



СУГС “Георги Димитров” - Скопје

# **Приказ на извод на функција со помош на динамички софтвер**

Изработил:  
Давид Гичев IV<sup>1</sup>

Ментор:  
Јасмина Маркоска

Скопје, 2019

## Вовед

---

Толкувањето на извод како стрмност на функција е интуитивно и визуелно разбирливо. Терминот стрмност потсетува на планинарење, од каде следува следната паралела: Еден пат по планина може да се следи како график на функција. Планинарот има  $x$  и  $y$  координата, каде  $y$  е функција од  $x$

(сл.1). Аналогно, правецот на движењето по планината/патот наликува на тангента која се движи по крива. Чувството на планинарот за стрмност во одредена точка од патот, соодветствува на коефициентот на правец на тангентата. Висината на планинарот може да се определи релативно или апсолутно. Имено, релативна би била висината во произволна точка во споредба со висината која ја има во почетната



Слика 1: Графички приказ на искачување по планина

позиција. Апсолутната висина би се определила според надморската висина на која се наоѓа планинарот во даден момент. Очигледно е дека стрмноста не зависи од начинот на определување на висината. Во кој било момент од качувањето, планинарот ја споредува моменталната висина со висината на околната површина. Според тоа, изводот на функцијата не зависи од апсолутната висина на функцијата, туку од множеството точки кои ја опкружуваат точката која ја разгледуваме. Конкретно, константите собироци во една функција не влијаат на нејзиниот извод. При определување на моменталната висина, планинарот го разгледува околниот релјеф. Нека видното поле на планинарот е ограничено на патот кој е зад него. Со други зборови, планинарот оди напред додека гледа назад. Во тој случај, стрмноста на патеката во даден момент ја определува гледајќи ги поминатите делови од патот што се најблизу до него. Аналогно, определување на лев извод од функција во дадена точка се постигнува со споредување на вредноста на функцијата во дадената точка, со вредностите на точките што се лево од неа. Истото важи и за определувањето на десниот извод, кој за планинарот е поприроден.

Во овој труд е изложено упатство за изградба на динамички аплет во програмата GeoGebra, со чија помош се доловува чувството за стрмност на една функција. Истовремено, интуицијата зад користењето на лимес кај диференцијалното сметање е визуелно прикажана.

## Наредби во GeoGebra кои ќе бидат користени

---

**Point(<List>)** - точка

List - листа од два елемента, т.е. координати на точката

**Segment(<Point A>,<Point B>)** - отсечка што поврзува две точки

**Sequence(<Expression>,<Variable i>,<Start Value a>,<End Value b>,<Increment>)** - низа

Expression - наредбата која се извршува во секој циклус

Variable i - променливата што ќе се менува со секој чекор од циклусот

Start Value a - почетна вредност на i

End Value b - крајна вредност на i

Increment - чекорот што го зазема i по извршување на еден циклус

**<Sequence niza>(<Index n>)** - n-ти елемент од низата niza,  $n \in \mathbb{N}$

**x(<Point>)** - x координата на точка (за y координата: **y(<Point>)**)

**Length(<Sequence>)** - број на елементи во дадена низа

**Line(<Point A>, <Point B>)** - права повлечена низ две точки

**If(<Condition>, <Expression A>, <Expression B>)**

Condition - услов, односно логички израз кој дава вредност точно или неточно како повратна вредност

Expression A - наредба што ќе се изврши доколку изразот е точен

Expression B - наредба што ќе се изврши во случај изразот да не е точен

Динамичкиот аплет кој е објаснет во продолжение, претставува пакет од повеќе интерактивни анимации. Секако, истите се меѓусебно поврзани, но постои опција само една од нив да биде прикажана во даден момент. Истото овозможува пренасочување на фокусот од еден кон друг дел од интуицијата за извод на функција. Целокупната функционалност е сместена во еден аплет, достапна преку checkbox-ови и поместувачи. Готова/Завршена верзија од истиот аплет е достапна на следниот линк: <https://www.geogebra.org/graphing/y6kmbk9m>

Програмата се состои од 3 дела (прикази):

-премин на секанта во тангента

-лимес-стил апроксимација на функција и извод на истата

-изводот како лимес од односот на растот на функцијата и растот на аргументот

Програмата ќе ја покажува функцијата во интервал од -10 до 10. Секако, ова може да се менува (на пример динамички, со поместувач).

## Пишување на кодот за аплетот

---

За добивање на низа од точки кои се апроксимација за функцијата  $f(x)$ , наредбата за низа изгледа вака:

$$Tochki = Sequence(Point(\{i, f(i)\}), i, -10, 10, 1/freq)$$

Каде  $freq$  е променлива чија вредност го кажува бројот на точки чии  $x$  координати се помеѓу секои два цели броеви. Потребно е истите да бидат меѓусебно поврзани, за таа цел се прави низа од отсечки:

$$Otschki = Sequence(Segment(Tochki(i), Tochki(i + 1)), i, 0, 20 * freq)$$

Притоа,  $20 * freq$  е бројот на точки кои ги користиме за апроксимација, истовремено број на елементи на низата  $Tochki$ . Замена за овој израз е  $Length(Tochki)$

За пресметување на (десен) извод во дадена точка се разгледува нејзината соседна точка. Конкретно, за дадена точка со координати  $(x_1, f(x_1))$  и нејзина соседна точка  $(x_2, f(x_2))$ , изводот во првата се апроксимира како однос од прирастите на вредноста на функцијата и аргументот.

$$f'(x_1) \approx \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

Запишано во GeoGebra, со користење на множеството од точки ( $Tochki$ ) предходно дефинирано, изводот се дефинира на следниот начин:

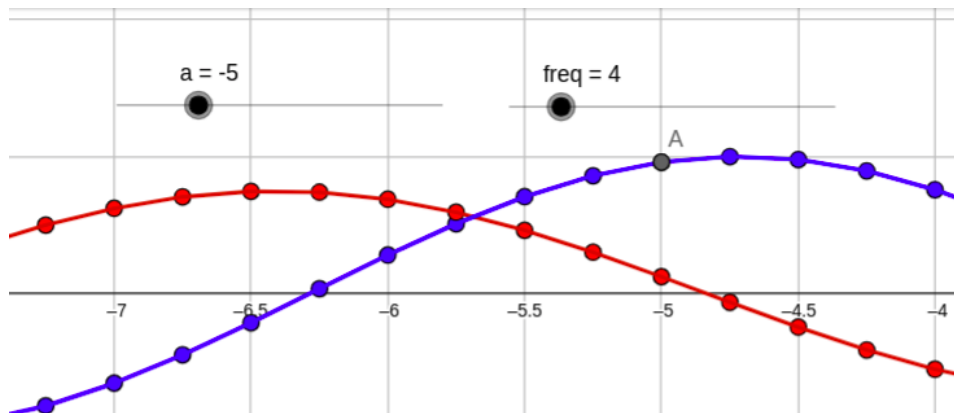
$$Izvod\_vo\_tochka = \frac{y(Tochki(2)) - y(Tochki(1))}{x(Tochki(2)) - x(Tochki(1))}$$

Истото е направено за секоја точка од  $Tochki$ :

$$Izvod = Sequence(Point(\{ x(Tochki(i)), ( y(Tochki(i + 1)) - y(Tochki(i)) ) / ( x(Tochki(i + 1)) - x(Tochki(i)) \}), i, 0, 20 * freq)$$

За да се поврзат секоја со секоја од овие точки се постапува исто како кај *Otsechki*  
 $Izvod\_otsechki = Sequence(Segment(izvod(i), izvod(i + 1)), i, 0, 20 * freq)$

Моментален изглед на програмата е прикажан на сл. 2.



Слика 2: Моментален изглед на програмата (заедно со поместувачи)

Следно, за приказот на тангентата се користи функција:

$$p(x) = y(izvod(indeks)) * (x - a) + y(izvod(indeks))$$

Каде  $a$  е апцисата на точката во која тангентата ја допира [апроксимацијата на] кривата, додека

$$index = (a + 10) * freq + 1$$

Притоа, за да се добие индексот на елементот во низата што одговара на точката со апциса  $a$ , се додава 10, па се множи изразот со  $freq$  за да се опфатат сите ненулни броеви, конечно се додава 1 за нулата.

Следен дел од оваа програма е приказ на преминот од секанта во тангента. За таа цел, се додава checkbox (*sekanta*) чија вредност менува дали изводот се определува помеѓу две соседни точки, или на повеќе. Растојанието помеѓу точките од кои ќе се гледа тангентата, воедно изводот, е контролирано со  $h$ . Минимална вредност на  $h$  е  $1/freq$ .

Тангентата се црта со наредбата:

$$tangenta(x) = If(sekanta, s, p)$$

Каде  $p$  е претходно-дефинираната функција за тангента, додека

$$s = Line(Point(\{a, f(a)\}), Point(\{a + h, f(a + h)\}))$$

Алтернативно, наместо да се внесуваат двете точки директно во наредбата, може надворешно да бидат дефинирани:

$$A = Point(\{a, f(a)\})$$

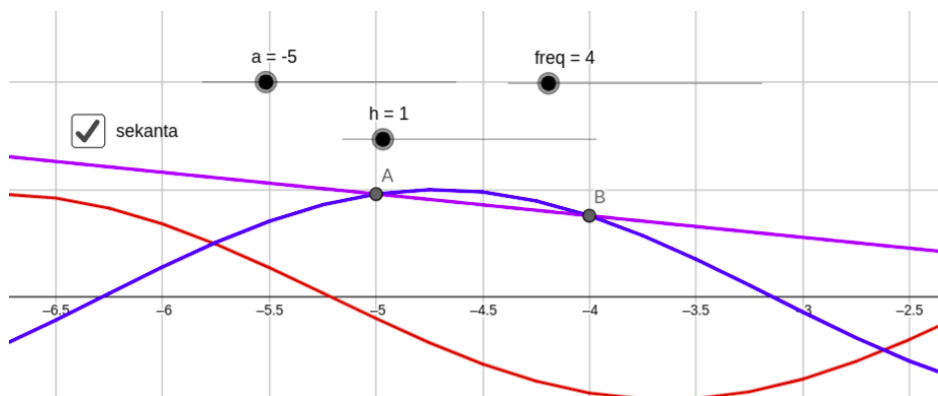
$$B = Point(\{a + h, f(a + h)\})$$

Притоа, кај Settings на B, во пределот Advanced, потребно е да се внесе “sekanta” во полето “Condition to show object”

Следува модификација на изразот за низата Izvod:

$$Izvod = Sequence(Point(\{ x(Tochki(i)), (y(Tochki(i + h * freq)) - y(Tochki(i)))/h\}), i, 0, 20 * freq)$$

На сл. 3 е прикажан моменталниот изглед на програмата.



Слика 3: По додавањето на функционалноста за секанта (прикажани се само отсечките)

Додатна опција која може да се додаде е autoMove, уште еден checkbox. Во делот Settings на променливата  $a$ , постои опција да се извршува наредба секој пат кога вредноста на  $a$  ќе се смени (Settings > Scripting > On Update). Во полето за внес се вметнува следната линија код:

$If(autoMove, CenterView((a + h/2, 0)))$  и со ова, екранот се движи заедно со точките

Пожелно е графикот од апроксимацијата на изводот на  $f(x)$  да биде покажан до точката која ја разгледуваме. Поточно, интервалот на кој изводот ќе биде дефиниран доколку се намали на  $[-10, a]$ , приказот ќе биде почист. За да се постигне ова треба да се дефинира нова функција:

$trace = Sequence(Izvod\ otsechki(i), i, 1, index - 1)$

Притоа, наредбата е всушност одделување на сите елементи кои се пред елементот кој одговара на моменталната вредност на  $a$

Последниот дел од оваа програма го анализира изводот како однос помеѓу прирастот на вредноста на функцијата и прирастот на аргументот. Се додава checkbox (triagolnik), кој одредува дали триаголникот, кој ги поврзува двете точки, е видлив или не.

$Pravoagolen\_triagolnik = Polygon((a+h, f(a)), (a+h, f(a+h)), (a, f(a)))$

Или, во случај A и B да се дефинирани посебно ( $A = \{a, f(a)\}$ ,  $B = \{a+h, f(a+h)\}$ )

$Pravoagolen\_triagolnik = Polygon((x(B), y(A)), A, B)$

Се препорачува да се дефинира аголот што лежи кај темето A

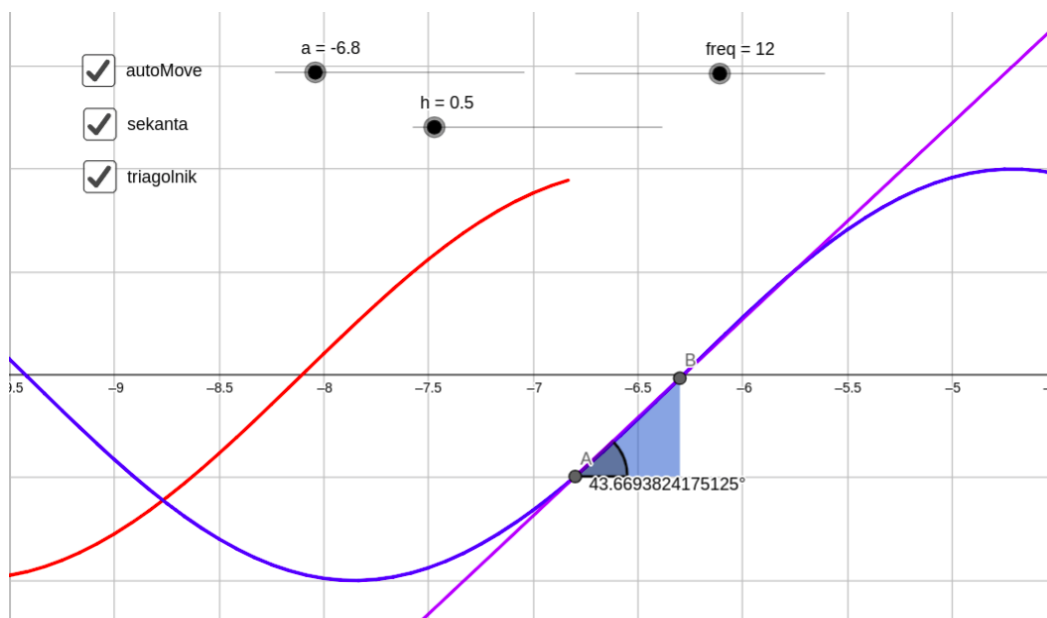
$$\text{Agol} = \text{Angle}((x(B), y(A)), A, B)$$

(Во Settings на аголот, интервалот треба да биде намален на  $0^\circ$ - $180^\circ$ )

Исто, во поместувањата на `pravoagolen_triagolnik`(фигурата) и `agol`, треба во пределот

Advanced > Condition to show object, да се внесе “`triagolnik`”. Пожелно е во истиот предел кај `triagolnik` (checkbox-от), да се внесе “`sekanta`”.

Конечниот изглед на програмата е прикажан на сл. 4. Примери за други функции кои можат да се разгледуваат со овој аплет се прикажани во галеријата на сл. 5 - сл. 9.



Слика 4: Финален изглед на програмата

## Заклучок

---

Геогбра како развојна околина нуди алатки преку кои се гледа нејзиниот потенцијал за употреба. Има верзии достапни за било кој оперативен систем, како и веб-верзија. Распространетоста и фактот што е бесплатна и едноставна за користење, ја прави оваа програма погодна за професори, студенти, па и ученици. Создавањето проекти преку оваа програма, ги прави истите лесно преносливи (во форма на линк или фајл), последователно забрзувајќи го целиот процес на нивно ширење. Инкорпорирање на активности кои вклучуваат визуелни елементи, ја збогатува наставата и дава осет за длабочината на математиката. Идејата за диференцијално сметање најлесно е изложлива геометриски. Секако, има повеќе начини за тоа, оваа програма е само еден пример.

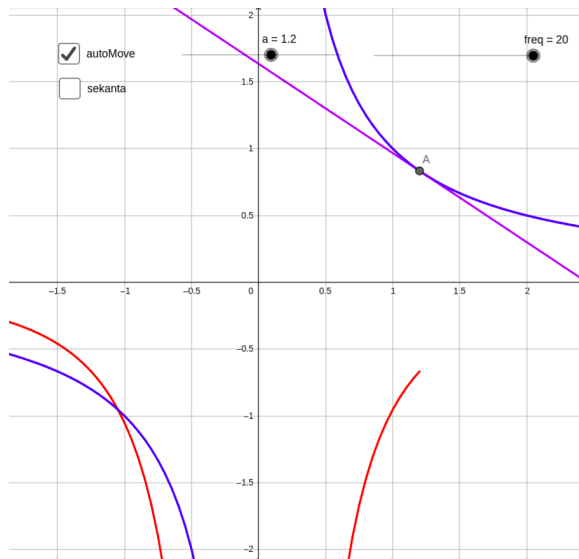


## Користена литература

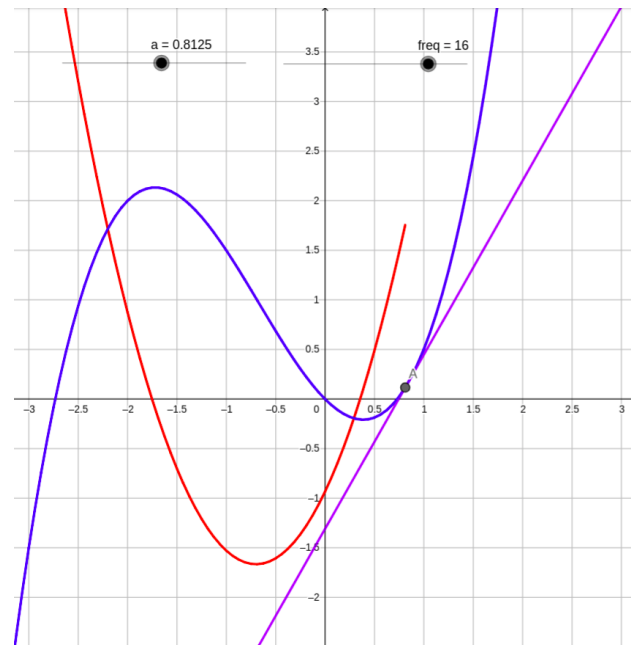
---

1. Упатство за користење на Geogebra (<https://wiki.geogebra.org/en/Manual>)

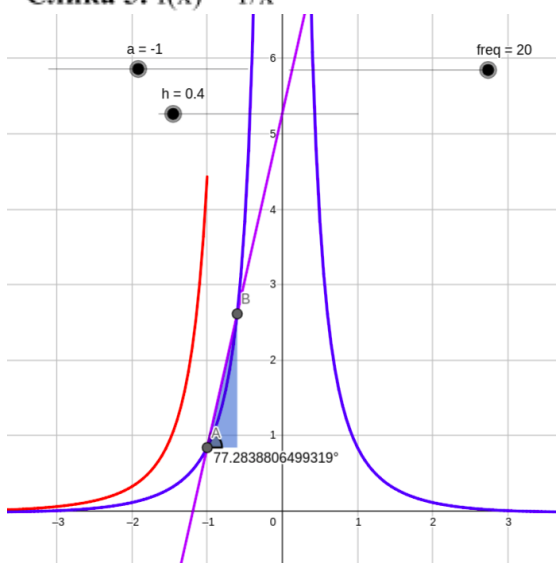
# Галерија



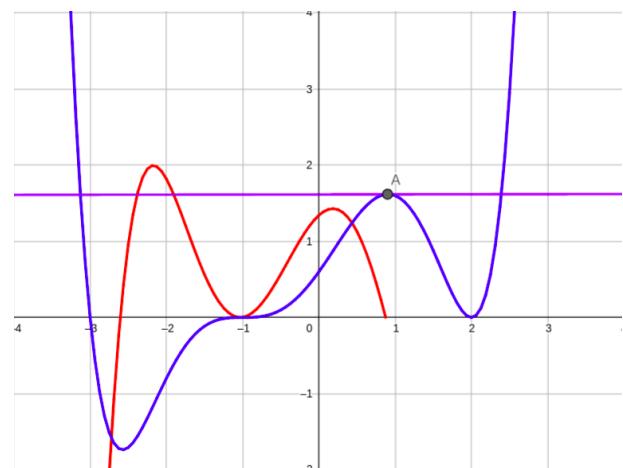
Слика 5:  $f(x) = 1/x$



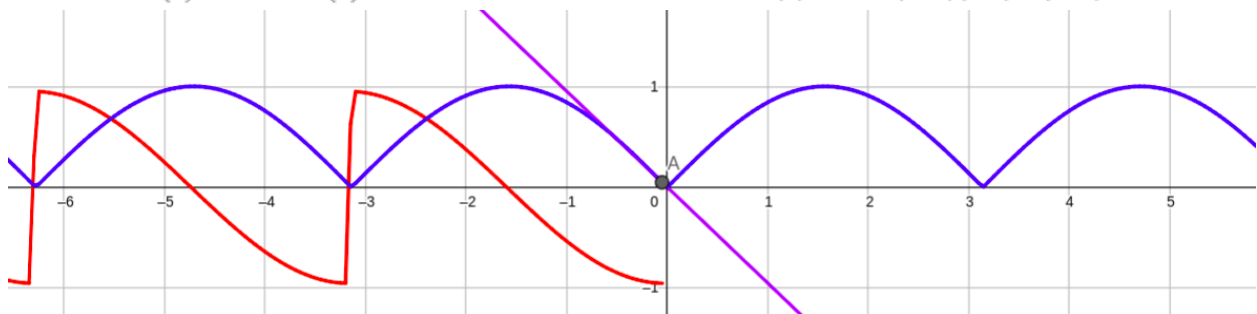
Слика 6:  $\frac{1}{3}x^3 + x^2 - x$



Слика 7:  $f(x) = 1/x^3 \sin(x)$



Слика 8:  $f(x) = 1/20 (x+3)(x-2)^2(x+1)^3$



Слика 9:  $f(x) = |\sin(x)|$