Polynomials

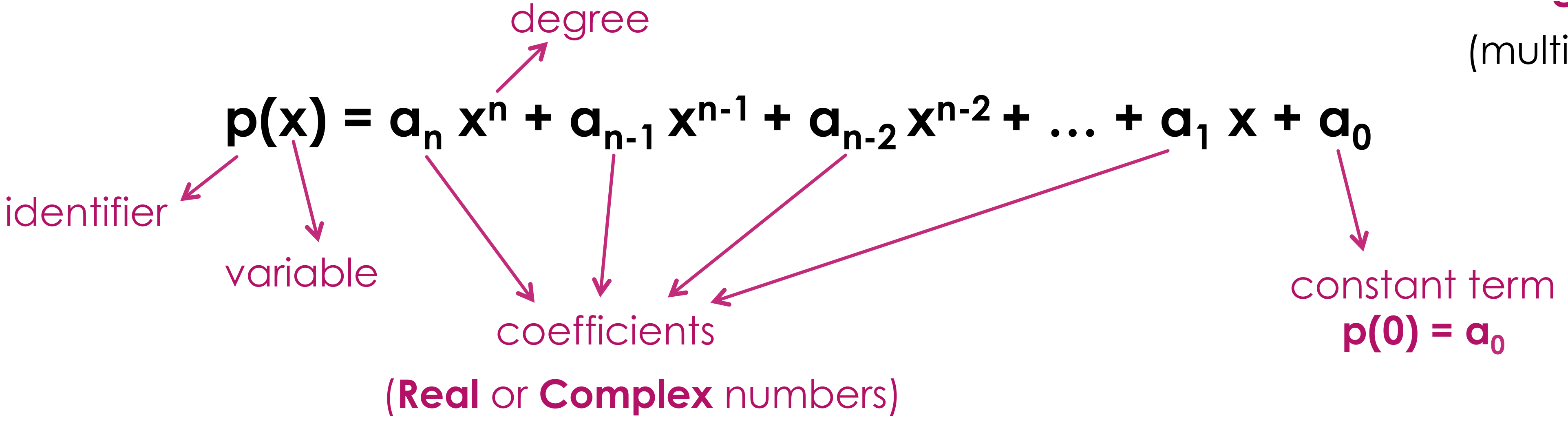
Polynomials

- ✓ Algebraic expressions (**single variable** or multi-variable)
- ✓ **Domain** of input variable: **real numbers** (more generally: **complex numbers**)
    - **Range** of output values: **real numbers** (more generally: **complex numbers**)
- ✓ Only **non-negative** integer powers (n = 0,1, 2, 3, 4, …)
- ✓ n is degree: deg(p(x)) = n

A polynomial of **deg = n** has maximum n **real** roots (**exactly n complex roots**)

Fundamental theorem of algebra

(multiplicity of roots)



degree

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_1 x + a_0$$

identifier

variable

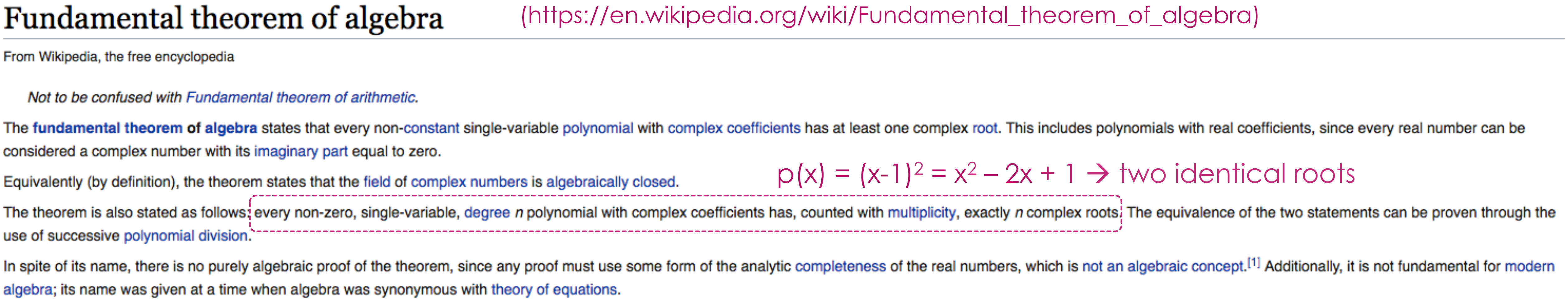
coefficients

(**Real** or **Complex** numbers)

constant term

$p(0) = a_0$

Polynomials



Fundamental theorem of algebra

(https://en.wikipedia.org/wiki/Fundamental_theorem_of_algebra)

From Wikipedia, the free encyclopedia

Not to be confused with Fundamental theorem of arithmetic.

The **fundamental theorem of algebra** states that every non-constant single-variable polynomial with complex coefficients has at least one complex root. This includes polynomials with real coefficients, since every real number can be considered a complex number with its imaginary part equal to zero.

Equivalently (by definition), the theorem states that the field of complex numbers is algebraically closed.

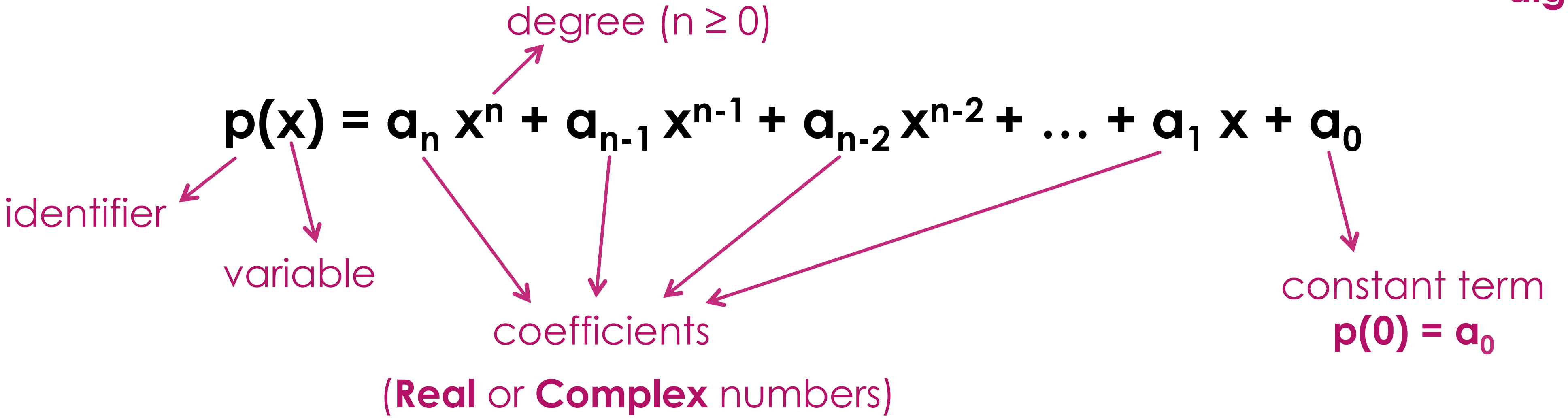
$p(x) = (x-1)^2 = x^2 – 2x + 1$ → two identical roots

The theorem is also stated as follows: every non-zero, single-variable, degree *n* polynomial with complex coefficients has, counted with multiplicity, exactly *n* complex roots. The equivalence of the two statements can be proven through the use of successive polynomial division.

In spite of its name, there is no purely algebraic proof of the theorem, since any proof must use some form of the analytic completeness of the real numbers, which is not an algebraic concept.[1] Additionally, it is not fundamental for modern algebra; its name was given at a time when algebra was synonymous with theory of equations.

A polynomial of **deg = n** has maximum n **real** roots (**exactly n complex roots**)

Fundamental theorem of algebra



degree $(n \geq 0)$

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_1 x + a_0$$

identifier

variable

coefficients

(**Real** or **Complex** numbers)

constant term

$\mathbf{p(0) = a_0}$

Polynomials

✓ What we need to know:

- Construct a polynomial (single variable) → use an array of real or complex numbers
- Degree → Highest power (integer number ≥ 0)
- **Value** & string representation → $p(x) = a_n x^n + \ldots + a_1 x + a_0$ where $\mathbf{a_n \neq 0}$
- Composition → $p(q(x))$
- Algebra: addition, subtraction → $p(x) \pm q(x) = s(x)$: polynomial
- Algebra: multiplication → $p(x) * q(x) = s(x)$: polynomial
- Division is more complicated → $p(x) / q(x) = s(x) + \mathbf{r(x)/q(x)}$ where $\mathbf{deg(r(x))<deg(q(x))}$
- Derivative → $p'(x) = s(x) = d(p(x))/dx$ where $\mathbf{deg(p'(x)) = deg(p(x))-1}$
- Integral → $s'(x) = p(x)$ where $\mathbf{deg(s(x)) = deg(p(x))+1}$
- **Roots (zeros)** → $p(x) = 0$
  - May be **real** or **complex**
  - Even if the coefficients are real, some roots may be complex numbers

Polynomials

✓ **Horner's method**

- **Most efficient way** of evaluating a polynomial
- Only **n summation** and **n multiplications** (n = deg(p(x)))

$$p(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n = \underbrace{a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + \overbrace{x(\overbrace{a_{n-1} + x a_n}^{\text{Sum}})}^{\text{Multiply}}\dots)))}_{}$$

This is just another way of writing the polynomial!
(**pretty clever way!!**)
n multiplications & *n* summation

- **Evaluating p(x_0)**
  - $b_n = a_n$
  - $b_{n-1} = a_{n-1} + b_n x_0$
  - $b_{n-2} = a_{n-2} + b_{n-1} x_0$
  - …
  - $\mathbf{b_0} = a_0 + b_1 x_0$

Evaluate this sequence
p(x_0) = b_0

Tip: Horner's method can be used to find composition p(q(x)).

Polynomials

✓ **Algebra**

- Addition, subtraction: $\mathbf{p(x) \pm q(x) = s(x)}$
  - Result is always a polynomial
  - Add coefficients term by term: $s_n = p_n \pm q_n$
  - $\mathbf{deg(s(x))} = \max(\deg(p(x)), \deg(q(x)))$ → general case
- Multiplication: $p(x) * q(x) = s(x)$ → $\deg(s) = \deg(p) + \deg(q)$
  - Result is always a polynomial
  - **Convolution** of coefficients: s_n = $\mathbf{p_n}$ **convolves** $\mathbf{q_n}$ → $\overbrace{\mathbf{s_n = p_n\, q_0 + p_{n-1}\, q_1 + \ldots + p_0\, q_n}}^{\text{Convolution sum}}$
- Division
  - Result is **not** always a polynomial
  - Result is a **rational** expression
  - Fundamental theorem of division: $p(x)/q(x) = s(x) + \overset{\text{remainder}}{\mathbf{r(x)}}/q(x)$ where $\deg(r(x))<\deg(q(x))$

Tip: implement algebra as part of the “operator overloading”.

Polynomials

- **Other useful libraries**
  - **Michael Flanagan's** Java Scientific Library
  - Archive (**jar**) available on this personal website:
    - ✓ https://www.ee.ucl.ac.uk/~mflanaga/java/
  - Source code available on github
    - https://github.com/bgithub1/flanagan
  - You can write **adapter methods** from your polynomial class to Flanagan's polynomial class
    - ✓ Adapters are very useful when using methods of different java libraries
      - They allow interfacing with other java libraries
    - ✓ We will see this in more details in "linear algebra" topic.

Adapter method: public **flanagan.Polynomial** toFlanaganPolynomial()

class

method name

Polynomial Division

- Result of division is **not** a polynomial in general
  - Instead it should return two polynomials
  - P(x) = B(x) **Q(x)** + **R(x)** → Q(x): quotient, R(x): remainder, P(x): dividend, B(x): divisor
  - Remember: this is an "**identity**" that holds for any real or complex number
- Fundamental Lemma: **deg(R(x)) < deg(B(x))**
  - Equality is not allowed
- Special cases
  - P(x) = (x-a) Q(x) + R → remainder is a number → **R = P(a)**
  - P(x) = (x-a)(x-b) Q(x) + (**m**x+**n**) → remainder is first order
    - **m** = (P(a)-P(b))/(a-b), **n** = (a P(b)-b P(a))/(a-b)
- What to do in the general case?
  - Must find both Q(x) and R(x)

Polynomial Division

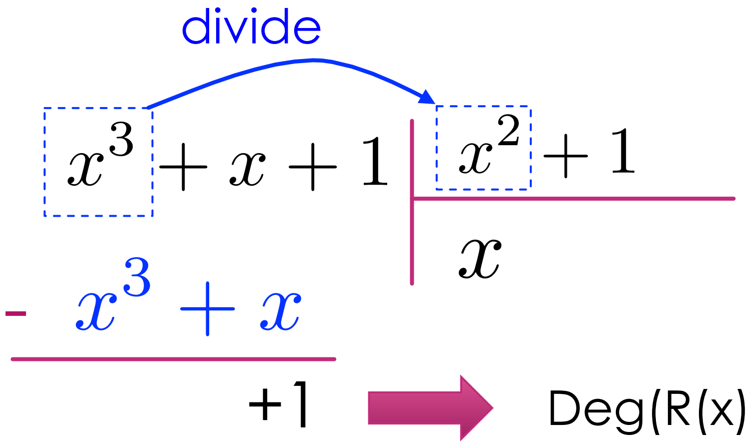
- Polynomial **Long Division**
  - Sort coefficients from highest degree to lowest
  - Make sure the coefficient of the highest degree is NOT zero
  - If **deg(P(x))<deg(B(x))** → Nothing to do: Q(x)=0, R(x)=P(x)

reduce() method

P(x) = B(x) Q(x) + R(x)

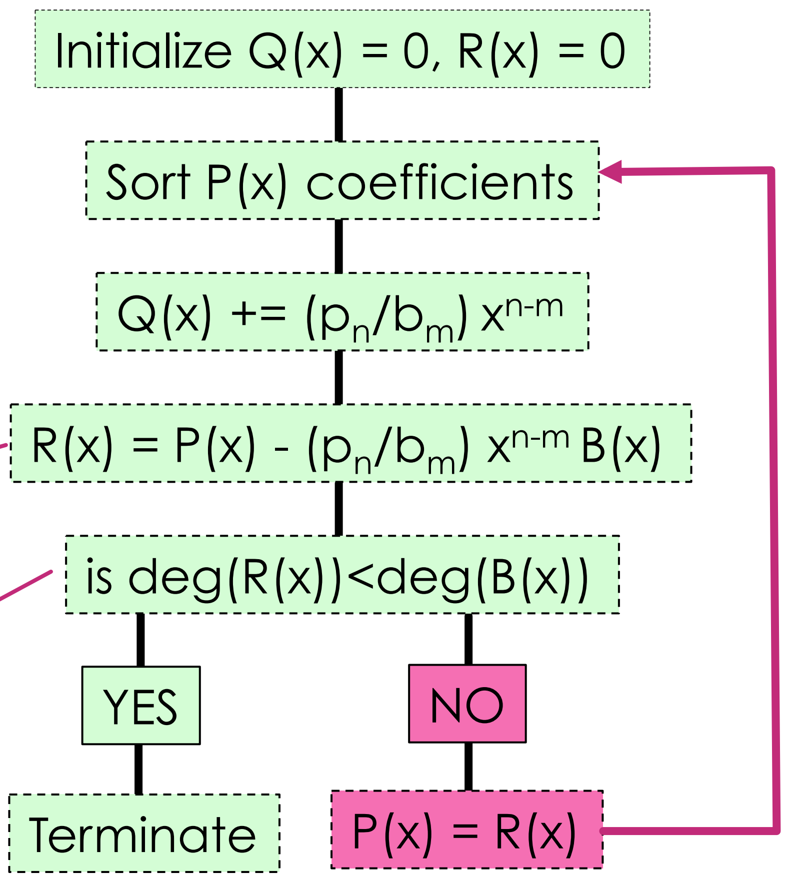
$$P(x) = p_n x^n + \ldots + p_1 x + p_0$$

$$B(x) = b_m x^m + \ldots + b_1 x + b_0$$



Deg(R(x))<deg(B(x)) : **STOP**

Flowchart



Taking use of multiplication and subtraction

Strictly less than (equality is not allowed)

Taylor Series

✓ **A polynomial series representing a smooth function f(x)**

- Polynomial expansion at a given point
    - Based on derivatives of the function

coefficient = 1/n!

n = 0 **n = 1** **n = 2**

$$f(x) = f(x_0) + \boxed{\frac{1}{1!}}\,\frac{df}{dx}\Big|_{x_0}(x - x_0) + \boxed{\frac{1}{2!}}\,\frac{d^2 f}{dx^2}\Big|_{x_0}(x - x_0)^2 + \ldots \Rightarrow$$

Polynomial

Point of interest

- Locally approximates the function
- **Example**: Taylor expansion of exp(x) at x = 0

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \ldots$$

- Different ways of implementing
    - Use the symbolic math library we developed before (call **diff()** multiple times)
    - Numerically evaluate higher order derivatives at each point
        - Richardson extrapolation

Taylor Series

- Taylor series has a **uniform** convergence
- You can apply term-by-term algebra
  - Add, subtract
  - Multiply, divide by a number
  - Differentiate
  - Integrate
- Necessary condition for the convergence
  - lim $|a_n(x-x_0)^n| \to 0$ as $n \to \infty$
- **Radius of convergence**

$$f(x) = \sum_{n=0}^{\infty} f_n (x-x_0)^n \qquad g(x) = \sum_{n=0}^{\infty} g_n (x-x_0)^n$$

$$\boxed{\begin{aligned} f(x) \pm g(x) &= \sum_{n=0}^{\infty} (f_n \pm g_n)(x-x_0)^n \\ c\, f(x) &= \sum_{n=0}^{\infty} (c\, f_n)(x-x_0)^n \\ f'(x) &= \sum_{n=1}^{\infty} (n\, f_n)(x-x_0)^{n-1} \end{aligned}}$$

$$R = \frac{1}{\lim_{n\to\infty} \sqrt[n]{|a_n|}} \Rightarrow \left\{ \begin{array}{l} \text{If } a_n\text{'s have the same sign as } n\to\infty \\ \qquad R = \lim\limits_{n\to\infty} \dfrac{a_n}{a_{n+1}} \\ \text{If } a_n\text{'s have alternating sign as } n\to\infty \\ \qquad b_n = \sqrt{\left|a_n a_{n-2} - a_{n-1}^2\right|} \Rightarrow R = \lim\limits_{n\to\infty} \dfrac{b_n}{b_{n+1}} \end{array} \right.$$

Cauchy–Hadamard theorem

R can be infinite

↓

f(x) : entire