Idea, Use the spectral mapping theorem to find other eigenvalues other than just the one with largest magnitude...

x_0 ~ choose randomly

$$y_{n+1} = A x_n$$

$$x_{n+1} = \frac{y_{n+1}}{\|y_{n+1}\|}$$

Define a new matrix $B = A - \alpha I$

How do the eigenvalues of B compare with eigenvalues of A?

Let ξ be an eigenvector of A with eigenvalue λ.

Then

$$A\xi = \lambda \xi.$$

What about B?

$$B\xi = (A - \alpha I)\xi = A\xi - \alpha\xi$$

$$= \lambda\xi - \alpha\xi = (\lambda - \alpha)\xi.$$

Therefore ξ is also an eigenvector of B, but with shifted eigenvalue given by $\lambda - \alpha$.

Suppose the eigenvalues of A are given by

$$\lambda_1 = 3, \quad \lambda_2 = 4, \quad \lambda_3 = 6$$

eigenvectors $\quad \xi_1 \quad \xi_2 \quad \xi_3$

Power iteration. $\quad y_{n+1} = A x_n$

$$x_{n+1} = \frac{y_{n+1}}{\|y_{n+1}\|}$$

Then x_n "converges" to the eigenvector ξ_3

and

$$\frac{x_n \cdot A x_n}{x_n \cdot x_n} \to \lambda_3 \quad \text{as } n \to \infty$$

```
julia> using LinearAlgebra

julia> S=rand(3,3) .- 0.5
3×3 Matrix{Float64}:
  0.0261003    0.480798    0.407647
 -0.0478098   -0.448344    0.252868
  0.137901    -0.390876   -0.233318

julia> D=diagm([3,4,6])
3×3 Matrix{Int64}:
 3  0  0
 0  4  0
 0  0  6

julia> A=S*D*inv(S)
3×3 Matrix{Float64}:
  5.21801    1.28406    0.025378
  1.01841    4.73558    0.408967
 -1.32488   -0.589416   3.04641

julia> eigvals(A)
3-element Vector{Float64}:
 3.000000000000001
 3.9999999999999987
```

similarity transform

Matrix with eigenvalues on the diagonal

Matrix with same eigenvalues

Use built-in function to check eigenvalues...

```
julia> x0=rand(3)
3-element Vector{Float64}:
 0.21670536054982703
 0.5060992463089435
 0.8797618172562653

julia> x=x0
       for j=1:100
           global x,y
           y=A*x
           x=y/norm(y)
       end

julia> x
3-element Vector{Float64}:
  0.7641894106318069
  0.47403405928409253
 -0.4373857054326958

julia> x'*A*x/(x'*x)
5.999999999999999
```

Set a random starting vector

do the power method

close to an eigenvector

use least squares (Rayleigh-Quotient) to approximate the corresponding eigenvalue

as expected, this number is close to 6, since that's the eigenvalue of largest magnitude.

To find another eigenvector and eigenvalue we use a simple form of the spectral mapping theorem to shift the eigenvalues of A so a different one has the largest magnitude...

Since we just found the eigenvalue $\lambda_3 = 6$, one can map that eigenvalue to zero so it has the smallest magnitude. The a different eigenvector will be found.

shift by 6

$B = A - 6I$

eigenvalues of B are now the shifts of the eigenvalues of A

$-3, -2, 0$

eigenvalue of largest magnitude

found this one

which means $-3+6=3$ is the eigenvalue of the original matrix A.

Shifted power method

```
julia> B=A-6*I
3×3 Matrix{Float64}:
 -0.781987   1.28406    0.025378
  1.01841   -1.26442    0.408967
 -1.32488   -0.589416  -2.95359

julia> x=x0
       for j=1:100
           global x,y
           y=B*x
           x=y/norm(y)
       end

julia> x
3-element Vector{Float64}:
  0.1760333940369508
 -0.32245321022423423
  0.9300732075486947

julia> beta=x'*B*x/(x'*x)
-3.0
```

The shifted matrix

use $\boxed{\uparrow}$ untill you get to this code block. Then press $\boxed{\leftarrow}$ to enter the editor and change the A to a B.

change this line

The approximate eigenvector

approximation of the eigenvalue of B. Shift it back to get the eigenvalue of A

$-3 + 6 = 3$

shift

eigenvalue of B

eigenvalue of A

Shifted inverse power method...

Define $B = \boxed{(A-\alpha I)^{-1}}$

How are the eigenvalues of B related to A?

That didn't work, so try again from the other way..

~~Let ξ be an eigenvector of A with value λ. Thus $A\xi = \lambda\xi$.~~

~~$B\xi = (A-\alpha I)^{-1}\xi = ?$ ...~~

Let ξ be an eigenvector of B with value β. Thus $B\xi = \beta\xi$

$$B\xi = (A-\alpha I)^{-1}\xi = \beta\xi$$

Since $B^{-1} = A - \alpha I$

want to reorganize this to identify an eigenvalue of A

Thus $$\xi = (A-\alpha I)\beta\xi$$

$$\xi = \beta A\xi - \alpha\beta\xi$$

$$(1+\alpha\beta)\xi = \beta A\xi$$

That's the eigenvalue.

$$A\xi = \left(\frac{1+\alpha\beta}{\beta}\right)\xi$$

Thus ξ is also an eigenvector of A with eigenvalue $\frac{1+\alpha\beta}{\beta} = \boxed{\frac{1}{\beta} + \alpha = \lambda}$

eigenvalue of B   shift   eigenvalue of A

Solve for β in terms of λ

$$\frac{1}{\beta} = \lambda - \alpha \;, \qquad \boxed{\beta = \frac{1}{\lambda - \alpha}}$$

inverse

shift

eigenvalue of B

eigenvalue of A

Thus

matrix transform

$$B = (A - \alpha I)^{-1}$$

same function

and

eigenvalue mapping

$$\beta = (\lambda - \alpha)^{-1}$$

Spectral mapping for the function $f(z) = (z-\alpha)^{-1}$

shift is α. Sorry, I could have chosen better letters. The book use σ instead for the shift.

where β is the eigenvalues of B and λ is the eigenvalues of A.

Consider these eigenvalues again...

$$\lambda_1 = 3, \quad \lambda_2 = 4, \quad \lambda_3 = 6$$

Have to guess a shift... in this case we know the answer (but the computer doesn't) so a shift is easy to guess...

Suppose you guessed $\alpha = 4.5$

$$\frac{1}{3-4.5}, \quad \frac{1}{4-4.5}, \quad \frac{1}{6-4.5}$$

$$= -\frac{2}{3}, \quad = -2, \quad = \frac{2}{3}$$

now the second eigenvalue has the largest magnitude...

```
julia> B=inv(A-4.5*I)
3×3 Matrix{Float64}:
 -0.0901183   1.64581    0.461474
  0.834237   -0.897845  -0.238043
 -0.256136   -1.136     -1.01204

julia> x=x0
       for j=1:100
           global x,y
           y=B*x
           x=y/norm(y)
       end

julia> x
3-element Vector{Float64}:
 -0.6286347815166934
  0.5862018221507268
  0.5110633377328362

julia> beta=x'*B*x/(x'*x)
-1.9999999999999987

julia> 1/beta+4.5
3.9999999999999996
```

inverse shifted matrix

power method for B

approximation of the eigenvector

The eigenvalue of B is approximately -2 as expected...

map the eigenvalue of B back to an eigenvalue of A (spectral mapping) using the relation

$$\boxed{\underbrace{\frac{1}{\beta}}_{\text{eigenvalue of B}} + \underbrace{\alpha}_{\text{shift}} = \underbrace{\lambda}_{\text{eigen value of A}}}$$

NOTE!

This method required one to guess that $\alpha = 4.5$ was close to the eigenvalue being searched for... as the iteration converges, one obtains a better approximations of that eigenvalue which can be used to speed convergence.

Suppose you guessed $\alpha = 4.5$ ← update this shift to speed later iterations

plug in initial guess

```
julia> B=inv(A-4.5*I)
3×3 Matrix{Float64}:
 -0.0901183   1.64581    0.461474
  0.834237   -0.897845  -0.238043
 -0.256136   -1.136     -1.01204

julia> y1=B*x0
       x1=y1/norm(y1)
3-element Vector{Float64}:
  0.607196147850177
 -0.24052627195689993
 -0.7572713849964365

julia> lambda=x1'*A*x1/(x1'*x1)
4.173164797050856
```

← one iteration

new guess for eigenvalue

$$\frac{1}{3-4.5}, \quad \frac{1}{4-4.5}, \quad \frac{1}{6-4.5}$$

$$-\frac{2}{3}, \quad -2, \quad \frac{2}{3}$$

$-.852\ldots, \quad -5.77\ldots, \quad 0.547\ldots$

after shifting by the new guess the eigenvalues of B are

```
julia> 1/(3-lambda)
-0.8523951643569905

julia> 1/(4-lambda)
-5.7748457944735305

julia> 1/(6-lambda)
0.5473947504326905
```

Since the rate of convergence depends on the ratio of the largest to the second largest eigenvalue (in magnitude) then the new shift will converge faster.

Note the exact values here will be different depending on the random value of x_0 and the random similarity S used to make A in the first place

In this case I ran it again so the values are not the same as in class use the new shift

```
julia> B=inv(A-4.173164797050856*I)
3×3 Matrix{Float64}:
 -1.05801    3.85818   1.37653
  1.63198   -3.08164  -1.08175
  0.390336  -2.92454  -1.9402

julia> y2=B*x1
       x2=y2/norm(y2)
3-element Vector{Float64}:
 -0.5971833916006509
  0.5831282051682319
  0.550757199801891

julia> lambda=x2'*A*x2/(x2'*x2)
3.9629049493121054
```

another iteration

a better guess for the eigenvalue...

Continue in this way results in accelerated convergence similar (but ~~not~~ as fast) as Newton's method.

use best shift so far

```
julia> B=inv(A-3.9629049493121054*I)
3×3 Matrix{Float64}:
  6.41948  -15.9679  -6.94759
 -5.38097   15.3468   6.69918
 -5.81931   13.2132   4.64386

julia> y3=B*x2
       x3=y3/norm(y3)
3-element Vector{Float64}:
 -0.6289807133628148
  0.5875022772401237
  0.5091407825496536

julia> lambda=x3'*A*x3/(x3'*x3)
3.999150022849284
```

another iteration

approximation of the eigenvalue is already quite good...

Our book calls this shifted-inverse iteration by a different name

from page 119

function RAYLEIGH-QUOTIENT-ITERATION(A, σ)
  $\vec{v} \leftarrow$ ARBITRARY(n)
  for $k \leftarrow 1, 2, 3, \ldots$
    $\vec{w} \leftarrow (A - \sigma I_{n\times n})^{-1}\vec{v}$
    $\vec{v} \leftarrow \vec{w}/\|\vec{w}\|$
    $\sigma \leftarrow \frac{\vec{v}^\top A\vec{v}}{\|\vec{v}\|_2^2}$
  return $\vec{v}$

update the shift

that's the matrix B

this is the least-squares aproximation of the eigenvalue . . .