# Iterative Methods

## Part 1: Krylov Subspaces & Arnoldi Iterations

The algorithms we have studied so far require $\mathcal{O}(m^3)$ flops to find

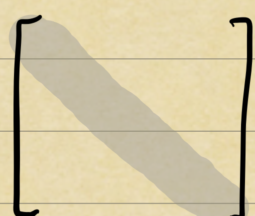
- solution of square linear system
- Least-squares soln. of overdet. lin. sys.
- Eigenvalues/eigenvectors of a matrix

For very large matrices, this is too much!

### Data-Sparse matrices

Matrices in "real world" applications often have special structure: they can be stored/represented with fewer than $\mathcal{O}(m^2)$ degrees of freedom.

**E.g.** PDE discretizations (e.g. FD, FEM, Fourier) are often banded (or "block banded")

$A =$ 

$\mathcal{O}(m)$ nonzeros clustered around diagonal

We can compute mat-vec $x \to Ax$ in $\mathcal{O}(m)$.

## Iterative methods

Rather than compute an exact soln in finitely many flops (a "direct method"), iterative methods try to construct good approximations from a small # of mat-vecs.

$$x \to \boxed{\begin{matrix}\text{black} \\ \text{box}\end{matrix}} \to Ax$$

When computing $x \to Ax$ is fast and the iterations converge rapidly, these methods can be extremely efficient.

## Power Method

We've already seen an example of an iterative method:

$$\hat{x}_k = A x_{k-1}$$
$$x_k = \hat{x}_k / \| \hat{x}_k \|$$

Usually, we have $x_k \to v$, where $Av = \underset{\uparrow \text{ largest eigenvalue}}{\lambda_1} v$

## Krylov Subspaces

The power method computes $x, Ax, A^2x, \ldots,$ but it only uses the most recent vector for approximation. Is this wasteful?

Idea: Construct best approximation to eigenvectors of $A$ from the Krylov Subspace

$$K_n(A,x) = \text{span}\{x, Ax, \ldots, A^{n-1}x\}$$

To form an approximation, project eigenvalue problem onto $K_n(A,x)$:

$$\text{Solve} \quad P(Av - \lambda v) = 0 \quad \text{s.t.} \quad Pv = v$$

$$\Rightarrow \quad PAPv = \lambda v$$

Eigenvalue problem on supspace $K_n(A,x)$ $\Rightarrow$ dimension $n \ll m$.

## Rayleigh-Ritz Projection

We can form the projector $P$ from an ONB for $K_n(A,x)$,

$$QR = \begin{bmatrix} | & | & & | \\ x & Ax & \cdots & A^{n-1}x \\ | & | & & | \end{bmatrix} \Rightarrow P = QQ^*$$

Then, $QQ^*AQQ^*v = \lambda v$. To reduce dimension

$$\underbrace{Q^*AQ}_{A_Q}\,(\underbrace{Q^*v}_{w}) = \lambda(\underbrace{Q^*v}_{w})$$

generalized Rayleigh Quotient

We can solve the $n \times n$ eigenvalue problem for $A_Q$ instead of the $m \times m$ eigenvalue problem for $A$.

## Arnoldi Iteration

Since the columns of $\begin{bmatrix} | & | & & | \\ x & Ax & \cdots & A^{n-1}x \\ | & | & & | \end{bmatrix}$ become increasingly good approximations to the dominant eigenvector, the Krylov matrix becomes increasingly ill-conditioned as the iterates progress. This leads to inaccurate Q-factor and instability.

<u>Idea:</u> Orthogonalize columns as soon as they are computed.

$$\begin{array}{ccccccccc}
x & & Ax & & A^2x & \cdots & A^3x & & \\
x & & Aq_1 & & Aq_2 & & Aq_3 & & \\
\downarrow & \nearrow & \downarrow & \nearrow & \downarrow & \nearrow & \downarrow & & \cdots \\
q_1 & & q_2 & & q_3 & & q_4 & &
\end{array}$$

Given $x$

$q_1 = x/\|x\|$

for $k = 1, 2, 3, \ldots$

$v = Aq_k$

for $j = 1, 2, \ldots, k$

$h_{jk} = q_j^* v$

$v = v - h_{jk} q_j$

$h_{k+1,k} = \|v\|$

$q_{k+1} = v/h_{k+1,k}$

Compare with Gram-Schmidt and Modified Gram-Schmidt (Lecture 9)

## Arnoldi Decomposition

$Aq_k$ is a lin. comb. of $q_{k+1}, q_k, \ldots$

$$A \begin{bmatrix} | & & | \\ q_1 & \cdots & q_n \\ | & & | \end{bmatrix} = \begin{bmatrix} | & & | & | \\ q_1 & \cdots & q_n & q_{n+1} \\ | & & | & | \end{bmatrix} \underbrace{\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & & \vdots \\ & h_{32} & \ddots & \\ & & \ddots & h_{nn} \\ & & & h_{n+1,n} \end{bmatrix}}_{\text{Hessenberg form}}$$

$$A_Q = \begin{bmatrix} - & q_1^* & - \\ & \vdots & \\ - & q_n^* & - \end{bmatrix} A \begin{bmatrix} | & & | \\ q_1 & \cdots & q_n \\ | & & | \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & \ddots & & \\ & \ddots & \ddots & \\ & & h_{n,n-1} & h_{nn} \end{bmatrix}}_{H_n}$$