МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тульский государственный университет»

Институт высокоточных систем им. В.П. Грязева

Кафедра систем автоматического управления

Утверждено на заседании кафедры
«Системы автоматического управления»
«13» января 2021 г., протокол № 8

Заведующий кафедрой

_____О.В.Горячев

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
**по выполнению лабораторных работ
по дисциплине (модулю)**

«Численные методы математики»

**основной профессиональной образовательной программы
высшего образования – программы специалитета**

по специальности
24.05.06 Системы управления летательными аппаратами

со специализацией
Системы управления беспилотными летательными аппаратами

Форма обучения: очная

Идентификационный номер образовательной программы: 240506-01-21

Тула 2021 год

Разработчик(и) методических указаний

Макаров Н.Н., профессор, д.т.н., профессор

(ФИО, должность, ученая степень, ученое звание) (подпись)

Оглавление

1 Использование системы Mathcad в численных расчётах. 4
2 Численное решение линейных алгебраических уравнений. Вычисление обратной матрицы. 9
3 Формирование характеристического многочлена матрицы. 15
4 Интерполяция и аппроксимация функций 17
5 Аппроксимация функции ортогональными многочленами 24
6 Полиномиальные формулы дифференцирования, порядок точности. Формулы Рунге. 31
7 Численное интегрирование. Квадратурные формулы. 37
8 Решение ОДУ методом Рунге-Кутты. Интегрирование с постоянным шагом. .45
Список литературы 49
Программное обеспечение и Интернет-ресурсы 49

1 Использование системы Mathcad в численных расчётах.

Основы использования системы MathCAD

MathCAD работает с *документами*. С точки зрения пользователя, документ - это чистый лист бумаги, на котором можно размещать блоки трех основных типов: математические выражения, текстовые фрагменты и графические области.

Расположение нетекстовых блоков в документе имеет принципиальное значение – *слева направо и сверху вниз.*

Математические выражения

К основным элементам математических выражений MathCAD относятся *типы данных, операторы, функции* и *управляющие структуры*.

Операторы

Операторы - элементы MathCAD, с помощью которых можно создавать математические выражения. К ним, например, относятся символы арифметических операций, знаки вычисления сумм, произведений, производной и интеграла и т.д.

Оператор определяет:

1. действие, которое должно выполняться над операндами;
2. сколько, где и какие операнды должны быть введены в оператор.

Операнд – число или выражение, на которое действует оператор. Например, в выражении **5! + 3** число **3** и выражение **5!** – операнды оператора **+** (плюс), а число **5** операнд оператора факториал (!). После указания *операндов* операторы становятся исполняемыми по документу блоками.

Типы данных

К *типам данных* относятся числовые константы, обычные и системные переменные, массивы (векторы и матрицы) и данные файлового типа.

Константами называют поименованные объекты, хранящие некоторые значения, которые не могут быть изменены. ***Переменные*** являются поименованными объектами, имеющими некоторое значение, которое может изменяться по ходу выполнения программы. Тип переменной определяется ее значением; переменные могут быть числовыми, строковыми, символьными и т. д. Имена констант, переменных и иных объектов называют *идентификаторами*. Идентификаторы в MathCAD представляют собой набор латинских или греческих букв и цифр.

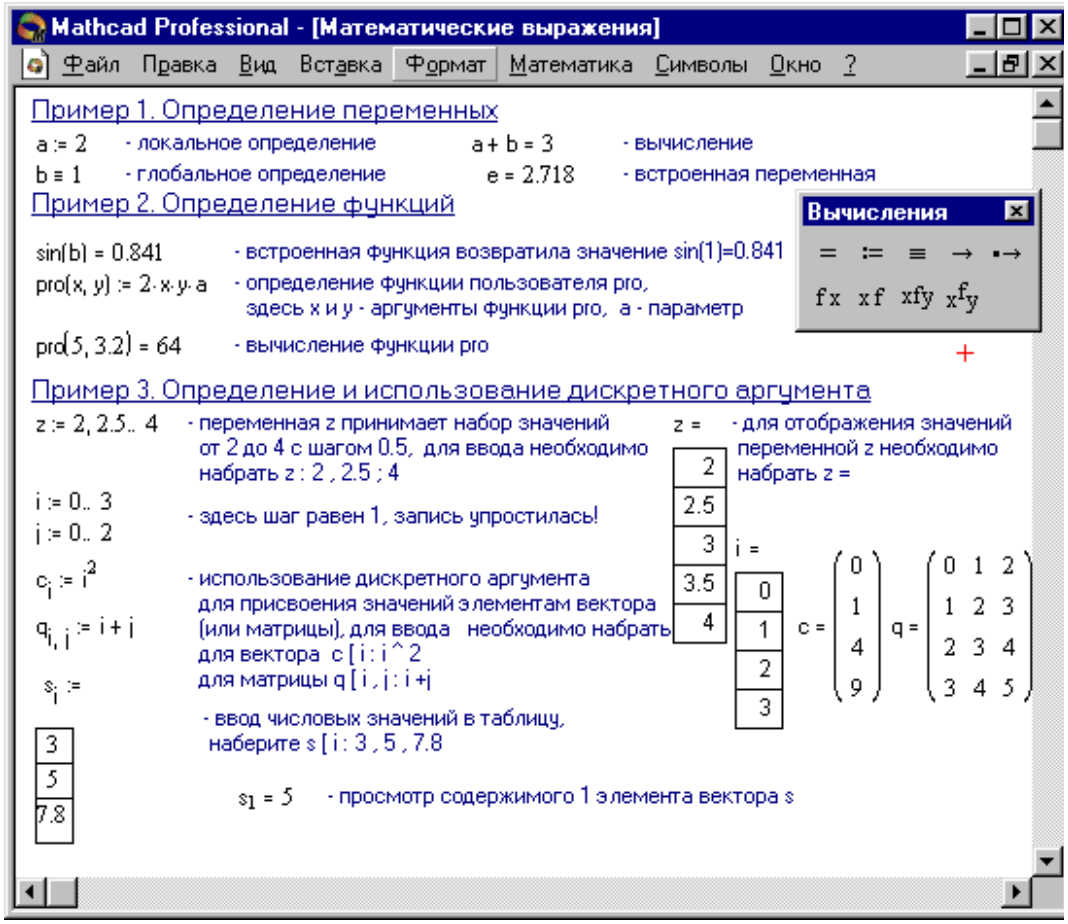


Рисунок 1. Математические выражения

В MathCAD содержится небольшая группа особых объектов, которые нельзя отнести ни к классу констант, ни к классу переменных, значения которых определены сразу после запуска программы. Их правильнее считать ***системными переменными***, имеющими предопределенные системой начальные значения (см. Приложение 1). Изменение значений системных переменных производят во вкладке **Встроенные переменные** диалогового окна **Math Options** команды **Математика ⇒ Опции.**

Обычные переменные отличаются от системных тем, что они должны быть предварительно *определены* пользователем, т. е. им необходимо хотя бы однажды *присвоить значение*. В качестве *оператора присваивания* используется знак **:=**, тогда как знак = отведен для *вывода значения* константы или переменной.

Если переменной присваивается начальное значение с помощью оператора **:=**, вызывается нажатием клавиши **:** (двоеточие) на клавиатуре, такое присваивание называется *локальным*. До этого присваивания переменная не определена и ее нельзя использовать. Однако с помощью знака **≡** (клавиша ~ на клавиатуре) можно обеспечить *глобальное* присваивание (см. Пример 1 Рисунка 1). MathCAD прочитывает весь документ дважды слева направо и сверху вниз. При первом проходе выполняются все действия, предписанные локальным оператором присваивания (**≡**), а при втором – производятся действия, предписанные локальным оператором присваивания (**:=**), и отображаются все необходимые результаты вычислений (=).

Существуют также жирный знак равенства **=** (комбинация клавиш **Ctrl** + =), который используется, например, как оператор приближенного равенства при решении систем уравнений, и символьный знак равенства → (комбинация клавиш **Ctrl** + **.**).

Дискретные аргументы - особый класс переменных, который в пакете MathCAD зачастую заменяет ***управляющие структуры***, называемые циклами (однако полноценной такая замена не является). Эти переменные имеют ряд фиксированных значений, либо целочисленных (1 способ), либо в виде чисел с определенным шагом, меняющихся от

начального значения до конечного (2 способ).

1. *Name* := *Nbegin* .. *Nend*,

где *Name* – имя переменной, *Nbegin* – ее начальное значение, *Nend* – конечное значение, **..** – символ, указывающий на изменение переменной в заданных пределах (вводится клавишей **;**). Если *Nbegin* < *Nend*, то шаг переменной будет равен +1, иначе –1.

2. *Name* := *Nbegin*, (*Nbegin* + *Step*) .. *Nend*

Здесь *Step* – заданный шаг изменения переменной (он должен быть положительным, если *Nbegin* < *Nend*, или отрицательным в обратном случае).

Дискретные аргументы значительно расширяют возможности MathCAD, позволяя выполнять многократные вычисления или циклы с повторяющимися вычислениями, формировать векторы и матрицы (Пример 3 Рисунка 1).

Массив - имеющая уникальное имя совокупность конечного числа числовых или символьных элементов, упорядоченных некоторым образом и имеющих определенные адреса. В пакете MathCAD используются массивы двух наиболее распространенных типов:

- одномерные (векторы);
- двумерные (матрицы).

Порядковый номер элемента, который является его адресом, называется *индексом*. Индексы могут иметь только целочисленные значения. Они могут начинаться с нуля или единицы, в соответствии со значением системной переменной **ORIGIN** (см. Приложение 1).

Векторы и матрицы можно задавать различными способами:

- с помощью команды **Вставка ⇒ Матрица**, или комбинации клавиш **Ctrl + M**, или щелчком на кнопке панели **Матрица,** заполнив массив пустых полей для не слишком больших массивов;
- с использованием дискретного аргумента, когда имеется некоторая явная зависимость для вычисления элементов через их индексы (Пример 3 Рисунка 1).

Функции

Функция – выражение, согласно которому проводятся некоторые вычисления с *аргументами* и определяется его числовое значение.

Следует особо отметить разницу между *аргументами* и *параметрами* функции. Переменные, указанные в скобках после имени функции, являются ее *аргументами* и заменяются при вычислении функции значениями из скобок. Переменные в правой части определения функции, не указанные скобках в левой части, являются *параметрами* и должны задаваться *до* определения функции (см. Пример 2 Рисунка 1).

Главным признаком функции является *возврат значения*, т.е. функция в ответ на обращение к ней по имени с указанием ее аргументов должна возвратить свое значение.

Функции в пакете MathCAD могут быть *встроенные* (см. Приложение 3), т. е. заблаговременно введенные разработчиками, и *определенные пользователем*.

Способы вставки встроенной функции:

1. Выбрать пункт меню **Вставка ⇒ Функция**.
2. Нажать комбинацию клавиш **Ctrl + E**.
3. Щелкнуть на кнопке .

Текстовые фрагменты

Текстовые фрагменты представляют собой куски текста, которые пользователь хотел бы видеть в своем документе. Существуют два вида текстовых фрагментов:

- *текстовая область* предназначена для небольших кусков текста - подписей, комментариев и т. п. Вставляется с помощью команды **Вставка ⇒ Текстовая регион** или комбинации клавиш **Shift + "** (двойная кавычка);
- *текстовый абзац* применяется в том случае, если необходимо работать с абзацами или страницами. Вставляется с помощью комбинации клавиш **Shift + Enter**.

Порядок выполнения лабораторной работы 1

Упражнение 1.

Вычислить:

$\sqrt{100} =$ |-10| = 10! = .

Это и все остальные задания снабдить комментариями, используя команду **Вставка ⇒ Текстовая область**.

Упражнение 2.

Определить переменные: $a := 3.4$, $b := 6.22$, $c \equiv 0.149$ (причем переменную c - глобально) и выражения:

$$Z := \frac{2ab + \sqrt[3]{c}}{\sqrt{\left(a^2 + b^{a+c}\right) \cdot c}} \qquad N := e^{\sin c} \cos\frac{a}{b}.$$

- Вычислить выражения.
- С помощью команды **Формат⇒Результат⇒Формат чисел⇒Число знаков** изменить точность отображения результатов вычисления *глобально*.

Упражнение 3.

Вывести на экран значение *системной константы* π и установить максимальный формат ее отображения *локально*.

Упражнение 4.

Выполнить следующие операции с комплексными числами:

$Z := -3 + 2i$ $|Z| =$ $Re(Z) =$ $Im(Z) =$ $arg(Z) =$

$\sqrt{Z} =$ $\sqrt{-5} =$ $2 \cdot Z =$ $Z1 := 1 + 2i$ $Z2 := 3 + 4i$

$Z1 + Z2 =$ $Z1 - Z2 =$ $Z1 \cdot Z2 =$ $Z1/Z2 =$

Упражнение 5.

Выполнить следующие операции:

$i := 1 .. 10$ $\sum_i i =$ $\prod_i (i+1) =$ $\int_0^{0.4} x^2 \cdot \lg(x+2)dx =$

$\int_{0.8}^{1.2} \frac{\operatorname{ctg} 2x}{(\sin 2x)^2} dx =$ $x := 2$ $\frac{d}{dx} x^5 =$ $\frac{d}{dx} \sin(x) =$

Контрольные вопросы

1. С помощью какого оператора можно вычислить выражение?
2. Как вставить текстовую область в документ Mathcad?
3. Чем отличается глобальное и локальное определение переменных? С помощью каких операторов они выполняются?

4. Как изменить формат отображения чисел для всего документа (глобально)?
5. Как изменить формат чисел для отдельного выражения?
6. Какие системные (предопределенные) переменные Вам известны? Как узнать их значение? Как изменить их значение?
7. Какие виды функций в Mathcad Вам известны?
8. Как вставить встроенную функцию в документ Mathcad?
9. С помощью каких операторов можно вычислять интегралы, производные, суммы и произведения?
10. Как определить дискретные переменные с произвольным шагом? Каков шаг по умолчанию?
11. Как определить индексированную переменную?
12. Какие виды массивов в Mathcad Вам известны?
13. Какая системная переменная определяет нижнюю границу индексации элементов массива?
14. Опишите способы создания массивов в Mathcad.
15. Как просмотреть содержимое массива, определенного через дискретный аргумент?

2 Численное решение линейных алгебраических уравнений. Вычисление обратной матрицы.

Решение системы линейных алгебраических уравнений (СЛАУ)

Численное решение СЛАУ возможно тремя способами: используя обратную матрицу, специальную функцию *lsolve*, или с помощью решающего блока *given – find*.

Все эти пути используют разные вычислительные методы и приводят порой к разным результатам. В особенности заметна разница результатов решения плохо обусловленных или вырожденных систем.

СЛАУ обычно записывают в матрично-векторной форме:

$$Ax = b,$$

где: A – матрица левых частей системы,

b – вектор правых частей,

x – неизвестный вектор.

Домножив обе части уравнения на обратную матрицу A^{-1}, получаем сразу в явном виде решение

$$x = A^{-1}b.$$

Например:

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9.1 \end{pmatrix} \qquad b := \begin{pmatrix} 2 \\ 5 \\ 2 \end{pmatrix} \qquad x := A^{-1} \cdot b \qquad x = \begin{pmatrix} -60 \\ 121 \\ -60 \end{pmatrix}$$

Тот же результат даст использование функции *lsolve*, которой требуется два аргумента – матрица левой части и вектор правой части:

$$lsolve(A, b) = \begin{pmatrix} -60 \\ 121 \\ -60 \end{pmatrix}$$

Наконец, можно воспользоваться решающим блоком, причём сделать это в двух вариантах: в матрично-векторной форме,

$$Given \;\; A \cdot x = b \qquad Find(x) = \begin{pmatrix} -60 \\ 121 \\ -60 \end{pmatrix}$$

или покоординатно:

$given$

$$a11\cdot x1 + a12\cdot x2 + a13\cdot x3 = b1$$

$$a21\cdot x1 + a22\cdot x2 + a23\cdot x3 = b2$$

$$a31\cdot x1 + a32\cdot x2 + a33\cdot x3 = b1$$

$$find(x1, x2, x3) = \begin{pmatrix} -60 \\ 121 \\ -60 \end{pmatrix}$$

Вычисление обратной матрицы.

Метод исключения может использоваться также при вычислении обратной матрицы. Обратная матрица является решением уравнения

$$A\cdot X = E,$$

где E – единичная матрица, а X – неизвестная матрица, которую требуется найти. Известно, что у неособой матрицы (определитель отличен от нуля) существует (единственная) обратная матрица. Если обозначить столбец матрицы верхним индексом в угловых скобках, то это уравнение можно переписать в виде набора обычных СЛАУ:

$$A\cdot X^{\langle k\rangle} = E^{\langle k\rangle}, \qquad k = 1, 2, \ldots n.$$

Поскольку матрица левой части A во всех этих уравнениях одна и та же, достаточно один раз привести её к треугольной форме. Обобщим задачу. Рассмотрим матричное уравнение

$$A\cdot X = B, \quad A = \left\{a_{i,j}\right\}_{n,n}, \quad B = \left\{b_{i,k}\right\}_{n,m}, \quad X = \left\{x_{i,k}\right\}_{n,m}.$$

Для его решения запишем алгоритм метода полного исключения, в котором будем использовать объединённую матрицу AB:

$$AB = \begin{pmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} & b_{1,1} & b_{1,2} & \ldots & b_{1,m} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,n} & b_{2,1} & b_{2,2} & \ldots & b_{2,m} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n,1} & a_{n,2} & \ldots & a_{n,n} & b_{n,1} & b_{n,2} & \ldots & b_{n,m} \end{pmatrix},$$

и запишем для неё алгоритм прямого хода, используя прежние обозначения. Исходную объединённую матрицу обозначим A^1B^1, матрицу, полученную в результате k-го цикла прямого хода, обозначим $A^{k+1}B^{k+1}$.

Очередной k-й цикл прямого хода начинаем с выбора ведущего элемента в k-ом столбце матрицы A^k: находим номер строки, содержащей ведущий элемент,

$$r = \arg\max_{k\le j\le n}\left|a^k_{j,k}\right|,$$

если этот номер отличен от k, то есть ведущий элемент не лежит на главной диагонали матрицы A, меняем местами k-ю и r-ю строки объединённой матрицы. Затем, если ведущий элемент не равен нулю (что свидетельствовало бы о вырожденности матрицы A), выполняем действия

$$c_{s,k} = \frac{a^k_{s,k}}{a^k_{k,k}}, \qquad a^{k+1}_{s,l} = a^k_{s,l} - c_{s,k}a^k_{kl}, \qquad b^{k+1}_{s,j} = b^k_{s,j} - c_{s,k}b^k_{k,j},$$

$$k = 1, 2, \ldots n-1, \qquad s, l = k+1, k+2, \ldots n, \qquad j = 1, 2, \ldots m.$$

Если на каком-то цикле ведущий элемент окажется нулём, то определитель матрицы равен нулю, ни решения, ни обратной матрицы не существует. Если же решение существует, то выполняем обратный ход по формулам:

$$x_{k,j} = \frac{1}{a_{k,k}^n}\left(b_{k,j}^n - \sum_{i=k+1}^{n} a_{k,i}^n x_{i,j}\right),\ k = n, n-1, n-2, \ldots 1, \quad j = 1, 2 \ldots m.$$

Пример

Вычислим обратную матрицу для примера 3.2. Для этого запишем соответствующее матричное уравнение :

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix} \cdot \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

которому соответствует объединённая матрица

$$A^1B^1 = \begin{pmatrix} 1 & 2 & 3 & 1 & 0 & 0 \\ 4 & 5 & 6 & 0 & 1 & 0 \\ 7 & 8 & 10 & 0 & 0 & 1 \end{pmatrix}.$$

Находим в первом столбце ведущий элемент, это 7, он стоит в 3-й строке. Меняем местами 1-ю строку с 3-й:

$$A^1B^1 = \begin{pmatrix} 7 & 8 & 10 & 0 & 0 & 1 \\ 4 & 5 & 6 & 0 & 1 & 0 \\ 1 & 2 & 3 & 1 & 0 & 0 \end{pmatrix}.$$

Выполняем первый цикл прямого хода. Пересчитываем 2-ю строку:

$$c_{2,1} = \frac{a_{2,1}^1}{a_{1,1}^1} = \frac{4}{7},$$

$$a_{2,2}^2 = a_{2,2}^1 - c_{2,1} \cdot a_{1,2}^1 = 5 - \frac{4}{7} \cdot 8 = \frac{3}{7}, \quad a_{2,3}^2 = a_{2,3}^1 - c_{2,1} \cdot a_{1,3}^1 = 6 - \frac{4}{7} \cdot 10 = \frac{2}{7},$$

$$b_{2,1}^2 = b_{2,1}^1 - c_{2,1} \cdot b_{1,1}^1 = 0 - \frac{4}{7} \cdot 0 = 0;$$

$$b_{2,2}^2 = b_{2,2}^1 - c_{2,1} \cdot b_{1,2}^1 = 1 - \frac{4}{7} \cdot 0 = 1,$$

$$b_{2,3}^2 = b_{2,3}^1 - c_{2,1} \cdot b_{1,3}^1 = 0 - \frac{4}{7} \cdot 1 = -\frac{4}{7}.$$

Теперь 3-ю строку:

$$c_{3,1}=\frac{a_{3,1}^1}{a_{1,1}^1}=\frac{1}{7},$$

$$a_{3,2}^2=a_{3,2}^1-c_{3,1}\cdot a_{1,2}^1=2-\frac{1}{7}\cdot 8=\frac{6}{7},\qquad a_{3,3}^2=a_{3,3}^1-c_{3,1}\cdot a_{1,3}^1=3-\frac{1}{7}\cdot 10=\frac{11}{7},$$

$$b_{3,1}^2=b_{3,1}^1-c_{3,1}\cdot b_{1,1}^1=1-\frac{1}{7}\cdot 0=1,$$

$$b_{3,2}^2=b_{3,2}^1-c_{3,1}\cdot b_{1,2}^1=0-\frac{1}{7}\cdot 0=0,$$

$$b_{3,3}^2=b_{3,3}^1-c_{2,1}\cdot b_{1,3}^1=0-\frac{1}{7}\cdot 1=-\frac{1}{7}.$$

Первый цикл прямого хода закончен с результатом:

$$A^2B^2=\begin{pmatrix} 7 & 8 & 10 & 0 & 0 & 1 \\ \frac{4}{7} & \frac{3}{7} & \frac{2}{7} & 0 & 1 & -\frac{4}{7} \\ \frac{1}{7} & \frac{6}{7} & \frac{11}{7} & 1 & 0 & -\frac{1}{7} \end{pmatrix}.$$

Совершим теперь 2-й цикл прямого хода, предварив его выбором ведущего элемента во втором столбце матрицы A^2B^2 вниз от главной диагонали. Это будет $\frac{6}{7}$ из третьей строки. Переставляем вторую строку с третьей:

$$A^2B^2=\begin{pmatrix} 7 & 8 & 10 & 0 & 0 & 1 \\ \frac{1}{7} & \frac{6}{7} & \frac{11}{7} & 0 & 1 & -\frac{1}{7} \\ \frac{4}{7} & \frac{3}{7} & \frac{2}{7} & 0 & 1 & -\frac{4}{7} \end{pmatrix}.$$

Наконец выполняем второй цикл прямого хода:

$$c_{3,2}=\frac{a_{3,2}^2}{a_{2,2}^2}=\frac{3/7}{6/7}=\frac{1}{2},$$

$$a_{3,3}^3=a_{3,3}^2-c_{3,2}\cdot a_{2,3}^2=\frac{2}{7}-\frac{1}{2}\cdot\frac{11}{7}=-\frac{1}{2},$$

$$b_{3,1}^3=b_{3,1}^2-c_{3,2}\cdot b_{2,1}^2=0-\frac{1}{2}\cdot 1=-\frac{1}{2},$$

$$b_{3,2}^3=b_{3,2}^2-c_{3,2}\cdot b_{2,2}^2=1-\frac{1}{2}\cdot 0=1,$$

$$b_{3,3}^3=b_{3,3}^2-c_{3,2}\cdot b_{2,3}^2=-\frac{4}{7}-\frac{1}{2}\cdot\left(-\frac{1}{7}\right)=-\frac{1}{2}.$$

То есть, окончательно имеем

$$A^3B^3=\begin{pmatrix} 7 & 8 & 10 & 0 & 0 & 1 \\ \frac{1}{7} & \frac{6}{7} & \frac{11}{7} & 1 & 0 & -\frac{1}{7} \\ \frac{4}{7} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 1 & -\frac{1}{2} \end{pmatrix}.$$

Выполним теперь обратный ход по формулам:

$$x_{3,1}=\frac{1}{a_{3,3}^3}\cdot b_{3,1}^3=-\frac{1}{1/2}\cdot\left(-\frac{1}{2}\right)=1,$$

$$x_{3,2}=\frac{1}{a_{3,3}^3}\cdot b_{3,2}^3=-\frac{1}{1/2}\cdot 1=-2,$$

$$x_{3,3}=\frac{1}{a_{3,3}^3}\cdot b_{3,3}^3=-\frac{1}{1/2}\cdot\left(-\frac{1}{2}\right)=1,$$

$$x_{2,1}=\frac{1}{a_{2,2}^3}\cdot\left(b_{2,1}^3-a_{2,3}^3\cdot x_{3,1}\right)=\frac{1}{6/7}\cdot\left(1-\frac{11}{7}\cdot 1\right)=-\frac{2}{3},$$

$$x_{2,2}=\frac{1}{a_{2,2}^3}\cdot\left(b_{2,2}^3-a_{2,3}^3\cdot x_3\right)=\frac{1}{6/7}\cdot\left(0-\frac{11}{7}\cdot(-2)\right)=\frac{11}{3},$$

$$x_{2,3}=\frac{1}{a_{2,2}^3}\cdot\left(b_{2,3}^3-a_{2,3}^3\cdot x_{3,3}\right)=\frac{1}{6/7}\cdot\left(-\frac{1}{7}-\frac{11}{7}\cdot 1\right)=-2,$$

$$x_{1,1}=\frac{1}{a_{2,2}^3}\cdot\left(b_{1,1}^3-a_{1,3}^3\cdot x_{3,1}-a_{1,2}^3\cdot x_{2,1}\right)=\frac{1}{7}\cdot\left(0-10\cdot 1-8\cdot\left(-\frac{2}{3}\right)\right)=-\frac{2}{3},$$

$$x_{2,2}=\frac{1}{a_{2,2}^3}\cdot\left(b_{2,2}^3-a_{2,3}^3\cdot x_3\right)=\frac{1}{7}\cdot\left(0-10\cdot(-2)-8\cdot\frac{11}{3}\right)=-\frac{4}{3},$$

$$x_{2,3}=\frac{1}{a_{2,2}^3}\cdot\left(b_{2,3}^3-a_{2,3}^3\cdot x_{3,3}\right)=\frac{1}{7}\cdot\left(1-10\cdot 1-8\cdot(-2)\right)=1.$$

Таким образом, получена обратная матрица

$$X=\begin{pmatrix}-\frac{2}{3} & -\frac{4}{3} & 1\\ -\frac{2}{3} & \frac{11}{3} & -2\\ 1 & -2 & 1\end{pmatrix},$$

правильность которой нетрудно проверить непосредственной подстановкой.

Порядок выполнения лабораторной работы

Упражнение 1.

Тремя способами решить систему линейных алгебраических уравнений. Проверить подстановкой. Сравнить решения, полученные разными методами. Оценить обусловленность системы численно, варьируя правую часть, и аналитически, используя функцию *cond.* Сделать выводы.

Варианты коэффициентов СЛАУ

№	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	a_{31}	a_{32}	a_{33}	b_1	b_2	b_3
1	-11.786	0.76	2.442	-8.01	-1.181	-10.631	6.8	0.477	9.023	10.942	0.944	-0.91
2	8.693	6.712	11.923	2.676	-5.611	8.163	-2.979	4.252	-11.788	-5.379	2.11	8.103
3	-0.362	5.849	-1.009	5.866	2.377	5.64	1.738	-8.363	-1.796	0.411	6.037	-7.944
4	-0.195	4.794	-8.46	-8.602	4.629	-1.763	11.198	-8.322	7.72	-7.408	7.612	-8.266
5	5.568	-5.29	4.374	5.326	-9.047	8.032	0.408	-1.771	10.784	1.189	-0.679	8.327
6	-1.054	11.591	5.74	-7.296	8.146	0.022	-11.34	1.742	0.752	8.233	3.782	8.211
7	-9.361	-4.462	-5.134	-8.633	8.031	2.406	-5.935	-11.961	7.35	-6.946	1.277	-9.269
8	6.053	1.042	-1.519	4.709	-1.52	1.869	3.088	0.1	4.698	-7.441	-7.719	-1.021
9	-9.659	-9.734	10.356	9.47	-6.544	-2.143	3.074	-1.161	2.348	8.515	2.995	1.578
10	-7.577	1.323	-6.171	2.513	2.031	-0.133	5.778	2.889	7.308	1.825	9.877	5.464
11	4.027	-4.44	-4.66	-9.394	8.429	-8.283	-10.096	3.384	1.082	-2.185	-0.827	-8.336
12	5.712	7.841	8.96	-4.798	-8.946	6.839	2.632	-10.265	3.692	-9.484	-6.55	10.079
13	3.907	-0.179	-0.076	0.222	4.516	2.551	-11.858	-9.585	8.718	5.936	-2.877	1.267
14	10.937	-7.762	-8.841	10.841	-11.332	-10.658	-8.84	8.743	5.319	-11.648	4.99	-6.779
15	-7.946	-3.816	-3.178	7.257	0.637	7.156	-8.524	-2.342	-8.723	-10.412	1.76	1.153
16	-4.528	7.392	-5.178	9.485	5.851	-3.333	-6.516	-6.493	1.018	-10.709	0.592	-9.728
17	9.396	-8.483	10.359	-10.87	-3.926	-2.434	10.708	0.822	4.645	-5.78	0.614	-2.431
18	2.053	4.416	-1.548	-11.827	2.437	1.84	-6.664	-11.897	-10.026	8.522	7.465	4.596
19	0.723	-8.573	5.47	-11.254	3.469	-9.949	10.749	-5.401	11.562	-4.986	-3.179	-6.753

Упражнение 2.

Вычислить с использованием решающего блока обратную матрицу от матрицы левых частей из упражнения 3. Сравнить с результатом, полученным штатным инструментом *Mathcad*. Проверить оба результата подстановкой. Оценить точность результата с использованием обусловленности матрицы.

Упражнение 3.

Выполнить метод Жордана-Гаусса вручную. Сравнить полученный результат с полученными ранее.

Контрольные вопросы

1. Какие три средства имеет Mathcad для решения систем линейных алгебраических уравнений?
2. Перечислите отличия в использовании решающего блока и функции *lsolve* при решении систем линейных алгебраических уравнений.
3. Можно ли и как вычислить обратную матрицу с помощью функции *lsolve*.
4. Какие аргументы требуются функции *lsolve*?

3 Формирование характеристического многочлена матрицы.

Теоретические сведения

Известно, что по значениям многочлена $P_n(\lambda) = (-\lambda)^n + \sum_{k=0}^{n-1} p_k \lambda^k$ степени n в n различных точках однозначно определяются все его коэффициенты. Действительно, пусть имеются точки x_k, $k = 1, 2, ...n$ и значения многочлена в них, $y_k = P_n(x_k)$, $k = 1, 2, ...n$. Это позволяет записать систему линейных алгебраических уравнений относительно коэффициентов многочлена:

$$\sum_{i=0}^{n-1} p_i \cdot x_k^i = y_k - (-1)^n x_k^n, \quad k = 1, 2, ...n.$$

Матрица этой системы имеет вид

$$V = \begin{pmatrix} 1 & x_1 & x_1^2 & ... & x_1^{n-1} \\ 1 & x_2 & x_2^2 & ... & x_2^{n-1} \\ 1 & x_3 & x_3^2 & ... & x_3^{n-1} \\ ... & ... & ... & ... & ... \\ 1 & x_n & x_n^2 & ... & x_n^{n-1} \end{pmatrix},$$

определитель такой матрицы, известный как *определитель Вандермонда* (*вандермондиан*), всегда отличен от нуля, если среди иксов нет повторяющихся значений. Известна даже формула вычисления вандермондиана:

$$\det(V) = \prod_{1 \le i < j \le n} \left(x_j - x_i\right).$$

Решая СЛАУ любым методом, находим коэффициенты характеристического многочлена.

Пример

Рассмотрим использование системы Mathcad для применения метода интерполяции.

Зададимся порядком матрицы.

$n := 3 \qquad ORIGIN := 1$

Сформируем диапазоны индексов элементов

$i := 1..3 \quad j := i$

Заполним матрицу случайными числами, распределёнными равномерно в диапазоне от -100 до 100

$X_{i,j} := \mathrm{runif}(1, -100, 100)_1 = \ldots$

Определим единичную матрицу

$E := \mathrm{identity}(3)$

Составим вектор значений, в которых будем вычислять определитель

$$\lambda := \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

И вычислим в этих точках значения правых частей линейной системы, которую будем решать

$y_i := \left|X - \lambda_i \cdot E\right| - \left(-\lambda_i\right)^n$

Матрица левой части этой системы состоит из степеней лямбд, начиная с нулевой

$A_{i,j} := \left(\lambda_i\right)^{j-1}$

Решаем полученную СЛАУ и получаем коэффициенты характеристического многочлена кроме старшего

$c := \mathrm{lsolve}(A, y)$

Коэффициент при старшей степени равен

$c_{n+1} := (-1)^n$

Находим корни этого многочлена

$$\Lambda := \mathrm{polyroots}(c) = \begin{pmatrix} -48.546 \\ 28.463 \\ 78.255 \end{pmatrix}$$

и сравниваем их с собственными числами, вычисленными маткадовской функцией

$$\text{eigenvals}(X) = \begin{pmatrix} -48.546 \\ 78.255 \\ 28.463 \end{pmatrix}$$

Те же числа, только в другом порядке.

Задание

Упражнение 1.

С помощью системы Mathcad составить и решить систему уравнений для нахождения коэффициентов характеристического уравнения матрицы 4-го порядка. Сформировать случайную матрицу и найти её характеристический многочлен методом интерполяции. Проверить результат штатными средствами системы Mathcad.

Упражнение 2.

Сформировать характеристический многочлен той же матрицы методом Данилевского. Сравнить с предыдущим методом.

Контрольные вопросы

5. Что такое собственное число матрицы?
6. Что такое характеристический многочлен матрицы?
7. Какие методы вычисление коэффициентов характеристического многочлена матрицы вам известны? В чём они заключаются?
8. Можно ли и как вычислить собственные числа матрицы с помощью системы Mathcad?

4 Интерполяция и аппроксимация функций

Интерполяция функций

Пусть имеется таблично заданная функция (таблица данных), представленная двумя векорами:

$$x := (1 \quad 2 \quad 3)^T \qquad y := (0 \quad 2 \quad -1)^T$$

Линейную интерполяцию этой таблицы осуществляет функция linterp. Ей требуется три аргумента: два вектора x и y, содержащие интерполируемую таблицу, и один скаляр - значение аргумента интерполируемой функции, для которого требуется вычислить интерполяционное значение. Вычислим интерполяционное значение в точке 1.25

$$linterp(x, y, 1.25) = 0.5$$

Интерполяция кубическим сплайном выполняется несколько сложнее. Она состоит из двух последовательных этапов: сначала по таблице данных вычисляются коэффициенты сплайна, для чего служит любая из трёх функций *cspline*, *lspline*, *pspline*. Они различаются лишь дополнительными условиями, дополняющими уравнения интерполяции и условия непрерывности первой и второй производных. Аргументами всех этих функций являются вектора x и y, то есть таблица данных.

$$c_c := cspline(x, y) \qquad c_l := lspline(x, y) \qquad c_p := pspline(x, y)$$

$$c_c = \begin{pmatrix} 0 \\ 3 \\ 2 \\ -10 \\ -5 \\ 0 \end{pmatrix} \qquad c_l = \begin{pmatrix} 0 \\ 3 \\ 0 \\ 0 \\ -7.5 \\ 0 \end{pmatrix} \qquad c_p = \begin{pmatrix} 0 \\ 3 \\ 1 \\ -5 \\ -5 \\ -5 \end{pmatrix}$$

Для вычисления интерполированных значений в любом варианте сплайна используется функция *interp* которой надо передать четыре аргумента – вычисленный вектор коэффициентов, вектор значений аргумента и вектор значений функции, которую интерполируют, и, наконец, то значение, для которого вычисляется интерполированное значение. Например в точке 1,25 значение интерполированной функции вычислится так:

$$interp(c_c, x, y, 1.25) = 1.242$$

Исходная таблица содержит 3 узла, следовательно, сплайн включает 2 кубических многочлена и имеет 8 коэффициентов, однако вычисляется только 6. Последние *n* (в нашем случае – 2) из вычисленных коэффициентов – это значения второй производной сплайна в узловых точках.

К сожалению, классическая ньютонова интерполяция полиномами в *Mathcad* не реализована. Но её нетрудно осуществить самостоятельно, воспользовавшись ньютоновой формулой

$$\varphi(x) = \sum_{i=1}^{n} y_i \cdot \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)}.$$

Аппроксимация функций методом наименьших квадратов

Сама по себе идея интерполяции, то есть требования, чтобы приближённая функция в точности проходила через все узлы имеющейся таблицы, не всегда оправдана. Во-первых, при длинных таблицах она приводит к сложным приближениям. Во-вторых, она не кажется разумной в тех случаях, когда таблица заведомо не точна, что всегда имеет место при использовании экспериментальных данных. Часто более полезным оказывается потребовать, чтобы приближённая функция проходила близко к узлам таблицы. Дадим этой близости количественный смысл:

$$\varepsilon\left(\{x_i, y_i\}_n, \varphi(x, \alpha)\right) = \sum_{i=1}^{n} \left(y_i - \varphi(x_i, \alpha)\right)^2$$

Функционал качества приближения носит название *среднеквадратичного отклонения* (СКО). Он равен сумме квадратов "промахов" приближённой функции по всем узлам таблицы. Теперь осталось лишь подобрать значение параметра α таким, чтобы СКО оказалось наименьшим из всех возможных:

$$\sum_{i=1}^{n}\left(y_i-\varphi\left(x_i,\alpha\right)\right)^2 \xrightarrow[\alpha]{} \min \qquad (1)$$

В точке минимума функции, как известно, все её частные производные обращаются в ноль, что приводит нас к следующей системе уравнений:

$$\frac{\partial}{\partial\alpha_k}\varepsilon\left(\left\{x_i,y_i\right\}_n,\varphi\left(x,\alpha\right)\right)=0,\quad k=1,2\text{K}\ m$$

Для задачи эта система уравнений приобретает вид

$$\sum_{i=1}^{n}\left[\varphi\left(x_i,\alpha\right)\frac{\partial}{\partial\alpha_k}\varphi\left(x_i,\alpha\right)\right]=\sum_{i=1}^{n}\left[y_i\frac{\partial}{\partial\alpha_k}\varphi\left(x_i,\alpha\right)\right],$$

$$k=1,2\text{K}\ m.$$

Задача минимизации СКО особенно просто решается, если в качестве приближённой функции использовать *обобщённый многочлен*.

Выберем произвольно m функций $\varphi_1\left(x\right),\varphi_2\left(x\right),\text{K}\ \varphi_n\left(x\right)$ и будем в качестве приближённой использовать их взвешенную сумму:

$$\varphi\left(x\right)=\sum_{i=1}^{m}\alpha_i\varphi_i\left(x\right). \qquad (2)$$

Именно такую взвешенную сумму и называют *обобщённым многочленом.* Входящие в него фиксированные функции $\varphi_i\left(x\right)$ называются *базисными*, а их совокупность – *базисом*. Подставляя (2) в (1), получаем окончательную формулировку задачи приближения:

$$\sum_{i=1}^{n}\left(y_i-\sum_{k=1}^{m}\alpha_k\varphi_k\left(x_i\right)\right)^2 \xrightarrow[\alpha]{} \min$$

Легко убедиться, что эта задача приводит к системе линейных алгебраических уравнений относительно коэффициентов обобщённого многочлена:

$$A\alpha=b,$$

где A – квадратная матрица размерности m.

b – вектор той же размерности,

определяемые выражениями

$$A_{i,j}=\sum_{k=1}^{n}\varphi_i\left(x_k\right)\varphi_j\left(x_k\right),$$

$$b_i=\sum_{k=1}^{n}y_k\varphi_i\left(x_k\right),\quad i,j=1,2\text{K}\ m.$$

Эта система всегда имеет единственное решение, если только матрица A – невырожденная. Такой способ получения приближённой функции получил название *метода наименьших квадратов* (МНК).

Описанный подход к получению приближённой функции встречается в литературе под разными названиями. В русскоязычных публикациях наиболее распространены термины *аппроксимация* и *регрессия*. В англоязычных публикациях кроме последнего термина, *Regression*, часто используется выражение *подгонка* (*Curve Fitting*).

В Маткаде достаточно большая группа функций нацелена на среднеквадратичную аппроксимацию (*Curve Fitting and Regression*). Из них следует выделить две наиболее универсальные – *genfit* и *linfit*. Кроме того часто полезно использовать функцию полиномиальной аппроксимации *regress*.

***linfit*(*vx*,*vy*,*F*)** возвращает вектор коэффициентов обобщённого многочлена, аппроксимирующего функцию, заданную таблично. Поскольку задача сводится к решению СЛАУ, начальное приближение не требуется.

- ***vx*** и ***vy*** – вектора одинаковой размерности, содержащие иксы и игреки таблично заданной функции,
- ***F*(*x*)** – векторная функция скалярного аргумента, координаты этой функции образуют базис аппроксимации, при вызове используется без аргументов.

Размерность векторов данных не может быть ниже, чем размерность базиса аппроксимации. *В случае совпадения размерностей мы получим интерполирующую функцию!*

***genfit*(*vx*,*vy*,*vg*,*F*)** возвращает вектор параметров аппроксимации.

- ***vx*** и ***vy*** - как и выше, вектора, содержащие исходную функцию,
- ***vg*** – вектор начальных приближений вычисляемых параметов,
- ***F*(*x*,*u*)** – векторная функция, в которой первая координата содержит собственно аппроксимирующую функцию, описанную как функцию собственно независимого аргумента и вектора параметров или списка параметров. Задача как раз и заключается в определении значений этих параметров. Остальные координаты – частные производные по параметрам. Все координаты должны иметь идентичный список аргументов.

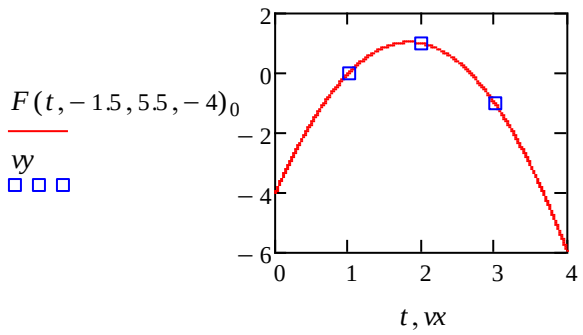
Функция, *genfit*, использует итерационные численные методы и поэтому нуждается в начальном приближении и не гарантирует результата, *linfit* и *regress* всегда дают единственный ответ и не нуждаются в начальном приближении.

Пример использования функции *genfit*:

$$F(x,a,b,c) := \begin{pmatrix} a\cdot x^2 + b\cdot x + c \\ x^2 \\ x \\ 1 \end{pmatrix}$$

$$vx := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad vy := \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} \qquad vg := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

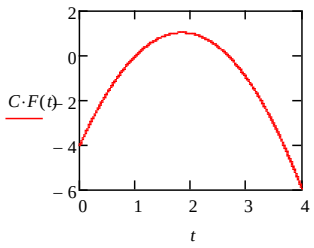
$$genfit(vx,vy,vg,F) = \begin{pmatrix} -1.5 \\ 5.5 \\ -4 \end{pmatrix}$$



Пример несколько примитивен, поскольку параметры *a,b,c* входят в аппроксимирующую функцию линейно, и логичнее было бы использовать функцию *linfit*, а не *genfit*. Как в следующем примере.

Пример использования функции *linfit*:

$$F(x) := \begin{pmatrix} x^2 \\ x \\ 1 \end{pmatrix} \qquad C := linfit(vx, vy, F) = \begin{pmatrix} -1.5 \\ 5.5 \\ -4 \end{pmatrix}$$



Порядок выполнения лабораторной работы

Для упражнений 1-3 требуется:

1) Табулировать заданную функцию в интервале (0; 3), то есть составить таблицу значений *x* и *y* в виде двух векторов размерности 5.
2) Построить графики исходной функции, табулированной функции и приближённой функции.
3) Оценить погрешность приближения и сделать вывод о качестве приближения.
4) Подобрать рациональный шаг таблицы (возможно, не постоянный).

Упражнение 1.

Выполнить линейную интерполяцию заданной функции.

Упражнение 2.

Выполнить интерполяцию заданной функции по формуле Ньютона.

Упражнение 3.

Выполнить интерполяцию заданной функции сплайном с различными доопределениями вторых производных. Сравнить результаты.

Упражнение 4.

Выполнить аппроксимацию таблично заданной функции с использованием функции *linfit*. Построить графики исходной и аппроксимированной функций.

Упражнение 5.

Выполнить аппроксимацию таблично заданной функции с использованием функции *genfit*. Построить графики исходной и аппроксимированной функций.

Варианты заданий для упражнений 1-3

№		№	
1	$f1(t) := e^{-2\cdot t}\cdot sin(3\cdot t)$	7	$f11(t) := ln(t+1)\cdot sin(3\cdot t)$
2	$f2(t) := sin(2.5t)\cdot cos(t)$	8	$f12(t) := sin(2.5\cdot t)\cdot cos\left(\frac{1.3}{t+1}\right)$
3	$f3(t) := e^{-t}\cdot cos(3\cdot t)$	9	$f13(t) := ln(t+1)\cdot cos(3\cdot t)$
4	$f4(t) := e^{-t}\cdot sin(4\cdot t)$	10	$f14(t) := sin\left(4\cdot t\cdot e^{-t}\right)$
5	$f5(t) := sin(2\cdot t)\cdot cos(1.3t)$	11	$f15(t) := sin(2\cdot t)\cdot cos(ln(3\cdot t+.1))$
6	$f6(t) := e^{-1.3\cdot t}\cdot cos(2\cdot t)$	12	$f16(t) := e^{-1.3\cdot t}\cdot cos\left(2\cdot t^2\right)$

Варианты заданий для упражнений 4-5

Значения аппроксимируемой функции по вариантам

X	0	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3
Y1	-0.022	0.396	0.27	0.023	-0.124	-0.046	-0.027	0.028	0.116	0.045	0.05
Y2	0.043	0.697	0.857	0.431	0.055	-0.078	0.257	0.425	0.174	-0.443	-0.954
Y 3	1.028	0.448	-0.12	-0.304	-0.305	-0.047	0.16	0.167	-0.09	-0.124	-0.035
Y 4	-0.031	0.631	0.376	-0.141	-0.285	-0.062	0.093	0.084	-0.05	-0.073	0.024
Y 5	0.009	0.548	0.7	0.379	0.021	-0.044	0.329	0.775	0.961	0.873	0.189
Y 6	0.936	0.597	0.146	-0.131	-0.207	-0.142	-0.085	-0.019	0.028	0.031	-0.015
Y 7	-0.081	0.276	0.506	0.217	-0.342	-0.811	-0.812	-0.083	0.901	1.203	0.545
Y 8	0.003	0.339	0.669	0.63	0.128	-0.476	-0.932	-0.84	-0.314	0.402	0.823
Y 9	-0.142	0.227	-0.067	-0.573	-0.757	-0.239	0.61	1.145	0.724	-0.254	-1.332
Y 10	0.019	0.798	1.056	0.962	1.02	1.051	0.837	0.922	0.712	0.687	0.51
Y 11	-0.062	0.586	0.651	0.495	0.196	0.08	-0.002	0.226	0.427	0.402	0.176
Y 12	1.026	0.707	0.391	0.03	-0.19	-0.065	0.118	-0.061	-0.021	0.093	-0.048

Базисы аппроксимации по вариантам.

F_0 – для аппроксимации с помощью функции *genfit*.

F_1 – F_4 – для аппроксимации с помощью функции *linfit*

№	F_0	F_1	F_2	F_3	F_4
1	$a\cdot e^{-b\cdot t}\cdot sin(c\cdot t)$	1	x	x^2	x^3
2	$a\cdot sin(b\cdot t)\cdot cos(c\cdot t)$	1	x	x^2	x^3
3	$a\cdot e^{-b\cdot t}\cdot cos(c\cdot t)$	1	x	x^2	x^3
4	$a\cdot e^{-b\cdot t}\cdot sin(c\cdot t)$	1	x	x^2	x^3
5	$a\cdot sin(b\cdot t)\cdot cos(c\cdot t)$	1	x	x^2	x^3
6	$a\cdot e^{-b\cdot t}\cdot cos(c\cdot t)$	1	x	x^2	x^3
7	$a\cdot ln(t+b)\cdot sin(c\cdot t)$	1	x	x^2	x^3
8	$sin(a\cdot t)\cdot cos\left(\frac{b}{t+c}\right)$	1	x	x^2	x^3
9	$a\cdot ln(t+b)\cdot cos(c\cdot t)$	1	x	x^2	x^3
10	$a\cdot sin\left(b\cdot t\cdot e^{-c\cdot t}\right)$	1	x	x^2	x^3
11	$sin(a\cdot t)\cdot cos(ln(b\cdot t+c))$	1	x	x^2	x^3
12	$a\cdot e^{-b\cdot t}\cdot cos\left(c\cdot t^2\right)$	1	x	x^2	x^3

Контрольные вопросы

В чём состоит интерполяция функции с математической точки зрения?

Какие аргументы требуются для функции *interp*?

Что вычисляет функция *interp*?

Что вычисляет и какие аргументы использует функция *linterp*?

Какие аргументы используют функции *cspline, pspline, lspline?*

Что вычисляют функции cspline, pspline, lspline?

Что такое сплайн

В чём заключается метод наименьших квадратов?

Что вычисляет и какие аргументы использует функция *linfit*?

Что вычисляет и какие аргументы использует функция *genfit*?

5 Аппроксимация функции ортогональными многочленами

Теоретические сведения

Существует также ряд ортонормированных базисов, основанных на степенях *x*. Они отличаются, во-первых, интервалами определения, во-вторых – весовыми функциями. Эти базисы получили название ортогональных многочленов. Рассмотрим наиболее важные из них. Первый мы уже почти построили, он именуется *многочленами Лежандра*, они определены на отрезке [-1,1] и имеют единичную весовую функцию.

Ниже приведены младшие члены некоторых бесконечных ортогональных степенных базисов и их авторы.

$$\begin{array}{ccccc}
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 & 1 & 1 \\
x\cdot\sqrt{\frac{3}{2}} & x & 2\cdot x & \sqrt{2}\cdot x & x-1 \\
\sqrt{5}\cdot\frac{3\cdot x^2-1}{2\cdot\sqrt{2}} & 2\cdot x^2-1 & 4\cdot x^2-1 & \frac{2\cdot x^2-1}{\sqrt{2}} & \frac{x^2-4\cdot x+2}{2} \\
\sqrt{7}\cdot\frac{5\cdot x^3-3\cdot x}{2\cdot\sqrt{2}} & 4\cdot x^3-3\cdot x & 8\cdot x^3-4\cdot x & \frac{2\cdot x^3-3\cdot x}{\sqrt{3}} & \frac{x^3-9\cdot x^2+18\cdot x-6}{6} \\
3\cdot\frac{35\cdot x^4-30\cdot x^2+3}{8\cdot\sqrt{2}} & 8\cdot x^4-8\cdot x^2+1 & 16\cdot x^4-12\cdot x^2+1 & \frac{16\cdot x^4-48\cdot x^2+12}{8\cdot\sqrt{6}} & \frac{x^4-16\cdot x^3+72\cdot x^2-96\cdot x+24}{24} \\
\sqrt{11}\cdot\frac{63\cdot x^5-70\cdot x^3+15\cdot x}{8\cdot\sqrt{2}} & 16\cdot x^5-20\cdot x^3+5\cdot x & 32\cdot x^5-32\cdot x^3+6\cdot x & \frac{32\cdot x^5-160\cdot x^3+120\cdot x}{16\cdot\sqrt{15}} & 1 \\
\textit{Лежандр} & \textit{Чебышев_1} & \textit{Чебышев_2} & \textit{Эрмит} & \textit{Лагерр}
\end{array} \qquad (0.1)$$

В следующей таблице приведены весовые функции и интервалы, на которых определены соответствующие базисы.

С именем Чебышева связано два типа многочленов, которые называют соответственно «*многочлены Чебышева первого рода*» и «*многочлены Чебышева второго рода*».

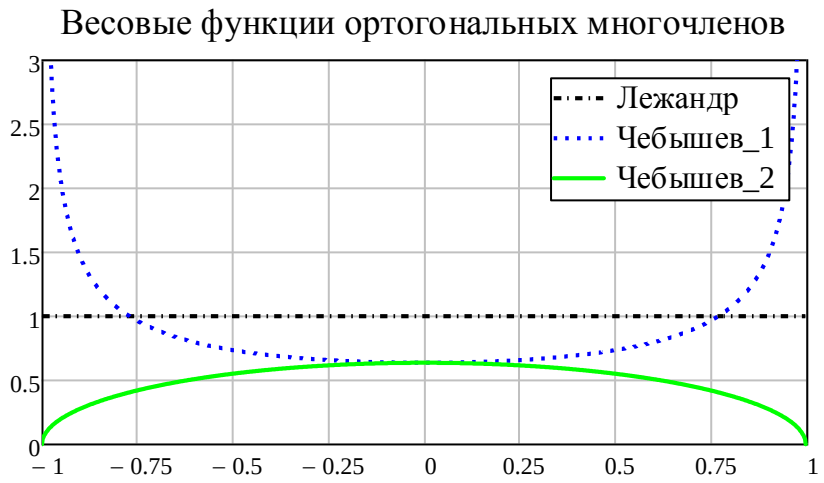
$$w(x) := \begin{bmatrix} 1 & \textit{Лежандр} & \text{"[-1,1]"} \\ \frac{2}{\pi\cdot\sqrt{\left(1-x^2\right)}} & \textit{Чебышев_1} & \text{"[-1,1]"} \\ \frac{2\cdot\sqrt{1-x^2}}{\pi} & \textit{Чебышев_2} & \text{"[-1,1]"} \\ \frac{exp\left(-x^2\right)}{\sqrt{\pi}} & \textit{Эрмит} & \text{"[-∞,∞]"} \\ exp(-x) & \textit{Лагерр} & \text{"[0,∞]"} \end{bmatrix}$$

Базисы Лежандра и Чебышёва определены на конечном отрезке $[-1,1]$. Если же требуется приблизить функцию на другом интервале, то придётся применить линейное преобразование переменных, переводящих требуемый отрезок $[a,\ b]$ в отрезок $[-1,1]$:

$$z(x)=-1+\frac{2}{b-a}(x-a),\quad x(z)=a+\frac{b-a}{2}(z+1),$$
$$x\in[a,b],\quad z\in[-1,1]. \tag{0.2}$$

Преобразовав по формуле (0.2) исходную функцию $y(x),\ x\in[a,b]$, в функцию $f(z)=y\left(a+\frac{b-a}{2}(z+1)\right)$, применим процедуру аппроксимации к этой новой функции и получим соответствующую приближенную функцию $\varphi(z)$. Если же надо вычислить приближение в точке *x*, то мы должны будем использовать её преобразованное значение: $\varphi(z(x))=\varphi\left(-1+\frac{2}{b-a}(x-a)\right)$

Теперь остановимся на различиях между базисами Лежандра и Чебышёва, которые заключаются в различных весовых функциях. На рисунке приведены графики соответствующих весовых функций.



В базисе Лежандра весовая функция постоянна на всём интервале. Это значит, что ошибка аппроксимации в любой точке интервала одинаково влияет на СКО. Результат аппроксимации при этом должен, по идее, обеспечить примерно одинаковую точность на всём интервале.

В базисе Чебышева 1-го рода весовая функция невелика в середине интервала и стремится к бесконечности на его концах. Это значит, что в формировании СКО основную долю вносят погрешности аппроксимации вблизи концов интервала, и результирующее приближение, минимизирующее СКО, может быть достаточно грубым в середине интервала, но вблизи концов погрешности должны быть небольшими, а сами концы вообще должны воспроизводиться максимально точно.

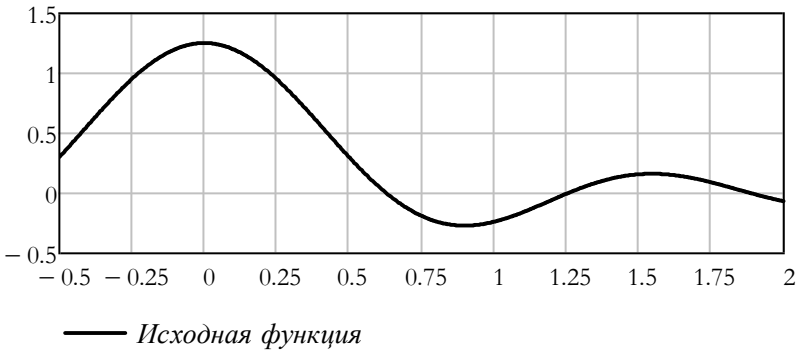
В базисе Чебышева 2-го рода весовая функция в середине интервала практически постоянна, а вот при приближении к краям убывает и стремится к нулю. Значит СКО в основном складывается из погрешностей в средних точках, а края интервала на СКО влияют слабо. Можно ожидать, что результат будет близок к исходной функции в середине интервала и может сильно отличаться от неё на краях.

Базис Эрмита определён на всей числовой оси, а базис Лагерра – на положительной полуоси. Использовать первый имеет смысл, если исходная функция интересует нас на всей числовой оси и преобразования координат тут не обязательны, а второй – когда исходная функция равна нулю или не интересна нам левее какого-то числа x_0. Если $x_0 \neq 0$, то можно использовать сдвиг вдоль оси абсцисс: $z = x - x_0, \quad x = z + x_0$.

Пример 8.2

Будем аппроксимировать функцию

$$y(x) = \frac{\sin(5x)}{4x}, \quad x \in [-0.5, 2.0], \quad a = -0.5, \quad b = 2..$$



Используем первые 6 членов базиса Лежандра (0.1). Для использования свойств ортонормированности этого базиса преобразуем аргумент по формуле (0.2):

$$x(z) = a + \frac{b-a}{2}(z+1), \quad z \in [-1,1],$$

$$y(z) = \frac{\sin(5x(z))}{4x(z)},$$

и аппроксимируем теперь уже $y(z)$ на отрезке [-1, 1]. Вычислим, учитывая ортонормированность базиса, коэффициенты аппроксимации по формулам

$$\varphi(z) = \sum_{k=0}^{m} c_k \cdot \varphi_k(z), \quad c_k = \int_{-1}^{1} \varphi_k(z) \cdot y(z) \cdot w(z) dz.$$

Интегрирование даёт

$$\tilde{n}_0 = 0.486047, \qquad \tilde{n}_1 = -0.503628, \qquad \tilde{n}_2 = 0.166891,$$

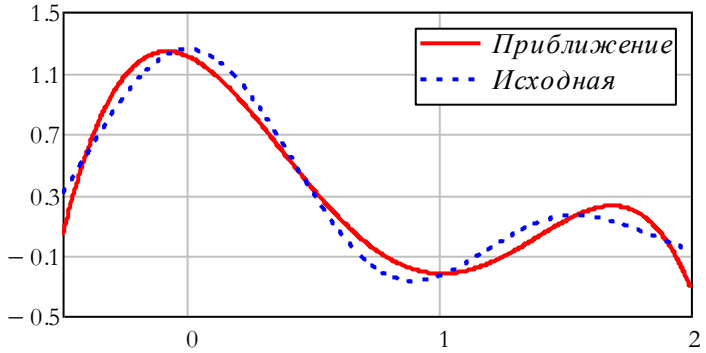
$$\tilde{n}_3 = 0.264177, \qquad \tilde{n}_4 = -0.356619, \qquad \tilde{n}_5 = -0.020116.$$

Теперь совершаем обратное преобразование аргумента и подставляем в аппроксимирующую функцию:

$$\varphi(x) = \sum_{k=0}^{m} c_k \cdot \varphi_k(z(x)), \qquad z(x) = -1 + \frac{2}{b-a}(x-a).$$

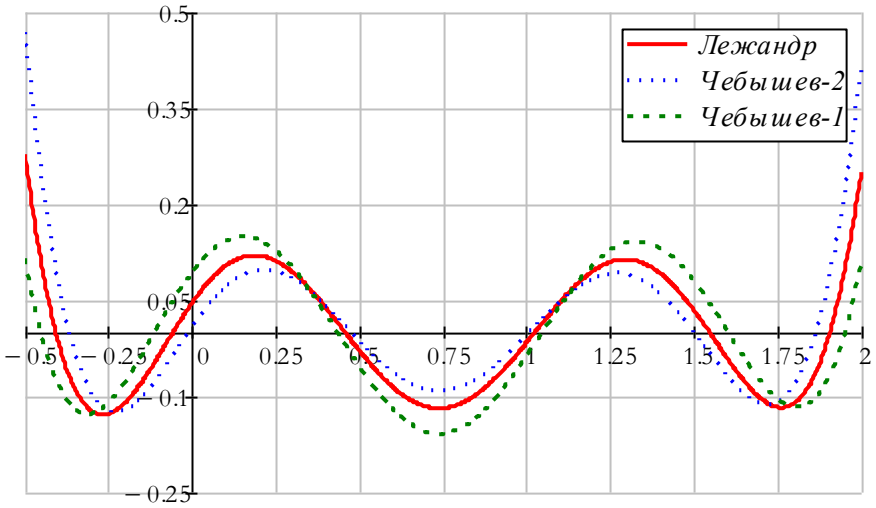
Результат приближения выглядит так:

Приближение базисом Лежандра



По той же схеме получим приближение многочленами Чебышева того же пятого порядка (первый член ряда имеет 0-й порядок). На рисунке показаны графики погрешности аппроксимации разными базисами. Погрешность приближения многочленами Лежандра лежит между двумя другими. Многочлены Чебышёва 1-го рода хуже всех приближают исходную функцию внутри интервала, но лучше всех – на его концах. Многочлены Чебышёва 2-го рода – наоборот, дают самую большую ошибку на концах, но самую маленькую внутри интервала.

Погрешности аппроксимации



Дискретный случай

Чаще всего приходится сталкиваться со среднеквадратичным приближением не аналитически, а таблично заданных функций $\{x_i, y_i\}_n$. Для этого применяют те же идеи и методы с естественной заменой интегрирования суммированием. При этом весовую функцию достаточно задать только в узловых точках таблицы, которая теперь может быть записана как $\{x_i, y_i, w_i\}_n$. Задана (выбрана) система фундаментальных функций – базис аппроксимации $\varphi_1(x), \varphi_2(x), \ldots \varphi_m(x)$ и требуется отыскать наилучший обобщённый многочлен на этом базисе, наилучшим образом приближающий имеющуюся таблицу. Критерий близости – средний квадрат

$$\varepsilon = \sum_{i=1}^{n} \left(y_i - \varphi(x_i) \right)^2 ,$$

который требуется сделать как можно меньше. Такой подход называется *методом наименьших квадратов* (МНК).

Коэффициенты такого многочлена по-прежнему удовлетворяют системе уравнений , только входящие в это уравнение элементы матрицы теперь вычисляются не интегрированием, как в , а суммированием:

$$A_{i,j} = \sum_{k=1}^{n} \varphi_i\left(x_k\right) \cdot \varphi_j\left(x_k\right) \cdot w_k, \quad i, j = 1, 2, \ldots m,$$

$$B_i = \sum_{k=1}^{n} y_k \cdot \varphi_i\left(x_k\right) \cdot w_k, \quad i = 1, 2, \ldots m.$$

Здесь надо обратить внимание на два момента.

Во-первых, для разрешимости системы с коэффициентами требуется невырожденность матрицы A, однако независимость базиса, определённая для непрерывной апппроксимации, теперь недостаточна. Требуется "*линейная независимость на решётке*", которая может быть определена следующим образом. Рассмотрим в n-мерном векторном пространстве R^n вектора $\Phi^i = \left(\varphi_i\left(x_1\right), \varphi_i\left(x_2\right), \ldots \varphi_i\left(x_n\right)\right)$, $i = 1, 2, \ldots m$, каждый из которых представляет собой значения i –й базисной функции на всех точках решётки $\left\{x_i\right\}_n$. векторов этих столько же, сколько базисных функций, m. Наряду с этими векторами рассмотрим ещё вектор $Y = \left(y_1, y_2, \ldots y_n\right)$ всех значений аппроксимируемой функции на той же решётке. Тогда цель аппроксимации может быть сформулирована как минимизация расстояния (в R^n) от вектора Y до линейной комбинации векторов Φ^i:

$$\left\| Y - \sum_{k=1}^{m} c_k \Phi^k \right\| \to \min.$$

При этом вектора Φ^i должны образовывать систему линейно независимых векторов, что и означает "*линейную независимость на решётке*" и обеспечивает невырожденность матрицы A.

Отсюда вытекает, в частности, что ортогональные многочлены на решётке могут и не быть ортогональными.

Во-вторых, у нас теперь есть две размерности – размерность таблицы n и размерность базиса m. Если они равны, то возможна интерполяция таблицы, то есть построение обобщённого многочлена, в точности проходящего через все узлы таблицы $\left\{x_i, y_i\right\}_n$, при этом СКО будет, естественно, равна нулю. Если многочленов больше, чем точек, задача становится недоопределённой – уравнений меньше, чем неизвестных, задача при этом единственного решения не имеет и вообще теряет смысл. Следовательно, должно выполняться условие $n > m$.

Пример

Имеется исходная таблица с тремя верными знаками:

k	1	2	3	4	5	6	7	8	9	10	11
x	-0.5	-0.25	0	0.25	0.5	0.75	1	1.25	1.5	1.75	2
y	0.299	0.949	1.25	0.949	0.299	-0.191	-0.24	$-6.64*10^{-3}$	0.156	0.089	-0.068

z	-1	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4	0.6	0.8	1

Последняя строка этой таблицы содержит преобразованный к стандартному интервалу [-1, 1] аргумент, для которого определены базисы Лежандра и Чебышёва. Для заполнения этой строки достаточно знать только количество точек в таблице.

Возьмём в качестве базиса многочлены Лежандра:

$$\varphi_0(x)=\frac{1}{\sqrt{2}},\ \varphi_1(x)=x\sqrt{\frac{3}{2}},\ \varphi_2(x)=\sqrt{5}\frac{3x^2-1}{2\sqrt{2}},\ \varphi_3(x)=\sqrt{7}\frac{5x^3-3x}{2\sqrt{2}},$$

$$\varphi_4(x)=3\frac{35x^4-30x^2+3}{8\sqrt{2}},\quad \varphi_5(x)=\frac{65x^5-70x^3+15x}{8\sqrt{2}}.$$

Теперь нам нужны значения базисных функций и весовой функции в узловых точках:

k	1	2	3	4	5	6	7	8	9	10	11
φ_0	0.707	0.707	0.707	0.707	0.707	0.707	0.707	0.707	0.707	0.707	0.707
φ_1	-1.225	-0.98	-0.735	-0.49	-0.245	0	0.245	0.49	0.735	0.98	1.225
φ_2	1.581	0.727	0.063	-0.411	-0.696	-0.791	-0.696	-0.411	0.063	0.727	1.581
φ_3	-1.871	-0.15	0.673	0.823	0.524	0	-0.524	-0.823	-0.673	0.15	1.871
φ_4	2.121	-0.494	-0.865	-0.24	0.492	0.795	0.492	-0.24	-0.865	-0.494	2.121
φ_5	-2.345	0.937	0.358	-0.635	-0.721	0	0.721	0.635	-0.358	-0.937	2.345
w	1	1	1	1	1	1	1	1	1	1	1

Теперь по формулам вычислим матрицу A и вектор B:

$$A=\begin{pmatrix} 5.5 & 0 & 1.23 & 0 & 1.997 & 0 \\ 0 & 6.6 & 0 & 2.823 & 0 & 4.358 \\ 1.23 & 0 & 7.997 & 0 & 4.763 & 0 \\ 0 & 2.823 & 0 & 9.856 & 0 & 7.176 \\ 1.997 & 0 & 4.763 & 0 & 12.219 & 0 \\ 0 & 4.358 & 0 & 7.176 & 0 & 14.858 \end{pmatrix}\qquad B=\begin{pmatrix} 2.466 \\ -2.696 \\ 0.932 \\ 0.99 \\ -1.588 \\ -0.659 \end{pmatrix}$$

Заметно, что на решётке базис отнюдь не ортогонален. Решив полученную систему линейных алгебраических уравнений, найдём коэффициенты аппроксимации:

$$c_0=0.508\quad c_1=-0.506\quad c_2=0.215\quad c_3=0.262\quad c_4=-0.297\quad c_5=-0.022$$

Сравним полученные коэффициенты с полученными по непрерывной функции и тому же базису:

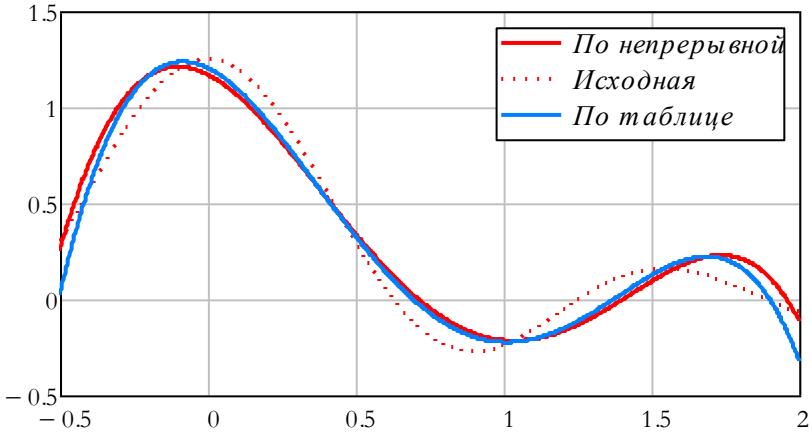
$$c_0=0.486,\quad c_1=-0.503,\ c_2=0.166,\ c_3=0.264,\ c_4=-0.356,\quad c_5=-0.020.$$

Коэффициенты, как видно, заметно отличаются.

Теперь записываем приближённую функцию, используя обратное преобразование аргумента:

$$\varphi(x)=\sum_{k=0}^{5}c_k\cdot\varphi_k\big(z(x)\big),\qquad z(x)=-1+\frac{2}{b-a}(x-a).$$

На графике приведены результаты наших вычислений, для сравнения приведена также аппроксимация, полученная выше по непрерывной функции.



Полученное приближение не совпадает, естественно, с полученным ранее по непрерывной функции, но не так уж сильно.

Порядок выполнения

Упражнение 1

Аппроксимировать заданную функцию многочленами Чебышева 1-го рода. Построить графики исходной и аппроксимированной функций.

Упражнение 2.

Аппроксимировать заданную функцию многочленами Чебышева 2-го рода. Построить графики исходной и аппроксимированной функций.

Упражнение 3.

Аппроксимировать заданную функцию многочленами Лежандра. Построить графики исходной и аппроксимированной функций.

Сравнить результаты аппроксимации разными многочленами.

Варианты заданий

№	Функция	Интервал	№	Функция	Интервал
1	$f1(t) := e^{-2\cdot t}\cdot sin(3\cdot t)$	0-1	7	$f11(t) := ln(t+1)\cdot sin(3\cdot t)$	1-3
2	$f2(t) := sin(2.5t)\cdot cos(t)$	1-2	8	$f12(t) := sin(2.5\cdot t)\cdot cos\left(\frac{1.3}{t+1}\right)$	0-6
3	$f3(t) := e^{-t}\cdot cos(3\cdot t)$	1-3	9	$f13(t) := ln(t+1)\cdot cos(3\cdot t)$	0-3
4	$f4(t) := e^{-t}\cdot sin(4\cdot t)$	0,5-1,5	10	$f14(t) := sin\left(4\cdot t\cdot e^{-t}\right)$	1-10
5	$f5(t) := sin(2\cdot t)\cdot cos(1.3t)$	2-4	11	$f15(t) := sin(2\cdot t)\cdot cos(ln(3\cdot t+.1))$	-1 – 2
6	$f6(t) := e^{-1.3\cdot t}\cdot cos(2\cdot t)$	1-5	12	$f16(t) := e^{-1.3\cdot t}\cdot cos\left(2\cdot t^2\right)$	2-5

Контрольные вопросы

1. Что такое ортогональные многочлены? В чём их достоинства?

2. Каков смысл весовой функции?
3. Как выполнять аппроксимацию на нестандартном интервале?

6 Полиномиальные формулы дифференцирования, порядок точности. Формулы Рунге.

Теоретические сведения

Пусть имеется таблично заданная функция $\{x_i, y_i\}_n$ и требуется вычислить её производную в точке *x*. Заменим исходную функцию интерполяционным многочленом Ньютона, дополнительно обозначив

$$\xi_i = x - x_i, \quad i = 1, 2, \ldots$$

для более компактной записи довольно длинных выражений

$$y(x) \approx \varphi(x) = y_1 + \xi_1 \nabla^1_{12} + \xi_1 \xi_2 \nabla^2_{123} + \ldots = y_1 + \sum_{k=1}^{n-1} \left(\nabla^k_{12,\ldots k+1} \prod_{i=1}^{k} \xi_i \right).$$

Продифференцируем по икс, имея в виду, что все разделённые разности не зависят от икса, а $\frac{d\xi_i}{dx} = 1$.

$$\varphi'(x) = \nabla^1_{12} + (\xi_1 + \xi_2) \nabla^2_{123} + (\xi_1 \cdot \xi_2 + \xi_1 \cdot \xi_3 + \xi_2 \cdot \xi_3) \nabla^3_{1234} + \ldots$$

Можно вычислить и вторую производную:

$$\varphi''(x) = 2\nabla^2_{123} + 2(\xi_1 + \xi_2 + \xi_3) \nabla^3_{1234} + \ldots$$

Можно вообще получить общую формулу *m*-той производной:

$$\varphi^{(m)}(x) = m! \cdot \nabla^m_{1,2,..m+1} + m! \cdot \left[\nabla^{m+1}_{12\ldots m+2} \cdot \sum_{i=1}^{m+1} \xi_i + \nabla^{m+2}_{12\ldots m+3} \cdot \sum_{j>i\geq 1}^{m+2} \xi_i \xi_j + \ldots \right].$$

Из формулы можно заметить, что для оценки *m*-той производной требуется как минимум $m+1$ точка. При этом оценка производной не зависит от икса и пропорциональна разделённой разности того же порядка:

$$\varphi^{(m)}(x) \approx m! \cdot \nabla^m_{1,2,..m+1}.$$

Ранее было сказано, что разделённая разность в аналогична производной. Теперь эта аналогия получила некоторое подтверждение.

Рассмотрим некоторые частные случаи формулы . Дополнительно назовём разность меду количеством точек и порядком производной *порядком* формулы численного дифференцирования и обозначим эту величину

$$p = n - m.$$

Первая производная по формуле первого порядка:

$$\varphi_1^{(1)}(x) = \frac{y_1 - y_2}{x_1 - x_2}.$$

В этой формуле справа отсутствует точка, в которой вычисляется производная, но всё-таки логично предполагать, что эта точка должна располагаться между x_1 и x_2, кроме того, расстояние между этими точками – величина малая. Эта формула рассматривается как приближённая оценка первой производной функции $y(x)$ в точке x:

$$y'(x) = \varphi_2^{(1)}(x) + R_1^1,$$

где R_1^1 – погрешность формулы первой производной (верхний индекс) первого порядка (нижний индекс). Оценим величину этой погрешности. Для этого, предполагая, что функция $y(x)$ достаточно гладкая, то есть имеет достаточное количество непрерывных производных во всех точках, разложим её в ряд Тэйлора в окрестности точки x:

$$y(x+\xi) = y(x) + \xi \cdot y'(x) + \frac{\xi^2}{2!} y''(x) + o(\xi^2),$$

где $o(\xi^2)$ – малая более высокого порядка, чем ξ^2. Представим в виде точки y_1, y_2,

$$y_1 = y(x_1) = y(x - \xi_1) = y(x) - \xi_1 y'(x) + \frac{\xi_1^2}{2} y''(x) + o(\xi_1^2),$$

$$y_2 = y(x) - \xi_2 y'(x) + \frac{\xi_2^2}{2} y''(x) + o(\xi_2^2).$$

Теперь подставим в .

$$\varphi_1^{(1)}(x) = \frac{y_1 - y_2}{x_1 - x_2} =$$

$$= \frac{y(x) - \xi_1 y'(x) + \frac{\xi_1^2}{2} y''(x) + o(\xi_1^2) - \left(y(x) - \xi_2 y'(x) + \frac{\xi_2^2}{2} y''(x) + o(\xi_2^2) \right)}{x_1 - x_2} =$$

$$= \frac{y'(x) \cdot (\xi_2 - \xi_1) + y''(x) \cdot \frac{\xi_1^2 - \xi_2^2}{2} + o(\xi^2)}{(\xi_2 - \xi_1)} = y'(x) - \frac{\xi_1 + \xi_2}{2} y''(x) + o(\xi).$$

Мы получили оценку погрешности, в которой малыми более высокого порядка можно пренебречь:

$$R_1^1 = \frac{\xi_1 + \xi_2}{2} y''(x).$$

Лучше использовать предельное, самое большое из возможных, абсолютное значение этой погрешности:

$$\left| R_1^1 \right| \le \frac{h}{2} M_2,$$

где

$$M_2 = \max_x \left| y''(x) \right|, \quad h = \left| x_2 - x_1 \right|.$$

Введём достаточно общее определение:

пусть для какой-то величины ψ имеется приближённая формула φ, которая зависит от (малого) параметра h и имеет вид

$$\psi = \varphi(h) + K \cdot h^p + o\left(h^p\right),$$

где K – некоторый множитель, не зависящий от h, тогда говорят, что приближение имеет *порядок точности p*, а выражение $K \cdot h^p$ называют (главным) *остаточным членом* или *остатком* приближения.

По этой терминологии формула численного дифференцирования имеет первый порядок точности, причём остаток оценивается формулой . Заметим, что порядок точности совпал с порядком формулы $n - m$. То же справедливо для всех полиномиальных формул численного дифференцирования. Оценка остаточного члена в общем виде такова:

$$\left| R_p^m \right| \le \frac{M_n}{n!} \max_i \left| \xi_i \right|^p, \quad M_n = \max_x \left| y^{(n)}(x) \right|,$$

то есть погрешность формулы *m*-той производной по *n* – точечной таблице имеет порядок точности $p = n - m$, причём погрешность пропорциональна *n*-той производной исходной функции.

Ранее приводилась общая формула погрешности приближённой формулы. Если известно, что формула имеет порядок точности *p*, то можно записать

$$f = \varphi(h) + K \cdot h^p + o\left(h^p\right)$$

где:

f – точное значение, нам не известное, конечно,

$\varphi(h)$ – приближённое значение, вычисленное при параметре (например, шаге таблицы) h, Kh^p – главная часть погрешности (остаточный член),

K – некоторый конечный (не малый) множитель,

$o\left(h^p\right)$ – малая более высокого порядка малости, чем h^p.

Априорные оценки погрешности используют максимальное по модулю возможное значение множителя *K*, но можно попытаться оценить его реальное, с учётом знака, значение. Пусть мы выполнили вычисление по формуле с каким-то конкретным значением *h*. Повторим вычисление с другим значением – $r \cdot h$, где *r* – некоторый конечный множитель. Формула будет верна и для этого вычисления:

$$f = \varphi(r \cdot h) + K \cdot (r \cdot h)^p + o\left((r \cdot h)^p\right).$$

Но, как известно из матанализа, умножение на конечный множитель не меняет порядка малой величины, поэтому можно записать:

$$f = \varphi(r \cdot h) + K \cdot r^p \cdot (h)^p + o\left(h^p\right).$$

Вычтем из последней формулы и перегруппируем слагаемые:

$$\varphi(h)-\varphi(r\cdot h)=K\cdot\left(r^p-1\right)\cdot h^p+o\left(h^p\right),$$

откуда получаем оценку главной части погрешности:

$$K\cdot h^p=\frac{\varphi(h)-\varphi(r\cdot h)}{r^p-1}+o\left(h^p\right).$$

Формула позволяет апостериорно получить главную часть погрешности обоих результатов, как полученного с шагом h, так и с изменённым шагом $r\cdot h$. Эта формула известна как *первая формула Рунге*.

Необходимо напомнить, что хотя для применения этой формулы и не требуется знание значений старших производных, их существование, однако, требуется, иначе формула просто не будет верна.

Пример

Пусть функция задана $y(x)=\ln(x)$ таблицей

i	1	2	3
x	1	1.1	1,2
y	0	0.09531	0.18232

и требуется вычислить её первую производную в точке $x=1$. Вообще-то это значение легко найти аналитически, оно равно 1, но нам интересно посмотреть, как с этой задачей справятся численные методы. Воспользуемся формулой первого порядка и обозначим

$$\varphi(h)=\frac{y_2-y_1}{h}=0.9531,\quad \varphi(2\cdot h)=\frac{y_3-y_1}{2\cdot h}=0.9116,\quad h=x_2-x_1=0.1$$

Порядок этих приближённых формул равен единице, а множитель $r=2$. Подставим полученные значения в формулу и получим главную часть погрешности:

$$\frac{\varphi(h)-\varphi(r\cdot h)}{r^p-1}=\frac{0.9531-0.9116}{2^1-1}=0.0415$$

Итак, главная часть погрешности результата при шаге h равна 0.0415. Точное значение погрешности $1-0.9531=0.0469$. Эту оценку погрешности можно использовать для уточнения результата, что даёт

$$0.9531+0.0415=0.9946,$$

гораздо ближе к точному значению, ошибка уже составляет всего 0.0054 то снизилась практически на порядок.

Примем теперь за исходное значение, поученное с удвоенным шагом, тогда множитель r будет равен $\frac{1}{2}$, и применим ту же технику:

$$\frac{\varphi(h)-\varphi(r\cdot h)}{r^p-1}=\frac{0.9116-0.9531}{\left(\frac{1}{2}\right)^1-1}=0.0830$$

Используя апостериорно полученную погрешность для исправления приближённого результата, получаем то же самое исправленное значение

$$0.9116 + 0.0830 = 0.9946,$$

то есть, если у нас есть два приближения полученные по разным решёткам, то всё равно, какое из них брать за основное при исправлении результата по формуле Рунге, ответ будет тот же самый. Впрочем, это можно показать и чисто аналитически, не прибегая к числовому примеру.

1.1 Повышение порядка приближённой формулы. Вторая формула Рунге.

Первая формула Рунге может использоваться для апостериорной оценки погрешности, но и для получения новых более точных приближённых формул. Для этого достаточно к исходной формуле прибавить поправку, которую даёт Первая формула Рунге. Пусть имеется приближённая формула $\varphi(h)$ (неважно, что она вычисляет), порядок точности p которой известен. Добавим в ней поправку в соответствии с и получим новую, более точную приближённую формулу $\tilde{\varphi}(h)$:

$$\tilde{\varphi}(h) = \varphi(h) + \frac{\varphi(h) - \varphi(r \cdot h)}{r^p - 1} = \frac{r^p \cdot \varphi(h) - \varphi(r \cdot h)}{r^p - 1},$$

которая известна как *вторая формула Рунге*. Порядок точности полученной формулы выше порядка исходной формулы по крайней мере на единицу, а может быть и больше.

Применим этот подход к формуле первой производной, которую мы использовали в примере 10.1.

$$\varphi(x_1, h) = \frac{y_2 - y_1}{h}, \quad \varphi(x_1, 2 \cdot h) = \frac{y_3 - y_1}{2 \cdot h}, \quad h = x_2 - x_1 = x_3 - x_2, \quad p = 1, \quad r = 2,$$

$$\tilde{\varphi}(x_1, h) = \frac{r^p \varphi(x_1, h) - \varphi(x_1, r \cdot h)}{r^p - 1} = \frac{2\dfrac{y_2 - y_1}{h} - \dfrac{y_3 - y_1}{2 \cdot h}}{2 - 1} = \frac{-y_3 + 4y_2 - 3y_1}{2 \cdot h}.$$

Итак, получена ещё одна формула численного дифференцирования:

$$y'(x_1) \approx \frac{-y_3 + 4y_2 - 3y_1}{2h}.$$

Порядок точности этой формулы обещает быть не ниже второго. Проверим это, используя разложение в ряд Тэйлора. Дополнительно обозначим для более компактной записи:

$$y_k^i = \left.\frac{dy^i}{dx}\right|_{x_k}.$$

Выразим узловые значения исходной функции в виде ряда Тэйлора

$$y_2 = y_1 + hy_1^1 + \frac{h^2}{2} y_1^2 + \frac{h^3}{6} y_1^3 + o\left(h^3\right),$$

$$y_3 = y_1 + 2hy_1^1 + \frac{4h^2}{2} y_1^2 + \frac{8h^3}{6} y_1^3 + o\left(h^3\right),$$

и подставим их в формулу :

$$\frac{y_3 + 4y_2 - 5y_1}{2h} = \frac{1}{2h}\left(2hy_1^1 - \frac{4}{6}h^3 y_1^3 + o\left(h^3\right)\right) = y_1^1 - \frac{h^2}{3} y_1^3 + o\left(h^2\right).$$

Но y_1^1 – это как раз то, что мы ищем, точное значение первой производной в точке x_1, а всё остальное – остаток, погрешность. Остаток, точнее его главная часть, имеет второй порядок малости, как и ожидалось, и пропорционален он, как и обещает общая формула, третьей производной исходной функции.

Получим теперь аналогичную формулу традиционным путём, дифференцируя интерполяционный многочлен Ньютона. В предыдущем разделе была получена формула первой производной по трём точкам:

$$\varphi'(x) = \nabla_{12}^1 + \left(\xi_1 + \xi_2\right)\nabla_{123}^2.$$

Подставим в неё x_1 в качестве x, тогда $\xi_1 = x - x_1 = 0, \quad \xi_2 = x - x_2 = -h,$

$$\nabla_{12}^1 = \frac{y_2 - y_1}{h}, \quad \nabla_{123}^2 = \frac{\frac{y_3 - y_2}{h} - \frac{y_2 - y_1}{h}}{2h},$$

$$\varphi'\left(x_1\right) = \frac{y_2 - y_1}{h} - h\frac{\frac{y_3 - y_2}{h} - \frac{y_2 - y_1}{h}}{2h} = \frac{4y_2 - 3y_1 - y_3}{2h}.$$

Получилась та же самая формула.

Итак, используя вторую формулу Рунге, мы получили формулу численного дифференцирования второго порядка точности, причём сделали это гораздо более простым путём, чем раньше.

Подведём итог.

Дважды вычислив одну и ту же приближённую формулу, порядок точности которой нам известен, по разным решёткам (имеющим разный шаг), мы можем, используя первую формулу Рунге, апостериорно оценить погрешность (точнее сказать, её главную часть) для обоих вычислений и использовать эту оценку для уточнения результата.

Имея приближённую формулу известного порядка точности, можно, используя вторую формулу Рунге, получить приближённую формулу более высокого порядка точности.

Задание

Вычислите численно значение производной своей функции по полиномиальной формуле 2-го порядка, оцените погрешность по 1-й формуле Рунге. Сравните с результатом, даваемым системой Mathcad в численном и аналитическом варианте. Сделайте выводы.

Варианты функции.

№	F_0
1	$a \cdot e^{-b \cdot t} \cdot sin(c \cdot t)$
2	$a \cdot sin(b \cdot t) \cdot cos(c \cdot t)$
3	$a \cdot e^{-b \cdot t} \cdot cos(c \cdot t)$
4	$a \cdot e^{-b \cdot t} \cdot sin(c \cdot t)$
5	$a \cdot sin(b \cdot t) \cdot cos(c \cdot t)$
6	$a \cdot e^{-b \cdot t} \cdot cos(c \cdot t)$
7	$a \cdot ln(t + b) \cdot sin(c \cdot t)$
8	$sin(a \cdot t) \cdot cos\left(\frac{b}{t + c}\right)$
9	$a \cdot ln(t + b) \cdot cos(c \cdot t)$
10	$a \cdot sin\left(b \cdot t \cdot e^{-c \cdot t}\right)$
11	$sin(a \cdot t) \cdot cos(ln(b \cdot t + c))$
12	$a \cdot e^{-b \cdot t} \cdot cos\left(c \cdot t^2\right)$

7 Численное интегрирование. Квадратурные формулы.

Пусть требуется найти определенный интеграл

$$F = \int_a^b y(x)dx$$

причем функция $y(x)$ непрерывна на отрезке [a,b]. Аналитическое выражение для интеграла удается получить крайне редко. Кроме того, зачастую неизвестно и аналитическое выражение для функции $y(x)$, которая может быть задана таблично. В таких случаях интеграл находят численно. Обычно $y(x)$ заменяют аппроксимирующей функцией $\varphi(x,\alpha)$, где $\alpha = (\alpha_0, \alpha_1, \alpha_2, ...\alpha_m)$– вектор параметров аппроксимации. Аппроксимирующая функция, естественно, должна легко интегрироваться аналитически. Чаще всего применяют аппроксимирующие обобщенные многочлены вида

$$y(x) = \sum_{i=0}^{m} a_i \cdot \varphi_i(x) + r(x),$$

где $r(x)$– остаточный член аппроксимации. Проинтегрировав последнюю формулу, получаем

$$F = \sum_{i=0}^{m} \alpha_i \cdot c_i + R, \quad c_i = \int_a^b \varphi_i(x)dx, \quad R = \int_a^b r(x)dx,$$

где R – остаточный член (погрешность) интеграла.

1.2 Квадратурные формулы.

В качестве аппроксимирующей функции очень удобно использовать интерполяционный многочлен Ньютона в его лагранжевой форме,

$$\varphi_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{(x - x_i)}{(x_k - x_i)},$$

когда коэффициентами (обобщённого) многочлена служат узловые значения исходной функции:

$$y(x) = \sum_{i=0}^{m} y_i \cdot \varphi_i(x) + r(x), \quad y_i = y(x_i).$$

Тогда превращается в

$$F = \sum_{i=0}^{m} y_i \cdot c_i + R, \quad c_i = \int_a^b \varphi_i(x)\,dx.$$

Формулы типа , в которых интеграл заменяется взвешенной суммой значений функции в некоторых точках, называются *квадратурными формулами*, или *квадратурами*. Весовые коэффициенты квадратурной формулы не зависят от интегрируемой функции, они целиком определяются табличными (узловыми) значениями аргумента.

Простейшей квадратурной формулой является известная из матанализа интегральная сумма Коши-Римана. Однако она несколько отличается от классической квадратуры, основанной на интерполяционной формуле. Простейшей классической квадратурой является *формула трапеций*. В матанализе её обосновывали на полуинтуитивном уровне, апеллируя к геометрической интерпретации интеграла через площадь (см. Рисунок 11.1).

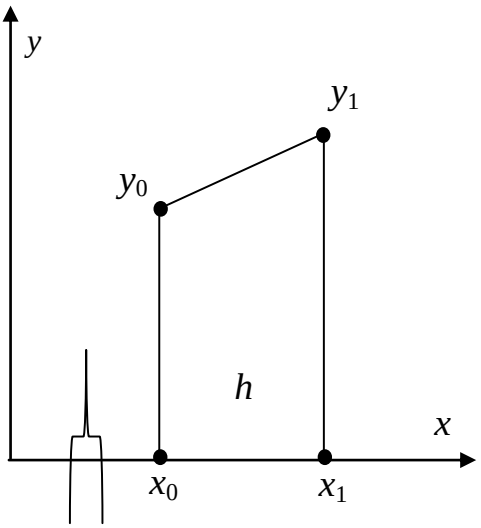


Рисунок 7.1. К формуле трапеций.

Выполним формально процедуру интерполяции по базису .

$$\varphi_0(x) = -\frac{x - x_1}{x_0 - x_1}, \quad \varphi_1(x) = \frac{x - x_0}{x_1 - x_0}, \quad h = x_1 - x_0,$$

$$\varphi(x) = -y_0 \frac{x - x_1}{h} + y_1 \frac{x - x_0}{h}.$$

Проинтегрировав базисные функции, получаем коэффициенты квадратуры . Снова, как и прежде, используем обозначения

$$\xi_i = x - x_i,$$

тогда

$$c_0 = \int_a^b \varphi_0(x)\,dx = -\frac{1}{h}\int_{-h}^{0} \xi_1 d\xi_1 = \frac{1}{2}h,$$

$$c_1 = \int_a^b \varphi_1(x)\,dx = \frac{1}{h}\int_{0}^{h} \xi_0 d\xi_0 = \frac{1}{2}h.$$

откуда и получаем классическую формулу трапеций

$$F = \frac{h}{2}(y_0 + y_1) + R.$$

Оценим остаточный член этой формулы, для чего прибегнем к разложению в ряд Тэйлора исходной функции в окрестности точки x_0. Для краткости будем обозначать

$$\xi = x - x_0, \qquad y_0^i = \frac{d^i}{dx^i} y(x_0).$$

$$y(x) = y_0 + \xi y_0^1 + \frac{\xi^2}{2} y_0^2 + \frac{\xi^3}{6} y_0^3 + o(\xi^3).$$

Проинтегрируем и получим точное значение интеграла:

$$F = h \cdot y_0 + \frac{h^2}{2} y_0^1 + \frac{h^3}{6} y_0^2 + \frac{h^4}{24} y_0^3 + o(h^4).$$

Теперь подставим в численную формулу

$$F = \frac{h}{2}(y_0 + y_1) + R = \frac{h}{2}\left(2y_0 + hy_0^1 + \frac{h^2}{2} y_0^2 + \frac{h^3}{6} y_0^3 + o(h^3)\right) + R$$

и вычтем из . Получаем

$$R = -\frac{h^3}{12} y_0^2 + o(h^3).$$

Получается, что формула трапеций имеет третий порядок точности. Ещё интересно отметить, что остаточный член, как и у формул численного дифференцирования, пропорционален старшей производной, только у формул численного дифференцирования порядок производной в формуле остаточного члена выше порядка точности, а в формуле трапеций – ниже.

Формула трапеций в том виде, как мы её получили, для практического применения не годится, поскольку в ней шаг решётки совпадает с интервалом интегрирования и не является и не может быть сделан малым. Значит точности она не обеспечит. Справиться с этой бедой можно, в принципе, двумя путями – повысить порядок аппроксимации, для чего потребуется использовать дополнительные точки внутри интервала интегрирования, или воспользоваться свойством аддитивности интеграла, разбить большой интервал на маленькие кусочки, вычислить интегралы по каждому маленькому интервалу, а потом все их суммировать, для чего также понадобятся дополнительные точки.

Рассмотрим второй путь.

1.3 Обобщённая формула трапеций и её порядок точности. Накопление ошибок.

Пусть для вычисления интеграла используется таблично заданная функция (x_i, y_i), $i = 0,1,2...n$. Будем считать, что точки в ней пронумерованы по возрастанию аргумента, а крайние точки совпадают с концами интервала интегрирования, $x_0 = a$, $x_n = b$. Будем считать для простоты и удобства, что таблица равномерная с шагом $h = x_i - x_{i-1}$. Перепишем в виде

$$F = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} y(x)\,dx + R$$

и заменим каждый интеграл в получившейся сумме интегралов формулой трапеций:

$$F = \sum_{i=1}^{n} \left(\frac{h}{2}(y_{i-1} + y_i) + R_i \right) = \frac{h}{2}(y_0 + 2y_1 + 2y_2 + ... + 2y_{n-1} + y_n) + \sum_{i=1}^{n} R_n.$$

Эта формула известна как *обобщённая формула трапеций*. Оценим её остаточный член, используя формулу :

$$R = \sum_{i=1}^{n} R_i = -\frac{h^2}{12} \sum_{i=1}^{n} h \cdot y_{i-1}^2 + o(h^3).$$

Сумма в последней формуле представляет собой интегральную сумму Коши-Римана для интеграла второй производной,

$$\sum_{i-1}^{n} h \cdot y_i^2 = \int_a^b y''(x)\,dx + o(h),$$

который по известной формуле равен приращению первой производной на интервале умноженному на длину интервала:

$$\int_a^b y''(x)\,dx \approx y'(b) - y'(a)$$

которую, с использованием теоремы Лагранжа, можно переписать в виде

$$y'(b) - y'(a) = (b-a)\,y'(x^*), \quad x^* \in [a,b],$$

где x^* – некоторая (какая-то, никто не знает какая именно) внутренняя точка интервала интегрирования. Окончательно остаток можно записать в виде

$$-\frac{h^2}{12}(b-a)y''(x^*).$$

Последняя формула позволяет сделать априорную оценку предельной погрешности в виде:

$$|R| \le h^2 \frac{b-a}{12} M_2, \quad M_2 = \max_{x\in[a,b]} |y''(x)|.$$

Обобщённая формула трапеций таким образом имеет уже только второй порядок точности. Здесь мы столкнулись с явлением, называемым *накоплением ошибки*. Суть его в том, что при многократном применении приближённой формулы погрешности накапливаются и сумма имеет порядок точности ниже, чем каждое из слагаемых.

Надо заметить, что оценка погрешности обобщённой формулы трапеций справедлива только для достаточно гладких подынтегральных функций, имеющих непрерывную вторую производную.

Итак, обобщённая формула трапеций имеет второй порядок точности, и погрешность её пропорциональна длине интервала интегрирования и второй производной подынтегральной функции. Формула обещает получение сколь угодно точного результата при достаточно частой сетке (малый шаг) и абсолютно точных данных и вычислениях.

Исследуя точность численного дифференцирования, мы установили, что использование слишком частой сетки приводит к возрастанию ошибок округления. Посмотрим, как обстоит дело с интегрированием. Пусть таблица значений функции имеет в каждом узле абсолютную погрешность δ. Тогда погрешность вычисления обобщённой формулы трапеций будет оцениваться как

$$\Delta \le n \cdot h \cdot \delta = (b-a)\cdot\delta,$$

то есть предельная ошибка округлений не зависит от величины шага, а только от длины интервала и точности данных. Полная же ошибка формулы, складывающаяся из остаточного члена и погрешностей округления, тем меньше, чем чаще сетка. Однако невыгодно делать сетку слишком мелкой, поскольку это всё равно не повысит точность результата (ошибки округления), а только увеличит трудоёмкость вычислений. Разумным представляется выбирать шаг таким, чтобы остаточный член был того же порядка, что и погрешности округлений,

$$R \approx \Delta, \quad h^2 \frac{b-a}{12} M_2 \approx (b-a)\cdot\delta,$$

откуда и получаем примерную величину разумного шага интегрирования

$$h \approx \sqrt{\frac{12\delta}{M_2}},$$

которая обеспечивает точность результата не хуже, чем

$$\Delta + R \approx 2(b-a)\delta.$$

Итак, точность формулы трапеций целиком определяется точностью данных и вычислений и длиной интервала интегрирования. Следует напомнить, что оценки , и являются *априорными предельными оценками ошибок*, реальные ошибки могут быть существенно меньше.

1.4 Формула Симпсона.

Выше было сказано, что для повышения точности численного вычисления интеграла есть два пути – разбиение интервала интегрирования на мелкие интервалы и повышение порядка аппроксимации подынтегральной функции. Первый путь привёл рас к формуле трапеций, пришло время испытать второй путь.

Пусть теперь мы будем использовать три точки из интервала интегрирования – концы и середину,

$$x_0 = a, \quad x_1 = \frac{a+b}{2}, \; x_2 = b,$$

обозначим ещё

$$x_0, \; x_1 = x_0 + h, \; x_2 = x_1 + h$$

$$y_0 = y(x_0), \; y_1 = y(x_1), \; y_2 = y(x_2).$$

На этой сетке построим интерполяционный многочлен Ньютона в форме Лагранжа и проинтегрируем его. То есть пройдём тот же путь, что и при выводе формулы трапеций. Будем сразу использовать обозначения , введённые выше.

$$\varphi_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{\xi_1\xi_2}{2h^2},$$

$$\varphi_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = -\frac{\xi_0\xi_2}{h^2},$$

$$\varphi_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{\xi_0\xi_1}{2h^2},$$

$$\varphi(x) = y_0\frac{\xi_1\xi_2}{2h^2} - y_1\frac{\xi_0\xi_2}{h^2} + y_2\frac{\xi_0\xi_1}{2h^2}.$$

Интегрируя , получаем

$$c_0 = \int_a^b \varphi_0(x)dx = \frac{1}{2h^2}\int_a^b \xi_1\xi_2 dx = \frac{1}{3}h,$$

$$c_1 = \int_a^b \varphi_1(x)dx = -\frac{1}{h^2}\int_a^b \xi_0\xi_2 d\xi_0 = \frac{4}{3}h,$$

$$c_1 = \int_a^b \varphi_1(x)dx = \frac{1}{2h^2}\int_a^b \xi_0\xi_1 d\xi_0 = \frac{1}{3}h.$$

Подставляя полученные выражения в , получаем квадратурную формулу

$$F = \frac{h}{3}(y_0 + 4y_1 + y_2) + R.$$

Проанализируем остаточный член полученной формулы тем же путём, что и для формулы трапеций. Разложение в ряд Тэйлора удобно сделать в окрестности точки x_1. Будем обозначать $\xi = x - x_1$, тогда

$$y(x)=y_1+\xi y_1^1+\frac{\xi^2}{2}y_1^2+\frac{\xi^3}{6}y_1^3+\frac{\xi^4}{24}y_1^4+\frac{\xi^5}{120}y_1^5+o\left(\xi^5\right),$$

$$y_0=y_1-h\cdot y_1^1+\frac{h^2}{2}y_1^2-\frac{h^3}{6}y_1^3+\frac{h^4}{24}y_1^4-\frac{h^5}{120}y_1^5+o\left(h^5\right),$$

$$y_2=y_1+h\cdot y_1^1+\frac{h^2}{2}y_1^2+\frac{h^3}{6}y_1^3+\frac{h^4}{24}y_1^4+\frac{h^5}{120}y_1^5+o\left(h^5\right).$$

Интегрируя разложенную в ряд Тэйлора функцию, заметим, что интегралы слагаемых с нечётными степенями ξ будут равны нулю в силу симметричности интервала для ($\xi\in\left[-h,h\right]$).

$$F=\int_{-h}^{h}y(\xi)d\xi=\frac{h}{3}\left(6y_1+h^2y_1^2+\frac{h^4}{20}y_1^4\right)+o\left(h^6\right).$$

После подстановки в получаем

$$F=\frac{h}{3}\left(6y_1+h^2y_1^2+\frac{h^4}{12}y_1^4\right)+R+o\left(h^6\right).$$

Вычитая из , получаем оценку остаточного члена:

$$R=-\frac{h^5}{90}y_1^4+o\left(h^6\right),$$

Таким образом, формула имеет пятый порядок точности.

Поскольку шаг таблицы не является малой величиной, снова используем аддитивность интеграла. Только теперь надо позаботиться, чтобы в исходной таблице $\left(x_i,y_i\right)$, $i=0,1,2...n$ количество интервалов было чётным, тогда мы разобьем все интервалы на пары, для каждой пары применим формулу и суммируем результаты. Для удобства введём ещё обозначения:

$$S_1=\sum_{i=1}^{\frac{n}{2}}y_{2i-1}=y_1+y_3+...+y_{n-1},$$

$$S_2=\sum_{i=1}^{\frac{n}{2}-1}y_{2i}=y_2+y_4+...y_{n-2}.$$

Здесь S_1 – сумма всех узловых значений функции с нечётными номерами, а S_2 –сумма значений с чётными номерами, исключая крайние, первое и последнее. Тогда окончательно получаем квадратуру

$$F=\frac{h}{3}\left(y_0+4S_1+2S_2+y_n\right)-\frac{h^5}{90}\sum_{i=1}^{\frac{n}{2}}y_{2i-1}^4+o\left(h^6\right),$$,

известную как *формула Симпсона*.

Рассмотрев главную часть остаточного члена в как интегральную сумму, можно записать главную часть погрешности формулы Симпсона как

$$R=\frac{h^4}{180}\sum_{i=1}^{\frac{n}{2}-1}2h\cdot y_{2i-1}^4+o\left(h^6\right)\approx-\frac{h^4}{180}\int_a^b y^{(4)}(x)dx\approx-\frac{h^4}{180}\left(y^{(3)}(b)-y^{(3)}(a)\right),$$

откуда следует её предельная оценка

$$|R|\le\frac{h^4(b-a)}{180}M_4,\quad M_4=\max_{x\in[a,b]}\left|\frac{d^4}{dx^4}y(x)\right|$$

Здесь, как и раньше, речь идёт об априорной оценке предельной ошибки интегрирования.

1.5 Применение формул Рунге

Пусть количество интервалов чётное, шаг равномерной таблицы равен h. Используя обозначения , запишем для такой таблицы формулу трапеций:

$$F(h)=\frac{h}{2}\left(y_0+2S_1+2S_2+y_n\right)+R_h.$$

Повторим вычисления по таблице с удвоенным шагом, тогда в формулу попадут только значения с чётными номерами:

$$F(2h)=h\left(y_0+2S_2+y_n\right)+R_{2h}.$$

Применим теперь первую формулу Рунге, учитывая, что формула трапеций имеет второй порядок точности, $p=2,\ r=2$.

$$R_h\approx\frac{F(h)-F(2h)}{2^2-1}=\frac{h}{6}\left(-y_0+2S_1-2S_2-y_n\right).$$

Мы получили хорошую апостериорную оценку остаточного члена формулы трапеций, причём для её вычисления практически не требуется дополнительных вычислений.

Подправим формулу трапеций, добавив в неё вычисленный остаток:

$$F(h)=\frac{h}{2}\left(y_0+2S_1+2S_2+y_n\right)+\frac{h}{6}\left(-y_0+2S_1-2S_2-y_n\right)=$$

$$=\frac{h}{3}\left(y_0+4S_1+2S_2+y_n\right).$$

Как видите, получилась формула Симпсона. Причём получение её оказалось много проще, чем классическим путём. Естественно, что ту же формулу можно было получить, применив вторую формулу Рунге.

Задание

Вычислите определённый интеграл заданной функции на интервале [-1,1] по 10, 20, 30 точкам по обобщённой формуле трапеций. Оцените точность по первой формуле Рунге. Сравните результат с полученным штатными средствами Mathcad. Сделайте выводы.

Варианты функции.

№	F_0
1	$a\cdot e^{-b\cdot t}\cdot sin(c\cdot t)$
2	$a\cdot sin(b\cdot t)\cdot cos(c\cdot t)$

3	$a\cdot e^{-b\cdot t}\cdot cos(c\cdot t)$
4	$a\cdot e^{-b\cdot t}\cdot sin(c\cdot t)$
5	$a\cdot sin(b\cdot t)\cdot cos(c\cdot t)$
6	$a\cdot e^{-b\cdot t}\cdot cos(c\cdot t)$
7	$a\cdot ln(t+b)\cdot sin(c\cdot t)$
8	$sin(a\cdot t)\cdot cos\left(\frac{b}{t+c}\right)$
9	$a\cdot ln(t+b)\cdot cos(c\cdot t)$
10	$a\cdot sin\left(b\cdot t\cdot e^{-c\cdot t}\right)$
11	$sin(a\cdot t)\cdot cos(ln(b\cdot t+c))$
12	$a\cdot e^{-b\cdot t}\cdot cos\left(c\cdot t^2\right)$

8 Решение ОДУ методом Рунге-Кутты. Интегрирование с постоянным шагом.

Использование решающего блока

Mathcad содержит несколько различных инструментов для решения обыкновенных дифференциальных уравнений (ОДУ) или систем ОДУ. Функция ***Odesolve*** используется в составе решающего блока, в котором описана решаемая система в произвольной форме, использующей обычную математическую нотацию. Кроме того имеется ещё несколько функций, которые могут использоваться самостоятельно, без решающего блока и требуют специальной записи решаемой системы.

Дифференциальное уравнение записывается внутри решающего блока в традиционной форме. Рассмотрим колебательное звено с параметрами

$T = 0.1$, $\xi = 0.2$, $K = 1$, $f(t)=1(t)$,

и входным сигналом $f(t) = sin(t)$. Выходной сигнал такого звена описывается ОДУ

$$T^2\cdot x''(t)+2\cdot\xi\cdot T\cdot x'(t)+x(t)=K\cdot f(t)$$

Решение этого уравнения можно получить с помощью такого решающего блока:

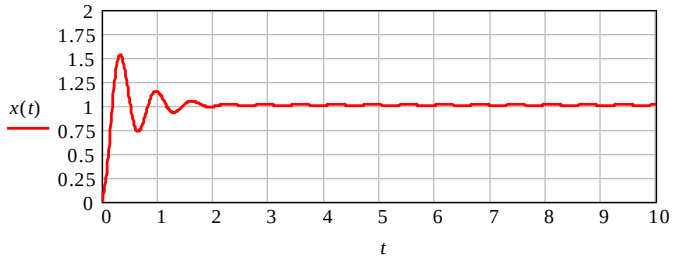
$given$

$T^2\cdot x''(t) + 2\cdot\xi\cdot T\cdot x'(t) + x(t) = K\cdot f(t)$

$x(0) = 1$ $\quad$ $x'(0) = 0$

$x := odesolve(t, 10)$

Внутри блока должно использоваться булевское равенство, ***ctrl***+"=". Для того, чтобы изобразить символ производной, штрих, надо нажать ***ctrl***+F7. Аргументами функции *Odesolve* в нашем случае являются имя свободной переменной (у нас это время *t*), и правая граница интервала, на котором решается уравнение (10). Вычисленным результатом, *x*, будет функция одной переменной, являющаяся решением содержащейся внутри блока задачи Коши (дифференциальное уравнение с начальными условиями).



Эта функция не является аналитическим выражением, для вычисления её значения *Mathcad* формирует численную процедуру, использующую численный алгоритм интегрирования дифференциального уравнения. Этим алгоритмом до некоторой степени можно управлять. Выполнив над словом *odesolve* **rightclick** (щелчок правой кнопкой мыши), можно выбрать из выпавшего меню один из 3-х вариантов: *Fixed, Adaptive, Stiff*.

Fixed – интегрирование методом Рунге-Кутты с постоянным (фиксированным) шагом,

Adaptive – интегрирование методом Рунге-Кутты с переменным (настраиваемым, адаптивным) шагом,

Stiff – интегрирование специальным методом, ориентированным на "жёсткие" системы, при этом игнорируются быстрые движения.

Обычно оптимальным является выбор настраиваемого шага интегрирования (*Adaptive*). Другие методы целесообразны в некоторых особых случаях.

Интервал, на котором определено найденное решение, задаётся, с одной стороны, начальными условиями (в нашем случае – 0), а с другой – аргументом функции *odesolve* (в нашем случае – 10). Вне этого интервала функция не определена, и попытка вычислить её значение приведёт к ошибке.

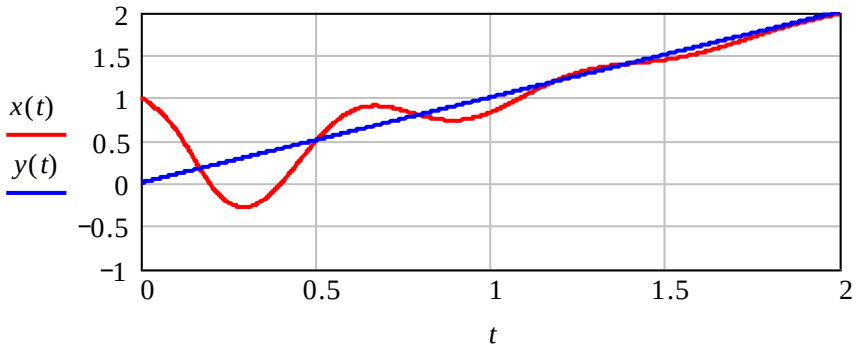
Дифференциальное уравнение может быть записано и в виде системы, причём не обязательно в 1-й нормальной форме Коши. Пусть перед колебательным звеном стоит интегратор с выходом *y*.

Given

$$T^2 \cdot x''(t) + 2 \cdot \xi \cdot T \cdot x'(t) + x(t) = y(t) \qquad x'(0) = 0 \qquad y(0) = 0$$

$$y'(t) = f(t) \qquad x(0) = 1$$

$$\begin{pmatrix} x \\ y \end{pmatrix} := Odesolve\left[\begin{pmatrix} x \\ y \end{pmatrix}, t, 2\right]$$



При этом несколько изменяется вызов функции *Odesolve*. Первым аргументом должен быть вектор, содержащий имена вычисляемых функций. Их должно быть столько же,

сколько дифференциальных уравнений содержит блок. Соответственно и результатом будет вектор, состоящий из имён функций скалярного аргумента.

Функция *odesolve* может принять и ещё один аргумент, количество промежуточных точек решения. Задание слишком маленького значения этого аргумента всегда приводит к ошибкам в решении.

Прямой вызов решателей

Другой способ решения ОДУ в *Mathcad* – прямой вызов решателей:

Rkfixed(x_0, t_1, t_2, *npoints*, *D*) – интегрирование методом Рунге-Кутты с постоянным шагом,

Rkadapt(x_0, t_1, t_2, *npoints*, *D*) – интегрирование методом Рунге-Кутты с настраиваемым (адаптивным) шагом,

Radau(x_0, t_1, t_2, *npoints*, *D*) – интегрирование "жёстких" систем.

Во всех этих функциях набор параметров одинаков:

x_0 – вектор начальных условий, соответствующих моменту времени t_1,

t_1, t_2 – концы интервала свободной переменной (у нас это время), на котором вычисляется решение,

npoints – количество узлов решения на заданном интервале,

D – имя вектор-функции правых частей системы ОДУ, записанной в первой нормальной форме Коши. Эта функция должна, в свою очередь, иметь аргументами свободную переменную (время) и вектор состояния.

Например, для последней рассмотренной системы, имеющей 3-й порядок, потребуется такая запись:

$$F(t,x) := \begin{pmatrix} x_1 \\ \dfrac{x_2}{T^2} - \dfrac{2\cdot\xi}{T}\cdot x_1 - \dfrac{1}{T^2}\cdot x_0 \\ f(t) \end{pmatrix} \qquad x0 := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$X := Rkadapt(x0, 0, 2, 100, F)$$

$X =$

	0	1	2	3
0	0	1	0	0
1	0.02	0.981	-1.89	0.02
2	0.04	0.926	-3.524	0.04
3	0.06	0.842	-4.859	0.06
4	0.08	0.734	-5.867	0.08
5	0.1	0.609	-6.534	0.1
6	0.12	0.475	-6.861	0.12
7	0.14	0.337	-6.862	0.14
8	0.16	0.202	-6.562	0.16

Результатом является матрица, содержащая *npoints* строк, в каждой строке на первом месте стоит значение свободной переменной (время), а следом – соответствующие этому моменту времени значения переменных состояния, в том порядке, в каком они записаны в определении функции *F*. Естественно, что результат будет тем же, что и в предыдущем примере.

Задание на работу

Задана передаточная функция системы (см. Варианты задания). Начальные условия принимаются нулевыми. На вход поступает 2 варианта сигнала – ступенчатый и гармонический. Время интегрирования подберите исходя из длительности переходных процессов. Требуется:

1. Записать одно дифференциальное уравнение и решить его для обоих вариантов входного сигнала с использованием решающего блока. Выбрать метод решения. Построить графики.
2. Представить систему в виде последовательного соединения элементарных звеньев, составить соответствующую систему дифференциальных уравнений и решить с использованием решающего блока. Построить графики.
3. Представить систему в виде параллельного соединения элементарных звеньев, составить соответствующую систему дифференциальных уравнений и решить с использованием решающего блока. Построить графики.
4. Сформировать системы дифференциальных уравнений в 1-й нормальной форме Коши, соответствующих одному дифференциальному уравнению, последовательному и параллельному соединению элементарных звеньев и решить их с использованием прямого вызова решателей. Выбрать метод решения. Построить графики.
5. Сравнить 4 описанных выше варианта решения по трудоёмкости и удобству применения. Сделать выводы о том, когда какой метод предпочтительней использовать.

Варианты задания.

Передаточная функция имеет вид:

$$W(p) = \frac{b_1 p + b_0}{a_3 p + a_2 p + a_1 p + a_0}$$

Входной сигнал:

$$y(t) = \sin(\omega \cdot t + \varphi)$$

№ варианта	a_0	a_1	a_2	a_3	b_0	b_1	ω	φ
1.	0.002	0.091	0.06	0.021	2.23	1.302	0.015	2.842
2.	0.009	0.092	0.095	0.068	2.892	0.363	0.055	1.429
3.	0.009	0.016	0.04	0.037	1.768	2.128	0.047	2.731
4.	0.001	0.057	0.019	0.012	4.98	2.007	0.031	0.123
5.	0.002	0.094	0.058	0.021	3.591	1.653	0.035	2.941
6.	0.008	0.058	0.063	0.066	1.697	0.257	0.061	0.518
7.	0.002	0.026	0.006	0.069	2.1	0.854	0.098	0.367
8.	0	0.036	0.027	0.009	4.356	2.604	0.016	0.372
9.	0.001	0.08	0.09	0.004	2.485	0.354	0.071	1.822
10.	0.003	0.008	0.073	0.031	1.93	2.157	0.078	2.298
11.	0	0.059	0.04	0.015	1.927	2.606	0.063	2.104
12.	0.002	0.021	0.09	0.008	4.884	0.95	0.069	2.541
13.	0.002	0.065	0.039	0.073	1.092	1.97	0.057	3.134
14.	0.009	0.061	0.036	0.07	1.607	2.503	0.055	0.642
15.	0.002	0.045	0.075	0.066	1.037	2.629	0.025	1.22
16.	0.006	0.04	0.074	0.011	3.575	2.746	0.05	1.825

Контрольные вопросы

1. В чём отличие функций Rkfixed, Rkadapt, Radau, Odesolve?
2. Как вставить символ дифференцирования в решающем блоке?
3. Что означает термин «жёсткая система»?
4. Как представить передаточную функцию в виде суммы элементарных дробей?
5. Как представить передаточную функцию в виде произведения элементарных дробей?
6. Как вычислить характеристические числа передаточной функции?
7. Какие численные методы интегрирования ОДУ может реализовать функция odesolve и как сожно управлять выбором конкретного метода?
8. Что оказывается результатом решения, число, массив, фунукция или иная структура?

Список литературы

1. Демидович, Б.П. Основы вычислительной математики : учеб.пособ.для втузов / Б.П.Демидович, И.А.Марон .— 6-е изд.,стер. — СПб.;М.;Краснодар : Лань, 2007 .— 672с (7 экз)

2. Макаров, Н.Н. Применение пакета MATHCAD в анализе и синтезе систем автоматического управления: учеб. пособие / Н.Н. Макаров, С.В. Феофилов. – Тула: Изд-во ТулГУ, 1007. –169 с. (16 экз.)

3. Макаров, Н. Н. Численные методы для задач анализа и синтеза систем автоматического управления : учебное пособие / Н. Н. Макаров, В. Е. Семашкин .— Тула : Изд-во ТулГУ, 2014 .— 124 с. (16 экз.)

4. Методические указания к выполнению лабораторных работ по дисциплине «Численные методы математики»// Электронный ресурс кафедры САУ.

Программное обеспечение и Интернет-ресурсы

1. Образовательный математический сайт Exponenta.ru:
http://www.exponenta.ru/
2. Система программирования ABC Pascal.
3. Математический пакет Mathcad.