Lecture September 25

Lanczos algorithm and wrapping up Eigenvalue discussions

$A \in \mathbb{R}^{n \times n}$ $(\mathbb{C}^{n \times n})$, symmetric

$x \in \mathbb{R}^{n}$ $(\mathbb{C}^{n})$

$$Ax = \lambda x$$

Direct methods:

+ Jacobi
+ Householder
÷ QR transform
÷ SVD $\quad A = U \Sigma V^T$

$$UU^T = U^T U = \mathbb{1}$$
$$VV^T = V^T V = \mathbb{1}$$
$$A \in \mathbb{R}^{n \times m}$$
$$U \in \mathbb{R}^{n \times n}$$
$$\Sigma \in \mathbb{R}^{n \times m}$$
$$V \in \mathbb{R}^{m \times m}$$

$$A = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \xrightarrow{SAS'} T = \begin{bmatrix} x & x & & & 0 \\ x & x & x & & \\ & \ddots & \ddots & \ddots & \\ 0 & & x & x & x \\ & & & x & x \end{bmatrix}$$

Bisection

$$\longrightarrow \quad D = \mathrm{diag}(\lambda_1 \lambda_2 \dots \lambda_n)$$

$n = 10^5 \Rightarrow A \sim 8\,\text{bytes} \times 10^{10}$

$\sim 10^{11}$ bytes $\sim$ 100 Gb.

Iterative methods:

$$\underline{T = Q^T A Q} \qquad Q^T Q = \mathbb{1}$$

$$= \begin{bmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & & & \beta_{n-1} \\ 0 & & & \beta_{n-1} & \alpha_n \end{bmatrix}$$

$$Q = [q_1 \; q_2 \; \dots \; q_n]$$

column vector of Q

$$\langle q_i, q_j \rangle = q_i^T q_j = \delta_{ij}$$

$$Q Q^T A Q = Q T \Rightarrow$$

$$AQ = QT$$

$$A[q_1\ q_2\ q_k\ q_n]$$

$$Aq_k = \beta_{k-1}q_{k-1} + \alpha_k q_k + \beta_k q_{k+1}$$

$$\beta_0 \cdot q_0 = 0$$

$$q_k^T A q_k = \alpha_k q_k^T q_k = \alpha_k$$

Define

$$r_k = (A - \alpha_k I)q_k - \beta_{k-1}q_{k-1}$$

$$q_k^2 = 1 \rightarrow q_{k+1} = r_k/\beta_k$$

$$\beta_k = \pm \|r_k\|_2 = \sqrt{\sum_i r_i^2}$$

$r_k = 0$, iteration breaks down

Algorithm

Guesses : $\beta_0 = 1$ $q_0 = 0$ $k = 0$

r_0 = random values

while $(\beta_k \neq 0)$

$$q_{k+1} = r_k/\beta_k$$

$$k = k+1$$

$$\alpha_k = q_k^T A q_k \qquad (Aq_k)$$

$$r_k = (A - \alpha_k I)q_k - \beta_{k-1}q_{k-1}$$

$\beta_k = \|r_k\|_2$

end while

After ($A \in \mathbb{R}^{n \times n}$) n-iterations we have the full Tridiagonal matrix.

Lanczos' method converges to the extreme values

λ_{min} and λ_{max}

why is this of interest for project 2? 2d+2e

— approx one: $A \in \mathbb{R}^{n \times n}$

$$= \begin{bmatrix} x & x & & & \\ x & \ddots & \ddots & & \\ & \ddots & \ddots & x & \\ & & x & x & \\ & & & & \end{bmatrix}$$

λ_i depends on $-n-$

— approx two: $u(\infty) = 0$

replaced by

$u(\underline{1}) = 0$

with Lanczos' method and

incorrect ? !

project 1:

suppose $n = 1000$

use Lanczos' method

iteratively and stop at

$m = 100$

$$\begin{bmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_2 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{99} \\ 0 & & & \beta_{99} & \alpha_{100} \end{bmatrix}$$

and diagonalize (Lapack or Eigen/Armadillo)

look λ_{min} (3 lowest)

— project 2

$$\tau = \frac{a_{ll} - a_{kk}}{2a_{kl}}$$

Jacobi Rotate $a_{kl} \to 0$

$\tan\theta = t$

$t^2 + 2\tau t - 1 = 0$

$t = \boxed{-\tau \pm \sqrt{1+\tau^2}}$ ⟸

choose $t = -\tau + \sqrt{1+\tau^2}$

τ ~~becomes~~ large, $t \to 0$

$$t = \frac{\left(-\tau + \sqrt{1+\tau^2}\right)\left(\tau + \sqrt{1+\tau^2}\right)}{\tau + \sqrt{1+\tau^2}}$$

$$= \boxed{\frac{1}{\tau + \sqrt{1+\tau^2}}} \Longleftarrow$$

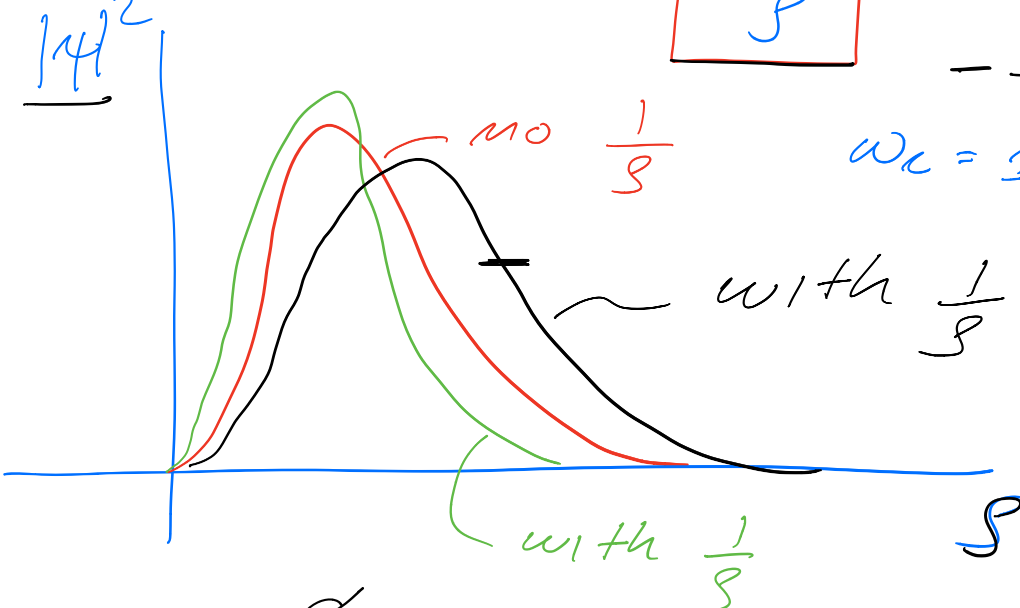
project 2e

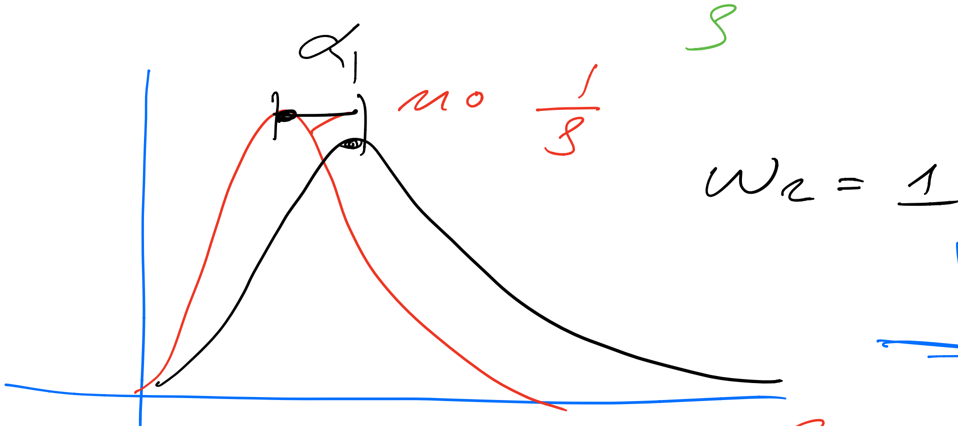
$$-\frac{d^2}{d\rho^2}\psi(\rho) + \omega_c^2\rho^2\psi(\rho)$$

$$\boxed{+\frac{1}{\rho}} = \lambda\psi(\rho)$$

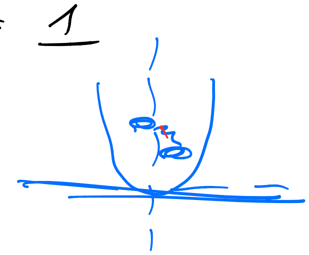
$$-\frac{1}{\rho}$$

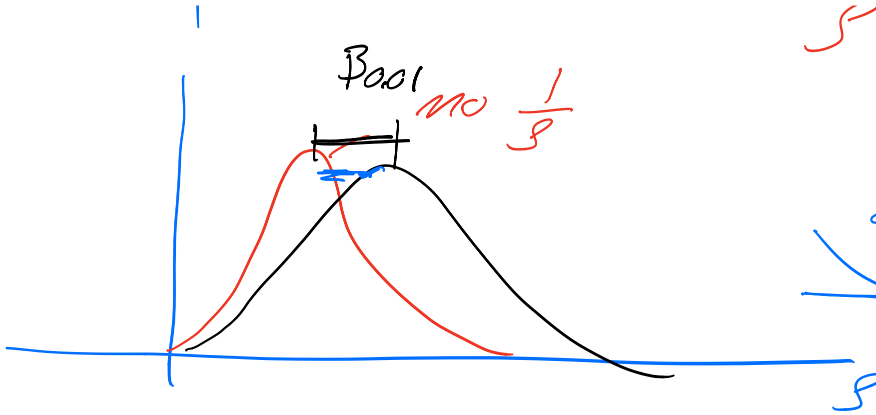
$\omega_c = 1$

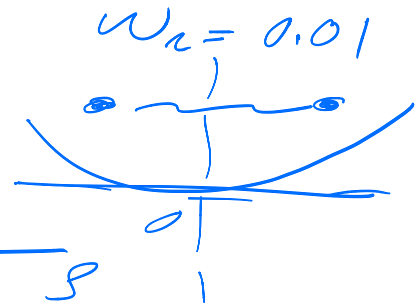




$\omega_c = 1$







$$\boxed{\begin{array}{l} \alpha_1 < \beta_{0.01} \quad ? \\ \boxed{\alpha_1 > \beta_{0.01}} \quad ? \end{array}}$$

This is what happens.

intuition

$\frac{1}{\rho}$ what is its range? infinite range.

Harmonic oscillator:

$E_0 = \hbar\omega \cdot 3/2$ $\qquad \omega \to 0$

$\qquad E_0 \to 0$

Term to the energy from the Coulomb interaction $1/\rho$ becomes smaller at a slower pace than the Harmonic oscillator, due the infinite range of $1/\rho$.

$$e^{-ge\cdot\mu}/g$$

Differential equations (ODE)

our most common type of equations

$$\frac{d^2 y(t)}{dt^2} = f\left(y, t, \frac{dy}{dt}\right)$$

Redefine in terms of two coupled first-order equations

$$v(t) = \frac{dy}{dt}$$

$$\frac{d^2 y}{dt^2} = \frac{d}{dt} v = f\left(y, t, \frac{dy}{dt}\right)$$

$$= f(y, t, v(t))$$

$$\frac{dy}{dt} = v(t)$$

$$\frac{dv}{dt} = f(y, t, v)$$

to defines initial conditions

$v(t_0) = v_0 \qquad y(t_0) = y_0$

Discretize: Define $t_{final} = t_n$

$t_{initial} = t_0$

$$h = \frac{t_n - t_0}{n} = \text{step size}$$

$$t_i = t_0 + i \cdot h$$

Taylor-expand (derivatives in p1 and p2)

Euler's forward formula for derivative

$$\left.\frac{dy}{dt}\right|_{t_i} = \frac{y(t_{i+1}) - y(t_i)}{h} + O(h)$$

$$= \frac{y_{i+1} - y_i}{h} + O(h)$$

$$\left.\frac{dy}{dt}\right|_{t_i} = v(t_i) = v_i \simeq \frac{y_{i+1} - y_i}{h}$$

$$y_{i+1} = y_i + h \cdot v_i$$

$$\left.\frac{dv}{dt}\right|_{t_i} \simeq \frac{v_{i+1} - v_i}{h}$$

$\Rightarrow$

Define y_0 and v_0 (initialization)

for $i = 1 : n$

$$y_{i+1} = y_i + h \cdot v_i$$

$$v_{i+1} = v_i + h \cdot f(y_i, t_i, v_i)$$

$$= v_i + h \cdot f_i$$

end for.

Euler's forward method.
Explicit (all previous values are known) scheme.

Euler's Backward method.
Implicit scheme. (later).