An example of an automatic differentiation based modelling system

Thomas Kaminski[1], *Ralf Giering*[1],
Marko Scholze[2], *Peter Rayner*[3], *Wolfgang Knorr*[4]

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

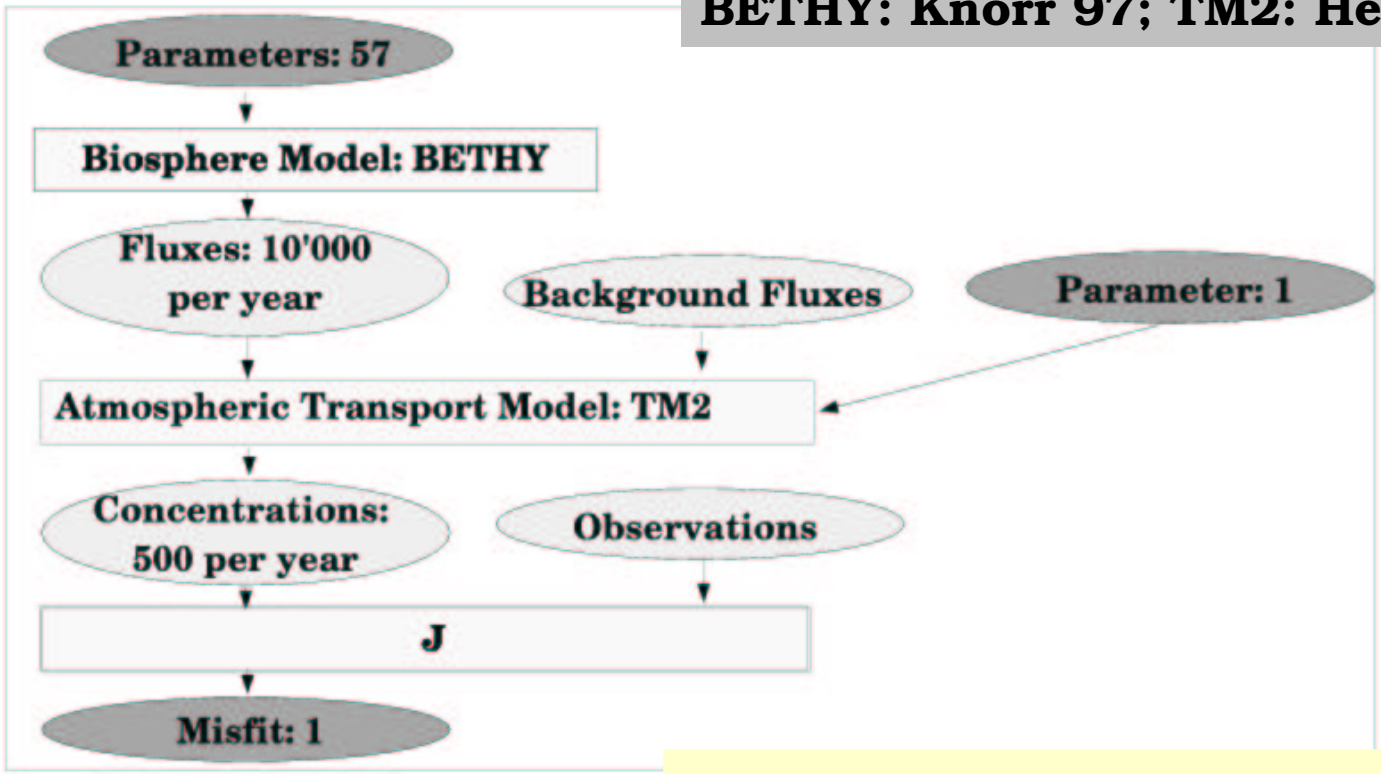
Max-Planck-Institut für Biogeochemie 2 Max-Planck-Institut für Meteorologie Max Planck Institute for Meteorology 3 CSIRO 1 FastOpt CAMELS

Overview

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

Setup for Calibration Step

BETHY: Knorr 97; TM2: Heimann 95



$$J(m) = \tfrac{1}{2}\,[(m-m_0)C_m^{-1}(m-m_0)+(M(m)-d)C_d^{-1}(M(m)-d)]$$

Gradient Method

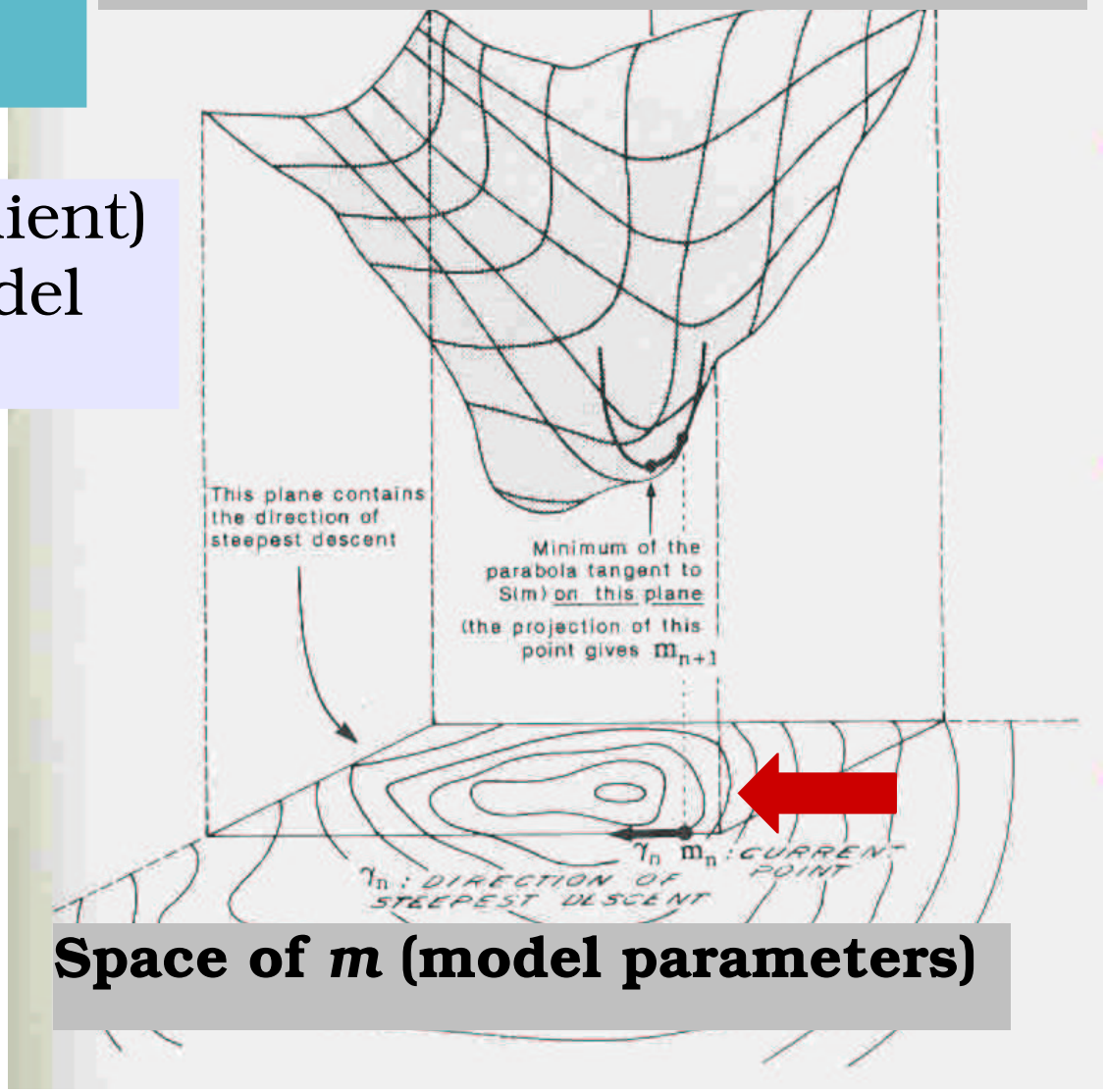
First derivative (Gradient) of $J(m)$ w.r.t. m (model parameters) :

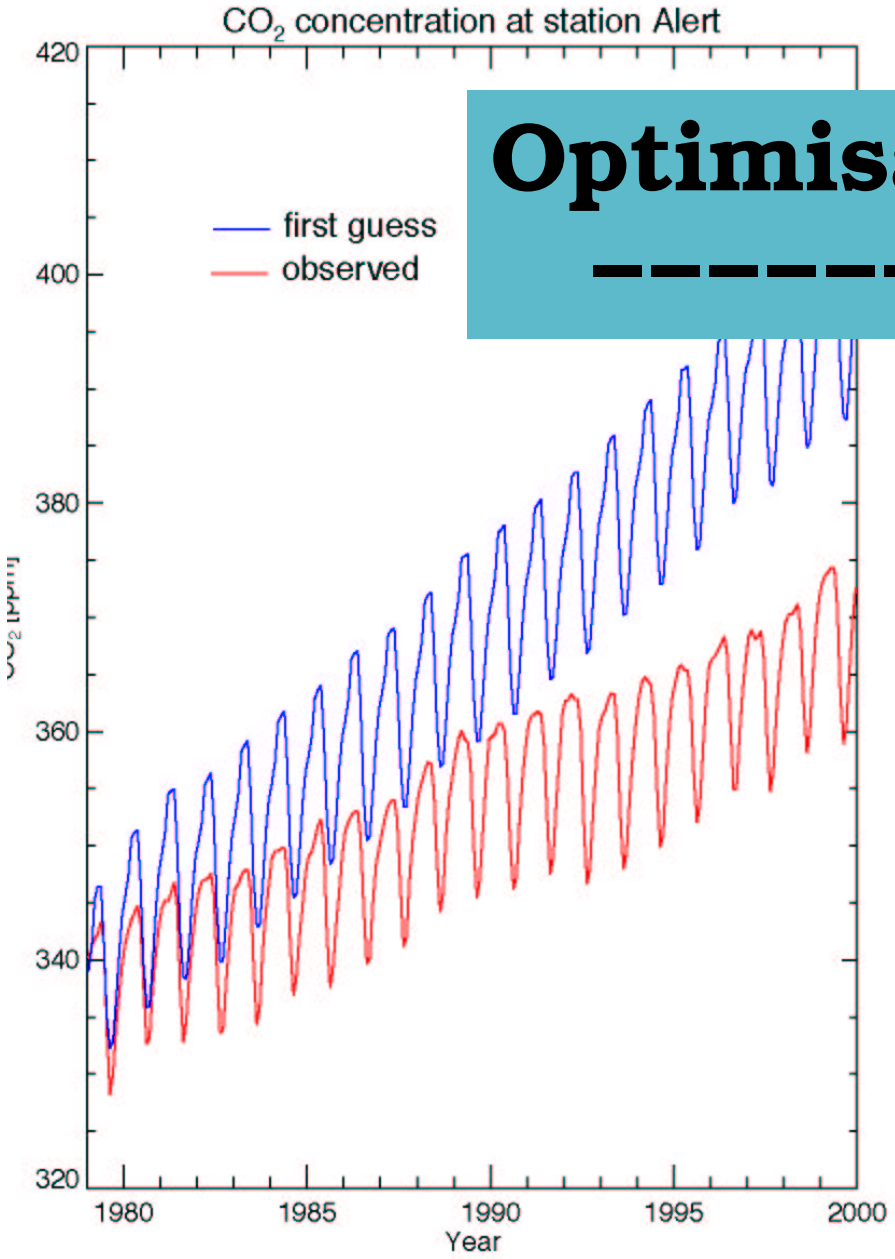
$$-\partial J(m)/\partial m$$

yields direction of steepest descent

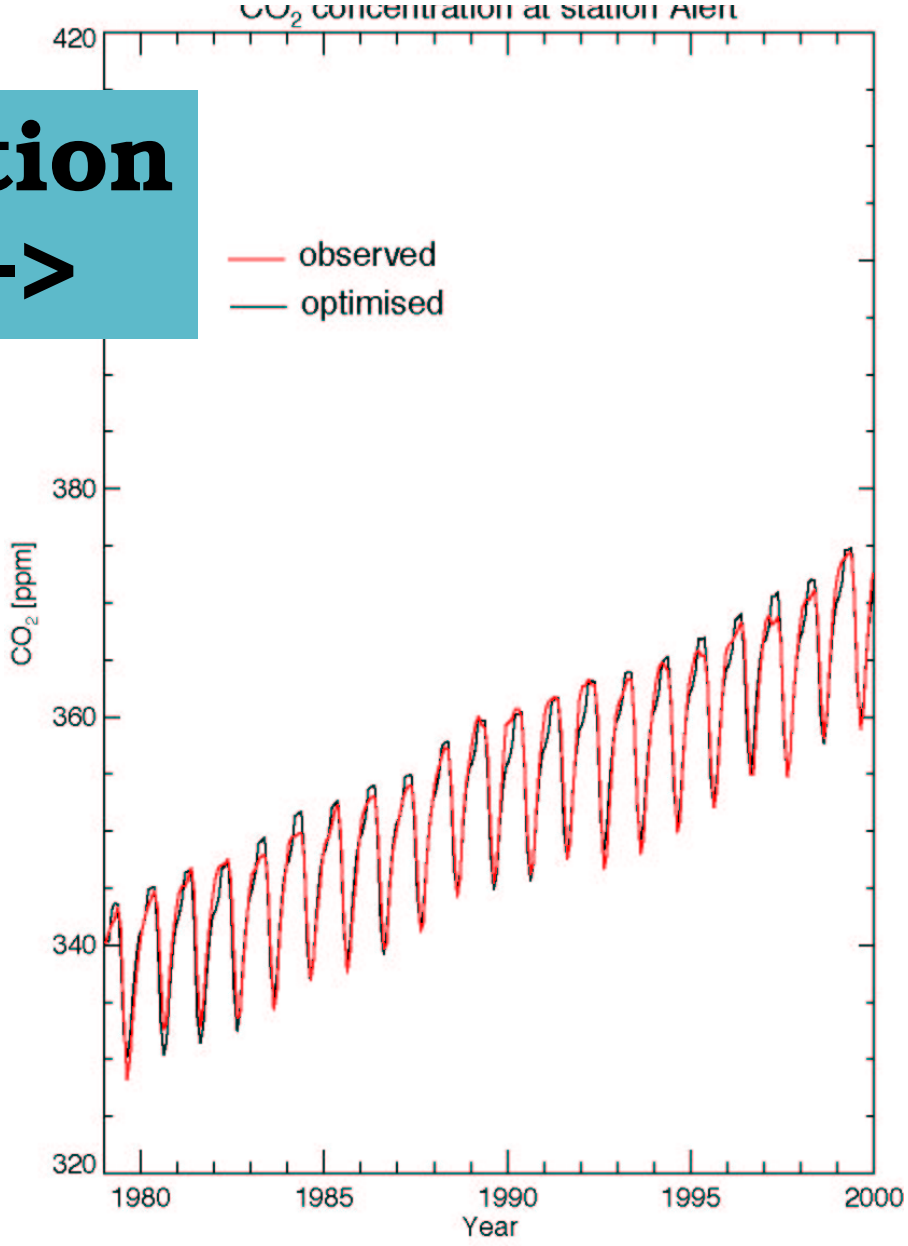
Figure taken from **Tarantola** '87

Cost Funtion J(m)





Optimisation ------->



Max-Planck-Institut für Biogeochemie

Max-Planck-Institut für Meteorologie
Max Planck Institute for Meteorology

CSIRO

FastOpt

CAMELS

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}

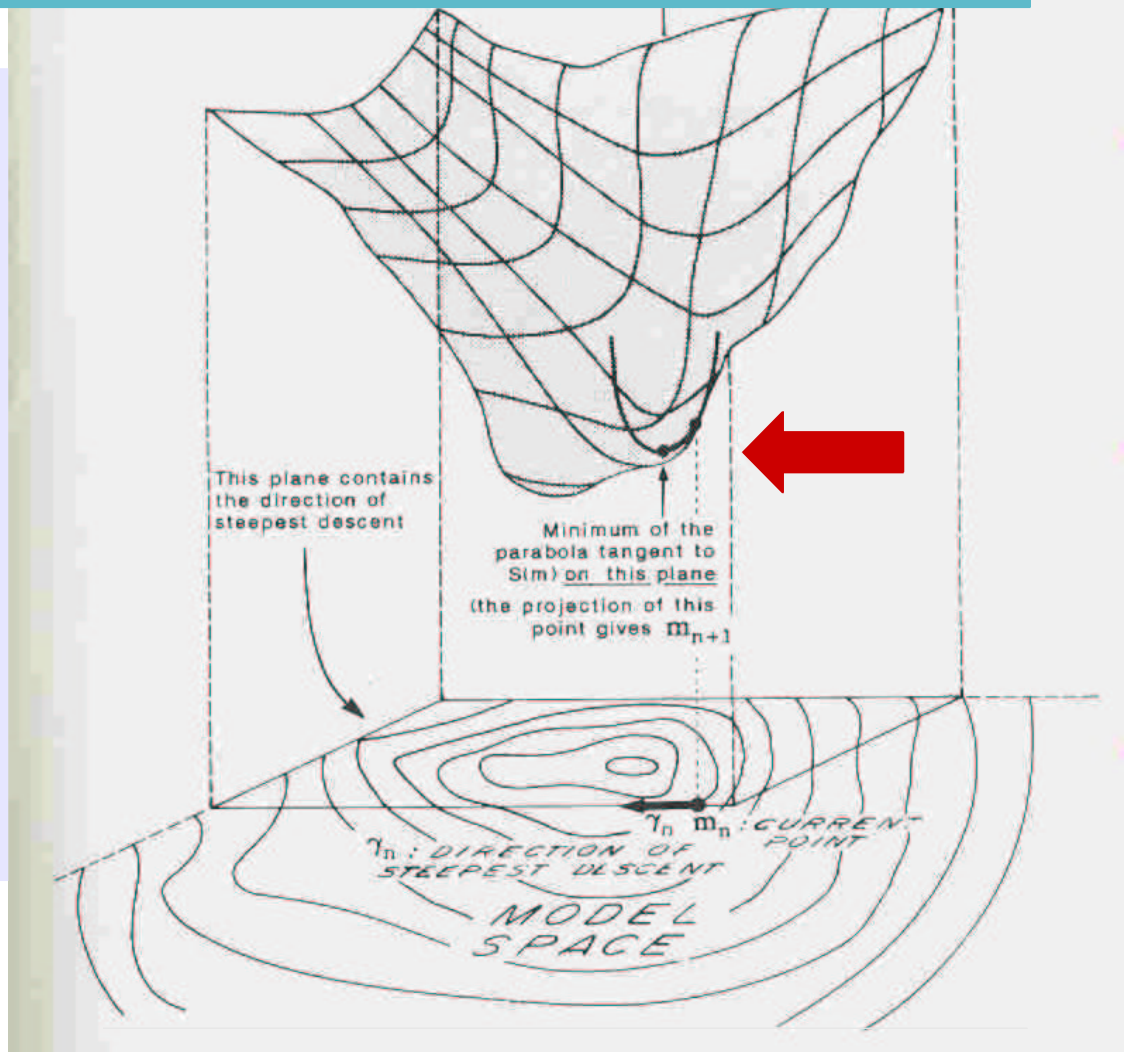


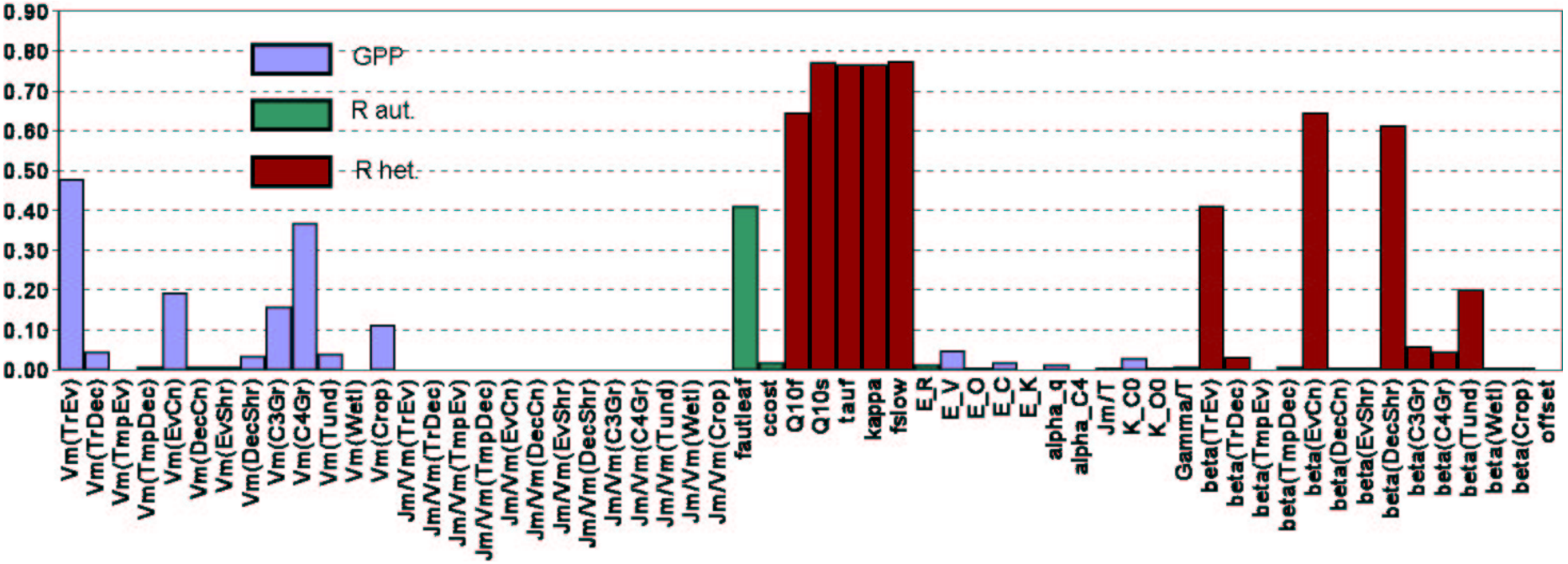
Figure taken from **Tarantola** '87

Covariances in Parameter Uncertainties

examples:

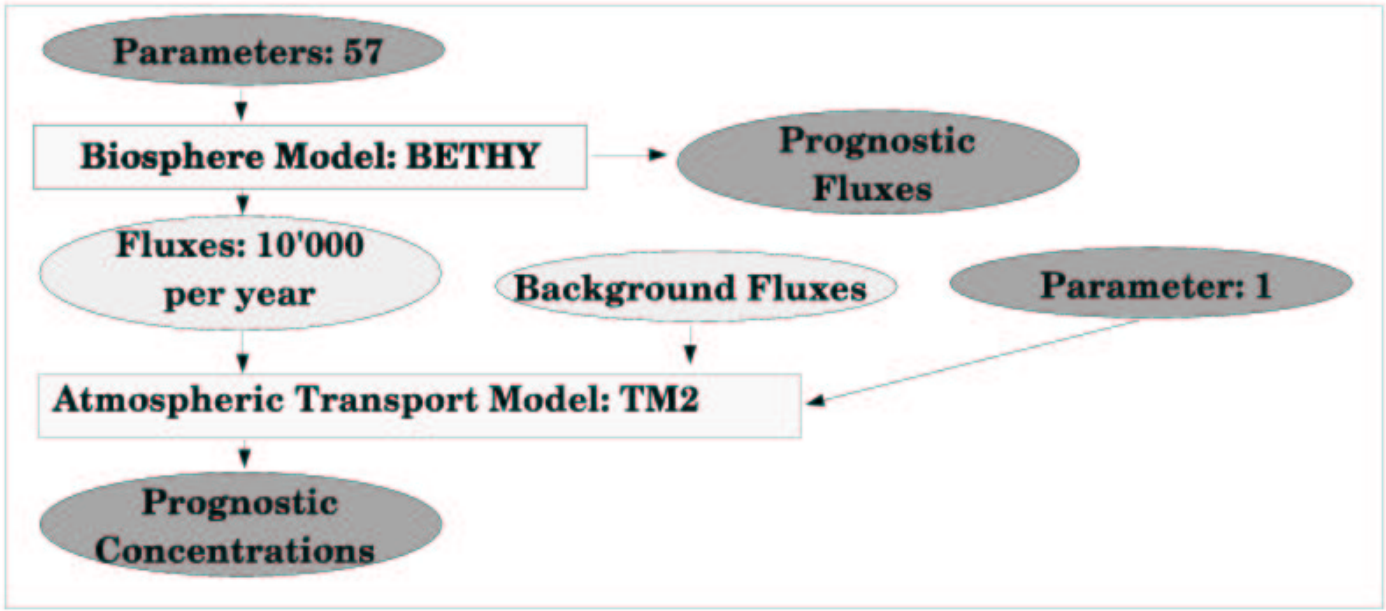
	first guess	optimized	prior unc.	opt.unc.	Vm(TrEv)	Vm(EvCn)	Vm(C3Gr)	Vm(Crop)
	µmol/m ^{2}s	µmol/m ^{2}s	%	%	error covariance			
Vm(TrEv)	60.0	43.2	20.0	10.5	**0.28**	0.02	−0.02	0.05
Vm(EvCn)	29.0	32.6	20.0	16.2	0.02	**0.65**	−0.10	0.08
Vm(C3Gr)	42.0	18.0	20.0	16.9	−0.02	−0.10	**0.71**	−0.31
Vm(Crop)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

Prognostics, e.g. CO_2 fluxes, are a function of model parameters

$$y = M(\mathbf{m}_{opt})$$

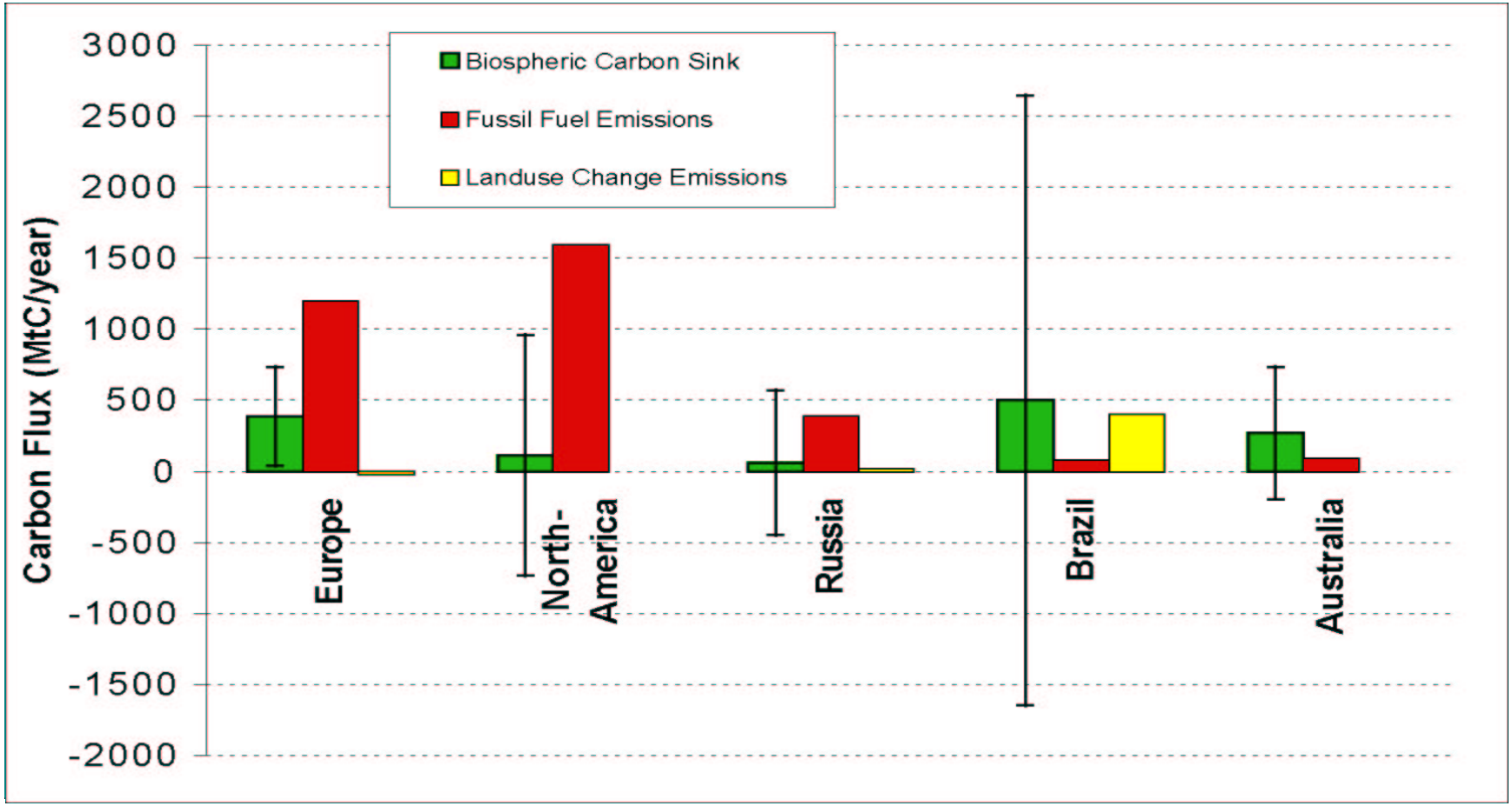
Posterior (after optimisation) covariance in uncertainties of prognostics

Jacobian matrix
(adjoint or tangent linear model)

$$C_{y,post} = (DM) C_{m,post} (DM)^T$$

error covariance
of parameters

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
- Initial and boundary conditions can be included as unknowns/control variables (we have 1 already)
- 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 derivative code generated from Fortran–90 source code of model (~5500 lines) by AD tool TAF
- Process fully automatic: no hand coding
- Good performance on Linux, XEON 2GHz, with Lahey lf95: –tpp –O

some larger TAF Derivatives

Model (Who)	Lines	Lang	TLM	ADM	Ckp	HES
NASA/NCAR (w. Todling & Lin)	87'000	F90	2.7	6.8	2 lev	–
MOM3 (Galanti & Tziperman)	50'000	F77	Yes	4.6	2 lev	–
MITGCM (ECCO Consortium)	100'000	F77	1.8	5.5	3 lev	11.0/1
BETHY (w. Knorr, Rayner, Scholze)	5'500	F90	1.5	3.4	2 lev	50/12
Nav.–Stokes–Solver (Hinze, Slawig)	450	F77	–	2.0	steady	–
NSC2KE	2'500	F77	2.4	3.4	steady	9.8/1
HB_AIRFOIL (Thomas & Hall)	8'000	F90	–	3.0		–

- **Lines: total number of Fortran lines without comments**
- **Numbers for TLM and ADM give CPU time for (function + gradient) relative to forward model**
- **HES format: CPU time for Hessian * n vectors rel. t. forw. model/ n**
- **2 (3) level checkpointing costs 1 (2) additional model run(s)**

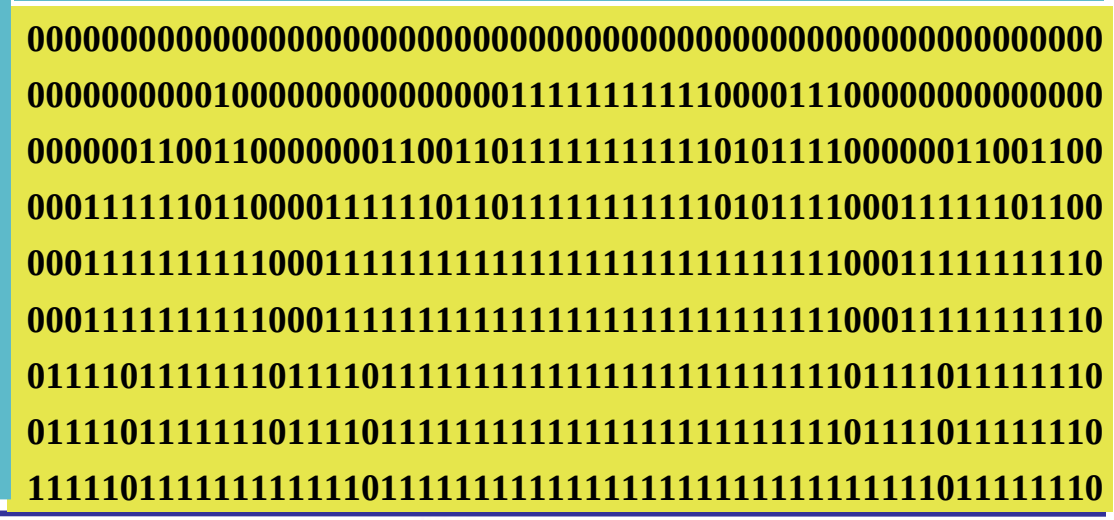
Automatic Sparsity Detection (ASD) in TAF

TAF generates code to evaluate the sparsity pattern of a Jacobian (ASD)
ASD–code propagates bit vectors –> is very efficient
Can do forward mode and reverse mode

Example: DM for M: Paramters –> Net CO2 Flux in latitude bands

Parameters –––>

latitude – –>

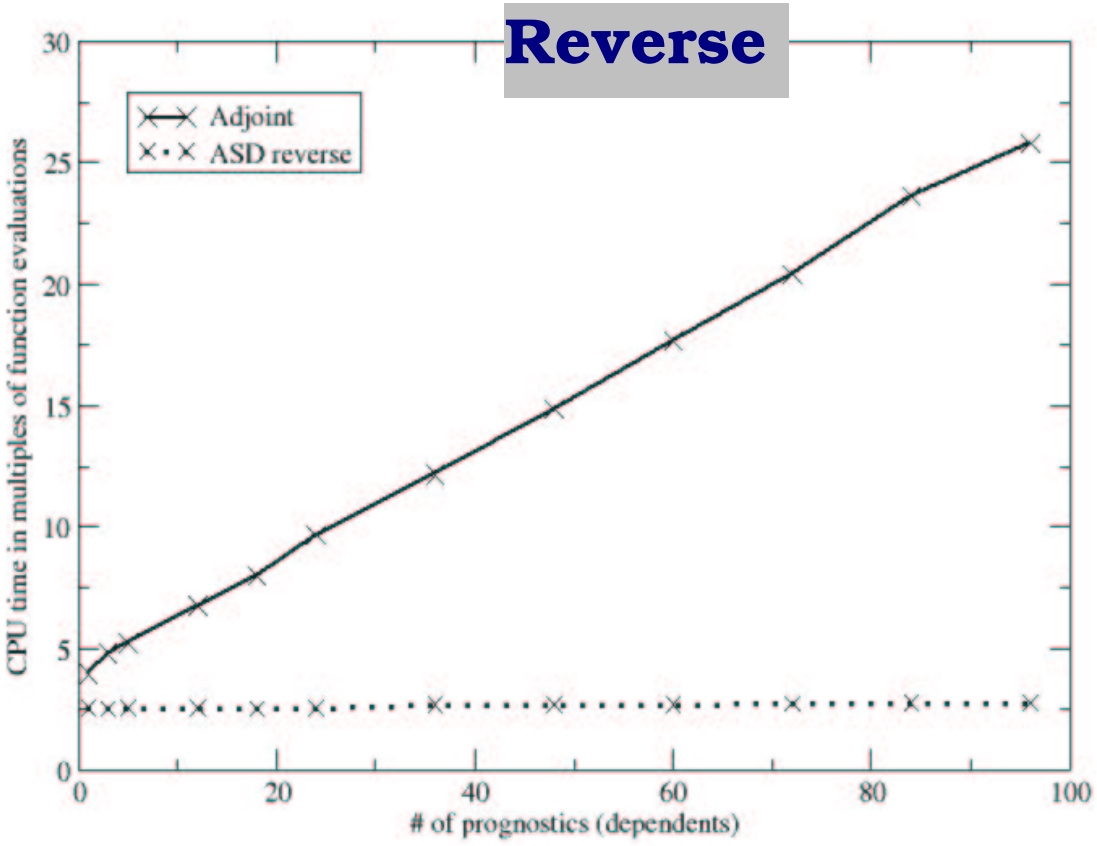


Jacobian and ASD Performance

Forward

58 independents
Jacobian: 12
ASD: 1.3

Reverse



CPU time quantified in multiples of function evaluations

Summary

- **AD is key technology in CCDAS: Allows propagation of uncertainties**
- **Concept can be generalised to other modelling systems**
- **Automatic differentiation essential to reduce the delay from model development to data assimilation**