Отчет

Шилов Максим

Постановка задачи

Пусть на отрезке $[a, b]$ задана непрерывная функция $f(x)$ и требуется решить уравнение

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

Любое решение $x^* \in [a, b]$ этого уравнения будем называть корнем (нулем) функции $f(x)$. Отметим, что каких-либо общих правил анализа расположения корней произвольной функции $f(x)$ на отрезке $[a, b]$ не существует.

Методы решения

1. Метод деления пополам (бисекций)

Пусть дано уравнение (1), в котором функция $f(x)$ непрерывна на $[a, b]$ и $f(a)f(b) < 0$. Для нахождения корня уравнения (1), принадлежащего отрезку $[a, b]$, делим этот отрезок пополам. Если $f(\frac{a+b}{2}) = 0$, то $x^* = \frac{a+b}{2}$ является корнем. Если $f(\frac{a+b}{2}) \neq 0$, то выбираем ту из половин $[a, \frac{a+b}{2}]$ или $[\frac{a+b}{2}, b]$, на концах которой функция $f(x)$ имеет противоположные знаки. Новый суженный отрезок обозначим $[a_1, b_1]$. Его снова делим пополам и проводим то же рассуждение и т.д. В результате получаем на каком-то этапе или точный корень уравнения (1), или последовательность вложенных друг в друга отрезков $[a_n, b_n]$ таких, что

$$f(a_n)f(b_n) < 0, \quad n = 1, 2, \dots, n, \dots, \qquad (2)$$

$$b_n - a_n = \frac{1}{2^n}(b - a). \qquad (3)$$

Очевидно, что левые концы a_n образуют монотонную неубывающую, а правые концы b_n - монотонную невозрастающую ограниченные последовательности. Поэтому каждая их таких последовательностей имеет предел, а из равенства (3) следует, что

$$x^* = \lim_{n\to\infty} b_n = \lim_{n\to\infty}(a_n + \frac{1}{2^n}(b - a)) = \lim_{n\to\infty} a_n + \lim_{n\to\infty} \frac{1}{2^n}(b - a) = \lim_{n\to\infty} a_n$$

Переходя в формуле (2) к пределу при $n \to \infty$, в силу непрерывности функции $f(x)$ ($f(a_n) \to f(x^*)$ и $f(b_n) \to f(x^*)$ при $n \to \infty$) получаем $(f(x^*))^2 \leq 0$. Отсюда $f(x^*) = 0$, т.е. x^* является корнем.

В качестве приближений к корню x^* уравнения (1) следует выбирать середины $x_n = \frac{a_n+b_n}{2}$ интервалов $[a_n, b_n]$. Так как x_n и x^* принадлежат $[a_n, b_n]$, то для них справедлива оценка $|x_n - x^*| \leq b_n - a_n$, позваляющая оценить точность получаемых приближений:

$$|x_n - x^*| \leq b_n - a_n = \frac{1}{2^n}(b - a)$$

2. Метод простой итерации

Пусть известно, что корень x^* уравнения (1) лежит на отрезке $G = [a, b]$.

1. Уравнение (1) равносильным преобразованием привести к виду $x = \varphi(x)$. Это преобразование может быть осуществлено различными путями, но для сходимости нужно обеспечить выполнение условия $|\varphi'(x)| \leqslant C < 1$ (C — некоторая константа). При этом задача сводится к нахождению абсциссы точки пересечения прямой $y = x$ и кривой $y = \varphi(x)$.

2. Задать начальное приближение $x^{(0)} \in [a, b]$ и малое положительное число ε. Положить $k = 0$.

3. Вычислить следующее приближение: $x^{(k+1)} = \varphi(x^{(k)})$.

4. Если $|x^{(k+1)} - x^{(k)}| \leqslant \varepsilon$, итерации завершаются и $x^* \cong x^{(k+1)}$. Если $|x^{(k+1)} - x^{(k)}| > \varepsilon$, положить $k = k + 1$ и перейти к п.3.

Теорема 1 (о сходимости метода простых итераций и единственности получаемого численного решения).

Пусть выполнены условия:

1. Нелинейное уравнение $x = \varphi(x)$ имеет решение $x^* \in G$.

2. Отображение $\varphi(x)$ является сжимающим в области G с некоторым коэффициентом C $(0 \leqslant C < 1)$.

Тогда:

а) решение x^* является единственным решением в области G;

б) последовательность $x^{(0)}, x^{(1)}, \ldots, x^{(k+1)}, \ldots$, определяемая по отображению на основе итерационного процесса, сходится к решению x^* со скоростью геометрической прогрессии, т.е. при выборе $x^{(0)}$ из условия $|x^* - x^{(0)}| < C_1$, где $C_1 > 0$ — некоторое малое число, справедливо неравенство

$$\left|x^* - x^{(k)}\right| \leqslant C^k \cdot \left|x_* - x^{(0)}\right|, \quad k = 0, 1, 2, \ldots$$

Теорема 2 (о достаточном условии сходимости метода простых итераций). Пусть выполнены условия:

1. Функция $\varphi(x)$ имеет производные для всех $x \in G$.

2. Существует число $C \left(0 \leqslant C < 1,\ C = \mathrm{const}\right)$, такое, что $|\varphi'(x)| \leqslant C$ для всех $x \in G$.

Тогда отображение $\varphi(x)$ является сжимающим в G с коэффициентом сжатия х и последовательность $x^{(0)}, x^{(1)}, \ldots, x^{(k+1)}, \ldots$, определяемая на основе итерационного процесса, сходится к решению x^*, то есть $x^{(k)} \to x^*$ при $k \to \infty$

Результаты численных экспериментов

Найти численно с точностью 10^{-6} и 10^{-12} все решения уравнения $x^2 - \sin x - 1 = 0$ на интервале $[-5, 5]$

- $x^2 - \sin x - 1 = 0 \quad \Rightarrow \quad x^2 = \sin x + 1 \quad \Rightarrow \quad 0 \le x^2 \le 2 \quad \Rightarrow \quad -\sqrt{2} \le x \le \sqrt{2}$

  все корни уравнения принадлежат отрезку $[-\sqrt{2}, \sqrt{2}]$
- $(x^2 - \sin x - 1)' = 0 \quad \Rightarrow \quad 2x = \cos x \quad \Rightarrow \quad -\frac{1}{2} \le x \le \frac{1}{2}$

  т.е. экстремум принадлежит отрезку $[-\frac{1}{2}, \frac{1}{2}]$
- Первый корень лежит в отрезке $[-\sqrt{2}, -\frac{1}{2}]$, а второй в $[\frac{1}{2}, \sqrt{2}]$

Метод дихотомии (биссекции)

Точность:	$x_0 \in [-\sqrt{2}, -\frac{1}{2}]$	$x_1 \in [\frac{1}{2}, \sqrt{2}]$	Итерации для x_0:	Итерации для x_1:
10^{-6}	-0.636732808748845	1.409623644904916	21	21
10^{-12}	-0.636732650805259	1.409624004002460	41	41
10^{-14}	-0.636732650805282	1.409624004002593	48	48

Метод простой итерации

$x^2 = \sin x + 1 \quad \Rightarrow \quad x = \pm\sqrt{\sin x + 1}$

$\Rightarrow \quad \varphi_0(x) = -\sqrt{\sin x + 1}$, $\varphi_1(x) = \sqrt{\sin x + 1}$, сжимающие отображения для отрицательного и положительного корней, удовлетворяющие 1 и 2 теореме.

Точность:	$x_0 \in [-\sqrt{2}, -\frac{1}{2}]$	$x_1 \in [\frac{1}{2}, \sqrt{2}]$	Итерации для x_0:	Итерации для x_1:
10^{-6}	-0.636732982701924	1.409623959692583	30	6
10^{-12}	-0.636732650805620	1.409624004002569	60	11
10^{-14}	-0.636732650805285	1.409624004002596	70	13

1 Метод дихотомии (биссекции)

```python
import numpy as np

def f(x):
    return (x**2)-np.sin(x)-1

interval = [[-np.sqrt(2), -1/2], [1/2, np.sqrt(2)]]

e = 1e-12

roots = []

iterations = []

for i in interval:

    iteration = 0

    a = i[0]
    b = i[1]

    while True:
        iteration+=1
        c = (a+b)/2
        if f(c) == 0:
            roots.append(c)
            break
        elif np.abs(b-a) <= e:
            roots.append(c)
            break
        if f(c)*f(a) < 0:
            b = c
        elif f(c)*f(b) < 0:
            a = c

    iterations.append(iteration)

print('Accuracy:', e)

for i in range(len(roots)):
    print('x{} = {}, {} iteration'.format(i, roots[i], iterations[i]))
```

2 Метод простой итерации

```
import numpy as np

def phi(x):
    return np.sqrt(np.sin(x)+1)

interval = [[-np.sqrt(2), -1/2], [1/2, np.sqrt(2)]]

e = 1e-12

roots = []

iterations = []

for i in interval:
    x0 = (i[0]+i[1])/2
    sign = np.sign(x0)
    xn = 0
    iteration = 0
    while True:
        iteration+=1
        xn = sign*phi(x0)
        if np.abs(x0-xn) <= e:
            roots.append(xn)
            iterations.append(iteration)
            break
        x0 = xn

print('Accuracy:', e)

for i in range(len(roots)):
    print('x{} = {}, {} iteration'.format(i, roots[i], iterations[i]))
```