Глава 18. Специализированные функции

В этой главе мы кратко опишем наиболее важные встроенные функции Mathcad, которые не были рассмотрены нами ранее в соответствующих тематических разделах. Это и такие элементарные функции, как синус или экспонента, которыми мы широко пользовались, не задумываясь об их систематизации, и весьма специфичные функции вроде полиномов Чебышева или финансовых функций.

Как вы помните, для того чтобы ввести какую-то функцию, нужно либо просто набрать ее имя с клавиатуры, либо обратиться к специальному окну Insert Function (Вставить функцию), которое можно вызвать, задействовав сочетание Ctrl+Shift+F или нажав специальную кнопку панели Standard (Стандартные). Для удобства все встроенные функции разделены на 28 тематических категорий, в соответствии с которыми мы и будем производить их описание в этой главе. Четыре дополнительные категории (Data Analysis, Image Processing, Signal Processing, Wavelets) содержат функции соответствующих пакетов расширений, которые становятся доступными только после встраивания пакета в систему. Кстати, многие функции принадлежат сразу нескольким категориям в связи с неоднозначностью выполняемых ими задач. В таких случаях мы, чтобы не повторяться, будем описывать их только в одном разделе, давая в других ссылку.

Описание нерассмотренных в предыдущих главах функций мы будем вести согласно списку тематических категорий окна Insert Function (Вставить функцию).

Всего в Mathcad сотни встроенных функций. Естественно, что запомнить все эти функции невозможно, да и делать этого, в общем, не нужно. Имея представление о наличии функции, выполняющей ту или иную задачу, вы без проблем сможете найти ее либо в соответствующей категории, просмотрев описания всех относящихся к ней функций, либо пролистав специальное приложение в конце книги. Кстати, большинство из встроенных функций Mathcad элементарно могут быть заданы и пользователем с помощью программ всего из нескольких строк. Это еще один весьма весомый довод того, что не стоит пытаться изучить абсолютно все встроенные функции Mathcad.

18.1. Функции Бесселя (Bessel)

Функциями Бесселя в математике принято называть цилиндрические функции — важнейший класс специальных функций, являющихся численным решением дифференциального уравнения вида

$$x^2 \cdot \frac{d^2}{dx^2} y + x \cdot \frac{d}{dx} y + \left(x^2 - n^2\right) \cdot y = 0$$

К приведенному уравнению сводятся многие краевые задачи математической физики, связанные с вопросами равновесия и установившимися колебаниями тел цилиндрической формы.

В Mathcad имеется несколько встроенных функций, соответствующих решению уравнения Бесселя с различными параметрами. Исходя из принятой в математике систематики, их можно разделить на следующие группы.

18.1.1. Обычные функции Бесселя

Функция Бесселя 1-го рода $J_\nu(x)$ порядка ν (в том числе дробного и отрицательного) задается в виде бесконечного ряда:

$$J(\nu, x) = \left(\frac{1}{2} \cdot x\right)^\nu \cdot \sum_{k=0}^{\infty} \frac{\left(-\frac{1}{4} \cdot x^2\right)^k}{k! \cdot \Gamma(\nu + k + 1)}$$

Данный ряд сходится при всех x, однако при x, удаленных от нуля, — крайне медленно.

В Mathcad для вычисления функций Бесселя 1-го рода имеются следующие встроенные средства.

- J0(x). Функция Бесселя 1-го рода 0-го порядка. Исторически первая из функций Бесселя была определена в 1738 году Яковом Бернулли в работе о колебаниях тяжелой цепи. Соответствует решению волнового уравнения при n=0 в цилиндрических координатах.
- J1(x). Функция Бесселя 1-го рода 1-го порядка. Решение волнового уравнения при n=1.
- Jn(m,x). Функция Бесселя 1-го рода m-го порядка. Решение волнового уравнения для произвольного n (n=m) (рис. 18.1).

Переменная x может принимать любые значения из действительной или комплексной области.

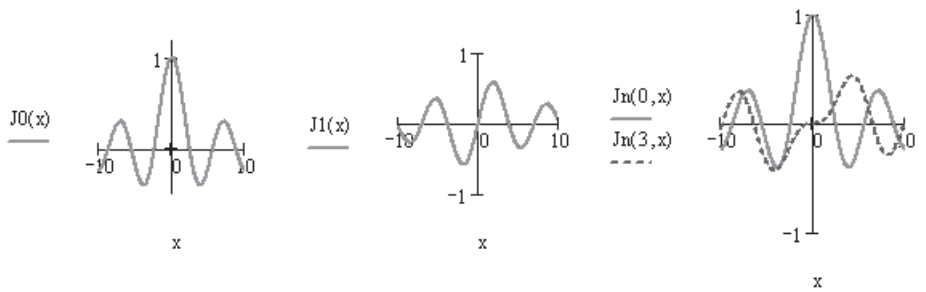

Рис. 18.1. Функции Бесселя 1-го рода

Функция Бесселя 2-го рода $Y_\nu(x)$ порядка ν (называемая также функцией Вебера) определяется через функцию Бесселя 1-го рода:

$$Y(\nu, x) = \lim_{\mu \to \nu} \frac{J(\mu, x)\cdot\cos(\mu\cdot\pi) - J(-\mu, x)}{\sin(\mu\cdot\pi)}$$

Однако вычислить данный предел в произвольной точке x удается лишь для дробного порядка ν.

Для вычисления функций Бесселя 2-го рода задействуйте следующие встроенные функции.

- `Y0(x)`. Функция Бесселя 2-го рода 0-го порядка.
- `Y1(x)`. Функция Бесселя 2-го рода 1-го порядка.
- `Yn(m,x)`. Функция Бесселя 2-го рода произвольного порядка.

Переменная x функций Бесселя 2-го порядка может принимать действительные положительные или комплексные значения.

18.1.2. Модифицированные функции Бесселя

Модифицированные функции Бесселя (или функции мнимого аргумента) 1-го рода $I_n(x)$ и 2-го рода $K_n(x)$ целого порядка n определяются следующими выражениями:

$$I(n, x) = (-i)^n \cdot Jn(n, ix)$$

$$K(n, x) = \frac{\pi}{2}\cdot i^{n+1}\cdot(Jn(n, i\cdot x) + i\cdot Yn(n, i\cdot x))$$

В Mathcad их можно вычислить с помощью перечисленных ниже функций.

- `I0(x)`. Модифицированная функция Бесселя 1-го рода 0-го порядка.
- `I1(x)`. Модифицированная функция Бесселя 1-го рода 1-го порядка.
- `In(m,x)`. Модифицированная функция Бесселя 1-го рода произвольного порядка (в том числе и дробного) (рис. 18.2).

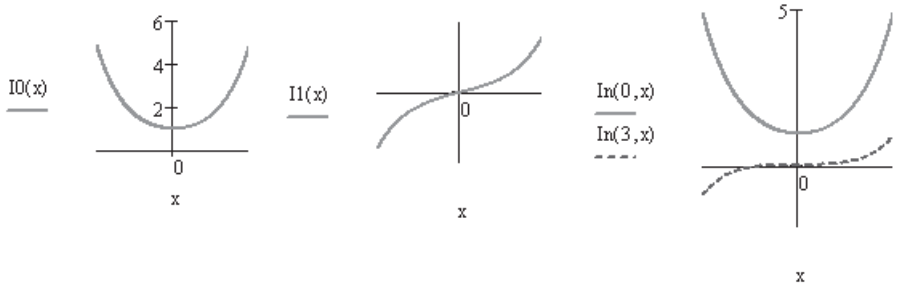

Рис. 18.2. Модифицированные функции Бесселя 1-го рода

- `K0(x)`. Модифицированная функция Бесселя 2-го рода 0-го порядка.
- `K1(x)`. Модифицированная функция Бесселя 2-го рода 1-го порядка.
- `Kn(m, x)`. Модифицированная функция Бесселя 2-го рода произвольного порядка.

Переменная x модифицированных функций Бесселя может принимать любые действительные или комплексные значения.

18.1.3. Функции Эйри

Функции Эйри являются частными решениями следующего дифференциального уравнения:

$$\frac{d^2}{dx^2}y - x \cdot y = 0$$

В Mathcad имеются две встроенные функции этой категории (рис. 18.3).

- Ai(x). Функция Эйри 1-го рода.
- Bi(x). Функция Эйри 2-го рода.

Значение переменной x для функции Bi(x) по величине не должно превышать 103.35.

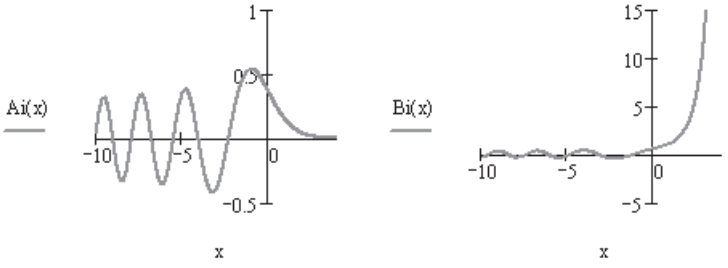

Рис. 18.3. Функции Эйри

18.1.4. Функции Ганкеля (Hankel)

Функции Ганкеля задаются следующими комбинациями функций Бесселя 1-го и 2-го рода произвольного порядка:

$$H1(m,x) = Jn(m,x) + i \cdot Yn(m,x)$$
$$H2(m,x) = Jn(m,x) - i \cdot Yn(m,x)$$

В Mathcad существуют две встроенные функции, соответствующие этим выражениям.

- H1(m,x). Функция Ганкеля 1-го рода произвольного порядка m.
- H2(m,x). Функция Ганкеля 2-го рода произвольного порядка m.

Функции Ганкеля также носят название функций Бесселя 3-го рода.

Для всех функций, перечисленных в разд. 18.1.1–18.1.4, существуют нормированные аналоги, определяемые выражением $f_{sc}(X)=\exp(-x)\cdot f(X)$, где x (в зависимости от функции) — действительный аргумент X либо действительная или мнимая часть комплексного аргумента X. В отличие от стандартных функций Бесселя, f(x) значения нормированных функций $f_{sc}(x)$ можно получить и при больших величинах аргумента. Задаются нормированные аналоги добавлением к имени функции окончания .sc.

18.1.5. Функции Бесселя–Кельвина

Комплексным обобщением модифицированных функций Бесселя являются функции Бесселя–Кельвина, комбинация которых ber(n,x)+i·bei(n,x) описывает решение следующего дифференциального уравнения:

$$x^2 \cdot \frac{d^2}{dx^2} y + x \cdot \left(\frac{d}{dx} y \right) - \left(i \cdot x^2 + n^2 \right) \cdot y = 0$$

Соответственно в Mathcad имеются две функции этой категории.

- bei(n,x) — мнимая часть функции Бесселя–Кельвина произвольного порядка n (n — целое неотрицательное число).
- ber(n,x) — действительная часть функции Бесселя–Кельвина произвольного порядка.

18.1.6. Сферические функции Бесселя

Сферические функции Бесселя описывают решение следующего дифференциального уравнения:

$$x^2 \cdot \frac{d^2}{dx^2} y + 2 \cdot \frac{d}{dx} y + \left[x^2 + n \cdot (n+1) \right] \cdot y = 0$$

Сферические функции 1-го и 2-го рода порядка n, где n — целое число, описываются следующими выражениями:

$$js(n,x) = \sqrt{\frac{\pi}{2x}} \cdot Jn\left(n + \frac{1}{2}, x \right) \qquad ys(n,x) = \sqrt{\frac{\pi}{2x}} \cdot Yn\left(n + \frac{1}{2}, x \right)$$

В Mathcad имеются две встроенные функции, относящиеся к этой категории:

- js(n,x) — сферическая функция Бесселя 1-го рода порядка n;
- ys(n,x) — сферическая функция Бесселя 2-го рода порядка n.

Переменная x для этих функций может принимать только положительные значения. Кроме того, x может быть равна 0 в случае использования ys при n≤0 (рис. 18.4).

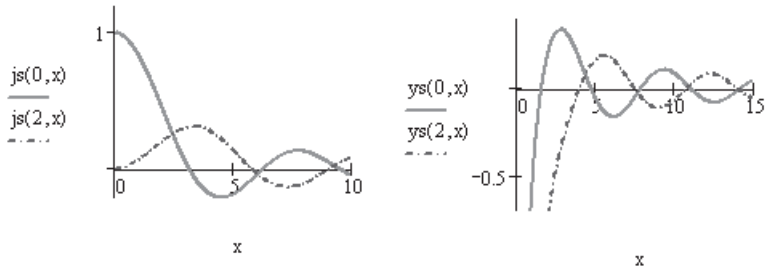

Рис. 18.4. Сферические функции Бесселя

18.2. Функции определения типа выражения (Expression Type)

Очень часто в программировании приходится учитывать тип выражения, поступившего на обработку. Так, например, при создании программы, вычисляющей сумму элементов вектора, следует учесть, что пользователь может попробовать просуммировать

и элементы матрицы или использовать эту программу по отношению к строке. Поэтому каким-то образом требуется поставить на входе алгоритма логический фильтр, запускающий программу, если тип поступивших данных приемлем, и возвращающий соответствующее сообщение об ошибке, если он некорректен.

В Mathcad имеются специальные функции, относящиеся к группе Expression Type (Тип выражения), позволяющие решать подобные проблемы. Причем особенностью является то, что вы не сможете их напрямую задать, в отличие от большинства других встроенных функций, используя более простые средства.

Описываемые функции являются по сути логическими, поэтому их результатом будет либо 0, если некоторое условие не выполняется, либо 1, если оно справедливо.

- IsArray(x). Эта функция определяет, является ли некоторое выражение x матрицей, вектором или тензором. Возвращает 0, если x — скаляр, и 1, если x — матрица, вектор или вложенный массив.
- IsNaN(x). Функция возвращает 1, если импортированные из внешнего файла данные (например, матрица x) содержат элементы, не интерпретируемые Mathcad как числа или строки, и 0 — в противном случае.
- IsScalar(x). С помощью этой функции можно определить, является ли x скаляром (действительным или комплексным) или же его значение выражено матрицей или строкой.

Пример 18.1. Использование функции IsScalar

$$M := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \qquad X := \pi \qquad Xi := i + \pi \qquad S := \text{"String"}$$

$$\text{IsScalar}(M) = 0 \qquad \text{IsScalar}(X) = 1 \qquad \text{IsScalar}(Xi) = 1 \qquad \text{IsScalar}(S) = 0$$

- IsString(x). Используя эту функцию, можно определить, является ли x строкой. Возвращает 1, если x — строковая переменная, и 0 — во всех остальных случаях.

18.3. Гиперболические функции (Hyperbolic)

При решении дифференциальных уравнений ответ обычно выражается через разнообразные сочетания экспонент e^z и e^{-z}. Было замечено, что между наиболее важными и часто встречающимися комбинациями можно устанавливать связь, подобную существующей между тригонометрическим функциям. Исходя из этого была разработана теория гиперболических функций, которая позволила значительно упростить решение целого ряда как чисто математических, так и прикладных задач.

В Mathcad гиперболические функции широко используются в результатах некоторых символьных преобразований (в основном, связанных с решением дифференциальных уравнений). Всего в систему встроено 12 таких функций: шесть прямых и шесть обратных.

- sinh(z) — гиперболический синус; cosh(x) — гиперболический косинус;

$$\sinh(x) = \frac{e^x - e^{-x}}{2} \qquad \cosh(x) = \frac{e^x + e^{-x}}{2}$$

- tanh(x) — гиперболический тангенс; coth(x) — гиперболический котангенс

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \qquad \coth(x) = \frac{1}{\tanh(x)}$$

- sech(x) — гиперболический секанс; csch(x) — гиперболический косеканс

$$\operatorname{sech}(x) = \frac{1}{\cosh(x)} \qquad \operatorname{csch}(x) = \frac{1}{\sinh(x)}$$

- acosh(x) — гиперболический арккосинус; asinh(x) — гиперболический арксинус;
- atanh(x) — гиперболический арктангенс; asech(x) — обратный гиперболический секанс;
- acsch(x) — обратный гиперболический косеканс; acoth(z) — гиперболический арккотангенс.

Mathcad способен вычислять значение гиперболических функций не только от действительного, от и от комплексного x.

18.4. Логарифмы и экспонента (Log and Exponential)

С тремя из пяти функций этой группы мы уже хорошо знакомы, так как не раз использовали их в примерах и, кроме того, встречались с ними, когда обсуждали рабочую панель Calculator (Калькулятор). Так как эти функции тривиальны, то подробно их обсуждать мы не будем и ограничимся простым перечислением:

- exp(x) — величина основания натурального логарифма в степени x. На практике же обычно используют непосредственное определение типа e^x и прибегают к описываемой функции лишь в том случае, если x представлен громоздким выражением;
- ln(x) — натуральный логарифм;
- lnΓ(x) (lnGamma(x)) — натуральный логарифм значения гамма-функции, вычисленного для заданного x;
- log(x) — десятичный логарифм; log(x,b) — логарифм x по основанию b.

В Mathcad понятие логарифма рассматривается в обобщенном плане, поэтому аргумент для соответствующих функций может быть как действительным положительным, так и отрицательным или комплексным числом. Единственное, для перечисленных выше функций x не может принимать значения 0. Однако в данной категории имеется функция ln0(x), полностью аналогичная ln(x), но возвращающая для x=0 число, равное машинной бесконечности.

18.5. Тригонометрические функции (Trigonometric)

В Mathcad имеются следующие тригонометрические функции:

- acos(x) — арккосинус; acot(x) — арккотангенс; acsc(x) — арккосеканс;
- angle(x,y) — функция служит для вычисления угла между линией, соединяющей точку (x,y) с началом координат, и положительной частью оси абсцисс;

- asec(x) — арксеканс; asin(x) — арксинус; atan(x) — арктангенс;
- atan2(x,y) — функция предназначена для определения угла между линией, соединяющей точку (x,y) с началом координат, и осью X. Отличие от функции angle состоит в том, что угол вычисляется не относительно положительной части оси абсцисс, а той, на которую проецируется точка;
- cos(x) — косинус; cot(x) — котангенс; csc(x) — косеканс;
- sec(x) — секанс; sin(x) — синус; tan(x) — тангенс;
- sinc(x) — возвращает значение предела sin(x)/x в точке x.

Все тригонометрические функции Mathcad могут работать как с комплексными, так и с действительными аргументами. Исключение составляют функции угла atan2 и angle, значение x для которых должно быть действительным, не равным 0.

В Mathcad тригонометрические функции одинаково успешно могут быть использованы как в численных, так и в символьных расчетах. В общем, все те преобразования с ними, которые вы можете провести на бумаге, абсолютно в той же форме можно осуществить и в Mathcad.

По умолчанию аргуме нты тригонометрических функций и результаты вычисления с помощью обратных тригонометрических функций в Mathcad рассматриваются в радианной мере. Для того же, чтобы перевести их в градусы, следует либо произвести непосредственный пересчет, либо использовать размерность deg (от англ. degrees — градусы).

Пример 18.2. Перевод радианной меры в градусную

$$\alpha 1 := \pi \qquad \alpha 2 := 180 \qquad \alpha 3 := 180 \cdot \mathrm{deg}$$

$$\sin(\alpha 1) = 0 \qquad \sin(\alpha 2) = -0.801 \qquad \sin(\alpha 3) = 0 \qquad \sin\left(\frac{\alpha 2}{180} \cdot \pi\right) = 0$$

$$\mathrm{atan}\left(\frac{1}{\sqrt{2}}\right) = 0.615 \qquad \mathrm{atan}\left(\frac{1}{\sqrt{2}}\right) = 0.196\pi \qquad \mathrm{atan}\left(\frac{1}{\sqrt{2}}\right) = 35.264\,\mathrm{deg}$$

Вычисление прямых тригонометрических функции на компьютере основано на использовании цепной дроби, которая получается при разложении функции tg(x):

$$\mathrm{tg}(x) = \cfrac{x}{1 + \cfrac{-x^2}{3 + \cfrac{-x^2}{5 + \ddots + \cfrac{-x^2}{2k+1}}}}$$

Остальные функции (sin(x), cos(x) и др.) выражаются, согласно известным тригонометрическим формулам, через тангенс половинного аргумента.

Применение цепной дроби для расчета тригонометрических функций оправдано ввиду ее быстрой сходимости по сравнению с рядом Тейлора. При небольшом количестве итераций (k=15) достигается точность вплоть до 18-го знака мантиссы при условии, что значение аргумента по модулю не превышает π/4. Чтобы убедиться в сказанном, создадим рекурсивный алгоритм, вычисляющий тангенс с помощью цепной дроби.

$$\mathrm{tg}(x) := \left| \begin{array}{l} x \leftarrow x - \mathrm{Trunc}(x, \pi) \\ k \leftarrow 15 \\ \mathrm{brain}(\mathrm{prev\_res}, k) \leftarrow \left| \begin{array}{ll} \mathrm{brain}\left(2 \cdot k + 1 + \dfrac{-x^2}{\mathrm{prev\_res}}, k - 1\right) & \text{if } k \geq 0 \\ \mathrm{return\ prev\_res} & \text{otherwise} \end{array} \right. \\ \mathrm{return}\ \dfrac{x}{\mathrm{brain}(2k + 1, k)} \end{array} \right.$$

Первая строка программы задает интервал изменения аргумента от $-\pi$ до π (условие, необходимое для быстрой сходимости алгоритма). Для любого x, меньшего π по абсолютной величине, возвращается исходное значение аргумента. В противном случае от него отнимается $n \cdot \pi$, где n — целая часть числа, полученного при делении исходного аргумента на p. Во второй строке задаем необходимое количество итераций. Далее снизу вверх подсчитывается цепная дробь. Первоначально вычисляется нижняя часть дроби $-x^2/(2k+1)$. Это есть первое значение параметра prev_res. На каждой итерации k рекурсивно вызываемая функция brain поднимается по цепной дроби на ступеньку вверх, прибавляет к prev_res нечетное $2(k-1)+1$ и вновь присваивает полученное значение параметру prev_res. Вычисления продолжаются до тех пор, пока не будет достигнута верхняя граница дроби (т.е. пока не исчерпается лимит итераций: if $k \geq 0$). В качестве ответа будет возвращена величина x, деленная на последнее значение параметра prev_res.

Чтобы проверить корректность работы алгоритма, сравним результаты его работы со значениями, полученными с помощью встроенной функции tan(x).

$$\mathrm{tg}\left(\frac{\pi}{8}\right) = 0.41421356237309501 \qquad \mathrm{tg}\left(\frac{8}{3}\pi\right) = -1.7320508075688796$$

$$\tan\left(\frac{\pi}{8}\right) = 0.41421356237309501 \qquad \tan\left(\frac{8}{3}\pi\right) = -1.732050807568881$$

Созданный нами алгоритм оказался на высоте: при $x<\pi$ полученные значения полностью совпадают с результатами работы функции tan(x), при $x>\pi$ различия появляются только в 14-м знаке после запятой.

В курсе математического анализа выводится соотношение, позволяющее оценить погрешность вычисления тангенса в точке x через разложение в цепную дробь для n итераций. Проверим с его помощью, какого порядка точности можно достичь с увеличением n.

$$\mathrm{err}(x, n) := \frac{x^{2n+1}}{\left[\prod_{i=1}^{n} (2 \cdot i - 1)\right]^2 \cdot (2n + 1)}$$

$$\mathrm{err}(\pi, 6)\ \mathrm{float}, 5 \rightarrow .20672\mathrm{e}{-2}$$

$$\mathrm{err}(\pi, 12)\ \mathrm{float}, 5 \rightarrow .10735\mathrm{e}{-11} \qquad \mathrm{err}(\pi, 15)\ \mathrm{float}, 5 \rightarrow .21721\mathrm{e}{-17}$$

Очевидно, с увеличением количества итераций точность расчета tg(x) возрастает очень быстро и погрешность стремится к нулю.

Как уже упоминалось выше, прямые тригонометрические функции определяются через тангенс половинного аргумента по известным формулам тригонометрии. В качестве примера приведем вычисление синуса и косинуса с использованием нашего алгоритма:

$$\text{my\_sin}(x) := \frac{2\cdot \text{tg}\left(\frac{x}{2}\right)}{1+\text{tg}\left(\frac{x}{2}\right)^2} \qquad \text{my\_cos}(x) := \frac{1-\text{tg}\left(\frac{x}{2}\right)^2}{1+\text{tg}\left(\frac{x}{2}\right)^2}$$

$$\text{my\_sin}\left(\frac{\pi}{3}\right) = 0.86602540378443855 \qquad \text{my\_cos}\left(\frac{\pi}{3}\right) = 0.50000000000000009$$

$$\sin\left(\frac{\pi}{3}\right) = 0.86602540378443855 \qquad \cos\left(\frac{\pi}{3}\right) = 0.50000000000000009$$

Как видите, созданные функции вычисляют синус и косинус ничуть не хуже встроенных.

Для вычисления арктангенса в окрестности нуля за n итераций используется ряд Маклорена:

$$\text{arctg}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \ldots + (-1)^n \cdot \frac{x^{2n+1}}{2n+1}$$

$$\text{arctg}(n,x) := \sum_{i=0}^{n}\left[(-1)^i \cdot \frac{x^{2i+1}}{2i+1}\right]$$

$$\text{arctg}(6,-0.1) = -0.09966865249116207 \qquad \text{arctg}(24,0.5) = 0.46364760900080588$$

$$\text{atan}(-0.1) = -0.09966865249116204 \qquad \text{atan}(0.5) = 0.46364760900080606$$

Все остальные обратные тригонометрические функции выражаются через арктангенс с помощью известных формул (в качестве примера приведем вычисление арксинуса):

$$\text{arcsin}(n,x) := \text{arctg}\left(n, \frac{x}{\sqrt{1-x^2}}\right)$$

$$\text{arcsin}(60,0.6) = 0.64350110879328435 \qquad \text{asin}(0.6) = 0.64350110879328435$$

18.6. Функции округления (Truncation and Round-Off)

Функции, позволяющие производить округление чисел, чрезвычайно широко используются в программировании. В Mathcad к этой категории относятся следующие функции:

- floor(x) — целая часть x (антье), то есть наибольшее целое число, не превосходящее x;
- Floor(x,y) — возвращает целую часть x, умноженную на y;

- ceil(x) — наименьшее целое число, большее либо равное x;
- Ceil(x,y) — наименьшее целое число, большее либо равное x/y, умноженное на y;
- trunc(x) — отбрасывает цифры после десятичной запятой;
- Trunc(x,y) — возвращает целую часть x/y, умноженную на y;
- round(x,n) — возвращает x, округленное до n десятичных позиций. Если n=0, то результатом будет округление до x до целого числа. При положительном n система отбросит все десятичные знаки, лежащие правее n-й позиции после запятой. Если же n будет определено как целое отрицательное число, то округление будет производиться в целой части x;
- Round(x,y) — возвращает округленное до целого знака x/y, умноженное на y.

Функции, имена которых начинаются с прописной буквы, связаны с обычными функциями округления следующим соотношением: F(x,y)=f(x/y)·y. Они оказываются полезными в том случае, если необходимо провести округление размерной величины, поскольку функции, имена которых начинаются со строчной буквы, не работают с размерностями. Чтобы округлить размерную величину, задайте в качестве второго параметра функции Floor(x,y), Ceil(x,y), Trunc(x,y) или Round(x,y) исходную единицу измерения. Помните, в независимости от указанной размерности система автоматически переведет размерную переменную в единицы определенной по умолчанию системы и выразит через них округленный результат. Поэтому, проводя округление величины, содержащей единицы измерения, выбирайте ту систему, в которой они являются базовыми (о том, как это сделать, рассказано в гл. 17).

Функции округления можно использовать и по отношению к комплексным числам. В данном случае операции округления проводятся по отдельности с действительной и мнимой частью.

Пример 18.3. Функции округления

$$\text{floor}(2.4) = 2 \qquad \text{ceil}(2.4) = 3 \qquad \text{trunc}(2.4) = 2$$

$$\text{floor}(-2.4) = -3 \qquad \text{ceil}(-2.4) = -2 \qquad \text{trunc}(-2.4) = -2$$

$$\text{floor}(2.6 + 1.4\cdot i) = 2 + i \qquad \text{ceil}(15.6 - 0.1\cdot i) = 16 \qquad \text{trunc}(0.7 + 1.3\cdot i) = i$$

Пример 18.4. Округление с заданными параметрами

$$\text{round}(234.583647, 0) = 235 \qquad \text{round}(234.583647, -1) = 230$$

$$\text{round}(234.583647, 3) = 234.584 \qquad \text{round}(234.583647, -2) = 200$$

$$\text{round}(234.583647, 5) = 234.58365 \qquad \text{round}(234.583647, -3) = 0$$

$$\text{round}(3.658 + 5.457\cdot i, 2) = 3.66 + 5.46i \qquad \text{round}(12.921 + 45.567\cdot i, -1) = 10 + 50i$$

Пример 18.5. Округление размерных величин

$$\text{Length} := 3.762\cdot m \qquad \text{Mass} := 75.95\cdot kg$$

$$\text{Floor}(\text{Length}, m) = 3\ m \qquad \text{Ceil}(\text{Mass}, kg) = 76\ kg$$

$$\text{Time} := 23.2\cdot s \qquad \text{Temp} := 273.15\cdot K$$

$$\text{Trunc}(\text{Time}, s) = 23\ s \qquad \text{Round}(\text{Temp}, K) = 273\ K$$

Кстати, производя округление, например, до 5-го знака, вы можете получить ответ с десятичной частью только из трех знаков. Связано это с тем, что по умолчанию все ответы численных расчетов отображаются в Mathcad с точностью до 0.001. Вопрос о том, как изменить формат численных результатов, весьма подробно рассматривается в гл. 2.

18.7. Функции условий и «ступенек» (Piecewise Continuous)

Функции этой группы используются прежде всего при создании алгоритмов в программировании и при всей своей простоте иногда могут быть незаменимы. Всего к этой категории относятся шесть функций.

- antisymmetrik tensor(i,j,k). Эта функция определяет, сколько перестановок нужно осуществить, чтобы преобразовать последовательность (0, 1, 2) в (i, j, k). Если количество перестановок четно, то функция возвращает 1, если нечетно то –1. Если же в заданной вами последовательности имеются повторения, то результатом работы рассматриваемой функции будет 0. Обязательным условием для параметров i, j, k является то, что они должны быть целыми числами от 0 до 2 (точнее, от ORIGIN до ORIGIN+2). В тексте документа указанная функция обозначается с помощью греческой буквы ε («ипсилон»: e+Ctrl+G).
- heaviside step(x). Ступенька Хевисайда. Возвращает 1, если x≥0, и 0 — в противном случае. Переменная x должна быть действительным числом. В документе обозначается греческой буквой ϕ («фи»: f+Ctrl+G).
- if(cond,x,y). Возвращает x, если логическое условие cond оказывается верным, и y — в противоположном случае.
- Kronecker delta (x,y). Дельта-символ Кронекера. Возвращает 1, если x=y, и 0 — во всех остальных случаях. В документ вводится как греческая δ («дельта»: d+Ctrl+G). Переменные x и y должны быть безразмерными целыми числами.
- sign(x). Функция сигнум. Возвращает 0 для x=0, 1 — для положительных x, –1 — для отрицательных. Переменная x должна быть действительным числом.
- until(x,y). Функция, возвращающая x до тех пор, пока выполняется условие y. Используется в расчетах с ранжированными переменными как аналог оператора цикла While.

Пример 18.6. Функции «ступенек» и условий

$$\varepsilon(0,1,2)=1 \quad \varepsilon(2,1,2)=0 \quad \varepsilon(0,2,1)=-1 \quad \varepsilon(2,1,0)=-1$$

$$\Phi(7)=1 \qquad \Phi(-1)=0 \qquad \Phi(0)=1$$

$$\delta(1,1)=1 \qquad \delta(1,0)=0$$

$$\mathrm{sign}(1)=1 \qquad \mathrm{sign}(0)=0 \qquad \mathrm{sign}(-1)=-1$$

$$f(x):=\mathrm{if}\left(\frac{\sin(x)}{x}=0,\text{"False"},\text{"True"}\right)$$

$$f(0)=\text{"False"} \qquad f\left(\frac{\pi}{2}\right)=\text{"True"}$$

$$f(\pi)=\text{"True"} \qquad f(\pi)\rightarrow\text{"False"}$$

Обратите внимание, что при численном и символьном вычислении функции if в приведенном примере были получены различные ответы. Подумайте, почему так произошло и какой из предложенных системой вариантов действительно верен.

18.8. Функции пользователя (User Defined)

Если вы профессиональный программист и владеете, например, C++, то при необходимости вы можете создавать различные приложения под Mathcad и самостоятельно. В том числе вы можете написать и свою функцию. Сохранив ее затем в специальной папке userEFI каталога программы в формате DLL, вы сможете использовать ее точно так же, как функции, созданные программистами Mathsoft. Так как специфика данного вопроса выходит далеко за пределы этой книги, то подробно о нем говорить мы не будем. Отметим лишь, что более подробную информацию вы можете получить в электронной книге Developer's References (Заметки для разработчиков) меню Help (Помощь) либо на сайте компании Mathsoft.

Чтобы категория User Defined не пустовала, создатели Mathcad занесли в нее две функции.

- kroneker(m,n). Произведение Кронекера для квадратных матриц m и n.

Пример 18.7. Произведение Кронекера

$$m := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad n := \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \qquad \text{kronecker}(m, n) = \begin{pmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{pmatrix}$$

- Psi(x). Пси-функция. Была введена для подсчета частичной суммы гармонического ряда. Определяется как:

$$\text{Psi}(x) = \frac{d}{dx} \ln(\Gamma(x))$$

Пример 18.8. Функция Psi

$$\sum_{i=1}^{100} \frac{1}{i} = 5.187 \qquad \sum_{i=1}^{n} \frac{1}{i} \rightarrow \text{Psi}(n + 1) + \gamma$$

$$\gamma := 0.577$$

$$\text{Psi}(101) + \gamma = 5.187 \qquad \text{PSI}(x) := \frac{d}{dx} \ln(\Gamma(x)) \qquad \text{PSI}(101) + \gamma = 5.187$$

18.9. Специальные функции (Special)

В этой категории собраны специальные функции, относящиеся к самым разным разделам математики, которым не нашлось места в более узких тематических категориях. Всего таких функций 14. Мы рассмотрим те функции, которые не упоминались в предыдущих главах.

- fhyper(a,b,c,x). Гипергеометрическая функция Гаусса. Является решением гипергеометрического уравнения:

$$x\cdot(1-x)\cdot\frac{d^2}{dx^2}y + [c-(a+b+1)\cdot x]\cdot\frac{d}{dx}y - a\cdot b\cdot y = 0$$

Для |x|<1 данная функция определяется с помощью гипергеометрического ряда:

$$F(a,b,c,x) = 1 + \frac{ab}{c}\cdot\frac{x}{1!} + \frac{a(a+1)\,b(b+1)}{c(c+1)}\cdot\frac{x^2}{2!} + \ldots + \frac{a(a+1)\ldots(a+n-1)\,b(b+1)(b+n-1)}{c(c+1)\ldots(c+n-1)}\cdot\frac{x^n}{n!}$$

Рассматриваемая функция очень важна для математики. Так, через нее, в частности, выражаются многие трансцендентные функции (см. пример 18.9). Все параметры для данной функции должны являться действительными числами, переменная x должна не превышать по абсолютной величине 1.

Пример 18.9. Гипергеометрическая функция Гаусса

$$x := 0.1$$

$$\ln(1+x) = 0.095 \qquad x\cdot\text{fhyper}(1,1,2,-x) = 0.095$$

$$\text{asin}(x) = 0.1 \qquad x\cdot\text{fhyper}\left(0.5,0.5,1.5,x^2\right) = 0.1$$

- mhyper(a,b,x). Конфлюэнтная гипергеометрическая функция. Является решением дифференциального уравнения вида

$$x\cdot\frac{d^2}{dx^2}y + (b-x)\cdot\frac{d}{dx}y - a\cdot y = 0$$

Все параметры для этой функции должны быть определены как действительные числа.

- Gamma(a,z). Гамма-функция Эйлера. Важнейшая трансцендентная аналитическая функция, распространяющая понятие факториала на нецелочисленные значения. Через гамма-функцию выражается огромное количество интегралов, специальных функций, бесконечных произведений, поэтому ее практическая и теоретическая значимость намного превышает важность любой другой встроенной функции этой категории.

В Mathcad подсчитать гамма-функцию можно в двух интерпретациях. Полная гамма-функция, определяемая несобственным интегралом:

$$\Gamma(z) = \int_0^\infty t^{z-1}\cdot e^{-t}\,dt$$

может быть подсчитана с помощью встроенной функции Γ(z) (аналогично некоторым другим рассмотренным выше функциям, обозначение гамма-функции в списке окна Insert Function (Вставить функцию) не совпадает с ее видом в документе). При этом z должна быть действительным или комплексным скаляром и не равняться 0, –1, –2, –3...

Также в Mathcad имеется встроенная функция Γ(a,x), позволяющая вычислять так называемую неполную гамма-функцию:

$$\Gamma(a,x)=\int_x^{\infty} t^{a-1}\cdot e^{-t}\,dt$$

Параметры a и x для этой функции должны быть положительными действительными числами.

Заготовка для вычисления обеих приведенных выше функций одна и та же. Если вам нужно будет найти значение обычной гамма-функции, то просто удалите в заготовке один из двух маркеров. Или лучше просто наберите нужный текст с клавиатуры (греческую Γ можно задать, введя латинскую G и нажав сочетание Ctrl+G).

При численном расчете полной гамма-функции на компьютере используется эффективный алгоритм аппроксимации, позволяющий с высокой точностью определять гамма-функцию для любого положительного аргумента (в том числе и комплексного, действительная часть которого больше либо равна 0). Расчетная формула метода:

$$\Gamma(x+1)=(x+\gamma+0.5)^{x+0.5}\cdot e^{-(x+\gamma+0.5)}\cdot\sqrt{2\pi}\cdot\left(p_0+\frac{p_1}{x+1}+\frac{p_2}{x+2}+\ldots+\frac{p_N}{x+N}\right)$$

При γ=5, N=6 и правильно подобранных параметрах p_n данный метод обеспечивает точность до 10-го знака мантиссы. Убедимся в этом, сравнив результаты, полученные с помощью встроенной функции Γ(x) и приведенной формулы.

$$p:=\begin{pmatrix} 1.000000000190015\\ 76.18009172947146\\ -86.50532032941677\\ 24.01409824083091\\ -1.231739572450155\\ 1.208650973866179\cdot10^{-3}\\ -5.395239384953\cdot10^{-6}\end{pmatrix}\qquad \mathrm{Gamma}(x):=\left[\frac{\sqrt{2\pi}}{x}\left(p_0+\sum_{i=1}^{6}\frac{p_i}{x+i}\right)\right]\cdot(x+5.5)^{x+0.5}\cdot e^{-(x+5.5)}$$

$$\mathrm{Gamma}(6)=120\qquad \Gamma(6)=120$$

$$\mathrm{Gamma}(20)=1.2164510041005978\times10^{17}\qquad \Gamma(20)=1.2164510040883201\times10^{17}$$

$$\mathrm{Gamma}(7+2\cdot i)=-428.51596944494986-315.40556017768529i$$

$$\Gamma(7+2\cdot i)=-428.51596944499607-315.40556017770183i$$

Для расчета неполной гамма-функции используется ряд, быстро сходящийся при значениях аргумента x, меньших a+1.

$$\text{Gamma}(a,x) := \Gamma(a) - e^{-x}\cdot x^{a}\cdot \sum_{n=0}^{25}\left(\frac{\Gamma(a)}{\Gamma(a+n+1)}\cdot x^{n}\right)$$

$$\text{Gamma}(5,3) = 19.566317868570529 \qquad \Gamma(5,3) = 19.566317868570533$$

При x>a+1 неполная гамма-функция вычисляется через разложение в цепную дробь.

$$\Gamma(a,x) = e^{-x}\cdot x^{a}\cdot\left[\cfrac{1}{x+1-a-\cfrac{1\cdot(1-a)}{x+3-a-\cfrac{2\cdot(2-a)}{x+5-a-\ddots-\cfrac{k\cdot(k-a)}{x+2k+1-a}}}}\right]$$

Ниже приведен пример рекурсивного алгоритма, вычисляющего неполную гамма-функцию с помощью цепной дроби (пояснение алгоритма см. в разд. 18.5). Отметим лишь то, что с увеличением a необходимо увеличивать и количество рекурсивных вызовов k, позволяющее получить результат с заданной точностью.

$$\begin{aligned}&\text{Gamma}(a,x):=\\&\quad k \leftarrow 7\\&\quad \text{brain}(\text{prev\_res},k) \leftarrow \begin{cases}\text{brain}\left[x+2\cdot k+1-a-\dfrac{(k+1)\cdot(k+1-a)}{\text{prev\_res}},k-1\right] & \text{if } k\geq 0\\ \text{prev\_res} & \text{otherwise}\end{cases}\\&\quad \text{return } \frac{1}{\text{brain}(x+2k+1-a,k)}\cdot e^{-x}\cdot x^{a}\end{aligned}$$

$$\text{Gamma}(7,5) = 548.77209334051599 \qquad \Gamma(7,5) = 548.77209334051588$$

Каждый из методов требует порядка $a^{1/2}$ итераций для обеспечения сходимости (максимальное их количество требуется при x близких к a, где гамма-функция изменяется быстрее всего).

С гамма-функцией работает и символьный процессор, поэтому она может принимать участие как в численных, так и в символьных расчетах.

Пример 18.10. Гамма-функция в расчетах различных типов (рис. 18.5)

$$\Gamma(6) = 120 \qquad 5! = 120 \qquad \int_0^{\infty} t^{5}\cdot e^{-t}\,dt = 120$$

$$\Gamma\left(\frac{1}{2}\right) \rightarrow \pi^{\frac{1}{2}} \qquad \Gamma(-3.1) = 1.492 \qquad \Gamma(1+i) = 0.498 - 0.155i$$

$$\frac{d}{dx}\Gamma(x) \rightarrow \text{Psi}(x)\cdot\Gamma(x) \qquad \text{Psi}(x)\cdot\Gamma(x)\ \text{solve},x \rightarrow (1.4616321449683623413)$$

$$\Gamma(3.1)\cdot 3.1 = 6.813 \qquad \Gamma(3.1+1) = 6.813 \qquad \Gamma(a,0) \rightarrow \Gamma(a)$$

$$\Gamma(3.1)\cdot\Gamma(1-3.1) = -10.166 \qquad \pi\cdot\csc(\pi\cdot 3.1) = -10.166$$

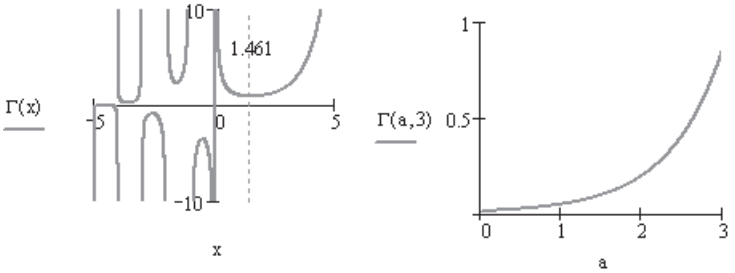

Рис. 18.5. Гамма-функция

- Her(n,x). Функция возвращает значение ортогонального полинома Эрмита степени n в точке x (n — целое положительное число, x — действительная величина). Полиномы Эрмита являются решением следующего дифференциального уравнения:

$$\frac{d^2}{dx^2}y - 2\cdot x\cdot\left(\frac{d}{dx}y\right) + 2\cdot n\cdot y = 0$$

Полиномы Эрмита в Mathcad можно вычислить и в аналитическом виде.

Пример 18.11. Полиномы Эрмита (рис. 18.6)

$$H(x,n) := (-1)^n\cdot e^{x^2}\cdot\frac{d^n}{dx^n}e^{-x^2}$$

$$H(x,1) \rightarrow 2\cdot e^{x^2}\cdot x\cdot e^{-x^2} \qquad 2\cdot e^{x^2}\cdot x\cdot e^{-x^2}\ \text{simplify}\ \rightarrow 2\cdot x$$

$$H(x,2) \rightarrow e^{x^2}\cdot\left(-2\cdot e^{-x^2} + 4\cdot x^2\cdot e^{-x^2}\right)$$

$$e^{x^2}\cdot\left(-2\cdot e^{-x^2} + 4\cdot x^2\cdot e^{-x^2}\right)\ \text{simplify}\ \rightarrow -2 + 4\cdot x^2$$

$$H(x,3) \rightarrow -e^{x^2}\cdot\left(12\cdot x\cdot e^{-x^2} - 8\cdot x^3\cdot e^{-x^2}\right)$$

$$-e^{x^2}\cdot\left(12\cdot x\cdot e^{-x^2} - 8\cdot x^3\cdot e^{-x^2}\right) \rightarrow -e^{x^2}\cdot\left(12\cdot x\cdot e^{-x^2} - 8\cdot x^3\cdot e^{-x^2}\right)$$

- ibeta(a,x,y). Эта функция служит для вычисления неполной бета-функции с параметром a, которая определяется с помощью следующего выражения:

$$\text{ibeta}(a,x,y) = \frac{\Gamma(x+y)}{\Gamma(x)\cdot\Gamma(y)}\cdot\int_0^a t^{x-1}\cdot(1-t)^{y-1}\,dt$$

Переменные x и y для данной функции должны быть положительными действительными числами, параметр a должен быть определен как положительная величина, принадлежащая интервалу от 0 до 1. Данная функция может участвовать и в символьных вычислениях.

Неполная бета-функция используется в основном в некоторых выкладках теории вероятностей. Как и в случае неполной гамма-функции, численный расчет неполной бета-функции проводится с применением цепной дроби.

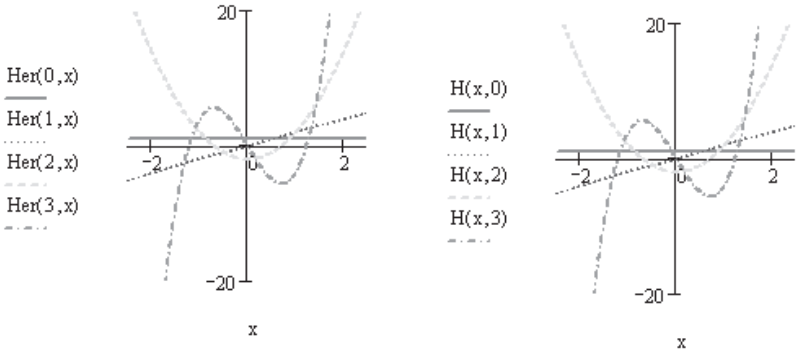

Рис. 18.6. Полиномы Эрмита

- Jac(n,a,b,x). Эта функция вычисляет значение полинома Якоби степени n в точке x с параметрами a и b. Степень n должна быть задана целым положительным числом, x должен быть действительной величиной, a и b — действительными числами больше –1.

  Многочлены Якоби являются частным случаем гипергеометрической функции и были получены при решении следующего дифференциального уравнения:

$$\left(1 - x^2\right)\cdot\frac{d^2}{dx^2}y + [b - a - (a + b + 2)\cdot x]\cdot\frac{d}{dx}y + n\cdot(n + a + b + 1)\cdot y = 0$$

  Используя вычислительные возможности символьного процессора Mathcad, полиномы Якоби можно рассчитывать и в аналитическом виде.

Пример 18.12. Аналитический расчет полиномов Якоби (рис. 18.7)

$$P(x,n,a,b) := \frac{(-1)^n}{2^n\cdot n!}\cdot\left[(1-x)^{-a}\cdot(1+x)^{-b}\right]\cdot\frac{d^n}{dx^n}\left[(1-x)^{a+n}\cdot(1+x)^{b+n}\right]$$

$$P(x,2,3,3)\ \text{simplify}\ \rightarrow \frac{-5}{4} + \frac{45}{4}\cdot x^2$$

$$P(x,3,1,1)\ \text{simplify}\ \rightarrow x\cdot\left(-3 + 7\cdot x^2\right)$$

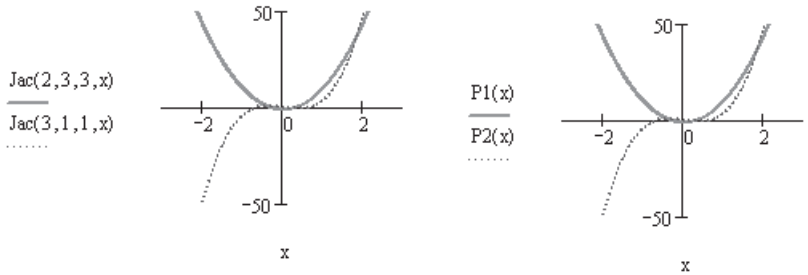

Рис. 18.7. Полиномы Якоби

- Tcheb(n,x). Полином Чебышева 1-го рода порядка n (рис. 18.8, слева). Частный случай полинома Якоби, полученный при решении дифференциального уравнения вида

$$\left(1 - x^2\right)\cdot\frac{d^2}{dx^2}y - x\cdot\frac{d}{dx}y + n^2\cdot y = 0$$

  Аналогично функции Jac, x для Tcheb должен быть действительной величиной, n — целым положительным числом.

- Ucheb(n,x). Полином Чебышева 2-го рода порядка n (рис. 18.8, справа). Является решением дифференциального уравнения вида

$$\left(1 - x^2\right)\cdot\frac{d^2}{dx^2}y - 3x\cdot\frac{d}{dx}y + n\cdot(n + 2)\cdot y = 0$$

  Параметры для Ucheb задаются по тем же правилам, что и для Tcheb.

  При необходимости в Mathcad можно найти и аналитический вид полинома Чебышева для данного набора параметров. В принципе, для этого даже можно использовать приведенный в примере 18.12 алгоритм.

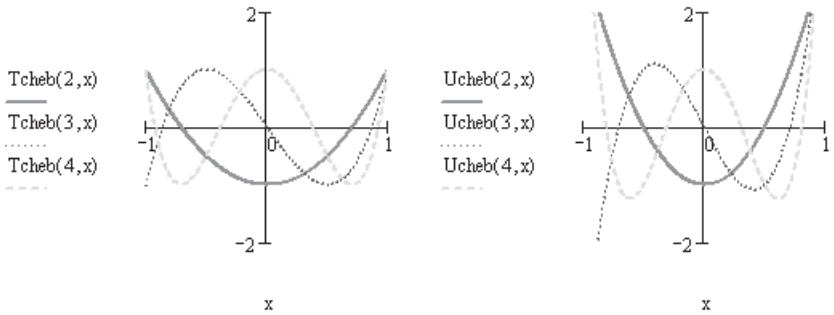

Рис. 18.8. Полиномы Чебышева

- Leg(n,x). Полином Лежандра степени n. Частный случай полинома Якоби, который был получен при решении следующего дифференциального уравнения:

$$\left(1 - x^2\right)\cdot\frac{d^2}{dx^2}y - 2x\cdot\frac{d}{dx}y + n\cdot(n + 1)\cdot y = 0$$

  Параметр порядка n должен быть целым положительным числом, переменная x — действительным скаляром.

- Lag(n,x). Многочлен Лагерра степени n в точке x. Является решением дифференциального уравнения вида:

$$x\cdot\frac{d^2}{dx^2}y + (1 - x)\cdot\frac{d}{dx}y + n\cdot y = 0$$

  Параметры для функции Lag задаются по таким же правилам, что и для функции Leg (рис. 18.9).

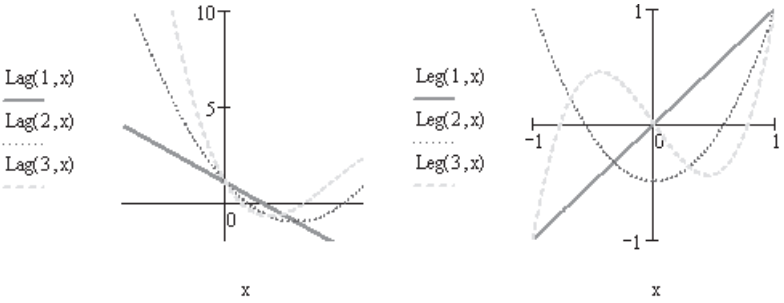

Рис. 18.9. Полиномы Лагерра и Лежандра

18.10. Строковые функции (String)

В Mathcad строки используются в основном для создания сообщений об ошибках или удобных для восприятия ответов. В приведенных ранее примерах мы не раз применяли строчные переменные, однако со встроенными функциями для работы с выражениями такого типа встречались лишь однажды (в гл. 5, при представлении комплексных чисел в тригонометрической и экспоненциальной форме). А между тем в Mathcad имеются довольно неплохие системные средства для работы со строками, позволяющие легко решать многие непростые задачи. В этом разделе мы поговорим о встроенных строковых функциях, а также приведем несколько примеров их использования.

Всего в категории String (Строковые) имеется 10 функций.

- concat(S1,S2,S3….). Пожалуй, самая важная строковая функция, позволяющая объединить несколько строк в одну. Последовательность слияния строк определяется порядком определения их имен в скобках concat. При работе с функцией concat следует учитывать, что пробел также рассматривается как символ.

Пример 18.13. Слияние строк

$$S1 := \text{"The variable must be a "} \qquad S2 := \text{"STRING"}$$

$$\text{concat}(S1, S2) = \text{"The variable must be a STRING"}$$

- error(S). С этой функцией мы уже встречались в гл. 4, посвященной программированию. С помощью функции error можно создавать характерные для Mathcad сообщения об ошибках на всплывающих желтых панелях (рис. 18.10).

$$Sec(x) := \text{if}\left(|\sin(x)| < 10^{-10}, \text{error("Found a singularity")}, \frac{1}{\sin(x)}\right)$$

$Sec(4) = -1.321$ $\quad Sec(\pi) =$ ∎

Found a singularity

Рис. 18.10. Использование функции error

На рис. 18.10 приведен пример создания функции, вычисляющей секанс. Данная функция не определена при sin(x)=0, поэтому мы, чтобы предупредить возможность некорректной работы алгоритма, должны предусмотреть эту ситуацию. Однако если в качестве условия вывода сообщения об ошибке мы используем строгое равенство синуса нулю, то при попытке найти секанс от π или от 2π будет получен совершенно неверный результат:

$$\mathrm{Sec}(\pi) = 8.166 \times 10^{15} \qquad \mathrm{Sec}(2\pi) = -4.083 \times 10^{15}$$

Причина возникшей ошибки заключается в том, что численный процессор Mathcad вычисляет специальные функции не точно, а приближенно. Для таких простых функций, как синус, ошибка возникает где-то в 14–15-м знаке, для более же сложных выражений она может быть куда значительнее. Конечно, при обычных расчетах погрешность в 15-м знаке не имеет ровным счетом никакого значения, однако в таких специфических задачах, как приведенная выше, она может приводить к некорректной работе алгоритма. Это связано с тем, что при использовании логического равенства числа считаются равными, если они совпадают вплоть до 15-го знака.

Обойти описанную проблему можно двумя способами:

- определив условием вывода сообщения об ошибке появление некоторой малой, но не нулевой величины синуса;
- сохранив прежний вид программы, производить расчет символьно. Этот способ не универсален, так что на практике его вряд ли стоит использовать.

Определив границу нуля для синуса как 10^{-10}, мы получим корректный ответ.

- IsString(x). Используя эту функцию, можно определить, является ли x строкой. Возвращает 1, если x — строковая переменная, и 0 — во всех остальных случаях.
- strlen(S). Эта функция служит для определения количества знаков в строке S, в том числе и пробелов.
- substr(S,m,n). Эта функция предназначена для выделения из строки S подстроки из n знаков, начиная с позиции m.
- search(S,Subs,m). Эта функция служит для определения стартовой позиции подстроки Subs в строке S. Параметр m — это позиция в строке S, с которой начинается поиск. Если функция search не находит нужной подстроки, то она возвращает в качестве ответа значение, на единицу меньшее ORIGIN.

В Mathcad 12 появилась возможность индексации символов в строках с помощью системной переменной ORIGIN. Для этого необходимо активировать параметр Use ORIGIN for string indexing (Использовать ORIGIN для индексирования в строках), расположенную на вкладке Calculation (Вычисления) окна Worksheet Options (Параметры документа) (открывается с помощью соответствующей команды меню Tools (Инструменты)). По умолчанию отсчет символов ведется с нуля, однако вы можете поменять нумерацию символов строки, присвоив выше переменной ORIGIN новое значение.

Пример 18.14. Функции IsString, strlen, substr и search

$$S := \text{"Yes or No"} \qquad \mathrm{IsString}(S) = 1$$

$$\mathrm{strlen}(S) = 9 \qquad \mathrm{substr}(S, 0, 3) = \text{"Yes"} \qquad \mathrm{search}(S, \text{"No"}, 0) = 7 \qquad \mathrm{search}(S, \text{"No!"}, 0) = -1$$

$$ORIGIN := 1$$

$$\mathrm{substr}(S, 1, 3) = \text{"Yes"} \qquad \mathrm{search}(S, \text{"No"}, 1) = 8 \qquad \mathrm{search}(S, \text{"No!"}, 1) = 0$$

- num2str(z). Эта функция служит для перевода числа z в строковое представление. Такая операция может быть полезна в том случае, если работать со строкой проще, чем с числом (например, в кодировании).
- str2num(S). Как можно догадаться по названию, эта функция выполняет задачу, обратную назначению num2str, а именно: переводит число из строкового формата в числовой.

Пример 18.15. Преобразование формата числа

$$Z := \text{num2str}(\pi) \qquad Z = \text{"3,14159265358979"}$$

$$\text{str2num("3.14159265358979")} = 3.142$$

- str2vec(S). Эта функция преобразует строку S в вектор соответствующих каждому ее элементу кодов ASCII.
- vec2str(v). Переводит вектор кодов ASCII в строку обычных символов.

Пример 18.16. Перевод числа в ASCII-код

$$Z := \text{num2str}(\pi) \qquad v := \text{str2vec}(Z)$$

$v^T =$

	0	1	2	3	4	5	6	7	8	9
0	51	44	49	52	49	53	57	50	54	53

$$\text{vec2str(v)} = \text{"3,14159265358979"}$$

С первого взгляда, функции ASCII кодов могут показаться абсолютно бесполезными. Однако на самом деле это совсем не так. Помимо чисто познавательного значения, они играют важную роль в кодировании информации.

Пример 18.17. Кодирование и декодирование числа π

$$Z := \text{num2str}(\pi) \qquad v := \text{str2vec}(Z)$$

$$\text{Vkod(v)} := \left| \begin{array}{l} \text{for } i \in 0..\text{last(v)} \\ \quad \left| \begin{array}{l} V_i \leftarrow v_i - \text{round}\left(\sqrt{v_i}\right) \text{ if } v_i \geq 128 \\ V_i \leftarrow v_i + \text{round}\left(\sqrt{v_i}\right) \end{array} \right. \\ V \end{array} \right.$$

$\text{Vkod(v)}^T =$

	0	1	2	3	4	5	6	7	8	9
0	58	51	56	59	56	60	65	57	61	60

$$v1 := \text{Vkod(v)}$$

$$\text{Letter} := \text{vec2str(v1)} \qquad \text{Letter} = \text{":38;8<A9=<:<?A>A"}$$

$$f(x) := x^2 + \text{floor}(x) \qquad x + \text{floor}\left(\sqrt{x}\right) = 60 \text{ solve}, x \rightarrow 53.0000000000000000$$

$$\text{Antikod(v)} := \left| \begin{array}{l} \text{for } i \in 0..\text{last(v)} \\ \quad \left| \begin{array}{l} x \leftarrow \sqrt{v_i} \\ V_i \leftarrow \text{root}\left(f(x) - v_i, x\right) \\ V_i \leftarrow \left(V_i\right)^2 \end{array} \right. \\ V \end{array} \right.$$

$$Z1 := \text{vec2str(Antikod(v1))} \qquad Z1 = \text{"3,14159265358979"}$$

18.11. Финансовые функции (Finance)

Функции, позволяющие проводить всевозможные финансовые расчеты, появились в Mathcad 2000. Так как область их применения более нежели специфична (на практике, как правило, для выполнения такого рода работы используют специализированные пакеты, такие как 1С: Бухгалтерия), то подробно на них останавливаться мы не будем и ограничимся примером использования одной из них. При необходимости более подробную информацию вы сможете получить в справочной системе программы. Всего в Mathcad имеется 18 функций, предназначенных для финансовых расчетов.

- cnper(rate,pv,fv). Определяет количество периодов, требуемых для получения некоторого значения вклада, если известен его текущий размер и процент начислений. Здесь:
  - rate — фиксированная учетная ставка, может быть представлена как в виде дроби, так и в виде числа с процентом (например, 0.05 и 5%);
  - pv — текущий размер вклада;
  - fv — необходимый размер вклада.

Пример 18.18. Начисления по вкладу «Прибыльный» составляют 12% годовых. Через сколько лет размер вклада увеличится вдвое, если первоначальная сумма вклада равнялась $10 000?

$$\text{cnper}(12\%, 10000, 20000) = 6.116 \qquad 6.116 \cdot 365 = 2.232 \times 10^3$$

Размер вклада увеличится вдвое через 6,116 года или 2232 дня.

Полный список финансовых функций с кратким пояснением назначения каждой из них можно найти в приложении в конце книги.