

CÓMO MULTIPLICAN Y DIVIDEN LAS CALCULADORAS?

BRUCE H. EDWARDS

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF FLORIDA

RAFAEL A. DÍAZ

PASADENA INDEPENDENT SCHOOL DISTRICT, SAM RAYBURN

HIGH SCHOOL, DEPARTMENT OF MATHEMATICS

USA

1 Introducción

Iniciamos este ensayo formulando el siguiente par de preguntas: ¿Cual es el método usado por la calculadora para evaluar el producto de dos números? ¿En qué forma efectua la división? Será sorprendente aprender en este artículo que el algoritmo para efectuar multiplicaciones y divisiones es en escencia el mismo que se usa para evaluar prácticamente todas las funciones con que la calculadora trabaja, a saber: El algoritmo CORDIC. Dicho algoritmo

no usa técnicas clásicas de cálculo, ni tampoco métodos de aproximación mediante funciones polinómicas o racionales. El algoritmo está basado en un sistema elemental de ecuaciones iterativas.

El algoritmo CORDIC (**CO**ordinate **R**otation **DI**gital **C**omputer) es relativamente nuevo. Originalmente el algoritmo fué presentado en 1959 por Volder [3] para calcular valores de funciones trigonométricas. Más tarde el algoritmo fué ampliado en 1971 por Walther [4] para evaluar valores de funciones hiperbólicas, logarítmicas y otras funciones elementales. Como se verá mas adelante en este artículo, el algoritmo CORDIC aún se usa actualmente para efectuar el cálculo en las operaciones básicas de la multiplicación y división.

Las restricciones de hardware en las calculadoras exigen que la evaluación de las funciones elementales sea hecha usando unicamente sumas, restas, desplazamiento de dígitos, comparaciones y el uso de ciertas constantes que están almacenadas en la máquina. Como usted observará al leer este artículo, el método CORDIC usa solo las cinco operaciones básicas que se acaban de mencionar.

Debido a que convertir números en base 10 a base 2 toma bastante tiempo, el algoritmo CORDIC está programado directamente en base 10. No obstante es mas fácil entender las ideas esenciales al usar el sistema binario. Una vez que se sabe que el sistema binario será el usado resulta claro que multiplicar por 2^k , donde k es un entero, no es otra cosa que un desplazamiento de dígitos.

En la próxima sección se discutirá el algoritmo CORDIC, posteriormente se observará de manera informal como trabaja el algoritmo para la división y multiplicación. La quinta sección presenta los teoremas de convergencia que hacen válido el uso del algoritmo. Finalmente se discutirá como usar el algoritmo CORDIC para evaluar

otro tipo de funciones.

2 El Algoritmo CORDIC

El algoritmo CORDIC tiene su base en las tres ecuaciones iterativas que se dan a continuación.

$$\begin{aligned}x_{k+1} &= x_k - m\delta_k y_k 2^{-k} \\y_{k+1} &= y_k + \delta_k x_k 2^{-k} \\z_{k+1} &= z_k - \delta_k \sigma_k\end{aligned}$$

Las constantes m , δ_k , y σ_k dependen del cálculo específico que se este realizando como se explica a continuación.

1. m es 0, 1 o -1 . $m = 1$ se usa para evaluar funciones trigonométricas e inversas de funciones trigonométricas. $m = -1$ es usado para evaluar funciones hiperbólicas, inversas de funciones hiperbólicas, funciones exponenciales y logarítmicas, y raíces cuadradas. Finalmente, $m = 0$ se usa para efectuar multiplicaciones y divisiones, que es a la vez el objetivo de este artículo.
2. δ_k es alguna de las dos funciones de signo que siguen.

$$\delta_k = \text{sgn}(z_k) = \begin{cases} 1 & , z_k \geq 0 \\ -1 & , z_k < 0 \end{cases}$$

o

$$\delta_k = -\text{sgn}(y_k) = \begin{cases} 1 & , y_k < 0 \\ -1 & , y_k \geq 0 \end{cases}$$

La primera es conocida con el nombre técnico de “rotation mode”, para valores de z que convergen a cero, y se usa en el algoritmo para multiplicar. La segunda se conoce técnicamente

con el nombre de “vectoring mode”, para valores de y que convergen a cero y se usa en el algoritmo para dividir. Debe notarse que en ambos casos el cálculo de δ_k requiere solamente una comparación.

3. Los números σ_k son constantes almacenadas que dependen de los valores de m . En este artículo se considerará el caso $m = 0$, en dicho caso $\sigma_k = 2^k$. Para $m = 1$, $\sigma_k = \tan^{-1} 2^{-k}$ y para $m = -1$, $\sigma_k = \tanh^{-1} 2^{-k}$, como se verá en la sección 6.

Para usar el algoritmo CORDIC se deben asignar valores iniciales apropiados para x_0, y_0 y z_0 . Los valores iniciales se deben restringir a un intervalo alrededor del origen con el propósito de garantizar la convergencia.

3 División

El algoritmo CORDIC para evaluar el cociente x_0/y_0 está basado en las siguientes ecuaciones iterativas:

$$\begin{aligned} x_{k+1} &= x_0 \\ y_{k+1} &= y_k + \delta_k x_0 2^{-k} \\ z_{k+1} &= z_k - \delta_k 2^{-k} \end{aligned}$$

En este caso $m = 0$ y $\delta_k = -\text{sgn}(y_k)$.

Los valores iniciales son por supuesto los de x_0, y_0 y se toma $z_0 = 0$. En la sección 5 se verá como una vez que las iteraciones van siendo realizadas los valores de y convergen a 0 y los valores de z convergen al cociente y_0/x_0 . También se asumirá que $x_0 > 0$, y $|y_0/x_0| \leq 2$ para garantizar que la sucesión de los correspondientes valores de z converja a y_0/x_0 .

Ejemplo

Supongamos que su calculadora va a evaluar el cociente $y_0/x_0 = 9/5$. A continuación aparecen las primeras cinco iteraciones y las cinco últimas. Téngase en cuenta también que $x_0 = 5$, $y_0 = 9$ y $n = 40$ donde n es el número de iteraciones realizadas.

	x	y	z
0	5.000000000000000	9.000000000000000	0
1	5.000000000000000	4.000000000000000	1.000000000000000
2	5.000000000000000	1.500000000000000	1.500000000000000
3	5.000000000000000	0.250000000000000	1.750000000000000
4	5.000000000000000	-0.375000000000000	1.875000000000000
5	5.000000000000000	-0.062500000000000	1.812500000000000
⋮			
36	5.000000000000000	-0.00000000008731	1.80000000001746
37	5.000000000000000	-0.00000000001455	1.80000000000291
38	5.000000000000000	0.00000000002183	1.79999999999563
39	5.000000000000000	0.00000000000364	1.79999999999927
40	5.000000000000000	-0.00000000000546	1.80000000000109

Nótese como los valores de y se van aproximando a cero, mientras los valores de z se van aproximando al cociente x_0/y_0 . ■

No es difícil ver por que la sucesión de los z_k converge a y_0/x_0 . Las ecuaciones para y_k y z_k en el algoritmo CORDIC implican que

$$y_n = y_0 + \sum_{k=0}^{n-1} \delta_k x_0 2^{-k} \quad \text{y} \quad z_n = - \sum_{k=0}^{n-1} \delta_k 2^{-k},$$

respectivamente. Por lo tanto

$$\frac{y_0}{x_0} = \frac{y_n - \sum_{k=0}^{n-1} \delta_k x_0 2^{-k}}{x_0} = \frac{y_n}{x_0} - \sum_{k=0}^{n-1} \delta_k 2^{-k} = \frac{y_n}{x_0} + z_n,$$

que nos da

$$\left| z_n - \frac{y_0}{x_0} \right| = \left| \frac{y_n}{x_0} \right|.$$

Puesto que y_n es cerca a zero, $z_n \approx y_0/x_0$. Demostraremos esta observación en la sección 5.

4 Multiplicación

El algoritmo CORDIC para evaluar el producto $x_0 z_0$ usa exactamente las mismas ecuaciones iterativas empleadas para la división, excepto que en este caso $\delta_k = \text{sgn}(z_k)$:

$$\begin{aligned} x_{k+1} &= x_0 \\ y_{k+1} &= y_k + \delta_k x_0 2^{-k} \\ z_{k+1} &= z_k - \delta_k 2^{-k} \end{aligned}$$

Los valores iniciales son por supuesto x_0 , z_0 y además $y_0 = 0$. A medida que se realizan las iteraciones, los valores de z convergen a 0 y los valores de y convergen al producto $x_0 z_0$. Se asumirá además que $x_0 > 0$ y que $|z_0| \leq 2$ para garantizar que los valores de y convergen al producto $x_0 z_0$.

Ejemplo

Supongamos que la calculadora va a evaluar el producto $x_0 z_0 = (3)(1.9)$. A continuación aparecen las cinco primeras iteraciones y las cinco últimas para realizar el cálculo con 40 iteraciones. Por supuesto, $x_0 = 3$, $z_0 = 1.9$ y $n = 40$.

	x	y	z
0	3.000000000000000	0	1.900000000000000
1	3.000000000000000	3.000000000000000	0.900000000000000
2	3.000000000000000	4.500000000000000	0.400000000000000
3	3.000000000000000	5.250000000000000	0.150000000000000
4	3.000000000000000	5.625000000000000	0.025000000000000
5	3.000000000000000	5.812500000000000	-0.037500000000000
$\vdots$			
36	3.000000000000000	5.69999999998254	0.000000000000582
37	3.000000000000000	5.70000000002619	-0.000000000000873
38	3.000000000000000	5.70000000000437	-0.000000000000146
39	3.000000000000000	5.69999999999345	0.000000000000218
40	3.000000000000000	5.69999999999891	0.000000000000036

Nótese que los valores de z convergen a cero mientras los valores de y se aproximan al producto $x_0 z_0 = 5.7$. ■

No es difícil observar que la sucesión de los y_k converge a $x_0 z_0$. Note que las ecuaciones básicas para y_{n+1} y z_{n+1} en el algoritmo implican que

$$y_n = \sum_{k=0}^{n-1} \delta_k x_0 2^{-k} \quad \text{y} \quad z_n \neq z_0 - \sum_{k=0}^{n-1} \delta_k 2^{-k},$$

respectivamente. Por lo tanto,

$$x_0 z_0 = x_0 \left(z_n + \sum_{k=0}^{n-1} \delta_k 2^{-k} \right) = x_0 z_n + y_n$$

de lo que se obtiene

$$|y_n - x_0 z_0| = |x_0 z_n|.$$

Y puesto que z_n se aproxima a cero, $y_n \approx x_0 z_0$. Esta última observación será demostrada en la sección 5.

5 Convergencia

Las propiedades de convergencia que poseen las sucesiones determinadas por el algoritmo CORDIC están basadas en el importante teorema de convergencia que sigue a continuación.

Teorema 1 de Convergencia. *Sea $\sigma_0 \geq \sigma_1 \geq \dots \geq \sigma_n > 0$ una sucesión decreciente de números reales positivos con la siguiente propiedad*

$$\sigma_k \leq \sigma_n + \sum_{j=k+1}^n \sigma_j, \text{ para } 0 \leq k < n.$$

Sea r un número real tal que

$$|r| \leq \sigma_n + \sum_{j=0}^n \sigma_j.$$

Defínase la siguiente sucesión de números reales: $s_0 = 0$ y $s_{k+1} = s_k + \rho_k \sigma_k$, $k = 0, 1, \dots, n-1$, donde

$$\rho_k = \operatorname{sgn}(r - s_k) = \begin{cases} 1 & , \quad r \geq s_k \\ -1 & , \quad r < s_k. \end{cases}$$

Entonces

$$|r - s_k| \leq \sigma_n + \sum_{j=k}^n \sigma_j, \text{ para } 0 \leq k \leq n.$$

En particular,

$$|r - s_n| \leq 2\sigma_n.$$

Demostración. La prueba se hace usando inducción sobre k . Para $k = 0$ obtenemos inmediatamente

$$|r - s_0| = |r| \leq \sigma_n + \sum_{j=0}^n \sigma_j.$$

Asumamos que el resultado es válido para k y considérese

$$|r - s_{k+1}| = |r - s_k - \rho_k \sigma_k|.$$

Si $r - s_k \geq 0$, entonces $\rho_k = 1$ y se tiene $|r - s_k - \rho_k \sigma_k| = |r - s_k| - \sigma_k$. Por otro lado, si $r - s_k < 0$, entonces $\rho_k = -1$ y

$$|r - s_k - \rho_k \sigma_k| = |r - s_k + \sigma_k| = |s_k - r - \sigma_k| = |r - s_k| - \sigma_k.$$

Por lo tanto en cualquier caso $|r - s_{k+1}| = |r - s_k - \rho_k \sigma_k| = |r - s_k| - \sigma_k$.

Usando la primera hipótesis se obtiene

$$-\left(\sigma_n + \sum_{j=k+1}^n \sigma_j\right) \leq -\sigma_k \leq |r - s_k| - \sigma_k.$$

Y por la hipótesis de inducción,

$$|r - s_k| - \sigma_k \leq \left(\sigma_n + \sum_{j=k}^n \sigma_j\right) - \sigma_k = \left(\sigma_n + \sum_{j=k+1}^n \sigma_j\right).$$

Al combinar este par de desigualdades, se obtiene

$$|r - s_{k+1}| = |r - s_k| - \sigma_k \leq \sigma_n + \sum_{j=k+1}^n \sigma_j.$$

Lo que completa la prueba al hacer $k+1 = n$ en la última desigualdad. ■

Ahora se demostrará que la sucesión $\sigma_k = 2^{-k}$ que se usa en el algoritmo de la multiplicación y división satisface el teorema de convergencia.

Teorema 2 *La sucesión $\sigma_k = 2^{-k}$, $k = 0, 1, \dots, n$ satisface la hipótesis del teorema de convergencia para todo $-2 \leq r \leq 2$. En particular, $|r - s_n| \leq 1/2^{n-1}$.*

Demostración. La sucesión $1, 1/2, 1/4, \dots, 1/2^n$ es claramente una sucesión decreciente de números positivos. Para todo k entre 0 y n , se tiene

$$\begin{aligned} \sigma_n + \sum_{j=k+1}^n \sigma_j &= \frac{1}{2^n} + \sum_{j=k+1}^n \frac{1}{2^j} \\ &= \frac{1}{2^n} + \frac{1}{2^{k+1}} \left(1 + \frac{1}{2} + \dots + \frac{1}{2^{n-k-1}} \right) \\ &= \frac{1}{2^n} + \frac{1}{2^{k+1}} \left(\frac{1 - \frac{1}{2^{n-k}}}{1 - \frac{1}{2}} \right) \\ &= \frac{1}{2^n} + \frac{1}{2^k} \left(1 - \frac{1}{2^{n-k}} \right) \\ &= \frac{1}{2^n} + \frac{1}{2^k} - \frac{1}{2^n} \\ &= \frac{1}{2^k} = \sigma_k \end{aligned}$$

Además,

$$\sigma_n + \sum_{j=0}^n \sigma_j = \frac{1}{2^n} + \sum_{j=0}^n \frac{1}{2^j} = \frac{1}{2^n} + \frac{1 - \frac{1}{2^{n+1}}}{\frac{1}{2}} = \frac{1}{2^n} + 2 - \frac{1}{2^n} = 2 \geq |r|.$$

Finalmente el Teorema de Convergencia implica que

$$|r - s_n| \leq 2\sigma_n = 2\frac{1}{2^n} = \frac{1}{2^{n-1}},$$

lo que completa la prueba. ■

Ahora estamos listos para probar formalmente que los algoritmos para la multiplicación y la división convergen como se afirmó en las secciones 3 y 4.

Teorema 3: Convergencia de la División. Sea $|y_0/x_0| \leq 2$, $z_0 = 0$, $x_0 > 0$, y $\delta_k = -\text{sgn}(y_k)$ en el algoritmo para dividir.

$$\begin{aligned}x_{k+1} &= x_0 \\y_{k+1} &= y_k + \delta_k x_0 2^{-k} \\z_{k+1} &= z_k - \delta_k 2^{-k}.\end{aligned}$$

Entonces

$$\left| z_n - \frac{y_0}{x_0} \right| \leq \frac{1}{2^{n-1}}.$$

Demostración Sea $r = y_0/x_0$. Se usará inducción sobre k para probar que la sucesión s_k en el teorema de convergencia es precisamente la sucesión z_k y que $\rho_k = \text{sgn}(r - s_k)$ es lo mismo que $-\delta_k = \text{sgn}(y_k)$. Obsérvese que para $k = 0$ se tiene $z_0 = s_0 = 0$ y $\rho_0 = \text{sgn}(r - s_0) = \text{sgn}(y_0/x_0) = \text{sgn}(y_0) = -\delta_0$, desde que $x_0 > 0$ por hipótesis.

Asumamos ahora que $\rho_k = \text{sgn}(r - s_k) = \text{sgn}(y_k) = -\delta_k$, y $s_k = z_k$. Para $k+1$ se tiene $s_{k+1} = s_k + \rho_k 2^{-k} = z_k - \delta_k 2^{-k} = z_{k+1}$. Además,

$$\begin{aligned}r - s_{k+1} &= \frac{y_0}{x_0} - (s_k + \rho_k 2^{-k}) \\&= \frac{y_0}{x_0} - z_k + \delta_k 2^{-k} \\&= \frac{1}{x_0} (y_0 - x_0 z_k + \delta_k x_0 2^{-k}) \\&= \frac{1}{x_0} (y_0 + \sum_{j=0}^{k-1} \delta_j x_0 2^{-j} + \delta_k x_0 2^{-k}) \\&= \frac{1}{x_0} (y_0 + \sum_{j=0}^k \delta_j x_0 2^{-j}) \\&= \frac{1}{x_0} (y_{k+1}).\end{aligned}$$

Y desde que $x_0 > 0$, se obtiene $\rho_{k+1} = \text{sgn}(r - s_{k+1}) = \text{sgn}(y_{k+1}) = -\delta_{k+1}$. Finalmente al usar el teorema de convergencia se concluye que

$$|r - s_n| = \left| \frac{y_0}{x_0} - z_n \right| \leq 2\sigma_n = \frac{1}{2^{n-1}}$$

■

Teorema 4: Convergencia de la Multiplicación. Sea $|z_0| \leq 2$, $y_0 = 0$, $x_0 > 0$, y $\delta_k = \text{sgn}(z_k)$ en el algoritmo para multiplicar, obteniendo

$$\begin{aligned} x_{k+1} &= x_0 \\ y_{k+1} &= y_k + \delta_k x_0 2^{-k} \\ z_{k+1} &= z_k - \delta_k 2^{-k}. \end{aligned}$$

Entonces

$$|y_n - x_0 z_0| \leq \frac{x_0}{2^{n-1}}.$$

Demostración En este caso, sea $r = z_0$ y se demostrará que la sucesión s_k es precisamente la sucesión de los y_k/x_0 . Los pasos son muy parecidos a los del teorema anterior. Es necesario comprobar que $\rho_k = \delta_k$ por inducción matemática y después aplicar el teorema de convergencia para concluir que

$$|r - s_n| = |z_0 - y_n/x_0| \leq 2\sigma_n = \frac{1}{2^{n-1}}$$

y

$$|x_0 z_0 - y_n| \leq \frac{x_0}{2^{n-1}}$$

■

6 Conclusiones

Se ha presentado el algoritmo CORDIC para la multiplicación y división. La tabla presentada a continuación ofrece un bosquejo completo e ilustra al mismo tiempo la gran versatilidad de este algoritmo.

	$\delta_k = \text{sgn}(z_k) \quad (z_k \rightarrow 0)$	$\delta_k = -\text{sgn}(y_k) \quad (y_k \rightarrow 0)$
$m = 0$ $\sigma_k = 2^{-k}$	x_0, z_0 dados, $y_0 = 0$ da $y_n \approx x_0 z_0$	x_0, y_0 dados, $z_0 = 0$ da $z_n \approx y_0/x_0$
$m = 1$ $\sigma_k = \tan^{-1} 2^{-k}$	$x_0 = K, z_0 = \theta, y_0 = 0$ da $x_n \approx \cos \theta, y_n \approx \sin \theta$	x_0, y_0 dados, $z_0 = 0$ $z_n \approx \tan^{-1}(y_0/x_0)$
$m = -1$ $\sigma_k = \tanh^{-1} 2^{-k}$ (algunos σ_k repetidos)	$x_0 = K', z_0 = \theta, y_0 = 0$ da $x_n \approx \cosh \theta, y_n \approx \sinh \theta$ $e^\theta \approx x_n + y_n$	x_0, y_0 dados, $z_0 = 0$ $z_n \approx \tanh^{-1}(y_0/x_0)$ $x_n \approx \sqrt{x_0^2 - y_0^2}/K'$

En esta tabla, $K = \prod_{j=0}^n \cos(\sigma_j)$ y K' es un producto similar al usar la función hiperbólica coseno. Aquí ya se ha mostrado que las constantes $\sigma_k = 2^{-k}$ satisfacen el teorema de convergencia. Además es un fácil ejercicio el mostrar que las constantes $\sigma_k = \tan^{-1} 2^{-k}$ también satisfacen dicho teorema. No obstante para las funciones hiperbólicas, $m = -1$, las constantes $\sigma_k = \tanh^{-1} 2^{-k}$ no satisfacen el teorema para todo valor de k . Por lo tanto, es necesaria la repetición de ciertos valores de σ_k , para los detalles se puede consultar [4].

Para calcular $\ln w$ se puede usar la identidad $\tanh^{-1} t = \frac{1}{2} \ln \frac{1+t}{1-t}$ y el caso hiperbólico CORDIC, inicializando $x_0 = w + 1, y_0 = w - 1$, para obtener:

$$z_n \approx \tanh^{-1}(y_0/x_0) = \frac{1}{2} \ln \left(\frac{1 + y_0/x_0}{1 - y_0/x_0} \right) = \frac{1}{2} \ln \left(\frac{x_0 + y_0}{x_0 - y_0} \right) = \frac{1}{2} \ln w.$$

De esta forma, $\ln w \approx 2z_n$. Finalmente, se puede demostrar que usando $x_0 = w + 0.25$ y $y_0 = w - 0.25$ se obtiene $\sqrt{w} \approx K' x_n$.

Referencias

- [1] . Bruce H. Edwards, Robert T. Jackson, and Jeremy M. Underwood, **How do Calculators Calculate**, Proceedings of the

Ninth Annual International Conference on Technology in Collegiate Mathematics, Reno, Nevada, 1996.

- [2] . Charles W. Schelin, **Calculator Function Approximation**, American Mathematical Monthly 90(5), 1983, pp 317-325.
- [3] . Jack E. Volder, **The CORDIC Trigonometric Computing Technique**, IRE Transactions on Electronic Computers, volume EC-8, No. 3, Septiembre 1959, pp. 330-334.
- [4] . J. S. Walther, **A Unified Algorithm for Elementary Functions**, Joint Computer Conference Proceedings, Spring 1971, pp 379-385.

BRUCE H. EDWARDS

DEPARTMENT OF MATHEMATICS

PO Box 118105

UNIVERSITY OF FLORIDA, GAINESVILLE, FLORIDA 32611

RAFAEL A. DÍAZ

PASADENA INDEPENDENT SCHOOL DISTRICT, SAM RAYBURN HIGH SCHOOL

DEPARTMENT OF MATHEMATICS

2121 CHERRYBROOK LANE

PASADENA, TEXAS 77502

USA.