Численные методы интегрирования

Баев А.Ж.

28 марта 2016 г.

Содержание

1 Вычисление интеграла

Постановка задачи. Дана непрерывная на отрезке $[a;b]$ функция $f(x)$. Необходимо вычислить:

$$I = \int_a^b f(x)dx.$$

1.1 Детерменированные методы решения

Основная идея: реализовать суммы Дарбу при равномерном разбиении.

$$I = \int_a^b f(x)dx \approx \sum f(\xi_i)\Delta_i.$$

Подробно методы можно посмотреть в [1, стр. 86] и [2, стр. 72]. Для детерминированных методов обычно вводится равномерная сетка на отрезке $[a;b]$. Для этого выбирается число разбиений n и вычисляется шаг разбиений $h = \frac{1}{n}$. Далее определяются узлы квадратурной формулы:

$$x_i = a + i * h, i = 0..n$$

Далее определются значения функции в этих узлах:

$$f_i = f(x_i), i = 0..n$$

Взвешенная сумма данных значений позволяет приблизить интеграл. Подбирая различные веса, можно получить различные приближения интеграла.

Пример. Например, вычислим интеграл:

$$I = \int_1^3 \big(1 + (x-1)(x-2)(x-3)(x-3)\big)dx.$$

Легко посчитать его аналитически:

$$I = \int_1^3 \big(1 + (x-1)(x-2)(x-3)(x-3)\big)dx =$$
$$= \int_{-1}^1 \big(1 + (t+1)t(t-1)(t-1)\big)dx =$$
$$= \int_{-1}^1 \big(t^4 - t^3 - t^2 + t + 1\big)dx = \frac{26}{15}$$

Введем сетку Ω на отрезке $[1;3]$ при $n = 4$. Тогда шаг сетки равен $h = \frac{1}{2}$, а сама сетка: $\Omega = \{x_i = 1 + 0.5 * i, i = 0, 1, 2, 3, 4\}$. Значения функции на сетке, через которые будем приближать интеграл:

$$\begin{aligned}
f_0 =& f(1) = 1\\
f_1 =& f(1.5) = \frac{7}{16}\\
f_2 =& f(2) = 1\\
f_3 =& f(2.5) = \frac{19}{16}\\
f_3 =& f(3) = 1
\end{aligned}$$

1.1.1 Метод левых прямоугольников

Основная идея: на каждом отрезке кусочно-постоянная аппроксимация. В итоге, находим сумму площадей прямоугольников.

$$I_n = \sum_{i=0}^{n-1} f_i h = h \sum_{i=0}^{n-1} f_i.$$

Погрешность: $|I - I_n| = O(h) = O\left(\frac{1}{N}\right)$.

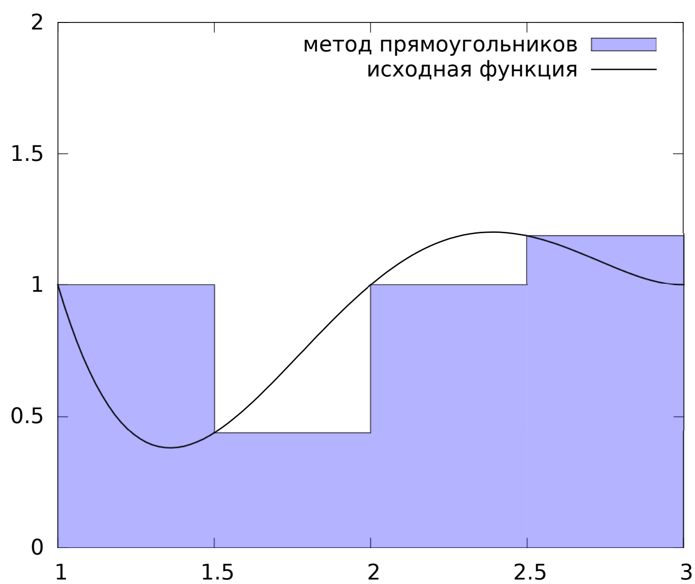

Рисунок 1: Метод прямоугольников

Интеграл для примера:

$$I_4 = \frac{1}{2}\left(1 + \frac{7}{16} + 1 + \frac{19}{16}\right) = \frac{29}{16}.$$

Погрешность:

$$|I_4 - I| = \left|\frac{29}{16} - \frac{26}{15}\right| = \frac{17}{120} \approx 0.1417.$$

1.1.2 Метод трапеций

Основная идея: на каждом отрезке кусочно-линейная аппроксимация. В итоге, находим сумму площадей прямоугольных трапеций.

$$\overline{I}_n = \sum_{i=0}^{n-1} \frac{f_i + f_{i+1}}{2} h = h\left(\frac{f_0}{2} + \sum_{i=1}^{n-1} f_i + \frac{f_n}{2}\right)$$

Погрешность: $|I - \overline{I}_n| = O(h^2) = O\left(\frac{1}{N^2}\right)$.

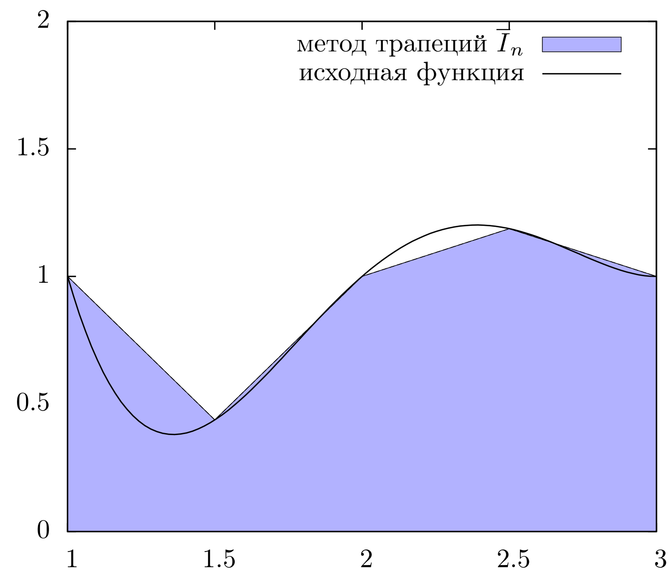

Рисунок 2: Метод трапеций

Интеграл для примера:

$$\overline{I}_4 = \frac{1}{2}\left(\frac{1}{2} + \frac{7}{16} + 1 + \frac{19}{16} + \frac{1}{2}\right) = \frac{29}{16}.$$

Стоит заметить, что данный пример показывает совсем небольшое различие между методом трапеций и методом прямоугольников в случае если функция принимает одинаковые значения на концах отрезка интегрирования.

Погрешность:

$$|\bar{I}_4 - I| = \left|\frac{29}{16} - \frac{26}{15}\right| = \frac{17}{120} \approx 0.1417.$$

1.1.3 Метод Симсона

Основная идея: на каждом отрезке кусочно-квадратичная аппроксимация (применим при $n = 2m$). В итоге, находим сумму площадей параболических трапеций. Для этого разумно уточнить, что площадь под параболой можно вычислить через значения в 3 подряд идущих точках сетки:

$$\int_a^b x^2 dx = \frac{1}{3}(b^3 - a^3) = \frac{b-a}{3}(b^2 + ba + a^2) = \frac{b-a}{6}\left(b^2 + (b+a)^2 + a^2\right).$$

Пусть $a = x_{2k}$, $b = x_{2k+2}$. Тогда $a + b = 2x_{2k+1}$ и $b - a = 2h$:

$$\int_{x_{2k}}^{x_{2k+2}} x^3 dx = \frac{h}{3}\left(f(x_{2k}) + 4f(x_{2k+1}) + f(x_{2k+2})\right).$$

Проведя m парабол через тройки подряд идущих узлов сетки получим формул Симсона для вычисления интеграла:

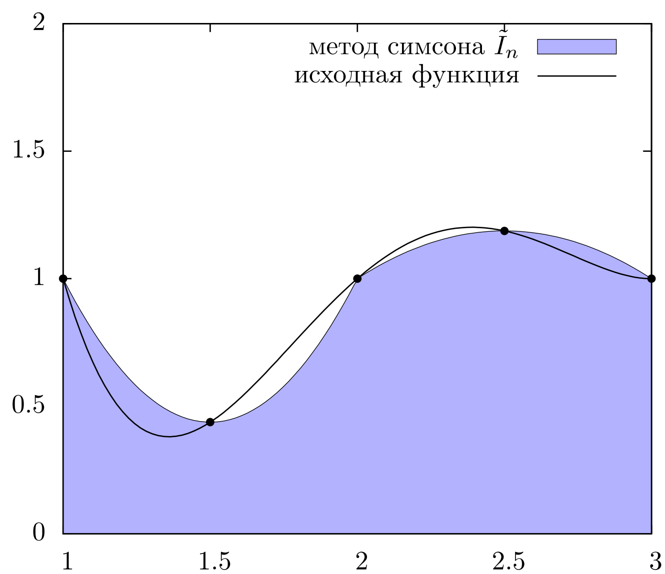

Рисунок 3: Метод Симсона

$$\tilde{I}_n = \sum_{i=0}^{m-1} \frac{f_{2i} + 4f_{2i+1} + f_{2i+2}}{3} h =$$
$$= h\left(\frac{f_0}{3} + \frac{4}{3}\sum_{i=1}^{m} f_{2i-1} + \frac{2}{3}\sum_{i=1}^{m} f_{2i} + \frac{f_n}{3}\right) =$$
$$= \frac{h}{3}\left(f_0 + 4f_1 + 2f_2 + 4f_3 + ... + 2f_{2n-2} + 4f_{2n-1} + f_{2n}\right).$$

Погрешность: $|I - \tilde{I}_n| = O(h^3) = O\left(\frac{1}{N^3}\right)$.

Интеграл для примера:

$$\tilde{I}_4 = \frac{1}{6}\left(1 * 1 + 4 * \frac{7}{16} + 2 * 1 + 4 * \frac{19}{16} + 1 * 1\right) = \frac{7}{4}.$$

Стоит заметить, что данный пример показывает совсем небольшое различие между методом трапеций и методом прямоугольников в случае если функция принимает одинаковые значения на концах отрезка интегрирования.

Погрешность:

$$|\tilde{I}_4 - I| = \left|\frac{7}{4} - \frac{26}{15}\right| = \frac{1}{60} \approx 0.0167.$$

1.2 Вероятностные методы решения

Основное отличие данных методов в том, что для реализации необходим генератор равномерно-распределенных случайных чисел. Такой класс методов часто называется методами Монте-Карло. Подробно можно прочитать про такие методы в [1, стр. 117].

1.2.1 Алгебраическое вероятностное приближение

Основная идея основана на суммах Дарбу:

$$I = \int_a^b f(x)dx \approx \sum f(\xi_i)\Delta_i.$$

Но реализовывать мы ее будем иначе: ξ_i — случайная равномерно распределенная на отрезке $[a; b]$ величина. Ее можно сгенерировать из равномерное распределенной на отрезке $[0; 1]$ случайной величины u_i:

$$\xi_i = a + u_i * (b - a).$$

Интеграл будет аппроксимироваться так:

$$\hat{I}_n = \frac{b-a}{N}\sum_{i=1}^{N} f(\xi_i).$$

Погрешность: $|I - \tilde{I}_n| = O\left(\frac{1}{\sqrt{N}}\right)$.

Для примера сгенерируем 20 случайных значений: 1.7888, 2.5969, 1.3951, 2.5365, 2.1079, 2.2577, 2.0268, 2.8324, 2.4346, 2.2139, 1.4858, 2.6084, 1.8019, 1.2176, 1.4365, 2.6782, 1.5921, 2.0486, 2.9456, 2.5427. Легко проверить, что:

$$\hat{I}_{20} = 1.824732.$$

Погрешность:

$$|\hat{I}_{20} - I| = \left|1.824732 - \frac{26}{15}\right| \approx 0.0914.$$

1.2.2 Геометрическое вероятностное приближение

Рассмотрим прямоугольник:

$$[a;b] \times [m, M],$$

где

$$m \le \min_{[a;b]} f(x)$$

$$M \ge \max_{[a;b]} f(x)$$

Площадь прямоугольника обозначим:

$$S = (b-a)(M-m)$$

.

Сгенерируем n случайных равномерно распределенных по каждой координате точек (x_i, y_i) (т.е. x_i равномерно распределенная по $[a;b]$, y_i равномерно распределенная по $[m;M]$). Посчитаем n_0 — количество точек, которые попали под график функции $f(x)$, т.е. $f(x_i) < y_i$. Причем, если точка y попадает ниже оси абсцисс, то считаем данную точку со знаком минус (то есть уменьшаем счетчик n_0 на один). Тогда верно, что:

$$\frac{n_0}{n} \approx \frac{I}{S}.$$

Значит:

$$I_n = \frac{n_0}{n} * S.$$

Погрешность: $|I - \tilde{I}_n| = O\left(\frac{1}{\sqrt{N}}\right)$. Правда стоит отметить, что старший коэффициент в данном методе будет меньше, чем в алгебраическом при выборе минимально возможного прямоугольника.

Для примера: $m = 0.0$, $M = 1.5$, $S = (b-a)*(M-m) = 2*1.5 = 3$, $n = 20$, $n_0 = 14$. Тогда:

$$\tilde{I}_{20} \approx \frac{14}{20} * 3 = \frac{21}{10}$$

Погрешность:

$$|\tilde{I}_{20} - I| = \left|\frac{21}{10} - \frac{26}{15}\right| = \frac{11}{30} \approx 0.3667.$$

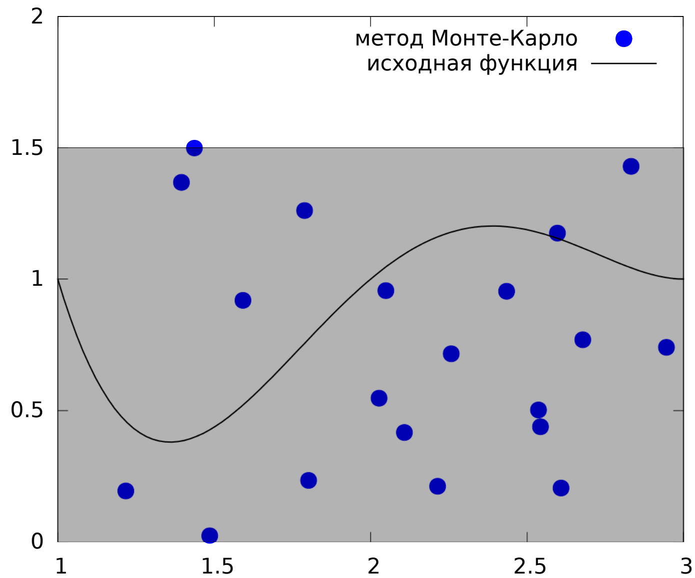

Рисунок 4: Метод Монте-Карло

1.3 Правило Рунге

Стоит отметить, что обычно вычисление интеграла требуется с заданной точностью, но часто бывает сложно заранее подобрать размер разбиения $n(\varepsilon)$ для вычисления интеграла с заданной точностью ε. Для автоматического выбора размера разбиения используется правило Рунге:

1. задается начальное $n = n_0$ (например 4);
2. вычисляется I_n и I_{2n};
3. если $|I_n - I_{2n}| < \varepsilon$, то I_{2n} и будет ответом;
4. иначе n удваивается $n := 2n$ и происходит возврат к шагу 2.

Псевдокод. Правило Рунге:

```
I2 := integral(n)
do
    I1 := I2
    n := 2 * n
    I2 := integral(n)
while |I2 - I1| > eps
```

Алгоритмическая сложность. Все методы имеют линейную сложность $\Theta(n) * m(\varepsilon)$, где $m(\varepsilon)$ — количество итераций.

2 Вычисление корней уравнений

Постановка задачи. Дана непрерывная на отрезке $[a;b]$ функция $f(x)$, которая имеет на этом отрезке единственный корень, причем $f(a)f(b) < 0$. Необходимо вычислить корень x с заранее заданной точностью ε:

$$|x^* - x| < \varepsilon, f(x) = 0.$$

Подробно с методами можно ознакомить в книге [1, стр. 138] и [2, стр. 130]

2.1 Метод деления отрезка пополам

Этот метод также называется «бинарный поиск» или «дихотомия». Рассмотрим функцию $f(x)$ такую, что:

$$f(l)f(r) < 0,$$

если $l < x < r$, где x — корень. То есть $f(l)$ и $f(r)$ разного знака справа и слева от корня. Вычислим значение в середине отрезка $m = \frac{l+r}{2}$. Если значение $f(b)$ такого же знака, что и $f(m)$, то корень гарантировано находится на отрезке $[l;m]$, иначе на отрезке $[m;r]$.

Пример. Найдем решение уравнения $x^2 - 2 = 0$ на отрезке $[0;2]$ с точностью $\varepsilon = 0.2$. Рассмотрим $f(x) = x^2 - 2$. Знаки на границах: $f(0) = -2 < 0$, $f(2) = 2 > 0$.

1. Текущие границы: $l = 0$, $r = 2$. Середина $m = \frac{0+2}{2} = 1$.

   Значение в середине $f(m) = f(1) = -1$ совпадает по знаку со значением $f(0) = -2$.

   Уменьшаем отрезок до $[1;2]$.

2. Текущие границы: $l = 1$, $r = 2$. Середина $m = \frac{1+2}{2} = \frac{3}{2}$.

   Значение в середине $f(m) = f\left(\frac{3}{2}\right) = \frac{1}{4}$ совпадает по знаку со значением $f(b) = 2$.

   Уменьшаем отрезок до $\left[1; \frac{3}{2}\right]$.

3. Текущие границы: $l = 1$, $r = \frac{3}{2}$. Середина $m = \frac{1+\frac{3}{2}}{2} = \frac{5}{4}$.

   Значение в середине $f(m) = f\left(\frac{5}{4}\right) = -\frac{7}{16}$ совпадает по знаку со значением $f(0) = -2$.

   Уменьшаем отрезок до $\left[\frac{5}{4}; \frac{3}{2}\right]$.

4. Текущие границы: $l = \frac{5}{4}$, $r = \frac{3}{2}$. Середина $m = \frac{\frac{5}{4}+\frac{3}{2}}{2} = \frac{11}{8}$.

   Значение в середине $f(m) = f\left(\frac{11}{8}\right) = -\frac{7}{64}$ совпадает по знаку со значением $f(0) = -2$.

   Уменьшаем отрезок до $\left[\frac{11}{8}; \frac{3}{2}\right]$. Заметим, что длина отрезка меньше ε: $\frac{3}{2} - \frac{11}{8} < 0.2$. Ответ (среднее между l и r): $\frac{23}{16}$ с точостью до 0.2.

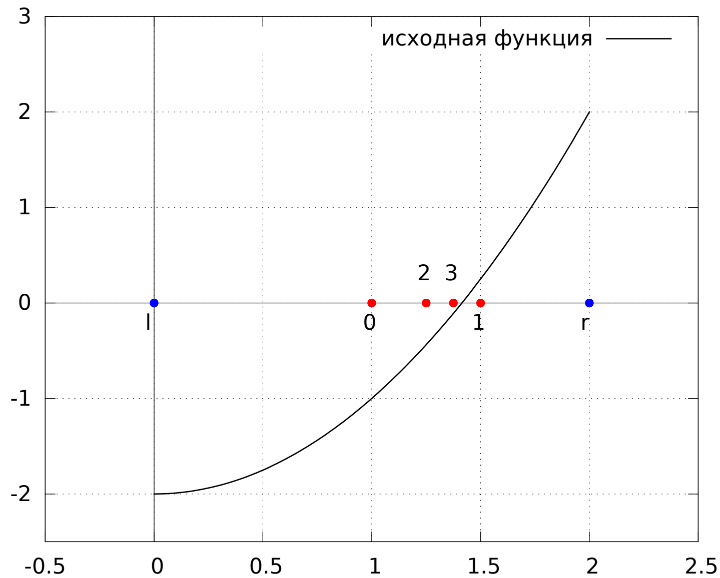

Рисунок 5: Метод деления отрезка пополам

Псевдокод.

```
l := a
r := b
s := f(b)
while |r - l| > eps
    m := (r + l) / 2
    if s * f(m) > 0
        l := m
    else
        r := m
return (l + r) / 2
```

Сходимость. Заметим, что после каждой итерации, область поиска корення уменьшается в 2 раза. Значит количество итераций можно определить из неравенства:

$$\frac{b-a}{2^n} \leq \varepsilon.$$

Откуда легко найти количество итераций:

$$n \geq \left[\log_2 \frac{b-a}{\varepsilon}\right].$$

Скорость сходимости «линейная» с параметром $\frac{1}{2}$:

$$|x - x_{k+1}| \leq \frac{1}{2}|x - x_k|.$$

2.2 Метод хорд

Рассмотрим функцию $f(x)$ такую, что $f(x)$ дважды дифференцируема на отрезке $[a;b]$, а первая и второая производная не изменяет знак на всем отрезке.

Проведем хорду через точки $(a;f(a))$ и $(b;f(b))$. Уравнение соответствующей прямой:

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

Найдем точку пересечения хорды и с осью абсцисс ($y = 0$):

$$x = a - f(a)\frac{b-a}{f(b)-f(a)}. \tag{1}$$

Эквивалентная формы:

$$x = b - f(b)\frac{b-a}{f(b)-f(a)} \tag{2}$$

и

$$x = \frac{af(b) - bf(a)}{f(b)-f(a)}. \tag{3}$$

Если $f'(x)f''(x) > 0$ на всем отрезке $[a;b]$, то строятся хорды с фиксированным правым концом в форме (2):

$$\begin{cases} x_0 &= a, \\ x_{k+1} &= b - f(b)\frac{b-x_k}{f(b)-f(x_k)} \end{cases}$$

Такая форма позволяет оптимизировать вычисления функции и умножения–деления. Например, в форме (1) и (2) будет 2 умножения–деления против 3 умножений–делений. Также вычисление функции в крайней точке можно вычислить заранее.

Если $f'(x)f''(x) < 0$ на всем отрезке $[a;b]$, то строятся хорды с фиксированным левым концом в форме (1):

$$\begin{cases} x_0 &= b, \\ x_{k+1} &= a - f(a)\frac{x_k-a}{f(x_k)-f(a)} \end{cases}$$

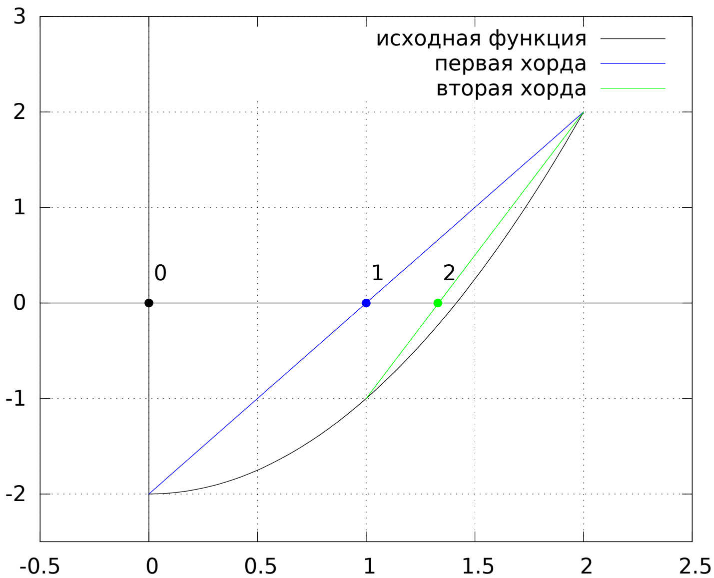

Рисунок 6: Метод хорд

Пример. Найдем решение уравнения $x^2 - 2 = 0$ на отрезке $[0; 2]$ с точностью $\varepsilon = 0.2$. Рассмотрим $f(x) = x^2 - 2$. Производные $f'(x) = 2x > 0$ и $f''(x) = 2 > 0$. Так как $f'(x)f''(x) > 0$, необходимо двигаться слева направо $(x_0 = 0)$ с фиксированным правым концом хорд.

Вычисления:

$$x_{k+1} = 2 - 2 * \frac{2 - x_k}{2 - f(x_k)}.$$

1. $x_0 = 0,\ f(x_0) = -2.$

   $x_1 = 2 - 2 * \frac{2-x_0}{2-f(x_0)} = 2 - 2 * \frac{2-0}{2+2} = 1.$

2. $x_1 = 1,\ f(x_1) = -1.$

   $x_2 = 2 - 2 * \frac{2-x_1}{2-f(x_1)} = 2 - 2 * \frac{2-1}{2+1} = \frac{4}{3}.$

3. $x_1 = \frac{4}{3},\ f(x_1) = -\frac{2}{9}.$

   $x_3 = 2 - 2 * \frac{2-x_2}{2-f(x_2)} = 2 - 2 * \frac{2-\frac{4}{3}}{2+\frac{2}{9}} = \frac{7}{5}.$

   Заметим, что расстояние между 2 последними итерациями меньше ε: $\left|\frac{7}{5} - \frac{4}{3}\right| = \frac{1}{15} < 0.2$. Ответ: $\frac{7}{5}$ с точостью до 0.2.

Псевдокод. Обозначим `f(x)` — исходную функцию, `f1(x)` — производная (вычисленная аналитически), `f2(x)` — вторая производная (вычисленная аналитически).

```
m := (r + l) / 2
if f1(m) * f2(m) > 0
    xnew := a
    fb := f(b)
    do
        xold := xnew
        xnew := b - fb * (b - xold) / (fb - f(xold))
    while |xold - xnew| > eps
else
    xnew := b
    fa := f(a)
    do
        xold := xnew
        xnew := a - fa * (xold - a) / (f(xold) - fa)
    while |xold - xnew| > eps
return xnew
```

Сходимость. В литературе можно найти вывод достаточных условий сходимости, которые релизованы выше. Скорость сходимости «линейная», то есть, существует такой $0 < L < 1$, что:

$$|x - x_{k+1}| \leq L|x - x_k|.$$

2.3 Метод Ньютона

Этот метод также называется «метод касательных». Рассмотрим функцию $f(x)$ такую, что $f(x)$ дважды дифференцируема на отрезке $[a; b]$, а первая и второая производная не изменяет знак на всем отрезке.

Проведем касательную к графику через точку $(x_k; f(x_k))$. Уравнение соответствующей прямой:

$$y = f'(x_k)(x - x_k) + f(x_k).$$

Найдем точку пересечения хорды и с осью абсцисс ($y = 0$):

$$x = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Откуда легко получить формулу для вычисления корня методом касательных:

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

Если $f'(x)f''(x) > 0$ на всем отрезке $[a; b]$, то в качестве начального приближения выбирается правый конец, а иначе — левый.

Пример. Найдем решение уравнения $x^2 - 2 = 0$ на отрезке $[0; 2]$ с точностью $\varepsilon = 0.2$. Рассмотрим $f(x) = x^2 - 2$. Производные $f'(x) = 2x > 0$ и $f''(x) = 2 > 0$. Так как $f'(x)f''(x) > 0$, необходимо двигаться справа налево ($x_0 = 2$).

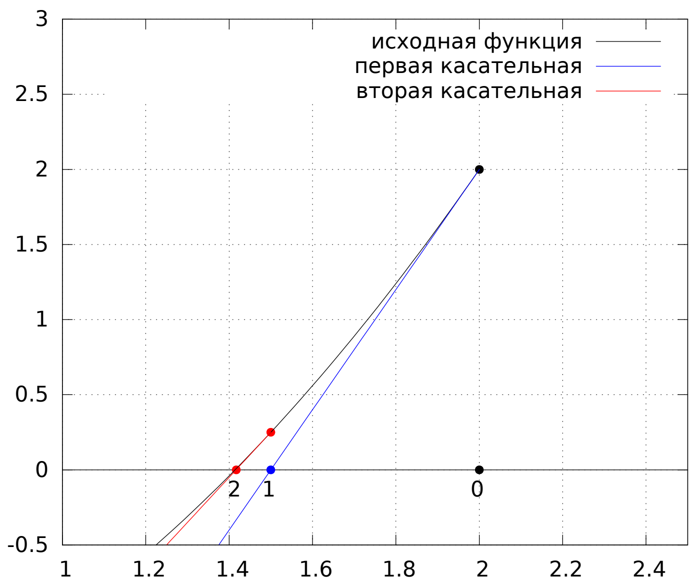

Рисунок 7: Метод касательных

Вычисления:

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

1. $x_0 = 2,\ f(x_0) = 2,\ f'(x_0) = 4.$

   $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 2 - \frac{1}{2} = \frac{3}{2}.$

2. $x_1 = \frac{3}{2},\ f(x_1) = \frac{1}{4},\ f'(x_1) = 3.$

   $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = \frac{3}{2} - \frac{1}{12} = \frac{17}{12}.$

   Заметим, что расстояние между 2 последними итерациями меньше ε: $\left|\frac{17}{12} - \frac{3}{2}\right| = \frac{1}{12} < 0.2$. Ответ: $\frac{17}{12}$ с точностью до 0.2.

Псевдокод.

```
m := (r + l) / 2
if f1(m) * f2(m) > 0
    xnew := a
```

```
else
    xnew := b
do
    xold := xnew
    xnew := b - f(xold) / f1(xold)
while |xold - xnew| > eps
return xnew
```

Сходимость. В литературе можно найти вывод достаточных условий, которые релизованы выше. Скорость сходимости «квадратичная», то есть, существует такой $0 < L < 1$, что:

$$|x - x_{k+1}| \leq L|x - x_k|^2.$$

3 Аппроксимация

Аппроксимация — математический метод, состоящий в замене одних математических объектов другими, в том или ином смысле близкими к исходным. Для дальнейшего описания введем сетку порядка n:

$$\Omega = \{x_i, i = 0, ..., n; x_0 < x_1 < ... < x_n\}.$$

3.1 Метод наименьших квадратов (вектор)

Постановка задачи. Дан набор измерений $y_i = f(x_i)$. Необходимо аппроксимировать вектор y вектором $\tilde{y}$, который получен линейной аппроксимацией на сетке Ω ($\tilde{y}_i = A * x_i + B$), причем с минимальным отклонением на сетке:

$$\min_{A,B}(y - \tilde{y}, y - \tilde{y}).$$

Метод наименьших квадратов. Минимизируем:

$$F(a, b) = \sum_{i=0}^{n}(ax_i + b - y_i)^2$$

Производные по a и b должны обнулиться:

$$\begin{cases} \sum_{i=0}^{n} x_i(ax_i + b - y_i) = 0 \\ \sum_{i=0}^{n}(ax_i + b - y_i) = 0 \end{cases}$$

$$\begin{cases} a\sum_{i=0}^{n} x_i^2 + b\sum_{i=0}^{n} x_i = \sum_{i=0}^{n} x_i y_i \\ a\sum_{i=0}^{n} x_i + b\sum_{i=0}^{n} 1 = \sum_{i=0}^{n} y_i \end{cases}$$

Введем обозначение средних величин:

$$\overline{z} = \frac{1}{n+1}\sum_{i=0}^{n} z_i.$$

Поделим каждое уравнение системы на $(n + 1)$ и получим:

$$\begin{cases} a\overline{x^2} + b\overline{x} = \overline{xy} \\ a\overline{x} + b = \overline{y} \end{cases} \tag{4}$$

Итоговые коэффициенты:

$$\begin{cases} a = \dfrac{\overline{xy} - \overline{x} * \overline{y}}{\overline{x^2} - \overline{x}^2} \\ b = \overline{y} - a\overline{x} \end{cases} \tag{5}$$

Пример. Дана сетка размером $N = 3$ на отрезке $[0;1]$ со значениями в узлах: $y_0 = 1$, $y_1 = 4$, $y_2 = 2$, $y_3 = 5$. Построить линейную аппроксимацию методом наименьших квадратов.

Рассмотрим 2 вектора: $x = \left(0, \frac{1}{3}, \frac{2}{3}, 1\right)$ и $y = (1, 4, 2, 5)$.

Вычислим средние величины:

$$\overline{x} = \frac{1}{4}(x_0 + x_1 + x_2 + x_3) = \frac{1}{4}\left(0 + \frac{1}{3} + \frac{2}{3} + 1\right) = \frac{1}{2}$$

$$\overline{y} = \frac{1}{4}(y_0 + y_1 + y_2 + y_3) = \frac{1}{4}(1 + 4 + 2 + 5) = 3$$

$$\overline{xy} = \frac{1}{4}(x_0y_0 + x_1y_1 + x_2y_2 + x_3y_3) = \frac{1}{4}\left(0 * 1 + \frac{1}{3} * 4 + \frac{2}{3} * 2 + 1 * 5\right) = \frac{23}{12}$$

$$\overline{x^2} = \frac{1}{4}(x_0^2 + x_1^2 + x_2^2 + x_3^2) = \frac{1}{4}\left(0 + \frac{1}{9} + \frac{4}{9} + 1\right) = \frac{7}{18}$$

Коэффициенты прямой:

$$a = \frac{\overline{xy} - \overline{x} * \overline{y}}{\overline{x^2} - \overline{x}^2} = \frac{\frac{23}{12} - \frac{1}{2} * 3}{\frac{7}{18} - \frac{1}{4}} = 3$$

$$b = \overline{y} - a\overline{x} = 3 - 3 * \frac{1}{2} = \frac{3}{2}$$

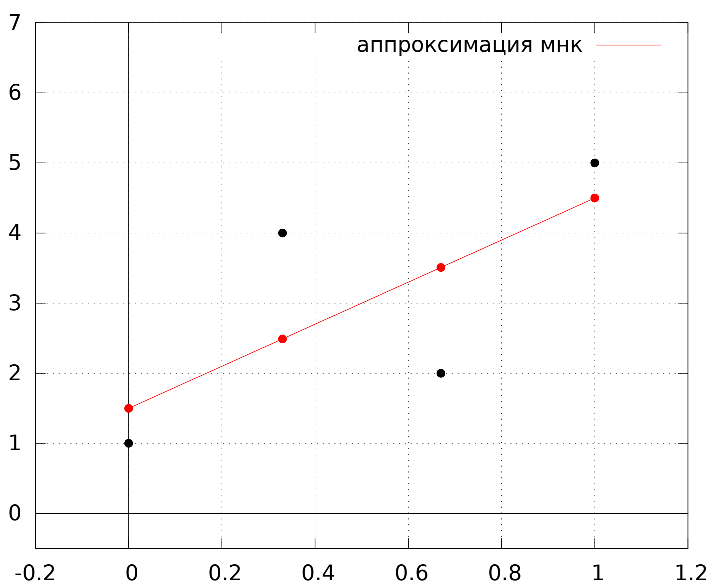

Рисунок 8: Линейная аппроксимация

Псевдокод. Код, вычисляющий коэффициенты:

```
prepare(n, x[], y[], var a, var b)
    mx := 0
    my := 0
    mxy := 0
    mxx := 0
    for i := 0 .. n
        mx := mx + x[i]
        my := my + y[i]
        mxy := mxy + x[i] * y[i]
        mxx := mxx + x[i] * x[i]
    mx := mx / (n + 1)
    my := my / (n + 1)
    mxy := mxy / (n + 1)
    mxx := mxx / (n + 1)
    a := (mxy - mx * my) / (mxx - mx * mx)
    b := my - a * mx
    return a, b
```

3.2 Метод наименьших квадратов (функция)

Общая задача аппроксимации МНК заключается в разложении данного элемента f из линейного нормированного пространства L по базису линейного подпространства φ_i, где $i = 0, ..., n$. Причем так, чтобы норма разности разложения f и самого f была минимальна:

$$\min_{c_i} ||f - \sum c_i \varphi_i||.$$

Заметим, что первая задача сводится к данной, если в взять пространство R^{n+1} и базисные элементы $(x_0, x_1, ..., x_n)$ и $(1, 1, ..., 1)$.

$$F(c_0, c_1, ..., c_n) = (f - \sum_{i=0}^{n} c_i \varphi_i, f - \sum_{j=0}^{n} c_j \varphi_j) =$$

$$= (f, f) - 2 \sum_{i=0}^{n} c_i (\varphi_i, f) + \sum_{i=0}^{n} c_i \sum_{j=0}^{n} c_j (\varphi_i, \varphi_j)$$

Производные по c_k должны обнулиться:

$$-2c_k(\varphi_k, f) + 2c_k \sum_{j=0}^{n} (\varphi_k, \varphi_j) = 0$$

.

Откуда и получаем систему, которая позволяет найти коэффициенты аппроксимации.

$$
\begin{pmatrix}
(\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & ... & (\varphi_0,\varphi_n) \\
(\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & ... & (\varphi_1,\varphi_n) \\
... & ... & ... & ... \\
(\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & ... & (\varphi_n,\varphi_n)
\end{pmatrix}
\begin{pmatrix} c_0 \\ c_1 \\ ... \\ c_n \end{pmatrix}
=
\begin{pmatrix} (\varphi_0,f) \\ (\varphi_1,f) \\ ... \\ (\varphi_n,f) \end{pmatrix}
\tag{6}
$$

Легко заметить соответствие данной системы с системой (4).

Рассмотрим на примере кусочно–линейной аппроксимации $f(x)$. Скалярное произведение функций определим стандартно:

$$
(f,g) = \int_0^1 f(x)g(x)dx.
$$

Для более простого описания введем равномерную сетку с шагом $h = \frac{1}{n}$. В качестве основной базисной функции выберем:

$$
\varphi(x) = \begin{cases} 1-x, & x \in [0;1] \\ 1+x, & x \in [-1;0) \\ 0, & x \notin [-1;1] \end{cases}
$$

Данную базисную функцию сожмем до интервала $[-h;h]$ и сдвинем на x_i для всех $i = 0,...,n$. Получим базис:

$$
\varphi_i(x) = \varphi\left(\frac{x-x_i}{h}\right)
$$

Вычислим значения матрицы Грамма (6). Легко увидеть, что:

$$
\int_{-1}^1 \varphi^2(x)dx = 2\int_0^1 \varphi^2(x)dx = 2\int_0^1 x^2dx = \frac{2}{3}.
$$

Значит диагональные элементы равны $(i = 1,...,n-1)$:

$$
(\varphi_i(x),\varphi_i(x)) = \frac{2h}{3}.
$$

Для первого и последнего элемента ответ будет в 2 раза меньше:

$$
(\varphi_0(x),\varphi_0(x)) = (\varphi_n(x),\varphi_n(x)) = \frac{h}{3}.
$$

Далее посмотрим скалярное произведение 2 подряд идущих базисных функций (над и под главной диагональю):

$$
(\varphi_i(x),\varphi_{i+1}(x)) = h\int_0^1 (1-x)xdx = \frac{h}{6}.
$$

Далее посмотрим скалярное произведение остальных пар $|i-j| > 1$. Ясно, что скалярное произведение функций нулевое:

$$
(\varphi_i(x),\varphi_j(x)) = 0.
$$

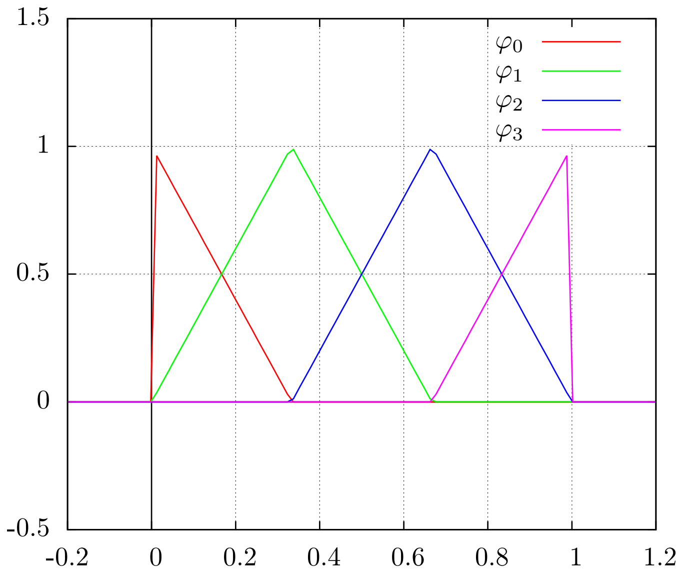

Рисунок 9: Базисные функции $\varphi(x)$

В итоге, необходимо найти решение СЛАУ (все строки матрицы умножены на $\frac{6}{h}$) с трехдиагональной матрицей:

$$\begin{pmatrix} 2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \cdots \\ c_{n-2} \\ c_{n-1} \\ c_n \end{pmatrix} = \frac{6}{h} \begin{pmatrix} (\varphi_0, f) \\ (\varphi_1, f) \\ (\varphi_2, f) \\ \cdots \\ (\varphi_{n-2}, f) \\ (\varphi_{n-1}, f) \\ (\varphi_n, f) \end{pmatrix} \qquad (7)$$

Отметим, что интегралы для правых частей надо будет вычислять численно. Причем для первой и последней компоненты с особенными границами:

$$(\varphi_i, f) = \begin{cases} \int_{x_i}^{x_{i+1}} \varphi_i(x) f(x) dx, i = 0 \\ \int_{x_{i-1}}^{x_{i+1}} \varphi_i(x) f(x) dx, i = 1..(n-1) \\ \int_{x_{i-1}}^{x_i} \varphi_i(x) f(x) dx, i = n \end{cases} .$$

Пример. Дана сетка размером $N = 3$ на отрезке $[0; 1]$. Построить кусочно–линейную аппроксимацию функции $f(x) = 54x^2$ методом наименьших квадратов с минимальным интегральным отклонением. Сетка для базисных функций $x_0 = 0$, $x_1 = \frac{1}{3}$, $x_2 = \frac{2}{3}$, $x_3 = 1$. Шаг $h = \frac{1}{3}$.

Опишем базисные функции кусочно–линейной аппроксимации:

$$\varphi_0(x) = \varphi\left(\frac{x-0}{1/3}\right) = \begin{cases} 1-3x & , x \in \left[0, \dfrac{1}{3}\right] \\ 0 & , x \notin \left[0, \dfrac{1}{3}\right] \end{cases}$$

$$\varphi_1(x) = \varphi\left(\frac{x-1/3}{1/3}\right) = \begin{cases} 0+3x & , x \in \left[0, \dfrac{1}{3}\right] \\ 2-3x & , x \in \left[\dfrac{1}{3}, \dfrac{2}{3}\right] \\ 0 & , x \notin \left[0, \dfrac{2}{3}\right] \end{cases}$$

$$\varphi_2(x) = \varphi\left(\frac{x-2/3}{1/3}\right) = \begin{cases} -1+3x & , x \in \left[\dfrac{1}{3}, \dfrac{2}{3}\right] \\ 3-3x & , x \in \left[\dfrac{2}{3}, 1\right] \\ 0 & , x \notin \left[\dfrac{1}{3}, 1\right] \end{cases}$$

$$\varphi_3(x) = \varphi\left(\frac{x-1}{1/3}\right) = \begin{cases} -2+3x & , x \in \left[\dfrac{2}{3}, 1\right] \\ 0 & , x \notin \left[\dfrac{2}{3}, 1\right] \end{cases}$$

Вычислим проекции функции $f(x) = 54x^2$ на базисные функции $\varphi_i(x)$:

$$(f, \varphi_0) = \int_{\frac{0}{3}}^{\frac{1}{3}} 54x^2(1-3x)dx = \frac{1}{6} \approx 0.16667$$

$$(f, \varphi_1) = \int_{\frac{0}{3}}^{\frac{1}{3}} 54x^2(0+3x)dx + \int_{\frac{1}{3}}^{\frac{2}{3}} 54x^2(2-3x)dx = \frac{7}{3} \approx 2.33333$$

$$(f, \varphi_1) = \int_{\frac{1}{3}}^{\frac{2}{3}} 54x^2(-1+3x)dx + \int_{\frac{2}{3}}^{\frac{3}{3}} 54x^2(3-3x)dx = \frac{25}{3} \approx 8.33333$$

$$(f, \varphi_1) = \int_{\frac{2}{3}}^{\frac{3}{3}} 54x^2(-2+3x)dx = \frac{43}{6} \approx 7.16667$$

Умножаем полученные значения на $\frac{6}{h} = 18$ и получаем вектор–столбец $(3, 42, 150, 129)^T$ для СЛАУ, определяющая коэффициенты аппрокосимации:

$$\begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 42 \\ 150 \\ 129 \end{pmatrix} \tag{8}$$

Решая данную систему методом прогонки, находим коэффициенты:

$$\begin{cases} c_0 = -1 \\ c_1 = 5 \\ c_2 = 23 \\ c_3 = 53 \end{cases}$$

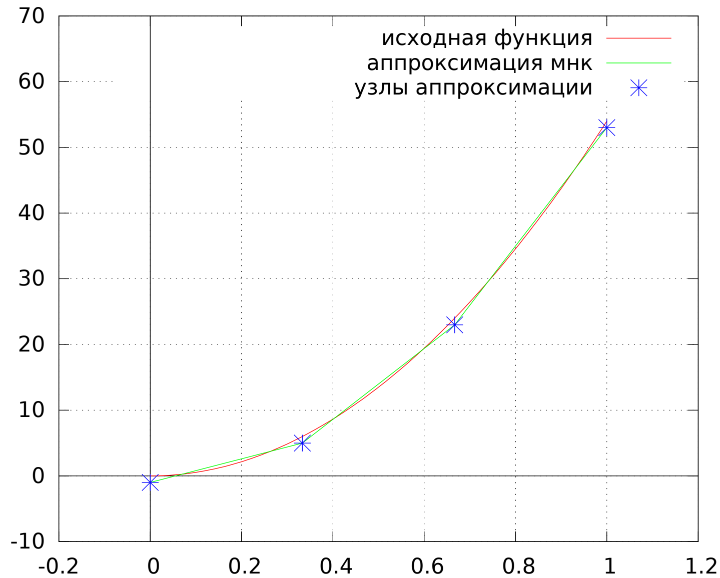

Рисунок 10: Кусочно-линейная аппроксимация

То есть аппроксимирующая функция выглядит так:

$$f(x) \approx -\varphi_0(x) + 5\varphi_1(x) + 23\varphi_2(x) + 53\varphi_3(x).$$

А именно, это кусочно линейная функция, которая проходит через узлы: $(0, -1)$, $\left(\frac{1}{3}, 5\right)$, $\left(\frac{2}{3}, 23\right)$, $(1, 53)$.

Псевдокод. Функции, задающие образ базиса и базисные функции (`xi` — центр x_i):

```
phi(x)
    if (x >= -1) and (x <= 0)
        return 1 + x
    if (x >= 0) and (x <= 1)
        return 1 - x
    return 0
```

```
phii(xi, h, x)
    return phi( (x - xi) / h)
```

Функция, которая вычисляет интеграл

$$\int_{xi+h*left}^{xi+h*right} f(x)\varphi_i(x)dx$$

методом трапеций (1000 трапеций). Причем $left$ и $right$ — границы функции $\varphi(x)$, которые используются как индикаторы левого и правого края (отличаются у первой и последней базисной функции).

```
integral(xi, h, left, right)
    m := 1000
    l = xi + left * h
    r = xi + right * h
    dt := (r - l) / m

    t := l
    s := 0.5 * f(l) * phii(xi, h, l) + 0.5 * f(r) * phii(xi, h, r)
    for j := 1 .. m-1
        t := t + dt
        s := s + f(t) * phii(xi, h, t)
    s := s * dt
    return s
```

Функция вычисления весовых коэффициентов c_i с помощью функций $integral$ — численное интегрирование и $sweep$ — метод прогонки.

```
prepare(n, a, b, var c[])
    h := (b - a) / n
    xi := 0

    dl[0] := 0
    dd[0] := 2
    du[0] := 1
    f[i] := 6 / h * integral(xi, h, 0, 1)
    xi := xi + h
    for i := 1 .. n-1
        dl[i] := 1
        dd[i] := 4
        du[i] := 1
        f[i] := 6 / h * integral(xi, h, -1, 1)
        xi := xi + h
    dl[n] := 1
    dd[n] := 2
    du[n] := 0
```

```
f[i] := 6 / h * integral(xi, h, -1, 0)

sweep(n+1, dl, dd, du, f, x)

for i := 0 .. n
c[i] = x[i]
return c[]
```

4 Интерполяция

Интерполяция — способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений. Для дальнейшего описания введем сетку порядка n:

$$\Omega = \{x_i, i = 0, ..., n; x_0 < x_1 < ... < x_n\}.$$

Подробнее можно ознакомить в [1, стр. 44] и [2, стр. 65].

4.1 Интерполяционный многочлен Лагранжа

Постановка задачи. Дан набор измерений $y_i = f(x_i)$. Необходимо построить многочлен n-й степени, который будет проходить через заданные точки.

Интерполяционный многочлен Лагранжа В качестве i-го элемента базиса построим многочлен, которые обнуляется во всех точках, кроме x_i (а в ней принимает значение 1):

$$\varphi_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}$$

Итоговый многочлен выглядит так:

$$P(x) = \sum_{i=0}^{n} y_i \varphi_i(x)$$

4.2 Сплайновая интерполяция

Постановка задачи. Пусть на отрезке $[0; 1]$ имеется равномерная сетка из N частей ($N + 1$ узлов). Шаг сетки: $h = \frac{1}{N}$. Узлы сетки: $x_i = i * h$. На этой сетке известны дискретные значения некоторой функции $f(x)$: $y_i = f(x_i)$. Необходимо построить функцию $p(x)$ такую, чтобы она совпадала с $f(x)$ в узлах ($p(x_i) = y_i$) и была достаточно гладкая (дважды непрерывно дифференцируемая).

Сплайны. На i-м отрезке $[x_i; x_{i+1}]$ будем аппроксимировать $f(x)$ сплайном – кубической функцией (для каждого отрезка функция будет отедельная):

$$P_i(x) = A_i * (x - x_i)^3 + B_i * (x - x_i)^2 + C_i * (x - x_i) + D_i, i = \overline{0, n-1}$$

1. Значение i-го сплайна на левом конце соответситвующего интервала равно y_i:

$$P_i(x_i) = y_i, i = \overline{0, n-1}$$

2. Значение i-го сплайна на левом конце соответситвующего интервала равно y_{i+1}:

$$P_i(x_{i+1}) = y_{i+1}, i = \overline{1, n}$$

3. Первые производные сплайнов непрерывны в узлах:

$$P_i'(x_{i+1}) = P_{i+1}'(x_{i+1}), i = \overline{0, n-2}$$

4. Вторые производные сплайнов непрерывны в узлах:

$$P_i''(x_{i+1}) = P_{i+1}''(x_{i+1}), i = \overline{0, n-2}$$

5. Концы – свободные:

$$P_0''(x_0) = 0, P_{n-1}''(x_n) = 0$$

Выпишем все условия через коэффициенты сплайнов:

$$\begin{array}{ll}
D_i = y_i, & i = \overline{0, n-1} \\
A_i h^3 + B_i h^2 + C_i h + y_i = y_{i+1}, & i = \overline{0, n-1} \\
3A_i h^2 + 2B_i h + C_i = C_{i+1}, & i = \overline{0, n-2} \\
6A_i h + 2B_i = 2B_{i+1}, & i = \overline{0, n-2} \\
2B_0 = 0 & \\
6A_{n-1} h + 2B_{n-1} = 0 &
\end{array}$$

Соотношение (1) позволяет найти сразу D_i:

$$D_i = y_i, i = \overline{0, n-1} \qquad (9)$$

Выразим A_i через (4) и (6):

$$A_i = \frac{B_{i+1} - B_i}{3h}, i = \overline{0, n-2}$$

$$A_{n-1} = \frac{0 - B_{n-1}}{3h}$$

Для единообразия введем фиктивный элемент $B_n = 0$ и получим формулы вычисления A_i:

$$A_i = \frac{B_{i+1} - B_i}{3h}, i = \overline{0, n-1} \qquad (10)$$

Подставим полученное выражение во все оставшиеся неразрешенные соотношения:

$$\begin{array}{ll}
(B_{i+1} - B_i)\frac{h^2}{3} + B_i h^2 + C_i h + y_i = y_{i+1}, & i = \overline{0, n-1} \\
(B_{i+1} - B_i)h + 2B_i h + C_i = C_{i+1}, & i = \overline{0, n-2} \\
B_0 = 0 & \\
B_n = 0 &
\end{array}$$

Выразим C_i из (1) и получим формулы вычисления C_i:

$$C_i = \frac{y_{i+1} - y_i}{h} - (B_{i+1} + 2B_i)\frac{h}{3}, i = \overline{0, n-1} \tag{11}$$

Подставим C_i в (2):

$$(B_{i+1} - B_i)h + 2B_i h + \frac{y_{i+1} - y_i}{h} - (B_{i+1} + 2B_i)\frac{h}{3} = \\ = \frac{y_{i+2} - y_{i+1}}{h} - (B_{i+2} + 2B_{i+1})\frac{h}{3}, i = \overline{0, n-2}$$

$$B_i + 4B_{i+1} + B_{i+2} = 3\frac{y_i - 2y_{i+1} + y_{i+2}}{h^2}, i = \overline{0, n-2}$$

Немного упростив, получим СЛАУ $(n+1)$-го порядка с трехдиагональной матрицей:

$$\begin{cases} B_0 = 0 \\ B_{i-1} + 4B_i + B_{i+1} = 3\dfrac{y_{i-1} - 2y_i + y_{i+1}}{h^2}, i = \overline{1, n-1} \\ B_n = 0 \end{cases} \tag{12}$$

А именно:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \\ \vdots \\ B_{n-2} \\ B_{n-1} \\ B_n \end{pmatrix} = \begin{pmatrix} 0 \\ \overline{y}_1 \\ \overline{y}_2 \\ \overline{y}_3 \\ \vdots \\ \overline{y}_{n-2} \\ \overline{y}_{n-1} \\ 0 \end{pmatrix} \tag{13}$$

где

$$\overline{y}_i = 3\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}$$

.

По диагоналям:

$$a = (0, 1, 1, 1, ..., 1, 1, 0)$$

$$b = (1, 4, 4, 4, ..., 4, 4, 1)$$

$$c = (0, 1, 1, 1, ..., 1, 1, 0)$$

$$f = (0, \overline{y}_1, \overline{y}_2, \overline{y}_3, ..., \overline{y}_{n-2}, \overline{y}_{n-1}, 0)$$

Методом прогонки можно решить систему $Ax = f$ и получить вектор коэффициентов x. Далее вычислить основные коэффициенты ($i = \overline{0, n-1}$):

$$\begin{cases} A_i = \dfrac{x_{i+1} - x_i}{3h} \\ B_i = x_i \\ C_i = \dfrac{y_{i+1} - y_i}{h} - (x_{i+1} + 2x_i)\dfrac{h}{3} \\ D_i = y_i \end{cases} \tag{14}$$

Пример. Дана сетка на $N = 3$ отрезка со значениями в узлах: $y_0 = 1$, $y_1 = 4$, $y_2 = 2$, $y_3 = 5$. Построить сплайновую аппроксимацию.

Шаг сетки: $h = 1/3$. Посчитаем правые части для СЛАУ:

$$\overline{y}_1 = 3(y_0 - 2y_1 + y_2)/h^2 = 3 * (1 - 2 * 4 + 2) * 9 = -135$$

$$\overline{y}_2 = 3(y_1 - 2y_2 + y_3)/h^2 = 3 * (4 - 2 * 2 + 5) * 9 = 135$$

Решаем методом прогонки:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -135 \\ 135 \\ 0 \end{pmatrix}$$

Получили решение СЛАУ: $x = (0, -45, 45, 0)$. Далее вычисляем коэффициенты A, B, C, D:

$A_0 = \frac{x_1 - x_0}{3h} = \frac{-45-0}{1} = -45$

$A_1 = \frac{x_2 - x_1}{3h} = \frac{45-(-45)}{1} = 90$

$A_2 = \frac{x_3 - x_2}{3h} = \frac{0-45}{1} = -45$

$B_0 = x_0 = 0$

$B_1 = x_1 = -45$

$B_2 = x_2 = 45$

$C_0 = \frac{y_1 - y_0}{h} - \frac{(x_1 + 2x_0)h}{3} = (4 - 1) * 3 - (-45 + 2 * 0) * \frac{1}{9} = 14$

$C_1 = \frac{y_2 - y_1}{h} - \frac{(x_2 + 2x_1)h}{3} = (2 - 4) * 3 - (45 + 2 * (-45)) * \frac{1}{9} = -1$

$C_2 = \frac{y_3 - y_2}{h} - \frac{(x_3 + 2x_2)h}{3} = (5 - 2) * 3 - (0 + 2 * 45) * \frac{1}{9} = -1$

$D_0 = y_0 = 1$

$D_1 = y_1 = 4$

$D_2 = y_2 = 2$

Сплайновая аппроксимация:

$$P(x) = \begin{cases} -45(x-0)^3 + 0(x-0)^2 + 14(x-0) + 1 & ,x \in \left[0; \dfrac{1}{3}\right) \\ 90\left(x - \dfrac{1}{3}\right)^3 - 45\left(x - \dfrac{1}{3}\right)^2 - \left(x - \dfrac{1}{3}\right) + 4 & ,x \in \left[\dfrac{1}{3}; \dfrac{2}{3}\right) \\ -45\left(x - \dfrac{2}{3}\right)^3 + 45\left(x - \dfrac{2}{3}\right)^2 - \left(x - \dfrac{2}{3}\right) + 2 & ,x \in \left[\dfrac{2}{3}; 1\right] \end{cases}$$

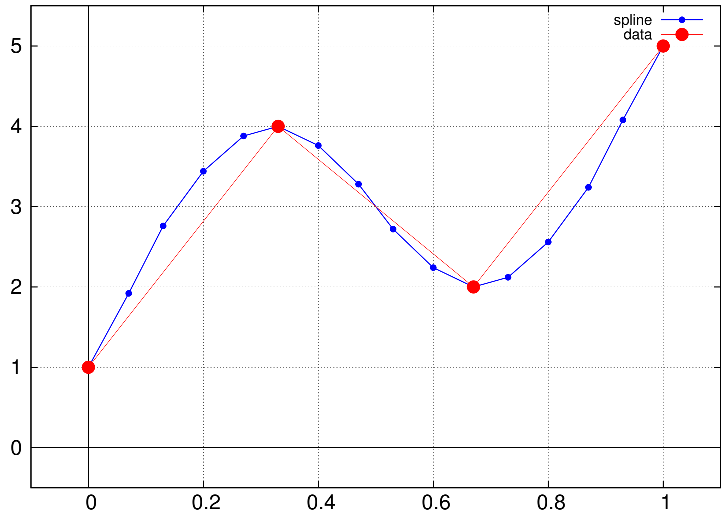

Рисунок 11: Сплайновая интерполяция

Псевдокод. Псевдокод для генерации коэффициентов:

```
splineGenerate(n, y[], var A[], var B[], var C[], var D[])
    h = 1 / n
    dl[0] := 0
    dd[0] := 1
    du[0] := 0
    f[0] := 0
    for i := 1 .. n-1
        dl[i] := 1
        dd[i] := 4
        du[i] := 1
        f[i] := 3 * (y[i-1] - 2 * y[i] + y[i+1]) / h / h
    dl[n] := 0
    dd[n] := 1
    du[n] := 0
    f[n] := 0
    sweep(n+1, dl, dd, du, f, x)
    for i := 0 .. (n-1)
        A[i] := (x[i+1] - x[i])/(3 * h)
        B[i] := x[i]
        C[i] := (y[i+1] - y[i]/h - (x[i+1] + 2 * x[i]) * h / 3
        D[i] := y[i]
    return A[], B[], C[], D[]
```

5 Вычисление экстремума функции

Постановка задачи. Дана непрерывная на отрезке $[a;b]$ функция $f(x)$, которая имеет на этом отрезке единственный локальный минимум (то есть функция выпукла вниз). Необходимо вычислить точку минимума x с заранее заданной точностью ε:

$$|x^* - x| < \varepsilon, f(x) = 0.$$

Подробнее можно ознакомить в [1, стр. 196].

5.1 Тернарный поиск

Этот метод технически напоминает «бинарный поиск» для поиска корней. Рассмотрим выпуклую вниз на отрезке $[l;r]$ функцию $f(x)$. Тогда любая хорда находится выше графика функции, то есть для любых $l < m_l < m_r < r$ верно:

$$\frac{m_r - \xi}{m_r - m_l} f(m_l) + \frac{\xi - m_l}{m_r - m_l} f(m_r) \geq f(\xi).$$

Если $f(m_l) > f(m_r)$, то точка минимума x_{min} не может находится на отрезке $[l;m_l]$. Иначе точка $(m_l;f(m_l))$ окажется выше хорды, проходящей через $(x_{min};f(x_{min}))$ и $(m_r;f(m_r))$. Значит, минимум гарантировано находится на отрезке $[m_l;r]$.

Если $f(m_l) < f(m_r)$, то аналогично получим, что минимум гарантировано находится на отрезке $[l;m_r]$.

В случае тернарного поиска m_l и m_r выбираются так, чтобы отрезок $[l;r]$ был разделен на 3 равных части:

$$\begin{cases} m_l = l + \frac{r-l}{3} \\ m_r = r - \frac{r-l}{3} \end{cases}$$

Пример. Найдем минимум функции $f(x) = x^3 - 3x$ на отрезке $[0;2]$ с точностью $\varepsilon = 0.5$.

1. Текущие границы: $l = 0$, $r = 2$. Середины:

$$m_l = 0 + \frac{2-0}{3} = \frac{2}{3}$$

$$m_r = 2 - \frac{2-0}{3} = \frac{4}{3}$$

Значение слева меньше, чем значение справа:

$$f(m_l) = f\left(\frac{2}{3}\right) = \frac{8}{27} - 2 = -\frac{46}{27}$$

$$f(m_r) = f\left(\frac{4}{3}\right) = \frac{62}{27} - 4 = -\frac{40}{27}$$

Уменьшаем отрезок до $\left[0; \frac{4}{3}\right]$.

2. Текущие границы: $l = 0$, $r = \frac{4}{3}$. Середины:

$$m_l = 0 + \frac{\frac{4}{3} - 0}{3} = \frac{4}{9}$$

$$m_r = \frac{4}{3} - \frac{\frac{4}{3} - 0}{3} = \frac{8}{9}$$

Значение справа меньше, чем значение слева:

$$f(m_l) = f\left(\frac{4}{9}\right) \approx -1.24554$$

$$f(m_r) = f\left(\frac{8}{9}\right) \approx -1.96433$$

Уменьшаем отрезок до $\left[\frac{4}{9}; \frac{4}{3}\right]$.

3. Текущие границы: $l = \frac{4}{9}$, $r = \frac{4}{3}$. Середины:

$$m_l = \frac{4}{9} + \frac{\frac{4}{3} - \frac{4}{9}}{3} = \frac{20}{27}$$

$$m_r = \frac{4}{3} - \frac{\frac{4}{3} - \frac{4}{9}}{3} = \frac{28}{27}$$

Значение справа меньше, чем значение слева:

$$f(m_l) = f\left(\frac{20}{27}\right) \approx -1.81578$$

$$f(m_r) = f\left(\frac{28}{27}\right) \approx -1.99583$$

Уменьшаем отрезок до $\left[\frac{20}{27}; \frac{4}{3}\right]$.

4. Текущие границы: $l = \frac{20}{27}$, $r = \frac{4}{3}$. Середины:

$$m_l = \frac{20}{27} + \frac{\frac{4}{3} - \frac{20}{27}}{3} = \frac{76}{81}$$

$$m_r = \frac{4}{3} - \frac{\frac{4}{3} - \frac{20}{27}}{3} = \frac{92}{81}$$

Значение слева меньше, чем значение справа:

$$f(m_l) = f\left(\frac{76}{81}\right) \approx -1.98880$$

$$f(m_r) = f\left(\frac{92}{81}\right) \approx -1.94217$$

Уменьшаем отрезок до $\left[\frac{20}{27}; \frac{92}{81}\right]$. Расстояние отрезка равно $\frac{32}{81} < 0.5$. Ответ: $\frac{76}{81}$ с точностью не менее 0.5. Точный ответ равен 1.

Псевдокод.

```
l := a
r := b
while |r - l| > eps
    ml := l + (r - l) / 3
    mr := r - (r - l) / 3
    if f(ml) < f(mr)
        l := ml
    else
        r := mr
return (l + r) / 2
```

Сходимость. Заметим, что после каждой итерации, область поиска кореня уменьшается в $\frac{3}{2}$ раза. Значит, количество итераций можно определить из неравенства:

$$(b-a)\left(\frac{2}{3}\right)^n \le \varepsilon.$$

Откуда легко найти количество итераций:

$$n \ge \left[\log_{\frac{2}{3}} \frac{b-a}{\varepsilon}\right].$$

Скорость сходимости «линейная» с параметром $\frac{2}{3}$:

$$|x - x_{k+1}| \le \frac{2}{3}|x - x_k|.$$

5.2 «Золотое» сечение

Этот метод является оптимизированным вариантом тернарного поиска. Будем считать, что самая дорогая вычислительная операция — это вычисление значения функции в точке (в тернарном поиске выполняется по 2 таких операции за ход). Возможно уменьшить количество таких вычислений до одного.

Пусть на текущем ходу имеется разбиение $l < m_l < m_r < r$. Если мы переходим к отрезк $[m_l; r]$, то m_r должно играть роль левой средней точки для отрезка $[m_l; r]$. Если мы переходим к отрезк $[l; m_r]$, то m_l должно играть роль правой средней точки для отрезка $[l; m_r]$.

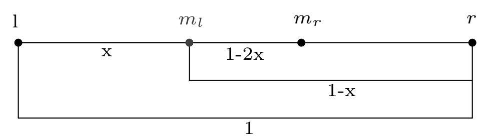

Рисунок 12: Подобное разбиение

Обозначим длину всего отрезка 1. А длину отрезка $[l; m_l]$ равной x. Тогда:

$$\frac{x}{1} = \frac{1-2x}{1-x}.$$

Соответствующее уравнение:

$$x^2 - 3x + 1 = 0.$$

Отбирая только корни из интервала от 0 до $\frac{1}{2}$, получим только один корень. Обозначим его через: $\xi = \frac{3-\sqrt{5}}{2} \approx 0.381966$.

Пример. Найдем минимум функции $f(x) = x^3 - 3x$ на отрезке $[0; 2]$ с точностью $\varepsilon = 0.5$. Обратите внимание на повторные вычисления m_l или m_r на каждом ходу, начиная со второго.

1. Текущие границы: $l = 0$, $r = 2$. Середины:

$$m_l = 0 + \xi * (2 - 0) \approx 0.76393$$

$$m_r = 2 - \xi * (2 - 0) \approx 1.23607$$

Значение справа меньше, чем значение слева:

$$f(m_l) = f(0.76393) \approx -1.81966$$

$$f(m_r) = f(1.23607) \approx -1.84597$$

Уменьшаем отрезок до $[0.76393; 2]$.

2. Текущие границы: $l = 0.76393$, $r = 2$. Середины:

$$m_l = 0.76393 + \xi * (2 - 0.76393) \approx 1.23607$$

$$m_r = 2 - \xi * (2 - 0.76393) \approx 1.52786$$

Значение слева меньше, чем значение слева:

$$f(m_l) = f(1.23607) \approx -1.84597$$

$$f(m_r) = f(1.52786) \approx -1.01699$$

Уменьшаем отрезок до $[0.76393; 1.52786]$.

3. Текущие границы: $l = 0.76393$, $r = 1.52786$. Середины:

$$m_l = 0.76393 + \xi * (1.52786 - 0.76393) \approx 1.05573$$

$$m_r = 1.52786 - \xi * (1.52786 - 0.76393) \approx 1.23607$$

Значение слева меньше, чем значение слева:

$$f(m_l) = f(1.05573) \approx -1.99051$$

$$f(m_r) = f(1.23607) \approx -1.84597$$

Уменьшаем отрезок до $[0.76393; 1.23607]$. Расстояние отрезка равно меньше 0.5. Ответ: 1.0 с точностью не менее 0.5. Точный ответ равен 1.

Разумеется здесь повезло с ответом (в первую очередь из-за того, что он оказался по в центре интервала поиска).

Псевдокод.

```
xi := (3 - sqrt(5)) / 2
l := a
r := b
ml := l + xi * (r - l)
mr := r - xi * (r - l)
fl := f(ml)
fr := f(mr)
while |r - l| > eps
    if fl > fr
        l := ml
        ml := mr
        mr := r - xi * (r - l)
        fml := fmr
        fmr := f(mr)
    else
        r := mr
        mr := ml
        ml := l + xi * (r - l)
        fmr := fml
        fml := f(ml)
return (l + r) / 2
```

Сходимость. Заметим, что после каждой итерации, область поиска кореня уменьшается в до величины $\lambda = \frac{\sqrt{5}-1}{2}$ от исходного размера. Значит, количество итераций можно определить из неравенства:

$$(b-a)\lambda^n \le \varepsilon.$$

Откуда легко найти количество итераций:

$$n \ge \left[\log_\lambda \frac{b-a}{\varepsilon}\right].$$

Скорость сходимости «линейная» с параметром $\frac{2}{3}$:

$$|x - x_{k+1}| \le \lambda |x - x_k|.$$

Стоит отметить, что метод «золотого» сечения имеет 2 премущества: лучшая константа сходимость и меньшее количество вызово функции.

5.3 Метод парабол №1

Рассмотрим функцию $f(x)$ такую, что $f(x)$ дважды дифференцируема на отрезке $[a; b]$ и выпукла вниз. Посмотрим как выглядит Первые 3 слагаемых ряда Тейлора в окрестности некоторой точки x_k:

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k) + \frac{f''(x_k)}{2}(x - x_k)^2.$$

Выражение справа обозначим $g(x)$. Если рассмотривать функцию справа, то ясно что это парабола. Минимум параболы (если он внутренний) определяется из условия:

$$g'(x_{min}) = 0.$$

Тогда:

$$f'(x_k) + f''(x_k)(x_{min} - x_k) = 0.$$

Здесь, стоит отметить, что минимум параболы это только приближение минимума исходной функции. Соответственно получаем итерациооный метод парабол №1:

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

Стоит обратить внимание, что это есть ничто иное, как метод Ньютона для производной, что вполне естественно. Условия выбора начального приближения такие же, как и методе Ньютона: $f'''(x) * f''(x) > 0$, то начинаем справа, иначе — слева.

Пример. Найдем минимум функции $f(x) = x^3 - 3x$ на отрезке $[0; 2]$ с точностью $\varepsilon = 0.5$.

Вычисления:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} = x_k - \frac{3x_k^2 - 3}{6x_k} = \frac{1}{2}\left(x_k + \frac{1}{x_k}\right).$$

Стоит отметить, что полученной соотношение есть формула Герона для приблизительного вычисления квадратного корня.

1. $x_0 = 2$
2. $x_1 = \frac{5}{4}$.
3. $x_2 = \frac{41}{40}$.

   Заметим, что расстояние между 2 последними итерациями меньше 0.5.
   Ответ: $\frac{41}{40}$ с точостью до 0.5.

Псевдокод.

```
m := (r + l) / 2
if f1(m) * f2(m) > 0
    xnew := a
else
    xnew := b
do
    xold := xnew
    xnew := b - f1(xold) / f2(xold)
while |xold - xnew| > eps
```

Сходимость. В литературе можно найти вывод достаточных условий, которые релизованы выше. Скорость сходимости «квадратичная», то есть, существует такой $0 < L < 1$, что:

$$|x - x_{k+1}| \leq L|x - x_k|^2.$$

5.4 Метод парабол №2

Часто, в случае отсутствия возможности вычислить $f'(x)$ и $f''(x)$, производные аппроксимируют численно с помощью малого параметра h:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

В итоге, получим итерациооный метод парабол №2:

$$x_{k+1} = x_k - \frac{h}{2}\frac{f(x_k+h) - f(x_k-h)}{f(x_k+h) - 2f(x_k) + f(x_k-h)}.$$

Псевдокод.

```
m := (r + l) / 2
if f1(m) * f2(m) > 0
    xnew := a
else
    xnew := b
do
    xold := xnew
    xnew := b - (f(xold + h) - f(xold - h)) /
     (f(xold + h) - 2 * f(xold) + f(xold - h)) * h / 2
while |xold - xnew| > eps
```

Обратите внимание, что порядок множителей (умножение на h в начале или в конце) может сильно изменить результат, за счет погрешности вычислительных округлений. В таких ситуациях стоит придерживаться правила: лучше всего считать вещественные числа в районе ± 1, где их плотность наилучшая.

Список литературы

[1] Калиткин Н.Н. Численные методы. - Спб.: БХВ-Петербург, 2014.

[2] Самарский А.А., Гулин А.В. Численные методы. - М.: Наука, 1989.