MAT 1206 – Introduction to MATLAB

CHAPTER 04: Applications in Linear algebra and basic calculus

Lesson 1:

Content

- Solving Basic Algebraic Equations
- Solving Quadratic Equations
- Finding roots of polynomial equations
- Solving System of Equations
- Factorization of Algebraic Expressions
- Simplification of Algebraic Expressions

Solving Basic Algebraic Equations

The **solve** function is used for solving algebraic equations. **S = solve(eqn,var)** solves the equation **eqn** for the variable **var**. If you do not specify **var**, it determines the variable to solve for. Before solve the equation, the **syms** function is used to define symbolic variables.

For example, solve(x + 1 == 2, x) solves the equation x + 1 = 2 for x.

```
syms x
eqn = x + 1 == 2;
S = solve(eqn,x)
```

```
syms x
eqn = x + 3*x == 2 + 5*x;
S = solve(eqn)
```

```
syms x
eqn = x + 3*x == 2 + 5*x;
S = solve(eqn,x)
```

Solving Quadratic Equations

The **solve** function can also solve higher order equations. It is often used to solve quadratic equations. The function returns the roots of the equation in an array.

Example: Solving a Quadratic Equation x^2 - 4x + 3 = 0.

```
syms x
eqn = x^2 - 4*x + 3 == 0;
sol = solve(eqn, x);
disp(sol);
```

```
syms a b c x
eqn = a*x^2 + b*x + c == 0;
solx = solve(eqn, x)
```

```
syms c x
eqn = x^2 + 2*x + c == 10;
sol = solve(eqn, x);
disp(sol);
```

Finding roots of polynomial equations

r = **roots(p**) returns the roots of the polynomial represented by **p** as a column vector. Input p is a vector containing n+1 polynomial coefficients, starting with the coefficient of x^n. A coefficient of 0 indicates an intermediate power that is not present in the equation. For example, p = [3 2 -2] represents the polynomial $3x^2 + 2x - 2$.

Solve the equation $3x^2 - 2x - 4 = 0$

```
coefficients = [3 -2 -4];
solutions = roots(coefficients);
disp(solutions);
```

solve the quartic equation 2x^4 - 8x^3 + 8x^2 - 4x = 0 using the roots function.

```
coefficients = [2, -8, 8, -4, 0];
solutions = roots(coefficients);
disp(solutions);
```

Solving System of Equations

The solve function can also be used to generate solutions of systems of equations involving more than one variables.

Example:

$$2x + 3y = 7$$

$$4x - 2y = 2$$

```
syms x y
eqns = [2*x + 3*y == 7, 4*x - 2*y == 2];
sol = solve(eqns, x, y);
disp(sol);
```

Example:

$$2x + 3y - z = 5$$

$$x - y + 2z = 3$$

$$3x + 2y + 2z = 2$$

```
syms x y z
eqns = [2*x + 3*y - z == 5, x - y + 2*z == 3, 3*x + 2*y + 2*z == 2];
sol = solve(eqns, x, y, z);
disp(sol);
```

Factorization of Algebraic Expressions

The factor function factorizes an expression an expression.

```
syms x
expr = x^2 - 4;
factored_expr = factor(expr);
disp(factored_expr);
```

```
syms x
expr = x^2 - 5*x + 6;
factored_expr = factor(expr);
disp(factored_expr);
```

```
syms x
expr = 3*x^3 - 12*x^2 + 12*x;
factored_expr = factor(expr);
disp(factored_expr);
```

Simplification of Algebraic Expressions

The **simplify** function simplifies an expression.

```
syms x
simplified_expr =simplify((x^4-16)/(x^2-4));
disp(simplified_expr )
```

```
syms x
simplified_expr = simplify((x^2 + 2*x + 1) / (x + 1));
disp(simplified_expr);
```

```
syms theta
expr = sin(theta)^2 + cos(theta)^2;
simplified_expr = simplify(expr);
disp(simplified_expr);
```

```
syms a b
expr = exp(a) * exp(b);
simplified_expr = simplify(expr);
disp(simplified_expr);
```

Cont.

```
syms x
expr = sin(x)^4 + 2*sin(x)^2*cos(x)^2 + cos(x)^4;
simplified_expr = simplify(expr);
disp(simplified_expr);
```

```
syms x
expr = (sin(x) + sin(3*x))/(cos(x) + cos(3*x));
simplified_expr = simplify(expr);
disp(simplified_expr);
```

```
syms x
expr = (exp(x) * exp(2*x)) / (exp(3*x) * exp(4*x));
simplified_expr = simplify(expr);
disp(simplified_expr);
```

Questions/queries?