Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра высшей математики

Отчет по лабораторной работе №6

«Численное решение задачи Коши для ОДУ первого порядка и их систем»

Вариант 1

Проверила:

Князева Л. П.

Выполнил:

ст. гр. 321901

Афанасьев А.А.

Минск 2024

Задание 1

Решить задачу Коши для дифференциального уравнения первого порядка на отрезке [0;1]:

а) Методом Эйлера-Коши с шагом $h_1 = 0{,}1$ и $h_2 = 0{,}25$, построить графики полученных решений:

Запишем исходное ОДУ с начальным условием:

$$1.1.\quad y' = \cos(2x + y) + x - y,\ \ y(0) = 0\,.$$

Задаемфункцию$f[x, y]$которая является правой частью дифференциального

уравнения:

```
f[x_, y_] := Cos[2 x + y] + x - y
```

Записываем границы отрезка, начальные условия, размер шага для двух случаев и приписываем переменным начальные значения:

```
a = 0; b = 1; x0 = 0; y0 = 0; h1 = 0.1; h2 = 0.05;
n1 = Floor[(b - a) / h1];
n2 = Floor[(b - a) / h2];
x = x0; y = y0;
```

Реализация метода Эйлера-Коши для двух шагов $h_1 = 0{,}1$ и $h_2 = 0{,}05$

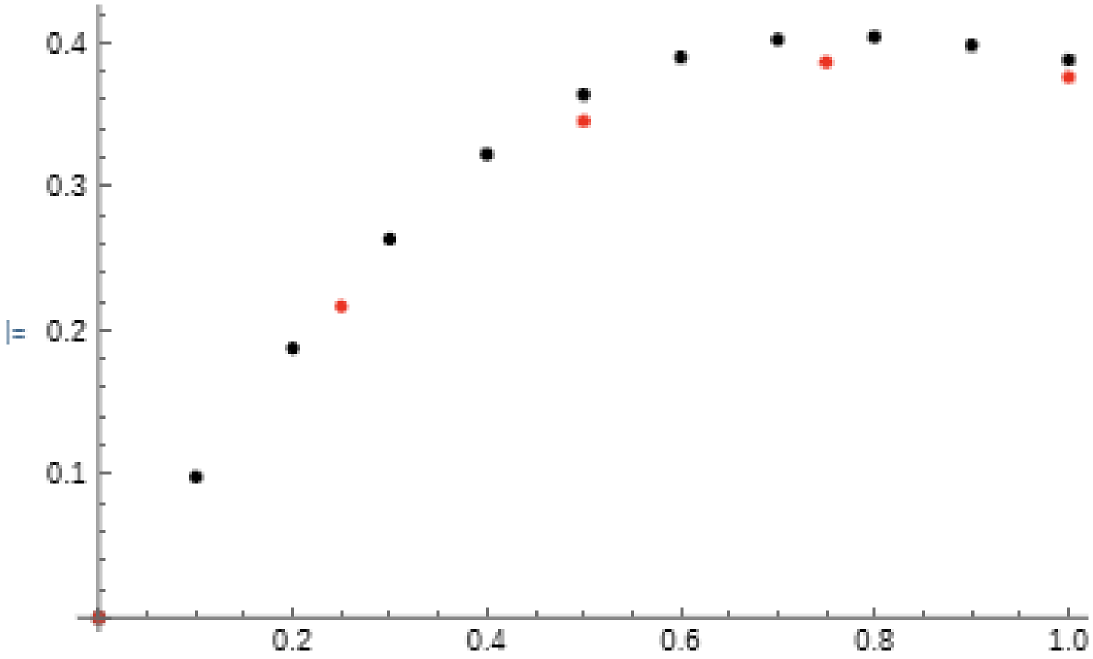
```
x = x0; y = y0;
ec1 = Table[{x, y} = {x + h1, y + h1 / 2 (f[x, y] + f[x + h1, y + h1 * f[x, y]])}, {i, n1}];
ec11 = Prepend[ec1, {x0, y0}];

x1 = x0; y1 = y0;
ec2 = Table[{x1, y1} = {x1 + h2, y1 + h2 / 2 (f[x1, y1] + f[x1 + h2, y1 + h2 * f[x1, y1]])}, {i, n2}];
ec22 = Prepend[ec2, {x0, y0}];
{MatrixForm[ec11], MatrixForm[ec22]}
```

Представим таблицы приближенных значений функции для шагов $h_1 = 0{,}1$ и $h_2 = 0{,}25$:

$$\left\{\begin{pmatrix} 0 & 0 \\ 0.1 & 0.0985514 \\ 0.2 & 0.18903 \\ 0.3 & 0.265538 \\ 0.4 & 0.325013 \\ 0.5 & 0.36697 \\ 0.6 & 0.392791 \\ 0.7 & 0.404951 \\ 0.8 & 0.406423 \\ 0.9 & 0.400298 \\ 1. & 0.389606 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0.25 & 0.229231 \\ 0.5 & 0.36681 \\ 0.75 & 0.40666 \\ 1. & 0.389488 \end{pmatrix}\right\}$$

Графики приближенных значений в виде точек:



б) Методом Рунге-Кутта четвертого порядка с шагом $h_1 = 0{,}1$ и $h_2 = 0{,}25$, построить графики полученных решений:

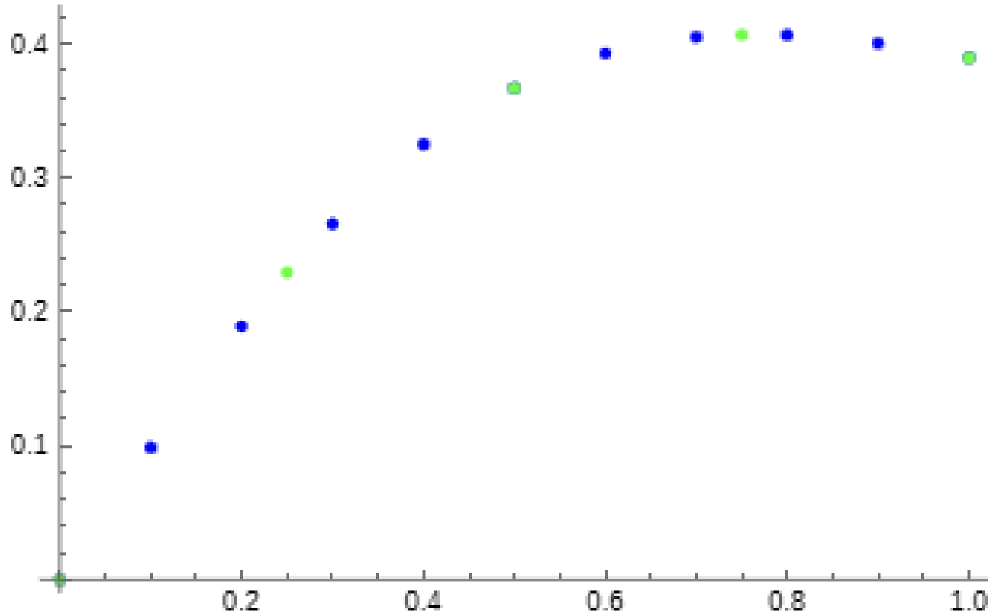
Реализуем метод Рунге-Кутта четвертого порядка для шага $h_1 = 0{,}1$ и $h_2 = 0{,}25$:

```
f[x_, y_] := Cos[2 x + y] + x - y
a = 0; b = 1; x0 = 0; y0 = 0; h1 = 0.1; h2 = 0.05;
n1 = Floor[(b - a) / h1];
n2 = Floor[(b - a) / h2];

sol1 = List[{x0, y0}];
x = x0; y = y0; x1 = x0; y1 = y0;

For[k = 1, k ≤ n1, k++,
  k1[x_, y_] := h1 * f[x, y];
  k2[x_, y_] := h1 * f[x + h1 / 2, y + k1[x, y] / 2];
  k3[x_, y_] := h1 * f[x + h1 / 2, y + k2[x, y] / 2];
  k4[x_, y_] := h1 * f[x + h1, y + k3[x, y]];
  xTemp = x; yTemp = y;
  y = yTemp + (k1[xTemp, yTemp] + 2 * k2[xTemp, yTemp] +
      2 * k3[xTemp, yTemp] + k4[xTemp, yTemp]) / 6;
  x = xTemp + h1;
  sol1 = Append[sol1, {x, y}]]
MatrixForm[sol1]
gr3 = ListPlot[sol1, PlotStyle → Blue]
```

$$\left\{\begin{pmatrix} 0 & 0 \\ 0.1 & 0.0985514 \\ 0.2 & 0.18903 \\ 0.3 & 0.265538 \\ 0.4 & 0.325013 \\ 0.5 & 0.36697 \\ 0.6 & 0.392791 \\ 0.7 & 0.404951 \\ 0.8 & 0.406423 \\ 0.9 & 0.400298 \\ 1. & 0.389606 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0.25 & 0.229231 \\ 0.5 & 0.36681 \\ 0.75 & 0.40666 \\ 1. & 0.389488 \end{pmatrix}\right\}$$



в) С помощью функций **DSolve** и **NDSolve**, построить графики:

Используем функцию **DSolve:**

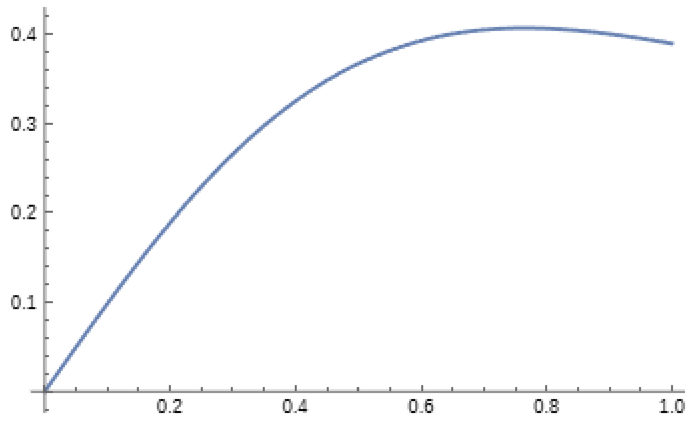
```
sol3 = DSolve[{y'[x] == f[x, y[x]], y[x0] == y0}, y[x], x];
y1[x_] = y[x] /. Flatten[sol3]
```

```
Solve: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.
ReplaceAll: {DSolve[{y'[x] == 1 + 4 Sin[x] y[x] - 1.5 y[«1»]^2, y[0] == 0}, y[x], x]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing.
```

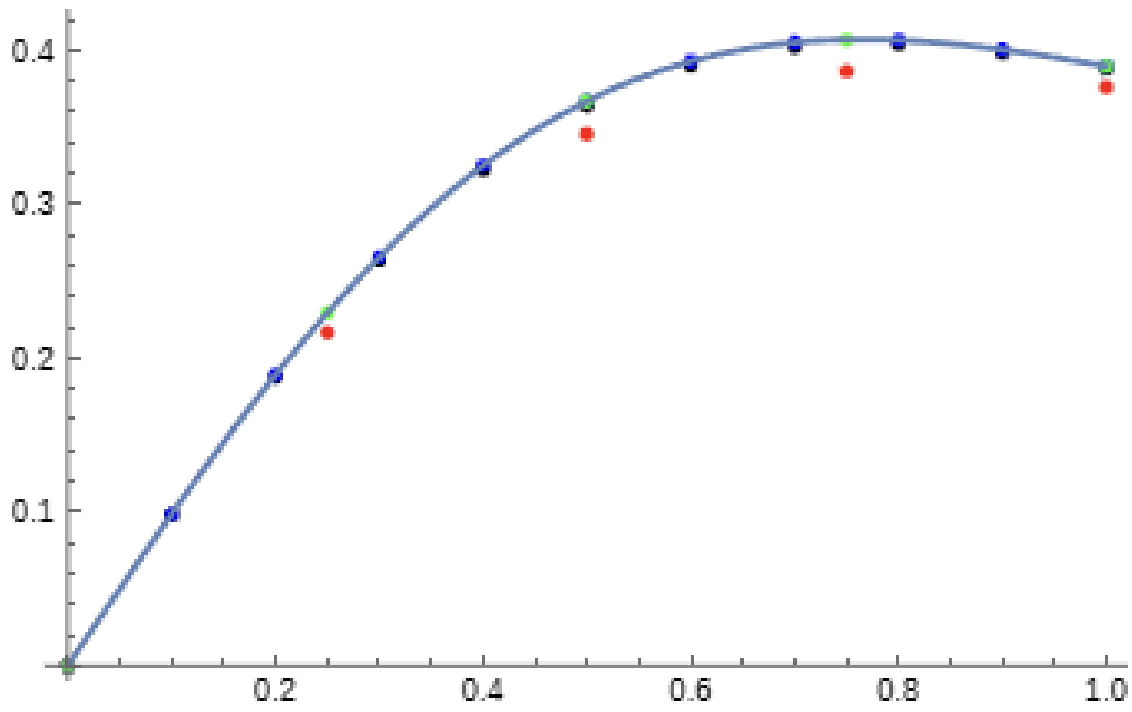
Среда *Mathematica Wolfram* не может представить аналитическое выражение для искомой функции.

Используем функцию **NDSolve:**

```
sol4 = NDSolve[{y'[x] == f[x, y[x]], y[x0] == y0}, y, {x, a, b}];
gr5 = Plot[Evaluate[y[x] /. sol4], {x, 0, 1}]
```



Изобразим на одной координатной прямой все полученные приближенные решения и график, полученный функцией **NDSolve:**



Так как все точки расположены очень близко к графику функции, нет причины выводить отдельно приближенные решения методов Эйлера-Коши и Рунге-Кутта четвертого порядка.

Сравнить все полученные решения. Сделать выводы о точности методов в зависимости от шага сетки:

По графику функции составим две таблицы ее значений для шагов $h_1 = 0{,}1$ и $h_2 = 0{,}25$ и найдем нормы вектор-столбцов абсолютных погрешностей (максимальная погрешность) обоих методов для двух значений шага:

```
tt2 = Table[{x0 + i*h2, y[x0 + i*h2] /. First[sol4]}, {i, 0, n2}];
MatrixForm[tt2]

tt3 = Table[{x0 + i*h1, y[x0 + i*h1] /. First[sol4]}, {i, 0, n1}];
MatrixForm[tt3]

deltah1EK = Norm[tt3 - ec11, ∞]

deltah1R = Norm[tt3 - sol1, ∞]

deltah2EK = Norm[tt2 - ec22, ∞]

deltah2R = Norm[tt2 - sol2, ∞]
```

Максимальная погрешность метода Эйлера-Коши для $h_1 = 0{,}1$:

0.00303988

Максимальная погрешность метода Рунге-Кутта для $h_1 = 0{,}1$:

4.15923×10^{-6}

Максимальная погрешность метода Эйлера-Коши для $h_2 = 0{,}25$:

0.0216299

Максимальная погрешность метода Рунге-Кутта для $h_2 = 0{,}25$:

0.000174113

Вывод: точность каждого из методов увеличивается с уменьшением шага *h*.

Метод Рунге-Кутта четвертого порядка точнее метода Эйлера-Коши.

Задание 2

Решить задачу Коши для системы двух дифференциальных уравнений на отрезке [0;1]

а) Методом Эйлера с шагом $h_1 = 0{,}1$ и $h_2 = 0{,}05$, построить графики полученных решений:

Исходная система и начальные условия:

$$\begin{cases} y' - y^2 + 3z = 0, y(0) = 1, \\ z' - 4\,y' - 7z = 0, z(0) = \ 0; \end{cases}$$

Выразим производные искомых функций и запишем границы отрезка, начальные условия, размер шага для двух случаев и приписываем переменным начальные значения:

```
f[x_, y_, z_] := 2*z;
g[x_, y_, z_] := 3*f[x, y, z] + 1;
a = 0; b = 1; y0 = 0; z0 = 0; x0 = 0; h = 0.1; n = Floor[(b - a) / h];
x = x0;
y = y0;
z = z0;
x = x0; y = y0; z = z0;
```

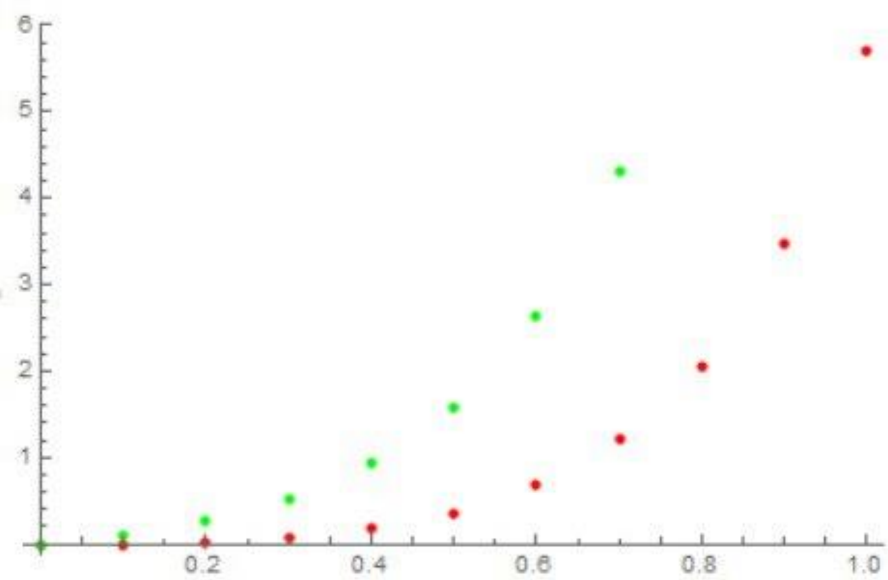
Создадим таблицы приближенных значений функций методом Эйлера для шага $h_1 = 0{,}1$:

```
eul1 = List[{x0, y0}];
eul2 = List[{x0, z0}];

eul = Table[{x, y, z} = {x + h, y + h*f[x, y, z], z + h*g[x, y, z]};
   eul1 = Append[eul1, {x, y}];
   eul2 = Append[eul2, {x, z}];
   {x, y, z}, {i, n}];
eul1
eul2
```

```
{{0, 0}, {0.1, 0.}, {0.2, 0.02}, {0.3, 0.072}, {0.4, 0.1752}, {0.5, 0.36032}, {0.6, 0.676512}, {0.7, 1.20242}, {0.8, 2.06387}, {0.9, 3.46219}, {1., 5.71951}}
{{0, 0}, {0.1, 0.1}, {0.2, 0.26}, {0.3, 0.516}, {0.4, 0.9256}, {0.5, 1.58096}, {0.6, 2.62954}, {0.7, 4.30726}, {0.8, 6.99161}, {0.9, 11.2866}, {1., 18.1585}}
```

Изобразим графически полученные решения:

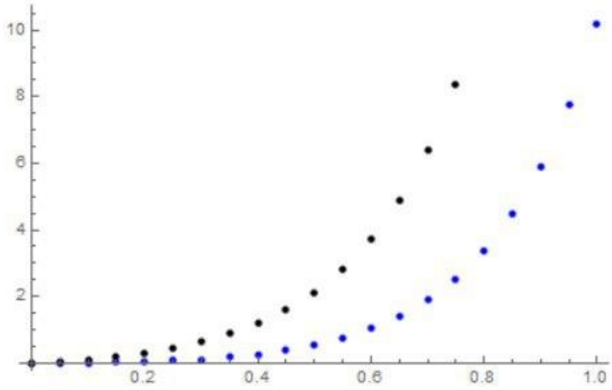


Создадим таблицы приближенных значений функций и выведем их графики методом Эйлера для шага $h_2 = 0{,}05$:

```
ClearAll[x, y, z, h, n]
h = 0.05; n = Floor[(b - a) / h];
y0 = 1; z0 = 0; x0 = 0;
eul3 = List[{x0, y0}];
eul4 = List[{x0, z0}];
x = x0; y = y0; z = z0;
eul = Table[{x, y, z} = {x + h, y + h * f[x, y, z], z + h * g[x, y, z]};
   eul3 = Append[eul3, {x, y}];
   eul4 = Append[eul4, {x, z}];
   {x, y, z}, {i, n}];
eul3
eul4
gr3 = ListPlot[eul3, PlotStyle → Blue];
gr4 = ListPlot[eul4, PlotStyle → Black];
Show[gr3, gr4]
```

```
{{0, 0}, {0.05, 0.}, {0.1, 0.005}, {0.15, 0.0165}, {0.2, 0.03645}, {0.25, 0.067385}, {0.3, 0.112601}, {0.35, 0.176381}, {0.4, 0.264295}, {0.45, 0.383583}, {0.5, 0.543658},
 {0.55, 0.756756}, {0.6, 1.03878}, {0.65, 1.41042}, {0.7, 1.89854}, {0.75, 2.53811}, {0.8, 3.37454}, {0.85, 4.4669}, {0.9, 5.89197}, {0.95, 7.74956}, {1., 10.1694}}

{{0, 0}, {0.05, 0.05}, {0.1, 0.115}, {0.15, 0.1995}, {0.2, 0.30935}, {0.25, 0.452155}, {0.3, 0.637802}, {0.35, 0.879142}, {0.4, 1.19288}, {0.45, 1.60075}, {0.5, 2.13097},
 {0.55, 2.82027}, {0.6, 3.71635}, {0.65, 4.88125}, {0.7, 6.39563}, {0.75, 8.36432}, {0.8, 10.9236}, {0.85, 14.2507}, {0.9, 18.5759}, {0.95, 24.1987}, {1., 31.5083}}
```



б) Методом Рунге-Кутта четвертого порядка с шагом $h_1 = 0{,}1$ и $h_2 = 0{,}05$, построить графики полученных решений:

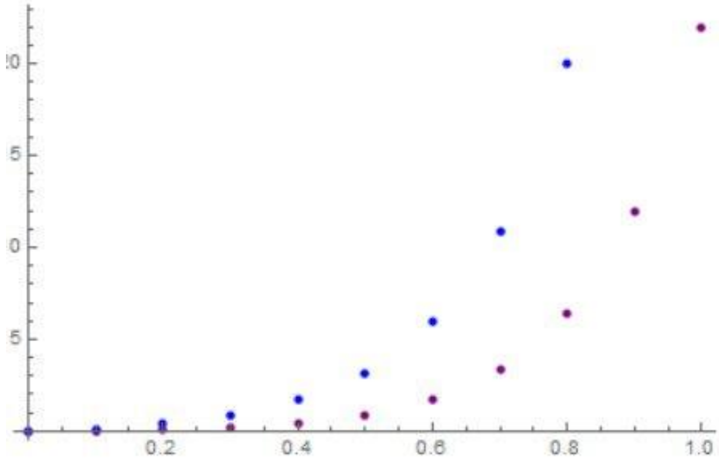
Запишем исходные данные, создадим таблицы и предадим переменным начальные значения:

```
ClearAll[x, y, z, x1, y1, x0, y0, z0]
f[x_, y_, z_] := 2 * z;
g[x_, y_, z_] := 3 * f[x, y, z] + 1;
a = 0; b = 1; y0 = 0; z0 = 0; x0 = 0; h = 0.1; n = Floor[(b - a) / h];
sol5 = List[{x0, y0}];
sol6 = List[{x0, z0}];
x = x0; y = y0; z = z0;
```

Для шага $h_1 = 0{,}1$ создадим таблицы приближенных решений и изобразим их графически:

```
For[k = 1, k < n + 1, k++,
 k1[x_, y_, z_] := h * f[x, y, z];
 k1z[x_, y_, z_] := h * g[x, y, z];
 k2z[x_, y_, z_] := h * g[x + h / 2, y + k1[x, y, z] / 2, z + k1z[x, y, z] / 2];
 k2[x_, y_, z_] := h * f[x + h / 2, y + k1[x, y, z] / 2, z + k1z[x, y, z] / 2];
 k3z[x_, y_, z_] := h * g[x + h / 2, y + k1[x, y, z] / 2, z + k2z[x, y, z] / 2];
 k3[x_, y_, z_] := h * f[x + h / 2, y + k2[x, y, z] / 2, z + k2z[x, y, z] / 2];
 k4z[x_, y_, z_] := h * g[x + h, y + k3[x, y, z], z + k3z[x, y, z]];
 k4[x_, y_, z_] := h * f[x + h, y + k3[x, y, z], z + k3z[x, y, z]];
 xTemp = x; yTemp = y; zTemp = z;
 y = yTemp + (k1[xTemp, yTemp, zTemp] + 2 * k2[xTemp, yTemp, zTemp] + 2 * k3[xTemp, yTemp, zTemp] + k4[xTemp, yTemp, zTemp]) / 6;
 z = zTemp + (k1z[xTemp, yTemp, zTemp] + 2 * k2z[xTemp, yTemp, zTemp] + 2 * k3z[xTemp, yTemp, zTemp] + k4z[xTemp, yTemp, zTemp]) / 6;
 x = xTemp + h;
 sol5 = Append[sol5, {x, y}];
 sol6 = Append[sol6, {x, z}]]
sol5
sol6
gr5 = ListPlot[sol5, PlotStyle → Purple];
gr6 = ListPlot[sol6, PlotStyle → Blue];
Show[gr5, gr6]
```

```
{{0, 0}, {0.1, 0.0123}, {0.2, 0.0620832}, {0.3, 0.180138}, {0.4, 0.422544}, {0.5, 0.891442}, {0.6, 1.77287}, {0.7, 3.40569}, {0.8, 6.40708}, {0.9, 11.9012}, {1., 21.9356}}
{{0, 0}, {0.1, 0.1369}, {0.2, 0.38625}, {0.3, 0.840415}, {0.4, 1.66763}, {0.5, 3.17433}, {0.6, 5.91862}, {0.7, 10.9171}, {0.8, 20.0212}, {0.9, 36.6036}, {1., 66.8067}}
```

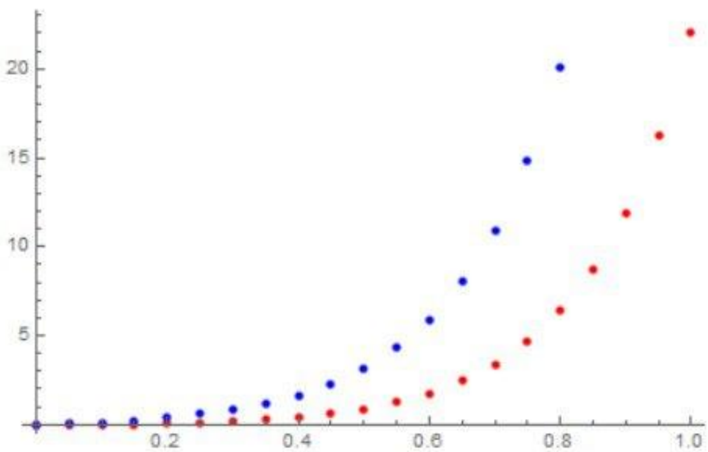


Для шага $h_2 = 0{,}05$ создадим таблицы приближенных решений и изобразим их графически:

```
ClearAll[x, y, z, h, n]
a = 0; b = 1; y0 = 1; z0 = 0; x0 = 0; h = 0.05; n = Floor[(b - a) / h];
sol7 = List[{x0, y0}];
sol8 = List[{x0, z0}];
x = x0; y = y0; z = z0;
For[k = 1, k < n + 1, k++,
 k1[x_, y_, z_] := h * f[x, y, z];
 k1z[x_, y_, z_] := h * g[x, y, z];
 k2z[x_, y_, z_] := h * g[x + h / 2, y + k1[x, y, z] / 2, z + k1z[x, y, z] / 2];
 k2[x_, y_, z_] := h * f[x + h / 2, y + k1[x, y, z] / 2, z + k1z[x, y, z] / 2];
 k3z[x_, y_, z_] := h * g[x + h / 2, y + k1[x, y, z] / 2, z + k2z[x, y, z] / 2];
 k3[x_, y_, z_] := h * f[x + h / 2, y + k2[x, y, z] / 2, z + k2z[x, y, z] / 2];
 k4z[x_, y_, z_] := h * g[x + h, y + k3[x, y, z], z + k3z[x, y, z]];
 k4[x_, y_, z_] := h * f[x + h, y + k3[x, y, z], z + k3z[x, y, z]];
 xTemp = x; yTemp = y; zTemp = z; (*Обновляем значения x,y,z (причем x опционален)*)
 x = xTemp + h;
 y = yTemp + (k1[xTemp, yTemp, zTemp] + 2 * k2[xTemp, yTemp, zTemp] + 2 * k3[xTemp, yTemp, zTemp] + k4[xTemp, yTemp, zTemp]) / 6;
 z = zTemp + (k1z[xTemp, yTemp, zTemp] + 2 * k2z[xTemp, yTemp, zTemp] + 2 * k3z[xTemp, yTemp, zTemp] + k4z[xTemp, yTemp, zTemp]) / 6;
 sol7 = Append[sol7, {x, y}];
 sol8 = Append[sol8, {x, z}]]
sol7
sol8
gr7 = ListPlot[sol7, PlotStyle → Red];
gr8 = ListPlot[sol8, PlotStyle → Blue];
Show[gr7, gr8]
```

```
{{0, 0}, {0.05, 0.00276875}, {0.1, 0.0123367}, {0.15, 0.0310826}, {0.2, 0.0622171}, {0.25, 0.110074}, {0.3, 0.180504}, {0.35, 0.281404}, {0.4, 0.423432}, {0.45, 0.620979},
 {0.5, 0.893465}, {0.55, 1.26711}, {0.6, 1.77729}, {0.65, 2.4718}, {0.7, 3.41509}, {0.75, 4.69421}, {0.8, 6.42665}, {0.85, 8.77099}, {0.9, 11.9413}, {0.95, 16.2265}, {1., 22.0167}}

{{0, 0}, {0.05, 0.0583062}, {0.1, 0.13701}, {0.15, 0.243248}, {0.2, 0.386651}, {0.25, 0.580223}, {0.3, 0.841512}, {0.35, 1.19421}, {0.4, 1.6703}, {0.45, 2.31294},
 {0.5, 3.18039}, {0.55, 4.35132}, {0.6, 5.93188}, {0.65, 8.06539}, {0.7, 10.9453}, {0.75, 14.8326}, {0.8, 20.08}, {0.85, 27.163}, {0.9, 36.7239}, {0.95, 49.6296}, {1., 67.0502}}
```



в) С помощью функций **DSolve** и **NDSolve**, построить графики:

Используем функцию **DSolve:**

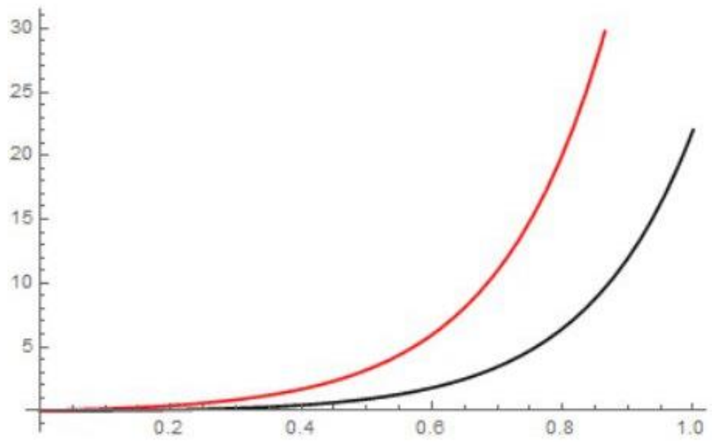
```
sol9 = DSolve[{y'[x] == f[x, y[x], z[x]], z'[x] == g[x, y[x], z[x]], y[x0] == y0, z[x0] == z0}, {y, z}, x];
ySol[x_] := y[x] /. Flatten[sol9[[1]]]
zSol[x_] := z[x] /. Flatten[sol9[[1]]]
ySol[x]
zSol[x]
```

$$\frac{1}{18}\left(-1+e^{6x}-6x\right)$$

$$\frac{1}{6}\left(-1+e^{6x}\right)$$

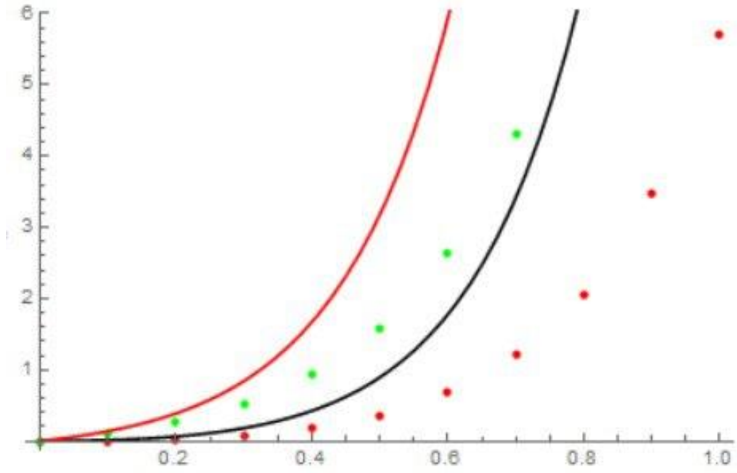
Используем функцию **NDSolve** и выведем графики искомых функций:

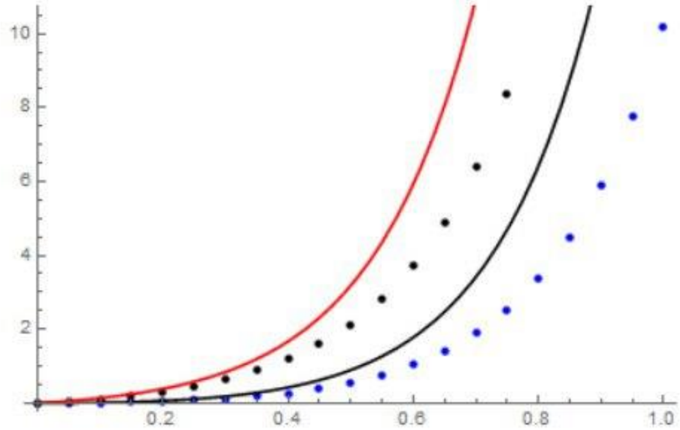
```
sol9 = DSolve[{y'[x] == f[x, y[x], z[x]], z'[x] == g[x, y[x], z[x]], y[x0] == y0, z[x0] == z0}, {y, z}, x];
ySol[x_] := y[x] /. Flatten[sol9[[1]]]
zSol[x_] := z[x] /. Flatten[sol9[[1]]]
ySol[x]
zSol[x]
```



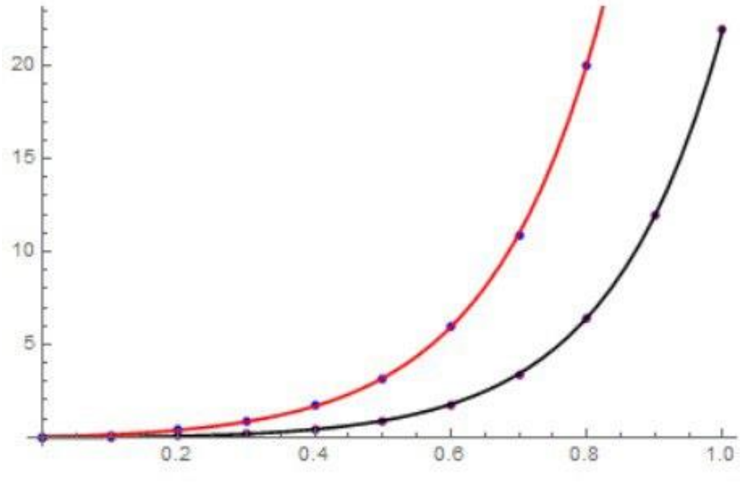
Сравнить все полученные решения:

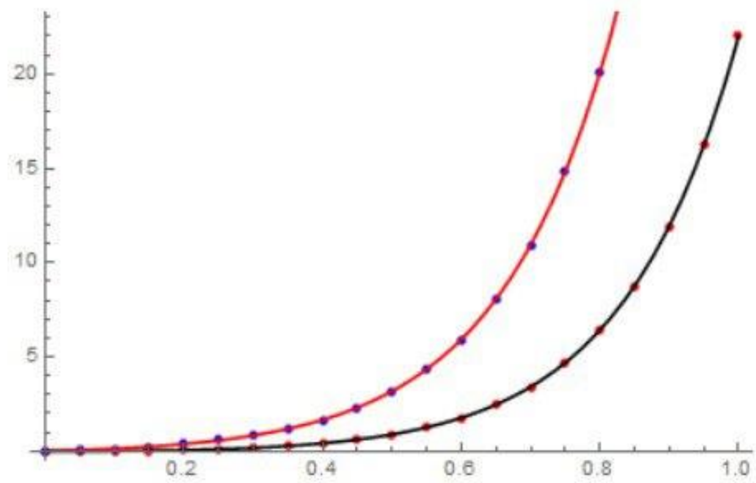
Изобразим приближенные решения метода Эйлера и искомые функции для шагов $h_1 = 0{,}1$ и $h_2 = 0{,}05$





Изобразим приближенные решения метода Рунге-Кутта и искомые функции для шагов $h_1 = 0{,}1$ и $h_2 = 0{,}05$





Найдем нормы абсолютных погрешностей (максимальная погрешность) каждого из методов для двух шагов:

```
tableY01 = Table[{x, y[x] /. sol10}, {x, a, b, 0.1}];
tableZ01 = Table[{x, z[x] /. sol10}, {x, a, b, 0.1}];
{tableY01, tableZ01}
tableY005 = Table[{x, y[x] /. sol10}, {x, a, b, 0.05}];
tableZ005 = Table[{x, z[x] /. sol10}, {x, a, b, 0.05}];
```

```
minLength = Min[Length[tableY01], Length[eul1]];

delta1y = Table[{tableY01[[i, 1]], tableY01[[i, 2]] - eul1[[i, 2]]}, {i, 1, minLength}];
normDelta1y = Norm[delta1y[[All, 2]], Infinity] // N;
normDelta1y

delta1z = Table[{tableZ01[[i, 1]], tableZ01[[i, 2]] - eul2[[i, 2]]}, {i, 1, minLength}];
normDelta1z = Norm[delta1z[[All, 2]], Infinity] // N;
normDelta1z

minLength2 = Min[Length[tableY005], Length[eul3]];

delta2y = Table[{tableY005[[i, 1]], tableY005[[i, 2]] - eul3[[i, 2]]}, {i, 1, minLength2}];
normDelta2y = Norm[delta2y[[All, 2]], Infinity] // N;
normDelta2y
delta2z = Table[{tableZ005[[i, 1]], tableZ005[[i, 2]] - eul4[[i, 2]]}, {i, 1, minLength2}];
normDelta2z = Norm[delta2z[[All, 2]], Infinity] // N;
normDelta2z

minLength3 = Min[Length[tableY01], Length[sol5]];

delta3y = Table[{tableY01[[i, 1]], tableY01[[i, 2]] - sol5[[i, 2]]}, {i, 1, minLength3}];
normDelta3y = Norm[delta3y[[All, 2]], Infinity] // N;
normDelta3y

delta3z = Table[{tableZ01[[i, 1]], tableZ01[[i, 2]] - sol6[[i, 2]]}, {i, 1, minLength3}];
normDelta3z = Norm[delta3z[[All, 2]], Infinity] // N;
normDelta3z


minLength4 = Min[Length[tableY005], Length[sol7]];

delta4y = Table[{tableY005[[i, 1]], tableY005[[i, 2]] - sol7[[i, 2]]}, {i, 1, minLength4}];
normDelta4y = Norm[delta4y[[All, 2]], Infinity] // N;
normDelta4y

delta4z = Table[{tableZ005[[i, 1]], tableZ005[[i, 2]] - sol8[[i, 2]]}, {i, 1, minLength4}];
normDelta4z = Norm[delta4z[[All, 2]], Infinity] // N;
normDelta4z
```

Максимальная погрешность метода Эйлера для шага $h_1 = 0{,}1$ для функций y и z:

16.3043 , 48.913

Максимальная погрешность метода Эйлера для шага $h_2 = 0{,}05$ для функций y и z:

11.8544 , 35.5632

Максимальная погрешность метода Рунге-Кутта четвертого порядка для шага $h_1 = 0{,}1$ для функций y и z:

0.0882654 , 0.264796

Максимальная погрешность метода Рунге-Кутта четвертого для шага $h_2 = 0{,}05$ для функций y и z:

0.00708184, 0.0212455

Вывод: точность каждого из методов увеличивается с уменьшением шага, метод Рунге-Кутта четвертого порядка точнее метода Эйлера.