

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное  
учреждение высшего образования  
Национальный исследовательский Нижегородский государственный университет  
им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Программная инженерия

**Курс «Системы компьютерной математики»**

**Отчет по заданию 38:**

**«Выплаты по кредиту. Решение трансцендентного  
уравнения»**

**Выполнил:**

студент группы 381908-3

Кучерявых Я.В.

Нижегород  
2023

## Постановка задачи

Предположим, что кто-то желает взять кредит на сумму \$100000.

Он выбирает среди следующих возможностей:

1. взять кредит на 10 лет с ежемесячными выплатами в \$1500;
2. взять кредит на 15 лет с ежемесячными выплатами в \$1250;
3. взять кредит на 20 лет с ежемесячными выплатами в \$1160.

Какой кредит выгоднее брать?

Можно просто вычислить переплату, а можно еще учесть процентные ставки.

Сами процентные ставки здесь не указаны, но можно их найти из *основного финансового уравнения*:

$$PV(1+i)^n + PMT(1+i)^{n-1} + PMT(1+i)^{n-2} + \dots + PMT + FV = 0, \quad (1)$$

связывающего следующие величины:

- $PV$  - величина кредита (present value);
- $FV$  - баланс после  $n$  платежей (future value);
- $PMT$  - величина периодического платежа со знаком минус (payment);
- $n$  - количество платежей;
- $i$  - процентная ставка (interest rate).

Например, для случая 1):

- $PV = 100000$ ;
- $FV = 0$  (после всех платежей банк и частное лицо друг другу ничего не должны);
- $PMT = -1500$ ;
- $n = 10 \times 12 = 120$ .

Используя формулу для суммы элементов геометрической прогрессии, уравнение 1 запишем в следующем виде:

$$PV(1+i)^n + PMT \frac{(1+i)^n - 1}{i} + FV = 0, \quad (2)$$

Относительно  $i$  это уравнение в общем случае аналитически решено быть не может, поэтому надо использовать численные методы.

1. Написать функцию, решающую уравнение 2. Ваша функция должна вызывать

*scipy.optimize.brentq*. Можете воспользоваться другим методом. Обоснуйте свой выбор.

2. Рассчитать годовые процентные ставки для примеров 1)-3). Годовая процентная ставка равна  $i_{\text{year}} = mi$ , где  $m$  - количество платежей в год (в нашем случае  $m = 12$ ).
3. Сравнить процентные ставки. Какой кредитный план выбрали бы вы?

## Решение

Для начала импортируем все нужные библиотеки:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import optimize
```

Определим переменные начальных условий и создадим массив *cases* с тремя кортежами для описания каждого случая из задания. Первый элемент в кортеже будет отвечать за *PMT*, а второй – за *n*.

```
PV = 100000 # Сумма кредита
FV = 0      # Остаток после всех платежей
m = 12      # Платежей в год

# Массив с тремя случаями
# (ежемесячный платеж, кол-во лет)
cases = [
    (-1500, 10,),
    (-1250, 15,),
    (-1160, 20,)
]
```

Создадим функцию, которая будет решать финансовое уравнение. На вход она принимает параметры *PV*, *FV*, *PMT* и *n*. Внутри первым делом создается локальная функция, описывающая уравнение. Затем она подставляется в *optimize.brentq* с параметрами диапазона поиска корня  $a = 0.001$ ,  $b = 1$ , и результат возвращается:

```
def solve_i(PV, FV, PMT, n):
    def payment_func(i):
        t = (1 + i)**n
        return PV*t + PMT * (t - 1) / i + FV # основное финансовое
        ↪ уравнение
    # вызываем brentq и ищем i в диапазоне от 0,1% до 100%
    return optimize.brentq(payment_func, 0.001, 1)
```

Создадим переменные, в которые будем записывать лучшую процентную ставку и лучший случай. Затем в цикле по массиву *cases* рассмотрим каждый случай по такому принципу:

1. Забираем переменные *PMT* и *n* из текущего кортежа;
2. получаем *i* с помощью нашей функции *solve\_i*;

3. перезаписываем  $i\_min$  как минимальное из  $i$  и  $i\_min$ , если оно изменяется, то перезаписываем  $best\_case$ ;
4. Выводим результат.

```

i_min = 1 # 100%, максимальное число, будем сюда искать минимальную
        ↪ ставку
best_case = -1

for idx, c in enumerate(cases):
    PMT = c[0] # ежемесячный платеж
    n = c[1] * 12 # кол-во платежей

    i = solve_i(PV, FV, PMT, n)
    i_min = min(i, i_min)
    if i_min == i:
        best_case = idx

    print("Случай", idx + 1)
    print("i =", i)
    print("i_y=", i * m)
    print()

```

После выполнения кода получились следующие результаты ( $i$  – месячная процентная ставка,  $i_y$  – годовая процентная ставка):

1. Случай №1:  $i = 1.09\%$ ,  $i_y = 13.12\%$
2. Случай №2:  $i = 1.06\%$ ,  $i_y = 12.77\%$
3. Случай №3:  $i = 1.07\%$ ,  $i_y = 12.84\%$

Выводим наименьшие показатели:

```

print("Лучший вариант:")
print("№", best_case + 1, sep=" ")
print("i =", i_min)

```

Вариант с наименьшими процентными ставками – 2.

## Заключение

Таким образом, мы написали функцию, решающую основное финансовое уравнение, и определили, что лучшим вариантом с наименьшими процентными ставками является вариант под номером 2.