

# CORDIC Accuracy Statistical Analysis

20160122

Copyright (c) 2015 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Dense Angle Area

Block & Offset Views

Subtrees and Indices

Residue Angle Distribution

Overlapping & Non-overlapping Region

Jitter Angle

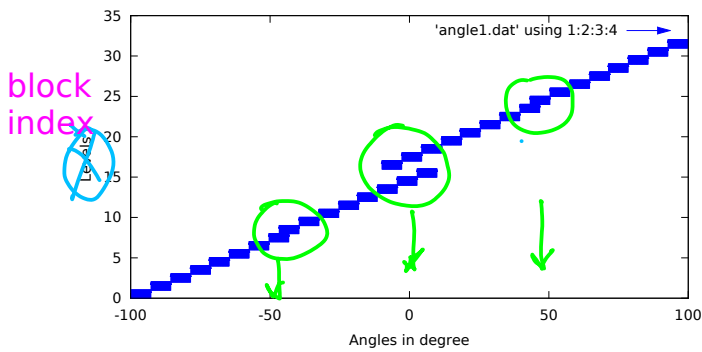
Basic Figures  
June 11, 2014

egb1.Leaf\_11.ang\_tree1.n2048.eps  
egb1.Leaf\_11.ang\_tree2.n2048.eps  
egb2.Leaf\_11.circle\_ang.n2048.eps  
egb3.Leaf\_11.line\_ang.i0.n2048.eps  
egb4.Leaf\_11.quantization.n2048.eps  
egb1.All\_11.ang\_tree1.n4095.eps  
egb1.All\_11.ang\_tree2.n4095.eps  
egb1.All\_11.ang\_tree3.n4095.eps  
egb2.All\_11.circle\_ang.n4095.eps  
egb3.All\_11.line\_ang.i0.n4095.eps  
egb4.All\_11.quantization.n4095.eps

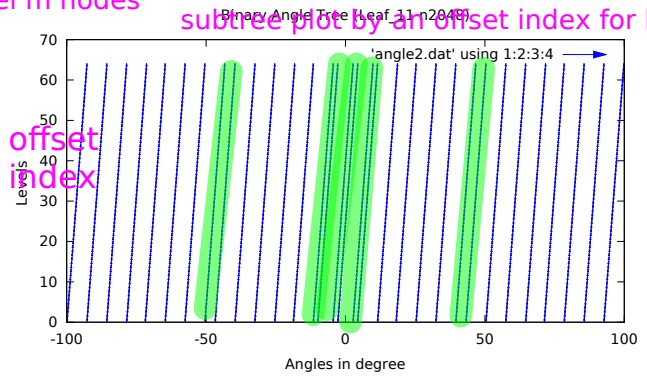
# Dense Angle Area



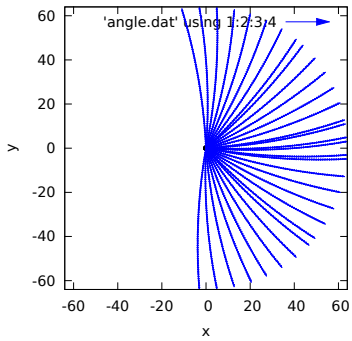
subtree plot by a block index for level m nodes



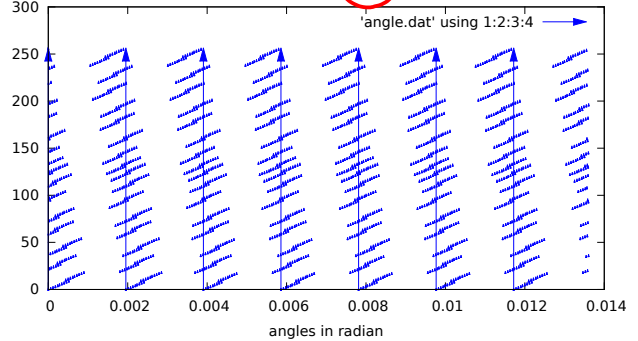
subtree plot by an offset index for level m nodes



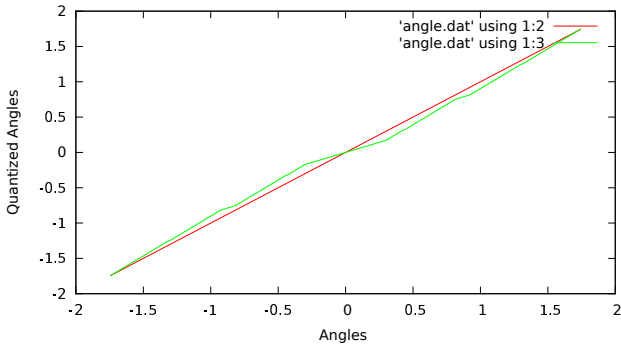
Circular angle vectors by the offset in a block (Leaf\_11 n2048)



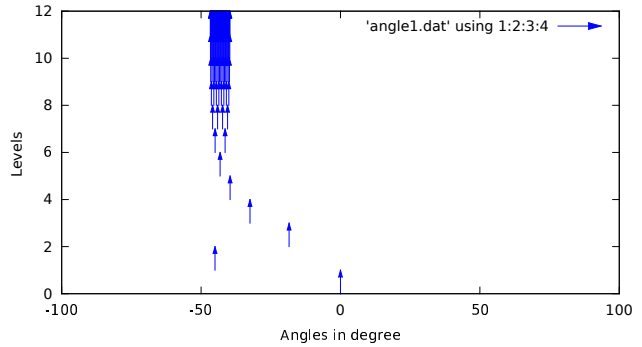
Linear angle vectors showing jitter (Leaf\_11 n2048)



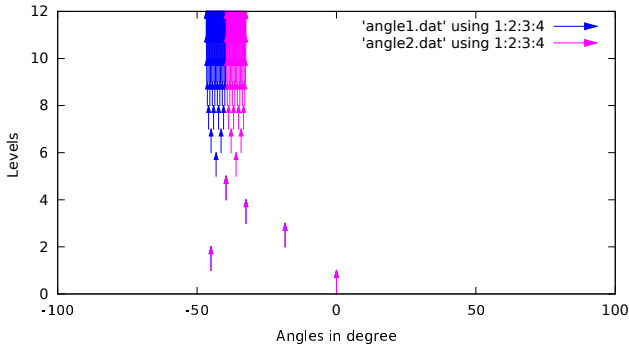
Quantization Effect (Leaf\_11 n2048)



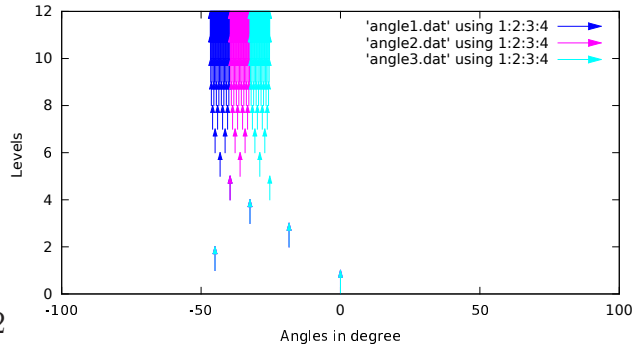
Binary Angle Tree (All\_11 n4095)



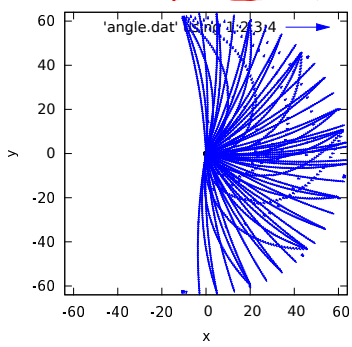
Binary Angle Tree (All\_11 n4095)



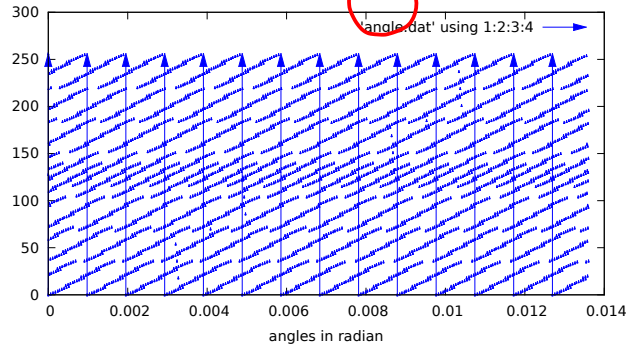
Binary Angle Tree (All\_11 n4095)



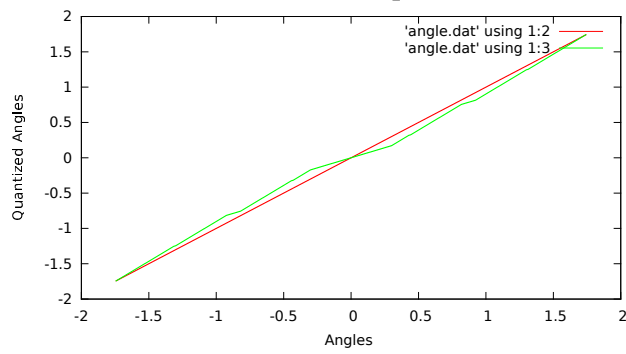
Circular angle vectors by the offset in a block (All\_11 n4095)



Linear angle vectors showing jitter (All\_11 n4095)



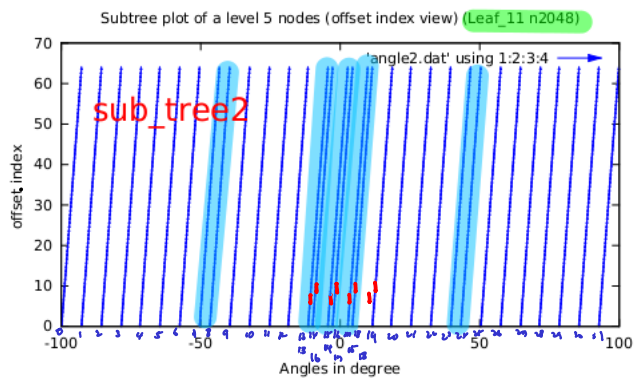
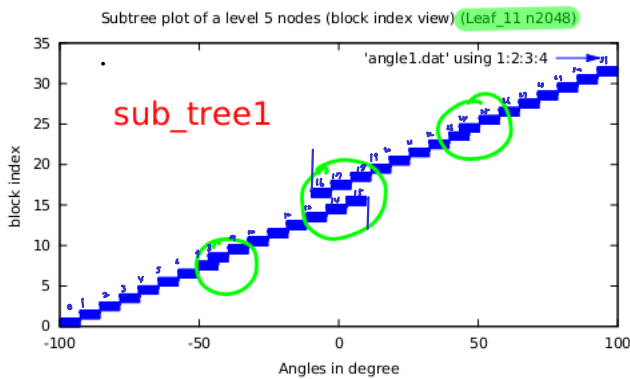
Quantization Effect (All\_11 n4095)



Small slope  
→ dense



# Block & Offset Index



angle precision  
 angles around 0 degree and 45 degrees  
 leaf level angle values share the same range

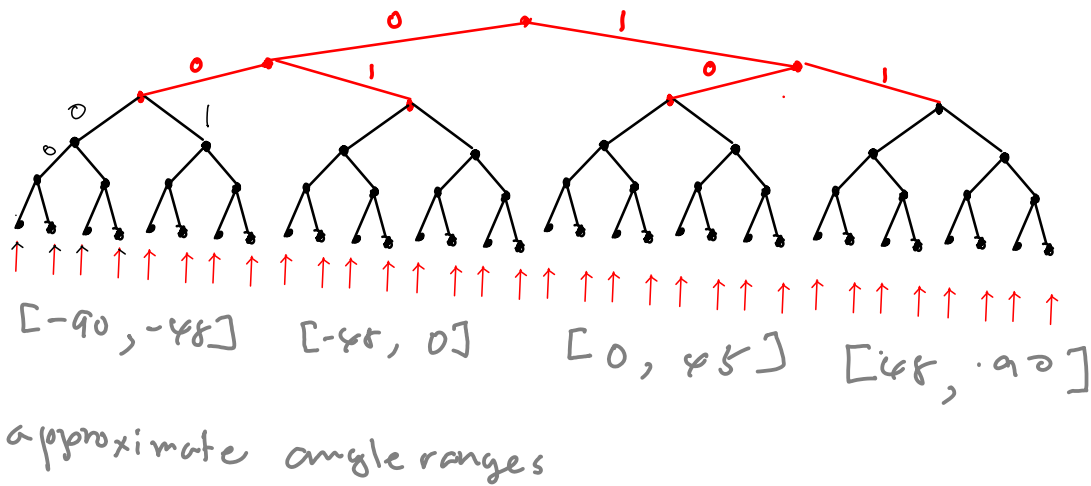
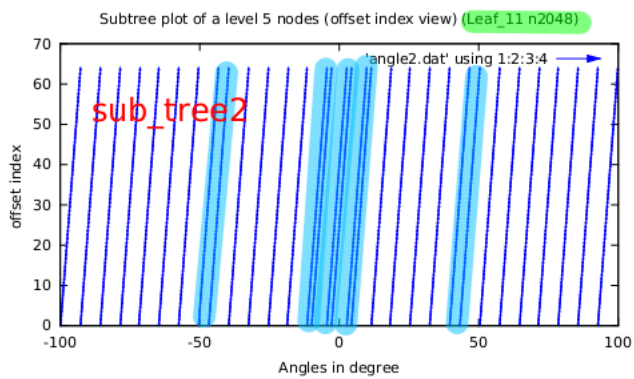
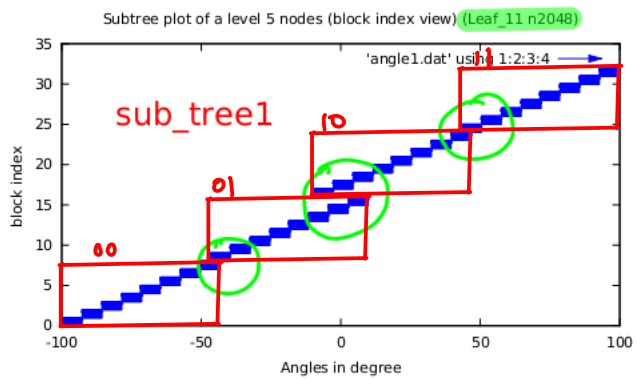
block first plot  
 x axis represents angle values in degree  
 y axis represents block index  
 when leaf level angle values are partitioned into blocks  
 block index for group number  
 offset index for individual number within a group  
 chungle of arrows within a specific block (group) looks like rectangle

offset first plot  
 the first slant line - the first block  
 the arrows in the first block is plotted as a small arrow  
 each block has a 65 members and thus offset index is less than 65  
 there are 32 blocks (groups)  
 each block has 64 angle point

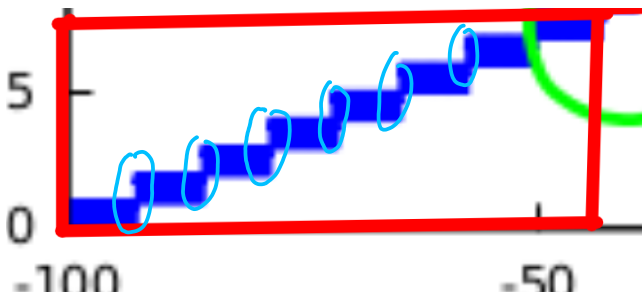
y axis represents offset index  
 As can be seen these group almost exclusive except that 3 regions

The effect of this is not yet analyzed throughly

Is this overlapped angles helps to improve angle precision?  
 Can this overlapped angles be removed without sacrificing the performance?  
 The total count of this overlapped region comprises almost 50%.





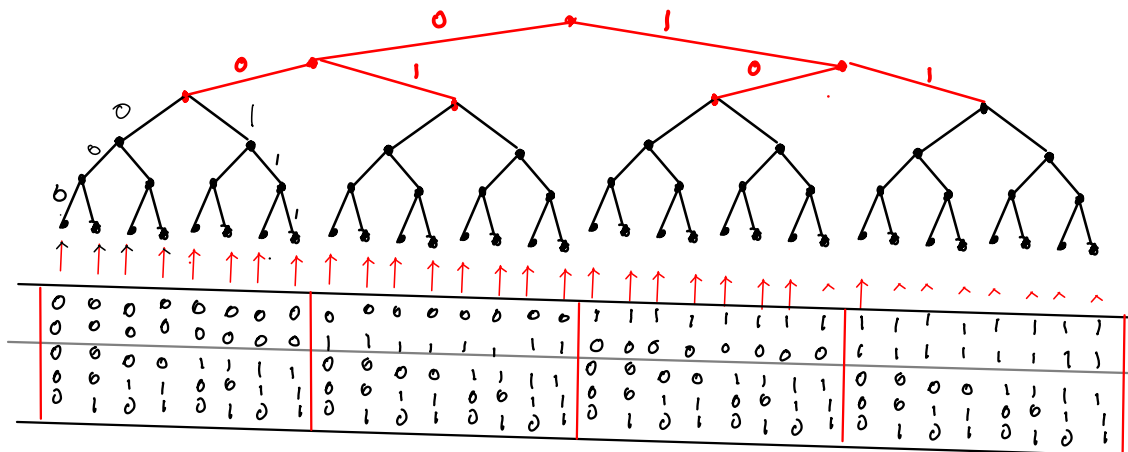
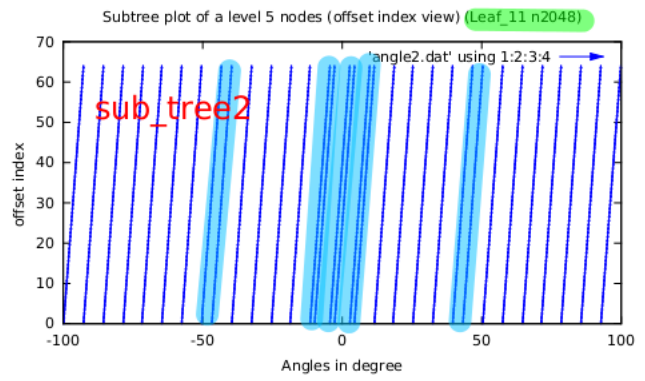
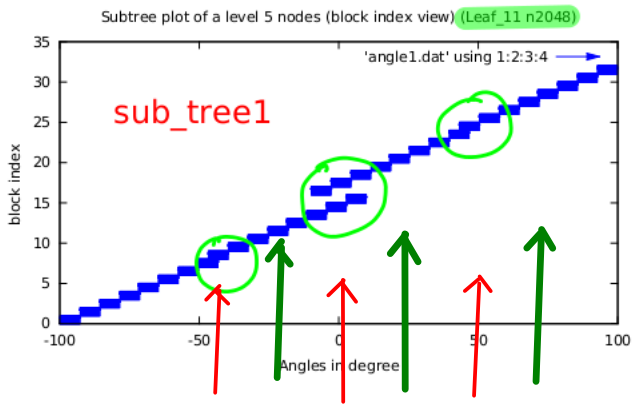


0 0	$[-90, -45]$	$\leftarrow$ $(-45)$
0 1	$[-45, 0]$	$\leftarrow$ $(0)$
1 0	$[0, 45]$	$\leftarrow$ $(45)$
1 1	$[45, 90]$	$\leftarrow$ $(45)$

approximate

$$\begin{aligned}
 -45 + 22.5 &= \frac{-22.5}{2} \\
 \parallel & \\
 x + 22.5 &= (y) \quad \text{find } (y)
 \end{aligned}$$

# Subtrees and Indices



Let angle points denote angle values of the leaf node in the CORDIC binary tree linear combination of  $\text{atan}(1/2^i)$  angle values where the constant coefficients are either +1 or -1.

the spacing of the each adjacent angle points is not uniform. Specially, in the region of 0, +45, -45 degrees the angle spacing value is extremely fluctuating

This comes from the fact that the range of angle points of a node and that of its sibling nodes are not exclusive but overlapped. So the overlapping region looks like more fine resolution compared with non-overlapped region.

The question is when the given angle is one of the these fine resolution area, the CORDIC algorithms can get benefits or just take more iterations to converge minimizing residue errors.

Another question is what is the average spacing value or resolution in angle - how to

block view

leaf angle points are grouped or partitioned into a block  
those angle points in a block have the same the same ancestor  
and the angle spacing is quite uniform.

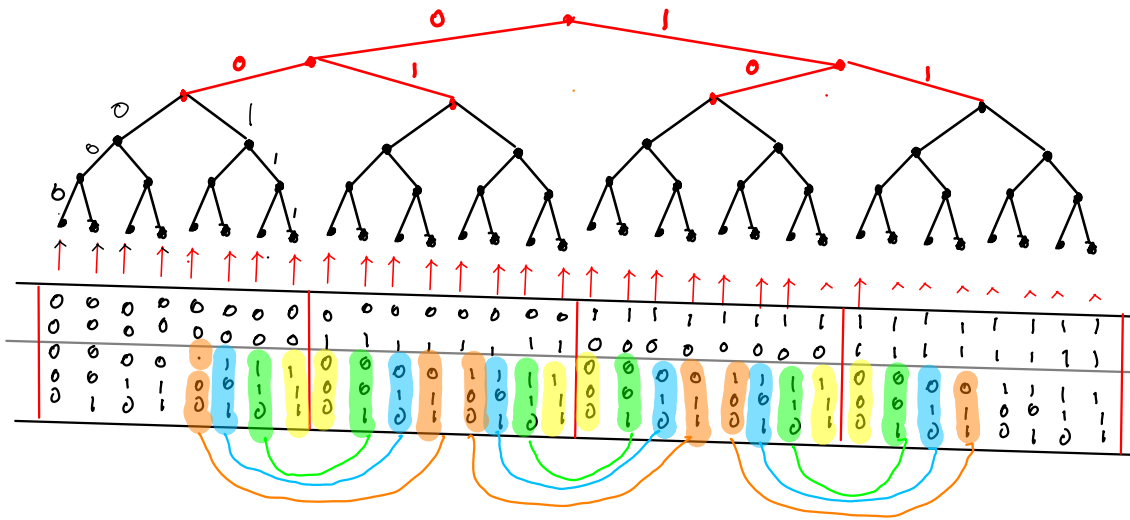
But angles points in a block are overlapped with those of adjacent  
blocks and angle and the increasing order is not maintained and  
resulting angles spacing becomes quite irregular

this effect comes from the fact that when the level is relatively great  
then  $\arctan(1/2^i)$  is approximated with  $1/2^j$ . this is not true  
when level  $i$  is not big.

block size is related to the ancestor level  
we can view how much angle range is overlapped with the adjacent  
blocks

offset view

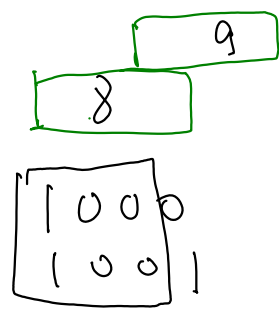
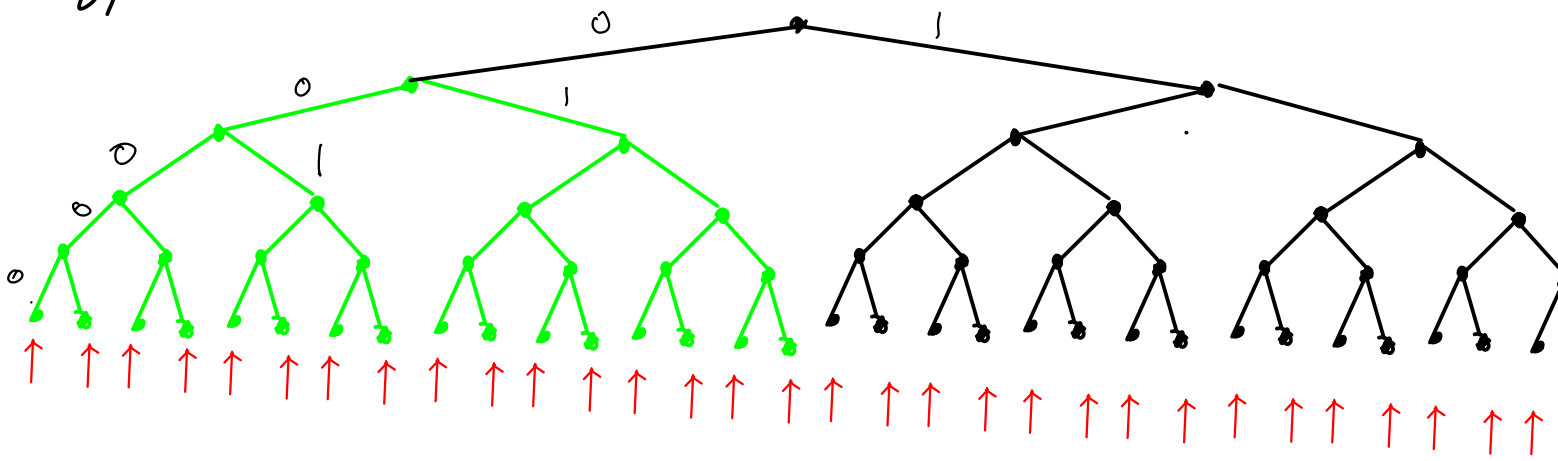
the strong slope existing in the offset view comes from the  
quite uniform angle spacing values.



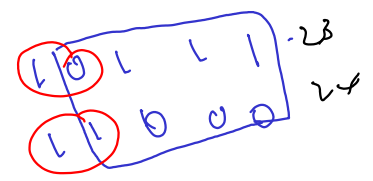
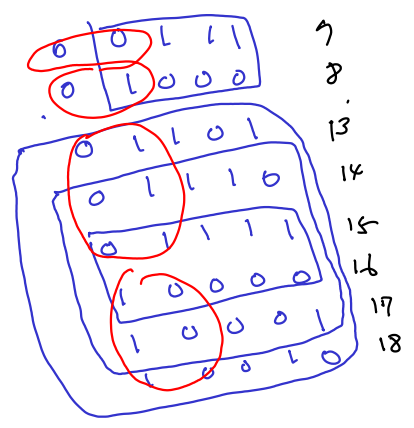
8<sup>9</sup> 11<sup>17</sup> 19<sup>23</sup>

blocks 7,8  
 blocks 13,14,15,16,17,18  
 blocks 23,24

16  
 ---  
 24

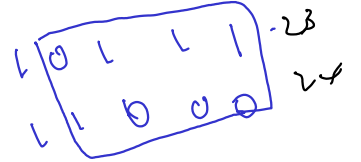
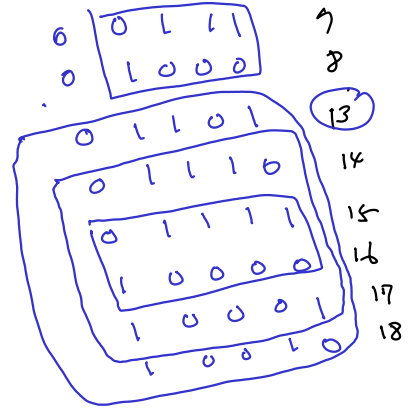
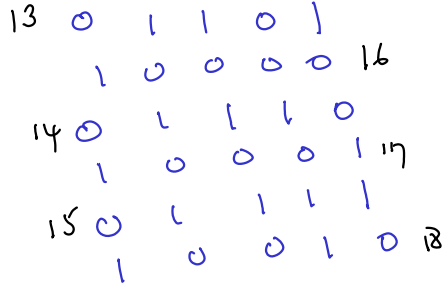
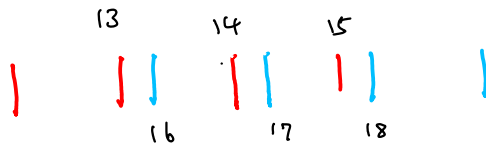


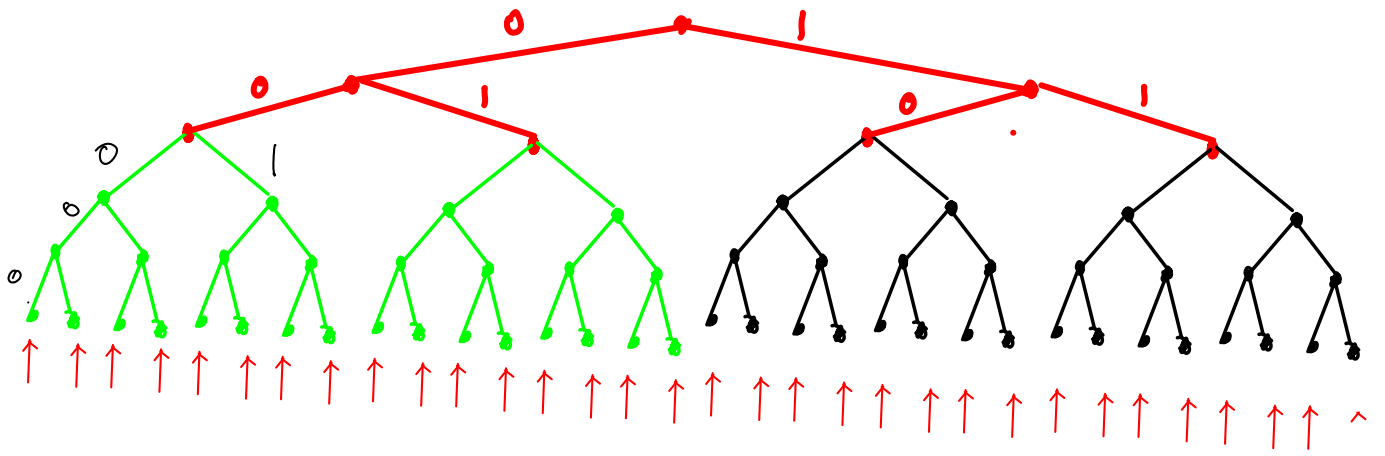
blocks 7,8  
 blocks 13,14,15,16,17,18  
 blocks 23,24



☆ round off errors  
 offset binary effect  
 always increasing ?

need to check!

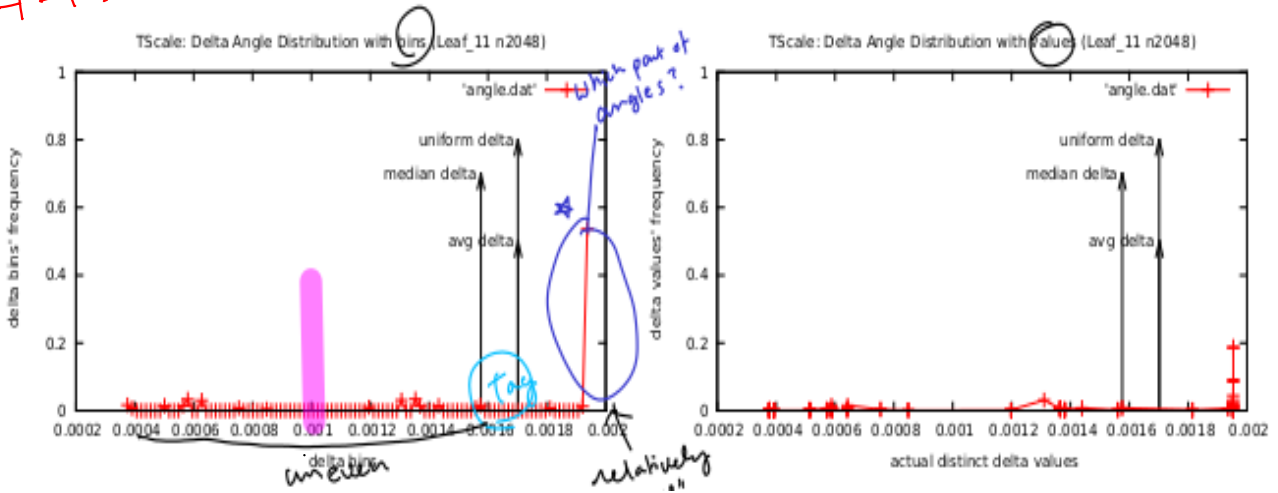




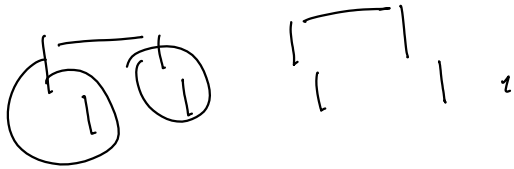


# Residue Angle Distribution

H → →



post-sorting



use tag

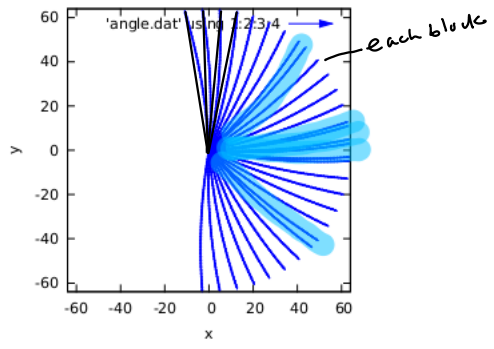
\* Separate samples

① overlapped region

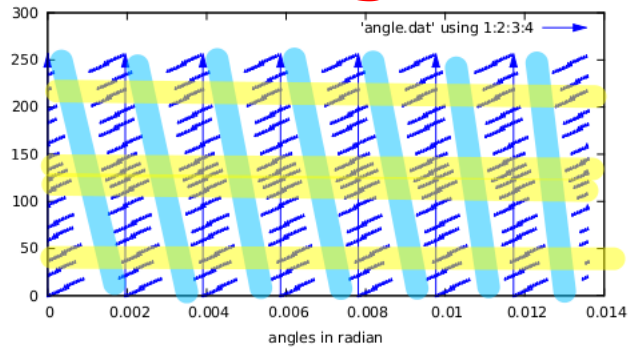
② non-overlapped region

45°, 90°, 135°

Circular angle vectors by the offset in a block (Leaf 11 n2048)



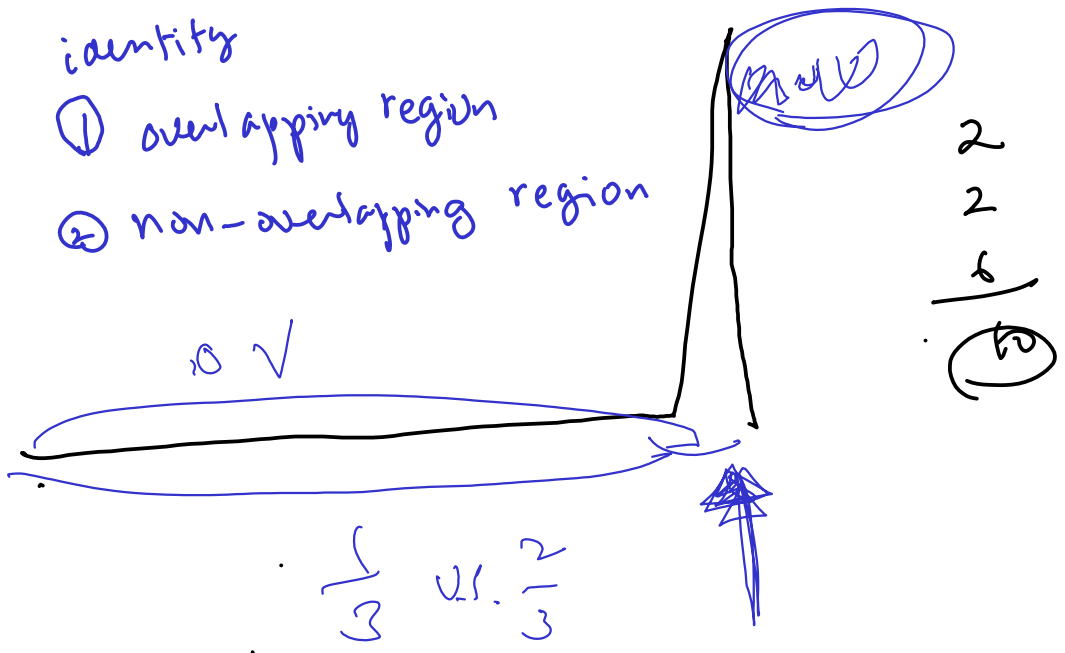
Linear angle vectors showing jitter (Leaf 11 n2048)



# Overlapping & Non-overlapping Region dense angle

identity

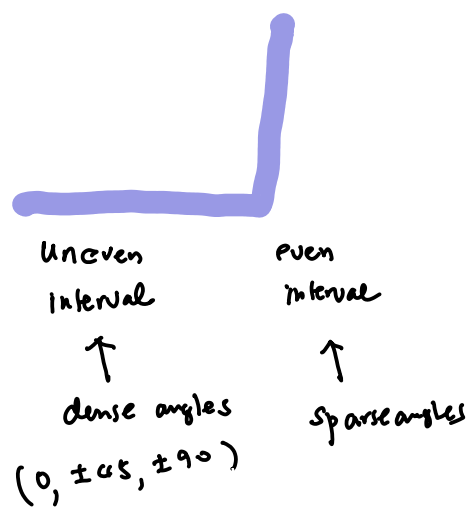
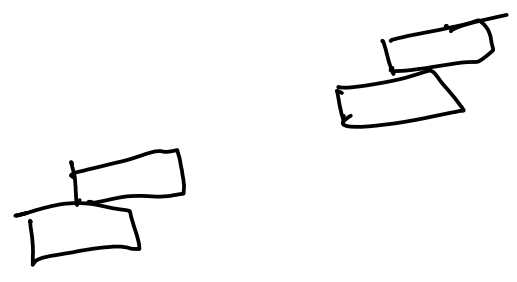
- ① overlapping region
- ② non-overlapping region



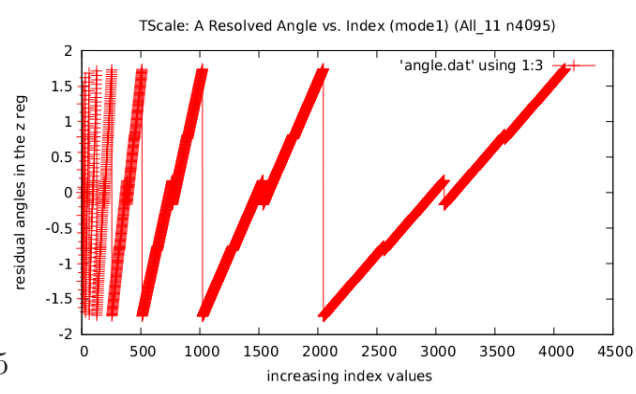
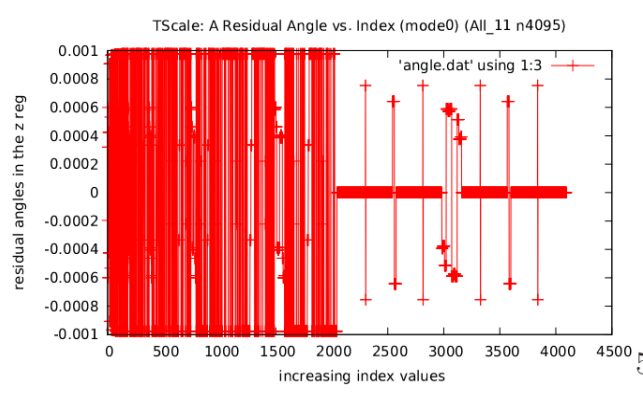
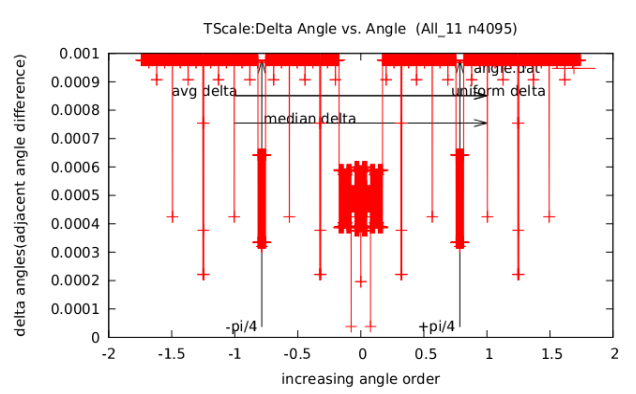
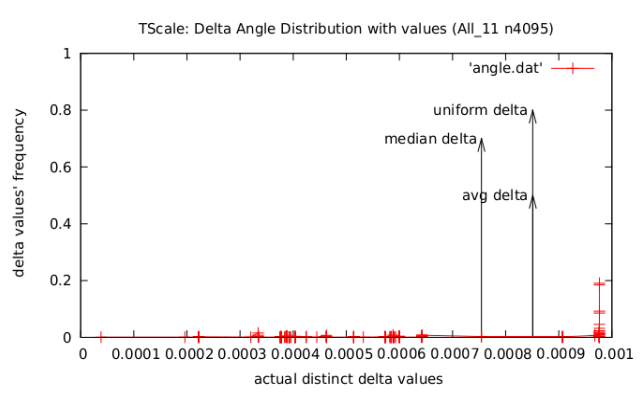
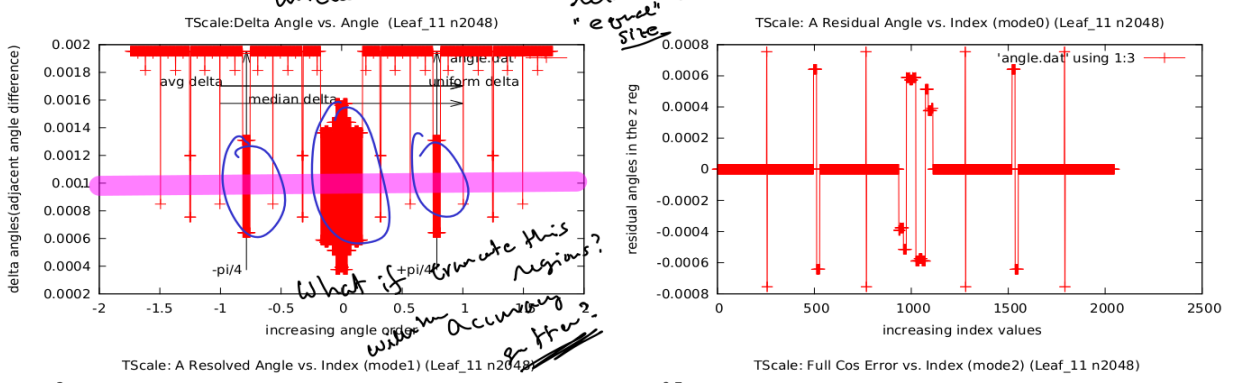
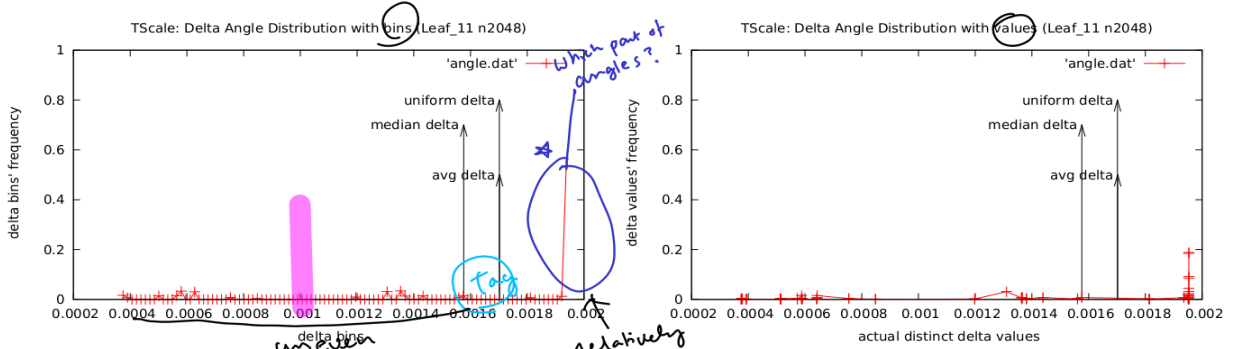
$\frac{1}{3}$  vs.  $\frac{2}{3}$

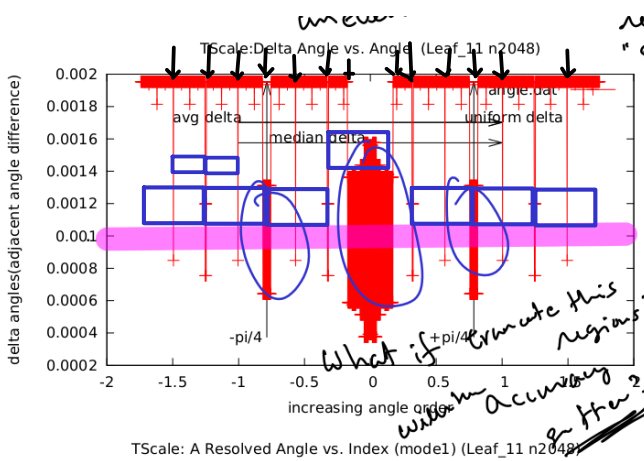
$\frac{32 \text{ blocks}}{10 \text{ blocks}} = 22 \text{ blocks}$

OV  
NON

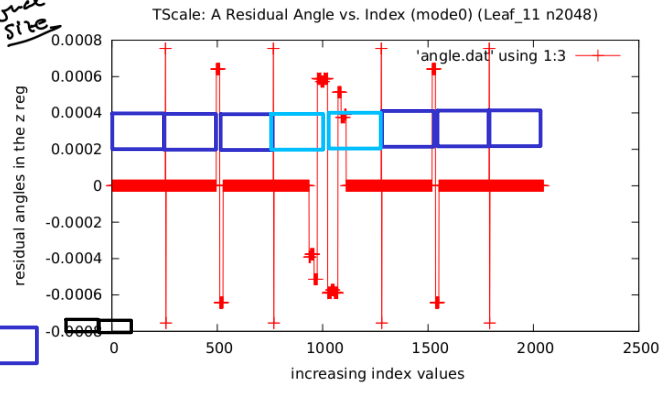


# \* Residue Distribution





TScale: A Resolved Angle vs. Index (mode1) (Leaf\_11 n2048)



TScale: Full Cos Error vs. Index (mode2) (Leaf\_11 n2048)

Overlapped Region → uneven resolution  
 Non-ov Region → relatively even results

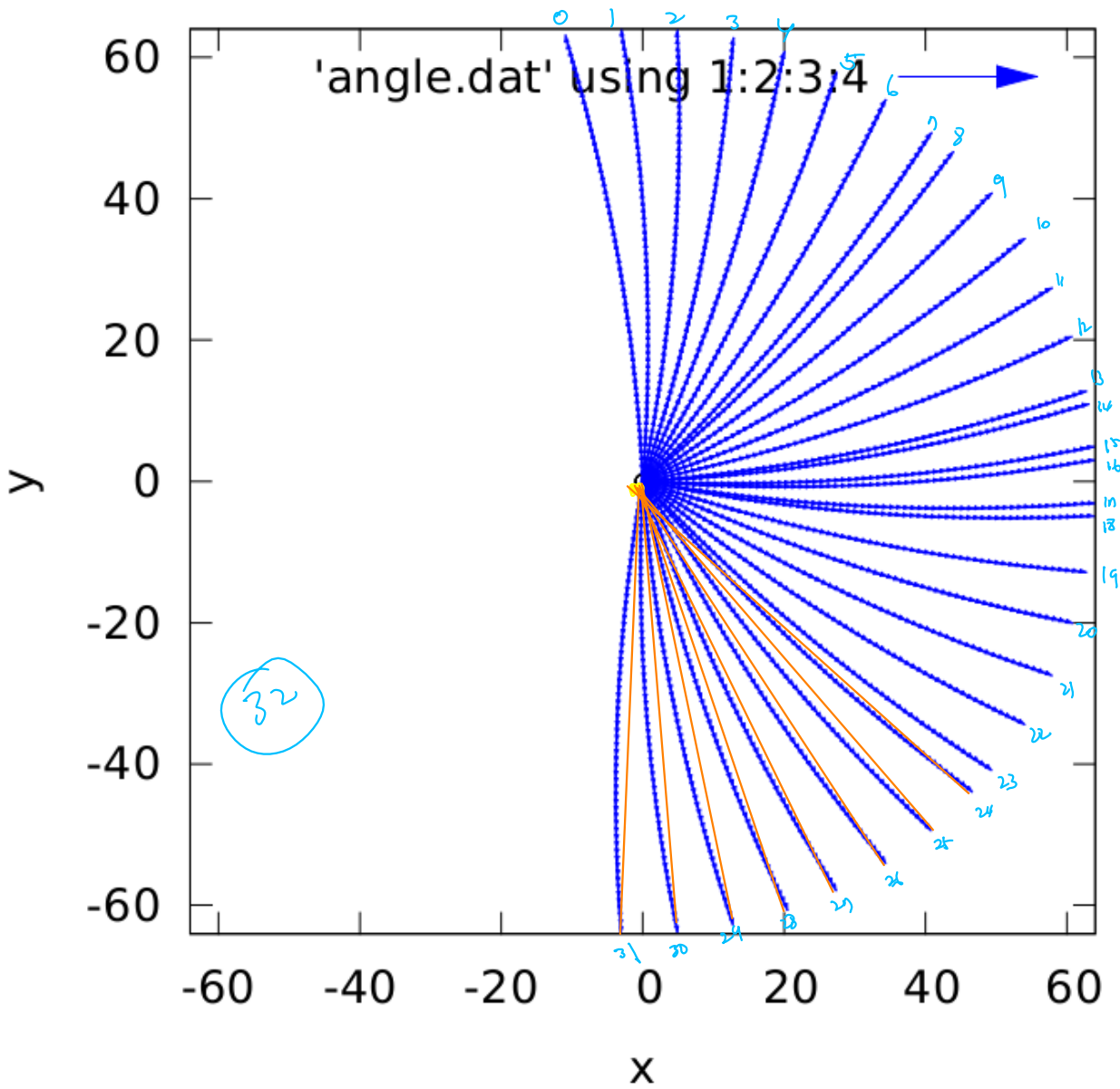
Seems difficult to converge a value in ov region

residual error large!!!



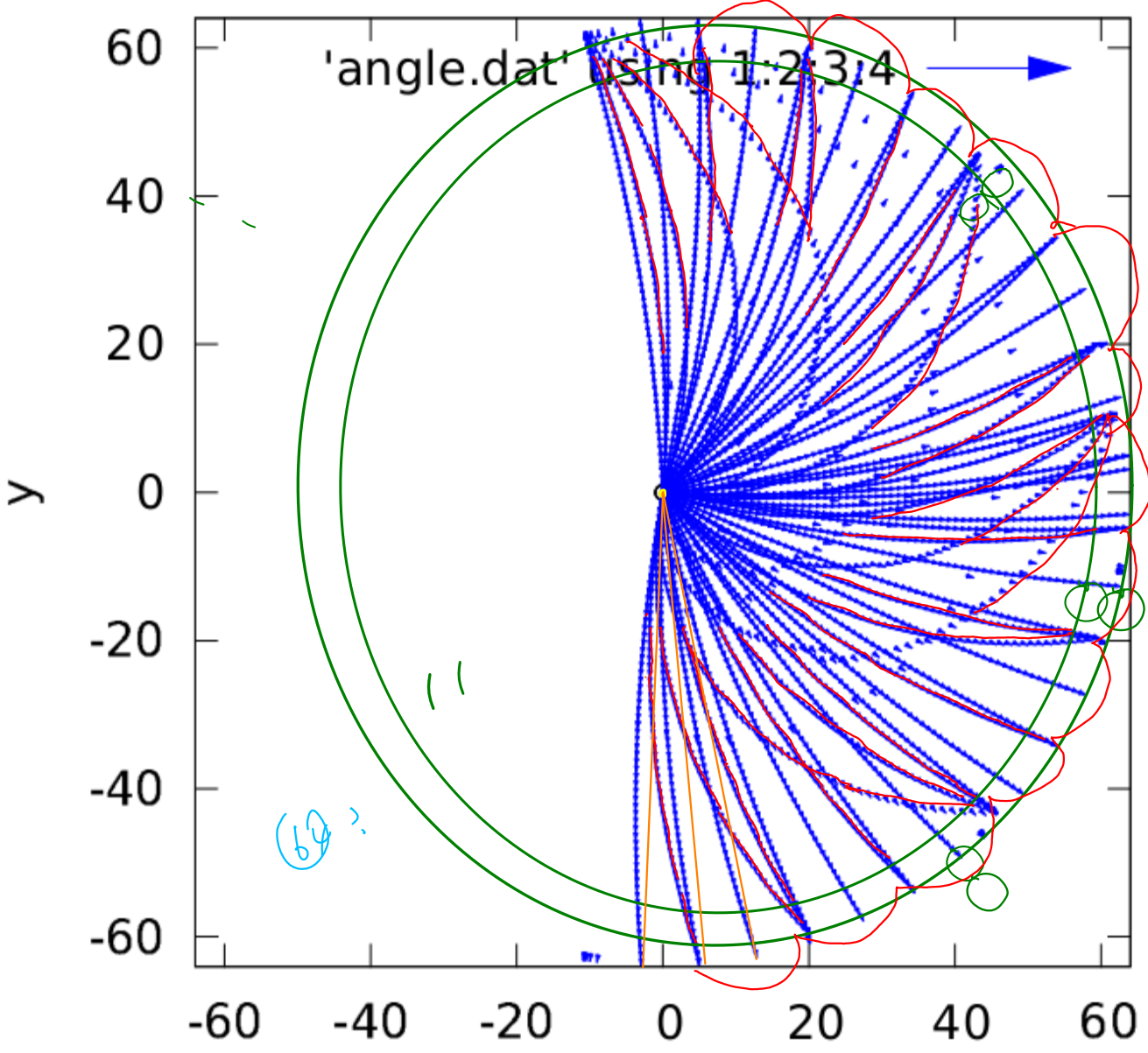
Leaf

circular angle vectors by the offset in a block (Leaf\_11 n2048)



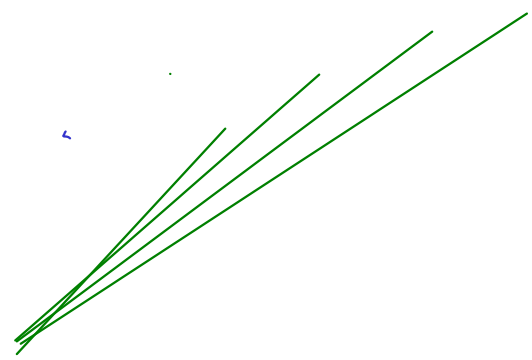
All

Circular angle vectors by the offset in a block (All\_11 n4095)

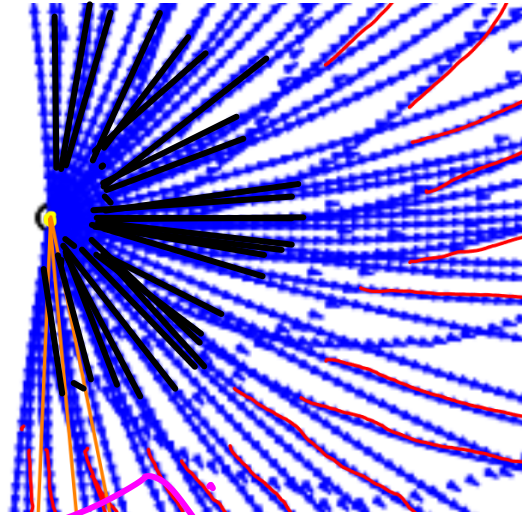
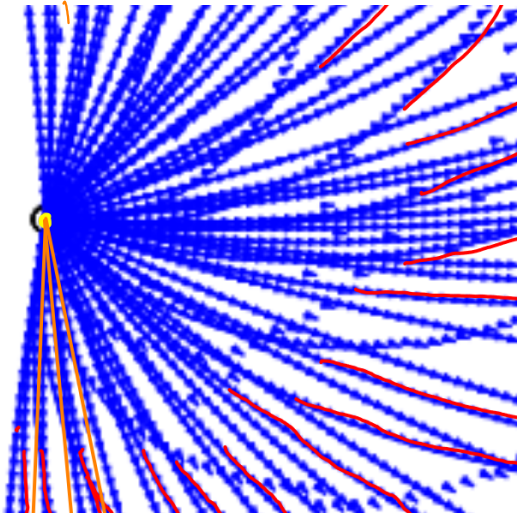


- Change colors  
- Change arrow

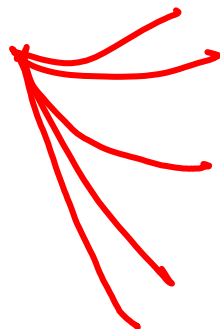
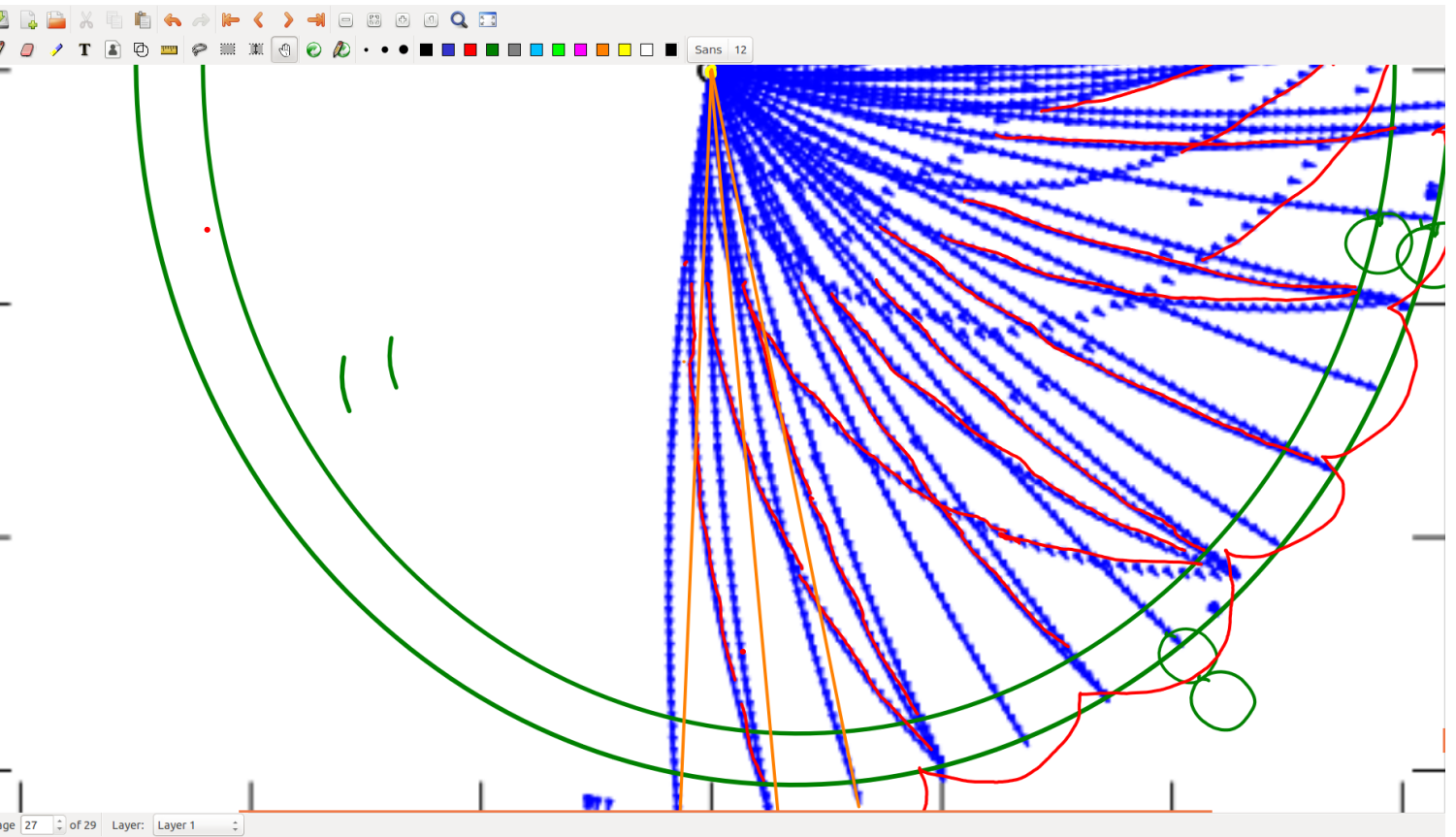
x lengths



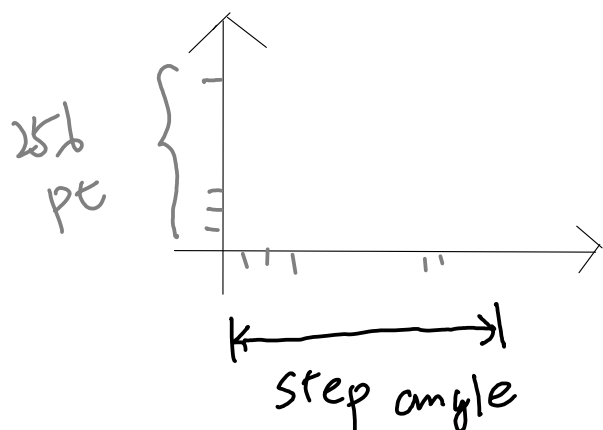




rotate



# Jitter Angle



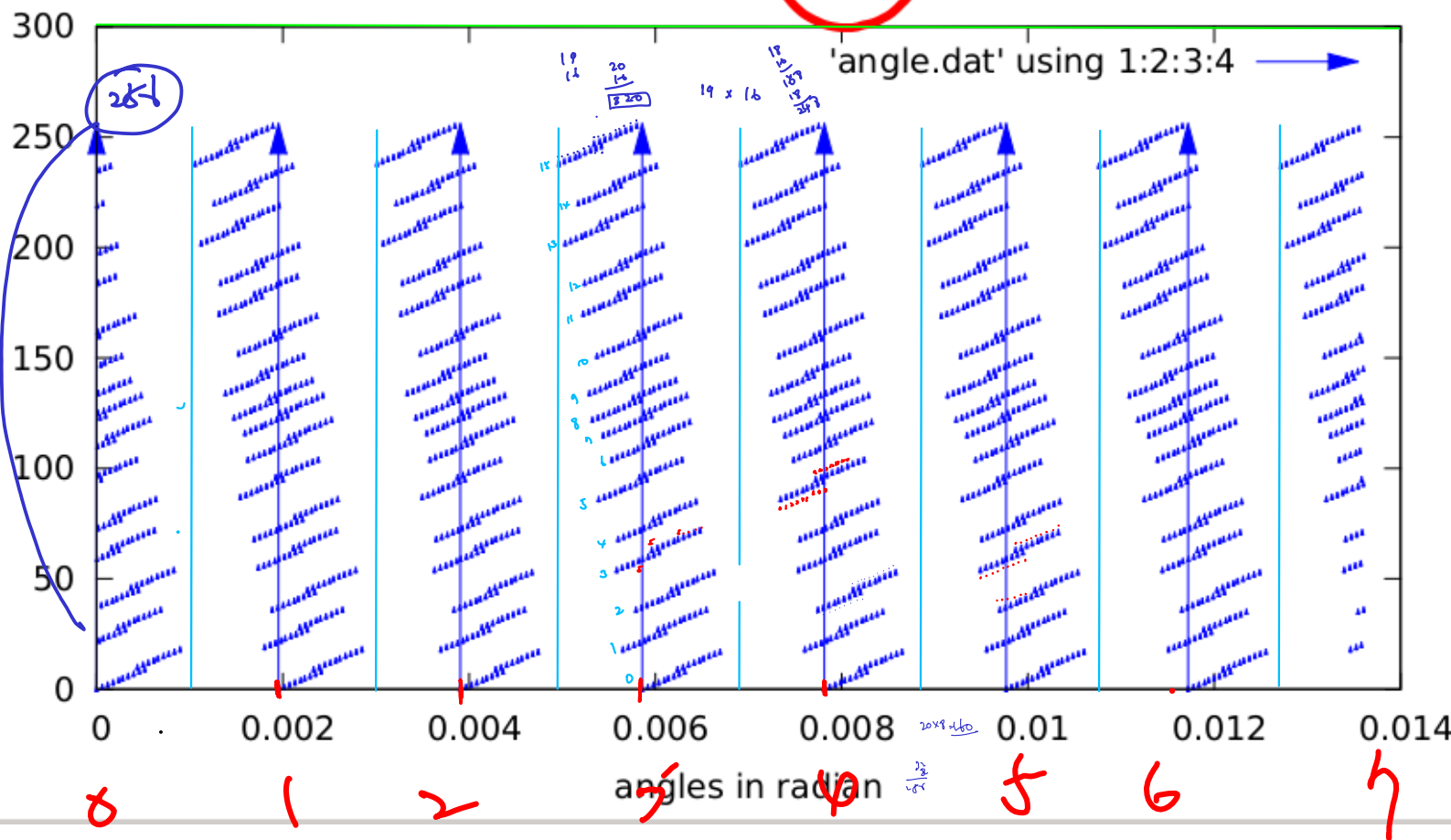
binnum = 256

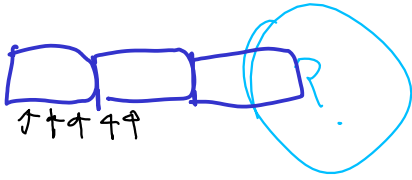
actually bin size

currently  $\sim$  step angle

relative  $\times$  position of each bin.

Linear angle vectors showing jitter (Leaf\_11 n2048)



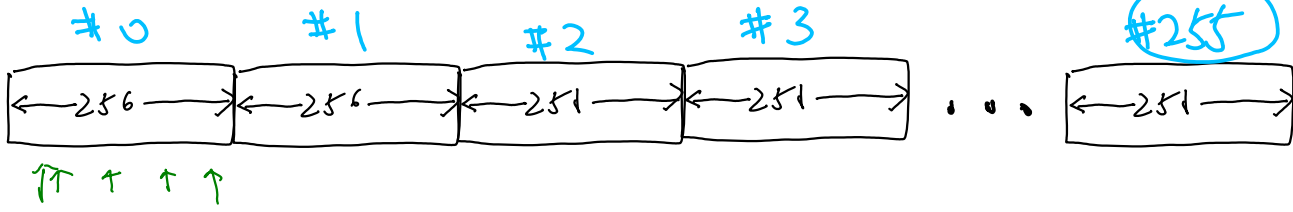


$$8 \times 16 =$$

$$2^{11} / (2^8) = 2^3 = 8$$

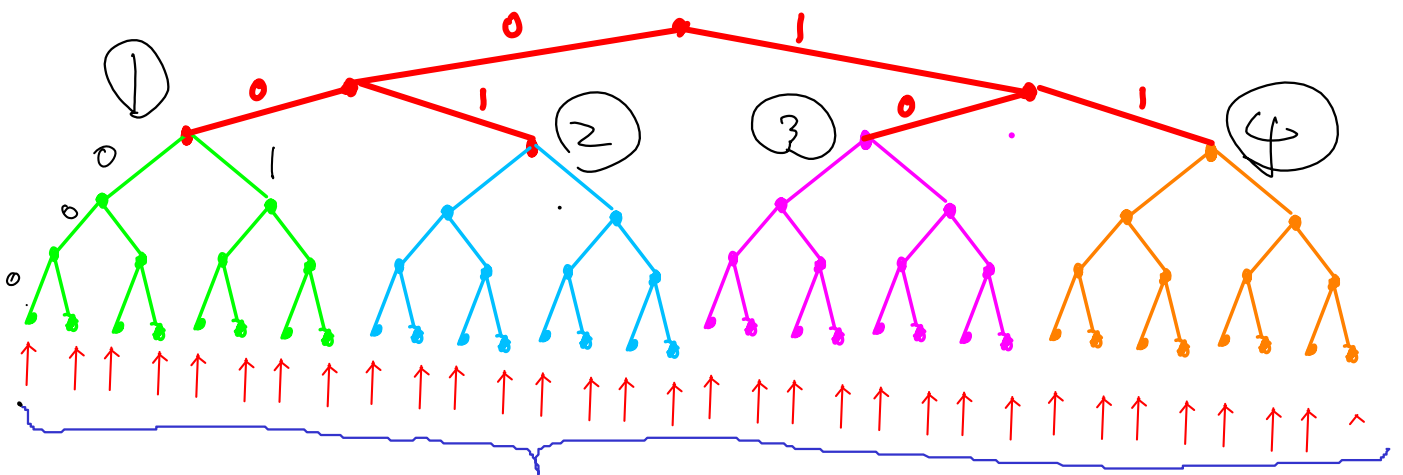
binnum

bin # →



un-even.

there are 256 bins



$A[]$  *sort*

Sorted Angle point  $B[]$

partition the whole angle range ( $B[nAngles - 1] - B[0]$ )  
 into 256 bins (= blocks)

binnum = 256  
 step =  $(B[n-1] - B[0]) / \text{binnum}$   
 ang =  $B[i] - B[0]$   
 hpos =  $\text{int}(ang / \text{step})$   
 xpos =  $\text{fmod}(ang / \text{step})$

each bin covers the "step" angle

rad  
 $(B[nAngles - 1] - B[0])$





$$\text{ang} = \beta[i] - \beta[0]$$

relative angle w.r.t.  $\beta[0]$



$$\text{Step} = \frac{\beta[nAngles - 1] - \beta[0]}{256} \quad \because \text{angle range per bin}$$

there are total 256 (=binnum) bins  
each bin covers the step angle.

$hpos = \text{int}(\text{ang} / \text{step}) \rightarrow$  one of 256 bins

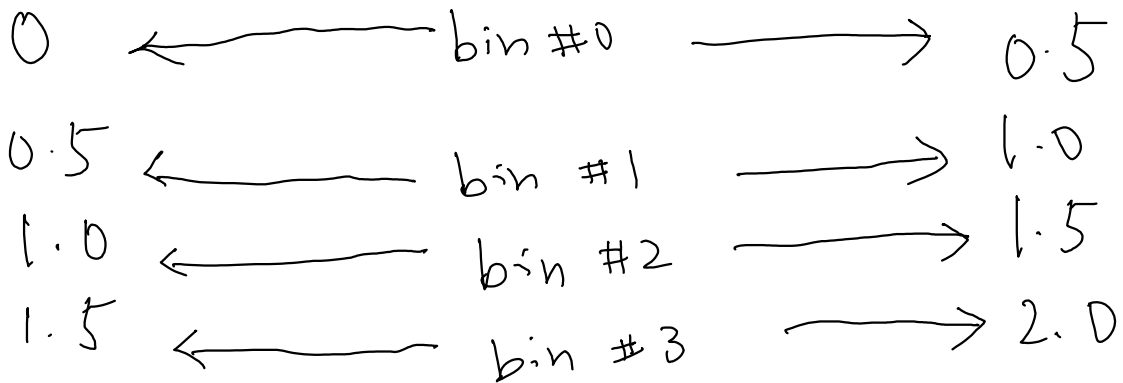
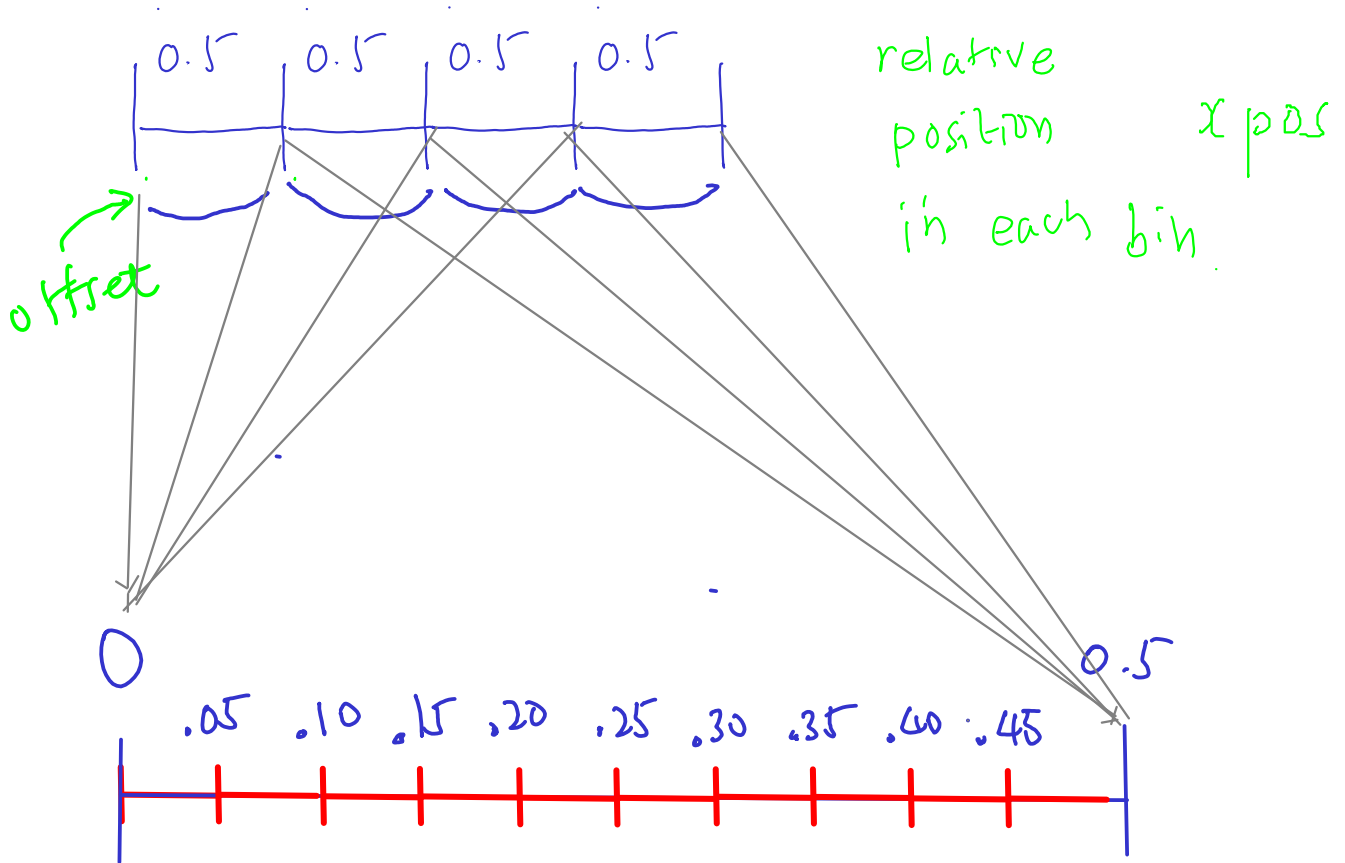
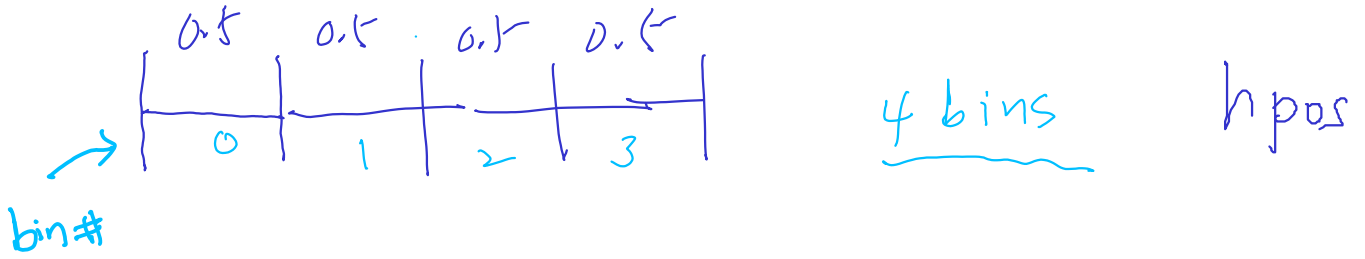
$xpos = \text{fmod}(\text{ang}, \text{step}) \rightarrow$  relative position of  $\text{ang}$  within a bin.

$\text{ang} / \text{step} \rightarrow$  bin # (block #)

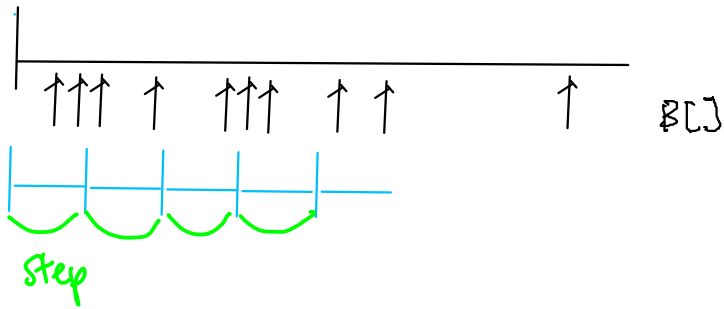
$\text{ang} \% \text{step} \rightarrow$  offset within a bin (block)

# Example ①

Step angle = 0.5



# Example (2)



$$\text{Step} = \frac{B[nAngles - 1] - B[0]}{256}$$

$\text{ang} / \text{step} \dots \rightarrow$  bin # (block #)

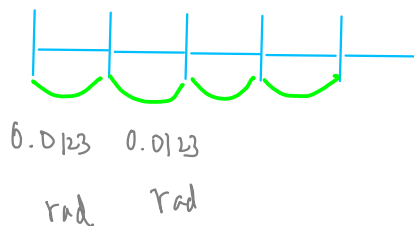
$\text{ang} \% \text{Step} \dots \rightarrow$  offset within a bin (block)

roughly  $B[0] \approx -90^\circ$ ,  $B[nAngles - 1] \approx +90^\circ$   
 angle variable falls into  $[0, 180^\circ]$   
 there are 256 bins  
 one bin covers  $180^\circ / 256 \Rightarrow \text{Step}$

step  $\Rightarrow$

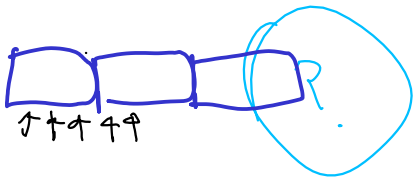
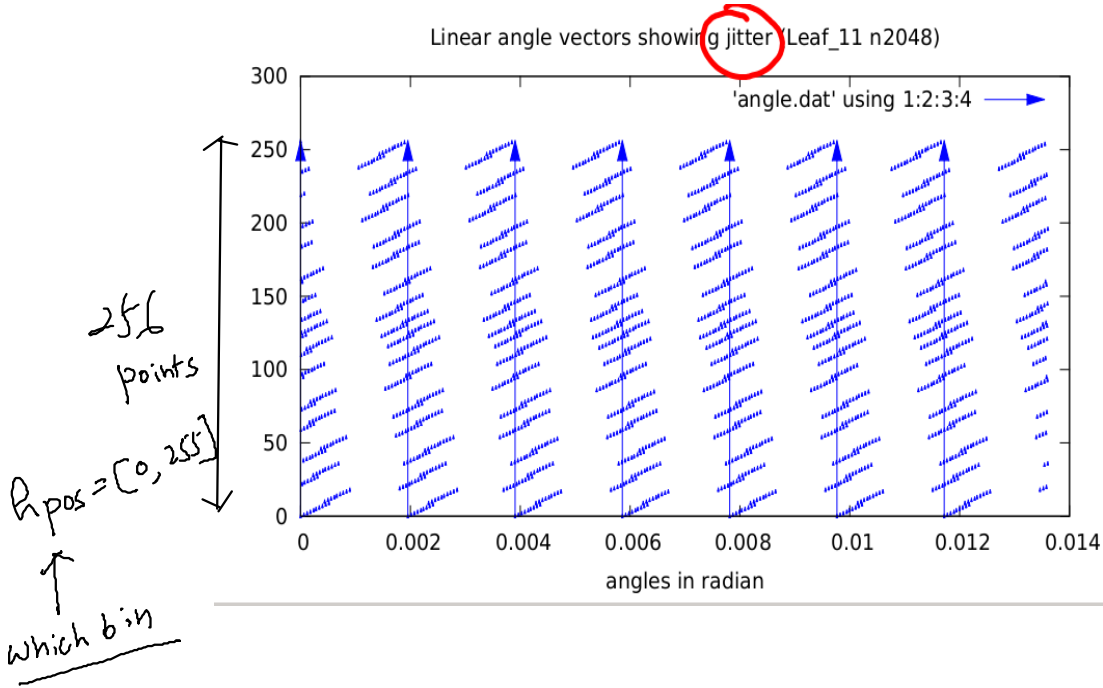
$$\frac{180}{256} = \underline{0.703 \text{ deg}}$$

$$0.703 \times \frac{\pi}{180} = \underline{0.0123 \text{ rad}}$$



$$0.0123 \times 256 = \pi \text{ radians}$$

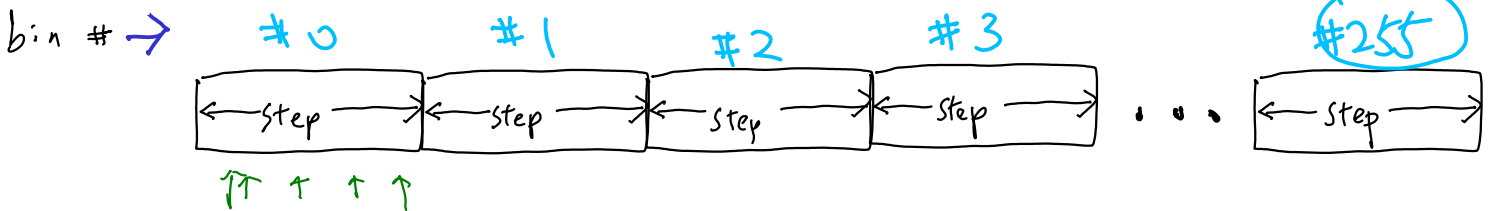
$\text{binnum} = 256$   
 $\text{step} = (B[n-1] - B[0]) / \text{binnum}$   
 $\text{ang} = B[i] - B[0]$   
 $\text{hpos} = \text{int}(\text{ang} / \text{step})$   
 $\text{xpos} = \text{fmod}(\text{ang} / \text{step})$



$8 \times 16 =$

$2^{11} / (2^8) = 2^3 = 8$

binnum



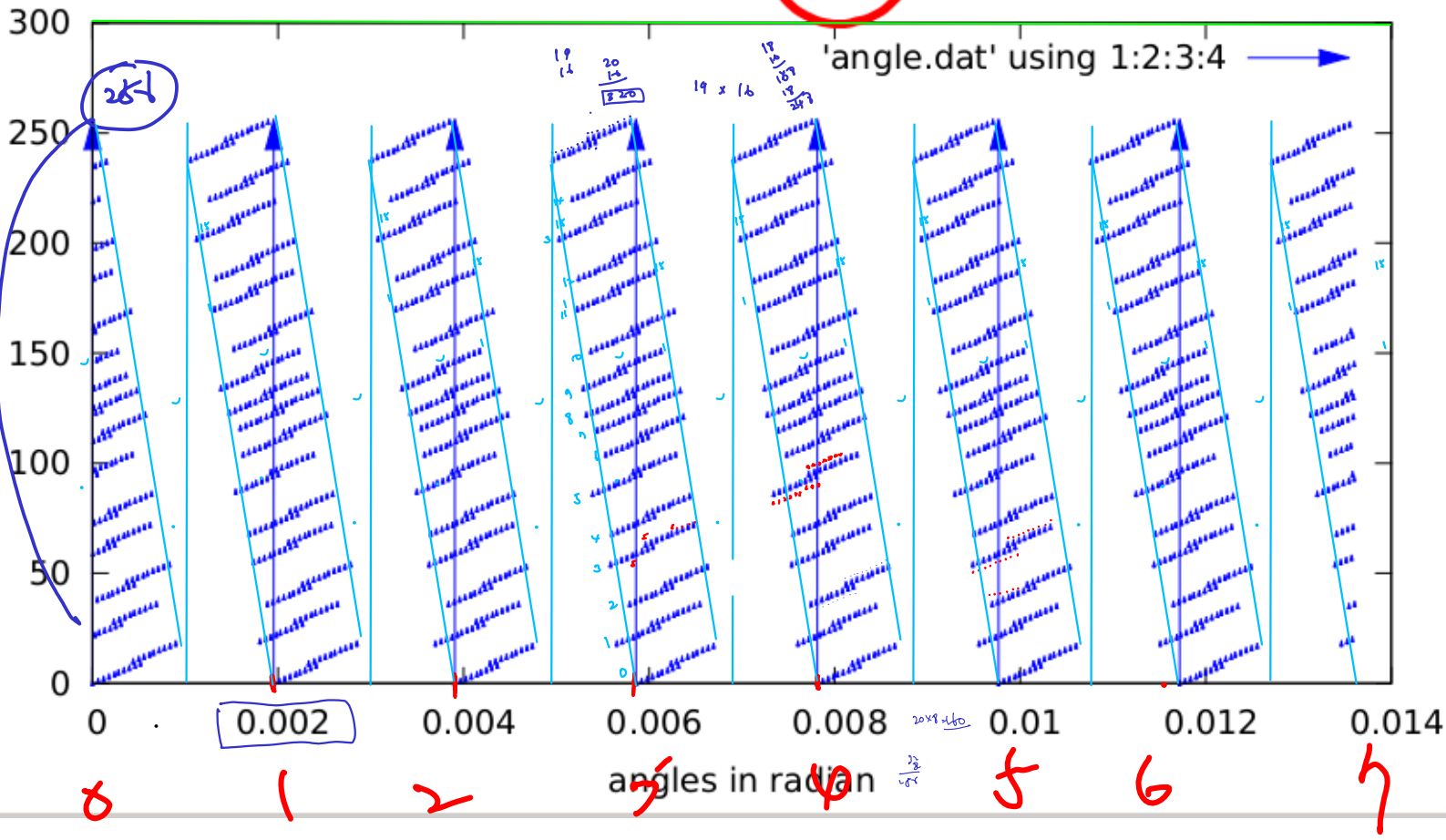
uneven distribution of angle points

there are 256 bins

$n \text{ Angles} / \text{binnum} = 2^{11} / 2^8 = 2^3 \Rightarrow$  average number of angle points per bin



Linear angle vectors showing jitter (Leaf\_11 n2048)



there are 256 (=binnum) bins

each bin covers the step angle

each bin contains  $n \text{ Angles} / \text{binnum} = 2^3 = 8$  angle points

if we view the average angle spacing as 0.002

the angle spacing drifts from  $[0.002 - 0.001, 0.002 + 0.001]$

$[0.001, 0.003]$

```

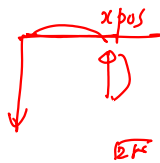
// int nPoints = getnAngles();
// double ang = get_min_angle();
// double rng = get_max_angle() - get_min_angle();
double binnum = 256;
double step = (B[nAngles-1] - B[0]) / binnum;
double ang = 0.0;
double xpos;
int hpos;

printf("* max=%f \n", B[0]);
printf("* min=%f \n", B[nAngles-1]);
printf("* step=%f \n", step);

```

```
ofstream myout;
```

```
myout.open("angle.dat");
```



bin size = 256



```

for (int i=0; i<nAngles; ++i) {
    ang = B[i] - B[0]; // relative angle
    hpos = int(ang / step); // which bin
    xpos = fmod(ang, step); // position within each bin
    myout << scientific << xpos << " " << hpos << " 0.0 1.0" << endl;

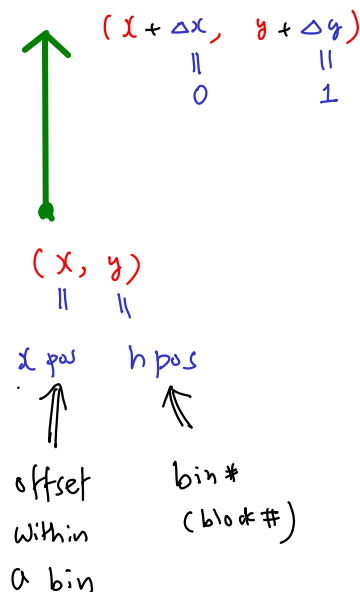
    if (hpos == 0) {
        myout << scientific << xpos << " " << hpos << " 0.0 " << binnum << endl;
    }
}

```

$$\frac{2048}{256} = 8 \text{ bins}$$

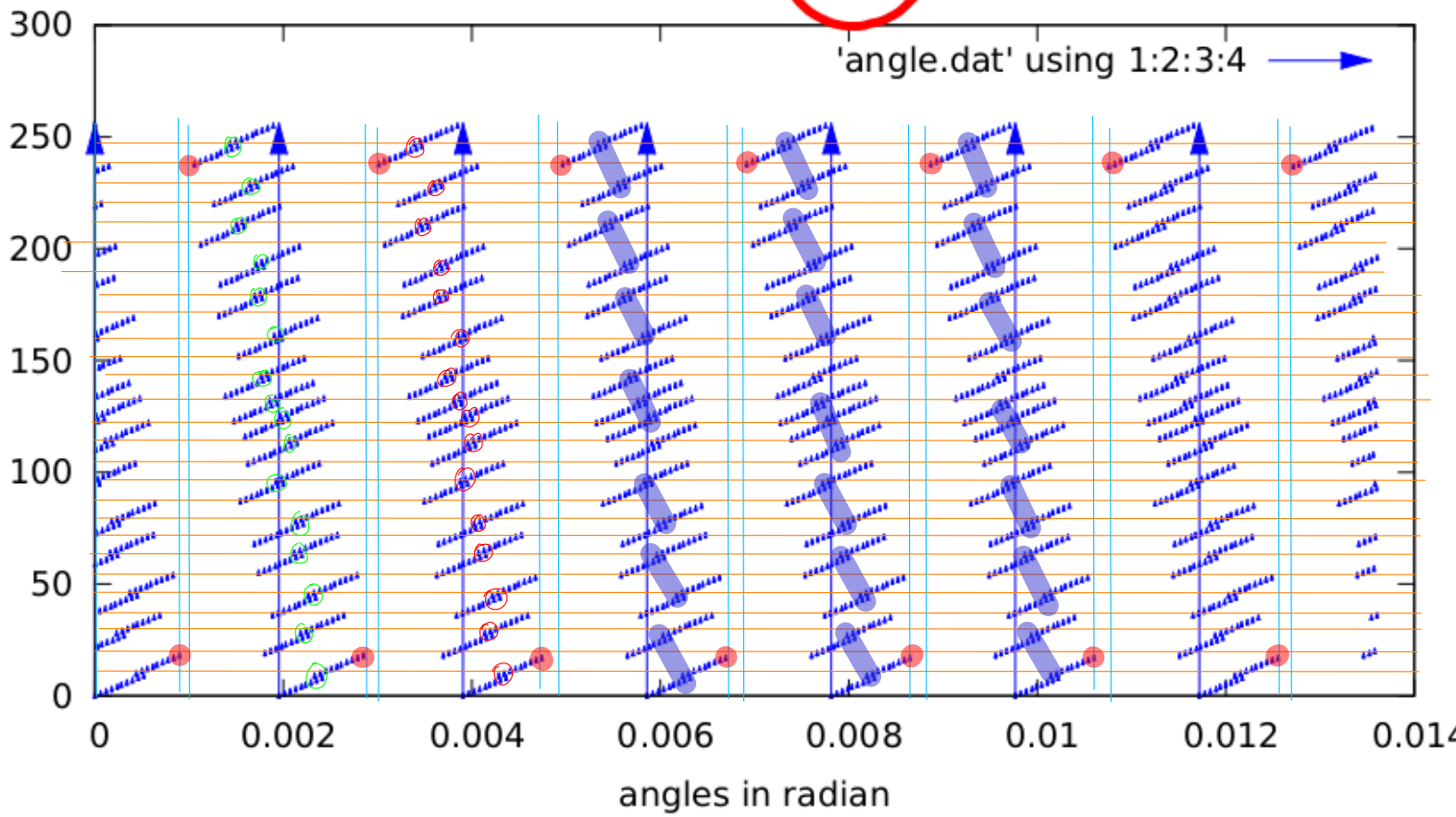
$$hpos = \text{int}(ang / \text{step})$$

$$xpos = \text{fmod}(ang, \text{step})$$



height position

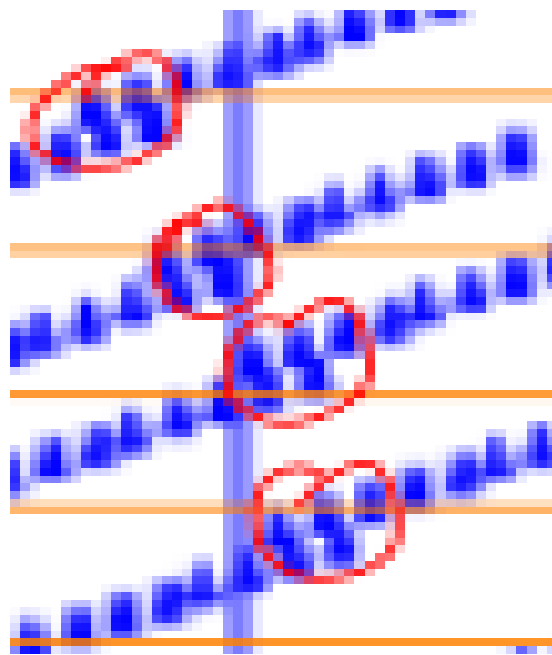
Linear angle vectors showing jitter (Leaf\_11 n2048)



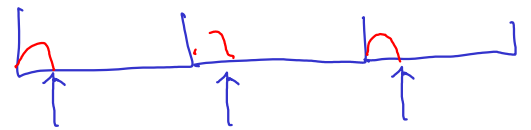
① change binnum

64	28	256	512	1024
32	16	8	4	

② What is this overlapping?

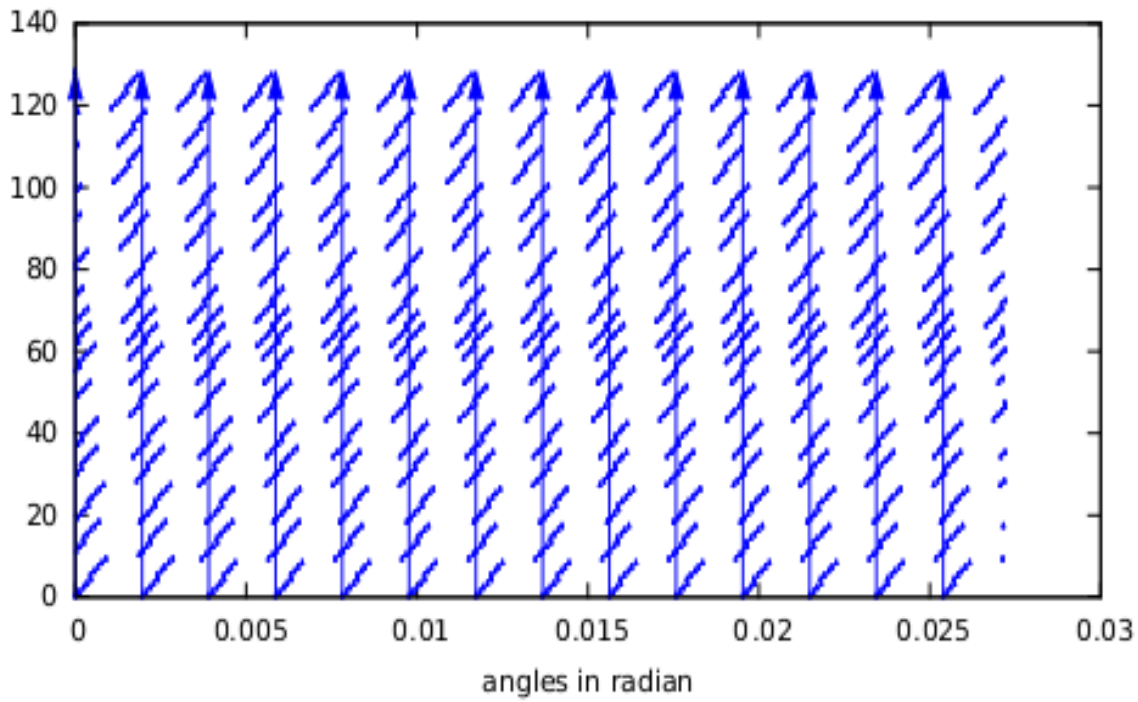


← different  
 ← bin number  
 ←

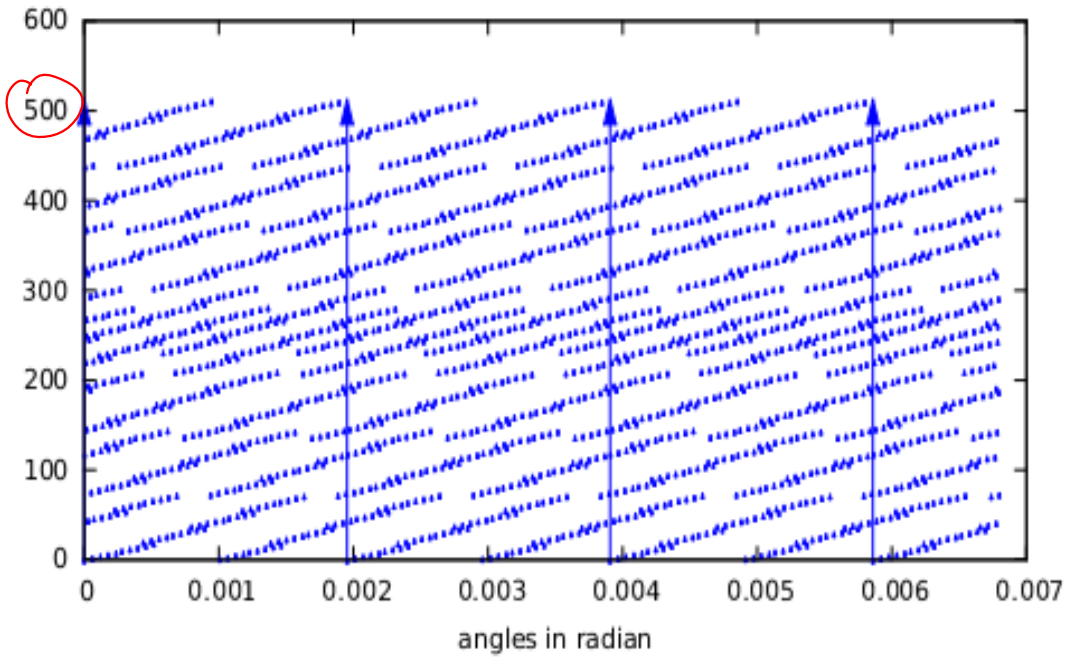


*binnum=256*

Linear angle vectors showing jitter (Leaf\_11 n2048)

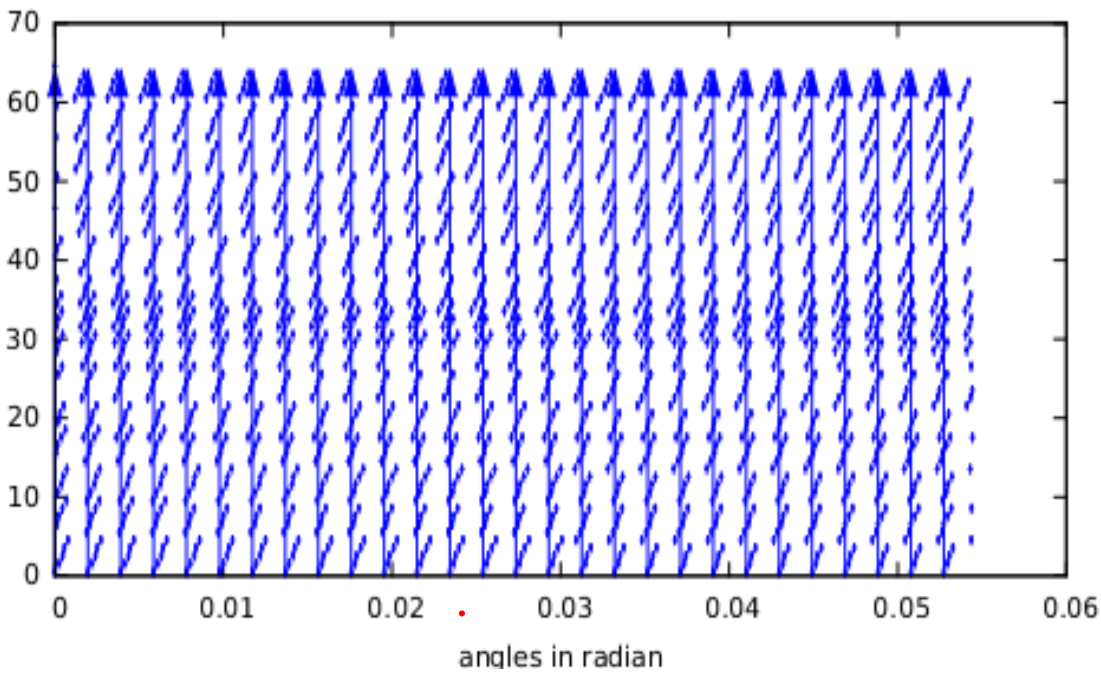


Linear angle vectors showing jitter (Leaf\_11 n2048)



*binnum = 512*

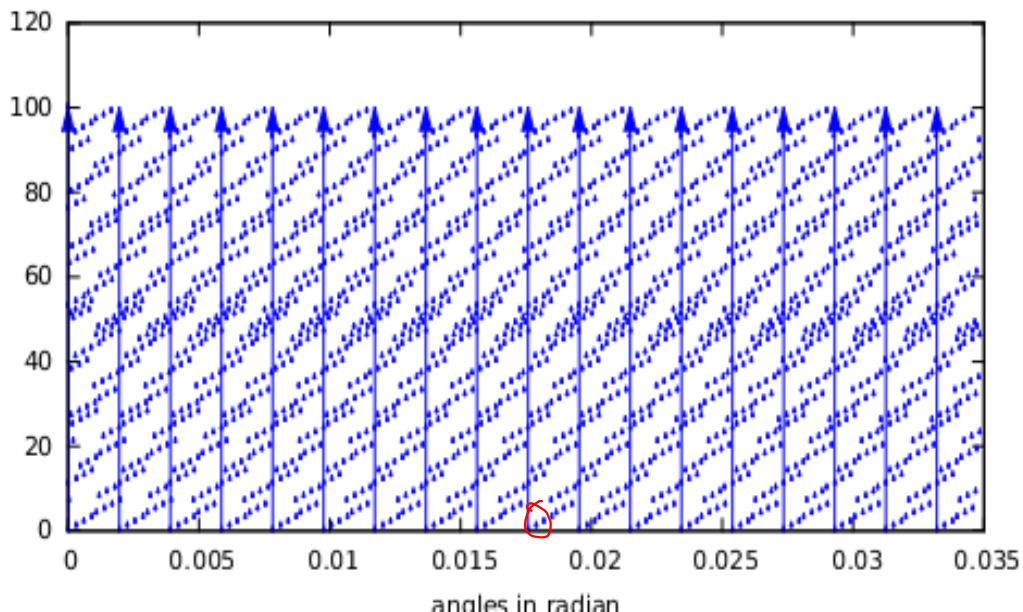
Linear angle vectors showing jitter (Leaf\_11 n2048)



*binnum = 64*

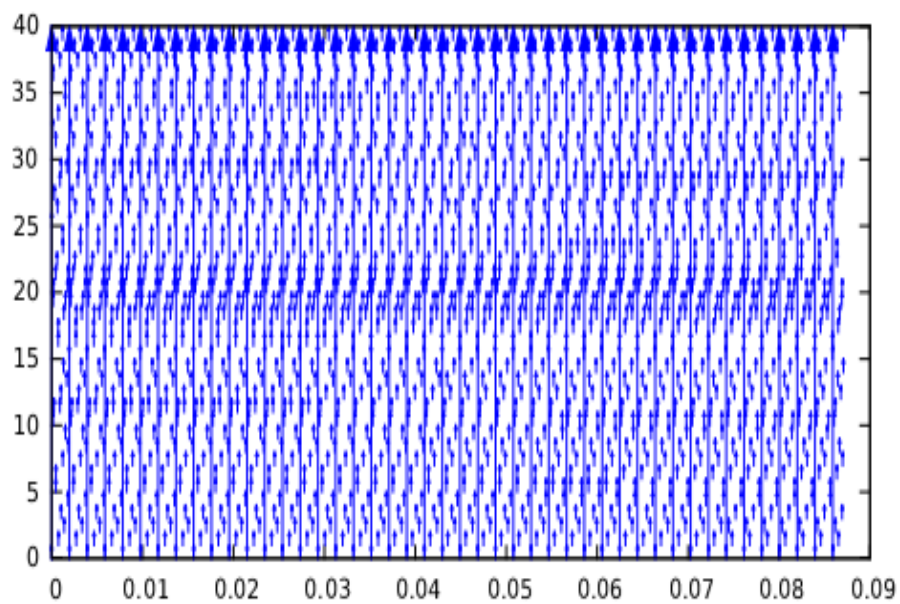
Linear angle vectors showing jitter (Leaf\_11 n2048)

*binnum = 100*

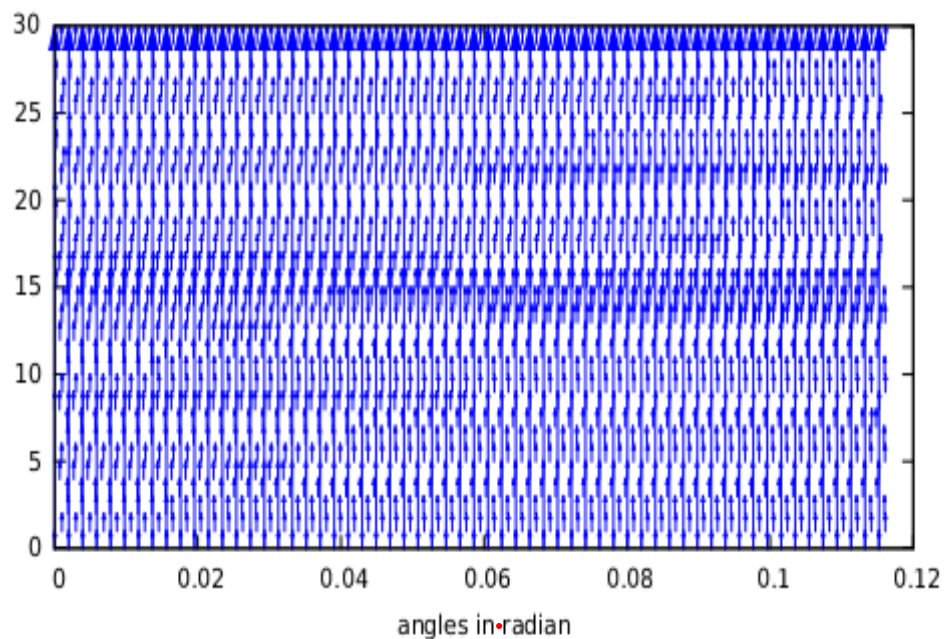




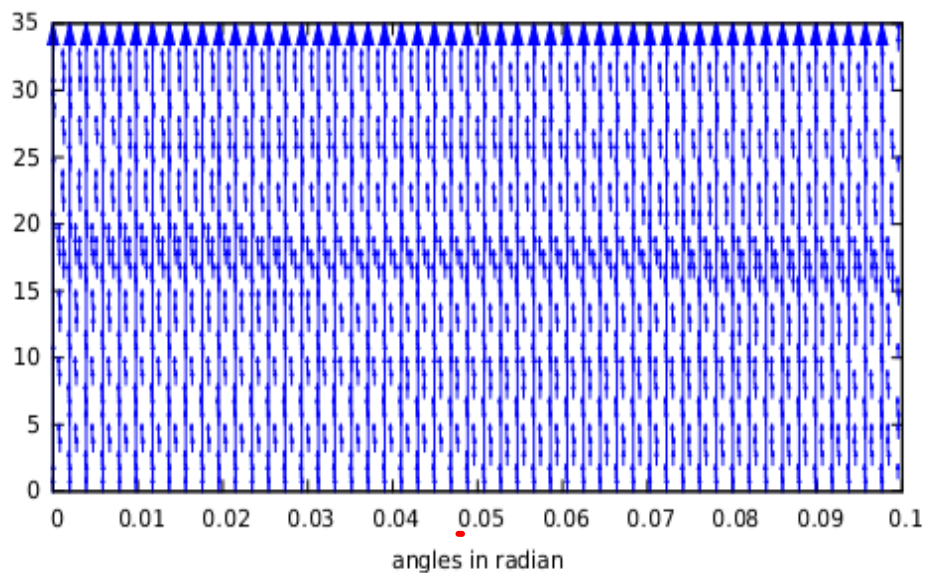
Linear angle vectors showing jitter (Leaf\_11 n2048)



Linear angle vectors showing jitter (Leaf\_11 n2048)



Linear angle vectors showing jitter (Leaf\_11 n2048)



Linear angle vectors showing jitter (Leaf\_11 n2048)

