Академический университет

Решение дифференциальных уравнений

Николай Жидков

28 апреля 2018 г.

1 Структура программы

Во-первых, с уравнением 2-го порядка из условия что-то не так, потому что функция $y(x)$ не является решением это уравнения и под начальные условия не подходит. Поэтому я поменял условия на $y(1) = -1, y(x) = -\sqrt{x}$, так все сходится.
Во-вторых, я использовал в коде в некоторый момент массивы из модуля $numpy$, потому что их можно складывать друг с другом и домножать на коэффициенты. Мне это было нужно, чтобы задачи из обоих пунктов решать одинковыми функциями.

2 Структура программы

Программа разделена на функции, записанные в файле solve.py. Содержательные функции:

- $evaluate_y(a, b, n, y0, f)$, выполняет проход метода Рунге-Кутта 3-го порядка:
  - Принимает границы отрезка a, b, количество отрезков n (количество узлов минус один), начальное значение $y0$ и функцию f.
  - Возвращает посчитанный массив Y (в первом пункте это массив значений в точках, во втором пункте это массив пар (значение, производная)).
- $solve(a, b, f, y0, start\_n, eps, max\_n)$), запускает Рунге-Кутта с нужными параметрами:
  - Принимает границы отрезка a, b, начальное значение $y0$ и функцию f, начальное количество отрезков $start\_n$, граничное значение для оценки Рунге eps и максимальное число отрезков max_n.

  - Вычисляет массив значений в точках, каждый раз увеличивая сетку вдвое. Останавливается, когда n стало равно max_n или когда оценки Рунге на двух последовательных прогонах стала меньше eps для всех значений.

- $deviations*(Y, Y\_t, start\_n, *)$, выводит информацию по вычислениям:
  - Принимает вычисленные значения Y, реальные значения Y_t и количство контрольных точек $start\_n$.
  - Печатает информацию об отклонениях в вычислениях.

- $process\_command\_line\_args()$, считывет аргументы командой строки:
  - Ничего не принимает
  - Возвращает метод для запуска run и ее параметр $param$.

- $test*(a, b, y0, start_n, eps, y_t, f)$, запускает вычисления для данного уравнения первого порядка:
  - Принимает границы отрезка a, b, начальное значение $y0$ и функцию f, начальное количество отрезков $start\_n$, граничное значение для оценки Рунге eps и функцию ответа y_t
  - Проводит вычисления, печатает ответ.

3 Структура файлов исходных данных

В данной задаче файлы не используются.

4 Примеры вызова из командной строки

Есть 4 режима работы для решения задач из условия

- Решаем уравнение первого порядка, ожидая порогового значения 0.001

  $python3\ solve.py\ --run=1\ --param=0.001$

- Решаем уравнение второго порядка, деля интервал на 20 отрезков

  $python3\ solve.py\ --run=2\ --param=20$

- Решаем уравнение второго порядка, вносим возмущение в $y(1)$ в размере 5 процентов (количество отрезков всегда 10).

  $python3\ solve.py\ --run=2.1\ --param=5$

- Решаем уравнение второго порядка, вносим возмущение в $y'(1)$ в размере 5 процентов (количество отрезков всегда 10).

  $python3\ solve.py\ --run=2.2\ --param=5$

5 Численный эксперимент

5.1 Уравнение первого порядка

eps	количество отрезков	максимальное отклонение интеграла в процентах от eps
0.001	20	6
0.0001	40	8
0.00001	80	11

Как видно из таблицы, с уменьшением eps на порядок, количество проходов увеличивается на 1 (что то же самое, число отрезков увеличивается вдвое). Точность, как легко видеть, падает также на порядок, что вполне ожидаемо.
Довольно неожиданным только является факт, что при том, что максимальное отклонение мы разрешаем eps, максимальное отклонение в контрольных узлах дает лишь 10 процентов от него.

5.2 Уравнение второго порядка, вычисление значений

max n	максимальное отклонение интеграла
10	0.00000243
20	0.00000029
40	0.00000004

Как видно, с увеличением числа узлов в два раза, максимальное отклонение падает на порядок.

5.3 Уравнение второго порядка, возмущения

В таблице записаны максимальные отклонения

Процент	В значение	В производную
3	0.03	0.00000243
4	0.04	0.00463403
5	0.05	0.01854162

В левом столбце стоят значения процентов, так как при изменении начального значения на p процентов, потом все значения примерно сохраняют это изменение. В целом видно, что производная гораздо меньше влияет на результат, что довольно логично, так изменение начального значение оказывает непосредственное влияние на дальнейшие вычисления.