ДОКУМЕНТАЦИЯ ПО РАБОТЕ С БИБЛИОТЕКОЙ taylor.dll

Общее описание

TAYLOR - программная система, открывающая точные и простые средства для широкого использования в вычислительной практике методов, связанных с представлением функций рядами Тейлора, методов асимптотических разложений, а также прямых методов высших производных.

TAYLOR разработан в Центральном экономико-математическом институте РАН. В основе системы лежит компьютерная «Алгебра дифференцирования» (АД) - совокупность специальных рекуррентных алгоритмов, автором которых является д.ф-м.н., проф. В.З. Беленький. Алгоритмы предназначены для вычисления по формулам дифференциального исчисления точного значения производных (произвольных порядков) от функций, заданных формулами любой сложности, начиная от суперпозиции элементарных функций и заканчивая функциями, заданными неявно.

Разработка алгоритмов «Алгебры дифференцирования» началась в 80-90-х годах, когда в среде DOS была разработана первая версия система TAYLOR. В 2009 году система была модернизирована для работы в Windows, создан интерфейс, позволяющий решать задачи в интерактивном режиме (TAYLOR-3).

Четвертая (итоговая) версия системы, TAYLOR-4, представляет собой динамическую библиотеку (taylor.dll), подключение которой дает возможность исследователю использовать в своей программной среде (Pascal, Delphi, Embarcadero) алгоритмы TAYLOR.

Автор алгоритмов: В.З. Беленький

Pascal-программа: О.А. Васильева

Интерактивная версия (в среде Windows): М.А. Милкова при консультациях В.Л. Ушковой

Dll-библиотека: М.А. Милкова

Обзор доступных процедур и функций

Каталог процедур можно разбить на процедуры двух уровней: Базовый уровень и Прикладной уровень.

Процедуры Базового уровня получают коэффициенты ряда Тейлора различных функций.

В Прикладной уровень включены процедуры, предназначенные для решения некоторых задач вычислительной математики.

Базовый уровень

В Базовый уровень включены процедуры, результатом которых являются коэффициенты $\{c_k\}$ отрезка ряда Тейлора разложения данной функции y:

$$y(x) = \sum_{k=0}^{n} c_k \, (x - x_0)^k \ ,$$

$$c_k := \frac{1}{k!} \cdot y^{(k)}(x_0) \ ;$$

Исходной информацией для процедур является:

- Идентификаторы переменных, с помощью которых записана формула функции
- Формула соответствующей функции. Функция y может быть задана различными способами; каждая из базовых процедур отвечает тому или иному способу (о способах задания функции y см. ниже).
- Начальные значения переменных, определяющих точку разложения ряда Тейлора
- Порядок разложения n $(n \leq 10)$.

Процедуры базового уровня (способы задания функции y)

Примечание: Здесь и далее мы используем идентификаторы x, y, t, однако, как было замечено выше, пользователь может выбирать любые идентификаторы для функций и их аргументов.

1.1 Явная функция. Задается явная формула функции $y = f(x)$

1.2 Параметрическая функция. Функция функции $y(x)$ определяется соотношениям $\mathrm{x} = \mathrm{f}_1(\mathrm{t}),\ \mathrm{y} = \mathrm{f}_2(\mathrm{t})$.

1.3 Неявная функция. Функция $\mathrm{y}(\mathrm{x})$ определяется тождеством $\mathrm{F}(\mathrm{x},\mathrm{y}) = \mathrm{F}(\mathrm{x}_0, \mathrm{y}_0)$.

1.4 Обратная функция. Разложение строится для функции $\mathrm{y}(\mathrm{x})$, обратной к заданной функции $\mathrm{x} = \mathrm{g}(\mathrm{y})$.

1.5 Функция, заданная обыкновенным дифференциальным уравнением (ОДУ) первого порядка. Функция $\mathrm{y}(\mathrm{x})$ определяется как интегральная кривая обыкновенного дифференциального уравнения:

$$y' = G(x, y), \qquad y(x_0) = y_0$$

1.6 Функция, заданная ОДУ высокого порядка m $(\mathrm{m} \leq 5)$. Функция $\mathrm{y}(\mathrm{x})$ определяется дифференциальным уравнением высокого порядка

$$\begin{cases} y^{(m)} = G(x, y, y', \ldots, y^{(m-1)}) \\ y(x_0) = y_{01} \\ y'(x_0) = y_{02} \\ \ldots \\ y^{(m-1)}(x_0) = y_{0m} \end{cases} \qquad m \leq 5$$

.

1.7 Вектор-функция размерности m $(\mathrm{m} \leq 5)$, заданная системой дифференциальных уравнений. Функция $\mathrm{y}(\mathrm{x})$ определяется дифференциальным уравнением высокого порядка

$$\begin{cases} y^{(m)} = G(x, y, y', \ldots, y^{(m-1)}) \\ y(x_0) = y_{01} \\ y'(x_0) = y_{02} \\ \ldots \\ y^{(m-1)}(x_0) = y_{0m} \end{cases} \qquad m \leq 5$$

.

1.8 Функция, заданная сингулярным ОДУ порядка m ($m \leq 5$). Функция $y(x)$ определяется дифференциальным уравнением с нулевым коэффициентом при старшей производной

$$\begin{cases} x \cdot y^{(m)}(x) = G[x, y, y', \dots, y^{(m-1)}] \\ x_0 = 0 \\ y(x_0) = y_{01} \\ y'(x_0) = y_{02} \\ \dots \\ y^{(m-1)}(x_0) = y_{0m} \end{cases} \qquad m \leq 4$$

.

Прикладной уровень

В Прикладной уровень включены процедуры, предназначенные для решения некоторых типовых и специальных задач. Прикладные процедуры можно разбить на 3 подуровня: Типовые задачи, Интегрирование, Вычисление таблиц.

Типовые задачи:

2.1 Вычисление корня функции в окрестности начальной точки, заданной способами 1.1-1.3 Базового уровня. Процедура находит корень уравнения в окрестности заданного начального значения.

2.2 Вычисление точки экстремума функции в окрестности начальной точки, заданной способами 1.1-1.3 Базового уровня

2.3 Решение системы двух уравнений в окрестности начального приближения (x_0, y_0)

$$\begin{cases} F_1(x,y) = 0 \\ F_2(x,y) = 0 \end{cases}$$

2.4 Вычисление определенного интеграла для заданной функции $f(x)$ и пределов интегрирования a, b :

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

2.5 Вычисление несобственного интеграла от быстроосциллирующей функции, заданной явно

$$Int = w \cdot \int_{x_0}^{\infty} f(x) \sin(wx)\, dx,$$

$f(x)$ – достаточно гладкая функция, экспоненциально убывающая на бесконечности, $w \gg 1$ - частота осцилляций (колебаний);

Интегрирование:

2.6 Интегрирование – решение задачи Коши (нахождение значения функции на правом конце) для дифференциальных уравнений, заданных способами 1.5-1.8

Вычисление таблиц:

2.7 Вычисление таблиц – вычисление таблиц значений функции $y(x)$ (задаваемой одним из способов 1.1-1.7) на некотором отрезке аргумента x(равномерная сетка с конечным числом точек), содержащем начальную точку x_0. В частности, получение такого рода таблиц позволяет строить графики функции на заданном отрезке, в том числе графики неявных функций.

Различные варианты задания процедур

Для некоторых процедур (процедуры 1.1-1.7, 2.1) допустимо использовать различные варианты задания функций:

Простой вариант - формулы функций содержат только основные переменные.

Вариант с параметрами - в записи формул могут присутствовать несколько (не более пяти) числовых параметров.

Блочный вариант – допускает в записи формул идентификаторы промежуточных функций (не более пяти, формулы промежуточных функций также задаются пользователем).

Блочно-параметрический вариант – объединяет вариант с параметрами и вариант с блоками (в том числе, в формулах промежуточных функций допускается использование параметров).

Синтаксис обращений к процедурам

Базовые процедуры

Далее представлены способы обращения к процедурам базового уровня:

1.1 Явная функция

Общие обозначения: xS - идентификатор переменной; x0 – начальное значение переменной xS; f – формула функции; n- порядок разложения; result – выходной массив коэффициентов.

Простой вариант:

Explicit (xS: char; x0: real; f: string3; n: byte; var result: arrofreal);

Вариант с параметрами:

ExplicitP (xS: char; x0: real; f: string3; n: byte; PS: string1; Zn: arrofreal1; var result: arrofreal);

где $f = f(xS, PS)$; PS – строка, содержащая идентификаторы используемых параметров; Zn – массив значений параметров.

Вариант с блоками:

ExplicitB(xS: char; x0: real; BS: string1; f: string3; gS: arrs7; n: byte; var result: arrofreal);

где $f = f(xS, BS)$; BS – строка, содержащая идентификаторы используемых блочных функций (блоков); gS – массив для задания формул блоковых функций.

Вариант с блоками и параметрами:

ExplicitPB(xS: char; x: real; BS: string1; f: string3; gS: arrs7; n: byte; PS: string1; Zn: arrofreal1; var result: arrofreal);

где $f = f(xS, BS, PS)$

1.2 Обратная функция

Общие обозначения: yS - идентификатор переменной; y0 – начальное значение переменной yS; g – формула функции; n- порядок разложения; result – выходной массив коэффициентов.

Invers (yS: char; y0: real; g: string3; n: byte; var result: arrofreal);

InversP (yS: char; y0: real; g: string3; n: byte; PS: string1; Zn: arrofreal1; var result: arrofreal);

где $\mathrm{g} = \mathrm{g}(\mathrm{yS}, \mathrm{PS})$; PS – строка, содержащая идентификаторы используемых параметров; Zn – массив значений параметров.

InversB (yS: char; y0: real; BS: string1; g: string3; gS: arrs7; n: byte; var w: arrofreal);

где $\mathrm{g} = \mathrm{g}(\mathrm{yS}, \mathrm{BS})$; BS – строка, содержащая идентификаторы используемых блочных функций (блоков); gS – массив для задания формул блоковых функций.

InversPB (yS: char; y0: real; BS: string1; g: string3; gS: arrs7; n: byte; PS: string1; Zn: arrofreal1; var result: arrofreal);

где $g = g(yS, BS, PS)$

1.3 Параметрическая функция

Общие обозначения: tS - идентификатор параметрической переменной; t0 – начальное значение переменной tS; f – массив для задания формул параметрических функции; n- порядок разложения; result – выходной массив коэффициентов.

Paramet (tS: char; t0: real; f: arrofstring; n: byte; var result: array of real);

где $f: f_1(tS), f_2(tS)$;

ParametP(tS: char; t0: real; f: arrs7; n: byte; PS: string1; Zn: arrofreal1; var result: arrofreal

где f: $f_1(tS, PS), f_2(tS, PS)$; PS – строка, содержащая идентификаторы используемых параметров; Zn – массив значений параметров.

ParametB(tS: char; t0: real; BS: string1; f: arrs7; gS: arrs7; n: byte; var result: arrofreal);

где f: $f_1(tS, BS), f_2(tS, BS)$; BS – строка, содержащая идентификаторы используемых блочных функций; gS – массив для задания формул блоковых функций.

ParametPB (tS: char; t0: real; f: arrofstring; BS: string1; gS: arrofstring; n: byte; PS: string1; Zn: arrofreal1; var result: arrofreal);

где f: $f_1(tS, BS, PS), f_2(tS, BS, PS)$

1.4 Неявная функция

Общие обозначения: xS, yS – идентификаторы переменных; x0, y0 – начальные значения переменных; f – формула функции, задаваемая уравнением $F(xS, yS) = F(x_{0,}y_0)$; n- порядок разложения; result – выходной массив коэффициентов.

Implicit(xS,yS: char; x0,y0: real; f: string3; n: byte; var result: array of real);

ImplicitP(xS,yS: char; x,y: real; f: string3; n: byte; PS: string1; Zn: arrofreal1; var w: arrofreal);

где $f = f(xS, \mathrm{yS}, \mathrm{PS})$; PS – строка, содержащая идентификаторы используемых параметров; Zn – массив значений параметров.

ImplicitB (xS,yS: char; x0,y0: real; BS: string1; f: string3; gS: arrs7; n: byte; var result: arrofreal);

где $f = f(xS, yS, BS)$; BS – строка, содержащая идентификаторы используемых блочных функций (блоков); gS – массив для задания формул блоковых функций.

ImplicitPB(xS,yS: char; x,y: real; BS: string1; f: string3; gS: arrs7; n: byte; PS: string1; Zn: arrofreal1; var w: arrofreal);

где $f = f(xS, yS, BS, PS)$

1.5 Функция, заданная ОДУ первого порядка

Общие обозначения: xS, yS – идентификаторы переменных; x0, y0 – начальные значения переменных; f – формула функции $G(xS, yS)$ (правая часть ОДУ); n- порядок разложения; result – выходной массив коэффициентов.

Diff1 (xS,yS: char; x0,y0: real; f: string3; n: byte; var result: arrofreal);

Diff1P(xS,yS: char; x,y: real; f: string3; n: byte; PS: string1; Zn: arrofreal1; var result: arrofreal);

где $f = G(xS, \mathrm{yS}, \mathrm{PS})$; PS – строка, содержащая идентификаторы используемых параметров; Zn – массив значений параметров.

Diff1B(xS,yS: char; x,y: real; BS: string1; f: string3; gS: arrs7; n: byte; var result: arrofreal);

где $f = f(xS, yS, BS)$; BS – строка, содержащая идентификаторы используемых блочных функций (блоков); gS – массив для задания формул блоковых функций.

Diff1PB(xS,yS: char; x,y: real; BS: string1; f: string3; gS: arrs7; n: byte; PS: string1; Zn: arrofreal1; var result: arrofreal);

где $f = f(xS, yS, BS, PS)$

1.6 Функция, заданная ОДУ высокого порядка $(\boldsymbol{m \leq 5})$

Общие обозначения: xS – идентификатор основной переменной; x0 – начальное значение; LS – список идентификаторов для искомой функции и ее производных до порядка (m-1) (длина списка определяет значение m); Yn – массив значений функции и ее производных в начальной точке; f – формула функции $F(xS, LS)$; n- порядок разложения; result – выходной массив коэффициентов.

Diffh (xS: char; x0: real; LS: string1; Yn: arrofreal1; f: string3; n: byte; var result: arrofreal);

DiffhP(xS: char; x: real; LS: string1; Yn: arrofreal1; f: string3; n: byte; PS: string1; Zn: arrofreal1; var w: arrofreal);

где $f = F(xS, \mathrm{LS}, \mathrm{PS})$; PS – строка, содержащая идентификаторы используемых параметров; Zn – массив значений параметров.

DiffhB(xS: char; x: real; LS: string1; Yn: arrofreal1; f: string3; n: byte; BS: string1; vS: arrs7; var w: arrofreal);

DiffhPB(xS: char; x: real; LS: string1; Yn: arrofreal1; f: string3; n: byte; BS: string1; vS: arrs7; PS: string1; Zn: arrofreal1; var w: arrofreal);

1.7 Вектор-функция, заданная системой дифференциальных уравнений порядка $m \le 5$

Diffs (xS: char; x0: real; LS: string1; Yn: arrofreal1; g: arrs7; n: byte; var result_m: arrofreal2)

xS – идентификатор основной переменной; x0 – начальное значение; LS – список идентификаторов, используемых для записи функций в системе (длина списка определяет значение m); Yn – массив значений функций в начальной точке; g – массив формул функций в системе; n- порядок разложения; $\mathrm{result\_m}$ – m-мерный выходной массив коэффициентов.

DiffsP(xS: char; x: real; LS: string1; Yn: arrofreal1; g: arrs7; n: byte; PS: string1; Zn: arrofreal1; var sf: arrofreal2);

Где g- массив формул функций g_iв системе, $g_i = G(xS, \mathrm{LS}, \mathrm{PS})$; PS – строка, содержащая идентификаторы используемых параметров; Zn – массив значений параметров.

DiffsB(xS: char; x: real; LS: string1; Yn: arrofreal1; g: arrs7; n: byte; BS: string1; vS: arrs7; var sf: arrofreal2);

Где g- массив формул функций g_iв системе, $g_i = G(xS, \mathrm{LS}, BS)$; BS – строка, содержащая идентификаторы используемых блочных функций (блоков); gS – массив для задания формул блоковых функций.

DiffsPB(xS: char; x: real; LS: string1; Yn: arrofreal1; g: arrs7; n: byte; BS: string1; vS: arrs7; PS: string1; Zn: arrofreal1; var sf: arrofreal2);

Где g- массив формул функций g_iв системе, $g_i = G(xS, BS, PS)$

1.8 Функция, заданная сингулярным ОДУ порядка $m \le 5$

SingODE (xS: char; LS: string1; Yn: arrofreal1; f: string3; n: byte; var result: arrofreal);

xS – идентификатор основной переменной; x0 не задается, оно по умолчанию равно 0; LS – список идентификаторов для искомой функции и ее производных до порядка (m-1) (длина списка определяет значение m); Yn – массив значений функции и ее производных в начальной точке; f – формула функции $F(xS, LS)$; n- порядок разложения; result – выходной массив коэффициентов.

Синтаксис обращений к процедурам Прикладного уровня

Далее представлены способы обращения к процедурам Прикладного уровня:

2.1 Вычисление корня функции

Явная функция:

Общие обозначения xS - идентификатор переменной; x0 – начальная точка, в окрестности которой находится корень уравнения; f – формула функции; n- порядок разложения; eps – точность вычисления (погрешность по значению функции) $x1$ – найденный корень уравнения.

Root (xS: char; x0: real; f: string3; n: byte; eps: real; var x1: real);

Процедура находит корень уравнения $f(x) = 0$.

RootP(xS: char; x: real; f: string3; n: byte; eps: real; PS: string1; Zn: arrofreal1; var x1: real);

Процедура находит корень уравнения $f(xS, \mathrm{PS}) = 0$; где PS – строка, содержащая идентификаторы используемых параметров; Zn – массив значений параметров.

RootB(xS: char; x: real; BS: string1; f: string3; gS: arrs7; n: byte; eps: real; var x1: real);

Процедура находит корень уравнения $f(xS, BS) = 0$; где BS – строка, содержащая идентификаторы используемых блочных функций (блоков); gS – массив для задания формул блоковых функций.

RootPB (xS: char; x: real; BS: string1; f: string3; gS: arrs7; n: byte; eps: real; PS: string1; Zn: arrofreal1; var x1: real);

Процедура находит корень уравнения$f(xS, BS, PS) = 0$

Параметрическая функция:

Обозначения: tS - идентификатор параметрической переменной; t0 – ее начальное значение; f – массив из формул параметрических функции $f_1(tS), f_2(tS)$; n- порядок разложения; eps – точность вычисления (погрешность по значению функции); результаты: значение параметра t и координаты x.

Rootparamet (tS: char; t0: real; f: arrs7; n: byte; eps: real; var t1,x1: real);

Неявная функция, заданная условием $\boldsymbol{F(x, y) = F(x_0, y_0)}$:

Обозначения: xS, yS – идентификаторы переменных; x0,y0 – их начальные значения; f – формула функции $F(x_s, y_s)$; n- порядок разложения; eps – точность вычисления (погрешность значению $|F(x_1, 0) - F(x_0, 0)|$); x – начальная точка, в окрестности которой находится корень; $x1$ – значение корня.

Rootimpl (xS,yS: char; x0,y0: real; f: string3; n: byte; eps: real; x: real; var x1: real);

2.2 Вычисление экстремума

Явная функция:

Обозначения: xS - идентификатор переменной; x0 – ее начальное значение; f – формула функции $f(xS)$; n- порядок разложения; eps – точность вычисления (погрешность по значению производной данной функции в точке экстремума); $x1, y1$ – координаты точки экстремума.

Et (xS: char; x0: real; f: string3; n: byte; eps: real; var x1: real; var y1: real);

Параметрическая функция:

Обозначения: tS - идентификатор параметрической переменной; t0 – ее начальное значение; f – формулы параметрических функции $f_1(tS), f_2(tS)$; n- порядок разложения; eps – точность (погрешность по значению производной функции f_2' в точке t1); результаты: значение параметра t и координаты точки экстремума $x1, y1$.

Точка экстремума t_1 находится из условия $\frac{\partial f_2}{\partial t} = 0$ с точностью eps. Если при этом $\left|\frac{\partial f_1}{\partial t}\right| < 10^{-6}$, то выдается сообщение о неопределенности результата.

Etparamet (tS: char; t0: real; f: arrs7; n: byte; eps: real; var t1,x1,y1: real);

Неявная функция, заданная условием $\boldsymbol{F}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{F}(\boldsymbol{x_0},\boldsymbol{y_0})$

Обозначения: xS, yS – идентификаторы переменных; x0,y0 – начальные значения координат точки экстремума; f – формула функции $F(x_s, y_s)$; n- порядок разложения; eps – точность вычисления (по значению частной производной $\partial F/\partial x$ в точке экстремума); $x1, y1$ – координаты точки экстремума.

Координаты (x_1, y_1) точки экстремума находятся как решение системы двух уравнений

$$\begin{cases} F(x,y) = 0 \\ \frac{\partial F}{\partial x}dx + \frac{\partial F}{\partial y}dy = 0 \end{cases}$$

С начальным приближением x_0, y_0и с точностью eps по значению координаты x_1. Если в точке экстремума $\left|\frac{\partial F}{\partial y}\right| < 10^{-6}$, то выдается сообщение о неопределенности результата.

Etimpl (xS,yS: char; x0,y0: real; f: string3; n: byte; eps: real; var x1,y1: real);

2.3 Решение системы двух уравнений

Обозначения: xS, yS – идентификаторы переменных; x0,y0 – начальные приближения искомых корней; f – формулы левых частей $F(xS, yS)$; n- порядок разложения; eps – точность (и по значению функций f, и одновременно по значению корней); $x1, y1$ – значения корней.

Rootsys(xS,yS: char; x,y: real; f: arrs7; n: byte; eps: real; var x1,y1: real);

2.4 Вычисление определенного интеграла от явной функции

Обозначения обозначения: xS – идентификатор переменной; a,b – пределы интегрирования; f – формула функции интегранта; n- порядок разложения.

Integral(xS: char; a,b: real; f: string3; n: byte): real;

IntegralP(xS: char; a,b: real; f: string3; n: byte; PS: string1; Zn: arrofreal1;): real;

Где $f = f(xS, PS)$, PS – строка, содержащая идентификаторы используемых параметров; Zn – массив значений параметров.

IntegralB(xS: char; a,b: real; BS: string1; f: string3; gS: arrs7; n: byte;): real;

Где $f = f(xS, BS)$, BS – строка, содержащая идентификаторы используемых блочных функций (блоков); gS – массив для задания формул блоковых функций.

IntegralPB(xS: char; a,b: real; BS: string1; f: string3; gS: arrs7; n: byte; PS: string1; Zn: arrofreal1;): real;

Где $f = f(xS, BS, PS)$,

2.5 Вычисление несобственного интеграла от быстроосциллирующей функции

Обозначения: xS – идентификатор переменной; fo – частота осцилляций (fo>>1); f – формула функции интегранта; n- порядок разложения.

Oscint(xS: char; x0, fo: real; fS: string3; n: byte): real;

2.6 Решение задачи Коши:

Решение задачи Коши для ОДУ первого порядка

Обозначения: xS, yS – идентификаторы переменных; a,b – пределы интегрирования; y0 – начальное значение $y_0 = y(a)$; f – формула правой части ОДУ; n- порядок разложения. Результат – значение $y(b)$.

IntODE (xS,yS: char; a,b: real; y0: real; f: string3; n: byte): real;

Решение задачи Коши для ОДУ высокого порядка $m \leq 5$

Обозначения: xS – идентификатор основной переменной; a,b – пределы интегрирования; LS – список идентификаторов для искомой функции и ее производных до порядка (m-1) (длина списка определяет порядок уравнения); Yn – их начальные значения; f – формула правой части ОДУ; n-порядок разложения; Z – имя выходного массива (значения на правом конце функции и ее производных).

IntHODE (xS: char; a,b: real; LS: string1; Yn: arrofreal1; f: string3; n: byte; var Z: arrofreal1);

Решение задачи Коши для сингулярного ОДУ порядка $m \leq 5$

Обозначения: xS – идентификатор основной переменной; b – правый предел интегрирования b>0 (по умолчанию начальная точка a= 0); LS – список идентификаторов для искомой функции и ее производных до порядка (m-1) (длина списка определяет порядок уравнения); Yn – их начальные значения; f – формула правой части ОДУ; n- порядок разложения; Z – имя выходного массива (значения на правом конце функции и ее производных).

Должно выполняться ограничение $n + m \leq 10$.

IntSing (xS: char; b: real; LS: string1; Yn: arrofreal1; fS: string3; n: byte; var Z: arrofreal1);

Решение задачи Коши для системы дифференциальных уравнений

Обозначения: xS – идентификатор основной переменной; a,b – пределы интегрирования; LS – список идентификаторов искомых функции; Yn – их начальные значения; f – формулы правых частей ОДУ; n- порядок разложения; Z – имя выходного массива (значения $LS(b)$).

Intsys(xS: char; a,b: real; LS: string1; Yn: arrofreal1; f: arrs7; n: byte; var Zn: arrofreal1);

Рассмотрим процедуры получения значений функций (вычисление таблиц) на равномерной сетке (для различных видов функций).

2.7. Вычисление таблиц:

Таблица для явной функции

Обозанчения: xS – идентификатор переменных; f – формула функции; a, b – концы отрезка значений аргумента, на котором строится таблица; m – число интервалов; points1, points2 – значения аргумента и функции соответственно.

Заданный отрезок [a,b] разбивается на Nt равных частей, и в узлах полученной равномерной сетки вычисляются табличные значения $x,\ y = f(x)$.

ExplTab (xS: char; f:string3; a: real; b:real; m: byte; var points1:arrofrealG; var points2: arrofrealG);

Таблица для параметрической функции

Обозначения: tS – идентификатор параметрической переменной; f - формулы функций $x = f_1(tS), y = f_2(tS)$; a, b - концы отрезка значений параметра tS, на котором строится таблица; m – число интервалов; pointsT, points1, points2 – значения параметра, аргумента x и функции y соответственно.

ParametTab (tS: char; f: arrs7; a,b: real; m: byte; var pointsT:arrofrealG; var points1:arrofrealG; var points2: arrofrealG);

Таблица для неявной функции заданной уравнением $\boldsymbol{F(x,y) = F(x_0, y_0)}$

Обозначения: xS, yS – идентификаторы переменных; x0, y0 – их начальные значения; f – формула функции $F(xS, yS)$; m1, m2 – число точек равномерной сетки аргумента xS слева от начального значения (m1) и справа от начального значения (m2); h – шаг сетки; eps – точность; n – порядок разложения; points1, points2 – полученные табличные значения.

Таблица строится в узлах равномерной сетки отрезка $[x_0 - m_1 h,\ x_0 + m_2 h]$ с общим числом точек ($m \coloneqq m_1 + m_2 + 1$), нумеруемых индексом i=0,1,…,m. В каждом узле x_i значение y_i находится процедурой нахождения корня уравнения с параметром как корень (относительно y) уравнения $f_i(y) \coloneqq F(x_i, y) - F(x_0, y_0) = 0$ с параметром x_i, с точностью eps по значению функции f_i.

ImplTab (xS,yS: char; x0,y0: real; f: string3; m1, m2: byte; h,eps: real; n: byte; var points1:arrofrealG; var points2: arrofrealG);

Таблица для обратной функции

Обозначения: xS, yS – идентификаторы переменных; y0 – начальное значение y; g – формула функции $x = g(y)$; m1, m2 – число точек равномерной сетки аргумента xS слева от начального значения $x_0 \coloneqq g(y_0)$ (m1) и справа от начального значения (m2); h – шаг сетки; eps – точность; n – порядок разложения; points1, points2 – полученные табличные значения.

Таблица строится в узлах равномерной сетки отрезка $[x_0 - m_1 h,\ x_0 + m_2 h]$. В каждом узле x_i значение y_i находится процедурой нахождения корня уравнения с параметром как корень уравнения g(y)=x_i с параметром x_i, с точностью eps по значению y.

Invtab(xS,yS: char; y0: real; g: string3; m1,m2: byte; h, eps: real; n: byte; var points1:arrofrealG; var points2: arrofrealG);

Таблица для интегральной кривой для ОДУ первого порядка

Обозначения: xS, yS – идентификаторы переменных; x0, y0 – их начальные значения; b – правый конец табличного отрезка (предполагается, что левый конец $a = x_0$) f – формула правой части ОДУ; m – число интервалов разбиения отрезка [a, b]; n – порядок разложения; points1, points2 – полученные табличные значения.

ODETab(xS,yS: char; x0,y0,b: real; f: string3; m,n: byte; var points1:arrofrealG; var points2: arrofrealG);

Таблица для интегральной кривой для ОДУ высокого порядка ($m \leq 5$)

Обозначения: xS – идентификатор основной переменной; a – ее начальное значение; b – правы конец табличного отрезка; LS – список идентификаторов для искомой функции и ее производных до порядка (m-1) (длина списка определяет порядок уравнения); Yn – их начальные значения; f – формула правой части ОДУ; m – число интервалов разбиения отрезка [a, b]; n- порядок разложения; points1, points2 – полученные табличные значения.

HODETab(xS: char; a,b: real; LS: string1; Yn: arrofreal1; f: string3; m, n: byte; var points1:arrofrealG; var points2: arrofrealG);

Таблица для интегральной кривой для сингулярного ОДУ (порядка $m \leq 5$)

Обозначения: xS – идентификатор основной переменной; b – правый предел интегрирования b>0 (по умолчанию начальная точка $x_0 = 0$); LS – список идентификаторов для искомой функции и ее производных до порядка (m-1) (длина списка определяет порядок уравнения); Yn – их начальные значения; f – формула правой части ОДУ; m – число интервалов разбиения отрезка [a, b]; n- порядок разложения; points1, points2 – полученные табличные значения.

SingTab (xS: char; b: real; LS: string1; Yn: arrofreal1; f: string3; m, n: byte; var points1:arrofrealG; var points2: arrofrealG);