Wide-field Imaging

Anna Scaife - Jodrell Bank Centre for Astrophysics

There are multiple effects that become important when making very wide-field interferometric images: non-coplanarity, primary beam rotation, direction-dependent phase calibration…

In this lecture we will focus on the first of these: **non-coplanarity,** otherwise known as **the w-effect.**

The structure of the lecture will be,

(i) an overview of the w-effect,
(ii) various solutions to the w-effect, and
(iii) worked examples.

The spatial coherence function measured by a radio interferometer is related to the spectral intensity as,

$$w(\sqrt{1-l^2-m^2}-1)$$

I(l,m) is the spectral intensity

$$V(u,v,w) = \int \frac{I(l,m)}{\sqrt{1-l^2-m^2}} \mathrm{e}^{j2\pi[ul+vm+w(\sqrt{1-l^2-m^2}-1)]}\,\mathrm{d}l\mathrm{d}m$$

(u,v,w) is the vector between the two interferometer elements expressed in units of wavelength

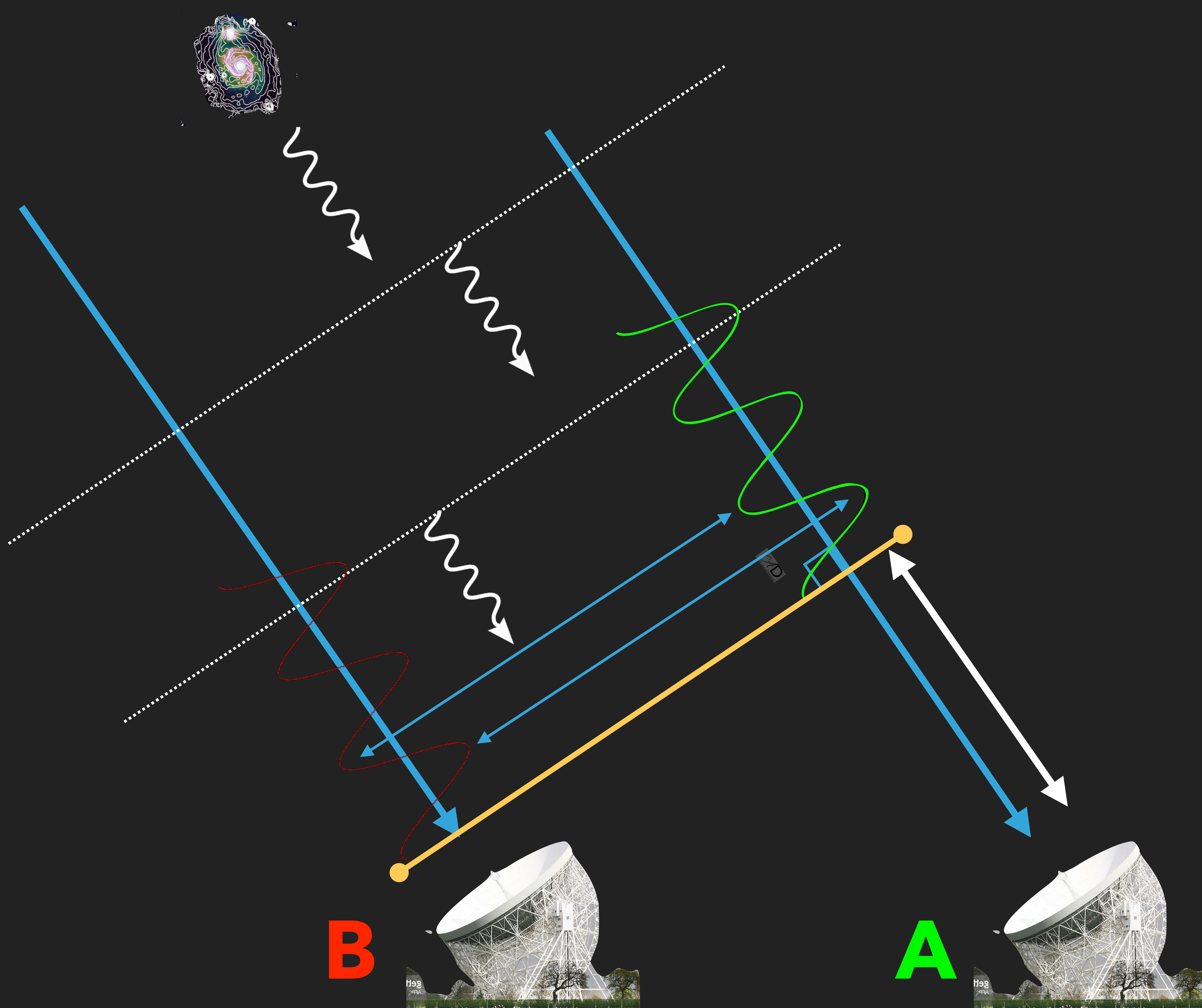
$$\boxed{w(\sqrt{1 - l^2 - m^2} - 1)}$$

If $2\pi w(\sqrt{1 - l^2 - m^2} - 1) \ll 1$ then,

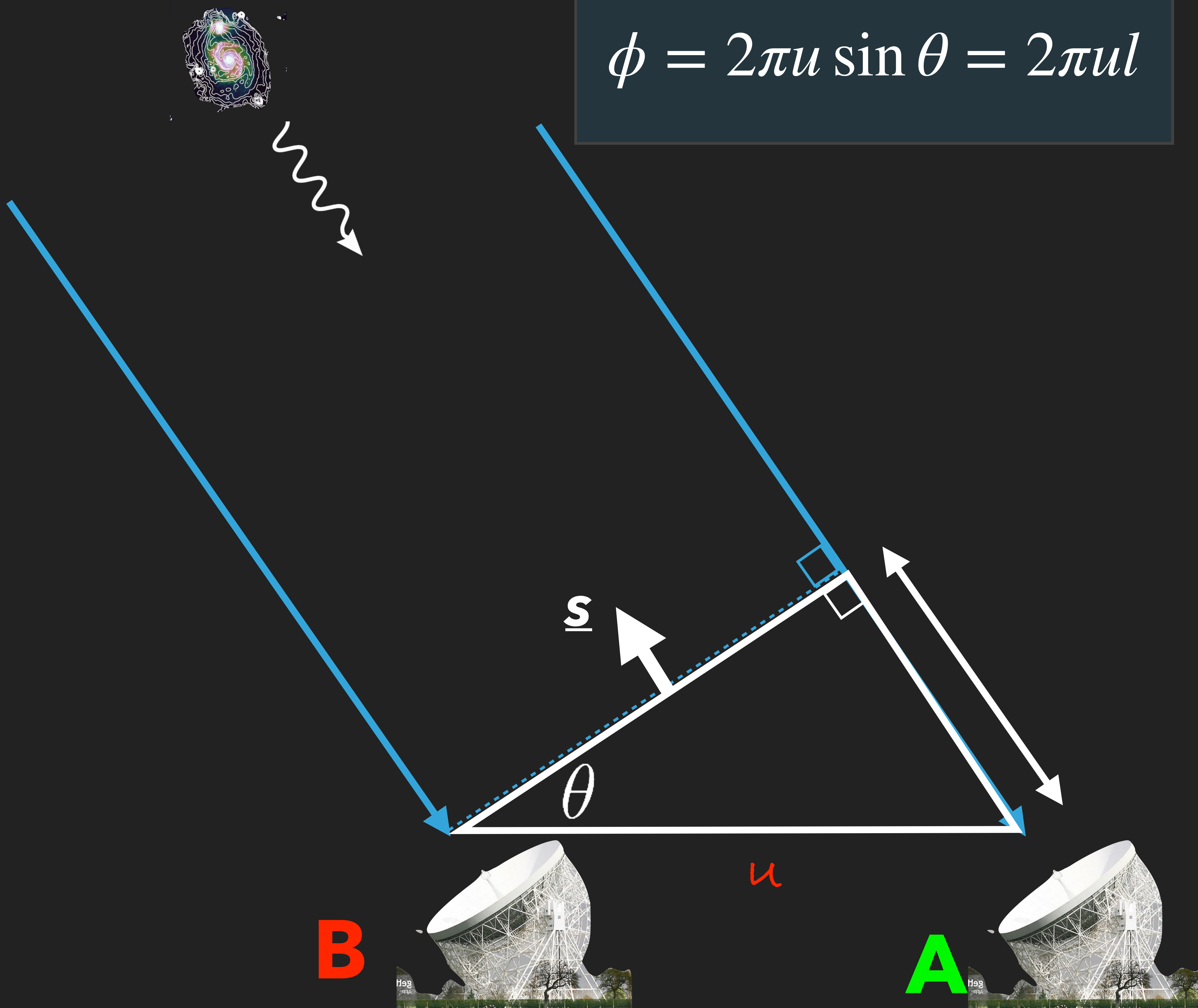
$$V(u, v) = \int \frac{I(l, m)}{\sqrt{1 - l^2 - m^2}} \mathrm{e}^{j2\pi[ul + vm]} \, \mathrm{d}l \, \mathrm{d}m$$

…just a 2d Fourier transform.

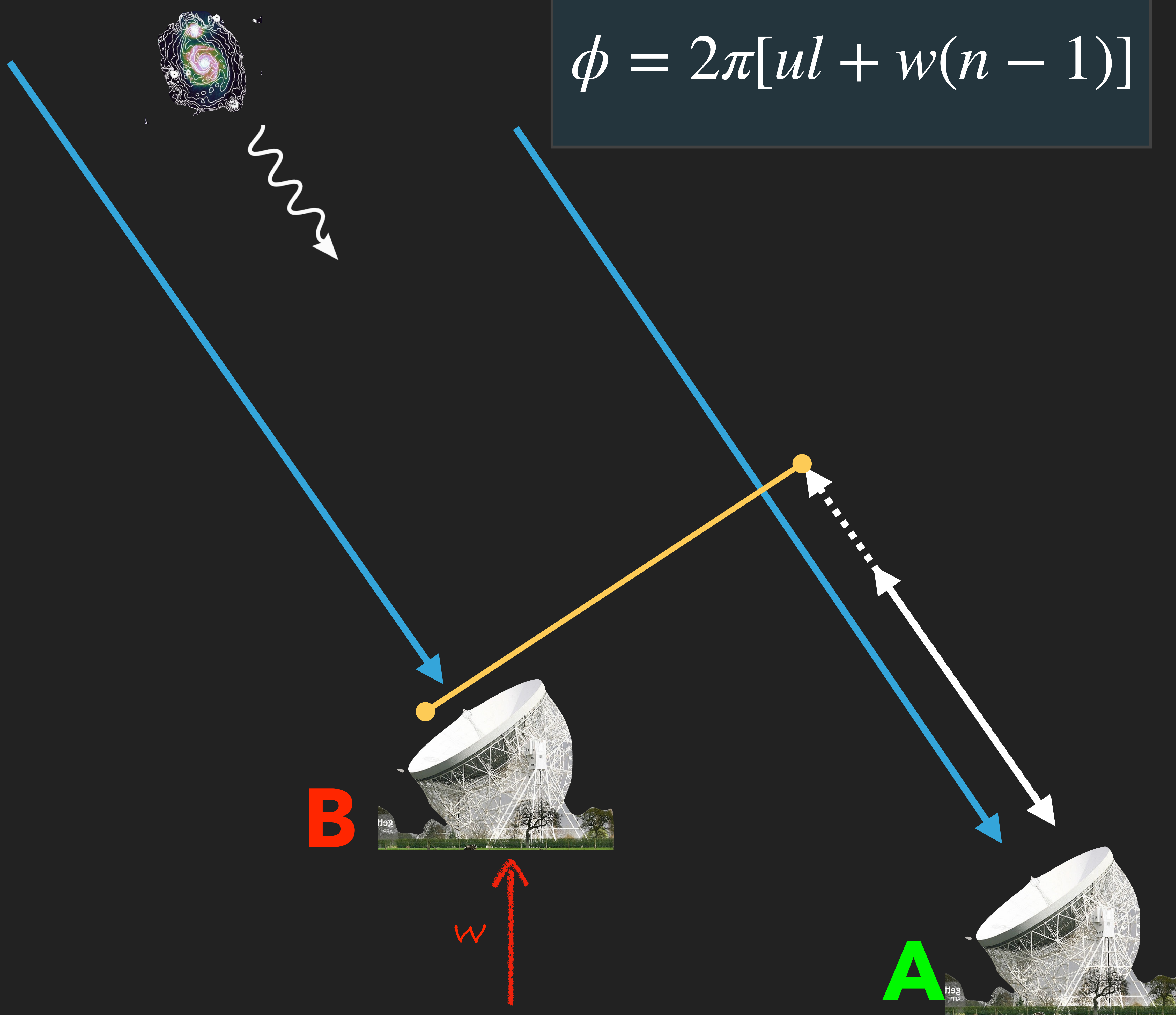
But if $2\pi w(\sqrt{1 - l^2 - m^2} - 1) \geq 1$ then we can't really use a 2d Fourier transform.


B
A

$$\phi = 2\pi u \sin\theta = 2\pi u l$$



$$\phi = 2\pi[ul + w(n-1)]$$



What happens if you use a 2d Fourier transform?

$$V(u,v,w) = \int \frac{I(l,m)}{\sqrt{1-l^2-m^2}} \mathrm{e}^{j2\pi[ul+vm]}\, \underbrace{\mathrm{e}^{j2\pi w(\sqrt{1-l^2-m^2}-1)]}}_{\mathrm{e}^{j\phi}}\, \mathrm{d}l\mathrm{d}m$$

The *w*-term is a phase term… but the phase is *position dependent*, i.e.

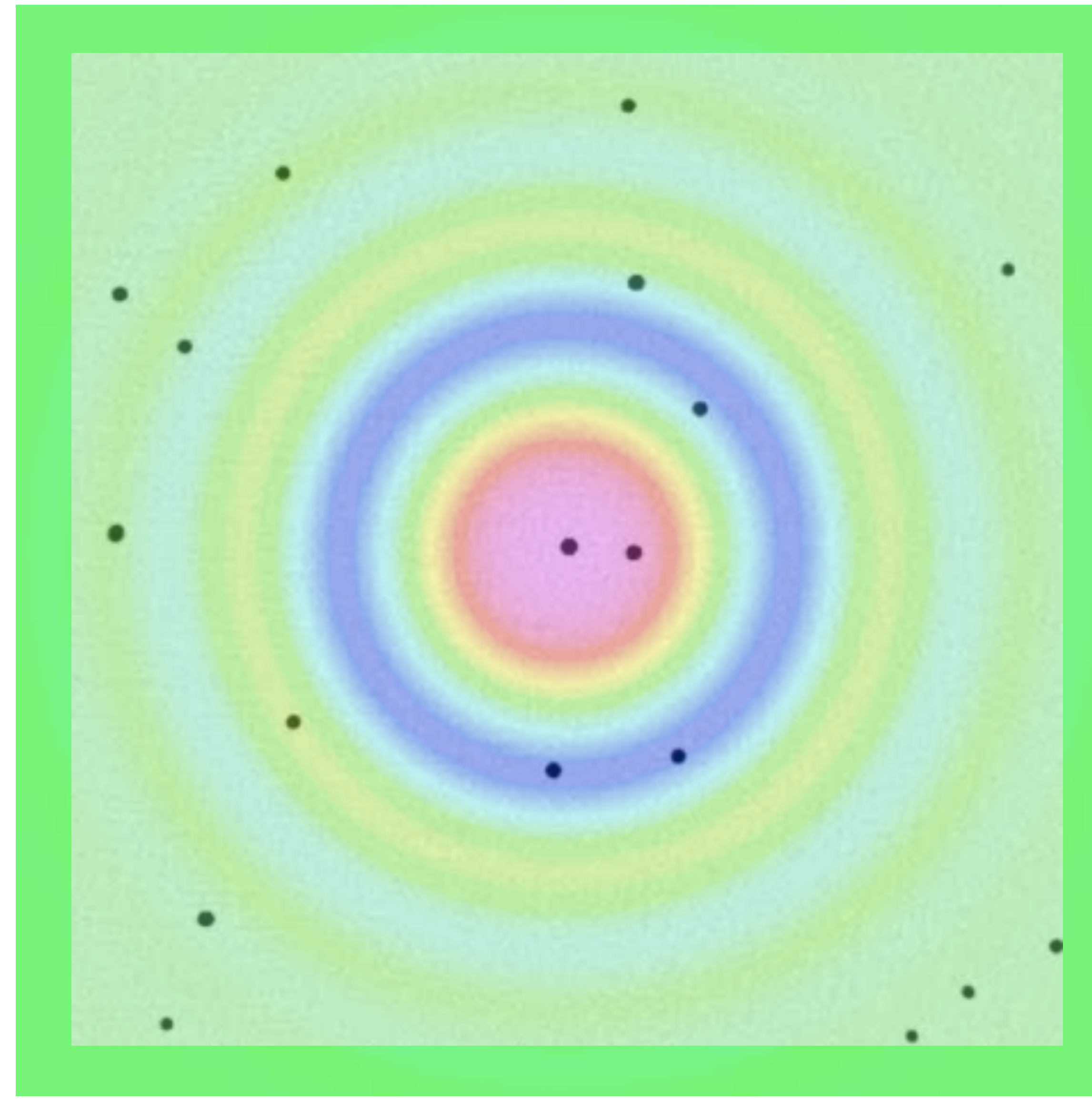
$$\phi(l,m,w)$$

The w-phase-screen

Effectively, every visibility with a different w-value sees a different sky.

The w-value changes as a function of time - so the measured sky is time variable.

$$\phi(l, m, w)$$

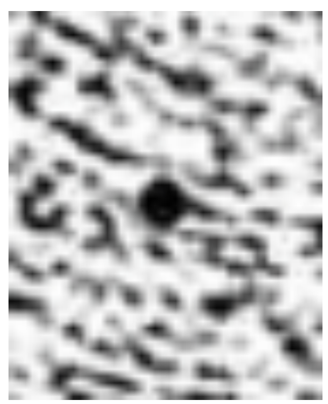


Cornwell, Golap & Bhatnagar, EVLA Memo 67

[In the image the w-kernel is also multiplied by the image plane counterpart of the anti-aliasing function.]

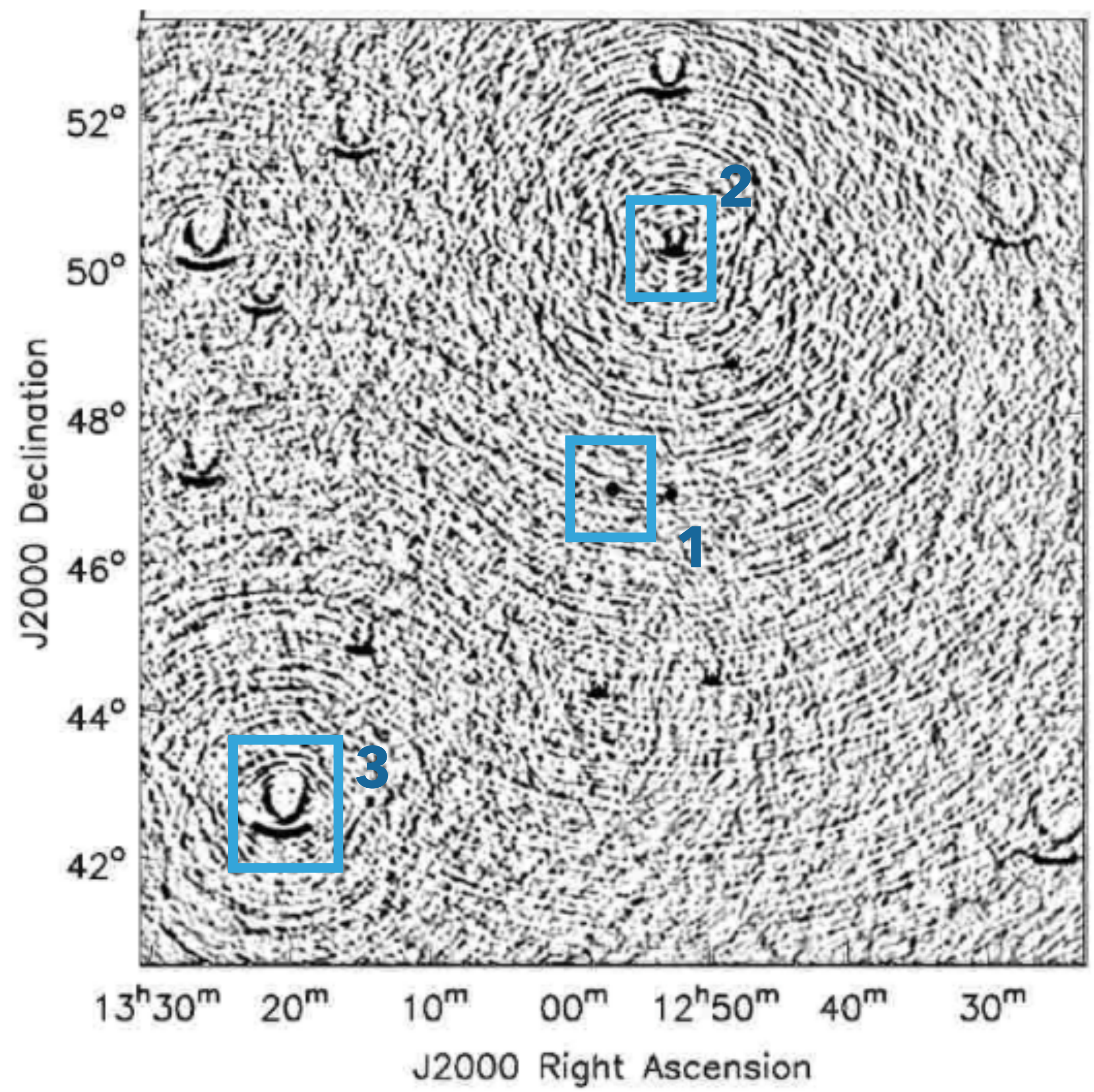
If we ignore the w-term and do a 2d transform we will see image distortions

2



1

3



$$|\Delta\phi| = |2\pi w(\sqrt{1 - l^2 - m^2} - 1)|$$


2


1


3

As we get further from the centre of the field the sources become more distorted

Any array with North-South extensions will have non-zero w-terms


VLA telescope

Why are the sources distorted like that?

$$V(u,v,w) = \int \frac{I(l,m)}{\sqrt{1-l^2-m^2}} e^{j2\pi[ul+vm+w(\sqrt{1-l^2-m^2}-1)]}\, \mathrm{d}l\mathrm{d}m$$

$$w = u \sin\chi \tan Z - v \cos\chi \tan Z$$

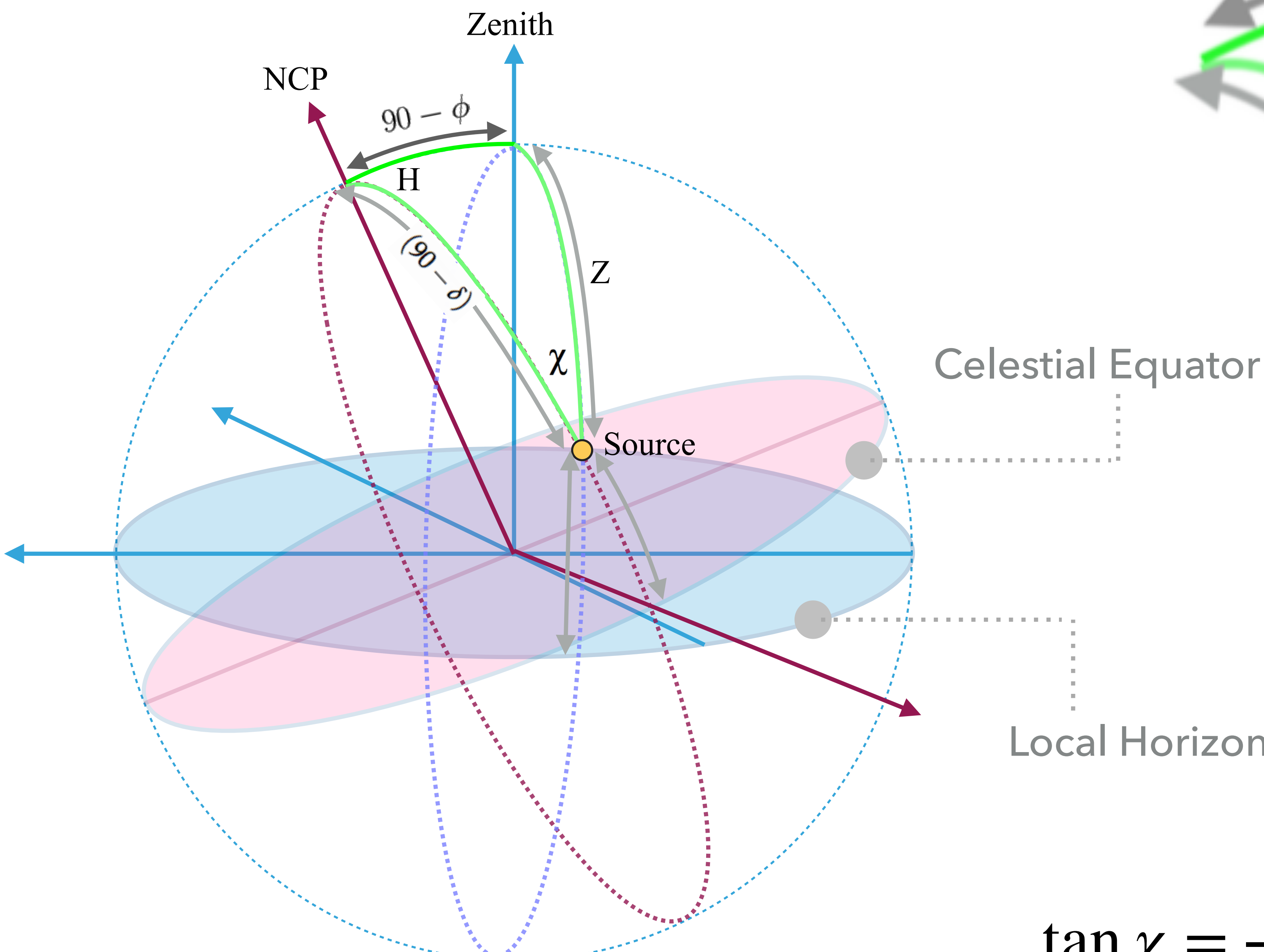
This can be re-written as,

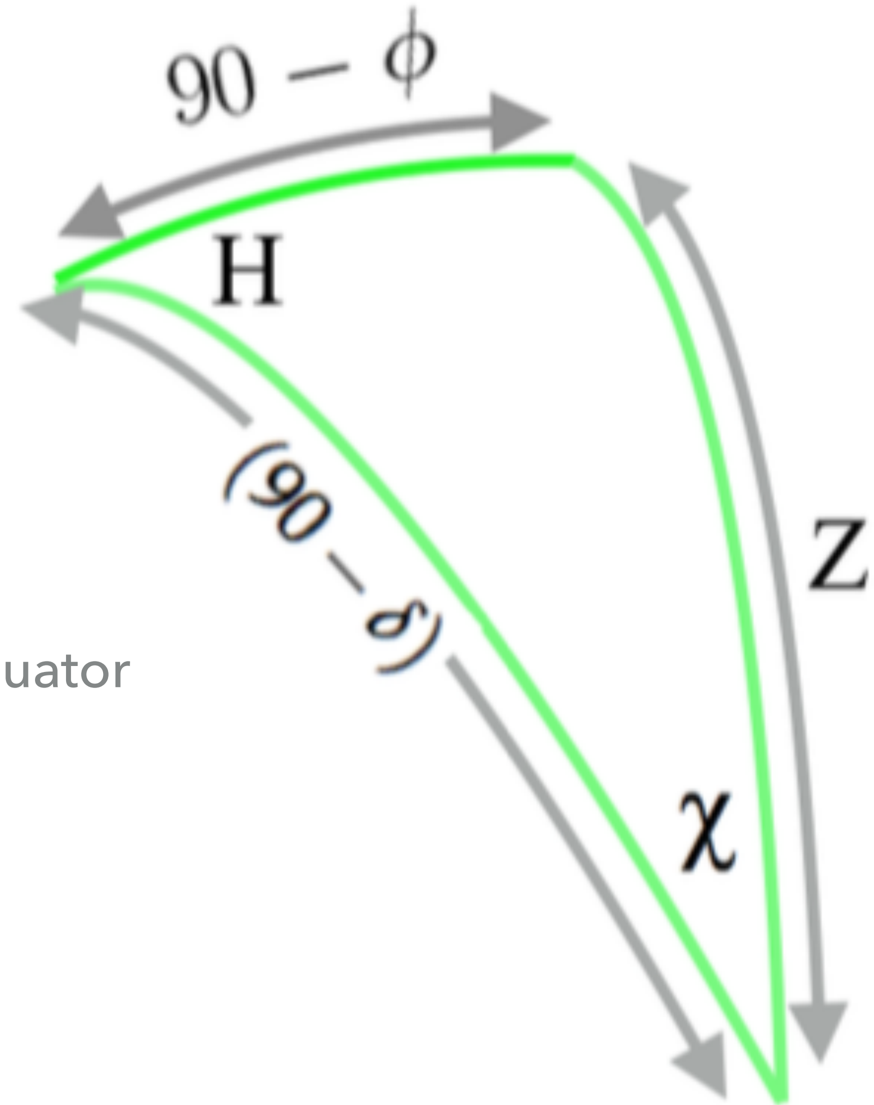
$$V(u,v,w) = \int \frac{I(l,m)}{\sqrt{1-l^2-m^2}} e^{j2\pi[ul'+vm']}\, \mathrm{d}l\mathrm{d}m$$

$$l' = l + \sin\chi \tan Z(\sqrt{1-l^2-m^2} - 1)$$
$$m' = m - \cos\chi \tan Z(\sqrt{1-l^2-m^2} - 1)$$

χ = Parallactic angle
Z = Zenith angle

Parallactic Angle & Zenith Angle





$$\tan\chi = \frac{\sin H}{\cos\delta\tan\phi - \sin\delta\cos H}$$

$$\cos Z = \sin\phi\sin\delta + \cos\phi\cos\delta\cos H$$

H = Hour angle
δ = Declination
ϕ = Latitude of array

(l',m') is a **warped** version of *(l,m)*

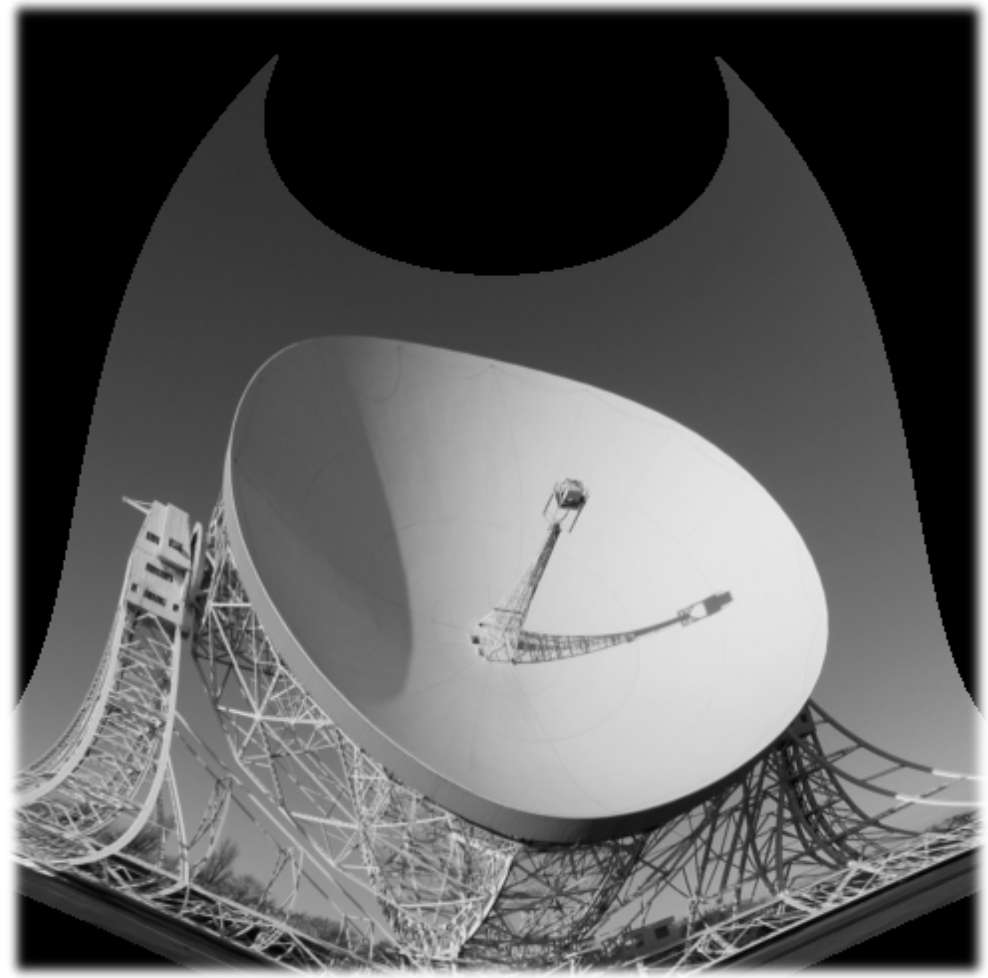


Image credit: Chris Skipper

$$l' = l + \sin\chi \tan Z\left(\sqrt{1 - l^2 - m^2} - 1\right)$$
$$m' = m - \cos\chi \tan Z\left(\sqrt{1 - l^2 - m^2} - 1\right)$$

On the 2d image plane, source motions follow conic sections

(l',m') scales quadratically with offset from the centre, i.e. sources that are further away move more.

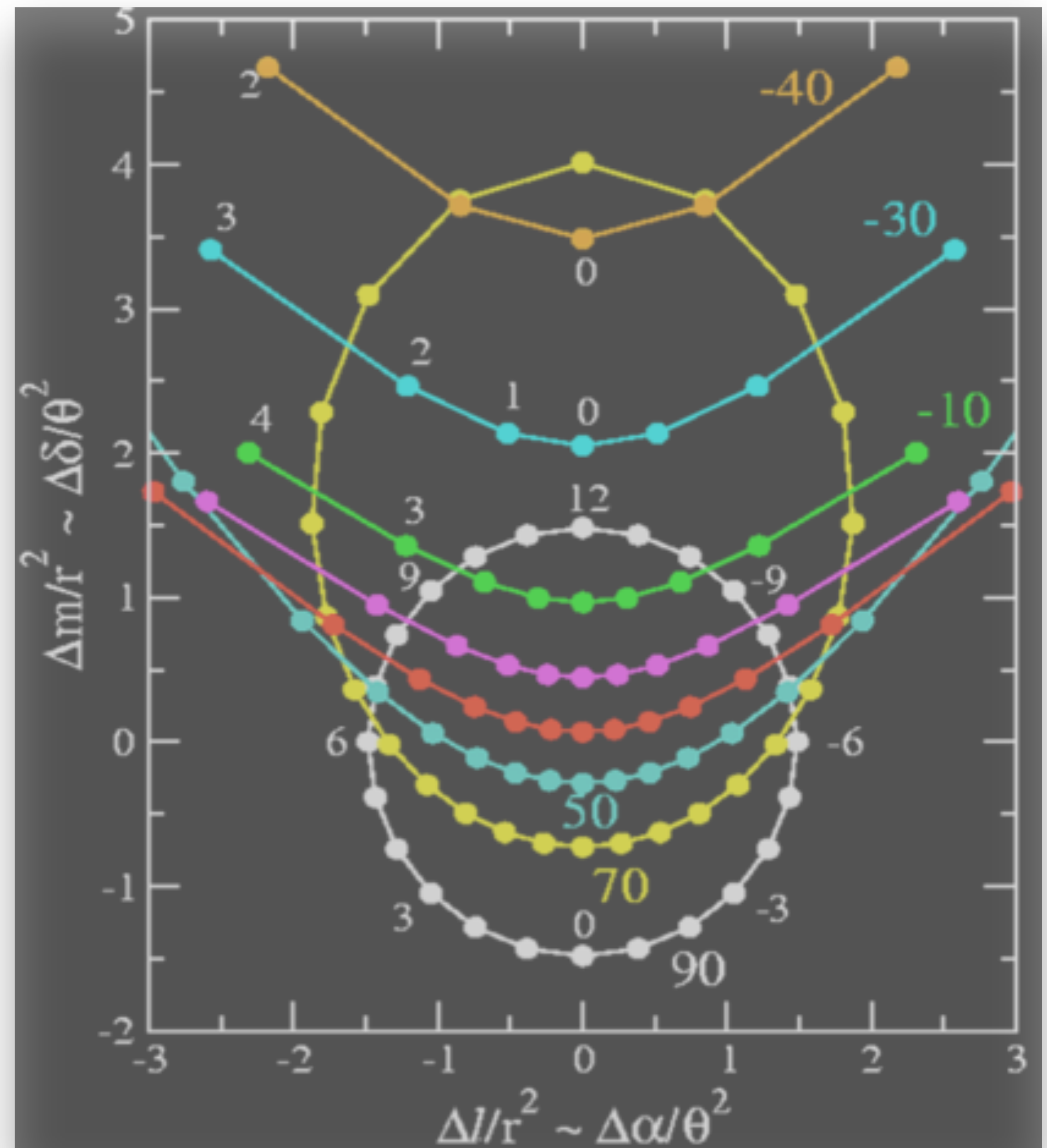


Image credit: Michael Bietenholtz

So how do we fix it?

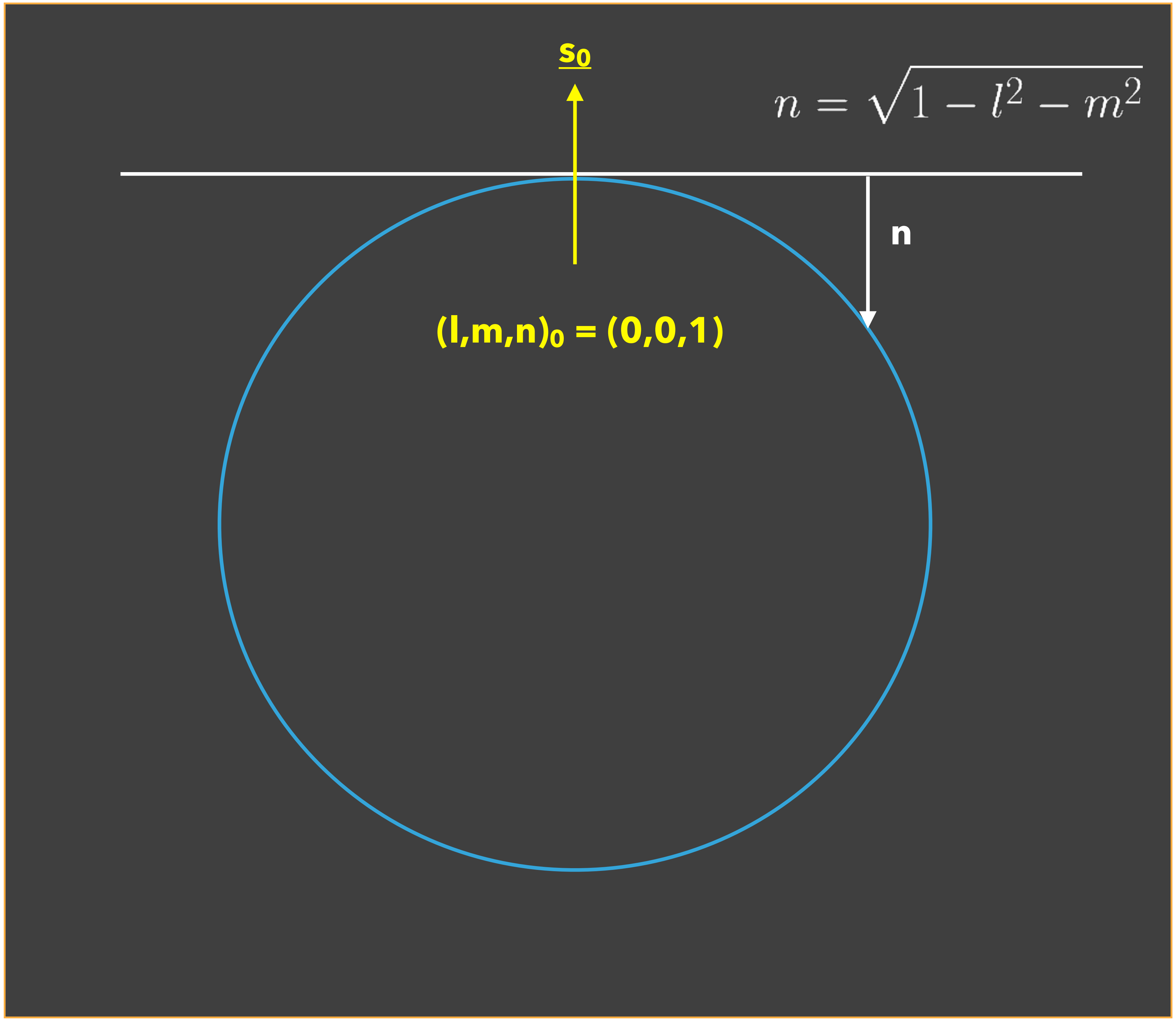
1. 3d Fourier transform
2. W-projection
3. Facetting

So how do we fix it?

If you take the 3d Fourier transform of V(u,v,w) then you recover F(l,m,n), where the only non-zero values lie on a 2d surface with

$$n = \sqrt{1 - l^2 - m^2}$$

B. Clark, VLA Memo 107, 1973



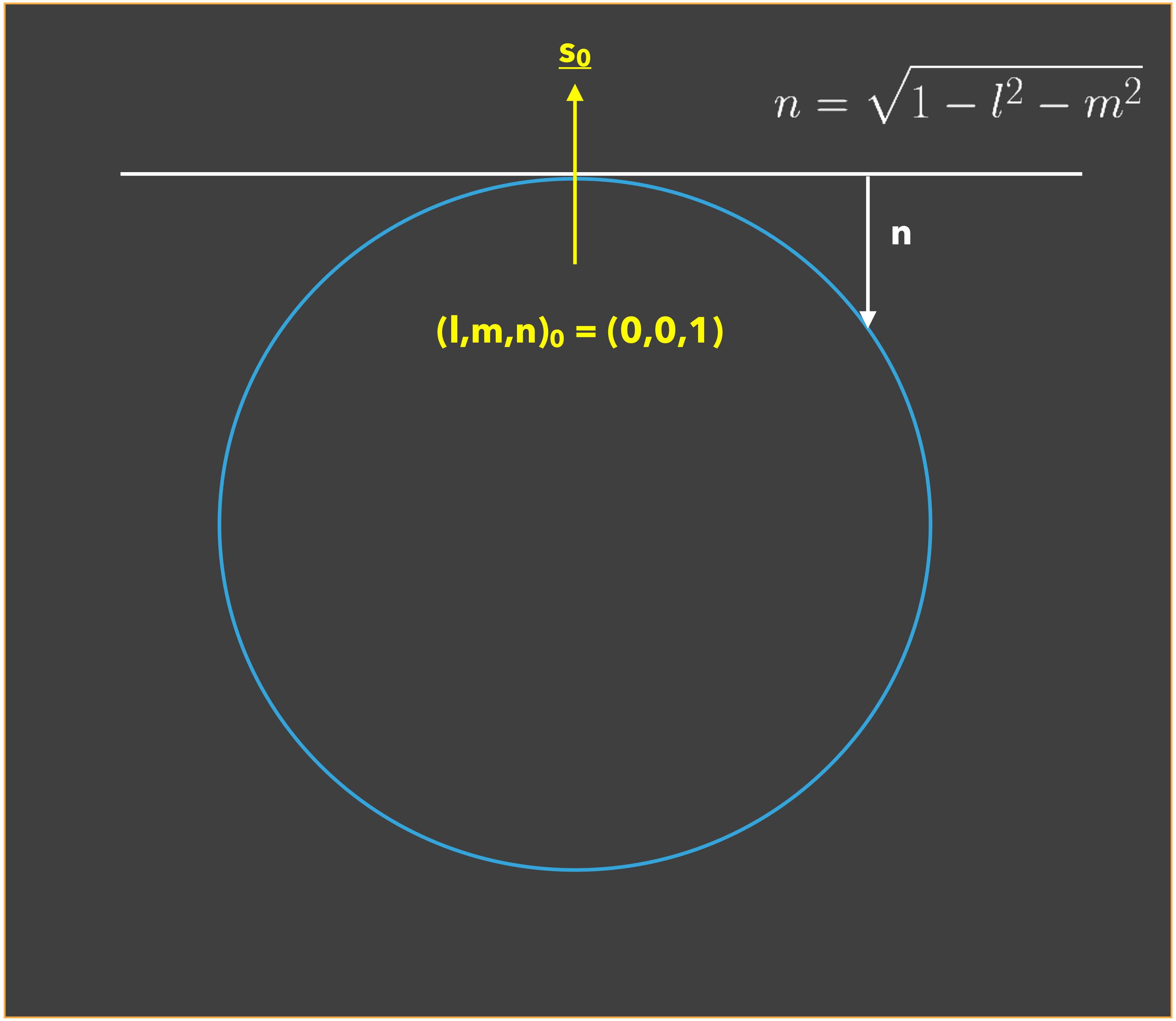
So how do we fix it?

We could do this, i.e. we could FFT in (u,v) and then DFT in (w), but… we'd end up with a cube where the n-direction was almost completely zero-valued.

Cornwell, Golap & Bhatnagar, EVLA Memo 67

Also, we do not have complete sampling in *(u,v,w)* - so *F(l,m,n)* will be convolved with a dirty beam in 3 dimensions.

Waldram & McGilchrist, 1990, MNRAS, 245, 532



$$V(u,v,w) = \int \frac{I(l,m)}{\sqrt{1-l^2-m^2}} \mathrm{e}^{j2\pi[ul+vm]} \, \boxed{\mathrm{e}^{j2\pi w(\sqrt{1-l^2-m^2}-1)]}} \, \mathrm{d}l\mathrm{d}m$$

The w-term is a phase screen multiplying the sky intensity distribution

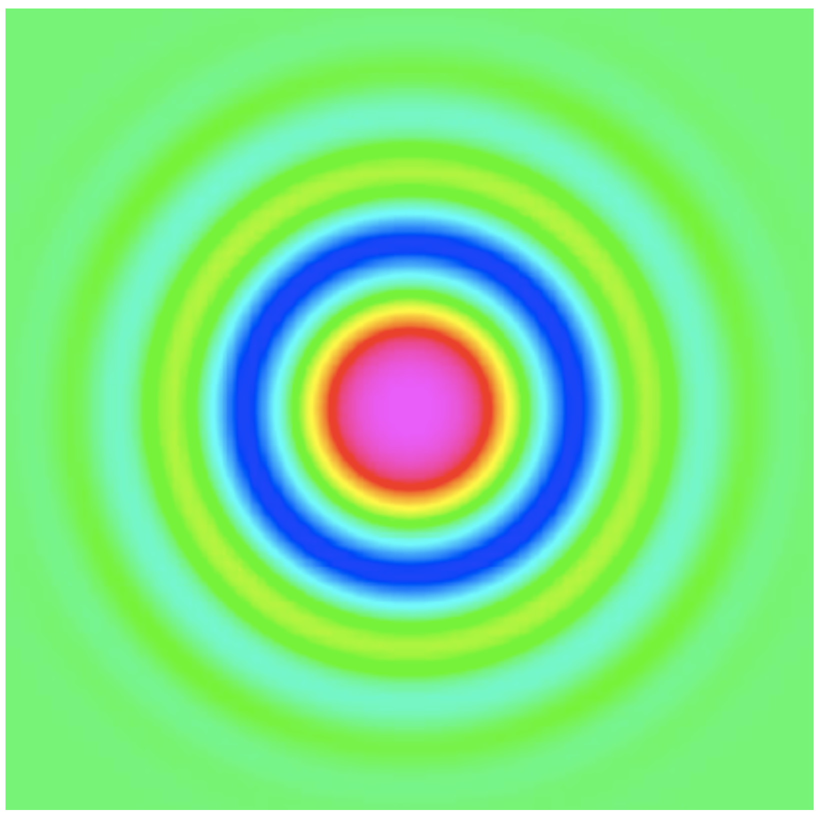
$$V(u,v,w) = \int \frac{I(l,m)}{\sqrt{1-l^2-m^2}} \boxed{G(l,m,w)} \, \mathrm{e}^{j2\pi[ul+vm]} \, \mathrm{d}l\mathrm{d}m$$

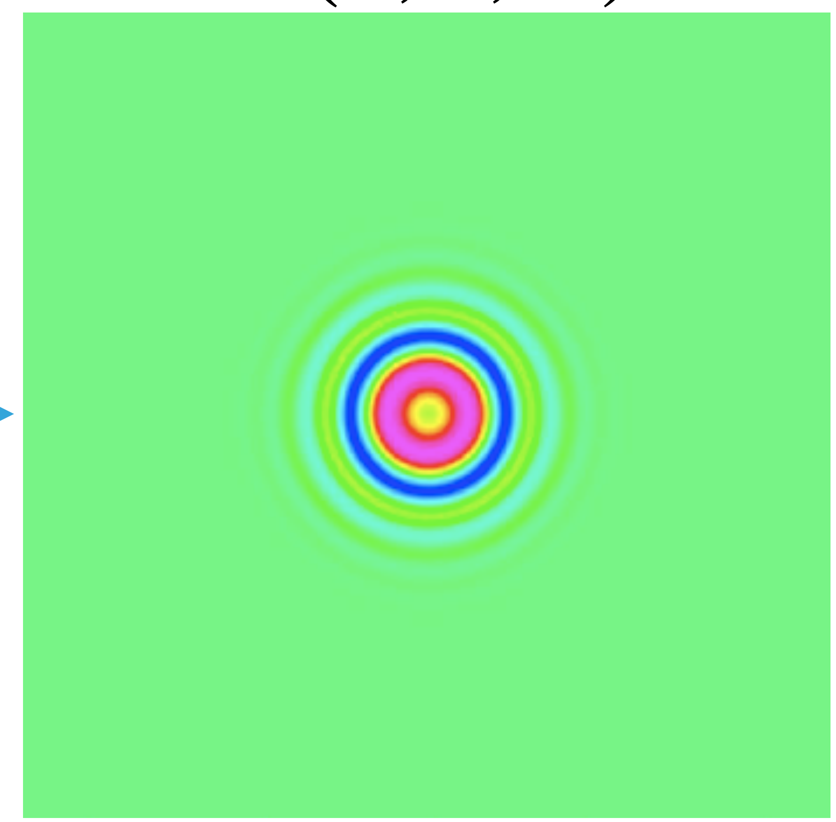
We can employ the convolution theorem to express this as a convolution in the visibility domain,

$$V(u,v,w) = \tilde{G}(u,v,w) \circledast V(u,v,w=0)$$

The convolution function is known as the **w-kernel**:

$$G(l, m, w) \qquad\qquad \tilde{G}(u, v, w)$$





$$(l, m) \qquad\qquad (u, v)$$

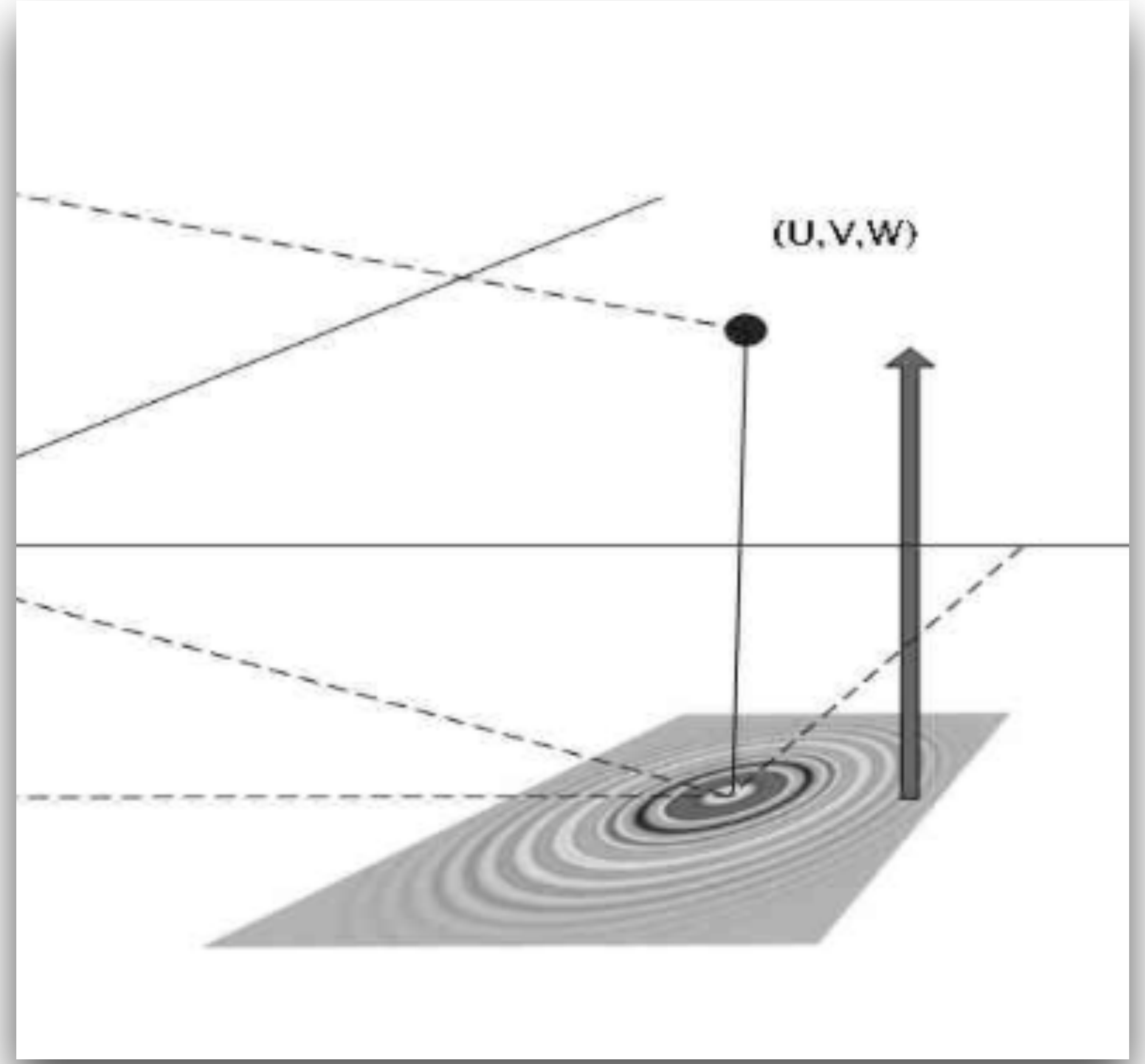
Cornwell, Golap & Bhatnagar, EVLA Memo 67

Mathematically this means that the *visibility* for non-zero w can be calculated from the visibility for w = 0.

$$V(u, v, w) = \tilde{G}(u, v, w) \circledast V(u, v, w = 0)$$

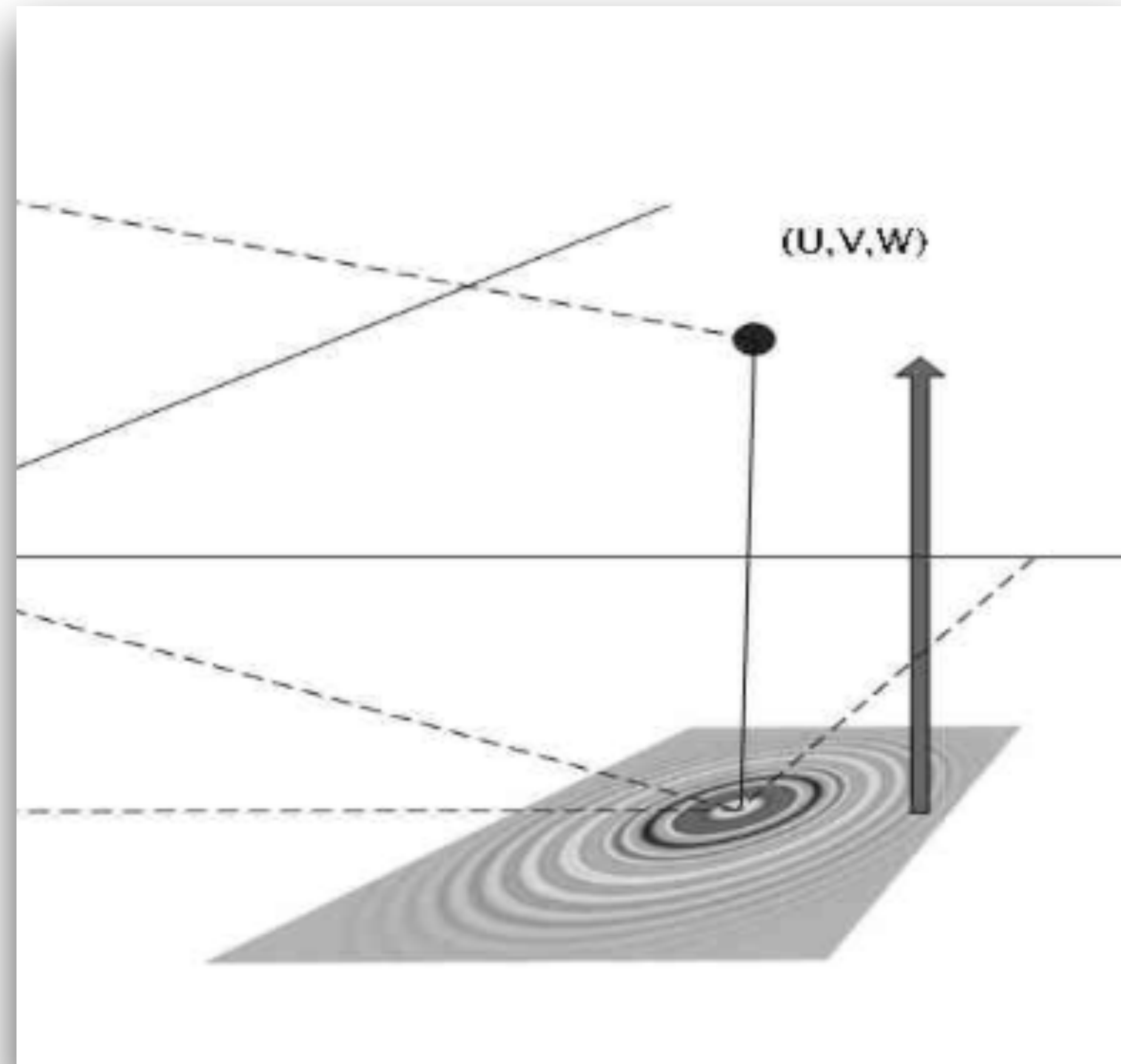
This is the same as projecting

$$V(u, v, w = 0) \longrightarrow V(u, v, w)$$



This is known as

w-projection



For making an image from visibilities we need to go the other way - from visibility data to images.

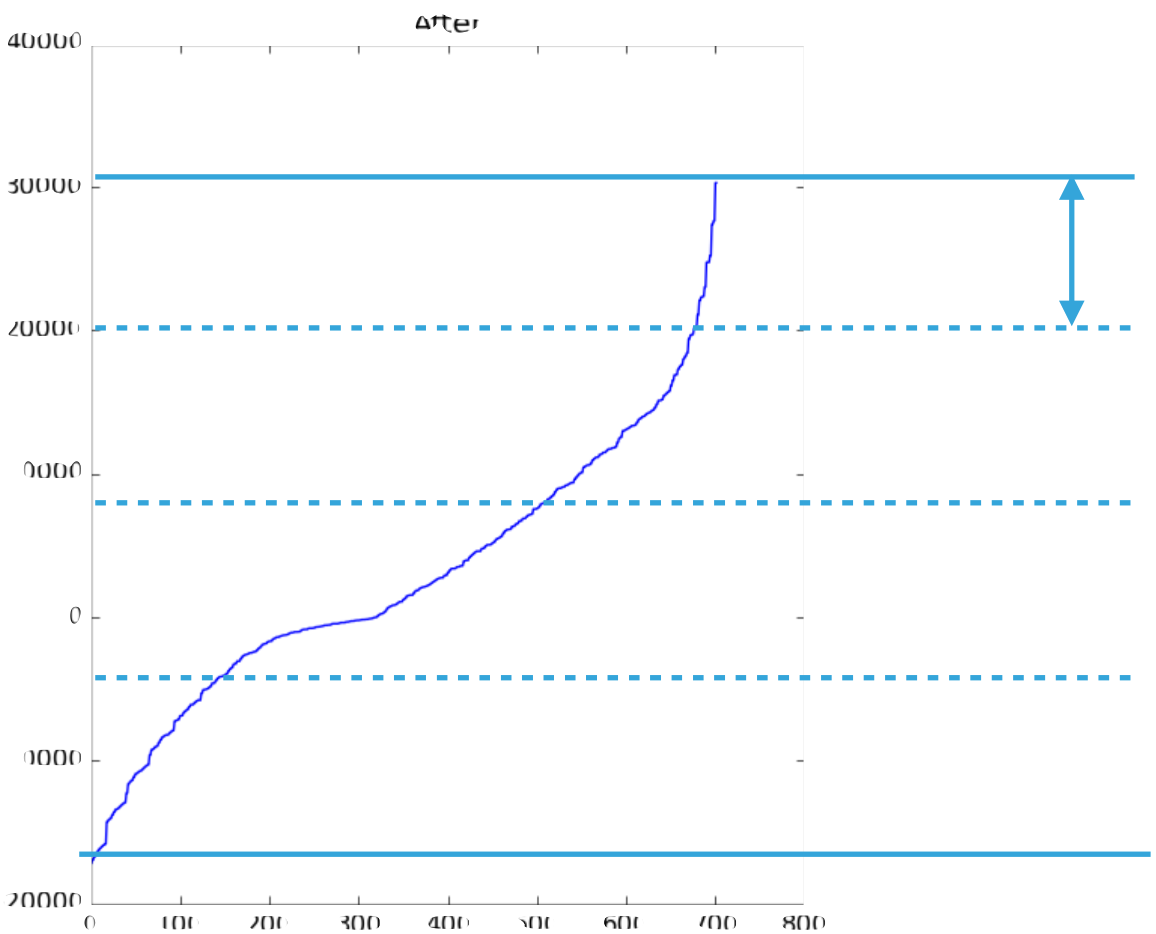
In this case we need to project

$$V(u, v, w) \longrightarrow V(u, v, w = 0)$$

in order to use the 2d Fourier Transform.

To do this we convolve *V(u,v,w)* with the inverse of *G(u,v,w)*, which conveniently is just *G(u,v,-w)*.

In practice most w-projection implementations do not calculate a w-kernel for every individual visibility.



Typically the visibility data is ordered in increasing w-value and then divided into *w-planes*.

A kernel is then created for each plane, with a w-value that represents the mean w-position within that plane.

The higher the number of w-planes, the more accurate the imaging will be - but also the more computationally expensive it will be…

If you are using CASA, the default number of w-planes is calculated using:

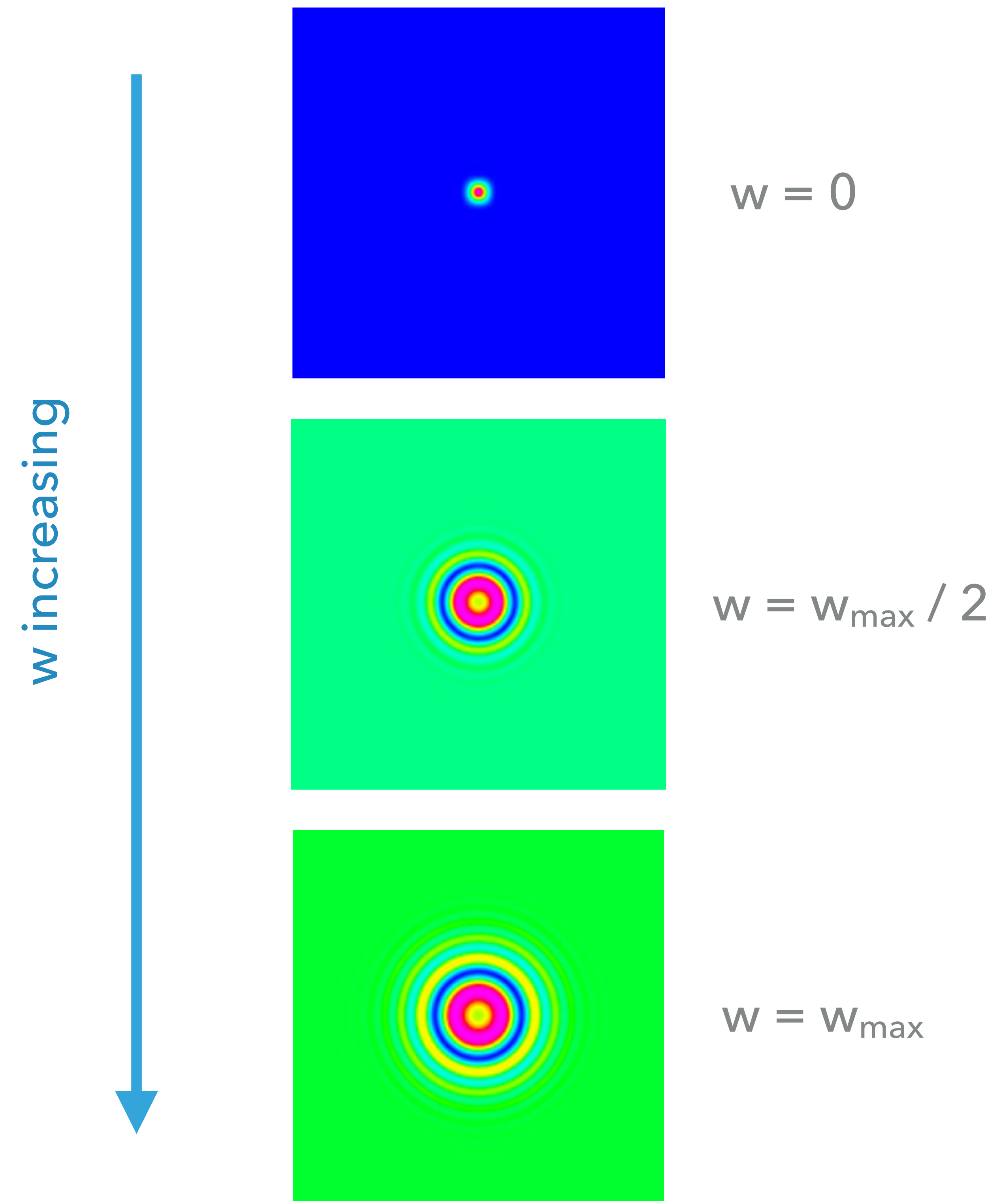
```
gridmode            =  'widefield'      #  Gridding kernel for FFT-based
                                        #   transforms, default='' None
     wprojplanes    =          -1       #  Number of w-projection planes for
                                        #   convolution; -1 => automatic
                                        #   determination
     facets         =           1       #  Number of facets along each axis
                                        #   (main image only)
```

$$N_w = B_{\mathrm{max}}(\mathrm{k}\lambda)\,\Omega(\mathrm{arcmin}^2)/600$$

As w increases, the size of the w-kernel also increases.

But, performing the convolution operation gets more expensive as the kernel gets larger.

CASA will look at the available memory on your computer and limit the size (“support”) of your w-kernels.

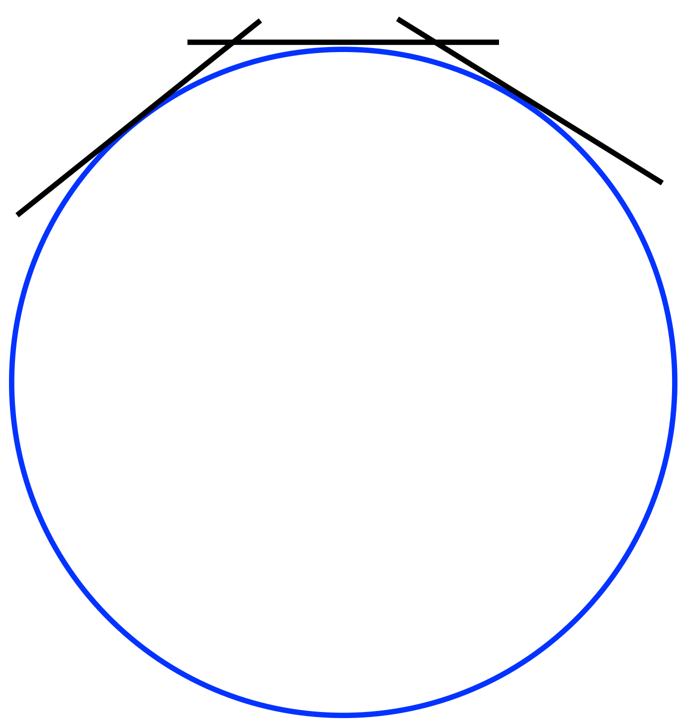


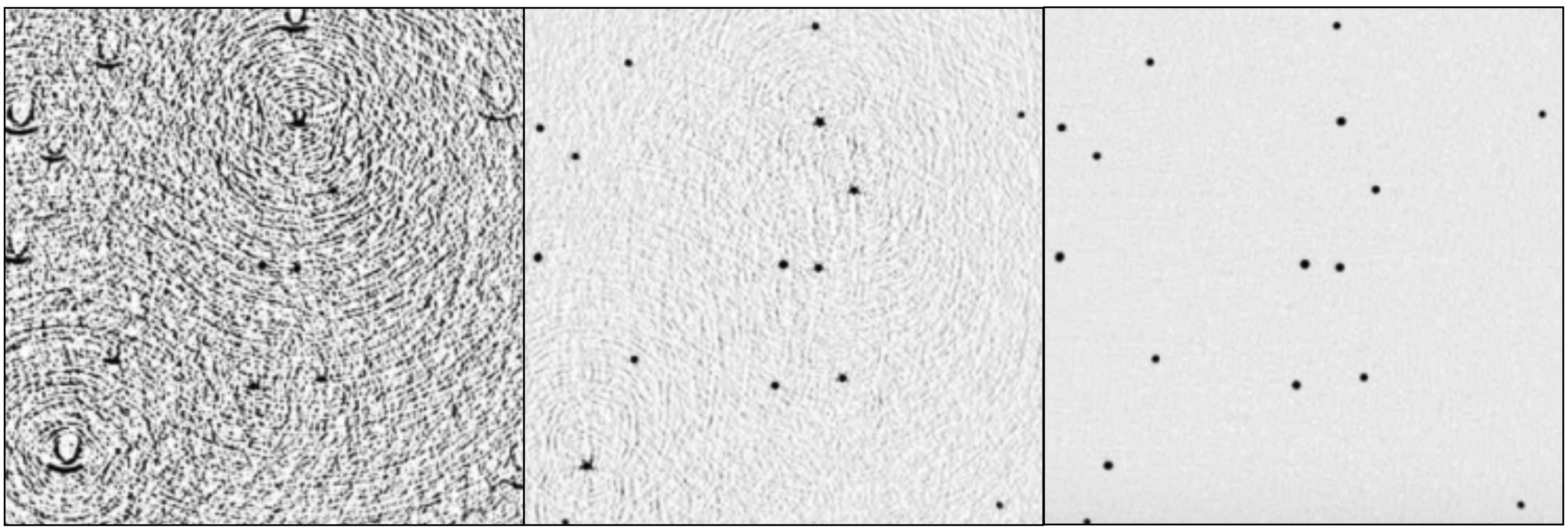
So how do we fix it?

(i) *w-projection* - probably using w-planes we use a w-kernel in the convolutional gridding step of the imaging process,

(ii) *w-snapshots* - we image each time step separately, re-project the warped images individually and then stack the results, and

(iii) *faceting* - we break the sky up into pieces and image them separately.





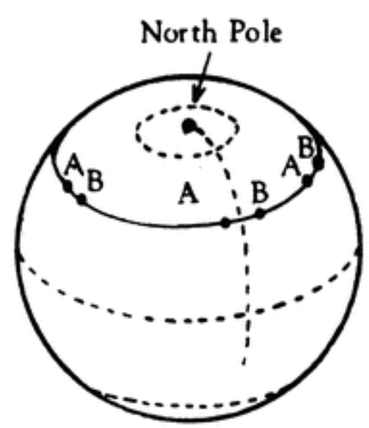
Cornwell, Golap & Bhatnagar, EVLA Memo 67

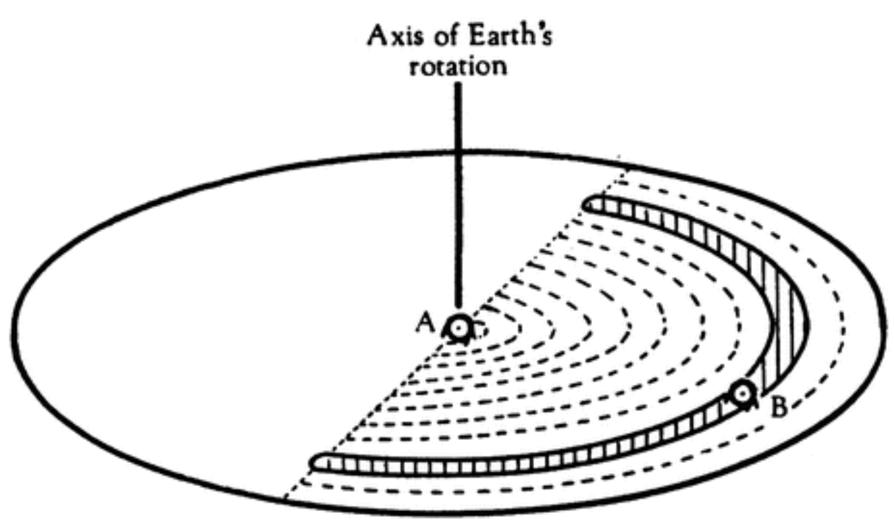
Summary

- *Non-coplanarity introduces an additional phase term into the visibility equation. This is time and direction variable.*
- *The effect of this phase term is to smear sources in a systematic manner. This smearing is a function of distance from the phase centre with sources further away being more distorted.*
- *Equivalently we can think of this phase term as warping the co-ordinate system on a snapshot by snapshot basis.*
- *To correct for this effect we can use (i) w-projection, which re-poses the phase multiplication as a convolution in visibility space, or (ii) w-snapshots, which re-projects the image plane for each snapshot and then stacks the images, or (iii) use faceting.*

Exercise: East-West Interferometers

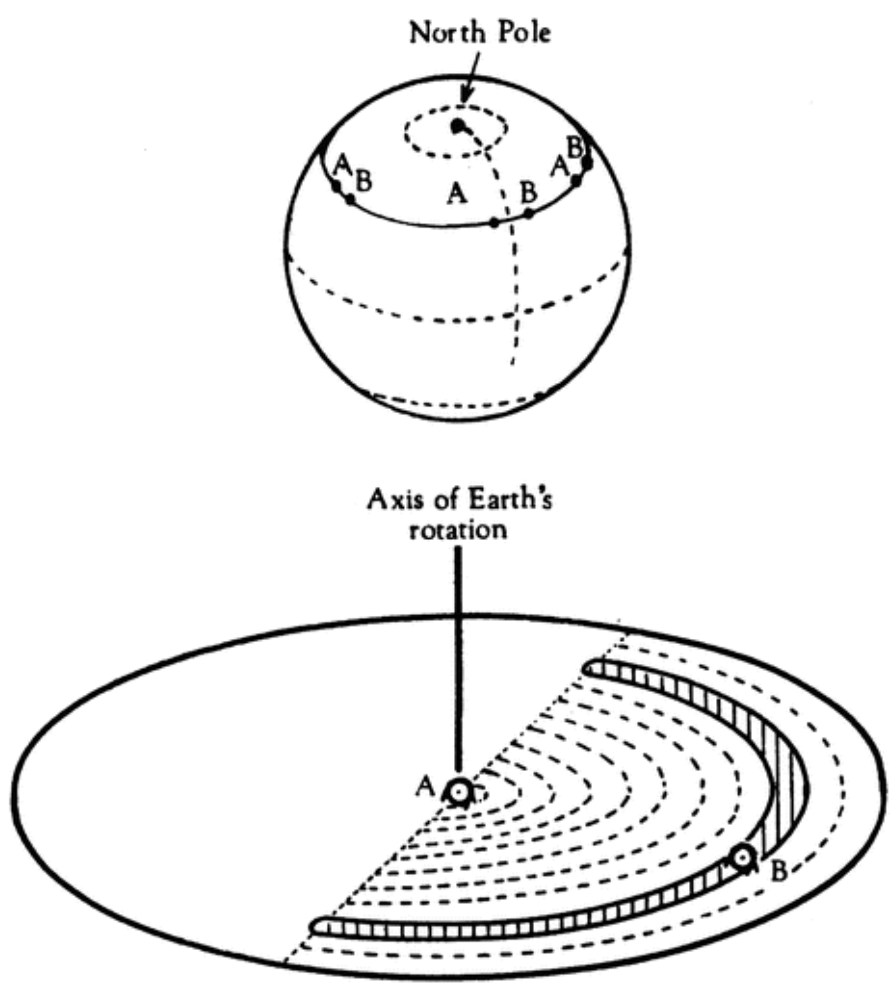
Once upon a time… interferometers were built in East-West formations and we didn't need to worry about non-coplanarity.





Exercise: East-West Interferometers

Once upon a time… interferometers were built in East-West formations and we didn't need to worry about non-coplanarity.



For East-West arrays, antenna separations remain constant in amplitude as a function of time when viewed from the North Pole.

Furthermore, they also lie in a *w=0* plane.

This is the same as saying that they sit on a *w=0* plane in *uvw*-space in a co-ordinate frame with the w-axis oriented towards the NCP.

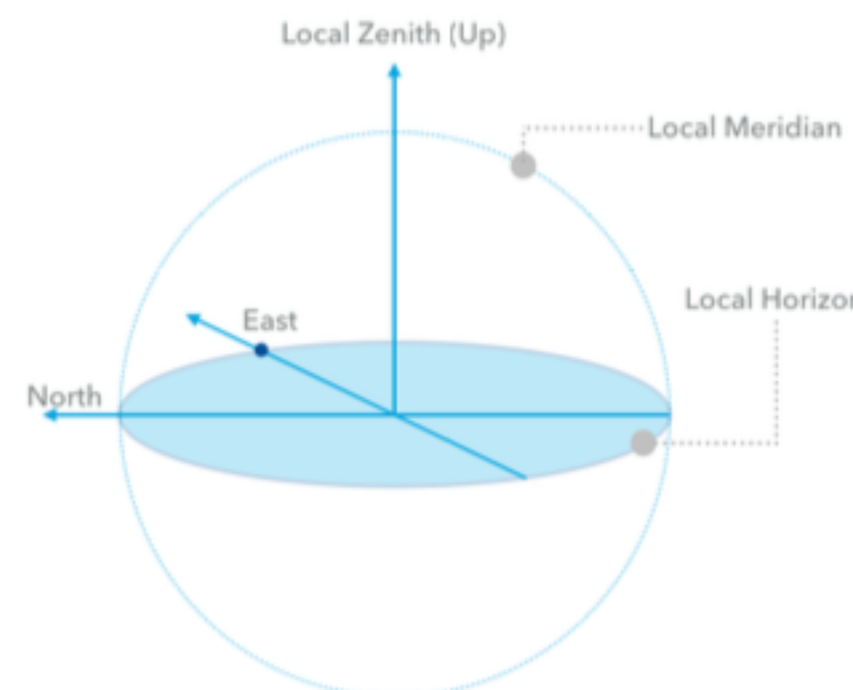
Let's demonstrate this to ourselves

We can define a three antenna East-West array with antenna coordinates in an East-North-Up (ENU) co-ordinate frame:

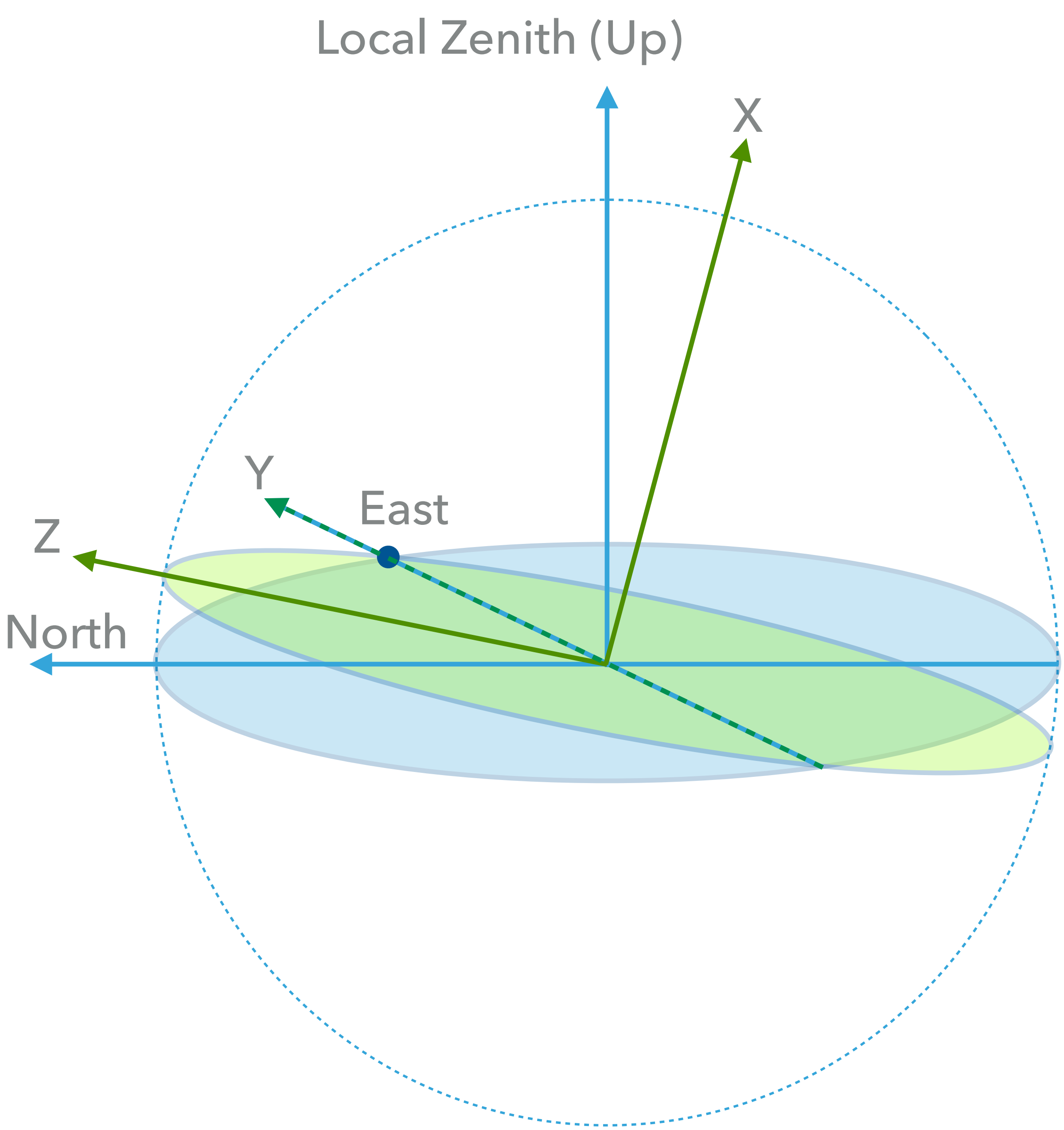
$$\mathrm{ae1} = \begin{bmatrix} -54 \\ 0 \\ 0 \end{bmatrix}; \quad \mathrm{ae2} = \begin{bmatrix} 164 \\ 0 \\ 0 \end{bmatrix}; \quad \mathrm{ae3} = \begin{bmatrix} 102 \\ 0 \\ 0 \end{bmatrix}$$

Let's put it somewhere on the Earth,

$$\ell = +34\,\text{degrees}, \quad b = -107\,\text{degrees}$$



ENU co-ordinate system



To calculate (u,v,w) for the antennas we need to first convert from ENU to a less local frame of reference.

Here we'll use the XYZ co-ordinate system, which has its z-axis aligned with the rotation axis of the Earth and its y-axis aligned with East.

To convert from ENU to XYZ we use the rotation matrix,

$$R = \begin{bmatrix} 0 & -\sin\phi & \cos\phi \\ 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \end{bmatrix}$$

In the XYZ frame our antenna positions are

$$\mathrm{ae1} = \begin{bmatrix} 0 \\ -54 \\ 0 \end{bmatrix}; \quad \mathrm{ae2} = \begin{bmatrix} 0 \\ 164 \\ 0 \end{bmatrix}; \quad \mathrm{ae3} = \begin{bmatrix} 0 \\ 102 \\ 0 \end{bmatrix}$$

To calculate *(u,v,w)* we need to define a source direction

$$\mathbf{s_0} = (H, \delta) = (-3.49\,\mathrm{hr}, +21^{\circ})$$

We can then calculate UVW from XYZ using the rotation matrix:

$$R = \begin{bmatrix} \sin H & \cos H & 0 \\ -\sin\delta\cos H & \sin\delta\sin H & \cos\delta \\ \cos\delta\cos H & -\cos\delta\sin H & \sin\delta \end{bmatrix}$$

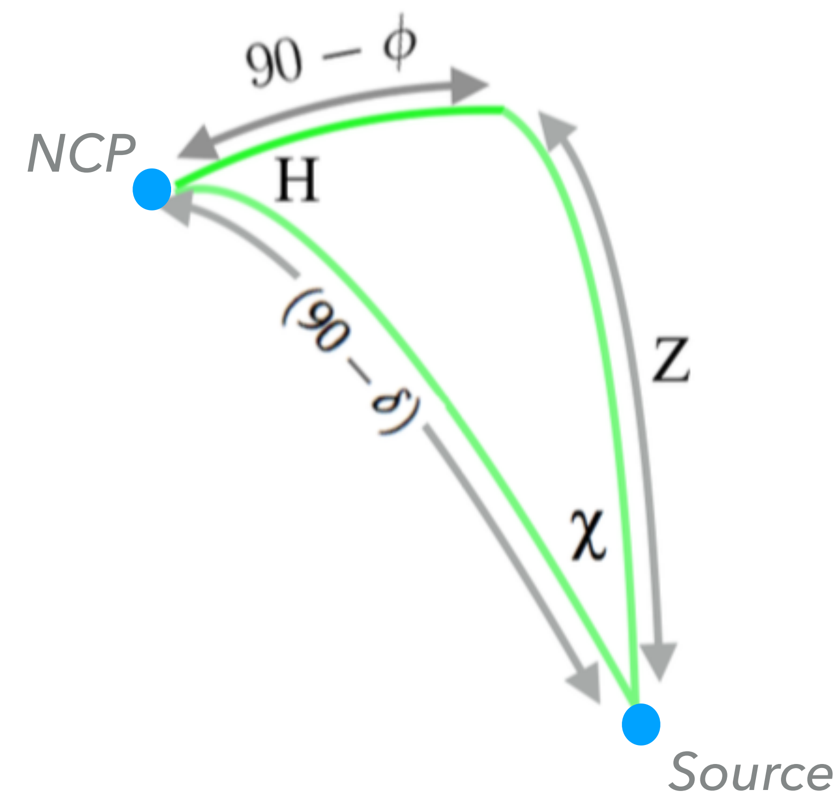
In the UVW frame our antenna positions are

$$\mathrm{ae1} = \begin{bmatrix} -46.82 \\ -9.6 \\ 25.2 \end{bmatrix}; \quad \mathrm{ae2} = \begin{bmatrix} 142.8 \\ 29.5 \\ -77.0 \end{bmatrix}; \quad \mathrm{ae3} = \begin{bmatrix} 89.0 \\ 18.4 \\ -47.9 \end{bmatrix}$$

The first thing to notice is that the *w*-value is not zero.

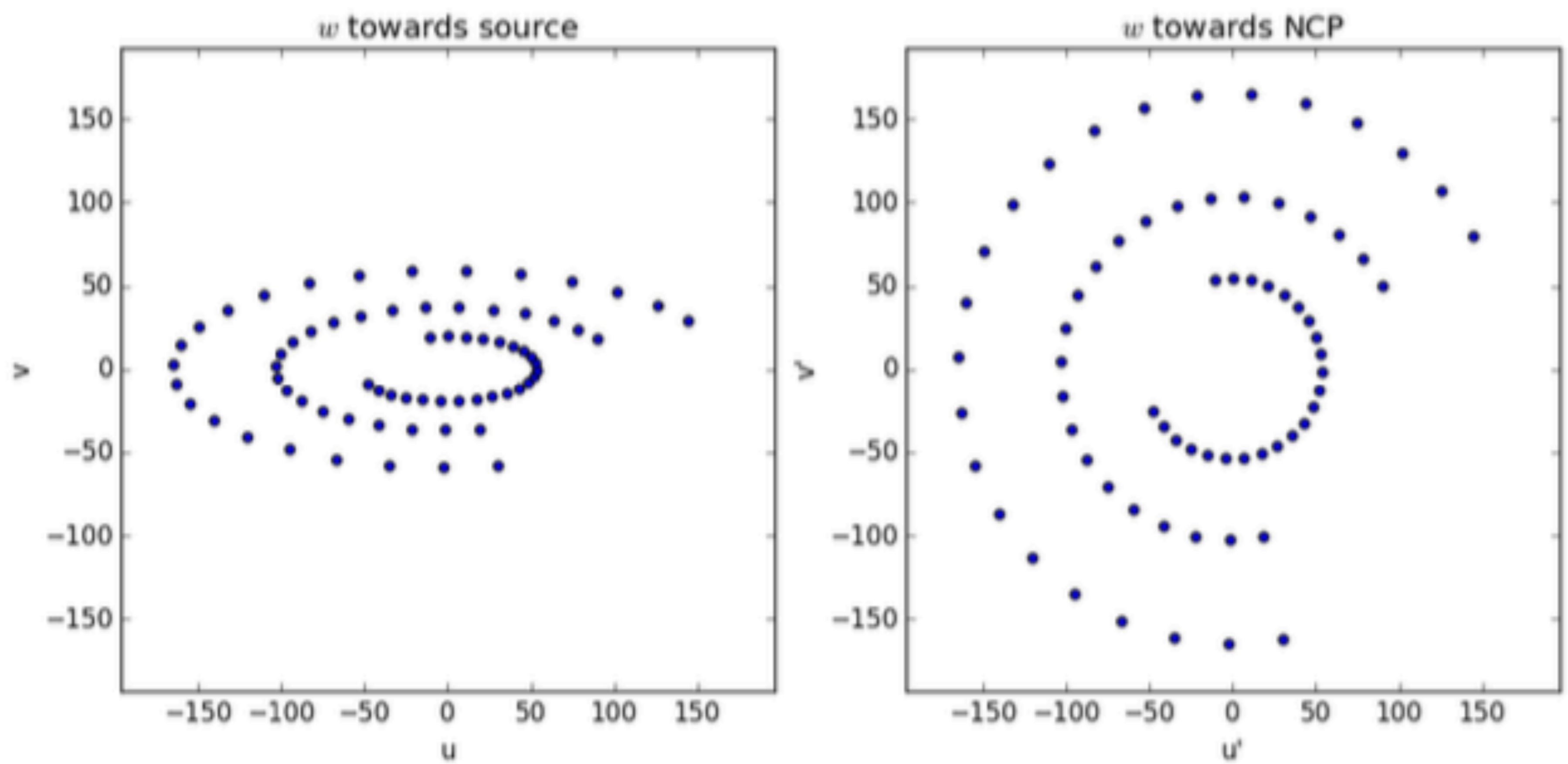
For the w-coordinate to be zero we need to be in a frame oriented with the *w*-axis towards the NCP, denoted *(uvw)'*:

$$\begin{aligned} u &= u' \\ v &= v' \sin\delta \\ w &= -v' \cos\delta = -v \cot\delta. \end{aligned}$$



You can verify this relationship using the numbers above.

If we plot the uv-coverage for a range of HA in the two frames it looks like this:



Additional Exercise

For an East-West array, all the antennas lie on the y-axis in the corresponding XYZ frame. This means that when we transform to UVW co-ordinates everything lies in a plane that pivots into *w != 0* around *v = 0*, i.e. the *u*-axis. This means we only need a single rotation to bring the *w*-axis inline with the NCP and that *w* is just a function of *v*:

$$w = -v\cot\delta$$

For an array with some North-South geometry the plane will not pivot solely around *v=0*, but rather in a plane that is a function of both *u* and *v*. This means that two rotations are needed to bring the *w*-axis in line with the NCP. In this case

$$w = u\sin\chi\tan Z - v\cos\chi\tan Z$$

Derive this expression.