Lecture 14: Krylov subspace methods for least squares problems

School of Mathematical Sciences, Xiamen University

1. Conjugate gradient for least squares problems (CGLS)

- The stable way to implement CG for the normal equations is called as CGLS.

Algorithm: CGLS for $\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2$

$$
\begin{aligned}
&\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0, \quad \mathbf{p}_0 = \mathbf{A}^*\mathbf{r}_0;\\
&\textbf{for } j = 1, 2, 3, \ldots,\\
&\qquad \alpha_j = \|\mathbf{A}^*\mathbf{r}_{j-1}\|_2^2 / \|\mathbf{A}\mathbf{p}_{j-1}\|_2^2;\\
&\qquad \mathbf{x}_j = \mathbf{x}_{j-1} + \alpha_j \mathbf{p}_{j-1};\\
&\qquad \mathbf{r}_j = \mathbf{r}_{j-1} - \alpha_j \mathbf{A}\mathbf{p}_{j-1};\\
&\qquad \beta_j = \|\mathbf{A}^*\mathbf{r}_j\|_2^2 / \|\mathbf{A}^*\mathbf{r}_{j-1}\|_2^2;\\
&\qquad \mathbf{p}_j = \mathbf{A}^*\mathbf{r}_j + \beta_j \mathbf{p}_{j-1};\\
&\textbf{end}
\end{aligned}
$$

- CGLS (also called CGNR) is mathematically equivalent to LSQR, which is based on Golub–Kahan bidiagonalization for $\begin{bmatrix} \mathbf{b} & \mathbf{A} \end{bmatrix}$.

2. Householder bidiagonalization

$$
\underset{\mathbf{A}}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}}
\xrightarrow{\mathbf{U}_1^* \cdot}
\underset{\mathbf{U}_1^*\mathbf{A}}{\begin{bmatrix} \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}
\xrightarrow{\cdot \mathbf{V}_1}
\underset{\mathbf{U}_1^*\mathbf{A}\mathbf{V}_1}{\begin{bmatrix} \times & \boldsymbol{\times} & \mathbf{0} & \mathbf{0} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}
$$

$$
\xrightarrow{\mathbf{U}_2^* \cdot}
\underset{\mathbf{U}_2^*\mathbf{U}_1^*\mathbf{A}\mathbf{V}_1}{\begin{bmatrix} \times & \times & & \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}
\xrightarrow{\cdot \mathbf{V}_2}
\underset{\mathbf{U}_2^*\mathbf{U}_1^*\mathbf{A}\mathbf{V}_1\mathbf{V}_2}{\begin{bmatrix} \times & \times & & \\ & \times & \boldsymbol{\times} & \mathbf{0} \\ & & \boldsymbol{\times} & \boldsymbol{\times} \\ & & \boldsymbol{\times} & \boldsymbol{\times} \\ & & \boldsymbol{\times} & \boldsymbol{\times} \\ & & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}
$$

Proposition 1 (Case $m \geq n$)

Every matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ *has a bidiagonal decomposition:*

$$\mathbf{A} = \mathbf{U}\begin{bmatrix}\mathbf{B}\\ \mathbf{0}\end{bmatrix}\mathbf{V}^* = \mathbf{U}\begin{bmatrix}\beta_1 & \alpha_1 & & \\ & \beta_2 & \ddots & \\ & & \ddots & \alpha_{n-1}\\ & & & \beta_n \\ & & & \\ & & & \end{bmatrix}\mathbf{V}^*,$$

where $\mathbf{B} \in \mathbb{R}^{n\times n}$ *is bidiagonal,* $\alpha_i \geq 0$, $\beta_i \geq 0$, $\mathbf{U} \in \mathbb{C}^{m\times m}$ *is unitary, and*

$$\mathbf{V} = \begin{bmatrix}1 & \mathbf{0}\\ \mathbf{0} & \mathbf{Q}\end{bmatrix} \in \mathbb{C}^{n\times n}$$

is unitary.

- Note that in this proposition and in the rest of this lecture we do not consider the stability issue.

- Another bidiagonalization algorithm: Note that

$$\mathbf{A}^*\mathbf{U}_n = \mathbf{V}\mathbf{B}^*, \qquad \mathbf{A}\mathbf{V} = \mathbf{U}_n\mathbf{B},$$

i.e.,

$$\mathbf{A}^* \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix} \begin{bmatrix} \beta_1 & & & \\ \alpha_1 & \beta_2 & & \\ & \ddots & \ddots & \\ & & \alpha_{n-1} & \beta_n \end{bmatrix}$$

and

$$\mathbf{A} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \beta_1 & \alpha_1 & & \\ & \beta_2 & \ddots & \\ & & \ddots & \alpha_{n-1} \\ & & & \beta_n \end{bmatrix}.$$

Equating column i on both sides, we get

$$\mathbf{A}^*\mathbf{u}_i = \beta_i \mathbf{v}_i + \alpha_i \mathbf{v}_{i+1}, \qquad 1 \le i \le n-1;$$

and

$$\mathbf{A}\mathbf{v}_i = \alpha_{i-1}\mathbf{u}_{i-1} + \beta_i \mathbf{u}_i, \qquad 2 \le i \le n.$$

Algorithm: Golub–Kahan bidiagonalization for **A**

$\beta_1 = \|\mathbf{a}_1\|_2, \quad \mathbf{u}_1 = \mathbf{a}_1/\beta_1, \quad \mathbf{v}_1 = \mathbf{e}_1$

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

    $\mathbf{v}_{i+1} = \mathbf{A}^*\mathbf{u}_i - \beta_i \mathbf{v}_i$

    $\alpha_i = \|\mathbf{v}_{i+1}\|_2$

    $\mathbf{v}_{i+1} = \mathbf{v}_{i+1}/\alpha_i$

    $\mathbf{u}_{i+1} = \mathbf{A}\mathbf{v}_{i+1} - \alpha_i \mathbf{u}_i$

    $\beta_{i+1} = \|\mathbf{u}_{i+1}\|_2$

    $\mathbf{u}_{i+1} = \mathbf{u}_{i+1}/\beta_{i+1}$

end

---

3. LSQR

- LSQR is based on Golub–Kahan bidiagonalization for $\begin{bmatrix}\mathbf{b} & \mathbf{A}\end{bmatrix}$:

$$\mathbf{U}^* \begin{bmatrix}\mathbf{b} & \mathbf{A}\end{bmatrix} \mathbf{V} = \begin{bmatrix}\mathbf{U}^*\mathbf{b} & \mathbf{U}^*\mathbf{A}\mathbf{Q}\end{bmatrix} = \begin{bmatrix}\beta_1\mathbf{e}_1 & \widetilde{\mathbf{B}} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}$$

$$= \left[\begin{array}{c|cccc} \beta_1 & \alpha_1 & & & \\ & \beta_2 & \ddots & & \\ & & \ddots & \alpha_n & \\ & & & \beta_{n+1} & \\ & & & & \\ & & & & \end{array}\right].$$

We can write the least squares problem as

$$\min_{\mathbf{x}\in\mathbb{C}^n} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 = \min_{\mathbf{x}\in\mathbb{C}^n} \left\| \begin{bmatrix}\mathbf{b} & \mathbf{A}\end{bmatrix} \begin{bmatrix}1 \\ -\mathbf{x}\end{bmatrix} \right\|_2 = \min_{\mathbf{y}\in\mathbb{C}^n} \left\| \beta_1\mathbf{e}_1 - \widetilde{\mathbf{B}}\mathbf{y} \right\|_2.$$

Algorithm: Golub–Kahan bidiagonalization for $\begin{bmatrix} \mathbf{b} & \mathbf{A} \end{bmatrix}$

$$
\begin{aligned}
&\beta_1 = \|\mathbf{b}\|_2, \quad \mathbf{u}_1 = \mathbf{b}/\beta_1, \quad \mathbf{q}_0 = \mathbf{0} \\
&\textbf{for } i = 1, 2, 3, \ldots, \\
&\qquad \mathbf{q}_i = \mathbf{A}^*\mathbf{u}_i - \beta_i \mathbf{q}_{i-1}, \\
&\qquad \alpha_i = \|\mathbf{q}_i\|_2 \\
&\qquad \mathbf{q}_i = \mathbf{q}_i/\alpha_i \\
&\qquad \mathbf{u}_{i+1} = \mathbf{A}\mathbf{q}_i - \alpha_i \mathbf{u}_i \\
&\qquad \beta_{i+1} = \|\mathbf{u}_{i+1}\|_2 \\
&\qquad \mathbf{u}_{i+1} = \mathbf{u}_{i+1}/\beta_{i+1} \\
&\textbf{end}
\end{aligned}
$$

Proposition 2

Assume that all α_i and β_i for $1 \le i \le k$ in the above algorithm are nonzero. Then the sets $\{\mathbf{u}_i\}_{i=1}^k$ and $\{\mathbf{q}_i\}_{i=1}^k$ are orthonormal bases for $\mathcal{K}_k(\mathbf{A}\mathbf{A}^*, \mathbf{b})$ and $\mathcal{K}_k(\mathbf{A}^*\mathbf{A}, \mathbf{A}^*\mathbf{b})$, respectively.

The proof is left as an exercise.

- Define the matrices

$$\mathbf{U}_k = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_k \end{bmatrix}, \qquad \mathbf{Q}_k = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_k \end{bmatrix},$$

and

$$\widetilde{\mathbf{B}}_{k+1} = \begin{bmatrix} \alpha_1 & & \\ \beta_2 & \ddots & \\ & \ddots & \alpha_k \\ & & \beta_{k+1} \end{bmatrix} \in \mathbb{C}^{(k+1)\times k}.$$

We have

$$\mathbf{A}\mathbf{Q}_k = \mathbf{U}_{k+1}\widetilde{\mathbf{B}}_{k+1}.$$

Assume that we want to find the best approximate solution in the subspace, $\mathcal{K}_k(\mathbf{A}^*\mathbf{A}, \mathbf{A}^*\mathbf{b}) = \text{range}(\mathbf{Q}_k)$, that is

$$\begin{aligned} \min_{\mathbf{y}\in\mathbb{C}^k} \|\mathbf{b} - \mathbf{A}\mathbf{Q}_k\mathbf{y}\|_2 &= \min_{\mathbf{y}\in\mathbb{C}^k} \|\mathbf{b} - \mathbf{U}_{k+1}\widetilde{\mathbf{B}}_{k+1}\mathbf{y}\|_2 \\ &= \min_{\mathbf{y}\in\mathbb{C}^k} \|\beta_1\mathbf{e}_1 - \widetilde{\mathbf{B}}_{k+1}\mathbf{y}\|_2. \end{aligned}$$

- The least squares problem with bidiagonal structure can be solved using a sequence of Givens rotations. Consider the matrix

$$\begin{bmatrix} \widetilde{\mathbf{B}}_{k+1} & \beta_1 \mathbf{e}_1 \end{bmatrix} = \begin{bmatrix} \alpha_1 & & & & & \beta_1 \\ \beta_2 & \alpha_2 & & & & 0 \\ & \beta_3 & \alpha_3 & & & 0 \\ & & \ddots & \ddots & & \vdots \\ & & & \beta_k & \alpha_k & 0 \\ & & & & \beta_{k+1} & 0 \end{bmatrix}.$$

In the first step we zero β_2 by using a Givens rotation:

$$\begin{bmatrix} \widehat{\alpha}_1 & \widehat{\beta}_1 & & & & \gamma_1 \\ 0 & \widehat{\alpha}_2 & & & & \widehat{\gamma}_2 \\ & \beta_3 & \alpha_3 & & & 0 \\ & & \ddots & \ddots & & \vdots \\ & & & \beta_k & \alpha_k & 0 \\ & & & & \beta_{k+1} & 0 \end{bmatrix}.$$

In the next step, we zero β_3 by using a Givens rotation:

$$\begin{bmatrix} \widehat{\alpha}_1 & \widehat{\beta}_1 & & & & \gamma_1 \\ 0 & \widehat{\alpha}_2 & \widehat{\beta}_2 & & & \gamma_2 \\ & 0 & \widehat{\alpha}_3 & & & \widehat{\gamma}_3 \\ & & \beta_4 & \ddots & & \vdots \\ & & & \ddots & \alpha_k & 0 \\ & & & & \beta_{k+1} & 0 \end{bmatrix}.$$

The final result after k steps is

$$\begin{bmatrix} \widehat{\alpha}_1 & \widehat{\beta}_1 & & & & \gamma_1 \\ & \widehat{\alpha}_2 & \widehat{\beta}_2 & & & \gamma_2 \\ & & \ddots & \ddots & & \vdots \\ & & & \ddots & \widehat{\beta}_{k-1} & \gamma_{k-1} \\ & & & & \widehat{\alpha}_k & \gamma_k \\ & & & & & \widehat{\gamma}_{k+1} \end{bmatrix} := \begin{bmatrix} \widehat{\mathbf{B}}_k & \boldsymbol{\gamma}_k \\ \mathbf{0} & \widehat{\gamma}_{k+1} \end{bmatrix}.$$

Actually, we have the QR factorization for $\begin{bmatrix} \widetilde{\mathbf{B}}_{k+1} & \beta_1 \mathbf{e}_1 \end{bmatrix}$:

$$\begin{bmatrix} \widetilde{\mathbf{B}}_{k+1} & \beta_1 \mathbf{e}_1 \end{bmatrix} = \widehat{\mathbf{Q}} \begin{bmatrix} \widehat{\mathbf{B}}_k & \boldsymbol{\gamma}_k \\ \mathbf{0} & \widehat{\gamma}_{k+1} \end{bmatrix},$$

i.e.,

$$\widetilde{\mathbf{B}}_{k+1} = \widehat{\mathbf{Q}} \begin{bmatrix} \widehat{\mathbf{B}}_k \\ \mathbf{0} \end{bmatrix}, \quad \text{and} \quad \beta_1 \mathbf{e}_1 = \widehat{\mathbf{Q}} \begin{bmatrix} \boldsymbol{\gamma}_k \\ \widehat{\gamma}_{k+1} \end{bmatrix}.$$

Then we have

$$\arg \min_{\mathbf{y} \in \mathbb{C}^k} \|\beta_1 \mathbf{e}_1 - \widetilde{\mathbf{B}}_{k+1} \mathbf{y}\|_2 = \widehat{\mathbf{B}}_k^{-1} \boldsymbol{\gamma}_k$$

and

$$\min_{\mathbf{y} \in \mathbb{C}^k} \|\beta_1 \mathbf{e}_1 - \widetilde{\mathbf{B}}_{k+1} \mathbf{y}\|_2 = |\widehat{\gamma}_{k+1}|.$$

Define the matrix

$$\mathbf{W}_k := \mathbf{Q}_k \widehat{\mathbf{B}}_k^{-1} = \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_k \end{bmatrix}.$$

We have

$$\mathbf{Q}_k = \mathbf{W}_k \widehat{\mathbf{B}}_k,$$

which implies $\mathbf{w}_k = (\mathbf{q}_k - \widehat{\beta}_{k-1}\mathbf{w}_{k-1})/\widehat{\alpha}_k$. We have the recurrence

$$\begin{aligned} \mathbf{x}_k = \mathbf{Q}_k \widehat{\mathbf{B}}_k^{-1} \boldsymbol{\gamma}_k = \mathbf{W}_k \boldsymbol{\gamma}_k = \begin{bmatrix} \mathbf{W}_{k-1} & \mathbf{w}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}_{k-1} \\ \gamma_k \end{bmatrix} \\ = \mathbf{W}_{k-1} \boldsymbol{\gamma}_{k-1} + \gamma_k \mathbf{w}_k \\ = \mathbf{x}_{k-1} + \gamma_k \mathbf{w}_k. \end{aligned}$$

- C. C. Paige and M. A. Saunders.
  LSQR: An algorithm for sparse linear equations and sparse least squares.
  ACM Trans. Math. Software, 8(1):43–71, 1982.

4. Other methods

- D. C.-L. Fong and M. Saunders.
  LSMR: An iterative algorithm for sparse least-squares problems.
  SIAM J. Sci. Comput., 33(5):2950–2971, 2011.
- R. Estrin, D. Orban, and M. A. Saunders.
  LSLQ: An iterative method for linear least-squares with an error minimization property.
  SIAM J. Matrix Anal. Appl., 40(1):254–275, 2019.
- R. Estrin, D. Orban, and M. A. Saunders.
  LNLQ: An iterative method for least-norm problems with an error minimization property.
  SIAM J. Matrix Anal. Appl., 40(3):1102–1124, 2019.
- Craig's method (also called CGNE): E. J. Craig. 1955
  The N-step iteration procedures. J. Math. and Phys., 34:64–73.