Численное дифференцирование

Методы численного дифференцирования применяются, если исходную функцию y*(x)* трудно или невозможно продифференцировать аналитически.

Пусть в некоторой точке x^* требуется вычислить производные первого, второго и т. д. порядков от дискретно заданной функции (1) . Могут иметь место следующие два случая: а) точка $x^* \in (x_{i-1}, x_i)$, $i = \overline{1, n}$, и б) точка $x^* = x_i$, $i = \overline{1, n-1}$, т. е. совпадает с одним из внутренних узлов заданной таблицы.

x_i	x_0	x_1	$\ldots$	x_n
y_i	y_0	y_1	$\ldots$	y_n

(1)

Тогда *в первом случае* заданная таблица сглаживается какой-либо функцией $\varphi(x)$, являющейся глобальным (локальным) интерполяционным многочленом или многочленом, полученным по МНК с некоторой погрешностью $R_n(x)$, в результате чего имеют место следующие равенства

$$\begin{aligned}
y(x) &= \varphi(x) + R_n(x), & y(x^*) &= \varphi(x^*) + R_n(x^*); \\
y'(x) &= \varphi'(x) + R'_n(x), & y'(x^*) &= \varphi'(x^*) + R'_n(x^*); \\
y''(x) &= \varphi''(x) + R''_n(x), & y''(x^*) &= \varphi''(x^*) + R''_n(x^*); \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots
\end{aligned} \qquad (2)$$

Следует отметить, что процедура численного дифференцирования является *некорректной* в том смысле, что близость искомой функции $y(x)$ и сглаживающей функции $\varphi(x)$ не гарантирует близости их производных (рис. 1). Более того, они могут иметь в одной и той же точке x^* производные различных знаков. Тем не менее формулы (2) широко используются на практике.

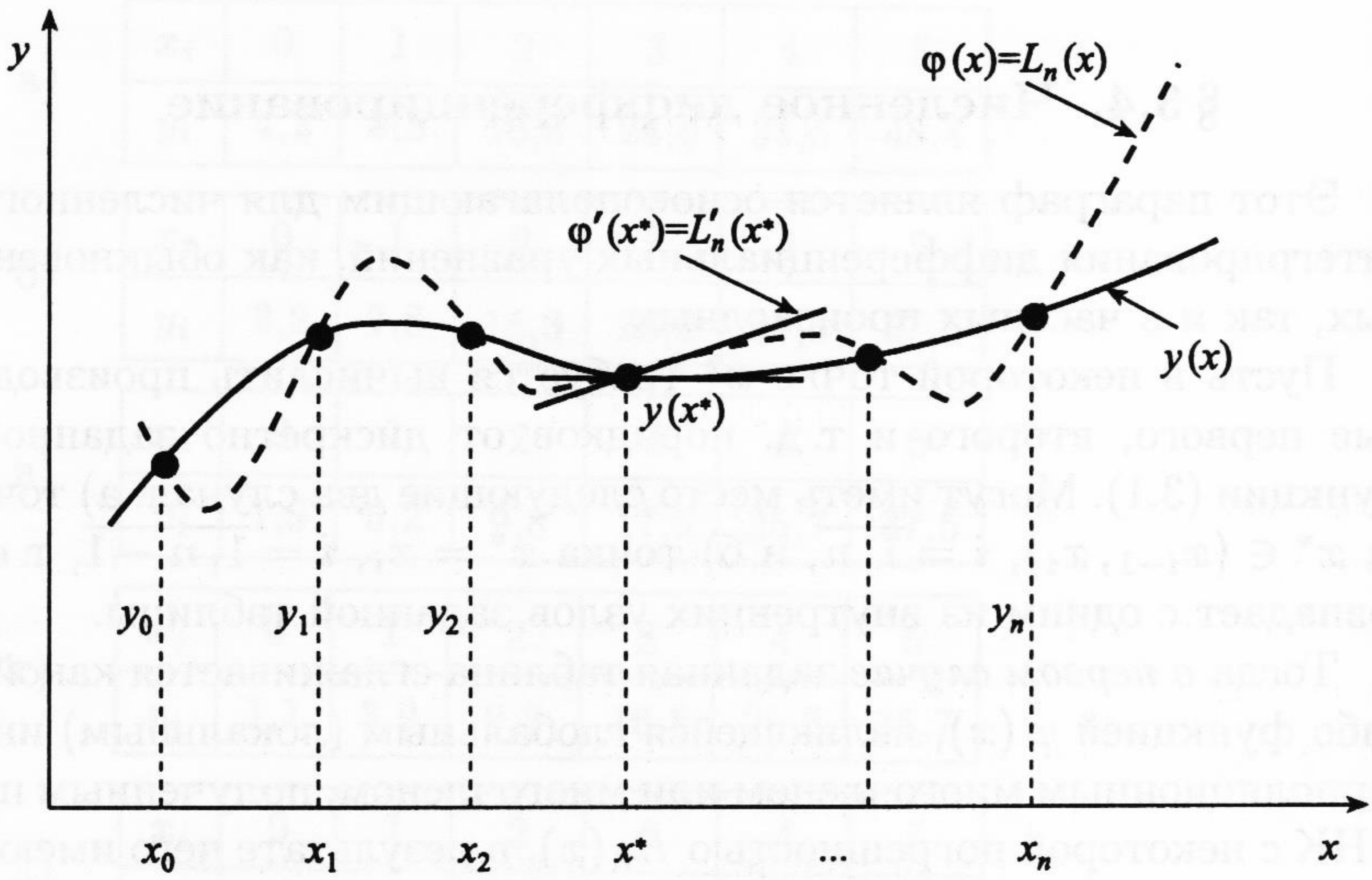

Рис. 1 . К понятию некорректности численного дифференцирования

Согласно ур.(2) погрешность производной интерполирующей функции равна производной от погрешности этой функции. То же самое справедливо и для производных высших порядков.

Во *втором случае* ($x^* = x_i$, $i = \overline{1, n-1}$) используется аппарат разложения функций в ряд Тейлора, для чего функция в точке x^* должна иметь достаточное число производных. С этой целью предполагается, что заданная таблица (3.1) является сеточной функцией для некоторой функции $y(x)$, имеющей в точке x^* производные до четвертого порядка включительно, т. е. что $y_i = y(x_i)$.

Тогда внутренний узел $x^* = x_i$, $i = \overline{1, n-1}$, окружают узлы x_{i-1}, x_{i+1} (рис. 3.10), причем $x_{i+1} - x_i = x_i - x_{i-1} = h$. Тогда, разлагая значения y_{i-1}, y_{i+1} *на точной функции* в ряд Тейлора в окрестности точки x_i до производной четвертого порядка включительно, получим

$$y_{i-1} = y(x_i - h) = y_i - y'_i h + y''_i \frac{h^2}{2} -$$

$$- y'''_i \frac{h^3}{6} + y^{IV}(\xi) \frac{h^4}{24}, \quad \xi \in (x_{i-1}, x_i), \qquad (3)$$

$$y_{i+1} = y(x_i + h) = y_i + y'_i h + y''_i \frac{h^2}{2} +$$

$$+ y'''_i \frac{h^3}{6} + y^{IV}(\xi) \frac{h^4}{24}, \quad \xi \in (x_i, x_{i+1}). \qquad (4)$$

Выразим вначале y'_i из (3), а затем из (4), разделив предварительно на h и оставляя слагаемые с первой степенью шага h, получим

$$\overline{y}'_i = \frac{y_i - y_{i-1}}{h} + O(h) = \frac{\Delta \overline{y}_i}{h} + O(h), \qquad (5)$$

$$y'_i = \frac{y_{i+1} - y_i}{h} + O(h) = \frac{\Delta y_i}{h} + O(h). \qquad (6)$$

Вычтем из (4) выражение (3), разделим полученное соотношение на $2h$, получим следующее значение производной первого порядка в точке x_i (слагаемые с производными четного порядка сокращаются за исключением слагаемого с производной четвертого порядка):

$$\overset{\circ}{y}{}'_i = \frac{y_{i+1} - y_{i-1}}{2h} + O(h^2) = \frac{\Delta \overset{\circ}{y}_i}{2h} + O(h^2), \qquad (7)$$

где $\Delta \overset{\circ}{y}_i = y_{i+1} - y_{i-1}$ — центральная разность первого порядка.

Выражение (5) определяет производную первого порядка в узле x_i с помощью отношения конечных разностей *слева*. Она имеет первый порядок аппроксимации относительно шага h.

Выражение (6) определяет производную первого порядка в узле x_i с помощью отношения конечных разностей *справа*. Она также имеет первый порядок аппроксимации относительно шага h.

Выражение (7) определяет производную первого порядка в узле x_i с помощью отношения *центральных разностей*. Она имеет второй порядок аппроксимации относительно шага h.

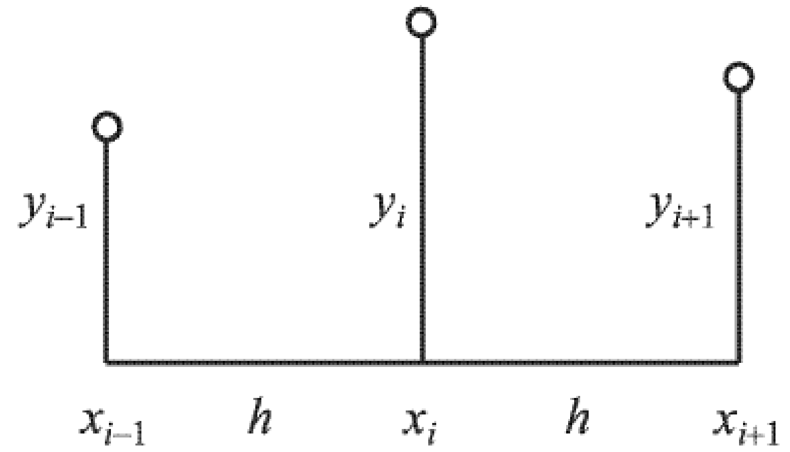

Рис. 2 Численное дифференцирование с помощью отношения конечных разностей

Сложим выражения (3) и (4) , разделим на h^2 (слагаемые с производными нечетного порядка сокращаются), получим

$$y''(x_i) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + O(h^2) = \frac{\Delta^2 y_i}{h^2} + O(h^2). \qquad (8)$$

Выражение (8) определяет производную второго порядка в узле x_i с помощью отношения центральных разностей второго порядка. Она имеет второй порядок аппроксимации относительно шага h.

Определим *порядок точности метода* численного дифференцирования с помощью отношения конечных разностей, для чего рассмотрим, как из выражений (3) , (4) получаются производные (5) – (7) . Например, производная (5) получена из (3) следующим образом:

$$\overline{y}'_i = \frac{y_i - y_{i-1}}{h} + y''_i \frac{h}{2} - y'''_i \frac{h^3}{2} + O(h^3).$$

Поскольку шаг h является дробной величиной, то главный член погрешности содержит первую степень h $\left(\text{слагаемое } y_i'' \dfrac{h}{2}\right)$, остальные члены погрешности являются членами более высокого порядка малости (слагаемые, начиная с h^2 и далее), поэтому, отбрасывая их и сохраняя главный член погрешности, следующий сразу за отношением конечных разностей, получим формулу (5) . Аналогично получены и остальные формулы.

Порядком точности метода численного дифференцирования с помощью отношения конечных разностей называют показатель степени h в главном члене погрешности.

Таким образом, односторонние производные (5) и (6) имеют 1-й порядок точности и поэтому менее точны, чем центрально-разностная производная (7) , имеющая второй порядок точности.

Вычисление первой производной в середине интервала по четырём узлам сетки:

$$f'(x_1 + 1.5h) = (-y_4 + 27y_3 - 27y_2 + y_1)/(24h) + O9(h^4), \qquad (9)$$

а второй производной в центре по пяти узлам:

$$f''(x_3) = (-y_5 + 16y_4 - 30y_3 + 16y_2 - y_1)/(12h^2) + O(h^4) \qquad (10)$$

Минимальное количество узлов, необходимое для вычисления конечных разностей какого-либо порядка, должно быть на единицу больше этого порядка.

Сравним разные методы численной оценки первой производной функции на примере известной функции $y=sin(x)$ на отрезке $[0, \pi]$. Перепишем уравнения (5), (6) и (7) в удобном для нас виде с учётом того, что задана равномерная сетка $0 - x_1 < \ldots < x_n - \pi$,

где $h = x_{i+1} - x_i$ - шаг сетки. Пусть вычисляется первая производная с помощью "правой" конечной разности:

$$f'(x_i) = (y_{i+1} - y_i)/h, \qquad i = 1, \ldots, n-1;$$

с помощью "левой" конечной разности:

$$f'(x_i) = (y_i - y_{i-1})/h, \qquad i = 2, \ldots, n;$$

с помощью центральной разности 2-го порядка точности:

$$f'(x_i) = (y_{i+1} - y_{i-1})/(2h) \qquad i = 2, \ldots, n-1;$$

и, наконец, с помощью формулы (9) повышенной точности – 4-го порядка:

$$f'(x_i - 0.5h) = (-y_{i+1} + 27y_i - 27y_{i-1} + y_{i-2})/(24h).$$

Вычисление всех четырёх производных производится в программе Data_sheet1.m.

Сравнение точности вычисления второй производной по формулам (8) и (10) проведено в программе Data_sheet2.m. В листинге Data_sheet3.m приведён код программы численной оценки второй производной в точке x=π/4 по формуле (10).

Data_sheet1.m

```
%Программа сравнения простейших формул
%численного дифференцирования с точным
%значением  производной
%определим шаг сетки
h=0.2;
%определим сетку
x=0:h:pi;
n=length(x);
%анализируемая функция y=sin(x)
%точное значение производной dy=cos(x)
dy=cos(x);
%определим производную по формуле (6)
%правой конечной разности и соответствующую
%абсолютную погрешность
for i=1:(n-1)
   dy1(i)=(sin(x(i+1))-sin(x(i)))/h;
   er1(i)=abs(dy(i)-dy1(i));
end
```

```
%определим производную по формуле (5)
%левой конечной разности и соответствующую
%абсолютную погрешность
for i=2:n
   dy2(i)=(sin(x(i))-sin(x(i-1)))/h;
   er2(i)=abs(dy(i)-dy2(i));
end
%определим производную по формуле (7)
%центральной разности и соответствующую
%абсолютную погрешность
for i=2:(n-1)
   dy3(i)=(sin(x(i+1))-sin(x(i-1)))/(2*h);
   er3(i)=abs(dy(i)-dy3(i));
end
```

```
%определим производную по формуле (9)
%четвертого порядка точности и соответствующую
%абсолютную погрешность, которая вычисляется
%в точке x(i)-0.5*h
for i=3:(n-1)
    dy4(i)=(-sin(x(i+1))+27*sin(x(i))-...
          27*sin(x(i-1))+sin(x(i-2)))/(24*h);
    er4(i)=abs(cos(x(i)-0.5*h)-dy4(i));
end
%рисуем все три абсолютные ошибки на одном графике
plot(x([1:(n-1)]),er1([1:(n-1)]),'-o',...
             x([2:n]),er2([2:n]),'-p',...
     x([2:(n-1)]),er3([2:(n-1)]),'-h',...
     x([3:(n-1)]),er4([3:(n-1)]),'-*');
```

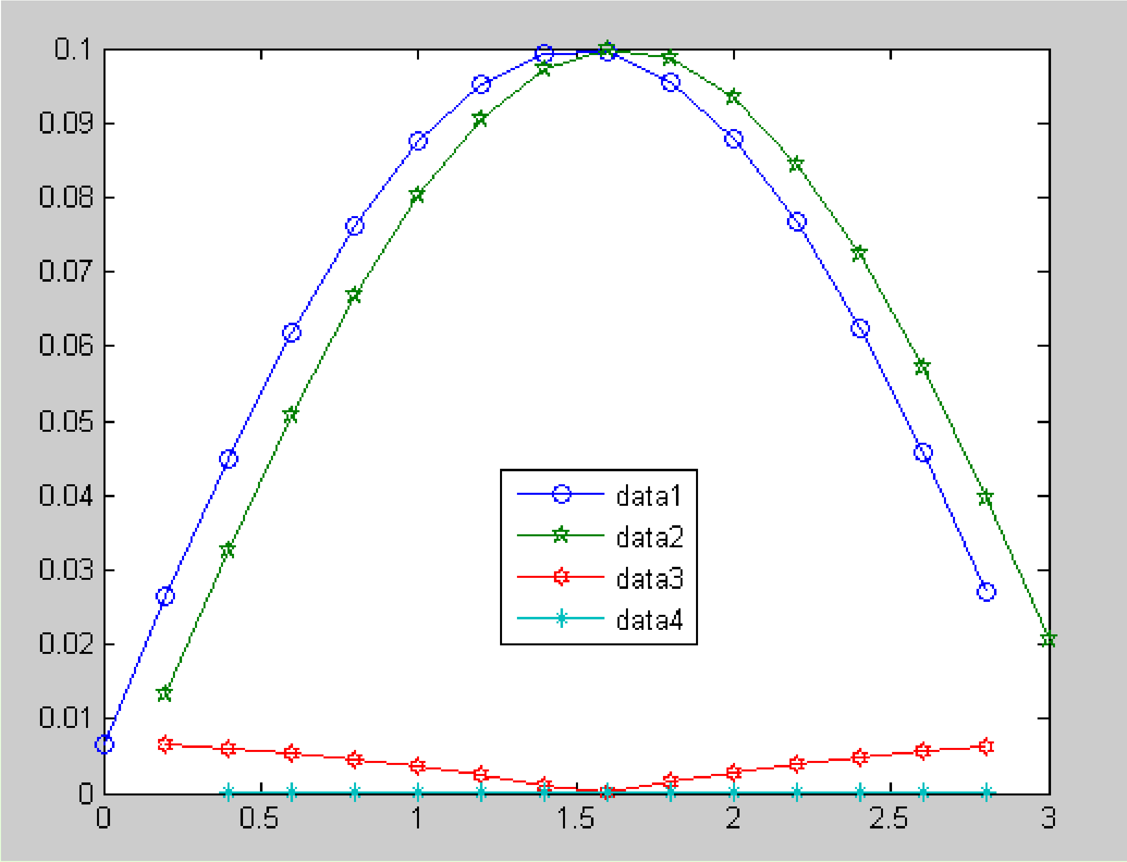

Рис. 3. Абсолютные ошибки формул численного дифференцирования (5)-(7) , (9)

Data_sheet2.m

```
%Программа сравнения двух формул для
% численной оценки второй производной
%определим шаг сетки
h=0.2;
%определим сетку
x=0:h:pi;
n=length(x);
%анализируемая функция y=sin(x)
%точное значение второй производной d2y=-sin(x)
d2y=-sin(x);
%определим вторую производную по формуле (8)
for i=2:(n-1)
   d2y1(i)=(sin(x(i+1))-2*sin(x(i))+sin(x(i-1)))/h^2;
   er1(i)=abs(d2y(i)-d2y1(i));
end
```

%определим вторую производную по формуле (10)

```
for i=3:(n-2)
    d2y2(i)=(-sin(x(i+2))+16*sin(x(i+1))-30*sin(x(i))+...
              16*sin(x(i-1))-sin(x(i-2)))/(12*h^2);
    er2(i)=abs(d2y(i)-d2y2(i));
end
```

%рисуем абсолютные ошибки на одном графике

```
plot(x([2:(n-1)]),er1([2:(n-1)]),'-o',...
     x([3:(n-2)]),er2([3:(n-2)]),'-p');
```

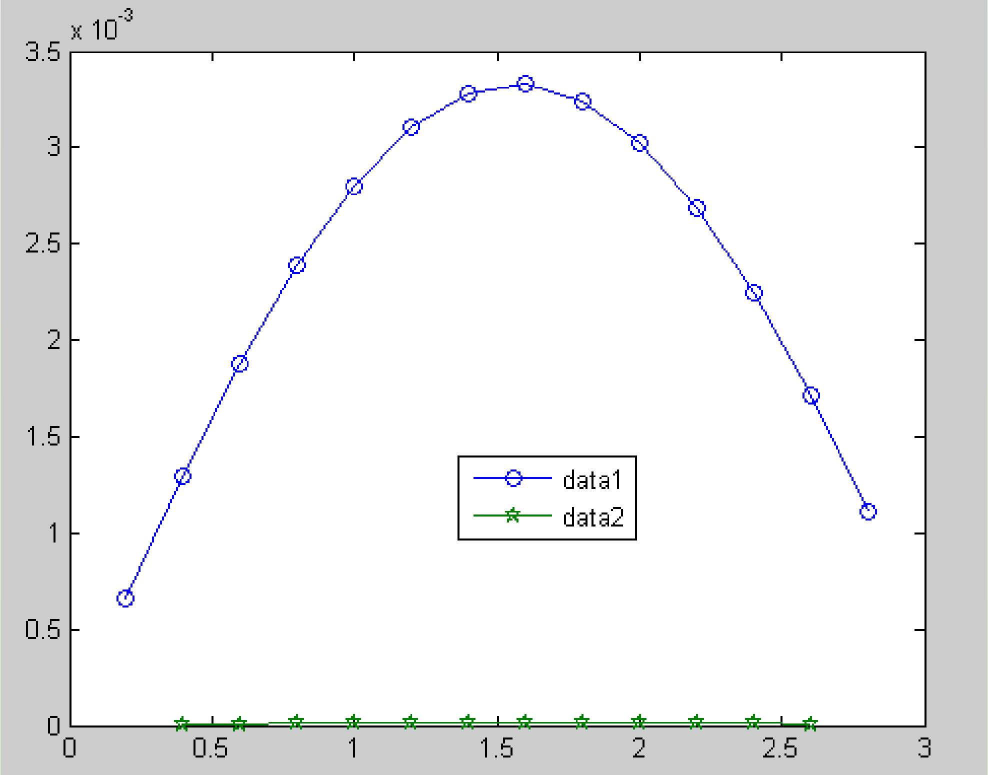

Рис. 4. Абсолютные ошибки формул численного дифференцирования (8) и (10).

При численном дифференцировании отсутствует устойчивость решения, т.к. приходится вычитать друг из друга близкие значения функции. Это приводит к уничтожению первых значащих цифр, т.е. к потере части достоверных знаков числа. А так как значения функции обычно известны с определенной погрешностью, то все значащие цифры могут быть потеряны. На графике рис.5. кривая (1) соответствует уменьшению погрешности дифференцирования при уменьшении шага; кривая (2) представляет собой неограниченно возрастающий (осциллирующий) вклад *неустранимой погрешности* исходных данных – значений функции *y(x)*. Критерий выхода за оптимальный шаг при его уменьшении – «разболтка» решения: зависимость результатов вычислений становится нерегулярно зависящей от величины шага.

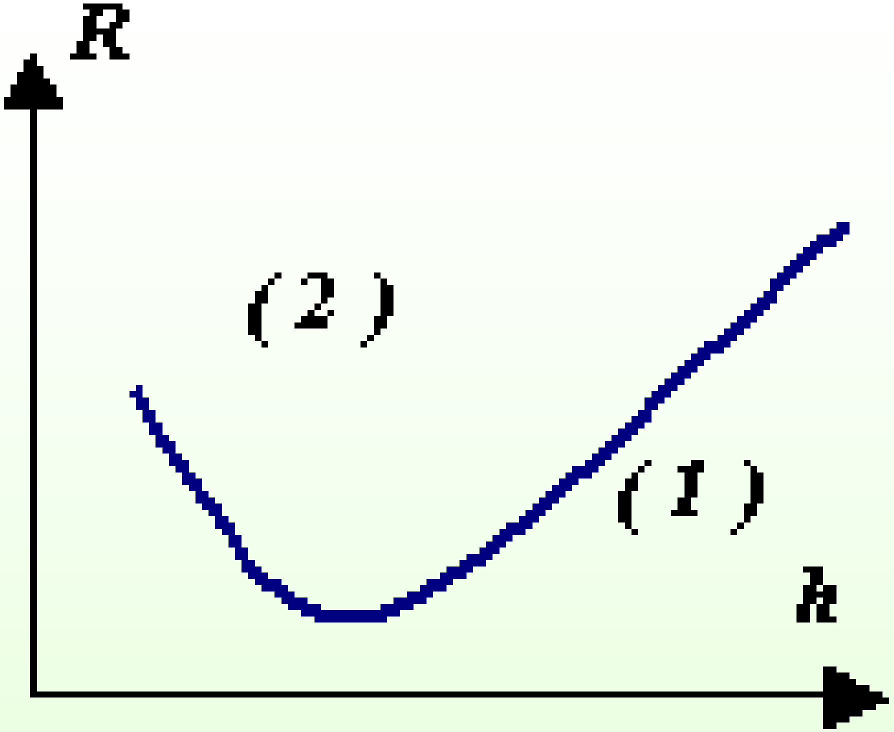

Рис.5. Зависимость погрешности численного дифференцирования от величины шага

Data_sheet3.m

```
%Программа анализа зависимости ошибки
%аппроксимации второй производной функции по ур.(8)
%в зависимости от величины шага сетки
%в фиксированной точке
%очищаем рабочую область
clear all
%выбираем начальное значение шага сетки
h(1)=1.0;
%определяем число делений шага пополам
n=30;
%формируем массив шагов сетки
for i=2:n
   h(i)=h(i-1)/2;
end
```

```
%выбираем фиксированную точку для
%функции численной оценки второй
%производной функции y=sin(x)
x=pi/4;
%находим численное значение второй производной
%при различных значениях h и сравниваем их с
%эталоном
for i=1:n
   d2y=(sin(x+h(i))-2*sin(x)+sin(x-h(i)))/h(i)^2;
   er(i)=abs(-sin(x)-d2y);
end
%строим график зависимости ошибки аппроксимации
%в зависимости от величины шага сетки h
loglog(h,er);
```

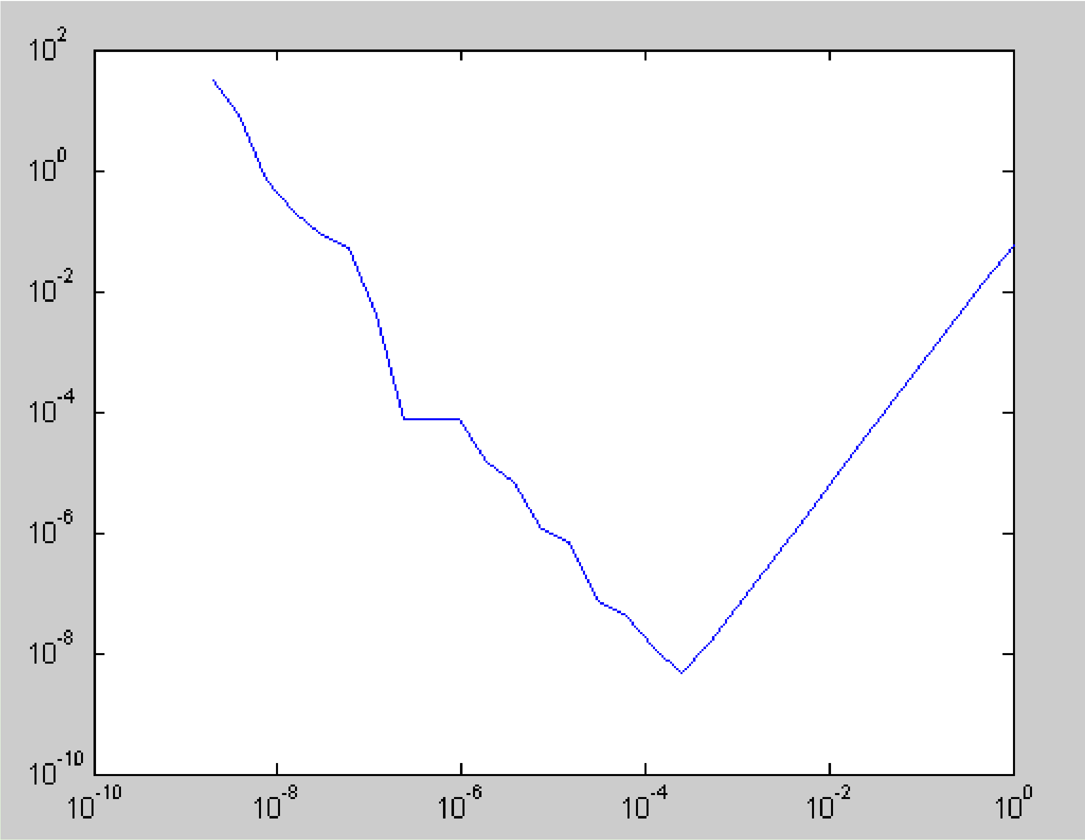

Рис. 6. Зависимость ошибки аппроксимации второй производной по формуле (8) в зависимости от шага сетки

Метод Рунге уточнения формул численного дифференцирования. Из формул (5) - (7) видно, что метод p-го порядка численного дифференцирования совпадает с показателем степени шага h в главном члене погрешности и имеет вид

$$f'(x) = \varphi_h'(x) + h^p \cdot \psi(x) + O\left(h^{p+1}\right) + O\left(h^{p+2}\right) + \dots, \qquad (11)$$

где

$$f'(x) \approx \varphi_h'(x),$$

а остаточный член имеет вид

$$R_p = h^p \psi(x) + O\left(h^{p+1}\right) + O\left(h^{p+2}\right) + \dots$$

С целью повышения на единицу порядка точности метода продифференцируем численно методом p-го порядка функцию $f(x_i) = y_i$, $i = \overline{0, n}$, с шагом h, получим выражение (11) . Затем продифференцируем численно функцию тем же методом p-го порядка, с шагом kh, $(k = 1/2;\ 1/4;\ 1/16;\ \dots)$, получим

$$f'(x) = \varphi'_{kh}(x) + (kh)^p \psi(x) + O\left(h^{p+1}\right). \qquad (12)$$

Вычитая из (12) выражение (11) и определяя из полученного равенства $\psi(x)$, находим

$$h^p \psi(x) = \frac{\varphi'_h(x) - \varphi'_{kh}(x)}{k^p - 1} + O\left(h^{p+1}\right). \qquad (13)$$

Выражение (13) можно использовать для оценки погрешности численного дифференцирования.

Подставляя (13) в (12), получаем окончательно

$$f'(x) = \varphi'_{kh}(x) + k^p \frac{\varphi'_h(x) - \varphi'_{kh}(x)}{k^p - 1} + O\left(h^{p+1}\right). \qquad (14)$$

Из (14) видно, что это уже метод порядка $p+1$, т. е. на порядок точнее. В этом и заключается процедура Рунге уточнения численного дифференцирования.

Метод Рунге обобщается на случай произвольного количества сеток q. В этом случае соответствующим образом модифицировав схему расчётов, можно повысить точность аппроксимации до уровня $O(h^{p+q-1})$. Такая схема расчётов называется *схемой Ромберга*.

Рассмотрим пример, иллюстрирующий работу метода Рунге. Выберем для тестирования схему численного дифференцирования (6) $f'(x_i) = (y_{i+1} - y_i)/h \quad i=1,\ldots n-1$, т.е. правую конечную разность, которая имеет первый порядок аппроксимации, т.е. $p=1$. Уточнение по Рунге должно повысить точность до 2-го порядка по шагу сетки. Тестируемой функцией будет выступать $y = sin(x)$.

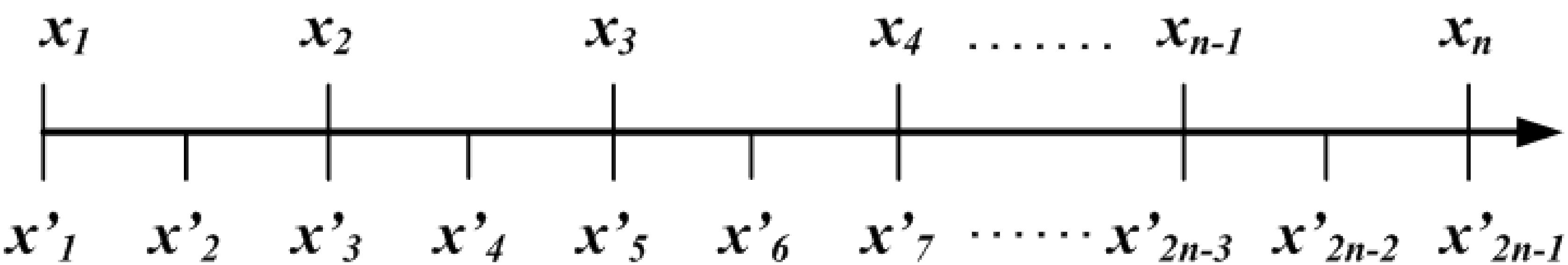

Рис. 7. Схема позиционирования двух сеток в методе Рунге

На рис. 7 приведена схема позиционирования двух сеток по методу Рунге, когда одна из них вдвое более подробная, чем другая. Программа Data_sheet4.m уточняет производную синуса по методу Рунге после пары расчётов на исходной сетке и на сетке с удвоенным количеством узлов.

Data_sheet4.m

```
%Программа иллюстрирующая метод
%Рунге по повышению точности численного
%дифференцирования путем комбинирования
%пары расчетов на двух равномерных сетках,
%причем вторая содержит вдвое большее
%количество узлов
%очищаем рабочее пространство
clear all
%определяем шаг исходной сетки
h=0.1;
%формируем исходную сетку
x=0:h:pi;
%определяем число узлов, входящих в исходную сетку
n=length(x);
%определяем шаг более подробной сетки
hm=h/2;
```

```
%создаем сетку вдвое более подробную
xm=0:hm:pi;
%определяем число узлов, входящих в более
%подробную сетку
m=length(xm);
%рассчитываем производную с помощью правой
%разности на исходной сетке и оцениваем
%соответствующую абсолютную ошибку
for i=1:(n-1)
   dy(i)=(sin(x(i+1))-sin(x(i)))/h;
   er1(i)=abs(cos(x(i))-dy(i));
end
```

```
%рассчитываем производную с помощью правой
%разности на более частой сетке и оцениваем
%соответствующую абсолютную ошибку
for i=1:(m-1)
   dym(i)=(sin(xm(i+1))-sin(xm(i)))/hm;
   er2(i)=abs(cos(xm(i))-dym(i));
end
%уточняем значение производной с помощью метода Рунге по ур.(14)
for i=1:(n-1)
   dyrunge(i)=dy(i)-2*(dy(i)-dym(2*i-1));
   er3(i)=abs(cos(x(i))-dyrunge(i));
end
%строим общий график со всеми тремя кривыми ошибок
plot(x([1:(n-1)]),er1([1:(n-1)]),'-o',...
    xm([1:2:(m-1)]),er2([1:2:(m-1)]),'-p',...
    x([1:(n-1)]),er3([1:(n-1)]),'-h');
```

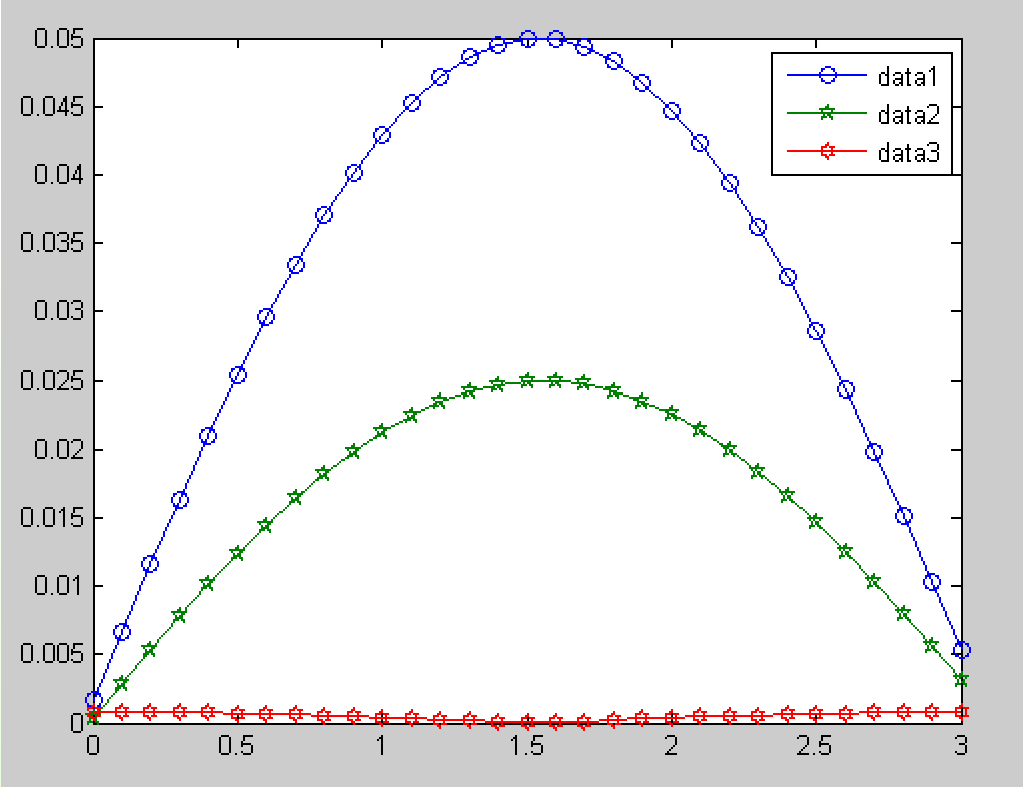

Рис.8. Ошибки двух схем численного дифференцирования: Для исходной схемы и для схемы с удвоенным количеством узлов и ошибка уточняющей процедуры Рунге

Более общий подход для получения выражений остаточных членов интерполяционных формул численного дифференцирования состоит в дифференцировании остаточного члена интерполяционной формулы Лагранжа [19, 61].

$$R_n(x) := f(x) - L_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\Pi_{n+1}(x), \tag{13.22}$$

где $L_n(x)$ — интерполяционный многочлен Лагранжа, построенный для $n+1$ раз дифференцируемой функции $f(x)$ по $n+1$ узлу $x_0, x_1, \ldots, x_n$ (неважно, в какой форме), ξ — некоторая точка из интервала (x_0, x_n), а $\Pi_{n+1}(x) := (x - x_0)(x - x_1)\ldots(x - x_n)$. Из (13.22) следует

$$R_n'(x) := f'(x) - L_n'(x) =$$

$$= \frac{1}{(n+1)!}\left\{\frac{d}{dx}\left[f^{(n+1)}(\xi)\right]\Pi_{n+1}(x) + f^{(n+1)}(\xi)\frac{d}{dx}\left[\Pi_{n+1}(x)\right]\right\}. \tag{13.23}$$

Если величина $\frac{d}{dx}\left[f^{(n+1)}(\xi)\right]$ ограничена, то при подстановке в последнее выражение узловых значений $x = x_i \quad (i = 0, 1, \ldots, n)$ за счет $\Pi_{n+1}(x_i) = 0$ получим простую формулу остаточного члена аппроксимаций $f'(x_i) \approx L'(x_i)$ первой производной в узлах интерполяции:

$$f'(x_i) - L'(x_i) = \frac{f^{(n+1)}(\xi_i)}{(n+1)!}\Pi'_{n+1}(x_i), \quad \xi_i \in (x_0, x_n). \quad (13.24)$$

В случае равноотстоящих узлов $x_i = x_0 + ih$, который здесь, в основном, и рассматривается,

$$\Pi'_{n+1}(x_i) = (-1)^{n-i} i!\,(n-i)!\,h^n,$$

вследствие чего равенство (13.24) трансформируется в формулу

$$f'(x_i) - y'_i = (-1)^{n-i}\frac{i!(n-i)!}{(n+1)!}h^n f^{(n+1)}(\xi_i). \quad (13.25)$$

Из нее отчетливо видно, что при аппроксимации первой производной в точках x_i значениями $y'_i := L'_n(x_i)$, получаемыми дифференцированием интерполяционного многочлена n-й степени, остаточный член имеет n-й порядок относительно шага аппроксимации h.

для оценивания погрешности численного дифференцирования при значениях аргумента, не совпадающих с узловыми, и для получения остаточных членов приближенных формул $f^{(k)}(x) \approx L_n^{(k)}(x)$ при $k > 1$ формула (13.23) малопригодна. В книге [19] на основе интерполяционной формулы Ньютона для неравных промежутков (8.43) и связей между разделенными разностями и производными выводятся следующие формулы остаточных членов:

$$f'(x) - L_n'(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\Pi_{n+1}'(x) + \frac{f^{(n+2)}(\xi_1)}{(n+2)!}\Pi_{n+1}(x), \qquad (13.28)$$

$$f''(x) - L_n''(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\Pi_{n+1}''(x) +$$

$$+ 2\frac{f^{(n+2)}(\xi_1)}{(n+2)!}\Pi_{n+1}'(x) + 2\frac{f^{(n+3)}(\xi_2)}{(n+3)!}\Pi_{n+1}(x), \qquad (13.29)$$

ξ_1 и ξ_2 – межузловые точки из двух соседних интервалов, в которых нужно найти погрешность вычисленной производной.

и вообще, для $k \in \{0, 1, \ldots, n\}$

$$f^{(k)}(x) - L_n^{(k)}(x) =$$

$$= \sum_{j=0}^{k} \frac{k!}{(k-j)!(n+j+1)!} f^{(n+j+1)}(\xi_j)\Pi_{n+1}^{(k-j)}(x). \quad (13.30)$$

Для оценивания точности результатов численного дифференцирования можно воспользоваться простейшими связями между производными и конечными или разделёнными разностями, принимая за $\max\left|f^{(n+1)}(x)\right|$ величину $\max\left\{\left|\Delta^{n+1} y_i\right|\right\}/h^{n+1}$ - для равноотстоящих узлов или

$$\max\{|(n+1)!f(x_0,x_1,\ldots,x_{n+1}|\}.$$

Численное дифференцирование в MATLAB

Символьное дифференцирование выполняет команда

diff(<fun>, <x>, <n>), где <fun> - символьная запись функции или её имя, <x> - переменная дифференцирования, <n> - порядок (номер) производной, которую необходимо найти.

Примеры вычисления производных аналитически.

```
>>diff ('5*sin(2*x)/(cos(2*x)) ',x,1)
ans =
10+10*sin(2*x)^2/cos(2*x)^2
%______________________________
>>syms x
>>F=sym('tan(ln(x) ^(1/3))');
>>diff(F,x,1)
ans =
1/3*(1+tan(log(x)^(1/3))^2)/log(x)^(2/3)/x
```

Функция дифференцирования имеет следующие особенности. Если переменная дифференцирования x в выражении diff отсутствует, а функция имеет вид diff(f), то программа не выдает ошибки. Она осуществляет дифференцирование по переменной функции f в порядке, обратном алфавиту.

Например, если функция f содержит переменные a, b, c, то дифференцирование будет выполнено по переменной c. Если при этом в составе аргументов содержится переменная x, то она имеет абсолютный приоритет, независимо от ее положения в алфавите переменных.

```
>> syms   a   b   c   x   w;
>> diff(a + b^2)
ans =
     2 * b
>> diff(a + c * b^3)
ans =
     b^3
>> diff(a * w + c * b^3)
ans =
     a
>> diff(x * a * w + b^3)
ans =
     a * w
```

Функция f может быть вектором и матрицей. В таких случаях откликом будет также вектор или матрица, элементами которой будут производные от исходных функций, образующих вектор или матрицу.

```
>> syms  a  x;
>> y = [ x * sin(x) ; x^5 ; exp(a * x) ];
>> diff (y, x)
ans =
      cos (x) * x + sin (x)
      5 * x^4
      a * exp (a * x)
```

Функция **polyder** предназначена для вычисления производной не только от полинома, но и от произведения и частного двух полиномов. Вызов **polyder** с одним аргументом – вектором, соответствующим полиному (члены в полиноме расположены в порядке убывания их степеней), приводит к вычислению вектора коэффициентов производной полинома:

```
>> p = [1 0 1 0 0 1];
>> p1 = polyder(p)
p1 =
     5     0     3     0     0
```

Для вычисления производной от произведения полиномов следует использовать polyder с двумя входными аргументами:

```
>> p = [1 0 1 0 0 1];
>> q = [1 2 3];
>> pq1 = polyder(p, q)
pq1 =
     7    12    20     8     9     2     2
```

Если необходимо найти производную отношения двух полиномов в виде дроби, числитель и знаменатель которой так же являются полиномами, то следует вызвать polyder с двумя выходными аргументами:

```
>> [n, d] = polyder(p, q)
n =
     3     8    16     4     9    -2    -2
d =
     1     4    10    12     9
```

Первый аргумент n результата содержит коэффициенты числителя, а второй d — знаменателя получающегося отношения полиномов.

Для вычисления численного значения производной в некоторой точке воспользуемся функцией, которая позволяет произвести подстановку одного выражения в другое:

subs (<fun>, <x>, <knot>).

В общем виде функция **subs** вызывается с тремя параметрами: первый – символьная функция или её имя, второй – переменная, подлежащая замене, третий – значение или выражение, которое нужно подставить вместо переменной.

```
>> syms x
>> f=sym('(3*x^2-7)/(2*x+1) ');
>>F=diff(f,x,1)     %Первая производная от f
F=
6*x/(2*x+1)-2*(3*x^2-7)/(2*x+1) ^2
>>subs (F,x,2)   %Значение первой производной в точке 2
ans=
2
```

```
>>subs(F,x,0)    %Значение первой производной в точке 0
ans=
14
>>subs(F,x,0-2)    %Значение первой производной в точке -2
ans=
2,8889
```

Для численного дифференцирования табличной функции двух переменных z(x,y) в MATLAB имеется функция, вычисляющая цифровой аналог градиента:

[gx,gy]=gradient(z,hx,hy)

Здесь:

- `z` — матрица значений дифференцируемой функции, в которой каждый столбец соответствует некоторому значению `x`, а каждая строка — некоторому значению `y`;
- `hx`, `hy` — шаги дифференцирования по `x` и по `y` (можно задать один параметр `h`, который будет использоваться для обоих аргументов, или не задавать его вовсе — тогда по умолчанию берется `h=1`);
- `gx`, `gy` — матрицы, содержащие компоненты градиента, как функции `x` и `y`. Размеры этих матриц такие же, как у матрицы `z`.

Для графического отображения градиента можно использовать функцию `quiver` (от англ. *quiver* — "колчан"), которая рисует векторы градиента в виде стрелок. Простейшая форма обращения к ней — `quiver(gx,gy)`.

Оператор Лапласа для функции `z(x,y)` — сумма вторых производных:

$$\Delta z = \partial^2 z/\partial x^2 + \partial^2 z/\partial y^2.$$

Для его вычисления в MATLAB применяется функция `del2`:

```
L = del2(z,hx,hy);
```

Здесь:

- `z` — матрица значений дифференцируемой функции, в которой каждый столбец соответствует некоторому значению `x`, а каждая строка — некоторому значению `y`;
- `hx`, `hy` — шаги дифференцирования по `x` и по `y` (можно задать один параметр `h`, который будет использоваться для обоих аргументов, или не задавать его вовсе — тогда по умолчанию берется `h=1`);
- `L` — матрица, содержащая значения $\Delta z/4$, как функции `x` и `y`. Размеры этой матрицы такие же, как у матрицы `z`.

При `hx=hy=h` цифровой аналог оператора Лапласа вычисляется по формуле

$$L(x,y)=((z(x-h,y)+(z(x+h,y)+z(x,y-h)+z(x,y+h))/4-z(x,y))/h^2$$

Задачи.

1. Найти 1-ую и 2-ую производные функции, заданной таблично, в точке x= -1,5. Оценить их погрешность. (Это пример, когда точка не совпадает с узлом, а находится между ними. Нужно построить интерполяционный полином , найти его производные и вычислить.)

x_i	-3	-2	-1	0	1	2	3
y_i	-0.71	-0.01	0.51	0.82	0.88	0.51	0.49

2. Это пример, когда точка совпадает с узлом. Нужно использовать конечно-разностные формулы вычисления производных по соседним узлам. Найти 1-ую и 2-ую производные функции y = ln(cos(x)) в точке x= 0,5 различными методами, а именно рассмотреть формулы простые и многоточечные. Оценить точность аппроксимации (это разница между значением производной, вычисленным по точной формуле, полученной аналитически, и её значением, вычисленным по конечно-разностным формулам), см. программы Data_sheet1.m - Data_sheet4.m.

Исследовать влияние величины шага на точность вычисления производных по различным формулам.