Iterative methods for sparse linear systems

Marco Caliari

June 9, 2014

1 Projection methods

Given a Hilbert space H and subspaces M and L, the *projection* Px of $x \in H$ onto M orthogonally to L is defined by

$$Px \in M, \quad (x - Px, y)_H = 0 \quad \forall y \in L$$

If $L = M$, than P is called *orthogonal projection* and in this case the following is true

$$\arg\min_{y \in M} \|x - y\| = Px$$

If the projection is not orthogonal, than it is called *oblique*. Let us consider the linear system

$$Ax = b$$

whose exact solution is denoted by $\bar{x} = x_0 + \bar{\delta}$.

Proposition 1. *If A is SPD and $\mathcal{L} = \mathcal{K}$, then a vector $\tilde{x}$ if the result of an orthogonal projection method onto $\mathcal{K}$ with the starting vector x_0, that is*

$$\begin{aligned} &\tilde{x} = x_0 + \tilde{\delta}, && \tilde{\delta} \in \mathcal{K} \\ &(b - A\tilde{x}, v) = 0, && \forall v \in \mathcal{L} = \mathcal{K} \end{aligned}$$

in and only if

$$\tilde{x} = \arg\min_{x \in x_0 + \mathcal{K}} E(x)$$

where, given $x = x_0 + \delta$,

$$E(x) = (A(\bar{x} - x), \bar{x} - x)^{1/2} = (A(\bar{\delta} - \delta), \bar{\delta} - \delta)^{1/2}$$

Proof. First of all, A can be written as $A = R^T R$ (Choleski). We have

$$E(\tilde{x}) = \min_{x \in x_0 + \mathcal{K}} E(x) = \min_{\delta \in \mathcal{K}} (A(\bar{\delta} - \delta), \bar{\delta} - \delta)^{1/2} = \min_{\delta \in \mathcal{K}} (R(\bar{\delta} - \delta), R(\bar{\delta} - \delta))^{1/2} =$$
$$= \min_{\delta \in \mathcal{K}} \|R(\bar{\delta} - \delta)\|_2 = \min_{\delta \in \mathcal{K}} \|R\bar{\delta} - R\delta\|_2$$

which is taken by $\tilde{\delta}$, where $\tilde{x} = x_0 + \tilde{\delta}$. But the minimum in $R\mathcal{K}$ is taken by the orthogonal projection of $R\bar{\delta}$ onto $R\mathcal{K}$, too. Therefore $R\tilde{\delta}$ is such a projection and satisfies, for any $w = Rv$, $v \in \mathcal{K}$,

$$(R\bar{\delta} - R\tilde{\delta}, w) = 0 = (R(\bar{\delta} - \tilde{\delta}), w) = (A(\bar{\delta} - \tilde{\delta}), v) = (A(\bar{x} - \tilde{x}), v) = (b - A\tilde{x}, v)$$

□

Proposition 2. *If A is non-singular and $\mathcal{L} = A\mathcal{K}$, then a vector $\tilde{x}$ if the result of an oblique projection method onto $\mathcal{K}$ orthogonally to $\mathcal{L}$ with the starting vector x_0, that is*

$$\begin{aligned} &\tilde{x} = x_0 + \tilde{\delta}, && \tilde{\delta} \in \mathcal{K} \\ &(b - A\tilde{x}, w) = 0, && \forall w \in \mathcal{L} = A\mathcal{K} \end{aligned}$$

in and only if

$$\tilde{x} = \arg \min_{x \in x_0 + \mathcal{K}} R(x)$$

where, given $x = x_0 + \delta$,

$$R(x) = \|b - Ax\|_2 = (b - Ax, b - Ax)^{1/2} = (A(\bar{x} - x), A(\bar{x} - x))^{1/2} = (A(\bar{\delta} - \delta), A(\bar{\delta} - \delta))^{1/2}$$

Proof. We have

$$R(\tilde{x}) = \min_{x \in x_0 + \mathcal{K}} R(x) = \min_{\delta \in \mathcal{K}} (A(\bar{\delta} - \delta), A(\bar{\delta} - \delta))^{1/2} =$$
$$= \min_{\delta \in \mathcal{K}} \|A(\bar{\delta} - \delta)\|_2 = \min_{\delta \in \mathcal{K}} \|A\bar{\delta} - A\delta\|_2$$

which is taken by $\tilde{\delta}$, where $\tilde{x} = x_0 + \tilde{\delta}$. But the minimum in $A\mathcal{K} = \mathcal{L}$ is taken by the *orthogonal* projection of $A\bar{\delta}$ onto $\mathcal{L}$, too. Therefore $A\tilde{\delta}$ is such a projection and satisfies, for any $w \in \mathcal{L}$,

$$(A\bar{\delta} - A\tilde{\delta}, w) = 0 = (A(\bar{\delta} - \tilde{\delta}), w) = (A(\bar{x} - \tilde{x}), w) = (b - A\tilde{x}, w)$$

□

1.1 Conjugate Gradient (CG) method

Given a SPD matrix A of dimension n, the idea is to solve

$$A\bar{x} = b$$

by minimizing the quadratic functional

$$J(x) = x^T A x - 2b^T x$$

whose gradient is $\nabla J(x) = 2Ax - 2b = -2r(x)$. If we introduce the error

$$e(x) = x - \bar{x}$$

we have $r(x) = -Ae(x)$. Moreover, if we consider the functional

$$E(x) = e(x)^T A e(x) = r(x)^T A^{-1} r(x)$$

we have $\nabla E(x) = \nabla J(x)$ and $E(x) \geq 0$ and $E(\bar{x}) = 0$. So, the minimization of $J(x)$ is equivalent to the minimization of $E(x)$. Starting from an initial vector x_0, we can use a *descent method* to find a sequence

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

in such a way that $E(x_{k+1}) < E(x_k)$. Given p_k, we can compute an *optimal* α_k in such a way that

$$\alpha_k = \arg\min_{\alpha} E(x_k + \alpha p_k)$$

It is

$$E(x_k + \alpha p_k) = E(x_k) - 2\alpha p_k^T r_k + \alpha^2 p_k^T A p_k$$

and therefore the minimum of the parabola $E(x_k + \alpha p_k)$ is taken at

$$\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$$

Proposition 3. *If α_k is optimal, then*

$$r_{k+1}^T p_k = p_k^T r_{k+1} = 0 \tag{2}$$

Proof. First of all, we have

$$r_{k+1} = b - Ax_{k+1} = b - A(x_k + \alpha_k p_k) = r_k - \alpha_k A p_k \tag{3}$$

and then

$$r_{k+1}^T p_k = r_k^T p_k - \alpha_k p_k^T A p_k = r_k^T p_k - p_k^T r_k = 0$$

□

The equation $E(x) = E(x_k)$ is that of an ellipsoid passing through x_k, with r_k a vector orthogonal to the surface and pointing inside.

Now we would like to have (we will see later why) a sequence of directions satisfying

$$\begin{aligned} &p_0 = r_0 \\ &p_{k+1}^T A p_k = 0, \quad k \geq 0 \end{aligned}$$

In particular, it is possible to compute p_{k+1} as

$$p_{k+1} = r_{k+1} + \beta_{k+1} p_k \tag{4}$$

by taking

$$\beta_{k+1} = -\frac{r_{k+1}^T A p_k}{p_k^T A p_k}$$

Now we observe that using (2) we get

$$p_k^T r_k = r_k^T r_k + \beta_k p_{k-1}^T r_k = r_k^T r_k$$

and therefore

$$\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k} = \frac{r_k^T r_k}{p_k^T A p_k}$$

Finally, from definition (4) of p_k we have

$$A p_k = A r_k + \beta_k A p_{k-1}$$

and therefore

$$p_k^T A p_k = p_k^T A r_k = r_k^T A p_k$$

Taking expression (3) for r_{k+1}, if we multiply by r_k^T we get

$$r_{k+1}^T r_k = r_k^T r_{k+1} = r_k^T r_k - \frac{r_k^T r_k}{p_k^T A p_k} r_k^T A p_k = 0$$

and if we multiply by r_{k+1}^T we get

$$r_{k+1}^T r_{k+1} = r_{k+1}^T r_k - \frac{r_k^T r_k}{p_k^T A p_k} r_{k+1}^T A p_k = -r_k^T r_k \frac{r_{k+1}^T A p_k}{p_k^T A p_k} = r_k^T r_k \beta_{k+1}$$

from which

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

We have therefore the following implementation of the method, knowns as *Hestenes–Stiefel*

- x_0 given, $p_0 = r_0 = b - Ax_0$
- FOR $k = 0, 1, \ldots$ UNTIL $\|r_k\|_2 \leq \text{tol} \cdot \|b\|_2$

$$
\begin{aligned}
w_k &= Ap_k \\
\alpha_k &= \frac{r_k^T r_k}{p_k^T w_k} \\
x_{k+1} &= x_k + \alpha_k p_k \\
r_{k+1} &= r_k - \alpha_k w_k \\
\beta_{k+1} &= \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \\
p_{k+1} &= r_{k+1} + \beta_{k+1} p_k
\end{aligned}
$$

END

1.1.1 Some properties of the CG method

It is possible to prove the following thorem

Theorem. *For $k \geq 1$, if $r_i \neq 0$ for $0 \leq i \leq k$, then*

$$p_i^T r_k = 0 \qquad i \leq k-1 \tag{5}$$

$$p_i^T A p_k = 0 \qquad i \leq k-1 \tag{6}$$

$$r_i^T r_k = 0 \qquad i \leq k-1 \tag{7}$$

$$\text{span}\{r_0, r_1, \ldots, r_k\} = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\} \tag{8}$$

$$\text{span}\{p_0, p_1, \ldots, p_k\} = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\} \tag{9}$$

Sketch of the proof. First of all, we observe that if for a certain i it is $r_i = 0$, then x_i is the exact solution.

The proof of all properties is by induction. The basic step of each statement is easy since $p_0 = r_0$. Then, it is important to assume all the statemets true for k and prove them for $k+1$. □

Definition. *The space $\mathcal{K}_k = \text{span}\{r_0, Ar_0, \ldots, A^{k-1} r_0\}$ is called* Krylov *space.*

The set $\{r_0, r_1, \ldots, r_{k-1}\}$ is an orthogonal basis for the Krylov space. Since A is SPD, the property $p_i^T A p_k = 0$, $i \leq k-1$ means $p_i^T A p_h = 0$ for $i, h \leq k$, $i \neq h$.

Definition. *A set of vectors different from 0 and satisfying*

$$v_i^T A v_h = 0, \quad \textit{for } i, h \leq k,\ i \neq h$$

is called a set of conjugate *(with respect to A) vectors.*

By construction, the approximate solution x_k produced by the algorithm is in the space $x_0 + \mathcal{K}_k$.

Theorem. *The approximate solution x_k produced by the algorithm satisfies*

$$E(x_k) = \inf_{x \in x_0 + \mathcal{K}_k} E(x)$$

Proof. Let us take a vector $x \in x_0 + \mathcal{K}_k$. It is of the form

$$x_0 + \sum_{i=0}^{k-1} \lambda_i p_i$$

and therefore, taking into account that p_i, $i = 0, 1, \ldots, k-1$ are conjugate vectors

$$E(x) = E\left(x_0 + \sum_{i=0}^{k-1} \lambda_i p_i\right) = E(x_0) - 2\sum_{i=0}^{k-1} \lambda_i p_i^T r_0 + \sum_{i=0}^{k-1} \lambda_i^2 p_i^T A p_i$$

Now, we observe that

$$p_i^T r_0 = p_i^T (r_1 + \alpha_0 A p_0) = p_i^T r_1 = p_i^T (r_2 + \alpha_1 A p_1) = p_i^T r_2 = \ldots = p_i^T r_i$$

Therefore

$$E(x) = E(x_0) - 2\sum_{i=0}^{k-1} \lambda_i p_i^T r_i + \sum_{i=0}^{k-1} \lambda_i^2 p_i^T A p_i$$

and the minimum is taken for $\lambda_i = \alpha_i$, $i \leq k-1$. □

Therefore, the solution x_k of the CG method is the result of an orthogonal projection method onto $\mathcal{K}_k$ (see Proposition 1). This is clear also from the properties of the method, since

$$0 = r_k^T r_i = (b - Ax_k, r_i), \quad 0 \leq i \leq k-1$$

and $\{r_0, r_1, \ldots, r_{k-1}\}$ is a basis for $\mathcal{K}_k$.

Proposition 4. *A set of conjugate vectors is a set of linear independent vectors.*

Proof. Let us suppose that

$$\sum_{i=1}^{k} c_i v_i = 0$$

with $c_j \neq 0$. Then

$$\left(\sum_{i=1}^{k} c_i v_i\right)^T A v_j = 0 = \sum_{i=1}^{k} c_i (v_i^T A v_j) = c_j v_j A^T v_j$$

Since A is SPD, the result cannot be 0, unless $v_j = 0$ (absurd). □

Proposition 5. *The CG algorithm converges in n iterations at maximum.*

Proof. The Krylov space $\mathcal{K}_k = \{p_0, p_1, \ldots, p_{k-1}\}$ has dimension n at maximum. $\square$

In practice, since it is not possible to compute truly conjugate directions in machine arithmetic, usually the CG algorithm is used as an iterative method (and it is sometimes called *semiiterative* method).

It is possible to prove the following convergence estimate

$$|||E_k||| = \sqrt{E(x_k)} \leq 2\left(\frac{\sqrt{\mathrm{cond}_2(A)}-1}{\sqrt{\mathrm{cond}_2(A)}+1}\right)^k |||E_0|||$$

Here $\mathrm{cond}_2(A)$ is the condition number in the 2-norm, that is

$$\mathrm{cond}_2(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \sqrt{\rho(A^T A)} \cdot \sqrt{\rho(A^{-T}A^{-1})} = \frac{\lambda_{\max}}{\lambda_{\min}}$$

There exists a slightly better estimate

$$|||E_k||| \leq 2\left(\frac{c^k}{1+c^{2k}}\right)|||E_0|||$$

where $c = \frac{\sqrt{\mathrm{cond}_2(A)}-1}{\sqrt{\mathrm{cond}_2(A)}+1}$ (see [1]).

1.1.2 Computational costs

If we want to reduce the initial error E_0 by a quantity ε, we have to take

$$2\left(\frac{\sqrt{\mathrm{cond}_2(A)}-1}{\sqrt{\mathrm{cond}_2(A)}+1}\right)^k = \varepsilon$$

from which

$$k = \frac{\ln\frac{\varepsilon}{2}}{\ln\left(\frac{\sqrt{\mathrm{cond}_2(A)}-1}{\sqrt{\mathrm{cond}_2(A)}+1}\right)} = \frac{\ln\frac{\varepsilon}{2}}{\ln\left(1-\frac{2}{\sqrt{\mathrm{cond}_2(A)}+1}\right)} \approx \frac{\ln\frac{\varepsilon}{2}}{-\frac{2}{\sqrt{\mathrm{cond}_2(A)+1}}} \approx \frac{1}{2}\ln\frac{2}{\varepsilon}\sqrt{\mathrm{cond}_2(A)}$$

For a matrix with $\mathrm{cond}_2(A) \approx h^{-2}$ the number of expected iterations is therefore $\mathcal{O}(1/h)$. The cost of a single iteration is $\mathcal{O}(n)$ if A is sparse. The algorithm does not explicitely require A, but only the “action” of A to a vector p_k.

2 Preconditioning

The idea is to change

$$A\bar{x} = b$$

into

$$P^{-1}A\bar{x} = P^{-1}b$$

in such a way that $P^{-1}A$ is better conditioned than A. The main problem for the CG algorthm is that even if P is SPD, $P^{-1}A$ is not SPD. We can therefore factorize P into $P = R^T R$ and consider the linear system

$$P^{-1}AR^{-1}\bar{y} = P^{-1}b \Leftrightarrow R^{-T}AR^{-1}\bar{y} = R^{-T}b, \quad R^{-1}\bar{y} = \bar{x}$$

Now, $\tilde{A} = R^{-T}AR^{-1}$ is SPD and we can solve the system $\tilde{A}\bar{y} = \tilde{b}$, $\tilde{b} = R^{-T}b$, with the CG method. Setting $\tilde{x}_k = Rx_k$, we have $\tilde{r}_k = \tilde{b}_k - \tilde{A}\tilde{x}_k = R^{-T}b - R^{-T}Ax_k = R^{-T}r_k$. It is possible then to arrange the CG algorithm for $\tilde{A}$, $\tilde{x}_0$ and $\tilde{b}$ as

- x_0 given, $r_0 = b - Ax_0$, $Pz_0 = r_0$, $p_0 = z_0$
- FOR $k = 0, 1, \ldots$ UNTIL $\|r_k\|_2 \leq \text{tol} \cdot \|b\|_2$

$$\begin{aligned}
w_k &= Ap_k \\
\alpha_k &= \frac{z_k^T r_k}{p_k^T w_k} \\
x_{k+1} &= x_k + \alpha_k p_k \\
r_{k+1} &= r_k - \alpha_k w_k \\
Pz_{k+1} &= r_{k+1} \\
\beta_{k+1} &= \frac{z_{k+1}^T r_{k+1}}{z_k^T r_k} \\
p_{k+1} &= z_{k+1} + \beta_{k+1} p_k
\end{aligned}$$

END

The directions p_k are still A conjugate directions (with $Pp_0 = r_0$). This algorithm requires the solution of the linear system $Pz_{k+1} = r_{k+1}$ at each iteration. Usually, P (if not diagonal) is factorized once and for all into $P = R^T R$, R the triangular Choleski factor, in such a way that z_{k+1} can be recovered by two simple triangular linear systems.

The algorithm does not explicitely require P, but only the action of P^{-1} to a vector z_{k+1}.

2.1 Differential preconditioners

If $u(x) \approx \bar{u}(x) \approx \tilde{u}(x)$ with

$$\bar{u}(x) = \sum_{i=1}^{m} \bar{u}_i \phi_i(x)$$

with $\bar{u}_i \approx u(x_i)$ and

$$\tilde{u}(x) = \sum_{j=1}^{n} \tilde{u}_j \psi_j(x), \quad n \leq m$$

with $\tilde{u}_j \approx u(y_j)$, then it is possbile to evaluate $\tilde{u}(x_i)$ by

$$[\tilde{u}(x_1), \ldots, \tilde{u}(x_m)]^T = R\tilde{u}, \quad R \in \mathbb{R}^{m \times n},\ R_{ij} = \psi_j(x_i)$$

and $\bar{u}(y_j)$ by

$$[\bar{u}(y_1), \ldots, \bar{u}(y_n)]^T = Q\bar{u}, \quad Q \in \mathbb{R}^{n \times m},\ Q_{ji} = \phi_i(y_j)$$

We also have

$$\begin{aligned} [u(x_1), \ldots, u(x_m)]^T &\approx \bar{u} \approx R\tilde{u} \\ [u(y_1), \ldots, u(y_n)]^T &\approx \tilde{u} \approx Q\bar{u} \end{aligned}$$

and

$$\begin{aligned} [u(x_1), \ldots, u(x_m)]^T &\approx RQ\bar{u} \\ [u(y_1), \ldots, u(y_n)]^T &\approx QR\tilde{u} \end{aligned}$$

Therefore

$$RQ \approx I_m, \quad QR \approx I_n$$

Thus, in order to solve the "difficult" problem

$$\bar{A}\bar{u} = \bar{b}$$

we may want to compute $\tilde{A}$ of the "easy" problem

$$\tilde{A}\tilde{u} = \tilde{b}$$

and then use the approximation

$$\bar{A}\bar{u} \approx R\tilde{A}Q\bar{u} \Leftrightarrow \bar{A} \approx R\tilde{A}Q$$

to compute a preconditioner $\bar{A}^{-1} \approx (R\tilde{A}Q)^{-1} \approx R\tilde{A}^{-1}Q$.

2.2 Algebraic preconditioners

References

[1] J. R. Shewchuk, An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, 1994, http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf.