

- (1) Suppose that an  $n$ -by- $n$  matrix  $A$  is symmetric and positive definite. Consider the following iteration:

$$\begin{aligned} A_0 &= A \\ \text{for } k &= 1, 2, 3, \dots \\ A_{k-1} &= G_k G_k^T \\ A_k &= G_k^T G_k \end{aligned}$$

where  $G_k G_k^T$  is the Cholesky factorization of a symmetric positive definite matrix.

Suppose we have a matrix  $A$  defined as above. Then,  $A$  has the LDLT decomposition as

$$A = LDL^T$$

where  $L$  is unit lower triangular and  $D$  is a positive diagonal matrix. Expanding further, we have

$$A = \underbrace{LD^{1/2}}_G \underbrace{(D^{1/2})^T L^T}_{G^T}.$$

Then, we have

$$G^T G = (D^{1/2})^T L^T L D^{1/2}$$

where  $L^T L$  is symmetric positive definite. Now, for any  $n$ -dimensional vector  $x$ , define the vector  $y = D^{1/2}x$ . Then,

$$x^T G^T G x = x^T (D^{1/2})^T L^T L D^{1/2} x = y^T (L^T L) y > 0$$

showing that  $G^T G$  is symmetric positive definite and so the iteration  $A_k = G_k^T G_k$  is well defined.

Now suppose we have the matrix

$$A = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

with  $a \geq c$  and  $A$  has eigenvalues  $\lambda_1 \geq \lambda_2 > 0$ . Then, if  $A_0 = A$ , our iteration yields

$$A_{k+1} = G_k^T G_k = G_k^{-1} G_k G_k^T G_k = G_k^{-1} A_k G_k$$

which implies

$$\begin{aligned} A_{k+1} &= G_k^{-1} A_k G_k \\ &= G_k^{-1} G_{k-1}^{-1} A_{k-1} G_{k-1} G_k \\ &\quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ &= G_k^{-1} G_{k-1}^{-1} \cdots G_0^{-1} A_0 G_0 \cdots G_{k-1} G_k \\ &= \bar{G}_k^{-1} A_0 \bar{G}_k \end{aligned}$$

where  $\bar{G}_k = G_0 \cdots G_{k-1} G_k$ . This shows that the  $(k+1)$ th iteration is similar to  $A$  and thus has the same eigenvalues as  $A$ . But, as  $k$  increases,  $\bar{G}_k$  grows larger and so

(2) Compute a QR step with the matrix

$$A = \begin{pmatrix} 2 & \varepsilon \\ \varepsilon & 1 \end{pmatrix}$$

(a) without a shift

First, we compute the QR decomposition of  $A$  as

$$A = \underbrace{\begin{pmatrix} \frac{2}{\sqrt{\varepsilon^2+4}} & \frac{\varepsilon \operatorname{sign}(\varepsilon^2-2)}{\sqrt{\varepsilon^2+4}} \\ \frac{\varepsilon}{\sqrt{\varepsilon^2+4}} & \frac{-2\operatorname{sign}(\varepsilon^2-2)}{\sqrt{\varepsilon^2+4}} \end{pmatrix}}_Q \underbrace{\begin{pmatrix} \sqrt{\varepsilon^2+4} & \frac{3\varepsilon}{\sqrt{\varepsilon^2+4}} \\ 0 & \frac{|\varepsilon^2-2|}{\sqrt{\varepsilon^2+4}} \end{pmatrix}}_R.$$

Then, we have the first QR step of  $A$  as

$$A_1 = RQ = \begin{pmatrix} 5 - \frac{12}{\varepsilon^2+4} & \operatorname{sign}(\varepsilon^2-2) \left( \varepsilon - \frac{6\varepsilon}{\varepsilon^2+4} \right) \\ \frac{\varepsilon|\varepsilon^2-2|}{\varepsilon^2+4} & \frac{-2(\varepsilon^2-2)}{\varepsilon^2+4} \end{pmatrix}$$

(b) with the shift  $\mu = 1$ .

First, we compute the QR decomposition of  $A - \mu I$  as

$$A - I = \underbrace{\begin{pmatrix} \frac{1}{\sqrt{\varepsilon^2+1}} & \frac{\varepsilon}{\sqrt{\varepsilon^2+1}} \\ \frac{\varepsilon}{\sqrt{\varepsilon^2+1}} & \frac{-1}{\sqrt{\varepsilon^2+1}} \end{pmatrix}}_Q \underbrace{\begin{pmatrix} \sqrt{\varepsilon^2+1} & \frac{\varepsilon}{\sqrt{\varepsilon^2+1}} \\ 0 & \frac{\varepsilon^2+1}{\sqrt{\varepsilon^2+1}} \end{pmatrix}}_R.$$

Then, we have the shifted QR step of  $A$  as

$$A_1 = RQ + I = \begin{pmatrix} 3 - \frac{1}{\varepsilon^2+1} & \varepsilon - \frac{\varepsilon}{\varepsilon^2+1} \\ \frac{\varepsilon^3}{\varepsilon^2+1} & \frac{1}{\varepsilon^2+1} \end{pmatrix}.$$

which appears to be better?

Now, comparing the non-shifted step and the shifted step, it appears that the shift is converging faster to the actual eigenvalues of  $A$  with one of the off diagonal entries on the order of  $O(\varepsilon^3)$ . This better convergence is in line with the potential for a shift to accelerate convergence of QR iteration. In this case,  $\mu = 1$  is pretty close to the perturbed eigenvalue of  $\lambda_2 = 1 + O(\varepsilon)$  and so we should expect the shifted QR step to be more accurate.

- (3) Implement QR iteration for a real symmetric tridiagonal matrix and demonstrate its performance on e.g.  $100 \times 100$  example.

*My code is given below.* Running my code on matrices under size  $20 \times 20$  was yielding a max norm error of up to  $10^{-14}$ . However, as the matrix got larger up to  $100 \times 100$ , I noticed the max norm error was approaching  $10^{-1}$  quite rapidly. I'm guessing the poor convergence is due to me using the black box QR algorithms in Julia instead of using the optimized Givens method in Golub and Van Loan for tridiagonal matrices. Still, the largest eigenvalues were still getting resolved to a couple of digits of accuracy.

*Note, some symbols are missing in my code snippet because  $\text{\LaTeX}$  does not support some unicode characters.*

---

```

1  #=
2  # QR iteration for tridiagonal matrices
3  #
4  # Author: Caleb Jacobs
5  # Date last modified: 19/02/2022
6  =#
7
8  using LinearAlgebra
9
10 function QRI(A, maxIts;  = 0)
11     for i = 1 : maxIts
12         T = qr(A - *I)
13         A = T.R * T.Q + *I
14     end
15
16     return A
17 end
18
19 function prob3(n)
20     d = rand(n)
21     dl = rand(n - 1)
22     A = Symmetric(Tridiagonal(dl, d, dl))
23     B = QRI(A, 200)
24     true = sort(eigvals(Matrix(A)), rev = true)
25     approx = sort(diag(B), rev = true)
26
27     display(true)
28     display(approx)
29
30     display(norm(true - approx, Inf))
31 end

```

---