_v \quad \Longrightarrow \quad m_v = \frac{\sigma^0_{soil} - A}{B}$$

  - Model parameters A and B depend on soil roughness and texture and hence change in space (and time)

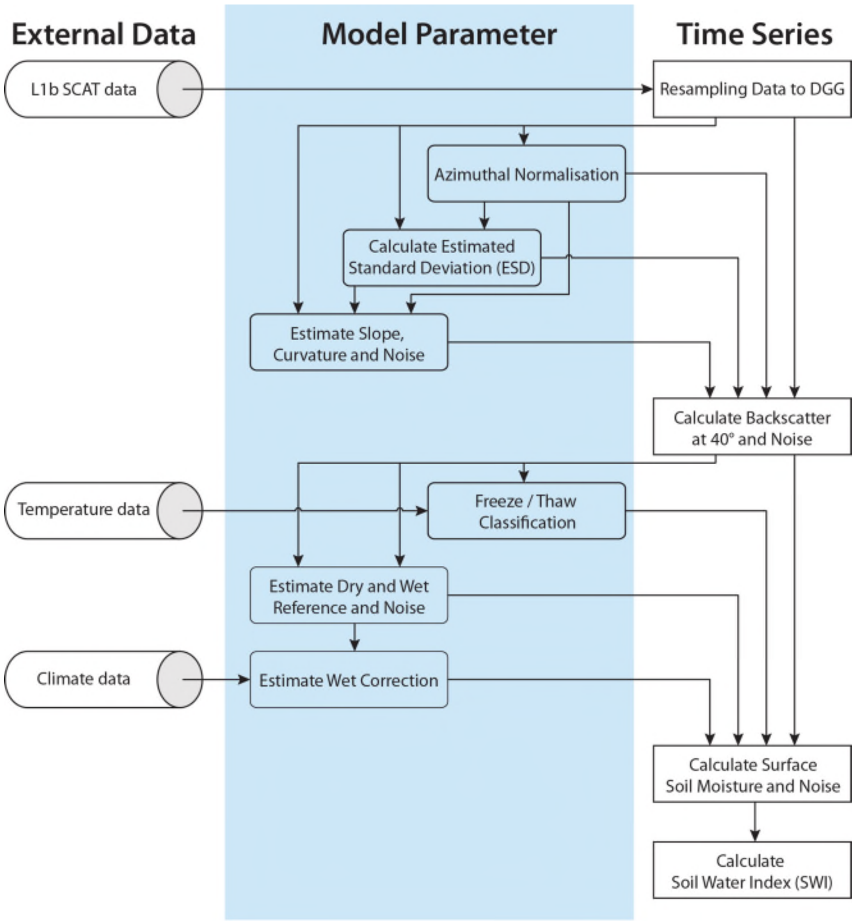
TU Wien model

$$\sigma^\circ(\theta, m_s, s, V) = \sigma^\circ{}_{dry}(\theta, s, V) + m_s[\sigma^\circ{}_{wet}(\theta, s, V) - \sigma^\circ{}_{dry}(\theta, s, V)]$$

- Spatially and temporally varying
  - $\sigma^\circ{}_{dry}$ dry reference
  - $\sigma^\circ{}_{wet}$ wet reference

Inversion of TU Wien Backscatter Model

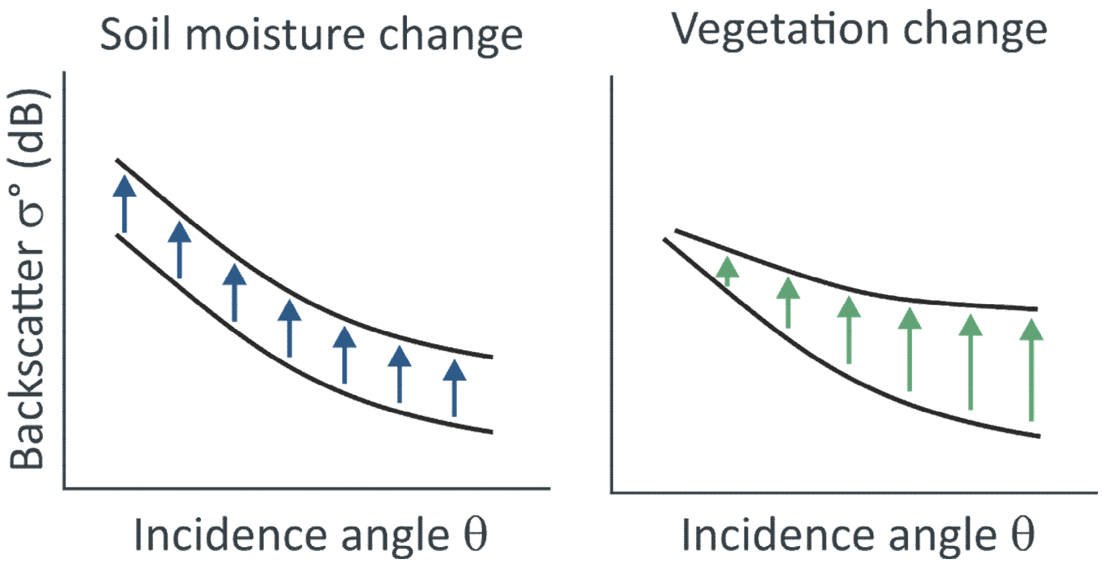
- TU Wien soil moisture retrieval is achieved by a stepwise (direct) inversion of TU Wien backscatter model
  - Stepwise ≠ iterative



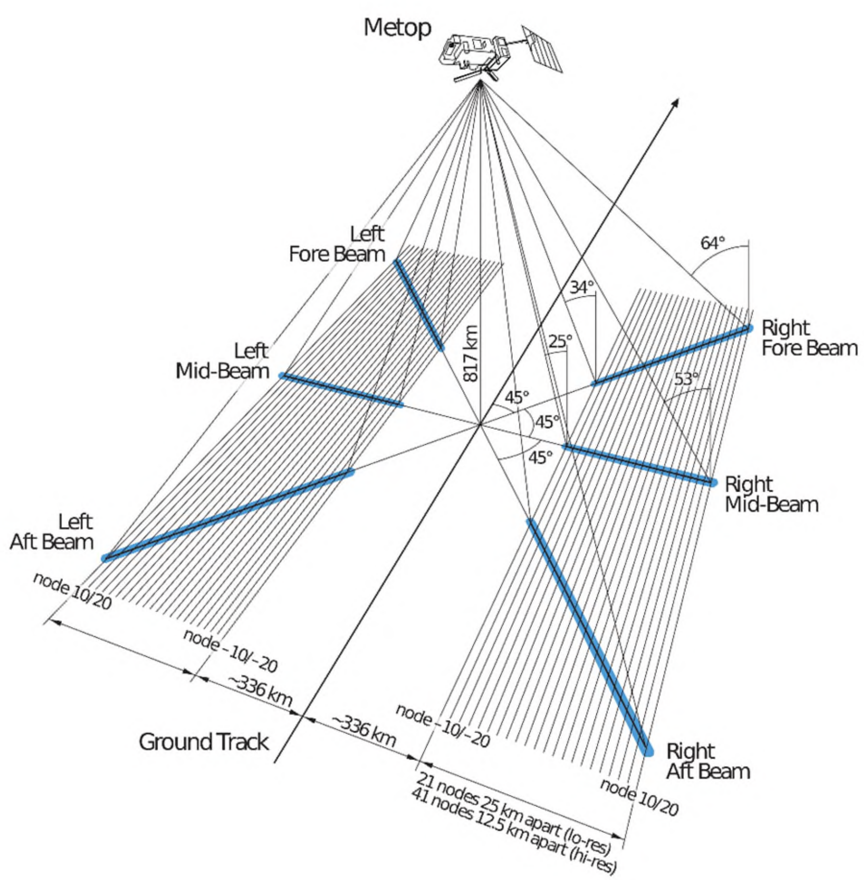
- Main processing steps (out of over a dozen)
  - Modelling of backscatter vs. incidence angle relationship
  - Incidence angle normalisation
  - Cross-over angles (correction of seasonal vegetation effects)
  - Dry- and wet references (model calibration)
  - Soil moisture estimation

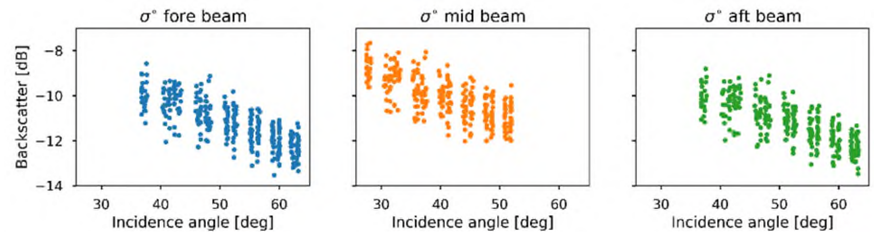
Backscatter vs. Incidence Angle

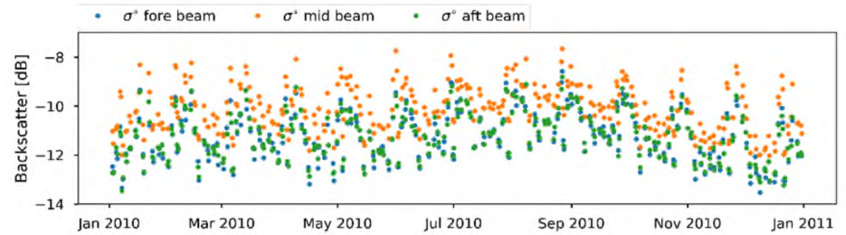
$$\underbrace{\sigma^0(\theta,t)}_{\text{measure}} = \sigma^0(\theta_{ref},t) + \underbrace{\sigma'(\theta_{ref},t)}_{\text{slope}}(\theta - \theta_{ref}) + \frac{1}{2}\underbrace{\sigma''(\theta_{ref},t)}_{\text{curvature}}(\theta - \theta_{ref})^2$$

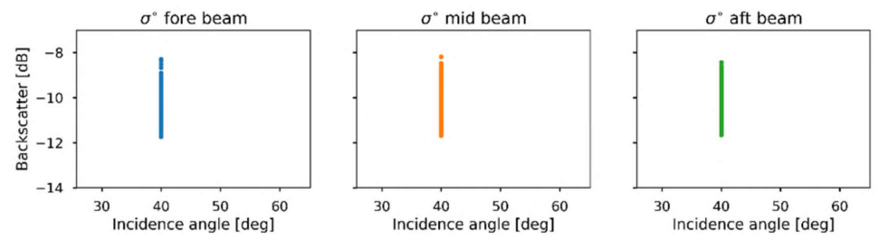


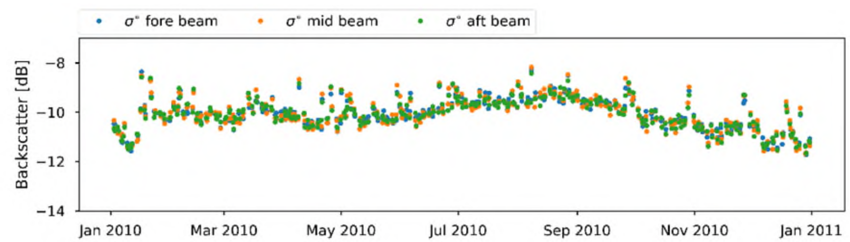
ASCAT Measurement Triplet





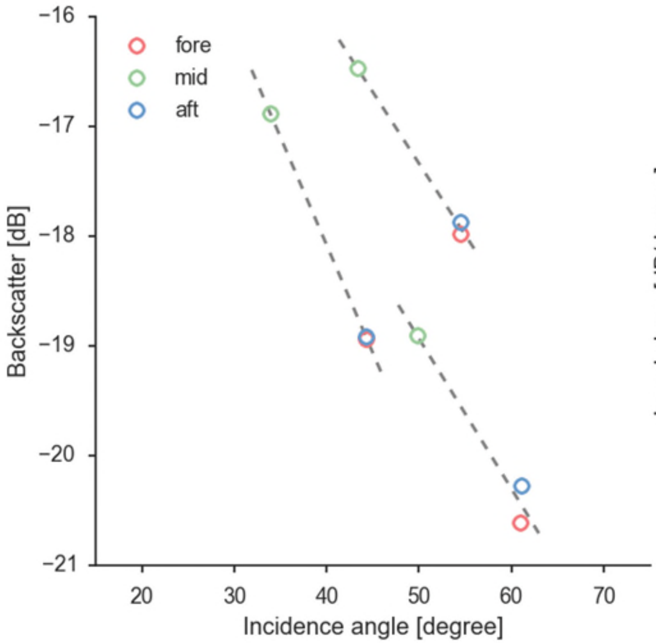


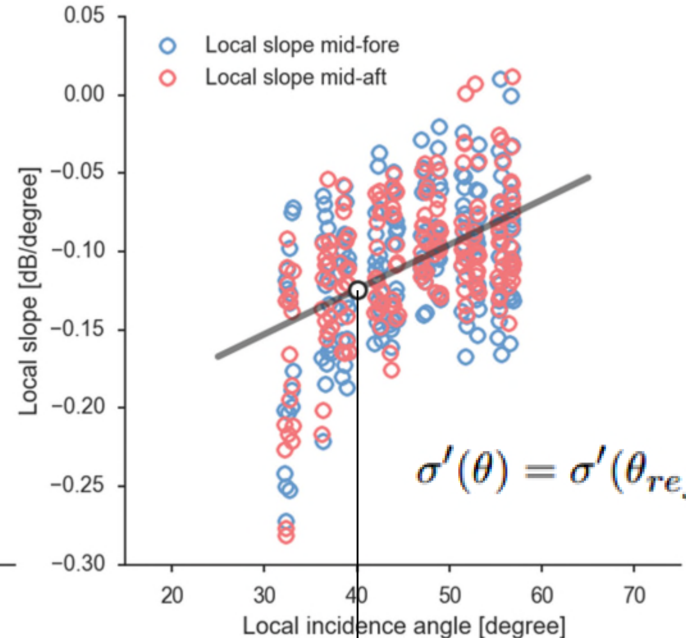




Estimating Slope and Curvature

- As ASCAT measures a backscatter triplet the local slope can be calculated, being quasi an – albeit noisy - observable





$$\sigma'(\theta) = \sigma'(\theta_{ref}) + \sigma''(\theta_{ref})(\theta - \theta_{ref})$$

Reference angle θ_{ref}

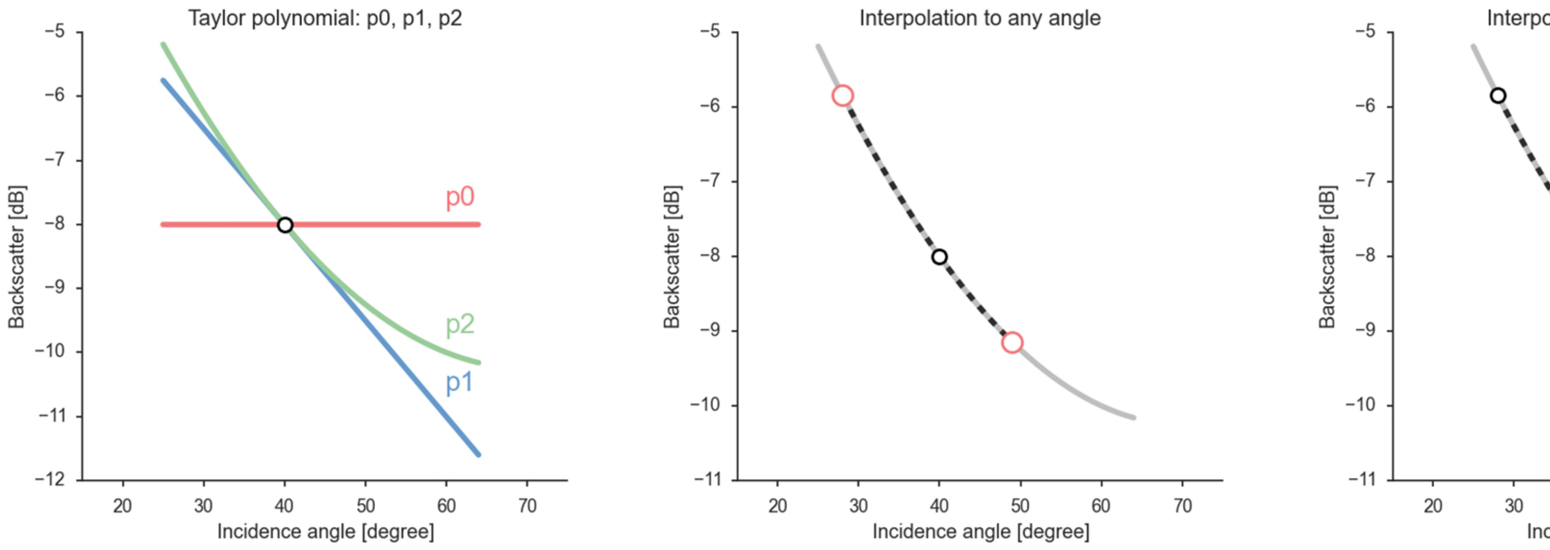
$$\sigma'\left(\frac{\theta_{mid}-\theta_{fore}}{2}\right)=\frac{\sigma^0_{mid}(\theta_{mid})-\sigma^0_{fore}(\theta_{fore})}{\theta_{mid}-\theta_{fore}}$$

$$\sigma'\left(\frac{\theta_{mid}-\theta_{aft}}{2}\right)=\frac{\sigma^0_{mid}(\theta_{mid})-\sigma^0_{aft}(\theta_{aft})}{\theta_{mid}-\theta_{aft}}$$

Incidence Angle Normalisation

- To make σ^0 measurements acquired from different orbits / incidence angles comparable, they are normalised to the reference angle θ_{ref}

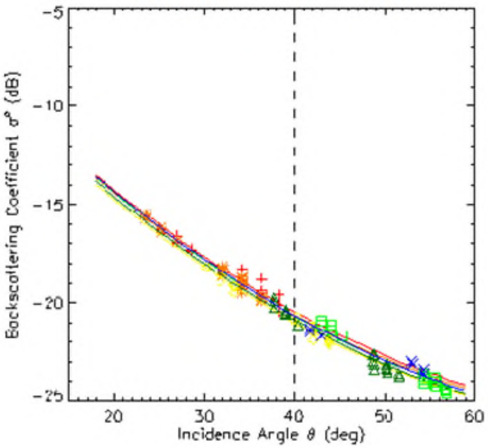
$$\sigma^0(\theta_{ref}, t) = \sigma^0(\theta, t) - \sigma'(\theta_{ref}, doy)(\theta - \theta_{ref}) - \frac{1}{2}\sigma''(\theta_{ref}, doy)(\theta - \theta_{ref})^2$$



Impact of Land Cover / Biomass

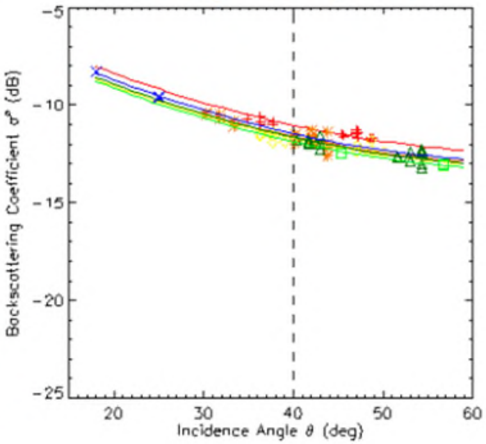
Steppe
Mongolia - Choybalsan (115E 48N)
pot. Biomass: 300 g/m²

28 Nov - 6 Dec 1993



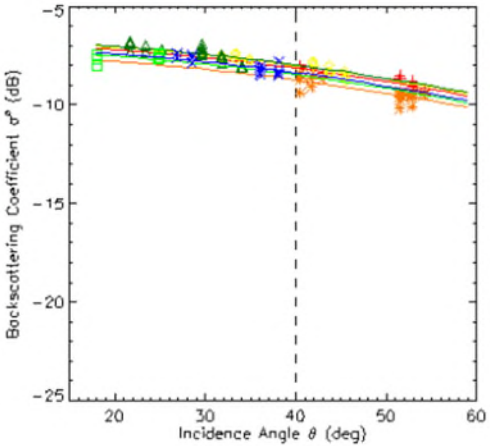
Temperate Zone
Russia - Roslav (33E 54N)
pot. Biomass: 900 g/m²

16 Sep - 24 Sep 1995

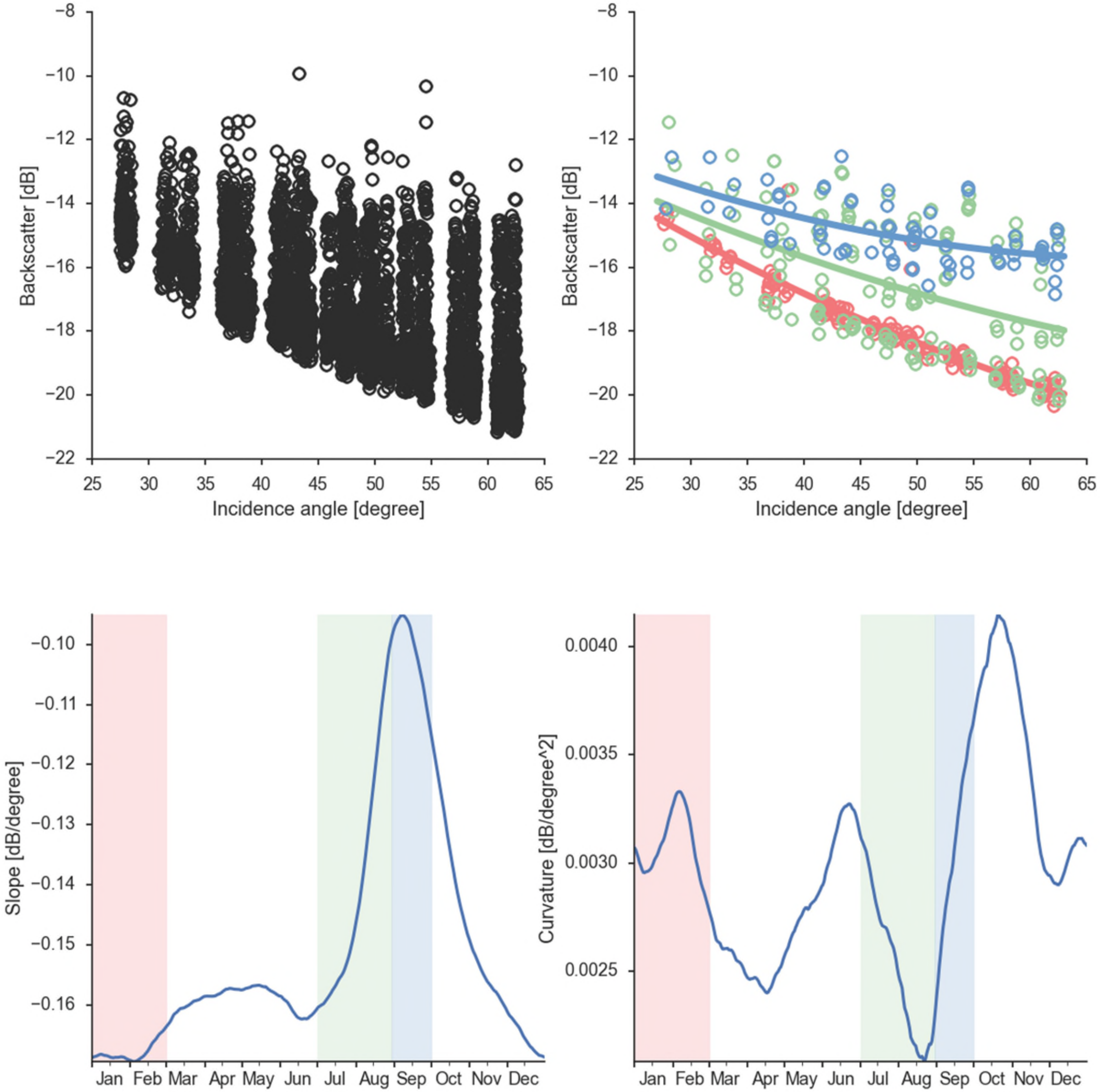


Tropical Forest
CAR - Berberati (16E 4N)
pot. Biomass: 1920 g/m²

15 Apr - 26 Apr 1997



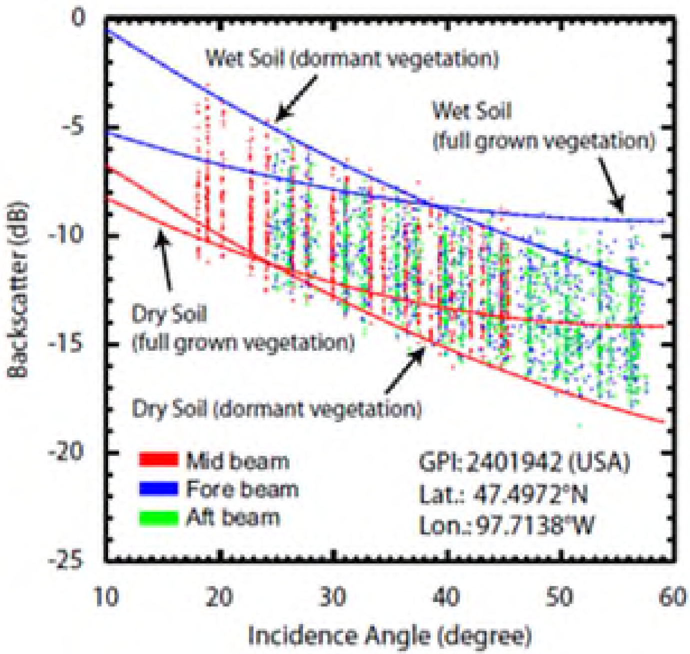
Impact of Season



Cross-Over Angles

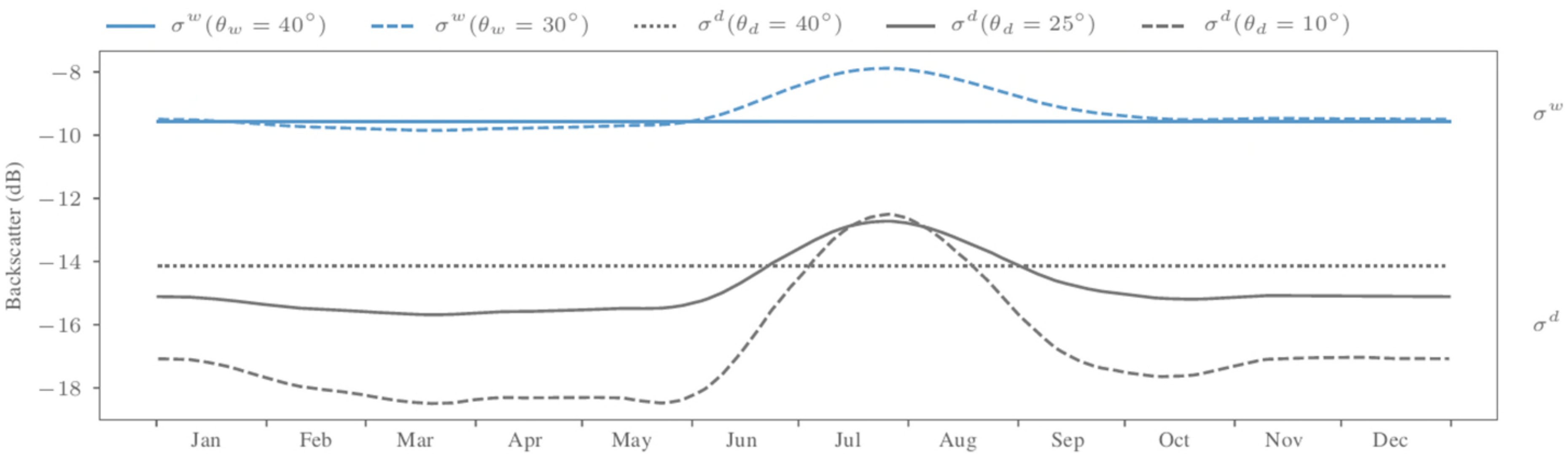
- From the $\omega - \tau$ model one can deduce that there may be an incidence angle where the effects of vegetation development cancel each other out
  - Assuming that vegetation phenology impacts mainly τ but not ω

$$\frac{\partial \sigma^0(\theta)}{\partial \gamma^2} = 0 \quad \longrightarrow \quad \frac{3\omega cos(\theta_{cross})}{4} = \sigma^0_{soil}(\theta_{cross})$$

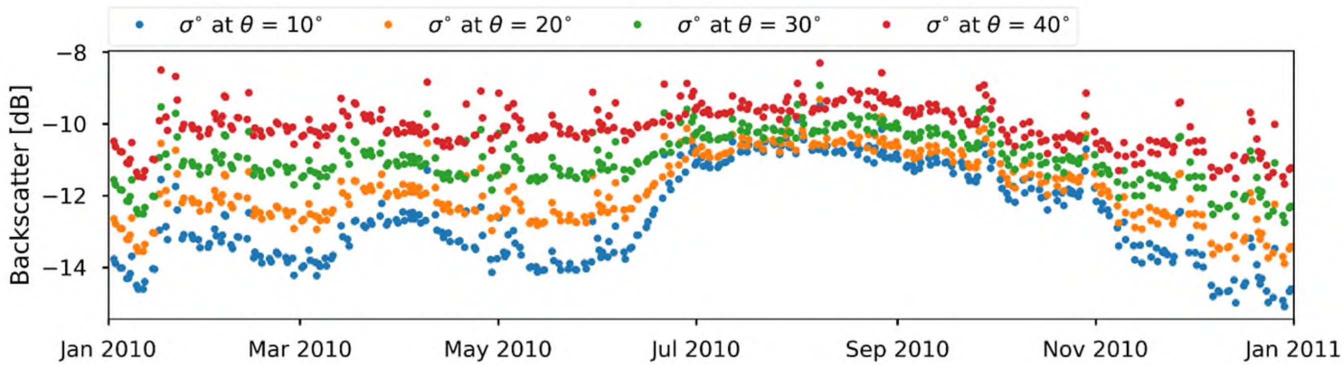


Strength of Vegetation Effects

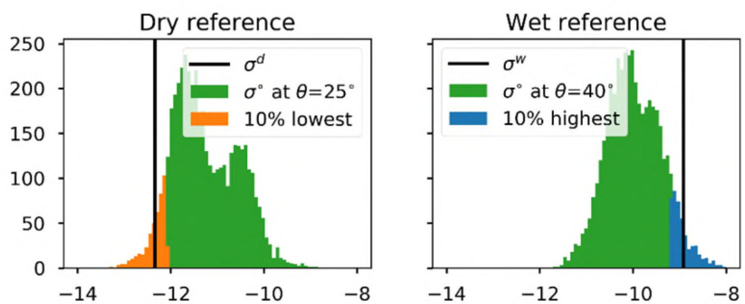
- The strength of seasonal vegetation effects on σ^0 at the reference angle (e.g. $\theta_{ref} = 40°$) depends on the choice of the cross-over angles for dry (θ_d) and wet (θ_w) conditions
  - The larger the distances $\left(\theta_{ref} - \theta_d\right)$ respectively $\left(\theta_{ref} - \theta_w\right)$ the stronger the correction
  - θ_d and θ_w are model parameters that need to be calibrated



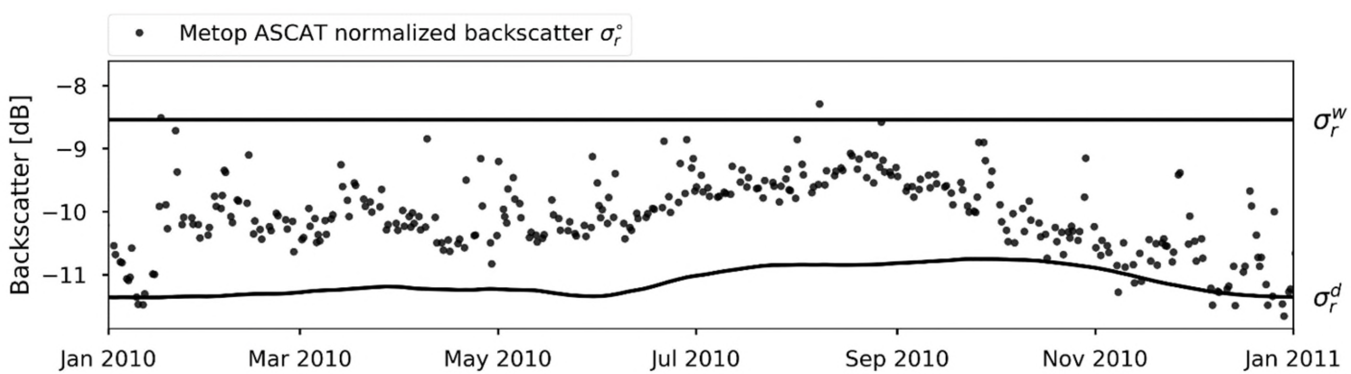
Estimation/Calibration of Backscatter References



Interpolation to arbitrary incidence angle possible for complete backscatter time series



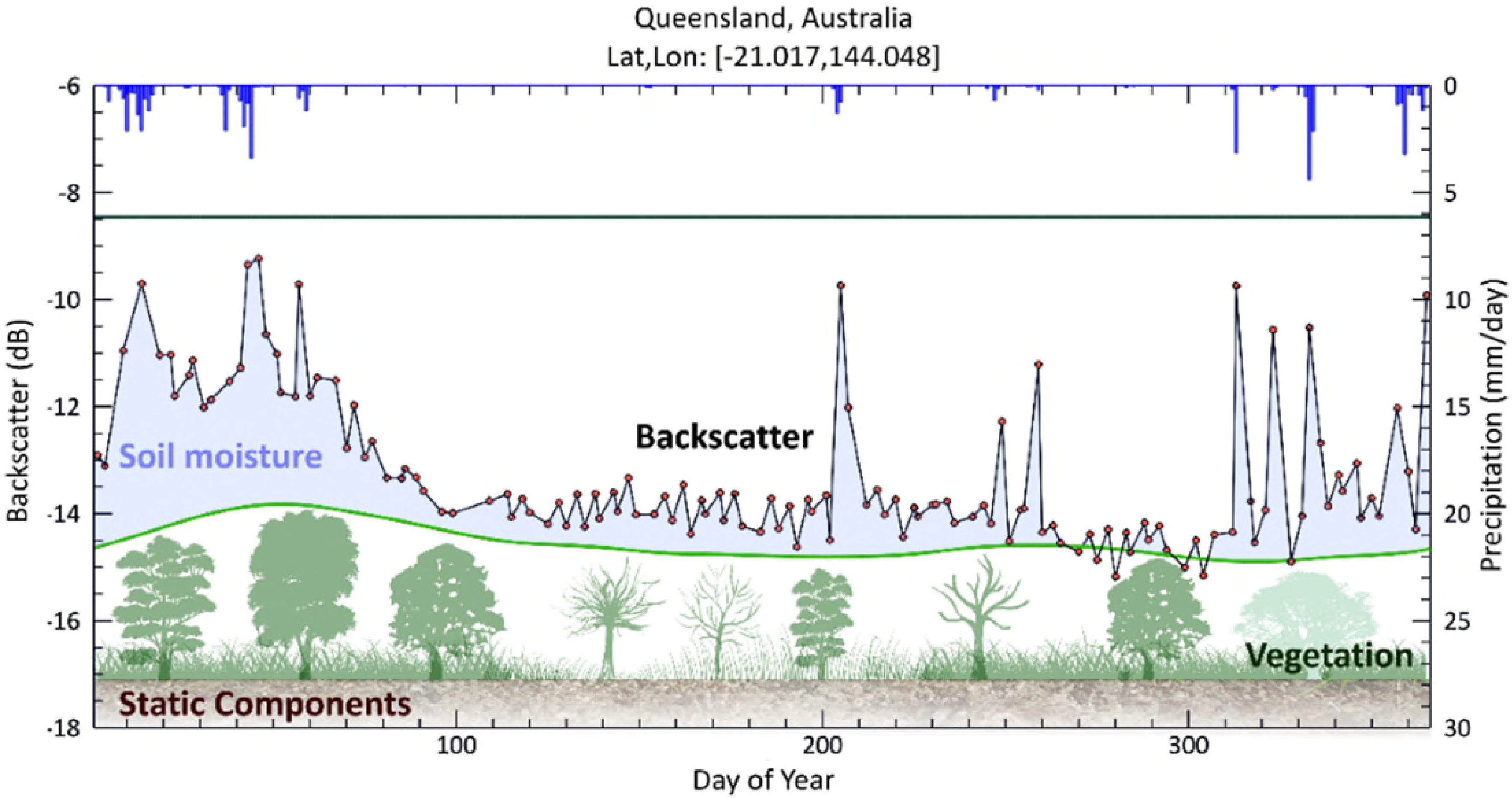
Estimation of dry and wet backscatter reference at dry and wet cross-over angle



Soil Moisture Estimation

$$m_s = \frac{\sigma^0(\theta, m_s, s, V) - \sigma^0_{dry}(\theta, s, V)}{\sigma^0_{wet}(\theta, s, V) - \sigma^0_{dry}(\theta, s, V)}$$

m_s … degree of saturation (0-1)



5.3 Iterative Nonlinear Optimization

- Nonlinear function minimization techniques are used to compute model parameters (calibration), and/or predict the values of the geophysical variable(s) of interest (model inversion) from observations

- To find $\hat{x}$ (and/or $\mathbf{p}$) given $\mathbf{\Omega}_i$ a cost or objective function $h(\cdot)$ is minimized through an iterative adjustment of the inputs x (and/or $\mathbf{p}$) into $f(\cdot)$

- The most important scenario is the minimization of a sum of squared differences

$$h(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{N}(f(\mathbf{x}, \mathbf{\Omega}_i, \mathbf{p}) - y_i)^2, \mathbf{x} \in \mathbb{R}^p$$

Iterative Procedure

- Gradient descent
- Newton‘s method
- Least squares and Gauss-Newton Method
- Levenberg Marquardt Method

Gradient descent

- **first-order iterative optimization** algorithm for finding a local minimum of a differentiable function

$$\mathrm{x}_{t+1} = \mathrm{x}_t - \alpha_t \nabla h(\mathrm{x})$$

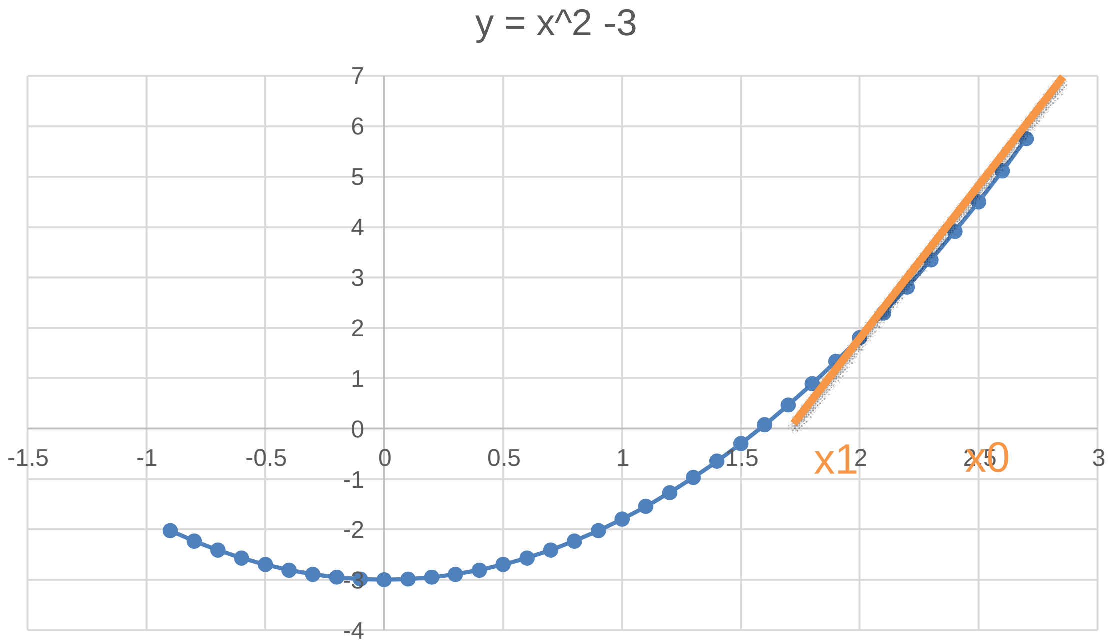
- $\nabla h(\mathrm{x})$ points in direction along which h(·) decreases fastest

- take repeated steps in the **opposite direction of the gradient**

- Can converge at local minimum or saddle point

Newton's Method

- **Comes from Newton's method for zero-finding**



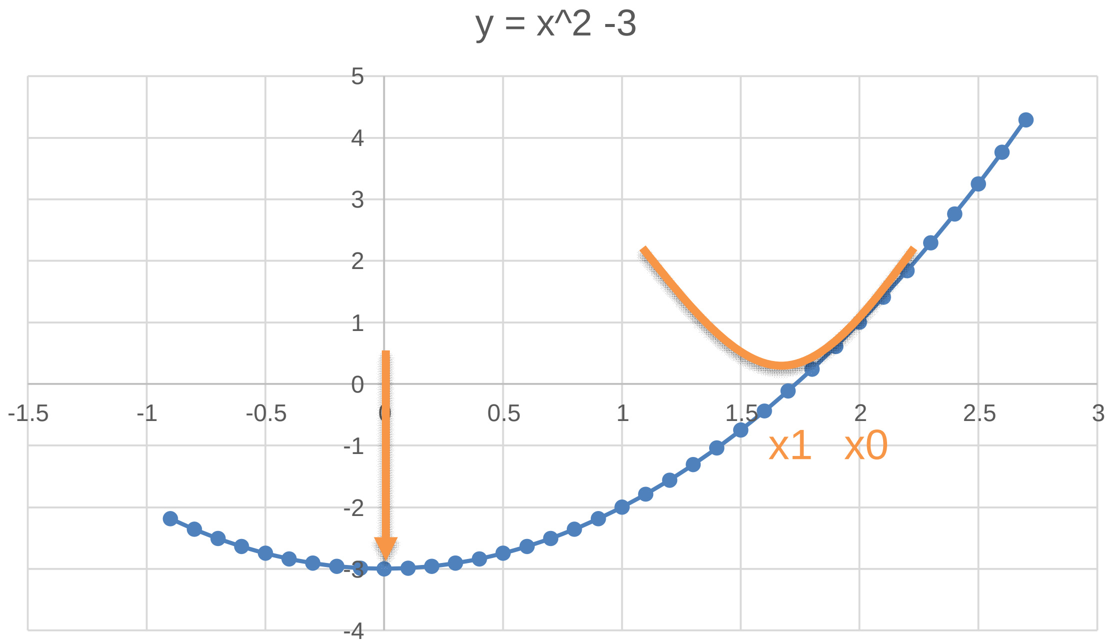
$$tan\theta = \frac{h(x_0)}{x_0 - x_1}$$

$$h'(x_{t0}) = \frac{h(x_0)}{x_0 - x_1}$$

$$x_{t+1} = x_t - \frac{h(x_t)}{h'(x_t)}$$

Newton's Method

- **To find minimum →Try to set derivative to 0**



$$x_{t+1} = x_t - \frac{h'(x_t)}{h''(x_t)}$$

Newton's Method

- For more accuracy make a second order approximation and minimize that
- Based on Taylor equation

$$h(\mathrm{x}\quad) \approx \underbrace{\boxed{h(\mathrm{x}_k) + \nabla h(\mathrm{x}_k)^T(\mathrm{x} - \mathrm{x}_k)}}_{\text{Linear part}} + \underbrace{\boxed{\frac{1}{2}(\mathrm{x} - \mathrm{x}_k)^T \nabla^2 h(\mathrm{x}_k)\quad(\mathrm{x} - \mathrm{x}_k)}}_{\text{Quadratic part}}$$

- $\nabla^2 h(\mathrm{x}_k) \quad = \frac{\partial^2 h(\mathrm{X})}{\partial x_i \partial x_j} = H$ Hessian of all partial derivatives in directions i and j

- Obtain it's gradient by differentiating w.r.t x:

$$\nabla h(\mathrm{x}\quad) \quad \approx \nabla h(\mathrm{x}_k) \quad + \nabla^2 h(\mathrm{x}_k) \quad (\mathrm{x} - \mathrm{x}_k)$$

Newton's Method

- Minimum: gradient vanishes!
- $\nabla h(\mathrm{x}\quad) = \nabla h(\mathrm{x}_{k+1}) = 0$

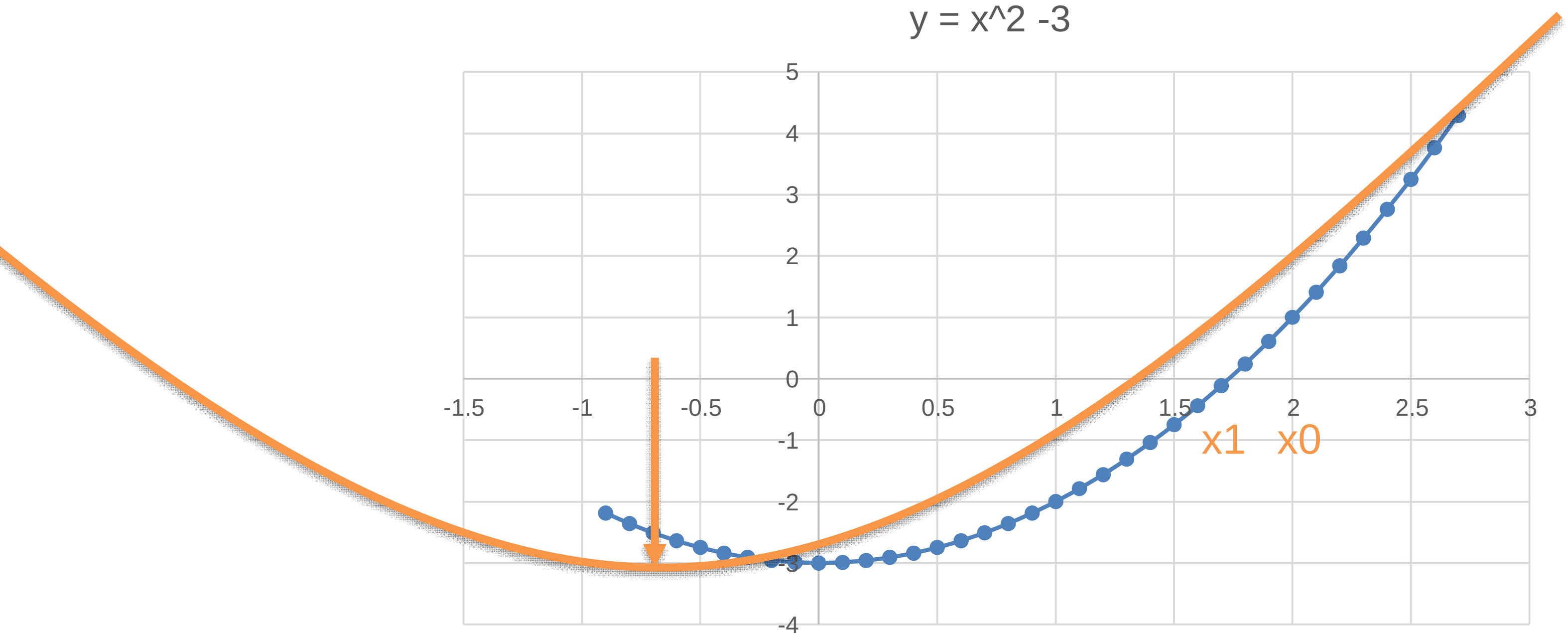
$$-\nabla h(\mathrm{x}_k) = \nabla^2 h(\mathrm{x}_k)\ (\mathrm{x}_{k+1} - \mathrm{x}_k)$$

- If Hessian is convertible this can be solved for x k+1

$$\mathrm{x}_{k+1} = \mathrm{x}_k - \nabla^2 h(\mathrm{x}_k)^{-1} \nabla h(\mathrm{x}_k\ )$$

Newton's Method

- Newton‘s method can overshoot, i.e not minimizing but maximizing



- Use Gradient descent with Newton: Marquardt-Levenberg

5.4 Approximations

- Numerical problems and/or performance considerations may impede the application of iterative nonlinear optimization techniques
- This forces one to introduce simplifying assumptions and/or find approximate functions of $f(\cdot)$

- **Lookup Tables**
  - Tabulate the output y_i of the forward model for a discrete subset of parameters x_i and features Ω_i
  - For a given observation y, the nearest tabulated output $argmin|y_i - y|$ is found, and its associated parameter vector x_i is returned as solution
  - Performance depends heavily on number of parameters and required accuracy

Artificial Neural Network

- A multi-layer perceptron (MLP) is a class of artificial neural network
- Uses a very general class of functions $\hat{f}_{MLP}$ that can approximate any sufficiently smooth target function
- Hierarchy of several layers of simple processing units or 'neurons'

- Network is trained to minimize e.g. least squares.
  - amount of error in the output compared to the expected result
  - Supervised learning