

libcKrylov algorithm collection

Pengliang Yang
Harbin Institute of Technology, China
Email: ypl.2100@gmail.com

June 17, 2023

Algorithm 1 Conjugate gradient algorithm for solving $Ax = b$ ($A^H = A$) (Shewchuk, 1994)

```
1:  $r_0 := b - Ax_0$ 
2:  $p_0 := r_0$ 
3: for  $k = 0, 1, \dots$  until convergence do
4:    $\alpha_k = \frac{r_k^H r_k}{p_k^H A p_k}$ 
5:    $x_{k+1} = x_k + \alpha_k p_k$ 
6:    $r_{k+1} = r_k - \alpha_k A p_k$ 
7:    $\beta_k = \frac{r_{k+1}^H r_{k+1}}{r_k^H r_k}$ 
8:    $p_{k+1} = r_{k+1} + \beta_k p_k$ 
9: end for
```

Algorithm 2 CGNR for solving $A^H A x = A^H b$ (Saad, 2003, algorithm 8.4)

```
1:  $r_0 = b - Ax_0$ 
2:  $z_0 = A^H r_0$ 
3:  $p_0 = z_0$ 
4: for  $k = 0, 1, \dots$  until convergence do
5:    $w_k = A p_k$ 
6:    $\alpha_k = (z_k, z_k) / (w_k, w_k)$ 
7:    $x_{k+1} = x_k + \alpha_k p_k$ 
8:    $r_{k+1} = r_k - \alpha_k w_k$ 
9:    $z_{k+1} = A^H r_{k+1}$ 
10:   $\beta_k = (z_{k+1}, z_{k+1}) / (z_k, z_k)$ 
11:   $p_{k+1} = z_{k+1} + \beta_k p_k$ 
12: end for
```

Algorithm 3 CGNE (Craig's method) (Saad, 2003, algorithm 8.5)

```
1:  $r_0 = b - Ax_0$ 
2:  $p_0 = A^H r_0$ 
3: for  $k = 0, 1, \dots$  until convergence do
4:    $\alpha_k = (r_k, r_k) / (p_k, p_k)$ 
5:    $x_{k+1} = x_k + \alpha_k p_k$ 
6:    $r_{k+1} = r_k - \alpha_k A p_k$ 
7:    $\beta_k = (r_{k+1}, r_{k+1}) / (r_k, r_k)$ 
8:    $p_{k+1} = A^H r_{k+1} + \beta_k p_k$ 
9: end for
```

Algorithm 4 Preconditioned Conjugate gradient algorithm for solving $Ax = b$ (Saad, 2003, algorithm 9.1)

```

1:  $r_0 := b - Ax_0 = b$ 
2:  $z_0 := M^{-1}r_0$ 
3:  $p_0 := z_0$ 
4: for  $k = 0, 1, \dots$  until convergence do
5:    $\alpha_k = \frac{r_k^H p_k}{p_k^H A p_k}$ 
6:    $x_{k+1} = x_k + \alpha_k p_k$ 
7:    $r_{k+1} = r_k - \alpha_k A p_k$ 
8:    $z_{k+1} := M^{-1}r_{k+1}$ 
9:    $\beta_k = \frac{r_{k+1}^H z_{k+1}}{r_k^H z_k}$ 
10:   $p_{k+1} = z_{k+1} + \beta_k p_k$ 
11: end for

```

Algorithm 5 BiCGStab (Chen et al., 2016), improved version from (Van der Vorst, 1992)

```

1:  $r_0 = b - Ax_0$ ,  $\tilde{r}_0$  arbitrary but  $(\tilde{r}_0, r_0) \neq 0$ 
2:  $p_0 = r_0$ 
3: for  $k = 0, 1, \dots$  until convergence do
4:    $\alpha_k = (r_k, \tilde{r}_0) / (A p_k, \tilde{r}_0)$ 
5:    $s_k = r_k - \alpha_k A p_k$ 
6:    $\omega_k = (A s_k, s_k) / (A s_k, A s_k)$ 
7:    $x_{k+1} = x_k + \alpha_k p_k + \omega_k s_k$ 
8:    $r_{k+1} = s_k - \omega_k A s_k$ 
9:    $\beta_k = (r_{k+1}, \tilde{r}_0) / (r_k, \tilde{r}_0) \cdot \alpha_k / \omega_k$ 
10:   $p_{k+1} = r_{k+1} + \beta_k (p_k - \omega_k A p_k)$ 
11: end for

```

Algorithm 6 BiCGStab with right preconditioning (Flexible BiCGStab) (Chen et al., 2016)

```

1:  $r_0 = b - Ax_0$ ,  $\tilde{r}_0$  arbitrary but  $(\tilde{r}_0, r_0) \neq 0$ 
2:  $p_0 = r_0$ 
3: for  $k = 0, 1, \dots$  until convergence do
4:    $\tilde{p}_k = M^{-1}p_k$ 
5:    $\alpha_k = (r_k, \tilde{r}_0) / (A \tilde{p}_k, \tilde{r}_0)$ 
6:    $s_k = r_k - \alpha_k A \tilde{p}_k$ 
7:    $\tilde{s}_k = M^{-1}s_k$ 
8:    $\omega_k = (A \tilde{s}_k, s_k) / (A \tilde{s}_k, A \tilde{s}_k)$ 
9:    $x_{k+1} = x_k + \alpha_k p_k + \omega_k s_k$ 
10:   $r_{k+1} = s_k - \omega_k A \tilde{s}_k$ 
11:   $\beta_k = (r_{k+1}, \tilde{r}_0) / (r_k, \tilde{r}_0) \cdot \alpha_k / \omega_k$ 
12:   $p_{k+1} = r_{k+1} + \beta_k (p_k - \omega_k A \tilde{p}_k)$ 
13: end for

```

Algorithm 7 GMRES (Saad, 2003, algorithm 6.9)

```
1:  $r_0 := b - Ax_0 = b$ 
2:  $\beta = \|r_0\|_2$ 
3:  $v_1 := r_0/\beta$ 
4: for  $j = 1, \dots, m$  do
5:   compute  $w_j = Av_j$ 
6:   for  $i = 1, \dots, j$  do
7:      $h_{ij} = (w_j, v_i)$ 
8:      $w_j = w_j - h_{ij}v_i$ 
9:   end for
10:   $h_{j+1,j} = \|w_j\|_2$ . If  $h_{j+1,j} = 0$ , set  $m = j$  and go to
11:   $v_{j+1} = w_j/h_{j+1,j}$ 
12:  solve least-squares problem  $\min_y \left\| \begin{bmatrix} \beta \\ 0 \\ \dots \\ 0 \end{bmatrix} - \tilde{H}_m y \right\|_2$  by Givens rotation
13: end for
14:  $x_m = x_0 + V_m y_m$ 
```

Algorithm 8 GMRES with right preconditioning (Flexible GMRES)(Saad, 2003, algorithm 9.5)

```
1:  $r_0 := b - Ax_0 = b$ 
2:  $\beta = \|r_0\|_2$ 
3:  $v_1 := r_0/\beta$ 
4: for  $j = 1, \dots, m$  do
5:   compute  $w_j = AM^{-1}v_j$ 
6:   for  $i = 1, \dots, j$  do
7:      $h_{ij} = (w_j, v_i)$ 
8:      $w_j = w_j - h_{ij}v_i$ 
9:   end for
10:   $h_{j+1,j} = \|w_j\|_2$ . If  $h_{j+1,j} = 0$ , set  $m = j$  and go to
11:   $v_{j+1} = w_j/h_{j+1,j}$ 
12:  solve least-squares problem  $\min_y \left\| \begin{bmatrix} \beta \\ 0 \\ \dots \\ 0 \end{bmatrix} - \tilde{H}_m y \right\|_2$  by Givens rotation
13: end for
14:  $x_m = x_0 + M^{-1}V_m y_m$ 
```

References

- Chen, J., McInnes, L. C., and Zhang, H. (2016). Analysis and practical use of flexible bicgstab. *Journal of Scientific Computing*, 68(2):803–825.
- Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia.
- Shewchuk, J. R. (1994). An introduction to the conjugate gradient method without the agonizing pain. Technical Report Computer Science Technical Report CMU-CS-94-125, School of computer science, Carnegie Mellon University.
- Van der Vorst, H. A. (1992). Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13(2):631–644.