Krylov Subspace Methods

Kenton Lam

September 10, 2019

These notes aim to describe Krylov subspace methods, one of the best options for solving linear systems of equations. Briefly, they have the following desirable properties:

- No explicit form of $\mathbf{A}$ is needed; only a matrix-vector product is required.
- Well-suited for large and sparse systems.
- Optimised variations of Krylov methods are available for specific matrix types.
- For approximate solutions, Krylov methods have good convergence/approximation properties.

This is based on MATH3204 lectures and notes, lectured by Fred Roosta. These notes are meant to give an intuitive understanding of the topic and omit many proofs which can be found in the lecture slides.

1 Introduction

The general form of a linear system is

$$\mathbf{A}\boldsymbol{x}^\star = \boldsymbol{b}$$

where $\mathbf{A} \in \mathbb{C}^{\times n}$ and $\mathbf{A}$ is invertible. Assume $\rho(\mathbf{I} - \mathbf{A}) < 1$. Then, we can write $\mathbf{A}^{-1}$ as a geometric series,

$$\mathbf{A}^{-1} = (\mathbf{I} - (\mathbf{I} - \mathbf{A}))^{-1} = \sum_{k=0}^{\infty} (\mathbf{I} - \mathbf{A})^k.$$

Suppose we have an initial guess $\boldsymbol{x}_0 \in \mathbb{C}^n$. Define the residual of this guess as $\boldsymbol{r}_0 = \mathbf{A}\boldsymbol{x}^\star - \mathbf{A}\boldsymbol{x}_0$. Then,

$$\begin{aligned} \boldsymbol{x}^\star &= \mathbf{A}^{-1}\boldsymbol{b} = \mathbf{A}^{-1}(\mathbf{A}\boldsymbol{x}^\star - \mathbf{A}\boldsymbol{x}_0 + \mathbf{A}\boldsymbol{x}_0) \\ &= \boldsymbol{x}_0 + \mathbf{A}^{-1}\boldsymbol{r}_0 = \boldsymbol{x}_0 + \sum_{k=0}^{\infty} (\mathbf{I} - \mathbf{A})^k \boldsymbol{r}_0 \end{aligned}$$

This is great, but largely useless if we need to compute infinitely many vectors to find $\boldsymbol{x}^\star$. However it turns out that we actually don't need to.

Theorem (Cayley–Hamilton)**.** *Let $p_n(\lambda) = \sum_{i=0}^{n} c_i \lambda^i$ be the characteristic polynomial of the matrix $\mathbf{A}$. Then, $p_n(\mathbf{A}) = 0$.*

This implies that $\mathbf{A}^{-1}$ can be written as a finite sum of linear combinations of powers of $\mathbf{A}$. Specifically, it is a matrix polynomial of degree at most $n-1$. As a result,

$$\boldsymbol{x}^\star \in \boldsymbol{x}_0 + \operatorname{Span}\left\{\boldsymbol{r}_0, \mathbf{A}\boldsymbol{r}_0, \dots, \mathbf{A}^{n-1}\mathrm{r}_0\right\}.$$

Suppose we only consider a subspace of this, so choose $k < n$

$$\boldsymbol{x}_k \in \boldsymbol{x}_0 + \operatorname{Span}\left\{\boldsymbol{r}_0, \mathbf{A}\boldsymbol{r}_0, \dots, \mathbf{A}^{k-1}\mathrm{r}_0\right\}.$$

This is the central question of Kylov methods. How good is the approximation $\boldsymbol{x}_k$ to $\boldsymbol{x}^\star$? What does "good" even mean?

In fact, Richardson iterations is a case of these subspace approximations. In Richardson,

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \sum_{i=0}^{k-1} \alpha_i \prod_{j=0}^{i-1} (\mathbf{I} - \alpha_j \mathbf{A}) \boldsymbol{r}_0$$

However, depending on our choice of α_k, we saw dramatically different convergence.

The great quest of Krylov subspace methods is to find the the "best" (in some sense) $\boldsymbol{x}_k \approx \boldsymbol{x}^\star$ for some $k \ll n$.

Definition (Krylov Subspace)**.** *The Krylov subspace of order k, generated by the matrix $\mathbf{A}$ and vector $\boldsymbol{v}$ is defined as*

$$\mathcal{K}_k(\mathbf{A}, \boldsymbol{v}) = \operatorname{Span}\left\{\boldsymbol{v}, \mathbf{A}\boldsymbol{v}, \dots, \mathbf{A}^{k-1}\boldsymbol{v}\right\}$$

for $k \geq 1$ and $\mathcal{K}_0(\mathbf{A}, \boldsymbol{v}) = \{\mathbf{0}\}$.

Because these subspaces are nested, their dimensions cannot grow indefinitely. At some point, the Krylov subspace will be large enough that it "contains" all the information we can extract from $\mathbf{A}$ through its multiplication by $\boldsymbol{v}$. Consider the simplest case when $\boldsymbol{v}$ is an eigenvector, then the Kyrlov space just has dimension 1 for all k.

Theorem (Grade of $\boldsymbol{v}$ with respect to $\mathbf{A}$)**.** *There exists a positive integer $t = t(\boldsymbol{v}, \mathbf{A})$, the grade of $\boldsymbol{v}$ with respect to $\mathbf{A}$ such that*

$$\dim \mathcal{K}_k(\mathbf{A}, \boldsymbol{v}) = \min\{k, t\}.$$

This means that for any $k \leq t$, all the generated vectors are linearly independent. After t, the new vectors are linearly dependent on the previous ones. This means that for $k > t$, $\mathcal{K}_k(\mathbf{A}, \boldsymbol{v}) = \mathcal{K}_{k+1}(\mathbf{A}, \boldsymbol{v})$. As a direct corollary of this,

$$t = \min\left\{k \mid \mathbf{A}^{-1}\boldsymbol{v} \in \mathcal{K}_k(\mathbf{A}, \boldsymbol{v})\right\}.$$

Recall that initially we had $\boldsymbol{x}^\star \in \boldsymbol{x}_0 + \mathcal{K}_n(\mathbf{A}, \boldsymbol{r}_0)$. Now, we have a more specific result that

$$\boldsymbol{x}^\star \in \boldsymbol{x}_0 + \mathcal{K}_t(\mathbf{A}, \boldsymbol{r}_0)$$

where $\boldsymbol{r}_0 = \boldsymbol{b} - \mathbf{A} - \boldsymbol{x}_0$ and t is the grade of $\boldsymbol{r}_0$ with respect to $\mathbf{A}$.

To summarise, standard Krylov subspace solvers can be descibed as follows.

Definition (Standard Krylov Subspace Method)**.** *A standard Krylov subspace method is an iterative method, which starting from some $\boldsymbol{x}_0$, generates an appropriate sequence of iterates $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\mathbf{A}, \boldsymbol{r}_0)$ until it finds $\boldsymbol{x}^\star$ in exactly t steps.*

The iterates are chosen appropriately such that if we terminate early, we have still $\boldsymbol{x}_k \approx \boldsymbol{x}^\star$ in some sense.

Note that not all Krylov methods are of this form. Some are bulid upon different types of subspace or work with multiple subspaces (e.g. they also consider $\mathcal{K}_k(\mathbf{A}^*, \boldsymbol{w})$).

Many terms are intentionally left vague in the above definition, because Krylov subspace solders differ among themselves in many aspects, such as

- the underlying Krylov subspace,
- the method in which $\boldsymbol{x}_k$ is chosen, and
- the sense in which $\boldsymbol{x}_k \approx \boldsymbol{x}^\star$ is measured.

Additionally, in exact arithmetic, Krylov methods have finite termination property (i.e. they will always finish with an exact solution in finite iterations). Unfortunately, this does not hold in finite-precision arithmetic, such as on a computer.

2 Computing a Basis

2.1 Motivation

How do we construct vectors from some vector space? With a basis for that space, of course!

Suppose the grade of $\boldsymbol{r}_0$ w.r.t. $\mathbf{A}$ is n, so the basis matrix

$$\mathbf{K} = \begin{bmatrix}\boldsymbol{r}_0 & \mathbf{A}\boldsymbol{r}_0 & \cdots & \mathbf{A}^{n-1}\boldsymbol{r}_0\end{bmatrix} \in \mathbb{R}^{n\times n}$$

is invertible. Then,

$$\begin{aligned}\mathbf{AK} &= \begin{bmatrix}\mathbf{A}\boldsymbol{r}_0 & \mathbf{A}^2\boldsymbol{r}_0 & \cdots & \mathbf{A}^n\boldsymbol{r}_0\end{bmatrix} \\ &= \mathbf{K}\underbrace{\begin{bmatrix}\boldsymbol{e}_2 & \boldsymbol{e}_3 & \cdots & \boldsymbol{e}_n & \mathbf{K}^{-1}\mathbf{A}^n\boldsymbol{r}_0\end{bmatrix}}_{\mathbf{C}\in\mathbb{R}^{n\times n}}\end{aligned}$$

By construction, $\mathbf{K}^{-1}\mathbf{AK} = \mathbf{C}$. It can be seen that $\mathbf{C}$ is an $n \times n$ matrix and upper Hessenberg. Although $\mathbf{C}$ is sparse and easy to work with, such a basis is practically useless for our purposes.

- Because $\mathbf{C}$ is $n \times n$, we need n matrix-vector products.
- $\mathbf{K}$ could be very dense even if $\mathbf{A}$ is sparse.
- $\mathbf{K}$ is ill-conditioned.

Suppose we take the $\mathbf{QR}$ decomposition of $\mathbf{K}$, so $\mathbf{K} = \mathbf{QR}$ where $\mathbf{Q}$ is orthogonal and $\mathbf{R}$ is upper triangular. Then,

$$\mathbf{Q}^\top\mathbf{AQ} = \mathbf{RK}^{-1}\mathbf{AKR}^{-1} = \mathbf{RCR}^{-1} = \mathbf{H}$$

where $\mathbf{H}$ is an upper Hessenberg matrix. It can be seen that Range $\mathbf{K}$ is the same as Range $\mathbf{Q}$, so $\mathbf{Q}$ also spans our Krylov subspace.

For the subspace $\mathcal{K}_k(\mathbf{A}, \boldsymbol{r}_0)$, $k \ll n$, we search for $\mathbf{Q}_k \in \mathbb{R}^{n\times k}$ such that

$$\mathbf{Q}_k^\top \mathbf{A}\mathbf{Q}_k = \mathbf{H}_k \in \mathbb{R}^{k\times k}$$

is upper Hessenberg. Note that this $\mathbf{H}_k$ is only $k \times k$ so all our computations can be done with this smaller matrix. To summarise, we aim to find $\mathbf{Q}_k$ with the following properties:

- The columns of $\mathbf{Q}_k$ form an orthonormal basis of $\mathcal{K}_k(\mathbf{A}, \boldsymbol{r}_0)$.
- $\mathbf{Q}_k^\top \mathbf{A}\mathbf{Q}_k = \mathbf{H}_k$ is upper Hessenberg.

In general, $\mathbf{A}\mathbf{Q}_k \neq \mathbf{Q}_k\mathbf{H}_k$ for any $k < n$. Why is this so? Suppose we left-multiply $\mathbf{Q}_k^\top \mathbf{A}\mathbf{Q}_k = \mathbf{H}_k$ by $\mathbf{Q}_k$. If this was orthgonal, we'd have $\mathbf{Q}_k\mathbf{Q}_k^\top = \mathbf{I}_n$. However, the matrix product $\mathbf{Q}_k\mathbf{Q}_k^\top$ is essentially a map $\mathbb{R}^n \to \mathbb{R}^k \to \mathbb{R}^n$. If this is identity, it implies a k-dimensional set can span an n-dimensional space, which is absurd since $k < n$.

To get equality, we adjust with an error term $\mathbf{E}_k \in \mathbb{R}^{n\times k}$,

$$\mathbf{A}\mathbf{Q}_k = \mathbf{Q}_k\mathbf{H}_k + \mathbf{E}_k.$$

In order to have $\mathbf{Q}_k^\top \mathbf{A}\mathbf{Q}_k = \mathbf{H}_k$ hold, we need $\mathbf{Q}_k^\top \mathbf{E}_k = \mathbf{0}$.

Suppose we have a vector $\boldsymbol{q}_{k+1}$, orthogonal to all $q_i \leq k$. Then, if $\mathbf{E}_k = \boldsymbol{q}_{k+1}\boldsymbol{h}_k^\top$ for any $\boldsymbol{h}_k \in \mathbb{R}^n$. Because $\boldsymbol{q}_{k+1}$ is orthogonal to every column of $\mathbf{Q}_k^\top$, we see that $\mathbf{Q}_k^\top \mathbf{E}_k = \mathbf{Q}_k^\top \boldsymbol{q}_{k+1}\boldsymbol{h}_k^\top = \mathbf{0}$. Because this holds for any $\boldsymbol{h}_k$, we choose $\boldsymbol{h}_k$ with zeros in all positions except the k-th, where it is $h_{k+1,k}$.

So $\mathbf{A}\mathbf{Q}_k = \mathbf{Q}_k\mathbf{H}_k + \boldsymbol{q}_{k+1}\boldsymbol{h}_k^\top$, and we can write

$$\mathbf{A}\mathbf{Q}_k = \underbrace{\begin{bmatrix}\mathbf{Q}_k & \boldsymbol{q}_{k+1}\end{bmatrix}}_{\mathbf{Q}_{k+1}} \underbrace{\begin{bmatrix}\mathbf{H}_k \\ \\ \boldsymbol{h}_k^\top\end{bmatrix}}_{\mathbf{H}_{k+1,k}}, \qquad \text{where } \boldsymbol{h}_k^\top = \begin{bmatrix}0 & \cdots & 0 & h_{k+1,k}\end{bmatrix}.$$

This gives us an expression for $\boldsymbol{q}_{k+1}$ given all the previous $\boldsymbol{q}_k$'s.

2.2 Arnoldi Process

The Arnoldi algorithm is a modified version of Gram-Schmidt which finds the desired $\mathbf{Q}_k$.

In the base case of $k = 1$, we just have $\boldsymbol{q}_1 = \boldsymbol{r}_0/\|\boldsymbol{r}_0\|$.

For $k = 2$, we have

$$\mathbf{A}\boldsymbol{q}_1 = \begin{bmatrix}\boldsymbol{q}_1 & \boldsymbol{q}_2\end{bmatrix}\begin{bmatrix}h_{11} \\ h_{21}\end{bmatrix} \implies \mathbf{A}\boldsymbol{q}_1 = h_{11}\boldsymbol{q}_1 + h_{21}\boldsymbol{q}_2.$$

We apply the fact that $\mathbf{Q}_k$ must have orthonormal columns.

$$\begin{aligned}
\boldsymbol{q}_1^\top \mathbf{A}\boldsymbol{q}_1 &= h_{11}\boldsymbol{q}_1^\top \boldsymbol{q}_1 + h_{21}\boldsymbol{q}_1^\top \boldsymbol{q}_2 &\implies h_{11} &= \langle \boldsymbol{q}_1, \mathbf{A}\boldsymbol{q}_1 \rangle \\
\mathbf{A}\boldsymbol{q}_1 - h_{11}\boldsymbol{q}_1 &= h_{21}\boldsymbol{q}_2 &\implies h_{21} &= \|\mathbf{A}\boldsymbol{q}_1 - h_{11}\boldsymbol{q}_1\| \\
& &\implies \boldsymbol{q}_2 &= \frac{\mathbf{A}\boldsymbol{q}_1 - h_{11}\boldsymbol{q}_1}{h_{21}}
\end{aligned}$$

Consider the general case of $k = j$. Observe that the j-th step adds one row and one column to $\mathbf{H}_{j+1,j}$, namely the $(j+1)$-th row and j-th column. As block matrices, this can be visualised as

$$\mathbf{A}\mathbf{Q}_j = \begin{bmatrix} \mathbf{Q}_j & \boldsymbol{q}_{j+1} \end{bmatrix} \begin{bmatrix} & & & h_{1j} \\ & \mathbf{H}_{j,j-1} & & \vdots \\ & & & h_{jj} \\ 0 & \cdots & 0 & h_{j+1,j} \end{bmatrix}.$$

Because we only change the last column and row in this iteration, this can be reduced to

$$\mathbf{A}\boldsymbol{q}_j = h_{1j}\boldsymbol{q}_1 + h_{2j}\boldsymbol{q}_2 + \cdots + h_{j+1,j}\boldsymbol{q}_{j+1}.$$

Through a similar process to the $k = 2$ case, we have

$$\begin{aligned}
h_{ij} &= \langle \boldsymbol{q}_i, \mathbf{A}\boldsymbol{q}_j \rangle \quad \text{for } i = 1, \ldots, j \\
h_{j+1,j} &= \|\mathbf{A}\boldsymbol{q}_j - \textstyle\sum_{i=1}^j h_{ij}\boldsymbol{q}_i\| \\
\boldsymbol{q}_{j+1} &= \frac{\mathbf{A}\boldsymbol{q}_j - \sum_{i=1}^j h_{ij}\boldsymbol{q}_i}{h_{j+1,j}}
\end{aligned}$$

From this, we can easily derive an algorithm for the Arnoldi process. If at any point, $h_{j+1,j} = 0$, then the algorithm terminates.

2.3 Results

Theorem. *Assume the Arnoldi process does not terminate before k steps. Then the vectors $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k\}$ form an orthonormal basis for $\mathcal{K}_k(\mathbf{A}, \boldsymbol{r}_0)$.*

However, if Arnoldi breaks down at step j, then that means q_{j+1} can be written as a linear combination of the previous vectors. This should sound familiar. In fact, we have a theorem.

Theorem. *The Arnoldi process breaks down at step j, i.e. $h_{j+1,j} = 0$, if and only if the grade of $\boldsymbol{r}_0$ w.r.t. $\mathbf{A}$ is j. That is, $t(\boldsymbol{r}_0, \boldsymbol{A}) = j$.*

3 Optimality Criterion

Among infinitely many vectors, what is the "best" $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\mathbf{A}, \boldsymbol{r}_0)$?

Note. For the remainder, we assume $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{R}^n$ for simplicity. That is, everything is real-valued.

Some natural optimality conditions are

- $\boldsymbol{x}_k = \underset{\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k(\mathbf{A},\boldsymbol{r}_0)}{\arg\min} \|\boldsymbol{x} - \boldsymbol{x}^\star\|_2$.
  - However, this is not quite possible since we don't know $\boldsymbol{x}^\star$.
- For $\mathbf{A} \succ \mathbf{0}$, minimise the energy norm:

$$\boldsymbol{x}_k = \underset{\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k(\mathbf{A},\boldsymbol{r}_0)}{\arg\min} \|\boldsymbol{x} - \boldsymbol{x}^\star\|_{\mathbf{A}} = \boldsymbol{x}_k = \underset{\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k(\mathbf{A},\boldsymbol{r}_0)}{\arg\min} \frac{1}{2}\langle \boldsymbol{x}, \mathbf{A}\boldsymbol{x}\rangle - \langle \boldsymbol{b}, \boldsymbol{x}\rangle.$$

  - This is conjugate gradient (CG).
- Minimise the residual:

$$\boldsymbol{x}_k = \underset{\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k(\mathbf{A},\boldsymbol{r}_0)}{\arg\min} \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_2$$

  - For symmetric $\mathbf{A}$, this is MINRES.
  - For nonsymmetric $\mathbf{A}$, this is GMRES.

3.1 Projection Methods

However, there is one condition which can unify many aspects of the analysis in a more general way for all Krylov methods:

- Find $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\mathbf{A}, \boldsymbol{r}_0)$ such that $\boldsymbol{r}_k \perp \mathcal{L}_k$ where $\mathcal{L}_k$ is some k-dimensional subspace.

These are called *projection methods.* The intuition behind projection methods is this:

- Once a basis has been constructed, there are k degrees of freedom for picking a point in a k-dimensional affine subspace.
- So, to uniquely determine a point, we need in general k constraints.
- A typical way is to impose that the residual is orthogonal to k linearly independent vectors, e.g. the basis of another k-dimensional subspace.

Projection methods have the following ingredients: a **search subspace** $\mathcal{K}_k$, a **constraint subspace** $\mathcal{L}_k$ and the **Petrov-Galerkin conditions** $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k$ such that $\boldsymbol{r}_k \perp \mathcal{L}_k$.

If $\mathcal{K}_k = \mathcal{L}_k$, this is orthogonal projection. If $\mathcal{L}_k \neq \mathcal{K}_k$, this is oblique projection.

3.2 Imposing Conditions

Let $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{z}_k$ where $\boldsymbol{x}_k \in \mathcal{K}_k$. How can we enforce these constraints? Specifically, we need

$$\boldsymbol{z}_k \in \mathcal{K}_k \quad \text{and} \quad \langle \boldsymbol{r}_0 - \mathbf{A}\boldsymbol{z}_k, \boldsymbol{w} \rangle = 0, \ \forall \ \boldsymbol{w} \in \mathcal{L}_k.$$

Suppose $\mathbf{K}_k$ and $\mathbf{L}_k$ are bases for $\mathcal{K}_k$ and $\mathcal{L}_k$ respectively. Then,

$$\begin{aligned} \boldsymbol{z}_k \in \mathcal{K}_k &\iff \boldsymbol{z}_k = \mathbf{K}_k \boldsymbol{y}_k, \text{ for some } \boldsymbol{y}_k \in \mathbb{R}^k \\ \boldsymbol{r}_0 - \mathbf{A}\boldsymbol{z}_k \perp \mathcal{L}_k &\iff \mathbf{L}_k^\top (\boldsymbol{r}_0 - \mathbf{A}\mathbf{K}_k \boldsymbol{y}_k) = \mathbf{0} \end{aligned}$$

If $\mathbf{L}_k^\top \mathbf{A}\mathbf{K}_k$ is non-singular, we can express $\boldsymbol{x}_k$ as

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \mathbf{K}_k (\mathbf{L}_k^\top \mathbf{A}\mathbf{K}_k)^{-1} \mathbf{L}_k^\top \boldsymbol{r}_0.$$

Note that this requires $\mathbf{L}_k^\top \mathbf{A}\mathbf{K}_k$ to be non-singular which can be violated even if $\mathbf{A}$ itself is non-singular.

Proposition. $\mathbf{L}_k^\top \mathbf{A}\mathbf{K}_k$ *is non-singular if*

- $\mathbf{A} \succ \mathbf{0}$ *and* $\mathcal{L} = \mathcal{K}$*, or*
- $\det \mathbf{A} \neq 0$ *and* $\mathcal{L} = \mathbf{A}\mathcal{K}$*.*

3.3 Relation to CG and GMRES

Theorem. *The case where* $\mathbf{A} \succ \mathbf{0}$ *and* $\mathcal{L}_k = \mathcal{K}_k$ *is equivalent to CG,*

$$\boldsymbol{x}_k = \underset{\boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k(\mathbf{A}, \boldsymbol{r}_0)}{\arg\min} \|\boldsymbol{x} - \boldsymbol{x}^\star\|_{\mathbf{A}} = \boldsymbol{x}_k = \underset{\boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k(\mathbf{A}, \boldsymbol{r}_0)}{\arg\min} \frac{1}{2}\langle \boldsymbol{x}, \mathbf{A}\boldsymbol{x} \rangle - \langle \boldsymbol{b}, \boldsymbol{x} \rangle.$$

The case where $\det \mathbf{A} \neq 0$ *and* $\mathcal{L}_k = \mathbf{A}\mathcal{K}_k$ *is equivalent to MINRES/GMRES,*

$$\boldsymbol{x}_k = \underset{\boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k(\mathbf{A}, \boldsymbol{r}_0)}{\arg\min} \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_2.$$

4 Examples of Krylov Subspace Methods

We seek $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{z}_k$, where

$$\boldsymbol{z}_k \in \mathcal{K}_k \quad \text{and} \quad \boldsymbol{r}_0 - \mathbf{A}\boldsymbol{z}_k \perp \mathcal{L}_k.$$

We know what the optimal criteris is, but how do we compute the optimal $\boldsymbol{x}_k$?

Computing the optimal $\boldsymbol{x}_k$ involves subproblems depending on the type of projection method being used. Recall that $\mathbf{A}\mathbf{Q}_k = \mathbf{Q}_{k+1}\mathbf{H}_{k+1,k}$ as from Arnoldi process.

- For orthogonal projection (CG), we have $\mathcal{L}_k = \mathcal{K}_k = \operatorname{Range}\mathbf{Q}_k$ and $\boldsymbol{x}_k = \boldsymbol{x}_0 + \mathbf{Q}_k\boldsymbol{y}$. Then,

  $$\mathbf{Q}_k^\top (\boldsymbol{r}_0 - \mathbf{A}\mathbf{Q}_k\boldsymbol{y}) = \mathbf{0} \iff \mathbf{Q}_k^\top\mathbf{A}\mathbf{Q}_k\boldsymbol{y} = \mathbf{Q}_k^\top\boldsymbol{r}_0 \iff \mathbf{H}_k\boldsymbol{y} = \|\boldsymbol{r}_0\|\,\boldsymbol{e}_1.$$

  Note that the last equality holds because the first component of $\mathbf{Q}_k^\top\boldsymbol{r}_0$ is just $\langle \boldsymbol{r}_0/\|\boldsymbol{r}_0\|, \boldsymbol{r}_0\rangle$ and its columns are orthogonal so the other elements evaluate to 0.

- For oblique projection (MINRES/GMRES), $\mathcal{L}_k = \mathbf{A}\mathcal{K}_k = \mathbf{A}\operatorname{Range}\mathbf{Q}_k$ and $\boldsymbol{x}_k = \boldsymbol{x}_0 + \mathbf{Q}_k\boldsymbol{y}$. Then,

  $$(\mathbf{A}\mathbf{Q}_k)^\top(\boldsymbol{r}_0 - \mathbf{A}\mathbf{Q}_k\boldsymbol{y}) = \mathbf{0} \iff \mathbf{Q}_k^\top\mathbf{A}^\top\mathbf{A}\mathbf{Q}_k\boldsymbol{y} = \mathbf{Q}_k^\top\mathbf{A}^\top\boldsymbol{r}_0$$

  but note that this is exactly the normal equation for a least squares system. Thus,

  $$\begin{aligned}\boldsymbol{y} &= \underset{\boldsymbol{y}\in\mathbb{R}^k}{\arg\min}\frac{1}{2}\|\mathbf{A}\mathbf{Q}_k\boldsymbol{y} - \boldsymbol{r}_0\|^2\\ &= \underset{\boldsymbol{y}\in\mathbb{R}^k}{\arg\min}\frac{1}{2}\|\mathbf{H}_{k+1,k}\boldsymbol{y} - \mathbf{Q}_{k+1}^\top\boldsymbol{r}_0\|^2\\ &= \underset{\boldsymbol{y}\in\mathbb{R}^k}{\arg\min}\frac{1}{2}\Big\|\mathbf{H}_{k+1,k}\boldsymbol{y} - \|\boldsymbol{r}_0\|\boldsymbol{e}_1\Big\|^2\end{aligned}$$

When $\mathbf{A}$ is symmetric, then so is $\mathbf{Q}_k^\top\mathbf{A}\mathbf{Q}_k$. However, because this is upper Hessenberg and symmetric, it must be tridiagonal. It is denoted $\mathbf{T}_k$ instead of $\mathbf{H}_k$. In this case, the Arnoldi process is called the Lanczos process,

$$\mathbf{A}\mathbf{Q}_k = \mathbf{Q}_{k+1}\mathbf{T}_{k+1,k}.$$

The above subproblems can also be rewritten with $\mathbf{T}_k$ instead of $\mathbf{H}_k$.

In a previous section, we discussed the breakdown of the Arnoldi (or equivalently Lanczos) method. Now, we show that this actually means we have found an exact solution.

Proposition. *If Arnoldi or Lanczos breaks down at step $t = t(\mathbf{A}, \boldsymbol{r}_0)$, then the iterate $\boldsymbol{x}_t$ from any projection method onto $\mathcal{K}_k(\mathbf{A}, \boldsymbol{r}_0)$ or $\mathbf{A}\mathcal{K}_k(\mathbf{A}, \boldsymbol{r}_0)$ is the exact solution.*

For this reason, this breakdown is sometimes called a "lucky breakdown".

4.1 Minimum Residual (MINRES/GMRES)

Suppose $\mathcal{L}_k = \mathbf{A}\mathcal{K}_k$ and $\mathbf{A}$ is not necessarily symmetric. Then, the subproblem of minimising

$$\left\| \mathbf{H}_{k+1,k}\boldsymbol{y} - \|\boldsymbol{r}_0\| \, \boldsymbol{e}_1 \right\|$$

can be solved using the reduced QR factorisation $\mathbf{H}_{k+1,k} = \mathbf{U}_{k+1,k}\mathbf{R}_k$ by

$$\boldsymbol{y} = \|\boldsymbol{r}_0\| \mathbf{R}_k^{-1} \mathbf{U}_{k+1,k}^\top \boldsymbol{e}_1$$

and the residual at the k-th iteration can be computed as

$$\|\boldsymbol{b} - \mathbf{A}\boldsymbol{x}_k\| = \|\boldsymbol{r}_0\| \sqrt{1 - \|\mathbf{U}_{k+1,k}^\top \boldsymbol{e}_1\|^2}.$$

This expression for $\|\boldsymbol{r}_k\|$ is particularly useful because we do not need to compute $\mathbf{A}\boldsymbol{x}_k$. Careful inspection will show that each iteration only requires one matrix-vector product, $\mathbf{A}\boldsymbol{q}_k$.

In the case of $\mathbf{A} = \mathbf{A}^\top$ (symmetric), the method is called MINRES and

$$\left\| \mathbf{T}_{k+1,k}\boldsymbol{y} - \|\boldsymbol{r}_0\| \, \boldsymbol{e}_1 \right\|$$

can be solved similarly to above (but more efficiently, of course).

4.2 Conjugate Gradient (CG)

Here, $\mathcal{L}_k = \mathcal{K}_k$. If $\mathbf{A} \neq \mathbf{A}^\top$, this is the full orthogonalisation method (FOM).

We consider CG, the case where $\mathbf{A} \succ 0$. Solving the linear system $\mathbf{T}_k\boldsymbol{y}_k = \|\boldsymbol{r}_0\|\boldsymbol{e}_1$ via Cholesky (LDL) factorisation will, in fact, give us the celebrated CG method.

Let $\mathbf{T}_k = \mathbf{L}_k\mathbf{D}_k\mathbf{L}_k^\top$ be the Cholesky factorisation of $\mathbf{T}_k$ where $\mathbf{L}_k$ is unit lower bi-diagonal[1] and $\mathbf{D}_k$ is diagonal. Let $\boldsymbol{x}_k = \boldsymbol{x}_0 + \mathbf{Q}_k\boldsymbol{y}_k$. Firstly,

$$\begin{aligned} \mathbf{T}_k = \mathbf{Q}_k^\top \mathbf{A} \mathbf{Q}_k \text{ and } \mathbf{T}_k\boldsymbol{y}_k = \|\boldsymbol{r}_0\|\boldsymbol{e}_1 \\ \implies \boldsymbol{y}_k = \|\boldsymbol{r}_0\| \mathbf{T}_k^{-1} \boldsymbol{e}_1 \end{aligned}$$

[1] That is, with 1's on the diagonal and non-zeros on the subdiagonal.

Using this in the expression for $\boldsymbol{x}_k$,

$$\begin{aligned}
\boldsymbol{x}_k &= \boldsymbol{x}_0 + \mathbf{Q}_k \boldsymbol{y}_k = \boldsymbol{x}_0 + \mathbf{Q}_k \|\boldsymbol{r}_0\| \mathbf{T}_k^{-1} \boldsymbol{e}_1 \\
&= \boldsymbol{x}_0 + \|\boldsymbol{r}_0\| \mathbf{Q}_k (\mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^\top)^{-1} \boldsymbol{e}_1 \\
&= \boldsymbol{x}_0 + \underbrace{\mathbf{Q}_k \mathbf{L}_k^{-\top}}_{\tilde{\mathbf{P}}_k} \underbrace{\|\boldsymbol{r}_0\| \mathbf{D}_k^{-1} \mathbf{L}_k^{-1} \boldsymbol{e}_1}_{\tilde{\boldsymbol{y}}_k} \\
\boldsymbol{x}_k &= \boldsymbol{x}_0 + \tilde{\mathbf{P}}_k \tilde{\boldsymbol{y}}_k
\end{aligned}$$

Since $\mathbf{L}_k$ is unit lower bi-diagonal, we can write $\mathbf{L}_k$ and its inverse as

$$\mathbf{L}_k = \begin{bmatrix} \mathbf{L}_{k-1} & \mathbf{0} \\ \star & 1 \end{bmatrix} \implies \mathbf{L}_k^{-1} = \begin{bmatrix} \mathbf{L}_{k-1}^{-1} & \mathbf{0} \\ \star & 1 \end{bmatrix}.$$

Below, let $\boldsymbol{e}_1^k$ denote a vector in $\mathbb{R}^k$ of zeros everywhere except a 1 in the first position. A recursive expression for $\tilde{\boldsymbol{y}}_k$ is

$$\begin{aligned}
\tilde{\boldsymbol{y}}_k &= \|\boldsymbol{r}_0\| \mathbf{D}_k^{-1} \mathbf{L}_k^{-1} \boldsymbol{e}_1^k \\
&= \|\boldsymbol{r}_0\| \begin{bmatrix} \mathbf{D}_{k-1}^{-1} & \mathbf{0} \\ \mathbf{0} & d_k^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{k-1}^{-1} & \mathbf{0} \\ \star & 1 \end{bmatrix} \boldsymbol{e}_1^k \\
&= \|\boldsymbol{r}_0\| \begin{bmatrix} \mathbf{D}_{k-1}^{-1} \mathbf{L}_{k-1}^{-1} & \mathbf{0} \\ \star & d_k^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_1^{k-1} \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} \tilde{\boldsymbol{y}}_{k-1} \\ \eta_k \end{bmatrix}
\end{aligned}$$

for a particular η_k which characterises the recursion.

Similarly for $\tilde{\mathbf{P}}_k$, we have

$$\begin{aligned}
\tilde{\mathbf{P}}_k &= \mathbf{Q}_k \mathbf{L}_k^{-\top} \\
&= \begin{bmatrix} \mathbf{Q}_{k-1} & \boldsymbol{q}_k \end{bmatrix} \begin{bmatrix} \mathbf{L}_k^{-\top} & \star \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{Q}_{k-1} \mathbf{L}_{k-1}^{-\top} & \tilde{\boldsymbol{p}}_k \end{bmatrix} \\
&= \begin{bmatrix} \tilde{\mathbf{P}}_{k-1} & \tilde{\boldsymbol{p}}_k \end{bmatrix}
\end{aligned}$$

Together, we have

$$\begin{aligned}
\boldsymbol{x}_k &= \boldsymbol{x}_0 + \tilde{\mathbf{P}}_k \tilde{\boldsymbol{y}}_k \\
&= \boldsymbol{x}_0 + \begin{bmatrix} \tilde{\mathbf{P}}_{k-1} & \tilde{\boldsymbol{p}}_k \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{y}}_{k-1} \\ \eta_k \end{bmatrix} \\
&= \boldsymbol{x}_0 + \tilde{\mathbf{P}}_{k-1} \tilde{\boldsymbol{y}}_{k-1} + \eta_k \tilde{\boldsymbol{p}}_k \\
&= \boldsymbol{x}_{k-1} + \eta_k \tilde{\boldsymbol{p}}_k
\end{aligned}$$

Also, from $\tilde{\mathbf{P}}_k \mathbf{L}_k^\top = \mathbf{Q}_k$, we get $\tilde{\boldsymbol{p}}_k = \boldsymbol{q}_k - \ell_{k-1} \tilde{\boldsymbol{p}}_{k-1}$ by equating the matrices.