**Чернівецький національний університет імені Юрія Федьковича**

**Факультет математики та інформатики**

**Кафедра математичного моделювання**

**Лабораторна робота №1**

з навчальної дисципліни: “Комп'ютерне моделювання жорстких процесів та систем”

Виконав: студент 3 курсу

311-б групи

спеціальності “Комп’ютерні науки”

Вайнагій Данило Вікторович.

Перевірив: ac. Іліка Світлана Анатоліївна

Варіант: 4

**Чернівці – 2023**

Завдання:  
Використовуючи задану багатокрокову різницеву схему, знайти розв’язок модельної задачі Коші. Вхід у таблицю підрахувати за явною одно кроковою різницевою схемою, у випадку неявної різницевої схеми розв’язок уточнювати методом простої ітерації або методом Ньютона.

// НА3-головна, ЯРК3-допоміжна

let epsilon, h\_0, h\_optimal;

const x\_0 = 1, y\_0 = 1, t\_0 = 0;

epsilon = 0.001

h\_0 = 0.1

h\_optimal = find\_h\_optimal(h\_0, epsilon)

console.log(h\_optimal)

function f(x, y, t) {

    return t / y

}

function g(x, y, t) {

    return -t / x

}

function find\_h\_optimal(h\_0, epsilon, x\_1, y\_1) {

    if (x\_1 == null && y\_1 == null) {

        x\_1 = x\_0 + h\_0 \* (f(x\_0, y\_0, t\_0))

        y\_1 = y\_0 + h\_0 \* (g(x\_0, y\_0, t\_0))

    }

    x\_avg = x\_0 + h\_0 / 2 \* (f(x\_0, y\_0, t\_0))

    y\_avg = y\_0 + h\_0 / 2 \* (g(x\_0, y\_0, t\_0))

    x\_1\_new = x\_avg + h\_0 / 2 \* (f(x\_avg, y\_avg, t\_0 + h\_0 / 2))

    y\_1\_new = y\_avg + h\_0 / 2 \* (g(x\_avg, y\_avg, t\_0 + h\_0 / 2))

    if (Math.abs(x\_1 - x\_1\_new) + Math.abs(y\_1 - y\_1\_new) > epsilon) {

        console.log("Not In range")

        h\_optimal = find\_h\_optimal(h\_0 / 2, epsilon, x\_avg, y\_avg);

        return h\_optimal

    }

    else {

        console.log("In range")

        h\_optimal = h\_0 / 2

        return h\_optimal;

    }

}

const arrayRange = (start, stop, step) =>

    Array.from(

        { length: (stop - start) / step + 1 },

        (value, index) => parseFloat((start + index \* step).toFixed(5))

    );

t = arrayRange(0, 1, h\_optimal)

//Вираховуєм точні значення

let x\_accurate = calculate\_x\_accurate(t)

function calculate\_x\_accurate(t) {

    let x\_accurate = []

    for (i in t) {

        x\_accurate[i] = Math.exp(t[i] \*\* 2 / 2)

    }

    return x\_accurate

}

let y\_accurate = calculate\_y\_accurate(t)

function calculate\_y\_accurate(t) {

    let y\_accurate = []

    for (i in t) {

        y\_accurate[i] = Math.exp(-(t[i] \*\* 2 / 2))

    }

    return y\_accurate

}

let x = [], y = [];

x[0] = x\_0;

y[0] = y\_0;

function RK3\_x(t, x, y, n) {

    function K1(n) {

        return f(x[n], y[n], t[n])

    }

    function K2(n) {

        return f(x[n], y[n] + (h\_optimal / 3) \* K1(n), t[n] + h\_optimal / 3)

    }

    function K3(n) {

        return f(x[n], y[n] + (h\_optimal \* 2 / 3) \* K2(n), t[n] + h\_optimal \* 2 / 3)

    }

    return x[n - 1] + (h\_optimal / 4) \* (K1(n - 1) + 3 \* K3(n - 1))

}

function RK3\_y(t, x, y, n) {

    function K1(n) {

        return g(x[n], y[n], t[n])

    }

    function K2(n) {

        return g(x[n] + (h\_optimal / 3) \* K1(n), y[n], t[n] + h\_optimal / 3)

    }

    function K3(n) {

        return g(x[n] + (h\_optimal \* 2 / 3) \* K2(n), y[n], t[n] + h\_optimal \* 2 / 3)

    }

    return y[n - 1] + (h\_optimal / 4) \* (K1(n - 1) + 3 \* K3(n - 1))

}

x[1] = RK3\_x(t, x, y, 1)

y[1] = RK3\_y(t, x, y, 1)

console.log(y[1])

function NA3\_x(t, x, y, n) {

    x\_n0=x[n-1]+f(x[n-1],y[n-1],t[n-1])

    y\_n0=y[n-1]+g(x[n-1],y[n-1],t[n-1])

    do{

    x\_n1=x[n - 1] + (h\_optimal / 12) \* (5 \* f(x\_n0, y\_n0, t[n]) + 8 \* f(x[n - 1], y[n - 1], t[n - 1]) - f(x[n - 2], y[n - 2], t[n - 2]))

    y\_n1=y[n - 1] + (h\_optimal / 12) \* (5 \* g(x\_n0, y\_n0, t[n]) + 8 \* g(x[n - 1], y[n - 1], t[n - 1]) - g(x[n - 2], y[n - 2], t[n - 2]))

    if((Math.abs(x\_n1-x\_n0)+Math.abs(y\_n1-y\_n0))<epsilon){

       return [x\_n1,y\_n1]

    }

    else{

         x\_n0=x\_n1;

        y\_n0=y\_n1;

        //console.log(n+'iter')

    }

    }

    while(true)

}

for(i=2;t[i]<=1;i++){

    xy\_i=NA3\_x(t,x,y,i);

    x[i]=xy\_i[0];

    y[i]=xy\_i[1];

}

let delta\_x=[],delta\_y=[];

delta\_x[0]=Math.abs(x\_accurate[0]-x[0])

delta\_y[0]=Math.abs(y\_accurate[0]-y[0])

for(i=1;i<x.length;i++){

    delta\_x[i]=Math.abs(x\_accurate[i]-x[i]).toFixed(15);

    delta\_y[i]=Math.abs(y\_accurate[i]-y[i]).toFixed(15);

}

dataArray=[t,x\_accurate,x,delta\_x,y\_accurate,y,delta\_y]

function createTable(tableData) {

    let table = document.createElement('table');

    let tableBody = document.createElement('tbody');

        let row = document.createElement('tr');

        THs=['t','x точне','x','delta x','y точне','y','delta y']

        THs.forEach(function(rowData) {

          let cell = document.createElement('th');

          cell.appendChild(document.createTextNode(rowData));

          row.appendChild(cell);

        });

        tableBody.appendChild(row);

      ;

    tableData.forEach(function(cellData) {

      let row = document.createElement('td');

      cellData.forEach(function(rowData) {

        let cell = document.createElement('tr');

        cell.appendChild(document.createTextNode(rowData));

        row.appendChild(cell);

      });

      tableBody.appendChild(row);

    });

    table.appendChild(tableBody);

    document.body.appendChild(table);

  }

  createTable(dataArray);

Точність 0,01

