Probabilistic Numerics
– Part I –
Integration and Differential Equations

Philipp Hennig

MLSS 2015
18 / 07 / 2015

Emmy Noether Group on Probabilistic Numerics
Department of Empirical Inference
Max Planck Institute for Intelligent Systems
Tübingen, Germany

Information Content of Partial Computations

division with remainder

$$2\ 3\ 7\ 3\ 6 \div 7\ 3\ 6 = X\ X\ .\ X\ X$$

Information Content of Partial Computations

division with remainder

$$
\begin{array}{rrrrrcrrrcrcrr}
2 & 3 & 7 & 3 & 6 & \div & 7 & 3 & 6 & = & 3 & X & . & X & X \\
2 & 2 & 0 & 8 & 0 & & & & & & & & & & \\
 & 1 & 6 & 5 & 6 & & & & & & & & & &
\end{array}
$$

Information Content of Partial Computations

division with remainder

$$
\begin{array}{cccccccccccccc}
2 & 3 & 7 & 3 & 6 & \div & 7 & 3 & 6 & = & 3 & 2 & . & X & X \\
2 & 2 & 0 & 8 & 0 & & & & & & & & & & \\
 & 1 & 6 & 5 & 6 & & & & & & & & & & \\
 & 1 & 4 & 7 & 2 & & & & & & & & & & \\
 & & 1 & 8 & 4 & & & & & & & & & &
\end{array}
$$

Information Content of Partial Computations

division with remainder

$$
\begin{array}{cccccccccccccc}
2 & 3 & 7 & 3 & 6 & \div & 7 & 3 & 6 & = & 3 & 2 & . & 2 & X \\
2 & 2 & 0 & 8 & 0 & & & & & & & & & & \\
 & 1 & 6 & 5 & 6 & & & & & & & & & & \\
 & 1 & 4 & 7 & 2 & & & & & & & & & & \\
 & & 1 & 8 & 4 & & & & & & & & & & \\
 & & 1 & 4 & 7 & . & 2 & & & & & & & & \\
 & & & 3 & 6 & . & 8 & & & & & & & &
\end{array}
$$

Information Content of Partial Computations

division with remainder

$$
\begin{array}{rrrrrrrrrrrrrr}
2 & 3 & 7 & 3 & 6 & \div & 7 & 3 & 6 & = & 3 & 2 & . & 2 & 5 \\
2 & 2 & 0 & 8 & 0 & & & & & & & & & & \\
 & 1 & 6 & 5 & 6 & & & & & & & & & & \\
 & 1 & 4 & 7 & 2 & & & & & & & & & & \\
 & & 1 & 8 & 4 & & & & & & & & & & \\
 & & 1 & 4 & 7 & . & 2 & & & & & & & & \\
 & & & 3 & 6 & . & 8 & & & & & & & & \\
 & & & 3 & 6 & . & 8 & & & & & & & & \\
 & & & & & & 0 & & & & & & & &
\end{array}
$$

What about ML computations?

Contemporary computational tasks are more challenging

What happens with

- a neural net if we stop the “training” of a neural network after four steps of sgd?
- . . . on only 1% of the data set?
- a GP regressor if we stop the Gauss-Jordan elemination after three steps?
- a DP mixture model if we only run MCMC for ten samples?
- a robotic controller built using all these methods?

What about ML computations?

Contemporary computational tasks are more challenging

What happens with

- a neural net if we stop the “training” of a neural network after four steps of sgd?
- . . . on only 1% of the data set?
- a GP regressor if we stop the Gauss-Jordan elemination after three steps?
- a DP mixture model if we only run MCMC for ten samples?
- a robotic controller built using all these methods?

As data-sets becomes infinite, ML models increasingly complex, and their applications permeate our lives,
we need to model effects of approximations more explicitly to achieve fast, reliable AI.

Machine learning methods are chains of numerical computations

- linear algebra (least-squares)
- optimization (training & fitting)
- integration (MCMC, marginalization)
- solving differential equations (RL, control)

Are these methods just black boxes on your shelf?

Numerical methods perform inference

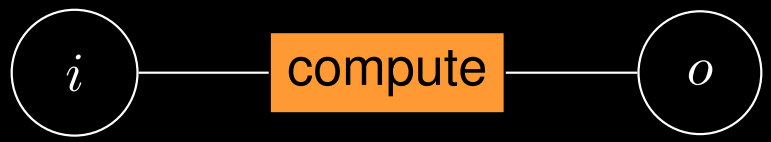
an old observation [Poincaré 1896, Diaconis 1988, O'Hagan 1992]

A numerical method
estimates a function's latent property
given the result of computations.

integration	estimates $\int_a^b f(x)\,dx$	given $\{f(x_i)\}$
linear algebra	estimates x s.t. $Ax = b$	given $\{As = y\}$
optimization	estimates x s.t. $\nabla f(x) = 0$	given $\{\nabla f(x_i)\}$
analysis	estimates $x(t)$ s.t. $x' = f(x,t)$,	given $\{f(x_i,t_i)\}$

- computations yield "data" / "observations"
- non-analytic quantities are "latent"
- even deterministic quantities can be uncertain.

If computation is inference, it should be possible to build
probabilistic numerical methods
that take in probability measures over inputs,
and return probability measures over outputs,
which quantify uncertainty arising from the uncertain input
and the finite information content of the computation.



Classic methods identified as maximum a-posteriori

probabilistic numerics is anchored in established theory

quadrature [Diaconis 1988]

Gaussian quadrature ⟷ Gaussian process regression

linear algebra [Hennig 2014]

conjugate gradients ⟷ Gaussian conditioning

nonlinear optimization [Hennig & Kiefel 2013]

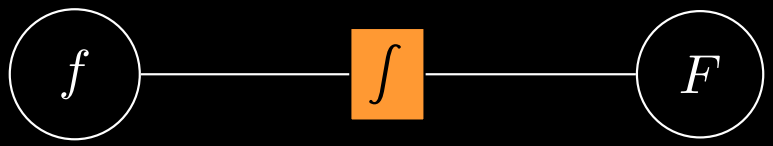
BFGS ⟷ autoregressive filtering

ordinary differential equations [Schober, Duvenaud & Hennig 2014]

Runge-Kutta ⟷ Gauss-Markov extrapolation

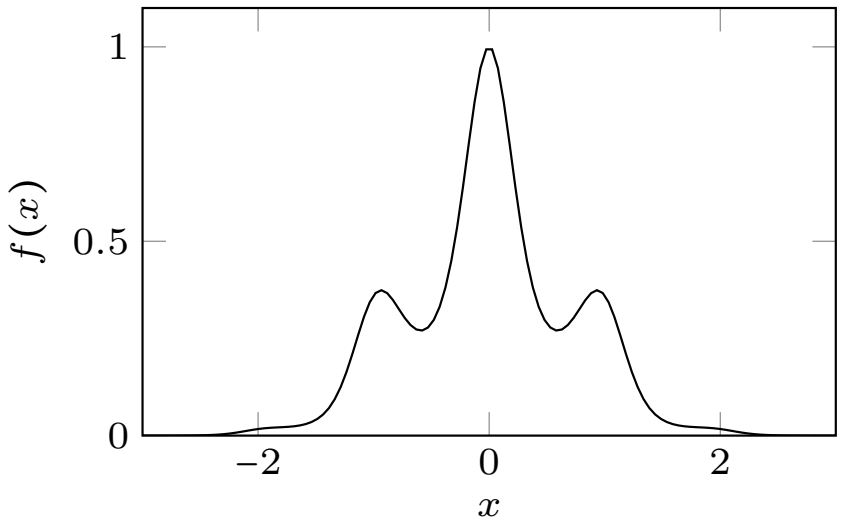
Integration

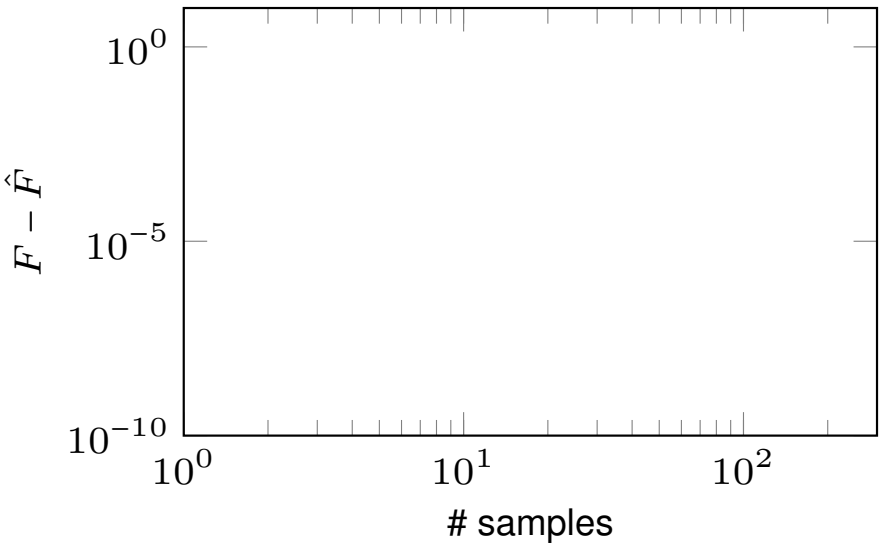
$$F = \int_a^b f(x)\,dx$$



Integration

a toy problem



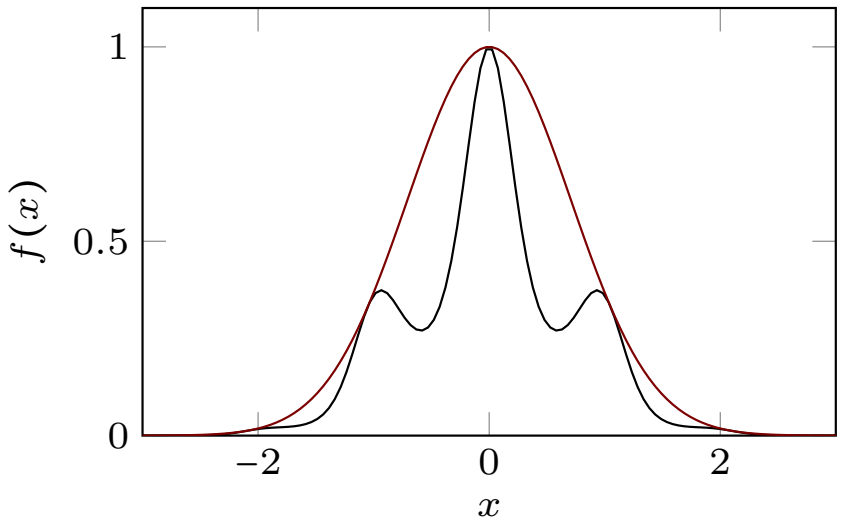


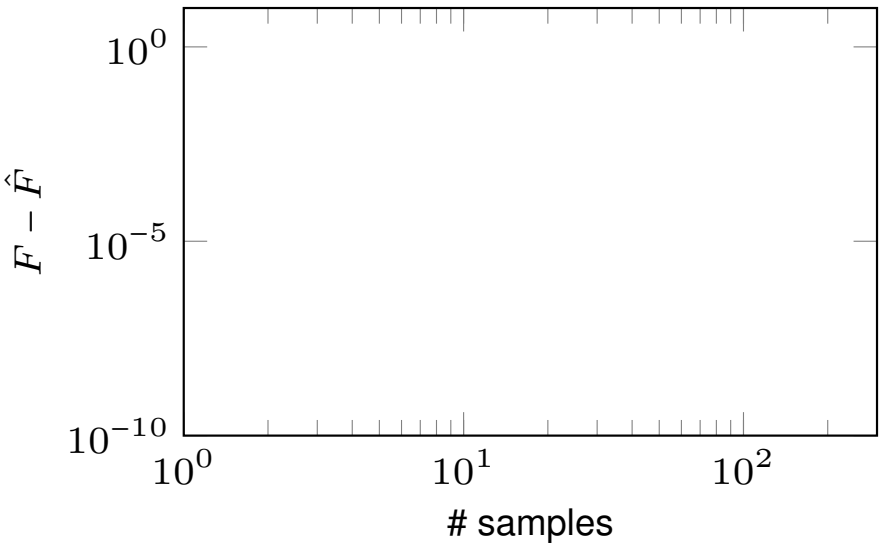
$$f(x) = \exp\left(-\sin^2(3x) - x^2\right)$$

$$F = \int_{-3}^{3} f(x)dx = ?$$

Integration

a toy problem





$$f(x) = \exp\left(-\sin^2(3x) - x^2\right)$$

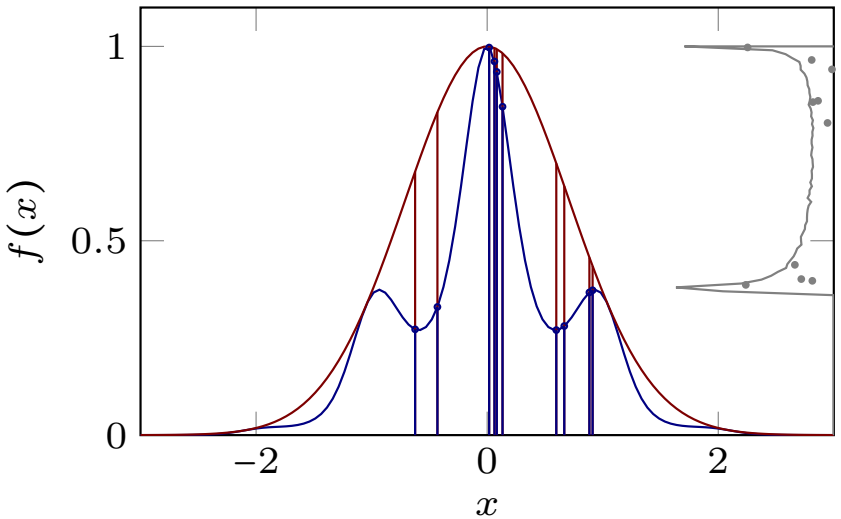
$$\leq \exp(-x^2)$$

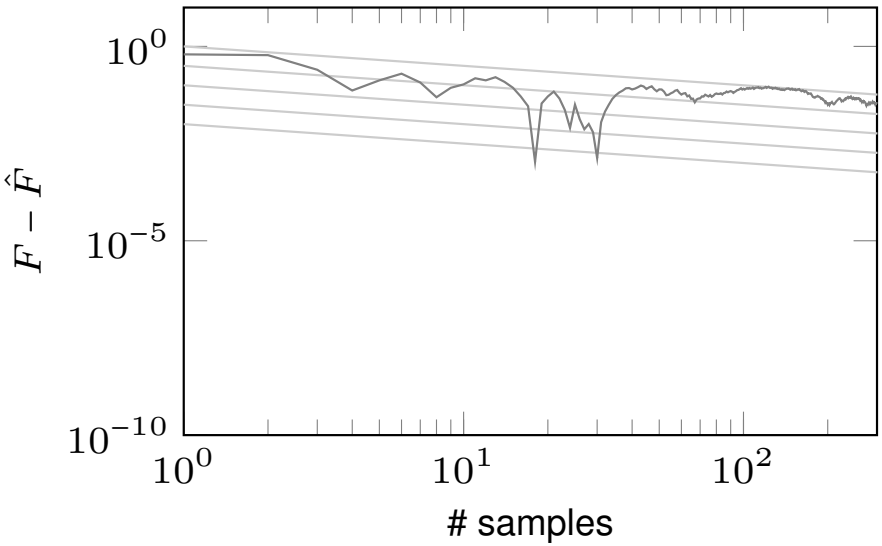
$$F = \int_{-3}^{3} f(x)dx = ?$$

$$\int \exp(-x^2) = \sqrt{\pi}$$

Monte Carlo

(almost) no assumptions, stochastic convergence





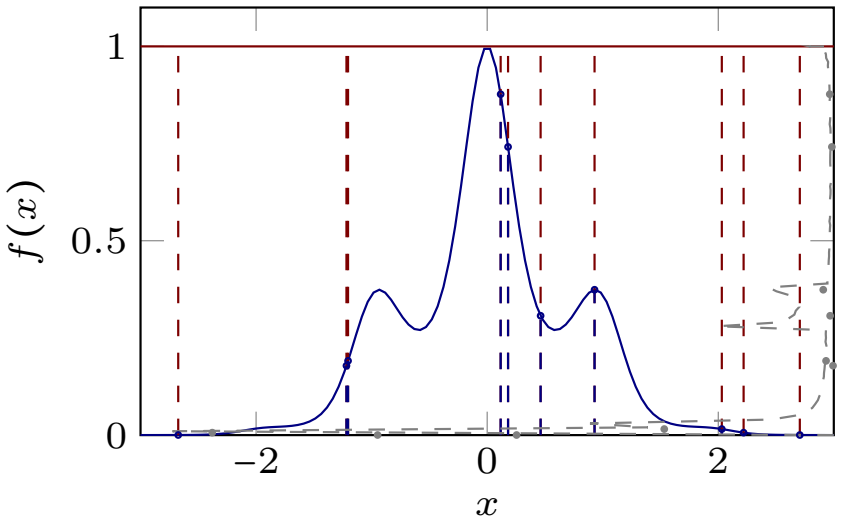
$$F = \int \exp\left(-\sin^2(3x) - x^2\right) dx$$

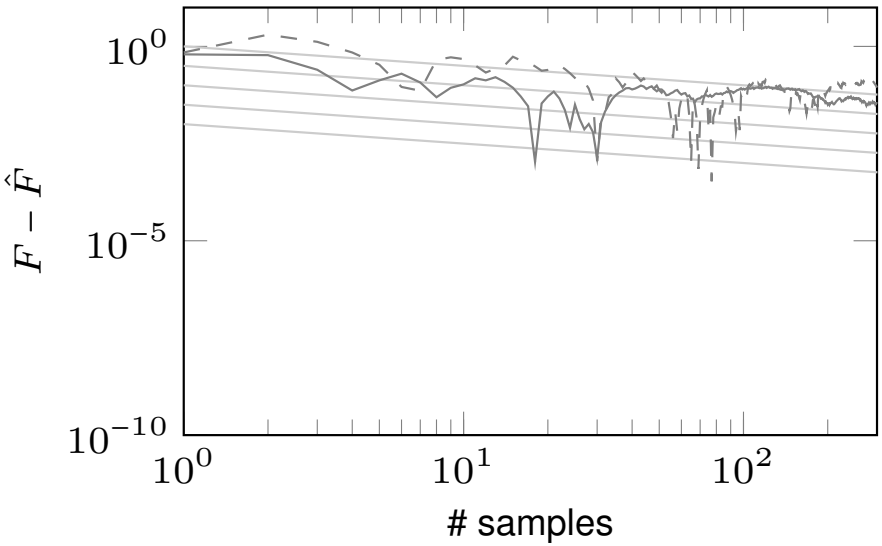
$$= Z \int \frac{f(x)}{g(x)} \frac{g(x)}{Z} dx \qquad Z = \int g(x) dx$$

$$\approx Z \frac{1}{N} \sum_i^N \frac{f(x_i)}{g(x_i)} = \hat{F} \qquad x_i \sim \frac{g(x)}{Z} \qquad \operatorname{var} \hat{F} = \frac{\operatorname{var}_g(f/g)}{N}$$

Monte Carlo

(almost) no assumptions, stochastic convergence





$$F = \int \exp\left(-\sin^2(3x) - x^2\right) dx$$

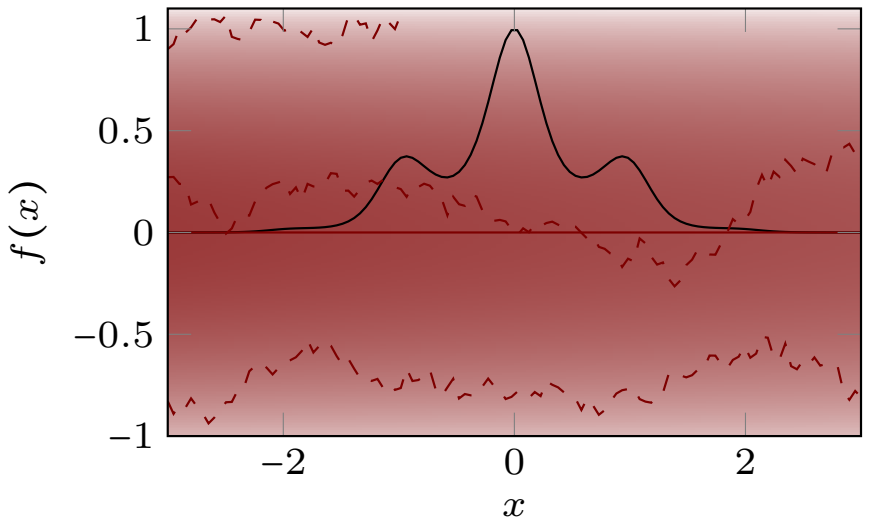
$$= Z \int \frac{f(x)}{g(x)} \frac{g(x)}{Z} dx \qquad Z = \int g(x) dx$$

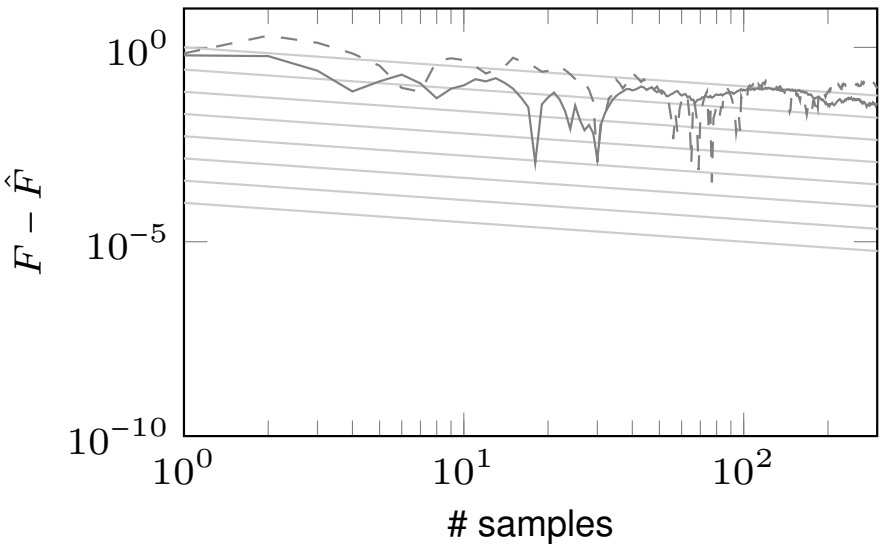
$$\approx Z \frac{1}{N} \sum_i^N \frac{f(x_i)}{g(x_i)} = \hat{F} \qquad x_i \sim \frac{g(x)}{Z} \qquad \operatorname{var} \hat{F} = \frac{\operatorname{var}_g(f/g)}{N}$$

- adding randomness enforces stochastic convergence

The probabilistic approach

integration as nonparametric inference [P. Diaconis, 1988, T. O'Hagan, 1991]





$$p(f) = \mathcal{GP}(f; 0, k) \qquad k(x, x') = \min(x, x') + b$$

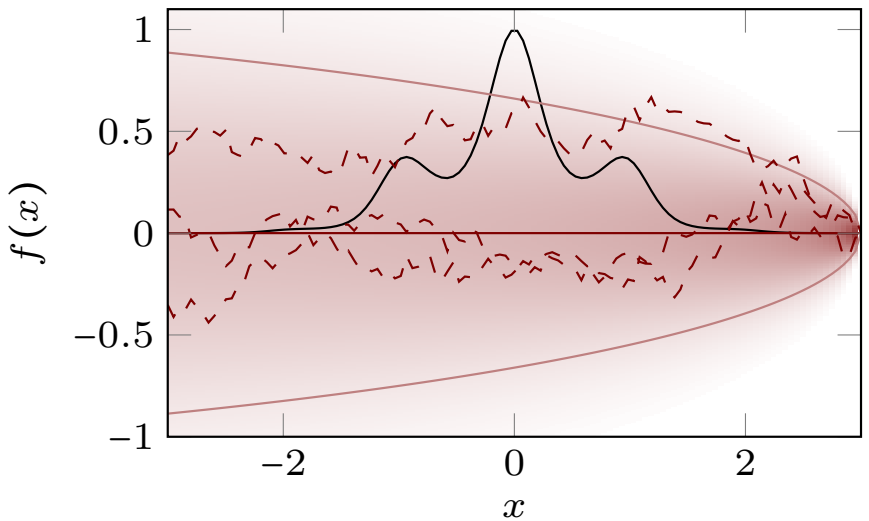
$$p(z) = \mathcal{N}(z; \mu, \Sigma) \quad \Rightarrow \quad p(Az) = \mathcal{N}(Az; A\mu, A\Sigma A^\top)$$

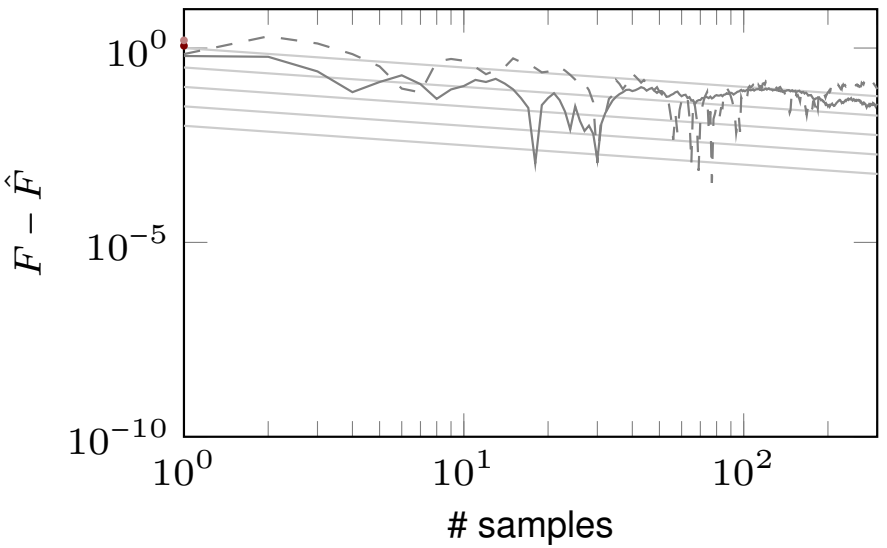
$$p\left(\int_a^b f(x)\,dx\right) = \mathcal{N}\left[\int_a^b f(x)\,dx; \int_a^b m(x)\,dx, \iint_a^b k(x, x')\,dx\,dx'\right]$$

$$= \mathcal{N}(F; 0, -1/6(b^3 - a^3) + 1/2[b^3 - 2a^2b + a^3] - (b - a)^2 c)$$

Active Collection of Information

choise of evaluation nodes [T. Minka, 2000]

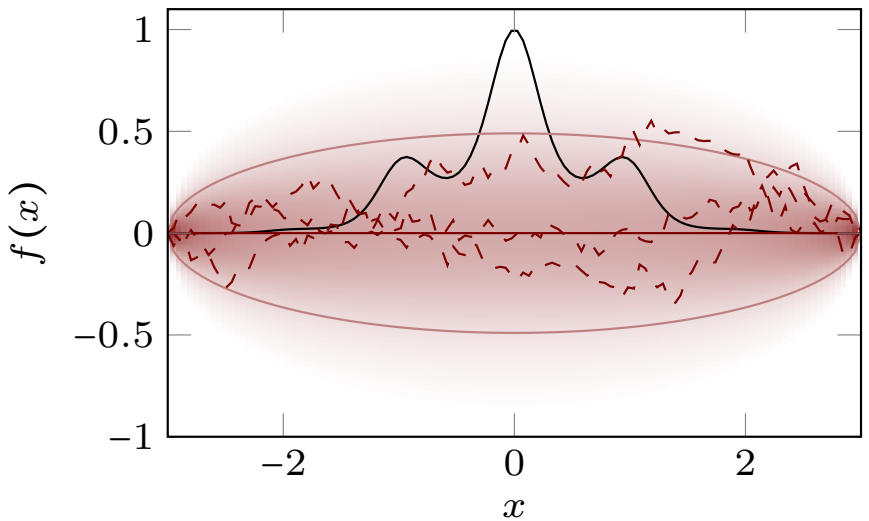


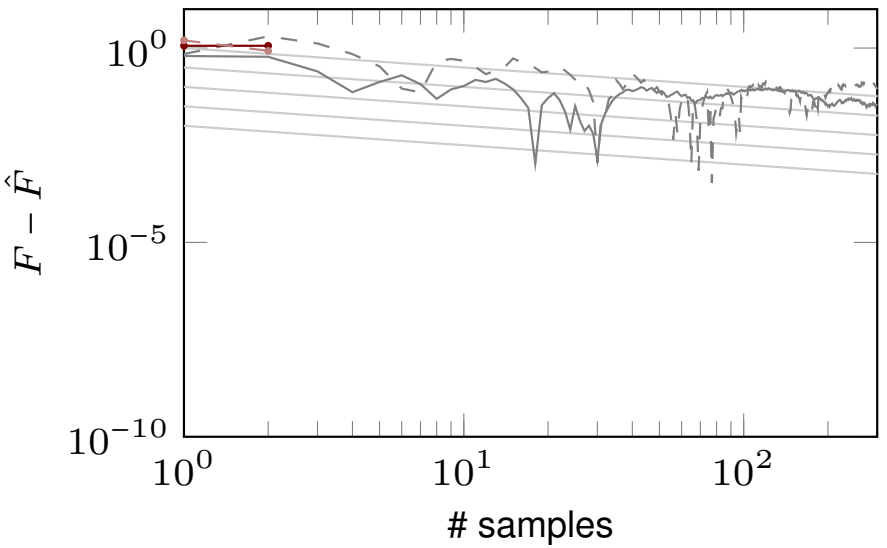


$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

Active Collection of Information

choise of evaluation nodes [T. Minka, 2000]

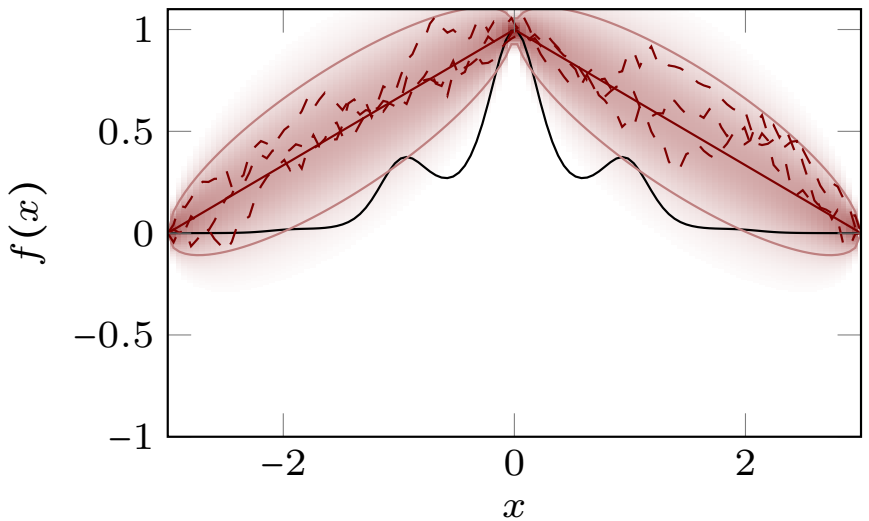


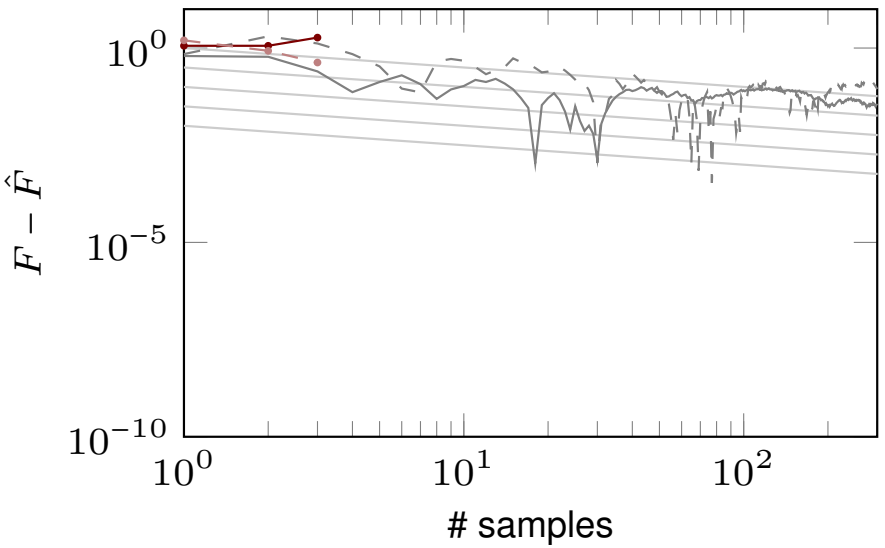


$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

Active Collection of Information

choise of evaluation nodes [T. Minka, 2000]



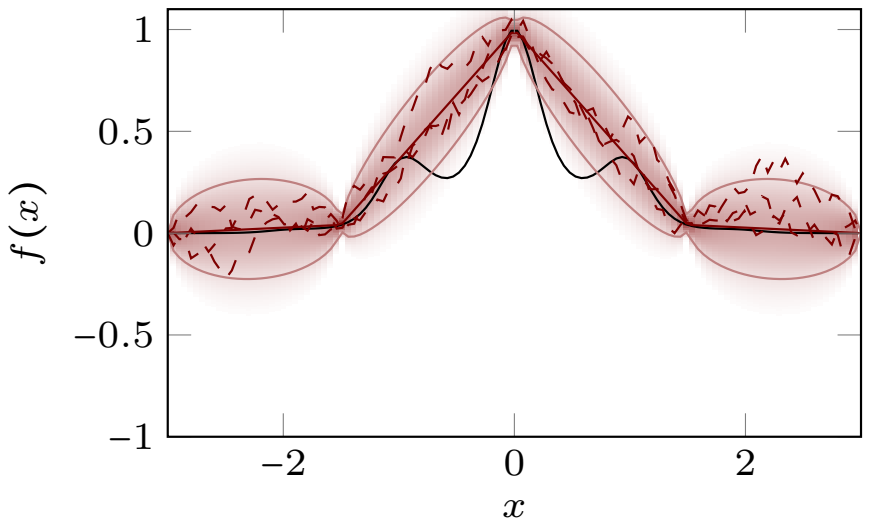


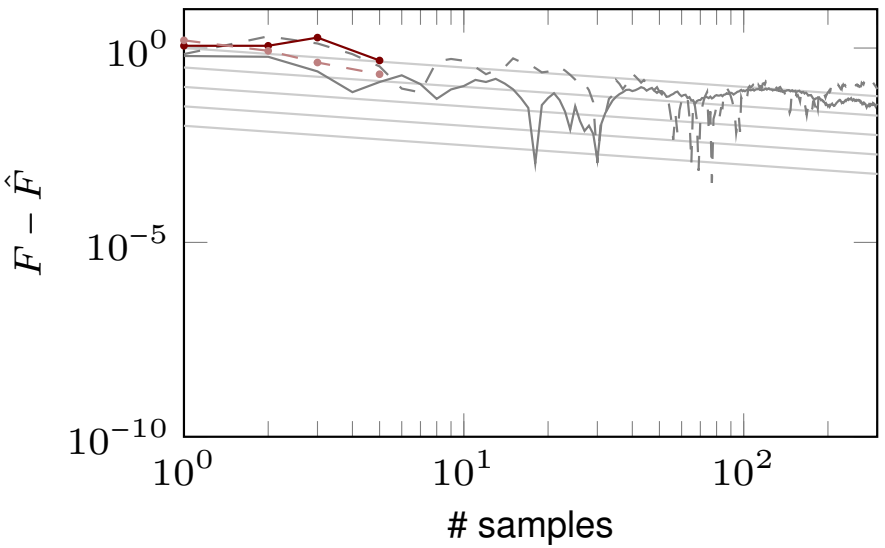
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

active node placement for maximum expected error reduction gives regular grid.

Active Collection of Information

choise of evaluation nodes [T. Minka, 2000]



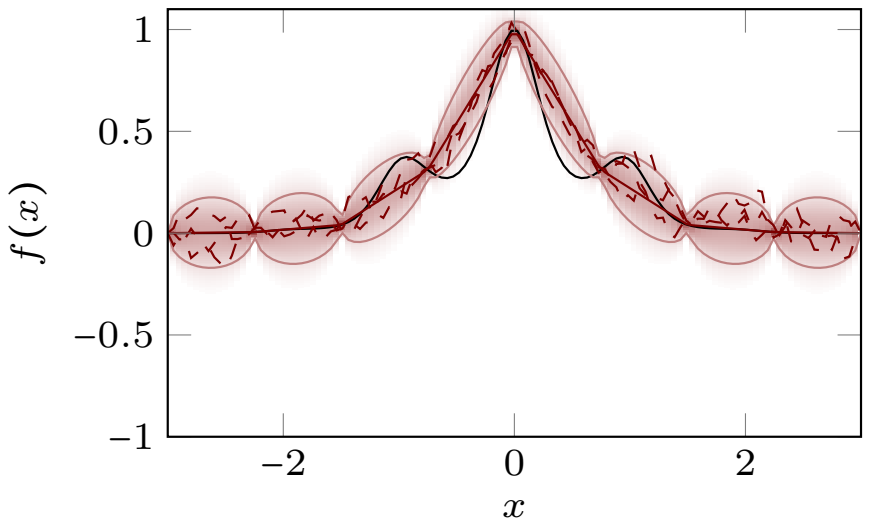


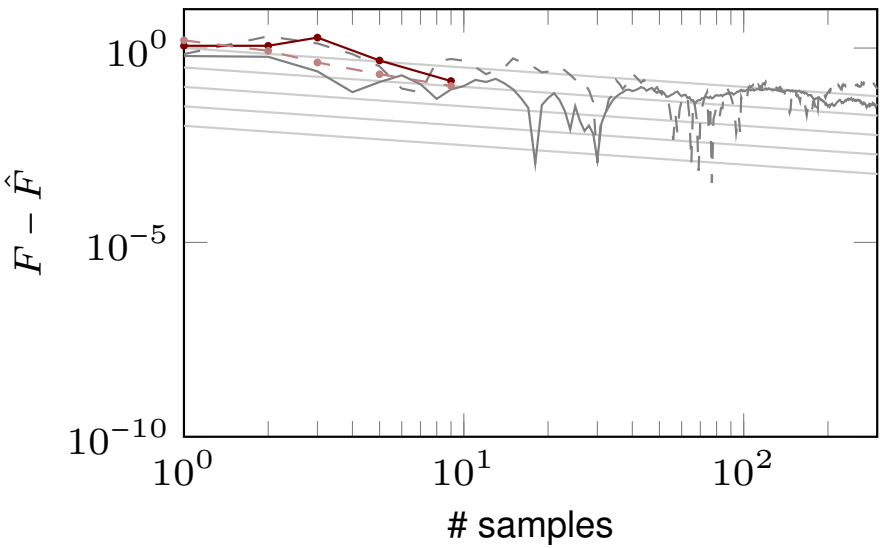
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

active node placement for maximum expected error reduction gives regular grid.

Active Collection of Information

choise of evaluation nodes [T. Minka, 2000]



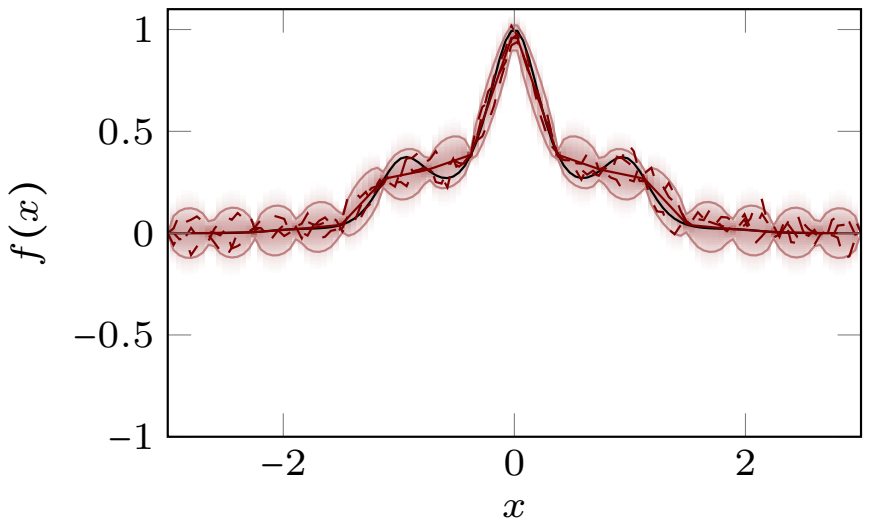


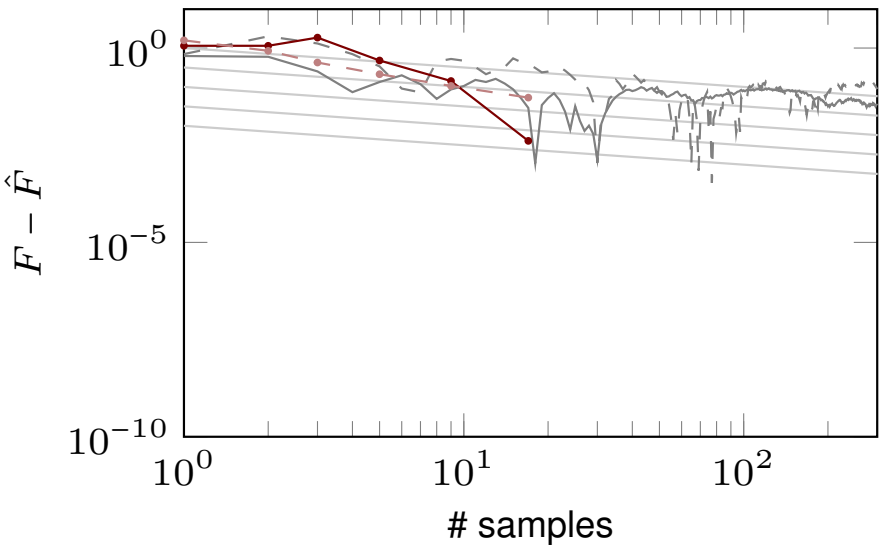
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

active node placement for maximum expected error reduction gives regular grid.

Active Collection of Information

choise of evaluation nodes [T. Minka, 2000]



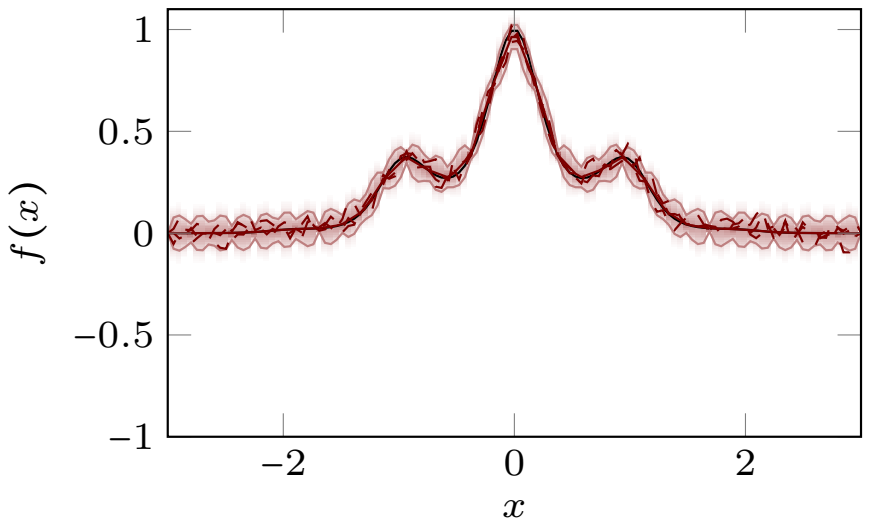


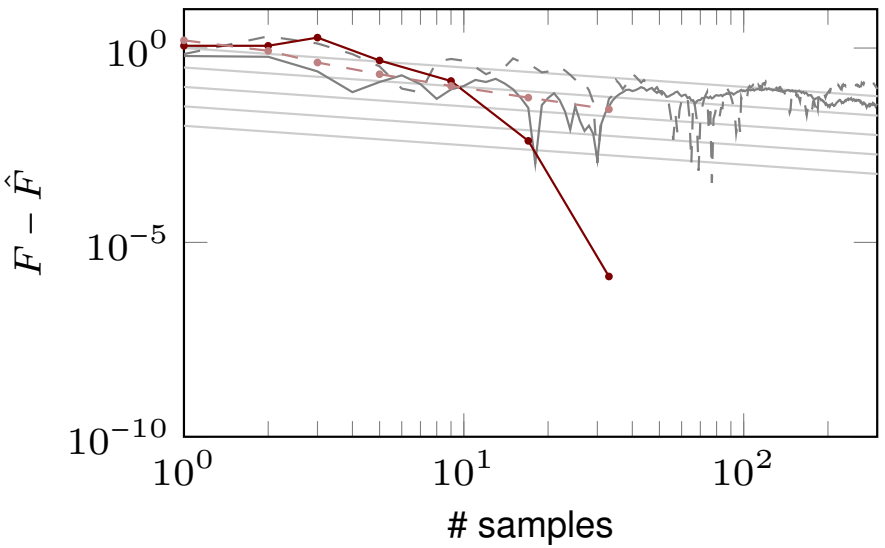
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

active node placement for maximum expected error reduction gives regular grid.

Active Collection of Information

choise of evaluation nodes [T. Minka, 2000]



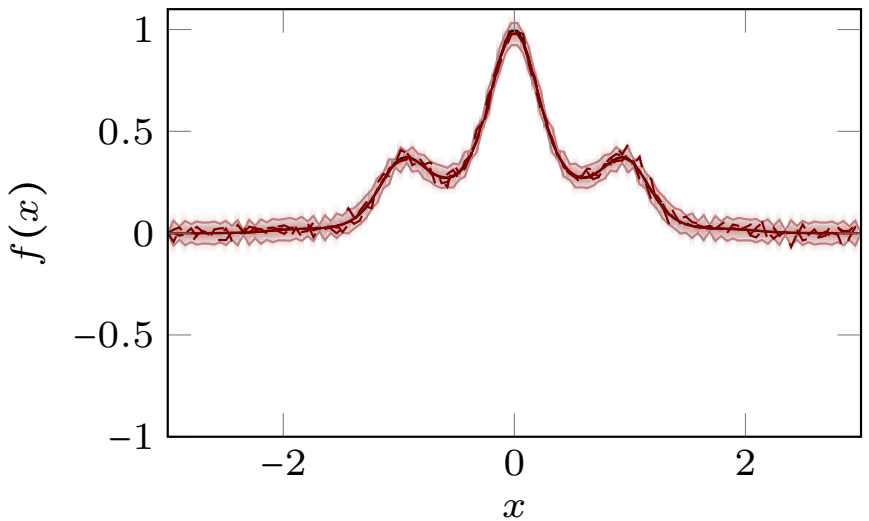


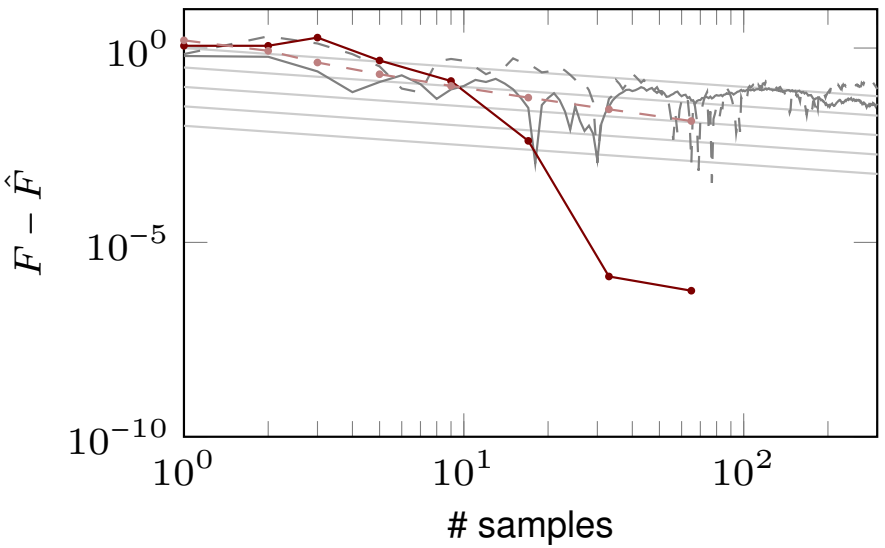
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

active node placement for maximum expected error reduction gives regular grid.

Active Collection of Information

choise of evaluation nodes [T. Minka, 2000]



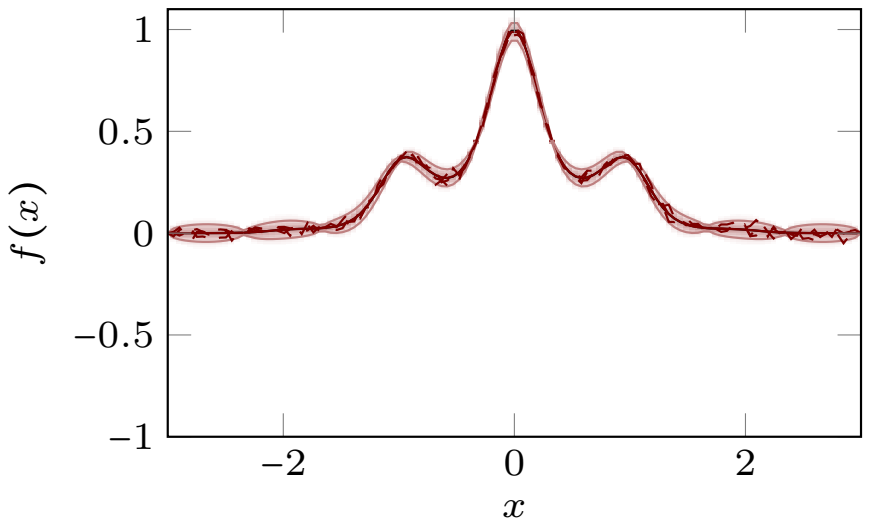


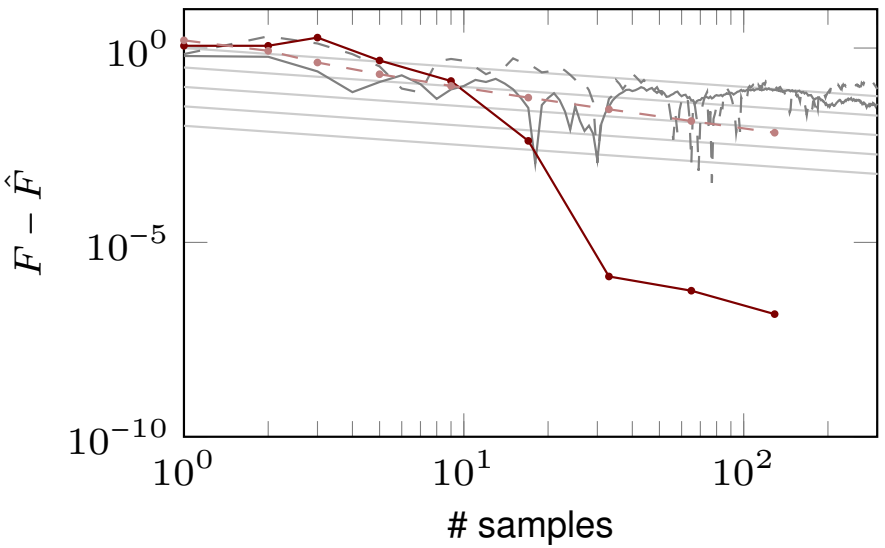
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

active node placement for maximum expected error reduction gives regular grid.

Active Collection of Information

choise of evaluation nodes [T. Minka, 2000]



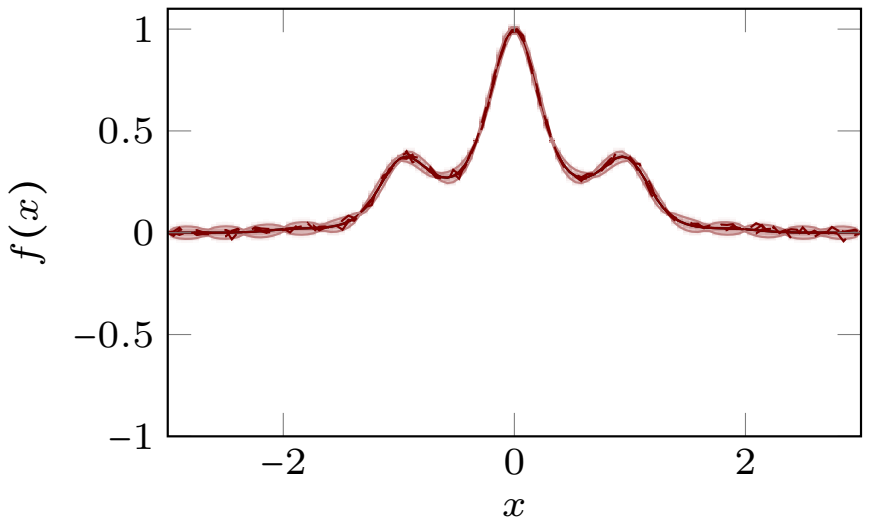


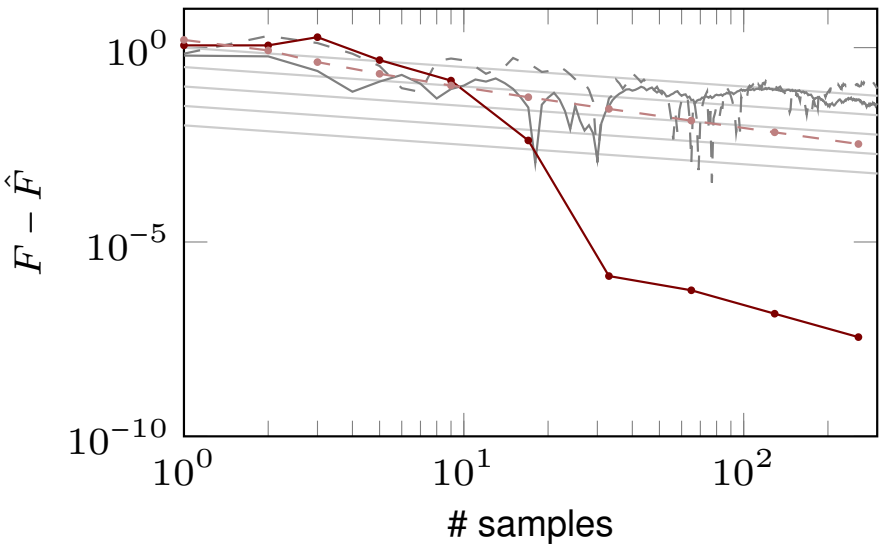
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

active node placement for maximum expected error reduction gives regular grid.

Active Collection of Information

choise of evaluation nodes [T. Minka, 2000]



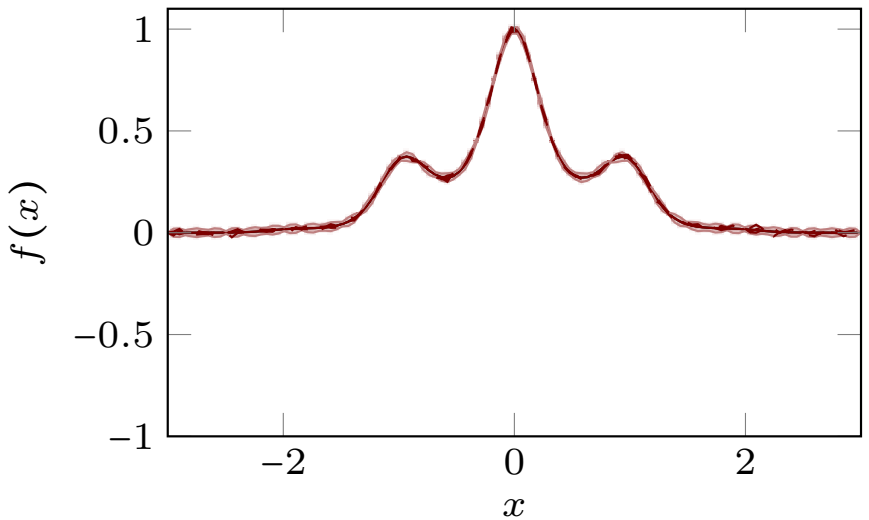


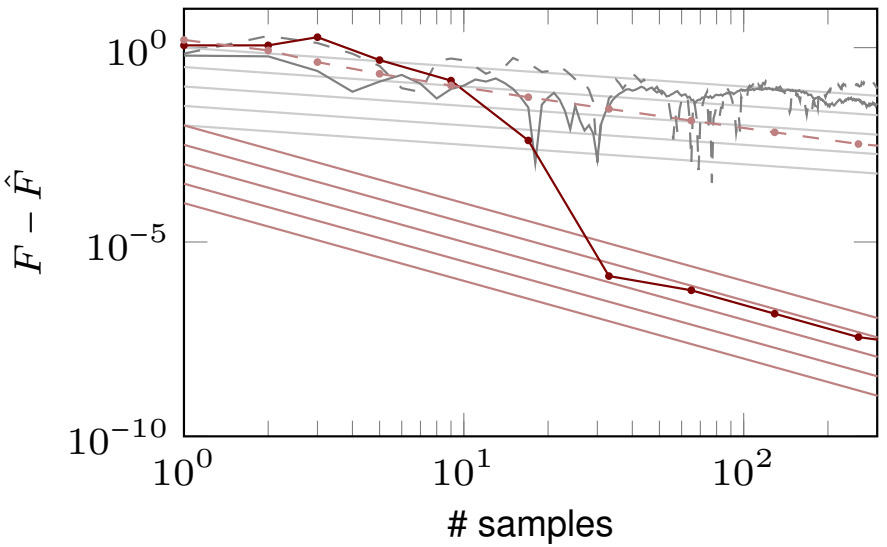
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

active node placement for maximum expected error reduction gives regular grid.

Active Collection of Information

choise of evaluation nodes [T. Minka, 2000]



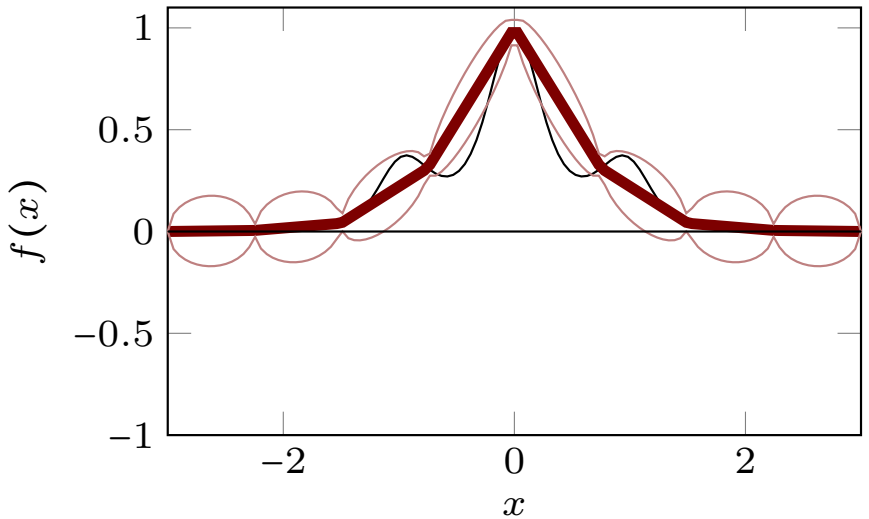


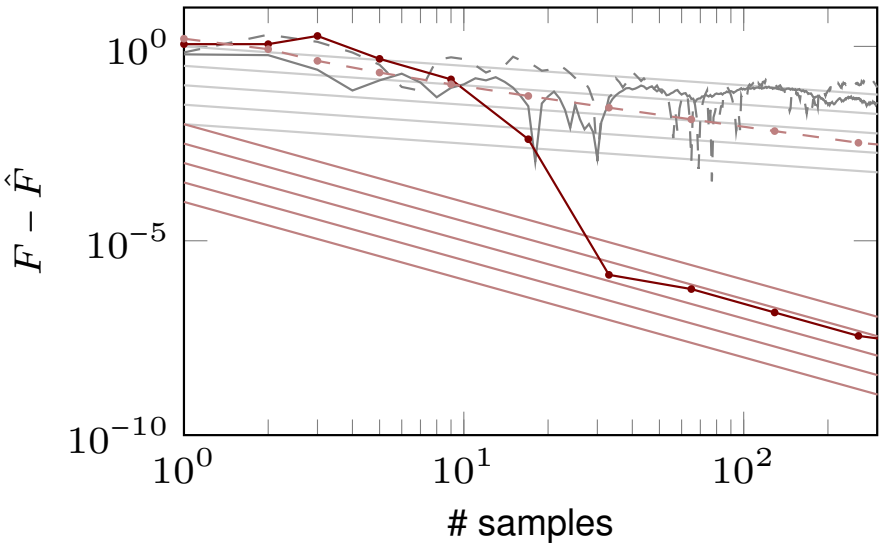
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

active node placement for maximum expected error reduction gives regular grid.

Posterior Mean is a Linear Spline. . .

A classic numerical method, derived as a learning machine [P. Diaconis, 1988, T. O'Hagan, 1991]





$$p(f \mid \boldsymbol{y}, \boldsymbol{x}) = \mathcal{GP}(f; m_{\boldsymbol{y}}, k_{\boldsymbol{y}})$$

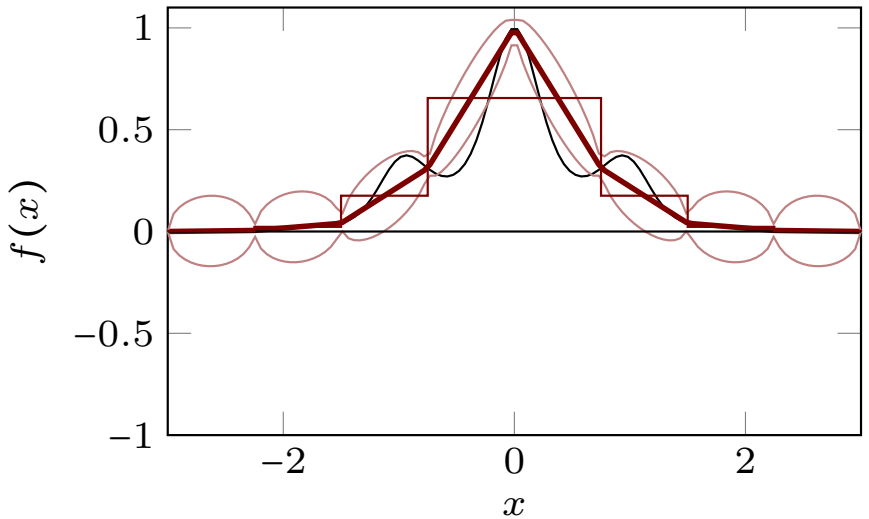
$$m_{\boldsymbol{y}}(x) = \sum_i k(x, x_i)\alpha_i$$

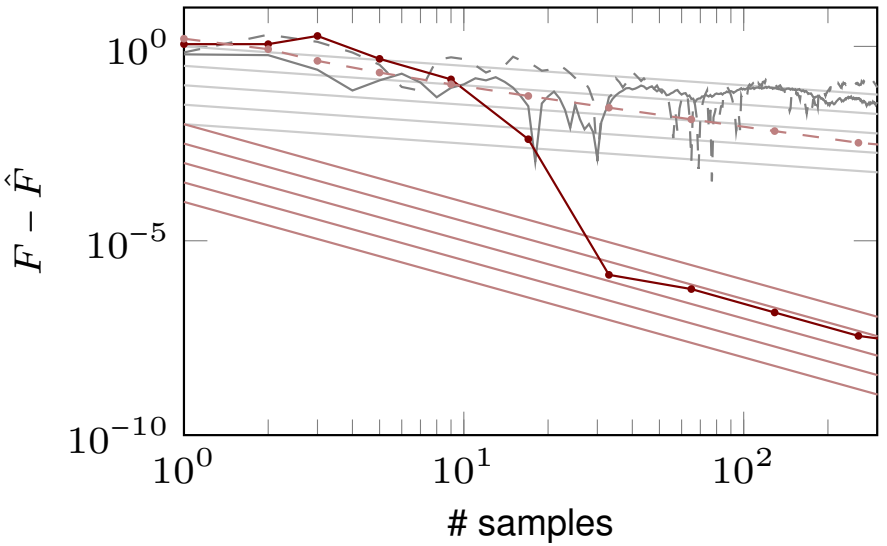
$$k(x, x_i) = \min(x, x_i) + b$$

$$\alpha_i = k_X X^{-1} \boldsymbol{y}$$

We just re-discovered the Trapezoid Rule!

A classic numerical method, derived as a learning machine [P. Diaconis, 1988, T. O'Hagan, 1991]

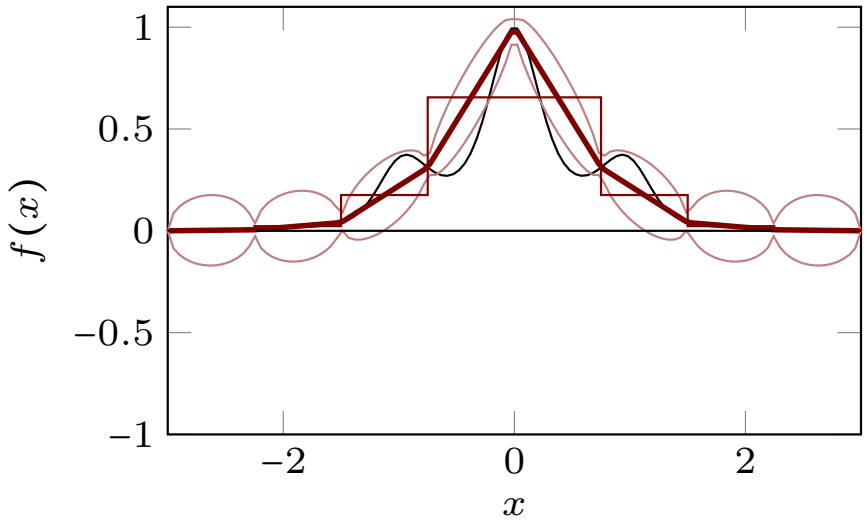


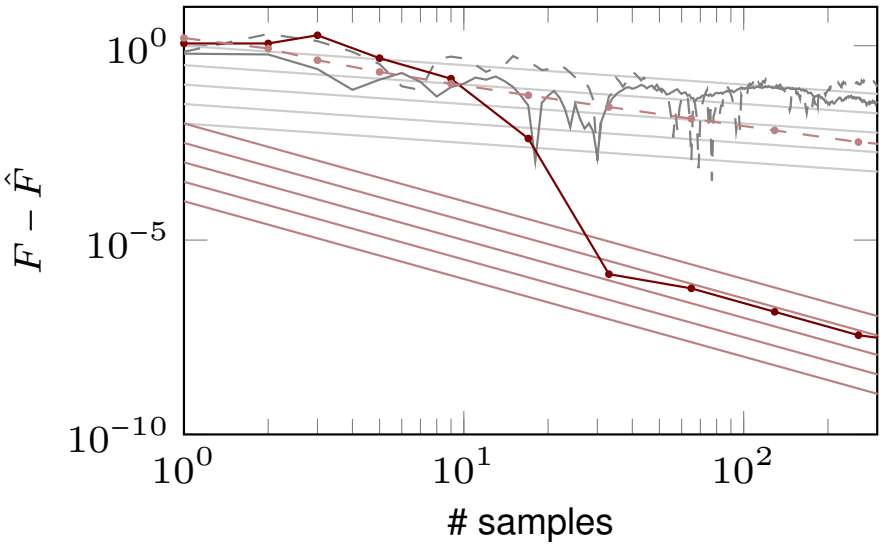


$$p(f \mid \boldsymbol{y}, \boldsymbol{x}) = \mathcal{GP}(f; m_{\boldsymbol{y}}, k_{\boldsymbol{y}}) \qquad k(x, x_i) = \min(x, x_i) + b$$

$$m_{\boldsymbol{y}}(x) = \sum_i k(x, x_i)\alpha_i \qquad \alpha_i = k_X X^{-1} \boldsymbol{y}$$

$$\mathsf{E}_{p(f \mid \boldsymbol{y})}[F] = \int m_{\boldsymbol{y}}(x)\, dx = \sum_{i=1}^{N-1} (x_{i+1} - x_i) \frac{1}{2} (f(x_{i+1}) + f(x_i))$$



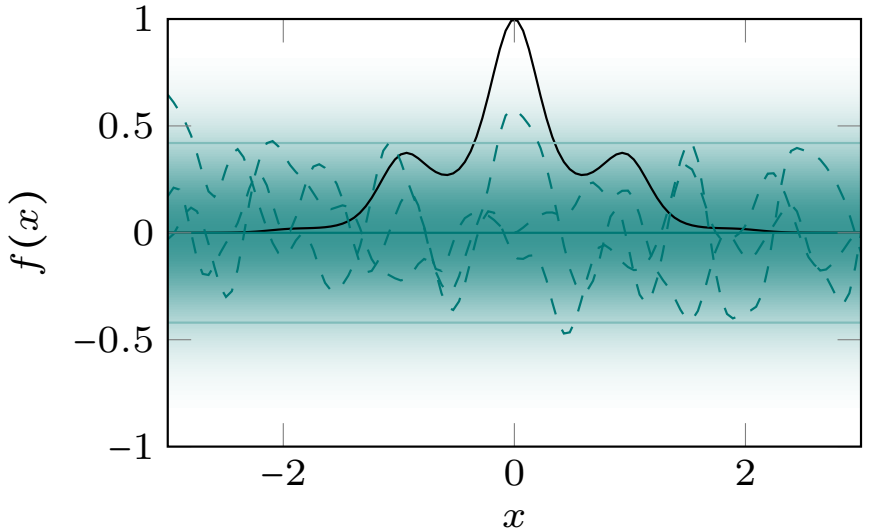


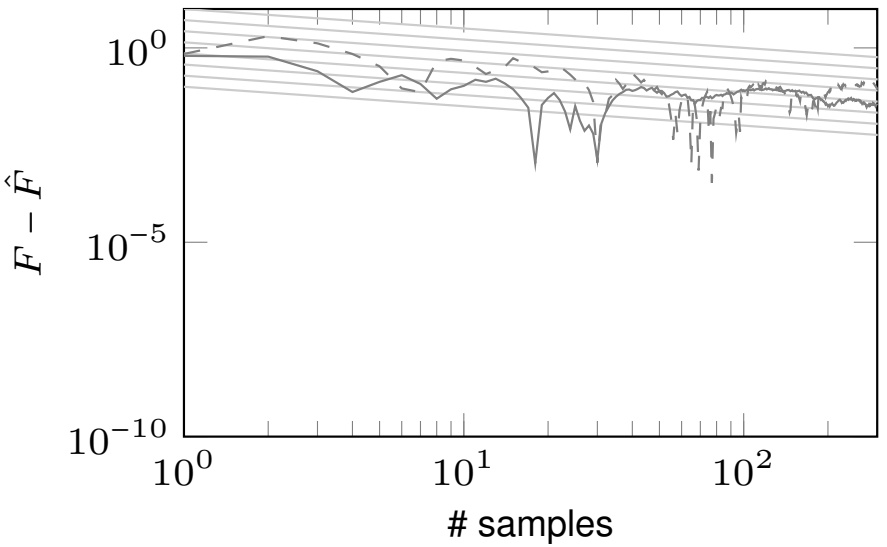
- The trapezoid rule is the maximum a-posteriori estimate for F under a Wiener process prior on f.
- Node placement by information maximization under this prior.

- introducing random numbers into a deterministic problem may not be a good strategy
- recipe for a probabilistic numerical method estimating x from $y(t)$:
  - choose model $p(x, y(t))$
  - choose action rule / policy / strategy $[t_1, \ldots, t_{i-1}, y(t_1), \ldots, y(t_{i-1})] \multimap t_i$
- some classic numerical methods can be derived entirely from an inference perspective, using classic statistical methods
  - the trapezoid rule is a MAP estimate under a Wiener process prior on the integrand; regular node placement arises from information-greediness
- the probabilistic interpretation as such does not ensure the posterior distribution is well calibrated

Customized Numerics

machine learning providing new numerics [Hennig, Osborne, Girolami, RSPA 2015]





$$k(x, x') = \exp\left(-\frac{(x - x')^2}{2\lambda^2}\right)$$

Encodes

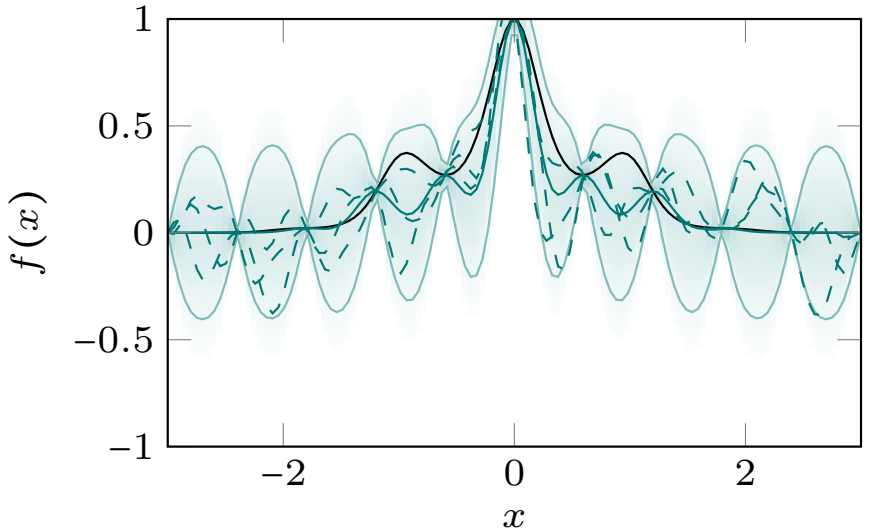
- smooth (infinitely differentiable) f
- exponentially decaying Fourier power spectrum

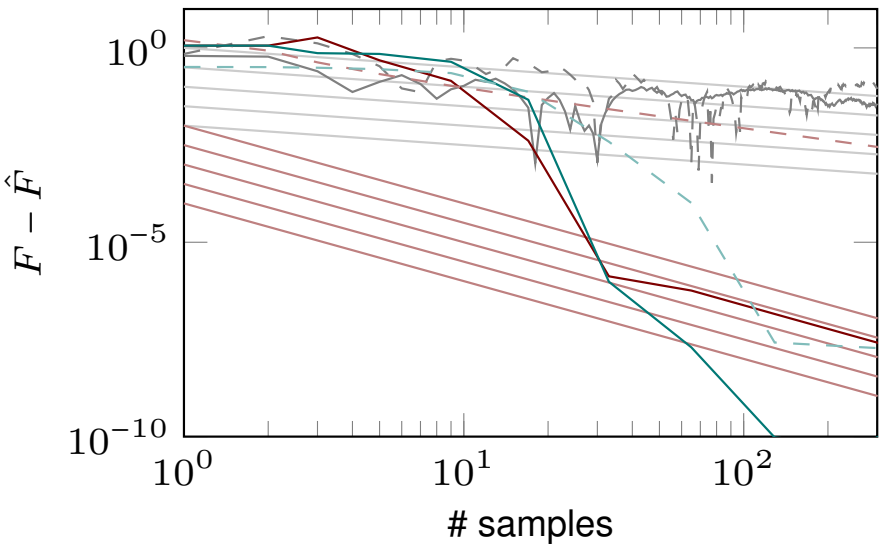
But ignores

- non-stationarity
- positivity

Customized Numerics

machine learning providing new numerics [Hennig, Osborne, Girolami, RSPA 2015]





$$k(x, x') = \exp\left(-\frac{(x - x')^2}{2\lambda^2}\right)$$

Encodes

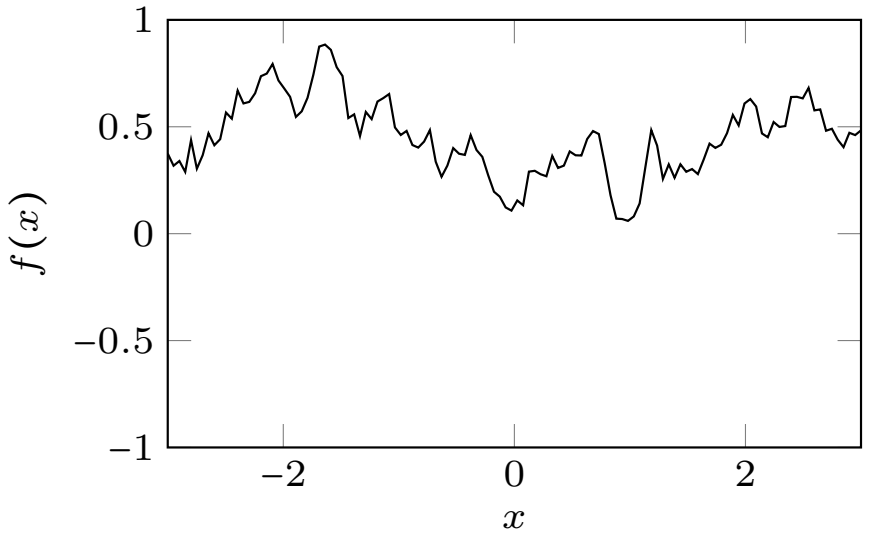
- smooth (infinitely differentiable) f
- exponentially decaying Fourier power spectrum

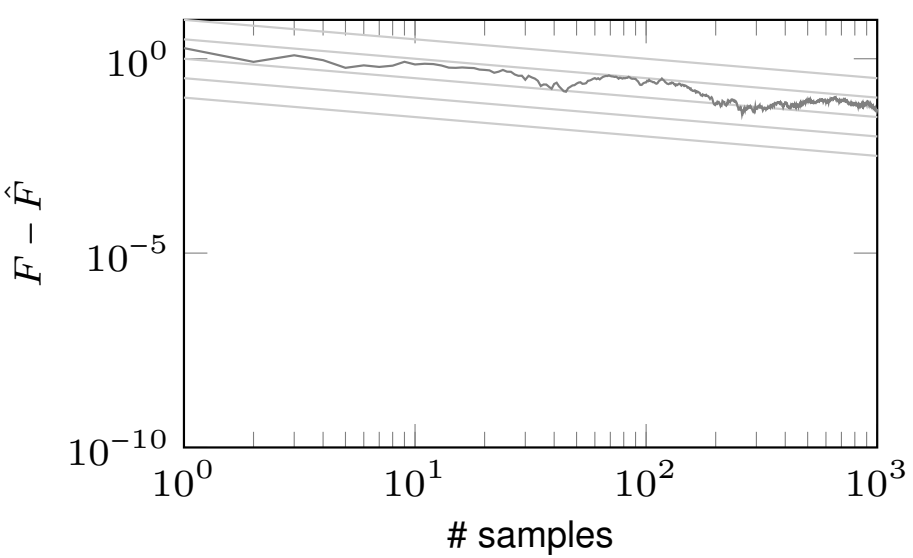
But ignores

- non-stationarity
- positivity

No Such Thing as a Free Lunch!

incorrect assumptions give arbitrarily bad performance [Hennig, Osborne, Girolami, RSPA 2015]

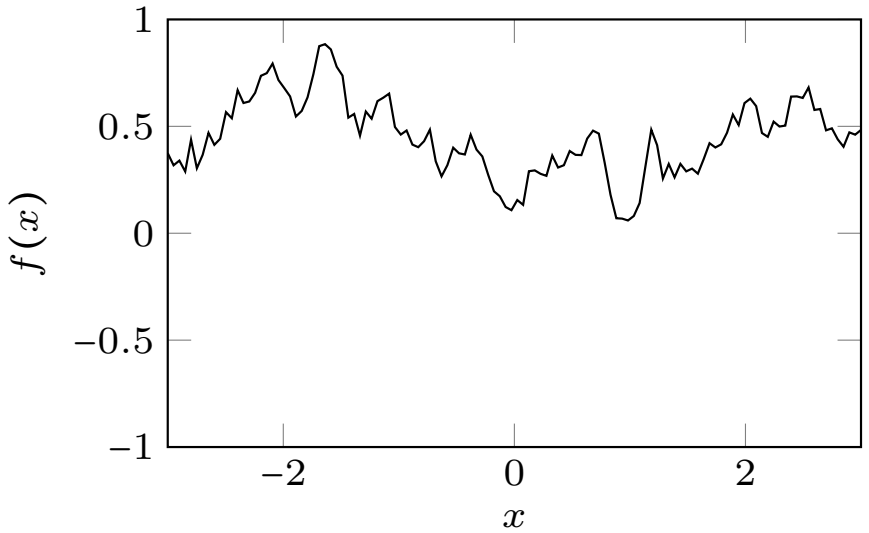


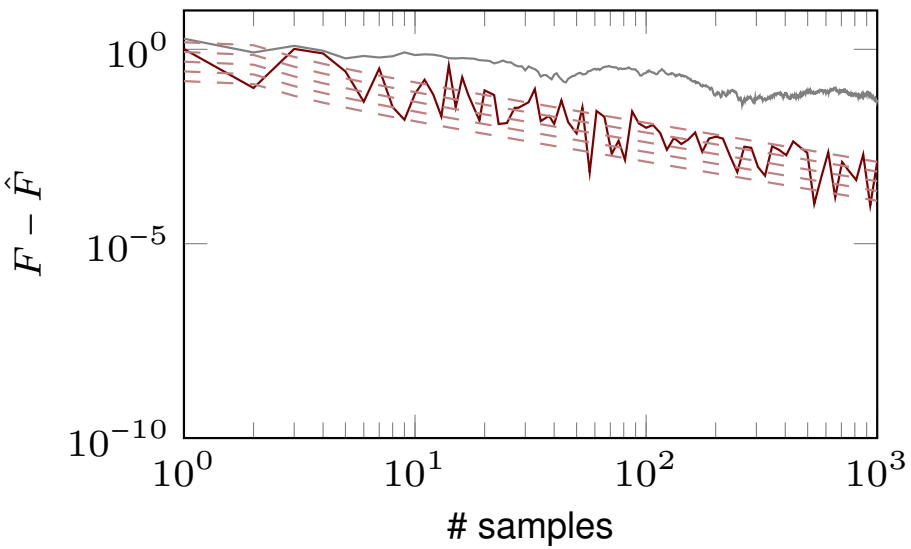


Computations collect information about a latent quantity.
The more valid prior information is available, the cheaper the computation—if the algorithm uses the prior information!

No Such Thing as a Free Lunch!

incorrect assumptions give arbitrarily bad performance [Hennig, Osborne, Girolami, RSPA 2015]

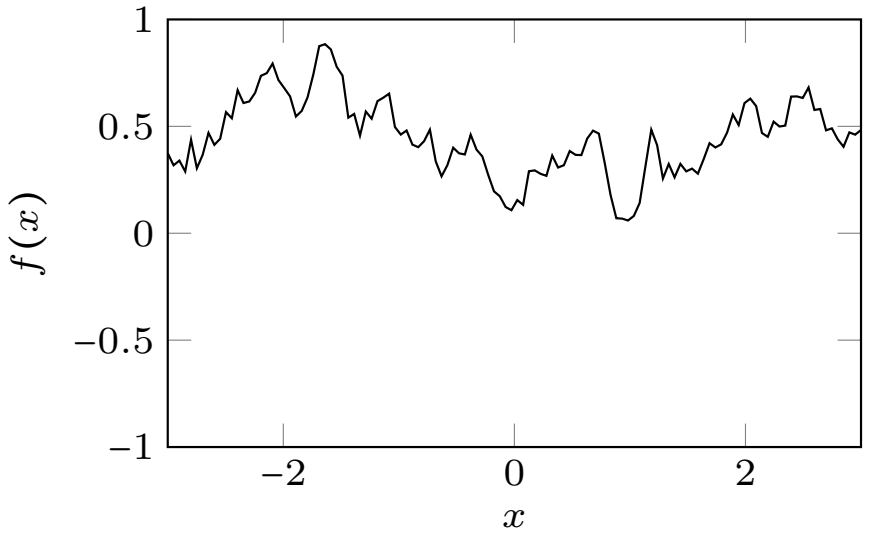


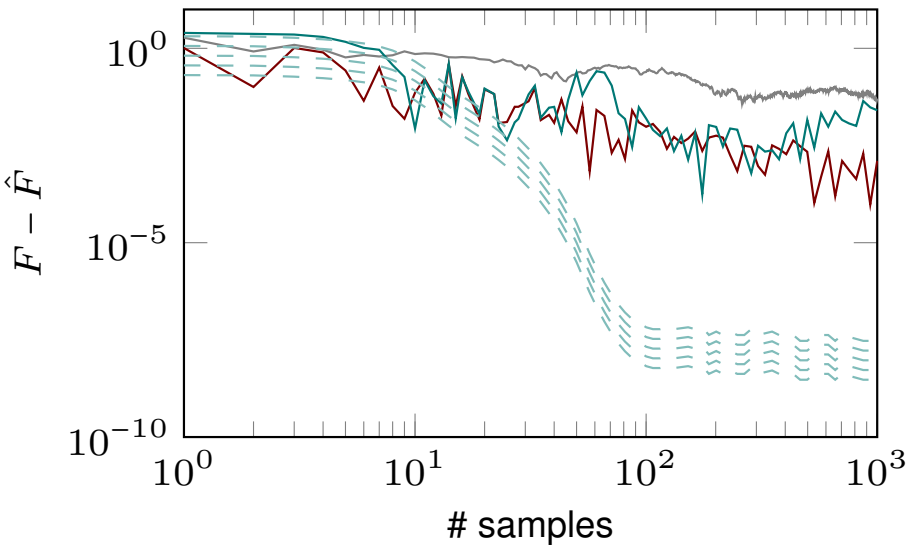


Computations collect information about a latent quantity.
The more valid prior information is available, the cheaper the computation—if the algorithm uses the prior information!

No Such Thing as a Free Lunch!

incorrect assumptions give arbitrarily bad performance [Hennig, Osborne, Girolami, RSPA 2015]

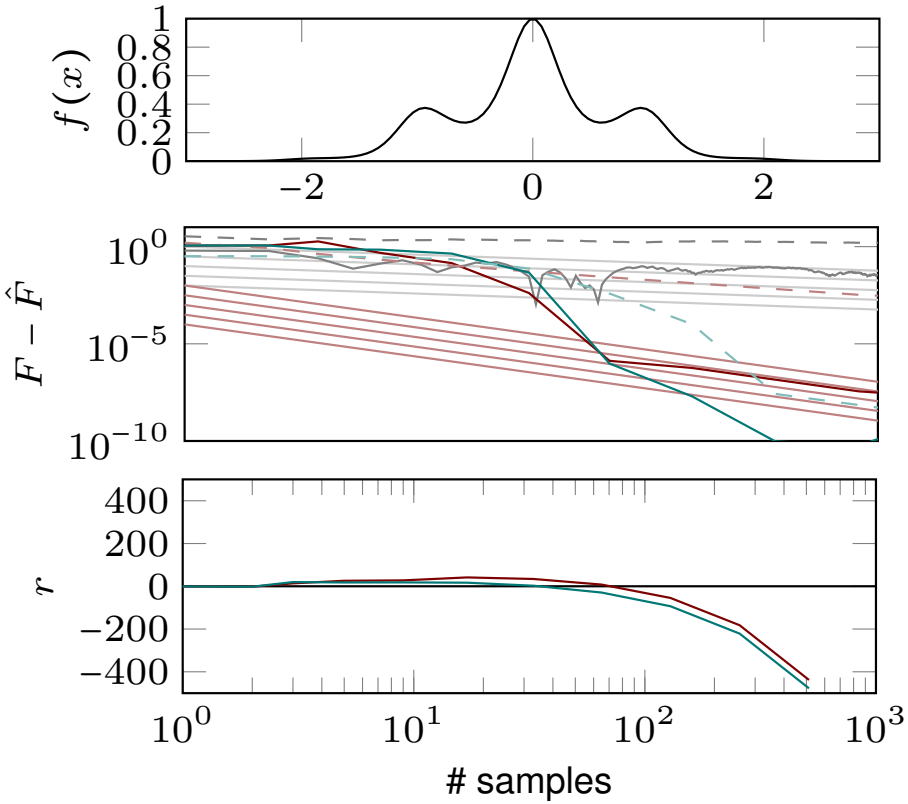


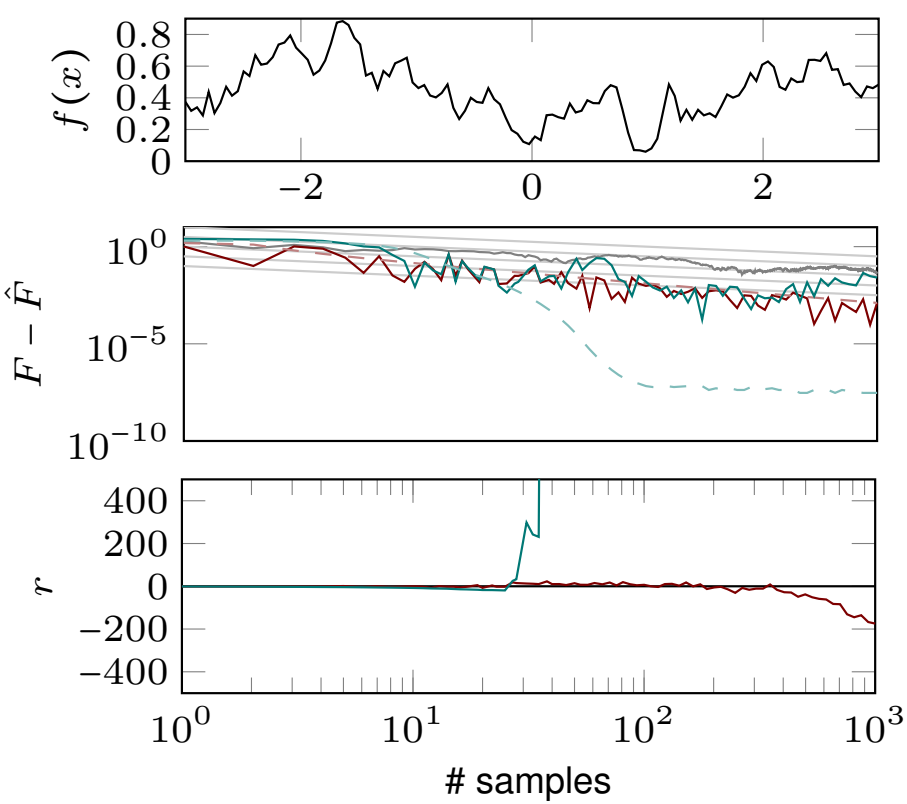


Computations collect information about a latent quantity.
The more valid prior information is available, the cheaper the computation—if the algorithm uses the prior information!

Model Mismatch can be Detected at Runtime

"a numerical conscience"





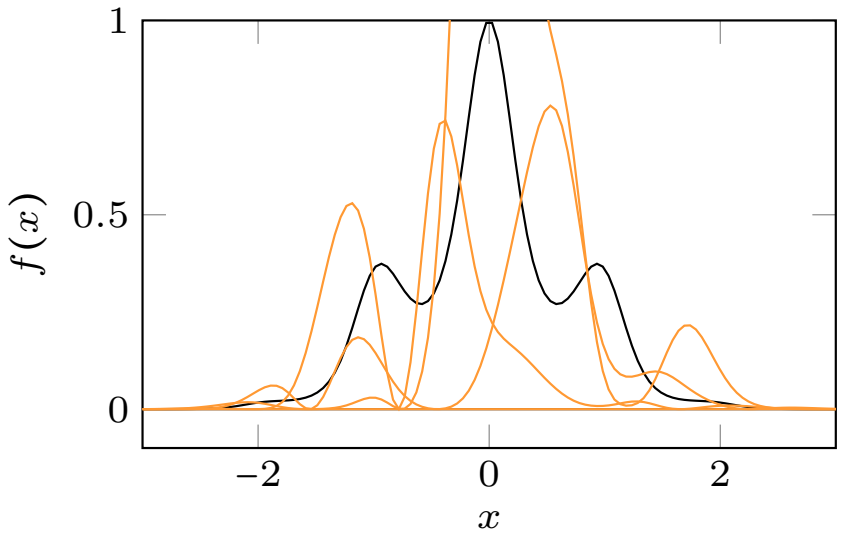
$$r = \log \frac{\mathsf{E}_{\tilde{f}}[p(\tilde{f}(\boldsymbol{x}))]}{p(f(\boldsymbol{x}))} = (f(\boldsymbol{x}) - \mu(\boldsymbol{x}))^{\top} K^{-1} (f(\boldsymbol{x}) - \mu(\boldsymbol{x})) - N$$

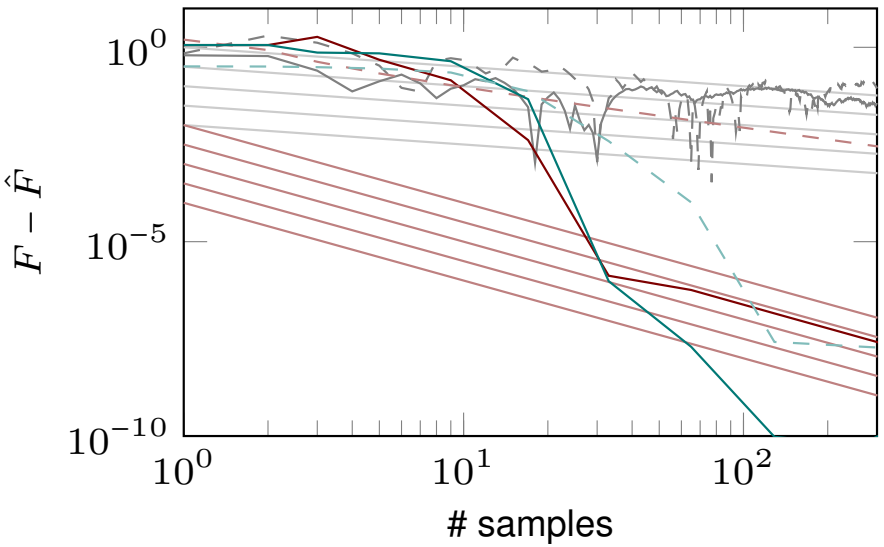
- including tangible prior information in the prior can give tailored numerics that drastically improve computational efficiency
- in contrast to physical data sources, in numerical problems, prior assumptions can be rigorously verified, because the problem is stated in a formal (programming) language!
- incorrect prior assumptions can catastrophically affect performance
- model assumptions can be adapted at runtime, using established statistical techniques, e.g. “type-II maximum likelihood”

So why is everyone (in ML) using MCMC?

Warped Sequential Active Bayesian Integration (WSABI)

[Gunther, Osborne, Garnett, Hennig, Roberts, NIPS 2014]



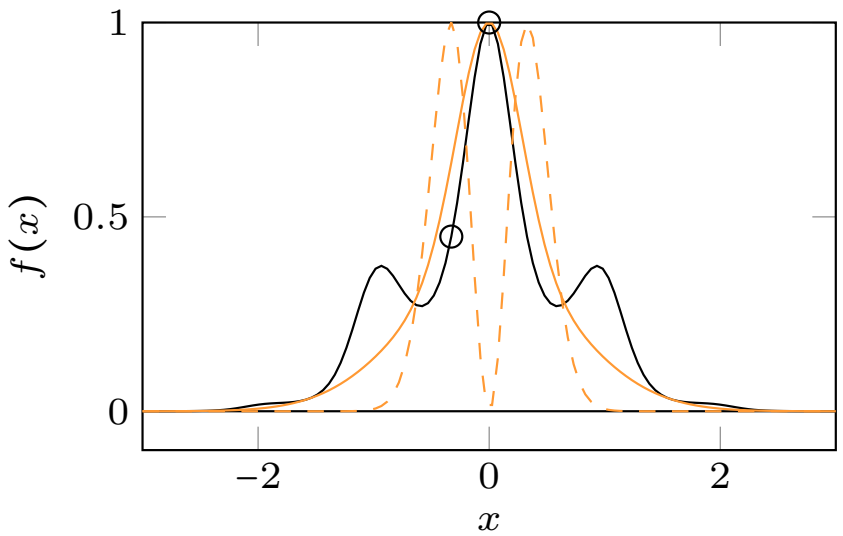


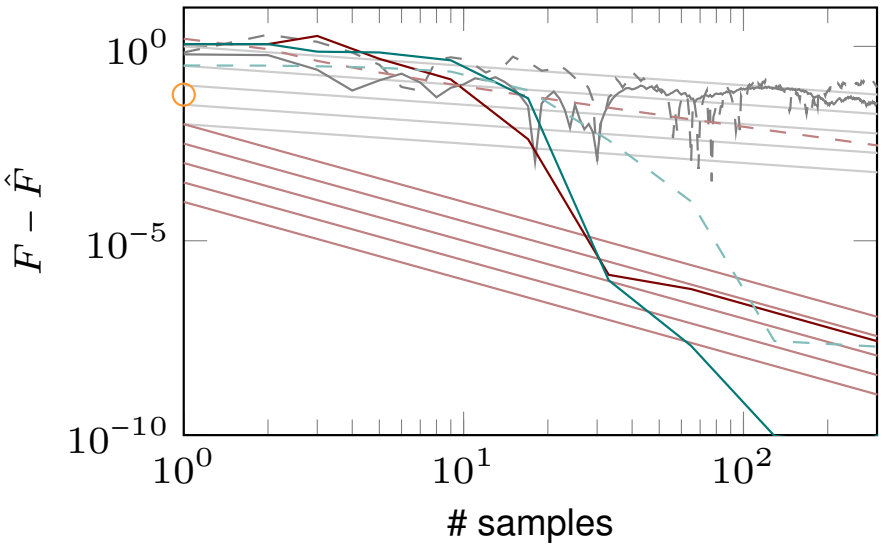
$$\sqrt{f(x)} \cdot exp(-x^2/2) \sim \mathcal{GP}\left[0, k = \exp(-1/2(x - x')^\top \Lambda^{-1}(x - x'))\right]$$

- encodes positivity
- encodes nonstationarity

Warped Sequential Active Bayesian Integration (WSABI)

[Gunther, Osborne, Garnett, Hennig, Roberts, NIPS 2014]



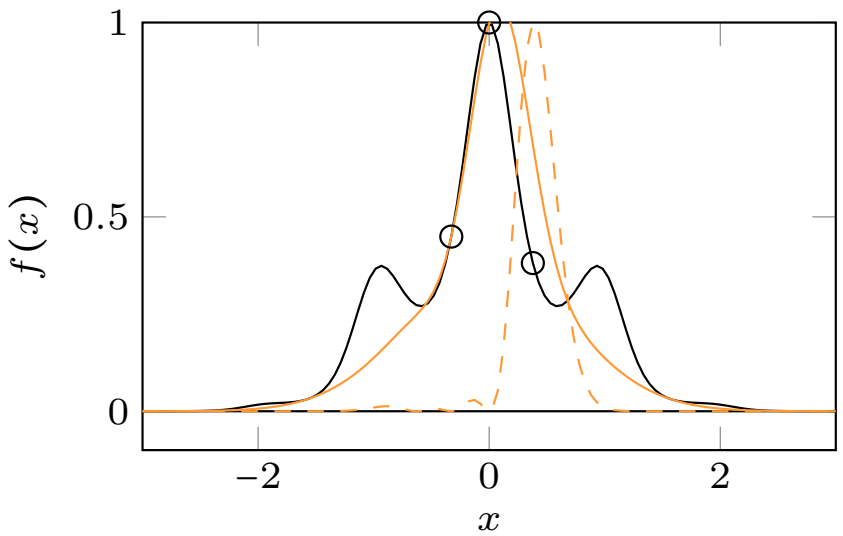


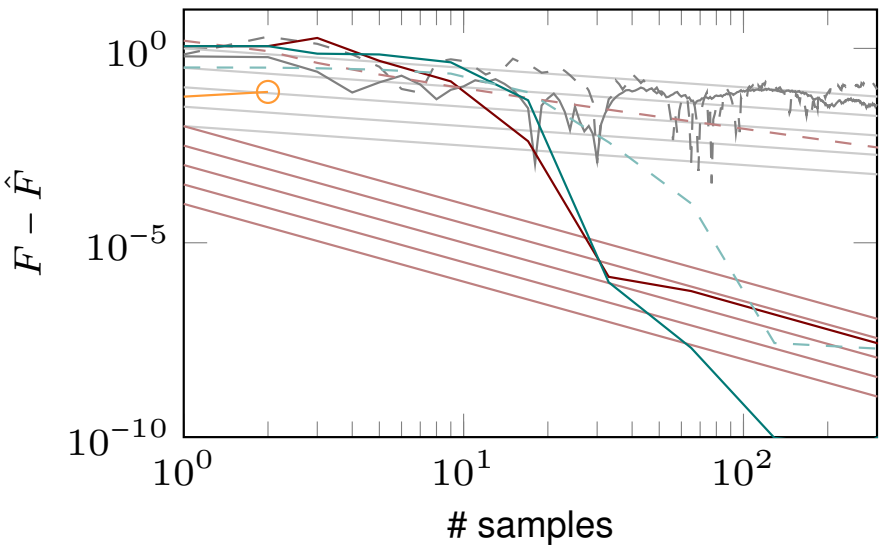
- select evaluation nodes at $\arg\max \mathrm{var}[f(x) \cdot \exp(-x^2)]$
- this scales to higher input dimensionality

more formal analysis in arXiv `1410.2392` [Oates et al.] & `1506.02681` [Briol et al.]

Warped Sequential Active Bayesian Integration (WSABI)

[Gunther, Osborne, Garnett, Hennig, Roberts, NIPS 2014]



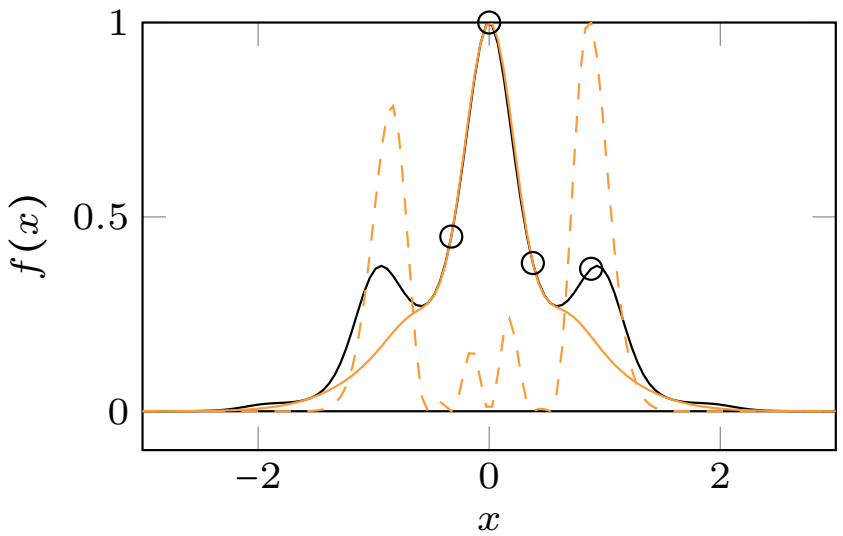


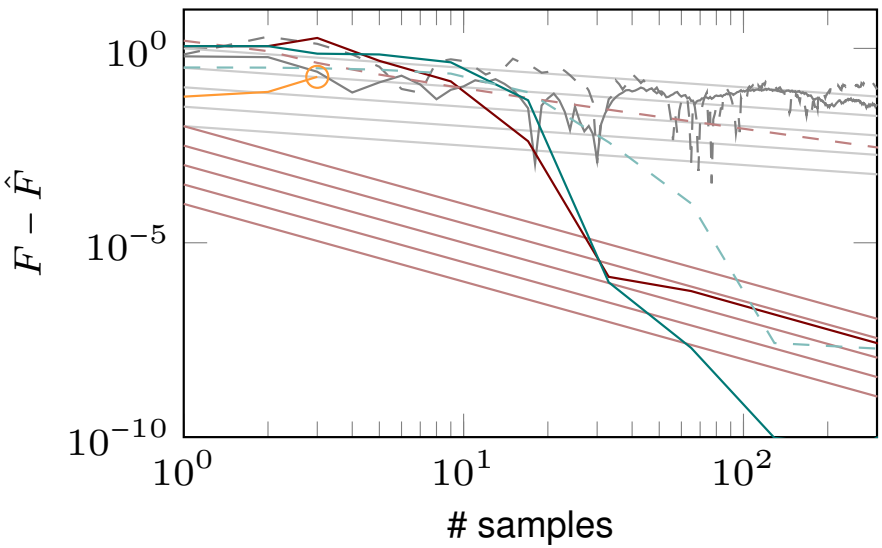
- select evaluation nodes at $\arg\max \mathrm{var}[f(x) \cdot \exp(-x^2)]$
- this scales to higher input dimensionality

more formal analysis in arXiv `1410.2392` [Oates et al.] & `1506.02681` [Briol et al.]

Warped Sequential Active Bayesian Integration (WSABI)

[Gunther, Osborne, Garnett, Hennig, Roberts, NIPS 2014]



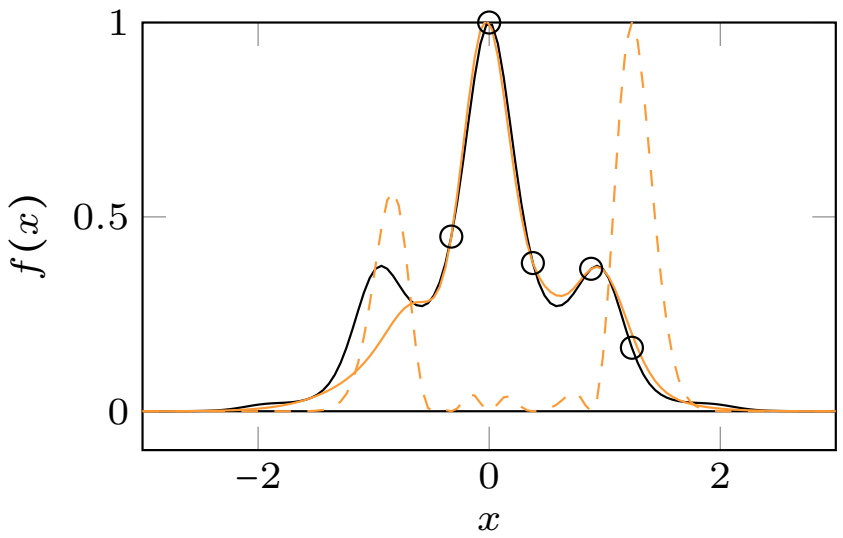


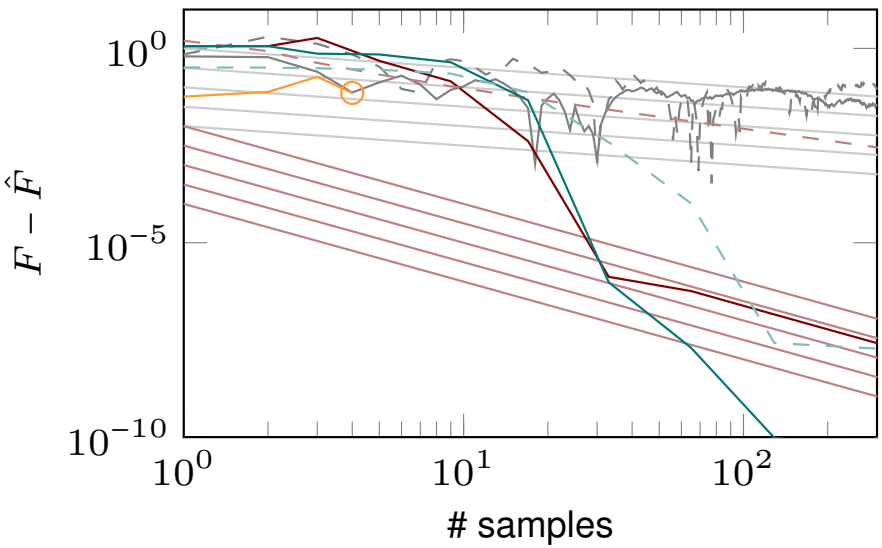
- select evaluation nodes at $\arg\max \mathrm{var}[f(x) \cdot \exp(-x^2)]$
- this scales to higher input dimensionality

more formal analysis in arXiv `1410.2392` [Oates et al.] & `1506.02681` [Briol et al.]

Warped Sequential Active Bayesian Integration (WSABI)

[Gunther, Osborne, Garnett, Hennig, Roberts, NIPS 2014]



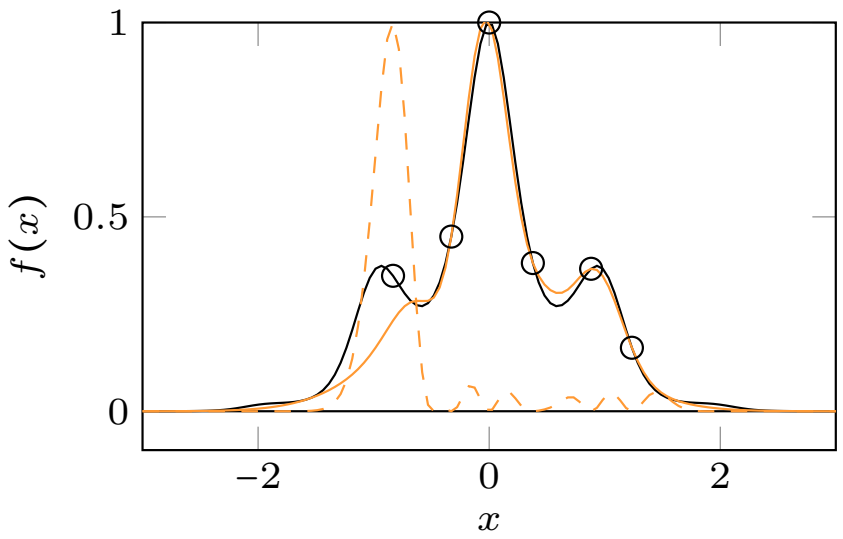


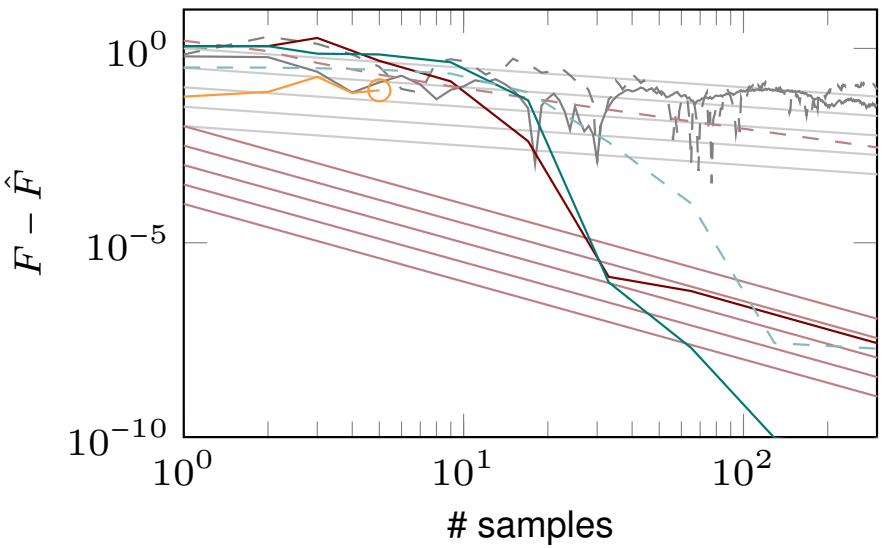
- select evaluation nodes at $\arg\max \mathrm{var}[f(x) \cdot \exp(-x^2)]$
- this scales to higher input dimensionality

more formal analysis in arXiv `1410.2392` [Oates et al.] & `1506.02681` [Briol et al.]

Warped Sequential Active Bayesian Integration (WSABI)

[Gunther, Osborne, Garnett, Hennig, Roberts, NIPS 2014]



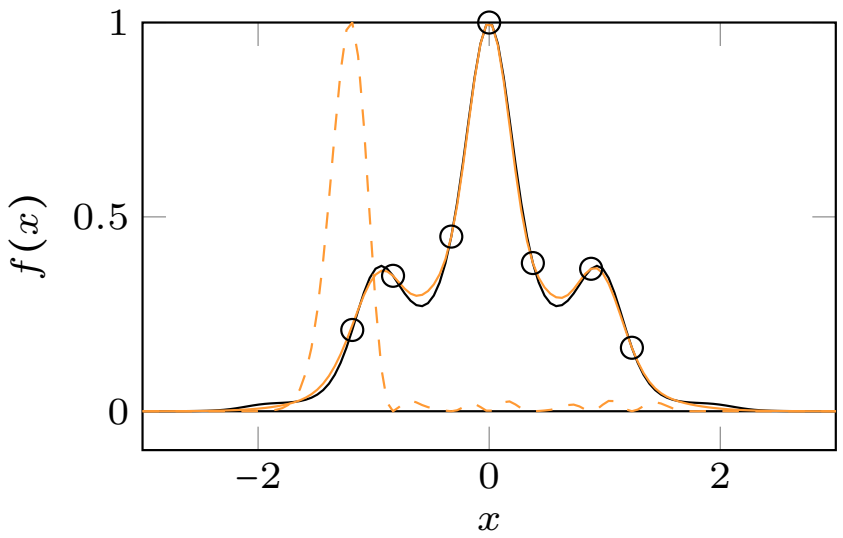


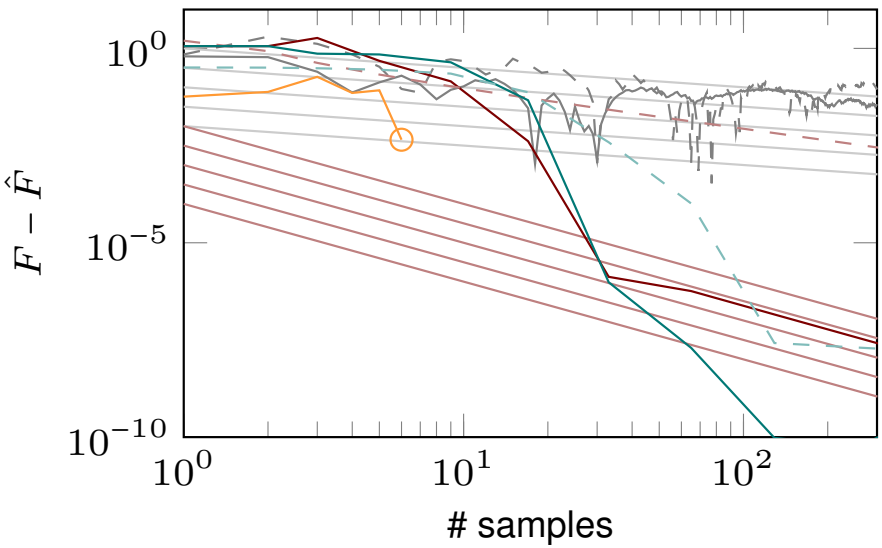
- select evaluation nodes at $\arg\max \mathrm{var}[f(x) \cdot \exp(-x^2)]$
- this scales to higher input dimensionality

more formal analysis in arXiv `1410.2392` [Oates et al.] & `1506.02681` [Briol et al.]

Warped Sequential Active Bayesian Integration (WSABI)

[Gunther, Osborne, Garnett, Hennig, Roberts, NIPS 2014]



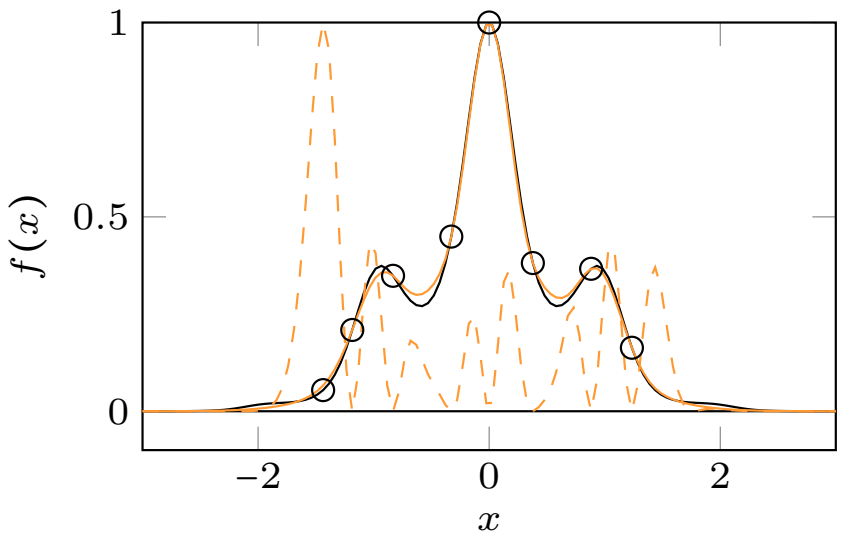


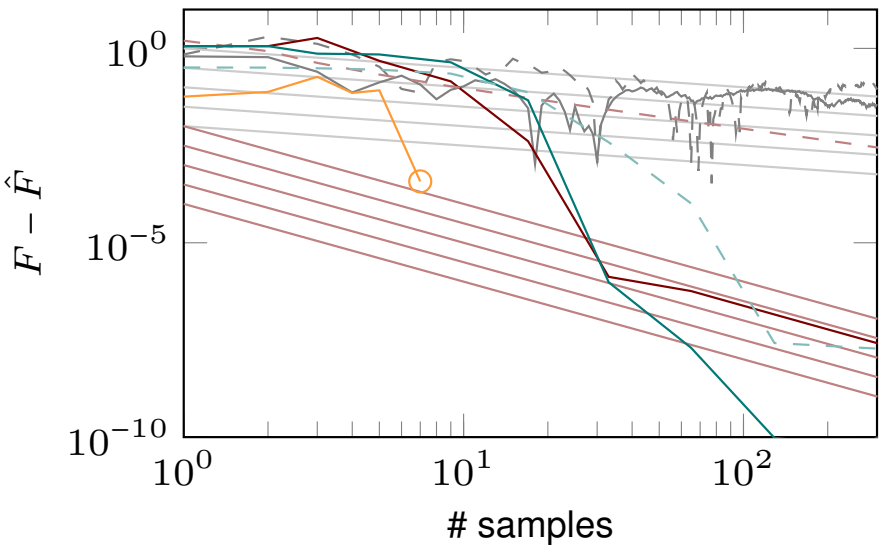
- select evaluation nodes at $\arg\max \mathrm{var}[f(x) \cdot \exp(-x^2)]$
- this scales to higher input dimensionality

more formal analysis in arXiv `1410.2392` [Oates et al.] & `1506.02681` [Briol et al.]

Warped Sequential Active Bayesian Integration (WSABI)

[Gunther, Osborne, Garnett, Hennig, Roberts, NIPS 2014]



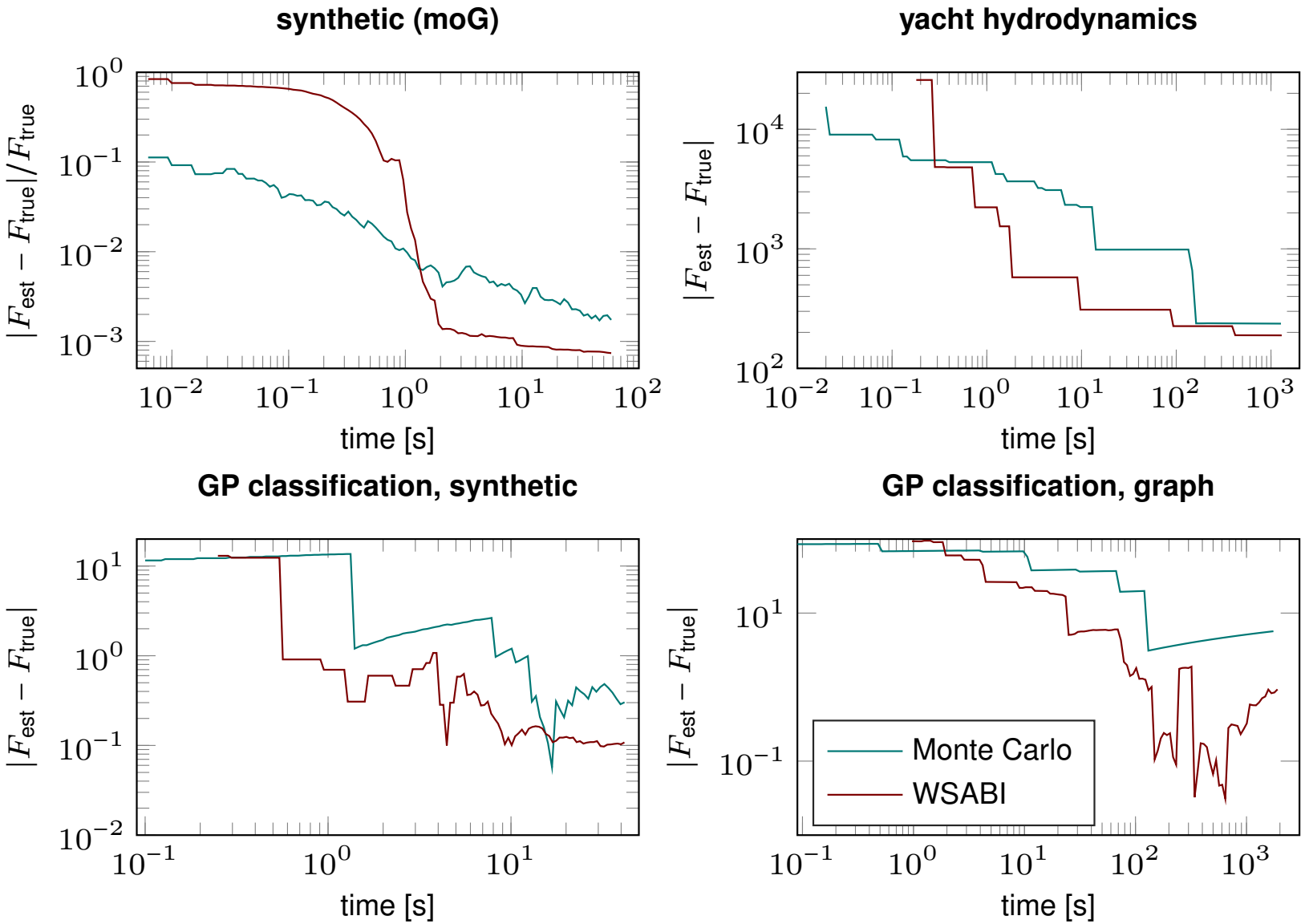


- select evaluation nodes at $\arg\max \mathrm{var}[f(x) \cdot \exp(-x^2)]$
- this scales to higher input dimensionality

more formal analysis in arXiv `1410.2392` [Oates et al.] & `1506.02681` [Briol et al.]

Probabilistic Numerics Need Not Be Expensive

WSABI is time-competititive with MCMC [Gunther, Osborne, Garnett, Hennig, Roberts, NIPS 2014]

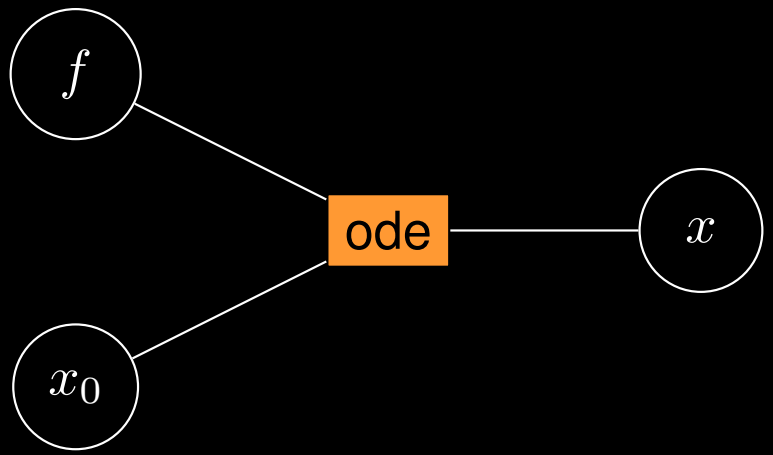


end of Integration part

- computation is inference
- there is a deep formal connection between basic numerical methods and basic statistical models
- ignoring salient prior information causes drastic computational cost increase. Black boxes may be convenient, but they are not efficient
- machine learning can help numerics, and vice versa

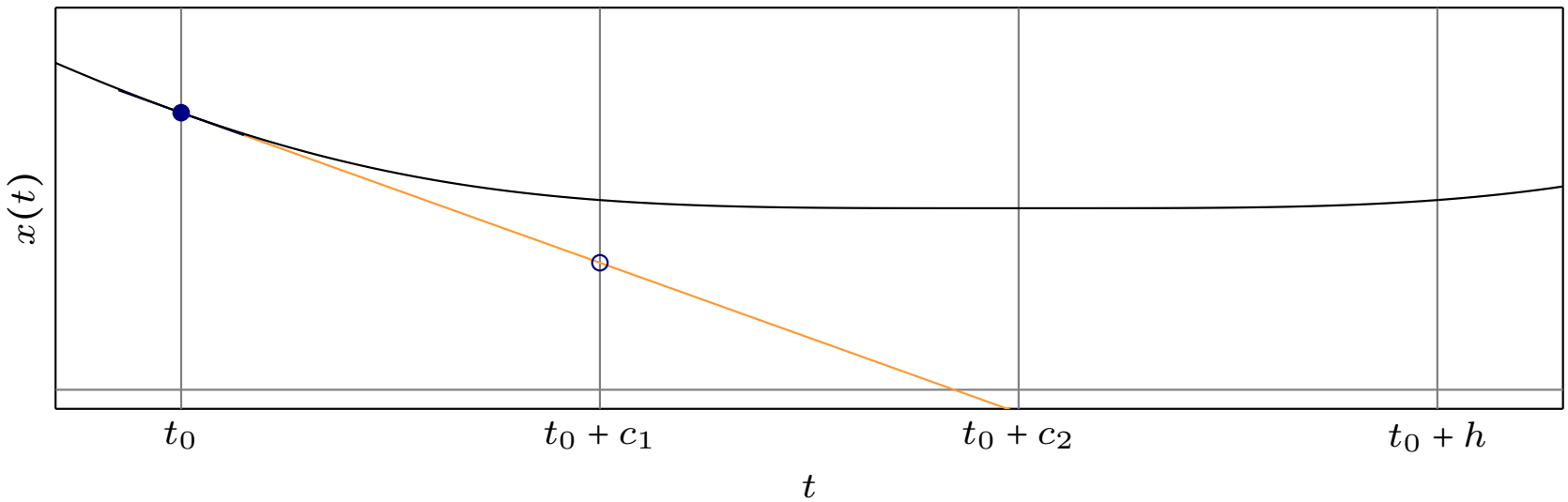
Ordinary Differential Equations

$$x'(t) = f(x(t), t) \qquad x(t_0) = x_0 \qquad x : \mathbb{R} \to \mathbb{R}^N$$



Runge-Kutta Methods

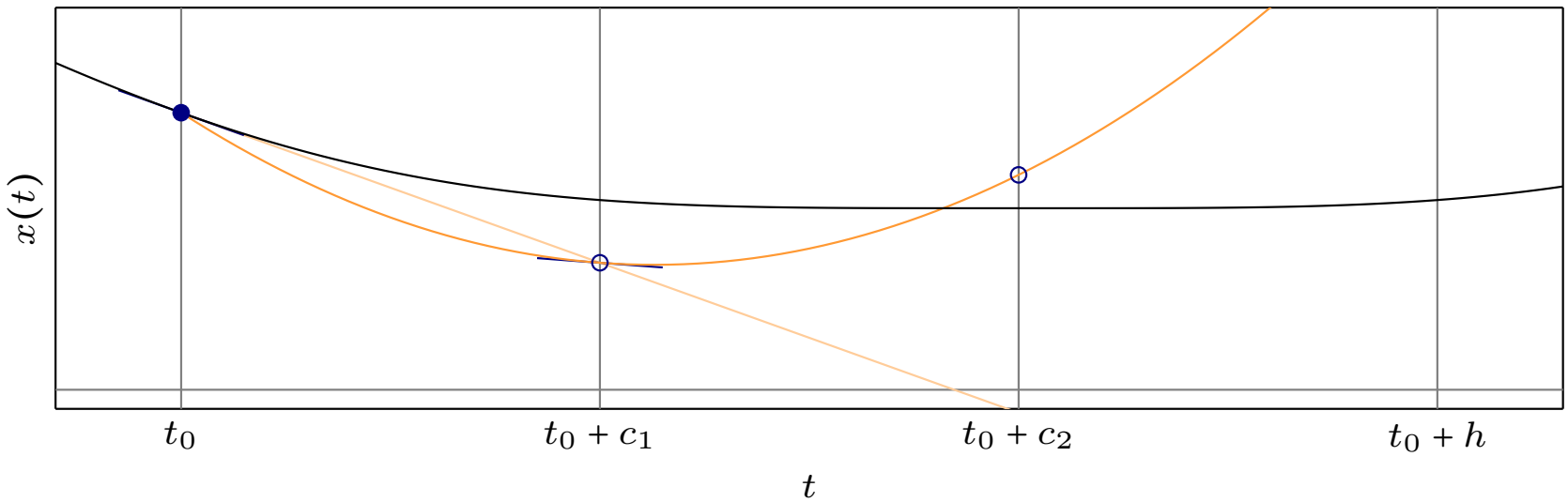
iterative linear extrapolation



0	1		$y_1 = f(1x_0, t_0)$
c_1	1	w_{11}	$y_2 = f(1x_0 + w_{11}y_1, t_0 + c_1)$
c_2			
h			

Runge-Kutta Methods

iterative linear extrapolation



0	1			
c_1	1	w_{11}		
c_2	1	w_{21}	w_{22}	
h				

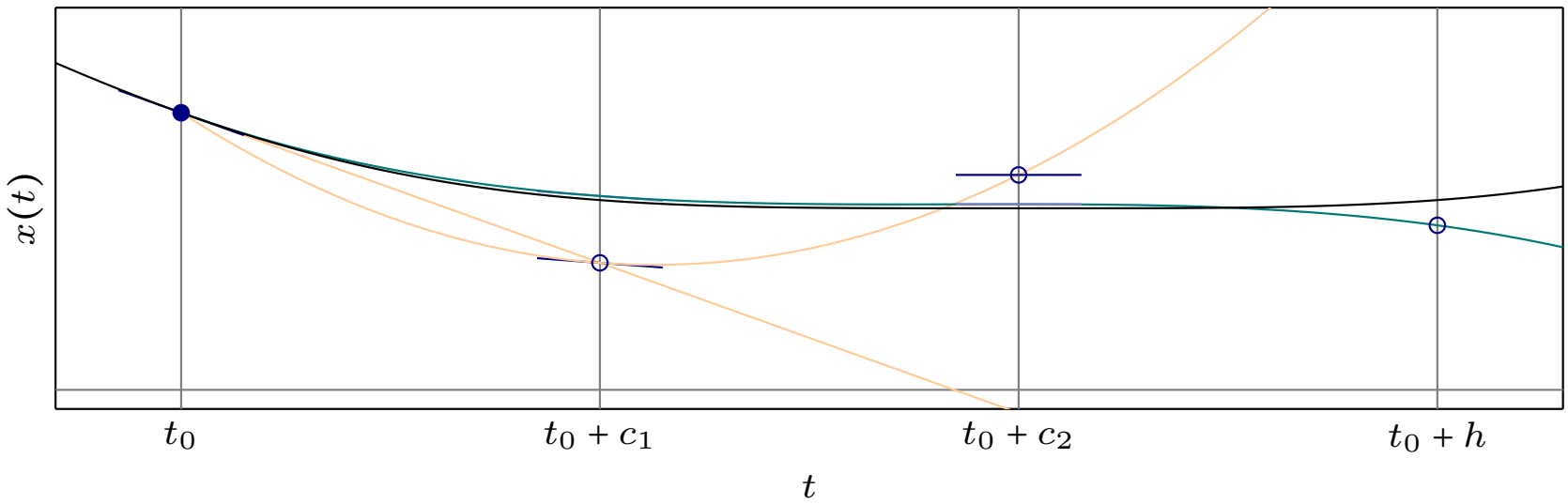
$$y_1 = f(1x_0, t_0)$$

$$y_2 = f\left(1x_0 + w_{11}y_1, t_0 + c_1\right)$$

$$y_{s+1} = f\left(1x_0 + \sum_i^s w_{si}y_i, t_0 + c_s\right)$$

Runge-Kutta Methods

iterative linear extrapolation



0	1			
c_1	1	w_{11}		
c_2	1	w_{21}	w_{22}	
h	1	b_1	b_2	b_3

$$y_1 = f(1x_0, t_0)$$

$$y_2 = f\left(1x_0 + w_{11}y_1, t_0 + c_1\right)$$

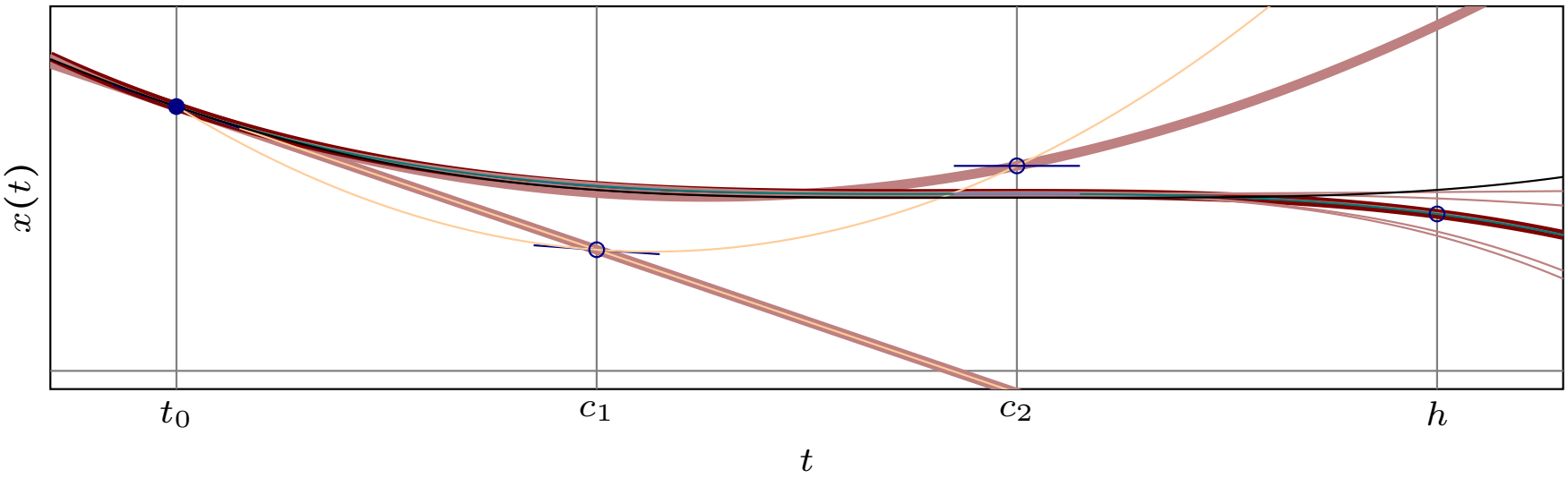
$$y_{s+1} = f\left(1x_0 + \textstyle\sum_i^s w_{si}y_i, t_0 + c_s\right)$$

$$\hat{x}(t_0 + h) = 1x_0 + \textstyle\sum_i b_i y_i$$

- RK choose (c, w, b) such that $\|\hat{x}(t_0 + h) - x(t_0 + h)\| = \mathcal{O}(h^p)$

Gauss-Markov inference on ODEs

a probabilistic model matching Runge-Kutta [Schober, Duvenaud, Hennig, NIPS 2014]



0	1			
c_1	1	w_{11}		
c_2	1	w_{21}	w_{22}	
h	1	b_1	b_2	b_3

$$y_1 = f\left(\mu_{|x_0}(t_0), t_0\right)$$

$$y_2 = f\left(\mu_{|x_0,y_1}(c_1), t_0 + c_1\right)$$

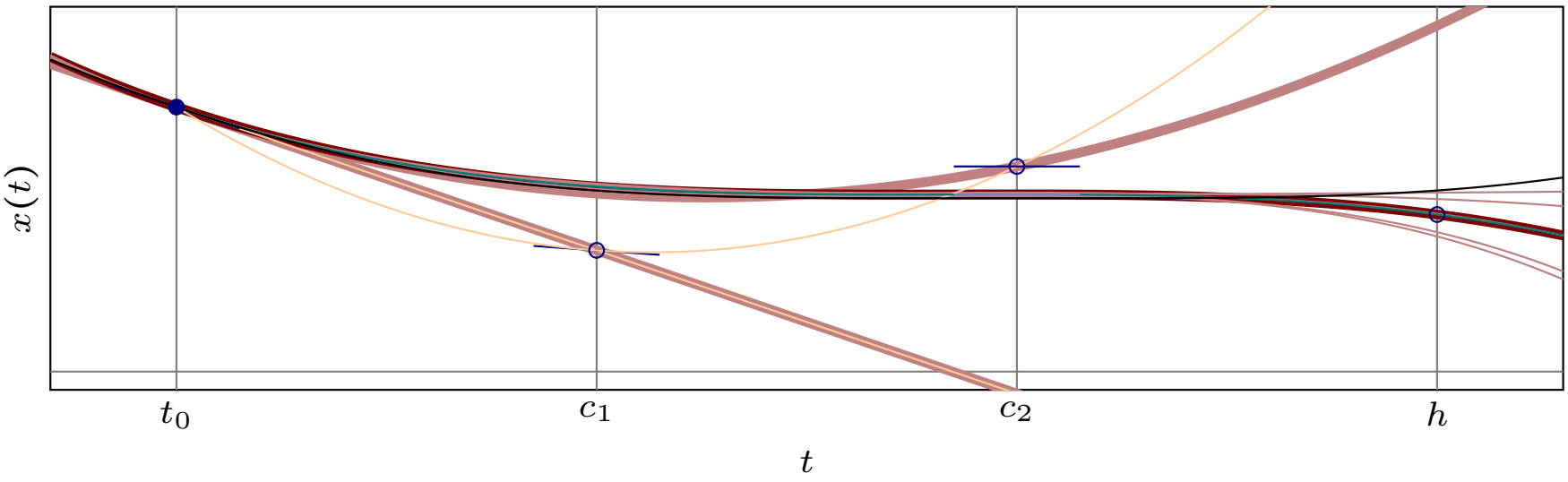
$$y_{s+1} = f\left(\mu_{|x_0,y_1,\cdots,y_s}(c_s), t_0 + c_s\right)$$

$$\hat{x}(t_0 + h) = \mu_{|x_0,y_i}(t_0 + h)$$

$$\mu_{|\boldsymbol{y}}(t) := \mathrm{cov}(x(t), y_i)\,\mathrm{cov}(\boldsymbol{y}, \boldsymbol{y})^{-1}\boldsymbol{y} = 1x_0 + \textstyle\sum_i w_{si} y_i$$

Gauss-Markov inference on ODEs

a probabilistic model matching Runge-Kutta [Schober, Duvenaud, Hennig, NIPS 2014]



- Linear extrapolation suggests Gaussian process model
- polynomial form suggests Wiener state-space model

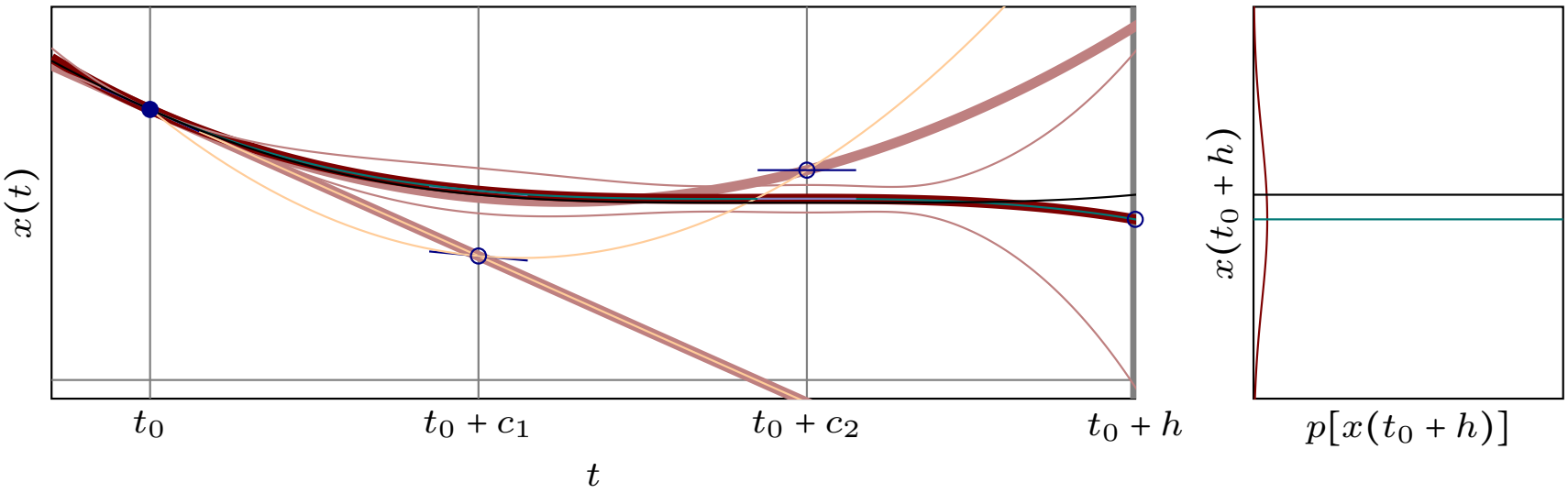
$$dz = F\,dt + L\,d\omega$$

$$d\begin{bmatrix} x(t) \\ x'(t) \\ \vdots \\ h^k/k!\,x^{(k)}(t) \end{bmatrix} = \frac{1}{h}\begin{bmatrix} 0 & 1 & \dots & \\ & 0 & 2 & \ddots \\ \vdots & & \ddots & k \\ & & & 0 \end{bmatrix} dt + \sigma \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} d\omega$$

- inference through filtering

Calibrating Uncertainty

within the parametrized class

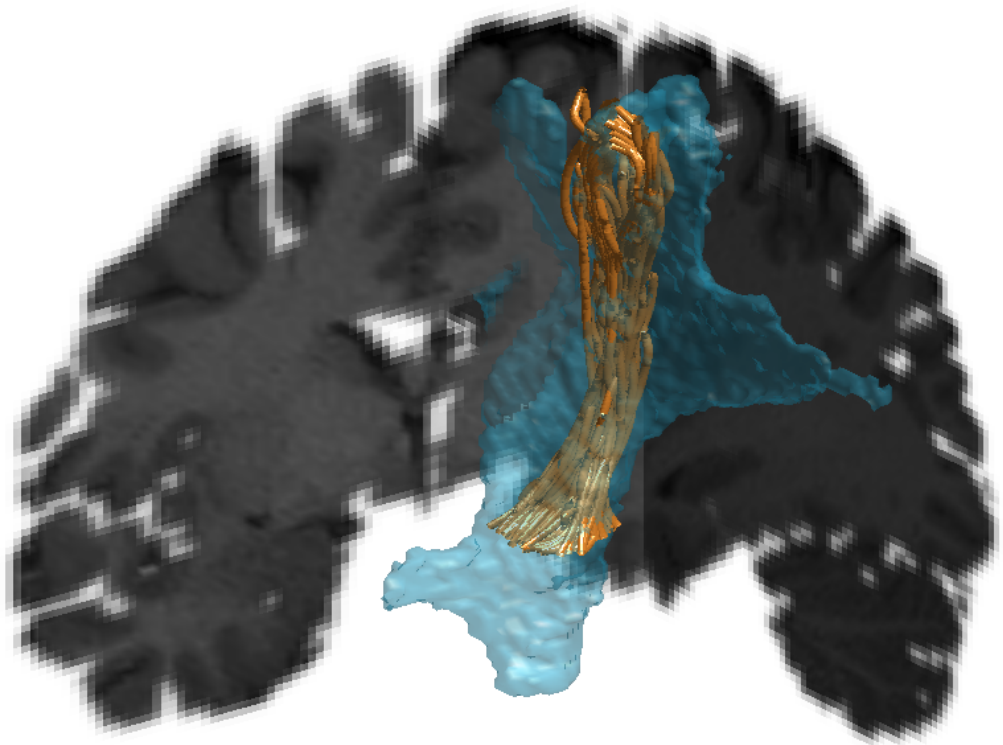


- posterior mean $\mu_{|y} = kK^{-1}y$ invariant under $k \to \theta^2 k$
- posterior covariance $k_{|y} = k - kK^{-1}k$ scaled by θ^2
- connection to local error estimation in existing methods

- as in integration, classic families of ODE solvers can be interpreted as MAP inference
- for each classic solver, whole family of probabilistic solvers with same point estimate, different uncertainties
- probabilistic solvers need not be expensive. Can even have exact same cost as classic methods

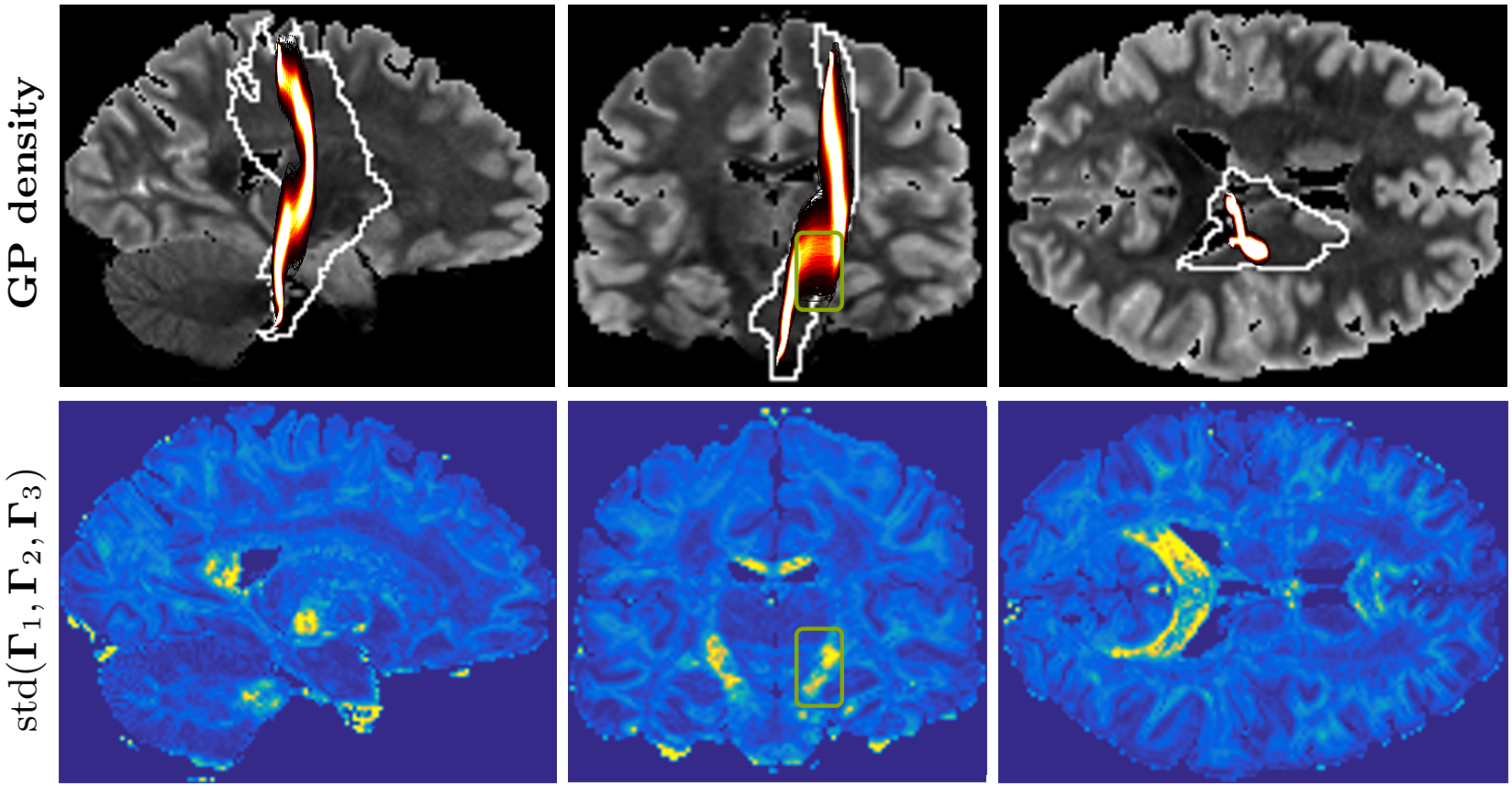
Visualizing Computational Uncertainty

Neural Pathways [M. Schober, N. Kasenburg, A. Feragen, P.H., S. Hauberg, MICCAI 2014]



Propagating Uncertainty

geodesics on an uncertain Riemannian metric [Hauberg, Schober, Liptrot, P.H., Feragen, MICCAI 2015]

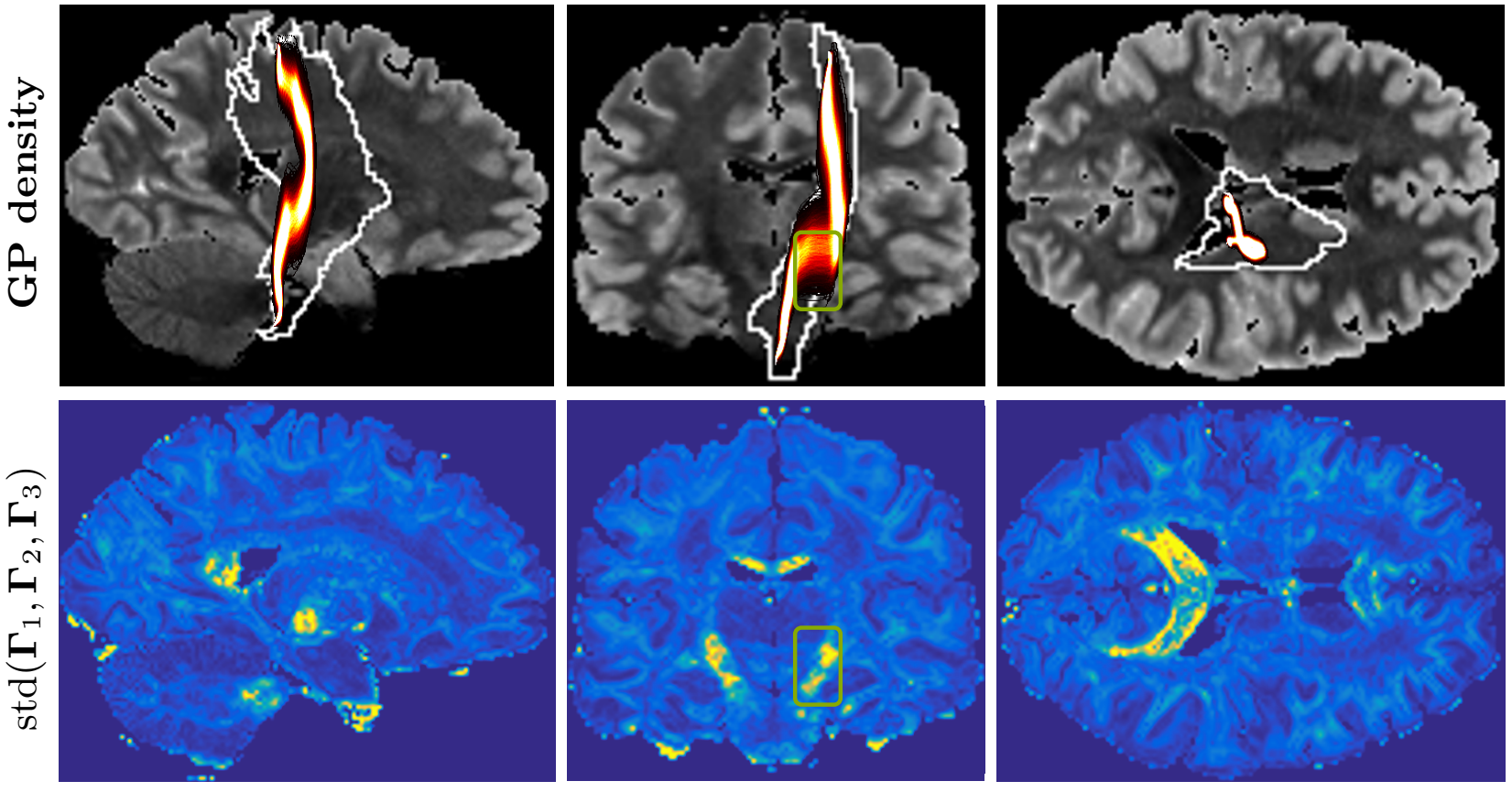


- Shortest paths (geodesics) on Riemannian manifold of metric M obey

$$x''(t) = -1/2 M^{-1}(x(t)) \left[\frac{\partial \overrightarrow{M}(x(t))}{\partial x(t)} \right]^{\top} (x'(t) \otimes x'(t)) = f(x(t), x'(t), t)$$

Propagating Uncertainty

geodesics on an uncertain Riemannian metric [Hauberg, Schober, Liptrot, P.H., Feragen, MICCAI 2015]



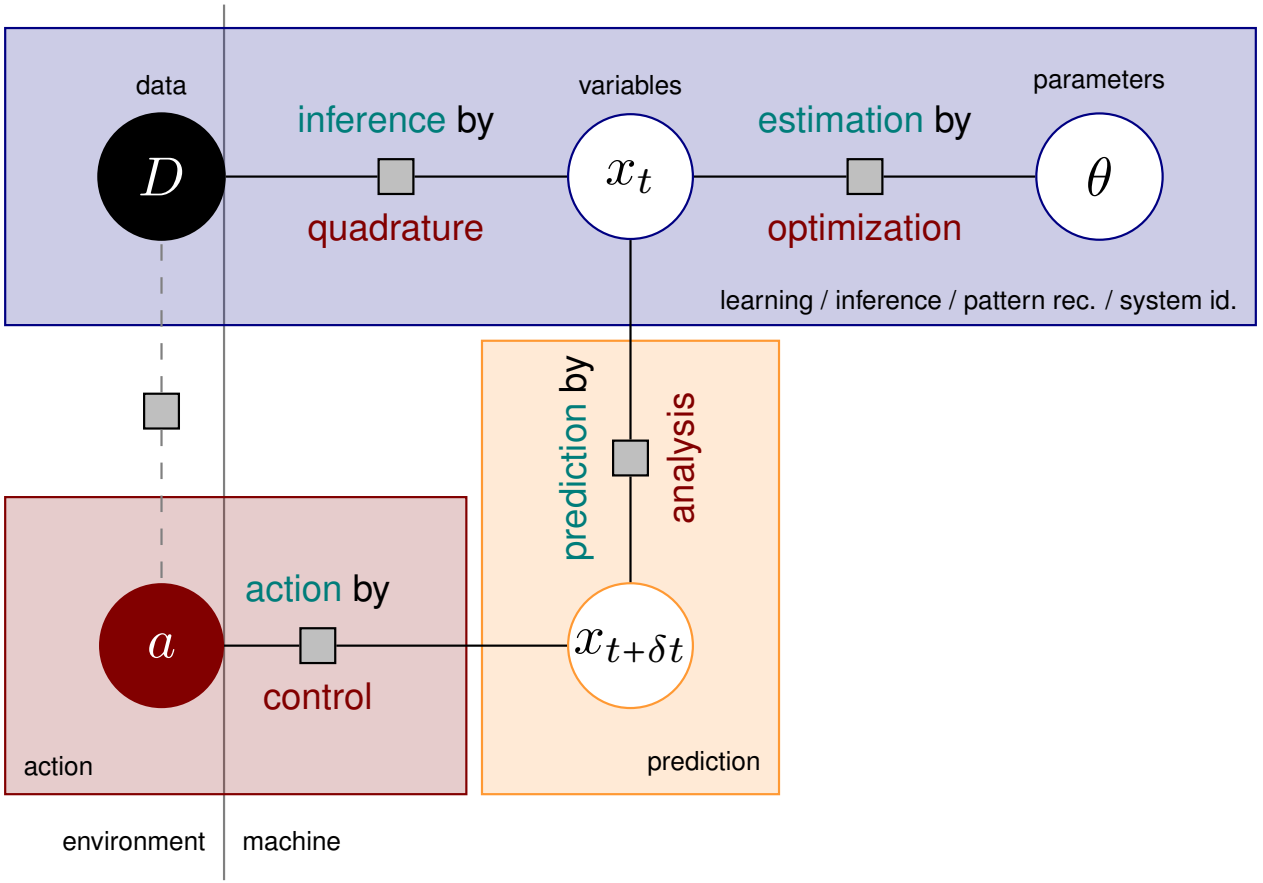
- Shortest paths (geodesics) on Riemannian manifold of metric M obey

$$x''(t) = -1/2 M^{-1}(x(t)) \left[\frac{\partial \overrightarrow{M}(x(t))}{\partial x(t)} \right]^{\top} (x'(t) \otimes x'(t)) = f(x(t), x'(t), t)$$

- what if M $\mathcal{N}(m, V)$ is uncertain (inferred from data)?

Uncertainty Across Composite Computations

interacting information requirements [Hennig, Osborne, Girolami, Proc. Royal Society A 2015]



- numerical methods able to deal with uncertain (probabilistic) inputs, and returning uncertain (probabilistic) outputs, allow control of computational effort
- (this is not the same as probabilistic programming!)

Summary

The probabilistic view of computation

- computation is (active) inference
- several classic method can be interpreted precisely as MAP inference
  - Gaussian Quadrature—Gaussian process regression
  - Runge-Kutta Methods—Autoregressive Filtering
  - [more to come on Monday]
- correct prior information can reduce runtime
- prior assumptions can be tested at runtime
- probabilistic uncertainty can be propagated between different numerical computations, allowing control of cost

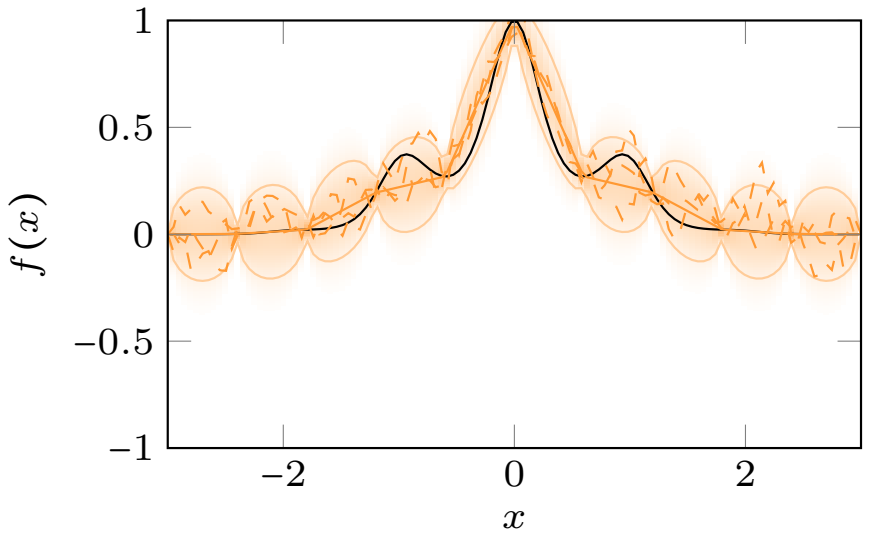
"[PN is] a re-emerging area of very active research"
Z. Ghahramani, Nature **521**, 452–459 (28 May 2015)

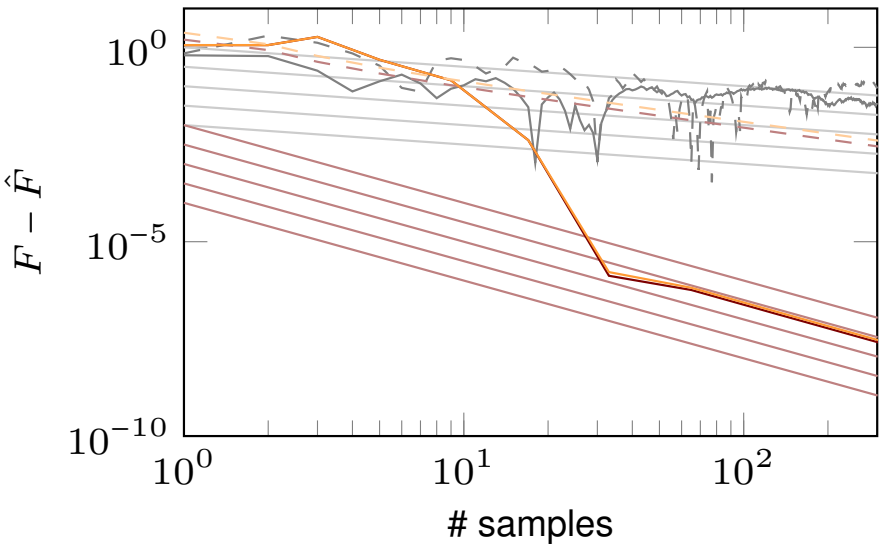
http://probabilistic-numerics.org

— Backup —

Monte Carlo is not all that Different

MC as a 'cautious' limit





$$p(f) = \mathcal{GP}(f; 0; k_\lambda^{OU} + c)$$

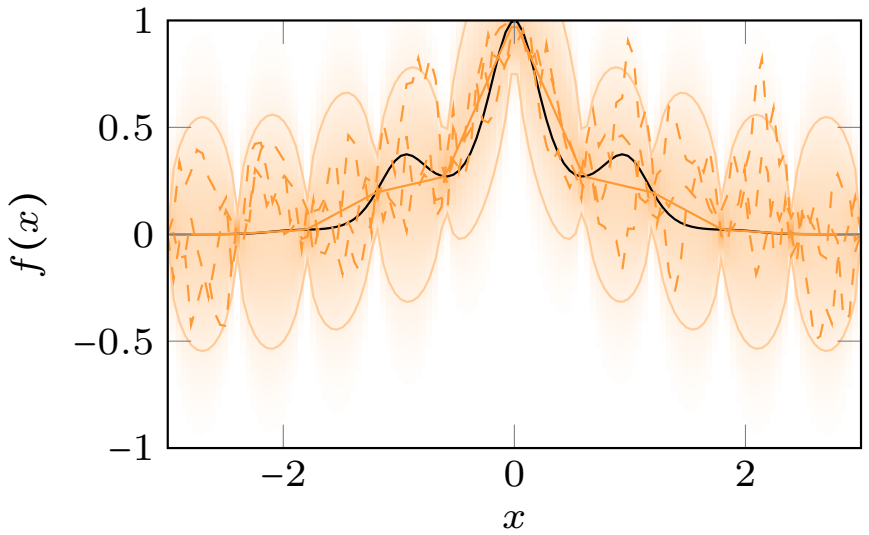
$$k_\lambda^{OU}(x, x') = \exp\left(-\frac{|x - x'|}{\lambda}\right)$$

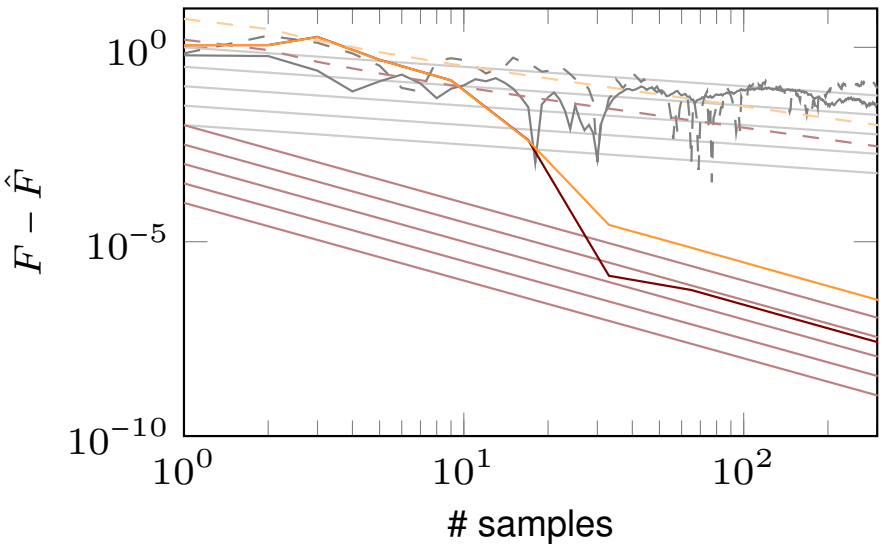
$$\lambda \to \infty \quad \Rightarrow \quad k(x, x') \stackrel{\triangle}{=} |x - x'| \quad \to \text{trapezoid rule}$$

$$\lambda \to 0 \quad \Rightarrow \quad k(x, x') \to \delta(x - x') \quad \to \text{averaging}$$

Monte Carlo is not all that Different

MC as a ‘cautious’ limit





$$p(f) = \mathcal{GP}(f; 0; k_\lambda^{OU} + c)$$

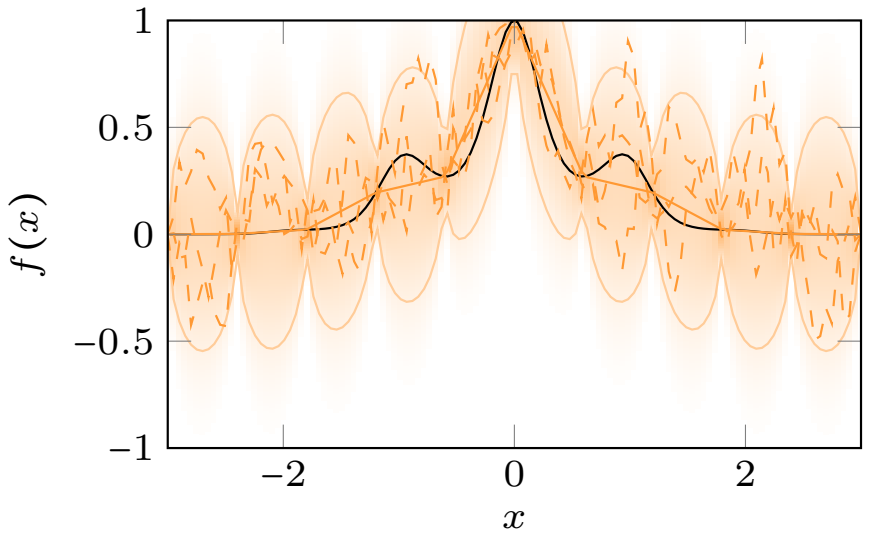
$$k_\lambda^{OU}(x, x') = \exp\left(-\frac{|x - x'|}{\lambda}\right)$$

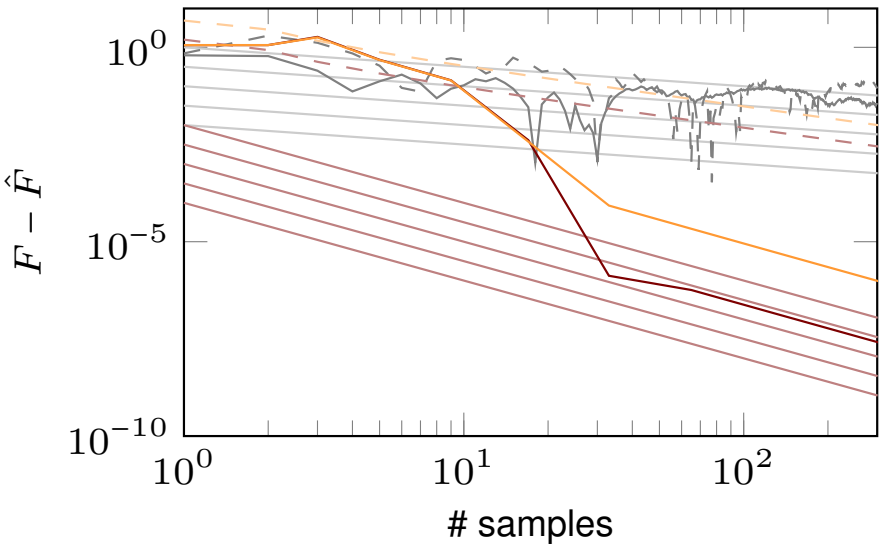
$$\lambda \to \infty \quad \Rightarrow \quad k(x, x') \stackrel{\triangle}{=} |x - x'| \quad \to \text{trapezoid rule}$$

$$\lambda \to 0 \quad \Rightarrow \quad k(x, x') \to \delta(x - x') \quad \to \text{averaging}$$

Monte Carlo is not all that Different

MC as a ‘cautious’ limit





$$p(f) = \mathcal{GP}(f; 0; k_\lambda^{OU} + c)$$

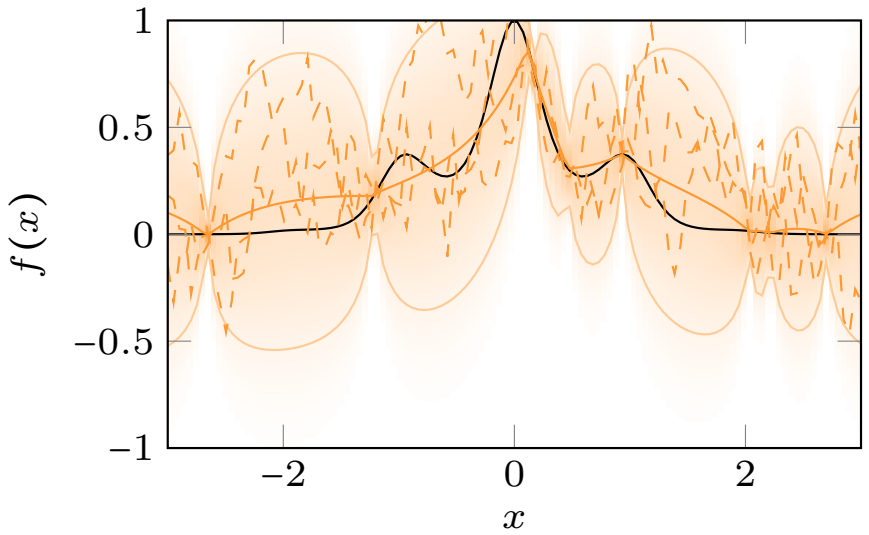
$$k_\lambda^{OU}(x, x') = \exp\left(-\frac{|x - x'|}{\lambda}\right)$$

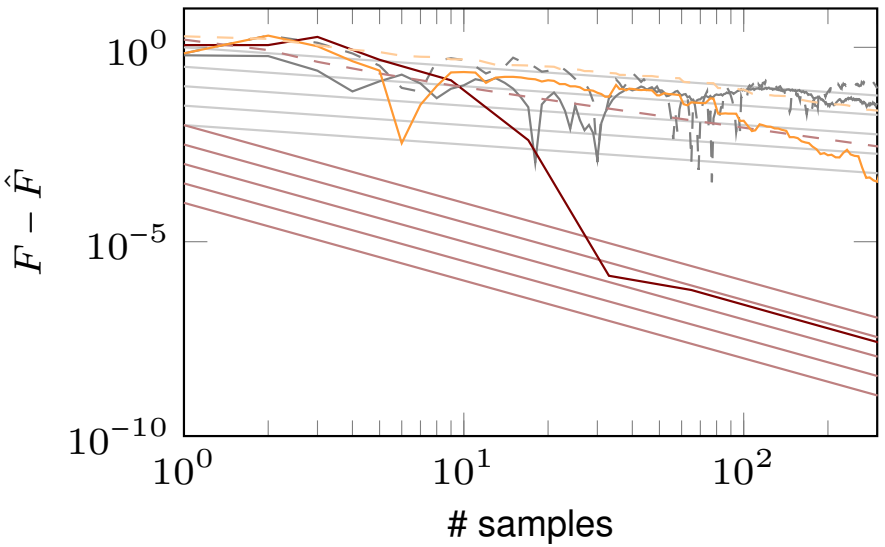
$$\lambda \to \infty \quad \Rightarrow \quad k(x, x') \stackrel{\triangle}{=} |x - x'| \quad \to \text{trapezoid rule}$$

$$\lambda \to 0 \quad \Rightarrow \quad k(x, x') \to \delta(x - x') \quad \to \text{averaging}$$

Monte Carlo is not all that Different

MC as a ‘cautious’ limit





$$p(f) = \mathcal{GP}(f; 0; k_\lambda^{OU} + c)$$

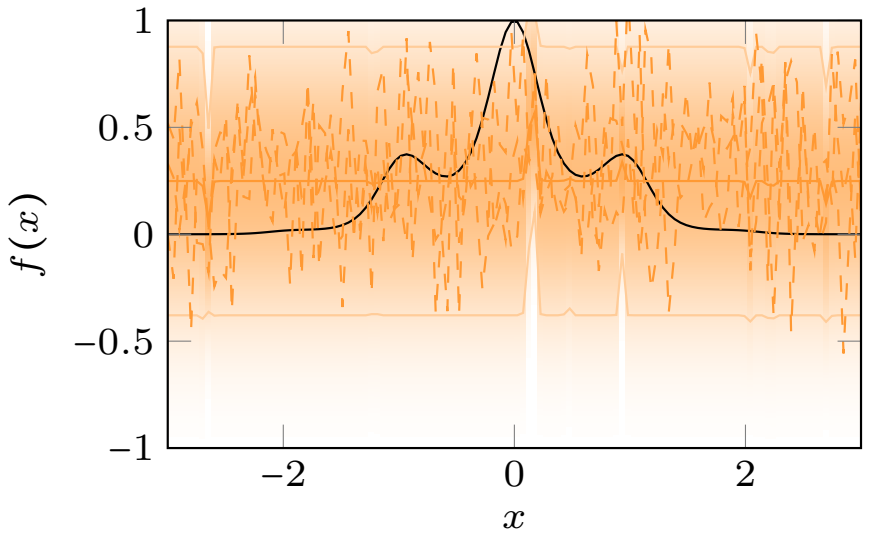
$$k_\lambda^{OU}(x, x') = \exp\left(-\frac{|x - x'|}{\lambda}\right)$$

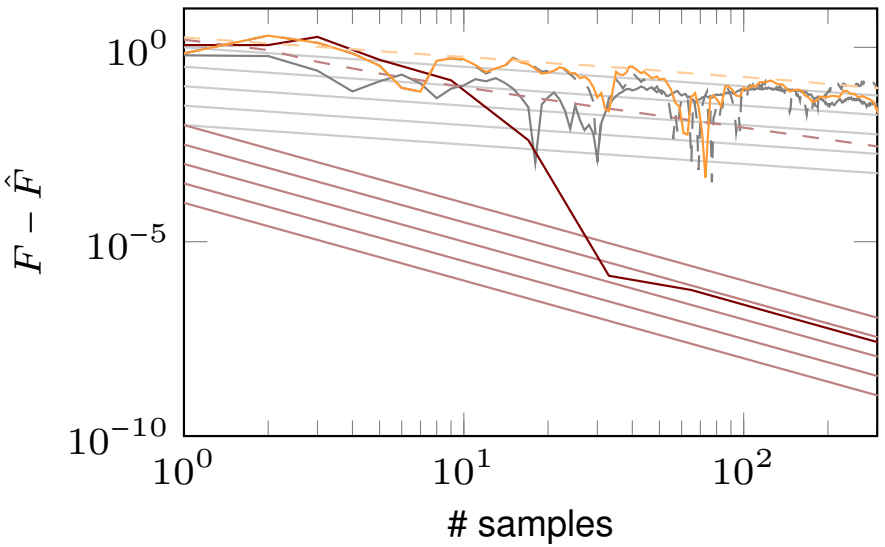
$$\lambda \to \infty \quad \Rightarrow \quad k(x, x') \stackrel{\triangle}{=} |x - x'| \quad \to \text{trapezoid rule}$$

$$\lambda \to 0 \quad \Rightarrow \quad k(x, x') \to \delta(x - x') \quad \to \text{averaging}$$

Monte Carlo is not all that Different

MC as a ‘cautious’ limit





$$p(f) = \mathcal{GP}(f; 0; k_\lambda^{OU} + c)$$

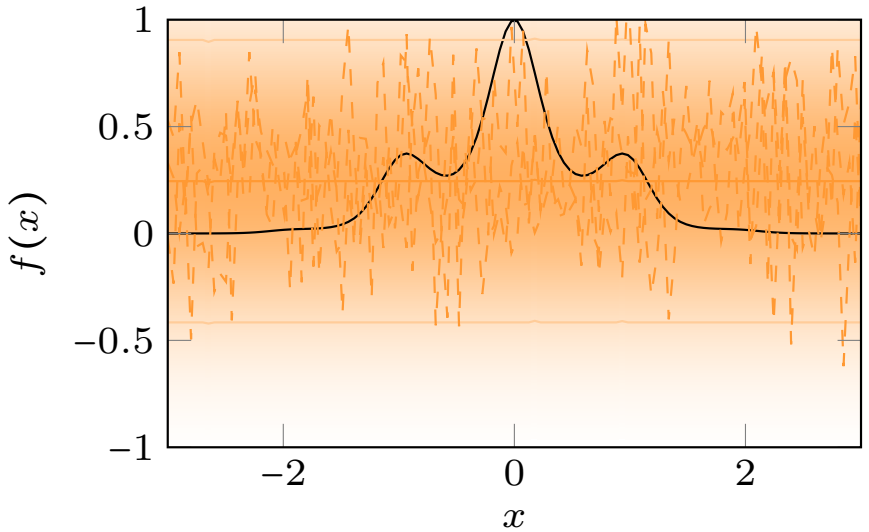
$$k_\lambda^{OU}(x, x') = \exp\left(-\frac{|x - x'|}{\lambda}\right)$$

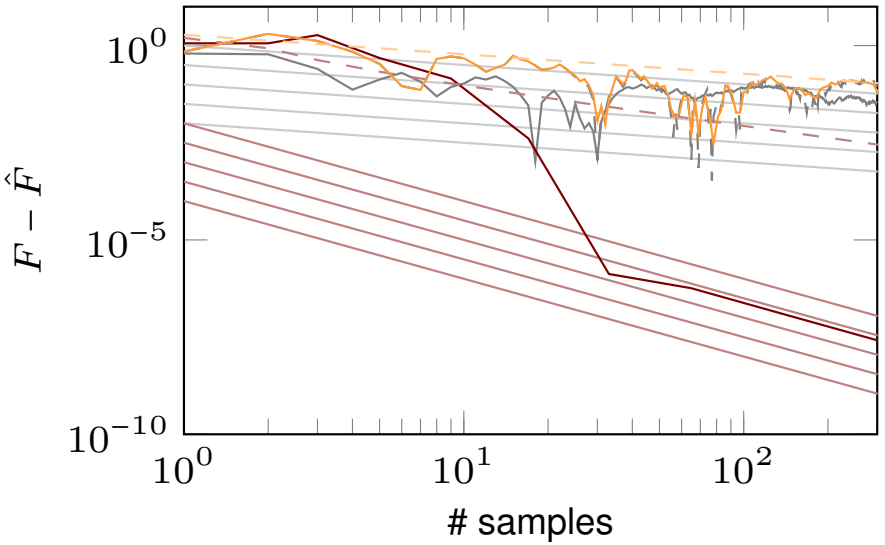
$$\lambda \to \infty \quad \Rightarrow \quad k(x, x') \stackrel{\triangle}{=} |x - x'| \quad \to \text{trapezoid rule}$$

$$\lambda \to 0 \quad \Rightarrow \quad k(x, x') \to \delta(x - x') \quad \to \text{averaging}$$

Monte Carlo is not all that Different

MC as a ‘cautious’ limit



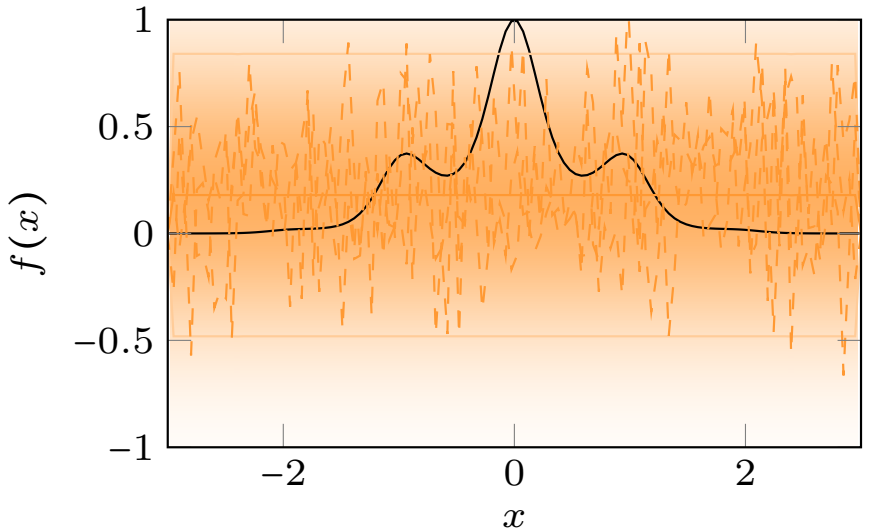


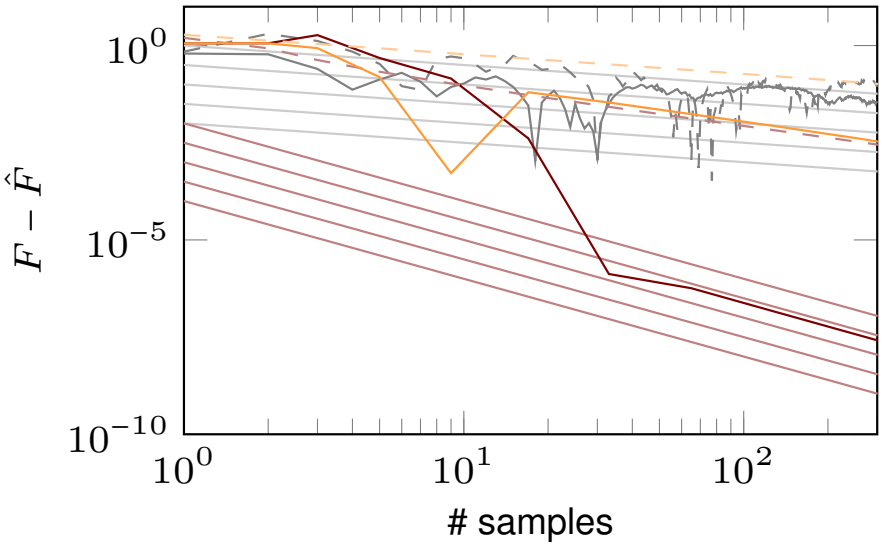
MC is optimal for totally unstructured (but integrable) f. But

- random node placement is not necessary!
  Introducing randomness to a deterministic problem is generally a bad idea.
- no integrand is totally unstructured. Prior knowledge is available!
  Computations should use as much available prior information as possible.

Monte Carlo is not all that Different

MC as a ‘cautious’ limit





MC is optimal for totally unstructured (but integrable) f. But

- random node placement is not necessary!
  Introducing randomness to a deterministic problem is generally a bad idea.
- no integrand is totally unstructured. Prior knowledge is available!
  Computations should use as much available prior information as possible.

What is a Random Number?

What is a sequence of random numbers?

- 662244111144334455553366666666
- 16939937510582097494459230781 6
- 7129042634726105902083360448 95
- 10001111110111111110010100000 1
- 01110000011100100110111101100011

What is a Random Number?

What is a sequence of random numbers?

- 662244111144334455553366666666 dice, doubled
- 16939937510582097494459230781 6
- 71290426347261059020833604489 5
- 10001111110111111110010100000 1
- 01110000011100100110111101100011

What is a Random Number?

What is a sequence of random numbers?

- 662244111144334455553366666666 — dice, doubled
- 16939937510582097494459230781 — 41-70th digits of π
- 71290426347261059020833604489 5
- 10001111110111111110010100000 1
- 01110000011100100110111101100011

What is a Random Number?

What is a sequence of random numbers?

- 662244111144334455553366666666 — dice, doubled
- 16939937510582097494459230781 6 — 41-70th digits of π
- 71290426347261059020833604489 5 — von Neumann method, seed 908344
- 10001111110111111110010100000 1
- 01110000011100100110111101100011

What is a Random Number?

What is a sequence of random numbers?

- 662244111144334455553366666666 — dice, doubled
- 16939937510582097494459230781 6 — 41-70th digits of π
- 71290426347261059020833604489 5 — von Neumann method, seed 908344
- 10001111110111111110010100000 1 — bits from G. Marsaglia's 'diehard CD'
- 01110000011100100110111101100011

What is a Random Number?

What is a sequence of random numbers?

- 662244111144334455553366666666 — dice, doubled
- 16939937510582097494459230781 6 — 41-70th digits of π
- 71290426347261059020833604489 5 — von Neumann method, seed 908344
- 10001111110111111110010100000 1 — bits from G. Marsaglia's 'diehard CD'
- 01110000011100100110111101100011 — deterministic sequence, corrupted by horizontal coin drop from unknown height

`http://www.stat.fsu.edu/pub/diehard/`

What is a Random Number?

What is a sequence of random numbers?

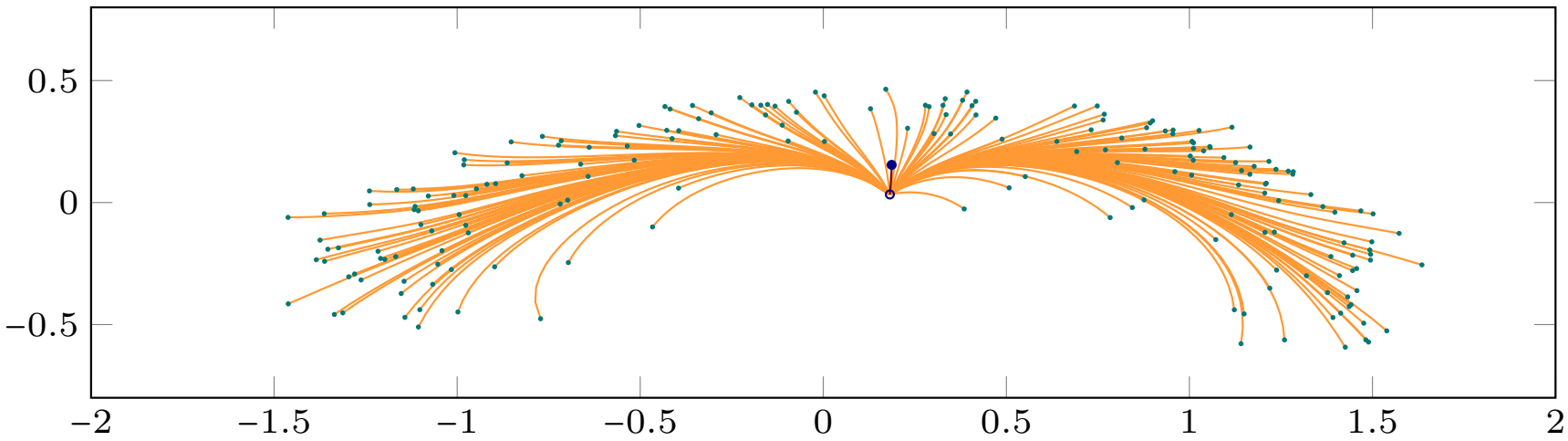
- 66224411114433445555336666666 — dice, doubled
- 16939937510582097494459230781 — 41-70th digits of π
- 71290426347261059020833604489 — von Neumann method, seed 908344
- 10001111110111111110010100001 — bits from G. Marsaglia's 'diehard CD'
- 01110000011100100110111101100011 — deterministic sequence, corrupted by horizontal coin drop from unknown height

- for use in Monte Carlo, the important property is freedom from patterns (because it implies anytime unbiasedness)
- in fact, use for MC only really requires the right density, disorder is just helpful for the argument
- for use in cryptography, the important property is unpredictability
- randomness is a philosophically dodgy concept
- uncertainty is a much clearer idea

`http://www.stat.fsu.edu/pub/diehard/`

How Expensive is a Computation?

Ex: Riemannian Statistics [Schober, Hauberg,Lawrence, Hennig; in prep.]



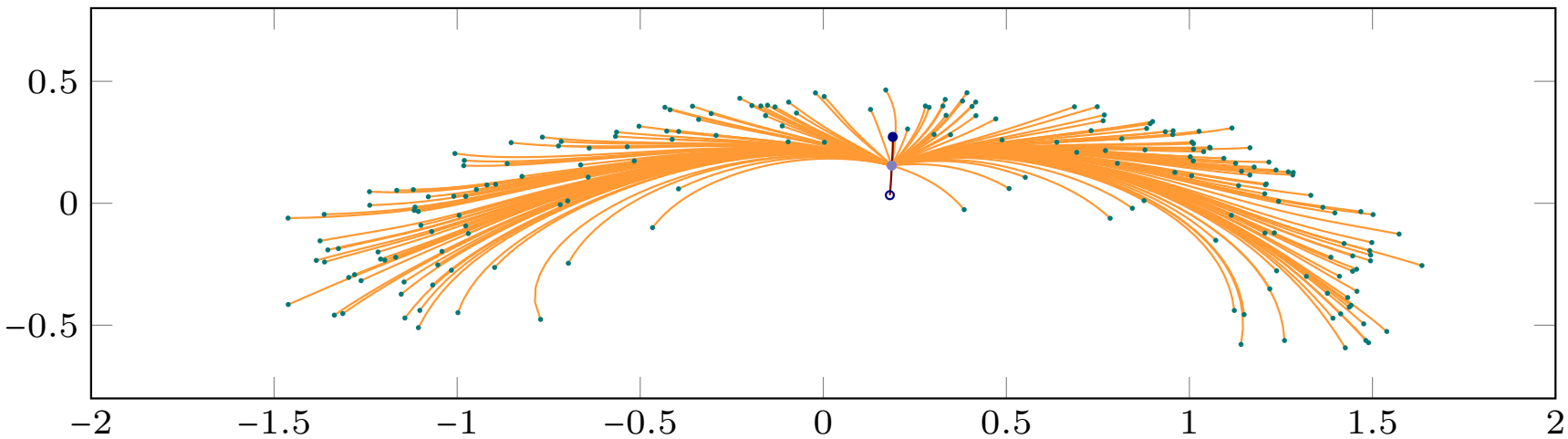
- Shortest paths (geodesics) on Riemannian manifold of metric M obey

$$x''(t) = -1/2 M^{-1}(x(t)) \left[\frac{\partial \overrightarrow{M}(x(t))}{\partial x(t)} \right]^{\top} (x'(t) \otimes x'(t)) = f(x(t), x'(t), t)$$

- Karcher mean μ of data set $\{x_i\}_{i=1,\ldots,N}$ is point minimizing $\sum_i^N$ Distance(μ, x_i)

How Expensive is a Computation?

Ex: Riemannian Statistics [Schober, Hauberg,Lawrence, Hennig; in prep.]



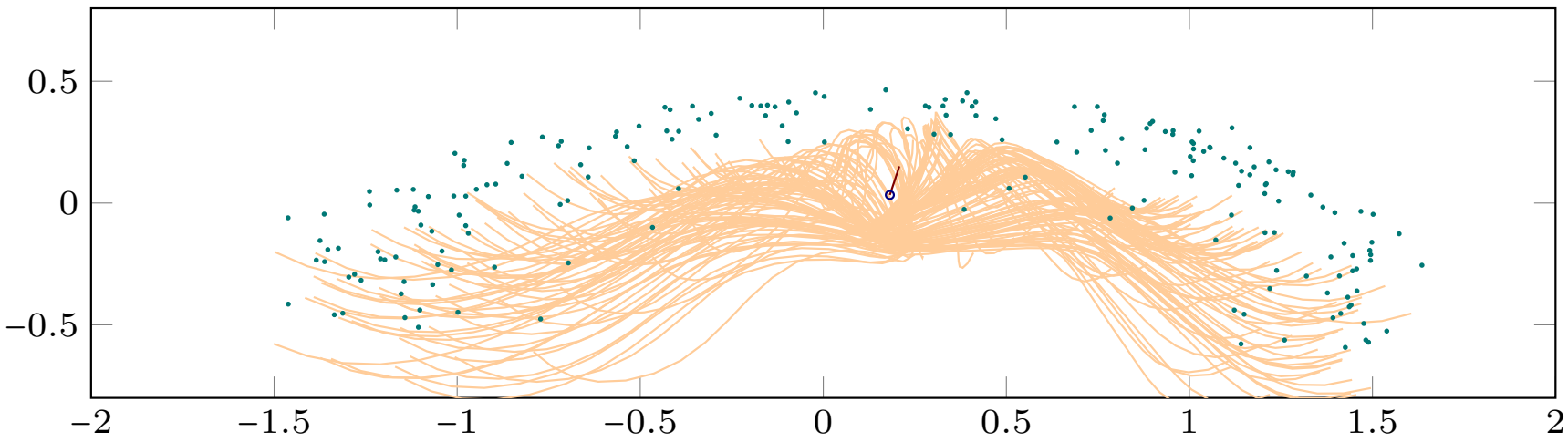
- to find Karcher mean, do gradient descent from initial guess μ_0

$$\mu_{k+1} = \mu_k - \alpha \nabla_\mu \sum_i^N \text{Distance}(\mu, x_i)$$

- requires solving N initial value problems, over and over.

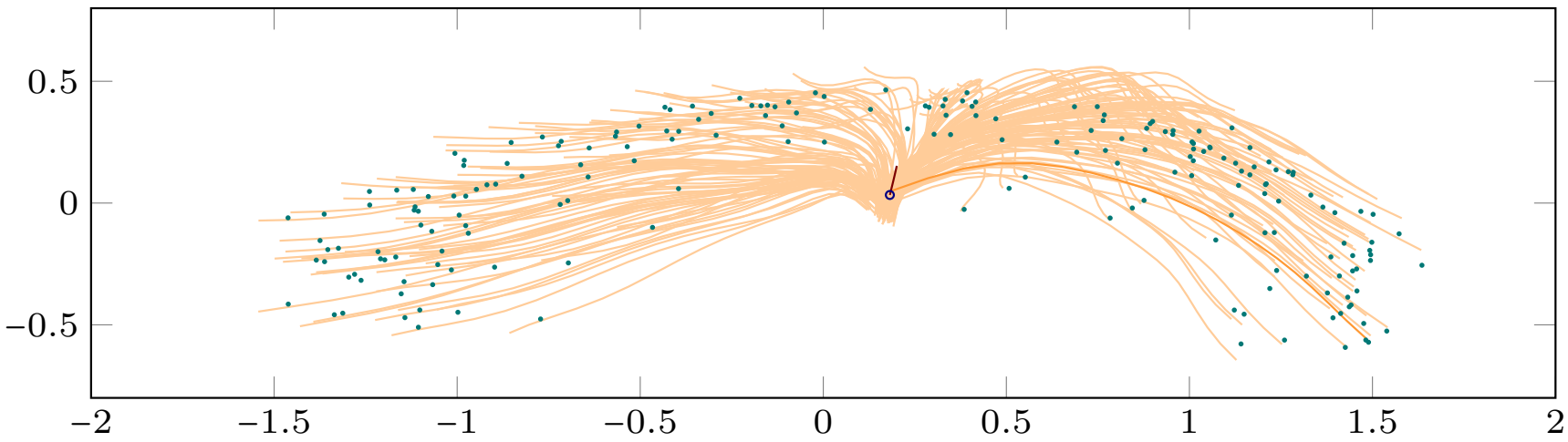
How Much Information is Needed?

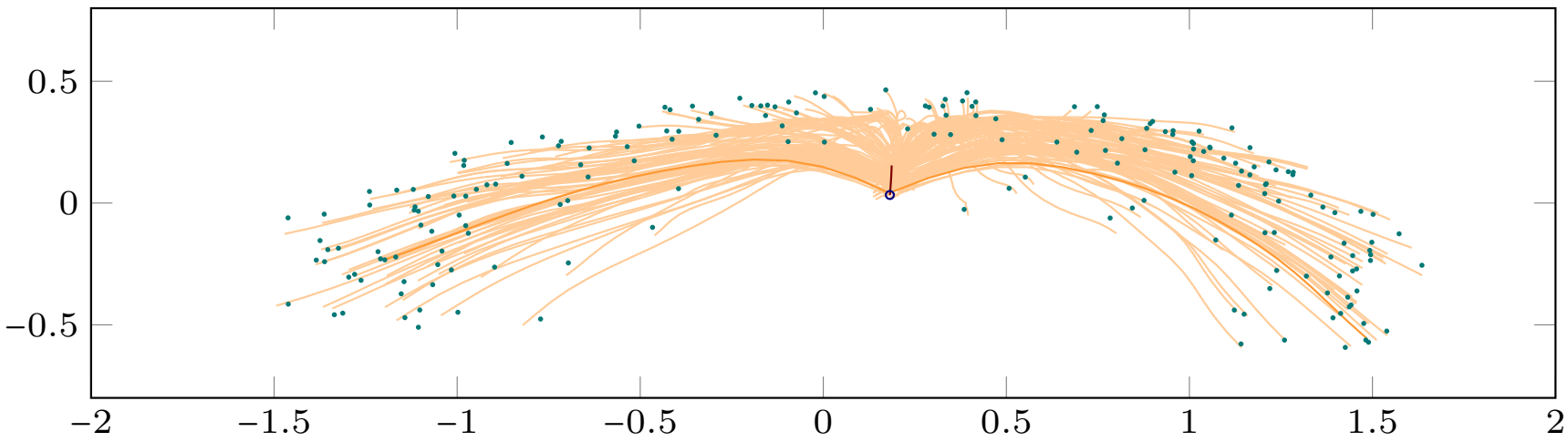
Ex: Riemannian Statistics [Schober, Hauberg,Lawrence, Hennig; in prep.]



How Much Information is Needed?

Ex: Riemannian Statistics [Schober, Hauberg,Lawrence, Hennig; in prep.]



How Much Information is Needed?

Ex: Riemannian Statistics [Schober, Hauberg,Lawrence, Hennig; in prep.]