“CORDIC を理解すること”にチャレンジする

2014.11.15 JA1VCW

1． はじめに

デジタル処理の本などを見ていると、sin, cos などの関数が出てくる。 ハードやプログラムでそれらの値を求めるための方法に “CORDIC” なるものがある。 関数電卓などに使われているらしい。
好奇心からインターネットで検索すると、解説記事がいくつもでてくるが、解らない！。 そのなかで
http://teamcoil.sp.u-tokai.ac.jp/calculator/100224/
というホームページ（HP）がみつかった。 どうやらこのアルゴリズム解説の定番のHP のようである。
曰く “サルでも分かるCORDIC アルゴリズム” だそうで、読ませていただくとてても上手に丁寧に説明されている。 読んで “解かった” となればサル同等以上になるが私の場合はそうはいかなかった。
まあ、私の知力がサル以下であることを認めるにやぶさかではないが、それもちょっとくやしいのでどこまで理解できるかチャレンジしてみようと思った。

2． このメモの内容について

前記HP を読んでおさらいした事のメモである。 オリジナルはあくまでもHP である。
このメモも個人的なものであるので、いろいろ不備があるのはご容赦。
多くの方はHP を読めばすぐに理解されるから、その時はこのメモは無用なので即刻破棄でOK。
また、HP を読んで解かったような気になっても、“それじゃやってみ” ということになったらできない事も多いので、冗長であるが一つ一つ確認しながら進めた。

3． 理解について

どうなったら理解できたといえるか、というのはよく分からないが、今回は“この方法で計算をさせて結果が合っている事”を勝手に目標とする。 実際は自分でExel で表計算を作り、またプログラムを書て正しい計算値が出れば良いとした。

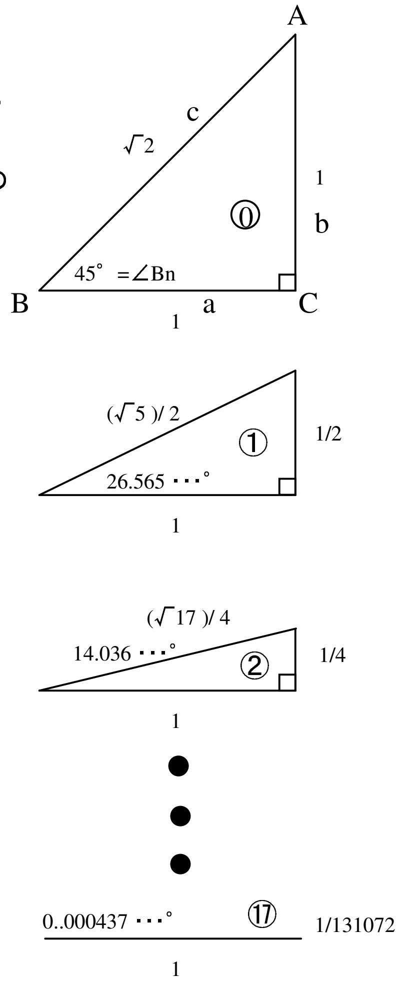
4． CORDIC とは

CORDIC は、与えられた角度αの三角比である sinα, cosα, tanα などの値を求めるための方法の一つである。
三角比以外にもlogや指数、平方根などの値も同様のアルゴリズムによって得られる。
この方法は加算、減算、テーブルの参照などを使用して、初期値から初めて必要な精度になるまで反復演算を行うことによってその値を求める。
特徴的な事は、反復演算で時間や手間のかかる掛け算や割り算を、なくす、或は最小にする、ビットシフトで実現する、などの工夫によってハードウェアやプログラムの量や実行時間が少なくるようなアルゴリズムである。
最初は前記HPにある一番理解し易そうな三角比についての手順を試した。
その後 arctan（Y/X）や√（X^2＋Y^2）についてもチャレンジした。

5.　CORDICを使用して三角比を求める

5.１ 計算の準備

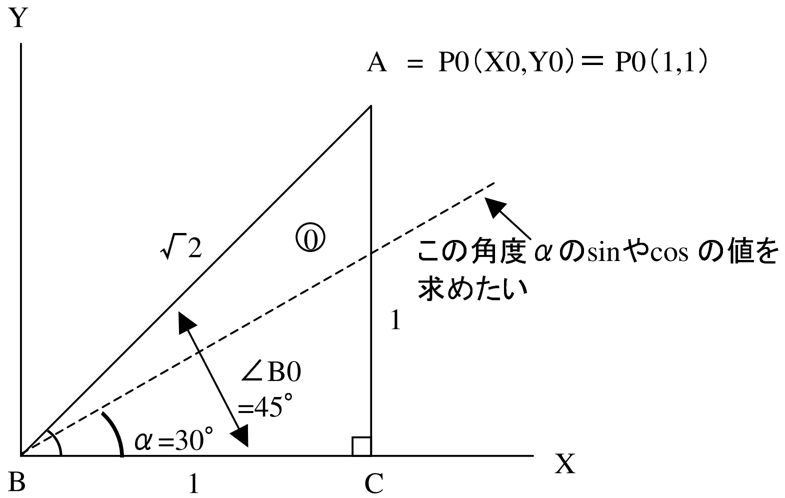
右図のような三角形をたくさん用意する。 直角をはさむ1辺の長さが常に1で、別の１辺が 1, 1/2, 1/4 ・・・と半分になってゆく三角形である。
各三角形に0,1,・・・n と番号を付けておく。
頂点 B の角度は、 ∠ABC = arctan(b/a)である。 この値はあらかじめ計算してテーブルにしておく。
定数 k = 1.646760258 をセットしておく。
n 番目の三角形の∠ABC を∠Bn とも記述する。
また、各辺の長さの関係は c^2 = a^2 + b^2 (ピタゴラスの定理)
ここでは単位を° で計算しているが、ラジアンでも可能である。
その場合はテーブルの内容もラジアンで設定する。



5.２ 計算の実行

与えられた角度をαとする。 例としてα=30° とする。

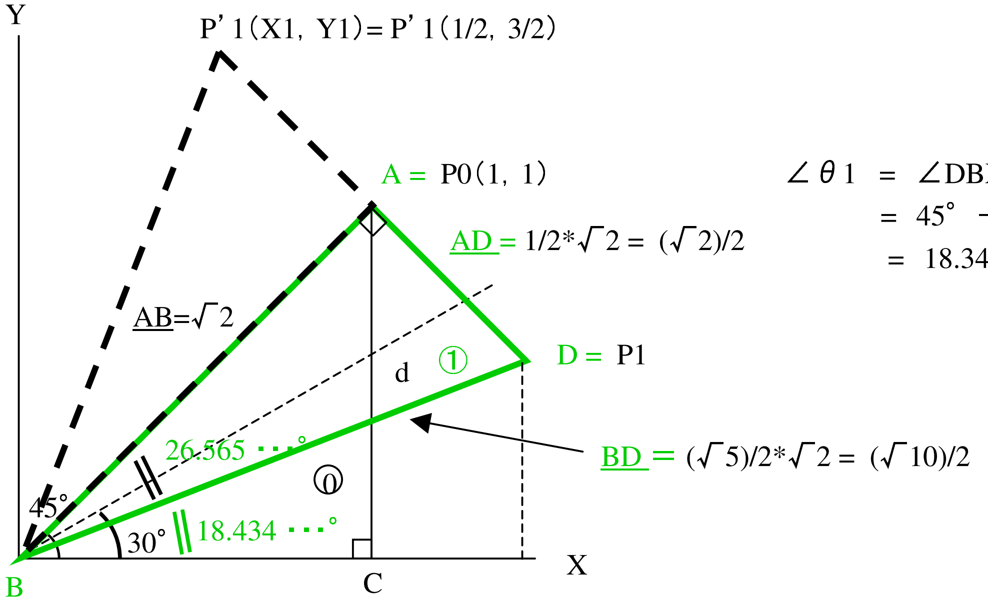
1） 最初は⓪の三角形を下図のように配置する。
∠ABX = ∠B0 = 45° (X軸と配置の三角形の斜辺との角)
この三角形の(αに対応する)角度を θ0 として
θ0 = ∠B0 = 45° からスタートする。



表記の注意 (私が勝手に決めた)
1) ここでは記述上数字を見やすくするために通常不要な()を使用する場合がある。
例 $\frac{\sqrt{5}}{2}$ => (√5)/2 ・・・ √(5/2)と紛らわしくならないように( )を付けた。
2) 辺を表す時に通常文字の上に線を引くが、ここでは下線とする。 辺AB => AB で表示する。
辺ABの長さも AB と表示する場合がある。
辺の表記の順序がまちまちの事がある。 AB を BAと表記することがあるが同じ辺である。
3) Pi(Xi, Yi)は i番目の点 Pi においてXi はそのX座標, Yi はそのY座標。
Pi(X)、Pi(Y) はそれぞれi番目の点 Pi においてのX座標, またY座標とする。
4) 説明によって、付随する図の頂点や辺の記号が違う場合がある。 説明に従うこと。
(ある図では頂点A だった点が、別の図では頂点D だったりすることがある)

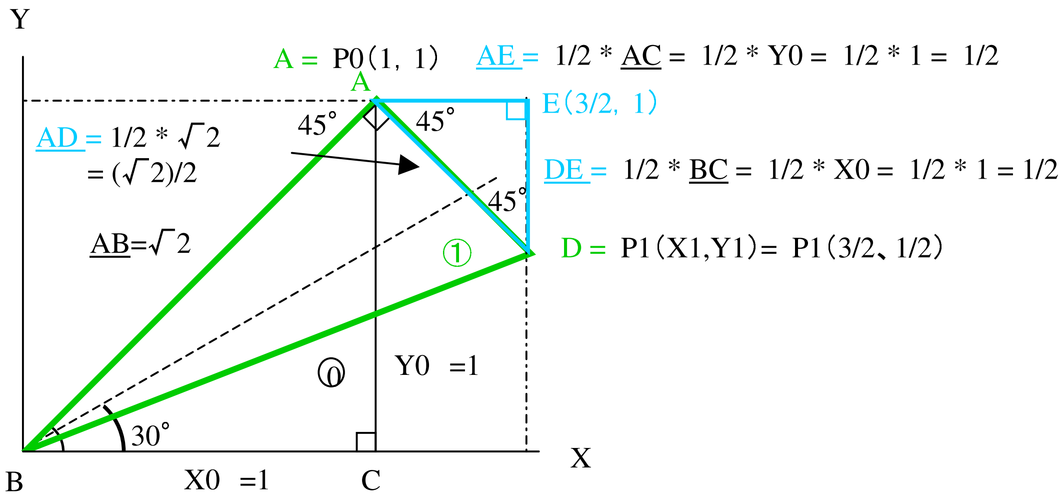
2）⓪は最初に配置した。 次に①の三角形を配置する。
αとθ0（X軸と与えられた三角形の斜辺との角 = ∠ABX = ∠B0 ）を比較して（ α － θn ）
α＜θ0 では①の三角形を図の方向に、直前の三角形の斜辺と重なり合うように相似拡大して
配置する。（ここでは勝手にマイナス方向と呼ぶことにする）
α≧θ0 ならば①の三角形を図の点線の方向に配置。（プラス方向と呼ぶ）
こうすると①の三角形の辺の長さは、①の辺<u>AB</u>を⓪の三角形斜辺に相似拡大したので
各辺の長さを√2倍にすれば良い。

各辺の長さおよび必要な角度の算出



∠θ1 = ∠DBX = ∠θ0－∠B1
= 45° － 26.565･･･°
= 18.343･･･°

頂点D の座標を求める



P1（X1,Y1）の座標を求める。

a) △ADEが△ABCと相似であり、また<u>AD</u> と <u>AB</u> は最初から1：2に設定されているので、比例関係で辺の長さで△ADEは△ABCの1/2倍となる。 従って△ADEの辺の長さは図の値になる。

b) P1（X1, Y1）の座標は図のようにP0（1, 1）から計算できる。

P1（X1,Y1）= P1（X0 + <u>AE</u>, Y0 －<u>DE</u> ）= P1（X0 + 1/2*Y0, Y0 －1/2*X0 ）
= P1（1 + 1/2, 1－1/2 ） = P1（3/2, 1/2）

仮に三角形をプラス方向に配置した場合（点線）は、 P’1はP1と頂点Aに対して対称の位置なので

P’1（X1, Y1）= P’1（X0－AE, Y0＋DE）= P’1（X0 －1/2*Y0, Y0 + 1/2*X0）

=P’1（1－1/2, 1＋1/2）= P’1（1/2, 3/2） となる。

また、∠θ’1 ＝ 45°＋ 26.565･･･° ＝ 71.565･･･° である。

3） ②の三角形を配置する

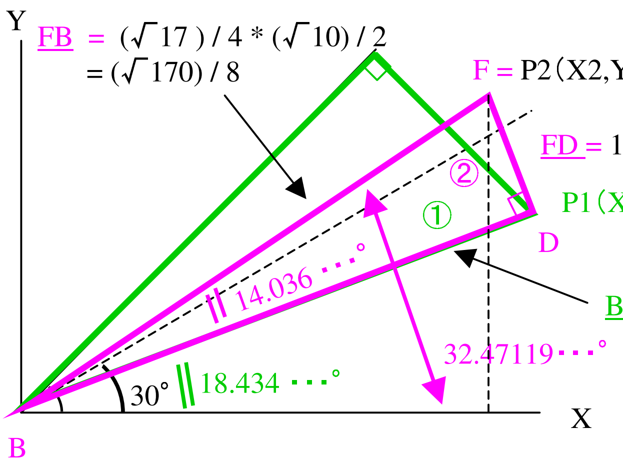
次に②の三角形を配置する。 αとθ1 を比較して（α－θ1）

α≧θ1 なので②の三角形を図の方向に（プラスの方向）、且つ①の三角形の斜辺と重なるように相似拡大して置く。 その時、各頂点の位置や辺の長さは図のようになる。

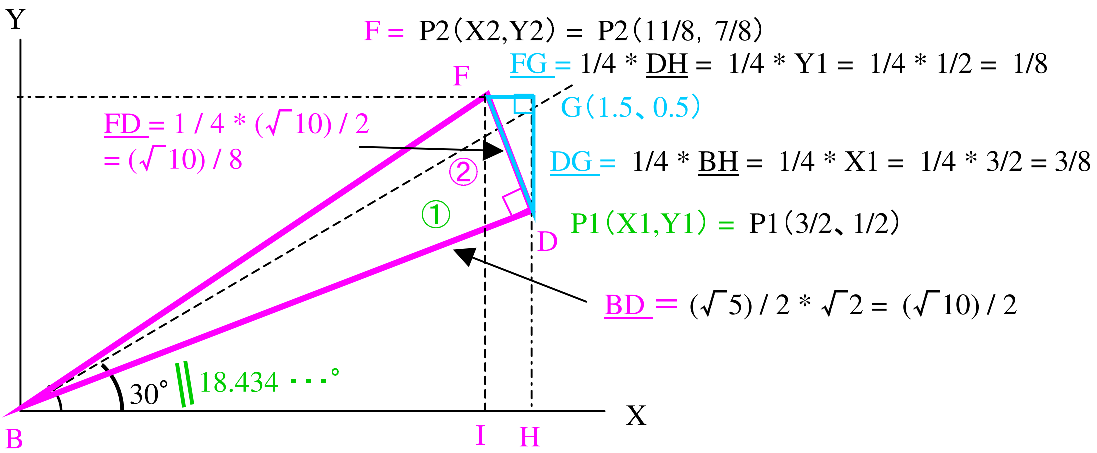
①の斜辺が(√10)/2なので、②の3辺も(√10)/2倍される。

各辺の長さおよび必要な角度の算出

∠θ2 ＝ 18.343･･･° ＋ 14.036･･･°
＝ 32.471･･･°



頂点F の座標を求める



P2（X2, Y2）の座標を求める。

a) △FDGが△DBH と相似であり、またFD とBD は1:4に設定されているので、比例関係で辺の長さで△FDGは△DBHの1/4倍になる。 従って△FDGの辺の長さは図の値になる。

b) P2（X2, Y2）の座標は図のようにP1（X1, Y1）から計算できる。

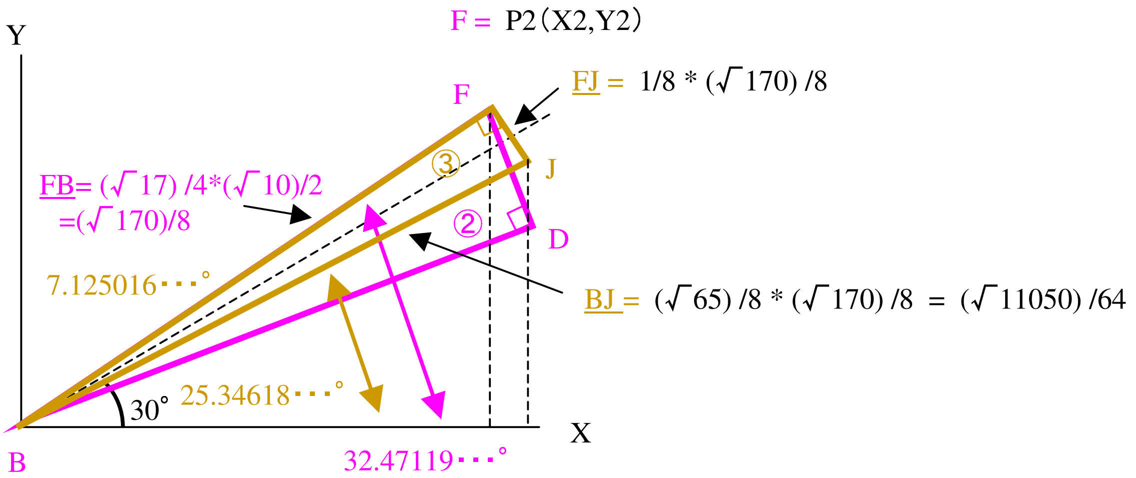
P2（X2,Y2）= P2（X1－FG, Y1＋DG）= P2（X1－1/4*Y1, Y1＋1/4*X1）

= P2（3/2－1/8, 1/2 + 3/8）= P2（11/8, 7/8）

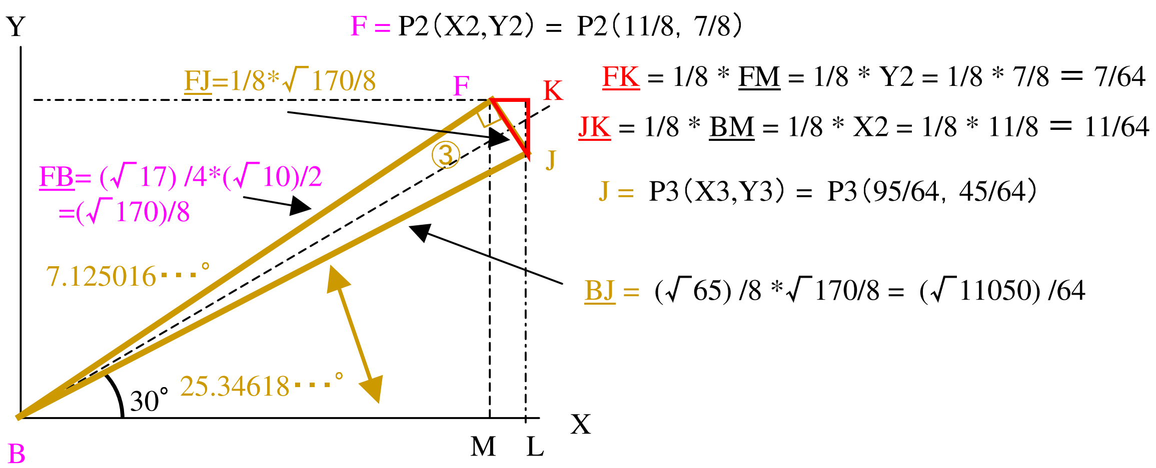
また、∠θ2 ＝ ∠FBX ＝ 18.434･･･° ＋14.036･･･° ＝ 32.47119･･･° である。

4） ③の三角形を使用する
③の三角形を使用する。　αと$\theta 1$ を比較して（引き算して）
A＞$\theta 2$ なので③の三角形を図の方向に、且つ②の三角形の斜辺と重なり合うように相似拡大して置く。
その時、各頂点の位置や辺の長さは図のようになる。
②の斜辺が$(\sqrt{170})/8$なので、③の3辺も$(\sqrt{170})/8$倍される。

各辺の長さおよび必要な角度の算出



頂点J の座標を求める



P3（X3, Y3）の座標を求める。

a) △FJKが△FBM と相似であり、またFJとFB は最初に定義して1:8に設定してあるので、辺の長さで△FJKは△FBMの1/8倍になる。

b) 従って△FJKの辺の長さは図の値になる。
P3（X3,Y3)=の座標は図のようにP2（X2, Y2）から計算できる。

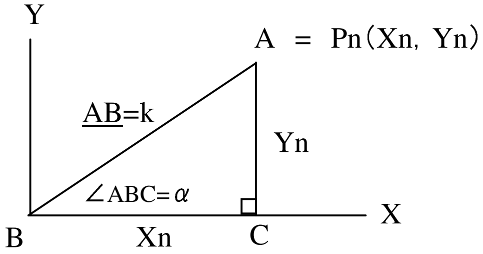
$$P3(X3,Y3) = P3(X2+\underline{FK},\ Y2-\underline{JK}) = P3(X2+1/8 * Y2,\ Y1-1/8 * X2)$$
$$= P3(11/8 + 7/64,\ 7/8 - 11/64) = P3(95/64,\ 45/64)$$

ここで $\angle\theta 3 = \angle JBX = 32.47119\cdots^\circ - 7.125016\cdots^\circ = 25.34618\cdots^\circ$

この作業を n まで繰り返す。 この例では n = 17 である。

三角形を重ねる度に角度αとの誤差は少なくなってゆく場合と、誤差が前回よりも大きくなる場合があるが、何度も繰り返せばだんだん誤差が少なくなるように収束する。
ただし計算の精度も必要で、計算の桁数が少ないと多く繰り返しても意味がない事になる。

5）必要な回数を反復
必要な回数（今回 n=17 ）を反復すると
$\angle\theta n \fallingdotseq \alpha$ となって Pn（Xn,Yn）のXnがX軸上の、YnがY軸と平行の三角形ができる。
斜辺は反復を繰り返す毎に定数 k に近くなる。
この k は三角形を重ねてゆくたびに計算してゆくと、一定値に収束する。
その値は 1.646760258 である。



k = 1.646760258 として、
求める三角比は
$\sin\alpha = Yn / k$
$\cos\alpha = Xn / k$
$\tan\alpha = Yn / Xn$
となる。

５．３　三角比を求めるCORDIC のまとめ

手順を整理してCORDIC の方法をまとめると次のようになる。
１）反復回数、arctan のテーブル、および定数 k をあらかじめ計算して用意しておく。
２）最初に 0 の三角形のパラメータをセットする。
P0(X0, Y0) = P0（1, 1） => X0 = 1, Y0 = 1　　$\angle B0 = 45°$　　$\theta 0 = \angle B0$
３）i = 1〜n まで、θ i の値によってa) かb) のどちらかの処理を行う。
a) $\alpha - \theta$ (i-1) < 0 の時は
Pi（Xi, Yi）= Pi (X(i-1) + 1/2^(i) * Y(i-1) , Y (i-1) − 1/2^(i) * X(i-1))
θ i = θ (i-1) −∠Bi
b) $\alpha - \theta$ (i-1) ≧ 0 の時は
Pi（Xi, Yi）= Pi (X(i-1) − 1/2^(i) * Y(i-1) , Y (i-1) + 1/2^(i) * X(i-1))
θ i = θ (i-1) ＋∠Bi

※ 注1

４）三角比は Pn（Xn, Yn）から次のように求まる。
$\sin\alpha = Yn / k$
$\cos\alpha = Xn / k$
$\tan\alpha = Yn / Xn$
ただしKでの割り算は、1 / k の値を定数としてそれとの掛け算としてもよい。
（割り算は時間や手間が多くかかるため）

※ 注1 Basicのプログラムとは書き方が違う点に注意。 右辺の(i-1) がBasicの書式では i となる。

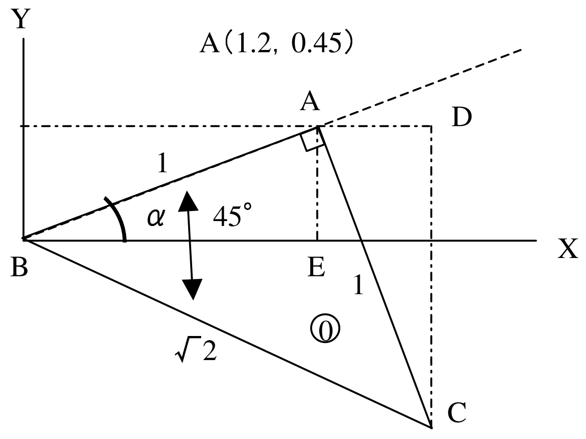
この方法で Excel と BASCOM のBasic でプログラムを組んでみた。
付２．、付３. にその結果を示す。

６． CORDICを使用して arctan（Y/X）と√（X^2＋Y^2）を求める。

三角比を求めたのと同じような方法で arctan（Y/X）と√（X^2＋Y^2）を求めてみる。
基本的な手順は似ているが、初期設定や最終結果の導き方など少しの違いはある。
１）スタートポイントはA（X, Y）である。 求める角度はαは0° から始める。
２） Yi ≧ 0 の時は三角形を図の方向に配置してマイナス方向と名付ける。
　また、Yi ＜ 0 の時はマイナス方向と名付ける。
３）三角形を配置するたびに、頂点 Pi の値は更新され、角度∠Bi をαに累算してゆく。
　最終的に i = n まで配置したときに 頂点 Pn（Xn, Yn ）およびαが得られる。
　求める値は次のようになる。
　　arctan（Y / X ） = α
　　√（X^2＋Y^2） = Xn / k　　k=1.646760258

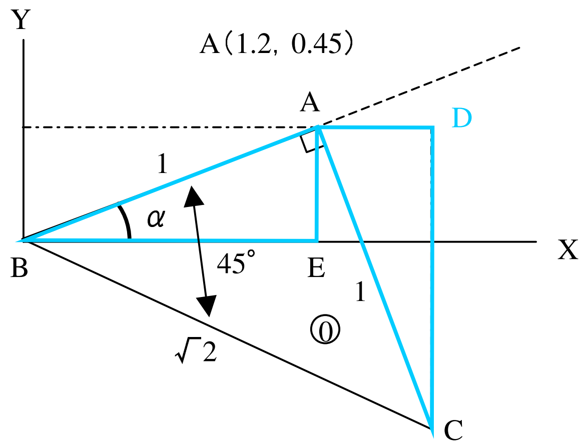
６．１ 計算例

例として X = 1.2, Y = 0.45 として arctan（Y / X ） および √（X^2 ＋Y^2 ）の
この2つの値を求める。



はじめのX,Yは頂点A（1.2, 0.45）である。
A（Y ） = 0.45 ≧ 0 なので
0 の三角形を図のように配置する。
（マイナス方向 ）
頂点Aが直角の頂点になるように配置。
AB = √（X^2 ＋Y^2 ） => 1 としておく。

三角形を手続きに従ってn回配置した時の
斜辺の長さはABのk倍になっているので、
ABを求めるためにはkで割ればよい。



C= P0（1.65, －0.75）

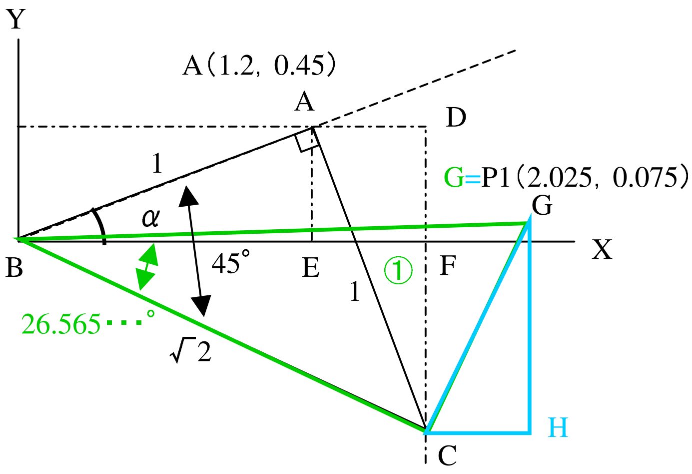
頂点C の座標を以下のようにして求める。
△ACD と △ABEは相似。
AB = AC なので AD = AE = 0.45
BE = CD なので DC = BE = 1.2

また A（Y ） ≧ 0 なので
C = P0（X0, Y0） = P0（BE＋AD, AE－DC）
　= P1（1.2＋0.45, 0.45－1.2） = P1（1.65, －0.75）

角度についてはYの座標の正負によって角度の
加算/減算を決める。
ここでは A（Y） = 0.45 ≧0 なので
符号は＋である。

θ0 = 0 + 45° = 45°

①の三角形を配置する。(△GBC)
△ABCの頂点C のY座標は　P0（Y ）< 0　なので、プラス方向に配置する。
大きさも図のように比例拡張する。



C= P0（1.65, −0.75）

△CGH　と　△CBFは相似。
CG ＝ 1/2* CB なので、辺の長さで
　△CGHは△CBF の1/2
　CH ＝ 1/2 * CF = 1/2 * （−0.75)= −0.375
同様にHG ＝ 1/2 * FB なので
　HG ＝ 1/2* FB = 1/2 * 1.65= 0.825

ここで P0（Y0)< 0 なので
G= P1（X1, Y1） = P1（X0−CH, Y0＋HG)
　= P1（1.65−（−0.375), −0.75＋0.825)
　= P1（2.025, 0.075）
P0（Y） ＝ −0.75 ＜ 0 なので
符号は−である。

θ1 = 45° −26.565･･･° ＝ 18.434･･･°

以下三角形 n まで繰り返し、その後次の計算を行えば値が求まる。
　　arctan（Y/X） ＝ θn　　√（X^2＋Y^2） ＝ Xn / k

6．2　arctan（Y/X)と√（X^2＋Y^2)の値を求めるCORDIC のまとめ
　手順を整理してCORDIC の方法をまとめると次のようになる。
1）反復回数、arctan のテーブル、および定数 k をあらかじめ計算して用意しておく。
2）最初に イニシャルのパラメータをセットする。
　A(X, Y）　　θ0 ＝ 0
3）Pi（Xi, Yi) を順に求める。 i = 0〜n まで、Yi の値によってa) かb) のどちらかの処理を行う。
　a) Yi ＜ 0 の時は
　　Pi（Xi, Yi) = Pi (X(i-1) − 1/2^(i) * Y(i-1) , Y (i-1) ＋ 1/2^(i) * Xi(I-1))
　　θi ＝ θ(i-1) −∠Bi　　（ただしi=0 のときはθ(i-1) ＝0)
　b) Yi ≧ 0 の時は
　　Pi（Xi, Yi) = Pi (X(i-1) ＋ 1/2^(i) * Y(i-1) , Y (i-1) − 1/2^(i) * Xi(I-1))
　　θi ＝ θ (i-1) + ∠ Bi　　（ただしi=0 のときはθ(i-1) ＝0)

※ 注1

4）結果は次のように求まる。
　　arctan（Y/X） ＝ θn　　√（X^2＋Y^2） ＝ Xn / k
　ただしKでの割り算は、1/k の値を定数としてそれとの掛け算としてもよい。
　（割り算は時間や手間が多くかかるため）

※ 注1 Basicのプログラムとは書き方が違う点に注意。 右辺の(i-1) がBasicの書式では i となる。

この方法で Excel と BASCOM のBasic でプログラムを組んでみた。 付4. 、付5. にその結果を示す。

７．　感想など

１）サルには近づけたでしょうか。

２）“そんなの数式があれば、内容はわからなくてもExcel やプログラムは作れるだろうに” と言われれば全くその通りだが、それでは不満なので今回ちょっとしつこく考えてみた。
それにしても、もし読まれる方がいらっしゃったら辟易したのではないか。

３）最近はシステムなどが高度化、複雑化しているので、機能をモジュール化してブラックボックスとし、さらに上位の機能を構成するという方法をとらないと、とてもシステムの実現が困難というのも事実である。

４）従って一度位は見てなるほどねと言って、あとは忘れてしまうのが良いかもしれない。（それがこれだ）

５）うまくできてる。 よくこういう方法を考えるものだなあというのが実感である。 優れた方法だから普及しているのだろう。

６）アルゴリズム自体は柔軟なので、三角比などという個別な用途ではなくてもっと包括的に理解ができると応用ができるが、そこまでとても頭が付いてゆかない。
LOGや平方根などもこの方法で計算できるので、検索してみたが前記のHP のように丁寧に説明されているものは見つからなかった。

７）BASCOM　AVR　を使ってBasicで三角比や√を求めるCORDIC も作ってみた。
浮動小数点演算なので、プログラム領域も思ったよりも消費した。 1.6kbyte強程度であった。
このモジュールを固定小数点を使用したアセンブラで書けば小型高速なものができそう。

８）数値をBinaryで扱うと、*2 は数値の1bit 左シフト、/2 は数値の1bit 右シフトで実行できる。
CORDIC はそれを意識してアルゴリズムの中に単に/　ではなく、/2 が使用してあるところがすごい。
それによってハードウェアの小型化や実行時間の減少が期待できるのだが、ExcelやBasicで実行させてもその特徴は生かせない。 今回はアルゴリズムを理解したいということでこのようにした。

９） sin, cos で最終結果を算出するのに “kで割る”処理があるが、これはあらかじめ1/kを計算しておいて掛ければ良い。 割り算よりも掛け算のほうが楽である。
さらに1/kは単なる比例係数と考えれば、処理によってはこの掛け算も不要にすることができる。

１０）BASCOM などでいろいろな値をCORDIC で計算させ、精度を理論値などと比較したり実行時間を調べたりするのも比較的容易だが、今回は実施しない。

１１）それにしても、なんと理解するということがたいへんなこと。 ここまで解るのに何週もかかった。
こんな程度でジタバタしていては、日暮れて道遠しなのだが・・・・
それでいて数カ月経過したら、忘れてしまって文章を読みなおすなどということが発生する。
そのためにもあとで解るような文章にしておかなきゃ。

１２）昔の事を見事に忘れている。
うちの子供の使ったはずの中高学校の数学教科書が、こんな事もあろうと取ってあった（先見の明！）ので参考にした。 大いに役立った。 ところでこの教科書とてもきれいで書き込みなど皆無であった。
どうなっているのだろう。

１３）今回の原稿はひどく時間がかかった。
チェックするたびにミスや見にくい所などが出てくる。
注意はしたつもりだが、まだ不具合があったらご容赦。
ただ、アルゴリズムという点ではあっていると思う。

付1．　2つの三角形の相似の理由

各図で2つの三角形の相似が出てくるが、その相似の理由は以下の通り。

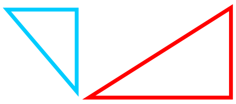
2つの三角形が相似である条件は

1)三辺の比が等しい

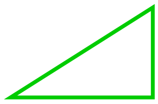
2) 2組の辺の比が等しく、そのはさむ角が等しい

3) 2組の角が、それぞれ等しい

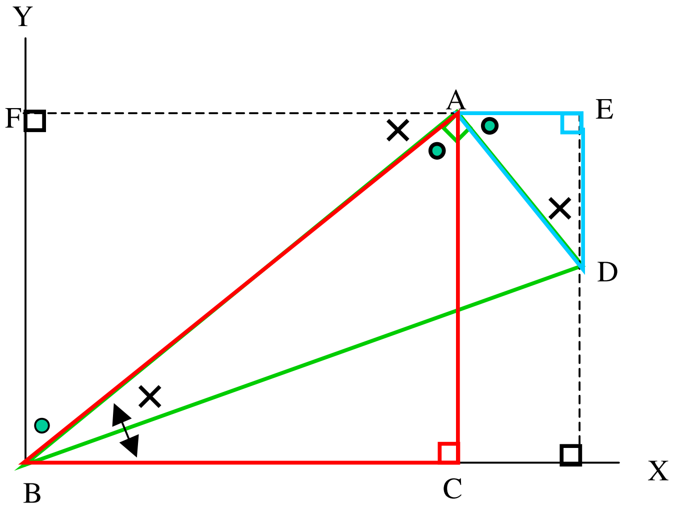
下の図では〇、×の角は、れぞれ等しく、3)の条件で相似となる。



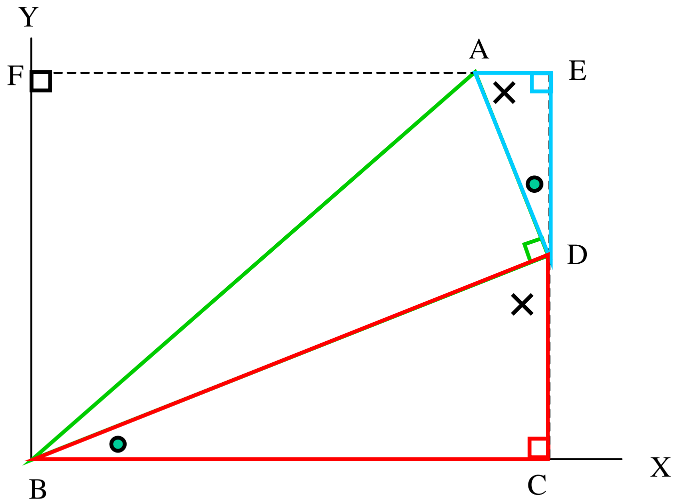
2つは相似である。



配置された三角形



配置された三角形の直角の頂点が
X軸と平行の線上にあるとき



配置された三角形の直角の頂点が
Y軸と平行の線上にあるとき

付2.　Excel でCORDICのアルゴリズムを使って三角比を求める

CORDICによって三角比求める

角度　（0°≦α≦90°）
α（°）　30　　k=　1.64676

	計算値	
sin α	0.500006	=Y17/k
cos α	0.866022	=X17/k
tan α	0.57736	=Y17/X17

⑦

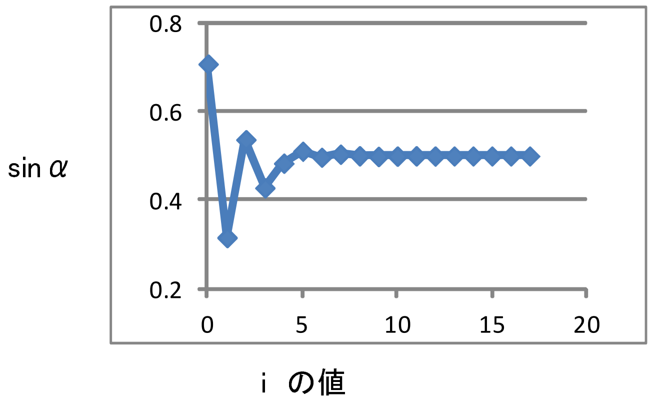
	理論値	誤差
sin α	0.5	6.01E-06
cos α	0.866025	-3.47E-06
tan α	0.57735	9.26E-06

計算過程　　スタート時に値を設定

i	Xi	Yi	θi（°）	sign	Bi（°）	斜辺	sin α
0	1	1	45	1	45	1.41421356	0.707107
1	1.5	0.5	18.43495	-1	26.56505	1.58113883	0.316228
2	1.375	0.875	32.47119	1	14.03624	1.6298006	0.536875
3	1.484375	0.703125	25.34618	-1	7.125016	1.64248407	0.428086
4	1.44043	0.795898	28.92251	1	3.576334	1.64568892	0.483626
5	1.415558	0.840912	30.71242	1	1.789911	1.64649228	0.510729
6	1.428697	0.818794	29.81725	-1	0.895174	1.64669325	0.497235
7	1.4223	0.829955	30.26486	1	0.447614	1.64674351	0.503998
8	1.425542	0.8244	30.04105	-1	0.223811	1.64675607	0.50062
9	1.427152	0.821615	29.92915	-1	0.111906	1.64675921	0.498929
10	1.42635	0.823009	29.9851	1	0.055953	1.64676	0.499775
11	1.425948	0.823706	30.01307	1	0.027976	1.64676019	0.500198
12	1.426149	0.823357	29.99909	-1	0.013988	1.64676024	0.499986
13	1.426049	0.823531	30.00608	1	0.006994	1.64676025	0.500092
14	1.426099	0.823444	30.00258	-1	0.003497	1.64676026	0.500039
15	1.426124	0.823401	30.00083	-1	0.001749	1.64676026	0.500013
16	1.426137	0.823379	29.99996	-1	0.000874	1.64676026	0.499999
17	1.426131	0.82339	30.0004	1	0.000437	1.64676026	0.500006
①	②	③	④	⑤	⑥	⑧	⑨

① i回め
② Pi　のX座標
③ Pi　のY座標
④ 三角形の頂角の累算。 4．2項における∠ABCの累算。
⑤ ④−⑥の符号。 これによって三角形の配置する方向を決める。
⑥ 三角形の頂角。 4．2項における∠ABC。
⑦ 理論値はExcel の関数の値を表示
⑨ i まで反復した時の斜辺の長さ
⑨ i まで反復した時のsin αの値

i　の増加に従ってsin30°　が
収束する様子



付3.　BASCOM　AVR　で実行してみる（三角比）

```
Rem  CORDIC

$regfile = "m328pdef.dat"
$crystal = 1000000

Config Base = 0

Dim Atantbl(18) As Single
Dim Xi As Single
Dim Yi As Single
Dim Angle As Single
Dim Sina As Single
Dim Cosa As Single
Dim Tana As Single
Dim Wka As Single
Dim Wkw As Single
Dim Wkx As Single
Dim Wky As Single
Dim Wkz As Single
Dim Count As Byte
Dim Num As Byte
Dim Kkk As Single

Init:
  Restore Tt:
  For Count = 0 To 17
    Read Atantbl(count)
  Next Count

  Xi = 1
  Yi = 1
  Wka = Atantbl(0)
  Kkk = 1.646760258
  Wkz = 1

  Angle = 30        ← 値を求めたい角度（°　）

  Start:
  For Count = 1 To 17
    Wkz = Wkz / 2
    Wkx = Xi * Wkz          ← Xi * 1/2^i
    Wky = Yi * Wkz          ← Yi * 1/2^i
    Wkw = Angle - Wka       ← α − θi

    If Wkw < 0 Then
      Xi = Xi + Wky
      Yi = Yi - Wkx
      Wka = Wka - Atantbl(count)
    Else
      Xi = Xi - Wky
      Yi = Yi + Wkx
      Wka = Wka + Atantbl(count)
    End If
  Next Count

  Sina = Yi / Kkk   ┐
  Cosa = Xi / Kkk   │ 結果
  Tana = Yi / Xi    ┘

Stop
End
```

右上に続く。

```
Tt:
  Data 45! , 26.565051177! , 14.036243468! , 7.125016348!
  Data 3.576334375! , 1.789910608! , 0.895173710! , 0.447614171!
  Data 0.223810500! , 0.111905677! , 0.055952919! , 0.027976453!
  Data 0.013988227! , 0.006994114! , 0.003497057! , 0.001748528!
  Data 0.000874264! , 0.000437132!
```

BASCOM　で作ってみたCORDIC である。
シミュレータで動作させてみた。 ただしこのプログラムでは、
a)　データを浮動小数点で扱っている。
b)　BASCOM 特有の“演算は二項に限る“という条件が
　あるので、少し冗長なところがある。
c)　最適化を行っているわけではない。 なるべく解りやすい
　様に書いてある。

計算値はちょっと見たところ、少数点以下5桁位は合っている。
tan はそれよりも少し悪いくらい。

付4.　Excel でCORDICのアルゴリズムを使って　arctan(Y/X)、√（X^2＋Y^2）を求める

CORDICによって　arctan(Y/X)、√（X^2+Y^2）　を求める

X,Y　　　（－90°　≦　arctan(Y/X)　＜　90°　）

X	1.2
Y	0.45

k=　1.64676

	計算値	
atan Y/X	20.55644	=θ17
√（ ）	1.281601	=X17/k

⑦

	理論値	誤差
atan α	20.55605	3.98E-04
√（ ）	1.281601	5.08E-11

計算過程　　　　スタート時に値を設定

i	Xi	Yi	θi(°)	sign	Bi(°)
	1.2	0.45	0	1	
0	1.65	-0.75	45	-1	45
1	2.025	0.075	18.43495	1	26.56505
2	2.04375	-0.43125	32.47119	-1	14.03624
3	2.097656	-0.17578	25.34618	-1	7.125016
4	2.108643	-0.04468	21.76984	-1	3.576334
5	2.110039	0.021217	19.97993	1	1.789911
6	2.11037	-0.01175	20.8751	-1	0.895174
7	2.110462	0.004735	20.42749	1	0.447614
8	2.110481	-0.00351	20.6513	-1	0.223811
9	2.110487	0.000613	20.5394	1	0.111906
10	2.110488	-0.00145	20.59535	-1	0.055953
11	2.110489	-0.00042	20.56737	-1	0.027976
12	2.110489	9.8E-05	20.55338	1	0.013988
13	2.110489	-0.00016	20.56038	-1	0.006994
14	2.110489	-3.1E-05	20.55688	-1	0.003497
15	2.110489	3.36E-05	20.55513	1	0.001749
16	2.110489	1.43E-06	20.55601	1	0.000874
17	2.110489	-1.5E-05	20.55644	---	0.000437
①	②	③	④	⑤	⑥

① i回め
② Pi　のX座標
③ Pi　のY座標
④ 三角形の頂角の累算。 4．2項におけθiの累算
⑤ Yi　の符号。 これによって三角形の配置する方向を決める
⑥ 三角形の頂角。 4．2項における∠ABC
⑦ 理論値はExcel の関数の値を表示

付5.　BASCOM　AVR　で実行してみる(arctan（Y/X)、√（X^2＋Y^2)）

```
Rem  CORDIC

$regfile = "m328pdef.dat"
$crystal = 1000000

Config Base = 0

Dim Atantbl(18) As Single
Dim Xi As Single
Dim Yi As Single
Dim Angle As Single
Dim Root As Single
Dim Atana As Single
Dim Wkx As Single
Dim Wky As Single
Dim Wkz As Single
Dim Count As Byte
Dim Kkk As Single

Init:
  Restore Tt:
  For Count = 0 To 17
    Read Atantbl(count)
  Next Count

  Xi = 1.2                  X,  Y
  Yi = 0.45
  Kkk = 1.646760258
  Wkz = 2                   スタート値

  Angle = 0                 スタート値

  Start:
  For Count = 0 To 17

    Wkz = Wkz / 2
    Wkx = Xi * Wkz          Xi * 1/2^i
    Wky = Yi * Wkz          Yi * 1/2^i

    If Yi >= 0 Then         Yi で判断
      Xi = Xi + Wky
      Yi = Yi - Wkx
      Angle = Angle + Atantbl(count)
    Else
      Xi = Xi - Wky
      Yi = Yi + Wkx
      Angle = Angle - Atantbl(count)
    End If
  Next Count

  Atana = Angle             求める角度(°)
  Root = Xi / Kkk           √（X^2＋Y^2)


Stop
End

Tt:
  Data 45! , 26.565051177! , 14.036243468! , 7.125016348!
  Data 3.576334375! , 1.789910608! , 0.895173710! , 0.447614171!
  Data 0.223810500! , 0.111905677! , 0.055952919! , 0.027976453!
  Data 0.013988227! , 0.006994114! , 0.003497057! , 0.001748528!
  Data 0.000874264! , 0.000437132!
```

BASCOM　で作ってみたCORDIC である。

シミュレータで動作させてみた。 ただしこのプログラムでは、

a) データを浮動小数点で扱っている。

b) BASCOM 特有の“演算は二項に限る“という条件があるので、少し冗長なところがある。

c) 最適化を行っているわけではない。 なるべく解りやすい様に書いてある。

以上