

**МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО
ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН**

САМАРКАНДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени Алишера Навои

Механика – математический факультет

КАФЕДРА «МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ»

На правах рукописи

УДК

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Для получения академической степени магистра по специальности

«5А _____ – вычислительная математика»

СИНДОРОВА ФАЗЛИДДИНА РАББИМОВИЧА

На тему:

**«СОСТАВЛЕНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ
НЕСТАЦИОНАРНЫХ РЕЖИМОВ ХИМИКО –
ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ И РЕШЕНИЕ ЕГО
ЧИСЛЕННЫМИ МЕТОДАМИ»**

Работа рассмотрена и допущена к защите:

Декан факультета:

проф. А.Солеев

Заведующий кафедрой:

доц. Э.Уринбоев

Научный руководитель:

доц. С.Амридинов

**A.Navoiy nomidagi SamDU Mexanika-matematika fakulteti "Hisoblash usullari" kafedrasining 2014 yil «4» iyundagi majlisi bayonnomasidan
KO'CHIRMA**

Qatnashdilar: Kafedra a'zolari, magistrant va bakalavr talabalar.

Kun tartibi:

1. Magistrlik dissertasiya ishining dastlabki himoyasi.

Eshitildi: Majlisni kafedra mudiri dots. A. Abdirashidov olib bordi, kun tartibi bilan tanishtirdi va magistr'larga so'z berildi.

Magistr: **Sindorov Fazliddin Rabbimovich**

Mavzu: "Ximiya texnologiya nostatsionar jarayonlarning matematik modelini tuzish va uni hisoblash matematikasi usullari bilan yechish"

Ma'ruzachiga quyidagi savollar berildi:

- Matematik modellashtirishning asosiy g'oyasi nima?
- Nostatsionar jarayonlarni matematik tavsifi nima?
- Reaktorlarni ishlash jarayonini matematik ifodasi qanday bo'ladi?

Ma'ruzachi savollarga qoniqarli javob berdi.

Taqrizchi: TATU Samarqand filiali dekani dotsent A. Qarshiyev nomzodi tavsiya etildi.

Taklif ovozga qo'yildi va bir ovozdan tasdiqlandi.

Qaror qilindi:

1. "Matematik modellashtirish" kafedrasida magistrant Sindorov Fazliddin tomonidan bajarilgan "Ximiya texnologiya nostatsionar jarayonlarning matematik modelini tuzish va uni hisoblash matematikasi usullari bilan yechish" mavzusidagi magistrlik dissertasiya ishining dastlabki himoyasi qoniqarli deb topilsin.
2. "Matematik modellashtirish" kafedrasida magistrant Sindorov Fazliddin tomonidan bajarilgan "Ximiya texnologiya nostatsionar jarayonlarning matematik modelini tuzish va uni hisoblash matematikasi usullari bilan yechish" mavzusidagi magistrlik dissertasiya ishiga taqrizchi etib TATU Samarqand filiali dekani dotsent A. Qarshiyev nomzodi tavsiya etilsin.
3. "Matematik modellashtirish" kafedrasida magistrant Sindorov Fazliddin tomonidan bajarilgan "Ximiya texnologiya nostatsionar jarayonlarning matematik modelini tuzish va uni hisoblash matematikasi usullari bilan yechish" mavzusidagi magistrlik dissertasiya ishi Yakuniy Davlat Nazorati Komissiyasiga himoya uchun taqdim qilinsin.

Ko'chirma asliga to'g'ri.

Kafedra majlisi kotibasi:

G.M.To'ychiyeva

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ОПТИМАЛЬНЫЕ АЛГОРИТМЫ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ.	5
1.1. Математическое и физическое моделирование и задачи, приводящих к ним	5
1.2. Методы численного интегрирования дифференциальных уравнений	7
1.3. Алгоритм и блок схема и программа метода Эйлера и Рунге-Кутта численного интегрирования дифференциальных уравнений	18
2. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ НЕСТАЦИОНАРНЫХ РЕЖИМОВ..	33
2.1. Математическая модель нестационарных режимов тарельчатой ректификационной колонны.	33
2.2. Математическая модель реактора идеального вытеснения и его численное решение.	36
3. РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ ХИМИИ И ХИМИЧЕСКОЙ ТЕХНОЛОГИИ.	40
3.1. Применение метода Эйлера и Рунге-Кутта для решения задач химии и химических технологии.	40
4. ЗАКЛЮЧЕНИЕ.	57
5. ЛИТЕРАТУРА.	60

ВВЕДЕНИЕ

Постановка задачи. Развитие науки техники тесно связаны с развитием компьютерных технологий и их применением для решения практических задач. Это особенно наглядно применяется при решении задач химии и химических технологий методами вычислительной математики. На основе вышеизложенного разработаны оптимальные алгоритмы численных методов и их применения в задачах химии и химических технологий и сопоставление полученных результатов с экспериментальными данными.

Актуальность задачи. Применение вычислительных методов с использованием современных компьютерной технологий при решении задач химии и химических технологий является самым перспективным направлением. Не проводя лабораторных экспериментов можно получить численные результаты с заданной степенью точности на компьютере.

Основная цель и задачи. Основной целью магистерской диссертационной работы является глубокое изучение численных методов и подбор оптимального метода для решения конкретной задачи химии и химических технологий. Составление оптимальной математической модели решаемой задачи, отражающий производственный процесс химии и химических технологий.

Методы исследования. Диссертационная работа посвящена математическому моделированию химико-технологических процессов и подбору оптимального метода решения поставленной задачи. В качестве метода исследования приняты: метод Эйлера, усовершенствованный метод Эйлера, метод Рунге-Кутты.

Составлены оптимальные алгоритмы, блок-схемы и программы решаемых задач. Полученные результаты сопоставлены с экспериментальными данными.

Научная ценность работы. Магистерская диссертационная работа посвящена одному из актуальных вопросов, т.е. применению методов

вычислительной математики при решении задач химии и химических технологий.

В качестве объекта исследования рассматривается работа реакторов, которая сводится и решению обыкновенных дифференциальных уравнений. Рассматриваемые задачи решаются методами Эйлера и Рунге-Кутты, составлены универсальные алгоритмы и программы.

Практическая ценность задачи. Составленные оптимальные алгоритмы, блок-схемы и программы можно успешно использовать при решении задач химии и химических технологий. При произвольных исходных данных решены конкретные задачи и полученные результаты сопоставлены с экспериментальными данными.

Следует отметить, что не проводя эксперимента, который требует значительных затрат, можно при помощи составленных математических моделей получить результаты с заданной точностью.

Основная содержание работы. Основная задача, поставленная в магистерской диссертации, это составление математических моделей реальных химико-технологических процессов и решение их методами вычислительной математики, с помощью современных компьютерных технологий. Сравнение полученных результатов с экспериментальными данными и разработка рекомендации для решения практических задач.

Аннотация. В магистерской диссертационной работе разрабатываются математические модели химико-технологических процессов и решение их численными методами с заданной степенью точности. Полученные результаты сравниваются с экспериментальными данными.

ГЛАВА I. ОПТИМАЛЬНЫЕ АЛГОРИТМЫ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ.

1.1 Математическое и физическое моделирование и задачи, приводящие к ним

Ключ к решению многих проблем дает метод математического моделирования, базирующийся на стратегии системного анализа, сущность которого заключается в представлении процесса как сложной взаимодействующей иерархической системы с последующим качественным анализом ее структуры, разработкой математического описания и оценкой неизвестных параметров.

Под математическим моделированием понимают изучение свойств объекта на математической модели. Математической моделью называется приближенное описание, какого — либо явления или процесса внешнего мира, выражаемого с помощью математической символики.

Математическое моделирование включает в себя три взаимодействующих этапа:

1. Составление математического описания изучаемого объекта.
2. Выбор метода решения для системы уравнений математического описания и реализация его в форме моделирующей программы.
3. Установление соответствия (адекватности) модели объекту.

Исходя из временного и пространственного признаков, в зависимости от конкретной реализации процесса и его аппаратного оформления все многообразие моделей можно разделить на 4 класса:

- 1) модели с сосредоточенными параметрами;
- 2) модели с распределенными параметрами;
- 3) статические модели;
- 4) динамические модели.

Математическая модель является системой уравнений математического описания, отражающей сущность протекающих в объекте явлений, для которой определен алгоритм решения, реализуемый в форме моделирующей программы.

Математическая модель должна рассматриваться в совокупности 3-х аспектов:

- 1) смысловой аспект представляет собой физическое описание природы модулируемого объекта.
- 2) аналитический аспект является математическим описанием процесса в виде некоторой системы уравнений.
- 3) вычислительный аспект есть метод и алгоритм решения системы уравнений математического описания.

При физическом моделировании, как правило, используют модели, отличающиеся от объекта моделирования масштабами. Процессы в физически подобных объектах имеют одинаковую физическую природу.

Физическое моделирование основано на использовании принципа подобия. Он позволяет из класса явлений, описываемых дифференциальными уравнениями выделить при помощи приведения к безразмерному виду группу взаимно — подобных явлений.

При физическом моделировании данное явление изучается на нем самом, воспроизводя его в разных масштабах и анализируя влияние физических особенностей и линейных размеров. Эксперимент производят непосредственно на изучаемом физическом процессе.

Основываясь на принципе подобия, можно сформулировать следующие требования:

- 1) Необходимо поставить условия однозначности процесса;
- 2) В опытах должны исследоваться зависимости между параметрами, входящими в условия однозначности;
- 3) Из условий однозначности следует составить критерии подобия, характеризующие процесс;
- 4) Ход явлений нужно описать критериальными уравнениями, определив экспериментально значение коэффициентов и показателей степени при критериях;
- 5) Переноса результаты опытов на условия, отличающиеся от экспериментальных, необходимо соблюдать подобие условий однозначности и

численное равенство критериев подобия, составленных на основе условий однозначности.

При правильном планировании эксперимента должны выполняться два первых требования. При масштабировании процесса особенно важны требования 3-5. Теоретически существует другая возможность использования экспериментальных результатов: если ход исследуемого явления удастся описать в виде системы уравнений, решая ее для новых условий, можно определить ход явления в этих условиях. Отсюда следует, что метод подобия имеет большое значение, хотя все чаще удается решать сложные системы уравнений благодаря использованию ЭВМ. [1,2]

1.2 Методы численного интегрирования дифференциальных уравнений

Рассмотрим дифференциальное уравнение первого порядка

$$y' = f(x, y). \quad (2.1)$$

Функцию $f(x, y)$ будем рассматривать заданной в некоторой области изменения ее аргументов $a \leq x \leq b, c \leq y \leq d$, причем переменные x, y можно рассматривать как декартовы координаты произвольной точки A области определения G (рис 1.)

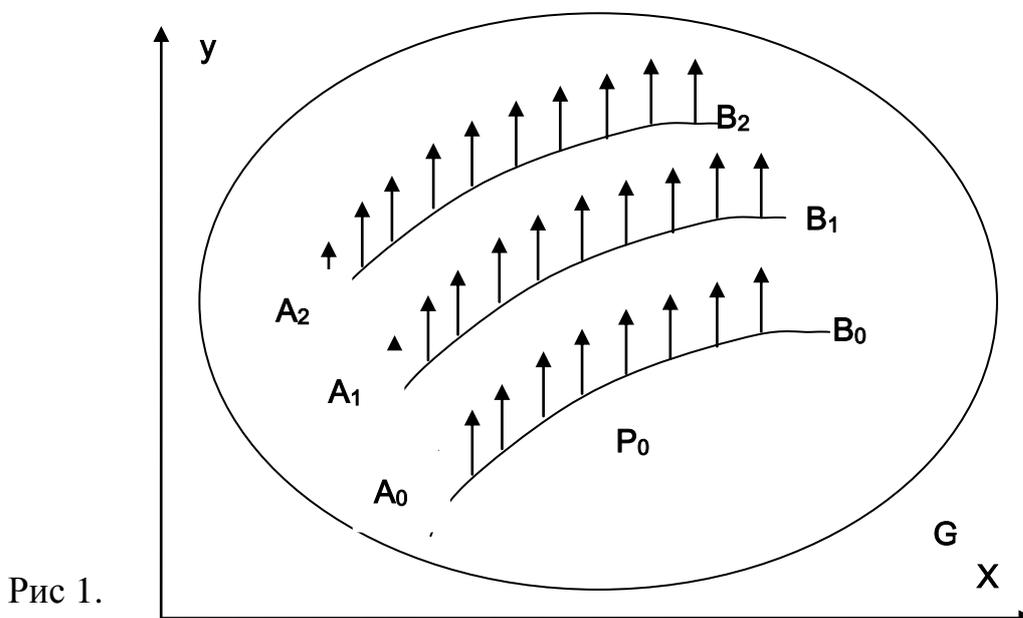


Рис 1.

Дифференциальное уравнение (1) устанавливает в каждой точке A

связь между координатами и производной от функции y в этой точке. Т.о. для любой точки области G по уравнению (1) можно вычислить производную, т.е. тангенс угла наклона кривой $f(x, y)$. Выбирая достаточно большое число точек в области G и вычисляя в каждой точке угол наклона, можно затем соединить точки, имеющие один и тот же угол наклона, некоторыми кривыми, которые называются изоклинами дифференциального уравнения. Любая функция $y(x)$, у которой в каждой точке области производная определяет направление, совпадающее с направлением, задаваемым для изоклины, проходящий через эту точку (кривая P_0, P_1, P_2, P_3) является решением уравнения (1). На этом свойстве изоклин основаны некоторые графо-аналитические методы интегрирования дифференциальных уравнений. Число кривых решения, направление касательных, к которым в каждой точке совпадает с полем направлений изоклин, вообще говоря, бесконечно, что непосредственно следует из рис. 1. Для того, чтобы выбрать из бесконечного числа решений единственное, необходимо зафиксировать некоторую начальную точку, через которую должно проходить решение. Это эквивалентно заданию дополнительного условия, накладываемого на решение дифференциального уравнения и называемого начальным условием.

Наряду с одним дифференциальным уравнением во многих теоретических и практических задачах используются также и системы дифференциальных уравнений. Система обыкновенных дифференциальных уравнений имеет столько уравнений, сколько в нее входит неизвестных функций, причем все неизвестные функции являются функциями одной независимой переменной. Для систем уравнений в частных производных число независимых переменных больше единицы, но число уравнений также равно числу неизвестных функций. При решении системы дифференциальных уравнений имеют важное значение, поскольку любое уравнение порядка выше первого может быть путем замены переменных преобразовано в систему уравнений первого порядка. Действительно, если имеется уравнение

$$y^{(n)} = f(x, y, y', y'', \dots, y^{(n-1)}), \quad (2.2)$$

то, полагая его можно записать в виде системы уравнений

$$\left. \begin{aligned} y' &= y_1 \\ y_1' &= y_2 \\ \dots & \dots \\ y_{n-2}' &= y_{n-1} \\ y_{n-1}' &= f(x, y, y_1, y_2, \dots, y_{n-1}) \end{aligned} \right\}$$

или воспользовавшись матричными обозначениями,

$$y' = f(x, y) \quad (2.3)$$

где y' и y - вектор функции.

Решением уравнения (2.3) будет вектор-функция y , определяющая некоторую линию в $n + 1$ -мерном пространстве, в котором начальное условие изображается как точка. Методы решения одного дифференциального уравнения первого порядка с начальным условием часто можно распространить и на системы уравнений первого порядка, а следовательно, на уравнения более высокого порядка. Поэтому в дальнейшем будут рассматриваться методы решения применительно только к уравнениям 1-го порядка.

Пусть для заданного интервала изменения аргумента требуется вычислить ряд значений функций, $y = y(x)$ являющийся решением уравнения (2.1), если известно начальное условие $y_0 = y(x_0)$,

Предположим, что правая часть уравнения (2.1) является аналитической функцией в области определения решения, тогда решение дифференциального уравнения является аналитическим и допускает разложение в ряд Тейлора. Разложим интегральную кривую уравнения в окрестности точки x_k в ряд Тейлора:

$$y(x_{k+1}) = y(x_k) + hy'(x_k) + \frac{h^2}{2!} y''(x_k) + \dots + \frac{h^m}{m!} y^{(m)}(x_k) + \frac{h^{m+1}}{(m+1)!} y^{(m+1)}(x_k) \quad (2.4)$$

где $h = x_{k+1} - x_k$ и $x_k \leq x \leq x_{k+1}$ величина h называется обычно шагом интегрирования.

Поскольку производная $y^{(m+1)}$ непрерывная и ограниченная,

шаг интегрирования можно выбрать настолько малым, что последним членом ряда (2.4) можно пренебречь. Поскольку $y_{k+1} = y(x_{k+1})$ выражение (2.4) запишется в виде

$$y(x_{k+1}) = y(x_k) + hy'_k + \frac{h^2}{2!} y'' + \dots + \frac{h^m}{m!} y^{(m)} \quad (2.5)$$

Выражение (1.5) является основной формулой интегрирования дифференциального уравнения (1.1) путем разложения решения в ряд Тейлора.

Полагая в ней последовательно $k = 0, 1, 2, \dots$, $n = \frac{b-a}{h}$ можно вычислить решение уравнения (2.1) в точках интервала (a, b) . Очевидно, чем больше членов разложения и чем меньше шаг интегрирования, тем точнее будет получено решение. Производные, входящие в формулу (2.5), вычисляются непосредственным дифференцированием уравнения (2.1). Поскольку первая производная равна

$$y'_k = f(x_k, y_k) \quad (2.6)$$

то выражение для второй производной может быть получено дифференцированием (2.6) по x :

$$y''_k = \frac{\partial}{\partial x} f(x_k, y_k) + f(x_k, y_k) \frac{\partial}{\partial y} f(x_k, y_k)$$

или

$$y''_k = f_x + ff_y, \quad (2.7)$$

где

$$f_x = \frac{\partial}{\partial x} f(x_k, y_k), \quad f_y = \frac{\partial}{\partial y} f(x_k, y_k), \quad f = f(x_k, y_k).$$

Аналогично выражение для третьей производной получается в результате дифференцирования выражения (2.7).

$$y'''_k = f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_xf_y + ff^2_y. \quad (2.8)$$

Если выражения (2.6) – (2.8) подставить в (2.5), то для вычисления y в точке $x = x_{k+1}$ получи

$$y_{k+1} = y_k + h \left(f + \frac{h}{2!} (f_x + f_y f) + \frac{h^2}{6} (f_{xx} + 2f_{xy} f + f^2 f_{yy} + f_{yx} + f f^2_y) \right) + O(h^4), \quad (2.9)$$

где $O(h^4)$ - остаточный член ряда, определяемый членами выше третьего порядка малости по h .

Принципиально формула - (2.5) может быть использована при интегрировании любого дифференциального уравнения с произвольной наперед заданной точностью, от которой будет зависеть число членов ряда. Однако с увеличением числа членов ряда увеличивается количество подлежащих определению производных, а следовательно, и объем вычислений. Вычисление производных с практической точки зрения весьма трудоемко, поэтому формулы разложения решения в ряд как метод решения дифференциальных уравнений не получили широкого распространения.

Обычно вместо разложения используются методы, опирающиеся на разложение в ряд Тейлора, но позволяющие получить решение без вычисления производных. Метод же отыскания решения с помощью рядов Тейлора главным образом используется как способ оценки точности других формул интегрирования.[1]

Метод Рунге-Кутты для решения системы обычных дифференциальных уравнений и применение ей в задачах химии и химических технологий.

Сначала приведем этот метод для одного уравнения. Наиболее распространенными в практике интегрирования обыкновенных дифференциальных уравнений являются формулы Рунге-Кутты. Эти формулы классифицируются по степени приближения их по точности к разложению решения в ряд Тейлора. Формулы, точные до второго, третьего, четвертого и т.д. членов разложения, носят название формул второго, третьего, четвертого и т.д. порядка соответственно. Достоинством формул Рунге-Кутты является то, что при их использовании не нужно вычислять производные выше первого порядка, а их основной недостаток – громоздкость и значительный объем вычислений на каждом шаге.

$\varphi_1'(0) = y_m' - p_{1,1} f(x_m, y_m)$. Условие (2.13) для $\varphi_1'(0)$ будет выполнено лишь в том случае, если $p_{1,1} = 1$, и, таким образом, при $c = 1$ расчетная формула

$$y_{m+1} = y_m + hf(x_m, y_m) \quad (2.14)$$

т.е. получила вид формулы Эйлера.

Пусть $c = 2$. Тогда

$$\begin{aligned} \xi_2 &= x_m + \alpha_2 h, & \xi_1 &= x_m + \alpha_1 h = x_m, \\ \eta_2 &= y_m + b_{2,1} k_1, & \eta_1 &= y_m. \end{aligned}$$

Поскольку

$$k_1 = hf(x_m, y_m), \quad k_2 = hf(x_m + \alpha_2 h, y_m + b_{2,1} k_1),$$

то

$$\begin{aligned} y_{m+1} &= y_m + [p_{2,1} k_1(h) + p_{2,2} k_2(h)] = \\ &= y_m + hp_{2,1} f(x_m, y_m) + hp_{2,2} [f(x_m + \alpha_2 h, y_m + b_{2,1} hf(x_m, y_m))]. \end{aligned} \quad (2.15)$$

Формула (2.15) содержит четыре неизвестных коэффициента - $p_{2,1}$, $p_{2,2}$, α_2 и $b_{2,1}$, которые необходимо определить исходя из условий (2.13). Для того чтобы найти значения этих коэффициентов, нужно записать выражения для $\varphi_2'(h)$ и $\varphi_2''(h)$ и положить $h = 0$. Тогда неизвестные коэффициенты могут быть найдены решением системы линейных уравнений.

Выражение для $\varphi_2(h)$ имеет вид

$$\varphi_2(h) = y_{m+1} - y_m - [p_{2,1} k_1(h) + p_{2,2} k_2(h)] \quad (2.16)$$

Вычислим $\varphi_2'(h)$

$$\varphi_2'(h) = y_{m+1}' - [p_{2,1} k_1'(h) + p_{2,2} k_2'(h)], \quad (2.17)$$

где

$$k_1'(h) = f(x_m, y_m) = f, \quad k_2'(h) = f(\xi_2, \eta_2) + h[\alpha_2 f_x + b_{2,1} ff_y].$$

Для $\varphi_2'(0)$ можно записать $\varphi_2'(0) = f - [p_{2,1} f + p_{2,2} f]$, где $f = f(x_m, y_m)$, откуда следует, что условие для $\varphi_2'(0)$ будет выполнено в том случае, если

$$p_{2,1} + p_{2,2} = 1. \quad (2.18)$$

Аналогично, дифференцируя $\varphi_2'(h)$ по h и учитывая, что $k_1''(h) = 0$, для $\varphi_2''(0)$ получим

$$\varphi_2''(0) = f_x + ff_y - 2p_{2,2}[\alpha_2 f_x + b_{2,1} ff_y],$$

откуда следует, что для $\varphi_2''(0) = 0$ должно быть

$$2p_{2,2}\alpha_2 = 1, \quad 2p_{2,2}b_{2,1} = 1. \quad (2.19)$$

Таким образом, коэффициенты $p_{2,1}$, $p_{2,2}$, α_2 , $b_{2,1}$ формулы Рунге – Кутта второго порядка удовлетворяют системе линейных уравнений

$$\begin{cases} p_{2,1} + p_{2,2} = 1, \\ p_{2,2}\alpha_2 = 1/2, \\ p_{2,2}b_{2,1} = 1/2. \end{cases} \quad (2.20)$$

Поскольку система (2.20) имеет четыре неизвестных, то число ее решений бесконечно. Каждое решение дает формулу интегрирования с ошибкой порядка h^3 .

На практике коэффициенты выбираются такими, чтобы формулы были удобны для вычислений. Например, если положить $\alpha_2 = 1$, то тогда $p_{2,2} = 1/2$; $p_{2,1} = 1/2$; $b_{2,1} = 1$.

Подстановки этих значений в выражение получим формулу

$$y_{m+1} = y_m + 1/2(k_1 + k_2) = y_m + h/2[f(x_m, y_m) + f(x_m + h, y_m + hy_m')],$$

являющуюся формулой усовершенствованного метода Эйлера – Коши.

Если же положить, что $p_{2,1} = 0$, то из (2.20) получим

$$p_{2,2} = 1; \quad b_{2,1} = \alpha_2 = 1/2.$$

Тогда после подстановки полученных значений в (2.15) последнее преобразуется к виду (2.12), т.е. получим формулу модифицированного метода Эйлера. Отсюда следует, что формулы Эйлера является частными случаями формул Рунге-Кутта первого и второго порядков. Наибольшее распространение в вычислительной практике имеют формулы Рунге-Кутта четвертого порядка, которые получаются из общих соотношений (2.11) – (2.13) при $c = 4$.

При этом

$$y_{m+1} = y_m + p_{4,1} k_1 + p_{4,2} k_2 + p_{4,3} k_3 + p_{4,4} k_4, \quad (2.21)$$

где

$$k_1(h) = hf(x_m, y_m),$$

$$k_2(h) = hf(x_m + \alpha_2 h, y_m + b_{2,1} K_1),$$

$$k_3(h) = hf(x_m + \alpha_3 h, y_m + b_{3,1} K_1 + b_{3,2} K_2),$$

$$k_4(h) = hf(x_m + \alpha_4 h, y_m + b_{4,1} K_1 + b_{4,2} K_2 + b_{4,3} K_3),$$

Коэффициенты выражения (2.21) также вычисляются исходя из условий (2.13) теперь уже для производных до четвертого порядка включительно. Точность формул четвертого порядка пропорциональна h^5 , т.е. формулы соответствуют разложению решения в ряд Тейлора до членов h^4 включительно.

При повышении порядка формулы возрастает произвол в выборе коэффициентов. Для формул четвертого порядка уже имеется две степени свободы, что приводит к еще большему разнообразию соотношений.

Примером формулы Рунге – Кутты четвертого порядка может служить соотношение

$$y_{m+1} = y_m + 1/6 (k_1 + 2k_2 + 2k_3 + k_4) + O(h^5). \quad (2.22)$$

где

$$k_1 = hf(x_m, y_m),$$

$$k_2 = hf(x_m + h/2, y_m + 1/2 k_1),$$

$$k_3 = hf(x_m + h/2, y_m + 1/2 k_2),$$

$$k_4 = hf(x_m + h, y_m + k_3).$$

На каждом шаге интегрирования по формулам Рунге – Кутты четвертого порядка необходимо четыре раза вычислять правую часть дифференциального уравнения. Это приведет к увеличению времени счета, однако компенсируется более высокой точностью формулы, в силу чего интегрирование можно вести с большим шагом. При использовании вычислительных машин выбор величины шага интегрирования

производится автоматически в процессе интегрирования. Изменение шага обычно производится по результатам сравнения решений, получаемых на некотором интервале интегрированием с целым и половинным шагом. Если результаты этих двух вычислений совпадают с заданной точностью, то шаг для дальнейшего интегрирования остается прежним или увеличивается, в противном случае уменьшается.

При автоматическом выборе шага интегрирования затраты машинного времени на решение дифференциального уравнения в значительной степени определяются стратегией определения шага, обеспечивающего заданную точность.

Так, для формул

$$y_{m+1} = y_m + 1/2 (k_1 + 4k_4 + k_5), \quad (2.23)$$

где

$$\begin{aligned} k_1 &= 1/3 hf(x_m, y_m), \\ k_2 &= 1/3 hf(x_m + 1/3h, y_m + k_1), \\ k_3 &= 1/3 hf(x_m + 1/3h, y_m + 1/2k_1 + 1/2k_2), \\ k_4 &= 1/3 hf(x_m + 1/2h, y_m + 3/8k_1 + 9/8k_3), \\ k_5 &= 1/3 hf(x_m + h, y_m + 3/2k_1 - 9/2k_3 + 6k_4), \end{aligned} \quad (2.23a)$$

предлагается следующая стратегия оценки точности интегрирования и выбора шага. Если правая часть меньше, чем $5/16 \varepsilon$, то шаг может быть удвоен. По оценке интегрирование по формулам (2.23) на 20% быстрее, чем по формулам (2.22) с постоянным шагом.

Теперь приведём метод Рунге-Кутты для системы дифференциальных уравнений.

Методы интегрирования одного дифференциального уравнения могут быть распространены и на системы обыкновенных дифференциальных уравнений.

Пусть задана система двух уравнений.

$$\begin{cases} y' = f(x, y, z) \\ z' = g(x, y, z) \end{cases} \quad (2.24)$$

с начальными условиями в точке $x = x_0$; $y|_{x=x_0} = y_0$; $z|_{x=x_0} = z_0$;

Для того, чтобы проинтегрировать систему (2.24), например, по формулам Рунге-Кутты 4-го порядка, достаточно применить формулы к каждому из уравнений

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ z_{i+1} &= z_i + \frac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4) \end{aligned} \quad (2.25)$$

где k_i и m_i от, вычисляются по соответствующим формулам

$$k_1 = hf(x_i, y_i, z_i) \quad (2.26)$$

$$k_2 = hf(x_i + h/2, y_i + k_1/2, z_i + m_1/2), \quad (2.27)$$

$$k_3 = hf(x_i + h/2, y_i + k_2/2, z_i + m_2/2), \quad (2.28)$$

$$k_4 = hf(x_i + h/2, y_i + k_3/2, z_i + m_3/2), \quad (2.29)$$

$$m_1 = hf(x_i, y_i, z_i) \quad (2.30)$$

$$m_2 = hf(x_i + h/2, y_i + k_1/2, z_i + m_1/2), \quad (2.31)$$

$$m_3 = hf(x_i + h/2, y_i + k_2/2, z_i + m_2/2), \quad (2.32)$$

$$m_4 = hf(x_i + h/2, y_i + k_3/2, z_i + m_3/2), \quad (2.33)$$

Для того, чтобы проинтегрировать систему (2.24) по формулам Рунге-Кутты 2-го порядка, достаточно применить формулу

$$y_{m+1} = y_m + 1/2(k_1 + k_2) = h/2[f(x_m, y_m) + f(x_m + h, y_m + hy'_m)],$$

к каждому из уравнений

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{2}(k_1 + k_2) \\ z_{i+1} &= z_i + \frac{1}{2}(m_1 + m_2) \end{aligned} \quad (2.34)$$

где

$$\begin{aligned}
k_1 &= hf(x_i, y_i, z_i), \\
k_2 &= hf(x_i + h, y_i + k_1, z_i + m_1/2), \\
m_1 &= hg(x_i, y_i, z_i), \\
m_2 &= hg(x_i + h, y_i + k_1, z_i + m_1)
\end{aligned}$$

Формула (2.34) является формулой усовершенствованного метода Эйлера – Коши для системы двух уравнений. [1,2]

1.3 Алгоритм, блок – схема и программа метода Эйлера, Рунге – Кутта численного интегрирования дифференциального уравнений

Формулы Эйлера. Пусть дано уравнение $y' = f(x, y)$ удовлетворяющее начальному условию $y_0 = y(x_0)$, решением которого является функция $y = y(x)$, определенная на интервале (a, b) (рис. 2). Выберем достаточно малый шаг h и построим систему точек

$$x_k = x_0 + kh \quad (k = 1, 2, \dots, n), \quad (x_0 = a), \quad (n = (b - a) / h).$$

Для интегрирования уравнения $y' = f(x, y)$ воспользуемся формулой

$$y_{k+1} = y_k + hy'_k + \frac{h^2}{2!} y''_k + \frac{h^m}{m!} y_k^{(m)},$$

ограничившись двумя членами ряда

$$y_{k+1} = y_k + hy'_k = y_k + hf(x_k, y_k). \quad (3.1)$$

Интегрирование по методу Эйлера заключается в последовательном применении формулы (1) к уравнению $y' = f(x, y)$ начиная с $k = 1$. При наличии начального условия $y|_{x=x_0} = y_0$ для вычисления y_1 не требуется никакой дополнительной информации: достаточно вычислить правую часть уравнения $y' = f(x, y)$ при заданных значениях аргумента x_0 и функции y_0 . Процедура вычисления y_1 аналогична проведению касательной в точке

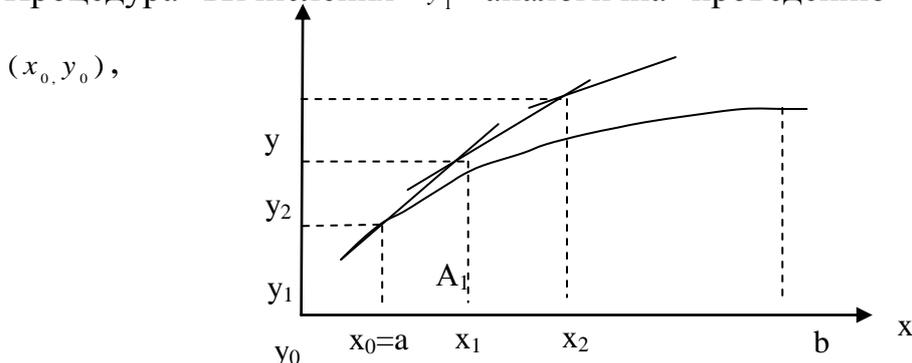


Рис. 2. Графическая иллюстрация метода Эйлера

тангенс угла наклона которой задается правой частью уравнения $y' = f(x, y)$ при $x = x_0$ и $y = y_0$, до пересечения с вертикальной прямой, проведенной из точки $x = x_1$.

На следующем шаге, т. е. при вычислении y_2 , снова определяется производная, но только в точке (x_1, y_1) и из точки пересечения прямых $x = x_1$ и $y = y_1$ (точка A_1 на рис.2) проводится касательная до пересечения с прямой $x = x_2$. Аналогичная процедура вычислений повторяется для y_3, y_4 и т. д.

Таким образом, в результате вычислений определяется некоторая ломаная линия, линейные отрезки которой имеют угол наклона, вычисляемый через производную в соответствующей точке интегральной кривой. Как следует из рис. 2, с ростом k ломаная линия все дальше отходит от истинного решения. Отсюда же из геометрических представлений легко заметить основной недостаток метода Эйлера: если, например, кривая решения выпукла, то ломаная кривая, вычисляемая на каждом шаге, будет отходить от нее вверх, поскольку для вычисления положения последующей точки используется производная в предыдущей. Очевидно, чем больше кривизна интегральной кривой и шаг интегрирования, тем значительнее это отклонение. Другим неприятным свойством этого метода является также то, что ошибка интегрирования накапливается, т.е. увеличивается с каждым шагом.

При интегрировании по методу Эйлера полагается, что производная по x от x_k до x_{k+1} — постоянная величина, что соответствует кусочно-линейному представлению интегральной кривой на отдельных участках интегрирования. Из формулы (12) следует, что для получения решения используется ряд Тейлора с точностью до членов первого порядка малости по h , поэтому ошибка на каждом шаге может быть оценена как

$$e_k = \frac{h^2}{2!} y_k''(x) \quad (x_k \leq x \leq x_{k+1}) \quad (3.2)$$

Несмотря на простоту реализации, метод Эйлера относительно редко применяется при интегрировании дифференциальных уравнений, поскольку не обеспечивает достаточной точности вычислений и очень часто бывает, неустойчив из-за накопления ошибок. Устойчивость метода увеличивается при уменьшении шага интегрирования, однако при интегрировании сложных уравнений уменьшение шага приводит к резкому увеличению объема вычислений и, как следствие, - к значительным затратам машинного времени.

Модифицированный метод Эйлера. В отличие от обычного метода Эйлера, когда для вычисления следующей точки интегральной кривой требуется информация только о предыдущей точке, модификация метода заключается в использовании прогноза поведения интегральной кривой в последующих точках. Модифицированный метод основан на усреднении положения концевой точки отрезка, которым заменяется интегральная кривая. Усреднение производится с учетом тангенса угла наклона в некоторой промежуточной точке, например в точке, отстоящей от начальной на половину шага интегрирования. Порядок построения решения в модифицированном методе Эйлера представлен на рис.3 и заключается в следующем. Проводится касательная A_1 через точку x_i, y_i с тангенсом угла наклона $f(x_i, y_i)$ до пересечения с прямой $x = x_i + h/2$ и в точке пересечения вычисляется производная, равная

$$k_i = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} y'_i\right),$$

с учетом которой проводится касательная в этой точке (прямая A_2). Далее из точки x_i, y_i проводится прямая A_0 , параллельная A_2 , до пересечения с прямой $x = x_{i+1}$, уравнение которой имеет вид

$$y_{i+1} = y_i + (x_{i+1} - x_i) k_i = y_i + hf\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} y'_i\right). \quad (3.3)$$

Формула (3.3) и есть формула модифицированного метода Эйлера. Она может быть записана иначе, если промежуточная точка, производная в которой определяет направление отрезка интегральной кривой,

отстоит от соседних точек на величину шага

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i), \quad (3.4)$$

Таким образом, при использовании модифицированного метода Эйлера на каждом шаге интегрирования необходимо иметь информацию о текущей и предшествующей точках. Поэтому этим методом нельзя воспользоваться на первом шаге интегрирования из начальной точки. На первом шаге обычно применяется простой метод Эйлера, после чего на последующих шагах уже можно использовать формулу (3.3).[1,2]

Усовершенствованный метод Эйлера – Коши заключается в том, что при вычислении следующей точки интегральной кривой производится усреднение тангенса угла наклона кривой в двух прилежащих точках. В этом случае тангенс угла наклона находится как среднее арифметическое тангенсов в точках (x_i, y_i) и (x_{i+1}, y_{i+1}) .

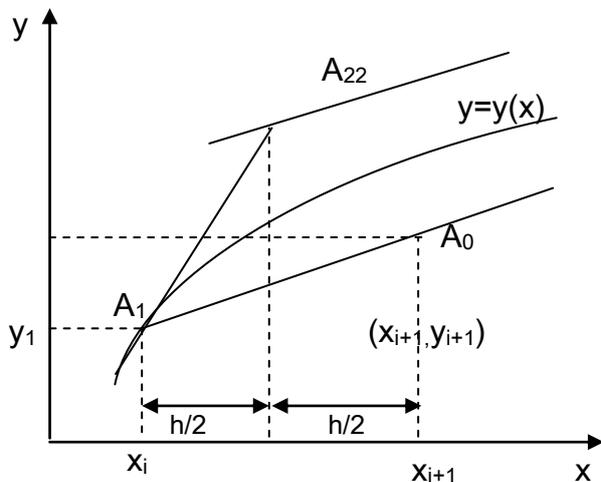


Рис.3. Графическая иллюстрация модифицированного метода Эйлера

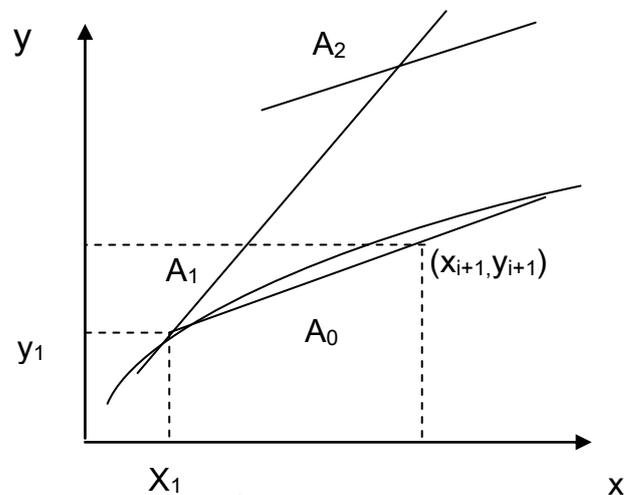


Рис. 4 Графическая иллюстрация модифицированного метода Эйлера- Коши

Поэтому аналогично модифицированному методу Эйлера здесь также на каждом шаге необходимо дважды вычислить значение правой части дифференциального уравнения.

Порядок построения решения в методе Эйлера – Коши представлен на рис. 3. Через точку (x_i, y_i) проводится касательная A_1 до пересечения с прямой $x = x_{i+1}$. В этой точке пересечения вычисляется тангенс угла наклона

касательной A_2 и из точки x_i, y_i проводится прямая A_0 , тангенс угла наклона которой есть среднее арифметическое тангенсов углов наклона прямых A_1 и A_2 , т.е.

$$k_i = \frac{1}{2} [f(x_i, y_i) + f(x_i + h, y_i + hy'_i)],$$

где $y'_i = f(x_i, y_i)$.

Для прямой A_0 можно записать уравнение

$$y_{i+1} = y_i + hk_i = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_i + h, y_i + hy'_i)], \quad (3.5)$$

которое и является формулой метода Эйлера – Коши.

При использовании формулы (3.5) сначала вычисляется приближенное решение в точке (x_{i+1}, y_{i+1})

$$y_{i+1}^{(0)} = y_i + hf(x_i, y_i), \quad (3.6)$$

которое затем уточняется.

Решение по формуле (3.5) можно улучшить, если воспользоваться итерационным способом уточнения каждого значения y_i . В этом случае, исходя из начального приближения (3.6), строится итерационный процесс по формуле

$$y_{i+1}^{(k+1)} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(k)})] \quad (k = 0, 1, 2, \dots, m, \dots) \quad (3.7)$$

и вычисления на каждом шаге повторяются до тех пор, пока два последовательных приближения $y_{i+1}^{(k)}$ и $y_{i+1}^{(k+1)}$ не совпадут с некоторой наперед заданной точностью.

Оценка точности формул Эйлера. Для того чтобы оценить точность улучшенных формул Эйлера, можно сравнить их с разложением решения в ряд Тейлора.

Для $f(x, y)$ в окрестности точки (x_i, y_i) имеем

$$f(x, y) = f(x_i, y_i) + (x - x_i) \frac{\partial f}{\partial x} + (y - y_i) \frac{\partial f}{\partial y} + \dots$$

Тогда для величин k_i в формулах (3) и (5) можно записать соответственно

$$k_i = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} y_i'\right) = f + \frac{h}{2}(f_x + ff_y) + 0(h^2),$$

$$k_i = \frac{1}{2}[f(x_i, y_i) + f(x_i + h, y_i + hy_i')] = \frac{1}{2}[f + f + hf_x + hff_y] +$$

$$+ 0(h^2) = f + \frac{h}{2}(f_x + ff_y) + 0(h^2).$$

Формулы (3.3) и (3.5) после подстановки соответствующих выражений для k_i приводятся к виду

$$y_{i+1} = y_i + hf + \frac{h^2}{2}(f_x + ff_y) + 0(h^3). \quad (3.8)$$

Из сравнения (3.16) и (3.8) можно заключить, что видоизмененные формулы Эйлера согласуются с разложением в ряд Тейлора с точностью до членов степени h^2 включительно, т.е. обеспечивают точность на порядок выше, чем формула (1). Можно также показать, что формула Эйлера – Коши с итерационным уточнением на каждом шаге позволяет получить точность порядка h^3 . [1]

Устойчивость решения. При решении дифференциальных уравнений численными методами помимо вопросов точности важную роль приобретают вопросы устойчивости решения. Под устойчивостью метода решения дифференциального уравнения понимается способность накопления и скорость роста ошибки интегрирования. Как уже отмечалось выше, при использовании формулы (3.1) ошибка вычислений накапливается в процессе интегрирования.

Скорость ее роста и абсолютную величину можно уменьшить только путем уменьшения шага интегрирования, однако при этом значительно возрастает объем вычислений. В месте с тем не существует способов выбора

шага интегрирования до начала решения, поэтому на практике шаг обычно выбирается исходя из сравнения решений, полученных с различным шагом, например с h и $h/2$. Если эти решения отличаются на величину, большую заданной точности, то производится уменьшение шага интегрирования. Оценка ошибки иногда может быть получена и другими способами, например исследованием производных правой части дифференциального уравнения. Однако это не всегда целесообразно, так как может оказаться, что повторное интегрирование с измененным шагом выполнить проще, чем исследовать поведение правой части уравнения на каждом шаге интегрирования.

Модифицированные формулы Эйлера относительно часто применяются в практике вычислений. Рассмотрим устойчивость решения, полученного по этим формулам.

Предположим, что для решения уравнения $y' = f(x, y)$ используется модифицированная формула (2.20). Тогда для вычисленного значения y_{i+1} в точке x_{i+1} можно записать

$$y_{i+1} = y_{i-1} + 2hy'_i. \quad (3.9)$$

Однако если Y_i - точное решение, то для него можно также записать

$$Y_{i+1} = Y_{i-1} + 2hY'_i + e_{ii}, \quad (3.10)$$

где e_i - ошибка расчета по формуле (3.9) на одном шаге интегрирования.

Если обозначить ошибку вычисления величины y_i через

$$\varepsilon_{i+1} = \varepsilon_{i-1} + 2h(Y'_i - y'_i) + \varepsilon_i = \varepsilon_{i-1} + 2h[f(x_i, Y_i) - f(x_i, y_i)] + e_i$$

то после вычитания выражения (3.9) из (3.10) получим

$$\varepsilon_{i+1} = \varepsilon_{i-1} + 2h(Y'_i - y'_i) + e_i = \varepsilon_{i-1} + 2h[f(x_i, Y_i) - f(x_i, y_i)] + e_i$$

Последнее выражение можно переписать, используя теорему о среднем в виде

$$\varepsilon_{i+1} = \varepsilon_{i-1} + 2h \frac{\partial f(x_i, g_i)}{\partial y} (Y_i - y_i) = \varepsilon_{i-1} + 2h \frac{\partial f}{\partial y} \varepsilon_i + e_i, \quad (3.11)$$

где $y_i \leq g_i \leq Y_i$.

Допустим, что e_i и $A = df / dy$ постоянны, тогда выражение (3.11) запишется в виде соотношения

$$\varepsilon_{i+1} - 2Ah \varepsilon_i = e_i, \quad (3.12)$$

которое, является линейным разностным неоднородным уравнением с постоянными коэффициентами.

Решение уравнения (3.12) может быть найдено как сумма общего решения однородного уравнения $\bar{\varepsilon}$ и частного решения неоднородного $\bar{\varepsilon}$.

Общее решение однородного уравнения (3.12) определяется корнями характеристического уравнения $p^2 - 2Ahp - 1 = 0$ которые равны

$$P_1 = Ah + \sqrt{A^2 h^2 + 1}, \quad P_2 = Ah - \sqrt{A^2 h^2 + 1}.$$

Следовательно, решение уравнения (4.12) есть

$$\tilde{\varepsilon}_i = c_1 P_1^i + c_2 P_2^i. \quad (3.13)$$

В зависимости от знака величины A один из корней характеристического уравнения (3.12) оказывается по модулю большим единицы. Поэтому при $i \rightarrow \infty$ общее решение неоднородного уравнения $\tilde{\varepsilon} \rightarrow \infty$, и, следовательно, модифицированный метод Эйлера является неустойчивым. Так же как и для простой формулы Эйлера, ошибка может быть уменьшена только за счет уменьшения шага интегрирования.

Аналогичными рассуждениями можно показать, что формула (3.5) является относительно устойчивой, так как по мере роста x относительная ошибка не возрастает.

Исследуем условия сходимости итерационной формулы (3.7). Пусть $y_{i+1}^{(k+1)}$ и $y_{i+1}^{(k)}$ - два последующих итерационных приближения, полученные по формуле (3.7). Тогда, очевидно, имеет место равенство

$$y_{i+1}^{(k+1)} - y_{i+1}^{(k)} = \frac{h}{2} (f(x_{i+1}, y_{i+1}^{(k)}) - f(x_{i+1}, y_{i+1}^{(k-1)})) . \quad (3.14)$$

Используя теорему о среднем значении и вводя обозначение ($\varepsilon_{k+1} = y_{i+1}^{(k+1)} - y_{i+1}^{(k)}$ - ошибка $k + 1$ -го приближения), равенство (14) можно переписать в виде

$$\varepsilon_{k+1} = \frac{h}{2} \frac{df}{dy} (y_{i+1}^{(k)} - y_{i+1}^{(k-1)}) \quad (3.15)$$

или
$$|\varepsilon_{k+1}| \leq \frac{Ah}{2} |\varepsilon_k| ,$$

где
$$A = \left| \frac{df}{dy} \right| .$$

Следовательно, итерационный процесс, определяемый формулой (3.7), будет сходящимся, если выполняется неравенство $Ah / 2 < 1$ или

$$h < \frac{2}{|A|} . \quad (3.16)$$

Равенства, аналогичные (3.14), могут быть записаны для $\varepsilon_k, \varepsilon_{k-1}, \dots, \varepsilon_1$. Тогда последовательно подставляя эти выражения в (3.15), можно записать

$$|\varepsilon_{k+1}| \leq (Ah / 2)^k |\varepsilon_1| . \quad (3.17)$$

Таким образом, при изменении k от 1 до m последовательность $y_{i+1}^{(k)}$ сходится к некоторому определенному значению, отличающемуся от точного решения на величину ошибки вычислений.

Из условия (3.17) следует, что чем больше величина A , тем меньше шаг, обеспечивающий сходимость итерационного процесса, и чем меньше величина шага, тем меньшее количество итераций потребуется для получения устойчивого решения в точке. Однако в общем случае нельзя однозначно определить, что выгоднее: взять меньший шаг, а следовательно, увеличить число шагов интегрирования, или же увеличить шаг, но производить большее число итераций на каждом шаге. Поэтому в практике вычислений обычно поступают следующим образом. Выбирается произвольный шаг интегрирования и производятся итерации по формуле (3.7). Если после двух итераций заданная точность не обеспечивается, то

размер шага уменьшается и расчеты повторяются. Эмпирически установлено, что такая стратегия обеспечивает минимальный объем вычислений при заданной точности. [2,3,11]

Формулы Рунге-Кутты. Наиболее распространенными в практике интегрирования обыкновенных дифференциальных уравнений является формулы Рунге – Кутты. Эти формулы классифицируются по степени приближения их по точности к разложению решения в ряд Тейлора. Формулы, точные до второго, третьего, четвертого и т.д. членов разложения, носят название формул второго, третьего, четвертого и т.д. порядка соответственно. Достоинством формул Рунге – Кутты является то, что при их использовании не нужно вычислять производные выше первого порядка вычислений на каждом шаге.

В основе формул Рунге – Кутты используется следующее допущение. Поскольку определение производных высокого порядка в разложении решения в ряд Тейлора сопряжено со значительными вычислительными трудностями, вместо уравнения (3.12) используется линейная комбинация вида.

$$y_{m+1} = y_m + p_{c,1}k_1(h) + p_{c,2}k_2(h) + \dots + p_{c,c}k_c(h), \quad (3.18)$$

где $p_{c,i}$ - постоянные коэффициенты при

функциях $k_i(h) = hf(\xi_i, \eta_i)$ ($i = 1, 2, \dots, c$), в которых

$$\xi_1 = x_m + \alpha_1 h, \eta_1 = y_m + b_{1,1}k_1(h) + b_{1,2}k_2(h) + \dots + b_{1,i-1}k_{i-1}(h) \quad (3.19)$$

а α_i и b_{ij} - постоянные, причем $\alpha_1 = 0$.

В уравнении (3.18) c определяет порядок формулы. Таким образом,

$$\begin{aligned} k_1(h) &= hf(x_m, y_m), \\ k_2(h) &= hf(x_m + \alpha_2 h, y_m + b_{2,1}k_1), \\ k_3(h) &= hf(x_m + \alpha_3 h, y_m + b_{3,1}k_1 + b_{3,2}k_2), \\ &\dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ k_c(h) &= hf(x_m + \alpha_c h, y_m + b_{c,1}k_1 + b_{c,2}k_2 + \dots + b_{c,c-1}k_{c-1}). \end{aligned}$$

Введем вспомогательную функцию

$$\varphi_c(h) = y_{m+1} - y_m - [p_{c,1}k_1(h) + p_{c,2}k_2(h) + \dots + p_{c,c}K_c(h)] \quad (3.20)$$

и потребуем, чтобы эта функция при $h = 0$ обладала свойствами

$$\varphi_c(0) = \varphi_c'(0) = \dots = \varphi_c^{(c)}(0) = 0, \quad \varphi_c^{(c+1)}(0) \neq 0 \quad (3.21)$$

Тогда соотношения (21) могут использоваться для выбора постоянных коэффициентов $p_{c,i}$, α_i , $b_{i,j}$.

Рассмотрим несколько частных случаев формулы (3.18)

Пусть $c = 1$. Тогда

$$y_{m+1} = y_m + p_{1,1}k_1 = y_m + p_{1,1}hf(x_m, y_m)$$

и

$$\varphi_1(h) = y_{m+1} - y_m - p_{1,1}hf(x_m, y_m),$$

$$\varphi_1'(0) = y_m' - p_{1,1}hf(x_m, y_m).$$

Поскольку при $h = 0$ $y_{m+1} = y_m$, то $\varphi_1(0) = 0$, а $\varphi_1'(0) = y_m' - p_{1,1}f(x_m, y_m)$. Условие (4.21) для $\varphi_1'(0)$ будет выполнено лишь в том случае, если $p_{1,1} = 1$, и, таким образом, при $c = 1$ расчетная формула

$$y_{m+1} = y_m + hf(x_m, y_m) \quad (3.22)$$

т.е. получила вид формулы Эйлера.

Пусть $c = 2$. Тогда

$$\xi_2 = x_m + \alpha_2 h, \quad \xi_1 = x_m + \alpha_1 h = x_m,$$

$$\eta_2 = y_m + b_{2,1}k_1, \quad \eta_1 = y_m.$$

Поскольку

$$k_1 = hf(x_m, y_m), \quad k_2 = hf(x_m + \alpha_2 h, y_m + b_{2,1}k_1),$$

то

$$\begin{aligned} y_{m+1} &= y_m + [p_{2,1}k_1(h) + p_{2,2}k_2(h)] = \\ &= y_m + hp_{2,1}f(x_m, y_m) + hp_{2,2}[f(x_m + \alpha_2 h, y_m + b_{2,1}hf(x_m, y_m))] \end{aligned} \quad (3.23)$$

Формула (3.23) содержит четыре неизвестных коэффициента - $p_{2,1}$, $p_{2,2}$, α_2 и $b_{2,1}$, которые необходимо определить исходя из условий

(3.21). Для того чтобы найти значения этих коэффициентов, нужно записать выражения для $\varphi_2'(h)$ и $\varphi_2''(h)$ и положить $h = 0$. Тогда неизвестные коэффициенты могут быть найдены решением системы линейных уравнений.

Выражение для $\varphi_2(h)$ имеет вид

$$\varphi_2(h) = y_{m+1} - y_m - [p_{2,1}k_1(h) + p_{2,2}k_2(h)] \quad (3.24)$$

Вычислим $\varphi_2'(h)$

$$\varphi_2'(h) = y_{m+1}' - [p_{2,1}k_1'(h) + p_{2,2}k_2'(h)], \quad (3.25)$$

где

$$k_1'(h) = f(x_m, y_m) = f, \quad k_2'(h) = f(\xi_2, \eta_2) + h[\alpha_2 f_x + b_{2,1} f f_y]$$

Для $\varphi_2'(0)$ можно записать $\varphi_2'(0) = f - [p_{2,1}f + p_{2,2}f]$, где $f = f(x_m, y_m)$, откуда следует, что условие (21) для $\varphi_2'(0)$ будет выполнено в том случае, если

$$p_{2,1} + p_{2,2} = 1. \quad (3.26)$$

Аналогично, дифференцируя $\varphi_2'(h)$ по h и учитывая, что $k_1''(h) = 0$, для $\varphi_2''(0)$ получим

$$\varphi_2''(0) = f_x + f f_y - 2 p_{2,2} [\alpha_2 f_x + b_{2,1} f f_y],$$

откуда следует, что для $\varphi_2''(0) = 0$ должно быть

$$2 p_{2,2} \alpha_2 = 1, \quad 2 p_{2,2} b_{2,1} = 1. \quad (3.27)$$

Таким образом, коэффициенты $p_{2,1}$, $p_{2,2}$, α_2 , $b_{2,1}$ формулы Рунге – Кутты второго порядка удовлетворяют системе линейных уравнений

$$\begin{aligned} p_{2,1} + p_{2,2} &= 1, \\ p_{2,2} \alpha_2 &= 1/2, \\ p_{2,2} b_{2,1} &= 1/2. \end{aligned} \quad (3.28)$$

Поскольку система (3.28) имеет четыре неизвестных, то число ее решений бесконечно. Каждое решение дает формулу интегрирования с

ошибкой порядка h^3 . На практике коэффициенты выбираются такими, чтобы формулы были удобны для вычислений. Например, если положить $\alpha_2 = 1$, то тогда $p_{2,2} = 1/2$; $p_{2,1} = 1/2$; $b_{2,1} = 1$. После подстановки этих значений в выражение (23) получим формулу

$$y_{m+1} = y_m + 1/2(k_1 + k_2) = y_m + h/2[f(x_m, y_m) + f(x_m + h, y_m + hy'_m)],$$

являющуюся формулой усовершенствованного метода Эйлера – Коши.

Если же положить, что $p_{2,1} = 0$, то из (3.28) получим $p_{2,2} = 1$; $b_{2,1} = \alpha_2 = 1/2$. Тогда после подстановки полученных значений в (3.23) последнее преобразуется к виду (3), т.е. получим формулу модифицированного метода Эйлера. Отсюда следует, что формулы Эйлера является частными случаями формул Рунге-Кутты первого и второго порядков. [1,2,15,16]

Наибольшее распространение в вычислительной практике имеют формулы Рунге-Кутты четвертого порядка, которые получаются из общих соотношений (3.18) – (3.21) при $c = 4$.

При этом

$$y_{m+1} = y_m + p_{4,1}k_1 + p_{4,2}k_2 + p_{4,3}k_3 + p_{4,4}k_4, \quad (3.29)$$

где

$$\begin{aligned} k_1(h) &= hf(x_m, y_m), \\ k_2(h) &= hf(x_m + \alpha_2 h, y_m + b_{2,1}K_1), \\ k_3(h) &= hf(x_m + \alpha_3 h, y_m + b_{3,1}K_1 + b_{3,2}K_2), \\ k_4(h) &= hf(x_m + \alpha_4 h, y_m + b_{4,1}K_1 + b_{4,2}K_2 + b_{4,3}K_3), \end{aligned}$$

Коэффициенты выражения (3.29) также вычисляются исходя из условий теперь уже для производных до четвертого порядка включительно. Точность формул четвертого порядка пропорциональна h^5 , т.е. формулы

соответствуют разложению решения в ряд Тейлора до членов h^4 включительно.

При повышении порядка формулы возрастает произвол в выборе коэффициентов. Для формул четвертого порядка уже имеется две степени свободы, что приводит к еще большему разнообразию соотношений.

Примером формулы Рунге – Кутты четвертого порядка может служить соотношение

$$y_{m+1} = y_m + 1/6 (k_1 + 2k_2 + 2k_3 + k_4) + O(h^5). \quad (3.30)$$

где

$$\begin{aligned} k_1 &= hf(x_m, y_m), \\ k_2 &= hf(x_m + h/2, y_m + 1/2k_1), \\ k_3 &= hf(x_m + h/2, y_m + 1/2k_2), \\ k_4 &= hf(x_m + h, y_m + k_3). \end{aligned} \quad (3.30a)$$

$$y_{m+1} = y_m + 1/2 (k_1 + 4k_4 + k_5). \quad (3.31)$$

$$\begin{aligned} k_1 &= 1/3 hf(x_m, y_m), \\ k_2 &= 1/3 hf(x_m + 1/3h, y_m + k_1), \\ k_3 &= 1/3 hf(x_m + 1/3h, y_m + 1/2k_1 + 1/2k_2), \\ k_4 &= 1/3 hf(x_m + 1/2h, y_m + 3/8k_1 + 8/9k_3), \\ k_5 &= hf(x_m + h/2, y_m + 1/2k_2) \end{aligned} \quad (3.31a)$$

предлагается следующая стратегия оценки точности интегрирования и выбора шага. Если правая часть выражения $\varepsilon = k_1 - 9/2k_3 + 4k_4 - 1/2k_5$ превышает заданную точность более чем в пять раз, то шаг уменьшается вдвое и интегрирование продолжается. Если же правая часть меньше, чем $5/16 \varepsilon$, то шаг может быть удвоен. По оценке интегрирование по формулам (3.30) на 20% быстрее, чем по формулам (3.30) с постоянным шагом.

Заключение

1-ый раздел посвящен математическому и физическому моделированию прикладных задач, преимущество этого метода перед экспериментами, который требует огромных затрат.

Приводятся конкретные примеры из области химии и химической технологии. Составляется математический модель задач химии и химической технологии .

Составлен оптимальный алгоритм, блок – схема и программы на современном языке Delphi.

ГЛАВА II. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ НЕСТАЦИОНАРНЫХ РЕЖИМОВ

2.1 Математическая модель нестационарных режимов тарельчатой ректификационной колонны

Математическое описание нестационарных процессов, происходящих в ректификационной колонне основывается на уравнениях материального и теплового балансов.

При решении ряда практических задач можно допустить, ряд допущений и исходя из принятых допущений математическое ректификационной колонны представляет систему дифференциальных уравнений, записанных для каждой тарелки.

Для произвольной тарелки дифференциальное уравнение, записанного для компонента j , имеет вид

$$L_{i+1}x_{i+1,j} + V_{i-1}Y_{i-1,j} - V_{i-1}Y_{i-1,j} - V_iY_{i,j} - L_ix_{i,j} = \frac{d(H_{L,i} + H_{V,i})}{dt}, \quad (1.1)$$

Уравнения, записанные для колонны, дефлегматора и тарелки питания при условии, что питание подается в жидкой фазе, соответственно имеют вид

$$L_1x_{1,j} - V_0Y_{0,j} - Wx_{0,j} = \frac{d(H_{L,0} + H_{V,0})}{dt}, \quad (1.2)$$

$$L_{j+1}x_{j+1,j} + V_{f-1}Y_{f-1,j} - V_fY_{f,j} - L_jx_{f,j} + Fx_{f,j} = \frac{d(H_{L,f} + H_{V,f})}{dt}, \quad (1.3)$$

$$V_NY_{N,j} - L_{N+1}x_{N+1,j} - Dx_{N+1,j} = \frac{d(H_{L,N+1} + H_{V,N+1})}{dt}. \quad (1.4)$$

где H_L - задержка жидкости на тарелке; H_V - задержка пара на тарелке;

Поскольку $H_L > H_V$, то задержкой пара на тарелке можно пренебречь без значительной потери точности. Тогда, учитывая, что

$$Y_{i,j} = K_{i,j} x_{i,j} = \frac{\alpha_j}{\sum_{z=1}^k a_z x_{i,j}} x_{i,j}, \quad (1.5)$$

систему уравнений (4.1) – (4.4) можно переписать в виде

$$\begin{pmatrix} \frac{dH_{0,j}}{dt} \\ \frac{dH_{1,j}}{dt} \\ \dots \\ \frac{dH_{f,j}}{dt} \\ \dots \\ \frac{dH_{n+1,j}}{dt} \end{pmatrix} = \begin{pmatrix} B & C & 0 & 0 & \dots & 0 \\ A & B & C & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & A & B & C & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & A & B & \dots \end{pmatrix} \begin{pmatrix} x_{0,j} \\ x_{1,j} \\ \dots \\ x_{f,j} \\ \dots \\ x_{n+1,j} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ Fx_{1,j} \\ \dots \\ 0 \end{pmatrix} \quad (1.6)$$

где

$$A = V_i K_{i,j}; \quad B_i = -(L_i + V_i K_{i,j});$$

$$dH / dt = A(t) X(t) + F(t). \quad (1.7)$$

Система (1.7) в общем виде неразрешима обычными способами. Для ее решения используется приближенными методами: Эйлера, Рунге – Кутта.

Далее приводится программа расчёта изменения концентраций на тарелках колонны.

Программа интегрирования системы $y'_k = f(x_k, y_k)$ выполняется следующим образом.

Вводятся исходные данные, характеризующие режим работы колонны и свойства разделяемой смеси: К — количество компонентов; N — число тарелок колонны; NF — местоположение ввода питания, INT — шаг вывода на печать; F — количество питания, кмоль/час; D — количество дистиллята, кмоль/час; L—количество орошения, кмоль/час; BT— шаг интегрирования; EPS —точность; XF — состав питания; ALF — коэффициенты относительной

летучести. Затем задаются начальные условия: количество и состав жидкости по высоте колонны полагаются равными количеству и составу питания соответственно (пусковой режим).[1,4]

Интегрирование системы уравнений начинается с вычисления правых частей уравнений. С этой целью вычисляются константы фазового равновесия и коэффициенты системы $y'_k = f(x_k, y_k)$. В соответствии с формулой интегрирование на каждом шаге производится дважды — это обеспечивается циклом по Р. В результате расчета определяется количество жидкости на тарелках по каждому из компонентов (массив Н), а затем уже состав. В процессе вычисления концентрации производится нормировка, поэтому для каждой из тарелок выполняется условие $\sum_{i=1}^k x_i = 1$.

После определения значений функции Н (х) в последующей точке производится проверка на окончание интегрирования. С этой целью вычисляется сумма потоков на каждой из тарелок колонны и сравнивается с соответствующей суммой в предыдущей точке. Если они не равны между собой с заданной точностью, то переменная z принимает ненулевое значение и после вывода решение будет продолжено с оператора с меткой *START*

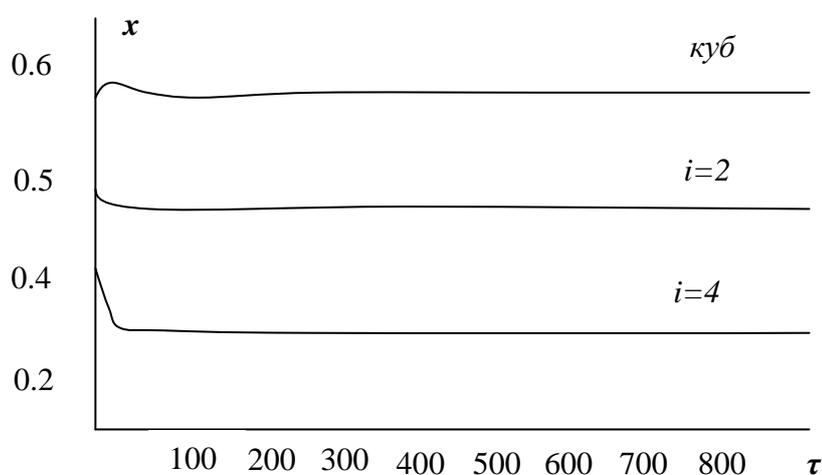


Рис.5. График изменения концентрации на тарелках колонны

На рис.5 представлены результаты расчета колонны для разделения бинарной смеси. Исходные данные $K = 2$, $EPS = 0,01$; $XF = 0,5$; $\alpha_1 = 1$; $N = 3$, $DT = 0,1$; $F = 2$; $a_3 = 2$; $NF = 2$; $L = 5$ $D = 1$.

2.2 Математическая модель реактора идеального вытеснения и его численное решение

Реактор идеального вытеснения представляет собой объект с распределенными параметрами. Поэтому математическое описание его нестандартных режимов представляется системой дифференциальных уравнений в частных производных, описывающей изменение концентраций реагентов и температуры как по длине реактора, так и во времени.

Стационарный режим работы реактора описывается обыкновенными дифференциальными уравнениями, определяющими изменение концентраций реагентов и температуры по длине зоны реакции [5]:

$$V \frac{dx_i}{dl} = S_b W_i \quad (i = 1, 2, \dots, m) \quad (2.1)$$

$$VC_p \frac{dT}{dl} = S_b Q_r + \frac{K_r F}{l_b} (T_x - T) \quad (2.2)$$

где S_b - площадь сечения зоны реакции, l_b - длина зоны реакции, l - расстояние от начала зоны реакции.

При использовании уравнений (2.1) и (2.2) для описания реактора вытеснения предполагаются справедливыми следующие допущения: реагирующая смесь идеально перемешивается в поперечном сечении потока; продольное перемешивание в потоке отсутствует; теплоемкость реагирующей смеси не изменяется в процессе химического превращения; теплопроводность смеси и стенок реактора в направлении движения потока можно пренебречь; поверхность теплообмена равномерно распределена по длине зоны реакции; количество реагирующей смеси при принятом способе выражения величины потока V не изменяется в процессе реакции. При

использовании уравнений (2.1) и (2.2) для математического описания конкретных процессов необходимо иметь соответствующие выражения для скоростей образования реагентов в рассматриваемом процессе w_i , а также выражение для суммарного теплового эффекта реакции Q , [4]

Общее выражение для скоростей образования реагентов в произвольной химической реакции может быть получено при использовании понятия "скорости стадии сложной реакции". При этом под скоростью j -й стадии понимается величина

$$r_j = \frac{w_{i,j}}{n_{i,j}} = - \frac{w_{k,j}}{n_{k,j}} \quad (2.3)$$

Определяемая для элементарной стадии реакции

(2.4) где $n_{k,j}$ стехиометрический коэффициент k -го реагента, являющегося продуктом данной стадии; $n_{i,j}$ - стехиометрический коэффициент i -го реагента, являющегося исходным в данной стадии; $w_{i,j}$ - скорость образования i -го реагента на данной стадии; $w_{k,j}$ - скорость образования k -го реагента, являющаяся отрицательной величиной, поскольку этот реагент на данной стадии расходуется. Например, для химической



можно рассмотреть следующие элементарные стадии:



Любую химическую реакцию можно описать матрицей стехиометрических коэффициентов $[M]$, число строк которой равно числу

реагентов, участвующих в реакции, а число столбцов равно числу стадии реакции. Элемент N_{kj} матрицы при этом определяется как стехиометрический коэффициент κ - го реагента на j -й стадии, причем этому коэффициенту приписывается знак минус, если κ -й реагент является исходным, и знак плюс, если он образуется на данной стадии. Например, для химической реакции (5) матрица стехиометрических коэффициентов имеет вид

$$N = \begin{pmatrix} -n_{11} & 0 & 0 \\ -n_{21} & -n_{22} & -n_{23} \\ 0 & 0 & -n_{43} \end{pmatrix} \quad (2.7)$$

Если для каждой стадии сложной реакции известны зависимости скорости γ , от концентраций реагентов и температуры, то с использованием матричных обозначений можно записать следующее соотношение:

$$w = [N]r \quad (2.8)$$

Здесь (w) - вектор скоростей образования реагентов в сложной реакции

$$w = \begin{pmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_m \end{pmatrix} \quad (2.9)$$

а r – вектор скоростей стадии реакции

$$r = \begin{pmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_s \end{pmatrix} \quad (2.10)$$

Например, для реакции (2.5) можно получить

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} = \begin{pmatrix} -n_{11} & 0 & 0 \\ -n_{21} & -n_{22} & -n_{23} \\ 0 & 0 & -n_{43} \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \quad (2.11)$$

Откуда следует, что

$$\begin{cases} w_1 = -n_{11} r_1 \\ w_2 = -n_{21} r_1 - n_{22} r_2 - n_{23} r_3 \\ w_3 = n_{32} r_2 \\ w_4 = n_{43} r_3 \end{cases} \quad (2.12)$$

С использованием понятия скорости стадии суммарный тепловой эффект сложной химической реакции может быть выражен как

$$Q_r = \sum_{j=1}^3 n_{ij} \cdot q_j \cdot r_j \quad (2.13)$$

где q_j - тепловой эффект стадии: отнесенный к 1 моль продукта реакции этой стадии, n_{ij} - стехиометрический коэффициент образующегося на j стадии продукта реакции; S – общее число стадий сложной реакции. Воспользовавшись векторным выражением для скоростей образования реагентов (2.8), систему уравнений (2.1) можно также представить в векторной форме

$$vd(x)/dl = S_b [N](r) \quad (2.14)$$

Суммарный тепловой эффект реакции Q . определяется выражением (2.13).

Система уравнений (2.1) и (2.2) должна интегрироваться при начальных условиях, соответствующих составу и температуре исходной смеси на входе в реактор, т.е.

$$x_i \Big|_{l=0} = x_i^{(0)}, \quad T \Big|_{l=0} = T^{(0)} \quad (2.15)$$

Ниже представлена программа расчета изотермического реактора вытеснения для произвольной кинетики.

В программе используется стандартная процедура решения системы уравнений (2.14) по формулам Рунге-Кутты 4-го порядка с некоторыми изменениями, внесенными на стадии отладки, и процедура вычисления правых частей системы.

Формальными параметрами процедуры являются : X_0 – начальные условия, EPS - точность интегрирования; PP - булевская переменная, значение которой при первом обращении к процедуре должно быть true, а при последующих - false; XFIN - конечное значение интервала интегрирования; YFIN - выходной параметр процедуры (массив значений концентраций). Интегрирование осуществляется с автоматическим выбором шага, поэтому XFIN — по существу является интервалом вывода промежуточных решений.

Формальными параметрами процедуры являются: X_0 - текущие значения аргумента и функции, при которых вычисляются правые части системы уравнений; N - число уравнений системы; Z - совокупность производных - выходной параметр процедуры.

Исходные данные: N=3; HOUT=0,1; M=2; EPS=0,01; X=0; SB=1,0; V=0,5;

$$y = \begin{pmatrix} 1,0 \\ 0,0 \\ 0,0 \end{pmatrix}; \quad Q = \begin{pmatrix} -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}; \quad KR = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Заключение

Во втором разделе рассматривается математическая модель нестационарных режимов тарельчатой ректификационной колонны. В качестве примера рассмотрена математическая модель реактора, идеального вытеснения и его численное решение. Полученные результаты сопоставлены с экспериментальными данными.

ГЛАВА III. РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ ХИМИИ И ХИМИЧЕСКОЙ ТЕХНОЛОГИИ

3.1 Применение алгоритма метода Эйлера и Рунге – Кутта для решения задач химии и химических технологий

Рассмотрим сравнительную оценку различных формул интегрирования на простейшем примере. Пусть в сосуд объемом V , заполненный жидкостью состава x_0 , с постоянной скоростью F подается жидкость состава x_F и с такой же скоростью выводится жидкость из сосуда. Полагая, что жидкость в сосуде идеально перемешивается, найти зависимость концентрации на выходе сосуда, используя формулы Эйлера и Рунге – Кутта.

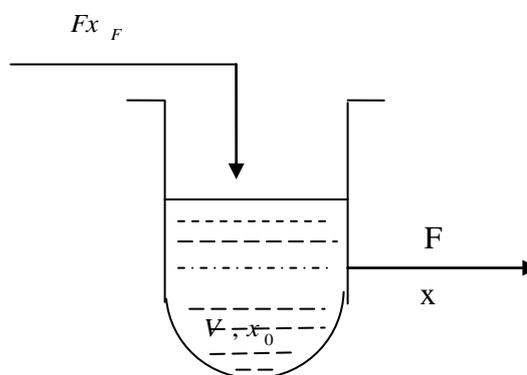


Рис. 5

Запишем уравнение материального баланса сосуда, схематически представленного на рис. 5.

$$F(x_F - x) = d(Vx) / dt. \quad (1.1)$$

Так как скорость ввода питания равна скорости вывода, то $dV / dt = 0$, и, таким образом, уравнение, описывающее изменение концентрации в сосуде, имеет вид

$$V(dx / dt) = F(x_F - x) \quad (1.2)$$

и интегрируется с начальным условием $x|_{t=0} = 0$.

Решением уравнения (1.2) является выражение

$$x = \exp(-Ft / V (x_0 - x_F) + x_F)$$

Решением уравнения (1.2) по формулам Эйлера (1- 3.1) и (1 - 3.2), а также по формулам Рунге – Кутта (3.31) для интервала времени $0 \leq t \leq 40$ приведено в табл.1

Исходные данные: $x_0 = 0$, $F = 1 \text{ м}^3 / \text{час}$, $x_F = 0,9$, $V = 5 \text{ м}^3$.

Таблица 1.

Метод	Шаг <i>H</i>	Независимая переменная <i>t</i>						
		1	5	10	15	20	30	40
Точное решение	-	0,1631	0,5689	0,7781	0,8552	0,8835	0,8977	0,8996
Рунге – Кутта	5	0,1631	0,5689	0,7781	0,8550	0,8831	0,8971	0,8996
	10	-	-	0,7782	0,8551	0,8835	0,8977	0,8996
	15	-	-	-	0,8552	-	0,8978	0,8997
	20	-	-	-	-	-	-	-
Эйлера	5	0,1661	0,6051	0,8300	0,8908	0,8997	0,9003	0,9699
	10	0,1646	0,5862	0,8033	0,8746	0,8946	0,8999	0,8999
	15	0,1641	0,5803	0,7947	0,8683	0,8914	0,8995	-
	20	0,1638	0,5773	0,7905	0,8651	0,8896	0,8992	-
Модифицированный метод Эйлера	5	0,1631	0,5663	0,7691	0,8409	0,8657	0,8658	0,7691
	10	-	0,5683	0,7763	0,8526	0,8809	0,8961	0,8986
	15	-	0,5687	0,7774	0,8541	0,8825	0,8972	0,8994
	20	-	-	0,7778	0,8546	0,8829	0,8975	0,8995

Из сравнения точного и приближенных решений можно заметить, что формула Эйлера на каждом шаге интегрирования дает завышенное значение функции, в то время как видоизмененные формулы Эйлера заниженные;

формулы Эйлера наименее устойчивы при увеличении шага интегрирования – в них колебательность решения проявляется уже при $h = 7$, а формулы Рунге – Кутта – при $h = 15$.

Интегрирование системы дифференциальных уравнений. Методы интегрирования одного дифференциального уравнения могут быть распространены и на системы обыкновенных дифференциальных уравнений. Пусть задана система двух уравнений

$$y' = f(x, y, z), \quad z' = g(x, y, z) \quad (1.3)$$

с начальными условиями в точке $x = x_0$; $y|_{x=x_0} = y_0$; $z|_{x=x_0} = z_0$

Для того чтобы проинтегрировать систему (2.3), например, по формулам Рунге—Кутта четвертого порядка, достаточно применить формулы (1 - 2.30) к каждому из уравнений

Для того, чтобы проинтегрировать систему (2.24), например, по формулам Рунге-Кутта 4-го порядка, достаточно применить формулы (2.22) к каждому из уравнений

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ z_{i+1} &= z_i + \frac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4) \end{aligned} \quad (1.4)$$

где k_i и m_i от, вычисляются по соответствующим формулам

$$k_1 = hf(x_i, y_i, z_i) \quad (1.5)$$

$$k_2 = hf(x_i + h/2, y_i + k_1/2, z_i + m_1/2), \quad (1.6)$$

$$k_3 = hf(x_i + h/2, y_i + k_2/2, z_i + m_2/2), \quad (1.7)$$

$$k_4 = hf(x_i + h, y_i + k_3, z_i + m_3), \quad (1.8)$$

$$m_1 = hg(x_i, y_i, z_i) \quad (1.9)$$

$$m_2 = hg(x_i + h/2, y_i + k_1/2, z_i + m_1/2), \quad (1.10)$$

$$m_3 = hg (x_i + h / 2, y_i + k_2 / 2, z_i + m_2 / 2), \quad (1.11)$$

$$m_4 = hg (x_i + h, y_i + k_3, z_i + m_3, \quad (1.12)$$

Конечно-разностные формулы. Формулы Рунге—Кутта четвертого порядка получили наибольшее распространение в практике интегрирования дифференциальных уравнений с использованием вычислительных машин. Однако даже при относительно высокой точности их применение связано со значительным объемом вычислений, особенно если правые части уравнений являются сложными выражениями. Основным недостатком этих формул является то, что приходится вычислять три-четыре значения функции и усреднять на каждом шаге интегрирования. Прогноз решения осуществляется исходя из информации лишь в данной точке, и совсем не используется информация о решении в предыдущих точках. В прикладных задачах, например, связанных с нестационарными процессами, решения часто представляют собой монотонные функции, приближающиеся к стационарному состоянию, причем значительные изменения тангенса угла наклона интегральной кривой наблюдаются только на начальном участке интегрирования. Поэтому для вычисления значения интегральной кривой в последующей точке иногда целесообразно аппроксимировать решение, используя информацию о нем в предыдущих точках, т. е. его предысторию.

Методы, в основе которых используется информация о решении в ряде предшествующих точек, называются конечно-разностными методами или методами прогноза и коррекции. В отличие от формул Рунге - Кутта, в этих методах на каждом шаге интегрирования правые части уравнений вычисляются один или два раза, а разность между прогнозируемым и скорректированным решениями дает оценку точности интегрирования и может быть использована для контроля величины шага.

При интегрировании конечно-разностными методами наибольшее распространение получили формулы, в которых решение аппроксимируется алгебраическими полиномами. В частности, формулы Ньютона - для интерпо-

лирования назад (формула 3.11-3.13) используются в методе Адамса, а формулы Ньютона для интерполирования вперед - в методе Милна. Рассмотрим порядок получения формул интегрирования для дифференциального уравнения первого порядка

$$y' = f(x, y), \quad (1.13)$$

для которого задано начальное условие $y|_{x=x_0} = y_0$.

Предположим, что некоторым образом удалось получить приближенные значения $y = y(x)$ для ряда равноотстоящих точек

Тогда приближенно можно считать, что

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} \varphi(x) dx \quad (1.14)$$

где $\varphi(x)$ — приближенное представление правой части уравнения (1.13), основанное на аппроксимации решения по найденным точкам т. е. $\varphi(x) = f(x, y(x))$,

Метод Адамса. Воспользовавшись интерполяционной формулой Ньютона (1- 3.11—1 - 3.23), с точностью до разностей третьего порядка включительно в выражении (6.14) можно записать

$$\varphi(x) = y'_i + t\Delta^1 y'_{i-1} + \frac{t(t+1)}{2!} \Delta^2 y_{i-2} + \frac{t(t+1)(t+2)}{3!} \Delta^3 y_{i-3} \quad (1.15)$$

где $i = (x - x_i) / h$

тогда

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} (y'_i + t\Delta^1 y'_{i-1} + \frac{t(t+1)}{2!} \Delta^2 y_{i-2} + \frac{t(t+1)(t+2)}{3!} \Delta^3 y_{i-3}) dx \quad (1.16)$$

Если в последнем выражении произвести замену переменной $dx = hdt$ то можно заметить, что многочлен $\varphi(x)$ не зависит ни от шага интегрирования h , ни от i , и, следовательно,

$$y_{i+1} = y_i + \int_0^1 (y'_i + t\Delta^1 y'_{i-1} + \frac{t(t+1)}{2!} \Delta^2 y'_{i-2} + \frac{t(t+1)(t+2)}{3!} \Delta^3 y'_{i-3}) dx \quad (1.17)$$

Почтенно интегрируя выражение (48), получим экстраполяционную формулу Адамса

$$y_{i+1} = y_i + h(y'_i + 1/2 \Delta^1 y'_{i-1} + 5/12 \Delta^2 y'_{i-2} + 2/8 \Delta^3 y'_{i-3}) \quad (1.18)$$

Если учесть, что

$$\Delta^1 y'_{i-1} = y'_i - y'_{i-1} \quad (1.19)$$

$$\Delta^2 y'_{i-1} = y'_i - 2y'_{i-1} + y'_{i-2} \quad (1.20)$$

$$\Delta^3 y'_{i-1} = y'_i - 3y'_{i-1} + 3y'_{i-2} - y'_{i-3} \quad (1.21)$$

то окончательно формула Адамса запишется в виде

$$y_{i+1} = y_i + h/24 (55 y'_i + 59 y'_{i-1} + 37 y'_{i-2} - 9 y'_{i-3}) \quad (1.22)$$

Формула (1.22) получена, исходя из интерполяционного многочлена, включающего разности третьего порядка. Аналогично могут быть получены формулы Адамса с другим числом точек интерполирования. Так, например, если ограничиться разностями первого порядка, то в результате будет получена формула Эйлера, Однако чем выше порядок интерполяционного многочлена, тем больше точек необходимо знать для начала вычислительного процесса.

В этом смысле конечно-разностные формулы не являются само начинающимися, поэтому для их применения необходимо знать решение в нескольких предыдущих точках, которое может быть получено любым другим ранее рассмотренным методом. Для применения формулы (1.22) таких точек должно быть четыре: y_0, y_1, y_2, y_3 , Только после этого, воспользовавшись уравнением (1.13) для вычисления соответствующих производных, можно применить формулу Адамса (1.22).

Точность разностных формул зависит от характера изменения разностей. Если третья разность, например, постоянна, то интегрирование по (1.22) будет точным. При работе на вычислительных машинах такой способ

оценки точности решения в принципе возможен, однако применяется редко. Чаще используется одновременное интегрирование по двум формулам различного порядка с последующим сравнением результатов. Если решения различается сильно, то шаг уменьшается и процесс повторяется.[1,3]

Текст программы:

```
unit Unit1;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
    Dialogs, StdCtrls, ExtCtrls, Buttons, Grids,
OleCtrls, Menus;
type
    TForm1 = class(TForm)
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        Label4: TLabel;
        Edit1: TEdit;
        Edit2: TEdit;
        Edit3: TEdit;
        Edit4: TEdit;
        BitBtn1: TBitBtn;
        BitBtn2: TBitBtn;
        BitBtn3: TBitBtn;
        Label7: TLabel;
        Label8: TLabel;
```

```

Label9: TLabel;
Label10: TLabel;
StringGrid2: TStringGrid;
MainMenu1: TMainMenu;
Fayl1: TMenuItem;
Hisoblash1: TMenuItem;
ozalash1: TMenuItem;
Natjalarnisaqlah1: TMenuItem;
Dasturhaqidal1: TMenuItem;
Yordam1: TMenuItem;
StringGrid1: TStringGrid;
StringGrid3: TStringGrid;
procedure BitBtn2Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure Hisoblash1Click(Sender: TObject);
procedure          Natjalarnisaqlah1Click(Sender:
TObject);
    procedure ozalash1Click(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var
Key: Char);
        procedure Edit2KeyPress(Sender: TObject; var
Key: Char);
            procedure Edit3KeyPress(Sender: TObject; var
Key: Char);
                procedure Edit4KeyPress(Sender: TObject; var
Key: Char);

```

```
private
    { Private declarations }
public
    { Public declarations }
end;
```

```
Const max=50;
```

```
Var
```

```
Form1: TForm1;
a,b,h:real;
k1,k2,k3,k4,dy:extended;
n1,x1,n,i,j,k:integer;
x0,f,xf,v,t:Real;
x:real;
q1,q2,q3,q4:Real;
```

```
implementation
```

```
{ $R *.dfm }
```

```
Function funksiya(f,v,xf,x,y:real):real;
begin
    Result:=(f/v)*(xf-y);
end;
```

```
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
    Edit1.Clear;
```

```

Edit2.Clear;
Edit3.Clear;
Edit4.Clear;
for i:=1 to 9 do
  for j:=1 to 4 do
    begin
      StringGrid1.Cells[i,j]:='';
      StringGrid2.Cells[i,j]:='';
      StringGrid3.Cells[i,j]:='';
    end;
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
  Edit1.Text:='0'; //x0
  Edit2.Text:='1'; //F
  Edit3.Text:='0,9'; //XF
  Edit4.Text:='5'; // V
  StringGrid1.Cells[0,0]:='H/t';
  StringGrid2.Cells[0,0]:='H/t';
  StringGrid3.Cells[0,0]:='H/t';
  j:=1;
  for i:=1 to 9 do
    begin
      StringGrid1.Cells[i,0]:=IntToStr(j);
      StringGrid2.Cells[i,0]:=IntToStr(j);
      StringGrid3.Cells[i,0]:=IntToStr(j);
      j:=i*5;
    end;
end;

```

```

j:=5;
for i:=1 to 4 do
  begin
    j:=i*5;
    StringGrid1.Cells[0,i]:=IntToStr(j);
    StringGrid2.Cells[0,i]:=IntToStr(j);
    StringGrid3.Cells[0,i]:=IntToStr(j);
  end;

end;

procedure TForm1.BitBtn3Click(Sender: TObject);
begin
  Close;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  if (Edit1.Text='') or (Edit2.Text='') or
(Edit3.Text='') or (Edit4.Text='')
  then ShowMessage('Ma`lumotlar yetarli emas!')
  else
  begin
    x0:=StrToFloat(Edit1.Text);
    f:=StrToFloat(Edit2.Text);
    xf:=StrToFloat(Edit3.Text);
    v:=StrToFloat(Edit4.Text);
    // Eyler suli
    t:=1;
  end;
end;

```

```

for k:=1 to 9 do
begin
for i:=1 to 4 do
begin
h:=t/(i*5);
x:=x0;
t:=0;
for j:=1 to i*5 do
begin
t:=t+h;
x:=x+h*funksiya(f,v,xf,t,x);
end;
StringGrid1.Cells[k,i]:=FloatToStrF(x,
ffGeneral, 8, 6);
end;
t:=k*5;
end;

// Runge -Kutta usuli
t:=1;
for k:=1 to 9 do
begin
for i:=1 to 4 do
begin
h:=t/(i*5);
x:=x0;
t:=0;
for j:=1 to i*5 do
begin

```

```

        q1:=h*funksiya(f,v,xf,t,x);
        q2:=h*funksiya(f,v,xf,t+h/2,x+q1/2);
        q3:=h*funksiya(f,v,xf,t+h/2,x+q2/2);
        q4:=h*funksiya(f,v,xf,t+h,x+q3);
        x:=x+(q1+2*q2+2*q3+q4)/6;
        t:=t+h;
    end;
    StringGrid2.Cells[k,i]:=FloatToStrF(x,
ffGeneral, 8, 6);
    end;
    t:=k*5;
end;

// Modifikatsion Eyler sulj
t:=1;
for k:=1 to 9 do
begin
    for i:=1 to 4 do
    begin
        h:=t/(i*5);
        x:=x0;
        t:=0;
        for j:=1 to i*5 do
            begin
                t:=t+h;
                x:=x+h*funksiya(f,v,xf,t+h/2,x+h*funksiya(f,v,xf,t,x)/2
);
            end;
        end;
    end;
end;

```

```

        StringGrid3.Cells[k,i]:=FloatToStrF(x,
ffGeneral, 8, 6);
        end;
        t:=k*5;
        end;
    end;
end;

procedure TForm1.Hisoblash1Click(Sender: TObject);
begin
    BitBtn1.Click;
end;

procedure      TForm1.Natjalarnisaqlah1Click(Sender:
TObject);
begin
    BitBtn3.Click;
end;

procedure TForm1.ozalash1Click(Sender: TObject);
begin
    BitBtn2.Click;
end;

procedure TForm1.Edit1KeyPress(Sender: TObject; var
Key: Char);
begin
    if Key=#13
    then Edit2.SetFocus;
end;

```

```
end;

procedure TForm1.Edit2KeyPress(Sender: TObject; var
Key: Char);
begin
    if Key=#13
    then Edit3.SetFocus;
end;

procedure TForm1.Edit3KeyPress(Sender: TObject; var
Key: Char);
begin
    if Key=#13
    then Edit4.SetFocus;
end;

procedure TForm1.Edit4KeyPress(Sender: TObject; var
Key: Char);
begin
    if Key=#13
    then BitBtn1.SetFocus;
end;
end.
```

Результаты расчётов

DASTUR
Eayl Dastur haqida Yordam

X0 ni kiriting: XF ni kiriting:

F ni kiriting: V ni kiriting:

Hisoblash Tozalash Chiqish

Natijalar:

Eyler usuli:

H/t	1	5	10	15	20	25	30	35	40
5	0,16616457	0,605088	0,830016	0,890784	0,899712	0,9	0,900288	0,909216	0,969984
10	0,16463447	0,5861894	0,80336324	0,87457723	0,89455804	0,89912109	0,89990563	0,89999469	0,89999991
15	0,16413295	0,58026207	0,79479802	0,86833407	0,89141453	0,89794471	0,89957683	0,89992769	0,89999024
20	0,16388376	0,57736267	0,79058101	0,86511642	0,88962371	0,89714591	0,89928187	0,89983688	0,89996709

Runge - Kutta usuli:

H/t	1	5	10	15	20	25	30	35	40
5	0,16314232	0,56890329	0,77812559	0,85495093	0,88307015	0,89332581	0,89705485	0,89837397	0,898699
10	0,16314232	0,5689082	0,77819441	0,85517998	0,88349625	0,89391179	0,89774509	0,89915838	0,89968153
15	0,16314232	0,56890845	0,77819753	0,85518952	0,88351245	0,89393172	0,89776513	0,89917594	0,89969553
20	0,16314232	0,56890848	0,77819802	0,855191	0,88351489	0,89393463	0,89776796	0,89917834	0,89969736

Modifikatsiyalangan Eyler usuli usuli:

H/t	1	5	10	15	20	25	30	35	40
5	0,16310182	0,56633414	0,76914598	0,84092789	0,86578164	0,871875	0,86578164	0,84092789	0,76914598
10	0,16313235	0,56831311	0,77629677	0,85259706	0,88097469	0,89181455	0,89612276	0,89791931	0,898699
15	0,16313791	0,56865056	0,77739784	0,85413828	0,88253389	0,89317182	0,89723385	0,89882639	0,89947239
20	0,16313985	0,56876524	0,77775979	0,8546239	0,88299723	0,89354282	0,89750329	0,89901216	0,89959782

Заключение

Третий раздел посвящен составлению оптимального алгоритма решения задач химии и химической технологии. Полученные результаты сравниваются с экспериментальными данными.

Кроме этого, рассматривается численная сходимость применяемых методов, для решения задач химии и химической технологии.

Заключение

В введении приводится постановка и актуальность задачи, основная цель и методы исследования, научно-практическая ценность и основное содержание магистерской диссертационной работы.

В 1.1 приводится математическое и физическое моделирование и задачи приводящие к ним. Здесь раскрывается основная задача физического и математического моделирования и некоторые принципы подобия.

1.2 посвящен методам численного интегрирования дифференциальных уравнений. Здесь приводится оптимальный алгоритм, который может быть использован при интегрировании любого дифференциального уравнения с произвольной наперед заданной точностью, от которой будет зависеть число членов ряда и объема вычисления, поэтому обычно при разложении используются методы, опирающиеся на разложение в ряд Тейлора, но позволявшие получить решение без вычисления производных.

В 1.3 разработан оптимальный алгоритм и блок схемы формул Рунге-Кутты и Эйлера и основное преимущество этих методов, а также графическая иллюстрация этих методов.

Кроме этого дается оценка точности формул интегрирования и методика определения устойчивости решения.

Наибольшее распространение в вычислительной практике имеют формулы Рунге-Кутты четвертого порядка. Точность формул четвертого порядка пропорциональна h^5 , т.е. разработанные алгоритмы метода соответствуют разложению решения в ряд Тейлора до членов h^4 включительно.

По оценке интегрирования по формулам $y_{m+1} = y_m + 1/6(k_1 + 2k_2 + 2k_3 + k_4) + O(h^5)$ на 20 % быстрее, чем по формулам с постоянным шагом.

В 2.1 рассмотрена математическая модель нестационарных режимов тарельчатой ректификационной колонны.

Математическое описание динамики, ректификационной колонны содержит: уравнения материального и теплового балансов. При решении практических задач был принят ряд допущений, исходя из принятых допущений математическое описание ректификационной колонны представляет систему дифференциальных уравнений, которые решаются методом Эйлера и Рунге-Кутты. Приводится сравнительный анализ этих методов на компьютере. Составлена программа расчёта для произвольных исходных данных и рекомендации по использованию полученных результатов.

В 2.2 рассматривается математическая модель реактора идеального вытеснения и её численное решение.

Приводится стационарный режим работы реактора описывается обыкновенными дифференциальными уравнениями, определяющими изменение концентраций реагентов и температуры по длине зоны режима.

При решении уравнения (5.1), (5.2) принимаются некоторые допущения, а потом решаются методом Рунге-Кутты 4-го порядка с некоторыми изменениями, внесенными на стадии отладки и при вычислении правых частей.

Полученные результаты сравнивают с данными, полученными экспериментальным путем и они совпадают с тремя знаками точности.

3.1 посвящен сравнительной оценке различных формул интегрировании. При нахождении концентрации на выходе сосуда, используются формулы Эйлера и Рунге-Кутты, которые описываются обыкновенными дифференциальными уравнениями.

Полученное уравнение сначала решается точными методами, а потом решается приближенными методами Рунге-Кутты, Эйлера, модифицированным методом Эйлера.

Из сравнения точного и приближенного решения можно заметить, что формула Эйлера на каждом шаге интегрирования дает завышенные значения функции, в то время как видоизмененные формулы Эйлера заниженные; формулы Эйлера наименее устойчивы при увеличении шага, а формулы Рунге-Кутта более устойчивы.

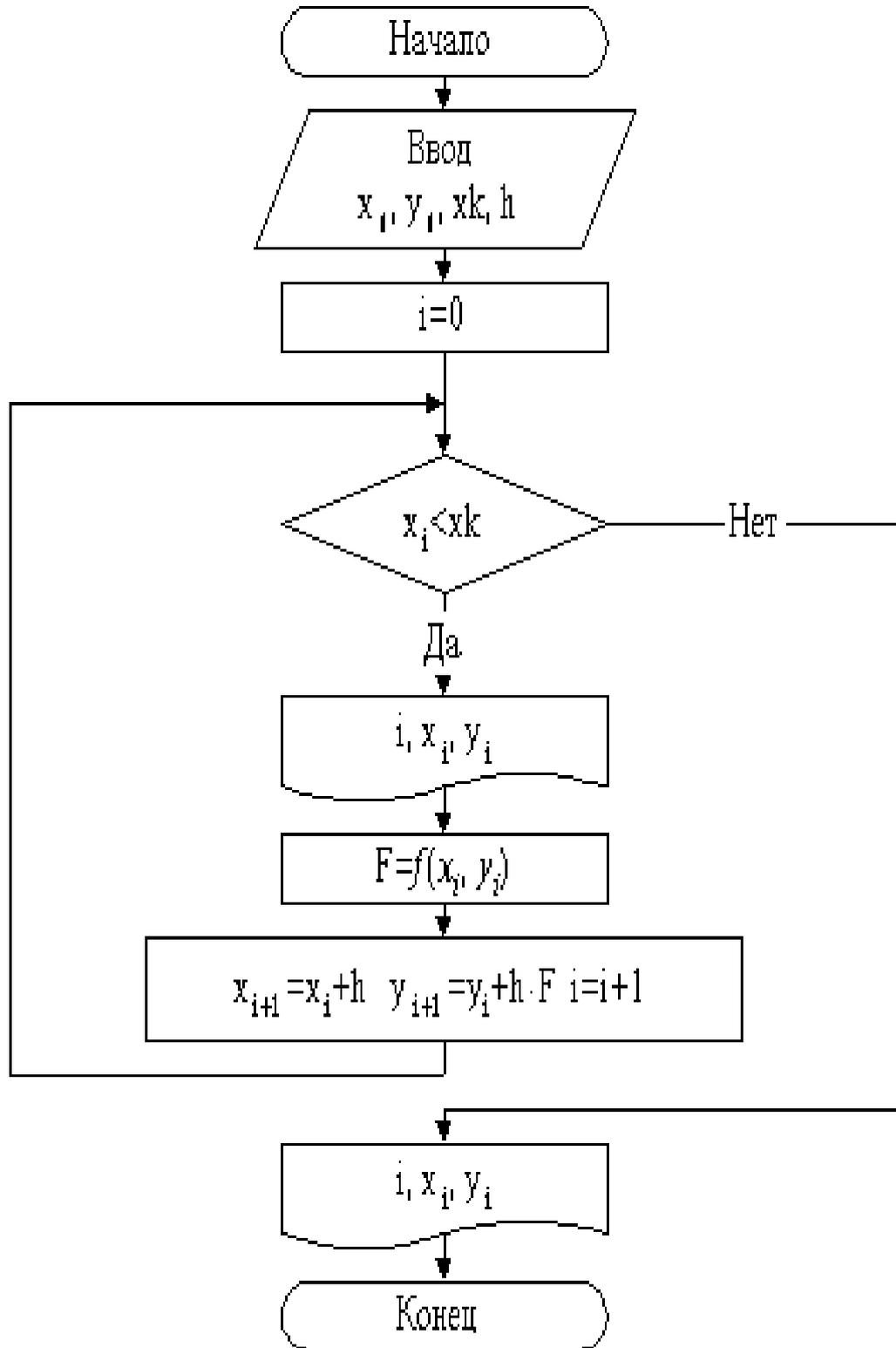
Далее в этом параграфе рассматриваются конечно-разностные методы или так называемые методы прогноза и коррекции. В отличие от формул Рунге-Кутта, в этих методах на каждом шаге интегрирования правые части уравнений вычисляются один или два раза, а разность между прогнозируемыми и скорректированными решениями дает оценку точности интегрирования и может быть использована для контроля величины шага.

ЛИТЕРАТУРА

1. В. Кафаров, В.Н. Ветохин, А.И. Бояринов. «Программирование и вычислительные методы в химии и химической технологии». –Москва: Изд-во «Наука», 1972.
2. М. Исроилов. «Хисоблаш методлари». -Тошкент, 1996.
3. Кузичкин Н.В, Саутин С.Н., Пунин А.Е., Холоднов В.А., Шибает В.А. “Методы и средства автоматизированного расчета химико-технологических производств”.
4. Вержбицкий В. М. Основы численных методов. – М.: Высшая школа, 2009.
5. Жидков В. Н. Вычислительная математика. – М, Академия, 2010.
6. Калиткин Н. Н. Численные методы. -С.Пб.: Изд-во БХВ-Петербург, 2011.
7. Марчук Г. И. Методы вычислительной математики. –М.: Изд-во Лань,2010.
8. Пантина И.В., Синчуков А. В. Вычислительная математика. –М, 2010.
9. Самарский А. А. Введение в численные методы–М.: Изд-во Лань,2009.
10. Турчак Л. И., Плотников П. В. Основы численных методов.- М.: ФИЗМАТЛИТ, 2002.
11. М. Исроилов Хисоблаш методлари. II-қисм. Т. 2006.
12. Бехвалов Н. С. Численные методы. М. Наука, 1973.
13. Березин Н. С., Жидков Н.П. Методы вычислений. М. Наука, 1966.
14. Демидович Б.П., Марон И. А. Основы вычислительной математики. М.: Наука,1970.
15. Амридинов С.А. Хисоблаш математикаси., ўқув предмети бўйича таълим технологиялари. СамДУ услубий қўлланма.Самарқанд, 2010.
16. Амридинов С.А. Хисоблаш математикаси., фанидан лаборатория ишларини бажариш бўйича услубий қўлланма.Самарқанд, 2008.

Приложение

Блок-схема метода Эйлера



```
unit Unit1;

interface
uses Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Buttons, Grids, OleCtrls,
Menus;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    StringGrid1: TStringGrid;
    StringGrid2: TStringGrid;
```

```

StringGrid3: TStringGrid;
OleContainer1: TOleContainer;
MainMenu1: TMainMenu;
Fayl1: TMenuItem;
Hisoblash1: TMenuItem;
Tozalash1: TMenuItem;
Natijalarnisaqlash1: TMenuItem;
Dasturhaqidal1: TMenuItem;
Yordam1: TMenuItem;
procedure BitBtn2Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure Hisoblash1Click(Sender: TObject);
procedure Natijalarnisaqlash1Click(Sender: TObject);
procedure Tozalash1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  i,j,n:Integer;
  a,b,x0,y0,h,x,y,eps,y1,z,q1,q2,q3,q4:real;
  tick:Cardinal;

implementation
{$R *.dfm}

```

```
function ff(x,y:real):real;
begin
  Result:=x+sin(y/3);
end;
```

```
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
  Edit1.Clear;
  Edit2.Clear;
  Edit3.Clear;
  Edit4.Clear;
  Edit5.Clear;
  for i:=0 to 2 do
    for j:=1 to n+1 do
      StringGrid1.Cells[i,j]:='';
  for i:=0 to 2 do
    for j:=1 to n+1 do
      StringGrid2.Cells[i,j]:='';
  for i:=0 to 2 do
    for j:=1 to n+1 do
      StringGrid3.Cells[i,j]:='';
end;
```

```
procedure TForm1.FormActivate(Sender: TObject);
begin
  Edit1.Text:='0';
  Edit2.Text:='1';
  Edit3.Text:='10';
  Edit4.Text:='0';
```

```

Edit5.Text:='1';
Label6.Caption:='Qadam: '+'0,1';
StringGrid1.Cells[0,0]:='I';
StringGrid1.Cells[1,0]:='X[i]';
StringGrid1.Cells[2,0]:='Y[i]';
StringGrid2.Cells[0,0]:='I';
StringGrid2.Cells[1,0]:='X[i]';
StringGrid2.Cells[2,0]:='Y[i]';
StringGrid3.Cells[0,0]:='I';
StringGrid3.Cells[1,0]:='X[i]';
StringGrid3.Cells[2,0]:='Y[i]';
end;

procedure TForm1.BitBtn3Click(Sender: TObject);
begin
    Close;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
    a:=StrToFloat(Edit1.Text);
    b:=StrToFloat(Edit2.Text);
    n:=StrToInt(Edit3.Text);
    StringGrid1.RowCount:=n+1;
    StringGrid2.RowCount:=n+1;
    StringGrid3.RowCount:=n+1;
    x0:=StrToFloat(Edit4.Text);
    y0:=StrToFloat(Edit5.Text);
    h:=(b-a)/n;
    x:=x0;y:=y0;

```

```

Label6.Caption:='Qadam: '+FloatToStr(h);
tick:=GetTickCount;
for i:=1 to n do
begin
  y:=y+h*ff(x,y);
  x:=h+x;
  StringGrid1.Cells[0,i]:=IntToStr(i);
  StringGrid1.Cells[1,i]:=FloatToStr(x);
  StringGrid1.Cells[2,i]:=FloatToStr(y);
end;
tick:=GetTickCount-tick;
Label8.Caption:='Eyler usuli: '+FloatToStr(tick)+'
msek';
x:=x0;y:=y0;eps:=0.001;
tick:=GetTickCount;
for i:=1 to n do
begin
  y1:=y0+h*ff(x,y);
  z:=y1;
  y:=y+(ff(x,y)+ff(x+h,z))*h/2;
  while abs(y-z)<eps do
  begin
x:=x+h;
z:=y1;
y:=y1+(ff(x,y1)+ff(x,z))*h/2;
y1:=y;
end;
x:=x+h;
  StringGrid2.Cells[0,i]:=IntToStr(i);
  StringGrid2.Cells[1,i]:=FloatToStr(x);

```

```

StringGrid2.Cells[2,i]:=FloatToStr(y);
end;
tick:=GetTickCount-tick;
Label9.Caption:='Eyler ketma-ket yaqinlashish usuli:
'+FloatToStr(tick)+' msek';
tick:=GetTickCount;
x:=x0;y:=y0;
for i:=1 to n do
begin
q1:=h*ff(x,y);
q2:=h*ff(x+h/2,y+q1/2);
q3:=h*ff(x+h/2,y+q2/2);
q4:=h*ff(x+h,y+q3);
y:=y+(q1+2*q2+2*q3+q4)/6;
x:=x+h;
StringGrid3.Cells[0,i]:=IntToStr(i);
StringGrid3.Cells[1,i]:=FloatToStr(x);
StringGrid3.Cells[2,i]:=FloatToStr(y);
end;
tick:=GetTickCount-tick;
Label10.Caption:='Runge-Kutta usuli:
'+FloatToStr(tick)+' msek';
end;

procedure TForm1.Hisoblash1Click(Sender: TObject);
begin
BitBtn1.Click;
end;

```

```
procedure TForm1.Natjalarni saqlah1Click(Sender:
TObject);
begin
    BitBtn3.Click;
end;

procedure TForm1.Tozalash1Click(Sender: TObject);
begin
    BitBtn2.Click;
end;
end.
```

Результаты расчётов

Oddiy differensial tenglamani yechish

Eyil Dastur haqida Yordam

$y' = x + \sin\left(\frac{y}{3}\right)$ birinchi tartibli differensial tenglamaning $x_0=0, y_0=1$ boshlang'ich shartni qanoatlantiruvchi $[0,1]$ oraliqda yechimini $h=0,1$ qadami bilan $\epsilon=0,001$ aniqlikda hisoblash.

A ni kiriting: X[0] ni kiriting:

B ni kiriting: Y[0] ni kiriting:

N ni kiriting: Qadam: 0,1

Hisoblash Tozalash Chiqish

Natijalar:

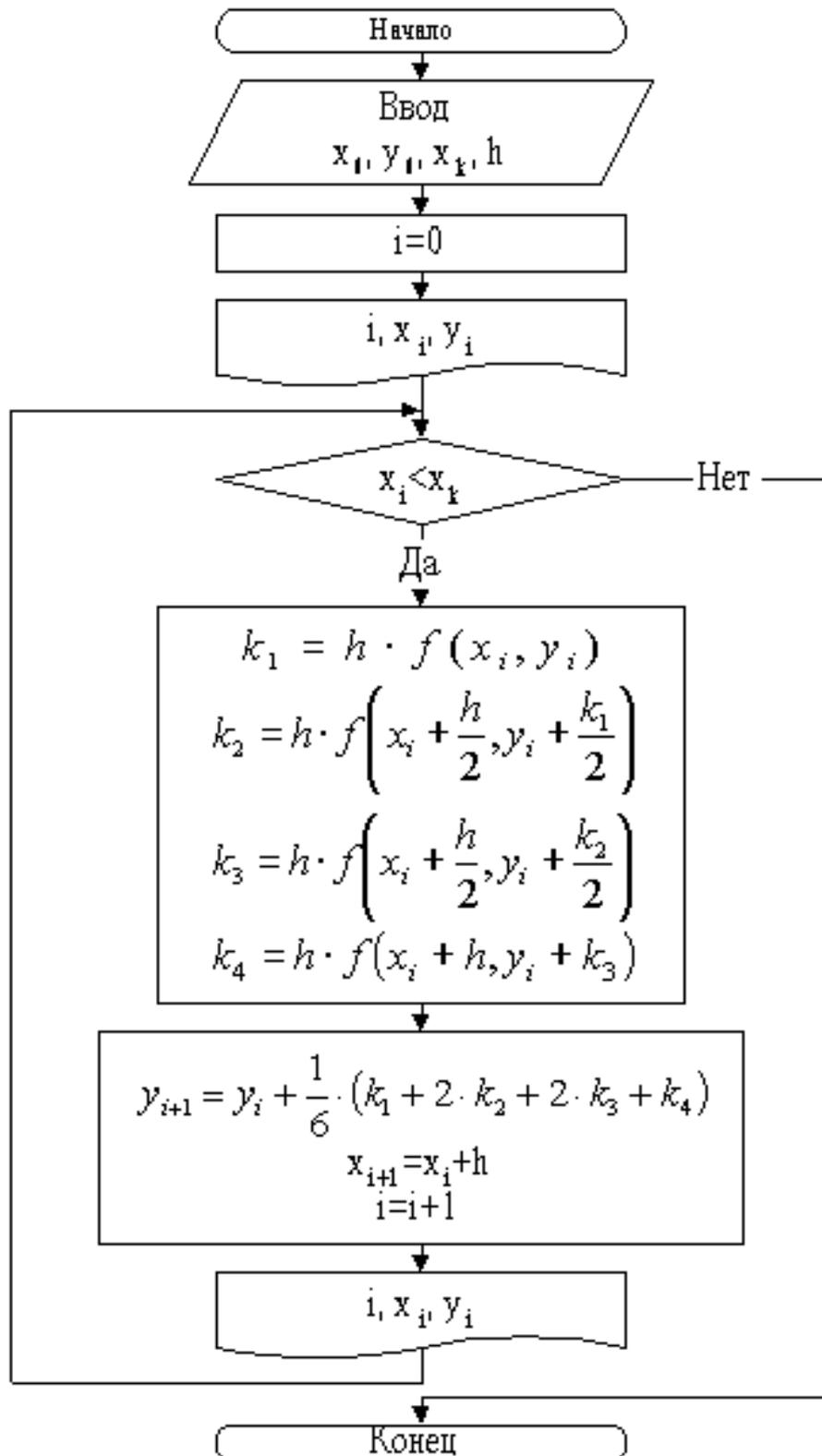
Eyler usuli: 0 msek Eyler ketma-ket yaqinlashish usuli: 0 msek Runge-Kutta usuli: 0 msek

I	X[i]	Y[i]
1	0,1	1,032719469
2	0,2	1,076467589
3	0,3	1,131584789
4	0,4	1,198416193
5	0,5	1,277309396
6	0,6	1,368611593
7	0,7	1,472665931
8	0,8	1,589806910
9	0,9	1,720354690
10	1	1,864608161

I	X[i]	Y[i]
1	0,1	1,038233794
2	0,2	1,087244021
3	0,3	1,147200916
4	0,4	1,218274028
5	0,5	1,300631053
6	0,6	1,394436405
7	0,7	1,499849488
8	0,8	1,617022621
9	0,9	1,746098576
10	1	1,887207695

I	X[i]	Y[i]
1	0,1	1,038292299
2	0,2	1,087962296
3	0,3	1,149361434
4	0,4	1,222843772
5	0,5	1,308763001
6	0,6	1,407468635
7	0,7	1,519301208
8	0,8	1,644586322
9	0,9	1,783627398
10	1	1,936696999

Блок-схема метода Рунге-Кутты



Текст программы:

```
unit Unit1;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes,
    Graphics, Controls, Forms,
    Dialogs, StdCtrls, ExtCtrls, Buttons, Grids, OleCtrls,
    Menus;

type

    TForm1 = class(TForm)
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        Label4: TLabel;
        Label5: TLabel;
        Label6: TLabel;
        Edit1: TEdit;
        Edit2: TEdit;
        Edit3: TEdit;
        Edit4: TEdit;
        Edit5: TEdit;
        BitBtn1: TBitBtn;
        BitBtn2: TBitBtn;
        BitBtn3: TBitBtn;
        Label7: TLabel;
        Label8: TLabel;
        Label9: TLabel;
    end;
```

```

Label10: TLabel;
StringGrid1: TStringGrid;
StringGrid2: TStringGrid;
StringGrid3: TStringGrid;
OleContainer1: TOleContainer;
MainMenu1: TMainMenu;
Fayl1: TMenuItem;
Hisoblash1: TMenuItem;
ozalash1: TMenuItem;
Natijalarnisaqlash1: TMenuItem;
Dasturhaqidal1: TMenuItem;
Yordam1: TMenuItem;
procedure BitBtn2Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure Hisoblash1Click(Sender: TObject);
procedure Natijalarnisaqlash1Click(Sender: TObject);
procedure ozalash1Click(Sender: TObject);
procedure Edit1KeyPress(Sender: TObject; var Key:
Char);
    procedure Edit2KeyPress(Sender: TObject; var Key:
Char);
        procedure Edit3KeyPress(Sender: TObject; var Key:
Char);
            procedure Edit4KeyPress(Sender: TObject; var Key:
Char);
                procedure Edit5KeyPress(Sender: TObject; var Key:
Char);
                    private

```

```

        { Private declarations }
public
        { Public declarations }
end;
Const max=50;

Var
    Form1: TForm1;
    KY,KZ:array[1..4] of real;
    a,b,h:real;
    k1,k2,k3,k4,dy:extended;
    x,k,z,y:array[ 1 ..max] of extended;
    n1,x1,n,i,j:integer;

implementation
{$R *.dfm}

// Y hasilasi:
Function f1(y,x,z:real):real;
begin
    f1:=z;
end;

// Z hasilasi:

Function f2(y,x,z:real):real;
begin
    f2:=4-z/x+y/(x*x);
end;

procedure TForm1.BitBtn2Click(Sender: TObject);

```

```

begin
  Edit1.Clear;
  Edit2.Clear;
  Edit3.Clear;
  Edit4.Clear;
  Edit5.Clear;
  for i:=0 to 3 do
    for j:=1 to n+1 do
      StringGrid1.Cells[i,j]:= '';
  for i:=0 to 3 do
    for j:=1 to n+1 do
      StringGrid2.Cells[i,j]:= '';
  for i:=0 to 3 do
    for j:=1 to n+1 do
      StringGrid3.Cells[i,j]:= '';
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
  Edit1.Text:='1';
  Edit2.Text:='1,5';
  Edit3.Text:='10';
  Edit4.Text:='3';
  Edit5.Text:='1,8';
  Label6.Caption:='Qadam: '+'0,1';
  StringGrid1.Cells[0,0]:='I';
  StringGrid1.Cells[1,0]:='X[i]';
  StringGrid1.Cells[2,0]:='Y[i]';
  StringGrid1.Cells[3,0]:='Z[i]';
  StringGrid2.Cells[0,0]:='I';

```

```

StringGrid2.Cells[1,0]:='X[i]';
StringGrid2.Cells[2,0]:='Y[i]';
StringGrid2.Cells[3,0]:='Z[i]';
StringGrid3.Cells[0,0]:='I';
StringGrid3.Cells[1,0]:='X[i]';
StringGrid3.Cells[2,0]:='Y[i]';
StringGrid3.Cells[3,0]:='Z[i]';
end;

procedure TForm1.BitBtn3Click(Sender: TObject);
begin
    Close;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
    if (Edit1.Text='') or (Edit2.Text='') or
(Edit3.Text='') or (Edit4.Text='') or (Edit5.Text='')
    then ShowMessage('Ma`lumotlar yetarli emas!')
    else
    begin
        a:=StrToFloat(Edit1.Text);
        b:=StrToFloat(Edit2.Text);
        n:=StrToInt(Edit3.Text);
        y[1]:=StrToFloat(Edit4.Text);
        z[1]:=strtofloat(edit5.Text);
        x[1]:=a; h:=(b-a)/n;
        Label6.Caption:='Qadam: H='+FloatToStr(h);
        StringGrid1.RowCount:=n+2;
        StringGrid2.RowCount:=n+2;
    end;
end;

```

```

StringGrid3.RowCount:=n+2;
//Runge-Kutta usuli:
for i:=2 to 4 do
  begin
    KY[1]:=h*f1(x[i-1],y[i-1],z[i-1]);
    KZ[1]:=h*f1(x[i-1],y[i-1],z[i-1]);
    KY[2]:=h*f1(x[i-1]+h/2,y[i-1]+KY[1]/2,z[i-1]+KZ[1]/2);
    KZ[2]:=h*f2(x[i-1]+h/2,y[i-1]+KY[1]/2,z[i-1]+KZ[1]/2);
    KY[3]:=h*f2(x[i-1]+h/2,y[i-1]+KY[2]/2,z[i-1]+KZ[2]/2);
    KZ[3]:=h*f2(x[i-1]+h/2,y[i-1]+KY[2]/2,z[i-1]+KZ[2]/2);
    KY[4]:=h*f2(x[i-1]+h,y[i-1]+KY[3],z[i-1]+KZ[3]);
    KZ[4]:=h*f2(x[i-1]+h,y[i-1]+KY[3],z[i-1]+KZ[3]);
    x[i]:=x[i-1]+h;
    y[i]:=y[i-1]+1/6*(KY[1]+2*KY[2]+2*KY[3]+KY[4]);
    z[i]:=z[i-1]+1/6*(KZ[1]+2*KZ[2]+2*KZ[3]+KZ[4]);
  end;
// Adams usuli
for i:=4 to n+1 do
  begin
    x[i+1]:=x[i]+h;
    dy:=(h/24)*(55*f1(x[i],y[i],z[i])-59*f1(x[i-1],y[i-1],z[i-1])+
    37*f1(x[i-2],y[i-2],z[i-2])-9*f1(x[i-3],y[i-3],z[i-3])));
    y[i+1]:=y[i]+dy;
  end;

```

```

        dy:=(h/24)*(55*f2(x[i],y[i],z[i])-59*f2(x[i-1],y[i-1],z[i-1])
+37*f2(x[i-2],y[i-2],z[i-2])-9*f2(x[i-3],y[i-3],z[i-3]));
        z[i+1]:=z[i]+dy;
    end;
    for i:=1 to n+1 do
    begin
        StringGrid1.Cells[0,i]:=IntToStr(i-1);

StringGrid1.Cells[1,i]:=FloatToStrF(x[i],ffGeneral,6,4);

StringGrid1.Cells[2,i]:=FloatToStrF(Y[i],ffGeneral,6,4);

StringGrid1.Cells[3,i]:=FloatToStrF(z[i],ffgeneral,6,4);
        end;
// Eyler!
    for i:=2 to n+1 do
    begin
        x[i]:=x[i-1]+h;
        y[i]:=y[i-1]+h*f1(x[i-1],y[i-1],z[i-1]);
        z[i]:=z[i-1]+h*f2(x[i-1],y[i-1],z[i-1]);
    end;
        for i:=1 to n+1 do
        begin
            StringGrid2.Cells[0,i]:=IntToStr(i-1);

StringGrid2.Cells[1,i]:=FloatToStrF(x[i],ffGeneral,6,4);

StringGrid2.Cells[2,i]:=FloatToStrF(Y[i],ffGeneral,6,4);

```

```

StringGrid2.Cells[3,i]:=FloatToStrF(z[i],ffGeneral,6,4);
    end;
//Runge!
    for i:=2 to n+1 do
        begin
            KY[1]:=h*f1(x[i-1],y[i-1],z[i-1]);
            KZ[1]:=h*f1(x[i-1],y[i-1],z[i-1]);
            KY[2]:=h*f1(x[i-1]+h/2,y[i-1]+KY[1]/2,z[i-
1]+KZ[1]/2);
            KZ[2]:=h*f2(x[i-1]+h/2,y[i-1]+KY[1]/2,z[i-
1]+KZ[1]/2);
            KY[3]:=h*f2(x[i-1]+h/2,y[i-1]+KY[2]/2,z[i-
1]+KZ[2]/2);
            KZ[3]:=h*f2(x[i-1]+h/2,y[i-1]+KY[2]/2,z[i-
1]+KZ[2]/2);
            KY[4]:=h*f2(x[i-1]+h,y[i-1]+KY[3],z[i-1]+KZ[3]);
            KZ[4]:=h*f2(x[i-1]+h,y[i-1]+KY[3],z[i-1 ]+KZ[3]);
            x[i]:=x[i-1]+h;
            y[i]:=y[i-1]+1/6*(KY[1]+2*KY[2]+2*KY[3]+KY[4]);
            z[i]:=z[i-1]+1/6*(KZ[1]+2*KZ[2]+2*KZ[3]+KZ[4]);
        end;
    for i:=1 to n+1 do
        begin
            StringGrid3.Cells[0,i]:=IntToStr(i-1);

StringGrid3.Cells[1,i]:=FloatToStrF(x[i],ffGeneral,6,4);

StringGrid3.Cells[2,i]:=FloatToStrF(Y[i],ffGeneral,6,4);

```

```

StringGrid3.Cells[3,i]:=FloatToStrF(z[i],ffGeneral,6,4);
    end;
    end;
end;

procedure TForm1.Hisoblash1Click(Sender: TObject);
begin
    BitBtn1.Click;
end;

procedure TForm1.Natijalarnisaqlash1Click(Sender:
TObject);
begin
    BitBtn3.Click;
end;

procedure TForm1.ozalash1Click(Sender: TObject);
begin
    BitBtn2.Click;
end;

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key:
Char);
begin
    if Key=#13
    then Edit2.SetFocus;
end;

```

```
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key:
Char);
begin
    if Key=#13
    then Edit3.SetFocus;
end;
```

```
procedure TForm1.Edit3KeyPress(Sender: TObject; var Key:
Char);
begin
    if Key=#13
    then Edit4.SetFocus;
end;
```

```
procedure TForm1.Edit4KeyPress(Sender: TObject; var Key:
Char);
begin
    if Key=#13
    then Edit5.SetFocus;
end;
```

```
procedure TForm1.Edit5KeyPress(Sender: TObject; var Key:
Char);
begin
    if Key=#13
    then BitBtn1.SetFocus;
end;
```

```
end.
```

Результаты расчётов

Ikkinchi tartibli differensial tenglamalar sistemasini yechish

Eyil Dastur haqida Yordam

$x^2y'' + xy' - y = 4x^2$ ikkinchi tartibli differensial tenglamaning $y(1)=3, y'(1)=1.8$ boshlang'ich shartlarni qanoatlantiruvchi $[1, 1.5]$ oraliqda h qadam bilan tugun nuqtalardagi qiymatlarini hisoblash.

A ni kiriting: Y[1] ni kiriting:

B ni kiriting: Z[1] ni kiriting:

N ni kiriting: Qadam: H=0,05

Natijalar:

Adams usuli:				Eylar usuli:				Runge usuli:			
I	X[i]	Y[i]	Z[i]	I	X[i]	Y[i]	Z[i]	I	X[i]	Y[i]	Z[i]
0	1	3	1,8	0	1	3	1,8	0	1	3	1,8
1	1,05	3,13295	1,96061	1	1,05	3,09	1,97556	1	1,05	3,13295	1,96061
2	1,1	3,26929	2,12137	2	1,1	3,18878	2,14909	2	1,1	3,26929	2,12137
3	1,15	3,40909	2,28238	3	1,15	3,29623	2,3208	3	1,15	3,40909	2,28238
4	1,2	3,52724	2,45329	4	1,2	3,41227	2,49089	4	1,2	3,55242	2,44377
5	1,25	3,65457	2,62242	5	1,25	3,53682	2,65954	5	1,25	3,69933	2,60561
6	1,3	3,78966	2,79081	6	1,3	3,66979	2,82694	6	1,3	3,8499	2,76801
7	1,35	3,93341	2,95802	7	1,35	3,81114	2,99325	7	1,35	4,00417	2,93104
8	1,4	4,08546	3,12438	8	1,4	3,9608	3,15863	8	1,4	4,16221	3,09477
9	1,45	4,24583	3,28998	9	1,45	4,11873	3,32322	9	1,45	4,32407	3,25927
10	1,5	4,41445	3,45497	10	1,5	4,28489	3,48715	10	1,5	4,4898	3,42462

