# Section 6 Problem

In this problem, you are provided the completed `polynomial.m` class in the lecture. Your task is to create a class `piecewise_polynomial.m` that will generalize `polynomial.m` to piecewise polynomial. A script `test_poly.m` has been provided to demonstrate use of the polynomial class.

A piecewise polynomial is a sequence of polynomials defined on neighboring intervals. To make this more precise, consider the domain $\Omega = [x_1, x_{N+1}] \subset \mathbf{R}$ (subset of the real number line) and define $x_i \in \Omega$ for $i = 2, \ldots, N$. Then, $\mathcal{I}_i = [x_i, x_{i+1}]$ for $i = 1, \ldots, N$ defines a partition of the domain $\Omega$. In the remainder of this, the endpoints of these intervals ($x_i$) will be called *knots*. A piecewise polynomial $p(x)$ over $\Omega$ is defined as

$$p(x) = \begin{cases} p_1(x) & \text{for } x \in [x_1, x_2] \\ p_2(x) & \text{for } x \in [x_2, x_3] \\ \vdots & \vdots \\ p_N(x) & \text{for } x \in [x_N, x_{N+1}] \end{cases}$$

where $p_i(x)$ is a polynomial.

### Task 1

Setup the class for `piecewise_polynomial.m`

1. `piecewise_polynomial` should have three properties

- `xi` - `double` array containing *knots*
- `npoly` - `double` scalar containing number of polynomials ( `= length(xi)-1`)
- `polys` - `polynomial` array of length `npoly`

2. the constructor of `piecewise_polynomial` should

- accept zero, one or two arguments
- if zero arguments passed, create the *`empty`* piecewise polynomial (all properties set to `[]` )
- if one argument passed of type
- * `piecewise_polynomial` - copy the properties of the input to the new object
- * `double` - assume the argument is `xi` and set the properties accordingly (`polys` empty)
- if two arguments passed
- * assume the first is a `double` containing the knots `xi`
- * assume the second is a `polynomial` array containing the polynomials of the piecewise polynomial

### Task 2

Implement the following methods, where $p(x)$ and $q(x)$ are piecewise polynomials

- `uplus`: $p(x) \mapsto +p(x)$

- `plus`: $p(x), q(x) \mapsto p(x) + q(x)$
- `uminus`: $p(x) \mapsto -p(x)$
- `minus`: $p(x), q(x) \mapsto p(x) - q(x)$
- `mtimes`: $p(x), q(x) \mapsto p(x)q(x)$
- `mpower`: $p(x), c \mapsto p(x)^c$ where $c$ is a non-negative integer
- `eq`: $p(x), q(x) \mapsto \{0, 1\}$, return `true` if $p$ and $q$ are equal for all $x \in \mathbf{R}$ (coefficients identical)
- `iszero`: $p(x) \mapsto \{0, 1\}$, return `true` if $p(x) = 0$ (all coefficients are zero)
- `integrate`: $p(x) \mapsto \int p(x)dx$
- `differentiate`: $p(x) \mapsto \frac{dp}{dx}(x)$
- `evaluate`: $p(x), \mathbf{v} \mapsto p(\mathbf{v})$, should accept vector inputs evaluate $p(x)$ at each entry and return the output as a vector
- `plot_it`, plot $p(x)$ over some specified domain

*Hint: You should not have to implement any polynomial routines yourself. Your class should store an array of* `polynomial` *objects which have all polynomial functionality required for this assignment implemented. Use the methods in the* `polynomial` *objects to perform polynomial operations.* `piecewise_polynomial.m` *should really just be a* wrapper *for* `polynomial.m` *to implement the piecewise functionality.*

### Task 3

Use `piecewise_polynomial.m` to create the following piecewise polynomials over the domain $\Omega = [-2, 2]$. The code for this part of the problem should be in your driver (display output by not including semicolons `;`).

$$p_1(x) = \begin{cases} x^3 - 2x^2 & \text{for } x \in [-2, -1] \\ 0 & \text{for } x \in [-1, 2] \end{cases} \qquad p_2(x) = \begin{cases} 0 & \text{for } x \in [-2, -1] \\ x & \text{for } x \in [-1, 1] \\ 0 & \text{for } x \in [1, 2] \end{cases}$$

$$p_3(x) = \begin{cases} 0 & \text{for } x \in [-2, 1] \\ x^2 - 3 & \text{for } x \in [1, 2] \end{cases}$$

- Compute and plot $p_4(x) = p_1(x) + p_2(x) + p_3(x)$
- Compute and plot $p_5(x) = p_1(x) - p_2(x) - p_3(x)$
- Compute and plot $p_6(x) = p_4(x)p_5(x)$
- Compute and plot $p_7(x) = p_2(x)^3$
- Compute and plot $p_8(x) = \int p_4(x)dx$
- Compute and plot $p_9(x) = \frac{dp_8}{dx}(x)$

### Checkpoint

Please answer the following questions and put the answers in the EdX page:

(A) Does $p_4(x)$ have only one local maximum?

(B) Is $p_5(x)$ a decreasing function?

(C) What is the value of $p_6(1.5)$?

(D) What is the maximum value of $p_7(x)$?

(E) If the integrating constant is zero, will $p_8(x)$ be always positive?

(F) Are $p_4(x)$ and $p_9(x)$ equal?