Optimising High-Dimensional Black-Box Functions with Gaussian Processes

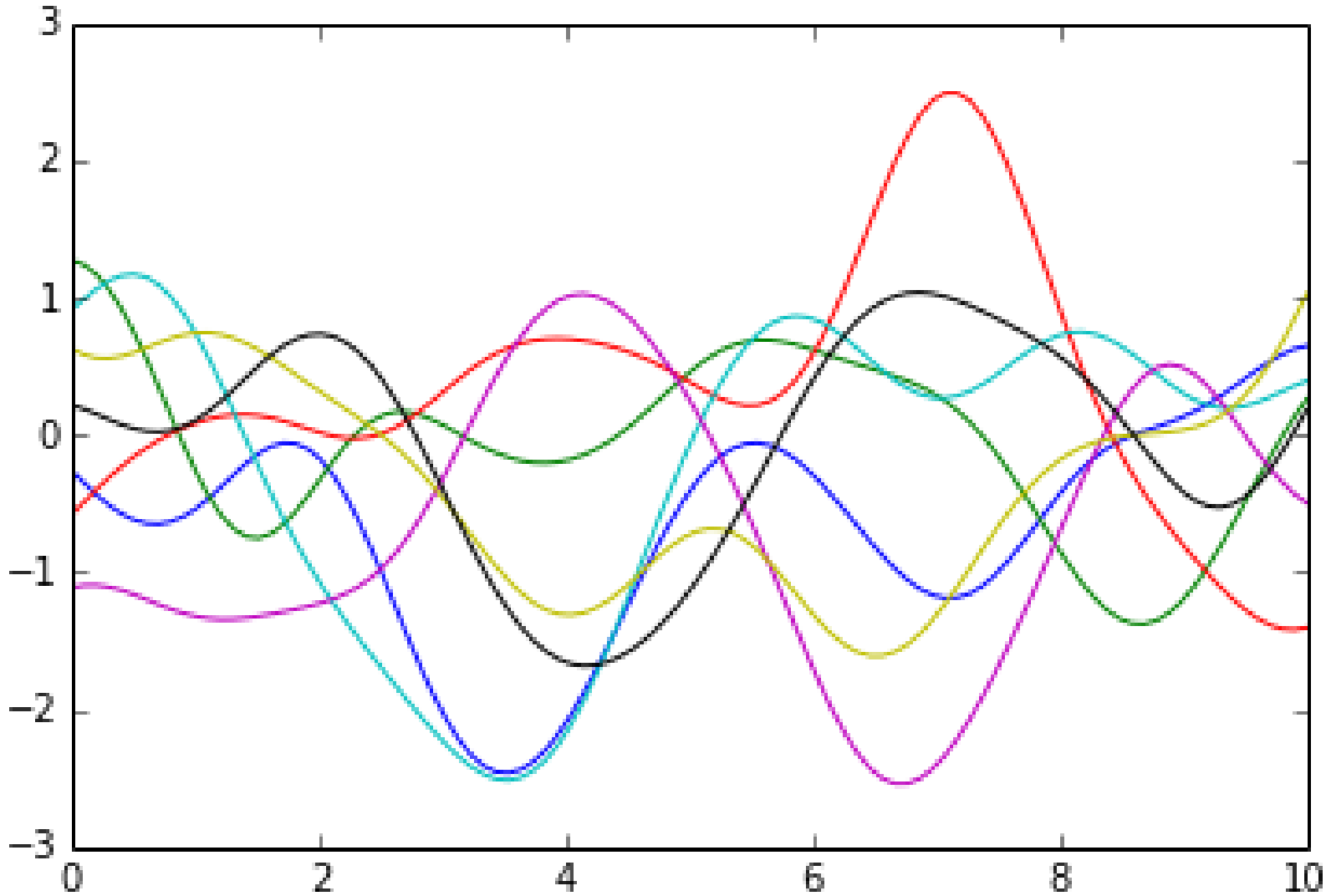
David A Roberts

12th August 2014

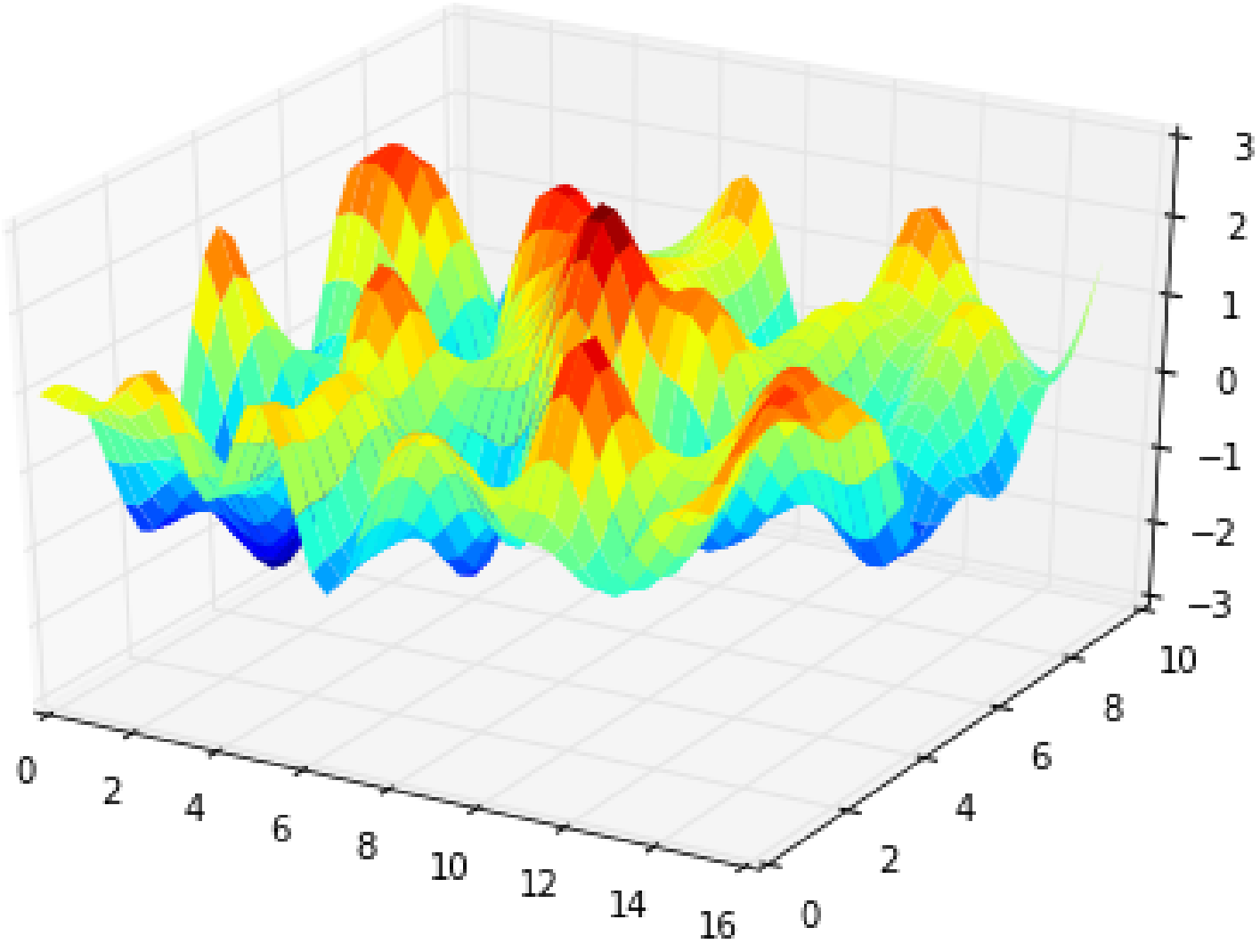
Introduction

- maximise *objective function*, by sampling its value at a number of points
- assumptions about structure of objective function ("smoothness", etc)
- goals (e.g. quality of maximum vs number of evaluations)
- Bayesian
  - assumptions = prior distribution (e.g. Gaussian process)
  - goals = utility (e.g. maximum function value found so far)
  - maximise expected utility wrt posterior
- difficulty (depends on priorities)
  - number of FEs required (tends to scale with dimensionality)
  - model-based computation

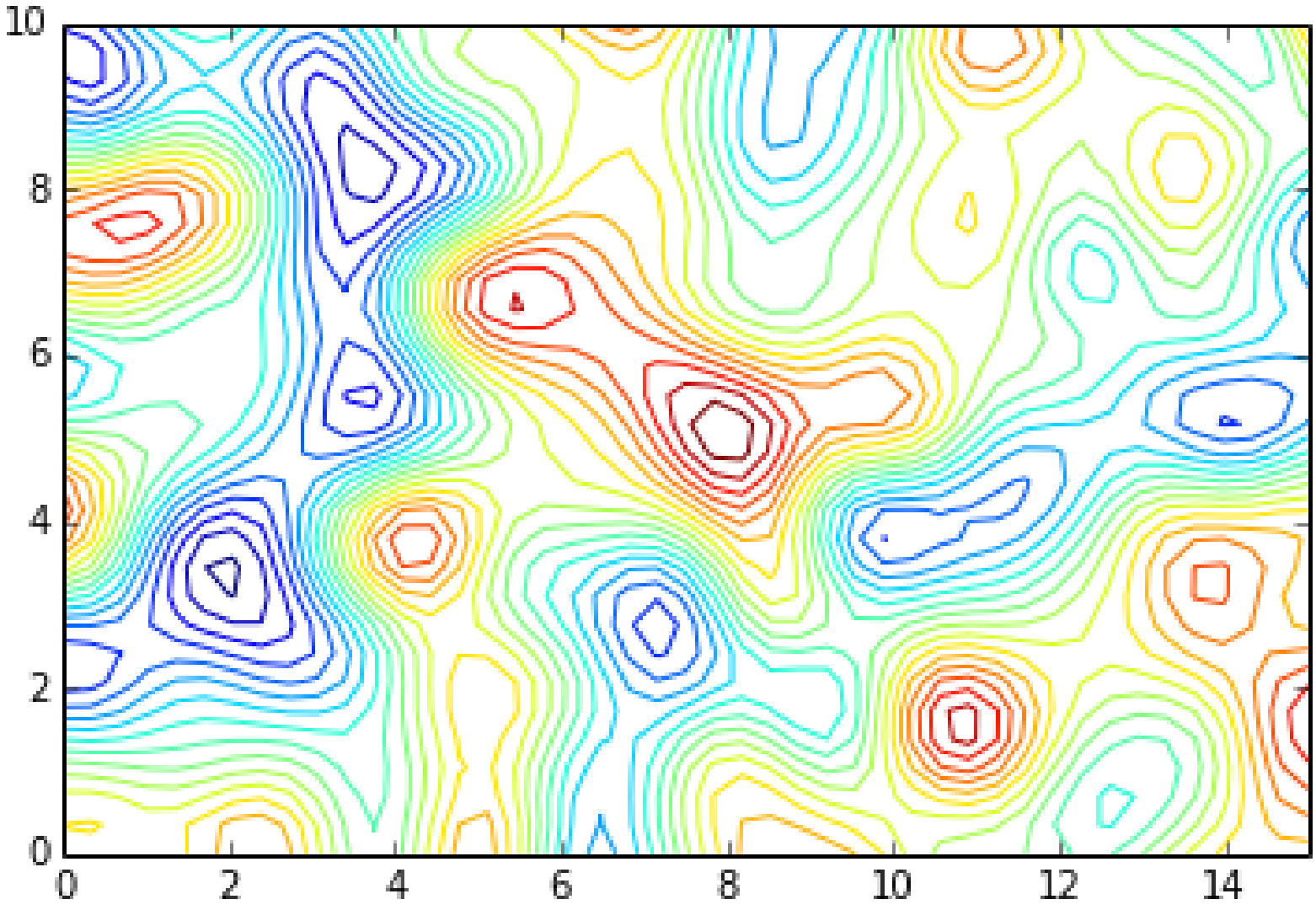
GP Prior (1D)



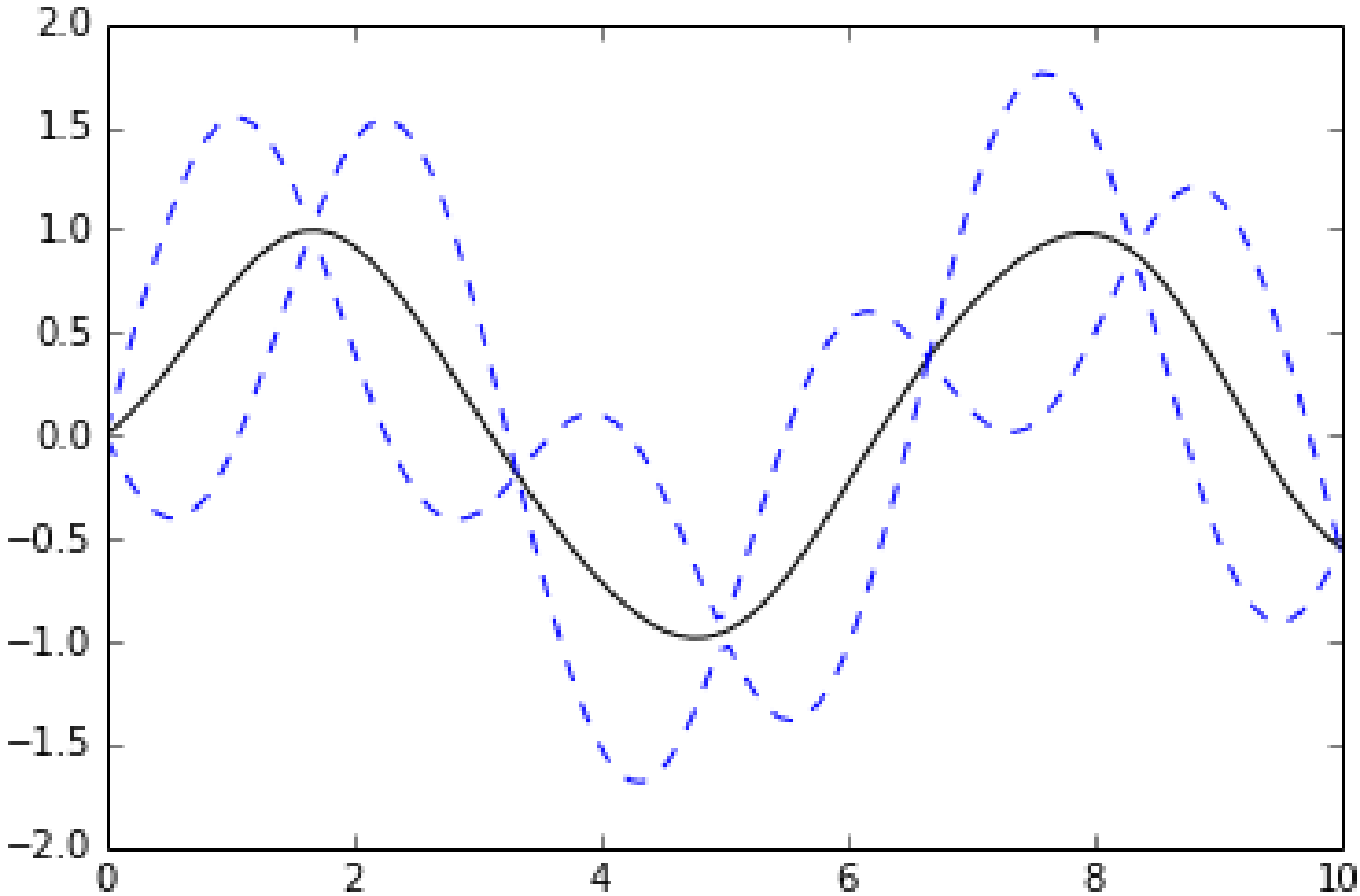
GP Prior (2D)



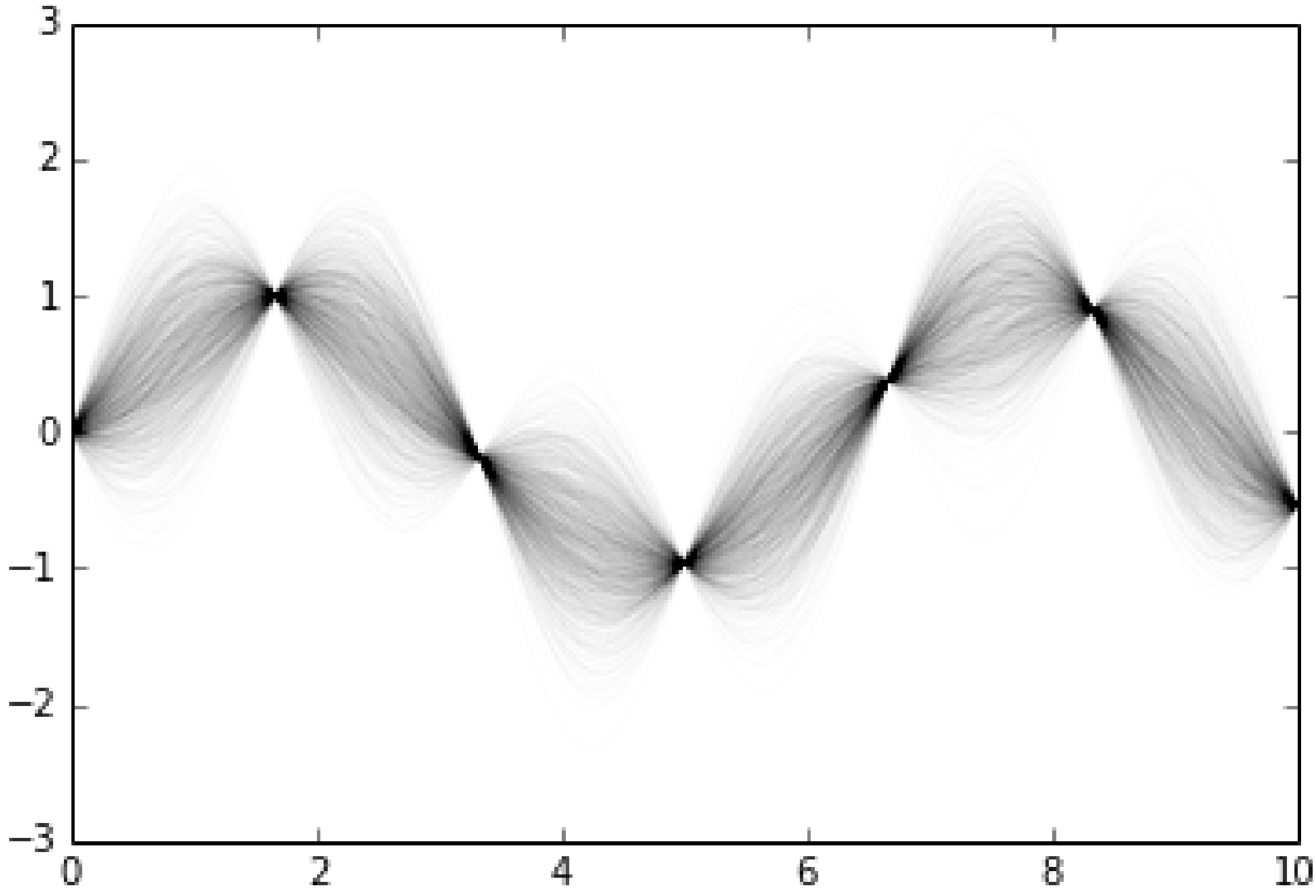
GP Prior (2D)



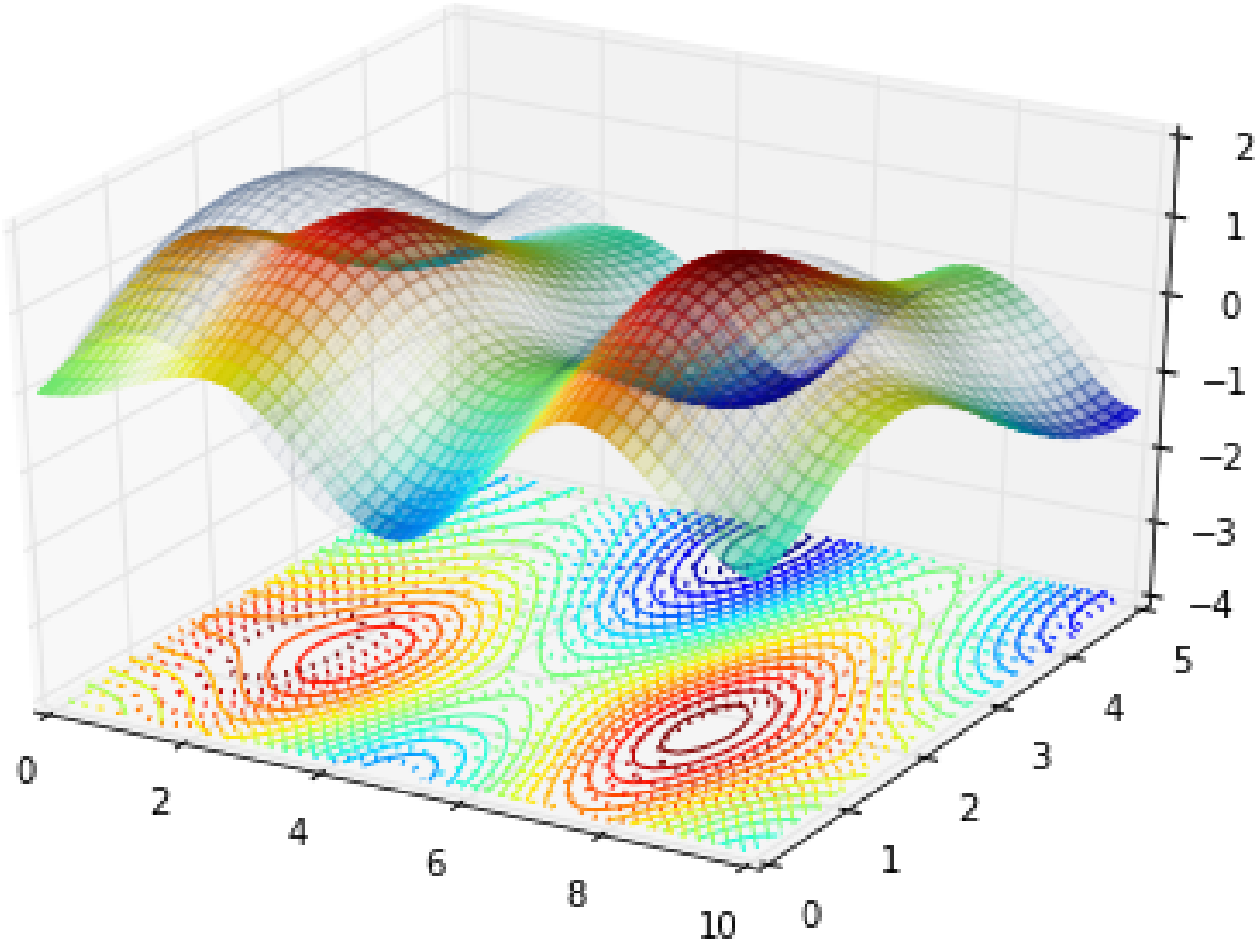
GP Posterior (1D)



GP Posterior (1D)



GP Posterior (2D)



Model

maximise	$\mathbb{E}\, U$
subject to	$f \sim \mathcal{N}(0, K_{\mathsf{SE}})$
	$U = f(x^*)$
	$y_t = f(x_t), \quad t \leq N$
with actions	$x_t \in \mathbb{R}^D, \quad t \leq N$
	$x^* \in \{x_t\}$
and observations	$y_t \in \mathbb{R}, \quad t \leq N$

Model

maximise	$\mathbb{E}\, U$
subject to	$f \sim \mathcal{N}(0, K_{\mathsf{SE}})$
	$U = f(x^*)$
	$y_t = f(x_t), \quad t \leq N$
with actions	$x_t \in \mathbb{R}^D, \quad t \leq N$
	$x^* \in \{x_t\}$
and observations	$y_t \in \mathbb{R}, \quad t \leq N$

- computational/FE costs?

Implementation

- computing Gaussian conditional distributions (inverting large covariance matrices)
- inferring unknowns (GP hyperparams — ML, MC, etc)
- approximate decision policy, selecting actions
- non-trivial, as model is quite abstract
- computation vs uncertainty

Matrix Inverse

- $\mathcal{O}(N \cdot N^3)$ total
- sparsity constraints (Markov network)
- low-rank approximations, hierarchical decomposition (n-body problem)
- active sets (ignore predictable observations)
- incremental updates

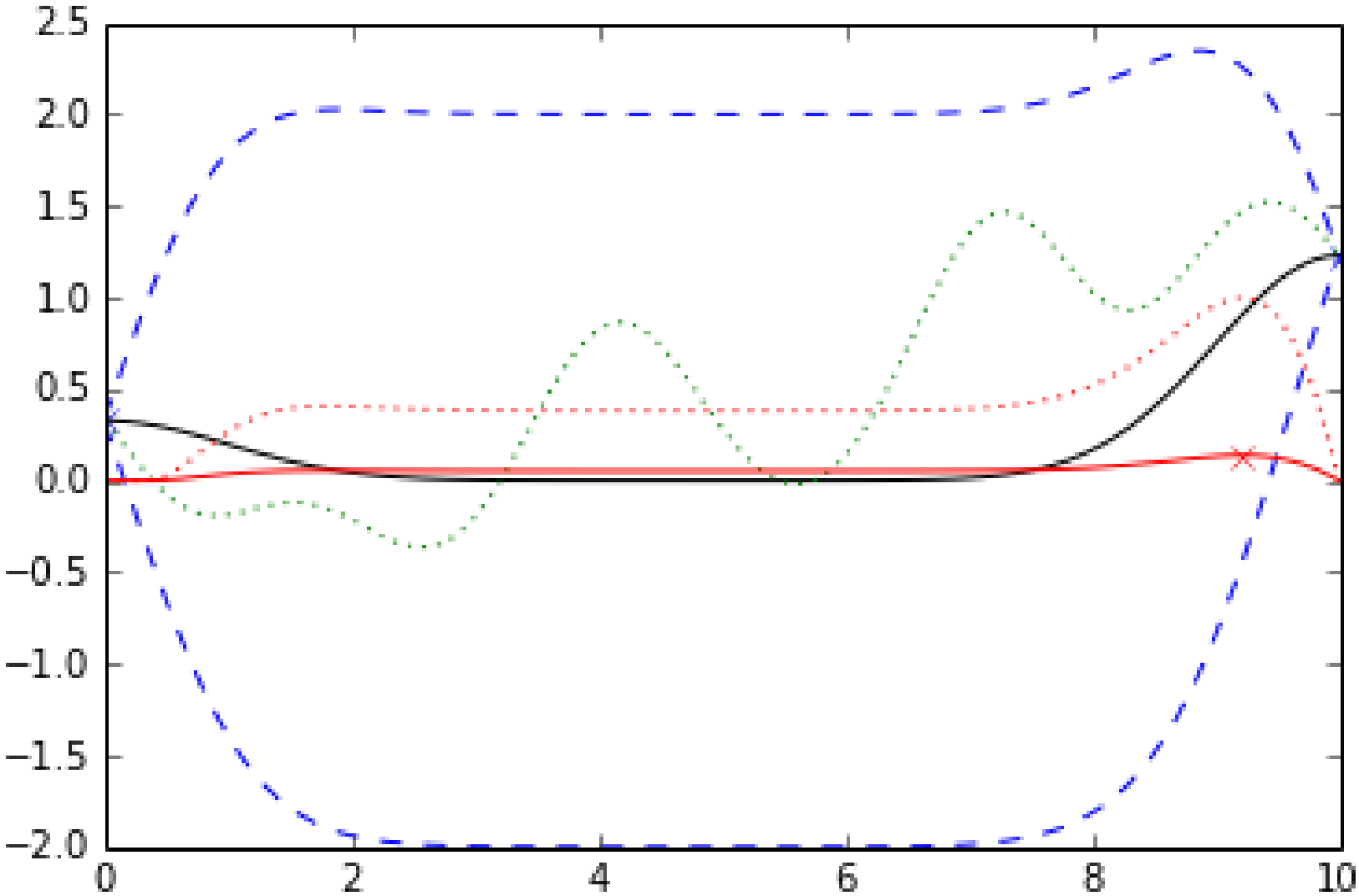
Numerical Stability

- assume more noise than expected (even if observations are noise-free)
  - numerical noise
  - cf. ridge regression

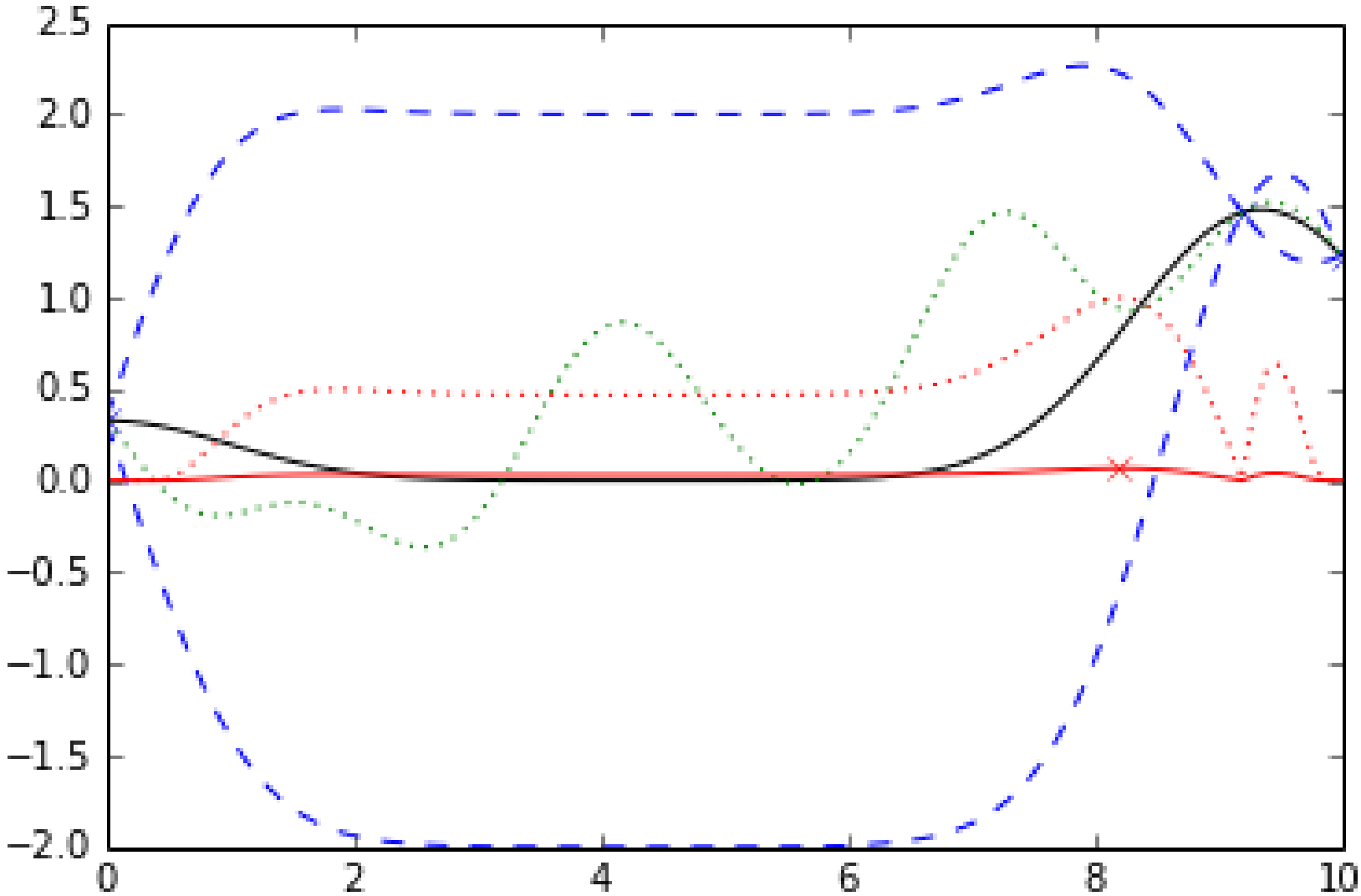
Approximate Decisions

- optimal decisions are hard
- pretend every sample is the last — maximise expected improvement
  - more accurate approximations possible
- optimiser within an optimiser
  - general-purpose: DIRECT, CG, CMA-ES, etc
  - but EI isn't a black-box
  - same dimensionality as objective function

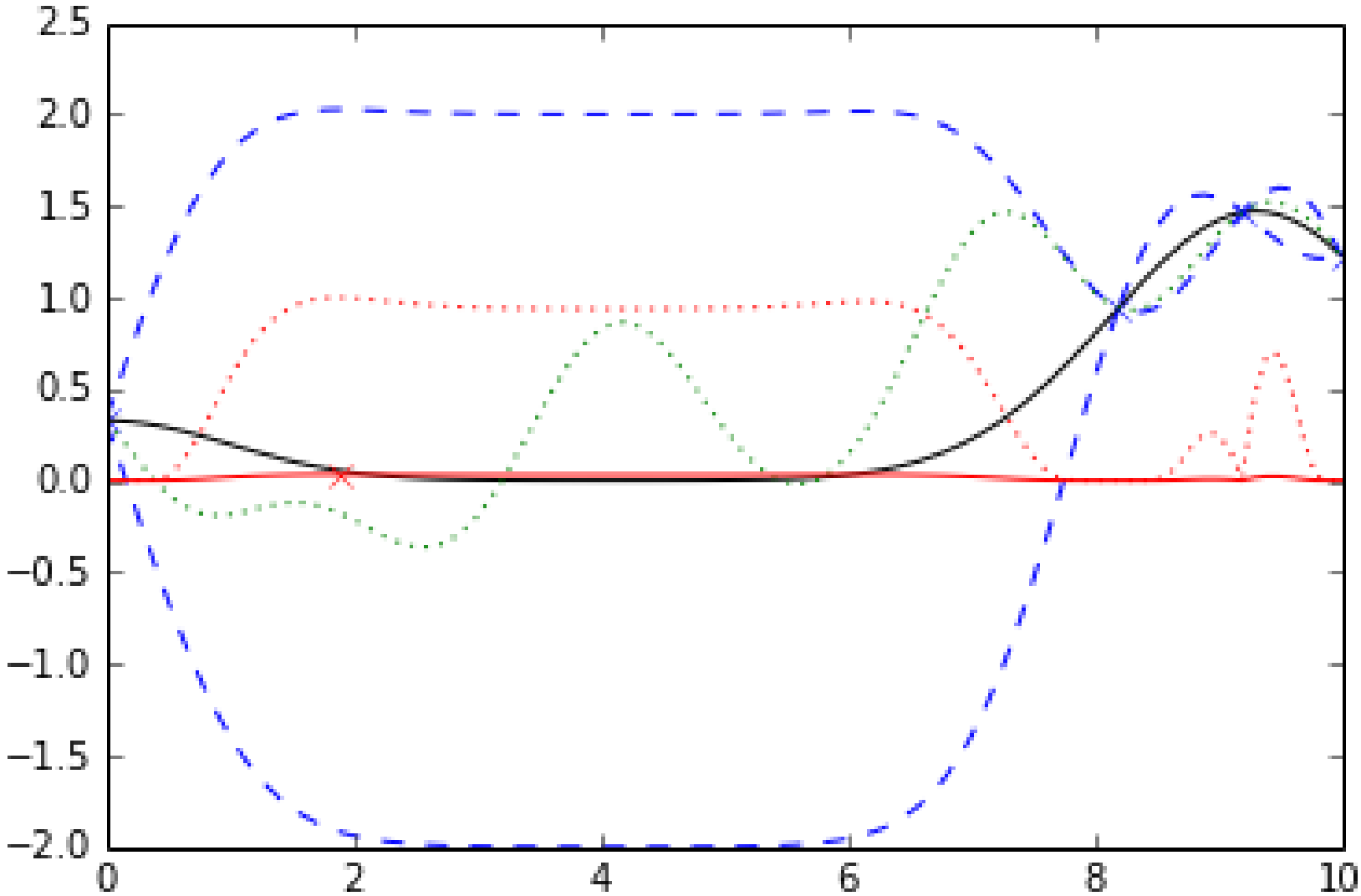
GPO (1D)



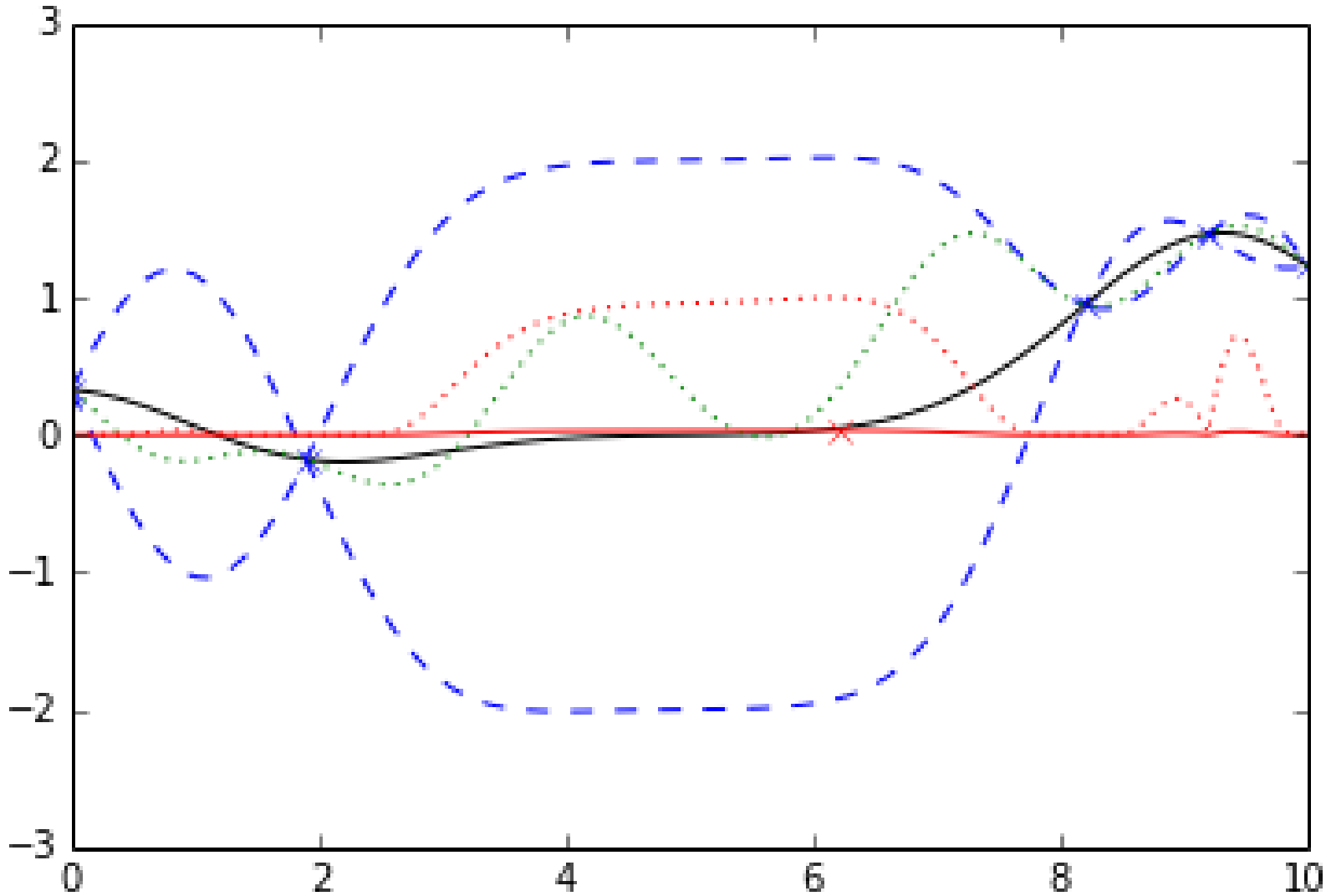
GPO (1D)



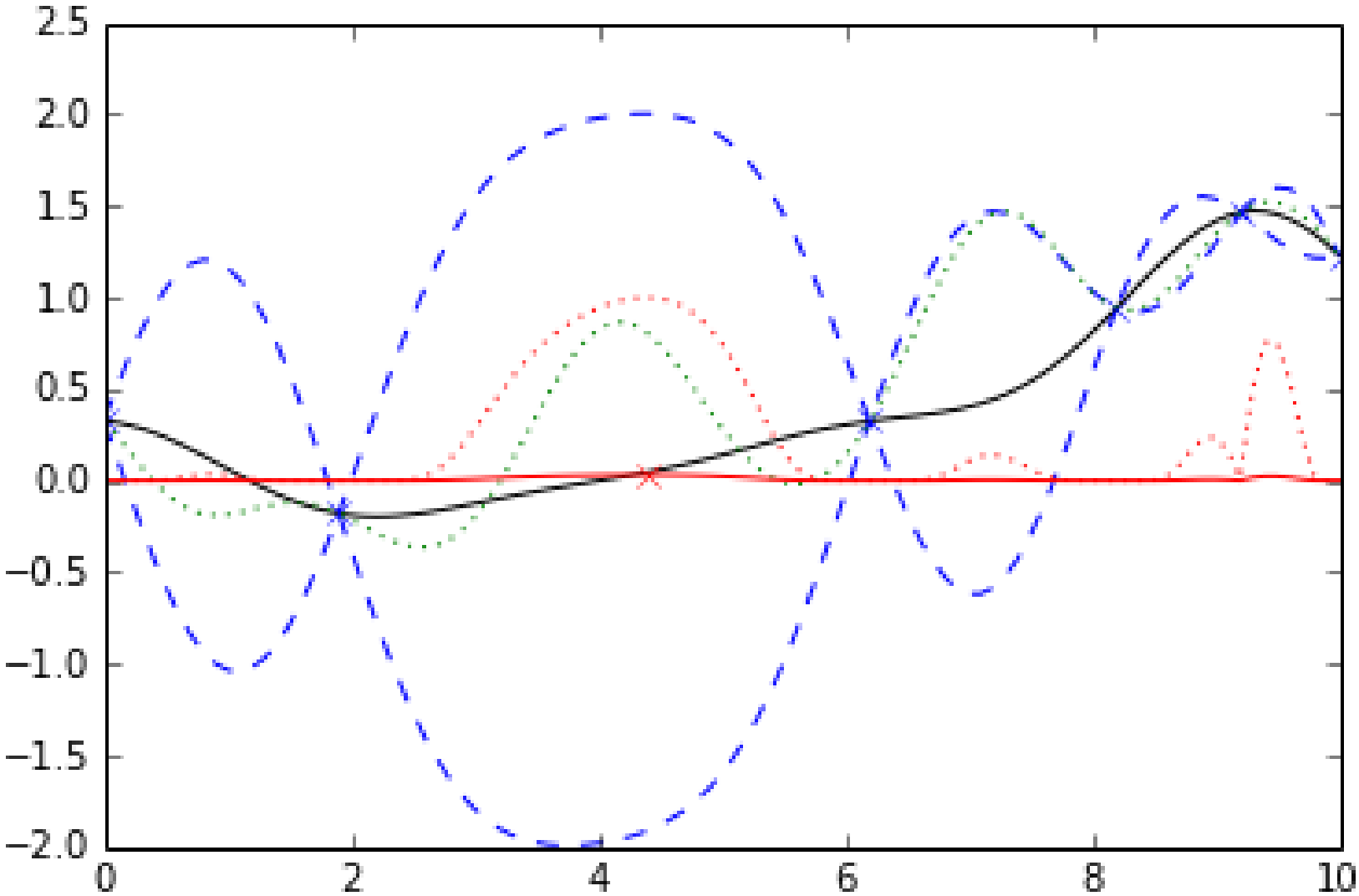
GPO (1D)



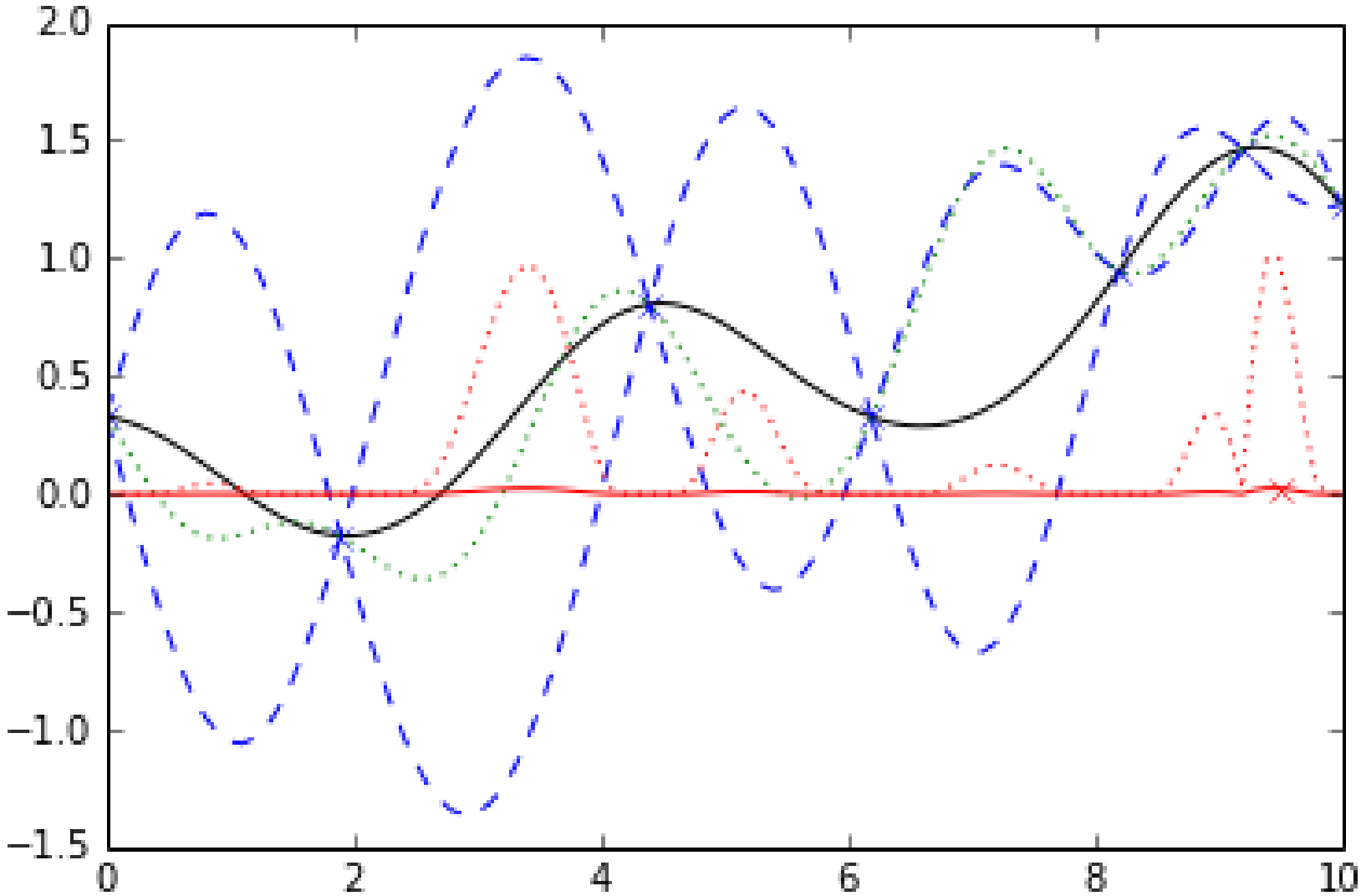
GPO (1D)



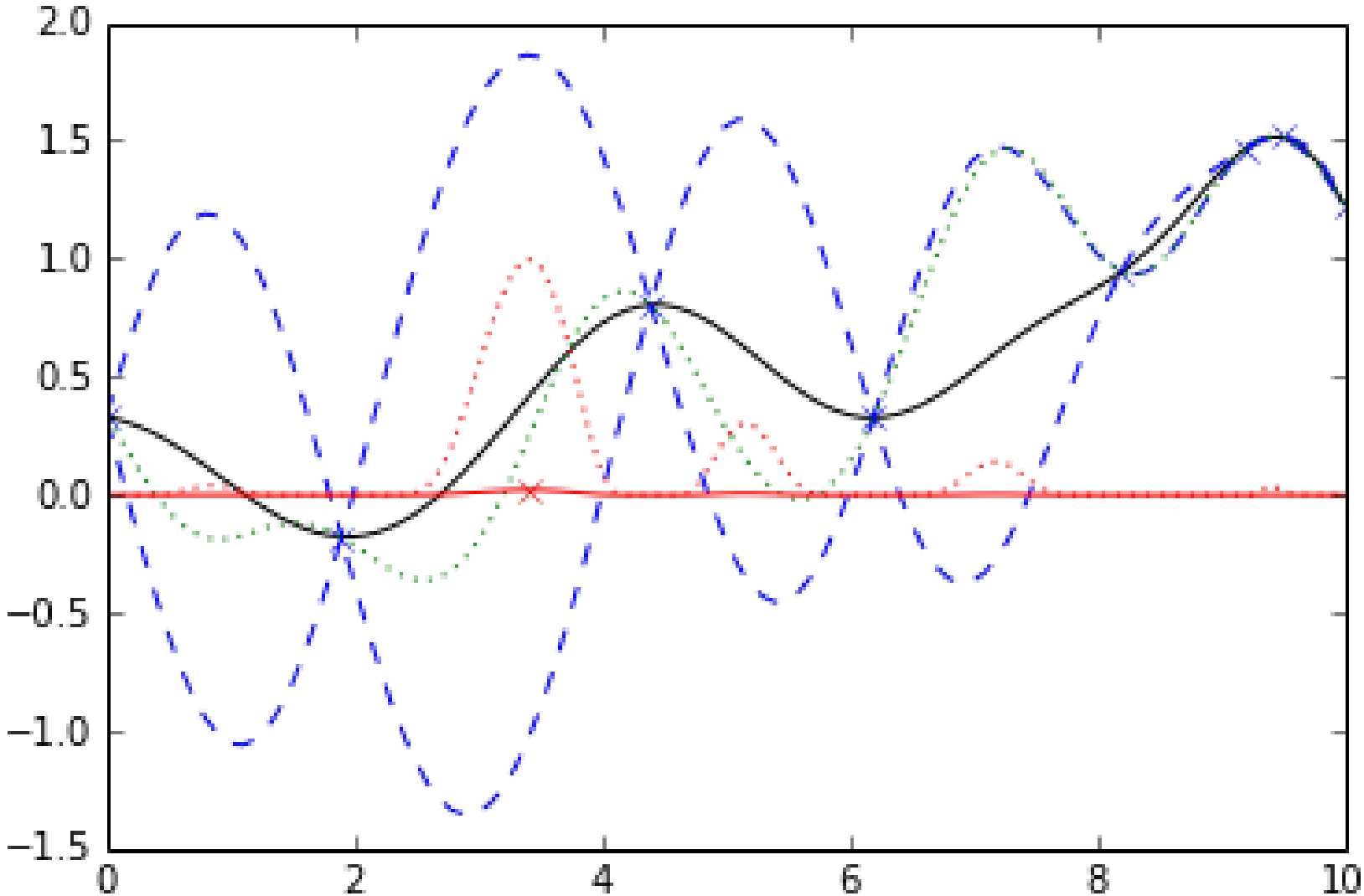
GPO (1D)



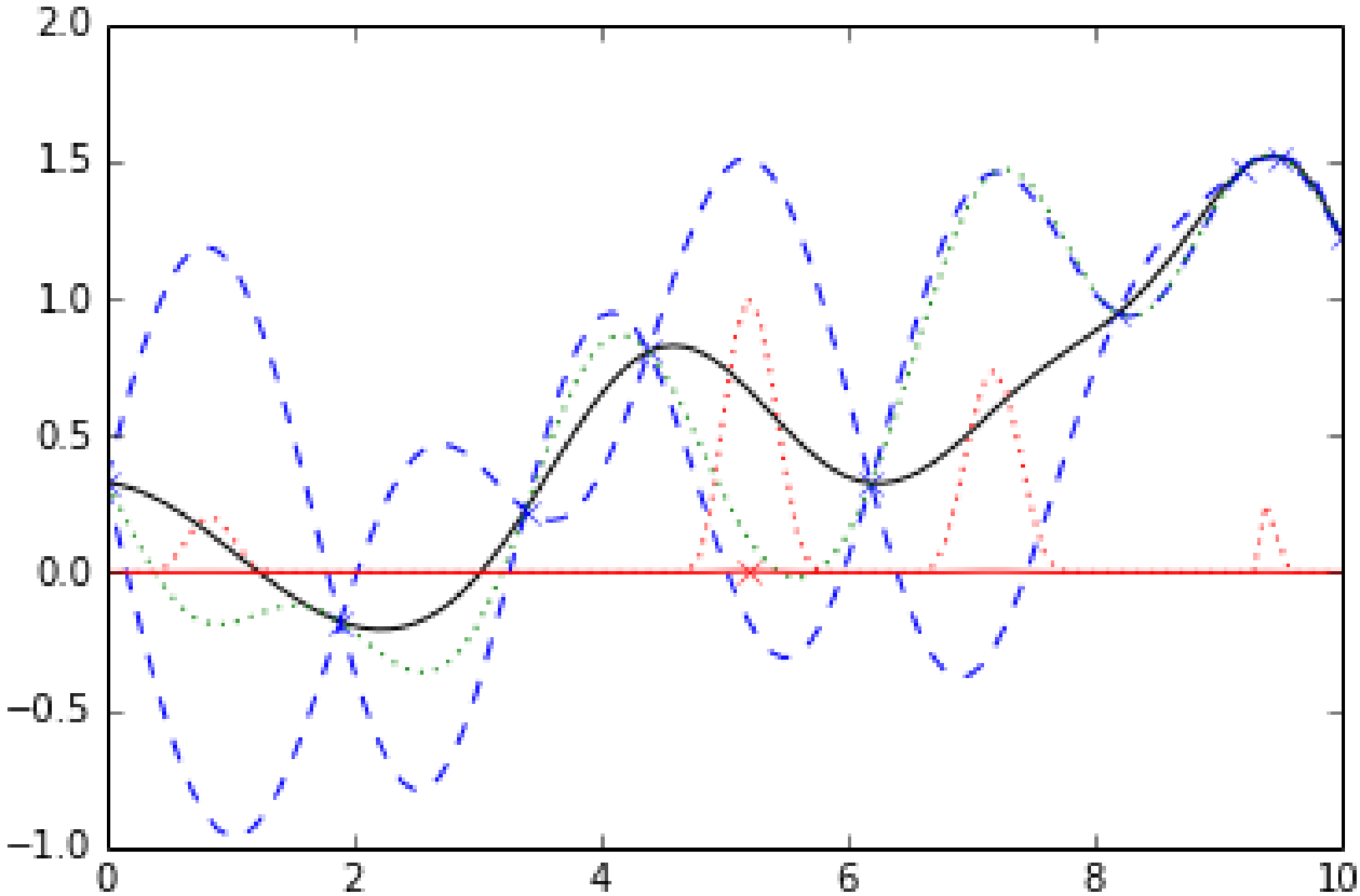
GPO (1D)



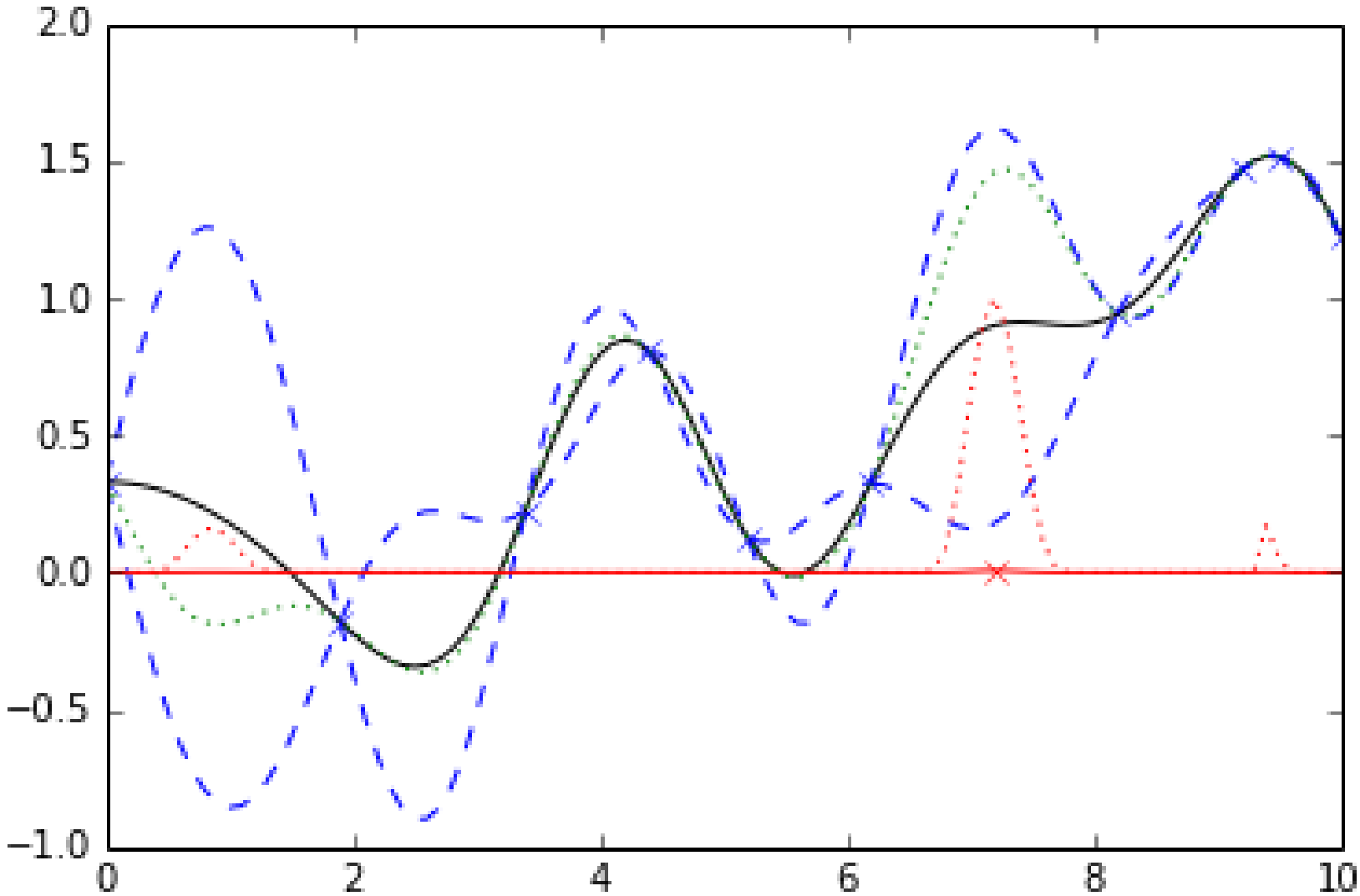
GPO (1D)



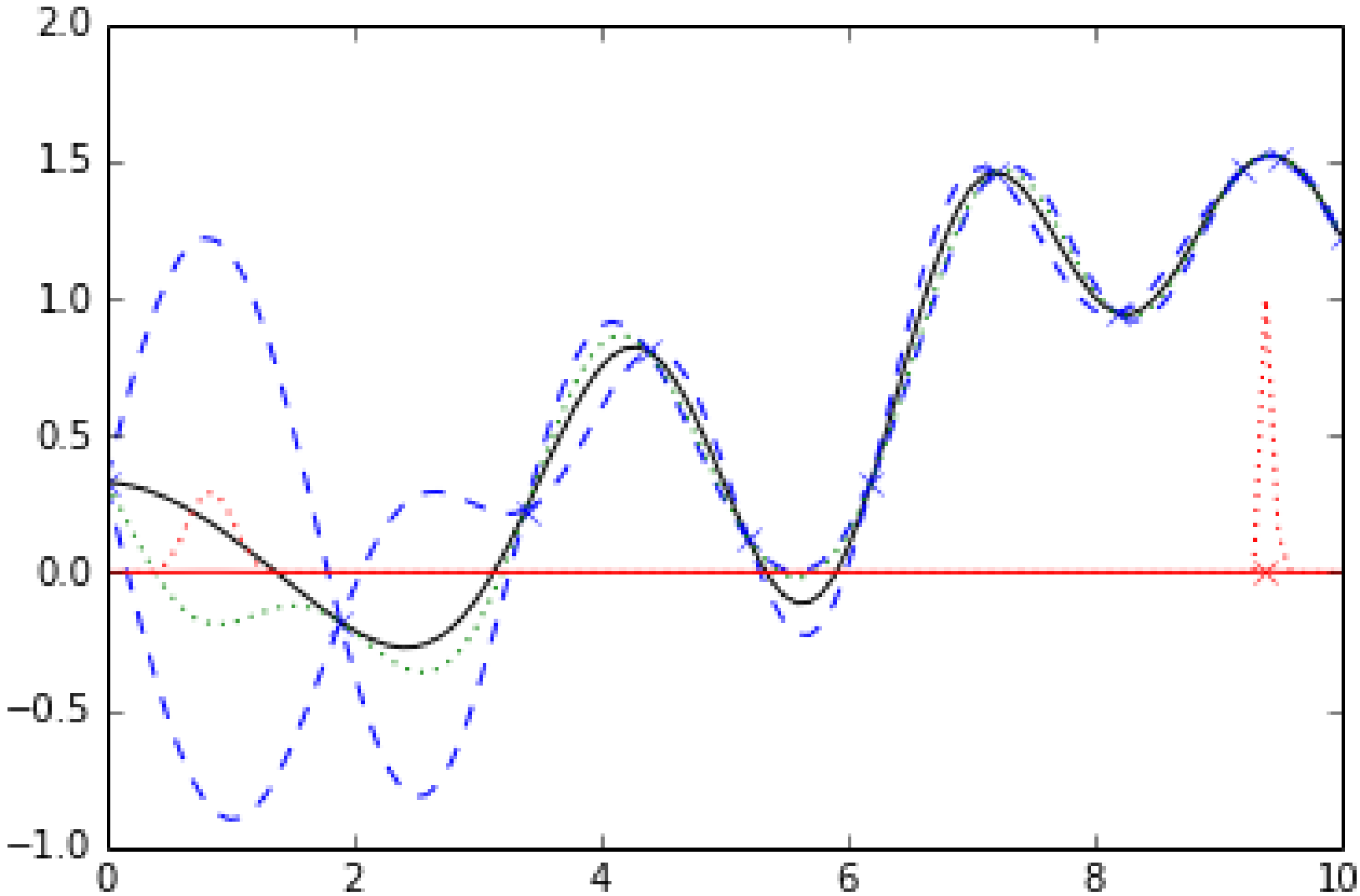
GPO (1D)



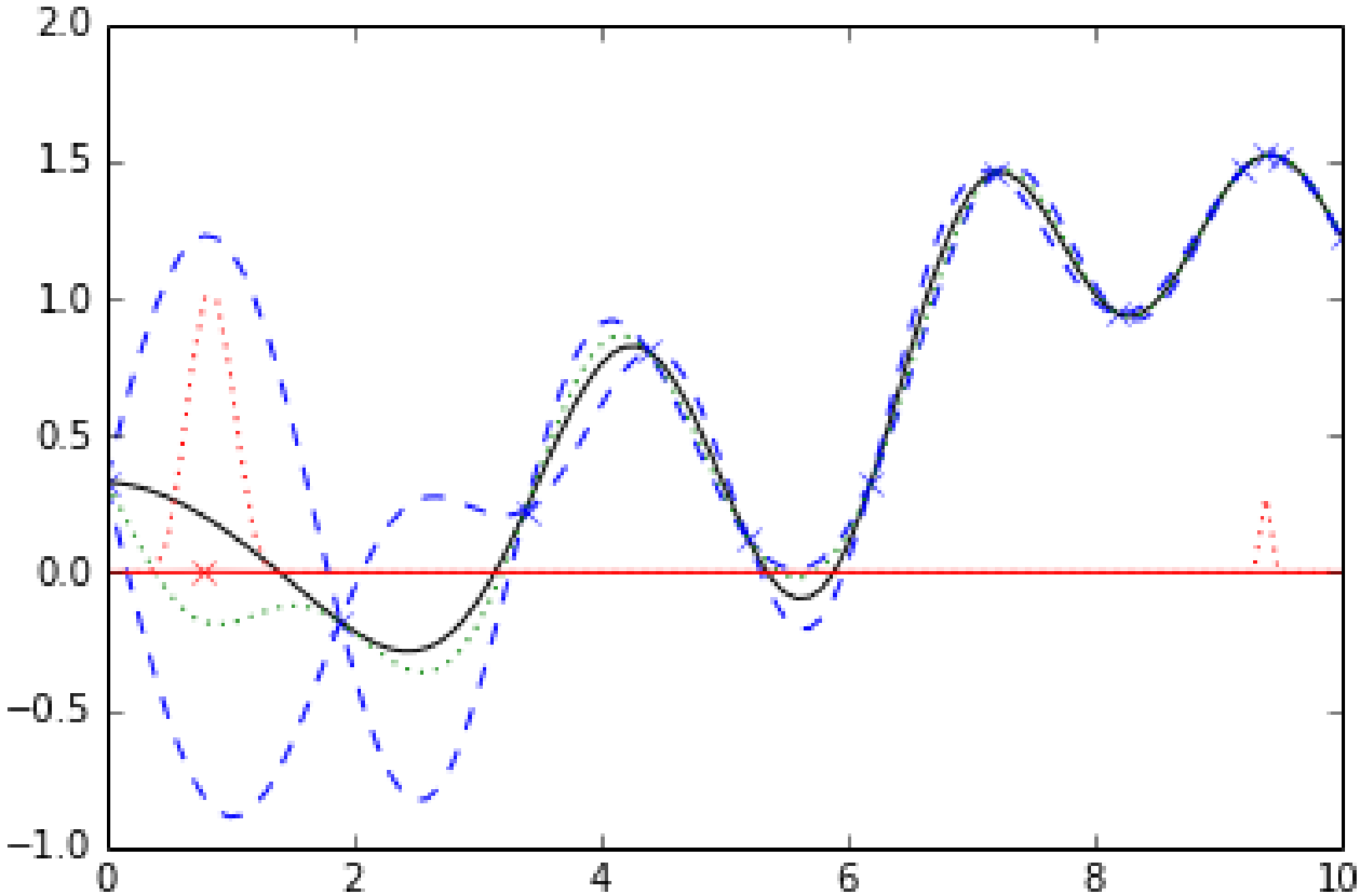
GPO (1D)



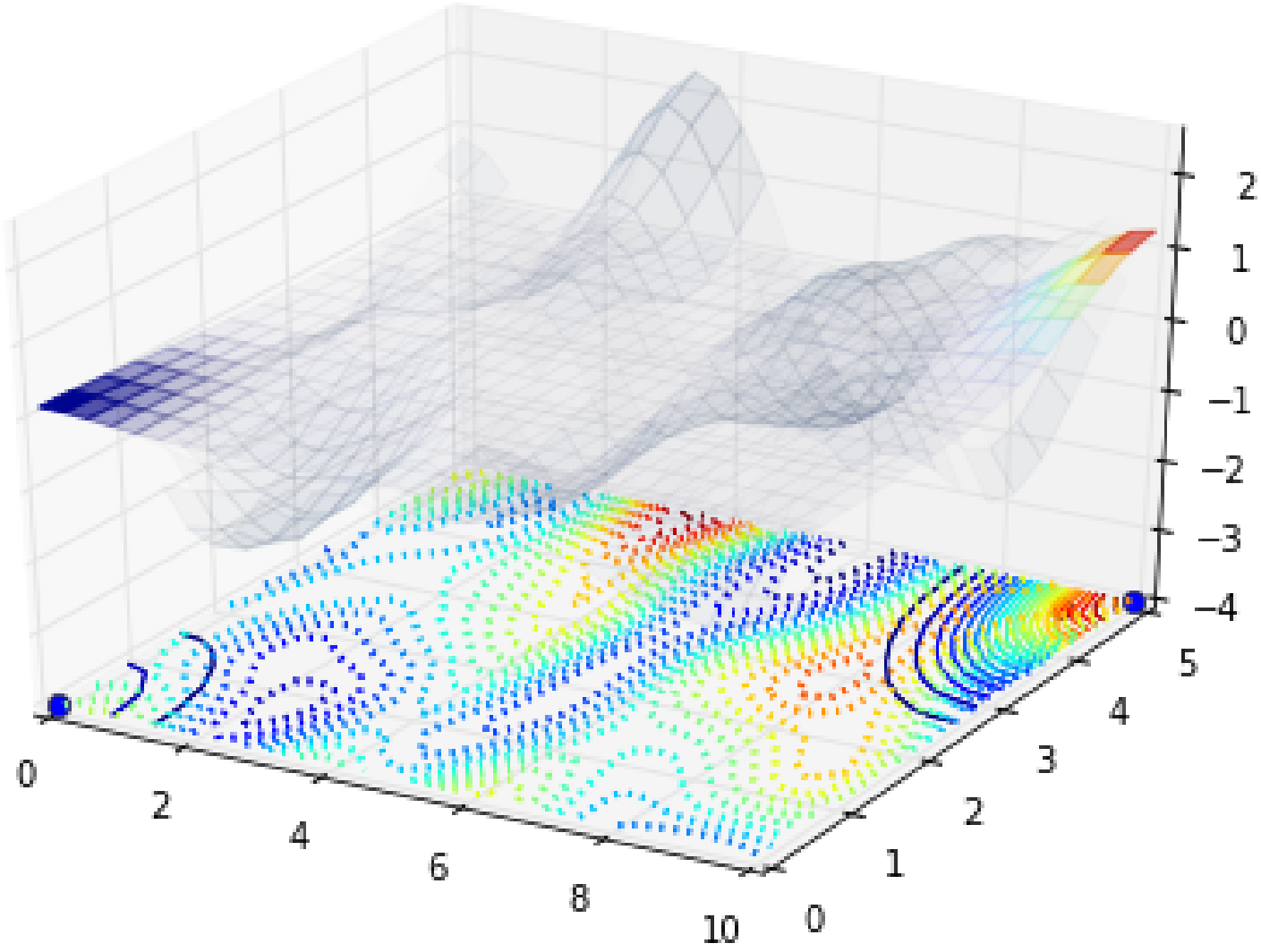
GPO (1D)



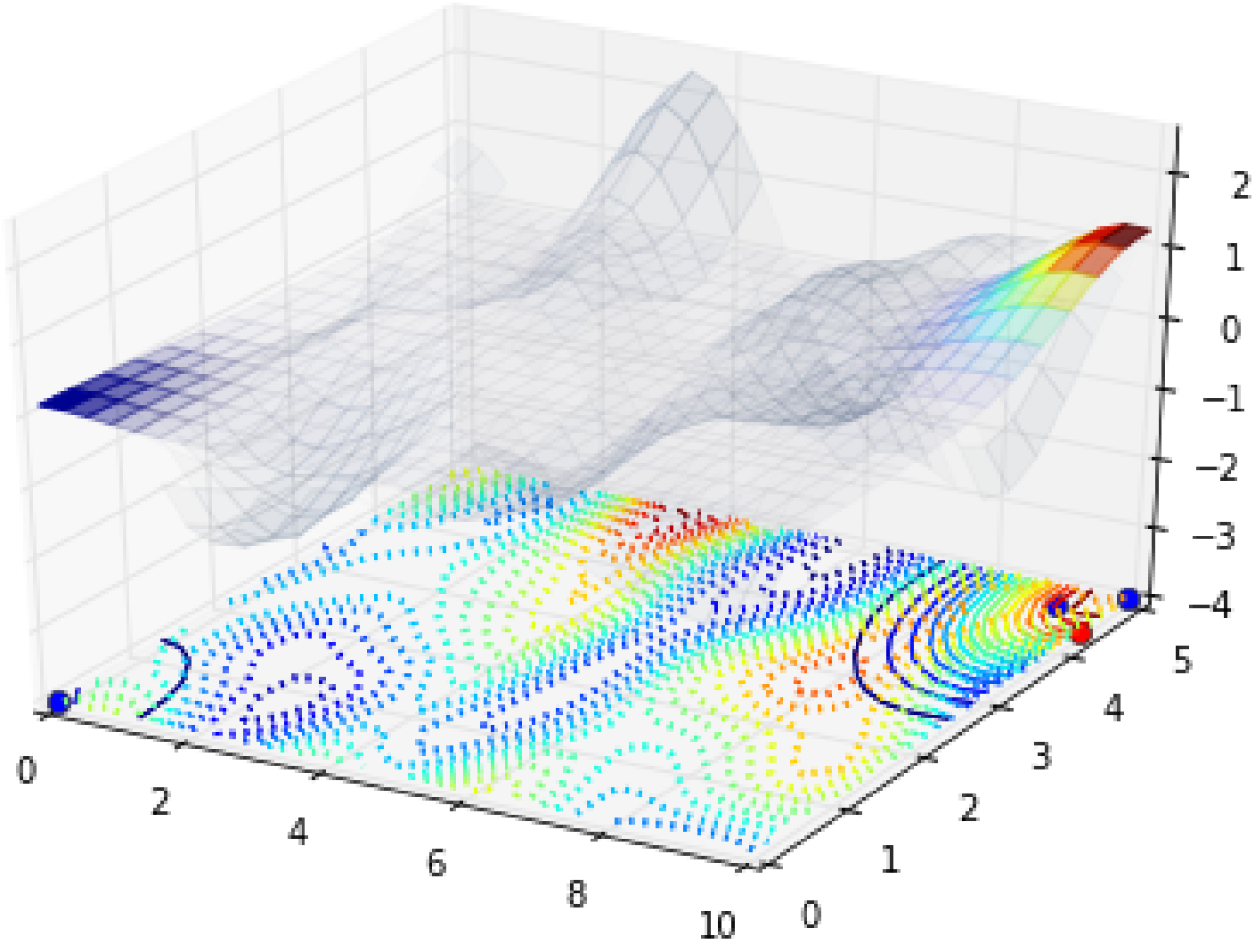
GPO (1D)



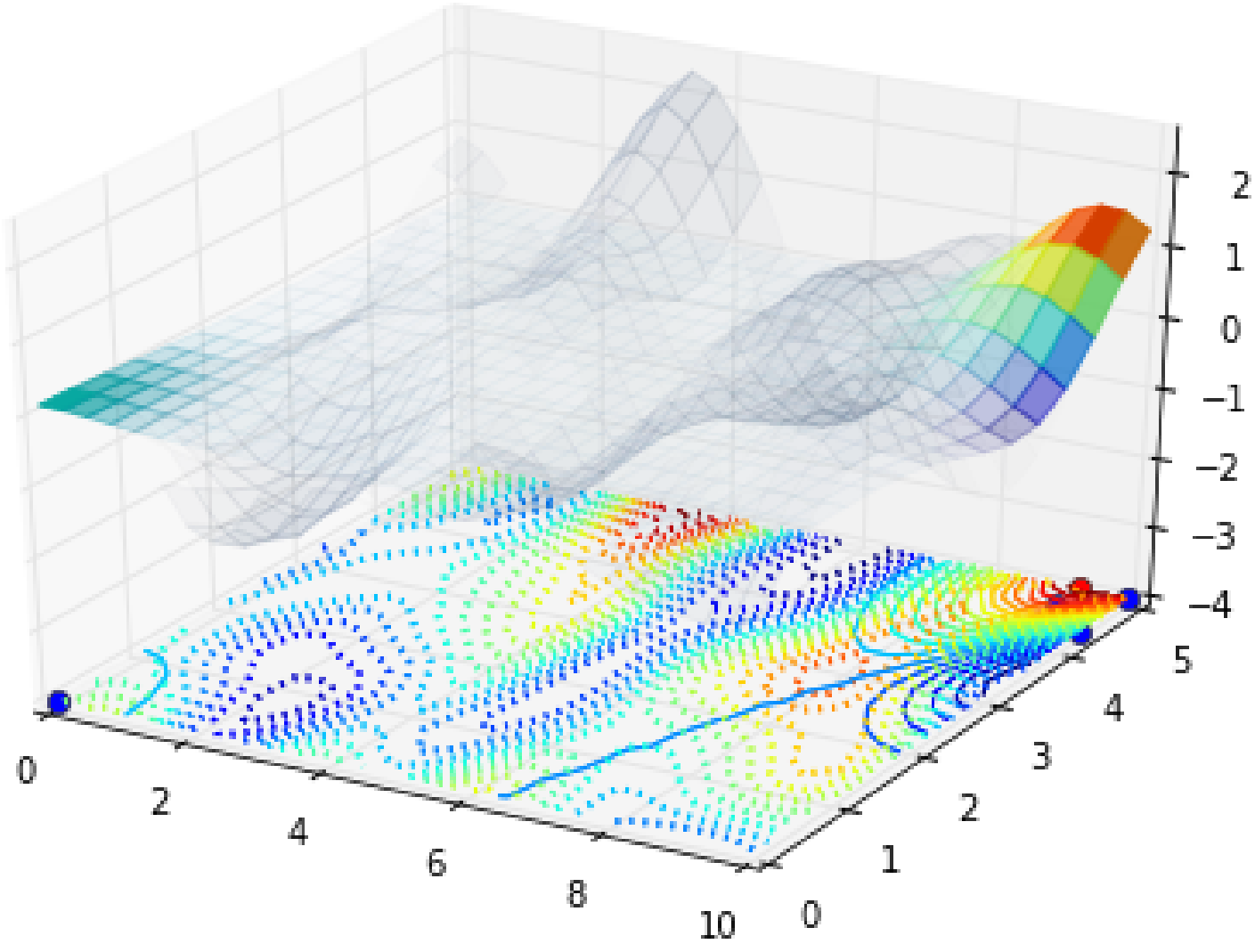
GPO (2D)



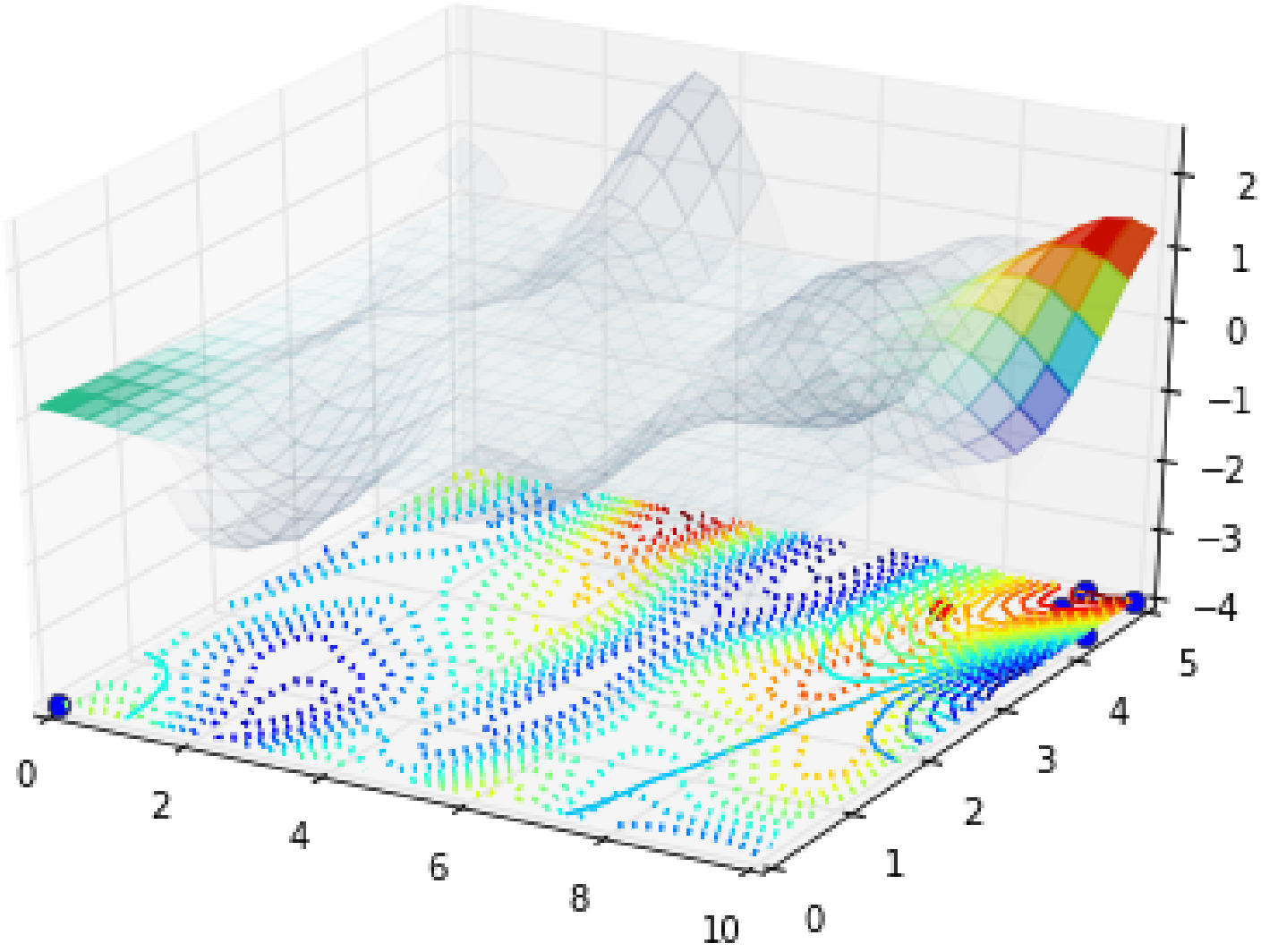
GPO (2D)



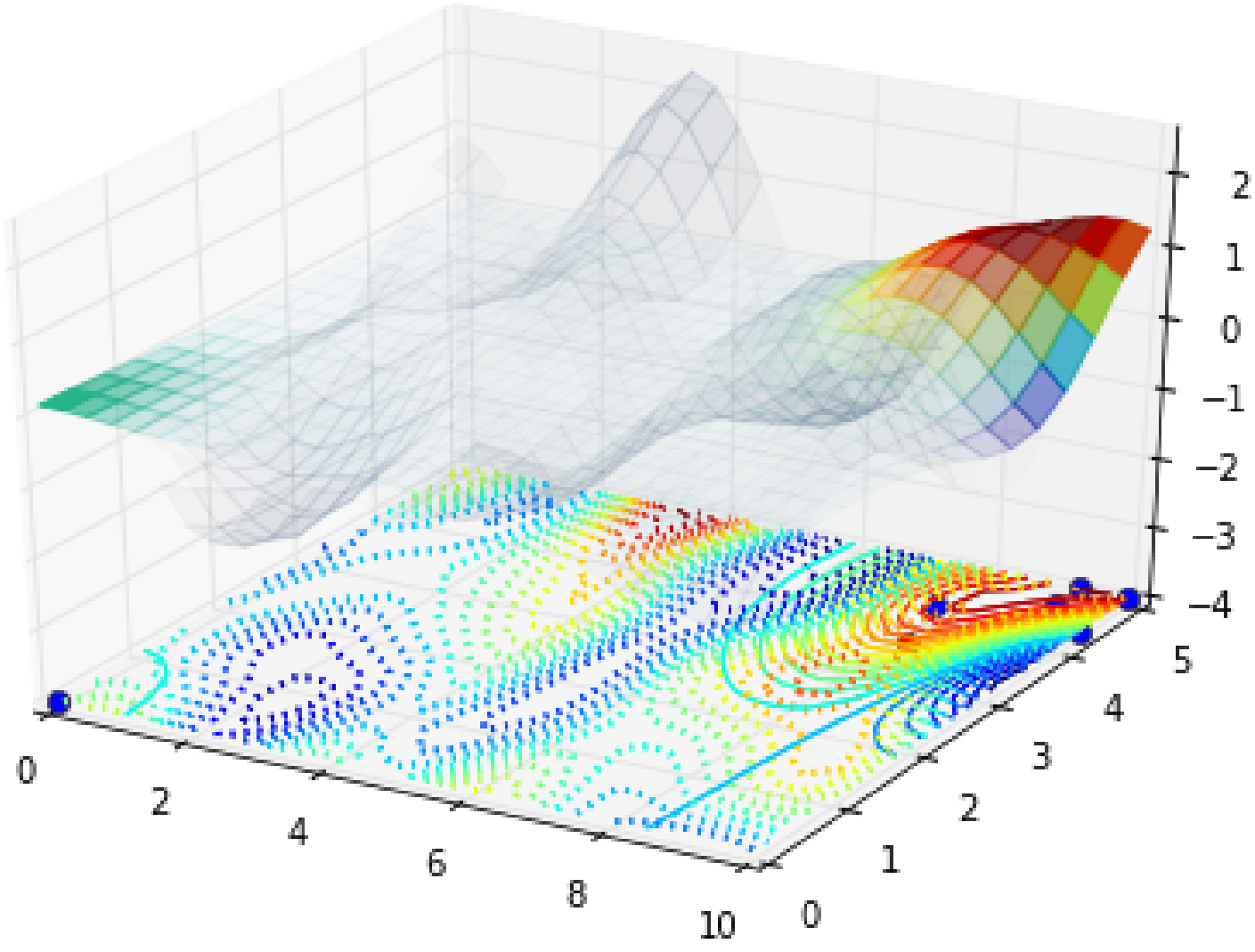
GPO (2D)



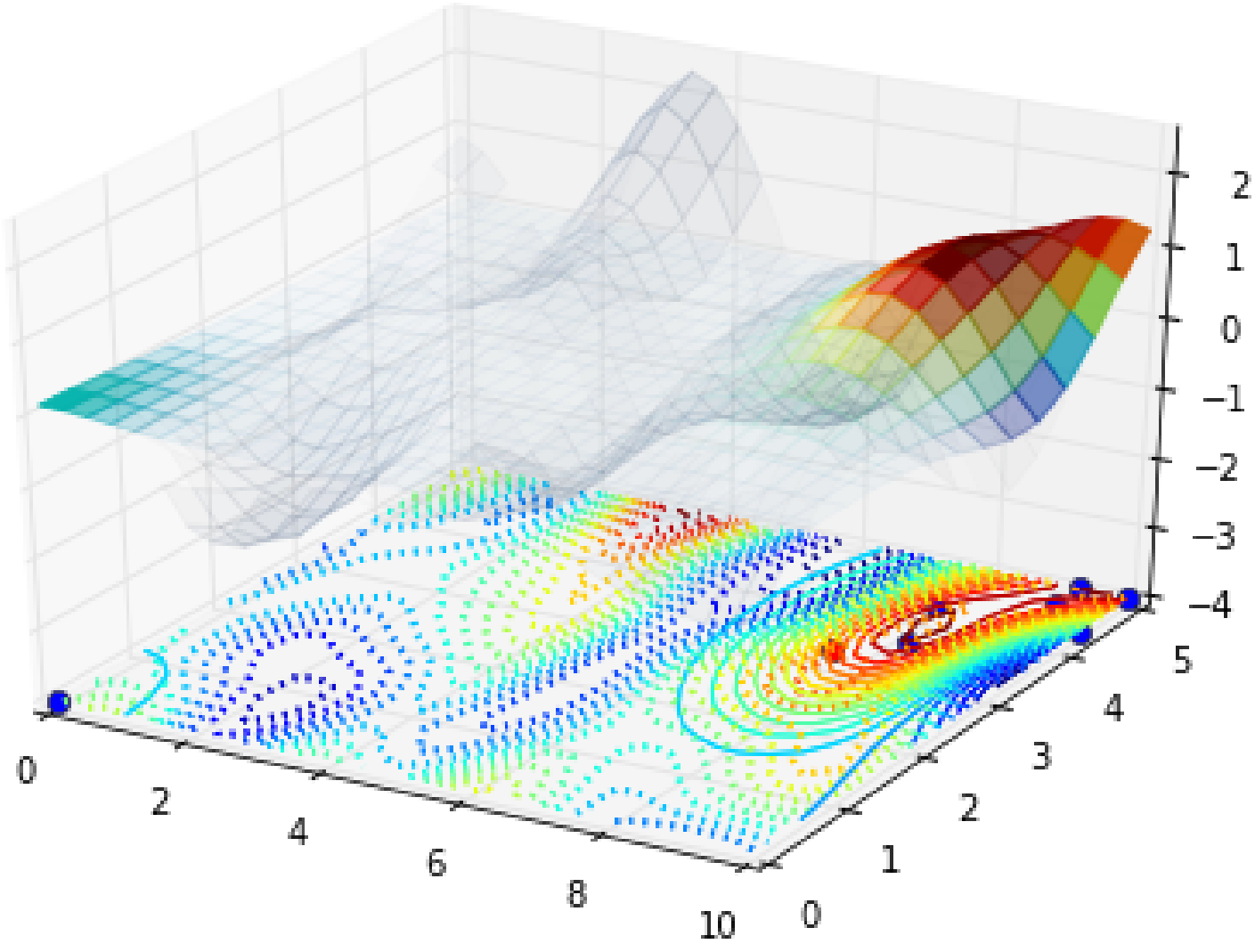
GPO (2D)



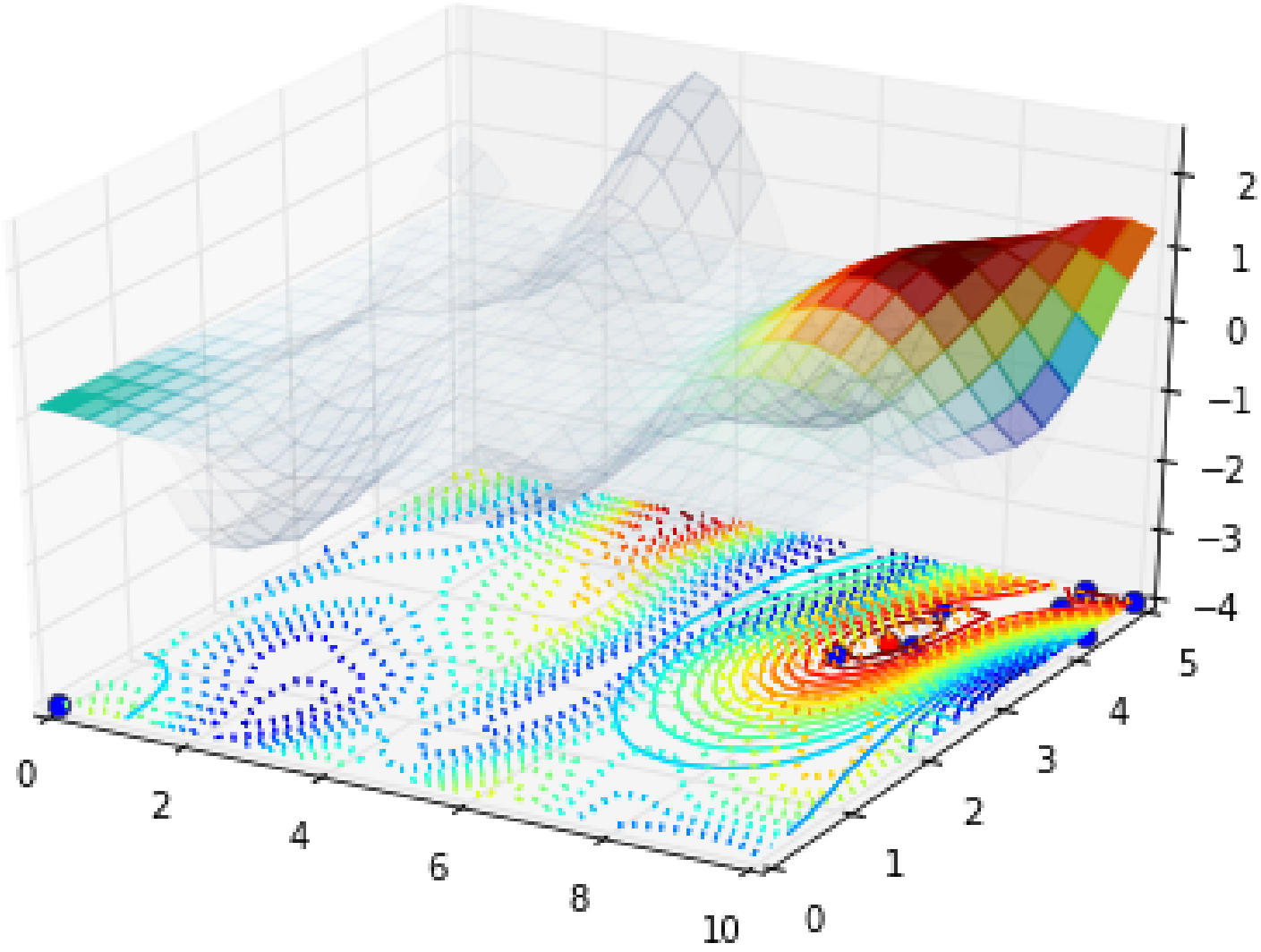
GPO (2D)



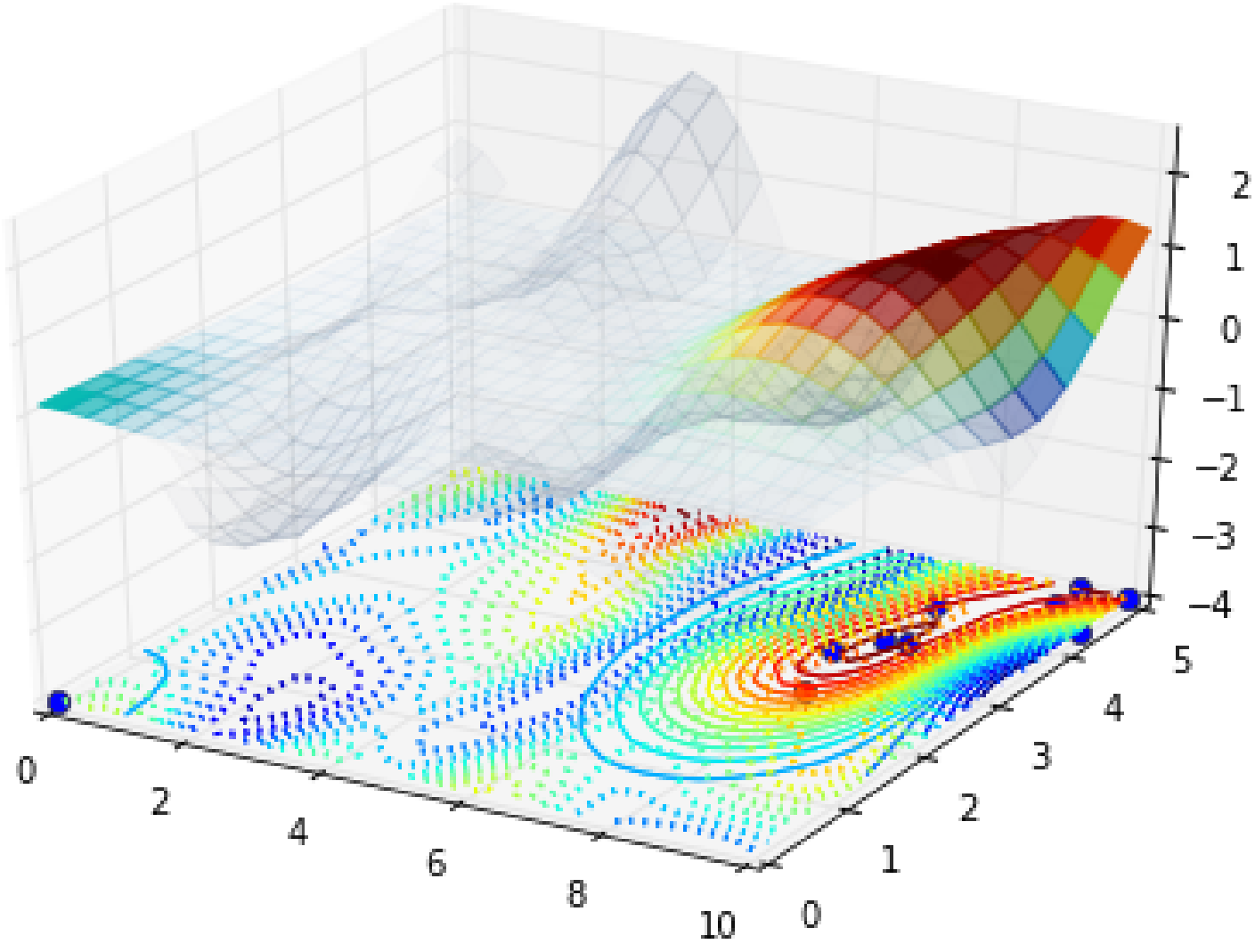
GPO (2D)



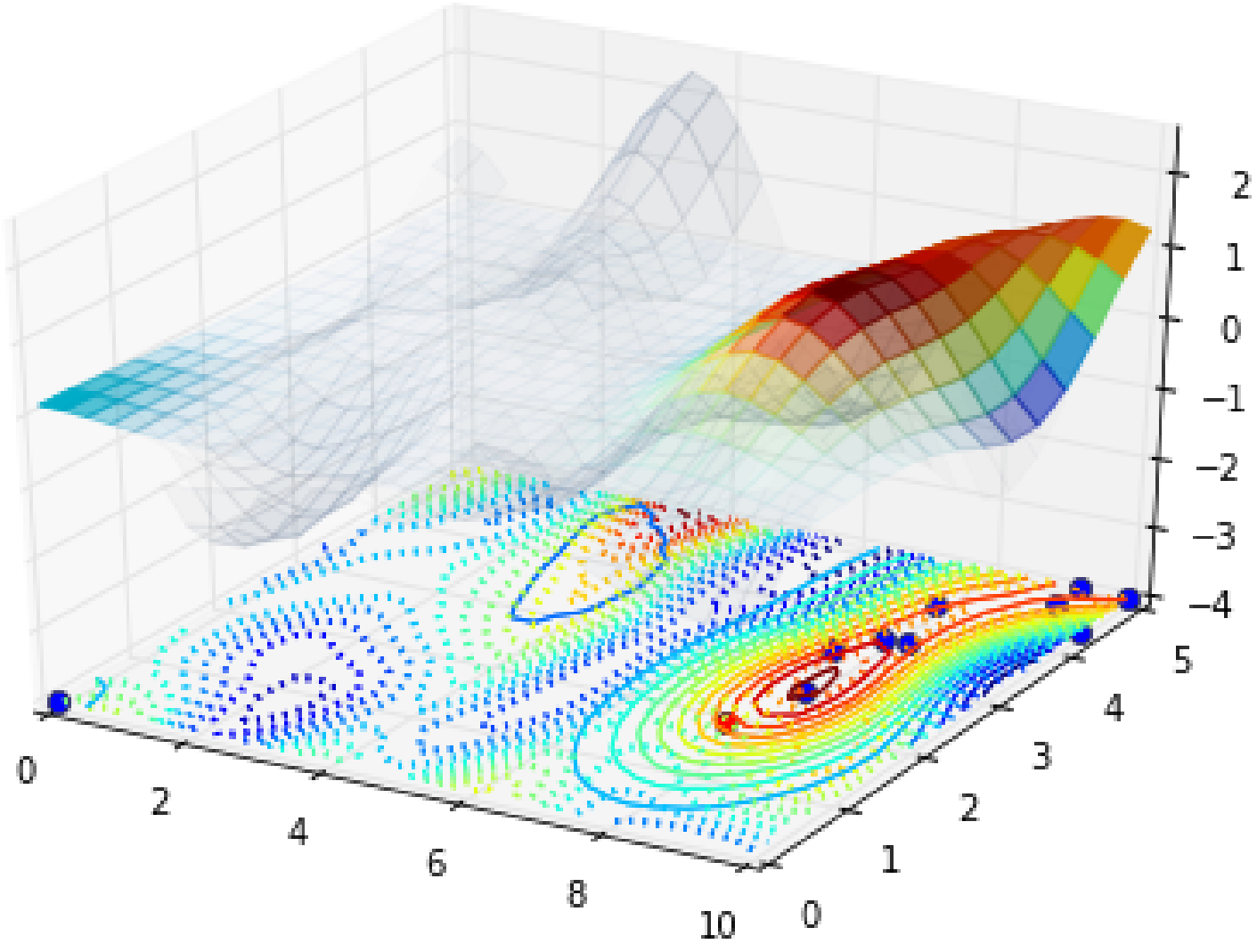
GPO (2D)



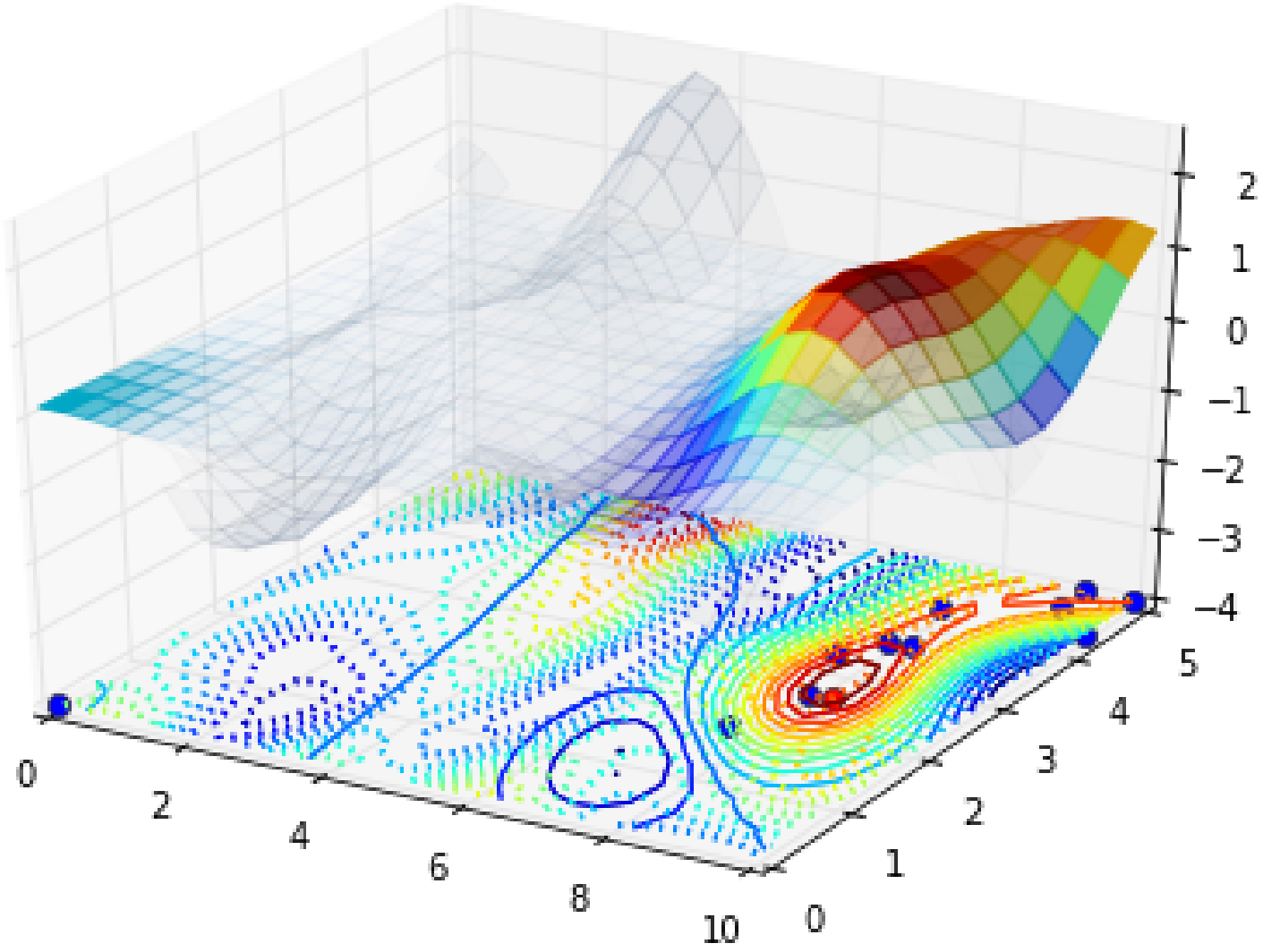
GPO (2D)



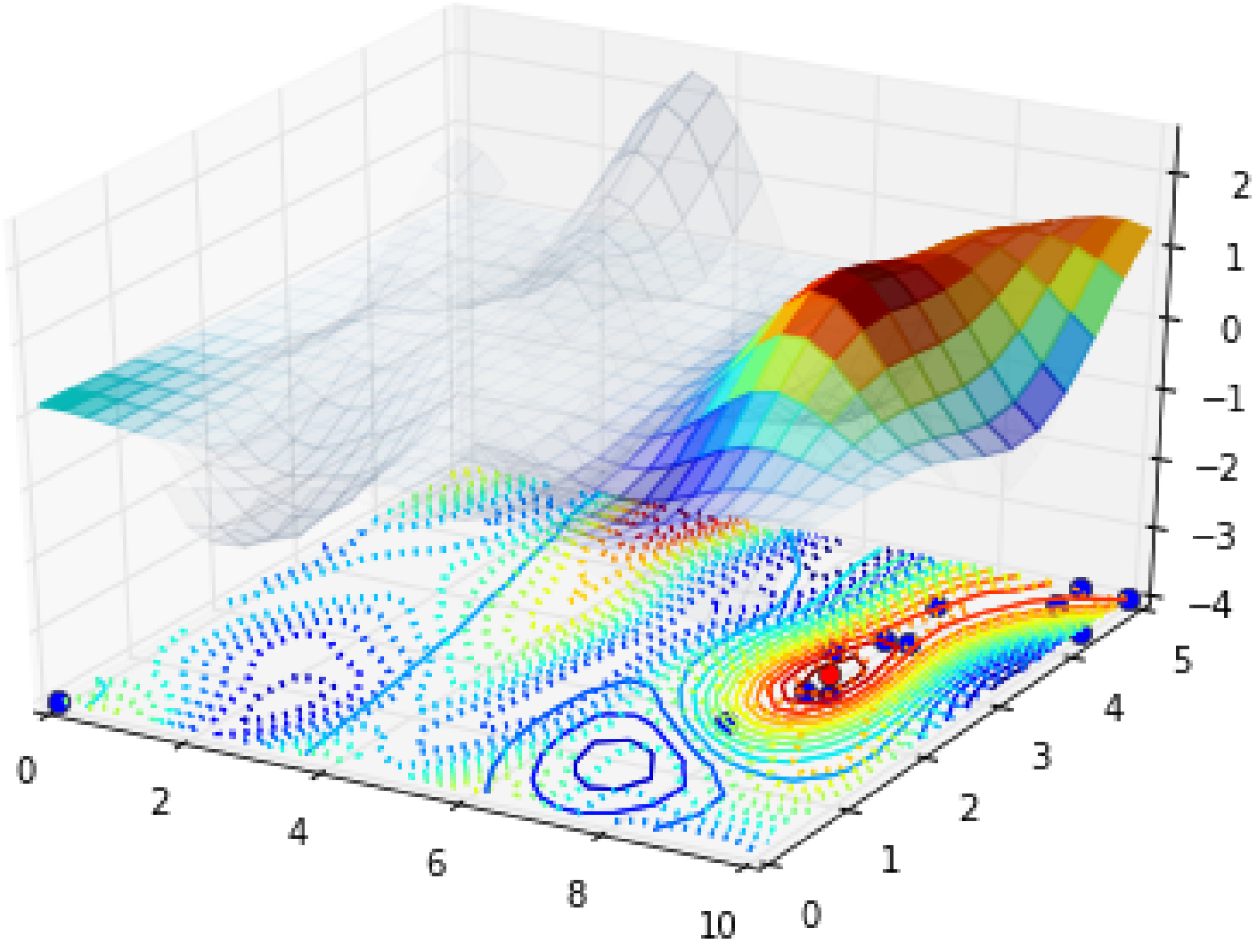
GPO (2D)



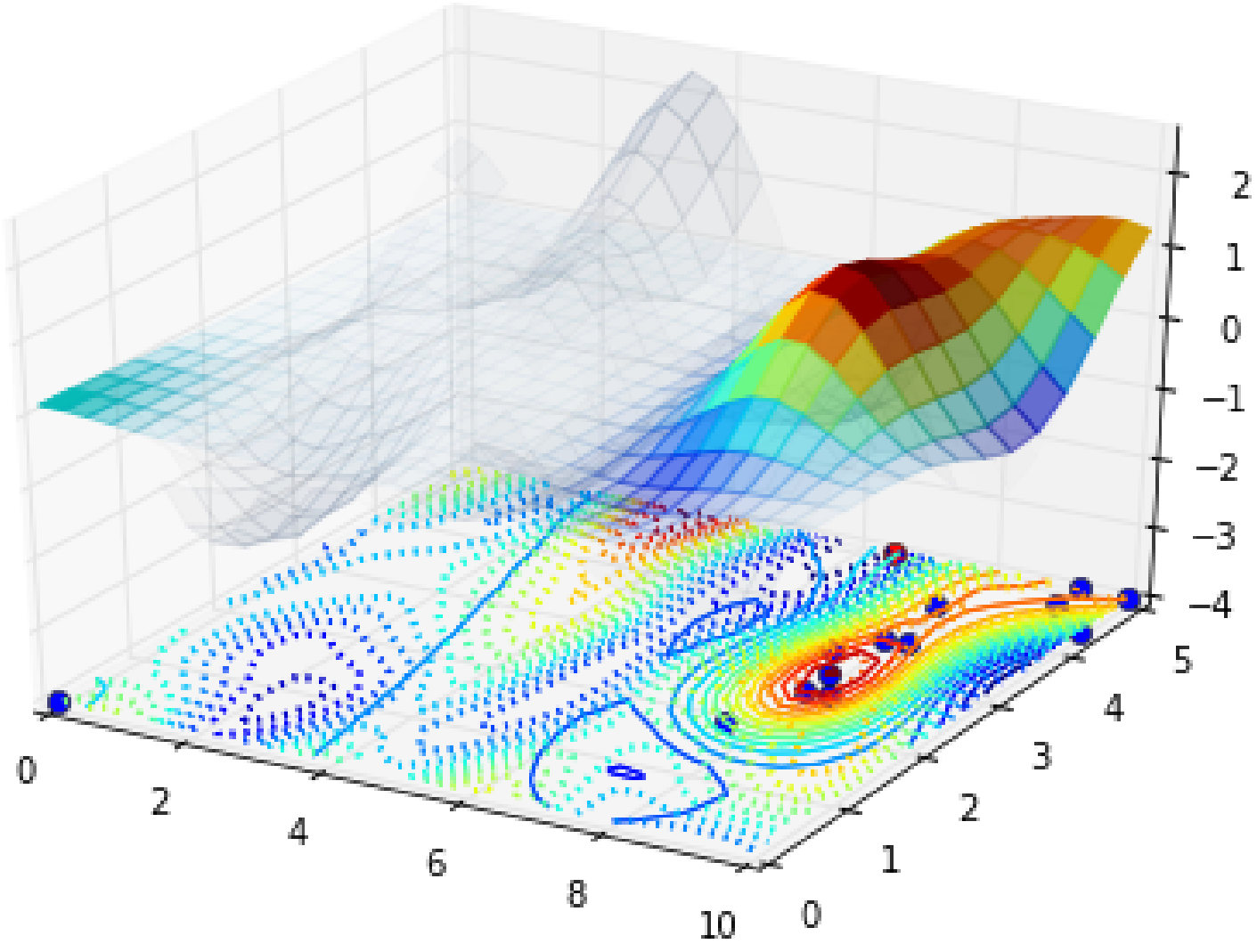
GPO (2D)



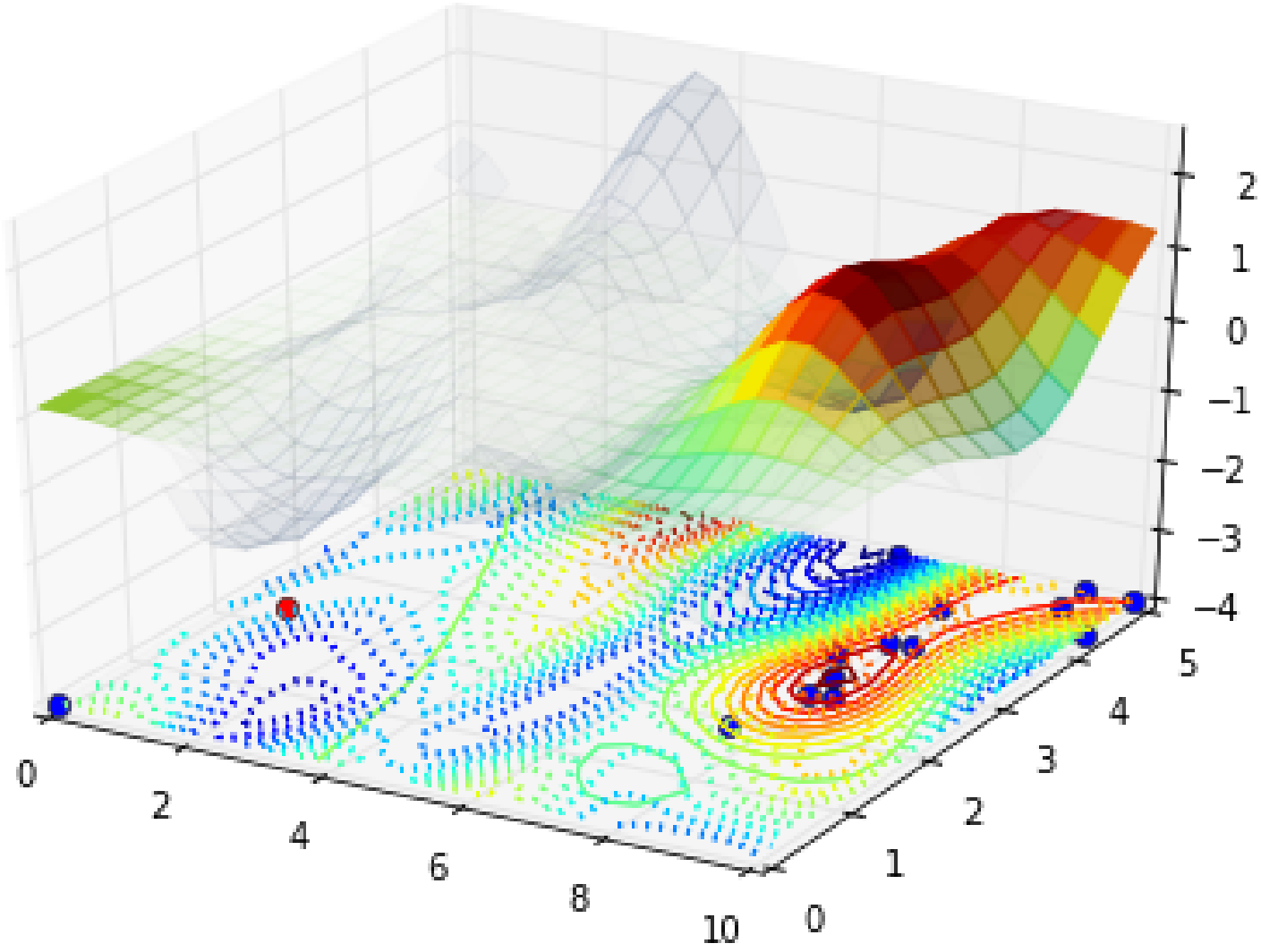
GPO (2D)



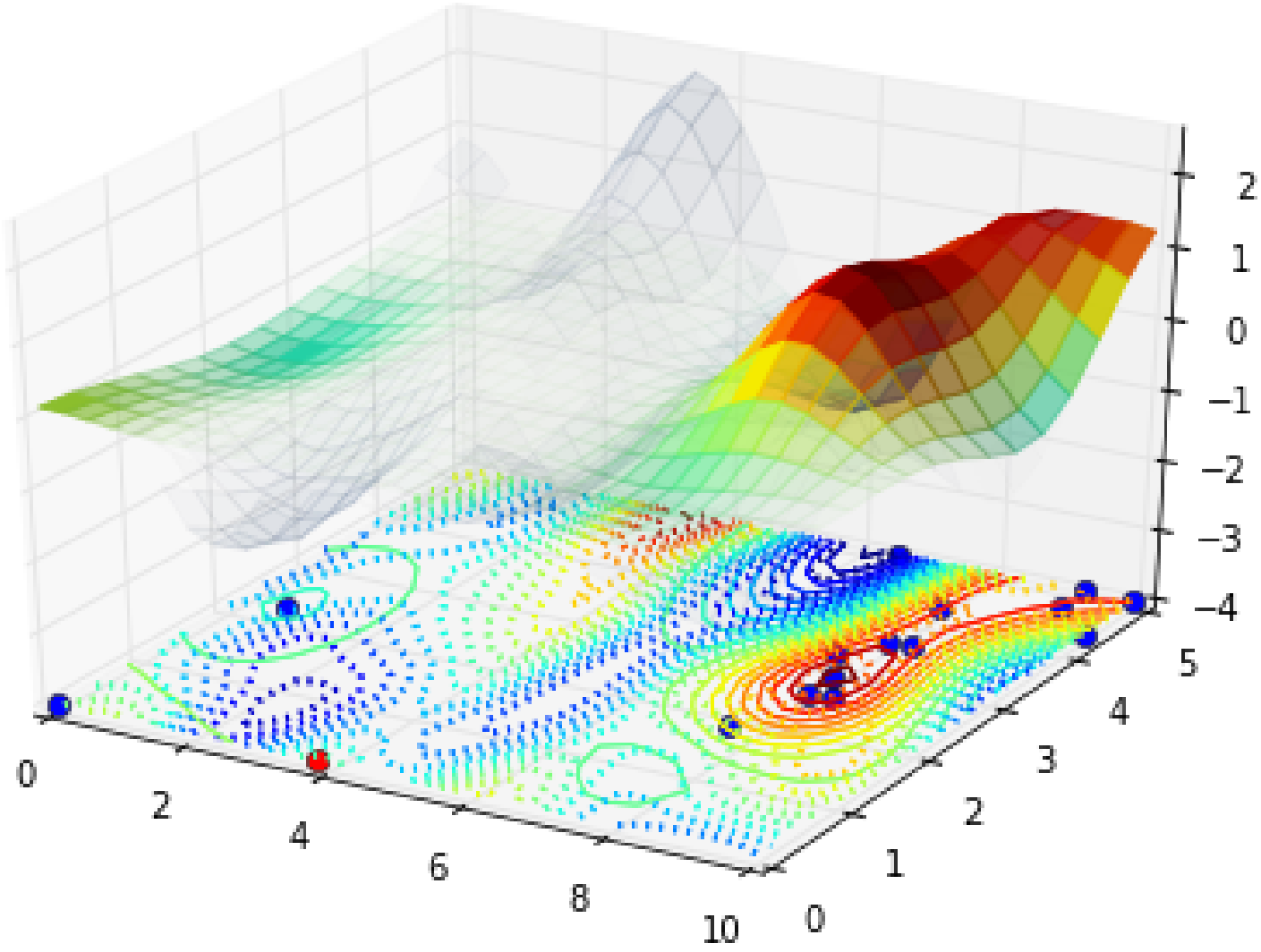
GPO (2D)



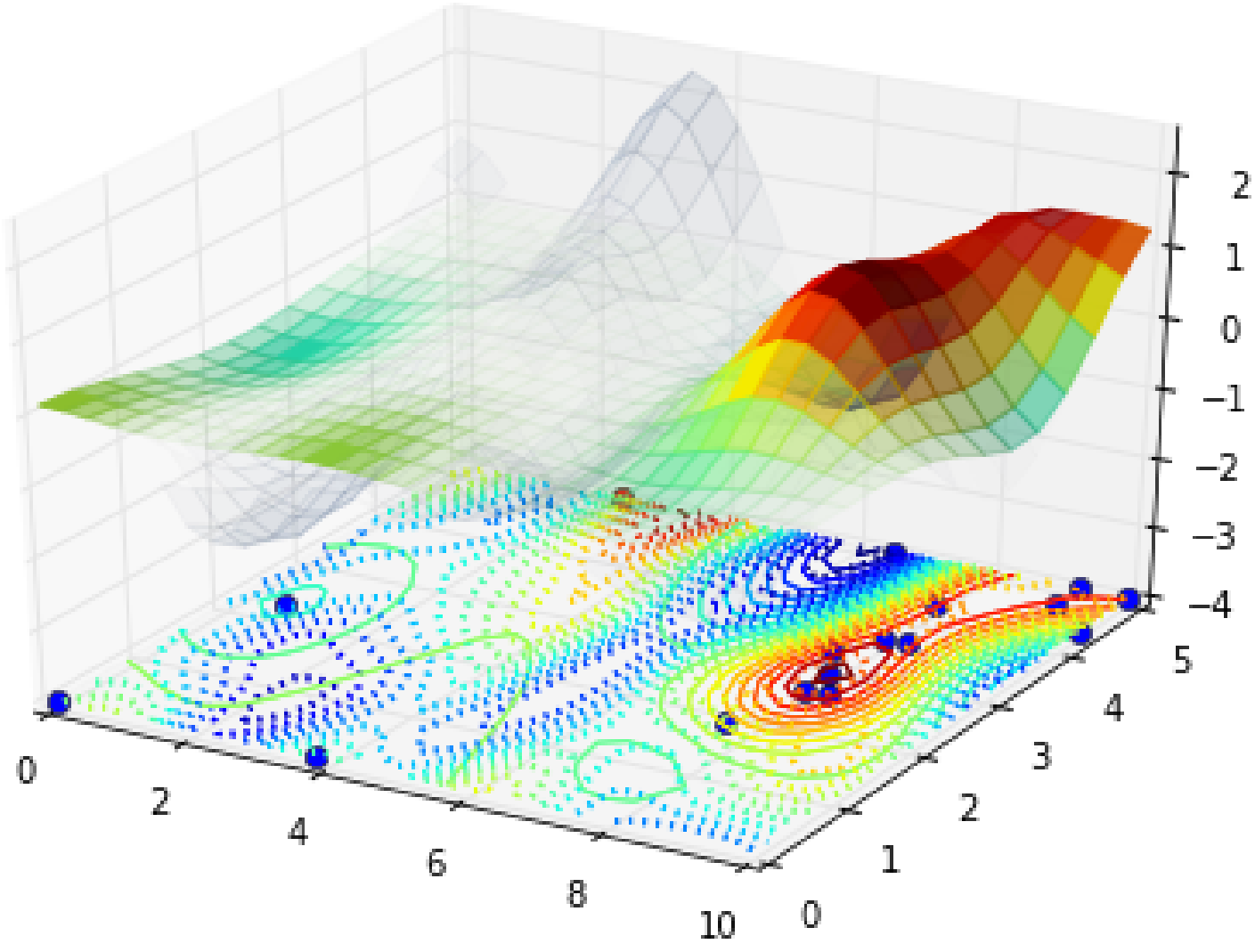
GPO (2D)



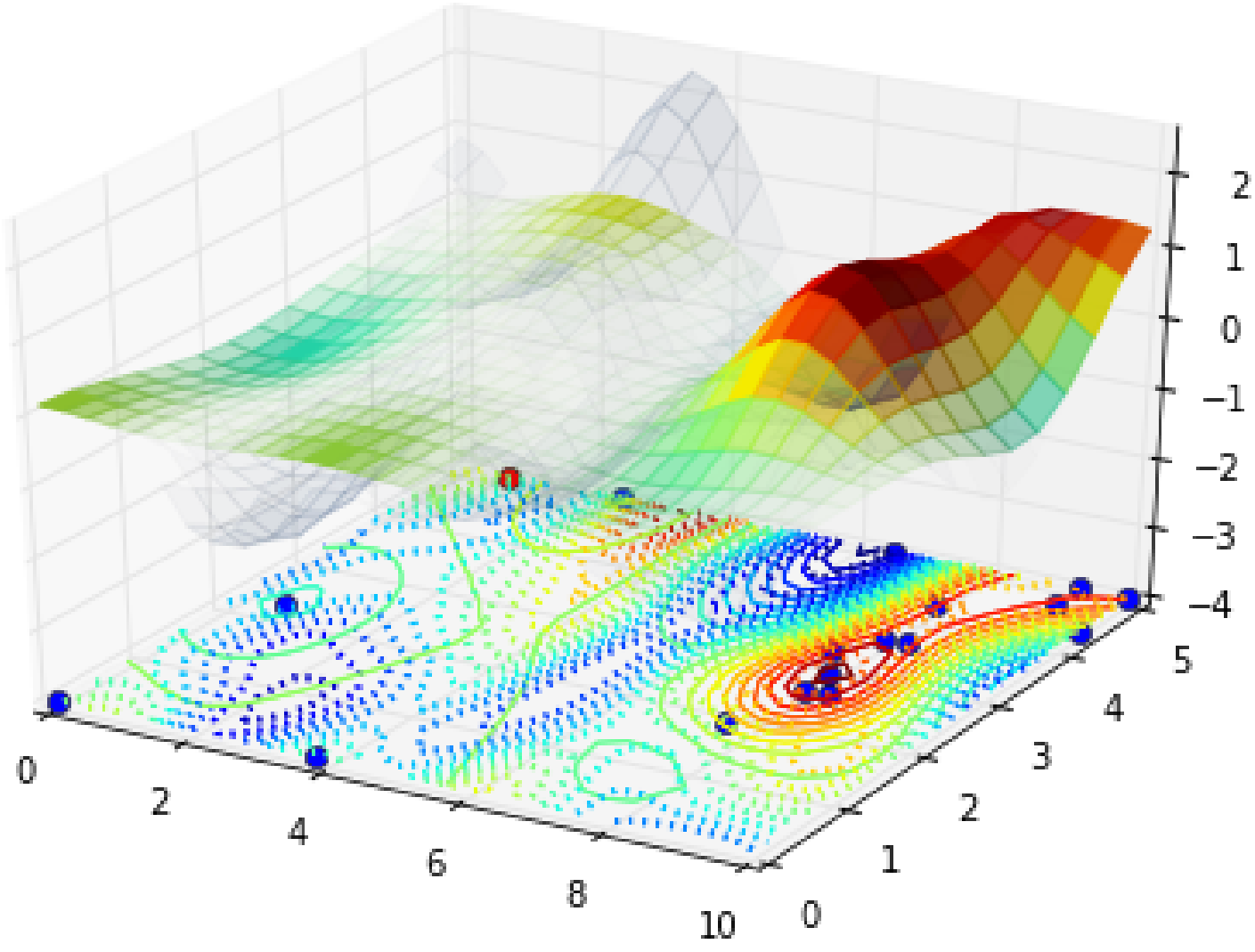
GPO (2D)



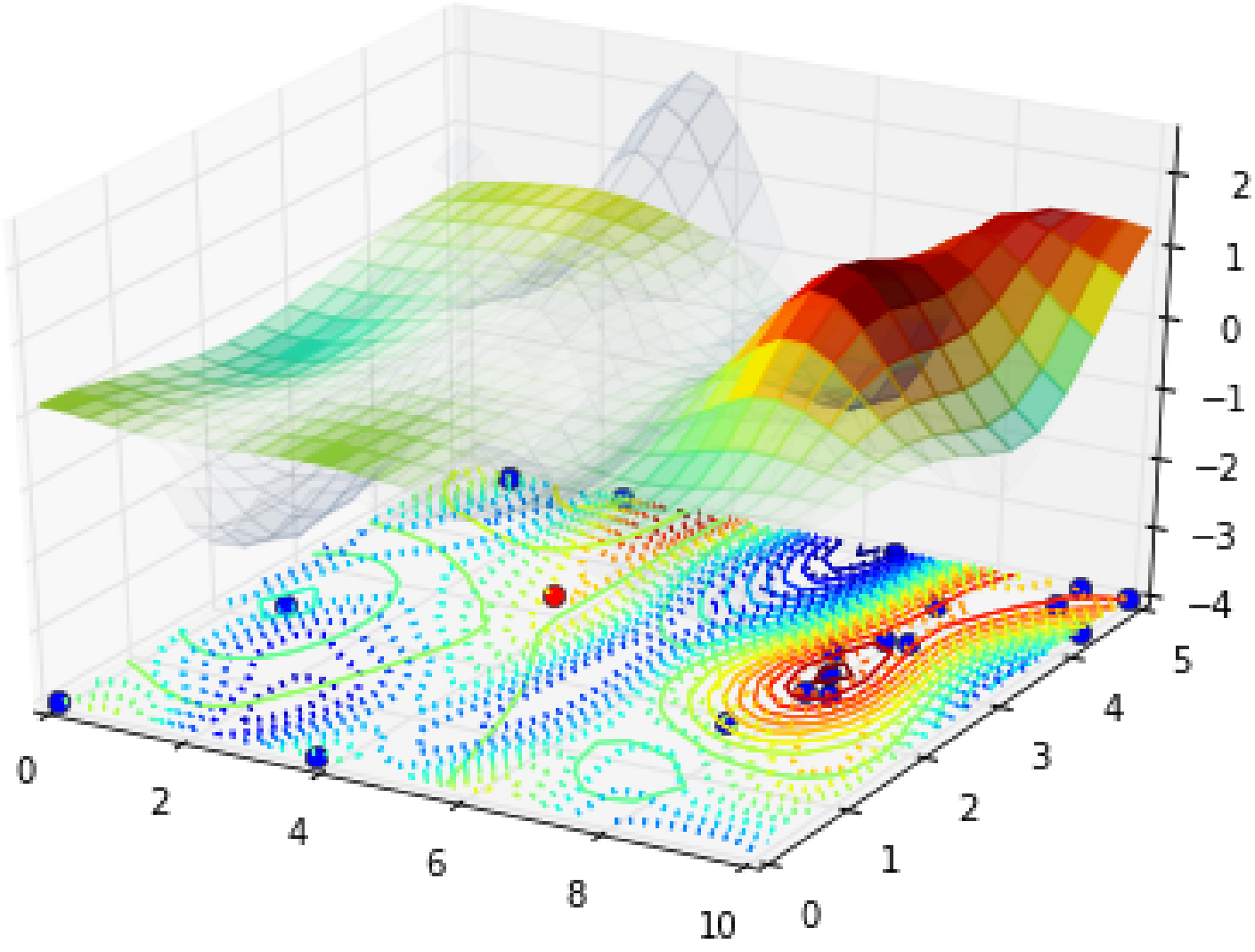
GPO (2D)



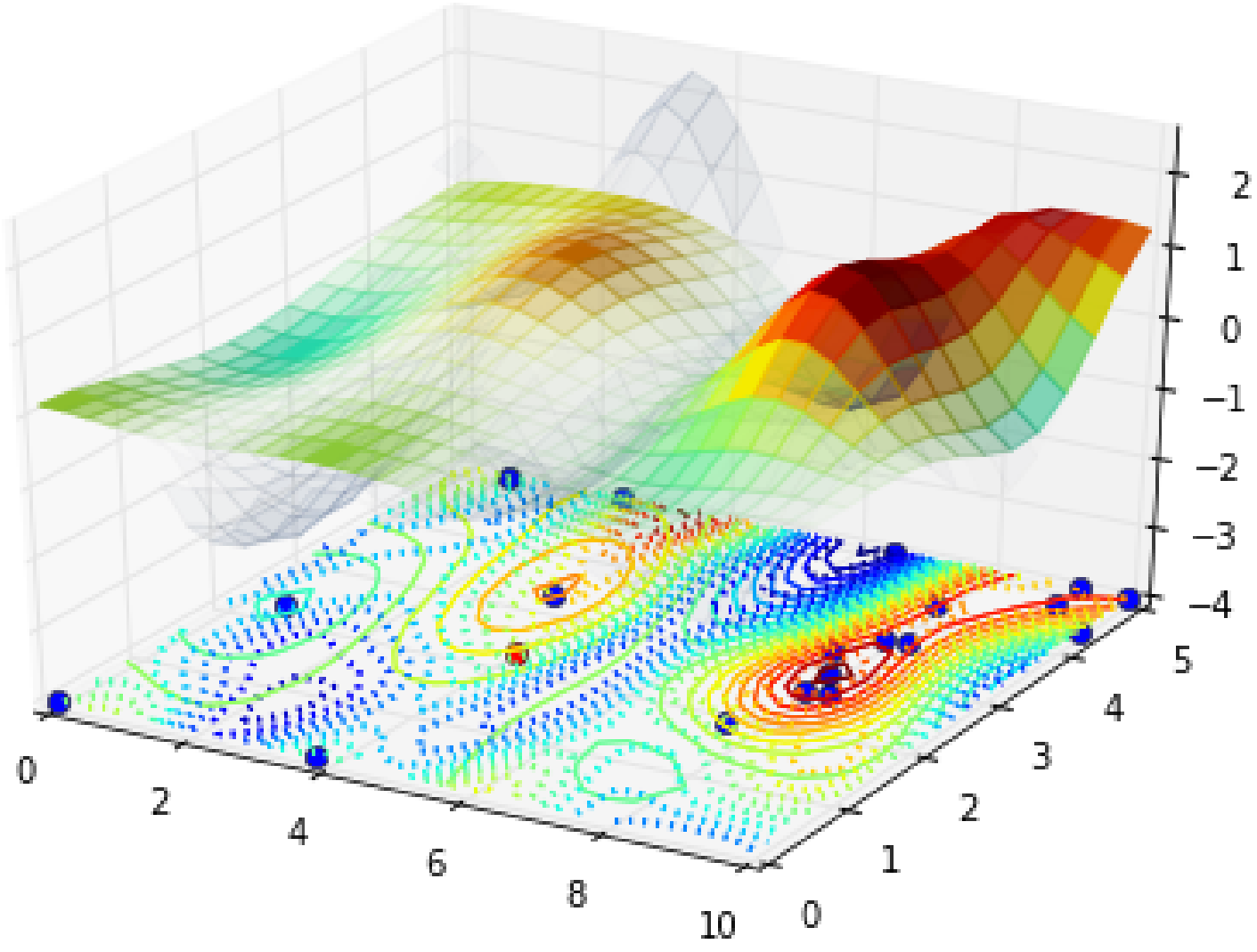
GPO (2D)



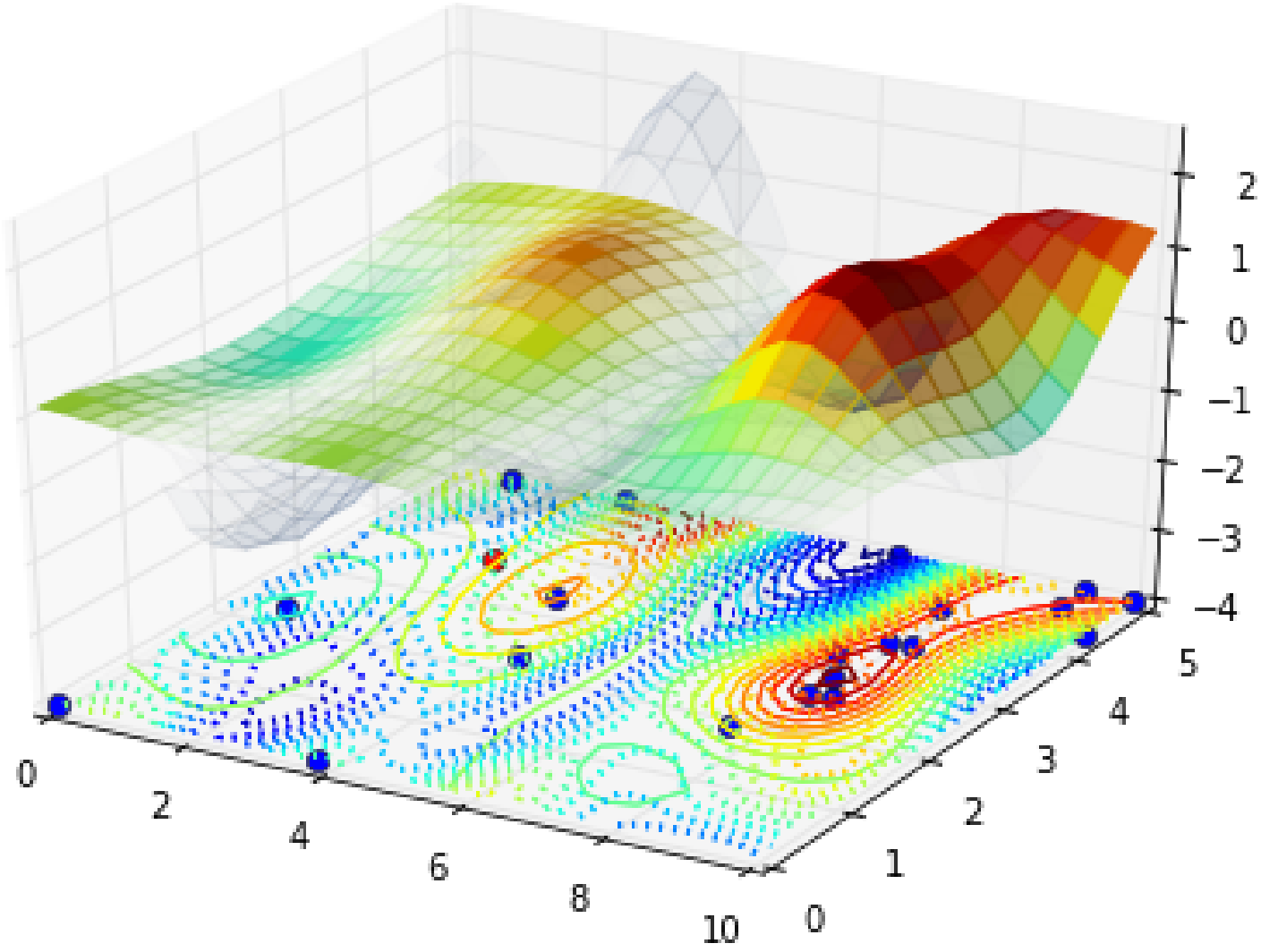
GPO (2D)



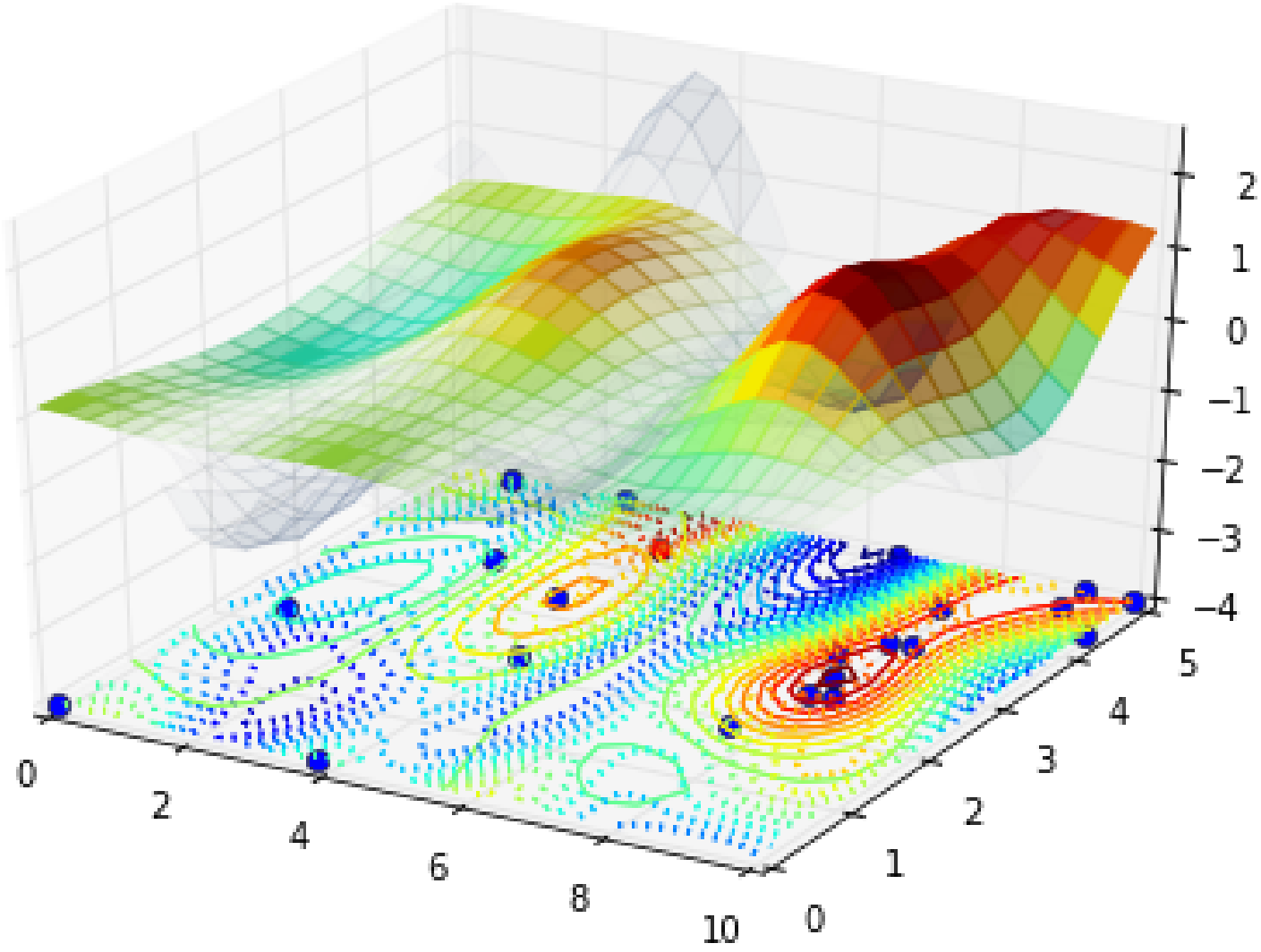
GPO (2D)



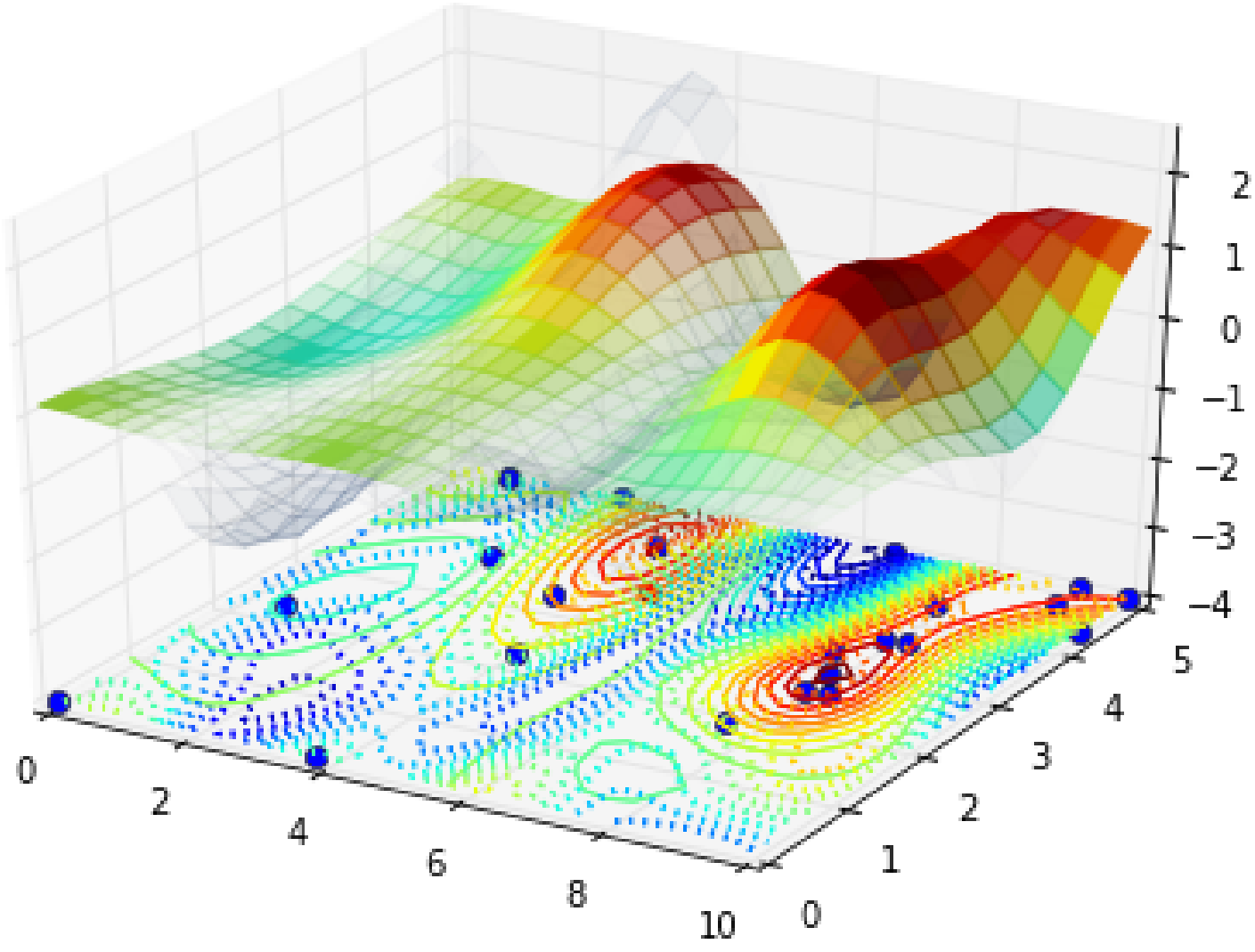
GPO (2D)



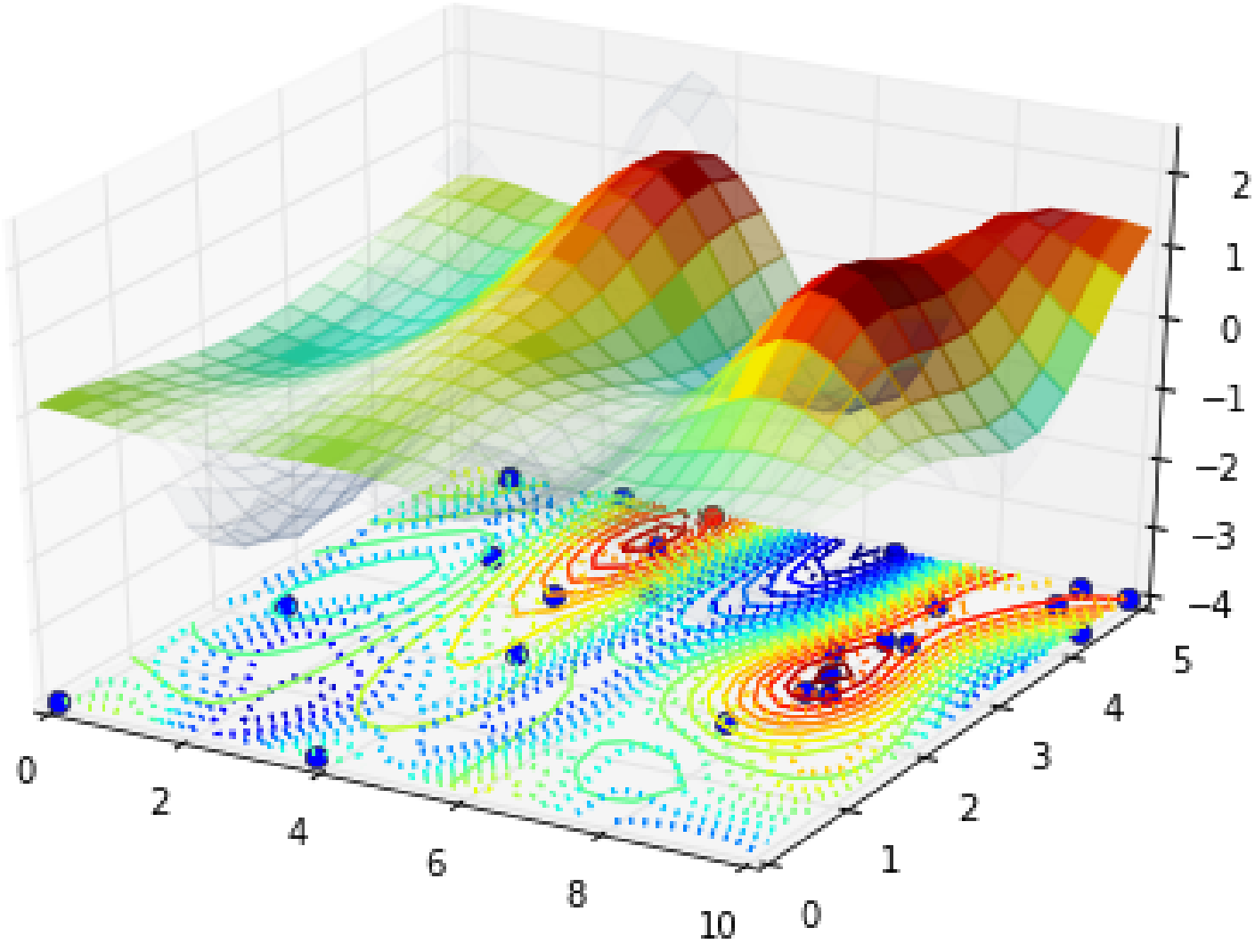
GPO (2D)



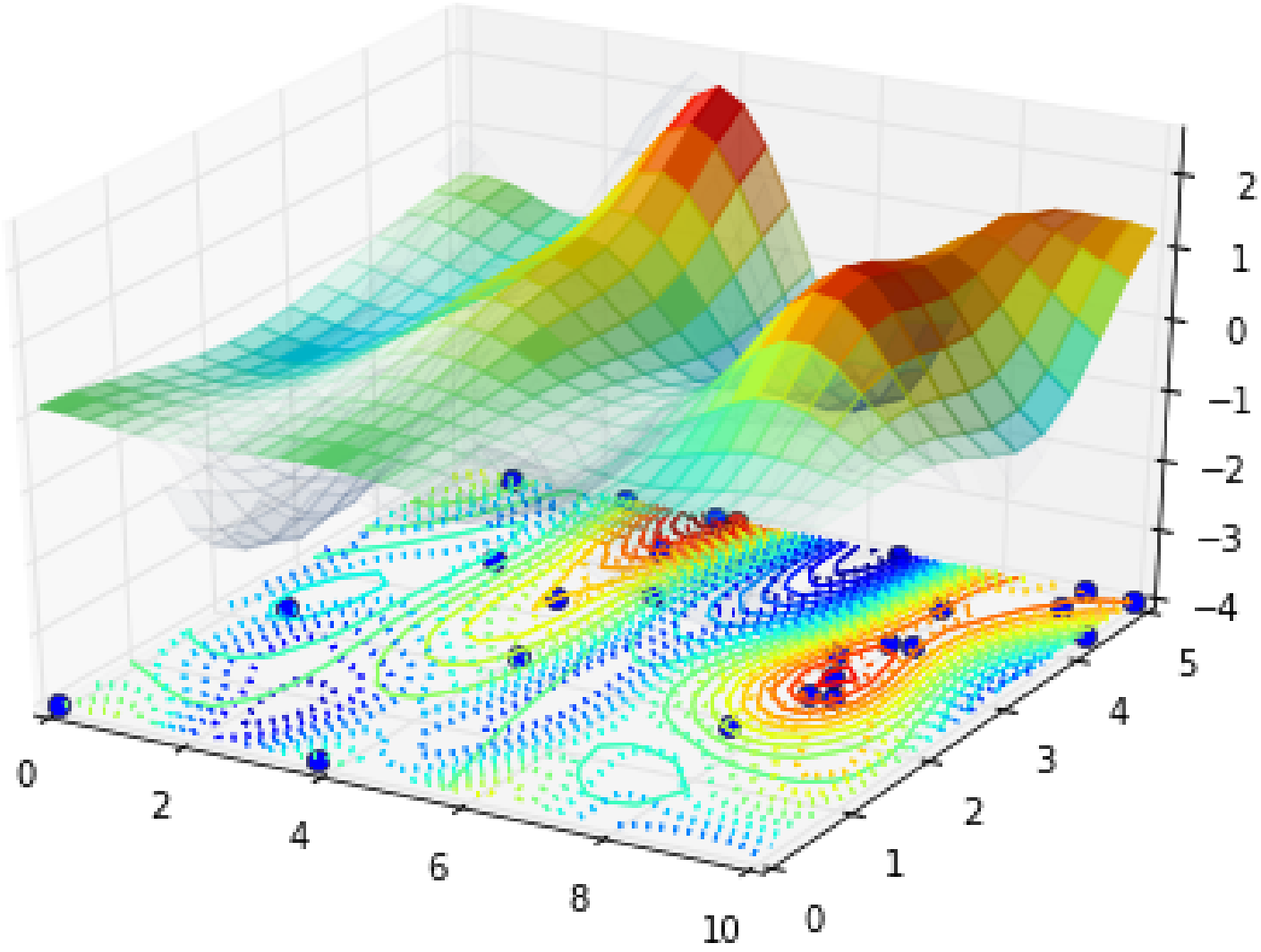
GPO (2D)



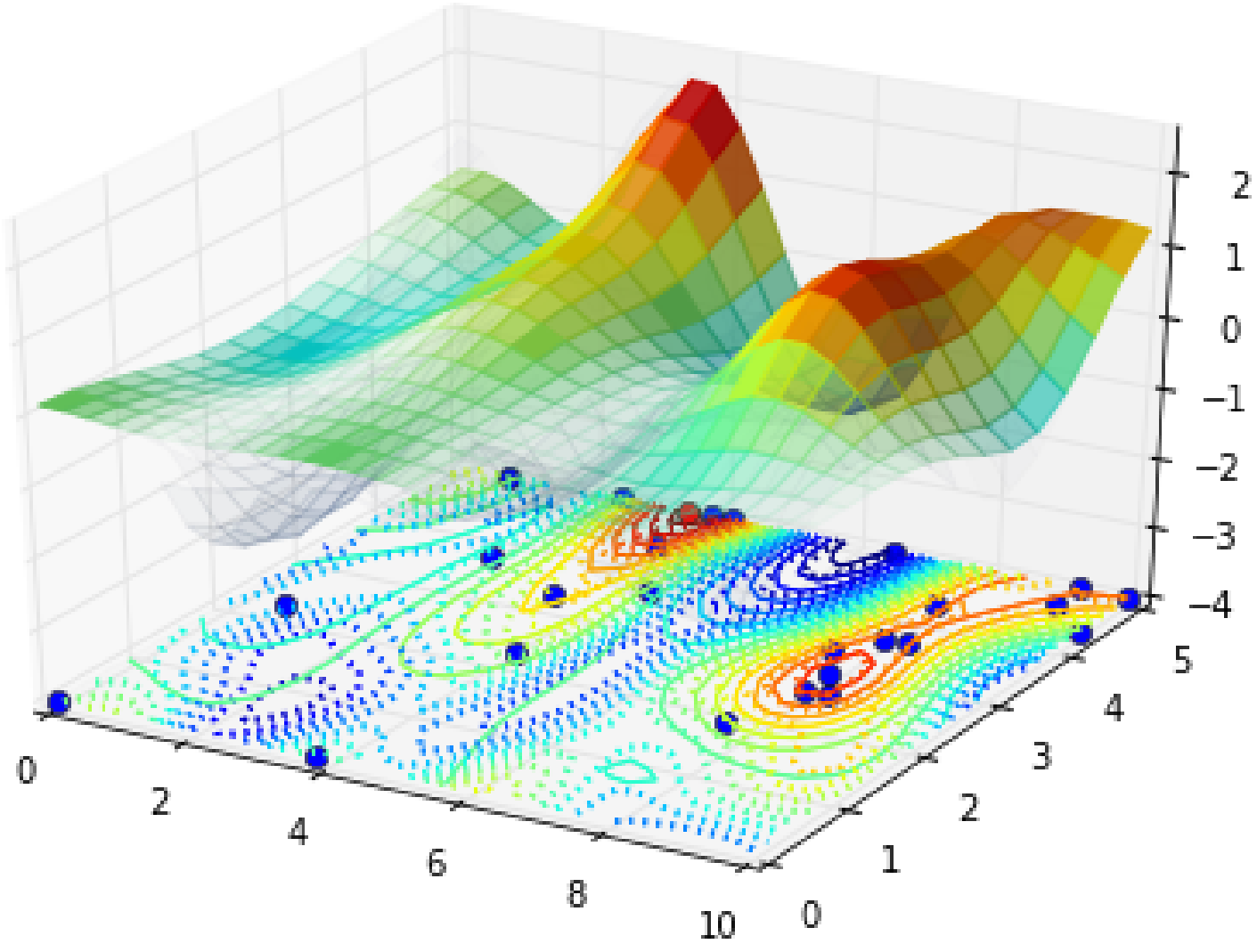
GPO (2D)



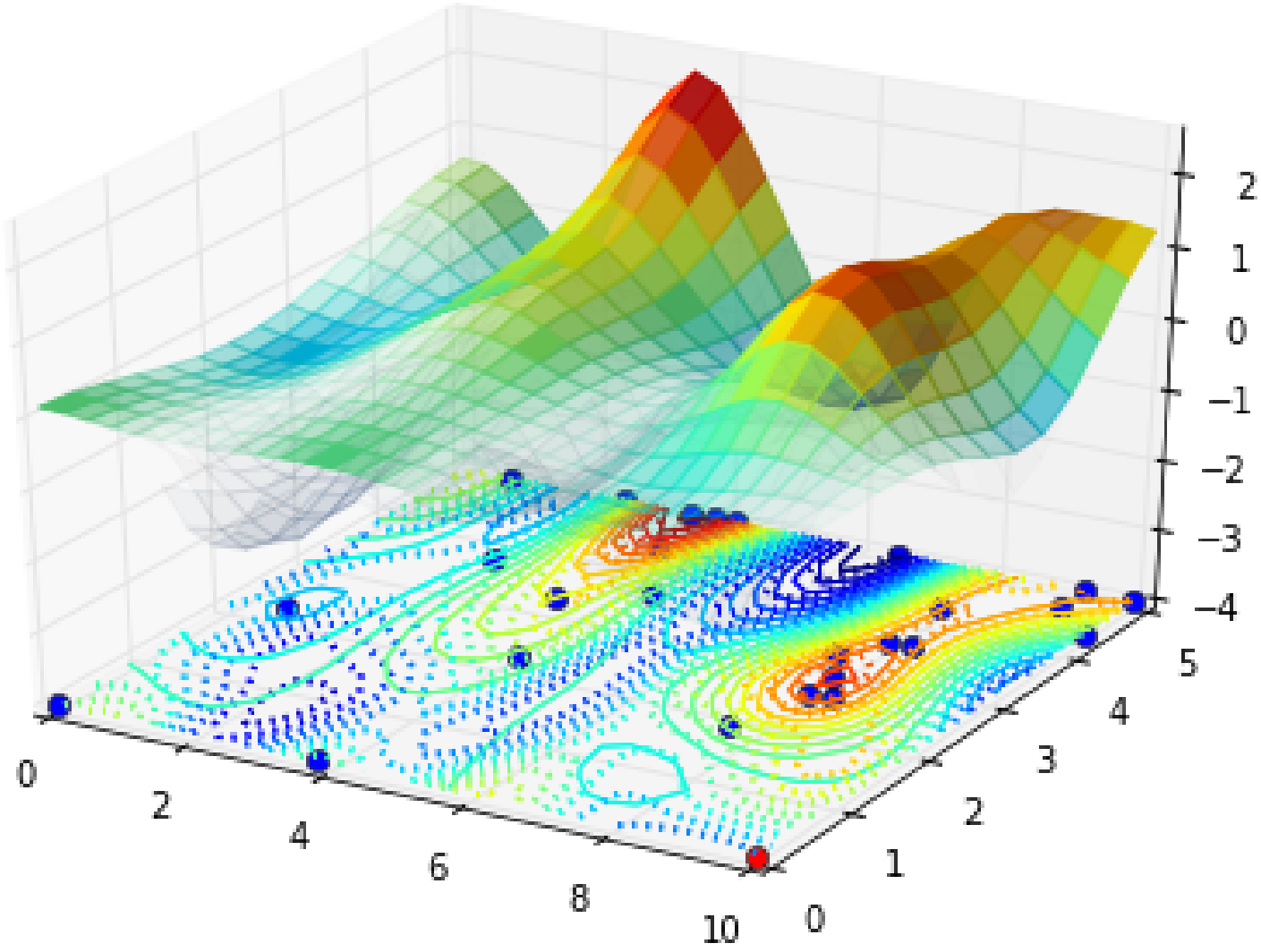
GPO (2D)



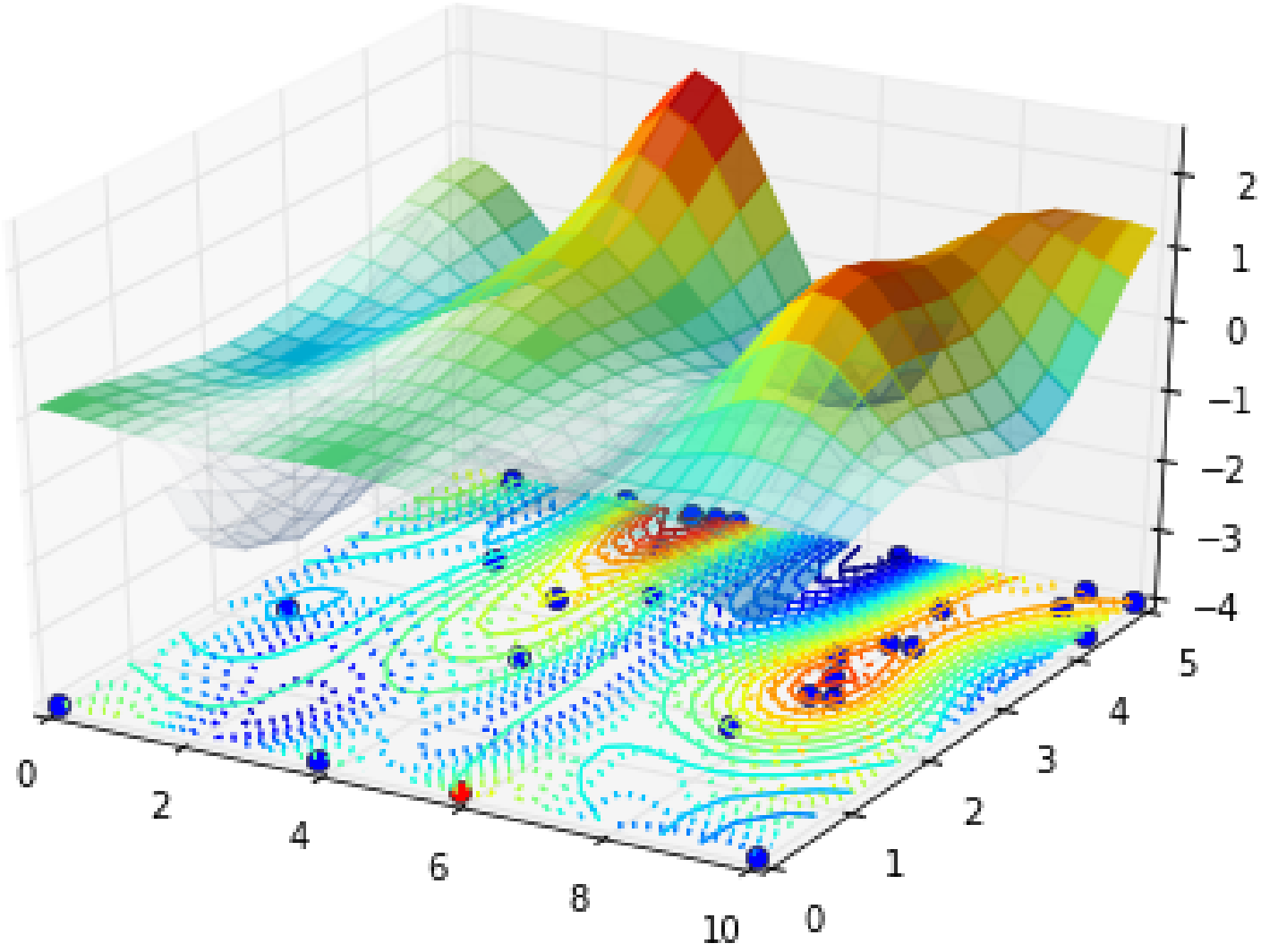
GPO (2D)



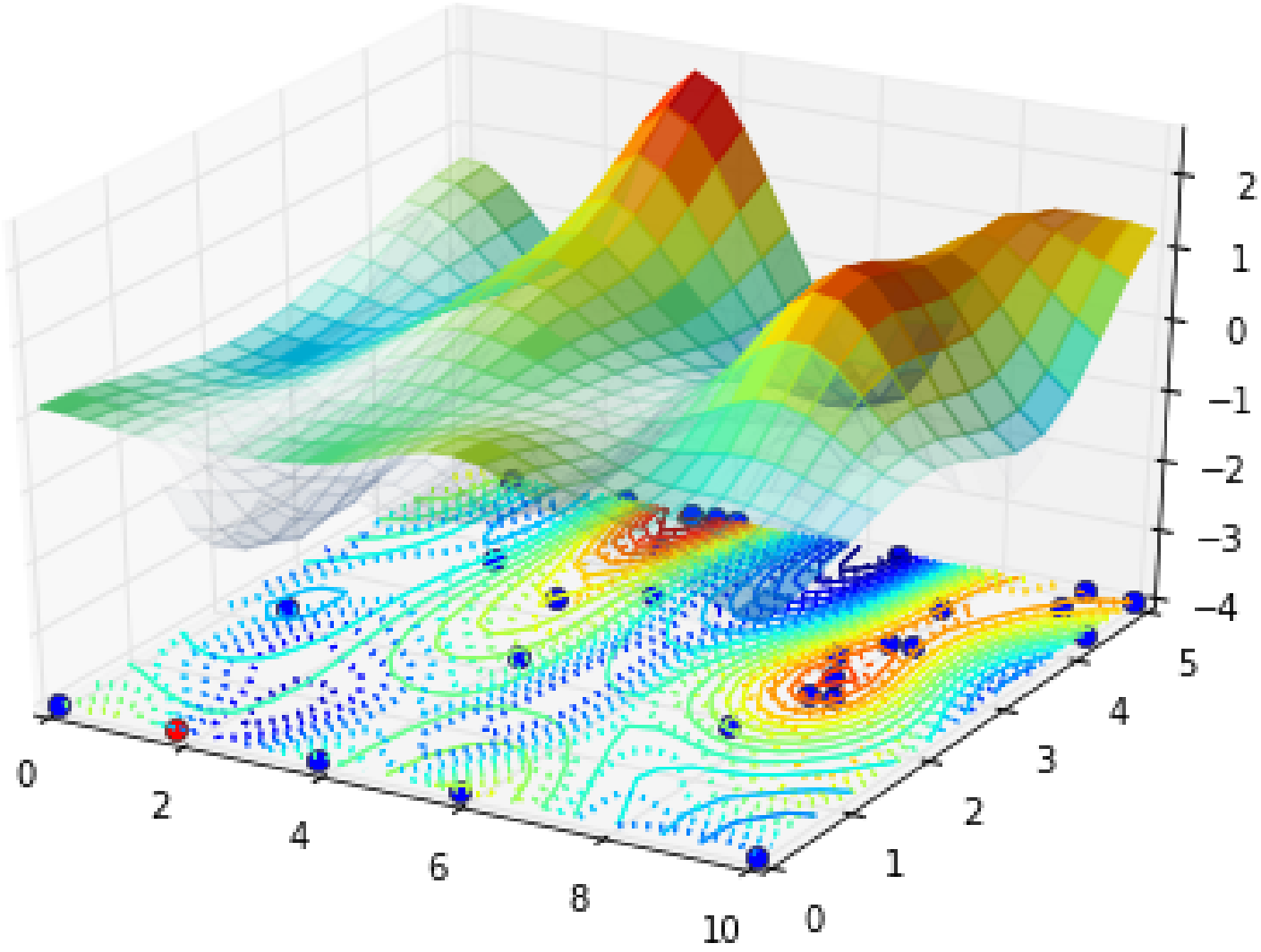
GPO (2D)



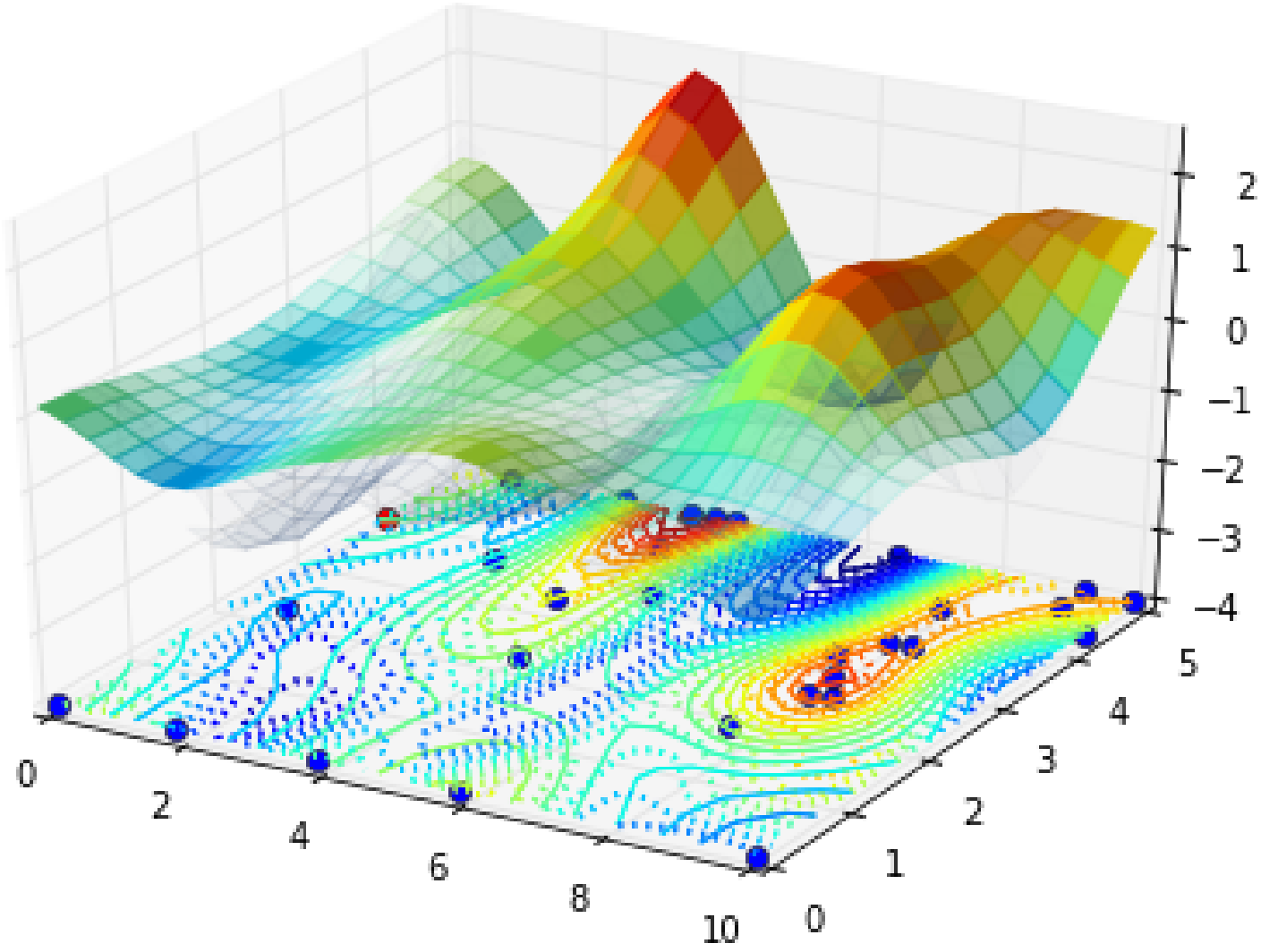
GPO (2D)



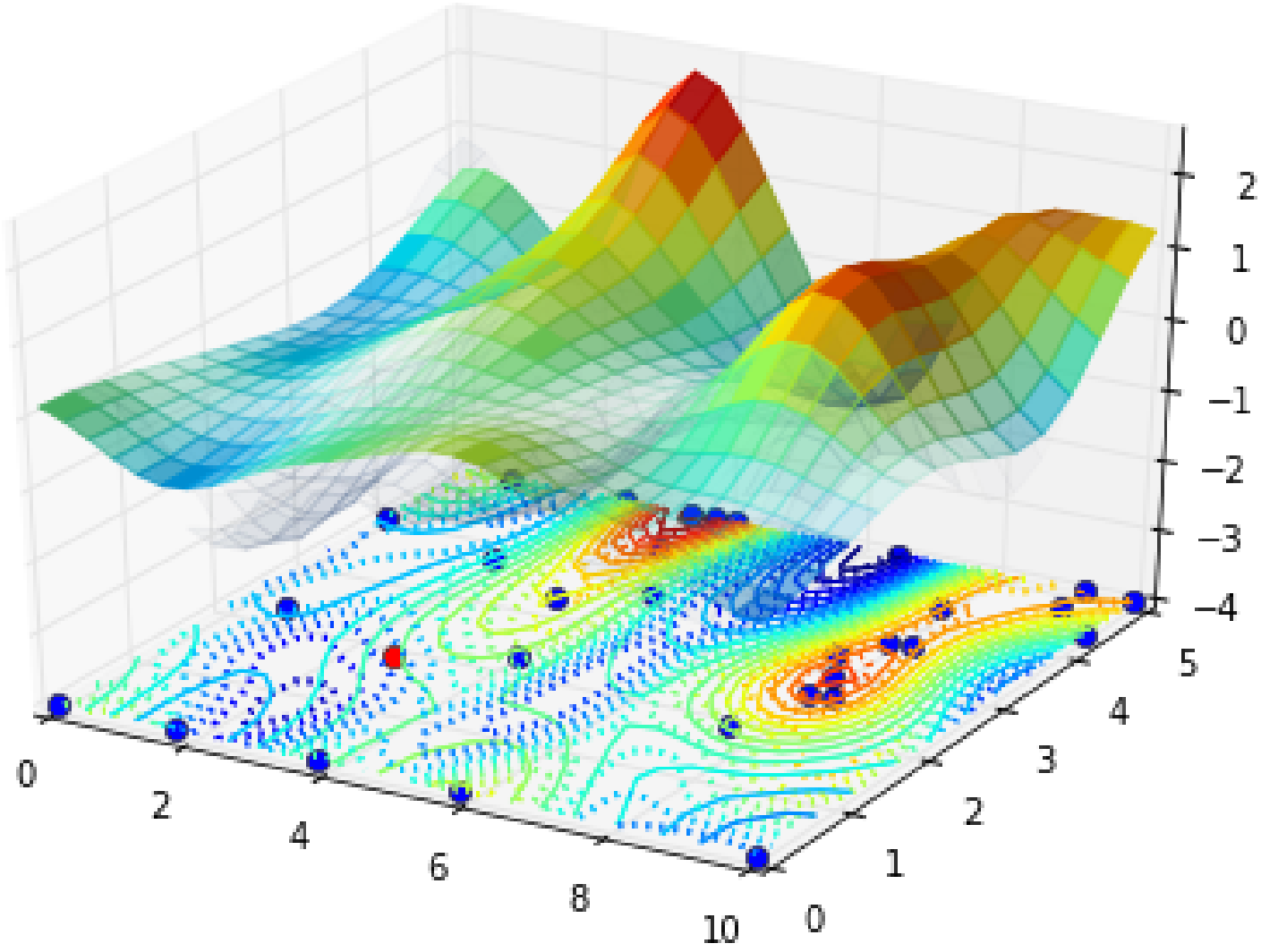
GPO (2D)



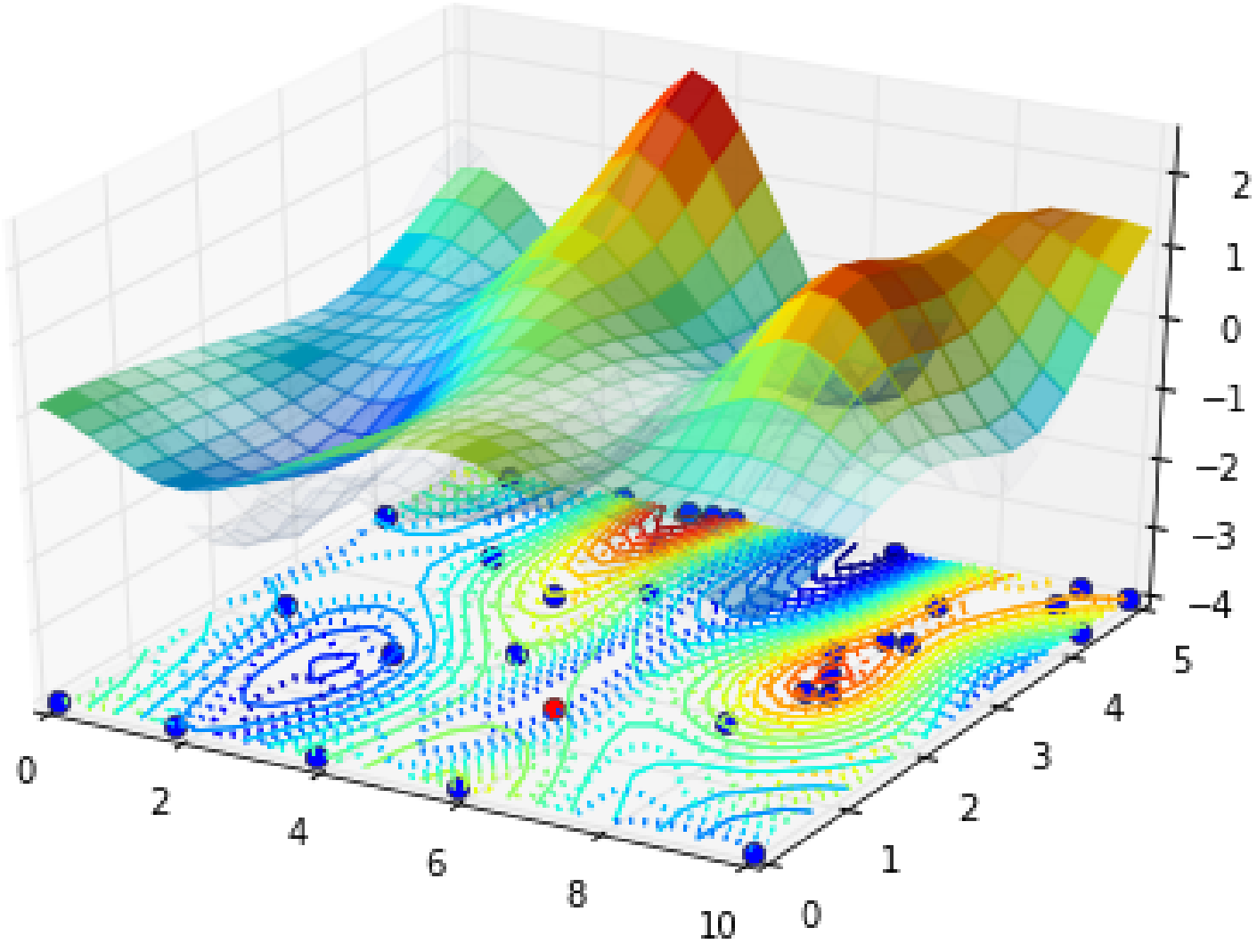
GPO (2D)



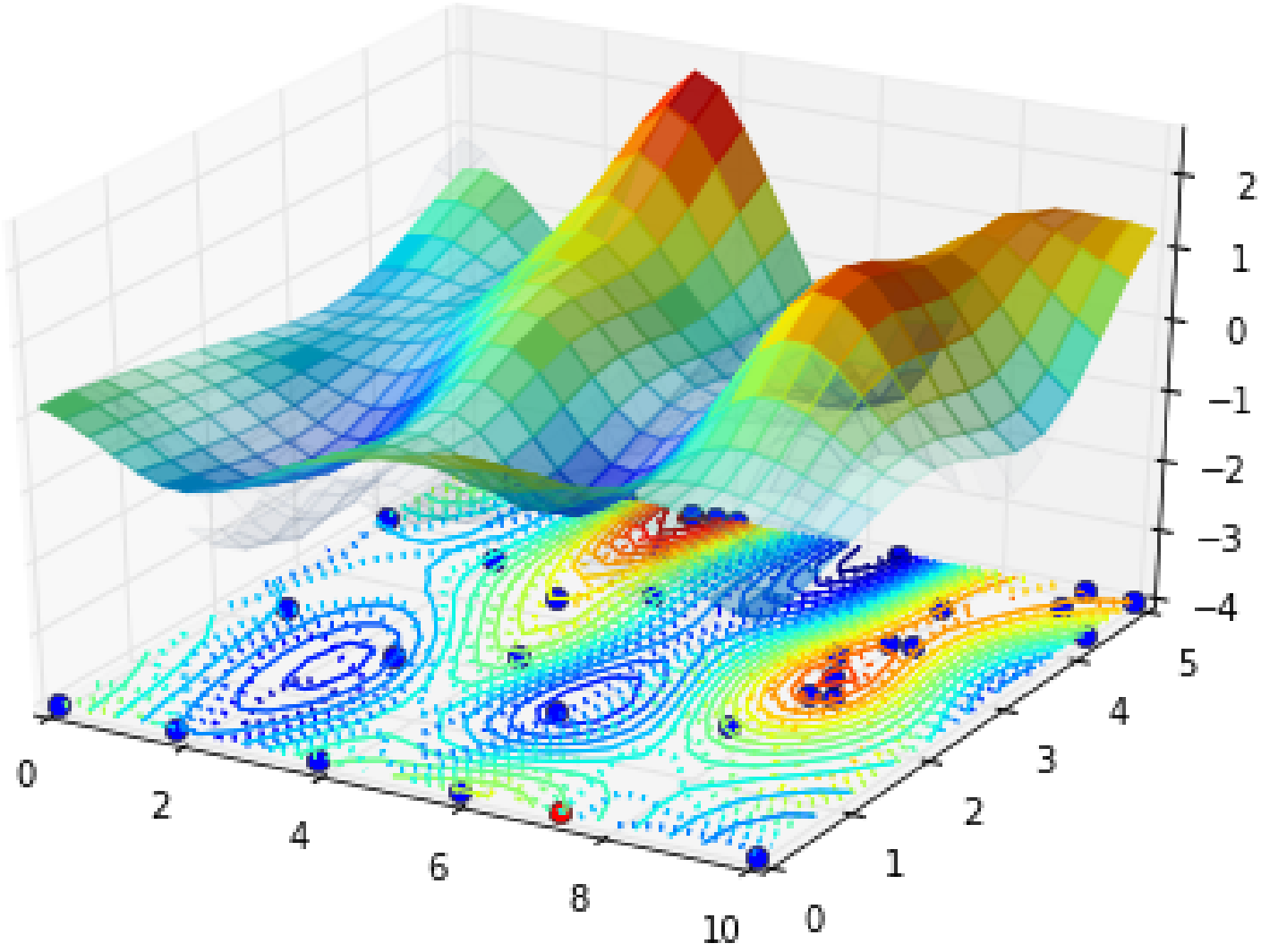
GPO (2D)



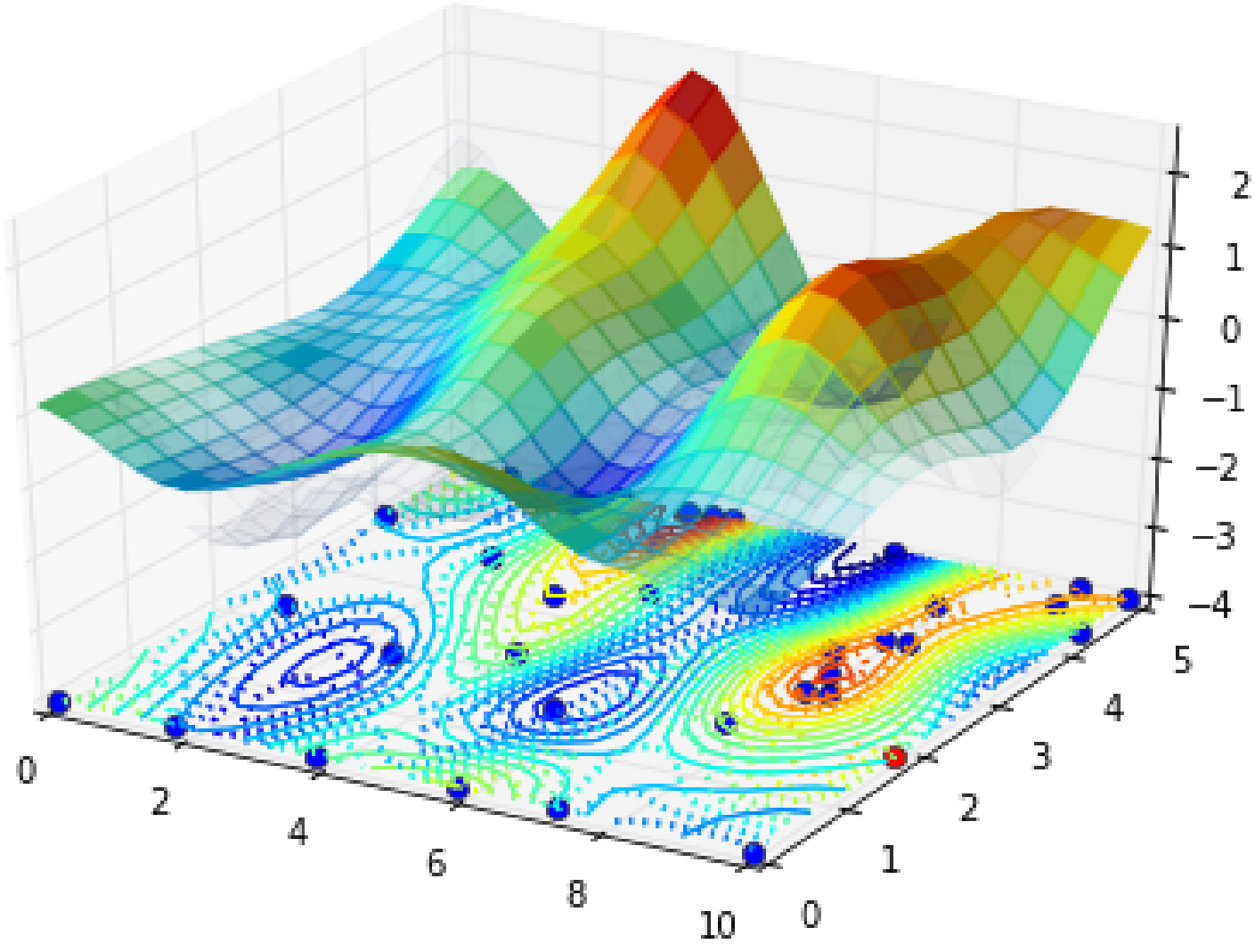
GPO (2D)



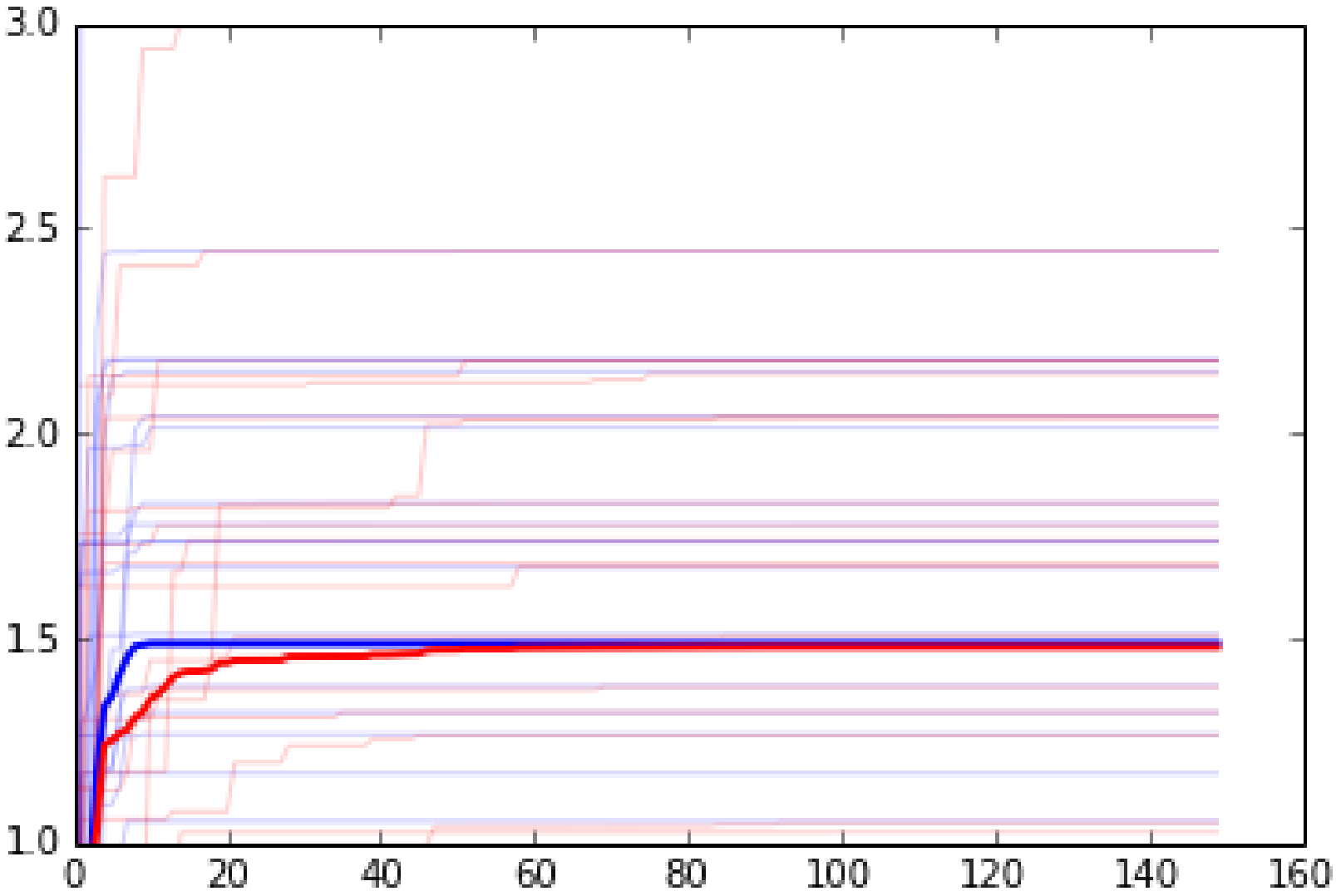
GPO (2D)



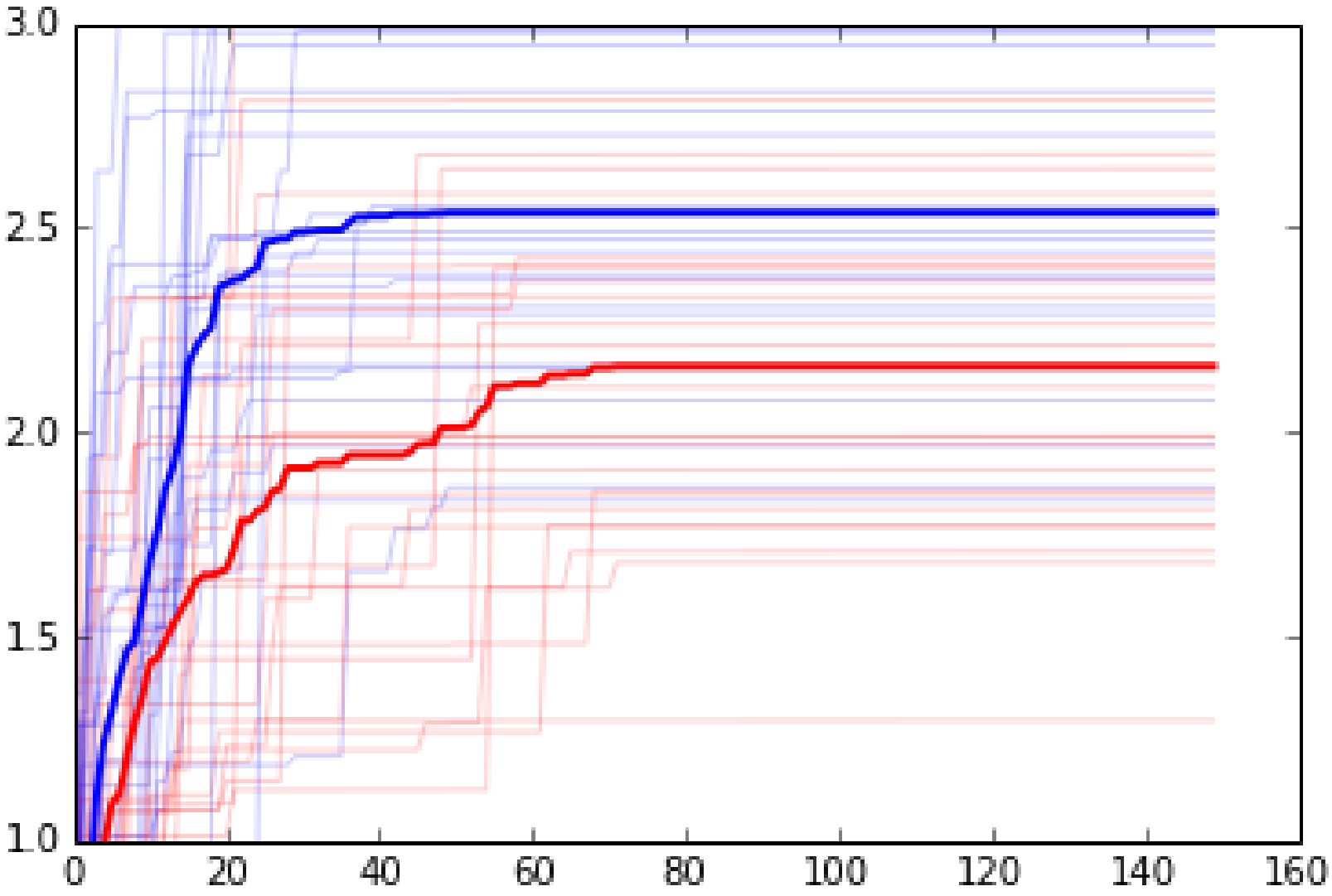
GPO (2D)



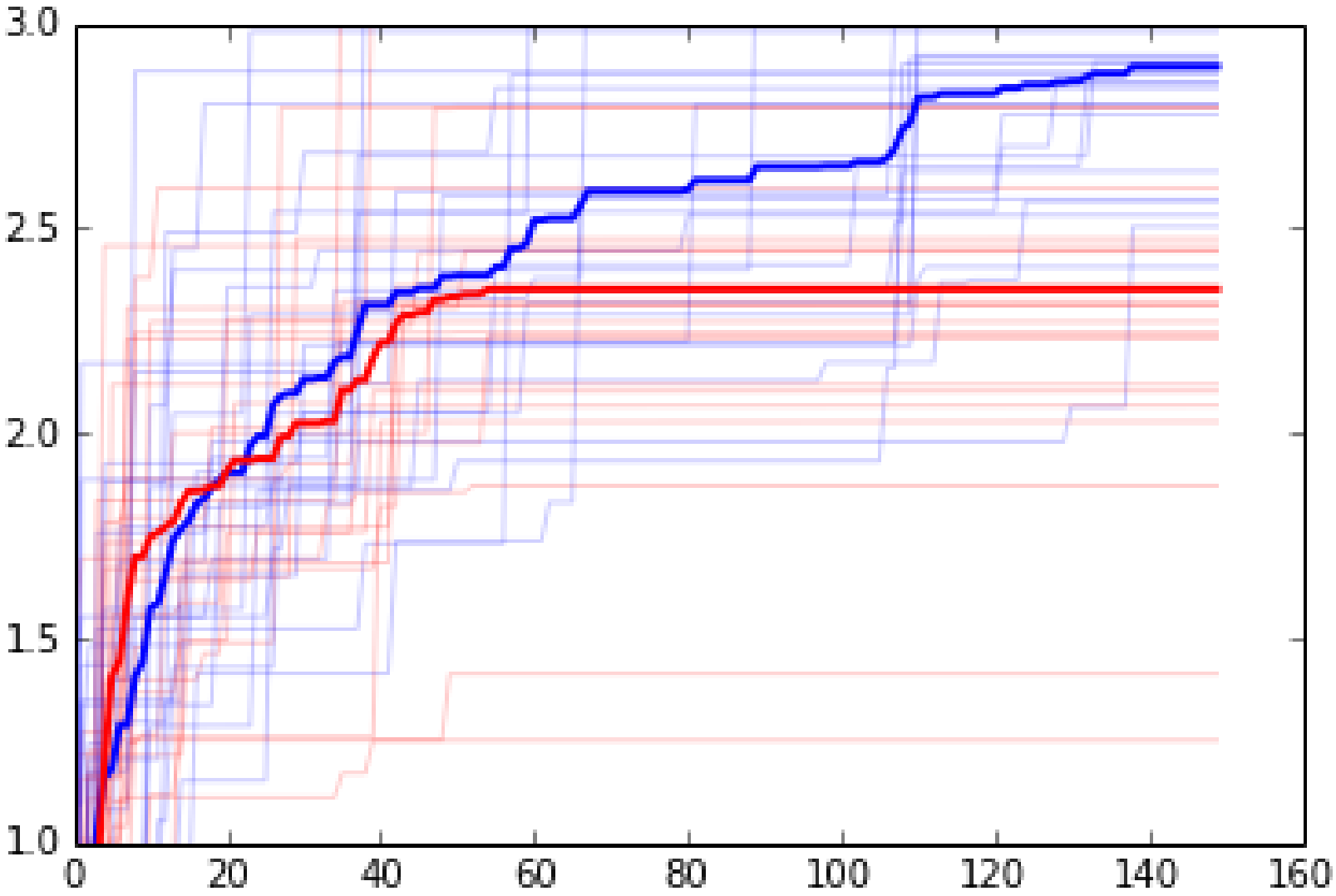
GPO vs SA (1D)



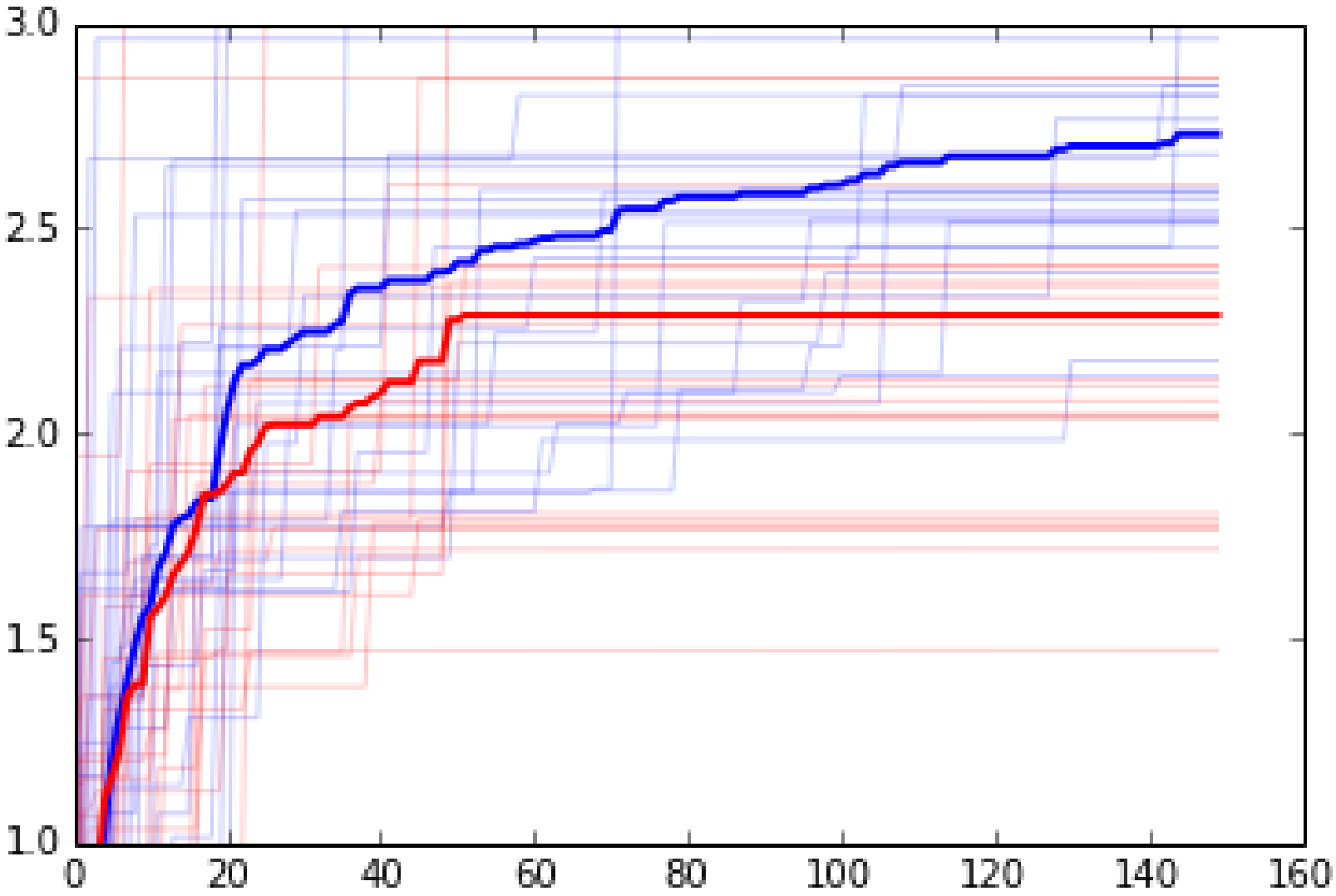
GPO vs SA (2D)



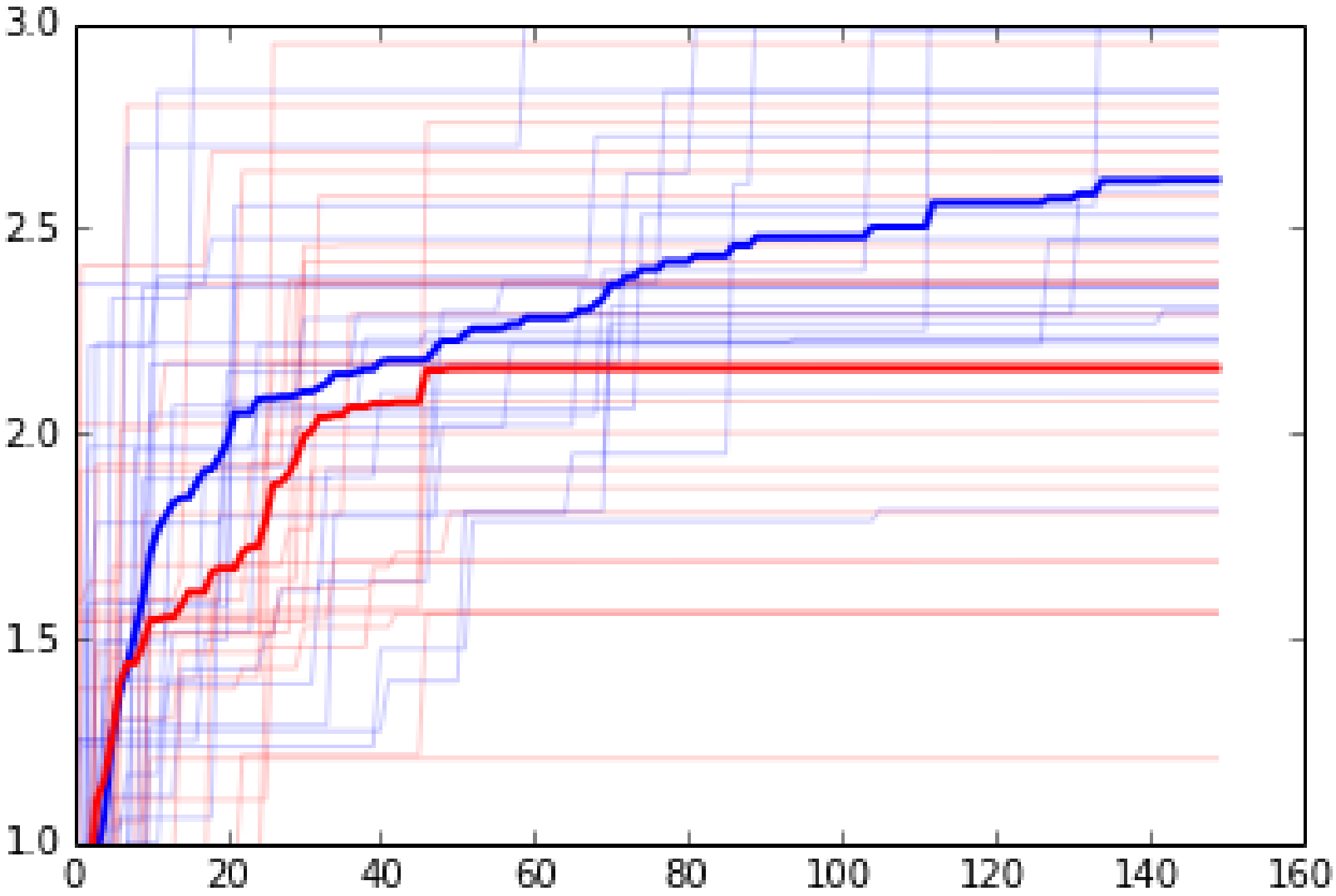
GPO vs SA (5D)



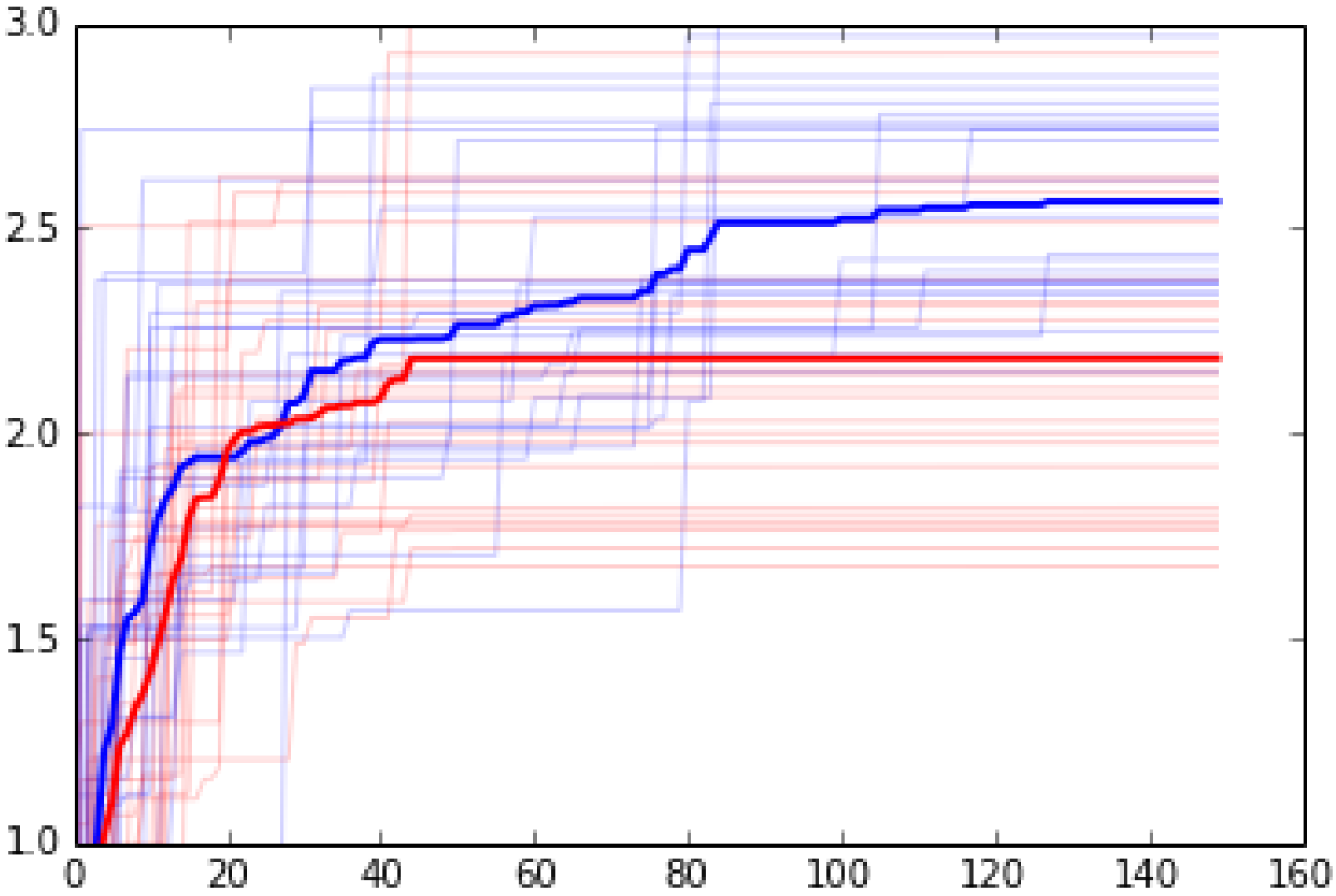
GPO vs SA (10D)



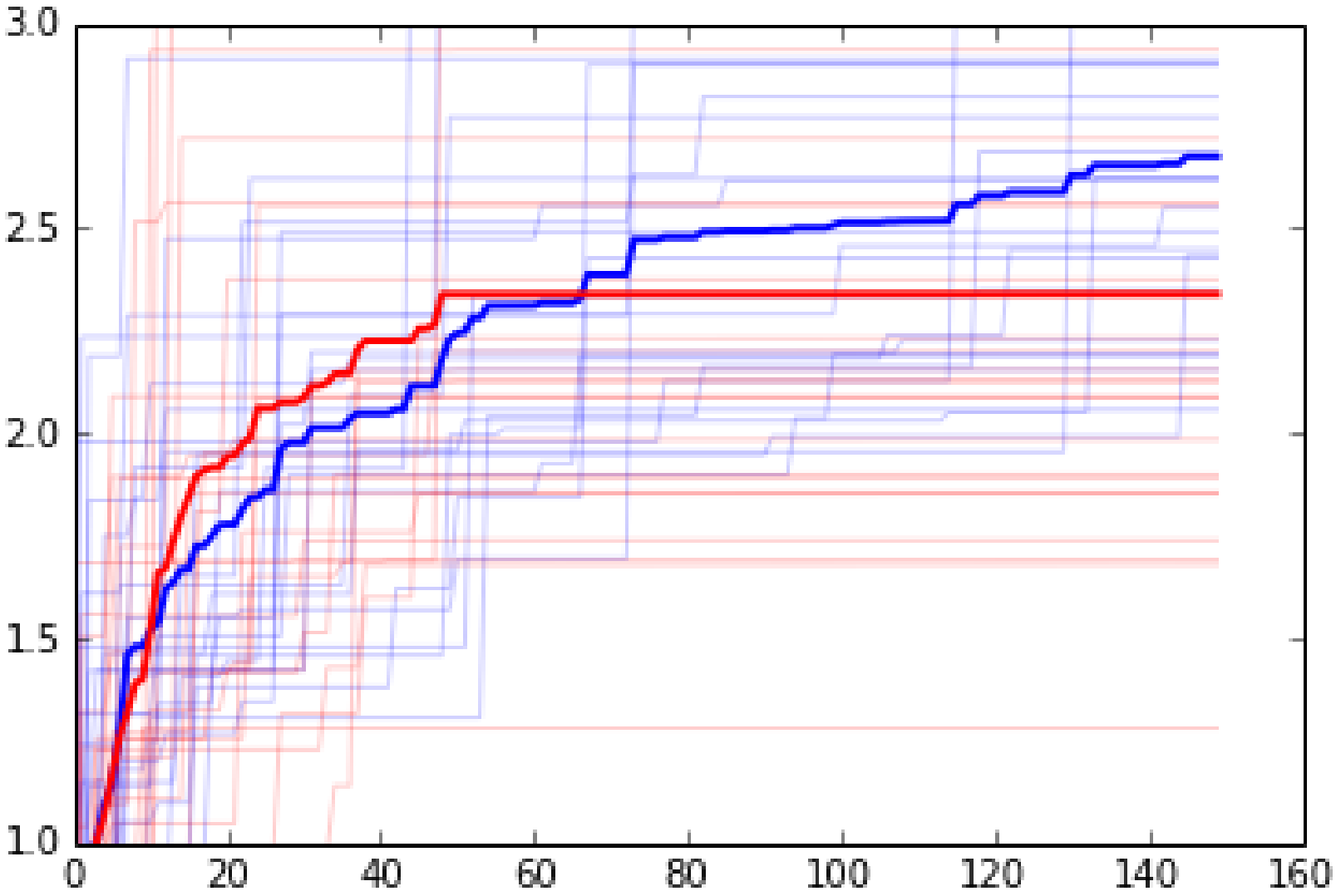
GPO vs SA (20D)



GPO vs SA (50D)



GPO vs SA (100D)



Conclusions

- works well for "small" N
- scales to high dimensions. . . but some difficulty scaling with N
- computationally expensive without a clever implementation
- one size doesn't fit all — different problems, different assumptions
  - if assumptions (or approximations) unsuitable for problem, GPO may quickly run out of promising samples
- scaling to larger N (matrix inversion)
- optimising EI (or equivalent)
- handling different utilities, observation noise
- multi-step lookahead