7. Нелинейные уравнения и системы

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 уравнения* проводят в два этапа. На *первом этапе* отделяют корни уравнения, т.е. находят достаточно тесные промежутки, в которых содержится только один корень. Эти промежутки называют *интервалами изоляции корня*. Определить интервалы изоляции корня можно, например, изобразив график функции. Идея *графического метода* основана на том, что непрерывная функции $f(x)$ имеет на интервале $[a, b]$ хотя бы один корень, если она поменяла знак $f(a) \cdot f(b)<0$. Границы интервала a и b называют *пределами интервала изоляции*. На *втором этапе* проводят *уточнение* отделенных корней, т.е. находят корни с заданной точностью.

7.1 Решение алгебраических уравнений

Любое уравнение $P(x)=0$, где $P(x)$ это многочлен (*полиномом*), отличный от нулевого, называется *алгебраическим уравнением* относительно переменных x. Всякое алгебраическое уравнение относительно x можно записать в виде

$$a_0+a_1 x+a_2 x^2+...+a_n x^n=0$$

где $a_0 \neq 0, n \geqslant 1$, a_i – коэффициенты алгебраического уравнения n–й степени. Например, линейное уравнение это алгебраическое уравнение первой степени, квадратное – второй, кубическое – третьей и так далее.

В Octave *определить алгебраическое уравнение* можно в виде вектора его коэффициентов $p=\{a_n, a_{n-1}, ..., a_{1,} a_0\}$. Например, полином $2x^5+3x^3-1=0$ задается вектором

```
>>> p=[2,0,3,0,-1]
p =
2 0 3 0 -1
```

Листинг 7.1

Рассмотрим функции, предназначенные для *действий над многочленами*.

Произведение двух многочленов вычисляет функция

`q=conv(p1, p2).`

где `p1` — многочлен степени n, `p2` — многочлен степени m. Функция формирует вектор `q`, соответствующий коэффициентам многочлена степени $n+m$, полученного в результате умножения `p1` на `p2`;

ЗАДАЧА 7.1. Определить многочлен, который получиться в результате умножения выражений $3x^4-7x^2+5$ и $x^3+2x-1=0$. Как видно из листинга 7.2 в результате имеем: $(3x^4-7x^2+5)(x^3+2x-1=0)=3x^7-x^5-3x^4-9x^3+7x^2+10x-5$.

```
>>> p1=[3 0 -7 0 5];
p2=[1 0 2 -1];
p=conv(p2,p1)
>>>p =
    3    0   -1   -3   -9    7   10   -5
```

Листинг 7.2

Частное и остаток от деления двух многочленов находит функция

`[q, r] =deconv((p1, p2).`

здесь, `p1` — многочлен степени n, `p2` — многочлен степени m. Функция формирует вектор `q` – коэффициенты многочлена, который получается в результате деления `p1` на `p2` и вектор `r` – коэффициенты многочлена, который является остатком от деления p1 на p2;

ЗАДАЧА 7.2. Найти частное и остаток от деления многочлена $x^6-x^5+3x^4-8x^2+x-10$ на многочлен $x^3+x-1=0$.

В результате имеем (листинг 7.3)

$$\frac{x^6-x^5+3x^4-8x^2+x-10}{x^3+x-1}=x^3-x^2+2x+2+\frac{1}{-11x^2+x-8}.$$

```
>>> p1=[1 -1 3 0 -8 1 -10];
p2=[1 0 1 -1];
[q,r]=deconv(p1,p2)
>>>q =
   1  -1   2   2
r =
    0    0    0    0  -11    1   -8
```

Листинг 7.3

Выполнить *разложение частного двух многочленов*, представляющих собой правильную дробь на простейшие рациональные дроби вида

$$\frac{P_1(x)}{P_2(x)}=\sum_{j=1}^{M}\frac{a_j}{(x-b_j)^{k_j}}+\sum_{i=1}^{N}c_i x^{N-i}$$

можно с помощью функции

```
[a, b, c, k] = residue(p1, p2)
```

где p1 — многочлен степени n (числитель), p2 — многочлен степени m (знаменатель), причем $n<m$. В результате работы функция формирует четыре вектора: a — вектор коэффициентов, расположенных в числителях простейших дробей, b — вектор коэффициентов, расположенных в знаменателях простейших дробей, k — вектор степеней знаменателей простейших дробей (кратность), c — вектор коэффициентов остаточного члена.

ЗАДАЧА 7.3. Разложить выражение $\frac{x^3+1}{x^4-3x^3+3x^2-x}$ на простейшие дроби. Проанализировав листинг 7.4 запишем решение:

$$\frac{x^3+1}{x^4-3x^3+3x^2-x}=\frac{2}{x-1}+\frac{1}{(x-1)^2}+\frac{2}{(x-1)^3}+\frac{-1}{x}.$$

Значение вектора c = [](0x0) говорит об отсутствии остаточного члена.

```
>>> p1=[1 0 0 1];
p2=[1 -3 3 -1 0];
[a,b,c,k]=residue(p1,p2)
>>>a =
   2.00000
   1.00000
   2.00000
  -1.00000
b =
   1.00000
   1.00000
   1.00000
   0.00000
c = [](0x0)
k =
   1
```

```
2
3
1
```

Листинг 7.4

ЗАДАЧА 7.4. Разложить выражение $\frac{7x^2+26x-9}{x^4+4x^3+4x^2-9}$ на простейшие дроби. Из листинга 7.5 видно, что значения векторов `a` и `b` — комплексные числа, то есть, на первый взгляд кажется, что выражение невозможно разложить на простейшие рациональные дроби. Однако задача имеет решение:

$$\frac{7x^2+26x-9}{x^4+4x^3+4x^2-9}=\frac{1}{x-1}+\frac{1}{x+3}+\frac{-2x+5}{x^2+2x+3}.$$

Выражение $\frac{-2x+5}{x^2+2x+3}$ — простейшая дробь вида $\frac{Mx+N}{x^2+px+q}$. Здесь знаменатель невозможно разложить на простые рациональные множители первой степени. Таким образом, функция `residue(p1, p2)` выполняет разложение только на простейшие дроби вида $\frac{a}{(x-b)^k}$.

```
>>>a =
  -1.3738 + 1.2229i
  -1.3738 - 1.2229i
   1.3738 + 0.7771i
   1.3738 - 0.7771i
b =
  -2.4426 + 1.0398i
  -2.4426 - 1.0398i
   0.4426 - 1.0398i
   0.4426 + 1.0398i
c = [](0x0)
k =
   1
   1
   1
   1
```

Листинг 7.5

Вычислить *значение многочлена в заданной точке* можно с помощью функции

`polyval(p, x)`

где `p` - многочлен степени *n*, `x` — значение, которое нужно подставить в многочлен.

ЗАДАЧА 7.5. Вычислить значение многочлена $x^6-x^5+3x^4-8x^2+x-10$ в точках $x_1=-1, x_2=1$ (листинг 7.6).

```
>>> p=[1 -1 3 0 -8 1 -10];
x=[-1,1];
polyval(p,x)
>>>ans =
  -14  -14
```

Листинг 7.6

Вычислить *производную от многочлена* позволяет функция

`polyder(p)`

где `p` - многочлен степени *n*. Функция формирует вектор коэффициентов многочлена, являющегося производной от `p`.

Производную произведения двух векторов вычисляет функция

```
polyder(p1, p2)
```

где `p1` и `p2` — многочлены.

Вызов функции в общем виде

```
[q, r]= polyder(p1, p2)
```

приведет к вычислению *производной от частного* `p1` на `p2` и выдаст результат в виде отношения полиномов `q` и `r`.

ЗАДАЧА 7.6. Вычислить производную от многочлена $x^6-x^5+3x^4-8x^2+x-10$ Листинг 7.7 показал, что решением задачи является многочлен

$$6x^5-5x^4+12x^3-16x+1\ .$$

```
>>> p=[1 -1 3 0 -8 1 -10];
polyder(p)
>>>ans =
    6   -5   12    0  -16    1
```

Листинг 7.7

ЗАДАЧА 7.7. Вычислить производную от многочлена $(3x^4-7x^2+5)(x^3+2x-1=0)$.

Листинг 7.8 показал, что решением задачи является многочлен

$$21x^6-5x^4-12x^3-27x^2+14x+10\ .$$

```
>>> p1=[3 0 -7 0 5];
p2=[1 0 2 -1];
polyder(p1,p2)
>>>ans =
   21    0   -5  -12  -27   14   10
```

Листинг 7.8

ЗАДАЧА 7.8. Вычислить производную от выражения $\dfrac{x^3+1}{x^4-3x^3+3x^2-x}$.

Из листинга 7.9 видно, что решение задачи имеет вид

$$\frac{-x^4-2x^3-4x+1}{x^6-4x^5+6x^4-4x^3+x^2}\ .$$

```
>>> p1=[1 0 0 1];
p2=[1 -3 3 -1 0];
[q,r]=polyder(p1,p2)
>>>q =
  -1  -2   0  -4   1
r =
  1  -4   6  -4   1  0  0
```

Листинг 7.9

Взять *интеграл от многочлена* позволяет функция

```
polyint(p [,K])
```

где `p` - многочлен степени *n*, `K` — постоянная интегрирования, значение `K` по умолчанию равно нулю. Функция формирует вектор коэффициентов многочлена, являющегося интегралом от `p`.

ЗАДАЧА 7.9. Найти $\int (x^2+2x+3)\,dx$.

Согласно листингу 7.10 решение имеет вид:

$$\int (x^2+2x+3)\,dx=0.333x^3+x^2+3x\ .$$

Если определить значение постоянной интегрирования (листинге 7.11), то решение будет таким:

$$\int (x^2+2x+3)\,dx=0.333x^3+x^2+3x+5 \quad .$$

```
>>> p=[1 2 3];
polyint(p)
>>> ans
0.33333 1.00000 3.00000 0.00000
```

Листинг 7.10

```
>>> p=[1 2 3];
polyint(p,5)
>>>ans =
   0.33333   1.00000   3.00000   5.00000
```

Листинг 7.11

Построить многочлен по заданному вектору его корней позволяет функция

`poly(x)`

где `x` — вектор корней искомого полинома.

ЗАДАЧА 7.10. Записать алгебраическое уравнение, если известно, что его корни $x_1=-2, x_2=3$. Согласно листингу 7.12 решение задачи имеет вид: $x^2-x-6=0$

```
>>> x=[-2 3];
poly(x)
>>>ans =
   1  -1  -6
```

Листинг 7.12

Решить *алгебраическое уравнение* $P(x)=0$ можно при помощи встроенной функции

`roots(p)`

где `p` - многочлен степени *n*. Функция формирует вектор, элементы которого являются корнями заданного полинома.

ЗАДАЧА 7.11. Решить алгебраическое уравнение $x^2-x-6=0$.

Из листинга 7.13 видно, что значения $x_1=-2, x_2=3$ являются решением уравнения.

```
p=[1 -1 -6];
roots(p)
>>>ans =
   3
  -2
```

Листинг 7.13

ЗАДАЧА 7.12. Найти корни полинома $2x^3-3x^2-12x-5=0$.

Найдем корни полинома, так как показано в листинге 7.14.

```
>>> p=[2 -3 -12 -5];
x=roots(p)
>>>x =
   3.44949
  -1.44949
  -0.50000
```

Листинг 7.14

Графическое решение заданного уравнения показано в листинге 7.15и на рис. 7.1. Точки пересечения графика с осью абсцисс и есть корнями уравнения. Не трудно заметить, что графическое решение совпадает с аналитическим (листинг 7.14).

```
cla;
okno1=figure();
x=-2:0.1:5.5;
y=2*x.^3-3*x.^2-12*x-5;
```

```
pol=plot(x,y);
set(pol,'LineWidth',3,'Color','k')
set(gca,'xlim',[-2,4]);
set(gca,'ylim',[-30,30]);
set(gca,'xtick',[-2:0.5:4]);
set(gca,'ytick',[-30:5:30]);
grid on;
xlabel('x');ylabel('y');
title('Plot y=2*x^3-3*x^2-12*x-5');
```

Листинг 7.15

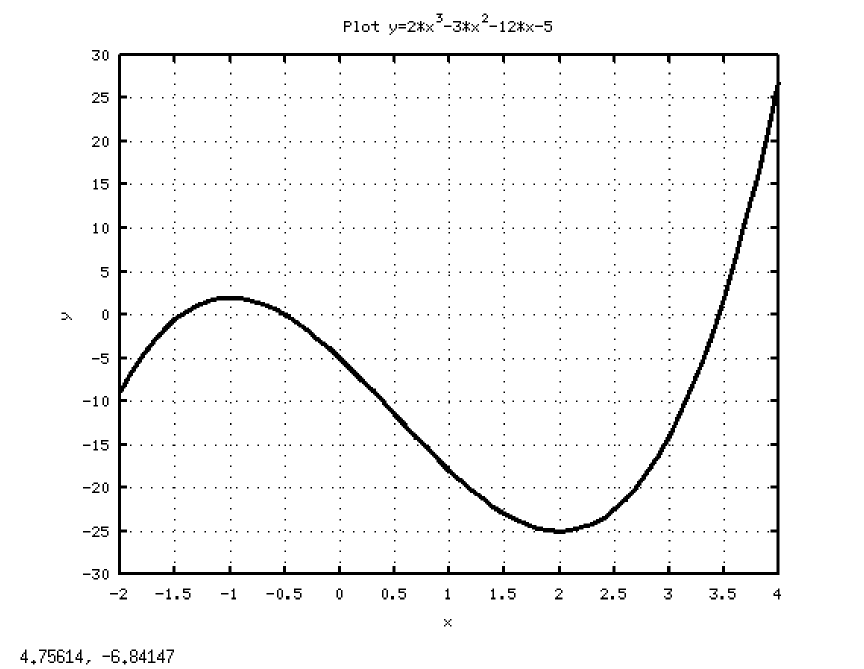

Рисунок *7.1*. Графическое решение задачи *7.12*

ЗАДАЧА 7.13. Найти решение уравнения $x^4+4x^3+4x^2-9=0$.

Графическое решение задачи было получено при помощи последовательности команд приведенных в листинге 7.16.

```
okno1=figure();
x=-4:0.1:2;
y=x.^4+4*x.^3+4*x.^2-9;
cla;
pol=plot(x,y);
set(pol,'LineWidth',3,'Color','k')
set(gca,'xlim',[-4,2]);
set(gca,'ylim',[-10,5]);
set(gca,'xtick',[-4:0.5:2]);
set(gca,'ytick',[-10:1:5]);
grid on;
xlabel('x');ylabel('y');
title('Plot y=x^4+4*x^3+4*x^2-9');
```

Листинг 7.16

На рис. 7.2 видно, что заданное алгебраическое уравнение имеет два действительных корня. Аналитическое решение задачи, представленное в листинге 7.17 показывает не только действительный, но и комплексные корни.

```
p=[1 4 4 0 -9];
x=roots(p)
>>>x =
  -3.00000 + 0.00000i
  -1.00000 + 1.41421i
  -1.00000 - 1.41421i
   1.00000 + 0.00000i
```

Листинг 7.17

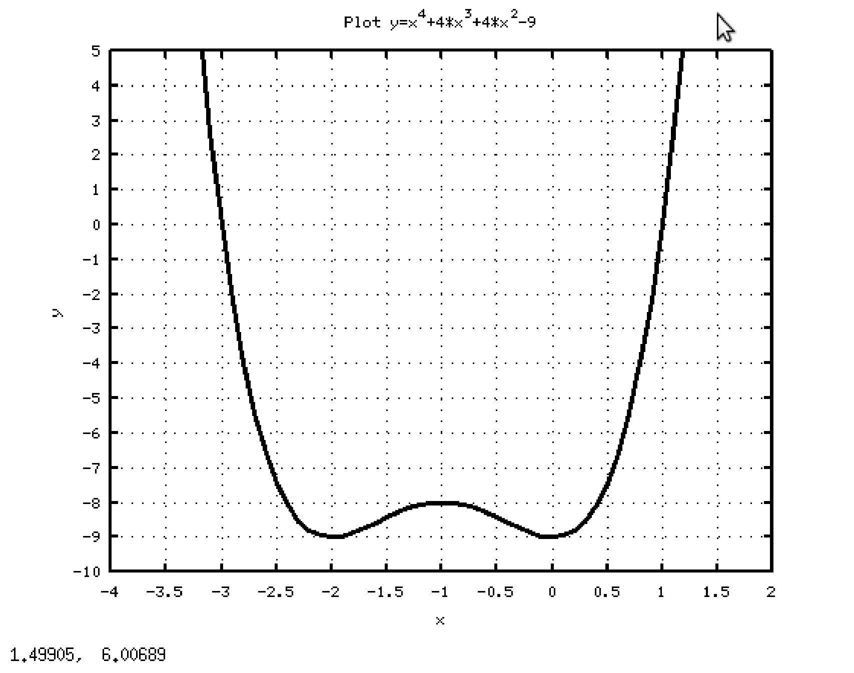

Рисунок *7.2*. Графическое решение задачи *7.13*

7.2 Решение трансцендентных уравнений

Уравнение, в котором неизвестное входит в аргумент трансцендентных функций, называется *трансцендентным уравнением*. К трансцендентным уравнениям принадлежат показательные, логарифмические, тригонометрические.

Для *решения трансцендентных уравнений* вида $f(x)=0$ в Octave существует функция

`fzero(name, x0)` или `fzero(name, [a, b])`,

где `name` – имя функции, вычисляющей левую часть уравнения, `x0` – начальное приближение к корню или интервал изоляции корня `[a, b]`.

Если функция вызывается в формате:

`[x, y]= fzero(name, x0)`,

то здесь `x` – корень уравнения, `y` – значение функции в точке `x`.

ЗАДАЧА 7.14. Найти решение уравнения:

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

Начнем решение данного трансцендентного уравнения с определения интервала изоляции корня. Воспользуемся для этого графическим методом. Построим график функции, указанной в левой части уравнения (листинга 7.18), создав предварительно функцию для ее определения.

```
%Функция для вычисления левой части уравнения f(x)=0
```

```
function y=f1(x)
y=((2*x-3).^2).^(1/3)-((x-1).^2).^(1/3);
end;
%Построение графика функции f(x)
okno1=figure();
x=-1:0.1:3;
y=f1(x);
cla;
pol=plot(x,y);
set(pol,'LineWidth',3,'Color','k')
set(gca,'xlim',[-1,3]);
set(gca,'ylim',[-1,1.5]);
set(gca,'xtick',[-1:0.5:3]);
set(gca,'ytick',[-1:0.5:1.5]);
grid on;
xlabel('x');ylabel('y');
title('Plot y=(2x-3)^(2/3)-(x-1)^(2/3)');
```

Листинг 7.18

На графике (рис. 7.3) видно, что функция $f(x)$ дважды пересекает ось Ox. Первый раз на интервале $[1, 1.5]$, второй - $[1.5, 2.5]$.

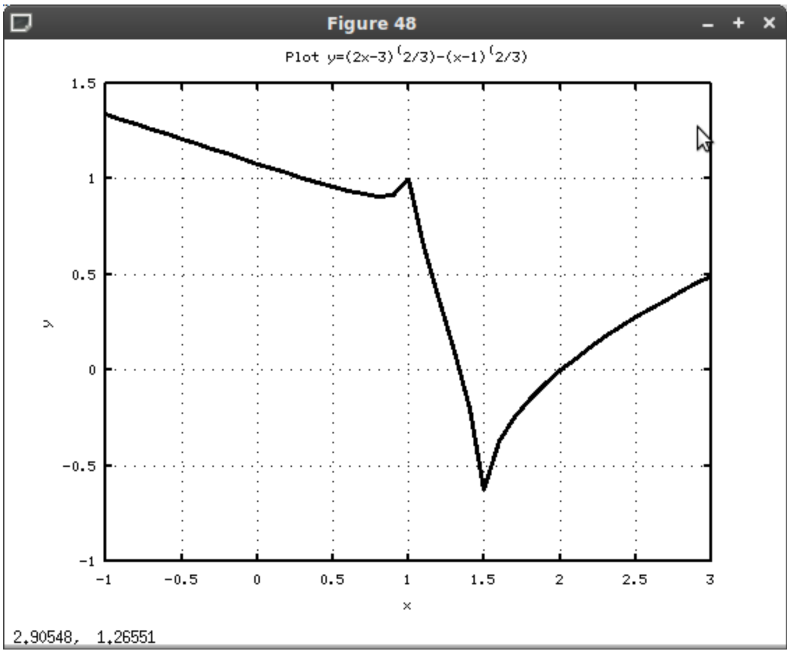

Рисунок *7.3*. Графическое решение задачи *7.14*

Уточним корни, полученные графическим методом. Создадим функцию, вычисляющую левую часть заданного уравнения (листинг 7.18) и обратимся к функции `fzero`, указав в качестве параметров имя созданной функции и число близкое к первому корню:

```
x1=fzero('f1', 1)
>>>x1=1.3333
```

Листинг 7.19

Теперь применим функцию `fzero`, указав в качестве параметров имя функции, вычисляющей левую часть тождества и интервал изоляции второго корня:

```
x2=fzero('f1', [1.5 2.5])
>>>x1= 2
```

Листинг 7.20

Не трудно заметить, что и в первом и во втором случае функция `fzero` правильно нашла корни заданного уравнения.

Листинг 7.21 содержит пример некорректного обращения к функции `fzero`, здесь интервал изоляции корня задан неверно. На графике видно, что на концах этого интервала функция знак не меняет. Или, другими словами, выбранный интервал содержит сразу два корня.

```
fzero('f1', [1 3])
>>>error: fzero: not a valid initial bracketing
error: called from:
error:   /usr/share/octave/3.2.3/m/optimization/fzero.m at
line 137, column 5
```

Листинг 7.21

В листинге 7.22 приведен пример обращения к функции `fzero` в полном формате:

```
[X(1),Y(1)]=fzero('f1', [1 1.5]);
[X(2),Y(2)]=fzero('f1', [1.5 2.5]);
X
Y
%Решение уравнения
>>>X =
   1.3333   2.0000
%Значения функции в точке X
>>>Y =
  -2.3870e-15   0.0000e+00
```

Листинг 7.22

ЗАДАЧА 7.15. Найти решение уравнения $x^4+4x^3+4x^2-9=0$.

Как видим, левая часть уравнения представляет собой полином. В задаче 7.13 было показано, что данное уравнение имеет четыре корня: два действительных и два мнимых (листинг 7.17).

Листинге 7.23 демонстрирует решение алгебраического уравнения при помощи функции `fzero`. Не трудно заметить, что результатом работы функции являются только действительные корни. Графическое решение (рис. 7.2) подтверждает это: функция дважды пересекает ось абсцисс.

```
>>> function y=f2(x)
y=x.^4+4*x.^3+4*x.^2-9;
end;
[X(1),Y(1)]=fzero('f2', [-4 -2]);
[X(2),Y(2)]=fzero('f2', [0 2]);
X
Y
>>>X =
  -3   1
>>>Y =
   0   0
```

Листинг 7.23

7.3 Решение систем нелинейных уравнений

Напомним, что 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*.

Несложную систему элементарной подстановкой можно привести к нелинейному уравнению. Рассмотрим несколько примеров, в которых описан такой прием решения нелинейной системы.

ЗАДАЧА 7.16. Решить систему уравнений:

$$\left\{\begin{matrix} x^2+2y^2=1 \\ x-y=1 \end{matrix}\right\}$$

Найдем графическое решение с помощью команд листинга 7.24. На рис.7.4 видно, что система имеет два решения.

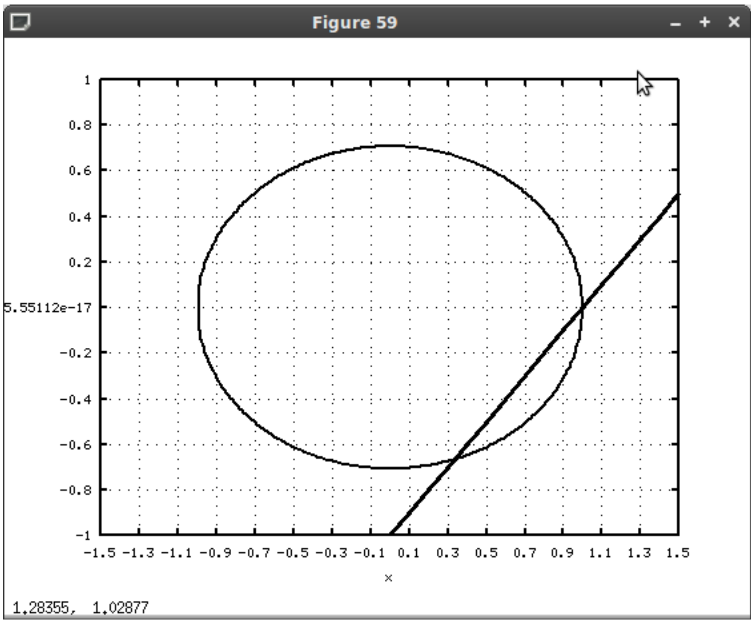

Рисунок *7.4*. Решение системы нелинейных уравнений из задачи *7.16*

```
%Верхняя часть эллипса
function y=f1(x)
y=sqrt((1-x.^2)/2);
end;
%Нижняя часть эллипса
function y=f2(x)
y=-sqrt((1-x.^2)/2);
end;
%Прямая
function y=f3(x)
y=x-1;
end;
%Построение графика
okno1=figure();
x1=-1:0.01:1; x2=-1.5:0.1:1.5;
cla;
L1=plot(x1,f1(x1),x1,f2(x1));
```

```
set(L1,'LineWidth',2,'Color','k')
hold on
L2=plot(x2,f3(x2));
set(L2,'LineWidth',3,'Color','k')
set(gca,'xlim',[-1.5,1.5]);
set(gca,'ylim',[-1,1]);
set(gca,'xtick',[-1.5:0.2:1.5]);
set(gca,'ytick',[-1:0.2:1]);
grid on;
xlabel('x');ylabel('y');
```

Листинг 7.24

Не сложно убедиться, что данная система легко сводится к одному уравнению:

$$\{y=x-1, x^2+2y^2=1\} \Rightarrow x^2+2(x-1)^2-1=0 \Rightarrow 3x^2-4x+1=0 \quad .$$

Решив это уравнение с помощью функции `roots`, найдем значения *x*. Затем подставим их в одно из уравнений системы, например во второе, и тем самым вычислим значения *y*. Листинг 7.25 содержит решение данной задачи. Понятно, что система имеет два решения $x_1=1, y_1=0$ и $x_2=0.333, y_1=-0.666$ (рис. 7.4). Графическое решение уравнения (листинг 7.26), к которому сводится система, показано на рис. 7.5.

```
>>> p=[3 -4 1];
x=roots(p)
y=x-1
>>>x =
   1.00000
   0.33333
>>>y =
  -1.1102e-16
  -6.6667e-01
```

Листинг 7.25

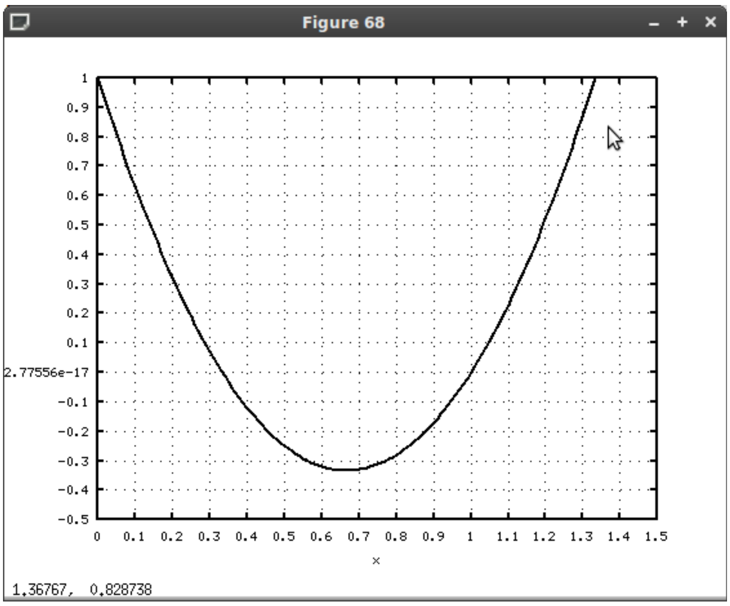

Рисунок *7.5:* Графическое решение задачи *7.16*

```
function y=f(x)
y=3*x.^2-4*x+1;
end;
```

```
okno1=figure();
x=0:0.01:1.5;
cla;
L=plot(x,f(x));
set(L,'LineWidth',2,'Color','k')
set(gca,'xlim',[0,1.5]);
set(gca,'ylim',[-0.5,1]);
set(gca,'xtick',[0:0.1:1.5]);
set(gca,'ytick',[-0.5:0.1:1]);
grid on;
xlabel('x');ylabel('y');
```

Листинг 7.26

ЗАДАЧА 7.17. Решить систему уравнений:

$$\left\{\begin{matrix} \sin(x+1)-y=1.2 \\ 2x+\cos(y)=2 \end{matrix}\right\}.$$

Проведем элементарные алгебраические преобразования и представим систему в виде одного уравнения:

$$\{y=\sin(x+1)-1.2, 2x+\cos(y)-2=0\} \Rightarrow 2x+\cos(\sin(x+1)-1.2)-2=0$$

Рис. 7.6 содержит графическое решение уравнения (листинг 7.27), к которому сводится система, его удобно использовать для выбора начального приближения функции `fzero`.

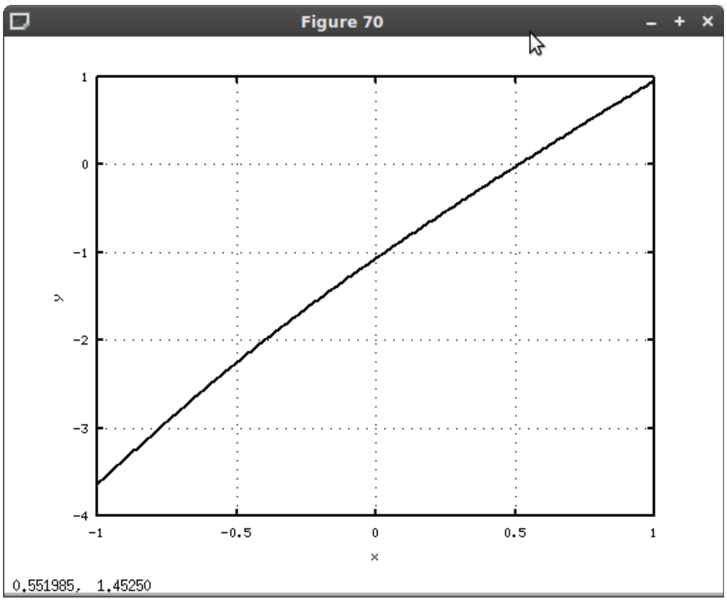

Рисунок *7.6*. Графическое решение задачи *7.17*

```
function y=fun(x)
z=sin(x+1)-1.2;
y=2*x+cos(z)-2;
end;
okno1=figure();
x=-1:0.01:1;
cla;
L=plot(x,fun(x));
set(L,'LineWidth',2,'Color','k')
grid on;xlabel('x');ylabel('y');
```

Листинг 7.27

Листинг 7.28 содержит решение заданной системы. Здесь значения x вычисляется при помощи функции `fzero`, а значение y определяется из первого уравнения системы.

```
>>>  %Решение системы
X=fzero('fun',0)
Y=sin(X+1)-1.2
>>>X =  0.51015
>>>Y = -0.20184
```

Листинг 7.28

Решить систему нелинейных уравнений, или одно нелинейное уравнение, в Octave можно с помощью функции

```
fsolve(fun, x0)
```

где `fun` – имя функции, которая определяет левую часть уравнения $f(x)=0$ или системы уравнений $\mathbf{F}(\mathbf{x})=\mathbf{0}$ (она должна принимать на входе вектор аргументов и возвращать вектор значений), `x0` – вектор приближений, относительно которого будет осуществляться поиск решения.

ЗАДАЧА 7.18. Найти решение системы нелинейных уравнений:

$$\begin{cases} \cos(x)+2y=2 \\ \dfrac{x^2}{3}-\dfrac{y^2}{3}=1 \end{cases} .$$

Решим систему графически, для чего выполним перечень команд указанных в листинге 7.29. Результат работы этих команд показан на рис. 7.7. Понятно, что система имеет два корня.

```
%Уравнения, описывающие линии гиперболы
function y=f1(x)
y=sqrt(x.^2-3);
end;
function y=f2(x)
y=-sqrt(x.^2-3);
end;
%Уравнение косинусоиды
function y=f3(x)
y=1-cos(x)/2;
end;
%Построение графика
okno1=figure();
x1=-5:0.001:-sqrt(3);
x2=sqrt(3):0.001:5;
x3=-5:0.1:5;
cla;
%Гипербола
L1=plot(x1,f1(x1),x1,f2(x1),x2,f1(x2),x2,f2(x2));
set(L1,'LineWidth',3,'Color','k')
hold on
%Косинусоида
L2=plot(x3,f3(x3));
set(L2,'LineWidth',3,'Color','k')
grid on;
xlabel('x');ylabel('y');
```

Листинг 7.29

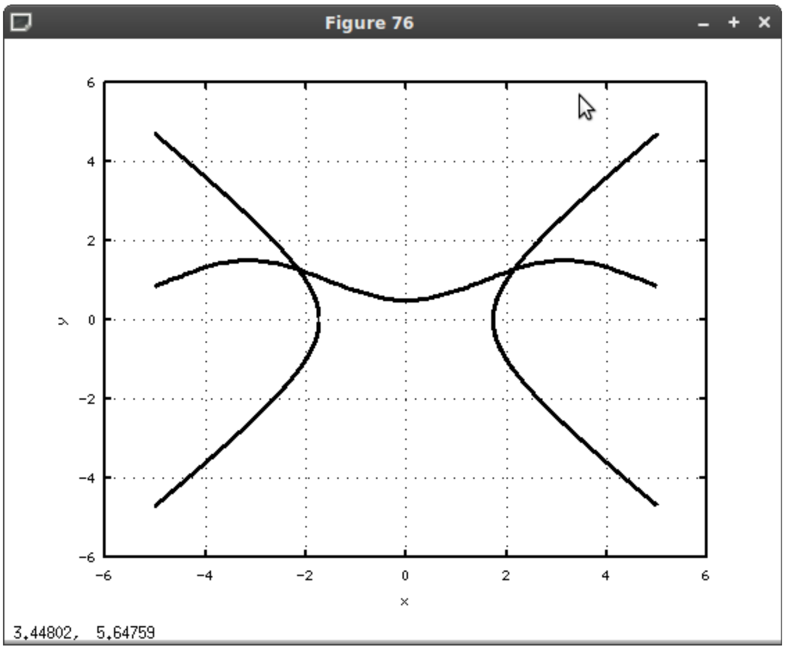

Рисунок *7.7*. Графическое решение задачи *7.18*

Составим функцию, соответствующую левой части системы (листинг 7.30). Здесь важно помнить, что все уравнения должны иметь вид $F(x)$=0. Кроме того, обратите внимание, что `x` и `y` в этой функции — векторы (`x` — вектор неизвестных, `y` — вектор решений).

```
function [y]=fun(x)
      y(1)=cos(x(1))+2*x(2)-2;
      y(2)=x(1)^2/3-x(2)^2/3-1;
end;
```

Листинг 7.30

Теперь решим систему (листинг 7.31), указав в качестве начального приближения в начале вектор $[-3,-1]$, затем $[1,3]$.

```
[X1_Y1]=fsolve('fun', [-3 -1])
[X2_Y2]=fsolve('fun', [1 3])
>>>X1_Y1 =
  -2.1499   1.2736
>>>X2_Y2 =
   2.1499   1.2736
```

Листинг 7.31

Понятно, что решением задачи являются пары $x_1=-2.15, y_1=1.27$ и $x_2=2.15, y_2=1.27$, что соответствует графическому решению (рис. 7.7).

Если функция решения нелинейных уравнений и систем имеет вид

```
[x, f, ex]=fsolve(fun, x0)
```

то здесь `x` – вектор решений системы, `f` – вектор значений уравнений системы для найденного значения `x`, `ex` – признак завершения алгоритма решения нелинейной системы, отрицательное значение параметра `ex` означает, что решение не найдено, ноль – досрочное прерывание вычислительного процесса при достижении максимально допустимого числа итераций, положительное значение подтверждает, что решение найдено с заданной точностью.

ЗАДАЧА 7.19. Решить систему нелинейных уравнений:

$$\begin{Bmatrix} x_1^2+x_2^2+x_3^2=1 \\ 2x_1^2+x_2^2-4x_3=0 \\ 3x_1^2-4x_2^2+x_3^2=0 \end{Bmatrix}.$$

Листинг 7.32 содержит функцию заданной системы и ее решение. Обратите внимание на выходные параметры функции `fsolve`. В нашем случае значения функции `f` для найденного решения `x` близки к нулю и признак завершения `ex` положительный, значит, найдено верное решение.

```
>>> function f=Y(x)
f(1)=x(1)^2+x(2)^2+x(3)^2-1;
f(2)=2*x(1)^2+x(2)^2-4*x(3);
f(3)=3*x(1)^2-4*x(2)+x(3)^2;
end
[x,f,ex]=fsolve('Y',[0.5 0.5 0.5])
>>>x =
   0.78520   0.49661   0.36992
f =
   1.7571e-08   3.5199e-08   5.2791e-08
ex =  1
```

Листинг 7.32

ЗАДАЧА 7.20. Решить систему:

$$\begin{Bmatrix} x^2+y^2=1 \\ 2\sin(x-1)+y=1 \end{Bmatrix}.$$

Графическое решение системы (рис. 7.8) показало, что она корней не имеет. Рисунок был получен с помощью команд листинга 7.33.

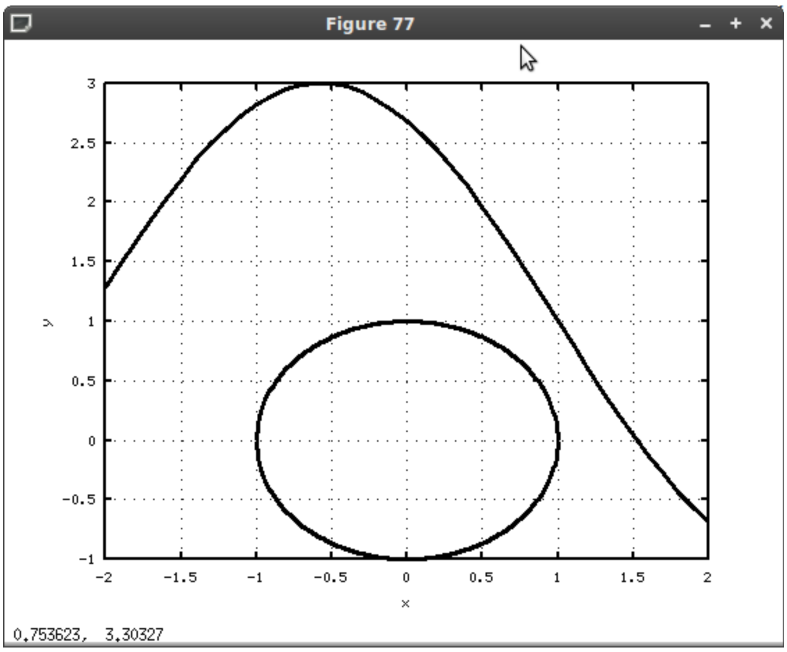

Рисунок *7.8*. Графическое решение задачи *7.20*

```
%Уравнения линий окружности
function y=f1(x)
y=sqrt(1-x.^2);
end;
function y=f2(x)
```

```
y=-sqrt(1-x.^2);
end;
%Уравнение синусоиды
function y=f3(x)
y=1-2*sin(x-1);
end;
okno1=figure();
x1=-1:0.01:1;
x3=-2:0.1:2;
cla;
%Окружность
L1=plot(x1,f1(x1),x1,f2(x1));
set(L1,'LineWidth',3,'Color','k')
hold on
%Синусоида
L2=plot(x3,f3(x3));
set(L2,'LineWidth',3,'Color','k')
grid on;
xlabel('x');ylabel('y');
```

Листинг 7.33

Однако применение к системе функции `fsolve` дает положительный ответ, что видно из листинга 7.34. Происходит это потому, что алгоритм, реализованный в этой функции, основан на минимизации суммы квадратов компонент вектор–функции. Следовательно, функция `fsolve` в этом случае нашла точку минимума, а наличие точки минимума не гарантирует существование корней системы в ее окрестности.

```
>>> function [y]=fun(x)
y(1)=x(1)^2+x(2)^2-1;
y(2)=2*sin(x(1)-1)+x(2)-1;
end;
[X1_Y1]=fsolve('fun', [1 1])
>>>X1_Y1 =
   1.04584   0.52342
```

Листинг 7.34

7.4 Решение нелинейных уравнений и систем в символьных переменных

Напомним, что для работы с символьными переменными в Octave должен быть подключен специальный пакет расширений **octave-symbolic**. Процедура установки пакетов расширений описана в первой главе. Техника работы с символьными переменными описана в п. 2.7 второй главы.

Для *решения системы нелинейных уравнений* или одного *нелинейного уравнения* можно воспользоваться функцией `symfsolve`.

ЗАДАЧА 7.21. Решить уравнение $\frac{e^x}{5}-3(2x-1)=0$.

Команды, с помощью которых выполнено графическое (рис.) и аналитическое решение представлены в листинге 7.35.

```
>>> clear all;
clf;cla;
symbols
x=sym("x");
```

```
y=Exp(x)/5-3*(2*x-1);
L=ezplot('exp(x)/5-3*(2*x-1)');
set(L,'LineWidth',2,'Color','k')
set(gca,'xlim',[-2,6]);
set(gca,'ylim',[-20,50]);
set(gca,'xtick',[-2:0.5:6]);
set(gca,'ytick',[-20:10:50]);
grid on;
xlabel('x');ylabel('y');
q1 = symfsolve(y,0)
q2 = symfsolve(y,4)
>>>q1 =  0.55825
>>>q2 =  4.8777
```

Листинг 7.35

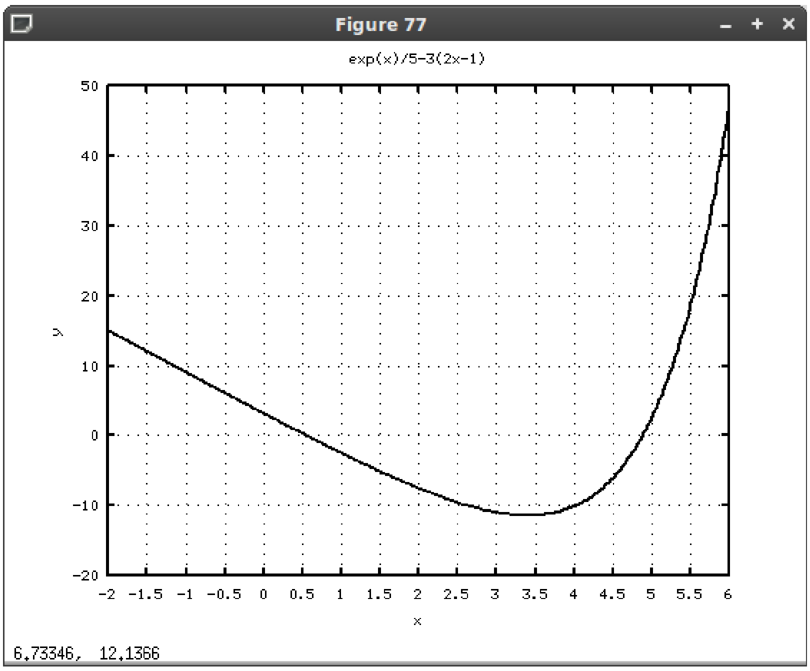

Рисунок *7.9*. Графическое решение задачи *7.21*

ЗАДАЧА 7.22. Решить систему:

$$\begin{cases} x^2+y^2+3x-2y=4 \\ x+2y=5 \end{cases}.$$

Решение системы (листинг 7.36) показало, что она имеет два корня $x_1=1, y_1=2$ и $x_2=-2.2, y_2=3.6$, что соответствует графическому решению (рис. 7.10).

```
>>> clear all;
clf;cla;
symbols
x=sym("x");
y=sym("y");
L1=ezplot('x^2+y^2+3*x-2*y-4');
set(L1,'LineWidth',2,'Color','k')
hold on
L2=ezplot('x+2*y-5');
set(L2,'LineWidth',2,'Color','k')
set(gca,'xlim',[-5,4]);
```

```
set(gca,'ylim',[-2,5]);
set(gca,'xtick',[-5:0.5:4]);
set(gca,'ytick',[-2:0.5:5]);
grid on;
xlabel('x');ylabel('y');
title('x^2+y^2+3x-2y=4, x+2y=5')
f1=x^2+y^2+3*x-2*y-4;
f2=x+2*y-5;
q1 = symfsolve(f1,f2,{x==0,y==1})
q2 = symfsolve(f1,f2,{x==-1,y==3})
>>>q1 =
   1.0000   2.0000
>>>q2 =
  -2.2000   3.6000
```

Листинг 7.36

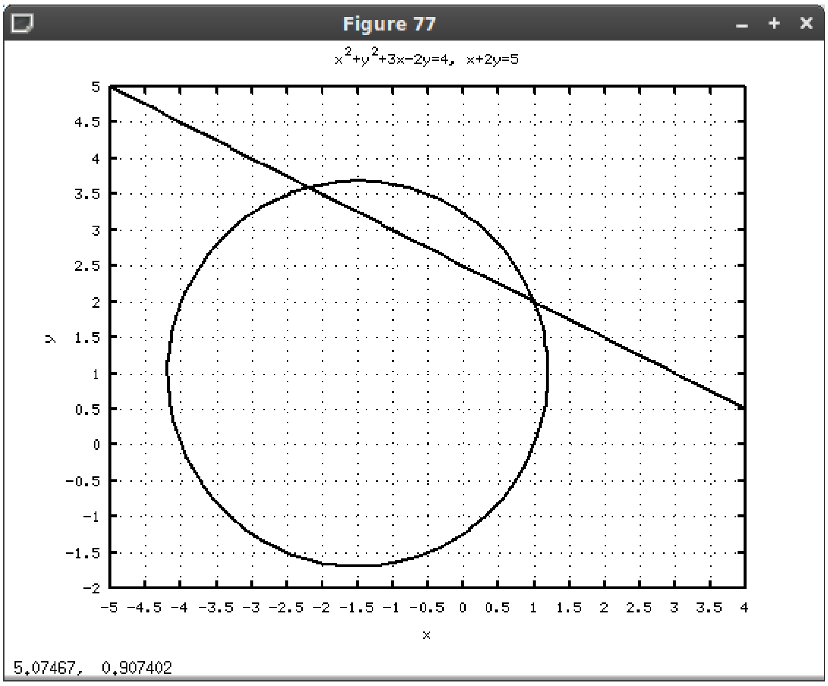

Рисунок *7.10*. Графическое решение задачи *7.22*