Convergence of the orthogonal iteration algorithm

Note: previously run on Julia Version 1.2.0 (2019-08-20)

Q1. Write code to create a matrix in $\mathbb{R}^{n*n}$ of size = 8 with eigenvalues 1, 0.2, 0.05, 0.017, 0.0085, 0.0042, 0.0021, 0.0011

In [ ]:
```
using Base
using LinearAlgebra
import Pkg; Pkg.add("Plots")
using Plots
using Printf
using Random
```

In [ ]:
```
D = zeros(8,8)
D[1,1] = 1
D[2,2] = 0.2
D[3,3] = 0.05
D[4,4] = 0.017
D[5,5] = 0.0085
D[6,6] = 0.0042
D[7,7] = 0.0021
D[8,8] = 0.0011

rng = MersenneTwister(2019)

P = rand(rng,8,8)
A = P*D*inv(P) # by diagnonalization theorem
println("Matrix A is")
A
```

In [ ]:
```
println("Eigenvalues of matrix A are")
eigvals(A)
```

Q2. Implement the orthogonal iteration algorithm. Print the values along the diagnoal of R_k at each iteartion k for $k = 1, \ldots, 5$. Print each number using at most 4 significant digits.

Q4. Assume that $\|A_k(p:n, 1:p-1)\|_2$ and $\|A_k(p+1:n, 1:p)\|_2$ are very small. Show that the entry $A_k(p,p)$ is very close to an eigenvalues of A.

Q5. Considering entry p along the diagonal, plot the convergence of the pth eigenvalue. Choose $p = 1, 2, and 3$, Use a semi-logarithmic plot. We would expect the following theoretical rate of convergence at step k for entry p:

$$\begin{array}{ll} \max(|\lambda_{p+1}/\lambda_p|^k, |\lambda_p/\lambda_{p-1}|^k) & 1 < p < n \\ |\lambda_2/\lambda_1|^k & p = 1 \\ |\lambda_n/\lambda_{n-1}|^k & p = n \end{array}$$

In [ ]:
```
# Q2 Setup

Qk = rand(rng,8,8)
Ak = zeros(8,8)
block = zeros(4,4)
y_2norm = []
y_p1 = []
y_p2 = []
y_p3 = []
```

In [ ]:
```
for k = 1:5
    println("===Itearation: ", k, "===")
    global Qk
    Rk = A*Qk
    Q,R = qr(Qk)
    println("## Q2 ## ")
    println("R[1,1] is ", round(R[1,1], sigdigits=4))
    println("R[2,2] is ", round(R[2,2], sigdigits=4))
    println("R[3,3] is ", round(R[3,3], sigdigits=4))
    println("R[4,4] is ", round(R[4,4], sigdigits=4))
    println("R[5,5] is ", round(R[5,5], sigdigits=4))
    println("R[6,6] is ", round(R[6,6], sigdigits=4))
    println("R[7,7] is ", round(R[7,7], sigdigits=4))
    println("R[8,8] is ", round(R[8,8], sigdigits=4))
    Qk = Q

    Ak = transpose(Qk)*A*Qk

    # Q3 Consider block [5:8, 1:4]
    block = Ak[5:8, 1:4]
    append!(y_2norm, opnorm(block,2))
```

```
    # Q4 Observe diagonal of Ak
    println("## Q4 ## ")

    # Note: the preblock here means A_k[p:n, 1:p-1]
    # postblock here means A_k[p+1:n, 1:p]
    # On boundary, when p=1, we only have block A_k[2:8, 1]
    # when p=8, we only have block A_k[8,1:7]
    for p = 1:8
        if p == 1
            postblock = Ak[2:8, 1]
            m = @sprintf " When p=1, block 2-norm is %1.4f, Ak diag va
l is %1.4f"\
            norm(postblock,2) Ak[1,1]
            println(m)
        elseif p == 8
            postblock = Ak[8, 1:7]
            m = @sprintf " When p=8, block 2-norm is %1.4f, Ak diag va
l is %1.4f"\
            norm(postblock,2) Ak[8,8]
        else
            preblock = Ak[p:8, 1:(p-1)]
            postblock = Ak[(p+1):8, 1:p]
            m = @sprintf \
            " When p = %1.0f, preblock norm is %1.4f, postblock norm i
s %1.4f, Ak diag is %1.4f"\
            p norm(preblock,2) norm(postblock,2) Ak[p,p]
            println(m)
        end
    end

    # Q5 Convergence of pth eval: p=1,2,3
    for p = 1:3
        if p == 1
            append!(y_p1, abs(1-Ak[1,1]))
        elseif p == 2
            append!(y_p2, abs(0.2-Ak[2,2]))
        else
            append!(y_p3, abs(0.05-Ak[3,3]))
        end
    end
end
```

Q3. Consider the block $(p+1:n, 1:p)$ in the matrix $A_k = Q_k^T A Q_k$, plot the 2-norm of this block as a function of k for $p = 4$. Compare with the analytical estiate, which states that it should decay like $|\frac{\lambda_{p+1}}{\lambda_p}|^k$

In [ ]:
```
plot(y_2norm; yscale =: log)
```

```
In [ ]: # Q5 plotting
        plot(y_p1; yscale=: log)
        plot(y_p2; yscale=: log)
        plot(y_p3; yscale=: log)
```

Observation: At early print of Q4 result, we would see quite large block 2-norms, and the diag value is not quite accurate; at late stages, the norms become much smaller with Ak diag being closer to eigenvalues.

Other written analyses:

Q1 Let $A \in \mathbb{R}^{8\times 8}$, $A = PDP^{-1}$

where $P \in \mathbb{R}^{8\times 8}$, a random matrix

$D \in \mathbb{R}^{8\times 8}$, a diagonal matrix w/ eval on diagonal

I generated P with MersemeTwister Random generator

Then I put specific eigenvalues on the diagonal of D.

By Diagonalization Theorem, the diagonal entries of D are eval of A that respectively corresponds to the LID eigenvectors in P.

I have printed out the approximated eigenvalues of A to the console output for reference.

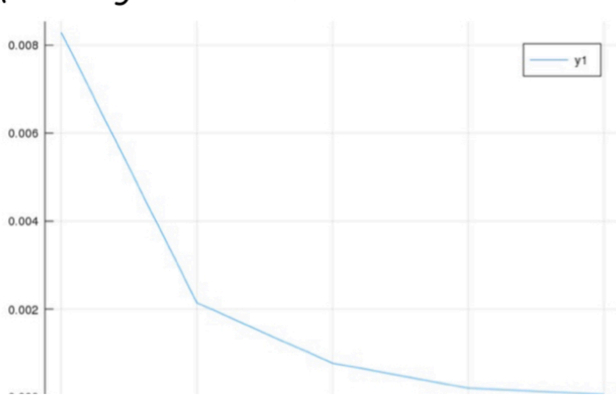
Q2 Marked as ## PART2 ## in output

Q3 Consider block $A_K[5:8, 1:4]$

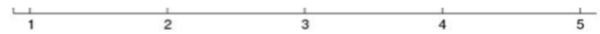
where $A_K = Q_K^T A Q_K$

I store the 2-norm of this block at each iteration to a list called y_2norm

plot(y_2norm)



plot(y_2norm; yscale=:log)



Analytically, the decay at each iteration K should be

$$\left|\frac{\lambda 5}{\lambda 4}\right|^K = \left|\frac{0.017}{0.0085}\right|^K = |2|^K$$

$$\log 2^K = K \cdot \log 2 \longrightarrow$$ linear as shown on the right

Generally this agrees with practical result.

Q4

Marked as ## PART 4 ## in console output.

Our assumption here is that

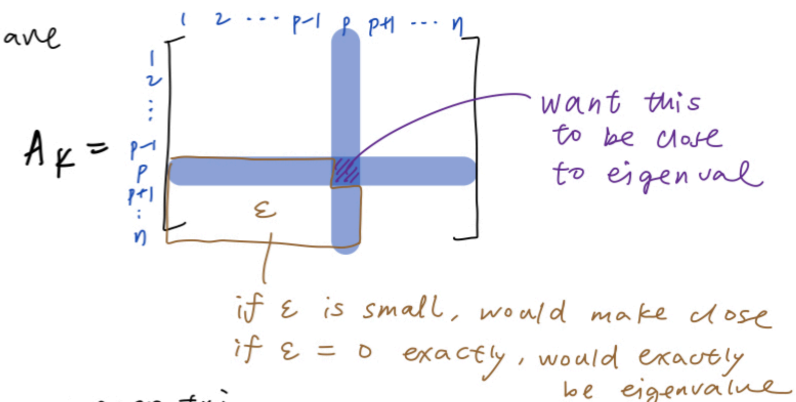
$\underbrace{\| A_K[p:8, 1:p-1] \|_2}_{\text{called preblock in my code}}$ AND $\underbrace{\| A_K[p+1:8, 1:p] \|_2}_{\text{called post block in my code}}$ are both small

so I've printed out the norms of blocks for every p at every iteration.

At early iterations, we can see that the norms of blocks are relatively big and that $A_K[p,p]$ for each p are not that close to eigenvalues of A.

But look at iteration 5, we can see that all norms are almost very small and that the diagonal values become closer to the eigenvalues of A.

Analytically, we have

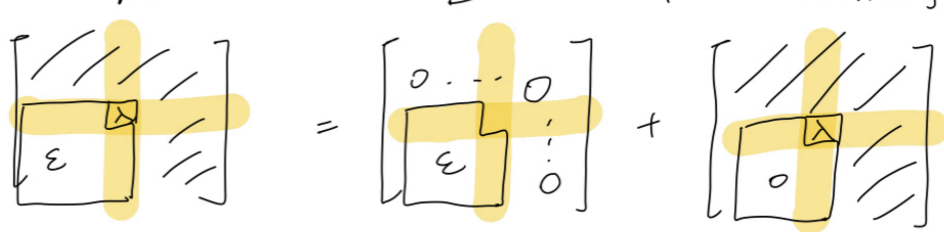


We know if T be an upper tri,

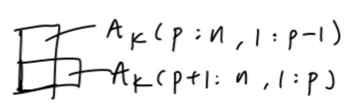
$$T = \begin{bmatrix} \alpha & * & * \\ 0 & \beta & * \\ 0 & 0 & \gamma \end{bmatrix} \quad \text{then } \lambda(T) = \lambda(\alpha) \cup \lambda(\beta) \cup \lambda(\gamma)$$

From this, we decompose A_K into two parts s.t.

$$A_K \quad = \quad E \quad + \quad A_{\text{error-free}}$$



$\| E \|_2 = \varepsilon$ ← is small because the norm of



are both small.

And we know that λ is an eigenval of $A_{error\text{-}free}$ from the above theorem. i.e. $\lambda \in \lambda(A_{error\text{-}free})$

$$\Rightarrow P_{A_{error\text{-}free}}(\lambda) = \det(A_{error\text{-}free} - \lambda I) = 0$$

since its a continuous polynomial and the perturbation by E is small,

$$\det(\underbrace{A_K}_{A_{error\text{-}free} + E} - \lambda I) \approx 0$$

#

In [ ]: