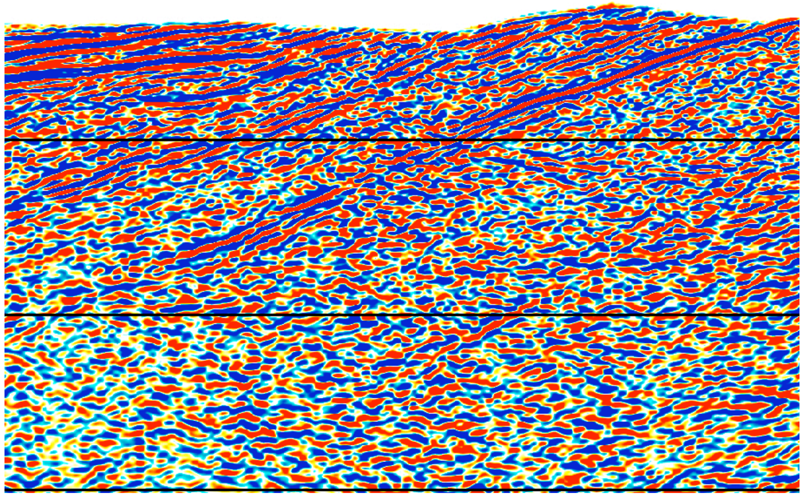
Metaheuristics in science and engineering

Klaus Mosegaard

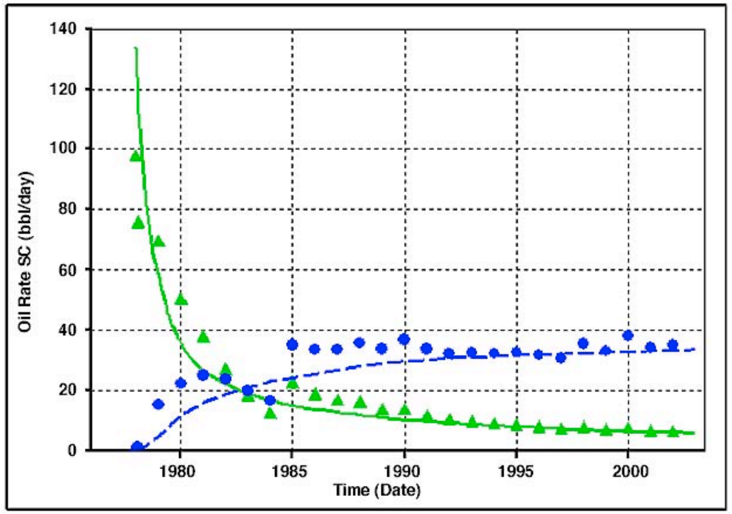
Niels Bohr Institute
University of Copenhagen

Example: A computational challenge in applied geoscience

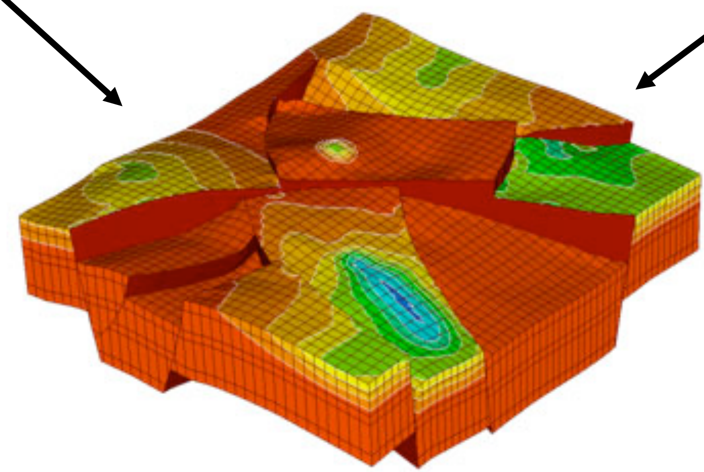
Seismic data



Oil production data



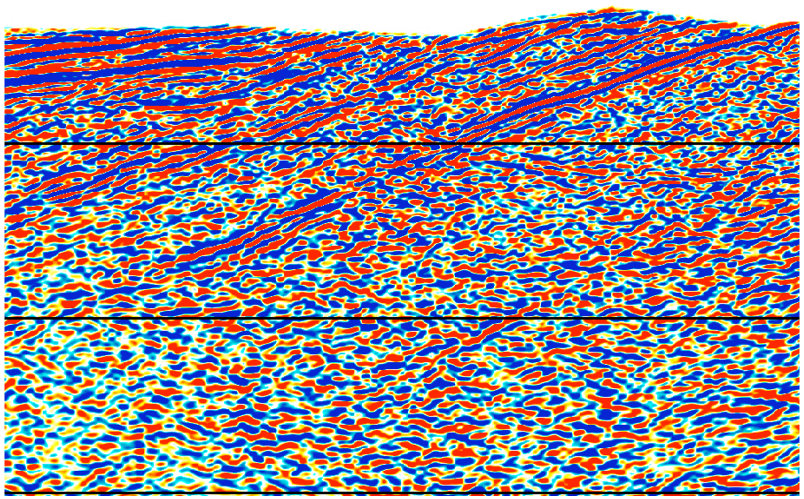
Seismic inversion



History matching

Seismic Inversion

Seismic data



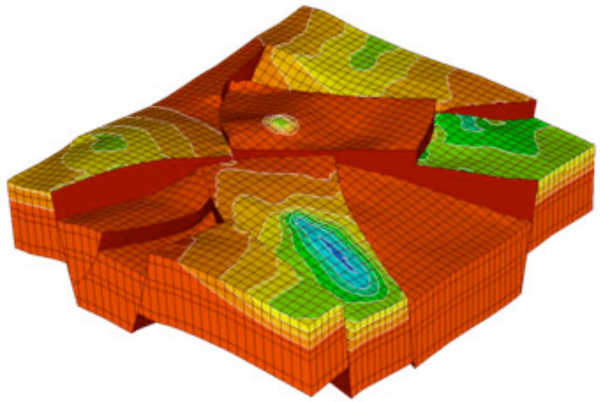
Optimization of data fit

Numerical solution of wave equations

$$\frac{\partial^2\theta}{\partial t^2} = \frac{\lambda + 2\mu}{\rho}\nabla^2\theta$$

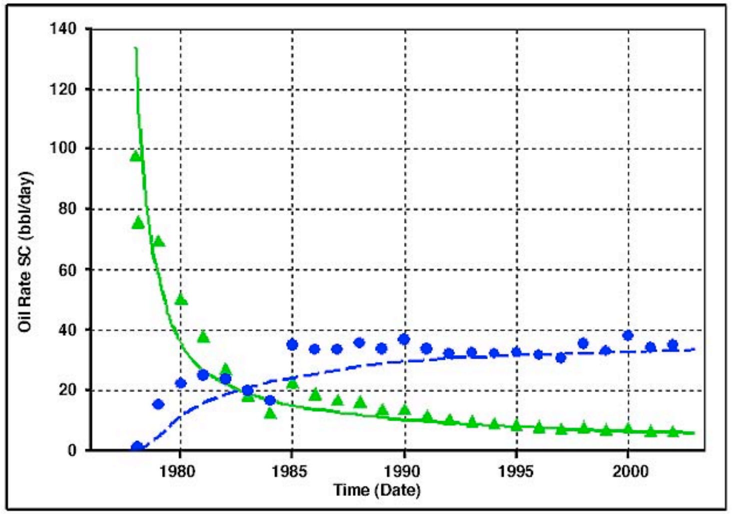
$$\frac{\partial^2(\nabla \times u_i)}{\partial t^2} = \frac{\mu}{\rho}\nabla^2(\nabla \times u_i)$$

Impedance/Porosity model



History Matching

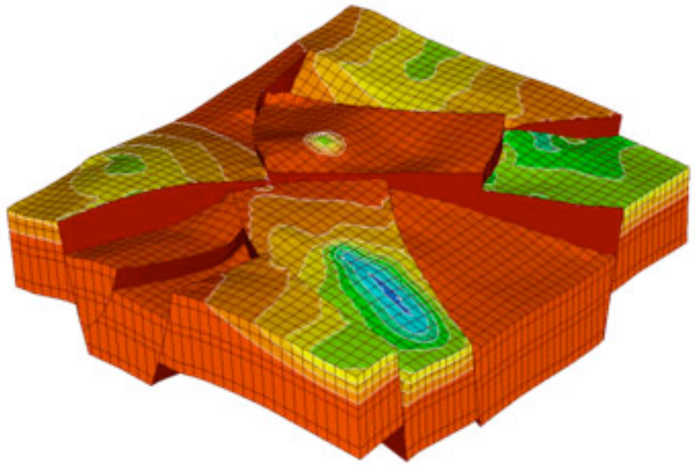
Production data



Optimization of data fit

Numerical solution of flow equations

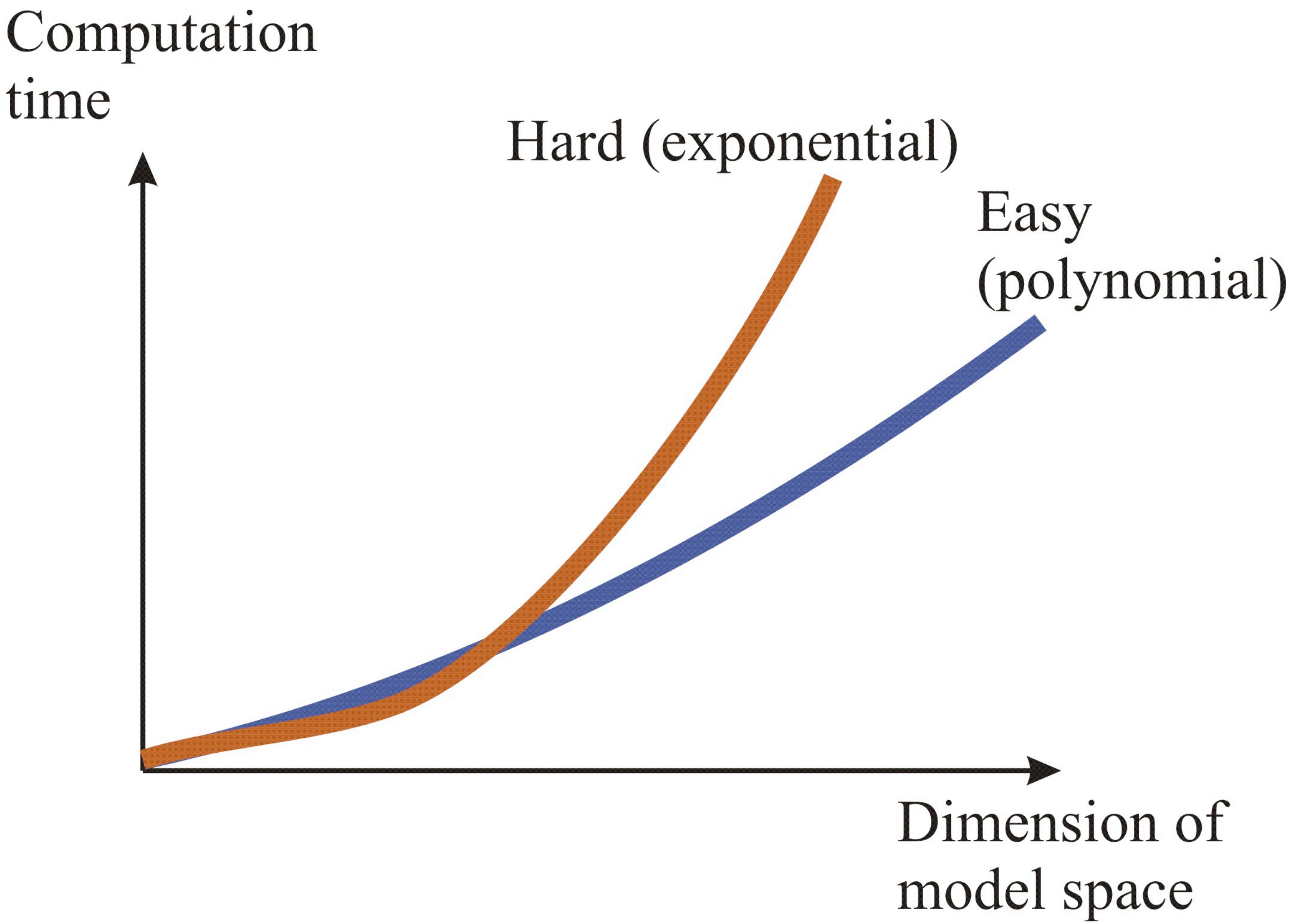
Porosity, permeabillity/ saturation model



$$\frac{\partial}{\partial t}\left[\varphi \rho_w S_w\right] - \nabla \cdot \left[\rho_w \frac{\mathbf{k} k_{rw}}{\mu_w}\left(\nabla P - \rho_w g \nabla Z\right)\right] + Q_w = 0$$

$$\frac{\partial}{\partial t}\left[\varphi \rho_o S_o\right] - \nabla \cdot \left[\rho_o \frac{\mathbf{k} k_{ro}}{\mu_o}\left(\nabla P - \rho_o g \nabla Z\right)\right] + Q_o = 0$$

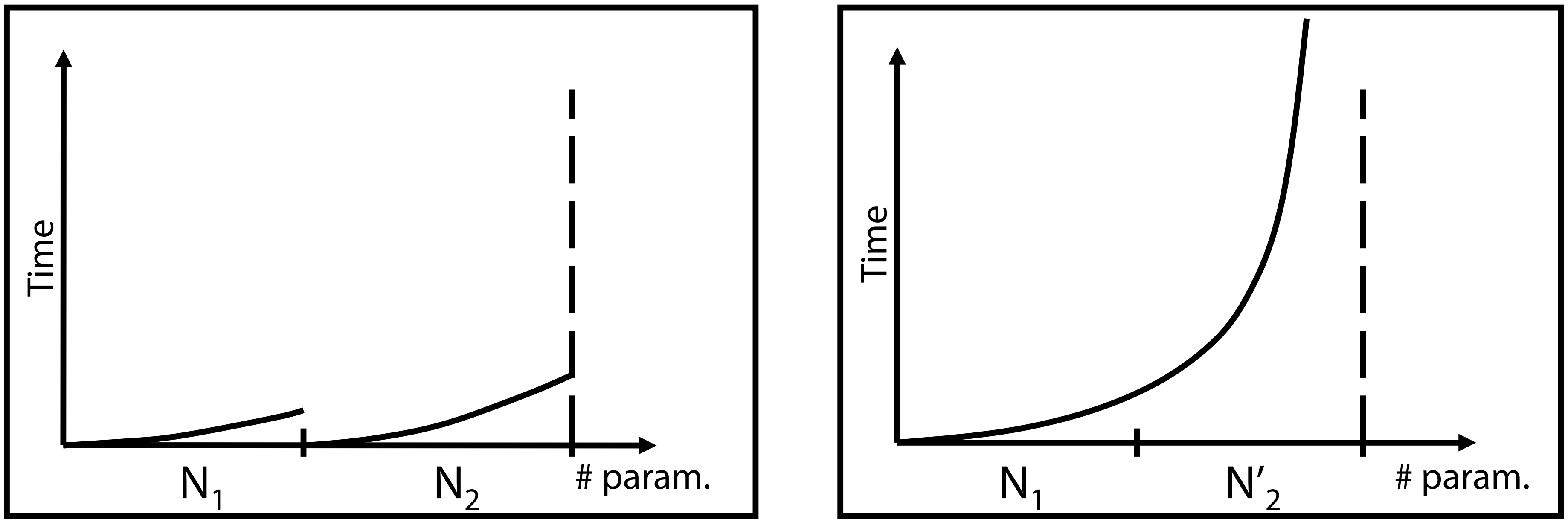
Computation time vs. number of unknowns



Combining Seismic Inversion and History Matching

Separate calculations
Acceptable computation times add up

Combined calculation
Computation time explodes!



Plan for the talk

We will investigate limitations for algorithms used in large-scale data fitting problems when

1. problem independent algorithms (metaheuristics) are used
2. the algorithms are adapted to special properties of the problem

We will propose a way of avoiding hard problems

Overall conclusions of the talk

Hard data-fitting problems: How do we make progress?

<u>Waiting for faster computers?</u>
No. Faster computers will not significantly improve our ability to solve hard problems.

<u>Waiting for smart algorithms (metaheuristics)?</u>
No – there are fundamental limitations to the performance of any of these algorithms.

<u>Using more problem-specific algorithms?</u>
Yes – the more problem specific, the better.

Formulation of the data fitting (inversion) problem

Data analysis in Science and engineering usually involves solution of an **inverse problem**, which again means **solving a set of equations**.

- Each datum provides a constraint in form of one equation

$$d_i = g_i(\mathbf{m}).$$

- We seek to find an approximate, simultaneous solution to all these equations by minimizing a **misfit function** like

$$S(\mathbf{m}) = \|\mathbf{d} - \mathbf{g}(\mathbf{m})\|^2.$$

  or maximizing a **fit function** like

$$f(\mathbf{m}) = \exp\left(-\frac{1}{2}\|\mathbf{d} - \mathbf{g}(\mathbf{m})\|^2\right).$$

  with respect to $\mathbf{m}$.

Minimizing a misfit function: can we measure the performance of the many available algorithms?

- Steepest descent
- Simulated Annealing
- Genetic Algorithm
- Taboo Search
- Neighbourhood Algorithm
- ...

To answer this question, we need to describe exactly the conditions we work under.

The blind search scenario

- In this scenario we have **no closed-form** mathematical expression for the right-hand-side of

$$d_i = g_i(\mathbf{m}).$$

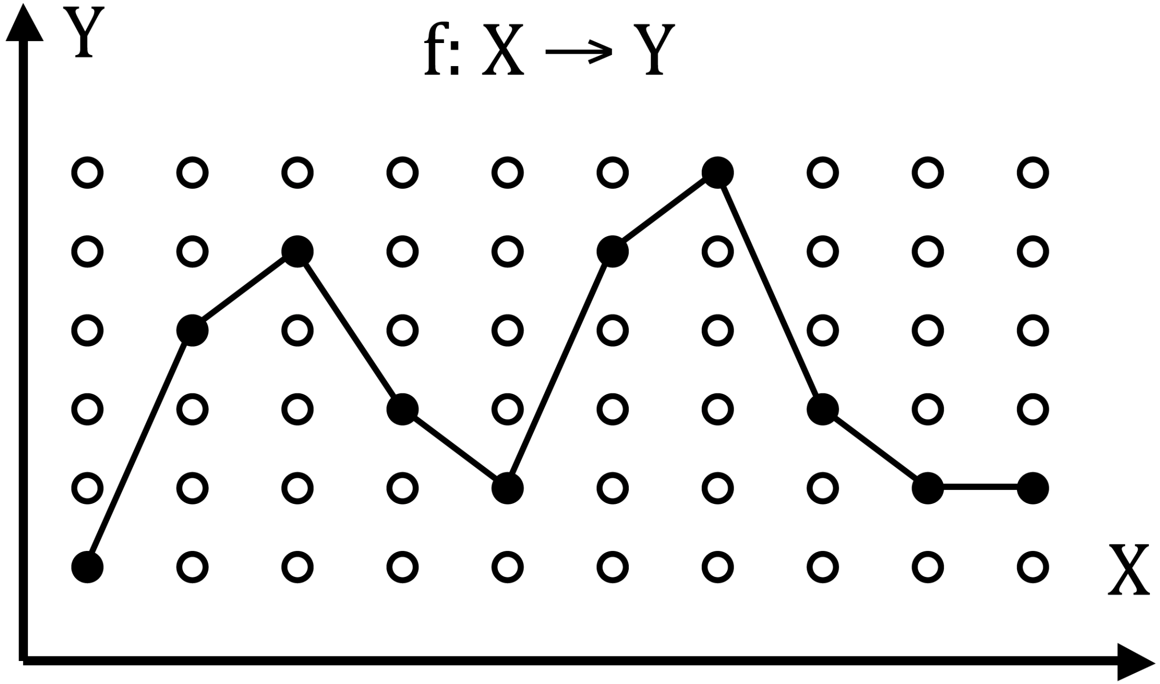
- We only have a programme that is able to **evaluate** $g_i(\mathbf{m})$ **for given values** of the parameters in $\mathbf{m}$.

In short:

$\boxed{\text{We are performing a } \textbf{blind search} \text{ for the solution.}}$

Notation (in a fully discrete formulation)

- Two *finite sets* X and Y,
- The set $\mathcal{F}_X$ of all fit functions/probability distributions $f : X \to Y$.

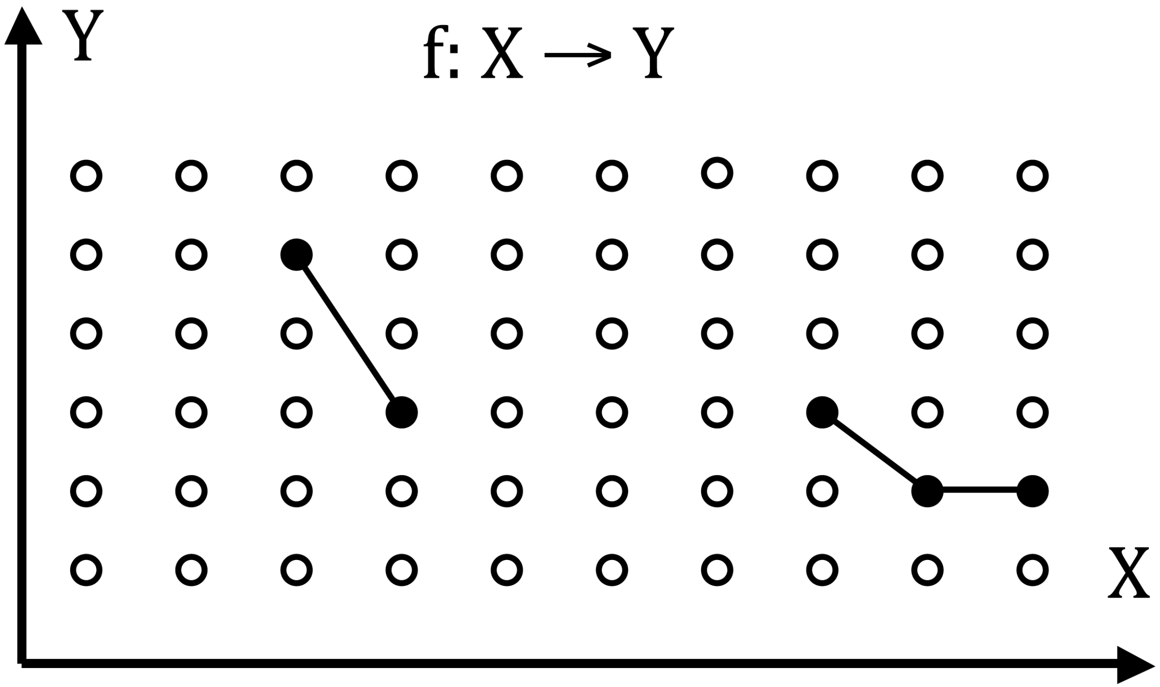


Notation

- A sample of size $m < |X|$:

$$\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}.$$

- The set $\mathcal{F}_{X|C}$ of all fit functions/probability distributions defined on X, but with fixed values in C.

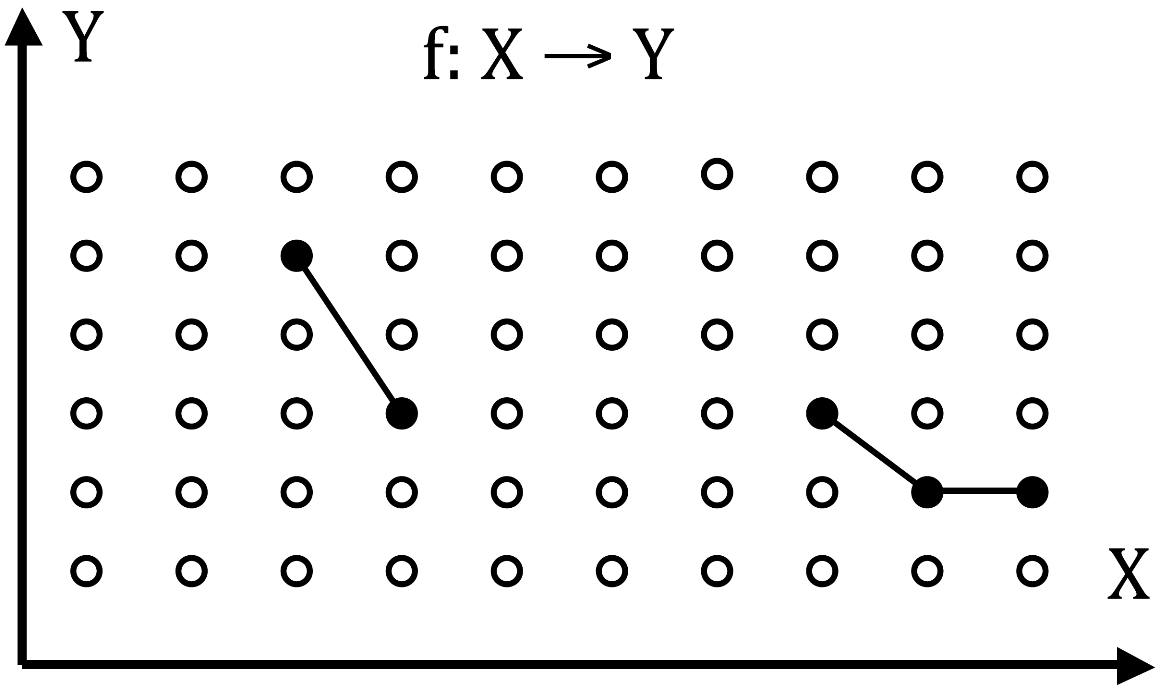


Lemma

The total number of functions intersecting the m samples is

$$|\mathcal{F}_{X|C}| = |Y|^{|X|-m}. \tag{1}$$

This number is independent of the location of the sample points.



No-Free-Lunch Theorem (Wolpert and Macready, 1995)

Theorem

NFL (Wolpert and Macready, 1995) *For* $f \in \mathcal{F}_X$ *and any pair of algorithms* a_1 *and* a_2*,*

$$P(y_1, \ldots, y_m | f, m, a_1) = P(y_1, \ldots, y_m | f, m, a_2) \tag{2}$$

where $P(\cdot|\cdot)$ *denotes conditional probability.*

Since any performance measure for inversion is a function of form $\Phi : Y^m \to \mathbb{R}$, for instance:

$$\Phi(y_1, \ldots, y_m) = \max\{y_1, \ldots, y_m\}, \tag{3}$$

which must be large for good performance, the NFL Theorem implies the following:

Corollary

(NFL for optimization) *The distribution of any performance measure for inversion, when all fit functions are equally probable (blind inversion), is exactly the same for all inversion algorithms.*

A serious objection to the NFL theorem

Postulate
"The No-Free-Lunch theorem essentially assumes that the fit functions is *white noise*, and this is unrealistic".

So, in real cases the situation is different from the NFL-scenario:

$\boxed{\text{We have a narrow set of (e.g. smooth) fit functions.}}$

This objection is based on the reasoning that the total number of ways a particular set of fit values $y_1, \ldots, y_m$ can be obtained in *a particular* set of m sample points $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ will in general depend on the location of these points.

Consider the case where f is 'noise of unknown color'

Averaging over all possible fit function families ('colors') gives:

Lemma

Consider all possible subfamilies of functions with fixed values on the sampled subset $C \in X$. The total number of functions in all these families (counting some functions more than once) depends only on the number m of elements in C, and not on the elements themselves.

Since the search algorithm only manifests itself through the selection of elements in C, we have

Corollary

The efficiency of all blind inversion schemes are exactly the same. *The expected performance over all 'colored' fit function families is the same for all inversion algorithms.*

Proof. (Similar to the original NFL theorem) ■

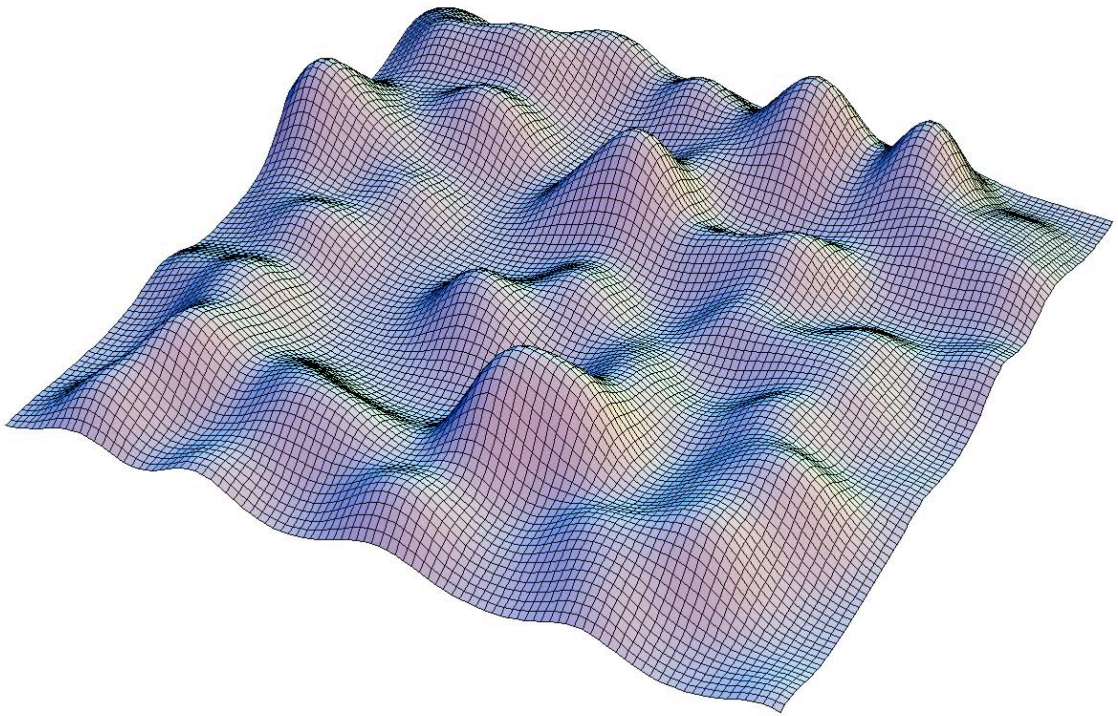
Conclusion

The efficiency of all blind inversion schemes:

- Simulated Annealing,
- Metropolis Algorithm,
- Genetic Algorithm,
- Taboo Search,
- Neighbourhood Algorithm,
- ...

when averaged over alle possible classes of inverse problems, are exactly the same.

Consider the case where certain properties of the fit functions is known by the algorithm



Assume that *f* can be expanded with respect to a set of **base functions**, centered in a grid of points

Here, the number of parameters needed to characterize *f* **grows exponentially with *N***

Example:

$$f(\mathbf{x};\sigma,r_1,\ldots,r_K) = \sum_{k=1}^{K} r_k \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_k\|^2}{2\,\sigma^2}\right)$$

The chain of reasoning

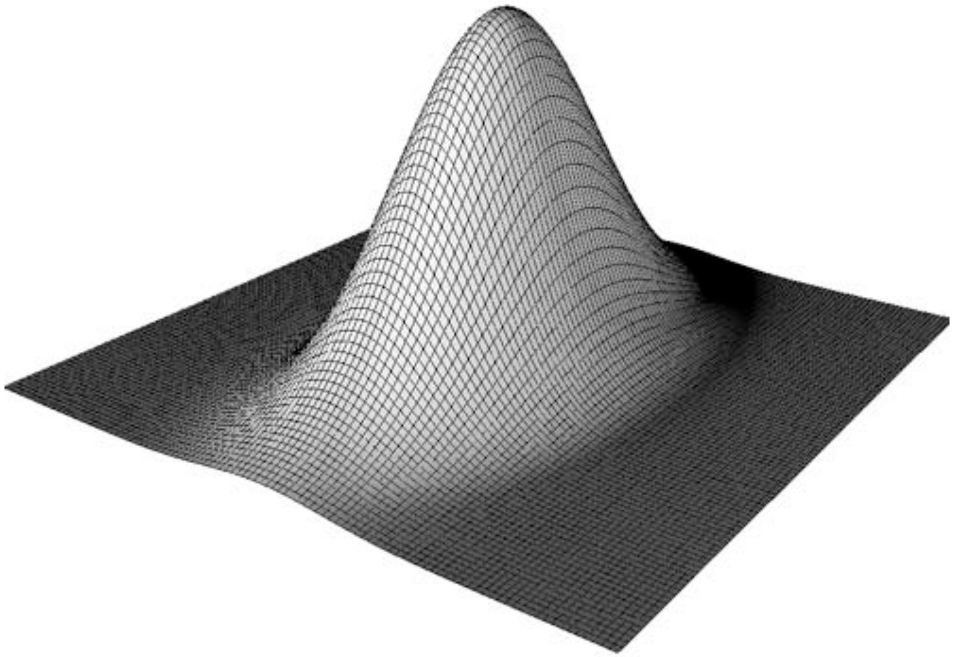
The number of parameters K_N needed to characterize a smooth fit function in an N-dimensional space grows exponentially with N.

Using Brouwer's theorem on *Invariance of dimension* (Brouwer ~1910), we can conclude that at least K_N function evaluations are needed to locate an extremum.

Consequently, the solution time for the best conceivable algorithm grows exponentially with N.

Some algorithms are better than others, but all algorithms will have exponential-time complexity

Consider the case where the fit function is known to be a Gaussian of unknown shape



A Gaussian over an *N*-dimensional space *M* is characterized by the *N* components of its mean vector, and the $N\,(N+1)/2$ components of its covariance matrix.

The family of Gaussians over an *N*-dimensional space is a manifold of dimension

$$N + N\,(N+1)/2$$

At least $N + N(N+1)/2$ function evaluations are required to characterize (“reconstruct”) an N-dimensional Gaussian.

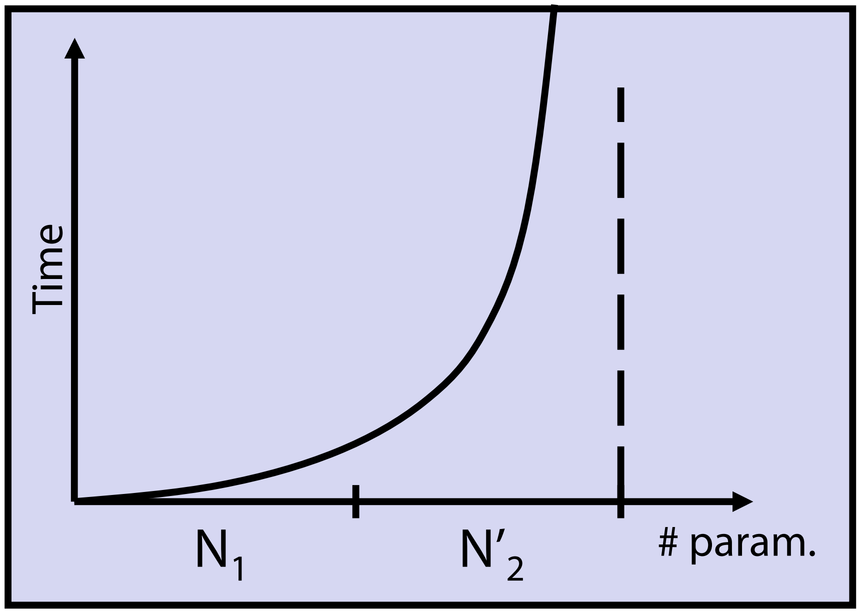
Consequently, the best conceivable algorithm needs $N + N(N+1)/2$ function evaluations to locate the maximum of an N-dimensional Gaussian!

The problem is polynomial (easy).

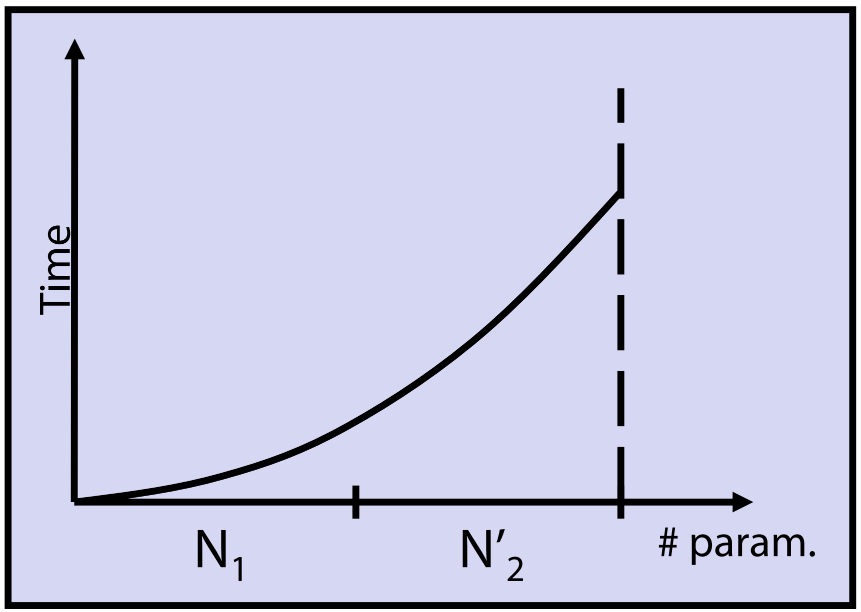
The conclusion to our analysis is: Problem-specific algorithms are needed

For instance: Replace hard calculations with easy (but approximate) calculations!

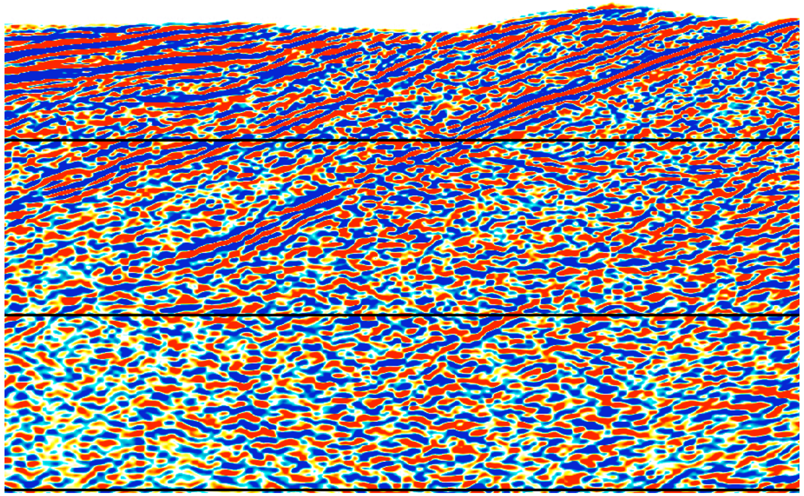
Combined geoscientific calculation:
Computation time explode!



`Faking' one of the solutions:
Computation times under control!

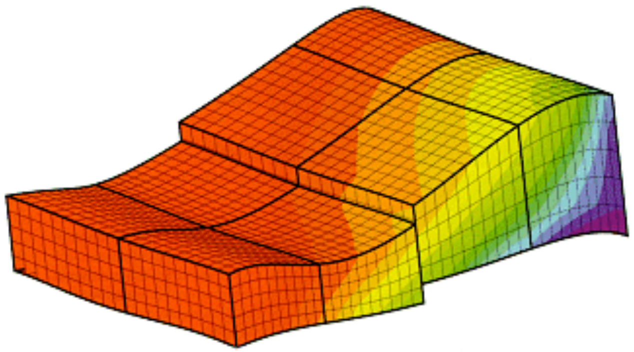


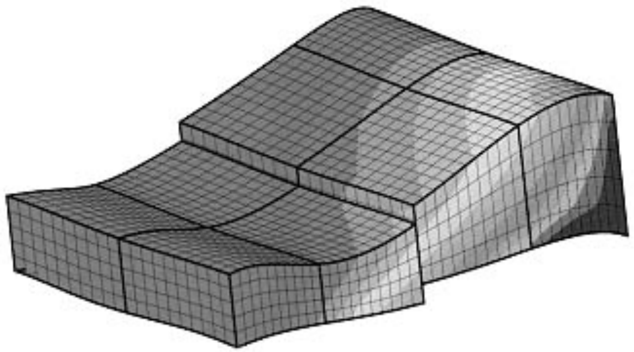
Replacing a hard problem with an easy one



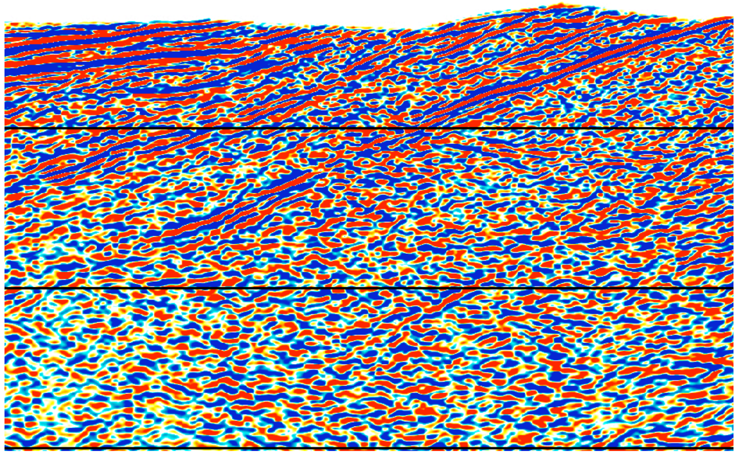
Seismic inversion
(hard, accurate)

Seismic pseudo-inversion
(easy, approximate)



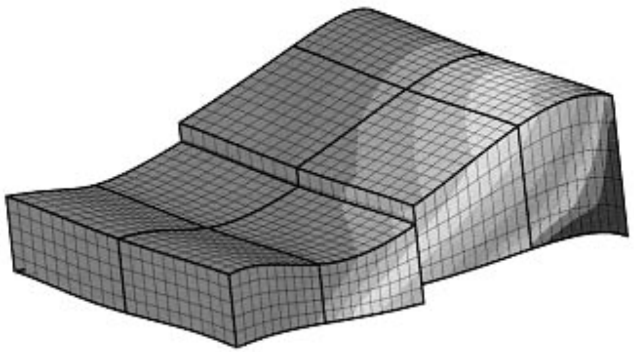


Learning/Non-linear Regression



Seismic data $\mathbf{s}(x,y,t)$

$$\mathcal{F} \downarrow$$



Crude reservoir model $\mathbf{m}(x,y,z)$

$$\mathbf{m} = \mathcal{F}(\mathbf{s})$$

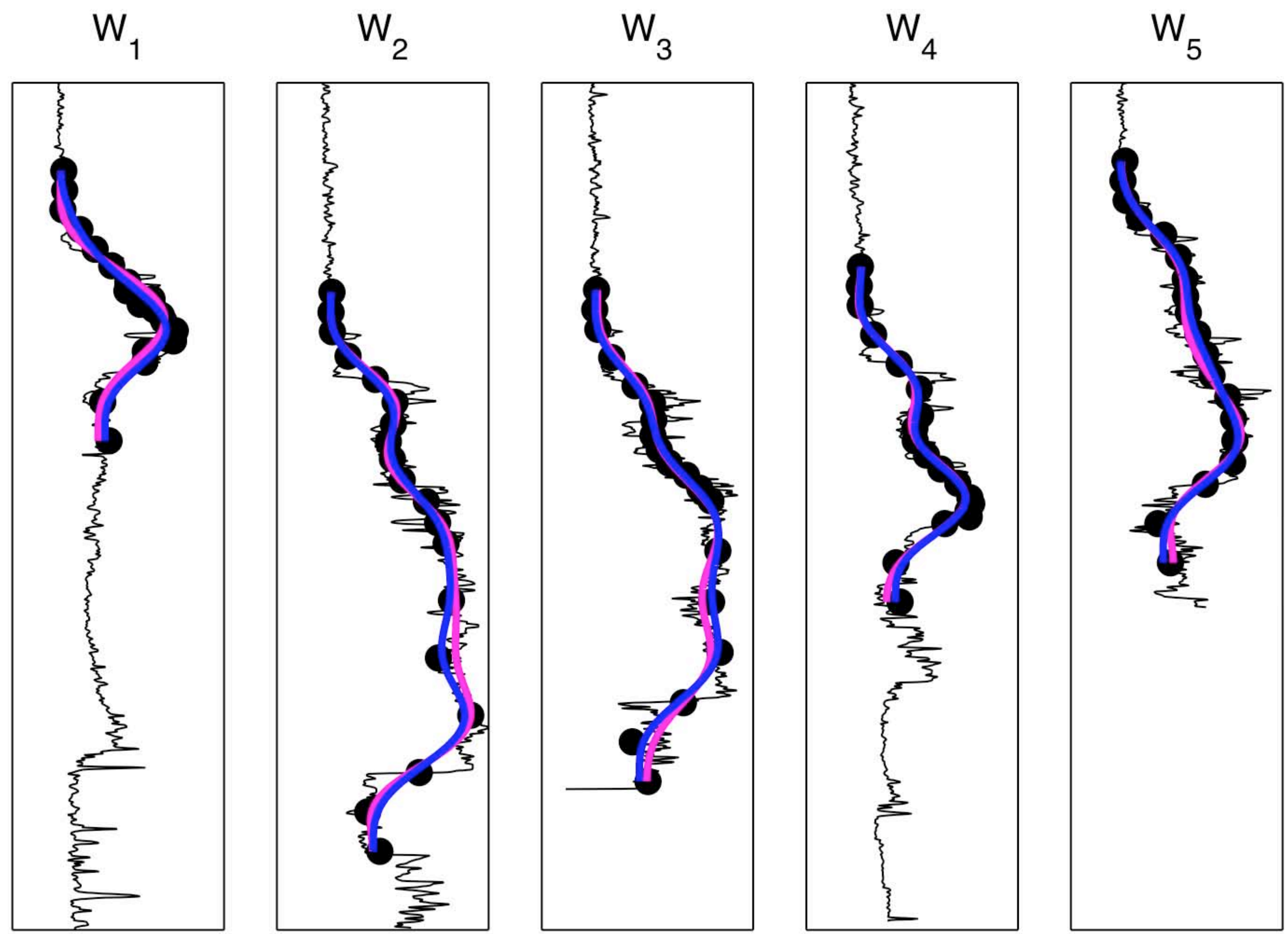
If we know $\mathcal{F}$ at the N well sites:

$$\mathbf{m}(x_1,y_1,z) = \mathcal{F}(\mathbf{s}(x_1,y_1,t))$$
$$\mathbf{m}(x_2,y_2,z) = \mathcal{F}(\mathbf{s}(x_2,y_2,t))$$
$$\ldots$$
$$\mathbf{m}(x_N,y_N,z) = \mathcal{F}(\mathbf{s}(x_N,y_N,t))$$

we may be able to compute $\mathcal{F}$
and predict **m** from **s** everywhere

Example: Seismic Pseudo-inversion

Porosity predicted from distant well data and a few seismic attributes using non-linear regression



From: Hansen et al., 2008 and Pedersen-Tatalovic et al., 2008

Conclusions

1. Data analysis in engineering and science is often an overwhelming computational challenge

2. Faster computers and smarter meta-heuristics will not significantly improve our ability to solve such problems

3. More problem-specific algorithms will be necessary. A way forward may be to use fast, approximate methods for most of the work.