

Отчет

1. Соков Дмитрий Денисович, 5030102/10401
2. Требуется запрограммировать метод решения задачи Коши для ОДУ, по вариантам (вариант 3: Метод Рунге-Кутты 3 порядка). 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.
3. While $t < T + h / 2$:
 $v0$
 $v = \text{next } v(h, v0)$
 $v1 = \text{next } v(h/2, v0)$
 while $\text{norm}(v - v1) > \text{eps}$:
 $h = h / 2$
 $v = \text{next } v(h, v0)$
 $v1 = \text{next } v(h/2, v0)$
 while $\text{norm}(v - v1) < \text{eps}/64$:
 $h = h*2$
 $v = \text{next } v(h, v0)$
 $v1 = \text{next } v(h/2, v0)$
 $t = t + h$
 $v0 = v$
4. Запустить файл main.py. Функция answer выводит ответы и необходимые графики для разных точностей.
5. Решения для разных значений заданной точности в виде таблицы, ввод перед ответами:

```
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
Точность равна 0.001
    1.500000    0.100000    0.00000e+00         0    1.000000    1.000000    2.000000
    1.700000    0.200000    1.72446e-05        18    0.927200    1.125625    2.443307
    1.900000    0.200000    2.10944e-05        27    0.860502    1.262439    2.984686
    2.100000    0.200000    2.58084e-05        36    0.799479    1.410035    3.645848
    2.300000    0.200000    3.15805e-05        45    0.743736    1.567418    4.453309
    2.500000    0.200000    3.86483e-05        54    0.692899    1.732786    5.439452

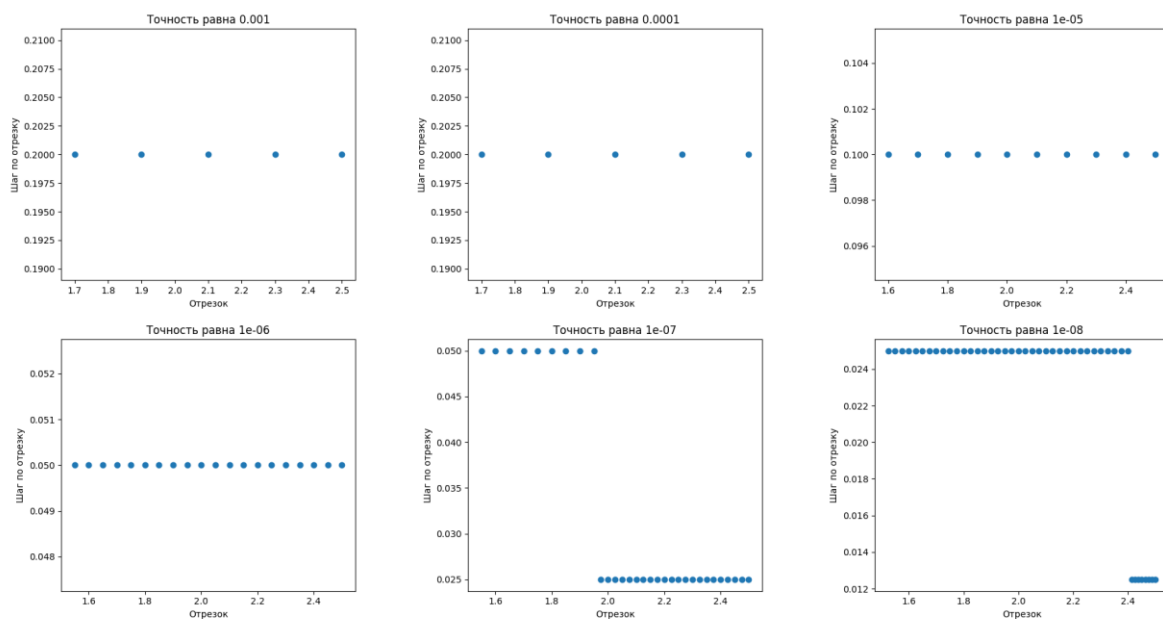
Точность равна 0.0001
    1.500000    0.100000    0.00000e+00         0    1.000000    1.000000    2.000000
    1.700000    0.200000    1.72446e-05        18    0.927200    1.125625    2.443307
    1.900000    0.200000    2.10944e-05        27    0.860502    1.262439    2.984686
    2.100000    0.200000    2.58084e-05        36    0.799479    1.410035    3.645848
    2.300000    0.200000    3.15805e-05        45    0.743736    1.567418    4.453309
    2.500000    0.200000    3.86483e-05        54    0.692899    1.732786    5.439452

Точность равна 1e-05
    1.500000    0.100000    0.00000e+00         0    1.000000    1.000000    2.000000
    1.600000    0.100000    1.06234e-06         9    0.962810    1.061402    2.210643
    1.700000    0.100000    1.17494e-06        18    0.927201    1.125614    2.443427
    1.800000    0.100000    1.29955e-06        27    0.893118    1.192627    2.700681
    1.900000    0.100000    1.43744e-06        36    0.860505    1.262407    2.984980
    2.000000    0.100000    1.59002e-06        45    0.829310    1.334888    3.299167
    2.100000    0.100000    1.75887e-06        54    0.799483    1.409968    3.646386
    2.200000    0.100000    1.94572e-06        63    0.770975    1.487503    4.030112
    2.300000    0.100000    2.15248e-06        72    0.743739    1.567296    4.454185
    2.400000    0.100000    2.38127e-06        81    0.717730    1.649096    4.922847
    2.500000    0.100000    2.63446e-06        90    0.692902    1.732581    5.440789

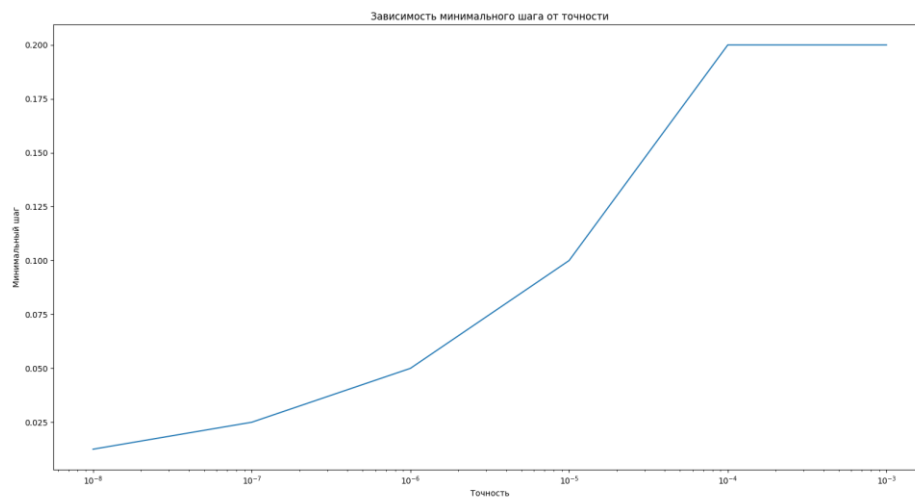
Точность равна 1e-06
    1.500000    0.100000    0.00000e+00         0    1.000000    1.000000    2.000000
    1.550000    0.050000    6.59194e-08        18    0.981204    1.030350    2.102694
    1.600000    0.050000    6.93246e-08        27    0.962810    1.061402    2.210650
    1.650000    0.050000    7.29067e-08        36    0.944812    1.093156    2.324139
    1.700000    0.050000    7.66748e-08        45    0.927202    1.125613    2.443444
    1.750000    0.050000    8.06387e-08        54    0.909973    1.158770    2.568862
    1.800000    0.050000    8.48084e-08        63    0.893118    1.192624    2.700709
    1.850000    0.050000    8.91948e-08        72    0.876631    1.227171    2.839312
    1.900000    0.050000    9.38090e-08        81    0.860505    1.262402    2.985020
    1.950000    0.050000    9.86629e-08        90    0.844733    1.298310    3.138195
    2.000000    0.050000    1.03769e-07         99    0.829310    1.334881    3.299222
    2.050000    0.050000    1.09140e-07        108    0.814229    1.372103    3.468502
    2.100000    0.050000    1.14791e-07        117    0.799483    1.409959    3.646459
    2.150000    0.050000    1.20734e-07        126    0.785067    1.448428    3.833538
    2.200000    0.050000    1.26987e-07        135    0.770975    1.487490    4.030207
    2.250000    0.050000    1.33564e-07        144    0.757201    1.527116    4.236956
    2.300000    0.050000    1.40483e-07        153    0.743740    1.567279    4.454304
    2.350000    0.050000    1.47762e-07        162    0.730584    1.607944    4.682793
    2.400000    0.050000    1.55418e-07        171    0.717730    1.649074    4.922995
    2.450000    0.050000    1.63472e-07        180    0.705171    1.690626    5.175511
    2.500000    0.050000    1.71945e-07        189    0.692903    1.732553    5.440971
```

Точность равна 1e-07						
1.500000	0.100000	0.00000e+00	0	1.000000	1.000000	2.000000
1.550000	0.050000	6.59194e-08	18	0.981204	1.030350	2.102694
1.600000	0.050000	6.93246e-08	27	0.962810	1.061402	2.210650
1.650000	0.050000	7.29067e-08	36	0.944812	1.093156	2.324139
1.700000	0.050000	7.66748e-08	45	0.927202	1.125613	2.443444
1.750000	0.050000	8.06387e-08	54	0.909973	1.158770	2.568862
1.800000	0.050000	8.48084e-08	63	0.893118	1.192624	2.700709
1.850000	0.050000	8.91948e-08	72	0.876631	1.227171	2.839312
1.900000	0.050000	9.38090e-08	81	0.860505	1.262402	2.985020
1.950000	0.050000	9.86629e-08	90	0.844733	1.298310	3.138195
1.975000	0.025000	6.46221e-09	108	0.836979	1.316513	3.217703
2.000000	0.025000	6.62733e-09	117	0.829310	1.334881	3.299223
2.025000	0.025000	6.79670e-09	126	0.821727	1.353412	3.382806
2.050000	0.025000	6.97041e-09	135	0.814229	1.372103	3.468504
2.075000	0.025000	7.14857e-09	144	0.806814	1.390952	3.556371
2.100000	0.025000	7.33130e-09	153	0.799483	1.409958	3.646462
2.125000	0.025000	7.51871e-09	162	0.792234	1.429118	3.738833
2.150000	0.025000	7.71094e-09	171	0.785067	1.448428	3.833542
2.175000	0.025000	7.90809e-09	180	0.777981	1.467886	3.930648
2.200000	0.025000	8.11030e-09	189	0.770975	1.487489	4.030211
2.225000	0.025000	8.31769e-09	198	0.764049	1.507233	4.132295
2.250000	0.025000	8.53040e-09	207	0.757201	1.527116	4.236962
2.275000	0.025000	8.74857e-09	216	0.750432	1.547132	4.344278
2.300000	0.025000	8.97233e-09	225	0.743740	1.567278	4.454311
2.325000	0.025000	9.20183e-09	234	0.737124	1.587550	4.567128
2.350000	0.025000	9.43721e-09	243	0.730584	1.607943	4.682801
2.375000	0.025000	9.67863e-09	252	0.724120	1.628452	4.801402
2.400000	0.025000	9.92624e-09	261	0.717730	1.649072	4.923005
2.425000	0.025000	1.01802e-08	270	0.711414	1.669798	5.047686
2.450000	0.025000	1.04407e-08	279	0.705171	1.690624	5.175522
2.475000	0.025000	1.07078e-08	288	0.699001	1.711543	5.306594
2.500000	0.025000	1.09818e-08	297	0.692903	1.732550	5.440984
Точность равна 1e-08						
1.500000	0.100000	0.00000e+00	0	1.000000	1.000000	2.000000
1.525000	0.025000	4.10516e-09	27	0.990551	1.015087	2.050706
1.550000	0.025000	4.20984e-09	36	0.981204	1.030350	2.102695
1.575000	0.025000	4.31721e-09	45	0.971957	1.045788	2.155999
1.600000	0.025000	4.42734e-09	54	0.962810	1.061402	2.210651
1.625000	0.025000	4.54028e-09	63	0.953762	1.077191	2.266687
1.650000	0.025000	4.65613e-09	72	0.944812	1.093156	2.324141
1.675000	0.025000	4.77494e-09	81	0.935959	1.109296	2.383048
1.700000	0.025000	4.89680e-09	90	0.927202	1.125612	2.443446
1.725000	0.025000	5.02179e-09	99	0.918540	1.142104	2.505372
1.750000	0.025000	5.14998e-09	108	0.909973	1.158770	2.568865
1.775000	0.025000	5.28146e-09	117	0.901499	1.175610	2.633965
1.800000	0.025000	5.41631e-09	126	0.893118	1.192624	2.700712
1.825000	0.025000	5.55462e-09	135	0.884829	1.209811	2.769148
1.850000	0.025000	5.69647e-09	144	0.876631	1.227170	2.839317
1.875000	0.025000	5.84197e-09	153	0.868523	1.244701	2.911261
1.900000	0.025000	5.99119e-09	162	0.860505	1.262402	2.985025
1.925000	0.025000	6.14424e-09	171	0.852575	1.280271	3.060656
1.950000	0.025000	6.30122e-09	180	0.844733	1.298309	3.138202
1.975000	0.025000	6.46222e-09	189	0.836979	1.316512	3.217709
2.000000	0.025000	6.62735e-09	198	0.829310	1.334880	3.299229
2.025000	0.025000	6.79671e-09	207	0.821727	1.353411	3.382812
2.050000	0.025000	6.97042e-09	216	0.814229	1.372102	3.468511
2.075000	0.025000	7.14858e-09	225	0.806814	1.390952	3.556378
2.100000	0.025000	7.33131e-09	234	0.799483	1.409957	3.646469
2.125000	0.025000	7.51873e-09	243	0.792234	1.429117	3.738840
2.150000	0.025000	7.71095e-09	252	0.785067	1.448427	3.833549
2.175000	0.025000	7.90811e-09	261	0.777981	1.467885	3.930655
2.200000	0.025000	8.11031e-09	270	0.770975	1.487488	4.030219
2.225000	0.025000	8.31771e-09	279	0.764049	1.507232	4.132303
2.250000	0.025000	8.53042e-09	288	0.757201	1.527114	4.236970
2.275000	0.025000	8.74859e-09	297	0.750432	1.547131	4.344287
2.300000	0.025000	8.97235e-09	306	0.743740	1.567277	4.454320
2.325000	0.025000	9.20185e-09	315	0.737124	1.587549	4.567137
2.350000	0.025000	9.43723e-09	324	0.730584	1.607942	4.682811
2.375000	0.025000	9.67865e-09	333	0.724120	1.628451	4.801412
2.400000	0.025000	9.92626e-09	342	0.717730	1.649071	4.923015
2.412500	0.012500	6.35116e-10	360	0.714563	1.659421	4.984966
2.412500	0.012500	6.35116e-10	360	0.714563	1.659421	4.984966
2.425000	0.012500	6.43190e-10	369	0.711414	1.669797	5.047696
2.437500	0.012500	6.51367e-10	378	0.708284	1.680197	5.111215
2.450000	0.012500	6.59647e-10	387	0.705172	1.690622	5.175533
2.462500	0.012500	6.68033e-10	396	0.702077	1.701071	5.240659
2.475000	0.012500	6.76526e-10	405	0.699001	1.711542	5.306605
2.487500	0.012500	6.85128e-10	414	0.695943	1.722035	5.373380
2.500000	0.012500	6.93839e-10	423	0.692903	1.732549	5.440995

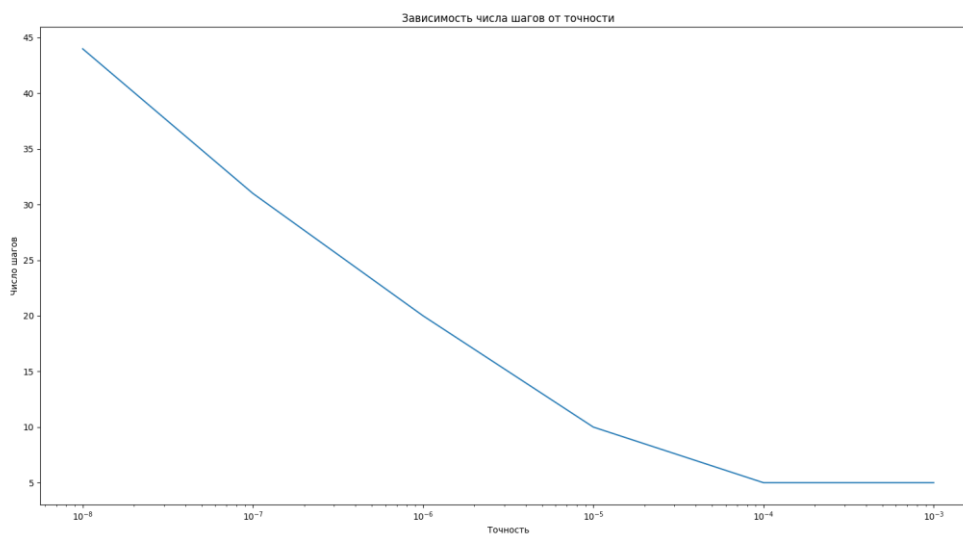
Изменение шага по отрезку для разных значений заданной точности:



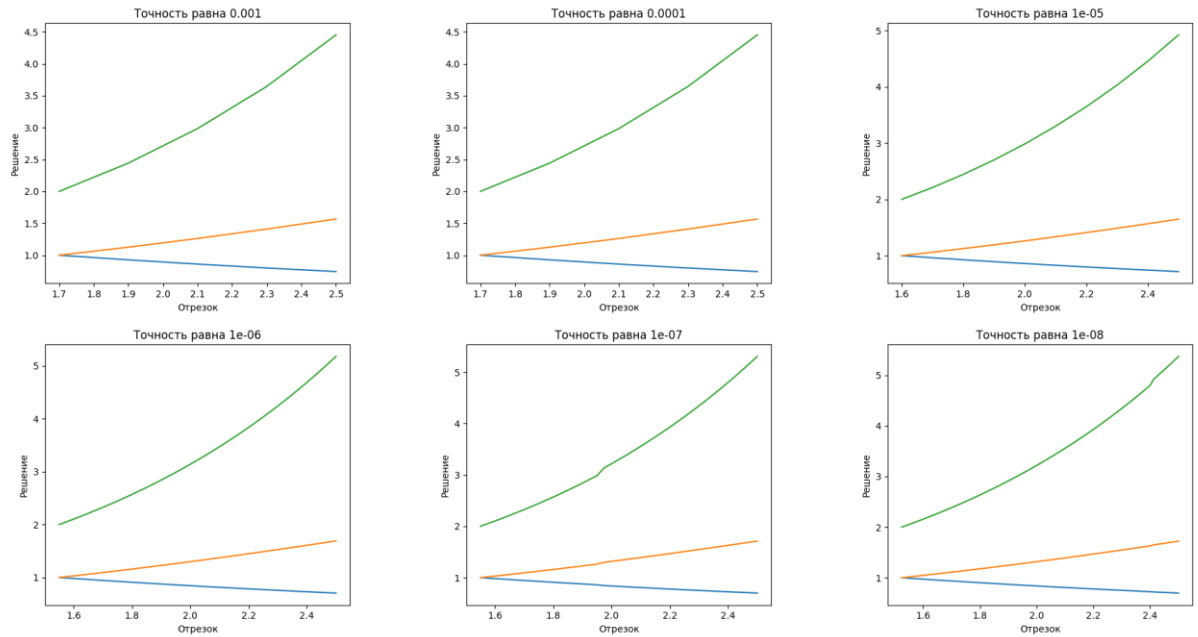
Зависимость минимального шага от заданной точности:



Зависимость числа шагов от заданной точности:



Решения для разных точностей:



Код:

```
import numpy as np
import matplotlib.pyplot as plt

def k1(h, t, v, kounter):
    return h * fs(t, v, kounter)

def k2(h, t, v, kounter, k1):
    return h * fs(t + h / 2, v + k1 / 2, kounter)

def k3(h, t, v, kounter, k1, k2):
    return h * fs(t + h, v + 2 * k2 - k1, kounter)

def next(h, t, v, kounter):
    k11 = k1(h, t, v, kounter)
    k21 = k2(h, t, v, kounter, k11)
    k31 = k3(h, t, v, kounter, k11, k21)
    return v + 1 / 6 * (k11 + 4 * k21 + k31)

def dlt(v, v1):
    return np.linalg.norm(np.array(v) - np.array(v1), ord=2) / (pow(2, 3) - 1)

def method(t, T, h, v, kounter, eps):
    v1 = v
    v2 = v
    x = []
    y = []
    y1 = []
    y2 = []
    y3 = []
    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 = 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 / 64:
            h *= 2
            v = next(h, t, v_0, kounter)
            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

def answer(t, T, h, v, kounter):
    epses = [0.001, 0.0001, 0.00001, 0.000001, 0.0000001, 0.00000001]
    k = 0
    minhes = []
    numstepses = []
    fig, ax = plt.subplots(2, 3)
    fig.tight_layout()
    fig1, ax1 = plt.subplots(2, 3)
    fig1.tight_layout()
    for item in epses:
        ax[k // 3, k % 3].set_title('Точность равна ' + str(item))
        ax[k // 3, k % 3].set_xlabel('Отрезок')
        ax[k // 3, k % 3].set_ylabel('Шаг по отрезку')
        ax1[k // 3, k % 3].set_title('Точность равна ' + str(item))
        ax1[k // 3, k % 3].set_xlabel('Отрезок')
        ax1[k // 3, k % 3].set_ylabel('Решение')
        print('Точность равна ', item)
        x, y, minh, numsteps, y1, y2, y3 = method(t, T, h, v, kounter, item)
        minhes.append(minh)
        numstepses.append(numsteps)
        ax[k // 3, k % 3].scatter(x, y)
        ax1[k // 3, k % 3].plot(x, y1)
        ax1[k // 3, k % 3].plot(x, y2)
        ax1[k // 3, k % 3].plot(x, y3)
        k += 1
    plt.figure()
    plt.semilogx(epses, minhes)
    plt.xlabel('Точность')
    plt.ylabel('Минимальный шаг')
    plt.title('Зависимость минимального шага от точности')

```

```

plt.figure()
plt.semilogx(epses, numstepses)
plt.xlabel('Точность')
plt.ylabel('Число шагов')
plt.title('Зависимость числа шагов от точности')
plt.show()

if __name__ == '__main__':
    func = []
    t_0 = float(input())
    T = float(input())
    h_0 = float(input())
    N_x = int(input())
    eps = float(input())
    n = int(input())
    for i in range(n + 3):
        func.append(input())
    b = ''
    for s in func:
        b = b + s + '\n'
    v_0 = input().split()
    for i in range(n):
        v_0[i] = float(v_0[i])
    t = t_0
    v = v_0
    v1 = v_0
    v2 = v_0
    h = h_0
    kounter = [0]
    exec(b)
    answer(t, T, h, v, kounter)

```