МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

Лабораторная работа №6

«Приближенное решение обыкновенных дифференциальных уравнений методом Эйлера с пересчетом»

по дисциплине

Вычислительная математика

РУКОВОДИТЕЛЬ:

________________ Суркова А.С.

СТУДЕНТ:

________________ Сухоруков В.А.

19-ИВТ-3

Работа защищена «___» ____________

С оценкой ________________________

Нижний Новгород 2021

Оглавление

Элементы оглавления не найдены.

Цель работы

Закрепление знаний и умений по численному решению обыкновенных дифференциальных уравнений методом Эйлера и методом Адамса.

Постановка задачи

Задание 1

Используя метод Эйлера и метод Эйлера с пересчетом, составить таблицу приближенных значений интеграла дифференциального уравнения $y'=f(x,y)$, удовлетворяющего начальным условиям $y(x0)=y0$ на отрезке [a,b]; шаг h=0.1. Все вычисления вести с четырехзначными знаками. Проверить полученные значения, используя метод Рунге-Кутты 4 порядка.

$$19.\ y' = x + \sin\frac{y}{\sqrt{3}},\quad y_0(1.1) = 1.5,\quad x \in [1.1;2.1]$$

Задание 2

Используя метод Адамса с третьими разностями составить таблицу приближенных значений интеграла дифференциального уравнения $y'=f(x,y)$, удовлетворяющего начальным условиям $y(x0)=y0$ на отрезке [0,1]; шаг h=0.1. Все вычисления вести с четырехзначными знаками. Начальный отрезок определить методом Рунге-Кутты. Проверить полученные значения, используя метод Эйлера с пересчетом.

$$19.\ y'=(0.8-y^2)\cos x+0.3y,\quad y(0)=0$$

Теоретические сведения

Постановка задачи Коши:

Найти решение дифференциального уравнения

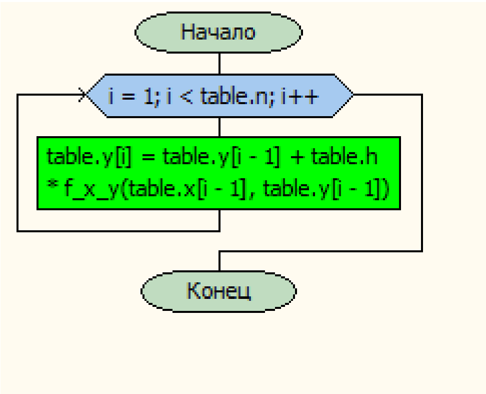
$y'=f(x,y)$, удовлетворяющее начальному условию $y(x_0)=y_0$. При численном решении уравнения задача ставится так: в точках x_0, x_1, x_2, ..., x_n найти приближения y_k ($k = 0,1,2,..., n$) для значений точного решения $y(x_k)$. Разность $\Delta x_k = x_{k+1} - x_k$ во многих случаях принимают постоянной h и называют шагом сетки, тогда $x_k = x_0 + kh$, $k=0,1,\ldots,n$.

Метод Эйлера

Метод Эйлера для решения указанной задачи Коши основан на непосредственной замене производной разностным отношением по приближенной формуле: $dy/dx=f(x,y)$, $\Delta y/ \Delta x=f(x,y)$ если обозначить $h= \Delta x$, то:

$$y(x+h)=y(x)+hf(x,y)$$

Приближенные значения y_k в точках $x_K = x_0 + hk$ вычисляются по формуле $y_{k+1} = y_k + hf(x_k, y_k)$



Нахождение y' методом Эйлера

Метод Адамса

Широко распространенным семейством многошаговых методов решения дифференциальных уравнений являются методы Адамса. В практических расчетах чаще всего используется вариант метода Адамса, имеющий четвертый порядок точности и использующий на каждом шаге результаты предыдущих четырех. Именно его и называют обычно методом Адамса.

Рассмотрим этот метод. Пусть найдены значения y_{i-3}, y_{i-2}, y_{i-1}, y_i - в четырех последовательных узлах и значения правой части - f_{i-3}, f_{i-2}, f_{i-1}, f_i, где $f_i = f(x_i, y_i)$. В качестве интерполяционного многочлена $P_3(x)$ можно взять многочлен Ньютона. В случае постоянного шага конечные разности для правой части в узле имеют вид:

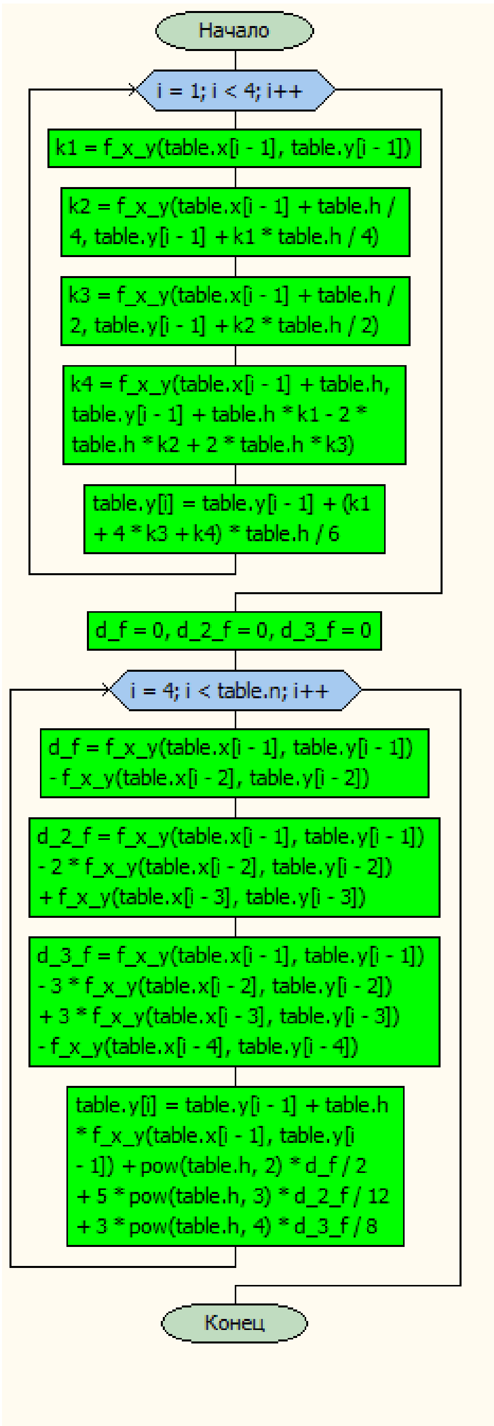
$$\Delta f_i = f_i - f_{i-1}$$

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

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

Тогда разностную схему четвертого порядка метода Адамса можно записать после необходимых преобразований в виде :

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



Нахождение y’ методом Адамса. Для нахождения первых 3 значений используется метод Рунге-Кутта

Расчетные данные

Задание 1

19. $y' = x + \sin\frac{y}{\sqrt{3}}$, $y_0(1.1) = 1.5$, $x \in [1.1;2.1]$

Решение методом Эйлера

f(xi)

xi	yi
1.1	1.5
1.2	1.6862
1.3	1.8889
1.4	2.1076
1.5	2.3413
1.6	2.589
1.7	2.8487
1.8	3.1184
1.9	3.3958
2	3.6783

Решение методом Эйлера с пересчётом

f(xi)

xi	yi
1.1	1.5
1.2	1.6894
1.3	1.8952
1.4	2.1166
1.5	2.3525
1.6	2.6012
1.7	2.8609
1.8	3.1294
1.9	3.4041
2	3.6827

Решение методом Рунге-Кутта

f(xi)

xi	yi
1.1	1.5
1.2	1.6946
1.3	1.9057
1.4	2.1325
1.5	2.3738
1.6	2.6279
1.7	2.8928
1.8	3.1661
1.9	3.4454
2	3.728

Задание 2

19. $y' = (0.8 - y^2)\cos x + 0.3y$, $y(0) = 0$

Решение методом Адамса

xi	f(xi) yi
0	0
0.1	0.08086
0.2	0.16206
0.3	0.24161
0.4	0.31957
0.5	0.39322
0.6	0.46137
0.7	0.52336
0.8	0.57898
0.9	0.62842
1	0.67217

Решение методом Эйлера с пересчётом

xi	f(xi) yi
0	0
0.1	0.08088
0.2	0.16249
0.3	0.24279
0.4	0.31994
0.5	0.39247
0.6	0.4594
0.7	0.5202
0.8	0.57479
0.9	0.62341
1	0.66657

Код программы

Value_function_table.h

```
#pragma once
#include<vector>
#include<iostream>
#include"Colors.h"
using namespace std;

/*Класс для описания таблицы значений функции*/
class Value_function_table{
public:
	vector<double>x;			//Координаты x точек
	vector<double>y;			//Координаты y точек
	size_t n;				//Количество точек

	double a;				//Левая граница для x
	double b;				//Правая граница для x
```

```
	double h;
			//Шаг

	Value_function_table() {
		n = 0;
		a = 0;
		b = 0;
		h = 0;
	}

	//Функция заполнения таблицы
	void set_value() {
		setlocale(LC_ALL, "Russian");
		cout<<Yellow << "X:[a,b]\nВведите a ";
		cin >> this->a;
		cout << "\nВведите b ";
		cin >> this->b;
		cout << "\nВведите шаг h ";
		cin >> this->h;
		cout << "\nВведите y0 ";
		double y0; cin >> y0;
		this->y.push_back(y0);


		for (double i = a; i <=b; i=i+h){
			this->x.push_back(i);
			this->y.push_back(0);
		}
		this->n = this->x.size();
	}

};
```

solution methods.h

```
#pragma once
#include<cmath>
#include<vector>
#include<iostream>
#include<iomanip>
#include"Value_function_table.h"

/*Библиотека для решения  дифференциального уравнения методом
Эйлера и Рунге-Кутта*/


/*Нахождение значения функции при заданных x и y
 *Параметр equation отвечает за выбор функции
 *Если equation=1, то находится значение функции
f(x,y)=x+sin(y/3^(1/2))
 *Если equation=2, то находится значение функции f(x,y)=(0.8-
y^2)*cos(x)+0.3y */
double f_x_y(double x, double y,int equation) {
```

```
	double res=0;
	if (equation == 1) {
		res = x + sin(y / pow(3, 0.5));
	}
	if (equation == 2) {
		res = (0.8 - y * y) * cos(x) + 0.3 * y;
	}
	return res;
}



/*Решение уравнения методом Эйлера*/
void Euler(Value_function_table table, int equation) {

	for (size_t i = 1; i < table.n; i++){
		table.y[i]=table.y[i - 1] + table.h * f_x_y(
		table.x[i - 1], table.y[i - 1],equation);
	}

	cout<<Blue<<"Решение методом Эйлера\n"
		<<Green << "\t\tf(xi)\n"
		<< "\txi\t|\tyi\n"
		<<"--------------------------------------\n";
	for (size_t i = 0; i < table.n; i++){
		cout << "\t" << table.x[i] << "\t|\t"
			<< setprecision(5)<< table.y[i] << "\n";
	}
}



/*Решение уравнения методом Эйлера с пересчётом*/
void Euler_recount(Value_function_table table, int equation) {
	double y;

	for (size_t i = 1; i < table.n; i++){
		y = table.y[i-1]+table.h*f_x_y(table.x[i - 1],
			table.y[i - 1],equation);

		table.y[i]=table.y[i - 1] + table.h / 2 *
			(f_x_y(table.x[i - 1], table.y[i - 1],equation) +
			f_x_y(table.x[i - 1], y,equation));
	}

	cout << Blue << "Решение методом Эйлера с пересчётом\n"
		<< Green << "\t\tf(xi)\n"
		<< "\txi\t|\tyi\n"
		<< "--------------------------------------\n";
	for (size_t i = 0; i < table.n; i++) {
		cout << "\t" << table.x[i] << "\t|\t"
			<< setprecision(5) << table.y[i] << "\n";
	}
```

```
}


/*Решение методом Рунге-Кутта*/
void Runge_Kutt(Value_function_table table, int equation) {
	double k1, k2, k3, k4;

	for (size_t i = 1; i < table.n; i++){
		k1 = f_x_y(table.x[i - 1], table.y[i - 1],equation);

		k2 = f_x_y(table.x[i - 1] + table.h / 4, table.y[i - 1]
		 + k1 * table.h / 4, equation);

		k3 = f_x_y(table.x[i - 1] + table.h / 2, table.y[i - 1]
		+ k2 * table.h / 2, equation);

		k4 = f_x_y(table.x[i - 1] + table.h, table.y[i - 1]
		 + table.h * k1 - 2 * table.h * k2 + 2 * table.h *
		k3, equation);

		table.y[i]=table.y[i - 1] + (k1 + 4*k3 + k4) *
			table.h / 6;
	}

	cout << Blue << "Решение методом Рунге-Кутта\n"
		<< Green << "\t\tf(xi)\n"
		<< "\txi\t|\tyi\n"
		<< "-------------------------------------\n";
	for (size_t i = 0; i < table.n; i++) {
		cout << "\t" << table.x[i] << "\t|\t"
		 << setprecision(5) << table.y[i] << "\n";
	}

}

/*Решение методом Адамса*/
void Adams(Value_function_table table, int equation) {
	//Нахождение первых 3 значений методом Рунге-Кутта
	double k1, k2, k3, k4;
	for (size_t i = 1; i < 4; i++) {
		k1 = f_x_y(table.x[i - 1], table.y[i - 1], equation);

		k2 = f_x_y(table.x[i - 1] + table.h / 4, table.y[i - 1]
		 + k1 * table.h / 4, equation);

		k3 = f_x_y(table.x[i - 1] + table.h / 2, table.y[i - 1]
		 + k2 * table.h / 2, equation);

		k4 = f_x_y(table.x[i - 1] + table.h, table.y[i - 1]
		 + table.h * k1 - 2 * table.h * k2 + 2 * table.h *
		 k3, equation);

		table.y[i] = table.y[i - 1] + (k1 + 4 * k3 + k4)
```

```
                    * table.h / 6;
    }

    //Нахождение остальных значений методом Адамса
    double d_f = 0, d_2_f = 0, d_3_f = 0;
    for (size_t i = 4; i < table.n; i++){
        d_f = f_x_y(table.x[i - 1], table.y[i - 1], equation) -
              f_x_y(table.x[i - 2], table.y[i - 2], equation);

        d_2_f= f_x_y(table.x[i - 1], table.y[i - 1], equation)
                  -2*f_x_y(table.x[i - 2], table.y[i - 2],
                   equation) + f_x_y(table.x[i - 3],
                   table.y[i - 3], equation);

        d_3_f = f_x_y(table.x[i - 1], table.y[i - 1], equation)
                  -3 * f_x_y(table.x[i - 2], table.y[i - 2],
                   equation) +3 * f_x_y(table.x[i - 3],
                   table.y[i - 3], equation) -
                   f_x_y(table.x[i - 4], table.y[i - 4],
                   equation);

        table.y[i] = table.y[i - 1] +table.h *
                    f_x_y(table.x[i - 1], table.y[i - 1],
                    equation) +pow(table.h, 2) * d_f / 2 +5 *
                   pow(table.h, 3) * d_2_f / 12 +3 *
                   pow(table.h, 4) * d_3_f / 8;
    }


    cout << Blue << "Решение методом Адамса\n"
        << Green << "\t\tf(xi)\n"
        << "\txi\t|\tyi\n"
        << "-------------------------------------\n";
    for (size_t i = 0; i < table.n; i++) {
        cout << "\t" << table.x[i] << "\t|\t"
             << setprecision(5) << table.y[i] << "\n";
    }
}
```

Main.cpp

```
#include "solution methods.h"
#include "Value_function_table.h"
#include "Colors.h"
#include<iomanip>
using namespace std;

int main() {
    setlocale(LC_ALL, "Russian");    //Включение русского языка
в консол
    cout<<Blue << "Решение уравнения
y'=f(x,y)=x+sin(y/3^(1/2))\n";
    Value_function_table table_f,table_s;
    table_f.set_value();
```

```
    Euler(table_f,1);
    Euler_recount(table_f,1);
    Runge_Kutt(table_f,1);
    cout << Blue
    << "Решение уравнения y'=f(x,y)=(0.8-y^2)*cos(x)+0.3y\n";
    table_s.set_value();
    Adams(table_s, 2);
    Euler_recount(table_s, 2);

    return 0;
}
```

Результаты работы программы

```
Решение уравнения y'=f(x,y)=x+sin(y/3^(1/2))
X:[a,b]
Введите a 1.1

Введите b 2.1

Введите шаг h 0.1

Введите y0 1.5
Решение методом Эйлера
                f(xi)
        xi      |       yi
-------------------------------------
        1.1     |       1.5
        1.2     |       1.6862
        1.3     |       1.8889
        1.4     |       2.1076
        1.5     |       2.3413
        1.6     |       2.589
        1.7     |       2.8487
        1.8     |       3.1184
        1.9     |       3.3958
        2       |       3.6783
Решение методом Эйлера с пересчётом
                f(xi)
        xi      |       yi
-------------------------------------
        1.1     |       1.5
        1.2     |       1.6894
        1.3     |       1.8952
        1.4     |       2.1166
        1.5     |       2.3525
        1.6     |       2.6012
        1.7     |       2.8609
        1.8     |       3.1294
        1.9     |       3.4041
        2       |       3.6827
Решение методом Рунге-Кутта
                f(xi)
        xi      |       yi
-------------------------------------
        1.1     |       1.5
        1.2     |       1.6946
        1.3     |       1.9057
        1.4     |       2.1325
        1.5     |       2.3738
        1.6     |       2.6279
        1.7     |       2.8928
        1.8     |       3.1661
        1.9     |       3.4454
        2       |       3.728
Решение уравнения y'=f(x,y)=(0.8-y^2)*cos(x)+0.3y
X:[a,b]
Введите a 0

Введите b 1

Введите шаг h 0.1

Введите y0 0
```

```
Решение методом Адамса
                f(xi)
        xi      |       yi
-------------------------------------
        0       |       0
        0.1     |       0.08086
        0.2     |       0.16206
        0.3     |       0.24161
        0.4     |       0.31957
        0.5     |       0.39322
        0.6     |       0.46137
        0.7     |       0.52336
        0.8     |       0.57898
        0.9     |       0.62842
        1       |       0.67217
Решение методом Эйлера с пересчётом
                f(xi)
        xi      |       yi
-------------------------------------
        0       |       0
        0.1     |       0.08088
        0.2     |       0.16249
        0.3     |       0.24279
        0.4     |       0.31994
        0.5     |       0.39247
        0.6     |       0.4594
        0.7     |       0.5202
        0.8     |       0.57479
        0.9     |       0.62341
        1       |       0.66657
```

Вывод

Закрепил знания и умения по численному решению обыкновенных дифференциальных уравнений методом Эйлера и методом Адамса. Сравнил значения, полученные разными способами решения, они получились равны в пределах погрешности.