Lecture 4 1

Interpolation

Data: $(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)$

Model: $\hat{y} = \hat{y}(x) = z(x)\, a$

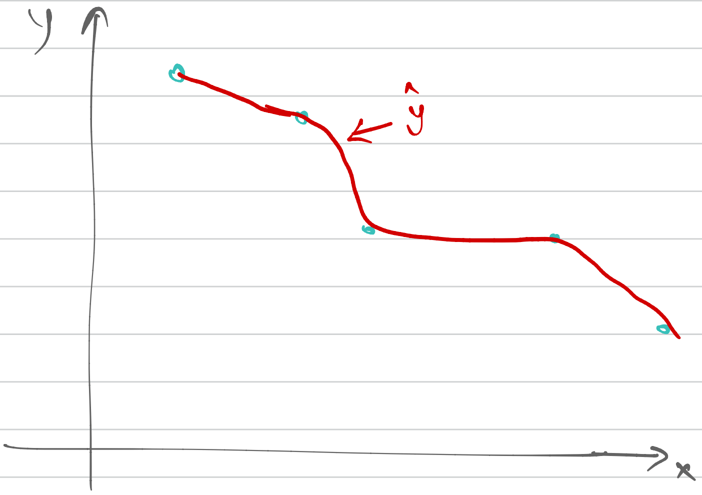
where

$z(x) =$ row vector function

$= [z_1(x), z_2(x), \ldots, z_m(x)]$

$a =$ column vector

$$= \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$



In case interpolation

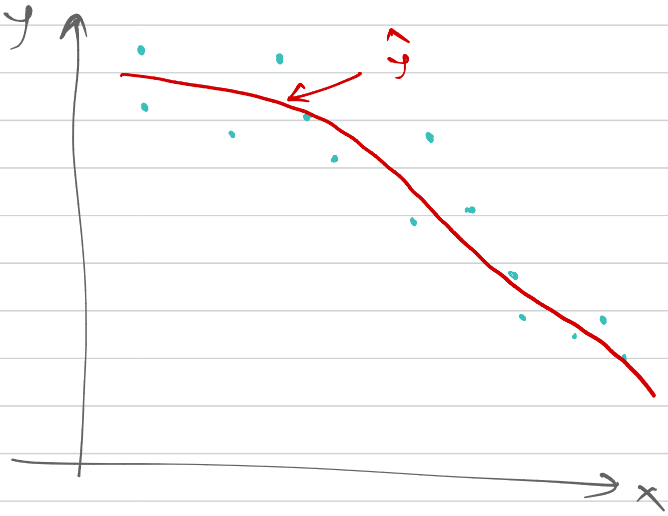
taking example of polynomial
$(z(x) = [1, x, x^2, \ldots, x^{m-1}])$

↓ we know that generally polynomial passing through n data points will be of the order of $n-1$

Regression (linear regression)

some data

Model: some model $\hat{y} = z(x)\, a$



so z must be basis of $(n-1)^{th}$ order polynomial

$$z(x) = [1, x, x^2, \dots, x^{n-1}]_{n\times 1}$$

$$m = n$$

Solving for a

Idea: match model to data at specified point $x^1, x^2, \dots, x^n$

$$\left.\begin{aligned} \hat{y}(x^1) &= y^1 \\ \hat{y}(x^2) &= y^2 \\ &\vdots \\ \hat{y}(x^n) &= y^n \end{aligned}\right\} \Rightarrow Ba = y$$

where $a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$

$$y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix}$$

$$B = \begin{bmatrix} \text{---} & z(x^1) & \text{---} \\ \text{---} & z(x^2) & \text{---} \\ & \vdots & \\ \text{---} & z(x^n) & \text{---} \end{bmatrix}$$

Solving for a

Idea: minimize errors

if error is a squared error then we have least squared method

squared errors

$$E = E(a) = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}(x^i) - y^i)^2$$

find a such that $E(a)$ is minimum

$\Updownarrow$ equivalent

Solve a using $Ja = b$

where $a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$

$$J = B^T B$$

$$b = B^T y$$

$$= \begin{bmatrix} z_1(x^1) & z_2(x^1) & \cdots & z_m(x^1) \\ z_1(x^2) & z_2(x^2) & \cdots & z_m(x^2) \\ \vdots & & & \\ z_1(x^n) & z_2(x^n) & \cdots & z_m(x^n) \end{bmatrix}$$

where B and y are defined in left-side

$$\boxed{Ba = y}$$

when can it be solved

(i) B must be square matrix

(ii) basis function must be independent

size $(B) = n \times m$

square matrix only if $m = n$

$$\boxed{Ja = b \Leftrightarrow B^T B a = B^T y}$$

(i) J must be square matrix

(ii) basis function must be independent

<u>Generally</u> $B_{n\times m}$

$$J_{m\times m} = B^T_{m\times n} \, B_{n\times m}$$

you can use polyfit to perform interpolation/regression using polynomial

$$x = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^n \end{bmatrix}, \quad y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix}, \quad p = \text{polyfit}(x, y, \ell)$$

you choose (ℓ is a order of polynomial)

If $l = n-1$ (I.e. $(n-1)^{th}$ polymial)
↳ interpolation

- Regression is solved using $B^T B a = B^T y$
  but it can be also used to solve interpolation

↓
for interpolation
↓
$m = n$

$$B = \begin{bmatrix} — & z(x^1) & — \\ & \vdots & \\ — & z(x^n) & — \end{bmatrix}_{n \times m}$$

— therefore B is a square matrix
— since I expect to be able to solve $Ba = y$
I will assume B^{-1} exist $\Rightarrow (B^T)^{-1}$ exist

— multiply both sides of $B^T B a = B^T y$ by $(B^T)^{-1}$

$$(B^T)^{-1} (B^T B a) = (B^T)^{-1} (B^T y)$$

$$\Rightarrow (B^T)^{-1} B^T \, B a = (B^T)^{-1} B^T y$$

$$\left(\begin{matrix} A^{-1}A = I \\ AA^{-1} = I \end{matrix} \right)$$

$$\Rightarrow \boxed{Ba = y}$$

$$\boxed{B^T B a = B^T y} \Longleftrightarrow \boxed{Ba = y}$$

↑
for interpolation

• System of ODEs

$$\frac{du}{dt} = Au + f \quad , \quad u(0) = u_0$$

where

A is $n \times n$ matrix (given)

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}$$ $n \times 1$ column vector function (given)

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$ $n \times 1$ column vector function (unknown)

$$u_0 = \begin{bmatrix} u_{0_1} \\ u_{0_2} \\ \vdots \\ u_{0_n} \end{bmatrix}$$ $n \times 1$ column vector (given)

what not to do:

(A)
```
u = zeros(1,n);        u0 = zeros(1,n);
u = u0; % first time step
uold = zero(1,n);

for k = 2: Nt+1
    uold = u;
    u = (I + dt*A) * uold + dt*f;
end
```

$$\underbrace{(I + dt*A)}_{\text{matrix } n \times n} * \underbrace{uold}_{1 \times n}$$

$t_1, t_2, \ldots, t_{N_t+1}$

(B) J = eye(n);

f = zeros(1,n);

uold = zeros(n,1);

$$\underbrace{\underset{n\times n}{J} * \underset{n\times 1}{uold}}_{n\times 1} + \underset{1\times n}{f};$$

u = J * uold + f;

- Saving solution in system of ODEs

$$\frac{du}{dt} = Au + f, \quad u(0) = u_0$$

discrete times: $t_1, t_2, \ldots, t_{N_t+1} = T$

(1) if you do not use solutions $u(t_1), u(t_2), \ldots$ $u(t_{N_t})$ but only require $u(t_{N_t+1})$ (i.e. $u(T)$)

```
u = zero(n,1);
u0 = zeros(n,1);
uold = zeros(n,1);
J = I + dt*A;
u = u0;   % take care of IC

for k = 2 : Nt + 1
    uold = u;
    u =      J * uold + dt*f;
```

end

u ⟶ u(T)

(ii) suppose you need solution at following time steps (proplem 2 (ii))

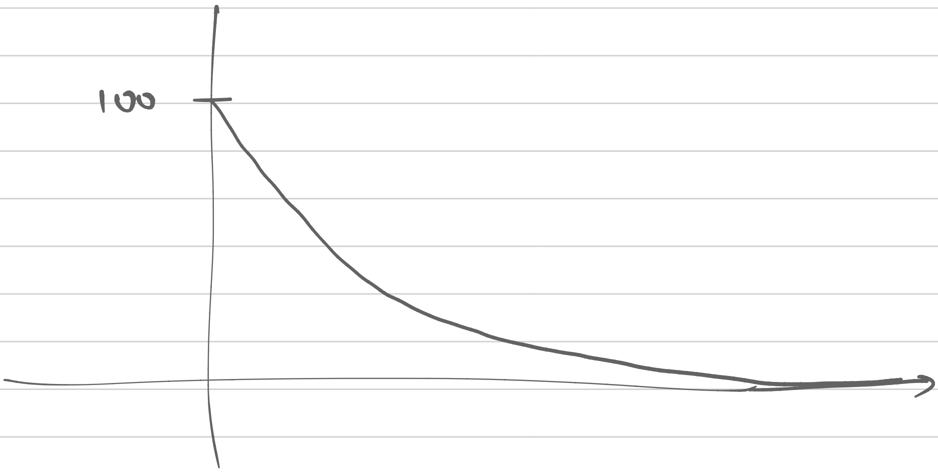
K = 201, 401, 601, 801, 1001

```
usave = zeros(n, 5);
u = zeros(n,1); uold = zeros(n,1); u0 = zeros(n,1);
count = 1;
J = I + dt*A;
u = u0;   % take care IC
for K = 2 : Nt + 1
    uold = u;
    u = J * uold + dt*f;
    if k == 201 or k == 401 or k == 601 or
       k == 801 or k == 1001
        usave(:, count) = u(:);
        count = count + 1;
    end
end
```

Problem 2 (iii): $T = 0.001$

$N_t = 1000, 100000, \ldots$ $\rightarrow \Delta t = \dfrac{T}{N_t}$

$H(T) \longrightarrow \text{plot}(H(T))$



• Discretizing PDE

$$\frac{\partial h(t,x)}{\partial t} = k(x)\frac{\partial^2 h(t,x)}{\partial x^2} + q_{ent}(t,x)$$

$x_1, x_2, \ldots, x_{N_x+1}$

Notation $\Rightarrow H_i(t) = h(t, x_i)$

write at typical x_i $(i = 2, \ldots, N_x+1)$

$$\frac{\partial h}{\partial t}(t,x_i) = k(x_i)\frac{\partial^2 h(t,x_i)}{\partial x^2} + q_{ent}(t,x_i)$$

$\boxed{\text{Case } i = 3, \ldots, N_x}$

$$\frac{\partial H_i}{\partial t} = k(x_i)\left[\frac{H_{i+1} - 2H_i + H_{i-1}}{\Delta x^2}\right] + q_{ent}(t,x_i)$$

$$[a_1, a_2, \ldots, a_n]\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

$$= \frac{k(x_i)}{\Delta x^2}\,[0, 0, \ldots, \underset{\substack{\text{at place}\\ i-2}}{1}, \underset{\substack{\text{at}\\ i-1}}{-2}, \underset{\substack{\text{at}\\ i}}{1}, 0, \ldots, 0]\begin{bmatrix} H_2 \\ H_3 \\ \vdots \\ H_{N_x+1} \end{bmatrix}$$

$\boxed{\text{Case } i = 2}$

$$\frac{\partial H_i}{\partial t} = \frac{k(x_i)}{\Delta x^2}\left[H_{i+1} - 2H_i + H_{i-1}\right] + q_{ent}(t,x_i)$$

$$= \frac{k(x_i)}{\Delta x^2}\,[-2, 1, 0, \ldots, 0]\begin{bmatrix} H_2 \\ H_3 \\ \vdots \\ H_{N_x+1} \end{bmatrix} + q_{ent}(t,x_i) + H_{i-1}\frac{k(x_i)}{\Delta x^2}$$

$\boxed{\text{Case } i = N_x + 1}$

$$\frac{\partial H_i}{\partial t} = \frac{k(x_i)}{\Delta x^2}\left[\text{from hints}\right] + q_{ent}(t,x_i)$$