Санкт-Петербургский национальный исследовательский университет ИТМО

Вычислительная математика

Лабораторная работа №6 ЧИСЛЕННОЕ РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Автор:

Ненов Владислав Александрович

Вариант 5

Группа №P32082

Преподаватель:

Екатерина Алексеевна Машина

Санкт-Петербург
2023

Цель работы

Решить задачу Коши для обыкновенных дифференциальных уравнений численными методами. № варианта задания лабораторной работы определяется как номер в списке группы согласно ИСУ.

Порядок выполнения работы

2. В программе численные методы решения обыкновенных дифференциальных уравнений (ОДУ) должен быть реализован в виде отдельного класса /метода/функции;
3. Пользователь выбирает ОДУ вида $y' = f(x, y)$ (не менее трех уравнений), из тех, которые предлагает программа;
4. Предусмотреть ввод исходных данных с клавиатуры: начальные условия $y_0 = y(x_0)$, интервал дифференцирования $[x_0, x_n]$, шаг h, точность ε;
5. Для исследования использовать одношаговые методы и многошаговые методы (см. табл.1);
6. Составить таблицу приближенных значений интеграла дифференциального уравнения, удовлетворяющего начальным условиям, для всех методов, реализуемых в программе;
7. Для оценки точности одношаговых методов использовать правило Рунге: $R = \frac{y^h - y^{h/2}}{2^p - 1} \le \varepsilon$;
8. Для оценки точности многошаговых методов использовать точное решение задачи: $\varepsilon = \max_{0 \le i \le n} |y_{i\text{точн}} - y_i|$;
9. Построить графики точного решения и полученного приближенного решения (разными цветами);
10. Программа должна быть протестирована при различных наборах данных, в том числе и некорректных.
11. Проанализировать результаты работы программы.

Описание алгоритма решения

По выданному варианту необходимо реализовать 3 алгоритма, 2 из которых являются одношаговыми и один - многошаговым.
Одношаговые: *Модифицированный метод Эйлера*, *Метод Рунге-Кутта*.
Многошаговые: *Метод Адамса*.
В программе реализован общий базовый класс *ODUSolveMethod*, в котором размещена логика итерации по интервалу значений функции, проверка точности метода и уменьшения шага в случае ее несоблюдения. Конкретные формулы и вычисления представлены в отдельных классах методов в методе *calculateNext*.

Рабочие формулы

Модифицированный метод эйлера

$$y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f\big(x_{i+1}, y_i + hf(x_i, y_i)\big)\right], \; i = 0, 1 \dots.$$

Метод Рунге-Кутта

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

$$k_1 = h \cdot f(x_i, y_i)$$
$$k_2 = h \cdot f(x_i + \frac{h}{2}, y_i + \frac{k_1}{2})$$
$$k_3 = h \cdot f(x_i + \frac{h}{2}, y_i + \frac{k_2}{2})$$
$$k_4 = h \cdot f(x_i + h, y_i + k_3)$$

Метод Адамса

$$y_{i+1} = y_i + hf_i + \frac{h^2}{2}\Delta f_i + \frac{5h^3}{12}\Delta^2 f_i + \frac{3h^4}{8}\Delta^3 f_i$$

$$\Delta f_i = f_i - f_{i-1}$$
$$\Delta^2 f_i = f_i - 2f_{i-1} + f_{i-2}$$
$$\Delta^3 f_i = f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}$$

Листинг программы

На языке Kotlin

```
const val MAX_ITERATIONS = 10_000

abstract class ODUSolveMethod(val start: Double, val end: Double, val y0:
Double, val func: (x: Double, y: Double) -> Double,
                             var h: Double, val eps: Double) {
   val result = ArrayList<Double>()
   abstract val name: String

   fun calculate() {
       var iters = 0
       while (result.size < (end-start)/h && iters < MAX_ITERATIONS) {
           onCalculateStart()
           var x = start+h
           while (x < end+h/2 && iters < MAX_ITERATIONS) {
               result.add(calculateNext(result, h, x))
               iters++
               if (!checkError(result, h, x)) {
                   onMethodReload()
                   break
               }
               x += h
           }
       }
       if (iters >= MAX_ITERATIONS)
           throw IllegalArgumentException()
   }

   protected open fun onCalculateStart() {
       result.clear()
       result.add(y0)
   }
   protected open fun onMethodReload() { h /= 2 }
   protected abstract fun calculateNext(calculated: List<Double>, h: Double,
x: Double) : Double
   protected abstract fun checkError(calculated: List<Double>, h: Double, x:
Double) : Boolean
}

abstract class OneStepODUSolveMethod(
   start: Double,
   end: Double,
   y0: Double,
   func: (x: Double, y: Double) -> Double,
   h: Double,
   eps: Double
) : ODUSolveMethod(start, end, y0, func, h, eps) {

   protected abstract val p: Double
```

```kotlin
    protected abstract fun calculateNext(yi: Double, h: Double, x: Double): Double

    override fun calculateNext(calculated: List<Double>, h: Double, x: Double) = calculateNext(calculated.last(), h, x)

    override fun checkError(calculated: List<Double>, h: Double, x: Double): Boolean {
        val y1 = calculated.last()
        val y2 = calculateNext(calculated[calculated.size-2], h/2, x)
        val k = (y1-y2).absoluteValue/(2.0.pow(p)-1)
        return k <= eps
    }
}

class ModEulerODUSolveMethod(
    start: Double,
    end: Double,
    y0: Double,
    func: (x: Double, y: Double) -> Double,
    h: Double,
    eps: Double
) : OneStepODUSolveMethod(start, end, y0, func, h, eps) {
    override val p: Double = 2.0
    override val name: String = "Модифицированный метод Эйлера"

    override fun calculateNext(yi: Double, h: Double, x: Double) : Double {
        val f = func(x-h, yi)
        return yi + h/2*(f + func(x, yi + h*f))
    }
}

class RungeKuttaODUSolveMethod(
    start: Double,
    end: Double,
    y0: Double,
    func: (x: Double, y: Double) -> Double,
    h: Double,
    eps: Double
) : OneStepODUSolveMethod(start, end, y0, func, h, eps) {
    override val p: Double = 4.0
    override val name: String = "Метод Рунге-Кутта"

    override fun calculateNext(yi: Double, h: Double, x: Double): Double {
        val xi = x-h
        val k1 = h*func(xi, yi)
        val k2 = h*func(xi+h/2, yi+k1/2)
        val k3 = h*func(xi+h/2, yi+k2/2)
        val k4 = h*func(xi+h, yi+k3)
        val r =  yi + 1.0/6.0*(k1 + 2*k2 + 2*k3 + k4)
        return r
    }
}
```

```kotlin
class AdamsODUSolveMethod(
   start: Double,
   end: Double,
   y0: Double,
   func: (x: Double, y: Double) -> Double,
   h: Double,
   eps: Double,
   private val exactFunc: (x: Double) -> Double
) : ODUSolveMethod(start, end, y0, func, h, eps) {

   private var rungeKuttaSolver = RungeKuttaODUSolveMethod(start, start+h*4,
y0, func, h, eps)
   private val fs = LinkedList<Double>()
   override val name: String = "Метод Адамса"

   override fun calculateNext(calculated: List<Double>, h: Double, x:
Double): Double {
       if (calculated.size < 4) {
           val fi = func(x, rungeKuttaSolver.result[calculated.size])
           fs.add(fi)
           return rungeKuttaSolver.result[calculated.size]
       }
       val dfi = fs[fs.size-1] - fs[fs.size-2]
       val ddfi = fs[fs.size-1] - 2*fs[fs.size-2] + fs[fs.size-3]
       val dddfi = fs[fs.size-1] - 3*fs[fs.size-2] + 3*fs[fs.size-3] -
fs[fs.size-4]
       val yi = calculated.last() + h*fs.last() + h.pow(2)/2*dfi +
5*h.pow(3)/12*ddfi + 3*h.pow(4)/8*dddfi
       fs.removeFirst()
       fs.add(func(x, yi))
       return yi
   }

   override fun onCalculateStart() {
       super.onCalculateStart()
       fs.clear()
       fs.add(func(start, y0))
       rungeKuttaSolver.calculate()
   }

   override fun onMethodReload() {
       super.onMethodReload()
       rungeKuttaSolver = RungeKuttaODUSolveMethod(start, start+h*4, y0,
func, h, eps)
   }

   override fun checkError(calculated: List<Double>, h: Double, x: Double):
Boolean {
       return (calculated.last()-exactFunc(x)).absoluteValue <= eps
   }
}
```

```kotlin
fun main() {
    println("Выберите функцию: \n1: y' = y+(1+x)*y^2 \n2: y' = 3*x^2-y \n3: y' = 10^(x+y)")
    val funcNum = readln()
    println("Введите данные в порядке: <начало интервала> <конец интервала> <шаг> <точность>")
    val (start, end, h, e) = readln().split(" ").map { it.toDoubleOrNull() }
    if (start == null || end == null || h == null || e == null || h < 0 || end < start) {
        println("Некорректные исходные данные")
        return
    }
    print("Введите начальные условия дифференцирование y(x0)=")
    val y0 = readln().toDoubleOrNull()
    if (y0 == null) {
        println("Некорректные исходные данные")
        return
    }

    val f1 = { x: Double, y: Double -> y + (1.0+x)*y.pow(2) }
    val df1 = { x: Double -> -exp(x)/( x*exp(x) + y0+exp(start)/(start*exp(start)) ) }

    val f2 = { x: Double, y: Double -> 3*x.pow(2)-y }
    val df2 = { x: Double -> (y0-3*start.pow(2)+6*start-6)* exp(start) /exp(x) + 3*x.pow(2) - 6*x + 6 }

    val f3 = { x: Double, y: Double -> 10.0.pow(x+y) }
    val c3 = 1.0/exp(y0*ln(10.0)) + 10.0*start
    val df3 = { x: Double -> -ln(c3 - 10.0.pow(x))/ln(10.0) }

    val f = when (funcNum) {
        "1" -> f1
        "2" -> f2
        "3" -> f3
        else -> {
            println("Некорректные исходные данные")
            return
        }
    }
    val df = when (funcNum) {
        "1" -> df1
        "2" -> df2
        "3" -> df3
        else -> {
            println("Некорректные исходные данные")
            return
        }
    }

    val methods = listOf(
        ModEulerODUSolveMethod(start, end, y0, f, h, e),
```

```
        RungeKuttaODUSolveMethod(start, end, y0, f, h, e),
        AdamsODUSolveMethod(start, end, y0, f, h, e, df)
    )


    // 1 1.5 0.1 0.001
    // -2 2 0.1 0.05
    // 0 0.9 0.1 0.05
    methods.mapNotNull {
        try {
            it.calculate()
        } catch (e: IllegalArgumentException) {
            println("${it.name} не смог найти решения задачи!")
            return@mapNotNull null
        }
        println(it.table(df))
        XYChartBuilder()
            .buildDefault(it.name)
            .drawFunction(start, end, it.h, "Точное значение", 1.8f,
Color.BLUE, df)
            .drawConnectedSeries(Series(it.interval(), it.result),
"${it.name} (h=${it.h})", Color.RED)
    }.show()
}
```

Примеры работы программы

Выберите функцию:
1: y' = y+(1+x)*y^2
2: y' = 3*x^2-y
3: y' = 10^(x+y)
1
Введите данные в порядке: <начало интервала> <конец интервала> <шаг> <точность>
1 1.5 0.1 0.001
Введите начальные условия дифференцирования y(x0)=-1

Модифицированный метод Эйлера
----

i	x	y	Модифицированный метод Эйлера
0	1,0000	-1,000	-1,000
1	1,0031	-0,997	-0,997
2	1,0063	-0,994	-0,994
3	1,0094	-0,991	-0,991
4	1,0125	-0,988	-0,988
5	1,0156	-0,985	-0,985
6	1,0188	-0,982	-0,982

7	1,0219	-0,979	-0,979
8	1,0250	-0,976	-0,976
9	1,0281	-0,973	-0,973
10	1,0313	-0,970	-0,970
11	1,0344	-0,967	-0,967
12	1,0375	-0,964	-0,964
13	1,0406	-0,961	-0,961
14	1,0438	-0,958	-0,958
15	1,0469	-0,955	-0,955
16	1,0500	-0,952	-0,952
17	1,0531	-0,950	-0,950
18	1,0563	-0,947	-0,947
19	1,0594	-0,944	-0,944
20	1,0625	-0,941	-0,941
21	1,0656	-0,938	-0,938
22	1,0688	-0,936	-0,936
23	1,0719	-0,933	-0,933
24	1,0750	-0,930	-0,930
25	1,0781	-0,928	-0,928
26	1,0813	-0,925	-0,925
27	1,0844	-0,922	-0,922
28	1,0875	-0,920	-0,920
29	1,0906	-0,917	-0,917
30	1,0938	-0,914	-0,914
31	1,0969	-0,912	-0,912
32	1,1000	-0,909	-0,909
33	1,1031	-0,907	-0,907
34	1,1063	-0,904	-0,904
35	1,1094	-0,901	-0,901
36	1,1125	-0,899	-0,899
37	1,1156	-0,896	-0,896
38	1,1188	-0,894	-0,894
39	1,1219	-0,891	-0,891
40	1,1250	-0,889	-0,889
41	1,1281	-0,886	-0,886
42	1,1313	-0,884	-0,884
43	1,1344	-0,882	-0,882
44	1,1375	-0,879	-0,879
45	1,1406	-0,877	-0,877
46	1,1438	-0,874	-0,874
47	1,1469	-0,872	-0,872
48	1,1500	-0,870	-0,870
49	1,1531	-0,867	-0,867

50	1,1563	-0,865	-0,865
51	1,1594	-0,863	-0,863
52	1,1625	-0,860	-0,860
53	1,1656	-0,858	-0,858
54	1,1688	-0,856	-0,856
55	1,1719	-0,853	-0,853
56	1,1750	-0,851	-0,851
57	1,1781	-0,849	-0,849
58	1,1813	-0,847	-0,847
59	1,1844	-0,844	-0,844
60	1,1875	-0,842	-0,842
61	1,1906	-0,840	-0,840
62	1,1938	-0,838	-0,838
63	1,1969	-0,836	-0,836
64	1,2000	-0,833	-0,833
65	1,2031	-0,831	-0,831
66	1,2063	-0,829	-0,829
67	1,2094	-0,827	-0,827
68	1,2125	-0,825	-0,825
69	1,2156	-0,823	-0,823
70	1,2188	-0,821	-0,821
71	1,2219	-0,818	-0,818
72	1,2250	-0,816	-0,816
73	1,2281	-0,814	-0,814
74	1,2313	-0,812	-0,812
75	1,2344	-0,810	-0,810
76	1,2375	-0,808	-0,808
77	1,2406	-0,806	-0,806
78	1,2438	-0,804	-0,804
79	1,2469	-0,802	-0,802
80	1,2500	-0,800	-0,800
81	1,2531	-0,798	-0,798
82	1,2563	-0,796	-0,796
83	1,2594	-0,794	-0,794
84	1,2625	-0,792	-0,792
85	1,2656	-0,790	-0,790
86	1,2688	-0,788	-0,788
87	1,2719	-0,786	-0,786
88	1,2750	-0,784	-0,784
89	1,2781	-0,782	-0,782
90	1,2813	-0,780	-0,780
91	1,2844	-0,779	-0,779
92	1,2875	-0,777	-0,777

93	1,2906	-0,775	-0,775
94	1,2938	-0,773	-0,773
95	1,2969	-0,771	-0,771
96	1,3000	-0,769	-0,769
97	1,3031	-0,767	-0,767
98	1,3063	-0,766	-0,766
99	1,3094	-0,764	-0,764
100	1,3125	-0,762	-0,762
101	1,3156	-0,760	-0,760
102	1,3188	-0,758	-0,758
103	1,3219	-0,757	-0,757
104	1,3250	-0,755	-0,755
105	1,3281	-0,753	-0,753
106	1,3313	-0,751	-0,751
107	1,3344	-0,749	-0,749
108	1,3375	-0,748	-0,748
109	1,3406	-0,746	-0,746
110	1,3438	-0,744	-0,744
111	1,3469	-0,742	-0,742
112	1,3500	-0,741	-0,741
113	1,3531	-0,739	-0,739
114	1,3563	-0,737	-0,737
115	1,3594	-0,736	-0,736
116	1,3625	-0,734	-0,734
117	1,3656	-0,732	-0,732
118	1,3688	-0,731	-0,731
119	1,3719	-0,729	-0,729
120	1,3750	-0,727	-0,727
121	1,3781	-0,726	-0,726
122	1,3813	-0,724	-0,724
123	1,3844	-0,722	-0,722
124	1,3875	-0,721	-0,721
125	1,3906	-0,719	-0,719
126	1,3938	-0,717	-0,717
127	1,3969	-0,716	-0,716
128	1,4000	-0,714	-0,714
129	1,4031	-0,713	-0,713
130	1,4063	-0,711	-0,711
131	1,4094	-0,710	-0,710
132	1,4125	-0,708	-0,708
133	1,4156	-0,706	-0,706
134	1,4188	-0,705	-0,705
135	1,4219	-0,703	-0,703

136	1,4250	-0,702	-0,702
137	1,4281	-0,700	-0,700
138	1,4313	-0,699	-0,699
139	1,4344	-0,697	-0,697
140	1,4375	-0,696	-0,696
141	1,4406	-0,694	-0,694
142	1,4438	-0,693	-0,693
143	1,4469	-0,691	-0,691
144	1,4500	-0,690	-0,690
145	1,4531	-0,688	-0,688
146	1,4563	-0,687	-0,687
147	1,4594	-0,685	-0,685
148	1,4625	-0,684	-0,684
149	1,4656	-0,682	-0,682
150	1,4688	-0,681	-0,681
151	1,4719	-0,679	-0,679
152	1,4750	-0,678	-0,678
153	1,4781	-0,677	-0,677
154	1,4813	-0,675	-0,675
155	1,4844	-0,674	-0,674
156	1,4875	-0,672	-0,672
157	1,4906	-0,671	-0,671
158	1,4938	-0,669	-0,669
159	1,4969	-0,668	-0,668
160	1,5000	-0,667	-0,667

Метод Рунге-Кутта

----

i	x	y	Метод Рунге-Кутта
0	1,0000	-1,000	-1,000
1	1,0250	-0,976	-0,976
2	1,0500	-0,952	-0,952
3	1,0750	-0,930	-0,930
4	1,1000	-0,909	-0,909
5	1,1250	-0,889	-0,889
6	1,1500	-0,870	-0,870
7	1,1750	-0,851	-0,851
8	1,2000	-0,833	-0,833
9	1,2250	-0,816	-0,816
10	1,2500	-0,800	-0,800
11	1,2750	-0,784	-0,784
12	1,3000	-0,769	-0,769

13	1,3250	-0,755	-0,755
14	1,3500	-0,741	-0,741
15	1,3750	-0,727	-0,727
16	1,4000	-0,714	-0,714
17	1,4250	-0,702	-0,702
18	1,4500	-0,690	-0,690
19	1,4750	-0,678	-0,678
20	1,5000	-0,667	-0,667

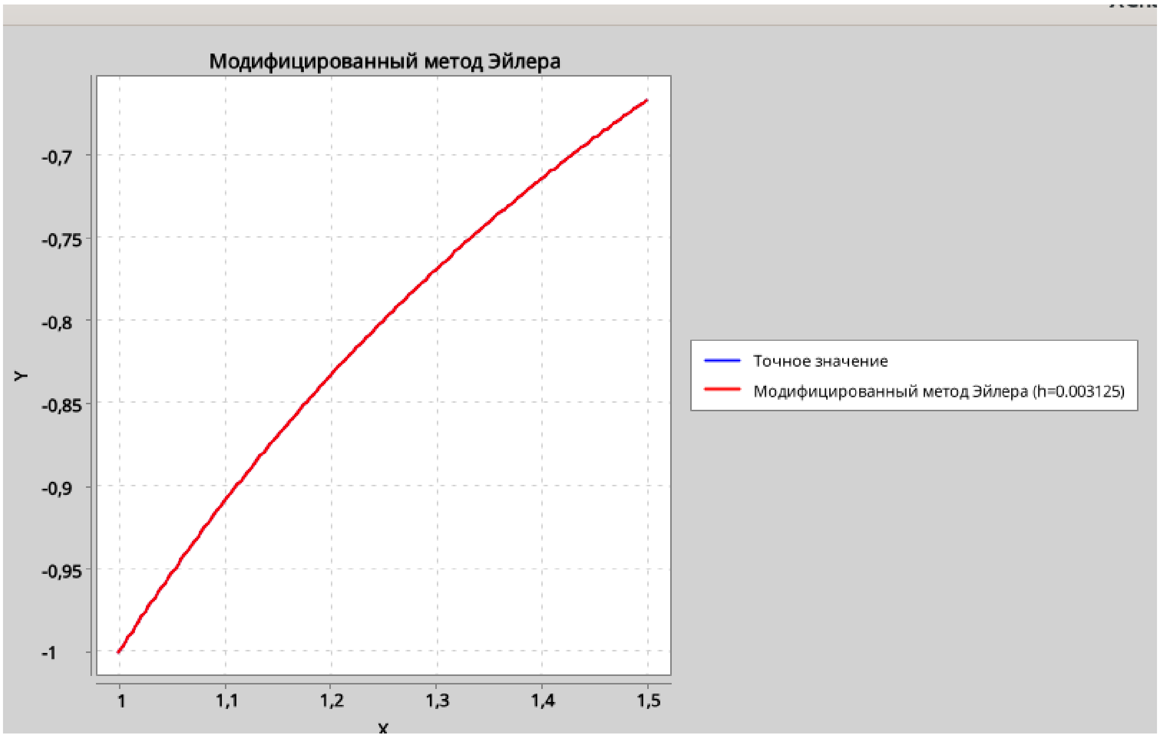
Метод Адамса

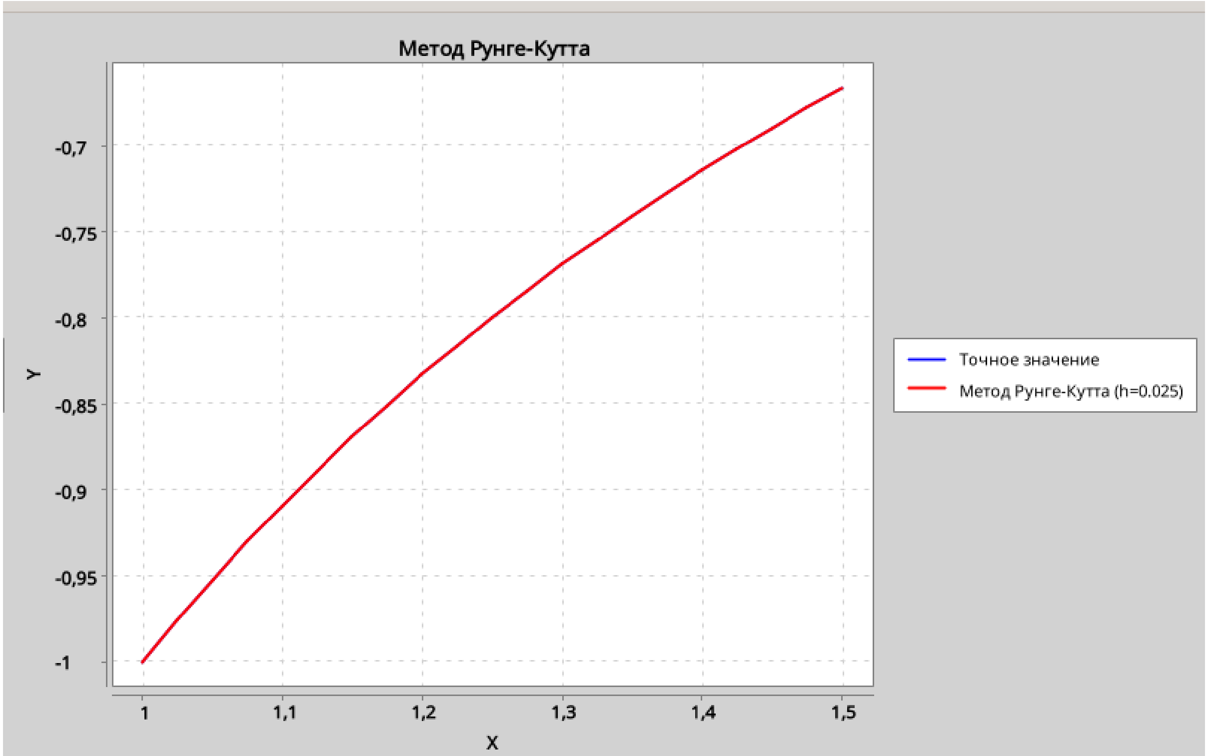
----

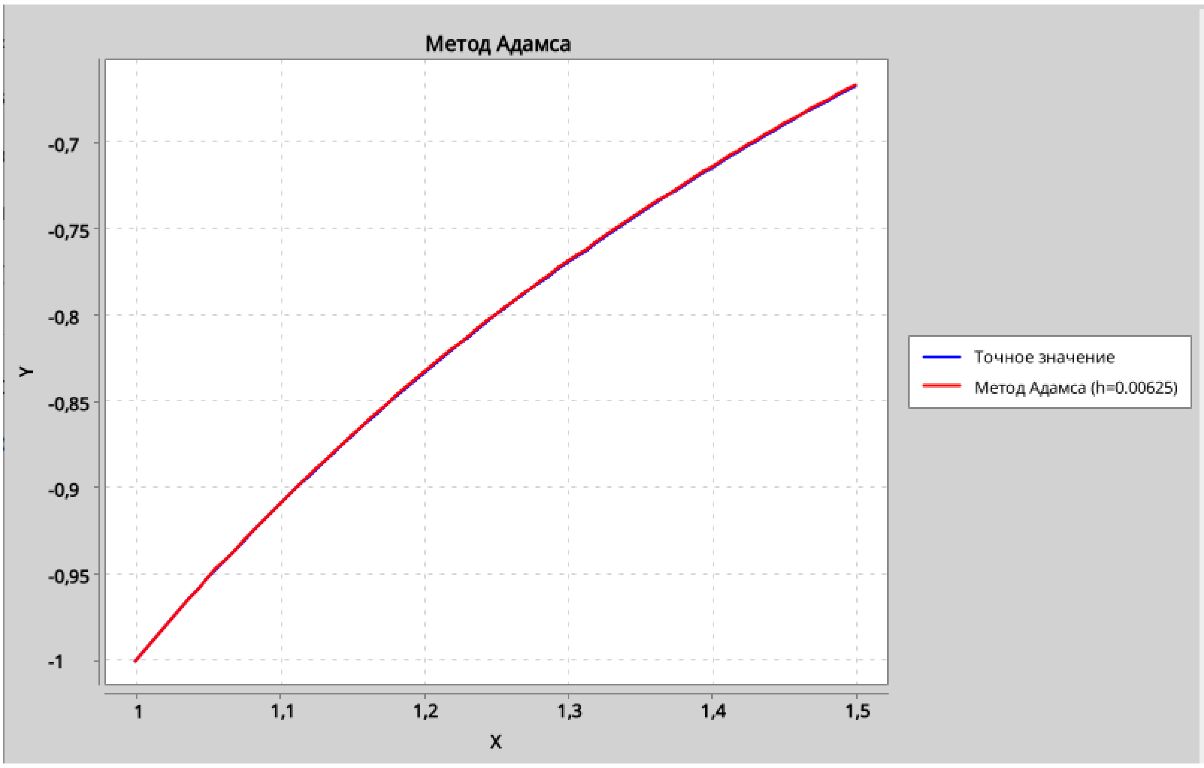
i	x	y	Метод Адамса
0	1,0000	-1,000	-1,000
1	1,0063	-0,994	-0,994
2	1,0125	-0,988	-0,988
3	1,0188	-0,982	-0,982
4	1,0250	-0,976	-0,976
5	1,0313	-0,970	-0,970
6	1,0375	-0,964	-0,964
7	1,0438	-0,958	-0,958
8	1,0500	-0,952	-0,952
9	1,0563	-0,947	-0,947
10	1,0625	-0,941	-0,941
11	1,0688	-0,936	-0,935
12	1,0750	-0,930	-0,930
13	1,0813	-0,925	-0,925
14	1,0875	-0,920	-0,919
15	1,0938	-0,914	-0,914
16	1,1000	-0,909	-0,909
17	1,1063	-0,904	-0,904
18	1,1125	-0,899	-0,898
19	1,1188	-0,894	-0,893
20	1,1250	-0,889	-0,888
21	1,1313	-0,884	-0,883
22	1,1375	-0,879	-0,879
23	1,1438	-0,874	-0,874
24	1,1500	-0,870	-0,869
25	1,1563	-0,865	-0,864
26	1,1625	-0,860	-0,860
27	1,1688	-0,856	-0,855
28	1,1750	-0,851	-0,850
29	1,1813	-0,847	-0,846

30	1,1875	-0,842	-0,841
31	1,1938	-0,838	-0,837
32	1,2000	-0,833	-0,833
33	1,2063	-0,829	-0,828
34	1,2125	-0,825	-0,824
35	1,2188	-0,821	-0,820
36	1,2250	-0,816	-0,816
37	1,2313	-0,812	-0,811
38	1,2375	-0,808	-0,807
39	1,2438	-0,804	-0,803
40	1,2500	-0,800	-0,799
41	1,2563	-0,796	-0,795
42	1,2625	-0,792	-0,791
43	1,2688	-0,788	-0,787
44	1,2750	-0,784	-0,784
45	1,2813	-0,780	-0,780
46	1,2875	-0,777	-0,776
47	1,2938	-0,773	-0,772
48	1,3000	-0,769	-0,768
49	1,3063	-0,766	-0,765
50	1,3125	-0,762	-0,761
51	1,3188	-0,758	-0,757
52	1,3250	-0,755	-0,754
53	1,3313	-0,751	-0,750
54	1,3375	-0,748	-0,747
55	1,3438	-0,744	-0,743
56	1,3500	-0,741	-0,740
57	1,3563	-0,737	-0,737
58	1,3625	-0,734	-0,733
59	1,3688	-0,731	-0,730
60	1,3750	-0,727	-0,726
61	1,3813	-0,724	-0,723
62	1,3875	-0,721	-0,720
63	1,3938	-0,717	-0,717
64	1,4000	-0,714	-0,713
65	1,4063	-0,711	-0,710
66	1,4125	-0,708	-0,707
67	1,4188	-0,705	-0,704
68	1,4250	-0,702	-0,701
69	1,4313	-0,699	-0,698
70	1,4375	-0,696	-0,695
71	1,4438	-0,693	-0,692
72	1,4500	-0,690	-0,689

73	1,4563	-0,687	-0,686
74	1,4625	-0,684	-0,683
75	1,4688	-0,681	-0,680
76	1,4750	-0,678	-0,677
77	1,4813	-0,675	-0,674
78	1,4875	-0,672	-0,671
79	1,4938	-0,669	-0,669




Метод Рунге-Кутта
-0,7
-0,75
-0,8
-0,85
-0,9
-0,95
-1
Y
1
1,1
1,2
1,3
1,4
1,5
X
Точное значение
Метод Рунге-Кутта (h=0.025)


Метод Адамса
-0,7
-0,75
-0,8
-0,85
-0,9
-0,95
-1
Y
1
1,1
1,2
1,3
1,4
1,5
X
Точное значение
Метод Адамса (h=0.00625)

Выберите функцию:
1: y' = y+(1+x)*y^2
2: y' = 3*x^2-y
3: y' = 10^(x+y)
2
Введите данные в порядке: <начало интервала> <конец интервала> <шаг>
<точность>
-2 -1 0.2 0.1
Введите начальные условия дифференцирования y(x0)=-3

Модифицированный метод Эйлера
----
i x y Модифицированный метод Эйлера
0 -2,000 -3,000 -3,000
1 -1,975 -2,633 -2,633
2 -1,950 -2,283 -2,283
3 -1,925 -1,949 -1,949
4 -1,900 -1,630 -1,630
5 -1,875 -1,326 -1,326
6 -1,850 -1,036 -1,036
7 -1,825 -0,760 -0,761
8 -1,800 -0,498 -0,499
9 -1,775 -0,249 -0,250
10 -1,750 -0,013 -0,013
11 -1,725 0,2110 0,2105
12 -1,700 0,4230 0,4225
13 -1,675 0,6235 0,6229
14 -1,650 0,8128 0,8122
15 -1,625 0,9913 0,9907
16 -1,600 1,1594 1,1588
17 -1,575 1,3175 1,3168
18 -1,550 1,4658 1,4651
19 -1,525 1,6047 1,6040
20 -1,500 1,7345 1,7338
21 -1,475 1,8555 1,8548
22 -1,450 1,9682 1,9674
23 -1,425 2,0726 2,0719
24 -1,400 2,1692 2,1685
25 -1,375 2,2582 2,2575
26 -1,350 2,3400 2,3393
27 -1,325 2,4147 2,4140
28 -1,300 2,4827 2,4819
29 -1,275 2,5442 2,5434

30	-1,250	2,5994	2,5987
31	-1,225	2,6487	2,6479
32	-1,200	2,6921	2,6914
33	-1,175	2,7301	2,7294
34	-1,150	2,7628	2,7621
35	-1,125	2,7904	2,7897
36	-1,100	2,8132	2,8125
37	-1,075	2,8313	2,8306
38	-1,050	2,8450	2,8443
39	-1,025	2,8545	2,8538
40	-1,000	2,8600	2,8593

Метод Рунге-Кутта

----

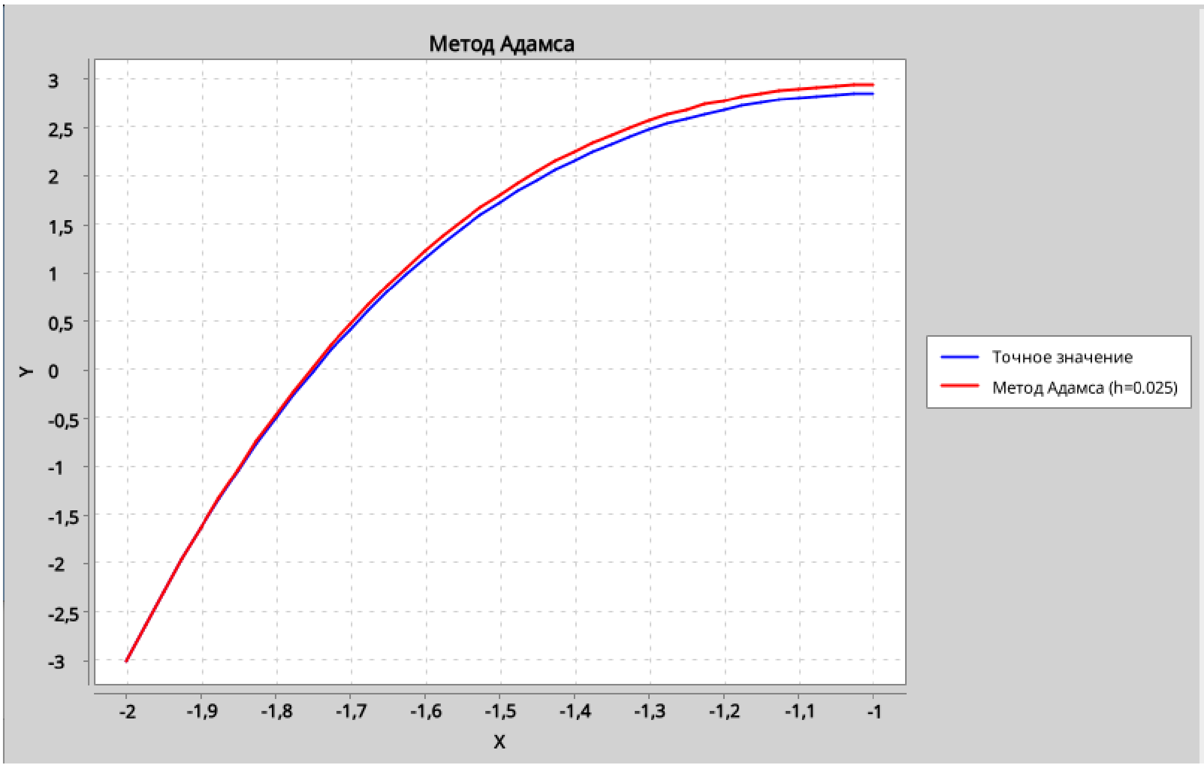
i	x	y	Метод Рунге-Кутта
0	-2,000	-3,000	-3,000
1	-1,800	-0,498	-0,498
2	-1,600	1,1594	1,1593
3	-1,400	2,1692	2,1691
4	-1,200	2,6921	2,6920
5	-1,000	2,8600	2,8599

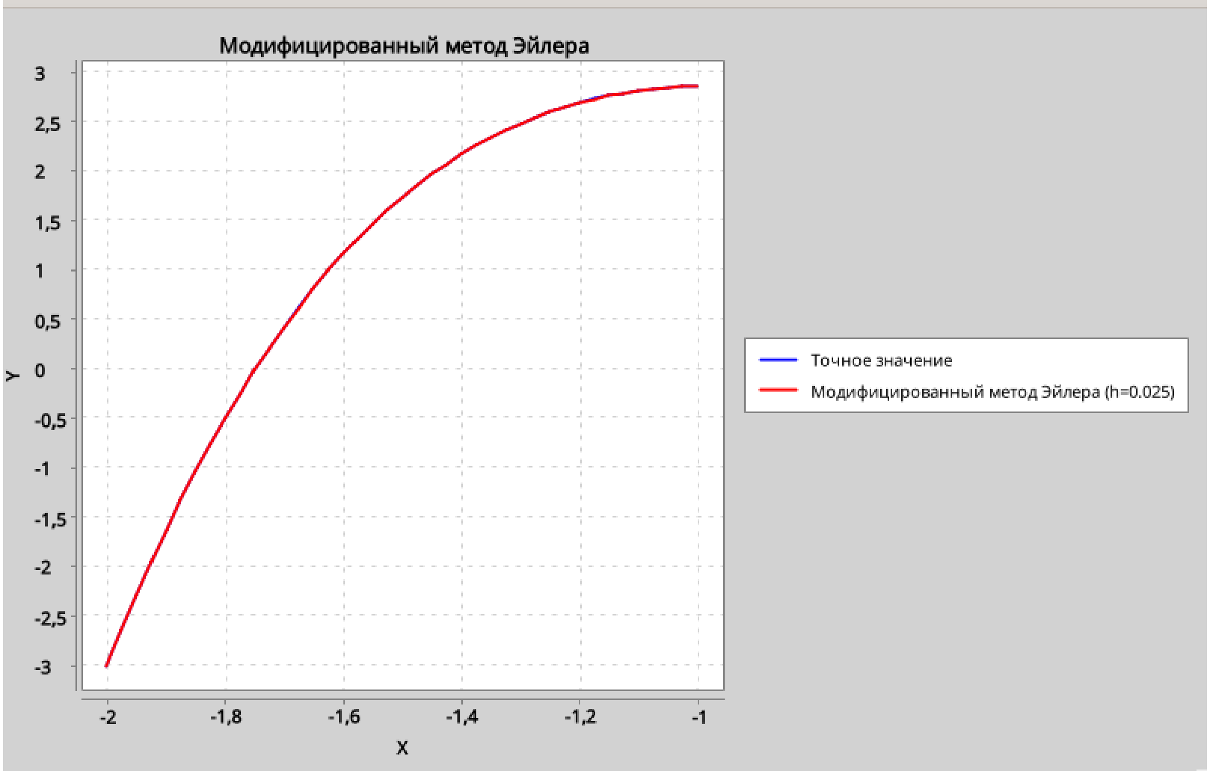
Метод Адамса

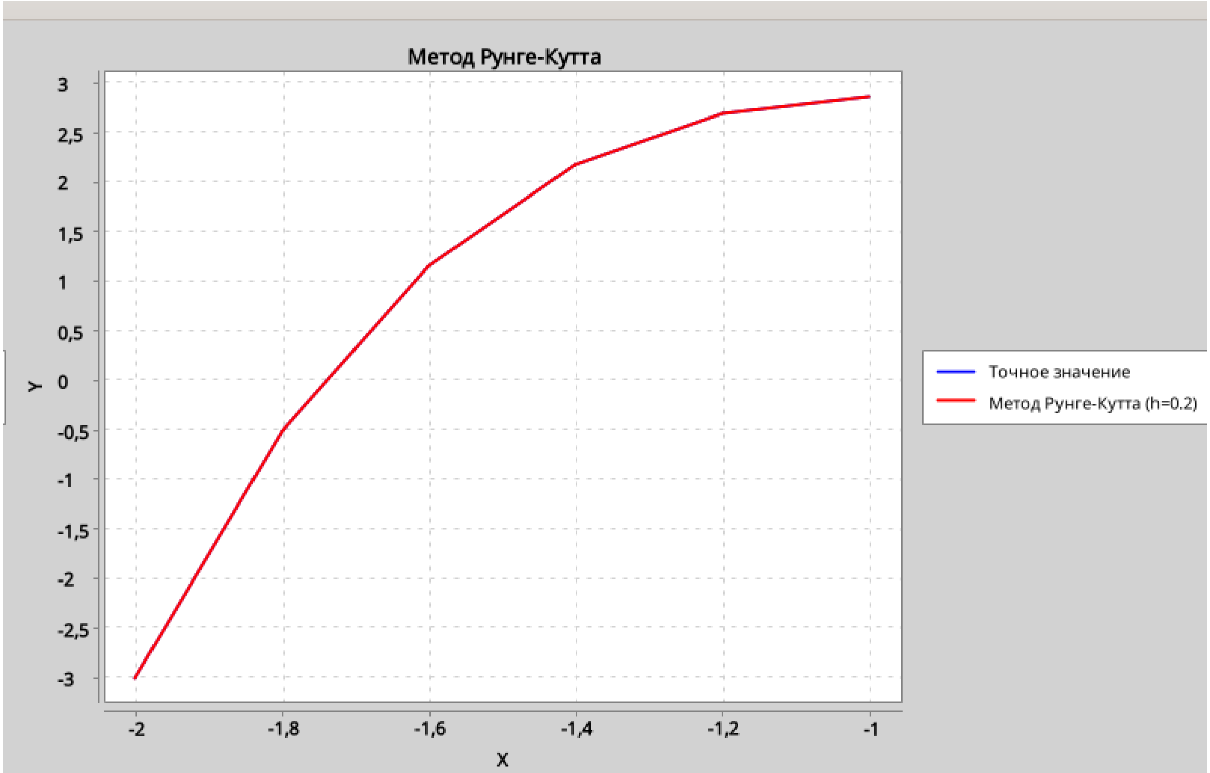
----

i	x	y	Метод Адамса
0	-2,000	-3,000	-3,000
1	-1,975	-2,633	-2,633
2	-1,950	-2,283	-2,283
3	-1,925	-1,949	-1,949
4	-1,900	-1,630	-1,622
5	-1,875	-1,326	-1,311
6	-1,850	-1,036	-1,015
7	-1,825	-0,760	-0,733
8	-1,800	-0,498	-0,465
9	-1,775	-0,249	-0,211
10	-1,750	-0,013	0,0308
11	-1,725	0,2110	0,2596
12	-1,700	0,4230	0,4761
13	-1,675	0,6235	0,6808
14	-1,650	0,8128	0,8741
15	-1,625	0,9913	1,0563

16	-1,600	1,1594	1,2278
17	-1,575	1,3175	1,3890
18	-1,550	1,4658	1,5402
19	-1,525	1,6047	1,6817
20	-1,500	1,7345	1,8140
21	-1,475	1,8555	1,9373
22	-1,450	1,9682	2,0519
23	-1,425	2,0726	2,1582
24	-1,400	2,1692	2,2564
25	-1,375	2,2582	2,3469
26	-1,350	2,3400	2,4300
27	-1,325	2,4147	2,5058
28	-1,300	2,4827	2,5747
29	-1,275	2,5442	2,6370
30	-1,250	2,5994	2,6930
31	-1,225	2,6487	2,7427
32	-1,200	2,6921	2,7867
33	-1,175	2,7301	2,8249
34	-1,150	2,7628	2,8578
35	-1,125	2,7904	2,8855
36	-1,100	2,8132	2,9082
37	-1,075	2,8313	2,9262
38	-1,050	2,8450	2,9396
39	-1,025	2,8545	2,9488
40	-1,000	2,8600	2,9538


Метод Адамса
3
2,5
2
1,5
1
0,5
0
-0,5
-1
-1,5
-2
-2,5
-3
Y
-2
-1,9
-1,8
-1,7
-1,6
-1,5
-1,4
-1,3
-1,2
-1,1
-1
X
Точное значение
Метод Адамса (h=0.025)


Модифицированный метод Эйлера
3
2,5
2
1,5
1
0,5
0
-0,5
-1
-1,5
-2
-2,5
-3
Y
-2
-1,8
-1,6
-1,4
-1,2
-1
X
Точное значение
Модифицированный метод Эйлера (h=0.025)


Метод Рунге-Кутта
3
2,5
2
1,5
1
0,5
0
-0,5
-1
-1,5
-2
-2,5
-3
Y
-2
-1,8
-1,6
-1,4
-1,2
-1
X
Точное значение
Метод Рунге-Кутта (h=0.2)

Выберите функцию:
1: y' = y+(1+x)*y^2
2: y' = 3*x^2-y
3: y' = 10^(x+y)
3
Введите данные в порядке: <начало интервала> <конец интервала> <шаг> <точность>
0 0.9 0.1 0.05
Введите начальные условия дифференцирования y(x0)=-5

Модифицированный метод Эйлера

----

i	x	y	Модифицированный метод Эйлера
0	0,0000	-5,000	-5,000
1	0,1000	-5,000	-5,000
2	0,2000	-5,000	-5,000
3	0,3000	-5,000	-5,000
4	0,4000	-5,000	-5,000
5	0,5000	-5,000	-5,000
6	0,6000	-5,000	-5,000
7	0,7000	-5,000	-5,000
8	0,8000	-5,000	-5,000
9	0,9000	-5,000	-5,000

Метод Рунге-Кутта

----

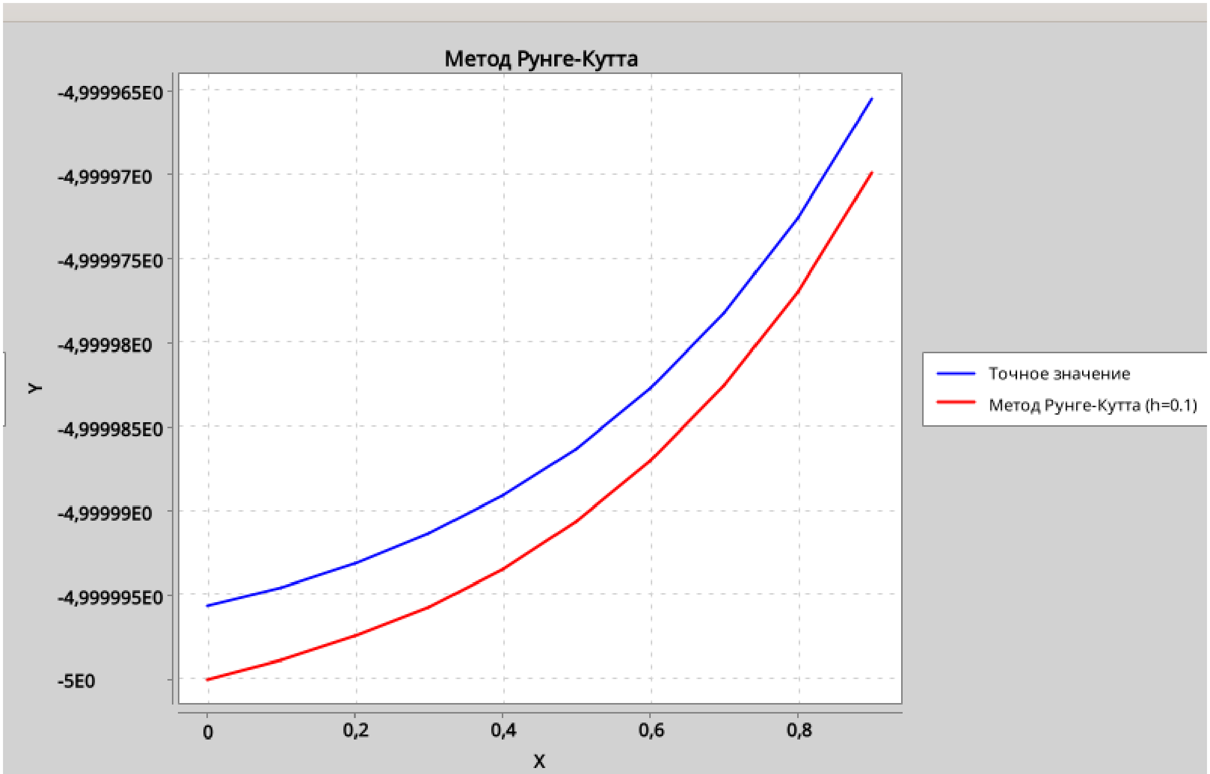
i	x	y	Метод Рунге-Кутта
0	0,0000	-5,000	-5,000
1	0,1000	-5,000	-5,000
2	0,2000	-5,000	-5,000
3	0,3000	-5,000	-5,000
4	0,4000	-5,000	-5,000
5	0,5000	-5,000	-5,000
6	0,6000	-5,000	-5,000
7	0,7000	-5,000	-5,000
8	0,8000	-5,000	-5,000
9	0,9000	-5,000	-5,000

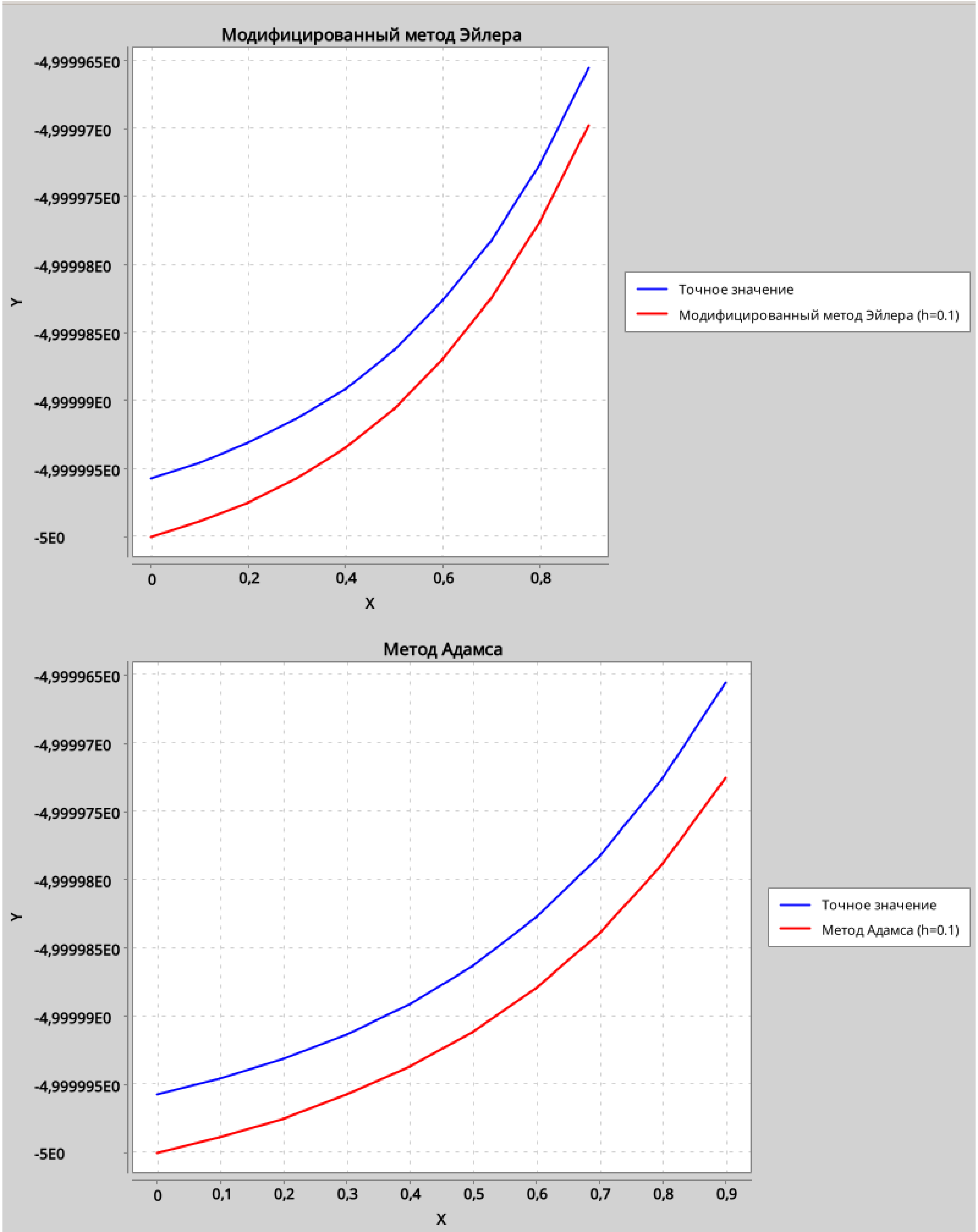
Метод Адамса

----

i x y Метод Адамса

0	0,0000	-5,000	-5,000
1	0,1000	-5,000	-5,000
2	0,2000	-5,000	-5,000
3	0,3000	-5,000	-5,000
4	0,4000	-5,000	-5,000
5	0,5000	-5,000	-5,000
6	0,6000	-5,000	-5,000
7	0,7000	-5,000	-5,000
8	0,8000	-5,000	-5,000
9	0,9000	-5,000	-5,000





Вывод

В ходе лабораторной работы была решена задача Коши для обыкновенных дифференциальных уравнений численными методами. Для этого была написана программа на языке Котлин, поддерживающая 3 метода вычисления: Модифицированный метод Эйлера, Метод Рунге-Кутта и Метод Адамса.