Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ОТЧЕТ

по лабораторной работе №2
по дисциплине «**Прикладная математика**».

Вариант №3 “Биржевые котировки”

Автор: Кирьянов Глеб Дмитриевич

Факультет: ФИТиП

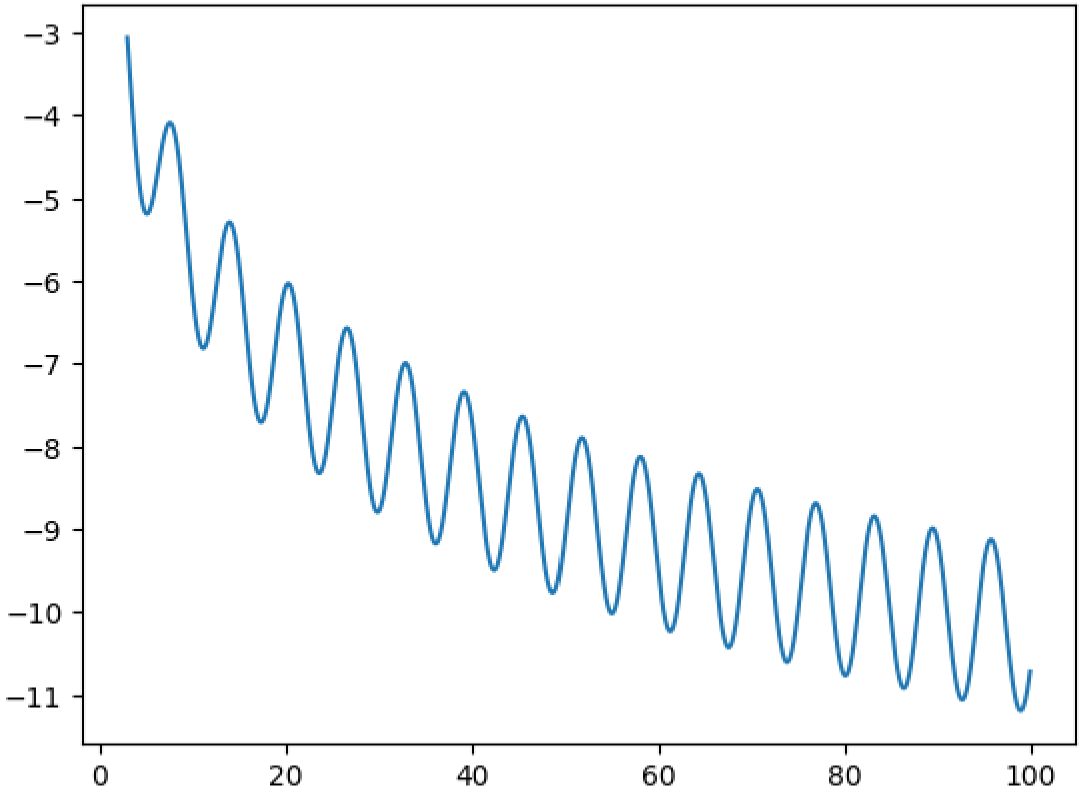
Группа: M32091

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург 2023

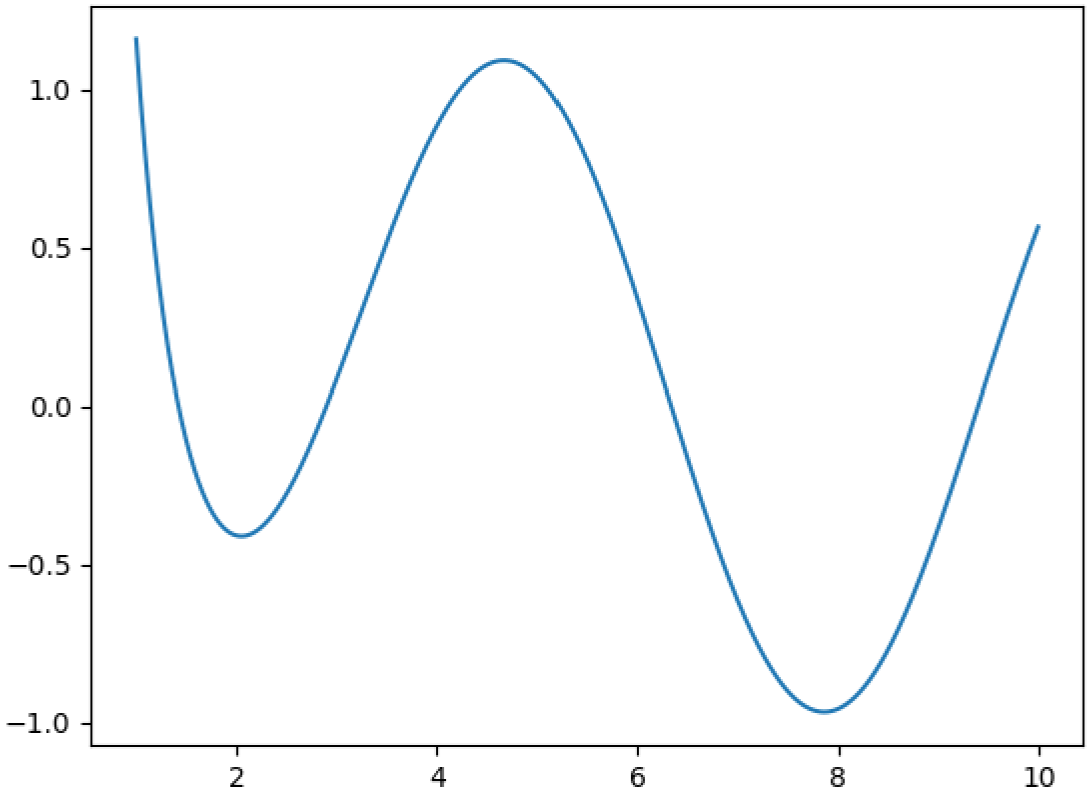
Исследуемая функция

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

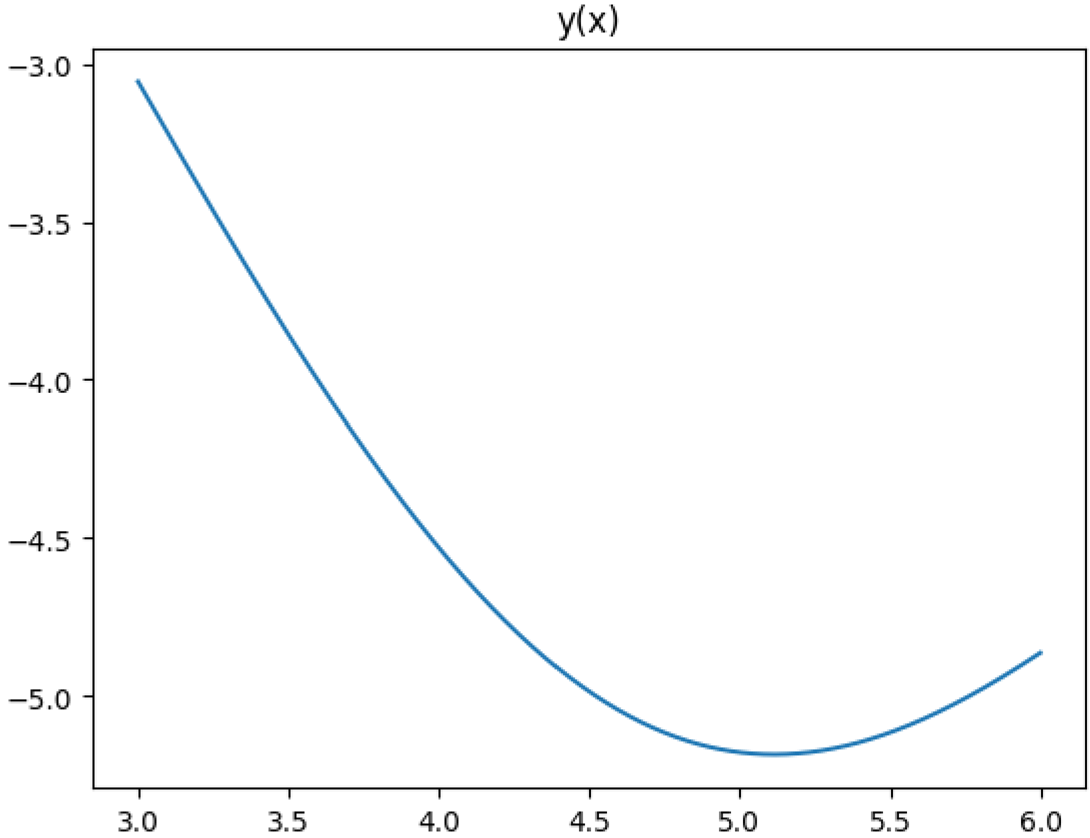


y`(x) = cos(x) - 2/x

y``(x) = -sin(x) + 2/(x^2)



Пусть участком исследования будет [a; b] = [3; 6]. На данном участке вторая производная функции неотрицательна, значит функция является на ней унимодальной.



Метод дихотомии

Пусть нам дан отрезок [l, r], тогда, согласно методу дихотомии этот отрезок делится пополам и в точках, отстоящих от центра($c = \frac{l+r}{2}$) этого отрезка на $\delta < \varepsilon/2$, рассчитываются значения данной функции $f(c - \delta)$ и $f(c + \delta)$.

Имеем три случая:

1) $f(c + \delta) > f(c - \delta) \Rightarrow$ минимум находится на отрезке [l, c]
2) $f(c + \delta) < f(c - \delta) \Rightarrow$ минимум находится на отрезке [c, r]
3) $f(c + \delta) = f(c - \delta) \Rightarrow$ минимум на отрезке $[c - \delta, c + \delta]$

Таким образом, в первых двух случаях мы исследуем полученный отрезок, пока его длина не станет $< \varepsilon$.

Примерное количество итераций для заданного $\varepsilon \Rightarrow ln(\frac{r-l}{\varepsilon})/ln(2)$, на каждой итерации производится два вычисления.

```
E = 0.001
[3; 6]
[4.4997; 6]
[4.4997; 5.2501]
```

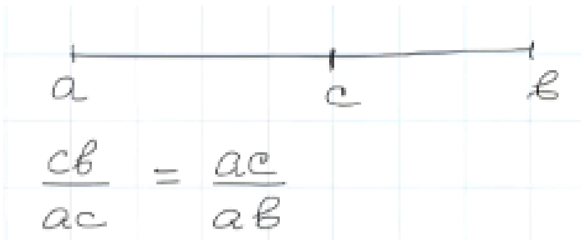
```
[4.8747; 5.2501]
[5.0622; 5.2501]
[5.0622; 5.1564]
[5.109; 5.1564]
[5.109; 5.133]
[5.109; 5.1212]
[5.109; 5.1154]
[5.112; 5.1154]
[5.1134; 5.1154]
[5.1134; 5.1146]
```

Метод золотого сечения

Пусть нам дан отрезок [l, r]. В данном методе мы выделяем две промежуточные точки x_1 и x_2 на расстоянии $s = \alpha(r - l)$ от конечных точек, а затем вычисляем в них значения функции.
Имеем два случая:
1) $f(x_1) \geq f(x_2)$ ⇒ минимум находится на отрезке [x_1, r]
2) $f(x_1) < f(x_2)$ ⇒ минимум находится на отрезке [l, x_2]

Аналогично методу дихотомии, исследуем полученный отрезок, пока его длина не станет $<$ ε.

В методе золотого сечения значение α подобрано так, что в полученном отрезке меньшей длины одна из промежуточных точек совпадает с промежуточной точкой предыдущего шага, таким образом мы сокращаем число вычислений целевой функции на всех шагах кроме первого до 1.



Это значение $\alpha = \frac{3-\sqrt{5}}{2} \approx 0.382$ и называется константой золотого сечения.
В итоге, метод золотого сечения работает за большее количество итераций(потому что отрезок уменьшается с меньшей скоростью), но за меньшее количество вычислений относительно метода дихотомии.

```
E = 0.001
[3; 6]
[4.1459; 6]
```

```
[4.8541; 6]
[4.8541; 5.5623]
[4.8541; 5.2918]
[5.0213; 5.2918]
[5.0213; 5.1885]
[5.0851; 5.1885]
[5.0851; 5.149]
[5.0851; 5.1246]
[5.1002; 5.1246]
[5.1095; 5.1246]
[5.1095; 5.1189]
[5.1095; 5.1153]
[5.1117; 5.1153]
[5.1131; 5.1153]
[5.1131; 5.1145]
```

Метод Фиббоначи

Это улучшение реализации поиска с помощью золотого сечения, служащего для нахождения минимума функции. Он точно также требует дважды вычислить значение функции на первой итерации, а на каждой последующей по одному. Отличается от метода золотого сечения тем, что от итерации к итерации коэффициент сокращения интервала неопределенности будет меняться согласно последовательности Фибоначчи, то есть он более эффективен по сходимости. В этом методе заранее требуется выбрать число итераций n.

Вычислим точки для первого отрезка по формулам:
$x1 = l + (r - l)\, fibonacci(n - 1) \,/\, fibonacci(n + 1)$
$x2 = l + (r - l)\, fibonacci(n) \,/\, fibonacci(n + 1)$
После чего на каждой итерации имеем два случая:
1) $f(x1) > f(x2) \Rightarrow$ на следующей итерации мы рассматриваем отрезок $[x1,\, r]$ с точками $x1_{\text{новое}} = x2$ и $x2_{\text{новое}} = r - (x1 - l)$
2) $f(x1) \le f(x2) \Rightarrow$ на следующей итерации мы рассматриваем отрезок $[l,\, x2]$ с точками $x2_{\text{новое}} = x1$ и $x1_{\text{новое}} = l + (r - x2)$

Две точки, в которых мы вычисляем значение функции всегда находятся симметрично относительно середины отрезка, поэтому для следующих рассматриваемых отрезков мы можем искать точки так, как описано выше.
В последней итерации мы получаем интервал, содержащий искомый минимум.

```
E = 0.001
[3; 6]
[4.15; 6]
[4.85; 6]
[4.85; 5.56]
[4.85; 5.29]
[5.02; 5.29]
[5.02; 5.19]
[5.09; 5.19]
[5.09; 5.15]
[5.085; 5.125]
[5.1; 5.125]
[5.11; 5.125]
[5.11; 5.119]
[5.113; 5.1188]
[5.113; 5.1165]
[5.113; 5.1153]
[5.113; 5.1142]
```

Метод парабол

В данном методе для минимизации заданной функции мы используем квадратичную функцию $p(x) = ax^2 + bx + c$

Пусть существуют такие точки $x_1 < x_2 < x_3$, тогда $x_{min} \in [x_1, x_3]$.

Коэффициенты нашей аппроксимирующей параболы найдем путем решения системы уравнений:

$ax_i^2 + bx_i + c = f_i = f(x_i),\ i = \overline{1, 3}$

Тогда минимум равен

$u = -\frac{b}{2a} = x_2 - \frac{(x_2 - x_1)^2(f_2 - f_3) - (x_2 - x_3)^2(f_2 - f_1)}{2((x_2 - x_1)(f_2 - f_3) - (x_2 - x_3)(f_2 - f_1))}$

Если выполняются условия $f_2 < f_1$ и $f_2 < f_3$, то наша точка u гарантированно попадает в интервал $[x_1, x_3]$. Сравнивая точки x_2 и u, находящиеся в нашем интервале $[x_1, x_3]$ мы можем сократить интервал поиска.

Данный метод обладает суперлинейной скоростью сходимости, однако эта скорость гарантируется только в малой окрестности точки минимума. На начальных стадиях оптимизации метод парабол может делать очень маленькие шаги или очень большие, что может приводить к неустойчивому поведению. Обязательное условие - вычисление значений функции в крайних точках исследуемого интервала.

```
E = 0.001
[3; 6]
[5; 6]
[5; 5.1567]
[5; 5.122]
[5; 5.1145]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
[5; 5.1142]
```

Комбинированный метод Брента

Метод золотого сечения представляет собой надежный способ оптимизации, который гарантированно сходится за гарантированное число итераций, но обладает лишь линейной скоростью сходимости. Метод парабол работает быстрее в малой окрестности оптимального решения, но может работать очень долго и неустойчиво на начальных стадиях итерационного процесса. Поэтому на практике для решения задачи одномерной оптимизации используется метод Брента, который эффективно комбинирует эти две стратегии.

В этом методе на каждой итерации отслеживается значение в шести точках: a, c, x, w, v, u.
$[a, c]$ - интервал поиска
x - точка, соответствующая наименьшему значению функции
w - точка, соответствующая второму снизу значению функции
v - предыдущее значение w.

В данном методе аппроксимирующая парабола строится с помощью трех наилучших точек x, w, v(если эти точки и их значения различны). При этом минимум аппроксимирующей параболы u принимается в качестве следующей точки оптимизационного процесса если:
1) $u \in [a, c]$ и отстоит от границ интервала не менее чем на ε

2) u отстоит от точки x не более, чем на половину от длины предпредыдущего шага

Если точка u отвергается, то следующая точка находится с помощью золотого сечения большего из интервалов $[a, x]$ и $[x, c]$.
Данные условия приема минимума параболы u необходимы, так как парабола на текущей итерации проводится через x, v, w, для которых не гарантируются соотношения v < x < w или w < x < v, из-за чего минимум параболы может оказаться вне интервала [a, c].
Первое условие помогает избежать слишком маленьких шагов в оптимизации, которые могут свидетельствовать о локальном застопоривании метода парабол.
Второе условие помогает избежать слишком больших шагов в оптимизации, которые могут соответствовать биениям в методе парабол.

```
E = 0.001
[3; 6]
[3.9271; 6]
[4.5; 6]
[5.0729; 6]
[5.1131; 6]
[5.1131; 5.1162]
[5.1131; 5.1152]
```

Результаты сравнения методов по количеству итераций и вычислений

	E	y_min	iterations	function_calculations
algorithm				
dichotomy	**1.000000e-01**	-5.184346	6.0	12.0
	1.000000e-02	-5.184396	10.0	20.0
	1.000000e-03	-5.184396	13.0	26.0
	1.000000e-04	-5.184396	16.0	32.0
	1.000000e-05	-5.184396	20.0	40.0

	1.000000e-06	-5.184396	23.0	46.0
	1.000000e-07	-5.184396	26.0	52.0
	1.000000e-08	-5.184396	30.0	60.0
	1.000000e-09	-5.184396	33.0	66.0
	1.000000e-10	-5.184396	36.0	72.0
golden_ration	**1.000000e-01**	-5.184342	8.0	10.0
	1.000000e-02	-5.184396	12.0	14.0
	1.000000e-03	-5.184396	17.0	19.0
	1.000000e-04	-5.184396	22.0	24.0
	1.000000e-05	-5.184396	24.0	26.0
	1.000000e-06	-5.184396	24.0	26.0
	1.000000e-07	-5.184396	24.0	26.0
	1.000000e-08	-5.184396	24.0	26.0
	1.000000e-09	-5.184396	24.0	26.0
	1.000000e-10	-5.184396	24.0	26.0
fibonacci	**1.000000e-01**	-5.184342	8.0	10.0
	1.000000e-02	-5.184396	12.0	14.0
	1.000000e-03	-5.184396	17.0	19.0

	1.000000e-04	-5.184396	22.0	24.0
	1.000000e-05	-5.184396	27.0	29.0
	1.000000e-06	-5.184396	31.0	33.0
	1.000000e-07	-5.184396	36.0	38.0
	1.000000e-08	-5.184396	45.0	47.0
	1.000000e-09	-5.184396	51.0	53.0
	1.000000e-10	-5.184396	58.0	59.0
parabols	**1.000000e-01**	-5.184396	19.0	22.0
	1.000000e-02	-5.184396	19.0	22.0
	1.000000e-03	-5.184396	19.0	22.0
	1.000000e-04	-5.184396	19.0	22.0
	1.000000e-05	-5.184396	20.0	23.0
	1.000000e-06	-5.184396	23.0	26.0
	1.000000e-07	-5.184396	26.0	29.0
	1.000000e-08	-5.184396	31.0	34.0
	1.000000e-09	-5.184396	37.0	40.0
	1.000000e-10	-5.184396	42.0	45.0
brent	**1.000000e-01**	-5.184396	6.0	7.0

1.000000e-02	-5.184396	9.0	10.0
1.000000e-03	-5.184396	7.0	8.0
1.000000e-04	-5.184396	9.0	10.0
1.000000e-05	-5.184396	12.0	13.0
1.000000e-06	-5.184396	14.0	15.0
1.000000e-07	-5.184396	8.0	9.0
1.000000e-08	-5.184396	11.0	12.0
1.000000e-09	-5.184396	17.0	18.0
1.000000e-10	-5.184396	22.0	23.0

Минимизация мультимодальной функции

		y_min	iterations	function_calculations
algorithm	E			
dichotomy	**1.000000e-01**	-12.010713	14.0	28.0
	1.000000e-02	-12.011009	17.0	34.0
	1.000000e-03	-12.011009	21.0	42.0
	1.000000e-04	-12.011009	24.0	48.0
	1.000000e-05	-12.011009	27.0	54.0
	1.000000e-06	-12.011009	31.0	62.0
golden_ration	**1.000000e-01**	-12.884008	19.0	21.0
	1.000000e-02	-12.884067	24.0	26.0
	1.000000e-03	-12.884067	24.0	26.0
	1.000000e-04	-12.884067	24.0	26.0
	1.000000e-05	-12.884067	24.0	26.0
	1.000000e-06	-12.884067	24.0	26.0

fibonacci	**1.000000e-01**	-12.884021	19.0	21.0
	1.000000e-02	-12.884068	23.0	25.0
	1.000000e-03	-12.884068	28.0	30.0
	1.000000e-04	-12.884068	33.0	35.0
	1.000000e-05	-12.884068	38.0	40.0
	1.000000e-06	-12.884068	47.0	49.0
parabols	**1.000000e-01**	-13.529597	16.0	19.0
	1.000000e-02	-13.529597	16.0	19.0
	1.000000e-03	-13.529597	19.0	22.0
	1.000000e-04	-13.529597	19.0	22.0
	1.000000e-05	-13.529597	19.0	22.0
	1.000000e-06	-13.529597	21.0	24.0
brent	**1.000000e-01**	-13.529597	13.0	14.0
	1.000000e-02	-13.529597	15.0	16.0
	1.000000e-03	-13.529597	20.0	21.0
	1.000000e-04	-13.529597	22.0	23.0
	1.000000e-05	-13.529597	21.0	22.0
	1.000000e-06	-13.529597	17.0	18.0

Судя по результатам применение данных алгоритмов минимизации на мультимодальных функциях выдает различные локальные минимумы.

Вывод:

Нами были изучены методы минимизации унимодальных функций без производных. Все они хорошо справляются со своей задачей, но каждый имеет свои плюсы и минусы, а также функции, на которых его применение будет лучше, поэтому выбрать универсальный алгоритм сложно. Но комбинированный метод Брента нам показался почти идеальным вариантом, так как сочетает в себе плюсы метода золотого сечения и метода парабол.

Код и рассчеты:
https://github.com/ElderEv1l/applied-math/blob/main/lab-2/lab2.ipynb