GCG-type Methods for Nonsymmetric Linear Systems

Tsung-Ming Huang

Department of Mathematics
National Taiwan Normal University

December 6, 2011

Outline

1. GCG method(Generalized Conjugate Gradient)
2. BCG method (A: unsymmetric)
3. The polynomial equivalent method of the CG method
4. Squaring the CG algorithm
5. Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems

Recall: A is s.p.d. Consider the quadratic functional

$$F(x) = \frac{1}{2}x^T Ax - x^T b$$

$$Ax^* = b \Longleftrightarrow \min_{x \in \mathbb{R}^n} F(x) = F(x^*)$$

Consider

$$\varphi(x) = \frac{1}{2}(b - Ax)^T A^{-1}(b - Ax) = F(x) + \frac{1}{2}b^T A^{-1} b, \tag{1}$$

where $\frac{1}{2}b^T A^{-1} b$ is a constant. Then

$$Ax^* = b \quad \Longleftrightarrow \quad \varphi(x^*) = \min_{x \in \mathbb{R}^n} \varphi(x) = [\min_{x \in \mathbb{R}^n} F(x)] + \frac{1}{2}b^T A^{-1} b$$

Algorithm: Conjugate Gradient method (CG-method)

Input: Given s.p.d. A, $b \in \mathbb{R}^n$ and $x_0 \in \mathbb{R}^n$ and $r_0 = b - Ax_0 = p_0$.

Set $k = 0$.
repeat
  Compute $\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$;
  Compute $x_{k+1} = x_k + \alpha_k p_k$;
  Compute $r_{k+1} = r_k - \alpha_k A p_k = b - Ax_{k+1}$;
  Compute $\beta_k = \frac{-r_{k+1}^T A p_k}{p_k^T A p_k}$;
  Compute $p_{k+1} = r_{k+1} + \beta_k p_k$;
  Set $k = k + 1$;
until $r_k = 0$

Numerator: $r_{k+1}^T((r_k - r_{k+1})/\alpha_k) = (-r_{k+1}^T r_{k+1})/\alpha_k$

Denominator: $p_k^T A p_k = (r_k^T + \beta_{k-1} p_{k-1}^T)((r_k - r_{k+1})/\alpha_k) = (r_k^T r_k)/\alpha_k$.

Remark 1

CG method does not need to compute any parameters. It only needs matrix vector and inner product of vectors. Hence it can not destroy the sparse structure of the matrix A.

The vectors r_k and p_k generated by CG-method satisfy:

$$\begin{aligned} p_i^T r_k &= (p_i, r_k) = 0, \quad i < k \\ r_i^T r_j &= (r_i, r_j) = 0, \quad i \neq j \\ p_i^T A p_j &= (p_i, A p_j) = 0, \quad i \neq j \end{aligned}$$

$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i p_i$ minimizes $F(x)$ over $x = x_0 + < p_0, \cdots, p_k >$.

GCG method(Generalized Conjugate Gradient)

GCG method is developed to minimize the residual of the linear equation under some special functional. In conjugate gradient method we take

$$\varphi(x) = \frac{1}{2}(b - Ax)^T A^{-1}(b - Ax) = \frac{1}{2} r^T A^{-1} r = \frac{1}{2}\|r\|_{A^{-1}}^2,$$

where $\|x\|_{A^{-1}} = \sqrt{x^T A^{-1} x}$.
Let A be a unsymmetric matrix. Consider the functional

$$f(x) = \frac{1}{2}(b - Ax)^T P(b - Ax),$$

where P is s.p.d. Thus $f(x) > 0$, unless $x^* = A^{-1}b \Rightarrow f(x^*) = 0$, so x^* minimizes the functional $f(x)$.

Different choices of P:

(i) $P = A^{-1}$ (A is s.p.d.) $\Rightarrow$ CG method (classical)

(ii) $P = I \;\Rightarrow$ GCR method (Generalized Conjugate Residual).

$$f(x) = \frac{1}{2}(b - Ax)^T(b - Ax) = \frac{1}{2}\|r\|_2^2$$

Here $\{r_i\}$ forms A-conjugate.

(iii) Consider $M^{-1}Ax = M^{-1}b$. Take $P = M^TM > 0 \;\Rightarrow$ GCGLS method (Generalized Conjugate Gradient Least Square).

(iv) Similar to (iii), take $P = (A + A^T)/2$ (note: P is not positive definite) and $M = (A + A^T)/2$ we get GCG method (by Concus, Golub and Widlund). In general, P is not necessary to be taken positive definite, but it must be symmetric ($P^T = P$). Therefore, the minimality property does not hold.

Let

$$(x, y)_o = x^T P y \Longrightarrow (x, y)_o = (y, x)_o.$$

Algorithm: GCG method

Input: Given A, $b \in \mathbb{R}^n$ and $x_0 \in \mathbb{R}^n$ and $r_0 = b - Ax_0 = p_0$.

Set $k = 0$.

repeat

  Compute $\alpha_k = (r_k, Ap_k)_o / (Ap_k, Ap_k)_o$;

  Compute $x_{k+1} = x_k + \alpha_k p_k$;

  Compute $r_{k+1} = r_k - \alpha_k A p_k = b - Ax_{k+1}$;

  Compute $\beta_k = -(Ar_{k+1}, Ap_i)_o / (Ap_i, Ap_i)_o$, for $i = 0, 1, \ldots, k$;

  Compute $p_{k+1} = r_{k+1} + \sum_{i=0}^{k} \beta_i^{(k)} p_i$;

  Set $k = k + 1$;

until $r_k = 0$

In GCG method, the choice of $\{\beta_i^{(k)}\}_{i=1}^{k}$ satisfy:

$$(r_{k+1}, Ap_i)_o = 0, \quad i \leq k \tag{2a}$$

$$(r_{k+1}, Ar_i)_o = 0, \quad i \leq k \tag{2b}$$

$$(Ap_i, Ap_j)_o = 0, \quad i \neq j \tag{2c}$$

Theorem 1

$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_k p_i$ *minimizes* $f(x) = \frac{1}{2}(b - Ax)^T P(b - Ax)$ *over* $x = x_0 + < p_0, \cdots, p_k >$, *where P is s.p.d.*

(The proof is the same as that of classical CG method).

If P is indefinite, which is allowed in GCG method, then the minimality property does not hold. x_{k+1} is the critical point of $f(x)$ over $x = x_0 + < p_0, \cdots, p_k >$.

Question

Can the GCG method break down? i.e., Can α_k in GCG method be zero?

Consider the numerator of α_k:

$$\begin{aligned}
(r_k, Ap_k)_o &= (r_k, Ar_k)_o \qquad \text{[by Line 7 in GCG Algorithm and (2a)]} \\
&= r_k^T P A r_k \\
&= r_k^T A^T P r_k \qquad \text{[Take transpose]} \\
&= r_k^T \frac{(PA + A^T P)}{2} r_k. \qquad (3)
\end{aligned}$$

From (3), if $(PA + A^T P)$ is positive definite, then $\alpha_k \neq 0$ unless $r_k = 0$. Hence if the matrix A satisfies $(PA + A^T P)$ positive definite, then GCG method can not break down.

From Lines 5 and 7 in GCG Algorithm, r_k and p_k can be rewritten by

$$r_k = \psi_k(A) r_0, \tag{4a}$$

$$p_k = \varphi_k(A) r_0, \tag{4b}$$

where ψ_k and φ_k are polynomials of degree $\leq k$ with $\psi_k(0) = 1$. From (4a) and (2b) follows that

$$(r_{k+1}, A^{i+1} r_0)_o = 0, \qquad i = 0, 1, \ldots, k. \tag{5}$$

From (4b) and Line 6 in GCG Algorithm, the numerator of $\beta_i^{(k)}$ can be expressed by

$$(A r_{k+1}, A p_i)_o = r_{k+1}^T A^T P A p_i = r_{k+1}^T A^T P A \varphi_i(A) r_0. \tag{6}$$

If $A^T P$ can be expressed by

$$A^T P = P\theta_s(A), \tag{7}$$

where θ_s is some polynomial of degree s. Then (6) can be written by

$$\begin{aligned} (Ar_{k+1}, Ap_i)_o &= r_{k+1}^T A^T P A \varphi_i(A) r_0 \\ &= r_{k+1}^T P \theta_s(A) A \varphi_i(A) r_0 \\ &= (r_{k+1}, A\theta_s(A)\varphi_i(A) r_0)_o. \end{aligned} \tag{8}$$

From (5) we know that if $s + i \leq k$, then (8) is zero, i.e.,$(Ar_{k+1}, Ap_i)_o = 0$. Hence

$$\beta_i^{(k)} = 0, \ i = 0, 1, \ldots, k - s.$$

But only in the special case s will be small.

For instance :

(i) In classical CG method, A is s.p.d, P is taking by A^{-1}. Then $A^T P = AA^{-1} = I = A^{-1}A = A^{-1}\theta_1(A)$, where $\theta_1(x) = x, s = 1$. So, $\beta_i^{(k)} = 0$, for all $i + 1 \le k$, it is only $\beta_k^{(k)} \ne 0$.

(ii) Concus, Golub and Widlund proposed GCG method, it solves $M^{-1}Ax = M^{-1}b$. (A: unsymmetric), where $M = (A + A^T)/2$ and $P = (A + A^T)/2$ (P may be indefinite).

- Check condition (7):

$$(M^{-1}A)^T P = A^T M^{-1} M = A^T = M(2I - M^{-1}A) = P(2I - M^{-1}A).$$

Then

$$\theta_s(M^{-1}A) = 2I - M^{-1}A,$$

where $\theta_1(x) = 2 - x,\ s = 1$. Thus $\beta_i^{(k)} = 0,\ i = 0, 1, \ldots, k-1$. Therefore we only use r_{k+1} and p_k to construct p_{k+1}.

- Check condition $A^T P + PA$:

$$(M^{-1}A)^T M + MM^{-1}A = A^T + A \quad \text{indefinite}$$

The method can possibly break down.

(iii) The other case $s = 1$ is BCG (BiCG) (See next paragraph).

Remark 2

Except the above three cases, the degree s is usually very large. That is, we need to save all directions p_i $(i = 0, 1, \ldots, k)$ in order to construct p_{k+1} satisfying the conjugate orthogonalization condition (2c). In GCG method, each iteration step needs to save $2k + 5$ vectors (x_{k+1}, r_{k+1}, p_{k+1}, $\{Ap_i\}_{i=0}^k$, $\{p_i\}_{i=0}^k$), $k + 3$ inner products (Here k is the iteration number). Hence, if k is large, then the space of storage and the computation cost can become very large and can not be acceptable. So, GCG method, in general, has some practical difficulty. Such as GCR, GMRES (by SAAD) methods, they preserve the optimality ($P > 0$), but it is too expensive (s is very large).

Modification:

(i) Restarted: If GCG method does not converge after $m+1$ iterations, then we take x_{k+1} as x_0 and restart GCG method. There are at most $2m+5$ saving vectors.

(ii) Truncated: The most expensive step of GCG method is to compute $\beta_i^{(k)},\ i=0,1,\ldots,k$ so that p_{k+1} satisfies (2c). We now release the condition (2c) to require that p_{k+1} and the nearest m direction $\{p_i\}_{i=k-m+1}^{k}$ satisfy the conjugate orthogonalization condition.

BCG method (A: unsymmetric)

BCG is similar to the CG, it does not need to save the search direction. But the norm of the residual by BCG does not preserve the minimal property.

Solve $Ax = b$ by considering $A^T y = c$ (phantom). Let

$$\tilde{A} = \begin{bmatrix} A & 0 \\ 0 & A^T \end{bmatrix}, \quad \tilde{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} b \\ c \end{bmatrix}.$$

Consider

$$\tilde{A}\tilde{x} = \tilde{b}.$$

Take $P = \begin{bmatrix} 0 & A^{-T} \\ A^{-1} & 0 \end{bmatrix}$ $(P = P^T)$. This implies

$$\tilde{A}^T P = P\tilde{A}.$$

From (7) we know that $s = 1$ for $\tilde{A}\tilde{x} = \tilde{b}$. Hence it only needs to save one direction p_k as in the classical CG method.

Apply GCG method to $\tilde{A}\tilde{x} = \tilde{b}$

Given $\tilde{x}_0 = \begin{bmatrix} x_0 \\ \hat{x}_0 \end{bmatrix}$, compute $\tilde{p}_0 = \tilde{r}_0 = \tilde{b} - \tilde{A}\tilde{x}_0 = \begin{bmatrix} r_0 \\ \hat{r}_0 \end{bmatrix}$.

Set $k = 0$.

repeat

  Compute $\alpha_k = (\tilde{r}_k, \tilde{A}\tilde{p}_k)_o / (\tilde{A}\tilde{p}_k, \tilde{A}\tilde{p}_k)_o$;

  Compute $\tilde{x}_{k+1} = \tilde{x}_k + \alpha_k \tilde{p}_k$;

  Compute $\tilde{r}_{k+1} = \tilde{r}_k - \alpha_k \tilde{A}\tilde{p}_k$;

  Compute $\beta_k = -(\tilde{A}\tilde{r}_{k+1}, \tilde{A}\tilde{p}_k)_o / (\tilde{A}\tilde{p}_k, \tilde{A}\tilde{p}_k)_o$;

  Compute $\tilde{p}_{k+1} = \tilde{r}_{k+1} + \beta_k \tilde{p}_k$;

  Set $k = k + 1$;

until $\tilde{r}_k = 0$

Simplification (BCG method)

Given x_0, compute $p_0 = r_0 = b - Ax_0$.
Choose $\hat{r}_0, \hat{p}_0 = \hat{r}_0$.
Set $k = 0$.
repeat
    Compute $\alpha_k = (\hat{r}_k, r_k)/(\hat{p}_k, Ap_k)$;
    Compute $x_{k+1} = x_k + \alpha_k p_k$;
    Compute $r_{k+1} = r_k - \alpha_k A p_k, \quad \hat{r}_{k+1} = \hat{r}_k - \alpha_k A^T \hat{p}_k$;
    Compute $\beta_k = (\hat{r}_{k+1}, r_{k+1})/(\hat{r}_k, r_k)$;
    Compute $p_{k+1} = r_{k+1} + \beta_k p_k, \quad \hat{p}_{k+1} = \hat{r}_{k+1} + \beta_k \hat{p}_k$;
    Set $k = k + 1$;
until $r_k = 0$

From above we have

$$(\tilde{A}\tilde{p}_k, \tilde{A}\tilde{p}_k)_o = [p_k^T A^T, \hat{p}_k^T A] \begin{bmatrix} 0 & A^{-T} \\ A^{-1} & 0 \end{bmatrix} \begin{bmatrix} Ap_k \\ A^T \hat{p}_k \end{bmatrix} = 2(\hat{p}_k, Ap_k).$$

BCG method satisfies the following relations:

$$r_k^T \hat{p}_i = \hat{r}_k^T p_i = 0, \quad i < k \tag{9a}$$
$$p_k^T A^T \hat{p}_i = \hat{p}_k^T A p_i = 0, \quad i < k \tag{9b}$$
$$r_k^T \hat{r}_i = \hat{r}_k^T r_i = 0, \quad i < k \tag{9c}$$

Definition 2

(9c) and (9b) are called biorthogonality and biconjugacy condition, respectively.

Property:

(i) In BCG method, the residual of the linear equation does not satisfy the minimal property, because P is taken by

$$P = \begin{pmatrix} 0 & A^{-T} \\ A^{-1} & 0 \end{pmatrix}$$

and P is symmetric, but not positive definite. The minimal value of the functional $f(x)$ may not exist.

(ii) BCG method can break down, because $Z = (\tilde{A}^T P + P\tilde{A})/2$ is not positive definite. From above discussion, α_k can be zero. But this case occurs very few.

GCG	
GCR, GCR(k)	BCG
Orthomin(k)	CGS
Orthodir	BiCGSTAB
Orthores	QMR
GMRES(m)	TFQMR
FOM	
Axelsson LS	

The polynomial equivalent method of the CG method

Consider first A is s.p.d.

CG-method

$r_0 = b - Ax_0 = p_0$.
Set $k = 0$.
repeat
  $\alpha_k = \frac{(r_k, p_k)}{(p_k, Ap_k)} = \frac{(r_k, r_k)}{(p_k, Ap_k)}$;
  $x_{k+1} = x_k + \alpha_k p_k$;
  $r_{k+1} = r_k - \alpha_k A p_k$;
  $\beta_k = \frac{-(r_{k+1}, Ap_k)}{(p_k, Ap_k)} = -\frac{(r_{k+1}, r_{k+1})}{(r_k, r_k)}$;
  $p_{k+1} = r_{k+1} + \beta_k p_k$;
  $k = k + 1$;
until $r_k = 0$

Equivalent CG-method

$r_0 = b - Ax_0 = p_0, \ p_{-1} = 1, \rho_{-1} = -1$.
Set $k = 0$.
repeat
  $\rho_k = r_k^T r_k, \ \beta_k = \frac{\rho_k}{\rho_{k-1}}$;
  $p_k = r_k + \beta_k p_{k-1}$;
  $\sigma_k = p_k^T A p_k, \ \alpha_k = \frac{\rho_k}{\sigma_k}$;
  $x_{k+1} = x_k + \alpha_k p_k$;
  $r_{k+1} = r_k - \alpha_k A p_k$;
  $k = k + 1$;
until $r_k = 0$

Remark 3

1. $E_k = r_k^T A^{-1} r_k = \min_{x \in x_0 + K_k} \|b - Ax\|_{A^{-1}}^2$
2. $r_i^T r_j = 0, \quad p_i^T A p_j = 0,\ i \neq j.$

From the structure of the new form of the CG method, we write

$$r_k = \varphi_k(A) r_0, \qquad p_k = \psi_k(A) r_0$$

where φ_k and ψ_k are polynomial of degree $\leq k$. Define $\varphi_0(\tau) \equiv 1$ and $\varphi_{-1}(\tau) \equiv 0$. Then we find

$$p_k = \varphi_k(A) r_0 + \beta_k \psi_{k-1}(A) r_0 \equiv \psi_k(A) r_0 \tag{10a}$$

with

$$\psi_k(\tau) \equiv \varphi_k(\tau) + \beta_k \psi_{k-1}(\tau), \tag{10b}$$

and

$$r_{k+1} = \varphi_k(A) r_0 - \alpha_k A \psi_k(A) r_0 \equiv \varphi_{k+1}(A) r_0 \tag{11a}$$

with

$$\varphi_{k+1}(\tau) \equiv \varphi_k(\tau) - \alpha_k \tau \psi_k(\tau). \tag{11b}$$

The polynomial equivalent method of the CG method :

$\varphi_0 \equiv 1,\ \varphi_{-1} \equiv 0,\ \rho_{-1} = 1.$
for $k = 0, 1, 2, \ldots$ **do**
  $\rho_k = (\varphi_k, \varphi_k)$, $\beta_k = \frac{\rho_k}{\rho_{k-1}}$;
  $\psi_k = \varphi_k + \beta_k \psi_{k-1}$;
  $\sigma_k = (\psi_k, \theta\psi_k)$, $\alpha_k = \frac{\rho_k}{\sigma_k}$;
  $\varphi_{k+1} = \varphi_k - \alpha_k \theta \psi_k$;
end for

where $\theta(\tau) = \tau$.

The minimization property reads

$$E_k = (\varphi_k, \theta^{-1}\varphi_k) = \min_{\varphi \in P^N} \frac{(\varphi, \theta^{-1}\varphi)}{\varphi(0)^2}.$$

We also have

$$(\varphi_i, \varphi_j) = 0, \quad i \neq j \text{ from } (r_i, r_j) = 0, \quad i \neq j.$$
$$(\psi_i, \theta\psi_j) = 0, \quad i \neq j \text{ from } (p_i, Ap_j) = 0, \quad i \neq j.$$

Theorem 3

Let $[\cdot, \cdot]$ be any symmetric bilinear form satisfying

$$[\varphi\chi, \psi] = [\varphi, \chi\psi] \quad \forall \varphi,\ \psi,\ \chi \in P^N.$$

Let the sequence of φ_i and ψ_i be constructed according to PE algorithm, but using $[\cdot, \cdot]$ instead $(\cdot, \cdot)$. Then as long as the algorithm does not break down by zero division, then φ_i and ψ_i satisfy

$$[\varphi_i, \varphi_j] = \rho_i \delta_{ij}, \qquad [\psi_i, \theta\psi_j] = \sigma_i \delta_{ij}$$

with $\theta(\tau) \equiv \tau$.

Bi-Conjugate Gradient algorithm

Given $r_0 = b - Ax_0$, $p_{-1} = \hat{p}_{-1}$ and $\hat{r}_0$ arbitrary.

for $k = 0, 1, 2, \ldots$ **do**

$\quad \rho_k = \hat{r}_k^T r_k$, $\beta_k = \rho_k / \rho_{k-1}$;

$\quad p_k = r_k + \beta_k p_{k-1}$, $\hat{p}_k = \hat{r}_k + \beta_k \hat{p}_{k-1}$;

$\quad \sigma_k = \hat{p}_k^T A p_k$, $\alpha_k = \rho_k / \sigma_k$;

$\quad r_{k+1} = r_k - \alpha_k A p_k$, $\hat{r}_{k+1} = \hat{r}_k - \alpha_k A^T \hat{p}_k$;

$\quad x_{k+1} = x_k + \alpha_k p_k$;

end for

Property:

$r_k = b - Ax_k$, $r_i^T \hat{r}_j = 0, \; i \neq j$ and $p_i^T A^T \hat{p}_j = 0, \; i \neq j$.

Consider

$$Ax = b, \quad A: \text{ nonsymmetric.}$$

Given x_0, $r_0 = b - Ax_0$, let $\hat{r}_0$ be a suitably chosen vector. Define $[\cdot, \cdot]$ by

$$[\varphi, \psi] = \hat{r}_0^T \varphi(A)\psi(A) r_0 = (\varphi(A^T)\hat{r}_0)^T \psi(A) r_0$$

and define $p_{-1} = \hat{p}_{-1} = 0$. (If A symmetric : $(\varphi, \psi) = r_0^T \varphi(A)\psi(A) r_0$). Then we have

$$\begin{aligned} r_k &= \varphi_k(A) r_0, \quad \hat{r}_k = \varphi_k(A^T)\hat{r}_0, \\ p_k &= \psi_k(A) r_0, \quad \hat{p}_k = \psi_k(A^T)\hat{r}_0 \end{aligned}$$

with φ_k and ψ_k according to (10b) and (11b). Indeed, these vectors can be produced by the Bi-Conjugate Gradient algorithm:

Squaring the CG algorithm: CGS Algorithm (SISC, 1989, Sonneveld)

Assume that Bi-CG is converging well. Then $r_k \to 0$ as $k \to \infty$. Because $r_k = \varphi_k(A) r_0$, $\varphi_k(A)$ behaves like contracting operators.

- Expect: $\varphi_k(A^T)$ behaves like contracting operators (i.e., $\hat{r}_k \to 0$). But "quasi-residuals" $\hat{r}_k$ is not exploited, they need to be computed for the ρ_k and σ_k.
- Disadvantage: Work of Bi-CG is twice the work of CG and in general $A^T v$ is not easy to compute. Especially if A is stored with a general data structure.

• Improvement: Using Polynomial equivalent algorithm to CG.
Since $\rho_k = [\varphi_k, \varphi_k]$ and $\sigma_k = [\psi_k, \theta\psi_k]$, $[\cdot, \cdot]$ has the property $[\varphi\chi, \psi] = [\varphi, \chi\psi]$. Let $\varphi_0 = 1$. Then

$$\rho_k = [\varphi_0, \varphi_k^2], \quad \sigma_k = [\varphi_0, \theta\psi_k^2].$$

$$\begin{cases} \varphi_{k+1} = \varphi_k - \alpha_k \theta \psi_k, \\ \psi_k = \varphi_k + \beta_k \psi_{k-1}. \end{cases}$$

Remark 4

$$\rho_k = \hat{r}_k^T r_k = (\varphi_k(A^T)\hat{r}_0)^T (\varphi_k(A) r_0) = \hat{r}_0^T \varphi_k^2(A) r_0,$$
$$\sigma_k = \hat{p}_k^T A p_k = (\psi_k(A^T)\hat{r}_0)^T A (\psi_k(A) r_0) = \hat{r}_0^T A \psi_k^2(A) r_0.$$

- Purpose:

1. Find an algorithm that generates the polynomial φ_k^2 and ψ_k^2 rather than φ_k and ψ_k.
2. Compute approximated solution x_k with $r_k = \varphi_k^2(A)r_0$ as residuals (try to interpret). Because $\rho_k = \hat{r}_0^T r_k$ with $r_k = \varphi_k^2(A)r_0$, $\hat{r}_k$ and $\hat{p}_k$ need not to be computed.

How to compute φ_k^2 and ψ_k^2?

$$\begin{aligned}\psi_k^2 &= [\varphi_k + \beta_k\psi_{k-1}]^2 = \varphi_k^2 + 2\beta_k\varphi_k\psi_{k-1} + \beta_k^2\psi_{k-1}^2,\\ \varphi_{k+1}^2 &= [\varphi_k - \alpha_k\theta\psi_k]^2 = \varphi_k^2 - 2\alpha_k\theta\varphi_k\psi_k + \alpha_k^2\theta^2\psi_k^2.\end{aligned}$$

Since

$$\varphi_k\psi_k = \varphi_k[\varphi_k + \beta_k\psi_{k-1}] = \varphi_k^2 + \beta_k\varphi_k\psi_{k-1},$$

we only need to compute $\varphi_k\psi_{k-1}$, φ_k^2 and ψ_k^2. Now define for $k \geq 0$:

$$\Phi_k = \varphi_k^2, \quad \Theta_k = \varphi_k\psi_{k-1}, \quad \Psi_{k-1} = \psi_{k-1}^2.$$

From

$$\begin{aligned}
\psi_k^2 &= \varphi_k^2 + 2\beta_k\varphi_k\psi_{k-1} + \beta_k^2\psi_{k-1}^2, \\
\varphi_{k+1}\psi_k &= (\varphi_k - \alpha_k\theta\psi_k)\psi_k = \varphi_k\psi_k - \alpha_k\theta\psi_k^2, \\
\varphi_{k+1}^2 &= \varphi_k^2 - 2\alpha_k\theta\varphi_k\psi_k + \alpha_k^2\theta^2\psi_k^2 \\
&= \varphi_k^2 - \alpha_k\theta(\varphi_k\psi_k - \alpha_k\theta\psi_k^2) - \alpha_k\theta\varphi_k\psi_k = \varphi_k^2 - \alpha_k\theta(\varphi_{k+1}\psi_k + \varphi_k\psi_k),
\end{aligned}$$

we have

$$\begin{aligned}
Y_k &= \varphi_k\psi_k = \Phi_k + \beta_k\Theta_k, \\
\Psi_k &= \Phi_k + 2\beta_k\Theta_k + \beta_k^2\Psi_{k-1} = Y_k + \beta_k\left(\Theta_k + \beta_k\Psi_{k-1}\right), \\
\Theta_{k+1} &= Y_k - \alpha_k\theta\Psi_k, \\
\Phi_{k+1} &= \Phi_k - \alpha_k\theta(Y_k + \Theta_{k+1}).
\end{aligned}$$

Bi-Conjugate Gradient

1: Given $r_0 = b - Ax_0$, $p_{-1} = \hat{p}_{-1}$ and arbitrary $\hat{r}_0$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad \rho_k = \hat{r}_k^T r_k$, $\beta_k = \rho_k / \rho_{k-1}$;
4: $\quad p_k = r_k + \beta_k p_{k-1}$, $\hat{p}_k = \hat{r}_k + \beta_k \hat{p}_{k-1}$;
5: $\quad \sigma_k = \hat{p}_k^T A p_k$, $\alpha_k = \rho_k / \sigma_k$;
6: $\quad r_{k+1} = r_k - \alpha_k A p_k$, $\hat{r}_{k+1} = \hat{r}_k - \alpha_k A^T \hat{p}_k$;
7: $\quad x_{k+1} = x_k + \alpha_k p_k$;
8: **end for**

CGS

1: $\Phi_0 \equiv 1$, $\Theta_0 \equiv \Psi_{-1} \equiv 0$, $\rho_{-1} = 1$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad \rho_k = [1, \Phi_k]$, $\beta_k = \rho_k / \rho_{k-1}$;
4: $\quad Y_k = \Phi_k + \beta_k \Theta_k$;
5: $\quad \Psi_k = Y_k + \beta_k(\Theta_k + \beta_k \Psi_{k-1})$;
6: $\quad \sigma_k = [1, \theta \Psi_k]$, $\alpha_k = \rho_k / \sigma_k$;
7: $\quad \Theta_{k+1} = Y_k - \alpha_k \theta \Psi_k$;
8: $\quad \Phi_{k+1} = \Phi_k - \alpha_k \theta (Y_k + \Theta_{k+1})$;
9: **end for**

CGS

$\Phi_0 \equiv 1,\ \Theta_0 \equiv \Psi_{-1} \equiv 0,$
$\rho_{-1} = 1.$
for $k = 0, 1, 2, \ldots$ **do**
  $\rho_k = [1, \Phi_k],\ \beta_k = \rho_k / \rho_{k-1};$
  $Y_k = \Phi_k + \beta_k \Theta_k;$
  $\Psi_k = Y_k + \beta_k(\Theta_k + \beta_k \Psi_{k-1});$
  $\sigma_k = [1, \theta \Psi_k],\ \alpha_k = \rho_k / \sigma_k;$
  $\Theta_{k+1} = Y_k - \alpha_k \theta \Psi_k;$
  $\Phi_{k+1} = \Phi_k - \alpha_k \theta (Y_k + \Theta_{k+1});$
end for

CGS Variant

Given $r_0 = b - Ax_0,$
$q_0 = p_{-1} = 0,\ \rho_{-1} = 1.$
for $k = 0, 1, 2, \ldots$ **do**
  $\rho_k = \hat{r}_0^T r_k,\ \beta_k = \rho_k / \rho_{k-1};$
  $u_k = r_k + \beta_k q_k;$
  $p_k = u_k + \beta_k(q_k + \beta_k p_{k-1});$
  $v_k = A p_k;$
  $\sigma_k = \hat{r}_0^T v_k,\ \alpha_k = \rho_k / \sigma_k;$
  $q_{k+1} = u_k - \alpha_k v_k;$
  $r_{k+1} = r_k - \alpha_k A(u_k + q_{k+1});$
  $x_{k+1} = x_k + \alpha_k (u_k + q_{k+1});$
end for

Define $r_k = \Phi_k(A) r_0,\ q_k = \Theta_k(A) r_0,\ p_k = \Psi_k(A) r_0$ and $u_k = Y_k(A) r_0.$

Since $r_0 = b - Ax_0$, $r_{k+1} - r_k = A(x_k - x_{k+1})$, we have that $r_k = b - Ax_k$. So this algorithm produces x_k of which the residual satisfy

$$r_k = \varphi_k^2(A) r_0.$$

Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems (SISC, 1992, Van der Vorst)

Bi-CG method

Given $r_0 = b - Ax_0$, $(\hat{r}_0, r_0) \neq 0$, $\rho_0 = 1$, $\hat{p}_0 = p_0 = 0$.
for $k = 1, 2, \ldots$ **do**
  $\rho_k = (\hat{r}_{k-1}, r_{k-1})$;
  $\beta_k = \rho_k / \rho_{k-1}$;
  $p_k = r_{k-1} + \beta_k p_{k-1}$, $\hat{p}_k = \hat{r}_{k-1} + \beta_k \hat{p}_{k-1}$;
  $v_k = Ap_k$;
  $\alpha_k = \rho_k / (\hat{p}_k, v_k)$;
  $x_k = x_{k-1} + \alpha_k p_k$;
  Stop here, if x_k is accurate enough.
  $r_k = r_{k-1} - \alpha_k v_k = r_{k-1} - \alpha_k A p_k$;
  $\hat{r}_k = \hat{r}_{k-1} - \alpha_k A^T \hat{p}_k$;
end for

Property:

(i) $r_k \perp \hat{r}_0, \ldots, \hat{r}_{k-1}$ and $\hat{r}_k \perp r_0, \ldots, r_{k-1}$.

(ii) three-term recurrence relations between $\{r_k\}$ and $\{\hat{r}_k\}$.

(iii) It terminates within n steps, but no minimal property.

Since

$$r_k^{Bi-CG} = \varphi_k(A) r_0, \quad \hat{r}_k^{Bi-CG} = \varphi_k(A^T) \hat{r}_0,$$

it implies that

$$(r_k, \hat{r}_i) = \left(\varphi_k(A) r_0, \varphi_i(A^T) \hat{r}_0\right) = (\varphi_i(A) \varphi_k(A) r_0, \hat{r}_0) = 0, \quad i < k.$$

CGS Method

Given x_0, $r_0 = b - Ax_0$, $(r_0, \hat{r}_0) \neq 0$, $\hat{r}_0 = r_0$, $\rho_0 = 1$, $p_0 = q_0 = 0$.
for $k = 1, 2, \ldots$ **do**
  $\rho_k = (\hat{r}_0, r_{k-1})$, $\beta_k = \rho_k / \rho_{k-1}$;
  $u_k = r_{k-1} + \beta_k q_{k-1}$;
  $p_k = u_k + \beta_k (q_{k-1} + \beta_k p_{k-1})$;
  $v_k = A p_k$;
  $\alpha_k = \rho_k / (\hat{r}_0, v_k)$;
  $q_k = u_k - \alpha_k v_k$;
  $w_k = u_k + q_k$;
  $x_k = x_{k-1} + \alpha_k w_k$;
  Stop here, if x_k is accurate enough.
  $r_k = r_{k-1} - \alpha_k A w_k$;
end for

We have $r_k^{\mathrm{CGS}} = \varphi_k(A)^2 r_0$.

From Bi-CG method we have $r_k^{Bi-CG} = \varphi_k(A)r_0$ and $p_{k+1} = \psi_k(A)r_0$. Thus we get

$$\psi_k(A)r_0 = \left(\varphi_k(A) + \beta_k\psi_{k-1}(A)\right) r_0,$$

and

$$\varphi_k(A)r_0 = \left(\varphi_{k-1}(A) - \alpha_k A\psi_{k-1}(A)\right) r_0,$$

where $\psi_k = \varphi_k + \beta_k\psi_{k-1}$ and $\varphi_k = \varphi_{k-1} - \alpha_k\theta\psi_{k-1}$. Since

$$\left(\varphi_k(A)r_0, \varphi_j(A^T)\hat{r}_0\right) = 0, \quad j < k,$$

it holds that

$$\varphi_k(A)r_0 \perp \hat{r}_0, A^T\hat{r}_0, \ldots, (A^T)^{k-1}\hat{r}_0$$

if and only if

$$(\tilde{\varphi}_j(A)\varphi_k(A)r_0, \hat{r}_0,) = 0$$

for some polynomial $\tilde{\varphi}_j$ of degree $j < k$ for $j = 0, 1, \ldots, k-1$.

In Bi-CG method, we take $\tilde{\varphi}_j = \varphi_j$, $\hat{r}_k = \varphi_k(A^T)\hat{r}_0$ and exploit it in CGS to get $r_k^{CGS} = \varphi_k^2(A)r_0$. Now $r_k = \tilde{\varphi}_k(A)\varphi_k(A)r_0$. How to choose $\tilde{\varphi}_k$ polynomial of degree k so that $\|r_k\|$ satisfies the minimum. Like polynomial, we can determine the optimal parameters of $\tilde{\varphi}_k$ so that $\|r_k\|$ satisfies the minimum. But the optimal parameters for the Chebychev polynomial are in general not easily obtainable. Now we take

$$\tilde{\varphi}_k \equiv \eta_k(x),$$

where

$$\eta_k(x) = (1 - \omega_1 x)(1 - \omega_2 x) \cdots (1 - \omega_k x).$$

Here ω_j are suitable constants to be selected.

Define

$$r_k = \eta_k(A)\varphi_k(A)r_0.$$

Then

$$\begin{aligned} r_k &= \eta_k(A)\varphi_k(A)r_0 \\ &= (1-\omega_k A)\eta_{k-1}(A)\left(\varphi_{k-1}(A) - \alpha_k A\psi_{k-1}(A)\right)r_0 \\ &= \left\{(\eta_{k-1}(A)\varphi_{k-1}(A) - \alpha_k A\eta_{k-1}(A)\psi_{k-1}(A))\right\}r_0 \\ &\quad -\omega_k A\left\{(\eta_{k-1}(A)\varphi_{k-1}(A) - \alpha_k A\eta_{k-1}(A)\psi_{k-1}(A))\right\}r_0 \\ &= r_{k-1} - \alpha_k Ap_k - \omega_k A(r_{k-1} - \alpha_k Ap_k) \end{aligned}$$

and

$$
\begin{aligned}
p_{k+1} &= \eta_k(A)\psi_k(A)r_0 \\
&= \eta_k(A)\left(\varphi_k(A) + \beta_k\psi_{k-1}(A)\right)r_0 \\
&= \eta_k(A)\varphi_k(A)r_0 + \beta_k(1-\omega_k A)\eta_{k-1}(A)\psi_{k-1}(A)r_0 \\
&= \eta_k(A)\varphi_k(A)r_0 + \beta_k\eta_{k-1}(A)\psi_{k-1}(A)r_0 \\
&\quad -\beta_k\omega_k A\eta_{k-1}(A)\psi_{k-1}(A)r_0 \\
&= r_k + \beta_k(p_k - \omega_k A p_k).
\end{aligned}
$$

Recover the constants ρ_k, β_k, and α_k in Bi-CG method.

We now compute β_k: Let

$$\hat{\rho}_{k+1} = (\hat{r}_0, \eta_k(A)\varphi_k(A)r_0) = \left(\eta_k(A^T)\hat{r}_0, \varphi_k(A)r_0\right).$$

From Bi-CG we have $\varphi_k(A)r_0 \perp$ all vectors $\mu_{k-1}(A^T)\hat{r}_0$, where μ_{k-1} is an arbitrary polynomial of degree $k-1$. Consider the highest order term of $\eta_k(A^T)$ (when computing $\hat{\rho}_{k+1}$) is $(-1)^k\omega_1\omega_2\cdots\omega_k(A^T)^k$. From Bi-CG method, we also have

$$\rho_{k+1} = \left(\varphi_k(A^T)\hat{r}_0, \varphi_k(A)r_0\right).$$

The highest order term of $\varphi_k(A^T)$ is $(-1)^k\alpha_1\cdots\alpha_k(A^T)^k$. Thus

$$\beta_k = (\hat{\rho}_k/\hat{\rho}_{k-1})\,(\alpha_{k-1}/\omega_{k-1}),$$

because

$$
\begin{aligned}
\beta_k &= \frac{\rho_k}{\rho_{k-1}} = \frac{(\alpha_1 \cdots \alpha_{k-1}(A^T)^{k-1}\hat{r}_0, \varphi_{k-1}(A)r_0)}{(\alpha_1 \cdots \alpha_{k-2}(A^T)^{k-2}\hat{r}_0, \varphi_{k-2}(A)r_0)} \\
&= \frac{\left(\frac{\alpha_1 \cdots \alpha_{k-1}}{\omega_1 \cdots \omega_{k-1}}\omega_1 \cdots \omega_{k-1}(A^T)^{k-1}\hat{r}_0, \varphi_{k-1}(A)r_0\right)}{\left(\frac{\alpha_1 \cdots \alpha_{k-2}}{\omega_1 \cdots \omega_{k-2}}\omega_1 \cdots \omega_{k-2}(A^T)^{k-2}\hat{r}_0, \varphi_{k-2}(A)r_0\right)} \\
&= (\hat{\rho}_k/\hat{\rho}_{k-1})\,(\alpha_{k-1}/\omega_{k-1}).
\end{aligned}
$$

Similarly, we can compute ρ_k and α_k. Let

$$
r_k = r_{k-1} - \gamma A y, \quad x_k = x_{k-1} + \gamma y \quad \text{(side product)}.
$$

Compute ω_k so that $r_k = \eta_k(A)\varphi_k(A)r_0$ is minimized in 2-norm as a function of ω_k.

Bi-CGSTAB Method

Given x_0, $r_0 = b - Ax_0$, $\hat{r}_0$ arbitrary, such that $(r_0, \hat{r}_0) \neq 0$, e.g. $\hat{r}_0 = r_0$, $\rho_0 = \alpha = \omega_0 = 1$, $v_0 = p_0 = 0$.

for $k = 1, 2, \ldots$ **do**

  $\rho_k = (\hat{r}_0, r_{k-1})$, $\beta = (\rho_k / \rho_{k-1})(\alpha / \omega_{k-1})$;

  $p_k = r_{k-1} + \beta(p_{k-1} - \omega_{k-1} v_{k-1})$;

  $v_k = Ap_k$;

  $\alpha = \rho_k / (\hat{r}_0, v_k)$;

  $s = r_{k-1} - \alpha v_k$;

  $t = As$;

  $\omega_k = (t, s)/(t, t)$;

  $x_k = x_{k-1} + \alpha p_k + \omega_k s \quad (= x_{k-1} + \alpha p_k + \omega_k (r_{k-1} - \alpha A p_k))$;

  Stop here, if x_k is accurate enough.

  $r_k = s - \omega_k t$ $(= r_{k-1} - \alpha A p_k - \omega_k A(r_{k-1} - \alpha A p_k) = r_{k-1} - A(\alpha p_k + \omega_k (r_{k-1} - \alpha A p_k))$;

end for

Preconditioned Bi-CGSTAB-P:

Rewrite $Ax = b$ as

$$\tilde{A}\tilde{x} = \tilde{b} \quad \text{with} \quad \tilde{A} = K_1^{-1} A K_2^{-1},$$

where $x = K_2^{-1}\tilde{x}$ and $\tilde{b} = K_1^{-1} b$. Then

$$\begin{aligned} \tilde{p}_k &:= K_1^{-1} p_k, \\ \tilde{v}_k &:= K_1^{-1} v_k, \\ \tilde{r}_k &:= K_1^{-1} r_k, \\ \tilde{s} &:= K_1^{-1} s, \\ \tilde{t} &:= K_1^{-1} t, \\ \tilde{x}_k &:= K_2 x_k, \\ \tilde{r}_0 &:= K_1^T \hat{r}_0. \end{aligned}$$