

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Кафедра «Автоматика и телемеханика»

Курсовая работа
по дисциплине «Информационные технологии»
на тему «Обработка данных в математических пакетах»

ПГУ 3.27.03.04.01 ПЗ

Направление подготовки - 27.03.04 Управление в технических системах

Профиль подготовки - Управление и информатика в технических системах

Выполнил студент:	Максаков Д.В.
Группа:	153ПА31
Руководитель:	к.т.н., доцент Берестень М.П.

Работа защищена с оценкой

Дата защиты

Пенза 2018

Реферат

Пояснительная записка содержит 29 листов формата А4, 5 рисунков, 3 приложения, 11 источников.

Ключевые слова: ИНТЕРПОЛЯЦИЯ, СГЛАЖИВАНИЕ ДАННЫХ, РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНОГО УРАВНЕНИЯ, ПОСТРОЕНИЕ ГРАФИКА, MATHCAD, ПРОГРАММА.

Цель работы – обработка данных с использованием Mathcad.

В результате выполнения курсовой работы решена задача интерполяции таблично заданной функции, сглаживания случайно сформированных данных, задача решения обыкновенного дифференциального уравнения. Все задачи решены с использованием программы Mathcad с представлением необходимой графической информации.

ПГУ 3.27.03.04.01 ПЗ

					ПГУ 3.27.03.04.01 ПЗ		
Изм.	Лист	№ докум.	Подп.	Дата			
Разраб.		Максаков Д.В.			Обработка данных в Математических пакетах	Лит.	Лист
Пров.		Берестень М.П.					
						4	28
Н. Контр.						Гр.153ПА51	
Утв.							

Содержание

Введение	6
1 Средства MathCad для решения задач	8
1.1 Средства MathCad для решения задачи интерполяции	8
1.2 Средства MathCad для решения задачи сглаживания данных.....	10
1.3 Средства MathCad для решения дифференциального уравнения	11
2 Алгоритмы решения задач средствами MathCad	14
2.1 Алгоритм интерполяции функции средствами MathCad	14
2.2 Алгоритм сглаживания данных средствами MathCad.....	15
2.3 Алгоритм решения дифференциального уравнения средствами MathCad	15
3 Решение задач и анализ результатов	17
3.1 Решение задачи интерполяции функции средствами MathCad и анализ результатов	17
3.2 Решение задачи сглаживания данных средствами Mathcad и анализ результатов	20
3.3 Решение дифференциального уравнения средствами Mathcad и анализ результатов	22
Заключение.....	25
Список использованных источников.....	26
Приложение А.....	27
Приложение Б	28
Приложение В.....	29

Введение

Для выполнения сложных расчетов, связанных с решением многих инженерных задач, может быть использована программа MathCad. Mathcad – это система компьютерной алгебры из класса систем автоматизированного проектирования, ориентированная на подготовку интерактивных документов с вычислениями и визуальным сопровождением, отличается лёгкостью использования и применения для коллективной работы. Mathcad имеет интуитивный и простой для использования интерфейс пользователя. Для ввода формул и данных можно использовать как клавиатуру, так и специальные панели инструментов. Характерной особенностью пакета является использование привычных стандартных математических обозначений, т.е. вид документа на экране максимально приближен к общепринятой математической нотации.

Несмотря на то, что эта программа, в основном, ориентирована на пользователей, не являющихся программистами, Mathcad также используется в сложных проектах, чтобы визуализировать результаты математического моделирования путём использования распределённых вычислений и традиционных языков программирования.

Пакет Mathcad позволяет:

- использовать обычный калькулятор для простых, повторяемых вычислений;
- вычислять и упрощать символьные выражения;
- вычислять интегралы и производные функции;
- решать системы линейных алгебраических уравнений, работать с матрицами и определителями;
- решать системы нелинейных алгебраических уравнений;
- строить графики как в декартовых и цилиндрических, так и в полярных координатах, различные диаграммы и гистограммы;
- создавать программы с разветвляющимися и циклическими алгоритмами, используя свой собственный, интуитивно понятный, язык программирования;

- решать дифференциальные уравнения;
- решать задачи теории вероятности и математической статистики;
- осуществлять обмен информацией с другими приложениями операционной системы Windows, такими, как Excel, Powerpoint, Word;
- документировать расчёты и создавать отчётную документацию;
- использовать более 600 встроенных математических функций;
- осуществлять поддержку шаблонов документов, форматирования текста, форматирования формул и т.д.

Таким образом, целью настоящей курсовой работы является изучение пакета Mathcad как средства решения инженерно-технических задач и реализация с его помощью нескольких примеров конкретных расчетных задач.

В курсовой работе решена задача интерполяции таблично заданной функции, сглаживания случайно сформированных данных, решение обыкновенного дифференциального уравнения. Все задачи решены с использованием пакета Mathcad с представлением необходимой графической информации.

1 Средства MathCad для решения задач

1.1 Средства MathCad для решения задачи интерполяции

Интерполяция использует значения некоторой функции, заданные в ряде точек, чтобы предсказать значения функции между ними. В MathCad можно соединять точки данных прямыми линиями (линейная интерполяция) или соединять их отрезками полинома (сплайн-интерполяция). Функции интерполяции определяют кривую, точно проходящую через заданные точки. Из-за этого результат очень чувствителен к ошибкам данных. Кроме того, следует убедиться, что каждый элемент массива, который используется в любой из функций, описанных в этом разделе, содержит определенное значение. Поскольку MathCAD присваивает нулевое значение любым элементам, которые не заданы.

При **линейной интерполяции** MathCad соединяет существующие точки данных прямыми линиями. Это выполняется функцией **linterp (vx, vy, x)**.

Функция *linterp* использует векторы данных vx и vy, чтобы возвратить интерполируемое значение u, соответствующее третьему аргументу x. Аргументы vx и vy должны быть векторами одинаковой длины. Вектор vx должен содержать вещественные значения, расположенные в порядке возрастания.

Эта функция соединяет точки данных отрезками прямых, создавая таким образом ломаную. Интерполируемое значение для конкретного x есть ордината u соответствующей точки ломаной.

Для значений x, расположенных перед первой точкой в векторе vx, MathCad продолжает ломаную прямой линией, проходящей через первые две точки данных. Для значений x, расположенных за последней точкой vx, MathCad продолжает ломаную прямой линией, проходящей через последние две точки данных.

Для получения наилучших результатов x должно находиться между самыми большими и самыми маленькими значениями vx – маловероятно, что будут полезны значения, вычисленные для x вне этого диапазона. При небольшом числе узловых точек (до 10) линейная интерполяция оказывается довольно гру-

бой. Функция *linterp* предназначена для интерполяции, а не для экстраполяции, и за пределами области определения может вести себя непредсказуемо.

Гораздо лучшие результаты дает **сплайн-интерполяция**. Сплайн-интерполяция позволяет провести кривую через набор точек таким образом, что первые и вторые производные кривой непрерывны в каждой точке. Эта кривая образуется путем создания ряда кубических полиномов, проходящих через наборы из трех смежных точек. Кубические полиномы затем состыкуются друг с другом, чтобы образовать одну кривую.

Для осуществления сплайн-интерполяции Mathcad предлагает четыре встроенные функции. Три из них служат для получения векторов вторых производных сплайн-функций при различных видах продолжения (экстраполяции). Эти три функции отличаются только граничными условиями:

- функция **lspline (vx, vy)** генерирует кривую сплайна, которая приближается к прямой линии в граничных точках;
- функция **pspline (vx, vy)** генерирует кривую сплайна, которая приближается к параболе в граничных точках.
- функция **cspline (vx, vy)** генерирует кривую сплайна, которая является кубическим полиномом в граничных точках.

Они возвращают вектор коэффициентов вторых производных, который будем называть *vs*. Аргументы *vx* и *vy* должны быть вещественными векторами одинаковой длины. Значения вектора *vx* должны быть расположены в порядке возрастания.

Принципиальным выбор типа полинома для сплайна может быть лишь в том случае, если планируется использовать его в качестве экстраполирующей функции. В основном же на практике интерес представляет качественное соединение именно внутренних точек, поэтому, скорее всего, никакой разницы от того, какая функция задействована для задания вектора вторых производных, не будет.

Наконец, четвертая функция **interp (vs, vx, vy, x)** возвращает интерполируемое значение *y*, соответствующее аргументу *x*. Вектор *vs* вычисляется на основе векторов данных *vx* и *vy* одной из функций *lspline*, *pspline* или *cspline*.

Интерполируемое значение для конкретного x есть ордината y соответствующей точки сплайна. Для значений x , расположенных перед первой точкой вектора vx , MathCad продолжает сплайн первой из составляющих его кубических парабол. Для значений x , расположенных за последней точкой vx , MathCad продолжает сплайн последней из составляющих его кубических парабол.

1.2 Средства MathCad для решения задачи сглаживания данных

Очень важной задачей при обработке результатов экспериментов является сглаживание полученных данных. Сглаживание всегда включает некоторый способ локального усреднения данных, при котором несистематические компоненты взаимно погашают друг друга.

Сглаживание предполагает использование набора значений y и возвращение нового набора значений y , который является более гладким, чем исходный набор. В отличие от регрессии и интерполяции сглаживание приводит к новому набору значений y , а не к функции, которая может оценивать значения между заданными точками данных. Каждый элемент в векторе должен иметь значение, т.к. MathCad присваивает значение 0 любым элементам, которые явно не определены.

В MathCad имеются три встроенные функции, позволяющие произвести сглаживание некоторой выборки данных:

– **medsmooth (y, n)**, где y – вектор исходных значений функции, n – параметр, определяющий количество окон сглаживания, на которое будет разбит интервал при обработке данных (n может быть только нечетным целым числом, строго меньшим, чем количество элементов в выборке). Эта функция реализует алгоритм «бегущих» медиан. Обязательным условием при ее применении является то, что эмпирические точки должны быть равномерно распределены на промежутке. Функция *medsmooth* будет оставлять первые и последние $(n-1)/2$ точки неизменяемыми. На практике длина окна сглаживания n должна быть мала по сравнению с длиной набора данных. Из всех встроенных функций

сглаживания MathCad *medsmooth* является наиболее надежной, однако и наименее универсальной функцией.

– **ksmooth (x, y, b)**, где x и y – векторы исходных значений аргумента и функции, b – ширина окна сглаживания (этот параметр по величине должен равняться общей величине нескольких промежутков, разделяющих в данной выборке соседние точки). Данная встроенная функция реализует сглаживание на основании алгоритма Гаусса. Это сглаживание наиболее полезно, когда данные взяты в точках, отделяемых друг от друга интервалами приблизительно равной ширины.

– **supsmooth (x, y)**, где x и y – векторы исходных значений аргумента и функции. Осуществляет сглаживание с помощью адаптивного алгоритма (в основе которого лежит метод наименьших квадратов), основанного на анализе взаимного расположения рассматриваемой точки и ближайших к ней (их количество зависит от особенностей поведения графика зависимости). Элементы вектора x должны быть расположены в порядке возрастания. Адаптивное сглаживание следует использовать, если длина интервалов между точками существенно изменяется.

Все описанные выше функции возвращают в качестве ответа вектор из такого же количества элементов, как в исходных выборках. Поэтому если эмпирических точек было обработано немного, то при построении графика, для получения корректной кривой, стоит использовать возможности сплайн-интерполяции MathCad.

1.3 Средства MathCad для решения дифференциального уравнения

Дифференциальные уравнения являются основой огромного количества расчетных задач из самых различных областей науки и техники – с их помощью можно описать задачи движения системы взаимодействующих материальных точек, химической кинетики, электрических цепей и др.

Дифференциальные уравнения – это уравнения, в которых неизвестные являются не переменные (т.е. числа), а функции одной или нескольких пере-

менных. Эти уравнения (или системы) включают соотношения между искомыми функциями и их производными. Если в уравнения входят производные только по одной переменной, то они называются обыкновенными дифференциальными уравнениями (ОДУ). В противном случае говорят об уравнениях в частных производных. Таким образом, решить (иногда говорят проинтегрировать) дифференциальное уравнение – значит, определить неизвестную функцию на определенном интервале изменения ее переменных. Как известно, одно обыкновенное дифференциальное уравнение или система ОДУ имеет единственное решение, если помимо уравнения определенным образом заданы начальные или граничные условия.

В MathCad нет средств символьного решения дифференциальных уравнений, но достаточно хорошо представлены численные методы их решения. В MathCad нет универсальной функции для решения дифференциальных уравнений, а есть около двадцати функций для различных видов уравнений, дополнительных условий и методов решения. Эти функции можно найти в библиотеке Insert/Function, категория «Differential Equation Solving» («Решение дифференциальных уравнений»).

Основные функции MathCad, дающие решения для ОДУ и систем ОДУ, представленных в обычной форме Коши:

- **rkfixed** (y , $x1$, $x2$, n , F) – возвращает матрицу решений методом Рунге-Кутты при фиксированном шаге (y – вектор начальных условий; $x1$, $x2$ – границы интервала для поиска решения; n – количество точек на интервале; F – вектор-функция первых производных).
- **Rkadapt** (y , $x1$, $x2$, n , F) – возвращает матрицу решений методом Рунге-Кутты с переменным шагом (y – вектор начальных условий; $x1$, $x2$ – границы интервала для поиска решения; n – количество точек на интервале; F – вектор-функция первых производных). Там, где решение меняется медленнее, шаг увеличивается, а в области быстрого изменения решения шаг уменьшается. Возвращается решение с равным шагом. Функция работает быстрее, чем *rkfixed*.
- **Bulstoer** (y , $x1$, $x2$, n , F) – дает более точное решение, методом Булирша-Штера (y – вектор начальных условий; $x1$, $x2$ – границы интервала для поиска

решения; n – количество точек на интервале; F – вектор-функция первых производных). Следует отметить, что имена функций *Rkadapt* и *Bulstoer* начинаются с прописной буквы. В MathCad для некоторых имен функций неважно, с какой буквы они записаны, но для перечисленных функций это принципиально, т.к. в MathCad также существуют функции с такими же именами, только записанные с маленькой буквы – *rkadap*, *bulstoer*. Эти функции используются в тех случаях, когда важным является решение задачи в конечной точке интервала интегрирования.

- Вычислительный блок **Given / Odesolve (x, b, step)**, которая возвращает решение дифференциального уравнения, описанного в блоке *Given* при заданных начальных условиях (x – переменная интегрирования, действительное число; b – конечная точка отрезка интегрирования; *step* – величина шага по переменной интегрирования (необязательный аргумент)). Если указан параметр *step*, то решение выполняется с фиксированным шагом, иначе – адаптивным методом. Хотя аналитическое выражение для этой функции не выводится, с ней можно выполнять математические преобразования, например, дифференцировать.

Функция *Odesolve* возвращает решение задачи в виде функции. Эта функция не имеет символьного представления и может только вернуть численное значение решения уравнения в любой точке интервала интегрирования. Функция *Odesolve* использует для решения дифференциальных уравнений наиболее популярный алгоритм Рунге-Кутты четвертого порядка, описанный в большинстве книг по методам вычислений. Он обеспечивает малую погрешность для широкого класса систем ОДУ за исключением жестких систем. Если щелчком правой кнопки мыши на блоке формул с функцией *Odesolve* вызвать контекстное меню, то можно изменить метод вычисления решения, выбрав один из трех вариантов: *Fixed* – метод Рунге-Кутты с фиксированным шагом интегрирования (этот метод используется по умолчанию), *Adaptive* – также метод Рунге-Кутты, но с переменным шагом, изменяемым в зависимости от скорости изменения функции решения, *Stiff* – метод, адаптированный для решения жестких уравнений и систем.

2 Алгоритмы решения задач средствами MathCad

2.1 Алгоритм интерполяции функции средствами MathCad

Для **линейной интерполяции** функции, заданной в табличном виде, необходимо выполнить следующие действия (ограничений на имена переменных и векторов нет, приведенные обозначения используются для примера):

1) Создать векторы v_x и v_y , содержащие заданные табличные значения аргумента x и функции y . Элементы v_x должны быть расположены в порядке возрастания.

2) Вычислить вектор $vs(v_x) = \text{interp}(v_x, v_y, v_x)$.

3) Построить графики функций: заданной в табличном виде $v_y(v_x)$ и вычисленной по линейной интерполяции $vs(v_x)$.

Для **сплайн-интерполяции** функции, заданной в табличном виде, необходимо выполнить следующие действия (ограничений на имена переменных и векторов нет, приведенные обозначения используются для примера):

1) Создать векторы v_x и v_y , содержащие заданные табличные значения аргумента x и функции y . Элементы v_x должны быть расположены в порядке возрастания.

2) Вычислить векторы вторых производных интерполяционной кривой в рассматриваемых точках:

$vs_1 = \text{lspline}(v_x, v_y)$ – для линейной сплайн-интерполяции,

$vs_2 = \text{pspline}(v_x, v_y)$ – для параболической сплайн-интерполяции,

$vs_3 = \text{cspline}(v_x, v_y)$ – для кубической сплайн-интерполяции.

3) Вычислить векторы сплайн-интерполяции:

$vk_1(v_x) = \text{interp}(vs_1, v_x, v_y, v_x)$ – для линейной сплайн-интерполяции,

$vk_2(v_x) = \text{interp}(vs_2, v_x, v_y, v_x)$ – для параболической сплайн-интерполяции,

$vk_3(v_x) = \text{interp}(vs_3, v_x, v_y, v_x)$ – для кубической сплайн-интерполяции.

4) Построить графики функций: заданной в табличном виде $v_y(v_x)$ и вычисленных по сплайн-интерполяции $vk_1(v_x)$, $vk_2(v_x)$, $vk_3(v_x)$.

2.2 Алгоритм сглаживания данных средствами MathCad

Для сглаживания данных необходимо выполнить следующие действия (ограничений на имена переменных и векторов нет, приведенные обозначения используются для примера):

1) Создать векторы исходных значений x и y . Элементы вектора x должны быть расположены в порядке возрастания.

2) Выполнить сглаживание данных с помощью встроенных функций MathCad:

$s_1 = \text{medsmooth}(y, n)$ – сглаживание с помощью скользящей медианы,

$s_2 = \text{ksmooth}(x, y, b)$ – сглаживание при помощи гауссова ядра данных,

$s_3 = \text{supsmooth}(x, y)$ – сглаживание методом наименьших квадратов.

3) Построить графики: исходных данных $y(x)$ и функций сглаживания $s_1(x)$, $s_2(x)$, $s_3(x)$.

2.3 Алгоритм решения дифференциального уравнения средствами MathCad

Для решения дифференциального уравнения с помощью блока *Given / Odesolve* необходимо выполнить следующие действия (ограничений на имена переменных и векторов нет, приведенные обозначения используются для примера):

1) Набрать ключевое слово *Given*.

2) Записать дифференциальное уравнение и начальные условия в форме логического равенства.

3) Набрать встроенную функцию *Odesolve* (x, b, step), где x – переменная интегрирования, действительное число; b – конечная точка отрезка интегрирования; step – величина шага по переменной интегрирования (необязательный аргумент).

4) Построить график полученного решения.

Для решения дифференциального уравнения с помощью встроенной функции необходимо выполнить следующие действия (ограничений на имена переменных и векторов нет, приведенные обозначения используются для примера):

- 1) Задать границы интервала поиска решения $x1$ и $x2$, количество точек на интервале n .
- 2) Задать начальные условия y .
- 3) Задать функция первых производных $D(x, y)$.
- 4) Набрать одну из встроенных функций: *rkfixed* ($y, x1, x2, n, F$), *Rkadapt* ($y, x1, x2, n, F$) или *Bulstoer* ($y, x1, x2, n, F$). Получить решение в виде таблицы.
- 5) Построить график полученного решения.

3 Решение задач и анализ результатов

3.1 Решение задачи интерполяции функции средствами MathCad и анализ результатов

Задание №1.

Выполнить линейную и сплайн-интерполяцию функции, заданной в табличном виде.

x_i	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
y_i	0	0.1	0.2	0.3	0.47	0.5	0.61	0.7	0.8	0.9

Построить графики исходной функции, а также результаты интерполяции. Провести сравнение полученных результатов.

Решение задачи **линейной интерполяции** функции:

1) Создать векторы заданных табличных значений аргумента и функции:

$$VX := \begin{pmatrix} 0 \\ 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \\ 0.6 \\ 0.7 \\ 0.8 \\ 0.9 \end{pmatrix} \quad VY := \begin{pmatrix} 0 \\ 0.1 \\ 0.2 \\ 0.3 \\ 0.47 \\ 0.5 \\ 0.61 \\ 0.7 \\ 0.8 \\ 0.9 \end{pmatrix}$$

2) Вычислить вектор линейной интерполяции:

$$\underline{A}(VX) := \text{linterp}(VX, VY, VX).$$

3) Построить графики функций: заданной в табличном виде $VY(VX)$ и вычисленной по линейной интерполяции $A(VX)$.

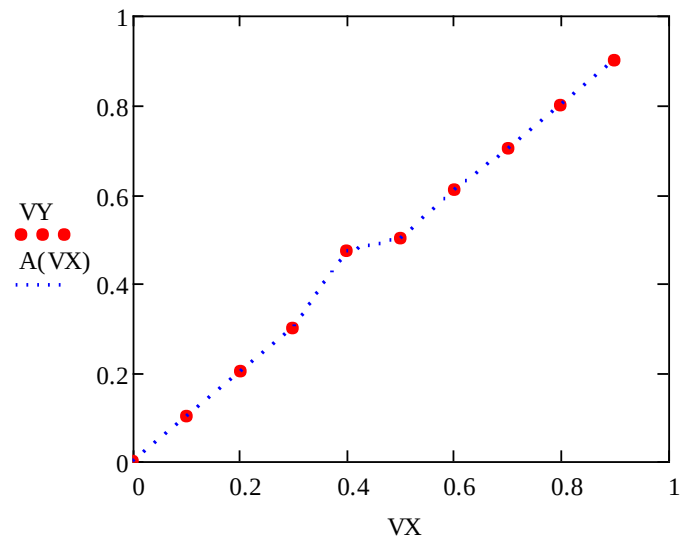


Рисунок 1 – Графики заданной $VY(VX)$ и вычисленной $A(VX)$ функций

Вывод: из анализа построенных графиков, можно сделать вывод о том, что график линейной интерполирующей функции полностью совпадает с графиком исходной функции и представляет собой набор отрезков, соединяющих точки исходных данных. Однако, при помощи интерполяции можно не только соединить точки данных линиями, но и вычислить соответствующие значения, которые могут использоваться для предсказания величины зависимости в тех точках, в которых она не была измерена на опыте. Таким образом, линейная интерполяция дает приемлемый результат для заданных исходных данных.

Решение задачи **сплайн-интерполяции** функции:

1) Создать векторы заданных табличных значений аргумента и функции:

$$VX := \begin{pmatrix} 0 \\ 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \\ 0.6 \\ 0.7 \\ 0.8 \\ 0.9 \end{pmatrix} \quad VY := \begin{pmatrix} 0 \\ 0.1 \\ 0.2 \\ 0.3 \\ 0.47 \\ 0.5 \\ 0.61 \\ 0.7 \\ 0.8 \\ 0.9 \end{pmatrix}$$

2) Вычислить векторы вторых производных интерполяционной кривой в рассматриваемых точках:

$$VSb := lspline(VX, VY)$$

$$VSd := cspline(VX, VY)$$

$$VSd := cspline(VX, VY)$$

3) Вычислить векторы сплайн-интерполяции:

$$B(VX) := \text{interp}(VSb, VX, VY, VX)$$

$$C(VX) := \text{interp}(VSd, VX, VY, VX)$$

$$D(VX) := \text{interp}(VSd, VX, VY, VX)$$

4) Построить графики функций: заданной в табличном виде $VY(VX)$ и вычисленных по сплайн-интерполяции $B(VX)$, $C(VX)$, $D(VX)$.

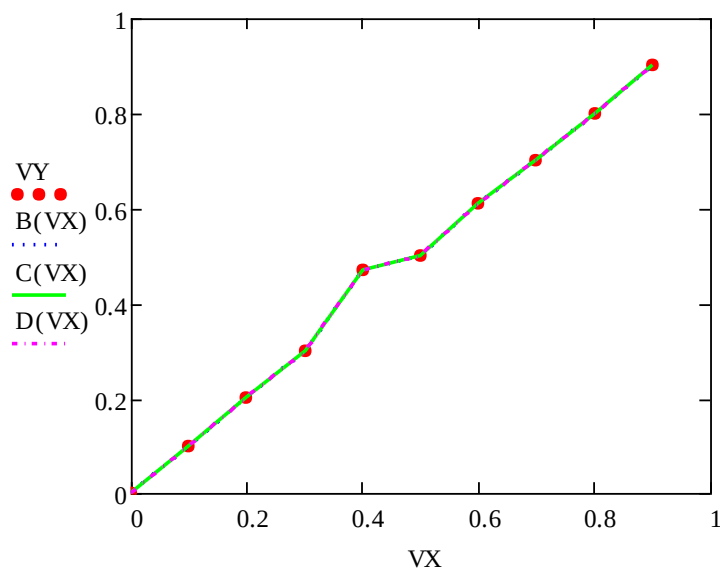


Рисунок 2 – Графики заданной $VY(VX)$ и вычисленных $B(VX)$, $C(VX)$, $D(VX)$ функций

Текст программы линейной и сплайн-интерполяции функции представлен в приложении А.

Вывод: из анализа построенных графиков, можно сделать вывод о том, что графики сплайн-интерполяции представляют собой плавные линии, соединяющих точки исходных данных. В пределах заданных значений все три линии, построенные с помощью полиномов разных типов, совпадают. Таким образом, сплайн-интерполяция дает хороший результат для заданных исходных данных.

3.2 Решение задачи сглаживания данных средствами Mathcad и анализ результатов

Задание №2.

Используя функцию *rnd*, сформировать 50 случайных чисел из диапазона [0,5]. Произвести сглаживание данных (с помощью различных встроенных функций). Построить графики исходной и сглаженной функций. Провести сравнение полученных результатов.

Решение:

1) Создать векторы исходных значений из 50 случайных чисел из диапазона [0,5]:

$$i := 1..50$$

$$x_i := i$$

$$y_i := \text{rnd}(5)$$

x =		0	y =		0
	0	0		0	0
	1	1		1	2.183
	2	2		2	2.889
	3	3		3	3.143
	4	4		4	2.521
	5	5		5	3.479
	6	6		6	0.95
	7	7		7	0.892
	8	8		8	2.287
	9	9		9	0.488
	10	10		10	0.472
	11	11		11	4.657
	12	12		12	4.473
	13	13		13	1.137
	14	14		14	2.054
	15	...		15	...

2) Выполнить сглаживание данных с помощью встроенных функций MathCad:

$a := \text{medsmooth}(y, 11)$ – сглаживание с помощью скользящей медианы,

$b := \text{ksmooth}(x, y, 11)$ – сглаживание при помощи гауссова ядра данных,

$\hat{c} := \text{supsmooth}(x, y)$ – сглаживание методом наименьших квадратов.

3) Построить графики: исходных данных $y_i(x_i)$ и сглаженных функций $a_i(x_i)$, $b_i(x_i)$, $c_i(x_i)$.

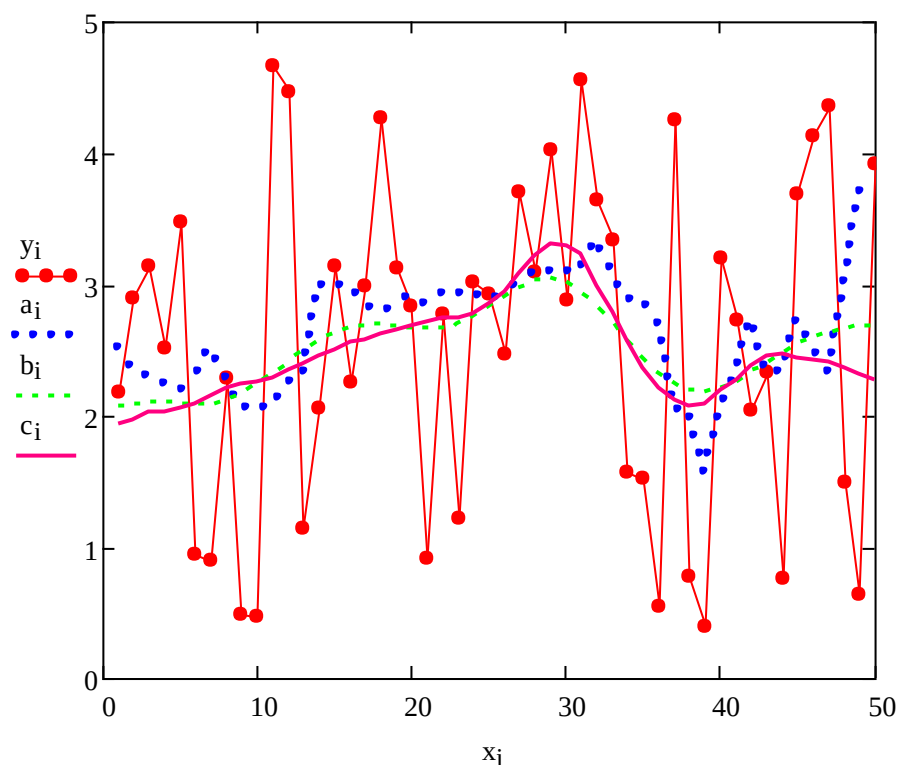


Рисунок 3 – Графики исходных данных $y_i(x_i)$ и сглаженных функций $a_i(x_i)$, $b_i(x_i)$, $c_i(x_i)$.

Текст программы сглаживания данных представлен в приложении Б.

Вывод: из анализа построенных графиков, можно сделать вывод, что принципиальным условием успеха при сглаживании данных является правильный выбор алгоритма в зависимости от характера исходных данных. Кроме того, результаты сглаживания данных с помощью функций *medsmooth* и *ksmooth* сильно зависят от выбора величины окна сглаживания. Таким образом, наилучший результат сглаживания данных показала функция *supsmooth* с адаптивным выбором различной ширины полосы сглаживания для различных частей данных.

3.3 Решение дифференциального уравнения средствами Mathcad и анализ результатов

Задание №3.

Сформировать и решить двумя способами дифференциальное уравнение, построить графики полученных решений.

Решение:

Найти решение обыкновенного дифференциального уравнения

$$y'(x) = x^{\cos(x)}$$

на отрезке $[0;10]$ при начальном условии $y(0)=1$.

Решения с помощью блока *Given / Odesolve*:

1) Набрать ключевое слово *Given*.

2) Записать дифференциальное уравнение и начальные условия в форме логического равенства:

$$\frac{d}{dx}y(x) = x^{\cos(x)}$$

$$y(0) = 1$$

3) Набрать встроенную функцию *Odesolve*:

$$Y := \text{Odesolve}(x, 10)$$

4) Построить график полученного решения $Y(x)$.

$x := 0, 0.01 \dots 10$

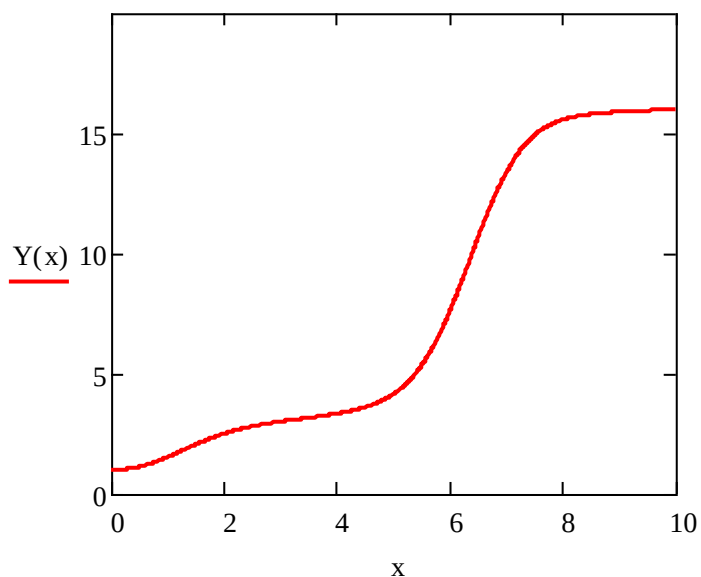


Рисунок 4 – График решения дифференциального уравнения с помощью блока *Given / Odesolve*

Решение дифференциального уравнения с помощью встроенной функции:

1) Задать границы интервала поиска решения и количество точек на интервале:
 $\underline{a} := 0$ $\underline{b} := 10$ $n := 100$

2) Задать начальные условия: $y := 1$

3) Задать функция первых производных:

$$\underline{D}(x, y) := x^{\cos(x)}$$

4) Набрать одну из встроенных функций и получить решение в виде таблицы:

$z := \text{rkfixed}(y, a, b, n, D)$ $z =$

	0	1
0	0	1
1	0.1	1.005
2	0.2	1.02
3	0.3	1.046
4	0.4	1.084
5	0.5	1.133
6	0.6	1.193
7	0.7	1.264
8	0.8	1.344
9	0.9	1.434
10	1	1.531
11	1.1	1.634
12	1.2	...

5) Построить график полученного решения z :

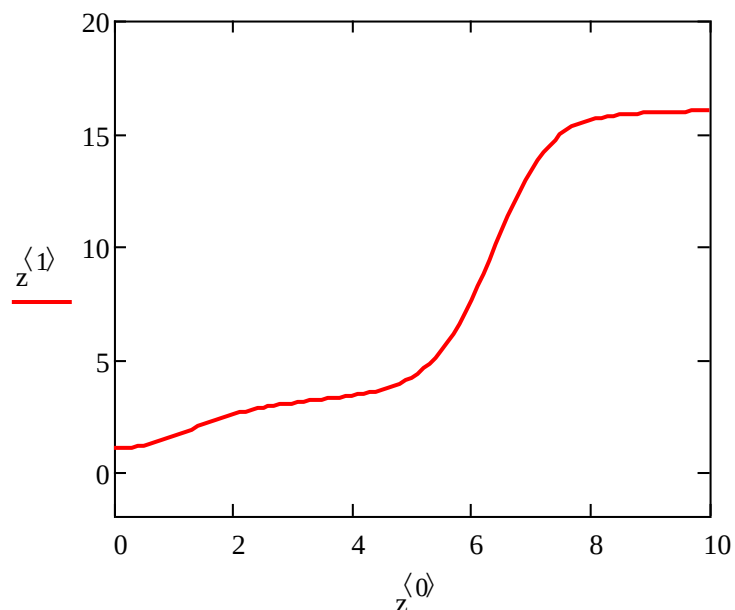


Рисунок 5 – График решения дифференциального уравнения с помощью встроенной функции *rkfixed*

Текст программы решения дифференциального уравнения с помощью блока *Given / Odesolve* и с помощью встроенной функции *rkfixed* представлен в приложении В.

Вывод: из анализа построенных графиков, можно сделать вывод, что результаты решения обыкновенного дифференциального уравнения с помощью блока *Given / Odesolve* и с помощью встроенной функции *rkfixed* полностью совпадают.

Заключение

В результате выполнения курсовой работы решена задача интерполяции таблично заданной функции, сглаживания случайно сформированных данных, задача интегрирования обыкновенного дифференциального уравнения. Все задачи решены с использованием программы Mathcad с представлением необходимой графической информации.

Список использованных источников

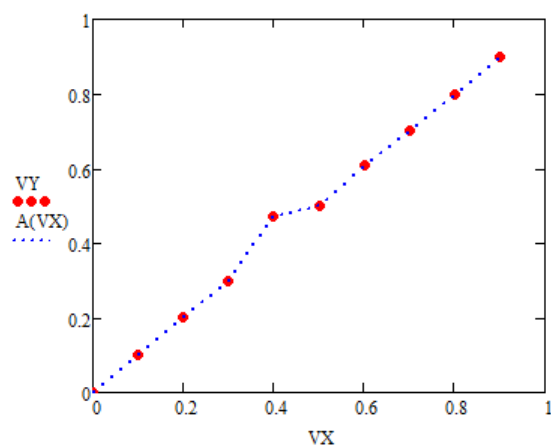
1. Гурский Д.А., Турбина Е.С. Вычисления в Mathcad 12. – СПб.: Питер, 2006, 544 с. ISBN 5-469-00639-5.
2. Вельмисов П.А., Киреев С.В. Дифференциальные уравнения в Mathcad : учебное пособие. – Ульяновск: УлГТУ, 2016, 109 с. ISBN 978-5-9795-15.
3. Макаров Е. Инженерные расчеты в Mathcad 15. Учебный курс. – СПб.: Питер, 2011.
4. Охорзин В.А. Прикладная математика в системе MATHCAD. Учебное пособие. 3-е изд. – СПб.: Лань, 2009, 352с. ISBN: 978-5-8114-0814-6.
5. Охорзин В.А. Компьютерное моделирование в системе Mathcad. – М.: Финансы и статистика, 2006, 144с. ISBN: 5-279-03037-6.
6. Кирьянов Д.В. Mathcad 13 (+ CD-ROM). – СПб.: БХВ-Петербург, 2006, 598 с. ISBN: 5-94157-850-4.
7. Дьяконов В. VisSim+Mathcad+MATLAB. Визуальное математическое моделирование. – М.: СОЛОН-Пресс, 2004.
8. Очков В.Ф. Mathcad 14 для студентов и инженеров: русская версия. – СПб.: BHV, 2009.
9. Тарасевич Ю. Информационные технологии в математике. - М: СОЛОН-Пресс, 2003.
10. Алексеев Е.Р., Чеснокова О.В. Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9. – М: ИТ Пресс, 2006, 496с. ISBN: 5-477-00208-5.
11. Бидасюк Ю.М. Mathcad для студента. – М.: Вильямс, 2006.

Текст программы интерполяции

$$VX := \begin{pmatrix} 0 \\ 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \\ 0.6 \\ 0.7 \\ 0.8 \\ 0.9 \end{pmatrix} \quad VY := \begin{pmatrix} 0 \\ 0.1 \\ 0.2 \\ 0.3 \\ 0.47 \\ 0.5 \\ 0.61 \\ 0.7 \\ 0.8 \\ 0.9 \end{pmatrix}$$

Линейная интерполяция

$A(VX) := \text{interp}(VX, VY, VX)$



Сплайн-интерполяция

Линейная сплайн-интерполяция

$VSb := \text{lspline}(VX, VY)$

$B(VX) := \text{interp}(VSb, VX, VY, VX)$

Параболическая сплайн-интерполяция

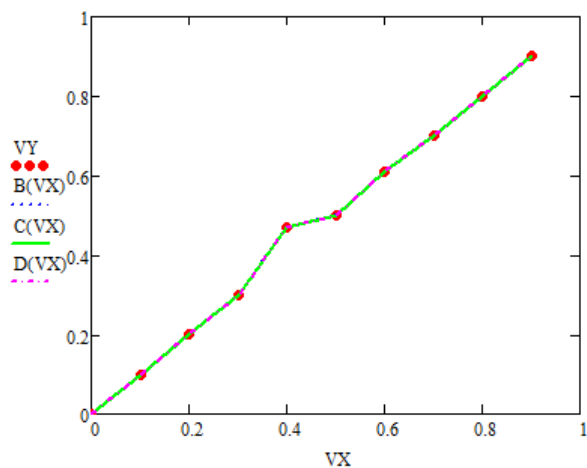
$VSc := \text{pspline}(VX, VY)$

$C(VX) := \text{interp}(VSc, VX, VY, VX)$

Кубическая сплайн-интерполяция

$VSd := \text{cspline}(VX, VY)$

$D(VX) := \text{interp}(VSd, VX, VY, VX)$



Текст программы сглаживания данных

$i := 1..50$

$x_i := i$

$y_i := \text{md}(5)$

	0		0
0	0	0	0
1	1	1	2.183
2	2	2	2.889
3	3	3	3.143
4	4	4	2.521
5	5	5	3.479
6	6	6	0.95
7	7	7	0.892
8	8	8	2.287
9	9	9	0.488
10	10	10	0.472
11	11	11	4.657
12	12	12	4.473
13	13	13	1.137
14	14	14	2.054
15	...	15	...

Сглаживание с помощью скользящей медианы

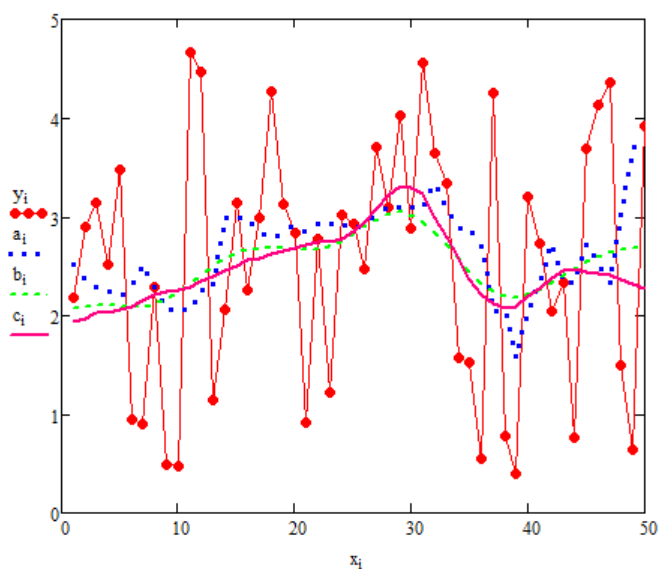
$a := \text{medsmooth}(y, 11)$

Сглаживание при помощи гауссова ядра данных

$b := \text{ksmooth}(x, y, 11)$

Сглаживание методом наименьших квадратов

$c := \text{supsmooth}(x, y)$



Текст программы решения дифференциального уравнения

Способ 1

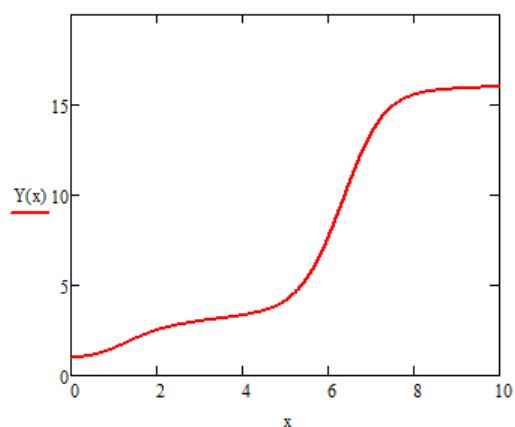
Given

$$\frac{d}{dx}y(x) = x^{\cos(x)}$$

$$y(0) = 1$$

$Y := \text{Odesolve}(x, 10)$

$x := 0, 0.01 \dots 10$



Способ 2

$a := 0$ $b := 10$ границы интервала поиска решения

$n := 100$ количество точек на интервале

$y := 1$ начальные условия

$D(x, y) := x^{\cos(x)}$ функция первых производных

$z := \text{rkfixed}(y, a, b, n, D)$

	0	1
0	0	1
1	0.1	1.005
2	0.2	1.02
3	0.3	1.046
4	0.4	1.084
5	0.5	1.133
6	0.6	1.193
7	0.7	1.264
8	0.8	1.344
9	0.9	1.434
10	1	1.531
11	1.1	1.634
12	1.2	...

$z =$

