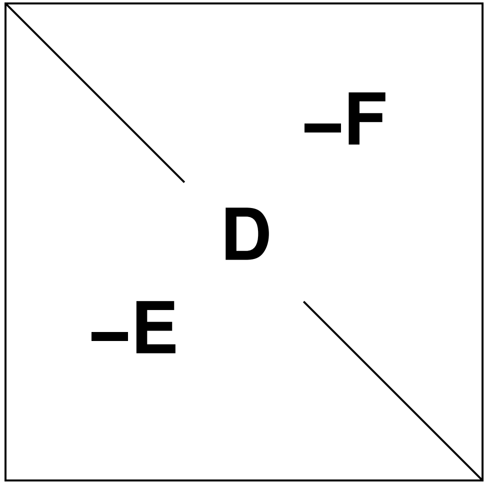
Iterative methods: relaxation techniques and projection methods

- *Basic relaxation methods: Jacobi, Gauss-Seidel, SOR*
- *Convergence results*
- *Introduction to projection-type techniques*
- *Sample one-dimensional Projection methods*
- *Some theory and interpretation –*
- *See Chapter 4 and Chapter 5 of text for details.*

Linear Systems: Basic Relaxation Schemes

Relaxation schemes: based on the decomposition $A = D - E - F$



D = diag(A), $-E$ = strict lower part of A and $-F$ its strict upper part.

➤ For example, Gauss-Seidel iteration :

$$(D - E)x^{(k+1)} = Fx^{(k)} + b$$

➤ Most common techniques 60 years ago.

➤ Now: used as smoothers in Multigrid or as preconditioners

Note: If $\rho_i^{(k)}$ = ith component of current residual $b - Ax$ then relaxation version of GS is:

$$\xi_i^{(k+1)} = \xi_i^{(k)} + \frac{\rho_i^{(k)}}{a_{ii}}$$

for $i = 1, \cdots, n$

Iteration matrices

➤ Jacobi, Gauss-Seidel, SOR, & SSOR iterations are of the form

$$x^{(k+1)} = Mx^{(k)} + f$$

- $M_{Jac} = D^{-1}(E + F) = I - D^{-1}A$
- $M_{GS}(A) = (D - E)^{-1}F = I - (D - E)^{-1}A$

SOR relaxation: $\xi_i^{(k+1)} = \omega \xi_i^{(GS,k+1)} + (1 - \omega)\xi_i^{(k)}$

- $M_{SOR}(A) = (D - \omega E)^{-1}(\omega F + (1 - \omega)D) = I - (\omega^{-1}D - E)^{-1}A$
- Related Splitting: $(D - \omega E)x^{(k+1)} = [\omega F + (1 - \omega)D]x^{(k)} + \omega b$

✍1 Matlab: take a look at: *gs.m*, *sor.m*, and *sorRelax.m* in iters/

Iteration matrices

Previous methods based on splitting A as: $A = M - N$

$$Mx = Nx + b \quad \to \quad Mx^{(k+1)} = Nx^{(k)} + b \to$$

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b \equiv Gx^{(k)} + f$$

Jacobi, Gauss-Seidel, SOR, & SSOR iterations are of the form

$$\begin{aligned} G_{Jac} &= D^{-1}(E + F) = I - D^{-1}A \\ G_{GS} &= (D - E)^{-1}F = I - (D - E)^{-1}A \\ G_{SOR} &= (D - \omega E)^{-1}(\omega F + (1 - \omega)D) \\ &= I - (\omega^{-1}D - E)^{-1}A \\ G_{SSOR} &= I - \omega(2 - \omega)(D - \omega F)^{-1}D(D - \omega E)^{-1}A \end{aligned}$$

General convergence result

Consider the iteration: $x^{(k+1)} = Gx^{(k)} + f$

(1) Assume that $\rho(G) < 1$. Then $I - G$ is non-singular and G has a fixed point. Iteration converges to a fixed point for any f and $x^{(0)}$.

(2) If iteration converges for any f and $x^{(0)}$ then $\rho(G) < 1$.

Example: Richardson's iteration

$$x^{(k+1)} = x^{(k)} + \alpha(b - Ax^{(k)})$$

✍2 Assume $\Lambda(A) \subset \mathbb{R}$. When does the iteration converge?

A few well-known results

➤ Jacobi and Gauss-Seidel converge for diagonal dominant matrices, i.e., matrices such that

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, i = 1, \cdots, n$$

➤ SOR converges for $0 < \omega < 2$ for SPD matrices

➤ The optimal ω is known in theory for an important class of matrices called 2-cyclic matrices or matrices with property A.

➤ A matrix has property A if it can be (symmetrically) permuted into a 2×2 block matrix whose diagonal blocks are diagonal.

$$PAP^T = \begin{bmatrix} D_1 & E \\ E^T & D_2 \end{bmatrix}$$

➤ Let A be a matrix which has property A. Then the eigenvalues λ of the SOR iteration matrix and the eigenvalues μ of the Jacobi iteration matrix are related by

$$(\lambda + \omega - 1)^2 = \lambda \omega^2 \mu^2$$

➤ The optimal ω for matrices with property A is given by

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho(B)^2}}$$

where B is the Jacobi iteration matrix.

An observation & Introduction to Preconditioning

➤ The iteration $x^{(k+1)} = Mx^{(k)} + f$ is attempting to solve $(I - M)x = f$. Since M is of the form $M = I - P^{-1}A$ this system can be rewritten as

$$P^{-1}Ax = P^{-1}b$$

where for SSOR, we have

$$P_{SSOR} = (D - \omega E)D^{-1}(D - \omega F)$$

referred to as the SSOR 'preconditioning' matrix.

In other words:

Relaxation Scheme $\Longleftrightarrow$ *Preconditioned Fixed Point Iteration*

Projection Methods

➤ The main idea of projection methods is to extract an approximate solution from a subspace.

➤ We define a subspace of approximants of dimension m and a set of m conditions to extract the solution

➤ These conditions are typically expressed by orthogonality constraints.

➤ This defines one basic step which is repeated until convergence (alternatively the dimension of the subspace is increased until convergence).

Example: Each relaxation step in Gauss-Seidel can be viewed as a projection step

Background on projectors

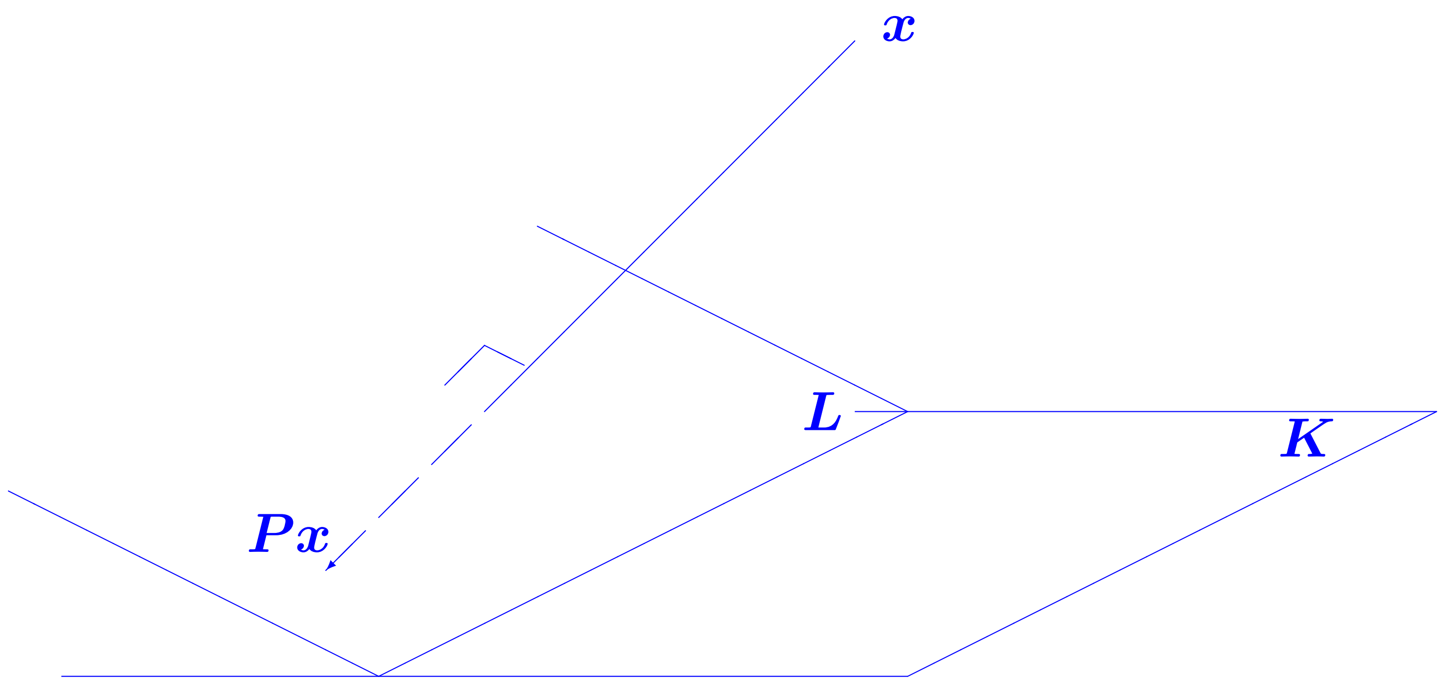
- P is a projector if it is idempotent: $P^2 = P$

Decomposition $\mathbb{R}^n = K \oplus S$ defines a (unique) projector P:

- From $x = x_1 + x_2$, set $Px = x_1$. In this case:
- $\mathbf{Ran}(P) = K$ and $\mathbf{Null}(P) = S$; $dim(K) = m \rightarrow dim(S) = n - m$.

- Pb: express mapping $x \rightarrow u = Px$ in terms of K, S
- Note $u \in K$, $x - u \in S$
- Express 2nd part with m constraints: let $L = S^\perp$, then

$u = Px$ iff $\begin{cases} u \in K \\ x - u \perp L \end{cases}$ ➤ Projection onto K and orthogonally to L



- Illustration: P projects onto K and orthogonally to L
- When $L = K$ projector is orthogonal.
- Note: $Px = 0$ iff $x \perp L$.

Projection methods for linear systems

- Initial Problem: $b - Ax = 0$

- Given two subspaces K and L of $\mathbb{R}^N$ of dimension m, define ...

Approximate problem: Find $\tilde{x} \in K$ such that $\underbrace{b - A\tilde{x} \perp L}_{\text{Petrov-Galerkin cond.}}$

- m degrees of freedom (K) + m constraints (L) $\to$

- To solve: A small linear system ('projected problem')

- Basic projection step. Typically a sequence of such steps are applied

➤ With a nonzero initial guess x_0, approximate problem is

$$\text{Find} \quad \tilde{x} \in x_0 + K \quad \text{such that} \quad b - A\tilde{x} \perp L$$

Write $\tilde{x} = x_0 + \delta$ and $r_0 = b - Ax_0$. $\rightarrow$ system for δ:

$$\text{Find } \delta \ \in K \text{ such that } r_0 - A\delta \perp L$$

✍3 Formulate Gauss-Seidel as a projection method -

✍4 Generalize Gauss-Seidel by defining subspaces consisting of 'blocks' of coordinates $\mathbf{span}\{e_i, e_{i+1}, ..., e_{i+p}\}$

Matrix representation:

Let

- $V = [v_1, \ldots, v_m]$ a basis of K &
- $W = [w_1, \ldots, w_m]$ a basis of L

➤ Write approximate solution as $\tilde{x} = x_0 + \delta \equiv x_0 + Vy$ where $y \in \mathbb{R}^m$. Then Petrov-Galerkin condition yields:

$$W^T(r_0 - AVy) = 0$$

➤ Therefore,

$$\tilde{x} = x_0 + V[W^T AV]^{-1} W^T r_0$$

Remark: In practice $W^T AV$ is known from algorithm and has a simple structure [tridiagonal, Hessenberg,..]

Prototype Projection Method

Until Convergence Do:

1. Select a pair of subspaces K, and L;

2. Choose bases: $V = [v_1, \ldots, v_m]$ for K and $W = [w_1, \ldots, w_m]$ for L.

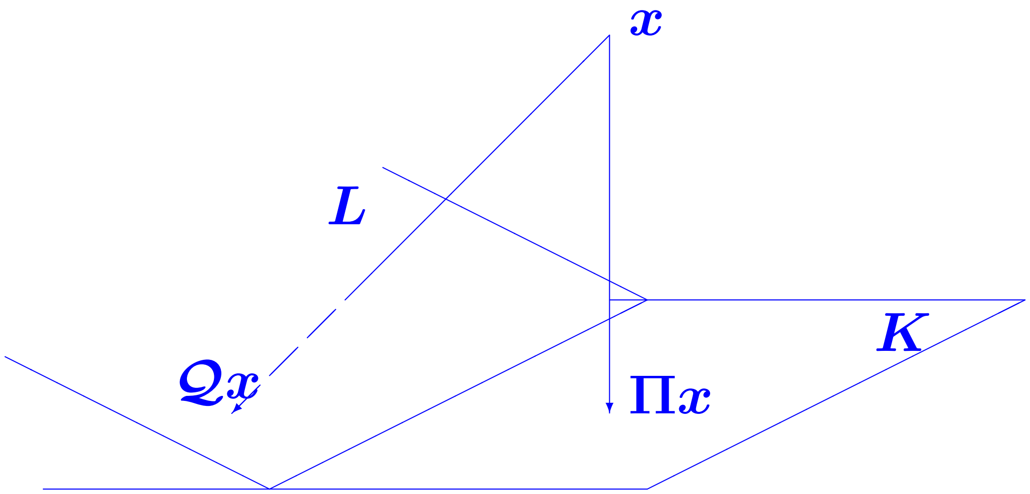
3. Compute :

$$r \leftarrow b - Ax,$$
$$y \leftarrow (W^T AV)^{-1} W^T r,$$
$$x \leftarrow x + Vy.$$

Projection methods: Operator form representation

➤ Let Π = the orthogonal projector onto K and $\mathcal{Q}$ the (oblique) projector onto K and orthogonally to L.

$$\Pi x \in K, \quad x - \Pi x \perp K$$
$$\mathcal{Q} x \in K, \quad x - \mathcal{Q} x \perp L$$



Assumption: no vector of K is $\perp$ to L

In the case $x_0 = 0$, approximate problem amounts to solving

$$\mathcal{Q}(b - Ax) = 0, \quad x \quad \in K$$

or in operator form (solution is Πx)

$$\mathcal{Q}(b - A\Pi x) = 0$$

Question: what accuracy can one expect? Let x^* be the exact solution

1) We can't do better than $\|(I - \Pi)x^*\|_2$: $\|\tilde{x} - x^*\|_2 \geq \|(I - \Pi)x^*\|_2$

2) The residual of the exact solution for the approximate problem satisfies:

$$\|b - \mathcal{Q}A\Pi x^*\|_2 \leq \|\mathcal{Q}A(I - \Pi)\|_2 \|(I - \Pi)x^*\|_2$$

Two Important Particular Cases.

1. $L = K$

- When A is SPD then $\|x^* - \tilde{x}\|_A = \min_{z \in K} \|x^* - z\|_A$.
- Class of Galerkin or Orthogonal projection methods
- Important member of this class: Conjugate Gradient (CG) method

2. $L = AK$.

In this case $\|b - A\tilde{x}\|_2 = \min_{z \in K} \|b - Az\|_2$

- Class of Minimal Residual Methods: CR, GCR, ORTHOMIN, GMRES, CGNR, ...

One-dimensional projection processes

$$K = span\{d\}$$
and
$$L = span\{e\}$$

Then $\tilde{x} = x + \alpha d$. Condition $r - A\delta \perp e$ yields

$$\alpha = \frac{(r,e)}{(Ad,e)}$$

- Three popular choices:

(1) Steepest descent

(2) Minimal residual iteration

(3) Residual norm steepest descent

1. Steepest descent.

A is SPD. Take at each step $d = r$ and $e = r$.

Iteration: $$\begin{aligned} & r \leftarrow b - Ax, \\ & \alpha \leftarrow (r, r)/(Ar, r) \\ & x \leftarrow x + \alpha r \end{aligned}$$

➤ Each step minimizes $f(x) = \|x - x^*\|_A^2 = (A(x - x^*), (x - x^*))$ in direction $-\nabla f$.

➤ Convergence guaranteed if A is SPD.

✍5 As is formulated, the above algorithm requires 2 'matvecs' per step. Reformulate it so only one is needed.

Convergence based on the Kantorovitch inequality: Let B be an SPD matrix, λ_{max}, λ_{min} its largest and smallest eigenvalues. Then,

$$\frac{(Bx,x)(B^{-1}x,x)}{(x,x)^2} \le \frac{(\lambda_{max}+\lambda_{min})^2}{4\,\lambda_{max}\lambda_{min}}, \quad \forall x \neq 0.$$

➤ This helps establish the convergence result

Let A an SPD matrix. Then, the A-norms of the error vectors $d_k = x_* - x_k$ generated by steepest descent satisfy:

$$\|d_{k+1}\|_A \le \frac{\lambda_{max}-\lambda_{min}}{\lambda_{max}+\lambda_{min}} \|d_k\|_A$$

➤ Algorithm converges for any initial guess x_0.

Proof: Observe $\|d_{k+1}\|_A^2 = (Ad_{k+1}, d_{k+1}) = (r_{k+1}, d_{k+1})$

- by substitution,

$$\|d_{k+1}\|_A^2 = (r_{k+1}, d_k - \alpha_k r_k)$$

- By construction $r_{k+1} \perp r_k$ so we get $\|d_{k+1}\|_A^2 = (r_{k+1}, d_k)$. Now:

$$\begin{aligned}\|d_{k+1}\|_A^2 &= (r_k - \alpha_k A r_k, d_k) \\ &= (r_k, A^{-1} r_k) - \alpha_k (r_k, r_k) \\ &= \|d_k\|_A^2 \left(1 - \frac{(r_k, r_k)}{(r_k, A r_k)} \times \frac{(r_k, r_k)}{(r_k, A^{-1} r_k)}\right).\end{aligned}$$

Result follows by applying the Kantorovich inequality. ■

2. Minimal residual iteration.

A positive definite ($A + A^T$ is SPD). Take at each step $d = r$ and $e = Ar$.

Iteration:
$$\begin{array}{l} r \leftarrow b - Ax, \\ \alpha \leftarrow (Ar, r)/(Ar, Ar) \\ x \leftarrow x + \alpha r \end{array}$$

- Each step minimizes $f(x) = \|b - Ax\|_2^2$ in direction r.
- Converges under the condition that $A + A^T$ is SPD.

✍6 As is formulated, the above algorithm would require 2 'matvecs' at each step. Reformulate it so that only one matvec is required

Convergence

Let A be a real positive definite matrix, and let

$$\mu = \lambda_{min}(A + A^T)/2, \quad \sigma = \|A\|_2.$$

Then the residual vectors generated by the Min. Res. Algorithm satisfy:

$$\|r_{k+1}\|_2 \leq \left(1 - \frac{\mu^2}{\sigma^2}\right)^{1/2} \|r_k\|_2$$

- In this case Min. Res. converges for any initial guess x_0.

Proof: Similar to steepest descent. Start with

$$\begin{aligned}\|r_{k+1}\|_2^2 &= (r_{k+1}, r_k - \alpha_k A r_k)\\ &= (r_{k+1}, r_k) - \alpha_k (r_{k+1}, A r_k).\end{aligned}$$

By construction, $r_{k+1} = r_k - \alpha_k A r_k$ is $\perp A r_k$, so:
$\|r_{k+1}\|_2^2 = (r_{k+1}, r_k) = (r_k - \alpha_k A r_k, r_k)$. Then:

$$\begin{aligned}\|r_{k+1}\|_2^2 &= (r_k, r_k) - \alpha_k (A r_k, r_k)\\ &= \|r_k\|_2^2 \left(1 - \frac{(A r_k, r_k)}{(r_k, r_k)} \frac{(A r_k, r_k)}{(A r_k, A r_k)}\right)\\ &= \|r_k\|_2^2 \left(1 - \frac{(A r_k, r_k)^2}{(r_k, r_k)^2} \frac{\|r_k\|_2^2}{\|A r_k\|_2^2}\right).\end{aligned}$$

Result follows from the inequalities $(Ax, x)/(x, x) \geq \mu > 0$ and $\|A r_k\|_2 \leq \|A\|_2 \, \|r_k\|_2$. ■

3. Residual norm steepest descent.

A is arbitrary (nonsingular). Take at each step $d = A^T r$ and $e = Ad$.

Iteration:
$$\begin{aligned} & r \leftarrow b - Ax, d = A^T r \\ & \alpha \leftarrow \|d\|_2^2 / \|Ad\|_2^2 \\ & x \leftarrow x + \alpha d \end{aligned}$$

➤ Each step minimizes $f(x) = \|b - Ax\|_2^2$ in direction $-\nabla f$.

➤ Important Note: equivalent to usual steepest descent applied to normal equations $A^T A x = A^T b$.

➤ Converges under the condition that A is nonsingular.

✍7 Take a look at *demo1DProj.m* in /iters.