

Университет ИТМО  
Мегафакультет компьютерных технологий и управления  
Факультет программной инженерии и компьютерной техники



ЛАБОРАТОРНАЯ РАБОТА №6  
ЧИСЛЕННОЕ РЕШЕНИЕ ОБЫКНОВЕННЫХ  
ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ  
Вариант №10

Группа: Р3211  
Студент: Орчиков Даниил Валерьевич  
Преподаватель: Малышева Татьяна Алексеевна

г. Санкт-Петербург  
2024

## Оглавление

Цель работы .....	2
Программная реализация задачи .....	2
Листинг программы .....	2
Примеры и результаты работы программы .....	4
Пример 1 .....	4
Пример 2 .....	5
Пример 3 .....	6
Вывод .....	6

## Цель работы

Решить задачу Коши для обыкновенных дифференциальных уравнений численными методами.

## Программная реализация задачи

### Листинг программы

Представлен только код, непосредственно выполняющий вычисления

Весь код можно посмотреть [тут \(GitHub\)](#)

```
function Euler_step(x0, y0, b, h) {
  let xi = x0
  let yi = y0
  let res = []
  res.push([xi, yi])
  for (let i = 0; i < Math.floor((b - x0) / h); i++) {
    yi = yi + h * f(xi, yi)
    xi += h
    res.push([xi, yi])
  }
  return res
}

function Euler(x0, y0, b, epsilon) {
  let v = 0
  let current_h = h
  let first_step = Euler_step(x0, y0, b, current_h)
  current_h /= 2
  let second_step = Euler_step(x0, y0, b, current_h)
  while (Math.abs(first_step[first_step.length - 1][1] - second_step[second_step.length - 1][1]) / (2 - 1) > epsilon && v < 10) {
    first_step = second_step
    current_h /= 2
    second_step = Euler_step(x0, y0, b, current_h)
    v++
  }
  console.log("Эйлер: " + current_h)
  return second_step
}

function Runge_Kutta_step(x0, y0, b, h) {
  let xi = x0
  let yi = y0
  let res = []
}
```

```

res.push([xi, yi])
for (let i = 0; i < Math.floor((b - x0) / h + 0.1); i++) {
    let k1 = h * f(xi, yi)
    let k2 = h * f(xi + h / 2, yi + k1 / 2)
    let k3 = h * f(xi + h / 2, yi + k2 / 2)
    let k4 = h * f(xi + h, yi + k3)
    yi = yi + 1 / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
    xi += h
    res.push([xi, yi])
}
return res
}

function Runge_Kutta(x0, y0, b, epsilon){
    let v = 0
    let current_h = h
    let first_step = Runge_Kutta_step(x0, y0, b, current_h)
    current_h /= 2
    let second_step = Runge_Kutta_step(x0, y0, b, current_h)
    while (Math.abs(first_step[first_step.length - 1][1] - second_step[second_step.length - 1][1]) / (2 ** 4 - 1) > epsilon && v < 10) {
        first_step = second_step
        current_h /= 2
        second_step = Runge_Kutta_step(x0, y0, b, current_h)
        v++
    }
    console.log("Рунге-Кутты: " + current_h)
    return second_step
}

function Milne_step(x0, y0, b, epsilon, h) {
    let res = Runge_Kutta_step(x0, y0, x0 + h * 4, h).slice(0, 4)
    let xi = x0 + h * 4
    for (let i = 3; i < Math.floor((b - x0) / h); i++) {
        let yi_p = res[i - 3][1] + 4 * h / 3 * (2 * f(res[i - 2][0], res[i - 2][1]) - f(res[i - 1][0], res[i - 1][1]) + 2 * f(res[i][0], res[i][1]))
        let yi_c = res[i - 1][1] + h / 3 * (f(res[i - 1][0], res[i - 1][1]) + 4 * f(res[i][0], res[i][1]) + f(xi, yi_p))
        while (Math.abs(yi_p - yi_c) > epsilon) {
            yi_p = yi_c
            yi_c = res[i - 1][1] + h / 3 * (f(res[i - 1][0], res[i - 1][1]) + 4 * f(res[i][0], res[i][1]) + f(xi, yi_p))
        }
        res.push([xi, yi_c])
        xi += h
    }
    return res
}

function Milne(x0, y0, b, epsilon){
    let v = 0
    let current_h = h
    let first_step = Milne_step(x0, y0, b, epsilon, current_h)
    current_h /= 2
    let second_step = Milne_step(x0, y0, b, epsilon, current_h)
    while (Math.max.apply(null, first_step.map((x,i)=>{Math.abs(x - second_step[i])})) > epsilon && v < 10) {
        first_step = second_step
        current_h /= 2
        second_step = Milne_step(x0, y0, b, epsilon, current_h)
        v++
    }
    console.log("Милн: " + current_h)
    return second_step
}

```

# Примеры и результаты работы программы

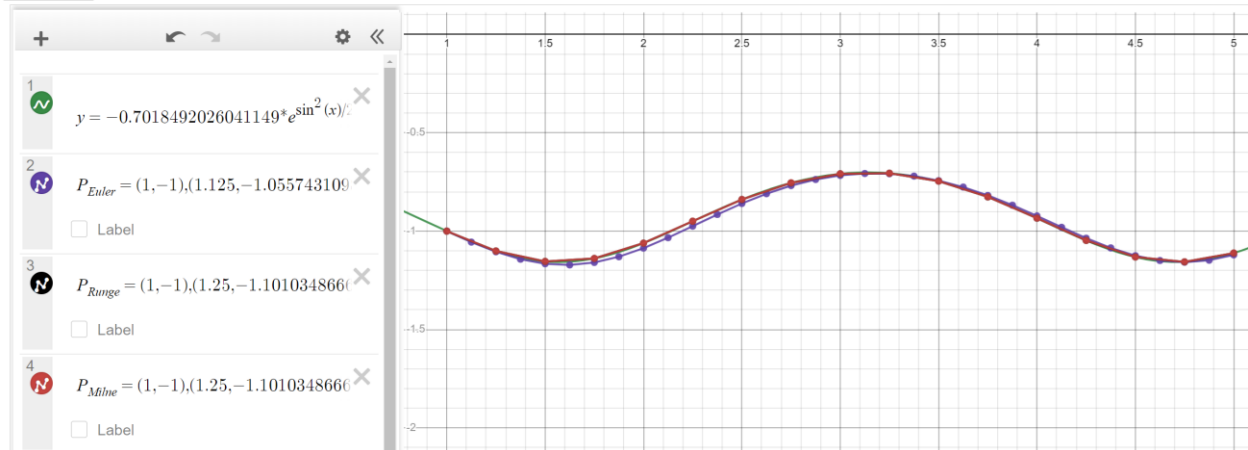
## Пример 1

Выберете функцию

$y' = \cos(x) * \sin(x) * y$

Диапазон:  $x_0 = 1$ ,  $x_n = 5$   $y(x_0) = -1$  Шаг = 0,5  $\varepsilon = 0,01$

Рассчитать



Метод Эйлера:

xi	yi
1	-1
1.125	-1.0557431098920635
1.25	-1.1048557855706764
1.375	-1.1428914771592407
1.5	-1.1660439255895054
1.625	-1.1717934356848083
1.75	-1.1593904811424514
1.875	-1.1300257686426514
2	-1.0866253735579592
2.125	-1.0333295600702388
2.25	-0.9748101718139627
2.375	-0.9156122992485468
2.5	-0.8596659618069232
2.625	-0.8100323500736468
2.75	-0.7688684936846164
2.875	-0.7375434427853994
3	-0.7168249600367083
3.125	-0.7070679318084788
3.25	-0.7083585841690455
3.375	-0.7205904029756184
3.5	-0.7434636958572836
3.625	-0.776411978984071
3.75	-0.8184685739776134
3.875	-0.8680996237806602
4	-0.9230463217353626
4.125	-0.9802361353635891
4.25	-1.035831537567445
4.375	-1.085473266042
4.5	-1.1247342372732372
4.625	-1.1497325012552582
4.75	-1.157776565523632
4.875	-1.147879707581861
5	-1.1209640973771615

Метод Рунге-Кутты:

xi	yi
1	-1
1.25	-1.1010348666899117
1.5	-1.154264766925966
1.75	-1.1389187116566863
2	-1.0611721222957575
2.25	-0.9499546071121946
2.5	-0.8394913763813534
2.75	-0.75486674194266
3	-0.708864490626539
3.25	-0.7059609714135981
3.5	-0.7463780472329572
3.75	-0.8263763829609541
4	-0.9345691723454246
4.25	-1.0475559317538323
4.5	-1.131722186868716
4.75	-1.1563302722971205
5	-1.1115164618753686

Метод Милна:

xi	yi
1	-1
1.25	-1.1010348666899117
1.5	-1.154264766925966
1.75	-1.1389187116566863
2	-1.0610620341692327
2.25	-0.949569753660152
2.5	-0.8391147614804041
2.75	-0.7545098770704523
3	-0.7086185204561755
3.25	-0.705640207359511
3.5	-0.7460694093283544
3.75	-0.8259641761442686
4	-0.9341941162864067
4.25	-1.0471544743760457
4.5	-1.1314478032086273
4.75	-1.1561073332678937
5	-1.1113205686759244

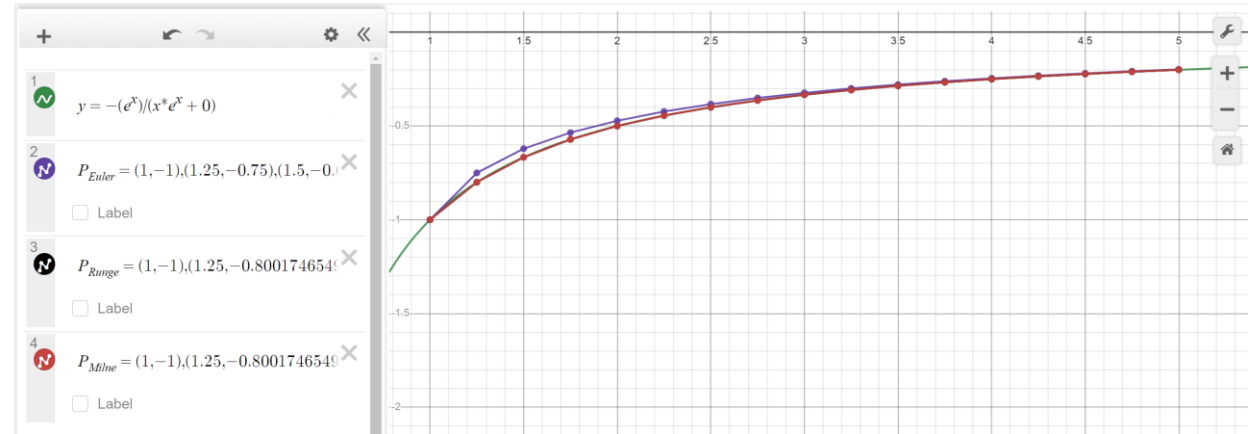
# Пример 2

Выберите функцию

$y' = (y + (1 + x)) * y^2$

Диапазон:  $x_0 = 1$ ,  $x_n = 5$ ,  $y(x_0) = -1$  Шаг = 0,5  $\epsilon = 0,01$

Рассчитать



Метод Эйлера:

xi	yi
1	-1
1.25	-0.75
1.5	-0.62109375
1.75	-0.5352687835693359
2	-0.47210851838030976
2.25	-0.42297080812994875
2.5	-0.3833537627317281
2.75	-0.3506021094391578
3	-0.3230134126022093
3.25	-0.2994291010318365
3.5	-0.279024978086016
3.75	-0.2611944169121306
4	-0.2454787745717076
4.25	-0.23152368225810047
4.5	-0.21905038254929138
4.75	-0.20783625680600173
5	-0.19770120089550236

Метод Рунге-Кутты:

xi	yi
1	-1
1.25	-0.8001746549505497
1.5	-0.6668332608334915
1.75	-0.571558745119387
2	-0.5000964957914322
2.25	-0.4445149618473122
2.5	-0.40005150584690435
2.75	-0.36367417814725383
3	-0.3333613104278341
3.25	-0.3077131907703189
3.5	-0.28573002061814945
3.75	-0.2666786370500872
4	-0.250009195101545
4.25	-0.23530124953570242
4.5	-0.22222780726623612
4.75	-0.21053073131634423
5	-0.20000352386632891

Метод Милна:

xi	yi
1	-1
1.25	-0.8001746549505497
1.5	-0.6668332608334915
1.75	-0.571558745119387
2	-0.4994184689272405
2.25	-0.4445670559116672
2.5	-0.399418125601427
2.75	-0.3637964373200119
3	-0.33289722320705767
3.25	-0.307868885617143
3.5	-0.28539207573577435
3.75	-0.26684523564656126
4	-0.24974741938333492
4.25	-0.23546865263903083
4.5	-0.22201053805009655
4.75	-0.2106948398225958
5	-0.19981299042031475

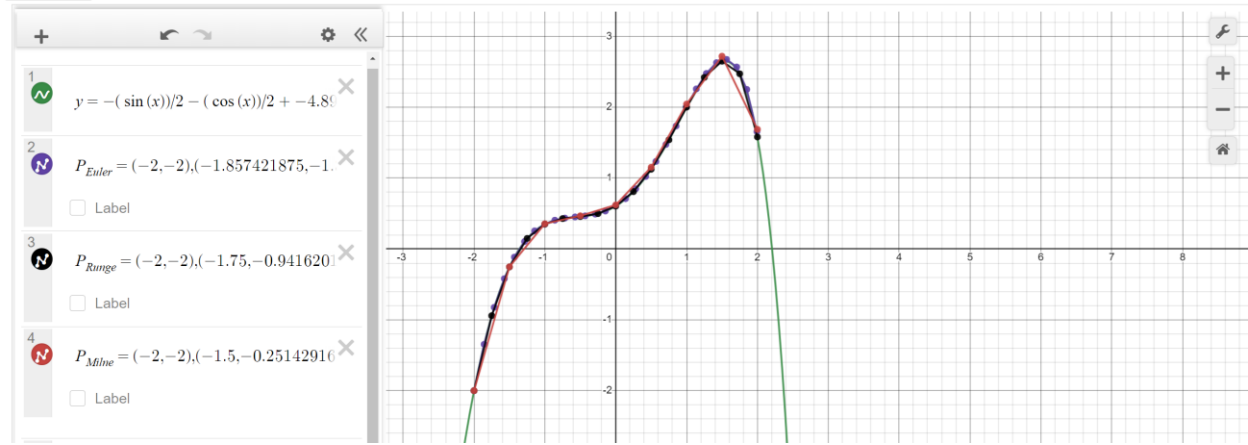
# Пример 3

Выберете функцию

y' =

Диапазон: x0 =  , x1 =  y(x0) =  Шаг =  ε =

Рассчитать



Метод Эйлера:

xi	yi
-2	-2
-1.8574	-1.3456785050302513
-1.7148	-0.8240925376905547
-1.5723	-0.4191316346194909
-1.4297	-0.11481704956713476
-1.2871	0.10461519951473382
-1.1445	0.254618529716671
-1.002	0.35022656131281354
-0.8594	0.40592964829314826
-0.7168	0.4355361905461227
-0.5742	0.4520175216600284
-0.4316	0.46733488366507553
-0.2891	0.4922466532353303
-0.1465	0.5360935654912963
-0.0039	0.6065591834082114
0.1387	0.7094022738246603
0.2813	0.8481570649016872
0.4238	1.0237965629598171
0.5664	1.234353185606858
0.709	1.4744899076507518
0.8516	1.7350138987759114
0.9941	2.002323236836587
1.1367	2.257775684050664
1.2793	2.476966687625242
1.4219	2.628901679112005
1.5645	2.6750453604583537
1.707	2.5682279354892787
1.8496	2.2513851224540655
1.9922	1.6561052070289783

Метод Рунге-Кутты:

xi	yi
-2	-2
-1.75	-0.9416201802559607
-1.5	-0.25407154136680377
-1.25	0.14853509535525072
-1	0.34956855407288623
-0.75	0.42808622378246036
-0.5	0.45661902553361666
-0.25	0.49847360317788836
0	0.6044671080971288
0.25	0.808973550630578
0.5	1.1251114607978285
0.75	1.5388378174823831
1	2.0016294724724624
1.25	2.4213262872797907
1.5	2.650574329888632
1.75	2.472135477133803
2	1.5801121506452989

Метод Милна:

xi	yi
-2	-2
-1.5	-0.2514291627094183
-1	0.35574233496550656
-0.5	0.4678230780187131
0	0.621092377128439
0.5	1.15279744897895
1	2.0452381984672363
1.5	2.7190756361887805
2	1.687399217413716

## Вывод

Во время выполнения данной лабораторной работы я познакомился с различными численными методами решения задачи Коши для обыкновенных дифф. уравнений и написал реализацию некоторых из них на языке JavaScript.