1 March 4th, 2021

1.1 Advanced Iterative Methods I

In this chapter, we will develop non-stationary iterative methods. We will first do this using the projection method, and then improve it by incorporating preconditioning. Throughout this chapter, we assume $Ax = b$, where A is SPD.

1.1.1 Projection Methods with SPD Matrices

Recall that for projection methods, the problem is reformulated as: Given x_0, generate $\tilde{x}$ by:

$$\begin{cases} \text{Find} & \tilde{x} \in x_0 + K \\ \text{s.t.} & b - A\tilde{x} \perp L \end{cases}$$

In other words, we are finding

In matrix terms,

- Let $V = \begin{bmatrix} v_1 & v_2 & \dots & v_m \end{bmatrix} \in \mathbb{R}^{n \times m}$ be a basis of K
- Let $W = \begin{bmatrix} w_1 & w_2 & \dots & w_m \end{bmatrix} \in \mathbb{R}^{n \times m}$ be a basis of L

Then:

$$\tilde{x} \in x_0 + K \implies \tilde{x} = x_0 + Vy \text{ for some } y \in \mathbb{R}^m$$

and

$$\begin{aligned} b - A\tilde{x} \perp L &\implies \langle b - A(x_0 + Vy), Wz \rangle = 0 \quad \forall z \in \mathbb{R}^m \\ &\iff W^T(b - A(x_0 + Vy)) = 0 \\ &\iff W^T AVy = W^T(Ax_0 - b). \end{aligned}$$

This means that each step, we only need to solve the system of linear equations: $W^T AVy = W^T(Ax_0 - b)$. Note that $W^T AV \in \mathbb{R}^{m \times m}$ which is smaller than n.

In order to preserve the structure of A (SPD-ness), we choose $K = L$ (so that $W = V$). Then, we need to solve:

$$V^T AVy = V^T(Ax_0 - b)$$

Remark 1.1 — Note that $V^T AV$ is SPD because A is.

Theorem 1.2
$\tilde{x}$ is optimal in the sense that:

$$\tilde{x} = \operatorname{argmin}_{x \in x_0 + K} \|x - x_*\|_A^2$$

where x_* is the true solution of $Ax = b$ and $\|\cdot\|_A$ is defined by $\|x\|_A = (x^T Ax)^{1/2}$.

In other words, if we project x_* into the subspace, then the resulting $\tilde{x}$ is the closest point.

Proof. Note that:

$$\begin{aligned} &\tilde{x} = \text{argmin}_{x \in x_0 + K} \|x - x_*\|_A^2 \\ \iff &\langle x_* - \tilde{x}, (x_0 + z) - \tilde{x} \rangle_A = 0 \quad \forall z \in K \\ \iff &\langle A(x_* - \tilde{x}), (x_0 + z) - \tilde{x} \rangle = 0 \quad \forall z \in K \\ \iff &\langle b - A\tilde{x}, (x_0 - \tilde{x}) + z \rangle = 0 \quad \forall z \in K. \end{aligned}$$

Since $\tilde{x}$ satisfies:

$$\begin{cases} \tilde{x} \in x_0 + K \implies x_0 - \tilde{x} \in K \\ \langle b - A\tilde{x}, z \rangle = 0 \quad \forall z \in K \end{cases}$$

$$\implies \langle b - A\tilde{x}, (x_0 - \tilde{x}) + z \rangle = 0 \quad \forall z \in K$$

□

If we let $P_K^{(A)}$ denote the projection onto K with $\| \cdot \|_A$, projection methods can be expressed as:

$$x_{k+1} = P_{x_k + K}^{(A)}(x_*), \quad k = 0, 1, 2, \ldots$$

Note that this means that error is non-increasing under A-norm, as:

$$\|x_{k+1} - x_*\|_A \leq \|x_K - x_*\|_A$$

However, this does not guarantee that the error converges to zero.

1.1.2 One-Dimensional Projection Methods

Now the question is how to choose K? In Gauss-Seidel, we choose the simplest one, i.e. e_i.

Given x_k, we choose K s.t. $\dim(K) = 1$, i.e.:

$$K = \text{span}\{d_k\}, \text{ where } d_k \in \mathbb{R}^n.$$

Remark 1.3 — d_k can be thought up as the direction.

Now we might ask what is the best d_k? We have:

$$x_{k+1} = x_k + \alpha_k d_k \text{ for some } \alpha_k \in \mathbb{R}$$

Assume we have fixed $\alpha_k \geq 0$ and $\|d_k\|_A = \beta$. We want $\|x_{k+1} - x_*\|_A^2$ minimized. This gives us:

$$\begin{aligned} \|x_{k+1} - x_*\|_A^2 &= \|(x_k + \alpha_k d_k) - x_*\|_A^2 \\ &= \|x_k - x_*\|_A^2 + \alpha_k^2 \|d_k\|_A^2 + 2\alpha_k \langle d_k, x_k - x_* \rangle_A. \end{aligned}$$

Note that $\|x_k - x_*\|_A^2$ and $\alpha_k^2\|d_k\|_A^2$ are constants, meaning that in order to minimize the error, we want:

$$\min_{d_k \in \mathbb{R}^n} \langle d_k, x_k - x_* \rangle_A$$

Note that the solution to this is:

$$d_k = -C(x_k - x_*)$$

where $C = \frac{\beta}{\|x_k - x_*\|_A} > 0$.

Remark 1.4 — The optimal d_k is in the opposite direction of $x_k - x_*$.

Finally, we choose:

$$K = \mathrm{span}\{d_k\} = \mathrm{span}\{x_k - x_*\}$$

However, we need to know x_*, which is not possible to know (since that is our goal). Thus this method is non-practical.

Remark 1.5 — This is optimal only if we fixed $\|d_k\|_A$.

Now let's consider fixing $\alpha_k \gtrsim 0$ and $\|d_k\|_2 = \beta$.

Remark 1.6 — $\alpha_k \gtrsim 0$ means that it is greater but approximately zero.

Again, we minimize $\|x_{k+1} - x_*\|_2^2$. Now we have:

$$\begin{aligned} \|x_{k+1} - x_*\|_A^2 &= \|(x_k + \alpha_k d_k) - x_*\|_A^2 \\ &= \|x_k - x_*\|_A^2 + \alpha_k^2\|d_k\|_A^2 + 2\alpha_k\langle d_k, x_k - x_* \rangle_A \quad \approx 2\alpha_k\langle d_k, x_k - x_* \rangle_A \quad (\text{since } \alpha_k \approx 0). \end{aligned}$$

Since α_k is a constant, we have:

$$\begin{aligned} &\min_{\|d_k\|_2 = \beta} \langle d_k, x_k - x_* \rangle_A \\ \iff &\min_{\|d_k\|_2 = \beta} \langle d_k, Ax_k - b \rangle. \end{aligned}$$

Remark 1.7 — Note that the 2-norm ball is an ellipsoid in $\mathbb{R}^n$ with A-norm, thus, we change it to use the standard inner product. After doing this, the optimal d_k is in the opposite direction of $Ax_k - b$.

The solution to this equation is:

$$d_k = -C(Ax_k - b)$$

where $c = \frac{\beta}{\|Ax_k - b\|_2} > 0$. Thus, the optimal choice of K is:

$$K = \mathrm{span}\{d_k\} = \mathrm{span}\{r_k\}, \quad \text{where } r_k = b - Ax_k$$

Now we have found the optimal K, but the next step is to find the optimal α_k.

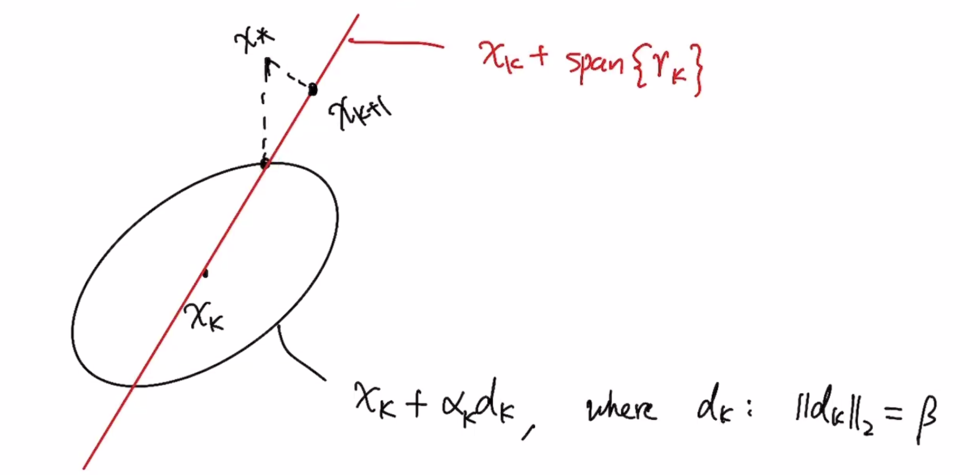

Figure 1: Pictorial Version of K

In other words:

$$\begin{aligned} &\min_{\alpha} \|(x_k + \alpha r_k) - x_*\|_A^2 \\ \iff &\min_{\alpha} \|(x_k - x_*)\|_A^2 + \alpha^2\|r_k\|_A^2 + 2\alpha\langle r_k, x_k - x_*\rangle_A. \end{aligned}$$

Takign derivative w.r.t. α and setting it to 0, we have:

$$2\alpha\|r_k\|_A^2 + 2\langle r_k, x_k - x_*\rangle_A = 0 \iff \alpha = -\frac{\langle r_k, x_k - x_*\rangle_A}{\|r_k\|_A^2} = -\frac{\langle r_k, Ax_k - b\rangle}{\|r_k\|_A^2} = \frac{\|r_k\|_2^2}{\|r_k\|_A^2}.$$

Thus the optimal 1D projection method is given in Algorithm 1.

Algorithm 1 Optimal 1D Projection Method

1: **for** $k = 0, 1, 2, \dots$ **do**
2: $r_k = b - Ax_k$
3: $\alpha_k = \|r_k\|_2^2 / \|r_k\|_A^2 = \frac{\langle r_k, r_k\rangle}{\langle Ar_k, r_k\rangle}$
4: $x_{k+1} = x_k + \alpha_k r_k$
5: **end for**

Remark 1.8 — Note that for each iteration, we use 2 matrix-vector products and $O(n)$ operations.

This can be improved to 1 matrix-vector product. If we have denote $p_k = Ar_k$, then:

$$\begin{aligned} r_{k+1} &= b - Ax_{k+1} \\ &= b - A(x_k + \alpha_k r_l) \\ &= b - Ax_k - \alpha_k Ar_k \\ &= r_k - \alpha_k p_k. \end{aligned}$$

This gives us Algorithm 2. Note that this is the algorithm Gradient descent for:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}x^T Ax - x^T b$$

Algorithm 2 Optimal 1D Projection Method Improved

$r_0 = b - Ax_0$
for $k = 0, 1, 2, \dots$ **do**
  $p_k = Ar_k$
  $\alpha_k = \frac{\langle r_k, r_k \rangle}{\langle p_k, r_k \rangle}$
  $x_{k+1} = x_k + \alpha_k x_k$
  $r_{k+1} = r_k - \alpha_k p_k$
end for

with the exact line search, i.e.:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

where $\alpha_k = \arg\min_{\alpha \in \mathbb{R}} f(x_k - \alpha \nabla f(x_k))$. Thus, the algorithm is called: **steepest descent**.