Chapter 1

Adding and Subtracting Polynomials

Watch Khan Academy's **Adding polynomias** video at https://youtu.be/ahdKdxsTj8E

When adding two monomials of the same degree, you sum their coefficients:

$$7x^3 + 4x^3 = 11x^3$$

Using this idea, when adding two polynomials, you convert it into one long polynomial and then simplify by combining terms with the same degree. For example:

$$\begin{aligned}
&(10x^3 - 2x + 13) + (-5x^2 + 7x - 12) \\
&\quad = 10x^3 + (-2)x + 13 + (-5)x^2 + 7x + (-12) \\
&\quad = 10x^3 + (-5)x^2 + (-2 + 7)x + (13 - 12) \\
&\quad = 10x^3 - 5x^2 + 5x + 1
\end{aligned}$$

Exercise 1 Adding Polynomials Practice

Add the following polynomials:

1. $2x^3 - 5x^2 + 3x - 9$ and $x^3 - 2x^2 - 2x - 9$

2. $3x^5 - 5x^3 + 3x^2 - x - 3$ and $2x^4 - 2x^3 - 2x^2 + x - 9$

Working Space

Answer on Page 7

Notice that in the second question, the degree 1 term disappears completely: $(-x) + x = 0$

One more tricky thing that can happen: Sometimes the coefficients don't add nicely. For example:

$$\pi x^2 - 3x^2 = (\pi - 3)x^2$$

That is as far as you can simplify it.

1.1 Subtraction

Now watch Khan Academy's **Subtracting polynomials** at https://youtu.be/5ZdxnFspyP8.

When subtracting one polynomial from the other, it is a lot like adding two polynomials. The difference: when make the two polynomials into one long polynomial, we multiply each monomial that is being subtracted by -1. For example:

$$
\begin{aligned}
&(2x^2 - 3x + 9) - (5x^2 - 7x + 4) \\
&= 2x^2 + (-3)x + 9 + (-5)x^2 + 7x + (-4) \\
&= (2 - 5)x^2 + (-3 + 7)x + (9 - 4) \\
&= -3x^2 + 4x + 5
\end{aligned}
$$

Exercise 2 Subtracting Polynomials Practice

Add the following polynomials:

1. $(2x^3 - 5x^2 + 3x - 9) - (x^3 - 2x^2 - 2x - 9)$

2. $(3x^5 - 5x^3 + 3x^2 - x - 3) - (2x^4 - 2x^3 - 2x^2 + x - 9)$

Working Space

Answer on Page 7

1.2 Adding Polynomials in Python

As a reminder, in our Python code, we are representing a polynomial with a list of coefficients. The first coefficient is the constant term. The last coefficient is the leading coefficient. So, we can imagine $-5x^3 + 3x^2 - 4x + 9$ and $2x^3 + 4x^2 - 9$ would look like this: *FIXME: Diagram here*

To add the two polynomials then, we sum the coefficients for each degree. *FIXME: Diagram here*

Create a file called `add_polynomials.py`, and type in the following:

```
def add_polynomials(a, b):
    degree_of_result = len(a)
    result = []
    for i in range(degree_of_result):
        coefficient_a = a[i]
        coefficient_b = b[i]
        result.append(coefficient_a + coefficient_b)
    return result
```

```
polynomial1 = [9.0, -4.0, 3.0, -5.0]
polynomial2 = [-9.0, 0.0, 4.0, 2.0]
polynomial3 = add_polynomials(polynomial1, polynomial2)

print('Sum =', polynomial3)
```

Run the program.

Unfortunately, this code only works if the polynomails are the same length. For example, try making `polynomial1` have a larger degree than `polynomial2`:

```
# x**4 - 5x**3 + 3x**2 - 4x + 9
polynomial1 = [9.0, -4.0, 3.0, -5.0, 1.0]

# 2x**3 + 4x**2 - 9
polynomial2 = [-9.0, 0.0, 4.0, 2.0]
polynomial3 = add_polynomials(polynomial1, polynomial2)
print('Sum =', polynomial3)
```

See the problem?

Exercise 3 Dealing with polynomials of different degrees

Working Space

Can you fix the function `add_polynomials` to handle polynomials of different degrees?

Here is a hint: In Python, there is a `max` function that returns the largest of the numbers it is passed.

```
biggest = max(5,7)
```

Here `biggest` would be set to 7.

Here is another hint: If you have an array `mylist`, `i`, a non-negative integer, is only a legit index if `i < len(mylist)`.

Answer on Page 7

1.3 Scalar multiplication of polynomials

If you multiply a polynomial with a number, the distributive property applies:

$$(3.1)(2x^2 + 3x + 1) = (6.2)x^2 + (9.3)x + 3.1$$

(When we are talking about things that are more complicated than a number, we use the word *scalar* to mean "Just a number". So this is the product of a scalar and a polynomial.)

In `add_polynomials.py`, add a function to that multiplies a scalar and a polynomial:

```
def scalar_polynomial_multiply(s, pn):
    result = []
    for coefficient in pn:
        result.append(s * coefficient)
    return result
```

Somewhere near the end of the program, test this function:

```
polynomial4 = scalar_polynomial_multiply(5.0, polynomial1)
print('Scalar product =', polynomial_to_string(polynomial4))
```

Exercise 4 Subtract polynomials in Python

Working Space

Now implement a function that does subtraction using `scalar_polynomial_multiply` and `add_polynomials`.

It should look like this:

```
def subtract_polynomial(a, b):
    ...Your code here...

polynomial5 = [9.0, -4.0, 3.0, -5.0]
polynomial6 = [-9.0, 0.0, 4.0, 2.0, 1.0]
polynomial7 = subtract_polynomial(polynomial5, polynomial6)
print('Difference =', polynomial_to_string(polynomial7))
```

Answer on Page 7

This is a draft chapter from the Kontinua Project. Please see our website (https://kontinua.org/) for more details.

Appendix A

Answers to Exercises

Answer to Exercise 1 (on page 2)

$3x^3 - 7x^2 + x - 18$ and $3x^5 - 7x^3 + x^2 - 12$

Answer to Exercise 2 (on page 3)

$x^3 - 3x^2 + 5x$ and $x^5 - 3x^3 + 5x^2 - 2x + 6$

Answer to Exercise 3 (on page 4)

```
def add_polynomials(a, b):
    degree_of_result = max(len(a), len(b))
    result = []
    for i in range(degree_of_result):
        if i < len(a):
            coefficient_a = a[i]
        else:
            coefficient_a = 0.0

        if i < len(b):
            coefficient_b = b[i]
        else:
            coefficient_b = 0.0

        result.append(coefficient_a + coefficient_b)
    return result
```

Answer to Exercise 4 (on page 5)

```
def subtract_polynomial(a, b):
    neg_b = scalar_polynomial_multiply(-1.0, b)
```

```
return add_polynomials(a, neg_b)
```

Index