

# Numerična napaka

## 7. VAJA

## 7.1 Numerična napaka-kvadratna enačba

Izračunaj kvadratno enačbo  $Ax^2 + Bx + C = 0$

Rešitve izračunaj na dva načina:

$$A = 1; \quad B = 1, 10, 100, \dots, 10^7; \quad C = -3.$$

$$x_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}.$$

$$x_{1,2} = \frac{-2C}{B \pm \sqrt{B^2 - 4AC}}.$$

Izpeljava:

$$x_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}.$$

$$x_{1,2} = \frac{(-B \pm \sqrt{B^2 - 4AC})(-B \mp \sqrt{B^2 - 4AC})}{2A(-B \mp \sqrt{B^2 - 4AC})} =$$

$$x_{1,2} = \frac{B^2 - B^2 + 4AC}{2A(-B \mp \sqrt{B^2 - 4AC})} = \frac{2C}{-B \mp \sqrt{B^2 - 4AC}}$$

Naredi preizkus tako, da rešitve vstaviš v kvadratno enačbo.  
Kaj ugotoviš?

\* Primer je vzet iz knjige J. Petrišič, Reševanje enačb, FS

# 7.1 Numerična napaka-kvadratna enačba

```
C:\vaje NM IR\numerične\nm\vaja 8\Untitled14.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
1      %napaka kvadratne enačbe
2      clc; clear all; format long
3      a=1;b=1;c=-3;
4      disp '          B          x1          x2          r plus          r minus'
5      for i=1:9
6          %rešitev prvi način
7          x1=(-b+sqrt(b*b-4*a*c))/(2*a);
8          x2=(-b-sqrt(b*b-4*a*c))/(2*a);
9          %preizkus
10         r1=a*x1^2+b*x1+c;
11         r2=a*x2^2+b*x2+c;
12         fprintf('%11i  %+15.9f  %+15.9e  %+15.9f  %+15.9f \n',b,x1,x2,r1,r2)
13         b=b*10;
14     end
15     disp ' '
16     disp '          B          x1          x2          r plus          r minus'
17     b=1;
18     for i=1:9
19         %rešitev drugi način
20         y1=-2*c/(b+sqrt(b*b-4*a*c));
21         y2=-2*c/(b-sqrt(b*b-4*a*c));
22         %preizkus
23         k1=a*y1^2+b*y1+c;
24         k2=a*y2^2+b*y2+c;
25         fprintf('%11i  %+15.9f  %+15.9e  %+15.9f  %+15.9f \n',b,y1,y2,k1,k2)
26         b=10^i;
27     end
script Ln 27 Col 4 OVR
```

# 7.1 Numerična napaka-kvadratna enačba

## Rezultati

```
Command Window
File Edit Debug Desktop Window Help
  B      x1      x2      r plus      r minus
  1      +1.302775638 -2.302775638e+000 +0.000000000 +0.000000000
  10     +0.291502622 -1.029150262e+001 +0.000000000 +0.000000000
  100    +0.029991005 -1.000299910e+002 +0.000000000 +0.000000000
  1000   +0.002999991 -1.000003000e+003 +0.000000000 +0.000000000
  10000  +0.000300000 -1.000000030e+004 -0.000000004 -0.000000015
  100000 +0.000030000 -1.000000000e+005 +0.000000289 +0.000000000
  1000000 +0.000003000 -1.000000000e+006 +0.000022843 +0.000000000
  10000000 +0.000000300 -1.000000000e+007 -0.001141310 +0.000000000
  100000000 +0.000000030 -1.000000000e+008 -0.019767761 +1.000000000

  B      x1      x2      r plus      r minus
  1      +1.302775638 -2.302775638e+000 +0.000000000 +0.000000000
  10     +0.291502622 -1.029150262e+001 +0.000000000 -0.000000000
  100    +0.029991005 -1.000299910e+002 +0.000000000 -0.000000001
  1000   +0.002999991 -1.000003000e+003 +0.000000000 -0.000009027
  10000  +0.000300000 -1.000000031e+004 +0.000000000 +0.132542491
  100000 +0.000030000 -9.999999040e+004 +0.000000000 -963.038169861
  1000000 +0.000003000 -9.999923856e+005 +0.000000000 -7614380.410278320
  10000000 +0.000000300 -1.000380581e+007 +0.000000000 +38072620860.609375000
  100000000 +0.000000030 -1.006632960e+008 +0.000000000 +66769561583613.000000000
>> |
```

## 7.1 Numerična napaka-kvadratna enačba

Na primeru vidimo, da dobimo pravilne rezultate pri

$$x_1 = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \quad \text{in} \quad x_2 = \frac{-2C}{B + \sqrt{B^2 - 4AC}}.$$

Kako bi združili oba rezultata pri reševanju kvadratne enačbe?

Opomba: funkcija  $\text{sign}()$  je definirana kot:

$$\text{sign}(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases}$$

# 7.1 Numerična napaka-kvadratna enačba

The image shows the MATLAB 7.6.0 (R2008a) interface. The Command Window displays the results of a script that solves a quadratic equation. The results are as follows:

Variable	Value	Label
x1	-10000000.000000300	vrednost
x2	+0.000000300	vrednost

The Editor window shows the following script:

```
1 %rešitev kvadratne enačbe s kombinacijo dveh enačb
2
3 clc;
4 clear all;
5
6
7 a=1;
8 b=1.e7;
9 c=-3;
10
11 x1=(-b-sign(b)*sqrt(b*b-4*a*c))/(2*a);
12 r1=a*x1^2+b*x1+c;
13
14 x2=-2*c/(b+sign(b)*sqrt(b*b-4*a*c));
15 r2=a*x2^2+b*x2+c;
16 disp '          x1          vrednost!'
17 fprintf('%+25.9f %+15.9f \n',x1,r1)
18 disp '          x2          vrednost!'
19 fprintf('%+25.9f %+15.9f \n',x2,r2)
20
```

$$x_1 = \frac{-B - \text{sign}(B)\sqrt{B^2 - 4AC}}{2A}.$$

$$x_2 = \frac{-2C}{B + \text{sign}(B)\sqrt{B^2 - 4AC}}.$$

## 7.2 Numerična napaka-integral

Izračunajte integral  $I_n = \int_0^1 e^{1-t} t^n dt$  s pomočjo rekurzijske formule

$$I_{n+1} = (n+1)I_n - 1; \quad n = 0, 1, 2, \dots \quad I_0 = e - 1$$

Izpeljava (integriranje po delih  $u = e^{-t}; dv = t^n dt$ ):

$$I_n = \int_0^1 e^{-t} t^n dt = \frac{e}{n+1} \left( t^{n+1} e^{-t} \Big|_0^1 + \int_0^1 e^{-t} t^{n+1} dt \right) = \frac{1}{n+1} (1 + I_{n+1}) \Rightarrow I_{n+1} = (n+1)I_n - 1$$

Izračunajte integral tudi pri obrnjeni rekurzijski formuli

$$I_n = \frac{I_{n+1} + 1}{n+1}$$

Opomba: Matlabova funkcija `single()` računa z manjšo natančnostjo (4 byti)

Matlabova funkcija `double()` računa z večjo natančnostjo (8 bytov)

# 7.2a Numerična napaka-integral

The image shows the MATLAB 7.6.0 (R2008a) environment. The Editor window displays a script named 'vaja8\_4.m' with the following code:

```
1 - clc;
2 - clear all;
3 - format long
4 - n=0;
5 - Ie=single(exp(1))-1; %enojna natančnost 4 byti
6 - Id=double(exp(1))-1; %dvojna natančnost 8 bytov
7 - fprintf(' n I(enojna natančnost) I(dvojna natančnost)\r');
8 - fprintf('%3i %25.6f %25.6f\r',n,Ie,Id);
9 - for n=0:21
10 - Ie=(n+1)*single(Ie)-1; %enojna natančnost 4 byti
11 - Id=(n+1)*double(Id)-1; %dvojna natančnost 8 bytov
12 - fprintf('%3i %25.6f %25.6f\r',n+1,Ie,Id);
13 - end
14 - disp(' ')
15 - disp('Spremenljivka Ie je zapisana v enojni natančnosti')
16 - whos Ie
17 - disp('Spremenljivka Id je zapisana v dvojni natančnosti')
18 - whos Id
19
```

The Command Window displays the output of the script, showing a table of values for  $n$ ,  $I$  (single precision), and  $I$  (double precision). The values for  $I$  (single precision) are rounded to 6 decimal places, while the values for  $I$  (double precision) are shown in full long format.

n	I(enojna natančnost)	I(dvojna natančnost)
0	1.718282	1.718282
1	0.718282	0.718282
2	0.436563	0.436564
3	0.309690	0.309691
4	0.238762	0.238764
5	0.193810	0.193819
6	0.162857	0.162916
7	0.139999	0.140415
8	0.119995	0.123323
9	0.079956	0.109911
10	-0.200439	0.099112
11	-3.204834	0.090234
12	-39.458008	0.082808
13	-513.954102	0.076508
14	-7196.357422	0.071119
15	-107946.359375	0.066784
16	-1727142.750000	0.068539
17	-29361428.000000	0.165170
18	-528505696.000000	1.973062
19	-10041608192.000000	36.488184
20	-200832172032.000000	728.763690
21	-4217475694592.000000	15303.037482
22	-92784463708160.000000	336665.824594

The Command Window also displays the following output:

```
Spremenljivka Ie je zapisana v enojni natančnosti
Name      Size      Bytes Class      Attributes

Ie        1x1              4 single

Spremenljivka Id je zapisana v dvojni natančnosti
Name      Size      Bytes Class      Attributes

Id        1x1              8 double
```



## 7.2b Numerična napaka-integral

The image shows the MATLAB 7.6.0 (R2008a) environment. The Command Window displays the results of running a script, showing the value of the integral  $I$  for different values of  $n$ . The Editor window shows the script code, which includes a function `Integral(n)` that calculates the integral and prints the results.

**Command Window Output:**

n	I(enojna natančnost)
6	0.000000
5	0.166667
4	0.233333
3	0.308333
2	0.436111
1	0.718056
0	1.718055

n	I(enojna natančnost)
11	0.000000
10	0.090909
9	0.109091
8	0.123232
7	0.140404
6	0.162915
5	0.193819
4	0.238764
3	0.309691
2	0.436564
1	0.718282
0	1.718282

**Editor Code:**

```
1 function Glavni
2     clc;
3     clear all;
4     Integral(6);
5     Integral(11);
6     end
7
8 function Integral(n)
9     I=0;
10    fprintf('  n      I(enojna natančnost) \r');
11    fprintf('%3i %25.6f \r',n,I);
12    n=n-1;
13    for n=n:-1:0
14        I=(single(I)+1)/(n+1); %enojna natančnost 4 byti
15        fprintf('%3i %25.6f \r',n,I);
16    end
17    disp ' '
18    end
19
20
```

## 7.3 Numerična napaka-zaporedje

Imamo zaporedje  $1, \frac{1}{3}, \frac{1}{9}, \frac{1}{27}, \dots, \frac{1}{3^n} = p_n, n = 0, 1, 2, \dots$

Zapiši zaporedje na tri načine:

-z Matlabovim vektorskim zapisom

-kot zaporedje:  $p_{n+1} = \frac{1}{3} p_n, p_0 = 1, n = 0, 1, 2, \dots$

-kot zaporedje:  $p_{n+1} = p_{n-1} - \frac{8}{3} p_n, p_0 = 1, p_1 = \frac{1}{3}$

Opazuj napako pri rekurzivnem odštevanju.

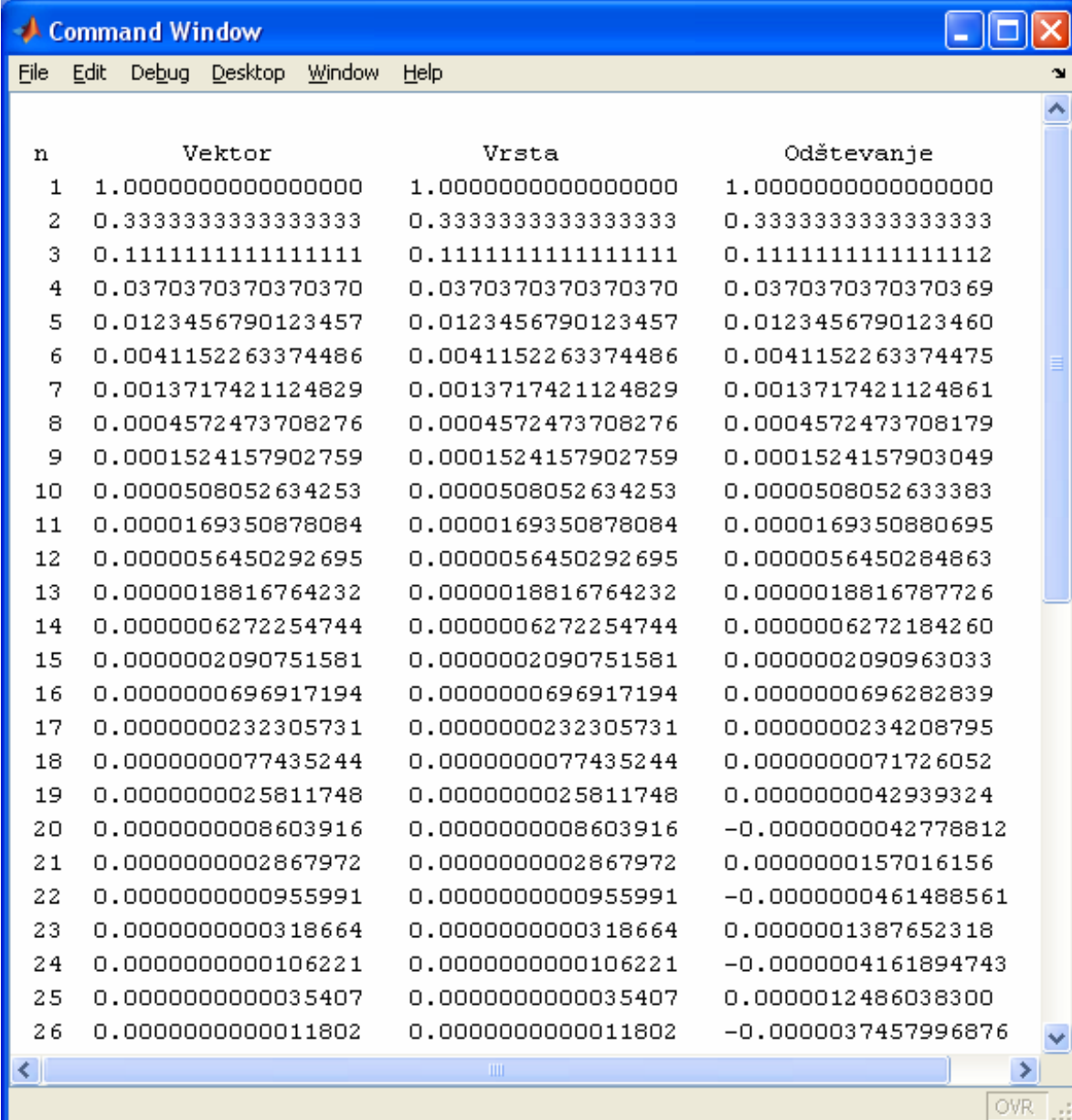
## 7.3 Numerična napaka-zaporedje

```
C:\vaje NM IR\numerične\nm\vaja 8\vaja8_7.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 -   clc;clear all;format long
2 -   N=26;
3 -   p1 = (1/3).^(0:N); %matlab izračun v vektorju
4
5 -   p2(1) = p1(1); %začetne vrednosti
6 -   p2(2) = p1(2);
7 -   p3(1) = p1(1);
8 -   p3(2) = p1(2);
9
10 -  for n = 1:N
11 -      p2(n+1) = (1/3)*p2(n); %izračun zaporedja 1/3^n
12
13 -  end
14 -      disp ' '
15 -  for n = 2:N,
16 -      p3(n+1) = p3(n-1)-8/3*p3(n); % izračun z odštevanjem
17
18 -  end
19 -      disp ' n          Vektor          Vrsta          Odštevanje'
20 -  for i=1:N
21 -      fprintf('%3i %18.16f %18.16f %18.16f\n',i,p1(i),p2(i),p3(i))
22 -  end
23
24 -      plot([p2',p3'])
25
```

script Ln 24 Col 16 OVR

## 7.3 Numerična napaka-zaporedje

Rezultat



n	Vektor	Vrsta	Odštevanje
1	1.0000000000000000	1.0000000000000000	1.0000000000000000
2	0.3333333333333333	0.3333333333333333	0.3333333333333333
3	0.1111111111111111	0.1111111111111111	0.1111111111111112
4	0.0370370370370370	0.0370370370370370	0.0370370370370369
5	0.0123456790123457	0.0123456790123457	0.0123456790123460
6	0.0041152263374486	0.0041152263374486	0.0041152263374475
7	0.0013717421124829	0.0013717421124829	0.0013717421124861
8	0.0004572473708276	0.0004572473708276	0.0004572473708179
9	0.0001524157902759	0.0001524157902759	0.0001524157903049
10	0.0000508052634253	0.0000508052634253	0.0000508052633383
11	0.0000169350878084	0.0000169350878084	0.0000169350880695
12	0.0000056450292695	0.0000056450292695	0.0000056450284863
13	0.0000018816764232	0.0000018816764232	0.0000018816787726
14	0.0000006272254744	0.0000006272254744	0.0000006272184260
15	0.0000002090751581	0.0000002090751581	0.0000002090963033
16	0.0000000696917194	0.0000000696917194	0.0000000696282839
17	0.0000000232305731	0.0000000232305731	0.0000000234208795
18	0.0000000077435244	0.0000000077435244	0.0000000071726052
19	0.0000000025811748	0.0000000025811748	0.0000000042939324
20	0.0000000008603916	0.0000000008603916	-0.0000000042778812
21	0.0000000002867972	0.0000000002867972	0.00000000157016156
22	0.0000000000955991	0.0000000000955991	-0.00000000461488561
23	0.0000000000318664	0.0000000000318664	0.000000001387652318
24	0.0000000000106221	0.0000000000106221	-0.000000004161894743
25	0.0000000000035407	0.0000000000035407	0.00000012486038300
26	0.0000000000011802	0.0000000000011802	-0.00000037457996876