MA3227 Numerical Analysis II

Lecture 6: GMRES

Simon Etter

National University of Singapore

2019/2020

GMRES

Introduction

LU factorisation is an all-or-nothing algorithm: we must run the full algorithm to extract any meaningful information, but when we do we get a result which is accurate up to machine precision.

Such an algorithm is called *direct*.

Over the next few weeks, we will look at methods which proceed in iterations. After each iteration, we get an approximation $x_k \approx x$, and we may terminate the algorithm if we are happy with x_k.

Such an algorithm is called *iterative*.

Iterative algorithms may perform better than LU factorisation when used correctly, but doing so can be challenging.

GMRES

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 $Ax = b$.

Subspace method

Given $V_k \in \mathbb{R}^{N \times k}$, approximate x by

$$x_k = V_k y_k \qquad \text{where} \qquad y_k = \arg\min \|AV_k y_k - b\|.$$

Terminology: $r = b - Ax_k$ is called the *residual* of x_k.

Krylov subspace method

Choose $V_k = \begin{pmatrix} b & Ab & \dots & A^{k-1}b \end{pmatrix}$.

The approximate solution x_k is then given by

$$x_k = p_{k-1}(A)\, b \qquad \text{where} \qquad p_{k-1} = \underset{p_{k-1} \in \mathcal{P}_{k-1}}{\arg\min} \left\| \big(A p_{k-1}(A) - I\big)\, b \right\|.$$

$\mathcal{P}_k = \left\{ p(x) \mid p(x) = \sum_{\ell=0}^{k} c_\ell x^\ell \right\}$ denotes the space of polynomials of degree $\leq k$.

GMRES

Remarks on Krylov subspace methods

- Terminology: Krylov subspace $= \operatorname{span}\{b, Ab, \ldots, A^{k-1}b\}$.
- 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.
  In this lecture, we will focus on the Generalised Minimal Residual (GMRES) method, which solves

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

  Note that this formula uses $\|\cdot\|_2$ while formula on previous slide uses $\|\cdot\|$.

GMRES

Implementing GMRES, the bad way

1. Assemble $V_k = \begin{pmatrix} b & Ab & \dots & A^{k-1}b \end{pmatrix}$.
2. Solve least squares problem $y_k = \arg\min \|AV_k y_k - b\|_2$.
3. Set $x_k = V_k y_k$.

See `gmres_unstable()` and `test()`.

Observation

Algorithm breaks down for $k \gtrsim 8$!

The following slides will explain why.

GMRES

Breakdown of naive GMRES

Notation and assumptions:

- Let λ_ℓ, u_ℓ be the eigenvalues and -vectors of A sorted such that $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_N|$.
  We will always assume that there are N distinct eigenvectors. Argument is similar but more technical otherwise.
- Let $c \in \mathbb{R}^N$ be such that $b = \sum_{\ell=1}^{N} c_\ell u_\ell$.
- Assume $|\lambda_1| > |\lambda_2|$ (argument is similar but more technical otherwise).

We observe:

- $A^k b = \sum_{\ell=1}^{N} c_\ell A^k u_\ell = \sum_{\ell=1}^{N} c_\ell \lambda_\ell^k u_\ell.$
- $\frac{|\lambda_\ell|^k}{|\lambda_1|^k} \to 0$ for $k \to \infty$ and all $\ell > 1$.

Hence we conclude that $\boxed{A^k b / \lambda_1^k \to c_1 u_1}$ for $k \to \infty$.

GMRES

Breakdown of naive GMRES (continued)

- $A^k b / \lambda_1^k \to c_1 u_1$ implies that the right-most columns of

$$V_k = \begin{pmatrix} b & Ab & \dots & A^{k-1}b \end{pmatrix}$$

  are almost collinear. See `normalised_krylov_vectors()`.
- We have seen in Lecture 5 (least squares) that computing $QR = AV_k$ is ill-conditioned if columns of AV_k are almost linearly dependent, i.e. rounding errors will be amplified in this case.

Armed with the above, we can now explain why `gmres_unstable()` breaks down for $k \gtrsim 8$:

- For $k \lesssim 8$, the columns of V_k are independent enough that the rounding errors remain small.
- For $k \gtrsim 8$, rounding errors are amplified to the extent that they spoil the accuracy of the result.

Question: Can we modify the GMRES algorithm to avoid excessive growth of rounding errors?

GMRES

Stabilising GMRES

Observations:

- The matrix $V_k = \begin{pmatrix} b & Ab & \dots & A^{k-1}b \end{pmatrix}$ is a bad representation for the subspace $K_k = \text{range}(V_k)$ because the columns of V_k are almost linearly dependent.
- GMRES solution x_k depends on subspace K_k but not on the matrix V_k which represents this subspace.
- The ideal representation of K_k would be an orthogonal matrix $Q_k \in \mathbb{R}^{N \times k}$ such that $K_k = \text{span}(Q_k)$ since orthogonality is the most extreme form of linear independence.
- Such a Q_k could be computed by first assembling V_k and then computing the QR factorisation of V_k, but of course this would run into the same conditioning problem as before.
- To get a numerically stable algorithm, we must interleave the steps "add column to Q_k" and "orthogonalise Q_k" as demonstrated on the next slide.

GMRES

Stabilising GMRES (continued)

Algorithm 1 Arnoldi iteration

$Q_1 = b/\|b\|_2$.
for $\ell = 1, \dots, k$ **do**
    $\tilde{q}_{\ell+1} = AQ_\ell[:, \ell]$.
    for $m = 1, \dots, \ell$ **do**
        $H[m, \ell] = Q_\ell[:, m]^T \, \tilde{q}_{\ell+1}$
        $\tilde{q}_{\ell+1} = \tilde{q}_{\ell+1} - Q_\ell[:, m] \, H[m, \ell]$
    end for
    $H[\ell + 1, \ell] = \|\tilde{q}_{k+1}\|_2$
    $Q_{\ell+1} = \left(Q_\ell \quad \middle| \quad \frac{\tilde{q}_{\ell+1}}{H[\ell+1,\ell]} \right)$
end for

This algorithm is almost the same as modified Gram-Schmidt from Lecture 5. The only substantial difference is that above we initialise $\tilde{q}_{\ell+1} = AQ_\ell[:, \ell]$ while for Gram-Schmidt we had $\tilde{q}_\ell = A[:, \ell]$.

The following slides list the key properties of this algorithm.

GMRES

Lemma

Let $Q = Q_k$ be the matrix computed by the Arnoldi iteration. Then,

$$\text{range}(Q) = \text{span}\{b, Ab, \ldots, A^{k-1}b\}$$

Proof (not examinable).

We will show by induction that for all $\ell = 1, \ldots, k$ we have

$$Q[:, \ell] = \sum_{m=0}^{\ell-1} c_m^{(\ell)} A^m b \qquad \text{with} \qquad c_{\ell-1}^{(\ell)} \neq 0.$$

The first part implies $\text{range}(Q) \subset \text{span}\{b, Ab, \ldots, A^{k-1}b\}$.
The second part guarantees that

$$A^{\ell-1}b = \tfrac{1}{c_{\ell-1}^{(\ell)}} \left(Q[:, \ell] - \sum_{m=0}^{\ell-2} c_m^{(\ell)} A^m b \right)$$

which can be used to inductively show that $A^{\ell-1}b \in \text{range}(Q)$.
The details are easy to work out, so I omit them here.

GMRES

Proof (continued).

Base: $Q[:,1] = b/\|b\|_2 = c_0^{(1)} A^0 b$ with $c_0^{(1)} = 1/\|b\|_2$.

Induction: We can rewrite lines 3, 6, 9 of the Arnolid iteration in the form

$$H[\ell+1,\ell]\, Q[:,\ell+1] = AQ[:,\ell] - \sum_{m=1}^{\ell} Q[:,m]\, H[m,\ell].$$

By induction hypothesis, the highest power of A in $AQ[:,\ell]$ is A^ℓ while all the green terms only go up to at most $A^{\ell-1}$.

This implies that $Q[:,\ell+1]$ can be written in the form

$$Q[:,\ell+1] = \sum_{m=0}^{\ell} c_m^{(\ell+1)} A^m b \qquad \text{with} \qquad c_\ell^{(\ell+1)} = \frac{c_{\ell-1}^{(\ell)}}{H[\ell+1,\ell]} \neq 0$$

as claimed.

GMRES

Lemma

Let Q_ℓ be the matrices computed by the Arnoldi iteration. Then,

$$Q_\ell^T Q_\ell = I$$

Proof. This follows straightforwardly from the discussion of the Gram-Schmidt orthogonalisation procedure in Lecture 5.

Corollary

The columns of Q_k are an orthogonal basis for $\operatorname{span}\{b, Ab, \ldots, A^{k-1}b\}$.

Implementing GMRES, the stable way

1. Run Arnoldi iteration to obtain Q_k, H_k.
2. Solve least squares problem $y_k = \arg\min \|AQ_k y_k - b\|_2$
3. Set $x_k = Q_k y_k$.

See `gmres_slow()`.

GMRES

Discussion

Assembling AQ_k in the above algorithm is expensive.

The next result shows that it is also unnecessary.

Lemma (Arnoldi relations)

$$AQ_\ell = Q_{\ell+1} \, H_\ell$$

where Q_ℓ are the matrices computed by the Arnoldi iteration and $H_\ell = H[1 : \ell + 1, 1 : \ell]$.

Proof (not examinable). As before, we rewrite lines 3, 6, 9 of the Arnoldi iteration in the form

$$H[\ell + 1, \ell] \, Q[:, \ell + 1] = AQ[:, \ell] - \sum_{m=1}^{\ell} Q[:, m] \, H[m, \ell].$$

Rearranging yields

$$AQ[:, \ell] = \sum_{m=1}^{\ell} Q[:, m] \, H[m, \ell] + Q[:, \ell + 1] \, H[\ell + 1, \ell] = Q_{\ell+1} \, H[:, \ell].$$

GMRES

Discussion

Arnoldi relations allow to rewrite the GMRES least squares problem as

$$\begin{aligned} y_k &= \arg\min \|AQ_k y_k - b\|_2 \\ &= \arg\min \|Q_{k+1} H_k y - b\|_2 \\ &= \arg\min \left\| H_k y_k - Q_{k+1}^T b \right\|_2 \\ &= \arg\min \left\| H_k y_k - \|b\|_2 \, e_1 \right\|_2 . \end{aligned}$$

On the third and fourth line, I used that $Q_{k+1}[:, 1] = b/\|b\|_2$. This immediately explains how we get from the third to the fourth line.

To see that the third equality holds, we note that

- even for a rectangular orthogonal matrix $Q_{k+1} \in \mathbb{R}^{N \times (k+1)}$ we have $\|Q_{k+1} v\|_2 = \|v\|_2$, and
- $b = Q_{k+1} Q_{k+1}^T b$ because $Q_{k+1}[:, 1] = b/\|b\|_2$ and hence $b \in \operatorname{range}(Q_{k+1})$ (recall the projector property of $Q_{k+1} Q_{k+1}^T$ from Lecture 5).

GMRES

Discussion (continued)

Advantages of rewriting

$$y_k = \arg\min \|AQ_k y_k - b\|_2 = \arg\min \Big\| H_k y_k - \|b\|_2 \, e_1 \Big\|_2.$$

- No more matrix products to assemble least squares matrix.
- $H_k \in \mathbb{R}^{(k+1)\times k}$ is much smaller than $AQ_k \in \mathbb{R}^{N\times k}$.
- H_k has special structure: $H_k[i,j] = 0$ if $i > j+1$, i.e.

$$H_k = \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ & x & x & x \\ & & x & x \\ & & & x \end{pmatrix}.$$

  Matrices of this form are called *Hessenberg*.
  QR factorisation of such H_k can be computed in $\mathcal{O}\big(k^2\big)$ operations.
  See literature for details (keyword: Householder reflectors).

See `gmres()` for final implementation of GMRES algorithm.

GMRES

Runtime of Arnoldi iteration

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

Summary: k matrix-vector products, $\mathcal{O}\big(Nk^2\big)$ other FLOP.

Runtime of Arnoldi-based GMRES

- Arnoldi: k matrix-vector products, $\mathcal{O}\big(Nk^2\big)$ other FLOP.
- Least squares: $\mathcal{O}(k^2)$ FLOP.
- $x_k = Q_k y_k$: $\mathcal{O}\big(Nk\big)$ FLOP.

Summary: k matrix-vector products, $\mathcal{O}\big(Nk^2\big)$ other FLOP.

GMRES

Discussion

GMRES runtime (copied from above):

$$k \text{ matrix-vector products, } \mathcal{O}\big(Nk^2\big) \text{ other FLOP.}$$

- GMRES runtime is $\mathcal{O}(N)$ if
  - runtime of matrix-vector product is $\mathcal{O}(N)$, and
  - sufficient accuracy can be achieve for $k = \mathcal{O}(1)$.

  The first condition is the case e.g. for sparse matrices like $\Delta_n^{(d)}$.

  We will return to the second condition in a later lecture.
- GMRES becomes expensive for large k due to the $\mathcal{O}(Nk^2)$ operations for orthogonalisation.
- Good news: orthogonalisation simplifies for symmetric matrices! See next lecture.

GMRES

Summary GMRES

- Core idea of GMRES: approximate $x = A^{-1}b$ by

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

- The above optimisation problem can be solved using linear algebra techniques. The resulting runtime is

$$k \text{ matrix-vector products}, \qquad \mathcal{O}\left(Nk^2\right) \text{ other FLOP}.$$