1 7th of December 2018 — F. Poloni

In this lecture we are interested in using Arnoldi method to solve linear systems.

We can use the *sparse eigenvalues funtion* that we saw in last lecture and the **G**eneralized **M**inimum **RES**idual (GMRES).

Our task is to approximate the solution of a large-scale linear system of the form $Ax = b$ and our approach is to look for "the closest thing to solution" inside $K_n(A, b)$.

Through Arnoldi of A, b and n, we obtained $[Q, H]$ and we can approximate the solution x as $Q_1 y_1 + Q_2 y_2 + \cdots + Q_n y_n = Qy$, which is a good approximation of the solution inside $K_n(A, b)$, formally

$$\min_{x \in K_n(A,b)} \|Ax - b\|, \ x = Q_n y.$$

which is equivalent to $\min_{y \in \mathbb{R}^n} \|AQ_n y - b\|$.

We can perform some more reductions and:

$$\begin{aligned} \|AQ_n y - b\| &\overset{(1)}{=} \|Q_{n+1} \underline{H}_n y - b\| \\ &\overset{(2)}{=} \|Q_{n+1} \underline{H}_n y - Q_{n+1} \|b\| e_1\| \\ &= \|Q_{n+1}(\underline{H}_n y - \|b\| e_1)\| \\ &\overset{(3)}{=} \|\underline{H}_n y - \|b\| e_1\|. \end{aligned} \tag{1.1}$$

where $\overset{(1)}{=}$ is due to the equivalence $AQ_n = Q_{n+1} H_n$, with $H_n \in M(n+1, n)$, $\overset{(2)}{=}$ follows from the fact that $q_1 = \frac{b}{\|b\|}$ and $\overset{(3)}{=}$ is explained recalling that Q_{n+1} is an orthogonal rectangular matrix in $M(mn + 1)$ and $\|z\| = \|Q_{n+1} z\|$, since $z^T z = z^T {Q_{n+1}}^T Q_{n+1} z$.

We got a LS problem of size $(n + 1) \times n$ (small), where $y \in \mathbb{R}^n$ and $e_1 \in \mathbb{R}^{n+1}$; moreover $\underline{H}$ has the following shape

$$H = \left[\begin{array}{cccc|cccc} * & \cdots & * & * & * & \cdots & \cdots & * \\ * & \cdots & * & * & * & \cdots & \cdots & * \\ 0 & \ddots & * & * & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & * & * & * & \cdots & \cdots & * \\ \hline & & & \circledast & * & \cdots & * & * \\ & & & & * & \cdots & * & * \\ & \mathbf{0} & & & 0 & \ddots & * & * \\ & & & & 0 & 0 & * & * \end{array}\right]$$

hence it is quite sparse.

$qr(H)$ can be computed in $O(n^2)$ using the fact that H is 'almost triangular' (Hessenberg matrix), although it is not a big optimization, since n Arnoldi steps need to be computed first.

Notice that instead of doing a QR at the end, we can compute QRs of $\underline{H}_1, \underline{H}_2, \ldots$ and update them at each step. This allows us to compute at each step a residual $\|Ax_n - b\|$ that we can use as stopping criterion.

Something on Matlab ...

Matlab has `gmres(A, b)` (and Python has `scipy.sparse.linalg.gmres`).

To estimate the convergence of GMRES we can see x as a plynomial ($x = p(A)b$, such that $p(t) = \alpha_0 + \alpha_1 t + \cdots + \alpha_{n-1}t^{n-1}$ is a polynomial of degree $n-1$).

As far as the residual is concerned $Ax - b = Ap(A)b - b = A(\alpha_0 I + \alpha_1 A + \cdots + \alpha_{n-1}A^{n-1})b - b = q(A)b$, where $q(t) = tp(t) - 1$. If $A = V\Lambda V^{-1}$ diagonalizable, then:

$$A^k = V \begin{bmatrix} {\lambda_1}^k & & \\ & \ddots & \\ & & {\lambda_m}^k \end{bmatrix} V^{-1} \text{ and } q(A) = V \begin{bmatrix} q(\lambda_1) & & \\ & \ddots & \\ & & q(\lambda_m) \end{bmatrix} V^{-1}$$

All this computation was needed to write the residual GMRES in a clearer form:

$$\begin{aligned} \min_{x \in K_n(A,b)} \|Ax - b\| &= \min_{\substack{q(x) = xp(x) - 1 \\ \text{of degree} \leq n}} \|Ap(A)b - b\| \\ &= \min_{\substack{q(x) = xp(x) - 1 \\ \text{of degree} \leq n}} \|q(A)b\| \\ &\leq (\min_{\cdots} \|q(A)\|)\, \|b\| \\ &= \min_{\cdots} \left\| V \begin{bmatrix} q(\lambda_1) & & \\ & \ddots & \\ & & q(\lambda_m) \end{bmatrix} V^{-1} \right\| \\ &\leq \min_{\cdots} \|V\| \left\| \begin{bmatrix} q(\lambda_1) & & \\ & \ddots & \\ & & q(\lambda_m) \end{bmatrix} \right\| \left\|V^{-1}\right\| \\ &\leq K(V) \left\| \min_{\cdots} \begin{bmatrix} q(\lambda_1) & & \\ & \ddots & \\ & & q(\lambda_m) \end{bmatrix} \right\| \end{aligned} \tag{1.2}$$

If A has very few distinct eigenvalues ($k \leq n$ of them), then we can find q such that $q(\lambda_i) = 0$ for all i and $q(0) = -1$, hence n steps of GMRES give us the exact solution.

If A has eigenvalues clustered in n points in the plane, we can find a polynomial q such that $q(\lambda_i)$ is small for all i.

Notice that Gauss operations on the rows of any matrix A (e.g. swapping rows or scalar multiplication of a row) change its eigenvalues, without changing the solution.

More generally, given $P \in M(n, \mathbb{R})$ we can change the problem $Ax = b$ to $PAx = Pb$. If P is invertible, the two systems have the same solution. However, the spectrum of PA may be much better (in the above sense) than the spectrum of A, leading to a faster solution with GMRES.

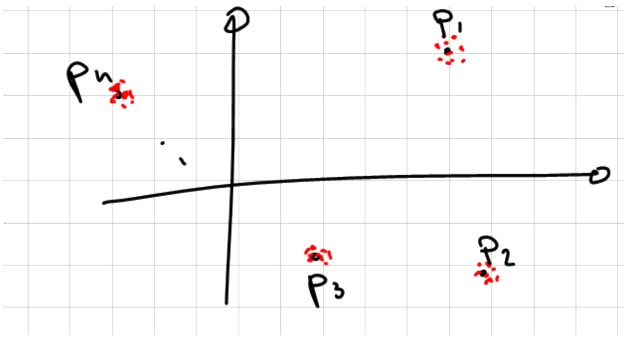

FIGURE 1.1: In this picture the eigenvalues are clustered around P_1, P_2, P_3 and P_4. We can find a polynomial q such that $q(\text{red points}) \approx 0$.

In particular, this happens if we manage to find $P \approx A^{-1}$. The perfect choice would be $P = A^{-1}$, but, of course, if we knew A^{-1} we would already have a way to solve linear systems: just compute the matrix multiplication $A^{-1}b$.

There are various techniques (often problem-dependent) to build effective *preconditioners* P. One comes from approximate LU factorizations of A (in a suitable sense); they are known as *incomplete* LU preconditioners.