#### ***Тема 5.5. Технология решения дифференциальных уравнений в среде MatLab***

В MatLab имеется ряд функций для решения задачи Коши. Одна из них - ode45 - использует метод Рунге-Кутты четвёртого-пятого порядка точности с автоматическим выбором размера шага.

Обращение к ode45 выполняется следующим образом

[T,Y] = ode45(ODEFUN,TSPAN,Y0).

**Входные параметры процедуры ode45( ):**

ODEFUN - имя функции (в виде строчной переменной), задающей правую часть системы дифференциальных уравнений. Уравнения должны быть записаны в нормальной форме u' = ODEFUN(T,Y),где u' - вычисляемый вектор-столбец производных;   
TSPAN= [T0 TFINAL] - вектор, задающий интервал изменения независимой переменной;T0 - начальная точка;

TFINAL - последнее значение аргумента, при котором завершается расчёт;   
Y0 - вектор начальных значений зависимых переменных.

**Выходные параметры ode45( ):**

T - вектор, содержащий отсчёты аргумента в точках решения;

Y - массив, содержащий вычисленные значения u и u' в точках, соответствующих отсчетам независимой переменной в T.

Требования к точности и другие параметры численного решения задаются в MatLab по умолчанию. Изменить эти настройки позволяет дополнительный аргумент OPTIONS.

**Пример 4.5-1. Решить дифференциальное уравнение второго порядка,**

**описывающее колебания в нелинейной системе (например, автогенераторе), известное как уравнение Ван-дер-Поля: u" + (u2 - b) u' + u = 0 ,где u, u' и u" - функция, её первая и вторая производные по времени t, b - произвольный параметр. Параметр b определяет потери в системе, а слагаемое u² в скобках - её нелинейные свойства (например, рабочие характеристики активного элемента автогенератора).**

Данное дифференциальное уравнение второго порядка можно привести к системе двух дифференциальных уравнений первого порядка

y1' = ( 1 –(y2)2 ) y1 - y2 ,   
y2' = y1 ,где y1=u' и y2=u, а параметр b принят равным единице.

Для выполнения расчётов воспользуемся внешней функцией vdp1( ), записанной в m-файле vdp1.m. Эта функция, вычисляющая правую часть уравнения Ван-дер-Поля, имеет следующий вид:

|  |
| --- |
| **- Пример 4.5-1** |
| **function dydt = vdp1(t,y) dydt = [y(2); (1-y(1)^2)\*y(2)-y(1)]** |

Для получения решения дифференциального уравнения, описываемого функцией vdp1( ) на интервале 0 < t < 20, при начальных условиях y1 = u' = 2 и y2 = u = 0 следует ввести в MatLab следующую строку:

|  |
| --- |
| **Пример 4.5-1** |
| **[t,y]=ode45(@vdp1,[0 20],[2 0])** |

На рисунке представлены рассчитанные функцией ode45 и построенные с помощью команды plot зависимости y2(t) ≡ u(t) и y1(t) ≡ u'(t).

|  |
| --- |
|  |

График показывает, что с течением времени колебания в системе нарастают. Однако в дальнейшем рост амплитуды колебаний замедляется, и система переходит в устойчивое состояние, описываемое представленными фазовыми кривыми y2 (y1). Кроме рассмотренной функции ode45( )MatLab содержит ряд других функций для решения задачи Коши - ode23( ), ode113(), ode15s(), ode23s(), ode23t(), ode23tb().

**Пример 4.5-2.Решить систему дифференциальных уравнений**

**с начальными условиями**



|  |
| --- |
| **Пример 4.5-2** |
| **Function F=FF(t,x)**  **F=[-4\*x(1)-13\*x(2)+exp(t);x(1)];**  **end**  **%Формирование вектора начальных условий**  **X0=[1;-1];**  **interval=[0.25 2];**  **[T,X]=ode45(@FF,interval,X0);**  **plot(T,X(:,1), ': ',T,X(:,2), '-');**  **legend('y', 'x-Решение');**  **grid on;** |