Отчёт по лабораторной работе №8

Иванов Анатолий Павлович

Цель работы:

Требуется запрограммировать модифицированный метод Эйлера. Программа должна работать для произвольной размерности системы уравнений.

Функция правой части системы и начальное условие подаются на вход программе. Вычисления должны производиться с пошаговым контролем точности по правилу Рунге. Если на текущем шаге точность не достигается, то шаг уменьшается в 2 раза, если достигнутая погрешность меньше заданной в 64 раза, то шаг увеличивается в 2 раза.

Входные данные:

На вход подаются несколько строк, в которых:

начало промежутка t_0

конец промежутка T

начальный шаг h_0

максимальное число вызовов функции правой части N_x

желаемая точность eps

число уравнений

следующие n+3 строк определяют функцию правой части на Python

последняя строка содержит n чисел - начальное условие

```
1.5
2.5
0.1
10000
0.0001
3
def fs(t, v, kounter):
#
#
  A = np.array([[-0.4, 0.02, 0], [0, 0.8, -0.1], [0.003, 0, 1]])
  kounter[0] += 1
  return np.dot(A, v)
1 1 2
```

Рисунок 1. “Входные данные”

Выходные данные

Программа печатает в консоль следующие столбцы, одна строчка соответствует одному шагу интегрирования:

1. значение t

2. значение шага h

3. оценка Рунге R

4. истраченое число вычислений правой части N

5. значения функций решений

Алгоритм

```
while t < T + h / 2:
    v_0 = v
    y1.append(v_0[0])
    y2.append(v_0[1])
    y3.append(v_0[2])
    print("{:13.6f}".format(t), "{:12.6f}".format(h), "{:15.5e}".format(dlt(v, v2)), "{:12d}".format(kounter[0]),
          end=' ')
    for vi in v_0:
        print("{:12.6f}".format(vi), end=' ')
    print()
    v = next(h, t, v_0, kounter)
    v1 = next(h / 2, t, v_0, kounter)
    v2 = next(h / 2, t + h / 2, v1, kounter)
    while dlt(v, v2) > eps:
        h /= 2
        v = v1
        v1 = next(h / 2, t, v_0, kounter)
        v2 = next(h / 2, t + h / 2, v1, kounter)
    t += h
    x.append(t)
    y.append(h)
print()
x.pop()
y.pop()
y1.pop()
y2.pop()
y3.pop()
return x, y, min(y), len(x), y1, y2, y3
```

Рисунок 2. “Алгоритм”

Результат работы программы:

Eps = 0.001

1.500000	0.100000	0	0	1.000000	1.000000	2.000000
1.600000	0.100000	8.45154e-05	6	0.962820	1.061398	2.210309
1.700000	0.100000	9.33737e-05	12	0.927221	1.125613	2.442690
1.800000	0.100000	1.03174e-04	18	0.893145	1.192637	2.699459
1.900000	0.100000	1.14015e-04	24	0.860540	1.262439	2.983178
2.000000	0.100000	1.26009e-04	30	0.829352	1.334956	3.296678
2.100000	0.100000	1.39277e-04	36	0.799532	1.410090	3.643086
2.200000	0.100000	1.53956e-04	42	0.771031	1.487698	4.025858
2.300000	0.100000	1.70196e-04	48	0.743801	1.567592	4.448812
2.400000	0.100000	1.88162e-04	54	0.717797	1.649522	4.916167
2.500000	0.100000	2.08039e-04	60	0.692976	1.733175	5.432587

Графики:

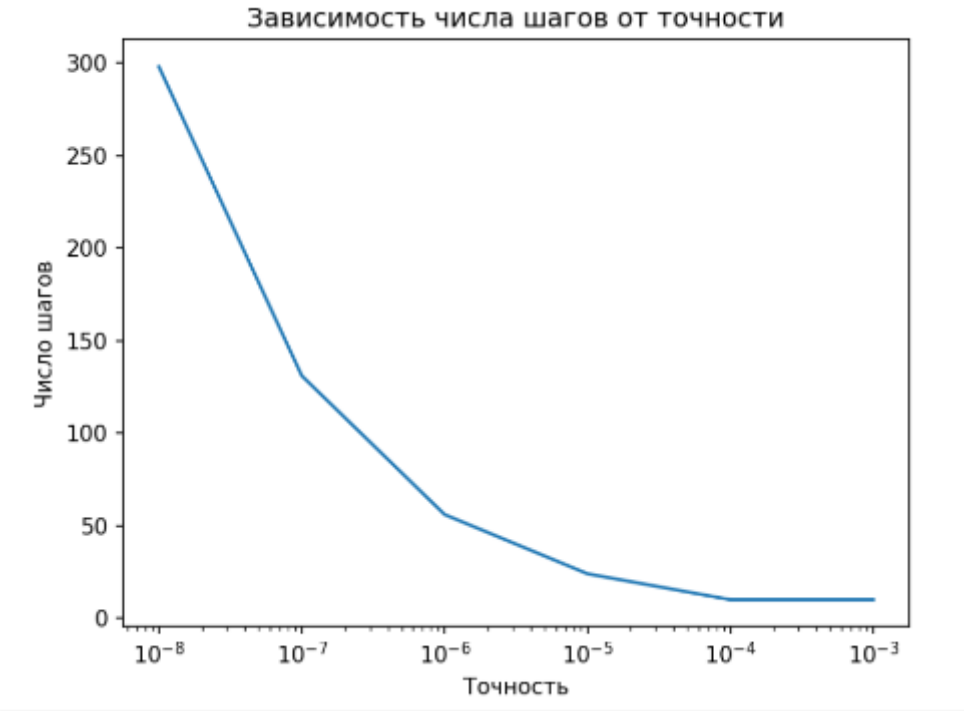

Рисунок 3. «График зависимости числа шагов от задаваемой точности»

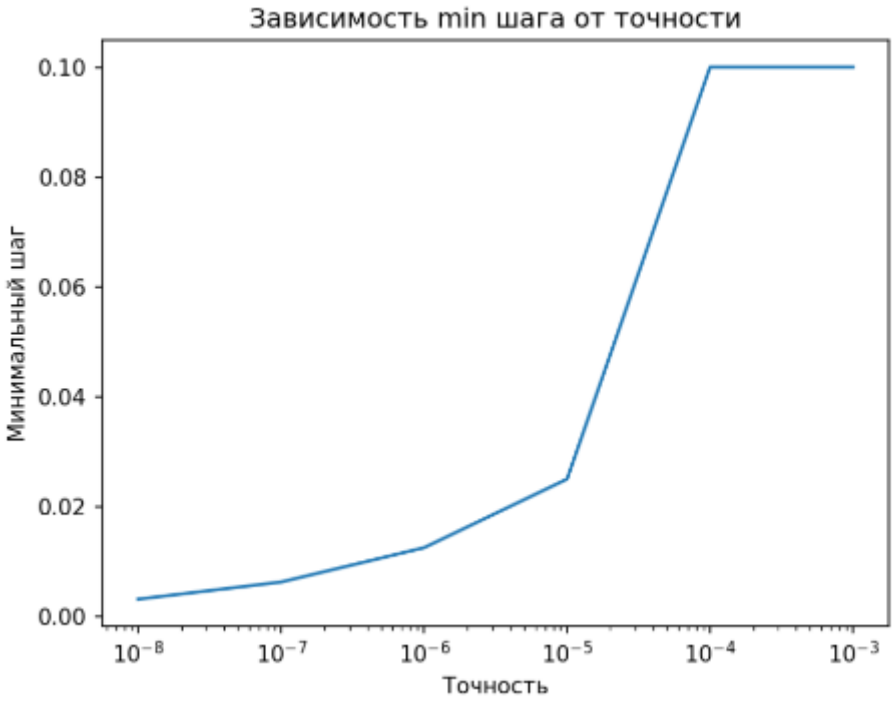

Рисунок 4. «График зависимости минимального шага от задаваемой точности»

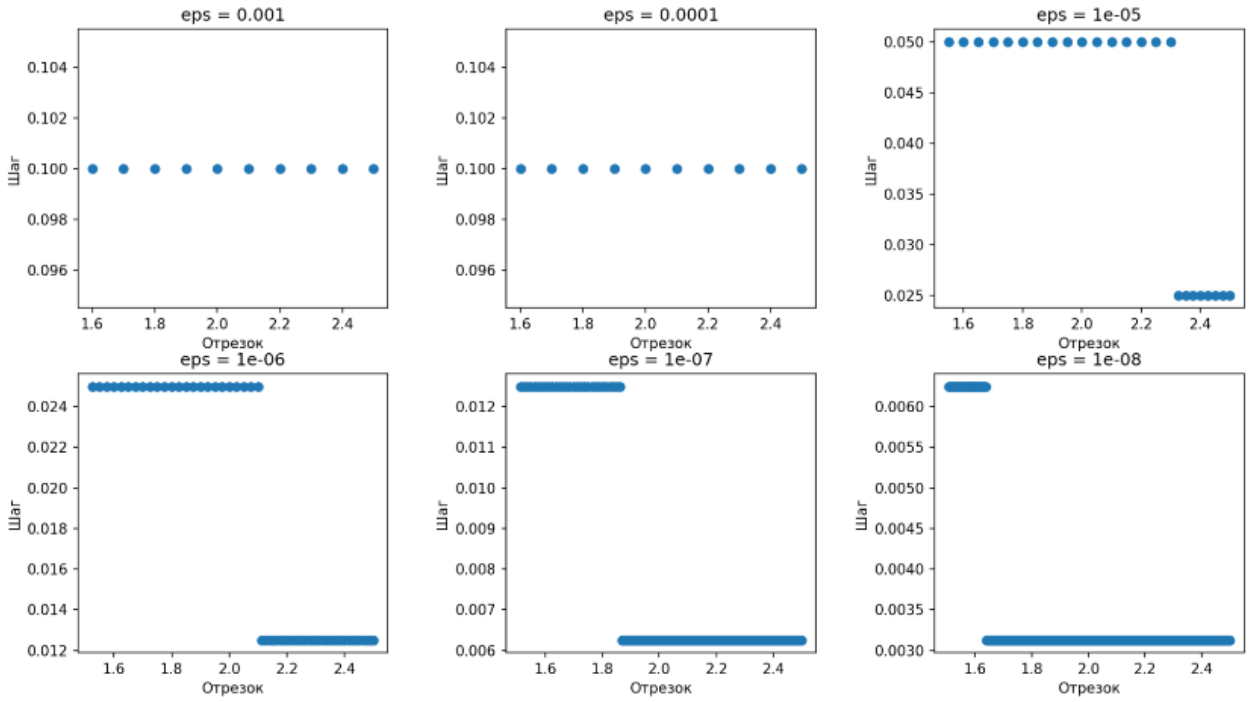

Рисунок 5. «Графики изменения шага по отрезку для разных значений задаваемой точности»

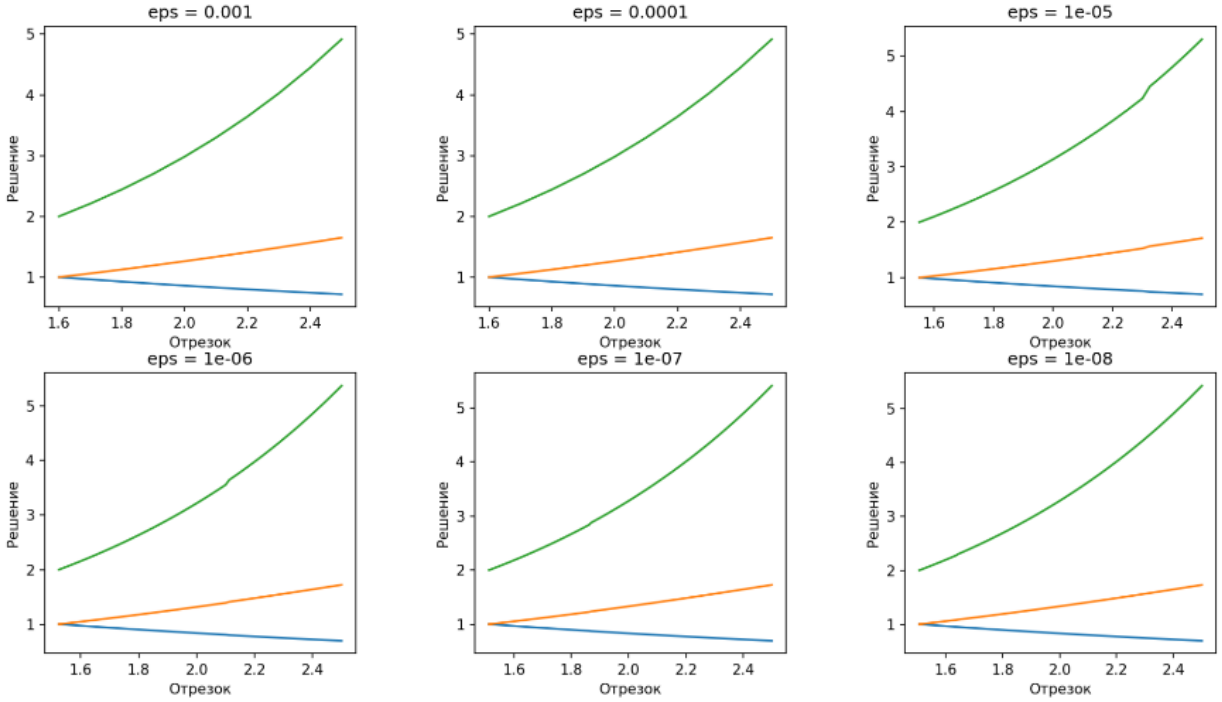

Рисунок 6. «Графики зависимости решения от задаваемой точности»