Càlcul de derivades de polinomis i avaluació mitjançant l'algorisme de Horner

March 4, 2025

Introducció

En aquesta pràctica es treballarà amb memòria dinàmica per gestionar una *matriu triangular* que emmagatzema els coeficients de les derivades successives d'un polinomi. A més, es calcularà el valor de cada derivada en un punt determinat utilitzant l'algorisme de Horner, una tècnica eficient per avaluar polinomis.

Descripció del problema

Donat un polinomi real $p(x)$ de grau n, s'han de dur a terme les operacions següents:

- Llegir el grau n i els $n+1$ coeficients $p_0, p_1, \ldots, p_n$.
- Calcular els coeficients de totes les derivades $p^{(i)}(x)$ per a $i = 0, 1, \ldots, n$.
- Emmagatzemar aquests coeficients en una matriu triangular implementada amb memòria dinàmica.
- Llegir un valor real x_0 i calcular $p^{(i)}(x_0)$ per a cada i utilitzant l'algorisme de Horner.

Algorisme de Horner

L'algorisme de Horner permet avaluar un polinomi en un punt de manera eficient, reduint el nombre d'operacions necessàries. Donat un polinomi de grau n:

$$p(x) = q_0 + q_1 x + q_2 x^2 + \cdots + q_n x^n,$$

es pot reescriure com:

$$p(x) = q_0 + x(q_1 + x(q_2 + x(q_3 + \cdots + x(q_n)))).$$

Aquest enfocament permet calcular el valor del polinomi mitjançant un recorregut iteratiu pels coeficients, millorant així l'eficiència computacional.

Objectius

Amb aquesta pràctica es pretén assolir els objectius següents:

- Comprendre i aplicar la memòria dinàmica per gestionar estructures de dades variables.
- Implementar la derivació simbòlica de polinomis de manera eficient.
- Aplicar l'algorisme de Horner per avaluar polinomis i analitzar la seva eficiència.

Enunciat

Implementa les funcions següents en C:

1. Escriu una funció `double horner(int n, double *pol, double x0)` que, donats un enter positiu n, un vector de coeficients *pol* i un valor real x_0, avaluï el polinomi en el punt x_0 utilitzant l'algorisme de Horner.
2. Escriu una funció `double **reserva_mem(int n)` que reservi memòria per a la matriu triangular, on s'han d'emmagatzemar els coeficients del polinomi i totes les seves derivades.
3. Escriu una funció `void liberar_mem(double **pol, int n)` que alliberi la memòria dinàmica utilitzada per la matriu triangular.
4. Implementa la funció `main` que:
    - Llegeixi el grau del polinomi n i els seus $n + 1$ coeficients.
    - Calculi i emmagatzemi els coeficients de les derivades en una matriu triangular.
    - Demani un valor x_0 i avaluï el polinomi i les seves derivades en aquest punt utilitzant l'algorisme de Horner.
    - Mostri per pantalla els coeficients de cada derivada i els valors calculats en x_0.

Exemple d'execució

```
n = ?
6
p0 p1 p2 ... pn ?
1 2 3 4 5 6 7

p^(0), grau 6, coeficients:
 +1.00  +2.00  +3.00  +4.00  +5.00  +6.00  +7.00

p^(1), grau 5, coeficients:
 +2.00  +6.00  +12.00  +20.00  +30.00  +42.00

p^(2), grau 4, coeficients:
 +6.00  +24.00  +60.00  +120.00  +210.00

p^(3), grau 3, coeficients:
 +24.00  +120.00  +360.00  +840.00

p^(4), grau 2, coeficients:
 +120.00  +720.00  +2520.00

p^(5), grau 1, coeficients:
 +720.00  +5040.00

p^(6), grau 0, coeficients:
 +5040.00
x0 = ?
0.25
p^(0)(x0) = +1.78
p^(1)(x0) = +4.72
p^(2)(x0) = +18.45
p^(3)(x0) = +89.62
p^(4)(x0) = +457.50
p^(5)(x0) = +1980.00
p^(6)(x0) = +5040.00
```