Универзитет у Београду

Електротехнички факултет

Одабрана поглавља нумеричке анализе

A picture containing text, watch

Description automatically generated

Пројектни задатак 2

Штурмова теорема

Студент:

Александра Богићевић 0390/17

Београд, школска година 2021/2022

**Задатак:**

Нека је дат реални полином *P(x)*над реалним сегментом [a, b].

1. А) Одредити Еуклидовим алгоритмом највећи заједнички делилац

*G(x) = GCD(P(x), P’(x))*

Б) За полином *P(x)= P(x)/ G(x)*, употребом Штурмове теореме, одредити број нула на [a, b].

1. Примена Штурмове теореме.

Нека је *k* број децимала на који се заокружује неки реалан број. За полином

са реалним коефицијентима

одредити низ наниже заокружених рационалних коефицијената

осређених по следећим правилима:

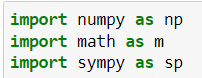
* Ако је тада ;
* Ако је тада где је навише заокружена цифра (уз евентуално последично заокруживање и претходних цифара за један број навише).

Одредити процедуру за формирање рационалног полинома

Нека је [a, b] сегмент са рационалним рубним тачкама. Наћи такав полином P(x) са реалним коефицијентима и број k, да на основу позитивности полинома T(x) над сегментом [a, b] имамо доказ о позитизивности полинома P(x) над сегментом [a, b].

**Напомена**:

За рачунање решења коришћен је програмски језик *Python* и његове библиотеке *numpy,math* и *sympy*. Радно окружење је *Jupуter notebook*. На дну документа се налази прилог са целокупним кодом.



**Решење (1):**

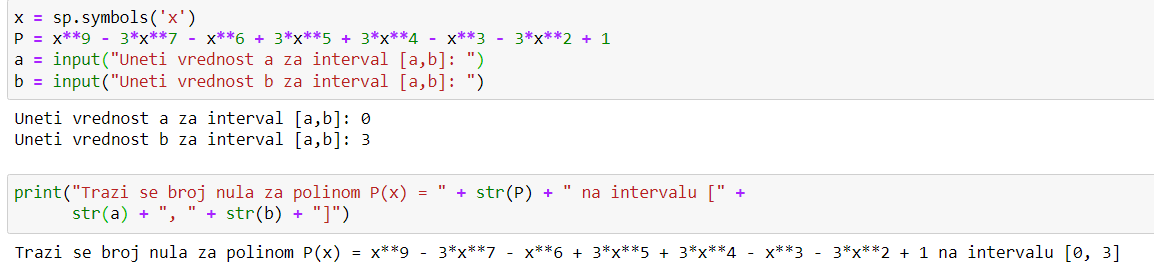
Пре свега треба дефинисати полином P(x), као и интервал [a, b]. Полином је кодован у програму, док се интервал уноси са стандардног улаза. Посматраћемо пар примера.

**Пример 1**

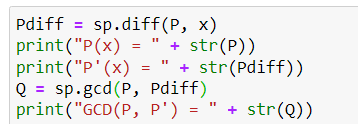
За вредности интервала уносимо вредности:

a = 0

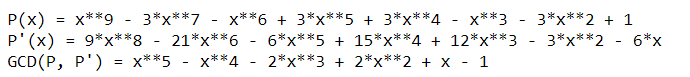
b = 3



Ставку (А) решавамо позивањем уграђене функције за тражење највећег заједничког делиоца (НЗД у наставку) за полином Р(х) и Р’(x). Добијена вредност се смешта у променљиву Q.



Излаз програма:



Добивши НЗД полинома Р(х) и Р’(x) можемо да почнемо решавање ставке (Б).

Први корак Штурмове теореме је формирање низа полинома

*Р0(х), Р1(х), Р2(х),...,Рi(x)*

као

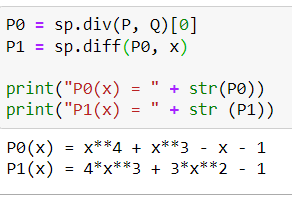
*P0(x) = P(x)/Q(x),*

*P1(x) = P’(x),*

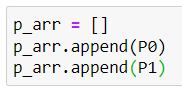
*Pi+1(x) = −REM(Pi(x), Pi−1(x))* редом за *i = 1, 2, . . . , r−1* и *Pr(x) = C– Const.*

Почетни полином Р0(х) израчунавамо као количник полинома Р(х) и Q(х).

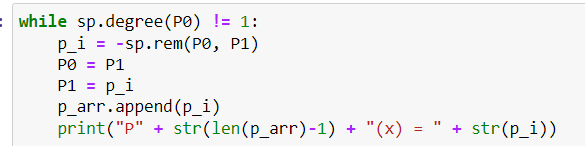
Полином Р1(х) израчунавамо као први извод полинома Р0(х).



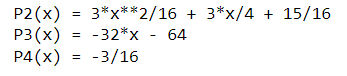
Добијене полиноме треба памтити како бисмо их касније користили, тако да уводимо низ полинома у којем чувамо претходно добијене полиноме Р0 и Р1.



Наставак израчунавања полинома може се представити једном петљом.



Након извршене петље добијамо следеће:



Овиме смо добили цео низ полинома и можемо прећи на наредни корак.

Следећи корак Штурмове теореме гласи:

Нека је *ξ ∈* [*a, b*] означимо *V* (*ξ*) број промена знакова у низу *P*0(*ξ*)*, P*1(*ξ*)*, . . . , Pr*(*ξ*), игноришући евентуално јављање корена полинома у том низу. Тада разлика

*N* = *V* (*a*) *− V* (*b*)

oдређује број нула полинома Р(х) на сегменту.

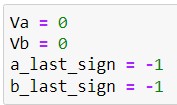
Другим речима:

За сваки полином у запамћеном низу полинома Р0(х), Р1(х), Р2(х),...,Рi(x) израчунаваћемо вредност за х=а и израчунати број промена знакова. Резултат ћемо сместити у променљиву Va. Исто то ћемо урадити и за х=b и резултат ћемо сместити у променљиву Vb.

Да бисмо добили број нула почетног полинома израчунаћемо разлику Va и Vb, и то сместити у променљиву N и тиме добити коначно решење.

У коду је то одрађено на следећи начин:

Уводимо нове променљиве

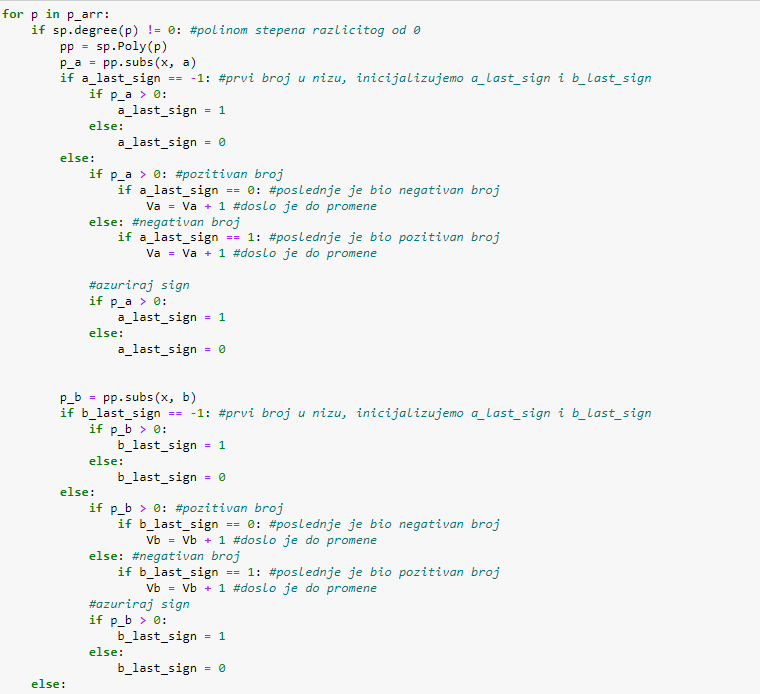


при чему Va и Vb представљају број промена знакова, и иницијално су постављени на нулу.

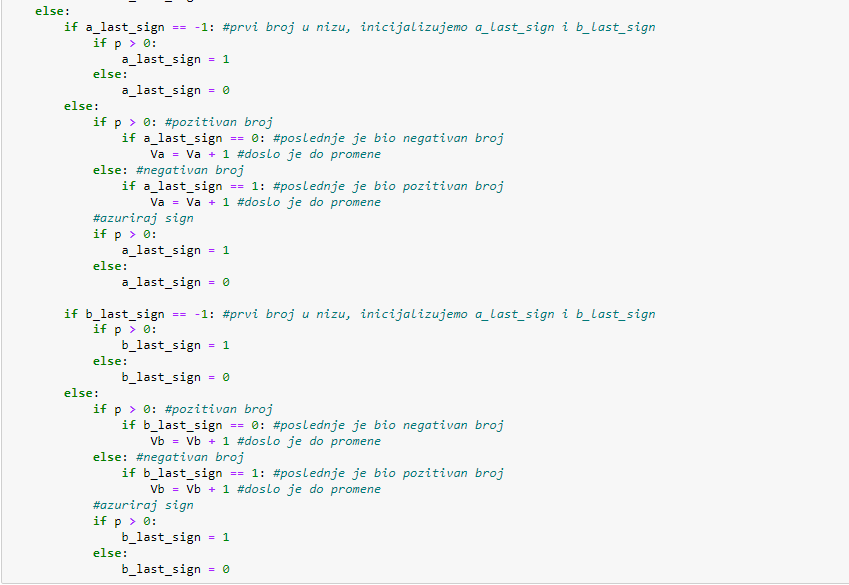
Променљиве a\_last\_sign и b\_last\_sign су помоћне променљиве за гледање да ли је дошло до промене знака или није. Оне означавају ког знака је последње израчунати полином. Вредност -1 означава иницијалну вредност, тј. да још увек ни један полином није израчунат. Уколико се добије негативна вредност полинома за х=а, а\_last\_sign се поставља на 0, а уколико се добије позитивна вредност, а\_last\_sign се поставља на 1. Исто важи и за х=b и b\_last\_sign.

При израчунавању сваког полинома пореди се његов знак са знаком претходно израчунатог полинома. Уколико је дошло до промене знака, увећава се променљива Va (или Vb) и ажурира се a\_last\_sign (или b\_last\_sign). Уколико није дошло до промене знака променљиве Va и Vb остају непромењене.

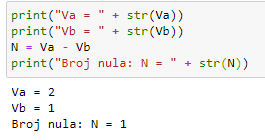
Да би замена х=а или х=b уопште била могућа, потребно је да степен полинома буде различит од нуле.



Уколико је степен полинома једнак нули, не врши се замена већ се само посматра знак и ажурирају вредности уколико је потребно.

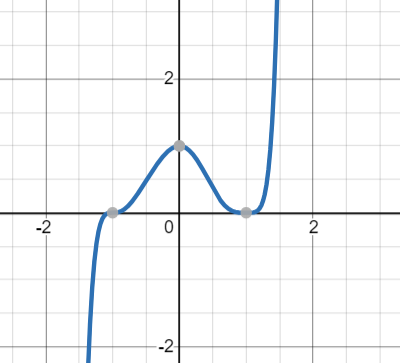


Када смо израчунали вредности Va и Vb можемо да израчунамо и коначни резултат N.



***Провера решења***:

Нуле полинома можемо уочити на следећем графику



Као што се може приметити, на интервалу [0, 3] заиста постоји једна нула.

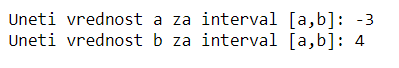
**Пример 2**

За вредности интервала уносимо вредности:

a = -3

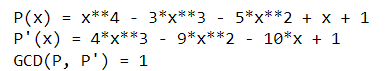
b = 4

***Излаз програма***:

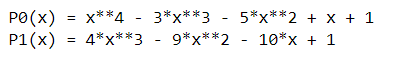


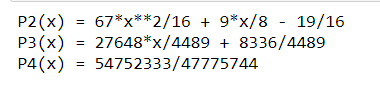


Тражење НЗД:

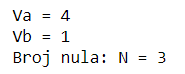


Формирање низа полинома:



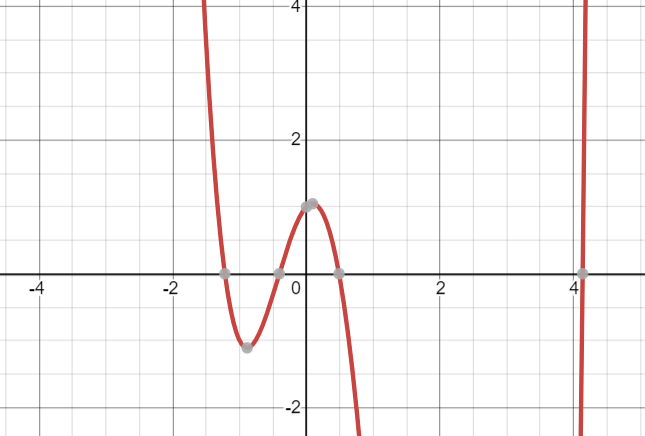


Број нула:



***Провера решења***:

Нуле полинома можемо уочити на следећем графику



Као што се може приметити, на интервалу [-3, 4] заиста постоје три нуле.

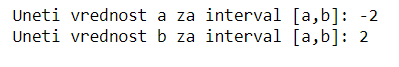
**Пример 3**

За вредности интервала уносимо вредности:

a = -2

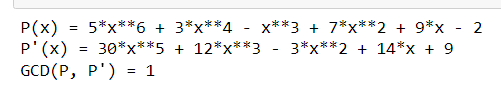
b = 2

***Излаз програма***:

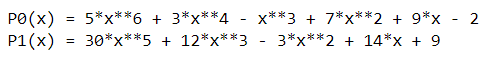


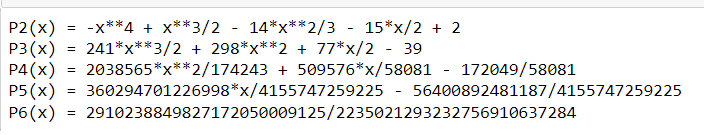


Тражење НЗД:

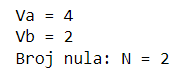


Формирање низа полинома:



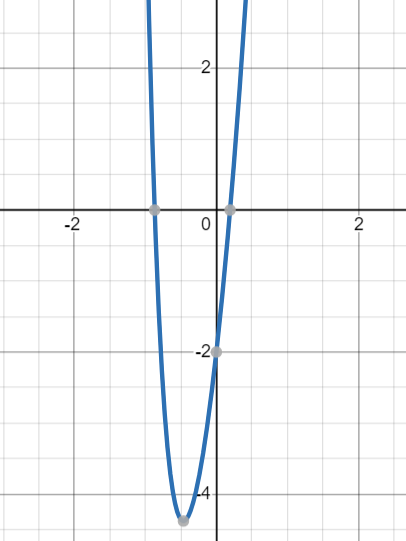


Број нула:



***Провера решења***:

Нуле полинома можемо уочити на следећем графику



Као што се може приметити, на интервалу [-2, 2] заиста постоје две нуле.

**Решење (2):**