Глава 2. Основы синтаксиса языка Си.

В данной главе не требуется (но допустимо) использование функций, которые будут изучаться позднее.

Методы решения уравнений

Если задано уравнение `f(x) = 0`, то с помощью компьютера его можно решить приближенно, практически с любой нужной точностью. Для этого можно применить один из описываемых ниже методов: метод дихотомии, метод хорд, метод касательных или метод итераций.

Подробно методы решения уравнений излагаются на лекциях (см. В.А. Антонюк, А.П. Иванов. «Программирование и информатика. Краткий конспект лекций.» – Москва, физический ф-т МГУ, 2015, (https://cmp.phys.msu.ru/sites/default/files/Informatics-2015.pdf, глава 2).

Методы касательных и итераций позволяют получить решение гораздо быстрее (за меньшее число итераций), чем методы дихотомии и хорд, однако, применимы они не к любым уравнениям: для того, чтобы их применить, нужно, чтобы выполнялись некоторые условия, накладываемые на производные функции, которая задает левую часть уравнения. Если такие условия не выполняются – то придется пользоваться менее эффективными, но гарантированно сходящимися методами: методом дихотомии или методом хорд.

Точностью решения будем называть найденную длину интервала по оси OX, гарантированно накрывающего истинное решение уравнения. Однако, такую величину можно явно получить только для первого метода решения – метода деления интервала пополам. Для остальных методов в качестве приближенной оценки точности будем использовать расстояние между двумя последовательными приближениями решения $|x_{i-1} - x_i|$.

Такой подход оправдан, так как описываемые методы обеспечивают монотонную сходимость приближений решения к истинному корню, а следовательно, расстояние между такими последовательными приближениями должно монотонно убывать и рано или поздно окажется меньше требуемой точности решения.

Невязкой решения называется модуль результата подстановки найденного решения в левую часть уравнения (для метода итераций это модуль разности такой подстановки и правой части уравнения). Другими словами, невязку можно назвать «точностью по оси OY». Невязка легко вычисляется для любого из приведенных методов решения уравнения.

Отметим, что в ходе численного решения уравнений в данном задании нужно добиться одновременного удовлетворения заданных условий и на точность решения и на его невязку.

Метод дихотомии

Метод дихотомии, он же метод деления отрезка пополам, он же метод вилки заключается в следующем:

1) Получаем от пользователя начальный интервал `[a, b]`, требуемую точность δ и невязку ε.
2) Проверяем, что на концах этого интервала функция левой части уравнения имеет разные знаки: `f(a) * f(b) < 0`?
3) Если нет – то решения на заданном интервале нет, либо оно не единственное, либо вообще функция имеет на нём разрыв. Печатаем диагностику и завершаем программу. Пользователь вашей программы должен попробовать подобрать начальный интервал получше.
4) Если знаки на концах интервала разные, то находим середину отрезка:
$$c = \frac{a+b}{2}$$
5) Проверяем, на каком из двух интервалов `[a, c]` или `[c, b]` функция меняет знак: `f(a) * f(c) < 0` или `f(c) * f(b) < 0` ?
6) Если функция меняет знак на первом интервале, то присваиваем переменной `b` значение `c`: `b = c`.
7) Иначе: `a = c`. Однако и для этого интервала нужно убедиться, что функция меняет на нём знак, так как функция может иметь разрывы и тогда решение построить не получится.
8) Повторяем п.п. 4-7 до того момента, когда интервал окажется меньше заданной точности δ и одновременно невязка уравнения окажется меньше заданного ε.

Точность данного метода определяется шириной интервала `|b-a|` на последней итерации, который гарантированно накрывает искомый корень.

Невязкой уравнения называется модуль подстановки текущего приближения корня `c` в уравнение: `|f(c)|`.

Таким образом, условие, при котором надо прекратить итерации, записывается так:

`|b-a| < δ && |f(c)| < ε`

Однако, обычно для построения данного решения используется цикл `while()` (продолжать, пока условие истинно), то есть, для записи условия такого цикла нужно построить логическое отрицание приведенного выше выражения – итерации нужно продолжать до тех пор, пока истинно:

`|b-a| ≥ δ || |f(c)| ≥ ε`

Такой подход следует использовать во всех приведенных методах решения уравнений, так как во всех заданиях требуется одновременное удовлетворение условий, заданных и для точности и для невязки, только точность в последующих методах определяется как расстояние между двумя последовательными приближениями корня, а не как в данном методе.

```
#include <stdio.h>
#include <math.h>
.............
    // Пусть решается уравнение: atan(x) - 1 = 0
    // (atan() – стандартная функция арктангенса)
    // fabs() – стандартная функция, вычисляющая модуль аргумента
    double c = a;
    while ( fabs(b-a) >= delta || fabs(f(c)) >= epsilon )
    {
          c = (a + b) / 2; // находим середину интервала
          if ( (atan(a)-1) * (atan(c)-1) <= 0 )
          {
                b = c; // если функция меняет знак на первом интервале
          } else if ( (atan(c)-1) * (atan(b)-1) <= 0 )
          {
                a = c; // если функция меняет знак на втором интервале
          } else
          {
                /* функция не меняет знак ни на одном интервале, она имеет разрывы
                или множественные корни на каждом из них, решение построить
                невозможно */
                printf("Error: cannot solve equation on [%lf, %lf]!", a, b);
                return 1; // завершаем функцию main() с кодом ошибки
          }
    }
.............
```

Метод хорд

Метод хорд очень похож на метод дихотомии, но часто достигает требуемого результата немного быстрее. Основное отличие заключается в способе разделения отрезка на две части:

1) Получаем от пользователя начальный интервал `[a, b]`, требуемую точность δ и невязку ε.
2) Проверяем, что на концах этого интервала функция левой части уравнения имеет разные знаки: `f(a) * f(b) < 0`?
3) Если нет – то решения на заданном интервале нет, либо оно не единственное. Печатаем диагностику и завершаем программу. Пользователь вашей программы должен попробовать подобрать начальный интервал получше.
4) Если знаки на концах интервала разные, то находим точку пересечения прямой, соединяющей точки `(a, f(a))` и `(b, f(b))`:
$$c = a + \left|\frac{f(a)}{f(b) - f(a)}\right| \cdot (b - a)$$
5) Проверяем, на каком из двух интервалов `[a, c]` или `[c, b]` функция меняет знак: `f(a) * f(c) < 0` или `f(c) * f(b) < 0 ?`
6) Если функция меняет знак на первом интервале, то присваиваем переменной `b` значение `c`: `b = c`.
7) Иначе: `a = c`. Однако и для этого интервала нужно убедиться, что функция меняет на нём знак, так как функция может иметь разрывы и тогда решение построить не получится.
8) Повторяем п.п. 4-7 до того момента, когда расстояние между двумя приближениями корня окажется меньше заданной точности δ и одновременно невязка уравнения окажется меньше заданного ε.

Для многих уравнений ширина интервала, на котором ведется поиск корня будет стремиться не к нулю, а к константе, поэтому тут точность решения будем оценивать, как расстояние между двумя последовательными приближенными решениями. Эта величина будет стремиться к нулю при увеличении количества итераций.

Метод касательных

Метод касательных называется также методом Ньютона, он заключается в следующем:

1) Получаем от пользователя начальное приближение x_0 (номер итерации `i=0`), требуемую точность

δ и невязку ε.

2) Проверяем условие сходимости $f(x_i) \cdot f''(x_i) > 0$ (то есть, проверяем, что функция постоянно выпуклая или вогнутая).
3) Если условие сходимости нарушено – печатаем диагностику и завершаем программу, метод не применим к данному уравнению на данной и последующих итерациях. Можно для диагностики выдать номер итерации и достигнутые значения корня, точности и невязки.
4) Если условие сходимости выполняется, то любое последующее уточнение значения корня получаем по итерационной формуле:

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

Эта формула получена как точка пересечения касательной в точке `xi` к графику функции `f(x)` с осью `OX`.

5) Повторяем п.п. 2-4 до того момента, когда ширина интервала `[xi+1, xi]` окажется меньше заданной точности δ и одновременно невязка уравнения окажется меньше заданного ε.

Важно: условие сходимости обязательно нужно проверять на каждой итерации цикла, а не только перед первой итерацией. Действительно, выпуклая функция может перестать быть выпуклой не сразу, а спустя сколько-то шагов движения к корню.

Метод итераций

Метод итераций применим и в тех случаях, когда условие сходимости метода касательных не выполняется, но на функцию левой части уравнения может быть наложено другое условие.

Перепишем уравнение в следующем виде:

$$F(x) = x$$

Тогда, если выполняется условие сжимающего отображения:

$$|F'(x)| < 1$$

То уравнение можно решать так:

1) Получаем от пользователя начальное приближение `x0` (номер итерации `i=0`), требуемую точность δ (невязка для данного метода не нужна, так как по построению она совпадает с оценкой точности).
2) Проверяем условие сходимости $|F'(x_i)| < 1$.
3) Если условие сходимости нарушено – печатаем диагностику и завершаем программу, метод не применим к данному уравнению на данной и последующих итерациях. Можно для диагностики выдать номер итерации и достигнутые значения корня, точности и невязки.
4) Если условие сходимости выполняется, то любое последующее уточнение значения корня получаем по итерационной формуле:

$$x_{i+1} = F(x_i)$$

5) Повторяем п.п. 2-4 до того момента, когда ширина интервала `[xi+1, xi]` окажется меньше заданной точности δ.

Важно: условие сходимости обязательно нужно проверять на каждой итерации цикла, а не только перед первой итерацией. И поясним, почему для данного метода не нужна невязка – оценка точности решения уравнения одновременно является и его невязкой:

```
|xi+1 - xi| = |F(xi) - xi|
```

Варианты задач для решения

Одним из описанных методов решить заданное уравнение.

Программа должна получить начальные значения переменной (начальный интервал для методов дихотомии и хорд, единственное начальное значение для методов касательных и итераций, требуемую точность и требуемую невязку, которые должны быть удовлетворены одновременно).

После этого программа должна произвести вычисление и напечатать:

1) *Исходное уравнение, исходный интервал и заданную точность;*
2) *Найденный корень;*
3) *Достигнутую точность: ширину последнего шага итерации (или расстояние между двумя последними приближениями корня);*
4) *Невязку: результат подстановки найденного неточного решения в левую часть уравнения;*
5) *Количество проделанных до достижения требуемой точности итераций цикла.*

Требуется, чтобы программа была устойчива к задаваемому пользователем интервалу, если корней вообще нет или корень не единственный или не выполняются условия сходимости для методов касательных и итераций нужно напечатать соответствующую диагностику. Не забывать также выдавать ошибки, если начальный интервал выходит за область допустимых значений функции уравнения.

Для методов касательных и итераций производные заданной функции найти аналитически.

Для предварительной оценки количества и расположения корней уравнения рекомендуется построить график функции уравнения: для этого можно воспользоваться любой программой электронных таблиц, для которой можно отдельно из своей программы распечатать значения нужной функции в достаточно широкой окрестности нуля, вычисленные с равномерным шагом по оси OX.

1. Вариант

Решить методом **дихотомии** уравнение:`ln(x) - 1/x = 0`

Функция натурального логарифма в Си: `log(x)`.

2. Вариант

Решить методом **хорд** уравнение: `exp(x) + x = 0`

Решить методом **касательных** уравнение:`1/x - 2*x = 0`

При анализе введенных данных учесть разрывность функции – если начальный интервал включает разрыв, то можно сразу выдавать ошибку.

Решить методом **дихотомии** уравнение: `ln(x) + 1 = 0`

Функция натурального логарифма в библиотеке языка Си: `log(x)`.

3. Вариант

Решить методом **хорд** уравнение: `arctg(x) - 1/2 = 0`

Функция арктангенса в библиотеке языка Си: `atan(x)`.

4. Вариант

Решить методом **касательных** уравнение: `exp(x) - 1/2 = 0`

5. Вариант

Решить методом **итераций** уравнение: `3 - ln(x-1) = x`

Обратить внимание, что условие сходимости выполняется не для любого интервала, входящего в область допустимых значений, необходимо это обнаруживать и выдавать соответствующее сообщение об ошибке. Корней у уравнения несколько и не любой из них можно найти предлагаемым методом. Нужно самостоятельно оценить, с какого начального приближения можно начинать искать возможные корни. Функция натурального логарифма в библиотеке языка Си: `log(x)`.

6. Вариант

Решить методом **касательных** уравнение: `exp(x) + ln(x) = 0`

Обратить внимание: условие сходимости нарушается для значений x, меньших корня. Можно уменьшить левую часть уравнения на константу 10 и оценить, как это влияет на допустимые начальные значения.

Функция натурального логарифма в библиотеке языка Си: `log(x)`.

7. Вариант

Решить методом **дихотомии** уравнение: `tg(2*x) – 1 - x = 0`

Обратить внимание на места разрывов функции, контролировать введенный начальный интервал.

Функция тангенса в библиотеке языка Си: `tan(x)`.

8. Вариант

Решить методом **хорд** уравнение: `2*exp(-x) - x = 0`

9. Вариант

Решить методом **касательных** уравнение: `exp(-3*x`^{2}`) - x - 1= 0`

Обратить внимание: у функции три близких корня и условие сходимости выполняется в довольно узкой области каждого из них. Нужно самостоятельно оценить, с какого начального приближения можно начинать искать каждый из трех корней.

10. Вариант Решить методом **итераций** уравнение: `tg(x+0.1)/3 = x` Обратить внимание: функция периодическая и разрывная и на каждом периоде есть три близко лежащих корня, при этом окрестность только одного из них удовлетворяет условию сходимости. Нужно самостоятельно оценить, с какого начального приближения можно начинать искать этот корень. Для этого рекомендуется отдельно построить графики функции и ее производной. Функция тангенса в библиотеке языка Си: `tan(x)`.
11. Вариант Решить методом **дихотомии** уравнение: `exp(-3*x) + 1 - x = 0`
12. Вариант Решить методом **хорд** уравнение: $\exp(-x^2) - x = 0$
13. Вариант Решить методом **касательных** уравнение: $\exp(-x^2) - x^2 = 0$ Обратить внимание: функция имеет два близко лежащих корня и условие сходимости выполняется в достаточно узкой области каждого из них. Нужно самостоятельно оценить, с какого начального приближения можно начинать искать каждый из двух корней.
14. Вариант Решить методом **касательных** уравнение: `2*exp(-3*x) + 1 - x = 0` Обратить внимание: условие сходимости выполняется только слева от корня, проверить это.
15. Вариант Решить методом **дихотомии** уравнение: `3*exp(-3*x) - x = 0`
16. Вариант Решить методом **хорд** уравнение: $\exp(-x^3) - 1 - x^3 = 0$
17. Вариант Решить методом **итераций** уравнение: `0.4*cos(2*x) = x`
18. Вариант Решить методом **хорд** уравнение: `tg(x/2)/10 - 1 - x = 0` Функция тангенса в библиотеке языка Си: `tan(x)`.
19. Вариант Решить методом **касательных** уравнение: `3*cos(x) - x = 0` Обратить внимание: уравнение имеет два близко лежащих корня и третий корень – чуть подальше. При этом условия сходимости для первых двух корней выполняются в довольно узкой их области, а для третьего корня условие сходимости не выполняется. Нужно самостоятельно оценить начальное приближение для первых двух корней и проверить, что третий корень найти методом касательных нельзя. .
20. Вариант Решить методом **итераций** уравнение: `exp(-x) = x` Обратить внимание: справа от корня условие сходимости выполняется всегда, а вот слева от корня – только на небольшом расстоянии от него. Самостоятельно определить, с какого минимального начального приближения (слева от корня) имеет смысл начинать его поиск?
21. Вариант Решить методом **дихотомии** уравнение: `exp(-3*x) - x = 0`

22. Вариант Решить методом **итераций** уравнение: `exp(-x²) + 2 = x` Проверить, что условие сходимости выполняется всюду, поэтому начинать можно с любого начального приближения.
23. Вариант Решить методом **хорд** уравнение: `1/x - x/2 = 0` Обратить внимание на разрывность функции, выдавать ошибку, если начальный интервал включает в себя разрыв.
24. Вариант Решить методом **дихотомии** уравнение: `(x² - 1)/(x² + 1) = 0`
25. Вариант Решить методом **хорд** уравнение: `(exp(-2x) - 2)/(exp(-x) + 1) = 0`
26. Вариант Решить методом **касательных** уравнение: `3·ln(x² + 1) - 1= 0` Обратить внимание: уравнение имеет два близко лежащих корня и условие сходимости выполняется в довольно узких интервалах вокруг каждого из них. Самостоятельно определить, с какого начального приближения нужно начинать искать каждый из корней. Функция натурального логарифма в Си: `log(x)`.
27. Вариант Решить методом **касательных** уравнение: `cos(x) + cos(x²) - 1.55 = x` Обратить внимание: уравнение имеет несколько близко лежащих корней, однако, условие сходимости метода выполняется в довольно узкой окрестности нескольких, но не всех корней. Нужно самостоятельно подобрать начальные приближения для нахождения каждого из корней, в окрестностях которых выполняются условия сходимости данного метода. Функция натурального логарифма в Си: `log(x)`.
28. Вариант Решить методом **дихотомии** уравнение: `3·ln(x² + 1) - ln(x³ + 1) - 1= 0` Уравнение имеет два корня, самостоятельно подобрать начальные интервалы для нахождения каждого из них.
29. Вариант Решить методом **дихотомии** уравнение: 3exp`(-3x)-x = 0`
30. Вариант Решить методом **хорд** уравнение: `exp(-2x) - x = 0`