QR-Algo (basic form)

$$\begin{cases} 1.\ A_0 := A \\ 2.\ A_\ell =: Q_\ell R_\ell \quad \text{(QR-decomp.)} \\ 3.\ A_{\ell+1} := R_\ell Q_\ell \quad (\to \text{conv to upper tri. form}) \end{cases}$$

efficiency:

$O(n^3)$ for each ℓ

so $O(n^4)$ in total $\to$ combine w. shift parameter to make more efficient

QR-Algo (Hessenberg form)

(A)

if A has Hessenb. form

$\Rightarrow$ cost of QR : $O(n^2)$

$\Rightarrow$ cost of multiplication with RQ : $O(n^2)$

QR-Algo (Deflation)

(B)

if $A = \begin{pmatrix} x & x & \cdots & - & - & x \\ x & - & & & & \vdots \\ & & & & & \\ * & & & & & x \\ 0 & 0 & \cdots & & 0 & \mu \end{pmatrix}$, $x \neq 0$, μ .. eval(A)

now one can look for λ_i on the submatrix

QR-Algo (with shift)

(tries to accelerate Deflation)

(C)

$$\begin{cases} 1.\ \text{init } A_0 := A \\ 2.\ \text{choose } \mu^{(\ell)} \\ 3.\ A_\ell - \mu^{(\ell)} := Q \cdot R \\ 4.\ A_{\ell+1} := R \cdot Q + \mu^{(\ell)} \end{cases}$$

$\mu^{(\ell)}$ can be chosen as $A_\ell(n,n)$.

The QR-Algo does implicitely an invers iter. for a certain starting value.
A good choice for μ is therefore a Rayleigh-quotient $\rightsquigarrow A_\ell(n,n)$

QR-Algo (combining (A)(B)(C))

$$\begin{cases} 1.\ \text{make } A \text{ Hessenberg}, \quad \mu^{(\ell)} := A_\ell(n,n) \\ \left.\begin{aligned} &2.\ A_\ell - \mu^{(\ell)} := Q \cdot R \\ &3.\ A_{\ell+1} = R \cdot Q + \mu \end{aligned}\right\} \text{until } A(n,n-1) \text{ is small} \\ 4.\ \text{recursion on sub matrix } A(1{:}n-1,\ 1{:}n-1) \end{cases}$$

8) Conjugate Gradient Method

iterative sol for $Ax^* = b$, A .. SPD

rule: dont compute A^{-1}, only use $x \mapsto A \cdot x$

Def.: $\begin{cases} (x,y)_A := x^T A y \\ K_\ell := \{r_0, Ar_0, \ldots A^{\ell-1} r_0\} \end{cases}$

$Ax^* = b \Leftrightarrow Ae_0 = r_0$

goal: approx x^* by $x_\ell = x_0 + e_\ell$

$\Rightarrow$ find $x_\ell \in x_0 + K_\ell$ st. $\|x^* - x_\ell\|_A \leq \|x^* - x\|_A \quad \forall x$

From that $\Rightarrow$ $(x^* - x_\ell, v)_A = 0 \quad \forall v \in K_\ell$

$(r_\ell, v)_2 = 0 \quad \forall v \in K_\ell$, $r_\ell := b - Ax_\ell$

One could solve $\ell \times \ell$ linear system, better: find x_ℓ as cheap update from $x_{\ell-1}$

Algor. CG

$$\begin{cases} d_\ell = r_\ell - \beta_{\ell-1} d_{\ell-1} \\ r_\ell = r_{\ell-1} - \alpha_\ell A d_{\ell-1} \\ x_\ell = x_{\ell-1} + \alpha_\ell d_{\ell-1} \end{cases} \qquad \begin{cases} \alpha_\ell = \dfrac{\|r_{\ell-1}\|_2^2}{\|d_{\ell-1}\|_A^2} \\ \beta_{\ell-1} = -\dfrac{\|r_\ell\|_2^2}{\|r_{\ell-1}\|_2^2} \end{cases}$$

Derivation: lect. notes

It is very economical wrt. memory req.

After N steps: exact solution

Convergence: $$\|x^* - x_\ell\|_A \le 2 \cdot \left| \frac{\sqrt{K} - 1}{\sqrt{K} + 1} \right|^\ell \cdot \|e_0\|_A$$

$$K := \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$$

GMRES iterative method for non-symm. A

seek $x_\ell \in x_0 + K_\ell$ s.t. $\|b - Ax_\ell\|_2 \le \|b - Ax\|_2 \quad \forall x$

we know $(b - Ax_\ell, v)_2 = 0 \ \forall v$, x_ℓ computed successively

Also exact after N steps!

How?
1. write $x_\ell = x_0 + Vy$, V.. orthog. basis of K_ℓ, $y \in \mathbb{R}^d$
2. $(b - Ax_\ell, v)_2 = 0 \Rightarrow \dots \Rightarrow x_\ell = x_0 + V\underbrace{(W^T A V)^{-1}}_{\in \mathbb{R}^{d \times d}} W^T r_0$
3. by constructing orthog. v_i $(V = (v_1, v_2 \dots))$

$\Rightarrow \|b - Ax_\ell\|_2 = \min_y \|\beta e_1 - H_\ell y\|_2$

Hessenberg

4. solve that for y using QR in $O(n^2)$ and $x_\ell = x_0 + V_\ell y_\ell$

Remark: if H_ℓ is not full rank: exact sol immediately

in practise if memory is full $\Rightarrow$ delete all and start with x_ℓ as x_0