

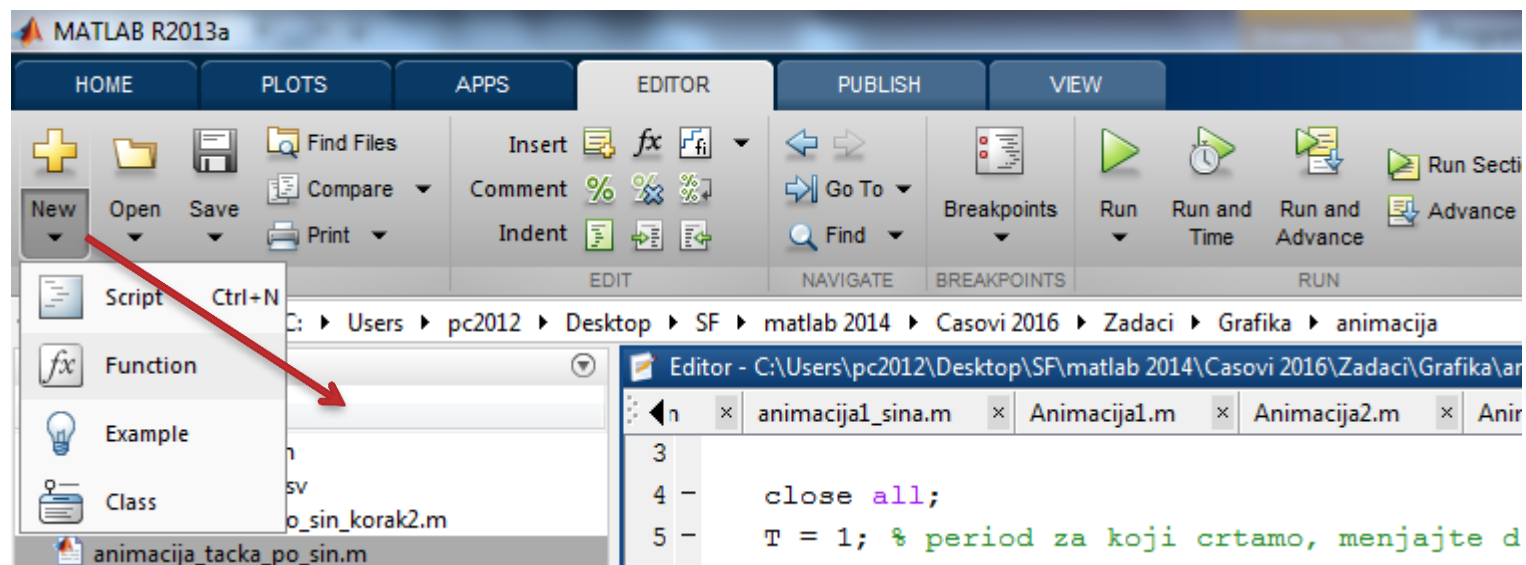


Matematički programski paketi
u saobraćaju i transportu

Funkcije i funkcijske
datoteke

Pravljenje funkcijske datoteke

- Pišu se i uređuju u Editoru, kao i skriptovi
- New, Function



Struktura funkcijske datoteke

- Prvi red funkcije mora biti red sa njenom definicijom
- **function [izlazni argumenti] = ime_funkcije (ulazni argumenti)**
- Izlazni argumenti dobijaju vrednosti koje im se dodeljuju u programu unutar tela funkcije. Pažnja – uvek se navode unutar uglastih zagrada
- Ulazni argumenti se navode unutar malih zagrada, razdvajaju se zarezima ako ih ima više i vrednosti im se navode pri pozivu funkcije.

Primer 1

function [y] = fun1(x) % može i bez [] ukoliko
funkcija ima samo jedan izlazni argument

```
y=exp(x^2)/sqrt(x^2+5);
```

```
end
```

- Funkcija sa jednim ulaznim argumentom i jednim izlaznim argumentom.

Pozivamo je sa

```
>>fun1(2)
```

Ili

```
>>y5 = fun1(5)
```

Primer 2

```
function z = fun2(x,y)
```

```
u = 3*x;
```

```
z = u + 6*y.^2;
```

```
end
```

```
>>fun2(3,7)
```

- x i y su LOKALNE promenljive za funkciju, što znači da su definisane i imaju vrednost samo unutar funkcije.
- Funkciju možemo pozvati i sa vektorima kao argumentima.

Lokalne i globalne promenljive

- Ako definišemo x i y u komandnom prozoru sa

```
>>x=3,y=7;
```

I pozovemo $\text{fun2}(x,y)$ dobićemo očekivanu vrednost.

```
>> u
```

I dalje nije vidljiva iz komandnog prozora, ostaje lokalna promenljiva za funkciju fun2 .

Globalne promenljive

- Ukoliko želimo da više funkcija deli određenu promenljivu, moramo je deklarirati kao globalnu promenljivu.
- `global ime_promenljive`
- Preporučuje se korišćenje dugih i opisnih imena za globalne promenljive, na primer `gravitaciona_konstanta` ili slično zbog lakšeg uočavanja razlike između njih i običnih promenljivih.

Broj ulaznih i izlaznih argumenata

- `nargin('ime_funkcije')` – određuje broj ulaznih argumenata funkcije

Primeri: `nargin('sin')=1`, `nargin('rem')=2`

- `nargout('ime_funkcije')` – određuje broj izlaznih argumenata funkcije

Primeri: `nargout('sin')=1`, `nargout('size')=2`

- Pogledati Matlab-ovu funkciju `sphere`

`>> type sphere`

Primer

- Napisati funkciju tablicamnoz čiji je broj ulaznih i izlaznih argumenata promenljiv na sledeći način:
 1. Izlazni argument je $n \times m$ tablica množenja za ulazne argumente m i n
 2. Opciono, izlazni argument može biti i suma svih elemenata u tablici
 3. Ako je ulazni argument samo i izlaz je $n \times n$ tablica

```
function [tablica suma] = tablmnoz(n, m)
if nargin < 2
m = n;
end
tablica = (1:n)' * (1:m);
if nargout == 2
suma = sum(tablica(:));
end
```

```
>> tablmnoz(4,3)
>> [s,t]=tablmnoz(5)
>> tablmnoz(-1)
```

Resenje 2 :

```
function [tablica suma] = tablmnoz2(n, m)
if nargin < 1
error('mora imati bar jedan ulazni argument');
end
if nargin < 2
m = n;
elseif ~isscalar(m) || m < 1 || m ~= fix(m)
error('m mora biti prirodan broj');
end
if ~isscalar(n) || n < 1 || n ~= fix(n)
error('n mora biti prirodan broj ');
end
tablica = (1:n)' * (1:m);
if nargout == 2
suma = sum(tablica(:));
end
```

Snimanje i upotreba funkcija

- Poželjno je da funkciju čuvamo pod imenom koje smo koristili u redu sa definicijom funkcije.
- Funkcija koju je definisao korisnik upotrebljava se na isti način kao ugrađena Matlab-ova funkcija.
- Direktorijum u kome je sačuvana funkcija mora biti tekući ili naveden u putanji pretraživanja

Anonimne i lokalne funkcije

- Funkcija koju definiše korisnik, koja je relativno jednostavan matematički izraz i definiše se unutar programa (u komandnom prozoru), a ne kao zasebna datoteka. Definišu se:

ime = @ (lista argumenata) mat.izraz

- Anonimne funkcije u Matlab-u zamenile su takozvane inline funkcije ili lokalne funkcije koje se definišu na sledeći način:
- ime = inline('mat.izraz zapisan kao string')

Primeri:

```
>> Vkocke = @ (x) x^3  
Vkocke(2)
```

```
>> krug = @ (x,y) 16*x^2 + 9*y^2
```

```
>> parabola = @ (x) x^2 + x + 1
```

```
>> f = inline('exp(x.^2)./sqrt(x)')  
f(2) ili feval(f,2) – function evaluate
```

```
>> F = 'cos', feval(F,pi/2)
```

```
>> G = @ sin, feval(G,pi/2)
```

Zadaci:

1. Napisati funkciju `krug` koja izračunava obim i površinu kruga. Ulazni argument je poluprečnik kruga.
2. Napisati funkciju `vektpr` koja izračunava vektorski proizvod dva vektora koji su ulazni argumenti.
3. Napisati funkciju `polar` koja Dekartove koordinate prebacuje u polarne koordinate.
4. Napisati funkciju po imenu `fact1` koja računa faktorijel prirodnog broja n koristeći `for` petlju. Ulazni argument je n , a izlaz je $n!$ Zatim napisati funkciju `fact2` koja računa faktorijel pomoću rekurzije.

5. Napisati funkciju `matrica1` koja ima ulazne argumente m i n , a kao izlazni argument matricu Q formata $2n \times 2m$ koja se sastoji od $4n \times m$ podmatrice. Elementi podmatrice u gornjem levom uglu su svi jednaki 0, elementi podmatrice u gornjem desnom uglu su svi jednaki 1, donji levi su jednaki 2 i donji desni su jednaki 3.

6. Napisati funkciju `mojatabela1` koja nema ni ulaznih ni izlaznih argumenata a treba da napravi tabelu od 4 kolone a , b , c i d na sledeci nacin:

- kolona a su kvadrati prvih 10 prirodnih brojeva
- kolona b su koreni prvih 10 prirodnih brojeva
- kolona c je aritmeticka sredina kolona a i b
- kolona d je razlika kolone b i kolone a , $d=b-a$.

7. Napisati funkciju mnozenjematrix koja kao ulazni argument ima dve matrice A i B, a kao izlazni argument daje $C=A*B$ (ne koristeci Matlab-ovu operaciju *) ukoliko je mnozenje moguće ili adekvatno obavestenje ukoliko nije moguće pomnoziti matrice A i B.

8. Napisati funkciju Kramer koja resava sistem linearnih jednačina $AX=B$ primenom Kramerovog pravila. Ulazni argumenti su A i B, a izlazni argument resenje X ukoliko postoji, ili adekvatno obavestenje u suprotnom.

9. Napisati funkciju nadji_nulu (f,x1,x2) koja nalazi nulu funkcije f na intervalu $[x1, x2]$ sa tacnoscu $1e-10$ koristeci Bolcano-Kosijevu teoremu. Na granicama intervala $[x1,x2]$ funkcija f mora imati suprotan znak.