

## 一种基于 CORDIC 算法的高精度反正切求解\*

仲雅莉<sup>1</sup>, 吴俊辉<sup>2</sup>, 刘炫<sup>2</sup>, 高萍<sup>1,3</sup>, 段晓辉<sup>1,4</sup>(1. 国家超级计算无锡中心, 江苏 无锡 214072; 2. 江南大学, 江苏 无锡 214122;  
3. 山东大学, 山东 济南 250100; 4. 清华大学, 北京 100084)

**摘要:** 传统的 CORDIC(坐标旋转计算机)算法进行高精度反正切求解时存在迭代次数多、收敛速度慢、资源消耗大等问题, 提出一种改进的高精度 CORDIC 算法。该方法利用传统的 CORDIC 算法迭代数次后得到正弦信息, 并利用角度和正弦值近似的原理, 对迭代后的结果进行误差补偿, 有效提高了相同迭代步数下的计算精度。实验数据表明: 32 位改进的 CORDIC 算法, 保证绝对误差小于  $5 \times 10^{-9}$  情况下, 查找表资源消耗降低了 64.8%, 触发器资源消耗降低了 35.3%, 输出时延降低了 53.3%。在分子动力学应用场景下, 触发器资源消耗可降低 63.2%, 输出时延可降低 60%。改进的 CORDIC 算法在资源消耗、输出时延方面优于传统 CORDIC 算法, 适用于高精度计算的场景。

**关键词:** 反正切求解; 迭代收敛; 误差补偿

中图分类号: TP391

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.212393

中文引用格式: 仲雅莉, 吴俊辉, 刘炫, 等. 一种基于 CORDIC 算法的高精度反正切求解[J]. 电子技术应用, 2022, 48(1): 12-17.

英文引用格式: Zhong Yali, Wu Junhui, Liu Xuan, et al. A high precision arctangent solution based on CORDIC algorithm[J]. Application of Electronic Technique, 2022, 48(1): 12-17.

## A high precision arctangent solution based on CORDIC algorithm

Zhong Yali<sup>1</sup>, Wu Junhui<sup>2</sup>, Liu Xuan<sup>2</sup>, Gao Ping<sup>1,3</sup>, Duan Xiaohui<sup>1,4</sup>(1. National Supercomputing Center in Wuxi, Wuxi 214072, China; 2. Jiangnan University, Wuxi 214122, China;  
3. Shandong University, Jinan 250100, China; 4. Tsinghua University, Beijing 100084, China)

**Abstract:** The traditional CORDIC (Coordinate Rotation Computer) algorithm has many iterations, slow convergence speed, and large resource consumption for high-precision arctangent. An improved high-precision CORDIC algorithm is proposed. This method uses the traditional CORDIC algorithm to obtain the sine information after several iterations, and uses the sine value to compensate the error of iteration results, which effectively improves the calculation accuracy. Experimental data shows that the 32 bit improved CORDIC algorithm ensures that the absolute error is less than  $5 \times 10^{-9}$ , the resource consumption of the lookup table is reduced by 64.8%, the resource consumption of the flip-flop is reduced by 35.3%, and the output delay is reduced by 53.3%. In molecular dynamics application scenarios, flip-flop resource consumption can be reduced by 63.2%, and output delay can be reduced by 60%. The improved CORDIC algorithm is superior to the traditional CORDIC algorithm in terms of resource consumption and output delay, and is suitable for high-precision computing applications.

**Key words:** CORDIC arctangent solution; iterative convergence; error compensation

## 0 引言

坐标旋转计算机(Coordinated Rotation Digital Computer, CORDIC)算法只有移位和加减运算, 便于在 FPGA 等硬件平台上实现复杂的三角函数、双曲函数、指数函数和复数求模等计算, 广泛应用于数字鉴相、数字上下变频、波形产生、快速傅里叶变换等方面<sup>[1-4]</sup>。

针对 CORDIC 算法中由于迭代次数多、输出延时大和精度较低等问题, 国内外很多学者进行了研究和改

进。文献[5]提出了基于自适应旋转角度的 CORDIC 算法, 该算法虽然减少了迭代次数, 但每次迭代都需要额外判断, 增加了实现难度。文献[6]提出了将查找表和传统 CORDIC 算法相融合, 通过查找表将角度值细化, 通过数学量化分析, 根据细化后的较小角度补码, 直接按位值进行角度单向旋转, 该算法随着对精度要求的提高, 查找表存储空间大大增加, 硬件资源消耗巨大。文献[7]提出了一种基于最佳一致逼近方法的幅度与相位补偿算法, 第一步是利用传统的 CORDIC 算法迭代数次后得到向量信息, 第二步是采用最佳逼近法进行多项式补偿,

\* 基金项目: 国家重点研发计划(2019YFA0709400); 国家自然科学基金青年基金(62102114)

该算法虽然提高了精度,但逼近算法需要存储多项式系数,硬件资源消耗较大。文献[8]提出一种基于CORDIC算法的反正切函数计算的改进算法,该算法对累加器中因截尾而产生的误差做了算法改进,仅增加了运算速度,精度并未明显提高。文献[9]提出了一种基于查找表的改进的CORDIC算法,该方法通过缩减有效数据位宽、合并迭代等手段节省了剩余角度 $Z$ 的计算量,该算法把迭代次数进行拆分,用了14次迭代和查找表的结合,硬件资源消耗较多,性能提升却不足。

在分子动力学模拟中<sup>[10-11]</sup>,成键力的计算引入了三角函数。常见的一种情况是从三维坐标计算角度,通常需要根据 $\sin\theta$ 和 $\cos\theta$ 的值求解反正切,该应用场景下对反正切的精度要求较高,传统的CORDIC算法很难满足<sup>[12-13]</sup>。

本文在文献[7]的基础上,简化逼近算法,提出了一种改进算法,测得不采用函数拟合也可得到高精度的剩余相位跟踪,从而有效提高了CORDIC求反正切算法的精度,降低了延时,减少了电路面积。

## 1 CORDIC 算法基本原理

在 $X-Y$ 坐标平面上将向量 $(x_0, y_0)$ 逆向旋转角度 $\theta$ 得到向量 $(x_1, y_1)$ ,如图1所示。两向量间坐标变换关系为:

$$\begin{cases} x_1 = x_0 \cos\theta - y_0 \sin\theta \\ y_1 = y_0 \cos\theta + x_0 \sin\theta \end{cases} \quad (1)$$

两边同时除以 $\cos\theta$ ,可得到伪旋转方程,即:

$$\begin{cases} x_1' = \frac{x_1}{\cos\theta} = x_0 - y_0 \tan\theta \\ y_1' = \frac{y_1}{\cos\theta} = y_0 + x_0 \tan\theta \end{cases} \quad (2)$$

除以 $\cos\theta$ 后,旋转角度还是正确的,但向量的模长发生了变化。

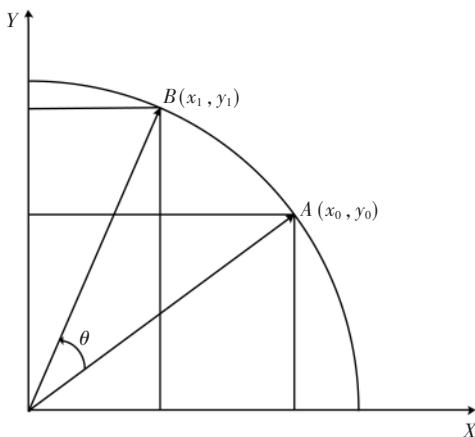


图1 向量旋转示意图

CORDIC求反正切的本质就是从坐标 $(x_0, y_0)$ 开始旋转,每次旋转固定角度 $\theta_i$ ,旋转方向为 $d_i = \text{sign}(y(i))$ ,旋转的目标是纵坐标 $y_i$ 趋近于0,进行 $N$ 步迭代,则有反正切角 $\theta = \sum_{i=0}^{N-1} d_i \theta_i$ 。为了能够简化运算部件,CORDIC算法

常采用每一小步的旋转角度 $\theta_i$ 满足 $\tan\theta_i = 2^{-i}$ ,从而可以通过简单移位来完成乘法运算,因此原始的算法可以简化为迭代移位相加算法,迭代方程为:

$$\begin{cases} x_{i+1} = x_i + 2^{-i} d_i y_i \\ y_{i+1} = y_i - 2^{-i} d_i x_i \\ z_{i+1} = z_i + d_i \theta_i \end{cases} \quad (3)$$

如果保持向量的模长,则可以通过加入旋转补偿因子 $K$ 实现。当 $N$ 确定时,经历 $N$ 次迭代后的 $K$ 是一个常数:

$$K = \prod_{i=0}^{N-1} \cos\theta_i = \prod_{i=0}^{N-1} \sqrt{\frac{1}{1+2^{-2i}}} \quad (4)$$

该补偿因子常在求三角函数 $\sin$ 和 $\cos$ 时使用。

## 2 基于CORDIC求反正切的算法改进

对CORDIC算法进行分析可以看出, $\theta_i = \arctan 2^{-i}$ 。而进行 $i$ 次迭代后,残余的角度的绝对值即是求反正切值时的误差。定义 $\varepsilon_i$ 为第 $i$ 次旋转后的误差,则:

$$\varepsilon_i = |\varepsilon_{i-1} - \theta_i| \quad (5)$$

当 $\varepsilon_{i-1}$ 极小时, $\varepsilon_i$ 取极大值 $\theta_i$ 。又有:

$$\lim_{\theta \rightarrow 0} \arctan\theta = \theta \quad (6)$$

所以CORDIC求反正切算法在残余角度较小时的误差以每次1 bit的速度收敛。对于定点数而言,即每次迭代使得精度收敛1 bit。以在 $[0, \frac{\pi}{2})$ 进行均匀取样仿真了不同迭代次数下的CORDIC最大误差如表1所示,对于相位精度要求极高的场景,如果要达到 $10^{-9}$ 数量级,需要迭代28次,其带来的硬件资源消耗和时延都是不可承受的。

表1 不同迭代次数下CORDIC算法的最大误差

迭代次数	$\theta_i$	最大误差
10	$1.95 \times 10^{-3}$	$1.93 \times 10^{-3}$
11	$9.76 \times 10^{-4}$	$9.5 \times 10^{-4}$
12	$4.88 \times 10^{-4}$	$4.7 \times 10^{-4}$
28	$7.45 \times 10^{-9}$	$7.44 \times 10^{-9}$
29	$3.72 \times 10^{-9}$	$3.71 \times 10^{-9}$
30	$1.86 \times 10^{-9}$	$1.86 \times 10^{-9}$

### 2.1 使用 $\sin\varepsilon$ 逼近 $\varepsilon$

当CORDIC经过第 $N$ 次旋转之后,设其估计的角度残余误差为 $\varepsilon$ ,显然:

$$\varepsilon \leq \arctan 2^{-(N-1)} < 2^{-(N-1)} \quad (7)$$

即 $N$ 次迭代后的残余误差小于 $2^{-(N-1)}$ 。根据微积分中正弦函数的极限,有:

$$\lim_{\varepsilon \rightarrow 0} \varepsilon \sim \sin\varepsilon \quad (8)$$

所以使用 $\sin\varepsilon$ 来逼近 $\varepsilon$ 是一种有潜力降低误差的手段。

由于 $\sin\varepsilon$ 与 $\varepsilon$ 在 $\varepsilon$ 比较小时非常接近,一种可以考虑的方案是直接采用 $\sin\varepsilon$ 来代替 $\varepsilon$ 。该方案的误差可以通过泰勒展开进行估计, $\varepsilon - \sin\varepsilon$ 在 $\varepsilon=0$ 处展开可得:

$$\varepsilon - \sin\varepsilon = \varepsilon - (\varepsilon - \frac{\varepsilon^3}{6} + o(\varepsilon^3)) = \frac{\varepsilon^3}{6} + o(\varepsilon^3) \quad (9)$$

也就是说,  $\sin\varepsilon$  和  $\varepsilon$  的差值以三次方的速度缩小, 比直接CORDIC的收敛速度快得多。当  $\varepsilon$  接近  $2^{-9} \approx 1.95 \times 10^{-3}$  时, 从式(9)可以估计出  $\varepsilon - \sin\varepsilon \approx 1.24 \times 10^{-9}$ , 意味着用CORDIC算法迭代10次后, 反正切的误差约为  $1.95 \times 10^{-3}$ , 而使用  $\sin\varepsilon$  对反正切结果进行校正后, 误差降为约  $1.24 \times 10^{-9}$ 。

实验的结果可以与上述估计相吻合: 令  $\text{err} = \varepsilon - \sin\varepsilon$ , 画出曲线如图2所示, 可以看出  $\varepsilon$  越小,  $\varepsilon$  与  $\sin\varepsilon$  的差值越小。当  $\varepsilon < 2^{-9}$  时,  $\varepsilon$  与  $\sin\varepsilon$  的差值可达到  $10^{-9}$  数量级, 这也意味着当CORDIC迭代10次后, 残余的  $\varepsilon$  足够小, 可以用  $\sin\varepsilon$  的值代替  $\varepsilon$  的值来修正CORDIC的结果。

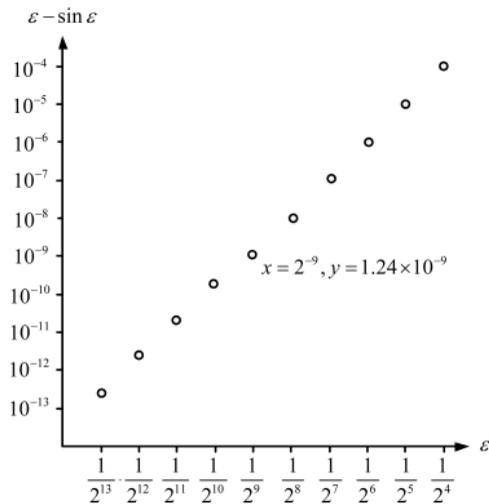


图2 区间  $[2^{-14}, 2^{-3}]$  上的  $\varepsilon - \sin\varepsilon$  (对数坐标轴)

根据上述对误差的论证, 在一定迭代步数后采用  $\sin\varepsilon$  代替  $\varepsilon$  是安全的, 如果输入的初始向量已经在单位圆上, 那么最终向量的纵坐标即为  $\sin\varepsilon$ 。所以本文对应修改了CORDIC求反正切的算法, 如算法1所示。

算法1 输入在  $x$  轴正半轴单位圆上的改进CORDIC算法

输入:

$x, y$ : 初始向量, 满足  $x^2 + y^2 = 1$  且  $x \geq 0$

$N, K_N$ : 迭代总次数和对应的缩放因子

输出:

$\theta$ : 估计的  $x, y$  与  $z$  轴的夹角

1: for  $i \in [0, N)$  do

2:  $d_i \leftarrow \text{sign}(y)$

3:  $x_{i+1} \leftarrow x_i + 2^{-i} d_i y_i$

4:  $y_{i+1} \leftarrow y_i - 2^{-i} d_i x_i$

5:  $z_{i+1} \leftarrow z_i + d_i \arctan 2^{-i} > \arctan 2^{-i}$

6: end for

7:  $\theta \leftarrow z_N + K_N y_N > K_N y_N \approx \sin(\theta - z)$

8: return  $\theta$

在迭代步数  $N$  的选取上, 可以根据式(7)和式(9), 按照所需的误差  $t$  来对  $N$  进行选取:

$$\text{err} = \varepsilon - \sin\varepsilon \approx \frac{(2^{-(N-1)})^3}{6} < t \quad (10)$$

即:

$$N \geq \lceil -\frac{\log_2 6t}{3} \rceil + 1 \quad (11)$$

而对于普通的CORDIC算法, 第  $N$  步的误差约为  $2^{-(N-1)}$ , 则需要:

$$N \geq \lceil \log_2 t \rceil + 1 \quad (12)$$

从式(11)和式(12)可以看出, 在需求精度较高的情况下, 本文改进的CORDIC算法只需要大约三分之一的迭代步就能实现与原有的CORDIC相似的精度。

### 2.2 $\sin\varepsilon$ 的计算

根据CORDIC的原理可知, CORDIC的旋转过程中向量模长会发生变化, 需要对模长进行修正。设迭代初始向量为  $(x_0, y_0)$ , 经过  $N$  步旋转后, 根据式(3)得到  $(x_N, y_N)$ ,  $z_N$ , 式(4)可求得旋转修正因子  $K$  使得:

$$K \sqrt{x_N^2 + y_N^2} = \sqrt{x_0^2 + y_0^2} \quad (13)$$

当  $(x_0, y_0)$  在单位圆上时, 如图3所示,  $l=1$ ,  $y_N$  即为  $\sin\varepsilon$ , 这就可以解决输入的坐标在单位圆上的问题。

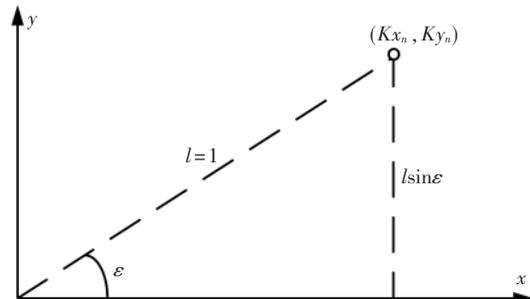


图3  $(x_0, y_0)$  在单位圆上时的余角

若输入的坐标  $(x_0, y_0)$  不在单位圆上, 则需要根据式(14)进行模值归一化:

$$s = \sqrt{x_0^2 + y_0^2} = \frac{1}{\sqrt{p}} \quad (14)$$

则:

$$Ks = \sqrt{x_N^2 + y_N^2} = s \sqrt{x_0^2 + y_0^2} = 1 \quad (15)$$

从而使得:

$$\begin{cases} \sin\varepsilon = Ks y_N \\ \theta \approx z_N + Ks y_N \end{cases} \quad (16)$$

如式(14)所示, 模值归一化引入了平方根倒数, 直接求解资源消耗大。可以用牛顿迭代法求得, 迭代公式如式(17)所示, 其中  $p$  是式(14)中初始坐标  $(x_0, y_0)$  的模值平方,  $q$  是对其平方根倒数的估计。

$$q_{n+1} = q_n (1.5 - \frac{pq_n^2}{2}) \quad (17)$$

在定点数实现牛顿迭代法时,由于没有尾码对齐的性质<sup>[14]</sup>,因此很难通过简单的变换就获得一个良好的初值<sup>[15]</sup>。本文考虑将  $p$  映射到  $[1, 4)$  之间,如式(18)所示:

$$\begin{cases} p=2^{2k}p', k \in \mathbb{Z} \wedge p' \in [1, 4) \\ \frac{1}{\sqrt{p}} = \frac{1}{\sqrt{2^{2k}p'}} = \frac{1}{2^k \sqrt{p'}} \end{cases} \quad (18)$$

其中  $2^{2k}$  和  $2^k$  均可通过移位实现。当  $p'$  落在  $[1, 4)$  之间时,可以分段通过线性变换获得一个较为良好的初值:

$$q_0 = \begin{cases} \frac{163}{128} - \frac{1196}{4096}p', & 1 \leq p' < 2 \\ \frac{115}{128} - \frac{422}{4096}p', & 2 \leq p' < 4 \end{cases} \quad (19)$$

式(19)中初值与结果的误差基本可以控制在  $2 \times 10^{-2}$  以内,如图 4 所示。这样只需两步牛顿迭代就可以获得较为准确的  $1/\sqrt{p}$ 。由于 CORDIC 旋转与模值无依赖性,模值归一化流程仅在最后一步将  $s$  与  $K_N$  相乘时会导致额外的一次乘法的时延。

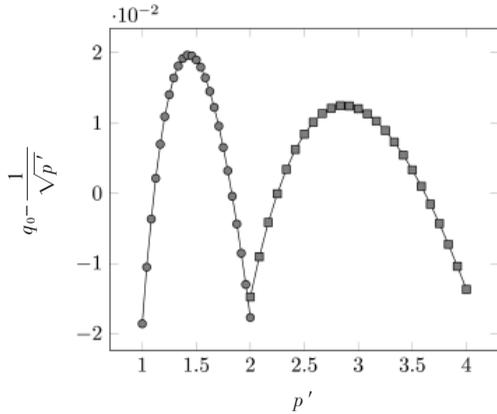


图 4 初值  $q_0$  与  $\frac{1}{\sqrt{p'}}$  之差

### 2.3 整体架构

改进的 CORDIC 求反正切算法整体架构如图 5 所示。其中,角度预处理是指把坐标转换到第一或第四象限,预处理后的坐标先按照传统 CORDIC 进行  $N$  级迭代,再根据式(16)进行纵坐标修正和误差补偿,最后结果映射还原到原始象限的角度。算法 2 展示了该架构对应的伪代码。

算法 2 整体的 CORDIC 算法架构

输入:

- $x, y$ : 输入数据的坐标;
- $u: (x, y)$  是否在单位圆上;
- $N, K_N$ : 迭代总次数和对应缩放因子;
- $T$ : 预处理查找表,  $T_i = \arctan 2^{-i}$

输出:

$\theta$ : 输出的角

- 1: function INV-SQRT( $p$ )
- 2:  $k \leftarrow \lfloor \log_4 p \rfloor$

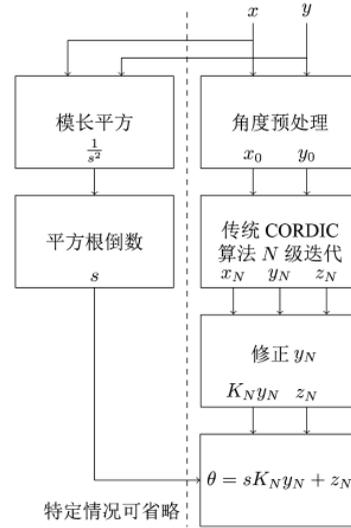


图 5 改进的 CORDIC 算法的整体流程

```

3:   $p' \leftarrow 2^{2k} p$ 
4:   $q_0 \leftarrow \begin{cases} \frac{163}{128} - \frac{1196}{4096} p', & p' \in [1, 2) \\ \frac{115}{128} - \frac{422}{4096} p', & p' \in [2, 4) \end{cases}$  >乘, 移位
5:   $q_1 \leftarrow q_0 \cdot \frac{3-p'q_0^2}{2}$ 
6:   $q_2 \leftarrow q_1 \cdot \frac{3-p'q_1^2}{2}$ 
7:  return  $2^{-k} q_2$ 
8:  end function
9:  function MAP( $x, y$ )>映射到第一/四象限
10:  if  $x < 0$  then
11:    return  $(-x, -y, \text{sign}(y) \cdot \pi)$ 
12:  else
13:    return  $(x, y, 0)$ 
14:  end if
15:  end function
16:  function ROTATE( $x_0, y_0, z_0$ )
17:  for  $i \in [0, N)$  do
18:     $d_i \leftarrow \text{sign}(y)$ 
19:     $x_{i+1} \leftarrow x_i + 2^{-i} d_i y_i$ 
20:     $y_{i+1} \leftarrow y_i - 2^{-i} d_i x_i$ 
21:     $z_{i+1} \leftarrow z_i + d_i T_i$ 
22:  end for
23:  return  $K_N y_N, z_N$ 
24:  end function
25:   $s \leftarrow \begin{cases} 1, & u \\ \text{INV-SQRT}(x^2+y^2), & \neg u \end{cases}$ 
26:   $x_0, y_0, z_0 \leftarrow \text{MAP}(x, y)$ 
27:   $x_N, y_N, z_N \leftarrow \text{ROTATE}(x_0, y_0, z_0)$ 
28:  return  $s y_N + z_N$ 
    
```

2.4 改进算法在分子动力学模拟中的应用

在分子动力学模拟中,常见的一种情况是从三维坐标计算角度。以分子动力学中的二面角  $\phi$  为例,很多实现就选择了使用  $\sin\phi$  和  $\cos\phi$  作为二维反正切函数的输入,从而获得值域为  $[-\pi, \pi)$  的结果。此时由于坐标显然在单位圆上,模长归一化的过程就可以省略。

图 6 展示了求解平面  $ijk$  和  $jkl$  间二面角  $\phi$  所需的向量,  $r$  是两个原子间的向量,  $n$  是平面的法向量。其中:

$$\begin{cases} n_{ijk} = r_{ij} \times r_{jk} \\ n_{jkl} = r_{jk} \times r_{kl} \end{cases} \quad (20)$$

$n'$  是  $n_{ijk}$  和  $n_{jkl}$  所确定的平面上与  $n_{ijk}$  垂直的向量:

$$n' = n_{ijk} \times r_{jk} \quad (21)$$

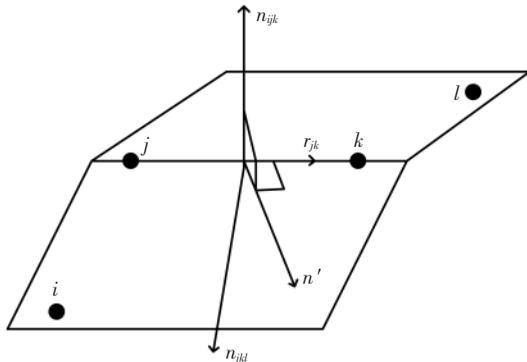


图 6 二面角求解时的向量

其二维视图如图 7 所示。这样,  $\phi$  的计算方式为:

$$\begin{cases} \cos\phi = \frac{n_{ijk} \times n_{jkl}}{|n_{ijk}| |n_{jkl}|} \\ \sin\phi = \cos(\phi - \frac{\pi}{2}) = \frac{n' \cdot n_{jkl}}{|n'| |n_{jkl}|} \\ \phi = \arctan(\cos\phi, \sin\phi) \end{cases} \quad (22)$$

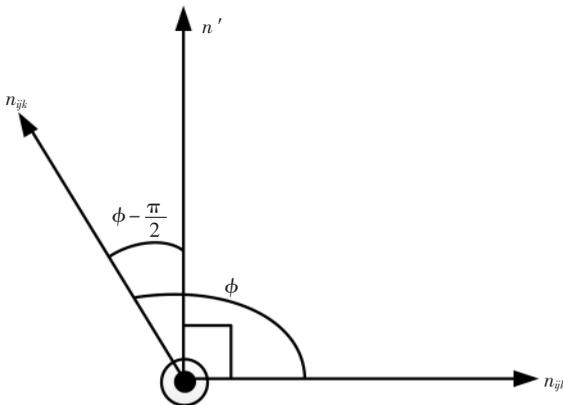


图 7 图 6 中几个向量的二维视图

也就是说此处的模长归一化已经隐含在了二面角计算的流程中,从而可以省略反正切函数的模长归一化过程。

类似地,在 CHARMM 键角的计算中:

$$V = K(\theta - \theta_0)^2 \quad (23)$$

这里  $V$  是键角的势能,  $K$  和  $\theta_0$  是经验参数,  $\theta$  是两个键形成键角的角度,对于以  $i$  原子为中心,  $j, k$  构成的键角,一般以反余弦函数求解:

$$\theta = \arccos \frac{r_{ij} \cdot r_{ik}}{|r_{ij}| |r_{ik}|} \quad (24)$$

计算原子受力  $F$  时需要根据链式法则求解  $V$  对三个原子坐标  $x$  的梯度,以  $j$  原子为例:

$$F_j = -\nabla_{x_j} V = -\frac{dV}{d\theta} \cdot \frac{d\theta}{d\cos\theta} \cdot \nabla_{x_j} \left( \frac{r_{ij} \cdot r_{ik}}{|r_{ij}| |r_{ik}|} \right) \quad (25)$$

其中  $\frac{d\theta}{d\cos\theta}$  可以利用反余弦函数的导数将  $\cos\theta$  代入,即:

$$\frac{d\theta}{d\cos\theta} = \frac{d\arccos(\cos\theta)}{d\cos\theta} = -\frac{1}{\sqrt{1-\cos^2\theta}} = -\frac{1}{\sin\theta} \quad (26)$$

也就是说,  $\cos\theta$  本来就需要计算,通过对计算流程进行修改后即可避免反正切函数进行模长归一化的过程。

综上所述,虽然在必要时可以利用牛顿迭代法实现模长归一化的过程,但是在分子动力学等求解具体计算几何问题的情境下,基本不需要额外的模长归一化实现。此时,改进的 CORIDC 算法可以更好地发挥其优势。

3 算法仿真和分析

将传统 CORDIC 求反正切算法和改进的 CORDIC 求反正切算法分别用 MATLAB 建模仿真,并与理论值进行了误差分析。输入数据的横坐标和纵坐标分别以 0.001 为间隔遍历  $[-1, 1]$ , 统计相同迭代次数下所有数据的最大绝对误差,性能曲线如图 8 所示。可以看出改进后的 CORDIC 算法性能提升明显,在迭代 10 次的情况下,精度从  $10^{-3}$  数量级提升到  $10^{-9}$  数量级。

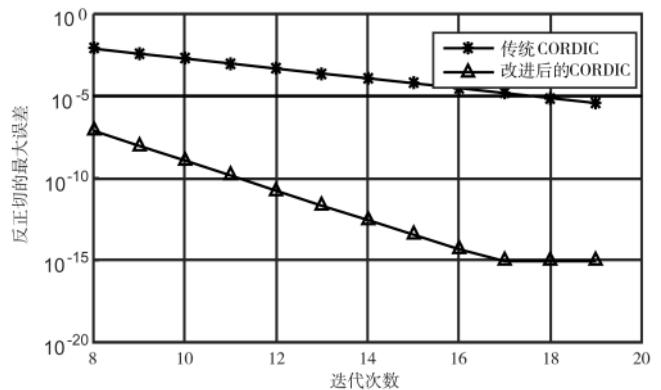


图 8 改进 CORDIC 算法与传统 CORDIC 算法的误差比较

本研究采用 Verilog HDL 语言进行设计,并在 Xilinx 公司的 Virtex UltraScale+ HBM VCU128 型号 FPGA 上完成了电路实现。整个模块的设计采用了 32 bit(3 bit 整数位+29 bit 小数位)的定点化输入输出。在保证绝对误差小于  $5 \times 10^{-9}$  的前提下,比较了硬件资源消耗如表 2 所示。

对于需要模值归一化的应用场景,归一化处理引入了 DSP 资源的消耗,但查找表(Look Up Table, LUT)资源

消耗降低了 64.8%，触发器(Flip-Flop, FF)资源消耗降低了 35.3%，输出时延降低了 53.3%。对于无模值归一化应用场景，LUT 资源消耗降低了 64.3%，FF 资源消耗降低了 61.6%，输出时延降低了 60%。改进的 CORDIC 算法在硬件消耗和时延方面有明显优势。

表 2 改进 CORDIC 算法  
与传统 CORDIC 算法的资源消耗对比

资源名称	传统 CORDIC 算法	改进 CORDIC 算法	
		单位圆	非单位圆
LUT	12 462	4 832	4 766
FF	2 910	1 197	2 015
DSP	0	4	61
IO	113	113	112
最大时延	30	12	14

#### 4 结论

针对传统 CORDIC 求反正切函数硬件资源消耗大，迭代次数多等问题，实现了一种改进的 CORDIC 算法计算反正切函数的解法。在保证与传统 CORDIC 算法精度数量级相同的情况下，减少了算法迭代次数，用旋转后的纵坐标补偿了相位，从而显著提高了算法运算速度。采用 32 bit 定点的硬件实现，在保证相同计算精度的前提下，改进 CORDIC 算法的硬件资源消耗和时延方面均有明显降低。算法仿真和 FPGA 实际测试结果表明，改进后的 CORDIC 算法适用于高速高精度场合。

该设计在包含模长归一化实现时已经比经典的 CORDIC 算法拥有更短的延时，在很多计算几何的具体问题中，可以通过对计算流程的适当重构从而使用单位向量作为输入，从而可以进一步降低该设计的开销，尤其适合分子动力学等三维空间中需要使用计算几何方法求解的问题。

#### 参考文献

- [1] AGGARWAL S, MEHER P K, KHARE K. Concept, design and implementation of reconfigurable CORDIC[J]. IEEE Transactions on Very Large Scale Integration(VLSI) Systems, 2016, 24(4): 1588-1592.
- [2] VALLS J, KUHLMANN M, PARHI K K. Evaluation of cordic algorithms for fpga design[J]. The Journal of VLSI Signal Processing, 2002, 32(3): 207-222.
- [3] JUANG T B, HSIAO S F, TSAI M Y. Para-CORDIC: Parallel CORDIC rotation algorithm[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2004, 51(8): 1515-1524.

- [4] PRASAD N, TRIPATHY M R, DAS A D, et al. Efficient VLSI implementation of CORDIC based direct digital synthesizer[J]. Advances in Intelligent Systems and Computing, 2015, 308(1): 97-603.
- [5] 李春娟, 李沙. 基于自适应旋转角度的 CORDIC 算法的设计与仿真[J]. 数字技术与应用, 2021, 39: 113-115.
- [6] 姚亚峰, 冯中秀, 陈朝. 直接旋转 CORDIC 算法及其高效实现[J]. 华中科技大学学报(自然科学版), 2016(44): 113-118.
- [7] 孙悦, 王传伟, 康龙飞, 等. 基于 CORDIC 的精确快速幅相解算方法[J]. 电子学报, 2018, 46(12): 164-170.
- [8] 刘小会, 许蕾, 刘海颖, 等. 基于 CORDIC 改进算法的反正切函数在 FPGA 中的实现[J]. 计算机技术与发展, 2013(11): 103-107.
- [9] 李雪, 徐洋洋, 邱雅倩, 等. 一种改进 CORDIC 算法的反正切计算[J]. 电视技术, 2019, 43(2): 19-23.
- [10] 郭禾, 龙珠, 王宇新, 等. 基于 FPGA 的分子动力学模拟系统设计[J]. 微电子学与计算机, 2009, 26(8): 246-248.
- [11] WANG H, PENG S, ZHU X, et al. A method to accelerate GROMACS in offload mode on Tianhe-2 supercomputer[C]// 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE, 2015: 781-784.
- [12] YANG C, GENG T, WANG T, et al. Molecular dynamics range-limited force evaluation optimized for FPGAs[C]// Proceedings of the 30th International Conference on Application-specific Systems, Architectures and Processors(ASAP). IEEE, 2019: 263-271.
- [13] Yang Chen, Geng Tong, Wang Tianqi, et al. Fully integrated on-FPGA molecular dynamics simulations[C]// Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2019: 1-31.
- [14] LOMONT C. Fast inverse square root[R]. West Lafayette: Purdue University, Department of Mathematics, 2013.
- [15] 周泉, 杨靓, 何卫强. 平方根倒数速算算法的精度优化[J]. 微电子学与计算机, 2019, 36(5): 64-69.

(收稿日期: 2021-11-29)

#### 作者简介:

仲雅莉(1987-), 女, 硕士, 工程师, 主要研究方向: 通信与信息系统。

吴俊辉(1997-), 男, 硕士研究生, 主要研究方向: 基于 FPGA 的算法加速。

段晓辉(1993-), 通信作者, 男, 博士后, 主要研究方向: 高性能分子动力学模拟, E-mail: sunrise\_duan@126.com。



扫码下载电子文档

## 版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所