



Article de synthèse

L'algorithme CORDIC

ÉRIC BRUNELLE, MATHÉMATIQUES,
CÉGEP SAINT-JEAN-SUR-RICHELIEU
eric.brunelle@cstjean.qc.ca

<http://www.cstjean.qc.ca/accueil>

MARC-ANDRÉ DÉSAUTELS, MATHÉMATIQUES,
CÉGEP SAINT-JEAN-SUR-RICHELIEU
marc-andre.desautels@cstjean.qc.ca

<http://www.cstjean.qc.ca/accueil>

Résumé

Cet article décrit un algorithme pour le calcul de certaines fonctions élémentaires telles que le sinus, le cosinus, la tangente et l'arc tangente. Cet algorithme est celui utilisé par les calculatrices. Le principe de cet algorithme est l'utilisation de rotations dans un système de coordonnées circulaires. Les seules opérations nécessaires à l'exécution de l'algorithme sont l'addition, la soustraction, la recherche dans une table et le déplacement d'une virgule dans la représentation d'un nombre en virgule flottante.

Mots clés : CORDIC, trigonométrie, algorithme, calculatrice.

1 Introduction

Pour obtenir un ordinateur de navigation plus performant sur les avions, il fallait trouver une façon de calculer rapidement des fonctions trigonométriques avec une assez grande précision. Afin de minimiser le temps de calcul, il fallait n'utiliser que les quatre opérations les plus rapides d'un ordinateur :

1. Additionner et soustraire deux nombres ;
2. Vérifier si un nombre est plus grand qu'un autre ;
3. Chercher une valeur dans une table mémorisée ;
4. Effectuer des opérations de type bit à bit, c'est-à-dire déplacer une virgule dans la représentation en virgule flottante d'un nombre. Cette opération revient à une multiplication ou une division par 2 dans le système binaire.

La première idée qui nous vient en tête est d'utiliser des approximations par des polynômes de Taylor. Cette approche présente deux grands désavantages :

1. La convergence est lente si nous nous trouvons loin du point de développement.
2. Le temps de calcul est grand avec le calcul des puissances et des divisions.

C'est en 1956 que Jack Volder a développé un algorithme pour le calcul de fonctions trigonométriques. Ensuite, John Stephen Walther a élargi l'algorithme aux calculs de fonctions hyperboliques, exponentielles et logarithmiques, créant ainsi une classe d'algorithmes. De plus, il a démontré que l'algorithme est applicable pour effectuer des multiplications et des divisions de nombres ainsi que le calcul de la racine carrée d'un nombre. Cet algorithme se nomme CORDIC pour "*COordinate Rotation DIgital Computer*". L'idée proviendrait cependant du mathématicien anglais Henry Briggs (1561-1630).

Dans un premier temps, nous développerons l'algorithme pour des fins de calculs de sinus, cosinus ou tangente d'un angle. Par la suite, nous l'utiliserons pour calculer la norme d'un vecteur ainsi que l'angle que celui-ci fait avec l'axe des x positifs. Le calcul de l'angle que le vecteur fait avec l'axe des x positifs revient à calculer une fonction arc tangente.

2 L'algorithme CORDIC pour des fonctions trigonométriques

L'algorithme CORDIC pour les fonctions trigonométriques se divise en deux modes : le mode rotation permettant de calculer le sinus, cosinus et la tangente d'un angle, et le mode vectoriel qui calcule entre autres la norme d'un vecteur et l'arc tangente d'une valeur. Débutons par le mode rotation qui nous permettra de comprendre le principe de l'algorithme.

2.1 Mode trigonométrique : rotation

Nous voulons trouver un algorithme permettant de calculer le sinus, le cosinus et la tangente d'un angle θ . Pour ce faire, supposons que $\theta \in [0, \frac{\pi}{2}]$. Il est aisé de retrouver les valeurs de sinus, de cosinus et de tangente dans les autres cas en utilisant la symétrie du cercle trigonométrique. Comme chaque fois que nous voulons déterminer des rapports trigonométriques, nous utilisons le cercle trigonométrique. Ainsi, nous pouvons faire correspondre à l'angle θ un point (x, y) sur ce cercle. Notons \vec{v} le vecteur position de ce point, c'est-à-dire le vecteur allant de l'origine au point (x, y) . Ce vecteur est de norme 1.

La puissance de l'algorithme CORDIC vient du principe que nous pouvons obtenir \vec{v} en appliquant au vecteur $\vec{v}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ une suite (théoriquement infinie, mais en pratique finie) de

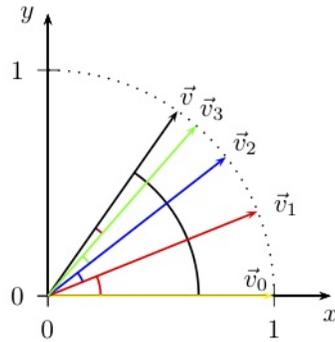


FIGURE 1 – Une suite de vecteurs \vec{v}_i s’approchant du vecteur \vec{v} . Dans ce cas-ci, tous les angles sont positifs, mais en général ce n’est pas le cas.

rotations d’angle θ_i . Ici, prenons une suite finie ou infinie telle que $0 \leq |\theta_{i+1}| < |\theta_i| \leq \frac{\pi}{2}$ (ceci signifie que nous avons une suite d’angles dont les valeurs absolues sont décroissantes) et que

$$\sum_{i=0}^n \theta_i = \theta .$$

Notons que les θ_i peuvent aussi être négatifs.

Soit $\vec{v}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$, le vecteur position obtenu après la $i^{\text{ème}}$ rotation. Alors le vecteur \vec{v}_{i+1} s’obtient par la rotation d’angle θ_i du vecteur \vec{v}_i , ce qui revient au produit matriciel suivant :

$$\vec{v}_{i+1} = \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} . \quad (2.1)$$

La figure 1 montre une suite de rotations du vecteur \vec{v}_0 permettant de s’approcher du vecteur \vec{v} voulu.

Nous désirons tout d’abord éliminer les fonctions trigonométriques dans le calcul précédent. Nous débuterons par mettre en évidence le cosinus dans l’équation 2.1. Nous obtenons l’équation suivante :

$$\vec{v}_{i+1} = \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \cos(\theta_i) \begin{bmatrix} 1 & -\tan(\theta_i) \\ \tan(\theta_i) & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} .$$

Observons maintenant le rôle du cosinus mis en évidence. Puisque les $\theta_i \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, nous avons que $0 \leq \cos(\theta_i) \leq 1$. Ainsi, si nous omettons les $\cos(\theta_i)$, les vecteurs obtenus auront une norme supérieure à 1, mais l’angle reste le même ainsi que les différents rapports trigonométriques.

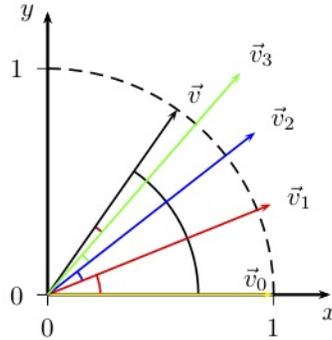


FIGURE 2 – Une suite de vecteurs \vec{v}_i s’approchant du vecteur \vec{v} sans la contraction.

La figure 2 montre une suite de rotations du vecteur \vec{v}_0 permettant de s’approcher du vecteur \vec{v} voulu lorsque nous omettons les $\cos(\theta_i)$. Remarquons que les normes des vecteurs \vec{v}_i augmentent, car les $\cos(\theta_i)$ servent à contracter les vecteurs \vec{v}_i .

Nous laisserons donc le cosinus de côté pour l’instant car, comme mentionné précédemment, les itérations suivantes nous mèneront vers les bons rapports trigonométriques. Notons par x_i^* et y_i^* les composantes de ces vecteurs. Ainsi :

$$\vec{v}_{i+1}^* = \begin{bmatrix} x_{i+1}^* \\ y_{i+1}^* \end{bmatrix} = \begin{bmatrix} 1 & -\tan(\theta_i) \\ \tan(\theta_i) & 1 \end{bmatrix} \begin{bmatrix} x_i^* \\ y_i^* \end{bmatrix} = \begin{bmatrix} x_i^* - \tan(\theta_i) \cdot y_i^* \\ y_i^* + \tan(\theta_i) \cdot x_i^* \end{bmatrix}. \quad (2.2)$$

Il reste toujours deux problèmes dans ces itérations au niveau de la rapidité du calcul informatique : le calcul de $\tan(\theta_i)$ et celui du produit. Par contre, il nous reste la possibilité de choisir les θ_i de façon adéquate. Si nous choisissons les θ_i tels que $\tan(\theta_i) = \pm 2^{-i}$ où $i = 0, 1, 2, \dots$, nous enlevons la problématique de la multiplication, car une multiplication par une puissance de deux en binaire revient à déplacer une virgule. Le signe de la tangente dépend du signe des θ_i , car nous pouvons faire des rotations dans le sens horaire ou anti-horaire.

Il ne reste plus qu’à déterminer les valeurs des θ_i pour la calculatrice. Allons-y en deux temps : la grandeur de l’angle, noté $|\theta_i|$ et son signe, noté σ_i . Nous trouvons les angles $|\theta_i|$ en résolvant l’équation suivante :

$$\tan(|\theta_i|) = 2^{-i}.$$

Ces calculs sont faits une fois pour toutes et les valeurs sont conservées dans la mémoire de l’appareil utilisant l’algorithme CORDIC. Nous gardons habituellement en mémoire les angles $|\theta_i|$ où i va habituellement de 0 à 40.

$\tan(\theta_i)$	$ \theta_i $	σ_i	$\sum \sigma_i \theta_i $
$2^0 = 1$	$0.785398 \equiv 45^\circ$	+	45°
$2^{-1} = 0.5$	$0.4636 \equiv 26.5651^\circ$	+	71.5651°
$2^{-2} = 0.25$	$0.2449 \equiv 14.0362^\circ$	-	57.5288°
$2^{-3} = 0.125$	$0.124 \equiv 7.1250^\circ$	-	50.4038°
$2^{-4} = 0.0625$	$0.0624 \equiv 3.5763^\circ$	+	53.9801°
$2^{-5} = 0.03125$	$0.03122 \equiv 1.7899^\circ$	+	55.7700°
$2^{-6} = 0.015625$	$0.015625 \equiv 0.8952^\circ$	-	54.8749°
$2^{-7} = 0.0078125$	$0.0078125 \equiv 0.4476^\circ$	+	55.3225°
$2^{-8} = 0.00390625$	$0.00390625 \equiv 0.2238^\circ$	-	55.0987°

TABLEAU 1 – Les itérations permettant de s’approcher de l’angle 55° .

Pour déterminer σ_i , la calculatrice garde en mémoire la somme des angles précédents. Si cette somme est inférieure à θ , alors le signe du prochain angle est positif, sinon il sera négatif. Tout cela afin de converger vers θ .

En réécrivant l’équation 2.2 de manière scalaire, nous obtenons :

$$x_{i+1}^* = x_i^* - \sigma_i \cdot 2^{-i} \cdot y_i^* , \quad (2.3)$$

$$y_{i+1}^* = y_i^* + \sigma_i \cdot 2^{-i} \cdot x_i^* . \quad (2.4)$$

Nous voulons représenter les différentes itérations lors du calcul de $\cos(55^\circ)$ et trouver la suite de rotations $\sum \sigma_i |\theta_i|$ qui nous permettra de s’approcher de la valeur voulue. Le tableau 1 nous donne la suite des premières itérations pour y parvenir. Rappelons que les $|\theta_i|$ sont fixes.

Le tableau 2 présente les valeurs des x_i^* et y_i^* . La figure 3 représente graphiquement la suite des itérations.

Nous sommes maintenant en mesure de déterminer les rapports trigonométriques à l’aide de la dernière itération des coordonnées x et y . Par contre, il faut calculer une racine carrée pour le sinus et le cosinus (pour la longueur de l’hypoténuse), ce qui est une opération qui ne fait pas partie des quatre opérations rapides de la calculatrice.

L’astuce pour éviter cette opération est d’utiliser les $\cos(\theta_i)$ que nous avons mis de côté précédemment.

Si nous effectuons n itérations à l’aide de l’algorithme CORDIC, le vecteur final est multiplié par le produit $K_n = \cos(\theta_0) \cdot \cos(\theta_1) \cdot \dots \cdot \cos(\theta_n)$. Puisque $\tan(|\theta_i|) = 2^{-i}$, il est possible de

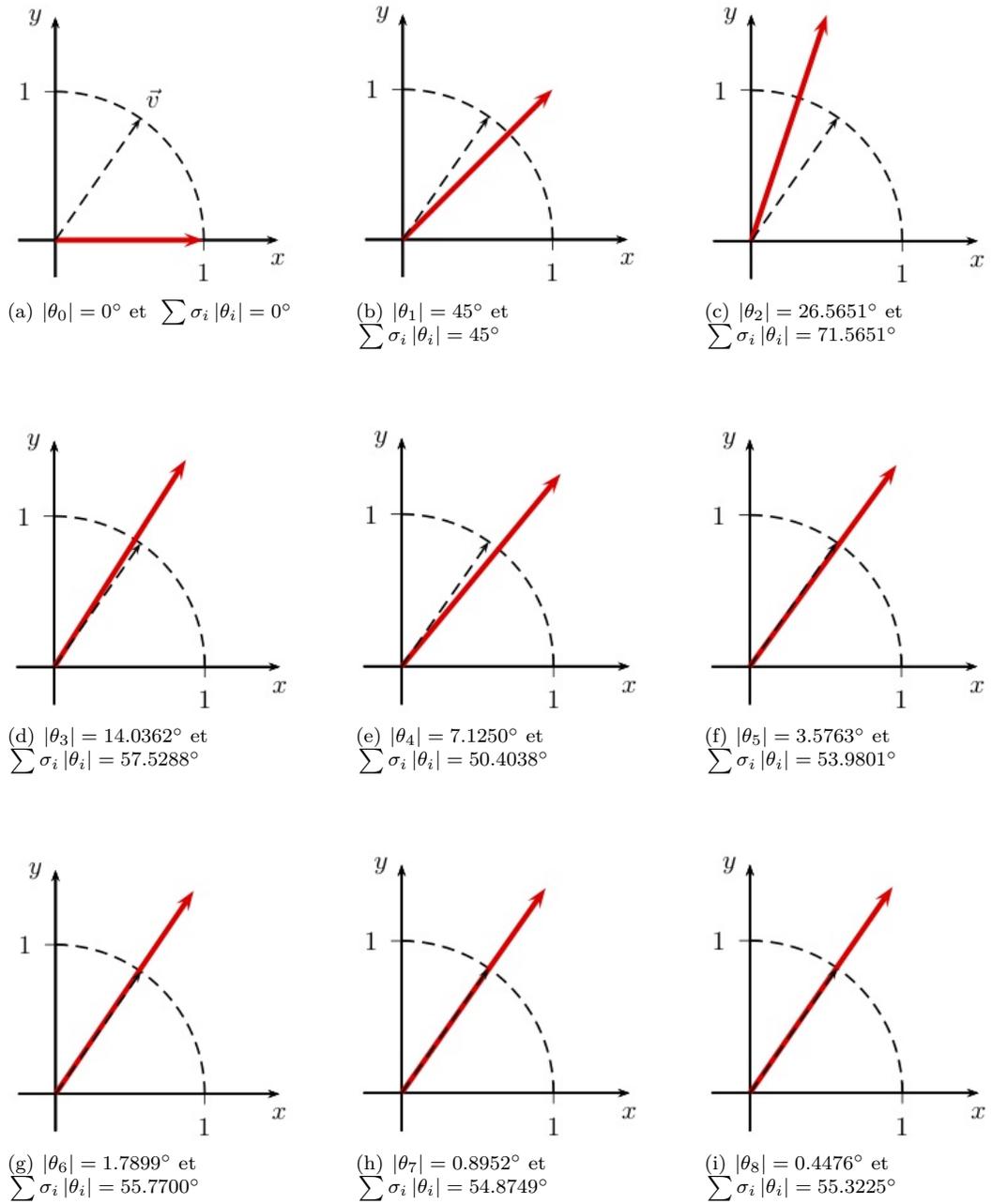


FIGURE 3 – La suite des itérations (sans contraction) permettant de s’approcher du vecteur ayant subi une rotation de 55° . Nous lisons la figure de haut en bas et de gauche à droite

$\sum \sigma_i \theta_i $	x_i^*	y_i^*	$\sqrt{(x_i^*)^2 + (y_i^*)^2}$
0°	1	0	1
45°	1	1	1.4142
71.5651°	0.5	1.5	1.5811
57.5288°	0.875	1.375	1.6298
50.4038°	1.0469	1.2656	1.6425
53.9801°	0.96777	1.3311	1.6457
55.77°	0.92618	1.3613	1.6465
54.8749°	0.94745	1.3468	1.6467
55.3225°	0.93693	1.3542	1.6467

TABLEAU 2 – Les itérations permettant de s’approcher du cosinus de l’angle 55° . La dernière colonne permet de constater que l’absence des facteurs $\cos(\theta_i)$ dilate les vecteurs et que la longueur des vecteurs semble tendre vers une constante.

précalculer les divers $\cos(\theta_i)$. En effet,

$$\cos(\theta_i) = \frac{1}{\sqrt{\sec^2(\theta_i)}} = \frac{1}{\sqrt{1 + \tan^2(\theta_i)}} = \frac{1}{\sqrt{1 + 2^{-2i}}} .$$

Nous avons donc :

$$K_n = \prod_{i=0}^n \cos(\theta_i) = \prod_{i=0}^n \frac{1}{\sqrt{1 + 2^{-2i}}} .$$

Il est possible de garder ce produit en mémoire dans la calculatrice pour, par exemple, les 40 premières itérations. En effet, l’algorithme CORDIC effectue habituellement un nombre fixe d’itérations, soit 40. Il est aussi possible de démontrer que ce produit converge vers une valeur près de 0.60725293500899, valeur que nous pouvons garder également en mémoire. Le calcul de ce facteur se fait à l’aide d’un logiciel.

Plutôt que d’effectuer l’algorithme CORDIC et de multiplier à la fin le vecteur obtenu par la constante K_n , il est possible de débiter l’algorithme, non pas avec le vecteur $[1, 0]$, mais plutôt avec le vecteur

$$[\cos(\theta_0) \cdot \cos(\theta_1) \cdot \dots \cdot \cos(\theta_n), 0] = [K_n, 0] .$$

Nous débutons donc l’algorithme avec un vecteur de norme inférieure à 1, mais au fil des itérations ce vecteur se dilatera et finira avec une norme 1.

Le tableau 3 présente les itérations dans le cas où $n = 9$ pour la recherche du $\cos(55^\circ)$ ou du $\sin(55^\circ)$. Ainsi, $\cos(55^\circ) \approx x_9^* = 0.57216$ et $\sin(55^\circ) \approx y_9^* = 0.82014$. La figure 4 présente graphiquement la suite des itérations.

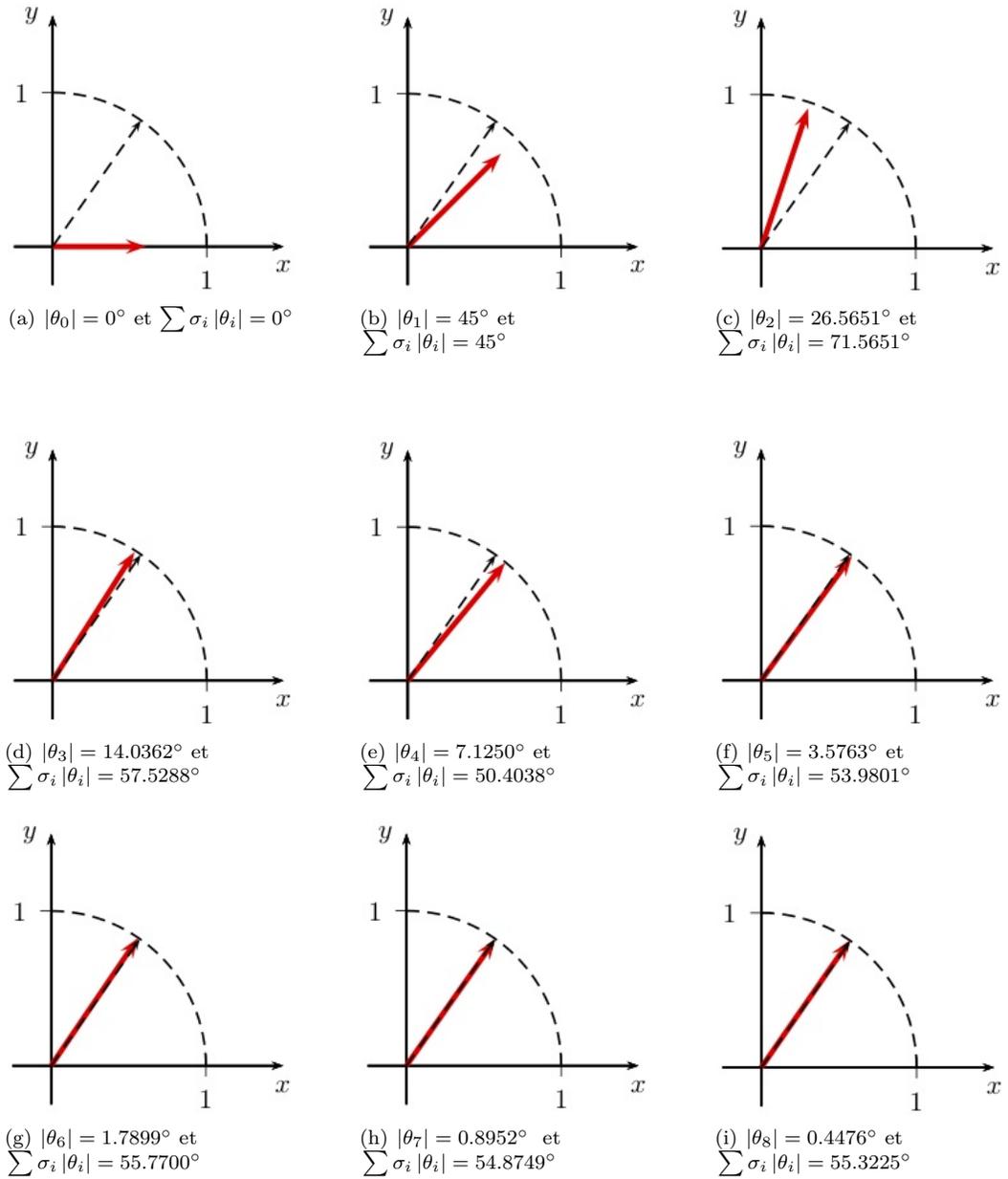


FIGURE 4 – La suite des itérations (avec contraction) permettant de s’approcher du vecteur ayant subi une rotation de 55° . Nous lisons la figure de haut en bas et de gauche à droite

$\sum \sigma_i \theta_i $	x_i^*	y_i^*	$\sqrt{(x_i^*)^2 + (y_i^*)^2}$
0.0000°	0.60725	0	0.60725
45.0000°	0.60725	0.60725	0.85879
71.5651°	0.30363	0.91088	0.96015
57.5288°	0.53135	0.83497	0.9897
50.4038°	0.63572	0.76856	0.99741
53.9801°	0.58768	0.80829	0.99935
55.7700°	0.56243	0.82665	0.99984
54.8749°	0.57534	0.81787	0.99996
55.3225°	0.56895	0.82236	0.99999
55.0987°	0.57216	0.82014	1.00000

TABLEAU 3 – Les itérations successives pour trouver à la fois $\cos(55^\circ)$ et $\sin(55^\circ)$.

2.2 Mode trigonométrique : vectoriel

Le second mode dans lequel nous pouvons utiliser l'algorithme CORDIC est le mode vectoriel. Avec ce mode, nous désirons trouver l'angle et la norme d'un vecteur de \mathbb{R}^2 . Pour simplifier la discussion, nous utiliserons un vecteur se trouvant dans le premier quadrant du cercle trigonométrique. Par des arguments de symétrie, il est possible d'utiliser ces résultats pour des vecteurs se trouvant dans les trois autres quadrants. Ce mode revient à appliquer l'algorithme à l'envers, c'est-à-dire trouver les rotations qui partent d'un vecteur \vec{v} donné pour obtenir le vecteur \vec{v}_0 dont la deuxième composante est nulle. Ce mode permet aussi de calculer l'arc tangente.

Soit le vecteur de départ $\vec{v} = \begin{bmatrix} x \\ y \end{bmatrix}$. Dans ce mode, nous désirons effectuer une suite de rotations pour rendre la coordonnée y du vecteur de départ aussi près de zéro que possible. Utilisons les équations 2.3 et 2.4 trouvées précédemment. Si la coordonnée y est positive, nous effectuons une rotation en sens anti-horaire et sinon nous effectuons une rotation en sens horaire. Après ces rotations, le vecteur obtenu sera de la forme $\begin{bmatrix} \sqrt{x^2 + y^2} \\ K_n \\ 0 \end{bmatrix}$ et si nous calculons la somme $\sum \sigma_i |\theta_i|$, nous aurons calculé l'angle que fait le vecteur avec l'axe des x positifs, ce qui correspond à $\text{Arctan}\left(\frac{y}{x}\right)$.

Calculons par exemple la norme et l'angle du vecteur $\vec{v} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$. Le tableau 4 présente la suite d'itérations permettant de trouver la norme et l'angle du vecteur \vec{v} . Comme le montre la colonne de droite de ce tableau, la norme du vecteur \vec{v}_i^* est d'abord inférieure à 1 et tend vers 1 au fur et à mesure des rotations.

$ \theta_1 $	σ_i	$\sum \sigma_i \theta_i $	x_i^*	y_i^*
0°		0°	3	4
45°	-	45°	7	1
26.5651°	+	71.5651°	7.5	-2.5
14.0362°	-	57.5288°	8.125	-0.625
7.1250°	-	50.4038°	8.2031	0.39063
3.5763°	+	53.9801°	8.2275	-0.12207
1.7899°	-	52.1902°	8.2314	0.13504
0.8952°	+	53.0854°	8.2335	0.0064254
0.4476°	+	53.5330°	8.2335	-0.057899
0.2238°	-	53.3092°	8.2337	-0.025736

TABLEAU 4 – La suite d’itérations permettant de trouver la norme et l’angle du vecteur \vec{v}

Nous remarquons donc que le vecteur $\vec{v} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$ fait un angle d’environ 53.3092° avec l’axe des x positifs. Pour calculer la norme de ce vecteur, il reste à multiplier la composante x du vecteur obtenu par K_n . Nous avons donc :

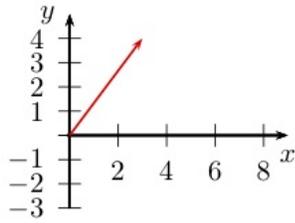
$$\|\vec{v}\| \approx K_9 \cdot 8.2337 = 0.6073 \cdot 8.2337 = 5.0003 .$$

La figure 5 présente graphiquement la suite des itérations.

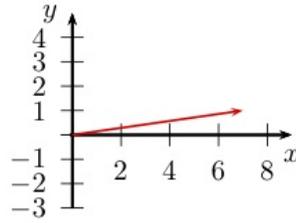
2.3 Convergence de l’algorithme

Une des questions qui mérite une réponse est de savoir si l’algorithme converge pour n’importe quel angle dans $[0, \frac{\pi}{2}]$. En pratique, comme la calculatrice n’effectue que 40 itérations lorsqu’elle effectue l’algorithme, cela donne une précision de 10 chiffres après la virgule.

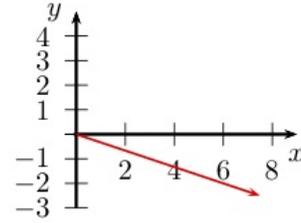
Puisque la calculatrice ne fait que 40 itérations et qu’à chaque itération nous avons le choix du signe de l’angle, il y a alors 2^{40} angles possibles, ce qui ne représente pas tous les angles dans cet intervalle. Nous pouvons montrer que l’angle maximal que peut atteindre l’algorithme est 1.73839 radians (cela se démontre une fois que nous avons calculé toutes les valeurs d’angles que nous désirons utiliser). Cet angle maximal est supérieur à $\frac{\pi}{2}$. Ainsi, pour la précision de la calculatrice, nous pouvons dire que l’algorithme converge pour $\theta \in [0, \frac{\pi}{2}]$.



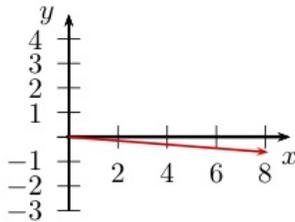
(a) $|\theta_0| = 0^\circ$ et $\sum \sigma_i |\theta_i| = 0^\circ$



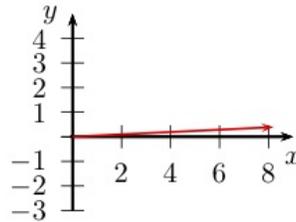
(b) $|\theta_1| = 45^\circ$ et $\sum \sigma_i |\theta_i| = 45^\circ$



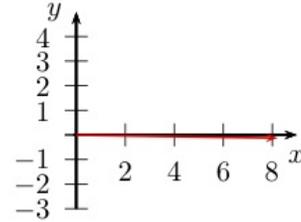
(c) $|\theta_2| = 26.5651^\circ$ et $\sum \sigma_i |\theta_i| = 71.5651^\circ$



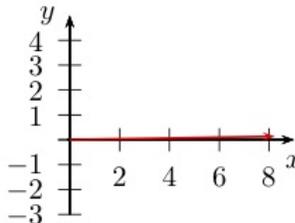
(d) $|\theta_3| = 14.0362^\circ$ et $\sum \sigma_i |\theta_i| = 57.5288^\circ$



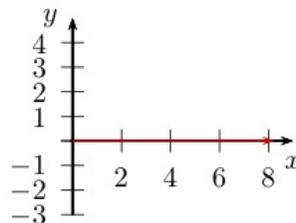
(e) $|\theta_4| = 7.1250^\circ$ et $\sum \sigma_i |\theta_i| = 50.4038^\circ$



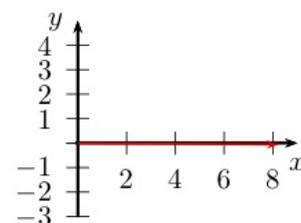
(f) $|\theta_5| = 3.5763^\circ$ et $\sum \sigma_i |\theta_i| = 53.9801^\circ$



(g) $|\theta_6| = 1.7899^\circ$ et $\sum \sigma_i |\theta_i| = 52.1902^\circ$



(h) $|\theta_7| = 0.8952^\circ$ et $\sum \sigma_i |\theta_i| = 53.0854^\circ$



(i) $|\theta_8| = 0.4476^\circ$ et $\sum \sigma_i |\theta_i| = 53.5330^\circ$

FIGURE 5 – La suite d'itérations permettant de rendre la composante en y du vecteur le plus près de 0 possible. Nous lisons la figure de haut en bas et de gauche à droite.

3 Conclusion

L'algorithme CORDIC utilisé en mode trigonométrique, comme présenté dans ce texte, permet donc de calculer le sinus et le cosinus d'un angle donné ainsi que la norme et l'angle d'un vecteur donné (ce qui implique que nous sommes en mesure de calculer la fonction arc tangente).

Puisque nous sommes en mesure de calculer le sinus et le cosinus d'un angle, il est simple de calculer les autres rapports trigonométriques :

$$\tan(\theta) = \frac{\sin(\theta)}{\cos(\theta)}, \quad \cot(\theta) = \frac{\cos(\theta)}{\sin(\theta)}, \quad \sec(\theta) = \frac{1}{\cos(\theta)}, \quad \csc(\theta) = \frac{1}{\sin(\theta)} .$$

Pour calculer l'arc tangente d'une certaine valeur, par exemple $\text{Arctan}(A)$, il suffit d'utiliser l'algorithme CORDIC à l'envers en utilisant le vecteur $\vec{v}_0 = \begin{bmatrix} 1 \\ A \end{bmatrix}$ comme vecteur de départ. De plus, nous pourrions calculer les autres fonctions trigonométriques inverses en utilisant les identités suivantes :

$$\begin{aligned} \text{Arcsin}(x) &= 2 \text{Arctan} \left(\frac{x}{1 + \sqrt{1 - x^2}} \right) , \\ \text{Arccos}(x) &= \frac{\pi}{2} - 2 \text{Arctan} \left(\frac{x}{1 + \sqrt{1 - x^2}} \right) , \\ \text{Arcsec}(x) &= \frac{\pi}{2} - 2 \text{Arctan} \left(\frac{1/x}{1 + \sqrt{1 - (1/x)^2}} \right) , \\ \text{Arccsc}(x) &= \text{Arctan} \left(\frac{1/x}{1 + \sqrt{1 - (1/x)^2}} \right) , \\ \text{Arccot}(x) &= \frac{\pi}{2} - \text{Arctan}(x) . \end{aligned}$$

Malheureusement, comme nous pouvons le constater, ces identités demandent d'effectuer des calculs de racine carrée. En pratique, il est possible de calculer un arcsinus à l'aide de l'algorithme CORDIC, mais celui-ci se complique et dépasse le cadre de cet article.

Références

- [1] Volder, Jack E., (1959). *The CORDIC Trigonometric Computing Technique*, *IRE Transactions on Electronic Computers*, pp. 330-334.
- [2] Walther, John S., (1971). *A Unified Algorithm for Elementary Functions*, *Proc. of Spring Joint Computer Conference*, pp. 379-385.

- [3] Andraka, Ray, (1998). *A Survey of CORDIC Algorithms for FPGA Based Computers, Proceedings of the 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays*, pp. 191-200,.
- [4] Devalland, Christophe, <http://cdeval.free.fr/IMG/pdf/cordic.pdf>, consulté le 14 janvier 2015.