Интерполяция и среднеквадратичное приближение функций

Задача ***приближения*** или ***аппроксимирования функции*** $f(x)$ на некотором числовом множестве X состоит в ее замене функцией определенного вида $Q(x)$ (чаще всего – многочленом), так чтобы значения функций $f(x)$ и $Q(x)$ мало отличались множестве X. Эта задача возникает в следующих случаях:

1) функция $f(x)$ задана таблично, т. е. известны ее значения в конечном числе точек $x_i \in X$, $i=\overline{0,n}$;
2) функция $f(x)$ задана аналитически, но имеет слишком сложный вид.

Далее будем рассматривать только задачи приближения ***многочленами***.

В зависимости от того, что считать мерой отклонения значений двух функций, выделяют различные виды аппроксимации.

Интерполяция

Пусть на отрезке $[a,b]$ даны $n+1$ значений аргумента $x_0=a$, x_1, $x_2,\ldots$, $x_n=b$ (***узлы интерполяции***) и соответствующие значения функции $f(x)$:

$$y_0=f(x_0),\ y_1=f(x_1),\ \ldots,\ y_n=f(x_n).$$

Будем считать, что функция $f(x)$ и многочлен $Q(x)$ *мало отличаются* друг от друга на отрезке $[a,b]$, если их значения совпадают в узлах интерполяции x_i $(i=\overline{0,n})$, т. е.

$$Q(x_0)=y_0,\ Q(x_1)=y_1,\ \ldots,\ Q(x_n)=y_n.$$

Задача построения такого многочлена $Q(x)$ называется ***задачей интерполирования***. При этом $Q(x)$ называется ***интерполяционным многочленом***.

Теорема. Существует один и только один многочлен $Q(x)$ степени не выше, чем n, который решает задачу интерполирования функции $f(x)$ по $n+1$ узлам, т.е. удовлетворяет условиям

$$Q(x_i)=f(x_i)\quad (i=\overline{0,n}).$$

Рассмотрим различные варианты построения интерполяционного многочлена.

1. Интерполяционная формула Лагранжа

Интерполяционный многочлен Лагранжа имеет вид:

$$L_n(x)=\sum_{i=0}^{n} y_i \frac{(x-x_0)(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)(x_i-x_1)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_n)}.$$

Введем обозначение:

$$\Pi_{n+1}(x)=(x-x_0)(x-x_1)\ldots(x-x_n).$$

Тогда интерполяционную формулу Лагранжа можно переписать в виде:

$$L_n(x)=\sum_{i=0}^{n} y_i \frac{\Pi_{n+1}(x)}{(x-x_i)\Pi'_{n+1}(x_i)}.$$

Абсолютная погрешность $|R_n(x)|=|f(x)-L_n(x)|$ интерполяционной формулы Лагранжа:

$$|R_n(x)|\le \frac{M_{n+1}}{(n+1)!}\cdot|\Pi_{n+1}(x)|,$$

где $M_{n+1}=\max\limits_{[a,\,b]}\left|f^{(n+1)}(x)\right|$.

Погрешность $|R_n(x)|$ зависит от двух множителей M_{n+1} и Π_{n+1}. Множитель M_{n+1} полностью определяется свойствами функции $f(x)$ и регулированию не поддается. Второй множитель Π_{n+1} можно изменять, так как он зависит от выбора узлов интерполирования.

Погрешность интерполирования многочленом степени n будет ***минимальной***, если в качестве узлов интерполяции брать точки

$$x_i=\frac{a+b}{2}+\frac{b-a}{2}\cdot t_i,$$

где t_i – корни многочлена Чебышёва $T_{n+1}(t)=\cos\big((n+1)\arccos t\big)$, т. е.

$$t_i=\cos\frac{\pi(2i+1)}{2n+2},\ i=\overline{0,n}.$$

Эти точки называются ***чебышёвскими узлами интерполяции***.

В случае интерполяции по чебышёвским узлам ее максимальная погрешность определяется неравенством

$$\max_{[a,\,b]}|R_n(x)|\le \frac{M_{n+1}}{(n+1)!}\cdot\frac{(b-a)^{n+1}}{2^{2n+1}}.$$

Такая оценка называется ***наилучшей равномерной оценкой погрешности интерполяции*** (ее невозможно улучшить для заданного n).

Если аналитическое выражение для функции $f(x)$ неизвестно, то, строго говоря, оценка погрешности является невозможной.

Недостатком интерполяционной формулы Лагранжа является то, что каждое из слагаемых (они представляет собой многочлены n-й степени) зависит от всех узлов интерполяции x_i. Поэтому при увеличении числа точек x_i и, следовательно, степени многочлена, каждое слагаемое формулы Лагранжа нужно вычислять заново.

2. Интерполяционные формулы Ньютона для равноотстоящих узлов

Предположим, что все узлы интерполяции на отрезке $[a, b]$ равноудалены друг от друга:

$$x_0 = a,\ x_1 = x_0 + h,\ x_2 = x_1 + h = x_0 + 2h, \ldots,\ x_n = x_0 + nh = b,$$

т. е. $x_i = x_0 + ih$, $i = \overline{0, n}$. При этом h называется ***шагом*** интерполяции, а ***узлы*** x_i называются ***равноотстоящими***. Пусть известны значения функции $f(x)$ в этих узлах: $f(x_i) = y_i$, $i = \overline{0, n}$.

Конечной разностью первого порядка функции $f(x)$ в точке x_i называется число, определяемое равенством

$$\Delta y_i = y_{i+1} - y_i,\ \ i = \overline{0, n-1}.$$

Из конечных разностей первого порядка образуют ***конечные разности второго порядка***:

$$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i = y_{i+2} - 2y_{i+1} + y_i,\ \ i = \overline{0, n-2}.$$

Аналогичным образом определяются ***конечные разности третьего и более высоких порядков***:

$$\Delta^3 y_i = \Delta^2 y_{i+1} - \Delta^2 y_i,\ \ i = \overline{0, n-3}, \ldots,$$

$$\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i,\ \ i = \overline{0, n-k},\ k = \overline{1, n}.$$

1) ***Первая интерполяционная формула Ньютона*** имеет вид:

$$P_n(x) = y_0 + t\,\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \ldots + \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_0,$$

где $t = \dfrac{x - x_0}{h}$, $\Delta^k y_0$ – конечная разность порядка k в точке x_0.

Эта формула также называется ***интерполяционной формулой Ньютона для интерполирования вперед***. Ей удобно пользоваться для вычисления значений функции $f(x)$ вблизи точки $x_0 = a$ на отрезке $[a,b]$, а также для ***экстраполирования назад***, т. е. в точках, не принадлежащих отрезку $[a,b]$, но достаточно близких к точке $x_0 = a$.

Абсолютная погрешность $|R_n(x)| = |f(x) - P_n(x)|$ первой интерполяционной формулы Ньютона:

$$|R_n(x)| \approx \frac{|t(t-1)\ldots(t-n+1)(t-n)|}{(n+1)!} \Delta^{n+1} y_0 .$$

2) ***Вторая интерполяционная формула Ньютона*** имеет вид:

$$P_n(x) = y_n + q\,\Delta y_{n-1} + \frac{q(q+1)}{2!}\Delta^2 y_{n-2} + \ldots + \frac{q(q+1)\ldots(q+n-1)}{n!}\Delta^n y_0 ,$$

где $q = \dfrac{x - x_n}{h}$, $\Delta^k y_i$ – конечная разность порядка k в точке x_i.

Эта формула также называется ***интерполяционной формулой Ньютона для интерполирования назад***. Ее используют для интерполирования назад и ***экстраполирования вперед***, т.е. для вычисления значений функции $f(x)$ вблизи точки $x_n = b$, где $|q|$ имеет малые значения.

Абсолютная погрешность второй интерполяционной формулы Ньютона:

$$|R_n(x)| \approx \frac{|q(q+1)\ldots(q+n)|}{(n+1)!} \Delta^{n+1} y_0 .$$

Заметим, что для оценки погрешности при интерполяции многочленами Ньютона n-й степени надо взять дополнительный узел и вычислить слагаемое ($n+1$)-й степени. Его абсолютное значение приближенно равно погрешности соответствующей интерполяционной формулы.

3. Интерполяционная формула Ньютона для неравноотстоящих узлов

Предположим, что узлы интерполяции $x_0 = a,\ x_1,\ x_2, \ldots,\ x_n = b$ – ***неравноотстоящие***, т. е. расстояния между ними, вообще говоря, неравные. Пусть известны значения функции $f(x)$ в этих узлах: $f(x_i) = y_i$, $i = \overline{0,n}$.

Разделенной разностью первого порядка функции $f(x)$ по точкам x_i и x_{i+1} называется выражение

$$f(x_i, x_{i+1}) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = \frac{\Delta y_i}{\Delta x_i}, \quad i = \overline{0, n-1}.$$

Разделенной разностью второго порядка функции $f(x)$ по точкам x_i, x_{i+1} и x_{i+2} называется выражение

$$f(x_i, x_{i+1}, x_{i+2}) = \frac{f(x_{i+1}, x_{i+2}) - f(x_i, x_{i+1})}{x_{i+2} - x_i}, \quad i = \overline{0, n-2}.$$

Разделенные разности более высоких порядков определяются аналогичным образом с помощью рекуррентных формул. ***Разделенная разность порядка k*** имеет вид:

$$f(x_i, x_{i+1}, \ldots, x_{i+k}) = \frac{f(x_{i+1}, x_{i+2}, \ldots, x_{i+k}) - f(x_i, x_{i+1}, \ldots, x_{i+k-1})}{x_{i+k} - x_i}, \quad i = \overline{0, n-k}.$$

Вычисление разделенных разностей можно оформить в виде таблицы.

x	$f(x)$	Разделенные разности			
		1	2	$\ldots$	n
x_0	$f(x_0)$	$f(x_0, x_1)$	$f(x_0, x_1, x_2)$	$\ldots$	$f(x_0, x_1, \ldots, x_n)$
x_1	$f(x_1)$	$f(x_1, x_2)$	$f(x_1, x_2, x_3)$	$\ldots$	–
x_2	$f(x_2)$	$f(x_2, x_3)$	$f(x_2, x_3, x_4)$	$\ldots$	–
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
x_{n-2}	$f(x_{n-2})$	$f(x_{n-2}, x_{n-1})$	$f(x_{n-2}, x_{n-1}, x_n)$	$\ldots$	–
x_{n-1}	$f(x_{n-1})$	$f(x_{n-1}, x_n)$	–	$\ldots$	–
x_n	$f(x_n)$	–	–	$\ldots$	–

Заметим, что добавление нового узла не меняет уже вычисленные разделенные разности. Таблица будет просто дополнена новым столбцом и новыми значениями разделенных разностей внизу ее старых столбцов.

Интерполяционная формула Ньютона для неравноотстоящих узлов имеет вид:

$$P_n(x) = f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) + \ldots$$
$$\ldots + f(x_0, x_1, \ldots, x_n)(x - x_0)(x - x_1)(x - x_2)\ldots(x - x_{n-1}).$$

Ее коэффициентами являются разделенные разности, расположенные в первой строке таблицы.

Данная формула используется для ***интерполирования вперед*** или ***экстраполирования назад***, т. е. для вычисления значений функции $f(x)$ вблизи точки $x_0 = a$.

Погрешность формулы можно оценить, добавив еще один узел x_{n+1}:

$$|R_n(x)| \approx |f(x_0, x_1, \ldots, x_n, x_{n+1})(x - x_0)(x - x_1)\ldots(x - x_{n-1})(x - x_n)|.$$

Равносильный вариант многочлена Ньютона для ***интерполирования назад*** или ***экстраполирования вперед*** можно записать, воспользовавшись нижней (или побочной) диагональю в таблице разделенных разностей:

$$P_n(x) = f(x_n) + f(x_{n-1}, x_n)(x - x_n) + f(x_{n-2}, x_{n-1}, x_n)(x - x_n)(x - x_{n-1}) + \ldots$$
$$\ldots + f(x_0, x_1, \ldots, x_n)(x - x_n)(x - x_{n-1})(x - x_{n-2})\ldots(x - x_1).$$

Указанные формулы Ньютона являются универсальными. Их можно применять при любом расположении узлов интерполирования.

Поскольку интерполяционный многочлен для функции $f(x)$ по одной и той же системе точек x_i $(i = \overline{0, n})$ единственен, то многочлены, построенные по различным формулам Ньютона и Лагранжа совпадают. Преимущество многочленов Ньютона перед многочленом Лагранжа состоит в том, что при добавлении нового узла в многочленах Ньютона появляется только дополнительное слагаемое, а все предыдущие слагаемые не изменяются.

Интерполяция кубическими сплайнами

Если функцию $f(x)$ нужно приблизить функцией $Q(x)$ на отрезке $[a, b]$, длина которого велика, или если функция $f(x)$ не является достаточно гладкой на $[a, b]$, то в качестве $Q(x)$ не имеет смысла использовать многочлены высоких степеней, так как использование большого числа узлов интерполяции на всем отрезке $[a, b]$ не всегда дает удовлетворительную точность приближения и значительно увеличивает количество вычислительных действий. В этом случае используют *кусочно-полиномиальную аппроксимацию* функции $f(x)$. Она состоит в том, что отрезок $[a, b]$ разбивают на частичные отрезки, на каждом из которых функцию $f(x)$ заменяют многочленами одной и той же невысокой степени.

Пусть функция $f(x)$ определена на отрезке $[a, b]$, отрезок $[a, b]$ разбит на n частичных отрезков $[x_{k-1}, x_k]$ $(k = \overline{1, n})$ точками $a = x_0 < x_1 < x_2 < \ldots < x_n = b$.

Сплайном $S_m(x)$ степени m называется определенная на отрезке $[a,b]$ l раз непрерывно-дифференцируемая функция, которая на каждом частичном отрезке $[x_{k-1}, x_k]$ $(k=\overline{1,n})$ является многочленом степени m. При этом точки x_k $(k=\overline{0,n})$ называются ***узлами сплайна***.

Разность d между степенью сплайна m и показателем ее гладкости l (т. е. порядком его наивысшей непрерывной производной) называется ***дефектом сплайна***: $d=m-l$. Для сплайна степени m дефекта d используется обозначение $S_m^d(x)$.

Если сплайн $S_m^d(x)$ строится для функции $f(x)$ по системе точек x_k $(k=\overline{0,n})$ так, чтобы выполнялись условия $S_m^d(x_k)=f(x_k)$, то он называется ***интерполяционным сплайном*** для функции $f(x)$.

Наиболее часто применяется интерполяционный кубический сплайн (т. е. степени 3) дефекта 1. Рассмотрим его подробно.

Пусть известны значения функции $f(x)$ в точках x_k $(a=x_0<x_1<x_2<\ldots<x_n=b)$: $f(x_k)=y_k$, $k=\overline{0,n}$. И пусть узлы сплайна совпадают с узлами интерполяции x_k. Обозначим расстояние между соседними узлами $h_k=x_k-x_{k-1}$, $k=\overline{1,n}$.

Интерполяционным кубическим сплайном дефекта 1 для функции $f(x)$ на отрезке $[a,b]$ называется функция $S_3^1(x)$, которая на каждом частичном отрезке $[x_{k-1}, x_k]$ $(k=\overline{1,n})$ является многочленом третьей степени $g_k(x)$, т. е. имеет вид

$$S_3^1(x)=\left\{g_k(x)=a_k+b_k(x-x_k)+c_k(x-x_k)^2+d_k(x-x_k)^3,\ x\in[x_{k-1},x_k]\right\}_{k=1}^{n},$$

и удовлетворяет условиям:

1) $S_3^1(x)$ дважды непрерывно дифференцируема на $[a,b]$;
2) $S_3^1(x_k)=y_k$, $k=\overline{0,n}$ (условия интерполяции);
3) $S_3^{1\,''}(a)=0$, $S_3^{1\,''}(b)=0$ (краевые условия).

Здесь краевыми условиями являются условия нулевой кривизны на концах отрезка. В общем случае, краевые условия могут задаваться различными способами.

Определенный таким образом интерполяционный сплайн также называют ***естественным*** или ***чертежным сплайном***. Его коэффициенты a_k, b_k, d_k $(k=\overline{1,n})$ находят по формулам:

$$a_k=y_k,$$

$$b_k = \frac{y_k - y_{k-1}}{h_k} + \frac{2}{3} h_k c_k + \frac{1}{3} h_k c_{k-1},$$

$$d_k = \frac{c_k - c_{k-1}}{3h_k},$$

где неизвестные c_k являются решением системы линейных уравнений

$$\begin{cases} h_{k-1} c_{k-2} + 2(h_{k-1} + h_k) c_{k-1} + h_k c_k = 3\left(\dfrac{y_k - y_{k-1}}{h_k} - \dfrac{y_{k-1} - y_{k-2}}{h_{k-1}} \right), \quad k = \overline{2, n-1}, \\ c_0 = 0, \quad c_n = 0. \end{cases}$$

Эта система является линейной системой с трехдиагональной матрицей. Ее решение можно найти, например, методом прогонки. Поскольку матрица системы имеет диагональное преобладание, то метод прогонки сходится, т. е. система имеет единственное решение.

Таким образом, по заданным в узлах интерполяции значениям функции $f(x)$ при заданных краевых условиях $S_3^{1''}(a) = 0$, $S_3^{1''}(b) = 0$ можно построить единственный кубический сплайн дефекта 1, интерполирующий функцию $f(x)$. Если функция $f(x)$ четырежды непрерывно дифференцируема на отрезке $[a, b]$, то для любого фиксированного n существует такая постоянная $C > 0$, что справедлива ***оценка погрешности интерполяции***

$$\left| f(x) - S_3^1(x) \right| \le C \cdot \Delta^4 \quad \text{для любого } x \in [a, b],$$

где $\Delta = \max\limits_{1 \le k \le n} (x_k - x_{k-1})$ – диаметр разбиения отрезка $[a, b]$.

Среднеквадратичное приближение функций.

Пусть значения функции $f(x)$ известны в точках $x_0 = a$, x_1, $x_2, \ldots, x_n = b$ отрезка $[a, b]$: $f(x_i) = y_i$, $i = \overline{0, n}$. Другими словами, функция $f(x)$ задана на дискретном множестве $X = \{x_0, x_1, \ldots, x_n\}$.

Квадратичным отклонением функции $Q(x)$ от функции $f(x)$ на дискретном множестве X называется величина S, равная

$$S = \sum_{i=0}^{n} \left(Q(x_i) - f(x_i) \right)^2 .$$

Построим многочлен m-й степени $Q_m(x) = a_0 + a_1 x + \ldots + a_m x^m$, такой, чтобы его *отклонение* от аппроксимируемой функции $f(x)$ на множестве X было наименьшим.

Если при аппроксимации в качестве меры близости двух функций используется *квадратичное отклонение*, то говорят, что функция $Q_m(x)$ построена с помощью ***метода наименьших квадратов*** или выполнено ***квадратичное*** (***среднеквадратичное***) ***приближение*** функции $f(x)$.

Найдем коэффициенты a_0, a_1, …, a_m многочлена $Q_m(x)$ так, чтобы величина его квадратичного отклонения от значений функции $f(x)$ на множестве $X=\{x_0, x_1, ..., x_n\}$

$$S=\sum_{i=0}^{n}\left(Q_m(x_i)-y_i\right)^2$$

принимала наименьшее значение.

Считая S функцией m переменных a_0, a_1, …, a_m, т. е. $S=S(a_0, a_1, \ldots, a_m)$, запишем необходимые условия ее локального минимума:

$$\frac{\partial S}{\partial a_j}=2\sum_{i=0}^{n}\left(a_0+a_1x_i+\ldots+a_mx_i^m-y_i\right)\cdot x_i^j=0, \quad j=\overline{0,m}.$$

В результате получим систему $m+1$ линейных уравнений относительно $m+1$ неизвестных a_0, a_1, …, a_m:

$$\begin{cases} a_0\sum\limits_{i=0}^{n}1+a_1\sum\limits_{i=0}^{n}x_i+\ldots+a_m\sum\limits_{i=0}^{n}x_i^m=\sum\limits_{i=0}^{n}y_i, \\ a_0\sum\limits_{i=0}^{n}x_i+a_1\sum\limits_{i=0}^{n}x_i^2+\ldots+a_m\sum\limits_{i=0}^{n}x_i^{m+1}=\sum\limits_{i=0}^{n}y_ix_i, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_0\sum\limits_{i=0}^{n}x_i^m+a_1\sum\limits_{i=0}^{n}x_i^{m+1}+\ldots+a_m\sum\limits_{i=0}^{n}x_i^{2m}=\sum\limits_{i=0}^{n}y_ix_i^m. \end{cases}$$

Можно доказать, что если среди точек x_0, x_1, $x_2, \ldots$, x_n нет совпадающих и $m \le n$, то определитель этой системы отличен от нуля и, следовательно, система имеет единственное решение. Многочлен $Q_m(x)$ с такими коэффициентами a_0, a_1, …, a_m будет иметь наименьшее квадратичное отклонение S от функции $f(x)$ на множестве X. Он также называется многочленом (полиномом) *наилучшего среднеквадратичного* (или *квадратичного*) *приближения*.

Если $m=n$, то аппроксимирующий многочлен $Q_m(x)$ совпадает с интерполяционным многочленом для узлов x_0, x_1, $x_2, \ldots$, x_n, при этом квадратичное отклонение $S=0$.

Если $m > n$, то задачу сводят к случаю $m = n$, так как не имеет смысла строить многочлен высокой степени *m*.

Основные функции пакета Mathematica, используемые для приближения функций.

InterpolatingPolynomial[*data, x*] – строит интерполяционный многочлен по формуле Ньютона от переменной *x* для функции, заданной таблицей значений *data*. Данные *data* должны иметь вид $\{\{x_1, y_1\}, \{x_2, y_2\}, \ldots, \{x_n, y_n\}\}$ или $\{y_1, y_2, \ldots, y_n\}$, во втором случае считается, что *x* последовательно принимает значения 1, 2, …, *n*. Для упрощения результата рекомендуется применять функцию ***Expand***.

Interpolation[*data*] – строит интерполирующую функцию для функции $y = f(x)$ по списку ее значений *data*. Данные *data* должны иметь вид $\{\{x_1, y_1\}, \{x_2, y_2\}, \ldots, \{x_n, y_n\}\}$ или $\{y_1, y_2, \ldots, y_n\}$, во втором случае считается, что *x* последовательно принимает значения 1, 2, …, *n*. Поддерживает опцию ***Method***, для которой возможны варианты ***"Spline"*** (сплайн-интерполяция) и **"Hermite"** (интерполяция по Эрмиту), например, ***Interpolation***[*data*, ***Method->"Spline"***]. Результат возвращает в виде объекта ***InterpolatingFunction***, который в системе **Mathematica** может быть использован как любая другая обычная функция.

SplineFit[*data, type*] – выполняет интерполяцию функции $f(x)$ сплайном вида *type* по списку ее значений *data*. Для того чтобы функция ***SplineFit*** была доступна, предварительно необходимо загрузить пакет сплайн-интерполяции с помощью команды ***Needs***[**"*Splines`*"**]. Данные *data* должны иметь вид $\{\{x_1, y_1\}, \{x_2, y_2\}, \ldots, \{x_n, y_n\}\}$. Возможные типы сплайна – ***Cubic***, ***Bezier***, ***CompositeBezier***. Например, ***SplineFit***[*data*,***Cubic***] – интерполяционный кубический сплайн. Результат возвращает в виде объекта ***SplineFunction***[*type, domain, interval*], который дает параметрическое представление интерполяционной кривой в виде $\{x[t], y[t]\}$. Параметр принимает значения из области *domain*. Если дать определенное значение параметра в качестве аргумента этого объекта, то он возвращает координаты соответствующей точки кривой.

Fit[*data*, $\{f_1, \ldots f_m\}$, *x*] – строит с помощью метода наименьших квадратов (МНК) аппроксимирующую функцию для функции $f(x)$ по списку ее значений *data* в виде линейной комбинации функций f_1, …, f_m относительно переменной *x*. Данные *data* должны иметь вид $\{\{x_1, y_1\}, \{x_2, y_2\}, \ldots, \{x_n, y_n\}\}$ или $\{y_1, y_2, \ldots, y_n\}$, во втором случае считается, что x последовательно принимает значения 1, 2, …, n. Для аппроксимации многочленом вида $Q_m(x) = a_0 + a_1 x + \ldots + a_m x^m$ в качестве функций f_1, f_2, …, f_m следует указать 1, x, …, x^m.

***ChebyshevT*[*n*,*t*]** – многочлен Чебышева первого рода $T_n(t)$ *n*-й степени относительно переменной *t*.

***Piecewise*[{**$\{f_1, cond_1\}, \{f_2, cond_2\}, \ldots, \{f_n, cond_n\}$**}]** – представляет кусочно-заданную функцию, равную выражениям f_1, f_2, …, f_n в областях, определяемых соответствующими условиями $cond_1$, $cond_2$,..., $cond_n$.

***FindMaximum*[**$\{f[x], a \le x \le b\}$, x] – находит локальный максимум функции $f(x)$ на отрезке $[a, b]$.

***ParametricPlot*[**$\{x[t], y[t]\}, \{t, t_{\min}, t_{\max}\}$**]** – строит параметрически заданную кривую $x = x(t)$, $y = y(t)$ при t, принимающем значения от $t_{\min}$ до $t_{\max}$.

Примеры интерполяции и аппроксимации функций средствами пакета Mathematica.

Пример 1. Используя значения функции

$$f(x) = \frac{5x^2 - 4x\,\mathrm{arctg}(3x+2) + \ln 2}{2x^2 + 7}$$

в равноотстоящих узлах x_i отрезка $[-2, 6]$ при разбиении его на 6 частей, выполнить следующие действия:

а) создать таблицу конечных разностей функции $f(x)$ по точкам $(x_i, f(x_i))$;

б) построить первый интерполяционный многочлен Ньютона для $f(x)$, проиллюстрировать графически;

в) интерполировать функцию $f(x)$ с помощью встроенных функций **InterpolatingPolynomial** и **Interpolation** пакета **Mathematica**, проиллюстрировать графически;

г) найти значения функции $f(x)$ и построенных интерполяционных функций в точке $x = 2{,}28$.

Δ Введем функцию $f(x)$, границы отрезка a и b, количество частей разбиения отрезка n, вычислим шаг h:

```
In[1]:= f[x_] = (5 x^2 - 4 x ArcTan[3 x + 2] + Log[2]) / (2 x^2 + 7);

In[2]:= a = -2; b = 6; n = 6; h = (b - a)/n;
```

Составим таблицу значений функции $f(x)$ в равноотстоящих узлах:

```
In[3]:= data = N[Table[{a + i h, f[a + i h]}, {i, 0, n}]]
              таблица значений

Out[3]= {{-2., 0.67244}, {-0.666667, 0.369554}, {0.666667, -0.0786098},
  {2., 0.608108}, {3.33333, 1.24608}, {4.66667, 1.61062}, {6., 1.82523}}
```

а) С помощью функции **Array** пакета **Mathematica** создадим массив dif размера $(n+1)\times(n+1)$ с начальными индексами $(0,0)$, в котором будем хранить конечные разности. С учетом введенных обозначений, dif[i, k] – это конечная разность порядка k в точке x_i.

```
In[4]:= Array[dif, {n + 1, n + 1}, {0, 0}];
        массив
```

Определим элементы массива dif, которые соответствуют пустым клеткам таблицы:

```
In[5]:= For[k = 1, k ≤ n, k++,
        цикл ДЛЯ
          For[i = n, i ≥ n - k, i--, dif[i, k] = ""]];
          цикл ДЛЯ
```

Заполним элементы первого столбца массива dif[i, 0] – конечные разности нулевого порядка. Им соответствуют значения $f(x_i)$, которые хранятся во втором столбце таблицы данных data, т. е. data[[i + 1, 2]]. Затем заполним элементы массива dif с конечными разностями остальных порядков:

```
In[6]:= For[i = 0, i ≤ n, i++, dif[i, 0] = data[[i + 1, 2]]];
        цикл ДЛЯ
        For[k = 1, k ≤ n, k++,
        цикл ДЛЯ
          For[i = 0, i ≤ n - k, i++,
          цикл ДЛЯ
           dif[i, k] = dif[i + 1, k - 1] - dif[i, k - 1]]];
```

Выведем построенную таблицу конечных разностей:

In[8]:= `tab = Array[dif, {n + 1, n + 1}, {0, 0}];`
массив

In[9]:= `PaddedForm[TableForm[tab], {6, 5}]`
форма числ⋯ табличная форма

Out[9]//PaddedForm=

0.67244	-0.30289	-0.14528	1.28016	-2.46378	3.42271	-4.03342
0.36955	-0.44816	1.13488	-1.18362	0.95893	-0.61071	
-0.07861	0.68672	-0.04874	-0.22470	0.34821		
0.60811	0.63797	-0.27344	0.12352			
1.24608	0.36453	-0.14992				
1.61062	0.21461					
1.82523						

б) Построим первый интерполяционный многочлен Ньютона pn1[x]:

In[10]:= $t = \frac{x - a}{h}$; `pn1[x_] = dif[0, 0]; p[t_] = 1;`

In[11]:= `For[k = 1, k ≤ n, k++,`
цикл ДЛЯ

```
    p[t_] = p[t] * (t - k + 1);
```

$$pn1[x\_] = pn1[x] + \frac{dif[0,\ k]}{k\,!} * p[t]\Big]$$

Выведем полученный многочлен:

In[12]:= `pn1[x]`

Out[12]=
$$0.67244 - 0.227165\ (2 + x) - 0.0544789\ (2 + x)\left(-1 + \frac{3\ (2 + x)}{4}\right) +$$
$$0.16002\ (2 + x)\left(-2 + \frac{3\ (2 + x)}{4}\right)\left(-1 + \frac{3\ (2 + x)}{4}\right) -$$
$$0.0769932\ (2 + x)\left(-3 + \frac{3\ (2 + x)}{4}\right)\left(-2 + \frac{3\ (2 + x)}{4}\right)\left(-1 + \frac{3\ (2 + x)}{4}\right) +$$
$$0.0213919\ (2 + x)\left(-4 + \frac{3\ (2 + x)}{4}\right)\left(-3 + \frac{3\ (2 + x)}{4}\right)\left(-2 + \frac{3\ (2 + x)}{4}\right)\left(-1 + \frac{3\ (2 + x)}{4}\right) -$$
$$0.00420148\ (2 + x)\left(-5 + \frac{3\ (2 + x)}{4}\right)\left(-4 + \frac{3\ (2 + x)}{4}\right)$$
$$\left(-3 + \frac{3\ (2 + x)}{4}\right)\left(-2 + \frac{3\ (2 + x)}{4}\right)\left(-1 + \frac{3\ (2 + x)}{4}\right)$$

Упростим вид многочлена pn1[x]:

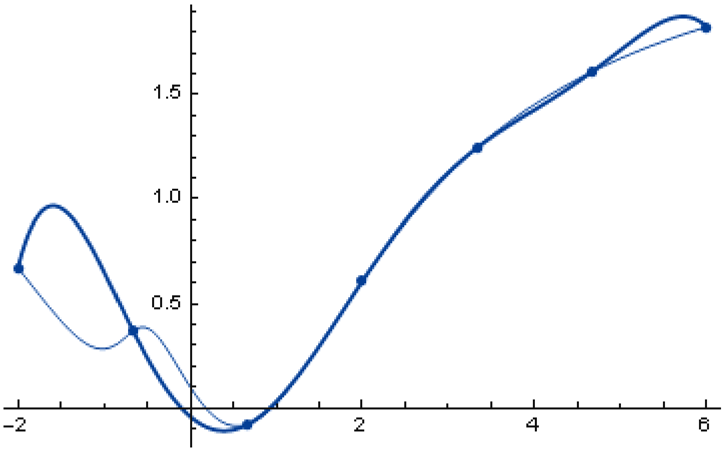
In[13]:= `pn1[x_] = Simplify[pn1[x]]`

Out[13]=
$$-0.0418053 - 0.349915\ x + 0.450958\ x^2 +$$
$$0.0244787\ x^3 - 0.0661214\ x^4 + 0.0147448\ x^5 - 0.000997032\ x^6$$

Построим графики (график многочлена Ньютона изображен жирной линией):

```
In[14]:= gr1 = Plot[f[x], {x, -2, 6}];
              |график функции
         gr2 = ListPlot[data, PlotStyle → PointSize[0.015]];
              |диаграмма разб⋯  |стиль графика |размер точки
         gr3 = Plot[pn1[x], {x, -2, 6}, PlotStyle → Thickness[0.0056]];
              |график функции              |стиль графика |толщина
         Show[gr1, gr2, gr3]
```

Out[15]=



в) Построим интерполяционный многочлен Ньютона с помощью встроенной функции **InterpolatingPolynomial** и упростим его вид:

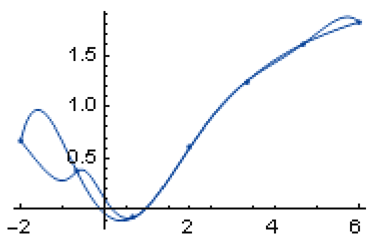
```
In[16]:= Np[x_] = InterpolatingPolynomial[data, x]
                  |интерполяционный многочлен
In[17]:= Np[x_] = Simplify[Np[x]]
                  |упростить
```

Out[17]= $-0.0418053 - 0.349915\,x + 0.450958\,x^2 + 0.0244787\,x^3 - 0.0661214\,x^4 + 0.0147448\,x^5 - 0.000997032\,x^6$

Построим графики:

```
In[18]:= gr4 = Plot[Np[x], {x, -2, 6}, PlotStyle → Thickness[0.0056]];
              |график функции              |стиль графика |толщина
In[19]:= Show[gr1, gr2, gr4, ImageSize → Small]
```

Out[19]=



Найдем максимум абсолютной погрешности интерполирования многочленом Np[x] на отрезке [–2, 6]:

In[20]:= `FindMaximum[Abs[f[x] - Np[x]], {x, -2, 6}]`
найти макси··· абсолютное значение

Out[20]= `{0.549889, {x → -1.40838}}`

Выполним интерполяцию с помощью встроенной функции **Interpolation**:

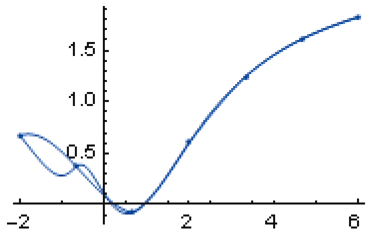
In[21]:= `Tfn = Interpolation[data]`

Out[21]= InterpolatingFunction[Domain: {{-2., 6.}} Output: scalar]

Построим графики:

In[22]:= `gr5 = Plot[Tfn[x], {x, -2, 6}, PlotStyle → Thickness[0.0056]];`

In[23]:= `Show[gr1, gr2, gr5, ImageSize → Small]`

Out[23]=


Найдем максимум абсолютной погрешности интерполирования функцией Tfn[x] на отрезке $[-2, 6]$:

In[24]:= `FindMaximum[Abs[f[x] - Tfn[x]], {x, -2, 6}]`

Out[24]= `{0.274704, {x → -1.2546}}`

г) Вычислим значения $f(x)$ и построенных интерполяционных функций в точке $x = 2{,}28$:

In[25]:= `{f[2.28], pn1[2.28], Np[2.28], Tfn[2.28]}`

Out[25]= `{0.769497, 0.77637, 0.77637, 0.753643}` ▲

Пример 2. Построить график кусочно-заданной на отрезке $[-2, 3]$ функции

$$f(x) = \begin{cases} a_1(x+2)^2 + b_1, & -2 \le x < -1, \\ a_2(x+1)^2 + b_2, & -1 \le x < 0, \\ a_3x^2 + b_3, & 0 \le x < 1, \\ a_4(x-1)^2 + b_4, & 1 \le x < 2, \\ a_5(x-2)^2 + b_5, & 2 \le x \le 3, \end{cases}$$

где $a_k = (-1)^k (2k-1)$ $(k = \overline{1,5})$, $b_1 = 3{,}5$, $b_k = a_{k-1}h^2 + b_{k-1}$ $(k = \overline{1,4})$, h – шаг разбиения отрезка $[-2, 3]$ на частичные отрезки. Вычислить $f(2{,}28)$, найти максимальное значение функции $f(x)$ на отрезке $[-2, 3]$.

Δ Отрезок $[-2, 3]$ разделен точками -1, 0, 1 и 2 на пять равных частичных отрезков. На каждом из промежутков $[x_{k-1}, x_k)$ $(k = \overline{1,4})$ и на отрезке $[x_4, x_5] = [2, 3]$ функция $f(x)$ имеет вид $y = a_k(x - x_{k-1})^2 + b_k$. Введем концы отрезка *a* и *b*, число частичных отрезков *n*, шаг *h*. Сформируем список *X* точек x_k $(k = \overline{0,5})$, причем точке x_k будет соответствовать элемент $X[[k+1]]$.

```
In[1]:= a = -2; b = 3; n = 5; h = (b - a)/n;

In[2]:= X = Table[a + h i, {i, 0, n}]
            таблица значений

Out[2]= {-2, -1, 0, 1, 2, 3}
```

Далее сформируем списки коэффициентов a_k и b_k:

```
In[3]:= Ak = Table[(-1)^k (2 k - 1), {k, 1, n}];
             таблица значений

In[4]:= Bk = Table[0, {k, 1, n}]; Bk[[1]] = 3.5;
             таблица значений

In[5]:= For[k = 2, k ≤ n, k++, Bk[[k]] = Ak[[k - 1]] h^2 + Bk[[k - 1]]]; Bk
        цикл ДЛЯ

Out[5]= {3.5, 2.5, 5.5, 0.5, 7.5}
```

Чтобы определить кусочно-заданную функцию $f(x)$ воспользуемся встроенной функцией **Piecewise**. Аргумент этой функции должен иметь вид $\{\{f_1, cond_1\}, \{f_2, cond_2\}, \ldots, \{f_n, cond_n\}\}$. Создадим соответствующий список dat:

```
In[6]:= dat = Table[If[k < n, {Ak[[k]] (x - X[[k]])^2 + Bk[[k]], X[[k]] ≤ x < X[[k + 1]]},
              табл⋯ условный оператор
            {Ak[[k]] (x - X[[k]])^2 + Bk[[k]], X[[k]] ≤ x ≤ X[[k + 1]]}], {k, 1, n}]

Out[6]= {{3.5 - (2 + x)^2, -2 ≤ x < -1}, {2.5 + 3 (1 + x)^2, -1 ≤ x < 0},
         {5.5 - 5 x^2, 0 ≤ x < 1}, {0.5 + 7 (-1 + x)^2, 1 ≤ x < 2}, {7.5 - 9 (-2 + x)^2, 2 ≤ x ≤ 3}}
```

Теперь можно задать функцию $f(x)$ (обозначим ее $\mathrm{f1}$):

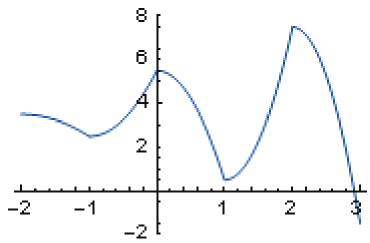
In[7]:= `f1 = Piecewise[dat]`

Out[7]= $\begin{cases} 3.5-(2+x)^2 & -2\le x<-1 \\ 2.5+3\,(1+x)^2 & -1\le x<0 \\ 5.5-5\,x^2 & 0\le x<1 \\ 0.5+7\,(-1+x)^2 & 1\le x<2 \\ 7.5-9\,(-2+x)^2 & 2\le x\le 3 \\ 0 & \text{True} \end{cases}$

Построим график функции:

In[8]:= `Plot[f1, {x, -2, 3}, ImageSize → Small]`

Out[8]=



Вычислим значение функции в точке $x = 2{,}28$ и найдем ее максимум на отрезке $[-2, 3]$:

In[9]:= `f1 /. x → 2.28`

Out[9]= 6.7944

In[10]:= `FindMaximum[f1, {x, -2, 3}]`
найти максимум

Out[10]= {3.5, {x → -2.}}

▲

Пример 3. Используя значения функции

$$f(x) = \frac{5x^2 - 4x\,\mathrm{arctg}(3x+2) + \ln 2}{2x^2 + 7}$$

в равноотстоящих узлах x_i отрезка $[-2, 6]$ при разбиении его на 6 частей, выполнить следующие действия:

а) выполнить интерполяцию сплайном с помощью функции **Interpolation[***data*, **Method->"Spline"]**, проиллюстрировать графически;

б) построить интерполяционный кубический сплайн с помощью функции **SplineFit[***data*,**Cubic]**, проиллюстрировать графически;

в) построить кубический многочлен наилучшего среднеквадратичного приближения с помощью функции **Fit**, проиллюстрировать графически;

г) найти значения функции $f(x)$ и построенных функций в точке $x = 2{,}28$.

Δ Введем функцию $f(x)$, границы отрезка a и b, количество частей разбиения отрезка n, вычислим шаг h и построим таблицу значений функции:

```
In[1]:= f[x_] = (5 x^2 - 4 x ArcTan[3 x + 2] + Log[2])/(2 x^2 + 7);

In[2]:= a = -2; b = 6; n = 6; h = (b - a)/n;

In[3]:= data = N[Table[{a + i h, f[a + i h]}, {i, 0, n}]];
```

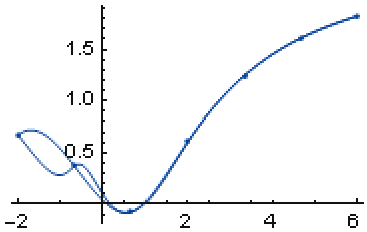
а) Построим интерполяционный сплайн с помощью функции **Interpolation**:

```
In[4]:= spl1 = Interpolation[data, Method → "Spline"]

Out[4]= InterpolatingFunction[ Domain: {{-2., 6.}}
                               Output: scalar ]
```

Построим графики (график сплайна изображен жирной линией):

```
In[5]:= gr1 = Plot[f[x], {x, -2, 6}];
        gr2 = ListPlot[data, PlotStyle → PointSize[0.015]];
        gr3 = Plot[spl1[x], {x, -2, 6}, PlotStyle → Thickness[0.0056]];

In[6]:= Show[gr1, gr2, gr3, ImageSize → Small]
```

Out[6]=



б) Построим интерполяционный кубический сплайн с помощью функции **SplineFit**. Предварительно загрузим пакет сплайн-интерполяции командой **Needs[“Splines`”]**), чтобы эта функция была доступна.

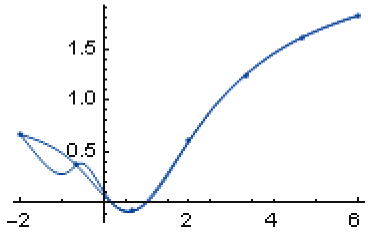
```
In[7]:= Needs["Splines`"]

In[8]:= spl2 = SplineFit[data, Cubic]

Out[8]= SplineFunction[Cubic, {0., 6.}, <>]
```

В результате получаем параметрически заданную функцию с указанной областью изменения параметра. Если интерполирование функцией **SplineFit** проводится по системе равноотстоящих узлов отрезка $[a,b]$, то между параметром t сплайна и переменной x существует линейная зависимость $t=\dfrac{x-a}{h}$, где h – шаг интерполяции.

Построим графики:

```
In[9]:= Clear[t]; gr4 = ParametricPlot[spl2[t], {t, 0, 6}];
In[10]:= Show[gr1, gr2, gr4, ImageSize → Small]
```

Out[10]=



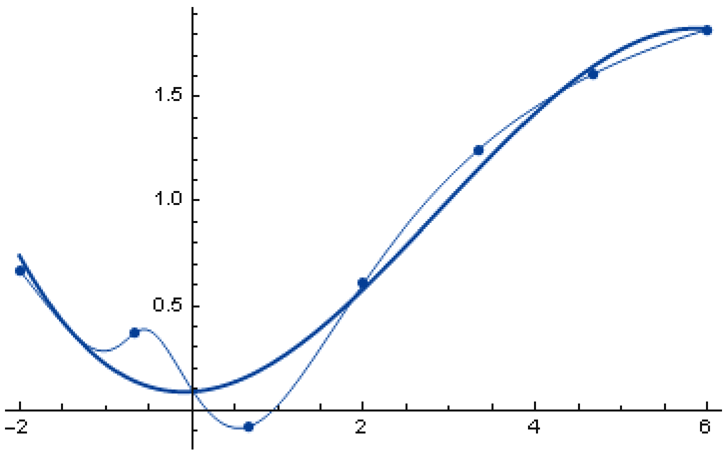
в) Аппроксимируем функцию $f(x)$ многочленом третьей степени наилучшего среднеквадратичного приближения с помощью функции **Fit**.

```
In[11]:= p3 = Fit[data, {1, x, x^2, x^3}, x]
Out[11]= 0.0879585 + 0.0264223 x + 0.143234 x^2 - 0.0165582 x^3
In[12]:= p[x_] = p3
Out[12]= 0.0879585 + 0.0264223 x + 0.143234 x^2 - 0.0165582 x^3
```

Построим графики:

```
In[13]:= gr5 = Plot[p[x], {x, -2, 6}, PlotStyle → Thickness[0.0056]];
In[14]:= Show[gr1, gr2, gr5]
```

Out[14]=



г) Вычислим значения $f(x)$ и построенных функций в точке $x = 2{,}28$:

```
In[15]:= {f[2.28], spl1[2.28], spl2[(2.28 - a)/h], p[2.28]}
Out[15]= {0.769497, 0.77404, {2.28, 0.775649}, 0.696537}
```

Как видно, функция spl2, построенная с помощью **SplineFit**, возвращает координаты точки кривой. ▲