Programación: Multiplicación de un polinomio por un binomio

Objetivos. Escribir una función que multiplique un polinomio por un binomio. Vamos a usar esta función en otras partes del curso.

Requisitos. Programación de ciclos `for`, programación de funciones nuevas. en GNU Octave, definición de una función en GNU Octave.

1. Lenguajes de programación. Estos ejercicios corresponden principalmente a los lenguajes de programación Matlab y GNU Octave, pero se pueden adaptar fácilmente a Julia, Wolfram Mathematica y cualquier otro lenguaje de programación con ciclos `for` en el cual los índices de elementos de arreglos empiezan con 1.

2. Repaso de algunos elementos de la sintaxis. Suponemos que

los índices de elementos empiezan en 1.

Para acostumbrarse a la sintaxis ejecute los siguientes comandos uno por uno en el intérprete. No es necesario teclear los # *comentarios.*

```
a = [10; 20; 30]      # crear un arreglo con elementos 10, 20, 30

length(a)      # longitud del arreglo

a(2)      # obtener el valor del segundo elemento

a(2) = 70      # modificar el valor del segundo elemento

display(a)      # ver el vector modificado

b = zeros(7, 1)      # crear un vector de longitud 7 con elementos nulos
```

3. Guardar polinomios como arreglos de sus coeficientes. Vamos a representar los polinomios como arreglos de sus coeficientes, empezando con el término independiente. Por ejemplo, el polinomio

$$f(x) = 7x^4 - 3x^2 + 5x + 4$$

se guardará como el arreglo de números $4, 5, -3, 0, 7$. En general, el polinomio

$$f(x) = a_1 + a_2x + a_3x^2 + \cdots + a_nx^{n-1}$$

guardamos como el arreglo con elementos $a_1, a_2, a_3, \ldots, a_n$. Aquí n es el número de coeficientes, o sea la *longitud* del arreglo de coeficientes. En vez de n se puede usar la variable $d = n - 1$, es decir, el *grado* del polinomio. Nótese que a_k es el coeficiente de x^{k-1}.

Fórmulas de multiplicación de un polinomio por un binomio (se recomienda deducirlas antes de la clase práctica)

4. Fórmulas para multilicar un polinomio de grado 3 por un binomio mónico. Sean $f(x)$ un polinomio de grado 3 y $g(x)$ un binomio mónico, es decir, un binomio cuyo coeficiente del grado mayor es 1:

$$f(x) = a_1 + a_2 x + a_3 x^2 + a_4 x^3, \qquad g(x) = b + x.$$

Notemos que el polinomio $f(x)$ es de grado 3 y por lo tanto tiene $\underbrace{\qquad}_{\text{¿cuántos?}}$ coeficientes.

El producto $f(x)g(x)$ es de grado $\underbrace{\qquad}_{?}$, esto es, tiene $\underbrace{\qquad}_{\text{¿cuántos?}}$ coeficientes.

Denotemos los coeficientes del producto $f(x)g(x)$ por $c_1, \ldots, \underbrace{\qquad}_{?}$:

$$c_1 + c_2 x + c_3 x^2 + c_4 x^3 + c_5 x^4 = (a_1 + a_2 x + a_3 x^2 + a_4 x^3)(b + x).$$

Exprese los coeficientes $c_1, \ldots, c_5$ a través de $a_1, \ldots, a_4$ y b:

$$c_1 = \underbrace{\qquad\qquad\qquad}_{?},$$

$$c_2 = \underbrace{\qquad\qquad\qquad}_{?},$$

$$c_3 = \underbrace{\qquad\qquad\qquad}_{?},$$

$$c_4 = \underbrace{\qquad\qquad\qquad}_{?},$$

$$c_5 = \underbrace{\qquad\qquad\qquad}_{?}.$$

Se puede ver que las fórmulas para c_2, c_3, c_4 tienen la misma estructura:

$$c_k = \underbrace{\qquad\qquad\qquad}_{?}, \qquad \text{para} \qquad \underbrace{\qquad}_{?} \le k \le \underbrace{\qquad}_{?}.$$

Las fórmulas para los "coeficientes extremos" c_1 y c_5 son un poco diferentes.

5. Fórmulas para multiplicar un polinomio por un binomio mónico.
Sean

$$f(x) = a_1 + a_2 x + \ldots + a_n x^{n-1}, \qquad g(x) = b + x.$$

Entonces el producto $f(x)g(x)$ es de grado $\underbrace{\qquad}_{?}$, esto es, tiene $\underbrace{\qquad\quad}_{\text{¿cuantos?}}$ coeficientes.

Denotemos por $c_1, \ldots, c_{n+1}$ los coeficientes del producto $f(x)g(x)$:

$$c_1 + c_2 x + \cdots + c_{n+1} x^n = (a_1 + a_2 x + \ldots + a_n x^{n-1})(b + x).$$

Exprese los coeficientes $c_1, \ldots, c_{n+1}$ a través de los coeficientes $a_1, \ldots, a_n$ y b.

$$c_1 = \underbrace{\qquad\qquad}_{?};$$

$$c_k = \underbrace{\qquad\qquad\qquad\qquad}_{?} \qquad \text{para} \qquad \underbrace{\qquad}_{?} \le k \le \underbrace{\qquad}_{?};$$

$$c_{n+1} = \underbrace{\qquad\qquad}_{?}.$$

6. Algoritmo mulpolbinom (pseudocódigo).

función mulpolbinom(a, b):

variables locales: n, c, k;

$n :=$ longitud(a);

$c :=$ arreglo nulo de longitud $\underbrace{\qquad\quad}_{?}$;

$c_1 := \underbrace{\qquad\quad}_{?}$;

para k de $\underbrace{\qquad\quad}_{?}$ a $\underbrace{\qquad\quad}_{?}$:

$c_k := \underbrace{\qquad\qquad\qquad\qquad}_{?}$;

$c_{n+1} := \underbrace{\qquad\quad}_{?}$;

Salida: $\underbrace{\qquad}_{?}$.

Programar la multiplicación de un polinomio por un binomio en algún lenguaje de programación

7. Problema mulpolbinom.
Traduzca el algoritmo mulpolbinom a un lenguaje de programación. En otras palabras, escriba una función que calcule los coeficientes del producto de un polinomio $f(x)$ por un binomio $b + x$.

- Entrada (argumentos de la función): a, b,
  donde a es el arreglo de los coeficientes de $f(x)$.
- Salida (que debe regresar la función): el arreglo de los coeficientes del producto.

Por ejemplo, en el lenguaje GNU Octave la función se puede definir de la siguiente manera (hay que guardarla en el archivo `mulpolbinom.m` y sustituir `...` por expresiones apropiadas):

```
function [c] = mulpolbinom(a, b),
   n = length(a);
   c = zeros(..., 1);
   c(1) = ...;
   for k = ... : ...,
      ...
   endfor
   ...
endfunction
```

En el lenguaje Matlab se escribe `end` en vez de `endfor` y `endfunction`.

8. Primera comprobación. Verifique a mano (en papel) que

$$(7 - 2x + 4x^2 - 5x^3)(3 + x) = 21 + x + 10x^2 - 11x^3 - 5x^4.$$

Ejecute en el intérprete el comando

```
mulpolbinom([7; -2; 4; -5], 3)
```

Se debe regresar el vector `[21; 1; 10; -11; -5]`.

9. Segunda comprobación. Multiplique a mano el polinomio $5x^2-7x+3$ por el binomio $x + 2$. Luego ejecute `mulpolbinom([3; -7; 5], 2)`.