Autor: Paweł Salwa

Zaimplementowałem 3 metody: rownego podzialu(bisection), siecznych (secant) i regula falsi. Okazuje się, że najszybciej wykonał się algorytm metody siecznych, wolniejsza była metoda regula falsi, a najwolniejsza była metoda równego podziału

Równanie charakterystyczne -1. + 39.*x + 4.*x*x - x**3 do macierzy z zad8 uzyskałem szybko i łatwo dzięki wolframalpha.com

Algorytmy te mają złożoności obliczenowe liniowe O(n).

Output:

```
wynik metody rownego podzialu:====================
-4.57408722583
0.0255743726157
8.54851285322
7.60681430165e-05 <-czas wykonania bisection()


wynik metody siecznych:====================
-4.57408722583
0.0255743726343
8.54851285322
3.3060385249e-05 <-czas wykonania secant()


Wynik metody regula falsi:====================
-4.57408722583
0.0255743724552
8.54851285321
4.09597693078e-05 <-czas wykonania regula()
```

Kod w pythonie (Salwa10.py):

```
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 06 00:51:34 2017

@author: salwus
"""

import timeit
from bisection import bisection
from regula import regula
from secant import secant

print "wynik metody rownego podzialu:===================="
print timeit.timeit( bisection, number=1 )  ,"<-czas wykonania bisection()"
```

```
print "\n\nwynik metody siecznych:====================="
print timeit.timeit( secant, number=1 )  ,"<-czas wykonania secant()"


print "\n\nwynik regula falsi:====================="
print timeit.timeit( regula, number=1 )  ,"<-czas wykonania regula()"
```

Kod w pythonie (bisection.py):

```
# -*- coding: utf-8 -*-
"""
Created on Sun Feb 05 21:37:30 2017

@author: salwus
"""

def bisection():
	a,b,c,d,e,f=-5.,-4.,0.,1.,8.,9.
	print bis(a,b)
	print bis(c,d)
	print bis(e,f)


def bis(a,b):
	fb=-1. + 39.*b + 4.*b*b - (b**3)

	while(True):
		x=(a+b)/2
		fx=-1. + 39.*x + 4.*x*x - x**3

		if(fx==0):
			break
		elif(fx*fb<0):
			a=x
		else:
			fb=fx
			b=x
		if(abs(fx) <= 0.00000001):#dokladnosc
			break
	return x
```

Kod w pythonie (regula.py):

```
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 06 00:12:31 2017

@author: salwus
"""

def regula():
	a,b,c,d,e,f=-5.,-4.,0.,1.,8.,9.
```

```
	print reg(a,b)
	print reg(c,d)
	print reg(e,f)

def reg(a,b):
	fb=-1. + 39.*b + 4.*b*b - (b**3)
	fa=-1. + 39.*a + 4.*a*a - (a**3)

	while(True):
		x= (fb*a-fa*b)/(fb-fa)
		fx=-1. + 39.*x + 4.*x*x - x**3

		if(fx==0):
			break
		elif(fx*fa<0):
			b=x
			fb=fx
		else:
			a=x
			fa=fx
		if(abs(fx) <= 0.00000001):#dokladnosc
			break
	return x
```

Kod w pythonie (secant.py):

```
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 06 00:43:16 2017

@author: salwus
"""

def secant():
	a,b,c,d,e,f=-5.,-4.,0.,1.,8.,9.
	print sec(a,b)
	print sec(c,d)
	print sec(e,f)

def sec(a,b):
	fb=-1. + 39.*b + 4.*b*b - (b**3)
	fa=-1. + 39.*a + 4.*a*a - (a**3)

	while(True):
		x=a -fa*(a-b) / (fa-fb)
		fx=-1. + 39.*x + 4.*x*x - x**3

		if(fx==0):
			break
		else:
			b=a
			a=x
```

```
                fb=fa
                fa=fx
        if(abs(fx) <= 0.00000001):#dokladnosc
                break
return x
```