UC SANTA BARBARA
engineering

1495
UNIVERSITY OF ABERDEEN

Inverse modeling of sediment deposition from turbidity currents

Lutz Lesshafft (UCSB)
Eckart Meiburg (UCSB)
Ben Kneller (U of Aberdeen)
Alison Marsden (UCSD)

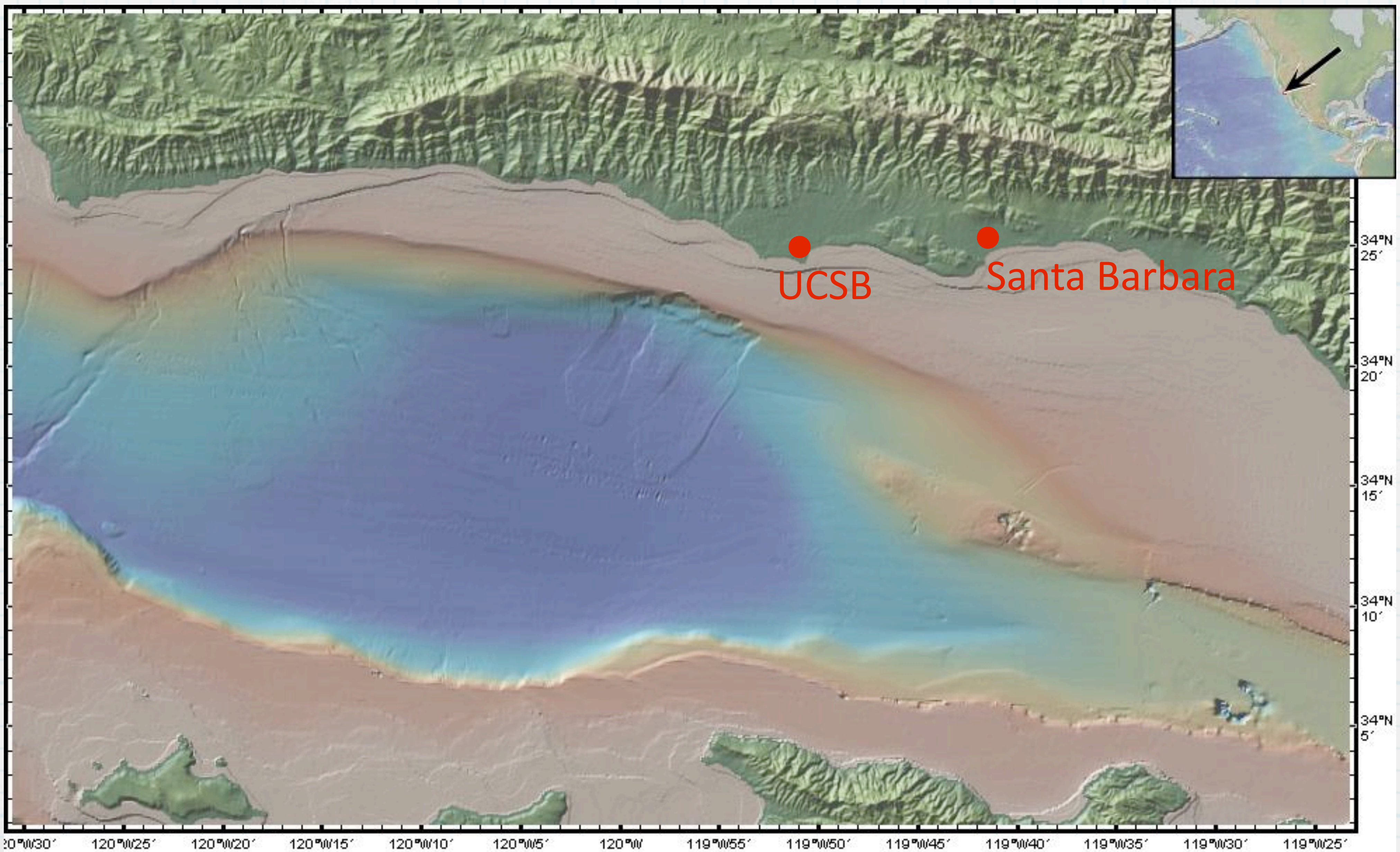
Gravity current workshop UCSB, June 2009

Outline

1. Intro: turbidite systems
2. DNS of 2D lock-exchange problem
3. Inversion: Surrogate Management Framework
4. Inversion results

Turbidity currents

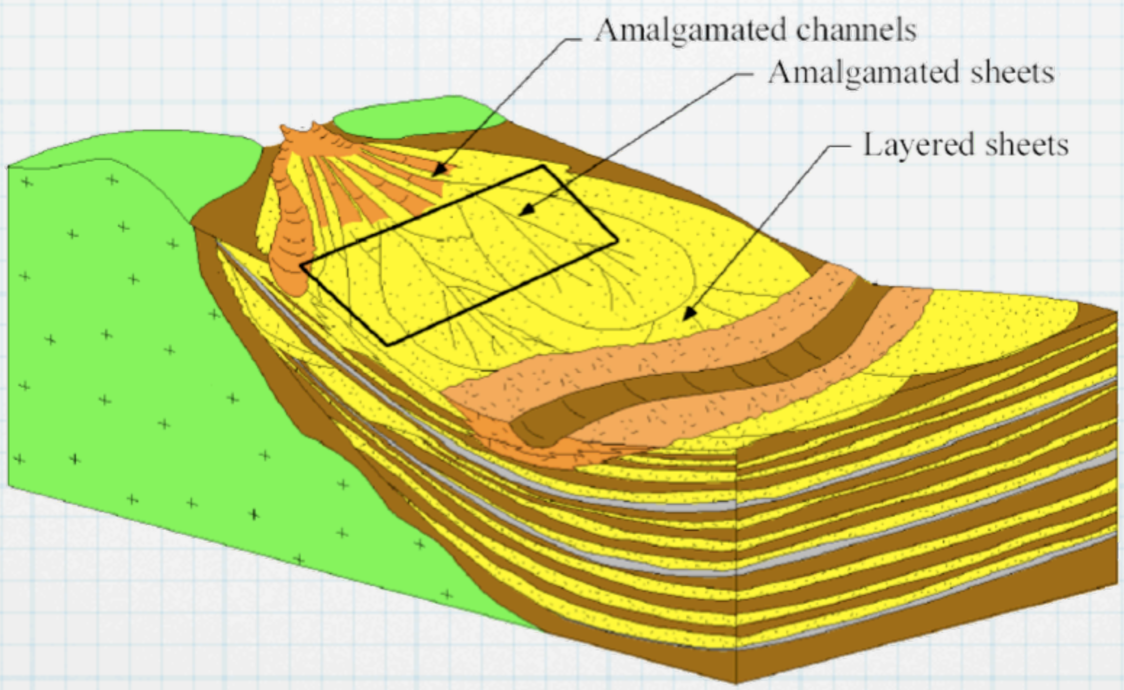
The Goleta slide, a few miles from UCSB campus



Credits: Brian Romans, “Clastic Detritus” blog

Turbidite systems

Ancient sediment deposits from turbidity currents form layered rock on the ocean floor.
Hydrocarbon reservoirs may form within these layers.



from Booth *et al*. 2000,
Deep-water reservoirs of the world,
Auger field, Gulf of Mexico

Turbidite modeling via flow inversion

Geological predictions of reservoir topography are mainly based on a few exploration wells (very expensive).

Our concept:

Reconstruct full deposit from CFD simulation of the original turbidity currents.

Task:

Identify suitable initial conditions, such that the simulated deposit matches measurements at control points (= exploration wells).

➛ iterative optimization

Direct numerical simulation

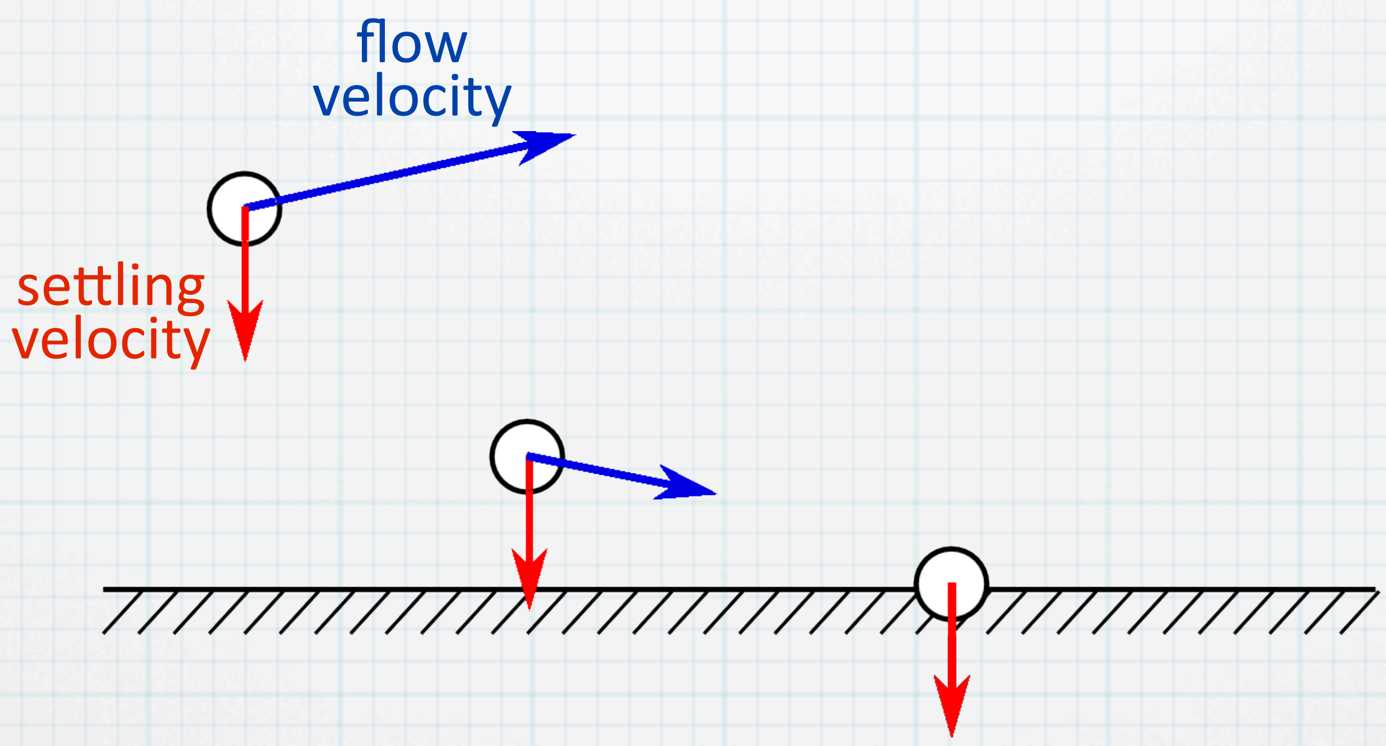
Governing equations: Navier-Stokes in Boussinesq approximation
(➛ dilute suspension, mass fraction of particles < 10%):

$$
\begin{aligned}
\nabla \cdot \vec{u}_f &= 0 \\
\frac{\partial \vec{u}_f}{\partial t} + (\vec{u}_f \cdot \nabla)\, \vec{u}_f &= -\nabla p + \frac{1}{Re} \nabla^2 \vec{u}_f + \sum_i \underbrace{c_i}_{\text{sediment concentrations}} \vec{e}_g \\
\frac{\partial c_i}{\partial t} + [(\vec{u}_f + \underbrace{\vec{v}_i^s}_{\text{settling velocity}}) \cdot \nabla]\, c_i &= \frac{1}{Sc\, Re} \nabla^2 c_i
\end{aligned}
$$

$$
Re = \frac{u_b\, L}{\nu}, \quad Sc = \frac{\nu}{\kappa}, \quad u_b = \sqrt{g L \Delta\rho / \rho_0}
$$

Deposition model

Sediment particles are allowed to settle through the bottom boundary, where they are recorded as deposit.



Deposit height in the continuum model:

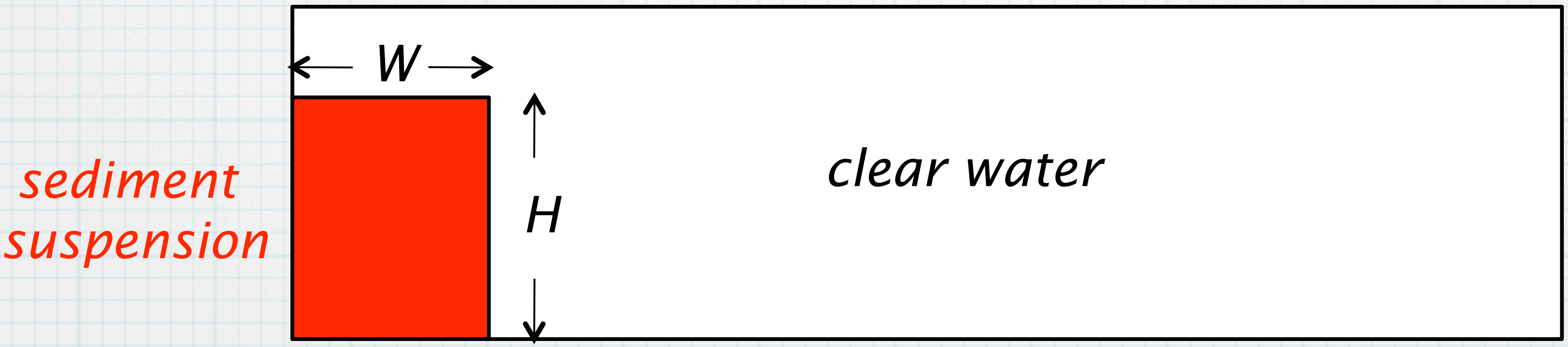
$$\frac{\partial h_i(x)}{\partial t} = \frac{v_i^s\, c_i(x, y=0)}{\sigma_i} \qquad \sigma_i = 1 : \text{ packing density}$$

Test problem for inversion

Simple academic test problem:

2D lock exchange configuration

with 3 different grain sizes in suspension

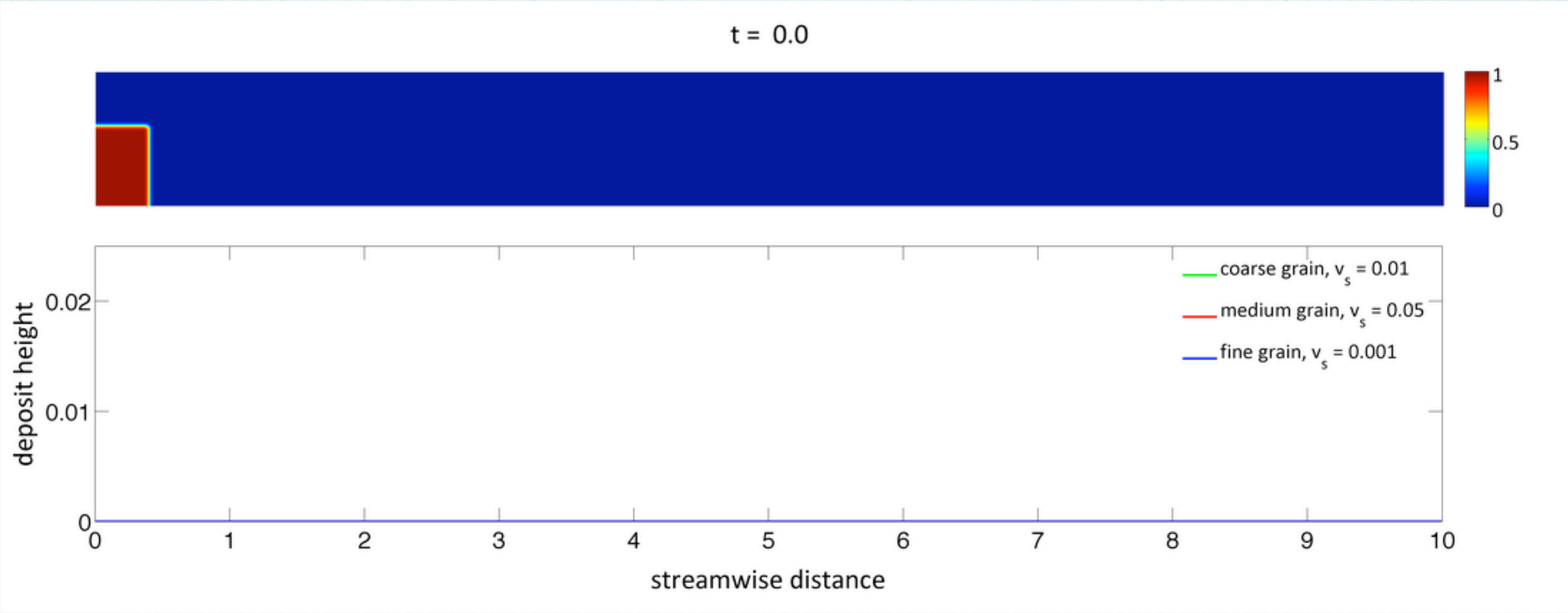


Parameters:

- H, W: lock dimensions
- Initial loading of each grain species
- Settling velocities (i.e. grain sizes)
- Reynolds number

Reference case simulation

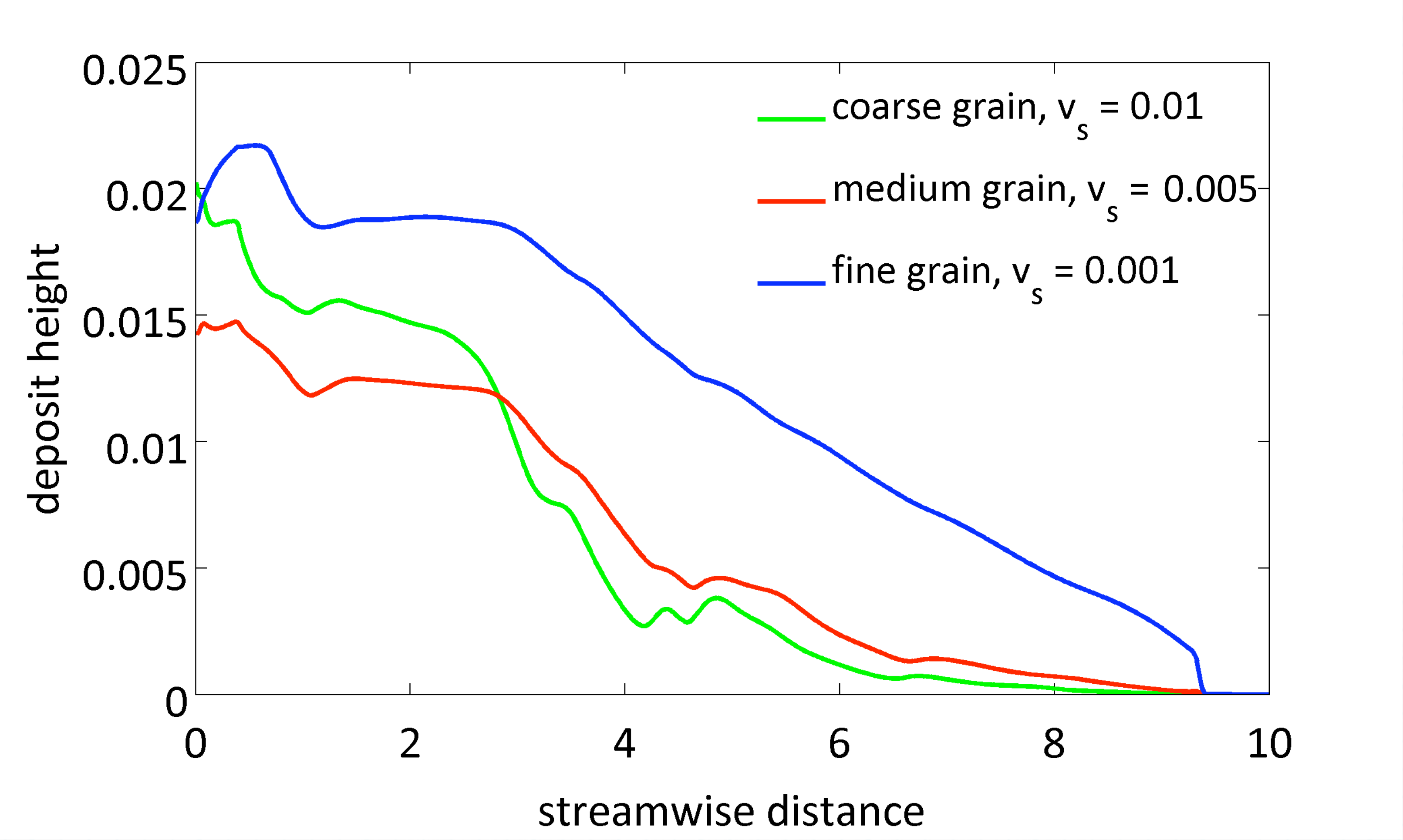
lock dimensions:	$W = 0.4$, $H = 0.6$,
initial concentrations:	$c_1 = 0.5$, $c_2 = 0.25$, $c_3 = 0.25$
settling velocities:	$v_1 = 0.001$, $v_2 = 0.005$, $v_3 = 0.01$
Reynolds number:	Re=2000



Code developed by M. Nasr-Azadani and B. Hall at UCSB

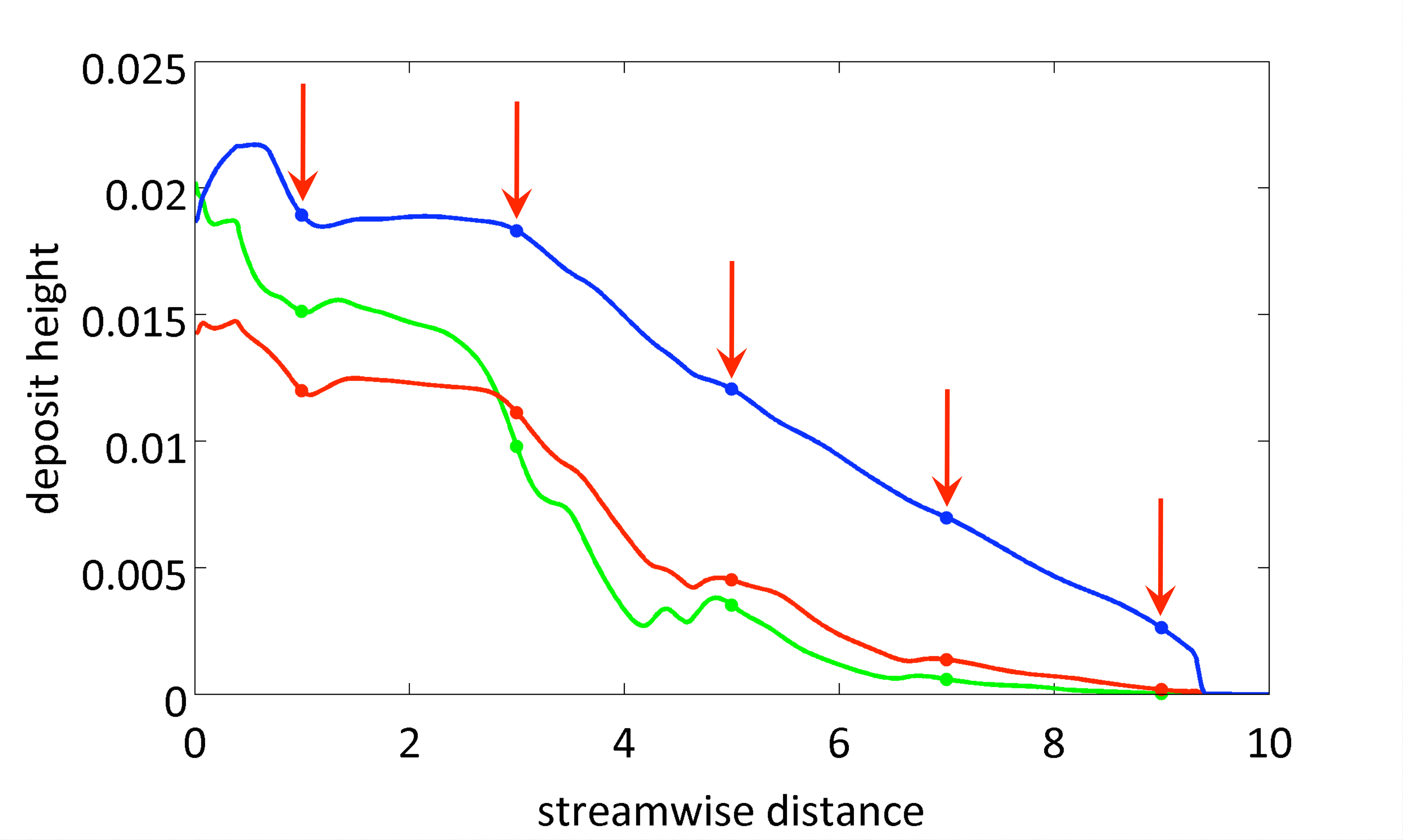
Reference case simulation

Retain the final deposit in only five points:



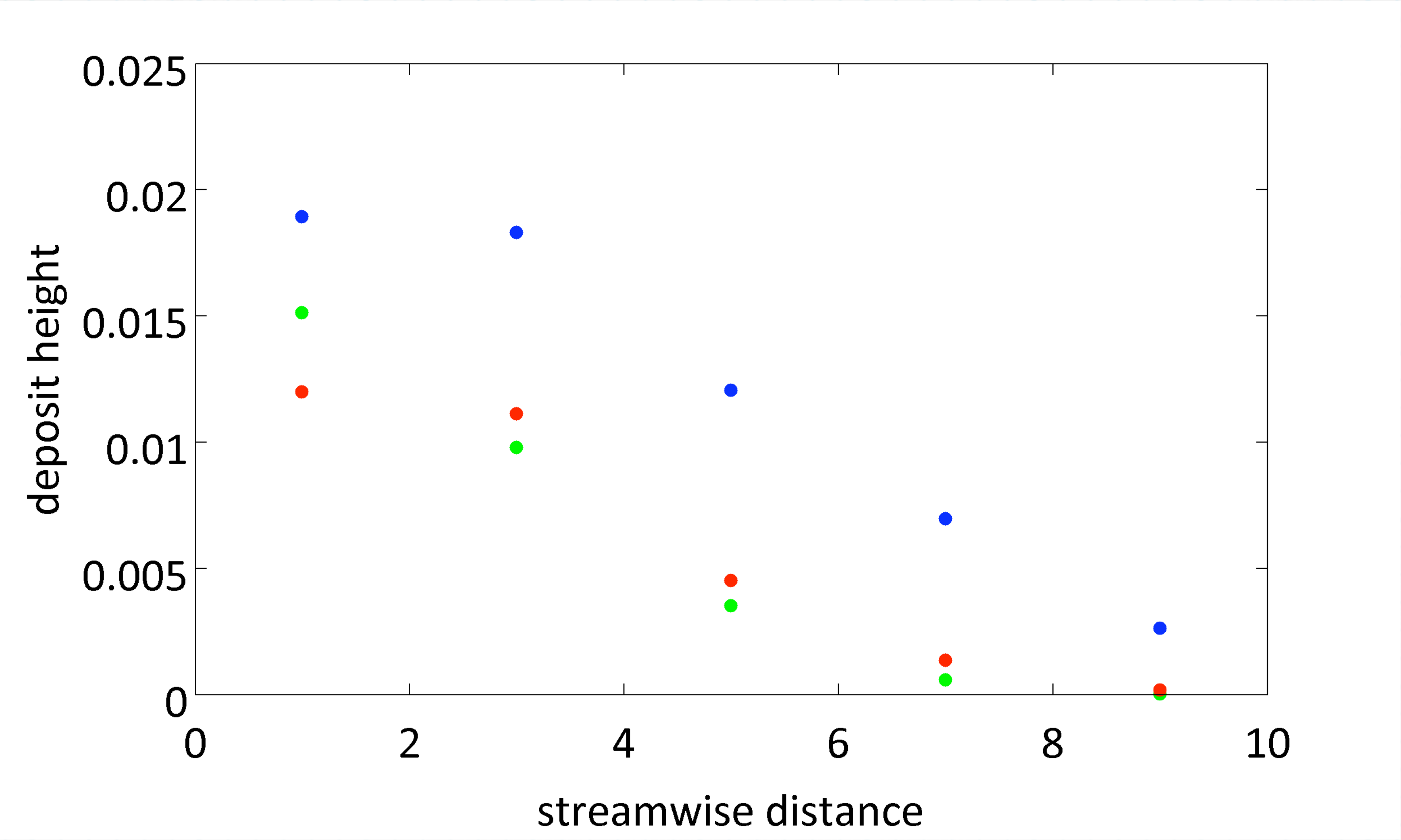
Reference case simulation

Retain the final deposit in only five points:



Reference case simulation

Retain the final deposit in only five points:

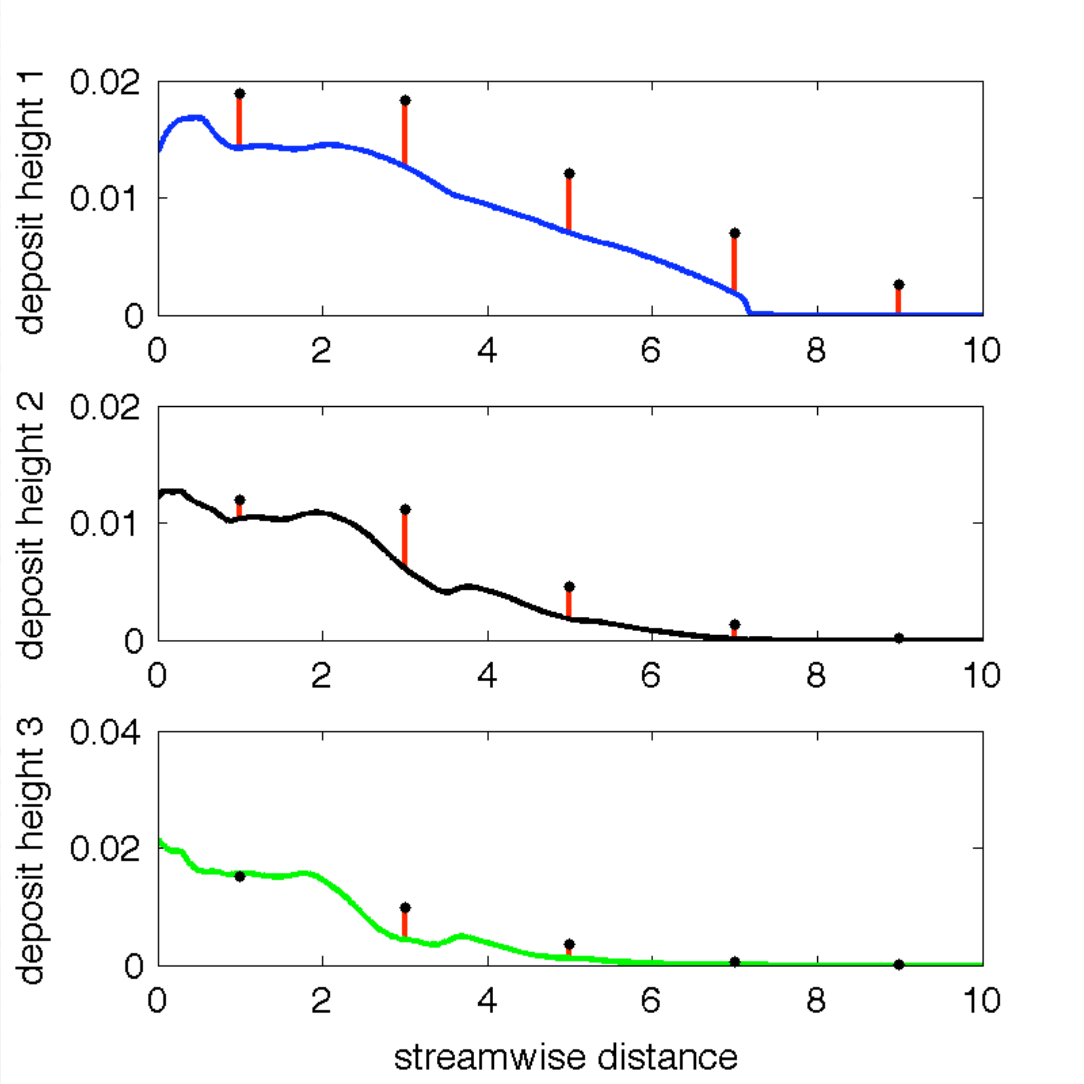


How can we reconstruct the initial run conditions from this information?

Optimization problem

Objective: Vary initial conditions of DNS in order to minimize "error" of the final deposit in our control points.

Cost function: sum of L_2-norms $\quad J = J_1 + J_2 + J_3$



$$J_1 = \left(\sum_{i=1}^{5} E(x_i)^2 \right)^{\frac{1}{2}}$$

$$J_2 = \left(\sum_{i=1}^{5} E(x_i)^2 \right)^{\frac{1}{2}}$$

$$J_3 = \left(\sum_{i=1}^{5} E(x_i)^2 \right)^{\frac{1}{2}}$$

Optimization problem

Objective: Vary initial conditions of DNS in order to minimize
"error" of the final deposit in our control points.

Cost function: sum of L_2-norms

Vary 5 free parameters:
lock width W, lock height H,
initial concentrations c_1, c_2, c_3 of three sediment species

Method: Surrogate Management Framework
(Booker et al. 1999, Marsden et al. 2007)
for gradient-free optimization.
Proven efficiency for numerically expensive problems.

Surrogate management framework

Alternate two algorithms:

search: explores the parameter space globally, predicts minima due to a Kriging interpolation function (cost function “surrogate”).

poll: probes the true cost function in the vicinity of the current best candidate, confirms local minimum.

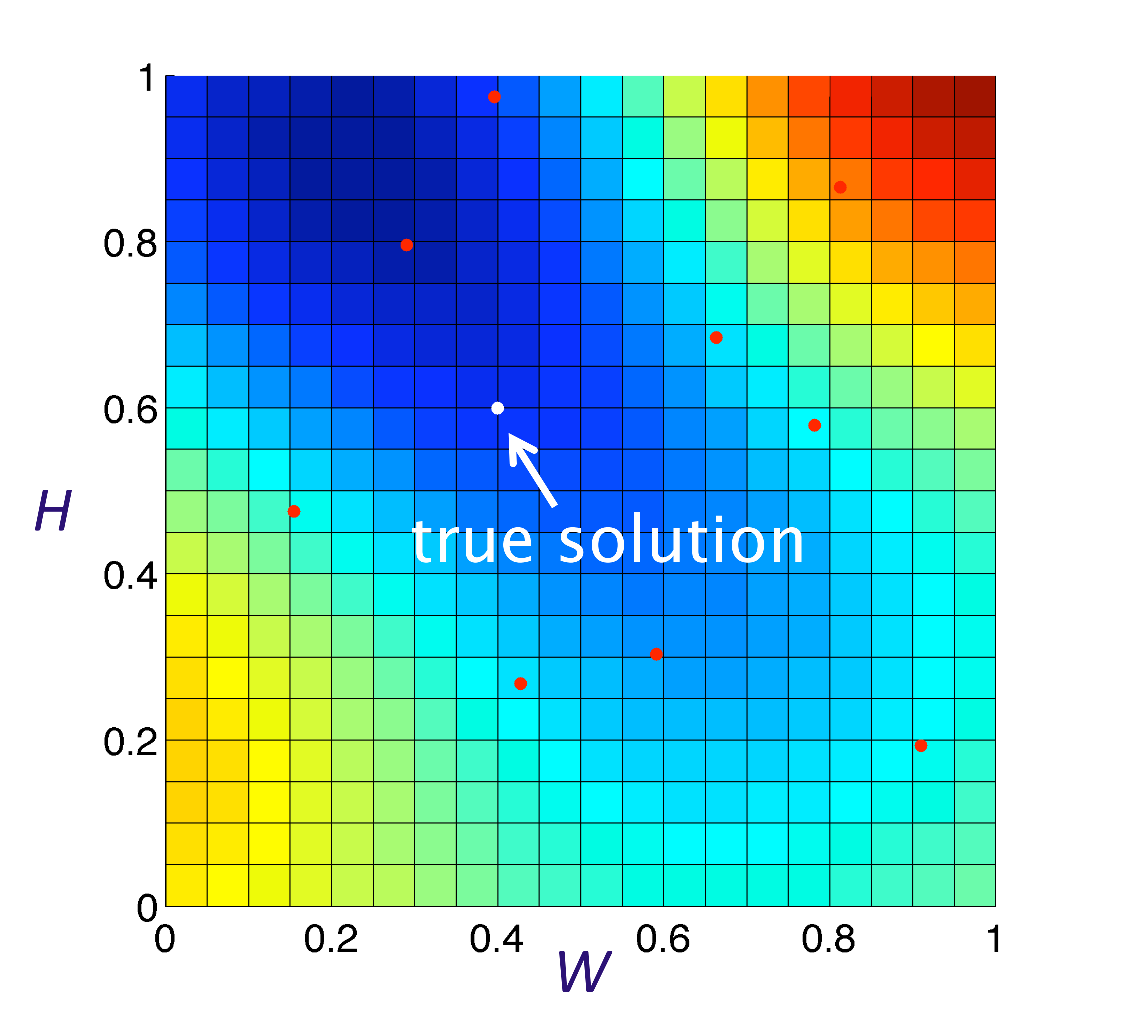
Iterate “search” while successful, then invoke “poll”, then switch back to “search” ... until converged!

Surrogate management framework

Example with only 2 free parameters:
lock width *W*, lock height *H*

1. Generate several configurations as initial guesses (Latin Hypercube Sampling). Run simulations, evaluate cost function.

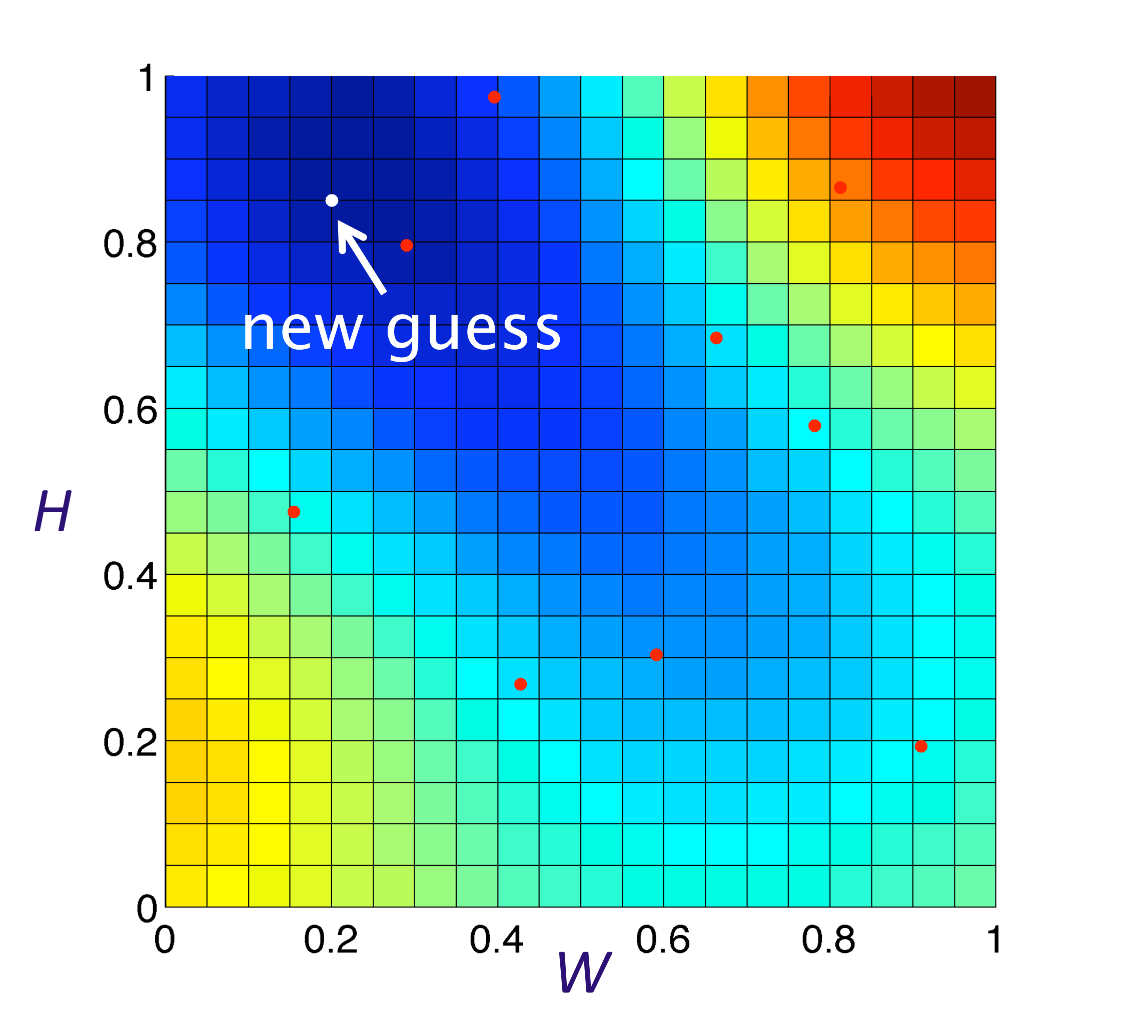
2. Construct a Kriging interpolation of the cost function.



Surrogate management framework

Example with only 2 free parameters:
lock width W, lock height H

3. Find global minimum of interpolation function, run new simulation with these parameters.



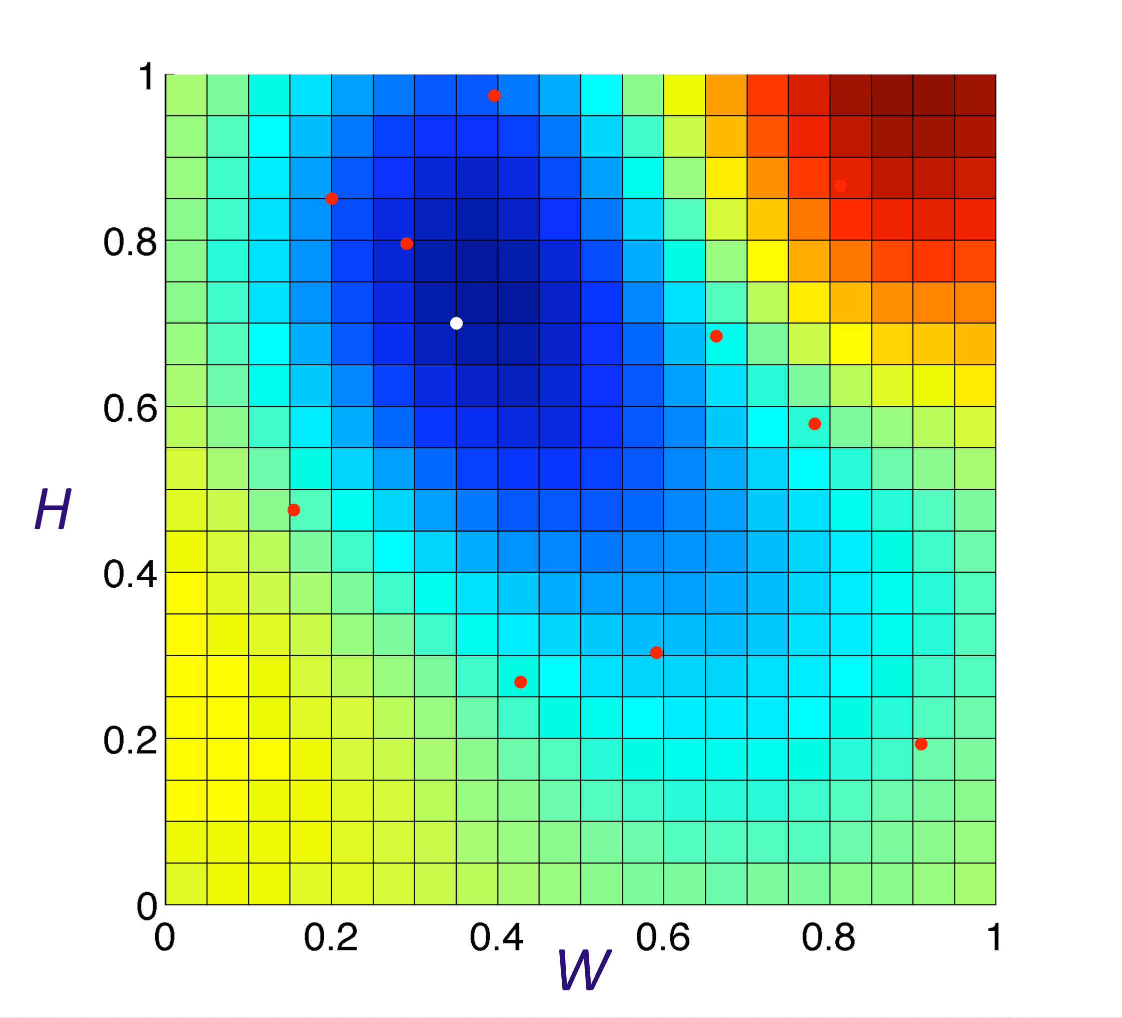
Surrogate management framework

Example with only 2 free parameters:
lock width *W*, lock height *H*

3. Find global minimum of interpolation function, run new simulation with these parameters.

4. Construct new interpolation.

Repeat, repeat, repeat…

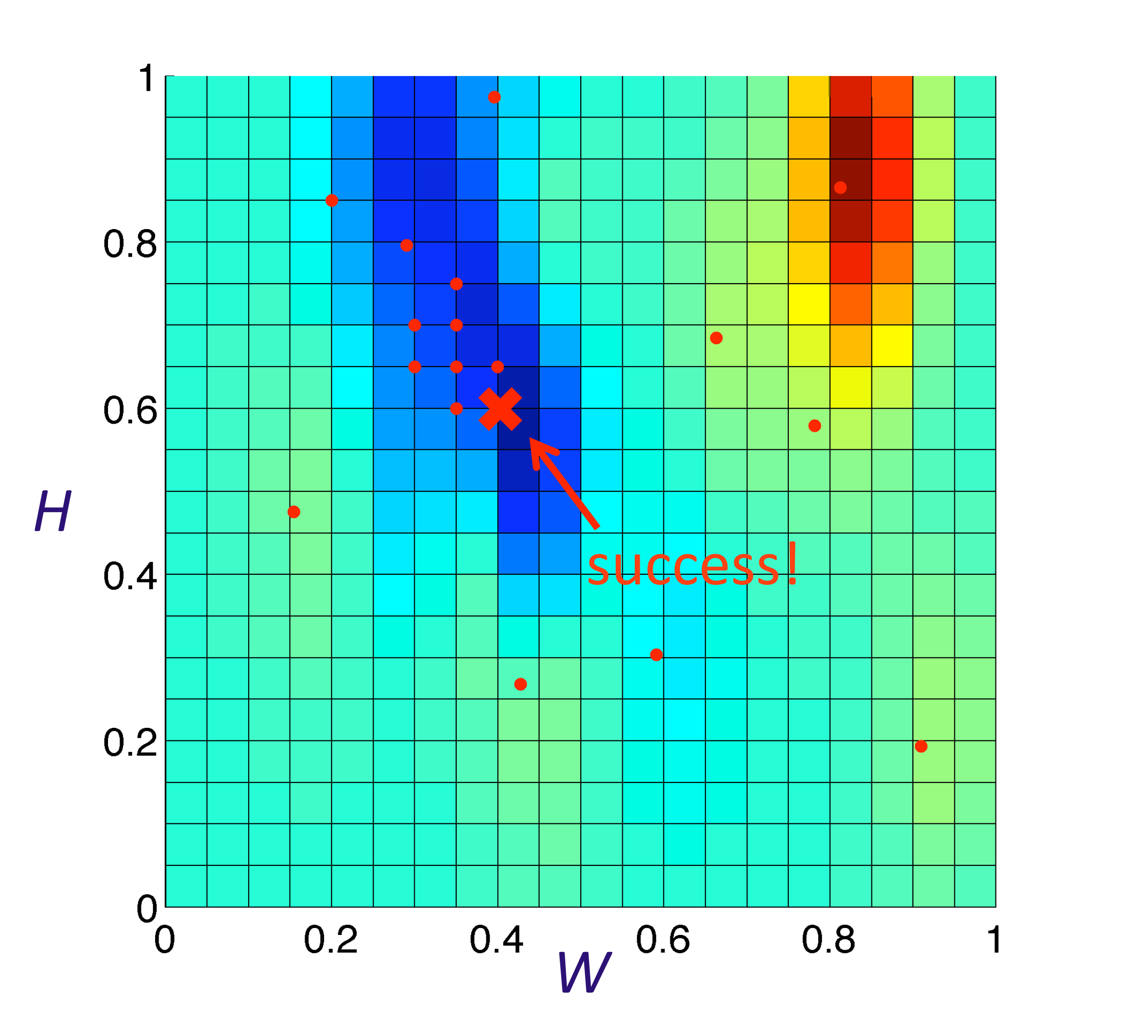


Surrogate management framework

Example with only 2 free parameters:
lock width *W*, lock height *H*

Algorithm converges to true solution after 8 "search" steps.

Lucky circumstance!
Kriging could converge to a nearby point instead, or become ill-conditioned.
Therefore: "poll" routine.

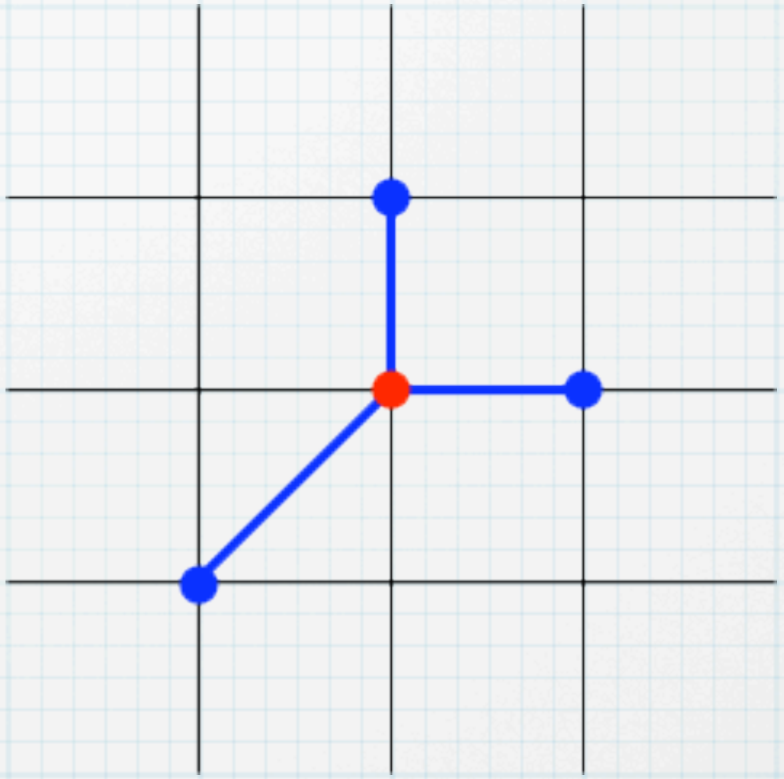


Surrogate management framework

POLL algorithm:

evaluate cost function on N+1
    mesh neighbors of the current
    best parameter setting.
The polling directions must form
    a *positive spanning basis*.

If improved point is found:
    new "search" step,
otherwise:
    assume we have found the
    global minimum.



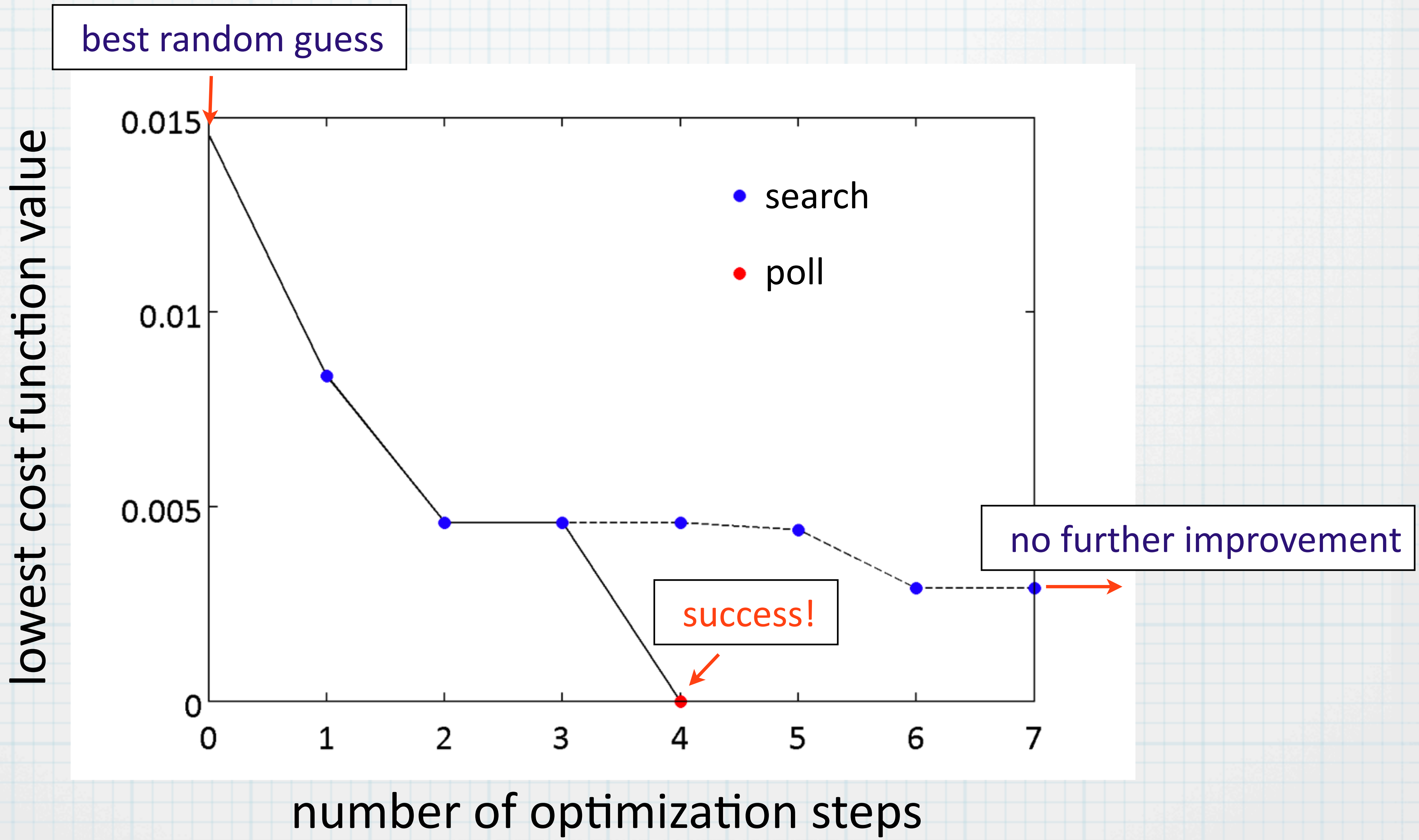
2D: 8 neighbor points
5D: 242 neighbor points

Inversion of the 5-parameter problem

- start with 50 initial guesses,
  build initial Kriging interpolation of cost function
- run five new simulations in parallel at each search step:
  - up to 3 minima as predicted by surrogate,
  - unexplored regions of parameter space (Cox & John 1997)
- run six new simulations in parallel at each poll step:
  estimate most promising polling directions by rescaling the deposit profile of current best candidate

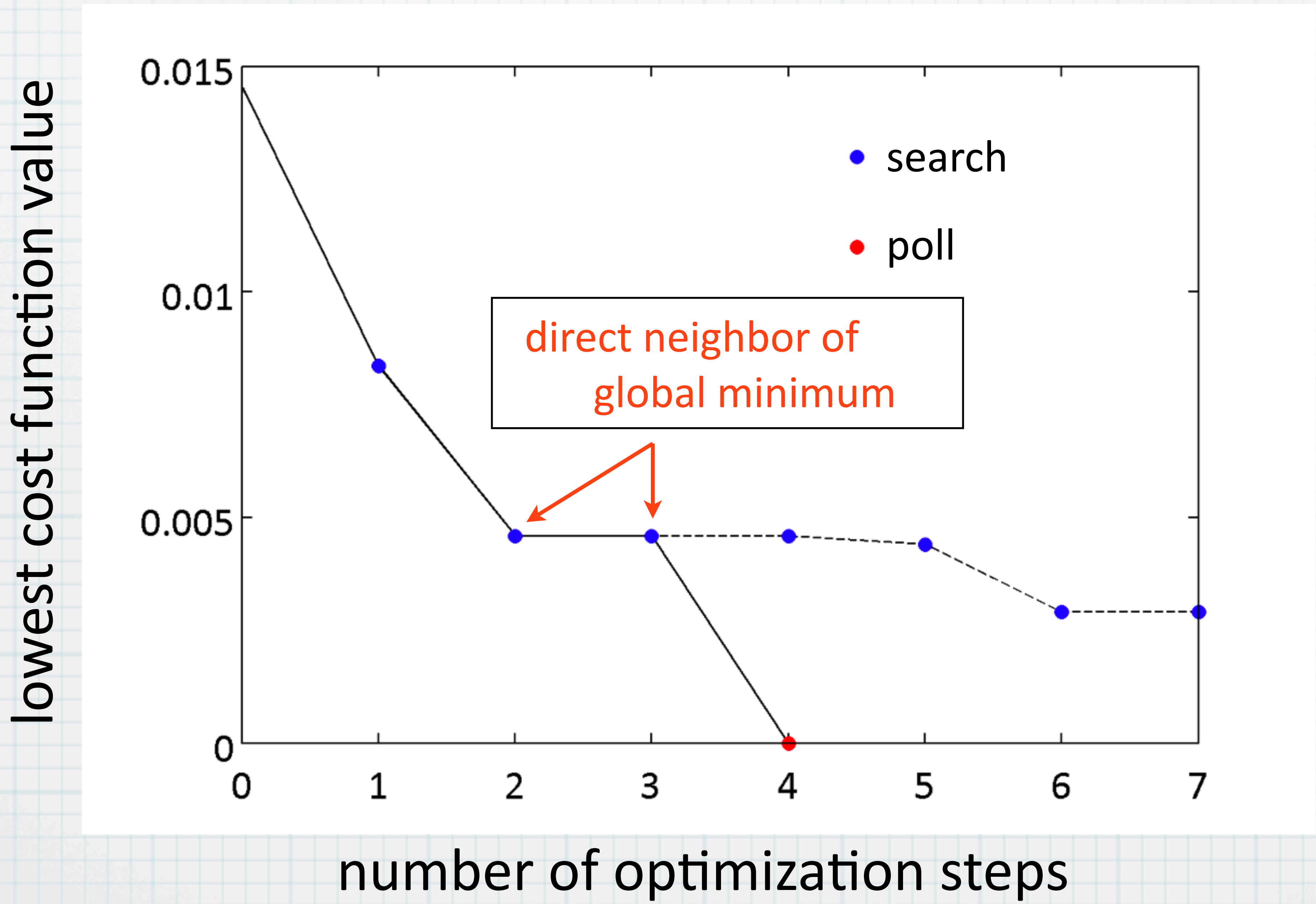
Results for the 5-parameter problem

Full convergence after four steps! (three “search”, one “poll”)



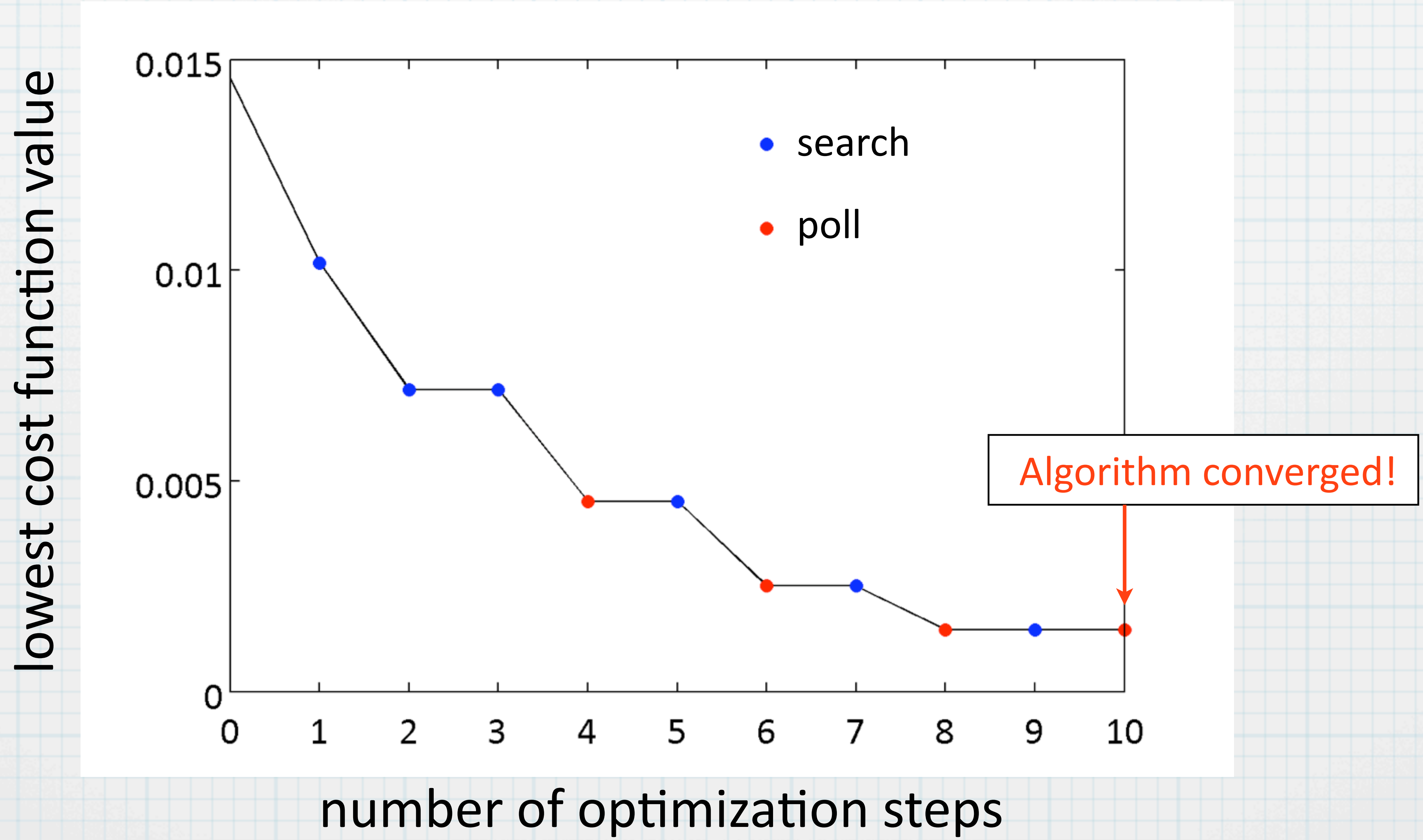
Results for the 5-parameter problem

Super-efficient algorithm? Pure luck? Exhaustive initial sampling?



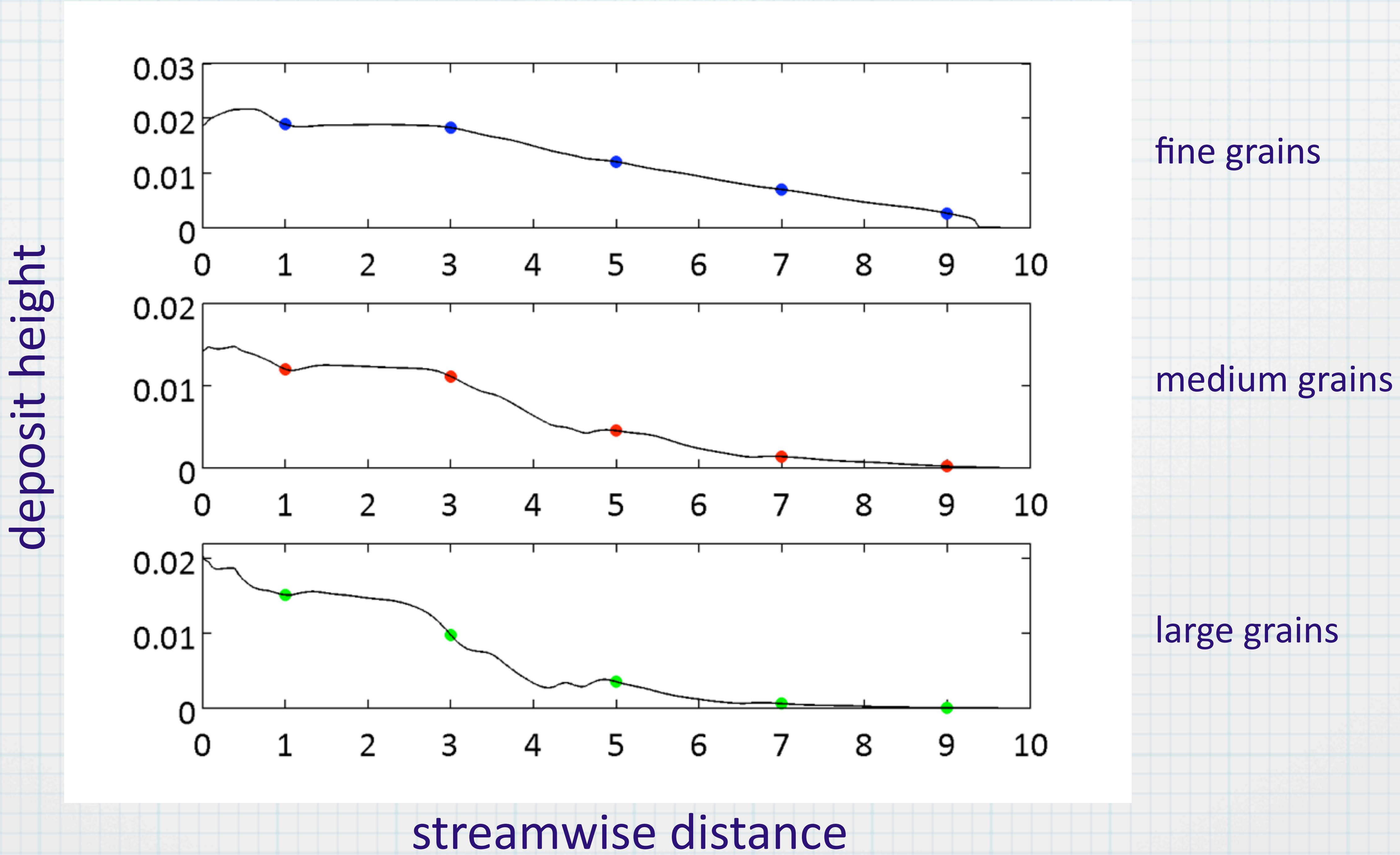
Results for the 5-parameter problem

Start with reduced set of 25 initial points:



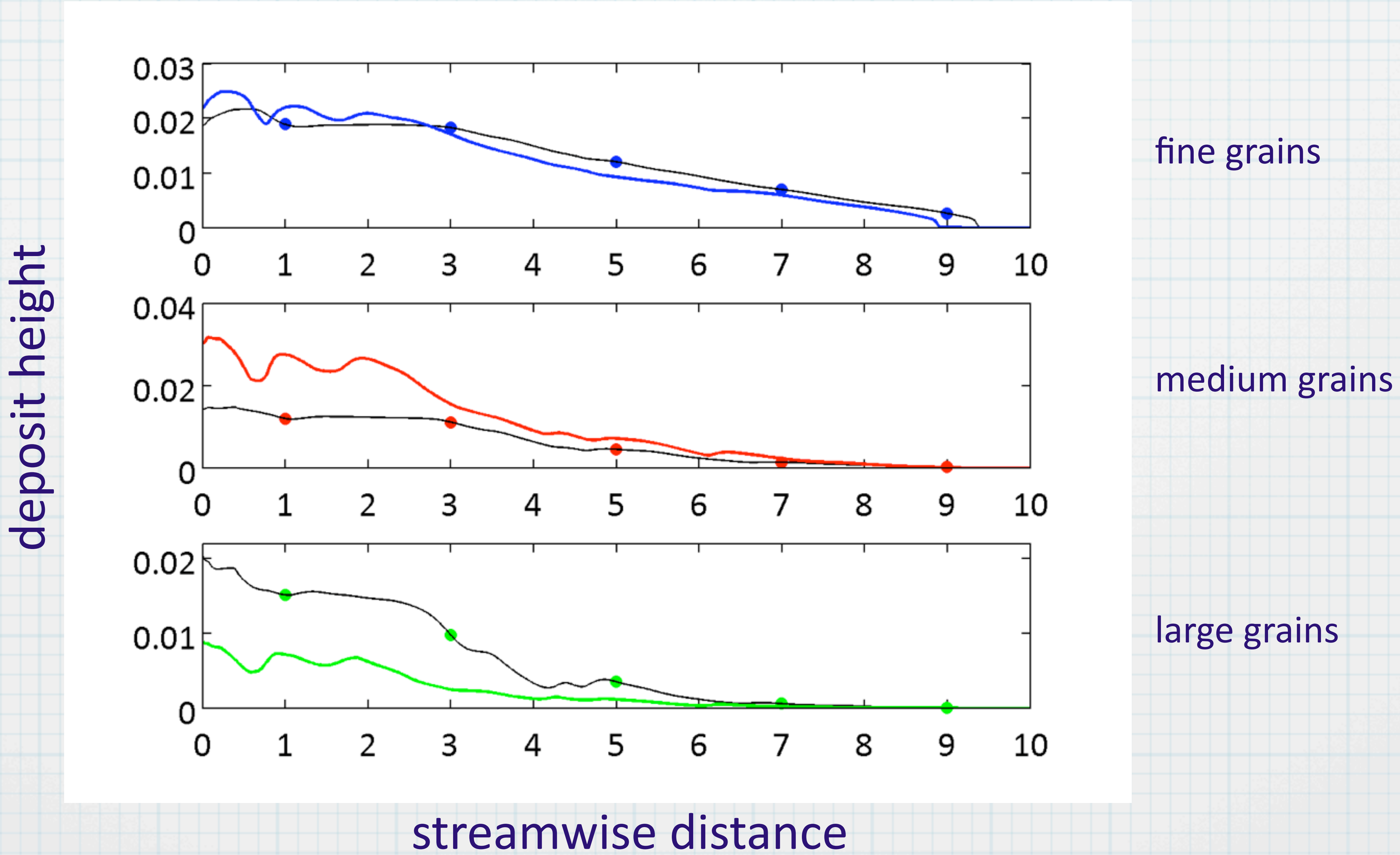
Results for the 5-parameter problem

Reference deposit profiles:



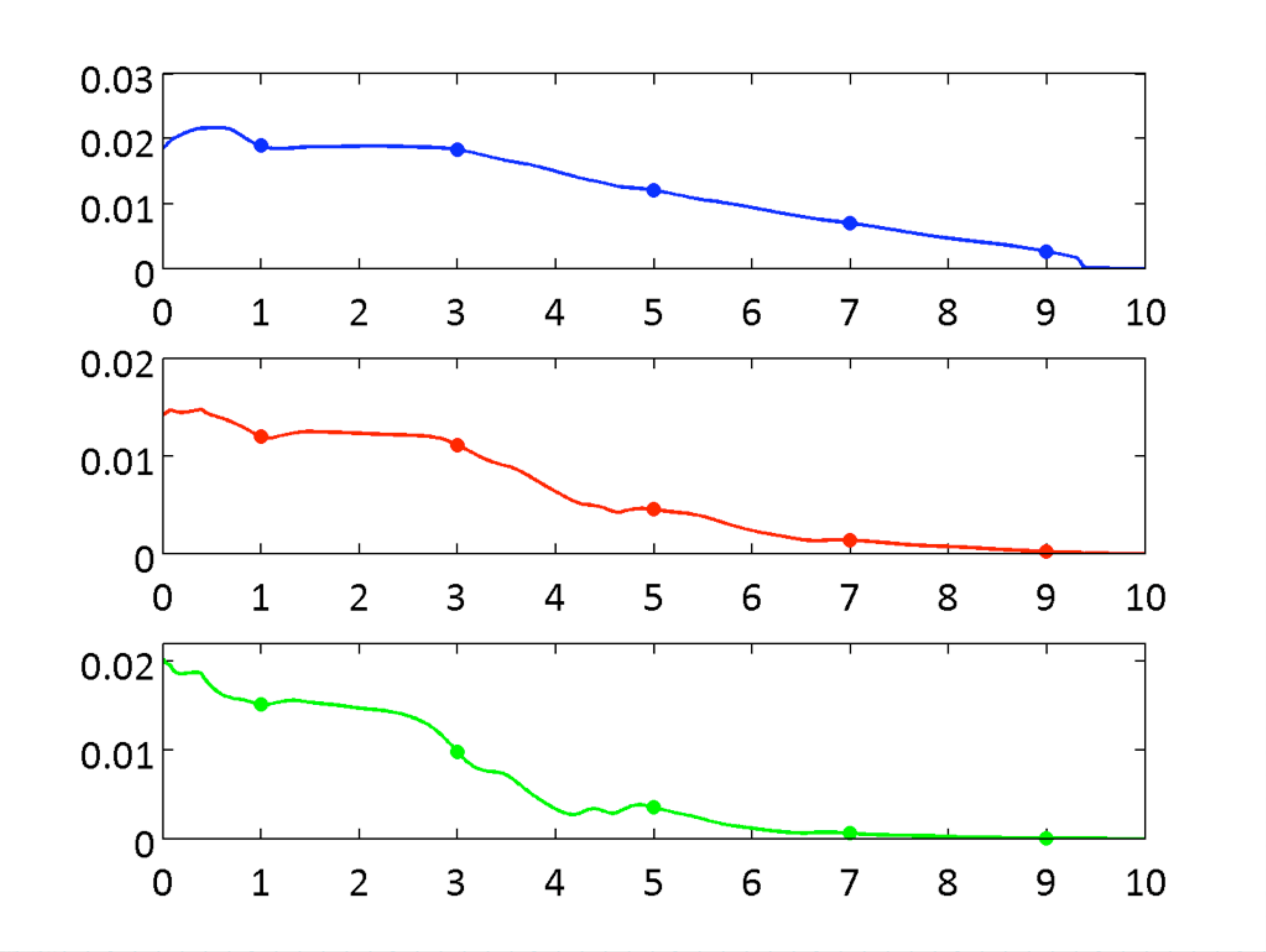
Results for the 5-parameter problem

Best initial random guess:



Results for the 5-parameter problem

Final result with 50 initial points: perfect matching



fine grains

medium grains

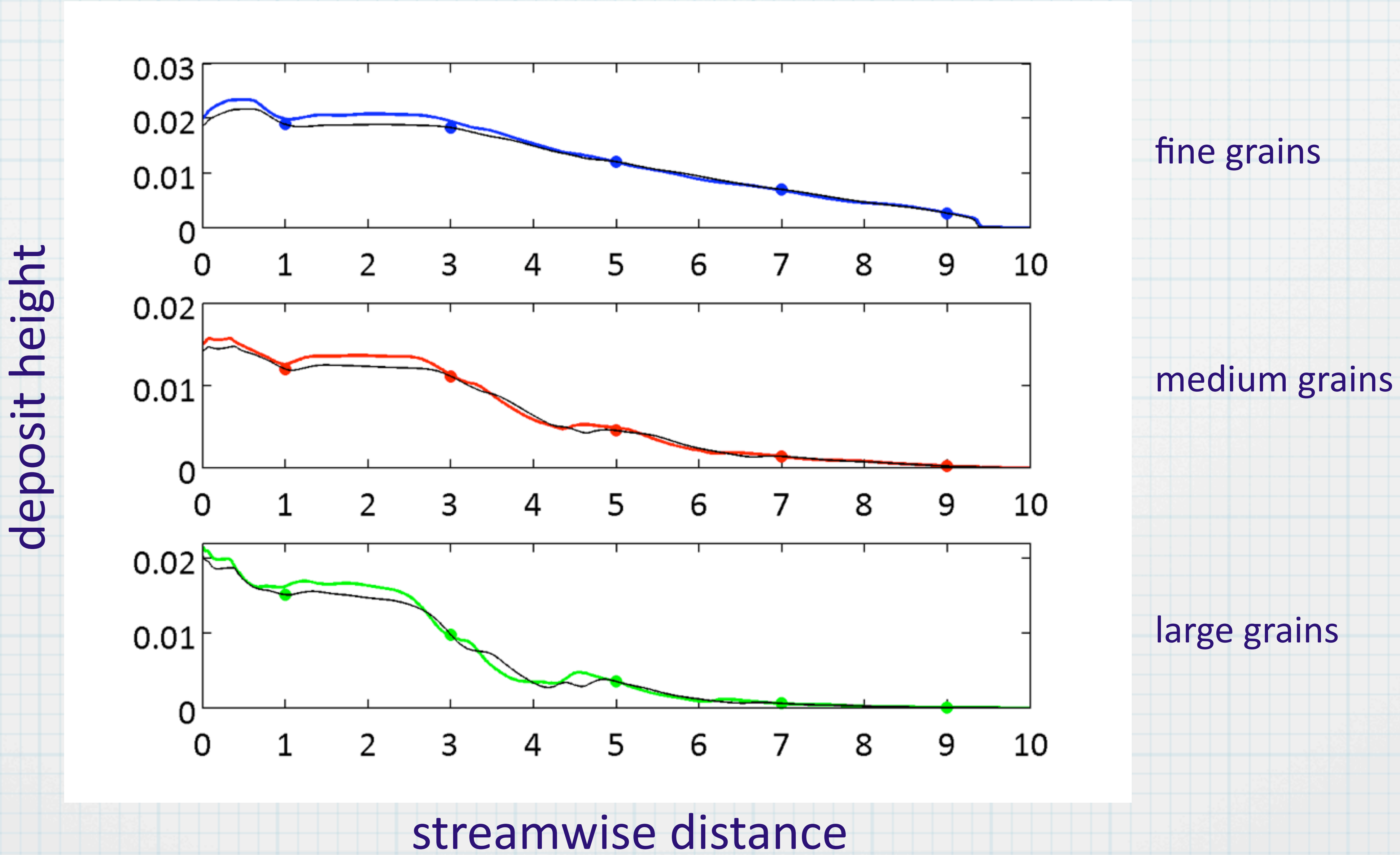
large grains

deposit height

streamwise distance

Results for the 5-parameter problem

Final result with 25 initial points:



Conclusions

- Formulation of the inverse problem (cost function based on few control points, optimization via SMF algorithm) leads to successful reconstruction of full deposit profiles within acceptable accuracy.
- Efficiency of "search" step depends on well-behaved Kriging interpolation. In both settings attempted here, the surrogate degenerates after few optimization steps.
- "Poll" algorithm provides no guarantee of convergence to exact minimum.
- Thorough initial sampling of the parameter space seems to be a good idea.

Outlook

- Field-scale simulations (3D, very high Reynolds number): implement RANS model
- Test inversion procedure with many successive turbidity currents (problem: resuspension)