6 Нелинейные уравнения и системы в SCILAB

Если нелинейное уравнение достаточно сложное, то отыскание его корней процесс нетривиальный. Рассмотрим, какими средствами обладает Scilab для решения этой задачи.

6.1 Алгебраические уравнения

Любое уравнение $P(x)=0$, где $P(x)$ это многочлен, отличный от нулевого, называется *алгебраическим уравнением* или *полиномом.* Всякое алгебраическое уравнение относительно x можно записать в виде $a_0x^n+a_1x^{n-1}+\dots+a_{n-1}x+a_n=0$, где $a_0\neq0$, $n\geq1$ и a_i коэффициенты алгебраического уравнения n й степени .Например ,линейное уравнение это алгебраическое уравнение первой степени, квадратное второй ,кубическое третьей и так далее .

Решение алгебраического уравнения в Scilab состоит из двух этапов. Необходимо задать полином $P(x)$ с помощью функции `poly`, а затем найти его корни применив функцию `roots`.

Итак, *определение полиномов* в Scilab осуществляет функция

`poly(a, "x", ["fl"]),`

где `a` это число или матрица чисел, `x` символьная переменная , `fl` необязательная символьная переменная, определяющая способ задания полинома. Символьная переменная `fl` может принимать только два значения `"roots"` или `"coeff"` (соответственно `"r"` или `"c"`). Если `fl=c`, то будет сформирован полином с коэффициентами, хранящимися в параметре `a`. Если же `fl=r`, то значения параметра `a` воспринимаются функцией как корни, для которых необходимо рассчитать коэффициенты соответствующего полинома. По умолчанию `fl=r`.

Следующий пример отражает создание полинома `p`, имеющего в качестве корня тройку, и полинома `f` с коэффициентом три.

```
-->p=poly(3,'x','r');
-->f=poly(3,'x','c');
-->p
 p  =
     - 3 + x
-->f
 f  =
      3
```

Листинг 6.1

Далее приведены примеры создания более сложных полиномов.

```
-->//Полином с корнями 1, 0 и 2
-->poly([1 0 2],'x')
 ans  =
              2   3
    2x - 3x + x
-->//Полином с коэффициентами 1, 0 и 2
```

```
-->poly([1 0 2],'x','c')
 ans  =
        2
    1 + 2x
```

Листинг 6.2

Рассмотрим примеры *символьных операций* с полиномами:

```
-->p1=poly([-1 2],'x','c')
 p1  =
  - 1 + 2x
-->p2=poly([3 -7 2],'x','c')
 p2  =
             2
    3 - 7x + 2x
-->p1+p2 //Сложение
 ans  =
             2
    2 - 5x + 2x
-->p1-p2 //Вычитание
 ans  =
             2
-  4 + 9x - 2x
-->p1*p2 //Умножение
 ans  =
                 2    3
-    - 3 + 13x - 16x + 4x
-->p1/p2 //Деление
 ans  =
      1
    -----
  - 3 + x
-->p1^2 //Возведение в степень
 ans  =
             2
    1 - 4x + 4x
-->p2^(-1) //Возведение в отрицательную степень
 ans  =
         1
    -----------
              2
    3 - 7x + 2x
```

Листинг 6.3

Функция

`roots(p)`

предназначена для *решения алгебраического уравнения*. Здесь `p` это полином созданный функцией `poly` и представляющий собой левую часть уравнения $P(x)=0$.

Решим несколько алгебраических уравнений.

ЗАДАЧА 6.1.

Найти корни полинома $2x^4-8x^3+8x^2-1=0$.

Для решения этой задачи необходимо задать полином `p`. Сделаем это при помощи функции `poly`, предварительно определив вектор коэффициентов `V`. Обратите внимание, что в уравнении отсутствует переменная `x` в первой степени, это означает, что соответствующий коэффициент равен нулю:

```
-->V=[-1 0 8 -8 2];
-->p=poly(V,'x','c')
 p  =
            2    3    4
   1 + 8x - 8x + 2x
```

Листинг 6.4

Теперь найдем корни полинома:

```
-->X=roots(p)
 X  =
!   0.4588039 !
! - 0.3065630 !
!   1.5411961 !
!   2.306563  !
```

Листинг 6.5

Графическое решение задачи[1], показанное на рис. 6.1 позволяет убедится, что корни найдены верно.

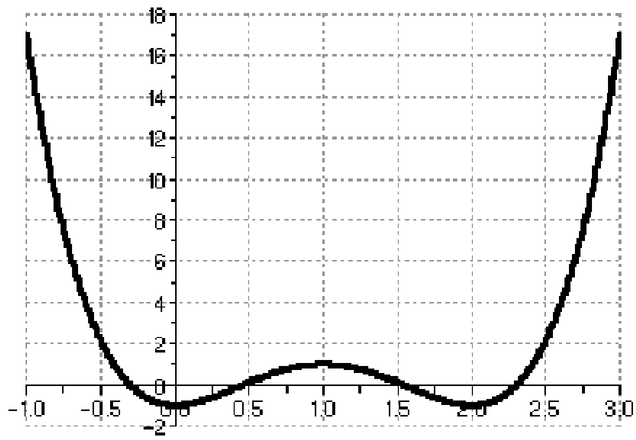

Рисунок 6.1. Графическое решение задачи 6.1

ЗАДАЧА 6.2

Найти корни полинома $x^3+0.4x^2+0.6x-1=0$.

Решение этой задачи аналогично решению предыдущей, разница заключается в способе вызова необходимых для этого функций:

```
-->roots(poly([-1 0.6 0.4 1],'x','c'))
```

1 Графическим решением уравнения $f(x)=0$ является точка пересечения линии $f(x)$ с осью абсцисс.

```
 ans  =
!   0.7153636              !
! - 0.5576818 + 1.0425361i !
! - 0.5576818 - 1.0425361i !
```

Листинг 6.6

Не трудно заметить, что полином имеет один действительный (рис.6.2) и два комплексных корня.

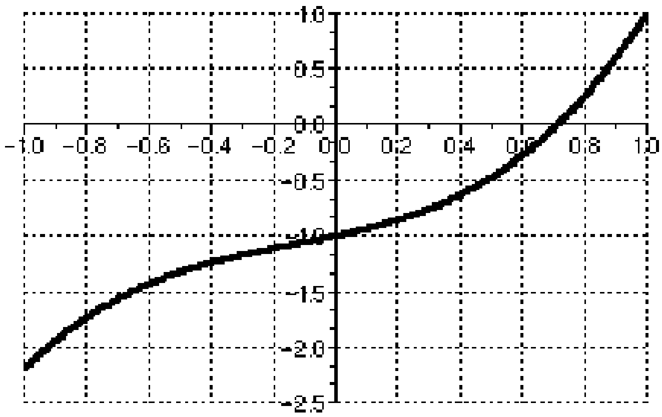

Рисунок 6.2. Графическое решение задачи 6.2.

ЗАДАЧА 6.3

Найти решение уравнения $y(x)=0$, если $y(x)=x^4-18x^2+6$.

Решение этой задачи представлено в листинге 6.7 и отличается от предыдущих лишь способом определения полинома. Графическое решение представлено на рис. 6.3.

```
-->x=poly(0,'x');
-->y=x^4-18*x^2+.6;
-->roots(y)
 ans  =
!   0.1827438 !
! - 0.1827438 !
! - 4.2387032 !
!   4.2387032 !
```

Листинг 6.7

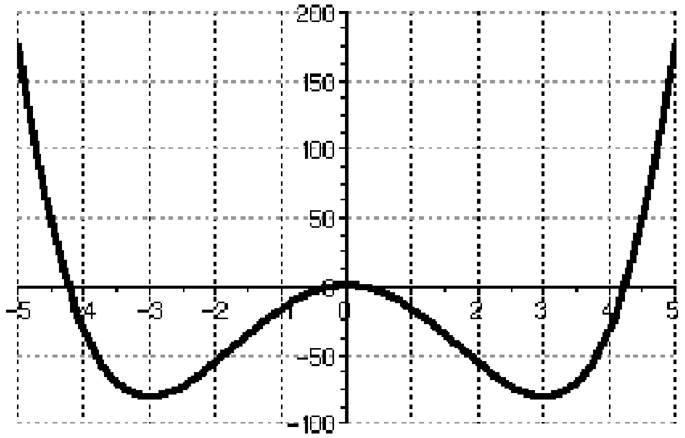

Рисунок 6.3. Графическое решение задачи 6.3

6.2 Трансцендентные уравнения

Уравнение $f(x)=0$, в котором неизвестное входит в аргумент трансцендентных функций, называется *трансцендентным уравнением.* К трансцендентным уравнениям принадлежат показательные, логарифмические, тригонометрические. В общем случае аналитическое решение уравнения $f(x)=0$ можно найти только для узкого класса функций. Чаще всего приходится решать это уравнение *численными методами.*

Численное решение нелинейного уравнения проводят в два этапа. В начале отделяют *корни уравнения*, то есть находят достаточно тесные промежутки, в которых содержится только один корень. Эти промежутки называют *интервалами изоляции корня*, определить их можно, изобразив график функции $f(x)$ или любым другим методом[2]. На втором этапе проводят уточнение отделенных корней, или, иначе говоря, находят корни с заданной точностью.

Для решения трансцендентных уравнений в Scilab применяют функцию

`fsolve(x0,f)`

где `x0` начальное приближение ,`f` функция ,описывающая левую часть уравнения y(x)=0.

Рассмотрим применение этой функции на примерах.

ЗАДАЧА 6.4

Найти решение уравнения $\sqrt[3]{(x-1)^2}-\sqrt[3]{x^2}=0$.

Определим интервал изоляции корня заданного уравнения. Воспользуемся графическим методом отделения корней. Если выражение, стоящее в правой части уравнения представить в виде разности двух функций $f(x)$ $g(x)=0$, то абсцисса точки пересечения линий $f(x)$ и $g(x)$ корень данного уравнения .В нашем случае $f(x)=\sqrt[3]{(x-1)^2}$, $g(x)=\sqrt[3]{x^2}$. На рис. 6.4 видно, что корень данного уравнения лежит в интервале [0;1].

Выберем ноль в качестве начального приближения, зададим функцию, описывающую уравнение и решим его:

```
-->deff('[y]=f1(x)','y1=((x-1)^2)^(1/3),y2=(x^2)^(1/3),y=y1-y2')
-->fsolve(0,f1)
 ans  =     0.5
```

Листинг 6.8

2 Методы определения интервала изоляции корня основаны на следующем свойстве: если непрерывная функции $f(x)$ на интервале $[a, b]$ поменяла знак, то есть $f(a)\cdot f(b)<0$, то она имеет на этом интервале хотя бы один корень.

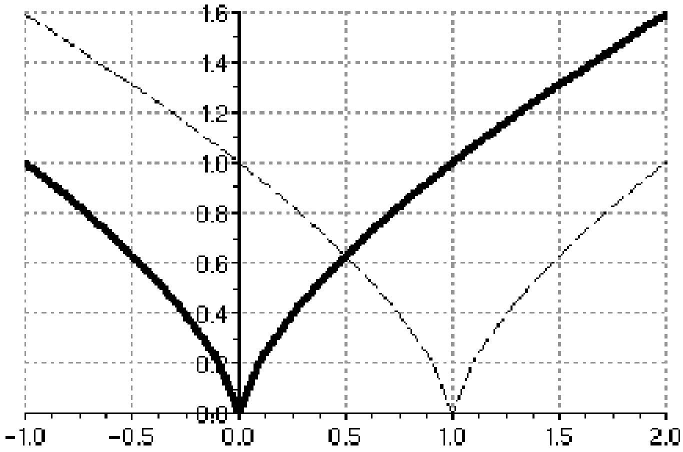

Рисунок 6.4. Графическое решение задачи 6.4

ЗАДАЧА 6.5

Найти корни уравнения $f(x)=e^x/5-2(x-1)^2$.

На рис. 6.5 видно, что график функции $f(x)$ трижды пересекает ось абсцисс, то есть уравнение имеет три корня.

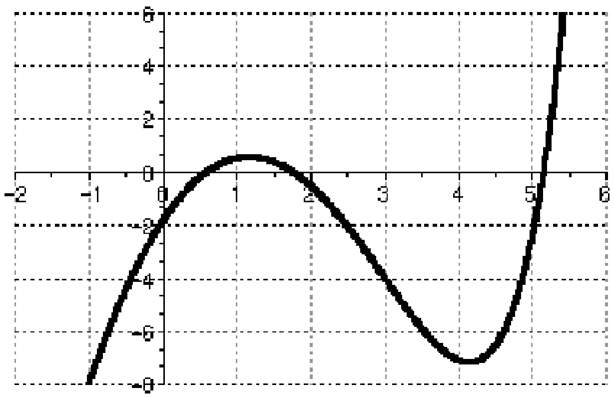

Рисунок 6.5. Графическое решение задачи 6.5

Последовательно вызывая функцию `fsolve` с различными начальными приближениями, получим все решения заданного уравнения:

```
-->deff('[y]=f(x)','y=exp(x)/5-2*(x-1)^2')
-->x(1)=fsolve(0,f);x(2)=fsolve(2,f);x(3)=fsolve(5,f);
-->x
 x  = !   0.5778406 !
      !   1.7638701 !
      !   5.1476865 !
```

Листинг 6.9

Кроме того, начальные приближения можно задать в виде вектора и тогда функцию можно вызвать один раз:

```
-->fsolve([0;2;5],f)
 ans  = !   0.5778406 !
        !   1.7638701 !
```

```
        !   5.1476865 !
```

Листинг 6.10

ЗАДАЧА 6.6

Вычислить корни уравнения $\sin(x)$-0.4x=0 в диапазоне [5π;5π].

Решение задачи представлено в листинге 6.11.

```
-->deff('[y]=fff(x)','y=-0.4+sin(x)')
-->V=[-5*%pi:%pi:5*%pi]; X=fsolve(V,fff);
-->X //Множество решений
 X  = !-16.11948 -12.154854 -9.8362948 -5.8716685 -3.5531095
  0.4115168 2.7300758 6.6947022 9.0132611 12.977887 15.296446!
```

Листинг 6.11

ЗАДАЧА 6.7

Найти решение уравнения $y(x)$=0, если $y(x)=x^5-x^3+1$.

Не трудно заметить, что заданное уравнение   полином пятой степени ,который имеет один действительный корень (рис. 6.6) в интервале от 2до 1.

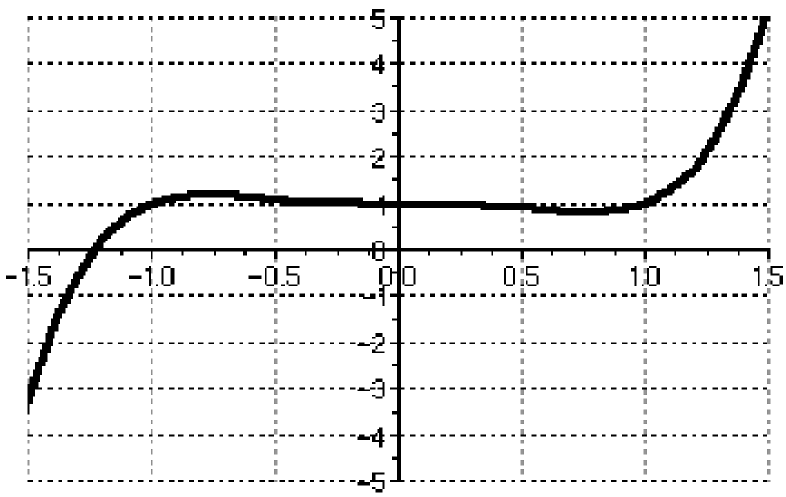

Рисунок 6.6. Графическое решение задачи 6.7

Решим эту задачу при помощи функции `fsolve`:

```
-->deff('[f]=y(x)','f=x^5-x^3+1')
-->X=fsolve(-2,y)
 X  = 1.2365057
```

Листинг 6.12

Теперь применим функцию `roots`:

```
-->roots(poly([1 0 0 -1 0 1],'x','c'))
 ans  =
!   0.9590477 + 0.4283660i !
!   0.9590477 - 0.4283660i !
! - 0.3407949 + 0.7854231i !
! - 0.3407949 - 0.7854231i !
! - 1.2365057              !
```

Листинг 6.13

Как видим, заданное уравнение, кроме действительного корня (листинг 6.12), имеет и мнимые (листинг 6.13). Поэтому для отыскания всех корней полинома лучше использовать функцию `roots`.

6.3 Системы уравнений

Если заданы m уравнений с n неизвестными и требуется найти последовательность из n чисел, которые одновременно удовлетворяют каждому из m уравнений, то говорят о *системе уравнений*. Для решения систем уравнений в Scilab так же применяют функцию `fsolve(x0,f)`.

ЗАДАЧА 6.8

Решить систему уравнений: $\{x^2+y^2=1;\ x^3-y=0\}$.

Графическое решение системы (рис. 6.7) показывает, что она имеет две пары корней.

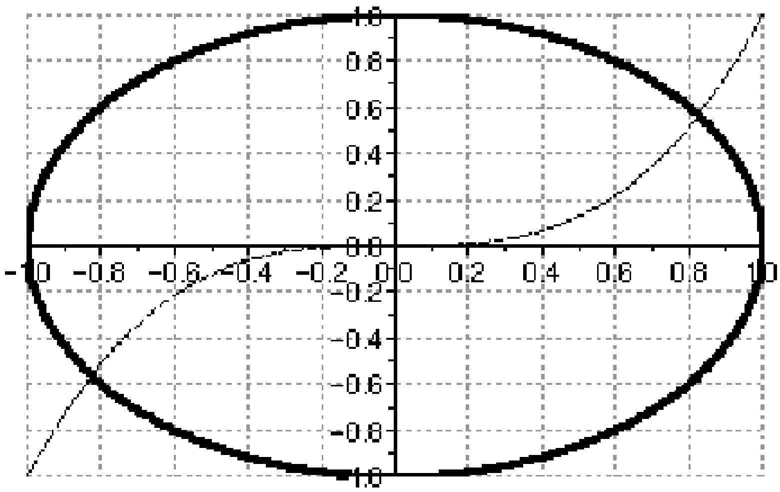

Рисунок 6.7. Графическое решение системы уравнений

Окружность и гипербола пересекаются в точках [0.8;0.6] и [-0.8;-0.6]. Эти значения приблизительны. Для того чтобы уточнить их, применим функцию `fsolve`, предварительно определив систему с помощью файл функции :

```
function [y]=fun(x)
y(1)=x(1)^2+x(2)^2-1;
y(2)=x(1)^3-x(2);
endfunction
-->exec('C:\fun.sce'); disp('exec done'); exec done
-->fsolve([0.5 0.5],fun)
 ans  = 0.8260314    0.5636242
-->fsolve([-0.5 -0.5],fun)
 ans  =  - 0.8260314  - 0.5636242
```

Листинг 6.14

ЗАДАЧА 6.9

В данной задаче исследуется система из трех нелинейных уравнений с тремя неизвестными:

```
function [y]=fun(x)
y(1)=x(1)^2+x(2)^2+x(3)^2-1
y(2)=2*x(1)^2+x(2)^2-4*x(3)
```

```
y(3)=3*x(1)^2-4*x(2)+x(3)^2
endfunction
-->exec('D:\scilab 3\fun');disp('exec done'); exec done
-->fsolve([0.5 0.5 0.5],fun)//решение системы
ans  = !   0.7851969    0.4966114    0.3699228 !
```

Листинг 6.15