Matrix A:

```
julia> A
5×5 Matrix{Float64}:
  0.439681   -0.277692    0.219266   -0.38998    -0.0577448
 -0.277692    0.547298   -0.0663005   0.30461    -0.237325
  0.219266   -0.0663005   0.442398   -0.0730695  -0.321961
 -0.38998     0.30461    -0.0730695   0.411968   -0.115089
 -0.0577448  -0.237325   -0.321961   -0.115089    0.48856
```

```
julia> b
5-element Vector{Float64}:
 -0.3956793738033185
 -0.2313558041293019
  0.026968362632653897
 -0.14249304757707748
 -0.28685391868931265
```

```
julia> wopt
1.988310570716017
```

```
julia> eigvals(T)
5-element Vector{Float64}:
 -0.0012919021297408961
  0.0
  0.40718492985589066
  0.7968609159352111
  0.995412825965209
```

```
julia> eigvals(Tw(wopt))
5-element Vector{Float64}:
 -0.9908792733769111
 -0.9883105707160169
 -0.17870047044728987
  0.5960964118284106
  0.9908792733769123
```

smaller!

```
julia> xk=zeros(5);for k=1:20
           global xk=wopt*(M\(N*xk+b))+(1-wopt)*xk
           println("|Ax-b|=",norm(A*xk-b))
       end
|Ax-b|=1.0875078654369321
|Ax-b|=0.6280101277289264
|Ax-b|=1.0236888439541603
|Ax-b|=0.673830522202391
|Ax-b|=0.9951880015009776
|Ax-b|=0.6886891926223562
|Ax-b|=0.9667854663151176
|Ax-b|=0.6924364733380893
|Ax-b|=0.9382682726945771
|Ax-b|=0.6921049941919438
|Ax-b|=0.9102488393199789
|Ax-b|=0.690307758032778
|Ax-b|=0.8829797341724379
|Ax-b|=0.6879866162211293
|Ax-b|=0.8565368855400082
|Ax-b|=0.6854722009670148
|Ax-b|=0.8309309074947885
|Ax-b|=0.682875896373431
|Ax-b|=0.806150137273439
|Ax-b|=0.6802314737986519
```

need extra parenthesis here!

conv. faster

```
julia> xk=zeros(5);for k=1:20
           global xk=M\(N*xk+b)
           println("|Ax-b|=",norm(A*xk-b))
       end
|Ax-b|=0.6836678468559619
|Ax-b|=0.711082395079733
|Ax-b|=0.7271945228278556
|Ax-b|=0.7362825479524141
|Ax-b|=0.7415807469283898
|Ax-b|=0.7447822785792533
|Ax-b|=0.7466567243619662
|Ax-b|=0.7475654547167941
|Ax-b|=0.7477063600533841
|Ax-b|=0.7472129232280572
|Ax-b|=0.7461905403193616
|Ax-b|=0.7447286534432284
|Ax-b|=0.7429045133880892
|Ax-b|=0.7407845015650331
|Ax-b|=0.7384249930085612
|Ax-b|=0.7358732852634114
|Ax-b|=0.7331686401089766
|Ax-b|=0.7303433654762106
|Ax-b|=0.7274238618898733
|Ax-b|=0.7244315832166409
```

Eigenvalues and eigenvectors...

(a) Calculate $w^j = By^{(j-1)}$.

(b) Find p such that $|w_p^{(j)}| = \max\{\, |w_i^{(j)}| : i = 1, \ldots, n \,\}$.

(c) Evaluate the approximation $\lambda_1^{(j)} \approx \dfrac{w^{(j)} \cdot y^{(j-1)}}{y^{(j-1)} \cdot y^{(j-1)}}$.

(d) Calculate $y^{(j)} = w^{(j)}/w_p^{(j)}$.

---

recall $B = A^TA$ So means B was symmetric and if A was non-singular than B is pos. definite

$Bx \cdot x > 0$ for all $x \in \mathbb{R}$

if A is singular then B is pos. semi-definite

$Bx \cdot x \geq 0$ for all $x \in \mathbb{R}$.

Theorem about the eigenvalues.

Spectral Theorem for Symmetric Matrices: If $B \in \mathbb{R}^{n \times n}$ and $B^T = B$ then the eigenvalues of B are real and there is an orthonormal basis of eigenvectors. Thus,

$BV_i = \lambda_i V_i$ for $i = 1, \ldots, n$. and $V_i \cdot V_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$

Let x be any (random vector)

$x = c_1V_1 + c_2V_2 + \cdots + c_nV_n$ since V_i's are basis.

$Bx = c_1BV_1 + c_2BV_2 + \cdots + c_nBV_n$

$= c_1\lambda_1V_1 + c_2\lambda_2V_2 + \cdots + c_n\lambda_nV_n$

λ_1 ↖ largest eigenvalue..

$\lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n \geq 0$

assume $\lambda_1 > 0$ otherwise $B = 0$.

$$B^k x = c_1 \lambda_1^k v_1 + c_2 \lambda_2^k v_2 + \cdots + c_n \lambda_n^k v_n$$

Normalize it

$$\frac{B^k x}{\|B^k x\|} = \frac{c_1 \lambda_1^k v_1 + c_2 \lambda_2^k v_2 + \cdots + c_n \lambda_n^k v_n}{\left\| c_1 \lambda_1^k v_1 + c_2 \lambda_2^k v_2 + \cdots + c_n \lambda_n^k v_n \right\|}$$

assume $\lambda_1 > 0$

$$= \frac{\lambda_1^k \left(c_1 v_1 + c_2 \frac{\lambda_2^k}{\lambda_1^k} v_2 + \cdots + c_n \frac{\lambda_n^k}{\lambda_1^k} v_n \right)}{\lambda_1^k \left\| c_1 v_1 + c_2 \frac{\lambda_2^k}{\lambda_1^k} v_2 + \cdots + c_n \frac{\lambda_n^k}{\lambda_1^k} v_n \right\|}$$

if $\lambda_1 < 0$ then $|\lambda_1|^k$

$$= \frac{\left(c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1}\right)^k v_n \right)}{\left\| c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1}\right)^k v_n \right\|}$$

assume $\lambda_1 > \lambda_2$

then these terms all go to zero

$$\longrightarrow \frac{c_1 v_1}{\|c_1 v_1\|} = \frac{v_1}{\|v_1\|} \quad \text{as } k \to \infty.$$

↑ eigenvector corresponding to largest eigenvalue...

How to find other eigenvalues?

One idea ... iterate a different matrix ...

How are eigenvalues of B^{-1} related to B?

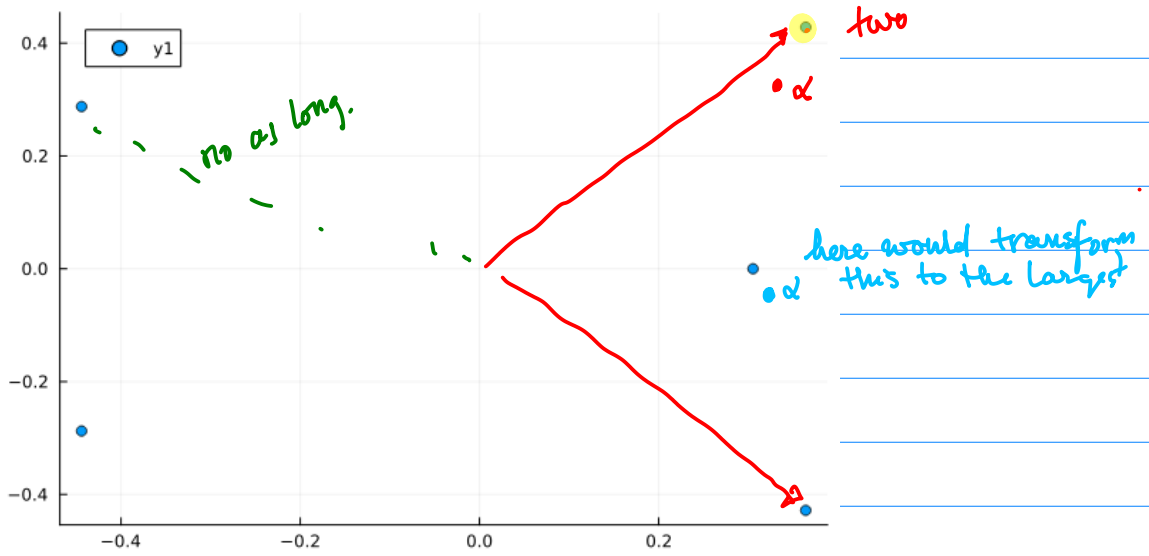
$$Bv = \lambda v \quad \text{so} \quad v = B^{-1} \lambda v \quad \text{so} \quad B^{-1} v = \frac{1}{\lambda} v$$

Now I can find the smallest eigenvalue of B by iterating B^{-1} and take $\frac{1}{\lambda}$ as the eigenvalue corresponding to the vector found.

$$(B+\alpha I)v = Bv+\alpha v = \lambda v+\alpha v = (\lambda+\alpha)v$$

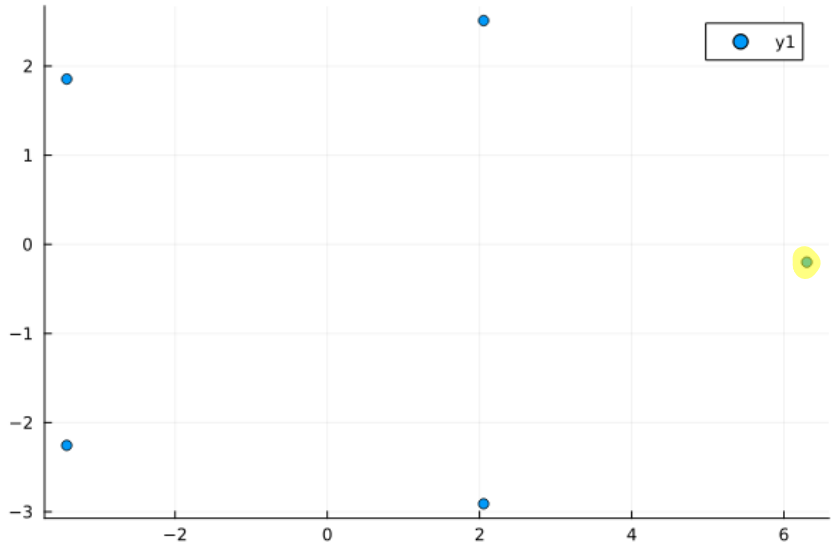
eigenvalues are shifted.

$$(B+\alpha I)^{-1}v = \frac{1}{\lambda+\alpha}v$$




```
julia> B=rand(5,5).-0.5
5×5 Matrix{Float64}:
 -0.177942      0.408618   0.212706   0.172212  -0.351553
 -0.000731852   0.275066  -0.268678  -0.071856   0.381457
 -0.434643     -0.258314  -0.425183   0.168473  -0.280436
  0.0376379    -0.37819   -0.250814   0.38254   -0.337529
 -0.29649      -0.471764  -0.286461   0.248271   0.0909604
```

```
julia> eigvals(B)
5-element Vector{ComplexF64}:
 -0.4440099425241164 - 0.28754913884357347im
 -0.4440099425241164 + 0.28754913884357347im
 0.30528997088134174 + 0.0im
 0.36408581327212297 - 0.42824928127425 35im
 0.36408581327212297 + 0.42824928127425 35im
```



now the largest eigenvalue is unique.

```
julia> lambdas=eigvals(B)
5-element Vector{ComplexF64}:
 -0.4440099425241164 - 0.28754913884357347im
 -0.4440099425241164 + 0.28754913884357347im
 0.30528997088134174 + 0.0im
 0.36408581327212297 - 0.42824928127425 35im
 0.36408581327212297 + 0.42824928127425 35im
```

```
julia> newlambdas=2.0./lambdas.-(0.25+.2im)
5-element Vector{ComplexF64}:
 -3.4234338392048627 + 1.8551750766050696im
 -3.4234338392048627 - 2.2551750766050698im
   6.301148713553213 - 0.2im
   2.054662029872525 + 2.5108165764626906im
   2.054662029872525 - 2.9108165764626 91im
```

```
julia> scatter([real.(newlambdas)],[imag.(newlambdas)])
```