ИНТЕРПОЛЯЦИЯ

Методы и компьютерные технологии их реализации

А. М. Половко
П. Н. Бутусов

bhv

УДК 618.3.06
ББК 32.973
П52

Половко А. М., Бутусов П. Н.

П52 Интерполяция. Методы и компьютерные технологии их реализации. — СПб.: БХВ-Петербург, 2004. — 320 с.: ил.

ISBN 5-94157-493-2

Книга содержит основы теории интерполяции: приведена классификация методов, рассматриваются точные и приближенные, однопараметрические и многопараметрические методы. Методика и компьютерные технологии иллюстрируются на примерах решения задач с помощью универсальных программных средств символьной математики Mathematica, Maple, Derive, Mathcad, MATLAB. Впервые описаны программные средства, позволяющие автоматизировать выбор вида функции интерполяции (CurveExpert, TableCurve, Simple Formula). Материал сопровождается большим количеством примеров и решением задач из различных областей знаний: химии, биологии, экономики, теории и практики надежности, эдукологии, теории массового обслуживания, таксации.

Для преподавателей, студентов и специалистов, занимающихся проблемами моделирования, планирования и статистической обработки эксперимента

УДК 618.3.06
ББК 32.973

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зав. редакцией	*Григорий Добин*
Редактор	*Римма Смоляк*
Компьютерная верстка	*Екатерины Трубниковой*
Корректор	*Евгений Камский*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Рецензенты:

Доктор технических наук, профессор Санкт-Петербургского института государственной противопожарной службы МЧС России Щербаков О. В.
Доктор технических наук, профессор Санкт-Петербургской государственной лесотехнической академии им. С. М. Кирова Гуров С. В.

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 25.03.04.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 25,80.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-493-2

Содержание

Глава 2

Компьютерные технологии решения задач интерполяции в системе Derive 5

2.1. Интерполяция точная в узлах. Универсальный метод

Решение задачи интерполяции универсальным методом сводится к следующей последовательности действий:

1. Выбор вида функции интерполяции.
2. Образование системы уравнений в виде матрицы.
3. Решение системы уравнений.
4. Проверка правильности решения.

Опишем процесс реализации этих действия в системе Derive.

Исходные данные вводятся в виде матрицы размером $n \times 2$, где n — число узлов функции $y = f(x)$, заданной в виде таблицы. Функция $y = f(x)$ представляется графически, аппроксимирующая функция $\varphi(x)$ подбирается визуально путем сравнения графика, представленного в виде точек, с графиками типичных функций.

Вводится (n раз) в память ПК уравнение функции $\varphi(x)$. При этом каждое последующее уравнение образуется из предыдущего путем подстановки в предыдущее уравнение новых переменных x и y. Далее образуется система уравнений в виде матрицы, элементами которой при вводе являются метки $\#K$, присваиваемые программой каждому из уравнений.

Решение системы уравнений осуществляется с помощью функции Solve, при этом Derive выдает ответ в виде численных значений коэффициентов функции интерполяции. Правильность решения задачи проверяется путем табулирования функции $\varphi(x)$ и сравнения с таблицей исходных данных.

Процедуры интерполяции с помощью Derive подробно покажем на примере.

Пример 2.1

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 2.1).

Таблица 2.1

	Значения переменных			
x	1	2	3	4
y	6.2	4.1	1.9	0.6

Необходимо найти $\varphi(x)$ — функцию интерполяции точную в узлах, если $\varphi(x)$ является многочленом n-ой степени: $y = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. В данном случае число узлов $n = 4$, тогда многочлен должен быть степени $n - 1$, т. е. $y = a_0 + a_1x + a_2x^2 + a_3x^3$.

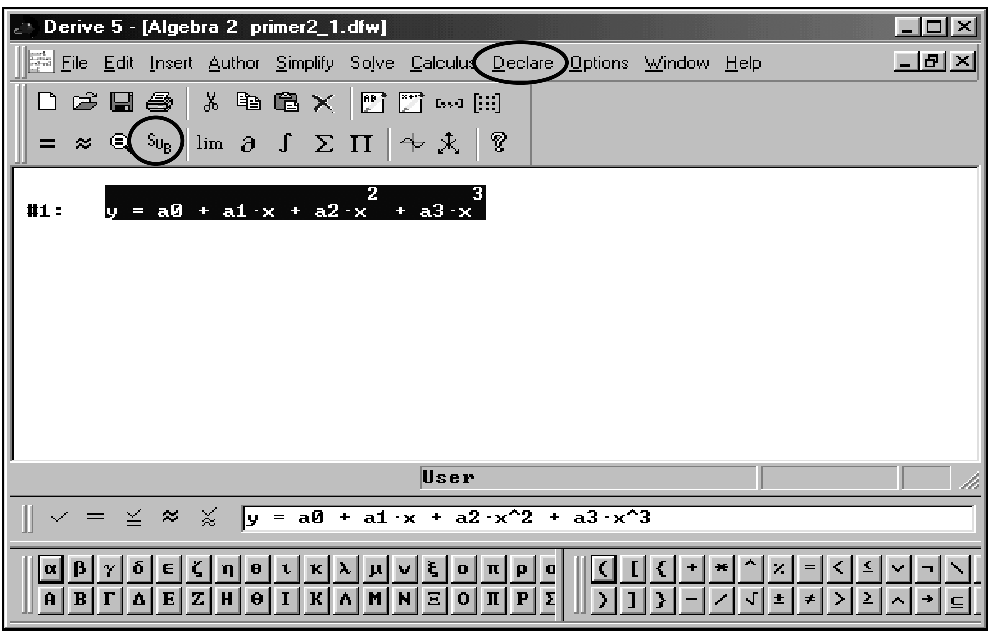


Рис. 2.1

Рассмотрим технологию решения задачи.

1. Настройка Derive на ввод переменных с индексами.

   В главном меню (рис. 2.1) надо нажать на пункт меню **Declare**, откроется диалоговое окно **Input Settings** (рис. 2.2). В группе **Input Mode** надо активизировать переключатель **Word**. Одновременно в группе **Case Sensitivity** активизируется переключатель **Insensitive** и в поле **Radix** устанавливается десятичная система счисления (**Decimal**). Надо нажать на клавишу <Enter> или щелкнуть по кнопке **OK**.

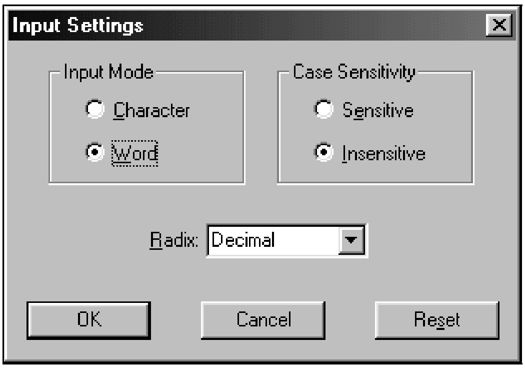


Рис. 2.2

2. Ввод уравнений.

   Активизируется строка пользователя, где надо ввести уравнение:

   `y = a0 + a1 * x + a2 * x^2 +a3 * x^3`, которое записывается в память нажатием клавиши <Enter> или щелчком мыши по кнопке (слева от строки пользователя). На экране отобразится уравнение с меткой #1 (см. рис. 2.1).

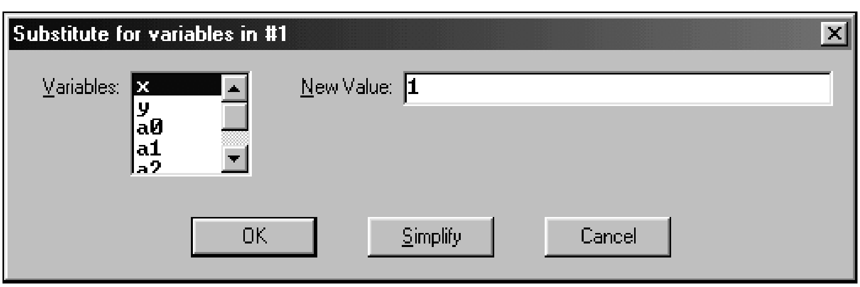


Рис. 2.3

Затем нужно щелкнуть мышью по кнопке **Sub** панели инструментов. На экране откроется диалоговое окно **Substitute for variables in #1** (рис. 2.3).

В поле **Variables** перечислены все переменные уравнения #1. Надо щелкнуть кнопкой мыши по переменной **x** и ввести ее значение в поле **New Value** (в нашем случае 1), затем щелкнуть по переменной **y** и ввести значение y (которое равно 6,2). После щелчка мыши по кнопке **OK** или нажатия клавиши <Enter> на экране отобразится уравнение:

#2: $6.2 = a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 + a_3 \cdot 1^3$.

После щелчка мышью по уравнению #1 производится подстановка значений $x = 2$, $y = 4.1$ аналогично предыдущему. Таким же способом образуются уравнения при $x = 3$, $y = 1.9$ и $x = 4$, $y = 0.6$. На экране с метками #2, #3, #4, #5 увидим следующие четыре уравнения:

#2: $6.2 = a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 + a_3 \cdot 1^3$;

#3: $4.1 = a_0 + a_1 \cdot 2 + a_2 \cdot 2^2 + a_3 \cdot 2^3$;

#4: $1.9 = a_0 + a_1 \cdot 3 + a_2 \cdot 3^2 + a_3 \cdot 3^3$;

#5: $0.6 = a_0 + a_1 \cdot 4 + a_2 \cdot 4^2 + a_3 \cdot 4^3$.

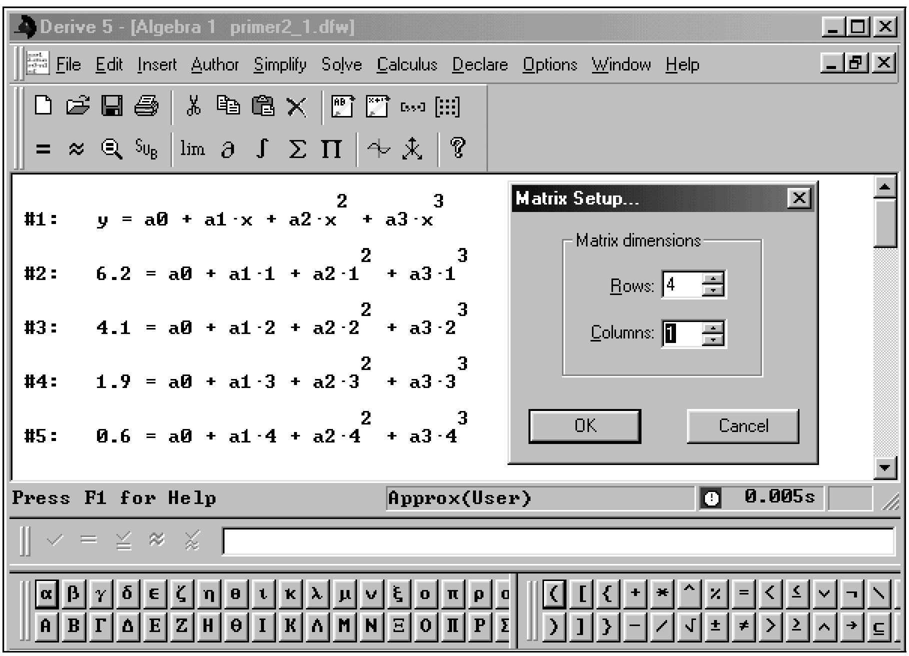


Рис. 2.4

3. Образование системы уравнений.

Выбираем (щелкая мышью) в главном меню пункт **Author | Matrix**. На экране откроется окно **Matrix Setup** (рис. 2.4). В поле **Rows** устанавливаем число уравнений (в данном случае 4), в поле **Colums** устанавливаем число столбцов (в нашем случае равное 1). Щелкаем по кнопке **OK**. На экране активизируется окно **Author 4 x 1 matrix** (рис. 2.5). Последовательно вводим метки (номера) уравнений: #2, #3, #4, #5, перемещая курсор клавишей <Tab> или щелчком левой кнопки мыши. После заполнения всех строк таблицы надо нажать клавишу <Enter> или щелкнуть мышью по кнопке **OK**. На экране появится система уравнений, отмеченная матричными линиями и имеющая метку #6 (рис. 2.6).

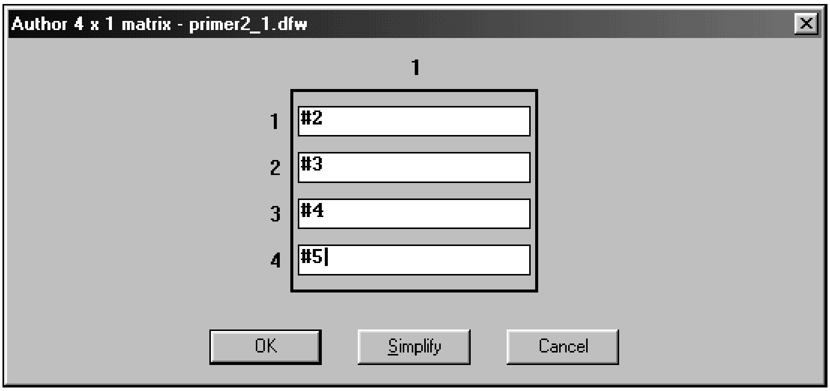


Рис. 2.5

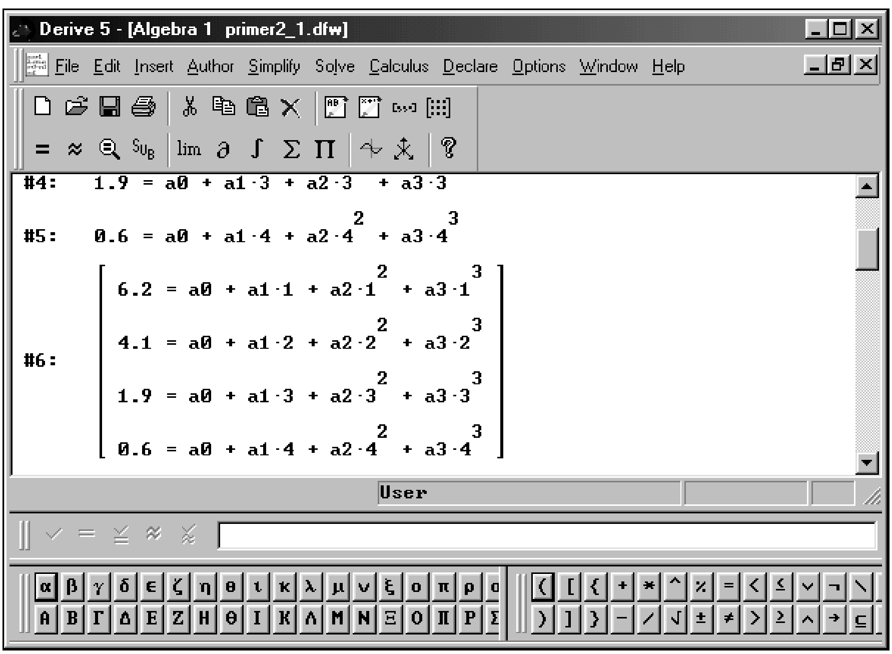


Рис. 2.6

4. Решение системы уравнений:

   - выполняем пункт главного меню **Solve | Expression**. На экране активизируется диалоговое окно **Solve Expression** (рис. 2.7).
   - в поле **Solution Variables** активизируем щелчком мыши все неизвестные a_0, a_1, a_2, a_3; в поле **Solution Method** активизируем переключатель **Either**, а в поле **Solution Domain** — переключатель **Real.**
   - щелкаем мышью по кнопке **Solve**.

   На экране увидим значения коэффициентов:

   $a_0 = 7.2$; $a_1 = -0.116666$; $a_2 = -1.05$; $a_3 = 0.166666$.

   Тогда интерполяционная формула будет иметь вид:

   $(x) = 7.2 - 0.116666x - 1.05x^2 + 0.166666x^3$.

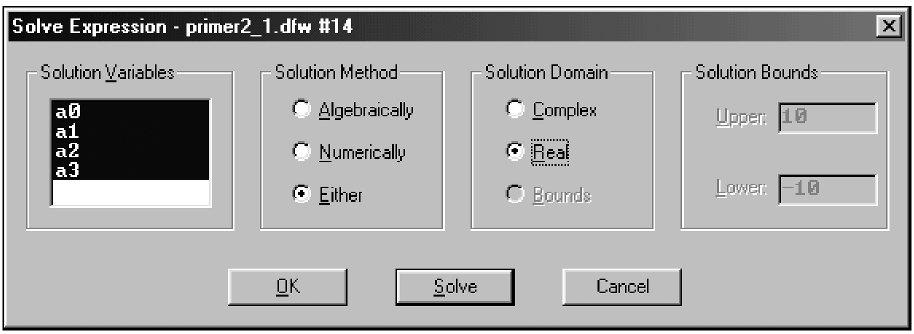


Рис. 2.7

5. Проверка правильности решения задачи интерполяции.

   Выполним табулирование функции (x) и сравним с исходными данными, приведенными в табл. 2.1. Табулирование функции в Derive осуществляется с помощью функции VECTOR([x, (x)], x, $x_н$, $x_к$, h) (где $x_н$, $x_к$ — начальное и конечное значения узлов интерполяции, h — шаг таблицы функции).

   Последовательность процедуры проверки решения в Derive:

   - ввод функции `φ(x)`. На экране появится выражение:

     #1: $7.2 - 0.116666x - 1.05x^2 + 0.166666x^3$.

   - ввод в строке пользователя команды:

     `VECTOR([x, #1], x, 1, 4, 1).`

     На экране отобразится эта команда:

     #2: VECTOR([x, $7.2 - 0.116666x^3 - 1.05x^2 + 0.166666x^3$], x, 1, 4, 1).

- выполнение команды Approximate (кнопка на панели инструментов). На экране (рис. 2.8) увидим ответ в виде следующей матрицы:

$$\#3:\begin{bmatrix} 1 & 6.2 \\ 2 & 4.09999 \\ 3 & 1.89998 \\ 4 & 0.59996 \end{bmatrix}$$

Сравнение результата с исходной таблицей показывает, что решение верно.

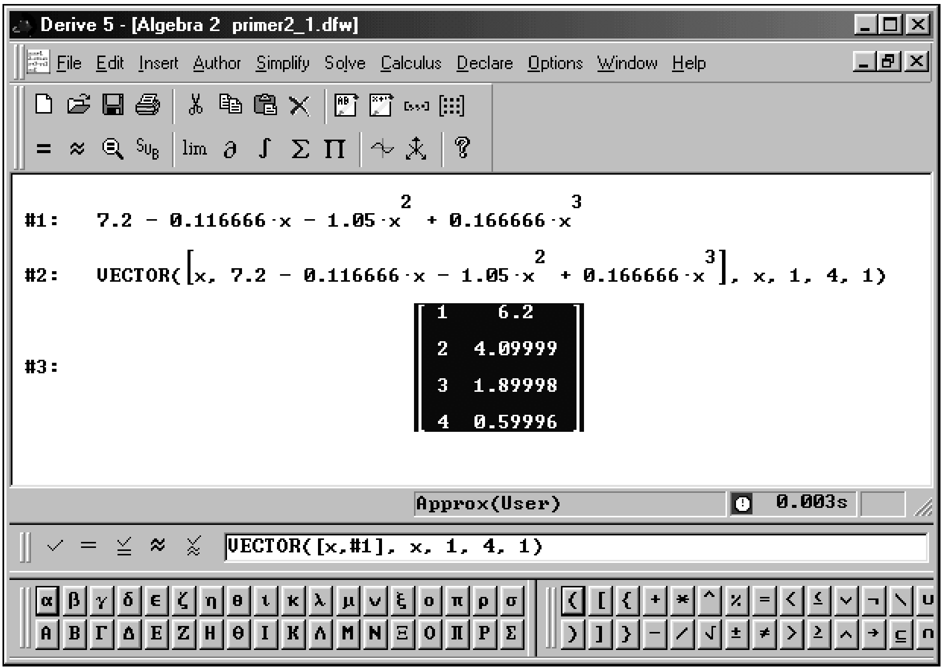


Рис. 2.8

Более рациональным может оказаться матричный способ решения системы уравнений. Пусть A — матрица коэффициентов, X — вектор неизвестных, B — вектор правых частей системы уравнений. В нашем случае:

$$A=\begin{vmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \\ 1 & x_4 & x_4^2 & x_4^3 \end{vmatrix}, \quad X=\begin{vmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{vmatrix}, \quad B=\begin{vmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{vmatrix}.$$

Тогда решение ищется в виде $X = A^{-1} \cdot B$.

В Derive матричный метод реализуется следующим образом.

Для ввода векторов X и B выполняем команду главного меню **Author | Vector**, в результате чего на экране активизируется окно **Vector Setup**. В поле **Elements** устанавливаем размер вектора (в нашем случае 4), щелкаем по кнопке **OK**. На экране появится таблица с четырьмя пустыми строками для будущего вектора. Последовательно вводим элементы вектора B: `6.2, 4.1, 1.9, 0.6` и щелкаем по кнопке **OK**. На экране увидим вектор:

[6.2, 4.1, 1.9, 0.6].

Аналогично образуется вектор неизвестных, который на экране будет иметь вид:

$[a_0, a_1, a_2, a_3]$.

Создаем матрицу коэффициентов A таким образом, как это описано ранее при создании системы уравнений. Размер матрицы 4×4.

Пусть метками векторов и матрицы являются:

- #1 — вектор правых частей системы уравнений;
- #2 — вектор неизвестных;
- #3 — матрица коэффициентов.

В строке пользователя вводим выражение: `#2 = #3^(-1)·#1`.

На экране появится выражение $X = A^{-1} \cdot B$, представленное в матричном виде.

Щелкаем по пункту главного меню **Simplify** (а можно по любой из двух кнопок панели инструментов: по кнопке = или по кнопке ≈ (команда Approximate)). На экране увидим приближенное решение в виде чисел, представленных в естественной форме (рис. 2.9.). (На экране промежуточное выражение с меткой #4 не появляется. Мы его приводим в тексте для большей наглядности процесса решения системы уравнений)

#1: [6.2, 4.1, 1.9, 0.6]

#2: $[a_0, a_1, a_2, a_3]$

$$\#3: \begin{vmatrix} 1 & 1 & 1^2 & 1^3 \\ 1 & 2 & 2^2 & 2^3 \\ 1 & 3 & 3^2 & 3^3 \\ 1 & 4 & 4^2 & 4^3 \end{vmatrix}$$

$$\#4:[a_0, a_1, a_2, a_3] = \begin{vmatrix} 1 & 1 & 1^2 & 1^3 \\ 1 & 2 & 2^2 & 2^3 \\ 1 & 3 & 3^2 & 3^3 \\ 1 & 4 & 4^2 & 4^3 \end{vmatrix}^{-1} \cdot [6.2,\ 4.1,\ 1.9,\ 0.6]$$

$$\#5:\ [a_0 = 7.2,\ a_1 = -0.116666,\ a_2 = -1.05,\ a_3 = 0.166666]$$

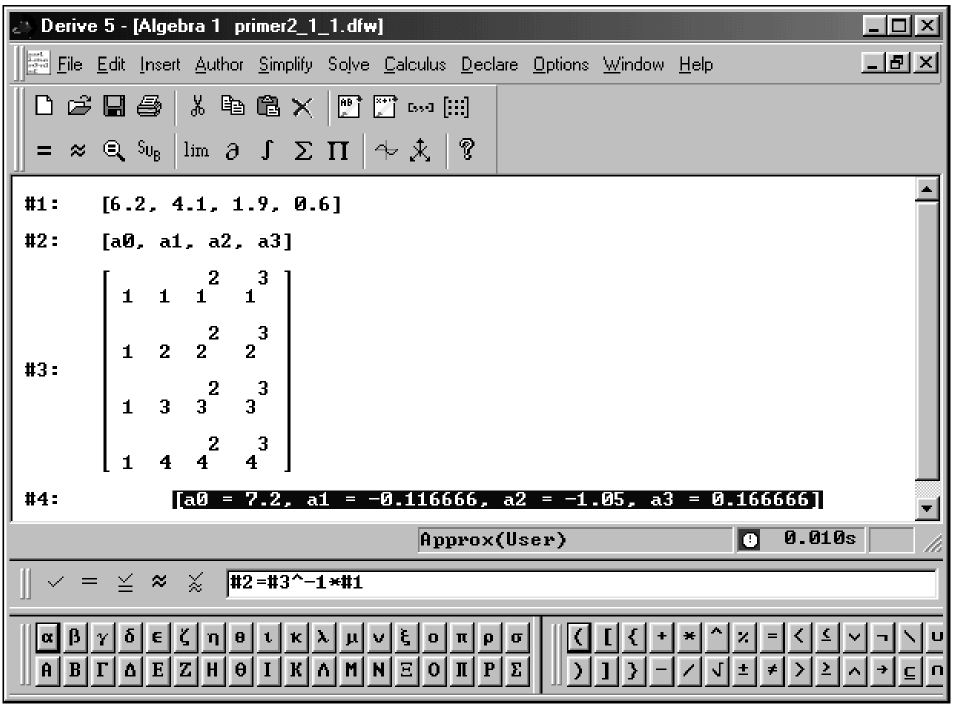


Рис. 2.9

Замечание

При вводе выражения $X = A^{-1} \cdot B$ необходимо соблюдать последовательность символов A и B. Нельзя писать $X = B \cdot A^{-1}$, т. к. решение будет ошибочным.

При большой размерности исходных данных интерполяция точная в узлах применяется для ограниченного диапазона изменения аргумента. При этом интерполяционные формулы могут быть различны для различных участков функции $y = f(x)$. Высокая точность интерполяции для всего диапазона исходных данных дает положительные результаты лишь в случае высокой точности этих данных и при наличии в изучаемом явлении физических закономерностей. При этом решение задачи интерполяции требует вычисления погрешности функции $\varphi(x)$. Рассмотрим типичный для этого случая пример.

Пример 2.2

Пусть зависимость высоты сосны h от ее возраста L, полученная в результате статистической обработки большого количества измерений, приведена в табл. 2.2.

Таблица 2.2

	Значения переменных											
***L*, лет**	20	30	40	50	60	70	80	90	100	110	120	130
***h*, м**	18	22	22.5	28.5	30.5	32	33	34	35	35.5	36	36.3

Необходимо найти математическую модель роста сосны в виде функции $h = \varphi(L)$ и определить высоту сосны для возраста 45 и 67 лет.

Будем предполагать, что существует объективный физический закон, определяющий зависимость параметров сосны от ее возраста, а данные таблицы получены с высокой точностью. Тогда для получения зависимости $h = \varphi(L)$ можно воспользоваться методом интерполяции точной в узлах при ограниченном числе узлов. Из таблицы видно, что функция $h = \varphi(L)$ нелинейна (табличные разности не постоянны). Предположим для начала, что функция $h = \varphi(L)$ является многочленом второй степени: $h = a + bL + cL^2$. Поскольку многочлен содержит три неизвестных — a, b, c, то составим следующую систему из трех уравнений для данных (20, 18), (70, 32), (130, 36.3), взятых из начала, конца и из средней части таблицы:

$18 = a + 20b + 400c$

$32 = a + 70b + 4900c$

$36.3 = a + 130b + 16900c.$

Последовательность решения задачи с помощью Derive описана ранее. В результате решения получим следующие значения коэффициентов:

- $a = 9.748484848$;
- $b = 0.4504545454$;
- $c = -0.001893939393$.

Тогда функция $h = (L)$ будет иметь вид:

$h = 9.75 + 0.45L - 0.0019L^2$.

Так как решение является приближенным и получено лишь по трем значениям исходной функции, то необходимо оценить точность интерполяционной формулы. За критерий близости функций выберем среднеквадратичную погрешность. Абсолютную среднеквадратичную погрешность вычислим по формуле:

$$\varepsilon = \frac{\sqrt{\sum_{i=1}^{n} \Delta_i^2}}{n} \ . \tag{2.1}$$

Здесь:

- ❒ $\Delta_i = (f(L_i) - (L_i))$ — разность между i-тым значением исходной функции $f(L)$ и значением функции (L), полученной в результате интерполяции;
- ❒ n — число точек таблицы (в нашем случае $n = 12$).

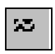
Рассмотрим технологию определения погрешности в Derive.

1. Образование векторов функций $f(L)$ и (L).

   Вектор функции $f(L)$ образуется так, как описано ранее. Вектор функции (L) образуется путем табулирования. Функция табулирования для нашего случая имеет вид:

   VECTOR(#2, x, 20, 130, 10), где:

   - #2 — метка функции (L) на экране;
   - 20, 130 — начальное и конечное значения;
   - 10 — шаг таблицы.

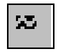
   После ввода функции и выполнения команды Approximate (которая выполняется нажатием кнопки  на панели инструментов), на экране появится вектор значений (L) с меткой #4.

   #4: [17.99, 21.54, 24.71, 27.5, 29.91, 31.94, 33.59, 34.86, 35.75, 36.26, 36.39, 36.14]

2. Образование суммы квадратов разностей векторов $f(L)$ и (L).

   Пусть #1 и #4 — метки векторов $f(L)$ и (L) на экране монитора. Тогда вектор определяется как разность (#1 − #4). После ввода этого выражения на экране обозначится разность векторов $f(L) - (L)$. Щелкаем по кнопке . На экране отобразится вектор разностей векторов с меткой (например, #6).

   #6: [0.01, 0.46, −2.21, 1, 0.59, 0.06, −0.59, −0.86, −0.75, −0.76, 0.39, 0.16]

   После ввода выражения #6^2 на экране увидим квадрат вектора разностей векторов $f(L)$ и (L). После щелчка мышью по кнопке  на экране отобразится значение суммы квадратов разностей векторов в виде числа с меткой (например, #8).

В нашем случае значение $\sum_{i=1}^{12} \Delta_i^2 = 8.853$, представленное в виде: #8: 8.853.

3. Вычисление погрешностей.

   Введем выражение: `sqrt(#8)/12`, затем нажмем клавишу <Enter>. На экране отобразится это выражение. После щелчка мыши по кнопке (Approximate) на экране увидим ответ: значение среднеквадратичной погрешности. В нашем случае $= 0.247949$ (с меткой #9).

   Вычисление максимальной относительной погрешности осуществляется по формуле:

$$\delta = \frac{\varepsilon}{y_{\min}} \cdot 100\ \% \ . \qquad (2.2)$$

   В нашем случае $y_{\min} = h_{\min} = 18$ м.

   Затем введем выражение `δ=#9/18*100`.

   После щелчка мыши по кнопке на экране увидим ответ:

   $= 1.38\ \%$.

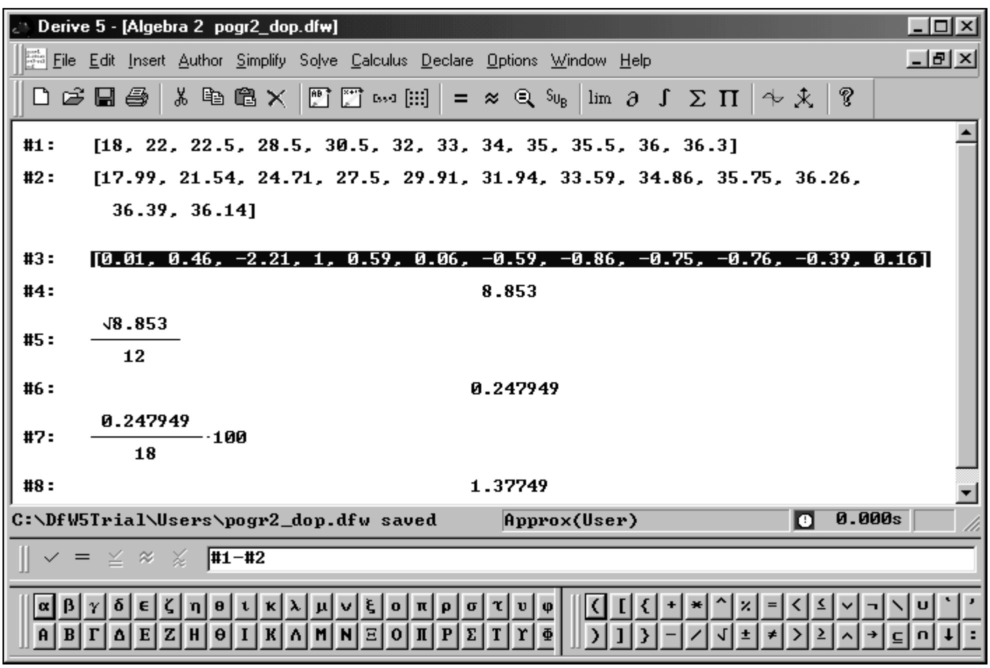


Рис. 2.10

Поскольку максимальная относительная погрешность мала, то полученная в результате интерполяции квадратичная функция может быть принята в качестве модели роста сосны. Процедуры и результаты вычисления погрешности показаны на рис. 2.10.

2.2. Интерполяция по методу Лагранжа. Функция POLY_INTERPOLATE

Технология интерполяции по методу Лагранжа на Derive заключается в следующих действиях:

1. Вызовем утилиту решения задач интерполяции командами: **File | Load | Utility | File | Hermite**. В результате на экране отобразится:

   #1: LOAD (Hermite.mth).

2. Введем матрицу исходных данных по методике, описанной в *разд. 2.1 данной главы*. На экране увидим матрицу с меткой #2.
3. Введем функцию: `LAGRANGE(#2,x)`. На экране увидим функцию с меткой #3 и матрицей исходных данных.
4. Для решения задачи выполним пункт **Simplify** (главное меню) или нажмем кнопку  на панели инструментов. На экране отобразится ответ в виде полинома степени $n - 1$, где n — число узлов функции $f(x)$.

Кроме функции LAGRANGE, в Derive имеется функция POLY_INTERPOLATE решения задач интерполяции многочленами степени n. Функция имеет вид:

POLY_INTERPOLATE(A, x).

Здесь:

- A — матрица исходных данных функции $f(x)$, заданной в виде таблицы;
- x — агрумент функции $f(x)$.

Рассмотрим технологию решения задачи с помощью функции POLY_INTERPOLATE(A, x) в системе Derive:

1. Создадим с помощью пунктов меню **Author | Matrix** матрицу исходных данных способом, описанным в *разд. 2.1 данной главы*. Пусть матрица имеет метку #1.
2. Введем в строке пользователя функцию: `POLY_INTERPOLATE(#1, x)`. На экране появится выражение функции, в котором вместо #1 находится матрица.
3. Выполним пункт главного меню **Simplify**. На экране отобразится ответ.