# Krylov Subspace Methods

Kenton Lam

September 8, 2019

This document aims 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. The lecure slides contain proofs of some theorems not proved here.

## 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,

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

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

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

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 $\alpha_k$, 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*

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

*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

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

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

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

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.

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

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

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