1. Evaluación de un Polinomio

Evaluar polinomios o funciones en general tiene muchos problemas, aún para el software profesional. Como se requiere poder evaluar el polinomio en las raíces encontradas, es necesario dedicarle un momento a los detalles. Sea $P(x) = a_0x^n + a_1x^{n-1} + ... + a_n$ un polinomio entonces, uno de los problemas que se enfrentan es evaluar el polinomio en valor dado x_0 de la manera más eficiente. Un método visto es Horner:

$$b_0 = a_0$$
$$b_k = a_k + b_{k-1}x_0(k = 1, ..., n-1)$$
$$b_n = P(x_0)$$

1. Implemente en R o Python el método de Horner para evaluar $f^{'}(x_0)$, tenga en cuenta la precisión de la computadora o unidad de redondeo.
2. Implemente el método de Horner si se realiza con números complejos, tenga en cuenta la precisión

2. Optima Aproximación Polinómica

La implementación de punto flotante de una función en un intervalo a menudo se reduce a una aproximación polinómica, siendo el polinomio típicamente proporcionado por el algoritmo Remez. Sin embargo, la evaluación de coma flotante de un polinomio de Remez a veces conduce a cancelaciones catastróficas.El algoritmo Remez es una metodología para localizar la aproximación racional minimax a una función. La cancelaciones que también hay que considerar en el caso de del método de Horner, suceden cuando algunos de los coeficientes polinómicos son de magnitud muy pequeña con respecto a otros. En este caso, es mejor forzar estos coeficientes a cero, lo que también reduce el recuento de operaciones. Esta técnica, usada clásicamente para funciones pares o impares, puede generalizarse a una clase de funciones mucho más grande.
Aplicar esta técnica para $f(x) = sin(x)$ en el intervalo $[-\pi/64, \pi/64]$ con una precisión deseada doble. Para cada caso, evalue la aproximación polinomica de la función, el error relativo y el número de operaciones necesarias

1. Aplique una aproximación de Taylor
2. Implemente el método de Remez