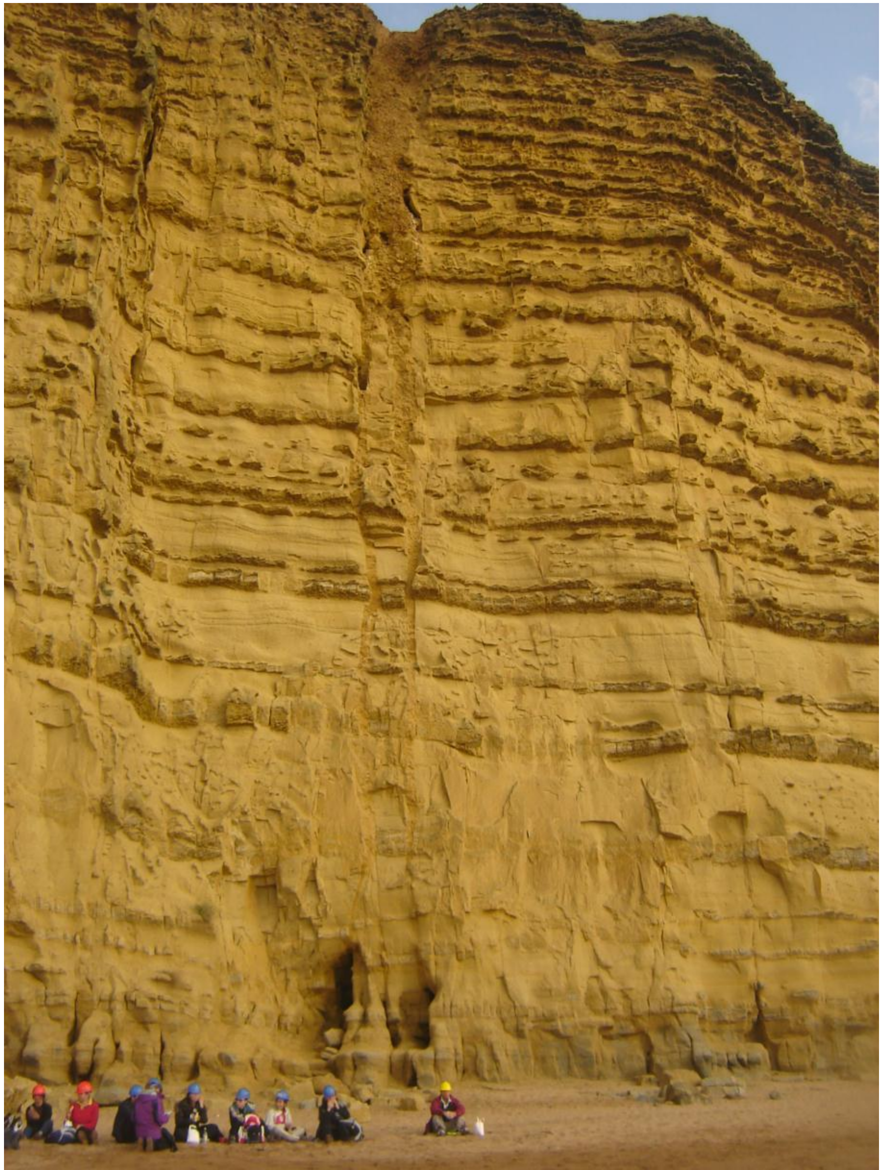
Parallel Tempering Applied to Reservoir History Matching

Jonathan Carter

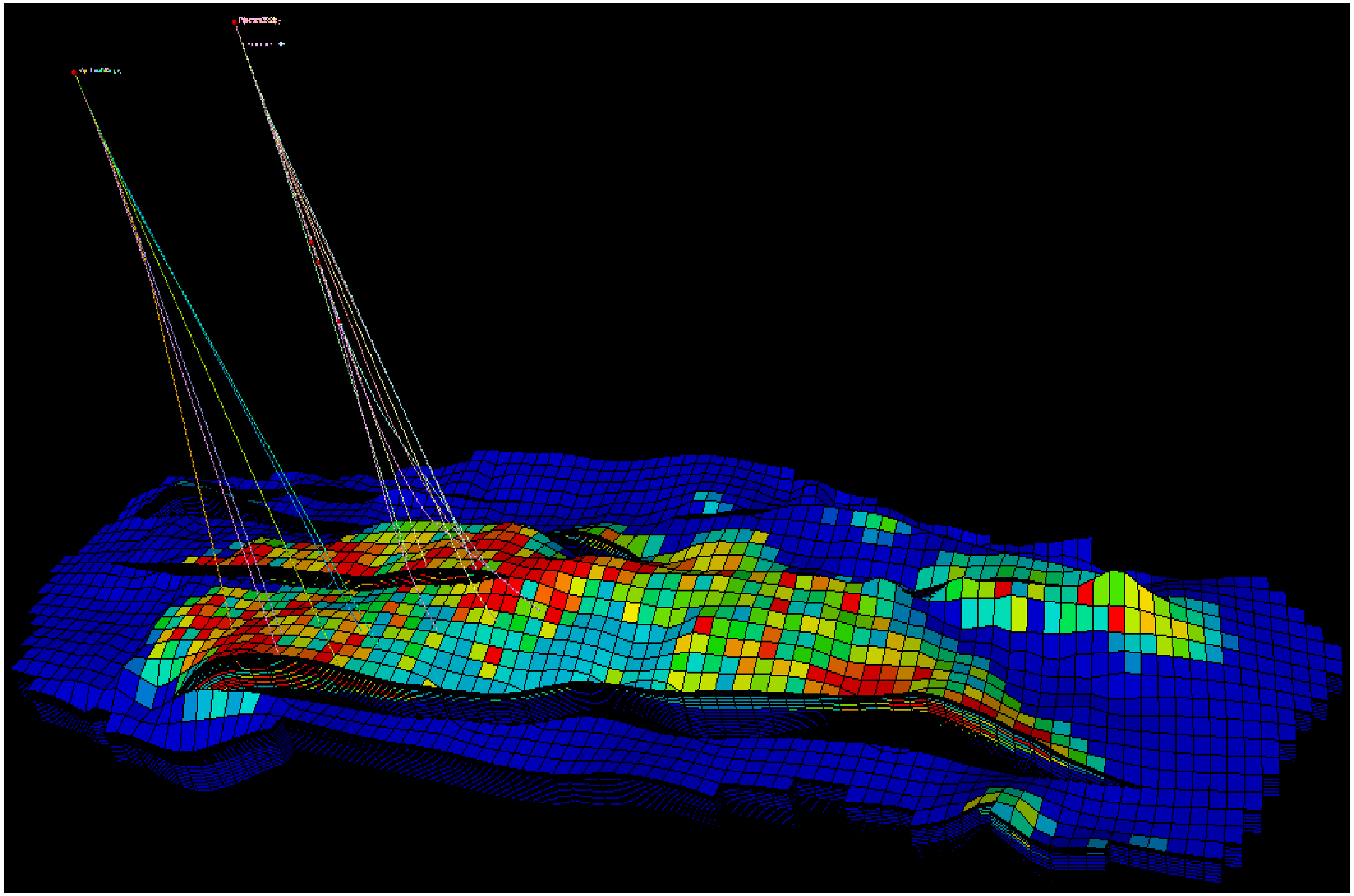
EON Innovation Centre for Exploration and Production

Some Geology

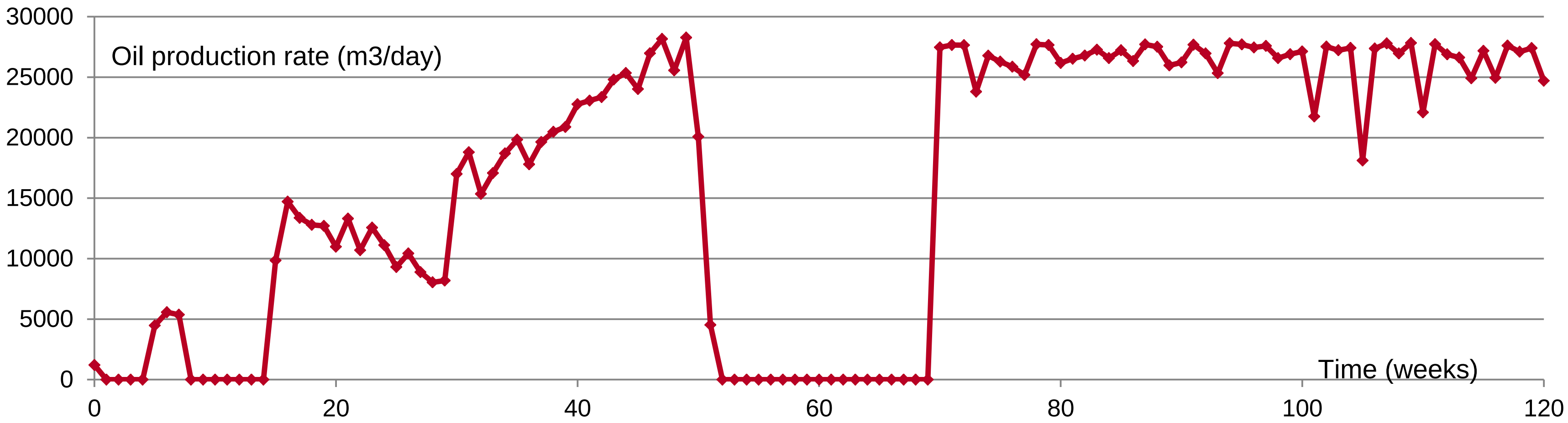
Some More Geology

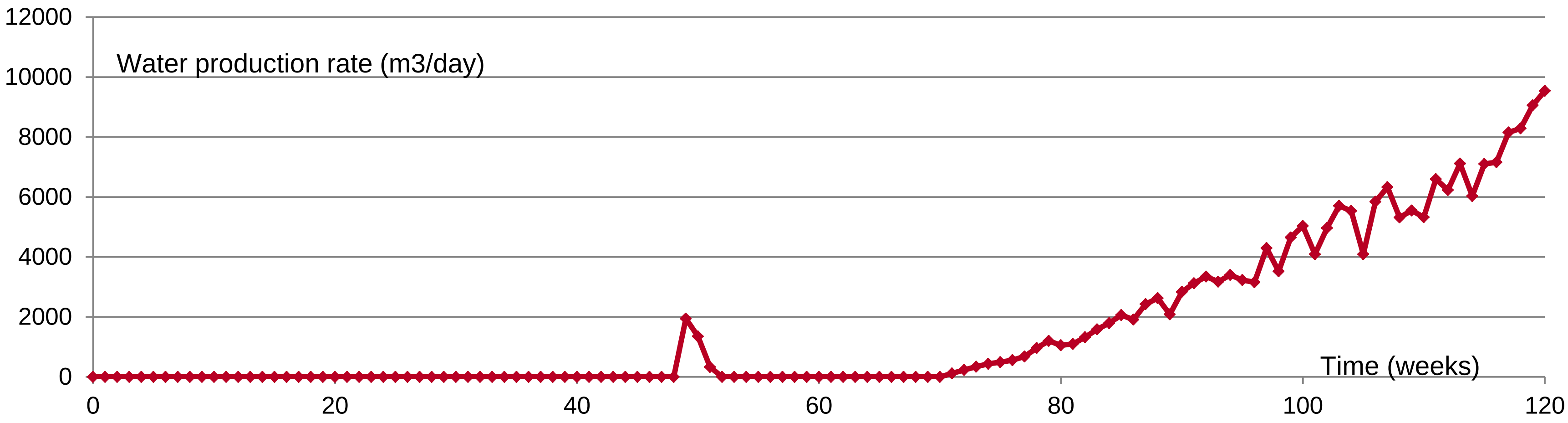


A Typical Simulation Model



Some Typical Production Data





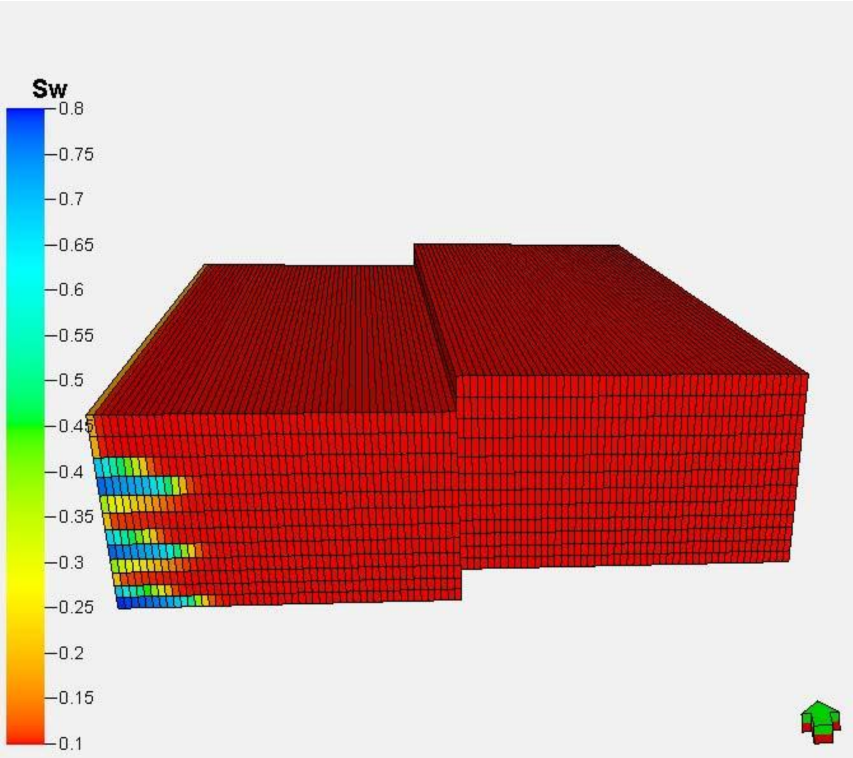
The Bet

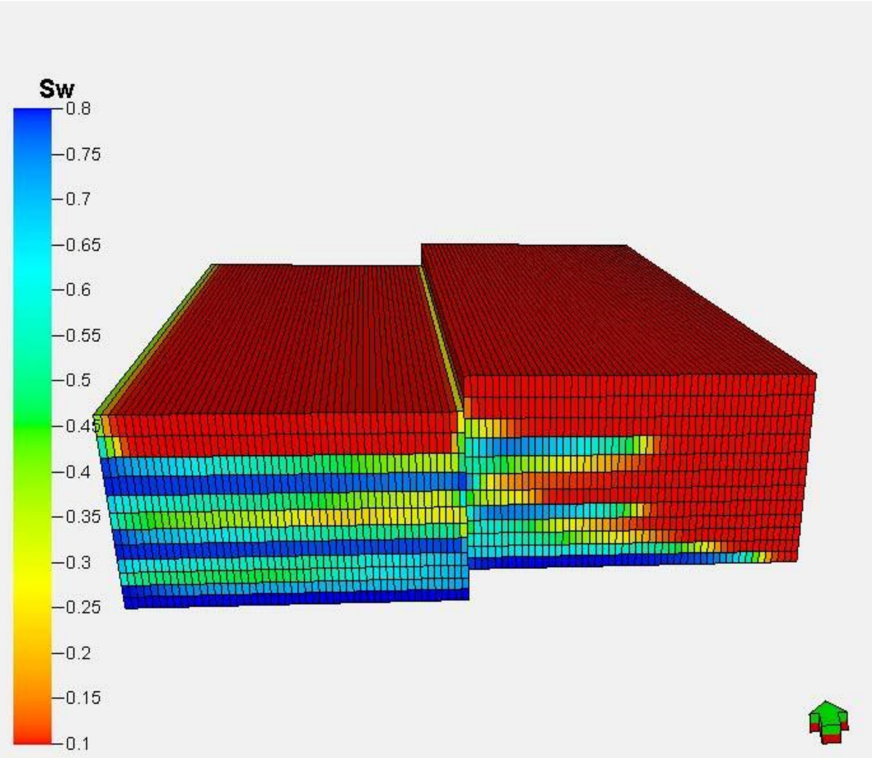
How many extra wells do I need to produce unswept oil?

Where should I put them?

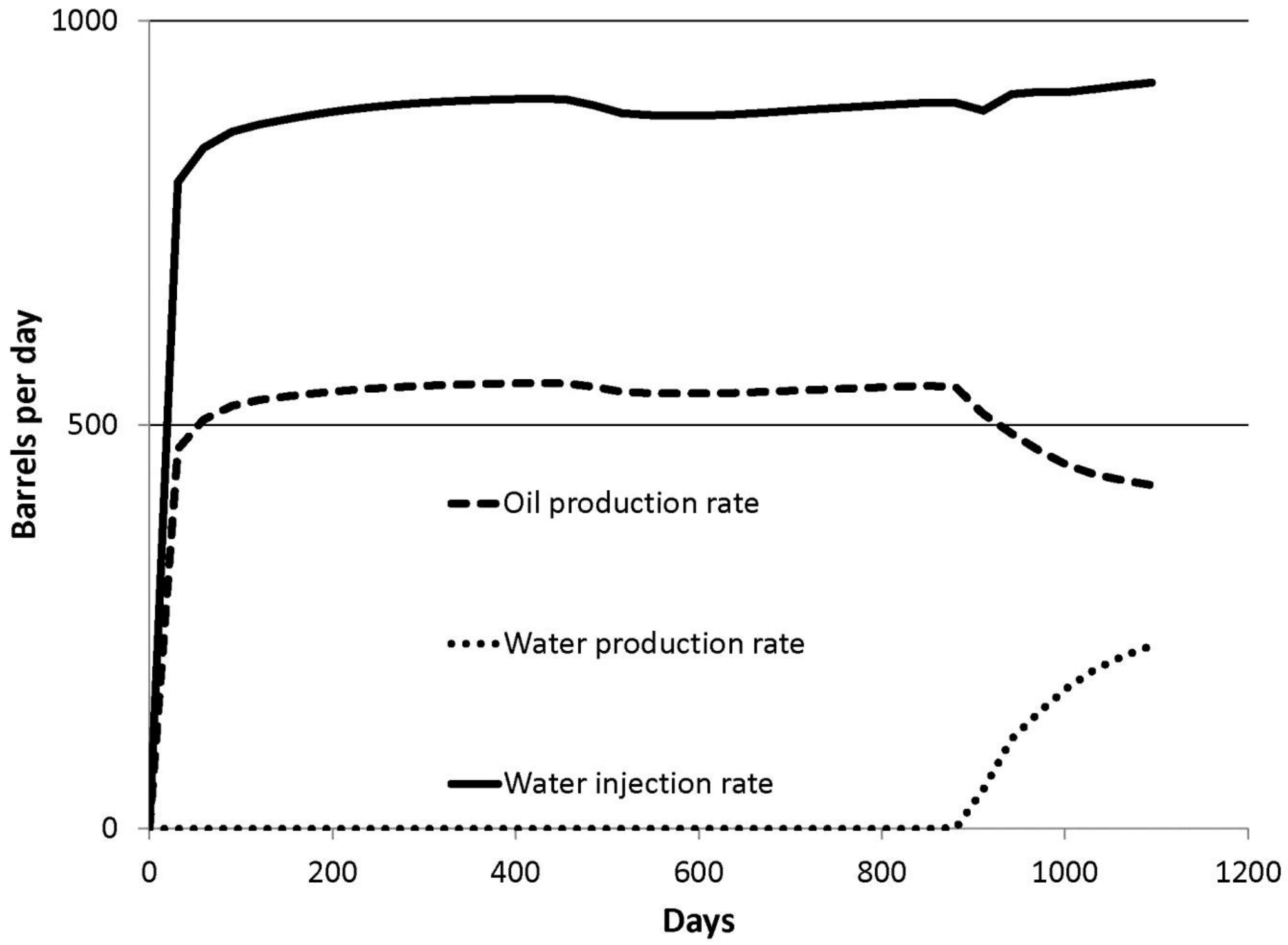
Cost per well $20M

The Test Model

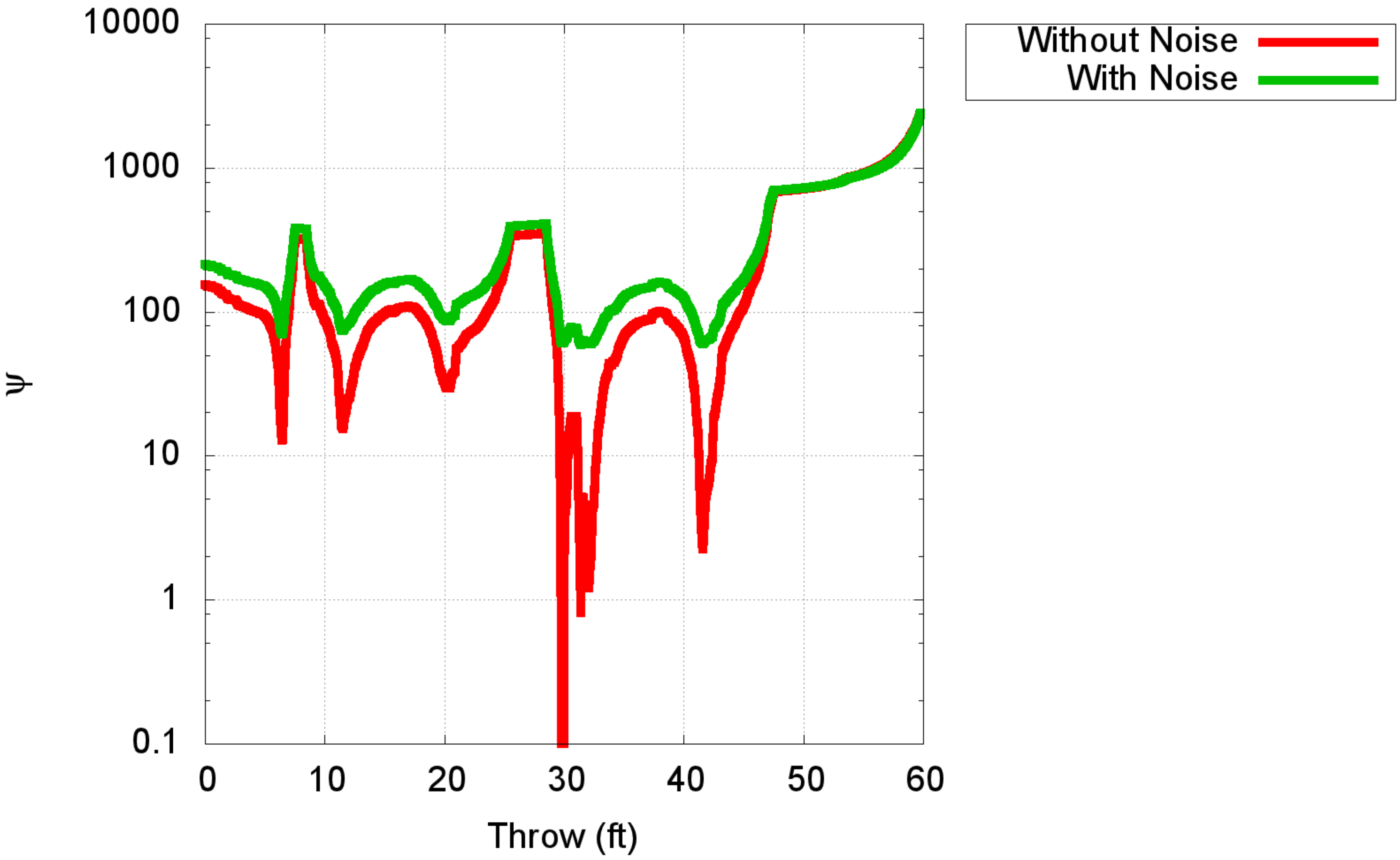




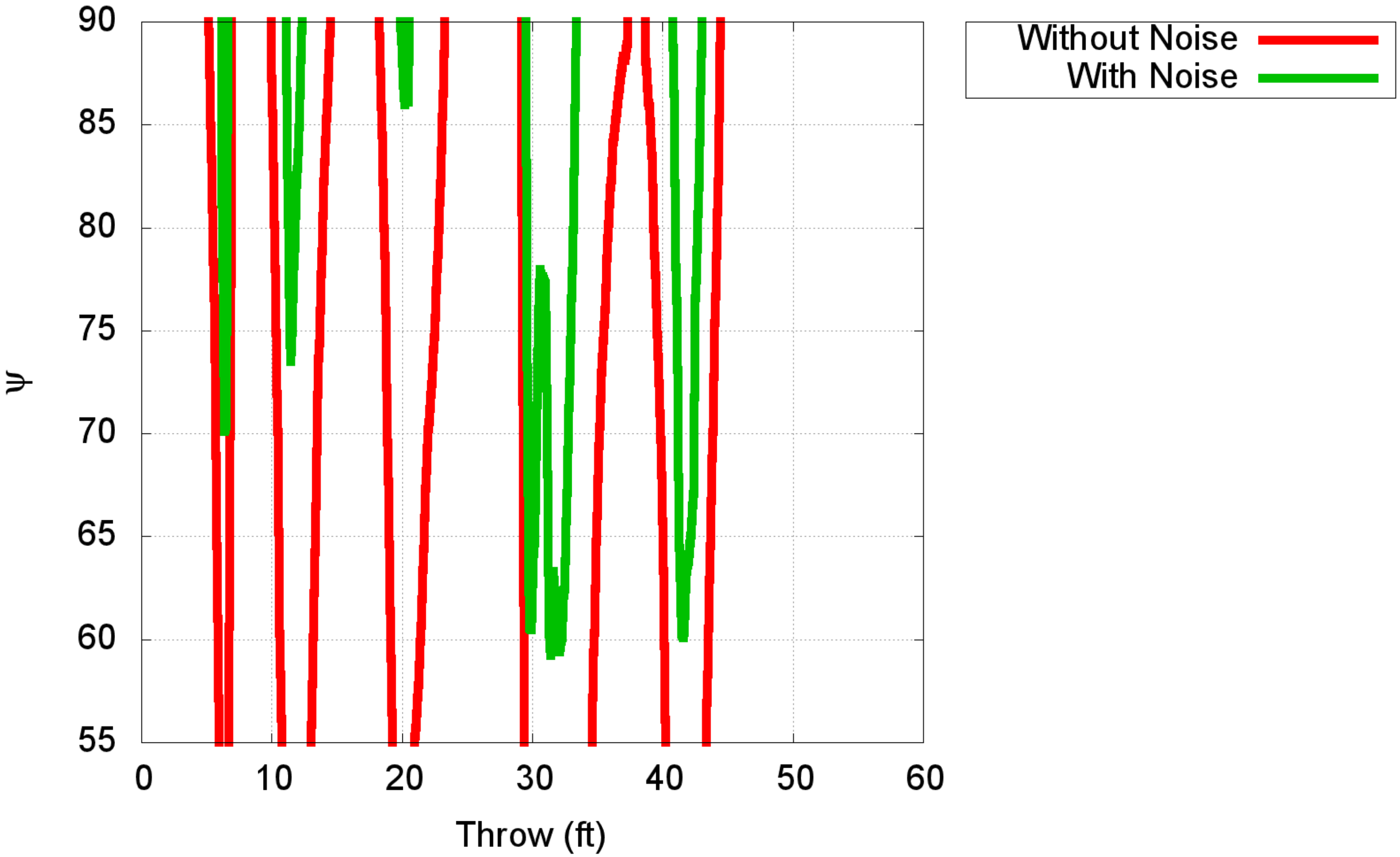
Production Data



Objective Function



Objective Function



Statement of the Problem

Find all of the different local optima (location and width) in a reasonable number of function evaluations

Each function evaluation typically takes 3-4 hours

We do not know in advance the number and location of local optima in parameter space

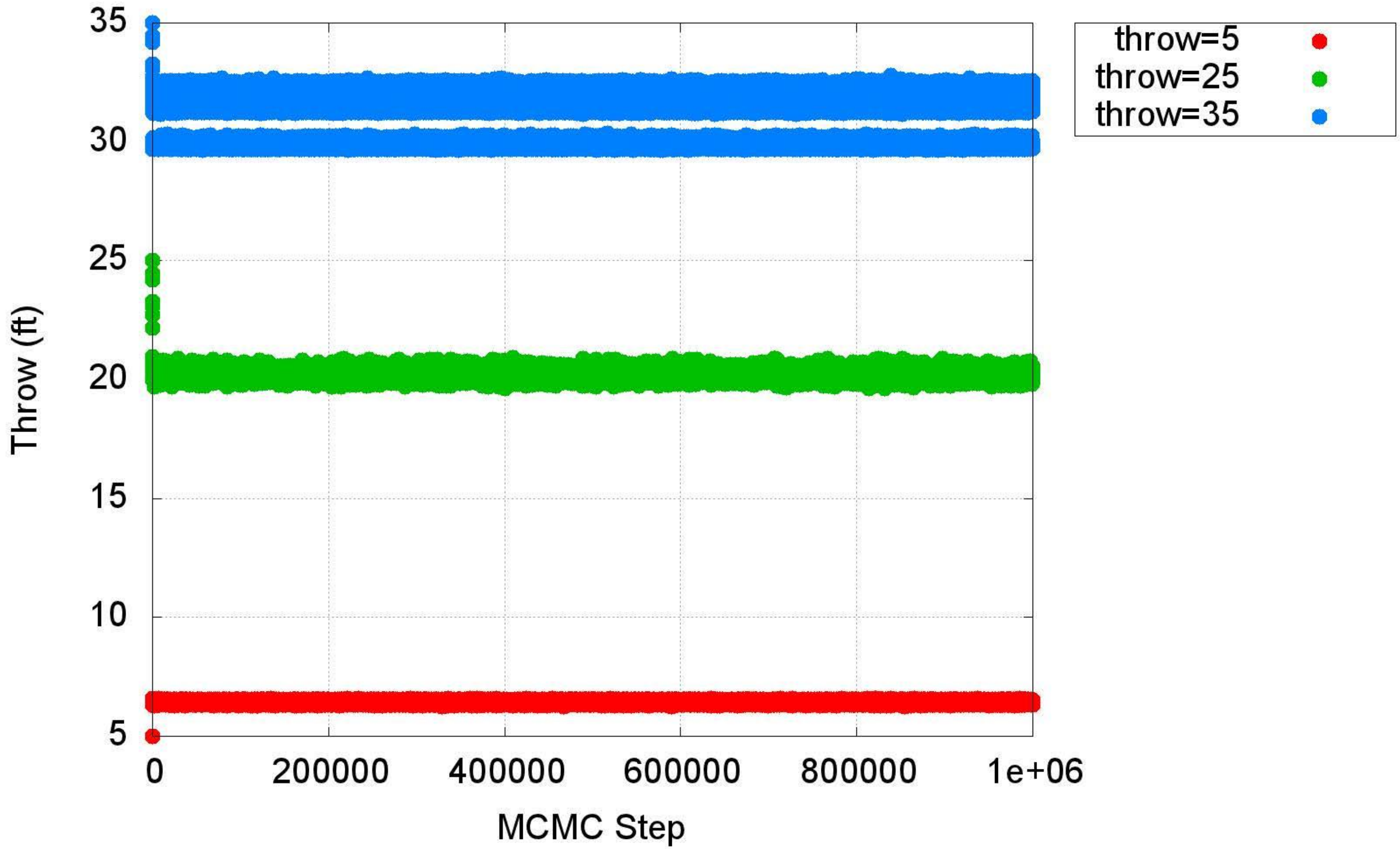
Standard Random Walk Metropolis Algorithm

$$\psi(x) = \sum_{\text{Observations}} (\text{Observation} - \text{Prediction})^2$$

$$x_{\text{prop}} = x_{\text{current}} + z \qquad z \sim N(0, \sigma^2)$$

$$\alpha = \exp\left(\min\left(0, \psi(x_{\text{current}}) - \psi(x_{\text{prop}})\right)\right)$$

Standard Random Walk Metropolis Algorithm



Our Solution: Parallel Tempering

The strength of this algorithm is its ability to explore difficult multimodal distributions

1. Run N Random Walk Metropolis algorithms in parallel

2. Each chain can be updated either using a standard proposal mechanism or swapping its state with an adjacent chain

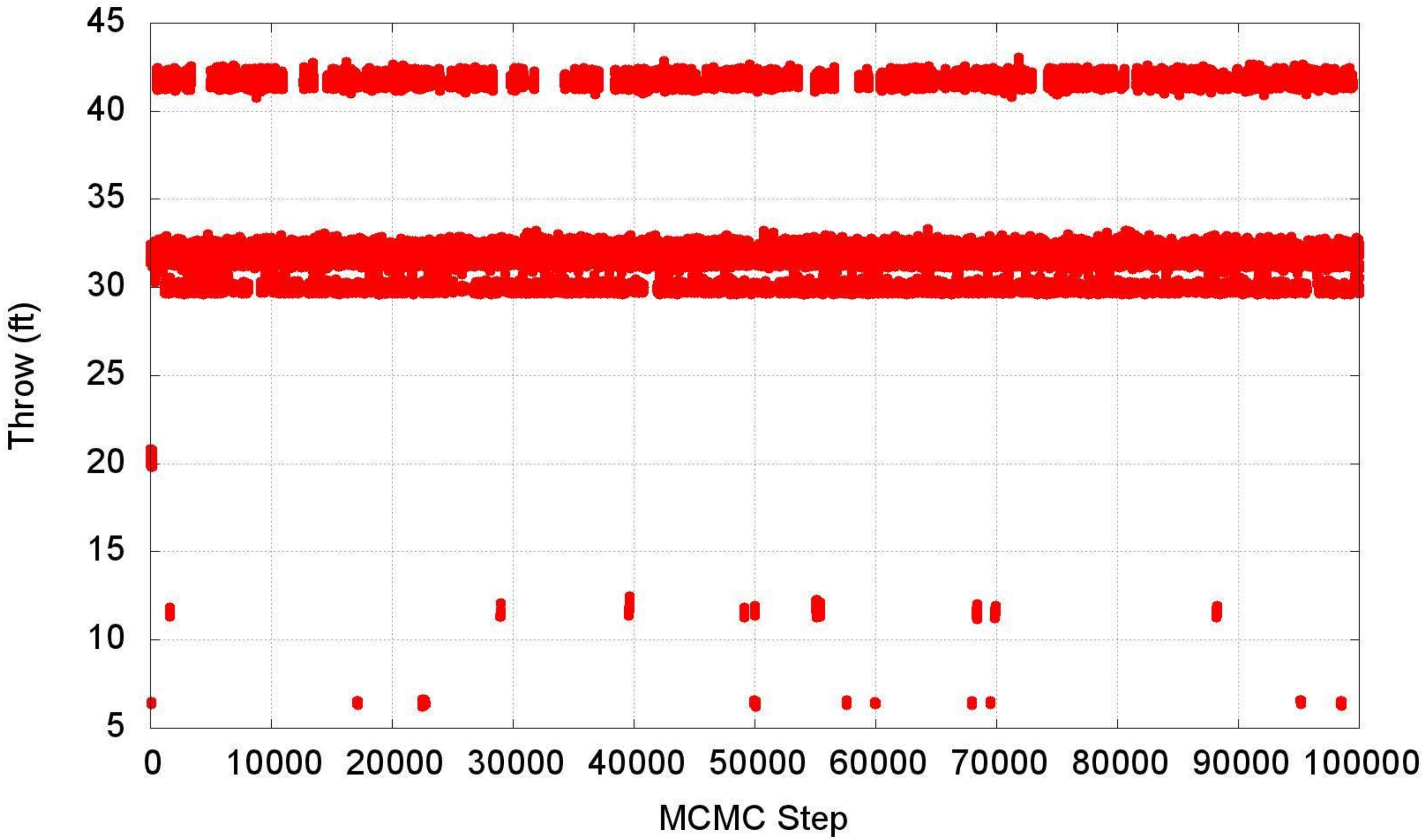
3. Acceptance probabilities are :

$$\alpha = \exp\left(\min\left(0, \frac{\psi(x_{\text{current}})}{T_i} - \frac{\psi(x_{\text{prop}})}{T_i}\right)\right)$$

or

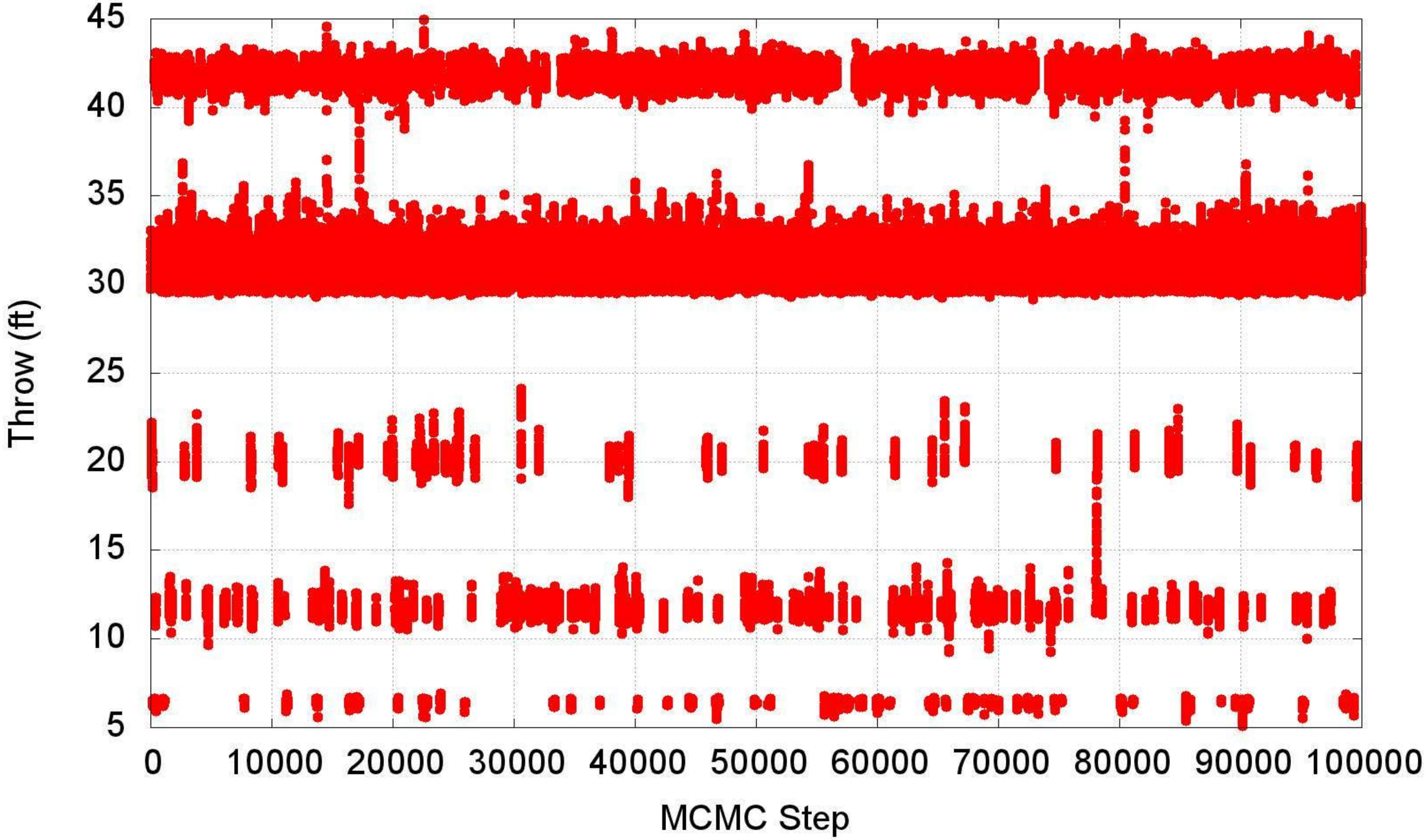
$$\alpha = \exp\left(\min\left(0, \frac{\psi(x_i) - \psi(x_j)}{T_i} + \frac{\psi(x_j) - \psi(x_i)}{T_j}\right)\right)$$

Our Solution: Parallel Tempering



(low temperature solutions)

Our Solution: Parallel Tempering



(high temperature solutions)

The Algorithm Details: temperature distribution

1. Minimum temperature – 1
2. Maximum temperature – average ψ across a random selection of states
3. Exponential distribution of temperatures across the RWM chains

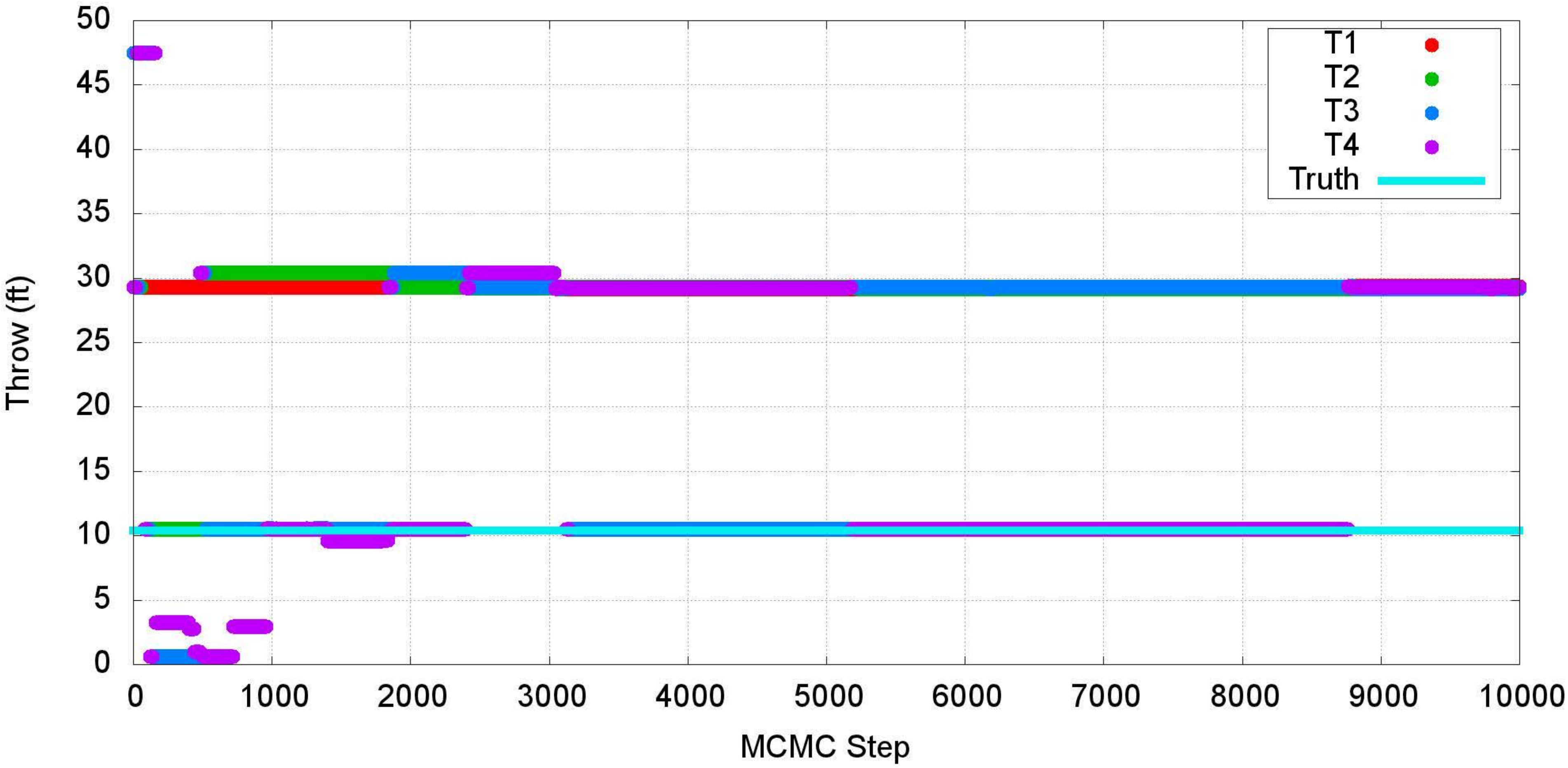
The Algorithm Details: step size distribution

1. Start all processors doing RWM chains at the highest temperature
2. Adjust step size until average acceptance probability is approximately 0.7
3. Reduce the temperature to the next highest value
4. If the acceptance probability is above 0.7 keep the step length, otherwise reduce the step length by half
5. Repeat step 4 until acceptable step length found, then go to step 3

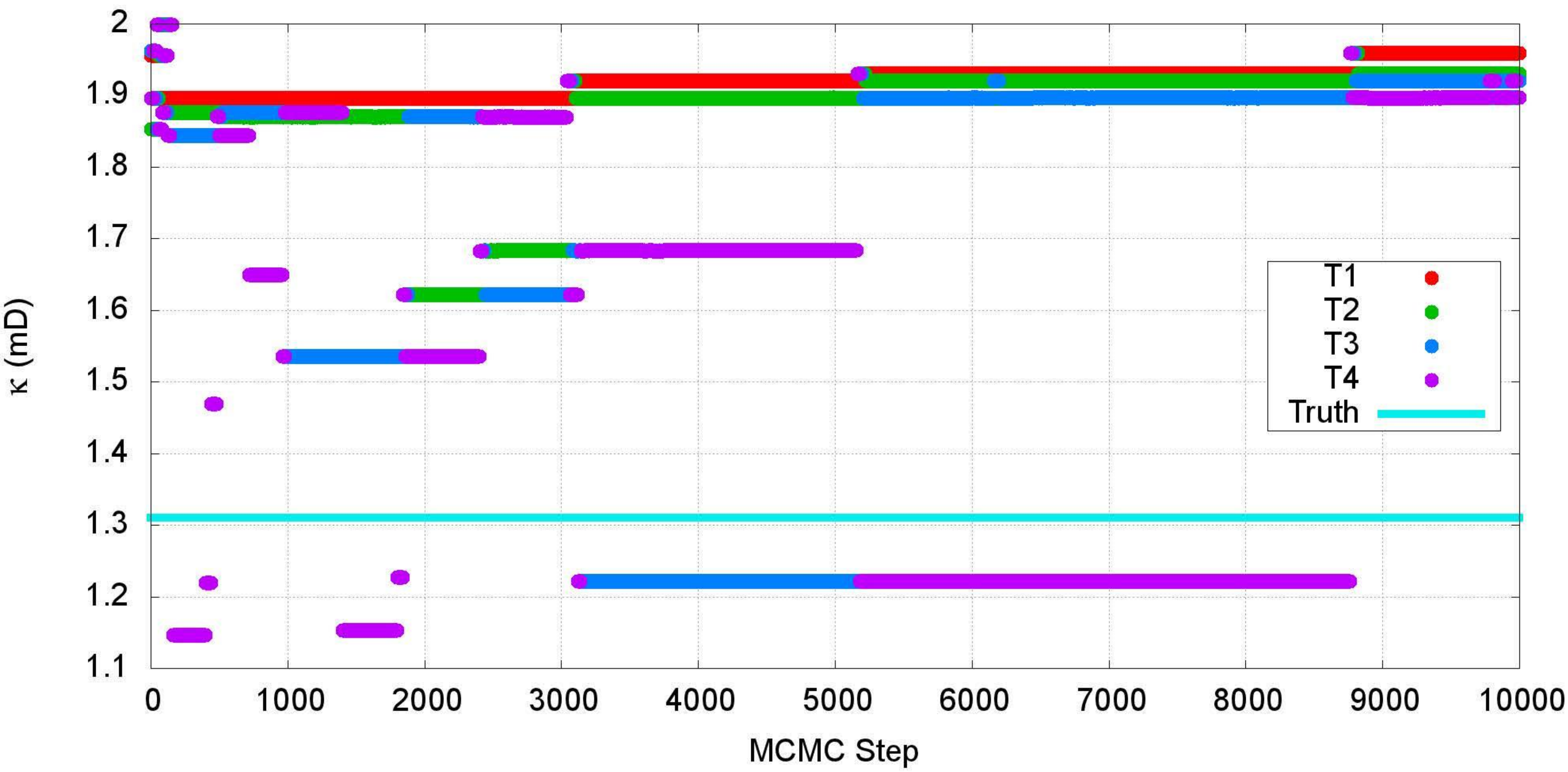
Three Parameter Model

Parameter	Min	Max	Truth
Throw (ft)	0.0	60.0	10.4
K_{SAND} (mD)	0.0	1000.0	131.7
K_{SHALE} (mD)	1.0	2.0	1.31

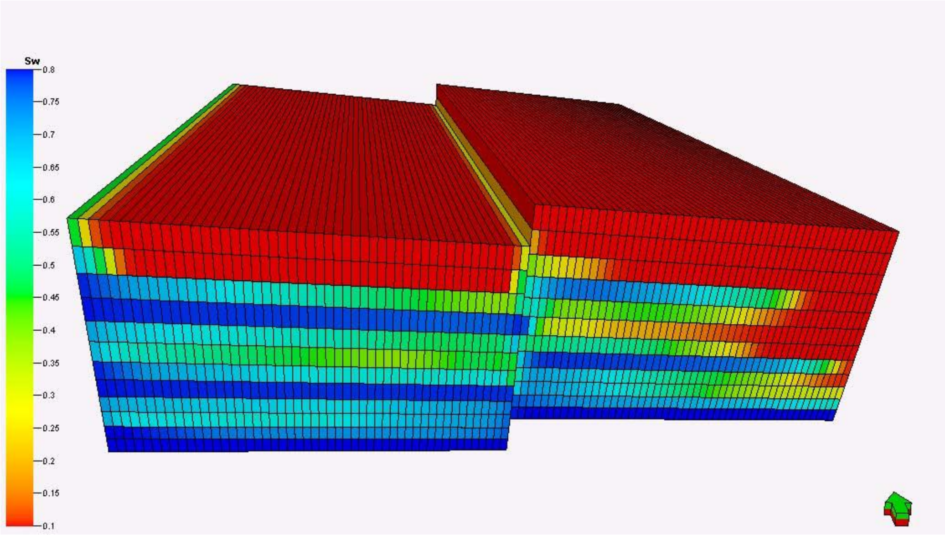
Three Parameter Problem With Noise

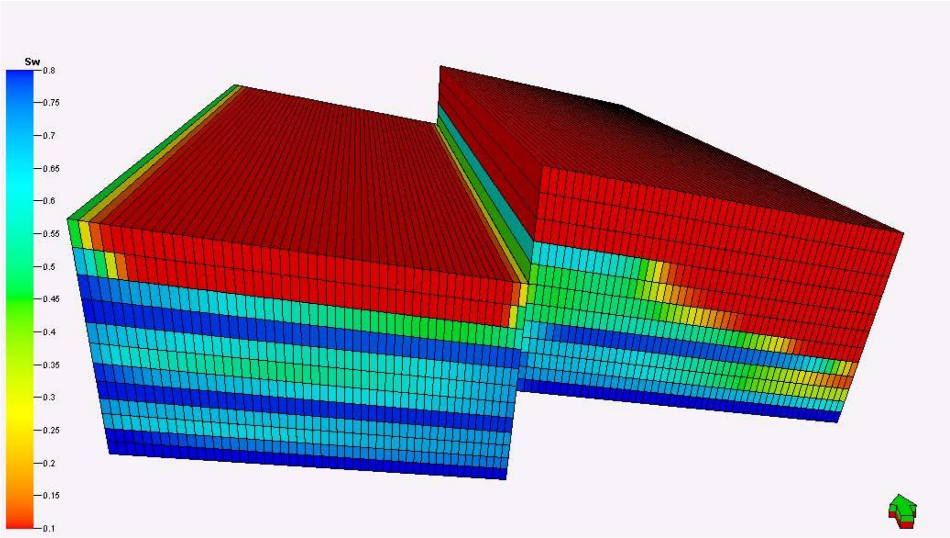


Three Parameter Problem With Noise

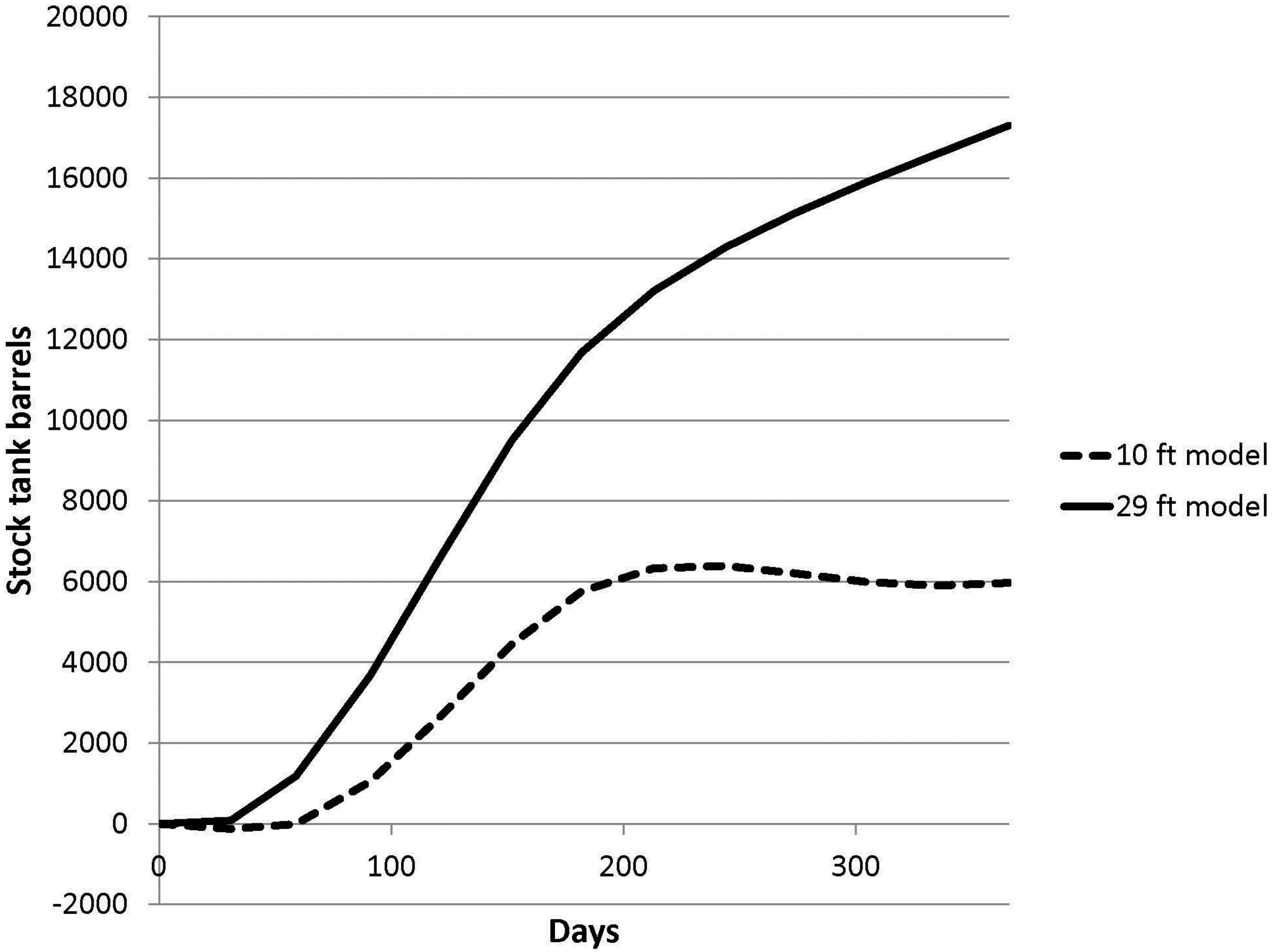


Physical Representation of the Solutions

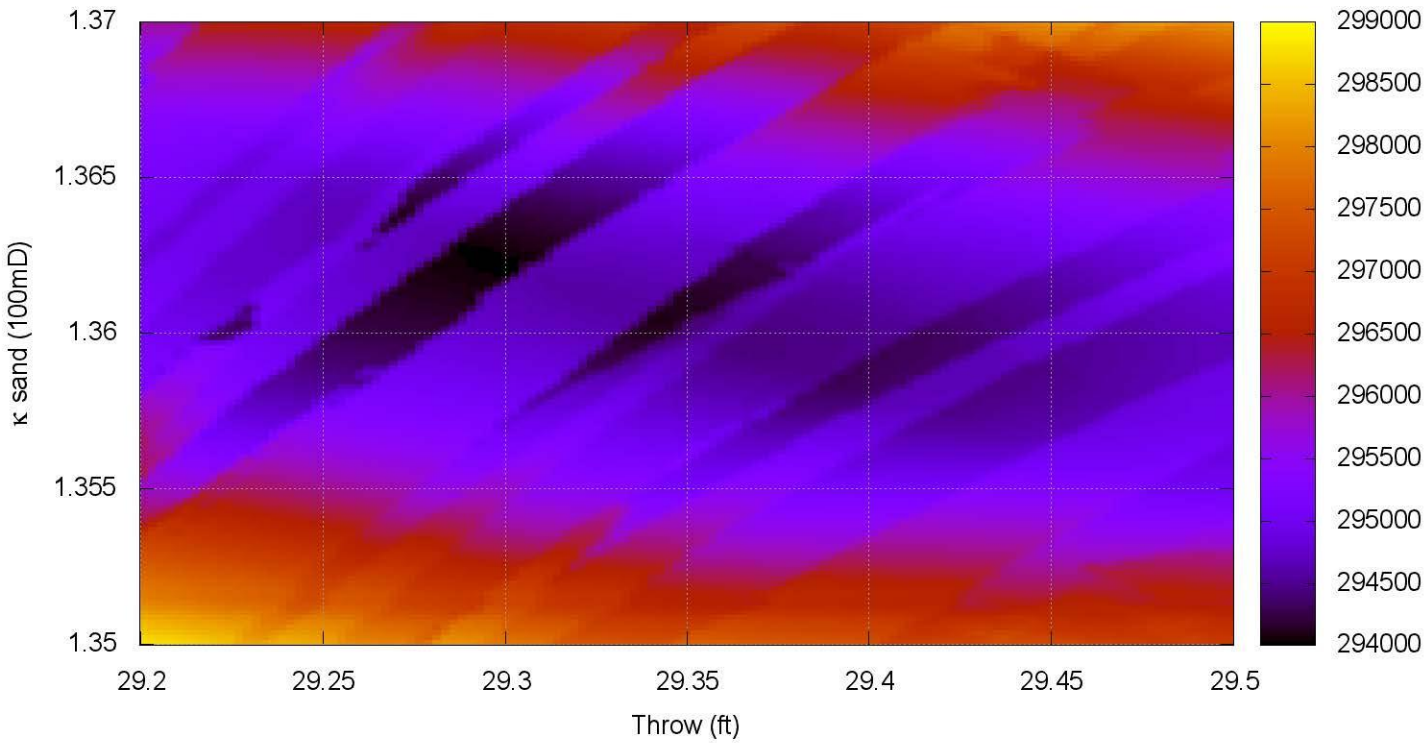




Importance of the Error



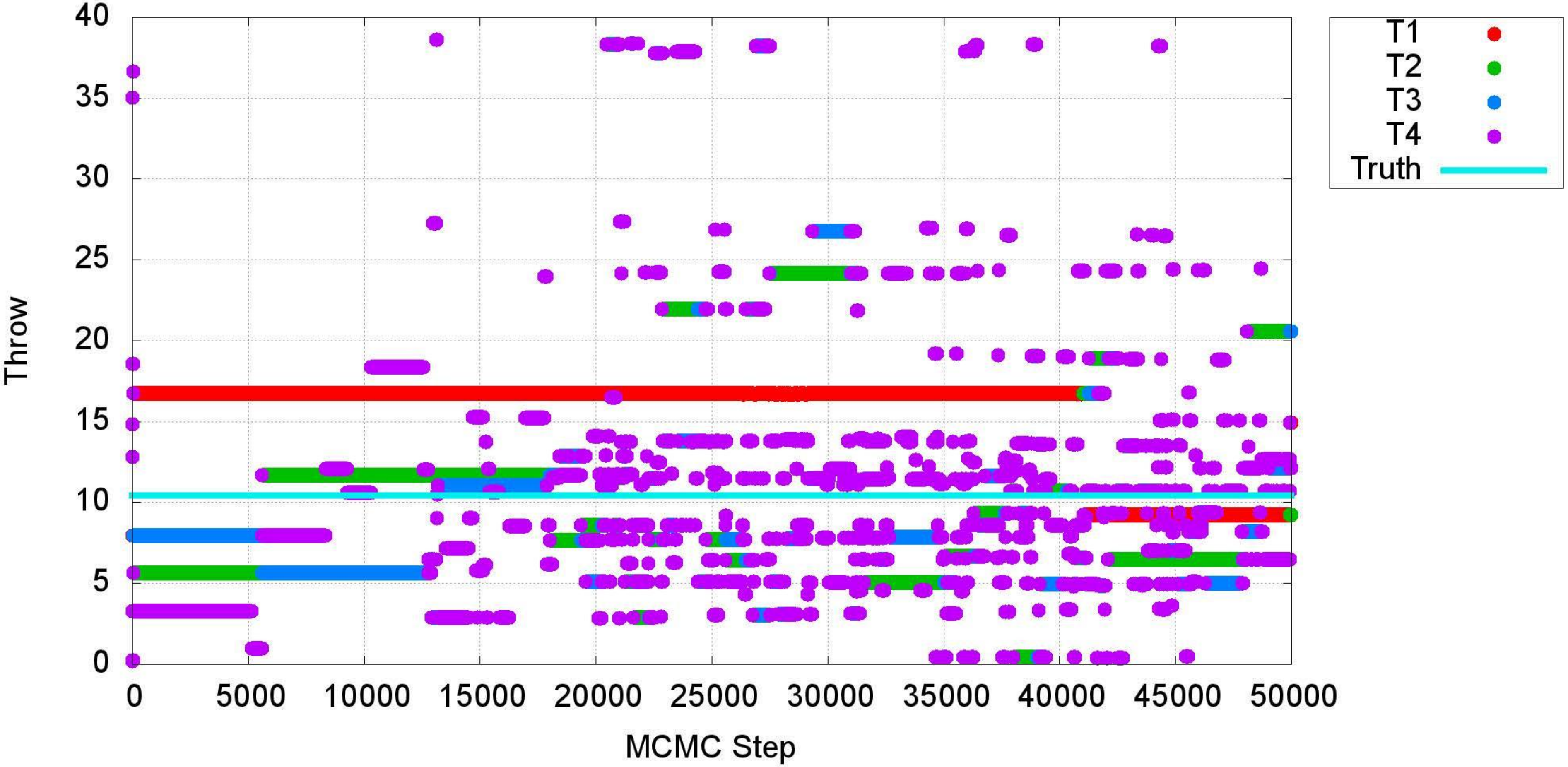
Smoothness of the Function



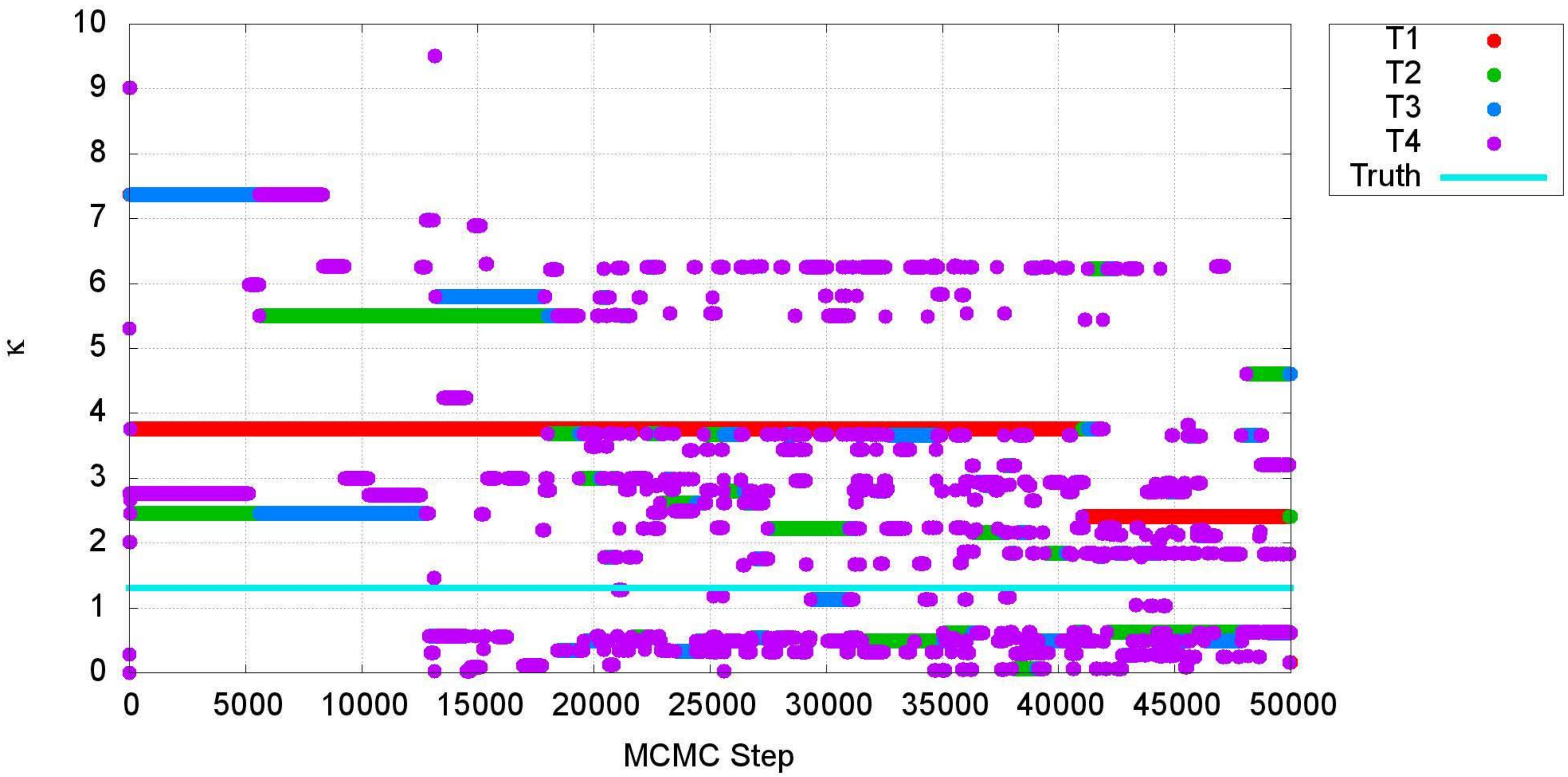
Thirteen Parameter Model

Parameter	Min	Max	Truth
Throw	0	60	10.4
Perm L1	0	10	1.31
Perm L2	100	200	131.7
Perm L3	0	10	1.31
Perm L4	100	200	131.7
Perm L5	0	10	1.31
Perm L6	100	200	131.7
Poro L1	0.1	1.0	0.15
Poro L2	0.1	1.0	0.30
Poro L3	0.1	1.0	0.15
Poro L4	0.1	1.0	0.30
Poro L5	0.1	1.0	0.15
Poro L6	0.1	1.0	0.30

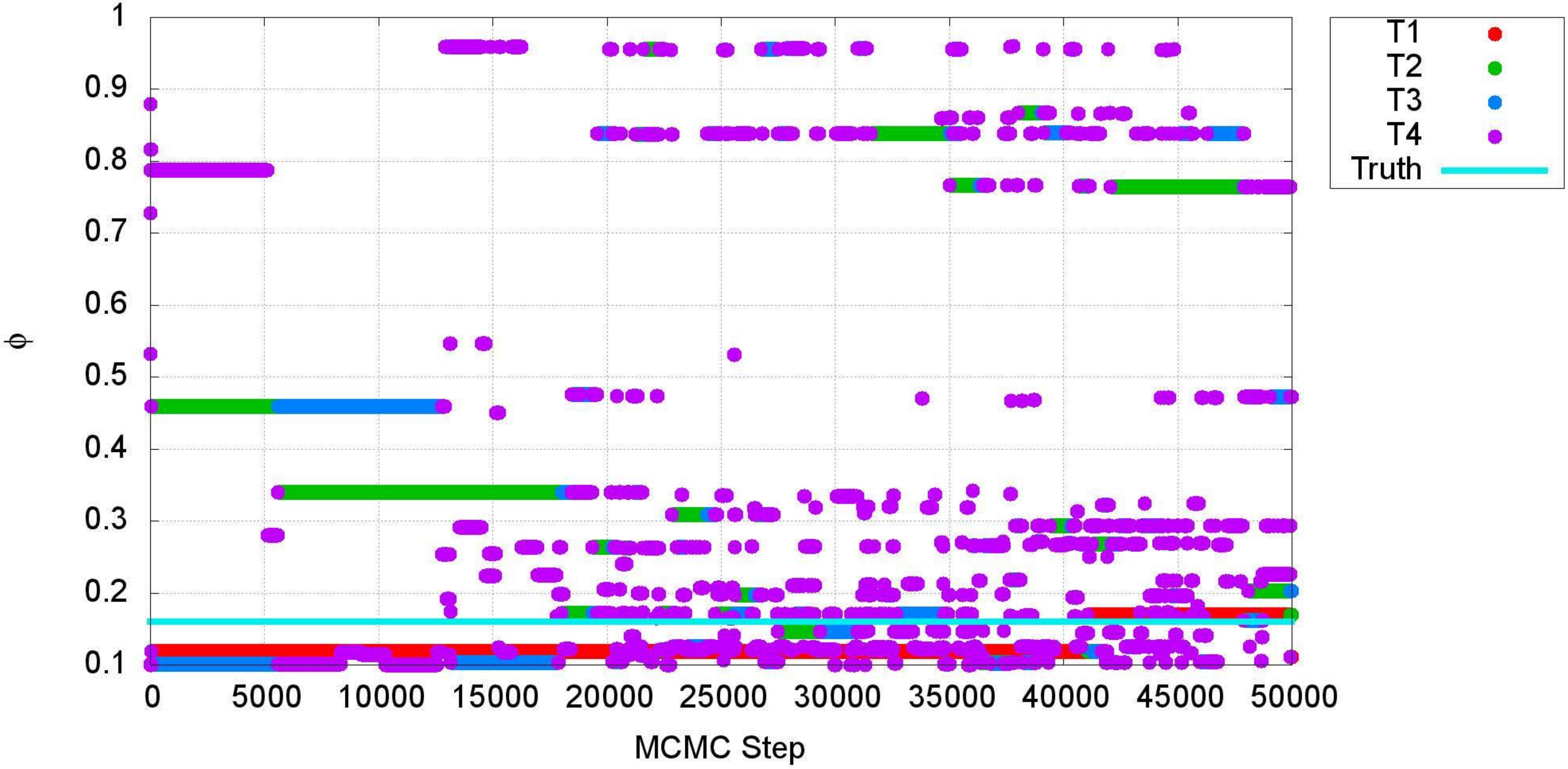
Thirteen Parameter Model



Thirteen Parameter Model



Thirteen Parameter Model



Conclusions

1. Parallel Tempering is the first algorithm I think I can trust to find enough solutions to the problem.

2. Either I find a few solutions that I can work with, or so many that I know that the available data and knowledge is insufficient.

3. The algorithm is still expensive.

Comput Geosci (2013) 17:43–65
DOI 10.1007/s10596-012-9313-3

ORIGINAL PAPER

History matching on the Imperial College fault model using parallel tempering

J. N. Carter · D. A. White

Received: 2 May 2012 / Accepted: 17 August 2012 / Published online: 30 September 2012

Abstract The history-matching inverse problem from petroleum engineering is analysed using the Imperial tive function is highly nonlinear, expensive to evaluate and multimodal. Previously used algorithms have