Taller 1 - Análisis Numérico

Jhonnier David Coronado Vanegas
Ingeniería de Sistemas
Pontificia Universidad Javeriana
Bogotá, Colombia
jhonnier.coronado@javeriana.edu.co

Juan Sebastian Valderrama Urquijo
Ingeniería de Sistemas
Pontificia Universidad Javeriana
Bogotá, Colombia
valderramajuan@javeriana.edu.co

I. Error de redondeo

Para la resolución de éste problema, se decidió aplicar simples multiplicaciones y divisiones. Esto, se realizó de esta forma tan “sencilla”, debido a que sólo se pide encontrar el error que existe en un máquina si se pide almacenar un número con mayor cantidad de cifras significativas que las posibles.

Tomando *n*, como el número de cifras significativas posibles que se pueden almacenar, y *x* el número que se se almacenará, se obtiene que:

$$valor\_truncado = \frac{(x * 10^{n-2})}{10^{n-2}}$$

El error de este resultado, se calcula restando al número inicial el valor truncado. Siendo el caso usado con; *x = 536.78*, *n = 4*. Así, tenemos.

$$valor\_truncado = \frac{(536.78 * 10^{4-2})}{10^{4-2}}$$

Como resultado final, se tiene que el *error* equivale a 0.08.

II. Algoritmo de la raíz cuadrada babilónico

En el caso de este problema, ya se nos da un algoritmo iterativo, basado en el algoritmo babilónico para calcular raíces cuadradas, siendo este:

Entrada:	n	Dato
	E	Error Permitido
	x	Valor inicial
Salida:	y	Respuesta calculada con error E

Algoritmo:

$$Mientras\ |x - y| > E:$$

$$x = y$$

$$y = \frac{1}{2} * (x + \frac{n}{x})$$

$$Fin$$

Teniendo como *n*, el dato al que se hallará la raíz siendo el caso del ejemplo, 7. *E*, como el error permitido con el que precisa calcular la raíz, es decir, el máximo error con el que se le permite al algoritmo operar. *x*, es un número que se aproxima al valor real de la raíz, mientras más lejano sea este número al resultado real de la misma, más lento e ineficiente será el algoritmo.

Tomando como ejemplo:

n = 7

E = 0.01

x = 2

Tenemos así como resultado, que la raíz aproximada de 7, con valor inicial 2, es igual a:

$$2.75$$

La precisión de este algoritmo, se puede saber de la misma forma que se calcula el *Error de Redondeo*, restando al valor que debería dar el resultado del algoritmo, así tenemos que:

$$\sqrt{7} \approx 2.64575131106$$

$$error\_resultado = |2.64575131106 - 2.75|$$

$$error\_resultado = 0.10424868893$$

Que, si se aproxima a dos cifras significativas se tiene que el resultado es:

$$error\_resultado = 0.10$$

Y como se puede observar, es igual al error máximo permitido.

Esta forma de calcular la raíz de un número *n* diverge en el caso de que el resultado de la raíz sea un número irracional, como es en el caso del 7, debido a que infinitamente se va acercando al resultado de la misma, siendo el único limitante, la capacidad de procesamiento de la máquina. En el caso contrario, de que la raíz sea un número racional, se sabe que converge dando como resultado el esperado.

III. TEOREMA DE TAYLOR

Para este problema, se solicita calcular la aproximación mediante el Teorema de Taylor, este teorema dice que

$Sea\ k \geq 1\ un\ entero\ y\ la\ función\ f: R \rightarrow R\ diferenciable\ k\ veces\ en\ el\ punto\ a\ \in R. Entonces\ existe\ una\ función\ h_k: R\ \rightarrow R\ tal\ que$:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(-a)^2 + \ldots + \frac{f^{(k)}(a)}{k!}(x-a)^k + h_k(x)(x-a)^k$$

El polinomio que aparece en el teorema de Taylor se denomina **polinomio de Taylor de orden *k*.**

$$P_k(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ldots + \frac{f^{(k)}(a)}{k!}(x-a)^k$$

de la función *f* en el punto *a*. El polinomio de Taylor es el único polinomio que "mejor aproxima en forma asintótica" en el sentido de que existe una función $h_k: R \rightarrow R$ y un polinomio *p* de orden *k* tal que

$$R_k(x) = f(x) - P_k(x)\ , \lim_{x \to a} h_k = 0\ ,$$

entonces $p = P_k$. El teorema de Taylor describe el comportamiento asintótico del término del resto

$$R_k(x) = f(x) - P_k(x),$$

el cual es el error de aproximación cuando se aproxima *f* con su polinomio de Taylor. Utilizando la notación o el teorema de Taylor se puede expresar de la siguiente forma:

$$R_k(x) = o(|x-a|^k), x \rightarrow a$$

Teniendo ya la explicación del **polinomio de Taylor**, así como la fórmula con la cual se aplica, se llega a lo siguiente:

Siendo **fx** la función a la cual se le desea aplicar el polinomio. A su vez *fx = f(x) =* e^x, y *n*, el grado hasta cuál queremos llegar el polinomio, teniendo en cuenta de que mientras más grande sea el grado, mayor cercano el resultado estará al valor real.

Sabemos que $e^0 = 1$,y que la derivada de la función exponencial es ella misma, podemos basarnos en el resultado conocido mencionado anteriormente para llegar a:

$e^x \approx 1 + \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \ldots + \frac{1}{n!}x^n$, entonces

$$e^x \approx 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!}$$

Como algoritmo tenemos que

$$resultado = 0$$

$$Para\ i = 0\ hasta\ n\ Hacer:$$

$$resultado = resultado + (x^i\ /\ i!)$$

$$retornar\ resultado$$

Con *x = 0.5* y *n = 1*, se obtiene que el resultado es igual a *1.500* y con *x = 0.5* y *n = 10*, se obtiene que el resultado es igual a *1.6487*, el cual es el resultado correcto si se redondea a *5* cifras significativas.

IV. CALCULAR EL TAMAÑO DEL ERROR

Para hacer cálculos con datos que tengan una medida del error, se procede a calcular el mayor posible resultado, esto es, a los datos que se poseen, se les suma la incertidumbre y se operan. Lo mismo se realiza de forma inversa, restando la incertidumbre, esto para encontrar el menor posible resultado.

Luego de obtenidos ambos valores, se halla el promedio de estos dos resultados. Siendo el resultante de esta operación, el valor más cercano a la realidad.

Siendo así, se tiene que:

$$v = 4 \qquad E_V = 0.1$$

$$t = 5 \qquad E_t = 0.1$$

$$d = vt$$

$$valor\_maximo = (v + E_v) * (t + E_t)$$

$$valor\_minimo = (v - E_v) * (t - E_t)$$

$$resultado = \frac{valor\_maximo + valor\_minimo}{2}$$

Para calcular el *error*, se suele al *valor_maximo* o al *valor_minimo* restarle el resultado.

Para calcular el *error absoluto* se suele al *resultado*, restarle la multiplicación de los valores sin tener en cuenta la incertidumbre.

Y por último, para calcular el error relativo, se suele al *error absoluto* dividirlo con el resultado.

Teniendo así que:

$$valor\_maximo = 20.91$$

$$valor\_minimo = 19.11$$

$$resultado = 20.01$$

$$error = 0.9$$

$$error\_relativo = 0.0004997501$$

$$error\_absoluto = 0.01$$

V. EVALUAR UN POLINOMIO

En este problema, se solicita encontrar un método o algoritmo mediante el cual se pueda resolver un polinomio de la forma más eficiente para la máquina, es decir, con la menor cantidad de multiplicaciones.

Para la resolución de este, se decidió usar el método de Horner, el cual sirve para calcular con *n* multiplicaciones un valor aproximado de un polinomio *P(x)* dado un x_0. Sea el método iterativo de Horner el siguiente:

Para un polinomio $P(x) = a_0 + a_1x + \ldots + a_nx^n$,la evaluación en $x_0$se dará como:

$$b_n = a_n$$

$$b_{n-1} = a_{n-1} + b_n * x_0$$

$$\ldots$$

$$b_0 = a_0 + b_1 * x_0$$

b_0 tendrá entonces el valor evaluado de $P(x_0)$. Sea $P_{Hi}(x_0)$ el valor del polinomio y $P_i(x)$; $x = x_0$usando el método de Horner.

a) Comparación del valor real de la evaluación de tres polinomios con el valor arrojado por el método de Horner y las multiplicaciones realizadas.

   i) $P_1(x) = 2x^4 - 3x^2 + 3x - 4;\ x_0 = -2$

   El valor real de $P_1(x_0) = 10$ se compara con $P_{H1}(x_0) = 10.0$ calculando usando únicamente 5 multiplicaciones.

   ii) $P_2(x) = 7x^5 + 6x^4 - 6x^3 + 3x - 4;\ x_0 = 3$

   El valor real de $P_1(x_0) = 2030$ se compara con $P_{H2}(x_0) = 2030.0$ calculando usando únicamente 6 multiplicaciones.

   iii) $P_3(x) = -5x^6 + 3x^4 + 2x^2 - 4x;\ x_0 = -1$

   El valor real de $P_1(x_0) = 4$ se compara con $P_{H3}(x_0) = 4.0$ calculando usando únicamente 7 multiplicaciones.

b) Demostración que el número mínimo de multiplicaciones es *n* siendo *n* el grado del polinomio

Sea $P_0(x) = a_0x^0 = a_0$. El número de multiplicaciones para hallar $P_0(x_0)$ es igual a 0. Por lo que se cumple para el primer caso *k* = 0.

Se asume por lo tanto que $P_k(x) = a_0 + a_1x + \ldots + a_kx^k$ y que $P_k(x_0) = a_0 + a_1x_0 + \ldots + a_kx_0^k$ tiene *k* multiplicaciones y que el método de Horner se expresa de la siguiente manera:

$$b_k = a_k$$

$$b_{k-1} = a_{k-1} + b_k * x_0$$

$$\ldots$$

$$b_0 = a_0 + b_1 * x_0$$

Y se debe llegar a la forma *k + 1* del polinomio:

$$P_{k+1}(x_0) = a_0 + a_1x_0 + \ldots + a_{k+1}x_0^{k+1}$$

o reescrito de otra forma:

$$P_{k+1}(x_0) = a_0 + x(a_1 + x(a_2 + \ldots + x(a_k + xa_{k+1})))$$

Se reescribe $P_k(x_0)$ y se reemplaza $a_k$por b_k (Primera instrucción del método):

$$P_k(x_0) = a_0 + x_0(a_1 + \ldots + x_0(a_{k-1} + x_0 * b_k))$$

Se añade una iteración al método de Horner para b_{k+1}, añadiendo una multiplicación más al método ($mult_{k+1} = k + 1$)

$$b_{k+1} = a_{k+1}$$

$$b_k = a_k + b_{k+1} * x_0$$

$$\ldots$$

$$b_0 = a_0 + b_1 * x_0$$

Y se reemplaza $b_k$en $P_k(x_0)$:

$$P_k(x_0) = a_0 + x_0(a_1 + \ldots + x_0(a_{k-1} + x_0 * (a_k + b_{k+1} * x_0))$$

Despejando la ecuación se llega a la forma:

$$P_k(x_0) = a_0 + x_0(a_1 + \ldots + x_0(a_{k-1} + x_0 * (a_k + b_{k+1} * x_0))$$

La cual es equivalente a $P_{k+1}(x_0)$, quedando demostrado el número de multiplicaciones iguales a *k*, el grado del polinomio.