CORDIC Cpp Implementation

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Based on the following site:

John Burkardt

CORDIC Approximation of Elementary Functions

http://people.sc.fsu.edu/~jburkardt/m_src/cordic/cordic.html

angle_shift (1)

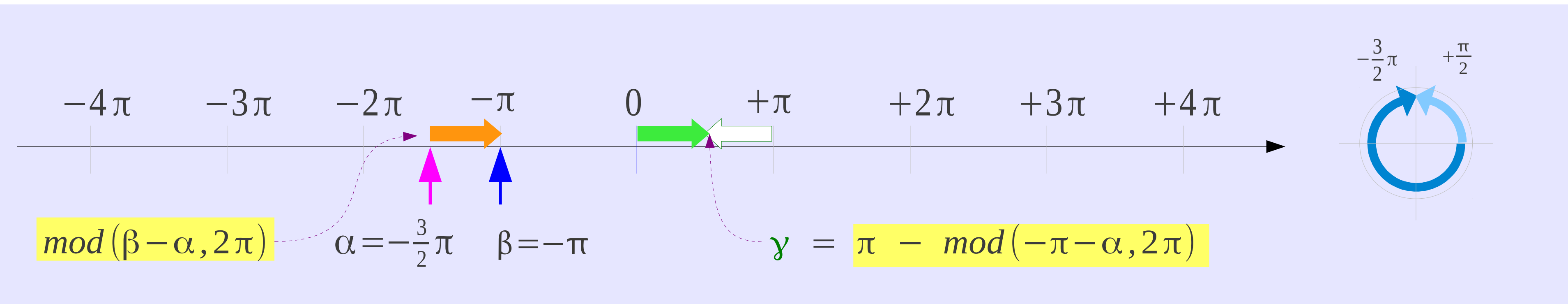
```
double angle_shift ( double alpha, double beta )
{
  double gamma;
  double pi = 3.141592653589793;

  if ( alpha < beta )  {
    gamma = beta - fmod ( beta - alpha, 2.0 * pi )
          + 2.0 * pi;
  }
  else  {
    gamma = beta + fmod ( alpha - beta, 2.0 * pi );
  }

  return gamma;
}

/*  Shift angle to interval [-pi,pi]. */
theta = angle_shift ( beta, -pi );
```

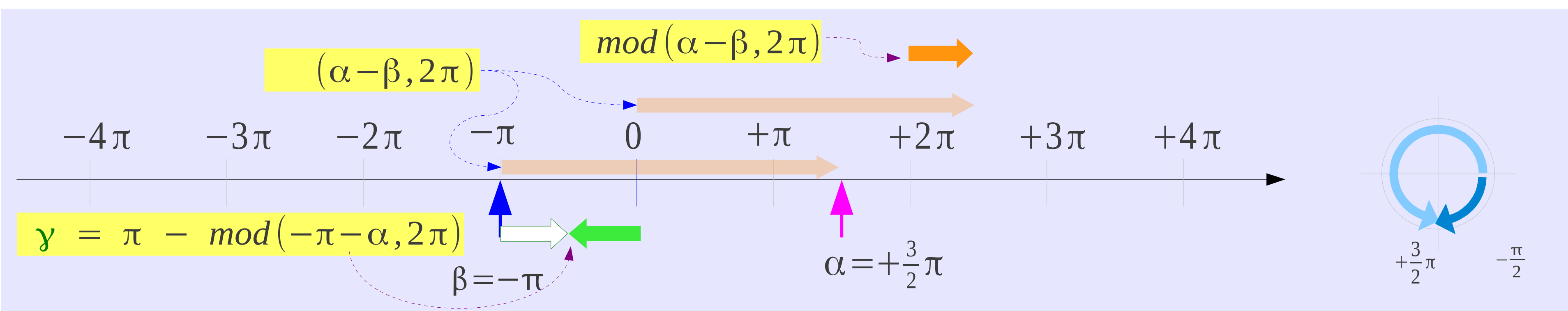
angle_shift (2)

$$\begin{cases} \gamma = \beta - mod(\beta-\alpha, 2\pi) + 2\pi & (\alpha < \beta) \\ \gamma = \beta + mod(\alpha-\beta, 2\pi) & (\alpha \geq \beta) \end{cases}$$

$$\begin{cases} \gamma = \pi - mod(-\pi-\alpha, 2\pi) & (\alpha < -\pi) \\ \gamma = -\pi + mod(\alpha+\pi, 2\pi) & (\alpha \geq -\pi) \end{cases}$$

$\beta = -\pi \Rightarrow -\pi < \gamma < +\pi$



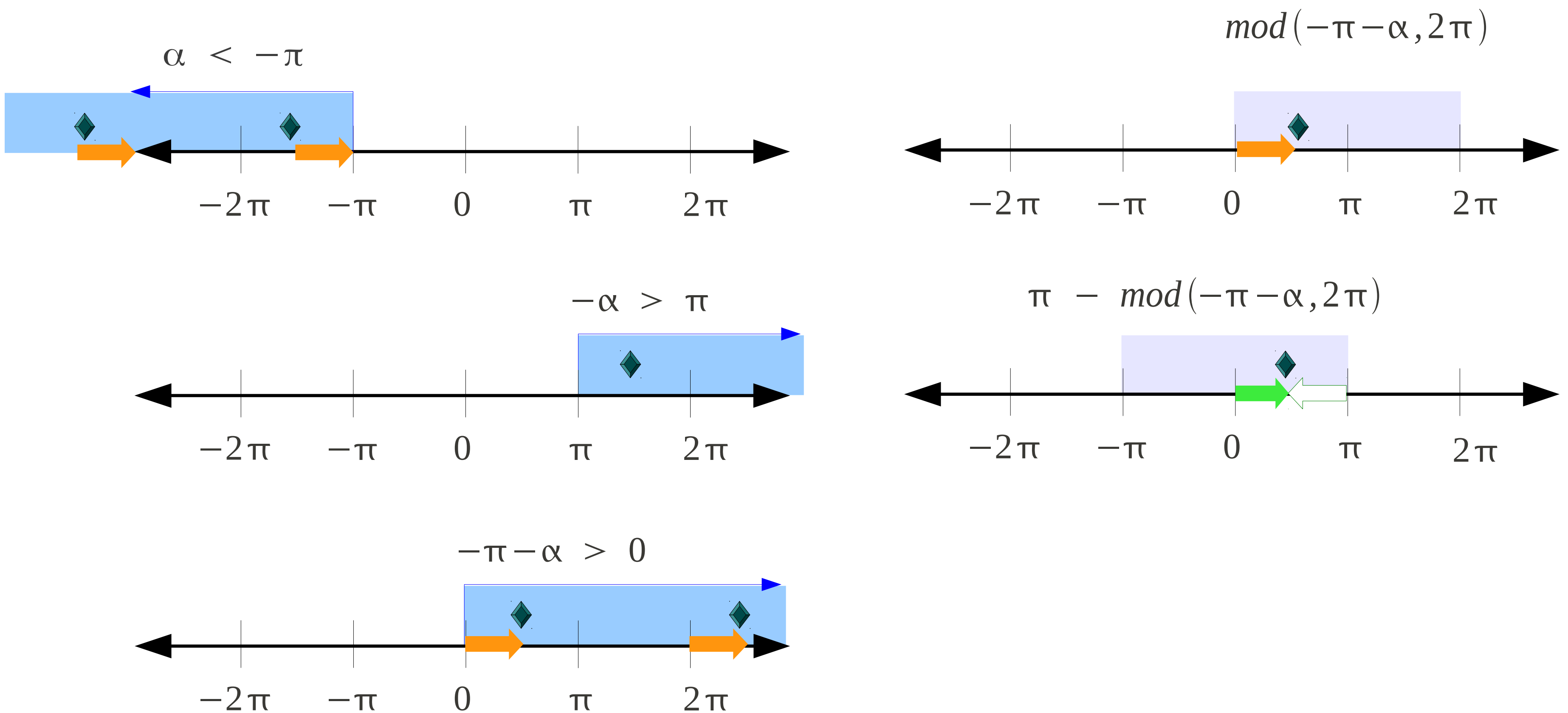


angle_shift (3)

$$\text{if} \quad \alpha < -\pi \qquad \gamma = \pi - mod(-\pi-\alpha, 2\pi)$$

$$\text{else} \qquad \gamma = -\pi + mod(\alpha+\pi, 2\pi)$$

$$\Rightarrow \quad -\pi < \gamma < +\pi$$

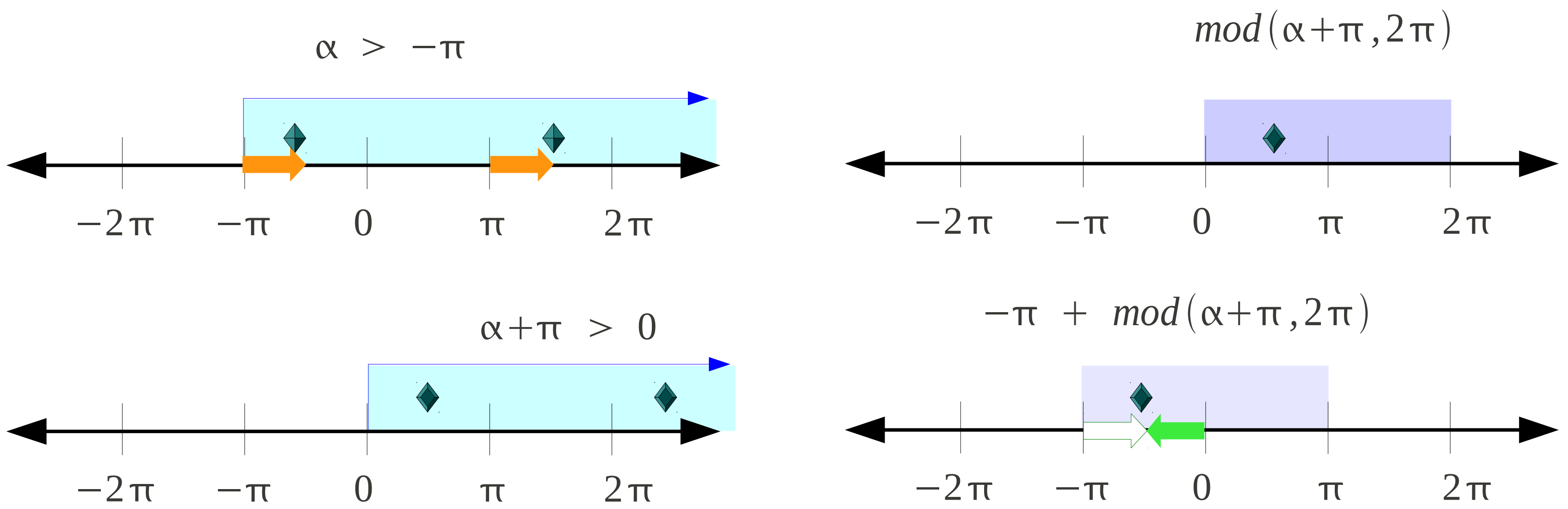


angle_shift (4)

if $\alpha < -\pi$ $\quad \gamma = \pi - mod(-\pi-\alpha, 2\pi)$

else $\quad \gamma = -\pi + mod(\alpha+\pi, 2\pi)$

$\Rightarrow \quad -\pi < \gamma < +\pi$



Further Reduce the range: [–π/2, +π/2]

input	β	angle in radian
	n	the number of iterations

$$\theta = angle\_shift(\beta, -\pi)$$

⇩

$$-\pi < \theta < +\pi$$

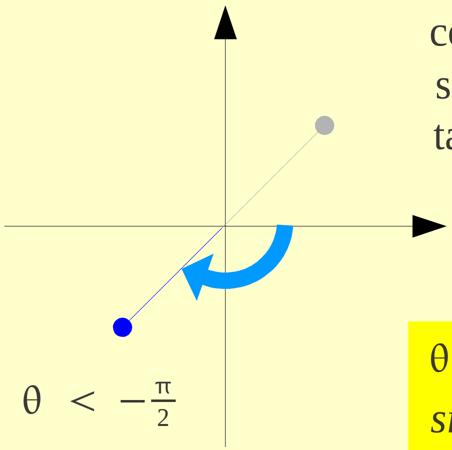
⇩

$\theta < -\frac{\pi}{2}$	$\theta \leftarrow \theta + \pi$ $signfactor \leftarrow -1.0$
$\theta > +\frac{\pi}{2}$	$\theta \leftarrow \theta - \pi$ $signfactor \leftarrow +1.0$

⇩

$$-\frac{\pi}{2} < \theta < +\frac{\pi}{2} \quad , \quad signfactor$$

$$\cos(\theta+\pi) = -\cos\theta$$
$$\sin(\theta+\pi) = -\sin\theta$$
$$\tan(\theta+\pi) = -\tan\theta$$



$\theta < -\frac{\pi}{2}$

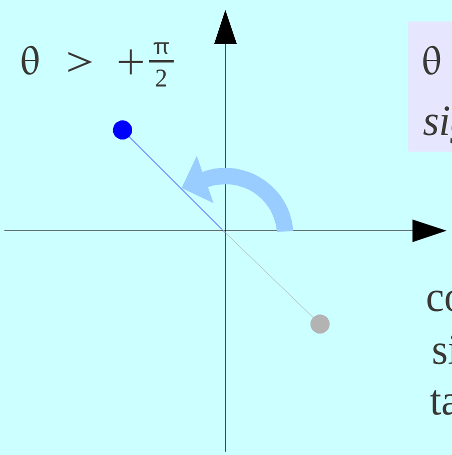
$\theta \leftarrow \theta + \pi$

$signfactor \leftarrow -1.0$

$\theta > +\frac{\pi}{2}$

$\theta \leftarrow \theta - \pi$

$signfactor \leftarrow +1.0$



$$\cos(\theta-\pi) = -\cos\theta$$
$$\sin(\theta-\pi) = -\sin\theta$$
$$\tan(\theta-\pi) = -\tan\theta$$

Precomputed Arrays : angles[60]

```
# define ANGLES_LENGTH 60
# define KPROD_LENGTH 33

  double angle;
  double angles[ANGLES_LENGTH] = {
    7.8539816339744830962E-01,
    4.6364760900080611621E-01,
    2.4497866312686415417E-01,

    …
    0.60725293500888125619,
    0.60725293500888125617 };
```

$\tan^{-1}(1/2^0)$

$\tan^{-1}(1/2^1)$

$\tan^{-1}(1/2^2)$

$\tan^{-1}(1/2^{58})$

$\tan^{-1}(1/2^{59})$

```
    if ( ANGLES_LENGTH < j+1 )  angle = angle / 2.0;
    else                        angle = angles[j];
```

$j \geq 60 \qquad \tan^{-1} 2^{-i} \approx 2^{-i}$

Precomputed Arrays : kprod[33]

```
# define ANGLES_LENGTH 60
# define KPROD_LENGTH 33

  double kprod[KPROD_LENGTH] = {
    0.70710678118654752440,
    0.63245553203367586640,
    0.61357199107789634961,
    …
    0.60725293500888125619,
    0.60725293500888125617 };
```

$$kprod[j] = \prod_{j=0}^{i} K_j$$

$$= \prod_{j=0}^{i} \cos\alpha_i = \prod_{j=0}^{i} \frac{1}{\sqrt{1+1/2^{2i}}}$$

$$K_i = \cos\alpha_i = \frac{1}{\sqrt{1+1/2^{2i}}}$$

$n \geq 33$ $\quad$ *use* $kprod[32]$

```
  if ( 0 < n )
  {
    *c = *c * kprod [ i4_min ( n, KPROD_LENGTH ) - 1 ];
    *s = *s * kprod [ i4_min ( n, KPROD_LENGTH ) - 1 ];
  }
```

cossin_cordic (1)

$\theta < 0 \Rightarrow \sigma = -1$
$\theta > 0 \Rightarrow \sigma = +1$

$$\theta = \theta - \sigma \cdot angle$$

$60 < j+1$	$angle = angle/2$
	$angle = angles(j+1)$

$$angles(60)$$

$$angles(1) \Rightarrow \tan^{-1}\left(\frac{1}{2}\right)$$

$$angles(2) \Rightarrow \tan^{-1}\left(\frac{1}{2^2}\right)$$

$$angles(3) \Rightarrow \tan^{-1}\left(\frac{1}{2^3}\right)$$

$$poweroftwo = 1.0$$

$$factor = \sigma \cdot poweroftwo$$

$$poweroftwo = poweroftwo/2$$

$$\left(\frac{1}{2}\right)^{L} = \left(\frac{1}{2}\right)^{j-1}$$

$$j = 1 \Rightarrow poweroftwo = 1/2^0$$

$$j = 2 \Rightarrow poweroftwo = 1/2^1$$

$$j = 3 \Rightarrow poweroftwo = 1/2^2$$

cossin_cordic (2)

```
*c = 1.0;
*s = 0.0;

poweroftwo = 1.0;
angle = angles[0];

for ( j = 1; j <= n; j++ )
{θ = θ − σ · angle
  if ( theta < 0.0 ) sigma = -1.0;
  else               sigma = 1.0;

  factor = sigma * poweroftwo;

  c2 =          *c - factor * *s;
  s2 = factor * *c +          *s;

  *c = c2;
  *s = s2;

  theta = theta - sigma * angle;

  poweroftwo = poweroftwo / 2.0;

  if ( ANGLES_LENGTH < j + 1 ) angle = angle / 2.0;
  else                         angle = angles[j];
}
```

$$R = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} = \cos\theta_i \begin{bmatrix} 1 & -\tan\theta_i \\ \tan\theta_i & 1 \end{bmatrix}$$

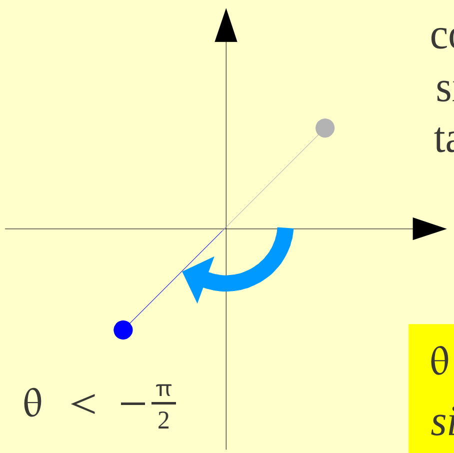
$$\begin{bmatrix} c \\ s \end{bmatrix} = \begin{bmatrix} 1 & -\sigma 2^{-i} \\ \sigma 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} c \\ s \end{bmatrix}$$

$$\theta = \theta - \sigma \cdot angle$$

Young Won Lim
2/28/12

cossin_cordic (3)

```
if ( 0 < n )
{
  *c = *c * kprod [ i4_min ( n, KPROD_LENGTH ) - 1 ];
  *s = *s * kprod [ i4_min ( n, KPROD_LENGTH ) - 1 ];
}

*c = sign_factor * *c;
*s = sign_factor * *s;
```

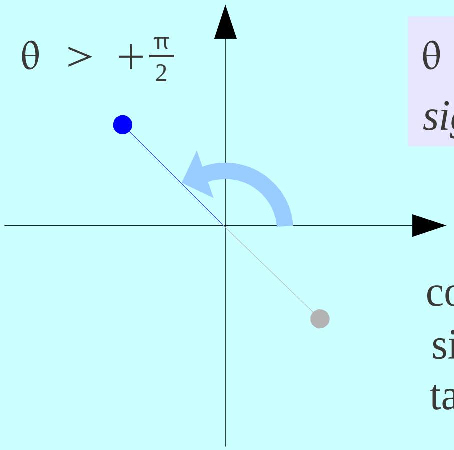


$$\cos(\theta+\pi) = -\cos\theta$$
$$\sin(\theta+\pi) = -\sin\theta$$
$$\tan(\theta+\pi) = -\tan\theta$$

$\theta < -\frac{\pi}{2}$

$\theta \leftarrow \theta+\pi$

$signfactor \leftarrow -1.0$



$\theta > +\frac{\pi}{2}$

$\theta \leftarrow \theta-\pi$

$signfactor \leftarrow +1.0$

$$\cos(\theta-\pi) = -\cos\theta$$
$$\sin(\theta-\pi) = -\sin\theta$$
$$\tan(\theta-\pi) = -\tan\theta$$

Young Won Lim
2/28/12

References

[1] http://en.wikipedia.org/
[2] http://people.sc.fsu.edu/~jburkardt/m_src/cordic/cordic.html