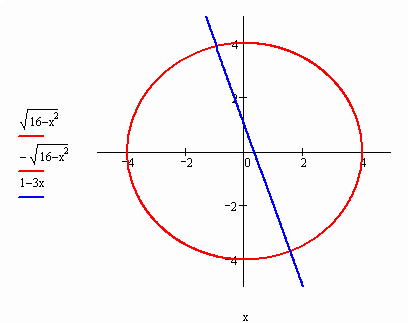
#### ***Тема 5.10. Технология решения систем нелинейных уравнений средствами MatLab***

Для решения систем нелинейных уравнений в среде программирования **MatLab** используется процедура fsolve( ). Процедура fsolve( ) решает уравнения и системы уравнений.

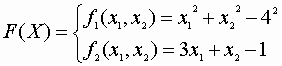
**Пример 4.10-1 Определить точки пересечения двух линий.**

Задача сводится к решению системы уравнений:



Необходимо подготовить файл-функцию для этой системы уравнений. Преобразованная система уравнений для составления файл-функции будет такой:

.



После такой подготовки составляем в редакторе файл-функцию.

|  |
| --- |
| Пример 4.10-1 |
| functionf=myfyn2(x)  f(1)=x(1)^2+x(2)^2-4^2;  f(2)=3\*x(1)+x(2)-1;  %Вызов процедуры fsolve со следующими параметрами:  >>x=fsolve(@myfyn2,[-1 4],optimset('Display','off'))  x =  -0.9610 3.8829  %Вторая точка пересечения:  >>x=fsolve(@myfyn2,[1 -4],optimset('Display','off'))  x =  1.5610 -3.6829  >> |

Единственный недостаток процедуры fsolve( )видится в том, что невозможно получить все решения системы уравнений сразу. Вариант решения зависит от начальной точки итерационного процесса.

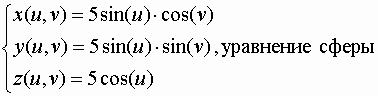
**Пример 4.10-2. Решить при помощи процедуры fsolve( ) следующую задачу:**

**найти линию пересечения сферы и параболоида вращения .**

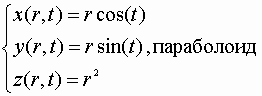


При решении задач такого типа в большинстве случаев, трудоемкость и успех решения зависят от удачно выбранной параметризации.

Данная формулировка задачи позволяет работать непосредственно с координатами . Исключить координату и в цикле, при фиксированномx, искать значениеy. Но при этом возникает двойственность решения и потребуются дополнительные усилия, чтобы корректно решить эту проблему. Произведем перепараметризацию этих поверхностей:



.



На первый взгляд, параметры и обозначают одно и тоже – угол с осью абцисс в плоскости , но не надо забывать, что параметры и относятся к сферической системе координат, а параметры и *t* - к цилиндрической. Конечно, если бы мы решали аналитически, приравняв параметры и , нашли значение и сказали бы, что решением является окружность радиуса на высоте . Об этом можно рассуждать достаточно долго и не в рамках наших материалов. А нам необходимо составить файл-функцию решения этой задачи, для передачи ее в процедуру **fsolve.**



|  |
| --- |
| **Пример 4.10-2** |
| **function f=myfyn3(p,t)**  **f(1)=5\*sin(p(1))\*cos(p(2))-p(3)\*cos(t);**  **f(2)=5\*sin(p(1))\*sin(p(2))-p(3)\*sin(t);**  **f(3)=5\*cos(p(1))-p(3)^2;** |

Теперь составим файл-функцию, в которой будет вызвана процедура **fsolve( )**, найдено решение и выведены все геометрические фигуры на один график.

|  |
| --- |
| **Пример 4.10-3** |
| **function myfyn4()**  **T=0:2\*pi/100:2\*pi;**  **M=length(T);**  **for i=1:M**  **c=T(i);**  **Vector=fsolve(@myfyn3,[1,1,1],optimset('Display','off'),c);**  **U(i)=Vector(1);**  **V(i)=Vector(2);**  **R(i)=Vector(3);**  **end;**  **for i=1:M**  **X(i)=5\*sin(U(i))\*cos(V(i));**  **Y(i)=5\*sin(U(i))\*sin(V(i));**  **Z(i)=5\*cos(U(i));**  **end;**  **u=0:pi/20:pi; m=length(u);**  **v=0:2\*pi/20:2\*pi; n=length(v);**  **for i=1:m**  **for j=1:n**  **Sx(i,j)=5\*sin(u(i))\*cos(v(j));**  **Sy(i,j)=5\*sin(u(i))\*sin(v(j));**  **Sz(i,j)=5\*cos(u(i));**  **end;**  **end;**  **r=0:5/20:sqrt(10); t=0:2\*pi/20:2\*pi; mP=length(r); nP=length(t);**  **for i=1:mP**  **for j=1:nP**  **Px(i,j)=r(i)\*cos(t(j));**  **Py(i,j)=r(i)\*sin(t(j));**  **Pz(i,j)=r(i)^2;**  **end;**  **end;**  **mesh(Sx,Sy,Sz);**  **hold on**  **mesh(Px,Py,Pz);**  **hold on**  **plot3(X,Y,Z,'bo-');** |

После этого остается вызвать эту файл-функцию из командной строки или из другого скрипт-файла.

|  |
| --- |
| **myfyn4** |

Таким образом, наша функция myfyn4( ) построила сферу, параболоид и линию их пересечения. Следует отметить, что масштаб по координатным осям разный, поэтому сфера выглядит как эллипсоид. Все параметры графика регулируются либо программно, либо в интерактивном режиме в среде программирования **MatLab**. Можно отрегулировать выводимый диапазон значений, цвет, толщину линий и так далее.