Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчёт по лабораторной работе № 3

Дисциплина: Вычислительная математика

Выполнил студент гр. 3530901/10001 _____ Д.Л. Симоновский
(подпись)

Руководитель _____ В.Н. Цыган
(подпись)

“28” февраля 2023 г.

Санкт-Петербург
2023

Оглавление

Задание:

Вариант 11:

Привести дифференциальное уравнение: $ty'' - (t+1)y' - 2(t-1)y = 0$
к системе двух дифференциальных уравнений первого порядка.

Начальные условия: $y(t=1) = e^2;\ y'(t=1) = 2e^2$

Точное решение: $y(t) = e^{2t}$

Решить на интервале: $1 \leq t \leq 2$

1. Используя программу RKF45 с шагом печати h_{print}= 0.1 и выбранной вами погрешностью EPS в диапазоне 0.001 – 0.00001, а также составить собственную программу и решить с шагом интегрирования h_{int} = 0.1
2. Используя метод Рунге-Кутты 3-й степени точности.

Сравнить результаты, полученные заданными приближенными способами, с точным решением.

Исследовать влияние величины шага интегрирования h_{int} на величины локальной и глобальной погрешностей решения заданного уравнения для чего решить уравнение, используя 2 – 3 значения шага интегрирования, существенно меньшие исходной величины 0.1 (например, h_{int} = 0.05; h_{int}= 0.025; h_{int}= 0.0125)

Инструменты:

Для работы был выбран язык программирования Python версии 3.11 по причине удобства его использования для поставленной задачи. Были выбраны следующие библиотеки:

- NumPy – для большей скорости расчетов и простоты обработки
- SciPy – для функции расчета решения дифура
- PrettyTable – для красивого вывода таблицы в консоль
- MatPlotLib – для вывода графиков

Ход выполнения работы:

Порядок действий:

Поставленное задание легко можно разбить на две глобальные задачи:

1. Сведение поставленной задачи к системе двух дифференциальных уравнений первого порядка.
2. Получить решение используя RKF45 и методы Рунге-Кутты 3-й степени

Первая задача:

Полная формулировка:
Привести дифференциальное уравнение: $ty'' - (t+1)y' - 2(t-1)y = 0$
к системе двух дифференциальных уравнений первого порядка.

Решение:

В начале сделаем коэффициент при старшей степени равным 1, для этого поделим уравнение на t:

$$y'' - \frac{(t+1)}{t}y' - \frac{2(t-1)}{t}y = 0$$

Возьмем $\alpha_1 = -\frac{(t+1)}{t}$ и $\alpha_2 = -\frac{2(t-1)}{t}$, получим:

$$y'' + \alpha_1 y' + \alpha_2 y = 0$$

Решение этого уравнение эквивалентно решению системы $\frac{dx}{dt} = Ax + f(t)$, где $f(t) = 0$, A – матрица Фробениуса вида: $A = \begin{pmatrix} -\alpha_1 & -\alpha_2 \\ 1 & 0 \end{pmatrix}, x = \begin{pmatrix} x^{(1)} \\ x^{(2)} \end{pmatrix} = \begin{pmatrix} y' \\ y \end{pmatrix}$. Таким образом получим систему:

$$\begin{cases} x^{(2)'} = x^{(1)} \\ x^{(1)'} = -\alpha_1 x^{(1)} - \alpha_2 x^{(2)} \end{cases}$$

Вторая задача:

Полная формулировка:

Решить систему дифференциальных уравнений перового порядка.
Начальные условия: $y(t=1) = e^2;\ y'(t=1) = 2e^2$
Точное решение: $y(t) = e^{2t}$

Решить на интервале: $1 \le t \le 2$

1. Используя программу RKF45 с шагом печати h_{print}= 0.1 и выбранной вами погрешностью EPS в диапазоне 0.001 – 0.00001, а также составить собственную программу и решить с шагом интегрирования h_{int} = 0.1
2. Используя метод Рунге-Кутты 3-й степени точности.

Сравнить результаты, полученные заданными приближенными способами, с точным решением.

Исследовать влияние величины шага интегрирования h_{int} на величины локальной и глобальной погрешностей решения заданного уравнения для чего решить уравнение, используя 2 – 3 значения шага интегрирования, существенно меньшие исходной величины 0.1 (например, h_{int} = 0.05; h_{int}= 0.025; h_{int}= 0.0125)

Решение:

Для дальнейшего использования сразу же зададим функцию, для получения значений системы уравнений:

```
def f(t, X):
    dX = np.zeros(X.shape)
    dX[0] = X[1]
    dX[1] = (t + 1) / t * X[1] + 2 * (t - 1) / t * X[0]
    return dX
```

Так же нам понадобится функция для получения точно значения решения (чтоб сравнивать погрешности):

```
def g(T):
    return np.e ** (2 * T)
```

Далее нам понадобится функция, которая будет моделировать RKF45, к счастью, в библиотеке scipy уже имеется подходящий вариант, правда требующий дополнительной настройки, а конкретно передачу параметра ‘dopri5’ для настройки интегратора и выставления параметра погрешности atol на значение 0.0001:

```
def rkf45(f, T, X0):
      runge = ode(f).set_integrator('dopri5', atol=0.0001).set_initial_value(X0, T[0])
      X = [X0, *[runge.integrate(T[i]) for i in range(1, len(T))]]
      return np.array([i[0] for i in X]), np.array([i[1] for i in X])
```

Метод Рунге-Кутты по заданию необходимо написать самостоятельно. Для начала вспомним, как выглядит метод Рунге-Кутты третьей степени:

$$\begin{cases} x_{n+1} = x_n + \dfrac{2k_1 + 3k_2 + 4k_3}{9}, \\ k_1 = hf(t_n, x_n), \\ k_2 = hf(t_n + \dfrac{h}{2}, x_n + \dfrac{k_1}{2}), \\ k_3 = hf(t_n + \dfrac{3h}{4}, x_n + \dfrac{3k_2}{4}) \end{cases}$$

Теперь необходимо реализовать его в виде метода:

```
def RK3(f, T, X0):
      X = np.zeros((len(T), len(X0)))
      X[0] = X0
      h = T[1] - T[0]
      for i in range(0, len(T) - 1):
          k_1 = h * f(T[i], X[i])
          k_2 = h * f(T[i] + h / 2, X[i] + k_1 / 2)
          k_3 = h * f(T[i] + 3 * h / 4, X[i] + 3 * k_2 / 4)
          X[i + 1] = (X[i] + (2 * k_1 + 3 * k_2 + 4 * k_3) / 9)
      return X[:, 0]
```

Все методы, необходимые для подсчета реализованы. Создадим отдельную функцию для подсчета решения с разным шагом (т.к. этого требует задание), в ней сразу же зададим начальные значения, узлы, по которым будет искаться решение и значение функции в этих точках (для дальнейшего подсчета погрешности):

```
def evaluate(h):
      # Начальные значения
      X0 = np.array([np.e ** 2, 2 * np.e ** 2])
      # Значения в узлах
      T = np.arange(1, 2 + h, h)
      Y = g(T)
```

Воспользовавшись методами, которые были приведены выше, выполним расчеты:

```
  # Расчет RKF45
  Y_RKF45, Y_derivative_RKF45 = rkf45(f, T, X0)
  # Расчет Рунге-Кутты
  Y_Runge_Kutta = Runge_Kutta(f, T, X0)
  # Погрешности
  Y_RKF45_error = Y - Y_RKF45
  Y_Runge_Kutta_error = Y - Y_Runge_Kutta
```

Далее необходимо вывести полученные значения на экран, используем для этого библиотеку MatPlotLib. Отдельно выведем графики погрешности и полученные значения:

```
def print_one_graph(t, y, title, id, count_graphs):
      """
      Функция для отрисовки одного графика
      """
      plt.subplot(1, count_graphs, id)
      plt.xlabel('t')
      plt.ylabel('y')
      plt.grid()
      plt.title(title)
      plt.plot(t, y, '-o')


  def print_graph(t_find, y_real, Y_RKF45, Y_Runge_Kutta, h):
      """
      Функция для отрисовки всех графиков
      """
      mpl.use('TkAgg')
      plt.figure(figsize=(15, 4))
      print_one_graph(t_find, y_real, 'Исходный график', 1, 3)
      print_one_graph(t_find, Y_RKF45, 'График RKF45', 2, 3)
      print_one_graph(t_find, Y_Runge_Kutta, 'График Рунге-Кутты', 3, 3)
      plt.savefig(f"Graphs_{h}.jpg")
      plt.show()


  def print_error_graph(t_find, Y_RKF45_error, Y_Runge_Kutta_error, h):
      """
      Функция для отрисовки погрешности
      """
      mpl.use('TkAgg')
      plt.figure(figsize=(15, 4))
      # Собственно сам график
      print_one_graph(t_find, Y_RKF45_error, 'Погрешность RKF45', 1, 2)
      print_one_graph(t_find, Y_Runge_Kutta_error, 'Погрешность Рунге-Кутты', 2, 2)
      plt.savefig(f"Error_{h}.jpg")
      plt.show()
```

Вызовем их из функции evaluate, для отрисовки всех графиков:

```
# Рисуем графики
print_graph(T, Y, Y_RKF45, Y_Runge_Kutta, h)
print_error_graph(T, Y_RKF45_error, Y_Runge_Kutta_error, h)
```

Далее для дополнительного анализа выведем значения в консоль, используя библиотеку prettytable:

```
def print_table(t_find, y_real, Y_RKF45, Y_RKF45_error, Y_Runge_Kutta, Y_Runge_Kutta_error,
  h):
      print(f'h = {h}')
      koef = {0.1: 1, 0.05: 2, 0.025: 4, 0.0125: 8}.get(h)
      pt = PrettyTable()
      pt.add_column('t', [f'{i:.1f}' for i in t_find[::koef]])
      pt.add_column('real y', [f'{i:.15f}' for i in y_real[::koef]])
      pt.add_column('RKF45 y', [f'{i:.15f}' for i in Y_RKF45[::koef]])
      pt.add_column('Delta RKF45 y', [f'{i:.15f}' for i in Y_RKF45_error[::koef]])
      pt.add_column('Runge Kutta y', [f'{i:.15f}' for i in Y_Runge_Kutta[::koef]])
      pt.add_column('Delta Runge Kutta y', [f'{i:.15f}' for i in
  Y_Runge_Kutta_error[::koef]])
      print(pt)
      print('First step of RKF45:', Y_RKF45_error[1])
      print('First step of Runge Kutta:', Y_Runge_Kutta_error[1])
      print('Global of RKF45:', Y_RKF45_error[::koef].sum())
      print('Global of Runge Kutta:', Y_Runge_Kutta_error[::koef].sum())
      print('h^4 is about:', h ** 4)
      print('h^4 / Runge Kutta first step:', h ** 4 / Y_Runge_Kutta_error[1])
```

Еще необходимо выполнить анализ погрешности, для этого будем сохранять её при вызове функции evaluate, а после выведем все это на экран в виде таблицы, используя уже известную библиотеку:

```
def print_table_error(Y_RKF45_error, Y_Runge_Kutta_error, h_list):
    pt = PrettyTable()
    pt.add_column('h', [f'{i:.4f}' for i in h_list])
    pt.add_column("Runge Kutta Error local", [f'{i[1]:.15f}' for i in Y_Runge_Kutta_error])
    pt.add_column('h**4 / Runge Kutta Error local',
                  [f'{i ** 4 / j[1]:.15f}' for i, j in zip(h_list, Y_Runge_Kutta_error)])
    print(pt)
    pt.clear()
    pt.add_column('h', [f'{i:.4f}' for i in h_list])
    pt.add_column("Runge Kutta Error global", [f'{i.sum():.15f}' for i in
Y_Runge_Kutta_error])
    pt.add_column('h**2 / Runge Kutta Error global',
                  [f'{i ** 2 / j.sum():.15f}' for i, j in zip(h_list, Y_Runge_Kutta_error)])
    print(pt)
    pt.clear()
    pt.add_column('h', [f'{i:.4f}' for i in h_list])
    pt.add_column("RKF45 Error local", [f'{i[1]:.15f}' for i in Y_RKF45_error])
    pt.add_column("RKF45 Error global", [f'{i.sum():.15f}' for i in Y_RKF45_error])
    pt.add_column("Runge Kutta Error local", [f'{i[1]:.15f}' for i in Y_Runge_Kutta_error])
    pt.add_column("Runge Kutta Error global", [f'{i.sum():.15f}' for i in
Y_Runge_Kutta_error])
    print(pt)
```

Результат:

Как и ожидалось, полученные графики практически не отличаются:

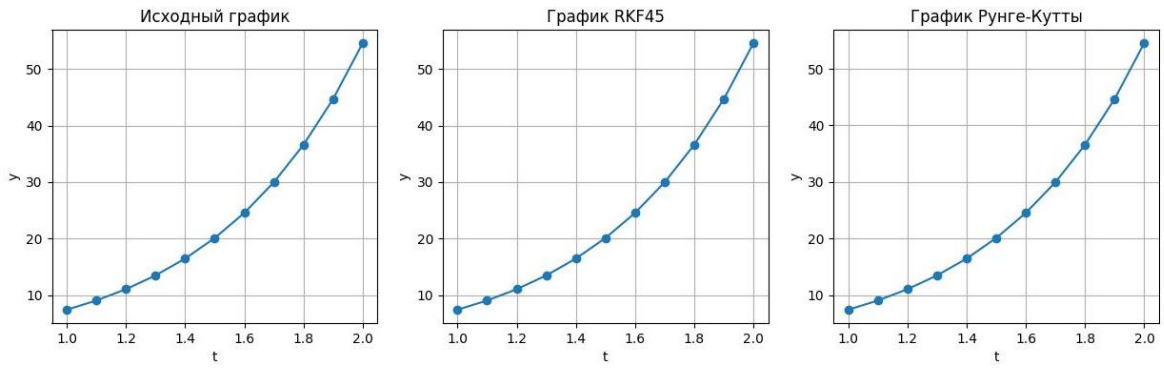

Рис. 1. График решения при h = 0.1

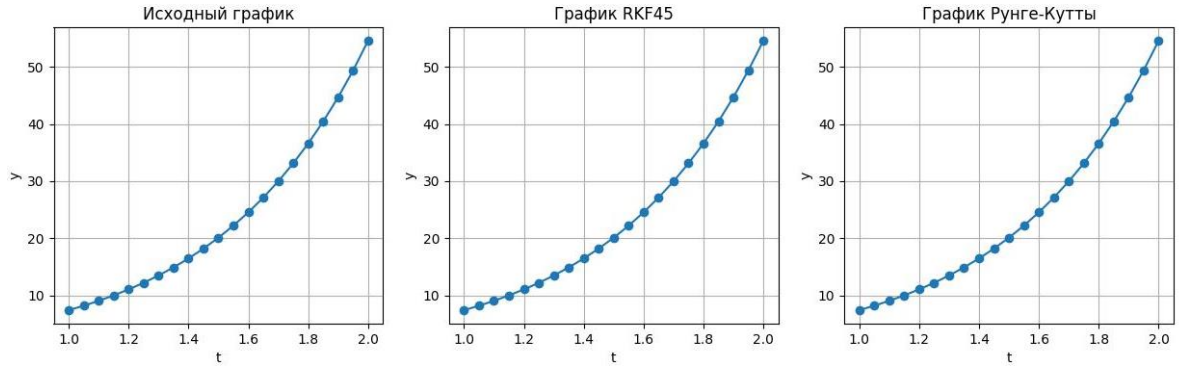

Рис. 2. График решения при h = 0.05

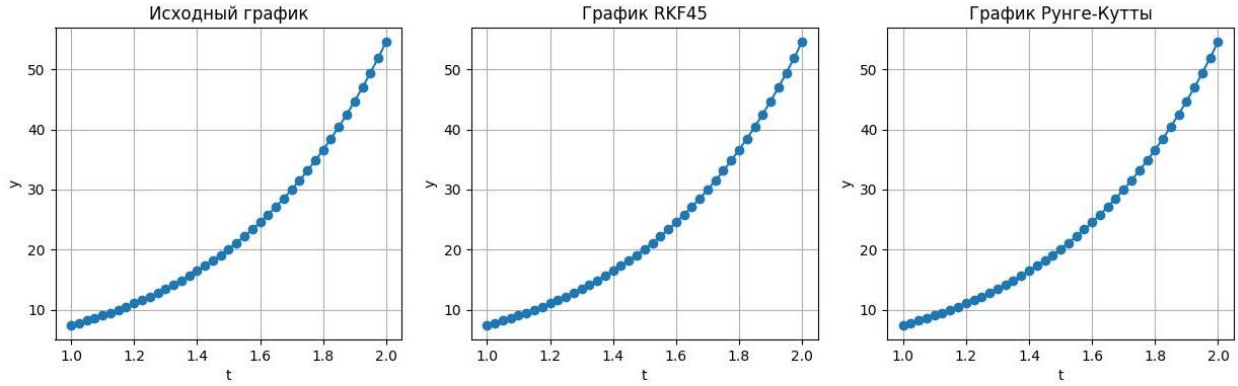

Рис. 3. График решения при h = 0.025

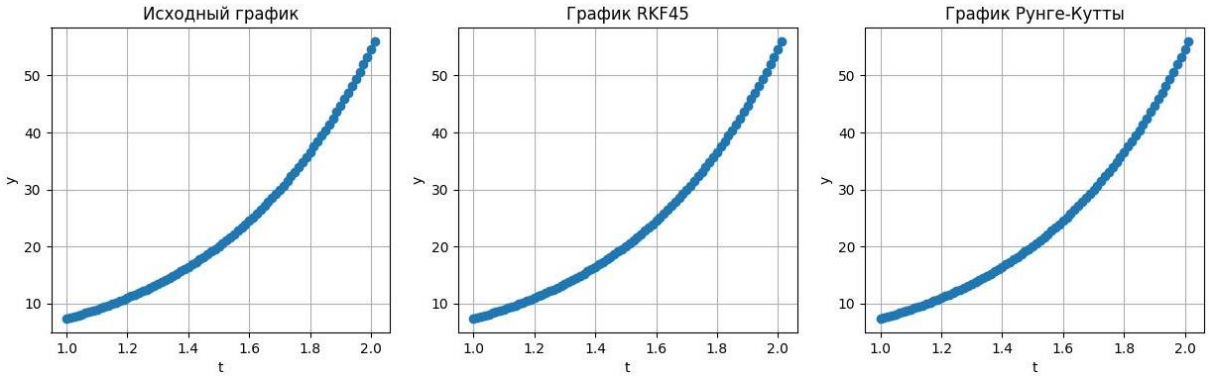

Рис. 4. График решения при h = 0.0125

Больший интерес представляют графики погрешности:

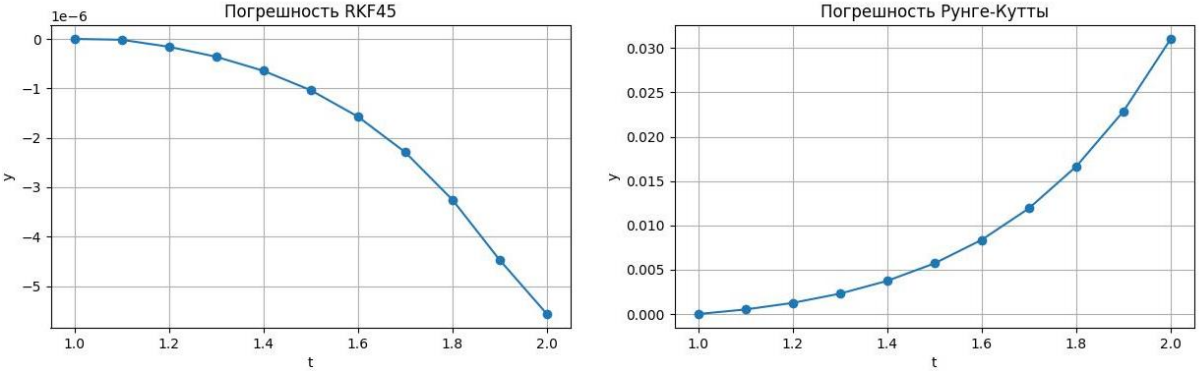

Рис. 5. График погрешности при h = 0.1

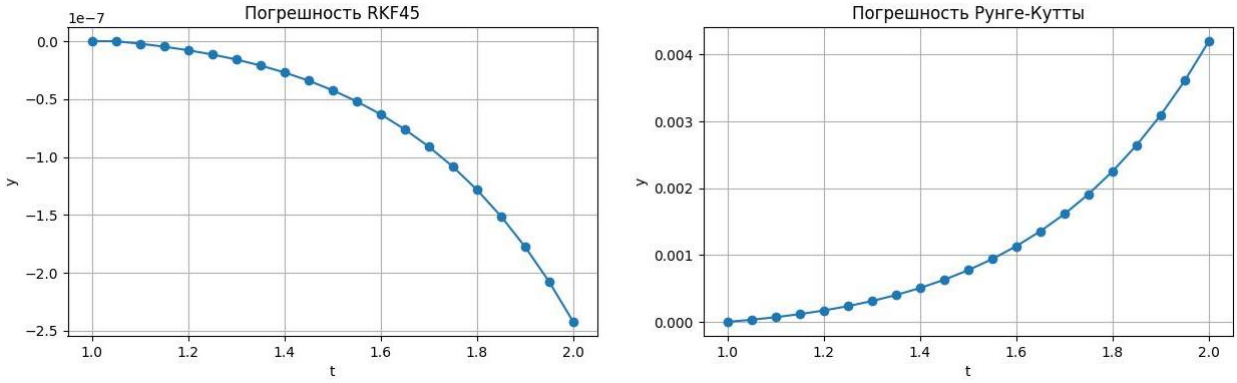

Рис. 6. График погрешности при h = 0.05

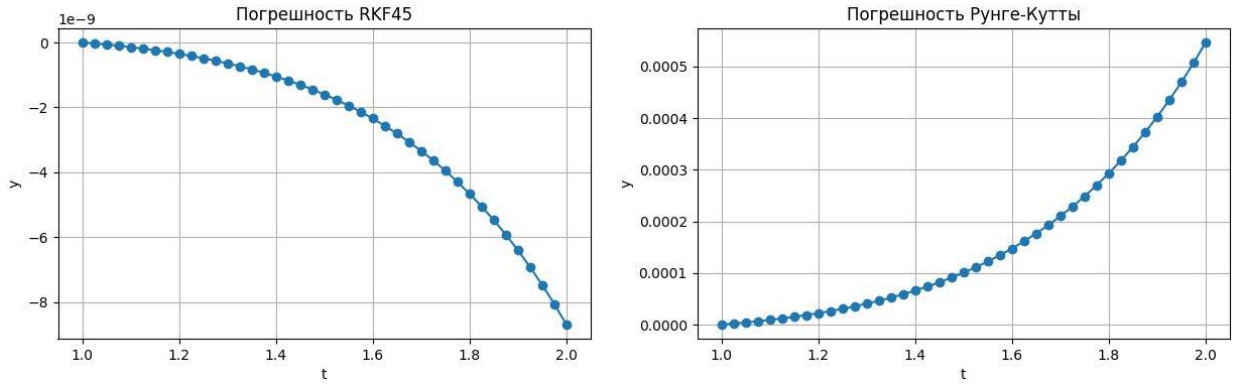

Рис. 7. График погрешности при h = 0.025

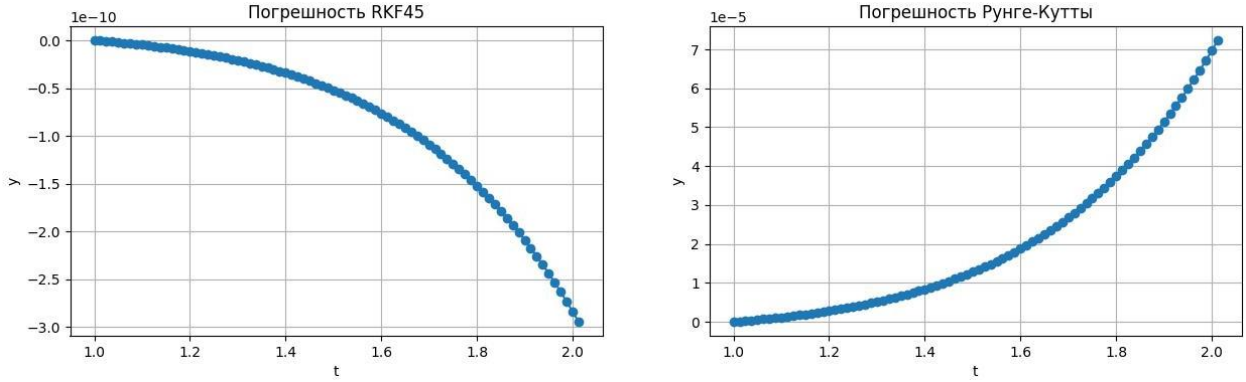

Рис. 8. График погрешности при h = 0.0125

Как можно заметить, форма самого графика остается одинаковой, однако порядок сильно меняется, при изменении шага.

Так же даже по графику заметно, что погрешность метода Рунге-Кутты третьей степени значительно проигрывает программе RKF45, что достаточно ожидаемо, ведь в основе этой программы лежат методы Рунге-Кутты четвертой и пятой степени точности.

Для анализа зависимости шага интегрирования h_{int} и величины локальной и глобальной погрешностей решения заданного уравнения обратимся к численным результатам исследования:

h = 0.1

t	real y	RKF45 y	Delta RKF45 y	Runge Kutta y	Delta Runge Kutta y
1.0	7.389056098930650	7.389056098930650	0.000000000000000	7.389056098930650	0.000000000000000
1.1	9.025013499434122	9.025013516869686	-0.000000017435564	9.024500515493969	0.000512983940153
1.2	11.023176380641605	11.023176538875457	-0.000000158233853	11.021923296256636	0.001253084384969
1.3	13.463738035001695	13.463738395525420	-0.000000360523725	13.461442319161440	0.002295715840255
1.4	16.444646771097059	16.444647415729236	-0.000000644632177	16.440908219135842	0.003738551961217
1.5	20.085536923187682	20.085537960060527	-0.000001036872845	20.079829238304580	0.005707684883102
1.6	24.532530197109370	24.532531768309873	-0.000001571200503	24.524164776382666	0.008365420726705
1.7	29.964100047397046	29.964102338701789	-0.000002291304742	29.952179913555369	0.011920133841677
1.8	36.598234443678031	36.598237696934191	-0.000003253256160	36.581595734422294	0.016638709255737
1.9	44.701184493300886	44.701188964687368	-0.000004471386482	44.678322256974440	0.022862236326446
2.0	54.598150033144321	54.598155598718769	-0.000005565574448	54.567124249851460	0.031025783292861

First step of RKF45: -1.743556410360725e-08
First step of Runge Kutta: 0.0005129839401529779
Global of RKF45: -1.9370420499598l4e-05
Global of Runge Kutta: 0.10432030445312179
h^4 is about: 0.00010000000000000002
h^4 / Runge Kutta first step: 0.1949378765545347

Рис. 9. Результат исследования при h = 0.1

h = 0.05

t	real y	RKF45 y	Delta RKF45 y	Runge Kutta y	Delta Runge Kutta y
1.0	7.389056098930650	7.389056098930650	0.000000000000000	7.389056098930650	0.000000000000000
1.1	9.025013499434122	9.025013501625160	-0.000000002191038	9.024944064111462	0.000069435322660
1.2	11.023176380641605	11.023176388458154	-0.000000007816549	11.023006764304872	0.000169616336732
1.3	13.463738035001695	13.463738050827361	-0.000000015825666	13.463427281404844	0.000310753596851
1.4	16.444646771097059	16.444646798095160	-0.000000026998102	16.444140699310093	0.000506071786965
1.5	20.085536923187682	20.085536965529673	-0.000000042341991	20.084764279314374	0.000772643873308
1.6	24.532530197109370	24.532530260266185	-0.000000063156815	24.531397750236174	0.001132446873196
1.7	29.964100047397046	29.964100138510030	-0.000000091112984	29.962486350914514	0.001613696482533
1.8	36.598234443678031	36.598234572030442	-0.000000128352411	36.595981911389273	0.002252532288757
1.9	44.701184493300886	44.701184670916255	-0.000000177615370	44.698089349913182	0.003095143387704
2.0	54.598150033144321	54.598150275544839	-0.000000242400517	54.593949586335228	0.004200446809094

First step of RKF45: -7.847944516470307e-11
First step of Runge Kutta: 3.1413899460375205e-05
Global of RKF45: -1.4647554884561487e-06
Global of Runge Kutta: 0.025995142229312762
h^4 is about: 6.250000000000001e-06
h^4 / Runge Kutta first step: 0.1989565162988954

Рис. 10. Результат исследования при h = 0.05

h = 0.025

t	real y	RKF45 y	Delta RKF45 y	Runge Kutta y	Delta Runge Kutta y
1.0	7.389056098930650	7.389056098930650	0.000000000000000	7.389056098930650	0.000000000000000
1.1	9.025013499434113	9.025013499577801	-0.000000000143688	9.025004466700494	0.000009032733619
1.2	11.023176380641585	11.023176380992583	-0.000000000350997	11.023154315441113	0.000022065200472
1.3	13.463738035001660	13.463738035644724	-0.000000000643064	13.463697609276819	0.000040425724841
1.4	16.444646771097002	16.444646772144257	-0.000000001047255	16.444580936340852	0.000065834756150
1.5	20.085536923187593	20.085536924786492	-0.000000001598899	20.085436409796969	0.000100513390624
1.6	24.532530197109239	24.532530199452719	-0.000000002343480	24.532382876383917	0.000147320725322
1.7	29.964100047396858	29.964100050736249	-0.000000003339391	29.963890119904967	0.000209927491891
1.8	36.598234443677775	36.598234448339191	-0.000000004661416	36.597941408375718	0.000293035302057
1.9	44.701184493300531	44.701184499705683	-0.000000006405152	44.700781840110089	0.000402653190442
2.0	54.598150033143838	54.598150041836362	-0.000000008692524	54.597603587064640	0.000546446079198

First step of RKF45: -3.091749078976136e-11
First step of Runge Kutta: 1.9436371978542866e-06
Global of RKF45: -1.0308740350239987e-07
Global of Runge Kutta: 0.006480486576304401
h^4 is about: 3.9062500000000007e-07
h^4 / Runge Kutta first step: 0.2009762935342242

Рис. 11. Результат исследования при h = 0.025

h = 0.0125

t	real y	RKF45 y	Delta RKF45 y	Runge Kutta y	Delta Runge Kutta y
1.0	7.389056098930650	7.389056098930650	0.000000000000000	7.389056098930650	0.000000000000000
1.1	9.025013499434113	9.025013499438806	-0.000000000004693	9.025012347561770	0.000001151872343
1.2	11.023176380641585	11.023176380653045	-0.000000000011459	11.023173566841647	0.000002813799938
1.3	13.463738035001660	13.463738035022654	-0.000000000020995	13.463732879827486	0.000005155174174
1.4	16.444646771097002	16.444646771131190	-0.000000000034188	16.444638375705598	0.000008395391404
1.5	20.085536923187593	20.085536923239793	-0.000000000052200	20.085524105495647	0.000012817691946
1.6	24.532530197109239	24.532530197185743	-0.000000000076504	24.532511410433283	0.000018786675955
1.7	29.964100047396858	29.964100047505873	-0.000000000109015	29.964073276951098	0.000026770445761
1.8	36.598234443677775	36.598234443829952	-0.000000000152177	36.598197075112964	0.000037368564811
1.9	44.701184493300531	44.701184493509636	-0.000000000209106	44.701133145977167	0.000051347323364
2.0	54.598150033143838	54.598150033427601	-0.000000000283762	54.598080348967869	0.000069684175969

First step of RKF45: -4.920508445138694e-13
First step of Runge Kutta: 1.2086842193781422e-07
Global of RKF45: -6.880966907374386e-09
Global of Runge Kutta: 0.001689689679643891
h^4 is about: 2.4414062500000004e-08
h^4 / Runge Kutta first step: 0.201988758590402

Рис. 12. Результат исследования при h = 0.0125

Можно заметить, что локальная погрешность первого шага метода Рунге-Кутты пропорциональна четвертой степени шага интегрирования, как и было предсказано:

h	Runge Kutta Error local	h**4 / Runge Kutta Error local
0.1000	0.000512983940153	0.194937876554535
0.0500	0.000031413899460	0.198956516298895
0.0250	0.000001943637198	0.200976293534224
0.0125	0.000000120868422	0.201988758590402

Из анализа результатов глобальной погрешности заметна её зависимость от квадрата шага интегрирования, что явно сильно хуже, чем если рассматривать локальную погрешность:

h	Runge Kutta Error global	h**2 / Runge Kutta Error global
0.1000	0.104320304453122	0.095858615946560
0.0500	0.025995142229313	0.096171814639311
0.0250	0.006480486576304	0.096443375453517
0.0125	0.001689689679644	0.092472601260683

Так же из приведенных результатов (рис. 5-12) более заметна разница между методами Рунге-Кутты 3 степени точности и программой RKF45, как после первого шага, так и при подсчете глобальной погрешности, что сильнее проявляется при уменьшении шага:

h	RKF45 Error local	RKF45 Error global	Runge Kutta Error local	Runge Kutta Error global
0.1000	-0.000000017435564	-0.000019370420500	0.000512983940153	0.104320304453122
0.0500	-0.000000000078479	-0.000001464755488	0.000031413899460	0.025995142229313
0.0250	-0.000000000030917	-0.000000103087404	0.000001943637198	0.006480486576304
0.0125	-0.000000000000492	-0.000000006880967	0.000000120868422	0.001689689679644

Замечание:

В ходе анализа замечено, что знак погрешности у метода Рунге-Кутты 3 степени отличается от знака погрешности аналога программы RKF45, что сомнительно, поскольку функция e^{2t} а также её первая производная $2e^{2t}$ являются монотонно растущими функциями.

В ходе проверки кода ошибки замечено не было.

Данные, полученные в ходе решения методом Рунге-Кутты 3 степени кажутся более правдоподобными, потому дополнительно проанализируем аналог программы RKF45. Для этого дополнительно проанализируем значение первой производной, полученное в ходе решения, для этого выведем её на экран:

```
def print_additiontal_research(T, Y_derivative_real, Y_derivative_RKF45):
      pt = PrettyTable()
      pt.add_column("T", [f'{i:.4f}' for i in T])
      pt.add_column("Y' real", Y_derivative_real)
      pt.add_column("Y' RKF45", Y_derivative_RKF45)
      pt.add_column("Y' delta", Y_derivative_real - Y_derivative_RKF45)
      print(pt)
      print('=' * 110)
```

Результат:

T	Y' real	Y' RKF45	Y' delta
1.0000	14.778112197861299	14.778112197861299	0.0
1.1000	18.050026998868244	18.050027033739372	-3.48711282072145e-08
1.2000	22.04635276128321	22.046353077750915	-3.1646770537463453e-07
1.3000	26.92747607000339	26.92747679105084	-7.21047449303569e-07
1.4000	32.88929354219412	32.88929483145847	-1.2892643539430537e-06
1.5000	40.171073846375364	40.171075920121055	-2.073745690722717e-06
1.6000	49.06506039421874	49.06506353661975	-3.1424010131786417e-06
1.7000	59.92820009479409	59.928204677403585	-4.58260949187661e-06
1.8000	73.19646888735606	73.19647539386838	-6.50651232092514e-06
1.9000	89.40236898660177	89.40237792937474	-8.942772964815049e-06
2.0000	109.19630006628864	109.19631119743752	-1.113114888084965e-05

Рис. 13. Результат исследования при h = 0.1

T	Y' real	Y' RKF45	Y' delta
1.0000	14.778112197861299	14.778112197861299	0.0
1.0500	16.3323398251353	16.33233982529226	-1.5695889032940613e-10
1.1000	18.050026998868244	18.05002700325032	-4.382076923548084e-09
1.1500	19.948364909629444	19.948364919123623	-9.494179664670 81e-09
1.2000	22.04635276128321	22.046352776916308	-1.5633098371381493e-08
1.2500	24.364987921406954	24.364987944365232	-2.29582788335847e-08
1.3000	26.92747607000339	26.927476101654722	-3.165133222182703e-08
1.3500	29.75946344974568	29.75946349166464	-4.1918958970654785e-08
1.4000	32.88929354219412	32.88929359619032	-5.399620306434372e-08
1.4500	36.34829073888614	36.34829080703628	-6.815013620098398e-08
1.5000	40.171073846375364	40.17107393105935	-8.46839824930612e-08
1.5500	44.3959025628833	44.39590266682509	-1.039417867332304e-07
1.6000	49.06506039421874	49.06506052053237	-1.2631362977799654e-07
1.6500	54.22527784131583	54.22527799355734	-1.5224151184156653e-07
1.7000	59.92820009479409	59.92820027702006	-1.8222596764871923e-07
1.7500	66.23090391738471	66.23090413421819	-2.1683348450096673e-07
1.8000	73.19646888735606	73.19646914406088	-2.5670482273199013e-07
1.8500	80.89460872013488	80.89460902269927	-3.025643877663242e-07
1.9000	89.40236898660177	89.40236934183251	-3.5523073904641933e-07
1.9500	98.80489821106049	98.8048986266889	-4.156284063583371e-07
2.0000	109.19630006628864	109.19630055108968	-4.848010348723619e-07

Рис. 14. Результат исследования при h = 0.05

T	Y' real	Y' RKF45	Y' delta
1.0000	14.778112197861299	14.778112197861299	0.0
1.0250	15.53580221261354	15.535802212675375	-6.183498157952272e-11
1.0500	16.332339825135293	16.332339825265304	-1.3001155707570433e-10
1.0750	17.169716794355775	17.169716794560795	-2.050200009762193e-10
1.1000	18.050026998868226	18.0500269991556	-2.873754567644937e-10
1.1250	18.97547167271703	18.975471673094667	-3.7763570048809925e-10
1.1500	19.94836490962942	19.948364910105813	-4.763940353313956e-10
1.1750	20.97113944945512	20.97113945003941	-5.842899497565668e-10
1.2000	22.04635276128317	22.046352761985165	-7.019949066489062e-10
1.2250	23.176693438446737	23.176693439276974	-8.302372123125679e-10
1.2500	24.3649879214069	24.36498792237668	-9.69780700188494e-10
1.2750	25.614207565326012	25.61420756644746	-1.1214495998501661e-09
1.3000	26.92747607000332	26.927476071289448	-1.2861285370036057e-09
1.3250	28.308077290751537	28.30807729221628	-1.464741217205301e-09
1.3500	29.75946344974559	29.75946345140388	-1.6582930584263522e-09
1.3750	31.285263768376254	31.285263770244093	-1.8678392166293634e-09
1.4000	32.889293542194004	32.88929354428851	-2.094509454764193e-09
1.4250	34.57556368113517	34.575563683474684	-2.339511695481633e-09
1.4500	36.348290738886	36.34829074149014	-2.604139126560767e-09
1.4750	38.21190745646316	38.21190745935291	-2.889748884626897e-09
1.5000	40.17107384637519	40.171073849572984	-3.197797582288331e-09
1.5250	42.23068884508106	42.2306888486109	-3.529841308136383 3e-09
1.5500	44.39590256288309	44.395902566770616	-3.887528521318018e-09
1.5750	46.67212916188523	46.672129166157845	-4.272614262390562e-09
1.6000	49.06506039421848	49.065060398905445	-4.686967258749064e-09
1.6250	51.58067983438588	51.58067983951846	-5.132577030053653e-09
1.6500	54.22527784131552	54.22527784692707	-5.611553888229537e-09
1.6750	57.00546728753427	57.00546729366043	-6.126157359176432e-09
1.7000	59.928200094793716	59.9282001014725	-6.67878197191385e-09
1.7250	63.000784617495526	63.00078462476749	-7.271964364008454e-09
1.7500	66.23090391738427	66.23090392529268	-7.908411703283491e-09
1.7750	69.62663497520364	69.62663498379467	-8.59103010952822e-09
1.8000	73.19646888735555	73.19646889667838	-9.322832283942262e-09
1.8250	76.94933209806378	76.94933210817089	-1.0107115144819545e-08
1.8500	80.89460872013427	80.89460873108156	-1.0947289297291718e-08
1.8750	85.04216400012503	85.04216401197208	-1.1847049563584733e-08
1.9000	89.40236898660106	89.40236899941137	-1.281030392874527e-08
1.9250	93.98612646315793	93.98612647699912	-1.3841187751495454e-08
1.9500	98.80489821105967	98.80489822600377	-1.4944106396796997e-08
1.9750	103.87073366966213	103.87073368578586	-1.61237352358512e-08
2.0000	109.19630006628768	109.19630008367272	-1.7385048067808384e-08

Рис. 15. Результат исследования при h = 0.025

T	Y' real	Y' RKF45	Y' delta
1.0000	14.778112197861299	14.778112197861299	0.0
1.0125	15.152221889273688	15.152221889274673	-9.841016890277388e-13
1.0250	15.53580221261354	15.535802212615557	-2.0179413695586845e-12
1.0375	15.929092918069538	15.929092918072643	-3.105071755271638e-12
1.0500	16.332339825135293	16.332339825139538	-4.245492846166599e-12
1.0625	16.74579497625452	16.745794976259962	-5.4427573559223674e-12
1.0750	17.169716794355775	17.169716794362472	-6.696865284538944e-12
1.0875	17.6043702443752	17.604370244383208	-8.007816632016329e-12
1.1000	18.050026998868226	18.05002699887761	-9.382716825712123e-12
1.1125	18.50696560781377	18.50696560782459	-1.0821565865626326e-11
1.1250	18.97547167271703	18.97547167272936	-1.2327916465437738e-11
1.1375	19.455838025119746	19.45583802513365	-1.3905321338825l6e-11
1.1500	19.94836490962942	19.94836490964497	-1.5550227772109793e-11
1.1625	20.45336017158197	20.453360171599247	-1.7276846620006836e-11
1.1750	20.97113944945512	20.971139449474194	-1.9074519741479896-11
1.1875	21.502026372152677	21.50202637217363	-2.0953905277565354e-11
1.2000	22.04635276128317	22.046352761306085	-2.291500322826323e-11
1.2125	22.604458838559125	22.60445883858409	-2.496491902093112e-11
1.2250	23.176693438446737	23.17669343847384	-2.710365265556902e-11
1.2375	23.763414226198755	23.763414226228086	-2.9331204132176936e-11
1.2500	24.3649879214069	24.36498792143856	-3.1658231591791264e-11
1.2625	24.981790527213565	24.98179052724765	-3.4084735034412006e-11
1.2750	25.614207565326012	25.61420756536262	-3.660716174636036e-11
1.2875	26.262634316979973	26.262634317019216	-3.9243275296030333e-11
1.3000	26.92747607000332	26.927476070045305	-4.198597025606432e-11
1.3125	27.609148372134126	27.609148372178968	-4.484235205381992e-11
1.3250	28.308077290751537	28.308077290799353	-4.781597340297594e-11
1.3375	29.024699679181754	29.02469967923267	-5.091393973088998e-11
1.3500	29.75946344974559	29.759463449799725	-5.413625103756203e-11
1.3625	30.512827853718232	30.51282785377572	-5.7486460036670906e-11
1.3750	31.285263768376254	31.28526376843723	-6.0975224869253e-11
1.3875	32.077253991311245	32.07725399137585	-6.460254553530831e-11
1.4000	32.889293542194004	32.88929354226238	-6.837552746219444e-11
1.4125	33.72188997217796	33.72188997225026	-7.22977233635902e-11
1.4250	34.57556368113517	34.57556368121155	-7.637623866685317e-11
1.4375	35.450848242923165	35.45084824300378	-8.061817879934097e-11
1.4500	36.348290738886	36.348290738971016	-8.501643833369599e-11
1.4625	37.26845209979788	37.26845209988747	-8.959233355199103e-11
1.4750	38.21190745646316	38.211907456557505	-9.43458644542261e-11
1.4875	39.17924649919184	39.17924649929112	-9.92770310404012e-11
1.5000	40.17107384637519	40.17107384647959	-1.0440004416523152e-10
1.5125	41.1880094223919	41.18800942250162	-1.0972200925607467e-10
1.5250	42.23068884508106	42.2306888451963	-1.1523582088557305e-10
1.5375	43.29976382302394	43.2997638231449	-1.2096279533579946e-10
1.5500	44.39590256288309	44.39590256301	-1.269100380341115e-10
1.5625	45.51979018705327	45.519790187186345	-1.3307754898050916e-10
1.5750	46.67212916188523	46.67212916202471	-1.3947953902970767e-10
1.5875	47.85363973675006	47.85363973689618	-1.46116008181707e-10
1.6000	49.06506039421848	49.065060394371486	-1.5300827271857997e-10
1.6125	50.307148311636496	50.307148311796645	-1.6014922721296898e-10
1.6250	51.58067983438588	51.58067983455344	-1.6756018794694683e-10
1.6375	52.886450961125334	52.88645096130057	-1.752340494931559e-10
1.6500	54.22527784131552	54.2252778414987	-1.8318502270631143e-10
1.6625	55.597997285339055	55.59799728553049	-1.914344238684862e-10
1.6750	57.00546728753427	57.00546728773426	-1.999893584070378e-10
1.6875	58.448567562469584	58.448567562678434	-2.0884982632196625e-10
1.7000	59.928200094793716	59.928200095011746	-2.1803003846798674e-10
1.7125	61.4452897030055	61.445289703233044	-2.2754420569981448e-10
1.7250	63.000784617495526	63.00078461773292	-2.3739232801744947e-10
1.7375	64.595657073221	64.5956570734686	-2.475957217029645e-10
1.7500	66.23090391738427	66.23090391764244	-2.581685976110748e-10
1.7625	67.90754723249471	67.90754723276383	-2.6912516659649555e-10
1.7750	69.62663497520364	69.62663497548411	-2.8046542865922675e-10
1.7875	71.38924163131134	71.38924163160354	-2.922035946539836e-10
1.8000	73.19646888735555	73.1964688876599	-3.043538754354813e-10
1.8125	75.04944631920155	75.0494463195185	-3.169446927131503e-10
1.8250	76.94933209806378	76.94933209839375	-3.2997604648699053e-10
1.8375	78.89731371440057	78.89731371474402	-3.43447936757002e-10
1.8500	80.89460872013427	80.89460872049168	-3.574029960873304e-10
1.8625	82.94246548966098	82.94246549003282	-3.7184122447797563e-10
1.8750	85.04216400012503	85.04216400051179	-3.8676262192893773e-10
1.8875	87.19501663144642	87.19501663184865	-4.022240318590775e-10
1.9000	89.40236898660106	89.40236898701927	-4.1821124341367977e-10
1.9125	91.6656007326663	91.66560073310107	-4.347668891568901e-10
1.9250	93.98612646315793	93.98612646360979	-4.518625473792781e-10
1.9375	96.36539658219696	96.36539658266653	-4.695692723544198e-10
1.9500	98.80489821105967	98.80489821154752	-4.878586423728848e-10
1.9625	101.30615611767655	101.30615611818334	-5.067875008535339e-10
1.9750	103.87073366966213	103.87073367018849	-5.26355847796367e-10
1.9875	106.50023381147047	106.50023381201706	-5.465921049108147e-10
2.0000	109.19630006628768	109.1963000668552	-5.675246939063072e-10

Рис. 16. Результат исследования при h = 0.0125

Из приведенных значений (рис. 13-16) видно, что не смотря на странности с погрешностью, значения все еще соответствуют ожиданиям, а именно:

1. При уменьшении шага точность увеличивается.

2. Чем дальше от начальных значений, тем меньше точность вычисленных результатов.
3. Результаты очень близки к реальным значениям.

Вероятнее всего ошибка кроется в реализации программы RKF45, однако для подтверждения необходимо читать исходный код используемой библиотеки.

Вывод:

В ходе выполненной работы мы привели линейное дифференциальное уравнение к системе двух дифференциальных уравнений первого порядка, после чего решили эту систему при заданных НУ, используя методы Рунге-Кутты третьей степени точности, а также программу RKF45. Была найдена зависимость шага интегрирования h и величины глобальной и локальной погрешности а также наглядно продемонстрирована разница между используемыми методами.

Листинг кода:

```
import numpy as np
from scipy.integrate import ode
import matplotlib as mpl
import matplotlib.pyplot as plt
from prettytable import PrettyTable


def rkf45(f, T, X0):
    """
    Решает `x' = f(t, x)` для каждого `t` в `T`
    С начальным значением `X0`, используя аналог rkf45
    """
    runge = ode(f).set_integrator('dopri5', atol=0.0001).set_initial_value(X0, T[0])
    X = [X0, *[runge.integrate(T[i]) for i in range(1, len(T))]]
    return np.array([i[0] for i in X]), np.array([i[1] for i in X])


def RK3(f, T, X0):
    """
    Решает `x' = f(t, x)` для каждого `t` в `T`
    С начальным значением `X0`, используя формулы Рунге-Кутты 3 степени
    """
    X = np.zeros((len(T), len(X0)))
    X[0] = X0
    h = T[1] - T[0]
    for i in range(0, len(T) - 1):
        k_1 = h * f(T[i], X[i])
        k_2 = h * f(T[i] + h / 2, X[i] + k_1 / 2)
        k_3 = h * f(T[i] + 3 * h / 4, X[i] + 3 * k_2 / 4)
        X[i + 1] = (X[i] + (2 * k_1 + 3 * k_2 + 4 * k_3) / 9)
    return X[:, 0]


def f(t, X):
    """
    Правая часть `x' = f(t, x)`.
    """
    dX = np.zeros(X.shape)
    dX[0] = X[1]
    dX[1] = (t + 1) / t * X[1] + 2 * (t - 1) / t * X[0]
    return dX


def g(T):
    """
    Точное решение
    """
    return np.e ** (2 * T)


def print_one_graph(t, y, title, id, count_graphs):
    """
    Функция для отрисовки одного графика
    """
    plt.subplot(1, count_graphs, id)
    plt.xlabel('t')
    plt.ylabel('y')
    plt.grid()
    plt.title(title)
    plt.plot(t, y, '-o')
```

```python
def print_graph(t_find, y_real, Y_RKF45, Y_Runge_Kutta, h):
    """
    Функция для отрисовки всех графиков
    """
    mpl.use('TkAgg')
    plt.figure(figsize=(15, 4))
    print_one_graph(t_find, y_real, 'Исходный график', 1, 3)
    print_one_graph(t_find, Y_RKF45, 'График RKF45', 2, 3)
    print_one_graph(t_find, Y_Runge_Kutta, 'График Рунге-Кутты', 3, 3)
    plt.savefig(f"Graphs_{h}.jpg")
    plt.show()


def print_error_graph(t_find, Y_RKF45_error, Y_Runge_Kutta_error, h):
    """
    Функция для отрисовки погрешности
    """
    mpl.use('TkAgg')
    plt.figure(figsize=(15, 4))
    # Собственно сам график
    print_one_graph(t_find, Y_RKF45_error, 'Погрешность RKF45', 1, 2)
    print_one_graph(t_find, Y_Runge_Kutta_error, 'Погрешность Рунге-Кутты', 2, 2)
    plt.savefig(f"Error_{h}.jpg")
    plt.show()


def print_table(t_find, y_real, Y_RKF45, Y_RKF45_error, Y_Runge_Kutta,
Y_Runge_Kutta_error, h):
    """
    Функция для отрисовки таблицы
    """
    print(f'h = {h}')
    koef = {0.1: 1, 0.05: 2, 0.025: 4, 0.0125: 8}.get(h)
    pt = PrettyTable()
    pt.add_column('t', [f'{i:.1f}' for i in t_find[::koef]])
    pt.add_column('real y', [f'{i:.15f}' for i in y_real[::koef]])
    pt.add_column('RKF45 y', [f'{i:.15f}' for i in Y_RKF45[::koef]])
    pt.add_column('Delta RKF45 y', [f'{i:.15f}' for i in Y_RKF45_error[::koef]])
    pt.add_column('Runge Kutta y', [f'{i:.15f}' for i in Y_Runge_Kutta[::koef]])
    pt.add_column('Delta Runge Kutta y', [f'{i:.15f}' for i in
Y_Runge_Kutta_error[::koef]])
    print(pt)
    print('First step of RKF45:', Y_RKF45_error[1])
    print('First step of Runge Kutta:', Y_Runge_Kutta_error[1])
    print('Global of RKF45:', Y_RKF45_error.sum())
    print('Global of Runge Kutta:', Y_Runge_Kutta_error.sum())
    print('h^4 is about:', h ** 4)
    print('h^4 / Runge Kutta first step:', h ** 4 / Y_Runge_Kutta_error[1])


def print_additiontal_research(T, Y_derivative_real, Y_derivative_RKF45):
    pt = PrettyTable()
    pt.add_column("T", [f'{i:.4f}' for i in T])
    pt.add_column("Y' real", Y_derivative_real)
    pt.add_column("Y' RKF45", Y_derivative_RKF45)
    pt.add_column("Y' delta", Y_derivative_real - Y_derivative_RKF45)
    print(pt)
    print('=' * 110)
```

```python
def evaluate(h):
    """
    Получение решения при разных шагах
    """
    # Начальные значения
    X0 = np.array([np.e ** 2, 2 * np.e ** 2])
    # Значения в узлах
    T = np.arange(1, 2 + h, h)
    Y = g(T)
    # Расчет RKF45
    Y_RKF45, Y_derivative_RKF45 = rkf45(f, T, X0)
    # Расчет Рунге-Кутты
    Y_Runge_Kutta = RK3(f, T, X0)
    # Погрешности
    Y_RKF45_error = Y - Y_RKF45
    Y_Runge_Kutta_error = Y - Y_Runge_Kutta
    # Рисуем графики
    print_graph(T, Y, Y_RKF45, Y_Runge_Kutta, h)
    print_error_graph(T, Y_RKF45_error, Y_Runge_Kutta_error, h)
    # Выводим данные в консоль
    print_table(T, Y, Y_RKF45, Y_RKF45_error, Y_Runge_Kutta, Y_Runge_Kutta_error, h)
    # Вывод для дополнительных исследований
    print_additiontal_research(T, 2 * (np.e ** (2 * T)), Y_derivative_RKF45)
    return Y_RKF45_error, Y_Runge_Kutta_error


def print_table_error(Y_RKF45_error, Y_Runge_Kutta_error, h_list):
    pt = PrettyTable()
    pt.add_column('h', [f'{i:.4f}' for i in h_list])
    pt.add_column("Runge Kutta Error local", [f'{i[1]:.15f}' for i in
Y_Runge_Kutta_error])
    pt.add_column('h**4 / Runge Kutta Error local',
                  [f'{i ** 4 / j[1]:.15f}' for i, j in zip(h_list, Y_Runge_Kutta_error)])
    print(pt)
    pt.clear()
    pt.add_column('h', [f'{i:.4f}' for i in h_list])
    pt.add_column("Runge Kutta Error global", [f'{i.sum():.15f}' for i in
Y_Runge_Kutta_error])
    pt.add_column('h**2 / Runge Kutta Error global',
                  [f'{i ** 2 / j.sum():.15f}' for i, j in zip(h_list,
Y_Runge_Kutta_error)])
    print(pt)
    pt.clear()
    pt.add_column('h', [f'{i:.4f}' for i in h_list])
    pt.add_column("RKF45 Error local", [f'{i[1]:.15f}' for i in Y_RKF45_error])
    pt.add_column("RKF45 Error global", [f'{i.sum():.15f}' for i in Y_RKF45_error])
    pt.add_column("Runge Kutta Error local", [f'{i[1]:.15f}' for i in
Y_Runge_Kutta_error])
    pt.add_column("Runge Kutta Error global", [f'{i.sum():.15f}' for i in
Y_Runge_Kutta_error])
    print(pt)


def main():
    h_list = [0.1 / (2 ** i) for i in range(4)]
    Y_RKF45_error = []
    Y_Runge_Kutta_error = []
    for h in h_list:
        error = evaluate(h)
        Y_RKF45_error.append(error[0])
        Y_Runge_Kutta_error.append(error[1])
    print_table_error(Y_RKF45_error, Y_Runge_Kutta_error, h_list)


if __name__ == '__main__':
    main()
```

Ссылки:

Листинг код: github.com

Документация по SciPy: docs.scipy.org