

Доклад

Программное решение обыкновенных дифференциальных уравнений

Демидова Екатерина Алексеевна

Содержание

1	Цель работы	5
2	Задачи	6
3	Общие сведения о дифференциальных уравнениях	7
4	Метод Эйлера	9
5	Методы Рунге-Кутты	11
6	Решение задачи Коши с помощью программных средств	13
6.1	Решение задачи Коши с помощью Octave	13
6.2	Решение задачи Коши с помощью Julia	16
7	Сравнительный анализ решений	20
8	Выводы	24
	Список литературы	25

Список иллюстраций

6.1	Графики решения методом ode23() и точного решения	15
6.2	Графики решения методом ode45() и точного решения	16
6.3	Графики решения методом BS3() и точного решения	18
6.4	Графики решения методом DP5() и точного решения	19
7.1	Абсолютные погрешности	20
7.2	Найденные точки решения	21
7.3	Найденные решения	21
7.4	Абсолютные погрешности	22
7.5	Найденные точки решения	22
7.6	Найденные решения	23

Список таблиц

5.1	Таблица Бутчера	11
5.2	Таблица Бутчера для метода Богацки-Шампина	12

1 Цель работы

Исследовать программные решения обыкновенных дифференциальных уравнений.

2 Задачи

- Описать метод Эйлера и методы Рунге-Кутты
- Исследовать возможности Octave и Julia для решения ОДУ
- Решить конкретную задачу Коши с помощью Octave и Julia
- Провести сравнительный анализ результатов

3 Общие сведения о дифференциальных уравнениях

Дифференциальным уравнением n -го порядка называется соотношение вида [1]:

$$F(x, y, y', y'', \dots, y^{(n)}) = 0$$

Решением дифференциального уравнения называется функция $y(x)$, которая обращает уравнение в тождество.

Системой дифференциальных уравнений n -го порядка называется система вида:

$$\begin{cases} x'_1 = f_1(t, x_1, x_2, \dots, x_n) \\ x'_2 = f_2(t, x_1, x_2, \dots, x_n) \\ \vdots \\ x'_n = f_n(t, x_1, x_2, \dots, x_n) \end{cases}$$

Произвольную систему дифференциальных уравнений любого порядка можно заменить некоторой эквивалентной системой уравнений первого порядка:

$$y'(x) = f_k(x, y(x)), \quad (1)$$

$$y = y_1, y_2, \dots, y_n, f = f_1, f_2, \dots, f_n.$$

Каждое дифференциальное уравнение, а также система, имеют бесконечное

количество решений, которые различаются друг от друга только константами. Чтобы однозначно определить решение, необходимо указать дополнительные начальные или граничные условия. Количество таких условий должно соответствовать порядку дифференциального уравнения или системы. В зависимости от формы дополнительных условий, дифференциальные уравнения классифицируются на:

- задачу Коши, в случае если все дополнительные условия заданы в одной (чаще начальной) точке интервала;
- краевую задачу, в случае, когда дополнительные условия заданы на границах интервала

Существуют два типа методов решения дифференциальных уравнений: точные (аналитические) и приближенные (численные). Классы уравнений, для которых разработаны методы получения точных решений, сравнительно узки и охватывают только малую часть возникающих на практике задач.

Численные методы - это алгоритмы вычисления приближенных значений искомого решения $y(x)$ уравнения (1) на некоторой равномерной выбранной сетке по x . Решение при этом получается в виде таблицы. Численные методы не позволяют найти общее решение системы (1). Они могут дать только частное решение, например решение задачи Коши.

4 Метод Эйлера

Метод Эйлера играет важную роль в теории численных методов решения ОДУ, хотя и не часто используется в практических расчетах из-за невысокой точности.

Рассмотрим задачу Коши для дифференциального уравнения

$$y'(x) = f(x, y) \quad (2)$$

удовлетворяющее начальному условию

$$y(x_0) = y_0 \quad (3)$$

Требуется найти численное решение уравнения (2) на интервале $(x_0; x_n)$, которое удовлетворяет начальному условию (3).

Проведём разбиение отрезка $[x_0; x_n]$ на n равных частей:

$$x_i = x_0 + ih, i = \overline{1, n}$$

$$h = \frac{x_n - x_0}{n}$$

Для вычисления значения функции в точке x_1 разложим функцию $y = y(x)$ в окрестности точки x_0 в ряд Тейлора:

$$y(x_1) = y(x_0 + h) = y(x_0) + y'(x_0)h + y''(x_0)\frac{h^2}{2} + \dots$$

При достаточно малом значении h членами выше второго порядка можно пренебречь и с учетом $y'(x_0) = f(x_0, y_0)$ получим следующую формулу для

вычисления приближенного значения функции $y(x)$ в точке x_1 :

$$y_2 = y_1 + hf(x_1, y_1)$$

Повторяя этот процесс, сформируем последовательность значений x_i в точках t_i по формуле:

$$y_{i+1} = y_i + hf(x_i, y_i) \quad (4)$$

Процесс нахождения значений функции x_i в узловых точках t_i по формуле (3) называется методом Эйлера[2]. Так как точность определяется отброшенными членами ряда, то точность метода Эйлера на каждом шаге составляет $O(h^2)$. в целом точность этого метода $O(h)$.

5 Методы Рунге-Кутты

Семейство явных методов Рунге — Кутты задаётся формулами:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^n b_i \mathbf{k}_i,$$

$$\left\{ \begin{array}{l} \mathbf{k}_1 = \mathbf{f}(x_n, \mathbf{y}_n), \\ \mathbf{k}_2 = \mathbf{f}(x_n + c_2 h, \mathbf{y}_n + a_{21} h \mathbf{k}_1), \\ \dots \\ \mathbf{k}_s = \mathbf{f}(x_n + c_s h, \mathbf{y}_n + a_{s1} h \mathbf{k}_1 + a_{s2} h \mathbf{k}_2 + \dots + a_{s,s-1} h \mathbf{k}_{s-1}). \end{array} \right.$$

Конкретный метод определяется числом s и коэффициентами b_i , a_{ij} и c_i . Эти коэффициенты часто упорядочивают в таблицу Бутчера.

Таблица 5.1: Таблица Бутчера

c_1	a_{11}	a_{12}	\dots	a_{1s}
c_2	a_{21}	a_{22}	\dots	a_{2s}
\vdots	\vdots	\vdots	\ddots	\vdots
c_s	a_{s1}	a_{s2}	\dots	a_{ss}
	b_1	b_2	\dots	b_s

Для коэффициентов метода Рунге — Кутты должны быть выполнены условия $\sum_{j=1}^{i-1} a_{ij} = c_i$ для $i = 2, \dots, s$.

Методы Рунге-Кутты имеют ряд достоинств.

- Позволяют получить приближенное решение с хорошей точностью.
- Являются явными, то есть значение y_{n+1} вычисляется по ранее найденным значениям за определённое число действий по определённым формулам.
- Допускают расчет переменным шагом, то есть можно уменьшить шаг там, где функция быстро меняется, и увеличить его в обратном случае.

В работе будут использоваться методы Богацки-Шампина и Дорманда Принса из семейства методов Рунге-Кутты.

Таблица 5.2: Таблица Бутчера для метода Богацки-Шампина

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{3}{4}$	0	$\frac{3}{4}$		
1	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{4}{9}$	
	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{4}{9}$	0
	$\frac{7}{24}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{8}$

6 Решение задачи Коши с помощью программных средств

Рассмотрим задачу Коши:

$$\begin{cases} y'(x) = 11y + e^{10x} + x^4 + x^3 - \sin(x) \\ y(0) = 5 \end{cases} \quad (5)$$

Известно точное решение задачи (5):

$$y(x) = -\frac{x^4}{11} - \frac{15x^3}{121} - \frac{45x^2}{1331} - \frac{90x}{14641} + \frac{\cos(x)}{122} + \frac{11\sin(x)}{122} + \frac{117739261e^{11x}}{19648222} - e^{10x} - \frac{90}{161051}$$

6.1 Решение задачи Коши с помощью Octave

В Octave нет метода Эйлера, однако есть методы Рунге-Кутты[3].

`ode23(@f, interval, X0, options)`, `ode45(@f, interval, X0, options)` — функции решений обыкновенных нежёстких дифференциальных уравнений (или систем) методом Рунге-Кутта 2-3-го и 4-5-го порядка точности соответственно.

Функции решают систему дифференциальных уравнений, автоматически подбирая шаг для достижения необходимой точности. Входными параметрами этих функций являются:

- f – вектор-функция для вычисления правой части дифференциального уравнения или системы;
- `interval` – массив из двух чисел, определяющий интервал интегрирования дифференциального уравнения или системы;
- X_0 – вектор начальных условий системы дифференциальных систем;
- `option` – параметры управления ходом решения дифференциального уравнения или системы.

При решении дифференциальных уравнений необходимо определить следующие параметры: - `RelTol` – относительная точность решения, значение по умолчанию 10^{-3} ; - `AbsTol` – абсолютная точность решения, значение по умолчанию 10^{-3} ; - `InitialStep` – начальное значение шага изменения независимой переменной, значение по умолчанию 0.025; - `MaxStep` – максимальное значение шага изменения независимой переменной, значение по умолчанию 0.025.

Все функции возвращают: - массив T - координат узлов сетки, в которых ищется решение; - матрицу X , i -й столбец которой является значением вектор-функции решения в узле T_i .

```
function dydx = f(x,y)
    dydx = y*11 +exp(10*x)+(x^4+x^3)-sin(x);
endfunction
```

```
[X23,Y23]=ode23(@f,[0 0.5],5);
```

При решении методом `ode23()` найдено 22 точки, приведём графики реального решения задачи (5) и численного(рис. 6.1).

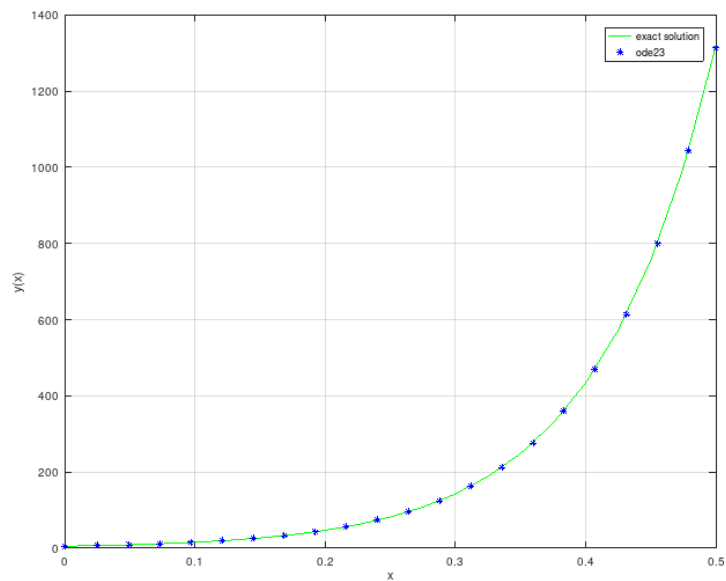


Рис. 6.1: Графики решения методом ode23() и точного решения

```
function dydx = f(x,y)
    dydx = y*11 + exp(10*x) + (x^4 + x^3) - sin(x);
endfunction
```

```
[X45,Y45]=ode45(@f,[0 0.5],5);
```

При решении методом ode45() найдена 21 точка, приведём графики реального решения задачи (5) и численного(рис. 6.2).

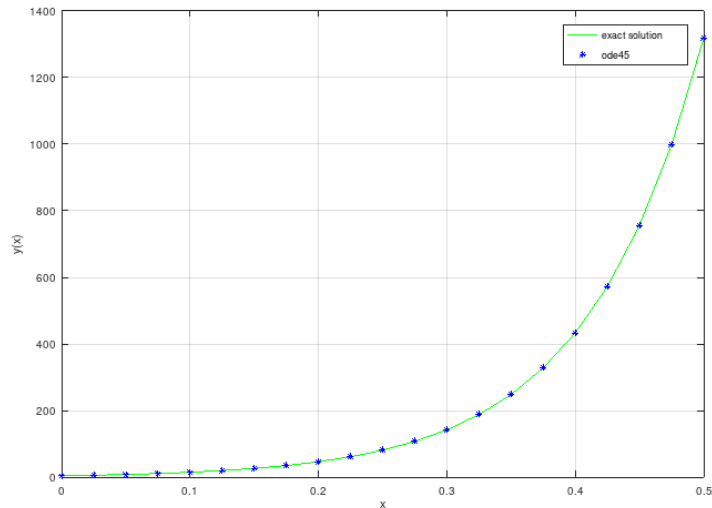


Рис. 6.2: Графики решения методом ode45() и точного решения

6.2 Решение задачи Коши с помощью Julia

Для того чтобы задать ОДУ используется конструктор `ODEProblem[4]`:

```
ODEProblem(f::ODEFunction,u0,tspan,p=NullParameters();kwargs...)
```

Входными параметрами этого конструктора являются:

- `f` – функция в ОДУ;
- `u0` – начальное состояние;
- `tspan` - временной интервал;
- `p` – параметры ОДУ;
- `kwargs` – параметры управления ходом решения.

Функция in-place — `f!(du,u,p,t)`, где возвращаемое значение игнорируется, и ожидается, что результат будет преобразован в значение `du`. Функция out-place - `du=f(u,p,t)`

Для решения ОДУ используется функция `solve`, на вход которой подаётся `ODEProblem` с разными параметрами. Наиболее полезными параметрами являются `abstol` – абсолютная точность, `reltol` – относительная точность, `saveat` -

шаг метода. По умолчанию $\text{abstol}=1\text{e-}6$, $\text{reltol}=1\text{e-}3$. dt - начальное значение шага изменения независимой переменной, dtmax – максимальное значение шага изменения независимой переменной. Также на вход подаётся метод решения, мы рассмотрим метод $\text{BS3}()$ - метод Рунге-Куты 2-го, 3-го порядка, эта функция соответствует $\text{ode23}()$ из Matlab и $\text{DP5}()$ - метод Рунге-Куты 4-го, 5-го порядка, эта функция соответствует $\text{ode45}()$ из Matlab.

При решении задачи (5) было найдено решение при следующих параметрах: абсолютная точность – $1\text{e-}3$, относительная точность - $1\text{e-}3$, начальное значение шага - 0.025, максимальное значение шага – 0.025.

```
function F!( y, p, x)
    dydx = x*11+exp(10*x)+(x.^4+x.^3)-sin(x);
end

begin
    x0 = 5.0
    X = (0.0, 0.5)
    prob = ODEProblem(F!, y0, x)
end

sol = solve(prob, BS3(), dtmax=0.025, dt=0.025, abstol = 1e-3, reltol = 1e-3);
```

При решении методом $\text{BS3}()$ найдено 24 точки, приведём графики реального решения задачи (5) и численного(рис. 6.3).

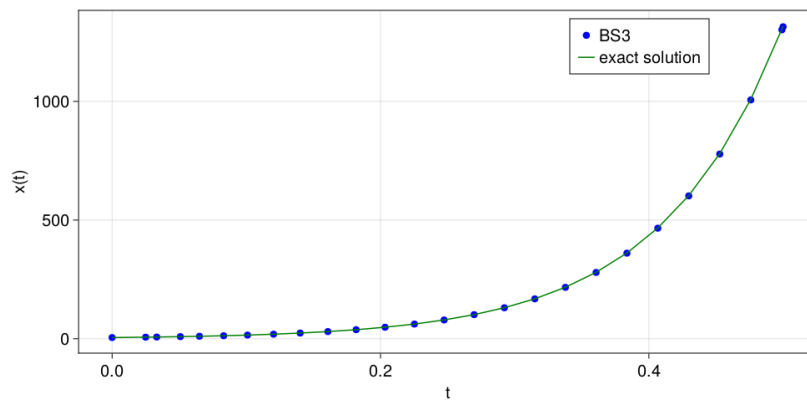


Рис. 6.3: Графики решения методом BS3() и точного решения

```
function F!( y, p, x)
    dydx = x*11+exp(10*x)+(x.^4+x.^3)-sin(x);
end

begin
    x0 = 5.0
    X = (0.0, 0.5)
    prob = ODEProblem(F!, y0, x)
end

sol = solve(prob, DP5(), dtmax=0.025, dt=0.025, abstol = 1e-3, reltol = 1e-3);
```

При решении методом DP5() найдена 21 точка, приведём графики реального решения задачи (5) и численного(рис. 6.4).

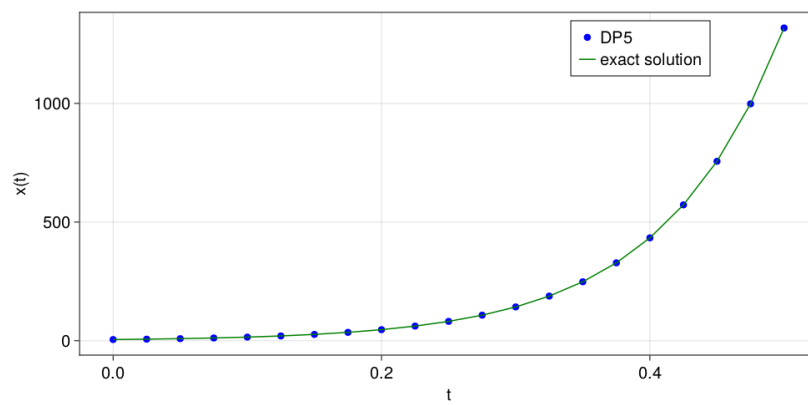


Рис. 6.4: Графики решения методом DP5() и точного решения

7 Сравнительный анализ решений

Абсолютная погрешность решения в Julia оказалась меньше, чем в Octave. Система Julia при решении нашла 26 точек и минимальный подобранный шаг оказался приблизительно равен 0.000889. Система Octave нашла решение в 22 точках, а минимальный шаг приблизительно равен 0.020739 (рис. 7.1).

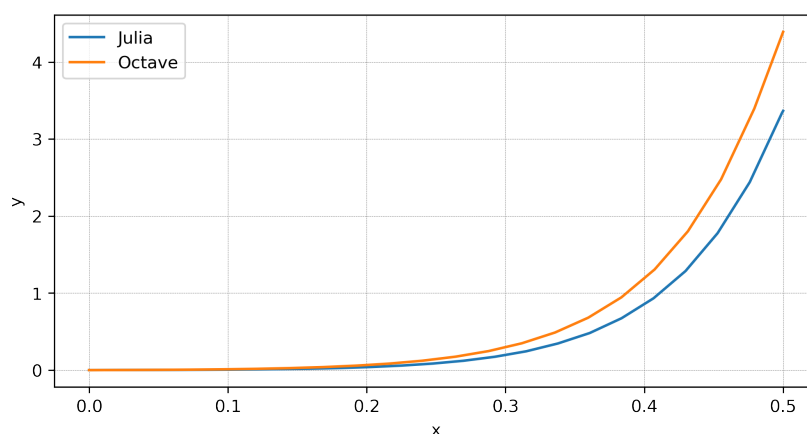


Рис. 7.1: Абсолютные погрешности

При построении решений, найденных обеими рассматриваемыми системами, можно заметить, что такая погрешность практически не влияет на построение графика по этим точкам. (рис. 7.2, 7.3).

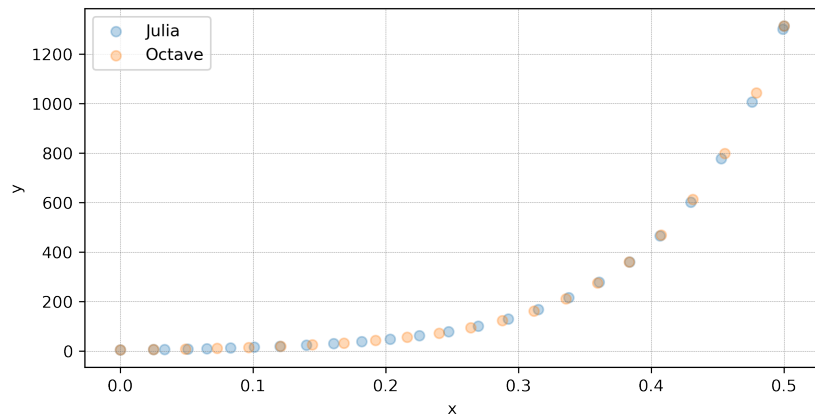


Рис. 7.2: Найденные точки решения

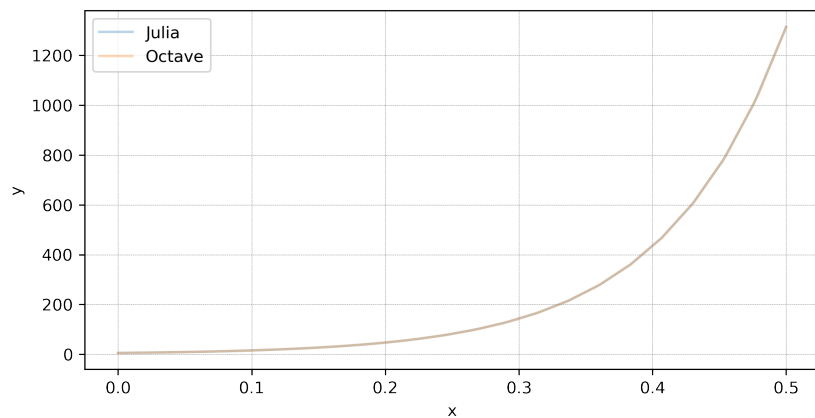


Рис. 7.3: Найденные решения

Абсолютная погрешность решения в Julia оказалась меньше, чем в Octave, однако незначительно, на последнем шаге разница в абсолютной погрешности приблизительно 0.0075. В обеих системах шаг был равен 0.025 и не менялся. (рис. 7.4).

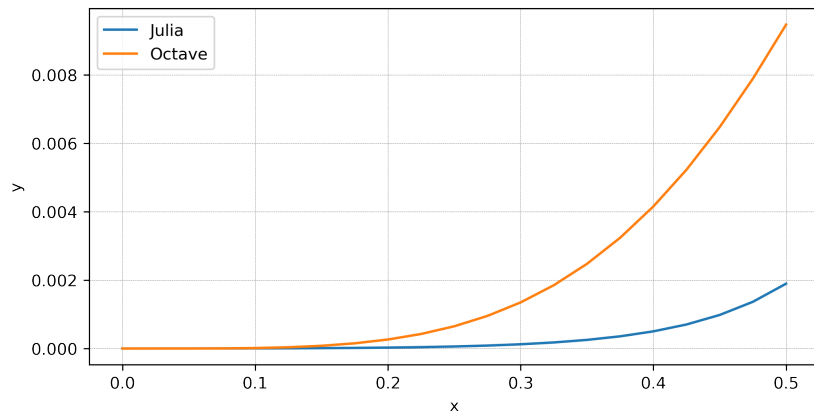


Рис. 7.4: Абсолютные погрешности

При построении решений, найденных обеими рассматриваемыми системами, можно заметить, что такая погрешность не влияет на построение графика по этим точкам. (рис. 7.5, 7.6).

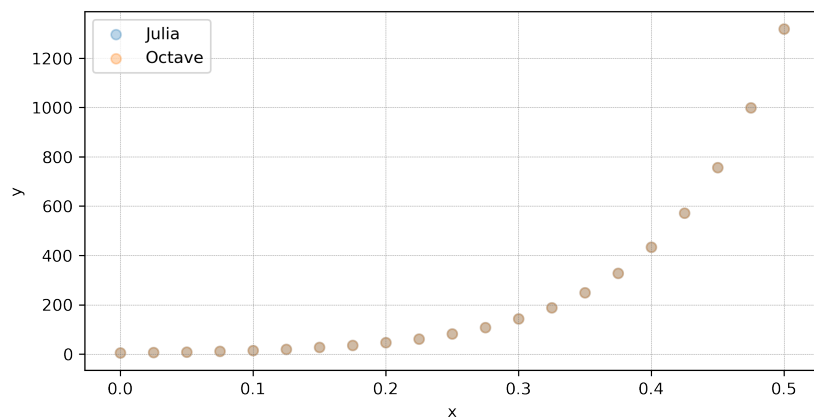


Рис. 7.5: Найденные точки решения

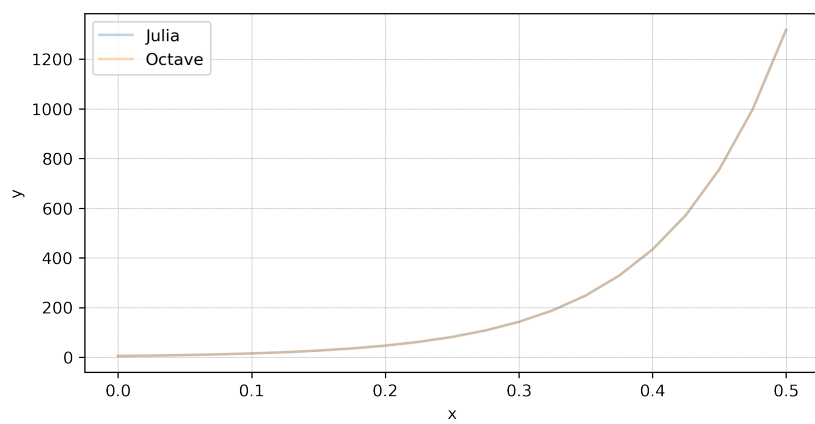


Рис. 7.6: Найденные решения

8 Выводы

В работе были исследованы программные решения обыкновенных дифференциальных уравнений с помощью Octave и Julia. С помощью методов Богацки-Шампина и Дорманда-Принса были получены решения конкретной Задачи Коши, при сравнении Julia показала большую точность, чем Octave, но не значительно. Все графики решений совпали с точным решением рассматриваемой задачи.

Список литературы

1. Понтрягин Л.С. Дифференциальные уравнения и их приложения. Наука, 1988. 207 с.
2. Калиткин Н.Н. Численные методы. Наука, 1978. 512 с.
3. GNU Octave Documentation [Электронный ресурс]. Free Software Foundation, 2023. URL: https://docs.octave.org/v4.2.0/Matlab_002dcompatible-solvers.html.
4. Julia Documentation [Электронный ресурс]. JuliaLang.org contributors, 2023. URL: <https://docs.sciml.ai/DiffEqDocs/stable/>.