**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Методы численного анализа

Лабораторная работа №3

Вариант 1

Крученкова Евгения Андреевича

студента 3 курса,

специальность «Прикладная математика»

Преподаватель:

доцент кафедры ВычМат,

кандидат физико-математических наук

Будник А.М.

Минск, 2022

Оглавление

[Постановка задачи. 3](#_Toc122398817)

[Листинг программы. 4](#_Toc122398818)

[1. Явный метод Эйлера 4](#_Toc122398819)

[2. Метод Рунге-Кутта 5](#_Toc122398820)

[3. Метод последовательного повышения порядка точности 8](#_Toc122398821)

[4. Интерполяционный метод Адамса 10](#_Toc122398822)

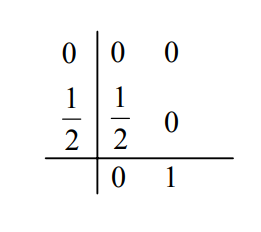
[Выводы 11](#_Toc122398823)

Постановка задачи.

Найти приближенное решение задачи Коши

на сетке узлов при 10-ти разбиениях отрезка интегрирования, применяя следующие методы:

1. Явный метод Эйлера
2. Метод Рунге-Кутта, построенный по таблице Бутчера вида:



1. Метод последовательного повышения порядка точности при q = 1,
2. Интерполяционный метод Адамса 3-го порядка точности с началом таблицы, построенным по соответсвующему методу Рунге-Кутта. Для реализации метода Адамса использовать алгоритм метода простой итерации.

Для проведения анализа полученных результатов необходимо:

А) Используя таблицу приближенных результатов, получить погрешности методов 1-3, считая «точным решением» метод Адамса.

Б) Исходя из вида главного члена локальной погрешности методов 1-3, объяснить разницу результатов.

В) На основе полученных численных и теоретических результатов сделать вывод о точности каждого метода 1-4.

Листинг программы.

**Header.h**

#pragma once

#define \_USE\_MATH\_DEFINES

#include <cmath>

#include <iostream>

#include <stdio.h>

#include <iomanip>

#include <fstream>

#include <string>

#include <time.h>

using namespace std;

const double E = pow(10, -6);

double F(double t, double u) {

return ((2\*u\*u\*u\*t)/(1-t\*t\*u\*u));

}

const double u\_0 = 1;

const double a = 2;

const double b = 2.4;

const double N = 10;

const double tau = (b - a) / N;

double\* JaME() {

double\* Y = new double[N+1];

Y[0] = u\_0;

for (int j = 0; j < N; j++) {

Y[j + 1] = Y[j] + tau \* F(a + tau \* j, Y[j]);

cout << "Значение в точке " << fixed << showpoint << setprecision(2) << a + (j + 1) \* tau << ": " << fixed << showpoint << setprecision(8) << Y[j + 1] << endl;

}

return Y;

}

double K\_2(double t, double u) {

return F(t + tau / 2, u + (tau / 2) \* F(t, u));

}

double\* MRCute() {

double\* Y = new double[N + 1];

Y[0] = u\_0;

for (int j = 0; j < N; j++) {

Y[j + 1] = Y[j] + tau \* K\_2(a + tau \* j, Y[j]);

cout << "Значение в точке " << fixed << showpoint << setprecision(2) << a + (j + 1) \* tau << ": " << fixed << showpoint << setprecision(8) << Y[j + 1] << endl;

}

return Y;

}

double\* MPPPT(double\* Y\_JaME) {

double\* Y = new double[N + 1];

Y[0] = u\_0;

for (int j = 0; j < N; j++) {

Y[j + 1] = Y[j] + (tau / 2) \* (F(a + tau \* j, Y[j]) + F(a + tau \* (j + 1), Y\_JaME[j + 1]));

cout << "Значение в точке " << fixed << showpoint << setprecision(2) << a + (j + 1) \* tau << ": " << fixed << showpoint << setprecision(8) << Y[j + 1] << endl;

}

return Y;

}

double\* NMA(double Y\_MRK) {

double\* Y = new double[N + 1];

Y[0] = u\_0;

Y[1] = Y\_MRK;

cout << "Значение в точке " << fixed << showpoint << setprecision(2) << a + 1 \* tau << ": " << fixed << showpoint << setprecision(8) << Y[1] << endl;

for (int j = 1; j < N; j++) {

int k = 0;

Y[j + 1] = Y[j];

double pred\_y;

do {

pred\_y = Y[j + 1];

Y[j + 1] = Y[j] + (tau / 12) \* (5 \* F(a + tau \* (j + 1), pred\_y) + 8 \* F(a + tau \* j, Y[j]) - F(a + tau \* (j - 1), Y[j - 1]));

} while (abs(Y[j + 1] - pred\_y) > pow(tau, 5));

cout << "Значение в точке " << fixed << showpoint << setprecision(2) << a + (j + 1) \* tau << ": " << fixed << showpoint << setprecision(8) << Y[j + 1] << endl;

}

return Y;

}

**Main.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include "Header.h"

#include <fstream>

#include<time.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Явный метод Эйлера:" << endl;

double\* Y\_me = JaME();

cout << endl << "Метод Рунге-Кутты:" << endl;

double\* Y\_mrk = MRCute();

cout << endl << "Метод последовательного повышения порядка точности:" << endl;

double\* Y\_mpppt = MPPPT(Y\_me);

cout << endl << "Метод Адамса:" << endl;

double\* Y\_ma = NMA(Y\_mrk[1]);

cout << endl << "Невязка для явного метода Эйлера:" << endl;

for (int j = 0; j < N; j++) {

cout << "Невязка в точке " << fixed << showpoint << setprecision(2) << a + (j + 1) \* tau << ": " << scientific << abs(Y\_me[j+1]-Y\_ma[j+1]) << endl;

}

cout << endl << "Невязка для метода Рунге-Кутты:" << endl;

for (int j = 0; j < N; j++) {

cout << "Невязка в точке " << fixed << showpoint << setprecision(2) << a + (j + 1) \* tau << ": " << scientific << abs(Y\_mrk[j + 1] - Y\_ma[j + 1]) << endl;

}

cout << endl << "Невязка для метода ПППТ:" << endl;

for (int j = 0; j < N; j++) {

cout << "Невязка в точке " << fixed << showpoint << setprecision(2) << a + (j + 1) \* tau << ": " << scientific << abs(Y\_mpppt[j + 1] - Y\_ma[j + 1]) << endl;

}

system("pause");

return 0;

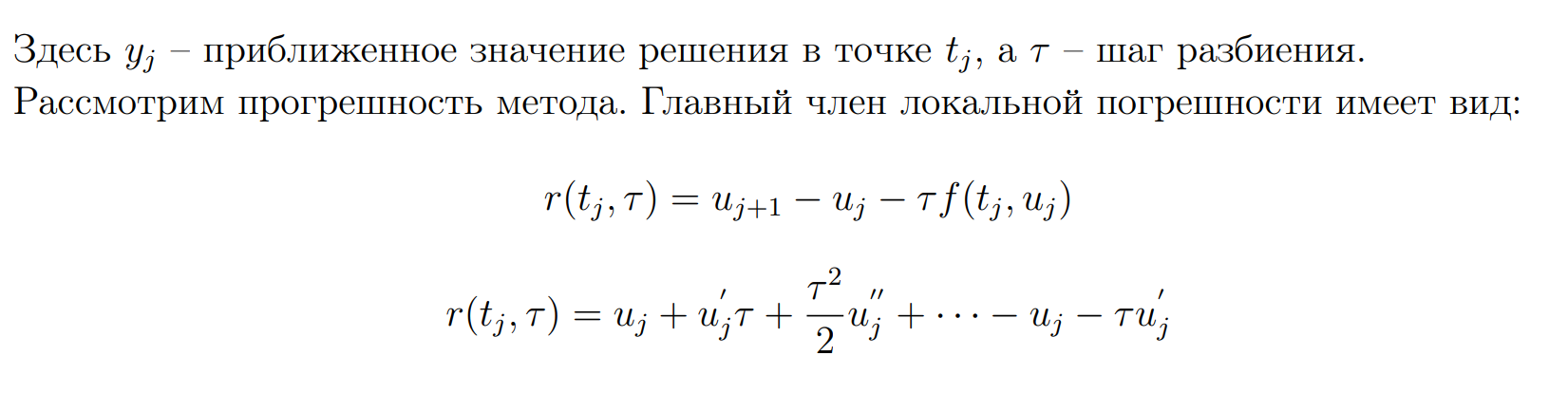
}

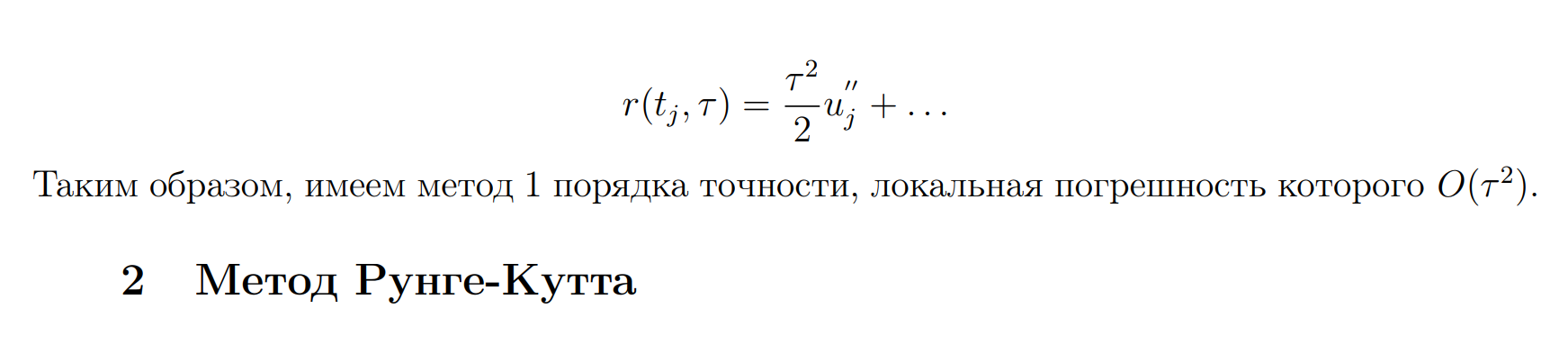
1. Явный метод Эйлера

Краткие теоретические сведения.

Явный метод Эйлера имеет следующий вид:

*, j = 0,1,…,N-1*





Результаты

Значения функции u(t) в точках

Значение в точке 2.04: 0.94666667

Значение в точке 2.08: 0.89594156

Значение в точке 2.12: 0.84754722

Значение в точке 2.16: 0.80121247

Значение в точке 2.20: 0.75666380

Значение в точке 2.24: 0.71361313

Значение в точке 2.28: 0.67173903

Значение в точке 2.32: 0.63065423

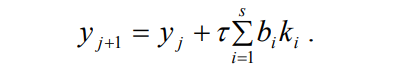
Значение в точке 2.36: 0.58984342

Значение в точке 2.40: 0.54852671

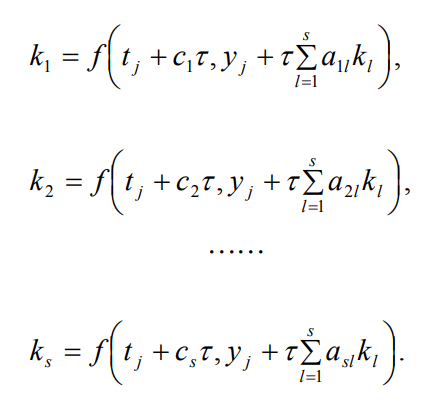
2. Метод Рунге-Кутта

Краткие теоретические сведения

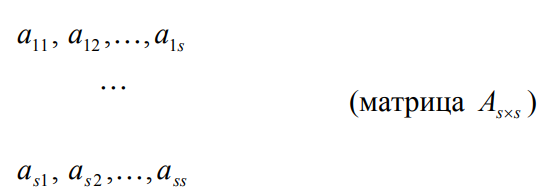
Метод Рунге-Кутта имеет вид:

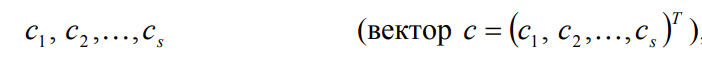


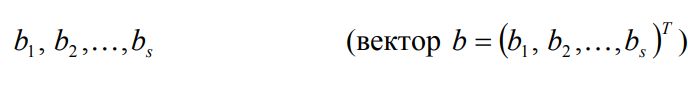
где



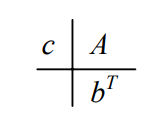
а все необходимые коэффициенты:



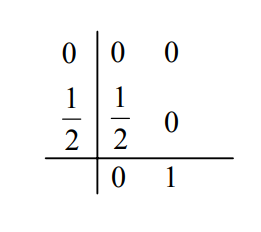




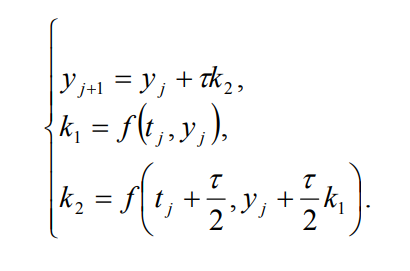
задаются в виде таблицы Бутчера следующего вида:



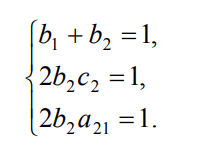
В нашем случае таблица Бутчера выглядит следующим образом:



Будем иметь явный метод, так как элементы матрицы А на главной диагоняли и выше равны нулю. Перепишем его в виде:



Определим порядок точности метода. Проверим условия для второго порядка точности:



Условия выполнены. Для третьего порядка точности условия выполнены не будут. Следовательно, имеем метод второго порядка точности с локальной погрешностью

Результаты

Значения функции u(t) в точках

Значение в точке 2.04: 0.94800059

Значение в точке 2.08: 0.89845458

Значение в точке 2.12: 0.85110090

Значение в точке 2.16: 0.80568173

Значение в точке 2.20: 0.76193349

Значение в точке 2.24: 0.71957369

Значение в точке 2.28: 0.67827966

Значение в точке 2.32: 0.63765027

Значение в точке 2.36: 0.59712825

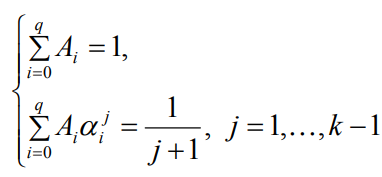
Значение в точке 2.40: 0.55581567

3. Метод последовательного повышения порядка точности

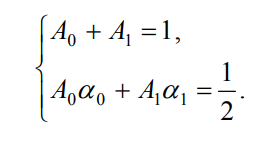
Краткие теоретические сведения

В общем случае метод будет иметь вид:

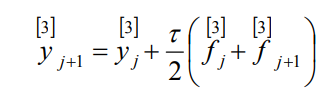
где для порядка точности находятся из системы:



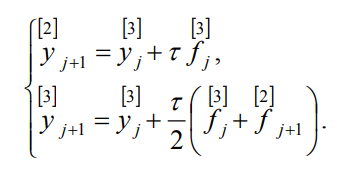
В нашем случае: k = 2, q = 1,



При получим .



Или в явном виде при помощи ЯМЭ:



При реализации передадим методу полученный в первом пункте вектор значений.

Порядок точности данного метода – второй, локальная погрешность будет

Результаты

Значение в точке 2.04: 0.94797078

Значение в точке 2.08: 0.89840167

Значение в точке 2.12: 0.85102423

Значение в точке 2.16: 0.80557229

Значение в точке 2.20: 0.76177194

Значение в точке 2.24: 0.71932656

Значение в точке 2.28: 0.67789204

Значение в точке 2.32: 0.63703077

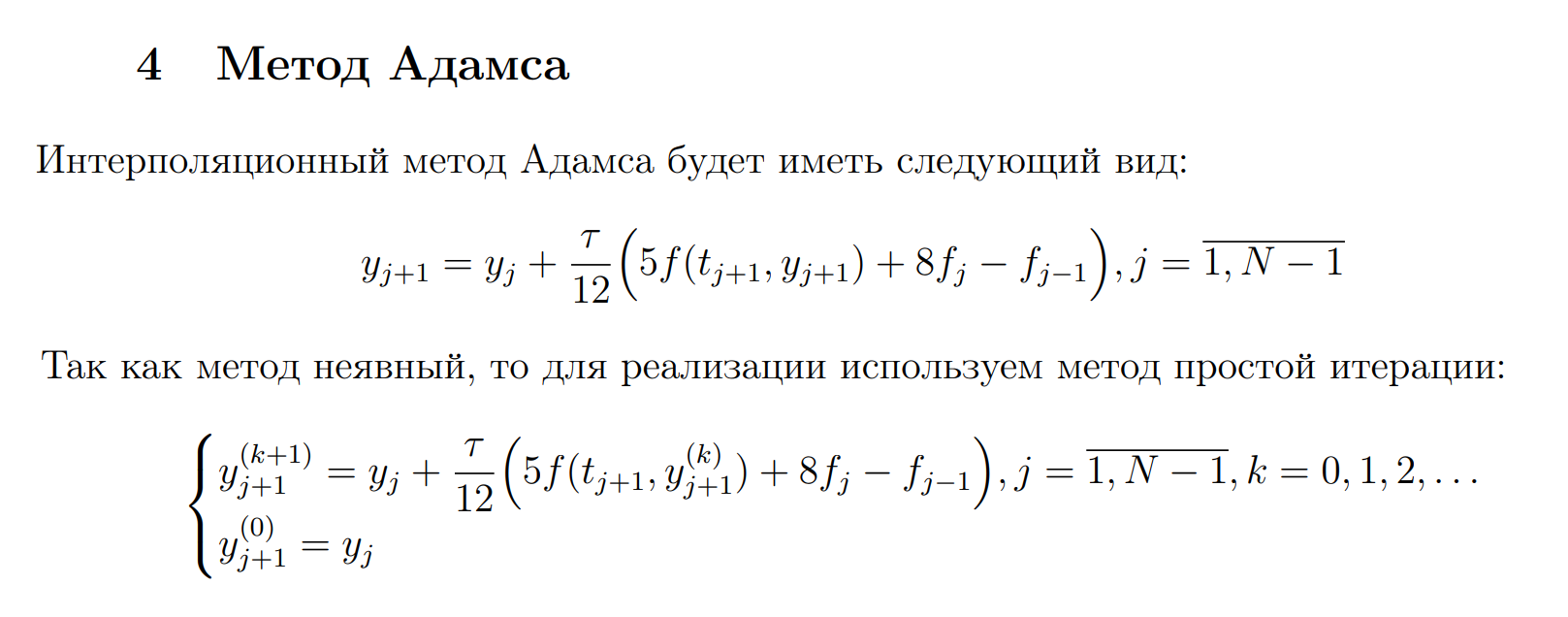
Значение в точке 2.36: 0.59611444

Значение в точке 2.40: 0.55407927

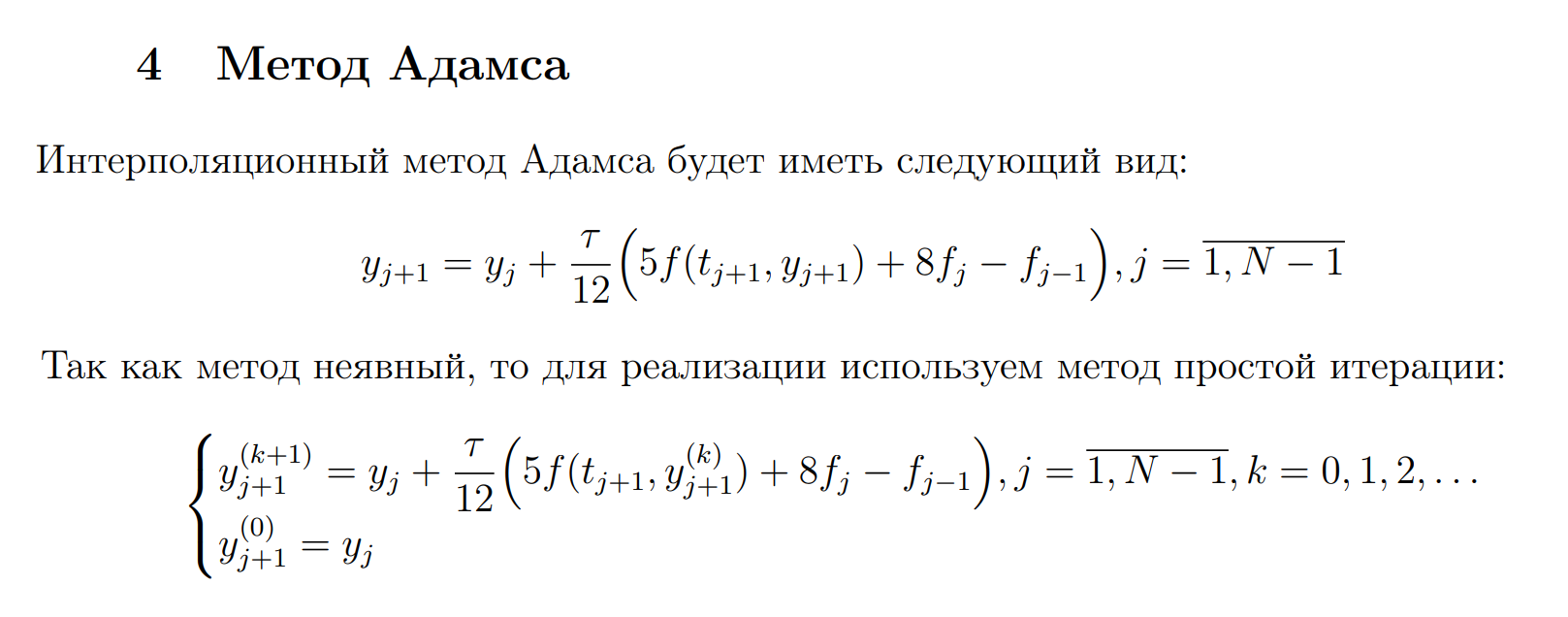
4. Интерполяционный метод Адамса

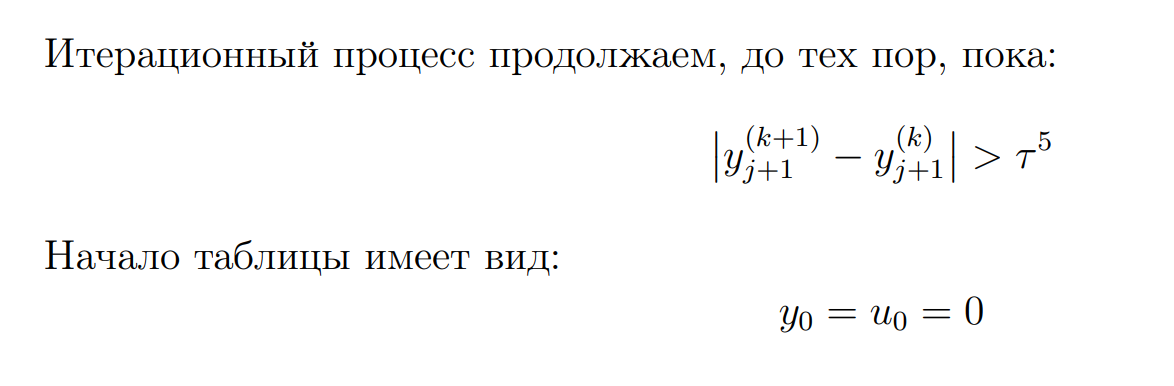
Краткие теоретические сведения

Метод Адамса третьего порядка будет иметь вид:



В нашем случае N = 10. Так как метод третьего порядка, то локальная погрешность метода будет





Для начала данного процесса нам необходимо начало таблицы: Его мы возьмём как результаты для из второго пункта.

Результаты

Значение в точке 2.04: 0.94800059

Значение в точке 2.08: 0.89844027

Значение в точке 2.12: 0.85107381

Значение в точке 2.16: 0.80564264

Значение в точке 2.20: 0.76188221

Значение в точке 2.24: 0.71950839

Значение в точке 2.28: 0.67819546

Значение в точке 2.32: 0.63753566

Значение в точке 2.36: 0.59695469

Значение в точке 2.40: 0.55549986

Выводы

Вычислим невязку по каждому из трёх первых методов, считая точным решением метода Адамса.

***1) Невязка для явного метода Эйлера:***

Невязка в точке 2.04: 1.33e-03

Невязка в точке 2.08: 2.50e-03

Невязка в точке 2.12: 3.53e-03

Невязка в точке 2.16: 4.43e-03

Невязка в точке 2.20: 5.22e-03

Невязка в точке 2.24: 5.90e-03

Невязка в точке 2.28: 6.46e-03

Невязка в точке 2.32: 6.88e-03

Невязка в точке 2.36: 7.11e-03

Невязка в точке 2.40: 6.97e-03

Во-первых, нетрудно заметить, что локальная погрешность метода, а в данном случае мы рассматриваем метод первого порядка с локальной погрешностью , приблизительна равна реальной погрешности метода. В нашем случае , степень при реальной погрешности такая же. Также можно отметить, что с каждым шагом погрешность возрастает, хоть в данном случае и незначительно.

***2) Невязка для метода Рунге-Кутты:***

Невязка в точке 2.04: 0.00e+00

Невязка в точке 2.08: 1.43e-05

Невязка в точке 2.12: 2.71e-05

Невязка в точке 2.16: 3.91e-05

Невязка в точке 2.20: 5.13e-05

Невязка в точке 2.24: 6.53e-05

Невязка в точке 2.28: 8.42e-05

Невязка в точке 2.32: 1.15e-04

Невязка в точке 2.36: 1.74e-04

Невязка в точке 2.40: 3.16e-04

Данный метод имеет второй порядок точности, его локальная погрешность, как было отмечено в описании метода, равна . В нашем случае . Первое приближенное значение имеет нулевую погрешность, так как начало таблицы для метода Адамса было построено именно по данному методу. Реальная погрешность второго значения не превосходит локальной погрешности. Далее происходит накопление погрешности, однако, как и требуется, она не возрастает на каком-либо шаге больее, чем на . Данный метод, как и ожидалось, получился точнее предыдущего метода первого порядка.

***3) Невязка для метода ПППТ:***

Невязка в точке 2.04: 2.98e-05

Невязка в точке 2.08: 3.86e-05

Невязка в точке 2.12: 4.96e-05

Невязка в точке 2.16: 7.03e-05

Невязка в точке 2.20: 1.10e-04

Невязка в точке 2.24: 1.82e-04

Невязка в точке 2.28: 3.03e-04

Невязка в точке 2.32: 5.05e-04

Невязка в точке 2.36: 8.40e-04

Невязка в точке 2.40: 1.42e-03

Данный метод также является методом второго порядка точности, его локальная погрешность равна . В нашем случае . И в данном методе реальна погрешность не превосходит локальной, однако на каждом шаге реальная погрешность возрастает быстрее, чем у предыдущего метода. Это может быть обусловлено константой при в , которая для данного метода могла оказаться больше, чем для предыдущего. К тому же данный метод чуть более трудоёмкий, чем предыдущий.

***4) Метод Адамса:***

Так как данный метод использовался в качестве точного решения, оценить его реальную погрешность не представляется возможным. Локальная погрешность данного метода равна . В нашем случае . Используя данный метод в качестве точного решения, мы получили правдивые оценки реальной погрешности, соответствующие локальным погрешностям и убедились в их прямой связи.