Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Институт электронной техники и приборостроения

Кафедра Информационная безопасность автоматизированных систем

Специальность 10.03.01 Информационная безопасность автоматизированных систем

**КУРСОВОЙ ПРОЕКТ**

по дисциплине «Вычислительная математика»

по теме «Метод Адамса как многошаговый метод для решения обыкновенных дифференциальных уравнений»

|  |  |
| --- | --- |
|  | Выполнил: студент 3 курса  учебной группы с1-ИБС32  очной формы обучения  Солодилов В.В. \_\_\_\_\_\_\_\_\_\_\_  Руководитель проекта:  проф. каф. ИБС Байбурин В.Б,  Комиссия по защите:  проф. каф. ИБС Байбурин В.Б.  доцент каф. ИБС Кожанова Е.Р. |

Курсовой проект защищен на оценку \_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись члена комиссии)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись члена комиссии)

Саратов 2022

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Кафедра Информационная безопасность автоматизированных систем

**Задание на курсовой проект**

студенту 3 курса учебной группы с1-ИБС32

Институт электронной техники и приборостроения

Солодилову Владимиру Владимировичу

по дисциплине «Вычислительная математика»

по теме «Метод Адамса как многошаговый метод для решения обыкновенных дифференциальных уравнений»

Сроки выполнения работы 10.03.2022 г.- 31.05.2022г.

Защита работы 31.05.2022г.

Руководитель проекта Байбурин В.Б.

Задание принял к исполнению Солодилов В.В.

**Содержание**

|  |  |
| --- | --- |
| Введение | 4 |
| 1. Основные понятия метода Адамса как многошагового метода для решения обыкновенных дифференциальных уравнений | 6 |
| 1.1. Метод Адамса-Бэшфорта | 8 |
| 1.2. Метод Адамса-Мултона | 10 |
| 2. Постановка задачи | 11 |
| 2.1. Формулирование задачи | 11 |
| 2.2. Решение задачи | 11 |
| 3. Программная реализация и тестирование | 13 |
| 3.1. Выбор языка программирования и построение блок-схемы | 13 |
| 3.2. Программная реализация | 13 |
| 3.3. Тестирование программы | 13 |
| Заключение | 16 |
| Список использованных источников | 17 |
| Приложение А. Блок-схема программы | 18 |
| Приложение Б. Листинг программы | 22 |

**Введение**

В современной вычислительной математике разработано множество численных методов решения обыкновенных дифференциальных уравнений разных порядков. Для того, чтобы применить данные методы, должна быть, например, определена задача Коши для этих уравнений и всех их производных кроме производных наивысшего для данных уравнений порядка. Например, задача Коши для однородных дифференциальных уравнений первого порядка ставится следующим образом:

|  |  |
| --- | --- |
|  | (1) |

Наиболее оптимальным численным методом решения таких уравнений является метод конечных разностей или сеточный метод. Благодаря сеточным методам, можно сводить приближенное решение уравнений к решению систем линейных алгебраических уравнений. Для теории разностных схем характерно, что для дифференциального уравнения существует решение искомой задачи, и оно имеет необходимое число производных, обеспечивающее максимальный порядок аппроксимации.

Разностные схемы расцениваются как операторные или операторно- разностные уравнения с линейными операторами, зависящими от параметра *h* и заданными на абстрактном линейном нормированном пространстве любого числа измерений. Основным понятием в теории разностных схем является понятие устойчивости. Достаточные условия устойчивости позволяют формулировать общий принцип регуляризации схем для получения разностных схем заданного качества. При построении разностной схемы необходимо заменить дифференциальный оператор некоторым разностным оператором, необходимо заменить область непрерывного изменения аргумента областью дискретного его изменения.

К численным методам решения однородных дифференциальных уравнений относят одношаговые и многошаговые методы, каждые из которых имеют явную и неявную реализацию. К одношаговым методам можно отнести методы Эйлера или Рунге-Кутты.

Одним из многошаговых методов решения однородных дифференциальных уравнений является метод Адамса. Он позволяет найти приближенные значения исходной функции при заданных значениях аргумента функции.

Цель курсового проекта – изучить метод Адамса как многошаговый метод для решения обыкновенных дифференциальных уравнений с дальнейшей программной реализацией.

Для решения поставленной задачи необходимо решить ряд задач:

1) Изучить основные типы методов Адамса, используемые при решения обыкновенных дифференциальных уравнений.

2) сформулировать задачу для дальнейшей программной реализации.

3) выполнить реализацию и тестирование программной реализации поставленной задачи.

Курсовой проект состоит из введения, трех глав, заключения, списка использованных источников и двух приложений. В приложении А представлены блок-схемы программы и подпрограмм. В приложении Б содержатся листинги программы.

1. **Основные понятия метода Адамса как многошагового метода решения однородных дифференциальных уравнений**

Многошаговый метод Адамса относится к конечно-разностным методам решения однородных дифференциальных уравнений. В отличие от одношаговых методов, в которых для расчёта значения функции в узловой точке используется значение этой функции в соседней точке, в конечно-разностных схемах для вычисления значения в узловой точке используют значения в нескольких соседних узловых точках.

Доказано, что при применении многошаговых численных методов Адамса для решения задачи Коши до двенадцатого порядка область устойчивости уменьшается. При дальнейшем увеличении порядка область устойчивости, а также точность метода возрастает. Кроме того, при одинаковой точности для многошаговых методов на одном шаге интегрирования требуется меньше вычислений правых частей дифференциальных уравнений, чем в методах Рунге-Кутты. К достоинствам методов Адамса относится и то обстоятельство, что в них легко меняется шаг интегрирования и порядок метода.

Рассмотрим общий случай задания формулами метода Адамса.

В общем случае линейные k-шаговые методы задаются формулами вида:

|  |  |
| --- | --- |
|  | (2) |

Методы Адамса имеют при *j > 1* и получены из условия максимального порядка при заданном *k*. Явные методы Адамса имеют порядок *k*, а неявные – порядок *k + 1*. При *k = 1* получаем, соответственно, явный метод и неявный метод трапеций. При *k = 2* и постоянном размере шага формулу явного метода можно представить как

|  |  |
| --- | --- |
|  | (3) |

А формулу неявного метода можно записать в виде

|  |  |
| --- | --- |
|  | (4) |

При *k ≥ 2* явные методы Адамса имеют очень ограниченные области устойчивости, поэтому самостоятельно они не применяются. Области устойчивости неявных методов Адамса при *k ≥ 2* также ограничены, что делает неэффективным их использование для решения жестких задач. В то же время сочетание явных и неявных формул Адамса позволяет построить весьма эффективные методы прогноза-коррекции.

При *k = 2* и *h = const* эти формулы можно представить в следующем виде:

|  |  |
| --- | --- |
|  | (5) |
|  | (6) |

На практике широко используются два типа методов Адамса – явные и неявные. Явные методы известны как методы Адамса-Бэшфорта, неявные – как методы Адамса-Мултона.

Рассмотрим применение формул (3-6) для каждого из данных типов метода Адамса.

* 1. **Метод Адамса-Бэшфорта**

Рассмотрим применение метода Адамса для решения задачи Коши:

|  |  |
| --- | --- |
|  | (7) |

Если подставить в формулу (7) точное значение *y(x)* и проинтегрировать уравнение на отрезке [], то получим, что

|  |  |
| --- | --- |
|  | (8) |

Заменяя в формуле функцию *f(x, y(x))* интерполяционным полиномом *P(x)*, получим приближенный метод:

|  |  |
| --- | --- |
|  | (9) |

Для того, чтобы построить полином *P(x)*, предположим, что - приближения к решению в точках . Полагаем, что узлы расположены равномерно с шагом h. Тогда

|  |  |
| --- | --- |
|  | (10) |

– приближение к *f(x,y(x))* в точках .

В качестве *P(x)* возьмем интерполяционный полином степени k, удовлетворяющий условиям:

|  |  |
| --- | --- |
|  | (11) |

Если проинтегрировать этот полином явно, то получим:

|  |  |
| --- | --- |
|  | (12) |

При *k = 0* полином P(x) – есть константа, равная , и формула превращается в обычный метод Эйлера.

При *k = 1*, *P(x)* является линейной функцией, проходящей через точки , то есть

|  |  |
| --- | --- |
|  | (13) |

Интегрируя этот полином от , получим двухшаговый метод Адамса-Бэшморта:

|  |  |
| --- | --- |
| , | (14) |

который использует информацию в двух точках .

Если *k = 2*, то *P(x)* представляет собой квадратичный полином, интерполирующий данные . Можно показать, что соответствующий метод имеет вид:

|  |  |
| --- | --- |
| . | (15) |

Если *k = 3*, то соответствующий метод определяется формулой

|  |  |
| --- | --- |
| . | (16) |

При *k = 4* имеем:

|  |  |
| --- | --- |
| . | (17) |

Отметим, что метод (15) является трехшаговым, (16) – четырехшаговым и (17) – пятишаговым. Формулы (14 - 17) известны как методы Адамса-Бэщфорта. Метод имеет второй порядок точности, поэтому его называют методом Адамса-Бэшфорта второго порядка. Аналогично, методы называются соответственно методами Адамса-Бэшфорта третьего, четвертого и пятого порядков.

* 1. **Метод Адамса-Мултона**

Методы Адамса-Бэшфорта используют уже известные значения в точках . При построении интерполяционного полинома можно использовать также точки . При этом возникает класс неявных m-шаговых методов, известных как методы Адамса-Мултона.

Формула, характеризующая данный метод, имеет вид:

|  |  |
| --- | --- |
|  | (18) |

и соответствует k-шаговому методу Адамса – Мултона. Данный метод также принято называть интерполяционным методом Адамса.

Если *k = 0*, то *P(x)* – линейная функция, проходящая через точки () и (), и соответствующий метод

|  |  |
| --- | --- |
|  | (19) |

является методом Адамса-Мултона второго порядка.

При *k = 1, 2, 3* получаем соответствующие методы

|  |  |
| --- | --- |
|  | (20) |
|  | (21) |
|  | (22) |

третьего, четвертого и пятого порядков аппроксимации. Соотношения содержат искомые значения неявно, поэтому для их реализации необходимо применять итерационные методы.

**2. Постановка задачи**

**2.1. Формулирование задачи**

Решить задачу Коши для однородного дифференциального уравнения первого порядка, используя метод Адамса-Бэшфорта третьего и четвертого порядков точности:

|  |  |
| --- | --- |
|  | (23) |

с заданным начальным условием:

,

точным решением, которого является функция:

|  |  |
| --- | --- |
| . | (24) |

**2.2. Решение задачи**

Решим данное дифференциальное уравнение методом Адамса третьего и четвертого порядков.

Для начала определим значение *y0 = 2.*

Вычислим шаг аргумента:

*.*

Определим число итераций и вычислим значения аргументов функции:

, где *.*

Найдем значения функции в правой части дифференциального уравнения:

*,*

где y – значение исходной функции, найденное методом Адамса.

Вычислим значение функции дифференциального уравнения для первых трех узлов, используя метод Эйлера:

Начиная с *n = 4*, воспользуемся формулой (15) для нахождения значений функций методом Адамса третьего порядка.

А для нахождения значений функций методом Адамса четвертого порядка воспользуемся формулой (16).

Применив вышеуказанные уравнения, произведем расчёт, согласно [7777], результаты, приведенные в источнике, совпадают с расчетом в ЬЫ Учсуд (рисунок 1):

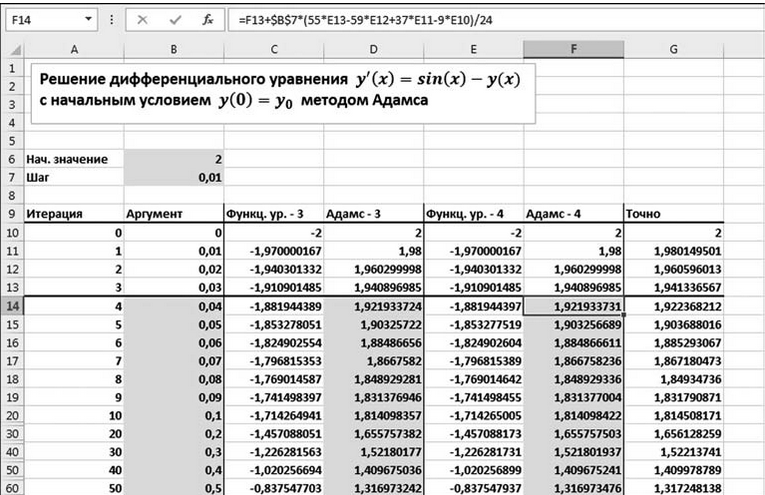


Рисунок 1 – Результат решения задачи при помощи интернет - ресурса

**3. Программная реализация и тестирование**

**3.1. Выбор языка программирования и построение блок-схемы**

В рамках курсовой работы был выбран язык программирования Java.

На основании решения задачи в главе 2 данного курсового проекта были построены блок-схемы (Приложение А):

1) основной программы main (1; Приложение А),

2) функции analyticalValue (2; Приложение А),

3) функции difFunctionValue(3; Приложение А),

4) функции xValue (4; Приложение А),

5) функции Adams3 (5; Приложение А),

6) функции Adams4 (6; Приложение А).

**3.2. Программная реализация**

Программа, реализующая метод Адамса третьего и четвертого порядков для конкретной задачи, состоит из:

1. основного метода main() (1, Приложение Б),
2. метода analyticalValue(double h, int N) (2; Приложение Б),
3. метода difFunctionValue(int N, double[] x, double h) (3; Приложение Б),
4. метода xValue(int N, double h) (4; Приложение Б),
5. метода Adams3(int N, double[] x, double h) (5; Приложение Б),
6. метода Adams4(int N, double[] x, double h) (6; Приложение Б).

**3.3. Тестирование программы**

В главе 2 курсового проекта был произведен расчет следующими инструментами:

1) MS Excel.

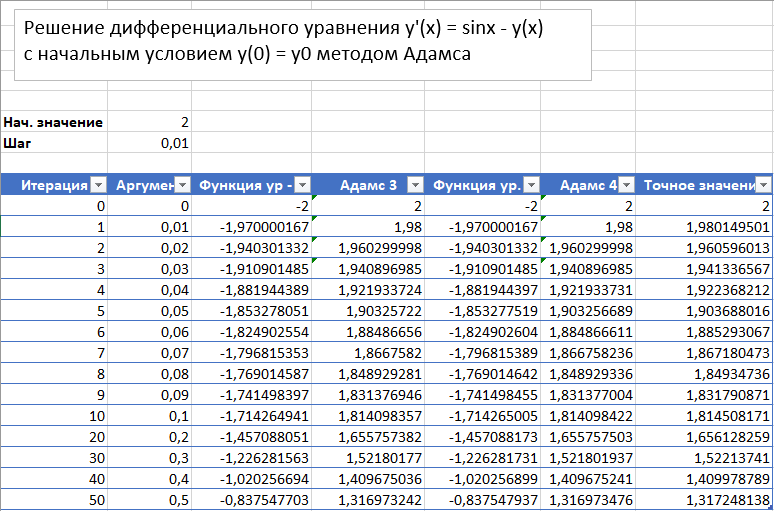


Рисунок 2 – Результат решения задачи в MS Excel

2) Литературный источник [7777].

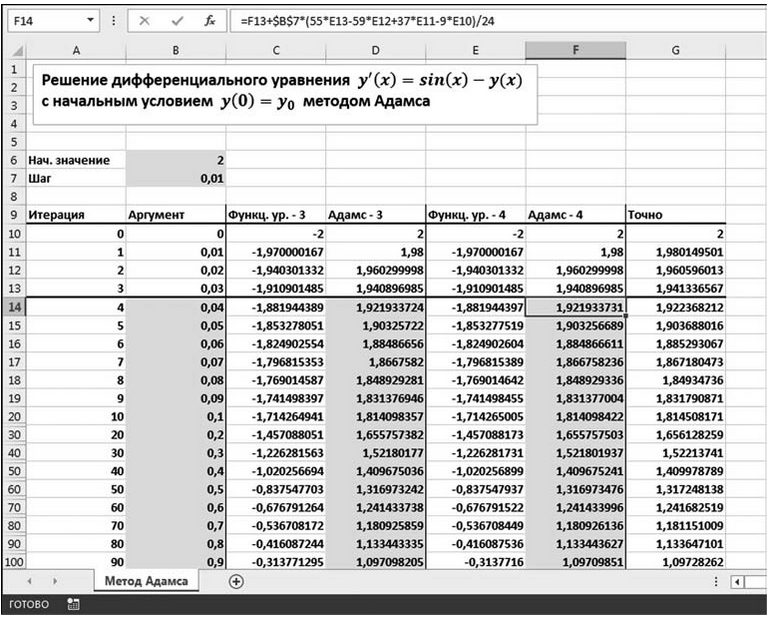


Рисунок 3 – Результат решения задачи [777]

Сравним полученные результаты с результатами программы (см. Приложение Б).

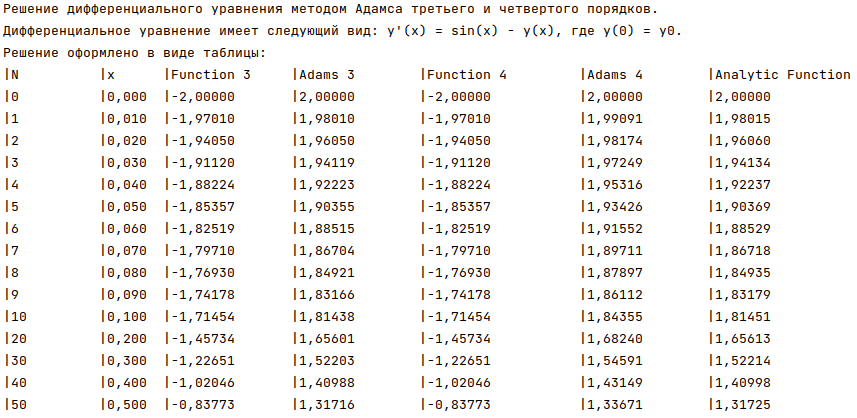


Рисунок 4 – Результат решения задачи при программной реализации

В рамках курсового проекта был изучен многошаговый метод Адамса для решения обыкновенных дифференциальных уравнений, а также написана программа на языке программирования Java, реализующая решение конкретного дифференциального уравнения методом Адамса третьего и четвертого порядков. Результаты программы и ручного расчёта совпали (рисунки 2, 4).

**Заключение**

В ходе выполнения курсовой работы были решены задачи:

1) Изучены основные понятия решения обыкновенных дифференциальных уравнений методом Адамса, а также типы данного метода. В ходе работы был использован метод Адамса-Бэшфорта третьего и четвертого порядков, так как он позволяет получить приближенные значения исходной функции за меньшее количество шагов по сравнению с одношаговыми методами. Полученные результаты были сравнены между собой и с аналитическим решением дифференциального уравнения, в результате чего было подтверждено, что метод Адамса четвертого порядка точнее того же метода, но третьего порядка.

2) Сформулирована задача для дальнейшей программной реализации.

Требуется решить задачу Коши для однородного дифференциального уравнения первого порядка:

,

при заданных начальных условиях .

Произведен расчёт при MS Excel и онлайн ресурса (…). Вывод: результаты совпали.

3) Выполнена реализация программы на языке программирования Java и тестирование программной реализации поставленной задачи. Программа состоит из основной программы и пяти подпрограмм, блок - схемы которых приведены в Приложении А, а их листинги в Приложении Б. Выполнено тестирование написанной программы. Результаты программы совпали с результатами расчета в главе 2.

Таким образом, все поставленные задачи решены, следовательно, цель курсового проекта достигнута.

**Список использованных источников**

!! 10-15 источников (не старше 2015)

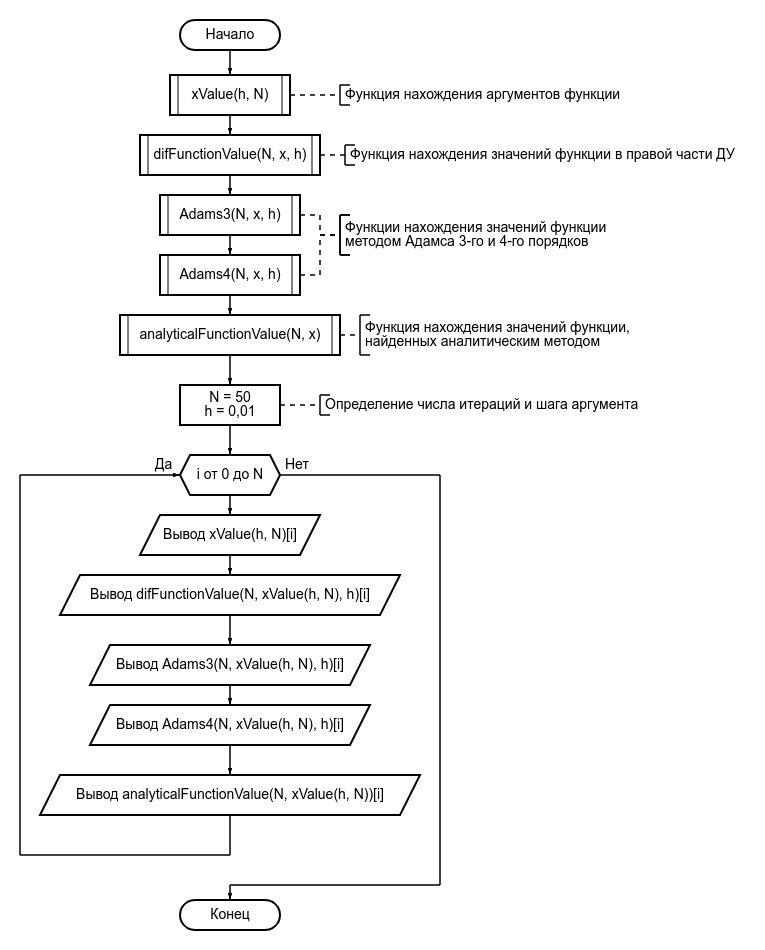
Elibrary

1. [Копченова, Н. В.](http://irbis.sstu.ru/cgi-bin/irbis64r_15/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=SGTU&P21DBN=SGTU&S21STN=1&S21REF=3&S21FMT=fullwebr&C21COM=S&S21CNR=10&S21P01=0&S21P02=1&S21P03=A=&S21STR=Копченова%2C%20Н.%20В.) Вычислительная математика в примерах и задачах : учеб. пособие / Н. В. Копченова, И. А. Марон. - 2-е изд., стер. - СПб.; Москва; Краснодар: Лань, 2008. - 368 с.

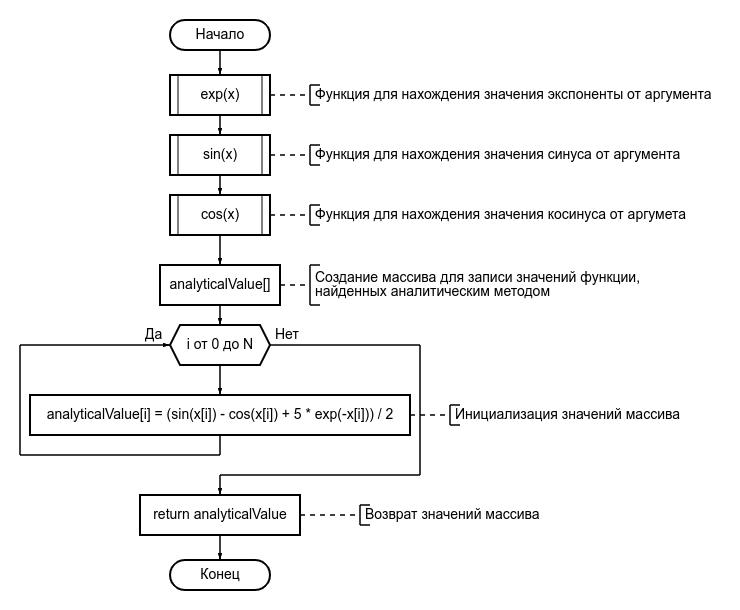
**Приложение А**

**Блок-схема программы**

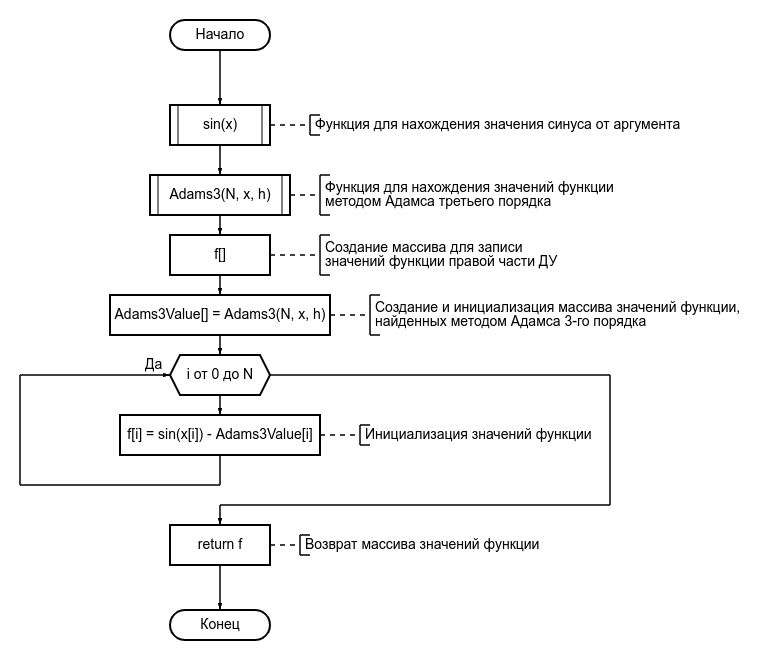
1) блок-схема программы main



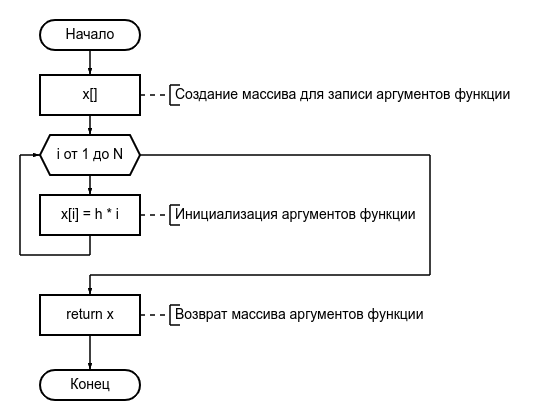
2) блок-схема функции analyticalValue



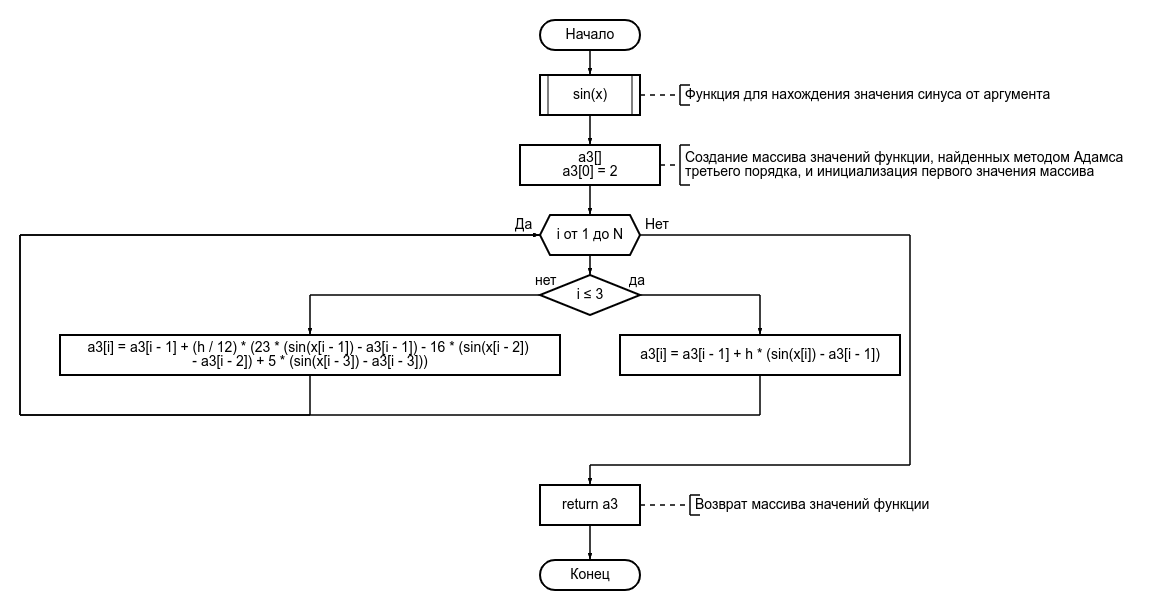
3) блок-схема функции difFunctionValue



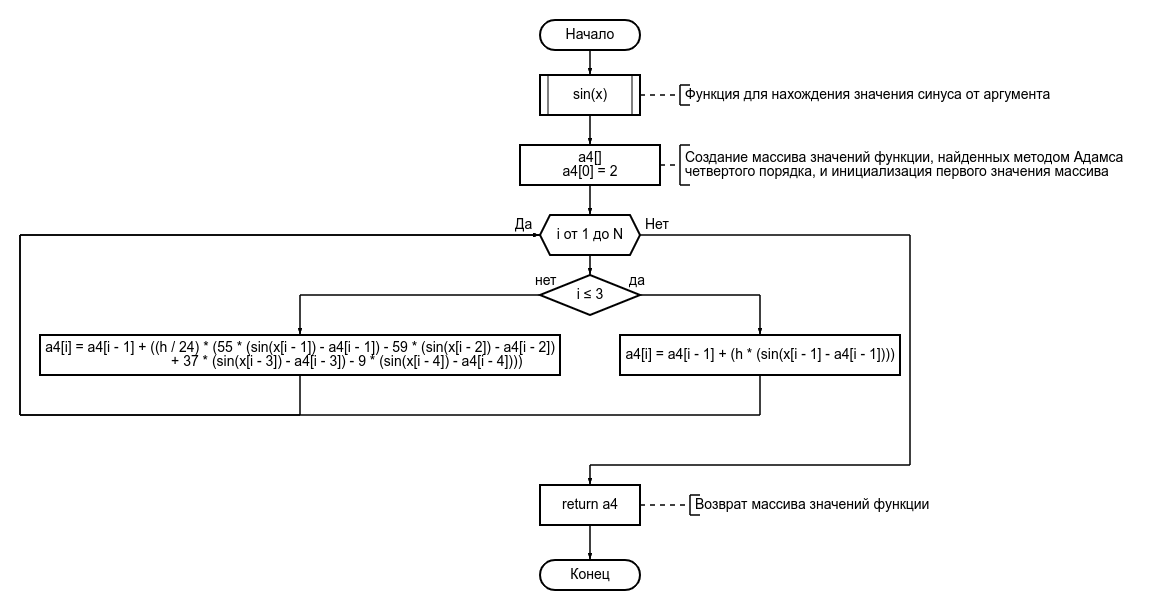
4) блок-схема функции xValue



5) блок-схема функции Adams3



6) блок-схема функции Adams4



**Приложение Б**

**Листинг программы**

1) программа main()

public static void main(String[] args) {

int N = 50; //Задаём число итераций

double h = 0.01; //Определяем шаг аругмента

System.out.println("Решение дифференциального уравнения методом Адамса третьего и четвертого порядков.");

System.out.println("Дифференциальное уравнение имеет вид y'(x) = sin(x) - y(x), где y(0) = y0.");

System.out.println("Решение оформлено в виде таблицы: ");

System.out.println("|N\t\t\t|x\t\t|Function 3\t\t|Adams 3\t\t|Function 4\t\t\t|Adams 4\t\t|Analytic Function");

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

if ((i > 0 && i < 10) || i % 10 == 0) {

System.out.println("|" + i + "\t\t\t|" + String.format("%.3f", xValue(h, N)[i]) + "\t|" + String.format("%.5f", difFunctionValue(N, xValue(h, N), h)[i]) + "\t\t|"

+ String.format("%.5f", Adams3(N, xValue(h, N), h)[i]) + "\t\t|" + String.format("%.5f", difFunctionValue(N, xValue(h, N), h)[i])

+ "\t\t\t|" + String.format("%.5f", Adams4(N, xValue(h, N), h)[i]) + "\t\t|" + String.format("%.5f", analyticalFunctionValue(N, xValue(h, N))[i])); //Выводим все значения, необходимые для выполнения задания

}

}

}

2) функция analyticalValue(double h, int N)

public static double[] analyticalFunctionValue(int N, double[] x) {

double[] analyticalValue = new double[N];

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

analyticalValue[i] = (Math.sin(x[i]) - Math.cos(x[i]) + 5 \* Math.exp(-x[i])) / 2; //Инициализируем массив значений исходной функции, найденных аналитическим методом

}

return analyticalValue;

}

3) функция difFunctionValue(int N, double[] x, double h)

public static double[] difFunctionValue(int N, double[] x, double h) {

double[] Adams3Value = Adams3(N, x, h);

double[] f = new double[N];

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

f[i] = Math.sin(x[i]) – Adams3Value[i]; //Инициализируем массив значений функций в правой части дифференциального уравнения

}

return f;

}

4) функция xValue(int N, double h)

public static double[] xValue(double h, int N) {

double[] x = new double[N];

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

x[i] = h \* i; //Инициализируем массив аргументов

}

return x;

}

5) функция Adams3(int N, double[] x, double h)

public static double[] Adams3(int N, double[] x, double h) {

double[] a3 = new double[N];

a3[0] = 2; //Задаем начальное значение функции, согласно условию

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

if (i <= 3) {

a3[i] = a3[i - 1] + h \* (Math.sin(x[i]) - a3[i – 1]); //Для нахождения первых трех элементов применяем метод Эйлера

} else {

a3[i] = a3[i - 1] + (h / 12) \* (23 \* (Math.sin(x[i - 1]) - a3[i - 1])

- 16 \* (Math.sin(x[i - 2]) - a3[i - 2]) + 5 \* (Math.sin(x[i - 3]) - a3[i – 3])); //Остальные значения функции находим, используя метод Адамса третьего порядка точности

}

}

return a3;

}

6) функция Adams4(int N, double[] x, double h)

public static double[] Adams4(int N, double[] x, double h) {

double[] a4 = new double[N];

a4[0] = 2; //Задаем начальное значение функции, согласно условию

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

if (i <= 3) {

a4[i] = a4[i - 1] + (h \* (Math.sin(x[i - 1] - a4[i – 1]))); //Для нахождения первых трех элементов применяем метод Эйлера

} else {

a4[i] = a4[i - 1] + ((h / 24) \* (55 \* (Math.sin(x[i - 1]) - a4[i - 1]) - 59 \* (Math.sin(x[i - 2]) - a4[i - 2])

+ 37 \* (Math.sin(x[i - 3]) - a4[i - 3]) - 9 \* (Math.sin(x[i - 4]) - a4[i – 4]))); //Остальные значения функции находим, используя метод Адамса третьего порядка точности

}

}

return a4;

}