Biconjugate gradient method

From Wikipedia, the free encyclopedia

In mathematics, more specifically in numerical linear algebra, the **biconjugate gradient method** is an algorithm to solve systems of linear equations

$$Ax = b.$$

Unlike the conjugate gradient method, this algorithm does not require the matrix A to be self-adjoint, but instead one needs to perform multiplications by the conjugate transpose A^*.

Contents [hide]

The algorithm [edit]

1. Choose initial guess x_0, two other vectors x_0^* and b^* and a preconditioner M
2. $r_0 \leftarrow b - A\,x_0$
3. $r_0^* \leftarrow b^* - x_0^*\,A^T$
4. $p_0 \leftarrow M^{-1} r_0$
5. $p_0^* \leftarrow r_0^* M^{-1}$
6. for $k = 0, 1, \ldots$ do
   1. $\alpha_k \leftarrow \dfrac{r_k^* M^{-1} r_k}{p_k^* A p_k}$
   2. $x_{k+1} \leftarrow x_k + \alpha_k \cdot p_k$
   3. $x_{k+1}^* \leftarrow x_k^* + \overline{\alpha_k} \cdot p_k^*$
   4. $r_{k+1} \leftarrow r_k - \alpha_k \cdot A p_k$
   5. $r_{k+1}^* \leftarrow r_k^* - \overline{\alpha_k} \cdot p_k^*\,A$
   6. $\beta_k \leftarrow \dfrac{r_{k+1}^* M^{-1} r_{k+1}}{r_k^* M^{-1} r_k}$
   7. $p_{k+1} \leftarrow M^{-1} r_{k+1} + \beta_k \cdot p_k$
   8. $p_{k+1}^* \leftarrow r_{k+1}^* M^{-1} + \overline{\beta_k} \cdot p_k^*$

In the above formulation, the computed r_k and r_k^* satisfy

$$r_k = b - Ax_k,$$
$$r_k^* = b^* - x_k^*\,A$$

and thus are the respective residuals corresponding to x_k and x_k^*, as approximate solutions to the systems

$$Ax = b,$$
$$x^*\,A = b^*;$$

x^* is the adjoint, and $\overline{\alpha}$ is the complex conjugate.

Unpreconditioned version of the algorithm [edit]

1. Choose initial guess x_0,
2. $r_0 \leftarrow b - A\,x_0$
3. $\hat{r}_0 \leftarrow \hat{b} - \hat{x}_0 A^T$
4. $p_0 \leftarrow r_0$
5. $\hat{p}_0 \leftarrow \hat{r}_0$
6. for $k = 0, 1, \ldots$ do
    1. $\alpha_k \leftarrow \dfrac{\hat{r}_k r_k}{\hat{p}_k A p_k}$
    2. $x_{k+1} \leftarrow x_k + \alpha_k \cdot p_k$
    3. $\hat{x}_{k+1} \leftarrow \hat{x}_k + \alpha_k \cdot \hat{p}_k$
    4. $r_{k+1} \leftarrow r_k - \alpha_k \cdot A p_k$
    5. $\hat{r}_{k+1} \leftarrow \hat{r}_k - \alpha_k \cdot \hat{p}_k A^T$
    6. $\beta_k \leftarrow \dfrac{\hat{r}_{k+1} r_{k+1}}{\hat{r}_k r_k}$
    7. $p_{k+1} \leftarrow r_{k+1} + \beta_k \cdot p_k$
    8. $\hat{p}_{k+1} \leftarrow \hat{r}_{k+1} + \beta_k \cdot \hat{p}_k$

Discussion [edit]

The biconjugate gradient method is numerically unstable[citation needed] (compare to the biconjugate gradient stabilized method), but very important from a theoretical point of view. Define the iteration steps by

$$x_k := x_j + P_k A^{-1}\left(b - A x_j\right),$$
$$x_k^* := x_j^* + \left(b^* - x_j^* A\right) P_k A^{-1},$$

where $j < k$ using the related projection

$$P_k := \mathbf{u}_k \left(\mathbf{v}_k^* A \mathbf{u}_k\right)^{-1} \mathbf{v}_k^* A,$$

with

$$\mathbf{u}_k = \left[u_0, u_1, \ldots, u_{k-1}\right],$$
$$\mathbf{v}_k = \left[v_0, v_1, \ldots, v_{k-1}\right].$$

These related projections may be iterated themselves as

$$P_{k+1} = P_k + \left(1 - P_k\right) u_k \otimes \frac{v_k^* A \left(1 - P_k\right)}{v_k^* A \left(1 - P_k\right) u_k}.$$

A relation to Quasi-Newton methods is given by $P_k = A_k^{-1} A$ and $x_{k+1} = x_k - A_{k+1}^{-1}\left(A x_k - b\right)$, where

$$A_{k+1}^{-1} = A_k^{-1} + \left(1 - A_k^{-1} A\right) u_k \otimes \frac{v_k^* \left(1 - A A_k^{-1}\right)}{v_k^* A \left(1 - A_k^{-1} A\right) u_k}.$$

The new directions

$$p_k = \left(1 - P_k\right) u_k,$$
$$p_k^* = v_k^* A \left(1 - P_k\right) A^{-1}$$

are then orthogonal to the residuals:

$$v_i^* r_k = p_i^* r_k = 0,$$
$$r_k^* u_j = r_k^* p_j = 0,$$

which themselves satisfy

$$r_k = A \left(1 - P_k\right) A^{-1} r_j,$$
$$r_k^* = r_j^* \left(1 - P_k\right)$$

where $i, j < k$.

The biconjugate gradient method now makes a special choice and uses the setting

$$u_k = M^{-1} r_k,$$
$$v_k^* = r_k^* M^{-1}.$$

With this particular choice, explicit evaluations of P_k and A^{-1} are avoided, and the algorithm takes the form stated above.

Properties [edit]

- If $A = A^*$ is self-adjoint, $x_0^* = x_0$ and $b^* = b$, then $r_k = r_k^*$, $p_k = p_k^*$, and the conjugate gradient method produces the same sequence $x_k = x_k^*$ at half the computational cost.
- The sequences produced by the algorithm are biorthogonal, i.e., $p_i^* A p_j = r_i^* M^{-1} r_j = 0$ for $i \neq j$.
- if $P_{j'}$ is a polynomial with $\deg(P_{j'}) + j < k$, then $r_k^* P_{j'}\left(M^{-1}A\right) u_j = 0$. The algorithm thus produces projections onto the Krylov subspace.
- if $P_{i'}$ is a polynomial with $i + \deg(P_{i'}) < k$, then $v_i^* P_{i'}\left(AM^{-1}\right) r_k = 0$.

See also [edit]

- Biconjugate gradient stabilized method
- Conjugate gradient method

References [edit]

- Fletcher, R. (1976). Watson, G. Alistair, ed. "Conjugate gradient methods for indefinite systems". *Numerical Analysis*. Lecture Notes in Mathematics (Springer Berlin / Heidelberg) **506**: 73–89. doi:10.1007/BFb0080109. ISBN 978-3-540-07610-0. ISSN 1617-9692.
- Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). "Section 2.7.6". *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). New York: Cambridge University Press. ISBN 978-0-521-88068-8