1. Понятие о моделировании.

2. Задание матриц и векторов, действия с квадратными матрицами, извлечение из матрицы строк и столбцов, разворачивание матрицы в вектор, создание матрицы значений функции

3. Виды моделирования. Математическое моделирование.

4. Графика в MATHCAD.

5. Численные методы как средство математического моделирования. Основные этапы и источники ошибок.

6. Понятие о методах типа Монте-Карло.

7. Необходимость тестирования компьютерной модели. Способы тестирования.

8. Отделение корня уравнения и уточнение корня уравнения. Метод бисекции (деления отрезка пополам).

9.Сравнение некоторых пакетов инженерных расчётов и моделирования.

10. Решение систем линейных алгебраических уравнений методом Гаусса.

11. Решение систем линейных алгебраических уравнений методом «прогонки».

12. Классификация методов решения задачи Коши.

13. Сведение произвольной системы ОДУ к системе ОДУ I-го порядка в нормализованном виде Коши. Пример.

14. Различные виды приближений данных.

15. Реализации в среде MathCad математической модели для простейшего осциллятора.

16. Приближение инженерных е интерполяциданных. Отличии и аппроксимации.

17. Пример использования Simulink MATLAB для решения ОДУ 2-го порядка.

18. Виды приближение инженерных данных. Аппроксимация, интерполяция, равномерное приближения

19. Получение модели колебательной системы со сосредоточенными параметрами на основе уравнений Лагранжа 2-го порядка.

20. Разложение аппроксиматора по системы базисных функций.

21. Пример получения уравнений движения для двухмассовой системы с поступательным движением масс.

22. Выполнение приближения по МНК на основе разложения аппроксиматора по системе базисных функций.

23. Примеры задач, приводящих к обыкновенным дифференциальным уравнениям.

24. Интегральная аппроксимация на отрезке по МНК

25. Методы решения задачи Коши для ОДУ.

26. Пример реализации MATHCAD метода аппроксимации по МНК.

27. Классификация методов решения задачи Коши для ОДУ. Одношаговые и многошаговые, явные и неявные методы.

28. Пример аппроксимации функции 2-х переменных в MATHCAD, значения которой заданы таблично.

29.Метод Эйлера и его модификации для решения ОДУ.

30. Алгебраическое интерполирование для приближения инженерных данных.

31 Основная идея методов Рунге-Кутта для решения ОДУ.

32 Интерполяционный полином Лагранжа.

33 Метод Рунге-Кутта 4-го порядка точности для решения ОДУ и его реализация в MathCad.

34 Примеры формул для моделирования случайных величин с разными законами распределения.

35 Оценка точности решения дифференциальных уравнений.

36 Использование случайных величин в задачах вычисления площадей и объемов

37 Способ Рунге для оценки погрешности ОДУ.

38 Метод Судзуо Какутани для решения граничных задач теории потенциала

39Эрмитовы кубические сплайны.

40Случайные блуждания при моделировании температурных полей.

41Понятие о жестких дифференциальных уравнениях.

42Эрмитов кубический сплайн.

43Метод стрельбы для решения граничных задач для ОДУ.

44Решение систем линейных алгебраических уравнений в MATHCAD.

**1.Понятие о моделировании.**

*Реальное явление – Представление человека об этом явлении- Модель явления -Предсказание- Сравнение с экспериментом – Уточнение модели*

*Модель*- искусственный объект, созданный человеком, заменяющий реальный объект.

Если модель заменяет объект с достаточной точностью, то она адекватна. Адекватность инженерных моделей зависит от инжен-го опыта, интуиции и подготовки специалистов.

*Основная задача модел-ия* – построение таких моделей, которые позвол. выполнять исследования на них, а не на реальных объектов.

**2.** Задание матриц и векторов, действия с квадратными матрицами, извлечение из матрицы строк и столбцов, разворачивание матрицы в вектор, создание матрицы значений функции

Панель операций с матрицами и векторами в **Matrix** открывается щелчком по кнопке  в панели математических инструментов. За кнопками панели закреплены следующие функции:

 – Задание размеров матрицы;

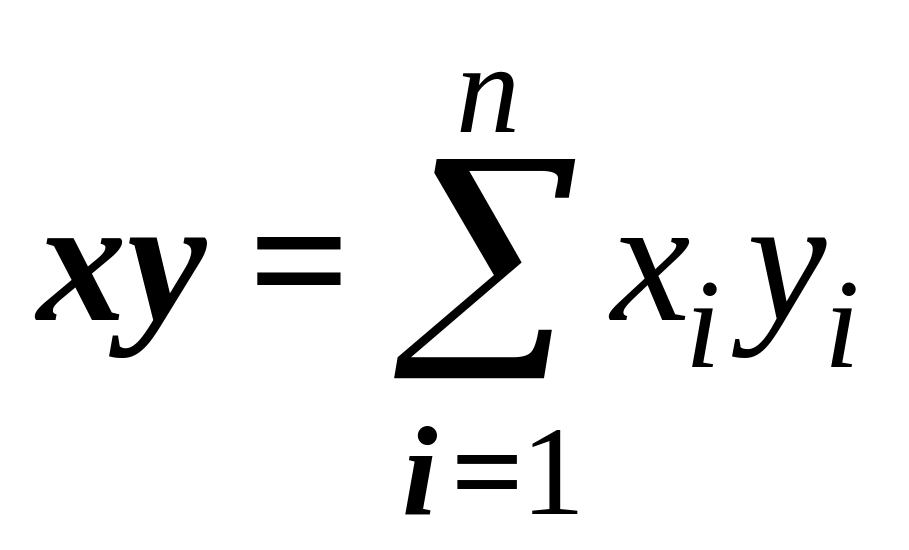
 – ввод нижнего индекса;

 – вычисление обратной матрицы;

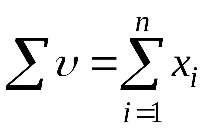
 – вычисление определителя матрицы:  ;

 – выделение столбца матрицы: *М*<*j*> – *j*-й столбец матрицы;

 – транспонирование матрицы: *М*={*mij*}, *MT*={*mji*},

 – вычисление скалярного произведения векторов:  ;

 – вычисление векторного произведения двух векторов: ***a***×***b***=(*a*2*b*2 – *a*3*b*2 –*a*2*b*1 –*a*1*b*2 –*a*2*b*1);

 – вычисление суммы компонент вектора:  ;

Номер первой строки (столбца) матрицы или первой компоненты вектора, хранится в Mathcad в переменной ORIGIN. ORIGIN:=1.

В MathCAD термин "origin" относится к началу координат в графической области программы. Он указывает на точку (0, 0) на координатной плоскости, где оси X и Y пересекаются.

Когда вы строите график функции или создаете диаграмму в MathCAD, позиция origin определяет, где будет находиться начало осей X и Y на графике. Вы можете перемещать origin и изменять его позицию в программе, чтобы лучше соответствовать вашим потребностям при создании графика.

Номер первой строки (столбца) матрицы или первой компоненты вектора, хранится в Mathcad в переменной **ORIGIN**.  
По умолчанию в Mathcad координаты векторов, столбцы и строки матрицы нумеруются начиная с **0** (**ORIGIN:=0**). Поскольку в математической записи чаще используется нумерация с **1,** удобно перед началом работы с матрицами  определять значение переменной **ORIGIN**равным **1**, выполнять команду  
**ORIGIN:=1.**

Когда переменной "origin" присваивается значение 0, начало координат будет находиться в левом нижнем углу графической плоскости в MathCAD. Это означает, что ось X будет направлена вправо, а ось Y - вверх.

Когда переменной "origin" присваивается значение 1, начало координат будет находиться в левом верхнем углу графической плоскости в MathCAD. Это означает, что ось X будет направлена вправо, а ось Y - вниз.

В MathCAD можно выполнять различные действия с квадратными матрицами. Некоторые из них включают:

1. Создание матрицы: Можно создать матрицу, используя квадратные скобки [] и разделять элементы запятыми или пробелами. Например, [1, 2, 3; 4, 5, 6; 7, 8, 9] создаст матрицу 3x3.

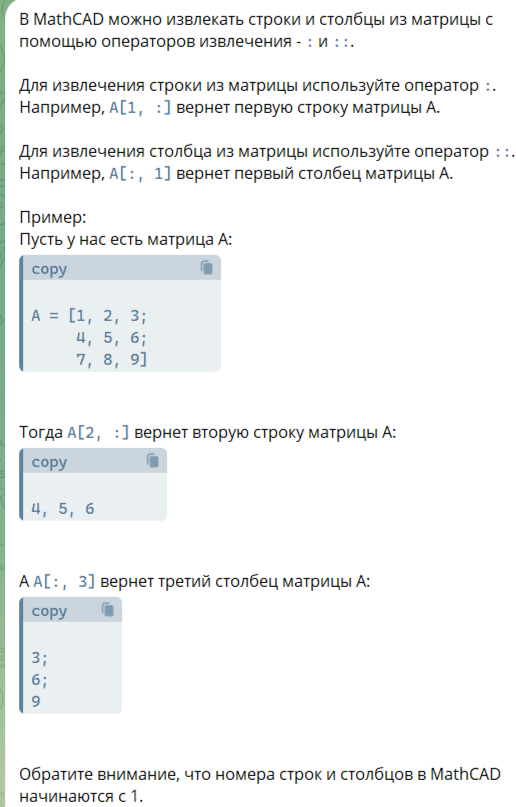
2. Сложение и вычитание матриц: Можно складывать или вычитать матрицы одинакового размера с помощью операторов + и -. Например, A + B сложит матрицы A и B элемент-wise.

3. Умножение матриц: Можно умножать матрицы с помощью оператора \*. В MathCAD для выполнения матричного перемножения необходимо использовать оператор . . Например, A \* B выполнит матричное умножение матриц A и B, и результат будет матрицей.

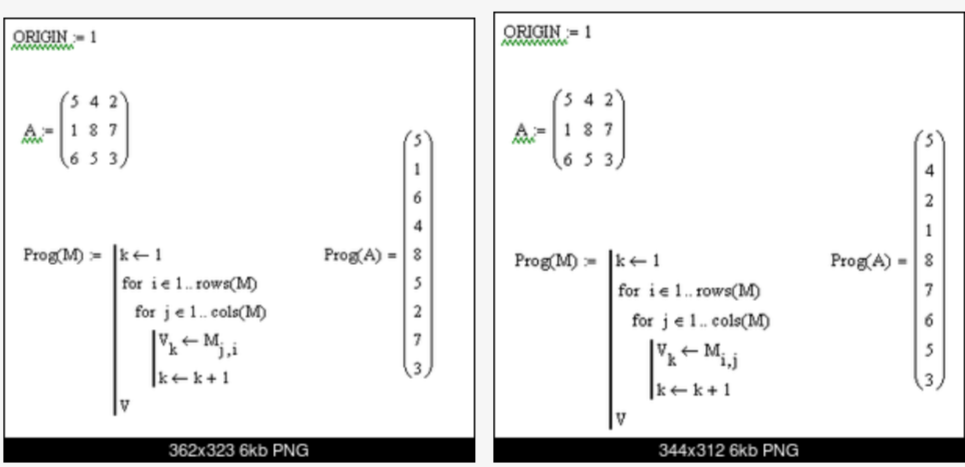
4. Транспонирование матрицы: Для транспонирования матрицы в MathCAD можно использовать функцию Transpose(). Например, Transpose(A) вернет транспонированную матрицу A.

5. Обратная матрица: Можно вычислить обратную матрицу с помощью функции Inverse(). Например, Inverse(A) вернет обратную матрицу матрицы A, если она существует.

6. Вычисление определителя: Для вычисления определителя матрицы в MathCAD можно использовать функцию Determinant(). Например, Determinant(A) вернет значение определителя матрицы A.



Разворачивание матрицы в вектор



**3.Виды моделирования. Математическое моделирование.**

1. ***Натурное****.* При таком моделировании объект изготавливается в натуральную величину из тех же материалов, которые будут использоваться в реальной эксплуатации и к которому прикладывают воздействие, как можно ближе воспроизводящее реальную среду.

*преимущества:* точные результаты при точности измерительной аппаратуры;

*недостатки*: высокая стоимость проведения эксперимента,часто испытуемый объект приходит в негодность после испытания, невозможность применения его в нек. Случаях.

1. ***Полунатурное.*** Объект изготавливают либо в измененном масштабе, либо из другого материала, либо еще с каким-то существенным отличием от реального.

*преимущества:* достаточно наглядное представление результатов;

*недостатки:* сложность пересчета полученных результатов к реальным.

1. ***Аналоговое*** базир-ся на философском принципе единства мира, который в частности выражается в том, что явления и процессы разной природы могут описываться одинаковыми по виду уравнениями.

*Преимущества:* решение получается мгновенно;

*Недостатки:* сложность пересчета полученного решения к искомым переменным.

1. **Математическое моделирование:** в этом случае в качестве искусственного объекта заменяющего реальный выступает математический объект. Этот объект может иметь формулу ур-ий, матриц, графов и т.д. Последовательность этапов переходов к компьютерной модели.

*Преимущества*: чаще всего моделирование происходит на информационных машинах, невысокая стоимость, универсальность.

**5. Численные методы как средство математического моделирования. Основные этапы и источники ошибок.**

Численные методы являются мощным инструментом математического моделирования, позволяющим решать сложные математические модели на компьютере. Эти методы используются для решения самых разных типов задач, включая дифференциальные уравнения, интегральные уравнения и задачи оптимизации. Они являются важным инструментом во многих областях, включая физику, инженерию, финансы и многие другие, позволяющим решать сложные задачи, не поддающиеся аналитическому решению. С ростом доступных вычислительных мощностей численные методы становятся все более популярными и важными для решения сложных математических моделей.

Этапы:

*Реальный объект(1) - Представление исследователя об объекте(2) - Упрощающее предположение(3) - Построение расчетной схемы (4) - Выбор уравнений для описания(5) - Выбор методов решения уравнений(6) - Выбор программ для реализации(7)- Готовая комп. модель(8)*

*Источники ошибок:*

1-2 неправильное представление исследователя,

2-3 необоснованное упрощающее предположение,

3-4 неправильное отображение взаим-ия объекта с внешней средой,

4-5 неправильный выбор уравнений для описания,

5-6 некот. уравнения требуют особые методы решения,

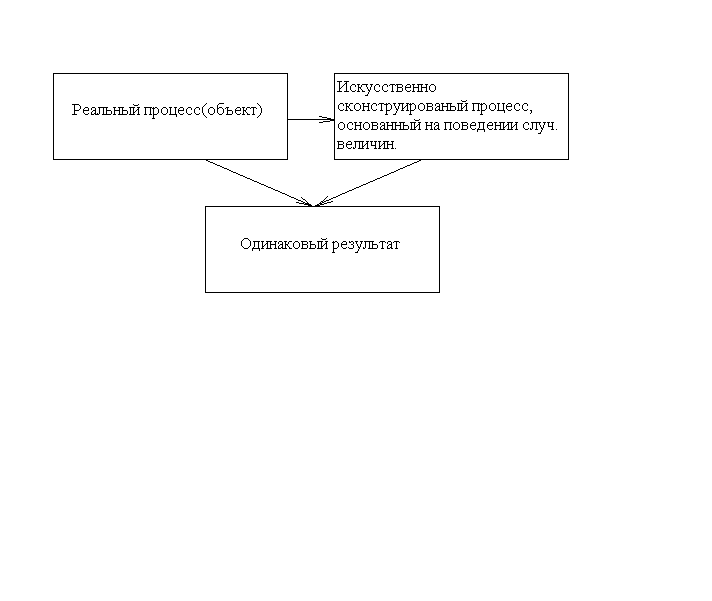
6-7 нужно правильно выбрать программную среду,

7-8 могут возникать семантические ошибки

**6. Понятие о методах типа Монте-Карло**.

В 1943г. в исследовательских лабораториях Лос-Анджелесе при разработки ядерной бомбы возникла задача об определении глубины проникновения электронов в заданное вещество. Решить её не удалось. Тогда Станислав Улом и Ждон фон Нейманом предложили подход основную стратегию, которую используют игроки при игре в кости. Эта стратегия была стратегия была основана на поведении случайных величин. По имени города- это метод получил название Монте – Карло. В дальнейшим по традиции многие методы стали называться метод Монте – Карло.

Изобразим схему метода.



Заметим, что этот подход может использоваться, как для моделирования явлений имеющих в своей основе поведение случайных величин (такие величины называются - стохастическими) так и для явлений процессов, где случайные величины не присутствуют (детерминированные).

**7. Тестирование**

Не всегда после создания модели мы можем быть уверены в том, что она ведёт себя так-же, как и реальный объект, поэтому нам нужно производить её тестирование.

Под тестированием понимают выбор таких параметров модели, при которых исследователь может предсказать поведение модели**.**

Тестирование : 1) сравнение рез-ов расчета с рез. Натурного экспер-та

1. Сравнение рез-ов от разных программ
2. Расчет на основе такой схемы, когда мы можем предсказать результ

**8. Отделение корня уравнения и уточнение корня уравнения. Метод бисекции (деления отрезка пополам).**

Процесс численного решения уравнения разбивается на два этапа: **отделение корней и уточнение корней.**

Пусть дано уравнение вида (2.1)https://studme.org/htm/img/15/3592/58.png

Говорят, что *корень Е* ***отделен на данном промежутке****,* если он содержится в этом промежутке и других корней на том же промежутке нет.

***Произвести полное отделение всех корней уравнения*** — значит разбить всю область допустимых значений на промежутки, в каждом из которых содержится только один корень или не содержится ни одного корня.

**Метод половинного деления (метод бисекции)**

Считаем, что отделение корней произведено и на интервале [a,b] расположен один корень, который необходимо уточнить с погрешностью ε.   
Итак, имеем f(a)f(b)<0. Метод дихотомии заключается в следующем. Определяем половину отрезка c=½(a+b) и вычисляем f(c). Проверяем следующие условия   
1. Если |f(c)| < ε, то c – корень. Здесь ε - заданная точность.   
2. Если f(c)f(a)<0, то корень лежит в интервале [a,c].   
3. Если f(c)f(b)<0, то корень лежит на отрезке[c,b].

Продолжая процесс половинного деления в выбранных подинтервалов, можно дойти до сколь угодно малого отрезка, содержащего корень ξ.   
Так как за каждую итерацию интервал, где расположен корень уменьшается в два раза, то через n итераций интервал будет равен: https://www.semestr.ru/images/math/optim/o1_image015.gif

**9. Сравнительные характеристики пакетов MATHCAD, MATLAB, MAPLE, MATHEMATICA. Основные приемы работы в MATHCAD.**

**MATHCAD** – программа появилась недавно.

Преимущества:

1) Действия в соответствующем документе записываются почти также как у инженера в тетради.

2) Также имеется графический интерфейс, ориентированный на пользователя.

3) Для быстрого входа надо 2 сеанса.

4) Навыки легко восстанавливаются после некоторого времени не работы с программой.

Недостатки**:**

1) Неудобная отладка.

2) Невозможность создавать файлы, независящие от среды (.exe).

3) Тексты встроенных процедур закрыты.

4) Недостаточно развитый интерфейс обмена данными с др. программами.

5) В среду не включены все процедуры символических вычислений.

**MATLAB** – давно появившаяся система, есть встроенный язык, на котором в текст. документе можно исследовать задачу. Каждая переменная явл. матрицей (в этом языке)

Преимущества:

1) Имеются TOOLBOX каждый из которых ориентир. на опр. область в расчетах или моделирование. Они сгруппированы в отдельные разделы, при установке вы можете пометить, которые вам нужны: решение дифуров, решение дифуров в частн. произв., интерполяция, оптимизация, нейронные сети, статистика, FEMLAB – работа с конечно-элементными моделями, работа с изображениями.

2) Открытый исходный код.

3) Можно использовать библиотеки из Fortran.

4) Можно создать независимый .exe файл.

5) Хорошо развит интерфейс обмена данными с внеш. программами.

**MAPLE** – законодатель в области символических вычислений.

Часть ядра куплена у MATHCAD. Ядро встроено в один из TOOLBOX MATLAB.

Используется для сложных математич. вычислений. Можно работать с графиками.

**10. Решение систем линейных алгебраических уравнений методом Гаусса.**

Цель элементарных преобразований в методе Гаусса - привести матрицу к ступенчатому виду.

Алгоритм:

1) Записать расширенную матрицу системы. Матрица системы – это матрица, составленная только из коэфф. при неизвестных, а расширенная матрица – такая же матрица + столбец свободных членов.

2) Выполнить некоторые действия, которые называют элементарными преобразованиями:

- строки можно переставлять местами;

- все пропорциональные строки следует удалять, оставив только одну;

- нулевую строку тоже следует удалять;

- строку матрицы можно умножить (разделить) на любое число, кроме нуля.

- к строке матрицы можно прибавить другую строку, умноженную на любое число, кроме нуля.

http://mathprofi.ru/g/metod_gaussa_dlya_chainikov_clip_image040.gif

(1) Ко второй строке прибавили первую строку, умноженную на –2.

(2) Делим вторую строку на 3.

**Цель элементарных преобразований** – привести матрицу к ступенчатому виду:

В результате элементарных преобразований получена эквивалентная исходной система уравнений:  
http://mathprofi.ru/g/metod_gaussa_dlya_chainikov_clip_image044.gif

**11. Алгоритм «прогонка» для решения системы линейных уравнений с диагональным преобладанием.**

Метод прогонки является частным случаем метода Гаусса и используется для решения систем линейных уравнений вида Ax = B, где A – трёхдиагональная матрица. Трёхдиагональной матрицей называется матрица такого вида, где во всех остальных местах, кроме главной диагонали и двух соседней с ней, стоят нули.

Метод прогонки состоит из 2 этапов: прямой и обратной прогонки.

На первом этапе определяются прогоночные коэффициенты.

На втором находят неизвестные x.

*d1x1+b1 x2=C1*

*a2 x1+ d2x2 +b2 x3=C2*

*a3 x2+ d3x3 +b3 x4=C3*

*an-1 ∙xn-2+dn-1∙xn-1 + bn-1∙xn =Cn-1*

*an xn-1+ dn∙xn  =Cn*

A= M= C=

C=A∙M **Mi=pi∙Mi+1+qi** (1)

d1M1+ b1M2 =

**M1**= + **q1** = **p1***=*

M1=p1∙M2+q1

a2∙M1+ d2∙M2+ b2∙x3=

a2∙( p1∙M2+q1)+ d2∙M2+ b2∙m3=

a2∙p1∙M2+ a2q1 + d2∙M2+ b2∙m3=

M2( a2∙p1+ d2 )+ a2q1 + b2∙m3=

**11**

M2 =

*Mi = ∙* Mi+1*+*, pi= qi=

*Mn=*

А затем проходим с конца и находим Mi по формуле 1

**13. Сведение системы ОДУ произвольного порядка к системе ОДУ первого порядка в нормализованной форме Коши.**

m\*X''+C\*X'+r\*X=P(t)

x(t0)=x0

x'(t0)=x'0

Введем новую систему функций: z1=x z2=x'=z'

Старшую производную уединяем в правой части X''=(P(t)-C\*X'-r\*X)/m

получаем систему:

z1'= X'=z2

z2'=(P(t)-C\*X'-r\*X)/m

z2'=(P(t)-C\* z2-r\* z1)/m

Начальные условия z1(t0)=x0 x2(t0)=x'0

Если есть система ОДУ из r уравнений порядка n1 n2 n3… nr, то ее можно свести к системе нормализованной форме Коши, кот. будет содержать n1+n2+n3+…nr уравнений 1-го порядка.

**14, 16, 18. Различные виды приближений данных.**

В различных областях при проведении эксперимента инженер получает таблицу данных, эти дискретные данные необходимо обработать таким образом, чтобы можно было восстановить незамеренные значения в промежуточных точках или может быть подобрать зависимость, которая приближенно воспроизводит закон, позволяя вместо дискретных данных использовать непрерывные зависимости.

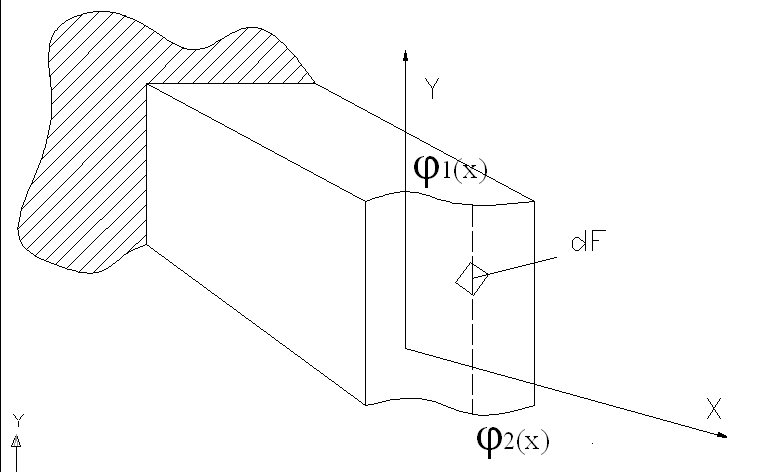


Рисунок 1.

Для того, чтобы рассчитать прочность и жесткость, необходимо знать геометрические характеристики: площадь, моменты инерции .



В этом случае необходимо знать зависимости и .

В реальности можно выполнить замеры на кромках. Таким образом, получим таблицу данных:

|  |  |
| --- | --- |
| X | X1 X2 X3 Xn |
| Y | Y1 Y2 Y3 … Yn |

Возникает вопрос: как по этой таблице восстановить значения ?

Для этого мы должны найти функцию F(x) ≈

-аппроксимируюмая ф-я. F(x) – аппроксиматор.

**Аппроксимация** - приближение или научный метод, состоящий в замене одних объектов другими, в каком-то смысле близкими к исходным, но более простыми.

Существуют различные способы приближения. Одним из таких методов является интерполирование. **Интерполяция** – один из видов аппроксимации, когда одна функция замещается другой функцией, совпадающей с ней в одних точках, а в других точках лишь приближающаяся к ней.Здесь требуют, чтобы приближающая функция совпадала с приближаемыми значениями, то есть если обозначить  как приближающую функцию, а приближающие значения , то получим

, .

Эти точки называются ее узлами. Чаще всего полученная таблица содержит не точные значения, а замеры, полученные с некоторой погрешностью. Поэтому требовать

**14,16,18**

совпадения с табличными значениями приближенной функции неуместно. Имеет смысл использовать другие критерии близости.

# 

# Поточечное среднеквадратическое приближение.



Рисунок 1.

Иногда возникает задача о приближении не табличных данных, а функции других функций. Пусть приближаемая функция  (является известной функцией), а приближающая (подбираемая функция).

Аналогом приближения, рассматриваемого на рисунке 1, в этом случае будет непрерывное приближение в среднеквадратическом смысле.

# Непрерывное приближение в среднеквадратичном смысле.



Рисунок 1.

**14,16,18**



Такое приближение может использоваться в том случае, если исходная функция очень сложно или долго вычисляется, при этом ее заменяют другой функцией, которая известна.

# Равномерное приближение.



Понятно, что равномерное распределение выдвигает самое жесткое требование

Иногда ставится очень жесткое условие: во всех точках некоторого отрезка [a,b] отклонение многочлена y(x) по абсолютной величине должно быть меньше заданной величины https://studfile.net/html/2706/1116/html_nbFHPa85lt.SKvF/img-nSvit_.png: https://studfile.net/html/2706/1116/html_nbFHPa85lt.SKvF/img-XFWJB3.png, https://studfile.net/html/2706/1116/html_nbFHPa85lt.SKvF/img-8fuixP.png.

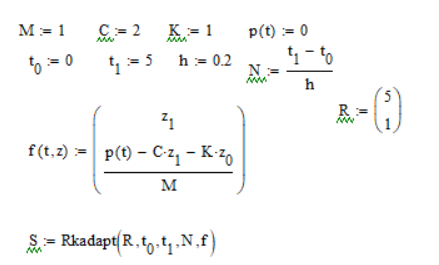
При выполнении такого условия говорят, что многочлен https://studfile.net/html/2706/1116/html_nbFHPa85lt.SKvF/img-dlXdY3.png равномерно аппроксимирует функцию https://studfile.net/html/2706/1116/html_nbFHPa85lt.SKvF/img-YtCEFk.png с точностью https://studfile.net/html/2706/1116/html_nbFHPa85lt.SKvF/img-fuMLAA.png на отрезке https://studfile.net/html/2706/1116/html_nbFHPa85lt.SKvF/img-uG2Fzy.png.

https://studfile.net/html/2706/1116/html_nbFHPa85lt.SKvF/img-mgpEjv.png.

**15. Реализации в среде MathCad математической модели для простейшего осциллятора.**

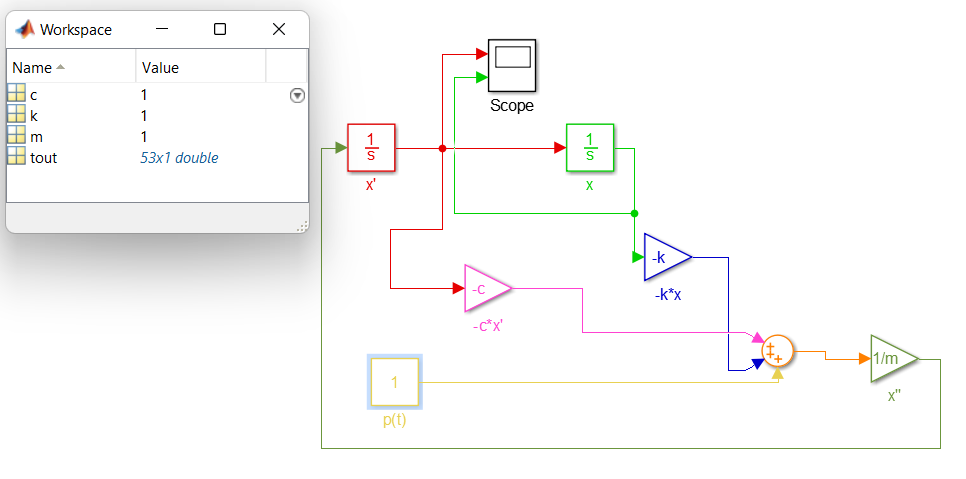
Для того, чтобы создать осциллятор нам нужно свети уравнение к нормализованной системе Коши 1-го порядка. Для решения задачи Коши имеется процедура Rkadapt(R,t0,t1,N,f) – процедура с адаптивным шагом( шаг меняется автоматически)

R- начальные условия, t0,t1 – начало и конец интервала, на котором будем находить решения. N- количество шагов. f-вектор с правой частью нормализованного вида Коши.



Так же для решения можно использовать rkfixed, в нём шаг постоянный, а не адаптивный.

**17. Пример использования Simulink MATLAB для решения ОДУ 2-го порядка.**



X''M=-X'C-XK+p(t)

**19.. Получение модели колебательной системы со сосредоточенными параметрами на основе уравнений Лагранжа 2-го порядка..**

Во многих случаях при моделировании механических систем приемлемы предположения о том, что масса системы сосредоточена лишь в конечном числе точек, соединенных между собой элементами типа пружин (элементы, накапливающие потенциальную энергию) и типа "демпфер" (элемент, рассеивающий энергию), в этом случае математическая модель, описывающая повеление рассматриваемой системы представляет собой систему обыкновенных дифференциальных уравнений (ОДУ). Рассмотрим следующий пример.

x1(t) и x2(t) – смещение относительно положения равновесия соответственно 1-й и 2-й массы; C1 и C2 - жесткости пружин

По гладкой плоскости без трения под действием внешней силы, изменяющейся во

x2(t)

x1(t)

C2

C1

M2

M1

времени по закону P(t), движутся два груза (рис.1).

Для построения математической модели следует воспользоваться уравнениями Лагранжа 2-го рода, которые приводят к системе двух обыкновенных дифференциальных уравнений второго порядка [1].

Здесь x1(t) и x2(t) - смещения относительно положения равновесия соответственно 1-й и 2-й массы;

**19**

- скорости смешений относительно положения равновесия соответственно 1-й и 2-й массы, где t - время;

Т - кинетическая энергия системы; П - потенциальная энергия системы.

В качестве 1 - й и 2- й степеней свободы примем x1 и x2.

Тогда уравнения примут вид:

Выполнение этого этапа базируется на использовании знаний курса теоретической механики. Подробности можно найти в книгах [1,2]. Результатом выполнения этого этапа являются уравнения. движения механической системы. Их подробный вывод следует привести в отчете.

Системы с рассредоточенными параметрами- в этой ситуации неизвестные величины(напр. Перемещение точек) являются уже функ-ми нескольких переменных. Поведение таких систем чаще всего описывается ДУЧП. Виды: 1)Метод граничных элементов 2)Метод конечных элементов 3) Метод сеток.

**20, 22. Разложение аппроксиматора по системе базисных функций.**

Возникает естественный вопрос, за счет чего же можно изменить значения критерия при аппроксимации. Очевидно,  должна зависеть от параметров, варьируя которыми мы и будем менять ее вид.

Если представить график функции  в виде проволоки, то изменяя эти параметры, мы будем по-разному изгибать эту проволоку.

Одним из естественных предположений для выбора функции является следующее:



- подбираемые варьируемые константы;

 - набор неизменяющихся функций, называемых базисными.

Пусть мы имеем  , тогда требование близости в среднеквадратическом смысле примет вид:



где



Найдем это выражение:





Получена система линейных уравнений:

**20, 22**

где - симметричная матрица.

 (1)

Уравнение (1) представляет собой функцию-аппроксиматор.

Необходимо определить  на интервале , а в качестве критерия близости выбираем:

 (2)



 (3)

Введем обозначения:

 (\*\*)

 (4)



 (5)



Если сравнить (\*\*) с формулой (\*) (смотрите ранее), то заметим, что суммирование по точкам заменено интегрированием по отрезку. Понятно, что в программной реализации в лабораторной работе по точечному среднеквадратичному приближению достаточно заменить (\*) на формулу (\*\*).

**21.** **Пример получения уравнений движения для двухмассовой системы с поступательным движением масс**

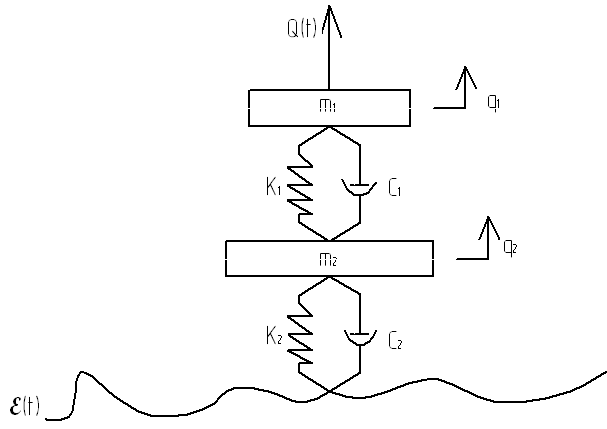


Рисунок 2.9. Колебательная двухмассовая система

Получим уравнения движения для двухмассовой системы. Движение происходит по дороге со скоростью .

закон дорожного возмущения (кинематическое возмущение).

m1 , m2 – массы;

k1 , k2 - жесткости упругих элементов;

с1 , с2 - характеристики демпфирующих элементов.

Введены обобщенные перемещения

- смещение верхней массы относительно дороги.

- смещение нижней массы относительно дороги.

**21**

При этом будем считать, что эти смещения соответствуют недеформированному состоянию пружины. Очевидно, что введённых обобщенных степеней достаточно для однозначного определения положения системы в любой момент времени. При этом ни одно из них не выражается через другое.

Составим выражения для абсолютных перемещений:

Запишем уравнения для удлинений пружин:

Запишем скорости удлинений:

Записываем выражение для кинетической энергии:

Записываем выражение для диссипативной функции:

Выражение для потенциальной энергии:

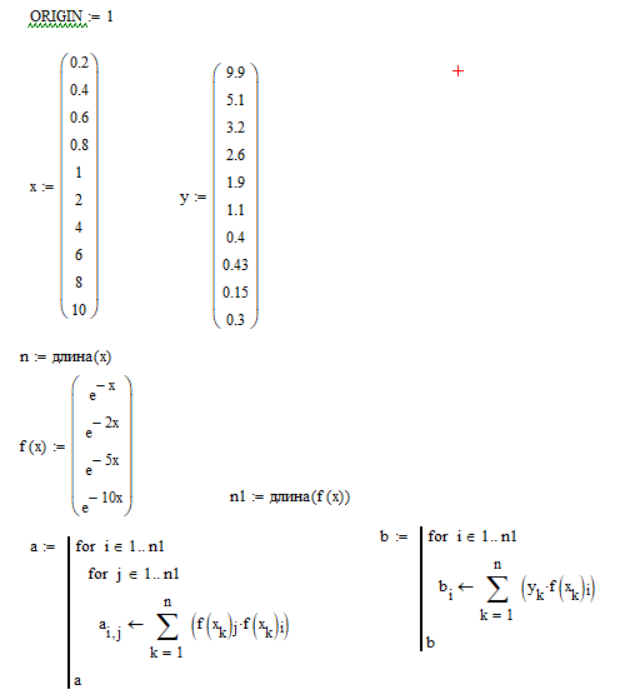
Выражаем :

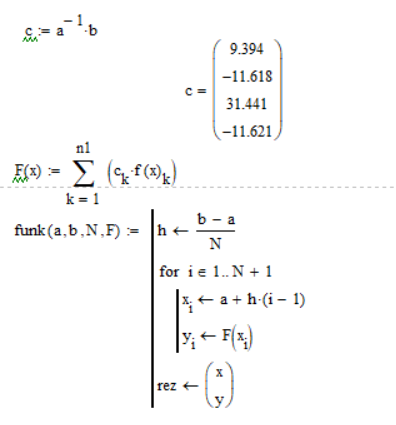
**21**

Вычислим нужные частные производные:

Окончательно система уравнений имеет вид:

**26. Пример реализации MATHCAD метода аппроксимации по МНК.**

****

**26.**

**23.** **Примеры задач, приводящих к обыкновенным дифференциальным уравнениям.**

1)Допустим, что в каждый момент времени *t*известна скорость *f*(*t*) точки, движущейся по оси *Ox*, где *f*(*t*) – функция, непрерывная на *(a,b).* Кроме того, будем считать, что известна абсцисса *x*0 этой точки в некоторый определённый момент времени *t*= *t*0 . Требуется найти закон движения точки, то есть зависимость абсциссы движущейся точки от времени.

2) В некотором экологически замкнутом районе живут два вида животных (например, рыси и зайцы). Зайцы (жертвы) питаются растительной пищей, имеющейся всегда в достаточном количестве (в рамках данной модели не учитывается ограниченность ресурсов растительной пищи). Рыси (хищники) могут питаться только зайцами. Необходимо определить, как будет меняться численность жертв и хищников с течением времени в такой экологической системе. Если популяция жертв увеличивается, вероятность встреч хищников с жертвами возрастает, и, соответственно, после некоторой временной задержки, растет популяция хищников.

**24. Интегральная аппроксимация на отрезке по МНК**

ci – неизвестный коэффициент

) – базисная функция

Если выполнить вычисления, то для определения можно получить систему линейных уравнений

с0()+ с1()+…+ сm()=b0

с0()+ с1()+…+ сm()=b1

…….

с0()+ с1()+…+ сm()=bn

ai,j=() =

bi=

A=

C= B=

Решив данную систему и найдя C можно обратиться к аппроксиматору.

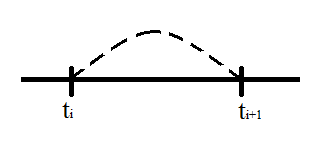
**25. Методы решения задачи Коши для ОДУ.**

**Смотри 29,31,33**

**27, 12. Классификация методов решения задачи Коши**

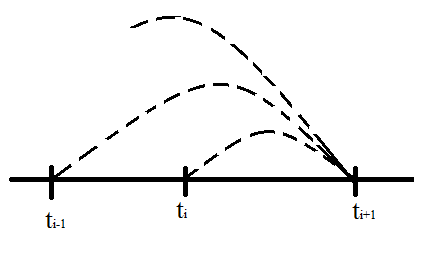
Рассмотрим наши методы относительно к так называемым шаговым (пошаговым) методам интегрирования ОДУ.

Шаг интегрирования может быть постоянным или переменным. Мы рассмотрим такие методы, где для получения решения в очередной момент времени нам требовалось знать решение только в предыдущий момент времени. Такие методы называются одношаговые.



Существуют методы, в которых для построения решения в данный момент времени нужно знать решение задачи в нескольких предшествующих моментах.

Многошаговый метод:

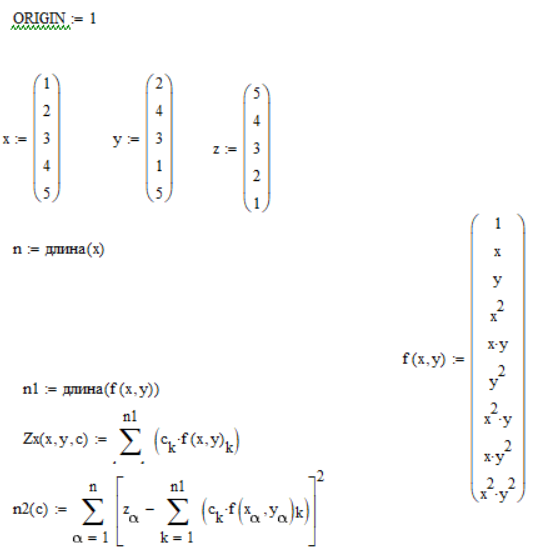


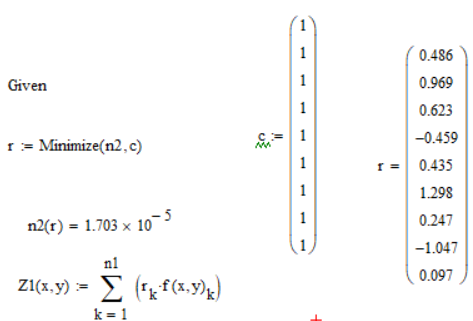
Многошаговый метод не может начать работу по известному лишь в одной точке начальному условию. То есть для его разгона требуется предварительно запустить какой-то одношаговый метод и с его помощью подготовить решение в нескольких точках. Эти методы еще называются несамостартующими или несаморазгоняющими.

В рассмотренных нами методах значение неизвестного решения выражалось в явном виде через решение в предшествующих точках. Поэтому такие методы называются явными.

Существуют и другие методы, где yi+1 – неизвестное решение – является решением какого-то уравнения, чаще всего нелинейного, связанного с видом правой части ОДУ, т.е. неизвестное решение неявно входит в какую-то зависимость. Такие методы называются неявными. В них дополнительно дополнительно придется решать дополнительное уравнение, в общем случае трансцендентное. Обычно такие уравнения решаются методом Ньютона. Такие методы имеют повышенную точность и соответственно позволяют значительно увеличить шаг по независимой переменной. Методы, в которых это свойство не используется называются явными.

**28.** **Пример аппроксимации функции 2-х переменных в MATHCAD, значения которой заданы таблично.**

****

**28**

**29. Метод Эйлера и его модификации для решения ОДУ**

Если заменить площадь криволинейной трапеции, представляющей собой величину этого интергала, например, на площадь прямоугольника

*-* Ф-ла Эйлера реализующая метод Эйлера. Можно показать, что локальная погрешность этой ф-лы , т.е. метод Эйлера является методом 1-го порядка точности.

y0

X

y=y(x)

Y

X0

X1

α0

h

Δy

y(x1)

**Модификации метода Эйлера**

h

α0

α1

y0

x0

x1

x

y

y0

Можно показать, что погрешность этой ф-лы – – метод 2-го порядка. Точность увеличивается на порядок, но приходится ещё раз обращаться к правой части ДУ

**Усовершенствованный**

Находим производную в точке x(tg a1), производную в точке x+h(tg a2). Находим новый тангенс угла наклона tg a\* = (tg a1 + tg a2)/2 и находим yh=

O(h3)

**30**. **Алгебраическое интерполирование.**

Алгебраическое интерполирование – это вид приближения, если приближенную функцию обозначить , то для таблицы из n точек ,  необходимо потребовать выполнения условия:





Понятно, что в такой общей постановке решением задачи может оказаться бесконечное решение функций, поэтому, чтобы сделать задачу определенной, необходимо сузить класс подбираемых функций. Как было рассмотрено выше, при наличии определенной информации следует подбирать приближенную функцию так, чтобы она согласовывалась с приближаемыми данными.

Если же такой информации нет, то одним из способов выбора является выбор полинома в качестве приближаемой функции. В этом случае интерполяцию называют ***алгебраической***.

Будем искать интерполянт в виде полинома степени не выше(n-1).

 (1)

Тогда условие (1) примет вид:

 (2)

Понятно, что вид интерполирующего полинома полностью определяется набором коэффициента.

Рассмотрим следующее утверждение:

Пусть в n попарно-различных точках  заданы числа, тогда существует единственный полином  степени не выше (n-1), удовлетворяющий условию (\*).

Доказательство:

Запишем это условие в каждой точке



Здесь  - неизвестные, а правые части известны:

**30**

 (4)

Эта матрица специального вида - матрица Вандерманда. Для нее легко показать, что определитель этой матрицы  отличен от нуля, но если это так, то система (4) имеет единственное решение.

Следствием этого утверждения является то, что вид интерполирующего полинома не зависит от способа его построения. А сам ход рассмотренных рассуждений дает один из способов построения такого полинома.

Рассмотренный способ называется классическим, т.к. в силу единственности интерполяционного полинома, каким бы способом его не строили, результат будет один и тот же.

**31.** **Основная идея методов Рунге-Кутта для решения ОДУ**

Интервал интеграла [x, x+h] t ϵ [0;1] t=x+αh

x

t

x

x+h

0

t

1

введем 3 набора параметров:

α1 α2 α3… αq

β10

β20,β21

⁞

βq0,...βq,q-1

A0, A1, …, Aq

ф0=hf(x,y)

ф1=hf(x+ α1h;y+ β10 ф0)

ф2=hf(x+ α2;y+ β20 ф0+ β21 ф1)

⁞

фq=hf(x+ αqh;y++ βq0 ф0+ βq1 ф1+ βq,q-1 фq-1)

Каждая из ф вычисляется на основе предыдущих ф.

Для оценки точности используется величина O(hm+1), где h – шаг, m – порядок метода.

Например, метод Эйлера 1 порядка, поэтому его точность O(h2), а у его модификации m=2, поэтому O(h3)

**32. Интерполяционная ф-ла Лагранжа** – один из наиболее распространенных способов построения интерполяционного полинома.

Введем предварительно в рассмотрение так называемые ***полиномы влияния .***

Этот полином должен удовлетворять следующим условиям:

1. степень полинома должна быть равна (n-1);
2. 



Очевидно, что полином степени (n-1) , равный нулю во всех узлах кроме i- того, имеет вид:



Остается определить константы С из условия 

Полином Лагранжа обычно обозначают .

Очевидно, что



Пример. Пусть имеется таблица из n=3 точек

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |



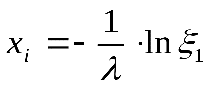
**33**. **Метод Рунге-Кутта 4-го порядка точности для решения ОДУ(Метод ) и его реализация в MathCad.**



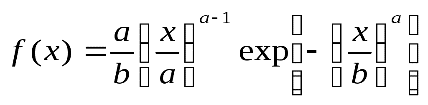
**34 Примеры формул для моделирования случайных величин с разными законами распределения.**

Экспоненциальный

Плтоность:https://studfile.net/html/2706/242/html_tcMsCHxiaN.KsKP/img-per3K7.png

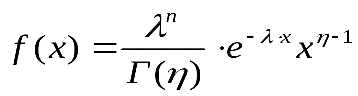
Формула для моделирования случайной величины

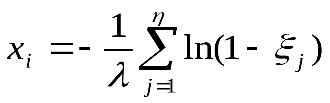
Вейбула

Плотность распределения

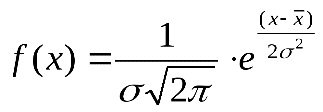
Формула для моделирования случайной величиныhttps://studfile.net/html/2706/242/html_tcMsCHxiaN.KsKP/img-tlxqOw.png

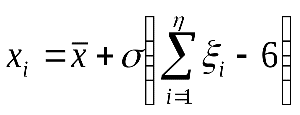
Гамма-распределение (https://studfile.net/html/2706/242/html_tcMsCHxiaN.KsKP/img-bfAj8F.png — целые числа)

Плотность распределения

Формула для моделирования случайной величины

Нормальное

Плотность распределения

Формула для моделирования случайной величины

**35,37 Способ Рунге для оценки погрешности ОДУ.**

При численном решении любой задачи возникает 2 вопроса:

1. С какой точностью необходимо решить задачу?
2. Какую цену

Чем большей точности требуется достигнуть, тем большее количество времени придется потратить на вычисление. Сущ. Определенные задачи, для которых приемлимая точность не может быть получена некоторыми классами методов.

Приемлимо к ДУ, проверить точность полученного решения можно с использованием некоторых способов:

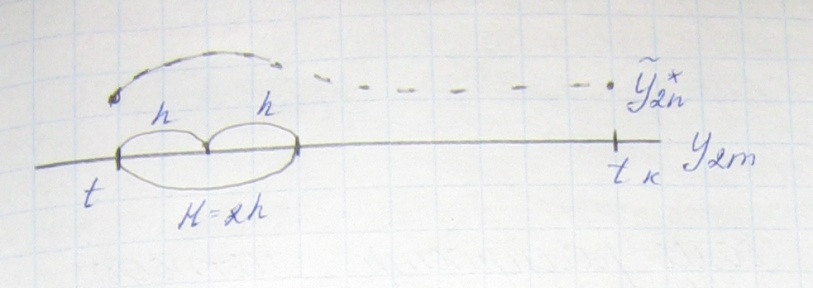
1. Двойной пересчет с помощью другого метода или более мелкого шага интегрирования.
2. Применение более грубой расчетной схемы, для которой можно качественно оценить решение.

**Способ Рунге для оценки такой погрешности.**

Предполагает, что на зад. интервале решение выполняется с постоянным по величине шагом, а затем решение выполняется с удвоенным по величине шагом H=2h. Пусть при этом используется метод m-го порядка точности. Тогда на отдельном шаге h величина локальной погрешности составит: .

Предполагаем, что c шагом h выполняется 2n шагов, тогда Н→n.

Предположение на котором основан метод Рунге заключается в том, что на отдельном шаге погрешность равна: 



- приближенное решение в конечной точке tk после 2n шагов величины h.

- приближенное решение в точке tk после n шагов величиной H=2h.

*Y*2n – неизвестное решение.

Считая на каждом шаге погрешность одинаковой, можно записать:



При вычислении с удвоенным шагом:



Система двух уравнений с двумя неизвестными A и *Y*2n. Решая, получим:



**35,37**

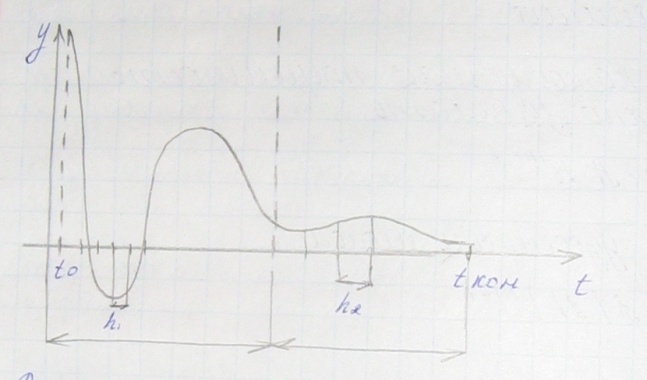




В частности, для более распространенного метода Рунге-Кутта 4-го порядка (m=4):



На этом подходе основаны алгоритмы с адаптацией шага:



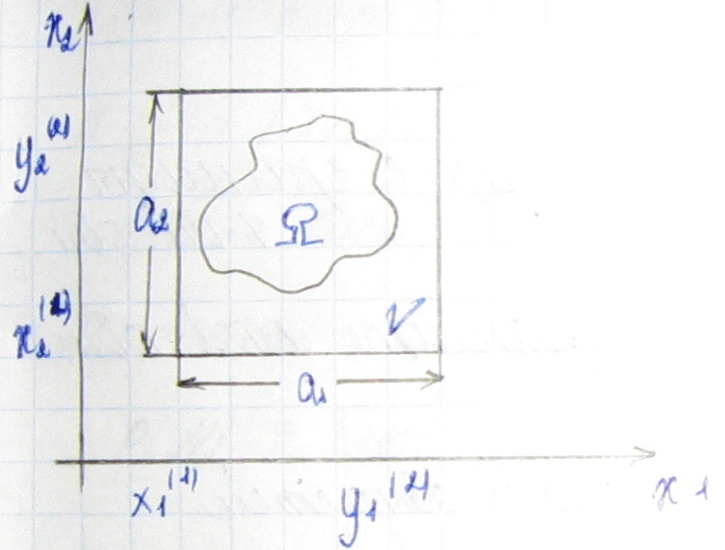
Решение меняется быстро, а область медленно. Для того, чтобы с достаточной точностью отследить быстрое изменение решения на 1 участке потребуется достаточно малый шаг. При выходе на 2 участок такой малый шаг уже не нужен, т.к. решение меняется гораздо медленнее. Управление величиной шага можно поручить программе, которая должна по истечению нескольких шагов отслеживать точность решения и если погрешность достаточно мала, увеличивать шаг(например вдвое), если велика дробить шаг.

Критерием будет:



**36 Использование СВ в задачах вычисления площадей и объемов**

Подход использованный в задаче Бюффона можно распространить на вычисление площадей и объемов произвольных областей.

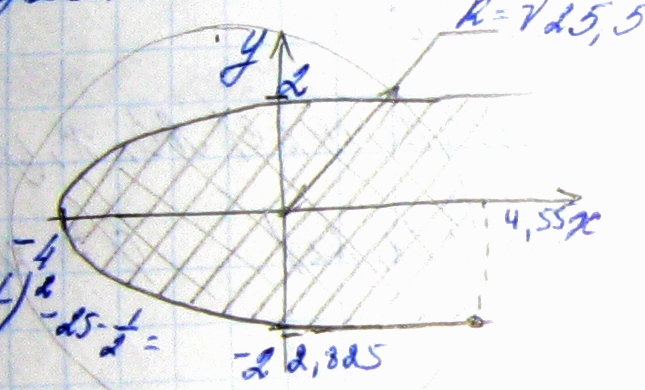


В этом подходе по умолчанию подразумевается возможность:

1. Найти такой параллелепипед V, в котором содержится область Ω
2. Иметь возможность выполнить проверку( точка принадлежит Ω)

Приведем пример документа Mathcad





ФайлMathcad































































Задаем число экспериментов:





Изобразим обращение к этой процедуре:



















**38, 40.Метод Судзуо-Какутани для решения граничных задач теории потенциала**

*Понятие граничных задач:*

Пусть имеется некоторая область Ω. Поведение сплошной среды внутри этой области описывается какими-то уравнениями, чаще всего это ДУ в частных произв. Неизвестн в этих уравнениях явл функция U(x,y,z). При этом на границе области известны либо значения самой функции либо производные от нее или какие-то их комбинации. Совокупность ДУ в частных производных кот-ым подчин неизвестн функция и граничные усл-я наз-ся граничной задачей.

*Решением* граничной задачи наз-ся такая ф-я кот удовлетв и ДУ и граничным усл-ям.

Связь методов решения граничных задач со сложностью границ области.

|  |  |
| --- | --- |
| Сложность границы | Методы решения |
| Границы простейшие (прямоугольник, круг) | Аналитические |
| «Стандартные» гран. Состоящие из плоскостей, цилиндрич поверхностей, конических | Метод конечных эл-ов, метод конечных разностей, метод граничных эл-ов, метод сеток |
| Запутанные границы | Методы типа Монте Карло, Судзуо-Какутани |

*Понятия о случайных блужданиях:*

(x2,y2)

h

h

(x0,y0)

Рассм след процесс:

Из произвольной точки (x0,y0) в случайном направл выполн шаг заданной величины. Из вновь получ точки опять в случайном направлении выполн шаг такой же величины и т.д. Если рассм много шагов такого проуесса, то появляется картина так назыв случайн блужданий. Такими процессами описыв бройновское движение частиц и др физических явлений.

(x1,y1)

Оказывается что расст А от точки блужд до текущей заваисит от кол-ва шагов по закону

A(h)=C Это означает что с цыеличением кол-ва шагов рано или поздно будет достигнуто любая сколь угодно удаленная от начала блуждания. Обратите внимание что здесь присутствует закон квадратного корня.

*Понятие о граничных задачах теории потенциала:*

Предположим что неизвестная функция в граничн задаче должна подчиняться уравнению:

Оказывается что ф-ии подчин этим ур-ям (они наз-ся гармоническими) обладают след св-ом.

**38,40**

(x0,y0)

гε

Ω

Решение в U(x0,y0) будет средним значением по любому замкнутому контуру Гε при малых диаметрах Гε окрестности. Такие ф-ии наз-ся потенциальными, они описывают задачу о распределении эл. Заряда в какой-то области. К тому типу задач относится следующее:

1-задача о распределении температур в теле при заданных значениях температуры (или конвекции тепла на пов-ти).

2-задача о перемещении точек сеч-я стержня при кручении

3-задача о распределении заряда.

Особенность этих решений задач теориипотенциала и положенного в основу метода случайных блужданий обоснованным японским математиком Судзуо-Какутани. Поясним ее на примере плоской пластины на краях которой температуры известны, а температуру в произв точке пластины нужно найти.

T(x,y) – неизвестная функция распределения температур внутри пластины.

T2(x0,y0)

(x0,y0)

Ω

ГΩ

ɥ(x,y)

T(x0,y0)=

T1(x0,y0)

Рассмотрим получение координат конца очередного шага

y0

(x0,y0)

α

h

(x1,y1)

y1=y0+hsinα

hcosα

y1

Угол альфа выбирается случайно по случайному закону распределения из интервала от нуля до 2-х пи.

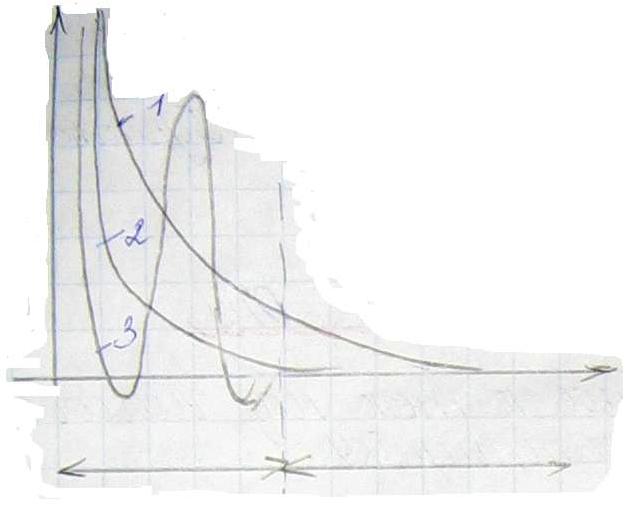
**39,42. Эрмитовы сплайны.**

Сплайн — функция в математике, область определения которой разбита на конечное число отрезков, на каждом из которых она совпадает с некоторым алгебраическим многочленом (полиномом). Максимальная из степеней использованных полиномов называется степенью сплайна. Разность между степенью сплайна и получившейся гладкостью (гладкость – порядок, до которого функция имеет непрерывные производные включительно) называется дефектом сплайна. Например, непрерывная ломаная есть сплайн степени 1 и дефекта 1. В современном понимании сплайны — это решения многоточечных краевых задач сеточными методами. Другими словами сплайн — это кусочно заданная функция, то есть совокупность нескольких функций, каждая из которых задана на каком-то множестве значений аргумента, причём эти множества попарно непересекающиеся.

Кубический эрмитов сплайн — сплайн, построенный из кубических полиномов с использованием эрмитовой интерполяции, в соответствии с которой интерполируемая функция задается не только своими значениями в n точках, но и её первыми производными. Для заданной интерполяционной сетки для *k=1..n*, и заданного значения независимой переменной x вычисление функции проводится в соответствующем интервале ( с известными граничными значениями функции p и её производной m. Для упрощения вычислений делается замена независимой переменной x на независимую переменную t по формуле . В результате такой замены левая граница интервала становится равной 0, а правая 1. Кубический полином, служащий для вычисления интерполируемой функции в соответствующем интервале имеет вид:

**41 Понятие о жестких дифференциальных уравнениях.**Существует ситуации, когда в соответствии с физическим содержанием задачи в решении присутствует несколько составляющих, имеющих существенно различные временные const в том смысле, что одни составляющие быстро изменяются по сравнению с другими.





Видно, что во 2-ой зоне вклад в решение 2-ой и 3-ей составляющей незначителен. Но для того, чтобы правильно отследить эти составляющие в 1-ой зоне потребуется достаточно мелкий шаг на всем интервале наблюдения. Такие задачи накладывают жесткие ограничения на величину шага интегрирования.

Рассмотрим следующую систему уравнений:





Через небольшое время наблюдения:



Если попытаться решить эту задачу методом Эйлера:

h=0.01 t=t0+h=0.01

u1=1+0.01(998+1998) = 30.96

v1=1+0.01(-999-1999) = -28.98

Если продолжить процесс интегрирования для следующих шагов, то расхождения окажутся еще больше. Этот пример показывает, что существуют задачи, для которых стандартные методы решения не подходят. Это пример жестких ДУ. Для решения этих задач разраб. специальные методы Гира И Пурлиж-Штерна.

**43 Метод стрельбы для решения граничных задач для ОДУ.**

Граничные задачи это когда условия задаются на границах, например для левого конца балки задано перемещение на 1, а для правого угол поворота 45.

в отличие от других задач сдесь условия задаются не только при х=0, но и при х=L.

Метод стрельбы:

αтек

α

h1

0

L

hтек

h

x

y

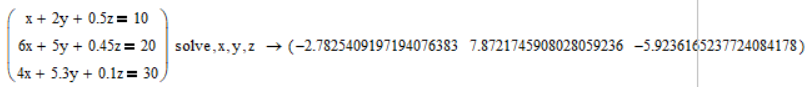
Пусть есть задача: на высоте h расположено орудие, выстрел из которого должен попасть в точку на высоте h2 на расстоянии L от орудия, усли считать что траектория описывается ф-ей y(x), то y(0)=h1, y(L)=h2. Если y(x) решение ДУ, то мы имеем граничную задачу для этого ДУ. Располагая орудие под разными углами, мы будем изменять hтек. Тогда идея для метода пристрелки в том, чтобы подобрать угол, чтобы выполнилось необх условие. Меняя углы мы будем подбирать такой угол. y(L)=h2 эквивалентно y'(x) (при х=0)=tgα

И тогда мы граничную задачу заменим начальной. В этом и заключается идея метода пристрелки.

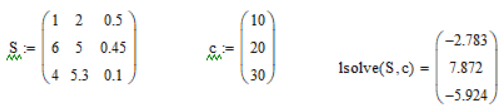
**44. Решение систем линейных алгебраических уравнений в MATHCAD.**

Есть 3 метода:

1. Ключевое слово solve, которое находится во вкладке symbolic.



2. Функция lsolve(), которая принимает матрицу коэффициентов и вектор свободных членов.



3. Численными методами Mathcad. Задаём начальные условия, а потом после слова Given пишем условия. Для нахождения ответа используем функцию Find(), которая принимает переменные с начальными условиями на вход.

