Эвристика выбора интервалов функции для применения метода приближённого поиска корней

Постановка задачи

Имеется непрерывная дифференцируемая функция $f(x)$ с известной производной $f'(x)$.

Задан интервал (a,b), на котором нужно найти все корни функции $f(x)$.

Введём в качестве дополнительного предположения утверждение, что расстояние между двумя корнями не меньше, чем $d > 0$:

$$| x_{i} - x_{j}| >= d$$

для всех i,j в $\{1,...,n\}$, где n - общее число корней.

Необходимо разделить и уточнить корни функции на заданном интервале.

Первичное выделение множества интервалов

Принципы разделения корней функции

Если точка x_k является корнем функции $f(x)$, то возможны два варианта:

1. В данной точке функция меняет знак на противоположный (слева меньше нуля, справа больше нуля; либо наоборот) и имеет ненулевую производную $f'(x_k) != 0$.
2. Данная точка является экстремумом функции (минимумом или максимумом), а производная в этой точке равна нулю $f'(x_k) = 0$.

Представим график функции $f(x)$ на отрезке (a,b) и ось абсцисс Ox. Пусть на графике функции задана точка x_{t}. Будем считать, что эта точка движется вдоль графика функции $f(x)$ вдоль направления, которое способствует уменьшению абсолютного значения ординаты $y = |f(x_{t})|$. Таким образом,

1. Если $f(x_{t}) < 0$ и $f'(x_{t}) < 0$, то точка движется влево вдоль оси Ox.
2. Если $f(x_{t}) < 0$ и $f'(x_{t}) > 0$, то точка движется вправо вдоль оси Ox.
3. Если $f(x_{t}) > 0$ и $f'(x_{t}) < 0$, то точка движется вправо

вдоль оси Ox.

4. Если f(x_{t}) > 0 и f'(x_{t}) > 0, то точка движется влево вдоль оси Ox.

5. Если f(x_{t}) = 0, то точка покоится.

6. Если f(x_{t}) != 0 и f'(x_{t}) = 0, то поведение функции необходимо доопределить.

   6.1 В 2-х случаях (локальный максимум ниже оси абсцисс и локальный минимум выше оси абсцисс) оба направления (вправо и влево) ведут к увеличению расстояния до оси абсцисс, поэтому в таком случае точка должна покоиться.

   6.2 В 2-х других случаях (локальный минимум ниже оси абсцисс и локальный максимум выше оси абсцисс) оба направления ведут к уменьшению расстояния до оси абсцисс, поэтому движение влево и вправо являются одинаково допустимыми.

Точки, в которых движение должно прекращаться, назовём стационарными. Согласно приведённым выше рассуждениям, множество стационарных точек включает в себя корни функции f(x), локальные максимумы, лежащие ниже оси абсцисс, и локальные минимумы, лежащие выше оси абсцисс. Нужно также отметить, что корень функции f(x) может являться локальным максимумом или локальным минимумом этой функции, как в случае функции f(x) = x^2 и точки x = 0.

На практике для непосредственное нахождения стационарных точек не существует допустимого решения, однако можно найти достаточно узкие интервалы вокруг стационарных точек, для которых можно запустить численные методы для определения типа стационарной точки и доуточнения корня в случае его наличия на интервале(метод дихотомии, метод Ньютона и др.).

Алгоритм моделирования движения точек для нахождения интервалов вокруг стационарной точек

1. Задаём N точек на интервале (a,b).
Точки могут быть выбраны случайным образом из равномерного распределения, либо расположены в узлах равномерной сетки.

2. Для каждой такой точки x запустить циклическую процедуру, изначально присвоив x_prev = x и s_prev = 0:

   2.1 Посчитать значения f0 = f(x_prev) и f1 = f'(x_prev).

   2.2 Посчитать значения b0 = f0 >= 0 ? 1 : -1 и
   b1 = f1 >= 0 ? 1 : -1.

   2.3 Ищем x_next = x_prev + s_next * v,

```
где s_next = -b0*b1 и
v = min (max (d, f0/f1), v_max),
где v_max - некоторое число, пропорциональное длине
исходного интервала, например, v_max = 0.01 * (b-a).

2.4 Если s_next * s_prev == -1, то можно прервать процедуру,
поскольку найден интервал, содержащий стационарную точку
(min(x_prev, x_next), max(x_prev, x_next)).

2.5 s_prev = s_next, x_prev = x_next.

2.6 Если число итераций превысило заданный лимит,
то прервать вычисления, считая что заданная движущаяся
точка не находит корень.

2.7 Если заданная движущаяся точка выходит
за пределы интервала ```(a,b)```, то прервать вычисления.
```

Группировка интервалов

Описание метода

Несколько блуждающих точек могут сойтись к интервалам, соответствующим одной и той же стационарной точке. Необходимо отфильтровать интервалы, оставив только те, которые соответствуют уникальным стационарным точкам.

Будем считать, что 2 интервала `I1 = (B1, E1)` и `I2 = (B2, E2)` соответствуют одной и той же стационарной точке, если выполняются следующие условия:

1. Эти интервалы имеют ненулевое пересечение `I3 = (B3, E3)`, `E3 > B3`;
2. Выполняется `Sign (f(B3)) == Sign (f(B1))`, `Sign (f(B3)) == Sign (f(B2))`;
3. Выполняется `Sign (f(E3)) == Sign (f(E1))`, `Sign (f(E3)) == Sign (f(E2))`;
4. Выполняется `Sign (f'(B3)) == Sign (f'(B1))`, `Sign (f'(B3)) == Sign (f'(B2))`;
5. Выполняется `Sign (f'(E3)) == Sign (f'(E1))`, `Sign (f'(E3)) == Sign (f'(E2))`;

где функция `Sign(x) = x > 0 ? 1 : -1`.

Если все перечисленные условия выполняются, то интервалы `I1` и `I2` могут быть заменены единственным интервалом `I3`.

Алгоритм группировки интервалов

```
Для каждого интервала I1 в исходном списке

    Для каждого интервала I2 в выходном списке

        Построить интервал I3  как пересечение интервалов
        I1 и I2.

        Проверить условия 1-5 на соответствие стационарных
        точек друг другу

        Если точки соответствуют, то

            убрать из выходного списка интервал I2
            и добавить туда интервал I3

            прервать внутренний цикл

    Если внутренний цикл не был прерван ни разу, то

        Добавить в выходной список интервал I1
```

Нахождение корней внутри интервалов

````
Для каждого найденного интервала I = (B(I),E(I))
со стационарной точкой внутри

1. Если f(B(I)) * f(E(I)) <= 0,
то интервал однозначно содержит корень,
который нужно искать одним из известных методов
(например, метод дихотомии или метод Ньютона),
применённым к функции f(x)).

2. Если f(B(I)) * f(E(I)) <= 0,
то на интервале может быть корень,
совпадающий с экстремумом функции f(x):

    2.1 Используя один из известных методов(например,
    метод дихотомии или метод Ньютона),
    необходимо найти корень x_der_root
    функции производной f'(x) на интервале I.

    2.2 Если |f(x_der_root)| < E, где E > 0 - малое значение,
    то x_der_root также является корнем функции ```f(x)```,
    в противном случае не является.
````