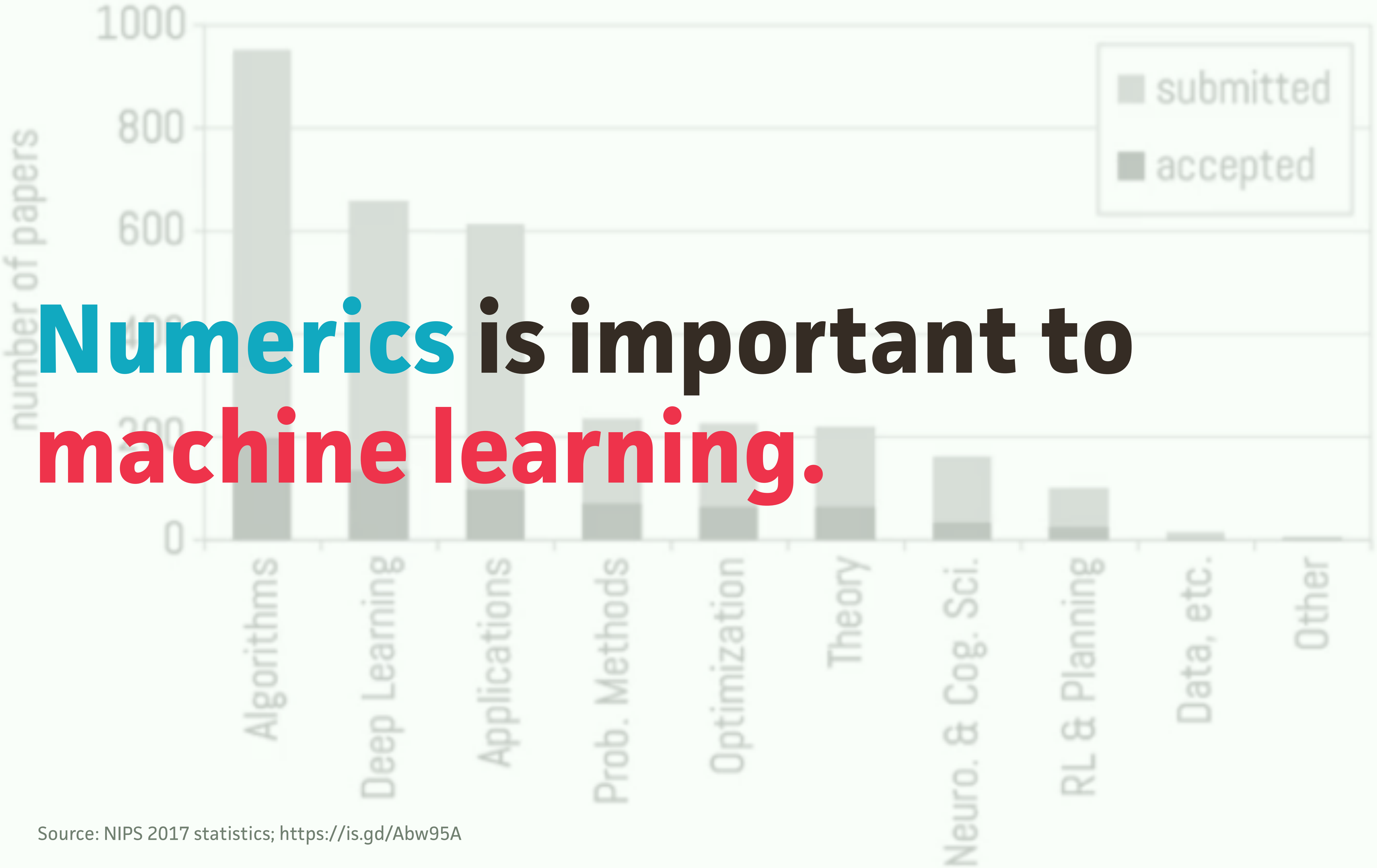
PROBABILISTIC NUMERICS:

NANO-MACHINE-LEARNING

Michael A Osborne, @maosbot

Numerics is important to machine learning.

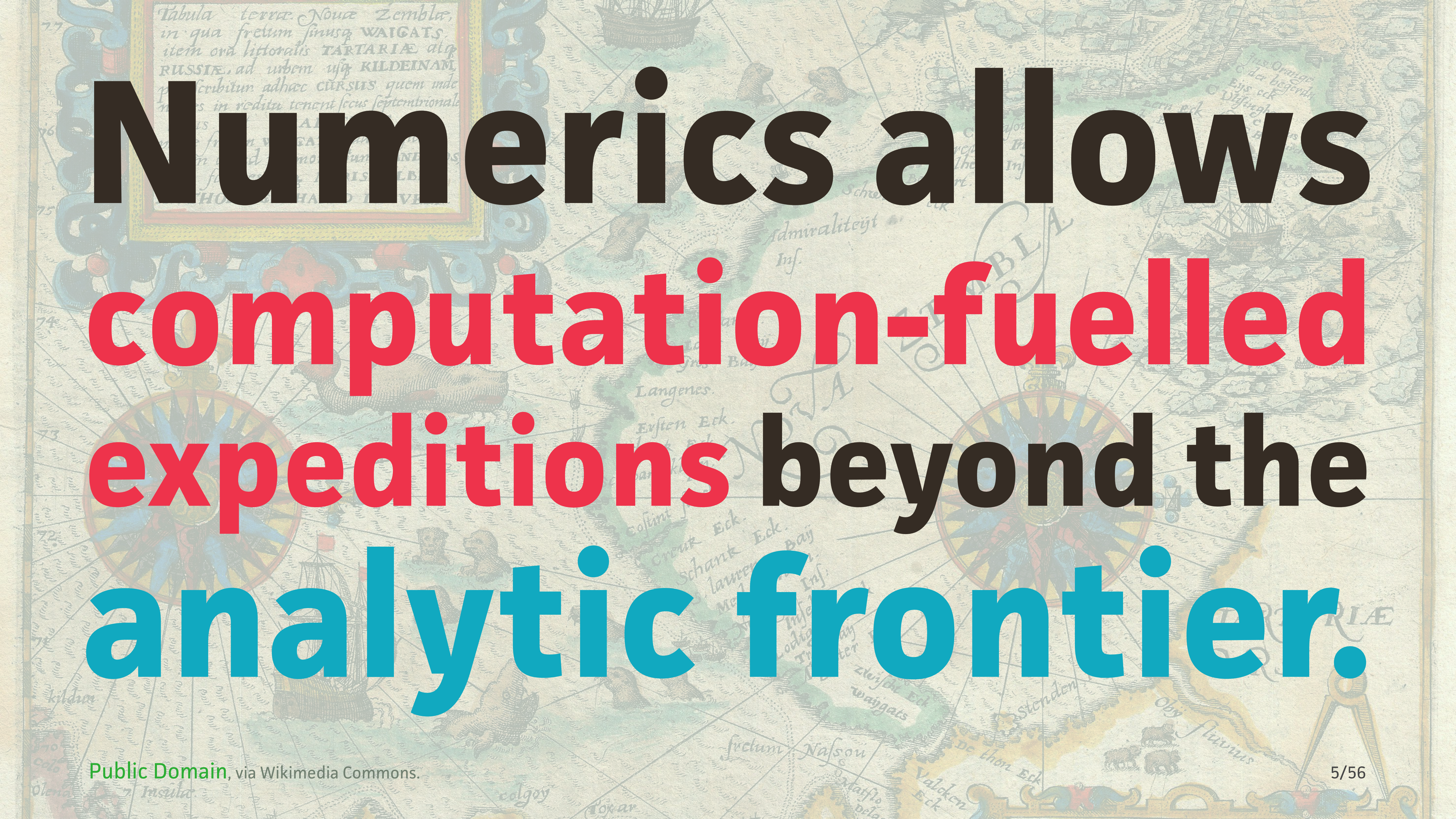


Source: NIPS 2017 statistics; https://is.gd/Abw95A

Which numerics problems have you needed solved in the last month?

1. Linear algebra.
2. Optimisation.
3. Global optimisation.
4. Integration.
5. Ordinary differential equations.

Public Domain, via Wikimedia Commons



Numerics allows computation-fuelled expeditions beyond the analytic frontier.

Public Domain, via Wikimedia Commons.

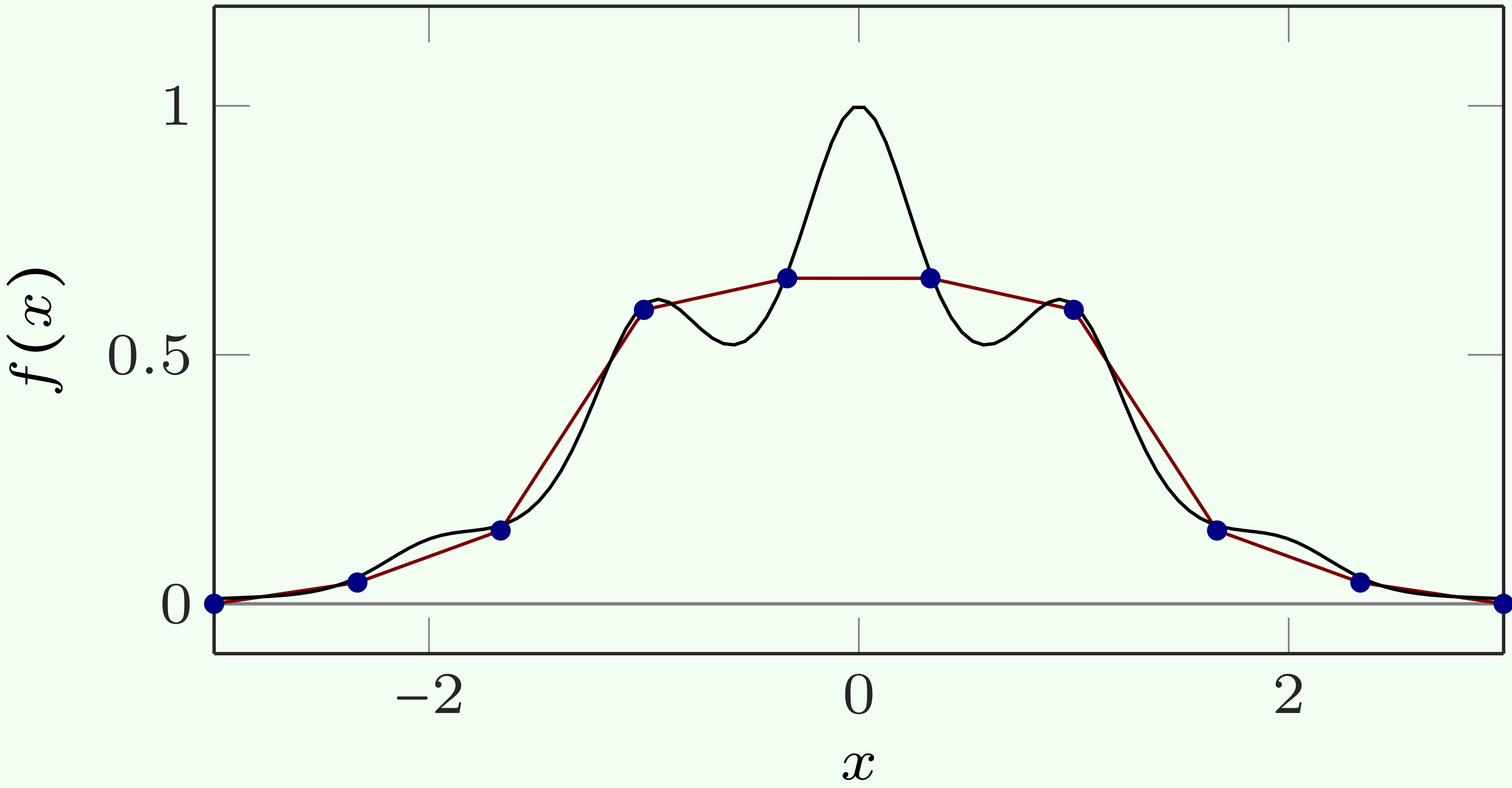
The answer to a numeric problem can only be approximated,

e.g.

$$F = \int_{-3}^{3} f(x)\mathrm{d}x$$

for

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



Machine learning treats algorithms as agents.

Probabilistic numerics treats numeric algorithms as agents.

As motivation:

1. numeric **error** is significant;
2. numeric methods are **generic;**
3. our numerics problems **tax our computation.**

By ESA/Hubble, CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=8788068.

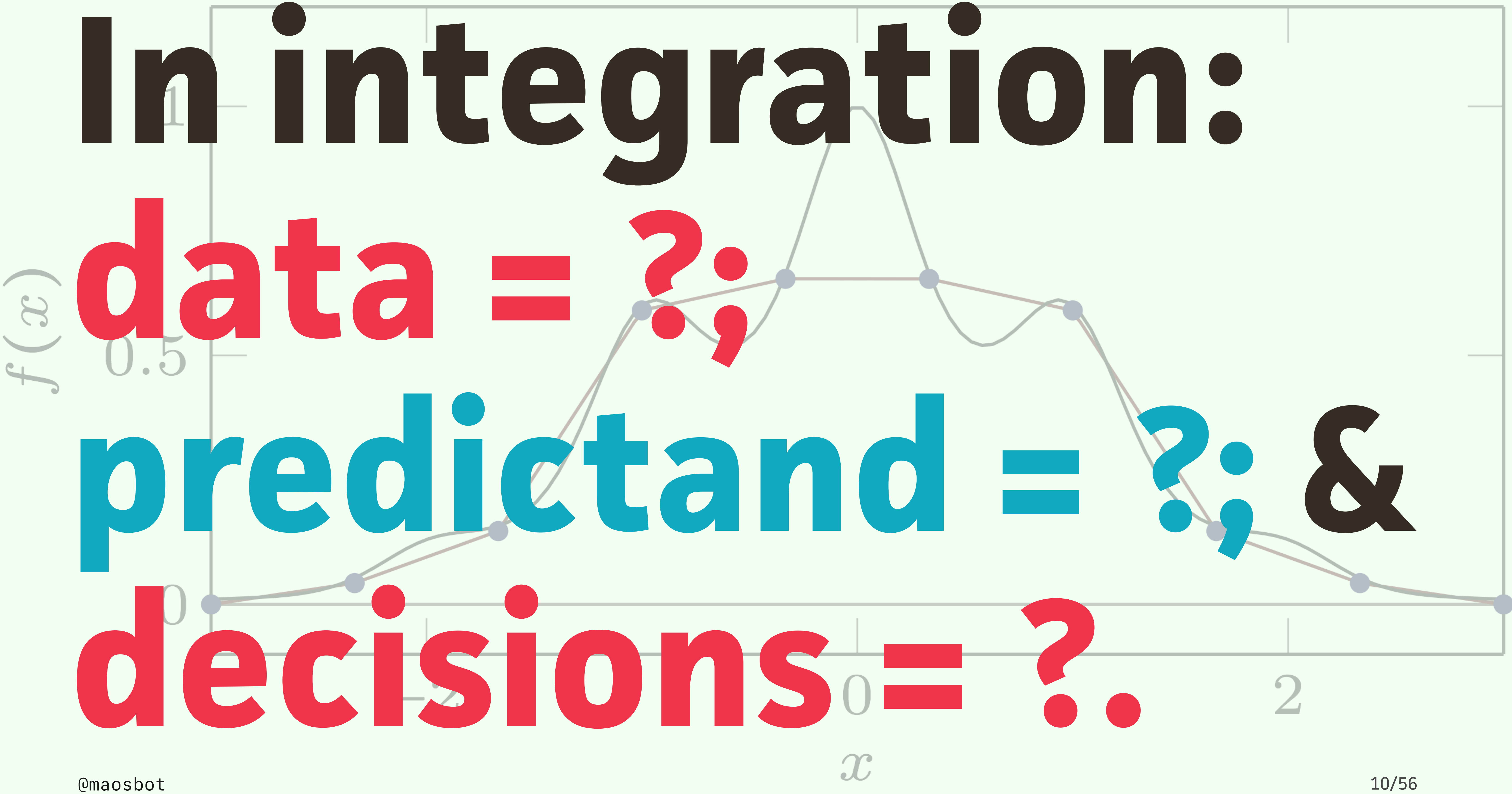
An agent receives data, predicts, & then makes decisions.

In integration:

data = ?;

predictand = ?; &

decisions = ?.



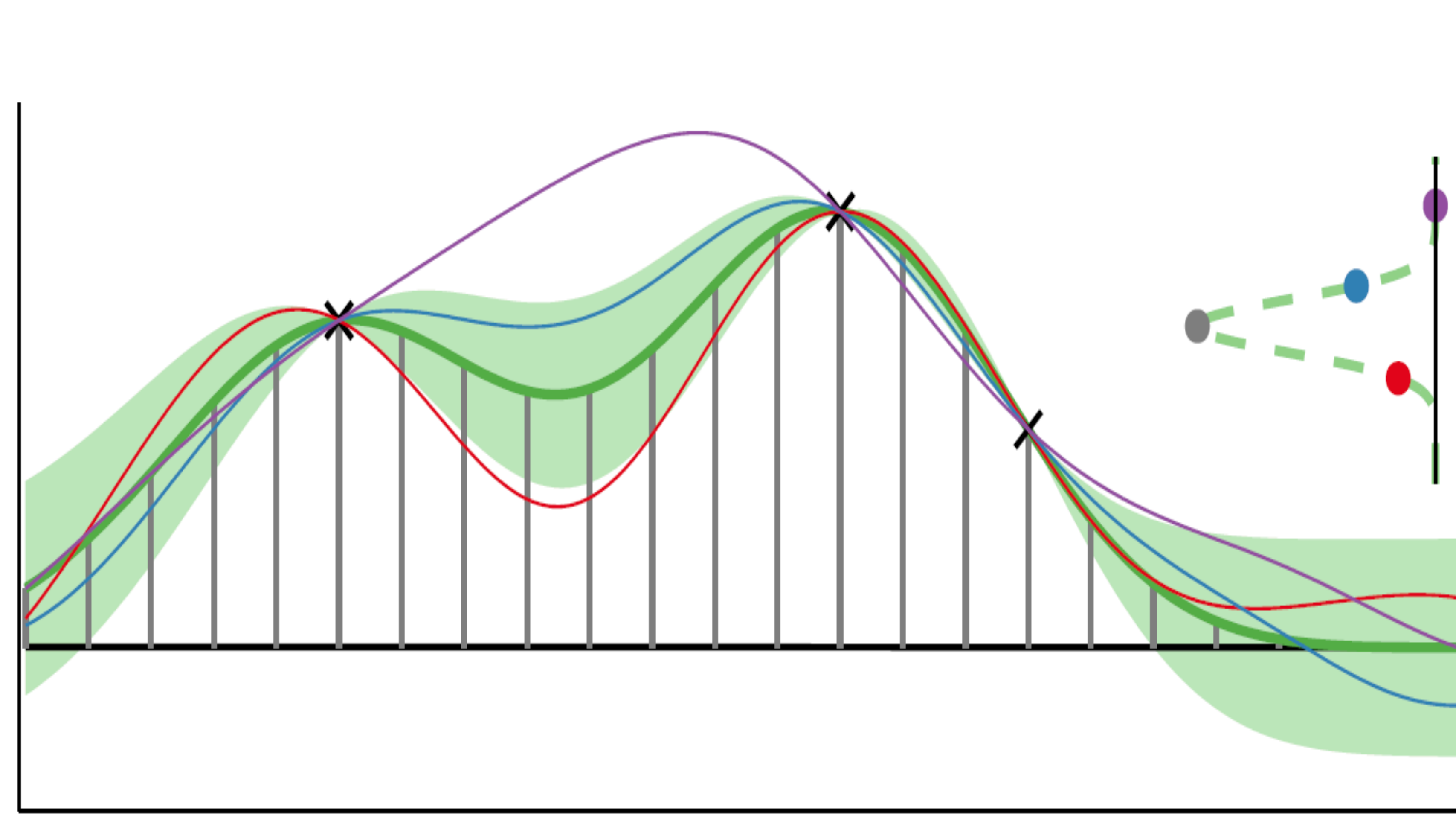
In integration:

data = evaluations;

predictand = integral; &

decisions = locations.

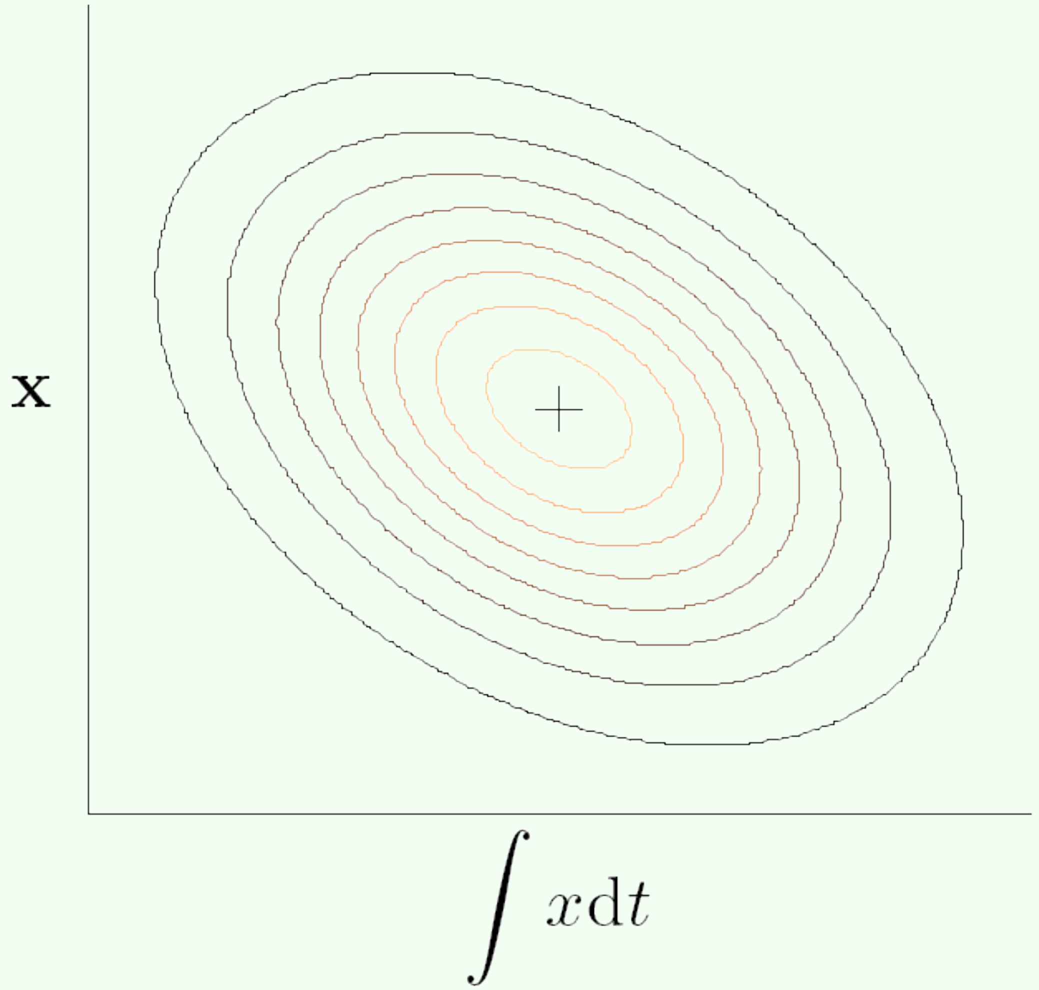
Bayesian quadrature is probabilistic numerics for integration.

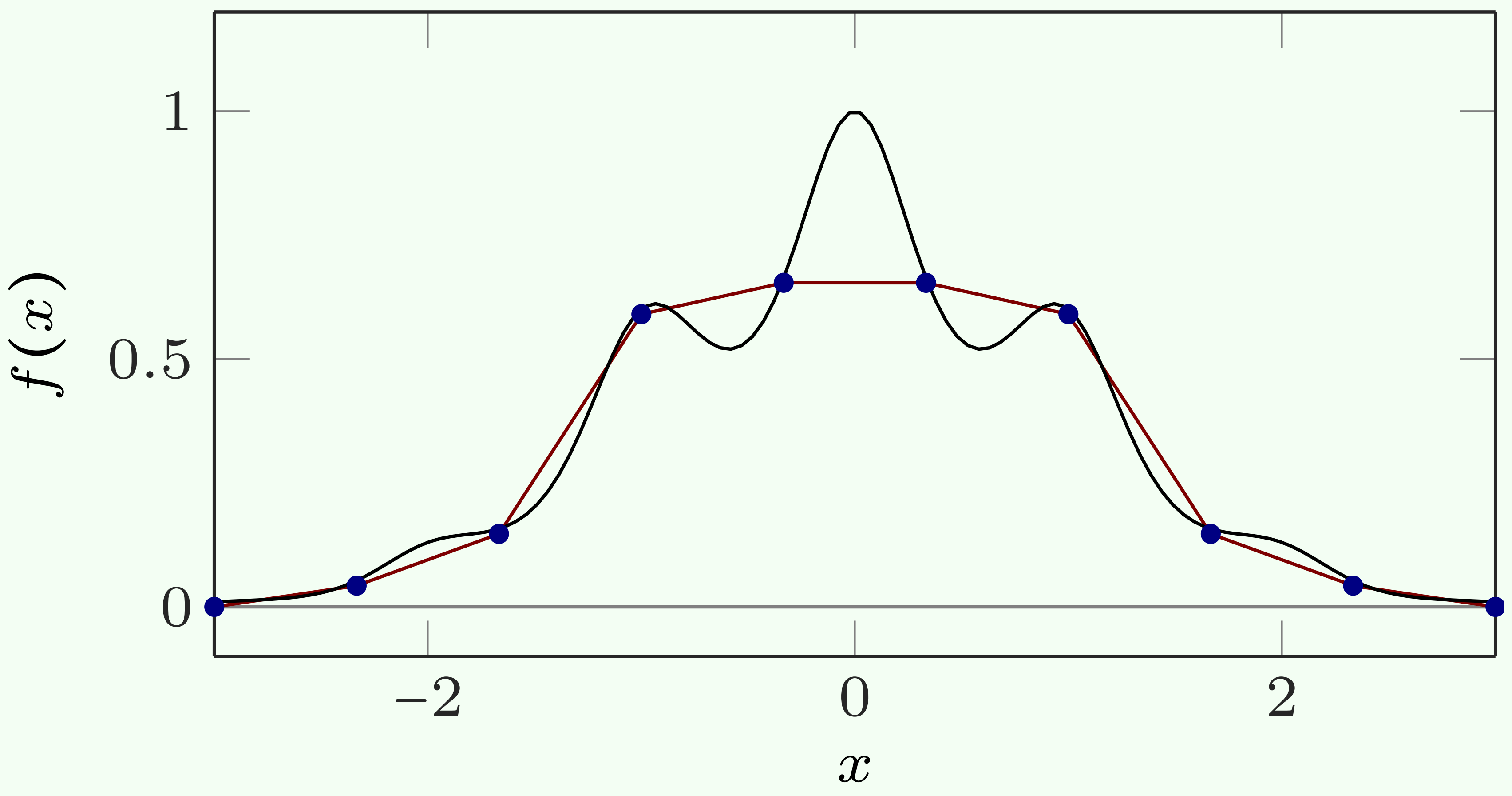


Source: D. Duvenaud.

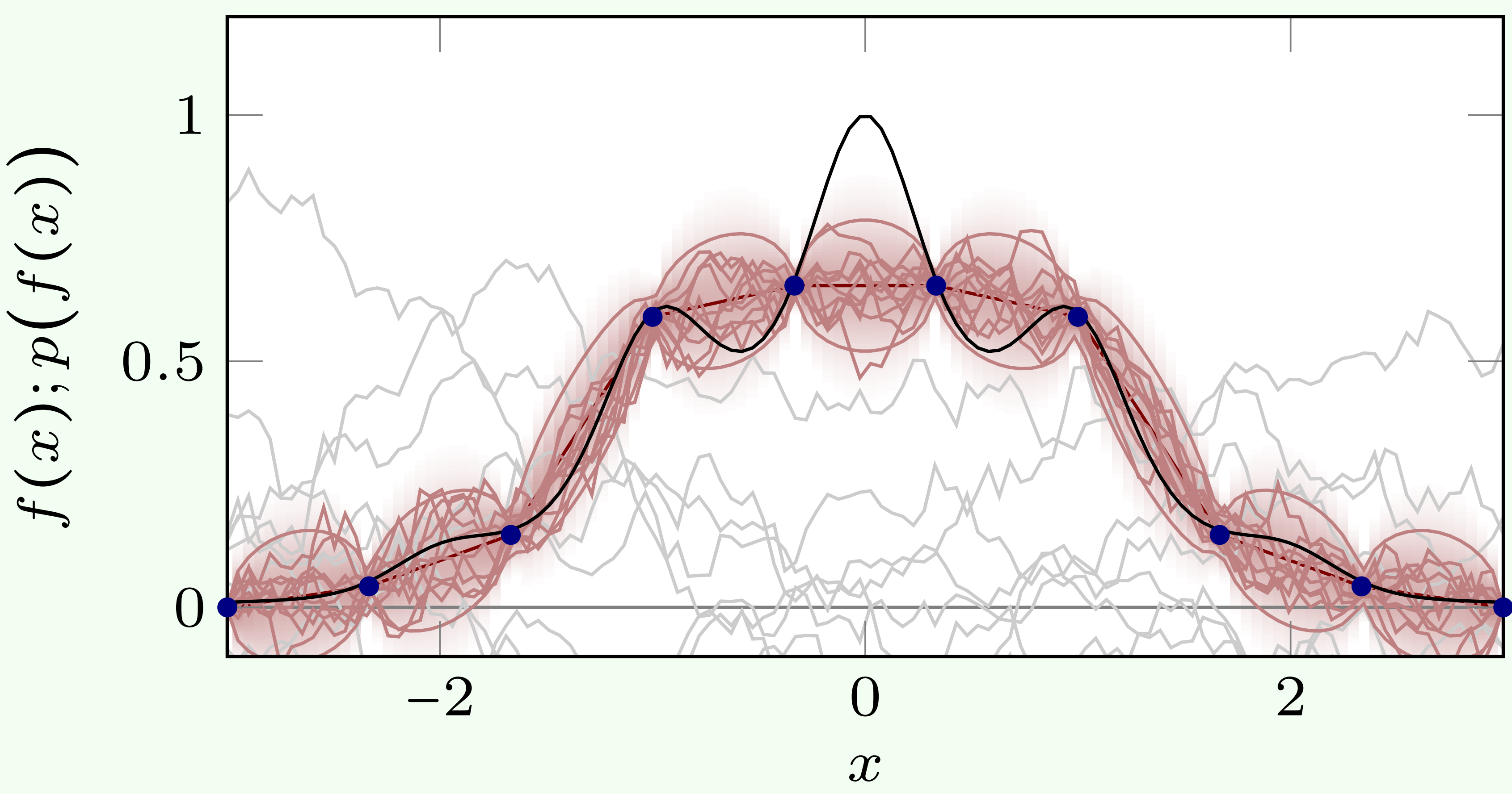
An agent is defined by its **prior** and loss function.

With a Gaussian process prior for the integrand, the integral is joint Gaussian.





$f(x)$
x
1
0.5
0
−2
0
2



$f(x); p(f(x))$
x
1
0.5
0
−2
0
2

The trapezoidal rule is the posterior mean estimate for the integral

$$F = \int_a^b f(x)\,\mathrm{d}x$$

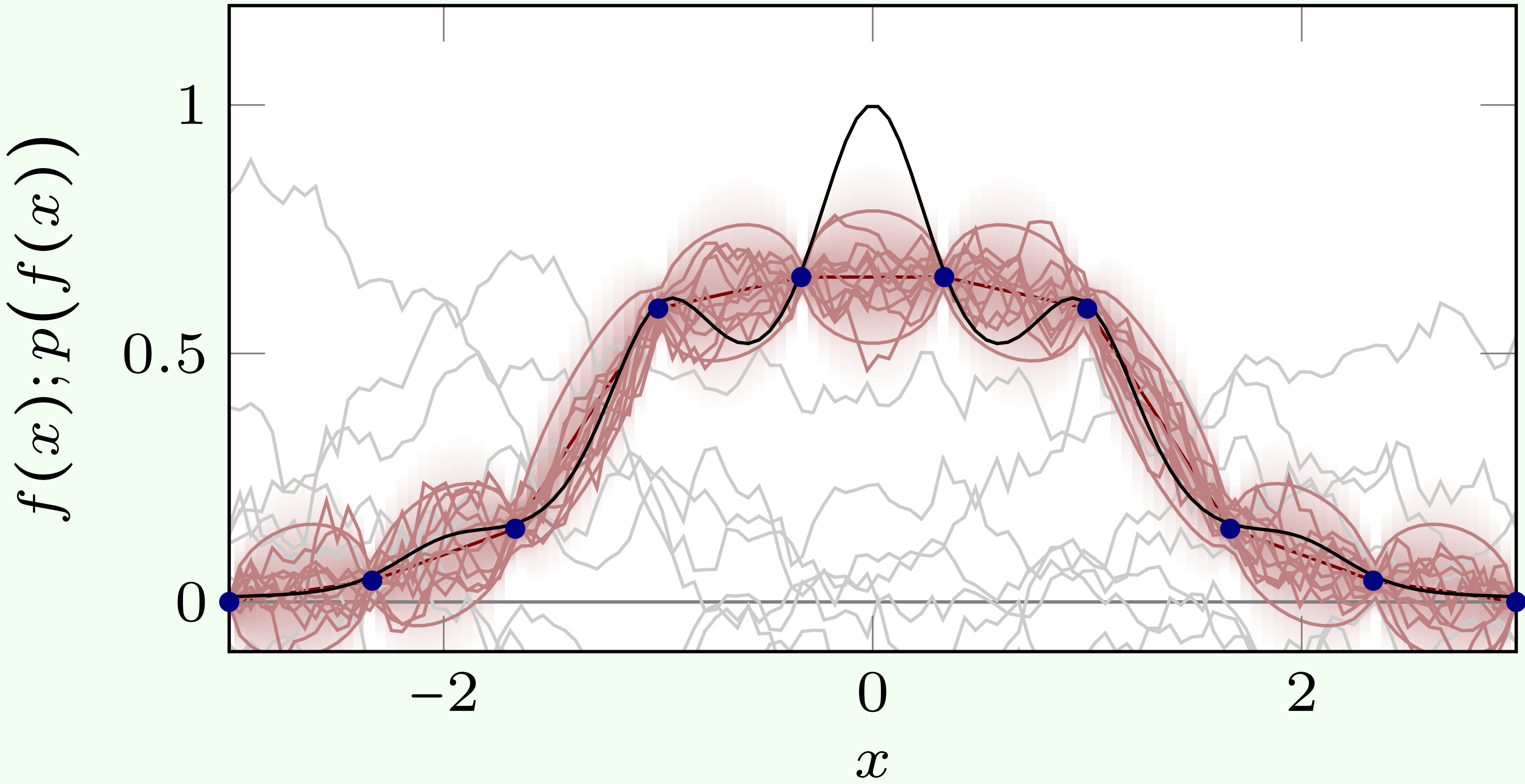
under any centered Wiener process prior

$$p(f) = \mathcal{GP}(f; 0, k)$$

with

$$k(x, x') = \theta^2 \left(\min(x, x') - \chi\right)$$

for arbitrary $\theta \in \mathfrak{R}_+$ and $\chi < a \in \mathfrak{R}$.



procedure INTEGRATE($@f, a, b, N, \theta$)

  $\delta := (b - a)/(N - 1)$ // choose step size

  $x \leftarrow a, y_1 = f(a), m \leftarrow 0, v \leftarrow 0,$ // initialise

  for $i = 2, \ldots, N$ **do**

    $x \leftarrow x + \delta$ // step

    $y_i \leftarrow f(x)$ // evaluate

    $m \leftarrow m + {}^{\delta}/2(y_{i-1} + y_i)$ // update estimate

    $v \leftarrow v + {}^{\delta^3}/12$ // update error estimate

  end for

  return $\mathbb{E}(F) = m, \mathrm{var}(F) = \theta^2 v$ // probabilistic output

end procedure

The trapezoid rule **is** Bayesian quadrature.

procedure INTEGRATE($@f, a, b, N, \theta$)

  $\delta := (b-a)/(N-1)$ // choose step size

  $x \leftarrow a, y_1 = f(a), m \leftarrow 0, v \leftarrow 0,$ // initialise

  for $i = 2, \dots, N$ **do**

    $x \leftarrow x + \delta$ // step

    $y_i \leftarrow f(x)$ // evaluate

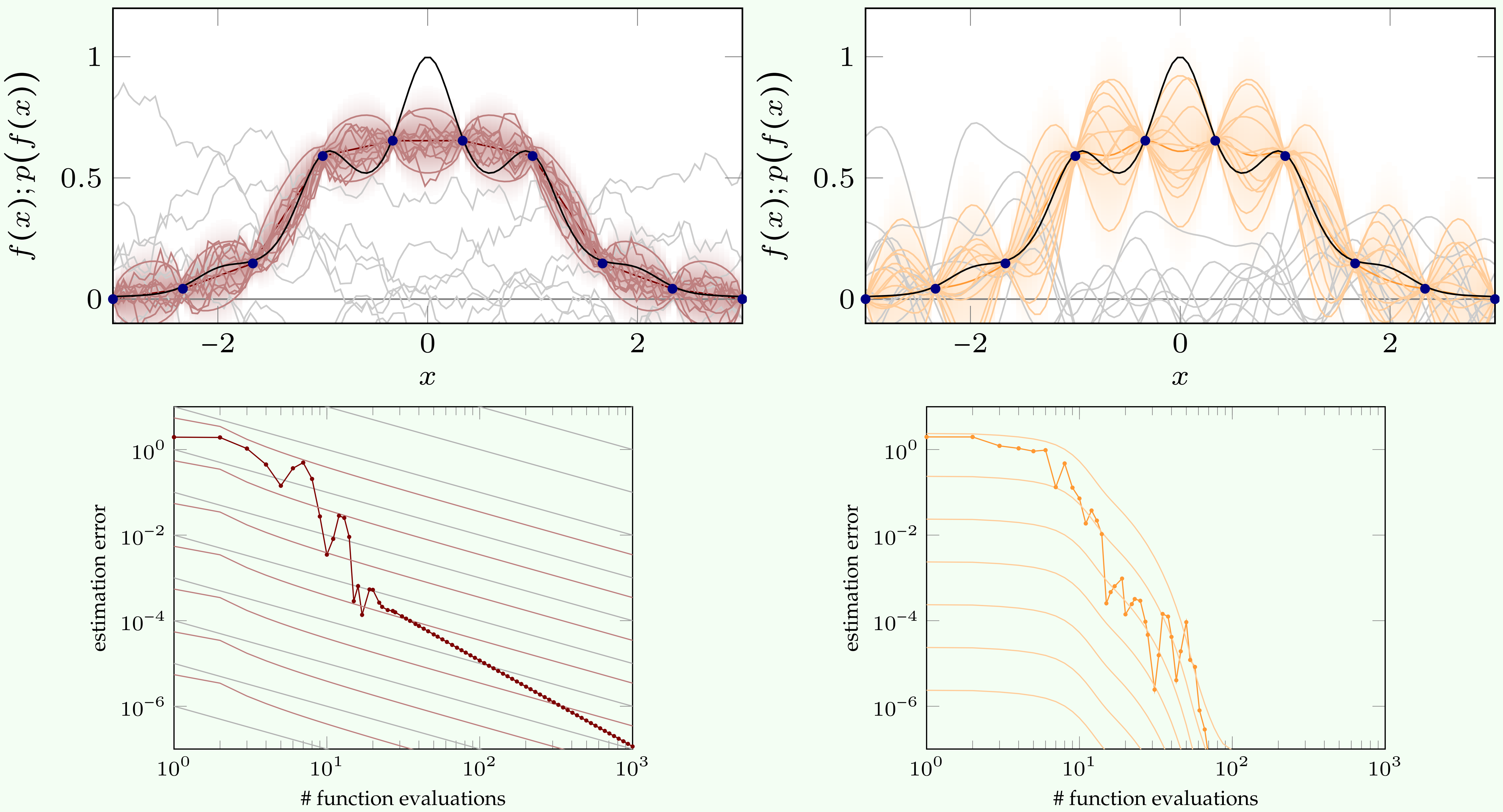
    $m \leftarrow m + \delta/2(y_{i-1} + y_i)$ // update estimate

    $v \leftarrow v + \delta^3/12$ // update error estimate

  end for

  return $\mathbb{E}(F) = m, \mathrm{var}(F) = \theta^2 v$ // probabilistic output

end procedure



f(x); p(f(x))
x
f(x); p(f(x))
x
estimation error
function evaluations
estimation error
function evaluations

Quiz: The convergence rate of the trapezoid rule is $\mathcal{O}(N^{-1})$: what is the rate of **Monte Carlo?**

1. $\mathcal{O}\big(\exp(-N)\big)$
2. $\mathcal{O}\left(\exp\left(-N^{-\frac{1}{2}}\right)\right)$
3. $\mathcal{O}(N^{-1})$
4. $\mathcal{O}(N^{-\frac{1}{2}})$

Quiz: The convergence rate of the trapezoid rule is $\mathcal{O}(N^{-1})$: what is the rate of Monte Carlo?

1. $\mathcal{O}\big(\exp(-N)\big)$
2. $\mathcal{O}\left(\exp\left(-N^{-\frac{1}{2}}\right)\right)$
3. $\mathcal{O}(N^{-1})$
4. $\boldsymbol{\mathcal{O}(N^{-\frac{1}{2}})}$ – **arguably the worst possible rate.**

Monte Carlo is also Bayesian quadrature.

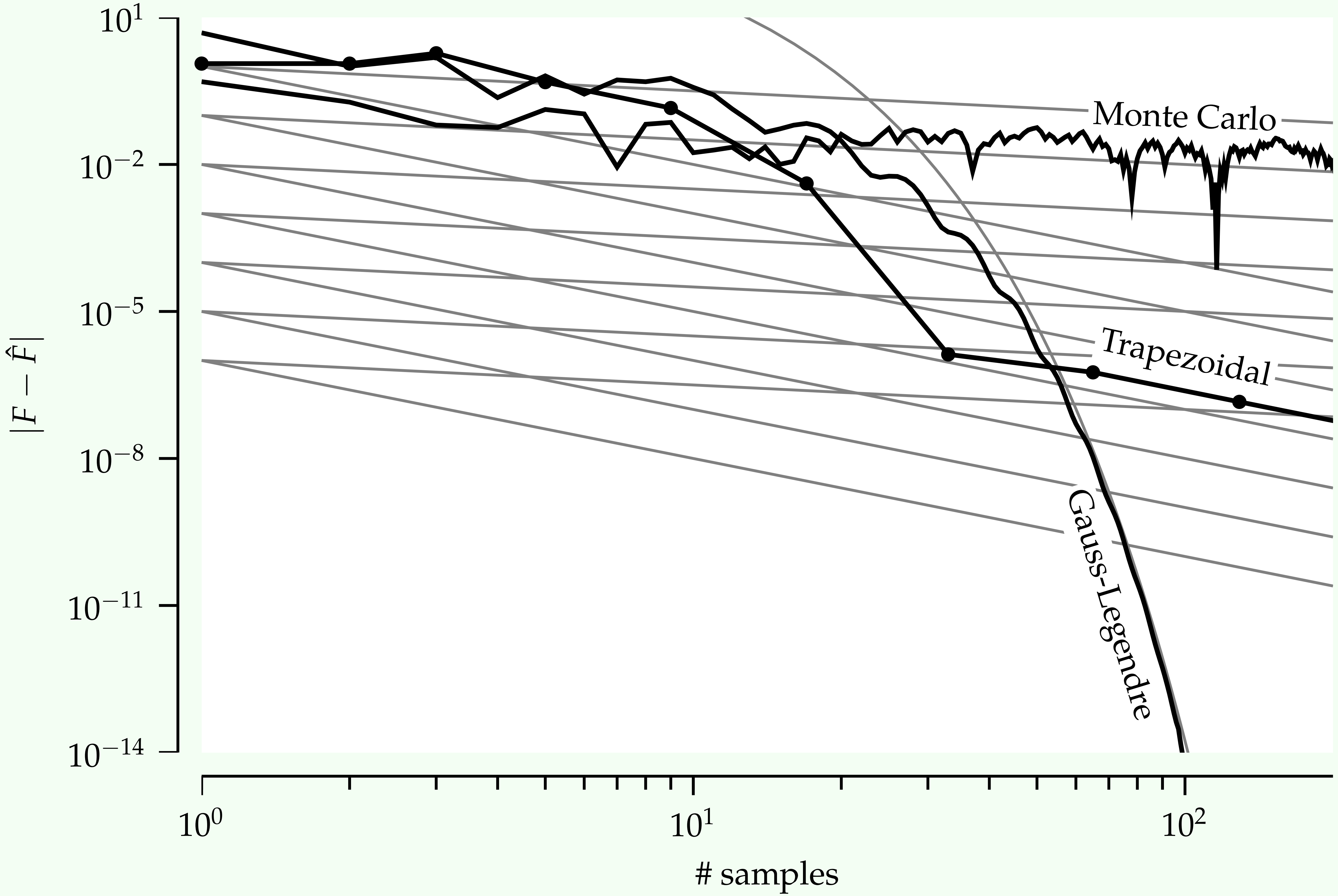
The Monte Carlo estimate

$$\int f(x)\,p(x)\,\mathrm{d}x \simeq \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

is maximum a-posteriori under the (improper) prior

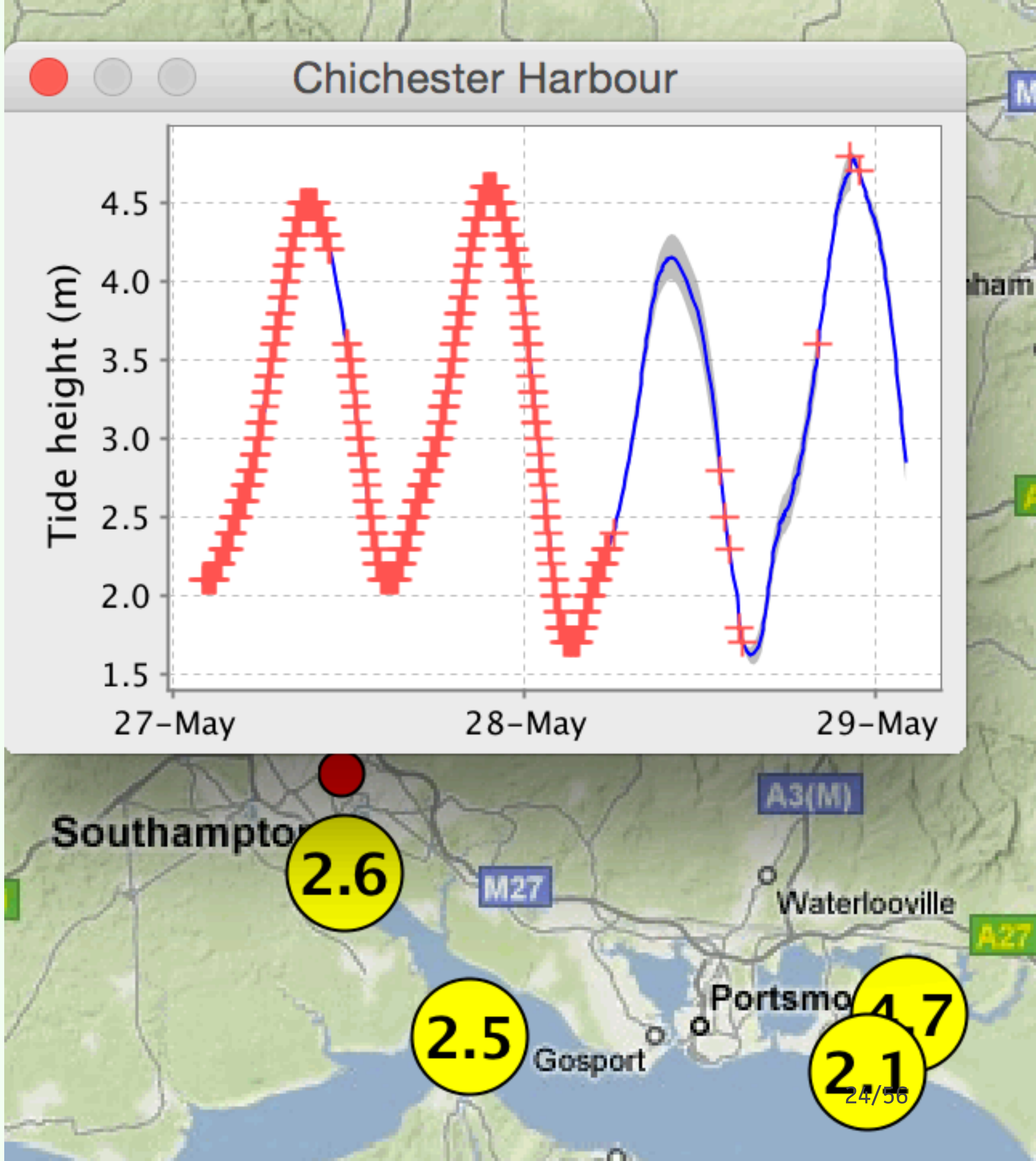
$$p(f) = \lim_{c\to 0} \mathcal{GP}\big(0, \theta^2\, \mathbb{I}(x = x') + c^{-1}\big)$$

for $\mathbb{I}$ the indicator function and with arbitrary $\theta \in \Re_+$. The corresponding posterior standard deviation estimate on the integral, $\theta\,(b-a)\big/\sqrt{N}$, matches the convergence rate of the Monte Carlo estimator.



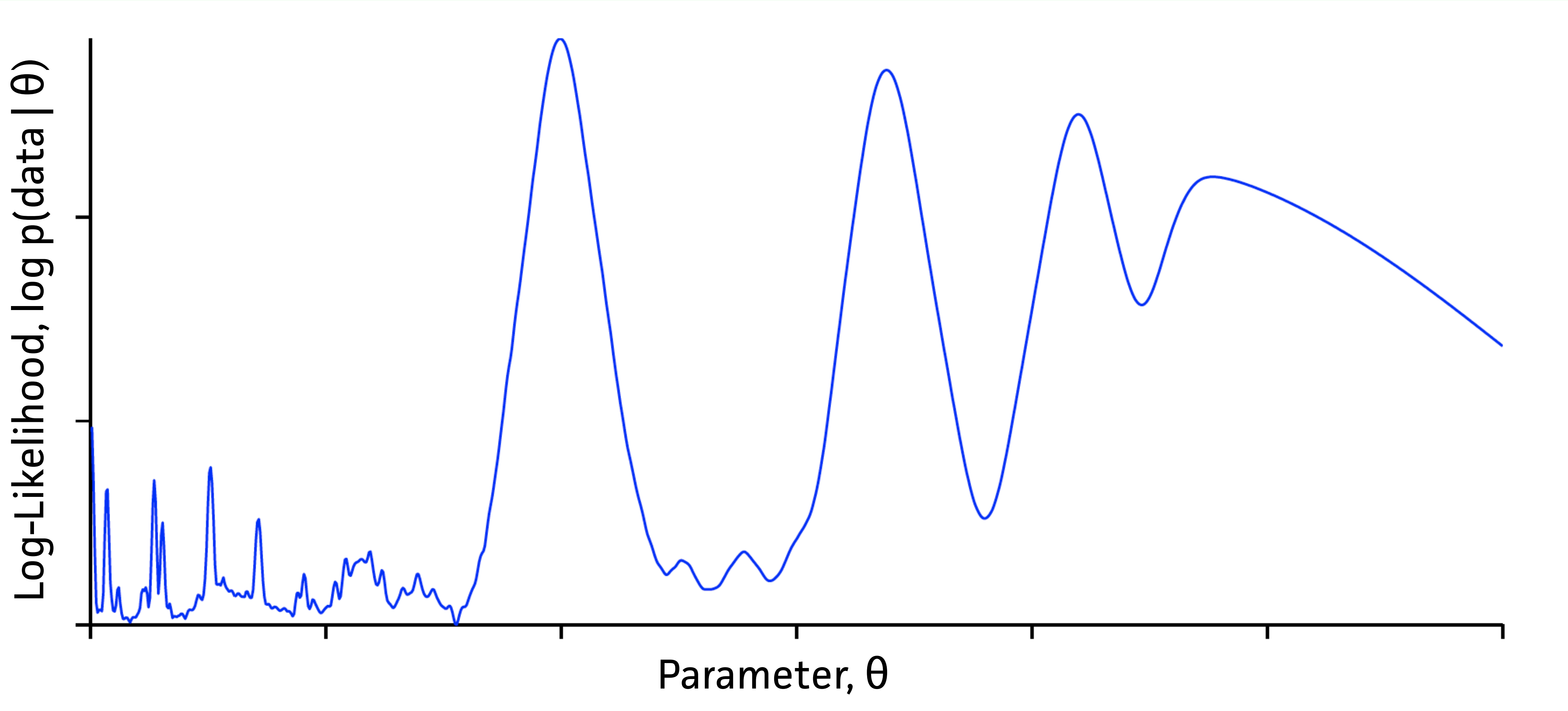
10^1
10^{-2}
10^{-5}
10^{-8}
10^{-11}
10^{-14}
$|F - \hat{F}|$
Monte Carlo
Trapezoidal
Gauss-Legendre
10^0
10^1
10^2
samples

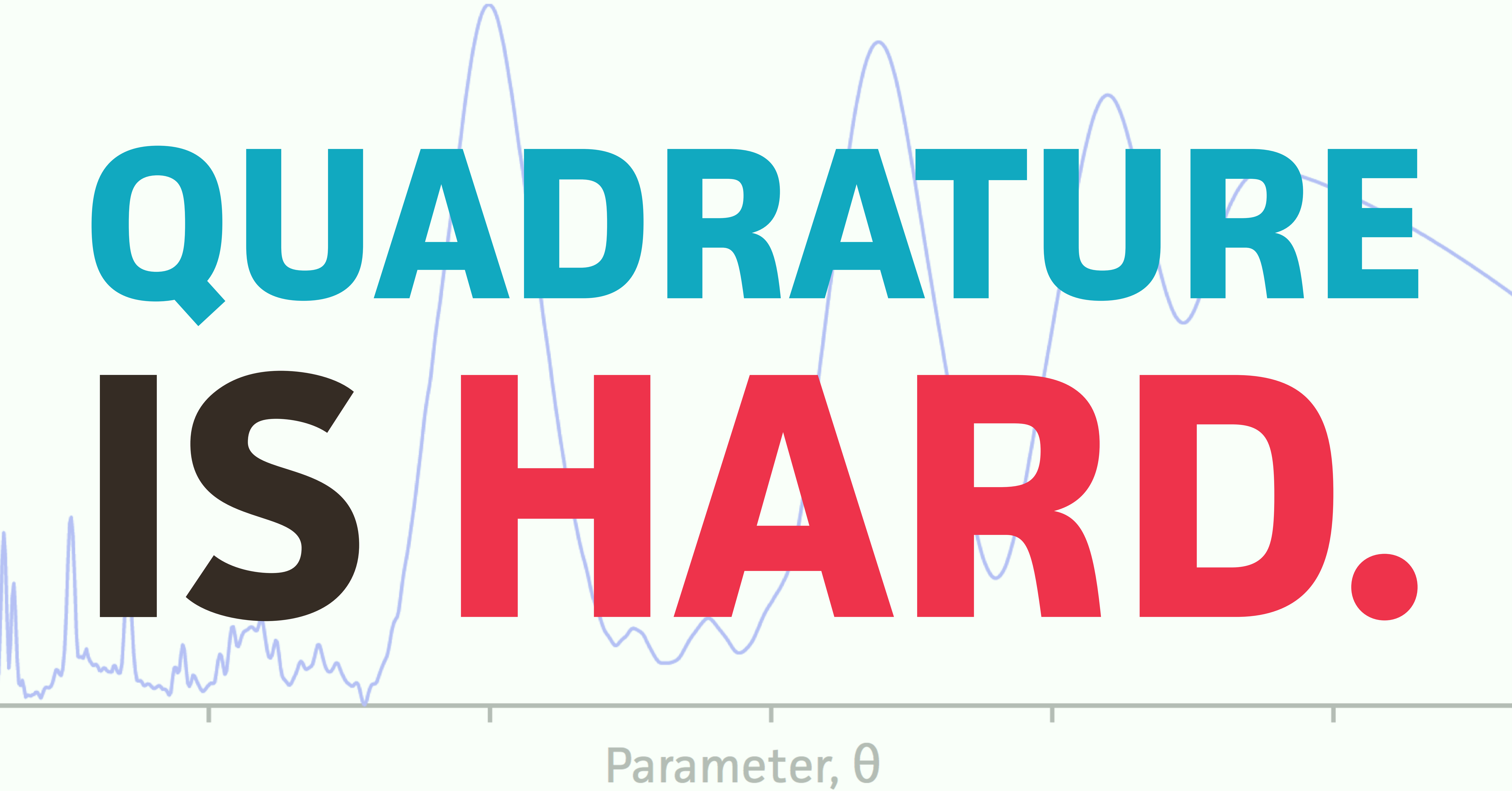
Quadrature is often required to manage model parameters.



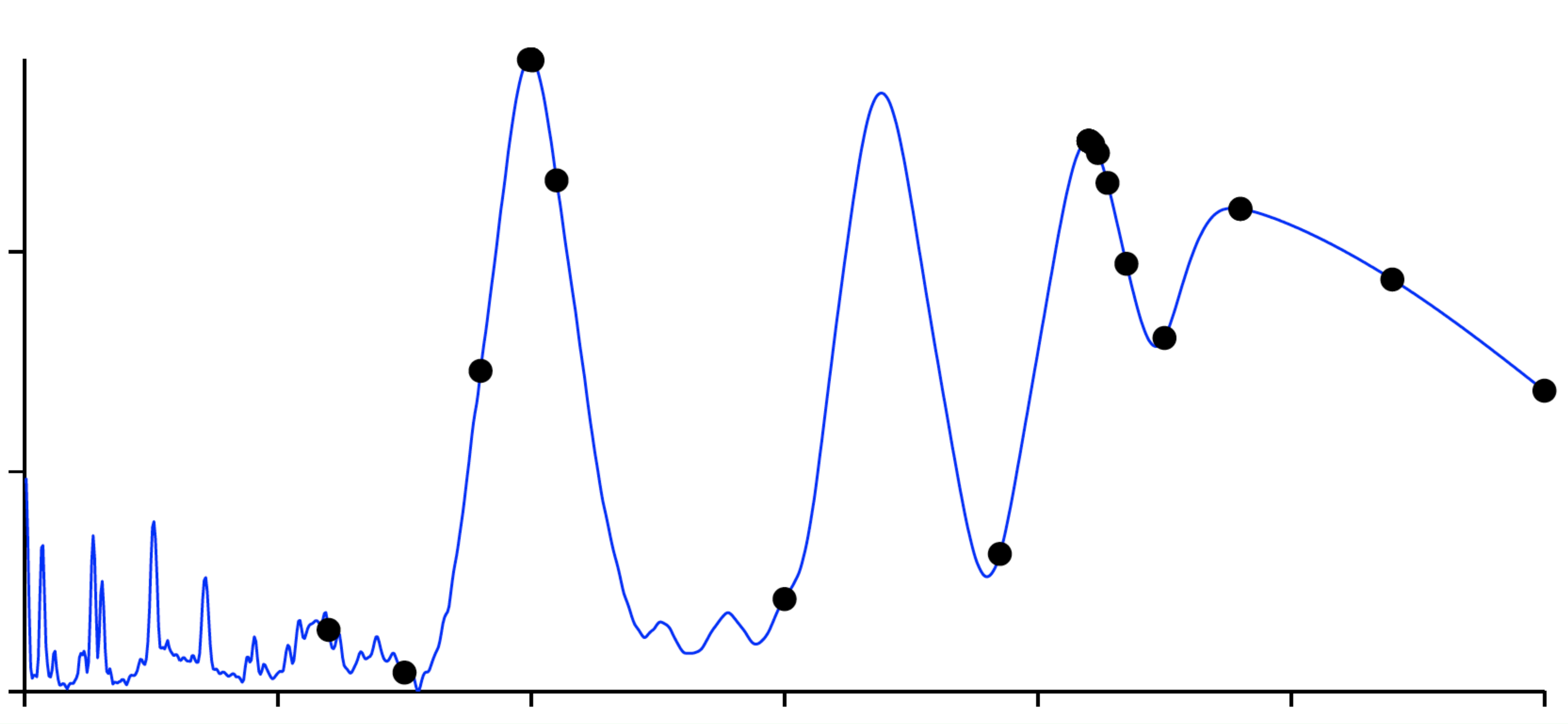
Managing parameters θ requires the **model evidence,**

$$p(\text{data}) = \int p(\text{data} \mid \theta)\, p(\theta) \mathrm{d}\theta.$$

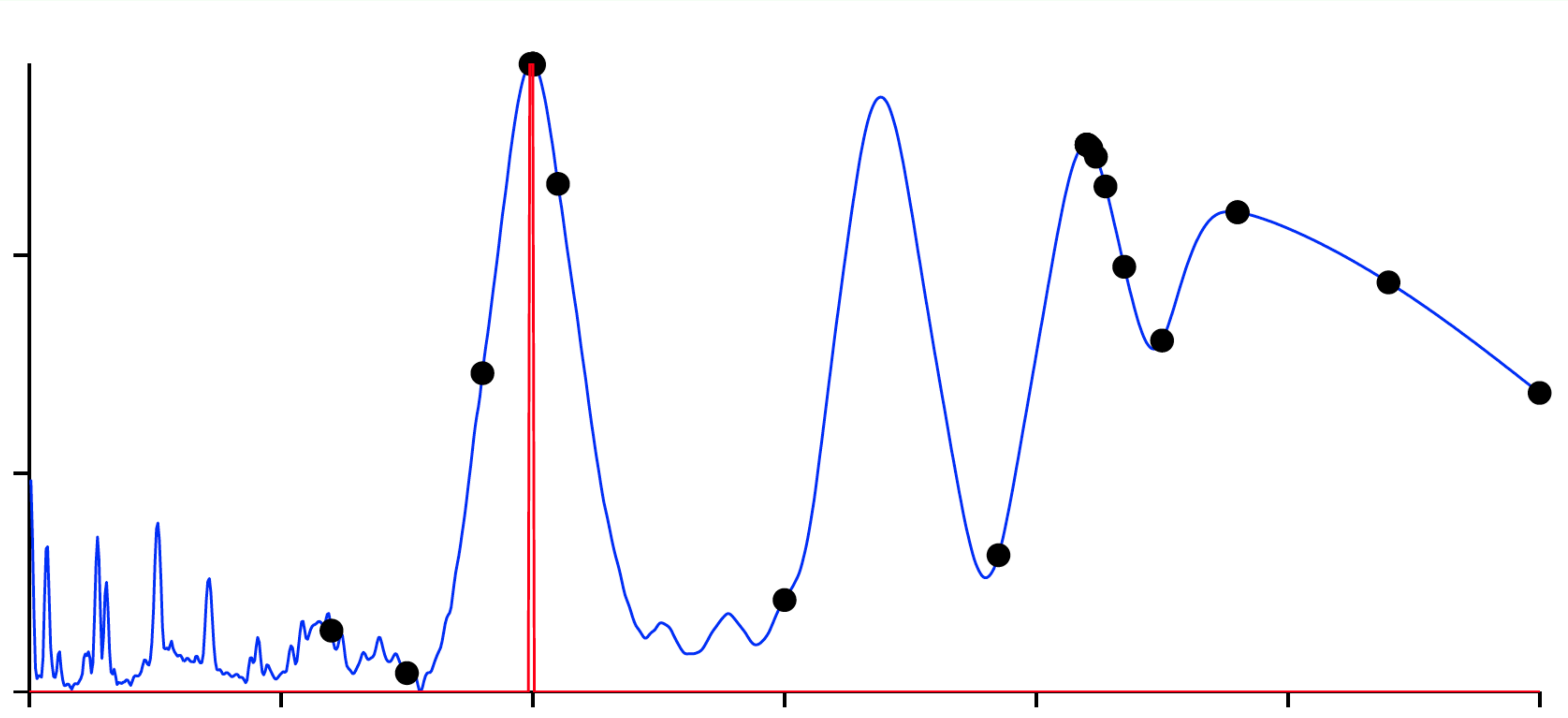



QUADRATURE
IS HARD.
Parameter, θ

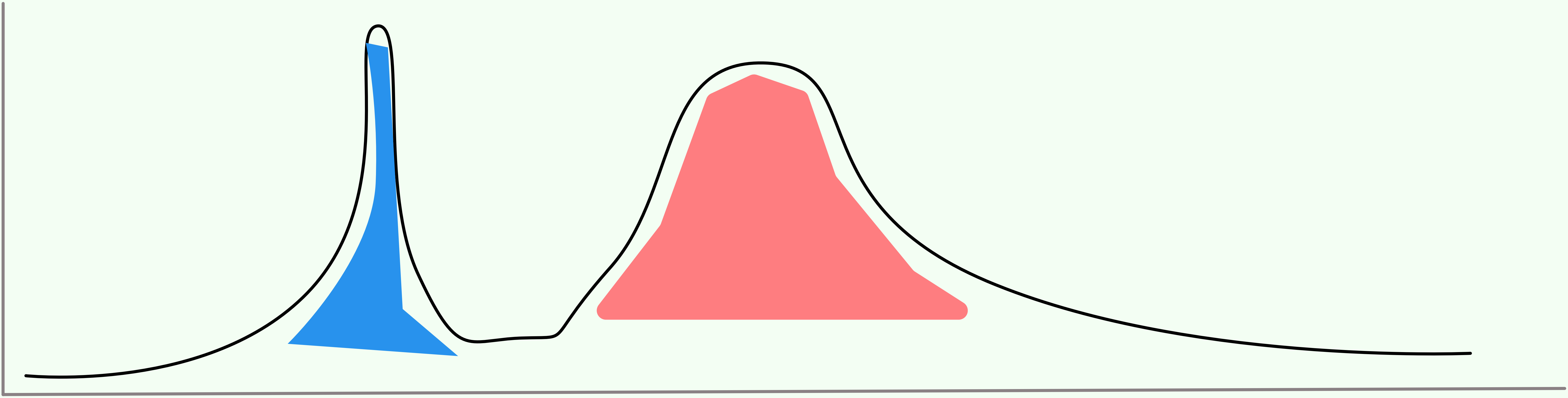
Optimisation (maximum likelihood, training) is often used in the place of quadrature.



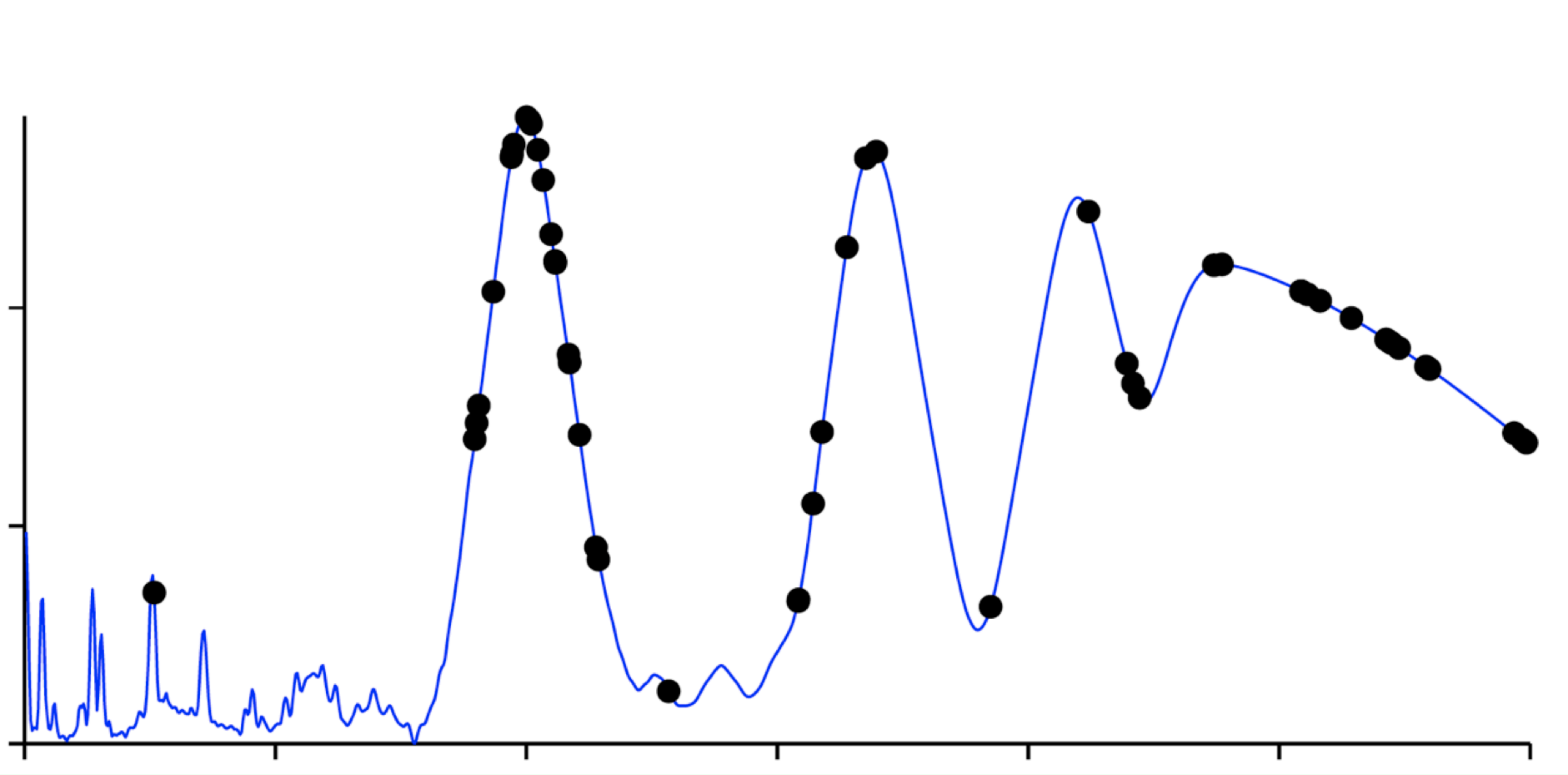
This approximates as $p(\mathrm{data}) \simeq \int p(\mathrm{data} \mid \theta)\, \delta(\theta - \theta_{\mathrm{max}})\, \mathrm{d}\theta$.



If optimising, **flat optima** are often a better representation of the integral than **narrow optima.**

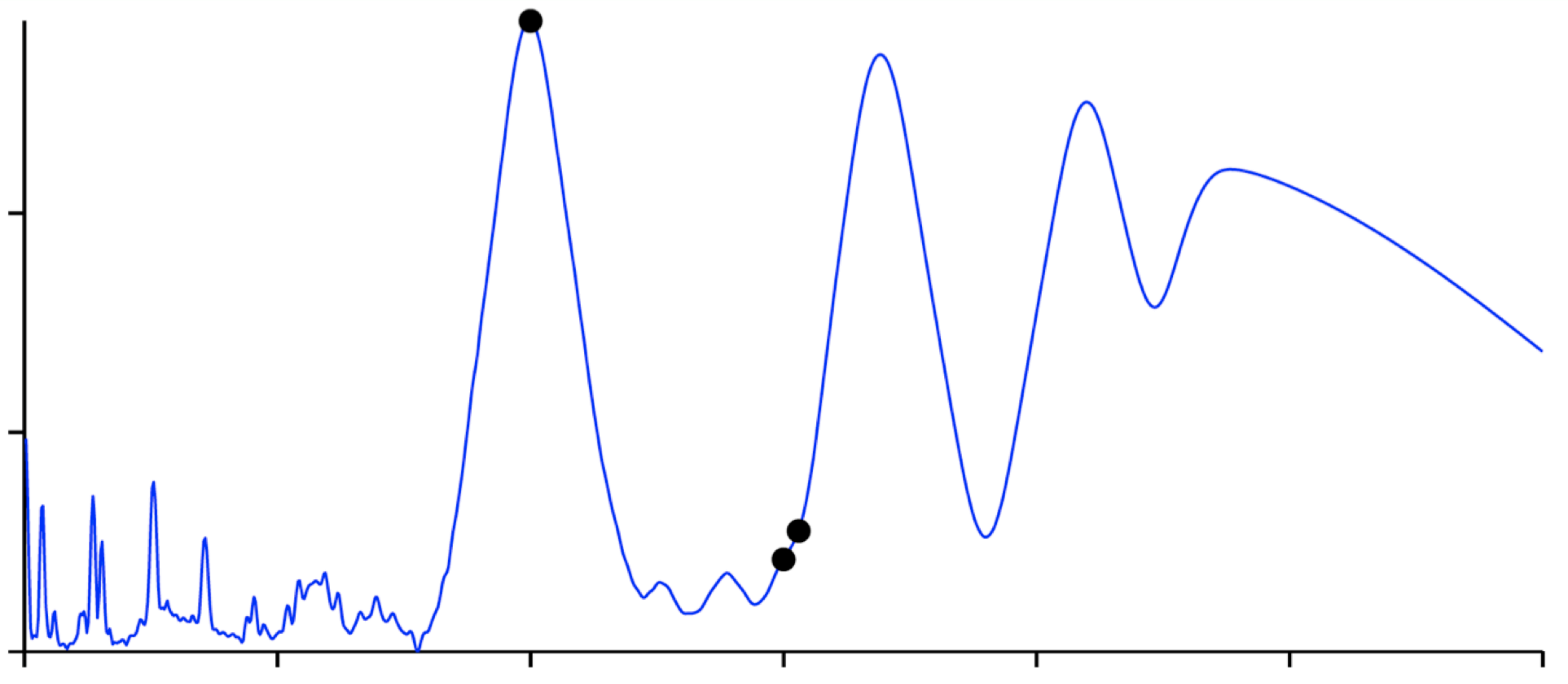


Monte Carlo has revolutionised Bayesian inference.

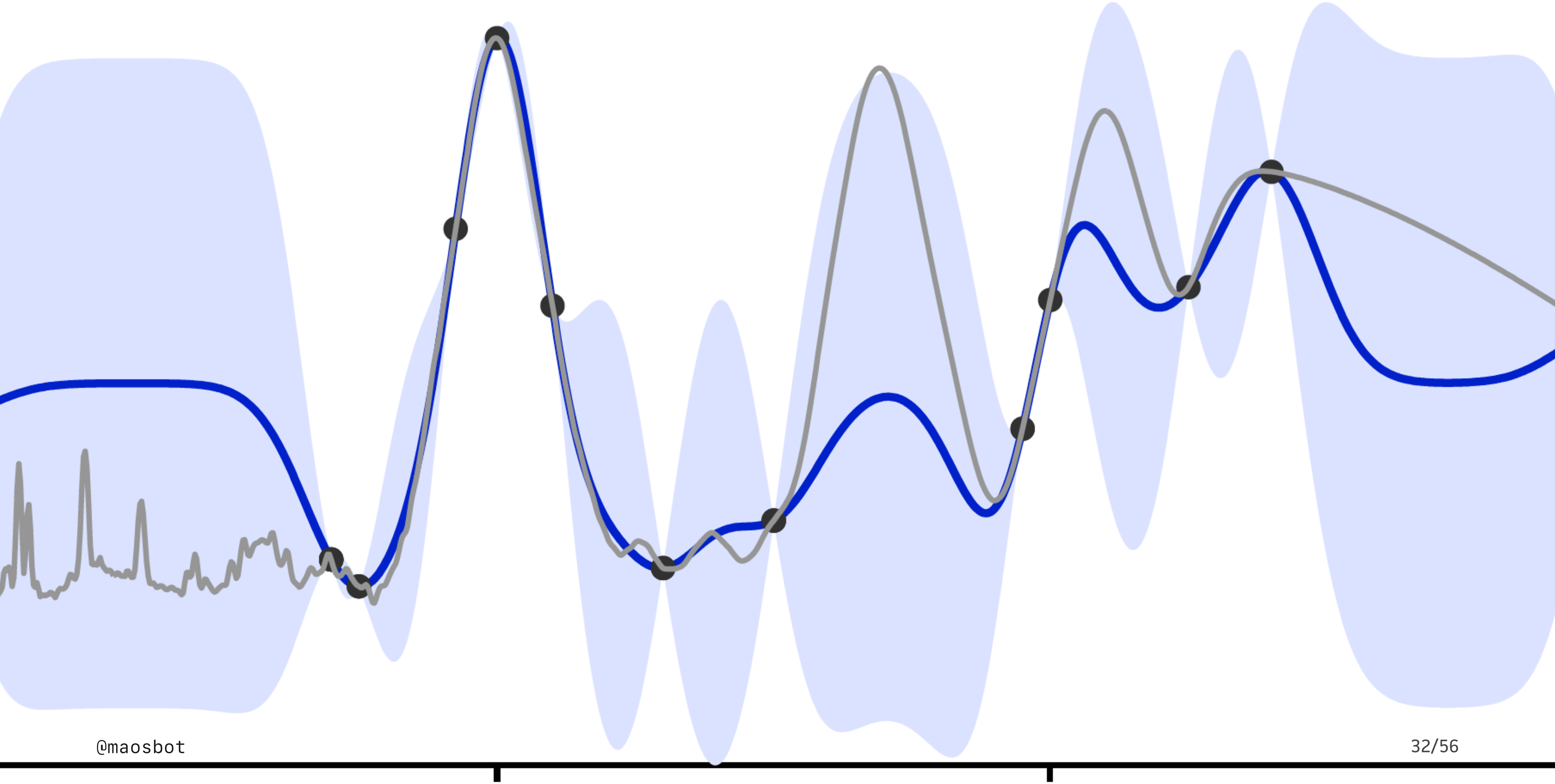


Monte Carlo estimators, $\int f(x)\,p(x)\,\mathrm{d}x \simeq \frac{1}{N}\sum_{i=1}^{N} f(x_i)$,

ignore relevant information.



O'Hagan, A. (1987). Monte Carlo is Fundamentally Unsound. Journal of the Royal Statistical Society. Series D (The Statistician).



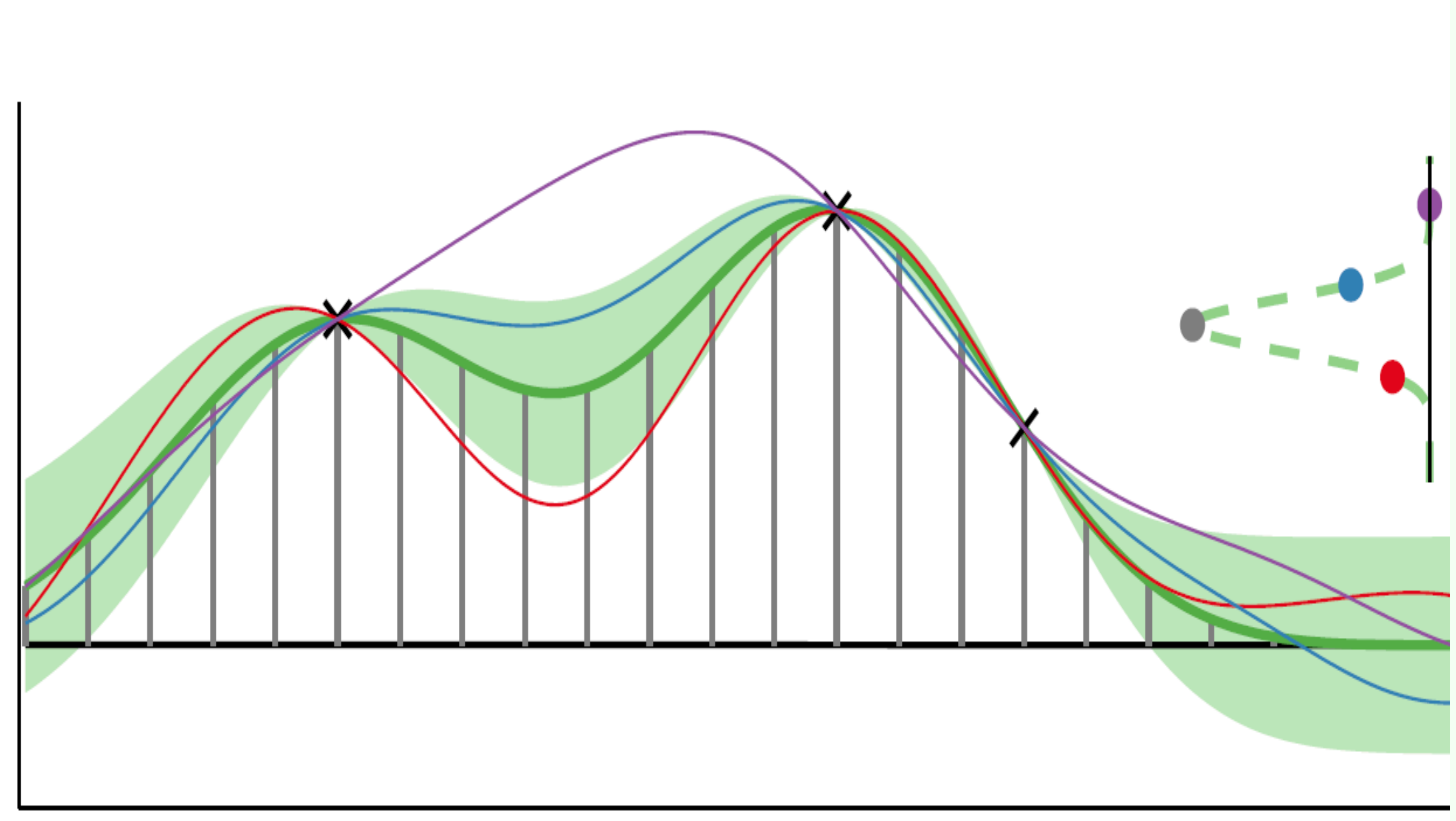
```
ea = params[]
wa = params[2:3]
secw, sesw = np.sqrt(ea)*np.cos(wa), np.sqrt(ea)*np.sin(wa)
```

We often have relevant prior knowledge: like the problem's source code.

```
bx, by = af.baseline_m(df.time, pv_base, df.px, df.py, nthr)
px, py = af.am_model_em(df.time, np.r_[pv_base, pv_1, pv_2], 2, 1
mx, my = 1e6*(bx+px), 1e6*(by+py)
```

The perfect prior is intractable.

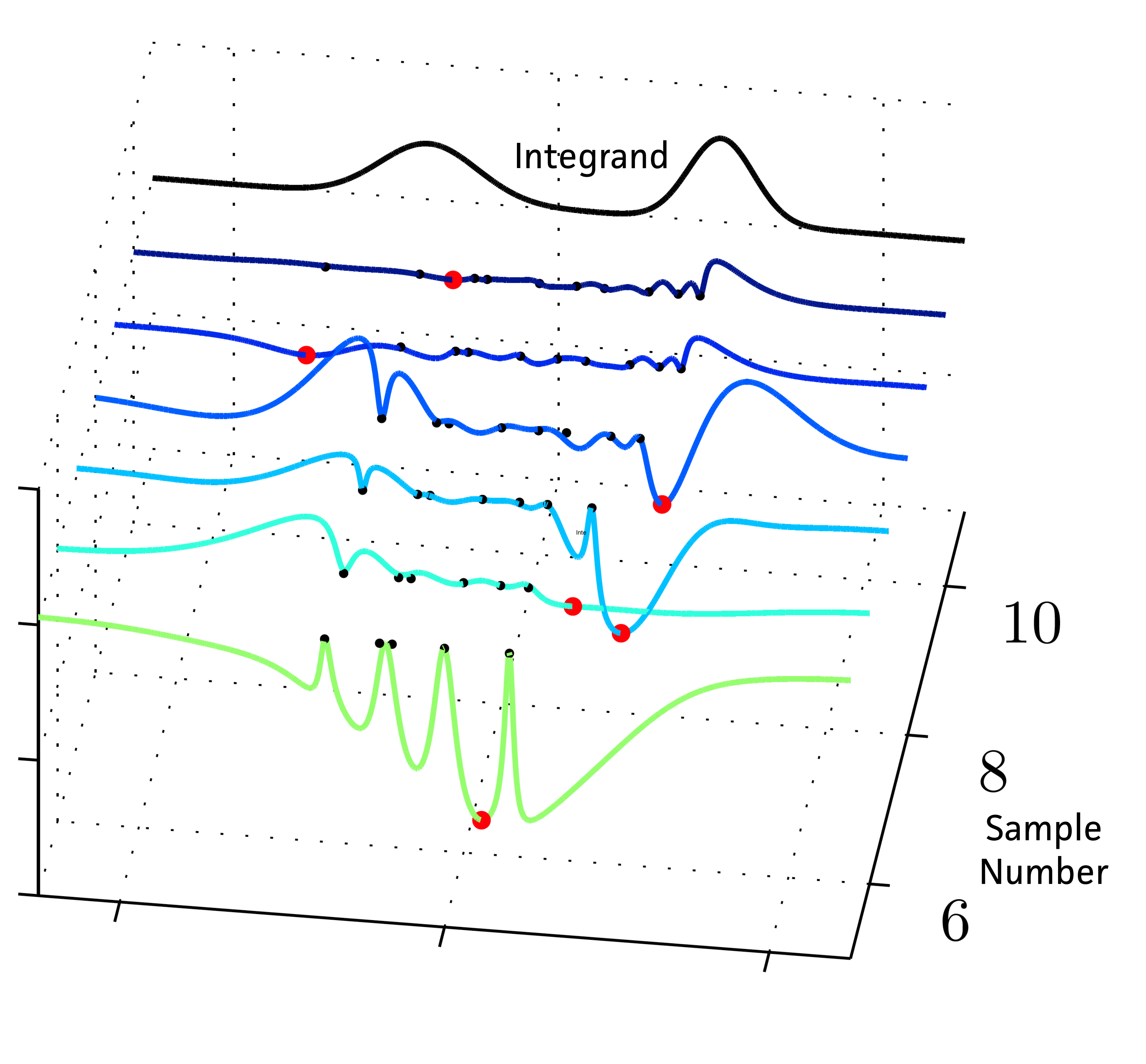
An agent is defined by its prior and loss function.



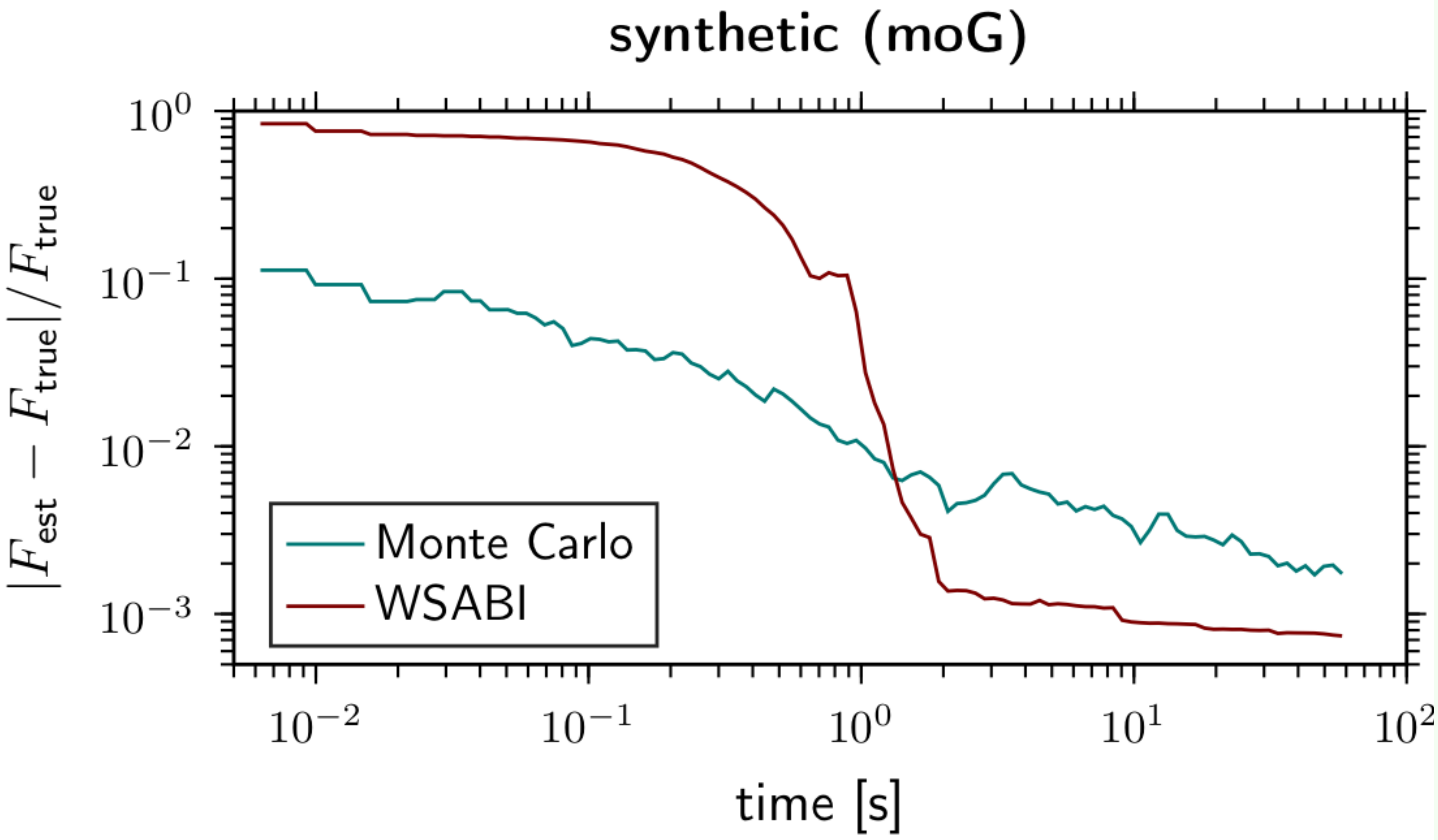
A natural **loss function** for quadrature is the **uncertainty in the integral.**

Source: D. Duvenaud.

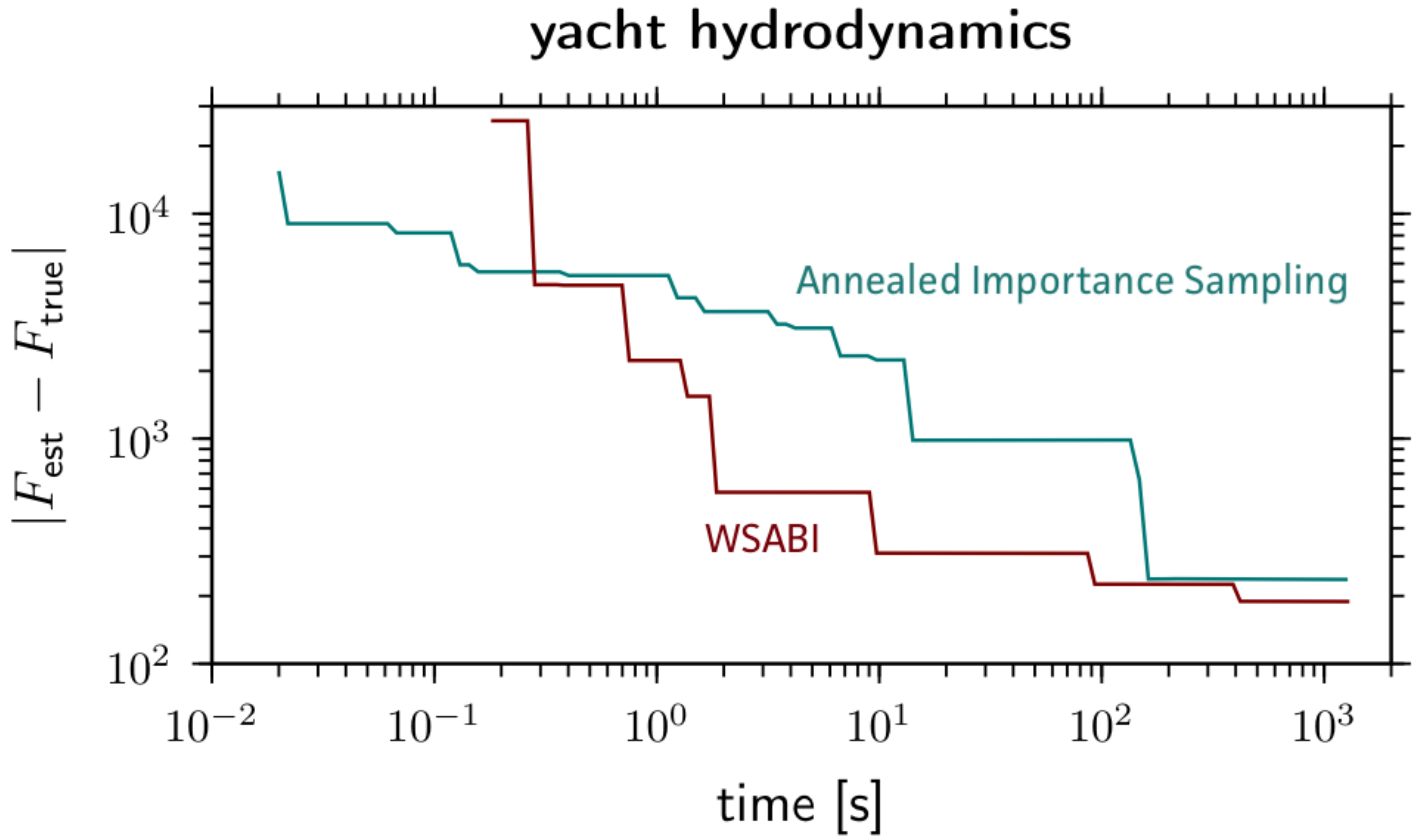
WSABI uses a loss that is the uncertainty in the integrand.



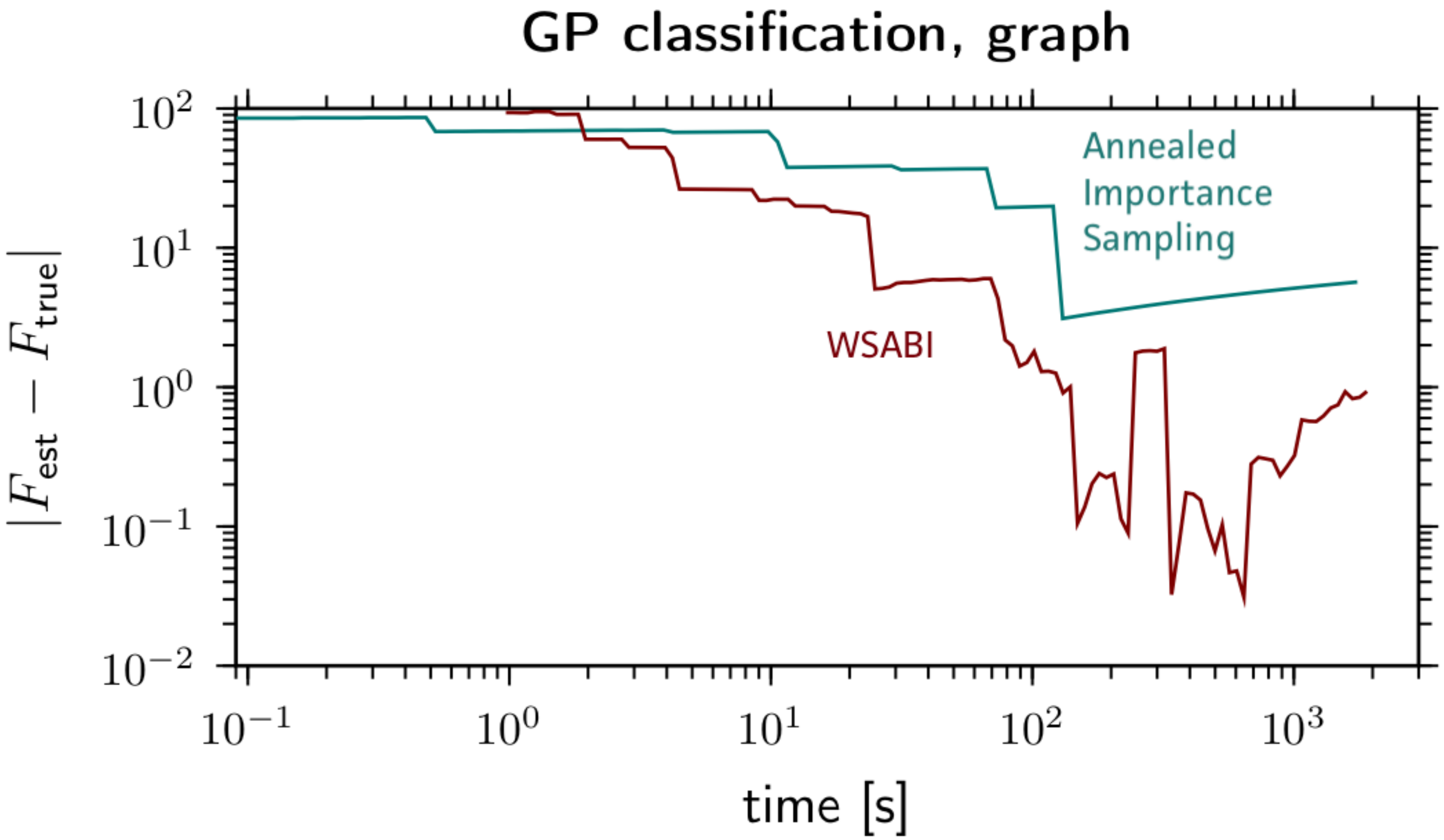
Source: Gunter, Osborne, Garnett, Hennig, & Roberts (2014). Sampling for Inference in Probabilistic Models with Fast Bayesian Quadrature. NIPS.



Source: Gunter, Osborne, Garnett, Hennig, & Roberts (2014). Sampling for Inference in Probabilistic Models with Fast Bayesian Quadrature. NIPS.



Source: Gunter, Osborne, Garnett, Hennig, & Roberts (2014). Sampling for Inference in Probabilistic Models with Fast Bayesian Quadrature. NIPS.



Source: Gunter, Osborne, Garnett, Hennig, & Roberts (2014). Sampling for Inference in Probabilistic Models with Fast Bayesian Quadrature. NIPS.

Overhead can set you free.

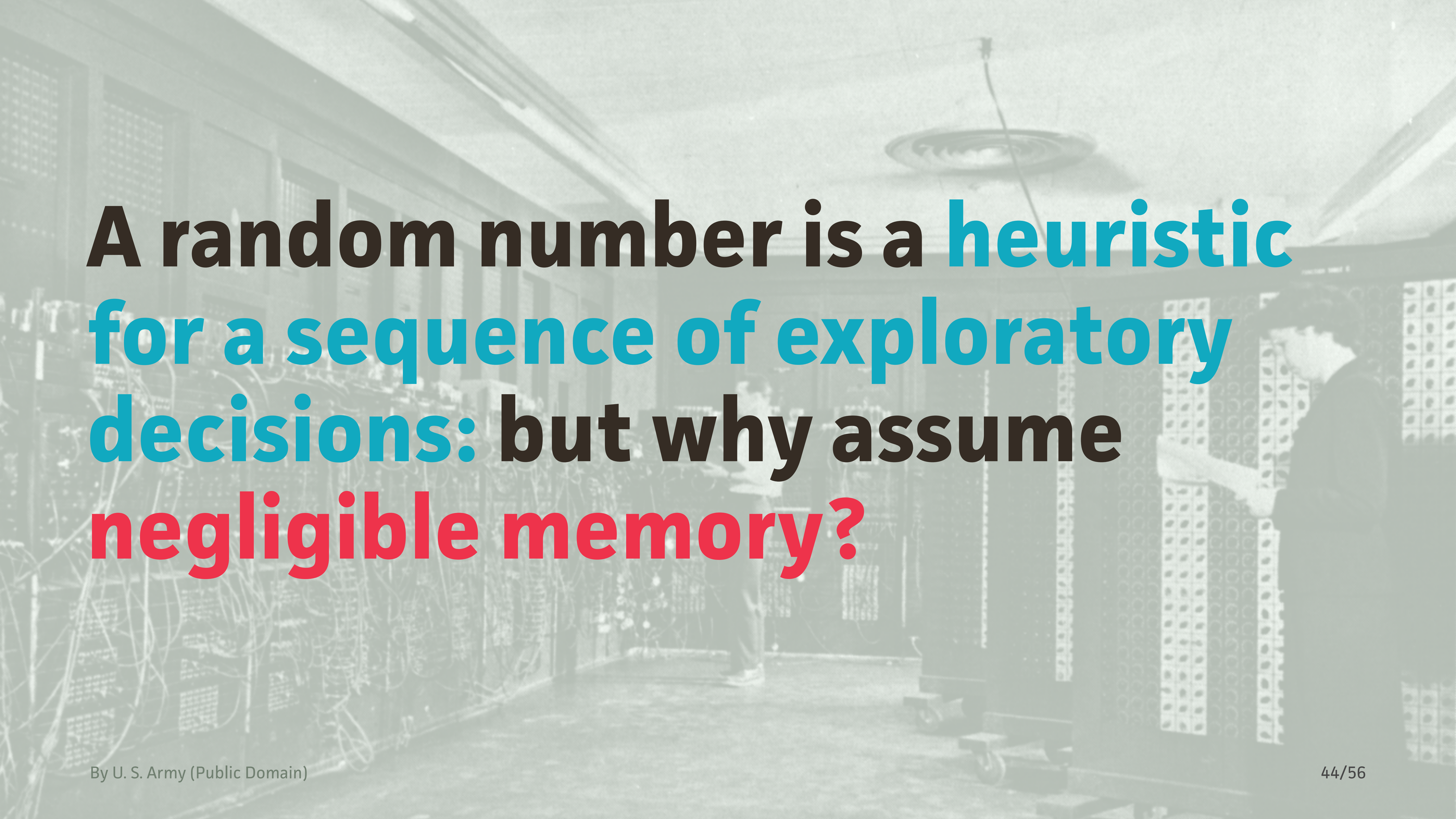
By Paulhaberstroh [CC BY-SA 4.0], from Wikimedia Commons

A RANDOM NUMBER IS A DECISION.

By MauricioEiji , Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 License.

A random number assumes a completely flat expected loss.

A random number is a heuristic for a sequence of exploratory decisions: but why assume negligible memory?



By U. S. Army (Public Domain)

Using random numbers makes your algorithm **unimprovable.**

Source: By Jake Archibald from London, England - Sebastian Vettel - Ferrari - Halo, CC BY 2.0. References: Henderson et al. “Deep Reinforcement Learning that Matters” (2017); Islam et al. “Reproducibility of Benchmarked Deep Reinforcement Learning Tasks for Continuous Control” (2017); Colas, Sigaud, and Oudeyer. “How Many Random Seeds? Statistical Power Analysis in Deep Reinforcement Learning Experiments” (2018); Mania, Guy, and Recht. “Simple random search provides a competitive approach to reinforcement learning” (2018).

A random number may be intended to be **unbiased**, but there are **no adversaries in numerics.**

Source: Brown, Mané, Roy, Abadi & Gilmer (2017) "Adversarial Patch".

Quiz: which of these sequences is random?

1. 6224441111111114444443333333
2. 1693993751058209749445923078
3. 7129042634726105902083360448
4. 1000111111011111111001010000

Quiz: which of these sequences is **random?**

1. 6224441111111114444443333333: seven d6 rolls with i repeats of the ith roll.
2. 1693993751058209749445923078: the 41st to 70th digits of π.
3. 7129042634726105902083360448: this sequence was generated by the von Neumann method with seed 908344.
4. 1000111111011111111001010000: digits taken from a CD-ROM published by George Marsaglia.

A finite string of random numbers is **encoding some bias!**

Recall:

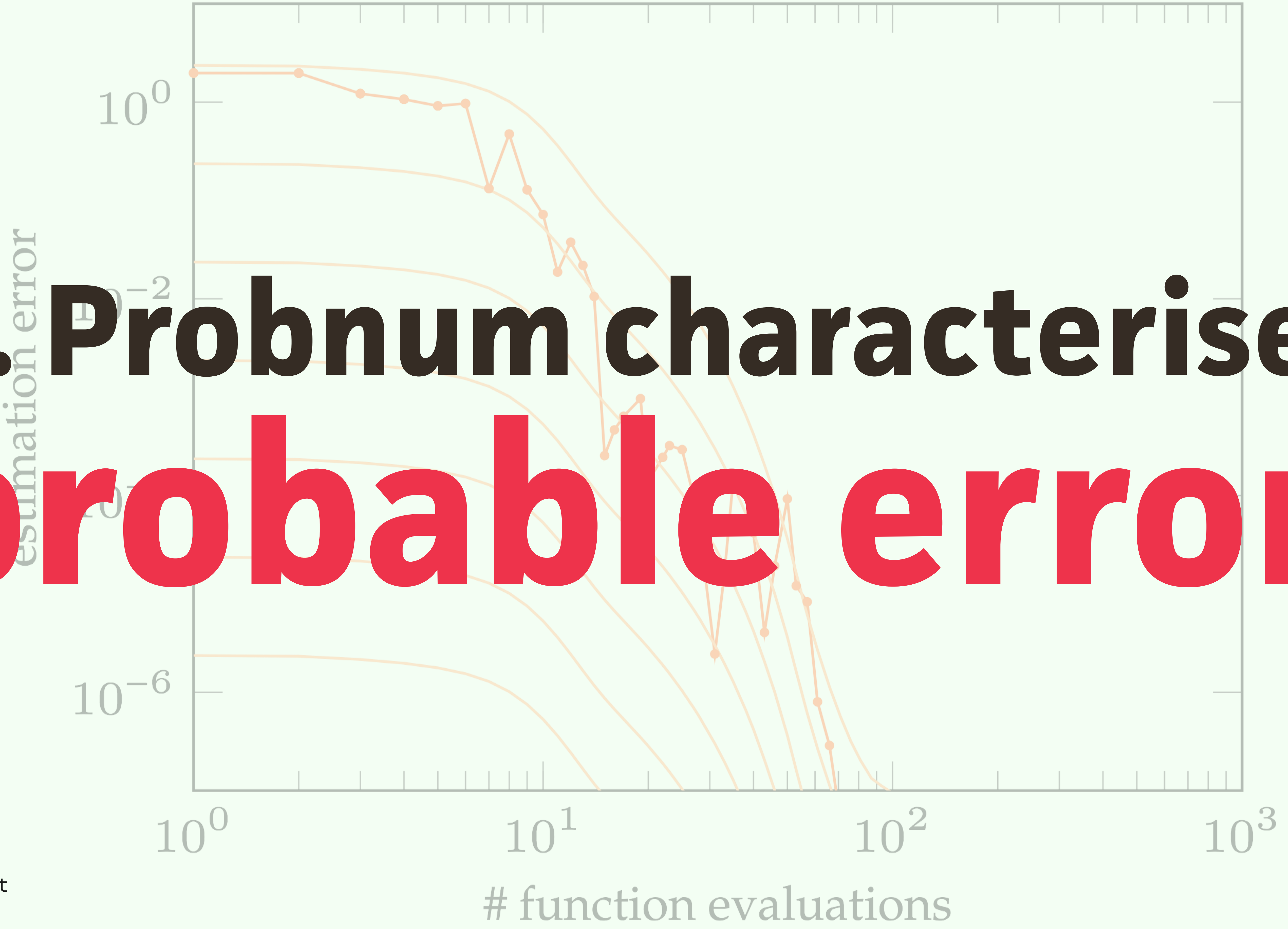
1. numeric error is significant;

2. numeric methods are generic;

3. our numerics problems tax our computation.

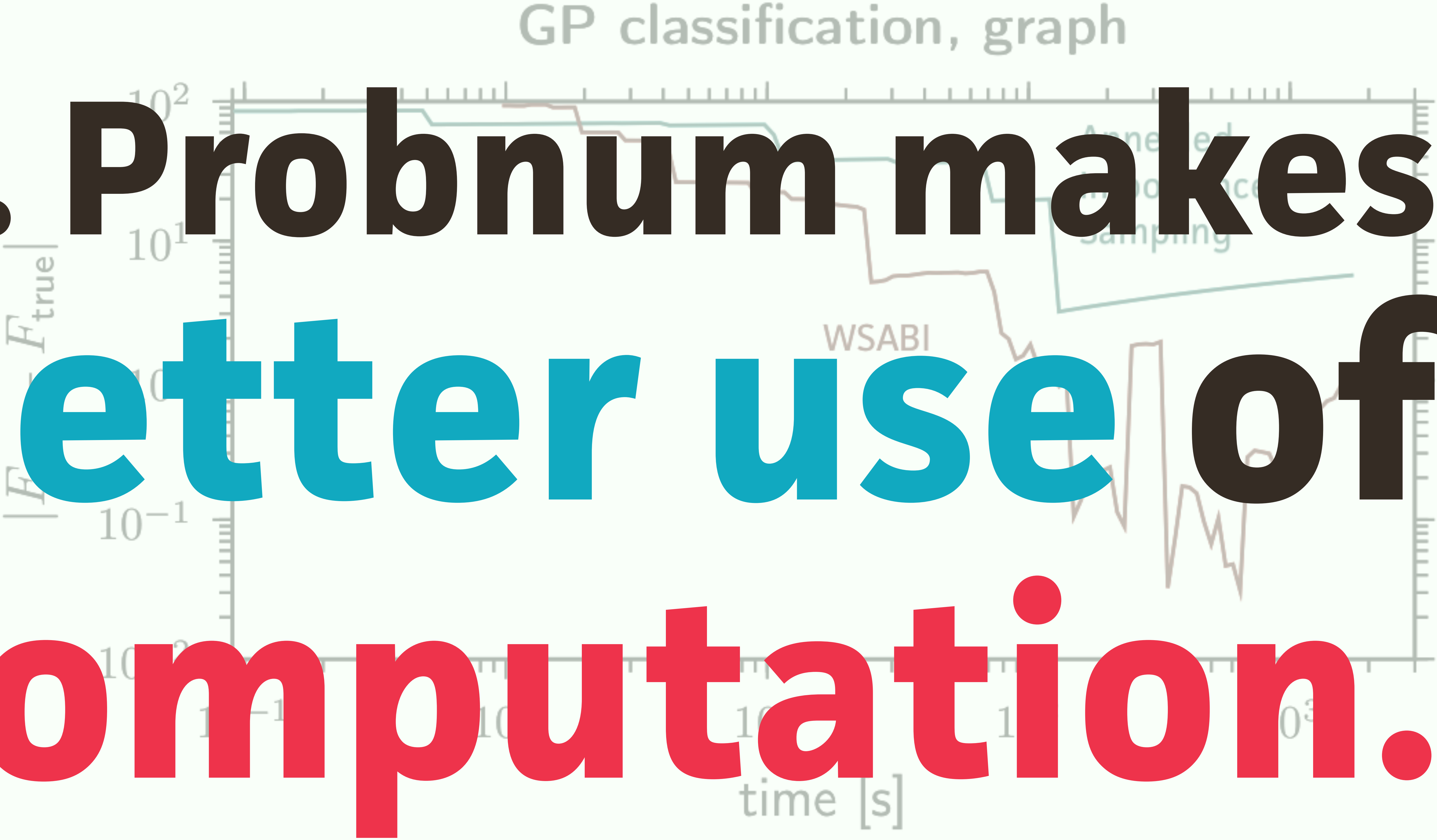
By ESA/Hubble, CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=8788068.

1. Probnum characterises **probable error.**



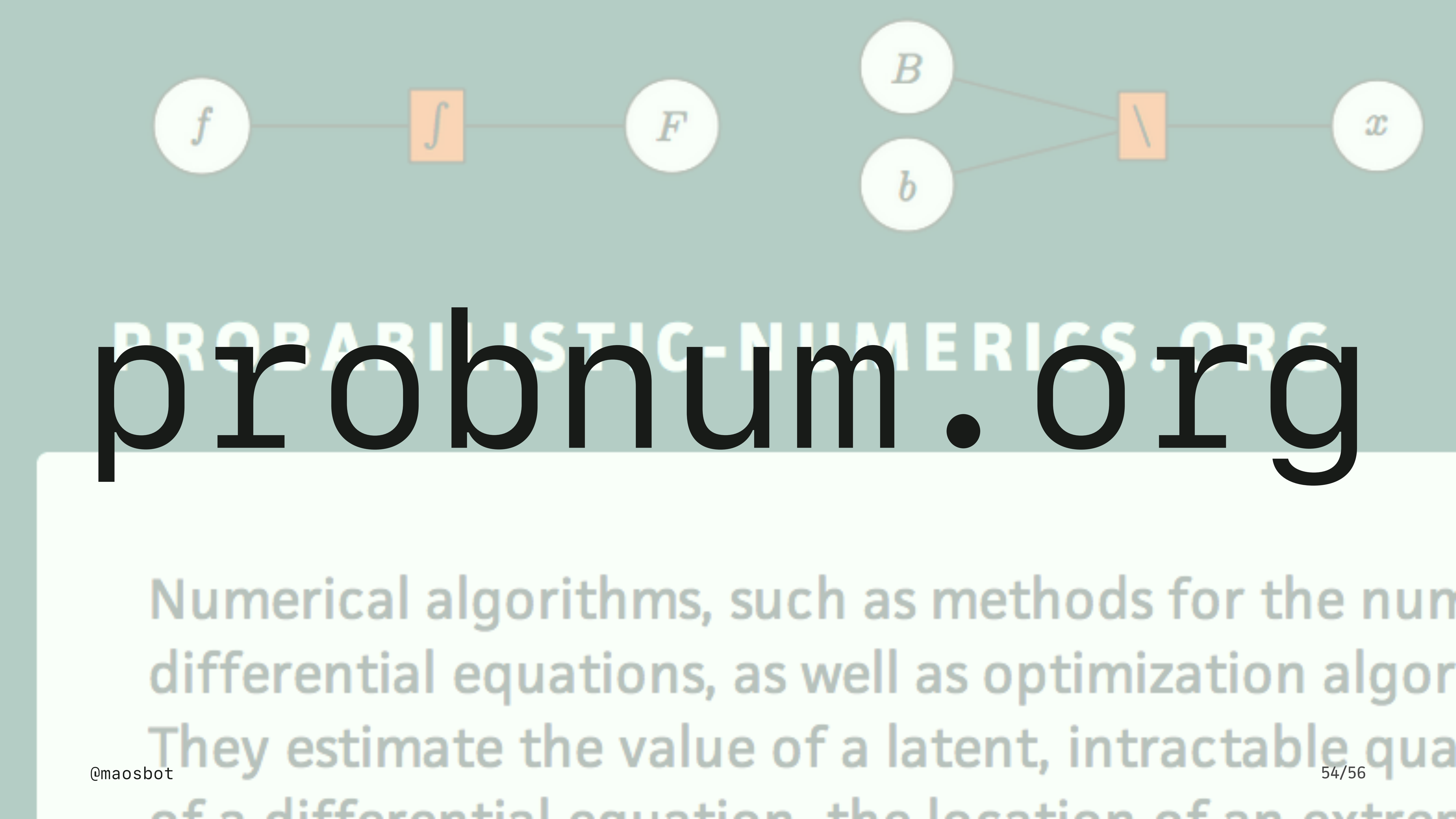
2. Probnum tailors procedures to problems.

3. Probnum makes better use of computation.



PROBABILISTIC-NUMERICS.ORG

Numerical algorithms, such as methods for the num
differential equations, as well as optimization algor
They estimate the value of a latent, intractable qua


f
F
B
b
x
PROBABILISTIC-NUMERICS.ORG
Numerical algorithms, such as methods for the num
differential equations, as well as optimization algor
They estimate the value of a latent, intractable qua

probnum.org

PROBABILISTIC-NUMERICS.ORG

LITERATURE

This page collects literature on all areas of probab
not hesitate to contact us. The fastest way to get
file in /_bibliography, then either send us a pull-re

QUICK-JUMP LINKS:

- General and Foundational
- Quadrature
- Linear Algebra
- Optimization
- Ordinary Differential Equations
- Partial Differential Equations

Huge thanks to **Philipp Hennig.**