NeurIPS | 2019

The Randomized Midpoint Method for Log-Concave Sampling

Ruoqi Shen
University of Washington

Yin Tat Lee
University of Washington
Microsoft Research

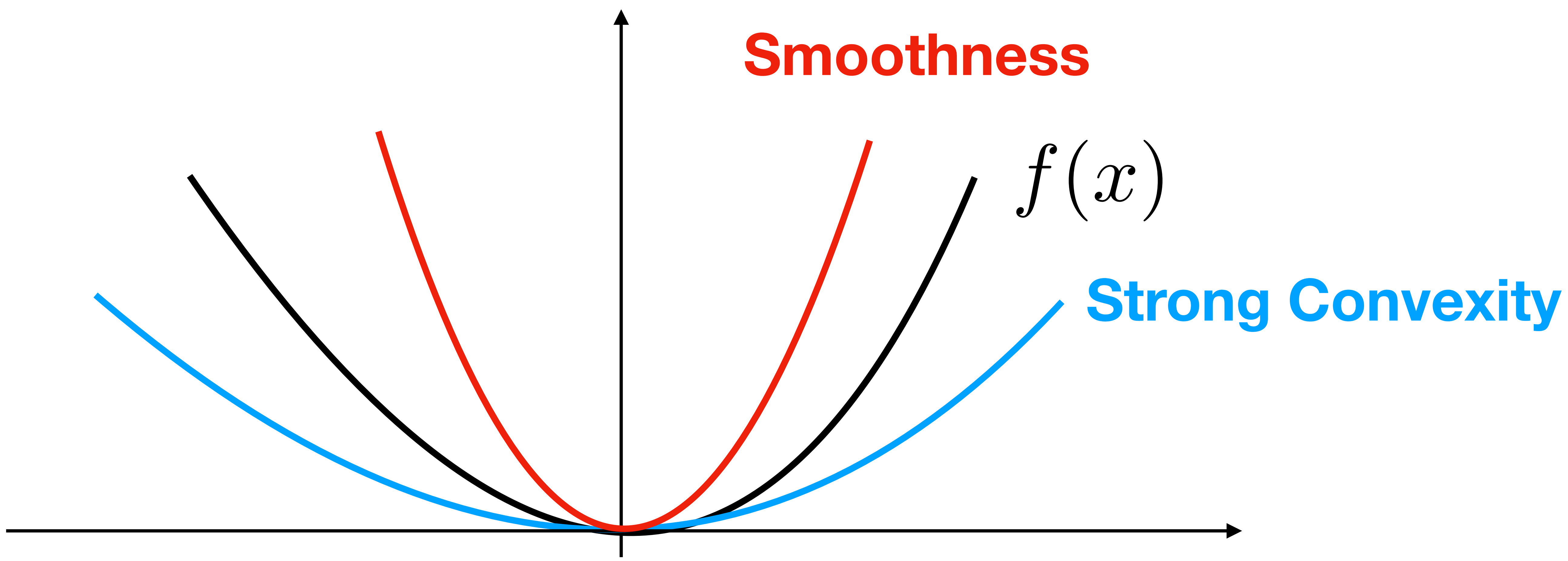
Thirty-third Conference on Neural Information Processing Systems

Problem

Sample from the target distribution $\Pi(x) \sim \exp(-f(x))$

Assume we have access to gradient oracle $\nabla f(x)$

Assume f **is strongly convex and smooth** $m \preceq \nabla^2 f(x) \preceq L\,, \quad \kappa = \frac{L}{m}$



Iteration Complexity

Algorithm	# Steps
Hit-and-Run [Lovász, Vempala, 2006]	$\tilde{O}\left(d^3 \log \frac{1}{\epsilon}\right)$
Langevin Diffusion [Durmus, Moulines, 2016]	$\tilde{O}\left(\kappa^2/\epsilon^2\right)$
Underdamped Langevin Diffusion [Cheng et al., 2017]	$\tilde{O}\left(\kappa^{1.5}/\epsilon + \kappa^2\right)$
Hamiltonian Monte Carlo [Chen, Vempala, 2019]	$\tilde{O}\left(\kappa^{1.5}/\epsilon\right)$
ULD with Randmized Midpoint Method **[Our paper]**	$\tilde{O}\left(\kappa^{7/6}/\epsilon^{1/3} + \kappa/\epsilon^{2/3}\right)$

Random Process

Underdamped Langevin Diffusion (ULD)

$$dv(t) = -\gamma v(t)dt - u\nabla f(x(t))dt + \sqrt{2\gamma u}dB_t$$
$$dx(t) = v(t)dt$$

The above process has stationary distribution

$$(x, v) \sim \exp\left(-f(x) - \frac{\|v\|^2}{2u}\right)$$

So one approach is to simulate ULD until it converges.

Simulate Random Process

For step size h, one step of the random process is given by

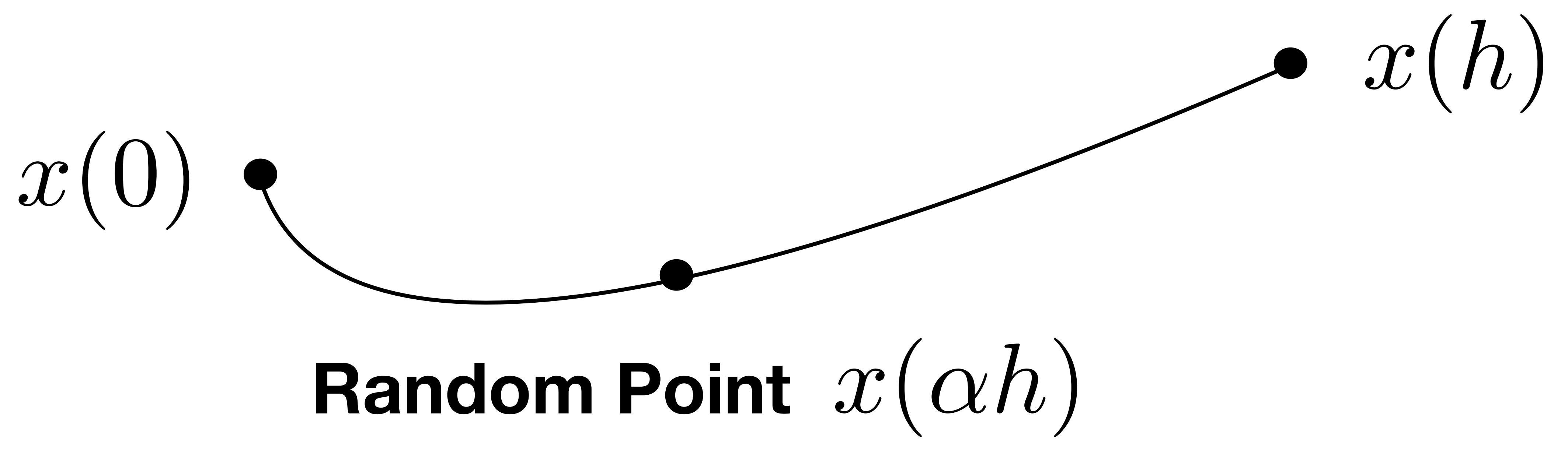
$$x(h) = x(0) + \frac{1-e^{-2h}}{2}v(0) - \frac{u}{2}\underline{\int_0^h \left(1-e^{-2(h-t)}\right)\nabla f(x(t))dt} + \sqrt{u}\int_0^h \left(1-e^{-2(h-t)}\right)dB_t$$

$$v(h) = v(0)e^{-2h} - \underline{u\left(\int_0^h e^{-2(h-t)}\nabla f(x(t))dt\right)} + 2\sqrt{u}\int_0^h e^{-2(h-t)}dB_t$$

Integrals are hard to approximate!

We use the Randomized Midpoint Method to approximate the integrals.

Randomized Midpoint Method

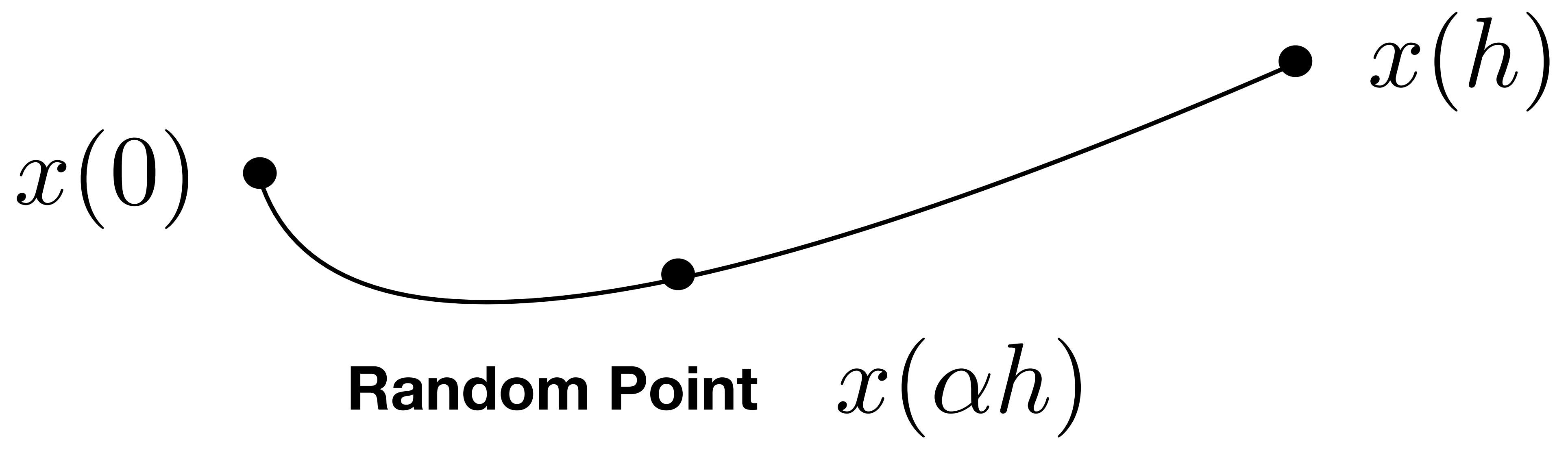


Goal: Approximate $x(h)$

As an example, we show how to approximate $x(h) = \int_0^h \nabla f(x(t))dt$ **,**

Step 1: Choose a random point $\alpha \sim \mathrm{Unif}[0, 1]$

Randomized Midpoint Method



Step 2: Approximate $x(\alpha h)$

$$x(\alpha h) = \int_0^{\alpha h} \nabla f(x(t))dt \approx \hat{x}(\alpha h) = \alpha h \boldsymbol{\nabla f(x(0))}$$

Step 3: Approximate $x(h)$ **using the approximation of** $x(\alpha h)$ **from step 2**

$$x(h) = \int_0^{h} \nabla f(x(t))dt \approx h\nabla \boldsymbol{f(\hat{x}(\alpha h))}$$

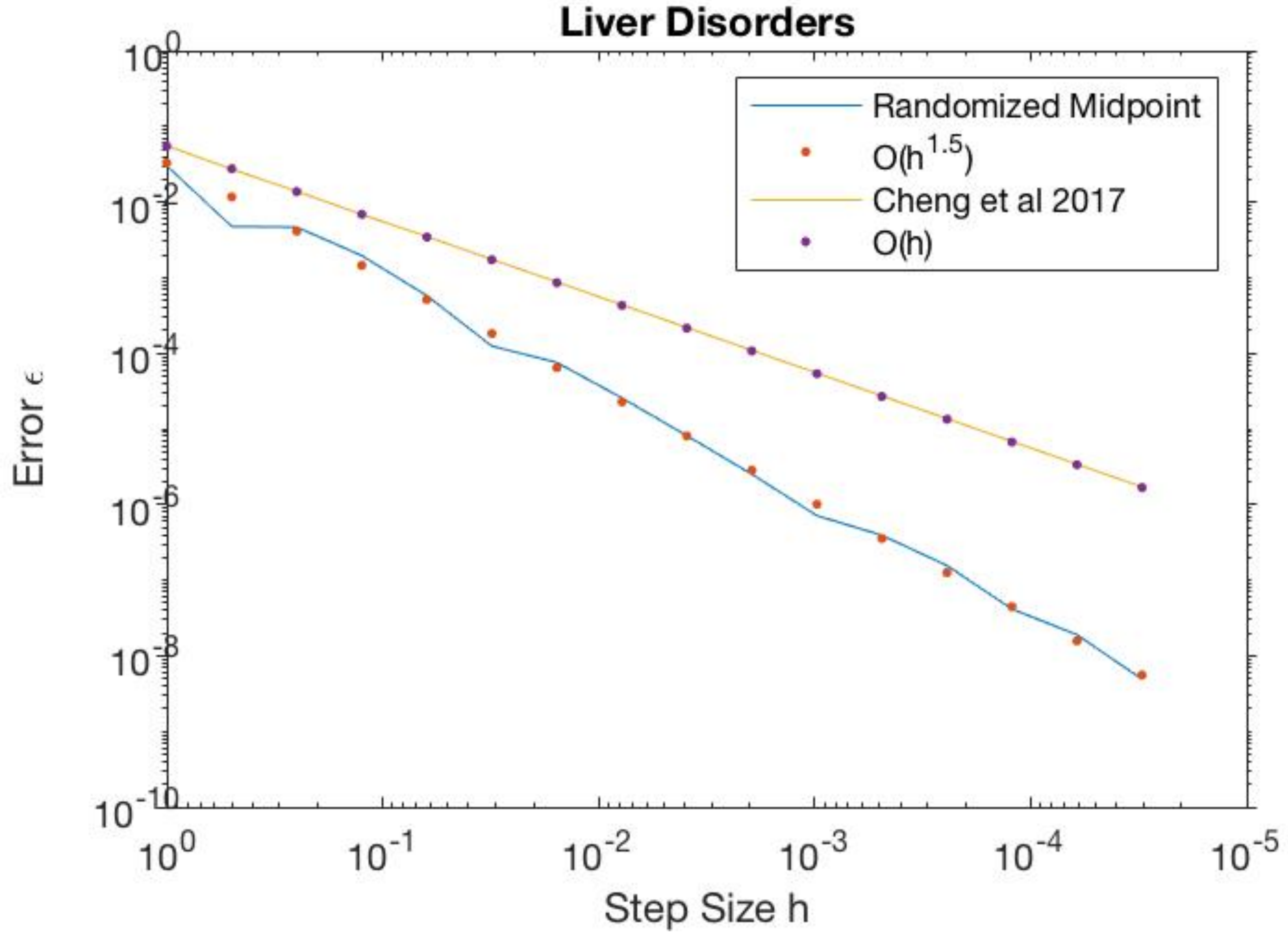
We use randomness to solve SDE/ODE more accurately.
Our estimator has very small bias.

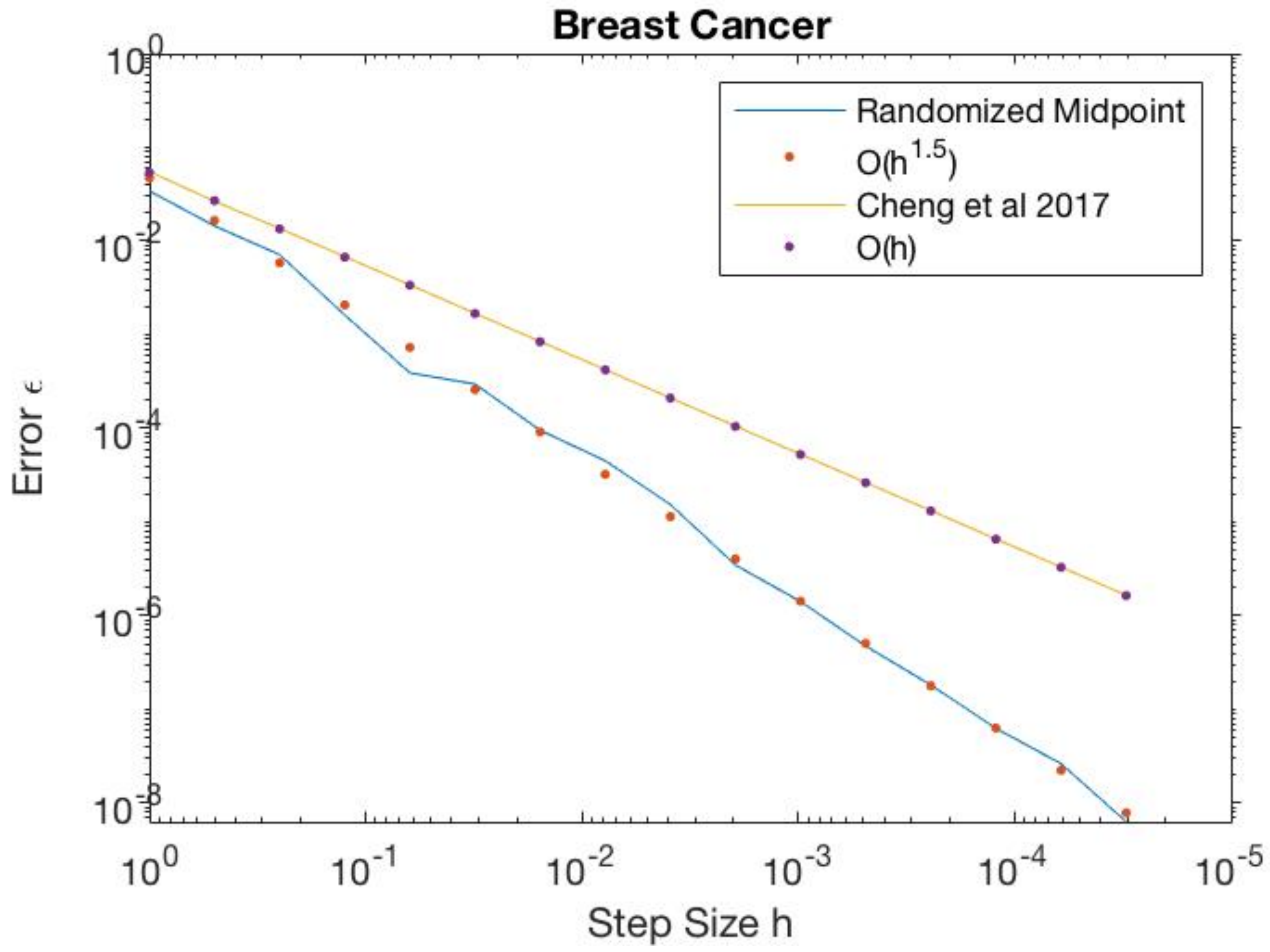
$h\nabla f(x(\alpha h))$ **is an unbiased estimator of** $\int_0^h \nabla f(x(t))dt$

Numerical Experiment

We observe a set of independent samples $\{x_i, y_i\}_{i=1}^m$

Sample from the target distribution $p^*(\theta) \propto \exp(-f(\theta))$ **,** $f(\theta) = \frac{\lambda}{2}\|\theta\|^2 + \frac{1}{m}\sum_{i=1}^{m} \log(\exp(-y_i x_i^T \theta) + 1)$





Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

Thank you!

Check our Poster #163

Ruoqi Shen
University of Washington
shenr3@cs.washington.edu