Abandoning the Cholesky factorisation
Efficient sampling from large Gaussian random vectors

Daniel Simpson*, Ian Turner¹, Chris Strickland¹ and Tony Pettitt¹
*Department of Mathematics, Norwegian University of Science and Technology, Trondheim, Norway
¹School of Mathematical Sciences, Queensland University of Technology, Brisbane, Australia

Why is this problem interesting?
Computations with large Gaussian random vectors are fundamental operations in modern computational statistics. They are required both when the model under investigation has some sort of (latent) Gaussian structure, and for a whole slew of modern MCMC methods that can attack more general problems. Two common methods are given below.

Problem 1 (Sparse)
Sample \( x \sim \mathcal{N}(0, Q^{-1}) \).
1. Factor the precision matrix \( Q = LL^T \).
2. Generate the sample \( x = L^{-T}z \), where \( z \sim \mathcal{N}(0, I) \).
   If \( Q \) is sparse, this is \( O(n^{3/2}) \).

Problem 2 (Block Circulant)
Sample \( x \sim \mathcal{N}(0, Q_2^{-1}) \).
1. Diagonalise the precision matrix \( Q_2 = FAF^T \).
2. Generate the sample \( x = FA^{-1/2}z \), where \( z \sim \mathcal{N}(0, I) \).
   Use FFTs for an efficient diagonalisation (\( O(n \log(n)) \)).

Why are these methods insufficient?
Consider two MCMC schemes for (approximate) conditional sampling from a log-Gaussian Cox process on a regular grid.
\[
\begin{align*}
y^T_{ij} | x_{ij} &\sim \mathcal{P}(e^{y^T_{ij}}) \\
x \sim \mathcal{N}(0, Q^{-1})
\end{align*}
\]
\( y_{ij} \) is the number of points in the \((i,j)\)th cell.
\( x \) takes a stationary Gaussian random field on a torus with an exponential covariance function.
\( Q \) is block-circulant and can be diagonalised with an FFT.
The condition number of \( Q \) is huge (\( O(h^{-1}) \), \( h \) is grid spacing)

An \( O(n \log n) \) method for sampling from a structured Gaussian random vectors

Remembering your roots
If we have a factorisation of the precision matrix \( Q = LL^T \), then it’s easy to see that \( x = L^{-T}z \), \( z \) i.i.d. standard normal, is a sample from \( \mathcal{N}(0, Q^{-1}) \).

- We need a version of \( L^{-T}z \) that we can compute using only matrix-vector products from \( Q \).
- We can use Krylov subspace methods to compute \( Q^{-1/2}z \).
- Convergence is analogous to the conjugate gradient method for computing \( Q^{-1}z \).

No easy down way
Why do basic Krylov methods not work (black line)?
- The rate of convergence depends on the condition number of \( Q \), which grows like \( O(h^{-1}) \).
- This is a standard problem with Krylov subspace methods.
- When solving \( Ax = b \), the solution is re-parameterisation (also known as preconditioning)!
- The figure shows the error in the sample from the second proposal distribution on a \( 32 \times 32 \) grid (y-axis) as the number of iterations increase (x-axis).

Preconditioning; or “Or not to centre”
The parameterisation of the model that we are given is not always the one we should use!
If we define a new variable \( z = F^T x \), then
\[
y \sim \mathcal{N}(0, (F^{-1}QF^{-T})^{-1})
\]
- By using \( y \) instead of \( x \) in the model, we replace \( Q \) with \( F^{-1}QF^{-T} \), which should be better behaved.
- In numerical linear algebra, this is known as preconditioning.
- Generic choice of \( F \) is the incomplete Cholesky (IC) factorisation of \( Q \).
- But in this case, we can do better!

But is it \( O(n \log n) \)?
Convergence depends on the condition number of \( F^{-1}(Q + A)F^{-T} \). It’s straightforward to show that this condition number is \( O(1) \) if we take \( F = Q^{1/2} \) (which can be computed in \( O(n \log n) \) operations using the FFT).
Hence the method is \( O(n \log n) \)
The following table shows how many iterates are needed to achieve an error of \( 10^{-8} \) problem gets larger.

<table>
<thead>
<tr>
<th>m (m x m grid)</th>
<th>16 32 64 128 256 512 1024 2048 4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditioned</td>
<td>6 6 6 6 6 6 6 6 6</td>
</tr>
<tr>
<td>Unpreconditioned</td>
<td>102 286 790 2166</td>
</tr>
</tbody>
</table>

mailto:daniel.simpson@math.ntnu.no