Solving Large Dense Linear Systems with Covariance Matrices

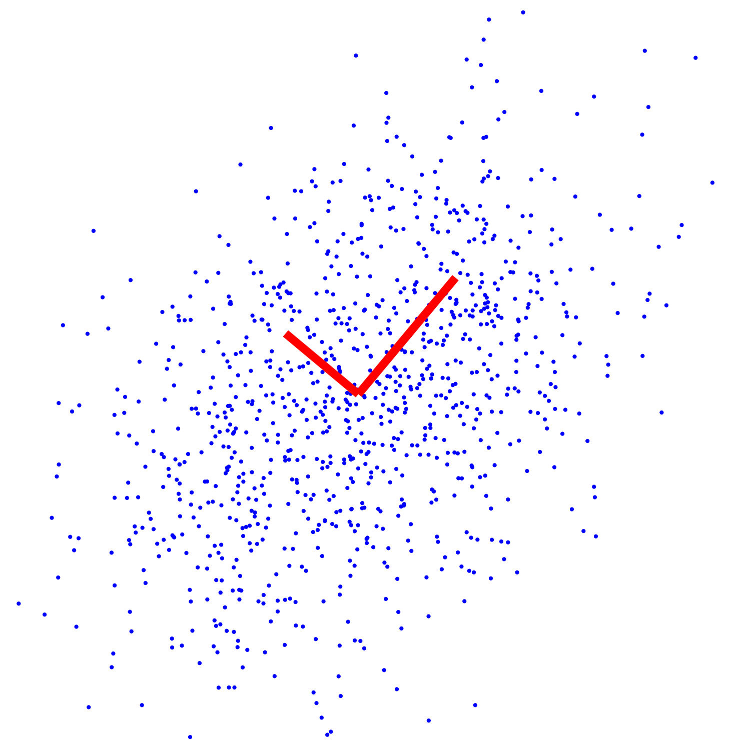
Jie Chen

Mathematics and Computer Science Division
Argonne National Laboratory

Introduction

Covariance matrices appear in every corner of statistical analysis:

- Multivariate statistics
- Stochastic processes
- Sampling
- Max likelihood fitting
- Interpolation; kriging
- Regression and classification
- Prediction; forecasting



The handling of covariance matrices incurs many matrix computations

Introduction

Example: Sampling

Generate a random vector from an n-dimensional normal distribution with mean μ and covariance matrix K. Steps:

1. Compute a Cholesky factorization $K = LL^T$
2. Generate a random vector z with i.i.d. standard normal variables
3. The vector $y = Lz + \mu$ is one such sample, because ...

$$\mathbb{E}[y] = L \cdot \mathbb{E}[z] + \mu = \mu$$

$$\operatorname{cov}[y] = \mathbb{E}[(y-\mu)(y-\mu)^T] = \mathbb{E}[(Lz)(z^T L^T)] = K$$

Can replace L by $K^{1/2}$, so need to compute $K^{1/2}z$.

Introduction

Example: Maximum likelihood estimation

[Opposite of sampling:] Given a vector y, what is the most likely normal distribution it comes from? Assuming that mean is zero and that the covariance matrix K is parameterized by θ, then maximize the likelihood

$$\mathcal{L}(\theta) := (2\pi)^{-\frac{n}{2}} (\det K)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} y^T K^{-1} y\right)$$

- Can use any optimization method to solve $\max \log \mathcal{L}$ or $\nabla \log \mathcal{L} = 0$
- Need to evaluate $\log(\det K)$ and $K^{-1} y$
- $\log(\det K) = \mathrm{tr}(\log K) \approx \frac{1}{N} \sum_{i=1}^{N} u_i (\log K) u_i$
- $[\log(\det K)]' = \mathrm{tr}(K^{-1} K' K^{-1}) \approx \frac{1}{N} \sum_{i=1}^{N} u_i (K^{-1} K' K^{-1}) u_i$

Introduction

Example: Interpolation

Given some points x_i $(i = 1, \dots, n)$ and their function values $f(x_i)$, what is the function value of an unknown point x_0? If we assume that

- f is a sample path of a stochastic process with covariance function ϕ
- $f(x_0) = \sum_{i=1}^{n} w_i f(x_i)$, then the weights w_i are computed as

$$\begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \phi(x_1, x_1) & \cdots & \phi(x_1, x_n) \\ \vdots & \ddots & \vdots \\ \phi(x_n, x_1) & \cdots & \phi(x_n, x_n) \end{bmatrix}^{-1} \begin{bmatrix} \phi(x_1, x_0) \\ \vdots \\ \phi(x_n, x_0) \end{bmatrix}$$

Where does the above formula come from? Recall our old friend, least squares:

$$\min_w \|y - Aw\| \implies w = (A^T A)^{-1}(A^T y).$$

Introduction

We focus on

Solving linear system with covariance matrix K, where $K_{ij} = \phi(x_i - x_j)$

What is so special/challenging about covariance matrices?

- Can be very large
- Can be fully dense
- Can be increasingly ill-conditioned as matrix size grows
- Can be associated with a large number of random right-hand sides
- Positive definite

Linear Solver

Consider the conjugate gradient method for solving $Kx = b$

Require: Initial guess x_0, preconditioner $M \approx K$

1: Compute $r_0 = b - Kx_0$, $z_0 = M^{-1}r_0$ and $p_0 = z_0$

2: **for** $j = 0, 1, \ldots$ until convergence **do**

3: $\quad \alpha_j = (r_j, r_j)/(Kp_j, p_j)$

4: $\quad x_{j+1} = x_j + \alpha_j p_j$

5: $\quad r_{j+1} = r_j - \alpha_j K p_j$

6: $\quad z_{j+1} = M^{-1} r_{j+1}$

7: $\quad \beta_j = (r_{j+1}, z_{j+1})/(r_j, z_j)$

8: $\quad p_{j+1} = z_{j+1} + \beta_j p_j$

9: **end for**

Check list:

- ☐ Matrix-vector mult?
- ☐ Preconditioner?
- ☐ Parallelism?

A Simple Case: Regular Grid

But not at all trivial...

Consider that $K_{ij} = \phi(x_i - x_j)$ where the x_i's are on a regular grid

- K is (multilevel) Toeplitz
- Multiplying K with vector p requires embedding K to a larger (multilevel) circulant matrix, and padding p with zeros
- Multiplying (multilevel) circulant matrix needs (multi-dimensional) FFT
- A (multilevel) circulant preconditioner M can be constructed
- CG can be extended by using the seed method or block method to handle multiple right-hand sides

Parallel implementation? ... is a headache ... because too many data transfers and global synchronizations

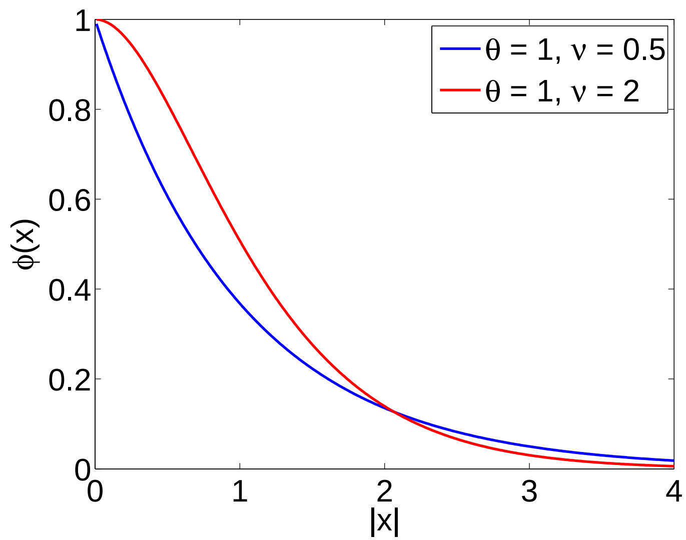
Preconditioning

In general, how do we precondition K?

Need some knowledge of ϕ... Matérn:

$$\phi(x) = \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\,\|x\|}{\theta} \right)^{\nu} \mathsf{K}_\nu \left(\frac{\sqrt{2\nu}\,\|x\|}{\theta} \right)$$

- θ: Scale parameter. Can also make function anisotropic.
- ν: Smoothness. Controls the differentiability of ϕ at 0.
- More flexible than Gaussian kernel (infinitely differentiable).
- When $\nu \to \infty$, it is Gaussian.



Spectral Density

(Covariance, spectral density) pair:

$$\phi(x) = \int_{\mathbb{R}^d} f(\omega) \exp(\mathbf{i}\, \omega^T x)\, d\omega, \qquad \text{with } f(\omega) > 0.$$

Spectral density of Matérn kernel

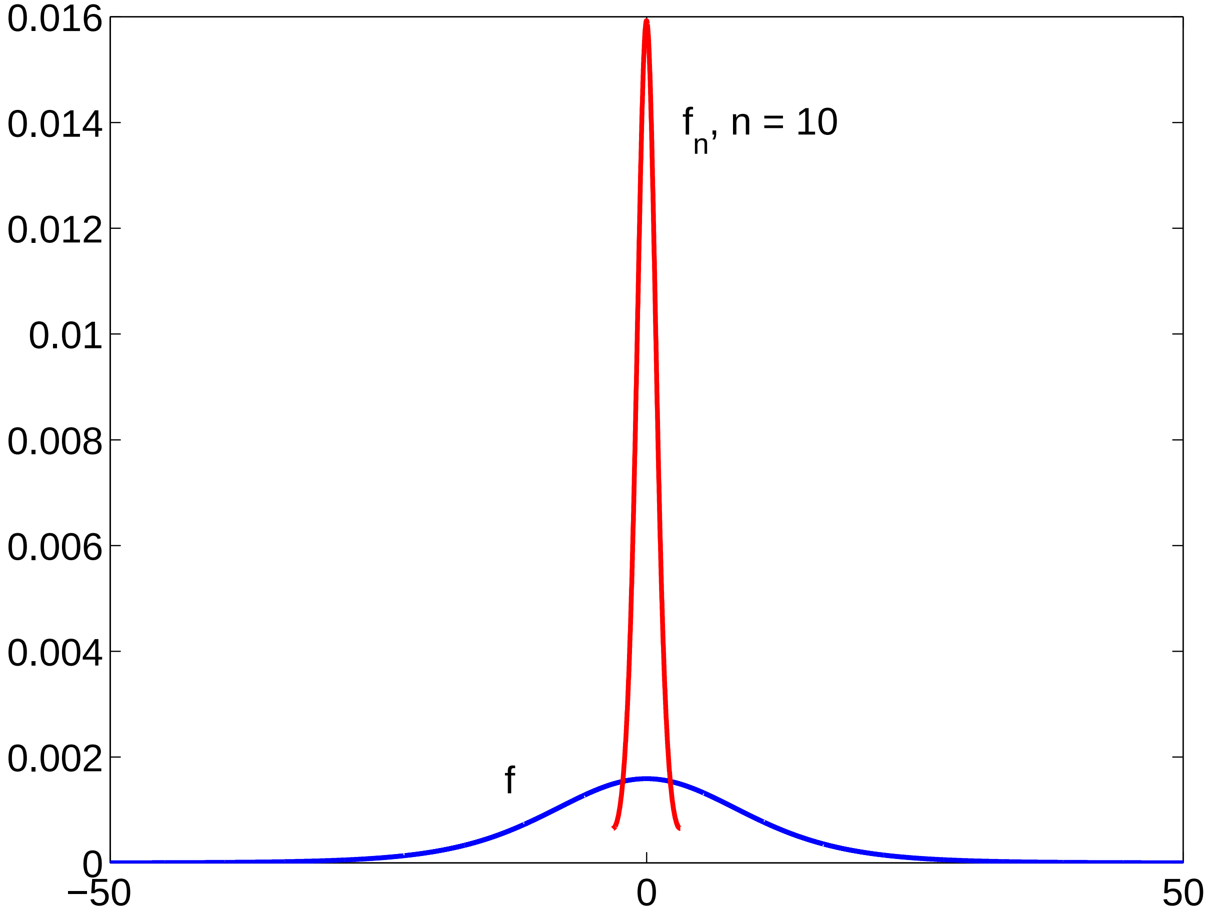
$$f(\omega) \propto \left(2\nu + \theta^2 \|\omega\|^2\right)^{-(\nu+d/2)}.$$

If regular grid, that is, $x_j = j/n$, then write

$$\phi(j/n) = \int_{[0,2\pi)^d} f_n(\omega) \exp(\mathbf{i}\, \omega^T j)\, d\omega$$

$$f_n(\omega) = n \sum_{l \in \mathbb{Z}^d} f\left(n \circ (\omega + 2\pi l)\right), \qquad \omega \in [0, 2\pi)^d.$$

Spectral Density



Spectrum

Bilinear form:

$$a^T K a = \sum_{j,l} a_j a_l \phi(x_j - x_l) = \int_{\mathbb{R}^d} f(\omega) \left| \sum_j a_j \exp(\mathbf{i}\, \omega^T x_j) \right|^2 d\omega.$$

If regular grid, that is, $x_j = j/n$, then write

$$\begin{aligned} a^T K a &= \int_{[0,2\pi)^d} f_n(\omega) \left| \sum_j a_j \exp(\mathbf{i}\, \omega^T j) \right|^2 d\omega \\ &\approx \frac{(2\pi)^d}{n} \sum_{0 \le k \le n-1} f_n(2\pi k/n) \left| \sum_{0 \le j \le n-1} a_j \exp(\mathbf{i}\, (2\pi k/n)^T j) \right|^2. \end{aligned}$$

Intuitively, eigenvalues of K "similar to" $(2\pi)^d f_n(2\pi k/n)$.

Spectrum

Definition: Two sets of real numbers $\{a_j^{(n)}\}_{j=1,\ldots,n}$ and $\{b_j^{(n)}\}_{j=1,\ldots,n}$ are equally distributed in the interval $[M_1, M_2]$ if for any continuous function $F : [M_1, M_2] \to \mathbb{R}$,

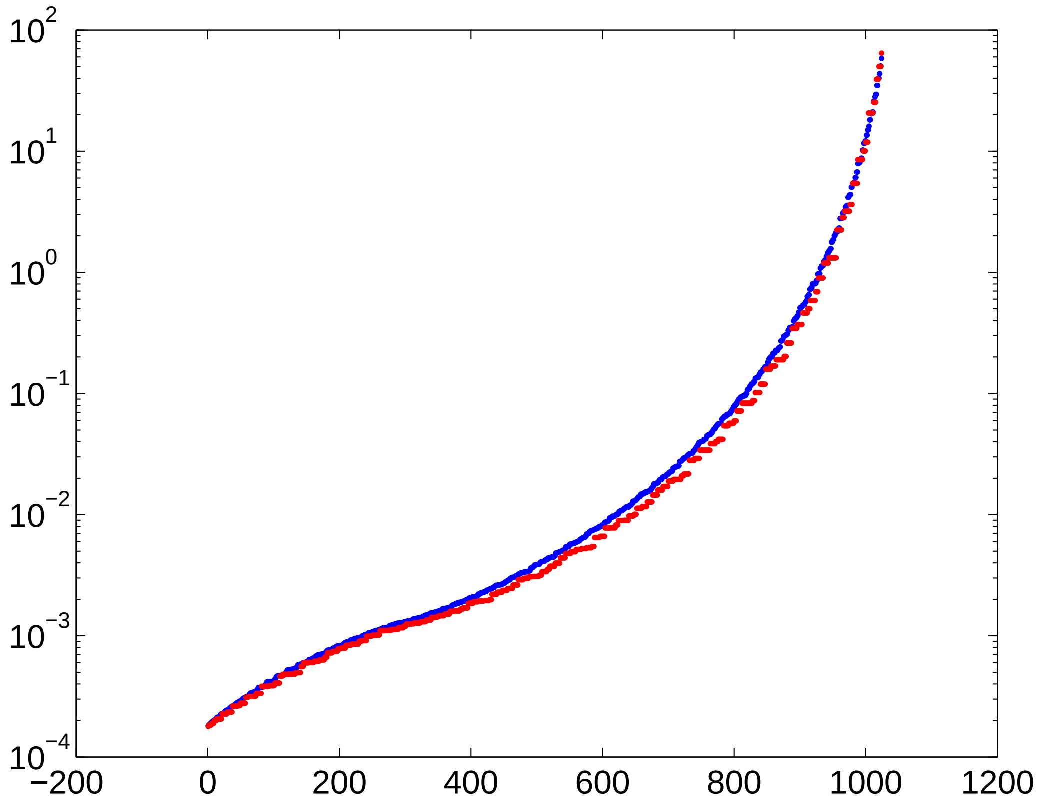
$$\lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} [F(a_j^{(n)}) - F(b_j^{(n)})] = 0.$$

Theorem: If $\phi \in L^1 \cap L^2$, then the set of eigenvalues of K/n and the set $\{(2\pi)^d f_n(2\pi j/n)/n\}$ are equally distributed.

Message: Loosely speaking, the spectrum of K is like f_n evenly sampled on $[0, 2\pi)$.

Spectrum

Matérn kernel ($d = 2$, $\nu = 3$). Blue: eigenvalues of K. Red: $(2\pi)^d f_n(2\pi j/n)$.



Preconditioning

Preconditioning idea: suppress the variation of f.

$$\Delta\phi(x) = \int_{\mathbb{R}^d} -\|\omega\|^2 f(\omega) \exp(\mathbf{i}\,\omega^T x)\, d\omega$$

Covariance	Spectral density
ϕ	$f(\omega) \propto \left(2\nu + \theta^2 \|\omega\|^2\right)^{-(\nu+d/2)} \asymp (1+\|\omega\|)^{-8}$
$\Delta\phi$	$\|\omega\|^2 f(\omega) \asymp \|\omega\|^2 (1+\|\omega\|)^{-8}$
$\Delta^2\phi$	$\|\omega\|^4 f(\omega) \asymp \|\omega\|^4 (1+\|\omega\|)^{-8}$
$\Delta^3\phi$	$\|\omega\|^6 (1+\|\omega\|)^{-8}$
$\Delta^4\phi$	$\|\omega\|^8 (1+\|\omega\|)^{-8}$
$\vdots$	$\vdots$

Preconditioning

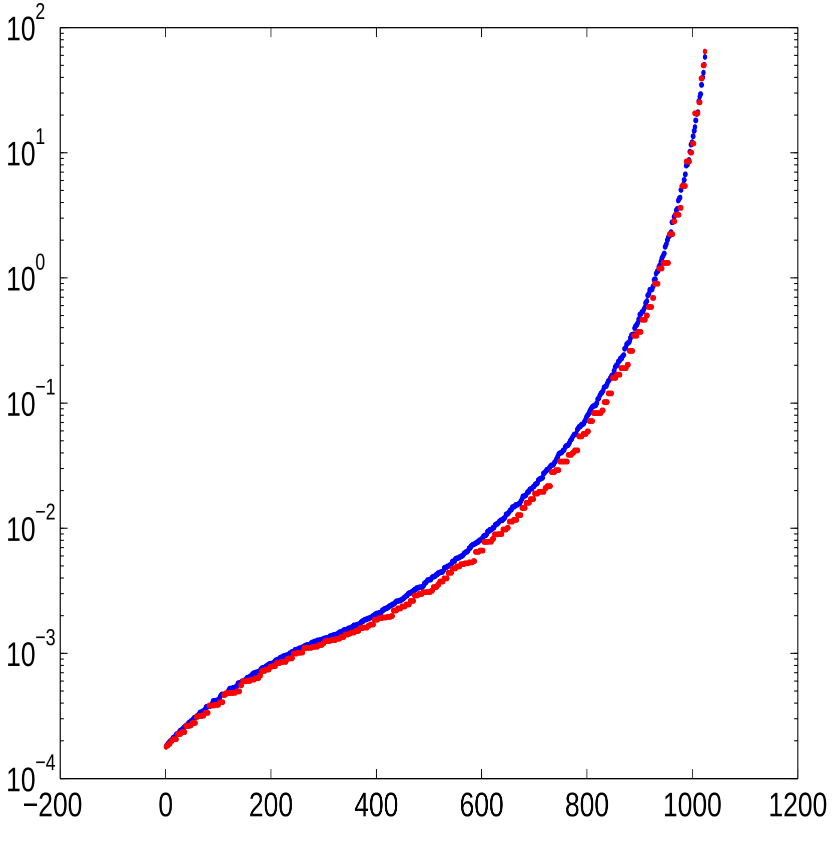
$$f_n^{[s]}(\omega) = \left[-4\sum_{p=1}^{d} n_p^2 \sin^2\left(\frac{\omega_p}{2}\right)\right]^s f_n(\omega)$$

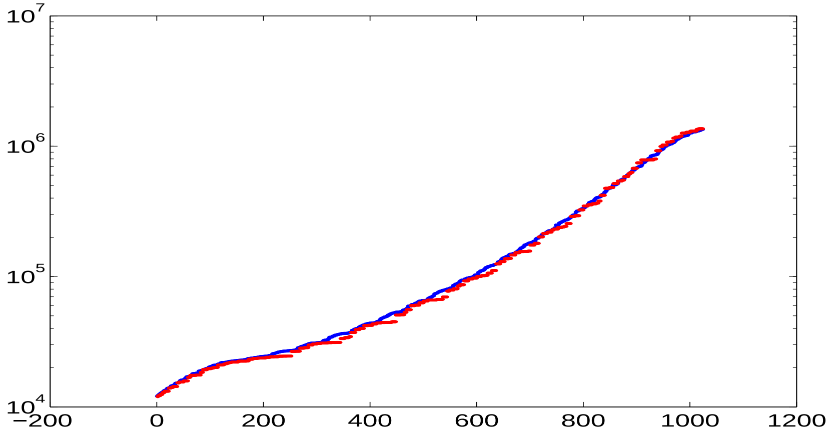
Discrete case: (L, D: discrete Laplacian)

Matrix	Entry
K	$\phi(j/n) = \int_{[0,2\pi)^d} f_n(\omega) \exp(\mathbf{i}\,\omega^T j)\, d\omega$
	$\mathrm{D}\,\phi(j/n) = \int_{[0,2\pi)^d} f_n^{[1]}(\omega) \exp(\mathbf{i}\,\omega^T j)\, d\omega$
$K^{[2]} = LKL^T$	$\mathrm{D}^2\,\phi(j/n) = \int_{[0,2\pi)^d} f_n^{[2]}(\omega) \exp(\mathbf{i}\,\omega^T j)\, d\omega$
	$\mathrm{D}^3\,\phi(j/n) = \int_{[0,2\pi)^d} f_n^{[3]}(\omega) \exp(\mathbf{i}\,\omega^T j)\, d\omega$
$K^{[4]} = LK^{[2]}L^T$	$\mathrm{D}^4\,\phi(j/n) = \int_{[0,2\pi)^d} f_n^{[4]}(\omega) \exp(\mathbf{i}\,\omega^T j)\, d\omega$

Preconditioning

Same Matérn kernel as in page 14. Left: original. Right: after Laplacian.





Preconditioning

Matrix	Entry
$K^{[2]} = LKL^T$	$\mathrm{D}^2\,\phi(j/n) = \int_{[0,2\pi)^d} f_n^{[2]}(\omega)\exp(\mathbf{i}\,\omega^T j)\,d\omega$

Note:

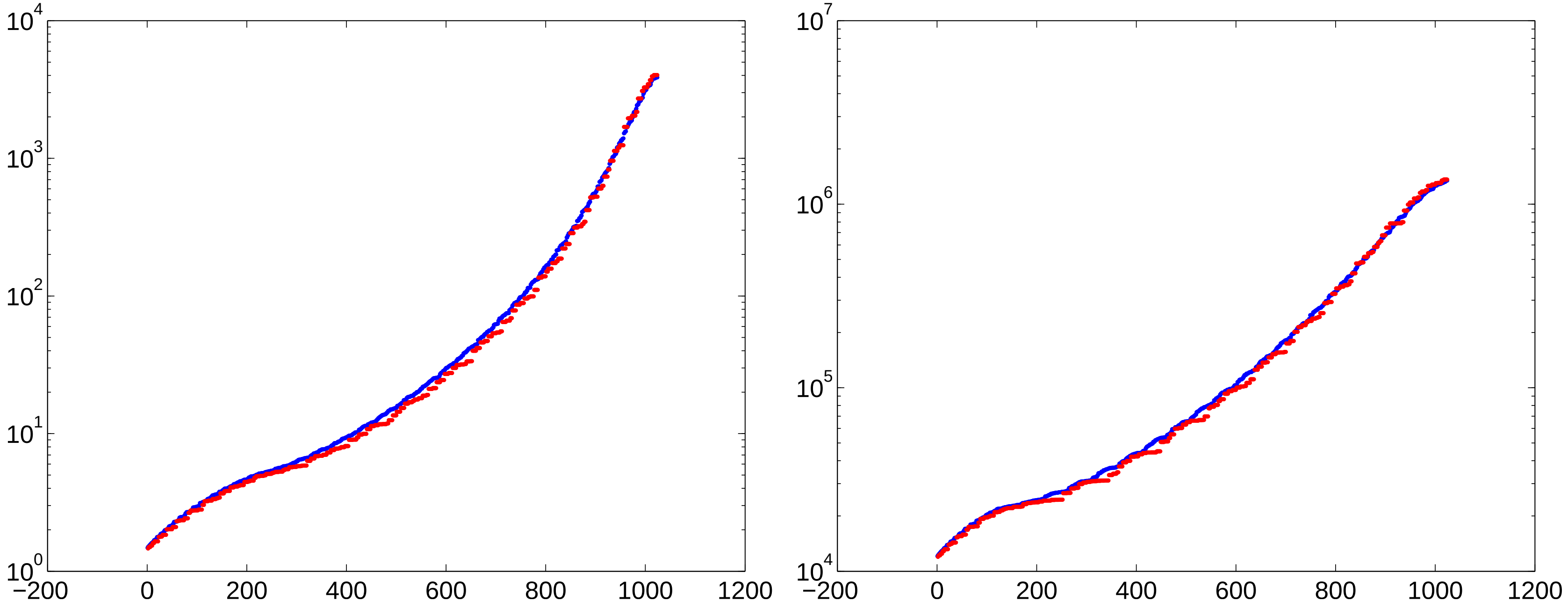
- Can define $K^{[s]}$, $\mathrm{D}^s\,\phi$ and $f_n^{[s]}$ for odd number s; but unknown how to write $K^{[s]}$ using L and K
- L has fewer rows than columns (unknown how to define on grid boundary)

Nevertheless,

Theorem: If all the partial derivatives of ϕ of order up to $2s+1$ belong to $L^1 \cap L^2$, then the eigenvalues of $K^{[s]}/n$ and the set $\{(2\pi)^d f_n^{[s]}(2\pi k/n)/n\}$ are equally distributed.

Preconditioning

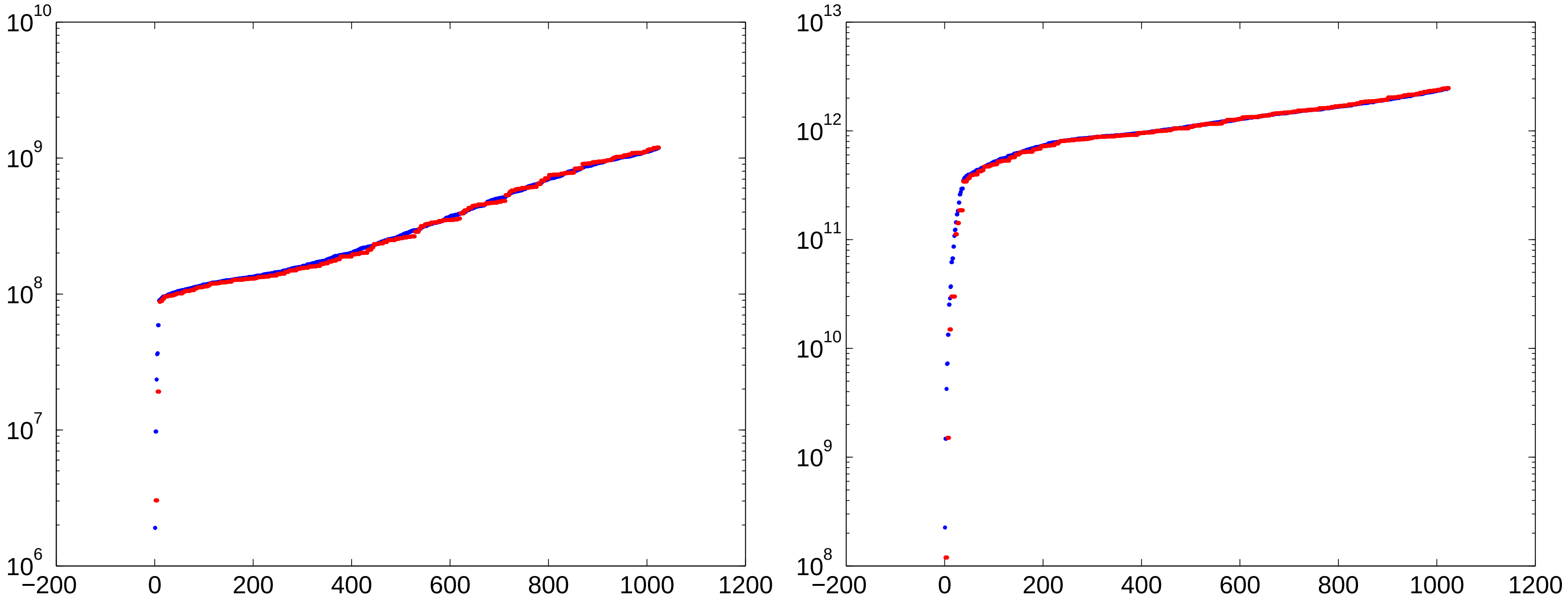
Left: $K^{[1]}$ and $(2\pi)^d f_n^{[1]}(2\pi j/n)$; Right: $K^{[2]}$ and $(2\pi)^d f_n^{[2]}(2\pi j/n)$.



Preconditioning

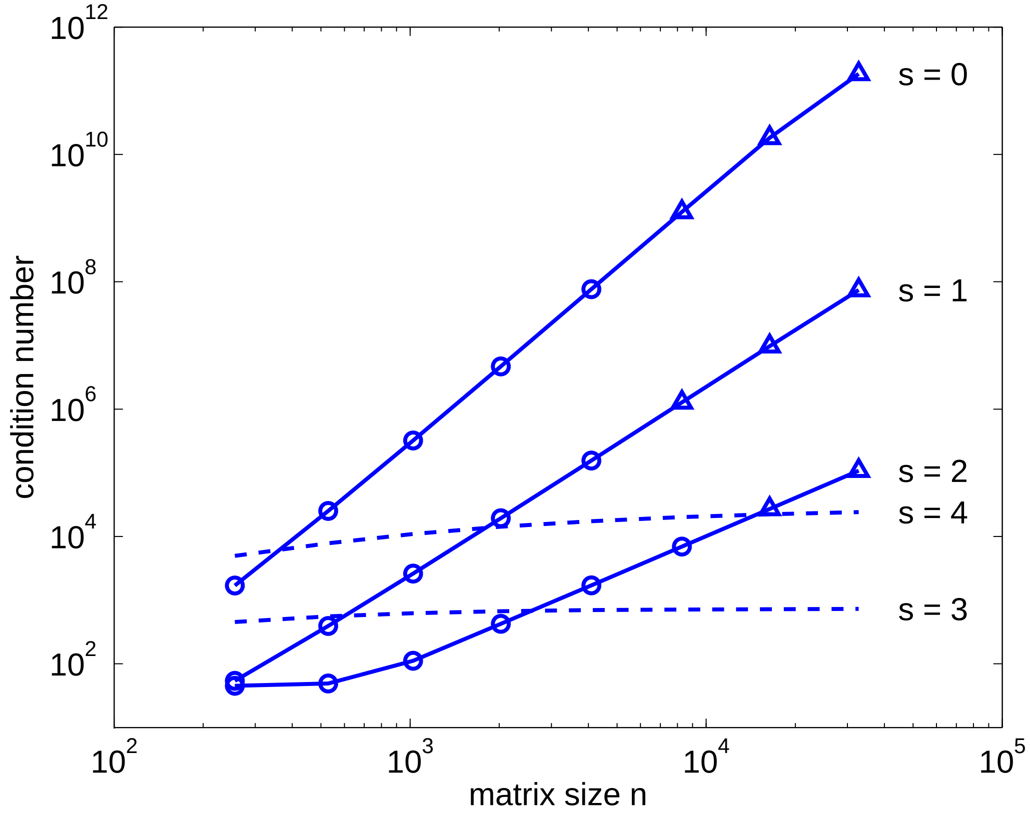
Can go further even though not supported by theorem

Left: $K^{[3]}$ and $(2\pi)^d f_n^{[3]}(2\pi j/n)$; Right: $K^{[4]}$ and $(2\pi)^d f_n^{[4]}(2\pi j/n)$.



Preconditioning

Growth of condition number of $K^{[s]}$



Irregular Grid

What about irregular grid?

$K^{[2]} = LKL^T$: Need to define L (discrete Laplacian)

$\longrightarrow$ consider finite element mesh

- The points $\{x_i\}$ define a domain Ω
- Nodal function $v_i(x)$ at x_i; piecewise linear
- For any twice differentiable u:

$$u \approx \sum_{i=1}^{n} u(x_i)\, v_i, \quad \Delta u \approx \sum_{i=1}^{n} \Delta u(x_i)\, v_i, \quad \nabla u \approx \sum_{i=1}^{n} u(x_i)\, \nabla v_i.$$

- ∇v_i not defined at edges and mesh nodes. Arbitrarily define them.

Irregular Grid

Green's identity

$$\int_{\Omega}(v\Delta u + \nabla v \cdot \nabla u) = \oint_{\partial\Omega} v(\nabla u \cdot \mathbf{n})$$

Discretization: for any $v = v_k$,

$$\sum_{i=1}^{n} \underbrace{\left[\int_{\Omega} v_k v_i\right]}_{M} \Delta u(x_i) + \sum_{i=1}^{n} \underbrace{\left[\int_{\Omega} \nabla v_k \cdot \nabla v_i\right]}_{-S} u(x_i) \approx \sum_{i=1}^{n} \underbrace{\left[\oint_{\partial\Omega} v_k \,(\nabla v_i \cdot \mathbf{n})\right]}_{B} u(x_i).$$

Matrix form:

$$M \cdot \Big[\Delta u(\boldsymbol{x}_i)\Big] \approx (B + S) \cdot \Big[u(\boldsymbol{x}_i)\Big].$$

Almost there...

Irregular Grid

Properties of

$$M = \left[\int_\Omega v_k v_i \right], \quad S = \left[- \int_\Omega \nabla v_k \cdot \nabla v_i \right], \quad B = \left[\oint_{\partial\Omega} v_k \left(\nabla v_i \cdot \mathbf{n} \right) \right]$$

- $M(k,k) = 2 \sum_{i \neq k} M(k,i)$
- Each row of S sum to zero. If x_k not on boundary, $\sum_i S(k,i)\, x_i = 0$
- Each row of B sum to zero. If x_k not on boundary, the row $B(k,:)$ is zero
- For each row k, $\sum_i (B+S)_{ki}\, x_i = 0$

Irregular Grid

Definition of L:

- $M \cdot \left[\Delta u(\boldsymbol{x}_i)\right] \approx (B+S) \cdot \left[u(\boldsymbol{x}_i)\right]$
- M' diagonal, $M'(k,k) = \frac{3}{2} M(k,k)$
- $M' \cdot \left[\Delta u(\boldsymbol{x}_i)\right] \approx (B+S) \cdot \left[u(\boldsymbol{x}_i)\right]$
- Remove rows and cols of M' corresponding to boundary: $M' \to \tilde{M}'$
- Remove rows of B corresponding to boundary: $B \to \tilde{B}$
- Remove rows of S corresponding to boundary: $S \to \tilde{S}$
- $\left[\Delta u(\boldsymbol{x}_i)\right] \approx (\tilde{M}')^{-1}(\tilde{B}+\tilde{S}) \cdot \left[u(\boldsymbol{x}_i)\right] = (\tilde{M}')^{-1}\tilde{S} \cdot \left[u(\boldsymbol{x}_i)\right]$

Irregular Grid

Definition of L:

$$L = (\tilde{M}')^{-1}\tilde{S}$$

Operator form (infinite mesh):

$$\mathrm{D}\,u(x_k) = \sum_{i=1}^{n} \frac{2S(k,i)}{3M(k,k)}\,u(x_i)$$

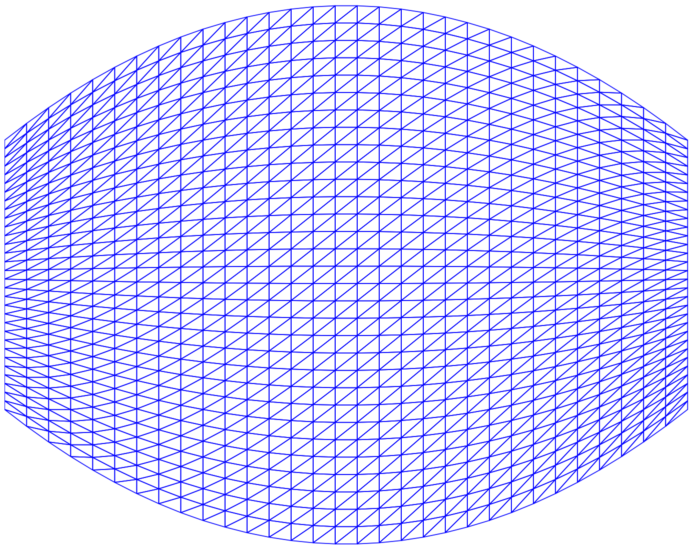
Theorem: For conforming mesh and any w vanishing on $\partial\Omega$,

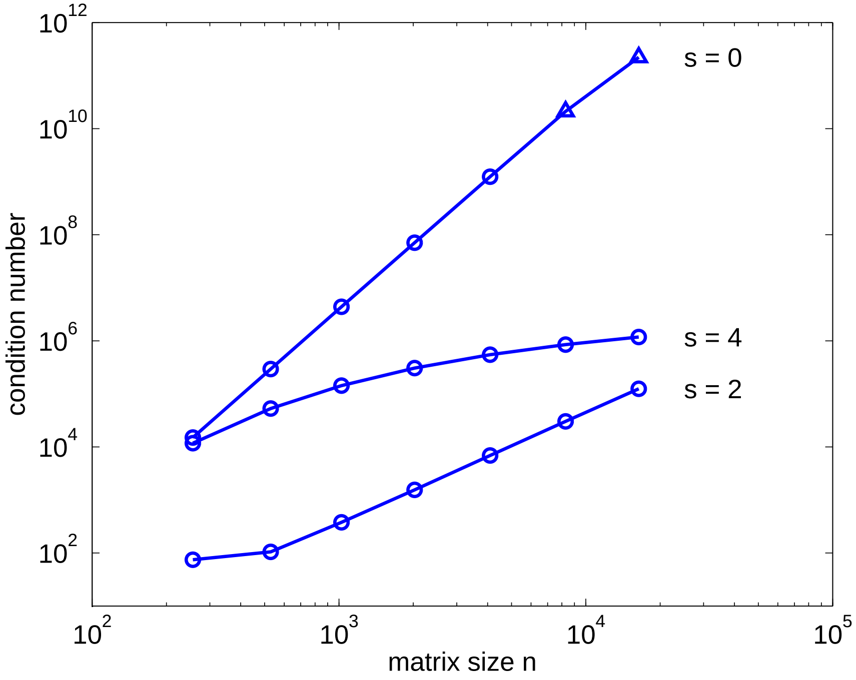
$$\left| \Big\langle [w(x_k)],\ [\Delta u(x_k) - \mathrm{D}\,u(x_k)] \Big\rangle_{M'} \right| \le C \cdot \mathrm{tr}(M') \cdot h,$$

where C is some constant and h is the maximum diameter of the elements.
Note: $\mathrm{tr}(M') = \frac{3}{d}\,\mathrm{meas}(\Omega)$, hence is fixed.

Irregular Grid

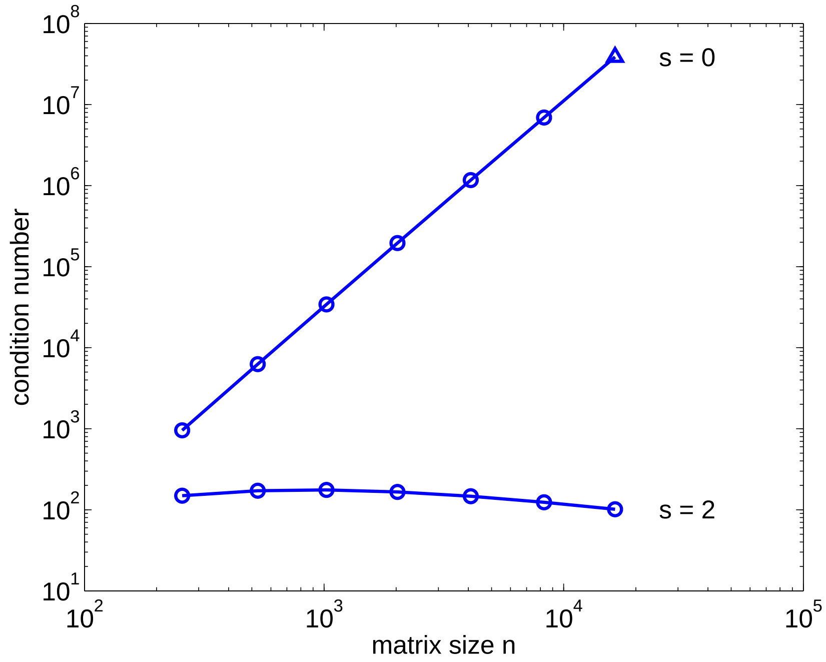
Growth of condition number of $K^{[s]}$. Matérn, $\nu = 3$



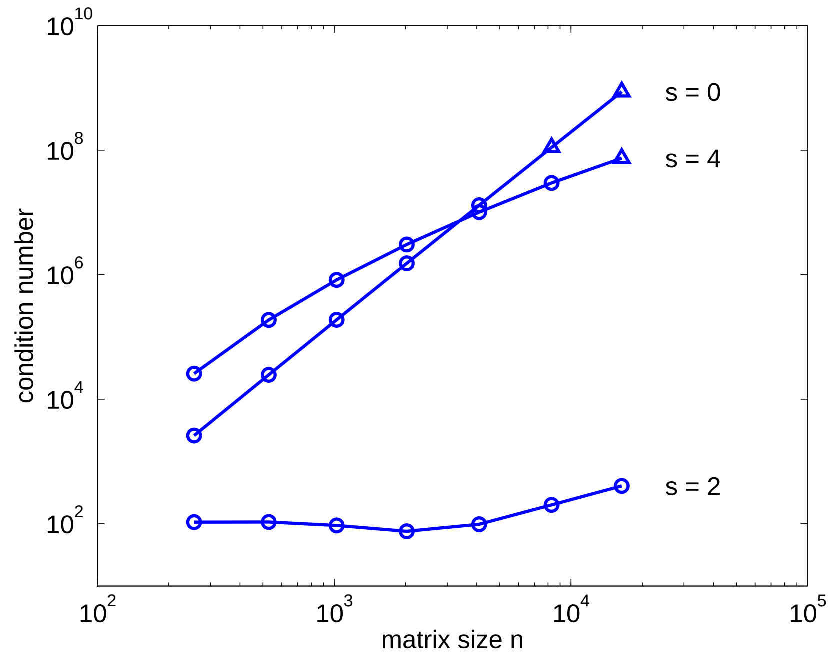


Irregular Grid

Left: $\nu = 1.5$



Right: $\nu = 2$



Summary

- Covariance matrix K, large, fully dense, increasingly ill conditioned
- K defined by covariance function $\phi(x)$
- Spectrum of K is tied to spectral density $f(\omega)$
- From spectral density f to one on grid: f_n
- Differentiating f gives better-behaved spectrum
- Define the linear transformation from K to $K^{[s]}$, for regular grid
- Extend the transformation for finite element mesh
- The preconditioning step is to multiply sparse matrix

- How about K-multiply?

Wish List

- (Regular grid case) Parallelization of conjugate gradient with FFT
- (General situation) Fast summation with a covariance kernel
- Better full-rank preconditioner
- Linear scaling
- Might be a good time to think about direct methods...