Содержание

Введение

Исторически, в процессе развития программирования возникло отдельное направление – языки программирования для проведения математических расчетов. Первымтаким языком стал Fortran (от formula translator). Но математические расчеты по-прежнему оставались сложным делом: применение компьютеров внесло новые трудности - прежде чем начинать расчеты пользователь должен был освоить основы программирования, изучить один, а то и несколько языков программирования и освоить достаточно специфические численные методы расчетов. Все это не придало математическим расчетам большей привлекательности.

Положение стало меняться к лучшему после разработки специализированных программных комплексов для автоматизации математических. и научно-технических расчетов. К таким комплексам относятся пакеты программ MatLab, Maple, Mathematica др.

Данное пособие посвящено описанию одного из самых мощных в удобных интегрированных пакетов программ для автоматизации математических и научно-технических расчетов - системы MathCad.

MathCad содержит текстовый редактор, мощный вычислитель и очень простой в применении графический процессор. Это позволяет готовить документы, по виду очень напоминающие статьи или разделы из книг. Язык общения с пользователем в системе MathCad предельно приближен к обычному математическому языку. Вычислитель системы содержит множество математических функций - от всем известных элементарных функций до специфических, таких как быстрое преобразование Фурье или сплайн-интерполяция.

Система MathCad универсальна - она может применяться везде, где используются математические методы анализа, расчета и моделирования. Ее с одинаковым успехом можно использовать как для проведения школьных расчетов, так и сложных расчетов, вполне достойных включения в серьезные диссертации и труды по математике. При этом центр тяжести расчетов перемещается с вопросов программирования компьютера на вопросы естественного математического описания алгоритмов решения математических и научно-технических задач.

Такое смещение акцентов позволяет упростить работу учащихся, а также оптимизировать учебное время, так как большее внимание можно обращать на методы моделирования, не затрачивая усилий на ручные расчеты и построение графиков.

Основные приципы работы в среде MathCad

Назначение системы

MathCad является интегрированной системой программирования, ориентированной на проведение математических, инженерно-технических, статистических и экономических расчетов. MathCad содержит текстовый редактор, вычислитель и графический процессор.

Текстовый редактор служит для ввода и редактирования текстов. Тексты являются комментариями, и входящие в них математические выражения не исполняются. Текст может состоять из слов, математических выражений и формул, различных спецзнаков. Вычислитель системы MathCad обеспечивает вычисления по сложным математическим формулам, имеет обширный набор встроенных математических функций, обеспечивает вычисления рядов, сумм и произведений. определенных интегралов и производных, позволяет решать линейные и нелинейные уравнения, проводить минимизацию функций. Имеется обширный набор векторных и матричных операций, возможность работы с комплексными числами. В вычислитель входят в такие мощные средства как линейная в сплайн-интерполяция, регрессия, быстрое преобразование Фурье, статистические расчеты. Легко можно менять разрядность чисел и погрешность итерационных методов.

Графический процессор служит для создания графиков. Простые графики некоторых функций пользователь может строить буквально в первые секунды знакомства с системой. По мере приобретения навыков работы с графическим процессором можно легко освоить и .другие графические средства - графики в логарифмическом масштабе, масштабные сетки с любым числом делений, построение линий, отмеченных точками, прямоугольниками и ромбиками, гистограмм и др. Графика ориентирована на решение типичных математических задач. Возможно быстрое изменение размеров графиков, наложение их на текстовые надписи и перемещение в любое место документа.

Простейшие вычисления

Простейшие вычисления можно выполнить, используя знак вывода результатов вычислений = (равенства), что соответствует схеме

Выражение =

В левой части равенства могут стоять любые математические выражения, содержащие встроенные в систему функции.

Документом в системе MathCad называется полное и математическое описание алгоритмов решения тех или иных задач. Документ в свою очередь состоит из блоков, т.е. отдельных частей. Блоки могут быть трех типов- текстовые, вычислительные и графические. Каждый блок занимает на экране некоторое пространство, ограниченное прямоугольной областью.

MathCad реализует вычисления слева направо и сверху вниз. Правильный порядок выполнения блоков- основа правильного функционирования системы. Например, если в некотором блоке содержатся операции, требующие данных из другого блока, то последний обязательно должен выполняться первым. И его местоположение должно предшествовать местоположению данного блока. Иная ситуация приведет к появлению ошибки.

Сигнал ошибки системе имеет вид надписи, заключенной в прямоугольник. От него отходит черта, указывающая на место ошибки. Таким образом, место ошибки легко найти.

Алфавитный указатель англоязычных сообщений об ошибках

array size mismatch	несоответствие размера массива
cannot be defined	не может быть определено
cannot take subscript	не содержит верхних (нижних) индексов
definition stack overflow	переполнение стека определений
did not find solution	решение не найдено
dimension to non real power	размерность массива – не целое число
domain error	ошибка области определения
duplicate	дублирование
equation too large	слишком большое выражение
error in constant	ошибка в константе
error in list	ошибка в списке
error in solve block	ошибка в блоке
file error	ошибка файла
file not found	файл не найден
illegal array operation	неверная операция с массивом
illegal context	неверный контекст
illegal factor	неверный множитель
illegal function name	неверное имя функции
illegal ORIGIN	неверное употребление ORIGIN
illegal range	неправильный диапазон
illegal tolerance	некорректная точность аппроксимации
incompatible units	несовместимые единицы
index out of bounds	индекс вне границ
interrupted	прервано
invalid order	неверный порядок
list too long	длинный входной список
misplased comma	неуместная запятая
missing operand	пропущенный операнд
missing operator	пропущенный знак операции
must be 3-vector	должно быть трехмерным вектором
must be array	должно быть массивом
must be dimensionless	должно быть безразмерным

must be increasing	должно быть возрастающим
must be integer	должно быть целым
must be nonzero	должно быть ненулевым
must be positive	должно быть положительным
must be range	должен быть диапазон
must be real	должно быть вещественным
must be scalar	должно быть скаляром
must be vector	должно быть вектором
nested solve block	следующий блок решения
no matching Given	нет соответствующего Given
no scalar value	не скалярная величина
not a name	не является именем
not converting	не конвертируется
only one array allowed	допустим только один массив
overflow	переполнение
significance lost	потеряны значащие цифры
singularity	деление на нуль
stack overflow	переполнение стека
subscript too large	слишком большой нижний индекс
too few arguments	слишком мало аргументов
too few constraints	слишком мало ограничений
too few elements	слишком мало элементов
too few subscripts	мало нижних индексов
too large to display	слишком велико, чтобы отобразить
too many arguments	слишком много аргументов
too many constraints	слишком много ограничений
too many points	слишком много точек
too many subscripts	слишком много индексов
undefindet	не определено
unmatched parenthesis	дисбаланс скобок
wrong size vector	неверный размер вектора

Система MathCad имеет ряд режимов работы. При первом включении устанавливается автоматический режим расчета. Такой режим позволяет выполнять вычисления сразу по мере ввода и редактирования документа. Однако, это может приводить к снижению быстродействия системы, поскольку на вычисления, нередко довольно сложные, система вынуждена затрачивать время. В ручном режиме ввод и редактирование документа происходят без выполнения вычислений. Реакция системы на действия пользователя становится более быстрой и редактирование оказывается более удобным. Для перехода к режиму вычислений при этом достаточно нажать функциональную клавишу F9. Вычисления охватят те блоки, которые расположены сверху от текущего положения курсора.

Текстовые блоки

Для ввода текстов, т.е. создания текстовых блоков документов, достаточно ввести знак "(кавычки) или воспользоваться меню или панелью инструментов.

Внутри текстового блока можно пользоваться стандартными приемами редактирования текста - перемещением курсора вверх и вниз, вправо и влево, уничтожением и вставкой символов. Как отмечалось, математические выражения в текстовом блоке играют роль комментариев и не выполняются. Например, надпись "Вычисляется sin(1)= не приведет к вычислению значения sin(1).

Алфавит MathCad

Алфавит системы MC содержит строчные и прописные латинские буквы, арабские цифры, ряд греческих букв и специальных знаков. С их помощью задаются имена встроенных функций и операторов, а также идентификаторы - имена вводимых пользователем переменных, констант и функций.

Идентификаторы должны начинаться с буквы, и их имена должны быть уникальными, т.е. неповторяющимися. В состав идентификаторов могут входить цифры и имена встроенных функций, но при наличии отличительных дополнений. Следующие идентификаторы являются допустимыми:

x, X, Alfa, my_function, U51;F_Cos, и т. д.

А вот эти идентификаторы недопустимы:

1U - начинается с цифры,

sin - совпадает с именем встроенной функции,

альфа - содержит не латинские буквы.

Греческие буквы вводятся либо при помощи панели инструментов, либо с клавиатуры. Для того чтобы ввести символ с клавиатуры, нужно нажать одну из клавиш с латинским символом, приведенных в таблице, а затем комбинацию клавиш Ctrl+G.

a →α	b →β	d →δ	e →ε	f →ϕ	G →Γ	h →η	→∞	l →λ
N →N	O →Ω	p →π	Q →θ	r →ρ	s →σ	t →τ	→υ	w →ω

В текстовых блоках могут использоваться знаки.

В состав алфавита системы MC входит и множество специальных математических символов, например, знак суммы ∑, интеграла и т.д. Ввод этих знаков производится также при наборе определенных комбинаций клавиш или панели инструментов. Иногда знаки на клавишах не совпадают с теми символами, которые вводятся ими. Например, знак двоеточия вводит символ присваивания :=.

Константы, переменные и равенства

Часто используются циклические вычисления, т.е. повторяющиеся заданное число раз. Примером мояет служить необходимость вычисления некоторого числа значений какой-либо функции f(x). Для этого нужно получить ряд значений аргумента x. Циклические вычисления характерны и для итерационных и рекурентных методов.

Для получения цикических вычислений с целочисленной управляющей переменной цикла используются переменные с заданными пределами изменения:

Имя переменной := Ннач .. Нкон

Здесь знак .. создается вводом символа точка с запятой ; , Ннач-начальное значение переменной и Нкон- конечное значение переменной. Если Ннач<Нкон, то шаг изменения переменной будет равен +1, а если Ннач>Нкон, то шаг будет равен -1.

Шаг изменения можно задать любым, используя другую:

Имя переменной := Ннач, Нслед .. Нкон, где Нслед- следующее, .вслед за Ннач,., значение переменной. Шаг в этом случае равен Нслед-Ннач.

Обычно присваивание переменной определенного значения отождествляют с ее равенством этому значению, например, если X=5, то, говорят, что значение переменной X равно пяти. Присваивание в системе MathCad реализуется с помощью знака := (двоеточие с равенством). Практически для этого достаточно ввести знак двоеточия. Итак, если ввести X : 5, то на экране появится X := 5.

Константой в системе MathCad называют числовые значения, величина которых не меняется - например, константа 5 имеет значение равное пяти в любом месте программы.

Условно к константам можно отнести и предварительно определенные - системные переменные. Это предварительно определенные переменные, значение которых задается в начале загрузки системы. Например:

π = 3.1415927 – число π

e = 2.71828 - основание натурального логарифма,

со "бесконечность"

Встроенные операторы

ОПЕРАТОР	ОБОЗНАЧЕНИЕ	КЛАВИША	ОПИСАНИЕ
Круглые скобки	(X)	`	Группа операторов
Нижний индекс	A_n	[	Возвращение индексированного элемента массива
Верхний индекс	$A^{<n>}$	[Ctrl]6	Выбор столбца из массива А. Возврат вектора.
Векторизация	$\overrightarrow{X}$	[Ctrl]-	Позволяет производить операции по экспрессии X для того, чтобы один элемент заменялся другим. Все векторы или матрицы в X должны быть одинакового размера.
Факториал	n!	!	Возвращает n(n-1)(n-2)… Целое число n не может быть отрицательным.
Транспонирование	A^T	[Ctrl]1	Возвращает матрицы, чьи ряды являются колонками A и чьи колонки являются рядами A. A может быть вектором или матрицей.
Степень	Z^w	^	Возводит Z в степень W.
Степень матрицы, обратная матрица	M^h	^	n^{th} – степень квадратной матрицы M (использование матрицы умножения). n должно быть целым числом. M^{-1} обратно пропорционально M. Другие отрицательные степени являются степенями инверсий. Возврат матрицы.
Отрицание	-X	-	Умножение X на –1
Сумма вектора	$\sum v$	[Ctrl]4	Суммы элементов вектора v; возвращение скаляра.
Квадратный корень	$\sqrt{Z}$ $\sqrt[n]{Z}$	\	Возвращение положительного квадратного корня для положительного Z; абсолютная величина для отрицательного или комплексного Z.
Корень n-й степени		[Ctrl]\	Возвращение n^{th} корня z; возвращает вещественное значение корня всякий раз, когда это возможно.
Модуль числа	|Z|	|	Возвращение $\sqrt{\mathrm{Re}(Z)^2+\mathrm{Im}(Z)^2}$
Размер вектора	|v|	|	Возвращение $\sqrt{v\cdot v}$, если все элементы в v являются вещественными. Возвращение $\sqrt{v\cdot v}$, если элемент в v является комплексным.
Детерминант	|M|	|	Возвращение определителя квадратной матрицы M.
Деление	X/Z $X\cdot Y$	/	Деление выражения X на скаляр Z, не равный нулю. Если X является массивом, то мы делим каждый элемент массива на Z.
Умножение	$u\cdot v$	*	Возвращает произведение X и Y, если как X, так и Y – скаляр. Умножаем каждый элемент Y на X, если Y – массив и X – скаляр. Возвращение точечного произведения (внутреннее произведение), если X и Y – векторы одинакового размера. Выполняем умножение матрицы, если X и Y – подобные матрицы.
Кросс-произведение		[Ctrl]8	Возвращение кросс-произведения (вектор произведения) для третьего элемента векторов u и v.
Суммирование	$\sum_{i=m}^{n} X$	[Ctrl][Shift]4	Произвести суммирование X для i=m,m+1,…,n. X может быть любым выражением. Нет необходимости возводить в степень i, но обычно это делают. m и n должны быть целыми числами.
Произведение	$\prod_{i=m}^{n} X$	[Ctrl][Shift]3	Выполнить итерацию произведения X для i=m,m+1,…,n. X может быть любым выражением. Нет необходимости возводить в степень i, но обычно это делают. m и n должны быть целыми числами.

ФУНКЦИИ. УСЛОВНЫЙ ОПЕРАТОР. МЕТОД ПРЯМОУГОЛЬНИКОВ

1.2 Способы задания функций.

Аналитический способ

При этом способе явно указываются математические действия, которые надо совершить над независимой переменной, чтобы получить значение функции.

Аналитический способ компактен (формула занимает мало места), легко воспроизводим (формулу легко переписать) и наиболее приспособлен к выполнению над функциями математических действий (сложение, умножение и т.п.). Однако он не всегда нагляден (не виден характер зависимости функции от аргумента) и для вычисления значений функции, если они требуются, необходимо произвести ряд выкладок, не всегда простых; да и далеко не все функции возможно и целесообразно записать в виде формулы.

Табличный способ

В табличном способе задания функции ее численные значения задаются с помощью таблицы при определенных дискретных численных значениях аргумента. Таблица имеет вид

x	x0	x1	x2	.....	xn
y	y0	y1	y2	.....	yn

Хорошо известные таблицы логарифмов, тригонометрических функций и т.д. - это примеры табличного задания функций. Таблицы часто получаются в результате обработки эксперимента, когда задаются значениями одной величины и измеряют значения другой.

Определение функций в Mathcad

При первом способе необходимо задать диапазон изменения переменной и собственно саму функциональную зависимость. Для примера зададим функцию y(z)=z2.

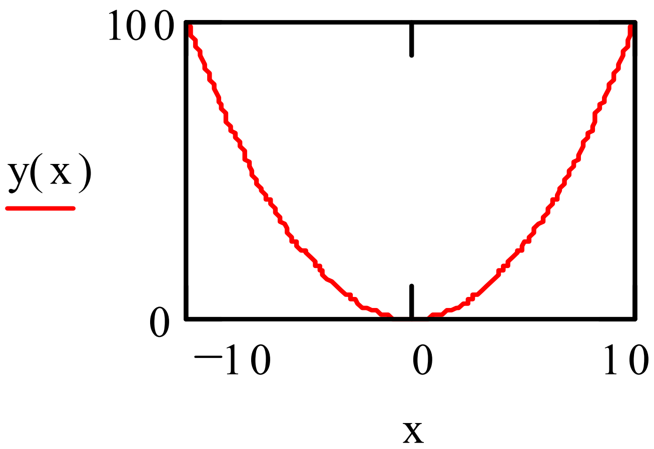
$$x := -10, -9.9 \,..\, 10$$

$$y(x) := x^2 \qquad (1.1)$$

Для отображения заданной функции на шаблоне графика необходимо по горизонтальной оси указать имя аргумента (в нашем случае x), а по вертикальной имя функции (y(x)). Для получения значения функции y(x) при некотором значении аргумента x0 необходимо подставить это значение в выражение, описывающее эту функцию, аналогично тому, как показано ниже. Для вывода результата на экран используется значок =(вычислить).

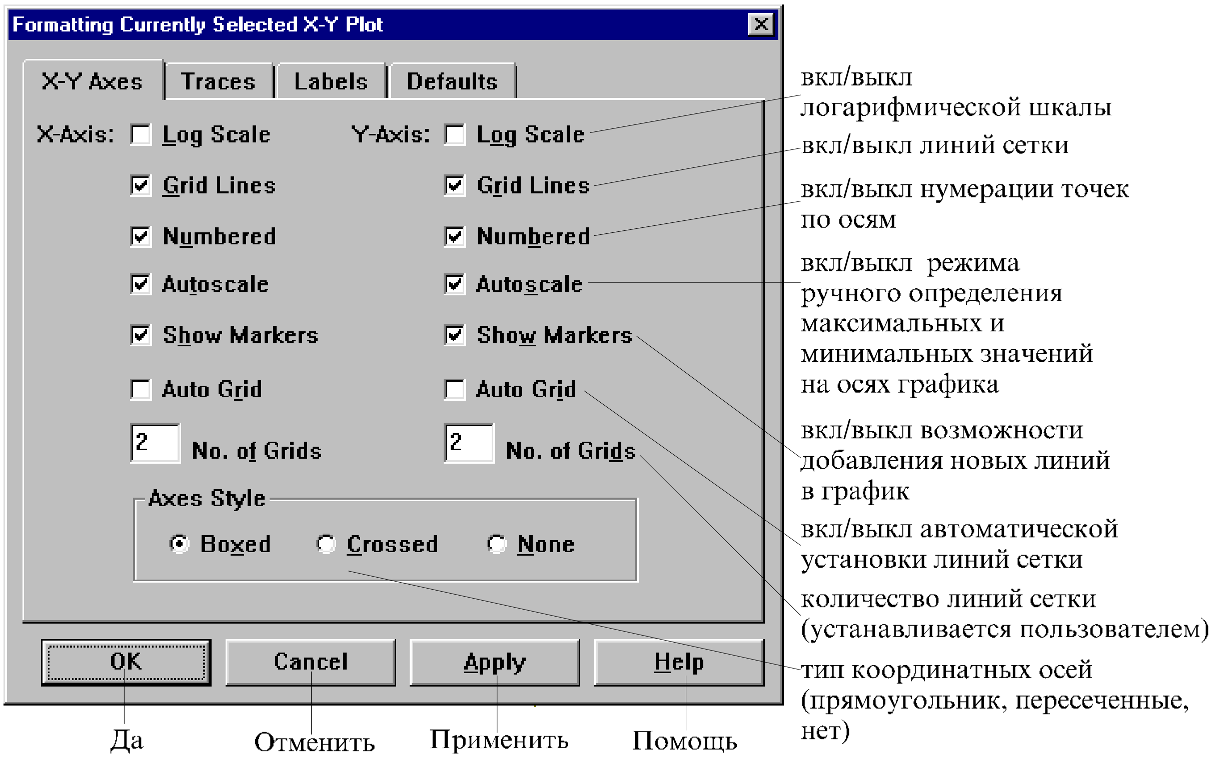
$y(x0)=$

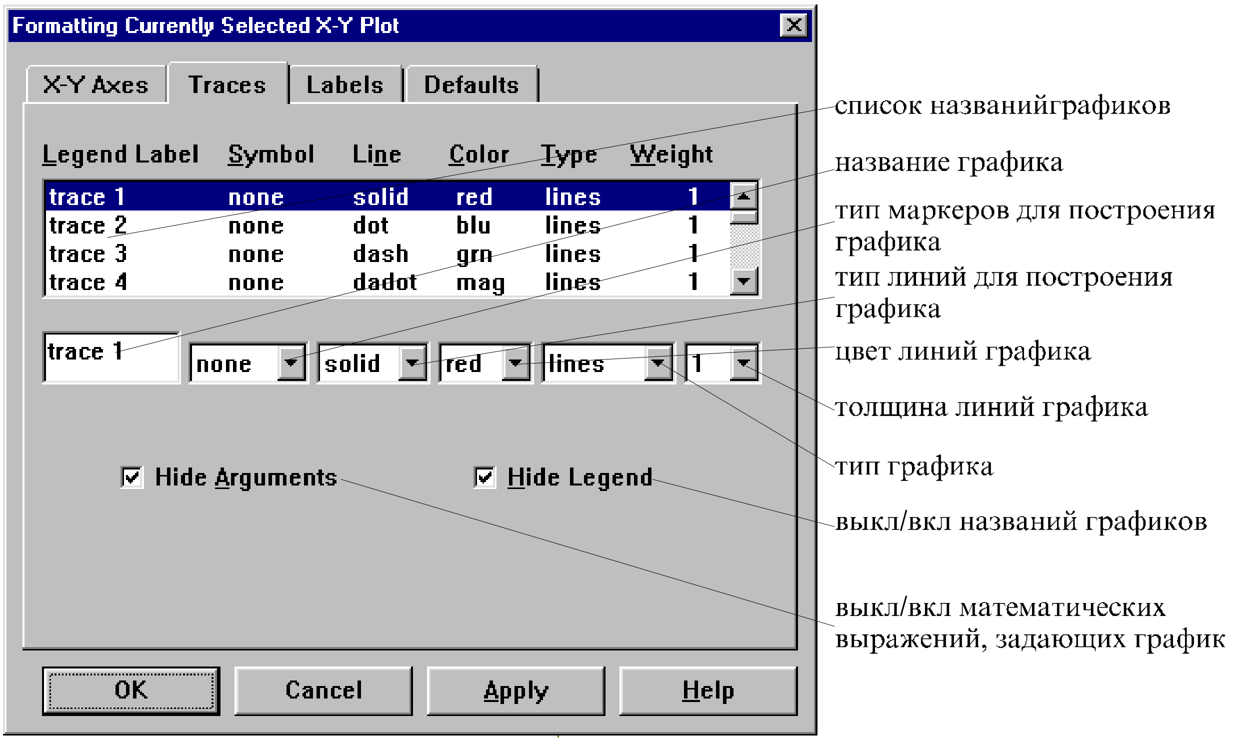
Вне зависимости от диапазона изменения аргумента x, значение функции y(x) может быть вычислено при любом значении x0.

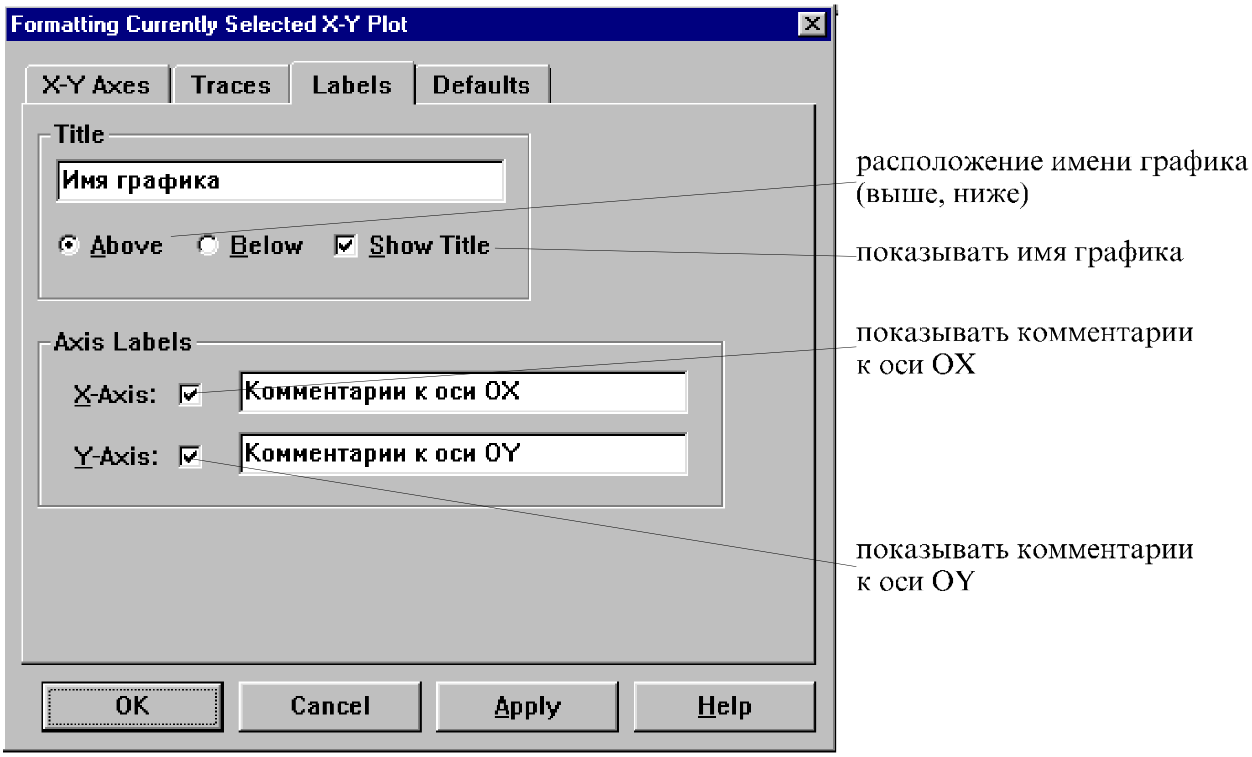


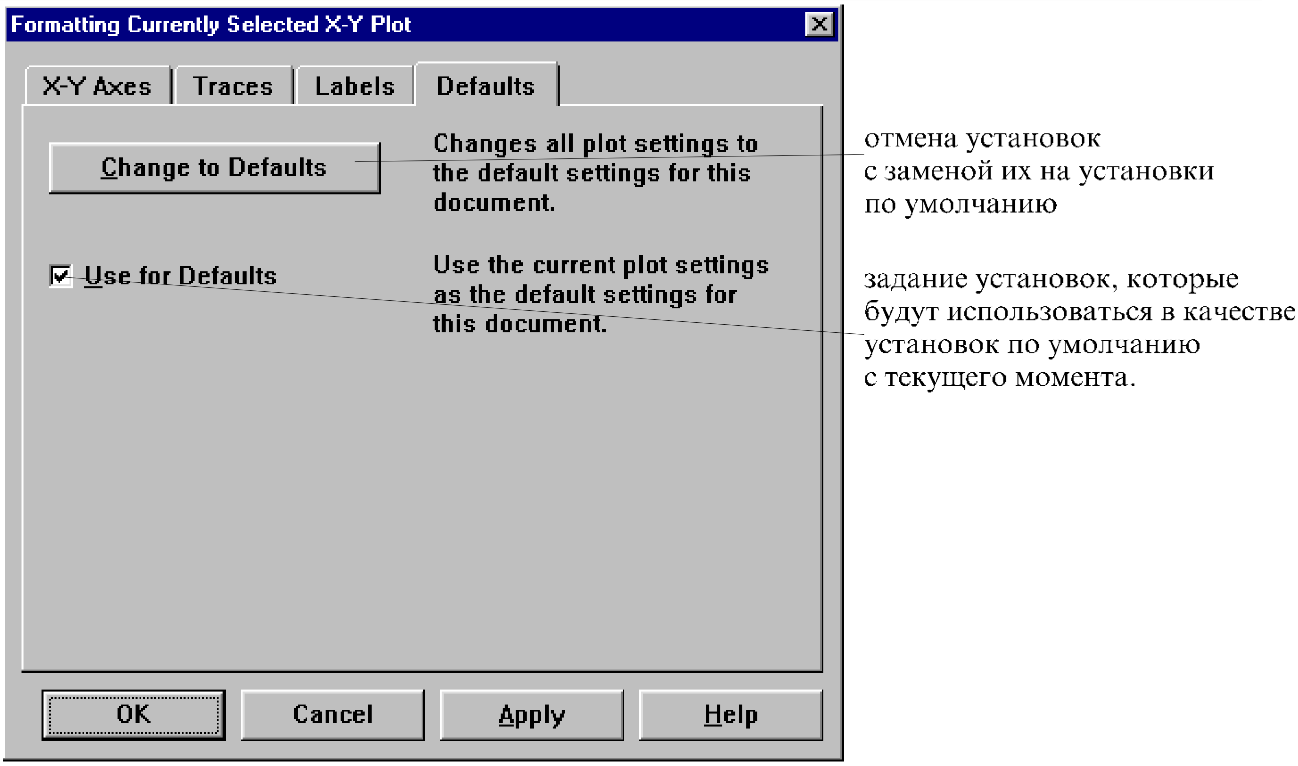
Поле графика создается вводом символа @. С помощью двойного щелчка можно вывести меню форматирования графика.

Меню форматирования графика




Formatting Currently Selected X-Y Plot
X-Y Axes
Traces
Labels
Defaults
Legend Label
Symbol
Line
Color
Type
Weight
trace 1 none solid red lines 1
trace 2 none dot blu lines 1
trace 3 none dash grn lines 1
trace 4 none dadot mag lines 1
trace 1
none
solid
red
lines
1
Hide Arguments
Hide Legend
OK
Cancel
Apply
Help
список названийграфиков
название графика
тип маркеров для построения графика
тип линий для построения графика
цвет линий графика
толщина линий графика
тип графика
выкл/вкл названий графиков
выкл/вкл математических выражений, задающих график

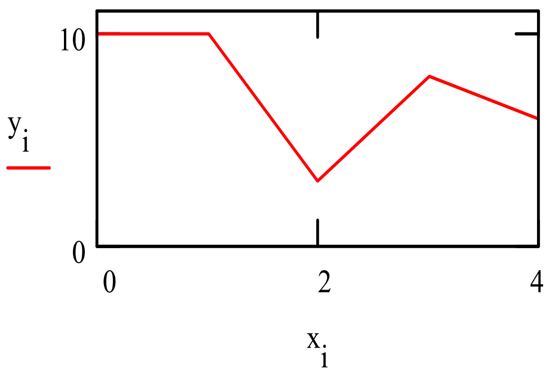

Formatting Currently Selected X-Y Plot
X-Y Axes
Traces
Labels
Defaults
Title
Имя графика
Above
Below
Show Title
Axis Labels
X-Axis:
Комментарии к оси OX
Y-Axis:
Комментарии к оси OY
OK
Cancel
Apply
Help
расположение имени графика (выше, ниже)
показывать имя графика
показывать комментарии к оси OX
показывать комментарии к оси OY



При табличном способе задания функции необходимо задать две таблицы x и y, имеющие вид

$$x := \begin{bmatrix} x0 \\ x1 \\ \ldots. \\ xn \end{bmatrix} \qquad y := \begin{bmatrix} y0 \\ y1 \\ \ldots. \\ yn \end{bmatrix}$$

Для отображения таблиц на шаблоне необходимо указать по горизонтальной и вертикальной осям шаблона имена элементов этих табличек (соответственно x_i и y_i)



Обратите внимание, что имя элемента таблицы содержит непосредственно имя таблицы и порядковый номер элемента в таблице (индекс элемента). Так как на шаблон необходимо вывести не один, а все элементы таблицы, то индекс должен быть переменной величиной, изменяющейся в диапазоне от нуля до величины на единицу меньшей числа элементов таблицы. Переменную, которая отражала бы диапазон изменения индекса, необходимо задать перед использованием шаблона для построения графика. Индексы могут быть только целыми положительными числами или нулем.

Задать таблицу можно, задав в отдельности каждый ее элемент (см. выражение 1.3).

$$x_0 = x0 \qquad y_0 = y0$$
$$x_1 = x1 \qquad y_1 = y1$$
$$\ldots\ldots \qquad \ldots\ldots$$
$$x_n = xn \qquad y_n = yn$$

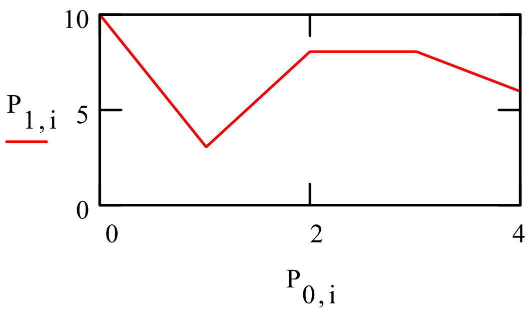
Можно несколько модернизировать этот способ задания функции, записав как значения аргумента, так и значения функции в одной таблице. Для этого необходимо значения аргумента расположить в первой строке таблицы, а значения функции во второй. Тогда табличное задание функции будет иметь вид

$$P := \begin{bmatrix} x0 & x1 & x2 & \ldots. & xn \\ y0 & y1 & y2 & \ldots. & yn \end{bmatrix}$$

или

$$P_{0,0} = x0 \quad P_{0,1} = x1 \quad P_{0,2} = x2 \quad \ldots.. \quad P_{0,n} = xn$$
$$P_{1,0} = y0 \quad P_{1,1} = y0 \quad P_{1,2} = y2 \quad \ldots.. \quad P_{1,n} = yn$$

Первый индекс указывает на номер строки, а второй на номер столбца.



Для отображения данной таблицы на шаблоне необходимо указать следующее: по горизонтальной оси элементы первой строки таблицы, по вертикальной оси - элементы второй строки таблицы. Кроме этого необходимо задать диапазон изменения индекса.

Функцию, заданную аналитическим выражением, можно также представить и в табличном виде. Для этого используется тот же механизм, что и для формирования таблицы с произвольными значениями x и y. При реализации данного способа задания функции необходимо, чтобы аргумент функции и значение индекса элемента были связаны между собой. Причем, эта связь должна быть такой, чтобы каждому значению аргумента соответствовал свой индекс. Таким образом, число значений, принимаемое аргументом, должно быть равно числу значений, принимаемых индексом. Как было отмечено выше, индекс может быть только целым положительным числом или нулем.

Наиболее просто эта задача решается, когда первое значение аргумента равно нулю, и далее он принимает только целые положительные значения.

$$x := 0 \,..\, 10$$

$$y_x := x^2$$

Если аргумент принимает произвольные целые значения, то их нужно преобразовать таким образом, чтобы первое значение индекса было равно нулю. Для этого необходимо к значению индекса прибавить первое значение аргумента, взятое с противоположным знаком.

$$x := -10 \,..\, 10$$

$$y_{x+10} := x^2$$

В случае , когда аргумент задан произвольными значениями, преобразование индекса выглядит аналогично тому, как это показано в выражении

$$x := -10, -9.9 \,..\, 10$$

$$y_{(x+10)\cdot 10} := x^2$$

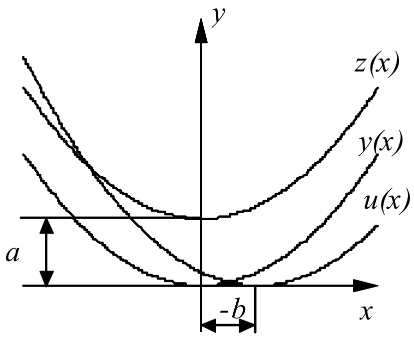
Встроенные функции

acos(z)	арккосинус
angle(x,y)	угол (в радианах) между положительным направлением оси x и радиусом-вектором точки (x,y)
arg(z)	аргумент комплексного числа z (в радианах)
atan(z)	арктангенс
ceil(x)	наименьшее целое, не превышающее x
cols(A)	число столбцов в матрице A
cos(z)	косинус
cspline(vx,vy)	коэффициенты кубического сплайна, построенного по векторам vx и vy
exp(z)	экспонента
fft(v)	быстрое преобразование Фурье вещественных чисел. v – вещественный вектор с 2^n элементами, где n – целое число. Возвращает вектор размера $2^{n-1}+1$
FFT(v)	то же, что и fft(v), но использует другие норму и знак
Find(var1,var2,...)	значения var1,var2,..., доставляющие решение системе уравнений. Число возвращаемых значений равно числу аргументов
floor(x)	наибольшее целое число, меньшее или равное x. x должно быть действительным
Given	ключевое слово, работающее в паре с функциями Find и Minerr
identity(n)	единичная квадратная матрица размером n
if(cond,x,y)	x, если cond больше 0, иначе y
ifft(v)	обратное преобразование Фурье, соответствующее fft. Берется вектор размером $2^{n-1}+1$, где n – целое число. Возвращение действительного вектора размером 2^n
IFFT(v)	обратное преобразование, соответствующее FFT
last(v)	индекс последнего элемента вектора v
linterp(vx,vy,x)	значение в точке x линейного интерполяционного многочлена векторов vx и vy
ln(z)	натуральный логарифм
log(z)	десятичный логарифм
lsolve(M,v)	решение системы линейных алгебраических уравнений вида M*x=v
lspline(vx,vy)	коэффициенты линейного сплайна, построенного по векторам vx и vy
matrix(n,m,f)	матрица, в которой (i,j)-й элемент содержит f(i,j), где i=0,1,..m и j=0,1,..n
max(A)	наибольший элемент в матрице A
min(A)	наименьший элемент в матрице A
mod(x,modulus)	остаток от деления x по модулю. Аргументы должны быть действительными. Результат имеет такой же знак, как и x
polyroots(v)	корни многочлена степени n, чьи коэффициенты находятся в векторе v, длина которого равна n^{+1}
pspline(vx,vy)	коэффициенты параболического сплайна, построенного по векторам vx и vy
root(expr,var)	значение переменной var, при которой выражение exspr равно нулю (в пределах точности TOL)
rows(A)	число строк в матрице A
tan(z)	тангенс

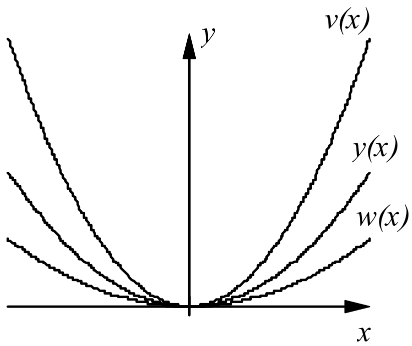
Преобразование графиков

В случае, если известны графики каких-либо функций, возникает необходимость построить графики других функций, выражающиеся через первые. Приведем несколько примеров таких преобразований графиков.

Определен график функции y(x)=f(x), необходимо построить графики функций z(x)=f(x)+a и u(x)=f(x+a) (a и b постоянные), причем величины z(x) и u(x) откладываются по той же оси, что и y(x). Тогда при любом x будет z(x)=y(x)+a, т.е. график функции z(x) получается из графика функции y(x) при помощи переноса вдоль оси y на a в положительном направлении. Что же касается функции u(x),то он получается из графика y(x) переносом на b вдоль оси x в отрицательном направлении.



Подобным образом строятся графики функций v(x)=k*f(x) и w(x)=f(k*x). График функции v(x) получается из графика y(x) путем равномерного растяжения от оси x в k раз, так как точки первого графика имеют при тех же абсциссах ординаты в k раз больше, чем у второго. График же функции w(x) получается из графика функции y(x) равномерным сжатием к оси y в k раз. Конечно, если 0<k<1, то вместо растяжения будет сжатие, и наоборот. Если k<0, то к указанным преобразованиям добавится еще зеркальное отражение графика от оси x для функции v(x) или от оси y для функции w(x).



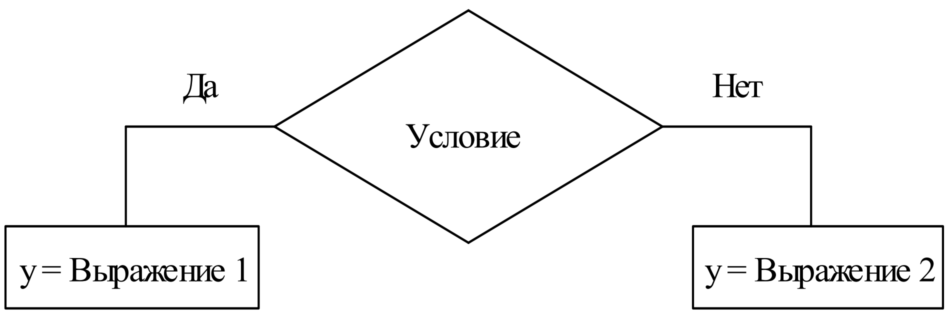
Функция if()

Результат функции if() зависит от выполнения или невыполнения какого-либо условия.

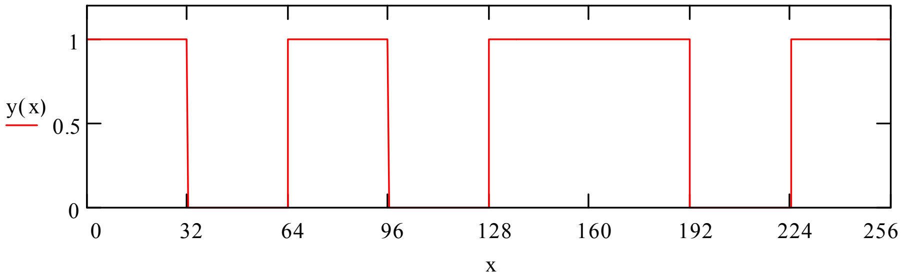
$$y := if\,(условие, выражение1, выражение2)$$

Если условие выполняется, то y принимает значение выражения1, иначе . выражения2.

В качестве выражение1 и выражение2 могут выступать константы, выражения или функции.



При помощи условной функции можно производить построение графиков, например, прямоугольных или треугольных импульсов.



Для этого необходимо задать переменную x, которая находится в диапазоне от 0 до 256 и функцию y(x)

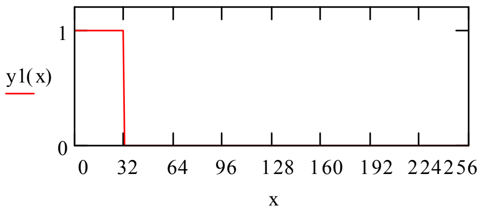
$$y(x) := if(x < 32,1, if(x < 64,0, if(x < 96,1, if(x < 128,0, if(x < 192,1, if(x < 224,0,1))))))$$

Эта запись довольно громоздка и плохо читаема. Чтобы облегчить процесс задания функции, прибегнем к приему, который называется “разбиением на простейшие”.

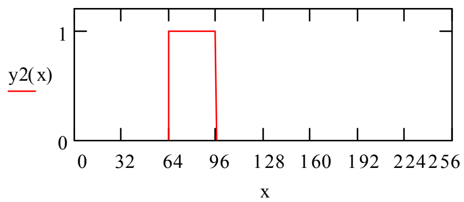
Для этого представим исходную функцию в виде суммы четырех простейших функций

$$y(x) := y1(x) + y2(x) + y3(x) + y4(x)$$

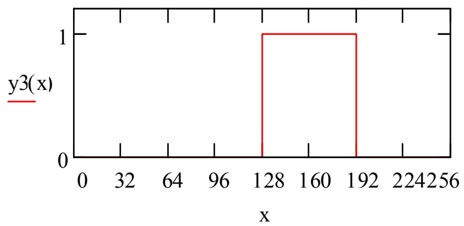
$$y1(x) := if(x < 32,\ 1,\ 0)$$



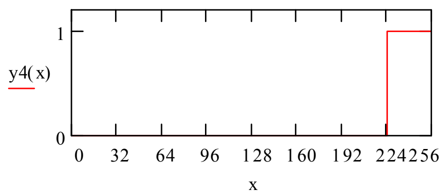
$$y2(x) := if(64 < x < 96,\ 1,\ 0)$$



$$y3(x) := if(128 < x < 192,\ 1,\ 0)$$



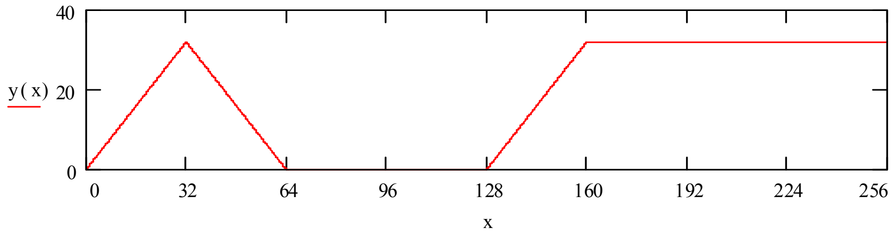
$$y4(x) := if(224 < x < 256,\ 1,\ 0)$$



Участки функции y(x) могут представлять собой не только функции вида f(x)=const, но и могут быть представлены любыми другими функциями.

Рассмотрим пример, когда различные участки функции y(x) задаются уравнениями вида f(x)=k*x+b. Рассмотренное выше построение прямоугольной функции является частным случаем этого типа функций при k=0.

Пусть требуется построить зависимость вида

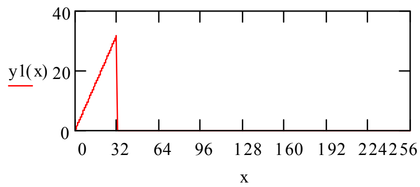


Путь решения задачи прежний: представляем сложную функцию в виде суммы простейших. Таких функций будет четыре.

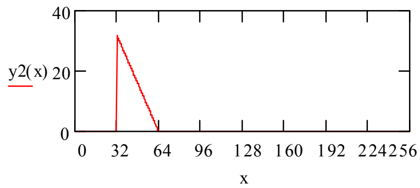
$$y(x) := y1(x) + y2(x) + y3(x) + y4(x)$$

Представление сложной функции с участками, заданными уравнением f(x)=k*x+b в виде суммы простейших функций

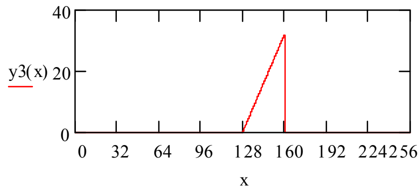
$$y1(x) := if(x < 32, x, 0)$$



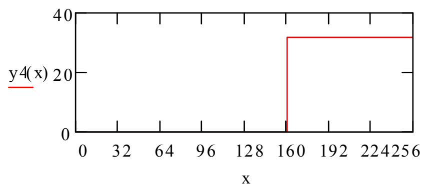
$$y2(x) := if(32 < x < 64, -x + 64, 0)$$



$$y3(x) := if(128 < x < 160, x - 128, 0)$$

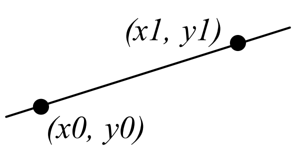


$$y3(x) := if(160 < x < 256, 32, 0)$$



Теперь необходимо найти уравнение прямой для каждого участка в отдельности. Наиболее легко оно определяется для четвертого участка, где f(x)=32.

Уравнение прямой имеет вид f(x)=k*x+b. Прямая считается заданной, если заданы две точки, лежащие на



этой прямой, так как через две точки можно провести прямую единственным способом.

Пусть задана произвольная пара точек (x0, y0) и (x1, y1) лежащих на искомой прямой. Для того чтобы определить уравнение прямой по двум точкам, необходимо координаты этих точек подставить в уравнение прямой в общем виде f(x)=k*x+b и решить систему линейных уравнений вида

$$\begin{cases} y0 = k \cdot x0 + b \\ y1 = k \cdot x1 + b \end{cases}$$

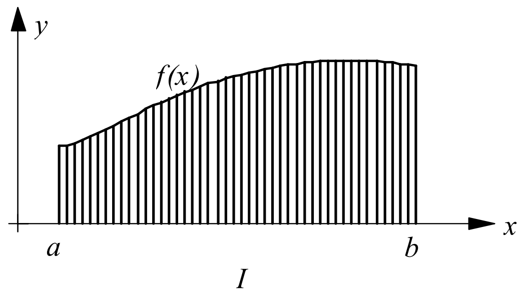
Эта система разрешима однозначно, так как у нас два уравнения с двумя неизвестными. Данную систему можно решить множеством способов. Вычтем из первого уравнения второе. При этом мы избавляемся от одного неизвестного, а именно b.Получим $y0 - y1 = k \cdot (x0 - x1)$

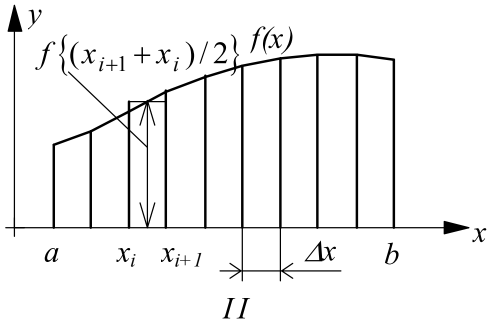
Отсюда . $k = \dfrac{y0 - y1}{x0 - x1}$

Определив коэффициент k и подставив его в любое из уравнений системы, находим коэффициент b

Метод прямоугольников

Дана некоторая функция f(x) на участке [a, b]. Необходимо определить площадь фигуры, ограниченной этой функцией и осью абсцисс (заштрихованная область).





Участок [a, b] разбивается на N интервалов. Каждый из интервалов заменяется прямоугольником, высота которого равна значению функции, вычисленному на середине этого интервала, а ширина - длине участка, которая определяется как (a-b)/N. Далее вычисляется сумма площадей этих прямоугольников, которая с некоторой точностью и равна искомой площади фигуры.

Вычислим площадь окружности.

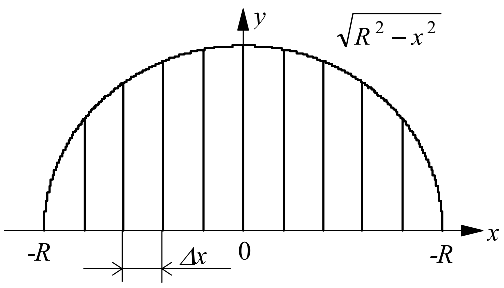
Площадь окружности радиуса R определяется выражением . $S_a = \pi \cdot R^2$

Оценим, как соотносятся площади окружности, вычисленной по аналитическому выражению и вычисленной методом “прямоугольников”.

Верхняя половина окружности определяется выражением $y(x) = \sqrt{R^2 - x^2}$

Определив площадь верхней половины окружности, легко можно найти ее полную площадь, умножив полученный результат на 2.

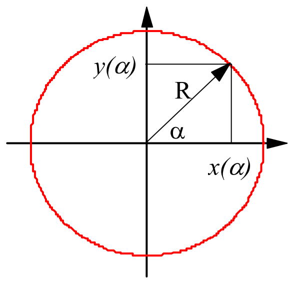
Разобьем диапазон изменения переменной x на N равных частей



Запись $y\left(-R+k\cdot\Delta x+\frac{\Delta x}{2}\right)$ означает, что необходимо вычислить значение функции $y(x)=\sqrt{R^2-x^2}$, подставив вместо x выражение $-R+k\cdot\Delta x+\frac{\Delta x}{2}$.

ПОСТРОЕНИЕ ФИГУР, ВПИСАННЫХ В ОКРУЖНОСТЬ

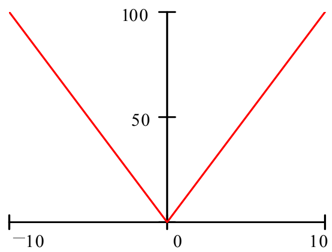
Построение фигур, вписанных в окружность



Любая точка, расположенная на окружности радиуса R, должна удовлетворять системе состоящей из двух уравнений вида:

$$\begin{cases} x(\alpha)=R\cdot\cos(\alpha) \\ y(\alpha)=R\cdot\sin(\alpha) \end{cases}$$

Как было сказано выше, при построении графиков в пакете Mathcad рассчитываются пары чисел, соответствующие значению переменной, и функции при этом значении переменной, а затем данные точки соединяются отрезками прямых линий. Так, например, если мы попросить вывести график параболы y(x)=x2, задав x следующим образом x := -10,0..10, то график будет иметь вид, лишь отдаленно напоминающий параболу.



Для увеличения точности расчетов необходимо уменьшить шаг, что позволит получить более плавную кривую. Необходимое количество точек расчета выбирается из критерия получения необходимой точности расчетов. Mathcad требует вводить значения аргумента тригонометрических функций в радианах.

Задание окружности с учетом вышесказанного имеет вид:

$\alpha := 0, 0.01 \,..\, 2\cdot\pi \qquad R := 10$

$x(\alpha) := R\cdot cos(\alpha)$

$y(\alpha) := R\cdot sin(\alpha)$

Так как окружность является вырожденым случаем многоугольника, вписанного в окружность, при числе вершин стремящемся к бесконечности, то алгоритм построения фигур, вписанных в окружность, аналогичен вышеприведенному алгоритму построения окружности, с той лишь разницей, что шаг изменения переменной по углу должен быть увеличен.

Например, построение квадрата выглядит следующим образом:

$\alpha := 0, \frac{\pi}{2} .. 2 \cdot \pi \quad R := 10$

$x(\alpha) := R \cdot cos(\alpha)$

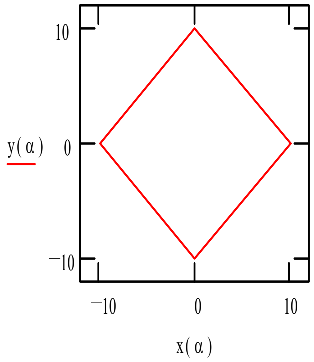
$y(\alpha) := R \cdot sin(\alpha)$

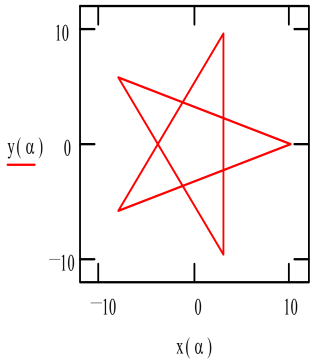
а для построения пятиконечной звезды необходимо сделать следующую запись:

$\alpha := 0, \frac{4 \cdot \pi}{5} .. 10 \cdot \pi \quad R := 10$

$x(\alpha) := R \cdot cos(\alpha)$

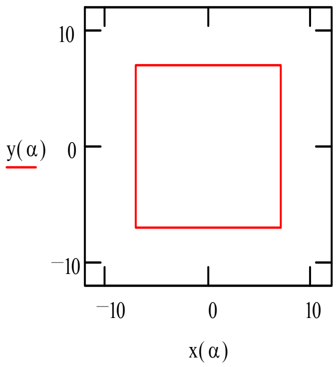
$y(\alpha) := R \cdot sin(\alpha)$

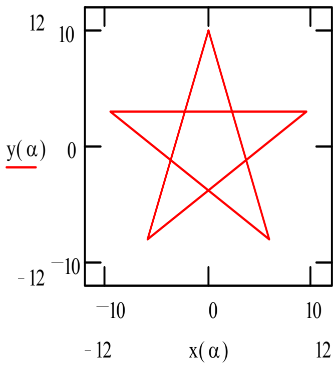




Для того, чтобы отобразть фигуры в привычном виде необходимо,чтобы первая точка рассчитывалась при нужном значении начального угла. Например, для квадрата поворот выглядит следующим образом:

$\alpha := \frac{\pi}{4}, \frac{\pi}{2} + \frac{\pi}{4} .. 2 \cdot \pi + \frac{\pi}{4}$





Однако это не всегда удобно и ненаглядно. Намного удобнее ввести начальный угол в аргументы cos и sin. Это можно сделать, например, для квадрата следующим образом:

$x(\alpha) := R \cdot cos(\alpha + \frac{\pi}{4})$

$y(\alpha) := R \cdot sin(\alpha + \frac{\pi}{4})$

Намного привычнее задавать угол не в радианах, а в градусах. При этом необходимо величину угла в скобках у тригонометрических функций преобразовать к радианам.

$$\alpha^{/} = \frac{\alpha^{o}}{180} \cdot \pi$$

$$\alpha^{o} = \frac{\alpha^{/}}{\pi} \cdot 180$$

Построение квадрата при этом выглядит следующим образом

$\alpha := 0,90\,..\,360 \qquad R := 10$

$$x(\alpha) := R \cdot cos\left(\frac{\alpha}{180} \cdot \pi\right)$$

$$y(\alpha) := R \cdot sin\left(\frac{\alpha}{180} \cdot \pi\right)$$

.

Существует еще один способ построения вписанных в окружность фигур. Он состоит в том, чтобы ввести шаг изменения угла под знак cos и sin. Применительно к построению квадрата это выглядит следующим образом:

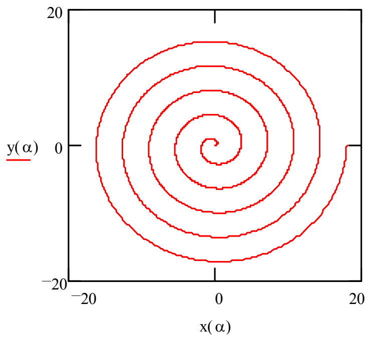
$\alpha := 0\,..\,4 \qquad R := 10 \qquad \Delta = \frac{\pi}{2}$

$x(\alpha) := R \cdot cos(\Delta \cdot \alpha)$

$y(\alpha) := R \cdot sin(\Delta \cdot \alpha)$

Спираль Архимеда

Одновременно с увеличением угла увеличивается и радиус. Таким образом, спираль раскручивается. Программа построения данной фигуры в Matcad выглядит следующим образом:



$k := 5 \qquad \alpha := 0\,..\,360 \cdot k \qquad R := 10$

$R(\alpha) := 0.01 \cdot R$

$$x(\alpha) := R(\alpha) \cdot cos\left(\frac{\alpha}{180} \cdot \pi\right)$$

$$y(\alpha) := R(\alpha) \cdot sin\left(\frac{\alpha}{180} \cdot \pi\right)$$

.

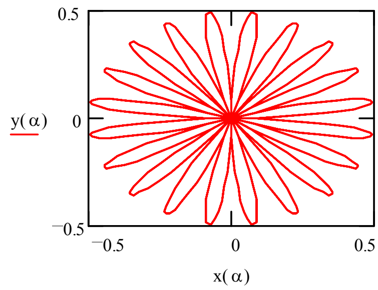
переменная к определяет число витков спирали.

Построенная таким образом спираль Архимеда имеет равномерный шаг. Для создания неравномерного шага закон изменения радиуса R(a) должен быть нелинейным, например, квадратичным или кубическим.

Выбирая различные законы изменения R(a) можно получить множество занятных фигур. Например, при

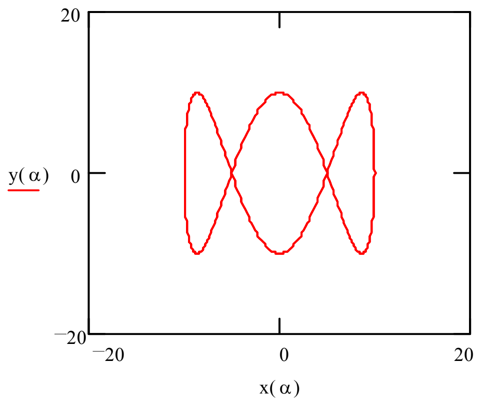
$$R(\alpha) = cos\left(\frac{5 \cdot \alpha}{180} \cdot \pi\right) \cdot sin\left(\frac{5 \cdot \alpha}{180} \cdot \pi\right)$$

получается фигура



Фигуры Лиссажу.

Внешний вид одной из огромного множества фигур Лиссажу



Для ее построения необходимо указать следующее:

$\alpha := 0 .. 360 \cdot 2 \qquad R := 10$

$$x(\alpha) := R \cdot \cos\left(\frac{\alpha}{180} \cdot \pi\right)$$

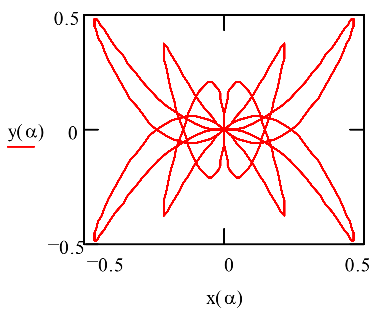
$$y(\alpha) := R \cdot \sin\left(\frac{3 \cdot \alpha}{180} \cdot \pi\right)$$

Характерно, что аргумент тригонометрической функции по одной координате меняется в три раза быстрее, чем по другой, в данном случае по координате y. Отношение аргументов определяет число замкнутых областей на графике. Если это отношение равно единице, как в случае окружности, то получается одна замкнутая область. В данном примере отношение равно трем, соответственно и число замкнутых областей тоже равно трем.

Комбинируя принцип построения фигур Лиссажу с различными законами изменения радиуса, можно получить еще более замысловатые фигуры. Например, при законе изменения радиуса

$$R(\alpha) = \cos\left(\frac{2 \cdot \alpha}{180} \cdot \pi\right) \cdot \sin\left(\frac{2 \cdot \alpha}{180} \cdot \pi\right)$$

и соотношении аргументов тригонометрических функций, равном пяти, получается фигура



Построение объемных фигур.

Трехмерная графика требует задания функции двух переменных f(x,y).

Один из возможныж способоб построения объемных фигур состоит в следующем. Необходимо сформировать прямоугольную таблицу, номера ячеек которой по горизонтали будут значениями переменной x, номера ячеек по вертикали будут значениями переменной y, а значения ячеек таблицы - значениями функции.

Вычисленные по ряду значений x и y значения функции f(x,y) используются для задания двумерной таблицы с именем M. Значения переменных x и y привязаны к номерам соответственно столбцов и строк таблицы.

Построение трехмерной поверхности рассмотрим на примере построения полусферы.

Уравнение полусферы в декартовой системе координат имеет вид

$$f(x,y)=\sqrt{R^2-x^2-y^2}$$

, где R - радиус сферы.

Переменные x и y при этом должны изменяться в пределах от -R до R (аналогично построению окружности на плоскости).

$$x_i := R\cdot\left(-1+2\cdot\frac{i}{I}\right),\qquad i := 0\,..\,I$$

$$y_j := R\cdot\left(-1+2\cdot\frac{j}{J}\right),\qquad j := 0\,..\,J$$

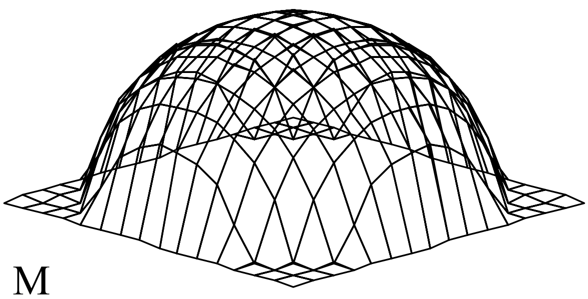
где I и J максимальное количество точек построения по осям x и y соответственно. Из выражений видно, что каждому значению индекса ставится в соответствие одно значение переменной.

Из выражений также видно, что при изменении индекса i и j, соответственно от 0 до I и от 0 до J, значения элементов таблицы изменяются в пределах от -R до R.

Остается лишь записать значения функции f(x,y), вычисленные при значениях аргументов , в таблицу M.

$$M_{i,j} := f\left(x_i, y_j\right)$$

Для отображения графика необходимо вызвать шаблон, предназначенный для построения поверхности. В настройках графика можно установить режим цветных поверхностей



M

МАТРИЦЫ

Матрицей называется прямоугольная таблица указанного вида, которая, в отличие от обычной прямоугольной таблицы, подчиняется определенным правилам сложения, вычитания, умножения и равенства. Элементы матрицы записываются при помощи двойного индекса. Обратите внимание, что i, j целые числа. Первый индекс указывает строку таблицы, на которой расположен элемент, а второй - ее столбец. Столбцы матрицы называют векторами-столбцами, а строки матрицы - векторами-строками.

Основные типы матриц.

В процессе работы с пакетом Mathcad вам понадобятся три основных типа матриц.

Прямоугольная, состоящая из m - строк и n - столбцов

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Матрица-столбец - матрица, состоящая из одного столбца и m - строк

$$\begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_m \end{bmatrix}$$

Матрица-строка - матрица, состоящая из одной строки и n - столбцов

Простейшие операции над матрицами.

Сложение и вычитание

Суммой матриц A и B называется матрица C, элемент которой определяется как

$$c_{ij} = a_{ij} + b_{ij}$$

Обратите внимание, что A+B=B+A

Замечание: Складывать можно матрицы, имеющие одинаковый размер.

Аналогично разностью матриц A и B называется матрица C, элемент которой определяется как

$$c_{ij} = a_{ij} - b_{ij}$$

Операции сложения и вычитания матриц в общем виде

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \pm \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11} \pm b_{11} & a_{12} \pm b_{12} \\ a_{21} \pm b_{21} & a_{22} \pm b_{22} \end{bmatrix}$$

Произведение матриц

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11} \cdot b_{11} + a_{12} \cdot b_{21} & a_{11} \cdot b_{12} + a_{12} \cdot b_{22} \\ a_{21} \cdot b_{11} + a_{22} \cdot b_{21} & a_{21} \cdot b_{12} + a_{22} \cdot b_{22} \end{bmatrix}$$

Как видим

- элемент результирующей матрицы с индексами 1,1 получатся перемножением первой строки первой матрицы на первый столбец второй матрицы,
- элемент результирующей матрицы с индексами 1,2 получатся перемножением первой строки первой матрицы на второй столбец второй матрицы,
- элемент результирующей матрицы с индексами 2,1 получатся перемножением второй строки первой матрицы на первый столбец второй матрицы,
- элемент результирующей матрицы с индексами 2,2 получатся перемножением второй строки первой матрицы на второй столбец второй матрицы.

Замечание: перемножать можно только матрицы, у которых число столбцов первой матрицы равно числу строк второй матрицы.

$$c_{ik} = \sum_{k=1}^{n} a_{ik} \cdot b_{ki}$$

Обращайте внимание на порядок записи матриц при перемножении, так как в общем случае

$$A \cdot B \neq B \cdot A$$

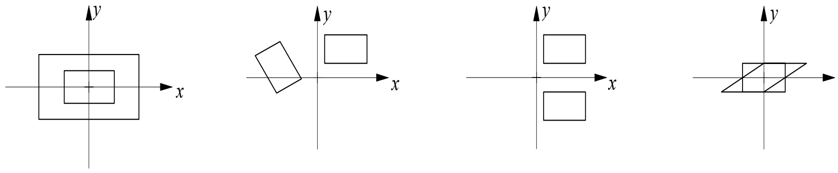
Умножение матрицы на число

$$k \cdot \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} k \cdot a_{11} & k \cdot a_{12} \\ k \cdot a_{21} & k \cdot a_{22} \end{bmatrix}$$

При этом каждый элемент матрицы умножается на это число.

Применение матричного аппарата для решения различных задач.

Одним из огромного многообразия вариантов применения матричного аппарата является линейное преобразование фигур на плоскости и в пространстве. Примерами линейных преобразований на плоскости могут являться масштабирование (увеличение размеров фигуры в k как по оси x так и по оси y), отображение (изменение знака на противоположный по одной или по обоим осям одновременно), сдвиг по осям координат и поворот (смещение всех точек фигуры на некоторый угол относительно заданной точки). Примеры этих преобразований приведены на Рис.3.1.



масштабирование поворот отображение сдвиг

Данные преобразования довольно часто используется при разработке игровых программ и программ обработки изображений, реализуемых как на плоскости так и в объемном варианте. Примером подобных программ могут служить также хранители экрана в ОС WINDOWS.

Если обозначить p1 – преобразованную матрицу, p – исходную, CONV – матрицу преобразования, преобразование можно провести по формуле

$$p1 = CONV \cdot p$$

№	вид преобразования	матрица преобразования
1	масштабирование в k раз по оси абсцисс и k раз по оси ординат	$\begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix}$
2	масштабирование в k раз по оси абсцисс и l раз по оси ординат	$\begin{bmatrix} k & 0 \\ 0 & l \end{bmatrix}$
3	отображение относительно оси абсцисс без масштабирования	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$
4	отображение относительно оси абсцисс с масштабированием в k раз по обоим осям	$\begin{bmatrix} -k & 0 \\ 0 & k \end{bmatrix}$
5	отображение относительно оси ординат без масштабирования	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
6	отображение относительно оси ординат с масштабированием по обоим осям	$\begin{bmatrix} k & 0 \\ 0 & -k \end{bmatrix}$
7	сдвиг по оси абсцисс	$\begin{bmatrix} 1 & \lambda \\ 0 & 1 \end{bmatrix}$
8	сдвиг по оси ординат	$\begin{bmatrix} 1 & 0 \\ \lambda & 1 \end{bmatrix}$
9	поворот на угол α относительно центра координат	$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$