

2016

2. ŠTUDENSKA KONFERENCA RAČUNALNIŠKA ORODJA V MATEMATIKI

ZBORNİK POVZETKOV IN PROGRAM

2. ŠTUDENSKA KONFERENCA RAČUNALNIŠKA ORODJA V
MATEMATIKI

Ljubljana, 2016

2. Študentska konferenca Računalniška orodja v matematiki (ROM)

Ljubljana, 15. in 16. februar 2016

Naslov:

Zbornik povzetkov

Uredil: Matija Lokar

Oblikovanje: Uroš Vaupotič

Izdala:

Fakulteta za matematiko in fiziko, Univerza v Ljubljani

Ljubljana, februar 2016

Predgovor

Pri reševanju številnih matematičnih problemov si lahko pomagamo z različnimi računalniškimi orodji. Ta nam lahko pomagajo pri vizualizaciji matematičnih objektov, pri raziskovanju njihovih značilnosti, omogočajo hitrejše in preprostejše pregledovanje določenih domnev ...

Zato je za matematika dobro, da je seznanjen z možnostmi, ki jih prinašajo tovrstni programi. Teh je na voljo res veliko. Nekateri so bolj primerni za eno vrsto nalog, drugi spet za drugo. Vsaj osnovni pregled nad številnimi različicami tovrstnih programov je zato zelo zaželen.

V sklopu predmeta Računalniška orodja v matematiki na visokošolskem študiju Praktična matematika Fakultete za matematiko in fiziko, Univerza v Ljubljani, spoznavamo razna orodja, ki jih potem študenti uporabljajo pri drugih predmetih tekom študija. Ker pa je zelo pomembno, da znajo oceniti primernost določenega orodja za posamezno nalogo, morajo v sklopu predmeta tudi proučiti njim še neznano računalniško orodje in drugim poročati o svojih ugotovitvah.

V tem zborniku so zbrani povzetki predstavitev, ki so jih pripravili študenti.

Matija Lokar

Kazalo

Različni načini dokazovanja Pitagorovega izreka.....	6
Računalniški program Graph	7
Monte Carlo metoda – skupek računalniških algoritmov za reševanje numeričnih problemov z uporabo naključnega izbiranja.....	8
Uporaba GeoGebre, Mathematice, Matlaba pri reševanju nalog na izpitu pri algebri.	9
Pascalov Trikotnik.....	10
Uporaba programa Wolfram Mathematica in programskega jezika Python za generiranje permutacij in podobnih matematičnih objektov.....	11
Uporaba Mathematice pri reševanju diofantskih enačb	12
GraphTea	13
Robocompass - <i>sistem za dinamične geometrijske konstrukcije in transformacije</i>	14
Izračun dolžine slovenske obale s pomočjo programa Wolfram Mathematica	15
Maxima	16
Uporaba GeoGebre in Mathematice pri konstrukciji Fermatove točke.....	17
Teorija Grafov.....	18
Calculator ++	19
MalMath: Step by step solver	20
Geogebra Graphing Calculator	21
Epski krogi	22
Konstrukcija cikloide s pomočjo GeoGebre	23
Reševanje rekurzivnih enačb	24
OEIS, njena uporaba in Sloanova vrzel.....	25
Eulerjeva premica in trilinearni koordinatni sistem	26
Cinderella	27
Cymath.....	28
Geometrija v Mathematici.....	29
Uporaba knjižnice NumPy pri problemih iz Linearne Algebre	30
Desmos	31
Microsoft Mathematics.....	32
Symbolab	33
Rotacije, translacije in skaliranje matematičnih objektov s pomočjo matrik.....	34

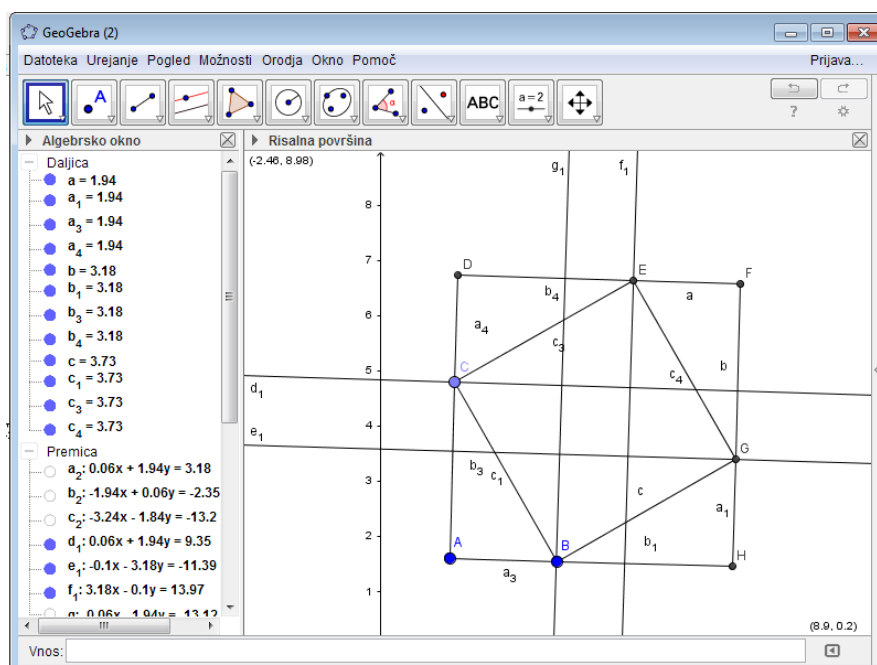
Uporaba GeoGebre in Mathematice pri reševanju matematičnih nalog zapletenih funkcij	35
Racionalne funkcije v programu Maxima.....	36
Preprosta linearna regresija po metodi najmanjših kvadratov z uporabo programa R.....	37
GeoGebraScript.....	38
GEUP 7 - program za pomoč pri geometriji	39
Bertrandov paradoks.....	40

Različni načini dokazovanja Pitagorovega izreka

Alenka Kejžar, Fakulteta za matematiko in fiziko Ljubljana, UL, alenka.kejzar@student.fmf.uni-lj.si

Že v osnovni šoli smo se pri geometriji naučili formulo Pitagorovega izreka: $a^2 + b^2 = c^2$, ki nam pove, da je kvadrat hipotenuze pravokotnega trikotnika enak vsoti kvadratov katet tega trikotnika. S pomočjo te formule smo reševali številne primere tako v osnovni kot srednji šoli, gimnaziji in kasneje tudi na fakulteti vendar nikoli nismo podvomili o pravilnosti le te. Prav zato sem se odločila, da si pogledam nekaj dokazov, s katerimi so matematiki iz različnih koncev sveta in časovnih obdobj dokazovali pravilnost Pitagorovega izreka.

Izmed vseh dokazov Pitagorovega izreka sem jih izbrala sedem in jih s pomočjo programov Wolfram Mathematica in GeoGebra tudi sama preizkusila. Nekaj teh dokazov pa si bomo ogledali tudi na sami predstavitvi.

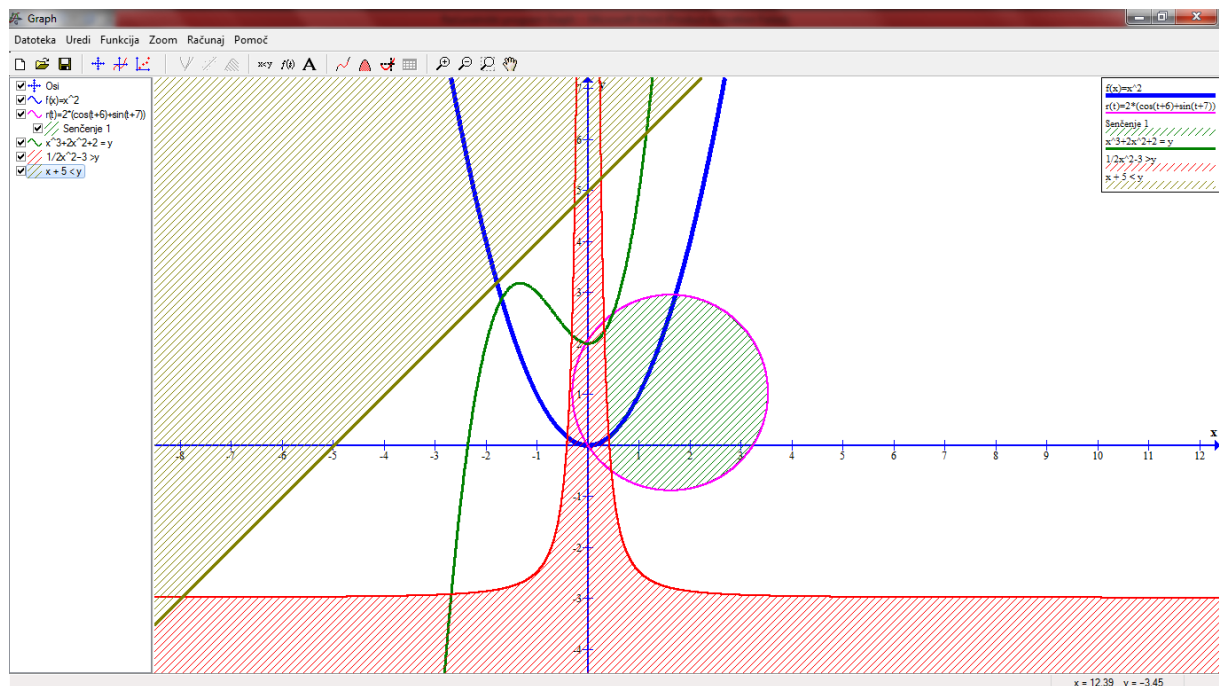


Računalniški program Graph

Keni Šuligoj, Fakulteta za matematiko in fiziko, UL, keni.suligoj@student.fmf.uni-lj.si

Predstavil bom program Graph s katerim lahko rišemo grafe matematičnih funkcij v koordinatnem sistemu. Funkcije so lahko s podane standardno, parametrično ali polarno funkcijo. Program je namenjen slikovni predstavi grafov in boljšemu razumevanju njegovih lastnosti, saj nam omogoča risanje grafov, relacij, točk, tangent,... . Z njegovo pomočjo lahko računamo tudi določene inrtegrale, itd.

Namenjen je vsem, ki na svojem področju dela potrebujejo pomoč pri vizualizaciji grafov. Zaradi svoje preprostosti uporabnik ne potrebuje nobenega predznanja.



Kazalo
Program

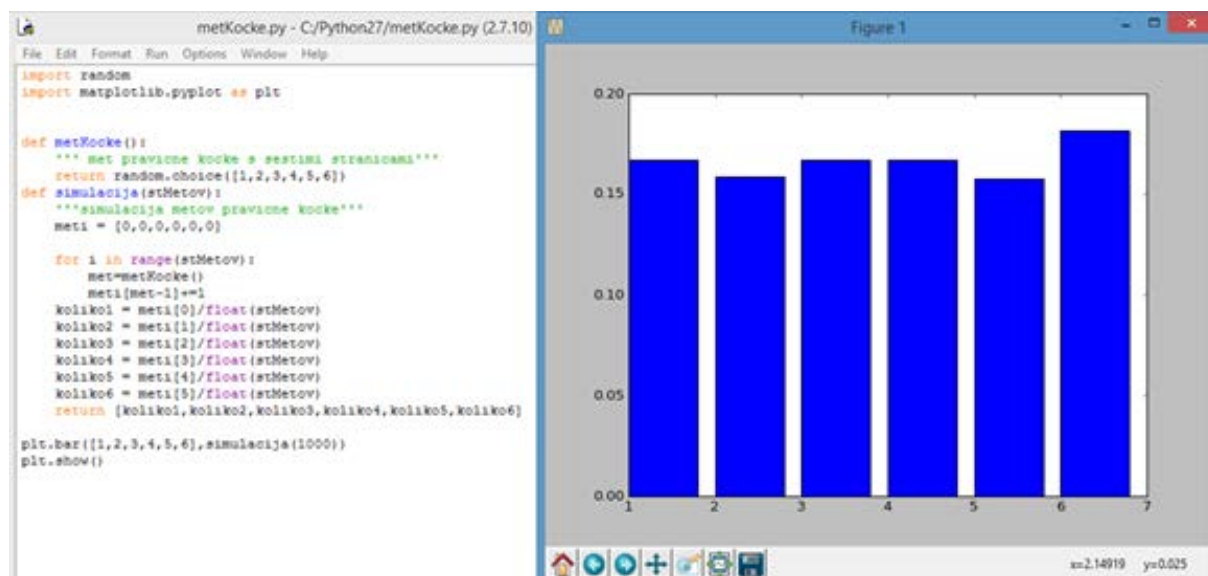
Monte Carlo metoda – skupek računalniških algoritmov za reševanje numeričnih problemov z uporabo naključnega izbiranja

Marko Jereb, Fakulteta za matematiko in fiziko, Univerza Ljubljana,
markojereb@yahoo.com

Monte Carlo metode/simulacije pogosto uporabljamo, kadar z drugimi numeričnimi metodami ne moremo rešiti problema. Pri teh metodah gre dejansko za to, da s pomočjo generiranja velike množice naključnih števil poskusimo priti do rešitve problema. Same metode največkrat razdelimo v tri različne razrede: optimizacija, numerična integracija in verjetnostna porazdelitev.

Na začetku si bomo pogledali predstavitev metode ter nekaj besed o zgodovini. Nato sledijo lažji primeri kjer lahko z MC metodami preverimo verjetnostno porazdelitev metov različne kocke (različne stranice, obtežena kocka...). Nakar sledi predstavitev uporabe na bolj življenskih primerih (finance, fizika) in zaključek na matematičnih problemih (uporaba pri računanju določenih integralov).

Vsi programi bodo napisani s Pythonom(2.7), rezultati pa predstavljeni s pomočjo matplotlib knjižnice.



[Kazalo](#)
[Program](#)

Uporaba GeoGebre, Mathematice, Matlab pri reševanju nalog na izpitu pri algebri.

Martin Češnovar, Fakulteta za matematiko in fiziko, UL,
martin.cesnovar@student.fmf.uni-lj.si

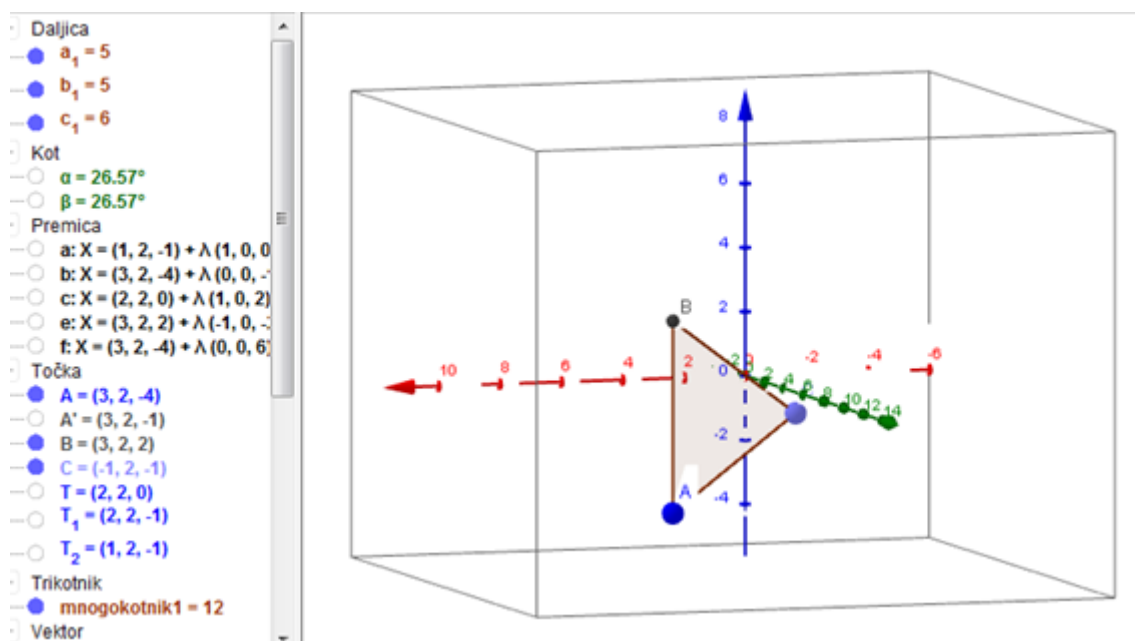
Z znanimi računalniškimi orodji bomo rešili izpit. Ta je dosegljiv na naslovu http://ucilnica1314.fmf.uni-lj.si/pluginfile.php/18474/mod_resource/content/1/resalgi21314.pdf.

Pri nalogi 1 bomo trikotnik narisali s pomočjo Geogebre. Pri tem bomo uporabili risanje v 3d.

Pri nalogi 2 bomo z različnimi orodji rešili problem, ko moramo iz delno znane matrike A in znanih lastnih vektorjev izračunati matrično funkcijo. Pri tem bomo pri programu Matlab spoznali zanke while in for.

Pri nalogi 3 bomo dokazali, da matrika ni normalna, poiskali bomo lastne vrednosti in lastne vektorje in poskušali bomo poiskati skalarni produkt v katerem je matrika normalna.

Pri nalogi 4 bomo poskušali z mathematico dokazati, da velja $\det(M) \cdot \det(V) = \det(A)$.



Kazalo
Program

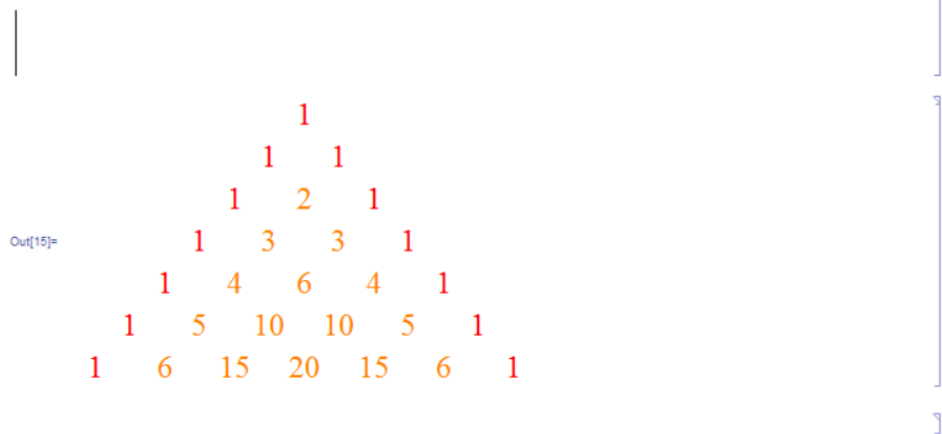
Pascalov Trikotnik

Nikoleta Krstić, Fakulteta za matematiko in fiziko, UL,
Nikoleta.Krstic@student.fmf.uni-lj.si

V tej predstavitvi bomo raziskovali lastnosti že dobro znanega Pascalovega trikotnika. Orodje, s katerim bomo delali, je Mathematica.

Večinoma smo že vsi slišali za Pascalov trikotnik. Imenuje se po francoskem matematiku. Blaise Pascal je bil namreč prvi, ki je v 17. stoletju obrnaval to temo. Vendar nas večina verjetno ne ve, za kaj je Pascalov trikotnik uporaben in koliko zanimivosti je skritih v tem trikotniku. V Pascalovem trikotniku se nahajajo binomska, trikotna, tetraedarska števila in Fibonaccijevo zaporedje. Poleg zgodovine si bomo ogledali, kako v programu Mathematica lahko na več načinov naredimo(narišemo) Pascalov trikotnik in kaj lahko še dodatno spreminjamo.

Pascalov trikotnik



[Kazalo](#)
[Program](#)

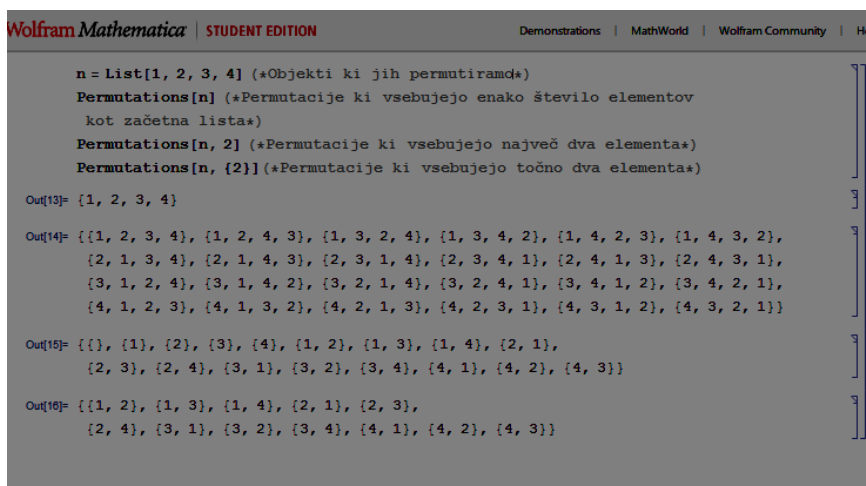
Uporaba programa Wolfram Mathematica in programskega jezika Python za generiranje permutacij in podobnih matematičnih objektov

Rastko Veriš, Fakulteta za matematiko in fiziko, rastko07@gmail.com

Za svojo temo permutacije sem si izbral orodji Wolfram Mathematica in Python.

Svojo predstavitev bom začel tako da bom govoril malo o teoriji seveda. V katerih granah matematike se koncept permutacij pojavi.

V nadaljevanju si bomo določene ukaze, kot so npr. *Permutations*, *Binomial* in drugi ogledali podrobneje. Kot se vidi na sliki spodaj, bom predstavil na kakšne vse načine lahko uporabljamo funkcije *Permutations* v izbranem orodju. V komentarjih na zaslonski sliki se vidi kaj dela vsaka "verzija" tiste funkcije.



```

Wolfram Mathematica | STUDENT EDITION
Demonstrations | MathWorld | Wolfram Community | Help

n = List[1, 2, 3, 4] (*Objekti ki jih permutiram*)
Permutations[n] (*Permutacije ki vsebujejo enako število elementov
kot začetna lista*)
Permutations[n, 2] (*Permutacije ki vsebujejo največ dva elementa*)
Permutations[n, {2}] (*Permutacije ki vsebujejo točno dva elementa*)

Out[13]= {1, 2, 3, 4}

Out[14]= {{1, 2, 3, 4}, {1, 2, 4, 3}, {1, 3, 2, 4}, {1, 3, 4, 2}, {1, 4, 2, 3}, {1, 4, 3, 2},
{2, 1, 3, 4}, {2, 1, 4, 3}, {2, 3, 1, 4}, {2, 3, 4, 1}, {2, 4, 1, 3}, {2, 4, 3, 1},
{3, 1, 2, 4}, {3, 1, 4, 2}, {3, 2, 1, 4}, {3, 2, 4, 1}, {3, 4, 1, 2}, {3, 4, 2, 1},
{4, 1, 2, 3}, {4, 1, 3, 2}, {4, 2, 1, 3}, {4, 2, 3, 1}, {4, 3, 1, 2}, {4, 3, 2, 1}}

Out[15]= {{}, {1}, {2}, {3}, {4}, {1, 2}, {1, 3}, {1, 4}, {2, 1},
{2, 3}, {2, 4}, {3, 1}, {3, 2}, {3, 4}, {4, 1}, {4, 2}, {4, 3}}

Out[16]= {{1, 2}, {1, 3}, {1, 4}, {2, 1}, {2, 3},
{2, 4}, {3, 1}, {3, 2}, {3, 4}, {4, 1}, {4, 2}, {4, 3}}

```

Zaradi svoje hitrosti računanja, mislim da je tisto orodje primerno za obravnavanje ene take teme.

Bom tudi predstavil tiste funkcije v programskem jeziku Python v modulu *itertools* ki vsebuje funkcije kot so *itertools.permutations* in *itertools.combinations*. Če bo časa, bom predstavil svojo funkcijo ki naj bi vrnila vse možne permutacije ene liste.

Svojo predstavitev bom zaključil z zanimivostjo in vprašanjem za publiko.

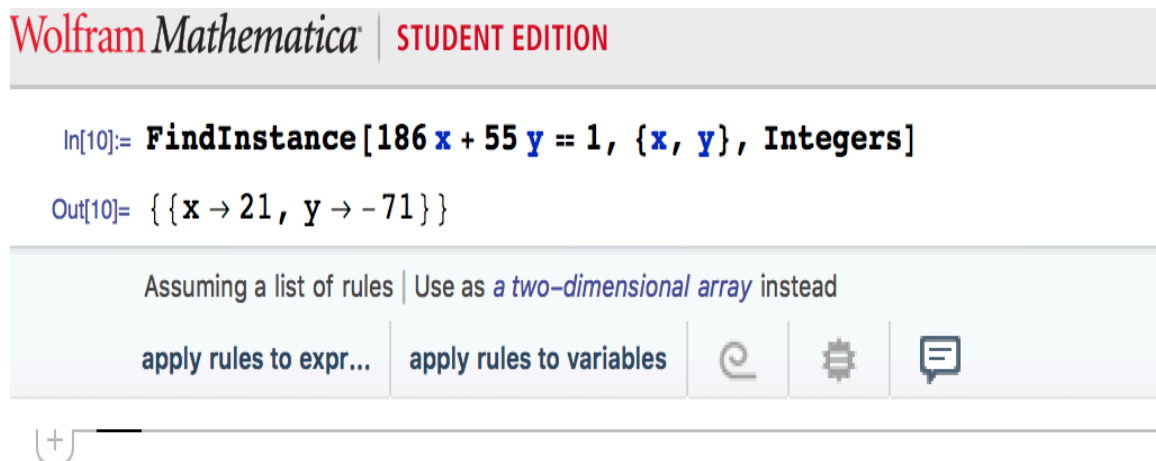
Uporaba Mathematice pri reševanju diofantskih enačb

Samantha Slaček, Fakulteta za matematiko in fiziko, UL,
samantha.slacek@student.fmf.uni-lj.si

O diofantskih enačbah govorimo, kadar moramo poiskati celoštevilске rešitve enačbe s celoštevilskimi koeficienti. S takšnimi problemi se je prvi ukvarjal grški matematik Diofant, ki je živel okoli leta 250 po našem štetju.

Poznamo več tipov diofantskih enačb. S pomočjo Mathematice lahko rešimo vsako izmed njih. Predstavila bom samo nekatere od teh.

Predvsem bom govorila, kako se v Mathematici sploh lotimo reševanja diofantskih enačb. Predstavila bom, katere ukaze vse potrebujemo ter te ukaze podrobno opisala. Pokazala bom tudi nekaj rešenih zgledov diofantskih enačb.



Wolfram Mathematica | STUDENT EDITION

```
In[10]:= FindInstance[186 x + 55 y == 1, {x, y}, Integers]
```

```
Out[10]= {{x -> 21, y -> -71}}
```

Assuming a list of rules | Use as a *two-dimensional array* instead

apply rules to expr... | apply rules to variables | [undo] | [settings] | [help]

[add new input]

[Kazalo](#)
[Program](#)

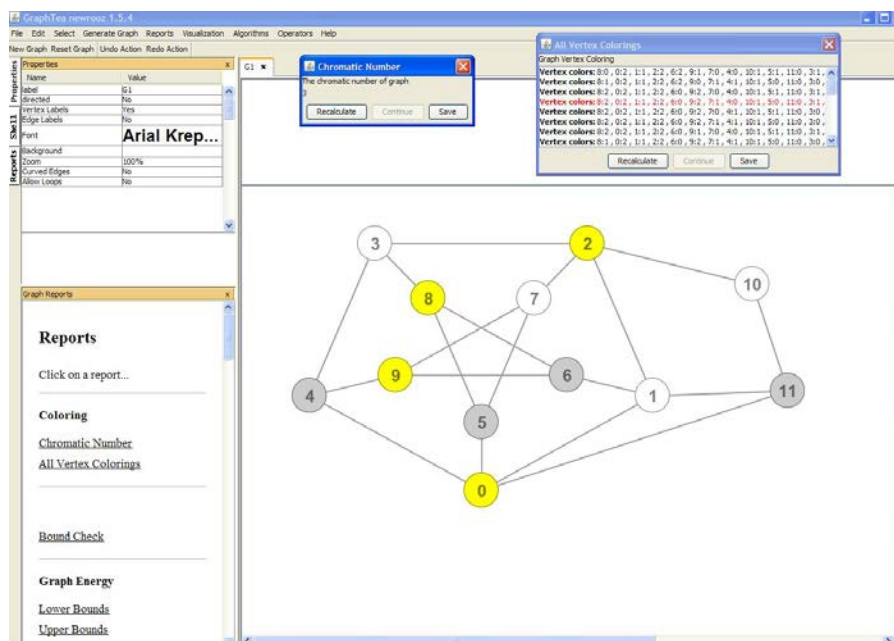
GraphTea

Anej Jereb, Fakulteta za matematiko in fiziko, UL,
 anej.jereb@student.fmf.uni-lj.si

Opisal bom program za računalnik GraphTea, dostopen na <http://www.graphtheorysoftware.com/>. Program je namenjen poučevanju, učenju in raziskovanju teorije grafov.

Program omogoča risanje grafov na tri načine. Preprosto z uporabo miške, z orodjem "Generate Graph", ki ustvari graf glede na parametre, ki mu jih podamo ter tako da graf naložimo iz neke datoteke, recimo narejene v MATLAB-u. Izgled grafa lahko spreminjamo po naših željah z obilico orodij in ga shranimo preprosto kot sliko ali v drugih formatih za uporabo v programih kot sta MATLAB in LaTeX. Graf analiziramo s pomočjo mnogih poročil, ki nam jih program lahko naredi. Na primer koliko je kromatično število, vsa možna barvanja, število trikotnikov, ali je graf Eulerjev itd. Omogočeno je tudi izvajanje algoritmov na grafih. Te nam lahko pokaže korak za korakom in ga lahko kadarkoli začasno prekinemo, tako da lažje spremljamo kako delujejo. To je zelo uporabno za učenje algoritmov, ki se jih uporablja v teoriji grafov.

V predstavitvi bom prikazal osnove uporabe programa in kako z njim rešiti probleme, na katere bodo študentje naleteli tekom študija Praktične matematike.



Kazalo
Program

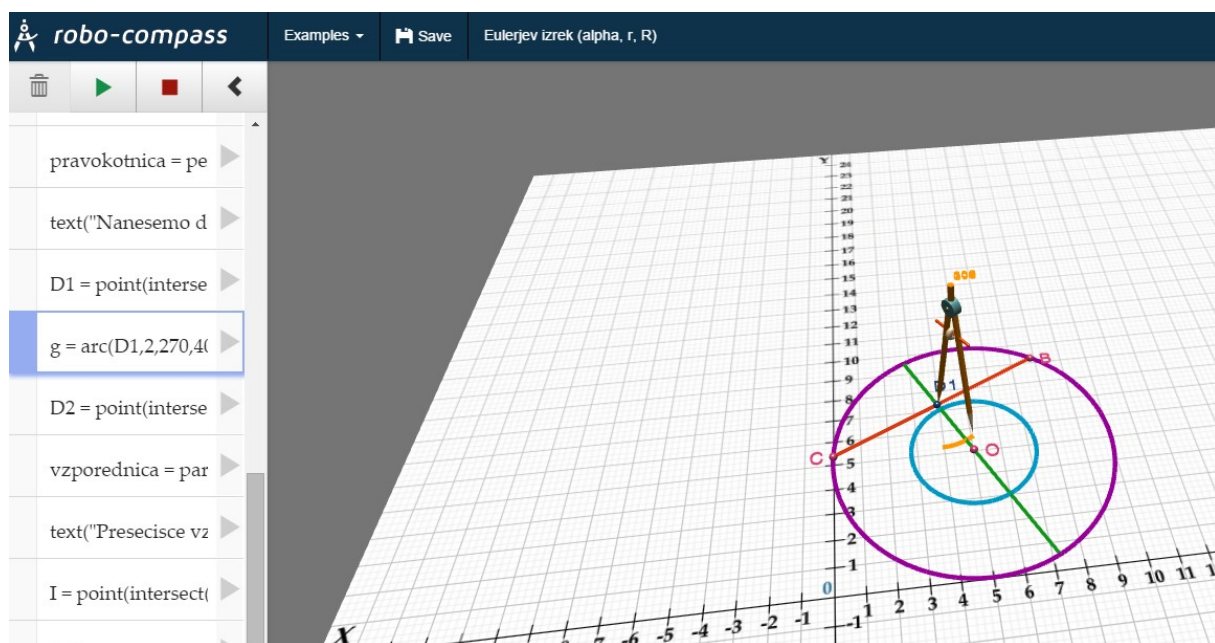
Robocompass - sistem za dinamične geometrijske konstrukcije in transformacije

Anja Trop, Fakulteta za matematiko in fiziko, UL, anja.trop@student.fmf.uni-lj.si

Robocompass je sistem za dinamično geometrijo in izvajanje transformacij nad geometrijskimi liki. Namenjen je predvsem učenju geometrije. Vgrajena ima orodja, ki posnemajo uporabo ravnila, šestila in kotomera, ki jih uporabljamo pri ročno izvedenih konstrukcijah. Aplikacija je brezplačno dostopna v spletni trgovini Crome. Simulira korake geometrijske konstrukcije, ki jih vnesemo s pomočjo vgrajenih ukazov. Je odličen interaktivni pripomoček zlasti za učence in dijake, ki se šele spoznavajo z osnovami geometrije. Poleg predvajanja korakov konstrukcije lahko dodamo še lasten opis postopka, ki se nam prikaže v pasici zgoraj. Tako nadgradimo kvaliteto predstavitve in omogočimo večjo razumljivost, kar je predvsem uporabno s strani pedagogov. Robocompass je tako tudi odličen pripomoček učiteljem, ki lahko z zanimivimi predstavitvami spodbudijo učence k dodatnemu odkrivanju in eksperimentiranju.

Aplikacija z registracijo uporabnika omogoča enostavno arhiviranje in dostopanje do datotek v oblaku ter generiranje spletnih povezav do delovnih listov. Tako širši krog ljudi na enostaven način dostopa do gradiva.

Na predavanju se bomo najprej seznanili z aplikacijo in zmožnostmi, nato si bomo ogledali nekaj vgrajenih ukazov, zaključili pa bomo s prikazom postopka konstrukcije trikotnika z uporabo Eulerjevega izreka.



[Kazalo](#)
[Program](#)

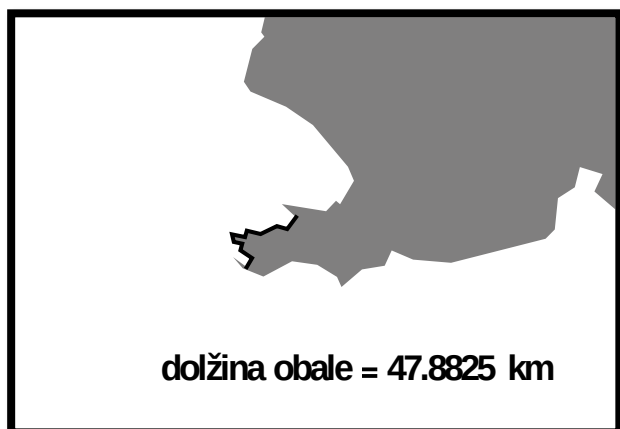
Izračun dolžine slovenske obale s pomočjo programa Wolfram Mathematica

Blaž Poljanec, Fakulteta za matematiko in fiziko, UL, blaz.poljanec@student.fmf.uni-lj.si

Zanima nas, kako dolga je slovenska obala. Pri tem nam bo v pomoč paradoks Benoita Mandelbrota (matematik; 1924-2010). V svojem članku iz leta 1967 'Kako dolga je angleška obala?' se ukvarja s tem, da dolžina obale ni točno določena oziroma jo je težko določiti. Pri tem vprašanju si je pomagal s fraktali (posameznimi manjšimi deli celote). Za preučevanje je vzel angleško obalo. Ker pa linija obale ni pravilnih, ravnih oblik, je dolžina odvisna od velikosti izbranih fraktalov. Prišel je do zaključka, da manjši kot so fraktali, bolj se prilegajo obali in posledično bolj natančno izmerijo njeno dolžino.

Nas zanima slovenska obala. Dolžino slednje bomo izračunali z Wolfram Mathematico. Gre za matematični program, ki nam prihrani ogromno časa pri samem računanju. Povezan je tudi z internetnim programom Wolfram Alpha, ki združuje znanja iz mnogih področij, ne zgolj iz matematike. Ta dva programa uporabnikom po celem svetu nudita veliko možnosti za ustvarjanje raznih prikazov. Nekateri so dostopni tudi na spletnem mestu Wolfram Demonstrations Project. Tam se nahaja tudi naš zgled, "How Long Is The Coast Of Britain" (dostopen na: <http://demonstrations.wolfram.com/HowLongIsTheCoastOfBritain/>).

Na predavanju si bomo ogledali, kako v Wolfram Mathematici s pomočjo ukaza Manipulate in drugih, z njim povezanih ukazov, čimbolj natančno prikazati dolžino slovenske obale po prej omenjenem paradoksu in zgledu.



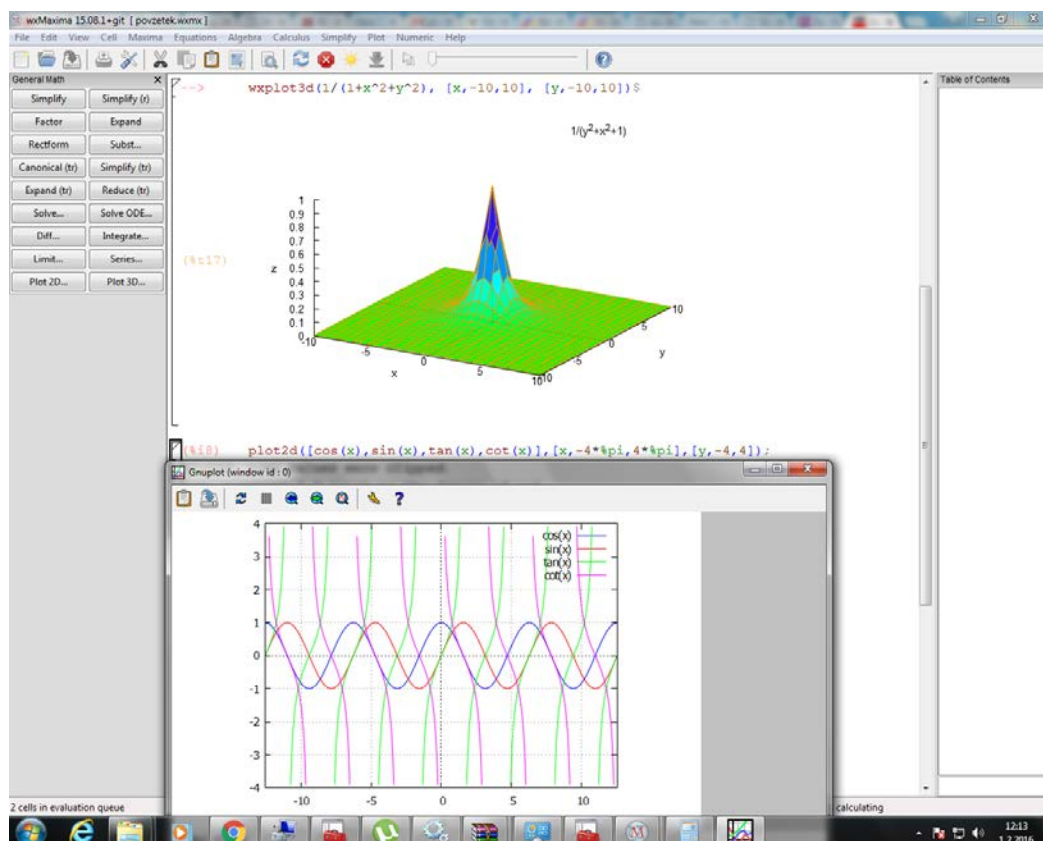
Maxima

Borna Kavčič, Fakulteta za matematiko in fiziko, UL, Borna.Kavcic@student.fmf.uni-lj.si

Orodje Maxima je orodje, s katerim lahko rešujemo veliko matematičnih problemov, namenjen je za reševanje tako nalog iz matematike kot tudi iz linearne algebre, orodje Maxima je zelo natančno, kar se tiče računanja. Maxima je zelo vsestransko orodje in lahko z njim izračunamo marsikaj kot npr. integrale, limite, vsote vrst, faktorizacija na praštevila, deljenje polinomov, inverze ter determinante matrik itd., lahko tudi rišemo grafe in sicer tako v 2D kot tudi v 3D obliki.

Opisoval bom ukaze, ki so na voljo v Maximi, torej kar mi študentje prvega letnika že poznamo, zraven bodo tudi zgledi, ter opisani postopki reševanja določenih nalog. Narisanih bo tudi nekaj grafov funkcij v 2D in 3D obliki.

Namen predstavitve bo boljše spoznanje orodja Maxima s predstavitvijo ukazov ter nekaj zgledi reševanja nalog, ter da je Maxima lahko dobra alternativa npr. Wolfram Mathematici ali pa Geogebri.



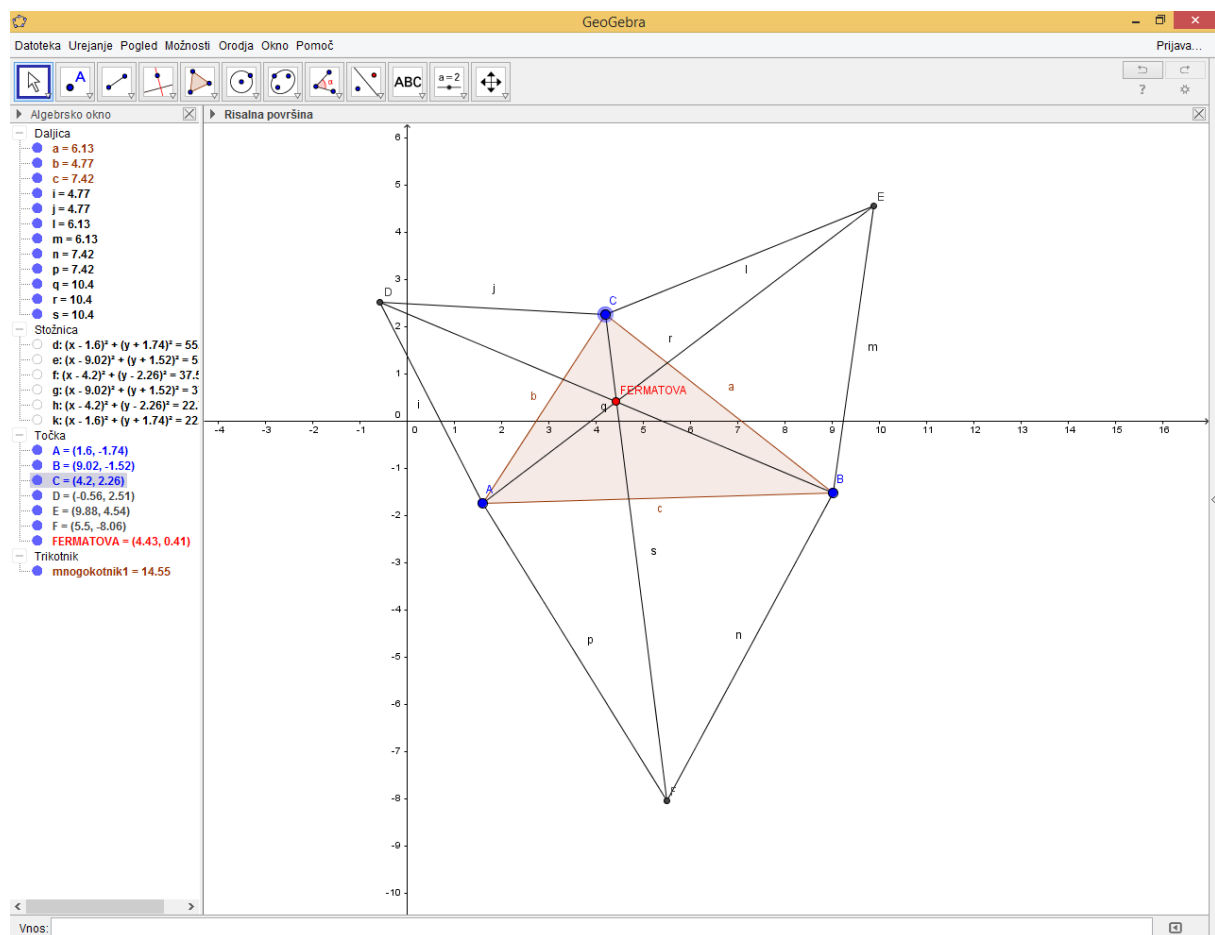
Uporaba GeoGebre in Mathematice pri konstrukciji Fermatove točke

Darko Petrović, Fakulteta za matematiko in fiziko, UL,
Darko.Petrovic@student.fmf.uni-lj.si

V prispevku si bomo ogledali kako lahko konstruiramo Fermatove točko z orodjima GeoGebra in Mathematica. Opis te je dosegljiv na naslovu https://en.wikipedia.org/wiki/Fermat_point.

Nazorno bomo prikazali na koliko vse načinov lahko narišemo Fermatovo točko. Našteli bomo vse njene značilnosti. Točko bomo narisali s pomočjo GeoGebre in njenimi enostavnimi vgrajenimi orodji. Nato bomo še izračunali lege Fermatove točke v različnih trikotnikih.

Trikotniki so pravilni, pravokotni, ostrokotni, topokotni ali enakokraki. V vsakem si bomo ogledali konstrukcijo Fermatove točke. Konstrukcijo trikotnikov in njihovih Fermatovih točk bomo poskušali izvesti tudi s programom Mathematica, navkljub znatno večji zapletenosti pri konstruiranju likov kot je v GeoGebri.



Kazalo
Program

Teorija Grafov

David Pančič, Fakulteta za matematiko in fiziko, UL, David.Pancic@student.fmf.uni-lj.si

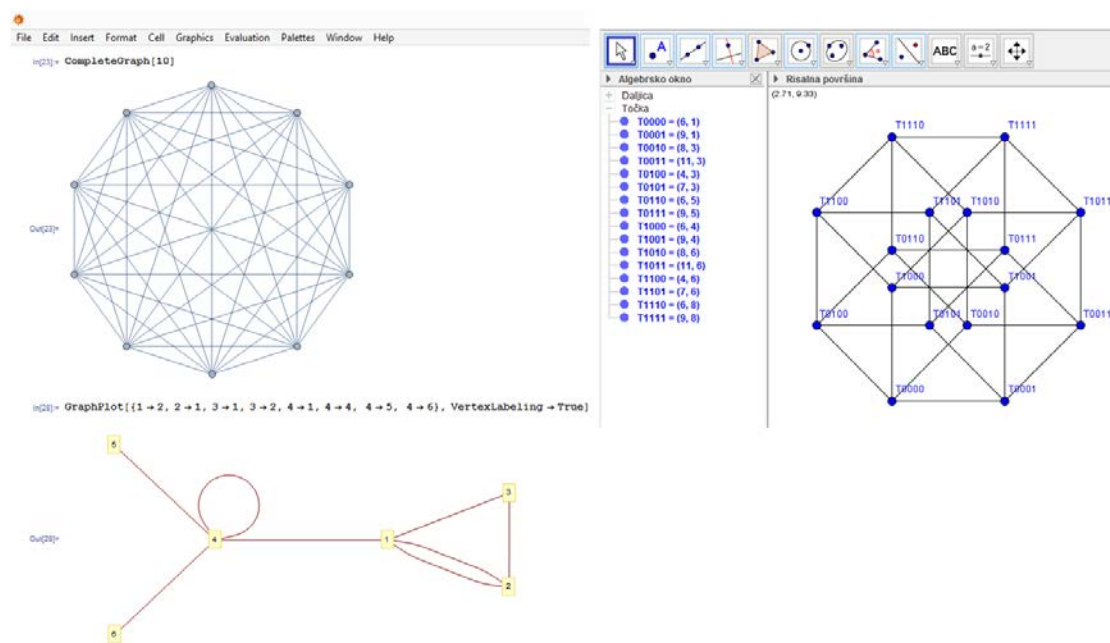
V tem predavanju se bomo ukvarjali s teorijo grafov s pomočjo programov Mathematica in GeoGebra.

Teorija grafov je matematična in računalniška disciplina ki raziskuje grafe. To so množice točk povezanih s povezavami.

Mathematica je sistem za numerično in simbolno računanje, v njem obstaja veliko ukazov vezanih za teorijo grafov. Ogledali si bomo nekatere od njih in pogledali, kako si lahko z njimi pomagamo pri reševanju nalog iz teorije grafov.

GeoGebra je prosto dostopen program za dinamično geometrijo. V njem ne obstajajo ukazi specifično za Teorijo grafov. Vseeno pa si z Geogebro lahko pomagamo pri reševanju tovrstnih nalog, če ne drugega vsaj pri pridobitvi občutka za kaj pri nalogi gre.

Skozi predstavitev bomo rešili nekaj kratkih nalog, analizirali grafe in preverili nekaj trditev.



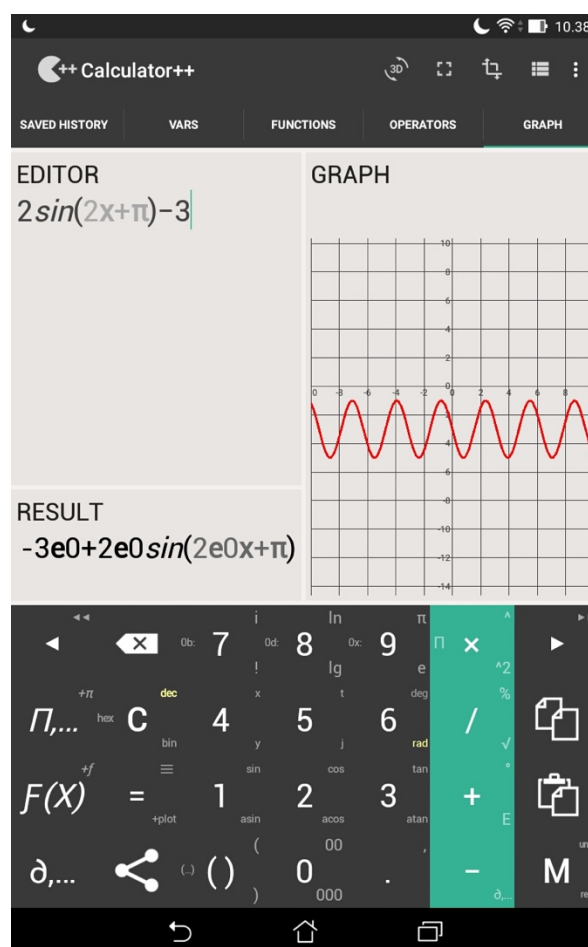
Kazalo
Program

Calculator ++

Erika Medoš, Fakulteta za matematiko in fiziko, UL, Erika.Medos@student.fmf.uni-lj.si

Calculator ++ je napreden kalkulator. Poleg enostavnih problemov lahko z njim rešujemo tudi kompleksnejše probleme. Na voljo sta dva načina: enostaven (simple) in napreden (engineer). Program ima na voljo veliko vgrajenih funkcij, uporabniku pa dopušča možnost dodajanja lastnih funkcij. Vgrajene ima tudi različne konstante, lahko pa dodamo tudi svoje. Calculator ++ ima opcijo graf (graph), ki omogoča risanje 2D in 3D grafov. Program je mogoče uporabljati na android napravah - na tablicah in mobilnih telefonih. Poleg dobimo še kalkulator, ki dela kot samostojen program in »lebdi« nad ostalimi aplikacijami.

V predstavitvi bom predstavila program bolj podrobno, ogledali si bomo tudi nekaj primerov, ki jih lahko rešimo s programom.



MalMath: Step by step solver

Erika Šavli, Fakulteta za matematiko in fiziko, UL, erika.savli@student.fmf.uni-lj.si

Mobilna aplikacija MalMath: Step by step solver je namenjena uporabi na pametnem telefonu ali tabličnem računalniku. Aplikacija je brezplačna in je za prenos na voljo v Google Play trgovini, za njeno delovanje pa ne potrebujemo internetne povezave.

MalMath je uporaben za reševanje limit, določenih in nedoločenih integralov, enačb, trigonometričnih enačb in tudi za risanje grafov funkcij. Kot pove že samo ime, step by step solver, je največja prednost aplikacije ta, da nam pri računanju, poleg rešitve, vrne tudi podroben opis korakov, s katerimi pridemo do le-te, pri risanju grafov pa poleg izrisanega grafa dobimo še analizo grafa, torej njegovo definicijsko območje, ničle, asimptote funkcije, ekstreme in intervale naraščanja in padanja. Takšen način podajanja rezultatov je koristen, saj se lahko iz korakov in njihovega opisa naučimo kako priti do rezultata oz. kako čim bolj natančno, z upoštevanjem vseh lastnosti, ki jih dobimo pri analizi grafa, ta graf tudi narisati.

V predstavitvi si bomo ogledali nekaj zgledov reševanja limit in risanja grafov iz vaj in domačih nalog predmeta Matematika 1.

The screenshot shows the MalMath app interface. On the left is a 'Home' screen with a calculator keypad and a list of functions. On the right is a 'Worksheet' screen showing a step-by-step solution for the limit $\lim_{x \rightarrow \infty} \left(\frac{2x+3}{2x+7} \right)^{2x}$.

Home Screen:

- Buttons: $\int$, $\int_a^b$, d/dx , $\lim$, ∞ , e
- Buttons: $\sin$, $\cos$, $\tan$, $\cot$, $\sec$, $\csc$
- Buttons: x^2 , y^x , $\sqrt{x}$, $x\sqrt{y}$, $\log$, $\ln$
- Buttons: x , y , z , π , θ , abc
- Buttons: 1, 2, 3, 4, 5, 6
- Buttons: 7, 8, 9, 0, (, .
- Buttons: +, -, *, ÷, =, $\frac{\square}{\square}$
- Buttons: GRAPH, <, >, $\Leftrightarrow$, $\Rightarrow$, SOLVE

Worksheet Screen:

Problem: $\lim_{x \rightarrow \infty} \left(\frac{2x+3}{2x+7} \right)^{2x}$

Step 1: $\lim_{x \rightarrow \infty} \left(\frac{2x+7}{2x+7} + \frac{2x+3-2x-7}{2x+7} \right)^{2x}$

Use $a/b = a/a + a-b/b$

Step 2: $\lim_{x \rightarrow \infty} \left(1 + \frac{-4}{2x+7} \right)^{2x}$

Simplify the quotient

Step 3: $\lim_{x \rightarrow \infty} \left(1 + \frac{-4}{2x+7} \right)^{2x+7} \left(1 + \frac{-4}{2x+7} \right)^{2x-7}$

Use $a^k x = a^k k a^{(x-k)}$

Step 4: $\lim_{x \rightarrow \infty} \left(1 + \frac{-4}{2x+7} \right)^{2x+7} \lim_{x \rightarrow \infty} \left(1 + \frac{-4}{2x+7} \right)^{2x-7}$

Use the limit property for product: $\lim_{x \rightarrow a} f(x)q(x) = \lim_{x \rightarrow a} f(x) \lim_{x \rightarrow a} q(x)$

Step 5: $e^{-4} \cdot \left(1 - \frac{4}{2x+7} \right)^{-7}$

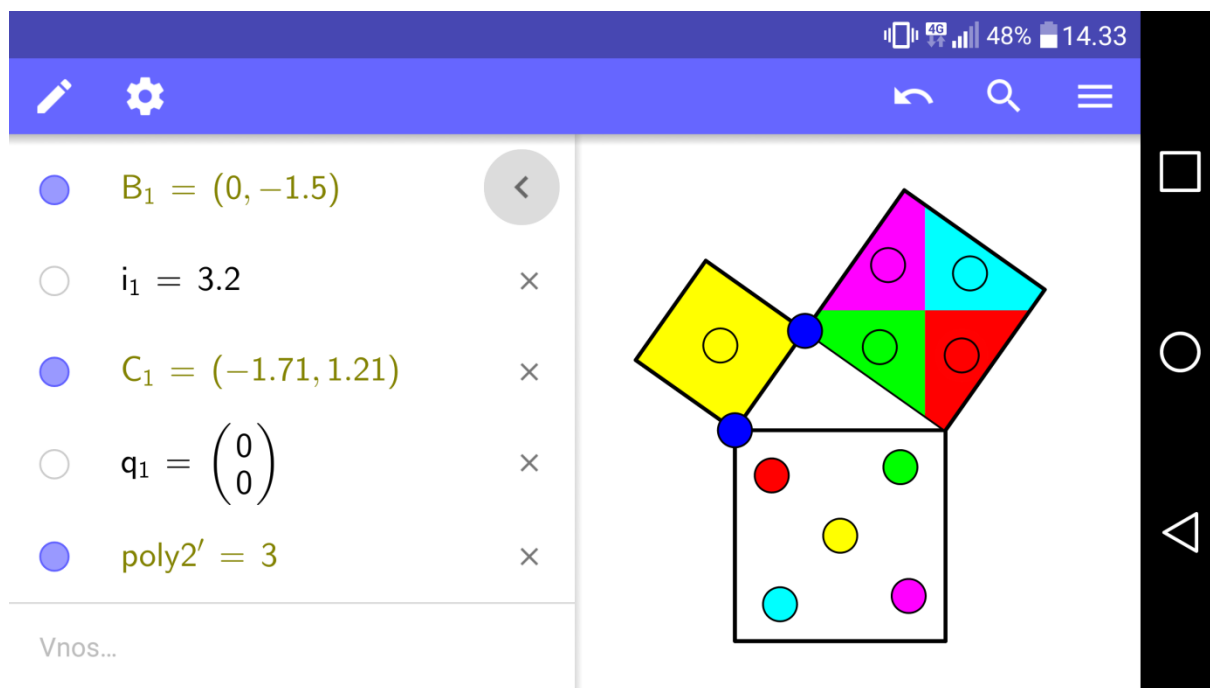
Simplify branches

Geogebra Graphing Calculator

Gaja Rihar, Fakulteta za matematiko in fiziko, UL, Gaja.Rihar@student.fmf.uni-lj.si

Geogebra Graphing Calculator je program, prirejen po računalniški različici svetovno znane Geogebre, le da je ta namenjen uporabi na pametnih telefonih (rahlo drugačna oblika za prilagajanje telefonskemu ekranu, a iste funkcije). S pomočjo interaktivnega sistema omogoča izobraževanje na tehničnem, matematičnem in naravoslovnem področju. Z mnogimi vgrajenimi funkcijami združuje geometrijo, risanje, algebro, tabele, statistiko in diferencialni račun. Predstavlja izredno dober način za vizualizacijo in nazorno predstavo v naravoslovju. Zaradi svojega dinamičnega prikaza matematike, preproste uporabe in predvsem poučne interaktivnosti je primeren za vse stopnje izobraževanja in je prijazen vsem uporabnikom.

Geogebra je svetovno znan in najbolj uporabljen program na svetu na tem področju. Različica namenjena pametnim telefonom tako doda še 'piko na i' s svojo lahko in hitro dostopnostjo. Datoteke s končnico .ggb, primerne za uporabo na računalniku, je moč odpreti tudi v mobilni verziji, kjer vse dela enako (problem lahko nastane le pri postavitvi objektov po prostoru, saj se dimenzije ekrana spremenijo). Izvoz gradiva je prav tako možen z različnimi oblračnimi programi, kot so Gmail, GoogleDrive, Bluetooth. Ob prijavi v GeoGebra Tube (uporabite lahko že kak obstoječ naslov npr. Facebook ipd.) pa se celotno delo lahko shrani na profil tako, da do njega lahko dostopamo z vseh naprav.



[Kazalo](#)
[Program](#)

Epski krogi

Gregor Vavdi, Fakulteta za matematiko in fiziko, UL, gregor.vavdi@student.fmf.uni-lj.si

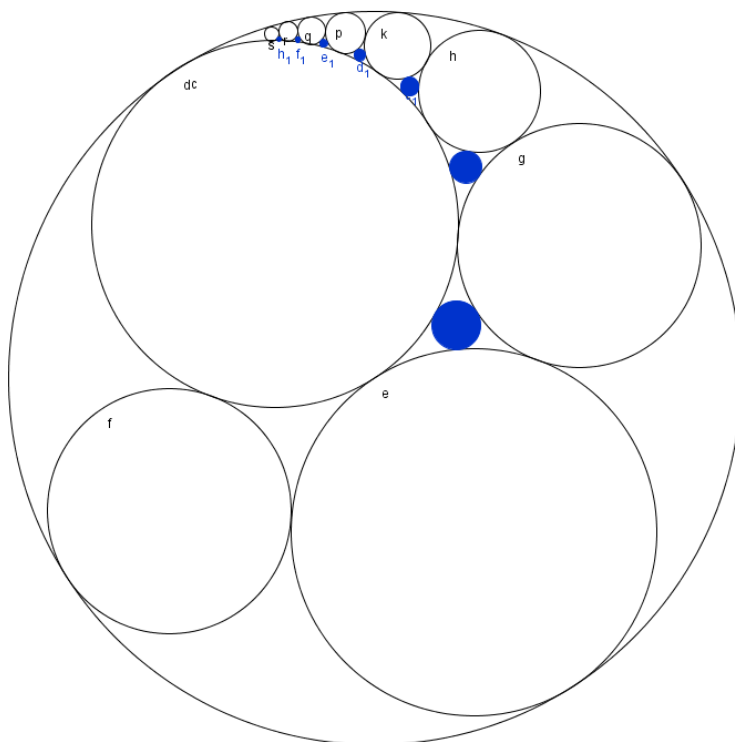
Ali znate določiti naslednji člen zaporedju:

$$\frac{1}{15}, \frac{1}{23}, \frac{1}{39}, \frac{1}{63}, ?$$

Zaporedje je na prvi pogled videti povsem običajno, vendar se v njem skriva veliko več kot mislimo na začetku.

Pokazali bomo, da je to zaporedje povezano s krogi. Vzeli bomo krog in z ustrezno konstrukcijo pričarali zgornje zaporedje. Ker pa smo matematiki, bomo vse skupaj še dokazali s pomočjo inverzije. Seveda si bomo pomagali s računalniškim orodjem GeoGebra, da bo stvar pravilno in natančno narisana.

Če ste preveč neučakani in bi že danes radi izvedeli kakšno povezavo ima zaporedje z krogom, si lahko to pogledate na naslednjem videu: https://www.youtube.com/watch?v=sG_6nIMZ8f4,



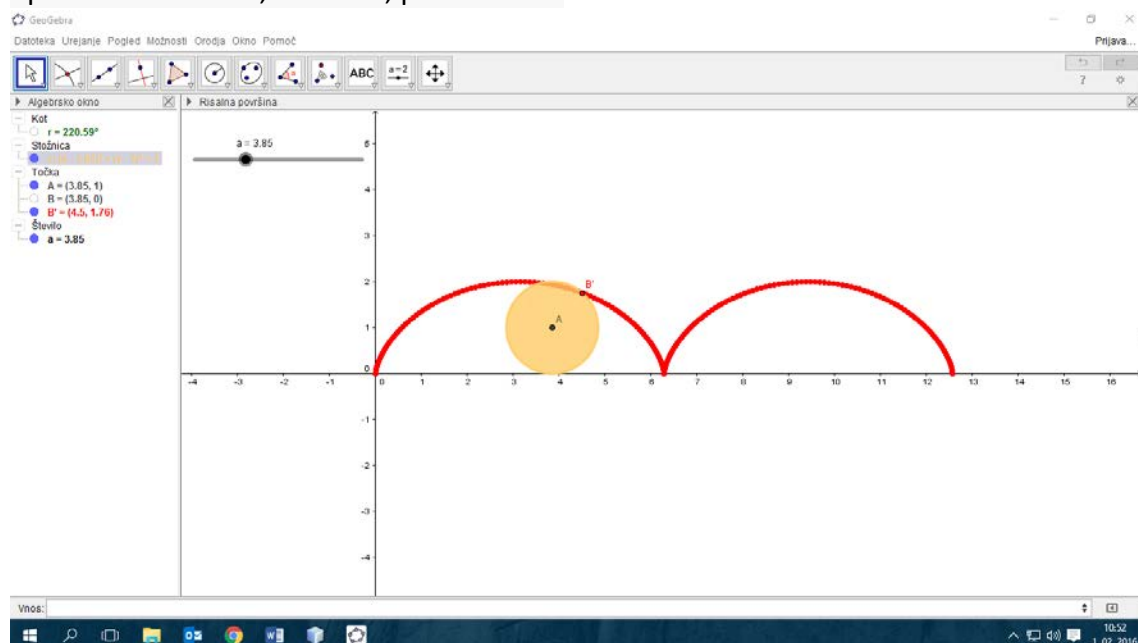
Konstrukcija cikloide s pomočjo GeoGebre

Jaka Pivk, Fakulteta za matematiko in fiziko, UL, jaka.pivk@student.fmf.uni-lj.si

Cikloida je krivulja, ki jo pri gibanju geometrijskega lika po neki premici ali krivulji opiše na tem liku izbrana točka. Na osnovi gradiva s spletnega naslova <http://mathandmultimedia.com/2010/07/30/geogebra-tutorial-constructing-a-cycloid/> si bomo ogledali, kako cikloido konstruiramo s pomočjo Geogebre.

Izberemo opcijo Rotiraj objekt okoli točke in kliknemo na točko A in B ter v vnosno polje vnesemo r. Pojavi se nam nova točka B' katera bo naredila krivuljo. Za lepšo izboljšavo na koncu lahko pobarvamo točko, omogočimo sled, pobarvamo objekt in omogočimo animacijo.

V prispevku si bomo ogledali, kakšne različne cikloide dobimo, če kot osnovni objekt uporabimo kvadrat, trikotnik, pravokotnik.



Reševanje rekurzivnih enačb

Jakob Valič, Fakulteta za matematiko in fiziko, UL, Jakob.Valic@student.fmf.uni-lj.si

V prispevku si bomo ogledali reševanje rekurzivnih enačb s pomočjo programa Mathematica. Gre za enačbe zaporedij, katerih naslednji člen je podan s kombinacijo prejšnjih.

Uporabljali bomo funkcijo **RSolve**, s katero Mathematica določi splošni člen zaporedja. Splošni člen je natančno določen v primeru, ko podamo tudi začetni člen zaporedja, ki je označen z a_0 . Če tega člena ne poznamo, nam Mathematica vrne splošni člen zaporedja, ki v svoji formuli vključuje eno ali več konstant, označenih s $C[1]$, $C[2]$, itd. Te konstante lahko naknadno določimo s pomočjo prepisovalnih pravil. Rekurzivno podano zaporedje lahko napišemo tudi v obliki funkcije, ki jo lahko predstavimo grafično. Namesto funkcije **RSolve** lahko uporabljamo tudi **RSolveValue**. Razlika je v tem, da nam **RSolve** poda rešitev v dvojnih zavirih oklepajih, **RSolveValue** pa brez oklepajev.

Na koncu si bomo pogledali še praktično uporabo reševanja rekurzivnih enačb, in sicer pri računanju korakov, potrebnih za premestitev n -obročev pri igri Hanojski stolpi; drugi primer bo iz obrestnih mer.

Primer reševanja preproste rekurzivne enačbe:

Reši rekurzivno enačbo $a[n+1] = (a[n]^2) / (a[n]+4)$.

V prvem primeru dobimo splošni člen zaporedja, v katerem nastopa nedoločena konstanta $C[1]$.

```
RSolve[a[n + 1] == (a[n] * 2) / (a[n] + 4), a[n], n]
```

$$\left\{ \left\{ a[n] \rightarrow \frac{2}{-1 + 2^n + 2^n C[1]} \right\} \right\}$$

V drugem primeru smo določili začetni člen zaporedja, zato je splošni člen zaporedja natančno določen.

```
RSolve[{a[n + 1] == (a[n] * 2) / (a[n] + 4), a[0] == 1}, a[n], n]
```

$$\left\{ \left\{ a[n] \rightarrow \frac{2}{-1 + 3 \times 2^n} \right\} \right\}$$

V tretjem primeru dobimo rešitev, napisano v obliki funkcije.

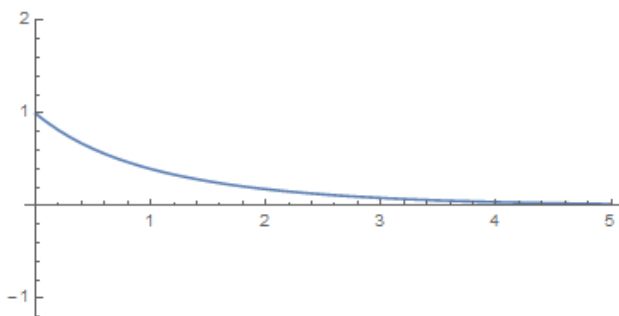
```
res = RSolveValue[{a[n + 1] == (a[n] * 2) / (a[n] + 4), a[0] == 1}, a, n]
```

$$\text{Function}\left[\{n\}, \frac{2}{-1 + 3 \times 2^n}\right]$$

To funkcijo lahko prikažemo grafično. Vrednosti na osi y pri določenem x nam povedo vrednost n -tega člena tega zaporedja, pri čemer je $n=x$.

```
f[x_] := res[x]
```

```
Plot[f[x], {x, 0, 5}, PlotRange -> {-2, 2}]
```



Jan Tomšič Pivk, Fakulteta za matematiko in fiziko, UL, jan.tomsic-pivk@student.fmf.uni-lj.si

V predavanju bom najprej na kratko opisal nastanek enciklopedije, nato pa se spustil v podrobnosti uporabe iskalnika in drugih orodij, ki so dostopna na strani ter ob nekaj primerih razložil kaj vse lahko v enciklopediji izvemo o nekem zaporedju.

THE ON-LINE ENCYCLOPEDIA
OF INTEGER SEQUENCES®

The OEIS Foundation is grateful to everyone who made a donation during our Annual Appeal. Visit the new and spectacular [Pictures from the OEIS](#) page!

a134204 Search [History](#)

(Greetings from [The On-Line Encyclopedia of Integer Sequences!](http://www.oeis.org))

Search: **a134204**

Displaying 1-10 of 15 results found.

page 1 2

Sort: [relevance](#) [references](#) [number](#) [modified](#) [created](#) Format: [long](#) [short](#) [data](#)

A134204 $a(0)=2$; for $n>0$, $a(n)$ = smallest prime not occurring earlier in the sequence such that $a(n-1)+a(n)$ is a multiple of n . If no such prime exists, the sequence terminates.

2, 3, 5, 7, 13, 17, 19, 23, 41, 51, 29, 37, 11, 67, 59, 61, 83, 53, 73, 79, 101, 109, 89, 233, 103, 47, 239, 139, 113, 293, 97, 151, 137, 127, 43, 167, 157, 509, 251, 373, 107, 467, 163, 181, 347, 193, 313, 439, 281, 307, 443, 271, 197, 227, 367, 733, 331, 353, 401, 71, 229 ([list](#); [graph](#); [refs](#); [listen](#); [history](#); [text](#); [internal format](#))

OFFSET	0,1
--------	-----

COMMENTS Is

This sequence infinite if and only if $a(n-1)$ never divides n for any n .
 This sequence exists for at least 800×10^6 terms (see [A133242](#), [A133243](#), [A232992](#)). - David Applegate, Nov 01 2007, Nov 15 2007
 The plot of primes less than 10^6 shows an interesting crosshatch pattern. Why? [\[I. D. Noy, Jul 12 2009\]](#) See also the graph of [A133244](#). - [N. J. A. Sloane](#), Apr 06 2013

Entries [A224221](#), [A224222](#) are similar sequences which terminate after 20 or so steps, while [A224223](#) and [A224229](#) are similar sequences whose status is also unknown. - [N. J. A. Sloane](#), Apr 05 2013

LINKS Robert Israel and Reinhard Zumkeller, [Table of \$n, a\(n\)\$ for \$n = 0..100000\$](#) initial 1000 terms from Robert Israel

David Applegate, [C++ Program](#) [For output see [A133242](#), [A133243](#), [A232992](#)]

T. D. Noe, [Graph of initial terms \(out to \$10^6\$ \)](#)

N. J. A. Sloane, [Eight Hateful Sequences](#), a short paper for the 8th Gathering for Gardner, May 2008.

EXAMPLE The primes that don't occur among terms $a(0)$ through $a(6)$ form the sequence 11, 23, 29, 31, ... Of these, 23 is the smallest that when added to $a(6)=19$ gets a multiple of 7 -- $19+23 = 42 = 6 \cdot 7$. ($19+11 = 30$, which is not a multiple of 7.) So $a(7) = 23$.

```
aa := {a[0]=2, a[1]=3}; a[n_] := a[n] = {an = First[ Complement[ Prime[ Range[1 + PrimePi[ Max[aa]]], aa]]; While[ Not[ FreeQ[aa, an] && Divisible[ a[n-1] + an, n]], an = NextPrime[an]]; AppendTo[aa, an]; an); Table[a[n], {n, 0, 60}] (* Jean-  
François Alcover, Oct 17 2012*)
```

T. D. Noe, Apr 05 2013, provided the following information about how his plot (see [link](#)) was obtained: I computed 500000 points and then plotted up to $y = 10^6$. Here's the Mma code (which takes a while to run):

```
t = {2}, Dofk = ceiling(t[[1]]/n).
```

Eulerjeva premica in trilinearni koordinatni sistem

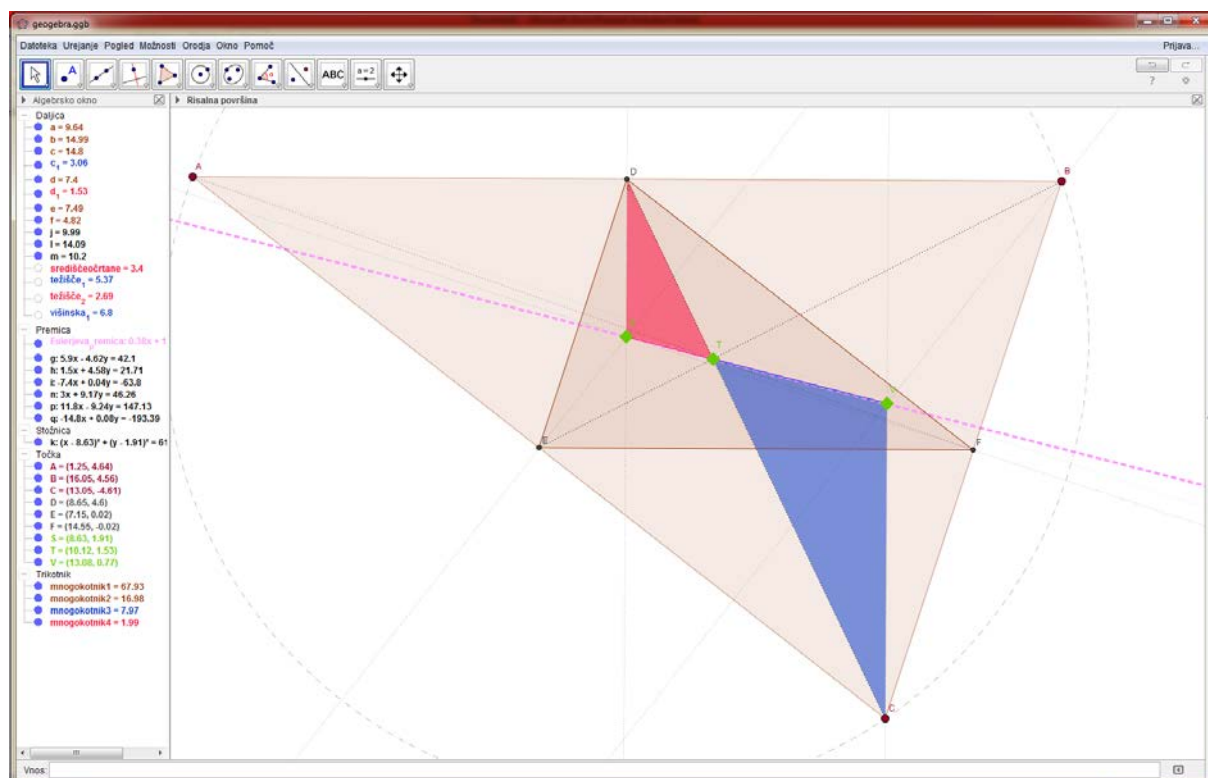
Jure Srabotnik, Fakulteta za matematiko in fiziko, UL,
jure.srabotnik@student.fmf.unilj.si

Eulerjeva premica je v geometriji premica v poljubnem neenakostraničnem trikotniku, ki poteka skozi posebne točke trikotnika. Ene izmed teh so: višinska točka, središče trikotniku očrtane krožnice in težišče trikotnika. S pomočjo podobnih trikotnikov bom dokazal, da višinska točka, središče očrtane krožnice in težišče trikotnika ležijo na isti premici.

Pri dokazovanju si bom pomagal s trilinearnim koordinatnim sistemom. Trilinearni koordinatni sistem opisuje lege točk glede na dani trikotnik. Trilinearne koordinate opisujejo relativne razdalje do treh stranic trikotnika; zato niso enolično določene, saj če jih množimo s poljubnim neničelnim realnim številom dobimo ponovno trilinearne koordinate dane točke.

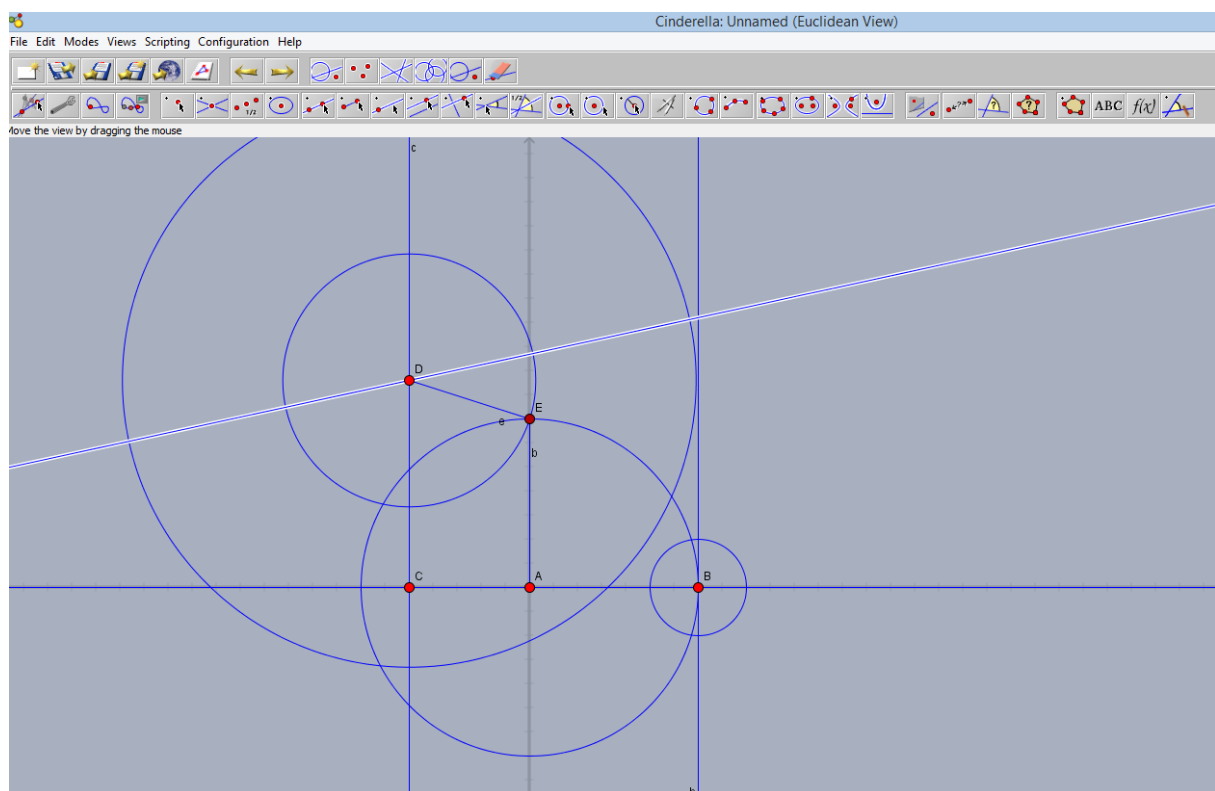
S trilinearnimi koordinatami točk bom ugotovil ali so le-te res kolinearne, saj so poljubne tri točke A, B, C s trilinearnimi koordinatami $a_1 : a_2 : a_3$, $b_1 : b_2 : b_3$, $c_1 : c_2 : c_3$ kolinearne samo če je determinanta $D =$

$$\begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix} \text{ enaka nič.}$$



Karin Stančič, Fakulteta za matematiko in fiziko, UL,
karin.stancic@student.fmf.uni-j.si

Med drugim si bomo ogledali konstrukcijo »Watt the walking beam«, kjer bomo sledili navodilom iz posnetka <https://www.youtube.com/watch?v=qwGilSJNFao>.



Cymath

Katja Bukovec, Fakulteta za matematiko in fiziko, UL,

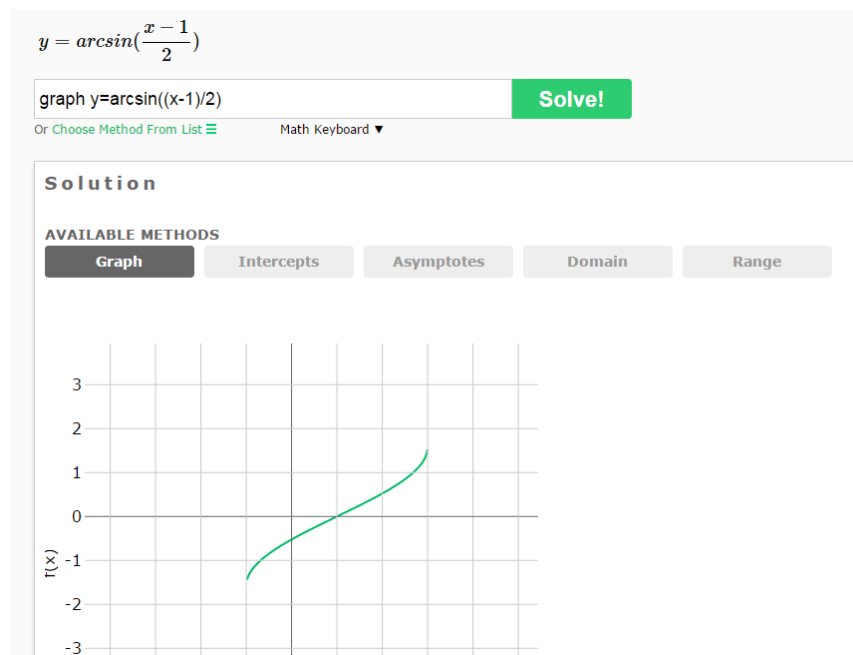
katja.bukovec@student.fmf.uni-lj.si

Cymath je internetna stran, namenjena reševanju matematičnih problemov. Dosegljiva je na internetu in ne zahteva prijave. Pri računanju nam izpiše celoten postopek reševanja. Ob vnosu si lahko izberemo metodo, s katero bomo rešili svoj problem. Na voljo imamo:

- dopolnjevanje enačbe do popolnega kvadrata, ki nas od splošne pripelje do temenske oblike kvadratne funkcije;
- diferenciacijo;
- reševanje enačb
- razširjanje enačb
- faktorizacijo
- risanje grafov
- integriranje
- razčlenjevanje na prafaktorje
- poenostavljanje enačb

Stran je dostopna na naslovu: <http://www.cymath.com/>. Na voljo je tudi kot mobilna aplikacija. Uporaba je podobna.

Na predavanju si bomo ogledali nekaj primerov reševanja matematičnih problemov s Cymath.



Geometrija v Mathematici

Lara Jerač, Fakulteta za matematiko in fiziko, UL, lara.jerac@student.fmf.uni-lj.si

Mathematica je programska oprema, ki jo uporabljamo na različnih znanstvenih področjih - v matematiki, računalništvu, ekonomiji, tehniki, fiziki, statistiki itd.

Podrobneje si bomo pogledali kako v Mathematici rešujemo geometrijske probleme.

Mathematica ima že vgrajene funkcije, s pomočjo katerih lahko računamo površino, obseg in volumen. Prav tako, lahko tudi rišemo skice likov in teles. To naredimo tako, da pokličemo funkcijo (ki je kar ime lika oz. telesa, ki ga želimo izrisati) in podamo podatke, s katerimi je naš lik/telo določen. Sliki lahko spreminjamo barvo in zorni kot.

V predstavitvi bomo izrisali paralelogram in si pogledali, kako računamo obseg in ploščino. Izračunali bomo tudi kote.

V 3D si bomo pogledali paralelepiped in kroglo. Narisali bomo nekaj različnih skic in računali volumen.

Spoznali bomo tudi ukaz, s pomočjo katerega določimo, na katerih vektorjih je paralelepiped podan.

Untitled-1 * - Wolfram Mathematica 10.0 Student Edition - Personal Use Only

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

Wolfram Mathematica STUDENT EDITION

Skica paralelepipeda podanega z vektorji $(1, 0, 0)$, $(1, 1, 0)$ in $(0, 1, 1)$ in je postavljen v izhodišče $(0, 0, 0)$.

```
In[3]:= p = Parallelepiped[{0, 0, 0}, {{1, 0, 0}, {1, 1, 0}, {0, 1, 1}}]
Out[3]= Parallelepiped[{0, 0, 0}, {{1, 0, 0}, {1, 1, 0}, {0, 1, 1}}]
In[4]:= Graphics3D[{Opacity[0.25], Blue, p}]
Out[4]=
```

Izračun volumna.

```
In[5]:= Volume[p]
Out[5]= 1
```

binary form prime? perfect number? Roman numerals more...

Uporaba knjižnice NumPy pri problemih iz Linearne Algebre

Larisa Carli, Fakulteta za matematiko in fiziko, UL, Larisa.Carli@student.fmf.uni-lj.si

NumPy (Numeric Python) je prosto dostopen dodatek programskemu jeziku Python, ki dodaja podporo za enostavno in učinkovito uporabo velikih, več-dimenzionalnih tabel oziroma matrik, na katerih lahko uporabljamo matematične in druge funkcije. Uporaba knjižnice zahteva le malo kode, zaradi česar je koda lahko berljiva in pregledna.

Na predstavitvi bomo povedali nekaj o kriptografiji in linearnih preslikavah, podali zglede, kje se področji uporabljata v naših vsakdanjih življenjih in pokazali, kako lahko uporabimo knjižnico NumPy pri nekaj enostavnih primerov uporabe linearnih transformacij in kodiranja sporočil, s pomočjo matričnega množenja.

Pokazali bomo, kako na različne načine s NumPy-jem preslikamo poljuben objekt in koordinate zapišemo v sintaksi primerno za kopiranje v Mathematico, s katero si bomo pomagali za prikaz teh objektov. Pogledali bomo tudi, kako s matriko zakodiramo poljubno sporočilo in ga potem odkodiramo s pomočjo njenega inverza.

```
def skalirnaMatrika(skalarji):
    """Sprejme seznam treh skalarjev, vrne skalirno matriko."""
    if len(skalarji) != 3:
        print("Skalarji morajo biti natanko trije.")
        return None
    skalarX, skalarY, skalarZ = skalarji[0], skalarji[1], skalarji[2]
    Rs = np.array([[skalarX, 0, 0],
                  [0, skalarY, 0],
                  [0, 0, skalarZ]])
    return Rs

def mathematicaSintaksa(matrika):
    """Sprejme seznam točk v niz, primeren za kopiranje v Mathematico (vrne niz)."""
    seznam = []
    # Najprej pretvorimo matriko v seznam stolpcev:
    for i in range(matrika.shape[1]): #rabimo število stolpcev
        vrstica = list(matrika[:, i]) #i-ti stolpec matrike
        seznam.append(vrstica)
    niz = ""
    #pretvorimo seznam v ustrezno oblikovan niz:
    for točka in seznam:
        niz += "(" + str(točka)[1:-1] + ", "
    return niz[:-2] + ")"

def rotirajOkoliOsi(alpha, os):
    """Zarotira objekt za alpha stolpcev okoli osi os (x, y, z). Alpha v radianih."""
    R = rotacijskaMatrikaOs(os, alpha)
    rotiranObjekt = np.dot(R, objekt)
    koordinate = mathematicaSintaksa(rotiranObjekt)
    print("Koordinate točk v sintaksi Mathematice so:\n{}\n".format(koordinate))

def skalirajObjekt(skalarX, skalarY, skalarZ):
    """Skalira objekt za skalarje v smerih x, y, z."""
    Rs = skalirnaMatrika(skalarji)
    skaliranObjekt = np.dot(Rs, objekt)
    koordinate = mathematicaSintaksa(skaliranObjekt)
    print("Koordinate točk v sintaksi Mathematice so:\n{}\n".format(koordinate))

def skalirajInRotirajObjektOkoliOsi(skalarji, alpha, os):
    """Naredi dve transformaciji na objektu. Sprejme seznam treh skalarjev, kot v radianih ter os."""
    Rs = skalirnaMatrika(skalarji)
    skaliranObjekt = np.dot(Rs, objekt)
    R = rotacijskaMatrikaOs(os, alpha)
    rotiranInSkaliranObjekt = np.dot(R, skaliranObjekt)
    koordinate = mathematicaSintaksa(rotiranInSkaliranObjekt)
    print("Koordinate točk v sintaksi Mathematice so:\n{}\n".format(koordinate))

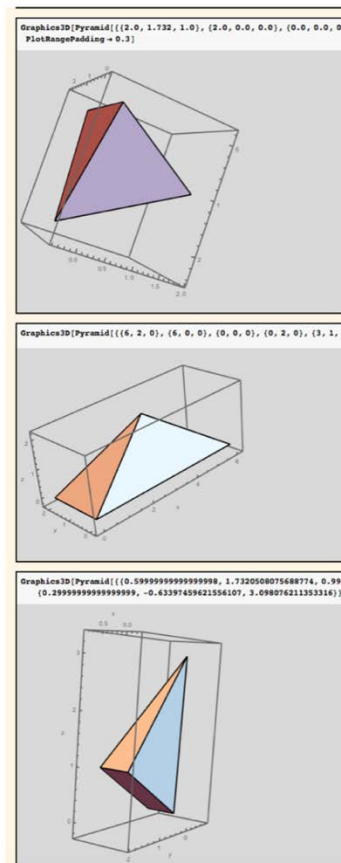
skalirajInRotirajObjektOkoliOsi([0.3, 1, 1.5], np.pi/6, "x")
```

```
[In [64]: %run kodiranje.py
Sporočilo: ZAKODIRAMO LAHKO POLJUBNO SPOROČILO.

Sporočilo izgleda zapisano s številskimi takole:
[[ 22.  15.  17.  15.  1.  15.  15.  20.  15.  16.  15.  12.]
 [ 1.  4.  3.  23.  8.  23.  12.  2.  23.  15.  3.  15.]
 [ 11.  9.  13.  12.  11.  16.  10.  14.  18.  17.  9.  26.]]

Sporočilo smo zakodirali z matriko
[[ -3  -3  -4]
 [ 1  1  1]
 [ 4  3  4]]
nakar izgleda tako:
[[ -113.  -93.  -106.  -162.  -71.  -178.  -121.  -122.  -186.  -161.  -90.  -185.]
 [ 34.  28.  31.  50.  20.  54.  37.  36.  56.  48.  27.  53.]
 [ 135.  108.  123.  177.  72.  193.  136.  142.  201.  177.  105.  197.]]

Po odkodiranju dobimo nazaj sporočilo:
ZAKODIRAMO LAHKO POLJUBNO SPOROČILO.
```



Kazalo

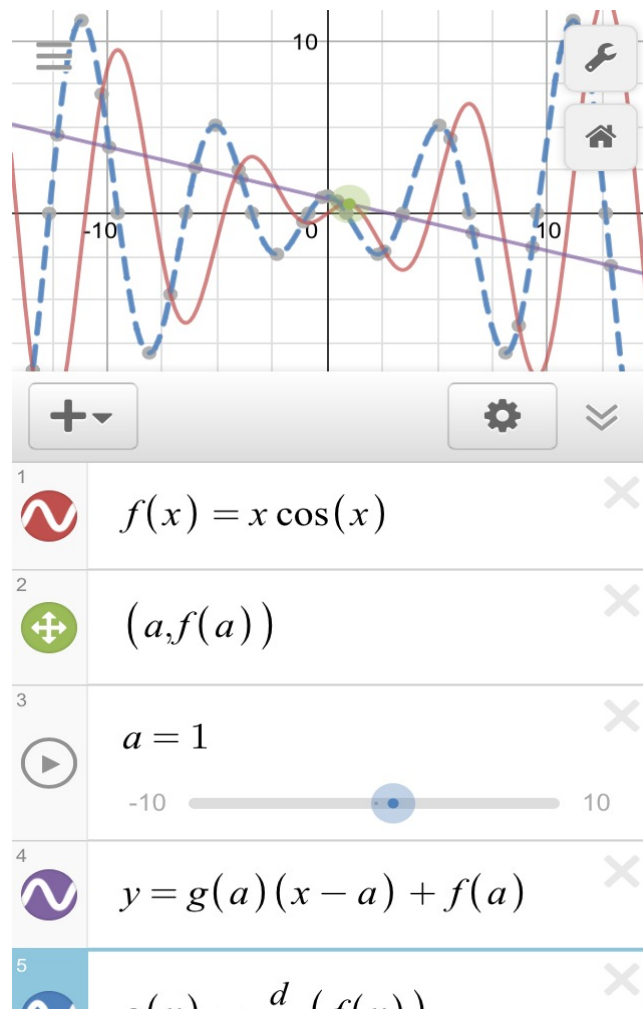
Program

Lea Kos, Fakulteta za matematiko in fiziko, UL, lea.kos@student.fmf.uni-lj.si

Desmos je aplikacija za risanje grafov, vnašanje tabel, drsnikov ter animacij grafov. Aplikacija je dostopna v app store in google play. Uporabljamo jo lahko na računalnikih, tablicah ter telefonih, brezplačno in brez interneta.

V program preprosto vnesemo poljubno funkcijo od linearnih pa vse do krožnih funkcij. Enostavno definiramo funkcijo in nam jo program nariše. Za funkcije, ki so podane z različnimi parametri, nam program ponudi drsnike, te pa lahko omejimo na določenem intervalu z določenim korakom.

Na predstavitvi bom predstavila kvadratno funkcijo ($f(x) = ax^2 + bx + c$), kjer bomo imeli parametre a, b in c . Ogledali pa si bomo tudi nekaj zanimivosti tega programa.



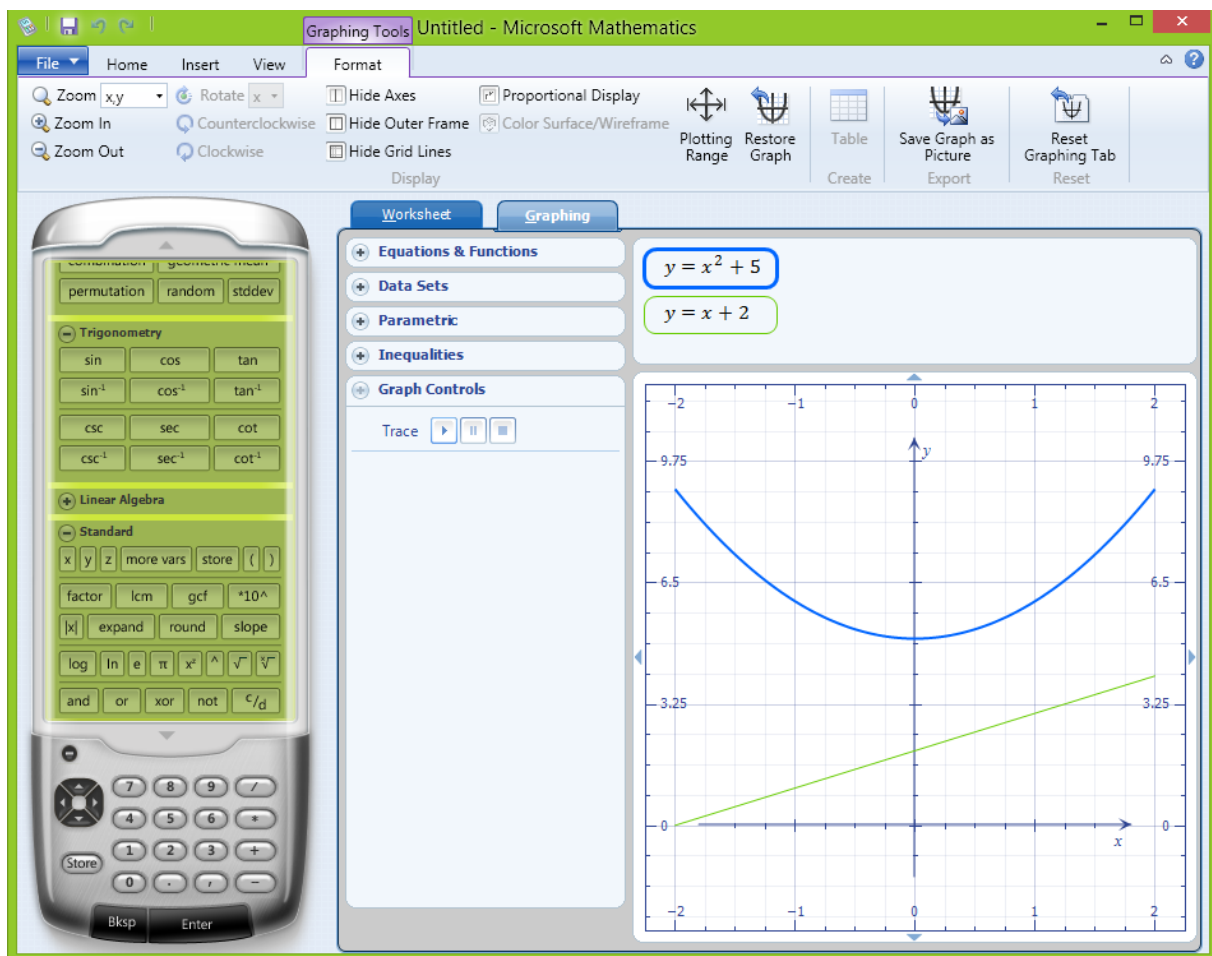
Microsoft Mathematics

Lea Temlin, Fakulteta za matematiko in fiziko, UL, lea.temlin@student.fmf.uni-lj.si

Spoznali bomo uporabo programa Microsoft Mathematics. Program si lahko prenesemo brezplačno: <https://www.microsoft.com/en-us/download/confirmation.aspx?id=15702>

Microsoft Mathematics je program za reševanje matematičnih problemov. Program je študentom namenjen kot učni pripomoček. Ponuja nam velik nabor matematičnih orodij, s katerimi si študentje lahko hitro in enostavno pomagajo pri računanju, vključno z osnovnim kalkulatorjem z vsemi funkcijami, ki je namenjen za delo tako kot ročni kalkulator. Program nam omogoča reševanje enačb, računanje odvodov, limit in integralov. Rišemo lahko grafe v 2D in 3D obliki. Na voljo je tudi orodje pretvornik enot, s katerim lahko hitro in enostavno pretvarjamo merske enote.

V predstavitvi si bomo ogledali uporabo osnovnega kalkulatorja in uporabo matematičnih orodij. Na konkretnih primerih bomo spoznali osnovne ukaze za računanje in risanje grafov.



Symbolab

Marina Kovač, Fakulteta za matematiko in fiziko, UL,
marina.kovac@student.fmf.uni-lj.si

Programov in orodij, s katerimi si lahko pomagamo pri reševanju matematičnih nalog in problemov, je veliko. Večina pa je takšnih, ki vrnejo samo končni rezultat. Ker pa večino napak pri reševanju nalog naredimo nekje med samim postopkom računanja, bi nam dostikrat prav prišel program, ki nam za določen problem vrne tudi postopek reševanja.

Symbolab je eno od takšnih orodij. Za probleme (seveda ne vse) iz različnih zvrsti matematike nam vrne tako rezultat, kot tudi postopek reševanja. Ker pa na žalost nobeno orodje ni popolno, včasih nastanejo tudi problemi. Tako na primer za kakšen primer vrne različen postopek reševanja od tistega, po katerem smo računali (vemo recimo, da lahko konvergenco vrste preverjamo z različnimi kriteriji). Tudi za vse probleme ne vrne postopka reševanja, in žal v takšnih primerih, tudi končnega rezultata ne.

V predstavitvi orodja si bomo najprej pogledali njegovo okolje. Spoznali se bomo z njegovimi prednostmi in slabostmi, rešili pa bomo tudi kakšen primer iz vaj za predmeta Matematika 1 in Linearna algebra.

The screenshot shows the Symbolab interface for the function $f(x) = \frac{1-x}{(x+1)^2}$. At the top, the function is entered in a text box, with a red "Go" button to its right. Below the input box are social media sharing icons (Facebook, Google+, Twitter, Email, Print) and links for "Graph" and "Examples".

The "Solution" section is expanded, showing the domain of the function. It states: "Domain of $\frac{1-x}{(x+1)^2}$:" followed by a box containing the solution: "Solution: $x < -1$ or $x > -1$ " and "Interval Notation: $(-\infty, -1) \cup (-1, \infty)$ ". A red link "« Hide Steps" is visible.

Below the domain, the "Steps" section is expanded. It includes a "Domain definition" section with the text: "The domain of a function is the set of input or argument values for which the function is real and defined". There is a "Find undefined (singularity) points:" section showing $x = -1$. A "Show Steps" button with a plus icon is also present.

The "The function domain" section shows the result: $x < -1$ or $x > -1$. A red link "click here to start symbolab practice »" is at the bottom left of this section, and a "Save" button is at the bottom right.

At the bottom, the "Range" of the function is shown. It states: "Range of $\frac{1-x}{(x+1)^2}$:" followed by a box containing the solution: "Solution: $-\frac{1}{8} \leq f(x) < 0$ or $f(x) > 0$ " and "Interval Notation: $[-\frac{1}{8}, 0) \cup (0, \infty)$ ". A red link "Show Steps »" is visible.

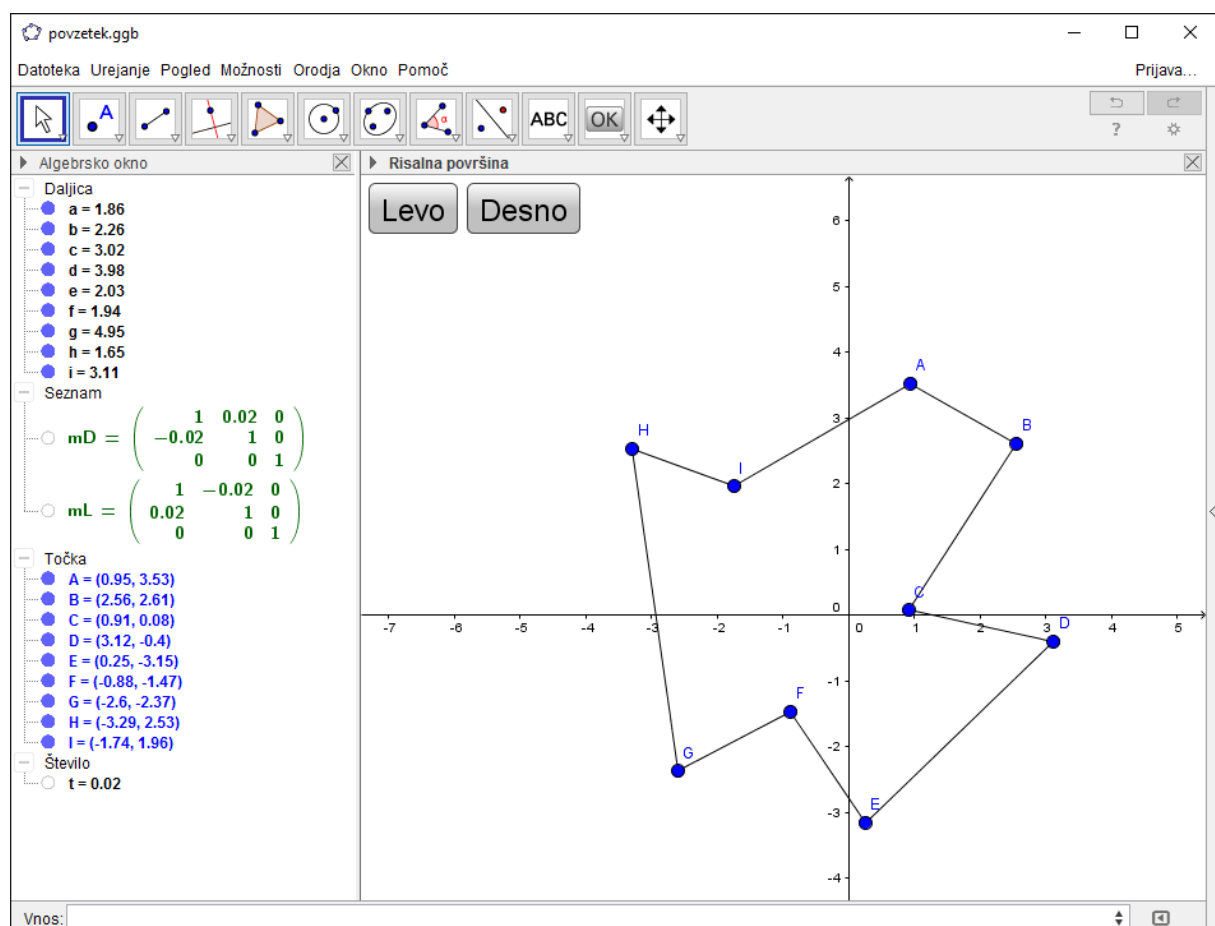
Rotacije, translacije in skaliranje matematičnih objektov s pomočjo matrik

Marko Dolenc, Fakulteta za matematiko in fiziko, UL, marko.dolenc@amis.net

V predstavitvi si bomo ogledali kako z različnimi računalniškimi orodji izvajati transformacije na geometrijskih objektih. S pomočjo teh transformacij lahko različne like poljubno prestavljamo, jih povečujemo in zmanjšujemo po poljubni osi in rotiramo.

Transformacije opišemo z matrikami različnih dimenzij (glede na potrebe). Ker so oglišča geometrijskih objektov točke oziroma vektorji, te preslikave lahko zelo učinkovito izvajamo na računalniku. Z množenjem matrik različnih osnovnih transformacij lahko tudi enostavno verižimo več preslikav v eno kompleksnejšo preslikavo oziroma pot.

V prispevku si bomo ogledali, kako prestaviti, zavrteti ter povečati ali pomanjšati poligon s pomočjo GeoGebre. Poleg tega si bomo te operacije pogledali še v Mathematici in Matlabu. Za konec si bomo ogledali še, kako te transformacije izvesti s pomočjo JavaScript in knjižnice Math.js.

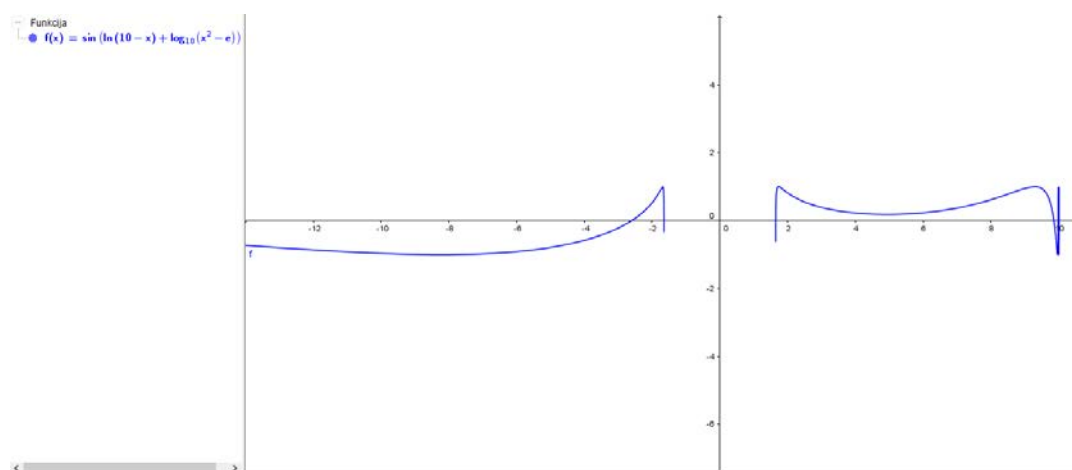


Kazalo
Program

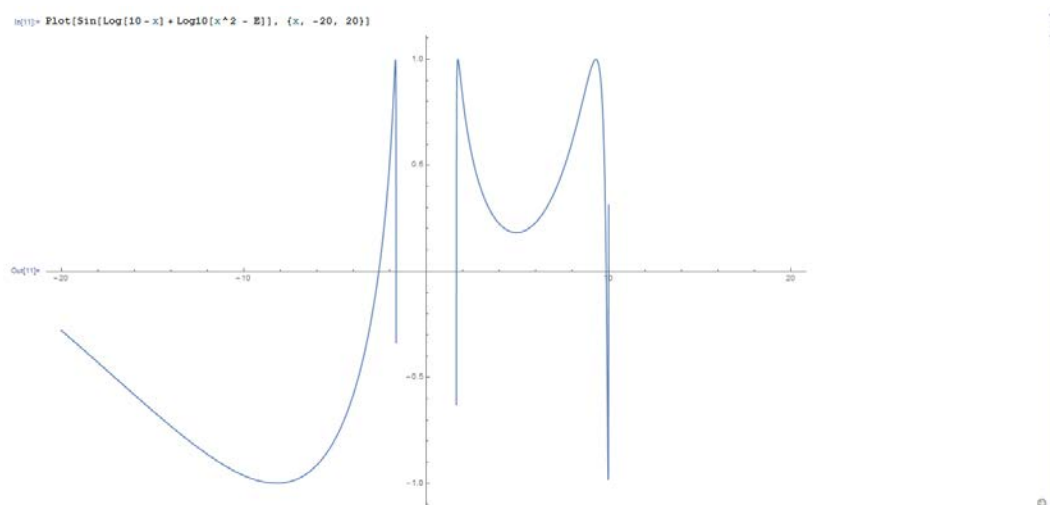
Uporaba GeoGebre in Mathematice pri reševanju matematičnih nalog zapletenih funkcij

Matej Jermančič, Fakulteta za matematiko in fiziko, UL,
matej.jermancic@student.fmf.uni-lj.si

Pri matematiki se pogosto srečujemo z risanjem in proučevanjem funkcij. Iz osnovnih lahko sestavimo poljubno elementarno funkcijo. Določene so tako spremenjene, da si že težko predstavljamo potek grafa in njene lastnosti. Tukaj nam pridejo prav računalniška orodja, s katerimi si lahko natančno narišemo graf katerekoli funkcije in ugotovimo njene značilnosti kot so ničle, poli, začetna vrednost, poli, asimptote itd. Pogledali si bomo primer reševanja matematične naloge s pomočjo GeoGebre in Mathematice. Primerjali bomo uporabnost orodij, enostavnost uporabe, način vnosa, interpretiranje vnesenih podatkov itd. Skratka lotili se bomo problema z zahtevnejšo elementarno funkcijo in primerjali katero orodje nam bolj olajša reševanje in razumevanje problema.



Slika 1 Primer reševanja v GeoGebri



Slika 2 Primer reševanja v Mathematici

Racionalne funkcije v programu Maxima

Tina Zwitter, Fakulteta za matematiko in fiziko, UL, tina.zwitter@student.fmf.uni-lj.si

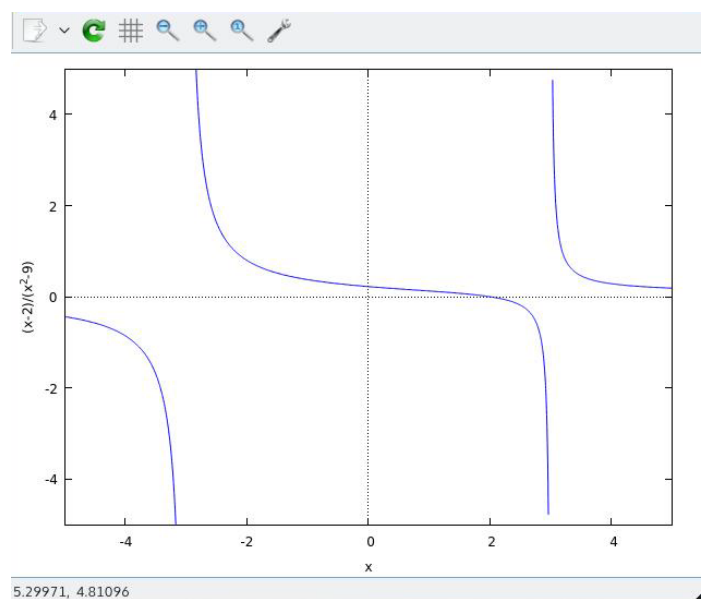
V program Maxima lahko definiramo funkcije. S prva bom definirala dva polinoma, in nato še racionalno funkcijo, ki bo definirana kot prvi polinom ulomljeno z drugim polinomom.

Izračunali bomo limito, ko gre x proti neskončnosti, da vidimo, kako se graf obnaša v neskončnosti, izračunali bomo odvod funkcije, izračunali bomo nedoločen in določen integral, narisali graf racionalne funkcije, izračunali bomo kakšno ima asimptoto.

To bom ponovila z različnimi racionalnimi funkcijami (ko je stopnja imenovalca večja od stopnje števca in obratno, ko je stopnja imenovalca enaka stopnji števca, ko sta vodilna koeficienta enaka in ko sta različna).

Na koncu bom rešila eno nalogo iz racionalnih funkcij.

```
File Edit Options Maxima Help
[Icons]
(%i1) f(x):= x-2;
(%o1) f(x) := x - 2
(%i2) g(x):=x^2-9;
(%o2) g(x) := x^2 - 9
(%i3) h(x):= f(x)/g(x);
(%o3) h(x) := f(x)/g(x)
(%i4) solve(h(x)=0,x);
(%o4) [x = 2]
(%i5) limit(h(x), x,inf);
(%o5) 0
(%i6) integrate(h(x), x);
(%o6) 5 log(x + 3) / 6 + log(x - 3) / 6
(%i7) integrate(h(x),x,0,2);
(%o7) 5 log(5) / 6 - log(3) / 6
(%i8) diff(h(x),x);
(%o8) 1 / (x^2 - 9) - 2(x - 2) / (x^2 - 9)^2
(%i9) h(5);
(%o9) 3 / 16
(%i10) plot2d(h(x), [x,-5,5],[y,-5,5]);
(%i11) plot2d(h(x), [x,-5,5],[y,-5,5]);
plot2d: some values were clipped.
(%i12)
```



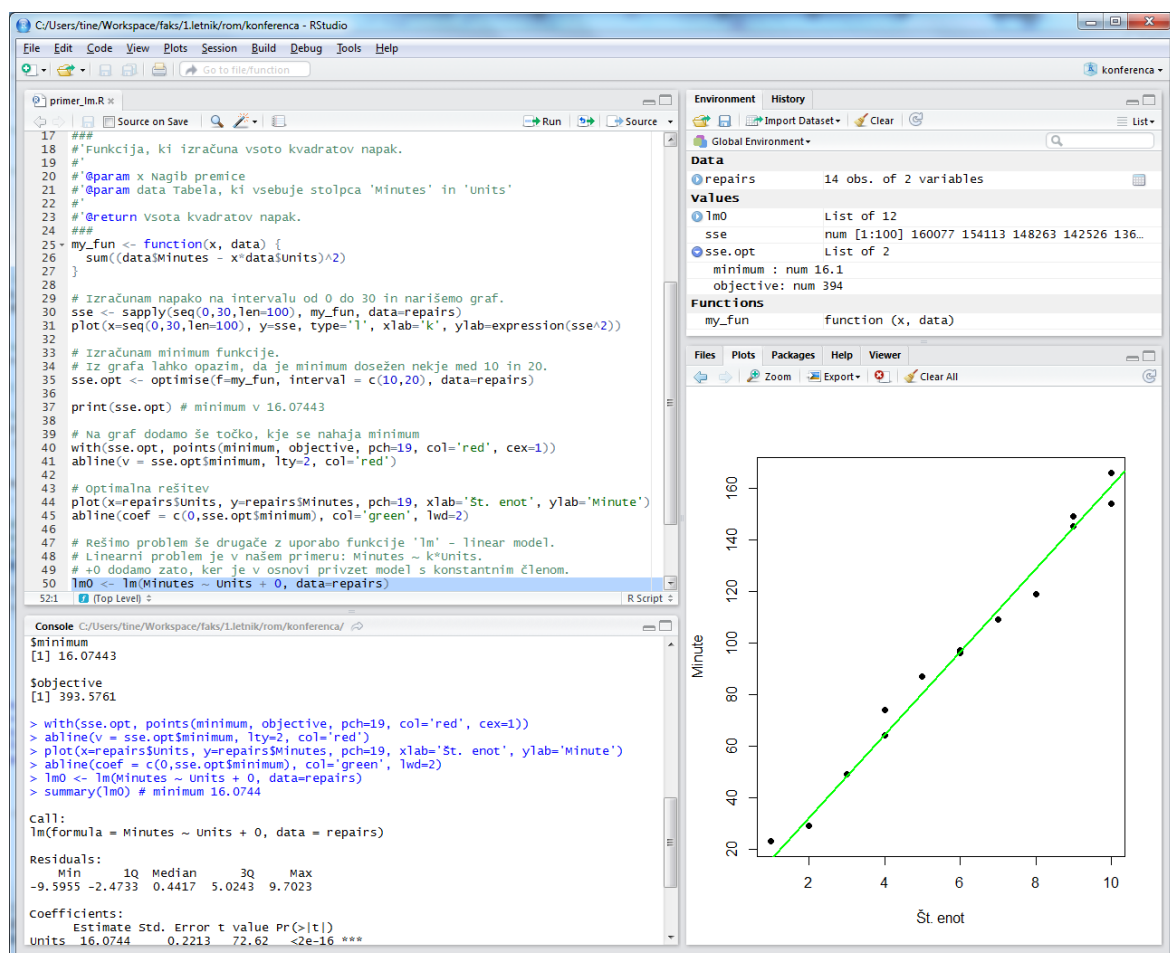
Preprosta linearna regresija po metodi najmanjših kvadratov z uporabo programa R

Tine Mlač, Fakulteta za matematiko in fiziko, UL, Tine.Mlac@student.fmf.uni-lj.si

V predstavitvi si bomo ogledali odprtokodni program R, namenjen predvsem statistični analizi in obdelavi podatkov. Za lažje delo z R-om si pomagamo z orodjem RStudio, ki nam omogoča urejanje R-ove izvorne kode v enotnem okolju (IDE).

Pogledali bomo kaj je to preprosta linearna regresija in kako jo rešujemo z uporabo metode najmanjših kvadratov.

Kot uporabo programskega jezika R-a bomo rešili enostaven primer linearne regresije. Najprej s pomočjo iskanja minimuma funkcije vsote najmanjših kvadratov nato pa bomo problem prevedli na linearni model, ga ponovno rešili in primerjali rešitvi.



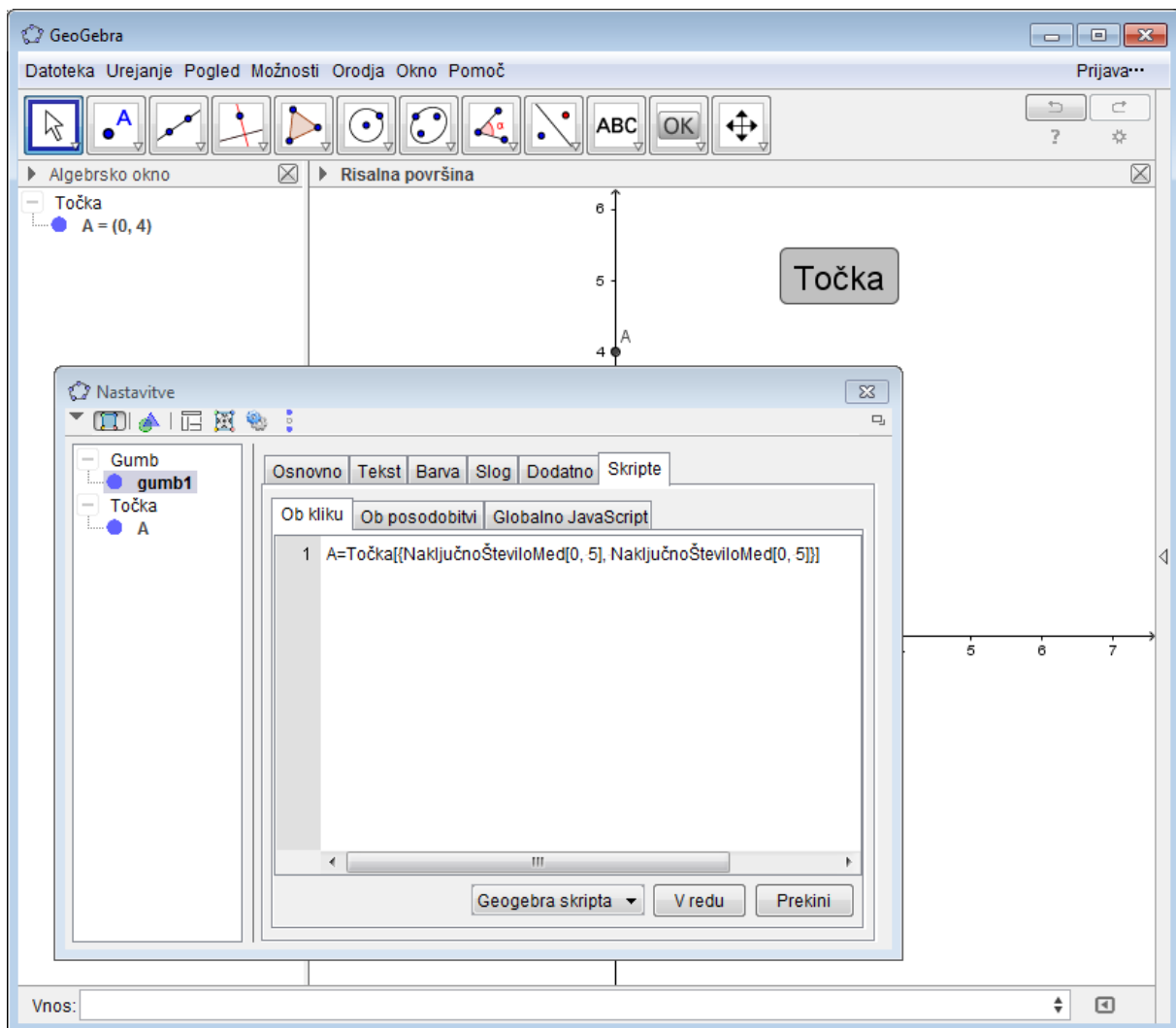
Kazalo
Program

GeoGebraScript

Tomaž Grmek, Fakulteta za matematiko in fiziko, UL, tomaz.grmek@student.fmf.uni-lj.si

GeoGebraScript oziroma GGBScript je skriptni jezik, ki je ob namestitvi že vgrajen v GeoGebro, lahko pa GGBScript uporabljamo tudi v spletni verziji GeoGebre. Omogoča nam, da lahko več ukazov izvedemo enega za drugim. Do skript dostopamo tako, da z desnim klikom kliknemo na objekt, izberemo lastnosti in izberemo zavihek Skripte. Nato izberemo zavihek s katerim določimo, ali se bodo ukazi izvedli, ko na objekt kliknemo, ali pa, ko se objekt posodobi. Uporabimo lahko ukaze, ki smo jih uporabljali v vnosni vrstici, je pa tudi nekaj ukazov, ki jih lahko uporabljamo le pri skriptiranju.

Pri predstavitvi bom na več konstrukcijah prikazal osnove GGBScripta.



[Kazalo](#)
[Program](#)

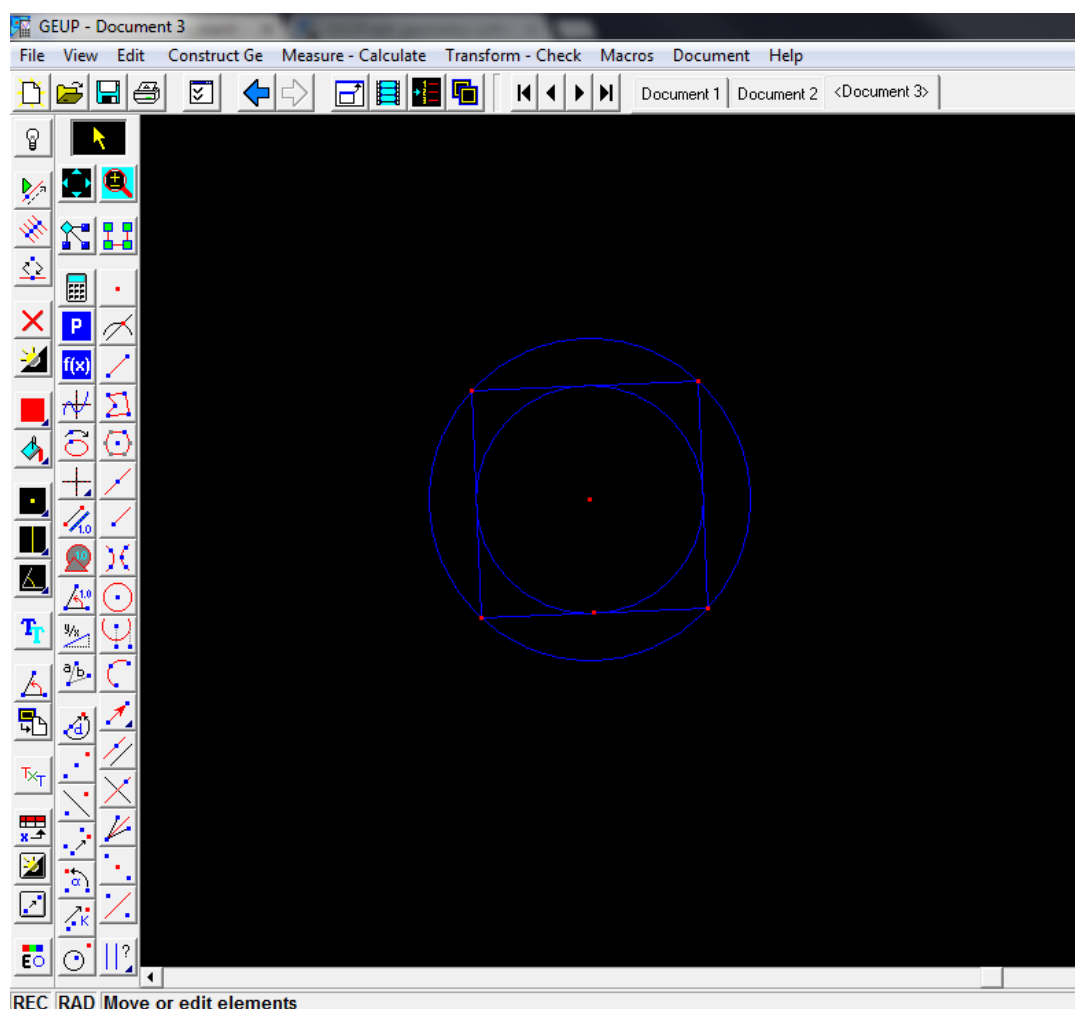
GEUP 7 - program za pomoč pri geometriji

Žan Jereb, Fakulteta za matematiko in fiziko, UL, zan.jereb@student.fmf.uni-lj.si

GEUP 7 je interaktivno geometrijsko orodje. Omogoča vizualno kreiranje geometrijskih elementov (vektro, liki, ...) in računanje z njimi. To orodje se ne uporablja samo za geometrijo ampak z njim si lahko pomagamo tudi pri kemiji, strojništvu in pri industrijskem oblikovanju.

GEUP 7 je še kar preprost program za uporabo, saj ne potrebujemo nobene kode, ampak klikamo na ikone in potem na prostor, ki je podan. Zanj ne potrebujemo predznanja drugih orodij. Orodje je zelo podobno GeoGebri, saj imata podobne funkcije in podoben namen za uporabo.

V predstavitvi si bomo ogledali osnovne funkcije GEUP 7 in tudi nekaj rešenih primerov, ki bodo tudi podani kot nerešeni.



[Kazalo](#)
[Program](#)

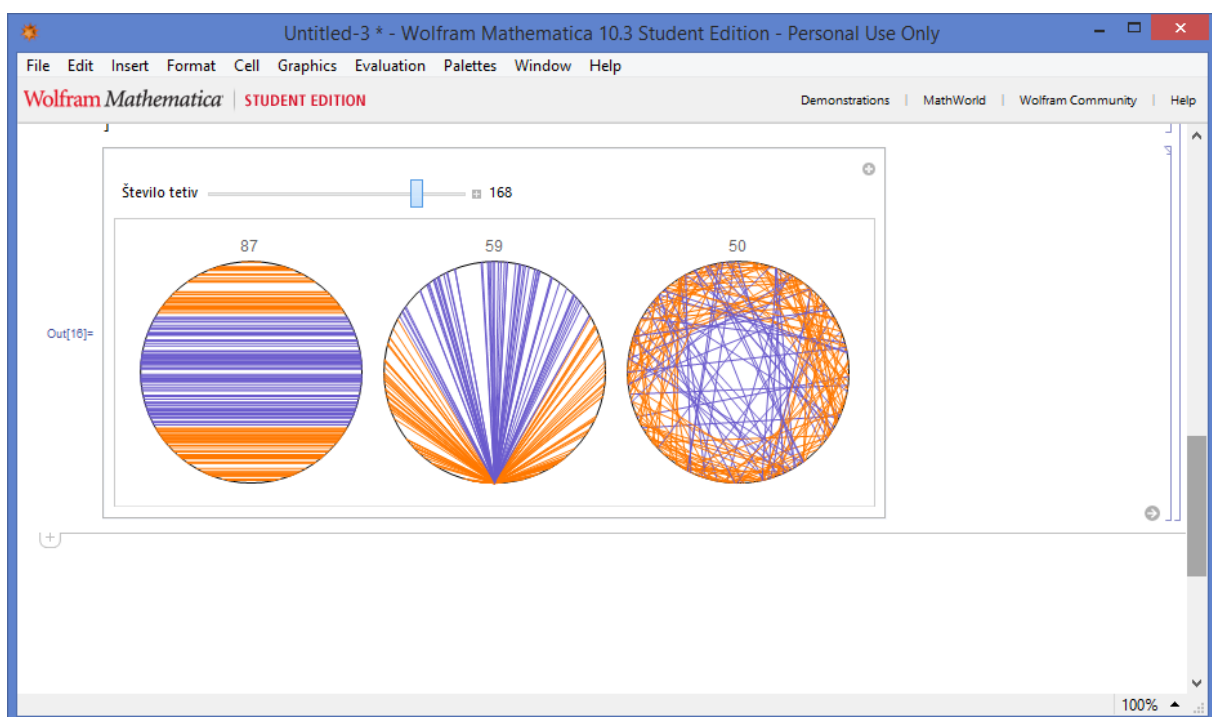
Bertrandov paradoks

Žiga Mrzljak, Fakulteta za matematiko in fiziko, UL, Ziga.Mrzljak1@student.fmf.uni-lj.si

Predstavil bom problem, ki se pojavi v teoriji verjetnosti (verjetnostni račun), ko metoda pri določanju naključne spremenljivke ni jasno določena. Bolj natančno si bomo ogledali različne primere določanja verjetnosti nekega dogodka, kjer pa vsi dajo drugačen (pravilen) rezultat.

Konkretno si bomo ogledali problem, ki ga je matematik Joseph Bertrand zastavil leta 1889: Imamo enakostranični trikotnik, ki mu očrtamo krožnico. Nato v tem krogu narišemo naključno izbrano tetivo. Kakšna je verjetnost, da je izbrana tetiva daljša od stranice včrtanega trikotnika?

Problem bom predstavil s pomočjo programa Mathematica.



[Kazalo](#)
[Program](#)

PROGRAM

1. DAN PONEDELJEK, 15.2.2016

8.30-8.40 Uradni nagovor

8.40-9.50

[Desmos](#) (Lea Kos)

[Geogebra Graphing Calculator](#) (Gaja Rihar)

[MalMath](#) (Erika Šavli)

[Calculator++](#) (Erika Medoš)

9.50-10.30

[Reševanje rekurzivnih enačb](#) (Jakob Valič)

[Spletna enciklopedija celoštevilskih zaporedij](#) (Jan Tomšič Pivk)

[Reševanje diofanskih enačb z računalnikom](#) (Samantha Slaček)

10.30-10.50

ODMOR

10.50-11.30

[Cymath](#) (Katja Bukovec)

[Epski krogi](#) (Gregor Vavdi)

[GraphTea](#) (Anej Jereb)

11.30-12.10

[Dolžina slovenske obale](#) (Blaž Poljanec)

[Fermatova točka](#) (Darko Petrović)

[Permutacije](#) (Rastko Veriš)

12.10-12.30

ODMOR

12.30-13.20

[Primerjava uporabe GeoGebre in Mathematice pri reševanju matematičnih nalog zapletenih funkcij](#) (Matej Jermančič)

[Symbolab](#) (Marina Kovač)

[Geometrija v Mathematici](#) (Lara Jerač)

[Maxima](#) (Borna Kavčič)

13.20-14.00

[Cinderella](#) (Karin Stančič)

[Pascalov trikotnik](#) (Nikoleta Krstić)

[Geogebra, Matlab, Mathematica pri reševanju izpita iz Algebre](#)

(Martin Česnovar)

2. DAN TOREK, 16.2.2016

8.30-9.20

[GEUP7](#) (Žan Jereb)[Delo z racionalnimi funkcijami v Maximi](#) (Tina Zwitter)[Cyclogoni](#) (Jaka Pivk)

9.20-9.40

ODMOR

9.40-10.30

[Preprosta linearna regresija po metodi dveh kvadratov z uporabo programa R](#) (Tine Mlač)[GeoGebraScript](#) (Tomaž Grmek)[Graph](#) (Keni Šuligoj)[Bertrandov paradoks v verjetnosti](#) (Žiga Mrzljak)

10.30-11.30

[RoboCompass](#) (Anja Trop)[Teorija grafov](#) (David Pančič)[Microsoft Mathematics](#) (Lea Temlin)[Uporaba knjižnice NumPy pri problemih iz Linearne Algebre](#)

(Larisa Carli)

11.30-11.50

ODMOR

11.50-12.50

[Metoda Monte Carlo](#) (Marko Jereb)[Različni načini dokazovanja Pitagorovega izreka](#) (Alenka Kežar)[Rotacije, translacije, skaliranje matematičnih objektov s pomočjo matrik](#) (Marko Dolenc)[Dokazovanje Eulerjeve premice s pomočjo podobnih trikotnikov in trilinearne koordinatnega sistema](#) (Jure Srabotnik)