


# ÚVOD DO INŽINIERSTVA A BEZPEČNOSŤ V ELEKTROTECHNIKE (2023)

## Cvičenie č. 7 – Úvod do MATLABU




### SKUPINA B

#### ZADANIE 1 – Úvodné nastavenie




Vytvorte nový priečinok s vaším menom na pracovnom počítači (Odporúčané umiestnenia: Moje dokumenty, Pracovná plocha). Napr.: `C:\Users\student\Documents\student`, kde `student` je novovytvorený priečinok. V prostredí Matlab v navigačnom paneli klikneme na ikonu  (*Browse for Folder*) a zadáme cestu k tomuto priečinku. Tým nastavíme pracovný priečinok, ktorého obsah sa automaticky zobrazí v ľavom paneli pod záložkou *Current Folder*. Nastavenie overte konzolovým príkazom:

```
>> pwd
```



#### ZADANIE 2 – Skript

Vytvorte nový skript v prostredí Matlab. Z hlavného menu v záložke **HOME** vyberte položku **New Script**  alebo položku **New** (ikona ) a z rolovacieho menu vyberte položku **Script** . Zobrazí sa nové okno s textovým editorom, kde zadajte nasledujúce príkazy (zdrojový kód):


```
i = 0;
for t = 0:.01:200
    i = i + 1;
    y(i) = cos(t);
end
```

Vytvorený skript uložte pomocou klávesovej skratky **Ctrl + S**, alebo kliknutím na ikonu  v hlavnom menu v záložke **EDITOR** s popisom **Save** pod názvom *prvy\_skript.m*. Skript spustíte stlačením tlačidla  **Run** v záložke **EDITOR** v hlavnom menu. Odmerajte čas vykonania tohto programu pridaním príkazu **tic** na začiatok a príkazu **toc** na koniec skriptu a jeho opätovným spustením. Zistite časy vykonania jednotlivých operácií programu pomocou profilovania spustením skriptu stlačením tlačidla **Run and Time**  z rolovacieho menu **Run** v záložke **EDITOR**.

#### ZADANIE 3 – Funkcie

Vytvorte novú funkciu v prostredí Matlab. Z hlavného menu v záložke **HOME** vyberte položku **New** (ikona ) a z rolovacieho menu vyberte položku **Function** . Zobrazí sa nové okno s textovým editorom, kde prepíšete existujúce príkazy (zdrojový kód) nasledovne:

```
function [y] = funkcia_iter(xtime)
    i = 0;
    for t = xtime
        i = i + 1;
        y(i) = cos(t);
    end
end
```

Vytvorenú funkciu uložte pomocou klávesovej skratky **Ctrl + S**, alebo kliknutím na ikonu  v hlavnom menu v záložke **EDITOR** s popisom **Save** pod názvom *funkcia\_iter.m*. Poznámka: *Názov funkcie a názov uloženého súboru v ktorom sa daná funkcia nachádza sa musia vždy zhodovať.* Funkciu po uložení zavoláme z akéhokoľvek skriptu, alebo priamo konzolovým príkazom:

```
>> y=funkcia_iter(0:0.1:200);
```

Podľa predchádzajúceho postupu vytvorte novú funkciu v prostredí Matlab s názvom *funkcia\_vekt* a obsahom:

```
function [y] = funkcia_vekt(xtime)
    y = cos(xtime);
end
```

Funkciu po uložení zavoláme z akéhokoľvek skriptu, alebo priamo konzolovým príkazom:

```
>> y=funkcia_vekt(0:0.1:200);
```

Porovnajme časy vykonania funkcií *funkcia\_iter* a *funkcia\_vekt* pomocou príkazu **timeit** novým skriptom s ľubovoľným názvom a obsahom:

```
% vytvoríme si casovy vektor t
t = 0:.01:200;
% vytvoríme handler (pointer) na funkcie
f_iter = @() funkcia_iter(t);
f_vekt = @() funkcia_vekt(t);
% odmerame cas vykonania funkcii
cas_vykonania_iter = timeit(f_iter)
cas_vykonania_vekt = timeit(f_vekt)
```

Sú obe implementácie zhodné z hľadiska výstupu a času vykonania? Diskutujte o dosiahnutých výsledkoch s vyučujúcim.

#### ZADANIE 4 – Odlad'ovanie chýb (debugovanie)

Podľa predchádzajúceho postupu vytvorte novú funkciu v prostredí Matlab s názvom *mystery\_fcn* a obsahom:

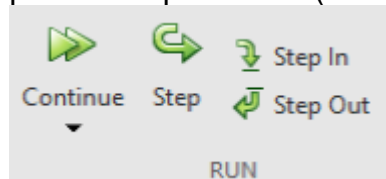
```
function [res] = mystery_fcn(n)
    for i=1:n
        r = sqrt(5);
        phi = (1+r)/2;
        psi = (1-r)/2;
        mystery_val = (phi^i - psi^i)/r;
    end
    res = mystery_val;
end
```


Pomocou odlad'ovania chýb (debugovania) v prostredí matlab zistite posledných 5 hodnôt premennej *mystery\_val* vo vnútri funkcie *mystery\_fcn* pri jej zavolaní príkazom:

```
>> res=mystery_fcn(10);
```

**Pomôcka:** Vo vnútri funkcie *mystery\_fcn* vytvoríme bod zastavenia tzv. **Breakpoint** nastavením kurzora na riadok, kde chceme daný bod zastavenia vytvoriť a stlačením klávesy **F12**, alebo kliknutím na číslo riadku pri editovaní funkcie. Nastavený breakpoint je reprezentovaný červeným štvorcom (s číslom riadku).

Pre spustenie módu odlad'ovania zavoláme funkciu s nastaveným bodom zastavenia priamo z konzoly (alebo stlačením tlačidla **Run** v prípade spustenia skriptu). V odlad'ovacom móde sledujeme premenné prostredia (tzv. **Workspace**) a posúvame sa postupnou exekúciou príkazov pomocou tlačidiel



Odlad'ovací režim zrušíme stlačením tlačidla **Stop** .