# Class 6 (Monday 21 November)

## Contents



These tasks are designed to be worked on in the practical class on Monday 21 November.

## Using CG

You can create a random 500 by 500 symmetric positive definite matrix by running:

```
import numpy as np
from numpy.random import RandomState

n = 500

rand = RandomState(0)

Q, _ = np.linalg.qr(rand.randn(n, n))
D = np.diag(rand.rand(n))
A = Q.T @ D @ Q
```

Solve $\mathbf{Ax} = \mathbf{b}$ for a random vector $\mathbf{b}$ using CG (`scipy.sparse.linalg.cg`). Make a plot showing the number of iterations vs the size of the residual.

## SPAI preconditioning

The SPAI preconditioner is defined by

$$\begin{aligned} \mathrm{C}_k &= \mathrm{A}\mathrm{M}_k \\ \mathrm{G}_k &= \mathrm{I} - \mathrm{C}_k \\ \alpha_k &= \mathrm{tr}(\mathrm{G}_k^{\mathrm{T}}\mathrm{A}\mathrm{G}_k)/\|\mathrm{A}\mathrm{G}_k\|_{\mathrm{F}}^2 \\ \mathrm{M}_{k+1} &= \mathrm{M}_k + \alpha_k \mathrm{G}_k \end{aligned}$$

Implement this preconditioner. Solve $\mathbf{Ax} = \mathbf{b}$ using $\mathrm{M}_m$ as a preconditioner for a range of values of $m$ and make a plot showing the number of iterations vs the size of the residual for each of these. If $m$ is too large, the preconditioner will take a long time to compute; if $m$ is too small, $\mathrm{M}_m$ will not be a good preconditioner. Experiment to find a good value to use for $m$.

You may wish to use the code included in the preconditioning section of the lecture notes as a template.

---

By Timo Betcke & Matthew Scroggs