
Телица Илья Денисович

гр. 221701

Вариант 12

Задание 1

```
In[1]:= f[x_, y_] = 0.2 x^2 + 5 * y^2; x0 = 0; y0 = 0.8; a = 0; b = 1;
h = 0.1; n =  $\frac{b - a}{h}$ ;
Koshi1 = Table[1, {x, 0, n}, {y, 1, 2}];
таблица значений
Koshi1[[1, 1]] = x0;
Koshi1[[1, 2]] = y0;
(*Метод Эйлера для грубых значений*)
For[k = 2, k ≤ n + 1, k++,
цикл для
  Koshi1[[k, 1]] = Koshi1[[k - 1, 1]] + h;
  Koshi1[[k, 2]] = Koshi1[[k - 1, 2]] + h * f[Koshi1[[k - 1, 1]], Koshi1[[k - 1, 2]]];
];

(*Уточнени метод Эйлера для метода Эйлера-Коши*)
For[цикл для k = 2, k ≤ n + 1, k++,
  Koshi1[[k, 2]] = Koshi1[[k - 1, 2]] +
     $\frac{h}{2}$  * (f[Koshi1[[k - 1, 1]], Koshi1[[k - 1, 2]]] + f[Koshi1[[k, 1]], Koshi1[[k, 2]]]);
];
Koshi1Graph = ListPlot[Koshi1];
диаграмма разброса данных

h = 0.05; n =  $\frac{b - a}{h}$ ;
Koshi2 = Table[1, {x, 0, n}, {y, 1, 2}];
таблица значений
Koshi2[[1, 1]] = x0;
Koshi2[[1, 2]] = y0;
(*Метод Эйлера для грубых значений*)
For[k = 2, k ≤ n + 1, k++,
цикл для
  Koshi2[[k, 1]] = Koshi2[[k - 1, 1]] + h;
  Koshi2[[k, 2]] = Koshi2[[k - 1, 2]] + h * f[Koshi2[[k - 1, 1]], Koshi2[[k - 1, 2]]];
];
```

(*Уточнени метод Эйлера для метода Эйлера-Коши*)

For[k = 2, k ≤ n + 1, k++,
цикл для

Koshi2[[k, 2]] = Koshi2[[k - 1, 2]] +

$\frac{h}{2} * (f[Koshi2[[k - 1, 1]], Koshi2[[k - 1, 2]]] + f[Koshi2[[k, 1]], Koshi2[[k, 2]]]);$

];

Koshi2Graph = ListPlot[Koshi2];

диаграмма разброса данных

Echo[Koshi1 // TableForm, "Метод Эйлера-Коши для h=0.1"];
дублировать на ... табличная форма

Show[Koshi1Graph]

показать

Print[]

печатать

Print[]

печатать

Print[]

печатать

Echo[Koshi2 // TableForm, "Метод Эйлера-Коши для h=0.05"];
дублировать на ... табличная форма

дублировать на ... табличная форма

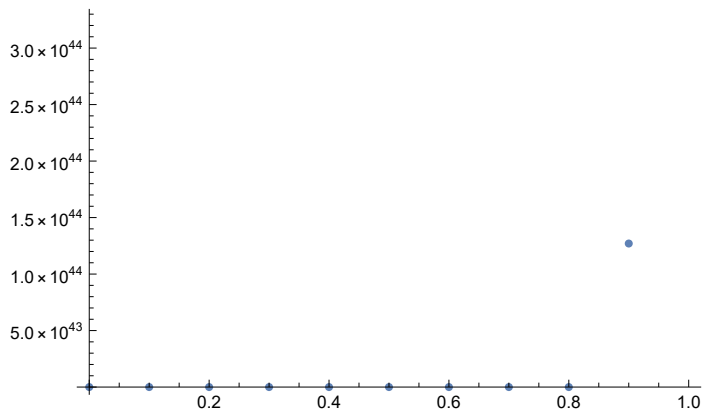
Show[Koshi2Graph]

показать

ListPlot: Value of option PlotRange -> {{0, 1.}, {0, 6.698938900702506 × 10¹²⁹²}} is not All, Full, Automatic, a positive machine number, or an appropriate list of range specifications.

	0	0.8
	0.1	1.2737
	0.2	2.44313
	0.3	6.6179
	0.4	36.2285
» Метод Эйлера-Коши для h=0.1	0.5	892.445
	0.6	503 685.
	0.7	1.5598 × 10 ¹¹
	0.8	1.4649 × 10 ²²
	0.9	1.27033 × 10 ⁴⁴
	1.	9.41971 × 10 ⁸⁷

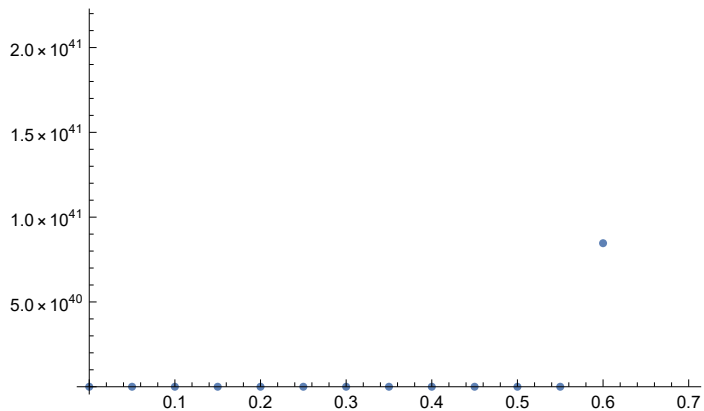
Out[17]=



» Метод Эйлера–Коши для $h=0.05$

0	0.8
0.05	0.995213
0.1	1.29622
0.15	1.80471
0.2	2.78546
0.25	5.10776
0.3	12.8594
0.35	61.5568
0.4	1165.81
0.45	392905.
0.5	4.40539×10^{10}
0.55	5.49048×10^{20}
0.6	8.46389×10^{40}
0.65	1.99796×10^{81}
0.7	1.10672×10^{162}
0.75	$3.37765918386280 \times 10^{323}$
0.8	$3.131380066376218 \times 10^{646}$
0.85	$2.679575560281003 \times 10^{1292}$
0.9	$1.954403420831445 \times 10^{2584}$
0.95	$1.035967473698403 \times 10^{5168}$
1.	$2.901179860900836 \times 10^{10335}$

Out[22]=



```

In[23]:= h = 0.1; n =  $\frac{b-a}{h}$ ;
x = x0; y = y0;
Runge1 = List[{x0, y0}];
| список
For[k = 1, k ≤ n, k++,
| цикл ДЛЯ
  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;
  Runge1 = Append[Runge1, {x, y}]
| добавить в конец
Runge1Graph = ListPlot[Runge1];
| диаграмма разброса данных

```

```

h = 0.05; n =  $\frac{b-a}{h}$ ;
x = x0; y = y0;
Runge2 = List[{x0, y0}];
| список
For[k = 1, k ≤ n, k++,
| цикл ДЛЯ
  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;
  Runge2 = Append[Runge2, {x, y}]
| добавить в конец
Runge2Graph = ListPlot[Runge2];
| диаграмма разброса данных

```

```

Echo[Runge1 // TableForm, "Метод Рунге-Кутта для h=0.1"];
| дублировать на ... | табличная форма
Show[Runge1Graph]
| показать
Print[]
| печатать
Print[]
| печатать
Print[]
| печатать
Echo[Runge2 // TableForm, "Метод Рунге-Кутта для h=0.05"];
| дублировать на ... | табличная форма
Show[Runge2Graph]
| показать

```

... ListPlot: Value of option PlotRange -> $\{\{0, 1.\}, \{0, 4.068923066884292 \times 10^{46424125}\}\}$ is not All, Full, Automatic, a positive machine number, or an appropriate list of range specifications.

... General: Overflow occurred in computation.

... General: Overflow occurred in computation.

... General: Overflow occurred in computation.

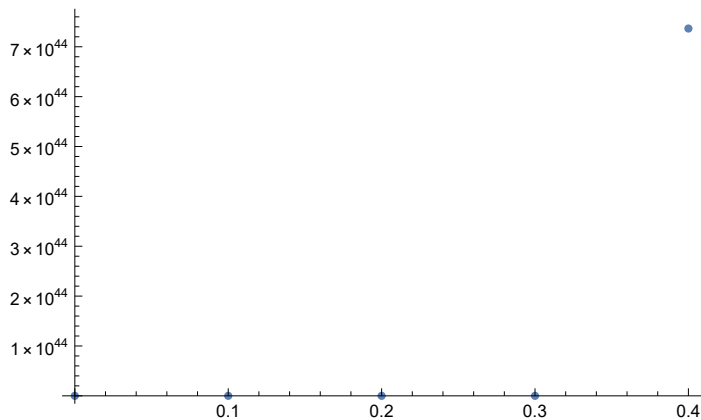
... General: Further output of General::ovfl will be suppressed during this calculation.

... ListPlot: Value of option PlotRange -> $\{\{0, 1.\}, \{0, 2.375078583252178 \times 10^{189322902248246}\}\}$ is not All, Full, Automatic, a positive machine number, or an appropriate list of range specifications.

» Метод Рунге–Кутты для $h=0.1$

0	0.8
0.1	1.33224
0.2	3.82825
0.3	2293.22
0.4	7.3659×10^{44}
0.5	$9.32514721257163 \times 10^{708}$
0.6	$4.060037191780140 \times 10^{11334}$
0.7	$6.768914953908361 \times 10^{181344}$
0.8	$2.411813866111534 \times 10^{2901508}$
0.9	$1.627569226753717 \times 10^{46424125}$
1.	$3.010720114935125 \times 10^{742785994}$

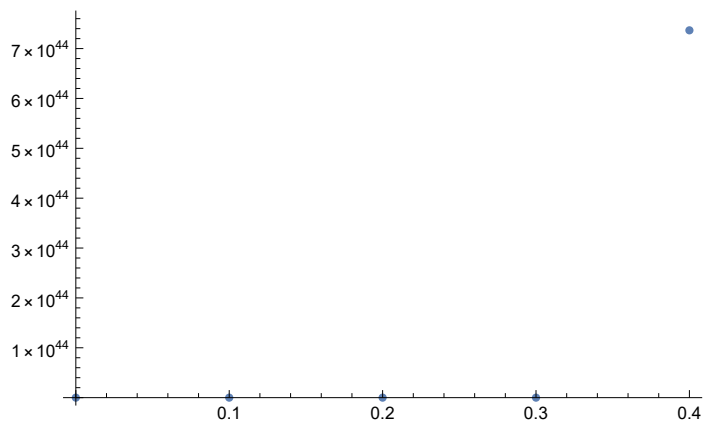
Out[34]=



» Метод Рунге–Кутты для $h=0.05$

0	0.8
0.1	1.33224
0.2	3.82825
0.3	2293.22
0.4	7.3659×10^{44}
0.5	$9.32514721257163 \times 10^{708}$
0.6	$4.060037191780140 \times 10^{11334}$
0.7	$6.768914953908361 \times 10^{181344}$
0.8	$2.411813866111534 \times 10^{2901508}$
0.9	$1.627569226753717 \times 10^{46424125}$
1.	$3.010720114935125 \times 10^{742785994}$

Out[39]=



```

In[40]:= Clear[x, y]
[очистить]
dsolve1 = DSolve[{y'[x] == f[x, y[x]], y[x0] == y0}, y[x], x];
[решить дифференциальные уравнения]
dsolve1Func[x_] = y[x] /. Flatten[dsolve1];
[уплостить]
dsolve1Graph = Plot[dsolveFunc[x], {x, a, b}];
[график функции]

Clear[x, y]
[очистить]
ndsolve1 = NDSolve[{y'[x] == f[x, y[x]], y[x0] == y0}, y[x], {x, a, b}];
[численно решить ДУ]
ndsolve1Graph = Plot[Evaluate[y[x] /. ndsolve1], {x, a, b}];
[гр... [вычислить]

Print["Метод DSolve"]
[печатать [решить дифференциальные уравнения]
Show[dsolve1Graph]
[показать]

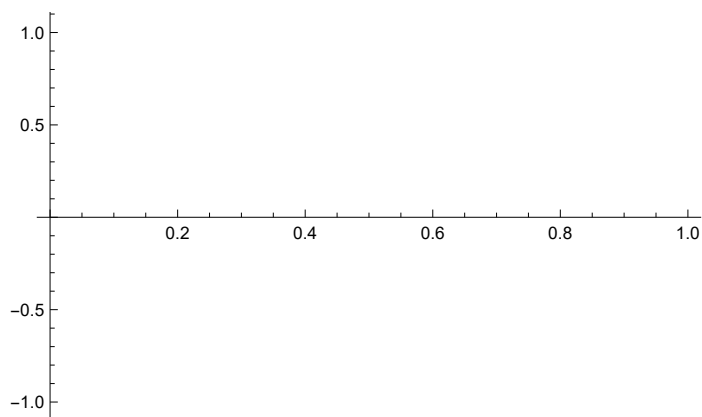
Print["Метод NDSolve"]
[печатать [численно решить ДУ]
Show[ndsolve1Graph]
[показать]

... NDSolve: At x == 0.2499674050289627, step size is effectively zero; singularity or stiff system suspected.

```

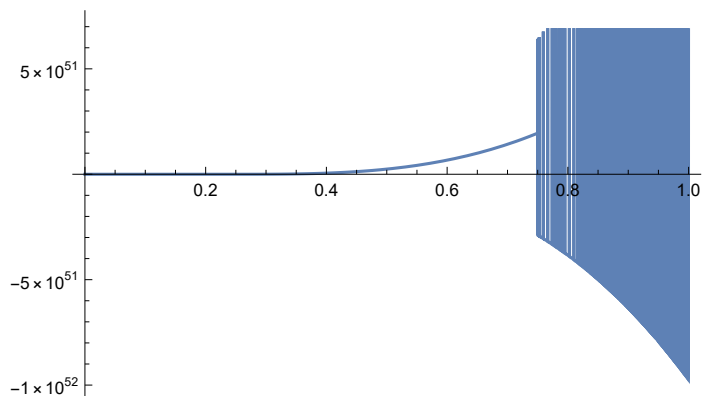
Метод DSolve

Out[48]=



Метод NDSolve

Out[50]=



При уменьшении шага точность решение дифференциального уравнения возрастает

Задание 2

```

In[51]:= f[y_, z_, x_] = 0.7 * y + z + Sin[2 * x];
                                         |синус

g[y_, z_] = y + 0.4 * z;

x0 = 0; y[x0] = 0; z[x0] = 0.6; a = 0; b = 1;
h = 0.1; n =  $\frac{b - a}{h}$ ;

Do[
|оператор цикла
  y[i + 1] = y[i] + h * f[y[i], z[i], i];
  z[i + 1] = z[i] + h * g[y[i], z[i]]; {i, 0, n}]

EulerYGraph1 = ListPlot[Table[{i * h, y[i]}, {i, 0, n}]];
|диаграмм·· |таблица значений
EulerZGraph1 = ListPlot[Table[{i * h, z[i]}, {i, 0, n}]];
|диаграмм·· |таблица значений
h = 0.05; n =  $\frac{b - a}{h}$ ;

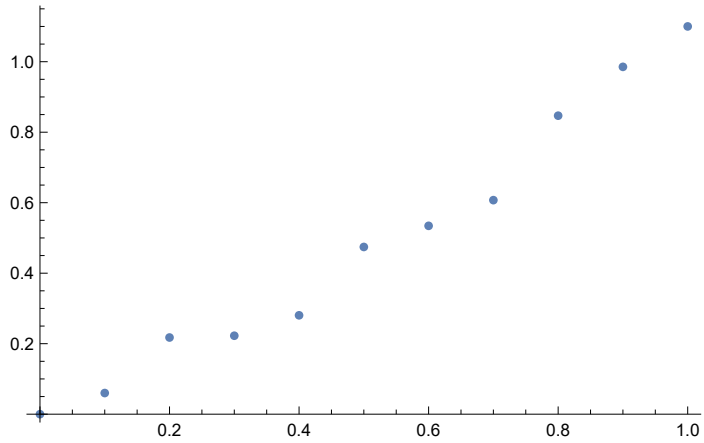
Do[
|оператор цикла
  y[i + 1] = y[i] + h * f[y[i], z[i], i];
  z[i + 1] = z[i] + h * g[y[i], z[i]]; {i, 0, n}]

EulerYGraph2 = ListPlot[Table[{i * h, y[i]}, {i, 0, n}]];
|диаграмм·· |таблица значений
EulerZGraph2 = ListPlot[Table[{i * h, z[i]}, {i, 0, n}]];
|диаграмм·· |таблица значений
Print["Решения методом Эйлера для шага h=0.1"]
|печатать
Show[EulerYGraph1]
|показать
Show[EulerZGraph1]
|показать
Print["Решения методом Эйлера для шага h=0.05"]
|печатать
Show[EulerYGraph2]
|показать
Show[EulerZGraph2]
|показать

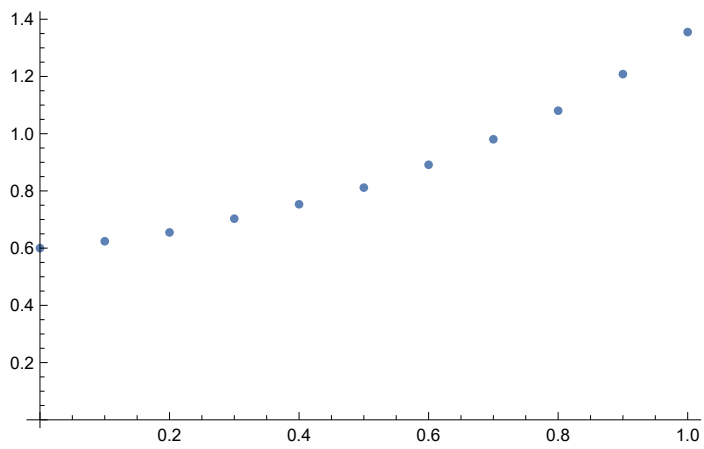
Решения методом Эйлера для шага h=0.1

```

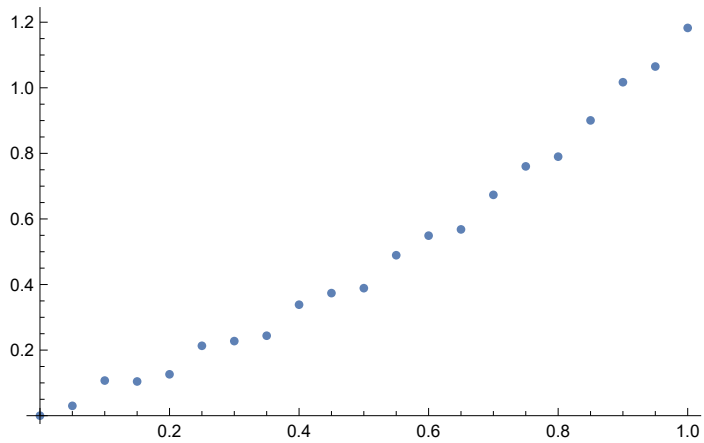

Out[63]=



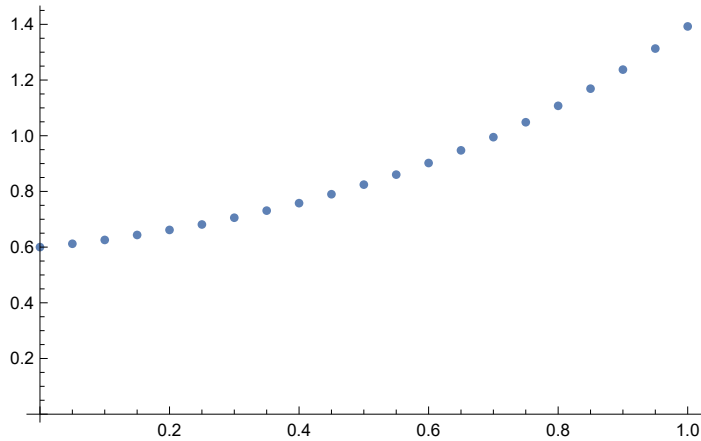
Out[64]=

Решения методом Эйлера для шага $h=0.05$

Out[66]=



Out[67]=



```

In[68]:= Clear[y, z];
          ОЧИСТИТЬ
y[x0] = 0; z[x0] = 1;
h = 0.1; n =  $\frac{b-a}{h}$ ;

Do[
  оператор цикла
  k1 = h * f[y[i], z[i], i];
  l1 = h * (y[i] - z[i]);
  k2 = h * f[y[i] + h / 2, z[i] + l1 / 2];
  l2 = h * g[y[i] + h / 2, z[i] + k1 / 2];
  k3 = h * f[y[i] + h / 2, z[i] + l2 / 2];
  l3 = h * g[y[i] + h / 2, z[i] + k2 / 2];
  k4 = h * f[y[i] + h, z[i] + l3];
  l4 = h * g[y[i] + h, z[i] + k3];
  y[i + 1] = y[i] + 1 / 6 * (k1 + 2 * k2 + 2 * k3 + k4);
  z[i + 1] = z[i] + 1 / 6 * (l1 + 2 * l2 + 2 * l3 + l4);, {i, 0, n}]
RungeYGraph1 = ListPlot[Table[{i * h, y[i]}, {i, 0, n}]];
               диаграмм · таблица значений
RungeZGraph1 = ListPlot[Table[{i * h, z[i]}, {i, 0, n}]];
               диаграмм · таблица значений

h = 0.05; n =  $\frac{b-a}{h}$ ;

Do[
  оператор цикла
  k1 = h * f[y[i], z[i], i];
  l1 = h * (y[i] - z[i]);
  k2 = h * f[y[i] + h / 2, z[i] + l1 / 2, i];
  l2 = h * g[y[i] + h / 2, z[i] + k1 / 2];
  k3 = h * f[y[i] + h / 2, z[i] + l2 / 2, i];
  l3 = h * g[y[i] + h / 2, z[i] + k2 / 2];
  k4 = h * f[y[i] + h, z[i] + l3, i];
  l4 = h * g[y[i] + h, z[i] + k3];
  y[i + 1] = y[i] + 1 / 6 * (k1 + 2 * k2 + 2 * k3 + k4);
  z[i + 1] = z[i] + 1 / 6 * (l1 + 2 * l2 + 2 * l3 + l4);, {i, 0, n}]
RungeYGraph2 = ListPlot[Table[{i * h, y[i]}, {i, 0, n}]];
               диаграмм · таблица значений

```

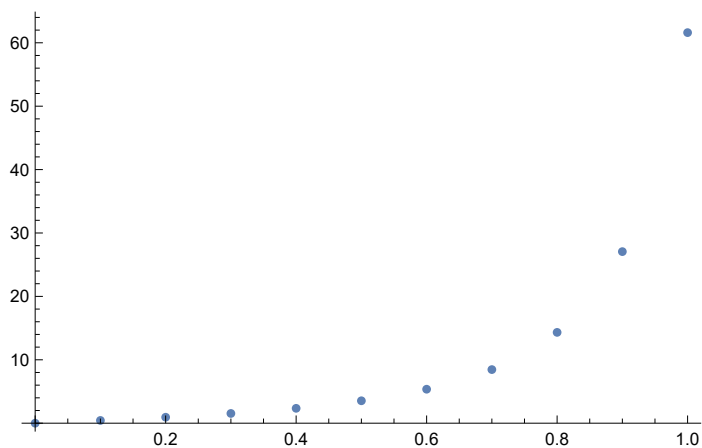
```

RungeZGraph2 = ListPlot[Table[{i * h, z[i]}, {i, 0, n}]];
Print["Решения методом Рунге-Кутты для шага h=0.1"]
Show[RungeYGraph1]
Show[RungeZGraph1]
Print["Решения методом Рунге-Кутты для шага h=0.05"]
Show[RungeYGraph2]
Show[RungeZGraph2]

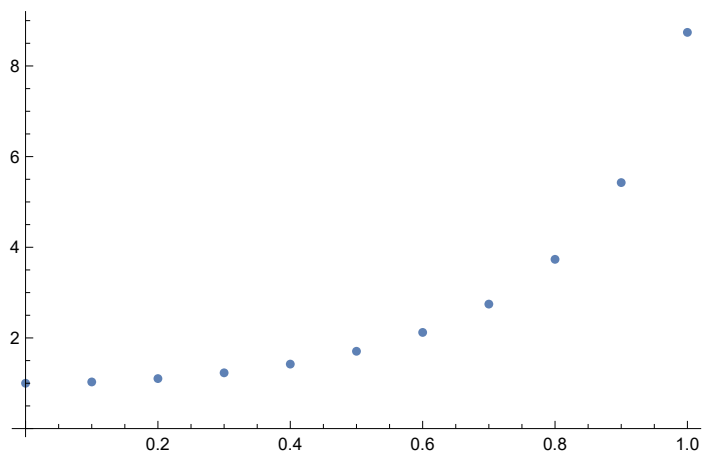
```

Решения методом Рунге-Кутты для шага h=0.1

Out[79]=

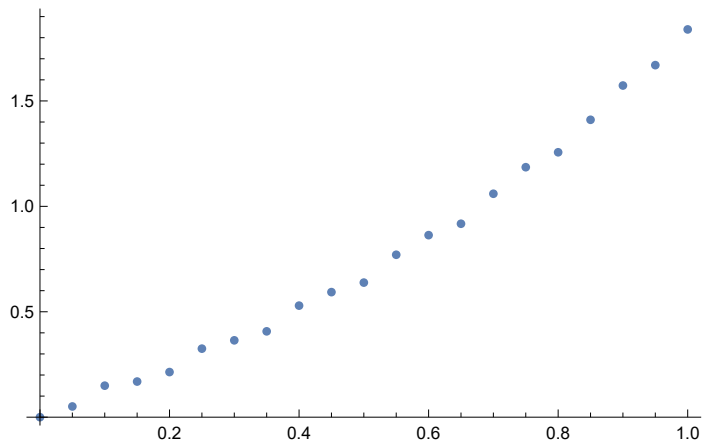


Out[80]=

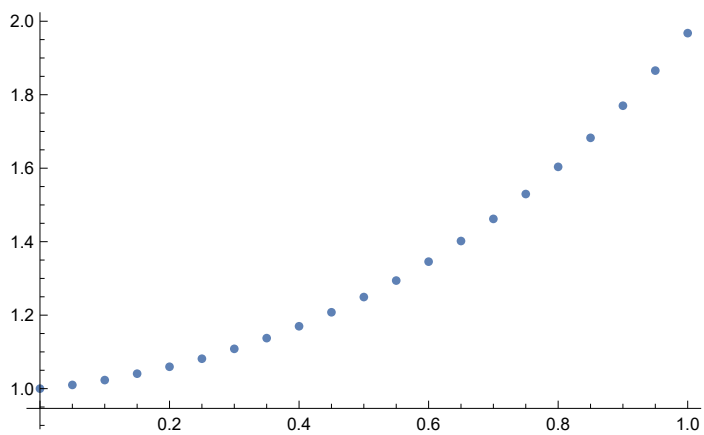


Решения методом Рунге-Кутты для шага h=0.05

Out[82]=



Out[83]=



In[132]:=

```

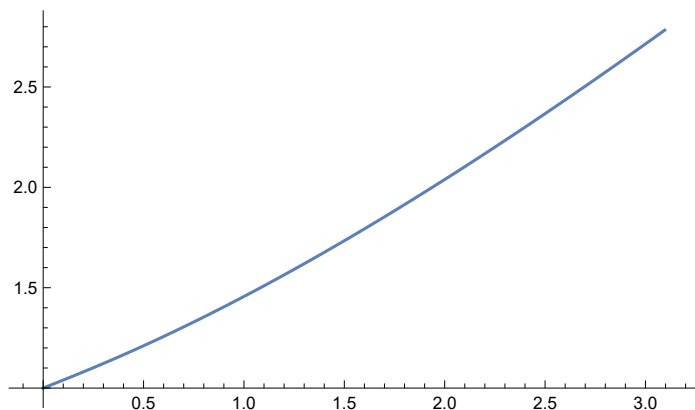
Clear[y, z];
очистить
sol1 = DSolve[{y'[x] == 0.7 * y[x] + z[x] + Sin[2 * x],
решить дифференциальные уравнения синус
  z'[x] == y[x] + 0.4 * z[x], y[0] == 0, z[0] == 1}, {y, z}, x];
sol2 = NDSolve[{y'[x] == 0.7 * y[x] + z[x] + Sin[2 * x],
численно решить ДУ синус
  z'[x] == y[x] + 0.4 * z[x], y[0] == 0, z[0] == 1}, {y, z}, {x, a, b}];
DSolveGraph = ParametricPlot[Evaluate[{y[x], z[x]} /. sol1], {x, a, b}];
график параметр... вычислить
NDSolveGraph = ParametricPlot[Evaluate[{y[x], z[x]} /. sol2], {x, a, b}];
график параметр... вычислить

Print["Решения методом DSolve"]
печатать решить дифференциальные уравнения
Show[DSolveGraph]
показать
Print["Решения методом NDSolve"]
печатать численно решить ДУ
Show[NDSolveGraph]
показать

```

Решения методом DSolve

Out[137]=



Решения методом NDSolve

Out[139]=

