PHAS0102: Techniques of High-Performance Computing

Sparse solvers

$$\mathrm{A}\mathbf{x} = \mathbf{b}$$

(known) sparse matrix

(unknown) vector

(known) vector

Symmetric matrices

- A matrix A is symmetric if $\mathrm{A}^{\mathrm{T}} = \mathrm{A}$
- eg $\begin{pmatrix} 7 & 1 & 0 \\ 1 & 0.5 & 4 \\ 0 & 4 & 0 \end{pmatrix}$

Positive definite matrices

- A matrix A is positive definite if for all non-zero vectors $\mathbf{x}$, $\mathbf{x}^{\mathrm{T}}\mathrm{A}\mathbf{x} > 0$
- eg $\begin{pmatrix} 7 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} 7 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = 7a^2 + 0.5b^2 + c^2$$

In the first part of today's lecture, we will look at some methods that can only by use for **symmetric positive definite** matrices.

Arnoldi iteration

$$\mathrm{V}_m = \texttt{orthogonalise}\left[\begin{pmatrix}\mathbf{b} & \mathrm{A}\mathbf{b} & \mathrm{A}^2\mathbf{b} & \mathrm{A}^2\mathbf{b} & \ldots \mathrm{A}^{m-1}\mathbf{b}\end{pmatrix}\right]$$

$$\mathrm{V}_m^{\mathrm{T}}\mathrm{A}\mathrm{V}_m\mathbf{y}_m = \mathrm{V}_m^{\mathrm{T}}\mathbf{b}$$

$$\mathbf{x} \approx \mathbf{x}_m = \mathrm{V}_m\mathbf{y}_m$$

$$\mathrm{A} \text{ is symmetric} \quad \Longrightarrow \quad \mathrm{V}_m^{\mathrm{T}}\mathrm{A}\mathrm{V}_m \text{ is symmetric}$$

Lanczos-Arnoldi

- Lanczos-Arnoldi is a specialised version of Arnoldi that make simplifications due to this symmetry.

An optimisation problem

$$\min_{\mathbf{x}} \underbrace{\left(\tfrac{1}{2}\mathbf{x}^{\mathrm{T}}\mathrm{A}\mathbf{x} - \mathbf{x}^{\mathrm{T}}\mathbf{b}\right)}_{f(\mathbf{x})}$$

$\nabla f = \mathrm{A}\mathbf{x} - \mathbf{b} \implies$ Turning point when $\mathrm{A}\mathbf{x} = \mathbf{b}$

$\mathrm{H}(f) = \mathrm{A}$ is positive definite $\implies$ Turning point is minimum

An optimisation method

$$\min_{\mathbf{x}} \left(\tfrac{1}{2}\mathbf{x}^{\mathrm{T}}\mathrm{A}\mathbf{x} - \mathbf{x}^{\mathrm{T}}\mathbf{b}\right)$$

Pick initial guess $\mathbf{x}_0$

Loop over i:

Pick a direction $\mathbf{d}_i$

Find $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i \mathbf{d}_i$ that minimises $\left(\tfrac{1}{2}\mathbf{x}^{\mathrm{T}}\mathrm{A}\mathbf{x} - \mathbf{x}^{\mathrm{T}}\mathbf{b}\right)$

Steepest descent method

$$\mathbf{d}_i = -\nabla f(\mathbf{x}_{i-1}) = \mathbf{b} - \mathrm{A}\mathbf{x}_{i-1}$$

$$\alpha_i = \frac{\mathbf{d}_i^{\mathrm{T}}\mathbf{d}_i}{\mathbf{d}_i^{\mathrm{T}}\mathrm{A}\mathbf{d}_i}$$

$$\boxed{\mathbf{x}_i = \mathbf{x}_{i-1} + \left(\frac{\mathbf{d}_i^{\mathrm{T}}\mathbf{d}_i}{\mathbf{d}_i^{\mathrm{T}}\mathrm{A}\mathbf{d}_i}\right)\mathbf{d}_i}$$

Conjugate directions

Pick a set of orthogonal directions $\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \ldots$

Pick α_i so that $\mathbf{e}_i^{\mathrm{T}} \mathbf{d}_i = 0$

$$\mathbf{e}_i = \mathbf{x}_i - \mathbf{x}^*, \quad \mathrm{A}\mathbf{x}^* = \mathbf{b}$$

$$\alpha_i = \frac{\mathbf{d}_i^{\mathrm{T}} \mathbf{r}_i}{\mathbf{d}_i^{\mathrm{T}} \mathrm{A} \mathbf{d}_i} \qquad \mathbf{r}_i = \mathbf{b} - \mathrm{A}\mathbf{x}_i$$

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \left(\frac{\mathbf{d}_i^{\mathrm{T}} \mathbf{r}_i}{\mathbf{d}_i^{\mathrm{T}} \mathrm{A} \mathbf{d}_i} \right) \mathbf{d}_i$$

Conjugate gradients (CG)

Pick a set of directions $\mathbf{d}_i = -\nabla f(\mathbf{x}_i)$ and orthogonalise them

Pick α_i so that $\mathbf{e}_i^{\mathrm{T}} \mathbf{d}_i = 0$

$$\mathbf{e}_i = \mathbf{x}_i - \mathbf{x}^*, \quad \mathrm{A}\mathbf{x}^* = \mathbf{b}$$

$$\alpha_i = \frac{\mathbf{d}_i^{\mathrm{T}} \mathbf{r}_i}{\mathbf{d}_i^{\mathrm{T}} \mathrm{A} \mathbf{d}_i} \qquad \mathbf{r}_i = \mathbf{b} - \mathrm{A}\mathbf{x}_i$$

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \left(\frac{\mathbf{d}_i^{\mathrm{T}} \mathbf{r}_i}{\mathbf{d}_i^{\mathrm{T}} \mathrm{A} \mathbf{d}_i} \right) \mathbf{d}_i$$

Condition number

The condition number of a matrix is

$$\kappa(\mathrm{A}) = \frac{\text{largest eigenvalue of A}}{\text{smallest eigenvalue of A}}$$

Convergence

Steepest descent

$$\|\mathbf{x}_i - \mathbf{x}^*\|_A \leqslant \left(\frac{\kappa - 1}{\kappa + 1}\right)^i \|\mathbf{x}_0 - \mathbf{x}^*\|_A$$

CG

$$\|\mathbf{x}_i - \mathbf{x}^*\|_A \leqslant 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^i \|\mathbf{x}_0 - \mathbf{x}^*\|_A$$

κ	10	50	100
$\frac{\kappa - 1}{\kappa + 1}$	0.82 11-12	0.96 60	0.98 110
$\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}$	0.52 3-4	0.75 8	0.82 11

Preconditioning

$$\mathrm{A}\mathbf{x} = \mathbf{b}$$

$$\mathrm{PA}\mathbf{x} = \mathrm{P}\mathbf{b} \qquad \mathrm{P}^{-1}\mathrm{A}\mathbf{x} = \mathrm{P}^{-1}\mathbf{b}$$

Aim: Pick P so that $\kappa(\mathrm{PA}) << \kappa(\mathrm{A})$

Preconditioning

- There is no general good preconditioner
  - Different preconditioners are looked for for different problems
  - Lots of different, highly specialised preconditioners exists

Preconditioners:
A simple preconditioner

$$\begin{pmatrix} 1/7 & 0 & 0 & 0 & 0 \\ 0 & 1/8 & 0 & 0 & 0 \\ 0 & 0 & 1/6 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 5 \end{pmatrix} \begin{pmatrix} 7 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0.2 & 0 \\ 0.5 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0.2 \end{pmatrix}$$

Preconditioners: Sparse approximate inverse

$$\|\mathrm{A}\|_{\mathrm{F}} := \sqrt{\sum_{i,j} |a_{i,j}|^2}$$

Aim: Find P so that $\|\mathrm{I} - \mathrm{AP}\|_{\mathrm{F}}$ is small

Preconditioners: Sparse approximate inverse

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

Is there time for a
[live Python demo]?