Vjezbe 1. Uvod u Mathematicu

Osnovne operacije

Radna datoteka naziva se notebook. Moze sluziti kao kalkulator :

```
In[1]:=  2 * 3

Out[1]=  6
```

Neka osnovna pravila :
 - naredbe se pokrece s Shift+Enter ili “malo” enter
 - imena funkcija uvijek pocinju velikim slovima, ako ime sadrži vise rijeci svaka pocinje velikim slovom
 - argumenti funkcija se odjeljuju uglatim zagradama [...]; (...) sluze za grupiranje izraza, {...} za liste, [[...]] za indeksiranje elemenata liste
 - komentari se mogu staviti unutar (*...*)

Primjeri:

```
In[2]:=  Pi
         N[Pi, 20]

Out[2]=  π

Out[3]=  3.1415926535897932385
```

```
In[4]:=  2^10
         2^10 (* Ctrl+6 *)
         1 / 5
         1
         — (*Ctrl+/ *)
         5
         (*Za ljepsi unos koristiti Palettes*)

Out[4]=  1024

Out[5]=  1024

         1
Out[6]=  —
         5

         1
Out[7]=  —
         5
```

Alternativni nacini pozivanja funkcije na argumentu :

```
In[8]:=  N[Pi]
         Pi // N
         N@Pi

Out[8]=  3.14159

Out[9]=  3.14159

Out[10]= 3.14159
```

Zaustavjanje ispisa:

```
In[11]:= 4 * 5;
         x = 4 * 5
         x = 4 * 5;

Out[12]= 20
```

Pristupanje prethodnom rezultatu :

```
In[14]:= 3 * 4

Out[14]= 12

In[15]:= % / 2

Out[15]= 6

In[16]:= % / 2

Out[16]= 3

In[17]:= %% / 2

Out[17]= 3

In[18]:= %%%% / 2

Out[18]= 6

In[19]:= Out[14]

Out[19]= 12
```

Help

Help je iznimno opsiran i detaljan - Documentation Center. Najlakse mu je pristupiti s F1.

```
? Plot
```

Plot[f, {x, x_{min}, x_{max}}] generates a plot of f as a function of x from x_{min} to x_{max}.
Plot[{f_1, f_2, …}, {x, x_{min}, x_{max}}] plots several functions f_i. ≫

Izrazi i definicije funkcija

```
In[20]:= Clear[x] (*brise vrijednost varijable x*)

In[21]:= x * (x + 1) * (x + 2)

Out[21]= x (1 + x) (2 + x)

In[22]:= Expand[x * (x + 1) * (x + 2)]
```

Out[22]= $2\,x + 3\,x^2 + x^3$

In[23]:= `Simplify`$\left[2\,x + 3\,x^2 + x^3\right]$

Out[23]= $x\,\left(2 + 3\,x + x^2\right)$

In[24]:= $\texttt{FullSimplify}\left[2\,x + 3\,x^2 + x^3\right]$

```
Out[24]= x (1 + x) (2 + x)
```

Definiranje varijabli i funkcija opcenito moze biti trenutno (=) i odgodeno (:=)

```
In[25]:= a = RandomReal[];
         b := RandomReal[];

In[27]:= Table[a, {5}]
         Table[b, {5}]

Out[27]= {0.312542, 0.312542, 0.312542, 0.312542, 0.312542}

Out[28]= {0.616499, 0.690937, 0.886864, 0.176985, 0.919929}
```

Funkcije je preporucljivo uvijek definirati odgodeno; argumenti se navode s _

```
In[29]:= f[x_] := x^2 + 5

In[30]:= f[2]

Out[30]= 9

In[31]:= g[x_, y_] := x + y + 3;

In[32]:= g[2, 4]

Out[32]= 9
```

Moguce je definirati i viselinijske funkcije, ali za tu svrhu bolje je koristiti module. Moduli omogucavaju i lokalno definiranje varijabli.

```
In[33]:= f[a_, b_, c_] := Module[{aa, bb},
           aa = a + b;
           bb = aa + c]

In[34]:= f[1, 2, 3]

Out[34]= 6
```

Simbolu je moguce dodijeliti vrijednost i s /. + ->

```
In[35]:= x^2 + x^3 /. x → 2

Out[35]= 12
```

Liste

Kreiranje i prikazivanje listi

```
In[36]:= list = {1, 2, 3, 4, 5}

Out[36]= {1, 2, 3, 4, 5}

In[37]:= Range[10]

Out[37]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In[38]:= Range[4, 8]

Out[38]= {4, 5, 6, 7, 8}

In[39]:= Range[0, 30, 3]

Out[39]= {0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30}

In[40]:= Range[0.5, 5.4, 0.75]

Out[40]= {0.5, 1.25, 2., 2.75, 3.5, 4.25, 5.}

In[41]:= Table[2^k, {k, 1, 10, 2}]
         Table[2^k, {k, 1, 10}]
         Table[2^k, {k, 10}]

Out[41]= {2, 8, 32, 128, 512}

Out[42]= {2, 4, 8, 16, 32, 64, 128, 256, 512, 1024}

Out[43]= {2, 4, 8, 16, 32, 64, 128, 256, 512, 1024}

In[44]:= Table[2, {5}]

Out[44]= {2, 2, 2, 2, 2}

In[45]:= Table[2 * i, {i, {5, 2, 7}}]

Out[45]= {10, 4, 14}

In[46]:= Table[{i, j}, {i, 1, 5}, {j, 2, 4}]

Out[46]= {{{1, 2}, {1, 3}, {1, 4}}, {{2, 2}, {2, 3}, {2, 4}},
          {{3, 2}, {3, 3}, {3, 4}}, {{4, 2}, {4, 3}, {4, 4}}, {{5, 2}, {5, 3}, {5, 4}}}

In[47]:= Table[i + j, {i, 1, 5}, {j, 2, 4}] // MatrixForm
         Table[i + j, {i, 1, 5}, {j, 2, 4}] // TableForm

Out[47]//MatrixForm=
          ( 3  4  5 )
          ( 4  5  6 )
          ( 5  6  7 )
          ( 6  7  8 )
          ( 7  8  9 )

Out[48]//TableForm=
         3     4     5
         4     5     6
         5     6     7
         6     7     8
         7     8     9
```

Operacije na listama

```
In[49]:= x = Range[10]

Out[49]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In[50]:= x[[4]]
         x[[{2, 3}]]
         x[[{2, 3, 8}]]
         x[[3 ;; 8]]

Out[50]= 4

Out[51]= {2, 3}

Out[52]= {2, 3, 8}

Out[53]= {3, 4, 5, 6, 7, 8}
```

```
In[54]:= Length[x]

Out[54]= 10
```

```
In[55]:= y = {{1, 2}, {3, 4}, {5, 6}};
         Dimensions[y]

Out[56]= {3, 2}
```

```
In[57]:= y[[2]]
         y[[3, 2]]
         y[[3]][[2]]

Out[57]= {3, 4}

Out[58]= 6

Out[59]= 6
```

```
In[60]:= x = Range[1, 10, 2]
         y = Range[5, 14, 2]
         Union[x, y]
         Join[x, y]
         Intersection[x, y]
         Min[x]
         Sort[x, Greater]

Out[60]= {1, 3, 5, 7, 9}

Out[61]= {5, 7, 9, 11, 13}

Out[62]= {1, 3, 5, 7, 9, 11, 13}

Out[63]= {1, 3, 5, 7, 9, 5, 7, 9, 11, 13}

Out[64]= {5, 7, 9}

Out[65]= 1

Out[66]= {9, 7, 5, 3, 1}
```

Vektorske i matricne operacije

```
In[67]:= vek = {2, 2};
         mat = {{1, 2}, {3, 4}};
```

```
In[69]:=  vek - 2 vek

Out[69]=  {-2, -2}

In[70]:=  mat.vek

Out[70]=  {6, 14}

In[71]:=  vek.vek

Out[71]=  8

In[72]:=  MatrixPower[mat, 4]
          Det[mat]
          Transpose[mat] // MatrixForm

Out[72]=  {{199, 290}, {435, 634}}

Out[73]=  -2

Out[74]//MatrixForm=
          ( 1  3 )
          ( 2  4 )
```

Logicki operatori i petlje

```
In[75]:=  x = 2;
          y = 3;

In[77]:=  x < y
          x == y
          x ≠ y
          x < y && 4 < 5

Out[77]=  True

Out[78]=  False

Out[79]=  True

Out[80]=  True

In[81]:=  If[x > y, 1, 0]

Out[81]=  0

In[82]:=  If[x > y, x = x + 2;, y = y + 2;]
          x
          y

Out[83]=  2

Out[84]=  5

In[85]:=  Do[x = x + 1, {20}]
          x

Out[86]=  22

In[87]:=  Clear[i]
```

```
In[88]:= For[i = 1, i ≤ 5, i++, Print[2 * i]]
         2
         4
         6
         8
         10

In[89]:= Clear[x];
         g[x_] := x^2;
         Nest[g, x, 5]
         NestList[g, x, 5]
```

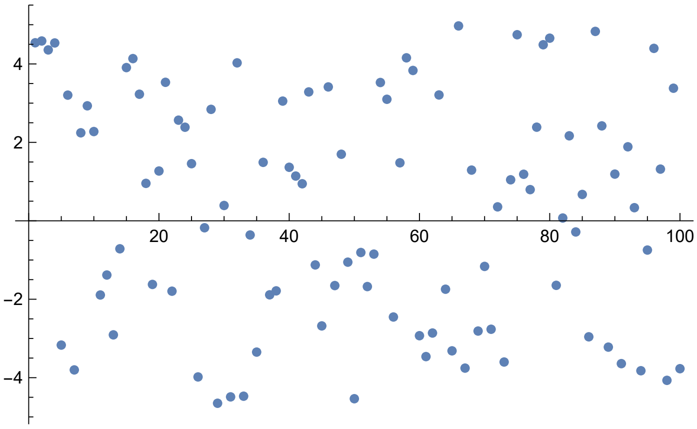
Out[91]= x^{32}

Out[92]= $\{x, x^2, x^4, x^8, x^{16}, x^{32}\}$

Grafika

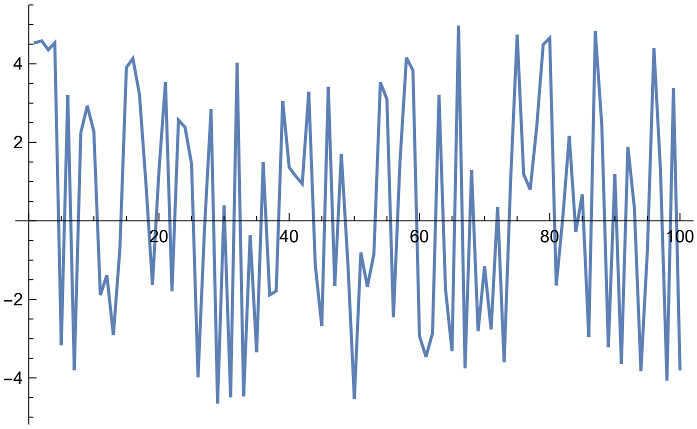
```
In[93]:= data = RandomReal[{-5, 5}, 100];

In[94]:= ListPlot[data]
```

Out[94]=



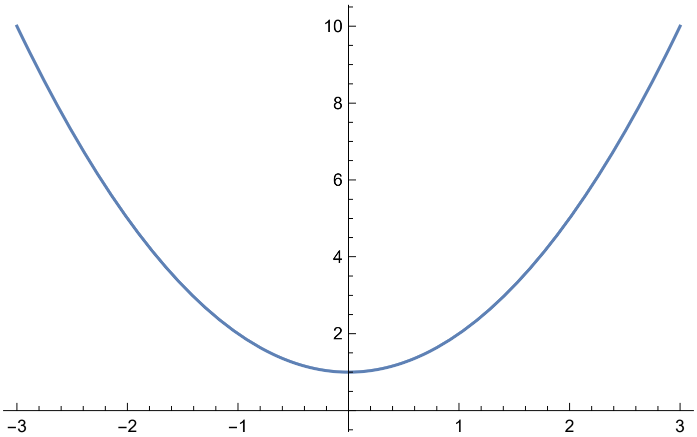
```
In[95]:= ListLinePlot[data]
```

Out[95]=



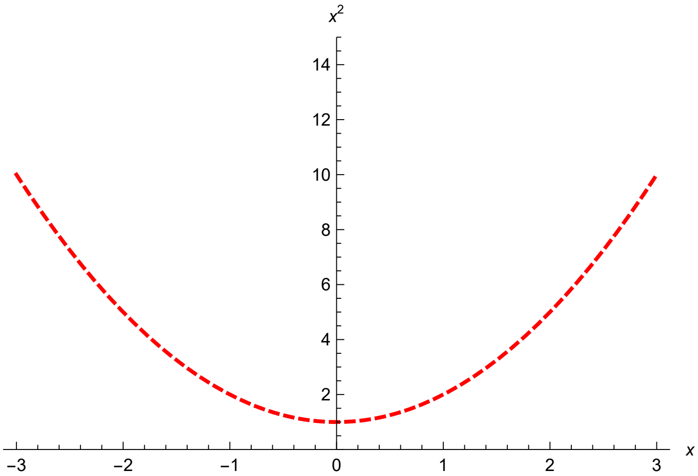
In[96]:=
```
f[x_] := x^2 + 1;
Plot[f[x], {x, -3, 3}]
```

Out[97]=



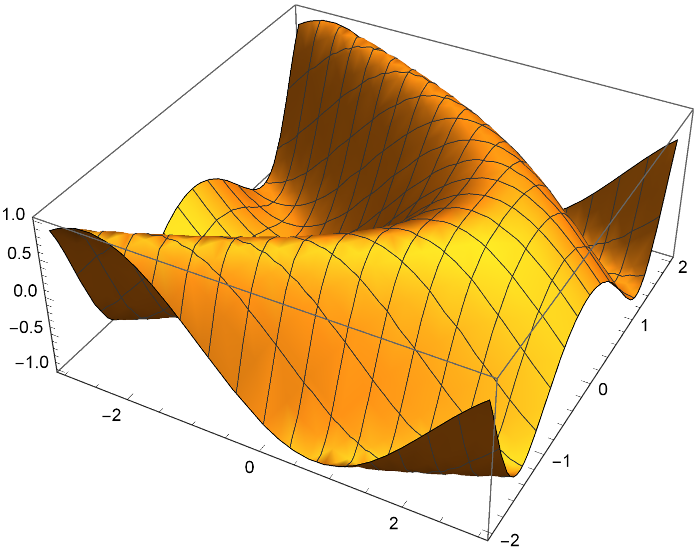
In[98]:=
```
Clear[x];
Plot[f[x], {x, -3, 3}, AxesLabel → {x, x^2}, AxesOrigin → {0, 0},
 PlotRange → {0, 15}, PlotStyle → {Red, Thick, Dashed}]
```

Out[99]=



In[100]:=
```
Plot3D[Sin[x + y^2], {x, -3, 3}, {y, -2, 2}]
```

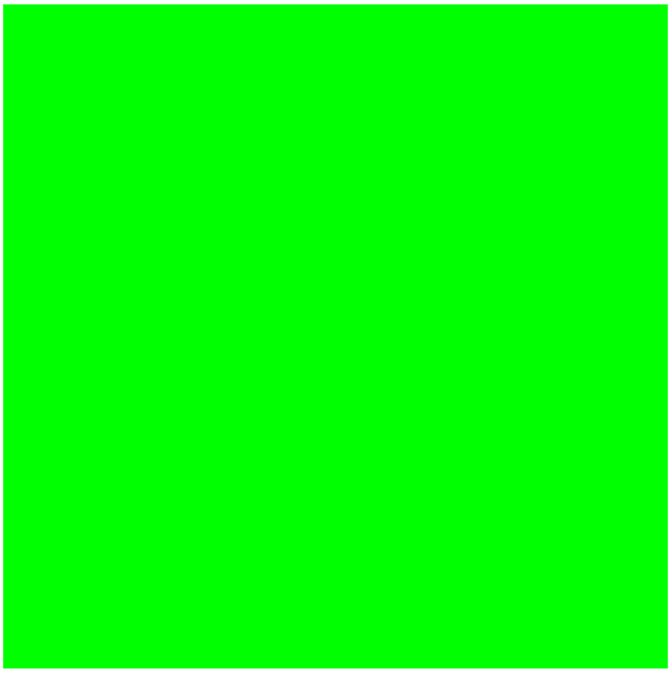
Out[100]=



Sve ove funkcije koriste tzv. primitvne oblike - točke, linije ...

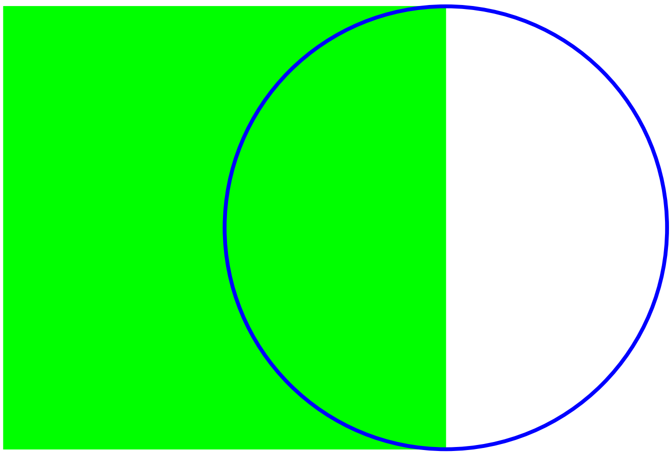
```
In[101]:= Graphics[{Thick, Green, Rectangle[{0, -1}, {2, 1}]}]
```

Out[101]=



```
In[102]:= Graphics[{Thick, Green, Rectangle[{0, -1}, {2, 1}], Blue, Circle[{2, 0}]}]
```

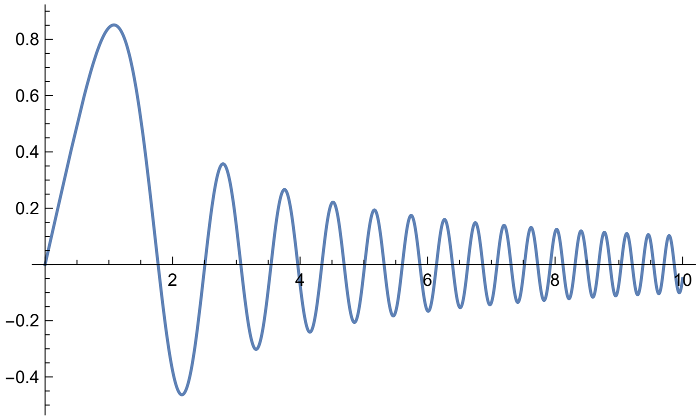
Out[102]=



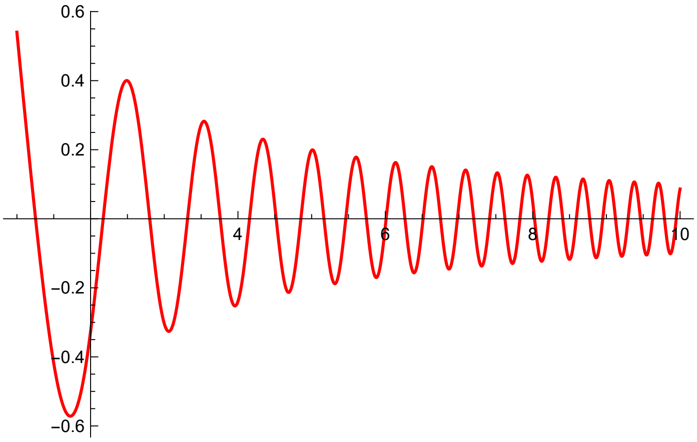
Nacrtati nekoliko grafova na jednoj slici najlakse je funkcijom Show

```
In[103]:= sl1 = Plot[Sin[x^2] / x, {x, 0, 10}]
          sl2 = Plot[Cos[x^2] / x, {x, 1, 10}, PlotStyle → Red]
          Show[{sl1, sl2}, Axes → False]
```

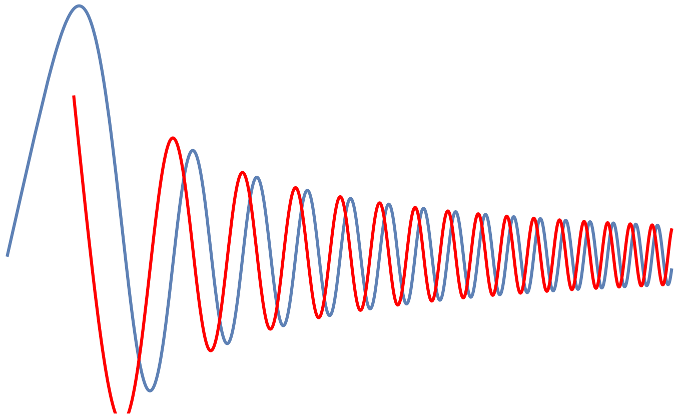
Out[103]=



Out[104]=



Out[105]=



Integracija s Wolfram Alpha

Od verzije 8 pretraživanje po Wolfram Alpha je integrirano u Mathematicu. Jednostavno se poziva s dva znaka = i ključnim riječima nakon toga.
U sljedećem primjeru klikom možemo doći do povijesnih podataka, a zatim na + i FComputable data dobiti listu za manipulaciju u Mathematici.

In[106]:= osijek weather

Input interpretation:

weather	Osijek, Croatia

Latest recorded weather for Osijek, Croatia: Show non–metric More

temperature	13 °C
relative humidity	72% (dew point: 8 °C)
wind speed	3.6 m/s

(32 minutes ago)

+ Units

Weather forecast for Osijek, Croatia: Show non–metric More days More details

Today:

between 10 °C and 18 °C

clear (all day) | rain (mid–afternoon onward)

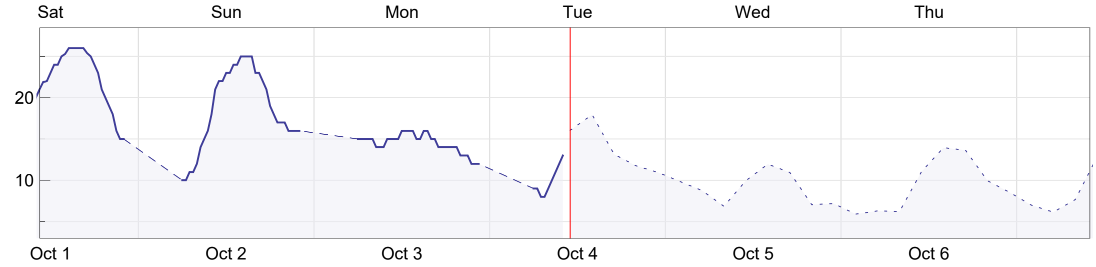
Tonight:

between 8 °C and 13 °C

rain (all night) | clear (evening to late night) | cloudy (late night to early morning) | partly cloudy (early morning onward)

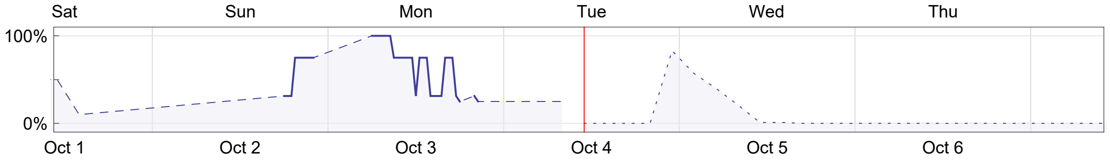
Weather history & forecast: Current week | Show non–metric More

Temperature:



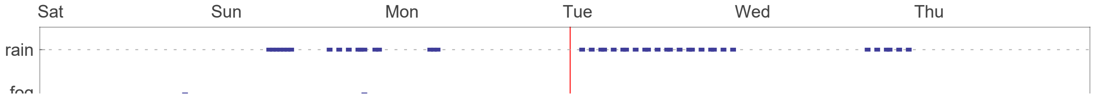
low: 6 °C Thu, Oct 6, 2:00am	average high: 17 °C average low: 9 °C	high: 26 °C Sat, Oct 1, 2:30pm, …

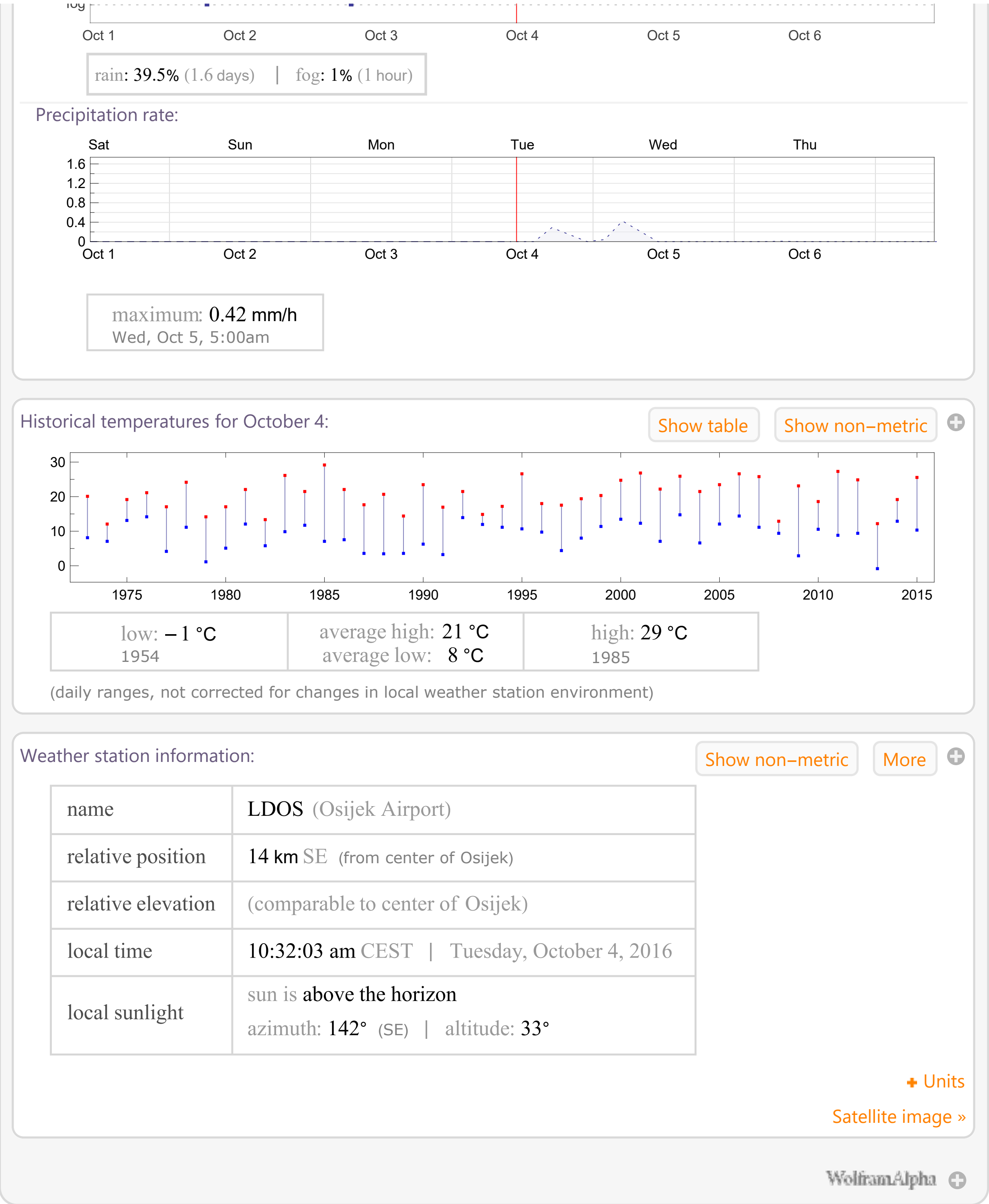
Cloud cover:



clear: 53.2% (2.1 days) | overcast: 7% (6.8 hours)

Conditions:





rain: 39.5% (1.6 days)	fog: 1% (1 hour)

Precipitation rate:

maximum: 0.42 mm/h
Wed, Oct 5, 5:00am

Historical temperatures for October 4:

low: – 1 °C 1954	average high: 21 °C average low: 8 °C	high: 29 °C 1985

(daily ranges, not corrected for changes in local weather station environment)

Weather station information:

name	LDOS (Osijek Airport)
relative position	14 km SE (from center of Osijek)
relative elevation	(comparable to center of Osijek)
local time	10:32:03 am CEST | Tuesday, October 4, 2016
local sunlight	sun is above the horizon azimuth: 142° (SE) | altitude: 33°

+ Units

Satellite image »

```
In[108]:= WolframAlpha["osijek weather",
           {{"WeatherCharts:WeatherData", 1}, "ComputableData"},
           PodStates → {"WeatherCharts:WeatherData__Past 5 years"}];
```

Osim toga, naredbu u Mathematici možemo pokušati upisati i slobodnim jezikom ako prije toga napišemo =.

In[109]:= **sum integers from 1 to 100**

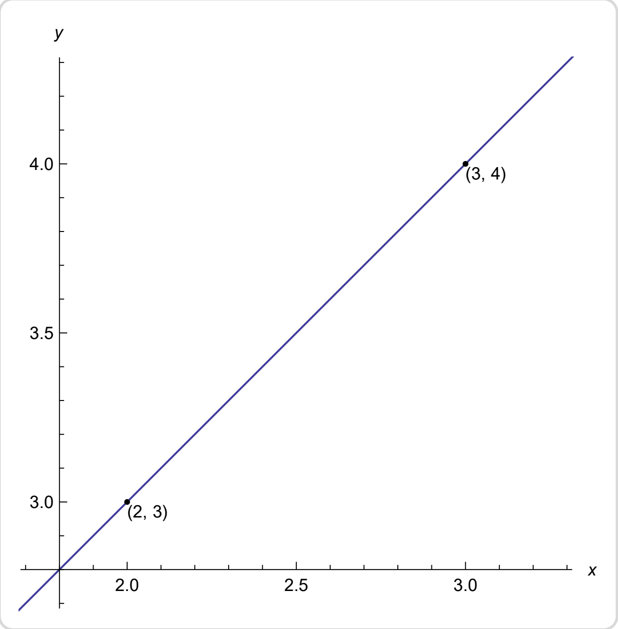
```
Total[Range[1, 100]]
```

Out[109]= 5050

In[110]:= **line through (2,3), (3,4)** »

↳ Visual representation

Out[110]=



In[111]:= **everything should be made simple**

↳ Result

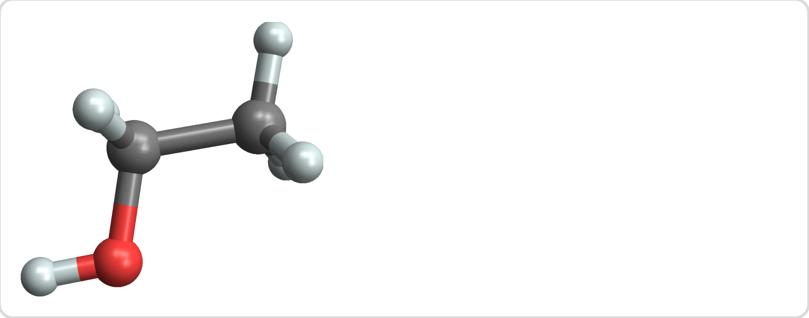
Out[111]= ...But not simpler.

(according to Albert Einstein)

In[112]:= **alcohol 3d structure** »

↳ Result

Out[112]=



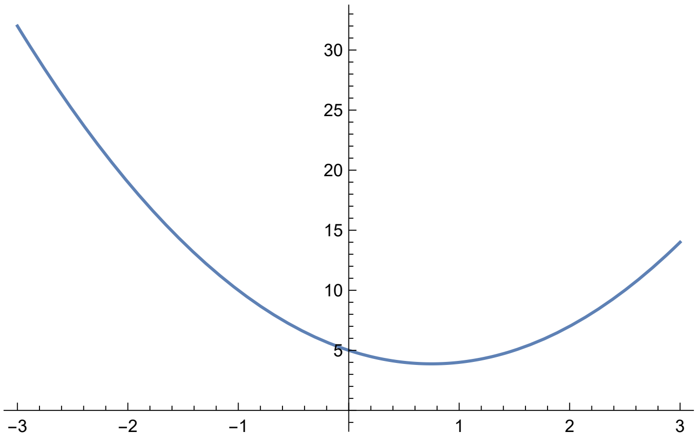
Optimizacijske procedure

U Mathematici postoji niz procedura za optimizaciju, lokalnu i globalnu, koje mogu biti simboličke i numeričke.

Simbolička optimizacija

```
In[113]:= f[x_] := 2 x^2 - 3 x + 5
          Plot[f[x], {x, -3, 3}, AxesOrigin → {0, 0}]
```

Out[114]=



```
In[115]:= Minimize[f[x], x]
          Minimize[{f[x], x ≤ 0}, x]
```

Out[115]= $\left\{\frac{31}{8}, \left\{x \to \frac{3}{4}\right\}\right\}$

```
Out[116]= {5, {x → 0}}
```

```
In[117]:= D[f[x], x]
          Solve[D[f[x], x] == 0, x]
```

```
Out[117]= - 3 + 4 x
```

Out[118]= $\left\{\left\{x \to \frac{3}{4}\right\}\right\}$

```
In[126]:= Clear[a, b, c, x, y]
          Minimize[a x^2 + b x + c, x]
```

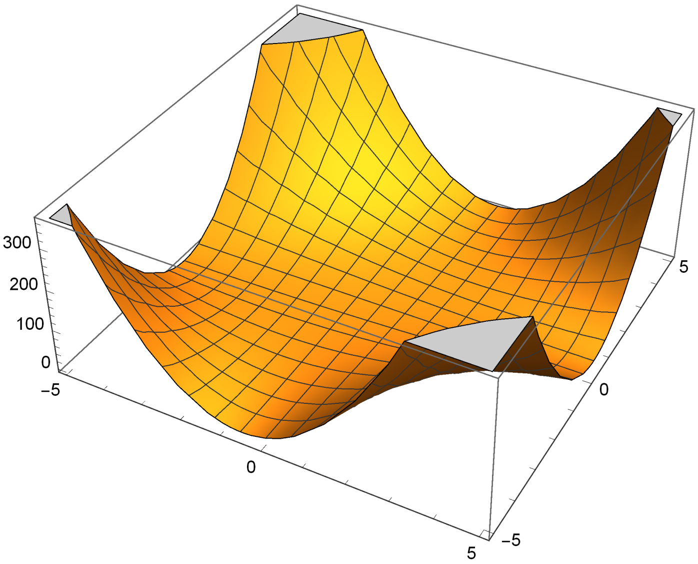
Out[127]=

$$\left\{\begin{cases} c & (b == 0\ \&\&\ a == 0)\ ||\ (b == 0\ \&\&\ a > 0) \\ \frac{-b^2+4\,a\,c}{4\,a} & (b > 0\ \&\&\ a > 0)\ ||\ (b < 0\ \&\&\ a > 0) \\ -\infty & \text{True} \end{cases},\right.$$

$$\left.\left\{x \to \begin{cases} -\frac{b}{2\,a} & (b > 0\ \&\&\ a > 0)\ ||\ (b < 0\ \&\&\ a > 0) \\ \text{Indeterminate} & (b == 0\ \&\&\ a < 0)\ ||\ (b > 0\ \&\&\ a \le 0)\ ||\ (b < 0\ \&\&\ a \le 0) \\ 0 & \text{True} \end{cases}\right\}\right\}$$

```
In[128]:= f[x_, y_] := (x y - 3) ^2 + 1;
          slfun = Plot3D[f[x, y], {x, -5, 5}, {y, -5, 5}]
          min = Minimize[f[x, y], {x, y}]
          (*pristupanje vrijednostima:*)
          {x, y} /. min[[2]]
```

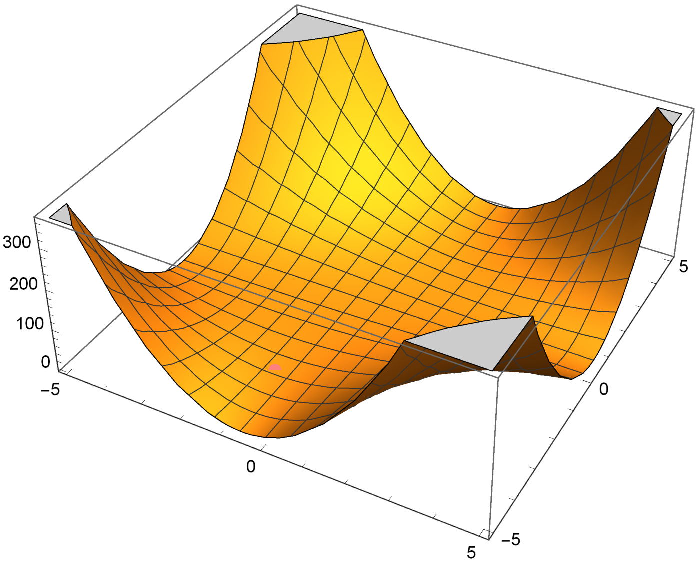
Out[129]=



```
Out[130]= {1, {x → -1, y → -3}}

Out[131]= {-1, -3}
```

```
In[132]:= sltocka = ListPointPlot3D[
              {{x, y, min[[1]]} /. min[[2]]}, PlotStyle → {PointSize[Large], Pink}];
          Show[{slfun, sltocka}]
```

Out[133]=

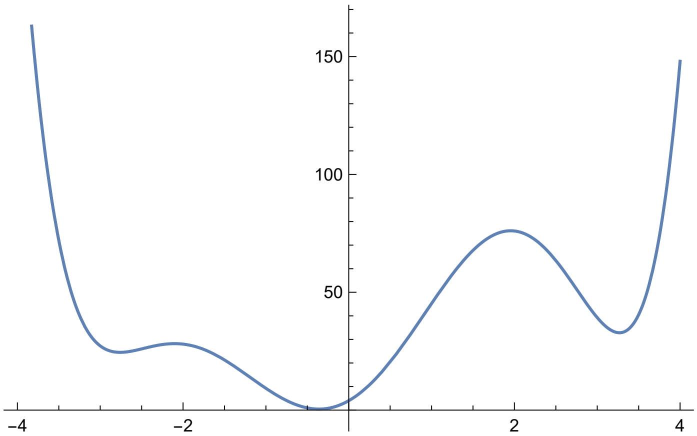


Numerička optimizacija

NMinimize uvijek pokušava naći globalni minimum, dok FindMinimum traži lokalni minimum iz zadane početne točke ili je automatski bira.

```
In[134]:= f[x_] := (0.5 x^3 - 5 x - 2) ^2 + 3 x^2
          Plot[f[x], {x, -4, 4}]
```

Out[135]=

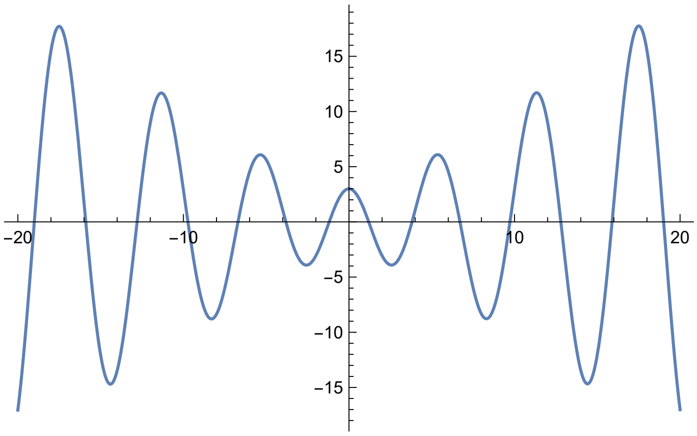


```
In[136]:= Minimize[f[x], x]

Out[136]= {0.438674, {x → -0.359743}}

In[138]:= NMinimize[f[x], x, Method → {"NelderMead"}]
          NMinimize[f[x], x, Method → {"SimulatedAnnealing"}]
          NMinimize[f[x], x, Method → {"DifferentialEvolution"}]
          NMinimize[f[x], x, Method → {"RandomSearch"}]

Out[138]= {0.438674, {x → -0.359743}}

Out[139]= {0.438674, {x → -0.359743}}

Out[140]= {0.438674, {x → -0.359743}}

Out[141]= {0.438674, {x → -0.359743}}

In[142]:= FindMinimum[f[x], x]

Out[142]= {0.438674, {x → -0.359743}}

In[143]:= FindMinimum[f[x], {x, 2}]

Out[143]= {32.8301, {x → 3.26795}}
```

Nekad NMinimize neće naći globalni minimum

```
In[399]:= f[x_] := -x Sin[x] + 3 Cos[x] + x / 1000
          Plot[f[x], {x, -20, 20}]
          NMinimize[f[x], x]
```

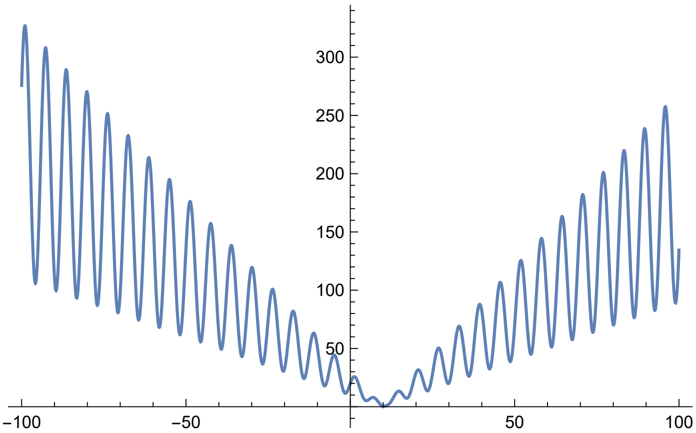
Out[400]=



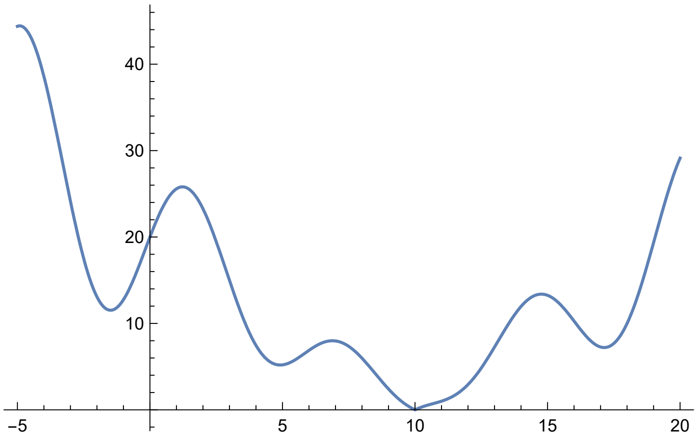
```
Out[401]= {-3.91085, {x → 2.57025}}
```

```
In[406]:= f[x_] := Abs[2 (x - 10) + (x - 10) * Sin[x]];
          Plot[f[x], {x, -100, 100}]
          Plot[f[x], {x, -5, 20}]
          NMinimize[f[x], x]
```

Out[407]=



Out[408]=



```
Out[409]= {11.5272, {x → -1.48327}}
```