Метод бисекции или *метод деления отрезка пополам* — простейший численный метод для решения нелинейных уравнений вида f(x)=0. Предполагается только непрерывность функции f(x). Поиск основывается на теореме о промежуточных значениях.

Обоснование

Алгоритм основывается на следующем следствии теоремы Больцано — Коши:

> Пусть функция $f(x) \in C([a,\ b])$, тогда если $f(a) > 0,\ f(b) < 0$, то $\exists c \in [a,\ b]:\ f(c) = 0$.

Таким образом, если мы ищем ноль, то на концах отрезка функция должна быть разных знаков. Разделим отрезок пополам и возьмём ту из половинок, для которой на концах функция по-прежнему принимает значения разных знаков. Если серединная точка оказалось искомым нулём, то процесс завершается.

Если задана точность вычисления ε, то процедуру следует продолжать до тех пор, пока длина отрезка не станет меньше ε.

Для поиска произвольного значения достаточно вычесть из значения функции искомое значение и искать ноль получившейся функции.

Описание алгоритма

Задача заключается в нахождении корней нелинейного уравнения

$$f(x) = 0. \qquad (1)$$

Для начала итераций необходимо знать интервал $[x_L, x_R]$ значений x, на концах которого функция принимает значения разных знаков:

$$f(x_L)f(x_R) < 0. \qquad (2)$$

Из непрерывности функции f и условия (2) следует, что на интервале $[x_L, x_R]$ существует хотя бы один корень уравнения (в случае наличия нескольких корней метод приводит к нахождению одного из них)

Выберем точку внутри интервала

$$x_M = (x_R + x_L)/2. \qquad (3)$$

Если $f(x_M) = 0$, то корень найден. Если $f(x_M) \neq 0$ разобьём интервал $[x_L, x_R]$ на два: $[x_L, x_M]$ и $[x_M, x_R]$. Теперь найдём новый интервал, в котором функция меняет знак. Пусть $f(x_L)f(x_M) < 0$ и соответственно корень находится внутри интервала $[x_L, x_M]$. Тогда обозначим $x_R = x_M$ и повторим описанную процедуру до достижения требуемой точности. За количество итераций N первоначальный отрезок делится в 2^N раз.

Поиск значения монотонной функции

метод половинного деления

Поиск значения монотонной функции, записанной в массиве, заключается в сравнении срединного элемента массива с искомым значением, и повторением алгоритма для той или другой половины, в зависимости от результата сравнения.

Пускай переменные L_b и U_b содержат, соответственно, левую и правую границы отрезка массива, где находится нужный нам элемент. Исследования начинаются со среднего элемента отрезка. Если искомое значение меньше среднего элемента, осуществляется переход к поиску в верхней половине отрезка, где все элементы меньше только что проверенного, то есть значением U_b становится $(M-1)$и на следующей итерации исследуется только половина массива. Т.о., в результате каждой проверки область поиска сужается вдвое.

Например, если длина массива равна 1023, после первого сравнения область сужается до 511 элементов, а после второй — до 255. Т.о. для поиска в массиве из 1023 элементов достаточно 10 сравнений.

```
l = леваяГраница - 1
r = праваяГраница + 1
while (r - l > 1) {
  середина = (l+r) / 2
  if (массив[середина] < значение)
     l = середина
  else
     r = середина
 }
 if (массив[r-1] ≠ значение)
   return -1 // элемент не найден
 else
   return r
```

Программный код

- x_n – начало отрезка по x;
- x_k – конец отрезка по x;
- x_i – середина отрезка по x;
- eps – требуемая точность вычислений.

Таким образом, весь алгоритм можно записать следующим образом (в псевдокоде):

1. Начало.
2. Ввод x_n, x_k, eps.
3. Если $F(x_n) = 0$, то Вывод (корень уравнения – x_n).
4. Если $F(x_k) = 0$, то Вывод (корень уравнения – x_k).
5. Пока ($F(x_i) <> 0$) и $|x_k - x_n| >$ eps повторять:
6. $xi := (x_k+x_n)/2$;
7. если $(F(x_n)*F(x_i) < 0)$, то $x_k := x_i$;
8. если $(F(x_i)*F(x_k) < 0)$, то $x_n := x_i$.
9. Вывод (Найден корень уравнения – x_i точности ε).
10. Конец.

Пример реализации алгоритма на языке Matlab

Метод простой итерации

В основе метода заложено понятие сжимающего отображения. Определим терминологию:

Говорят, что функция φ осуществляет ***сжимающее отображение*** на $[a,\ b]$, если

1. $\forall x \in [a,\ b] : \varphi(x) \in [a,\ b]$
2. $\exists \alpha < 1 : \forall x_1, x_2 \in [a,\ b] \quad ||\varphi(x_1) - \varphi(x_2)|| \le \alpha ||x_1 - x_2||$

Тогда основная теорема будет выглядеть так:

Теорема Банаха (принцип сжимающих отображений).
Если φ— сжимающее отображение на $[a,\ b]$, то:

1. у $x = \varphi(x) \quad \exists! x^* \in [a,\ b]$— корень;
2. итерационная последовательность $x_{i+1} = \varphi(x_i)$ сходится к этому корню;
3. для очередного члена x_n справедливо $||x_n - x^*|| \le \dfrac{\alpha^n ||x_1 - x_0||}{1 - \alpha}$.

Поясним смысл параметра α. Согласно теореме Лагранжа имеем:

$$\varphi(x) \in C^1[a,\ b]. \quad \forall x_1, x_2 \in (a,\ b), \quad x_1 < x_2 \quad \exists \xi \in (x_1,\ x_2) : \quad \varphi'(\xi)(x_2$$

Отсюда следует, что $\alpha \approx |\varphi'(\xi)|$. Таким образом, для сходимости метода достаточно, чтобы $\forall x \in [a,\ b] \quad |\varphi'(x)| \le 1$.

1. $x_3 = \varphi(x_2)$

....

и так далее, пока $||x_{i+1} - x_i|| > \varepsilon$

Применительно к СЛАУ

Рассмотрим систему:

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n & = & b_1 \\ & \ldots & \\ a_{n1}x_1 + \ldots + a_{nn}x_n & = & b_n \end{cases}$$

Для неё итерационное вычисление будет выглядеть так:

метод простой итерации

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}^{i+1} = \begin{pmatrix} a_{11}+1 & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22}+1 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn}+1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}^{i} - \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Сходимость метода будет осуществлять $\left\| \begin{matrix} a_{11}+1 & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn}+1 \end{matrix} \right\| < 1$

Следует отметить, что для оценки сходимости вычисляется не определитель матрицы, а норма матрицы. Поэтому в данном случае поставлены двойные вертикальные черты, а не одинарные.

Алгоритм

1. Условие $f(x) = 0$ преобразуется к виду $x = \varphi(x)$, где $\varphi(x)$ — сжимающая
2. Задаётся начальное приближение и точность $x_0, \quad \varepsilon, \quad i = 0$
3. Вычисляется очередная итерация $x_{i+1} = \varphi(x_i)$
   - Если $||x_{i+1} - x_i|| > \varepsilon$, то $i = i + 1$ и возврат к шагу 3.
   - Иначе $x = x_{i+1}+$ и остановка.

[

Одномерный случай

Для того, чтобы решить уравнение $f(x) = 0$, пользуясь методом простой итерации, необходимо привести его к виду $x = \varphi(x)$, где φ — сжимающее отображение. Чтобы отображение было наиболее эффективно, необходимо, чтобы в точке очередной итерации x^* выполнялось $\varphi'(x^*) = 0$. Будем искать решение данного уравнения в виде $\varphi(x) = x + \alpha(x)f(x)$, тогда:

$$\varphi'(x^*) = 1 + \alpha'(x^*)f(x^*) + \alpha(x^*)f'(x^*) = 0$$

Воспользуемся тем, что $f(x) = 0$, и получим окончательную формулу для $\alpha(x)$:

$$\alpha(x) = -\frac{1}{f'(x)}$$

С учётом этого сжимающая функция примет вид:

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

метод простых итераций

Тогда алгоритм нахождения численного решения уравнения $f(x) = 0$ сводится к итерационной процедуре вычисления:

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

Многомерный случай

Обобщим полученный результат на многомерный случай.

Выбирая некоторое начальное приближение $\vec{x}^{[0]}$, находят последовательнь е приближения $\vec{x}^{[j+1]}$ путем решения систем уравнений:

$$f_i + \sum_{k=1}^{n} \frac{\partial f_i}{\partial x_k}(x_k^{[j+1]} - x_k^{[j]}) = 0, \quad i = 1, 2, \ldots, n$$

,

где $x^{[j]} = \left(x_1^{[j]} \ldots x_k^{[j]} \ldots x_n^{[j]}\right), \quad j = 0, 1, 2, \ldots$.

Литература

1. *Амосов А.А., Дубинский Ю. А., Копченова Н.П.* Вычислительные методы для инженеров. — М.: Мир, 1998.
2. *Бахвалов Н.С., Жидков Н.П., Кобельков Г.Г.* Численные методы. — 8-е изд.. — М.: Лаборатория Базовых Знаний, 2000.
3. *Волков Е.А.* Численные методы. — М.: Физматлит, 2003.
4. *Коршунов Ю.М., Коршунов Ю.М.* Математические основы кибернетики. — М.: Энергоатомиздат, 1972.
5. *Калиткин Н.Н.* Численные методы. — М.: Наука, 1978.

Метод Ньютона, **алгоритм Ньютона** (также известный как **метод касательных**) — это итерационный численный метод нахождения корня (нуля) заданной функции. Метод был впервые предложен английским физиком, математиком и астрономом Исааком Ньютоном (1643—1727), под именем которого и обрёл свою известность. Поиск решения осуществляется путём построения последовательных приближений и основан на принципах простой итерации. Метод обладает квадратичной сходимостью. Улучшением метода является метод хорд и касательных. Также метод Ньютона может быть использован для решения задач оптимизации, в которых требуется определить нуль первой производной либо градиента в случае многомерного пространства. [править]
Описание метода

Обоснование

Чтобы численно решить уравнение $f(x) = 0$ методом простой итерации, его необходимо привести к следующей форме: $x = \varphi(x)$, где φ — сжимающее отображение.

Для наилучшей сходимости метода в точке очередного приближения x^* должно выполняться условие $\varphi'(x^*) = 0$. Решение данного уравнения ищут в виде $\varphi(x) = x + \alpha(x)f(x)$, тогда:

$$\varphi'(x^*) = 1 + \alpha'(x^*)f(x^*) + \alpha(x^*)f'(x^*) = 0.$$

В предположении, что точка приближения «достаточно близка» к корню $\tilde{x}$, и что заданная функция непрерывна $(f(x^*) \approx f(\tilde{x}) = 0)$, окончательная формула для $\alpha(x)$ такова:

$$\alpha(x) = -\frac{1}{f'(x)}.$$

С учётом этого функция $\varphi(x)$ определяется выражением:

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

Эта функция в окрестности корня осуществляет сжимающее отображение[1], и алгоритм нахождения численного решения уравнения $f(x) = 0$ сводится к итерационной процедуре вычисления:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

По теореме Банаха последовательность приближений стремится к корню уравнения $f(x) = 0$.

f' и f'' - непрер, сохраняю определенные знаки на $[a; b]$

2) $f(a) \cdot f(b) < 0$

3) $f(x_0) \cdot f''(x_0) > 0$

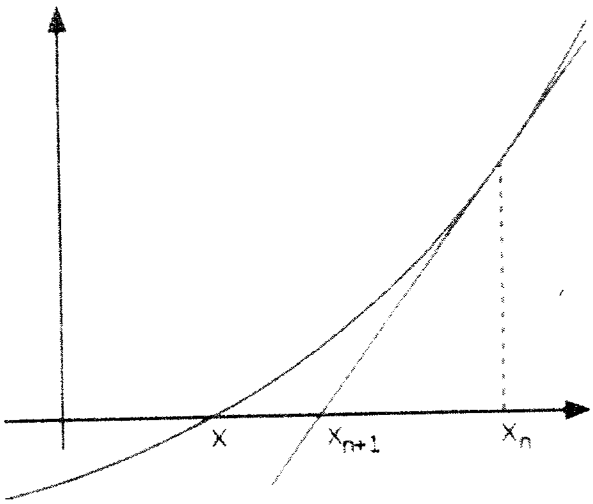

Иллюстрация метода Ньютона (синим изображена функция $f(x)$, нуль которой необходимо найти, красным — касательная в точке очередного приближения x_n). Здесь мы можем увидеть, что последующее приближение x_{n+1}лучше предыдущего x_n.

Геометрическая интерпретация

Основная идея метода заключается в следующем: задаётся начальное приближение вблизи предположительного корня, после чего строится касательная к исследуемой функции в точке приближения, для которой находится пересечение с осью абсцисс. Эта точка и берётся в качестве следующего приближения. И так далее, пока не будет достигнута необходимая точность.

Пусть $f(x)\colon [a,\ b] \to \mathbb{R}$ — определённая на отрезке $[a,\ b]$и дифференцируемая на нём вещественнозначная функция. Тогда формула итеративного исчисления приближений может быть выведена следующим образом:

$$f'(x_n) = \operatorname{tg}\alpha = \frac{\Delta y}{\Delta x} = \frac{f(x_n) - 0}{x_n - x_{n+1}} = \frac{0 - f(x_n)}{x_{n+1} - x_n},$$

где α — угол наклона касательной в точке x_n.

Следовательно искомое выражение для x_{n+1}имеет вид:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Итерационный процесс начинается с некоего начального приближения x_0 (чем ближе к нулю, тем лучше, но если предположения о нахождении решения отсутствуют, методом проб и ошибок можно сузить область возможных значений, применив теорему о промежуточных значениях).

Алгоритм

1. Задается начальное приближение x_0.

Метод Ньютона

2. Пока не выполнено условие остановки, в качестве которого можно взять $|x_{n+1} - x_n| < \varepsilon$ или $|f(x_{n+1})| < \varepsilon$ (то есть погрешность в нужных пределах), вычисляют новое приближение: $$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Пример

Иллюстрация применения метода Ньютона к функции $f(x) = \cos x - x^3$ с начальным приближением в

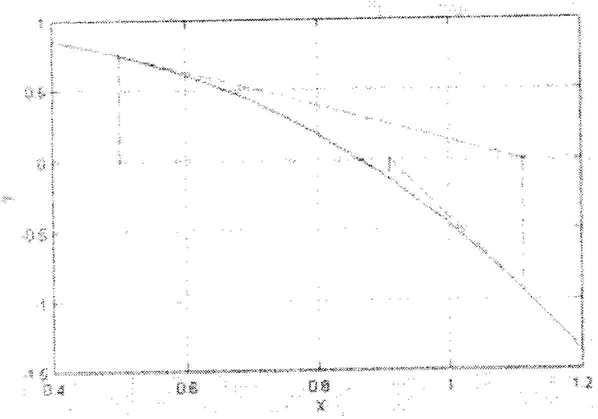

График последовательных приближений.

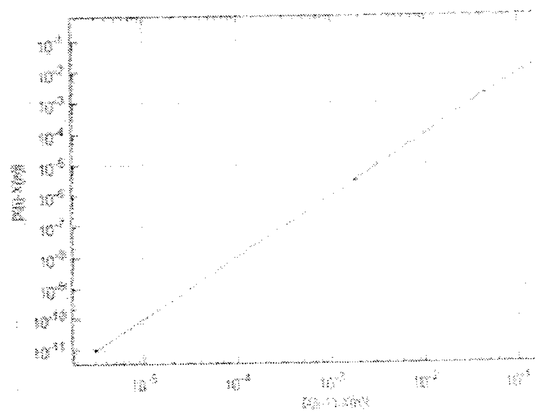

График сходимости.

Согласно способу практического определения скорость сходимости может быть оценена как тангенс угла наклона г есть в данном случае равна двум.

Рассмотрим задачу о нахождении положительных x, для которых $\cos x = x^3$. Эта задача может быть представлена как задача нахождения нуля функции $f(x) = \cos x - x^3$. Имеем выражение для производной $f'(x) = -\sin x - 3x^2$. Так как $\cos x \leqslant 1$ для всех x и $x^3 > 1$ для $x > 1$, очевидно, что решение лежит между 0 и 1. Возьмём в качестве начального приближения значение $x_0 = 0{,}5$, тогда:

$$\begin{aligned}
x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} &= 1{,}112\,141\,637\,097.\\
x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} &= \underline{0{,}9}09\,672\,693\,736.\\
x_3 &= x_2 - \frac{f(x_2)}{f'(x_2)} &= \underline{0{,}86}7\,263\,818\,209.\\
x_4 &= x_3 - \frac{f(x_3)}{f'(x_3)} &= \underline{0{,}865\,47}7\,135\,298.\\
x_5 &= x_4 - \frac{f(x_4)}{f'(x_4)} &= \underline{0{,}865\,474\,033\,1}11.\\
x_6 &= x_5 - \frac{f(x_5)}{f'(x_5)} &= \underline{0{,}865\,474\,033\,102}.
\end{aligned}$$

Подчёркиванием отмечены верные значащие цифры. Видно, что их количество от шага к шагу растёт (приблизительно удваиваясь с каждым шагом): от 1 к 2, от 2 к 5, от 5 к 10, иллюстрируя квадратичную скорость сходимости.

Метод Ньютона 4

Условия применения

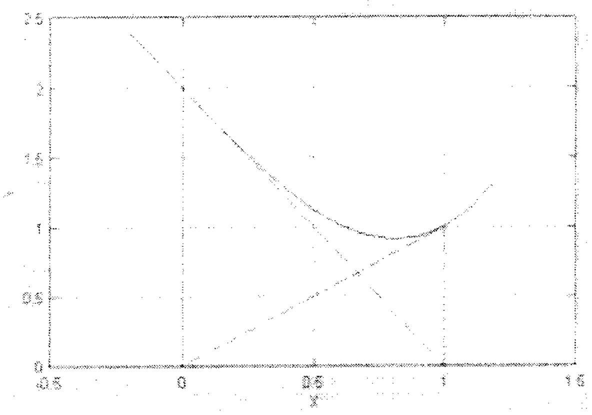

Иллюстрация расхождения метода Ньютона, применённого к функции $f(x)=x^3-2x+2$ с начальным приближением в точке $x_0=0$.

Рассмотрим ряд примеров, указывающих на недостатки метода.

Контрпримеры

- Если начальное приближение недостаточно близко к решению, то метод может не сойтись.

Пусть

$$f(x)=x^3-2x+2.$$

Тогда

$$x_{n+1}=x_n-\frac{x_n^3-2x_n+2}{3x_n^2-2}.$$

Возьмём нуль в качестве начального приближения. Первая итерация даст в качестве приближения единицу. В свою очередь, вторая снова даст нуль. Метод зациклится и решение не будет найдено. В общем случае построение последовательности приближений может быть очень запутанным.