

# Lecture 14: Krylov subspace methods for least squares problems



School of Mathematical Sciences, Xiamen University

# 1. Conjugate gradient for least squares problems (CGLS)

- CGLS is an implementation of CG for the normal equations.

---

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

---

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0, \quad \mathbf{p}_0 = \mathbf{A}^* \mathbf{r}_0;$

**for**  $j = 1, 2, 3, \dots,$

$\alpha_j = \|\mathbf{A}^* \mathbf{r}_{j-1}\|_2^2 / \|\mathbf{A} \mathbf{p}_{j-1}\|_2^2;$

$\mathbf{x}_j = \mathbf{x}_{j-1} + \alpha_j \mathbf{p}_{j-1};$

$\mathbf{r}_j = \mathbf{r}_{j-1} - \alpha_j \mathbf{A} \mathbf{p}_{j-1};$

$\beta_j = \|\mathbf{A}^* \mathbf{r}_j\|_2^2 / \|\mathbf{A}^* \mathbf{r}_{j-1}\|_2^2;$

$\mathbf{p}_j = \mathbf{A}^* \mathbf{r}_j + \beta_j \mathbf{p}_{j-1};$

**end**

---

- Assume that  $\mathbf{A}$  has full column rank. We have

$$\begin{aligned} \mathbf{x}_j &= \underset{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_j(\mathbf{A}^* \mathbf{A}, \mathbf{A}^* \mathbf{r}_0)}{\operatorname{argmin}} \quad \|\mathbf{A}^\dagger \mathbf{b} - \mathbf{x}\|_{\mathbf{A}^* \mathbf{A}} \\ &= \underset{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_j(\mathbf{A}^* \mathbf{A}, \mathbf{A}^* \mathbf{r}_0)}{\operatorname{argmin}} \quad \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2. \end{aligned}$$

## 2. Householder bidiagonalization

$$\begin{array}{ccc}
 \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} & \xrightarrow{\mathbf{U}_1^* \cdot} & \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix} \\
 \mathbf{A} & & \mathbf{U}_1^* \mathbf{A}
 \end{array}
 \xrightarrow{\cdot \mathbf{V}_1}
 \begin{array}{c}
 \begin{bmatrix} \times & \times & 0 & 0 \\ & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix} \\
 \mathbf{U}_1^* \mathbf{A} \mathbf{V}_1
 \end{array}$$

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

## 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

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

is unitary.

The proof is left as an exercise.

- Golub–Kahan bidiagonalization: Note that

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

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

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

and

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

---

**Algorithm:** Golub–Kahan bidiagonalization for  $\mathbf{A}$

---

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

**for**  $i = 1, 2, 3, \dots,$

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

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

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

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

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

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

**end**

---

### 3. LSQR

- Consider Householder bidiagonalization of  $\begin{bmatrix} \mathbf{b} & \mathbf{A} \end{bmatrix}$ :

$$\begin{aligned} \mathbf{U}^* \begin{bmatrix} \mathbf{b} & \mathbf{A} \end{bmatrix} \mathbf{V} &= \begin{bmatrix} \mathbf{U}^* \mathbf{b} & \mathbf{U}^* \mathbf{A} \mathbf{Q} \end{bmatrix} = \begin{bmatrix} \beta_1 \mathbf{e}_1 & \tilde{\mathbf{B}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \left[ \begin{array}{c|cccc} \beta_1 & \alpha_1 & & & \\ & \beta_2 & \ddots & & \\ & & \ddots & \alpha_n & \\ & & & \beta_{n+1} & \end{array} \right]. \end{aligned}$$

Using  $\mathbf{y} := \mathbf{Q}^* \mathbf{x}$ , we can write the least squares problem as

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

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

---

**Algorithm:** Golub–Kahan bidiagonalization for  $[\mathbf{b} \ \mathbf{A}]$ 

---

$$\beta_1 = \|\mathbf{b}\|_2, \quad \mathbf{u}_1 = \mathbf{b}/\beta_1, \quad \mathbf{q}_0 = \mathbf{0}$$

**for**  $i = 1, 2, 3, \dots$ ,

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

$$\alpha_i = \|\mathbf{q}_i\|_2$$

$$\mathbf{q}_i = \mathbf{q}_i / \alpha_i$$

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

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

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

**end**

---

## Proposition 2

*Assume that all  $\alpha_i$  and  $\beta_i$  for  $1 \leq i \leq j$  in the above algorithm are nonzero. Then the sets  $\{\mathbf{u}_i\}_{i=1}^j$  and  $\{\mathbf{q}_i\}_{i=1}^j$  are orthonormal bases for  $\mathcal{K}_j(\mathbf{A}\mathbf{A}^*, \mathbf{b})$  and  $\mathcal{K}_j(\mathbf{A}^* \mathbf{A}, \mathbf{A}^* \mathbf{b})$ , respectively.*

The proof is left as an exercise.



- Define the matrices

$$\mathbf{U}_j = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_j], \quad \mathbf{Q}_j = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \cdots \quad \mathbf{q}_j],$$

and

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

We have

$$\mathbf{A}\mathbf{Q}_j = \mathbf{U}_{j+1}\tilde{\mathbf{B}}_{j+1}.$$

At step  $j$ , LSQR seeks the best approximate solution  $\mathbf{x}_j = \mathbf{Q}_j\mathbf{y}_j$  in  $\mathcal{K}_j(\mathbf{A}^*\mathbf{A}, \mathbf{A}^*\mathbf{b})$ , where  $\mathbf{y}_j$  solves

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

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

$$\begin{bmatrix} \tilde{\mathbf{B}}_{j+1} & \beta_1 \mathbf{e}_1 \end{bmatrix} = \begin{bmatrix} \alpha_1 & & & & & \beta_1 \\ \beta_2 & \alpha_2 & & & & 0 \\ & \beta_3 & \alpha_3 & & & 0 \\ & & \ddots & \ddots & & \vdots \\ & & & \beta_j & \alpha_j & 0 \\ & & & & \beta_{j+1} & 0 \end{bmatrix}.$$

In the first step we zero  $\beta_2$  by using a Givens rotation  $\mathbf{G}_1$ :

$$\mathbf{G}_1 \begin{bmatrix} \tilde{\mathbf{B}}_{j+1} & \beta_1 \mathbf{e}_1 \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_1 & \hat{\beta}_1 & & & & \gamma_1 \\ 0 & \tilde{\alpha}_2 & & & & \hat{\gamma}_2 \\ & \beta_3 & \alpha_3 & & & 0 \\ & & \ddots & \ddots & & \vdots \\ & & & \beta_j & \alpha_j & 0 \\ & & & & \beta_{j+1} & 0 \end{bmatrix}.$$

In the next step, we zero  $\beta_3$  by using a Givens rotation  $\mathbf{G}_2$ :

$$\mathbf{G}_2 \mathbf{G}_1 \begin{bmatrix} \tilde{\mathbf{B}}_{j+1} & \beta_1 \mathbf{e}_1 \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_1 & \hat{\beta}_1 & & & & \gamma_1 \\ 0 & \hat{\alpha}_2 & \hat{\beta}_2 & & & \gamma_2 \\ & 0 & \tilde{\alpha}_3 & & & \hat{\gamma}_3 \\ & & \beta_4 & \ddots & & \vdots \\ & & & \ddots & \alpha_j & 0 \\ & & & & \beta_{j+1} & 0 \end{bmatrix}.$$

The final result after  $j$  steps is  $\mathbf{G}_j \cdots \mathbf{G}_2 \mathbf{G}_1 \begin{bmatrix} \tilde{\mathbf{B}}_{j+1} & \beta_1 \mathbf{e}_1 \end{bmatrix}$ :

$$\begin{bmatrix} \hat{\alpha}_1 & \hat{\beta}_1 & & & & \gamma_1 \\ & \hat{\alpha}_2 & \hat{\beta}_2 & & & \gamma_2 \\ & & \ddots & \ddots & & \vdots \\ & & & \ddots & \hat{\beta}_{j-1} & \gamma_{j-1} \\ & & & & \hat{\alpha}_j & \gamma_j \\ & & & & & \hat{\gamma}_{j+1} \end{bmatrix} := \begin{bmatrix} \hat{\mathbf{B}}_j & \boldsymbol{\gamma}_j \\ \mathbf{0} & \hat{\gamma}_{j+1} \end{bmatrix}.$$

Define  $\widehat{\mathbf{Q}} := \mathbf{G}_1^\top \mathbf{G}_2^\top \cdots \mathbf{G}_j^\top$ . We obtain the QR factorization:

$$\begin{bmatrix} \widetilde{\mathbf{B}}_{j+1} & \beta_1 \mathbf{e}_1 \end{bmatrix} = \widehat{\mathbf{Q}} \begin{bmatrix} \widehat{\mathbf{B}}_j & \gamma_j \\ \mathbf{0} & \widehat{\gamma}_{j+1} \end{bmatrix},$$

which implies

$$\widetilde{\mathbf{B}}_{j+1} = \widehat{\mathbf{Q}} \begin{bmatrix} \widehat{\mathbf{B}}_j \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \beta_1 \mathbf{e}_1 = \widehat{\mathbf{Q}} \begin{bmatrix} \gamma_j \\ \widehat{\gamma}_{j+1} \end{bmatrix}.$$

Thus we have

$$\mathbf{y}_j = \arg \min_{\mathbf{y} \in \mathbb{C}^j} \|\beta_1 \mathbf{e}_1 - \widetilde{\mathbf{B}}_{j+1} \mathbf{y}\|_2 = \widehat{\mathbf{B}}_j^{-1} \gamma_j$$

and

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

- Define the matrix

$$\mathbf{W}_j := \mathbf{Q}_j \hat{\mathbf{B}}_j^{-1} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_j].$$

We have

$$\mathbf{Q}_j = \mathbf{W}_j \hat{\mathbf{B}}_j,$$

which implies  $\mathbf{w}_j = (\mathbf{q}_j - \hat{\beta}_{j-1} \mathbf{w}_{j-1}) / \hat{\alpha}_j$ . We have the recurrence

$$\begin{aligned} \mathbf{x}_j &= \mathbf{Q}_j \hat{\mathbf{B}}_j^{-1} \gamma_j = \mathbf{W}_j \gamma_j = [\mathbf{W}_{j-1} \quad \mathbf{w}_j] \begin{bmatrix} \gamma_{j-1} \\ \gamma_j \end{bmatrix} \\ &= \mathbf{W}_{j-1} \gamma_{j-1} + \gamma_j \mathbf{w}_j \\ &= \mathbf{x}_{j-1} + \gamma_j \mathbf{w}_j. \end{aligned}$$

- 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.