Задача 5. РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ.

Решение нелинейных (трансцендентных или алгебраических) уравнений вида f(x)=0 заключается в отыскании одного или нескольких корней, т.е. таких значений аргумента x, для которых функция f(x) обращается в нуль. В общем случае функции f(x) не имеют аналитических формул для своих корней, поэтому приходится использовать приближенные методы.

Решение нелинейных уравнений обычно состоит из двух этапов:

1) Отделение, или локализация корней, т.е. отыскание таких отрезков [a,b] (одного или нескольких), внутри которых имеется только один корень нелинейного уравнения.

2) Уточнение приближенного значения корня до некоторой заданной степени точности.

Отделить корни в некоторых случаях можно графически. Так, если уравнение имеет вид cos(x)-x=0, то переписав его в виде cos(x)=x и построив графики функций y_1=cos(x) и y_2=x, найдем приближенное значение корня как точку пересечения функций y_1 и y_2.

Локализовать корень уравнения можно также программно, вычисляя функцию f(x) для значнний x, изменяющихся с некоторым заданным шагом h. При этом отыскиваются два таких соседних значения x, для которых f(x) имеет противоположные знаки. Алгоритм отделения корня удобно оформить в виде подпрограммы. В качестве параметров подпрограммы можно выбрать следующие: F,XL,XR,H,A,B,IER.

Входные параметры:

F - имя внешней функции f(x);

XL,XR - соответственно левая и правая границы отрезка оси x, на котором отделяется корень уравнения;

H - шаг перебора аргумента функции.

Выходные параметры:

A,B - соответственно левая и правая границы отрезка [a,b], содержащего первый, считая от точки XL вправо, корень нелинейного уравнения;

IER - код ошибки. IER=0, если на участке от XL до XR с шагом H найден отрезок [A,B], содержащий корень уравнения; IER=1, если корень не локализован.

Алгоритм отделения корня нелинейного уравнения может быть следующим:

1) Зададим IER=1, т.е. предполагаем, что на отрезке от XL до XR корней нет. В дальнейшем проверим, так ли это.

2) Зададим начальное значение аргумента функции X=XL.

3) Вычислим значение функции в точке X: Y=F(X).

4) Изменим значение аргумента функции на величину шага : X=X+H .

5) Проверим, лежит ли X внутри [XL,XR]: если X>XR , то возвращаемся в вызывающую программу. В этом случае отрезок [A,B] не найден,IER=1.

6) Вычислим значение функции в новой точке X: Z=F(X).

7) Проверим, изменила ли функция знак при последнем изменении аргумента. Если не изменила, то опять меняем X : если Y*Z>0 идти на 4.

8) В противном случае определим границы отрезка [A,B] по формулам A=X-H, B=X, изменим код ошибки IER=0 и вернемся в вызывающую программу.

Для уточнения корня нелинейного уравнения до заданной точности можно воспользоваться методами половинного деления, Ньютона, простых итераций и другими.

МЕТОД ПОЛОВИННОГО ДЕЛЕНИЯ состоит в построении последовательности вложенных отрезков, на концах которых функция принимает значения разных знаков. Каждый последующий отрезок получают делением предыдущего пополам. Процесс построения последовательности отрезков позволяет найти корень уравнения f(x)=0 с любой заданной точностью.

В качестве параметров подпрограммы метода половинного деления можно выбрать следующие: F,A,B,EPS,C,N,IER.

Входные параметры:

F - имя внешней функции f(x);

A,B - соответственно левая и правая граница отрезка,содержащего один корень уравнения f(x)=0;

EPS - точность вычисления корня.

Выходные параметры:

C - корень уравнения (если он найден);

N - количество итераций, которое потребовалось выполнить для вычисления корня с заданной точностью;

IER - код ошибки. IER=0, если корень найден; IER=1, если A>B или на отрезке [A,B] нет корня, т.е. если F(A)*F(B)>0.

Алгоритм метода половинного деления:

1) Зададим код ошибки IER=1.

2) Проверим, правильно ли заданы исходные данные: если F(A)*F(B)>0 или A>B, то возвращаемся в вызывающую программу.

3) Зададим текущие границы отрезка, содержащего корень A1=A, B1=B; зададим начальное число итераций N=0 и IER=0; вычислим значение функции в точке A1: Y=F(A1).
4) Вычислим середину отрезка [A1,B1]: C=.5*(A1+B1); изменим счетчик итераций на единицу : N=N+1.
5) Если длина текущего отрезка [A1,B1] меньше заданной точности, то возвращаемся в вызывающую программу.
6) Проверим, внутри какого из отрезков [A1,C] или [C,B1] лежит корень уравнения. Если корень лежит внутри [A1,C], то изменяем правую текущую границу отрезка, содержащего корень, в противном случае - левую: если F(A1)* F(C)<0, то B1=C, иначе A1=C; итерационный процесс повторяется с пункта 4.

Метод половинного деления - наиболее универсальный метод отыскания корней нелиней ных уравнений. К его недостаткам относится невысокая скорость сходимости.

МЕТОД НЬЮТОНА (метод касательных) является одним из наиболее эффективных методов нахождения корней нелинейных уравнений. Он состоит в построении итерационной последовательности $x_{i+1}=x_i-f(x_i)/f'(x_i)$, сходящейся к корню уравнения f(x)=0. Геометрическая интерпретация метода следующая: если через точку с координатами $(x_i ,f(x_i))$ провести касательную к функции y=f(x), то точка пересечения этой касательной с осью абсцисс принимается за уточненное значение корня x_{i+1}. Метод Ньютона особенно эффективен, когда известно хорошее приближение корня и в окрестности корня функция имеет большую крутизну.

В качестве параметров подпрограммы метода Ньютона можно рекомендовать следующие: F,PF,X0,EPS,NMAX,X,N,IER.

Входные параметры:

F - имя внешней функции f(x);

PF - имя внешней функции f '(x);

X0 - начальное приближение для корня уравнения;

EPS - точность вычисления корня;

NMAX - максимальное количество итерационных циклов.

Выходные параметры:

X - корень уравнения;

N - количество итераций, выполненных для вычисления корня с заданной точностью;

IER - код ошибки. Если корень найден, то IER=0; если корень не найден за NMAX итерационных циклов, то IER=1.

Используемый в подпрограмме метода Ньютона алгоритм:

1) Зададим: IER=1; начальное значение количества итераций N=0; текущее значение корня X=X0.

2) Выполним итерационный цикл: изменим содержимое счетчика итераций на единицу N=N+1; если количество итераций превысило NMAX, осуществим возврат в вызывающую программу; вычислим уточненное значение корня по формуле

Y=X-F(X)/PF(X); вычислим модуль разности между уточненным и текущим значениями корня E=ABS(Y-X); в качестве нового текущего зададим уточненное значение корня X=Y; проверим, следует ли продолжать уточнение корня: если E>EPS, то итерационный цикл следует повторить с начала пункта 2.

3) Если точность вычисления корня достигнута, то следует изменить код ошибки IER=0 и осуществить возврат в вызывающую программу.

Один из недостатков метода Ньютона состоит в том, что пользуясь им, приходится дифференцировать функцию f(x). Если сделать это затруднительно, то можно производную заменить конечной разностью.

МЕТОД ПРОСТЫХ ИТЕРАЦИЙ состоит в замене исходного уравнения f(x)=0 эквивалентным ему уравнением x=p(x) и в построении последовательности $x_{i+1}=p(x_i)$, сходящейся к точному решению. Достаточным условием сходимости метода является |p'(x)|<1, которое должно выполняться для всех x . При использовании метода простых итераций основную трудность представляет выбор функции p(x), удовлетворяющей условию сходимости. При этом, если уравнение имеет несколько корней, то для вычисления каждого из них необходимо выбирать свой вид функции p(x). Например, уравнение ln(1.5*x)-2*x+4=0 имеет два корня. Переписав его как ln(1.5*x)=2*x-4, можно получить 1.5*x=exp(2*x-4), тогда x=exp(2*x-4)/1.5 и p_1(x)=exp(2*x-4)/1.5. Используя функцию p_1(x) вычислим первый, близкий к нулю, корень. Если исходное уравнение записать как 2*x=ln(1.5*x)+4 , то получим функцию p_2(x)=(ln(1.5*x)+4)/2, применяя которую вычислим второй корень. При составлении подпрограммы метода простых итераций можно использовать алгоритм метода Ньютона, определяя уточненное значение корня по формуле Y=P(X), где P - имя внешней функции p(x).

Перед составлением головной программы необходимо провести графический анализ поведения функции y=f(x). При этом определяется количество нулей функции. Функция может иметь один, несколько или бесконечное множество нулей. В каждом конкретном случае при решении нелинейного уравнения f(x)=0 ставится своя задача, которая может быть сфоррмулирована как: а) найти единственный корень; б) найти первый положительный корень; в) найти первые М корней, начиная с точки XL; г) найти все корни на отрезке [XL,XR] и т.д.. В зависимости от поставленной задачи готовятся исходные данные и составляется алгоритм решения.

Рассмотрим пример. Пусть решается уравнение tg(x)-ctg(x)+x=0. Это уравнение имеет бесчисленное множество корней. Сформулируем задачу так: найти все корни уравнения на отрезке [-1.5, 4]. Алгоритм программы может быть следующим:

1) Описываются внешние функции, являющиеся фактическими параметрами при обращении к процедурам.

2) Задаются исходные данные (отрезок [XL,XR], на котором ищутся корни, точность вычисления корней EPS). С помощью подпрограммы локализации корней на промежутке от XL до XR определяется первый от точки XL отрезок [A, B], на котором функция меняет знак. Подпрограмма локализации корня может и не найти такого отрезка. Поэтому после работы подпрограммы проверяется ее код ошибки: если IER=1, то необходимо выдать на печать сообщение о том, что отрезок [A,B] не найден и остановить выполнение программы.

4) В дальнейшем, при поиске второго, третьего и т.д. корней отделение корней необходимо проводить не с исходной точки XL, а начиная с правой границы предыдущего отрезка, на котором функция сменила знак. Поэтому необходимо выполнить операцию XL=B.

5) Обращаясь к подпрограмме метода половинного деления, можно уточнить до заданной точности точку смены знака функцией f(x). Рассматриваемая функция имеет не только нули, но и точки разрыва, которые отделяются аналогично нулям. После обращения к подпрограмме метода половинного деления следует проверить, является ли найденная точка C корнем уравнения или нет. Это можно сделать, вычислив значение модуля функции в точке C или анализируя поведение производной на концах отрезка [A,B]. Если C - точка разрыва, то необходимо передать управление на пункт 3 и определить следующий отрезок [A, B], на котором функция меняет знак.

6) В соответствии с заданием в программе должно быть предусмотрено уточнение корней методами Ньютона и простых итераций. В качестве начального приближения для этих методов можно взять середину отрезка [A, B].

7) После вычисления и печати значения очередного корня управление передается на п. 3.

ЗАДАНИЕ. Используя методы половинного деления и Ньютона, найти корни нелинейных уравнений $f_1(x)=0$ и $f_2(x)=0$. Методом простых итераций найти корни уравнения $f_2(x)$. Функции $f_1(x)$ и $f_2(x)$ выбрать из таблицы по номеру варианта.

При выполнении задания требуется:

1) Построить графики функций $y_1=f_1(x)$ и $y_2=f_2(x)$ для того, чтобы определить количество корней и отрезки, где они расположены. Функции могут иметь один, несколько или бесчисленное множество корней. Необходимо решить вопрос с преподавателем, какие корни подлежат определению.

2) Составить все необходимое для расчетов программное обеспечение: подпрограммы-функции для вычисления $f_1(x)$,$f_1'(x)$,$f_2(x)$,$f_2'(x)$,$p_1(x)$,$p_2(x)$; подпрограммы для локализации корней, методов половинного деления, Ньютона, простых итераций; головную программу. 3) Провести счет. Результаты оформить в виде таблицы. Предусмотреть печать не только окончательных, но и промежуточных результатов (отрезков [A, B], числа итераций, кодов ошибок и т.д.).

ВАРИАНТЫ ЗАДАНИЙ

N	f1(x)	f2(x)
1	x+sqrt(x)+x**(1/3)-2.5	tg(1.5773*x)-2.3041*x
2	cos(2/x)-2*sin(1/x)+1/x	ln(7.6221*x)-8.591*x+1.5
3	cos(x)-exp(-x**2/2)+x-1	9.33*sin(6.977*x)-7.25*x
4	sin(x**2)+cos(x**2)-10*x	0.9737*exp(-.5067*x)-x
5	tg(x/2)-ctg(x/2)+x	tg(2.2083*x)-3.2258*x
6	x-sin(x)-.25	ln(6.0976*x)-6.872*x+2.01
7	sqrt(x)-cos(.387*x)	7.67*sin(5.983*x)-6.01*x
8	x**2+4*sin(x)	0.9286*exp(-.5185*x)-x
9	1.8*x**2-sin(10*x)	tg(3.7855*x)-5.5301*x
10	3*x-cos(x)-1	ln(4.5732*x)-3.963*x+5.154
11	2*x-ln(x)-7	6.67*sin(5.387*x)-5.25*x
12	x**2*cos(2*x)-1	0.5458*exp(-.5391*x)-x
13	sin(x+pi/3)-.5*x	tg(9.1483*x)-13.364*x
14	cos(x+pi/8)-x**3	ln(3.9634*x)-4.4868x+2.01
15	2*arctg(x)-3*x+2	5.67*sin(4.794*x)-4.55*x
16	3*x-cos(x)-1	0.7593*exp(-.5683*x)-x
17	(x-3)*cos(x)-1	tg(5.9937*x)-8.7558*x
18	sqrt(1-x)-tg(x)	ln(3.0488*x)-3.436*x+2.5
19	ctg(1.05*x)-x**2	4.33*sin(4.008*x)-3.55*x

20	.6*3**x-2.3*x-3	0.5909*exp(-.6286*x)-x