

CUNEF February 12

# Preconditioning: A General Introduction with a Focus on Sampling

Max Hird (University of Waterloo)

Joint work with Sam Livingstone (UCL)

<https://jmlr.org/papers/v26/23-1633.html>



# Part I: Conditioning

# Conditioning

20th C Maths starts being concerned with *computability* and not simply *conceivability*:

$$\begin{array}{l}
 e_1 \quad 1.4x + 0.9y = 2.7 \\
 e_2 \quad -0.8x + 1.7y = -1.2
 \end{array}
 \left. \vphantom{\begin{array}{l} e_1 \\ e_2 \end{array}} \right\} \iff
 \begin{array}{l}
 0.01 \times e_1 + e_2 \quad -0.786x + 1.709y = -1.173 \\
 e_2 \quad -0.800x + 1.700y = -1.200
 \end{array}
 \left. \vphantom{\begin{array}{l} 0.01 \times e_1 + e_2 \\ e_2 \end{array}} \right\}$$

well-conditioned
ill-conditioned

Turing coins the *condition number* and defines it in multiple ways:

- N-condition number:  $\|A\|_F \|A^{-1}\|_F$  where  $\|A\|_F := \sqrt{\text{Tr}(A^*A)}$
- M-condition number:  $M(A)M(A^{-1})$  where  $M(A) := \max_{ij} |m_{ij}|$

The condition number  $\geq 1$ , and 1 is the best possible value



# Conditioning

Nowadays the problem of matrix inversion has the condition number  $\kappa(Ax = b) = \|A\|_{\text{op}}\|A^{-1}\|_{\text{op}}$

It is the worst error in the output given a noisy input: say we observe  $b + \delta b$  instead of  $b$

Relative input error:  $\frac{\|b + \delta b - b\|}{\|b\|} = \frac{\|\delta b\|}{\|b\|}$

Relative output error:  $\frac{\|A^{-1}b - A^{-1}(b + \delta b)\|}{\|A^{-1}b\|} = \frac{\|A^{-1}\delta b\|}{\|A^{-1}b\|}$

Worst relative output error relative to the relative input error:

$$\kappa(Ax = b) := \sup_{b, \delta b \neq 0} \left\{ \frac{\|A^{-1}\delta b\|}{\|A^{-1}b\|} / \frac{\|\delta b\|}{\|b\|} \right\}$$

# Conditioning

$$\begin{aligned}\kappa(Ax = b) &:= \sup_{b, \delta b \neq 0} \left\{ \frac{\|A^{-1}\delta b\|}{\|A^{-1}b\|} / \frac{\|\delta b\|}{\|b\|} \right\} \\ &= \sup_{b, \delta b \neq 0} \left\{ \frac{\|A^{-1}\delta b\|}{\|\delta b\|} \frac{\|b\|}{\|A^{-1}b\|} \right\} \\ &= \sup_{b \neq 0} \left\{ \frac{\|b\|}{\|A^{-1}b\|} \right\} \sup_{\delta b \neq 0} \left\{ \frac{\|A^{-1}\delta b\|}{\|\delta b\|} \right\} \\ &= \sup_{c \neq 0} \left\{ \frac{\|Ac\|}{\|c\|} \right\} \|A^{-1}\|_{\text{op}} \\ &= \|A\|_{\text{op}} \|A^{-1}\|_{\text{op}}\end{aligned}$$

# Turing Matrix Example

Recall:  $\kappa(Ax = b) = \|A\|_{\text{op}}\|A^{-1}\|_{\text{op}}$

$$\left. \begin{array}{l} e_1 \quad 1.4x + 0.9y = 2.7 \\ e_2 \quad -0.8x + 1.7y = -1.2 \end{array} \right\} \iff$$

well-conditioned

$$\kappa \approx 1.2$$

$$\left. \begin{array}{l} 0.01 \times e_1 + e_2 \quad -0.786x + 1.709y = -1.173 \\ e_2 \quad -0.800x + 1.700y = -1.200 \end{array} \right\}$$

ill-conditioned

$$\kappa \approx 228$$

# Conditioning

$\|A\| \|A^{-1}\|$  is also important in many other scenarios:

- Matrix Multiplication
- Explicit Matrix Inversion:  $\frac{\|A^{-1} - (A + E)^{-1}\|}{\|A^{-1}\|} / \frac{\|E\|}{\|A\|} \leq \|A\| \|A^{-1}\|$
- Iterative Inversion Methods: (from [Qu et al. 2022, <https://arxiv.org/abs/2209.00809>])

	Jacobi	Gauss-Seidel	Steepest Descent	Conjugate Gradient
linear convergence rates	$\frac{\kappa(A)-1}{\kappa(A)+1}$	$\frac{\kappa(A)-1}{\kappa(A)+1}$	$\left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^2$	$\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}$

**Table 1** Rates of linear convergence of some iterative methods for solving the system  $Ax = b$

In many cases the condition number is as hard to calculate as the original problem



# From Problems to Algorithms

Recall the initial motivations for the concept of conditioning

The problems  $\{Ax = b, \lambda_1(A), \lambda_d(A), \dots\}$  all admit 'time based' solvers/algorithms

In these contexts  $\|A\| \|A^{-1}\|$  has a different meaning:

e.g.  $\nabla$ -descent on  $\frac{1}{2}w^T A w - b^T w$  with  $A > 0$  (solution @  $w^* = A^{-1}b$ )

Algorithm:  $w^{k+1} = w^k - \alpha(Aw^k - b)$

Decompose along the eigenvectors of  $A$ :  $x^k := Q^T(w^k - w^*)$  giving

$$x_i^{k+1} = (1 - \alpha\lambda_i(A))x_i^k = (1 - \alpha\lambda_i(A))^{k+1}x_i^0$$



# From Problems to Algorithms

$\nabla$ -descent on  $\frac{1}{2}w^T Aw - b^T w$  with  $A > 0$  (solution @  $w^* = A^{-1}b$ )

$$x_i^{k+1} = (1 - \alpha\lambda_i(A))x_i^k = (1 - \alpha\lambda_i(A))^{k+1}x_i^0 \quad (x^k := Q^T(w^k - w^*))$$

Rates of convergence are dominated by those along extremal eigenvectors

So is the choice of  $\alpha$

$$\text{Optimal } \alpha = \frac{2}{\lambda_1(A) + \lambda_d(A)} = \frac{1}{\lambda_d(A)} \frac{2}{\frac{\lambda_1(A)}{\lambda_d(A)} + 1} = \frac{2\|A^{-1}\|}{\|A\|\|A^{-1}\| + 1} \quad (\lambda_d = \|A^{-1}\|^{-1})$$

$$\text{Optimal rate} = \frac{\frac{\lambda_1(A)}{\lambda_d(A)} - 1}{\frac{\lambda_1(A)}{\lambda_d(A)} + 1} = \frac{\|A\|\|A^{-1}\| - 1}{\|A\|\|A^{-1}\| + 1}$$

So both *stability and rate of convergence* are governed by  $\kappa(Ax = b) = \|A\|\|A^{-1}\|$

Not only does  $\kappa$  describe the generic difficulty of computing a solution, it dictates performance of particular algorithms.

# Sampling

We are interested in computing  $\mathbb{E}_\pi[f(X)]$

$\pi$  is sufficiently complex that we do not have immediate access to its properties either

- Analytically
- Or by sampling from  $\pi$  independently and forming a Monte Carlo estimator

Markov chain Monte Carlo (MCMC): form a Markov chain  $\{X_t\}_{t=1}^N$  such that  $\mathcal{L}(X_t) \rightarrow \pi$

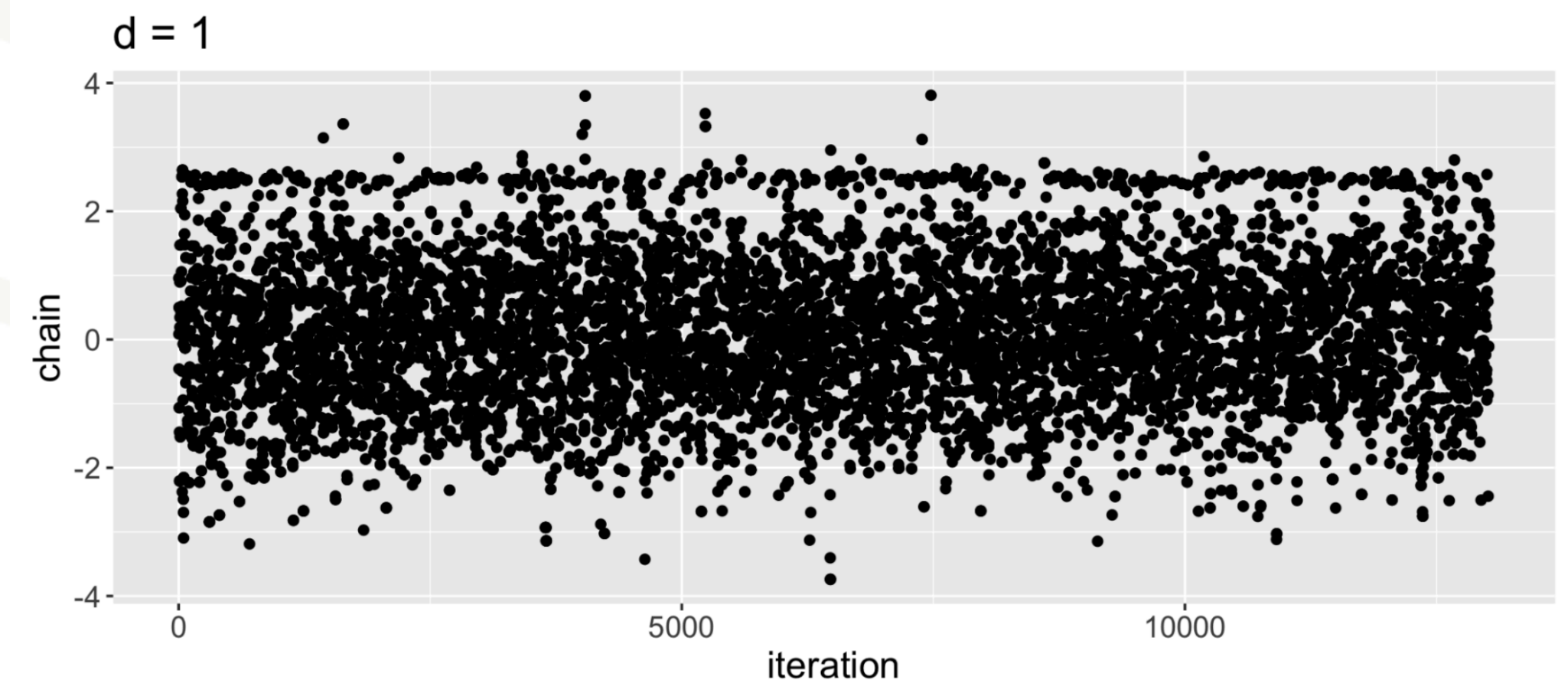
Every MCMC method usually comes with the following guarantee:

$$\frac{1}{N} \sum_{t=1}^N f(X_t) \rightarrow \mathbb{E}_\pi[f(X)] \text{ for all } X_1 \sim \mu_0$$

for  $f$  and  $\pi$  in given classes

## Effective Sample Size:

# independent samples to achieve an estimator of the same variance





# Mixing Time and Spectral Gap

A good proxy for the performance of MCMC methods is the 'speed' at which  $\mathcal{L}(X_t) \rightarrow \pi$

## Spectral Gap

Let  $X_0 \sim \pi$  and  $X_1$  be sampled according to one step of the Markov chain starting at  $X_0$

Then (under simplifying assumptions) we define the spectral gap of the chain as:

$$\gamma := 1 - \sup_{\text{Var}_{\pi}(g) < \infty} \text{Corr}(g(X_0), g(X_1))$$

Chain converges geometrically in the  $\chi^2$  divergence with rate  $1 - \gamma$

## $\varepsilon$ -Mixing Time

Let  $\varepsilon > 0$  and  $d$  be a distance-like function on the space of probability distributions

We define (informally) the  $\varepsilon$ -mixing time of a Markov chain in a distance  $d$  as:

$$\tau_d(\varepsilon) := \inf\{t : d(\mathcal{L}(X_t), \pi) \leq \varepsilon\}$$

Implicit: initial distribution i.e.  $\mathcal{L}(X_0)$



# Condition number in Sampling

Target in the form  $\pi \propto \exp(-U(x))$  on  $\mathbb{R}^d$  such that  $m\mathbf{I}_d \leq \nabla_x^2 U(x) \leq M\mathbf{I}_d$  for all  $x \in \mathbb{R}^d$ :  
 $U : \mathbb{R}^d \rightarrow \mathbb{R}$  is *m-strongly convex* and *M-smooth*

*m*-strong convexity:

Unimodal

*m* measures the curvature of  $U(x)$

e.g. posterior with concave log-likelihood, Gaussian prior

*M*-smoothness:

- $\nabla_x U(x)$  is *M*-Lipschitz
- Discretisations work nicely
- Minimum average acceptance ( $\alpha_0$ )
- controlled [Andrieu et al 2022]

The condition number associated with *sampling from*  $\pi$  is

$$\kappa := \sup_{x \in \mathbb{R}^d} \|\nabla_x^2 U(x)\|_2 \sup_{x \in \mathbb{R}^d} \|\nabla_x^2 U(x)^{-1}\|_2$$

If  $m\mathbf{I}_d \leq \nabla_x^2 U(x) \leq M\mathbf{I}_d$  is tight  $\kappa = M/m$

As  $\kappa \rightarrow 1$ , the eigenvalues of  $\nabla_x^2 U(x)$  get squeezed together, and  $\pi$  starts to look more like an isotropic Gaussian

# Importance of the sampling condition number

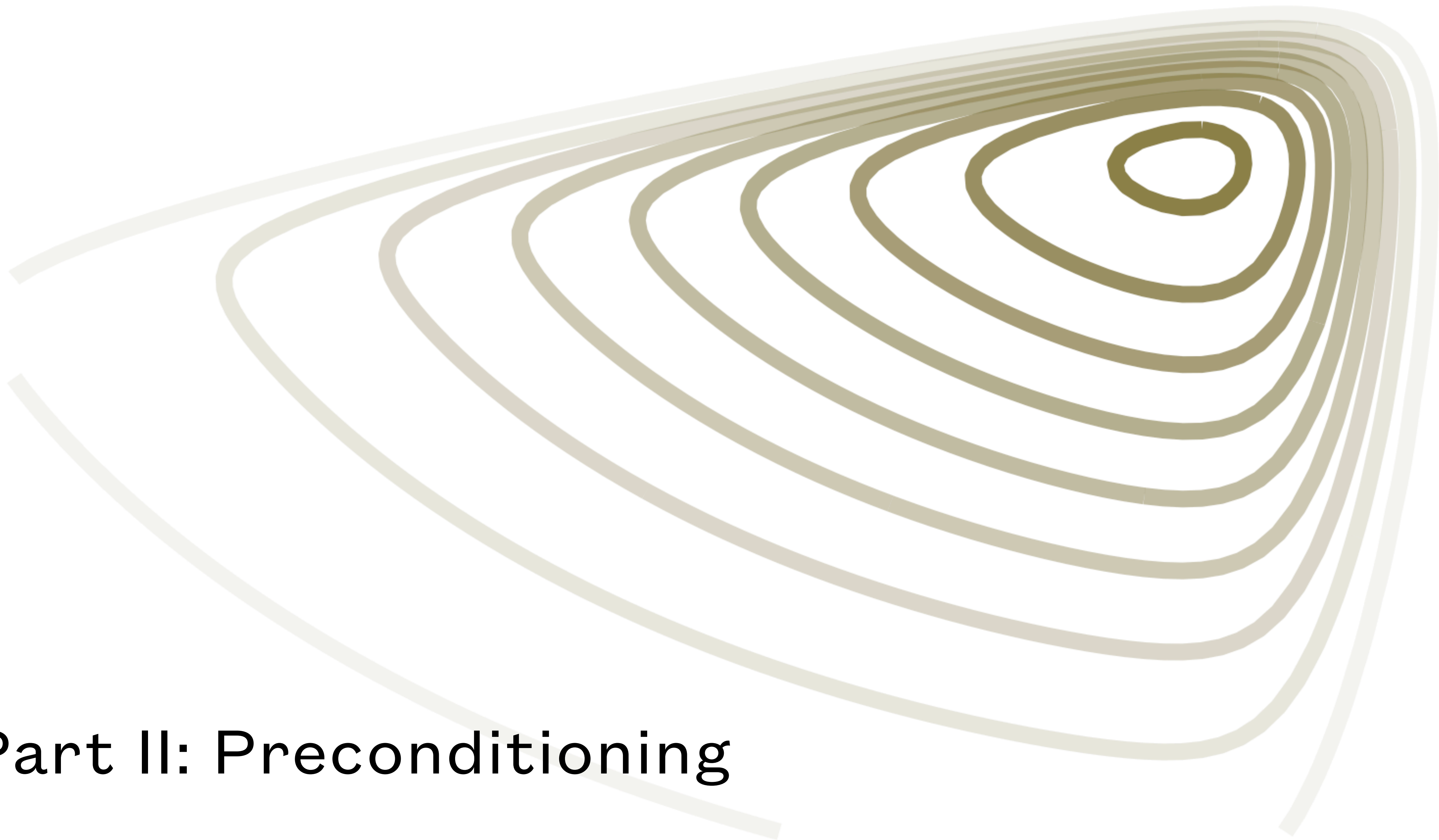
Introductory Material

	Spectral Gap		$\epsilon$ -Mixing Time	
Upper bounds	$\exists$	$O(\frac{\sqrt{\log d}}{\kappa\sqrt{d}})$ (on a Gaussian) Lee et al. [2021]	$\forall$	$O(d\kappa^2 \log \frac{1}{\epsilon})$ Dwivedi et al. [2019]
		$O(\frac{\log d}{\kappa d})$ Lee et al. [2021]		$O(d\kappa \log \frac{1}{\epsilon})$ Dwivedi et al. [2019]
		$O(\frac{\sqrt{\log d}}{\kappa\sqrt{d}})$ (on a Gaussian) Lee et al. [2021]		$O(d^{\frac{2}{3}}\kappa \log \frac{1}{\epsilon})$ Chen et al. [2019]
Lower bounds	$\forall$	$O(\frac{1}{\kappa d})$ Andrieu et al. [2022]	$\exists$	$O(\frac{\kappa d}{\log^2 d})$ ( $\epsilon = e^{-1}$ ) Lee et al. [2021]

All bounds up to logarithmic factors, mixing times in TV

Questions?





# Part II: Preconditioning

# Preconditioning

Preconditioning is a transformation from the original problem to a new one

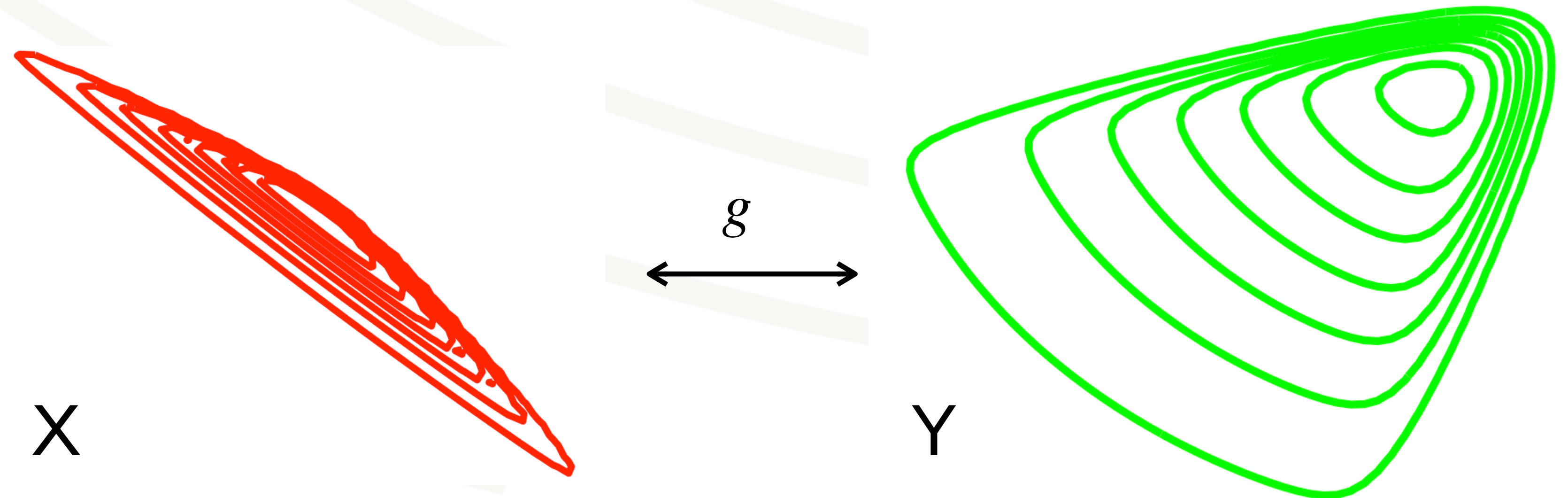
We do it to reduce the condition number:

e.g. starting with  $Ax = b$  make the transformation  $y = Mx, c = N^{-1}b$  to get to the problem  $NAMy = c$  with condition number  $\|NAM\| \|M^{-1}A^{-1}N^{-1}\|$

When  $Y = g(X) = LX$  for the condition number of sampling from the distribution of  $Y$  is

$$\kappa_L := \sup_{y \in \mathbf{R}^d} \|\nabla_y^2 \tilde{U}(y)\|_2 \sup_{y \in \mathbf{R}^d} \|\nabla_y^2 \tilde{U}(y)^{-1}\|_2 = \sup_{x \in \mathbf{R}^d} \|L^{-T} \nabla_x^2 U(x) L^{-1}\|_2 \sup_{x \in \mathbf{R}^d} \|L \nabla_x^2 U(x)^{-1} L^T\|_2$$

Used in all major MCMC software packages even though theory is lacking.



# Linear Preconditioning for Sampling

Intuition: set  $L$  to be the square root of some *representative* of  $\nabla_x^2 U(x)$  and hope that  $\kappa_L \ll \kappa$ , doesn't always work:

Let  $\Sigma_\pi := \text{Cov}_\pi(X)$

Diagonal Preconditioning:  $L = \text{diag}(\Sigma_\pi)^{-\frac{1}{2}}$

Gaussian target:

$$\nabla_x^2 U(x) = \Sigma_\pi^{-1} \text{ so } \kappa_L = \|\text{diag}(\Sigma_\pi)^{\frac{1}{2}} \Sigma_\pi^{-1} \text{diag}(\Sigma_\pi)^{\frac{1}{2}}\|_2 \|\text{diag}(\Sigma_\pi)^{-\frac{1}{2}} \Sigma_\pi \text{diag}(\Sigma_\pi)^{-\frac{1}{2}}\|_2 = \|C_\pi^{-1}\|_2 \|C_\pi\|_2$$

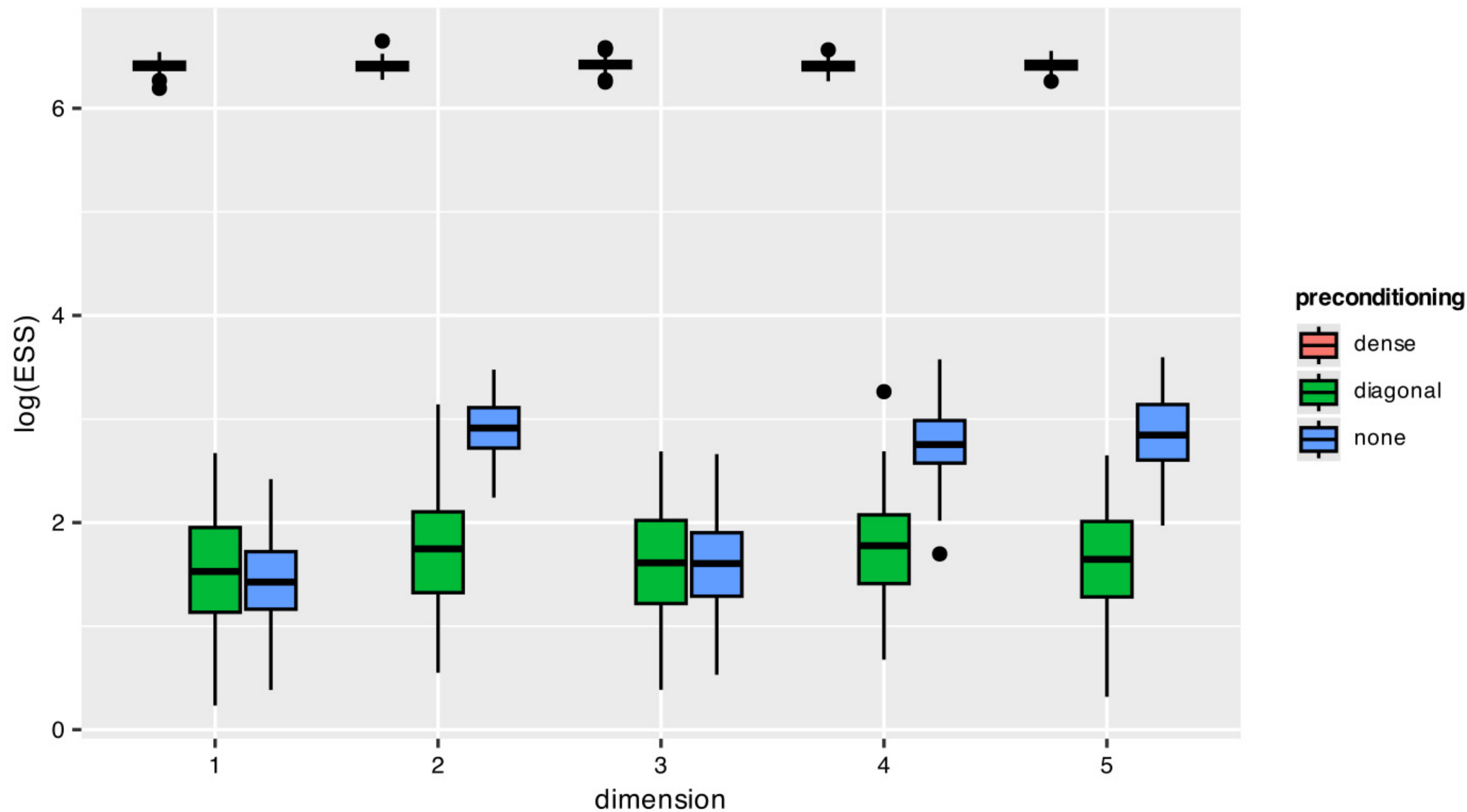
There exist Gaussian targets for which  $L = \text{diag}(\Sigma_\pi)^{-\frac{1}{2}}$  *increases the condition number*

$$\Sigma_\pi = \begin{pmatrix} 4.07, & -3.90, & 1.66 \\ -3.90, & 3.73, & -1.59 \\ 1.66, & -1.59, & 0.72 \end{pmatrix} \implies \kappa = 23,000, \kappa_L = 31,000$$



# Diagonal Preconditioning for Sampling

*Our Contribution*



# Linear Preconditioning: Bounding $\kappa_L$

Theorem: For a given preconditioner  $L \in GL_d(\mathbb{R})$  such that there exists  $\epsilon > 0$  for which

$$\|\nabla_x^2 U(x) - LL^T\|_2 \leq m\epsilon \quad (1)$$

for all  $x \in \mathbb{R}^d$ , we can bound  $\kappa_L$  in the following way:

$$\kappa_L \leq 1 + f(\epsilon)$$

Where  $f$  is an explicit polynomial, monotonic,  $\lim_{\epsilon \rightarrow 0+} f(\epsilon) = 0$

We verify the following preconditioners

- $LL^T = \text{Cov}_\pi(X)^{-1}$
- $LL^T = A$  where  $\nabla^2 U(x) = A + B(x)$

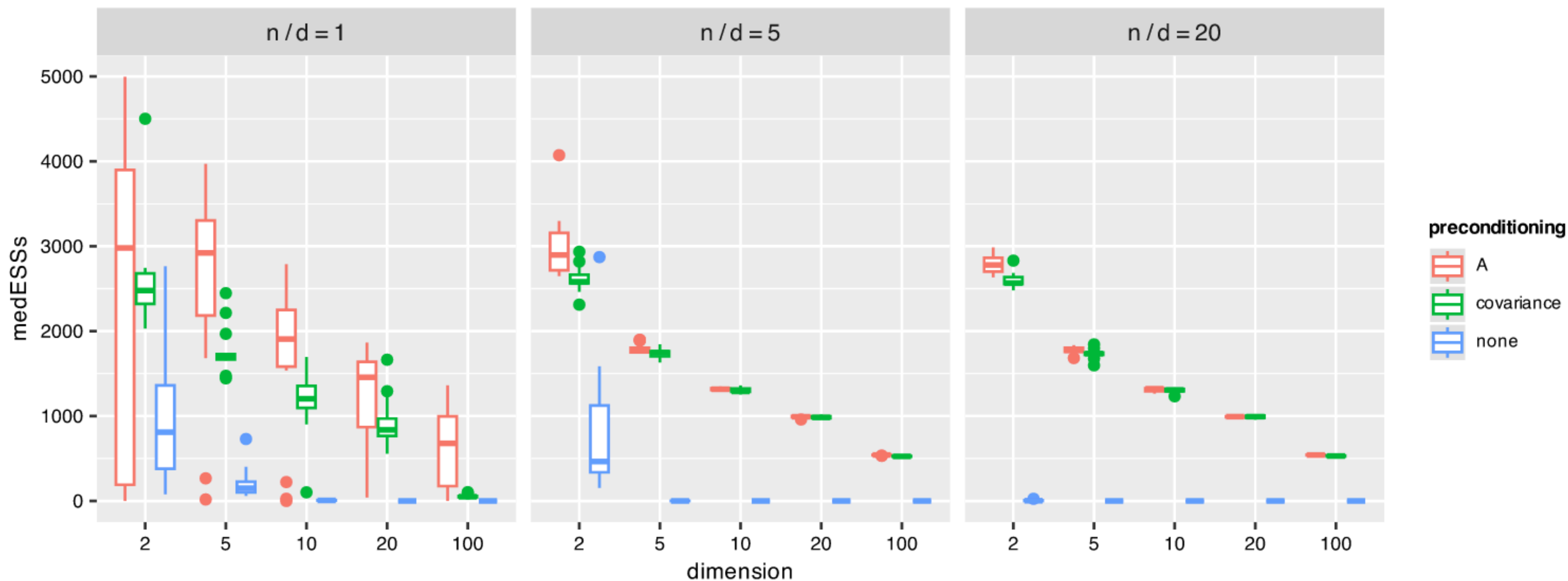


Figure 4: Boxplots of the medians of the ESSs across configurations of  $(n, d)$  with different preconditioners on the Bayesian linear regression with a Hyperbolic prior. The leftmost boxplot in each grouping corresponds to preconditioning with  $L = \sigma(X^T X)^{1/2}$  ('A' in the legend), the middle boxplot has  $L = \Sigma_\pi^{-1/2}$  ('covariance' in the legend), the rightmost has  $L = \mathbf{I}_d$  ('none' in the legend).



# Tight $\kappa$ Dependence of the Spectral Gap of Random Walk Metropolis

Our Contribution

The Spectral Gap of a Markov chain determines how quickly it converges

A bigger spectral gap means the Markov chain forgets its initial distribution quicker

Let  $\gamma_\kappa$  be the spectral gap of the Random Walk Metropolis Markov chain on a target with condition number  $\kappa$

Theorem: Assume that there exists  $\epsilon > 0$  such that

$$\|\nabla^2 U(x) - \nabla^2 U(y)\| \leq m\epsilon$$

for all  $x, y \in \mathbb{R}^d$ . Then

$$C\xi \exp(-2\xi) \frac{1}{\kappa} \frac{1}{d} \leq \gamma_\kappa \leq (1 + 2\epsilon) \frac{\xi}{2} \frac{1}{\kappa} \frac{1}{d}$$

where  $C = 1.972 \times 10^{-4}$  and  $\xi > 0$  depends on tuning parameters.

Corollary:

$$0 \leq \frac{\gamma_\kappa}{\gamma_{\kappa_L}} \leq 1$$

# Take-aways

- Conditioning describes how well an algorithm works on a problem via a quantity known as the condition number
- Finding the condition number is often as hard as the problem itself: bounds on it are useful since...
- It is ubiquitous in the fields of numerical linear algebra and convex optimisation. It is less well known in sampling, but nonetheless important.
- Preconditioning is a transformation which lowers the condition number.
- We provide results on current preconditioning practices in sampling.
- We provide generic bounds on the condition number.
- We assert conditions under which the dependence of the spectral gap of a popular algorithm is tight with respect to the condition number

# Work in Progress

- Review paper on Preconditioning for MCMC
  - Joint work with Sam Power, University of Bristol
- Non-asymptotic guarantees to learn a preconditioner
  - Joint work with Jeffrey Negrea, University of Waterloo and Florian Maire, Université de Montréal
  - In submission to the Conference on Learning Theory
- Adaptive computationally lightweight preconditioning method
  - Joint work with Samuel Livingstone, University College London
- Choosing a Riemannian metric preconditioner with Stein kernels
  - Joint work with Jeffrey Negrea and Jan Moisel, University of Waterloo





Thanks!

<https://jmlr.org/papers/v26/23-1633.html>