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

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

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. With this in mind, let's compute the first few iterations of A exactly:

$$A = A_0 = L_0 L_0^T = \begin{pmatrix} \sqrt{a} & 0 \\ \frac{b}{\sqrt{a}} & \sqrt{c - \frac{b^2}{a}} \end{pmatrix} \begin{pmatrix} \sqrt{a} & \frac{b}{\sqrt{a}} \\ 0 & \sqrt{c - \frac{b^2}{a}} \end{pmatrix}$$

which implies

$$A_1 = L_0^T L_0 = \begin{pmatrix} a + \frac{b^2}{a} & \frac{b}{\sqrt{a}} \sqrt{c - \frac{b^2}{a}} \\ \frac{b}{\sqrt{a}} \sqrt{c - \frac{b^2}{a}} & c - \frac{b^2}{a} \end{pmatrix}.$$

From this, we can see that the diagonal entries of A_1 have grown while the off-diagonal entries have become smaller. You can continue doing this, say in Mathematica, and the trend continues showing the off-diagonal entries of our iterates are decaying. So, in the limit, the off-diagonal entries of our iterate will go to zero leaving only a diagonal matrix. But, from before, we know this diagonal matrix is similar to A and so it has the same eigenvalues. So, the final diagonal matrix has diagonal entries equal to λ_1 and λ_2.

(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

$$A = \underbrace{\begin{pmatrix} \frac{2}{\sqrt{\varepsilon^2+4}} & \frac{\varepsilon \text{sign}(\varepsilon^2-2)}{\sqrt{\varepsilon^2+4}} \\ \frac{\varepsilon}{\sqrt{\varepsilon^2+4}} & \frac{-2\text{sign}(\varepsilon^2-2)}{\sqrt{\varepsilon^2+4}} \end{pmatrix}}_{Q} \underbrace{\begin{pmatrix} \sqrt{\varepsilon^2+4} & \frac{3\varepsilon}{\sqrt{\varepsilon^2+4}} \\ 0 & \frac{|\varepsilon^2-2|}{\sqrt{\varepsilon^2+4}} \end{pmatrix}}_{R}.$$

Then, we have the first QR step of A as

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

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

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

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

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. 100×100 example.

My code is given below. Running my code on matrices under size 20×20 was yielding a max norm error of up to 10^{-14}. However, as the matrix got larger up to 100×100, 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 LaTeXdoes not support some unicode characters.

```
#=
# QR iteration for tridiagonal matrices
#
# Author: Caleb Jacobs
# Date last modified: 19/02/2022
=#

using LinearAlgebra

function QRI(A, maxIts;  = 0)
    for i  1 : maxIts
        T = qr(A -  *I)
        A = T.R * T.Q +  *I
    end

    return A
end

function prob3(n)
    d  = rand(n)
    dl = rand(n - 1)
    A  = Symmetric(Tridiagonal(dl, d, dl))
    B = QRI(A, 200)
     true   = sort(eigvals(Matrix(A)), rev = true)
     approx = sort(diag(B), rev = true)

    display( true)
    display( approx)

    display(norm( true -  approx, Inf))
end
```
