Отчет по лабораторной работе номер 6

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

1. Решить задачу Коши для дифференциального уравнения первого порядка на отрезке [0,1] : а) методом Эйлера-Коши с шагом h1= 0,1 и h2= 0,05, построить графики полученных решений;

б) методом Рунге-Кутта 4-го порядка с шагом h1= 0,1 и h2= 0,05, построить графики полученных решений;

в) с помощью функций DSolve и NDSolve, построить графики.Сравнить все полученные решения. Сделать выводы о точности методов в зависимости от шага сетки.

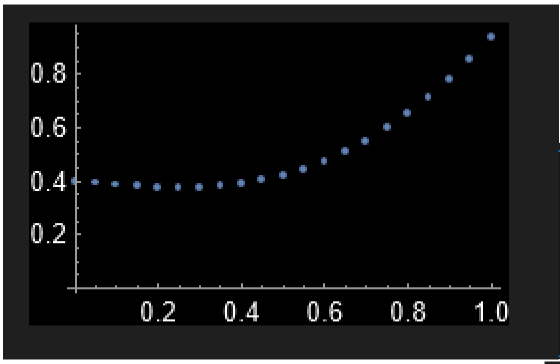
Вариант 15: y ′ = 2,5x^2 – 0,9y^2 , y(0) = 0,4.

Вычисление для шага 0.05

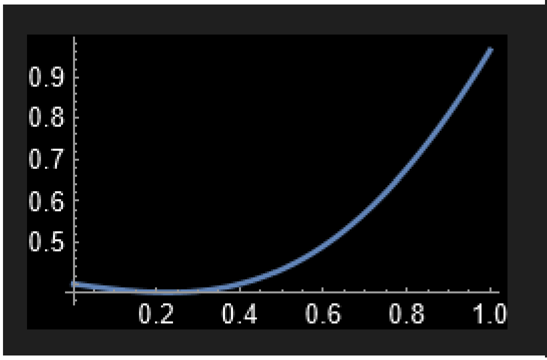
```
f[x_, y_] = 2.5 x^2 - 0.9 y^2; (* Определяем функцию *)
y0 = 0.4; (* Начальное значение y *)
x0 = 0; (* Начальное значение x *)
a = 0; (* Начало интервала *)
b = 1; (* Конец интервала *)
h = 0.05; (* Шаг интегрирования *)
n = Floor[(b - a)/h]; (* Количество шагов *)
```

Метод эйлера

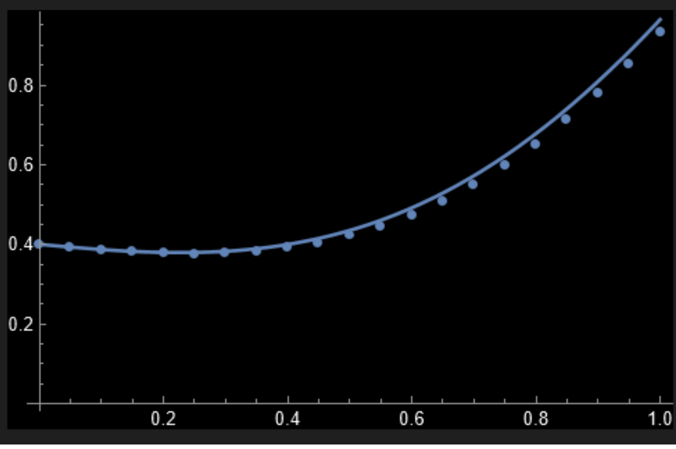
```
x = x0;
y = y0;
eul1 = Table[
  {x, y} = {x + h, y + h*f[x, y]},{i, n}
];

eul1 = Prepend[eul1, {x0, y0}]

gr1 = ListPlot[eul1, ImageSize -> Small]
```

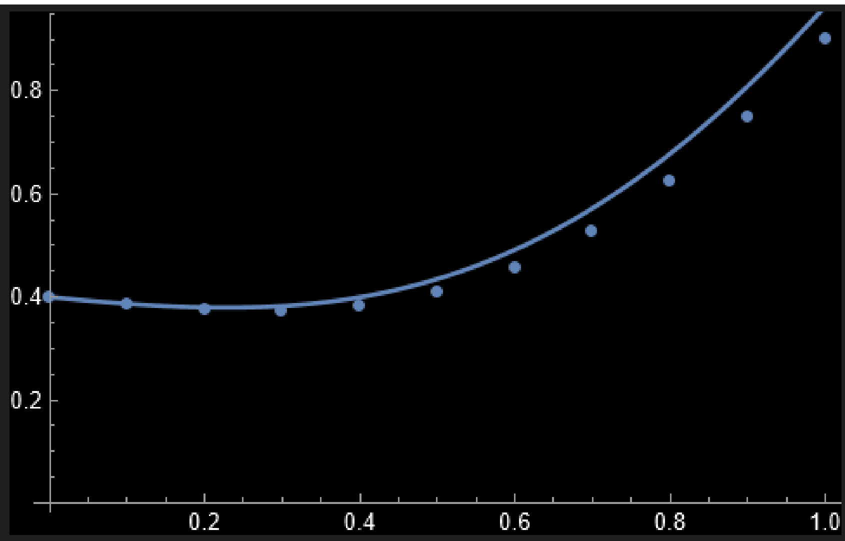



```
Clear[x, y];
sol = DSolve[{y'[x] == f[x, y[x]], y[x0] == y0}, y[x], x];
y1[x_] = y[x] /. Flatten[sol]
gr2 = Plot[y1[x], {x, a, b}, ImageSize -> Small]
```




```
Show[gr1, gr2, ImageSize-> Medium]
```





Исходя из полученных данных видно, что метод Эйлера имеет большую точность приближения при уменьшении шага интегрирования (что также показывает максимальная погрешность)

```
delta = Table[y1[x0 + i*h] - eul1[[i, 2]], {i, n + 1}]
Norm[delta, Infinity]
```

```
0.115993
```

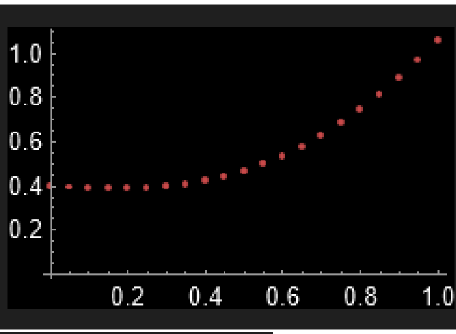
для шага 0.05

```
0.238701
```

для шага 0.1 (максимальная погрешность отличается в 2 раза)

Метод рунге-кутта

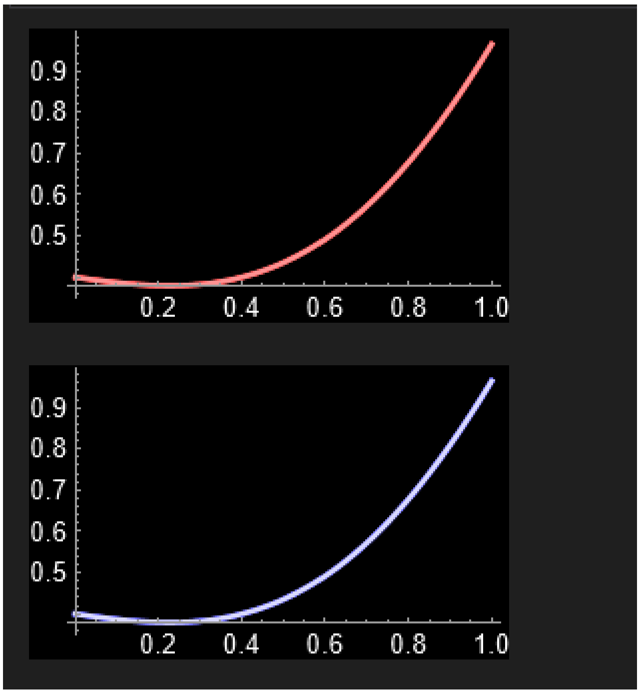
```
sol1 = List[{x0, y0}];
x=x0;
y=y0;
For[k=1 , k <= n, k++,
    k1[x_,y_] := h*f[x, y];
    k2[x_,y_] := h*f[x + h/2, y + k1[x,y]/2];
    k3[x_,y_] := h*f[x + h/2, y + k2[x,y]/2];
    k4[x_,y_] := h*f[x + h, y + k3[x,y]];
    x=x+h;
    y = y + (k1[x,y] + 2*k2[x,y] + 2*k3[x,y] + k4[x,y])/6;
    sol1 = Append[sol1, {x, y}];

];
sol1
gr1 = ListPlot[sol1, ImageSize -> Small, PlotStyle -> Pink]
```



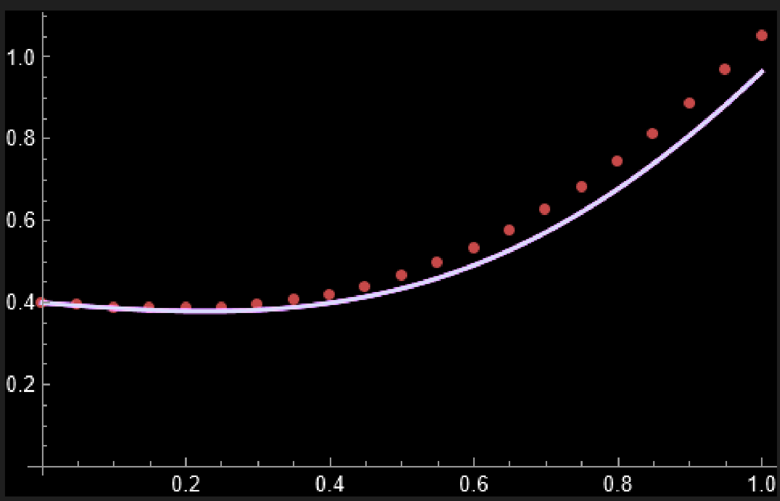

```
Clear[x,y];
sol2 = DSolve[{y'[x] == f[x, y[x]], y[x0] == y0}, y[x], x];
y1[x_] = y[x] /. Flatten[sol2]
```

```
sol3 = NDSolve[{y'[x] == f[x, y[x]], y[x0] == y0}, y[x], {x, a, b}]
```

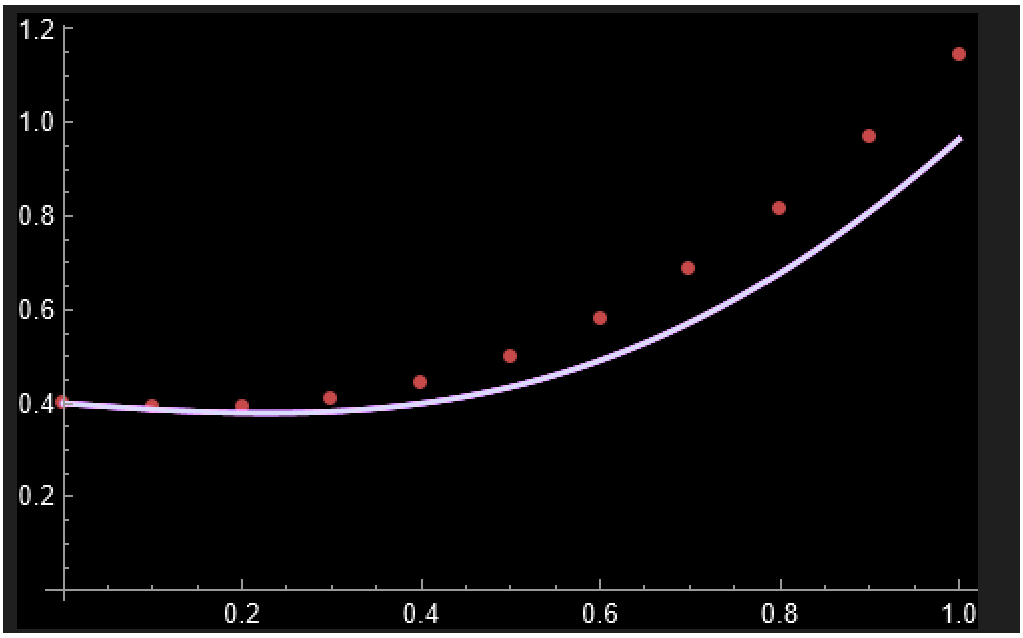
```
gr2 = Plot[y1[x], {x, a, b}, ImageSize -> Small, PlotStyle -> Red]
gr3 = Plot[Evaluate[y[x] /. sol3], {x, a, b}, ImageSize -> Small, PlotStyle -> Blue]
```




```
Show[gr1, gr2, gr3, ImageSize -> Medium, PlotRange -> All]
```



для шага 0.05



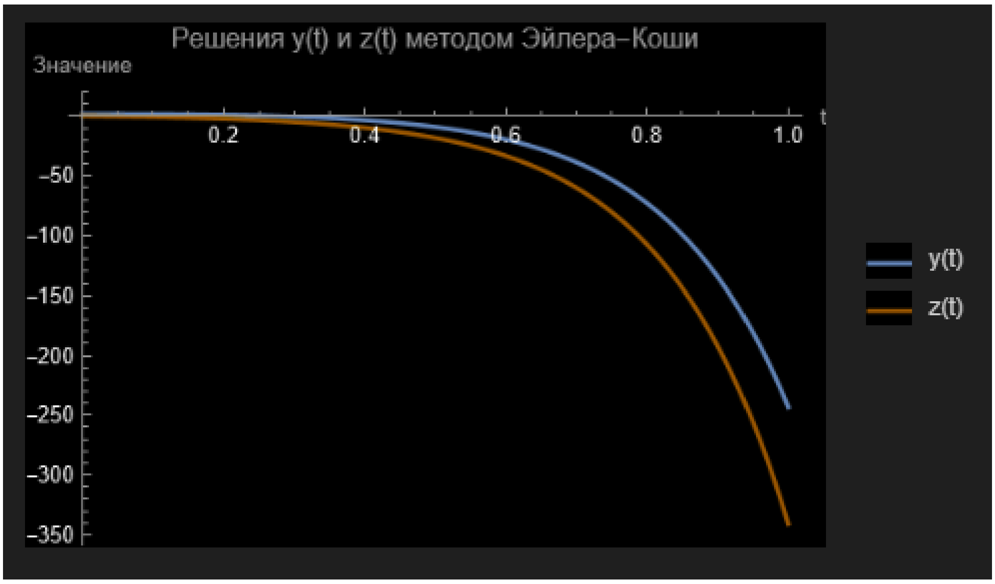
для шага 0.1

Для метода рунге-кутта ситуация аналогичная

Часть 2

метод эйлера

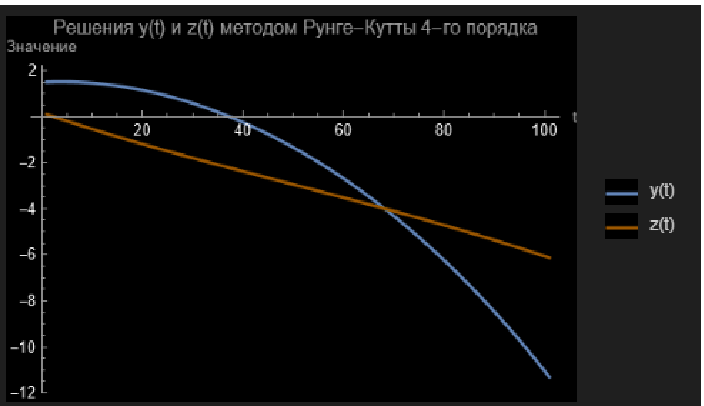
```
(* Определим уравнения *)
f1[t_, y_, z_] := 0.3*y + 4*z
f2[t_, y_, z_, yp_] := 1.6*yp - z - 8

(* Начальные условия *)
y0 = 1.5;
z0 = 0.1;
a = 0; (* начальное время *)
b = 1; (* конечное время *)
h = 0.01; (* шаг интегрирования *)

(* Численное решение методом Эйлера-Коши *)
nSteps = Floor[(b - a)/h]; (* число шагов *)
t = a;
y = y0;
z = z0;

(* Списки для хранения значений решений *)
tValues = Table[0, {nSteps + 1}];
yValues = Table[0, {nSteps + 1}];
zValues = Table[0, {nSteps + 1}];

(* Начальные значения *)
tValues[[1]] = t;
yValues[[1]] = y;
zValues[[1]] = z;

For[i = 1, i <= nSteps, i++,
  (* Шаг Эйлера *)
  yp = f1[t, y, z];
  yEuler = y + h * yp;
  zEuler = z + h * f2[t, y, z, yp];

  (* Использование промежуточных значений *)
  ypMid = f1[t + h, yEuler, zEuler];
  y = y + h/2 * (yp + ypMid);
  z = z + h/2 * (f2[t, y, z, yp] + f2[t + h, yEuler, zEuler, ypMid]);
  t = t + h;

  tValues[[i + 1]] = t;
  yValues[[i + 1]] = y;
  zValues[[i + 1]] = z;
]

(* Пример построения графиков решений *)
 gr1 = ListLinePlot[{Transpose[{tValues, yValues}], Transpose[{tValues, zValues}]},
 PlotLegends -> {"y(t)", "z(t)"}, PlotLabel -> "Решения y(t) и z(t) методом Эйлера-Коши",
 AxesLabel -> {"t", "Значение"}]
```



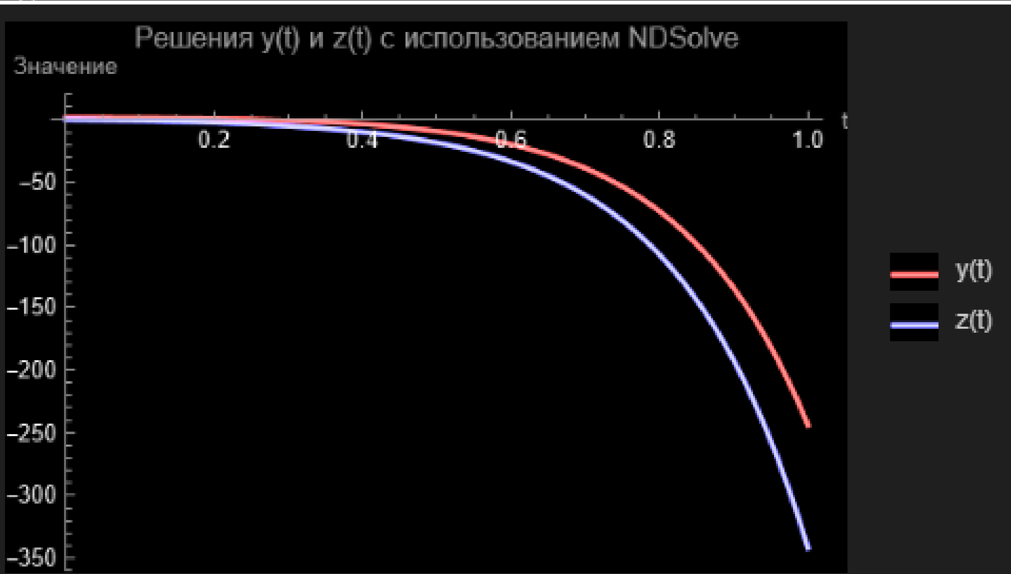
Метод рунге-кутта в моем случае не смог привести к достаточно приближенному решения, возможно из-за моей ошибки. Поэтому на итоговом графике его не будет

Рунге кутт

```
Clear[y, z];
y0 = 1.5;
z0 = 0.1;

f[y_, z_] := 0.3*y + 4*z;
g[y_, z_] := 1.6*0.3*y - z - 8;

a = 0; (* Начало на интервала *)
b = 1;  (* Край на интервала *)

h = 0.01;
n = Floor[(b - a)/h];
solRK4 = {{a, y0, z0}};
x = a;
y = y0;
z = z0;
Do[
  k1 = h*f[y, z];
  l1 = h*g[y, z];
  k2 = h*f[y + k1/2, z + l1/2];
  l2 = h*g[y + k1/2, z + l1/2];
  k3 = h*f[y + k2/2, z + l2/2];
  l3 = h*g[y + k2/2, z + l2/2];
  k4 = h*f[y + k3, z + l3];
  l4 = h*g[y + k3, z + l3];
  y = y + (k1 + 2 k2 + 2 k3 + k4)/6;
  z = z + (l1 + 2 l2 + 2 l3 + l4)/6;
  x = x + h;
  AppendTo[solRK4, {x, y, z}],
  {n}
]

solRK4;
gr2 = ListLinePlot[{Transpose[solRK4][[2]], Transpose[solRK4][[3]]},
 PlotLegends -> {"y(t)", "z(t)"},
 PlotLabel -> "Решения y(t) и z(t) методом Рунге-Кутты 4-го порядка",
 AxesLabel -> {"t", "Значение"}]
```



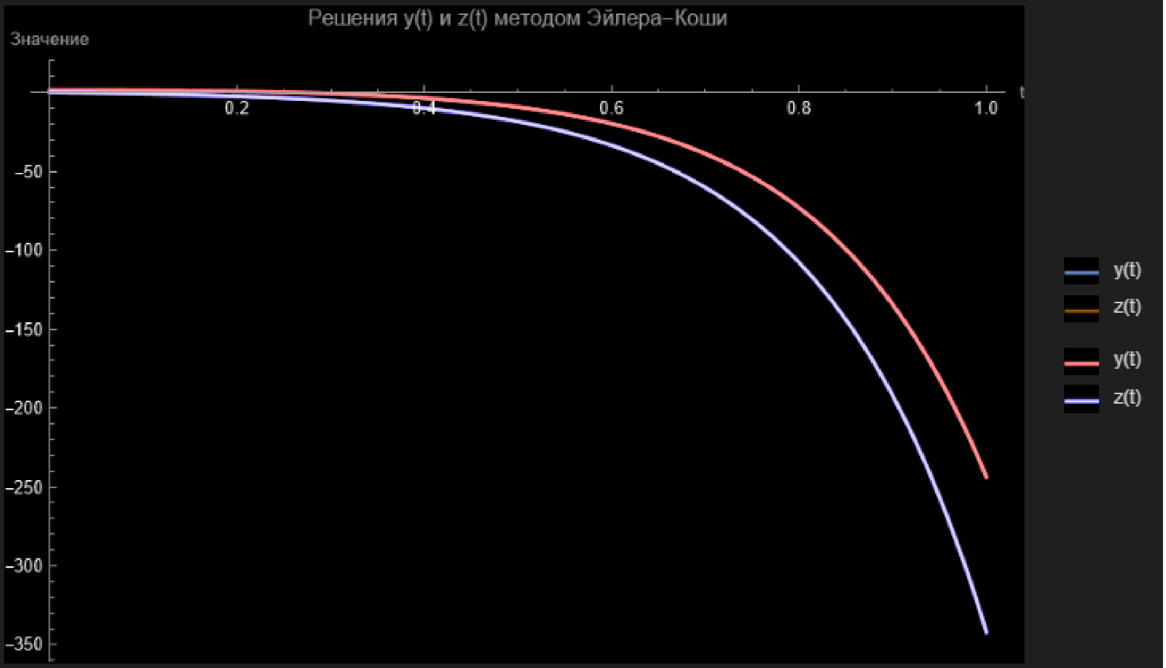

```
Clear[y, z,x];
y0 = 1.5;
z0 = 0.1;
a = 0; (* начальное значение x *)
b = 1; (* конечное значение x *)

(* Определим систему уравнений для NDSolve *)
eq1 = y'[x] == 0.3*y[x] + 4*z[x];
eq2 = z'[x] == 1.6*y'[x] - z[x] - 8;
ics = {y[a] == y0, z[a] == z0};

(* Численное решение с использованием NDSolve *)
solution = NDSolve[{eq1, eq2, ics}, {y, z}, {x, a, b}];
gr3 = Plot[Evaluate[{y[x], z[x]} /. solution], {x, a, b},
 PlotLegends -> {"y(t)", "z(t)"}, PlotLabel -> "Решения y(t) и z(t) с использованием NDSolve",
 AxesLabel -> {"t", "Значение"}, PlotStyle -> {Red, Blue}]
```



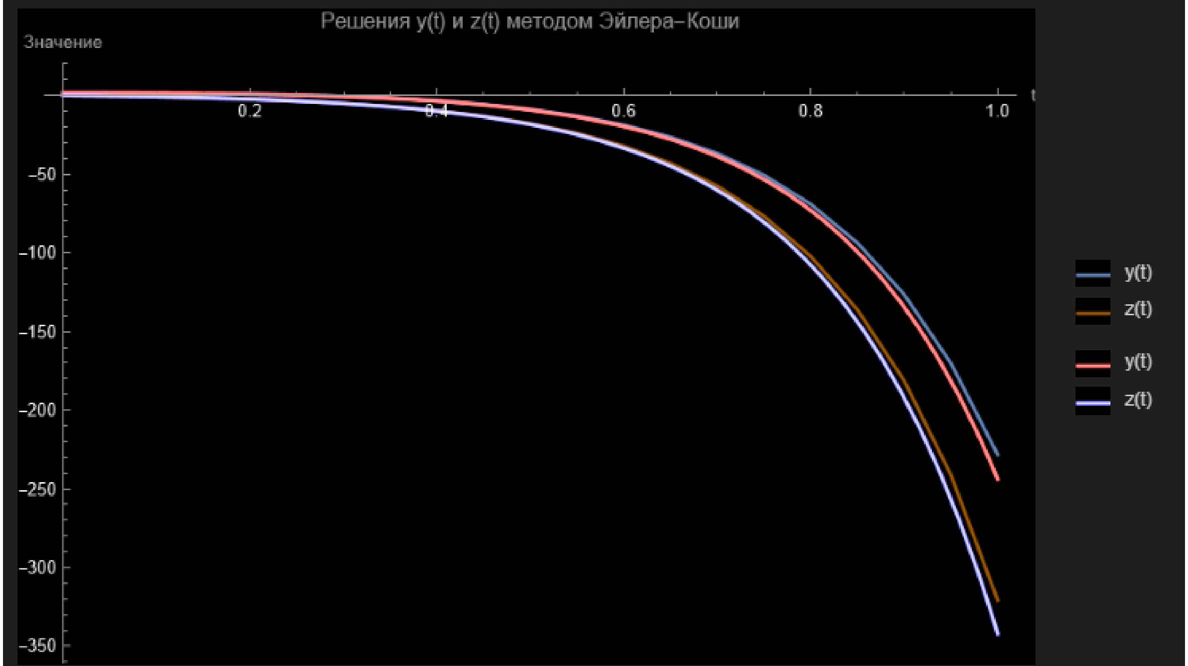
Для шага 0.01

```
Show[gr1, gr3, ImageSize -> Large, PlotRange -> All]
```

0.6s



Для шага 0.05



Исходя из полученных результатов видно что при решении системы методом эйлера-коши при уменьшении шага увеличивается точность приближения (уменьшается погрешность)