Министерство образования Республики Беларусь
Учреждение образования
“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Методы численного анализа

ОТЧЁТ
к лабораторной работе 9
на тему

Методы Эйлера и Рунге-Кутта

Выполнил: студент группы 053502
Герчик Артём Вадимович

Проверил: Анисимов Владимир Яковлевич

Минск 2022

Оглавление

Цели выполнения задания

1) Изучить решение задачи Коши для обыкновенных дифференциальных уравнений методом Эйлера и методом Рунге-Кутта.

Краткие теоретические сведения

Рассмотрим дифференциальное уравнение $y' = f(x, y)$ с начальным условием $y(x_0) = y_0$. Будем предполагать, что *f(x,y)* непрерывная и непрерывно дифференцируемая по y функция в окрестности замкнутой области

$$D = \{(x, y) \mid a \le x \le b, c \le y \le d\},$$

содержащей внутри себя точку (x_0, y_0).

Требуется решить задачу Коши: найти непрерывно дифференцируемую функцию *y=y(x)*, такую что $y'(x) = f(x, y(x))$ при всех $x \in [a,b]$ и $y(x_0) = y_0$.

Разобьем отрезок *[a, b]* с помощью точек разбиения $a = x_0, x_1, ..., x_n = b$ с шагом $h = (b - a)/n$. Тогда узлы разбиения имеют вид $x_k = x_0 + kh,\ k = \overline{0, n}$.
Пусть $y(x_0), y(x_1), ..., y(x_n)$ - значения функции в точках разбиения.

1) Метод Эйлера

Построим рекуррентную последовательность:

$$y_{k+1} = y_k + hf(x_k, y_k), \qquad k = 0,1... \qquad (9.1)$$

$$y_0 = y(x_0),$$

которую называют последовательностью Эйлера. Соединяя ломаными все точки (x_k, y_k), полученные из рекуррентной последовательности Эйлера, получим ломаную линию, приближающую график решения $y = y(x)$. Функция, график которой совпадает с ломаной Эйлера, принимается за приближенное решение задачи Коши.

Точность метода Эйлера на всем отрезке $[a, b]$ будет *O(h)*.

Для повышения точности вычислений иногда используется модифицированный метод Эйлера, в котором рекуррентная последовательность Эйлера вычисляется по формулам

$$y_{k+1} = y_k + hf(x_k + \frac{h}{2}, y_k + \frac{h}{2} f(x_k, y_k)), \qquad k = 0,1\ldots,n-1. \qquad (9.2)$$

Модифицированный метод Эйлера обычно дает более точное приближение решения.

Пример. Пусть требуется решить задачу Коши:

$$\begin{cases} y' = -y, \ \ x \in [0,1] \\ y(0) = 1. \end{cases}$$

Полагая $h = 0{,}2$ и используя метод Эйлера, получим, как легко убедиться, из формулы Эйлера (9.1)

$$y_{k+1} = y_k + 0.2 \cdot (-y_k) = 0.8 \cdot y_k.$$

С другой стороны, используя модифицированный метод Эйлера, получим в силу формулы (2) рекуррентную последовательность

$$y_{k+1} = y_k + 0.2 \cdot (-y_k) = 0.82 \cdot y_k \ .$$

Поскольку точным решением задачи Коши, как легко проверить, является функция $y = e^{-x}$, можно сравнить точность обоих методов.

	0	1	2	3	4	5
x_k	0	0.2	0.4	0.6	0.8	1
y_k	1	0.8	0.64	0.572	0.4086	0.3277
$y_k^{модиф}$	1	0.82	0.6724	0.5514	0.4521	0.3708
e^{-x}	1	0.8187	0.6703	0.5488	0.4493	0.3679

Общепризнанным недостатком метода Эйлера является его не достаточно высокая точность. Несомненным достоинством метода Эйлера является его простота.

2) Метод Рунге-Кутта четвертого порядка.

На каждом шаге производится вычисление коэффициентов K_1, K_2, K_3, K_4:

$K_1 = hf(x_k, y_k)$;

$K_2 = hf(x_k + \frac{h}{2}, y_k + \frac{K_1}{2})$;

$K_3 = hf(x_k + \frac{h}{2}, y_k + \frac{K_2}{2})$;

$K_4 = hf(x_k + h, y_k + K_3)$.

Затем вычисляем

$$y_{k+1} = y_k + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4).$$

Данный метод имеет точность $O(h^4)$ на $[a,b]$.

Рассмотрим пример, который мы использовали для иллюстрации точности метода Эйлера.

Пример. Требуется решить задачу Коши:

$$\begin{cases} y' = -y \\ y(0) = 1 \end{cases}$$ на отрезке [0, 1].

Выберем шаг $h = 0{,}2$. Результат вычислений поместим в таблицу.

	0	1	2	3	4	5
x_k	0	0.2	0.4	0.6	0.8	1
y_k	1	0.8187	0.6703	0.5487	0.4493	0.3678
e^{-x}	1	0.8187	0.6703	0.5488	0.4493	0.3679

Таким образом, метод Рунге-Кутта 4-го порядка отличается очень высокой точностью. К определенным его недостаткам относится большая сложность и трудоемкость (на каждом шаге необходимо четырежды вычислять значения функции f вместо одного раза в методе Эйлера).

Отметим, что на практике выбирают начальную длину шага h таким образом, чтобы $h^4 < \varepsilon$, где ε - заданная точность вычисления решения. Затем шаг выбирают вдвое меньшим и останавливают вычисления, если разность полученных значений y_k со значениями, полученными при начальном выборе шага меньше ε. В противном случае шаг еще раз уменьшают вдвое и т.д.

Программная реализация

```
count_of_dots = 10 ** 3
eps = 10 ** -3

# # EXAMPLE 1 #
# def yder(x, y): return x
#
#
# y0 = 0
# left_border, right_border = -1, 0
#
#
# def ans(x): return (1 / 2) * x ** 2
#
#
# xplot = np.linspace(left_border, right_border,
count_of_dots)
# yplot = [ans(x_dot) for x_dot in xplot]
#
#
# # EXAMPLE 1 #


# # EXAMPLE 2 #
# def yder(x, y): return x + y
#
#
# y0 = 0
# left_border, right_border = -1, 1
#
#
# def ans(x): return -x - 1 + np.exp(x)
#
#
```

```
# xplot = np.linspace(left_border, right_border,
count_of_dots)
# yplot = [ans(x_dot) for x_dot in xplot]
#
#
# # EXAMPLE 2 #

# MAINT TASK #
m, a = 1.0, 0.7


def yder(x, y): return (a * (1 - y ** 2)) / ((1 + m) * x
** 2 + y ** 2 + 1)


y0 = 0
left_border, right_border = 0, 1


# MAIN TASK #


def euler_method(x_dot, N):
    y_dots = [y0]
    h = x_dot / N
    for i in range(N):
        x = i * h
        y = y_dots[-1]
        y_dots += [y + h * yder(x, y)]
    return y_dots


def runge_kutta_method(x_dot, N):
    y_dots = [y0]
    h = x_dot / N
```

```
    for i in range(N):
        x = i * h
        y = y_dots[-1]
        K1 = h * yder(x, y)
        K2 = h * yder(x + h / 2, y + K1 / 2)
        K3 = h * yder(x + h / 2, y + K2 / 2)
        K4 = h * yder(x + h, y + K3)
        y_dots += [y + 1 / 6 * (K1 + 2 * K2 + 2 * K3 +
K4)]
    return y_dots


def get_value_at_chosed_point(method, x):
    n = 1
    while True:
        olddots, newdots = method(x, n), method(x, 2 * n)
        if max(abs(newdots[2 * i] - olddots[i]) for i in
range(n + 1)) < eps:
            return newdots[-1], 2 * n

        else:
            n *= 2


def create_dots(method, x_dots):
    maxn = 0
    midn = []
    for x in x_dots:
        y, n = get_value_at_chosed_point(method, x)
        maxn = max(maxn, n)
        midn += [n]
    midn = sum(midn) / len(x_dots)
    return midn, maxn
```

```
print("Методы Эйлера и Рунге-Кутта\n")

print("Точки для вычислений: ", count_of_dots)
print("Точность: ", eps)

x_dots = [left_border + (right_border - left_border) /
count_of_dots * i for i in range(count_of_dots + 1)]

midn, maxn = create_dots(euler_method, x_dots)
print('\n\nМЕТОД ЭЙЛЕРА:')
print(f'Среднее: {midn}')
print(f'Максимальное: {maxn}')

midn, maxn = create_dots(runge_kutta_method, x_dots)
print('\n\nМЕТОД РУНГЕ-КУТТА:')
print(f'Среднее: {midn}')
print(f'Максимальное: {maxn}')
```

Полученные результаты

Тестовый пример 1

$y' = x, \quad y(0) = 0, \quad [0,1]$
Точность : 10^{-4}
Кол-во точек: 10^3

Ответ:

Методы Эйлера и Рунге-Кутта

Точки для вычислений: 1000
Точность: 0.0001

МЕТОД ЭЙЛЕРА:
Количество отрезков:
Среднее: 2455.932067932068
Максимальное: 8192

МЕТОД РУНГЕ-КУТТА:
Количество отрезков:
Среднее: 2.0
Максимальное: 2

Тестовый пример 2

$y' = x + y, \quad y(0) = 0, \quad [-1,1]$
Точность : 10^{-3}
Кол-во точек: 10^{3}

Ответ:

Методы Эйлера и Рунге-Кутта

Точки для вычислений: 1000
Точность: 0.001

МЕТОД ЭЙЛЕРА:
Количество отрезков:
Среднее: 326.12787212787214
Максимальное: 2048

МЕТОД РУНГЕ-КУТТА:
Количество отрезков:
Среднее: 2.675324675324675
Максимальное: 4

ЗАДАНИЕ
Вариант 7

ЗАДАНИЕ. С помощью метода Эйлера, а затем метода Рунге-Кутта найти с точностью до 0.001 решения следующих уравнений на отрезке [0; 1].

$$y' = \frac{a(1-y^2)}{(1+m)x^2 + y^2 + 1}, \quad y(0) = 0,$$

где значения параметров a и m принимают следующие значения для вариантов k.

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
m	1.0	1.5	2.0	1.0	1.5	2.0	1.0	1.5	2.0	1.0	1.5	2.0	1.0	2.0
a	0.5	0.7	0.9	1.1	1.3	0.5	0.7	0.9	1.1	1.3	0.5	0.7	0.9	1.0

$$y' = \frac{a\left(1-y^2\right)}{(1+m)x^2 + y^2 + 1}, \quad y(0) = 0, \qquad где \quad m = 1.0, \ \ a = 0.7$$

Ответ:

Методы Эйлера и Рунге-Кутта

Точки для вычислений: 1000
Точность: 0.001

МЕТОД ЭЙЛЕРА:
Количество отрезков:
Среднее: 116.55144855144856
Максимальное: 256

МЕТОД РУНГЕ-КУТТА:
Количество отрезков:
Среднее: 2.711288711288711
Максимальное: 4

Выводы

Таким образом, в ходе выполнения лабораторной работы были освоены метод Эйлера, метод Рунге-Кутта четвёртого порядка для решения задачи Коши для обыкновенных дифференциальных уравнений. Составлена компьютерная программа, на тестовых примерах проверена правильность её работы, по количеству необходимых для этого отрезков сравнена трудоёмкость методов.