A prototype of a data assimilation system based automatic differentiation

Thomas Kaminski[1], Ralf Giering[1],

Marko Scholze[2], Peter Rayner[3], Wolfgang Knorr[4]

Copy of presentation at http://www.FastOpt.com

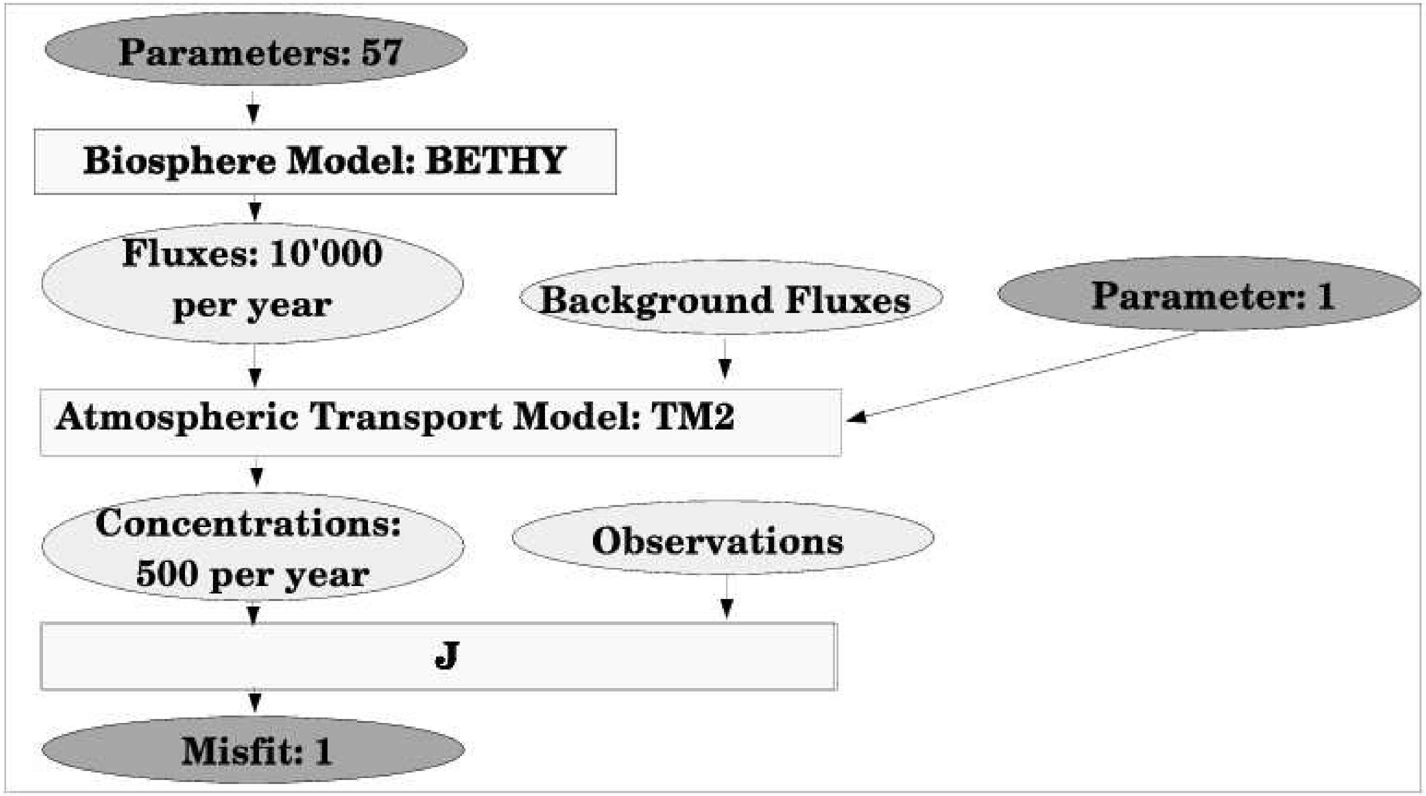
4 Max-Planck-Institut für Biogeochemie MPI-BGC Jena 2 Max-Planck-Institut für Meteorologie Max Planck Institute for Meteorology 3 CSIRO 1 FastOpt

CAMELS

Overview

- Calibration step
- Prognostic step
- Model development within system
- Automatic Differentiation
- Summary

Setup for Calibration Step



BETHY: Knorr 97; TM2: Heimann 95

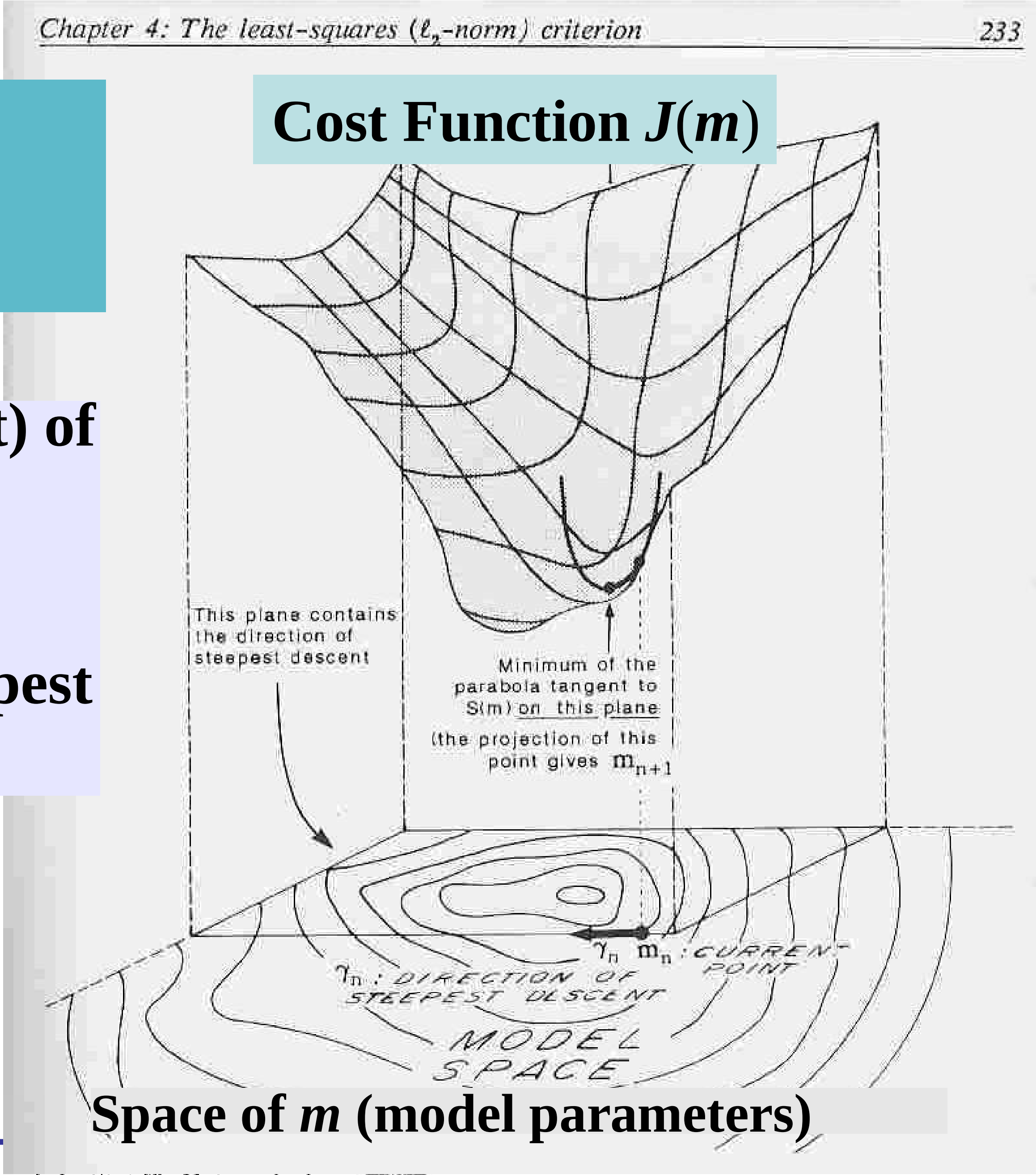
Gradient Method

First derivative (Gradient) of *J*(*m*) w.r.t. *m* (model parameters) :

$$-\partial J(m)/\partial m$$

yields direction of steepest descent

Figure taken from **Tarantola** '87



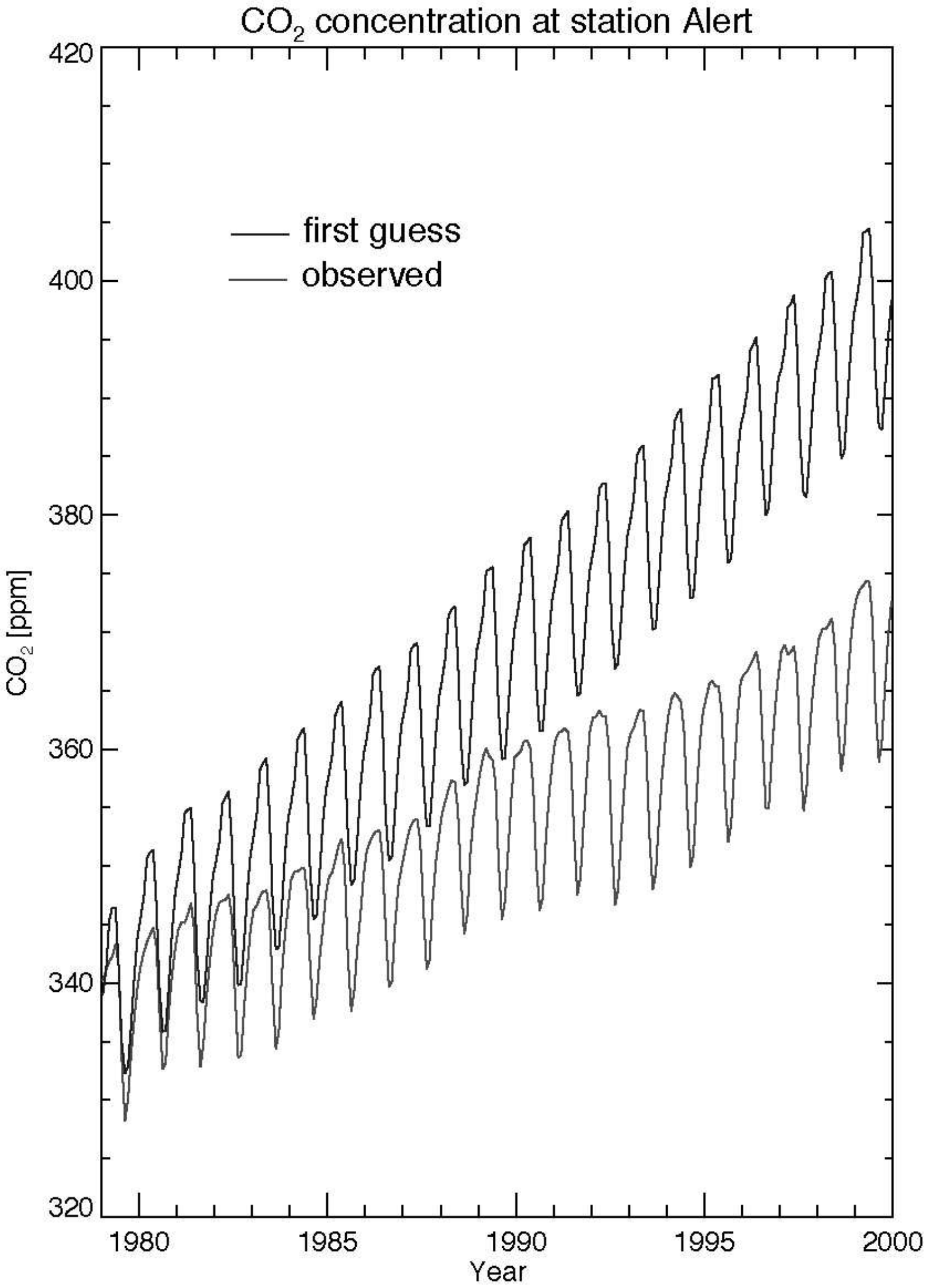
Chapter 4: The least-squares (ℓ_2-norm) criterion 233

Cost Function *J*(*m*)

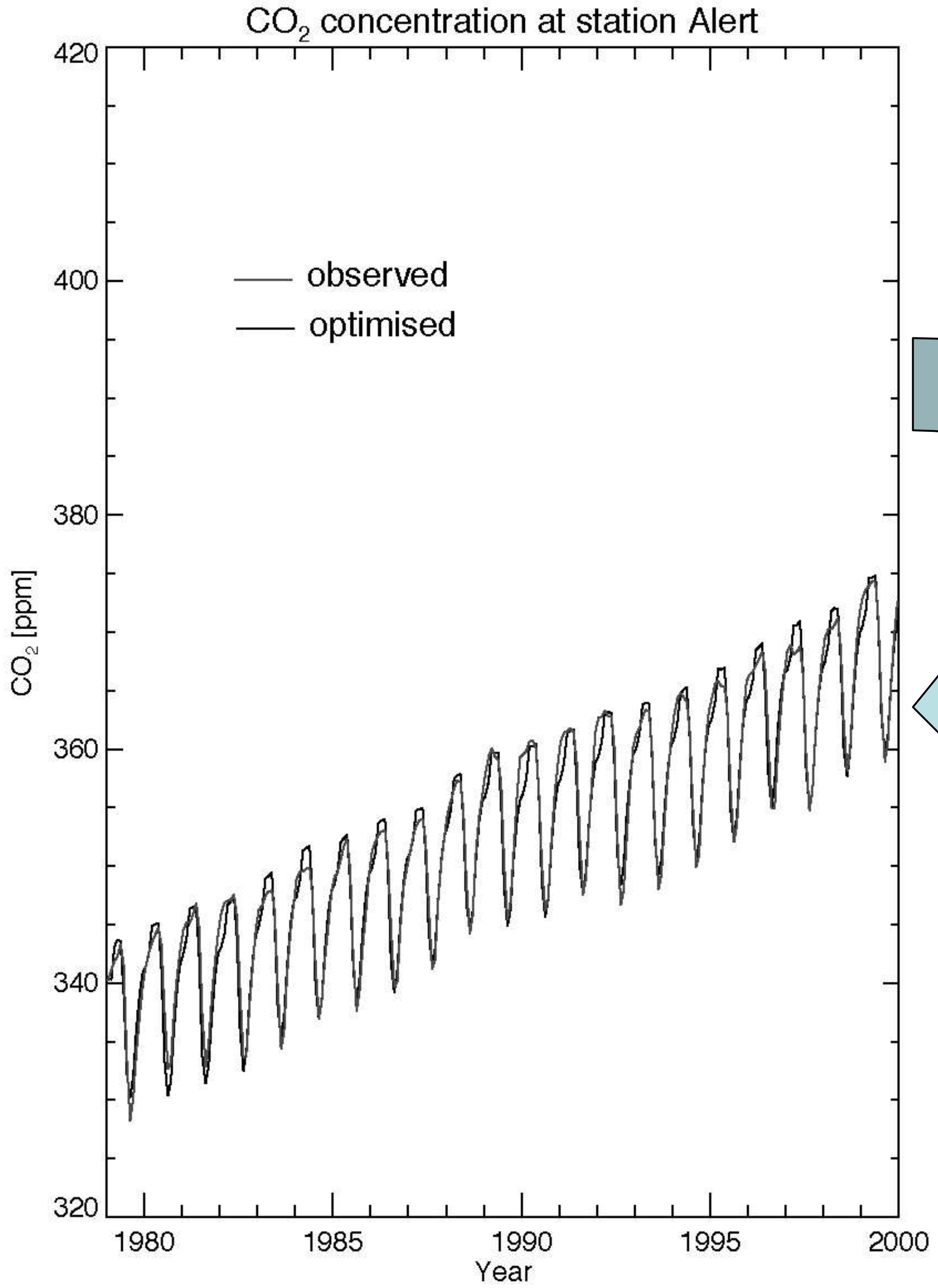
This plane contains the direction of steepest descent
Minimum of the parabola tangent to S(m) on this plane
(the projection of this point gives m_{n+1}
γ_n m_n : CURRENT POINT
γ_n : DIRECTION OF STEEPEST DESCENT
MODEL SPACE

Space of *m* (model parameters)

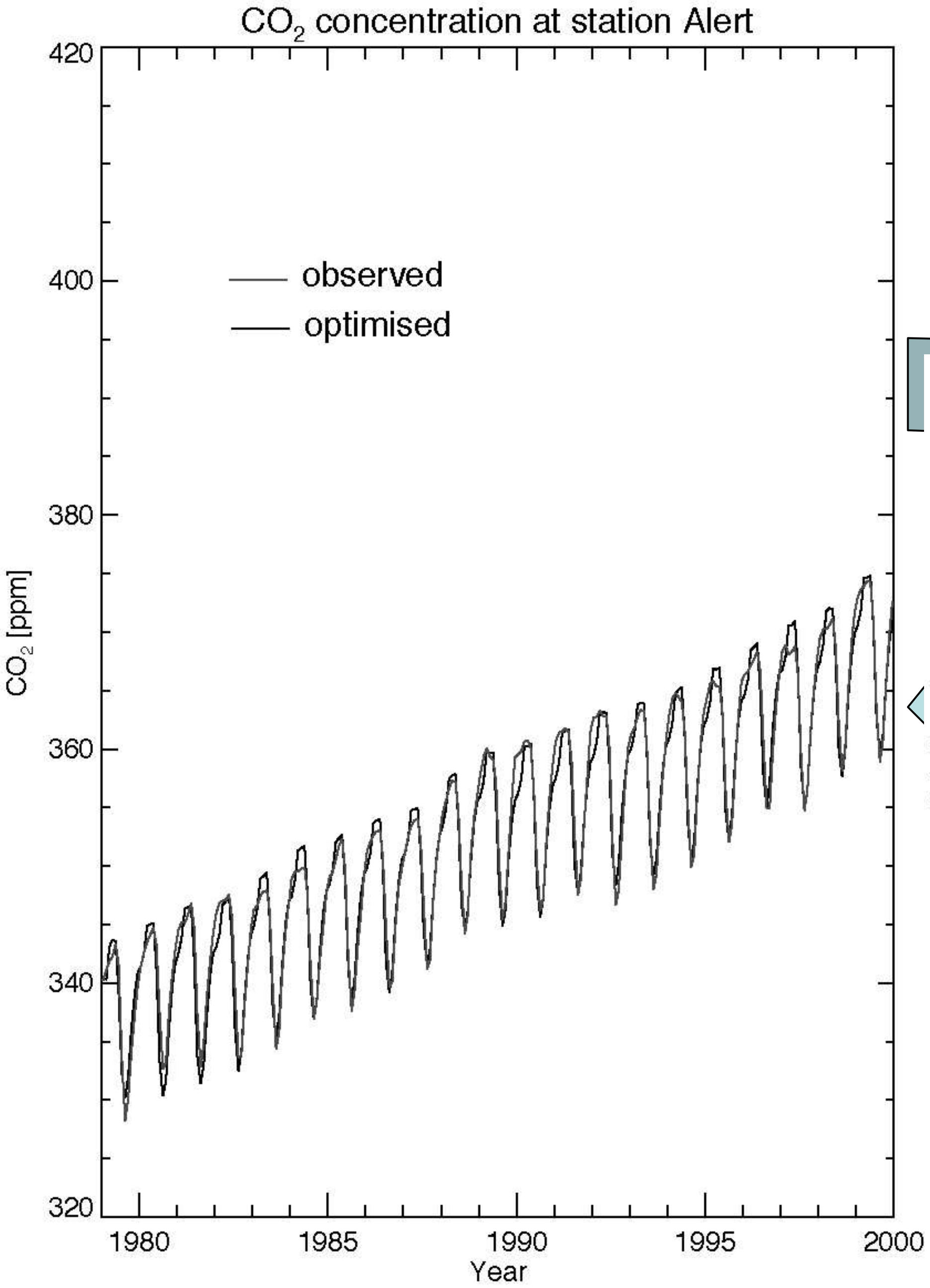
Optimisation

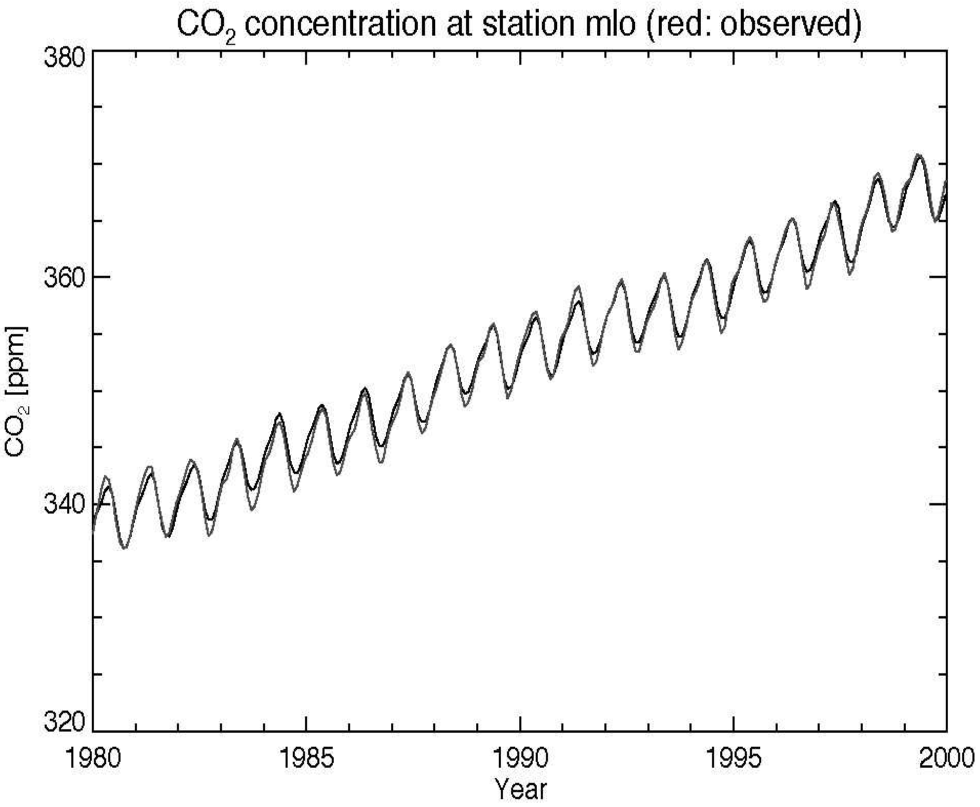


Optimisation

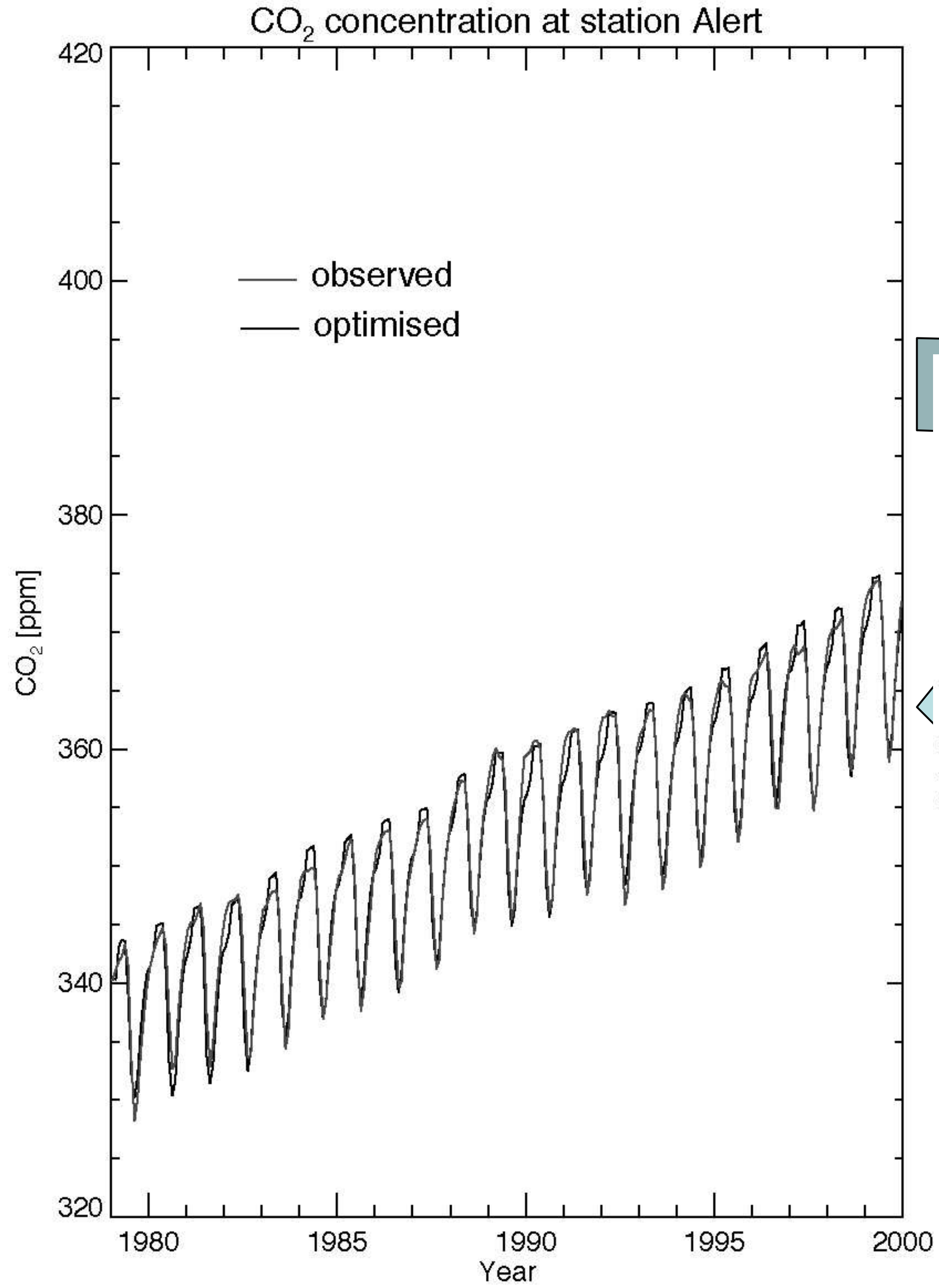


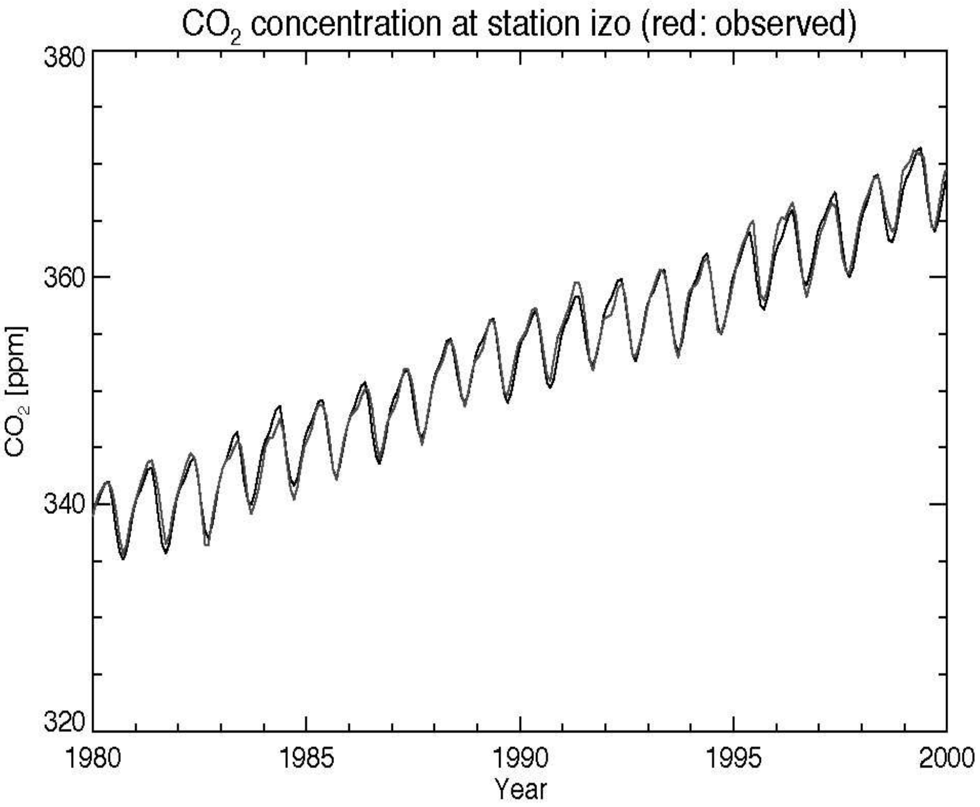
Optimisation



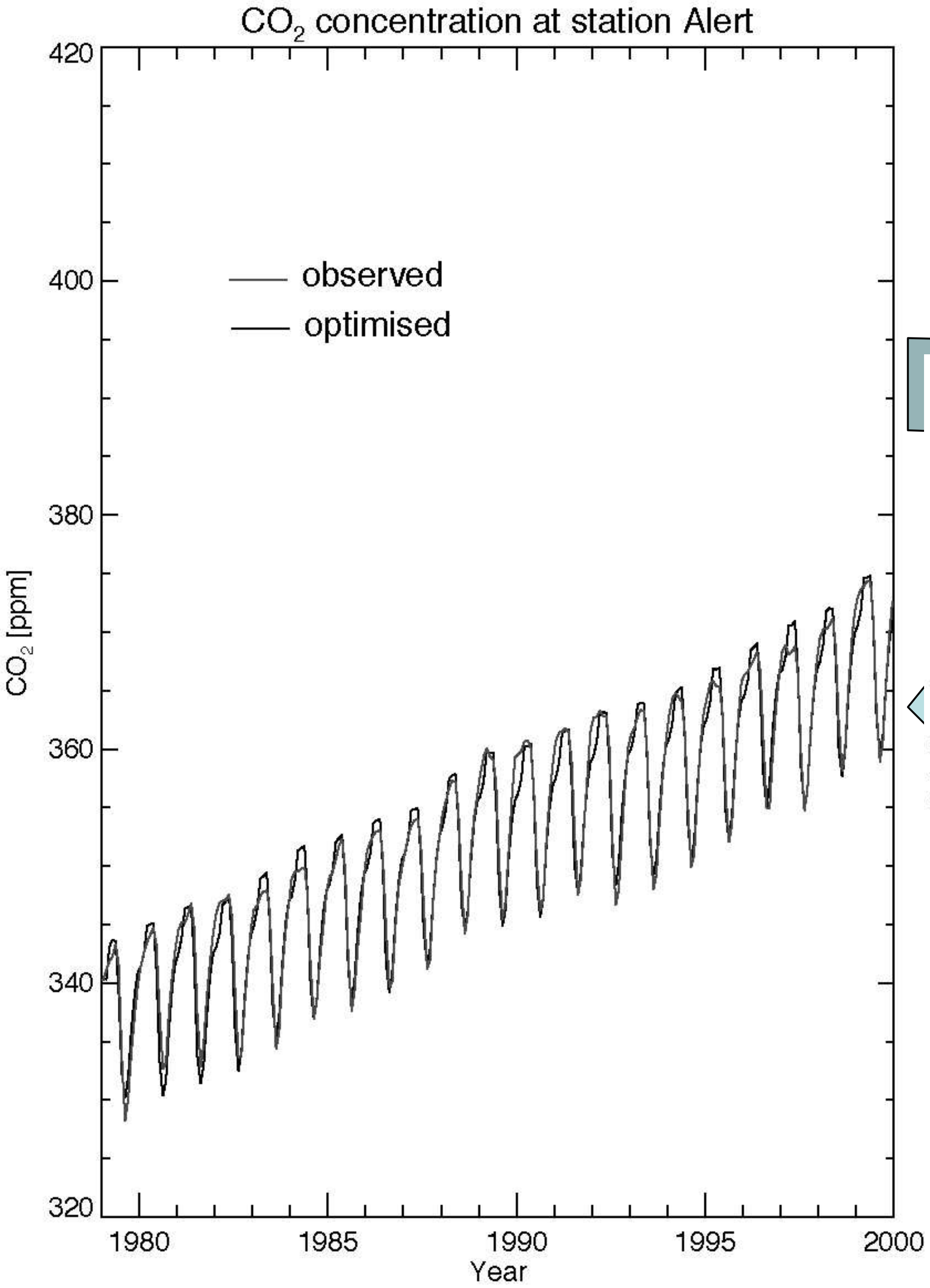


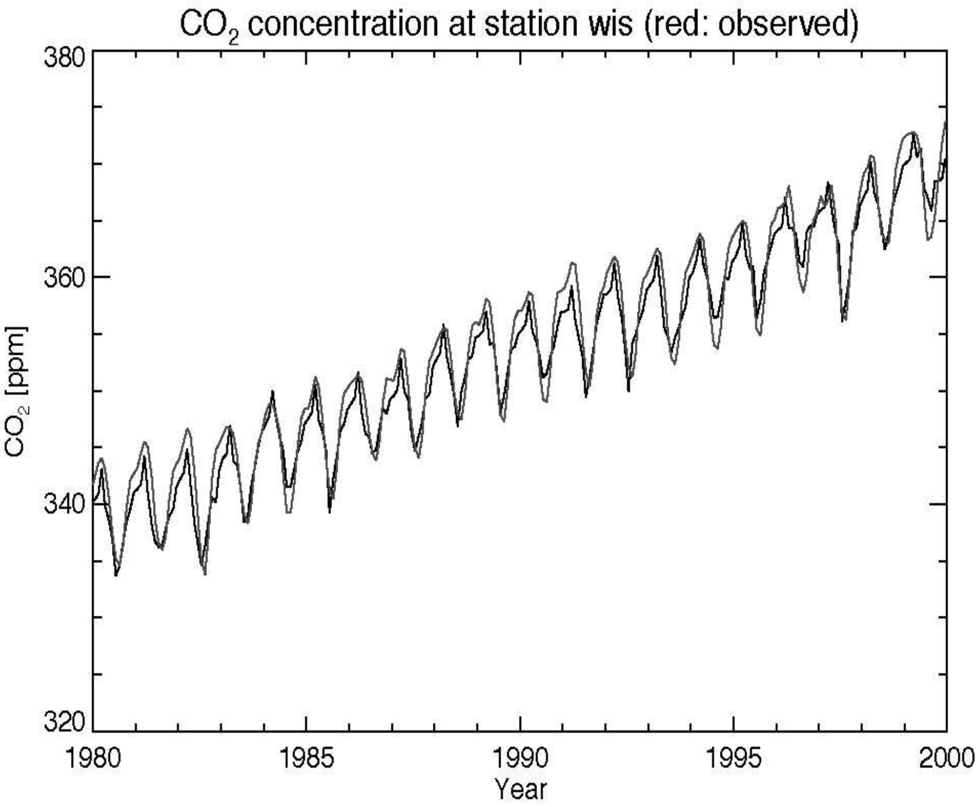
Optimisation



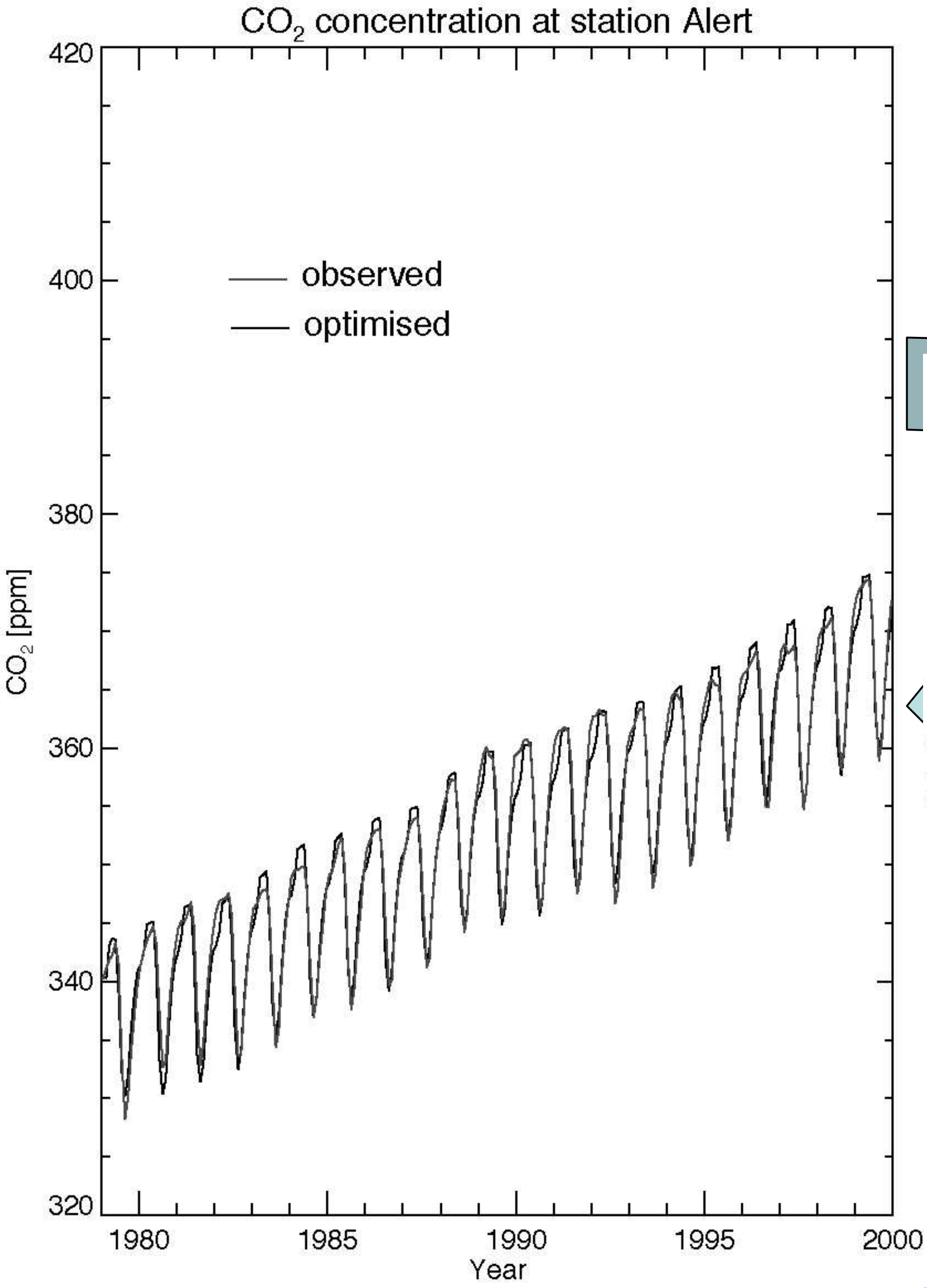


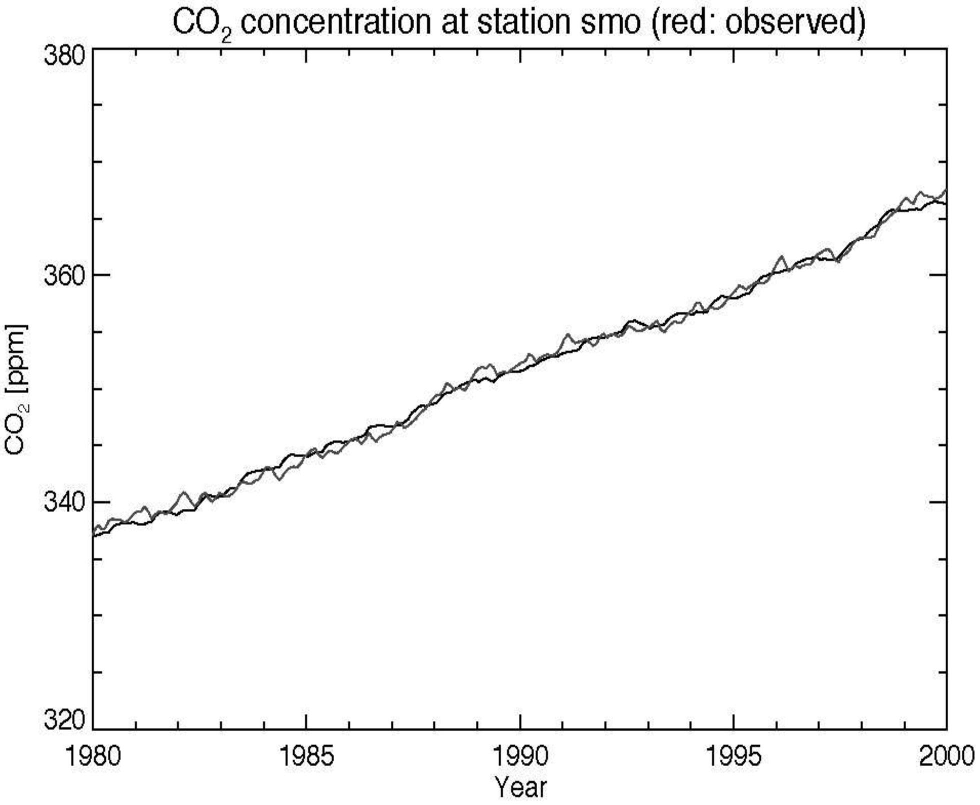
Optimisation


CO$_2$ concentration at station Alert

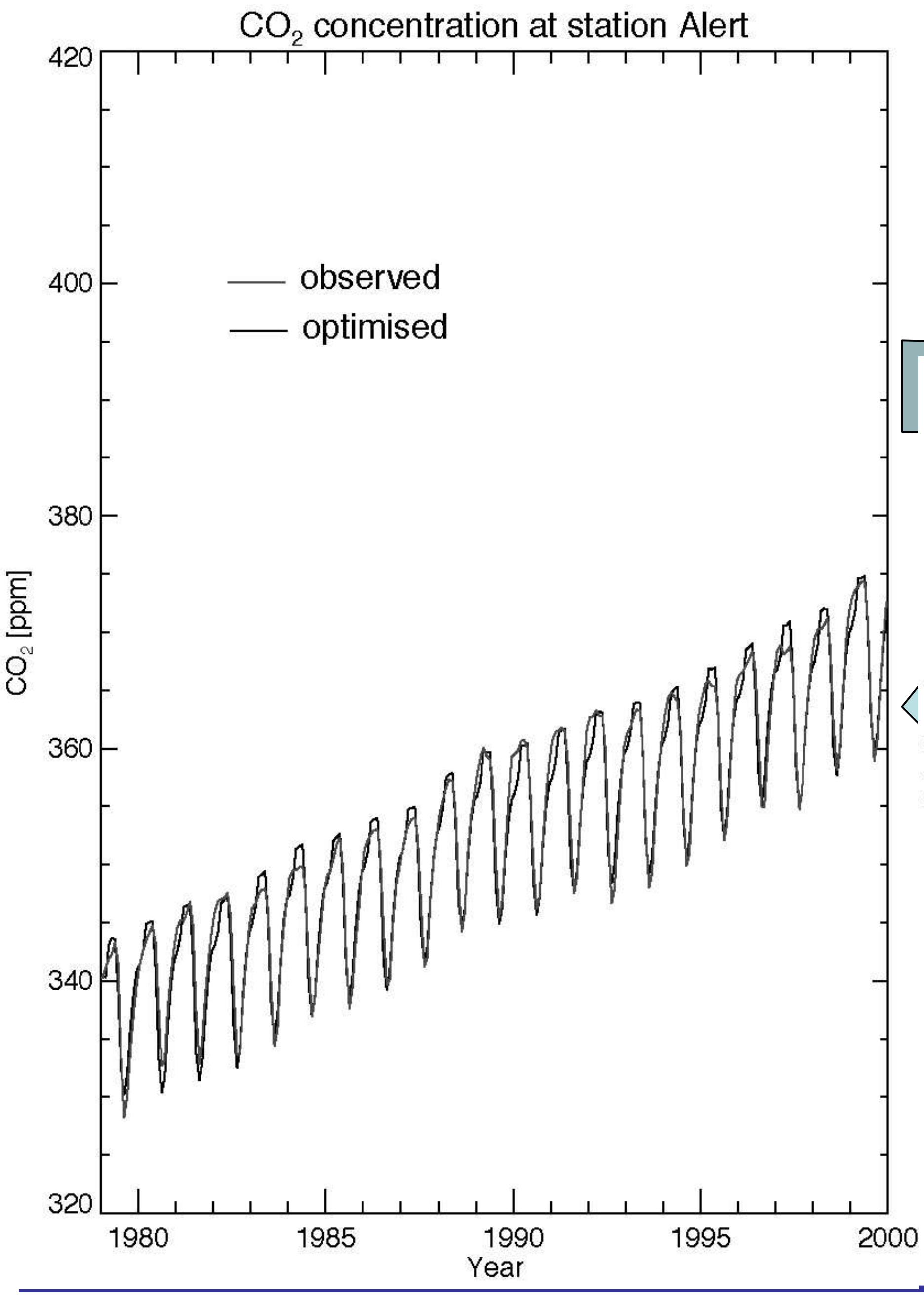

CO$_2$ concentration at station wis (red: observed)

Optimisation

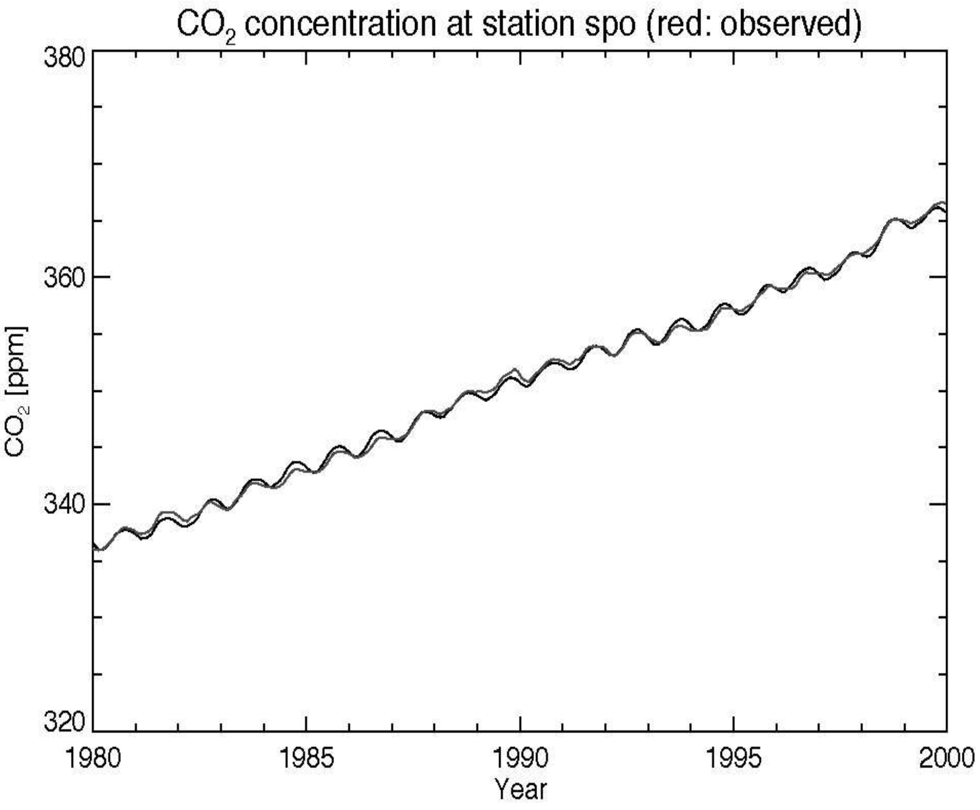




Optimisation



CO_2 concentration at station Alert



CO_2 concentration at station spo (red: observed)

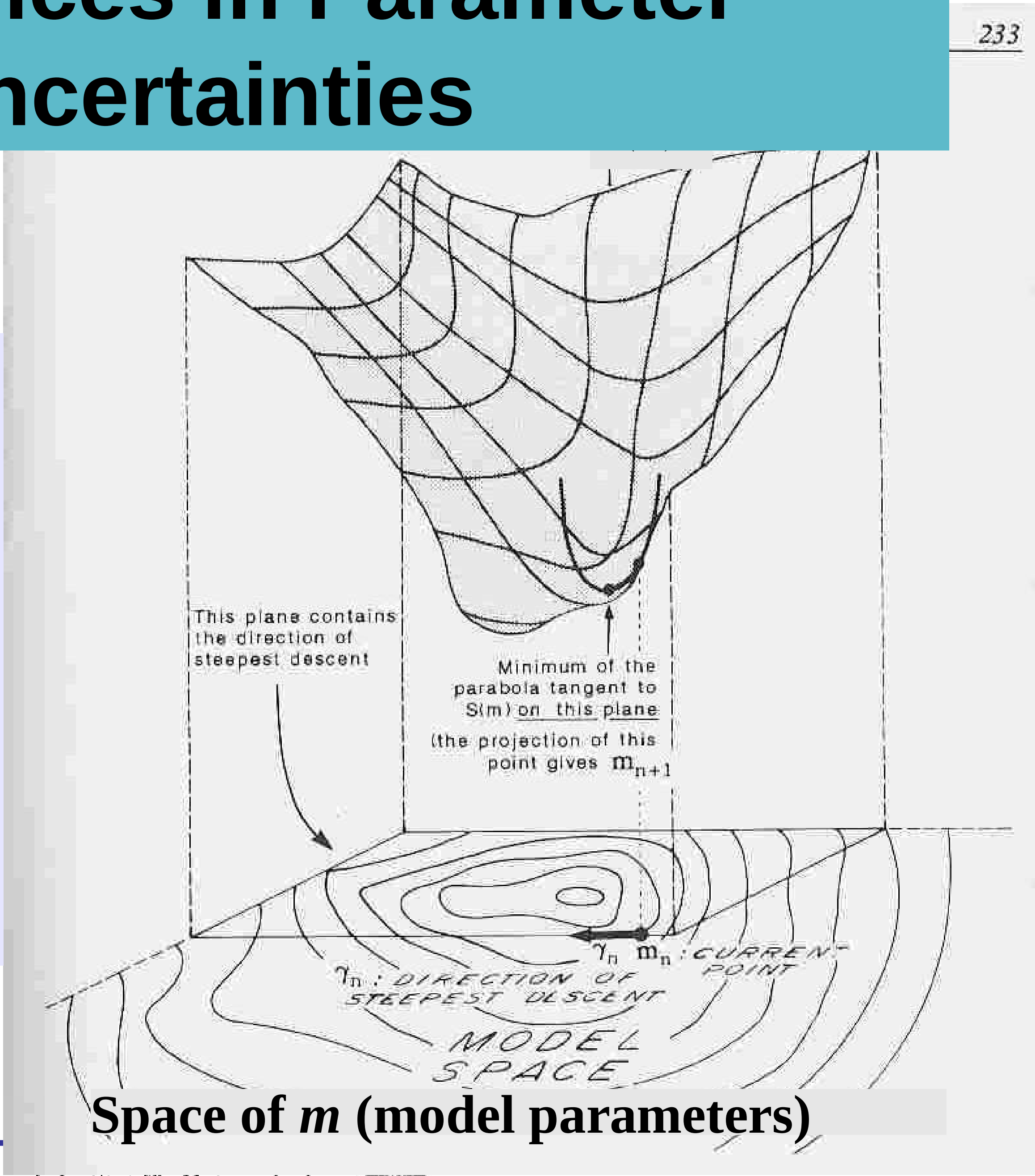
Covariances in Parameter Uncertainties

Second Derivative (Hessian) of *J*(*m*):

$$\partial^2 J(m)/\partial m^2$$

yields curvature of *J*, provides estimated uncertainty in m_{opt}

Figure taken from **Tarantola** '87


233
This plane contains the direction of steepest descent
Minimum of the parabola tangent to S(m) on this plane
(the projection of this point gives m_{n+1}
γ_n m_n : CURRENT POINT
γ_n : DIRECTION OF STEEPEST DESCENT
MODEL SPACE

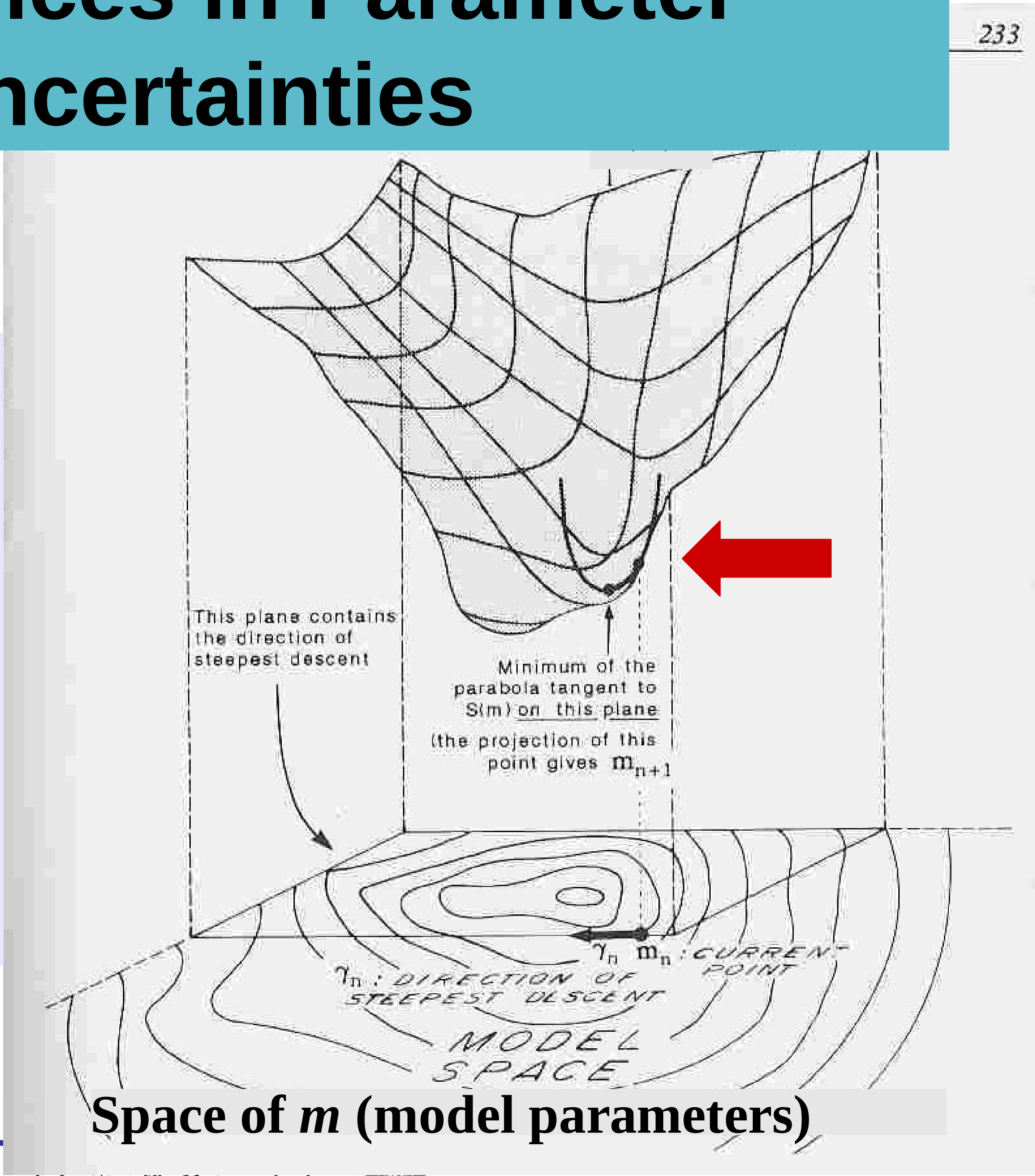
Space of *m* (model parameters)

Covariances in Parameter Uncertainties

Second Derivative (Hessian) of *J*(*m*):

$$\partial^2 J(m)/\partial m^2$$

yields curvature of *J*, provides estimated uncertainty in m_{opt}


233
This plane contains the direction of steepest descent
Minimum of the parabola tangent to S(m) on this plane (the projection of this point gives m_{n+1}
γ_n m_n : CURRENT POINT
γ_n : DIRECTION OF STEEPEST DESCENT
MODEL SPACE

Space of *m* (model parameters)

Figure taken from **Tarantola** '87

Covariances in Parameter Uncertainties

Cost function (misift):

model prediction

measurements

$$J(\vec{m}) = \frac{1}{2}[\vec{m} - \vec{m}_0]\mathbf{C}_{m0}^{-1}[\vec{m} - \vec{m}_0] + \frac{1}{2}[\vec{y}(\vec{m}) - \vec{y}_0]\mathbf{C}_{y}^{-1}[\vec{y}(\vec{m}) - \vec{y}_0]$$

assumed
model parameters

a priori
parameter values

a priori covariance matrix
of parameter uncertainty

covariance of uncertainty in
measurements + model

Covariances in Parameter Uncertainties

Cost function (misift):

model prediction

measurements

$$J(\vec{m}) = \frac{1}{2}[\vec{m} - \vec{m}_0]\mathbf{C}_{m0}^{-1}[\vec{m} - \vec{m}_0] + \frac{1}{2}[\vec{y}(\vec{m}) - \vec{y}_0]\mathbf{C}_{y}^{-1}[\vec{y}(\vec{m}) - \vec{y}_0]$$

assumed
model parameters

a priori
parameter values

a priori covariance matrix
of parameter uncertainty

covariance of uncertainty in
measurements + model

Covar. of parameter uncertainties after optimisation:

$$\mathbf{C}_m = \left\{ \frac{\partial^2 J}{\partial m_{i,j}^2} \right\}^{-1}$$

= inverse Hessian

Covariances in Parameter Uncertainties

Cost function (misift):

model prediction

measurements

$$J(\vec{m}) = \frac{1}{2}[\vec{m} - \vec{m}_0]\mathbf{C}^{-1}_{m0}[\vec{m} - \vec{m}_0] + \frac{1}{2}[\vec{y}(\vec{m}) - \vec{y}_0]\mathbf{C}^{-1}_{y}[\vec{y}(\vec{m}) - \vec{y}_0]$$

assumed
model parameters

a priori
parameter values

a priori covariance matrix
of parameter uncertainty

covariance of uncertainty in
measurements + model

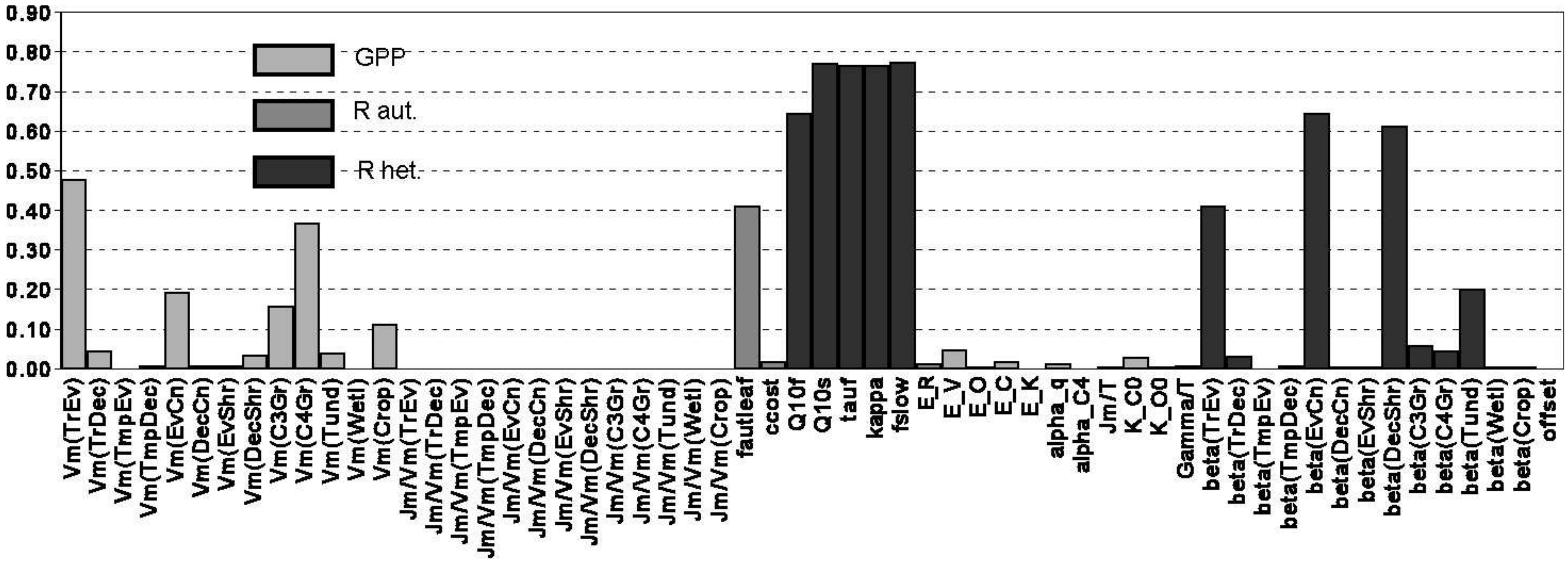
Covar. of parameter uncertainties after optimisation:

$$\mathbf{C}_m = \left\{\frac{\partial^2 J}{\partial m^2_{i,j}}\right\}^{-1}$$ *= inverse Hessian*

examples:

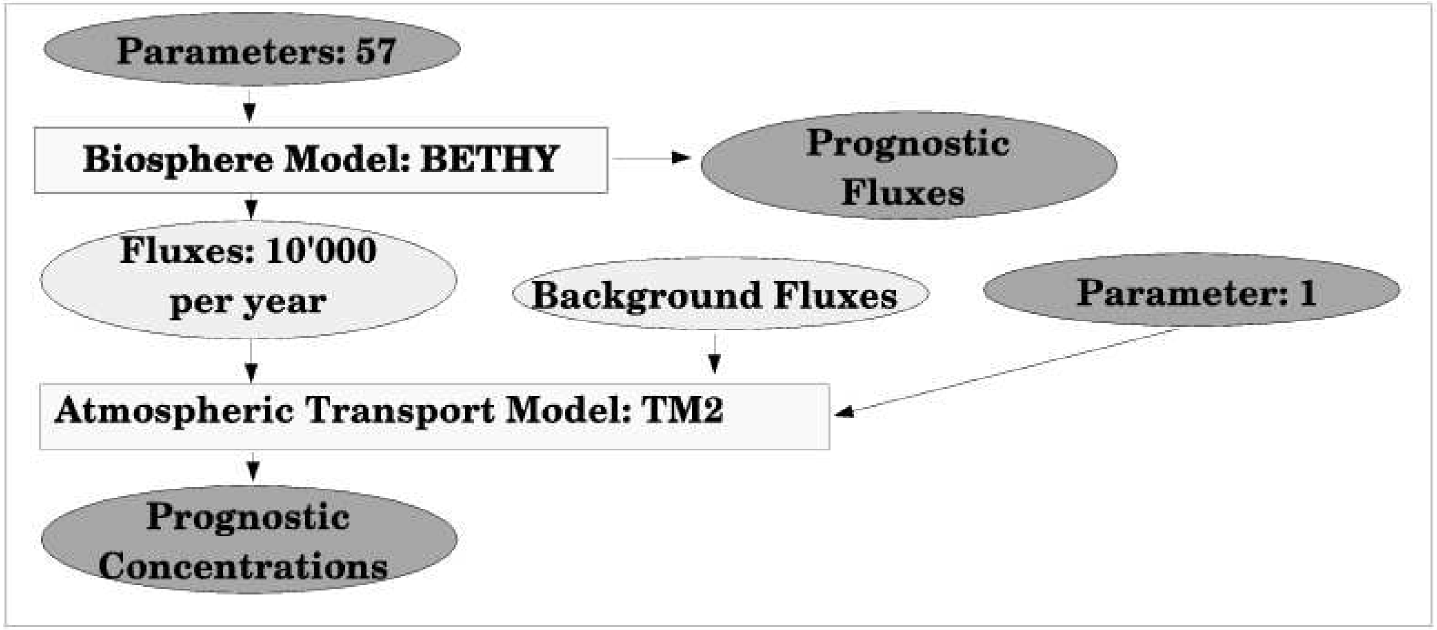
	first guess	optimized	prior unc.	opt.unc.	Vm(TrEv)	Vm(EvCn)	Vm(C3Gr)	Vm(Crop)
	μmol/m^2 s	μmol/m^2 s	%	%	error covariance			
Vm(TrEv)	60.0	43.2	20.0	10.5	**0.28**	0.02	-0.02	0.05
Vm(EvCn)	29.0	32.6	20.0	16.2	0.02	**0.65**	-0.10	0.08
Vm(C3Gr)	42.0	18.0	20.0	16.9	-0.02	-0.10	**0.71**	-0.31
Vm(Crop)	117.0	45.4	20.0	17.8	0.05	0.08	-0.31	**0.80**

Relative reduction of uncertainties



Observations resolve about 10-15 directions in parameter space

Setup for prognostic step



BETHY: Knorr 97; TM2: Heimann 95

Covariances in Uncertainties of Prognostics

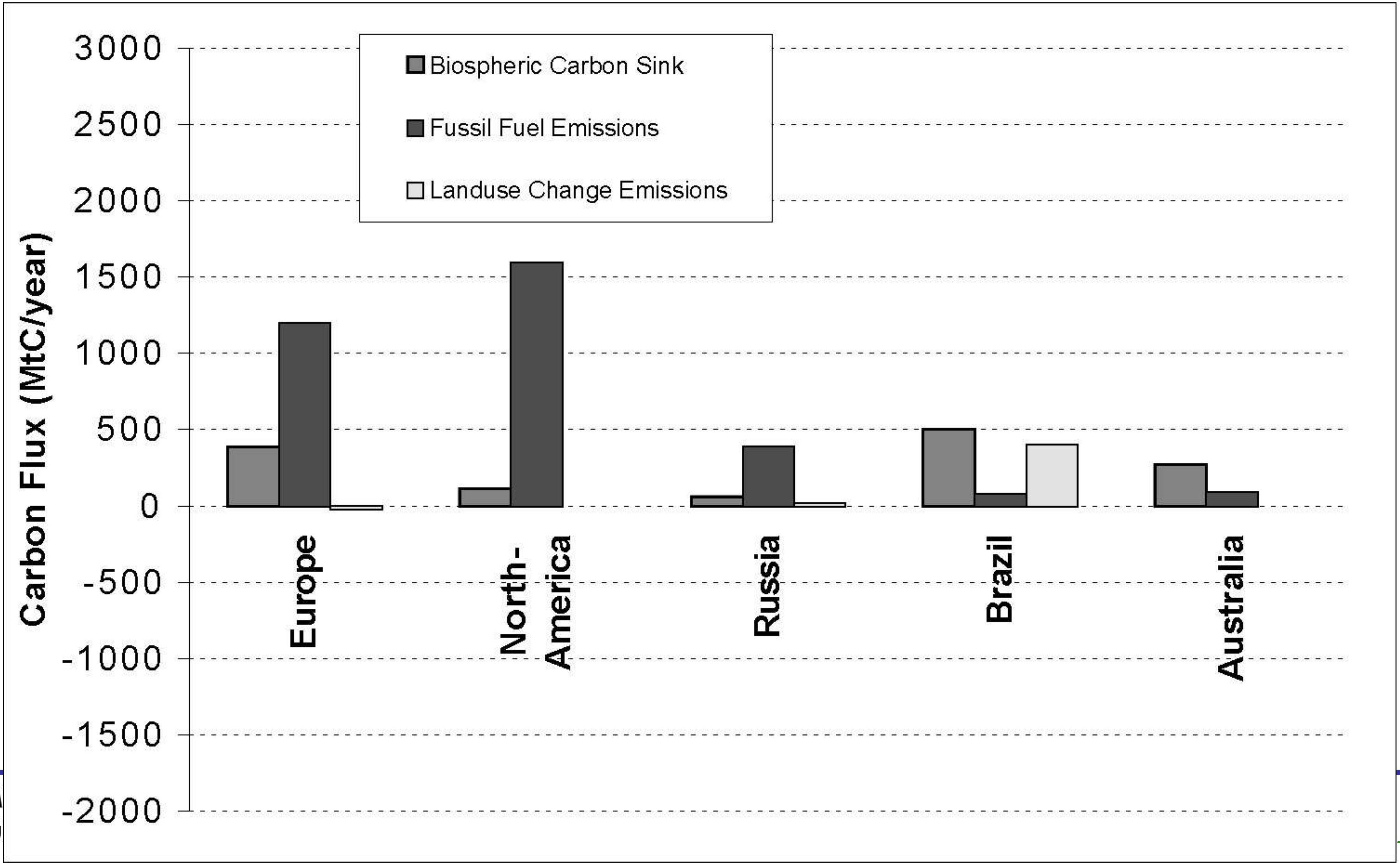
Covariance in uncertainties of prognostics, *y*, after optimisation (e.g. CO$_2$ fluxes):

Jacobian matrix (adjoint or tangent linear model)

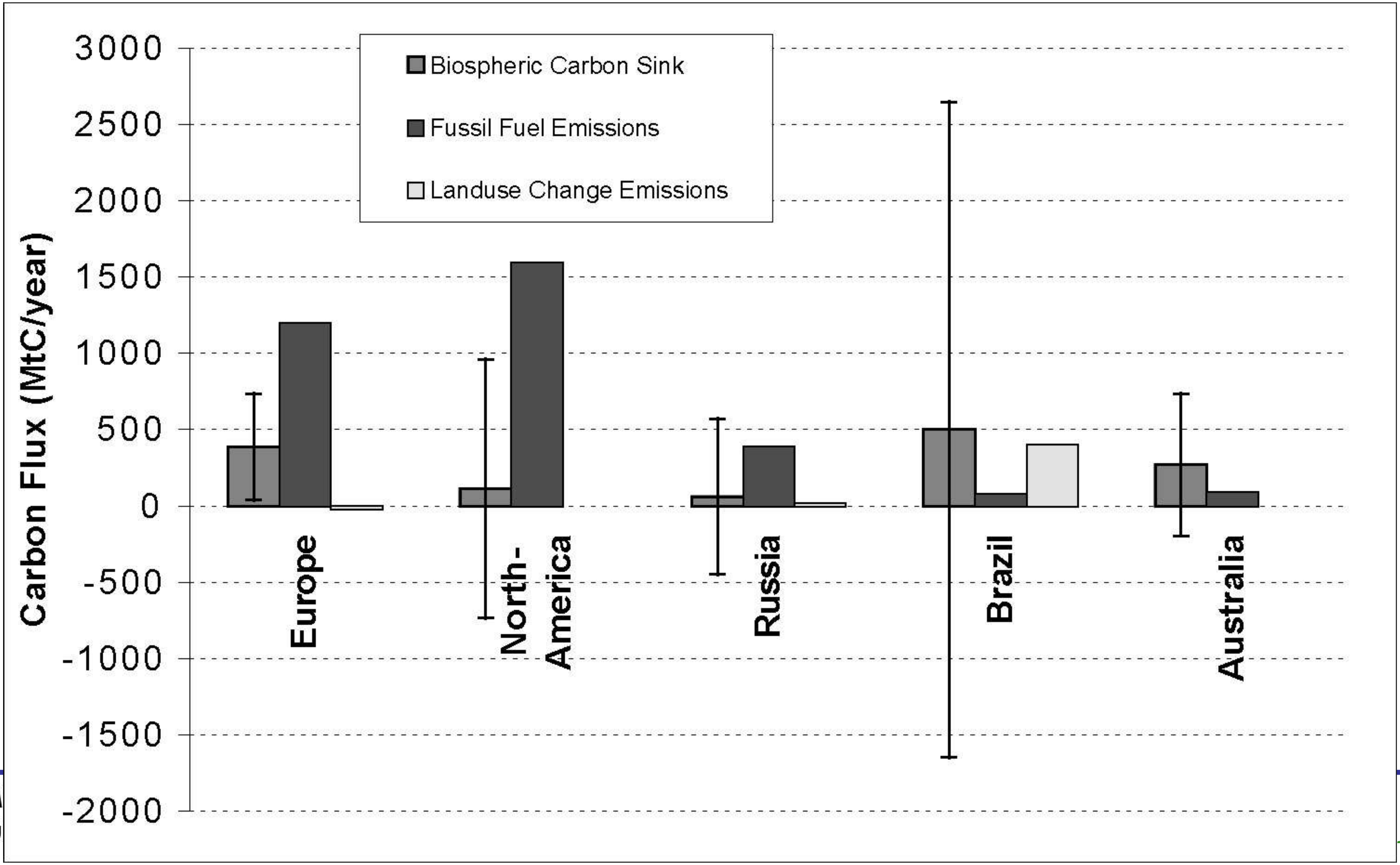
$$\mathbf{C}_y(\vec{m}_{opt}) = \left(\frac{\partial y_i(\vec{m}_{opt})}{\partial m_j}\right) \mathbf{C}_m \left(\frac{\partial y_i(\vec{m}_{opt})}{\partial m_j}\right)^T$$

error covariance of parameters

Regional Net Carbon Balance and Uncertainties



Regional Net Carbon Balance and Uncertainties



Model development within System

- System can test a given combination observational data + model formulation with uncertain parameters, and deliver optimal parameters, prognostics, and their a posteriori uncertainties
- Model is developed further within system
- Model development benefits from sensitivity information and comparison with data (often brutal!)
- Work is ongoing, numbers are from model formulation we are not yet happy with...

Automatic Differentiation

- Uses adjoint, tangent linear and Hessian code
- All this derivative code generated from F90 source code of model (~5500 lines)
  by automatic differentiation tool TAF
- CPU time in multiples of model
  (on Linux: 2 XEON 2GHz):
  tangent linear: 2.1
  adjoint: 3.4
  Hessian * 12 vectors: 50

Summary

- Concept can be generalised to other modeling systems, e.g.
  -> Ocean: MIT model, presentation of Heimbach et al.
  -> NWP: DAO fvGCM, poster Giering et al.
- Automatic differentiation helps to reduce the delay from model development to data assimilation