МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**«Челябинский государственный университет»**

**(ФГБОУ ВО «ЧелГУ»)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Математический факультет\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование факультета (института, филиала))

\_\_\_\_\_\_\_\_\_\_кафедра вычислительной математики\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование кафедры)

**КУРСОВАЯ РАБОТА**

Метод Адамса для решения дифференциальных уравнений с заданным начальным условием

(тема)

Выполнил студент\_\_\_Бучнев Д.М.\_\_\_\_\_\_

(Ф.И.О.)

группы\_МП-202\_

очной/заочной формы обучения

направления подготовки (специальности)

Прикладная математика и информатика\_\_\_\_\_\_\_\_\_\_\_\_

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

(подпись)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

|  |  |
| --- | --- |
|  | Научный руководитель  Фамилия, имя, отчество\_Соколинская И.М.  Должность\_\_доцент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Ученая степень \_\_кандидат физ-мат наук\_\_  Ученое звание \_доцент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_\_\_г. |

Челябинск

20\_\_\_

# Содержание:

[1. Введение 3](#_Toc74785859)

[2. Методы Адамса 4](#_Toc74785860)

[3. Метод Рунге-Кутты 6](#_Toc74785861)

[4. Применение методов Адамса к решению данного дифференциального уравнения 8](#_Toc74785862)

[5. Полученные результаты 9](#_Toc74785863)

[6. Заключение 11](#_Toc74785864)

[7. Список литературы 12](#_Toc74785865)

[8. Приложение 13](#_Toc74785866)

# Введение

Обыкновенным дифференциальным уравнением порядка r называется уравнение

которое связывает независимую переменную *x*, искомую функцию *y =* и ее производные .

Решение (интегрирование) дифференциального уравнения (1) заключается в отыскании функций (решений, интегралов) , которые удовлетворяют этому уравнению для всех значений *x* в определенном конечном или бесконечном интервале *(a, b)*.

Общее решение обыкновенного дифференциального уравнения порядка *r* имеет вид

где — произвольные постоянные, частный выбор которых дает частное решение.

В задаче Коши (начальной задаче) требуется найти частное решение, удовлетворяющее *r* начальным условиям

по которым определяются *r* постоянных .

Численные методы – это алгоритмы вычисления приближенных значений искомой функции на некотором конечном множестве точек. Решение при этом получается в виде таблицы. Численные методы не позволяют найти общее решение уравнения (1). С их помощью можно определить лишь частное решение задач (1) или (3), но они применимы к широким классам уравнений и всем типам задач.

# Методы Адамса

Пусть на отрезке  задано дифференциальное уравнение

с начальным условием

.

Будем искать значения приближенного решения этой задачи лишь в фиксированных точках данного отрезка. Выбранные узловые точки будем считать равноотстоящими:

Также каким-либо образом найдены также значения . Нужно найти приближенное решение задачи (4), (5) в следующих точках: Приближенные значения решения задачи могут быть вычислены методами Рунге-Кутты либо Эйлера. Это «начало решения» должно быть вычислено с большей точностью.

Расчетное правило экстраполяционного метода Адамса имеет вид

где – конечные разности *k*-гопорядка,

и для остаточного члена имеет место следующая оценка:

где

В экстраполяционном методе Адамса убывание абсолютных величин слагаемых происходит главным образом за счет убывания абсолютных величин конечных разностей.

Рассмотрим алгоритм интерполяционного метода Адамса

где

Для остаточного члена имеет место следующая оценка:

где

Использование вычислительной схемы интерполяционного метода Адамса связано со следующей особенностью. Так как в равенстве (7) величина входит в правую часть, начиная со второго члена, то имеем уравнение для нахождения этой величины. В практике используют обычно такой прием: по формулам (6) экстраполяционного метода Адамса вычисляют («предсказанное» значение), затем это значение применяют в правой части формулы (7) для нахождения («коррекционное значение), далее это «коррекционное» значение можно снова уточнить по формуле (7), пока не получится требуемая точность.

# Метод Рунге-Кутты

Рассмотрим задачу Коши на отрезке для дифференциального уравнения

с начальным условием

.

Будем искать значения приближенного решения этой задачи лишь в фиксированных точках данного отрезка. Выбранные узловые точки будем считать равноотстоящими:

Метод Рунге-Кутты – одношаговый метод решения задачи (8), (9), т.е. такой метод, который позволяет найти приближенное значение решения заданной задачи в узле по информации об этом решении лишь в одной предыдущей узловой точке . Обозначим через приближенное значение искомого решения в точке .

Рассмотрим метод типа Рунге-Кутты четвертого порядка точности, который является одним из распространенных методов решения задач с начальными условиями для обыкновенных дифференциальных уравнений. Данный метод описывается следующими шестью соотношениями:

(10)

(11)

(12)

Порядком (или степенью) точности метода типа Рунге-Кутты называют такое число s, для которого погрешность приближенного равенства будет величиной порядка . Для нашего метода погрешность будет величиной порядка .

В практике для контроля вычислений применяют двойной пересчет, т.е. сначала вычисляют решение с шагом *h*, затем с шагом *h*/2. В заданных точках приближенное решение должно совпадать в пределах заданной точности.

# Применение методов Адамса к решению данного дифференциального уравнения

Методами Адамса продолжить на несколько шагов таблицу значений решений уравнения . «Начальный отрезок» найти методом Рунге-Кутты.

Решение: Значения в точках были вычислены с помощью метода Рунге-Кутты (12). Решение сначала вычислено по экстраполяционной формуле (6), затем уточнено по интерполяционной формуле (7). Результаты вычислений приведены в табл.1.

Порядок заполнения таблицы:

1)записываем в таблицу значения x=1,0; 1,1; …; 1,5 и соответствующие им значения , которые вычислены по методу Рунге-Кутты; находим , и составляем таблицу конечных разностей. В нее заносим только значащие цифры конечной разности, чтобы не загромождать её нулями. Например, если , то заносим 309678, а если используем конечную разность в вычислениях, добавляем недостающие нули, например если , то в вычитаниях применяется ;

2)по формуле (6) при находим , вычисляем ;

3)по и находим , записываем результат и вычисляем ;

4)по формуле (7) при вычисляем скорректированное значение ;

5)переходим к следующему шагу.

В процессе вычислений используем заново полученные конечные разности

# Полученные результаты

Мной была написана программа на языке C++, реализующая метод Адамса, а также метод Рунге-Кутты для решения линейных дифференциальных уравнений с заданным начальным условием.

Решение дифференциального уравнения методом Адамса.

Уравнение (13)

Решение методом Адамса:

Табл.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| k |  |  |  |  |  |  |  |  |
| 0 | 0,0 | -1,000000 | 0,099833 | -0,100000 | -149376 | -97925 | 2471 | 954 |
| 1 | 0,1 | -0,900167 | 0,098836 | 0,0995004 | -247301 | -95454 | 3425 | 920 |
| 2 | 0,2 | -0,801331 | 0,096851 | 0,0980067 | -342755 | -92029 | 4344 | 876 |
| 3 | 0,3 | -0,704480 | 0,093898 | 0,0955337 | -434784 | -87669 | 5220 | 824 |
| 4 | 0,4 | -0,610582 | 0,090007 | 0,0921061 | -522470 | -82465 | 6044 |  |
| 5 | 0,5 | -0,520575 | 0,085217 | 0,0877583 | -604934 | -76421 |  |  |
| 6 | 0,6 | -0,435358 | 0,079575 | 0.0825336 | -681355 |  |  |  |
| 6 | 0,6 | -0,435358 | 0,079575 | 0.0825336 | -681355 |  |  |  |
| 7 | 0,7 | -0,355782 |  | 0.0764842 |  |  |  |  |
| 7 | 0,7 | -0,355782 |  | 0.0764842 |  |  |  |  |

Проверим насколько сильно приближения отклоняются от аналитического решения уравнения (13) .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *k* |  |  |  |  |
| 1 | 0,1 | -0,9001666127 | -0,9001665834 |  |
| 2 | 0,2 | -0,8013307269 | -0,8013306692 |  |
| 3 | 0,3 | -0,7044798747 | -0,7044797933 |  |
| 4 | 0,4 | -0,6105817548 | -0,6105816577 |  |
| 5 | 0,5 | -0,5205745637 | -0,5205744614 |  |
| 6 | 0,6 | -0,4353576314 | -0,4353575266 |  |
| 7 | 0,7 | -0,3557824226 | -0,3557823128 |  |

Как видим, разница между аналитическим решением и приближениями методом Адамса меньше, чем оценка остаточного члена .

# Заключение

Метод Адамса позволяет найти приближенные значения искомой функции с довольно высокой точностью.

Мною была решена задача нахождения на некотором множестве точек приближенных значений искомой функции, являющейся решением линейного дифференциального уравнения с заданным начальным условием, с помощью метода Адамса и написана программа на языке С++, реализующая этот метод. Программа является универсальной и может быть использована для нахождения решений любого линейного дифференциального уравнения.

# Список литературы

1) Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы: Учебное пособие. – М.: Наука, 1987.

2) Калиткин Н.Н. Численные методы. – М.: Наука, 1978.

3) Сборник задач по методам вычислений. Под ред. П.И. Монастырного – Минск: издательство БГУ, 1983.

4) Медведева Н.Б., Рязанов К.А. Численные методы: методические указания к лабораторным работам. – Челябинск: издательство ЧГУ, 1998.

5) Русина Л.Г. Вычислительная математика. Численные методы интегрирования и решения дифференциальных уравнений и систем. Издательство «Лань», 2021.

# Приложение

Программа на языке С++, реализующая метод Адамса для решения линейных дифференциальных уравнений с заданным начальным условием.

#include<cmath>

#include<iostream>

//Метод Рунге-Кутта четвертого порядка точности для нахождения значений приближенного решения задачи Коши с начальным значением y0 на отрезке [x0, X] с шагом h

std::vector<double> methodRungeKutta(double y0, double x0, double X, double h, func f){

auto yi {y0},

xi {x0};

std::vector<double> Values;

Values.push\_back(yi);

while(xi < X){

auto K1 {h \* f(xi, yi)};

auto K2 {h \* f(xi + h / 2, yi + K1 / 2)};

auto K3 {h \* f(xi + h / 2, yi + K2 / 2)};

auto K4 {h \* f(xi + h, yi + K3)};

auto DeltaY {(K1 + 2 \* K2 + 2 \* K3 + K4) / 6};

yi += DeltaY;

xi += h;

Values.push\_back(yi);

}

return Values;

}

//Интерполяционный метод Адамса четвертого порядка точности для вычисления коррекционного значения yn с заданной точностью eps

void correctionMethodAdams(std::vector<double> &Values, const std::vector<double> &X, double h, double eps, func f){

std::vector<double> Eta(5); //Для хранения значений

std::vector<double> Delta1(4); //Для хранения значений

std::vector<double> Delta2(3); //Для хранения значений

std::vector<double> Delta3(2); //Для хранения значений

auto EtaSize {Eta.size()}, //Чтобы не вызывать метод size()

Delta1Size {Delta1.size()}, //несколько раз в циклах for

Delta2Size {Delta2.size()}, //заранее сохраняю их значения

Delta3Size {Delta3.size()};

auto N {Values.size()};

auto flag {false};

double Delta4 {0.0}; //Для хранения значения

while(!flag){

for(auto i {0}; i < EtaSize; ++i){

Eta[i] = h \* f(X[N - i - 1], Values[N - i - 1]);

// std::cout << "Cor Eta[" << N - i - 1 << "] = " << Eta[i];

// std::cout << std::endl;

}

for(auto i {0}; i < Delta1Size; ++i){

Delta1[i] = Eta[i] - Eta[i + 1];

// std::cout << "Cor Delta1[" << Delta1Size - i - 1 << "] = ";

// std::cout << Delta1[i] << std::endl;

}

for(auto i {0}; i < Delta2Size; ++i){

Delta2[i] = Delta1[i] - Delta1[i + 1];

// std::cout << "Cor Delta2[" << Delta2Size - i - 1 << "] = ";

// std::cout << Delta2[i] << std::endl;

}

for(auto i {0}; i < Delta3Size; ++i){

Delta3[i] = Delta2[i] - Delta2[i + 1];

// std::cout << "Cor Delta3[" << Delta3Size - i - 1 << "] = " << Delta3[i] << std::endl;

}

Delta4 = Delta3[0] - Delta3[1];

// std::cout << "Cor Delta4 = " << Delta4 << std::endl;

auto yn {Values[N - 2] + Eta[0] - Delta1[0] / 2 - Delta2[0] / 12 - Delta3[0] / 24 - 19 \* Delta4 / 720};

flag = std::abs(Values[N - 1] - yn) < eps;

Values[N - 1] = yn;

// printf("Corrected Values[%d] = %f\n", N - 1, Values[N - 1]);

}

}

//Экстраполяционный метод Адамса четвертого порядка точности для нахождения значений приближенного решения задачи Коши с начальным значением и найденными значениями (i = 1, ..., n) по методу Рунге-Кутты

void methodAdams(std::vector<double> &Values, double x0, double xn, double b, double h, func f){

std::vector<double> X(Values.size());//Для хранения значений [x0, b]

std::vector<double> Eta(5); //Для хранения значений

std::vector<double> Delta1(4); //Для хранения значений

std::vector<double> Delta2(3); //Для хранения значений

std::vector<double> Delta3(2); //Для хранения значений

auto EtaSize {Eta.size()}, //Чтобы не вызывать метод size()

Delta1Size {Delta1.size()}, //несколько раз в циклах for

Delta2Size {Delta2.size()}, //заранее сохраняю их значения

Delta3Size {Delta3.size()};

double Delta4 {0.}; //Для хранения значения

auto N {Values.size()};

auto xi {x0};

for(auto i {0}; i < N; ++i){

X[i] = xi;

xi += h;

}

xi -= h;

while(xi < b){

for(auto i {0}; i < EtaSize; ++i){

Eta[i] = h \* f(X[N - i - 1], Values[N - i - 1]);

// std::cout << "Eta[" << N - i - 1 << "] = " << Eta[i] << std::endl;

}

for(auto i {0}; i < Delta1Size; ++i){

Delta1[i] = Eta[i] - Eta[i + 1];

// std::cout << "Delta1[" << Delta1Size - i - 1 << "] = ";

// std::cout << Delta1[i] << std::endl;

}

for(auto i {0}; i < Delta2Size; ++i){

Delta2[i] = Delta1[i] - Delta1[i + 1];

// std::cout << "Delta2[" << Delta2Size - i - 1 << "] = ";

// std::cout << Delta2[i] << std::endl;

}

for(auto i {0}; i < Delta3Size; ++i){

Delta3[i] = Delta2[i] - Delta2[i + 1];

// std::cout << "Delta3[" << Delta3Size - i - 1 << "] = " ;

// std::cout << Delta3[i]<< std::endl;

}

Delta4 = Delta3[0] - Delta3[1];

// std::cout << "Delta4 = " << Delta4 << std::endl;

Values.push\_back(Values[N - 1] + Eta[0] + Delta1[0] / 2 + 5 \* Delta2[0] / 12 + 3 \* Delta3[0] / 8 + 251 \* Delta4 / 720);

xi += h;

X.push\_back(xi);

N++;

// printf("Values[%d] = %.6f\n", N - 1,Values[N - 1]);

//так как остаточный член оценивается как ,

//то возьмем eps за эту величину

correctionMethodAdams(Values, X, h, h \* h \* h \* h \* h \* h \* 95 / 283, f);

}

}