Компьютерное моделирование при обработке опытных данных

Любому специалисту в своей практической деятельности приходится изучать зависимости между различными параметрами исследуемых объектов, процессов и систем.

Из всех способов задания зависимостей наиболее удобным является аналитический способ задания зависимости в виде функции.

Однако на практике специалист чаще всего получает зависимости между исследуемыми параметрами экспериментально в виде таблицы дискретных значений.

При обработке опытных данных наиболее часто возникают две задачи : задача интерполирования и задача аппроксимации.

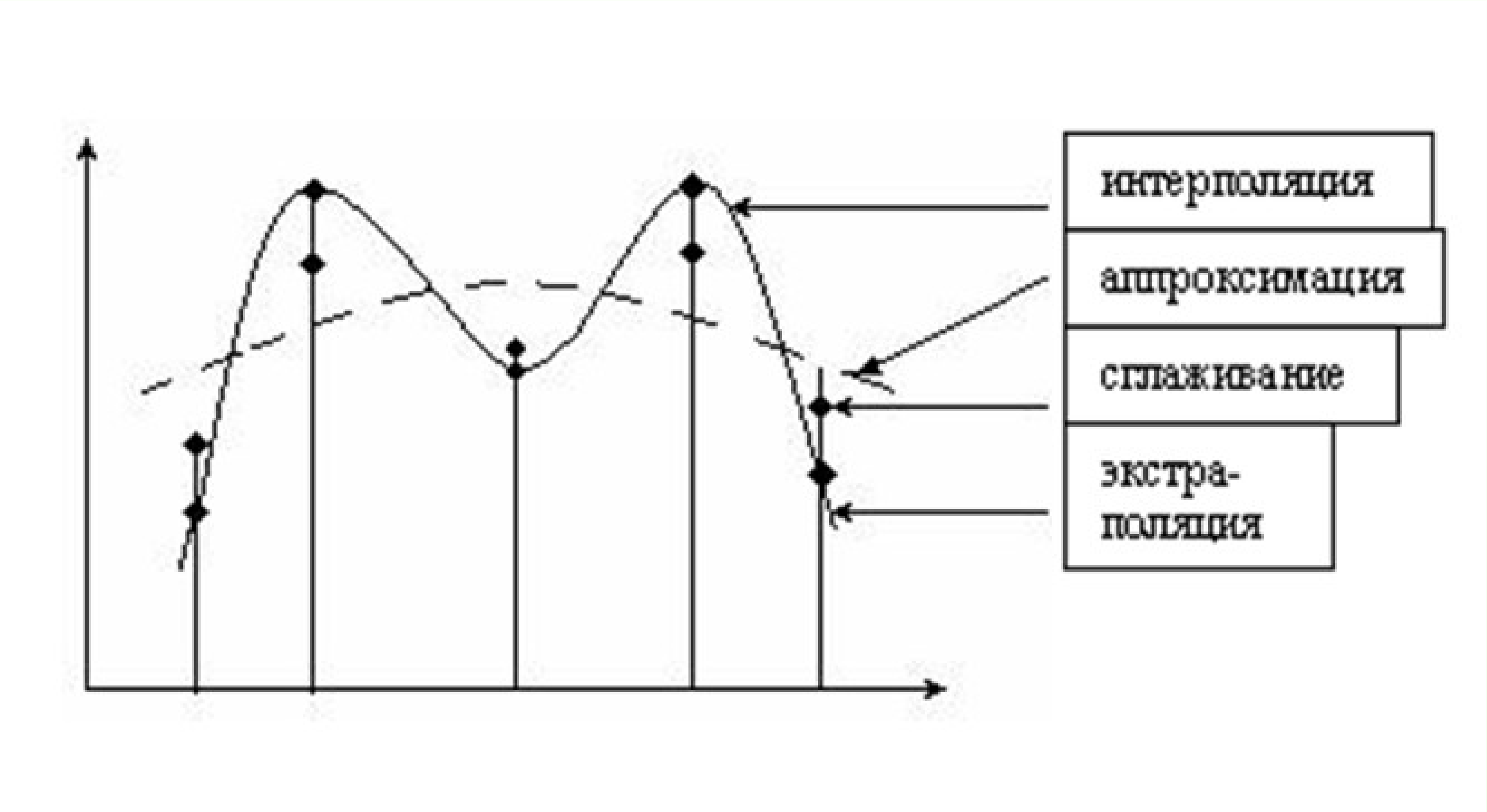


Рис. 1. Обработка экспериментальных данных различными вычислительными методами

- **Интерполяция** – от лат. interpolatio – interpolare - подновлять - нахождение по ряду данных значений функции её промежуточных значений, т.е дополнение значений функции в промежутках между точными её значениями приближёнными .
- **Апроксимация** – от лат. – approximare – приближаться – приближённое выражение каких-либо величин через другие, более известные, более точные величины.

Интерполирование функций

Дана табличная функция, т.е. дана таблица, в которой для некоторых дискретных значений аргумента x_i, расположенных в порядке возрастания, заданы соответствующие значения функции y_i:

i	x	y
0	x_0	y_0
1	x_1	y_1
2	x_2	y_2
...	...	...
i	x_i	y_i
...	...	...
n	x_n	y_n

$$y_i = f(x), i = \overline{0, n}. \qquad (1)$$

Точки с координатами (x_i, y_i) называются **узловыми точками** или **узлами**. Количество узлов в табличной функции равно N=n+1. На графике табличная функция представляется в виде совокупности узловых точек. Длина участка [x_0, x_n] равна (x_n - x_0).

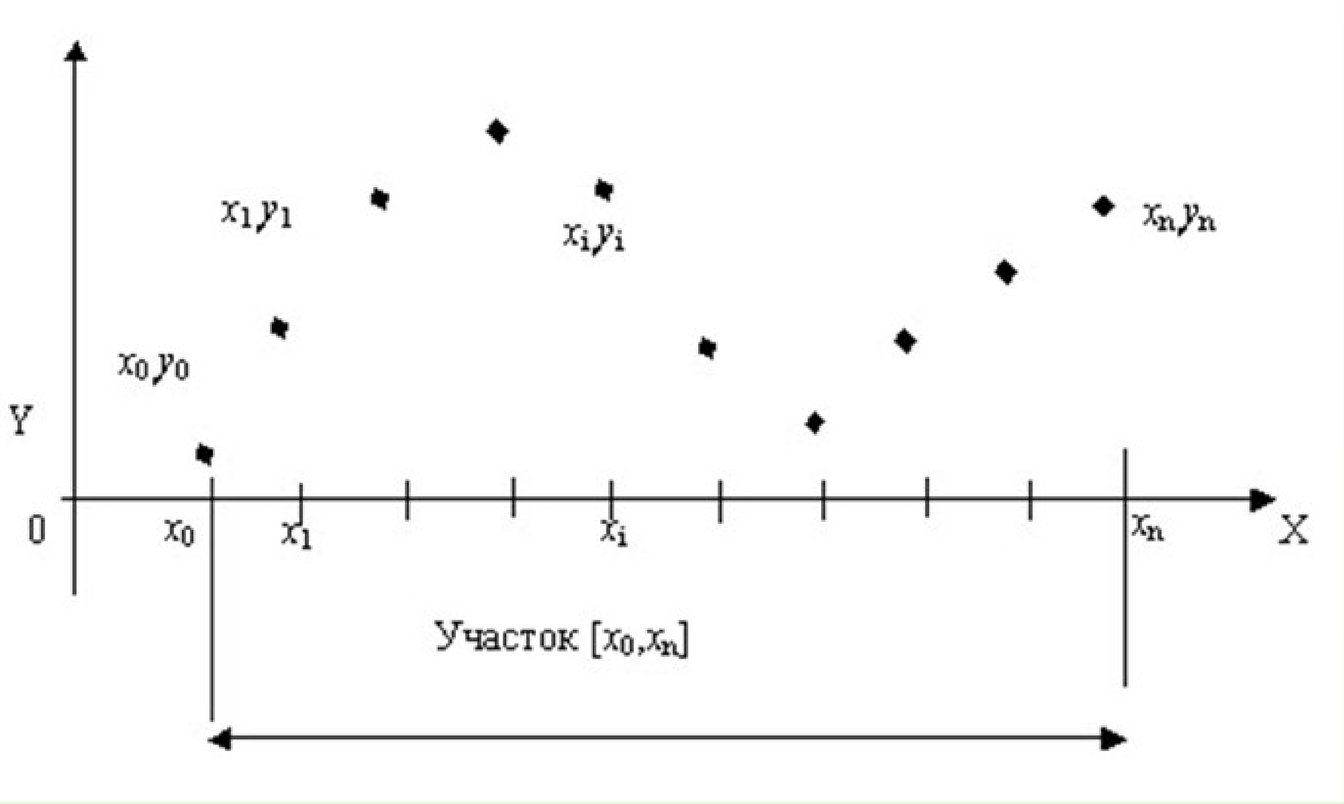


Рис. 2. График зависимости функции y_i от x_i.

Задача **интерполирования функции** (или задача **интерполяции**) состоит в том, чтобы найти значения y_k табличной функции в любой промежуточной точке x_k, расположенной внутри интервала $[x_0, x_n]$, т.е.

$$x_1 < x_k < x_{i+1} \quad \text{и} \quad x_k \in [x_0, x_n].$$

Задача **экстраполирования функции** (или задача **экстраполяции**) состоит в том, чтобы найти значения y_l табличной функции в точке x_l, которая не входит в интервал $[x_0, x_n]$, т.е.

$$x_l < x_0; x_l > x_n.$$

Такую задачу часто называют задачей прогноза. Обе эти задачи решаются при помощи нахождения аналитического выражения некоторой вспомогательной функции F(x), которая приближала бы заданную табличную функцию, т.е. в узловых точках принимала бы значение табличных функций

$$F(x_i) = y_i, i = 0, 1, 2, \ldots, n.$$

Искомую функцию F(x) будем искать в виде **канонического полинома степени n:**

$$P_n(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \ldots + a_{n-1}x^1 + a_nx^0. \qquad (2)$$

Этот многочлен должен пройти через все узловые точки, т.е.

$$P_n(x_i) = y_i. \qquad (3)$$

Поэтому степень многочлена n зависит от количества узловых точек N и равна количеству узловых точек минус один, т.е. n=N-1. Многочлен вида (2), который проходит через все узловые точки табличной функции, называется **интерполяционным многочленом**.

Итак, для решения задачи интерполирования необходимо:

для функции $F(x_i) = y_i, i = 0, 1, 2, \ldots, n$ заданной таблично, построить интерполяционный многочлен степени n, который проходит через все узловые точки таблицы, где n - степень многочлена, равная количеству узловых точек N минус один.

В результате, в любой другой промежуточной точке x_k, расположенной внутри отрезка $[x_0,x_n]$ выполняется приближенное равенство $P_n(x_k) = f(x_k) = y_k$.

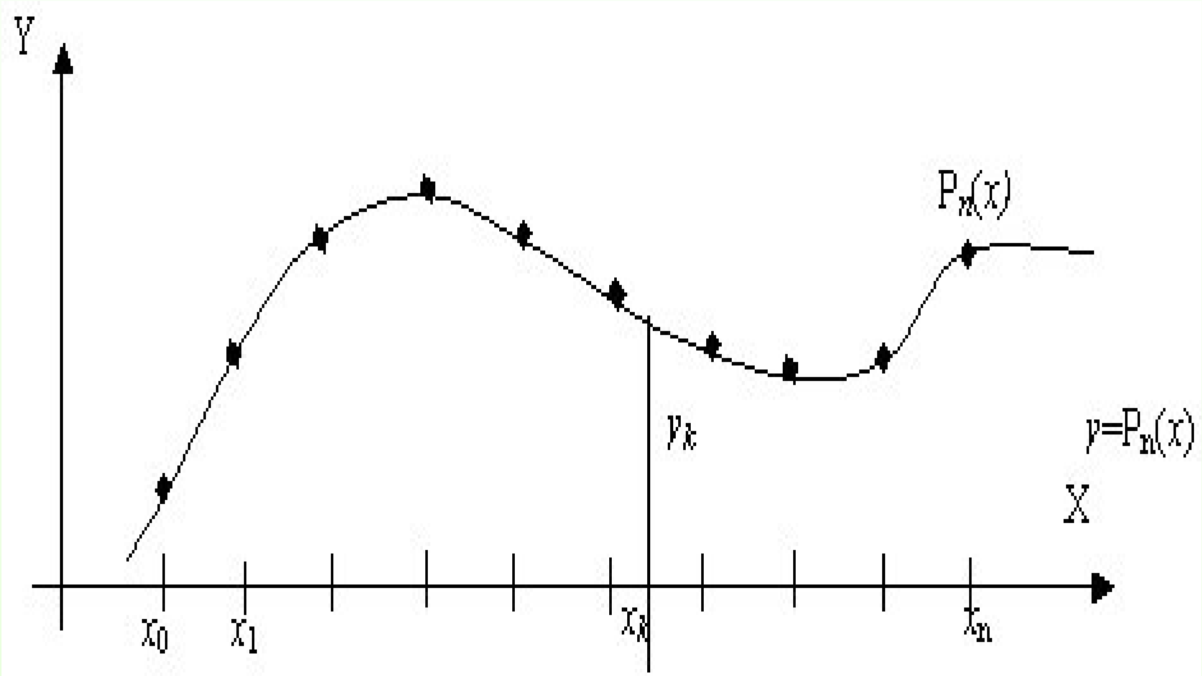


Рис. 3. Графическая иллюстрация интерполирования табличной функции

Для построения интерполяционного многочлена вида (2) необходимо определить его коэффициенты a_0, a_1, :, a_n, т.е. a_i i=0,1,2,:,n. Количество неизвестных коэффициентов равно n+1=N, где n-степень многочлена (2), N-количество узловых точек табличной функции (1). Учитывая уравнение (3) и то, что интерполяционный многочлен должен пройти через каждую узловую точку (x_i, y_i) таблицы, имеем

$$a_0 x_i^n + a_1 x_i^{n-1} + \ldots + a_{n-1} x_i + a_n = y_i, i = 0, 1, \ldots, n. \qquad (4)$$

Подставляя в уравнение (4) каждую узловую точку таблицы получаем систему линейных алгебраических уравнений:

$$\begin{cases} a_0 x_0^n + a_1 x_0^{n-1} + \ldots + a_{n-1} x_0 + a_n = y_0, \\ a_0 x_1^n + a_1 x_1^{n-1} + \ldots + a_{n-1} x_1 + a_n = y_1, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_0 x_n^n + a_1 x_n^{n-1} + \ldots + a_{n-1} x_n + a_n = y_n. \end{cases} \qquad (5)$$

Решая эту систему относительно неизвестных a_0, a_1, a_2, …, a_n , найдём коэффициенты многочлена (2).

Получившаяся СЛАУ относительно свободных параметров a_i имеет решение, если среди узлов x_i нет совпадающих. Ее определитель –

$$W = \begin{vmatrix} x_0^n & \dots & x_0 & 1 \\ x_1^n & \dots & x_1 & 1 \\ \dots & \dots & \dots & 1 \\ x_n^n & \dots & x_n & 1 \end{vmatrix}$$

- называется **определитель Вандермонда**.

Если det (W) $\neq$ 0, то решение системы линейных алгебраических уравнений (5) в матричном виде методом обратной матрицы таково:

$$a = W^{-1} y$$

Полином Лагранжа

Интерполяционный многочлен может быть построен при помощи специальных интерполяционных формул Лагранжа, Ньютона, Стерлинга, Бесселя и др.

Интерполяционный многочлен по формуле Лагранжа имеет вид:

$$
\begin{aligned}
L_n(x) = {} & \frac{(x-x_1)(x-x_2)(x-x_3)\ldots\ldots(x-x_n)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)\ldots(x_0-x_n)} \cdot y_0 + \\
& + \frac{(x-x_0)(x-x_2)(x-x_3)\ldots\ldots(x-x_n)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)\ldots(x_1-x_n)} \cdot y_1 + \\
& + \frac{(x-x_0)(x-x_1)(x-x_3)\ldots\ldots(x-x_n)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)\ldots(x_2-x_n)} \cdot y_2 + \ldots \\
& + \frac{(x-x_0)(x-x_1)(x-x_2)\ldots\ldots(x-x_n-1)}{(x_n-x_0)(x_n-x_1)(x_n-x_2)\ldots(x_n-x_n-1)} \cdot y_n .
\end{aligned}
\qquad (6)
$$

Чтобы убедиться, что выполняется условие $L_n(x_i) = y_i$, подставим значения аргумента: если $x=x_0$, то $L_n(x_0) = y_0$, если $x=x_1$, то $L_n(x_1) = y_1$, … , если $x=x_n$, то $L_n(x_n) = y_n$, т.е. $L_n(x_k) \approx y_k$

Свернем формулу Лагранжа (6). В результате получим

$$L n(x)=\sum_{j=0}^{n} B_j \cdot y_j,$$ где $$B_j=\prod_{i=0}^{n} \frac{x-x_i}{x_j-x_i}, \quad i \neq j$$

При построении алгоритма используют конструкцию из двух включенных циклов:

внутренним циклом накапливаем произведение $B_j=\prod_{i=0}^{n} \frac{x-x_i}{x_j-x_i}, i \neq j$

внешним циклом накапливаем сумму $L=\sum_{j=0}^{n} B_j \cdot y_j$, В качестве примера запишем интерполяционные многочлены Лагранжа первой и второй степени.

При $n = 1$ информация об интерполируемой функции $y = f(x)$ сосредоточена в двух точках: $(x_0; y_0)$ и $(x_1; y_1)$. Многочлен Лагранжа в этом случае составляется с помощью двух базисных многочленов первой степени ($l_0(x)$ и $l_1(x)$) и имеет вид

$$L_1(x) = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1. \qquad (\ .7)$$

При $n = 2$ по трехточечной таблице

$f(x)$:

x	x_0	x_1	x_2
y	y_0	y_1	y_2

можно образовать три базисных многочлена ($l_0(x)$, $l_1(x)$ и $l_2(x)$) и, соответственно, интерполяционный многочлен Лагранжа второй степени

$$L_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} y_1 +$$

$$+ \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} y_2 .$$

Приближенные равенства

$$f(x) \approx L_1(x) \qquad \text{и} \qquad f(x) \approx L_2(x)$$

называют соответственно ***формулами линейной и квадратичной интерполяции***. Геометрически они означают подмену графика функции $y = f(x)$ на некотором отрезке $[a, b]$ оси абсцисс, содержащем точки x_0, x_1 в первом и x_0, x_1, x_2 во втором случаях, соответствующими участками прямой линии и квадратичной параболы, проходящих через заданные точки координатной плоскости.

КОНЕЧНЫЕ РАЗНОСТИ

Зададимся целью придать интерполяционной формуле более простой вид, подобный виду широко используемой в математическом анализе формулы Тейлора. Если в интерполяционном многочлене Лагранжа (6) все слагаемые однотипны и играют одинаковую роль в образовании результата, хотелось бы иметь такое представление интерполяционного многочлена, в котором, как и в многочлене Тейлора, слагаемые располагались бы в порядке убывания их значимости. Такая структура интерполяционного многочлена позволила бы более просто перестраивать его степень, добавляя или отбрасывая удаленные от начала его записи члены.

Поставленной цели будем добиваться сначала для несколько суженной постановки задачи интерполяции. А именно, будем считать, что интерполируемая функция $y = f(x)$ задана своими значениями $y_0, y_1, \ldots, y_n$ на системе ***равноотстоящих узлов*** $x_0, x_1, \ldots, x_n$, т.е. таких, что любой узел x_i этой ***сетки*** можно представить в виде

$$x_i = x_0 + ih,$$

где $i = 0, 1, \ldots, n$, а $h > 0$ — некоторая постоянная величина, называемая ***шагом сетки*** (таблицы).

Прежде чем строить желаемые интерполяционные формулы, рассмотрим **элементы теории конечных разностей**.

Вычитая из каждого последующего члена конечной последовательности из $n+1$ чисел $y_0, y_1, \ldots, y_n$ предыдущий, образуем n ***конечных разностей первого порядка***

$$\Delta y_0 := y_1 - y_0, \quad \Delta y_1 := y_2 - y_1, \quad \ldots \quad, \quad \Delta y_{n-1} := y_n - y_{n-1}$$

или, проще, n ***первых разностей*** данной табличной функции. Из них, в свою очередь, таким же образом можно получить $n-1$ ***конечных разностей второго порядка***, или ***вторых разностей***:

$$\Delta^2 y_0 := \Delta y_1 - \Delta y_0, \quad \Delta^2 y_1 := \Delta y_2 - \Delta y_1, \quad \ldots \quad,$$
$$\Delta^2 y_{n-2} := \Delta y_{n-1} - \Delta y_{n-2}.$$

Этот процесс построения разностей может быть продолжен, и весь он, очевидно, описывается одной рекуррентной формулой, выражающей ***конечную разность k-го порядка*** $\Delta^k y_i$ через разности $(k-1)$-го порядка:

$$\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i, \qquad (8.\quad)$$

где $k = 1, 2, \ldots, n$ и $\Delta^0 y_i := y_i$.

В некоторых случаях требуется знать **выражения конечных разностей непосредственно через значения функции**, лежащей в их основе. Для нескольких первых порядков разностей их можно получить прямой подстановкой:

$$\Delta y_i = y_{i+1} - y_i,$$

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

$$\Delta^3 y_i = \Delta^2 y_{i+1} - \Delta^2 y_i = y_{i+3} - 2y_{i+2} + y_{i+1} - (y_{i+2} - 2y_{i+1} + y_i) =$$

$$= y_{i+3} - 3y_{i+2} + 3y_{i+1} - y_i, \quad \text{и т.д.}$$

Привлекая определение производной, можно обнаружить прямую **связь между конечными разностями и производными**. А именно, если учесть, что

$$\lim_{h\to 0} \frac{y_{i+1} - y_i}{h} = \lim_{h\to 0} \frac{f(x_i + h) - f(x_i)}{h} = f'(x_i),$$

то можно сказать, что при малых h имеет место приближенное равенство

$$\Delta y_i \approx f'(x_i)h,$$

т.е. первые разности характеризуют первую производную функции $f(x)$, по значениям которой они составлены. Пользуясь этим, имеем для вторых разностей:

$$\frac{\Delta^2 y_i}{h^2} = \frac{\Delta y_{i+1} - \Delta y_i}{h^2} = \frac{\dfrac{y_{i+2} - y_{i+1}}{h} - \dfrac{y_{i+1} - y_i}{h}}{h} \approx$$

$$\approx \frac{f'(x_i + h) - f'(x_i)}{h} \approx f''(x_i),$$

т.е. $\Delta^2 y_i \approx f''(x_i)h^2$, и, вообще,

$$\Delta^k y_i \approx f^{(k)}(x_i)h^k. \qquad (9)$$

Таким образом, на конечные разности можно смотреть как на некоторый аналог производных*). Отсюда справедливость многих их свойств, одинаковых со свойствами производных.

Отметим лишь **простейшие свойства конечных разностей**:

1) *конечные разности постоянной равны нулю* (очевидно);

2) *постоянный множитель у функции можно выносить за знак конечной разности.*

Действительно,

$$\Delta(Cy(x)) = Cy(x+h) - Cy(x) = C[y(x+h) - y(x)] = C\Delta y(x)$$

при любых фиксированных x и постоянной C;

3) *конечная разность от суммы двух функций равна сумме их конечных разностей в одной и той же точке.*

Свойство проверяется непосредственно: при любых x

$$\Delta(u(x)+v(x)) = u(x+h)+v(x+h)-(u(x)+v(x)) =$$
$$= u(x+h)-u(x)+v(x+h)-v(x) = \Delta u(x)+\Delta v(x).$$

Свойства 2 и 3 характеризуют операцию взятия конечной разности как линейную операцию.

n-е конечные разности многочлена n-й степени постоянны, а $(n+1)$-е и все последующие равны нулю.

Более важным для понимания сути полиномиального интерполирования является утверждение, обратное сделанному выше выводу. А именно, доказано [58, 123], что *если конечные разности n-го порядка некоторой функции $y = y(x)$ постоянны в любой точке x при различных фиксированных шагах h, то эта функция $y(x)$ есть многочлен степени n.*

Для функции $y = f(x)$, заданной таблицей своих значений $y_0, y_1, \ldots, y_n$ в узлах $x_0, x_1, \ldots, x_n$, где $x_i = x_0 + ih$, конечные разности разных порядков удобно помещать в одну общую таблицу с узлами и значениями функции (последние можно интерпретировать как конечные разности нулевого порядка, см. (8.)). Эту общую таблицу называют ***таблицей конечных разностей***. Заметим, что кроме принятого здесь так называемого диагонального расположения конечных разностей, когда числа в каждом столбце записываются со смещением на полстроки так, как это показано в табл.8.4, часто применяют горизонтальное расположение, где $\Delta y_i, \Delta^2 y_i$ и другие разности с индексом i помещают в одной строке с x_i, y_i.

Таблица 8.4

Диагональная таблица конечных разностей

x_0	y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$	$\Delta^4 y_0$	$\cdots$
x_1	y_1	Δy_1	$\Delta^2 y_1$	$\Delta^3 y_1$	$\Delta^4 y_1$	
x_2	y_2	Δy_2	$\Delta^2 y_2$	$\Delta^3 y_2$	$\vdots$	
x_3	y_3	Δy_3	$\Delta^2 y_3$	$\vdots$		
x_4	y_4	Δy_4	$\vdots$			
x_5	y_5	$\vdots$				
$\vdots$	$\vdots$					

Пример 8.3. Составим таблицу конечных разностей для функции $y = x\ln^2 x$ по ее значениям, вычисленным с тремя знаками после запятой в точках $x_i = 0.4 + 0.2i$, где $i = 0, 1, \ldots, 10$. В соответствии с формой, задаваемой табл.8.4, заполняем табл.8.5 всеми возможными для этого случая конечными разностями. Проанализируем ее.

Учитывая связь между конечными разностями и производными соответствующих порядков (см. (9)), по смене знаков чисел в столбце Δy_i можно судить о наличии минимума функции в окрестности точки $x = 1$, а положительность всех чисел в столбце $\Delta^2 y_i$ говорит о выпуклости вниз графика данной функции на всем рассматриваемом промежутке $[0.4, 2.4]$.

Далее замечаем, что абсолютные величины конечных разностей сначала убывают с увеличением их порядка, а затем начинают увеличиваться. Это т и п и ч н о е поведение конечных разностей при ограниченной точности задания значений сеточной функции.

Природу наблюдаемого в примере 8.3 поведения модулей конечных разностей нетрудно понять. Если шаг достаточно мал, а данная табличная функция — достаточно гладкая, то сначала происходит естественное убывание $\left|\Delta^k y_i\right|$ с увеличением k, в силу упомянутой связи (9). Когда эти величины становятся достаточно малыми, бóльшую роль начинают играть продукты взаимодействия исходных ошибок округления (так называемый ***шум округлений***).

Что происходит с одной отдельно взятой ошибкой величины ε у значения y_i, можно проследить по табл.8.6. Как видим, с ростом порядка разностей она «расползается» по таблице и увеличивается по абсолютной величине.

Таблица 8.5

Конечные разности функции $y = x\ln^2 x$

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$	$\Delta^7 y_i$	$\Delta^8 y_i$	$\Delta^9 y_i$	$\Delta^{10} y_i$
0.4	0.336										
		–0.179									
0.6	0.157		0.062								
		–0.117		0.015							
0.8	0.040		0.077		–0.012						
		–0.040		0.003		0.008					
1.0	0.000		0.080		–0.004		–0.007				
		0.040		–0.001		0.001		0.013			
1.2	0.040		0.079		–0.003		0.006		–0.035		
		0.119		–0.004		0.007		–0.022		0.090	
1.4	0.159		0.075		0.004		–0.016		0.055		–0.207
		0.194		0.000		–0.009		0.033		–0.117	
1.6	0.353		0.075		–0.005		0.017		–0.062		
		0.269		–0.005		0.008		–0.029			
1.8	0.622		0.070		0.003		–0.012				
		0.339		–0.002		–0.004					
2.0	0.961		0.068		–0.001						
		0.407		–0.003							
2.2	1.368		0.065								
		0.472									
2.4	1.840										

Таблица 8.6

Продвижение ошибки по таблице конечных разностей

…	…	…	…	…	…	
x_{i-2}	y_{i-2}		$\Delta^2 y_{i-3}$		$\Delta^4 y_{i-4}+\varepsilon$	
		Δy_{i-2}		$\Delta^3 y_{i-3}+\varepsilon$		
x_{i-1}	y_{i-1}		$\Delta^2 y_{i-2}+\varepsilon$		$\Delta^4 y_{i-3}-4\varepsilon$	
		$\Delta y_{i-1}+\varepsilon$		$\Delta^3 y_{i-2}-3\varepsilon$		…
x_i	$y_i+\varepsilon$		$\Delta^2 y_{i-1}-2\varepsilon$		$\Delta^4 y_{i-2}+6\varepsilon$	
		$\Delta y_i-\varepsilon$		$\Delta^3 y_{i-1}+3\varepsilon$		…
x_{i+1}	y_{i+1}		$\Delta^2 y_i+\varepsilon$		$\Delta^4 y_{i-1}-4\varepsilon$	
		Δy_{i+1}		$\Delta^3 y_i-\varepsilon$		
x_{i+2}	y_{i+2}		$\Delta^2 y_{i+1}$		$\Delta^4 y_i+\varepsilon$	
…	…	…	…	…	…	

Погрешности, имеющиеся у каждого из данных значений функции, с ростом порядка разностей все больше взаимодействуют.

Из сделанных наблюдений напрашивается следующий вывод. Если какой-то столбец в таблице конечных разностей (в ее эксплуатируемой части) состоит из чисел, абсолютные величины которых составляют всего несколько единиц десятичного знака, являющегося последним в записи исходных значений функции, скажем, не превосходят величины 10ε, где ε — абсолютная погрешность исходных данных, то эти конечные разности и разности всех последующих порядков не несут практически никакой информации о функции, и их не следует использовать. Разности же предшествующего столбца называются ***практически постоянными***, и их порядок определяет степень многочлена, которую можно и должно использовать для идеальной в данных условиях полиномиальной интерполяции.

Вспоминая о том, что многочлен k-й степени имеет k-е разности постоянными, а все последующие — нулевыми, приходим к заключению, что *если k-е разности таблицы конечных разностей некоторой функции практически постоянны, то эта функция ведет себя в рассматриваемой области, как многочлен k-й степени*; эту степень и следует применять для интерполирования с наибольшей для данных реалий точностью.

Обратимся к числовой табл.8.5 нашего примера. Видим, что если исключить из рассмотрения верхнюю диагональную строку, то для всей остальной части таблицы третьи разности удовлетворят условию $\left|\Delta^3 y_i\right| \le 10 \cdot 0.0005$ (где 0.0005 — предельная абсолютная погрешность значений y_i). В такой ситуации разности более высоких порядков не следовало вообще вычислять, а разности второго порядка можно считать практически постоянными, т.е. для подсчета любых промежуточных значений данной функции, за исключением, быть может, тех, которые находятся вблизи узла $x_0 = 0.4$, нужно применять квадратичную интерполяцию.

Полином Ньютона

Дана табличная функция: $y_i = f(x), i = \overline{0, n}.$ Необходимо найти значение этой функции в промежуточной точке, например, x=D, причем $D \in [x_0, x_n]$. Для решения задачи строим интерполяционный многочлен. Интерполяционный многочлен по формуле Ньютона имеет вид:

$$\begin{aligned} N_n(x) = f(x_0) + (x - x_0) \cdot f(x_0, x_1) + \\ + (x - x_0) \cdot (x - x_1) \cdot f(x_0, x_1, x_2) + \\ + (x - x_0) \cdot (x - x_1) \cdot (x - x_2) \cdot f(x_0, x_1, x_2, x_3) + \ldots + \\ + (x - x_0) \cdot (x - x_1) \cdot \ldots \cdot (x - x_{n-1}) \cdot f(x_0, x_1, \ldots, x_n), \end{aligned} \qquad (10)$$

где n - степень многочлена,

$f(x_0), f(x_0, x_1), f(x_0, x_1, x_2), f(x_0, x_1, \ldots, x_n)$ - **разделенные разности** 0-го, 1-го, 2-го,:., n-го порядка, соответственно. Значения $f(x_0)$, $f(x_1)$, : , $f(x_n)$, т.е. значения табличной функции в узлах, называются разделенными разностями нулевого порядка (k=0).

Отношение $f(x_0, x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$ называется разделенной разностью первого порядка (k=1) на участке $[x_0, x_1]$ и равно разности разделенных разностей нулевого порядка на концах участка $[x_0, x_1]$, разделенной на длину этого участка.

Для произвольного участка $[x_i, x_{i+1}]$ разделенная разность первого порядка (k=1) равна

$$f(x_i, x_{i+1}) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}.$$

Отношение $f(x_0, x_1, x_2) = \frac{f(x_1, x_2) - f(x_0, x_1)}{x_2 - x_0}$ называется разделенной разностью второго порядка (k=2) на участке $[x_0, x_2]$ и равно разности разделенных разностей первого порядка, разделенной на длину участка $[x_0, x_2]$.

Для произвольного участка $[x_i, x_{i+2}]$ разделенная разность второго порядка (k=2) равна

$$f(x_i, x_{i+1}, x_{i+2}) = \frac{f(x_{i+1}, x_{i+2}) - f(x_i, x_{i+1})}{x_{i+2} - x_i}$$

Таким образом, разделенная разность k-го порядка на участке [x_i, x_{i+k}] может быть определена через разделенные разности (k-1)-го порядка по рекуррентной формуле:

$$f(x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+k}) = \frac{f(x_{i+1}, x_{i+2}, \ldots, x_{i+k}) - f(x_i, x_{i+1}, \ldots, x_{i+k-1})}{x_{i+k} - x_i} \quad (11)$$

где *k=1,…,n* *i= 0,…,n-k* *n* - степень многочлена.

Максимальное значение k равно n. Тогда i =0 и разделенная разность n-го порядка на участке [x_0,x_n] равна

$$f(x_0, x_i, \ldots, x_n) = \frac{f(x_1, x_2, \ldots, x_n) - f(x_0, x_1, \ldots, x_{n-1})}{x_n - x_0}$$

т.е. равна разности разделенных разностей (n-1)-го порядка, поделенной на длину участка [x_0,x_n]. При длине участков в знаменателе →0, разделённые разности → производным того же порядка. При x=x_0 , многочлен (10) равен $L_n(x_0) = f(x_0) = y_0.$

При x=x_1, многочлен (10) равен

$$N_n(x_1) = f(x_0) + (x_1 - x_0) \cdot f(x_0, x_1) = f(x_0) + (x_1 - x_0) \cdot \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f(x_1) = y_1$$

При $x=x_2$, многочлен равен

$$N_n(x_2) = f(x_0) + (x_2 - x_0)\cdot f(x_0, x_1) + (x_2 - x_0)\cdot(x_2 - x_1)\cdot f(x_0, x_1, x_2) =$$
$$= f(x_0) + (x_2 - x_0)\cdot\frac{f(x_1) - f(x_0)}{x_1 - x_0} + (x_2 - x_0)\cdot(x_2 - x_1)\cdot\frac{f(x_1, x_2) - f(x_0, x_1)}{x_2 - x_0} = f(x_2) = y_2$$

<u>Преимущества:</u> интерполяция по Ньютону имеет некоторые преимущества по сравнению интерполяцией по Лагранжу. Каждое слагаемое интерполяционного многочлена Лагранжа зависит от всех значений табличной функции y_i, i=0,1,:n. Поэтому при изменении количества узловых точек N и степени многочлена n (n=N-1) интерполяционный многочлен Лагранжа требуется строить заново. В многочлене Ньютона при изменении количества узловых точек N и степени многочлена n требуется только добавить или отбросить соответствующее число стандартных слагаемых в формуле Ньютона (10). Это удобно на практике и ускоряет процесс вычислений.

Пример построения полинома Ньютона

Дана табличная функция:

i	x_i	y_i
0	2	0,693147
1	3	1,098613
2	4	1,386295
3	5	1,609438

Вычислить разделенные разности 1-го, 2-го, 3-го порядков (n=3) и занести их в диагональную Таблицу. Разделенные разности первого порядка:

$$f(x_0,x_1)=\frac{f(x_1)-f(x_0)}{x_1-x_0}=\frac{1{,}098613-0{,}693147}{3-2}=0{,}405466.$$

$$f(x_1,x_2)=\frac{f(x_2)-f(x_1)}{x_2-x_1}=\frac{1{,}386295-1{,}098613}{4-3}=0{,}287682.$$

$$f(x_2,x_3)=\frac{f(x_3)-f(x_2)}{x_3-x_2}=\frac{1{,}609438-1{,}386295}{5-4}=0{,}223143.$$

Разделенные разности второго порядка:

$$f(x_0,x_1,x_2)=\frac{f(x_1,x_2)-f(x_0,x_1)}{x_2-x_0}=\frac{0{,}287682-0{,}405466}{4-2}=-0{,}058892.$$

$$f(x_1,x_2,x_3)=\frac{f(x_2,x_3)-f(x_0,x_2)}{x_3-x_1}=\frac{0{,}223143-0{,}287682}{5-3}=-0{,}0322695.$$

Разделенная разность третьего порядка:

$$f(x0,x1,x2,x3)=\frac{f(x_1,x_2,x_3)-f(x_0,x_1,x_2)}{x_3-x_0}=\frac{-0{,}0322695-(-0{,}058892)}{5-2}=0{,}00887416$$

Таблица 11.1. Диагональная таблица разделенных разностей

i	x_i	Разделенная разность			
		0-го пор.	1-го пор.	2-го пор.	3-го пор.
0	2	0,693147			
			0,405466		
1	3	1,098613		-0,058892	
			0,287682		0,00887416
2	4	1,386295		-0,0322695	
			0,223143		
3	5	1,60943			

Интерполяционный многочлен Ньютона для заданной табличной функции имеет вид:

$$N_3(x) = f(x_0) + (x - x_0) \cdot f(x_0, x_1) + (x - x_0)(x - x_1) \cdot f(x_0, x_1, x) +$$
$$+ (x - x_0)(x - x_1)(x - x_2) \cdot f(x_0, x_1, x_2, x_3) = 0,693147 + (x - 2) \cdot 0,405466 +$$
$$+ (x - 2)(x - 3) \cdot (-0,058892) + (x - 2)(x - 3)(x - 4) \cdot 0,0887416.$$

Далее полученный интерполяционный многочлен Ньютона можно привести к нормальному виду

$$N_3(x) = a_0 x^3 + a_1 x^2 + a_2 x + a_3$$

и использовать его для решения задач интерполирования или прогноза.

Погрешность полиномиальной интерполяции

Не смотря на множество интерполирующих кривых, которые возможно построить, может быть получен единственный полином n-ой степени для интерполяции заданной табличной функции. В силу единственности интерполяционного многочлена все интерполяционные многочлены, построенные различными способами, есть лишь различные формы его представления. Именно поэтому полиномы, построенные по формулам Лагранжа и Ньютона сводятся в результате преобразований к каноническому полиному той же степени. Поэтому и погрешность расчётов значений, лежащих между узлами, по этим многочленам одинакова и для случая табличной функции с равноотстоящими узлами в точке *x* определяется соотношением :

$$R_n(x) = f(x) - P_n(x) \le \frac{\max\limits_{a \le x \le b} \left| f^{(n+1)}(x) \right|}{\left[(n+1)!\right]} \cdot \left| \prod_{i=0}^{n} (x - x_i) \right| \qquad (12)$$

где $f^{(n+1)}(x)$ – $(n+1)$-ая производная функции $f(x)$,

$$\prod_{i=0}^{n} (x - x_i) = (x - x_0)(x - x_1)\dots(x - x_n)$$

Увеличение числа узлов и, соответственно, степени полинома $P_n(x)$ ведет к увеличению погрешности из-за роста производных $|f^{(n)}(x)|$.

Входящая в состав погрешности величина $\prod_{i=0}^{n}(x-x_i) = w_n(x)$ ведет себя при постоянном шаге так, как показано на рис. 4.

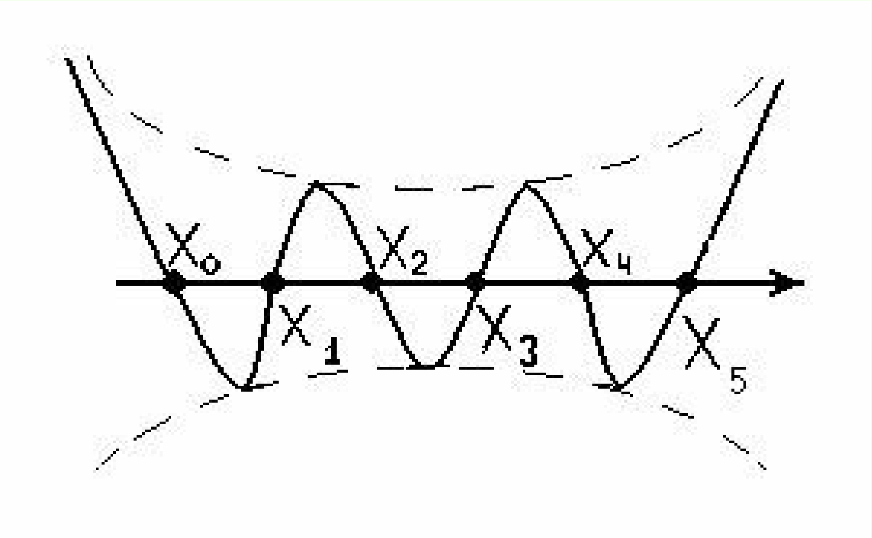


Рис.4. График зависимости $\prod_{i=0}^{n}(x-x_i) = w_n(x)$

Основной недостаток интерполирования с помощью многочленов – неустранимые колебания, которые претерпевает кривая в промежутках между узлами. При этом повышение степени интерполяционного полинома для большинства решаемых уравнений приводит не к уменьшению, а к увеличению погрешности, особенно на концах интервала.

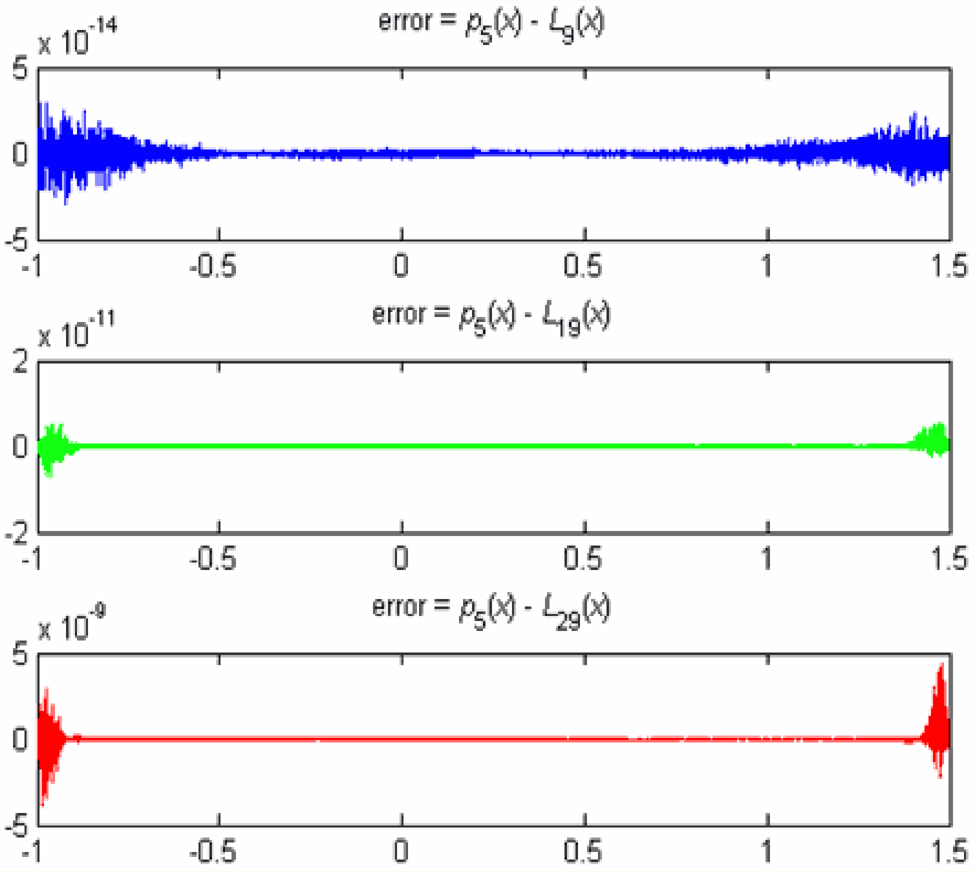
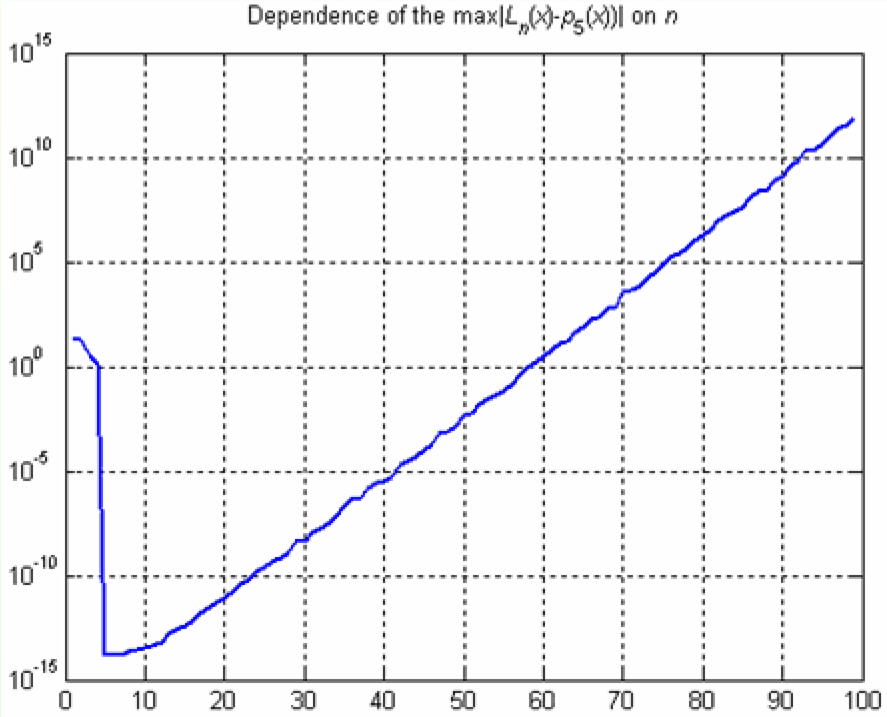


Рис.5. Поведение ошибки для интерполяционных полиномов $L_9(x)$, $L_{19}(x)$, $L_{29}(x)$

Ошибки округлений при вычислении значений интерполяционного полинома приводят к тому, что ошибка интерполирования растет (порядка 10^{-14} для интерполяционного полинома 9-ой степени, порядка 10^{-11} для интерполяционного полинома 19-ой степени и порядка 10^{-9} для интерполяционного полинома 29-ой степени), хотя она должна была бы равняться нулю (при вычислениях в точной арифметике). Заметим, что характер распределения ошибки сохраняется - чем ближе к границам отрезка интерполирования, тем ошибка больше.

Рис. 6. Зависимость ошибки интерполяции от степени интерполяционного полинома



При решении задачи экстраполирования функции с помощью интерполяционного многочлена вычисление значения функции за пределами отрезка $[x_0,x_n]$ обычно производят не далее, чем на один шаг h, равный наименьшей величине $|x_{i+1} - x_i|$, так как за пределами отрезка $[x_0,x_n]$ погрешности, как правило, увеличиваются.

Пример Рассмотрим квадратичную интерполяцию функции $y = \sin x$ на отрезке $\left[0, \frac{\pi}{2}\right]$ по ее трем значениям: $\sin 0 = 0$, $\sin\frac{\pi}{4} = \frac{\sqrt{2}}{2}$, $\sin\frac{\pi}{2} = 1$.

По формуле 6 строим многочлен Лагранжа второй степени

$$L_2(x) = \frac{\left(x - \frac{\pi}{4}\right)\left(x - \frac{\pi}{2}\right)}{\left(0 - \frac{\pi}{4}\right)\left(0 - \frac{\pi}{2}\right)} \cdot 0 + \frac{(x-0)\left(x - \frac{\pi}{2}\right)}{\left(\frac{\pi}{4} - 0\right)\left(\frac{\pi}{4} - \frac{\pi}{2}\right)} \cdot \frac{\sqrt{2}}{2} + \frac{(x-0)\left(x - \frac{\pi}{4}\right)}{\left(\frac{\pi}{2} - 0\right)\left(\frac{\pi}{2} - \frac{\pi}{4}\right)} \cdot 1,$$

или после преобразований

$$L_2(x) = \frac{8}{\pi^2} x\left[\left(1 - \sqrt{2}\right)x + \left(\frac{\sqrt{2}}{2} - \frac{1}{4}\right)\pi\right]. \qquad (13)$$

Остаточный член для этого случая получаем по формуле (14), учитывая, что $n = 2$, $(\sin x)''' = -\cos x$ и $\Pi_3(x) = x\left(x - \frac{\pi}{4}\right)\left(x - \frac{\pi}{2}\right)$.

Имеем:

$$R_2(x) = \sin x - L_2(x) = \frac{-\cos\xi}{3!}x\left(x-\frac{\pi}{4}\right)\left(x-\frac{\pi}{2}\right).$$

Так как точка $\xi \in \left(0, \frac{\pi}{2}\right)$ неизвестна, можно делать лишь оценки $|R_2(x)|$, полагая $M_3 = \max\limits_{x\in\left[0,\frac{\pi}{2}\right]} |-\cos x| = 1$. Найдя максимальное значение $|\Pi_3(x)|$,

реализуемое в двух точках данного отрезка $x_{1,2} = \frac{\pi}{12}\left(3 \pm \sqrt{3}\right)$ и не превосходящее 0.568, по формуле (12) оцениваем сверху величину допустимого отклонения дуги параболы (13) от данной синусоиды на промежутке интерполирования:

$$\max_{x\in\left[0,\frac{\pi}{2}\right]} |\sin x - L_2(x)| \le \frac{M_3}{3!} \max_{x\in\left[0,\frac{\pi}{2}\right]} |\Pi_3(x)| < \frac{0.568}{6} \approx 0.095.$$

Подставим в полученный интерполяционный многочлен (13) контрольную точку $\tilde{x} = \frac{\pi}{6}$. Получим приближенное значение $L_2\left(\frac{\pi}{6}\right) = \frac{4\sqrt{2}-1}{9} \approx 0.517$, отличающееся от значения $\sin\frac{\pi}{6} = 0.5$ на величину ≈ 0.017, меньшую, чем это допускается оценкой по формуле (12):

$$\left|\sin\frac{\pi}{6} - L_2\left(\frac{\pi}{6}\right)\right| \le \frac{M_3}{3!}\left|\Pi_3\left(\frac{\pi}{6}\right)\right| = \frac{1}{6}\left(\frac{\pi}{6}\right)^3 \approx 0.075.$$

Наблюдаем типичную картину: фактическая погрешность меньше оценки погрешности в точке, которая, в свою очередь, меньше максимальной погрешности на отрезке

Поскольку между разделенными разностями и производными соответствующих порядков существует соотношение :

$$f^{(n)}(x) \sim n!\, f_{01\ldots n}\ ,$$

где n – степень производной, а между конечными разностями и производными соответствующих порядков существует соотношение:

$$f^{(n)}(x) \sim \Delta^n(y_i)\ /\ h^n$$

то это используется в численном дифференцировании и при оценке погрешностей интерполяции. Рассмотрим это на следующем примере.

Пример.

Оценить погрешность в точке x*=0,5 многочленной интерполяции, если функция определена из эксперимента.

x_i	$x_0 = -1$	$x_1 = 0$	$x_2 = 1$
y_i	$y_0 = 1/3$	$y_1 = 1$	$y_2 = 3$

Поскольку $n = 2$, то необходимо построить интерполяционные многочлены $L_2(x)$ и $N_2(x)$:

$$L_2(x) = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} +$$

$$+ y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{2}{3}x^2 + \frac{4}{3}x + 1.$$

Для определения конечных разностей, входящих в интерполяционный многочлен Ньютона, удобно пользоваться таблицей конечных разностей, в которую конечная разность k-го порядка в узле x_i определяется как $\Delta^{(k)} y_i = \Delta^{(k-1)} y_{i+1} - \Delta^{(k-1)} y_i$, $k = 1, 2, \ldots; i = \overline{0, n}$.

x_i	y_i	Δy_i	$\Delta^2 y_i$
-1	$\underline{1/3}$		
		$1 - 1/3 = \underline{2/3}$	
0	1		$2 - 2/3 = \underline{4/3}$
		$3 - 1 = 2$	
1	3		

Тогда

$$N_2(x) = y_0 + \frac{\Delta y_0}{1!h}(x - x_0) + \frac{\Delta^2 y_0}{2!h^2}(x - x_0)(x - x_1) =$$
$$= \frac{1}{3} + \frac{2/3}{1 \cdot 1}(x + 1) + \frac{4/3}{2!1^2}(x + 1)(x - 0) = \frac{2}{3}x^2 + \frac{4}{3}x + 1.$$

Верхнюю оценку погрешности можно получить по формуле:

$$|R_n(\widetilde{x})| = |f(\widetilde{x}) - L_n(\widetilde{x})| \leq \frac{\left|\Delta^{(n+1)}(y_0)\right|}{h^{n+1}(n+1)!} \left|(\overline{x} - x_0)(\overline{x} - x_1)\dots(\overline{x} - x_n)\right|,$$

Дополнив таблицу ещё одним узлом интерполяции со значением функции, полученным с помощью, например экстраполяции.

	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$
	-1	$1/3$			
	0	1	$2/3$		
	1	3	2	$4/3$	
Доп. узел	2	7	4	2	$2/3$

$$|y(x^*) - L_3(x^*)| \leqslant \frac{\Delta^3 y_0}{3!h^3} \left|(x^* - x_0)(x^* - x_1)(x^* - x_2)\right| =$$

$$= \frac{2/3}{3!\,1^3} \left|(0{,}5 + 1)(0{,}5 - 0)(0{,}5 - 1)\right| = 0{,}125$$

Итак, поскольку каждое слагаемое интерполяционного многочлена Лагранжа зависит от всех значений табличной функции y_i, i=0,1,…,n, и при изменении количества узловых точек N и степени многочлена n (n=N-1) многочлен Лагранжа требуется строить заново, то рекомендуется его использовать, если необходимо вычислить значение в небольшом количестве точек.

Для расчёта во многих точках рационально использовать полином Ньютона.

При построении же канонического полинома приходится решать СЛАУ (5), что весьма не сложно при использовании современной вычислительной техники, но может оказаться затруднительным при малых значениях определителя Вандермонда.

Можно строить полиномы, не только проходящие через заданные точки, но и имеющие в них заданные касательные (интерполяционный многочлен Эрмита) или заданную кривизну. Количество всех полагаемых условий должно быть

n-1, если n – степень полинома.

Глобальная и кусочно-полиномиальная интерполяция

Пусть функция *f(x)* интерполируется на отрезке [*a,b*]. Метод решения этой задачи с помощью единого многочлена для всего отрезка называют **глобальной полиномиальной интерполяцией**.

В вычислительной практике такой поход применяется редко в силу различных причин. Одна из причин в том, что необходимо задать стратегию выбора узлов при интерполяции функции $f(x)$ многочленами все возрастающей степени n.

Теорема Фабера. Какова бы ни была стратегия выбора узлов интерполяции, найдется непрерывная на отрезке $[a,b]$ функция f, для которой $\max\limits_{[a,b]} |f(x) - P_n(x)| \to \infty$ при $n \to \infty$. Таким образом, теорема Фабера отрицает существование единой для всех непрерывных функций стратегии выбора узлов. Проиллюстрируем сказанное примером.

Предположим, что выбираем равноотстоящие узлы, то есть $x_i = a + ih$, $i = 0,1,\ldots n$, где $h = (b-a)/n$. Покажем, что для функции Рунге такая стратегия является неудачной.

% Интерполяция функции Рунге

```
% Введём функцию Рунге
f = inline('1./(1+25*x.^2)');
```

% Вычислим таблицу значений

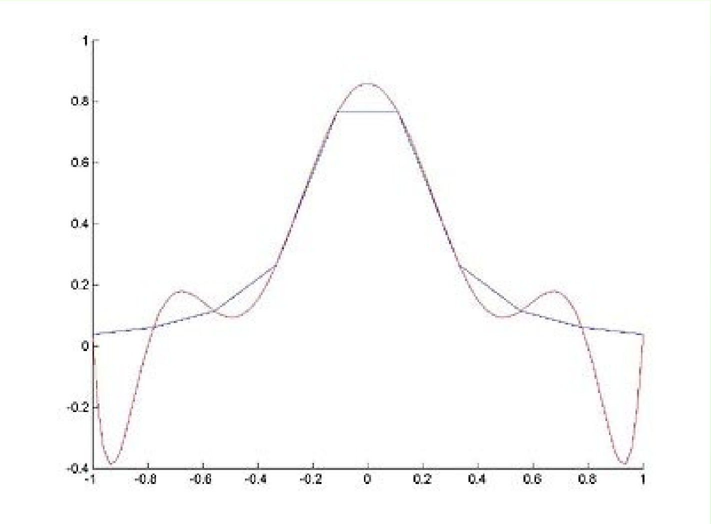
```
x = linspace(-1, 1, 10);
y = f(x);
```

% Проинтерполируем функцию Рунге многочленом 10-й степени

```
p = polyfit(x, y, 10);
xx = linspace(-1, 1, 100);
yy = polyval(p, xx);
axes('NextPlot', 'Add');
```

% Покажем, что глобальная интерполяция плохо работает для функции Рунге

```
plot(x, y);
plot(xx, yy, 'Color', 'r');
```



% С увеличением узлов сетки, ситуация только ухудщается

% Вычислим таблицу значений. 20 узлов сетки

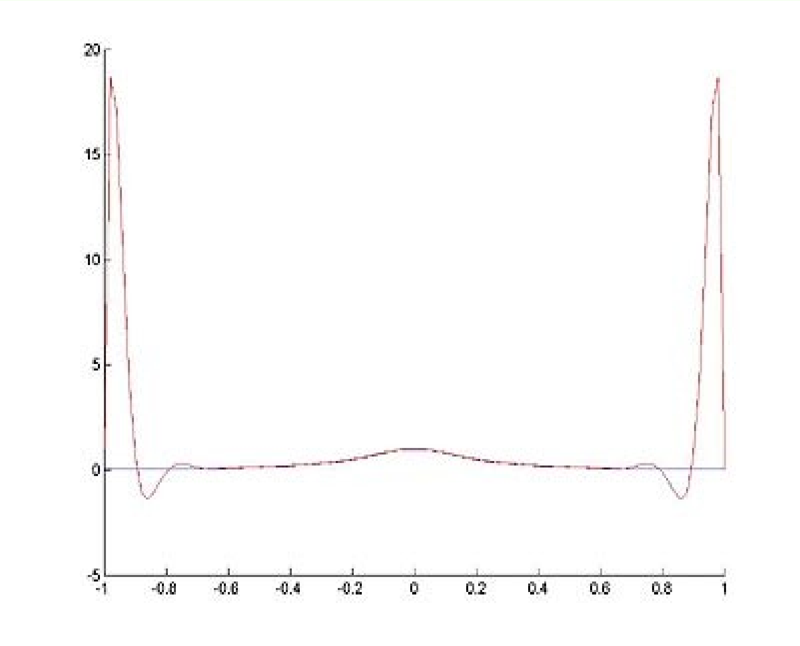
```
x = linspace(-1, 1, 20);
y = f(x);
```

% Проинтерполируем функцию Рунге многочленом 20-й степени

```
p = polyfit(x, y, 20);
xx = linspace(-1, 1, 100);
yy = polyval(p, xx);
figure
axes('NextPlot', 'Add');
```

% Покажем, что глобальная интерполяция плохо работает для функции Рунге

```
plot(x, y);
plot(xx, yy, 'Color', 'r');
```



Начиная с $n \geq 7$ глобальная многочленная интерполяция становится неустойчивой в том смысле, что погрешности возрастают, так как записанный интерполяционный многочлен требует гладкости по производным 7-го и высших порядков.

Поэтому обычную многочленную интерполяцию осуществляют максимум по 3-4 узлам (для 3-х узлов 2-ой степени, 4-х узлов 3-й степени). Интерполяцию по нескольким узлам таблицы называют ***локальной***: линейной по каждым двум узлам с помощью интерполяционных многочленов $L_1(x)$, квадратичной с помощью интерполяционных многочленов $L_2(x)$, и так далее.

На практике чаще используют **кусочно-полиномиальную интерполяцию**: исходный отрезок разбивается на части и на каждом отрезке малой длины исходная функция заменяется многочленом невысокой степени.

При **кусочно-линейной интерполяции** узловые точки соединяются отрезками прямых, то есть через каждые две точки , проводится полином первой степени.

% Кусочно-линейная интерполяция функции Рунге

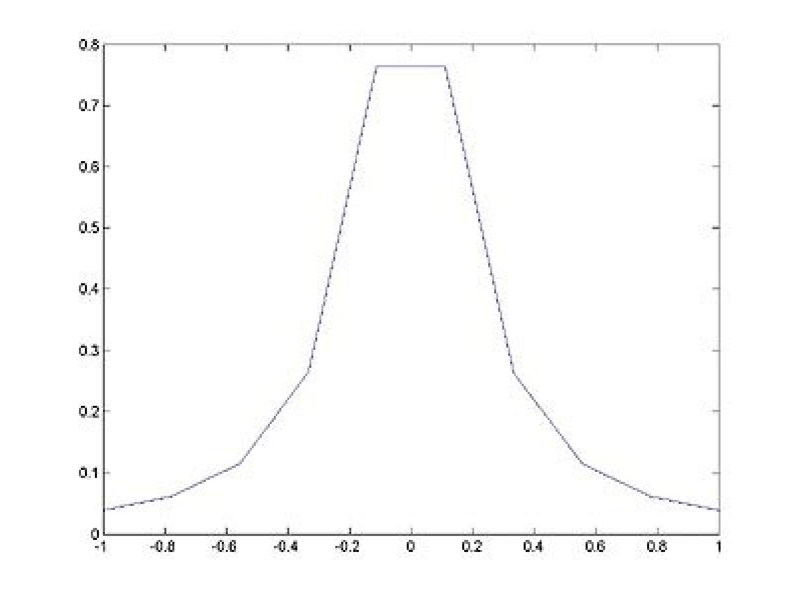
```
% Введём функцию Рунге
f = inline('1./(1+25*x.^2)');
```

% Вычислим таблицу значений

```
x = linspace(-1, 1, 10);
y = f(x);
```

% Начертим график кусочно-линейной аппроксимации

```
plot(x, y);
```



Однако такая локальная интерполяция страдает тем недостатком, что интерполирующая функция в узлах стыковки многочлена имеет непрерывность только нулевого порядка. От этого недостатка свободна сплайн-интерполяция.

Интерполяция сплайнами

Происхождение термина “сплайны” связано с гибкой чертежной линейкой, которой пользовались для рисования гладких кривых, проходящих через заданные точки.

Из теории упругости следует, что получающаяся кривая имеет постоянную кривизну и разрывы возникают лишь в третьей производной. Используя теорию изгиба бруса при малых деформациях, можно показать, что сплайн - это группа кубических многочленов, в местах сопряжения которых первая и вторая производные непрерывны. Такие функции называются **кубическими сплайнами**. Для их построения необходимо задать коэффициенты, которые единственным образом определяют многочлен в промежутке между данными точками.

Например, для некоторых функций (рис.7) необходимо задать все кубические функции $q_1(x)$, $q_2(x)$, …,$q_n(x)$.

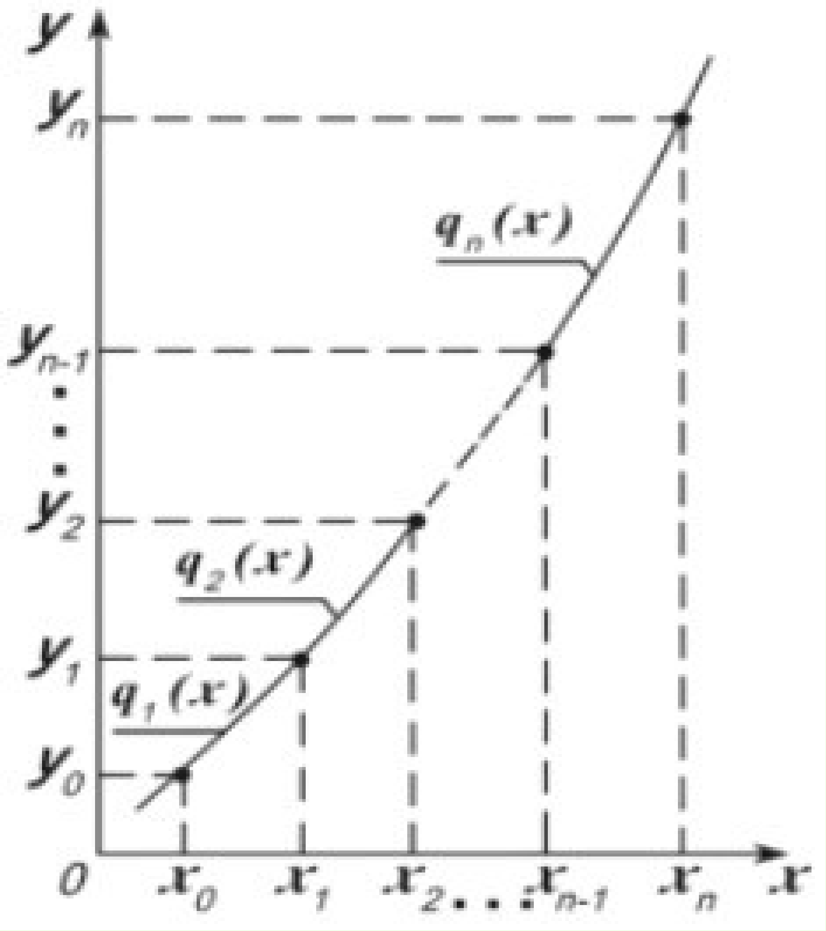


Рис.7. Графическая иллюстрация интерполяции сплайнами

В общем случае эти многочлены имеют вид:

$$q_i(x) = k_{1i} + k_{2i}x + k_{3i}x^2 + k_{4i}x^3, i = \overline{1, n},$$

где k_{ij} - коэффициенты, определяемые описанными ранее условиями, количество которых равно 4n. Для их определения необходимо построить и решить систему алгебраических уравнений порядка 4n. Первые 2n условий требуют, чтобы сплайны соприкасались в заданных точках:

$$q_i(x_i) = y_i, i = \overline{1, n};$$
$$q_{i+1}(x_i) = y_i, i = \overline{0, n-1}.$$

Следующие (2n-2) условий требуют, чтобы в местах соприкосновения сплайнов были равны первые и вторые производные:

$$q'_{i+1}(x_i) = q'_i(x_i), i = \overline{1, n-1};$$
$$q''_{i+1}(x_i) = q''_i(x_i), i = \overline{1, n-1}.$$

Система алгебраических уравнений имеет решение, если число уравнений соответствует числу неизвестных.

Для этого необходимо ввести еще два уравнения для концов интервала. Обычно задают условия свободных концов сплайна :

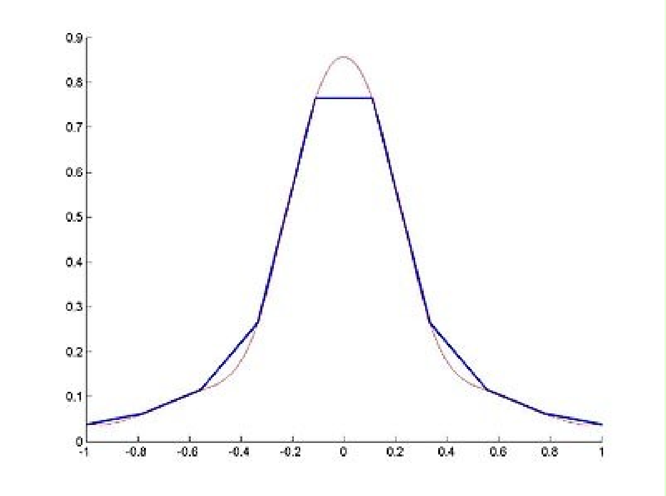
$$q_1''(x_0) = 0, ..., q_n''(x_n) = 0.$$

При построении алгоритма метода первые и вторые производные удобно аппроксимировать разделенными разностями соответствующих порядков.

Полученный таким образом сплайн называется естественным кубическим сплайном. Найдя коэффициенты сплайна, используют эту кусочно-гладкую полиноминальную функцию для представления данных при интерполяции. Разность между степенью сплайна и наивысшим порядком непрерывной на отрезке [*a,b*] производной называют **дефектом сплайна**. Например, кусочно-линейная функция является сплайном первой степени с дефектом, равным единице. Действительно, на отрезке [*a,b*] непрерывна её нулевая производная. В то же время на каждом частичном отрезке она совпадает с некоторым многочленом первой степени. Наиболее широкое распространение получили сплайны 3 степени (кубические сплайны) с дефектом равным 1 или 2.

Сплайновая интерполяция хороша тем, что требует знания в узлах только значений функции, но не ее производных.

```
% Построить интерполяцию сплайнами функции Рунге
% Введём функцию Рунге
f = inline('1./(1+25*x.^2)');
% Вычислим таблицу значений
x = linspace(-1, 1, 10);
y = f(x);
% Вычислим сплайн-интерполяцию
xx = linspace(-1, 1, 100);
yy = spline(x, y, xx);
% Начертим графики
axes('NextPlot', 'Add');
plot(x, y, 'LineWidth', 2);
% Красным на графике - аппроксимация, жирным - исходная функция
plot(xx, yy, 'Color', 'r');
```



Реализация интерполяции в MATLAB

Существует большое число различных способов определения коэффициентов функции интерполяции. Выбор способа зависит от формулировки задачи и метода интерполяции. Система MATLAB позволяет определить коэффициенты функции интерполяции следующими способами:

- использованием встроенных функций системы;
- решением алгебраических линейных или нелинейных уравнений;
- непосредственным вычислением коэффициентов полиномов.

Проверку правильности решения задачи можно осуществить следующими способами:

- табулированием функции аппроксимации и сравнением ее результатов с исходными данными;
- использованием графических образов;
- вычислением погрешности математической модели.

Для реализации интерполяции в MATLAB есть встроенная функция **interp1** следующего вида: **yi = interp1 (x,y,xi, metod),**

где **x** – массив абсцисс экспериментальных точек, **y** – массив ординат экспериментальных точек, **xi** – точки, в которых необходимо вычислить значения с помощью сплайна, **metod** – определяет метод построения сплайна. Этот параметр может принимать следующие значения:

'nearest' – интерполяция по соседним точкам – этот метод построения кусочной функции, при котором значение в любой точке равно значению в ближайшей узловой точке – интерполяция полиномами 0-ой степени;

'linear' – линейная сплайн-интерполяция - интерполяция полиномами 1-ой степени (применяется по умолчанию, если способ интерполирования не задан);

'cubic' – интерполяция кубическим полиномом;

'spline' – интерполяция кубическим сплайном;

'pchip' - интерполяция кубическим эрмитовым сплайном.

Функция **interp1 (x,y, xi, metod)** возвращает значения интерполирующей функции в точках **xi**, но не позволяет получить интерполяционный полином в аналитическом виде.

interp1(x,Y,xi,method,'extrap') - позволяет экстраполировать функцию за пределы интервала.

yi = interp1(x,Y,xi)

yi = interp1(x,Y,xi,method)

yi = interp1(x,Y,xi,method,'extrap‘)

Функция **vander(x)** строит матрицу Вандермонда.

W= vander(x), где матрица W вычисляется так:

W(i,j)= x(i)^(n-j), n = length(x)

Пример. Построение интерполяционного полинома.

x=[1;2;3;4]

y=[2;2;4;5]

W= vander(x)

*a=inv(W)*y*

Функция **spline (x,y,xx)** осуществляет кубическую сплайн-интерполяцию функции, заданной векторами **x** и **y**, и вычисляет значение интерполяционного кубического сплайна в точках **xx**.

yy= spline (x,y,xx)

Интерполяция двумерных данных связана с построением функции двух переменных, приближающей заданные в точках (**xi,yi**) значения **zi**. Для этого следует задать промежуточные узлы командой **meshgrid** и воспользоваться функцией **interp2(x, y, z, xi, yi, 'method')**, которая реализует один из способов интерполирования. В зависимости от значения последнего аргумента:

'nearest' – интерполяция по соседним элементам;

'bilinear' или **'linear'** – билинейная интерполяция (применяется по умолчанию, если способ интерполирования не задан);

'bicubic' или **'cubic'** – интерполяция бикубическими сплайнами;

'spline' - интерполяция кубическими сплайнами.

Для ускорения вычислений в случае равноотстоящих точек с монотонно изменяющимися координатами, используются, соответственно, аргументы **‘*nearest’, ‘*linear’, ‘*cubic’** или **‘*spline’**. **interp2(...,method, extrapval) -** позволяет экстраполировать функцию за пределы интервала.

Сравнение первых трёх вышеперечисленных способов интерполяции может быть осуществлено при помощи программы interp2dem, текст которой приведён. Для избегания утомительного ввода таблицы двумерных данных они генерируются при помощи некоторой функции двух переменных.

```
% генерирование значений табличной функции
[X, Y] = meshgrid(0:0.2:1);
Z = sin(3*pi*X). sin(3*pi*Y). *exp(-X.^2 – Y. ^2);
% визуализация табличной функции
subplot (2, 2, 1)
surf(X, Y, Z)
title (‘табличная функция’)
% создание сетки для промежуточных значений
[Xi, Yi,] = meshgrid 0:0. 02:1)
```

```
% интерполяция по соседним значениям
ZiNear = interp2(X, Y, Z, Xi, Yi, ‘nearest’);
% билинейная интерполяция (можно не указывать ‘bilinear’)
ZiBiLin = interp2(X, Y, Z, Xi, Yi, ‘bilinear’);
% бикубическая интерполяция
ZiBiCub = interp2(X, Y, Z, Xi, Yi, ‘cubic’);
% вывод результатов
subplot (2, 2, 2)
surf (Xi, Yi, ZiNear)
title (‘по соседним значениям (near)’)
subplot (2, 2, 3)
surf (Xi, Yi, ZiBiLin)
title (‘билинейная (bilinear)’)
subplot (2, 2, 4)
surf (Xi, Yi, ZiBiCub)
title (‘бикубическая(bicubic)’)
```

Для аппроксимации трёхмерных данных служит функция **interp3(X,Y,Z,V,XI,YI,ZI)**

VI = interp3(X,Y,Z,V,XI,YI,ZI)

VI = interp3(X,Y,Z,V,XI,YI,ZI,method)

VI = interp3(X,Y,Z,V,XI,YI,ZI,method,extrapval),

для многомерных – **interpn**. Создание многомерных сеток осуществляется функцией **ndgrid**.

interp3(X,Y,Z,V,XI,YI,ZI,method,extrapval) - позволяет экстраполировать функцию за пределы интервала.

VI = interpn(X1,X2,X3,...,V,Y1,Y2,Y3,...)

VI = interpn(...,method)

VI = interpn(...,method,extrapval)

Многомерное приближение производится аналогично двумерному.

Функция **polyfit(x,y,n)** находит коэффициенты полинома, аппроксимирующего функцию, заданную векторами **x** и **y**; **n** – степень полинома: P = polyfit(x,y,n)

где $P(x)= p_1x^n+p_2x^{n-1}+\ldots+p_nx+p_{n+1}$

Если задать **n** значение , равное N-1, где N- размерность векторов **x** и **y**, то будут рассчитаны коэффициенты интерполяционного полинома.

Функция **polyval(p,t)** вычисляет значение полинома, заданного коэффициентами в порядке убывания степени переменной x, в точке **x=t**:

yt = polyval(P, t)

Система MATLAB позволяет обоснованно выбрать степень полинома при полиномиальной интерполяции путем вычисления табличных разностей. Для этой цели служит функция `diff()`, имеющая вид:

```
diff(v, n)
```

где:

- v — вектор функции $y(x)$;
- n — порядок конечных разностей.

Пример . Определить методом табличных разностей степень интерполяционного полинома.

Определим степень полинома, воспользовавшись функцией `diff()`. Процедуры в системе MATLAB имеют вид:

```
>> y=[6.36,6.85,7.34,7.84,8.08,8.32,8.57,8.7,8.82,8.94];
>> diff(y,1)
ans =
     0.49  0.49   0.5   0.24  0.24  0.25   0.13  0.12  0.12
>> diff(y, 2)
ans =
     0   0.01   -0.26   0   0.01   -0.12   -0.01   0
```

Табличные разности второго порядка малы и практически одинаковы, поэтому интерполяционный полином должен быть не выше второй степени.

Задания

1. Функция задана таблично. По таблице конечных разностей определить степень полинома, наиболее точно интерполирующего данную функцию. Построить канонический полином, используя определитель Вандермонда и стандартные операторы MATLAB. Построить график полученной полиномиальной зависимости. Отметить на нём узловые точки. Найти приближённое значение функции с помощью интерполяционных полиномов в точках x_1=0,308 x_2=0,325 x_3=0,312.

X	0,298	0,303	0,310	0,317	0,323	0,330
y	3,25578	3,17639	3,12180	3,04819	2,98755	2,91950

Оценить точность интерполяции в точке x=0,312.

2.

Найти приближенное значение функции при заданном значении аргумента с помощью сплайн-интерполяции в точках $X_1 = 3,75$, $X_2 = 4,75$, $X_3 = 5,25$. Функция задана таблично (табл. 3.10). Изобразить график построенной интерполяционной зависимости, экспериментальные и рассчитанные точки на экране дисплея.

x	2	2.5	3	3.5	4	4.5	5	5.5	6
y	5.197	7.78	11.14	15.09	19.245	23.11	26.25	28.6	30.3