

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа №1

«Методы одномерной оптимизации нулевого порядка»

Выполнил студент группы №М32111

Чу Тхи Фыонг Тхао

Данг Хоанг Дык

Преподаватель

Москаленко М. А.

САНКТ-ПЕТЕРБУРГ

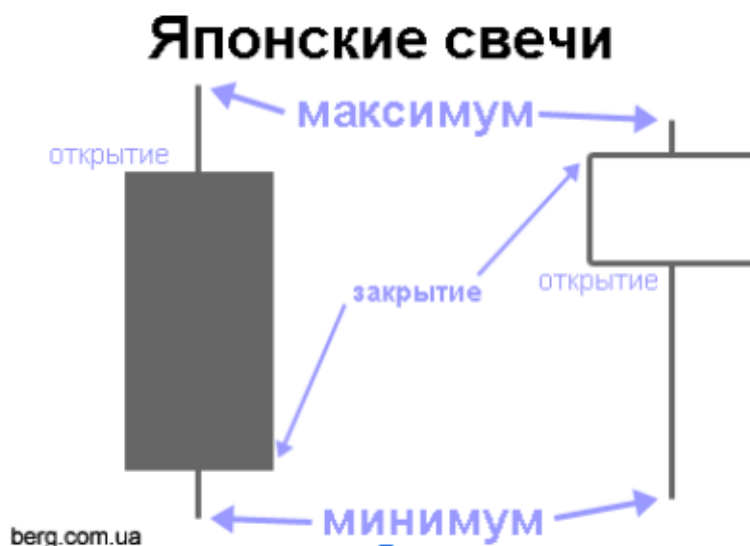
2021

Цель работы: Изучаем методы одномерной оптимизации (метод дихотомии, метод золотое сечение, метод Фибоначи, метод парабол, метод Брента). Знаем какой случай использовать какой метод. Силы и слабые каждого метода.

Задача

Вариант 3

Биржевые котировки



Одним из наиболее важных показателей для анализа поведения биржи является минимальная цена акции за день. Этот показатель рассчитывается на различных периодах времени наравне с максимальной ценой акции за период, цене на начало и конец рассматриваемого периода, а также объемом продаж. Все вместе эти показатели образуют знакомый многим инструмент - японские свечи.

Предположим, что аналитик разработал магический алгоритм, предсказывающий поведение акции. Результатом работы алгоритма является участок функции, который отражает движение котировок на период следующей “свечи”:

$$y = \sin(x) - \ln x^2 - 1$$

По данному предсказанию определите момент времени (в условных единицах), в который вы могли бы совершить покупку акций по минимальной цене.

Реализция алгоритм

Ссылка: <https://github.com/chu412/Applied-Mathematics/blob/main/Lab1/Lab1.py>

Метод дихотомии

```
Lab1 > Lab1.py > Bine
1  from math import *
2  def y(x):
3      try:
4          y = sin(x) - log(x*x) - 1
5          return sin(x) - log(x*x) - 1
6      except:
7          return None
8
9
10 def dichotomy_method(a,b,eps):
11     itt = 0
12     calculations = 0
13
14     while(b-a>=eps):
15         x0 = (a+b)/2
16         itt+=1
17         calculations +=2
18         y1 = y(x0 - eps/2)
19         y2 = y(x0 + eps/2)
20
21         if(y1<=y2):
22             b = x0
23         else:
24             a = x0
25
26     return [a,itt,calculations]
```

Метод золотого сечения

```
28 ✓ def golden_ratio(a,b,eps):
29     itt = 0
30     calculations = 2
31     c = a + (b-a)*( 3 - 5**0.5)/2
32     d = b - (b-a)*( 3 - 5**0.5)/2
33     y1 = y(c)
34     y2 = y(d)
35
36 ✓ while(b-a>=eps):
37     itt+=1
38     # print(a," ",b)
39
40 ✓     if(y1>y2):
41         # print("!")
42         a = c
43         c = d
44         y1=y2
45         d = b - (b-a)*( 3 - 5**0.5 )/2
46         calculations +=1
47         y2 = y(d)
48 ✓     else:
49         b = d
50         d = c
51         y2 = y1
52         c = a + (b-a)*(3 - 5**0.5 )/2
53         calculations +=1
54         y1 = y(c)
55
56     return [c,itt,calculations]
```

Метод Фибоначчи

```
58  def Bine(n):  
59      return (((1 + 5**0.5)/2)**n) - (((1 - 5**0.5)/2)**n)/(5**0.5)  
60  
61  def Fibonacci(a,b,eps):  
62      n = int((b-a)/(10*eps) + 0.5)  
63  
64      itt = 0  
65      calculations = 2  
66      #воспользуемся формулой Бинэ для определения примерного значения числа Фибоначчи  
67      x1 = a + Bine(n-2)/Bine(n)*(b-a)  
68      x2 = b - Bine(n-2)/Bine(n)*(b-a)  
69  
70      y1 = y(x1)  
71      y2 = y(x2)  
72  
73      # print(Bine(n-2))  
74  
75      while(n>itt + 3):  
76          x1 = a + Bine(n - itt - 2)/Bine(n - itt)*(b-a)  
77          x2 = b - Bine(n - itt - 2)/Bine(n - itt)*(b-a)  
78  
79  
80          itt+=1  
81          calculations +=2  
82  
83          if(y1>y2):  
84              a = x1  
85              # x2 = a + Bine(n - itt - 2)/Bine(n - itt)*(b-a)  
86              calculations +=1  
87  
88              y2 = y(x2)  
89          else:  
90              b = x2  
91              # x1 = b - Bine(n - itt - 2)/Bine(n - itt)*(b-a)  
92              calculations +=1  
93  
94              y1 = y(x1)  
95  
96  
97      return [a,itt,calculations]  
98
```

Метод парабол

```
99  def parabola(a,b,eps):
100      x1 = a
101      x2 = (a+b)/2
102      x3 = b
103
104      calculations = 3
105      itt = 0
106      f1 = y(x1)
107      f2 = y(x2)
108      f3 = y(x3)
109
110      x2 = (x1+x3)/2
111
112      #ищем нужные точки пока не выполниться равенство f1 >= f2 <= f3
113      while(x3-x1)>=eps and not (f1 >= f2 <= f3):
114          x2 = (x1+x3)/2
115          itt+=1
116          calculations +=2
117          f1 = y(x2 - eps/2)
118          f3 = y(x2 + eps/2)
119
120          if(f1 >= f2 <= f3):
121              break
122
123          if(f1<=f3):
124              x3 = x2
125          else:
126              x1 = x2
127
128      calculations +=1
129      x2 = (x1+x3)/2
130      f2 = y(x2)
131
132      a0 = f1
133      a1 = (f2-f1)/(x2-x1)
134      a2 = ((f3 - f1)/(x3-x1) - (f2 - f1)/(x2-x1))/(x3-x2)
135      x_d = (x1+x2 - a1/a2)/2
136
```

```
138 while(abs(x_d - x2)>=eps):
139     # print(x1," ",x2," ",x3)
140     itt +=1
141
142     a0 = f1
143     a1 = (f2-f1)/(x2-x1)
144     a2 = ((f3 - f1)/(x3-x1) - (f2 - f1)/(x2-x1))/(x3-x2)
145
146     x_d = (x1+x2 - a1/a2)/2
147
148     calculations +=1
149     f_d = y(x_d)
150
151     if(f_d>=f2):
152
153         x1 = x_d
154         f1 = f_d
155     else:
156         # print("!")
157         x1 = x2
158         f1 = f2
159         x2 = x_d
160
161
162     return [x2,itt,calculations]
```

Метод Брента

```
175 def Brenta(a,b,eps):
176     r = (3 - 5**0.5)/2
177     x=w=v = a - r*(b-a)
178     d_c = d_p = b - a
179
180     itt = 0
181     calculations = 0
182     u = None
183     while(max(x - a , b - x) > eps):
184         itt+=1
185         g = d_p/2
186         d_p = d_c
187         # print(x," ",w," ",u)
188
189         u = H_parab(x,w,u,y(x),y(w),y(u))
190
191         if(u == None or u<a or u>b or abs(u-x)>g):
192             if(x<(a+b)/2):
193                 u=x + r*(b-x)
194                 d_p = b-x
195             else:
196                 u=x - r*(x - a)
197                 d_p = x - a
198
199
200         d_c = abs(u - x)
201
202         calculations +=2
203         f_u = y(u)
204         f_x = y(x)
205
206         if(f_u>f_x):
207             if(u<x):
208                 a = u
209             else:
210                 b = u
211
212
213         calculations +=1
214         f_w = y(w)
215
216         if(f_u<=f_w or w==x):
217             v = w
218             w = u
219         elif(f_u <= f_w or v == x or v == w):
220             v = u
```



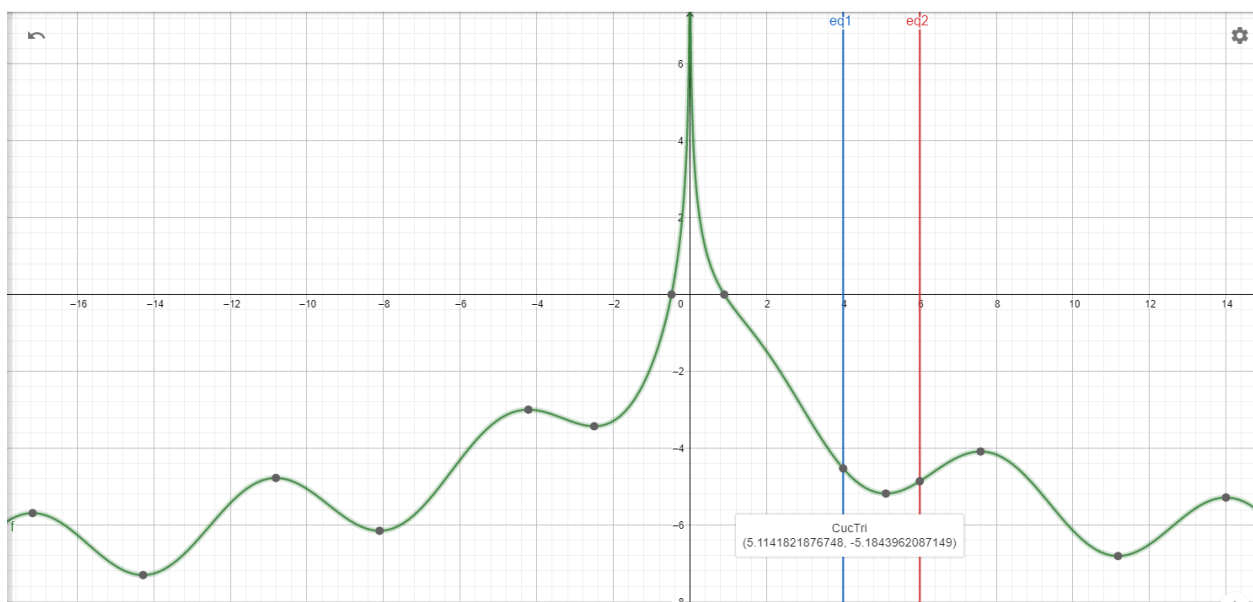
```

221         else:
222             if(u<x):
223                 b = x
224             else:
225                 a = x
226             v = w
227             w = x
228             x = u
229         return [u,itt,calculations]
230
231

```

Тесты для функции изадания

$$y = \sin(x) - \ln(x^2) - 1$$



Пример: $[a, b] = [4, 6]$, $\varepsilon = 0.005$

```

242
243     print("")
244     print(dichotomy_method(4,6,0.005))
245     print(golden_ratio(4,6,0.005))
246     print(Fibonacci(4,6,0.005))
247     print(parabola(4,6,0.005))
248     print(Brenta(4,6,0.005))
249

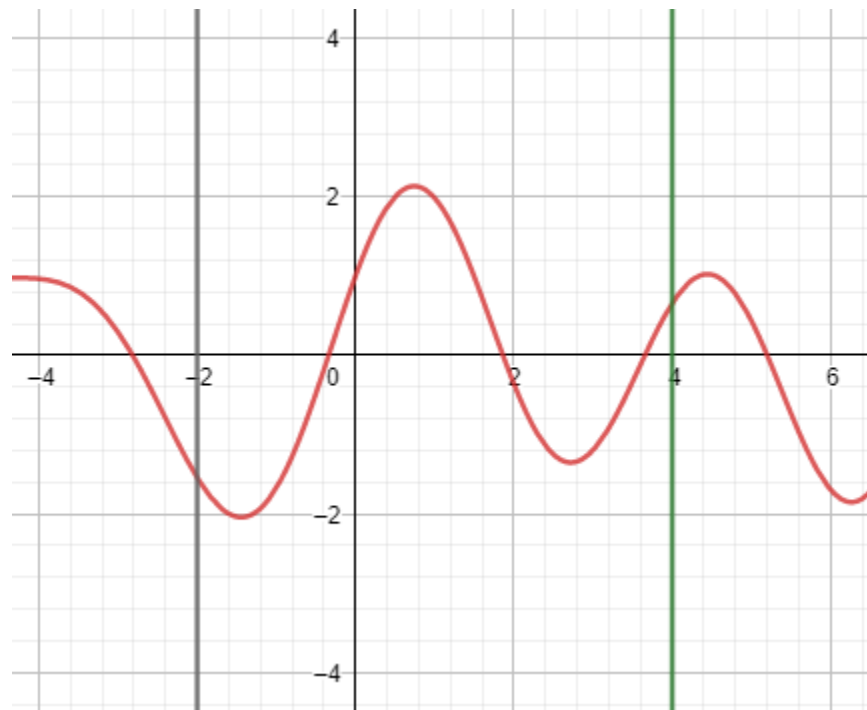
```

Результать

```
[5.11328125, 9, 18]  
[5.113655592309255, 13, 15]  
[5.23606797749979, 37, 113]  
[5.17312988331682, 1, 5]  
[5.117494349760253, 14, 37]
```

Тесты для многомадальных функций

$$y = \sin x + \cos(\sqrt{2}x) + \sin(\sqrt{3}x)$$



Пример: $[a, b] = [-2, 4]$, $\varepsilon = 0.005$

```
8  def y(x):  
9      try:  
10         y = sin(x) + cos((2**0.5)*x) + sin((3**0.5)*x)  
11         return sin(x) + cos((2**0.5)*x) + sin((3**0.5)*x)  
12     except:  
13         return None
```

```

237 print("")
238 print(dichotomy_method(-2,4, 0.005))
239 print(golden_ratio(-2,4, 0.005))
240 print(Fibonacci(-2,4, 0.005))
241 print([parabola(-2,4, 0.005)])
242 print(Brenta(-2,4, 0.005))

```

Результат

```

[2.716796875, 11, 22]
[2.7201072017448, 15, 17]
[3.9999999999999996, 117, 353]
[2.71826171875, 11, 26]
[-1.4397048375501413, 16, 43]

```

Таблица

№	Метод	a	b	ϵ	Искомая точка	Количество итерации	Количество вычислений функции
1	Дихотоми	4	6	0.005	5.11328125	9	18
	Золотого сечения				5.113655923	13	15
	Фиббоначи				5.236067977	37	113
	парабол				5.173129883	1	5
	Брента				5.117494349	14	37
2	Дихотоми	4	7	0.005	5.11328125	10	20
	Золотого сечения				5.1135486289 988945	14	16
	Фиббоначи				4.708203932	57	173
	парабол				5.349289883	1	5
	Брента				5.114907906	14	35

3	Дихотомии	-20	-18	0.004	-20	9	18
	Золотого сечения				-19.99853725	13	15
	Фиббоначи				-20	47	143
	парабол				-19.99804687	9	22
	Брента				-20.51798711	13	35
4	Дихотомии	-120	-118	0.003	-120	10	20
	Золотого сечения				-119.9990937	14	16
	Фиббоначи				-120	64	194
	парабол				-119.9990234	10	24
	Брента				-120.7626796	8	24
5	Дихотомии	1	4	0.003	3.9970703125	10	20
	Золотого сечения				3.998640684	15	17
	Фиббоначи				3.9999999999	97	293
	парабол				3.9985351562	10	24
	Брента				3.9981214776	16	32
6	Дихотомии	8	14	0.003	11.172851562	11	22
	Золотого сечения				11.175340139	16	18
	Фиббоначи				12.583592135	197	593
	парабол				11.132651778	2	6
	Брента				11.709635995	9	27
	Дихотомии				-8.103515625	11	22

7	Золотого сечения	-10	-6	0.002	-8.103086384	16	18
	Фиббоначи				-8.832815729	197	593
	парабол				-8.094559236	2	6
	Брента				-8.102394103	17	47