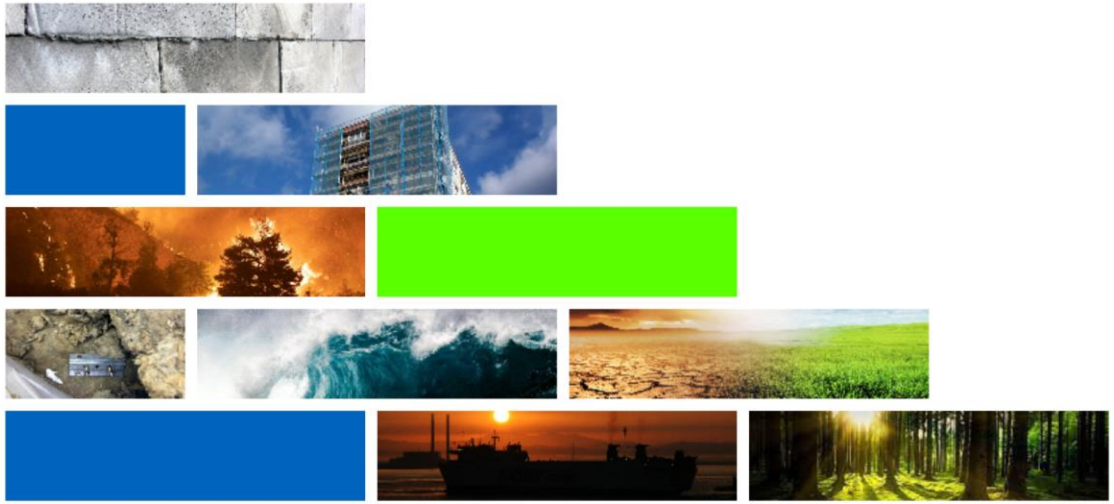
FAKULTA
STAVEBNÍ
ČVUT V PRAZE

102FY_2
Fyzika 2 G



Ing. Jan Trejbal, Ph.D.

Katedra fyziky
FSv ČVUT
Jan.trejbal@fsv.cvut.cz
http://people.fsv.cvut.cz/~trejbja4/index.html

Další možnosti využití cyklů

- Příkaz „while“
- Příklad: pomocí Newtonovy metody najděte druhou odmocninu ze zadaného reálného čísla

$$x = \sqrt{a} \rightarrow x_{i+1} = \frac{1}{2}\left(x_i + \frac{a}{x_i}\right)$$

$$x_0 = a$$

```
a = 16;
x1 = a; diference = 10;            % nulta iterace
while abs(diference) > 1e-10
x2 = 1/2*(x1 + a/x1);              % update x2
diference = x2 - x1;               % update dif
x1 = x2;                           % update x1
disp(x2);                          % zobrazeni
end
```

Podmínky – příkaz „switch“

- Větvení programu na takové případy, kdy se dotazovaná proměnná může rovnat vybraným hodnotám

```
% Priklad switch/case/otherwise


switch dotazovana promenna
  case pripad jedna
  % telo programu pro pripad jedna
  case pripad dva
  % telo programu pro pripad dva
  otherwise
  % telo programu pro ani jeden z uvedenych pripadu
end
```

Podmínky – příkaz „switch“

- Větvení programu na takové případy, kdy se dotazovaná proměnná může rovnat vybraným hodnotám

```
vek = 14;
switch vek
   case 14
   disp('Promenna vek je rovna 14.‘)
   case 15
   disp('Promenna vek je rovna 15.‘)
   otherwise
   disp('Promenna vek neni 14 ani 15.')
end
```

Vektorizace

- Vhodné pro výpočty s velkými soubory dat

```
% Priklad vektorizace souctu kazdeho licheho prvku matice
A = rand(500000,1); % Matice nahodnych cisel (radky,sloupce)
% Pomoci for cyklu
t1 = tic; % mereni casu t1 "od"
S1 = 0;
for in = 1:2:length(A) % Od 1, po 2, do delky vektoru
S1 = S1 + A(in); % scitani
end
S1
t1 = toc(t1) % mereni casu t1 "do"

% Vyuziti vektorizace
t2 = tic; % mereni casu t2 "od"
S2 = sum(A(1:2:end)) % souce vybranych prvku
t2 = toc(t2) % mereni casu t2 "do"
```

Vektorizace

```
% Priklad souctu ctvercu vybranych prvku matice
A = rand(100,100);
% Pomoci for cyklu
t1 = tic; % mereni casu t1 "od"
S1 = 0;
for in = 1:2:size(A,1) % cyklus pres radky
        for jn = 1:2:size(A,2) % cyklus pres sloupce
        S1 = S1 + A(in,jn)^2; % suma ctvercu
        end
end
S1
t1 = toc(t1) % mereni casu t1 "do"

% Vektorizaci
t2 = tic; % mereni casu t2 "od"
S2 = sum(sum(A(1:2:end,1:2:end).^2)) % mocneni teckovanym operatorem a soucet funkcemi
t2 = toc(t2) % mereni casu t2 "do"
```

Symbolické výpočty

- MATLAB umožňuje operace s analytickými vztahy
- Umožňuje derivovat a integrovat pomocí symbolických proměnných

```
% Definice symbolickych promennych
syms a b c

% Pouziti
M = a*b;
N = a/c;
```

Symbolické výpočty

- MATLAB umožňuje operace s analytickými vztahy
- Umožňuje derivovat a integrovat pomocí symbolických proměnných

```
% Definice symbolickych promennych
syms a b c

% Derivace
diff( cos(b) )
diff( sin(a*b)*exp(c), b ) % parcialni derivace podle b

% Integrace
int( cos(b) )
int( sin(a)^2, a , 0 , 1 ) % urcity integral podle a od 0 do 1
```

Symbolické výpočty

- Příklad 1: funkce harmonického kmitání je $u_{(t)} = Asin(\omega t + \varphi)$
- Napište funkci rychlosti kmitajícího bodu
- Napište funkci zrychlení kmitajícího bodu

```
% Definice symbolickych promennych
syms u_t t A Omega Phi


u_t = A*sin(Omega*t+Phi);


v_t = diff (u_t) % První derivace d/dt
a_t = diff (u_t,2) % Druha derivace d^2/dt
```

Symbolické výpočty

- Příklad 2: Jaká práce A se vykoná při stlačení nárazníkové pružiny vagonu o l = 5 cm, když síla potřebná na stlačení přímo úměrná zkrácení pružiny, tj. $F = kx$, kde $k = 6 \times 10^6 Nm^{-1}$.

```
% Definice symbolickych promennych
syms F_x k x

k = 6.e6; % Tuhost pruziny
F_x = k*x; % Predpis funkce
A_x = int (F_x, x, 0, 0.05) % Urcity integral na intervalu (0-0.05)
```

Řešení soustavy rovnic pomocí symbolických výpočtů

- Vyřešte pomocí symbolických výpočtů následující soustavu rovnic

$$3x + y - z = 7$$
$$x + 2y - 5z = 15$$
$$3x + 5y + 2z = 9$$

```
syms x y z; % Declare x and y as symbolic variables

% Define the equations
eq1 = 3*x + y - z == 7;
eq2 = x + 2*y - 5*z == 15;
eq3 = 3*x + 5*y + 2*z == 9;

% Solve the system of equations
solution = solve([eq1, eq2, eq3], [x, y, z]);

% Display the solution
x_solution = solution.x;
y_solution = solution.y;
z_solution = solution.z;

disp(['x = ', char(x_solution)]);
disp(['y = ', char(y_solution)]);
disp(['z = ', char(z_solution)]);
```

Funkce: vážený průměr

- Speciální případ MNČ

$$\bar{q} = \frac{\sum_{i=1}^{N} p_i L_i}{\sum_{i=1}^{N} p_i}$$

$$u_C(\bar{q}) = \sqrt{\frac{\sum_{i=1}^{N} p_i (\bar{q} - L_i)^2}{(N-1)\sum_{i=1}^{N} p_i}}$$

```
function [xp, uxp] = vazeny_prumer(x, p)

xp = sum(p.*x)/sum(p);

e = xp - x;

uxp = sqrt(sum(p.*(e.^2))/(length(x)-1)/sum(p));

end
```

Funkce: vážený průměr

- Výpočet studijního průměru podle počtu kreditů

Předmět	Známka	Kredity
MA1	C	7
SM1	B	5
FY1	D	4
XEPS	A	1

```
% MA1: 7
% SM1: 5
% FY1: 4
% XEPS: 1

% Znamky:
% MA1: C
% SM1: B
% FY1: D
% XEPS: A

% Obdrzene znamky
A = 1;
B = 1.5;
C = 2.0;
D = 2.5;
E = 3.0;
Znamky = [C, B, D, A];
```

Funkce: vážený průměr

- Výpočet studijního průměru podle počtu kreditů

Předmět	Známka	Kredity
MA1	C	7
SM1	B	5
FY1	D	4
XEPS	A	1

```
% Kredity
Kredity = [7, 5, 4, 1];


[p, uC] = vazeny_prumer(Znamky, Kredity);
p
uC
```

Příklad: šikmý vrh

- Těleso bylo vrženou rychlostí $v_0 = 1000\ ms^{-1}$ pod elevačním úhlem $\alpha = 30°$. Určete vzdálenost x_{max}, do které dopadlo a maximální výšku y_{max}, do které vystoupalo.

```
syms g t alpha v_0 x_max y_max

v_x = v_0*cosd(alpha);
v_y = v_0*sind(alpha) - g*t;

x = int(v_x, t)
y = int(v_y, t)

v_0 = 1000;
alpha = 30;
g = 9.81;

eq1 = v_0*sind(alpha) - g*t/2 == 0;
eq2 = v_0*cosd(alpha)*t - x_max == 0;
eq3 = v_0*sind(alpha)*t - g*(t^2)/4 - y_max == 0;


solution = solve([eq1, eq2, eq3], [t, x_max, y_max]);
```

Příklad: šikmý vrh

- Těleso bylo vrženou rychlostí $v_0 = 1000\ ms^{-1}$ pod elevačním úhlem $\alpha = 30°$. Určete vzdálenost x_{max}, do které dopadlo a maximální výšku y_{max}, do které vystoupalo.

```
solution = solve([eq1, eq2, eq3], [t, x_max, y_max]);
t_solution = solution.t;
x_max_solution = solution.x_max;
y_max_solution = solution.y_max;


disp(['t = ', char(t_solution)]);
disp(['x_max = ', char(x_max_solution)]);
disp(['y_max = ', char(y_max_solution)]);
```

FAKULTA
STAVEBNÍ
ČVUT V PRAZE

Původní verzi prezentace připravil doc. Ing. Petr Pokorný, Ph.D.