Projection-Based Iterative Methods, I

Introduction

- We wish to develop an iterative solution method for the system $A\underline{x} = \underline{b}$, where $A \in \mathbb{R}^{n\times n}$ is assumed to symmetric positive definite (SPD) unless otherwise indicated.

- Our main interest in iterative methods is for *matrix-free* applications, in which one does not have explicit access to the matrix A but instead simply has a black box that, given a vector $\underline{p}$ returns another vector $\underline{w} = A\underline{p}$.

- To introduce the topic, we make three observations about Jacobi relaxation,

$$\underline{x}_{k+1} \;=\; \underline{x}_k + D^{-1}(\underline{b} - A\underline{x}_k) \qquad , \tag{1}$$

  where $D =$diag$\{a_{ii}\}$.

- **Observation 1:** Subtracting (1) from $\underline{x} = \underline{x} + D^{-1}(\underline{b} - A\underline{x})$ yields the associated equation for the error, $\underline{e}_k := \underline{x} - \underline{x}_k$,

$$\underline{e}_{k+1} \;=\; (I - D^{-1}A)\,\underline{e}_k, \tag{2}$$

or

$$\underline{e}_k \;=\; (I - D^{-1}A)^k\,\underline{e}_0. \tag{3}$$

- For convergence, we require $\rho(I - D^{-1}A) < 1$, where, for any matrix C with eigenvalues λ_i, $\rho(C) = max_i|\lambda_i|$ is the *spectral radius* of C.

- Convergence will be rapid if $\rho(I - D^{-1}A) \ll 1$.

- **Observation 2:** With $\underline{x}_0 = 0$ we have $\underline{e}_0 = \underline{x}$. Using the error equation (3) we can generate an explicit polynomial form for $\underline{x}_k$:

$$\begin{aligned} \underline{x} - \underline{x}_k &= (I - D^{-1}A)^k \, \underline{x} \\ &= \left[I - a_1 D^{-1}A - a_2 (D^{-1}A)^2 - \cdots - a_k (D^{-1}A)^k\right] \underline{x}, \end{aligned} \tag{4}$$

from which

$$\underline{x}_k = \left[a_1 I + a_2 D^{-1}A^2 + \cdots + a_k (D^{-1}A)^{k-1}\right] D^{-1}\underline{b} \tag{5}$$

$$\in K_k(D^{-1}A; D^{-1}\underline{b}). \tag{6}$$

- Here, K_k denotes the *Krylov subspace*.

- For any matrix $C \in \mathbb{R}^{n\times n}$ and vector $\underline{v} \in \mathcal{R}^n$, we define

$$K_k(C, \underline{v}) := span\left\{\underline{v}, C\underline{v}, C^2\underline{v}, \ldots C^{k-1}\underline{v}\right\} = \mathbb{P}_{k-1}(C)\underline{c}, \tag{7}$$

with $\mathbb{P}_{k-1}$ denoting the space of polynomials of degree $\leq k$-1 in the argument.

- **Observation 3:** The polynomials (3) and (5) do not reflect any properties of our particular D^{-1}, A, or $\underline{b}$. The polynomial coefficients a_j derive from the binomial expansion for $(I - D^{-1}A)^k$ and are thus not likely to be optimal.

- The projection methods developed below (of which conjugate gradients and GMRES are two of the most common examples) allow us to find the *best fit*, $\underline{x}_k \in K_k$ such that

$$\|\underline{x} - \underline{x}_k\|_* \;\leq\; \|\underline{x} - \underline{v}\|_* \; \forall \underline{v} \in K_k, \tag{8}$$

  for a particular norm $\|\cdot\|_*$ to be determined.

- If A is SPD, projection methods have a per-iteration cost that is comparable to Jacobi relaxation (1) but that require far fewer iterations to converge.

- Before proceding, we make two comments about the matrix pair D and A.

- First, we can relax the assumption that D is the diagonal of A and instead assume that it is some more general preconditioner (perhaps also SPD if A is).

- Second, we can also rescale the system in terms of a matrix for which $D = I$. If $a_{ii} > 0$ (e.g., if A is SPD), then $A\underline{x} = \underline{b}$ is equivalent to $\tilde{A}\underline{\tilde{x}} = \underline{\tilde{b}}$ with $\tilde{a}_{ii} \equiv 1$ and

$$\tilde{A} := D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \qquad \underline{\tilde{x}} := D^{\frac{1}{2}} \underline{x}, \qquad \underline{\tilde{b}} := D^{-\frac{1}{2}} \underline{b}.$$

- Consequently, we develop the projection schemes initially without reference to the preconditioner and seek $\underline{x}_k \in K_k(A; \underline{b})$. We will revisit preconditioning in a separate lecture.

Projection methods

- Projection methods are based upon the idea of minimizing the error between $\underline{x}$ and the kth iterate, $\underline{x}_k$, in an appropriately chosen norm.

- $\underline{x}_k$ is selected from a k-dimensional subspace of $\mathbb{R}^n$ denoted by

$$\mathcal{V}_k = span\{\underline{v}_1\, \underline{v}_2\, \ldots\, \underline{v}_k\}.$$

- At present, we make no assumptions about $\mathcal{V}_k$ aside from the fact that basis vectors, or *search directions*, $\underline{v}_j$, should be linearly independent and hence a spanning set of $\mathcal{V}_k$.

- Note that the minimization problem is well-defined even in the absence of linear independence, but the methods prescribed below would break down and stability is potentially compromised.

- To derive the projection-based approximation we start with

$$\underline{x}_k \;=\; \sum_{j=1}^{k} \beta_j \underline{v}_j \;\in\; \mathcal{V}_k. \tag{9}$$

- Our task is to find coefficients β_j such that

$$\|\underline{x} - \underline{x}_k\|_{\mathrm{w}} \;\leq\; \|\underline{x} - \underline{w}\|_{\mathrm{w}} \;\forall \underline{w} \in \mathcal{V}_k. \tag{10}$$

- To be concrete, for some SPD matrix W, define

$$(\underline{v}, \underline{w})_{\mathrm{w}} \;:=\; \underline{v}^T W \underline{w} \tag{11}$$

$$\|\underline{v}\|_{\mathrm{w}} \;:=\; (\underline{v}, \underline{v})_{\mathrm{w}}^{\frac{1}{2}} \tag{12}$$

- Later, we will choose $W = A$ if A is SPD or $W = A^T A$ if A is nonsymmetric.

Minimization $\Longleftrightarrow$ Orthogonalization

- We need to derive a set of equations to find the best-fit, $\underline{x}_k$.

- We start with its definining property.

- Define $\underline{e}_k = \underline{x} - \underline{x}_k$ and $\underline{x}_k$ as the *minimizer* satisfying

$$\begin{aligned} \|\underline{x} - \underline{x}_k\|_{\mathrm{w}} &\leq \|\underline{x} - \underline{w}\|_{\mathrm{w}} \; \forall \underline{w} \in \mathcal{V}_k & (13) \\ &= \|\underline{x} - (\underline{x}_k + \epsilon \underline{v})\|_{\mathrm{w}}. & (14) \end{aligned}$$

  The second expression holds for any $\epsilon \in \mathbb{R}^1$ and $\underline{v} \in \mathcal{V}_k$.

- Replacing $\underline{x} - \underline{x}_k =: \underline{e}_k$ and expanding the bilinear forms, the projection requirement (13) becomes

$$\begin{aligned} \|\underline{e}_k\|_{\mathrm{w}}^2 \;\leq\; \|\underline{e}_k + \epsilon\underline{v}\|_{\mathrm{w}}^2 \;&=\; (\underline{e}_k + \epsilon\underline{v})^T W(\underline{e}_k + \epsilon\underline{v}) \qquad (15)\\ &=\; \underline{e}_k^T W\underline{e}_k^T \,+\, \epsilon\underline{v}^T W\underline{e}_k \,+\, \epsilon\underline{e}_k^T W\underline{v} \,+\, \epsilon^2\underline{v}^T W\underline{v},\\ &=\; \|\underline{e}_k\|_{\mathrm{w}}^2 \,+\, 2\epsilon\underline{v}^T W\underline{e}_k \,+\, \epsilon^2\|\underline{v}\|_{\mathrm{w}}^2 \qquad \forall\,(\epsilon, \underline{v} \in \mathcal{V}_k). \end{aligned}$$

- From the final inequality and the fact that ϵ can be of arbritrary sign it is clear that the error will be minimized if and only if

$$\underline{v}^T W\underline{e}_k = 0 \;\forall\, \underline{v} \in \mathcal{V}_k. \qquad (16)$$

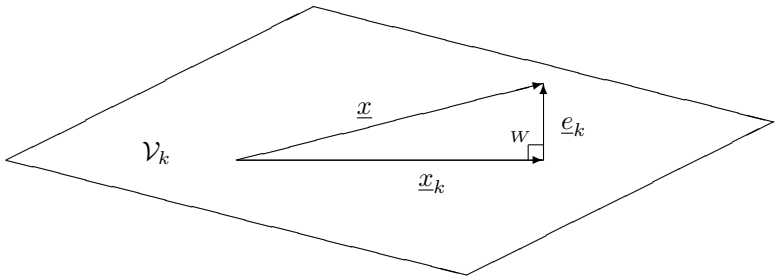
  Otherwise, it is always possible to find a value of ϵ such that the expression on the right of (15) is smaller than $\|\underline{e}_k\|_{\mathrm{w}}$, which is not possible if $\underline{x}_k$ is the minimizer of $\|\underline{x} - \underline{v}\|_{\mathrm{w}}^2$.

- We conclude that a necessary condition for $\underline{x}_k$ to be the best fit is

$$(\underline{v}, \underline{e}_k)_{\mathrm{w}} \;=\; 0 \qquad \forall\, \underline{v} \in \mathcal{V}_k. \tag{17}$$

  We say that the error is *W-orthogonal* to $\mathcal{V}_k$ ($\underline{e}_k \perp_{\mathrm{w}} \mathcal{V}_k$) and that $\underline{x}_k$ is the *projection* of $\underline{x}$ onto $\mathcal{V}_k$.

- This situation is illustrated in the accompanying figure.



- Thus, we have the following important concept:

$$\textbf{Minimization} \quad \equiv \quad \textbf{Orthogonal Projection}.$$

- An equivalent expression is that the kth iterate satisfies the *projection statement*,

$$(\underline{v}, \underline{x}_k)_W \;=\; (\underline{v}, \underline{x})_W, \qquad \forall\, \underline{v} \,\in\, \mathcal{V}_k. \tag{18}$$

- $\underline{x}_k$ being the minimizer of $\|\underline{e}_k\|_{\mathrm{w}}$ is equivalent to $\underline{e}_k$ being orthogonal to $\mathcal{V}_k$ (with respect to $(.,.)_{\mathrm{w}}$).

- Note that in the limit of $k \longrightarrow n$, we have $\mathcal{V}_k = \mathbb{R}^n$, implying that $\underline{e}_k$ is orthogonal to all $\mathbb{R}^n$.

- This can only be true if $\underline{e}_k = \underline{0}$ which implies $\underline{x}_k = \underline{x}$.

- The projection scheme is exact in this limit, modulo round-off error. (CG was originally viewed as a *direct solver*.)

Solution Generation

- We can use (16) to generate a system of equations to find the unknown basis coefficients β_j in (9).

- We'll assume that A is SPD and will minimize in the A-norm (i.e., $W = A$).

- Moreover, we'll assume that we have a basis,

$$\text{span}\{\underline{p}_1, \underline{p}_2, \ldots, \underline{p}_k\} = \text{span}\{\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_k\} = \mathcal{V}_k,$$

  with unknown basis coefficients α_j.

- Denoting these two sets of vectors by the $n \times k$ matrices P_k and V_k, respectively, the change of basis will allow us to transform a given set V_k to the set P_k that will have appropriate orthogonality properties.

- We note that the range, or column spaces, are the same, $\mathcal{R}(P_k) = \mathcal{R}(V_k)$, but the bases (the actual columns) are different.

- For now, we proceed assuming we have the set P_k in hand.

- Starting with

$$\underline{x}_k \;=\; \sum_{j=1}^{k} \underline{p}_j \, \alpha_j, \tag{19}$$

(16) implies, for $i = 1, \ldots, k$,

$$(\underline{p}_i, \underline{e}_k)_A \;:=\; \underline{p}_i^T \, A \, \underline{e}_k \;=\; 0 \tag{20}$$

$$\underline{p}_i^T \, A \, (\underline{x} \,-\, \underline{x}_k) \;=\; 0 \tag{21}$$

$$\underline{p}_i^T \, A \, \underline{x}_k \;=\; \underline{p}_i^T \, A \, \underline{x} \tag{22}$$

$$\underline{p}_i^T \, A \, \underline{x}_k \;=\; \underline{p}_i^T \, \underline{b} \tag{23}$$

$$\sum_{j=1}^{k} \underline{p}_i^T \, A \, \underline{p}_j \alpha_j \;=\; \underline{p}_i^T \, \underline{b}, \tag{24}$$

which constitutes k equations in k unknown basis coefficients α_j, $j = 1, \ldots, k$.

- Defining

$$A_k \;=\; P_k^T A P_k \tag{25}$$

$$P_k \;=\; [\underline{p}_1, \ldots, \underline{p}_k] \tag{26}$$

$$\underline{\alpha} \;=\; [\alpha_1, \ldots, \alpha_k]^T \tag{27}$$

$$\underline{b}_k \;=\; [\underline{p}_1^T \underline{b}, \ldots, \underline{p}_k^T \underline{b}]^T, \tag{28}$$

we have

$$A_k \underline{\alpha} \;=\; \underline{b}_k \quad . \tag{29}$$

- Note that, because P_k is of full rank (the $\underline{p}_i$'s are linearly independent) and A is symmetric positive definite (SPD), A_k is also SPD and the $k \times k$ system in (29) is solvable.

- Once $\underline{\alpha}$ is found, $\underline{x}_k$ is computed using (19),

$$\underline{x}_k = P_k\underline{\alpha} = P_k(P_k^T A P_k)^{-1} P_k^T \underline{b} \tag{30}$$

$$= P_k(P_k^T A P_k)^{-1} P_k^T A\underline{x}. \tag{31}$$

- The expression on the right of (31) is a classic projection statement.

- It represents the A-orthogonal projection of $\underline{x}$ onto $\mathcal{R}(P_k)$.

- The solution $\underline{x}_k$ is in $\mathcal{R}(P_k)$, the units of A and A^{-1} cancel out, as do the units of P_k.

- Thus, $\underline{x}_k$ has the same units as $\underline{x}$, as it should.

- It is shorter in the A-norm than $\underline{x}$, $\|\underline{x}_k\|_A < \|\underline{x}_k\|_A$, unless $\underline{x}$ is in $\mathcal{R}(P_k)$, in which case $\underline{x}_k = \underline{x}$.

Basis for $\mathcal{V}_k$

- We now seek a procedure for generating the basis set $\{\underline{p}_j\}$.

- The minimization procedure (29) is greatly simplified if the basis vectors are A-conjugate, implying:

$$\underline{p}_i^T A \underline{p}_j \;=\; 0 \qquad i \;\neq\; j \qquad . \tag{32}$$

- This results in A_k being diagonal, as its entries are simply $(A_k)_{ij} = \underline{p}_i^T A \underline{p}_j$, and leads immediately to a closed form expression for α_j:

$$\alpha_j \quad = \quad \frac{\underline{p}_j^T \underline{b}}{\underline{p}_j^T A \underline{p}_j} \qquad , \tag{33}$$

  from which,

$$\underline{x}_k \quad = \quad \sum_{j=1}^{k} \frac{\underline{p}_j^T \underline{b}}{\underline{p}_j^T A \underline{p}_j} \underline{p}_j \;=\; \sum_{j=1}^{k} \frac{(\underline{p}_j, \underline{x})_A}{(\underline{p}_j, \underline{p}_j)_A} \underline{p}_j \,. \tag{34}$$

  The second expression is the familiar form associated with the projection of $\underline{x}$ onto the orthogonal basis vectors $\underline{p}_j$, with respect to $(.,.)_A$.

- The next approximate solution, or iterate, $\underline{x}_{k+1}$, of course has the same form as $\underline{x}_k$, with a change in subscript.

- Consequently,

$$\underline{x}_{k+1} = \sum_{j=1}^{k+1} \frac{\underline{p}_j^T \underline{b}}{\underline{p}_j^T A \underline{p}_j} \underline{p}_j \tag{35}$$

$$= \underline{x}_k + \frac{\underline{p}_{k+1}^T \underline{b}}{\underline{p}_{k+1}^T A \underline{p}_{k+1}} \underline{p}_{k+1} \tag{36}$$

$$= \underline{x}_k + \alpha_{k+1} \underline{p}_{k+1} \quad . \tag{37}$$

- The computation of $\underline{x}_{k+1}$ is therefore a simple correction to $\underline{x}_k$;

- Only one additional coefficient, α_{k+1}, needs to be computed to update $\underline{x}_k$ and this coefficient depends only upon the current search direction $\underline{p}_{k+1}$ and the initial residual, $\underline{b}$.

- Thus, (37) gives a two-term *recurrence* relation for $\underline{x}_k$ implying that there is no need to refer to the entire space $\mathcal{V}_k$ to compute successive approximations, $\underline{x}_k$.

- We emphasize that the simple recurrence (37) holds for any A-conjugate (i.e., A-orthogonal) basis and *only* for A-conjugate bases.

- The essence of the projection procedure is the following:
  - Choose $\underline{p}_k \in \mathcal{V}^k \notin \mathcal{V}^{k-1}$ such that $\underline{p}_k A \underline{p}_j = 0, j < k$.
  - Compute $\alpha_k = \frac{\underline{p}_k^T \underline{b}}{\underline{p}_k^T A \underline{p}_k} = \frac{\underline{p}_k^T (\underline{b} - A\underline{x}_{k-1})}{\underline{p}_k^T A \underline{p}_k} = \frac{\underline{p}_k^T \underline{r}_{k-1}}{\underline{p}_k^T A \underline{p}_k}$
  - Update solution: $\underline{x}_k = \underline{x}_{k-1} + \alpha_k \underline{p}_k$.
  - Update residual: $\underline{r}_k = \underline{r}_{k-1} - \alpha_k A \underline{p}_k$.

- Here, we have introduced the important (computable!) *residual vector*,

$$\underline{r}_k \;:=\; \underline{b} \,-\, A\underline{x}_k \;=\; A\underline{x} \,-\, A\underline{x}_k \;=\; A\underline{e}_k, \tag{38}$$

  with the error vector, $\underline{e}_k := \underline{x} - \underline{x}_k$.

- The residual vector is thus a direct measure of the current iteration error (the only one available, in fact).

- It can be computed in a stable recursive way using (37).

- Let $\underline{w}_k := A\underline{p}_k$, which is need in the denominator of (36).

- Then the last three steps of our A-conjugate projector will be of the form

$$\underline{w}_k \;=\; A\underline{p}_k, \quad \alpha_k \;=\; \frac{\underline{p}_k^T \underline{r}_{k-1}}{\underline{p}_k^T \underline{w}_k} \tag{39}$$

$$\underline{x}_k \;=\; \underline{x}_{k-1} \,+\, \alpha_k \underline{p}_k \tag{40}$$

$$\underline{r}_k \;=\; \underline{r}_{k-1} \,-\, \alpha_k \underline{w}_k. \tag{41}$$

Generating P_k

- More generaly, given any linearly-independent approximation space $\mathcal{V}_k = \text{span}\{\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_k\}$, we generate the requisite A-conjugate search direction p_k using the following Gram-Schmidt procedure.

$$\underline{p}_k = \underline{v}_k - \Pi_A(P_{k-1}, \underline{v}_k) \tag{42}$$

$$= \underline{v}_k - \sum_{j=1}^{k-1} \underline{p}_j \beta_j \tag{43}$$

  with

$$\beta_j = \frac{(\underline{p}_j, \underline{v}_k)_A}{(\underline{p}_j, \underline{p}_j)_A} = \frac{\underline{v}_k^T A \underline{p}_j}{\underline{p}_j^T A \underline{p}_j}. \tag{44}$$

- Thus, $\underline{p}_k$ is essentially $\underline{v}_k$ minus the A-orthogonal projection of $\underline{v}_k$ onto $\mathcal{R}(P_{k-1})$=span$\{\mathcal{V}_{k-1}\}$.

Krylov Subspace Projection Methods

- Note that computation of the projection in (36) requires matrix vector products of the form $A\underline{p}_j$.

- Consequently, we can choose the Krylov subspace

$$K_k(A;b) := \{\underline{b}\,, A\underline{b}\,, \ldots, A^{k-1}\underline{b}\} \tag{45}$$

  as our subspace $\mathcal{V}_k$.

  It is clear that if $\underline{p}_1 = \underline{b}$, then $\underline{p}_2$ will be a linear combination of $\underline{b}$ and $A\underline{b}$. Since $A\underline{b}$ is required in the minimization over $\mathcal{V}_1$ there is no additional cost (in terms of matrix-vector products) associated with forming $\underline{p}_2$.

- An important property of the residual vector $\underline{r}_k$ is that it is L^2-orthogonal to the search space $\mathcal{V}_k$, which follows because $\underline{e}_k$ is A-orthogonal to $\mathcal{V}_k$.

- For all $\underline{v} \in \mathcal{V}_k$,

$$0 \;=\; \underline{v}^T A \underline{e}_k = \underline{v}^T \underline{r}_k. \tag{46}$$

- Thus, $R_k := [\underline{r}_0 \; \underline{r}_1 \; \ldots \; \underline{r}_{k-1}]$ is an orthogonal basis and we can use it as an approximation space.

- That is, take $\mathcal{V}_k := \mathcal{R}(R_k)$.

- If our initial guess $\underline{x}_0 = 0$, we have $\underline{r}_0 = \underline{b}$.

- Starting with $\underline{v}_1 = \underline{r}_0$, successively construct new basis vectors from the residual vectors.

- This will lead to a Krylov subspace approximation, $\mathcal{V}_k = K_k(A; \underline{b})$.

- Because we are using *projection* (unlike Jacobi iteration), we refer to such schemes as *Krylov subspace projection* (KSP) methods.

- Combining the choice $\mathcal{V}_k = \mathcal{R}(R_k)$ with our previous projection steps leads to Version 1 of our KSP, which is a variant of conjugate gradient iteration.

- Starting with $\underline{x}_0 = 0$, $\underline{p}_0 = 0$, $\underline{w}_0 = 0$, $\underline{r}_0 = \underline{b}$,

$$\text{for} \quad k = 1, \ldots, k_{\max} \tag{47}$$

$$\underline{p}_k \;=\; \underline{r}_{k-1} \;-\; \sum_{j=1}^{k-1} \underline{p}_j \beta_j, \qquad \beta_j = \frac{r_{k-1}^T \underline{w}_j}{\underline{p}_j^T \underline{w}_j} \tag{48}$$

$$\underline{w}_k \;=\; A\underline{p}_k, \quad \alpha_k \;=\; \frac{\underline{p}_k^T \underline{r}_{k-1}}{\underline{p}_k^T \underline{w}_k} \tag{49}$$

$$\underline{x}_k \;=\; \underline{x}_{k-1} \;+\; \alpha_k \underline{p}_k \tag{50}$$

$$\underline{r}_k \;=\; \underline{r}_{k-1} \;-\; \alpha_k \underline{w}_k. \tag{51}$$

- It is common to terminate when $\|r_k\|_2 \leq tol$, which of course corresponds to $\|e_k\|_{A^TA} \leq tol$, but other stopping criteria are possible.

- **A critical observation** is that $\underline{p}_i^T A \underline{p}_j = \underline{p}_i^T \underline{w}_j = 0$ for all $i \neq j$. From this it is easy to show that $\beta_j = 0$ for all $j < k-1$ in (48), such that we do not need to store all the $\underline{p}_j$ and $\underline{w}_j$ vectors.

- Thus, the conjugate gradient (CG) algorithm constructs the best-fit approximation for $\underline{x}_k \in \mathbb{P}_{k-1}(A)\underline{b}$, with respect to the A-norm, using only $O(n)$ work and storage.

- A condensed variant of CG is given below. Starting with $\underline{x} = 0$, $\underline{p} = 0$, $\underline{w} = 0$, $\underline{r} = \underline{b}$, and $\rho_1 = 1$;

$$\text{for} \quad k = 1, \ldots, k_{\max} \tag{52}$$

$$\rho_0 = \rho_1, \quad \rho_1 = \underline{r}^T \underline{r} \tag{53}$$

$$\beta = \frac{\rho_1}{\rho_0} \tag{54}$$

$$\underline{p} = \underline{r} + \beta \underline{p}, \tag{55}$$

$$\underline{w} = A\underline{p}, \quad \alpha = \frac{\rho_1}{\underline{p}^T \underline{w}} \tag{56}$$

$$\underline{x} = \underline{x} + \alpha \underline{p} \tag{57}$$

$$\underline{r} = \underline{r} - \alpha \underline{w}. \tag{58}$$

- Notice that the storage for this unpreconditioned CG code is $4n$: n values each for $\underline{x}$, $\underline{r}$, $\underline{p}$, and $\underline{w}$, plus whatever storage is required to effect the action of A upon $\underline{p}$.

- On a parallel computer, the time-consuming parts of the algorithm are primarily the matrix-vector product, $\underline{w} = A\underline{p}$, and the inner-products, $\underline{r}^T \underline{r}$ and $\underline{p}^T \underline{w}$.

- In the preconditioned variant it's often the case that applying the preconditioner is the most expensive step.

Short-Term Recurrence for Orthogonalization

- Let's summarize the basic space relationships.

- From (55) we have that

$$\underline{p}_k \in [\underline{r}_0\, \underline{r}_1 \ \ldots\ \underline{r}_{k-1}] = R_k, \tag{59}$$

where for conciseness we imply that, for any matrix C_k, $\underline{c} \in C_k$ means that $\underline{c}$ is in, $\mathcal{R}(C_k)$, which is the column space of C_k.

- So, we have

$$\begin{aligned} \underline{p}_k &\in [\underline{p}_1 \ \ \underline{p}_2 \ \ \ldots \ \ \underline{p}_k \ \ ] = P_k \\ &\in [\underline{r}_0 \ \ \underline{r}_1 \ \ \ldots \ \ \underline{r}_{k-1} \ \ ] = R_k \\ &\in [\underline{b} \ \ A\underline{b} \ \ \ldots \ \ A^{k-1}\underline{b} \ ] = K_k(A, \underline{b}). \end{aligned} \tag{60}$$

- The range of the matrices P_k and R_k are consequently the same and equal to the Krylov subspace, $K_k(A, \underline{b})$.

- From $\underline{r}_k \equiv A\underline{e}_k$ and the best fit property of $\underline{x}_k$, we have

$$0 = \underline{e}_k^T A P_k = (A\underline{e}_k)^T P_k = \underline{r}_k^T P_k, \tag{61}$$

which is to say that the residual $\underline{r}_k$ is L^2-orthogonal to the search space, P_k.

- Note that $A\underline{p}_j \in P_k$ for $j = 1, \ldots, k-1$.

- Consequently,

$$\underline{p}_j^T A \underline{r}_{k-1} = 0, \ \ j = 1, \ldots, k-1. \tag{62}$$

- So we find that $\beta_j = 0$ for $j = 1, \ldots, k-2$ in (48).

- Moreover,

$$\underline{p}_{k-1}^T A \underline{r}_{k-1} \neq 0. \ \text{(Why?)} \tag{63}$$

Key Take-Aways: A SPD

- Richardson (Jacobi) / CG share the *same* approximation space, $K_k(A; \underline{b})$

- CG produces the *best-fit*, $\|\underline{x} - \underline{x}_k\|_A \ \leq \ \|\underline{x} - \underline{v}\|_A, \forall \ \underline{v} \ \in \ K_k$.

- For any approximation space, $\mathcal{V}_k$, the best-fit appoximation is the unique $\underline{x}_k$ satisfying, for all $\underline{v} \ \in \ \mathcal{V}_k$,

$$(\underline{v}, \underline{x}_k)_A \ = \ (\underline{v}, \underline{x})_A.$$

- If the columns of $P_k := [\underline{p}_1 \ \underline{p}_2 \ \cdots \underline{p}_k]$ satisfy

$$\underline{p}_k \ \in \ \mathcal{V}_k, \tag{64}$$

$$\underline{p}_i^T A \underline{p}_j \ = \ 0, \ \ i \neq j, \tag{65}$$

  then the projection can be computed using a short-term recurrence,

$$\underline{x}_k \ = \ \underline{x}_{k-1} \ + \ \frac{(\underline{p}_k, \underline{x})_A}{(\underline{p}_k, \underline{p}_k)_A},$$

  which is *computable*.

- If the columns of $\mathcal{V}_k = K_k$, than the orthogonalization (65) can also be computed with a short-term recurrence.

- In this case, the work and storage for CG is $O(n)$ per iteration.