MA5233 Computational Mathematics

Lecture 10: Krylov Subspace Methods: Algorithms

Simon Etter

National University of Singapore

2019/2020

Krylov Subspace Methods: Algorithms

Krylov subspace methods

A new class of algorithms for solving linear systems.

Review: LU factorisation for solving $Ax = b$

Good: black-box algorithm.

- Pass A and b, get x with errors close to machine precision without any extra input from user.

Bad: expensive!

- Typically does not scale linearly in the matrix size, even for sparse matrices.

Krylov subspace methods outperform LU factorisation for some important linear systems (when used correctly).

Krylov Subspace Methods: Algorithms

Problem statement

Given invertible $A \in \mathbb{R}^{N \times N}$ and $b \in \mathbb{R}^N$, find $x \in \mathbb{R}^N$ such that $Ax = b$.

Note that the problem dimension is denoted by N rather than n.

Subspace methods

Given $V \in \mathbb{R}^{N \times n}$, approximate x by

$$\tilde{x} := Vy \qquad \text{where} \qquad y = \arg\min \|AVy - b\|.$$

Terminology: $r := b - A\tilde{x}$ is called the *residual* of $\tilde{x}$.

Krylov subspace methods

Choose $V := \begin{pmatrix} b & Ab & \dots & A^{n-1}b \end{pmatrix} \iff \tilde{x} = \sum_{k=0}^{n-1} y_k \, A^k b$.

The approximate solution $\tilde{x}$ is thus given by

$$\tilde{x} := p_{n-1}(A)\, b \qquad \text{where} \qquad p_{n-1} := \arg\min_{p_{n-1} \in \mathcal{P}_{n-1}} \left\| \left(Ap_{n-1}(A) - I\right) b \right\|$$

and $\mathcal{P}_n := \left\{ p(x) \mid p(x) = \sum_{k=0}^{n} c_k x^k \right\}$.

Krylov Subspace Methods: Algorithms

Remarks on Krylov subspace methods

- Terminology: Krylov subspace $= \operatorname{span}\{b, Ab, \dots, A^{n-1}b\}$.
- We will discuss pros and cons of Krylov subspaces later.
  For now, let us focus on the *how* rather than the *why*.
- There are several distinct but related Krylov subspace methods.
  For now, we will focus on the Generalised Minimal Residual (GMRES) method, which solves

  $$\tilde{x} := p_{n-1}(A)\, b \qquad \text{where} \qquad p_{n-1} := \operatorname*{arg\,min}_{p_{n-1} \in \mathcal{P}_{n-1}} \big\| \big(A p_{n-1}(A) - I\big)\, b \big\|_2,$$

  i.e. GMRES minimises the two-norm of the residual.

Krylov Subspace Methods: Algorithms

Implementing GMRES, the bad way

1. Assemble $V := \begin{pmatrix} b & Ab & \dots & A^{n-1}b \end{pmatrix}$.
2. Solve least squares problem $y = \arg\min \|AVy - b\|_2$.
3. Set $\tilde{x} = Vy$.

See `10_krylov_subspace_methods.jl`.

Observation

Algorithm breaks down for $n \gtrsim 8$!
Educated guess: since algorithm works for small n, break-down is likely due to rounding errors.
The following slides will look into this more closely.

Krylov Subspace Methods: Algorithms

Recap: conditioning of least squares problem

Least-squares problem $\arg\min_x \|Ax - b\|_2$ is well-conditioned if

- columns of A are linearly independent, and
- the angle between $\mathrm{span}(A)$ and b is not too large.

Application to GMRES

Least squares problem to solve is $\arg\min_y \|AVy - b\|_2$.

Observations regarding angle:

- $\mathrm{span}(AV) = \{Ab, \dots, A^n b\}$ and b are prescribed, so there is nothing we can do about the angle between them.
- If angle is large, then least squares problem is ill-conditioned. However, in this case $\tilde{x}$ will be a bad approximation to x anyway; hence we don't care.
- In previous example, angle must be zero since $\dim(V) = N$.
- Conclusion: angle is not responsible for failure.

The following slide studies linear independence of the columns of V.

Krylov Subspace Methods: Algorithms

Why the naive implementation of GMRES breaks down

Let λ_ℓ, u_ℓ be the eigenvalues and -vectors of A, sorted such that

$$|\lambda_1| \leq |\lambda_2| \leq \ldots \leq |\lambda_N|.$$

Let $b = \sum_{\ell=1}^{N} c_\ell u_\ell$. Then,

$$A^k b = \sum_{\ell=1}^{N} c_\ell \, A^k u_\ell = \sum_{\ell=1}^{N} c_\ell \, \lambda_\ell^k u_\ell.$$

Observation: If $|\lambda_1| < |\lambda_2|$, then $\frac{|\lambda_1|}{|\lambda_2|}$ vanishes exponentially.

Conclusion:

- $\frac{A^k b}{\|A^k b\|_2}$ approaches u_N for large k.
- Columns of V become almost linearly dependent.
- $\arg\min_y \|AVy - b\|_2$ becomes ill-conditioned.

Krylov Subspace Methods: Algorithms

Key to stabilising GMRES

Find orthogonal basis for $\text{span}\{b, Ab, \ldots, A^{n-1}b\}$.

Such a basis can be found using the following algorithm.

Algorithm 1 Arnoldi iteration

```
q_0 = b/||b||_2.
for k = 0, ..., n-1 do
    q̃_{k+1} = A q_k.
    for ℓ = 0, ..., k do
        H_{ℓk} = q_ℓ^T q̃_{k+1}
        q̃_{k+1} = q̃_{k+1} − H_{ℓk} q_ℓ
    end for
    H_{k+1,k} = ||q̃_{k+1}||_2
    q_{k+1} = q̃_{k+1} / H_{k+1,k}
end for
```

Following slides list the key properties of this algorithm.

Krylov Subspace Methods: Algorithms

Lemma (Arnoldi relations)

$$AQ_n = Q_{n+1} H_n$$

where

- $Q_k = \begin{pmatrix} q_0 & \dots & q_{k-1} \end{pmatrix} \in \mathbb{R}^{N \times k}$, and
- $H_n \in \mathbb{R}^{(n+1)\times n}$ is the matrix whose entries are given in the algorithm.

Proof. Rewrite lines 3, 6, 9 in the form

$$H_{k+1,k} q_{k+1} = Aq_k - \sum_{\ell=0}^{k} H_{\ell k} q_\ell,$$

which can be rearranged to

$$Aq_k = H_{k+1,k} q_{k+1} + \sum_{\ell=0}^{k} H_{\ell,k} q_\ell.$$

Krylov Subspace Methods: Algorithms

Lemma

$$\text{span}\{q_0, \ldots, q_{n-1}\} = \text{span}\{b, Ab, \ldots, A^{n-1}b\}$$

Proof. Show by induction that $q_k = \sum_{\ell=0}^{k} c_\ell A^\ell b$ with $c_k \neq 0$:
Base: $q_0 = b/\|b\|_2 = c_0 A^0 b$.
Induction: We have

$$H_{k+1,k}\, q_{k+1} = A q_k - \sum_{\ell=0}^{k} H_{\ell,k}\, q_\ell$$

By induction hypothesis, the highest power of A in Aq_k is A^{k+1} while all other terms only go up to A^k.

Lemma

$$q_k^T q_\ell = \delta_{k\ell}$$

Proof. Arnoldi iteration is effectively the Gram-Schmidt orthogonalisation procedure applied to $b, Ab, \ldots, A^{n-1}b$.

Krylov Subspace Methods: Algorithms

Corollary

$q_0, \ldots, q_{n-1}$ is an orthogonal basis for $\mathrm{span}\{b, Ab, \ldots, A^{n-1}b\}$;
hence $\arg\min_y \|AQy - b\|_2$ is well-conditioned.

Implementing GMRES, the stable way

1. Run Arnoldi iteration to obtain Q_{n+1}, H_n.
2. Solve least squares problem $y = \arg\min \|AQ_n y - b\|_2$
3. Set $\tilde{x} = Q_n\, y$.

Assembling AQ_n is unnecessarily costly.
See next slide for how we can do better.

Krylov Subspace Methods: Algorithms

The GMRES least squares problem

Least squares problem can be rewritten as

$$\begin{aligned} y &= \arg\min \|AQ_n y - b\|_2 \\ &= \arg\min \|Q_{n+1} H_n y - b\|_2 \\ &= \arg\min \left\| H_n y - Q_{n+1}^T b \right\|_2 \qquad (Q_{n+1} \text{ is orthogonal}) \\ &= \arg\min \left\| H_n\, y - \|b\|_2\, e_1 \right\|_2 \qquad (q_0 = b/\|b\|_2,\ (e_1)_k = \delta_{k1}). \end{aligned}$$

Advantages:

- No more matrix products to assemble least squares matrix.
- $H \in \mathbb{R}^{(n+1)\times n}$ is much smaller than $AQ_n \in \mathbb{R}^{N\times n}$.
- H has special structure: $H_{ij} = 0$ if $i > j+1 \iff H = \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ & x & x & x \\ & & x & x \\ & & & x \end{pmatrix}$.

  Such matrices are called *Hessenberg*.
  QR factorisation of Hessenberg matrix can be computed in $\mathcal{O}(n^2)$ rather than $\mathcal{O}(n^3)$ FLOP.

Krylov Subspace Methods: Algorithms

Cost of Arnoldi iteration

- Line 3: n matrix-vector products.
- Lines 5, 6: $\mathcal{O}\left(Nn^2\right)$ FLOP.
  - $\mathcal{O}(N)$ FLOP per execution of either line.
  - Number of executions: $\sum_{k=0}^{n-1} \sum_{\ell=0}^{k} 1 = \sum_{k=0}^{n-1}(k+1) = \frac{n\,(n+1)}{2}$.
- Lines 8, 9: $\mathcal{O}(Nn)$ FLOP.

Summary: n matrix-vector products, $\mathcal{O}\left(Nn^2\right)$ other FLOP.

Cost of Arnoldi-based GMRES

- Arnoldi: n matrix-vector products, $\mathcal{O}\left(Nn^2\right)$ other FLOP.
- Least squares: $\mathcal{O}(n^2)$ FLOP.
- $\tilde{x} = Q_n y$: $\mathcal{O}\left(Nn\right)$ FLOP.

Summary: n matrix-vector products, $\mathcal{O}\left(Nn^2\right)$ other FLOP.

Krylov Subspace Methods: Algorithms

Summary GMRES

Given A, b and n, find $\tilde{x} := p_{n-1}(A)\, b$ where

$$p_{n-1} := \underset{p_{n-1} \in \mathcal{P}_{n-1}}{\arg\min} \left\| \left(A p_{n-1}(A) - I\right) b \right\|_2.$$

Cost: n matrix-vector products and $\mathcal{O}\left(Nn^2\right)$ other FLOP.

Discussion

- Linear scaling in N if matrix-vector product is $\mathcal{O}(N)$ (unlike LU factorisation).
- Expensive for large n due to $\mathcal{O}(Nn^2)$ FLOP for orthogonalisation.
- Good news: orthogonalisation simplifies for symmetric matrices!

Krylov Subspace Methods: Algorithms

GMRES applied to symmetric matrices

Key observation: A symmetric $\implies$ H is tridiagonal.

Proof 1. Multiplying $AQ_n = Q_{n+1}H_n$ with Q_n^T from left yields

$$Q_n^T A Q_n = \begin{pmatrix} Q_n^T Q_n & Q_n^T q_n \end{pmatrix} H_n = \begin{pmatrix} I & 0 \end{pmatrix} H_n =: \tilde{H}_n$$

($\tilde{H}_n \in \mathbb{R}^{n\times n}$ is obtained from H_n by removing last row).
$\tilde{H}_n$ is Hessenberg and symmetric $\implies$ H_n is tridiagonal.

Proof 2. By construction, $q_k = \sum_{m=0}^{k} c_m A^m b$ and

$$q_k^T \left(\sum_{m'=0}^{k'} c'_{m'} A^{m'} b \right) = 0 \qquad \text{if } k' < k.$$

Thus, for $\ell < k-1$ we obtain

$$H_{\ell k} = (Aq_k)^T q_\ell = q_k^T A q_\ell = q_k^T \left(\sum_{m=0}^{\ell} c_m A^{m+1} b \right) = 0.$$

Krylov Subspace Methods: Algorithms

GMRES applied to symmetric matrices

Consequence of observation from last slide:
we only have to orthogonalise $\tilde{q}_{k+1} = Aq_k$ against q_k and q_{k-1}.
All other inner products $q_\ell^T \tilde{q}_{k+1}$ are automatically 0.

The resulting modification of Arnoldi is known as Lanczos iteration.

Algorithm 2 Lanczos iteration

$q_0 = b/\|b\|_2$
for $k = 0, \ldots, n-1$ **do**
    $\tilde{q}_{k+1} = Aq_k$
    $H_{kk} = q_k^T \tilde{q}_{k+1}$
    if $k = 0$ **then**
        $\tilde{q}_{k+1} = \tilde{q}_{k+1} - H_{kk} q_k$
    else
        $\tilde{q}_{k+1} = \tilde{q}_{k+1} - H_{kk} q_k - H_{k-1,k} q_{k-1}$
    end if
    $H_{k+1,k} = H_{k,k+1} = \|\tilde{q}_{k+1}\|$
    $q_{k+1} = \tilde{q}_{k+1}/H_{k+1,k}$
end for

Krylov Subspace Methods: Algorithms

Terminology

GMRES applied to symmetric matrices and using Lanczos instead of Arnoldi is known as MinRes (Minimal Residual).

Discussion of MinRes

- Cost of MinRes reduces to n matrix-vector products and $\mathcal{O}(Nn)$ other FLOP.
- It is possible to interleave the Lanczos iteration and the QR factorisation of H such that only five vectors (rather than all the q_k as in GMRES) need to be stored. This is important for very large-scale computations where memory constraints are a concern.
- In practice, the q_k computed by Lanczos may fail to be orthogonal due to rounding errors. This limits the accuracy reachable with this algorithm. Some implementations include additional steps to improve orthogonality.

Krylov Subspace Methods: Algorithms

Conjugate Gradients (CG)

Let A be symmetric positive definite (SPD, $v^T A v > 0$ for all nonzero v). Conjugate gradient approximation $\tilde{x}$ to the solution to $Ax = b$ is given by

$$\tilde{x} := p_{n-1}(A)\, b \qquad \text{where} \qquad p_{n-1} := \underset{p_{n-1} \in \mathcal{P}_{n-1}}{\arg\min} \left\| \left(A p_{n-1}(A) - I \right) b \right\|_{A^{-1}}$$

and $\|v\|_{A^{-1}} := \sqrt{v^T A^{-1} v}$ (note that this is a norm since A is SPD).

Discussion

- Conjugate gradients is most well-known Krylov subspace method.
- It can be implemented using only four vectors and somewhat fewer FLOP than MinRes (but cost is still n matrix-vector products and $\mathcal{O}(Nn)$ other FLOP).
- Let $r = A\tilde{x} - b = A(\tilde{x} - x) = Ae$. Then,

$$\|r\|_{A^{-1}}^2 = r^T A^{-1} r = e^T A A^{-1} A e = e^T A e = \|e\|_A^2.$$

  Hence, CG minimises error norm $\|e\|_A$.
  Small $\|e\|_A$ is sometimes more relevant than small $\|r\|_2$.

Krylov Subspace Methods: Algorithms

Summary of Krylov subspace methods

- GMRES:

  $$\tilde{x} := p_{n-1}(A)\, b \qquad \text{where} \qquad p_{n-1} := \operatorname*{arg\,min}_{p_{n-1} \in \mathcal{P}_{n-1}} \left\| \left(A p_{n-1}(A) - I \right) b \right\|_2$$

  Cost: n matrix-vector products, $\mathcal{O}(Nn^2)$ other FLOP.
- MinRes: GMRES applied to symmetric matrix.
  Cost: n matrix-vector products, $\mathcal{O}(Nn)$ other FLOP.
- Conjugate gradients:

  $$\tilde{x} := p_{n-1}(A)\, b \qquad \text{where} \qquad p_{n-1} := \operatorname*{arg\,min}_{p_{n-1} \in \mathcal{P}_{n-1}} \left\| \left(A p_{n-1}(A) - I \right) b \right\|_{A^{-1}}$$

  Slightly cheaper than MinRes, but same cost in $\mathcal{O}$-sense.
  Only works for symmetric positive definite (SPD) matrices.

Krylov Subspace Methods: Algorithms

Discussion

- Krylov subspace methods are perhaps the most confusing topic of this module. I recommend to stick with high-level definitions as much as possible and fill in details only when needed.
- Conjugate gradient is algorithm of choice for SPD matrices. MinRes is algorithm of choice for symmetric indefinite matrices.
- $\mathcal{O}(Nn^2)$ scaling of GMRES is often a problem in practice. Many alternative algorithms exist which avoid the n^2 factor at the price of other disadvantages
    - Conjugate gradients applied to $A^T A x = A^T b$
    - Restarted GMRES
    - BiCGSTAB

Krylov Subspace Methods: Algorithms

References and further reading

Recommended since closest to presentation above:

- L. N. Trefethen and D. Bau. *Numerical Linear Algebra.* Society for Industrial and Applied Mathematics (1997),

Other references:

- G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press (1996),
- J. W. Demmel. *Applied Numerical Linear Algebra.* Society for Industrial and Applied Mathematics (1997), `doi:10.1137/1.9781611971446`
- N. J. Higham. *Accuracy and Stability of Numerical Algorithms.* Society for Industrial and Applied Mathematics (2002), `doi:10.1137/1.9780898718027`