

Chapter 13

Conjugate-Gradient Method

The conjugate-gradient method is an iterative technique for solving large systems of linear equations $\mathbf{Ax} = \mathbf{b}$ when the coefficient matrix \mathbf{A} is symmetric and positive-definite:

$$\mathbf{A} \in \mathbb{R}^{n \times n}, \quad \mathbf{A}^T = \mathbf{A}, \quad (13.1)$$

$$\forall \mathbf{x} \in \mathbb{R}_v^n : \quad \mathbf{x}^T \mathbf{Ax} > 0. \quad (13.2)$$

$(\mathbf{x} \neq \mathbf{0})$

One way to approach the problem of solving $\mathbf{Ax} = \mathbf{b}$ is to find a functional of \mathbf{y} which is minimized when \mathbf{y} is equal to the solution vector, \mathbf{x} . Since \mathbf{A} is symmetric and positive definite, we can define the new inner product

$$\langle\langle \mathbf{y}, \mathbf{x} \rangle\rangle := (\mathbf{y}, \mathbf{Ax}) := \mathbf{y}^T \mathbf{Ax}. \quad (13.3)$$

Then

$$\langle\langle \mathbf{y} - \mathbf{x}, \mathbf{y} - \mathbf{x} \rangle\rangle \geq 0. \quad (13.4)$$

equality occurring iff $\mathbf{y} = \mathbf{x}$. But

$$\begin{aligned} \frac{1}{2} \langle\langle \mathbf{y} - \mathbf{x}, \mathbf{y} - \mathbf{x} \rangle\rangle &= \frac{1}{2} \langle\langle \mathbf{y}, \mathbf{y} \rangle\rangle - \langle\langle \mathbf{y}, \mathbf{x} \rangle\rangle + \frac{1}{2} \langle\langle \mathbf{x}, \mathbf{x} \rangle\rangle \\ &= \frac{1}{2} (\mathbf{y}, \mathbf{Ay}) - (\mathbf{y}, \mathbf{Ax}) + \frac{1}{2} (\mathbf{x}, \mathbf{Ax}). \end{aligned} \quad (13.5)$$

⁰© Copyright 1993, 1994, 1995 by C. D. Cantrell. All rights reserved. This document may not be reproduced or transmitted in whole or in part by any mechanical, electronic or optical means, or by any combination of such means, without the written permission of the author.

So far the functional depends explicitly upon the unknown exact solution, \mathbf{x} . Therefore we remove the constant term $\frac{1}{2}(\mathbf{x}, \mathbf{Ax})$ and use the linear equation to replace \mathbf{Ax} with \mathbf{b} . We have therefore shown that the vector \mathbf{y} which minimizes the functional

$$\phi[\mathbf{y}] := \frac{1}{2}(\mathbf{y}, \mathbf{Ay}) - (\mathbf{y}, \mathbf{b}) \quad (13.6)$$

solves the linear equation $\mathbf{Ax} = \mathbf{b}$.

Since minimizing $\phi[\mathbf{y}]$ is equivalent to minimizing $\langle\langle \mathbf{z}, \mathbf{z} \rangle\rangle = (\mathbf{z}, \mathbf{Az})$, where $\mathbf{z} = \mathbf{y} - \mathbf{x}$, it may be helpful to have a geometrical interpretation of $\langle\langle \mathbf{z}, \mathbf{z} \rangle\rangle$. The vectors \mathbf{z} which satisfy the equation

$$\langle\langle \mathbf{z}, \mathbf{z} \rangle\rangle = (\mathbf{z}, \mathbf{Az}) = 1 \quad (13.7)$$

lie on the hyperellipsoid described by the equation

$$z^i a_{ij} z^j = 1. \quad (13.8)$$

Since \mathbf{A} is real and symmetric, we can find an orthogonal transformation \mathbf{V} such that

$$\mathbf{V}^T \mathbf{A} \mathbf{V} = \Lambda. \quad (13.9)$$

where $\Lambda = \text{diag}[\lambda_1, \dots, \lambda_n]$ and each $\lambda_i > 0$.

Then

$$\begin{aligned} (\mathbf{z}, \mathbf{Az}) &= \mathbf{z}^T \mathbf{Az} = \mathbf{z}^T \mathbf{V} \Lambda \mathbf{V} \mathbf{z} \\ &= \mathbf{w}^T \Lambda \mathbf{w} \end{aligned} \quad (13.10)$$

where $\mathbf{w} = \mathbf{Vz}$ is the same abstract vector as \mathbf{z} , referred to new axes. Then the equation of the hyperellipsoid is

$$\sum_{i=1}^n (w^i)^2 \lambda_i = 1. \quad (13.11)$$

Let $\lambda_i = \sigma_i = \frac{1}{c_i^2}$ where $c_i > 0$, and assume that

$$\sigma_1 = \frac{1}{c_1^2} \geq \dots \geq \frac{1}{c_n^2} = \sigma_n. \quad (13.12)$$

Then the equation of the hyperellipsoid is

$$\sum_{i=1}^n \left(\frac{w^i}{c_i}\right)^2 = 1, \quad (13.13)$$

The semi major axis is c_n , and the semi minor axis is c_1 . The condition member of \mathbf{A} is

$$\text{cond}_2[\mathbf{A}] = \frac{\sigma_1}{\sigma_n} = \frac{c_n^2}{c_1^2}. \quad (13.14)$$

A well-conditioned matrix corresponds to a hyperellipsoid which is nearly spherical, while a poorly conditioned matrix corresponds to an elongated, needle-like hyperellipsoid.

In An iterative approach to minimizing $\phi[\mathbf{y}]$, one constructs a sequence $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k, \dots\}$ of approximate solutions and attempts to reduce the norm of the residual

$$\mathbf{r}_k := \mathbf{b} - \mathbf{A}\mathbf{x}_k \quad (13.15)$$

at each step. Let

$$\mathbf{x}_k := \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k$$

We write $\alpha_k \mathbf{p}_k$ instead of \mathbf{p}_k because α_k appears in other conjugate-gradient formulas.

For $k=1, \dots, n-1$: Given \mathbf{p}_k , we minimize $\phi[\mathbf{x}_k + \alpha_k \mathbf{p}_k]$ with respect to α in order to obtain α_k . Now

$$\begin{aligned} \phi[\mathbf{x}_{k-1} + \alpha \mathbf{p}_k] &= \frac{1}{2}(\mathbf{x}_{k-1} + \alpha \mathbf{p}_k, \mathbf{A}(\mathbf{x}_{k-1} + \alpha \mathbf{p}_k)) - (\mathbf{x}_{k-1} + \alpha \mathbf{p}_k, \mathbf{b}) \\ &= \frac{1}{2}(\mathbf{x}_{k-1}, \mathbf{A}\mathbf{x}_{k-1}) + \alpha(\mathbf{p}_k, \mathbf{A}\mathbf{x}_{k-1}) + \frac{1}{2}\alpha^2(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k) - \\ &\quad (\mathbf{x}_{k-1}, \mathbf{b}) - \alpha(\mathbf{p}_k, \mathbf{b}) \\ &= \left[\frac{1}{2}(\mathbf{x}_{k-1}, \mathbf{A}\mathbf{x}_{k-1}) - (\mathbf{x}_{k-1}, \mathbf{b}) \right] - \alpha(\mathbf{p}_k, \mathbf{r}_{k-1}) + \\ &\quad \frac{1}{2}\alpha^2(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k). \end{aligned} \quad (13.16)$$

The minimum of this expression occurs at

$$\alpha = \alpha_k := \frac{(\mathbf{p}_k, \mathbf{r}_{k-1})}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)} \quad (13.17)$$

An obvious but poor choice of the vectors \mathbf{p}_k is the method of steepest descent,

$$\mathbf{p}_k = \mathbf{r}_{k-1}. \quad (13.18)$$

When \mathbf{A} is ill-conditioned this method is exceedingly slow to converge. To understand why, we note that

$$\frac{\partial \phi}{\partial x_{k-1}^j} = (\mathbf{A}\mathbf{x}_{k-1} - \mathbf{b})^j = -\mathbf{r}_{k-1}^j. \quad (13.19)$$

Then the residual \mathbf{r}_k is equal to minus the gradient of ϕ at \mathbf{x}_{k-1} . When we change \mathbf{x}_{k-1} by $\alpha_k \mathbf{r}_k$, we move in the direction of maximum negative change of ϕ (hence the name, "steepest descent"). Unfortunately, when \mathbf{A} is ill-conditioned and when \mathbf{x}_{k-1} does not lie on the major axis, then the direction of steepest descent is very nearly transverse to the major axis. Thus the sequence $\{\mathbf{x}_k\}$ wanders back and forth across a narrow valley with steep walls, descending very slowly towards the global minimum.

A better approach would be to ensure that \mathbf{x}_k is one of the partial sums in an expansion of the exact solution, \mathbf{x} , in terms of n vectors \mathbf{p}_k which are mutually **A-conjugate**, *i.e.*, which are mutually orthogonal under the inner product $\langle \cdot, \cdot \rangle$:

$$k \neq l \Rightarrow \langle \mathbf{p}_k, \mathbf{p}_l \rangle = 0. \quad (13.20)$$

Then the vectors $\{\mathbf{p}_k\}$ are linearly independent, hence a basis. Let

$$\mathbf{x}_k := \sum_{j=1}^k \frac{\langle \mathbf{p}_j, \mathbf{x} \rangle}{\langle \mathbf{p}_j, \mathbf{p}_j \rangle} \mathbf{p}_j, \Rightarrow \mathbf{x}_k = \mathbf{x}_{k-1} + \frac{\langle \mathbf{p}_k, \mathbf{x} \rangle}{\langle \mathbf{p}_k, \mathbf{p}_k \rangle} \mathbf{p}_k, \quad \mathbf{x}_0 = \mathbf{0}. \quad (13.21)$$

Then there can be only n \mathbf{x}'_k s, and we are guaranteed that convergence will occur in n steps! A more useful point of view is that exact orthogonality is hard to achieve numerically, and that therefore we would be well advised to look upon this as another iterative approach.

To turn this idea into an iterative method, we must get rid of the unknown \mathbf{x} . That's easy:

$$\langle \mathbf{p}_k, \mathbf{x} \rangle = (\mathbf{p}_k, \mathbf{Ax}) = (\mathbf{p}_k, \mathbf{b}). \quad (13.22)$$

Since $\mathbf{b} = \mathbf{Ax}_{k-1} + \mathbf{r}_{k-1}$, we have

$$(\mathbf{p}_k, \mathbf{b}) = (\mathbf{p}_k, \mathbf{Ax}_{k-1}) + (\mathbf{p}_k, \mathbf{r}_{k-1}) = \langle \mathbf{p}_k, \mathbf{r}_{k-1} \rangle + (\mathbf{p}_k, \mathbf{r}_{k-1}). \quad (13.23)$$

But \mathbf{x}_{k-1} is a linear combination of $\mathbf{p}_1, \dots, \mathbf{p}_{k-1}$, to which \mathbf{p}_k is orthogonal. Then

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k \quad (13.24)$$

where

$$\alpha_k = \frac{(\mathbf{p}_k, \mathbf{r}_{k-1})}{(\mathbf{p}_k, \mathbf{Ap}_k)} \quad (13.25)$$

The remaining "detail" is to find an algorithm for constructing an orthogonal set $\{\mathbf{p}_i\}$. since we know already that \mathbf{r}_{k-1} is in the direction of steepest descent, we choose \mathbf{p}_k as the component of \mathbf{r}_{k-1} which is **A-orthogonal** to the subspace

$$\mathcal{W}_{k-1} := \text{span}[\mathbf{p}_1, \dots, \mathbf{p}_{k-1}]. \quad (13.26)$$

We construct the \mathbf{A} -orthogonal projector on \mathcal{W}_{k-1} . Since $(\mathbf{p}_i, \mathbf{A}\mathbf{y}) = (\mathbf{A}\mathbf{p}_i, \mathbf{y})$, \mathbf{A} -orthogonality to $\{\mathbf{p}_1, \dots, \mathbf{p}_{k-1}\}$ is the same as ordinary orthogonality to $\{\mathbf{A}\mathbf{p}_1, \dots, \mathbf{A}\mathbf{p}_{k-1}\}$. Then the \mathbf{A} -orthogonal projector (call it \mathbf{P}_{k-1}) on \mathcal{W}_{k-1} is the same as the ordinary orthogonal projector on $\mathbf{A}\mathcal{W}_{k-1} = \text{span}[\mathbf{A}\mathbf{p}_1, \dots, \mathbf{A}\mathbf{p}_{k-1}]$.

Define the matrices

$$\mathbf{Q}_{k-1} := [\mathbf{p}_1, \dots, \mathbf{p}_{k-1}] \quad (13.27)$$

and $\mathbf{C}_{k-1} := \mathbf{A}\mathbf{Q}_{k-1} = [\mathbf{A}\mathbf{p}_1, \dots, \mathbf{A}\mathbf{p}_{k-1}]$. Then the \mathbf{A} -orthogonal projector on \mathcal{W}_{k-1} is the ordinary orthogonal projector on $\text{range}[\mathbf{C}_{k-1}]$.

According to the SVD,

$$\mathbf{P}_{k-1} = \mathbf{P}_{\text{range}[\mathbf{C}_{k-1}]} = \mathbf{C}_{k-1}\mathbf{C}_{k-1}^+ \quad (13.28)$$

The component of \mathbf{r}_{k-1} which is \mathbf{A} -orthogonal to \mathcal{W}_{k-1} is therefore

$$\mathbf{p}_k := (\mathbf{1} - \mathbf{P}_{k-1})\mathbf{r}_{k-1} = (\mathbf{1} - \mathbf{C}_{k-1}\mathbf{C}_{k-1}^+)\mathbf{r}_{k-1}.$$

Another way to make the same statement is to say that

$$\mathbf{w}_{k-1} := \mathbf{C}_{k-1}\mathbf{C}_{k-1}^+\mathbf{r}_{k-1} \in \mathcal{W}_{k-1} \quad (13.29)$$

solves the least-squares problem of minimizing

$$\|\mathbf{p}\|_2^2 = \|\mathbf{r}_{k-1} - \mathbf{w}\|_2^2 \quad (13.30)$$

with respect to $\mathbf{w} \in \mathbf{A}\mathcal{W}_{k-1}$.

Rather than compute the residual \mathbf{r}_k as $\mathbf{b} - \mathbf{A}\mathbf{x}_k$, we can get the same result with less computational effort as follows:

$$\mathbf{r}_k - \mathbf{r}_{k-1} = \mathbf{b} - \mathbf{A}\mathbf{x}_k - (\mathbf{b} - \mathbf{A}\mathbf{x}_{k-1}) = \mathbf{A}(\mathbf{x}_{k-1} - \mathbf{x}_k) \quad (13.31)$$

$$\Rightarrow \quad \mathbf{r}_k - \mathbf{r}_{k-1} = -\alpha_k \mathbf{A}\mathbf{p}_k.$$

(Note that we have to compute $\mathbf{A}\mathbf{p}_k$ in order to compute α_k .)

We now have a workable algorithm for minimizing $\phi[\mathbf{y}]$:

Choose $\mathbf{x}_0 = \mathbf{0}$. Then $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 = \mathbf{b}$. Set $\mathbf{P}_1 = \mathbf{r}_0 = \mathbf{b}$. [This choice of \mathbf{x}_0 does not work if \mathbf{b} is an eigenvector of \mathbf{A} , for, if $\mathbf{A}\mathbf{b} = \lambda\mathbf{b}$, then the solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$ is $\mathbf{x} = \frac{1}{\lambda}\mathbf{b}$.]

For $k = 1, \dots, n - 1$:

$$\begin{aligned} \text{Compute } \alpha_k &= \frac{(\mathbf{p}_k, \mathbf{r}_{k-1})}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)} \\ \mathbf{r}_k &= \mathbf{r}_{k-1} - \alpha_k \mathbf{A}\mathbf{p}_k \\ \mathbf{x}_k &= \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k \\ \mathbf{p}_{k+1} &= (\mathbf{1} - \mathbf{C}_{k-1} \mathbf{C}_{k-1}^+) \mathbf{r}_k. \end{aligned} \quad (13.32)$$

For example,

$$\begin{aligned} \alpha_k &= \frac{(\mathbf{p}_1, \mathbf{r}_0)}{(\mathbf{p}_1, \mathbf{A}\mathbf{p}_1)} = \frac{(\mathbf{b}, \mathbf{b})}{(\mathbf{b}, \mathbf{A}\mathbf{b})} \\ \mathbf{x}_1 &= \underbrace{\mathbf{x}_0}_0 + \frac{(\mathbf{b}, \mathbf{b})}{(\mathbf{b}, \mathbf{A}\mathbf{b})} \mathbf{b} \\ \mathbf{r}_1 &= \mathbf{b} - \mathbf{A}\mathbf{x}_1 = \mathbf{b} - \frac{(\mathbf{b}, \mathbf{b})}{(\mathbf{b}, \mathbf{A}\mathbf{b})} \mathbf{A}\mathbf{b}. \end{aligned} \quad (13.33)$$

Note that $(\mathbf{r}_0, \mathbf{r}_1) = (\mathbf{b}, \mathbf{r}_1) = 0$ and that $(\mathbf{p}_1, \mathbf{r}_1) = (\mathbf{b}, \mathbf{r}_1) = 0$ while $(\mathbf{p}_1, \mathbf{r}_0) \neq 0$. If \mathbf{b} is an eigenvector of \mathbf{A} , then $\mathbf{r}_1 = \mathbf{0}$.

The above method requires the computation of $\mathbf{C}_{k-1} = \mathbf{A}\mathbf{Q}_{k-1}$ and an SVD to find \mathbf{C}_{k-1}^+ . If \mathbf{A} is a very large matrix, then we need to find a better way.

We begin with the observation that $\mathbf{p}_1, \dots, \mathbf{p}_{k-1}$ are orthogonal to \mathbf{r}_k . Certainly $\mathbf{p}_k \perp \mathbf{r}_k$:

$$(\mathbf{p}_k, \mathbf{r}_k) = (\mathbf{p}_k, \mathbf{r}_{k-1}) - \alpha_k (\mathbf{p}_k, \mathbf{A}\mathbf{p}_k) = 0. \quad (13.34)$$

Since

$$\alpha_k = \frac{(\mathbf{p}_k, \mathbf{r}_{k-1})}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)} \quad (13.35)$$

For \mathbf{p}_j (where $j < k$), we write \mathbf{r}_k as the telescoping sum

$$\begin{aligned} \mathbf{r}_k &= (\mathbf{r}_k - \mathbf{r}_{k-1}) + (\mathbf{r}_{k-1} - \mathbf{r}_{k-2}) + \dots + (\mathbf{r}_{j+1} - \mathbf{r}_j) + \mathbf{r}_j \\ &= -\alpha_k \mathbf{A}\mathbf{p}_k - \alpha_{k-1} \mathbf{A}\mathbf{p}_{k-1} - \dots - \alpha_{j+1} \mathbf{A}\mathbf{p}_{j+1} + \mathbf{r}_j \\ \Rightarrow (\mathbf{p}_j, \mathbf{r}_k) &= (\mathbf{p}_j, \mathbf{r}_j) = 0. \end{aligned} \quad (13.36)$$

Since

$$(\mathbf{p}_j, \mathbf{A}\mathbf{p}_{j+1}) = \dots = (\mathbf{p}_j, \mathbf{A}\mathbf{p}_k) = 0, \quad (13.37)$$

it follows that $\mathbf{r}_k \perp \mathcal{W}_k$.

We show that our CG algorithm implies that if \mathbf{b} is not an eigenvector of \mathbf{A} , then

$$\mathcal{W}_k = \text{span}[\mathbf{r}_0, \dots, \mathbf{r}_{k-1}] = \text{span}[\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^{k-1}\mathbf{b}]$$

These statements are true for $k = 1$. Proceeding by induction, we assume that they hold for k and that $\mathbf{r}_1, \dots, \mathbf{r}_{k-1}$ are nonzero. Then

$$\begin{aligned} \mathbf{r}_k &= \mathbf{r}_{k-1} - \alpha_k \mathbf{A}\mathbf{p}_k \\ \Rightarrow \mathbf{r}_k &\in \text{span}[\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^{k-1}\mathbf{b}, \mathbf{A}^k\mathbf{b}]. \\ \Rightarrow \text{span}[\mathbf{r}_0, \dots, \mathbf{r}_k] &\subseteq \text{span}[\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^k\mathbf{b}]. \end{aligned} \quad (13.38)$$

Now

$$\mathbf{p}_{k+1} = \mathbf{r}_k - \mathbf{C}_k \mathbf{C}_k^+ \mathbf{r}_k = \mathbf{r}_k - \mathbf{A}\mathbf{Q}_k \mathbf{C}_k^+ \mathbf{r}_k = \mathbf{r}_k - \mathbf{A}\mathbf{Q}_k \mathbf{z}_k, \quad (13.39)$$

where $\mathbf{z}_k := \mathbf{C}_k^+ \mathbf{r}_k$. Then $\mathbf{Q}_k \mathbf{z}_k$ is a LC of the columns $\mathbf{p}_1, \dots, \mathbf{p}_k$, hence a LC of $\mathbf{b}, \dots, \mathbf{A}^{k-1}\mathbf{b}$. Clearly, then, $\mathbf{A}\mathbf{Q}_k \mathbf{z}_k$ is a LC of $\mathbf{A}\mathbf{b}, \dots, \mathbf{A}^k\mathbf{b}$. It follows that

$$\mathbf{p}_{k+1} \in \text{span}[\mathbf{b}, \dots, \mathbf{A}^k\mathbf{b}]. \quad (13.40)$$

$$\Rightarrow \mathcal{W}_{k+1} \subseteq \text{span}[\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^k\mathbf{b}]. \quad (13.41)$$

$$\begin{aligned} \text{Now } \dim[\mathcal{W}_{k+1}] &= k+1 \leq \dim[\text{span}[\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^k\mathbf{b}]] \\ &\leq k+1 \end{aligned} \quad (13.42)$$

(since there are $k+1$ vectors $\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^k\mathbf{b}$). It follows that

$$\mathcal{W}_{k+1} = \text{span}[\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^k\mathbf{b}]. \quad (13.43)$$

It is also the case that $\mathbf{A}\mathcal{W}_k = \mathbf{A} \text{span}[\mathbf{b}, \dots, \mathbf{A}^{k-1}\mathbf{b}] \subset \mathcal{W}_{k+1}$ and that $\{\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^k\mathbf{b}\}$ is LI (otherwise a $k+1$ -dimensional space would be spanned by fewer than $k+1$ LI vectors).

To show that $\text{span}[\mathbf{r}_0, \dots, \mathbf{r}_k] = \mathcal{W}_{k+1}$, we must show that the \mathbf{r}_j 's are LI. By the inductive hypothesis, $\mathcal{W}_j = \text{span}[\mathbf{r}_0, \dots, \mathbf{r}_{j-1}]$ for $j = 1, \dots, k$. But for every such j , $\mathbf{r}_j \perp \mathcal{W}_j \Rightarrow \mathbf{r}_j \perp \mathbf{r}_0, \dots, \mathbf{r}_j \perp \mathbf{r}_{j-1}$ for every $j = 1, \dots, k$. Then \mathbf{r}_j is LI of $\{\mathbf{r}_0, \dots, \mathbf{r}_{j-1}\}$

$$\begin{aligned} \Rightarrow \mathbf{r}_0, \dots, \mathbf{r}_k &\text{ is LI} \\ \Rightarrow \dim[\text{span}[\mathbf{r}_0, \dots, \mathbf{r}_k]] &= k+1. \end{aligned} \quad (13.44)$$

$$\Rightarrow \text{span}[\mathbf{r}_0, \dots, \mathbf{r}_k] = \text{span}[\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^k\mathbf{b}] = \mathcal{W}_{k+1}$$

We now simplify the expression for \mathbf{p}_{k+1} ,

$$\mathbf{p}_{k+1} = \mathbf{r}_k - \mathbf{p}_k \mathbf{r}_k, \quad (13.45)$$

in order to eliminate matrix multiplication. We show that it is possible to ensure that \mathbf{p}_{k+1} is \mathbf{A} -orthogonal to \mathbf{w}_k much more simply than with a projection operator, by setting

$$\mathbf{p}_{k+1} = \mathbf{r}_k + \beta_k \mathbf{p}_k \quad (13.46)$$

and choosing the constant β_k appropriately.

For every $j = 1, \dots, k-1$,

$$(\mathbf{A}\mathbf{p}_j, \mathbf{p}_{k+1}) = (\mathbf{A}\mathbf{p}_j, \mathbf{r}_k) + \beta_k (\mathbf{A}\mathbf{p}_j, \mathbf{p}_k). \quad (13.47)$$

If $\{\mathbf{p}_1, \dots, \mathbf{p}_k\}$ is a mutually \mathbf{A} -orthogonal set, then the 2nd term on the right vanishes certainly $\mathbf{A}\mathbf{p}_j \in \mathcal{W}_k$ (since $\mathbf{A}\mathcal{W}_j \subset \mathcal{W}_{j+1} \subseteq \mathcal{W}_k$). Since $\mathbf{r}_k \perp \mathcal{W}_k$, the first term on the right vanishes, establishing that \mathbf{p}_{k+1} is \mathbf{A} -orthogonal to $\mathbf{p}_1, \dots, \mathbf{p}_{k-1}$. If $j = k$, then $(\mathbf{A}\mathbf{p}_k, \mathbf{p}_{k+1}) = 0$ iff

$$\boxed{\beta_k = -\frac{(\mathbf{p}_k, \mathbf{A}\mathbf{r}_k)}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)}} \quad (13.48)$$

We can use (13.48) to simplify the numerator of our expansion for α_k ,

$$\alpha_k = \frac{(\mathbf{p}_k, \mathbf{r}_{k-1})}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)}. \quad (13.49)$$

Since $\mathbf{p}_k = \mathbf{r}_{k-1} + \beta_{k-1}\mathbf{p}_{k-1}$, it follows that

$$(\mathbf{p}_k, \mathbf{r}_{k-1}) = (\mathbf{r}_{k-1}, \mathbf{r}_{k-1}) \quad (13.50)$$

because $(\mathbf{p}_k, \mathbf{r}_{k-1}) = 0$. Then $\boxed{\alpha_k = \frac{(\mathbf{r}_{k-1}, \mathbf{r}_{k-1})}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)}}$

The recurrence relation for the residual vector,

$$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k \mathbf{A}\mathbf{p}_k, \quad (13.51)$$

can be used to simplify our expression for β_k . We have

$$(\mathbf{r}_k, \mathbf{r}_k) = -\alpha_k (\mathbf{r}_k, \mathbf{A}\mathbf{p}_k) \quad (13.52)$$

since $\mathbf{r}_k \perp \mathbf{r}_{k-1}$. Then

$$(\mathbf{r}_k, \mathbf{A}\mathbf{p}_k) = -\frac{1}{\alpha_k} (\mathbf{r}_k, \mathbf{r}_k). \quad (13.53)$$

It follows that

$$\beta_k = -\frac{(\mathbf{p}_k, \mathbf{A}\mathbf{r}_k)}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)} = \frac{1}{\alpha_k} \frac{(\mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)} = \frac{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)}{(\mathbf{r}_{k-1}, \mathbf{r}_{k-1})} \cdot \frac{(\mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)} \quad (13.54)$$

and therefore that

$$\boxed{\beta_k = \frac{(\mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{r}_{k-1}, \mathbf{r}_{k-1})}} \quad (13.55)$$

The **conjugate-gradient algorithm** is

$$\mathbf{x}_0 = \mathbf{0}, \quad \mathbf{r}_0 = \mathbf{b}, \quad \mathbf{p}_1 = \mathbf{r}_0 = \mathbf{b}.$$

For $k = 1, \dots, n-1$:

$$\alpha_k = \frac{(\mathbf{r}_{k-1}, \mathbf{r}_{k-1})}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)}$$

$$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k \mathbf{A}\mathbf{p}_k$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{p}_k$$

$$\beta_k = \frac{(\mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{r}_{k-1}, \mathbf{r}_{k-1})}$$

$$\mathbf{p}_{k+1} = \mathbf{r}_k + \beta_k \mathbf{p}_k$$

Convergence of the CG algorithm

If $\mathbf{A} = \mathbf{1} + \mathbf{B}$ and $\text{rank}[\mathbf{B}] = r$, then the CGA converges in at most $r+1$ steps, for

$$\begin{aligned} \dim[\mathcal{W}_k] &= \dim[\text{span}[\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^{k-1}\mathbf{b}]] \\ &= \dim[\text{span}[\mathbf{b}, \mathbf{B}\mathbf{b}, \dots, \mathbf{B}^{k-1}\mathbf{b}]] \\ &\leq r+1, \end{aligned} \quad (13.56)$$

since $r = \dim[\text{range}[\mathbf{B}]]$ by definition.

For example, if $\mathbf{B} = \mathbf{Q}$ then $\mathbf{x}_0 = \mathbf{0}, \mathbf{r}_0 = \mathbf{b}, \mathbf{p}_1 = \mathbf{b}$,

$$\alpha_1 = \frac{(\mathbf{r}_0, \mathbf{r}_0)}{(\mathbf{p}_1, \mathbf{A}\mathbf{p}_1)} = \frac{(\mathbf{b}, \mathbf{b})}{(\mathbf{b}, \mathbf{b})} = 1, \quad \text{and} \quad \mathbf{x}_1 = \mathbf{x}_0 + \alpha_1 \mathbf{A}\mathbf{p}_1 = \mathbf{b}.$$

The method has converged in $r+1$ steps, where $r = 0$.

It can be shown (D.G Luenberger, *Intro. to Linear & Nonlinear Programming* (Addison-Wesley, 1993), p.187) that

$$\|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{A}} \leq \|\mathbf{x} - \mathbf{x}_0\|_{\mathbf{A}} \left[\frac{\sqrt{\text{cond}_2[\mathbf{A}] - 1}}{\sqrt{\text{cond}_2[\mathbf{A}] + 1}} \right]^k \quad (13.57)$$

where $\|\mathbf{y}\|_{\mathbf{A}} := (\mathbf{y}, \mathbf{A}\mathbf{y})^{1/2}$.

Although this error estimate is often unrealistically conservative, it implies that one way to guarantee that the CG method converges rapidly is to ensure that \mathbf{A} (or some symmetric, positive-definite transform of \mathbf{A}) has a condition number approximately equal to 1.

Preconditioning a matrix for the CG algorithm

Since \mathbf{A} is real, symmetric and positive-definite, there exists a real orthogonal matrix \mathbf{V} such that

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \quad (13.58)$$

$$\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{1} \quad (13.59)$$

$$\mathbf{\Lambda} = \text{diag}[\lambda_1, \dots, \lambda_n] \quad (13.60)$$

$$\lambda_1 \geq \dots \geq \lambda_n \geq 0. \quad (13.61)$$

Let

$$\left. \begin{aligned} \mathbf{C} &:= \mathbf{V}\mathbf{\Lambda}^{1/2}\mathbf{V}^T \\ \mathbf{\Lambda}^{1/2} &:= \text{diag}[\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n}] \end{aligned} \right\} \Rightarrow \mathbf{C}^T = \mathbf{C}. \quad (13.62)$$

Then

$$\begin{aligned} \mathbf{C}^{-1} &= \mathbf{V}\mathbf{\Lambda}^{-1/2}\mathbf{V}^T \\ \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1} &= \mathbf{V}\mathbf{\Lambda}^{-1/2}\mathbf{\Lambda}\mathbf{\Lambda}^{-1/2}\mathbf{V}^T = \mathbf{1}. \end{aligned} \quad (13.63)$$

It follows that there always exists a symmetric, positive-definite matrix \mathbf{C} such that $\mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}$ has condition number 1. In practice, one tries to lower the condition number, but not necessarily to 1.

The system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is equivalent to the system

$$\mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}\mathbf{C}\mathbf{x} = \mathbf{C}^{-1}\mathbf{b}. \quad (13.64)$$

Let

$$\left. \begin{aligned} \mathbf{A}' &:= \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}, \\ \mathbf{x}' &:= \mathbf{C}\mathbf{x}, \\ \mathbf{b}' &:= \mathbf{C}^{-1}\mathbf{b} \end{aligned} \right\} \Rightarrow \mathbf{A}'\mathbf{x}' = \mathbf{b}'. \quad (13.65)$$

If we apply the CG method to the new system

$$\alpha'_k = \frac{(\mathbf{r}'_{k-1}, \mathbf{r}'_{k-1})}{(\mathbf{p}'_k, \mathbf{A}'\mathbf{p}'_k)}, \quad \mathbf{r}'_k = \mathbf{r}'_{k-1} - \alpha'_k \mathbf{A}'\mathbf{p}'_k, \quad (13.66)$$

we have $\mathbf{x}'_k = \mathbf{x}'_{k-1} + \alpha'_k \mathbf{p}'_k$, $\beta'_k = \frac{(\mathbf{r}'_k, \mathbf{r}'_k)}{(\mathbf{r}'_{k-1}, \mathbf{r}'_{k-1})}$, and

$$\mathbf{p}'_{k+1} = \mathbf{r}'_k + \beta'_k \mathbf{p}'_k. \quad (13.67)$$

In order to avoid the computational labor needed to compute $\mathbf{A}' = \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}$, we define $\mathbf{C}^{-1}\mathbf{b}'' := \mathbf{b}'$, $\mathbf{C}\mathbf{p}''_k := \mathbf{p}'_k$, $\mathbf{C}\mathbf{x}''_k := \mathbf{x}'_k$, and $\mathbf{C}^{-1}\mathbf{r}''_k := \mathbf{r}'_k$. Since $\mathbf{C}^{-1}\mathbf{r}_k = \mathbf{C}^{-1}\mathbf{b} - \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}\mathbf{C}\mathbf{x}_k$, it follows that $\mathbf{b}'' = \mathbf{b}$,

$$\alpha'_k = \frac{(\mathbf{r}''_{k-1}, \mathbf{C}^{-2}\mathbf{r}''_{k-1})}{(\mathbf{p}''_k, \mathbf{A}\mathbf{p}''_k)}, \quad \beta'_k = \frac{(\mathbf{r}''_k, \mathbf{C}^{-2}\mathbf{r}''_k)}{(\mathbf{r}''_{k-1}, \mathbf{C}^{-2}\mathbf{r}''_{k-1})}, \quad (13.68)$$

and $\mathbf{x}''_k = \mathbf{x}''_{k-1} + \alpha'_k \mathbf{p}''_k$, $\mathbf{r}''_k = \mathbf{r}''_{k-1} - \alpha'_k \mathbf{A}\mathbf{p}''_k$.

The change of variables from primed to double-primed makes \mathbf{A} appear instead of $\mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}$. Indeed, \mathbf{C} enters only in the inner products $(\mathbf{r}''_j, \mathbf{C}^{-2}\mathbf{r}''_j)$.

Since it is preferable to obtain $\mathbf{C}^{-2}\mathbf{r}''_j$ as the solution of a system of linear equations, we let

$$\begin{aligned} \mathbf{M} &:= \mathbf{C}^2 \\ \mathbf{M}\mathbf{z}_j &:= \mathbf{r}''_j. \end{aligned} \quad (13.69)$$

Then

$$\mathbf{p}''_{k+1} = \mathbf{z}_k + \beta'_k \mathbf{p}''_k. \quad (13.70)$$

The **preconditioned conjugate-gradient algorithm** is

$$\mathbf{x}_0 = \mathbf{0}, \quad \mathbf{r}_0 = \mathbf{b}; \text{ solve } \mathbf{M}\mathbf{z}_0 = \mathbf{b}; \quad \mathbf{p}_1 = \mathbf{z}_0$$

For $k = 1, \dots, n-1$:

$$\begin{aligned} \alpha'_k &= \frac{(\mathbf{z}_{k-1}, \mathbf{r}_{k-1})}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)} \\ \mathbf{r}_k &= \mathbf{r}_{k-1} - \alpha'_k \mathbf{A}\mathbf{p}_k \\ \mathbf{x}_k &= \mathbf{x}_{k-1} + \alpha'_k \mathbf{p}_k \\ \text{solve } \mathbf{M}\mathbf{z}_k &= \mathbf{r}_k \\ \beta'_k &= \frac{(\mathbf{z}_k, \mathbf{r}_k)}{(\mathbf{z}_{k-1}, \mathbf{r}_{k-1})} \\ \mathbf{p}_{k+1} &= \mathbf{z}_k + \beta'_k \mathbf{p}_k, \end{aligned} \quad (13.71)$$

Since

$$\left\{ \begin{array}{l} (\mathbf{r}'_j, \mathbf{r}'_l) = 0 \\ (\mathbf{p}'_j, \mathbf{A}'\mathbf{p}'_l) = 0 \end{array} \right\} \text{ if } j \neq l \quad (13.72)$$

we have

$$\left\{ \begin{array}{l} (\mathbf{r}_j, \mathbf{M}^{-1}\mathbf{r}_l) = 0 \\ (\mathbf{p}_j, \mathbf{A}\mathbf{p}_l) = 0 \end{array} \right\} \text{ if } j \neq l. \quad (13.73)$$

In order for this approach to be useful, \mathbf{M} must be chosen such that the computational effort required to solve $\mathbf{M}\mathbf{z}_j = \mathbf{r}_j$ is much smaller than $\frac{1}{N}$ times the effort normally required to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$ in approximately N steps.