ФГБОУ ВПО «Воронежский государственный технический университет»

Кафедра компьютерных интеллектуальных технологий проектирования

381-2014

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторной работы
по дисциплине «Математическое обеспечение САПР»
для студентов направления
230100.62 «Информатика и вычислительная техника»
(профиль «Системы автоматизированного
проектирования в машиностроении»)
заочной формы обучения

Воронеж 2014

Составители: канд. техн. наук Д.Н. Пименов,
ст. преп. А.А. Пак

УДК 681.3

Методические указания к выполнению лабораторной работы по дисциплине «Математическое обеспечение САПР» для студентов направления 230100.62 «Информатика и вычислительная техника» (профиль «Системы автоматизированного проектирования в машиностроении») заочной формы обучения /ФГБОУ ВПО «Воронежский государственный технический университет»;сост.Д.Н.Пименов, А.А. Пак. Воронеж, 2014. 62 с.

Методические указания содержат необходимые для выполнения лабораторной работы теоретические сведения, примеры выполнения заданий.

Предназначены для студентов 3 курса.

Методические указания подготовлены в электронном виде в текстовом редакторе Word и содержатся в файле «Математическое обеспечение САПР Лабораторная работа.doc».

Библиогр.: 10 назв.

Рецензент канд.физ.-мат. наук, доц. В.В. Горбунов

Ответственный за выпуск зав. кафедрой д-р техн. наук, проф. М.И. Чижов

Издается по решению редакционно-издательского совета Воронежского государственного технического университета

ЛАБОРАТОРНАЯ РАБОТА № 1
ОСНОВЫ РАБОТЫ С MathCAD

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

Математические выражения

К основным элементам математических выражений Mathcad относятся т*ипы данных, операторы, функции и управляющие структуры*.

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. Имена констант, переменных и иных объектов называют *идентификаторами*. Идентификаторы в Mathcad представляют собой набор латинских или греческих букв и цифр.

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.

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.

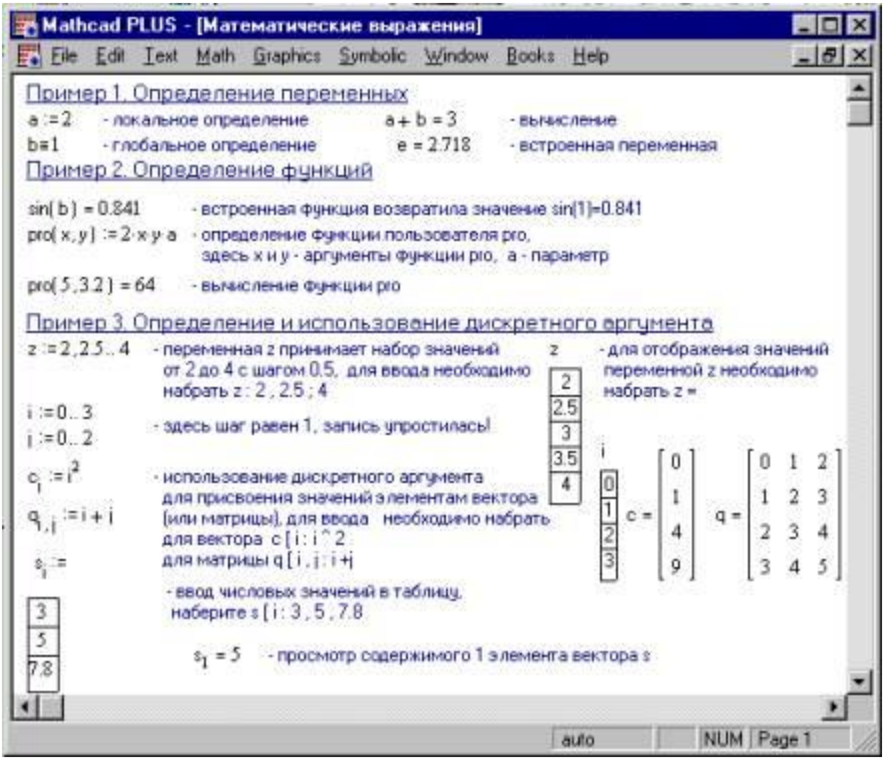

Рис. 1. Математические выражения

Если переменной присваивается начальное значение с помощью оператора :=, такое присваивание называется локальным. До этого присваивания переменная не определена и ее нельзя использовать. Однако с помощью знака ≡можно обеспечить глобальное присваивание (см. Пример 1 Рисунка 1). Существует также жирный знак равенства, который используется, например, как оператор приближенного равенства при решении систем уравнений.

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. После указания операндов (параметров операторов) операторы становятся исполняемыми по документу блоками, например, 2 + 5 -оператор сложения с двумя операндами. В Приложении 2 данного пособия приведен список наиболее часто используемых операторов.

Функции

В пакете Mathcad имеется множество встроенных функций, т.е. функций, заблаговременно введенных разработчиками (см. Приложение 3). 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.

Важной особенностью пакета является возможность задания внешних функций, или функций пользователя. 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).

Дискретные аргументы

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, либо в виде чисел с определенным шагом, меняющихся от начального значения до конечного.

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

Массивы

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) и 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 **Math->Matrics**,
- с использованием дискретного аргумента (Пример 3 Рис. 1).

Текстовые фрагменты

Текстовые фрагменты представляют собой куски текста, которые пользователь хотел бы видеть в своем документе. Существуют два вида текстовых фрагментов - текстовая область (**region**) и текстовый диапазон (**band**). Текстовые области предназначены для небольших кусков текста - подписей, комментариев и т.п. Текстовые диапазоны применяются в том случае, если необходимо работать с абзацами или страницами.

Графические области

Графические области делятся на три основных типа - двумерные графики, трехмерные графики и импортированные графические образы. Двумерные и трехмерные графики строятся самим Mathcad на основании обработанных данных.

Создание анимационного клипа

Mathcad имеет встроенную переменную **FRAME**, чье единственное назначение - управление анимациями:

- Создайте объект, чей вид зависит от **FRAME**.
- Выберите **Windows->Animation->Create** для вызова диалогового окна.

- Заключите в выделяющий пунктирный прямоугольник часть рабочего документа, которую нужно анимировать.
- Установите нижние и верхние границы **FRAME**.
- В поле **At (Темп)** введите значение скорости воспроизведения (кадр/сек).
- Выберите **Animate**. Сейчас анимация только создается.
- Сохраните анимацию как **AVI** файл (Save as).
- Воспроизведите сохраненную анимацию **Windows->Animation-> Playback**.

Сообщения об ошибках

При выполнении вычислений возможны ошибки. Сообщение об ошибке в Mathcad выводится в красном прямоугольнике, от которого отходит линия, указывающая на место ошибки. В Приложении 4 приведен список сообщений об ошибках.

Порядок выполнения лабораторной работы № 1

Задание 1. Вычислить:

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

Это и все остальные задания снабдить комментариями, используя команды **Text□□ Create Text Region** или **Text□□ Create Text Paragraph**.

Задание 2. Іпределить переменные: $a := 3.4$, $b := 6.22$, $c \equiv 0.149$ (причем переменную c - глобально) и выражения:

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

- вычислить выражения.
- С помощью команды **Math□□ Numerical Format□□ Displayed Precision** изменить точность отображения результатов вычисления *глобально*.

Задание 3. Вывести на экран значение *системной константы* □ и установить максимальный формат ее отображения *локально*.

Задание 4. Выполнить следующие операции с комплексными числами:

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

$\sqrt{Z}$ = , $\sqrt{-5}$ =, 2 *Z* = ,

*Z*1:= 1+2*i*, *Z*2:= 3+4*i*, *Z*1+*Z*2 = , *Z*1 - *Z*2 = , *Z*1 *Z*2 = , *Z*1/*Z*2 = .

Задание 5. Выполнить следующие операции:

i := 1 .. 10, $\sum_i i$ = , $\prod_i (i+1)$ = ,

$\int_0^{0.4} x^2 \cdot \lg(x+2)\,dx$ = , $\int_{0.8}^{1.2} \dfrac{\operatorname{ctg} 2x}{(\sin 2x)^2}\,dx$ = ,

x := 2, $\dfrac{d}{dx} x^5$ = , $\dfrac{d}{dx} \sin(x)$ = .

i	d_i	S_i	R_i
0	0.5	3.3	2
1	1	5.9	3.9
2	1.5	7	4.5
3	2	6.3	3.7
4	2.5	4.2	1.2

Задание 6. Определить векторы *d, S* и *R* через дискретный аргумент *i*. Отобразить графически таблично заданные функции $R_i(d_i)$ и $S_i(d_i)$, используя команду **Graphics□□ Create X-Y Plot**. Чтобы оформить график, необходимо выполнить следующие команды:

- Щелкнуть мышью на графике, чтобы выделить его, при этом MathCAD заменит меню **Graphics** на меню **X-Y Plot.**
- Выбрать **X-Y Plot□□ Format** (появится диалоговое окно **"Formatting Currently Selected X-Y Plot"**) и отформатировать график так, чтобы в каждой узловой точке графика функции $S_i(d_i)$ стоял знак вида □ (**Traces□□ Symbol□□ box**), а график функции $R_i(d_i)$ отобразить в виде гистограммы (**Trace□□ tupe□□ bar**).
- Нанести линии сетки на график (**X-Y Axes□□ Grid Lines**) и отобразить легенду (**Traces□□ Hide Legend**).

Задание 7. Построить декартовы (**X-Y Plot**) и полярные (**Polar Plot**) графики следующих функций:

$$X(\alpha) := \cos(\alpha)\cdot\sin(\alpha),$$

$$Y(\alpha) := 1.5\cos(\alpha)^2 - 1,$$

$$P(\alpha) := \cos(\alpha).$$

Для этого необходимо определить □ как дискретный аргумент на интервале от 0 до 2 □ с шагом □ /30.

Определить по графику **X-Y Plot** координаты любой из точек пересечения графиков *Y*(□) и *P*(□), для этого необходимо:

- Выделить график и выбрать **X-Y Plot□□ Zoom** (появится диалоговое окно **"X-Y Zoom"**) для увеличения части графика в области точки пересечения.
- На чертеже выделить пунктирным прямоугольником окрестность точки пересечения графиков *Y*(□) и *P*(□), которую нужно увеличить.
- Нажать кнопку **Zoom**, чтобы перерисовать график.
- Чтобы сделать это изображение постоянным, выбрать **Accept**.
- Выбрать **X-Y Plot□□ Trace** (появится диалоговое окно **"X-Y Trace"**).
- Внутри чертежа нажать кнопку мыши и переместить указатель мыши на точку, чьи координаты нужно увидеть.
- Выбрать **Copy <u>X</u>** (или **Copy <u>Y</u>**), на свободном поле документа набрать *Xper* := (или *Yper* :=) и выбрать пункт меню **Edit□□ Paste.**

Вычислить значения функций *X*(□) и *Y*(□) при □ :=□ □ 2.

Задание 8. Используя команду **Math□□ Matrics** создать матрицу *Q* размером 6□на 6, заполнить ее произвольно и отобразить графически с помощью команды **Graphics□□ Create Surface Plot**.

Задание 9. Построить график поверхности (**Surface Plot**) и карту линий уровня (**Contour Plot**) для функции двух переменных:

$$X(t, \alpha) := t \cdot \cos\ \alpha \cdot \sin\ \alpha.$$

- Определить функцию *X*(*t*,□)

- Задать на осях переменных *t* и □ по 41 точке (*i*:=*0..40*, *j*:=*0..40*): для переменной t_i со значениями, изменяющимися от -5 до 5 с шагом 0.25 (t_i := -5 + 0.25 *i*), а для переменной □ $_j$ - от 0 до 2□ с шагом □ /20 (□ $_j$:= □ /20 *j*).
- Определить матрицу $M_{i,\ j}$:= *X*(t_i,□ $_j$) и отобразить ее графически.

С помощью команды **Graphics□□ 3D Plot Format** вызвать диалоговое окно "**3D Plot Format**" и изменить:

- характеристики просмотра (**View□□ Rotation, Tilt**),
- цвета и линии поверхности (**Color&Lines□□ Shading**),
- параметры осей (**Axes**),
- вид заголовка графика (**Title**).

Задание 10. Используя переменную **FRAME** и команду **Animation□□ Create**, создать анимационные клипы с помощью данных приведенных в таблице:

Варианты задания 10

№ варианта	Переменные	Функции	FRAME	Тип графика
1	*x* := 0,0.1..30	*f*(*x*):=*x* + FRAME	от 0 до 20	полярный (**Polar Plot**)
2	*i*:=0..FRAME +1	g_i:=0.5 *i* cos(*i*) h_i:=*i*sin(*i*) k_i:=2 *i*	от 0 до 50	трехмерный точечный график (**3D Scatter**

				Plot) границы на осях Min Max x -50 50 y -50 50 z 0 50
3	i:=0..20 j:=0..20	$f(x,y)$:=sin(x^2+y^2+FRAME) x_i:=-1.5+0.15i y_j:=-1.5+0.15j	от 0 до 50	график поверхности (**Surface Plot**)
4	r:= FRAME, R := 6 m := 0..20 n := 0..20	$x_{m,n}$:=(R+r□ cos(v_n)) cos(w_m) $y_{m,n}$:=(R+$r$cos(v_n)) sin(w_m) $z_{m,n}$:= r sin(v_n)	от 0 до 11	график параметрической поверхности (**Surface Plot**) (границы на всех осях установить от -11 до 11)
5	r:= FRAME, R := 6 m := 0..20 n := 0..20	$x_{m,n}$:=(R+r□ cos(v_n))□ cos(w_m) $y_{m,n}$:=(R+r□ cos(v_n))□ sin(w_m) $z_{m,n}$:= r□ sin(v_n)	от 0 до 11	график параметрической поверхности (**Surface Plot**) (границы на

				всех осях установить от -11 до 11)

ЛАБОРАТОРНАЯ РАБОТА № 2

РЕШЕНИЕ УРАВНЕНИЙ

Как известно, многие уравнения и системы уравнений не имеют аналитических решений. В первую очередь это относится к большинству трансцендентных уравнений. Доказано также, что нельзя построить формулу, по которой можно было бы решить произвольное алгебраическое уравнение степени выше четвертой. Однако такие уравнения могут решаться итерационными методами с заданной точностью.

Итерационные методы

Задача нахождения корня уравнения $f(x) = 0$ итерационными методами состоит в следующем:

- *отделение корней* - отыскание приближенного значения корня (например, графическим методом);
- *уточнение корней* - доведение их значений до заданной степени точности □ .

При использовании *метода Ньютона* необходимо задаться начальным приближением x_0, расположенным достаточно близко к точному значению корня. Итерационный процесс строится по формуле:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad f'(x_i) \neq 0,\ i = 0, 1, \dots \tag{1}$$

Метод *простых итераций* решения уравнения $f(x) = 0$ состоит в замене исходного уравнения эквивалентным ему уравнением x = □ (x) и построении итерационной последовательности по формуле:

$$x_{i+}1 = □\ (x_i),\quad i = 0, 1, \dots$$

Достаточным условием сходимости рассмотренных итерационных процессов является выполнение неравенства

$$|x_i - x_{i-1}| \leq \varepsilon$$

на каждом шаге итерации.

until**(***a*, *z***)** возвращает *z*, пока выражение *a* не становится отрицательным; *a* должно содержать дискретный аргумент.

Рис.2 иллюстрирует использование функции *until* для реализации *метода Ньютона*.

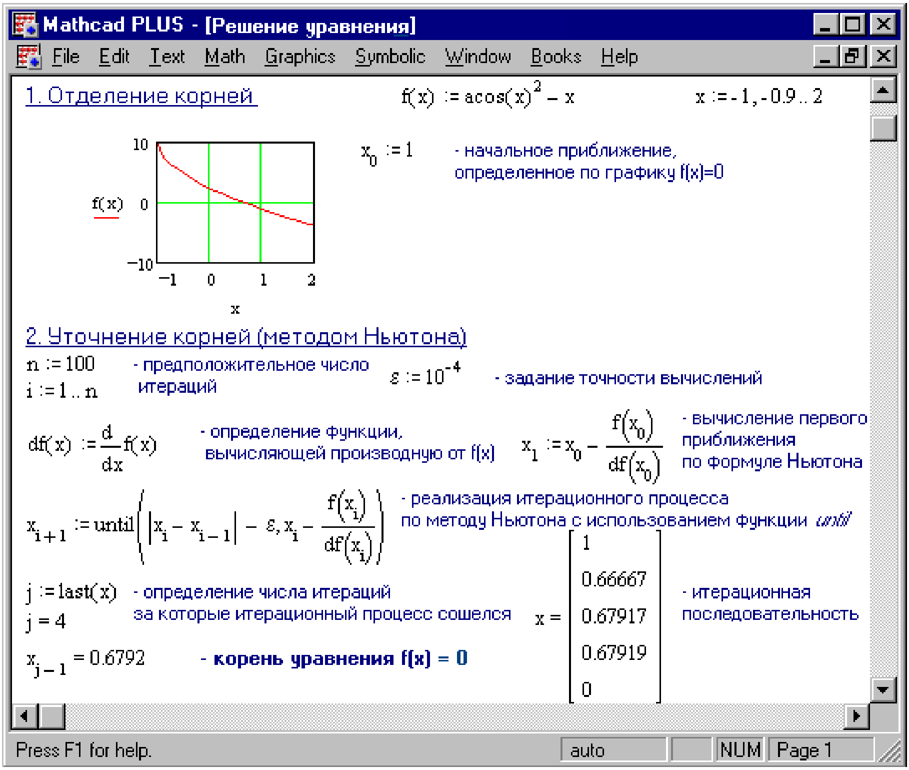

Рис. 2. Решение уравнений средствами Mathcad

Для простейших уравнений вида $f(x) = 0$ решение находится с помощью функции *root.*

root(*f*(*z*), *z*)	Возвращает значение *z*, при котором выражение или функция *f*(*z*) обращается в 0. Оба аргумента этой функции должны быть скалярами. Функция возвращает скаляр.

Первый аргумент - или функция, определенная где-либо в рабочем документе, или выражение. Второй аргумент - имя переменной, которая используется в выражении. Этой переменной перед использованием функции *root* необходимо присвоить числовое значение.

Для нахождения корней выражения, имеющего вид

$$v_n x^n + \ldots + v_2 x^2 + v_1 x + v_0,$$

лучше использовать функцию *polyroots*, нежели *root*. В отличие от функции *root*, функция *polyroots* не требует начального приближения и возвращает сразу все корни, как вещественные, так и комплексные.

polyroots(**v**)	Возвращает корни полинома степени *n*. Коэффициенты полинома находятся в векторе **v** длины *n* + 1. Возвращает вектор длины *n*, состоящий из корней полинома.

Системы линейных уравнений удобно решать с помощью функции *lsolve*.

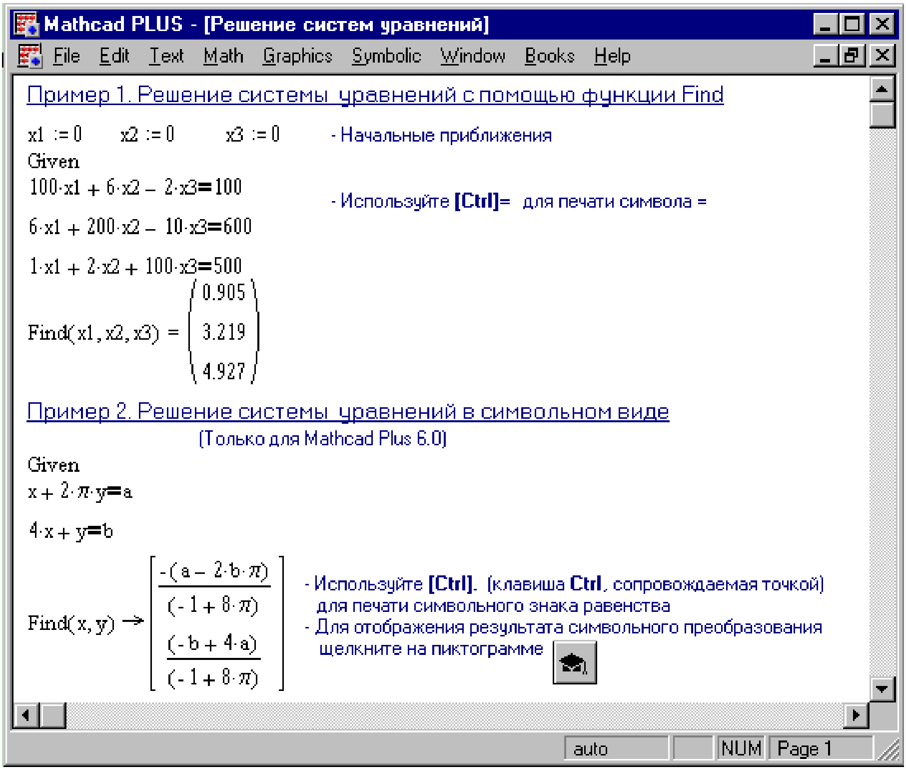

Рис.3. Решение систем уравнений

lsolve**(M, v)**	Возвращается вектор решения **z** такой, что **M** □ **z = v.**

При решении систем уравнений используется специальный вычислительный блок, открываемый служебным словом *Given* и оканчивающийся выражением с функциями *Find* или *Minerr*.

Find(*z1, z2, . . .*)	Возвращает точное решение системы уравнений. Число аргументов должно быть равно числу неизвестных.
Minerr(*z1, z2, . . .*)	Возвращает приближенное решение системы уравнений. Число аргументов должно быть равно числу неизвестных.

Пример 1 на Рис.2 иллюстрирует решение системы уравнений с помощью вычислительного блока *Given ... Find*.

Символьное решение уравнений и систем уравнений

Если задано некоторое выражение *f*(*x*) и отмечена переменная *x*, то команда **Symbolic □ Solve for Variable (Решить относительно переменной)** возвращает символьные значения указанной переменной *x*, при которой *f*(*x*) = 0.

> Если вы работаете с пакетом Mathcad PLUS 5.0, не забудьте предварительно использовать команду **Symbolic □ Load Symbolic Processor** для загрузки символьного процессора.

Если вы работаете с пакетом Mathcad PLUS 6.0, то сможете решать символьно не только уравнения, но и системы уравнений. Чтобы решить систему уравнений в символьном виде, не нужно задавать начальные приближения. Пример 2 Рисунка 2 показывает решение системы уравнений в символьном виде.

Порядок выполнения лабораторной работы 2

Задание 1. Построить график функции $f(x)$ и приблизительно определить один из корней уравнения.

Решить уравнение $f(x) = 0$ с точностью $\varepsilon = 10^{-4}$:

- с помощью встроенной функции Mathcad *root*;
- методом Ньютона (касательных), используя функцию *until*;
- методом итераций, используя функцию *until*.

Определить число итераций в каждом методе, с помощью функции *last*.

Варианты задания 1

№ варианта	$f(x)$	№ варианта	$f(x)$	№ варианта	$f(x)$
1	$3\sin(\sqrt{x}) + 0.35x - 3.8$ $x \in [2, 3]$	**6**	$0.25x^3 + x - 2$ $x \in [0, 2]$	**11**	$\sqrt{2x^2 + 1.2 - \cos x} - 1$ $x \in [0, 1]$
2	$x - \dfrac{1}{3 + \sin(3.6x)}$ $x \in [0, 1]$	**7**	$\arccos \dfrac{1 - x^2}{1 + x^2} - x$ $x \in [2, 3]$	**12**	$\cos\left(\dfrac{2}{x}\right) - 2\sin\left(\dfrac{1}{x}\right) + \dfrac{1}{x}$ $x \in [1, 2]$
3	$\arccos x - \sqrt{1 - 0.3x^3}$ $x \in [0, 1]$	**8**	$3x - 4\ln x - 5$ $x \in [2, 4]$	**13**	$0.1x^2 - x\ln x$ $x \in [1, 2]$
4	$\sqrt{1 - 0.4x^2} - \arcsin x$ $x \in [0, 1]$	**9**	$e^x - e^{-x} - 2$ $x \in [0, 1]$	**14**	$1 - x + \sin x - \ln(1 + x)$ $x \in [0, 2]$
5	$3x - 14 + e^x - e^{-x}$ $x \in [1, 3]$	**10**	$\sqrt{1 - x} - \mathrm{tg}\, x$ $x \in [0, 1]$	**15**	$e^{x-1} - x^3 - x$ $x \in [0, 1]$

Задание 2. Для полинома $g(x)$ выполнить следующие действия:

- с помощью команды **Symbolic Polynomial Coefficients** создать вектор V, содержащий коэффициенты полинома;
- решить уравнение $g(x) = 0$ с помощью функции *polyroots*;
- решить уравнение символьно, используя команду **Symbolic Solve for Variable**;
- разложить на множители, используя **Symbolic Factor Expression.**

Варианты задания 2

№ варианта	$g(x)$	№ варианта	$g(x)$
1	$x^4 - 2x^3 + x^2 - 12x + 20$	**9**	$x^4 + x^3 - 17x^2 - 45x - 100$
2	$x^4 + 6x^3 + x^2 - 4x - 60$	**10**	$x^4 - 5x^3 + x^2 - 15x + 50$
3	$x^4 - 14x^2 - 40x - 75$	**11**	$x^4 - 4x^3 - 2x^2 - 20x + 25$
4	$x^4 - x^3 + x^2 - 11x + 10$	**12**	$x^4 + 5x^3 + 7x^2 + 7x - 20$
5	$x^4 - x^3 - 29x^2 - 71x -140$	**13**	$x^4 - 7x^3 + 7x^2 - 5x + 100$
6	$x^4 + 7x^3 + 9x^2 + 13x - 30$	**14**	$x^4 + 10x^3 +36x^2 +70x+ 75$
7	$x^4 + 3x^3 - 23x^2 - 55x - 150$	**15**	$x^4 + 9x^3 + 31x^2 + 59x+ 60$
8	$x^4 - 6x^3 + 4x^2 + 10x + 75$		

Задание 3. Решить систему линейных уравнений:

- используя функции *Find*;
- матричным способом, используя функцию *lsolve.*

Варианты задания 3

№ варианта	Система линейных уравнений	№ варианта	Система линейных уравнений
1	$\begin{cases} 2x_1 + x_2 + 2x_3 + 3x_4 = 8 \\ 3x_1 + 3x_3 = 6 \\ 2x_1 - x_2 + 3x_4 = 4 \\ x_1 + 2x_2 - x_3 + 2x_4 = 4 \end{cases}$	**9**	$\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 = -4 \\ x_1 - 3x_2 - 6x_4 = -7 \\ 2x_2 - x_3 + 2x_4 = 2 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -2 \end{cases}$
2	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 22 \\ 2x_1 + 3x_2 + x_3 + 2x_4 = 17 \\ x_1 + x_2 + x_3 - x_4 = 8 \\ x_1 - 2x_3 - 3x_4 = -7 \end{cases}$	**10**	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 26 \\ 2x_1 + 3x_2 + 4x_3 + x_4 = 34 \\ 3x_1 + 4x_2 + x_3 + 2x_4 = 26 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$
3	$\begin{cases} 9x_1 + 10x_2 - 7x_3 - x_4 = 23 \\ 7x_1 - x_3 - 5x_4 = 37 \\ 5x_1 - 2x_3 + x_4 = 22 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$	**11**	$\begin{cases} 2x_1 - 8x_2 - 3x_3 - 2x_4 = -18 \\ x_1 - 2x_2 + 3x_3 - 2x_4 = 28 \\ x_2 + x_3 + x_4 = 10 \\ 11x_2 + x_3 + 2x_4 = 21 \end{cases}$
4	$\begin{cases} 6x_1 - x_2 + 10x_3 - x_4 = 158 \\ 2x_1 + x_2 + 10x_3 + 7x_4 = 128 \\ 3x_1 - 2x_2 - 2x_3 - x_4 = 7 \\ x_1 - 12x_2 + 2x_3 - x_4 = 17 \end{cases}$	**12**	$\begin{cases} 2x_1 - x_2 + 4x_3 + x_4 = 66 \\ 2x_2 - 6x_3 + x_4 = -63 \\ 8x_1 - 3x_2 + 6x_3 - 5x_4 = 146 \\ 2x_1 - 7x_2 + 6x_3 - x_4 = 80 \end{cases}$
5	$\begin{cases} x_1 - 2x_2 + 6x_3 + x_4 = 88 \\ 5x_1 + 2x_3 - 3x_4 = 88 \\ 7x_1 - 3x_2 + 7x_3 + 2x_4 = 181 \\ 3x_1 - 7x_2 + 5x_3 + 2x_4 = 99 \end{cases}$	**13**	$\begin{cases} 2x_1 - 3x_3 - 2x_4 = -16 \\ 2x_1 - x_2 + 13x_3 + 4x_4 = 213 \\ 3x_1 + x_2 + 2x_3 + x_4 = 72 \\ x_1 - 12x_3 - 5x_4 = -159 \end{cases}$
6	$\begin{cases} x_1 - 2x_2 - 8x_4 = -7 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -8 \\ x_1 + x_2 - 5x_3 + x_4 = -10 \\ 2x_1 - x_2 + 2x_4 = 7 \end{cases}$	**14**	$\begin{cases} 7x_1 + 7x_2 - 7x_3 - 2x_4 = 5 \\ 3x_1 + 4x_2 + 5x_3 + 8x_4 = 60 \\ 2x_1 + 2x_2 + 2x_3 + x_4 = 27 \\ 2x_1 - 2x_3 - x_4 = -1 \end{cases}$

7	$\begin{cases} 2x_1 + 2x_2 + 6x_3 + x_4 = 15 \\ -x_2 + 2x_3 + x_4 = 18 \\ 4x_1 - 3x_2 + x_3 - 5x_4 = 37 \\ 3x_1 - 5x_2 + x_3 - x_4 = 30 \end{cases}$	**15**	$\begin{cases} 6x_1 - 9x_2 + 5x_3 + x_4 = 124 \\ 7x_2 - 5x_3 - x_4 = -54 \\ 5x_1 - 5x_2 + 2x_3 + 4x_4 = 83 \\ 3x_1 - 9x_2 + x_3 + 6x_4 = 45 \end{cases}$
8	$\begin{cases} 4x_1 - 5x_2 + 7x_3 + 5x_4 = 165 \\ 2x_1 + x_2 - 3x_3 - x_4 = -15 \\ 9x_1 + 4x_3 - x_4 = 194 \\ x_1 - x_2 - 2x_3 - 3x_4 = -19 \end{cases}$		

Задание 4. Преобразовать нелинейные уравнения системы к виду $f_1(x) = y$ и $f_2(y) = x$..

Построить их графики и определить начальное приближение решения. Решить систему нелинейных уравнений, используя функцию *Minerr*.

Варианты задания 4

№ варианта	Система нелинейных уравнений	№ варианта	Система нелинейных уравнений
1	$\begin{cases} \sin x + 2y = 2, \\ \cos(y-1) + x = 0.7. \end{cases}$	**9**	$\begin{cases} \sin y + x = -0.4, \\ 2y - \cos(x+1) = 0. \end{cases}$
2	$\begin{cases} \sin(x+0.5) - y = 1, \\ \cos(y-2) + x = 0. \end{cases}$	**10**	$\begin{cases} \sin(x+2) - y = 1.5, \\ \cos(y-2) + x = 0.5. \end{cases}$
3	$\begin{cases} \cos x + y = 1.5, \\ 2x - \sin(y-0{,}5) = 1. \end{cases}$	**11**	$\begin{cases} \cos(x+0.5) - y = 2, \\ \sin y - 2x = 1. \end{cases}$
4	$\begin{cases} \cos(x+0{,}5) + y = 0.8, \\ \sin y - 2x = 1{,}6. \end{cases}$	**12**	$\begin{cases} \cos(x-2) + y = 0, \\ \sin(y+0.5) - x = 1. \end{cases}$
5	$\begin{cases} \sin(x-1) = 1.3 - y, \\ x - \sin(y+1) = 0.8. \end{cases}$	**13**	$\begin{cases} \cos(x+0.5) + y = 1, \\ \sin(y+0.5) - x = 1. \end{cases}$
6	$\begin{cases} \cos(x+0.5) + y = 1, \\ \sin y - 2x = 2. \end{cases}$	**14**	$\begin{cases} \sin(x) - 2y = 1, \\ \cos(y+0.5) - x = 2. \end{cases}$

7	$\begin{cases} -\sin(x+1)+y=0.8, \\ \sin(y-1)+x=1.3. \end{cases}$	**15**	$\begin{cases} 2y-\sin(x-0.5)=1, \\ \cos(y)+x=1.5. \end{cases}$
8	$\begin{cases} \sin(x)-2y=1, \\ \sin(y-1)+x=1.3. \end{cases}$		

Задание 5. Символьно решить системы уравнений:

$$\begin{cases} 3x+4\pi y=a, \\ 2x+y=b. \end{cases} \quad \begin{cases} 2y-\pi z=a, \\ \pi z-z=b, \\ 3y+x=c. \end{cases}$$

ЛАБОРАТОРНАЯ РАБОТА № 3
ИНТЕРПОЛЯЦИЯ И ПРЕДСКАЗАНИЕ

Аппроксимация функций заключается в приближенной замене заданной функции $f(x)$ некоторой функцией □ (x) так, чтобы отклонение функции □ (x) от $f(x)$ в заданной области было наименьшим. Функция □ (x) при этом называется *аппроксимирующей*. Типичной задачей аппроксимации функций является задача *интерполяции*. Необходимость *интерполяции* функций в основном связана с двумя причинами:

1. Функция $f(x)$ имеет сложное аналитическое описание, вызывающее определенные трудности при его использовании (например, $f(x)$ является спецфункцией: гамма-функцией, эллиптической функцией и др.).
2. Аналитическое описание функции $f(x)$ неизвестно, т.е. $f(x)$ задана таблично. При этом необходимо иметь аналитическое описание приближенно представляющее $f(x)$ (например, для вычисления:

значений $f(x)$ в произвольных точках, определения интегралов и производных от $f(x)$ и т. п.)

Интерполяция

Простейшая задача *интерполяции* заключается в следующем. Для заданных $n + 1$ точек $x_i = x_0, x_1, \ldots, x_n$, которые называются *узлами интерполяции,* и значений в этих точках некоторой функции $f(x_i) = y_0, y_1, \ldots, y_n$ построить полином $\varphi(x)$ (*интерполяционный полином*) степени n вида

$$\varphi(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0, \qquad (1)$$

принимающий в узлах интерполяции x_i те же значения y_i, что и функция $f(x_i)$:

$$\varphi(x_0) = y_0,\ \varphi(x_1) = y_1,\ \ldots,\ \varphi(x_n) = y_n,\ i=0,1\ \ldots,\ n. \qquad (2)$$

Глобальная интерполяция

Простейшим видом *глобальной интерполяции* является *параболическая интерполяция*, когда, используя описанные выше условия (2), для отыскания неизвестных $n + 1$ коэффициентов $a_0, a_1, \ldots, a_n$ выражения (1) получают систему из $n + 1$ уравнений:

$$\begin{cases} a_n x_0^n + a_{n-1} x_0^{n-1} + \ldots + a_1 x_0 + a_0 = y_0, \\ a_n x_1^n + a_{n-1} x_1^{n-1} + \ldots + a_1 x_1 + a_0 = y_1, \\ \cdot \quad \cdot \quad \cdot \quad \cdot \\ a_n x_n^n + a_{n-1} x_n^{n-1} + \ldots + a_1 x_n + a_0 = y_n \end{cases} . \qquad (3)$$

Интерполяционная формула Лагранжа:

$$L_n(x) = \sum_{i=0}^{n} y_i \frac{(x - x_0)(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)(x_i - x_1)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}. \qquad (4)$$

Для построения *интерполяционной формулы Лагранжа* в Mathcad удобно использовать функцию *if*

if**(***cond*, *tval*, *fval***)**	Возвращает значение *tval*, если *cond* отличен от 0 (истина). Возвращает значение *fval*, если *cond* равен 0 (ложь).

Часто интерполирование ведется для функций, заданных таблично с *равноотстоящими* значениями аргумента (h_i = x_{i+}1 - x_i = const). Введем предварительно понятие *конечных разностей:*

$$\Delta y_i = y_{i+1} - y_i, (i = 0, 1, \ldots, n - 1)$$
$$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i, (i = 0, 1, \ldots, n - 2)$$
$$\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i, (i = 0, 1, \ldots, n - k)$$

С учетом введенных обозначений *первая интерполяционная формула Ньютона* имеет вид:

$$t = \frac{x - x_0}{h}, \qquad (5)$$
$$P_{n1}(x) = P_{n1}(x_0 + th) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \ldots$$
$$\ldots + \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_0.$$

Вторая интерполяционная формула имеет вид:

$$t = \frac{x - x_n}{h},$$
$$P_{n2}(x) = P_{n2}(x_n + th) = y_n + t\Delta y_{n-1} + \frac{t(t+1)}{2!}\Delta^2 y_{n-2} + \ldots$$
$$\ldots + \frac{t(t+1)\ldots(t+n-1)}{n!}\Delta^n y_0. \qquad (6)$$

Однако, интерполяция при большом числе узлов приводит к необходимости работать с многочленами высокой степени (например, 50-й или даже 100-й), что неприемлемо как с точки зрения вычислений, так и из-за склонности таких многочленов к осцилляции (колебаниям) между узлами сетки. Поэтому на практике часто используют интерполяцию кусочными многочленами (или *локальную интерполяцию*).

Локальная интерполяция

При *локальной* интерполяции между различными узлами выбираются различные многочлены невысокой степени. В среде Mathcad есть для этого инструментарий: средства *линейной интерполяции* (функция *linterp*) и *интерполяции сплайном* (функция *interp*) - линейным (*lspline*), параболическим (*pspline*) и кубическим (*cspline*). Рисунок 4 показывает некоторые примеры локальной интерполяции.

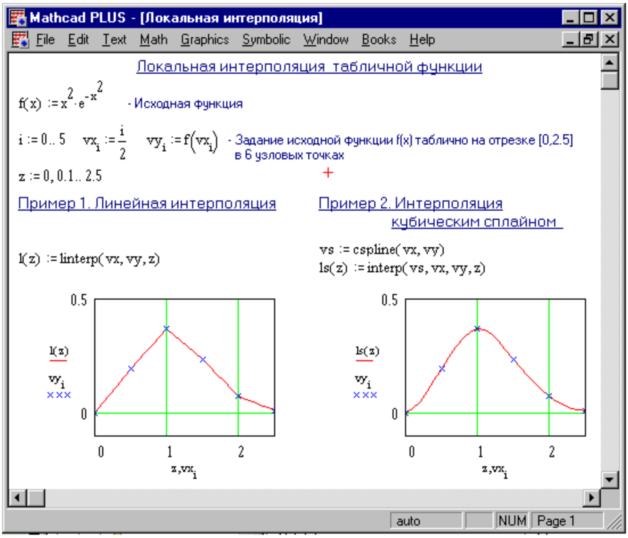

Рис. 4. Локальная интерполяция

linterp(*vx*, *vy*, *x*)	Использует векторы данных *vx* и *vy*, чтобы возвратить линейно интерполируемое значение *y*, соответствующее третьему аргументу *x*.
lspline(*vx*, *vy*) pspline(*vx*, *vy*) cspline(*vx*, *vy*)	Все эти функции возвращают вектор коэффициентов вторых производных, который мы будем называть *vs*. Вектор *vs*, используется в функции *interp*:
interp(*vs*, *vx*, *vy*, *x*)	Возвращает интерполируемое значение *y*, соответствующее аргументу *x*.

Предсказание

Если необходимо оценить значения функции в точках не принадлежащих отрезку [x_0, x_n], используйте функцию *predict* (Рисунок 5).

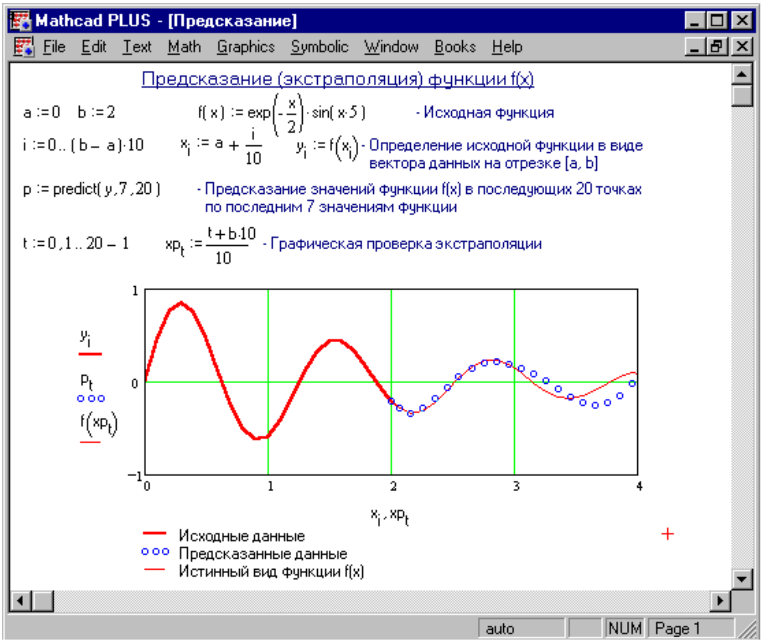

Рис. 5. Экстраполяция функций

predict(*v*, *m*, *n*)	Возвращает *n* предсказанных значений, основанных на *m* последовательных значениях вектора данных *v*.

Порядок выполнения лабораторной работы 3

Задание 1. Вычислить значения заданной функции $y_i = f(x_i)$ в узлах интерполяции $x_i = a + h\,i$, где $h = (b - a)/10$, $i = 0, 1, ..., 10$, на отрезке $[a, b]$.

Варианты заданий

№ варианта	$f(x)$	$[a, b]$	№ варианта	$f(x)$	$[a, b]$
1	$\sin x^2$	[0, 2]	**9**	$x \cdot \cos(x + \ln(1 + x))$	[1, 5]
2	$\cos x^2$	[0, 2]	**10**	$10 \cdot \ln 2x/(1 + x)$	[1, 5]
3	$e^{\sin x}$	[0, 5]	**11**	$\sin x^2 \cdot e^{-(x/2)^2}$	[0, 3]
4	$1/(0.5 + x^2)$	[0, 2]	**12**	$\cos\left(x + \cos^3 x\right)$	[0, 2]
5	$e^{-(x + \sin x)}$	[2, 5]	**13**	$\cos(x + e^{\cos x})$	[3, 6]
6	$1/(1 + e^{-x})$	[0, 4]	**14**	$\cos(2x + x^2)$	[0, 1]
7	$\sin(x + e^{\sin x})$	[0, 3]	**15**	$e^{\cos x} \cos x^2$	[0, 2]
8	$e^{-(x + 1/x)}$	[1, 3]			

Задание 2. По вычисленной таблице (x_i, y_i) провести *параболическую интерполяцию.*

Для нахождения коэффициентов искомого полинома (1) необходимо составить систему линейных алгебраических уравнений (3).

Систему уравнений решить матрично с использованием функции *lsolve*.

Построить график интерполяционного многочлена и отметить на нем узловые точки (x_i, y_i).

Задание 3. Для вычисленной табличной функции составить формулу интерполяционного многочлена *Лагранжа*, используя операторы суммирования и перемножения по дискретному аргументу, а также функцию *if*.

Построить график интерполяционного многочлена и отметить на нем узловые точки (x_i, y_i).

Задание 4. Провести интерполирование заданной функции с помощью 1ой и 2ой интерполяционных формул *Ньютона*.

Построить графики интерполяционных многочленов и отметить на нем узловые точки (x_i, y_i).

Задание 5. Провести *линейную интерполяцию* заданной функции с помощью встроенной интерполяционной функции *linterp*.

Построить график функции *linterp* и отметить на нем узловые точки (x_i, y_i).

Задание 6. Провести *сплайн-интерполяцию* с помощью функций *lspline, pspline, cspline* и *interp*.

Построить график функции *interp* и отметить на нем узловые точки (x_i, y_i).

Задание 7. Вычислить значения заданной функции $y_i = f(x_i)$ в точках $x_i = a + i/10$, где, $i = 0, 1, ..., 10(b - a)$, на отрезке $[a, b]$.

С использованием функции *predict* выполнить *предсказание* (*экстраполяцию*) полученного вектора данных y_i в последующих **10** точках по последним 7 значениям функции.

Отобразить графически имеющиеся данные, предсказанные данные и истинный вид функции $f(x)$.

ЛАБОРАТОРНАЯ РАБОТА № 4
МАТЕМАТИЧЕСКАЯ ОБРАБОТКА РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Изучая в лабораторной работе № 3 теорию интерполяции, вы познакомились с интерполяционными формулами, которые в точности воспроизводят значения данной функции в узлах интерполяции. Однако в ряде случаев выполнение этого условия затруднительно или даже нецелесообразно:

1. Если заданные величины x и y являются экспериментальными данными, то могут содержать в себе существенные ошибки, т.к. получены в результате измерений или наблюдений. Поэтому построение аппроксимирующего многочлена, воспроизводящего в точности заданное значение функции, означало бы тщательное копирование допущенных при измерениях ошибок.

2. Если имеются точные значения функции в некоторых точках, но число таких точек n весьма велико, то

интерполяционный многочлен будет очень высокой степени (если только разности не будут становиться постоянными).

Поэтому возникает задача построения многочлена некоторой вполне определенной степени, но меньшей чем n - 1, который хотя и не дает точных значений функции в узлах интерполяции, но достаточно близко к ним подходит.

Метод наименьших квадратов

Метод наименьших квадратов (*МНК*) состоит в следующем: для данных значений $x = x_0, x_1, \ldots, x_n$ и $y = y_0, y_1, \ldots, y_n$ подобрать многочлен заданной степени $m<n$ вида

$$\varphi(x) = a_m x^m + a_{m-1} x^{m-1} + \ldots + a_1 x + a_0, \tag{1}$$

принимающий в заданных точках x_i значения как можно более близкие к табличным значениям y_i. Коэффициенты a_i многочлена (1) находят из решения системы

$$\begin{cases} b_{00}a_0 + b_{01}a_1 + \ldots + b_{0m}a_m = c_0, \\ b_{10}a_0 + b_{11}a_1 + \ldots + b_{1m}a_m = c_1, \\ \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ b_{m0}a_0 + b_{m1}a_1 + \ldots + b_{mm}a_m = c_m. \end{cases} \tag{2}$$

где $$b_{k,l} = \sum_{i=0}^{n} x_i^{k+l}, \; c_k = \sum_{i=0}^{n} x_i^k y_i, \quad k,l = 0,1,\ldots,m. \tag{3}$$

Регрессионный анализ

Пусть имеются два ряда чисел $x = x_0, x_1, \ldots, x_n$ и $y = y_0, y_1, \ldots, y_n$, при этом предполагается, что ряд y каким-либо образом зависит от ряда x. *Задача регрессионного анализа* состоит в восстановлении математической зависимости (*регрессии*) $y(x)$ по результатам измерений (x_i, y_i), $i = 0, 1, \ldots, n$.

Mathcad включает ряд функций для вычисления регрессии. Функции отличаются прежде всего типом кривой, которую они используют, чтобы аппроксимировать данные.

Линейная регрессия

Встроенные функции *intercept* (to intercept - отложить отрезок на линии) и *slope* (наклон) решают самую простую и распространенную задачу *линейной регрессии* экспериментальных данных:

$f(x) = slope(vx, vy)\, x + intercept(vx, vy)$

slope(*vx*, *vy*)	Возвращает скаляр: наклон линии для данных из *vx* и *vy*.
intercept(*vx*, *vy*)	Возвращает скаляр: смещение по оси ординат линии регрессии для данных из *vx* и *vy*.

Полиномиальная регрессия

Используйте функцию *regress*, когда нужно получить единственный полином произвольной степени, чтобы приблизить все данные. Не рекомендуется делать степень аппроксимирующего полинома выше 4 - 6, поскольку погрешности реализации регрессии сильно возрастают.

regress**(***vx*, *vy*, *n***)**	Возвращает вектор *vs*, требуемый *interp* (см. Лабораторную работу № 3), чтобы найти полином порядка *n*, который наилучшим образом приближает данные из *vx* и *vy*.

Пример 1 Рисунка 6 иллюстрирует использование функции *regress*. Так как *regress* приближает все точки данных, используя один полином, это не дает хороший результат, когда данные не связаны единой полиномиальной зависимостью.

Функция *loess* облегчает эти проблемы, выполняя локальное приближение. Вместо одного полинома *loess* создает различные полиномы второго порядка в зависимости от расположения на кривой (см. Пример 2 Рисунка 6).

*loess***(***vx*, *vy*, *span***)**	Возвращает вектор *vs*, требуемый *interp*, чтобы найти набор полиномов второго порядка, которые наилучшим образом приближают определенные окрестности выборочных точек, определенных в векторах *vx* и *vy*. Аргумент *span* > 0 определяет, насколько большие

	окрестности *loess* будет использовать при выполнении локального приближения.

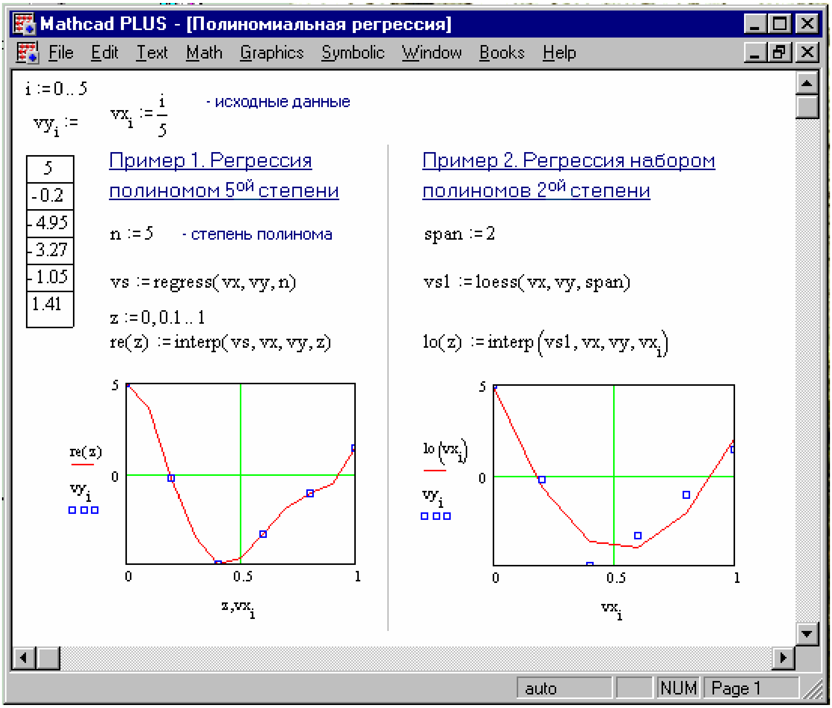

Рис. 6. Полиномиальная регрессия

Обобщенная регрессия

Линейная или полиномиальная регрессия не во всех случаях подходят для описания зависимости данных. Бывает, что нужно искать эту зависимость в виде линейных комбинаций произвольных функций, ни одна из которых не является полиномом. Если предполагается, что данные могли бы быть смоделированы в виде линейной комбинации произвольных функций

$$f(x) = a_0 f_0(x) + a_1 f_1(x) + \ldots + a_n f_n(x),$$

следует использовать *linfit*, чтобы вычислить a_i. Это так называемая *линейная регрессия общего вида* (Пример 1 Рисунка 7).

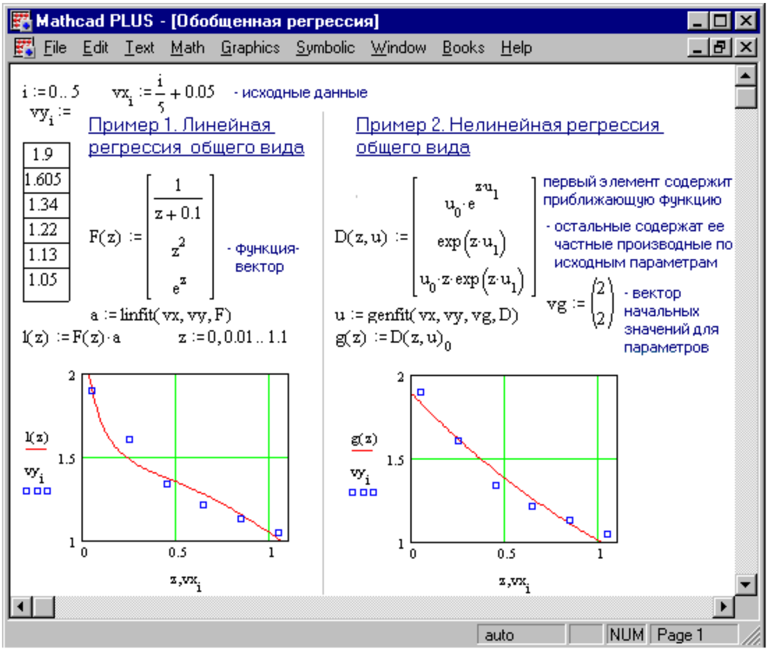

Рис. 7. Обобщенная регрессия

linfit(*vx*, *vy*, *F*)	Возвращает вектор коэффициентов линейной регрессии общего вида, чтобы создать

	линейную комбинацию функций из F, дающую наилучшую аппроксимацию данных из векторов *vx* и *vy*. F - функция-вектор, состоящая из функций, которые нужно объединить в виде линейной регрессии.

Если данные должны быть смоделированы в виде

$f(x) = f(x, u_0, u_1, \ldots, u_n)$,

нужно использовать функцию *genfit*, чтобы найти неизвестные параметры u_i. Это *нелинейная регрессия общего вида* (Пример 2 Рисунка 7).

genfit**(***vx*, *vy*, *vg*, *F***)**	Возвращает вектор n параметров $u_0, u_1, \ldots, u_n$-1, которые обеспечивают наилучшее приближение данных из *vx* и *vy* функцией f, зависящей от x и параметров $u_0, u_1, \ldots, u_{n-}$1. F - функция-вектор, состоящая из f и ее частных производных (вот здесь-то и пригодятся средства символьной математики (Подробнее см. Лабораторную работу 5)) относительно параметров. *vg* - n-мерный вектор начальных значений для n параметров.

Сглаживание

Сглаживание предполагает использование набора значений *y* (и возможно *x*) 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.

ksmooth(*vx*, *vy*, *b*)	Возвращает *n*-мерный вектор, созданный сглаживанием при помощи гауссова ядра данных из *n*-мерного вектора *vy*. Параметр *b* управляет окном сглаживания и должен быть в несколько раз больше величины интервала между точками *x*.
medsmooth(*vy*, *m*)	Возвращает *n*-мерный вектор, созданный сглаживанием *n*-мерного вектора *vy* с помощью скользящей медианы. *m* - ширина окна, по которому происходит сглаживание, причем *m* должно быть нечетным числом и $m < n$.
supsmooth(*vx*, *vy*)	Возвращает *n*-мерный вектор, созданный локальным использованием симметричной линейной процедуры сглаживания МНК.

Порядок выполнения лабораторной работы № 4

Задание 1. Создайте таблицу экспериментальных данных:

$x_i = a + h\,i$, $i = 0, 1, \ldots, 10$, $h=(b - a)/10$ на отрезке $[a, b]$.

Варианты задания 1

№ варианта	y_i	$[a,b]$
1	2.86; 2.21; 2.96; 3.27; 3.58; 3.76; 3.93; 3.67; 3.90; 3.64; 4.09	[0, 1]
2	1.14; 1.02; 1.64; 1.64; 1.96; 2.17; 2.64; 3.25; 3.47; 3.89; 3.36;	[-1, 1]
3	4.70; 4.64; 4.57; 4.45; 4.40; 4.34; 4.27; 4.37; 4.42; 4.50; 4.62	[2, 4]
4	0.43; 0.99; 2.07; 2.54; 1.67; 1.29; 1.24; 0.66; 0.43; 0.35; 0.70	[2, 4]
5	1.55; 1.97; 1.29; 0.94; 0.88; 0.09; 0.02; 0.84; 0.81; 0.09; 0.15	[1, 4]
6	3.24; 1.72; 1.95; 2.77; 2.47; 0.97; 1.75; 1.55; 0.12; 0.70; 1.19	[0, 4]
7	2.56; 1.92; 2.85; 2.94; 2.39; 2.16; 2.51; 2.10; 1.77; 2.28; 1.70	[-1, 2]
8	1.77; 0.92; 2.21; 1.50; 3.21; 3.46; 3.70; 4.02; 4.36; 4.82; 4.03	[-1, 3]
9	1.53; 0.45; 1.68; 0.12; 0.68; 2.36; 2.58; 2.53; 3.45; 2.70; 2.82	[4, 8]
10	2.50; 3.90; 3.54; 4.63; 3.87; 5.25; 4.83; 3.24; 3.08; 3.00; 4.70	[0, 5]
11	2.95; 3.38; 2.71; 2.37; 2.29; 2.75; 2.76; 2.74; 2.57; 2.40; 2.99	[1, 5]
12	-0.23; -0.03; -0.98; -0.97; -0.43; -0.91; -0.27; -0.19; 0.88; 1.06; 0.72	[2, 4]
13	2.36; 0.03; -0.38; -1.33; 0.25; -1.36; 0.95; 3.16; 4.03; 4.92; 4.20	[0, 2]
14	3.82; 4.07; 3.53; 4.83; 5.53; 5.04; 5.09; 5.87; 5.53; 4.72; 4.73	[3, 4]
15	2.35; 2.16; 2.39; 2.39; 2.18; 2.09; 2.44; 2.56; 3.35; 3.22; 2.65	[-3, 4]

Задание 2. Аппроксимировать многочленами 2-ой и 6-ой степени по *методу наименьших квадратов* функцию, заданную таблицей значений x_i и y_i и сравнить качество приближений. Построить графики многочленов и отметить узловые точки (x_i, y_i).

Задание 3. Для приведенных в таблице экспериментальных данных (x_i, y_i) определить параметры *линейной регрессии* с использованием встроенных функций Mathcad *slope* и *intercept*. Отобразить графически совокупность точек векторов x_i и y_i и результаты проведенной линейной регрессии.

Задание 4. Аппроксимировать данные из векторов x_i и y_i

- полиномом 4-ой степени при помощи функций *regress* и *interp*;
- наборами полиномов второго порядка с помощью функций *loess* и *interp*, (при *span* равном 0,5 и 2,5).

Отобразите графически результаты аппроксимации.

Задание 5. Аппроксимировать экспериментальные данные из таблиц значений x_i и y_i линейной комбинацией функций:

$$f(x) = a_1 f_1(x) + a_2 f_2(x) + a_3 f_3(x).$$

Коэффициенты вектора *a* найти с помощью функции *linfit*. Отобразить графически совокупность точек векторов x_i и y_i и результаты проведенной *линейной регрессии общего вида.*

Варианты задания 5

№ варианта	$f_1(x)$	$f_2(x)$	$f_3(x)$
1	e^x	$1/\sqrt{1+2\cos^2 x}$	$\sin x$
2	$1/(1+x^2)$	e^x	$\sin(3x)$
3	$1/(1+x^2)$	$e^{\sin x}$	x
4	$\operatorname{arctg} x$	$\ln \ln x$	$\sin x$

5	$e^{-x^2/2}$	$1 / x$	e^{-x}
6	$(1+x)/(2+x)$	$\cos(x/10)$	$\cos x$
7	$1/(1+e^{x^2})$	$\sqrt{1+x^2}$	$\cos x$
8	$\cos(x/2)$	$2 - \cos x$	$\sin(x/2)$
9	$1/(1+e^x)$	$\mathrm{arctg}\sqrt{x}$	$\sin(3x)$
10	$\ln(x+5)$	$\sqrt{1+x}$	$\sin x$
11	$1 / x$	$\sqrt{1+x}$	$1 / x^2$
12	$\cos x$	$1/(1+x+x^2)$	$1/(1+x)$
13	e^x	$\cos 4x$	$- e^{x/2}$
14	$\sqrt{1+e^{-x}}$	$e^{x/3}$	$\sin^2(3x)$
15	$1/(1+x+x^2)$	$\cos(x/10)$	$\cos(x/10)$

Задание 6. Аппроксимировать экспериментальные данные из таблиц значений x_i и y_i функцией вида:

$$f(x) = e^{u_0 + u_1 \cdot x + u_2 \cdot x^2}.$$

Параметры вектора *u* найти с помощью функции *genfit*. Отобразить графически совокупность точек векторов x_i и y_i и результаты проведенной *нелинейной регрессии общего вида.*

Задание 7. Выполнить сглаживание экспериментальной функции, заданной таблицей значений x_i и y_i с помощью встроенных функций Mathcad: *medsmooth*, *ksmooth* и *supsmooth*. Результаты сглаживания.

ЛАБОРАТОРНАЯ РАБОТА № 5
ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ И ДИФФЕРЕНЦИРОВАНИЕ

Численное интегрирование

Формулы для приближенного вычисления определенных интегралов применяются очень часто. Дело в том, что для большого числа элементарных функций первообразные уже не выражаются через элементарные функции, в результате чего нельзя вычислить определенный интеграл с помощью формулы Ньютона-Лейбница.

Встречаются также и случаи, когда приходится прибегать к формулам приближенного интегрирования даже для таких интегралов, которые могут быть найдены в конечном виде, но такое выражение оказывается слишком сложным. Особенно важны формулы приближенного интегрирования при решении задач, содержащих функции, заданные таблично.

Квадратурные формулы

Наиболее распространенным подходом к численному вычислению интеграла

$$I = \int_a^b f(x)dx$$

является разбиение отрезка $[a, b]$ на n равных частей $a = x_0 < x_1 < \ldots < x_n = b$ с шагом $h = \frac{b-a}{n}$, интерполирование функции $y = f(x)$ на отрезке $[a, b]$ (получение интерполяционного многочлена □ (x)) и замена в (1) интеграла интегральной суммой:

$$I_n = \int_a^b \varphi(x)dx = \sum_{i=0}^{n} A_i \cdot y_i \qquad (2)$$

, I_n □ I.

Соотношения вида (2) называют *квадратурными формулами*.

В простейших случаях в качестве интерполяционного многочлена □ (x) берут ступенчатую, кусочно-линейную или кусочно-параболическую функции, а также полином степени $k = n$ (□ (x) = x^k) для которых квадратурные формулы принимают вид (см. Пример 1 Рисунка 8):

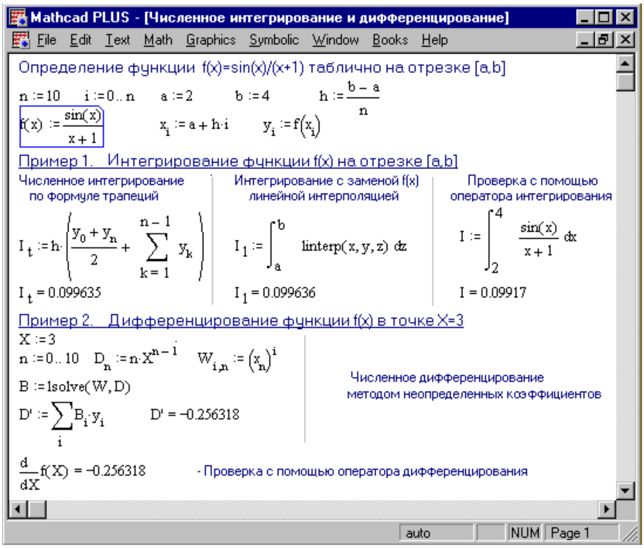

Рис. 8. Численное интегрирование и дифференцирование

Формула прямоугольников:

$$\begin{cases} I_0 = A_0 + A_1 + \ldots + A_n \\ I_1 = A_0 x_0 + A_1 x_1 + \ldots A_n x_n, \\ . \quad . \quad . \quad . \\ I_n = A_0 x_0^n + A_1 x_1^n + \ldots A_n x_n^n, \end{cases} \tag{6}$$

где $I_k = \int_a^b x^k = \dfrac{b^{k+1} - a^{k+1}}{k+1}$, $k = 0, 1, \ldots, n$.

$$I_n = h\sum_{i=1}^{n} f(x_{i-1/2}), \quad x_{i-1/2} = x_{i-1} + \frac{1}{2}h, \qquad i = 1, 2, \ldots, n; \tag{3}$$

формула трапеций:

$$I_n = h \cdot \left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i \right); \tag{4}$$

формула Симпсона (n - четное число):

$$I_n = \frac{h}{3}(y_0 + 4(y_1 + y_3 + \ldots + y_{n-1}) + 2(y_2 + y_4 + \ldots + y_{n-2}) + y_n); \tag{5}$$

метод неопределенных коэффициентов состоит в вычислении определенного интеграла (1) с помощью формулы (2) коэффициенты A_i, которой находятся в результате решения следующей системы уравнений:

Метод Монте-Карло

Во многих задачах исходные данные носят случайный характер, поэтому для их решения должен применяться статистико-вероятностный подход. На основе такого подхода и построен метод статистических испытаний, называемый также методом *Монте-Карло*.

Пусть □ - равномерно распределенная на отрезке [a, b] случайная величина, :

$$\int_a^b f(x)dx = \frac{b-a}{n}\sum_{i=0}^{n} f(\eta_i) \qquad (7)$$

.

Для генерирования последовательности случайных чисел с нормальным законом распределения в Mathcad возможно использовать функцию *rnd*

rnd**(***x***)**	Возвращает равномерно распределенное случайное число между 0 и *x*.

Для реализации метода Монте-Карло удобно использовать функцию *mean*

mean**(***A***)**	Возвращает среднее арифметическое значение элементов массива *A*.

Численное дифференцирование

Численное дифференцирование аналитически или таблично заданной функции $f(x)$ на отрезке $[a, b]$ в точке $x = X$ заключается в замене $f(x)$ интерполяционным полиномом (x), i?iecaiaiua $\frac{d^m \varphi(x)}{dx^m} \approx \frac{d^m f(x)}{dx^m}$ которого можно найти аналитически с помощью соответствующих формул:

$$\frac{d^m f(x)}{dx^m} \approx \frac{d^m \varphi(x)}{dx^m} = \sum_{i=0}^{n} B_i \cdot y_i \quad (8)$$

Метод неопределенных коэффициентов (см. Пример 2 Рис.8.) предполагает использование в качестве интерполяционного многочлена (x) полином степени $k = n$ ($(x) = (X - x_i)^k$), а коэффициенты B_i формулы (8) находятся в результате решения следующей системы уравнений:

$$\begin{cases} D_0 = B_0 + B_1 + \ldots + B_n, \\ D_1 = B_0 x_0 + B_1 x_1 + \ldots + B_n x_n, \\ . \quad . \quad . \quad . \\ D_n = B_0 x_0^n + B_1 x_1^n + \ldots + B_n x_n^n, \end{cases} \quad (9)$$

где $D_k = \left(X^k\right)' = k \cdot X^{k-1}$, $k = 0, 1, \ldots, n$.

Символьное интегрирование и дифференцирование

Для вычисления интегралов (или нахождения первообразных) аналитически заданной функции используется команда **Symbolic□□ Integrate on Variable (Интегрировать по переменной)**. Она возвращает символьное значение неопределенного интеграла по указанной маркером ввода переменной. Выражение, в состав которого входит переменная, является подынтегральной функцией.

Команда **Symbolic□□ Differentiate on Variable (Дифференцировать по переменной)** возвращает символьное значение производной выражения по той переменной, которая указана курсором. Для вычисления производных высшего порядка нужно повторить вычисление необходимое число раз.

Результат символьного преобразования иногда содержит *специальные функции*, которые не являются частью списка *встроенных функций* Mathcad. Вот определения некоторых из них:

□ - константа Эйлера,

Ci(x) = □ + ln(x) + $\int_0^x \frac{\cos(t)-1}{t}\,dt$, Si(x) = $\int_0^x \frac{\sin(t)}{t}\,dt$,

Chi(x) = □ + ln(x) + $\int_0^x \frac{\cosh(t)-1}{t}\,dt$, Shi(x) = $\int_0^x \frac{\sinh(t)}{t}\,dt$.

Порядок выполнения лабораторной работы № 5

Задание 1. Определить функцию $f(x)$ таблично, вычислив значения $y_i = f(x_i)$ в точках $x_i = a + h\,i$, $i = 0, 1, ..., 8$,

$h=(b - a)/8$ на отрезке $[a, b]$.

Варианты задания 1

№ варианта	$f(x)$	$[a, b]$	$[c, d]$
1	$1/(\mathrm{tg}2x+1)$	[0.4, 0.8]	[2, 2.1]
2	$\cos 3x/(1-\cos 3x)_2$	[0.8, 1.6]	[-1, -0.9]
3	$1/(x\sqrt{x^3+4})$	[0.18, 0.98]	[0.5, 0.6]
4	$\sin x/(1+\sin x)$	[0.8, 1.6]	[2, 2.1]
5	$x^2\ \lg(x+2)$	[0, 0.4]	[1.5, 1.6]
6	$x^2\ \mathrm{arctg}(x/3)$	[0.8, 1.6]	[1, 1.1]
7	$e^{2x}\sin 3x$	[0.4, 1.2]	[2, 2.1]
8	$\mathrm{ctg}2x/(\sin 2x)_2$	[0.8, 1.2]	[1, 1.1]
9	$(x + 1) \sin x$	[1, 5]	[1, 1.1]
10	$5x + x \lg x$	[0.2, 1]	[1.3, 1.4]
11	$(2x + 3) \sin x$	[0.4, 1.2]	[0.5, 0.6]
12	$\cos x/(2x+5)$	[0.4, 1.2]	[1, 1.1]
13	$1/(1 + x + x^2)$	[0, 4]	[2, 2.1]

14	$(1+x)/(2+x)$	[0.4, 0.8]	[1.5, 1.6]
15	$\sqrt{1+e^{-x}}$	[0.4, 1.2]	[0.5, 0.6]

Задание 2. Вычислить интеграл $\int_a^b f(x)dx$:

- с помощью встроенного *оператора интегрирования*;
- по формуле *прямоугольников*;
- по формуле *Симпсона*;
- с помощью встроенного *оператора интегрирования* и интерполяцией табличной функции *кубическим сплайном* (функции *cspline* и *interp*);
- методом *неопределенных коэффициентов* для численного интегрирования.

Задание 3. Вычислить интеграл $\int_a^b f(x)dx$ методом *Монте-Карло*. Для этого необходимо:

- определить диапазон случайных чисел, например *j*: = 0..1000;
- определить с помощью функции *rnd* равномерно распределенную случайную величину □ $_j$ на отрезке интегрирования [*a*, *b*];
- создать вектор $F_j = f($□$_j)$;
- с помощью функции *mean* вычислить интеграл.

Задание 4. Найти первообразную аналитически заданной функции $f(x)$, используя команду **Symbolic □ Integrate on Variable**.

Задание 5. Вычислить значения первой и второй производных функции $f(x)$ в точке $X = c$:

- с помощью операторов дифференцирования Mathcad;
- *методом неопределенных коэффициентов* для численного дифференцирования. Определить функцию $f(x)$ таблично, вычислив значения $y_i = f(x_i)$ в точках $x_i = c + h\cdot i$, $i = 0, 1, ..., 10$, $h = 0.01$ на отрезке $[c, d]$.

Задание 6. Определить символьное значение первой и второй производных $f(x)$, используя команду **Symbolic → Differentiate on Variable**.

ЛАБОРАТОРНАЯ РАБОТА № 6 РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Решение обыкновенных дифференциальных уравнений (ОДУ) широко применяется в практике научно-технических расчетов. Хотя линейные ОДУ могут иметь решения в виде специальных функций, многие физические системы нелинейны и описываются нелинейными ОДУ, не имеющими аналитического решения. В этом случае приходиться использовать численные методы решения ОДУ.

Чтобы решить ОДУ, необходимо знать значения зависимой переменной и (или) производных при некоторых значениях независимой переменной. Если эти дополнительные условия задаются при одном значении независимой переменной, то такая задача называется *задачей Коши*. Если же условия задаются при двух или более значениях независимой переменной, то задача называется *краевой*.

Задача Коши

Задачу Коши можно сформулировать следующим образом: пусть дано ОДУ:

$$\frac{dy}{dx} = f(x, y)$$

и начальное условие $y(x_0) = y_0$. Требуется найти функцию $y(x)$, удовлетворяющую как указанному уравнению, так и начальному условию.

Численное решение задачи Коши состоит в построении таблицы приближенных значений $y_1, y_2, ..., y_n$ решения уравнения $y(x)$ в точках $x_1, x_2, ..., x_n$. Чаще всего $x_i = x_0 + ih$, $i = 0, 1, ..., n$, где h - шаг приращения переменной x, n - число интервалов решения с шагом h.

Рассмотрим здесь две группы численных методов решения задачи Коши: *одношаговые* и *многошаговые*.

Одношаговые методы

Одношаговые методы - это методы, в которых для нахождения следующей точки на кривой $y = f(x)$ требуется информация лишь об одном предыдущем шаге. Простейшим из одношаговых методов является метод Эйлера:

$$y_{i+1} = y_i + f(x_i, y_i)h, \quad i = 0, 1, ..., n - 1. \qquad (2)$$

Метод Эйлера имеет невысокую точность (порядка h).

Для достижения более высокой точности (порядка h^4) используют *метод Рунге-Кутта* четвертого порядка:

$$y_{i+1} = y_i + \frac{k_0 + 2\cdot k_1 + 2\cdot k_2 + k_3}{6}, \qquad (3)$$

где

$$k_0 = h\cdot f(x_i, y_i),$$
$$k_1 = h\cdot f\left(x_i + \frac{h}{2}, y_i + \frac{k_0}{2}\right),$$
$$k_2 = h\cdot f\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right),$$
$$k_3 = h\cdot f(x_i + h, y_i + k_2).$$

Многошаговые методы

В *многошаговых методах* для отыскивания следующей точки кривой *y = f(x)* требуется информация *более чем об одной* из предыдущих точек.

Пусть найдены значения $y_{i-3}, y_{i-2}, y_{i-1}, y_i$ в четырех последовательных точках. При этом имеются также вычисленные ранее значения правой части уравнения (1) $f_{i-3}, f_{i-2}, f_{i-1}, f_i$. Тогда схему *метода Адамса* можно

$$y_{i+1} = y_i + h\cdot f_i + \frac{h}{2}\cdot\Delta f_i + \frac{5\cdot h}{12}\cdot\Delta^2 f_i + \frac{3\cdot h}{8}\cdot\Delta^3 f_i, \qquad (4)$$

$i = 3, 4, ..., n - 1$.

представить в виде:

где конечные разности в точке x_i имеют вид:

$$\Delta f_i = f_i - f_{i-1},$$
$$\Delta^2 f_i = f_i - 2 f_{i-1} + f_{i-2}, \qquad (5)$$
$$\Delta^3 f_i = f_i - 3 f_{i-1} + 3 f_{i-2} - f_{i-3}.$$

Решение задачи Коши средствами Mathcad

Инструментарий для решения ОДУ (систем ОДУ) различного порядка в Mathcad представлен широким спектром встроенных функций, работа одной из которых (*rkfixed* - метод Рунге-Кутта (*rk*) четвертого порядка с фиксированным (*fixed*) шагом интегрирования) показана на Рисунке 9.

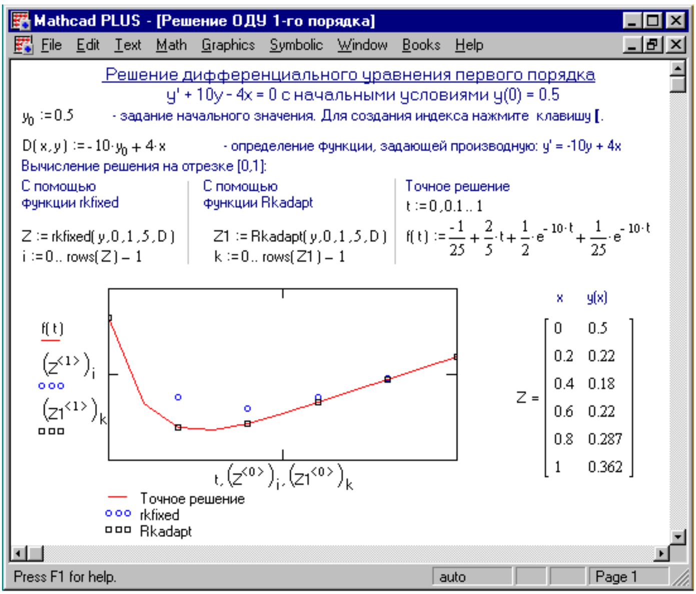

Рис.9. Решение ОДУ 1-го порядка

rkfixed**(***y, a, b, n, D***)**	Возвращает матрицу с *p* + 1 столбцами и *n* + 1 строками (*p* - количество уравнений или порядок уравнения, *n* - число шагов на интервале [*a*, *b*]) - таблицу решений системы: первый столбец - это значения аргумента *x*, а последующие столбцы - значения ординат решения. *y* - вектор начальных условий размерности *n*. *D(x, y)* - функция-вектор из *n* элементов, содержащая первые производные неизвестных функций.

Можно решить задачу более точно (более быстро), если уменьшить шаг *h* там, где производная меняется быстро, и увеличить шаг там, где она ведет себя более спокойно. Для этого предусмотрена функция *Rkadapt* (*adaption* - адаптация). Аргументы и матрица, возвращаемая функцией *Rkadapt*, такие же, как при *rkfixed* (см. Рисунок 9). Решение системы ОДУ показано на Рисунке 11 (Пример 2).

Краевые задачи

Краевая задача формулируется следующим образом: пусть на отрезке [*a, b*] требуется найти решение дифференциального уравнения (для простоты изложение будем вести на примере ОДУ второго порядка):

$$\frac{d^2y}{dx^2} = f(x, y, y') \qquad (6)$$

, при граничных условиях *y(a) = A, y(b) = B*.

В этом случае Mathcad предлагает использовать функцию *sbval*, чтобы найти недостающие начальные условия в точке *a*.

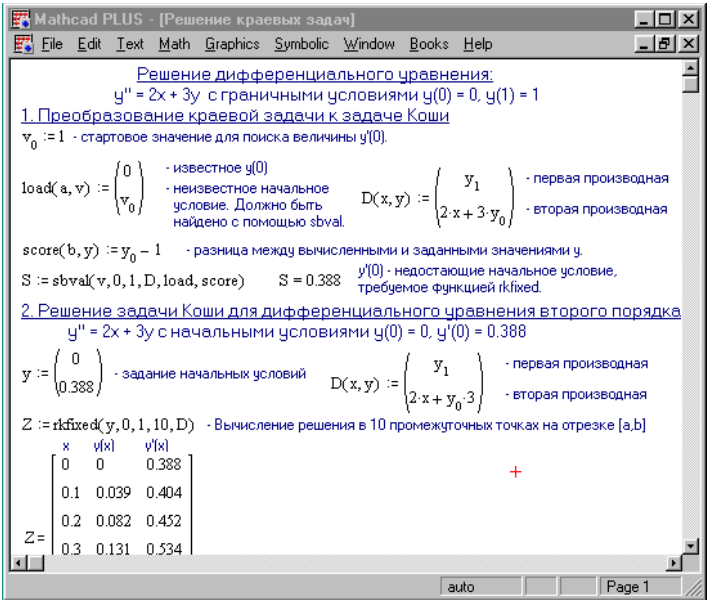

Рис.10. Решение краевой задачи

Sbval(*v, a, b , D, load , score*)	Возвращает вектор, содержащий недостающие начальные условия в точке *a*. Вектор *v* задает начальные приближения, *a, b* - граничные точки интервала решений, *D*(*x, y*) - функция-вектор с первыми производными неизвестных функций. *load*(*a, v*) - функция-вектор, возвращающая значение начальных условий в точке *a*. *score*(*b, y*) - функция-вектор, каждый элемент которого содержит разность между начальным условием заданным в точке *b*, и значением искомого решения в этой точке.

После того, как эти недостающие начальные условия будут получены, можно решать обычную задачу с начальными условиями - *задачу Коши*, используя любую из функций, описанных выше (Рисунок 9).Пример решения краевой задачи показан на Рисунке 10.

Символьное решение линейных дифференциальных уравнений

Для получения аналитического решения линейных ОДУ в Mathcad необходимо выполнить следующие действия (Пример 1 Рисунка 11):

> Если вы работаете с пакетом Mathcad 5.0, не забудьте предварительно выполнить команду **Symbolic Load Symbolic Processor** для загрузки символьного процессора.. Пропустите этот пункт, если вы работаете с пакетом Mathcad 6.0.

- Напечатать исходное уравнение, используя операторы дифференцирования и комбинацию клавиш **[Ctrl]=** для печати символа =.
- Отметив независимую переменную, выполнить прямое преобразование Лапласа **Symbolic Transforms Laplace Transform (Преобразование Лапласа).** Результат для ОДУ выше 1-го порядка будет помещен в буфер обмена. Вызовите его нажав клавишу **F4**.
- По результатам преобразования Лапласа “вручную” составить алгебраическое уравнение, приняв обозначения $L = laplace(y(t), t, s)$, $C1 = y(0)$ и $C2 = diff(y(0), 0)$.
- Решить составленное алгебраическое уравнение относительно переменной L, используя команду

Symbolic □ Solve for Variable (Решить относительно переменной).

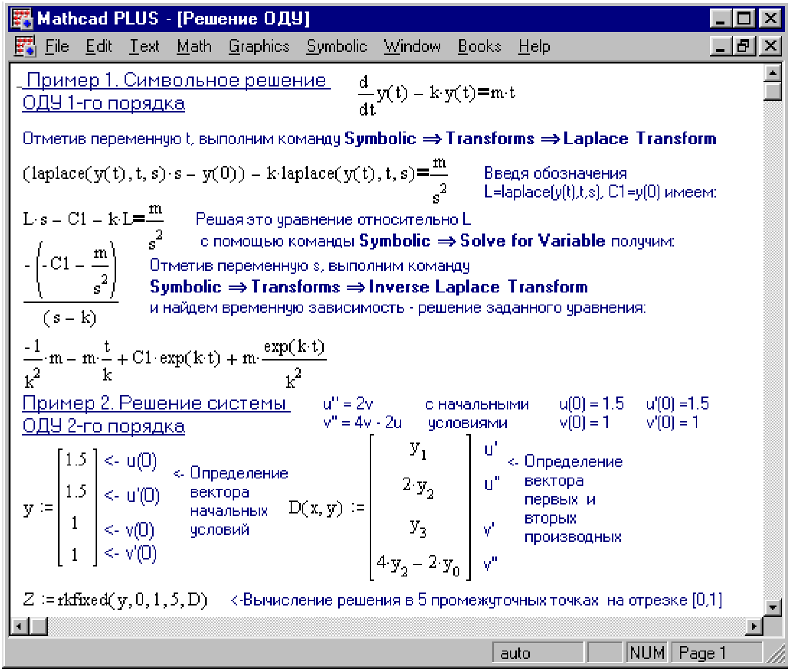

Рис.11. Некоторые возможности решения ОДУ в Mathcad

- Отметить переменную *s* и произведя обратное преобразование Лапласа **Symbolic □ Transforms □ Inverse Laplace Transform (Обратное преобразование Лапласа)** получить решение заданного ОДУ в виде временной зависимости.

Порядок выполнения лабораторной работы № 6

Задание 1. Решить задачу Коши: $\dfrac{dy}{dx} = f(x, y)$, $y(0) = 1$

с шагом $h = 0.1$ на отрезке [0, 1]:

- методом *Эйлера*;
- методом *Рунге-Кутта* (коэффициенты k_i задать как функции от x и y);
- методом *Адамса*;
- используя функцию *rkfixed*.

Варианты задания 1

№ варианта	$f(x, y)$	№ варианта	$f(x, y)$	№ варианта	$f(x, y)$
1	$x + y$	**6**	$2y - \cos 2x$	**11**	$2y + 3e^{-x}$
2	$2x^2 + 2y$	**7**	$y - e^{x/2} + 2$	**12**	$y/2 - e^{-x}$
3	$e^x - 3y$	**8**	$3y - 2\sin x$	**13**	$y + (\cos x)/3$
4	$y - \sin x$	**9**	$e^{2x} - y$	**14**	$y - 4x + 5$
5	$y/3 - x^2$	**10**	$2\sin x + y$	**15**	$2x - y/3 - e^x$

Задание 2. Построить графики решений, полученных методами *Эйлера, Рунге-Кутта, Адамса* и с помощью функции *rkfixed.*

Вычислить в точке $x = 1$ относительную погрешность для каждого метода.

Задание 3. Найти аналитическое (точное) решение ОДУ из задания 1 с помощью преобразований Лапласа (команды

Symbolic □ **Transforms** □ **Laplace Transform** и **Inverse Laplace Transform**).

Задание 4. Решить задачу Коши для системы ОДУ при заданных начальных условиях на отрезке [0, 2] с шагом h = 0.2. Решать с помощью функции *rkfixed*. Построить графики функций *u(t)* и *v(t).*

Варианты задания 4

№ варианта	Система ОДУ	Начальные условия				№ варианта	Система ОДУ	Начальные условия			
		$u(0)$	$u'(0)$	$v(0)$	$v'(0)$			$u(0)$	$u'(0)$	$v(0)$	$v'(0)$
1	$\begin{cases} u'' = 2v + u \\ v'' = 4v - 2u \end{cases}$	1.5	1.5	1	1	**9**	$\begin{cases} u'' = 1/2 + v \\ v'' = 4 - u + t \end{cases}$	2	0	-1	1
2	$\begin{cases} u'' = -v + 3u \\ v'' = v - 2u \end{cases}$	-1	1	-1.5	3	**10**	$\begin{cases} u'' = -v + t \\ v'' = v + 3u \end{cases}$	-1	2	-1.5	0
3	$\begin{cases} u'' = 2v - u \\ v'' = 4v + u \end{cases}$	1.5	1.5	1	1	**11**	$\begin{cases} u'' = v - u - t \\ v'' = 2v + u \end{cases}$	1.5	1.5	-1	-1
4	$\begin{cases} u'' = 5v \\ v'' = v + 2u + t \end{cases}$	1	1.5	0	2	**12**	$\begin{cases} u'' = 5v + t \\ v'' = 3v + u \end{cases}$	-1	1.5	0	-2
5	$\begin{cases} u'' = v + u + t \\ v'' = v + 2u - t \end{cases}$	0.5	1.5	-1	2	**13**	$\begin{cases} u'' = v + u \\ v'' = v + u - t \end{cases}$	-0.5	1	-1	2
6	$\begin{cases} u'' = 2v + u + t \\ v'' = 4v \end{cases}$	0.5	2	1	2	**14**	$\begin{cases} u'' = 2v - u \\ v'' = 4v + t \end{cases}$	0	-2	0	2
7	$\begin{cases} u'' = -v + t \\ v'' = 5v - 7u \end{cases}$	5	5	-1	1	**15**	$\begin{cases} u'' = v - 2t \\ v'' = v + 3u \end{cases}$	3	3	-1	1
8	$\begin{cases} u'' = v - 5u \\ v'' = 2v + u + t \end{cases}$	1.5	1	3	1						

Задание 5. На отрезке [a, b] с использованием функций *load, score* и *sbval* преобразовать краевую задачу:

$$\frac{d^2y}{dx^2} = f(x, y, y') \text{ при граничных условиях } y(a) = A,\ y(b) = B$$

к задаче Коши и найти решение заданного ОДУ в 10 промежуточных точках с помощью функции *rkfixed*.

Варианты задания 5

№ варианта	$f(x, y, y')$	Граничные условия			
		a	b	$y(a)$	$y(b)$
1	$e^x y + \cos x$	1	2	0	0
2	$y \sin x + e^{-x}$	2	3	1	0
3	$y \cos x + \mathrm{tg}\, x$	0	1	0	0.45
4	$x^3 y + \cos x$	0	1	1	0
5	$x + e^x y/(1 - x)$	2	4	1	0.14
6	$x^2 y + 1/(1 + x)$	1	3	0	0.17
7	$y \cos x + \cos^2 x$	1	2	0	0
8	$(2 + x)\, y + \mathrm{arctg}\, x$	0	3	0	0.22
9	$(5 - x)\, y + x$	2	4	0	-1.2
10	$e^{-x} y + 2 e^{-x}$	0	1.5	2.4	0
11	$e^{-x} y/x + x$	-3	-2	3	0
12	$(x^2 + 1/x)\, y + 1/x^2$	2	3	0	0
13	$(10 - x)\, y + x$	-1	0	2	0
14	$y/x^2 + x$	1	3	1.5	0
15	$y \ln x + 1 + x$	7	8	0	0

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Омельченко В.П. Математика: учеб. пособие / В.П. Омельченко, Э.В. Курбатова.- Ростов н/Д: Феникс, 2008.- 380 с.

2. Пантина И.В. вычислительная математика / И.В. Пантина,А.В. Синчуков.-М.: Маркет ДС, 2010. - 176с.

1. Амосов А.А. Вычислительные методы для инженеров / А.А. Амосов, Ю.А. Дубинский, Н.В. Копченова.- М.: Высш. шк.,1994.- 544 с.

2. Турчак Л.И. Основы численных методов / Л.И. Турчак. - М.: Наука, 1987. – 320 с.

3. Демидович Б.Н. Основы вычислительной математики / Б.Н. Демидович, И.А. Марон. - М.: Высш. шк., 1994. - 172 с.

4. Боглаев Ю.П. Вычислительная математика и программирование / Ю.П. Боглаев.- М.: Высш. шк., 1990. – 544с.

5. Семенов М.П., Катрахова А.А. Жучкова В.В. Основы численных методов / М.П.Семенов, А.А. Катрахова, В.В. Жучкова.- Воронеж: Изд-во ВГТУ, 1997. - 62 с.

6. Ушаков Д.М. Введение в математические основы САПР / Д.М. Ушаков, В.П.Корячко, И.П.Норенков.-

7. Федорков Е.Д.Численные методы : учеб. пособие / Е. Д. Федорков, А. И. Бобров. – Воронеж : ВГТУ, 2004. - 164 с. - 33-00.

8. Федорков, Е.Д. Вычислительная математика: Учеб. пособие / Е. Д. Федорков, А. И. Бобров. - Воронеж: ВГТУ, 2006. - 168 с.

СОДЕРЖАНИЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторной работы
по дисциплине «Математическое обеспечение САПР»
для студентов направления
230100.62 «Информатика и вычислительная техника»
(профиль «Системы автоматизированного проектирования
в машиностроении»)
заочной формы обучения

Составители:

Пименов Денис Николаевич

Пак Алла Анатольевна

В авторской редакции

Компьютерный набор А.А. Пак

Подписано к изданию . 15.12 .2014.

Уч.-изд. л. 3,6 . «С»

ФГБОУ ВПО «Воронежский государственный технический
университет»
394026 Воронеж, Московский просп., 14