Finding eigenvalues and eigenvectors

$$
\underset{A \neq A^*}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}}
\xrightarrow{\text{Phase 1}}
\underset{H}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}}
\xrightarrow{\text{Phase 2}}
\underset{T}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}.
$$

uses Householder reflectors and can be done in a fixed number of steps

uses iterative procedure to obtain approximations...

Chapter 26 does the reflections...

Chapter 28 works on iterative approximations..

Chapter 27 is independent and reviews classical algorithms for approximating eigenvalues and vectors...

```
julia> function powermethod(v0)
           vn=v0
           for n=1:10
               wn=A*vn
               vn=wn/norm(wn)
               print("lambda=",vn'*A*vn)
           end
       end
powermethod (generic function with 1 method)
```

```
julia> A=rand(4,4)
4×4 Matrix{Float64}:
 0.98748   0.651018  0.885124  0.0261628
 0.951147  0.378067  0.416188  0.57458
 0.789824  0.806327  0.950815  0.505372
 0.368743  0.895383  0.366291  0.807251
```

```
julia> using LinearAlgebra

julia> eigvals(A)
4-element Vector{Float64}:
 -0.3733801108176234
  0.26729532855772986
  0.6214566361971311
  2.6082420172712113
```

There are 4 different eigenvalues, each has an eigenvector so there is a basis of eigenvectors in this case...

```
julia> v0=rand(4)
4-element Vector{Float64}:
 0.08271531770108864
 0.1793736093726439
 0.354571695809508
 0.6492078951605074
```

converges to the eigenvalue of largest magnitude...

```
julia> powermethod(v0)
lambda=2.4736194933436164
lambda=2.5822336221464597
lambda=2.6033385488053864
lambda=2.607045794225087 8
lambda=2.607968009461653 3
lambda=2.608175390272833 3
lambda=2.60822632581003 2
lambda=2.608238249645876 5
lambda=2.608241123334567 5
lambda=2.60824180369362 3
```

- Why does it converge?
- Can we make it converge to a different eigenvalue? Yes, inverse power method...

Explanation: Assumptions

① That there is a basis of eigenvectors
② That the eigenvalue of largest magnitude is unique...

Could assume $A^T = A$ in which case the spectral theorem implies there is a basis of eigenvectors and they are all real so one doesn't have to worry about complex conjugate pairs... Still might have to worry about two eigenvalues with the same magnitude...

Let $A \in \mathbb{R}^{n \times n}$ and

① $A\xi_i = \lambda_i \xi_i$ for $i = 1, \dots, n$

where all the ξ_i are independent and

② $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$

eigenvalue of largest magnitude

Let $v_0 \in \mathbb{R}^n$ be a randomly chosen vector. Since the ξ_i's are a basis, we can write v_0 with respect to that basis...

$$v_0 = c_1 \xi_1 + c_2 \xi_2 + \dots + c_n \xi_n$$

for some coefficients $c_i \in \mathbb{R}$

Unless one is very (un)-lucky all these coefficients will be non-zero... from a statistical point of view the probability is zero of one or any of the coefficients being zero.

```
wn=A*vn
vn=wn/norm(wn)
```

$$\begin{cases} w_1 = Av_0 \\ v_1 = \dfrac{w_1}{\|w_1\|} \end{cases}$$

$$w_1 = A v_0 = A(c_1 \xi_1 + c_2 \xi_2 + \dots + c_n \xi_n)$$

$$= c_1 A\xi_1 + c_2 A\xi_2 + \dots + c_n A\xi_n$$

$$= c_1 \lambda_1 \xi_1 + c_2 \lambda_2 \xi_2 + \dots + c_n \lambda_n \xi_n$$

$$v_1 = \frac{w_1}{\|w_1\|} = \frac{c_1 \lambda_1 \xi_1 + c_2 \lambda_2 \xi_2 + \dots + c_n \lambda_n \xi_n}{\|c_1 \lambda_1 \xi_1 + c_2 \lambda_2 \xi_2 + \dots + c_n \lambda_n \xi_n\|}$$

now do it again...

$$w_2 = Av_1 = A\,\frac{c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n}{\|c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n\|}$$

$$= \frac{c_1\lambda_1 A\xi_1 + c_2\lambda_2 A\xi_2 + \cdots + c_n\lambda_n A\xi_n}{\|c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n\|}$$

$$= \frac{c_1\lambda_1^2\xi_1 + c_2\lambda_2^2\xi_2 + \cdots + c_n\lambda_n^2\xi_n}{\|c_1\lambda_1\xi_1 + c_2\lambda_2\xi_2 + \cdots + c_n\lambda_n\xi_n\|}$$

$$v_2 = \frac{w_2}{\|w_2\|} = \frac{c_1\lambda_1^2\xi_1 + c_2\lambda_2^2\xi_2 + \cdots + c_n\lambda_n^2\xi_n}{\|c_1\lambda_1^2\xi_1 + c_2\lambda_2^2\xi_2 + \cdots + c_n\lambda_n^2\xi_n\|}$$

$\vdots$ in general

$$v_k = \frac{c_1\lambda_1^k\xi_1 + c_2\lambda_2^k\xi_2 + \cdots + c_n\lambda_n^k\xi_n}{\|c_1\lambda_1^k\xi_1 + c_2\lambda_2^k\xi_2 + \cdots + c_n\lambda_n^k\xi_n\|}$$

Want to take the limit as $k \to \infty$

$$\lim_{k\to\infty} v_k = \lim_{k\to\infty} \frac{c_1\lambda_1^k\xi_1 + c_2\lambda_2^k\xi_2 + \cdots + c_n\lambda_n^k\xi_n}{\|c_1\lambda_1^k\xi_1 + c_2\lambda_2^k\xi_2 + \cdots + c_n\lambda_n^k\xi_n\|}$$

$$= \lim_{k\to\infty} \frac{\lambda_1^k\left(c_1\xi_1 + c_2\overbrace{\left(\frac{\lambda_2}{\lambda_1}\right)^k}^{\to 0}\xi_2 + \cdots + c_n\overbrace{\left(\frac{\lambda_n}{\lambda_1}\right)^k}^{\to 0}\right)}{\left\|\lambda_1^k\left(c_1\xi_1 + c_2\underbrace{\left(\frac{\lambda_2}{\lambda_1}\right)^k}_{\to 0}\xi_2 + \cdots + c_n\underbrace{\left(\frac{\lambda_n}{\lambda_1}\right)^k}_{\to 0}\right)\right\|}$$

$$= \lim_{k\to\infty} \frac{\lambda_1^k}{\|\lambda_1\|^k}\,\frac{c_1\xi_1}{\underbrace{\|c_1\xi_1\|}_{\text{Just a constant}}} =$$

something like the eigenvector, except it might not converge.

Cancel if $\lambda_1 > 0$

may not converge..., unless $\lambda_1 > 0$

... But the approximations vn'*A*vn for the eigenvalue do converge ... Next time...