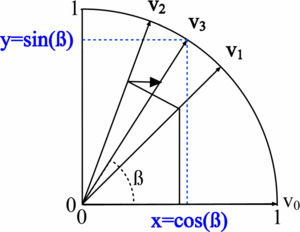
**[CORDIC算法计算正余弦](https://www.cnblogs.com/huangqiwei/p/4177067.html)**

网上有很多关于CORDIC算法的资料，看了之后觉得还是wikipedia讲述的更加清晰，特此总结+转载

<http://en.wikipedia.org/wiki/CORDIC>

**算法思想**



CORDIC算法是一种对目标值进行逼近的迭代算法，且迭代次数越多精度越高。迭代过程中仅仅需要除2运算和加减运算，因此特别适合硬件方式实现。在单位圆中，圆上角β点的x坐标和y坐标分别对应β的cos和sin值，因此，求角β的正弦值的CORDICn次迭代过程如下：

1、以(1,0)为初始点，向靠近β的方向旋转arctan(1)=45°得到点v1

2、v1向靠近β的方向旋转角度arctan(1/2)得到点v2

3、点vi向靠近β的方向旋转角度arctanc(1/(2^i))得到vi+1

4、当i+1=n时，停止，vn的坐标便是所求正余弦值

**坐标旋转**

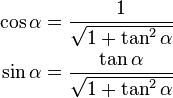
每一次cordic迭代都是以此旋转计算，通过让https://images0.cnblogs.com/blog/694915/201412/212119232805820.png乘以旋转矩阵https://images0.cnblogs.com/blog/694915/201412/212119335938068.png来实现，如下式：

https://images0.cnblogs.com/blog/694915/201412/212119449684360.png

旋转矩阵通过下式来计算：

https://images0.cnblogs.com/blog/694915/201412/212121056083458.png

而cos和sin函数可用下式化为tan



因此https://images0.cnblogs.com/blog/694915/201412/212121201718474.png可化为：

https://images0.cnblogs.com/blog/694915/201412/212122346245364.png

如果让tanhttps://images0.cnblogs.com/blog/694915/201412/212123098121580.png取值https://images0.cnblogs.com/blog/694915/201412/212122591085916.png，那么vi坐标与矩阵的乘法运算均可用移位来实现，此时旋转公式为：

https://images0.cnblogs.com/blog/694915/201412/212123362499503.png

其中，

https://images0.cnblogs.com/blog/694915/201412/212124073744895.png

Ki可以在迭代完成后单独计算，最终只需乘以Kn即可：

https://images0.cnblogs.com/blog/694915/201412/212124261873680.png

随着n的增加，Kn趋于稳定

https://images0.cnblogs.com/blog/694915/201412/212124499379090.png

https://images0.cnblogs.com/blog/694915/201412/212125207806268.png取±1，它决定着是顺时针旋转还是逆时针旋转。

β则根据下式进行逼近，每次迭代都应朝着向β靠近的方向旋转。

https://images0.cnblogs.com/blog/694915/201412/212125569211327.png

https://images0.cnblogs.com/blog/694915/201412/212123098121580.png的值可以通过查表法来获得。