👀

Численные методы

1) Источники и классификация погрешностей. Абсолютная и относительная погрешности. Значащие и верные цифры в записи чисел. Погрешности арифметических операций. Погрешности функций.

Источники:

- Неточное условие математической задачи
- Неточный метод решения задачи

Классификация погрешностей:

- Погрешность данных
- Погрешность математической модели
- Погрешность метода
- Вычислительная погрешность

Абсолютная погрешность - это модуль разницы между приближенным значением и точным.

$$\triangle = |a - a_p|$$

Относительная погрешность - это частное модуля разницы приближенного значения и точного и модуля точного значения.

$$\delta = \frac{|a - a_p|}{|a|} = \frac{\triangle}{|a|}$$

$$\triangle = \delta|a|$$

Значащие цифры:

- Все ненулевые цифры
- Нулевые цифры, стоящие между ненулевыми
- Нулевые цифры, сохраняющие разрядность при округлении

n значащих цифр **верны в узком смысле**, когда абсолютная погрешность не превышает половины единиц разряда, соответствующей n-й значащей цифре.

n значащих цифр **верны в широком смысле**, когда абсолютная погрешность не превосходит единицы разряда, соответствующей n-й значащей цифре.

Погрешность арифметических операций

- Сложение:

$$\triangle_u = \triangle_x + \triangle_y$$

$$\delta_u \leq max(\delta_{x1}, \delta_{x2}, ..., \delta_{xn})$$

- Вычитание:

$$\delta_u = \frac{\triangle_x + \triangle_y}{|x - y|}$$

- Умножение:

$$\delta_u = \delta_x + \delta_y$$

Если k - const:

$$\triangle_u = |k|\triangle_x$$

$$\delta_u = \delta_x$$

2) Прямая и обратная задачи теории погрешностей. Методы решения обратных задач теории погрешностей.

Задачу вычислить погрешность функции в случае, когда заданы погрешности аргументов называют **прямой задачей теории погрешностей**.

Задача определения допустимой погрешности аргументов по заданной допустимой погрешности функции называется **обратной задачей теории погрешностей**.

Вычисление допустимых абсолютных погрешностей:

$$\triangle\widetilde{y} \approx |f'(\widetilde{x})|\triangle\widetilde{x}$$
$$\triangle\widetilde{x} \approx \frac{\triangle\widetilde{y}}{|f'(\widetilde{x})|}$$

Для функций нескольких переменных обратная задача не имеет общего алгоритма решений, решается только при некоторых ограничениях. Например, если значения всех аргументов можно определить с любой точность, то применяют **принцип равных влияний**. В таком случае считают, что все слагаемые равны между собой, тогда допустимая абсолютная погрешность:

$$\triangle\widetilde{x}_i = \frac{\triangle\widetilde{y}}{|\frac{\delta f(\widetilde{x}_1, ..., \widetilde{x}_n)}{\delta x_i}|}$$

3) Метод Гаусса. Решение систем линейных алгебраических уравнений и вычисление определителей матриц методом Гаусса.

Рассмотрим систему из трех уравнений с тремя неизвестными:

$$\begin{aligned} &\mathrm{I}: && a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ &\mathrm{II}: && a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ &\mathrm{III}: && a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{aligned} \tag{3.2}$$

Система уравнений (3.2) приводится к эквивалентной системе с треугольной матрицей:

$$\begin{aligned} &\mathrm{I}: && a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ &\mathrm{II}': && a'_{22}x_2 + a'_{23}x_3 = b'_2 \\ &\mathrm{III}'': && a''_{33}x_3 = b''_3 \end{aligned} \tag{3.3}$$

Достигается это при помощи цепочки элементарных преобразований, при которых из каждой строки вычитаются некоторые кратные величины расположенных выше строк.

Процесс приведения системы (3.2) к системе (3.3) называется прямым ходом, а нахождение неизвестных x_1, x_2, x_3 из системы (3.3) называется обратным ходом.

Чтобы посчитать определитель методом Гаусса, аналогично приводим СЛАУ к треугольному виду и умножаем все элементы на главной диагонали.

4) Решение систем линейных алгебраических уравнений методом прогонки.

Начальный вид СЛАУ (трехдиагональная матрица):

$$a_i x_{x-1} + b_i x_x + c_i x_{x+1} = d_i$$

$$a_1 = 0, c_n = 0$$

Сначала находим U и V (прямой ход):

$$U_i = -\frac{c_i}{a_i U_{i-1} + b_i}$$

$$V_i = \frac{d_i - a_i V_{i-1}}{a_i U_{i-1} + b_i}$$

Соответственно:

$$U_1 = 0, V_n = x_n$$

$$V_1 = \frac{d_i}{b_i}$$

И по формуле находим остальные x (обратный ход):

$$x_i = U_i x_{i+1} + V_i$$

Невязки:

$$r_i = d_i - a_i x_{i-1} - b_i x_i - c_i x_{i+1}$$

5) Решение систем линейных алгебраических уравнений методом ведущего элемента.

Модифицированный метод Гаусса. Выбираем ведущий элемент. Элементы, которые находятся на строке и столбце - рабочие. (Пусть k - ведущая строка, l - ведущий столбец). В некоторых примерах ведущий элемент - наибольший по модулю.
1) Делим на ведущий элемент столбец, где он расположен (l). Получаем некие **коэффициенты (μ), соответствующие каждой строке**. В матрице оставшиеся элементы кроме рабочей строки зануляются.

$$\begin{pmatrix} 0 \\ ... \\ 0 \\ a_{kl} \\ 0 \\ ... \\ 0 \end{pmatrix}$$

2) **Далее рабочие строка и столбец не участвуют в вычислениях!** Рабочий столбец остается зануленным. Рабочая строка - не изменяется. Оставшаяся матрица пересчитывается как:

$$(a_{ij} - \mu a_{kj}) x_j$$

p.s. для каждой строки будет свой коэффициент μ, a_{kj} - элемент ведущей строки, a_{ij} - элемент того же столбца и строки, которую расчитываем.

3) Так проводим вычисления, пока не останется в последнем уравнении 1 неизвестная.

4) Обратный ход аналогичен методу Гаусса.

7) Решение систем линейных алгебраических уравнений методом простых итераций. Сходимость метода и оценка погрешности.

Расчет начинаем с начального приближения x^(0). Вычислительный процесс приводит к новому вектору:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)})$$

Условие сходимости: преобладание диагональных элементов.

$$\sum_{j \neq j} |a_{ij}| < a_{ii}$$

Оценка погрешности:

$$max x_i |x_i^{(k+1)} - x_i^{(k)}| < \varepsilon$$

8) Решение систем линейных алгебраических уравнений методом Зейделя. Сходимость метода и оценка погрешности.

Аналогично методу Якоби, только в методе Зейделя мы используем уже найденные компоненты с меньшими номерами:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)})$$

Условие сходимости: преобладание диагональных элементов.

$$\sum_{j \neq j} |a_{ij}| < a_{ii}$$

Оценка погрешности:

$$max x_i |x_i^{(k+1)} - x_i^{(k)}| < \varepsilon$$

9) Решение нелинейных уравнений. Метод половинного деления. Сходимость метода и оценка погрешности.

Определяем интервал, где находится приближенное решение. Обычно графически. **Условие сходимости:**

$$F(a)F(b) < 0$$

То есть, в данном интервали значения функции имеют разные знаки, соответственно внутри интервала есть решение.

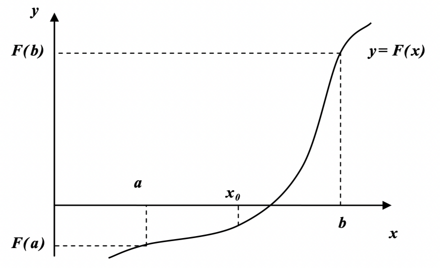
Начальное приближение:

$$x^0 = \frac{a+b}{2}$$

Анализируем f(a), f(b), f(x^0) и выбираем новый интервал, где знаки отличаются. В качемтве 1-й итерации корня принимаем середину нового отрезка.

Оценка погрешности:

$$|b - a| < \varepsilon$$



10) Решение нелинейных уравнений. Метод простой итерации. Сходимость метода и оценка погрешности.

Для этого метода необходимо привести F(x) = 0 к виду x = phi(x).

$$\varphi(x) = x - \frac{F(x)}{M}$$

M - постоянная неизвестная величина, определяющаяся из **условий сходимости метода**:

$$0 < |\varphi'(x)| < 1$$

$$|1 - \frac{F'(x_0)}{M}| < 1$$

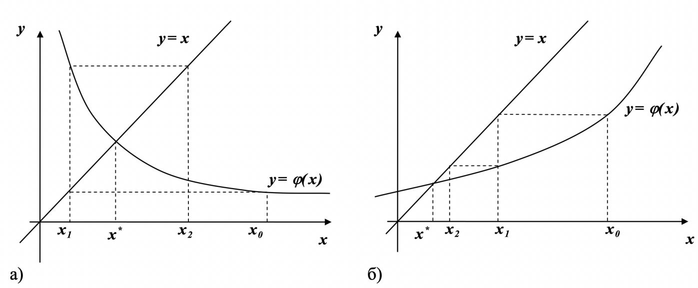
Или:

$$M = 1,01 * F'(x_0)$$

$$x_{k+1} = \varphi(x_k) = x_k - \frac{F(x_k)}{M}$$

Оценка погрешности:

$$|x_{k+1} - x_k| < \varepsilon$$



а):

$$-1 < \varphi'(x) < 0$$

б):

$$0 < \varphi'(x) < 1$$

11) Решение нелинейных уравнений. Метод Ньютона. Сходимость метода и оценка погрешности.

Метод основан на построении касательных к y = F(x) и ищется точка пересечения касательной с осью абсцисс. За начальное приближение x0 принимается тот конец отрезка [a, b], на котором:

$$F(x_0)F''(x_0) > 0$$

Уравнение касательной в точке M0:

$$y - F(x_0) = F'(x_0)(x - x_0)$$

При x = x1, y = y1 = 0:

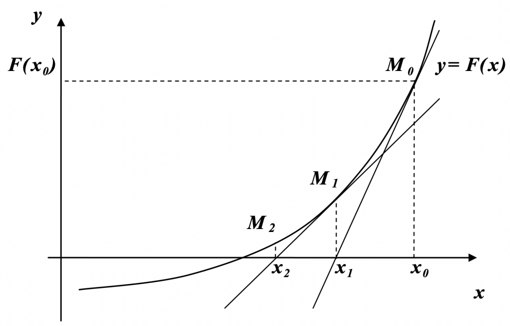
$$x_1 = x_0 - \frac{F(x_0)}{F'(x_0)}$$

Соответственно:

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)}$$

Оценка погрешности:

$$|x_{k+1} - x_k| < \varepsilon$$

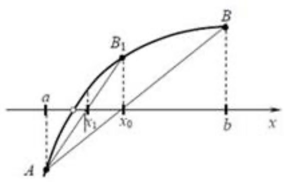


12) Решение нелинейных уравнений. Метод хорд. Сходимость метода и оценка погрешности.

Условие сходимости:

$$F(a)F(b) < 0$$

В отличие от метода половинного деления, метод хорд предлагает, что деление рассматриваемого интервала будет выполняться не в его середине, а в точке пересечения хорды с осью абсцисс (ось - X).



Уравнение прямой (хорды), которая проходит через точки A и B имеет следующий вид:

$$\frac{x - a}{b - a} = \frac{y - f(a)}{f(b) - f(a)}$$

Тогда при x = x_0 и y = 0, точка пересечения хорды и оси абсцисс будет:

$$x_0 = \frac{-f(a)(b - a)}{f(b) - f(a)} + a$$

За новый интервал аналогично методу половинного деления принимаем тот, на концах которого значения функции принимают разные знаки:

$$f(a)f(b) < 0$$

Оценка погрешности:

$$|x_{k+1} - x_k| < \varepsilon$$

13) Решение систем нелинейных уравнений. Метод Ньютона на примере системы 2×2. Сходимость метода и оценка погрешности.

СНУ:

F1(x1,...,xn)=0
F2(x1,...,xn)=0
...
Fn(x1,...,xn)=0

Суть метода: выделить из уравнений линейные части, которые являются главными при малых приращениях аргументов.

$$F(x,y) = 0 \qquad G(x,y) = 0$$

При каком-то приближении xk, yk поправки △x_k = x_k+1 - x_k и △y_k = y_k+1 - y_k можно найти через:

$$F(x_k + \triangle x_k, y_k + \triangle y_k) = 0 \qquad G(x_k + \triangle x_k, y_k + \triangle y_k) = 0$$

Раскладываем в ряд Телойра по △x и △y получаем систему:

$$\frac{\delta F(x_k, y_k)}{x_k} \triangle x_k + \frac{\delta F(x_k, y_k)}{y_k} \triangle y_k = -F(x_k, y_k)$$

$$\frac{\delta G(x_k, y_k)}{x_k} \triangle x_k + \frac{\delta G(x_k, y_k)}{y_k} \triangle y_k = -G(x_k, y_k)$$

Решая эту систему, находим △x_k, △y_k.

Т.о. решение системы методом Ньютона состоит в построении итерационной последовательности:

$$x_{k+1} = x_k + \triangle x_k$$

$$y_{k+1} = y_k + \triangle y_k$$

Подведем итог:

Чтобы решить СНУ методом Ньютона, надо сначала найти начальное приближение (x0), найти частные производные, разложить ряд Тейлора, найти △x_0, △y_0, найти x1, y1, далее подставляем в уравнения системы ряда Тейлора x_1, y_1,находим △x_1, △y_1 и считаем x_2, y_2 и так далее.

Оценка погрешности:

$$max_{1 \le j \le n} |x_i^{(k+1)} - x_i^{(k)}| < \varepsilon$$

(то есть сравниваем x и y отдельно)

14) Постановка задач приближения функций. Интерполяционный многочлен. Единственность решения задачи интерполяции.

Аппроксимация - восстановление аналитической функции по отдельным значениям.

Для задачи **интерполяции критерий близости** аппроксимирующей функции к исходным данным x_i, y_i рассматривается как **совпадение значений в заданных точках**, называемых **узлами интерполяции**.

$$\tilde{y}(x_i) = y_i$$

Если функция задана в виде полинома (многочлена), то это **интерполяционный полином** и м.б. записан в форме Лагранжа или Ньютона.

Единственность решения задачи интерполяции:

Пусть дан следующий интерполяционный многочлен:

$$P_n(x) = \sum_{k=0}^{n} a_k x^{n-k}$$

Тогда получаем систему:

$$\begin{cases} a_0x_0^n + a_1x_0^{n-1} + ... + a_n = y_0 \\ a_0x_1^n + a_1x_1^{n-1} + ... + a_n = y_1 \\ ... \\ a_0x_n^n + a_1x_n^{n-1} + ... + a_n = y_n \end{cases}$$

Поскольку все x_i различны, △ отличен от 0, следовательно, система имеет единственное решение.

15) Интерполяционный многочлен Лагранжа.

$$P(x) = L_{n-1}(x) = \sum_{y=1}^{n} y_i(x)l_i(x)$$

$$l_i(x) = \frac{(x-x_1)(x-x_2)...(x-x_{i-1})(x-x_{i+1})...(x-x_{n-1})(x-x_n)}{(x_i-x_1)(x_i-x_2)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_{n-1})(x_i-x_n)}$$

16) Интерполяционный многочлен Ньютона.

$$N_{n-1}(x) = \triangle^0(x_1) + \triangle^1(x_1,x_2)(x-x_1) + \triangle^2(x_1,x_2,x_3)(x-x_1)(x-x_2) + ... + \triangle^{n-1}(x_1,x_2,...,x_n)(x-x_1)(x$$

$$\triangle^0(x_i) = y_i$$
$$\triangle^1(x_i,x_k) = \frac{\triangle^0(x_i) - \triangle^0(x_k)}{x_i - x_k}$$
$$\triangle^2(x_i,x_k,x_j) = \frac{\triangle^1(x_i,x_k) - \triangle^1(x_k,x_j)}{x_i - x_j}$$

17) Метод наименьших квадратов. Оценка погрешности метода.

Необходимо минимизировать сумму квадратов:

$$S = \sum_{i=1}^{n} (\tilde{y}(x_i) - y_i)^2 \to min$$

Линейная аппроксимация (~y(x) = ax + b):

Минимум в точка, когда частные проивзодные равны 0:

$$\frac{\delta S(a,b)}{\delta a} = 0 \qquad \frac{\delta S(a,b)}{\delta b} = 0$$

$$\sum_{i=1}^{n} 2(ax_i + b - y_i)x_i = 0 \qquad \sum_{i=1}^{n} 2(ax_i + b - y_i) = 0$$

На 2 сокращаем и выносим слева переменные a и b:

$$\begin{cases} a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_iy_i \\ a\sum_{i=1}^{n} x_i + bn = \sum_{i=1}^{n} y_i \end{cases}$$

Находим a и b.

Полиномиальная аппроксимация (~y(x) = ax^2 + bx + c):

Минимум в точка, когда частные проивзодные равны 0:

$$\frac{\delta S(a,b,c)}{\delta a}=0 \qquad \frac{\delta S(a,b,c)}{\delta b}=0 \qquad \frac{\delta S(a,b,c)}{\delta c}=0$$

$$\sum_{i=1}^{n} 2(ax_i^2+bx_i+c-y_i)x_i^2=0 \qquad \sum_{i=1}^{n} 2(ax_i^2+bx_i+c-y_i)x_i=0 \qquad \sum_{i=1}^{n} 2(ax_i^2+bx_i+c-y_i)=0$$

На 2 сокращаем и выносим слева переменные a, b и c:

$$\begin{cases} a\sum_{i=1}^n x_i^4 + b\sum_{i=1}^n x_i^3 + c\sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i^2 y_i \\ a\sum_{i=1}^n x_i^3 + b\sum_{i=1}^n x_i^2 + c\sum_{i=1}^n x_i = \sum_{i=1}^n x_i y_i \\ a\sum_{i=1}^n x_i^2 + b\sum_{i=1}^n x_i + cn = \sum_{i=1}^n y_i \end{cases}$$

Находим a,b и c.

Оценка погрешности:

$$s=\sqrt{\frac{\sum_{i=1}^n (\tilde{y}(x_i)-y_i)^2}{n}}$$

18) Проблемы интерполяции функции двух переменных.

1. Не любое число узлов интерполяции выгодно. Если для одной переменной степень многочлена была взаимно однозначно связана с числом узлов, то для двух переменных многочлен *n*-ой степени

$$P_n(x,y)=\sum_{k+m=0}^{n} a_{km}x^k y^m$$

имеет (n+1)(n+2)/2 узлов.

2. Не всякое расположение узлов допустимо: в одномерном случае узлы не должны были совпадать. Теперь же для интерполяции многочленом P1(x,y) необходимо, чтобы узлы не лежали на прямой в плоскости (x, y) . При интерполяции многочленом Pn(x,y) требуется, чтобы узлы не лежали на кривой n-го порядка.

Поэтому для хорошей интерполяции сетка должна быть регулярно построенной, а не представлять собой совокупность беспорядочно расположенных точек.

19) Построение интерполяционного многочлена для функции двух переменных.

рассмотрим специальный случай двумерной интерполяции.

1. Возьмем $\frac{(n+1)(n+2)}{2}$ узлов и расположим их следующим образом

$$
\begin{array}{lllll}
(x_0, y_0), & (x_1, y_0), & \ldots, & (x_{n-1}, y_0), & (x_n, y_0), \\
(x_0, y_1), & (x_1, y_1), & \ldots, & (x_{n-1}, y_1), & \\
. & . & . & . & . \\
(x_0, y_{n-1}), & (x_1, y_{n-1}), & & & \\
(x_0, y_n), & & & &
\end{array}
$$

2. Проверим, что нет кривой n-го порядка, проходящей через все эти узлы.

3. Построим интерполяционный полином $P_n(x, y)$ по нашим узлам.

$P_n(x_i, y_j)$ через z_{ij}, где z_{ij} – соответствующие значения функции в узлах.

Образуем разность $P_n(x, y) - P_{n-1}(x, y) = A_{n,0}(x - x_0)\ldots(x - x_{n-1}) + A_{n-1,1}(x - x_0)\ldots(x - x_{n-2})(y - y_0) + {}$

$+ A_{n-2,2}(x - x_0)\ldots(x - x_{n-3})(y - y_0)(y - y_1) + \ldots + A_{0,n}(y - y_0)(y - y_1)\ldots(y - y_{n-1})$.

В точке (x_i, y_{n-i}) все его члены обратятся в нуль за исключением $A_{i,n-i}(x_i - x_0)\ldots(x_i - x_{i-1})(y_{n-i} - y_0)\ldots(y_{n-i} - y_{n-i-1})$. Таким образом, коэффициенты $A_{i,n-i}$ определяются однозначно. В силу единственности интерполяционного многочлена по выбранным нами узлам это и будет единственным значением разности.

4. Выразим теперь A_{ij} через значения функции $z_{kl} = f(x_k, y_l)$ в узлах сетки. Подставляя в правую часть (x_0, y_0) получим $A_{00} = z_{00} \equiv f(x_0, y_0)$.
В точке (x_1, y_0) $P_n = f(x_1, y_0)$, а правая часть равна $A_{00} + A_{10}(x_1 - x_0)$, следовательно $A_{10} = \frac{f(x_1, y_0) - f(x_0, y_0)}{x_1 - x_0}$.

Это отношение является разделенной разностью функции $f(x, y_0)$ при фиксированном $y = y_0$. Обозначим его через $f(x_0; x_1; y_0)$. Аналогично получаем $A_{01} = f(x_0; y_0; y_1)$. Зафиксируем теперь $y = y_0$, получаем

$$P(x, y_0) = A_{00} + A_{10}(x - x_0) + A_{20}(x - x_0)(x - x_1) + \ldots + A_{n0}(x - x_0)\ldots(x - x_{n-1}).$$

Это интерполяционный полином относительно x, принимающий в точках (x_i, y_0) значения $f(x_i, y_0)$. Следовательно, $A_{i0} = f(x_0; x_1; \ldots x_i; y_0)$.

$$A_{k1} = \frac{f(x_0; x_1; \ldots; x_k; y_1) - f(x_0; x_1; \ldots; x_k; y_0)}{y_1 - y_0}.$$

Это разделенная разность по переменной y, обозначим ее так

$$A_{k1} = f(x_0; x_1; \ldots; x_k; y_0; y_1).$$

Рассуждая как и прежде, найдем

$$A_{k0} + A_{k1}(y_m - y_0) + \ldots + A_{km}(y_m - y_0)\ldots(y_m - y_{m-1}) = f(x_0; x_1; \ldots; x_k; y_m).$$

Рассматривая это выражение, как функцию y_m, получаем

$$A_{ki} = f(x_0; x_1; \ldots; x_k; y_0; y_1; \ldots; y_i).$$

Таким образом, окончательно нашу интерполяционную формулу можем записать в виде

$$P_n(x, y) = \sum_{i=0}^{n} \sum_{j=0}^{n-i} f(x_0; \ldots; x_i; y_0; \ldots; y_j) \cdot \prod_{p=0}^{i-1}(x - x_p) \prod_{q=0}^{j-1}(y - y_q). \qquad (14.30)$$

20) Задача численного дифференцирования. Вычисление производных с помощью интерполяционных многочленов.

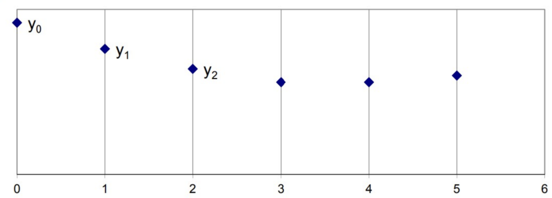
Допустим, что в некоторой точке x у функции f(x) существует производная r-того порядка f^(r)(x), которую точно вычислить либо не удается, либо слишком сложно. В этом случае для приближенного нахождения производных функции используются **формулы численного дифференцирования**.

Задача численного дифференцирования состоит в приближенном вычислении производных функции f(x) по заданным в конечном числе точек значениям этой функции.

Для вычисления производгных с помощью интерполяционных многочленов, по заданным в конечном числе точек значениям функции находим интерполяционный полином (в форме Лагранжа или Ньютона) и вычисляем производные.

21) Вычисление производных с помощью конечных разностей.

Виды конечных разностей:



Левые разности: $\Delta y_1 = y_1 - y_0, \quad \Delta x = h, \quad y_1' \approx \dfrac{y_1 - y_0}{h}$

Правые разности: $\Delta y_1 = y_2 - y_1, \quad \Delta x = h, \quad y_1' \approx \dfrac{y_2 - y_1}{h}$

Центральные разности: $\Delta y_1 = y_2 - y_0, \quad \Delta x = 2h, \quad y_1' \approx \dfrac{y_2 - y_0}{2h}$

Вторая производная

$$y_1'' = (y_1')' \approx \frac{y_2' - y_1'}{h} \approx \frac{(y_2 - y_1)/h - (y_1 - y_0)/h}{h} = \frac{y_2 - 2y_1 + y_0}{h^2}$$

При вычислении второй производной также можно использовать левые, правые, или центральные разности

22) Интерполирование сплайнами. Построение кубического сплайна.

Сплайн интерполяция предполагает представление интерполирующей функции в виде **комбинации разных функций**, соответствующих отрезкам между соседними узлами.

Сплайн интерполяция функции y=y(x) определяет **набор функций сплайнов** f_i(x), аппроксимирующих y(x) на интервалах x_{i-1} ≤ x < x_i:

$$y(x) = \begin{cases} f_1(x) & x_0 \le x < x_1 \\ f_2(x) & x_1 \le x < x_2 \\ ... & ... \\ f_n(x) & x_{n-1} \le x < x_n \end{cases}$$

Кубический сплайн:

$$f_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$
$$h_i = x_i - x_{i-1}$$

Можно сразу найти a_i:

$$a_i = y_{i-1}$$

Далее составляем систему:

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}})$$
$$c_1 = 0$$

Решаем методом прогонки, получаем коэффициенты c_i.

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i) \qquad b_n = \frac{y_n - y_{n-1}}{h_n} - \frac{2}{3}h_n c_n$$

$$d_i = \frac{c_{i+1} - ci}{3h_i} \qquad d_n = -\frac{c_n}{3h_n}$$

23) Численное интегрирование с помощью формулы прямоугольников.

Численное интегрирование:

$$I = \int_a^b P_m(x)dx$$

P_m(x) - полином по значениям узлов, на которые мы поделим f(x).

$$I = \sum_{i=0}^{n-1} f(x_i)(x_{i+1} - x_i)$$

Для равноотстающих узлов:

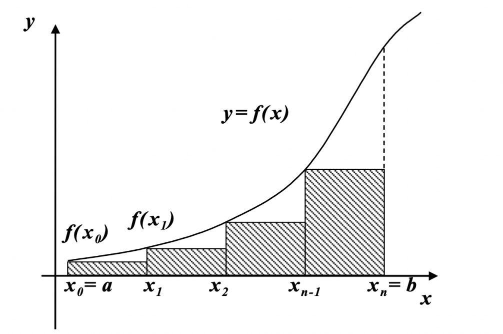
$$I = h\sum_{i=0}^{n-1} f(x_i)$$

— ф-ла левых прямоугольников

$$I = h\sum_{i=1}^{n} f(x_i)$$

— ф-ла правых прямоугольников

$$h_{i+1} = x_{i+1} - x_i$$



Формула средних прямоугольников:

$$I = h\sum_{i=0}^{n} f(x_{i+0,5})$$

$$x_{i+0,5} = x_i + \frac{h}{2}$$

24) Численное интегрирование с помощью формулы трапеций.

В данном методе полином 1-й степени. По формуле Лагранжа:

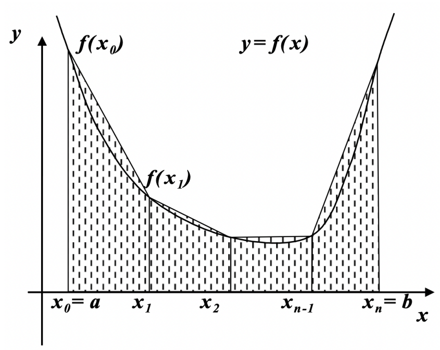
$$P_1(x) = f(x_i)\frac{x - x_{i+1}}{x_i - x_{i+1}} + f(x_{i+1})\frac{x - x_i}{x_{i+1} - x_i}$$

Соответственно:

$$I = \frac{1}{2}\sum_{i=0}^{n-1}(f(x_i) + f(x_{i+1}))(x_{i+1} - x_i)$$

Для равноотстающих узлов:

$$I = \frac{h}{2}\sum_{i=0}^{n-1}(f(x_i) + f(x_{i+1}))$$



25) Численное интегрирование с помощью формулы Симпсона.

Интервал [a, b] делим на 2n частей и группируем узлы x_{i-1}, x_i, x_{i+1}. Используем полином 2-й степени:

$$P_2(x) = f(x_{i-1})\frac{(x - x_i)(x - x_{i+1})}{(x_{i-1} - x_i)(x_{i-1} - x_{i+1})} + f(x_i)\frac{(x - x_{i-1})(x - x_{i+1})}{(x_i - x_{i-1})(x_i - x_{i+1})} + f(x_{i+1})\frac{(x - x_{i-1})(x - x_i)}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)}$$

Соответственно:

$$I = \frac{h}{3}\sum_{k=0}^{n-1}[f(x_{2k}) + 4f(x_{2k+1}) + f(x_{2k+2})]$$

Или:

$$I = \frac{h}{3}(f(a) + 4f(x_1) + 2f(x_2) + 4f(x_3) + ... + 4f(x_{n-1}) + f(b))$$

26) Интерполяционная квадратурная формула Лагранжа.

Вычислим интеграл $\int_a^b f(x)dx$, заменяя подынтегральную функцию интерполяционным многочленом с узлами $x_k = x_0 + kh,\ k = \overline{0,n}$, где $x_0 = a,\ h = \frac{b-a}{n}$. Получим

$$\int_a^b f(x)dx \approx \int_a^b \sum_{k=0}^{n} l_k(x) y_k = \sum_{k=0}^{n} y_k \int_a^b l_k(x)dx = \sum_{k=0}^{n} y_k A_k ,$$

где $A_k = \int_a^b \frac{\omega_k(x)}{\omega_k(x_k)} dx$

и $\omega_k(x) = (x - x_0)\cdot \ldots \cdot (x - x_{k-1})\cdot(x - x_{k+1})\cdot \ldots \cdot (x - x_n)$.

Формулу

$$\int_a^b f(x)dx = \sum_{k=0}^{n} y_k A_k \qquad (.2)$$

называют интерполяционной квадратурной формулой Лагранжа.

27) Численное решение задачи Коши для ОДУ 1-го порядка методом Эйлера.

Метод Эйлера

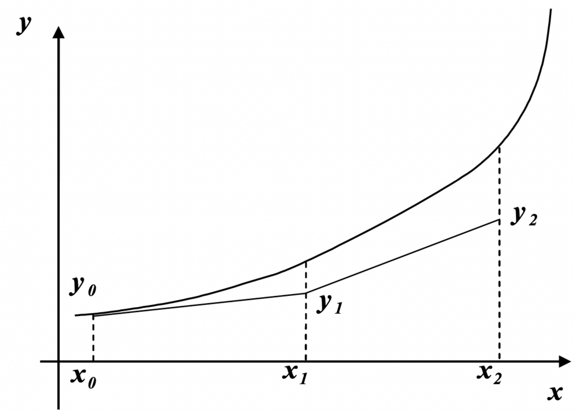
Необходимо посчитать производную:

$$y' = f(x, y)$$

$$y'(x_i) = \frac{\triangle y}{\triangle x} = \frac{y(x_{i+1}) - y(x_i)}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{h}$$
$$y'(x_i) = f(x_i, y_i)$$
$$y_{i+1} = hf(x_i, y_i) + y_i$$

— формула Эйлера.

Соответственно, берем начальное условие (x_0, y_0) и с шагом считаем x_i и подставляем.



Модифицированный метод Эйлера

Сначала ищем приближенное значение:

$$\tilde{y}_{i+1} = y_i + hf(x_i, y_i)$$

Потом точное:

$$y_{i+1} = y_i + \frac{h}{2}(f(x_i, y_i) + f(x_{i+1}, \tilde{y}_{i+1}))$$

Метод Рунге-Кутта

Наиболее употребительная схема 4-го порядка:

$$\begin{gathered}
y_{i+1} = y_i + (k_0 + 2k_1 + 2k_2 + k_1)/6 \\
k_0 = hf(x_i, y_i) \\
k_1 = hf(x_i + h/2, y_i + k_0/2) \\
k_2 = hf(x_i + h/2, y_i + k_1/2) \\
k_3 = hf(x_i + h, y_i + k_2)
\end{gathered}$$