< Matlab lecture in functions >

To define a function in Matlab:

```
> f = @(x, y)  x^2 + y^2 ;
> f = inline ( ' x^2 + y^2 ' );
* (function  f = myfun(x, y)
      f = x^2 + y^2 ;

> f(1, 2)          % value at (1, 2)
> fv = feval (f, xx, yy)   % values at (xx(:), yy(:))
> ezcontour(@f, [-1 1 -1 1])
```

Finite difference approximate to derivative

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Let $h > 0$ be a small real value.

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \quad : \text{forward difference}$$

$$f'(x) \approx \frac{f(x) - f(x-h)}{h} \quad : \text{backward difference}$$

$\Big\}\ O(h)$

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \quad : \text{central difference}$$

$\Big\}\ O(h^2)$

$$f''(x) \approx \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}$$

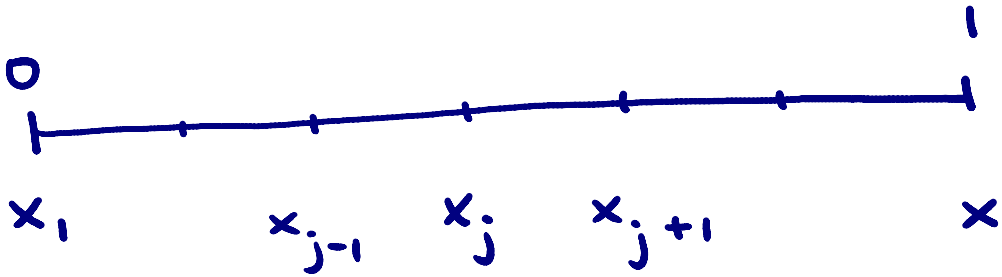
$\Big\}\ O(h^2)$

```
% function to find the derivatives
function y = prime_f(f, x, h, mode)
    f = fcnchk(f);    % accepts a string ft.
    if nargin < 3, h = sqrt(eps); mode = 'f'; end
    if nargin < 4, mode = 'f'; end
    if mode = 'f'
        y = (feval(f, x+h) - feval(f,x)) / h;
    elseif mode = 'b'
        y = (feval(f, x) - feval(f, x+h)) / h;
    else
        y = (feval(f, x+h) - feval(f, x-h)) / 2h;
    end
```

```
> f = @(x)  5x^2 + cos(x) + 2x;
> x = linspace(-pi, pi, 100);
> h = 1/100;
> fp = prime_f(f, x, h, 'f');
> fx = f(x);
> plot(x, fx, x, fp)
```

< ODE >

$$\begin{cases} u' = f(x) & \text{in } (0,1) \\ u(0) = \alpha \end{cases}$$

Let 

, $h = \frac{1}{n-1}$

(i.e. `x = linspace(0,1,n);`)

Denote by $u_j = u(x_j)$. Then,

by FDM (finite difference method)

$$u_j' \approx \frac{u_{j+1} - u_j}{h}, \quad j = 1, 2, \cdots, n-1.$$

$$\Rightarrow \quad \frac{u_{j+1} - u_j}{h} = f_j$$

$$\Rightarrow \quad u_{j+1} = u_j + h \cdot f_j \quad , \; j = 1, 2, \cdots, n-1.$$

$$* \quad u(x+h) = u(x) + h\,u'(x) + O(h^2)$$

$$\Rightarrow \quad u(x+h) \approx u(x) + h\,u'(x)$$

$$\Rightarrow \quad u_{j+1} \approx u_j + h \cdot u_j'$$

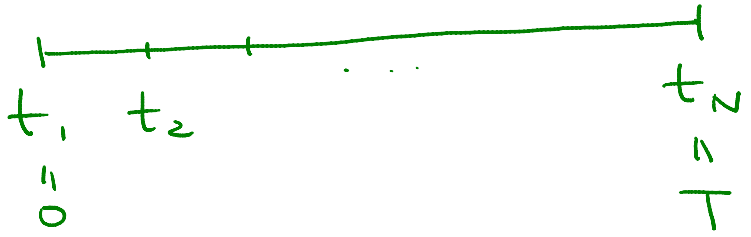
$$\Rightarrow \quad u_{j+1} \approx u_j + h \cdot f_j \qquad (\because) \; u_j' = f_j$$

< 2nd-order Boundary value problem >

$$\begin{cases} -u''(t) + a(t)\,u'(t) + b(t)\,u(t) = f(t), & (0, T) \\ u(0) = \alpha, \quad u(T) = \beta \end{cases}$$

Let $t_i = (i-1)\,h$, $h = T/(N-1)$

$u_i = u(t_i)$.



$$u_i' \approx \frac{u_{i+1} - u_{i-1}}{2h}$$

$$u_i'' \approx \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}$$

$i = 2, 3, \cdots, N-1$

$a_i = a(t_i)$, $b_i = b(t_i)$, $f_i = f(t_i)$

Substituting $t = t_i$,

$$-u_i'' + a_i u_i' + b_i u_i = f_i$$

$$\Rightarrow -\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + a_i \frac{u_{i+1} - u_{i-1}}{2h} + b_i u_i = f_i$$

$$\left(-\frac{1}{h^2} - \frac{a_i}{2h}\right) u_{i-1} + \left(\frac{2}{h^2} + b_i\right) u_i + \left(-\frac{1}{h^2} + \frac{a_i}{2h}\right) u_{i+1} = f_i \quad \left.\begin{matrix} i = 2, \cdots, \\ N-1 \end{matrix}\right\}$$

Note $u_1 = u(0) = \alpha$

$u_N = u(T) = \beta$

For $i = 2$,

$$\left(-\frac{1}{h^2} - \frac{a_2}{2h}\right) u_1 + \left(\frac{2}{h^2} + b_2\right) u_2 + \left(-\frac{1}{h^2} + \frac{a_2}{2h}\right) u_3 = f_2$$

$$\therefore \left(\frac{2}{h^2} + b_2\right) u_2 + \left(-\frac{1}{h^2} + \frac{a_2}{2h}\right) u_3 = f_2 + \left(\frac{1}{h^2} + \frac{a_2}{2h}\right) \alpha$$

For $i = 3, 4, \cdots, N-2$

$$\left(-\frac{1}{h^2} - \frac{a_i}{2h}\right) u_{i-1} + \left(\frac{2}{h^2} + b_i\right) u_i + \left(-\frac{1}{h^2} + \frac{a_i}{2h}\right) u_{i+1} = f_i$$

For $i = N-1$

$$\left(-\frac{1}{h^2} - \frac{a_{N-1}}{2h}\right) u_{N-2} + \left(\frac{2}{h^2} + b_{N-1}\right) u_{N-1} = f_{N-1} - \left(-\frac{1}{h^2} + \frac{a_{N-1}}{2h}\right) \beta$$

< Basic Algorithm >

$$\underbrace{\left(-\frac{1}{h^2}-\frac{a_i}{2h}\right)}_{A_i}u_{i-1}+\underbrace{\left(\frac{2}{h^2}+b_i\right)}_{B_i}u_i+\underbrace{\left(-\frac{1}{h^2}+\frac{a_i}{2h}\right)}_{C_i}u_{i+1}=f_i$$

$$\begin{bmatrix} B_2 & C_2 & & & \bigcirc \\ A_3 & B_3 & C_3 & & \\ & A_4 & B_4 & C_4 & \\ & & \ddots & \ddots & \ddots \\ \bigcirc & & A_{N-2} & B_{N-2} & C_{N-2} \\ & & & A_{N-1} & B_{N-1} \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ u_4 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} f_2 - A_2\alpha \\ f_3 \\ f_4 \\ \vdots \\ f_{N-2} \\ f_{N-1} - C_{N-1}\beta \end{bmatrix}$$

⟨New Algorithm⟩

For $i = 2, 3, \cdots, N-1$

$$A_i u_{i-1} + B_i u_i + C_i u_{i+1} = f_i .$$

From Boundary conditions

$$u_1 = \alpha , \qquad u_N = \beta .$$

$$\begin{bmatrix} 1 & & & & \\ A_2 & B_2 & C_2 & & \\ & A_3 & B_3 & C_3 & \\ & \ddots & \ddots & \ddots & \\ & & A_{N-1} & B_{N-1} & C_{N-1} \\ & & & & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} \alpha \\ f_2 \\ f_3 \\ \vdots \\ f_{N-1} \\ \beta \end{bmatrix}$$