1 March 2nd, 2021

1.1 General Framework of Stationary Iterations

1.1.1 Matrix Splitting

Given non singular matrix $A \in \mathbb{R}^{n\times n}$, we split it as:

$$A = M - N,$$

where $M, N \in \mathbb{R}^{n\times n}$. Then:

$$Ax = b \iff (M - N)x = b \iff Mx = Nx + b$$

Now, if we assume that M is easy to invert, e.g. diagonal, we can obtain:

$$\begin{aligned} & x = M^{-1}Nx + M^{-1}b \\ \iff & x = (I - M^{-1}A)x + M^{-1}b. \end{aligned}$$

We can then construct an iteration:

$$x_{k+1} = (I - M^{-1}A)x_k + M^{-1}b$$

For different stationary iterations, we have:

Jacobi: $M = A$

Gauss-Seidel: $M = D - E$

Backward Gauss-Seidel: $M = D - F$

SOR: $M = \frac{1}{\omega}(D - \omega E)$

For the convergence, the algorithm converges to the solution of $Ax = b$ with any x_0 if and only if $\rho(I - M^{-1}A) < 1$.

1.1.2 Preconditioned Richardson Iteration

Assume A is SPD. Solve $Ax = b$ is the same as solving the optimizaiton problem:

$$\min_{x\in\mathbb{R}^n} f(x), \quad f(x) = \frac{1}{2}x^TAx - x^Tb$$

This is because:

$$\nabla f(x) = Ax - b \implies \nabla^2 f(x) = A \implies f \text{ is convex}$$

Remark 1.1 — Since A is SPD, $f(x)$ is strongly convex.

Since this is a convex optimization problem, we can apply gradient descent:

$$\begin{aligned} & x_{k+1} = x_k - \alpha\nabla f(x_k) \\ \implies & x_{k+1} = x_k - \alpha\nabla\alpha(Ax_k - b) \end{aligned}$$

i.e.

$$x_{k+1} = (I - \alpha A)x_k + \alpha b$$

where $\alpha > 0$ is a constant. This is called the **Richardson Iteration**.

Remark 1.2 — Richardson is a special case of matrix splitting where $M = \frac{1}{\alpha}I$.

For the convergence, we have:

$$G = I - \alpha A.$$

Let $\Lambda(A) = \{\lambda : \lambda \text{ is an eigenvalue of A}\}$. We have:

$$\rho(G) = \max_{\lambda \in \Lambda(A)} |1 - \alpha\lambda|$$

If we let λ_{min} and λ_{max} be the min and max eigenvalues of A, we have:

$$\rho(G) = \max\{|1 - \alpha\lambda_{\text{min}}|, |1 - \alpha\lambda_{\text{max}}|\}$$

Since $|1 - \alpha\lambda|$ is a piecewise linear function. By direct calculation, we have:

$$\rho(G) < 1 \implies |1 - \alpha\lambda_{\text{max}}| = \alpha\lambda_{\text{max}} - 1 < 1 \implies \alpha < \frac{2}{\lambda_{\text{max}}}$$

Thus, we have:

$$\alpha \in (0, \frac{2}{\lambda_{\text{max}}})$$

for the iteration to converge.

In order to have optimal convergence speed, we consider:

$$\alpha_{\text{ opt}} = \arg\min_{\alpha} \rho(G) \iff \min_{\alpha > 0} \max_{\lambda \in \Lambda(A)} |1 - \alpha\lambda|$$

Then it is easy to check that:

$$1 - \alpha_{\text{opt}} = \alpha_{\text{ opt}}\lambda_{\text{max}} - 1 \implies \alpha_{\text{ opt}} = \frac{2}{\lambda_{\text{min}} + \lambda_{\text{max}}}$$

and

$$\rho_{\text{ opt}}(G) = 1 - \alpha_{\text{ opt}}\lambda_{\text{min}} = \frac{\lambda_{\text{max}} - \lambda_{\text{min}}}{\lambda_{\text{max}} + \lambda_{\text{min}}} = \frac{\gamma - 1}{\gamma + 1}$$

where $\gamma = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} = \frac{\|A\|_2}{1/\|A^{-1}\|_2} = \|A\|_2 \cdot \|A^{-1}\|_2$ which is the **condition number** of A as shown in Assignment 1.

Remark 1.3 — The convergence is slow if γ is big, as such we want to see if we can improve it.

Remark 1.4 — Intuitively, we would have a slow convergence if we have a flat level set. Meanwhile, if we have a round level set, the gradient descent would be fast. This is because of the ratio of λ_{max} and λ_{min}.

In addition, we should note that the gradient depends on the inner product in $\mathbb{R}^n$. As such, to improve the Richardson iteration, we change the inner product such that the level set of $f(x)$ in the new inner product space is very round.

Definition 1.5 (**Weighted Inner Product**).

$$\langle x, y\rangle_P = x^T Py, \qquad \text{where} P \text{ is SPD.}$$

Under the weighted inner product space, since:

$$f(y) = f(x) + \langle y - x, Ax - b\rangle + o(\|x - y\|_2)$$

we have:

$$f(y) = f(x) + \langle y - x, P^{-1}(Ax - b)\rangle_P + o(\|x - y\|_P)$$

Thus, we have:

$$\nabla_P f(x) = P^{-1}(Ax - b)$$

> **Remark 1.6** — This is because the gradient ($\langle y-x, Ax-b\rangle$) is a linear approximation at point x. This is the definition of the **Frechet derivative** in Hilbert spaces.

> **Remark 1.7** — $o(\|x-y\|_2) = o(\|x-y\|_P)$ since vector norms are equivalent in finite dimensional space.

As such, the gradient descent under weighted inner product is:

$$x_{k+1} = x_k - \alpha P^{-1}(Ax_k - b)$$

Note that α can be absorbed into P^{-1} since P is SPD, thus giving us:

$$x_{k+1} = x_k - P^{-1}(Ax_k - b) \iff x_{k+1} = (I - P^{-1}A)x_k + P^{-1}b$$

This is called the **preconditioned gradient descent**. Similarly, for the convergence, we have:

$$\rho(I - P^{-1}A) < 1$$

and the optimal convergence rate is:

$$p_{\text{ opt}} = \frac{\gamma(P^{-1}A) - 1}{\gamma(P^{-1}A) + 1}$$

where $\gamma(P^{-1}A)$ is the condition number of $P^{-1}A$.

> **Remark 1.8** — Note that the condition number of P before and after absorbing α is the same, since we are just scaling it.

To be a good preconditioner, P has to satisfy the following:

1. P is SPD.
2. P is easy to invert so that P^{-1} is easy to compute
3. $\gamma(P^{-1}A)$ has to be small (or equivalent $P \approx A$).

There are a few special cases:

- $P = D$ (diagonal part of A) - Jacobi iteration
- Symmetric G-S

1.1.3 Projection Methods

Let K and L be two m-dimensional subspaces in $\mathbb{R}^n$. Given $x_0 \in \mathbb{R}^n$, we obtain a better solution $\tilde{x}$ of $Ax = b$ by:

$$\begin{cases} \text{Find} & \tilde{x} \in x_0 + K \\ \text{s.t.} & b - A\tilde{x} \perp L \end{cases} \iff \begin{cases} \tilde{x} = x_0 + \delta, & \delta \in K \\ \langle b - A(x_0 + \delta), \omega \rangle = 0, & \forall \omega \in L \end{cases}$$

A pictorial illustration is shown in Figure 1.

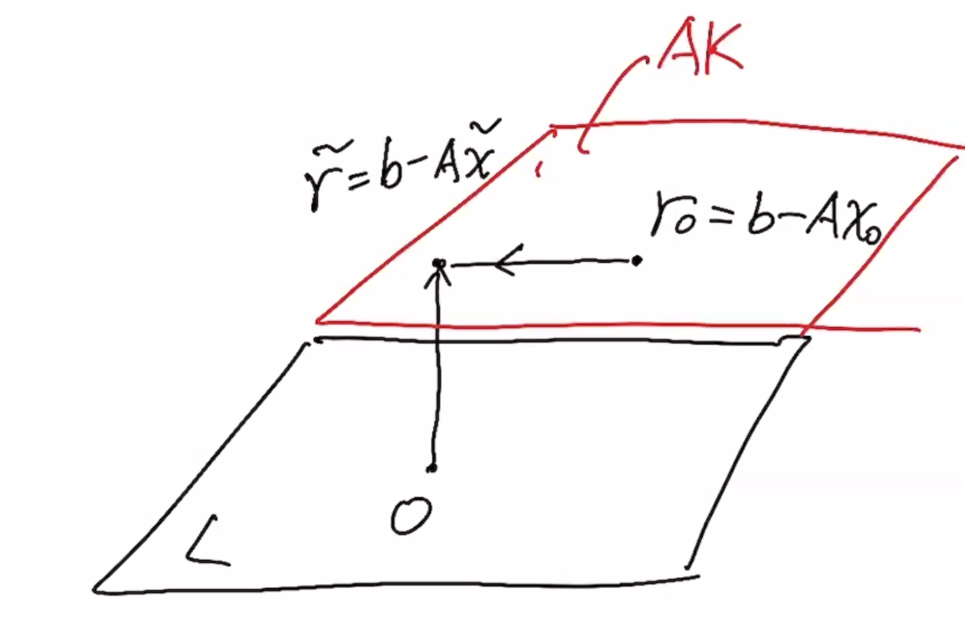

Figure 1: Pictorial Representation of the Projection Method

If we choose $K = L = \text{span}\{e_i\}$

- $\tilde{x} = x_0 + \delta$, $\delta \in \text{span}\{e_i\}$ (only the i-th component of x_0 is changed)
- $b - A\tilde{x} \perp \text{span}\{e_i\}$ (the i-th equation has an error 0).

we obtain Gauss-Seidel.

There are a few other variants of the projection methods. For example, we can choose two families of subspaces: K_i, L_i, $i = 1, \ldots \ell$. Given x_0, we obtain $\tilde{x}$ by:

$$\tilde{x} = x_0 + \delta_1 + \ldots + \delta_\ell, \quad \text{where} \begin{cases} \delta_i \in K_i \\ b - A(x_0 + \delta_i) \perp L_i \end{cases}$$

If we have $K_i = L_i = \text{span}\{e_i\}$ we have the Jacobi iteration.

We can have several other choices of K and L:

Multigrid Method: $K = L = \text{span}\{e_1\} \ldots \text{span}\{e_n\}$ on fine grid.

Then we do $\text{span}\{e_1\} \ldots \text{span}\{e_{n/2}\}$ on the coarse grid, etc.

Domain Decomposition: We first partition Ω into overlapping spaces into Ω_1, Ω_2, and then we set $K = L = \text{span}\{e_i, i \in \Omega_1\}$, and then $\text{span}\{e_i, i \in \Omega_2\}$.

Remark 1.9 — For both the methods mentioned above, K and L are fixed, making the iterative methods fixed.