CORDIC

***CO**rdinate **R**otation **DI**gital **C**omputer*

Dominique Dallet

IMS Laboratory

dominique.dallet@ims-bordeaux.fr

Schedule

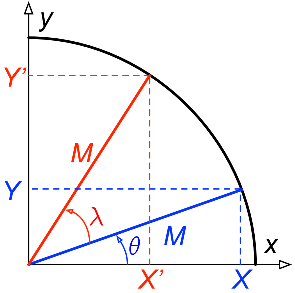
Why CORDIC algorithm ?

- Alternative way to implement trigonometric functions proposed by J.E. Volder [1], solving either :

$$\begin{cases} X' &= M\left(X\cos(\lambda) - Y\sin(\lambda)\right) \\ Y' &= M\left(Y\cos(\lambda) + X\sin(\lambda)\right) \end{cases}$$

or

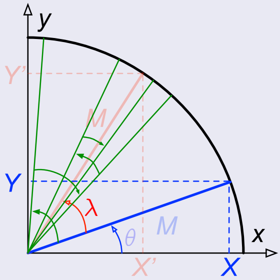
$$\begin{cases} R &= M\sqrt{(X^2 + Y^2)} \\ \theta &= \arctan(Y/X) \end{cases}$$



- It allows the calculation of $\sqrt{a^2 + b^2}$ and it has been extended to hyperbolic functions, multiplications and divisions.
- Using minor modifications, it allows the calculation of other functions such as $\sqrt{()}$, exp() and log()

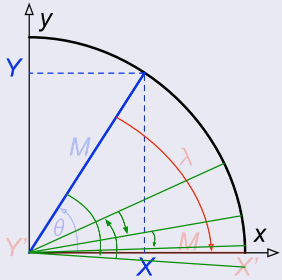
How does it work ?

Rotation



The coordinate components of a vector and a angle rotation are given and the coordinate components of the original vector, after **rotation** through the given angle, are computed [1].

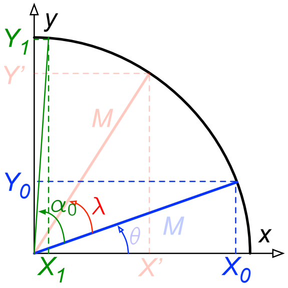
Vectoring



The coordinate components of a vector and an angle of rotation are given and, using **vectoring** the magnitude and angular argument of the original vector are computed [1].

$$\begin{cases} X_0 = M\cos(\theta) \\ Y_0 = M\sin(\theta) \end{cases}$$

$$\begin{cases} X_1 = M\cos(\theta+\alpha_0) \\ Y_1 = M\sin(\theta+\alpha_0) \end{cases}$$



Mathematical expression

$$\begin{aligned} X_1 &= M\cos(\theta)\cos(\alpha_0) - M\sin(\theta)\sin(\alpha_0) = X_0\cos(\alpha_0) - Y_0\sin(\alpha_0) \\ Y_1 &= M\sin(\theta)\cos(\alpha_0) + M\cos(\theta)\sin(\alpha_0) = Y_0\cos(\alpha_0) + X_0\sin(\alpha_0) \end{aligned}$$

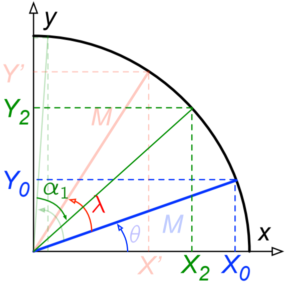
Matrix form

$$\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha_0) & -\sin(\alpha_0) \\ \sin(\alpha_0) & \cos(\alpha_0) \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} = ROT(\alpha_0) \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

GIVENS rotation transform (perfect rotation)

$$\begin{cases} X_0 & = & M\cos(\theta) \\ Y_0 & = & M\sin(\theta) \end{cases}$$

$$\begin{cases} X_2 & = & M\cos(\overbrace{\theta+\alpha_0}^{\theta_1} - \alpha_1) \\ Y_2 & = & M\sin(\underbrace{\theta+\alpha_0}_{\theta_1} - \alpha_1) \end{cases}$$



Mathematical expression (with $d_1 = -1$)

$$\begin{aligned} X_2 &= X_1\cos(-\alpha_1) - Y_1\sin(-\alpha_1) = X_1\cos(\alpha_1) - d_1 Y_1\sin(-\alpha_1) \\ Y_2 &= Y_1\cos(-\alpha_1) + X_1\sin(-\alpha_1) = Y_1\cos(\alpha_1) + d_1 X_1\sin(\alpha_1) \end{aligned}$$

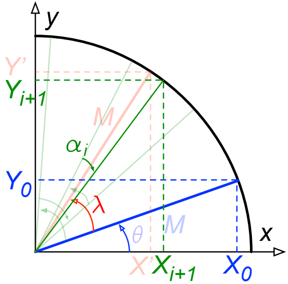
Matrix form

$$\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \cos(\alpha_1) & -d_1\sin(\alpha_1) \\ \sin(\alpha_1) & d_1\cos(\alpha_1) \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = ROT(\alpha_1) \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$$

GIVENS rotation transform (perfect rotation)

$$\lambda = \sum_{i=0}^{\infty} d_i \alpha_i$$

$$ROT(\lambda) = \prod_{i=0}^{\infty} ROT(\alpha_i)$$



Mathematical expression (with $d_i = \pm 1$)

$$\begin{aligned} X_{i+1} &= X_i \cos(\alpha_i) - d_i Y_i \sin(\alpha_i) \\ Y_{i+1} &= Y_i \cos(\alpha_i) + d_i X_i \sin(\alpha_i) \end{aligned}$$

Matrix form

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \prod_{i=0}^{\infty} ROT(\alpha_i) \begin{bmatrix} X \\ Y \end{bmatrix}$$

Schedule

How to choose α_i ?

Imlementation constraint [1]

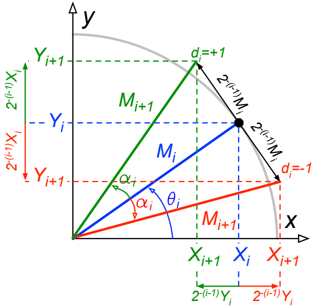
The "rotation" of coordinate components through $d_i\alpha_i = \pm\alpha_i$, should be accomplished by the simple process of **shifting** and **adding**

$$
\begin{aligned}
X_{i+1} &= X_i\cos(\alpha_i) - d_i Y_i \sin(\alpha_i) \\
Y_{i+1} &= Y_i\cos(\alpha_i) + d_i X_i \sin(\alpha_i)
\end{aligned}
$$

Algorithm based on pseudo rotations

1. First rotation with $\alpha_0 = \pm\pi/2$ leading to :
$$
\begin{aligned}
X_1 &= -d_0 Y_0 \\
Y_1 &= +d_0 X_0
\end{aligned}
$$
2. Rest of computing steps with $\alpha_i = \arctan\left(2^{-(i-1)}\right)$, $i \geq 1$, leading to :
$$
\begin{aligned}
X_{i+1} &= X_i - d_i \cdot 2^{-(i-1)} \cdot Y_i \\
Y_{i+1} &= Y_i + d_i \cdot 2^{-(i-1)} \cdot X_i
\end{aligned}
$$

Pseudo rotation behavior

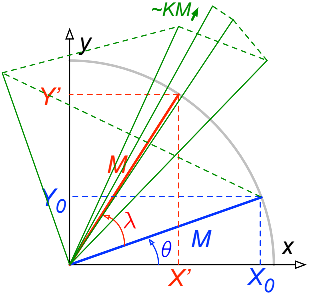


$$
\begin{aligned}
X_{i+1} &= \sqrt{1+2^{-2(i-1)}}M_i\cos(\theta_i + d_i\alpha_i) = X_i - d_i 2^{-(i-1)}Y_i \\
Y_{i+1} &= \sqrt{1+2^{-2(i-1)}}M_i\sin(\theta_i + d_i\alpha_i) = Y_i + d_i 2^{-(i-1)}X_i
\end{aligned}
$$

Using $X_0 = M\cos(\theta)$ and $Y_0 = M\sin(\theta)$, after $n+1$ rotations

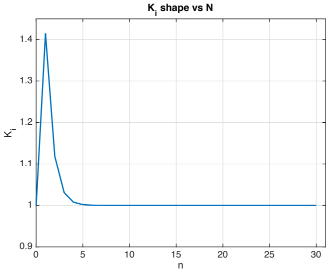
$$
\begin{aligned}
X_{n+1} &= \overbrace{\sqrt{1+2^{-0}}\sqrt{1+2^{-2}}\ldots\sqrt{1+2^{-2(n-1)}}}^{K} M\cos(\theta + d_0\pi/2 + d_1\alpha_1 + \ldots + d_n\alpha_n) \\
Y_{n+1} &= \underbrace{\sqrt{1+2^{-0}}\sqrt{1+2^{-2}}\ldots\sqrt{1+2^{-2(n-1)}}}_{K} M\sin(\theta + d_0\pi/2 + d_1\alpha_1 + \ldots + d_n\alpha_n)
\end{aligned}
$$

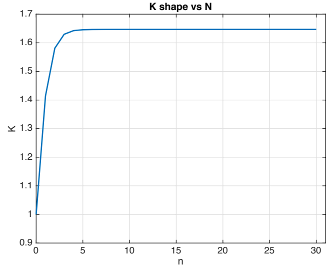
What's about K ?



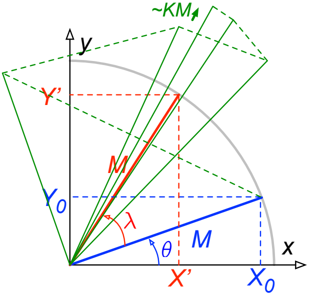
$$K = \prod_{1}^{n} \sqrt{1 + 2^{-2(i-1)}}$$

For $n = 14 : \ldots, \; K \approx 1.64676025\ldots$



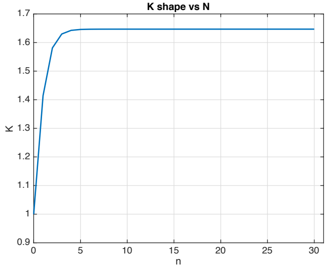


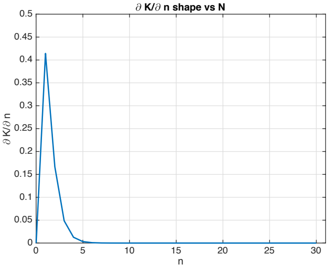
What's about K ?



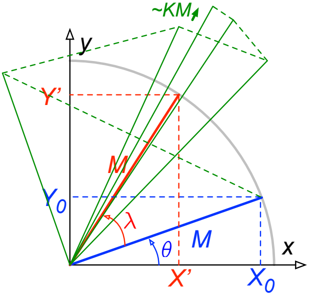
$$K = \prod_{1}^{n} \sqrt{1 + 2^{-2(i-1)}}$$

For $n = 14 : \ldots, K \approx 1.64676025\ldots$



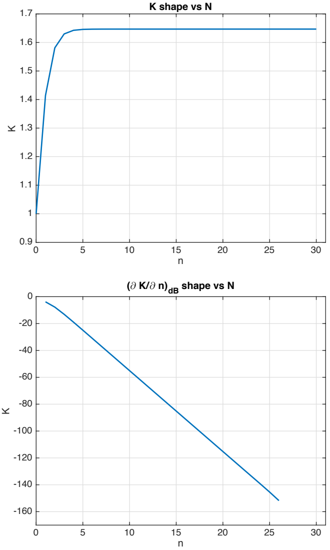


What's about K ?

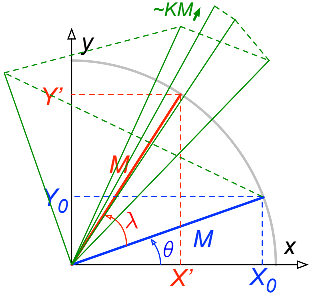


$$K = \prod_{1}^{n} \sqrt{1 + 2^{-2(i-1)}}$$

For $n = 14 : \ldots, K \approx 1.64676025\ldots$

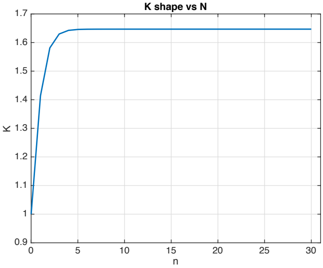


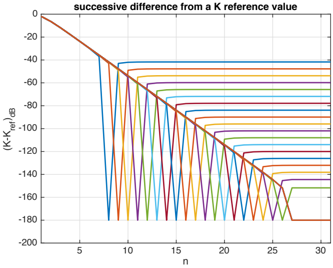
What's about K ?



$$K = \prod_{1}^{n} \sqrt{1 + 2^{-2(i-1)}}$$

For $n = 14 : \ldots,\ K \approx 1.64676025\ldots$





How to control the desired over-all rotation ? [1]

From

$$X_{n+1} = \sqrt{1+2^{-0}}\sqrt{1+2^{-2}}\ldots\sqrt{1+2^{-2(n-1)}}M\cos\left(\theta + d_0\pi/2 + d_1\alpha_1 + \ldots + d_n\alpha_n\right)$$
$$Y_{n+1} = \sqrt{1+2^{-0}}\sqrt{1+2^{-2}}\ldots\sqrt{1+2^{-2(n-1)}}M\sin\left(\theta + d_0\pi/2 + d_1\alpha_1 + \ldots + d_n\alpha_n\right)$$

it is necessary to obtain these expressions :

$$\begin{aligned} X_{n+1} &= KM\cos(\theta+\lambda) \\ Y_{n+1} &= KM\sin(\theta+\lambda) \end{aligned}$$

Rotation

$$\lambda \cong d_0\pi/2 + d_1\alpha_1 + \ldots + d_n\alpha_n$$

Vectoring

$$-\theta \cong d_0\pi/2 + d_1\alpha_1 + \ldots + d_n\alpha_n$$

Some properties associated with α_i

1. For any angle, λ or θ, there must be at least one set of values for the d_i operators that will satisfy the previous relations
2. $$\begin{aligned} |\lambda,\theta| &\leq \alpha_0 + \alpha_1 + \ldots + \alpha_n + \alpha_n \rightarrow -\pi \leq \lambda,\theta \leq +\pi \\ \alpha_i &\leq \alpha_{i+1} + \alpha_{i+2} + \ldots + \alpha_n + \alpha_n \end{aligned}$$

How to control the desired over-all rotation ? [1]

Rotation mode

1. Sensing the sign of the angle of the rotation (or remainder if $i \geq 1$)
2. Either substracting or adding the α_i constant corresponding to the particular step :

$$|\lambda_{i+1}| = ||\lambda_i| - \alpha_i|$$

☞ taking into account the first step, it results in :

$$-\alpha_0 \leq |\lambda| - \alpha_0 \leq \alpha_1 + \alpha_2 + \ldots + \alpha_n + \alpha_n$$

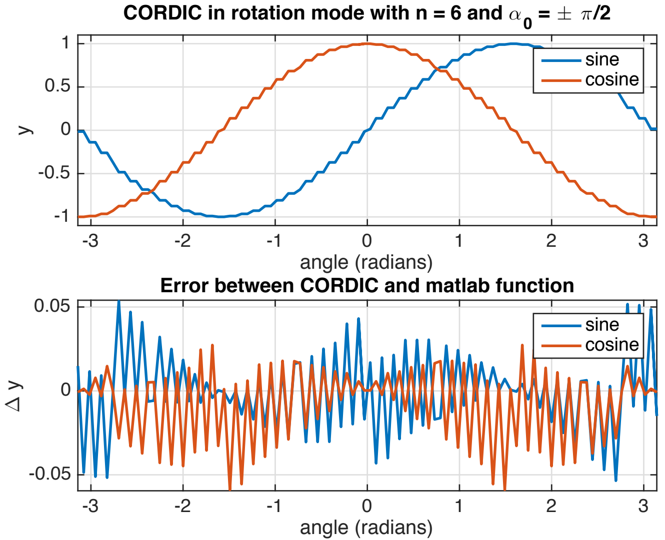
☞ Continuation of the nulling sequence through α_n results in :

$$|\lambda_{n+1}| \leq \alpha_n$$

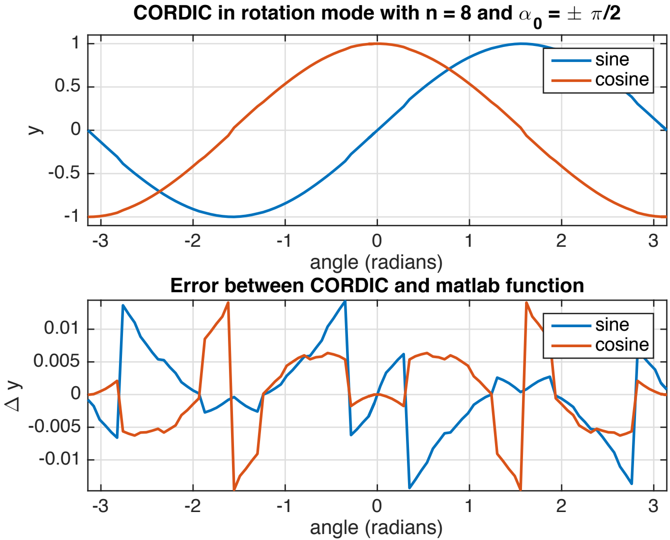
Vectoring mode

1. λ is replaced by θ
2. The sign of the angle θ_i is obtained by sensing the sign of Y_i

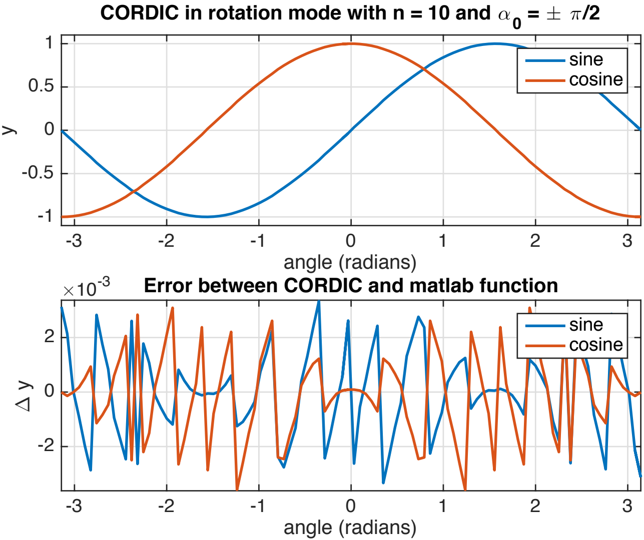
First results wit different n values



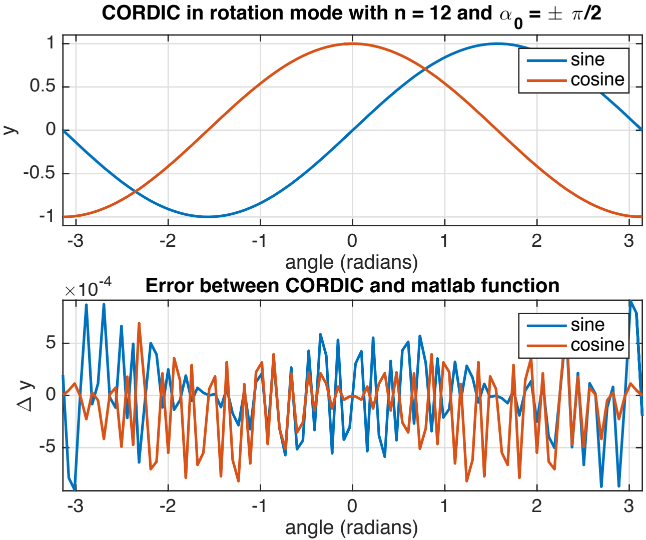
First results wit different n values



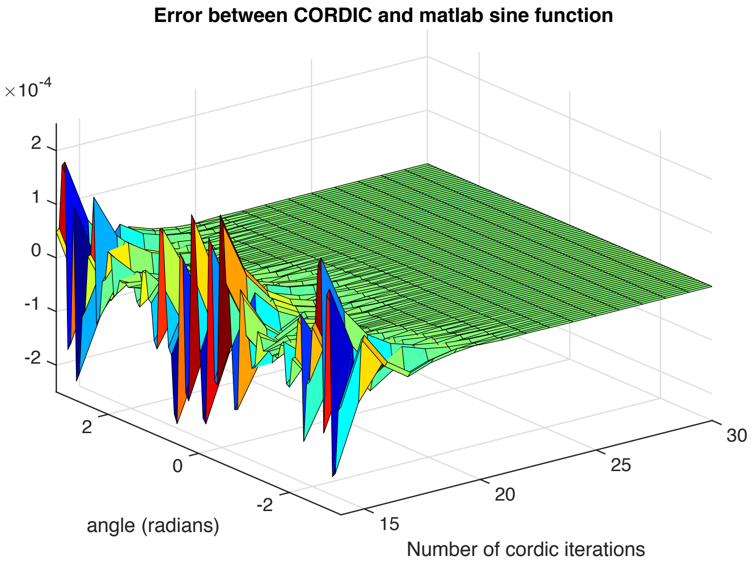
First results wit different *n* values



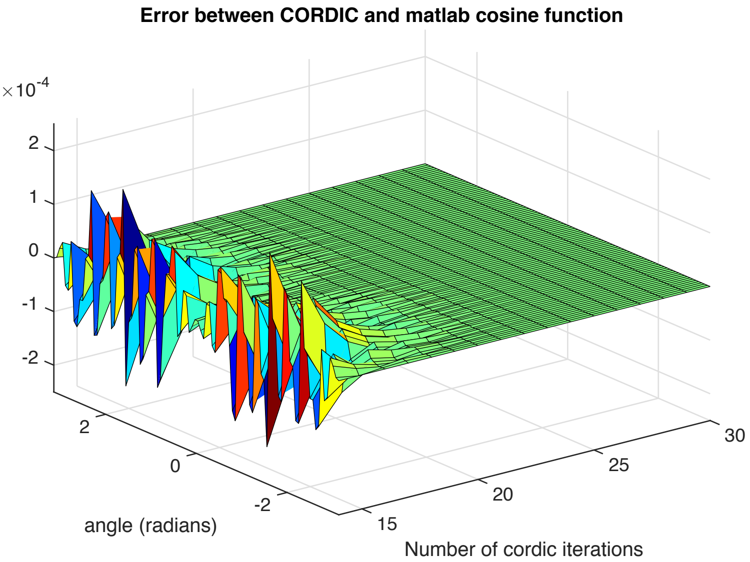
First results wit different n values



First results wit different n values



First results wit different n values



References I

[1] Jack E. Volder.
The cordic trigonometric computing technique.
IRE Transactions on Electronic Computers, EC-8(3) :330–334, September 1959.