Приближённое решение нелинейных уравнений

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Математико-механический факультет

2020

Содержание I

1. Отделение корней
2. Уточнение корней
   - Метод Ньютона (метод касательных)
   - Метод секущих
   - Метод хорд
   - Метод простой итерации (метод последовательных приближений)
3. Задание 1
4. Задание 2

Приближённое решение нелинейных уравнений I

Дано уравнение

$$f(x) = 0, \tag{1}$$

где функция $f(x)$ определена и непрерывна на некотором конечном или бесконечном интервале $a < x < b$.
Всякое значение x^* такое, что $f(x^*) = 0$, называется корнем уравнения (1) или нулем функции $f(x)$.
Приближенное нахождение изолированных действительных корней уравнения складывается из двух этапов

1. Отделение корней, т. е. установление промежутков $[\alpha, \beta]$, в которых содержится один и только один корень уравнения (1).
   Использование этих промежутков для определения начальных приближений к корням.
2. Уточнение приближенных корней.

Отделение корней

Для отделения корней следует построить таблицу значений функции или график функции, найти промежутки, на концах которых функция $f(x)$ имеет разные знаки.
Тогда внутри этих промежутков содержится по крайней мере один корень уравнения $f(x) = 0$.
Нужно тем или иным образом убедиться, что данный корень является единственным.
Для уменьшения длин промежутков может быть использован метод половинного деления (бисекции).

Метод бисекции

Полагаем $[a_0, b_0] = [a, b]$.
Пусть $c_0 = (a_0 + b_0)/2$.
Далее строим последовательность промежутков $\{[a_k, b_k]\},\; k = 1,\, 2, \ldots$

$$[a_k, b_k] = \begin{cases} [a_{k-1}, c_{k-1}], & \text{если } f(a_{k-1}) \cdot f(c_{k-1}) < 0, \\ [c_{k-1}, b_{k-1}], & \text{если } f(c_{k-1}) \cdot f(b_{k-1}) < 0. \end{cases}$$

Здесь на каждом шаге длина промежутка уменьшается вдвое, так что

$$b_k - a_k = (b - a)/2^k.$$

Уточнение корней I

Для уточнения корней используются итерационные методы.
При решении задачи итерационными методами важно следующее:

- расчетная формула;
- условие сходимости;
- порядок сходимости (скорость сходимости): число $\alpha \geqslant 1$ называют порядком сходимости последовательности $\{x_k\}$ к x^*, если $x_k \to x^*$ и существует постоянная C, такая что при всех k

$$|x_k - x^*| \leqslant C\,|x_{k-1} - x^*|^{\alpha}; \tag{2}$$

Уточнение корней II

- получение решения с заданной точностью ε, т. е. критерий окончания.
  Здесь имеется в виду нахождение x_k такого, что выполняется условие $|x_k - x^*| < \varepsilon$.

Оценки погрешности

Существуют априорные и апостериорные оценки для фактической погрешности $|x_k - x^*|$.
Априорные оценки часто бывают сильно завышены.
Легко показать, что для оценки точности приближения x_k любого итерационного метода в предположении, что $f'(x) \neq 0$, можно воспользоваться неравенством

$$|x_k - x^*| \leqslant \frac{|f(x_k)|}{m_1}, \tag{3}$$

где $m_1 = \min |f'(x)|$ при $a \leqslant x \leqslant b$.

Метод Ньютона (метод касательных)

Дано уравнение (1).
Пусть функция $f(x)$ — вещественная и находим вещественный корень x^*.
Будем предполагать, что на отрезке $[a, b]$ таком, что $f(a)f(b) < 0$, $f(x)$ дважды непрерывно дифференцируема и $f'(x), f''(x)$ — знакопостоянны.
Выбираем $x_0 \in [a, b]$. Заменим уравнение в окрестности x_0 приближённым уравнением

$$f(x_0) + f'(x_0)(x - x_0) = 0,$$

левая часть которого есть линейная часть разложения функции $f(x)$ в ряд Тейлора в окрестности точки x_0.

Отсюда

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

Действуя аналогично, получаем расчетную формулу метода Ньютона

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})}, \; k = 1, 2, \ldots . \tag{4}$$

Метод Ньютона имеет простой геометрический смысл: x_k есть абсцисса точки пересечения касательной к графику функции $f(x)$, построенной в точке $(x_{k-1}, f(x_{k-1}))$, с осью абсцисс.

Теорема о сходимости

Теорема 1 (о сходимости)

Если

1. $f(a)f(b) < 0$,
2. $f'(x),\ \ f''(x)$ *сохраняют определенные знаки при* $x \in [a,b]$),
3. $f(x_0)f''(x_0) > 0,\ \ x_0 \in [a,b]$,

то $x_k \to x^*$, *причем скорость сходимости определяется неравенством*

$$|x_k - x^*| \leqslant \frac{M_2}{2m_1}(x_{k-1} - x^*)^2. \tag{5}$$

Здесь

$$m_1 = \min_{x\in[a,b]} |f'(x)|,\ M_2 = \max_{x\in[a,b]} |f''(x)|. \tag{6}$$

Если $f(x_0)\, f''(x_0) < 0$, то можно не прийти к $x = x^*$, если x_0 не очень хорошее.
Так как метод Ньютона имеет второй порядок сходимости, то можно пользоваться следующим критерием оценки погрешности: если

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

то

$$|x_k - x^*| < \varepsilon.$$

Замечание 1

Если $f'(x^*) = 0$, то квадратичной сходимости может и не быть.
Например, пусть $f(x) = x^2$, корень $x^* = 0$ — корень второй кратности, расчетная формула имеет вид: $x_{k+1} = x_k/2$ и сходимость линейная (если $x_0 \neq 0$).
Второго порядка сходимости для корня кратности p можно достичь, применяя расчетную формулу вида

$$x_k = x_{k-1} - p\,\frac{f(x_{k-1})}{f'(x_{k-1})}, \quad k = 1,\, 2,\, \ldots \tag{7}$$

Замечание 2

Иногда целесообразно применять модифицированный метод Ньютона с расчетной формулой

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_0)}, \; k = 1, 2, \ldots \tag{8}$$

Скорость сходимости модифицированного метода значительно меньше.

Замечание 3

Приведем расчетную формулу метода третьего порядка сходимости:

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})} - \frac{f^2(x_{k-1})\, f''(x_{k-1})}{2\,(f'(x_{k-1}))^3}, \quad k = 1,\, 2,\, \ldots \qquad (9)$$

Метод секущих I

Заменяя производную в расчетной формуле метода Ньютона её приближенным значением по формулам численного дифференцирования, получаем расчетную формулу

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_{k-1})}(x_k - x_{k-1}), \quad k = 1,\, 2, \ldots . \quad (10)$$

Порядок сходимости метода секущих определяется неравенством

$$|x_k - x^*| \leqslant \frac{M_2}{2m_1}(x_{k-1} - x^*)^\alpha, \quad (11)$$

где $\alpha = \dfrac{\sqrt{5}+1}{2} \approx 1.618$, а m_1 и M_2 имеют определенный в (6) смысл.

Метод секущих II

В методе секущих можно пользоваться критерием: если $|x_k - x_{k-1}| < \varepsilon$, то $|x_k - x^*| < \varepsilon$.

Метод хорд I

Пусть известен промежуток $[a, b]$, такой что $f(a) \cdot f(b) < 0$ и $f''(x) > 0$. Рассмотрим два возможных случая.

1. $f(a) < 0$, соответственно $f(b) > 0$. В этом случае конец b неподвижен и последовательные приближения при $x_0 = a$

$$x_{k+1} = x_k - \frac{f(x_k)}{f(b) - f(x_k)}(b - x_k), \quad k = 0,\ 1, \ldots . \tag{12}$$

образуют монотонно возрастающую последовательность, причем

$$a = x_0 < x_1 < x_2 < \ldots < x_k < x_{k+1} < \ldots < x^* < b.$$

Метод хорд II

2. $f(a) > 0$, соответственно $f(b) < 0$. В этом случае конец a неподвижен и последовательные приближения при $x_0 = b$

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(a)}(x_k - a), \quad k = 0, 1, \ldots. \quad (13)$$

образуют монотонно убывающую последовательность, причем

$$a < x^* < \ldots < x_{k+1} < x_k < \ldots < x_1 < x_0 = b.$$

Метод хорд III

Пределы этих последовательностей x^* существуют, так как они ограничены и монотонны.
Для оценки точности можно воспользоваться уже известным неравенством (3)

$$|x_k - x^*| \leqslant \frac{|f(x_k)|}{m_1},$$

и

$$|x_k - x^*| \leqslant \frac{M_1 - m_1}{m_1} |x_k - x_{k-1}|, \qquad (14)$$

где $m_1 = \min |f'(x)|, \;\; M_1 = \max |f'(x)|$ при $a \leqslant x \leqslant b.$
Геометрически метод эквивалентен замене кривой $y = f(x)$ хордами, проходящими через точки $(x_k, f(x_k))$, $(x_{k+1}, f(x_{k+1}))$, k=0, 1,. . .

Метод хорд IV

Порядок метода — первый и нельзя пользоваться в качестве критерия модулем разности двух соседних приближений.

Метод простой итерации (метод последовательных приближений)

Заменим уравнение (1) равносильным уравнением

$$x = \varphi(x), \tag{15}$$

где $\varphi(x)$ — непрерывна.
Расчетная формула метода

$$x_k = \varphi(x_{k-1}), \quad k = 1, 2, \ldots \tag{16}$$

Если эта последовательность сходящаяся, то $\lim\limits_{k\to\infty} x_k = x^*$.

Условие сходимости

Кратко сформулируем условие сходимости.
Пусть в некоторой окрестности (a, b) корня x^* уравнения (15) производная $\varphi'(x)$ сохраняет постоянный знак и выполнено неравенство

$$|\varphi'(x)| \leqslant q < 1. \tag{17}$$

Тогда, если производная $\varphi'(x)$ положительна, то последовательные приближения (16) $(x_0 \in (a, b))$ сходятся к корню x^* монотонно,
если производная $\varphi'(x)$ отрицательна, то последовательные приближения колеблются около корня x^*.

Оценки погрешности

Априорная оценка погрешности

$$|x^* - x_k| \leqslant \frac{q^k}{1-q}|x_1 - x_0|. \tag{18}$$

Апостериорная оценка погрешности

$$|x^* - x_k| \leqslant \frac{q}{1-q}|x_k - x_{k-1}|. \tag{19}$$

Замечание 4

Как показывает оценка (19), ошибочно было бы пользоваться в качестве критерия получения решения с заданной точностью ε совпадения x_k и x_{k-1} с точностью ε.

Замечание 5

Напомним, что приводить уравнение вида (1) к виду (15) следует так чтобы $|\varphi'(x)| \leqslant q < 1$, причем, чем меньше число q, тем быстрее, вообще говоря, последовательные приближения сходятся к корню x^*.
Укажем один достаточно общий прием приведения. Пусть искомый корень x^* уравнения лежит на отрезке $[a, b]$, причем

$$0 < m_1 \leqslant f'(x) \leqslant M_1$$

при $a \leqslant x \leqslant b$. Заменим уравнение (1) эквивалентным ему уравнением

$$x = x - \lambda f(x) \ \ (\lambda > 0).$$

Из условия сходимости получаем, что можно взять $\lambda = \dfrac{1}{M_1}$ и тогда

$$q = 1 - \frac{m_1}{M_1} < 1.$$

Замечание 6

Формулу (4) метода Ньютона можно рассматривать как формулу метода итераций для уравнения $x = \varphi(x)$, где

$$\varphi(x) = x - f(x)/f'(x).$$

Легко проверить, что $\varphi'(x^*) = 0$.
Поэтому следует ожидать квадратичную сходимость метода.

Задание 1 I

Дано уравнение $f(x) = 0$.
Требуется

1. Отделить все корни или корни на указанном интервале.
2. Сузить интервалы, определенные выше, в несколько раз, используя метод половинного деления.
3. Вычислить корни методом Ньютона (или модифицированным) с точностью $\varepsilon = 0.000001$. Эти значения корней далее будем считать "точными", ранее и далее в таблице они обозначены x^*.

Задание 1 II

4. Используя интервалы из первого или второго пункта, найти требуемые корни с точностью $\varepsilon = 0.0001$ методом секущих. В качестве критерия использовать модуль разности между двумя соседними приближениями. Сравнить с фактической погрешностью.
5. Используя интервалы из первого или второго пункта, найти требуемые корни с точностью $\varepsilon = 0.001$ методом хорд. В качестве критерия использовать оценку (14). Сравнить с фактической погрешностью.
6. Вычислить корни методом итераций с точностью $\varepsilon = 0.00001$, выбрав в качестве x_0 то же значение, что и в методе Ньютона.
7. Сравнить результаты, количество итераций.

Для численной реализации методов должны быть созданы подпрограммы с параметрами:

- x_0 — нулевое приближение к корню (в методе Ньютона и в методе итераций);
- ε — заданная точность;
- kmax — максимальное количество итераций (для исключения зацикливания).

Подпрограмма должна возвращать либо x_k, такое что $|x_k - x^*| < \varepsilon$, либо $x_{k\max}$.

Результаты методов уточнения оформить в виде таблицы 1.

Таблица 1

k	x_k	$x_k - x_{k-1}$	$x_k - x^*$	$f(x_k)$
0		—		
1				
...	...	...	...	...

Задание 2

Дано уравнение $P_n(x) = 0$, где

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n \tag{20}$$

— многочлен Лежандра, корни которого являются узлами квадратурной формулы Гаусса.
Требуется, используя метод Ньютона, вычислить указанный по номеру (по возрастанию) корень многочлена Лежандра с заданной точностью ε.

Для реализации метода Ньютона должна быть создана подпрограмма с параметрами:

- x_0 — нулевое приближение к корню;
- ε — заданная точность;
- kmax — максимальное количество итераций (для исключения зацикливания);
- x^* — "точное" значение корня[1].

Подпрограмма должна возвращать либо x_k такое, что $|x_k - x^*| < \varepsilon$, либо $x_{k\max}$ и количество итераций.

[1] "Точное" значение корня можно найти или встроенной функцией или методом Ньютона с $\varepsilon = 10^{-15}$.

Результаты оформить в виде таблицы 2.

Таблица 2

k	x_k	$x_k - x_{k-1}$	$x_k - x^*$	$P_n(x_k)$
0		—		
1				
...	...	...	...	...

Замечание 7

Нулевое приближение к корню удобно вычислять по формуле

$$x_0(i) = -\cos\left(\frac{i - 1/4}{n + 1/2}\,\pi\right). \tag{21}$$

Здесь i — номер корня, n — степень многочлена Лежандра.

Свойства многочлена Лежандра I

Многочлен Лежандра (20) является частным случаем многочлена Якоби $P_n^{(\alpha,\beta)}(x)$ при $\alpha = \beta = 0$. Приведем необходимые сведения об этих многочленах.
Нам потребуется частный случай многочленов, когда $\alpha = \beta = k$ — целое число.

1. $P_0^{(k,k)}(x) = 1, \quad P_1^{(k,k)}(x) = (k+1)\,x.$
2. Три последовательных многочлена Якоби связаны рекуррентной формулой: $P_{n+2}^{(k,k)}(x) =$
$$\frac{(n+k+2)(2n+2k+3)x \cdot P_{n+1}^{(k,k)}(x) - (n+k+2)(n+k+1)P_n^{(k,k)}(x)}{(n+2k+2)(n+2)},$$
$n = 0,\ 1,\ 2,\ \ldots$
3. Правила дифференцирования многочленов Якоби:
$$\left[P_n^{(k,k)}(x)\right]' = \frac{n+2k+1}{2}\,P_{n-1}^{(k+1,k+1)}(x), \quad n \geqslant 1.$$

Для вычисления значения многочлена и его производной в точке предпочтительнее использовать вышеприведенную рекуррентную формулу, так как счет многочлена через его коэффициенты неустойчив при больших значениях n.
Как видно, в данном случае понадобится вычислять не только $P_n^{(0,0)}(x)$, но и $P_{n-1}^{(1,1)}(x)$.

В связи с этим для реализации метода потребуется создать подпрограмму с параметрами:

- n — степень многочлена;
- k — верхний индекс у многочлена Якоби;
- x — точка, в которой вычисляется значение многочлена.

эта программа на языке MATLAB может иметь, например, следующее содержание:

```
function tmp=Precx(n,k,x)
tmp0=1;
tmp1=(k+1)*x;
tmp=tmp1;
    for i=1:n-1
        tmp=((i+k+2)*(2*i+2*k+3)*x*tmp1-
            (i+k+2)*(i+k+1)*tmp0)/((i+2*k+2)*(i+2));
        tmp0=tmp1;
        tmp1=tmp;
    end
end
```

```
%Использование встроенных функций MATLAB
syms x;
n=3
num=3
%Используется встроенная функция для построения
%полинома Лежандра
P=vpa(collect(legendreP(n,x)))
%Используется встроенная функция для построения
%полинома Якоби
Pj=vpa(collect(jacobiP(n,k1,k2,x)
dP = diff(P) %дифференцирование полинома Лежандра
%Вычисление и сортировка по возрастанию
%всех корней полинома
Xsolve=sort(vpa(solve(P,x)))
%Извлечение корня по заданному порядковому номеру
xexact=Xsolve(num)
```