International GODAE Summer School

La Londe-Les-Maures, 20.9.-1.10.2004

On the Use of Data Assimilation in Biogeochemical Modelling

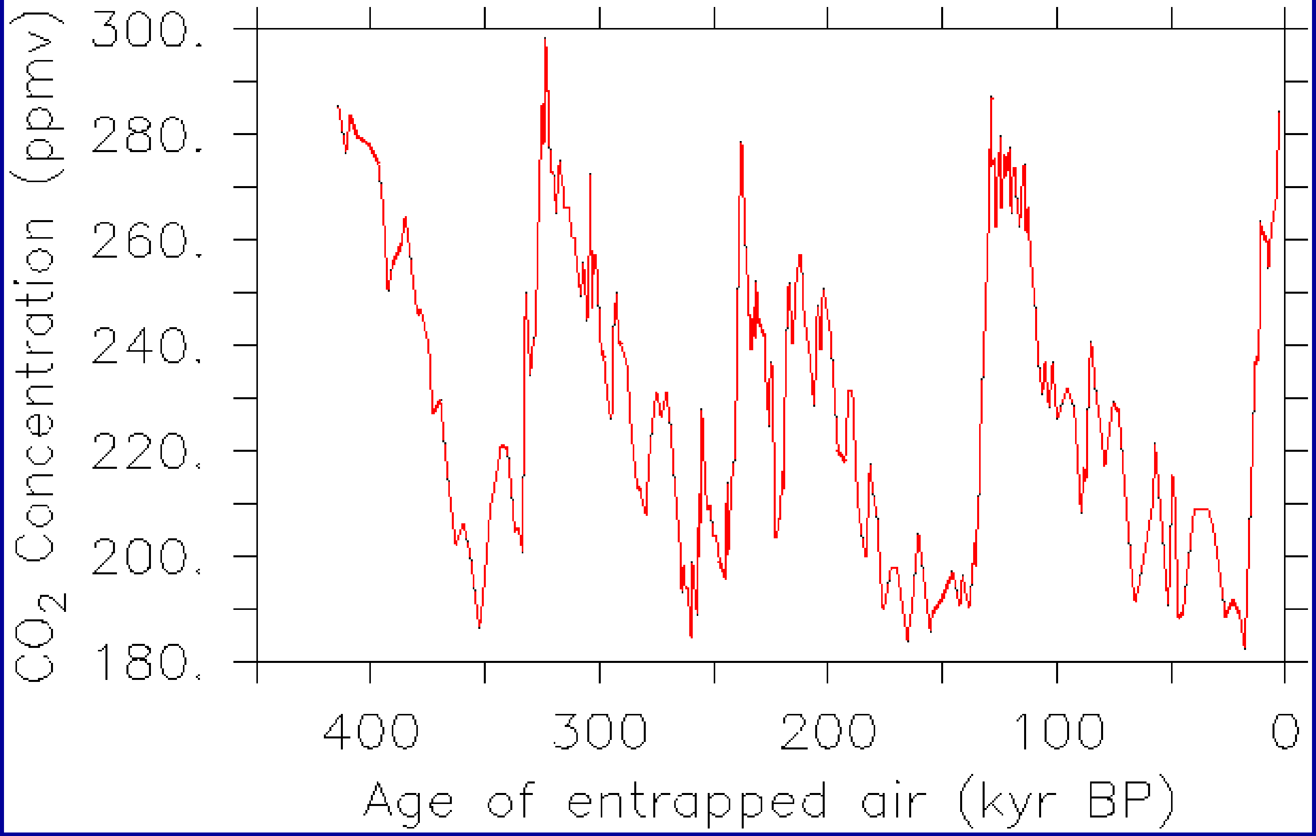
Andreas Oschlies

School of Ocean & Earth Science,

Southampton Oceanography Centre, UK

Why care about marine biogeochemical cycles?

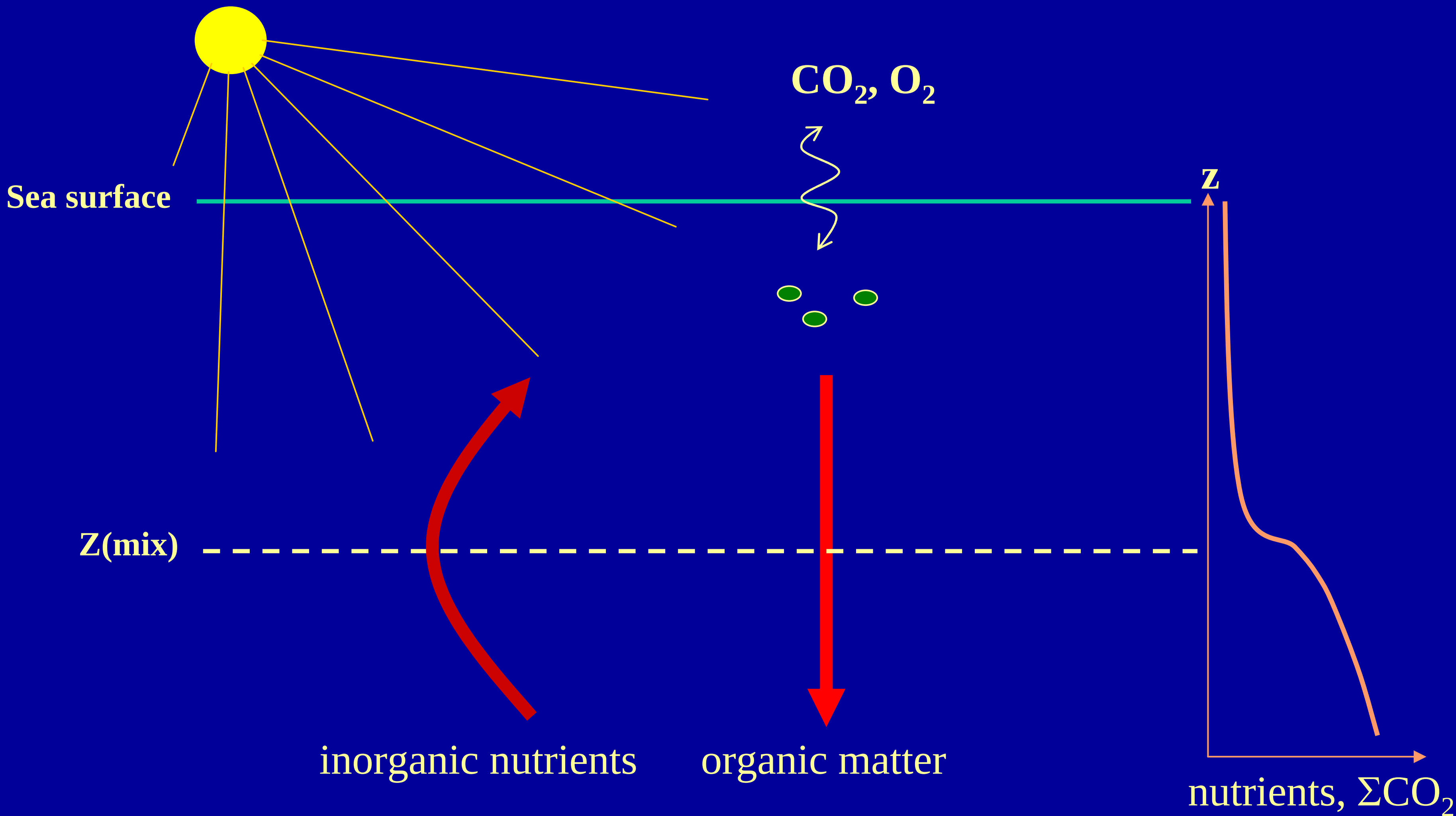
Vostok, Antarctica Ice Core Atmospheric CO_2 Record



(Barnola et al., 1999)

(It's not only fish!)

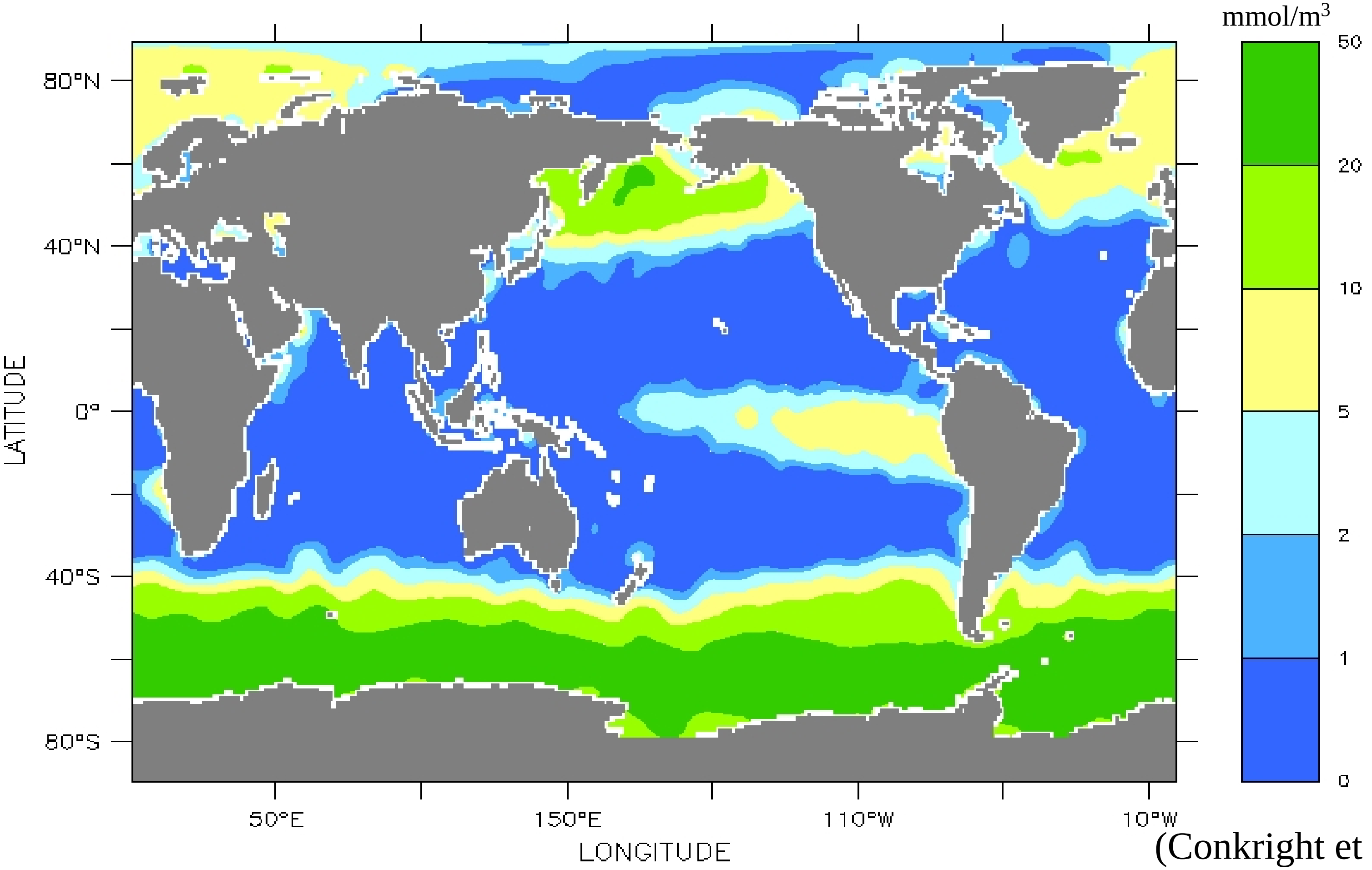
The Biological Pump



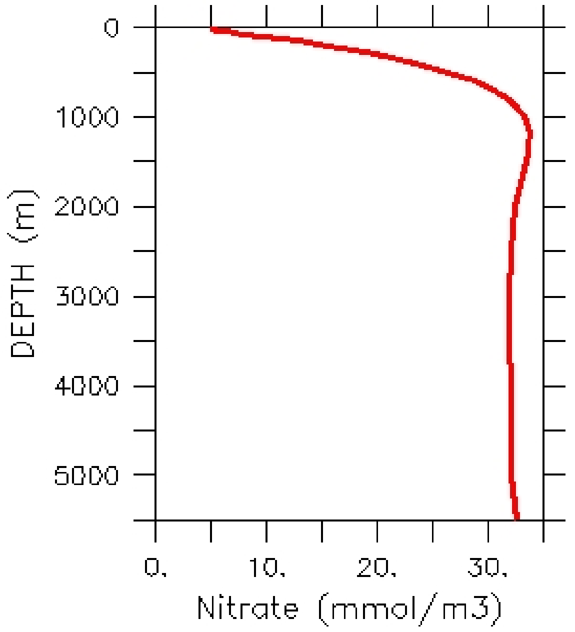
``relatively constant´´ C:N:P:-O_2

``Potential´´ of today´s biological pump

Present-day sea-surface nitrate concentrations



Mean profile



(Conkright et al., 1994)

Controls are not well understood

Why care about data assimilation?

Ideal world:

- Have perfect model that correctly reproduces perfect data.
- Model generates a fully consistent 4D picture of the real ocean/atmosphere.

Real world: Have to cope with

- Imperfect models (in particular **systematic** deficiencies!)
- Imperfect data (measurement errors, methodological uncertainties, sampling problems,...)
- Often poor data coverage (e.g., mainly surface data (satellites!), few winter data, more data of production than of remineralisation,...)

Why care about data assimilation?

Real world is a particularly difficult subject for biogeochemical modellers:

- Apart from mass conservation, theoretical fundations are weak.
  - no bgc analog to the Navier-Stokes equations
  - many (most?) species + their function probably still unknown
  - limited lab/culture studies (“zoo“ species)
- Large number of data that are difficult to interpret, few data that are easy to interpret.
  - ocean colour data
  - “historical“ measurement protocols (e.g. 14C incubation)
  - mostly stock measurements, few rate measurements

Why care about data assimilation?

State estimation

- Improves hindcast or forecast.
- In bgc modelling only of value for short-term forecasts (memory of initial conditions much shorter than annual cycle)
- Usually assumes zero model bias.

Parameter estimation

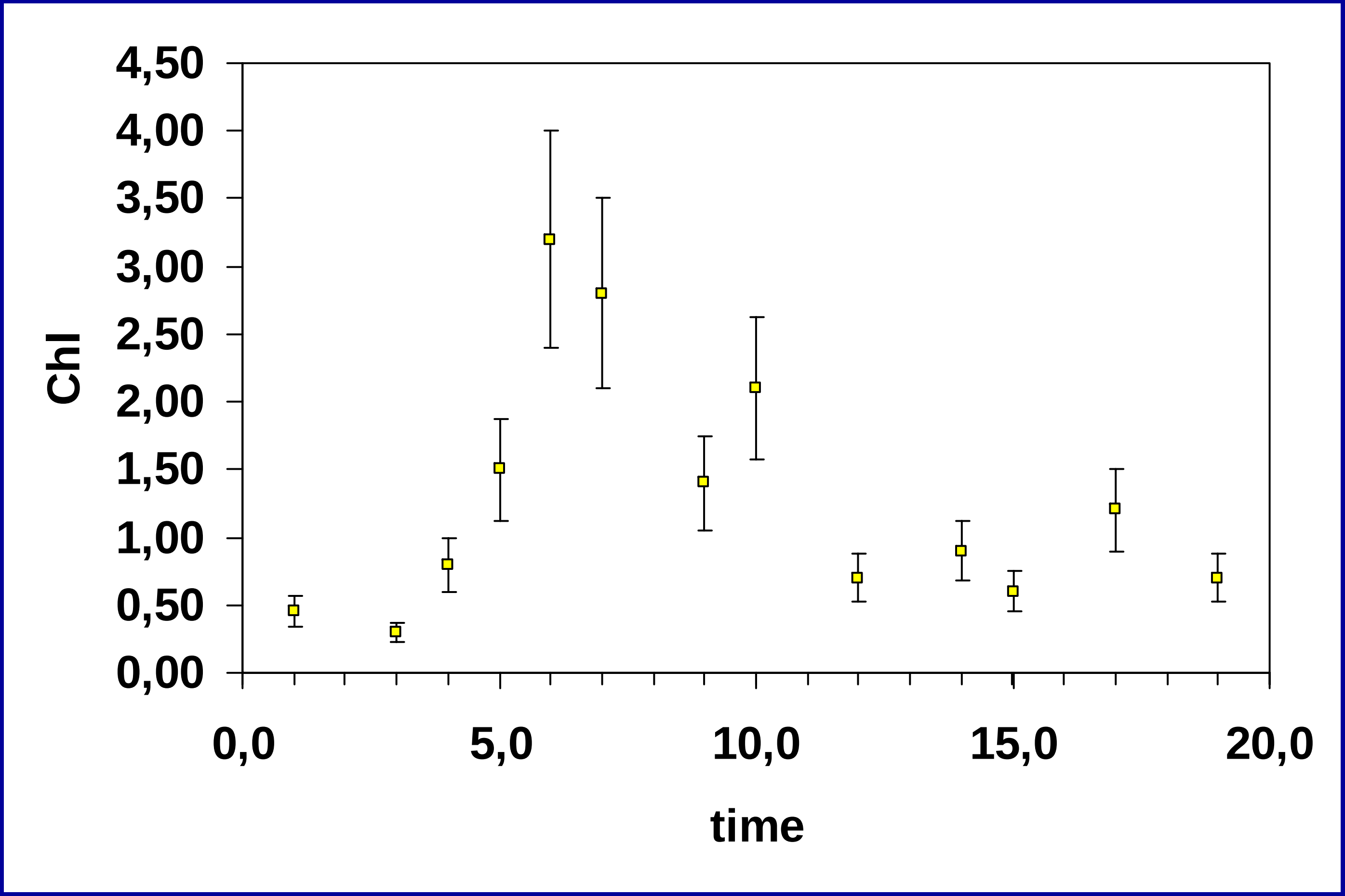
- Treats model dynamics as falsifiable hypothesis.
- May improve long-term forecast

Why care about data assimilation?

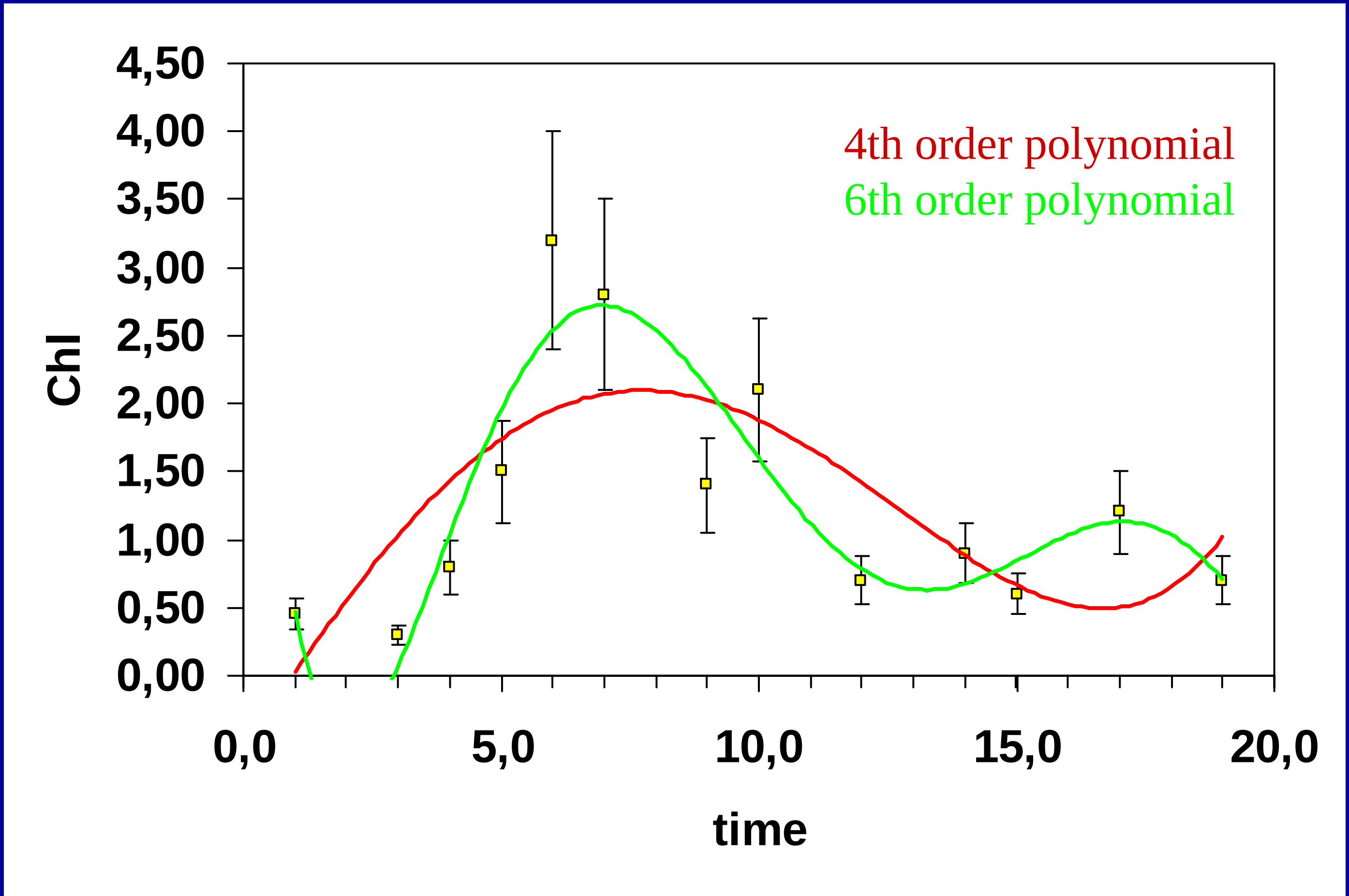
Observationalist´s view:

- Interpolate between isolated data points and different data types. (Often: large number of data that are difficult to interpret, few data that are easy to interpret.)
- Identify most valuable data (Observing System Simulation Experiments).

Example: Phytoplankton spring bloom



Example: Phytoplankton spring bloom



Polynomial-fit trajectory has no dynamical/mechanistic significance (e.g., negative values).

Why care about data assimilation?

Modeller´s view:

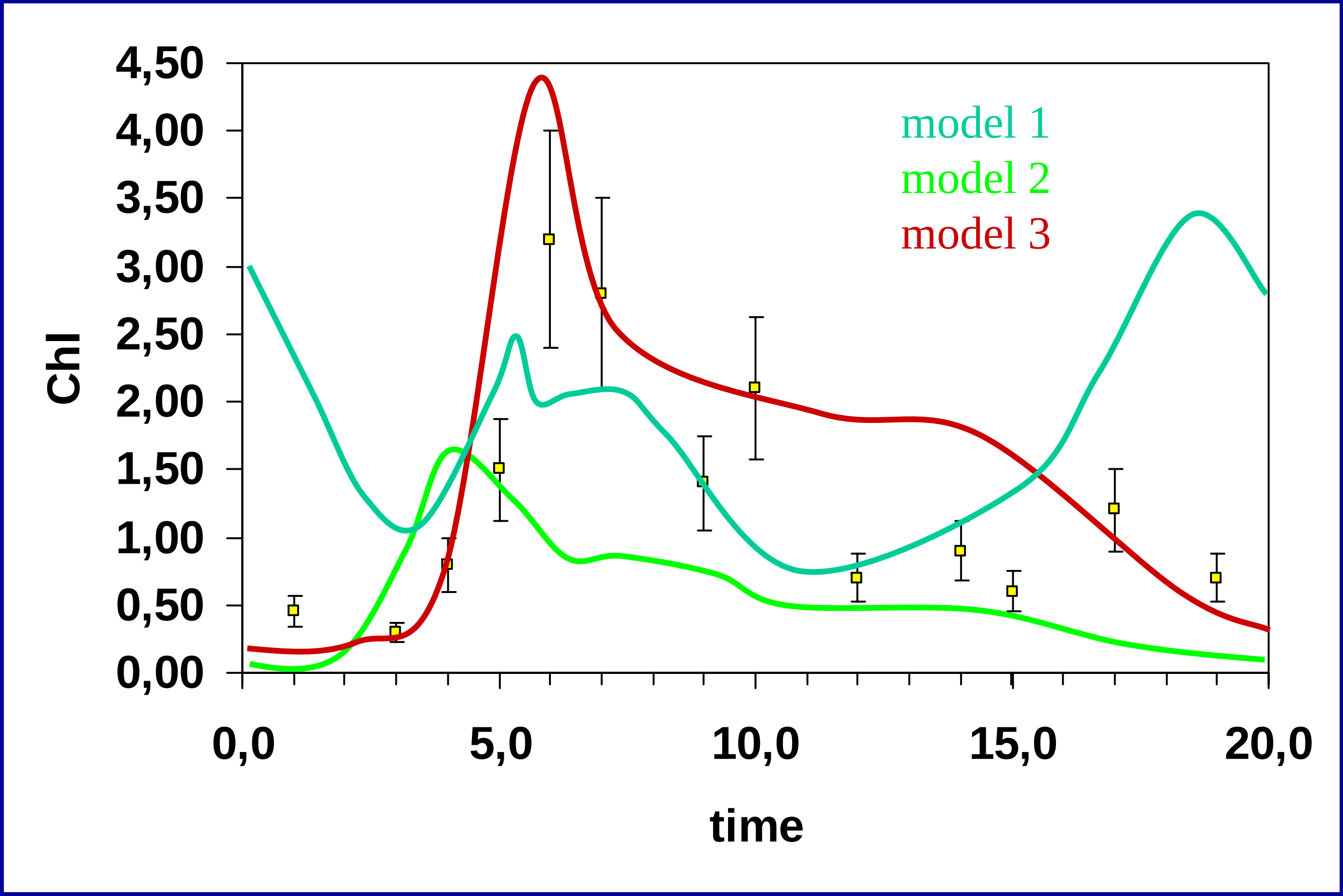
- Don´t know the governing equations (ecological modeling!).
  (Have some understanding of (belief in?) some ecological principles.)

- Data assimilation as hypothesis (i.e., model) testing.

- Correct for model deficiencies and improve model results.

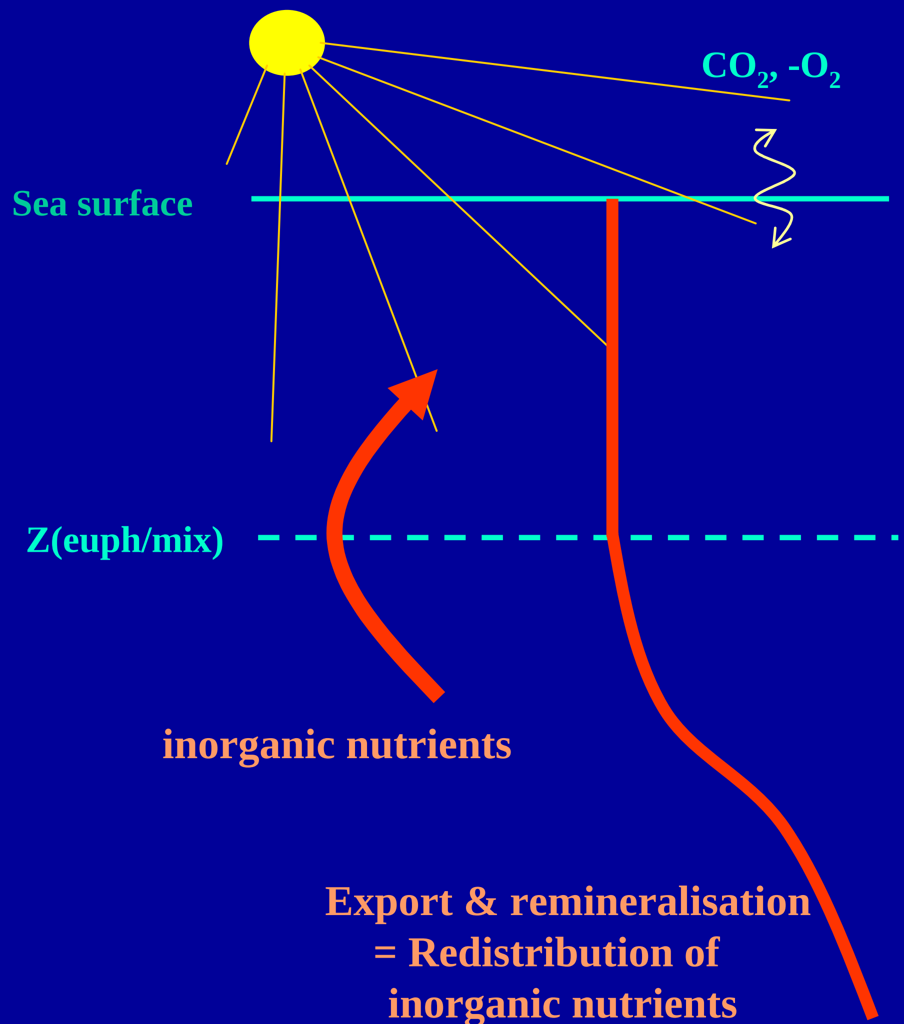
- Allow for systematic evaluation of model errors and suggest model improvements.

Example: Phytoplankton spring bloom



How can we combine models and data in a useful way?

Ecosystem Model Types: (i) Nutrient-Restoring



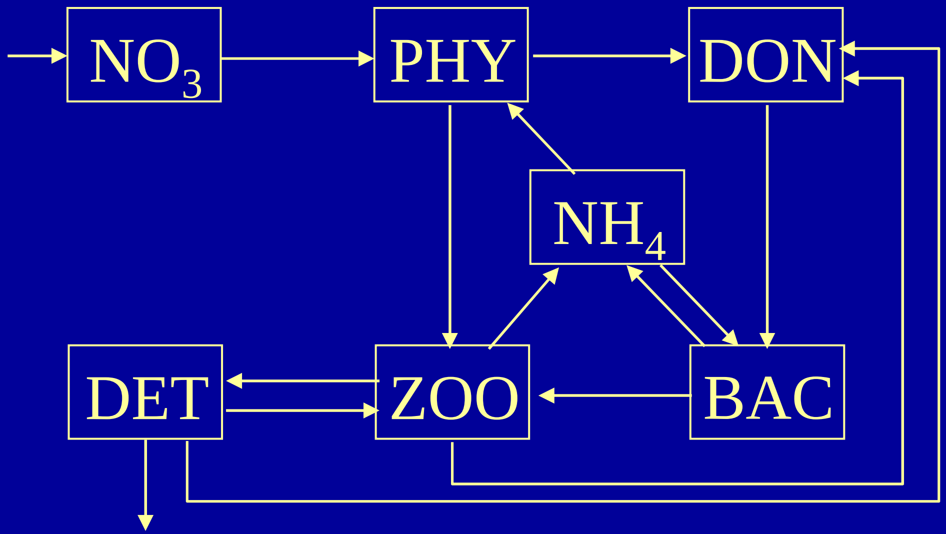
2 - 4 Parameters:

- nutrient uptake rate
- remineralisation profile

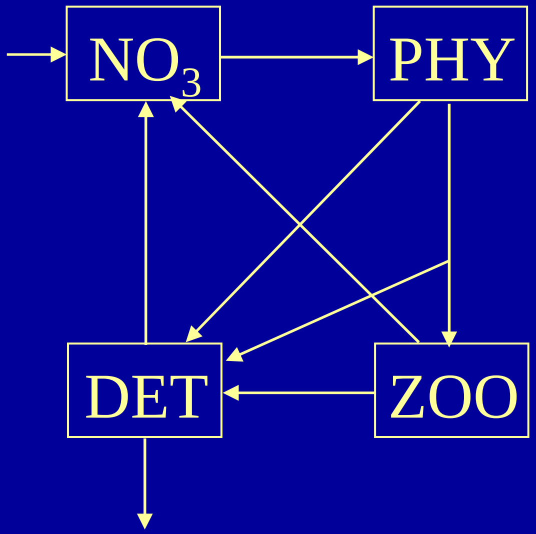
Examples:

- Bacastow & Maier-Reimer (1990,91)
- Najjar et al. (1992)
- OCMIP 1 & 2

Ecosystem Model Types: (ii) NPZD-type



(Fasham et al., 1990)



NPZD = Nutrient-Phytoplankton-Zooplankton-Detritus

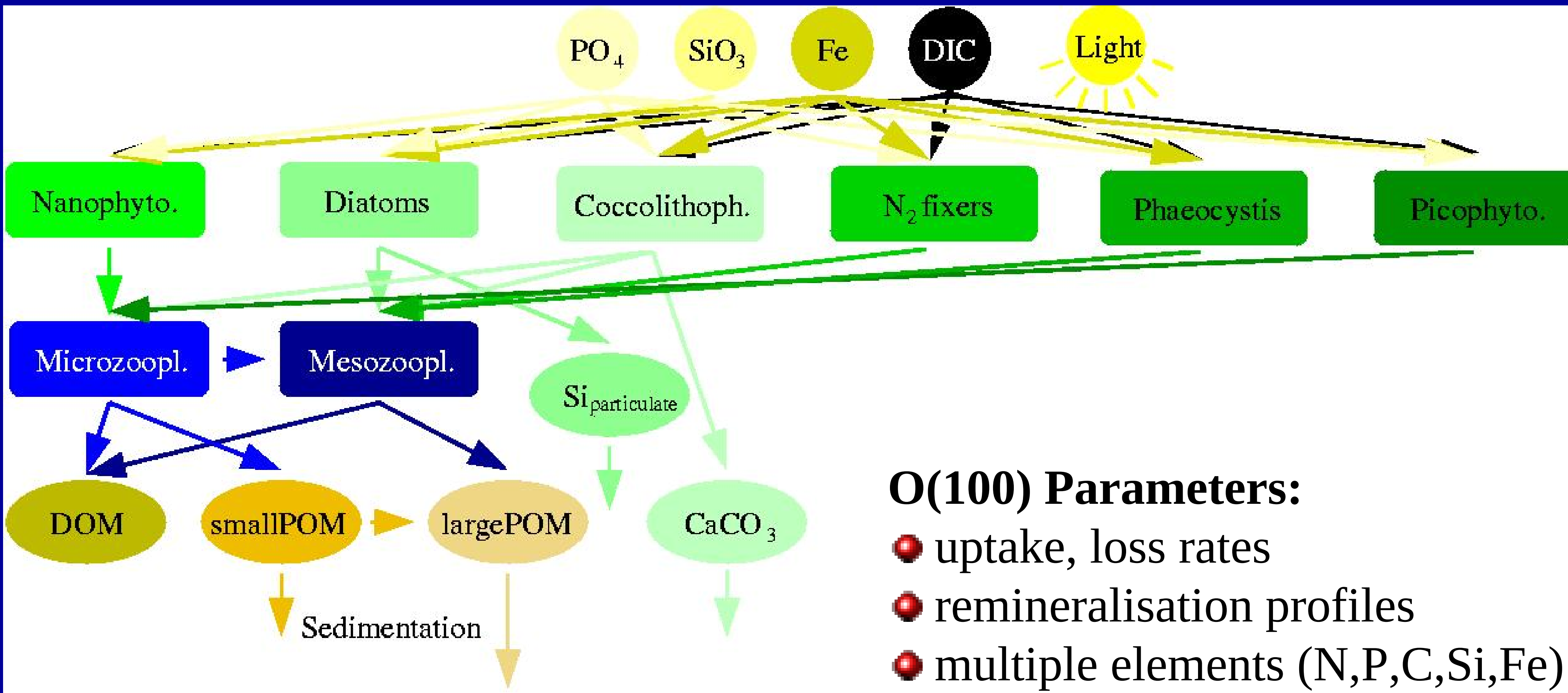
10-30 Parameters:

- uptake, loss rates
- remineralisation profile

Examples:

- Basin scale
  (Sarmiento et al., 1993; Fasham et al. ,1993; Chai et al., 1996; McCreary et al., 1996)
- Global Ocean
  (Six & Maier-Reimer, 1996)
- eddy-permitting basin scale
  (Oschlies and Garcon, 1998, 1999)
- eddy-resolving basin scale
  (Oschlies, 2002)

Ecosystem Model Types: (iii) "functional-group" type



O(100) Parameters:

- uptake, loss rates
- remineralisation profiles
- multiple elements (N,P,C,Si,Fe)

Examples:

- Moore et al. (2002)
- Aumont et al. (2004)
- "Dynamic Green Ocean Model" consortium

Dimension of parameter space

Ecosystem model	Number of adjustable parameters
Restoring	2-4
NPZD-type	10-30
Multiple functional groups, multiple elemental cycles	100-300
OGCMs	> 100 000

Most parameters will have natural bounds

(e.g., positiveness, physiological constraints)!

What is data assimilation?

Basic idea: Combine the most useful bits of observations and models.

- Extract information particularly from “high-quality“ data.
- Use natural laws coded into the model to interpolate (extrapolate) between different observations and data types. (mass/energy/momentum conservation, ecological rules, ...)

Basic requirements:

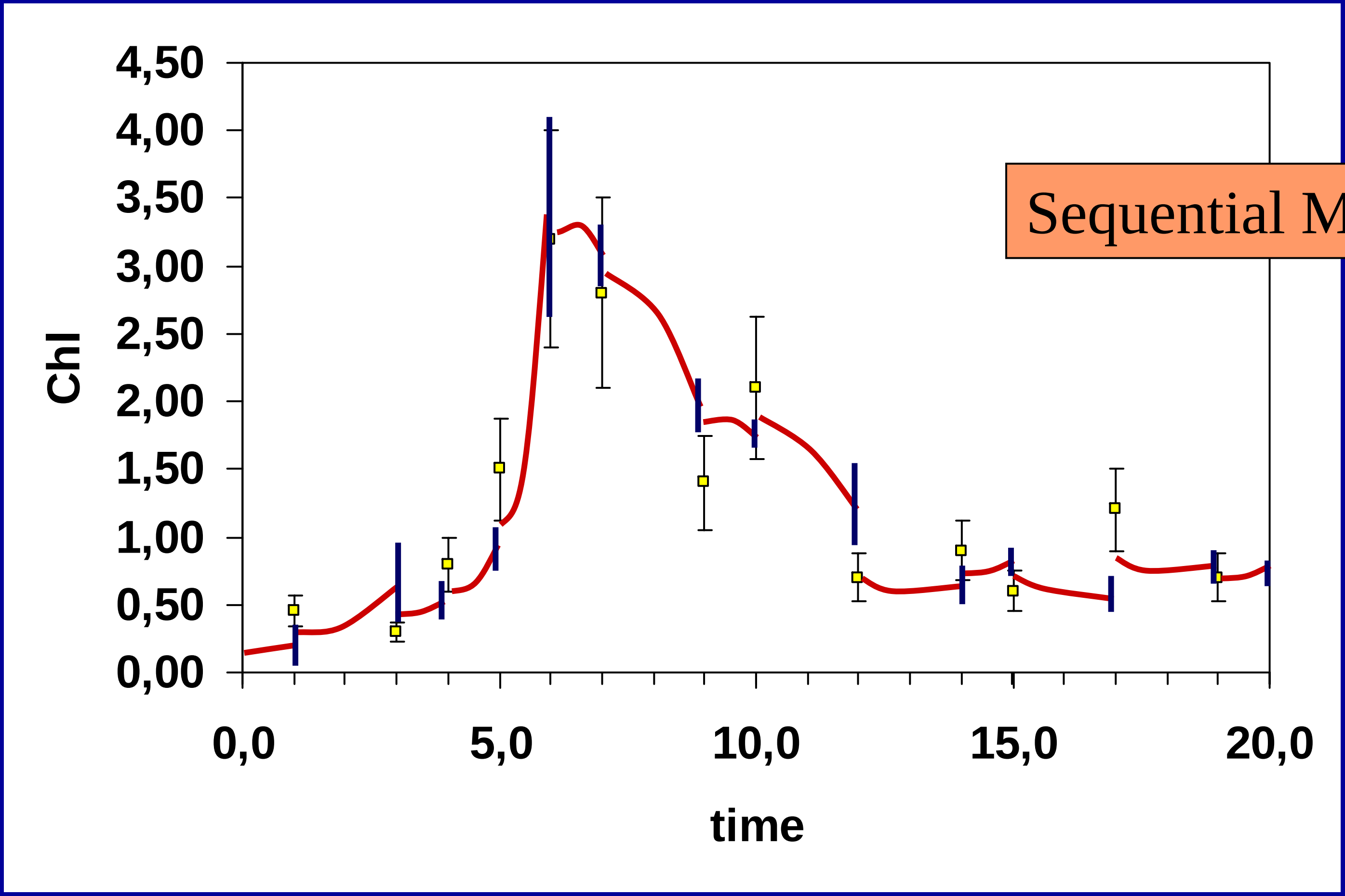
- Need to have some idea about data qualitity/errors.
- Need to have some idea about model quality/errors.

Sequential Methods

- Make use only of past observations along model trajectory.
- “Accumulate“ information along the model trajectory.
- Aim to improve present state vector.
- Kalman filter generates error covariance matrix of state vector (this is the computationally expensive part!).
- Little emphasis on dynamically consistent model trajectory.
- Employed by many operational forecast systems.

Data assimilation concepts

Optimal Interpolation / Kalman filter



- Computes error covariance of model state vector: Statistically optimal interpolation of full state vector whenever observations are available.
- Model trajectory is only piecewise consistent.
- Information accumulates with time (only past observations are exploited)

Variational Methods

- Search for “optimal“ model trajectory that fits the data and exactly obeys model dynamics.

Clue:

- Dynamical model solution depends on a set of control parameters (initial conditions, biological parameters, physical parameters, forcing,...).
- Control parameter vector **p**.

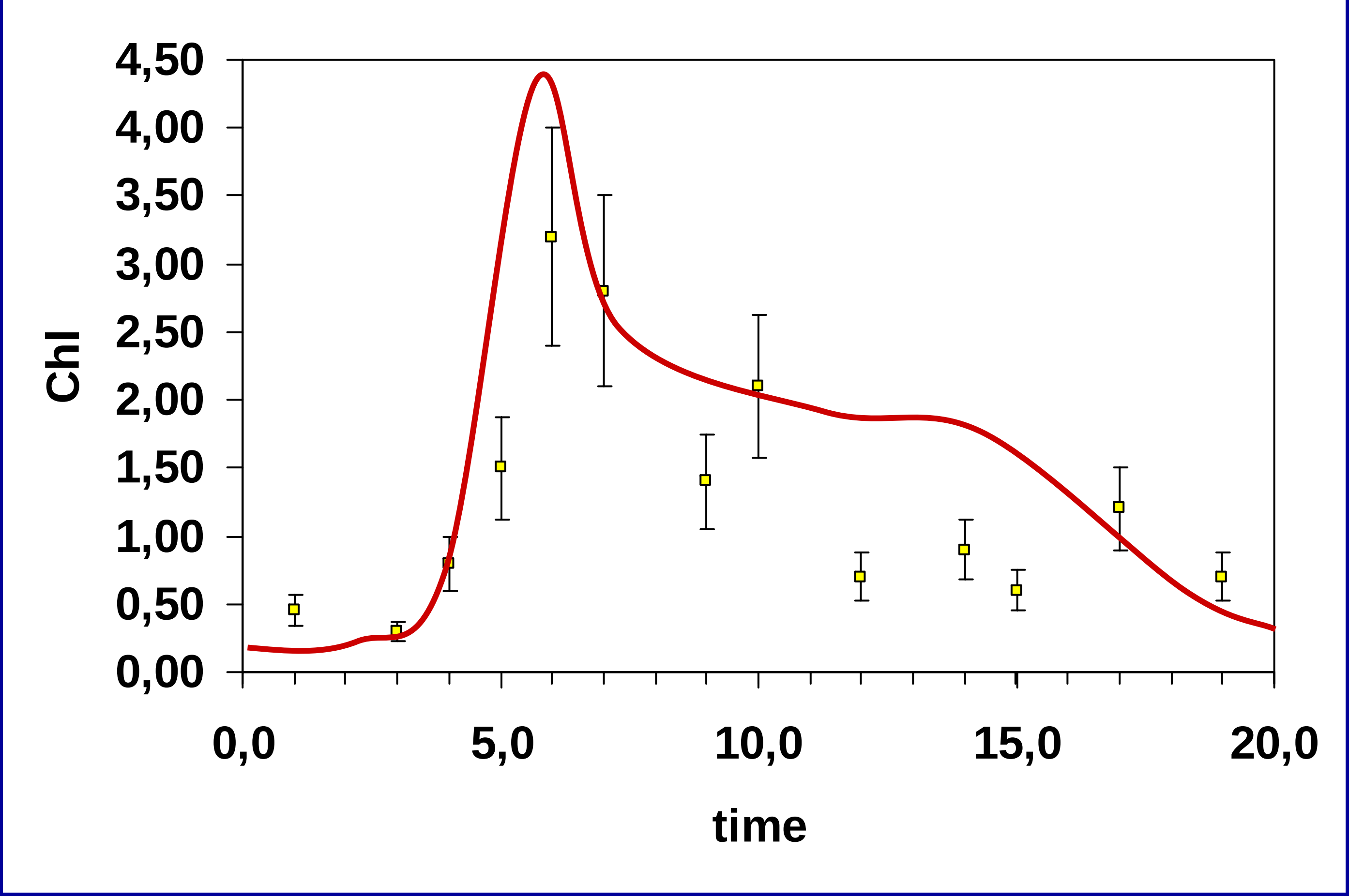
Define cost function of model-data misfits: $J = J(\mathbf{p})$

e.g.

$$J(\mathbf{p}) = \sum_{j=1}^{M} \left[d_j - m_j(\mathbf{p})\right]^2$$

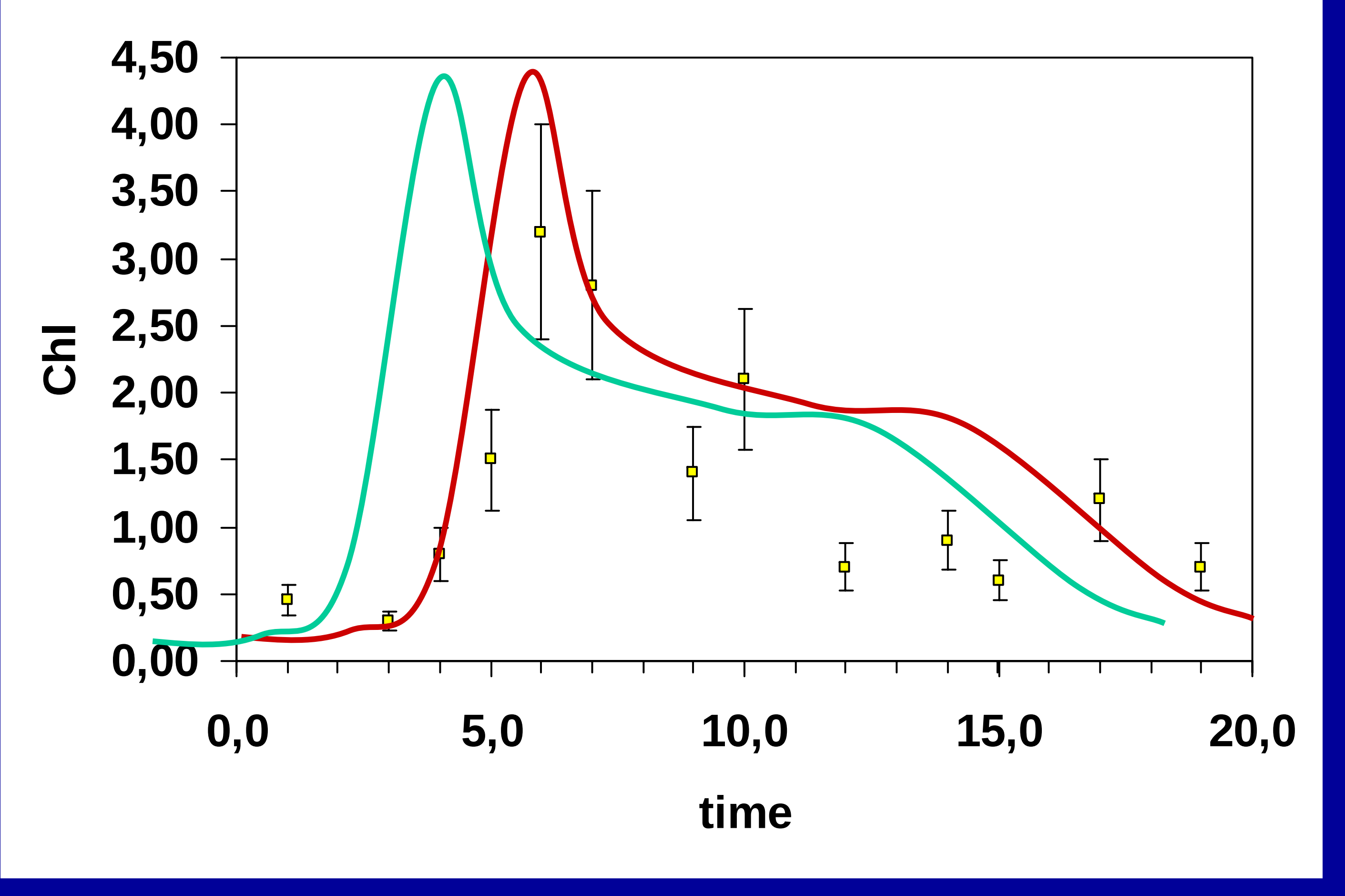
How to define misfit?

Particularly challenging for biogeochemical modelling!



How to define misfit?

What about phase errors in underlying physical model?



Variational Methods

- Account for error covariance of model-data misfits. Introduce weighting matrix **W**:

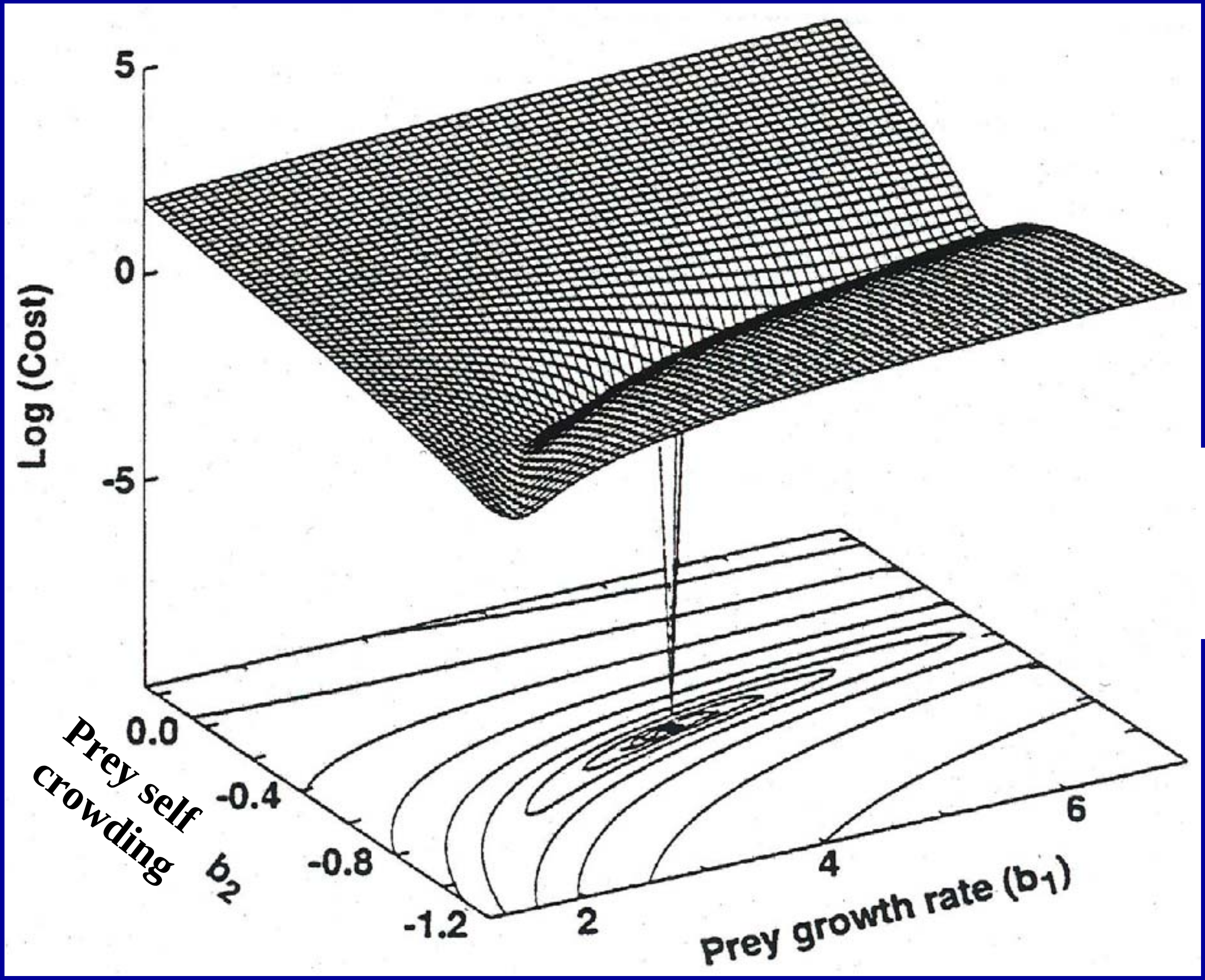
$$J(\mathbf{p}) = \sum_{i,j=1}^{M} [d_i - m_i(\mathbf{p})]\, W_{ij}\, [d_j - m_j(\mathbf{p})]$$

- Optimal model solution for $J(\mathbf{p}) = J(\mathbf{p}^{opt}) = \min!$
- Have to determine set of optimal control parameters $\mathbf{p}^{opt}$.
- Problem of constrained optimisation
(constraints are model dynamics in $m_i(\mathbf{p})$).

Cost function

$$J(\mathbf{p}) = \sum_{i,j=1}^{M} [d_i - m_i(\mathbf{p})]\, W_{ij}\, [d_j - m_j(\mathbf{p})]$$

- Quantifies model-data misfit.
- $J(\mathbf{p})$ is a function of control parameters $\mathbf{p}$!



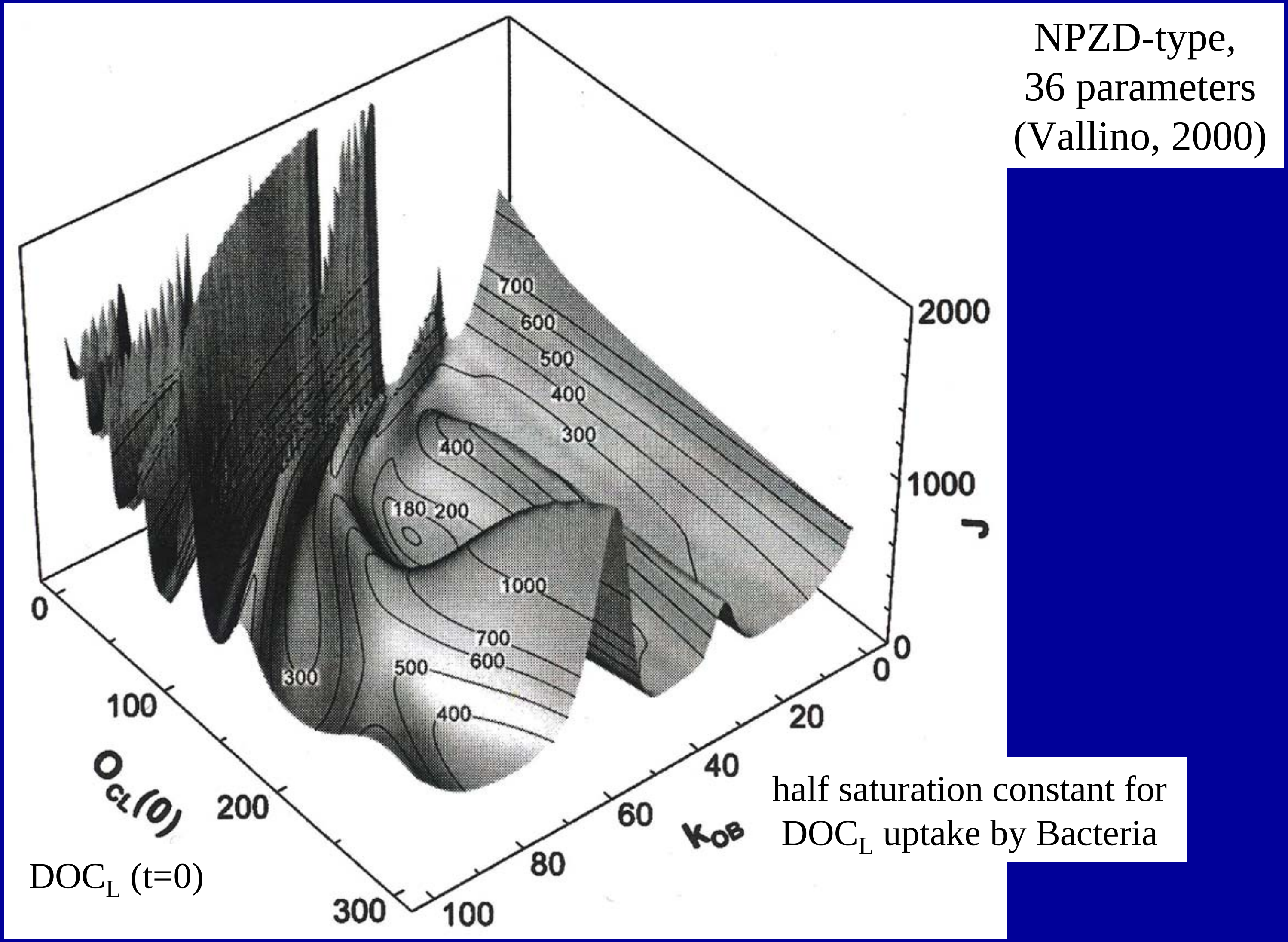
Predator-Prey model, 6 parameters (Lawson et al., 1995)

Difficult to visualise for more than 2 parameters!

Cost function

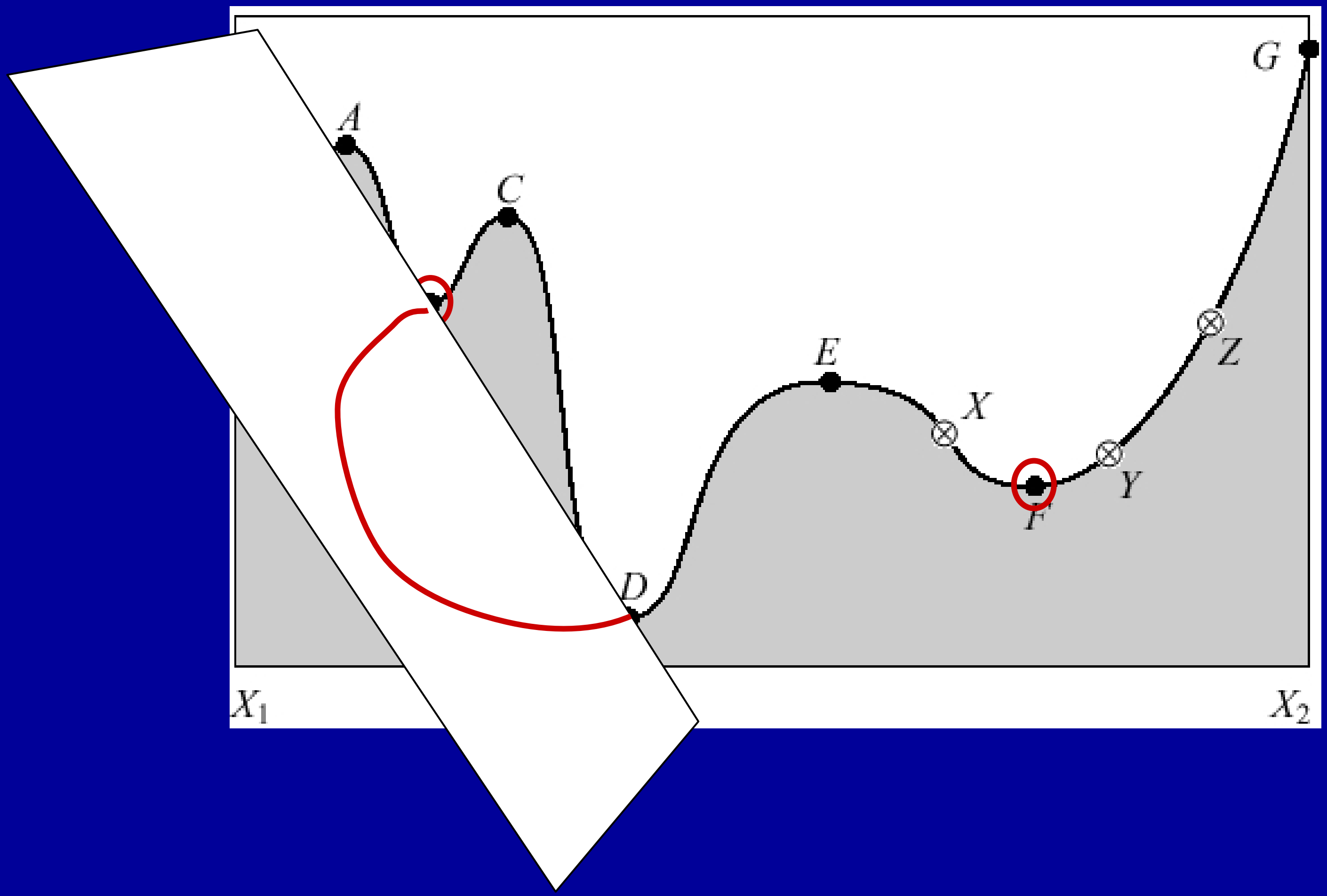
$$J(\mathbf{p}) = \sum_{i,j=1}^{M} [d_i - m_i(\mathbf{p})]\, W_{ij}\, [d_j - m_j(\mathbf{p})]$$

- May have complicated form (particularly for strongly non-linear models!).



On local minima

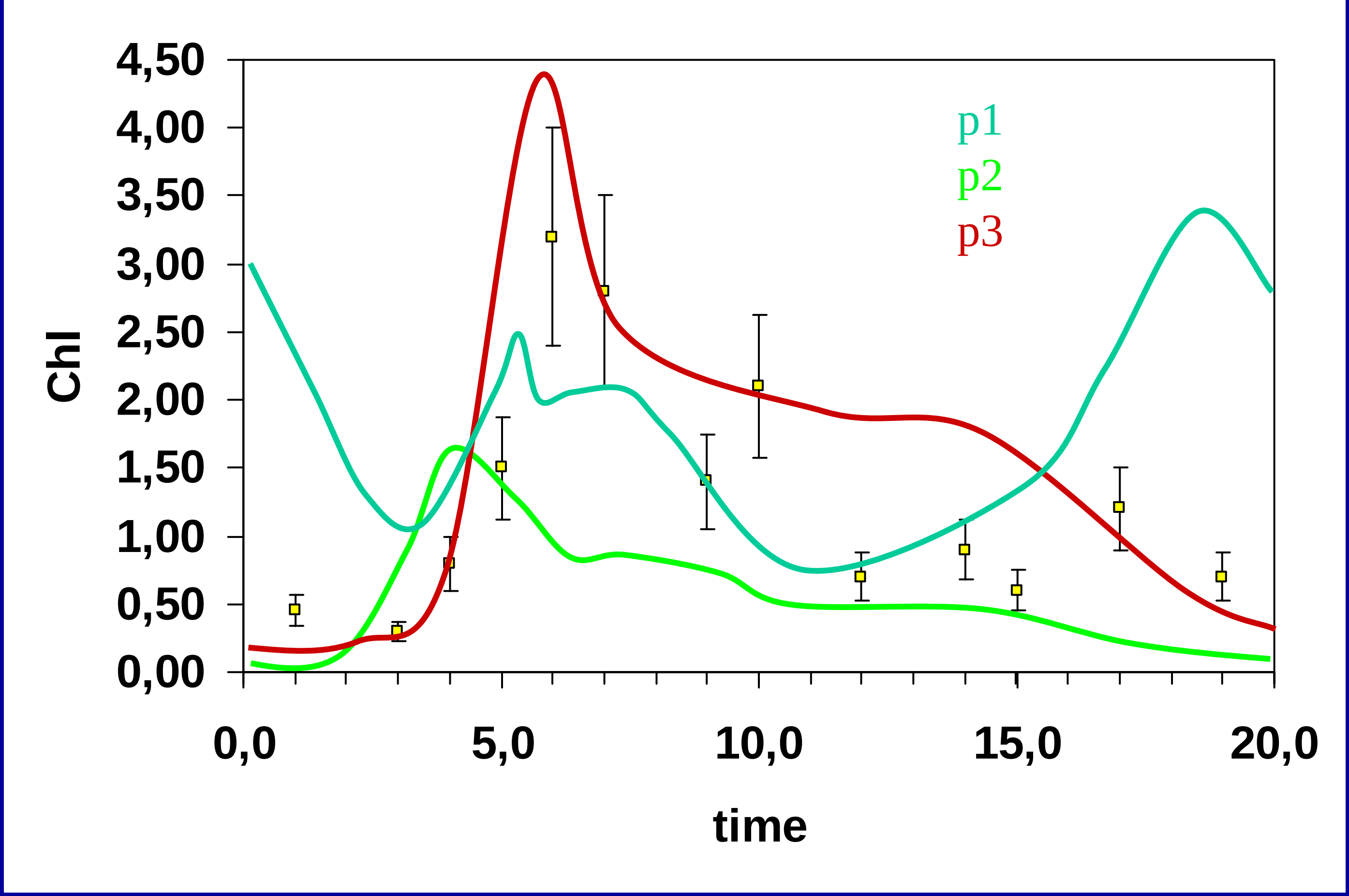
- What looks like a local minimum in one dimension...



- ...is not necessarily a local minimum in a higher dimensional space!

Brief review of the different methods

Theme: Vary control parameters to minimise model-data misfit!



How to find the cost function minimum?

Three main types of minimisation algorithms:

- Brute force: Scanning of parameter space.
  Utilise only $J(\mathbf{p})$.

very few parameters
(10 trials, n parameters => 10^n)

- Brute force + memory: Stochastic methods.
  Utilise $J(\mathbf{p})$ and information from previous iterations.
  Examples: Simulated annealing, genetic algorithms,...

few parameters, local minima

- Elegant & thoughtful: Gradient descent methods.
  Utilise $J(\mathbf{p})$ and $\text{grad}_p J(\mathbf{p})$.
  Example: Adjoint method.

many parameters, no local minima

Method of choice will depend on number of parameters and cost function shape!

Method Selection

Ecosystem model	Number of adjustable parameters	
Restoring	2-4	brute force & others
NPZD-type	10-30	stochastic methods, adjoint method
Multiple functional groups, multiple elemental cycles	100-300	stochastic(?), adjoint method(?)
OGCMs	> 100 000	adjoint method

Stochastic Methods: Simulated Annealing

From solid state physics: slow cooling of metals, alloys,...

- to avoid errors in crystal lattice (local minima of potential energy)
- to obtain perfect cristal (minimum energy state).

Boltzman factor: *exp(-ΔE/kT)* ~ Probability for up-hill *ΔE* being accepted)

Metropolis function (Metropolis et al., 1953) :

$$f(J(\mathbf{p}_{old}), J(\mathbf{p}_{new}), T) = \exp\left(-\frac{J(\mathbf{p}_{new}) - J(\mathbf{p}_{old})}{T}\right)$$

Algorithm: random walk with decreasing probability for up-hill steps.

1) $\mathbf{p}_{new}$ = generate($\mathbf{p}_{old}$)

2) Accept or reject $\mathbf{p}_{new}$ according to Metropolis function.

3) Decrease “temperature“ T and goto 1) until convergence.

Simulated Annealing

Example:

Matear (1995): 0D models run at OWS Papa

1. NPZ model (Evans & Parslow, 1985) , 14 parameters
2. NPZ_1Z_2 model, 18 parameters
3. 7-compartment model (Fasham et al., 1990) , 25 parameters

Conclusions:

- Available data do not justify complex models.
  All models fit data about equally well!
- Can constrain < 10 parameters (using N, P, PP, (Z) observations).
- Simulated annealing superior to conjugate gradient method for models 2 and 3 (presumably because of local minima).

Stochastic Methods: Genetic Algorithms (GA)

Concept: Survival of the fittest.

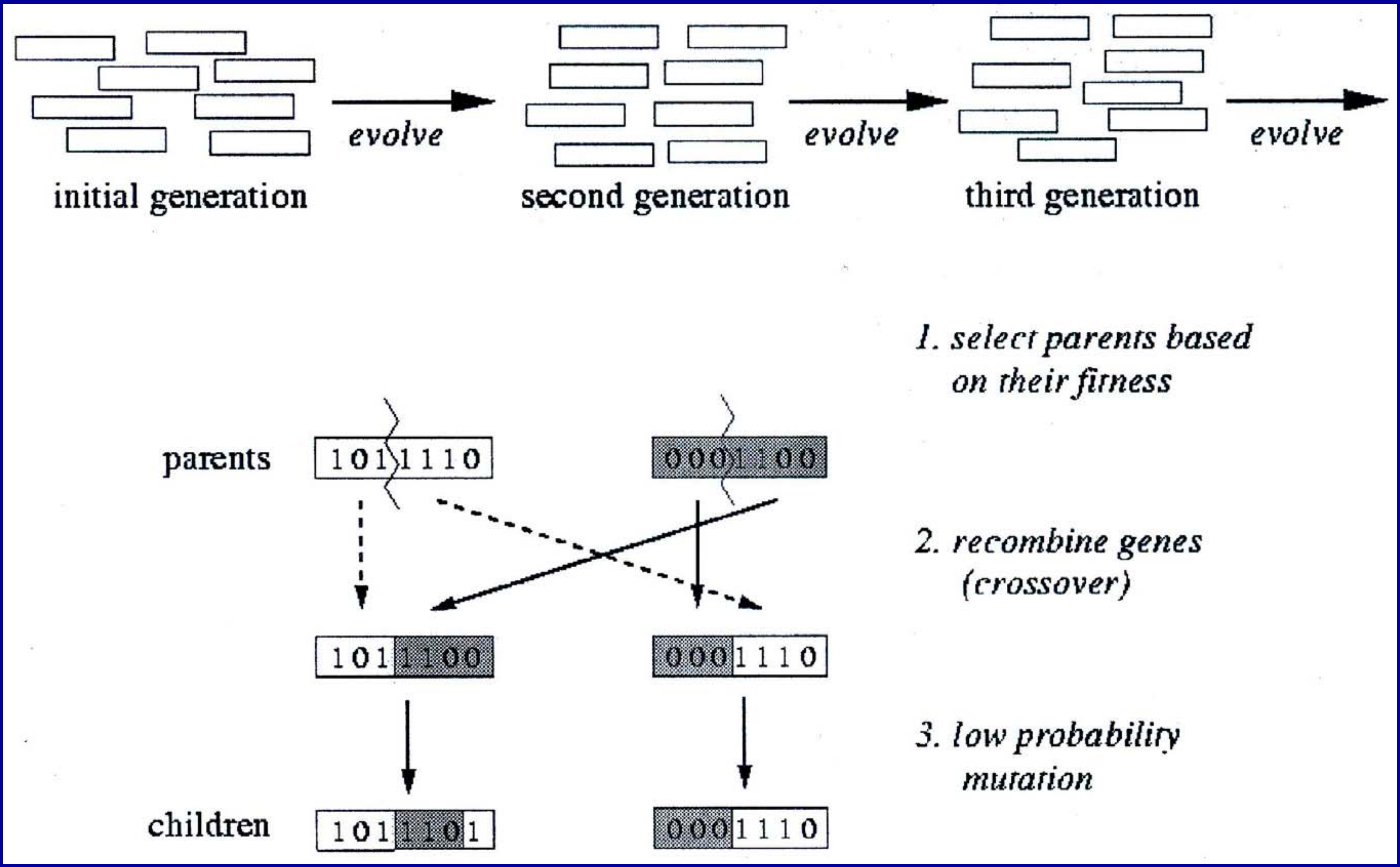
Fitness of parameter set **p** is measured by the cost function $J(\mathbf{p})$.

- Chromosome: parameter vector **p** (typically in binary notation)
- Generation: suite of parameter vectors **p**
- Reproduction: Recombination and mutation of one **p**-generation.
- Selection: according to cost function $J(\mathbf{p})$.

Sometimes (μGA):

- Elitism: fittest parameter vector always survives one generation.

Stochastic Methods: Genetic Algorithms (GA)



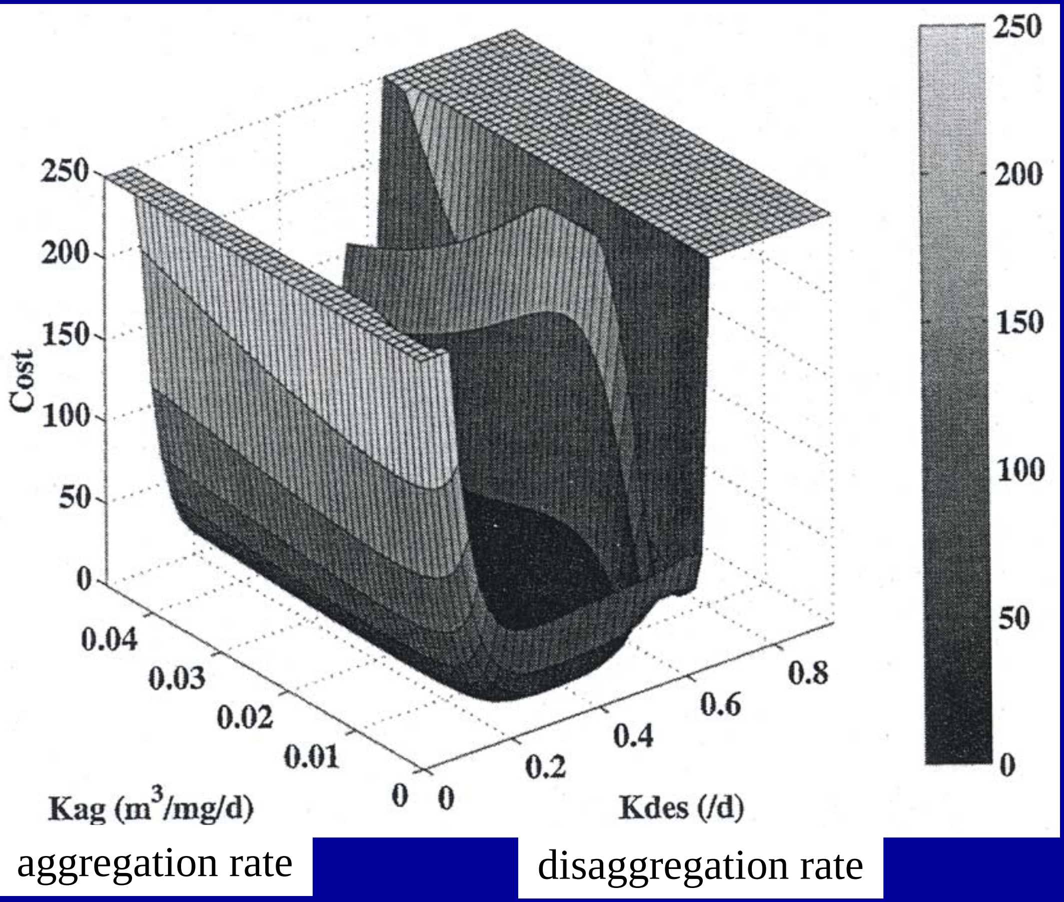
Genetic Algorithms (GA)

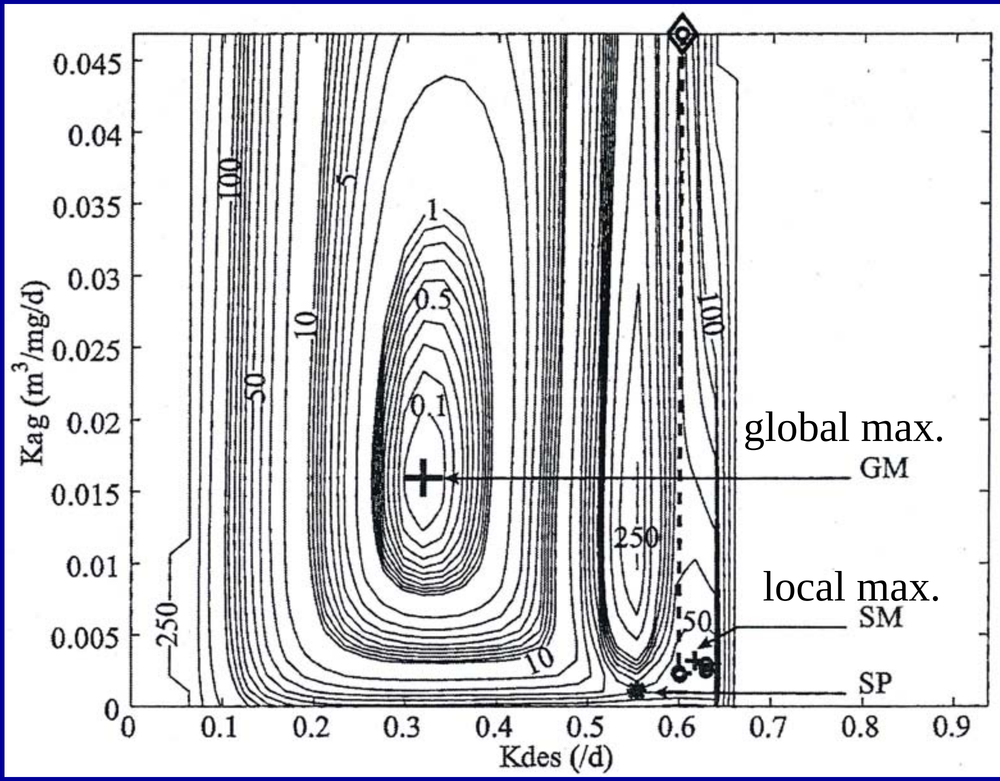
Example 1:

- Athias et al. (2000): Identical twin experiments (model of oceanic particle cycling: dissolved, suspended, sinking), method intercomparison.

Genetic Algorithms (GA)

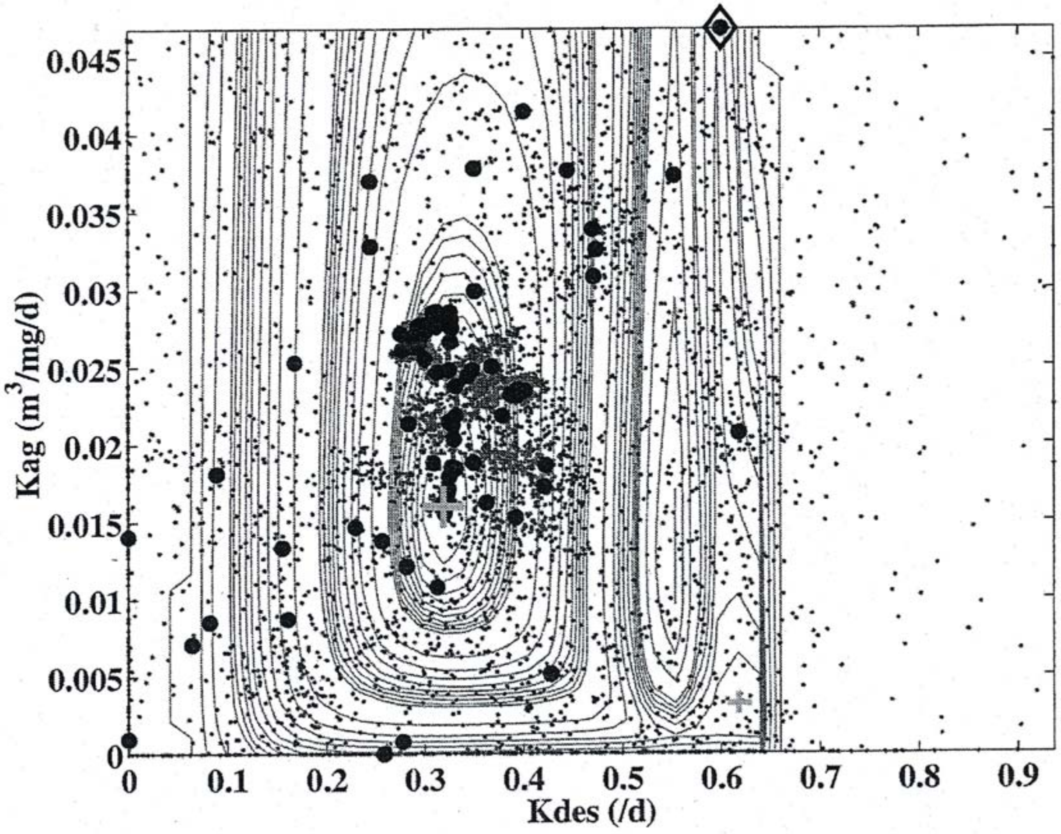
2D section of cost function (Athias et al., 2000)

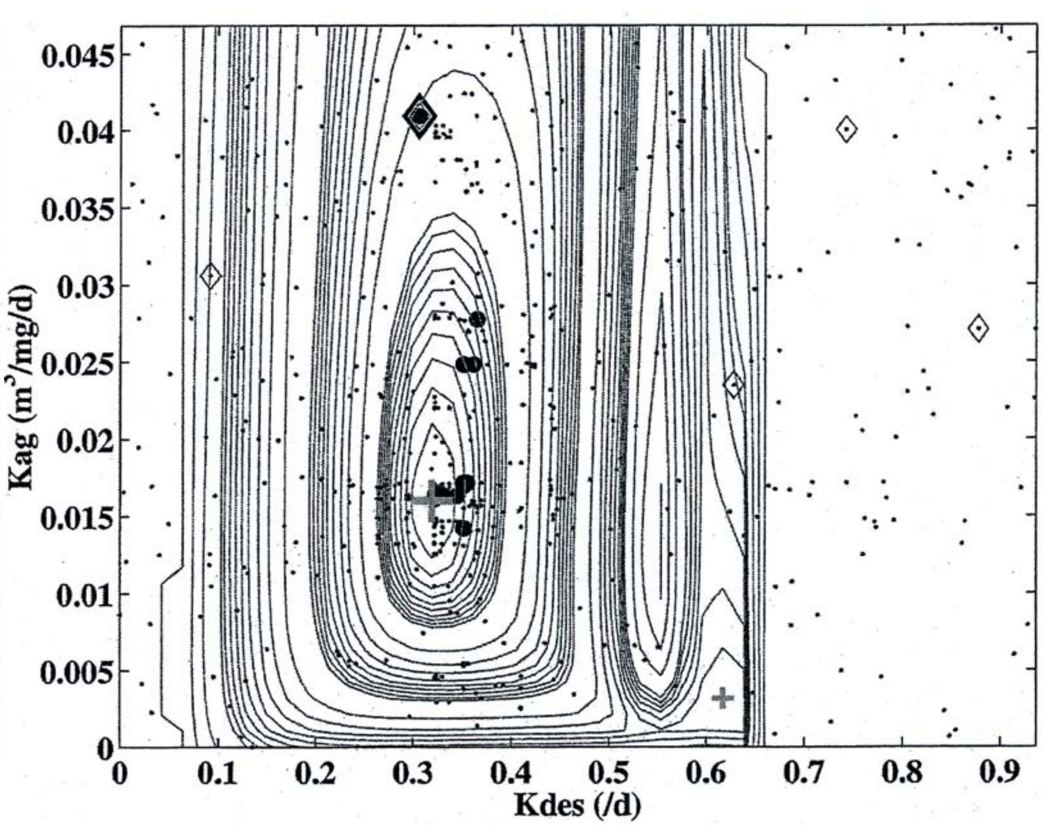




Simulated Annealing versus Genetic Algorithms (GA)

(Athias et al., 2000)





21627 parameter vectors ,
black dots: best points at each of 156 annealing steps.

1000 parameter vectors
black dots: best individuals of each of 200 generations.

Genetic Algorithms (GA)

Example 1:

- Athias et al. (2000): Identical twin experiments, method intercomparison.

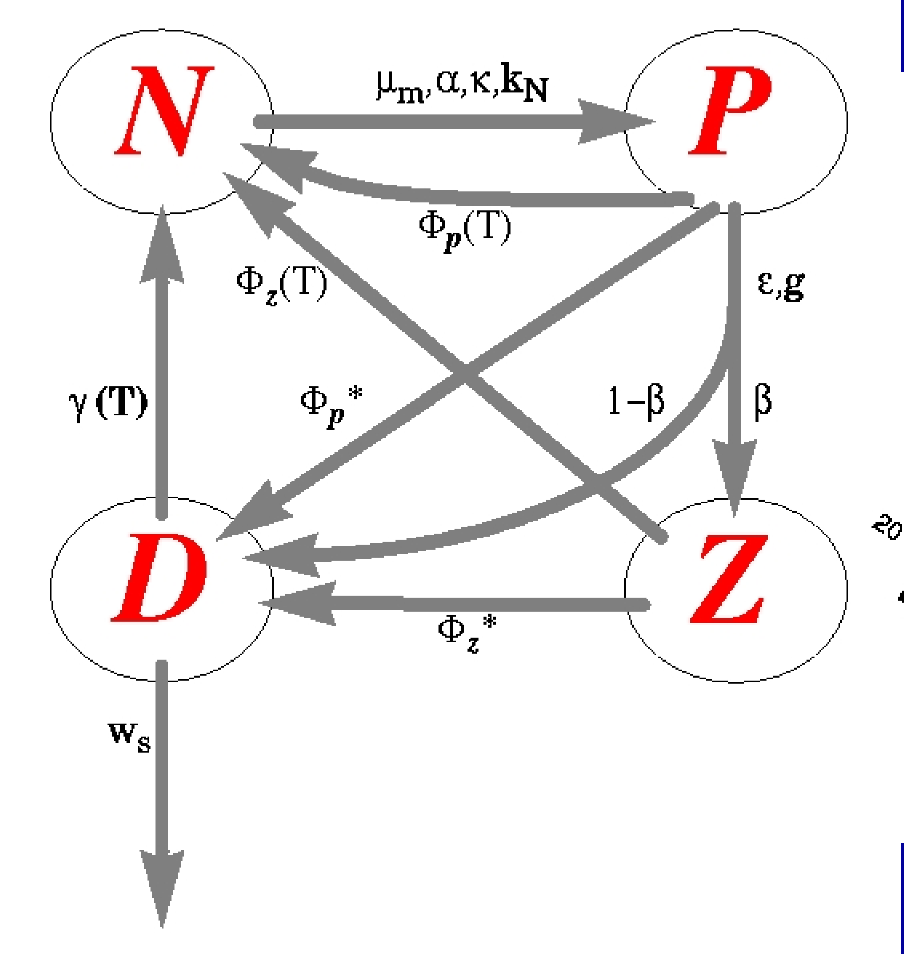
Conclusion:

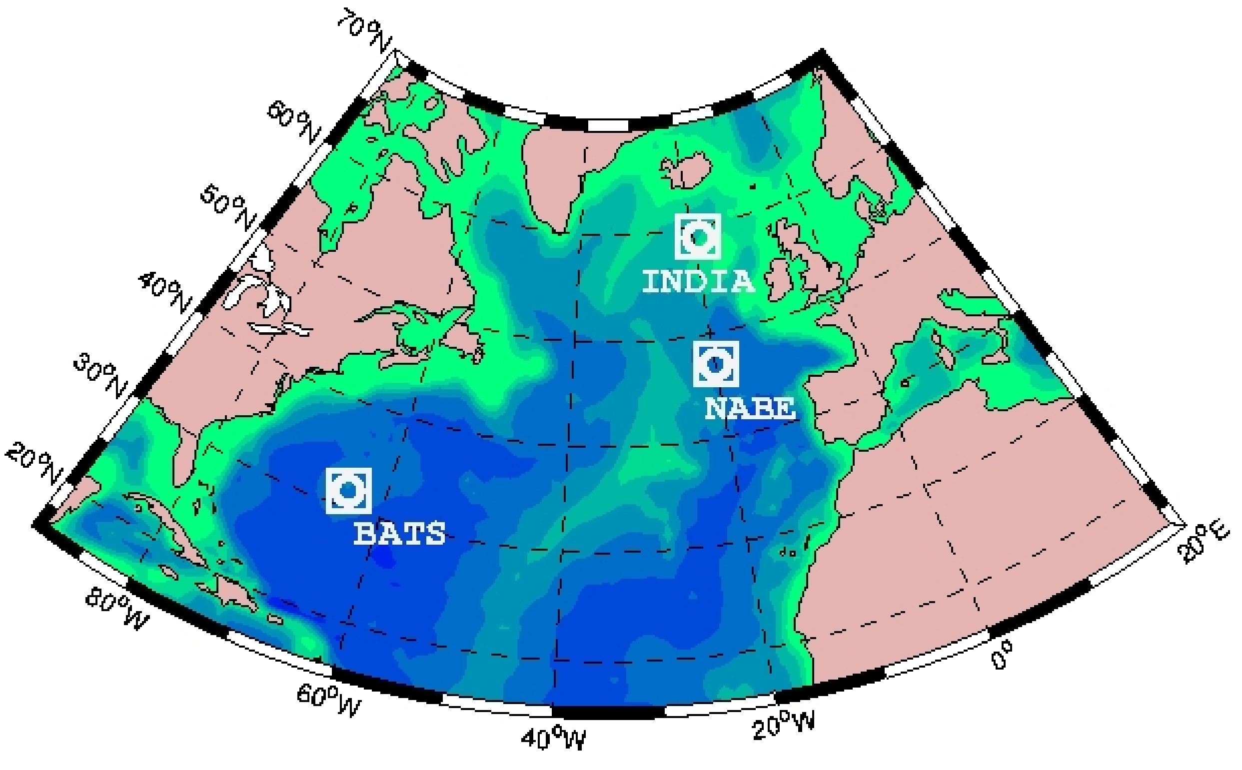
- GA faster and more robust (in avoiding local minima) than simulated annealing.

Genetic Algorithms (GA)

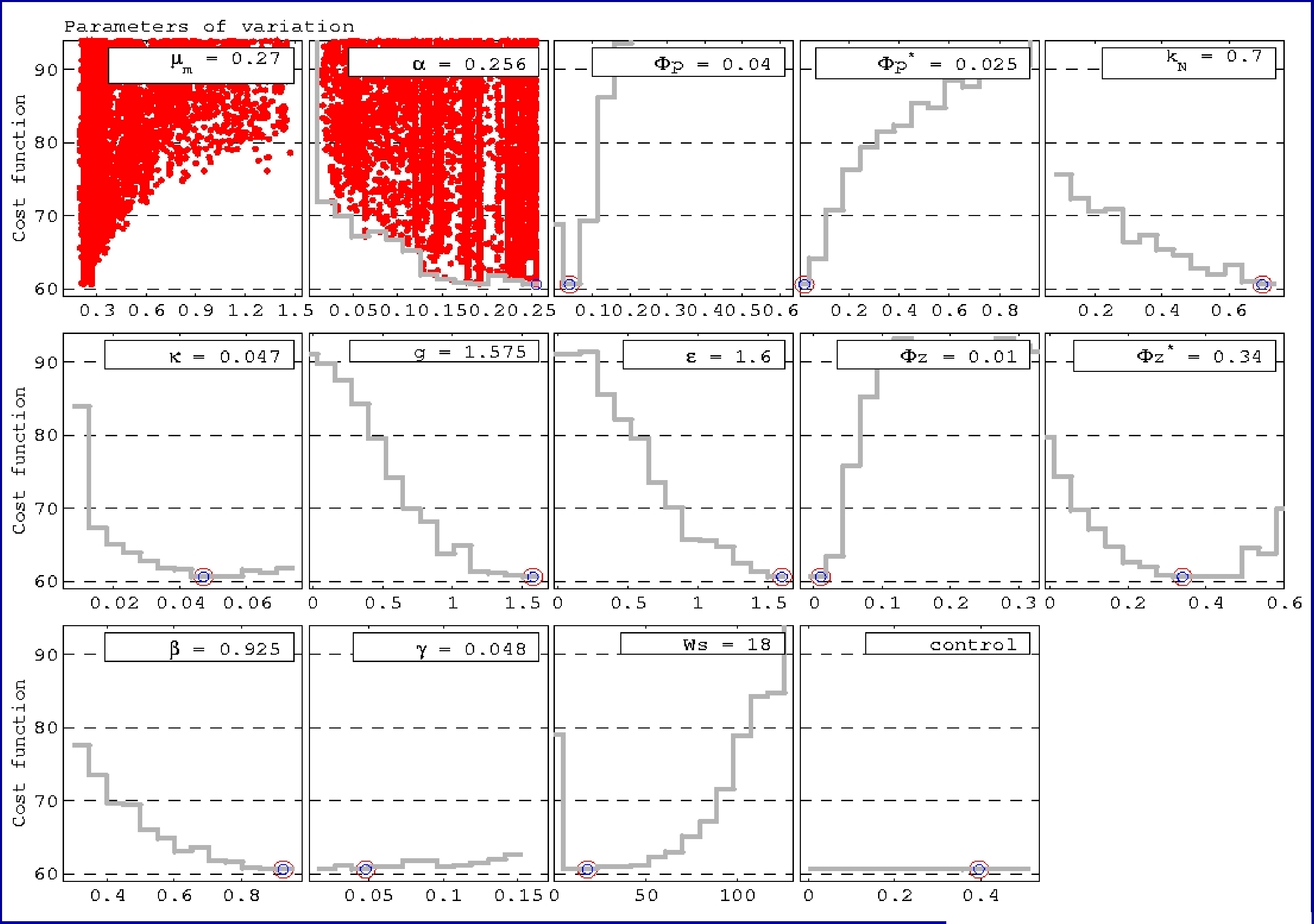
Example 2:

- Schartau & Oschlies (2003): 1D NPZD model, simultaneous optimisation at BATS, NABE, OWS India.





Genetic Algorithms (GA)



(Schartau & Oschlies, 2003)

Genetic Algorithms (GA)

Example 2, Conclusions:

- Can essentially constrain all of the 13 parameters. BUT...
    - ...some optimal parameters at the prior limits!
    - ...model-data misfit on average about 3 standard deviations!
- GA is robust, needs 26 000 iterations
  (13 individuals per generation, 2000 generations).
- Weighting coefficients that enter the cost function are the tricky and to some extent always subjective part (weighting of different observations at different stations, steady state constraints,...).

Variational Methods: Adjoint Method

- Cost function:

$$J(\mathbf{p}) = \sum_{i,j=1}^{M} \left[d_i - m_i(\mathbf{p})\right] W_{ij} \left[d_j - m_j(\mathbf{p})\right]$$

- Minimize $J(\mathbf{p})$ with model dynamics $E_j(\mathbf{x},\mathbf{p}) = 0$ as strong constraint!
- Introduce Lagrange multipliers λ_j => Lagrange function L :

$$L(\mathbf{p}, \lambda, \mathbf{x}) = J(\mathbf{p}) + \sum_{j}^{j_{max}} \lambda_j E_j$$

Unconstrained minimisation of $L(\mathbf{p},\lambda,\mathbf{x})$ = constrained minimisation of $J(\mathbf{p})$.

Unconstrained minimisation of Lagrange function

$$L(\mathbf{p}, \lambda, \mathbf{x}) = J(\mathbf{p}) + \sum_{j}^{j_{max}} \lambda_j E_j$$

$$\frac{\partial L}{\partial \lambda_j} = E_j = 0$$

$$\frac{\partial L}{\partial x_i} = \sum_{\mu\nu} [d_\mu - m_\mu(\mathbf{p})] \, W_{\mu\nu} \frac{\partial m_\nu}{\partial x_i} + \sum_j \lambda_j \frac{\partial E_j}{\partial x_i} = 0$$

$$\frac{\partial L}{\partial p_i} = \sum_{\mu\nu} [d_\mu - m_\mu(\mathbf{p})] \, W_{\mu\nu} \frac{\partial m_\nu}{\partial p_i} + \sum_j \lambda_j \frac{\partial E_j}{\partial p_i} = 0$$

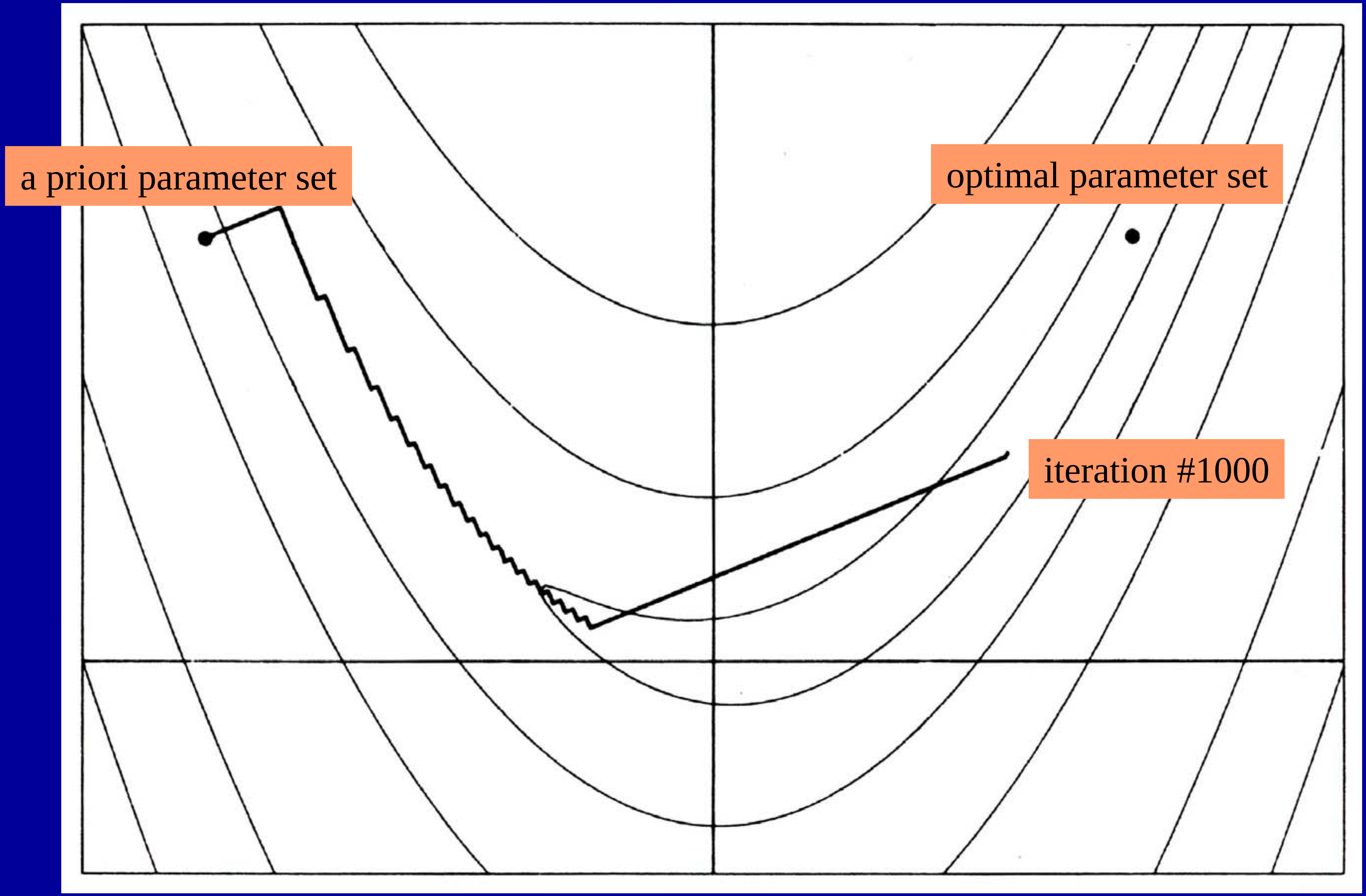
- Same number of equations as unknowns!

Adjoint Method

- Main advantage:
  - Computes complete vector $\text{grad}_p J(\mathbf{p})$ in a single run of forward + adjoint model.
  - Very efficient for high-dimensional parameter space!
- Main disadvantage:
  - Requires coding of the adjoint model (➔ automatic differentiation).
  - Some problems with strong non-linearities (e.g., “if“ statements).
- Useful only together with efficient gradient descend algorithm!
  - May have problems with local minima (if these exist...)

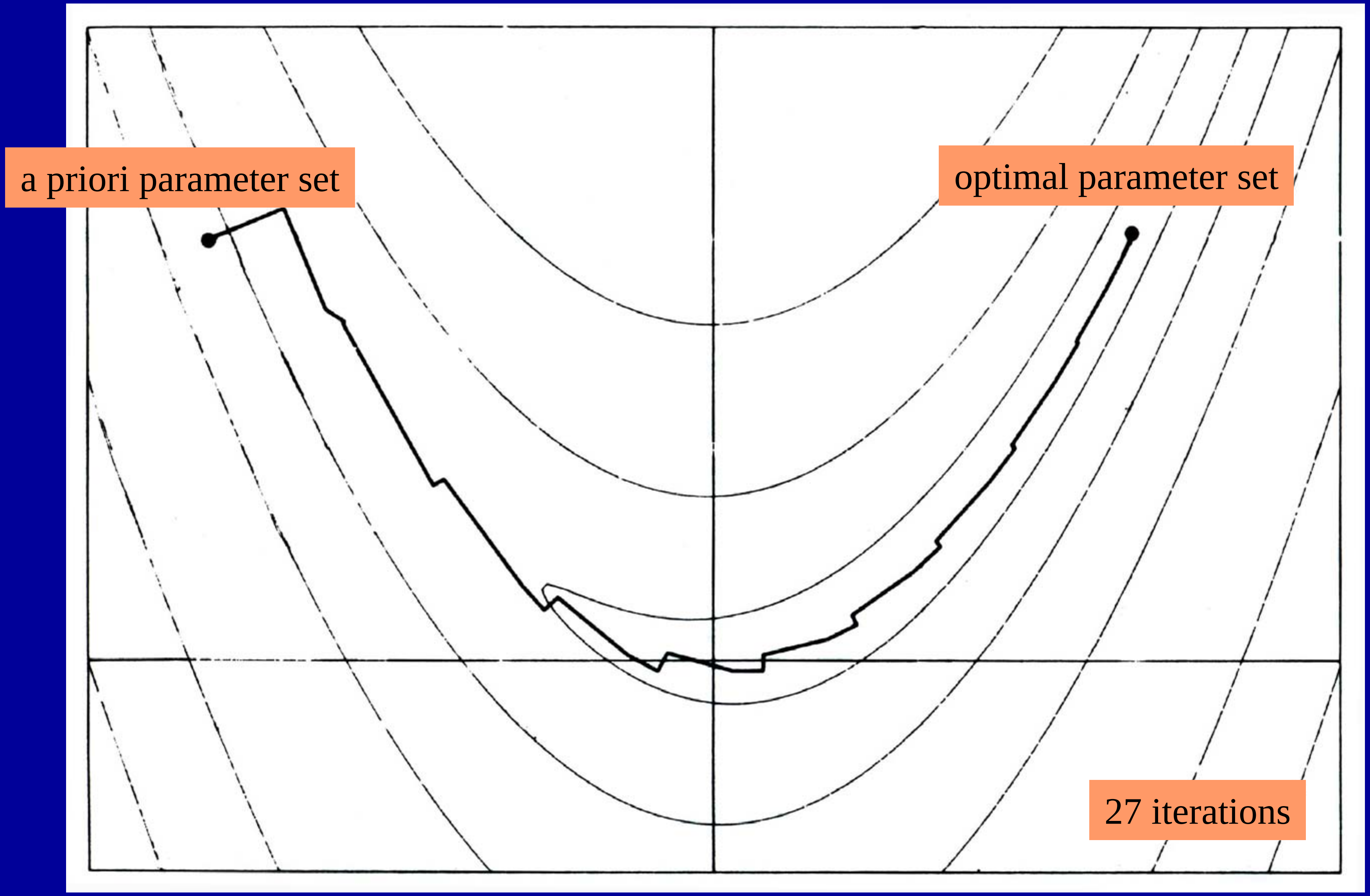
Adjoint Method: Yields Gradient Information!

How to exploit this information? (i) Steepest-descent algorithm



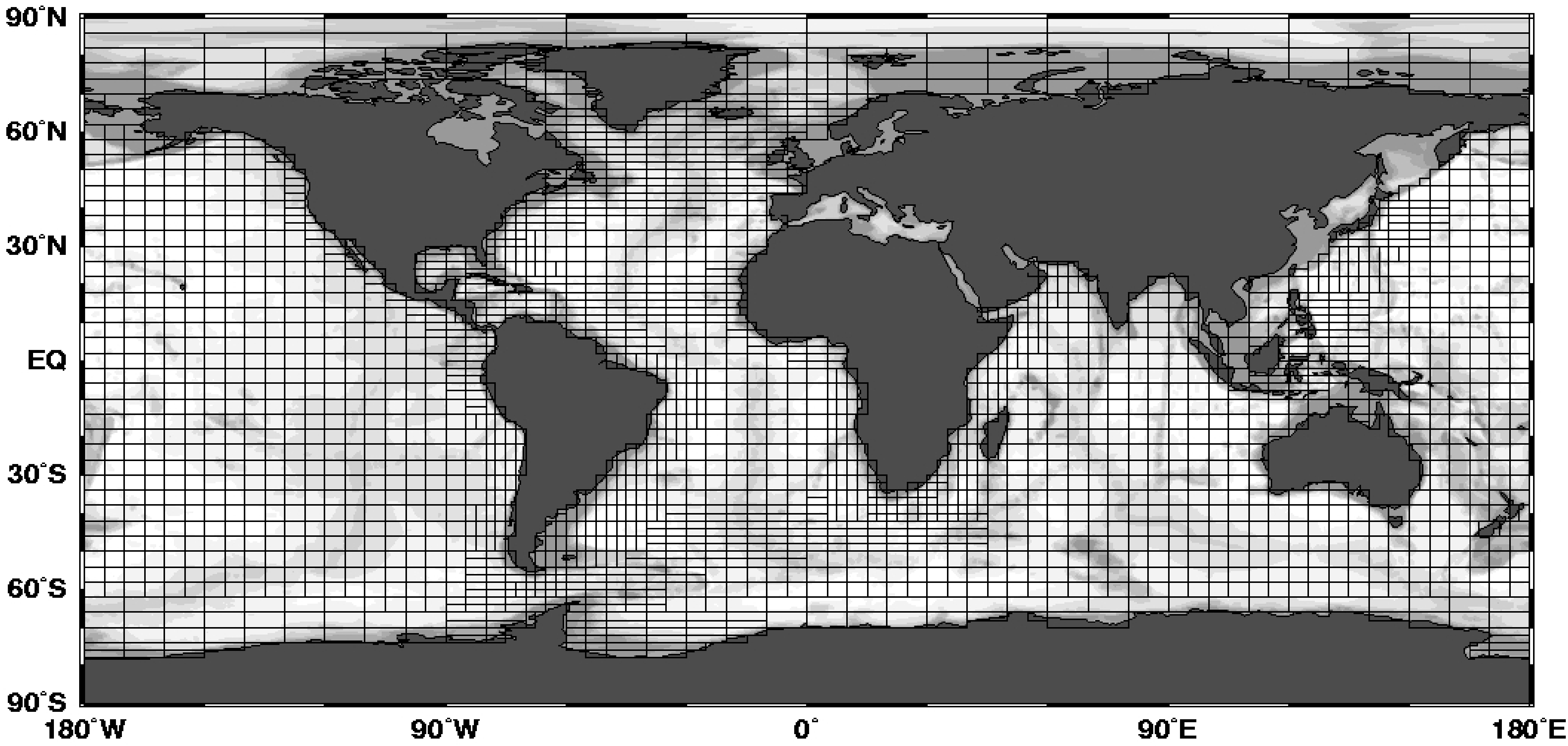
Adjoint Method: Yields Gradient Information!

How to exploit this information? (ii) Conjugate-gradient algorithm



→ Exact knowledge of the gradient may not be that important!

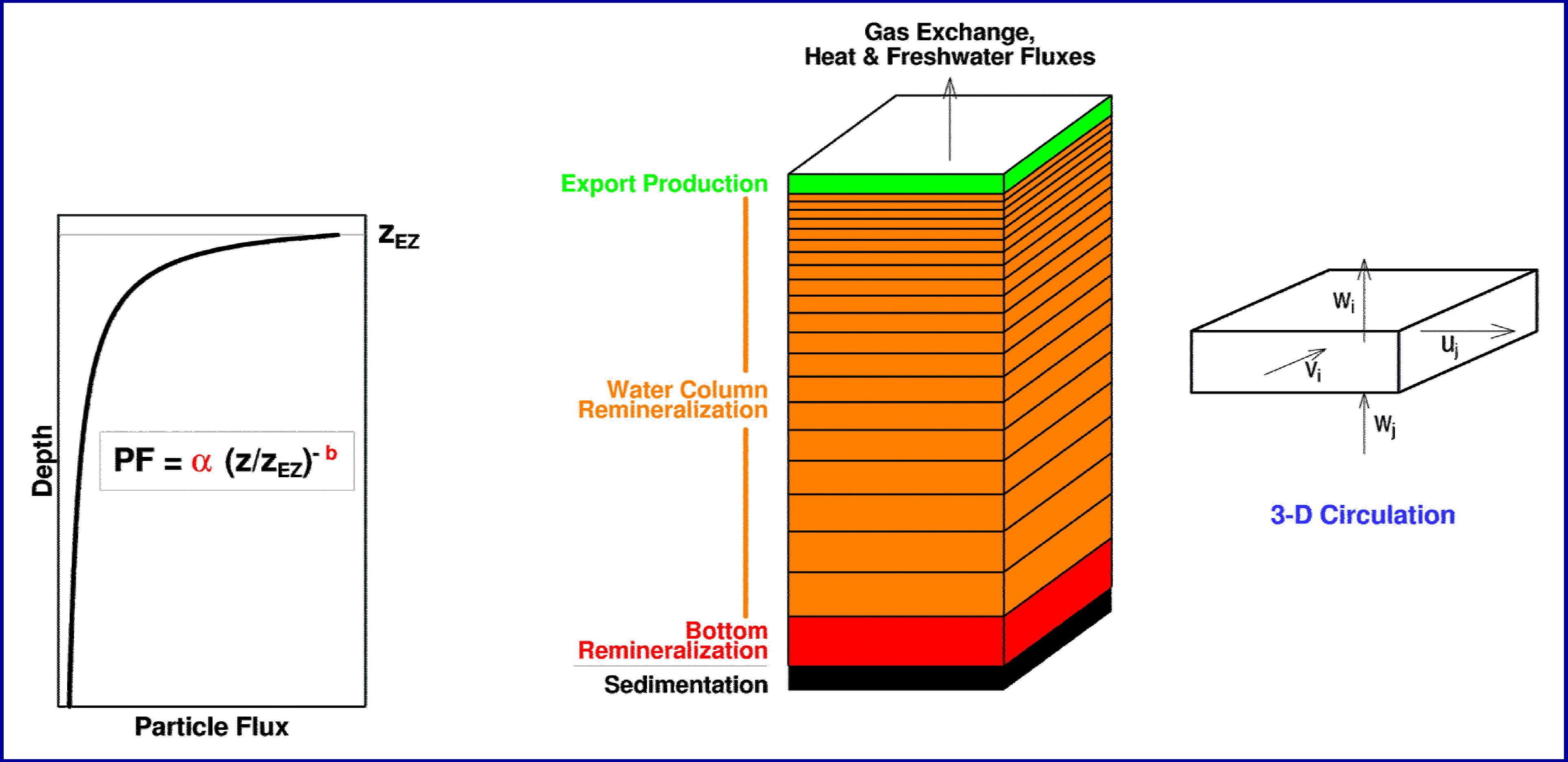
Adjoint Method: Example
Coupled biogeochemical-circulation model
(Schlitzer, 2000)



- 3D model, 26 vertical levels, steady state,
- mass & tracer conservation, close to geostrophy

Coupled biogeochemical-circulation model

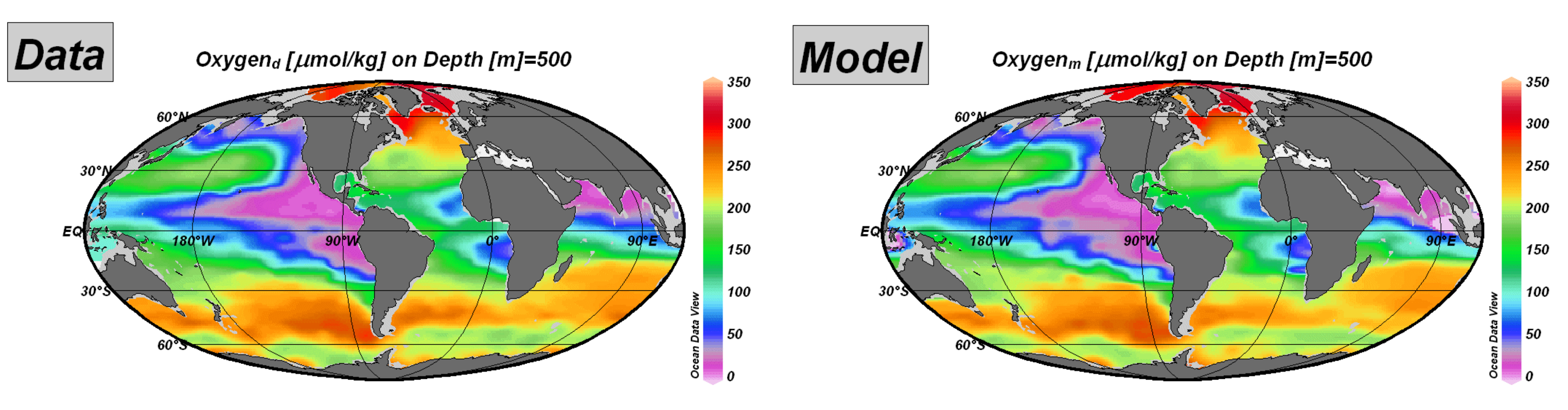
(Schlitzer, 2000)



- 102 306 parameters (80% physical, 20% biogeochemical),
- GEOSECS, WOCE, JGOFS data (> 14 000 profiles),
- steady state assumption

Results

(Schlitzer, 2000)



- Good agreement of modeled tracer fields with observations (although some processes like denitrification, N2-fixation are not yet included!).
- Some violation of physical laws.
- Steady state (e.g., no seasonal cycle)!

What can data assimilation tell us about ecological models?

Complexity of presently used ecosystem models

Ecosystem model	stoichiometry	Number of adjustable parameters
Restoring	usually Redfield	O(1)
NPZD-type	usually Redfield	O(10)
Multiple functional groups, multiple elemental cycles	prognostic	O(100)

- ``Intuitively´´: More complex models are more realistic.

What can data assimilation tell us about ecological models?

Parameter estimation studies (so far NPZD-type only)

(Fasham & Evans, 1995; Matear, 1995; Prunet et al., 1996; Hurtt & Armstrong, 1996/1999; Spitz et al., 1998/2001; Fennel et al., 2001; Schartau et al., 2001; Friedrichs, 2002;....)

- Only 10-15 parameters can be constrained.
  - Lots of unconstrained degrees of freedom. Makes extrapolation to different climate conditions problematic.
  - Are models too complex?
- Model-data fits remain relatively poor.
  - Errors in physical forcing.
  - Are models not complex enough?

- Do we yet have the right model structures?

Ecological Modelling: How can we proceed?

- Model development guided by data assimilation.
  Identify and remove redundancies.
  Add complexity after analysis of residuals.
  - Incubation experiments (sea & lab).
  - Mesocosm experiments.
  - JGOFS time-series sites, satellite data.
  - Paleo data.

  Time & space scale

- Do not disregard alternative model structures
  (e.g., based on size, energy, membrane surfaces, ...)
- Be ambituous! Search for “Kepler‘s Laws“ instead of “Ptolomaic Epicycles“.

The End

Simulated Annealing

Input:

- cost function,
- a priori region in parameter space,
- cooling algorithm.

Advantages:

- robust (arbitrary $J(\mathbf{p})$), model, time stepping, first guess $\mathbf{p}$).

Disadvantages:

- large number of iterations.

Sequential Methods

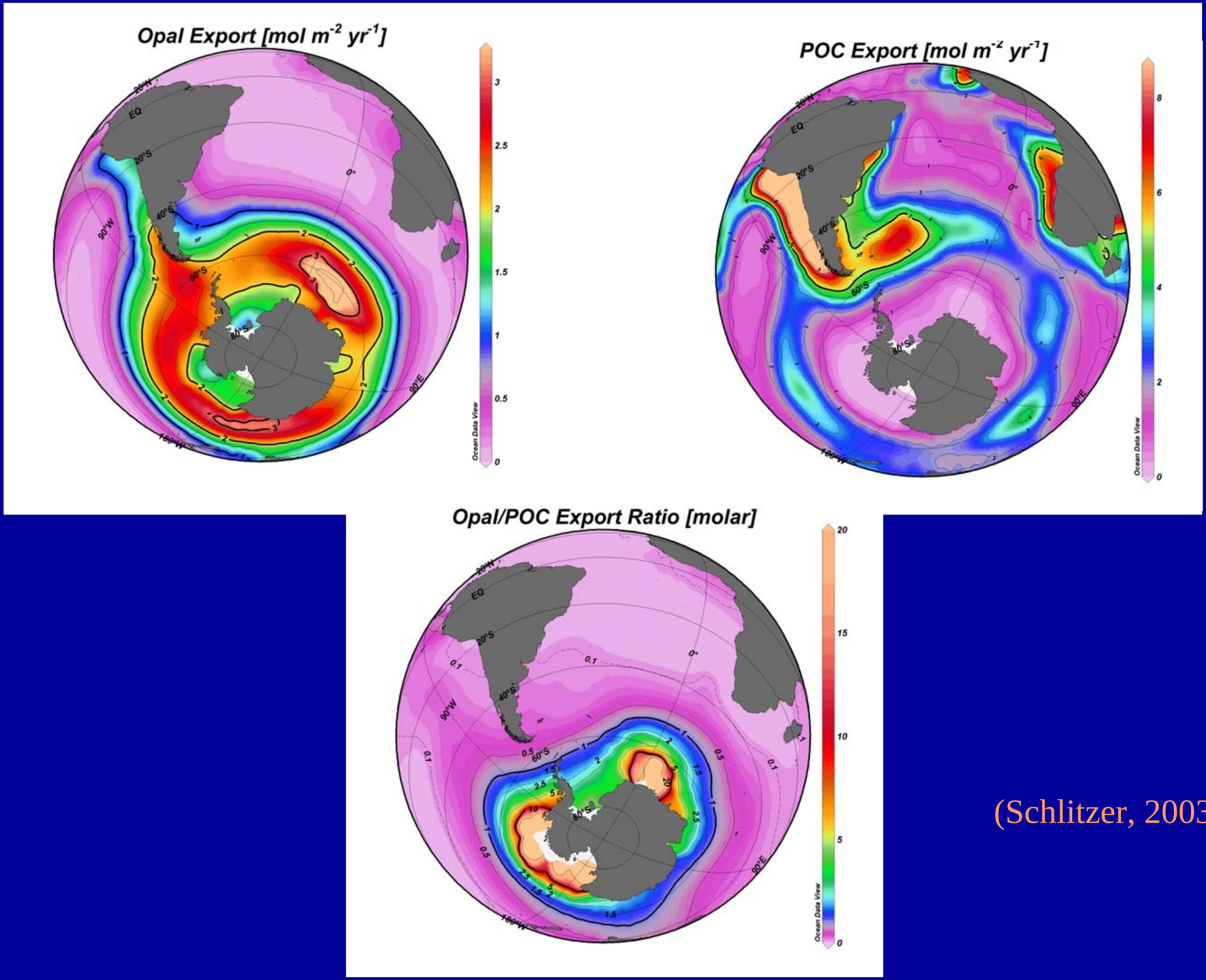
- Make use only of past observations along model trajectory.
- “Accumulate“ information along the model trajectory.
- Aim to improve present state vector.
- Kalman filter generates error covariance matrix of state vector (this is computationally expensive part!).
- Little emphasis on dynamically consistent model trajectory.
- Dynamical interpretation of results often difficult.

- Employed by many operational forecast systems.

Possible SOLAS-related applications:

- Fore-/hindcasting of surface $p\mathrm{CO}_2$, phytoplankton etc.

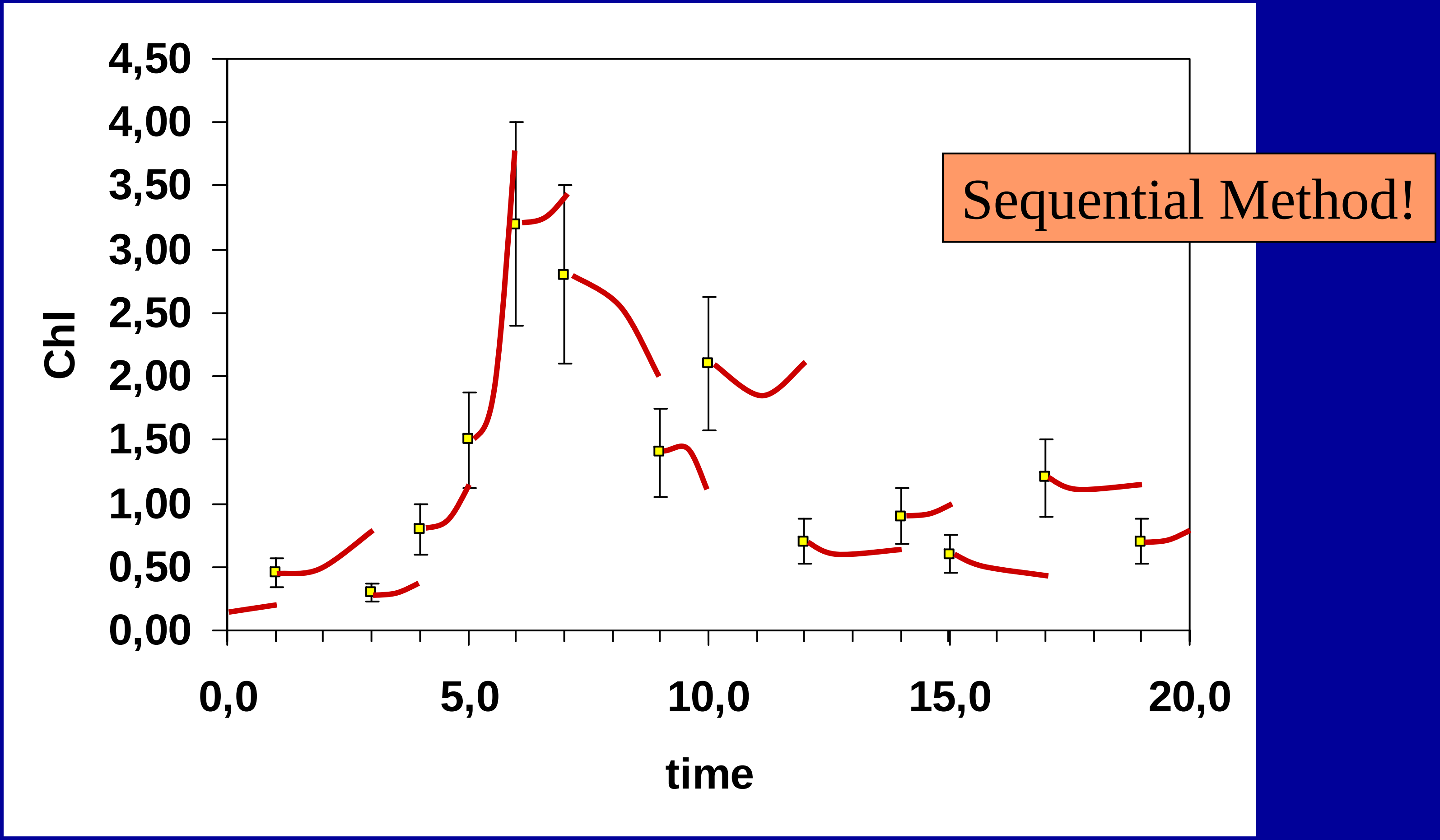
Results



(Schlitzer, 2003)

Data assimilation concepts

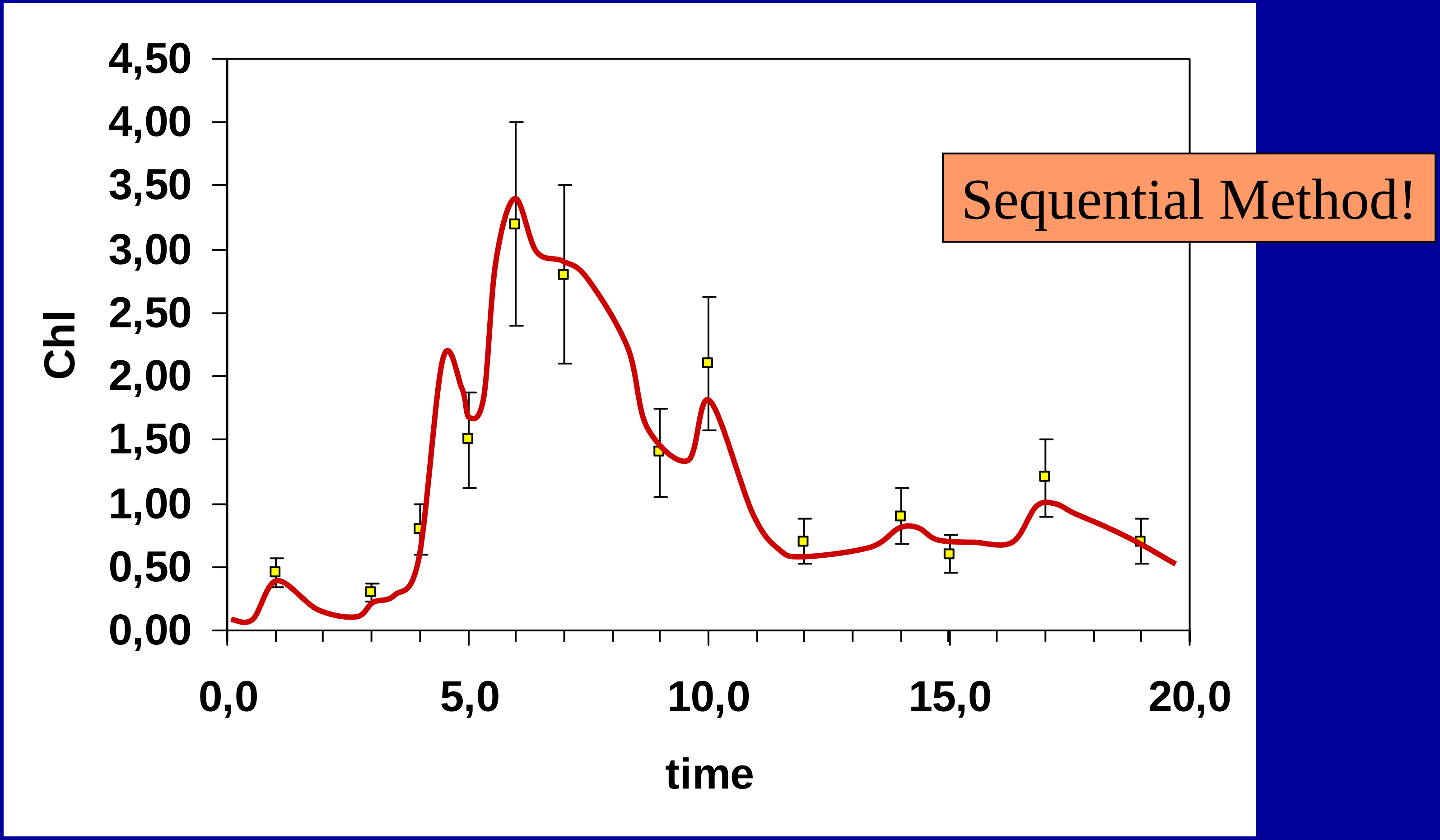
Direct insertion (Ishizaka, 1990)



- Only observed variable is updated, no consistent changes of other state variables!
- No consistent model trajectory, violation of conservation equations (mass!)

Data assimilation concepts

Nudging / Newtonian Relaxation



- Adds unrealistic forcing term to prognostic equations.
- Only observed variable is nudged, no consistent changes of other state variables!
- Perturbation of model dynamics, violation of conservation equations.

On parameters and variables

(Geoff Evans, US JGOFS newsletter, 200x)

Variable:

- Product of all circumstances that created it (e.g., phytoplankton biomass).
- Will vary over time and space.

Parameter:

- Describes the rules of a process (e.g., maximum growth rate).
- Constant in time and space (although different games will be played according to the same rules).

Two distinct objectives

Find “best“ state estimate.

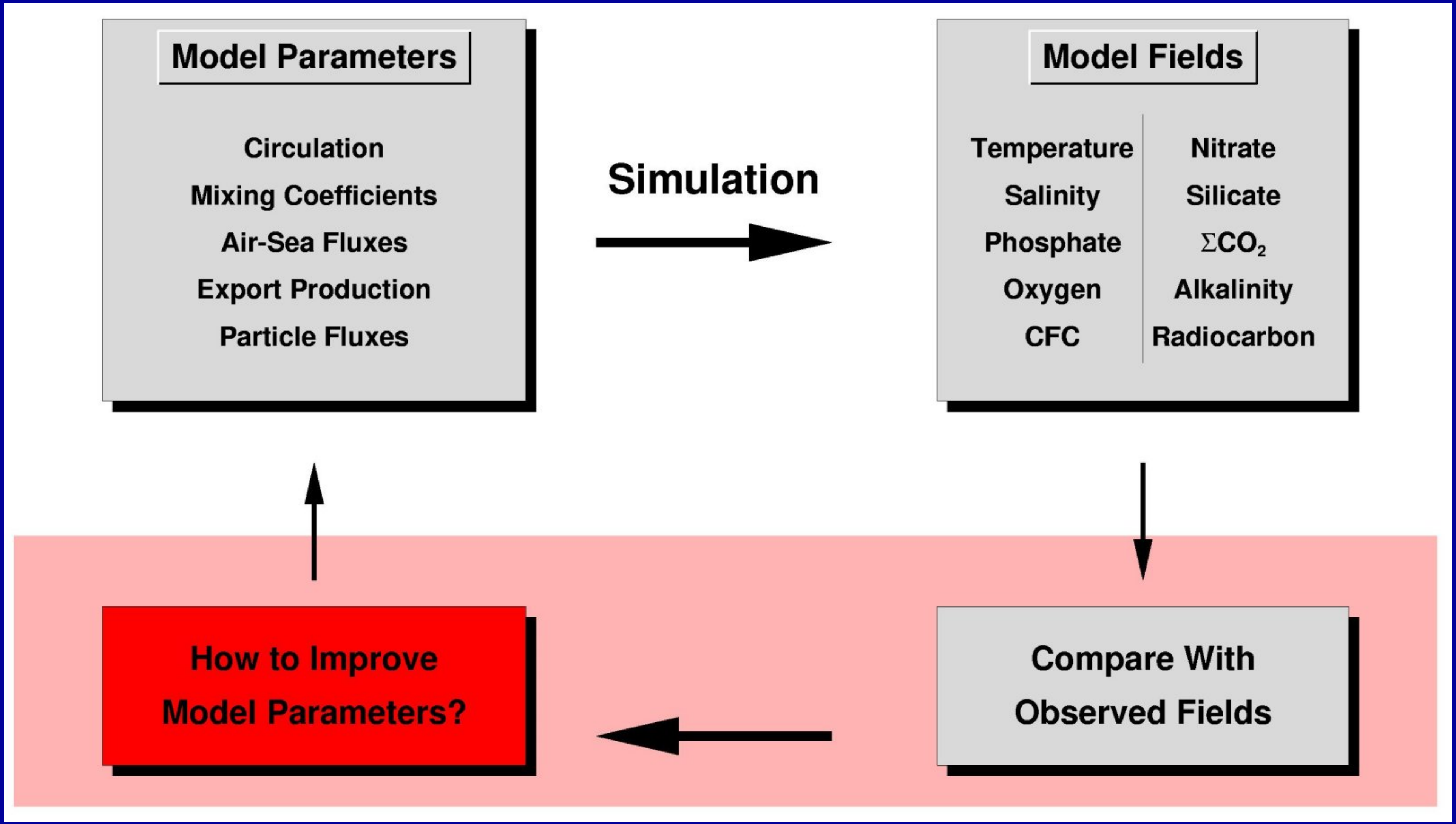
- Minimise expected rms error of 3D state (e.g., weather forecast).
- Tends to suppress variability on scales not observed (smooth climatology has smaller rms error than “noisy“ model state).

Find “best“ model trajectory / dynamical solution.

- Allows for analysis of the underlying dynamics.
- Implies strong confidence in model dynamics.
- Adjust initial conditions, boundary conditions, internal parameters.

Inverse approach

(Schlitzer, 2000)



Sequential Assimilation

- Blending of new observation, x_{obs} , with model forecast, x_f .
- BLUE (best linear unbiased estimate) x_a :

$$x_a = \frac{\frac{x_f}{\sigma_f} + \frac{x_{obs}}{\sigma_{obs}}}{\frac{1}{\sigma_f} + \frac{1}{\sigma_{obs}}} \quad , \qquad \sigma_a = \frac{1}{\frac{1}{\sigma_f} + \frac{1}{\sigma_{obs}}}$$

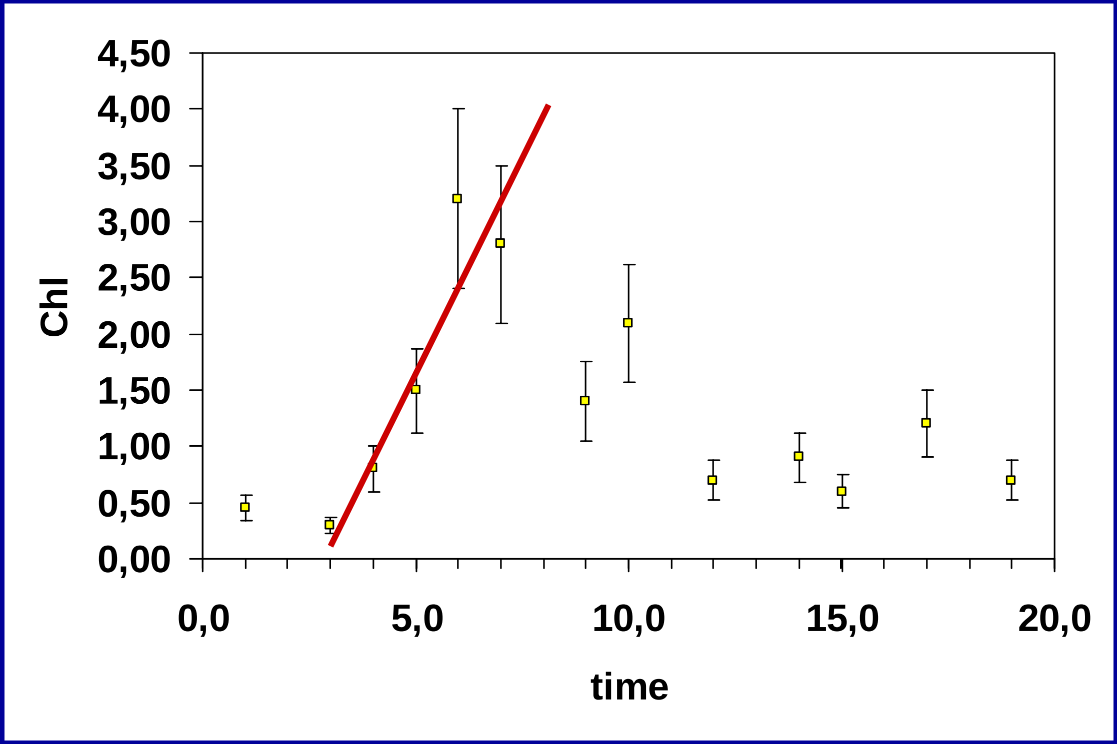
- Kalman filter computes temporal evolution of both state vector x and error covariance matrix σ :

$$\mathbf{x}_{n+1} = \mathbf{A}\,\mathbf{x}_n$$

$$\sigma^2_{n+1} = \mathbf{A}\,\sigma^2_n\,\mathbf{A}^t + \mathbf{e}^2_n$$

Adjoint Method

Example:
linear spring bloom model
(simple but wrong!)



Model as differential equation:

$$\frac{d^2 m}{dt^2} = 0$$

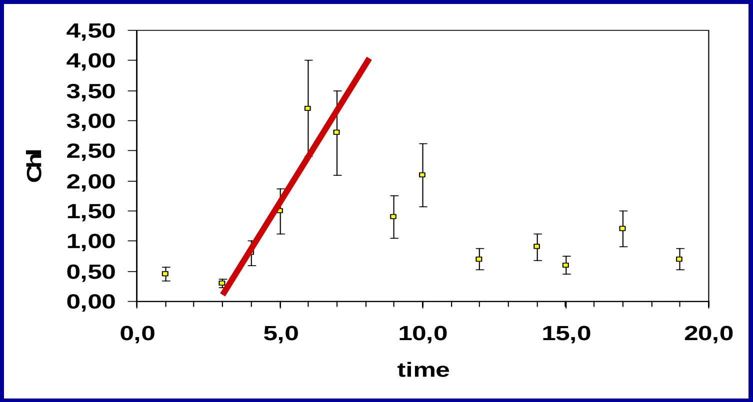
discretized: $m_{j+1} - 2m_j + m_{j-1} = 0$, with time step index j .

$\Rightarrow$ $j_{max} - 1$ equations, $j_{max} + 1$ unknowns.

$\Rightarrow$ 2 independent variables (i.e., parameters!), e.g., m_0 , m_1 .

Adjoint Method

Example:
linear spring bloom model



Cost function:

$$J = \frac{1}{2\sigma^2} \sum_{\nu} (m_\nu - d_\nu)^2$$

constrained minimisation, 2 equations, j_{max}+1 unknowns

$$\left(\frac{\partial J}{\partial m_0}\right)_{model} = \frac{1}{\sigma^2} \sum_{\nu} (m_\nu - d_\nu) \left(\frac{\partial m_\nu(m_0, m_1)}{\partial m_0}\right)_{model}$$

$$\left(\frac{\partial J}{\partial m_1}\right)_{model} = \frac{1}{\sigma^2} \sum_{\nu} (m_\nu - d_\nu) \left(\frac{\partial m_\nu(m_0, m_1)}{\partial m_1}\right)_{model}$$

➔ either make extensive use of chain rule or introduce Lagrangian multipliers!

Adjoint Method: Example

Lagrange function: $L = \frac{1}{2\sigma^2}\sum_\nu (m_\nu - d_\nu)^2 + \sum_{j=1}^{j_{max}-1} \lambda_j \left(m_{j+1} - 2\,m_j + m_{j-1}\right)$

1.) solve for $m_{j_{max}+1}$

$$\frac{\partial L}{\partial \lambda_j} = m_{j+1} - 2\,m_j + m_{j-1} = 0, \quad {\scriptstyle j=1,\ldots,j_{max}-1}$$

"forward" model

6.) solve for L_{m_1}

$$\frac{\partial L}{\partial m_0} = \frac{1}{\sigma^2}(m_\nu - d_\nu)\delta_{\nu 0} + \lambda_1 \neq 0$$

5.) solve for L_{m_1}

$$\frac{\partial L}{\partial m_1} = \frac{1}{\sigma^2}(m_\nu - d_\nu)\delta_{\nu 1} + \lambda_2 - 2\lambda_1 \neq 0$$

$\mathrm{grad}_p J$

4.) solve for λ_1

$$\frac{\partial L}{\partial m_j} = \frac{1}{\sigma^2}(m_\nu - d_\nu)\delta_{\nu j} + \lambda_{j+1} - 2\lambda_j + \lambda_{j-1} = 0$$

"adjoint" model

3.) solve for $\lambda_{j_{max}-2}$

$$\frac{\partial L}{\partial m_{j_{max}-1}} = \frac{1}{\sigma^2}(m_\nu - d_\nu)\delta_{\nu j_{max}-1} - 2\lambda_{j_{max}-1} + \lambda_{j_{max}-2} = 0$$

2.) solve for $\lambda_{j_{max}-1}$

$$\frac{\partial L}{\partial m_{j_{max}}} = \frac{1}{\sigma^2}(m_\nu - d_\nu)\delta_{\nu j_{max}} + \lambda_{j_{max}-1} = 0$$

How to define misfit?

- Different dimensions (rates, concentrations,…)
  - normalise by scale factor S_i: $(d_i - m_i)/S_i$

- Size of S_i? Standard deviation of observation error (if known…)?
  - often: $S_i = d_i$, $S_i = m_i$, $S_i = <d>$, $S_i = d_{max}$,… Implications?

- Asymmetry between positive and negative misfits?

- Different numbers of observations for different data types (e.g., surface chlorophyll, zooplankton grazing rate)
  - serial correlations (weight by 1/N)?
  - cross correlations ?