

Санкт-Петербургский государственный политехнический  
университет им. Петра Великого

Кафедра “Прикладная математика”

Отчет по лабораторной работе №8  
Методы оптимизации

**Работу выполнили студент группы № 5030102/10401**

Дубовицкий Владислав Александрович

**Преподаватель**

Козлов Константин Николаевич

Санкт-Петербург, 2024

## Условие:

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  $\epsilon$
- число уравнений
- следующие  $n+3$  строк определяют функцию правой части на Python
- последняя строка содержит  $n$  чисел - начальное условие

## Выходные данные

---

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

1. значение  $t$
2. значение шага  $h$
3. оценка Рунге  $R$
4. истраченное число вычислений правой части  $N$
5. значения функций решений

## Алгоритм:

```
while t < T and kounter[0] < N_x:
    v_First = euler_Modf(t,v,h)
    v_Second = euler_Modf(t,v,h/2)
    v_Second = euler_Modf(t + h/2, v_Second, h/2)

    R = np.linalg.norm(v_First - v_Second) / (pow(2,2) - 1)

    if R > eps:
        h /= 2

    elif R < (eps / 64):
        h *= 2

    else:
        v = v_First
        t += h

    if t + h > T:
        h = T - t
```

Результат:

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

Eps = 0.0001						
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
1.750000	0.050000	1.28171e-05	24	0.909992	1.158775	2.568019
1.800000	0.050000	1.34742e-05	30	0.893138	1.192634	2.699768
1.850000	0.050000	1.41654e-05	36	0.876652	1.227187	2.838267
1.900000	0.050000	1.48925e-05	42	0.860527	1.262426	2.983862
1.950000	0.050000	1.56573e-05	48	0.844756	1.298342	3.136916
2.000000	0.050000	1.64617e-05	54	0.829333	1.334924	3.297812
2.050000	0.050000	1.73079e-05	60	0.814253	1.372157	3.466951
2.100000	0.050000	1.81979e-05	66	0.799508	1.410026	3.644757
2.150000	0.050000	1.91341e-05	72	0.785092	1.448511	3.831672
2.200000	0.050000	2.01189e-05	78	0.771001	1.487590	4.028165
2.250000	0.050000	2.11548e-05	84	0.757227	1.527236	4.234726
2.300000	0.050000	2.22444e-05	90	0.743766	1.567420	4.451871
2.350000	0.050000	2.33906e-05	96	0.730612	1.608110	4.680143
2.400000	0.050000	2.45962e-05	102	0.717758	1.649267	4.920111
2.450000	0.050000	2.58644e-05	108	0.705200	1.690849	5.172377
2.500000	0.050000	2.71984e-05	114	0.692932	1.732810	5.437568

Eps = 1e-05

1.500000	0.100000	0	0	1.000000	1.000000	2.000000
1.525000	0.025000	1.30836e-06	18	0.990551	1.015087	2.050701
1.550000	0.025000	1.34140e-06	24	0.981204	1.030350	2.102684
1.575000	0.025000	1.37528e-06	30	0.971958	1.045788	2.155982
1.600000	0.025000	1.41003e-06	36	0.962811	1.061402	2.210629
1.625000	0.025000	1.44568e-06	42	0.953763	1.077191	2.266658
1.650000	0.025000	1.48224e-06	48	0.944813	1.093156	2.324105
1.675000	0.025000	1.51973e-06	54	0.935960	1.109297	2.383005
1.700000	0.025000	1.55818e-06	60	0.927203	1.125613	2.443396
1.725000	0.025000	1.59762e-06	66	0.918541	1.142104	2.505315
1.750000	0.025000	1.63807e-06	72	0.909974	1.158770	2.568800
1.775000	0.025000	1.67955e-06	78	0.901500	1.175611	2.633891
1.800000	0.025000	1.72210e-06	84	0.893120	1.192625	2.700630
1.825000	0.025000	1.76574e-06	90	0.884831	1.209813	2.769057
1.850000	0.025000	1.81049e-06	96	0.876633	1.227172	2.839216
1.875000	0.025000	1.85639e-06	102	0.868525	1.244703	2.911150
1.900000	0.025000	1.90347e-06	108	0.860507	1.262405	2.984904
1.925000	0.025000	1.95175e-06	114	0.852577	1.280275	3.060524
1.950000	0.025000	2.00127e-06	120	0.844736	1.298313	3.138058
1.975000	0.025000	2.05206e-06	126	0.836981	1.316517	3.217554
2.000000	0.025000	2.10414e-06	132	0.829313	1.334886	3.299062
2.025000	0.025000	2.15757e-06	138	0.821730	1.353417	3.382632
2.050000	0.025000	2.21236e-06	144	0.814231	1.372109	3.468317
2.075000	0.025000	2.26855e-06	150	0.806817	1.390960	3.556170
2.100000	0.025000	2.32618e-06	156	0.799486	1.409967	3.646247
2.125000	0.025000	2.38529e-06	162	0.792237	1.429128	3.738603
2.150000	0.025000	2.44592e-06	168	0.785070	1.448439	3.833296
2.175000	0.025000	2.50809e-06	174	0.777984	1.467899	3.930386
2.200000	0.025000	2.57186e-06	180	0.770979	1.487503	4.029932
2.225000	0.025000	2.63727e-06	186	0.764052	1.507249	4.131998
2.250000	0.025000	2.70435e-06	192	0.757205	1.527133	4.236647
2.275000	0.025000	2.77315e-06	198	0.750435	1.547151	4.343945
2.300000	0.025000	2.84371e-06	204	0.743743	1.567299	4.453957
2.325000	0.025000	2.91608e-06	210	0.737128	1.587573	4.566754
2.350000	0.025000	2.99030e-06	216	0.730588	1.607968	4.682406
2.375000	0.025000	3.06643e-06	222	0.724124	1.628480	4.800985
2.400000	0.025000	3.14450e-06	228	0.717734	1.649102	4.922564
2.425000	0.025000	3.22458e-06	234	0.711418	1.669831	5.047221
2.450000	0.025000	3.30671e-06	240	0.705176	1.690660	5.175033
2.475000	0.025000	3.39095e-06	246	0.699006	1.711582	5.306079
2.500000	0.025000	3.47734e-06	252	0.692907	1.732592	5.440442

Eps = 1e-06

1.500000	0.100000	0	0	1.000000	1.000000	2.000000
1.512500	0.012500	1.63290e-07	24	0.995263	1.007522	2.025194
1.525000	0.012500	1.65339e-07	30	0.990551	1.015087	2.050705
1.537500	0.012500	1.67414e-07	36	0.985865	1.022697	2.076536
1.550000	0.012500	1.69515e-07	42	0.981204	1.030350	2.102692
1.562500	0.012500	1.71643e-07	48	0.976568	1.038047	2.129177
1.575000	0.012500	1.73799e-07	54	0.971957	1.045788	2.155995
1.587500	0.012500	1.75981e-07	60	0.967371	1.053573	2.183149
1.600000	0.012500	1.78192e-07	66	0.962810	1.061402	2.210646
1.612500	0.012500	1.80430e-07	72	0.958274	1.069274	2.238488
1.625000	0.012500	1.82697e-07	78	0.953762	1.077191	2.266680
1.637500	0.012500	1.84993e-07	84	0.949275	1.085151	2.295226
1.650000	0.012500	1.87318e-07	90	0.944812	1.093156	2.324132
1.662500	0.012500	1.89673e-07	96	0.940373	1.101204	2.353401
1.675000	0.012500	1.92057e-07	102	0.935959	1.109296	2.383037
1.687500	0.012500	1.94472e-07	108	0.931568	1.117433	2.413047
1.700000	0.012500	1.96918e-07	114	0.927202	1.125612	2.443433
1.712500	0.012500	1.99395e-07	120	0.922859	1.133836	2.474202
1.725000	0.012500	2.01903e-07	126	0.918540	1.142104	2.505358

...

2.262500	0.012500	3.46121e-07	384	0.753808	1.537111	4.290211
2.275000	0.012500	3.50497e-07	390	0.750433	1.547136	4.344202
2.287500	0.012500	3.54928e-07	396	0.747077	1.557193	4.398872
2.300000	0.012500	3.59416e-07	402	0.743741	1.567282	4.454230
2.312500	0.012500	3.63961e-07	408	0.740423	1.577403	4.510284
2.325000	0.012500	3.68564e-07	414	0.737125	1.587555	4.567043
2.337500	0.012500	3.73226e-07	420	0.733846	1.597737	4.624515
2.350000	0.012500	3.77947e-07	426	0.730585	1.607948	4.682710
2.362500	0.012500	3.82728e-07	432	0.727344	1.618189	4.741638
2.375000	0.012500	3.87570e-07	438	0.724121	1.628458	4.801306
2.387500	0.012500	3.92474e-07	444	0.720917	1.638755	4.861725
2.400000	0.012500	3.97440e-07	450	0.717731	1.649079	4.922903
2.412500	0.012500	4.02469e-07	456	0.714564	1.659429	4.984851
2.425000	0.012500	4.07563e-07	462	0.711415	1.669805	5.047578
2.437500	0.012500	4.12721e-07	468	0.708285	1.680206	5.111094
2.450000	0.012500	4.17945e-07	474	0.705173	1.690631	5.175409
2.462500	0.012500	4.23235e-07	480	0.702078	1.701080	5.240532
2.475000	0.012500	4.28593e-07	486	0.699002	1.711552	5.306475
2.487500	0.012500	4.34019e-07	492	0.695944	1.722045	5.373246
2.500000	0.012500	4.39514e-07	498	0.692904	1.732559	5.440858

Eps = 1e-07

1.500000	0.100000	0	0	1.000000	1.000000	2.000000
1.506250	0.006250	2.03954e-08	30	0.997628	1.003755	2.012558
1.512500	0.006250	2.05230e-08	36	0.995263	1.007522	2.025194
1.518750	0.006250	2.06513e-08	42	0.992904	1.011299	2.037910
1.525000	0.006250	2.07805e-08	48	0.990551	1.015087	2.050705
1.531250	0.006250	2.09105e-08	54	0.988205	1.018887	2.063581
1.537500	0.006250	2.10413e-08	60	0.985865	1.022697	2.076537
1.543750	0.006250	2.11729e-08	66	0.983531	1.026518	2.089575
1.550000	0.006250	2.13054e-08	72	0.981204	1.030350	2.102694
1.556250	0.006250	2.14388e-08	78	0.978883	1.034193	2.115895
1.562500	0.006250	2.15729e-08	84	0.976568	1.038047	2.129179
1.568750	0.006250	2.17079e-08	90	0.974259	1.041912	2.142547
1.575000	0.006250	2.18438e-08	96	0.971957	1.045788	2.155998
1.581250	0.006250	2.19806e-08	102	0.969661	1.049675	2.169533
1.587500	0.006250	2.21182e-08	108	0.967371	1.053573	2.183153
1.593750	0.006250	2.22566e-08	114	0.965088	1.057482	2.196859
1.600000	0.006250	2.23960e-08	120	0.962810	1.061402	2.210650
1.606250	0.006250	2.25362e-08	126	0.960539	1.065332	2.224528
1.612500	0.006250	2.26774e-08	132	0.958274	1.069274	2.238493

...

2.387500	0.006250	4.93307e-08	876	0.720916	1.638749	4.861808
2.393750	0.006250	4.96419e-08	882	0.719321	1.643907	4.892303
2.400000	0.006250	4.99550e-08	888	0.717730	1.649072	4.922989
2.406250	0.006250	5.02701e-08	894	0.716145	1.654244	4.953867
2.412500	0.006250	5.05871e-08	900	0.714563	1.659422	4.984939
2.418750	0.006250	5.09062e-08	906	0.712987	1.664607	5.016206
2.425000	0.006250	5.12274e-08	912	0.711415	1.669798	5.047668
2.431250	0.006250	5.15505e-08	918	0.709847	1.674996	5.079328
2.437500	0.006250	5.18757e-08	924	0.708284	1.680199	5.111187
2.443750	0.006250	5.22030e-08	930	0.706726	1.685409	5.143245
2.450000	0.006250	5.25324e-08	936	0.705172	1.690624	5.175504
2.456250	0.006250	5.28638e-08	942	0.703622	1.695845	5.207965
2.462500	0.006250	5.31974e-08	948	0.702078	1.701073	5.240630
2.468750	0.006250	5.35330e-08	954	0.700537	1.706305	5.273499
2.475000	0.006250	5.38708e-08	960	0.699002	1.711544	5.306575
2.481250	0.006250	5.42108e-08	966	0.697470	1.716788	5.339857
2.487500	0.006250	5.45529e-08	972	0.695943	1.722037	5.373349
2.493750	0.006250	5.48972e-08	978	0.694421	1.727291	5.407050
2.500000	0.006250	5.52436e-08	984	0.692903	1.732551	5.440963

Eps = 1e-08

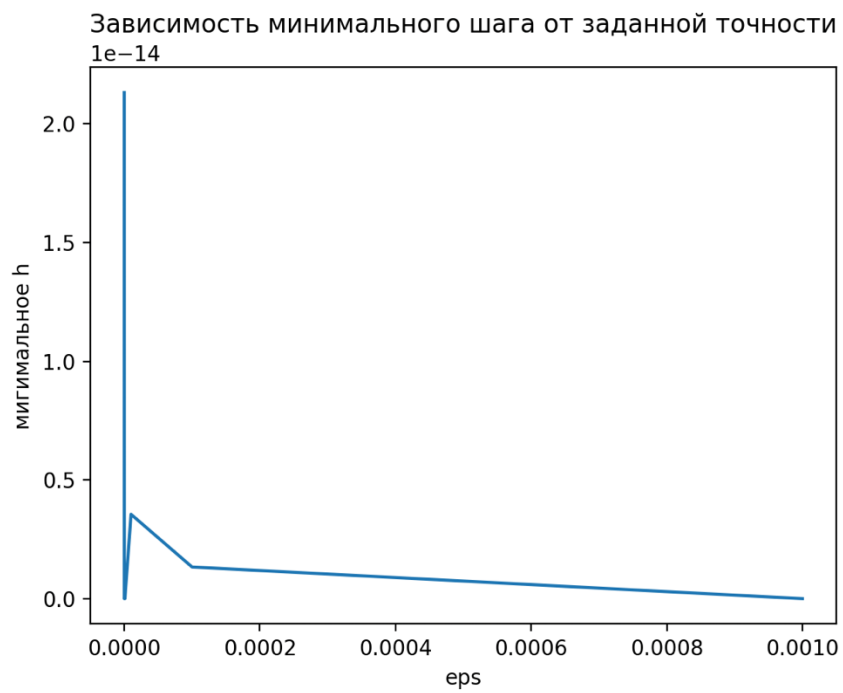
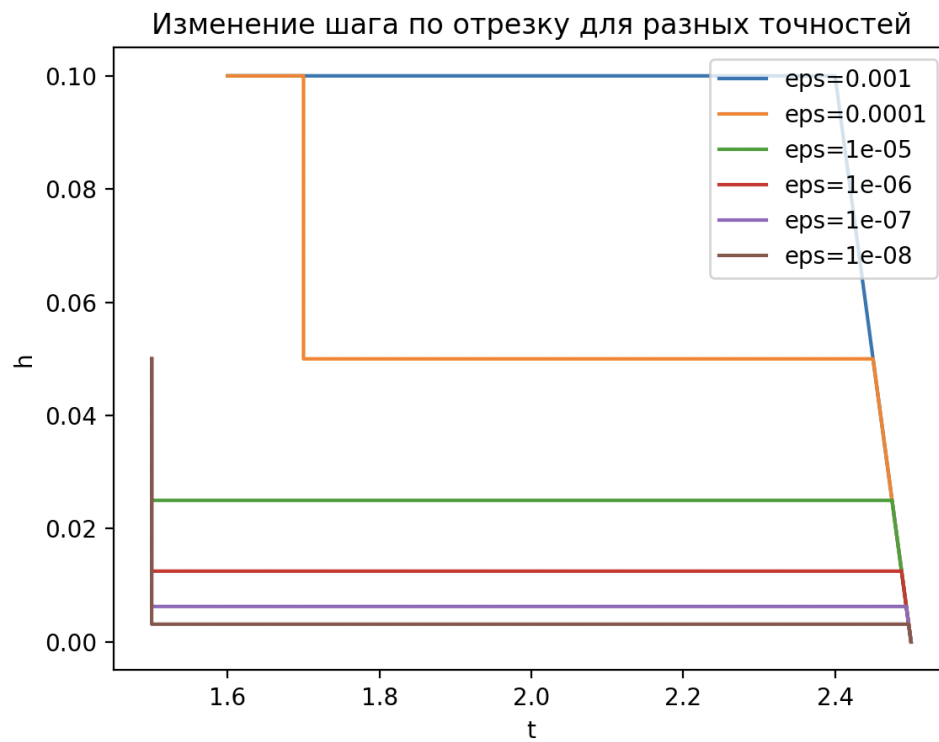
1.500000	0.100000	0	0	1.000000	1.000000	2.000000
1.503125	0.003125	2.54843e-09	36	0.998813	1.001876	2.006269
1.506250	0.003125	2.55639e-09	42	0.997628	1.003755	2.012558
1.509375	0.003125	2.56437e-09	48	0.996445	1.005637	2.018866
1.512500	0.003125	2.57238e-09	54	0.995263	1.007522	2.025195
1.515625	0.003125	2.58041e-09	60	0.994082	1.009409	2.031542
1.518750	0.003125	2.58847e-09	66	0.992904	1.011299	2.037910
1.521875	0.003125	2.59655e-09	72	0.991727	1.013192	2.044298
1.525000	0.003125	2.60466e-09	78	0.990551	1.015087	2.050706
1.528125	0.003125	2.61279e-09	84	0.989377	1.016986	2.057134
1.531250	0.003125	2.62095e-09	90	0.988205	1.018887	2.063581
1.534375	0.003125	2.62914e-09	96	0.987034	1.020790	2.070049
1.537500	0.003125	2.63735e-09	102	0.985865	1.022697	2.076538
1.540625	0.003125	2.64559e-09	108	0.984697	1.024606	2.083046
1.543750	0.003125	2.65385e-09	114	0.983531	1.026518	2.089575
1.546875	0.003125	2.66214e-09	120	0.982367	1.028433	2.096125
1.550000	0.003125	2.67046e-09	126	0.981204	1.030350	2.102694
1.553125	0.003125	2.67880e-09	132	0.980042	1.032270	2.109285
1.556250	0.003125	2.68717e-09	138	0.978883	1.034193	2.115896
1.559375	0.003125	2.69557e-09	144	0.977725	1.036119	2.122527
1.562500	0.003125	2.70399e-09	150	0.976568	1.038047	2.129180

...

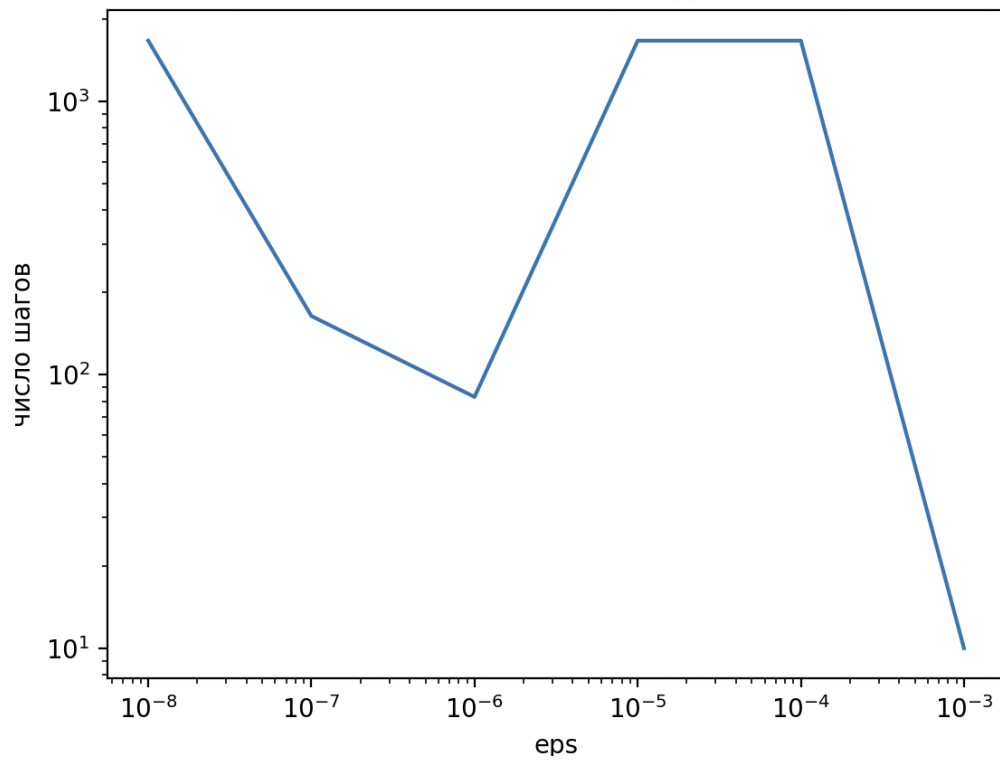
2.443750	0.003125	6.54339e-09	1842	0.706725	1.685407	5.143268
2.446875	0.003125	6.56400e-09	1848	0.705948	1.688014	5.159373
2.450000	0.003125	6.58467e-09	1854	0.705172	1.690622	5.175528
2.453125	0.003125	6.60541e-09	1860	0.704396	1.693232	5.191733
2.456250	0.003125	6.62622e-09	1866	0.703622	1.695844	5.207989
2.459375	0.003125	6.64709e-09	1872	0.702849	1.698456	5.224296
2.462500	0.003125	6.66803e-09	1878	0.702077	1.701071	5.240654
2.465625	0.003125	6.68903e-09	1884	0.701307	1.703686	5.257063
2.468750	0.003125	6.71010e-09	1890	0.700537	1.706304	5.273524
2.471875	0.003125	6.73124e-09	1896	0.699769	1.708922	5.290036
2.475000	0.003125	6.75245e-09	1902	0.699001	1.711542	5.306600
2.478125	0.003125	6.77372e-09	1908	0.698235	1.714163	5.323215
2.481250	0.003125	6.79506e-09	1914	0.697470	1.716786	5.339883
2.484375	0.003125	6.81647e-09	1920	0.696706	1.719410	5.356603
2.487500	0.003125	6.83794e-09	1926	0.695943	1.722035	5.373375
2.490625	0.003125	6.85948e-09	1932	0.695181	1.724661	5.390199
2.493750	0.003125	6.88109e-09	1938	0.694421	1.727289	5.407077
2.496875	0.003125	6.90277e-09	1944	0.693661	1.729918	5.424007
2.500000	0.003125	6.92452e-09	1950	0.692903	1.732549	5.440990



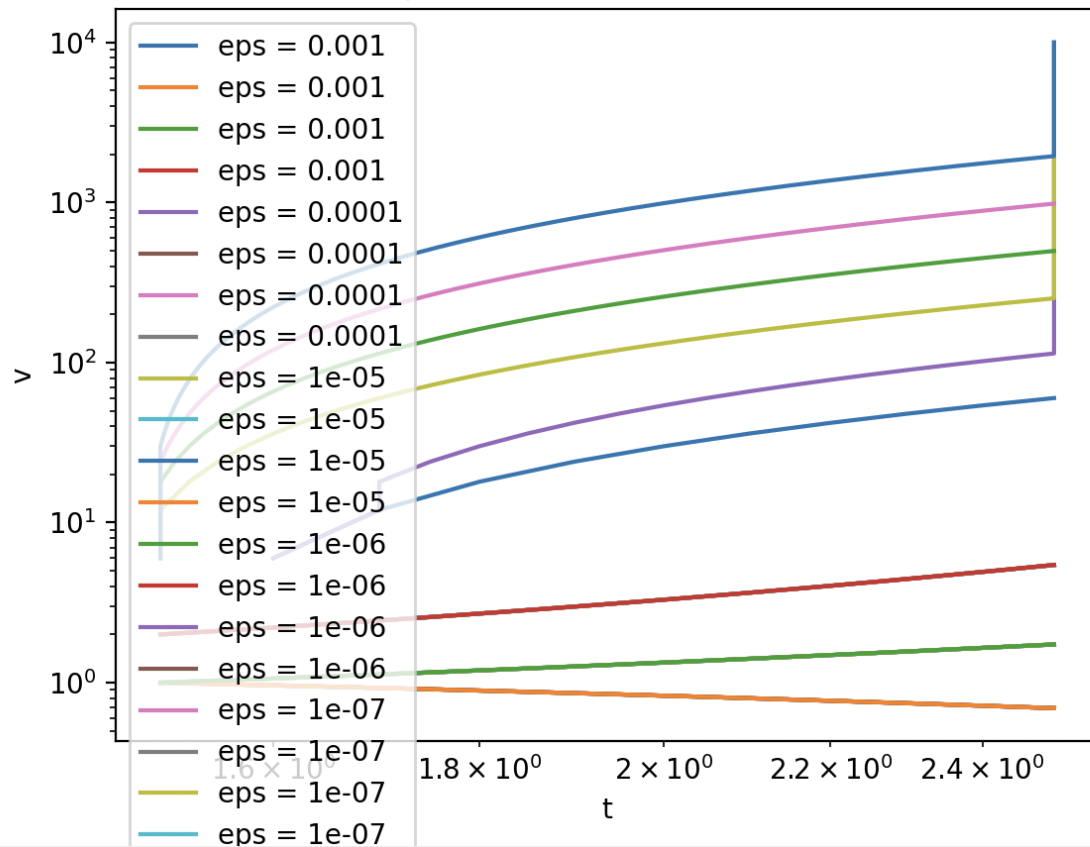
# Графики:



Зависимость числа шагов от заданной точности



Решение для разных значений заданной точности



## Код:

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

def modified_euler_method(rhs_func, initial_condition, t0, T, h0, N_x,
eps):
    t = t0
    h = h0
    v = np.array(initial_condition)
    kounter = [0]
    results = []

    print("{:12.6f} {:12.6f} {:12s} {:12d} {:12.6f} {:12.6f}
{:12.6f}".format(
        t, h, "0", kounter[0], *v))

    def euler_Modf(t,v,h):
        v_hat = rhs_func(t, v, kounter)
        v_tilde = rhs_func(t + h, v + h * v_hat, kounter)
        return v + (h / 2) * (v_hat + v_tilde)

    while t < T and kounter[0] < N_x:
        v_First = euler_Modf(t,v,h)
        v_Second = euler_Modf(t,v,h/2)
        v_Second = euler_Modf(t + h/2, v_Second, h/2)

        R = np.linalg.norm(v_First - v_Second) / (pow(2,2) - 1)

        if R > eps:
            h /= 2

        elif R < (eps / 64):
            h *= 2

        else:
            v = v_First
            t += h

            print("{:12.6f} {:12.6f} {:12.5e} {:12d} {:12.6f} {:12.6f}
{:12.6f}".format(
                t, h, R, kounter[0], *v))

            if t + h > T:
                h = T - t

            results.append((t, h, R, kounter[0], *v))

    return results

t0 = 1.5
T = 2.5
h0 = 0.1
N_x = 10000
eps = 0.0001
```

```

eps_count = [0.001, 0.0001, 0.00001, 0.000001, 0.0000001, 0.00000001]

kounter_mas = []
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)

initial_condition = [1, 1, 2]

result_list = []
for eps in eps_count:
    print("Eps = ", eps)
    results = modified_euler_method(fs, initial_condition, t0, T, h0, N_x,
eps)
    result_list.append(results)
    print(" ")

for i, results in enumerate(result_list):
    t_values = [r[0] for r in results]
    h_values = [r[1] for r in results]
    plt.plot(t_values, h_values, label = f'eps={eps_count[i]}')
plt.xlabel('t')
plt.ylabel('h')
plt.title('Изменение шага по отрезку для разных точностей')
plt.legend()
plt.show()

min_h_values = [min([r[1] for r in results]) for results in result_list]
plt.plot(eps_count, min_h_values)
plt.xlabel('eps')
plt.ylabel('минимальное h')
plt.title('Зависимость минимального шага от заданной точности')
plt.show()

num_steps = [len(results) for results in result_list]
plt.loglog(eps_count, num_steps)
plt.xlabel('eps')
plt.ylabel('число шагов')
plt.title('Зависимость числа шагов от заданной точности')
plt.show()

for i, results in enumerate(result_list):
    t_values = [r[0] for r in results]
    v_values = [r[3:] for r in results]
    plt.loglog(t_values, v_values, label = f'eps = {eps_count[i]}')
plt.xlabel('t')
plt.ylabel('v')
plt.title('Решение для разных значений заданной точности')
plt.legend()
plt.show()

```