

Odvajanje in integriranje

11. VAJA

11.1 Odvajanje tabelarično podane funkcije

Tabeliraj pet ekvidistančnih točk in izračunaj prvi, drugi, tretji in četrti odvod tabelarične funkcije v središčni točki po centralno diferenčnih formulah:

$$y'_i = \frac{y_{i+1} - y_{i-1}}{2h}$$

$$y''_i = \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}$$

$$y'''_i = \frac{-y_{i-2} + 2y_{i-1} - 2y_{i+1} + y_{i+2}}{2h^3}$$

$$y^{IV}_i = \frac{y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2}}{h^4}$$

$$h = x_{i+1} - x_i$$

x	y
2	4
3	6
4	2
5	8
6	3

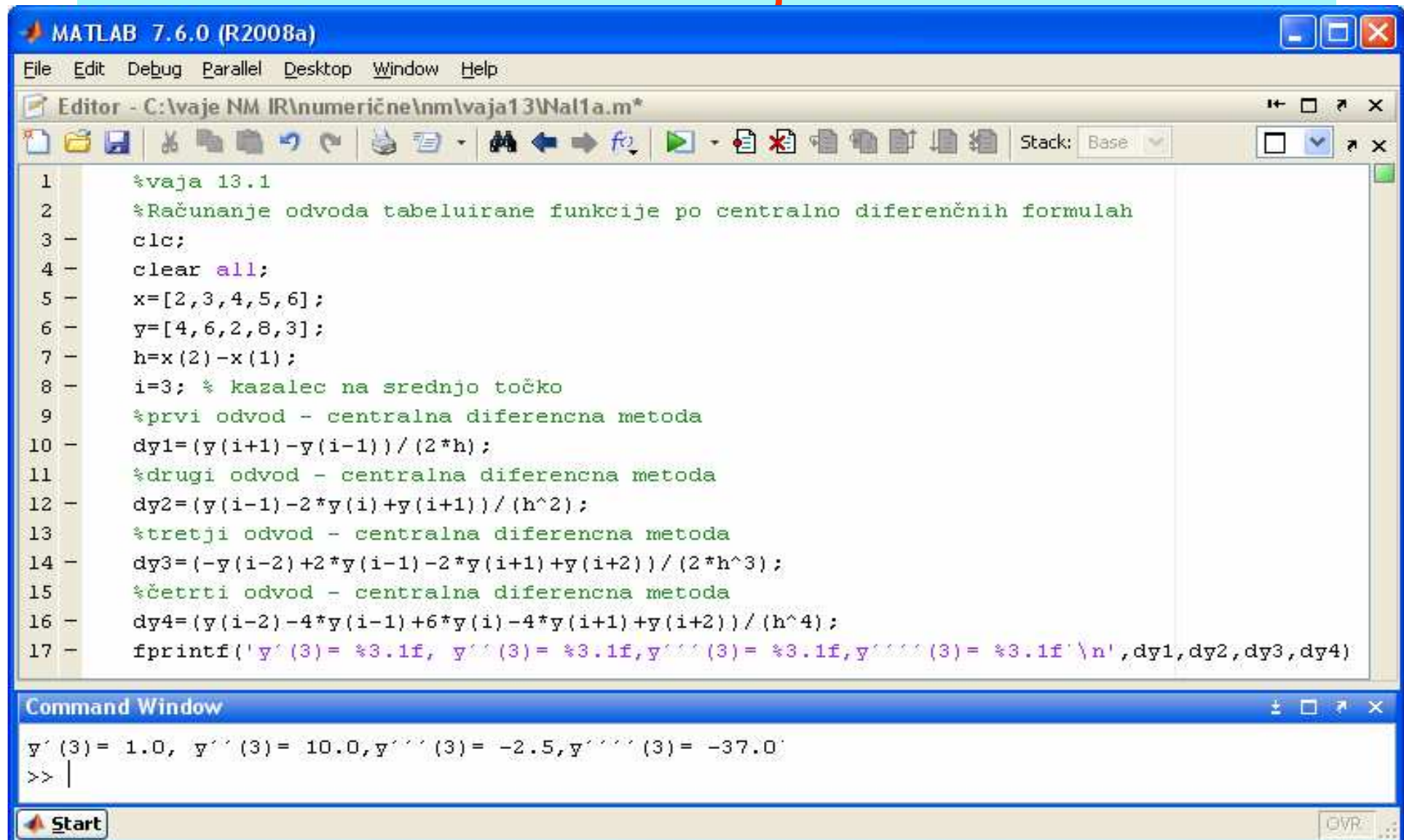
$$y'(4) = 1$$

$$y''(4) = 10$$

$$y'''(4) = -2.5$$

$$y^{IV}(4) = -37$$

11.1 Odvajanje tabelarično podane funkcije



The image shows a MATLAB 7.6.0 (R2008a) window with a script editor and a command window. The script editor contains the following code:

```
1 %vaja 13.1
2 %Računanje odvoda tabelirane funkcije po centralno diferenčnih formulah
3 - clc;
4 - clear all;
5 - x=[2,3,4,5,6];
6 - y=[4,6,2,8,3];
7 - h=x(2)-x(1);
8 - i=3; % kazalec na srednjo točko
9 %prvi odvod - centralna diferenčna metoda
10 - dy1=(y(i+1)-y(i-1))/(2*h);
11 %drugi odvod - centralna diferenčna metoda
12 - dy2=(y(i-1)-2*y(i)+y(i+1))/(h^2);
13 %tretji odvod - centralna diferenčna metoda
14 - dy3=(-y(i-2)+2*y(i-1)-2*y(i+1)+y(i+2))/(2*h^3);
15 %četrti odvod - centralna diferenčna metoda
16 - dy4=(y(i-2)-4*y(i-1)+6*y(i)-4*y(i+1)+y(i+2))/(h^4);
17 - fprintf('y'(3)= %3.1f, y''(3)= %3.1f,y'''(3)= %3.1f,y''''(3)= %3.1f\n',dy1,dy2,dy3,dy4)
```

The Command Window shows the output of the script:

```
y'(3) = 1.0, y''(3) = 10.0,y'''(3) = -2.5,y''''(3) = -37.0
>> |
```

The window title is "MATLAB 7.6.0 (R2008a)" and the editor title is "Editor - C:\vaje NM IR\numerične\nm\vaja13\Wal1a.m*". The Command Window title is "Command Window".

11.2 Odvajanje z Matlab funkcijami

Naj bo pot s , ki jo opravi telo, podana s funkcijo
 $s = 3t^4 + 5t^3 + 2t^2 + 6t + 7$.

Izračunaj hitrost v in pospešek a s funkcijami:

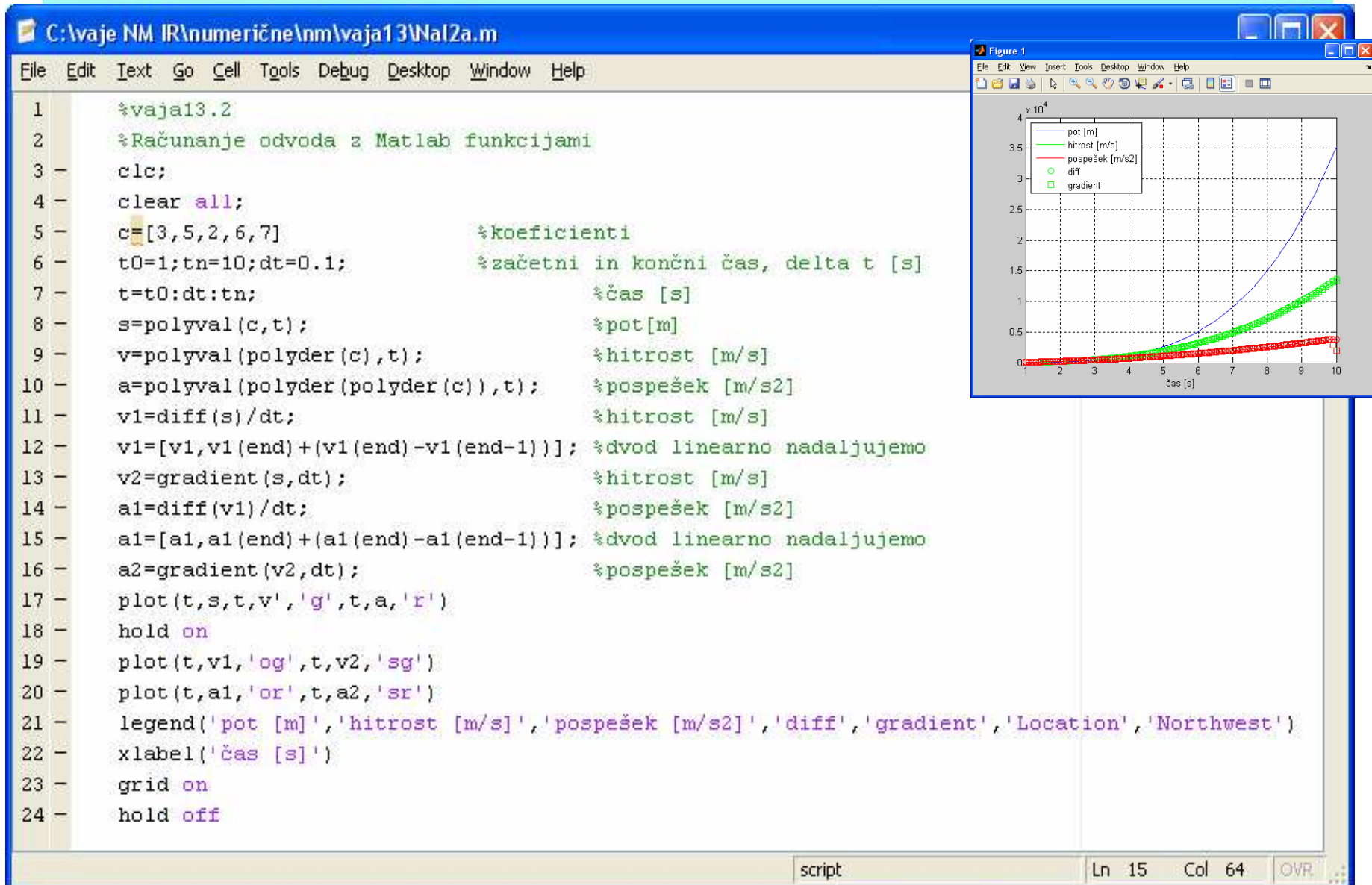
-*polyder*

-*diff*

-*gradient*

$$v(t) = \frac{ds}{dt}; \quad a(t) = \frac{dv}{dt}$$

11.2 Odvajanje z Matlab funkcijami



11.3 Integriranje po Trapezni metodi

Izračunaj določeni integral po trapezni metodi na 5 mest natančno. Rezultat preveri z Matlab orodji za integriranje.

$$I = \int_{-1}^1 \cos(x^2) e^x dx$$

11.3 Integriranje po Trapezni metodi

```
Editor - C:\vaje NM IR\numerične\nm\vaja13\Na13b.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1      %Trapezna metoda
2 -    clc; clear all;
3 -    f=inline('cos(x.^2).*exp(x)','x');
4 -    a=-1;b=1;n=2;
5 -    x=linspace(a,b,n);
6 -    y=f(x);
7 -    eps=10e-6;
8 -    h=(b-a)/(n-1);
9      %sestavljena trapezna formula
10 -   TS=h/2*sum([y(1),2*y(2:n-1),y(n)]);
11 -   for i=1:100
12 -       n=2*n-1;
13 -       h=(b-a)/(n-1);
14 -       x=linspace(a,b,n);y=f(x);
15 -       TN=h/2*sum([y(1),2*y(2:n-1),y(n)]);
16 -       T=trapz(x,y); %Matlabova funkcija
17 -       if abs((TN-TS)/3)<eps, break, end;
18 -       TS=TN;
19 -   end
20     %izpis
21 -   fprintf('Vrednost integrala %15.10f\n',TN);
22 -   fprintf('Vrednost integrala (matlab) %15.10f\n',T);
23     %Test
24     %Vrednost integrala      2.0872240820
25     %Vrednost integrala (matlab)  2.0872240820
script Ln 25 Col 45 OVR
```

11.4 Integriranje po Simpsonovi metodi

Izračunaj določeni integral po Simpsonovi metodi na 5 mest natančno. Rezultat preveri z Matlab orodji za integriranje.

$$I = \int_{-1}^1 \cos(x^2) e^x dx$$

11.4 Integriranje po Simpsonovi metodi

```
C:\vaje NM IR\numerične\nm\vaja13\Na14a.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
1 %Simpsonova metoda
2 - clc; clear all;
3 - f=inline('cos(x.^2).*exp(x)','x');
4 - a=-1;b=1;n=3;
5 - x=linspace(a,b,n);
6 - y=f(x);
7 - eps=10e-6;
8 - h=(b-a)/(n-1);
9 %sestavljena simpsonova formula
10 - SS=h/3*sum([y(1),2*y(3:2:n-2),4*y(2:2:n-1),y(n)]);
11 - for i=1:100
12 -     n=2*n-1;
13 -     h=(b-a)/(n-1);
14 -     x=linspace(a,b,n);y=f(x);
15 -     SN=h/3*sum([y(1),2*y(3:2:n-2),4*y(2:2:n-1),y(n)]);
16 -     if abs((SN-SS)/15)<eps, break, end;
17 -     SS=SN;
18 - end
19 - S=quad(f,a,b,eps);%Matlabova funkcija
20 %izpis
21 - fprintf('Vrednost integrala %15.10f\n',SN);
22 - fprintf('Vrednost integrala (quad) %15.10f\n',S);
23 %Test
24 %Vrednost integrala      2.0872255905
25 %Vrednost integrala (quad)  2.0872288223
script Ln 8 Col 15 OVR
```

11.5 Integriranje po Rombergovi metodi

Izračunaj določeni integral po Rombergovi metodi na 5 mest natančno. Rezultat preveri z Matlab orodji za integriranje.

$$I = \int_{-1}^1 \cos(x^2) e^x dx$$

Rombergova formula:

$$R_{i,j} = \frac{4^{j-1} R_{i,j-1} - R_{i-1,j-1}}{4^{j-1} - 1}, (j = 2, \dots, i), (i = 2, \dots, k)$$

Prve vrednosti $R_{i,1}$ izračunamo po Trapezni metodi.

11.5 Integriranje po Rombergovi metodi

```
Editor - C:\WAJENM~1\NUMERI~1\nm\vaja13\Na14a.m
File Edit Text Desktop Window Help
[Icons]
1 %Rombergova metoda
2 clc; clear all;
3 f=inline('cos(x.^2).*exp(x)','x');
4 a=-1;b=1;n=2;
5 x=linspace(a,b,n);
6 y=f(x);
7 eps=10e-6;
8 h=(b-a)/(n-1);
9 %Začnemo s Trapezno metodo
10 R(1,1)=trapz(x,y);
11 for i=2:100
12     n=2*n-1;
13     x=linspace(a,b,n);
14     y=f(x);
15     R(i,1)=trapz(x,y);
16     for j=2:i
17         R(i,j)=(4^(j-1)*R(i,j-1)-R(i-1,j-1))/(4^(j-1)-1);
18     end
19     if abs(R(i,i)-R(i-1,i-1))<eps; break; end
20 end
21 format long
22 R %izpis
23 fprintf('Vrednost integrala po Rombergovi metodi %15.10f\n',R(i,i));
24 fprintf('Vrednost integrala (quad) %15.10f\n',quad(f,a,b,eps));
25 %Test
26 %Vrednost integrala      2.0872255905
27 %Vrednost integrala (quad)  2.0872288223
script Ln 27 Col 43 OVR
```