Chapter 5. Polynomials

Scilab provides robust tools for working with polynomials as formal mathematical objects. This chapter covers the definition, operations, and functions applications to polynomials and rational fractions.

1. Polynomials and rational fractions:

Scilab uses the predefined variable **%s** to represent the monomial of degree 1. Polynomials and rational fractions can be created through simple expressions:

```
--> p = 1 + %s + 2 * %s^2

 p = [polynomial] of s

  1 +s +2s^2

--> q=p/(1+%s)

 q = [rational] of s

   1 +s +2s^2

   ----------

      1 +s
```

To define a polynomial explicitly, use the **poly** command:

```
--> x = poly(0, 'x')

 x = [polynomial] of x

  x

--> p = x^2 + 2*x + 1

 p = [polynomial] of x

  1 +2x +x^2
```

A polynomial can also be defined by specifying its **roots**:

```
--> p1 = poly([0, 1, 2], 's', 'roots')

 p1 = [polynomial] of s

  2s -3s^2 +s^3
```

Alternatively, a polynomial can be created from its **coefficients**, listed in ascending order of degree:

```
--> p2 = poly([1, 2, 3], 's', 'coeff')

 p2 = [polynomial] of s

  1 +2s +3s^2
```

The characteristic polynomial of a square matrix **A** is defined as **det(A-xI)=p(x)**:

```
--> A = [1 2; 3 4]

 A = [2x2 double]

   1.   2.

   3.   4.

--> p = poly(A, 'x')

 p = [polynomial] of x

  -2 -5x +x^2
```

2. Zeros of a polynomial:

In Scilab, the zeros (or roots) of a polynomial are the values of **x** for which the polynomial equals zero. To find them, we use the function `roots()`. For example, consider the polynomial **P(x) =** $x^3-6x^2+11x-6$. Its coefficients are represented by the vector **`[-6 11 -6 1]`**. The instruction **`r = roots(p)`** gives the zeros of **P(x)**.

```
--> p = poly([-6 11 -6 1], 's', 'coeff')

 p = [polynomial] of s

  -6 +11s -6s^2 +s^3

--> roots(p)

 ans = [3x1 double]

   3. + 0.i

   2. + 0.i

   1. + 0.i
```

3. Operations on Polynomials:

Polynomials with the same variable can be added, subtracted, multiplied, divided, or raised to a power:

```
--> p1 = poly([0, 1, 2], 's', 'roots')
 p1 = [polynomial] of s
  2s -3s^2 +s^3
--> p2 = poly([1, 2, 3], 's', 'coeff')
 p2 = [polynomial] of s
  1 +2s +3s^2
--> p1 + p2
 ans = [polynomial] of s
  1 +4s +s^3
--> p1 - p2
 ans = [polynomial] of s
```

```
  -1 -6s^2 +s^3
--> p1*p2
 ans = [polynomial] of s
  2s +s^2 +s^3 -7s^4 +3s^5

--> p1 / p2
 ans = [rational] of s
   2s -3s^2 +s^3
   -------------
    1 +2s +3s^2
--> p1^4
 ans = [polynomial] of s
  16s^4 -96s^5 +248s^6 -360s^7 +321s^8 -180s^9 +62s^10
-12s^11 +s^12
```

Scilab provides commands for Euclidean and polynomial division:

pdiv(p1, p2) performs Euclidean division.

ldiv(p1, p2, k) divides polynomials following increasing powers.

```
--> pdiv(p1, p2)
 ans = [polynomial] of s
  -1.2222222 +0.3333333s
--> ldiv(p1,p2,4)
 ans = [4x1 double]
  -1.5
   1.75
  -0.875
   0.4375
```

Scilab offers several functions for polynomial manipulation:

Function	**Description**
`degree(p)`	Returns the degree of polynomial p.
`derivat(p)`	Computes the derivative of p.
`coeff(p)`	Returns the coefficients of p in ascending order.
`roots(p)`	Finds the roots of p.
`varn(p)`	Returns the variable name of p.
`factors(p)`	Factorizes p into simpler components.
`bezout(p, q)`	Computes the greatest common divisor.
`horner(p, t)`	Evaluates p at value t.
`simp(q)`	simplifies rational fractions q.