



**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/~trejba4/index.html>

Podmínky – příkaz „if“

- Základní obecné pravidlo
 - Když je příkaz za if splněn, zobrazí se přednastavená odpověď (podmínka)

% Příklad if/elseif/else

```

test1 = false;
test2 = true;
if test1
disp('Podminka 1 byla splněna.');
```

```
elseif test2
```

```
disp('Podminka 1 nebyla splněna.');
```

```
disp('Podminka 2 byla splněna.');
```

```
else
```

```
disp('Nebyla splněna žádná z podmínek.');
```

```
end
```

Podmínky – příkaz „if“

- Relační a logické operátory
 - == je rovno
 - ~= není rovno
 - < je menší
 - <= je menší nebo rovno
 - > je větší
 - >= je větší nebo rovno
 - && a zároveň (sloučení podmínek)
 - || nebo
 - ~ ne

Podmínky – příkaz „if“

% Okrajove hodnoty

```
pocet_vypitych_piv_1 = 2;
pocet_vypitych_piv_2 = 5;
pocet_vypitych_piv_3 = 10;
```

% Pocet skutečne vypitych piv

```
Piva = 11;
```

```
if ( Piva <= pocet_vypitych_piv_1 )
    disp (['Celkem nuda!']);
elseif (Piva > pocet_vypitych_piv_1) && (Piva <= pocet_vypitych_piv_2)
    disp (['Ujde to!']);
elseif (Piva > pocet_vypitych_piv_2) && (Piva <= pocet_vypitych_piv_3)
    disp (['Jsem stastny!']);
else
    disp (['Rovnou usinam, vypit ', num2str(Piva) ' piv je masakr.']);
end
```

- Použití pro opakující se část kódu
 - Iterace
 - Procházení prvku matice
 - Načítání textu po jednotlivých řádcích
 - Klíčové slovo: *for*, za ním následuje určitá proměnná, která se mění dle zadaných pravidel (vektor – jeho délka určuje počet opakování)

```

% Vypsani cisel 1 az 10
for n = 1:10
disp(['Cislo ' num2str(n)]);
end

```

Animace, využití cyklu

```
% Příklad zrychlení
```

```
a = 4;
```

```
t = linspace(0,10,100);
```

```
s_t = a*(t.^2)/2;
```

```
figure (1)
```

```
xlabel('t [s]'); ylabel('s_(t) [m]');
```

```
grid on; hold on;
```

```
xlim ([0 max(t)]); ylim([0 max(s_t)]);
```

```
for i = 1:length(s_t)
```

```
plot(t(i), s_t(i), 'x');
```

```
pause(0.1);
```

```
End
```

```
hold off
```

```
% Zrychlení
```

```
% Nezávislá proměnná
```

```
% Závislá proměnná
```

```
% Popisy os
```

```
% Mřížka a držení maxim os
```

```
% Maxima os
```

```
% Opakující se cyklus
```

```
% Plot grafu
```

```
% Casový interval zobrazení
```

```
% Konec cyklu
```

Funkce

- Definice začíná slovem „function“
- 1 volitelný parametr se píše volně za slovem „function“
- Více volitelných parametrů se uvozuje do []
- Vstupní parametry jsou za jménem v ()
- Funkce se ukládají do samostatného m-file (*.m)

```
function [vystupy] = jmeno(vstupy)
% popis funkce pro help a doc
vystupy = vstupy;
end
```

Funkce: vytvořte funkci pro num. výpočet integrálu

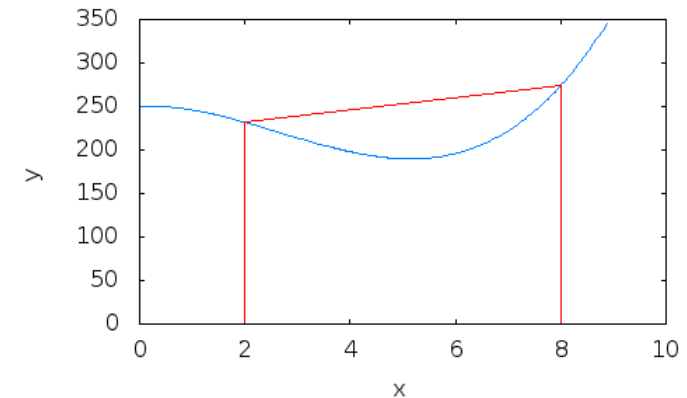
- Lichoběžníkový integrál

$$S = \int_a^b f(x) dx \approx \sum_{i=1}^N \frac{f(x_{i-1}) + f(x_i)}{2} \Delta x_i$$

```
function I = intlich(x,y)
% I = intlich(x,y)
% vypocte numericky integral lichobeznikovym pravidlem
```

```
a = x(1); b = x(end);
fa = y(1); fb = y(end);
N = length(x)-1;
```

```
I = (b - a)/N*((fa + fb)/2 + sum(y(2:end-1)));
end
```



Funkce: vytvořte funkci pro num. výpočet integrálu

- Zadaná funkce $f_{y(x)}$
- Volání funkce pro numerický výpočet integrálu

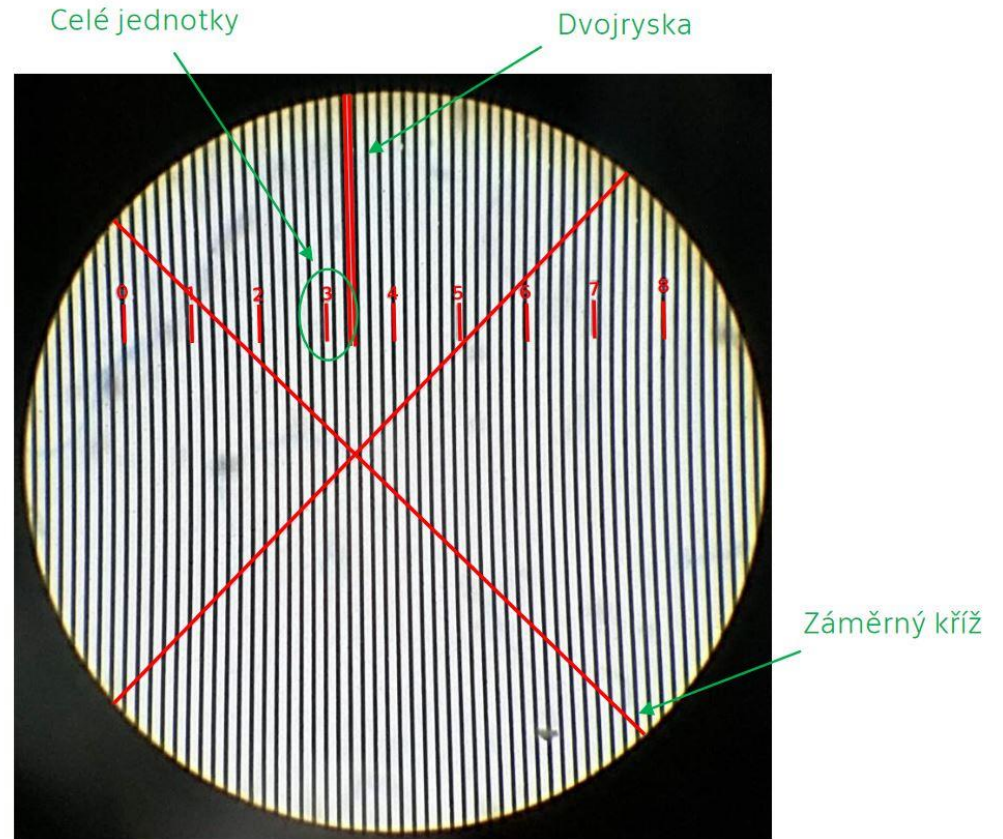
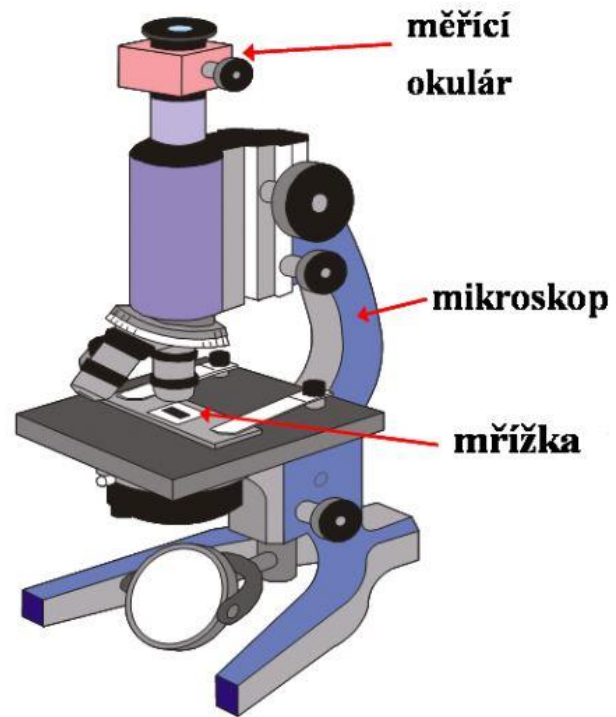
```
x = linspace(0,10,100);      % Nezavisla promenna
y = x.^2+5.*x;              % fy(x) zavisla promenna
% y = (x.^4)/4 - 7*x.^2 + 10; fy2(x) další funkce na zkousku
```

```
I = intlich(x,y)           % Vola funkci
```

Funkce: vytvořte funkci pro MPM

- V úloze A je zadáno změřit mřížkovou konstantu na mikroskopu
- Vzdálenost mezi jednotlivými úsečkami je měřena pomocí objektivového mikrometru
- Pro výpočet se používá MPM
- Na objektivu 45 byly změřeny následující hodnoty:
 - 0.03, 0.565, 1.09, 1.61, 2.095, 2.67, 3.20, 3.73, 4.24, 4.77 (viz Excel)
- Vytvořte funkci pro výpočet průměrné vzdálenosti a směrodatné odchylky

Funkce: vytvořte funkci pro MPM



Funkce: vytvořte funkci pro MPM

- Velmi užitečný pomocník je AI, např. ChatGPT
- Vyzkoušejte následující postup:
 - <https://chat.openai.com/>
 - Formulujte co nejpřesněji dotaz
 - Zkopírujte a vyzkoušejte kód
 - Pokud kód nepracuje dle představ, vysvětlete AI problém a požádejte o opravu



ChatGPT

Funkce: vytvořte funkci pro MPM

% Specify the file name and path

```
file = 'Data_1.xlsx';
```

% Read data from the Excel file

```
data = xlsread(file, 'A:A');
```

% If you have multiple columns and you want to concatenate them into a single vector

% You can use the colon operator to specify the range of columns

% For example, if you want to read data from the first column:

```
% data = xlsread(file, 'A:A');
```

% If you have multiple rows and want to concatenate them into a single vector,

% you can use the colon operator similarly for rows.

% For example, if you want to read data from the first row:

```
% data = xlsread(file, '1:1');
```

Funkce: vytvořte funkci pro MPM

% Display the imported data

```
disp('Imported data:');
```

```
disp(data);
```

% Now you have your data stored in the variable 'data' as a vector.

Funkce: vytvořte funkci pro MPM

- MPM připomenutí

x_1	x_{k+1}	$\Delta_1 = x_{k+1} - x_1$
x_2	x_{k+2}	$\Delta_2 = x_{k+2} - x_2$
\vdots	\vdots	\vdots
x_{k-1}	x_{N-1}	$\Delta_{k-1} = x_{N-1} - x_{k-1}$
x_k	x_N	$\Delta_k = x_N - x_k$

Průměrná hodnota

$$x_1, x_2, \dots, x_k, x_{k+1}, \dots, x_N \quad k = N/2$$

$$\Delta_i = x_{k+i} - x_i$$

$$\Delta x_i = \frac{\Delta_i}{k} = \frac{x_{k+i} - x_i}{k}$$

$$\overline{\Delta x} = \frac{1}{k} \sum_{i=1}^k \Delta x_i$$

Nejistota

$$u_A(\Delta x_i) = \sqrt{\frac{\sum_{i=1}^k (\Delta x_i - \overline{\Delta x})^2}{k-1}}$$

Funkce: vytvořte funkci pro MPM

- MPM varianta s podrobným postupem

%% MPM_1: Metoda postupnych mereni s rozepsanym vypoctem

function [vystup] = MPM(x,y) % x vektor vstupnich hodnot mereni musi mit sudy pocet prvku, y rozliseni pristroje

```
N=length(x); % pocet prvku
k=N/2; % pocet radku (polovina poctu prvku)
N=size(x); % dimenze matice/vektoru N
while N(2)==1 % pokud pocet sloupce ==1...
x=x'; %... transponuj na radkovy vektor
End
```

```
A=x(1:k); % Vybere z vektoru x pozice 1 az k
B=x(k+1:end); % Vybere z vektoru y pozice k+1 az N
A=vertcat(A,B)'; % Seradi A vedle B
A=A(:,2)-A(:,1); % Odecte prislusne sloupce
A=A./(k); % Podeli pocetem radku
B=1/(k)*sum(A); % Vypocte prumer
```


Funkce: vytvořte funkci pro MPM

- MPM varianta s
podrobným postupem

```
A=sqrt(sum((A-B).^2)/((k)-1));
A=A/sqrt(k);
u_b=y/sqrt(12);
u_b=1/(k)*sqrt(2/k)*u_b;
u_c=sqrt(A^2+u_b^2);
vystup=[B, u_c];
```

```
% Smerodatna odchylka
% Nejistota aritmetického prumeru
% Nejistota z rozliseni

% Vysledna nejistota
% Vysledek odeslany do volajiciho skriptu
```

Funkce: vytvořte funkci pro MPM

- Volání funkce z hlavního skriptu

```

file = 'Data_1.xlsx';           % Nacteni dat, viz vyse
data = xlsread(file, 'A:A');    % Vyber dat z prvního sloupce
x = data';                     % Transpozice na radkový vektor
rozliseni = 0.01;              % Rozliseni mikroskopu
Prum = MPM(x,rozliseni);       % Volani funkce

```

Funkce: vytvořte funkci pro MPM

- Odeslání komentovaných a správně zaokrouhlených výsledků na terminál

```

disp(['Průměrná měřena vzdálenost mezi useckami je ', num2str(Prum(1:1)) ' mm']);
disp(['Výsledná nejistota z průměru je ', num2str(round(Prum(2:2), 4)) ' mm']);

```

Funkce: vytvořte funkci pro MPM

- MPM varianta s využitím matematických a statistických funkcí
 - mean
 - std
 - sqrt

```
function [xx, uC_x] = MPM_2(x, uB_x)
k = length(x)/2;
dx = (x(k+1:end)-x(1:k))/k;
xx = mean(dx);
uA_xx = std(dx);
uB_xx = 1/k*sqrt(2/k)*uB_x;
uC_x = sqrt(uA_xx^2 + uB_xx^2);
end
```

Funkce: vytvořte funkci pro MPM

- MPM varianta s využitím matematických a statistických funkcí
 - mean
 - std
 - sqrt

```
x = data';
```

```
uB_x = 0.01;
```

```
Prum = MPM_2(x,uB_x); % Volani
```

```
disp(['Prumerna merena vzdalenost mezi useckami je ', num2str(Prum(1:1)) ' mm']);
```



**FAKULTA
STAVEBNÍ
ČVUT V PRAZE**

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

