wprojection

Algorithm & implementation
Usual Equation

\[ V(u, v, w) = \int \frac{I(l, m)}{\sqrt{1-l^2-m^2}} e^{-2\pi i \left[ ul + \bar{v}m + w(\sqrt{1-l^2-m^2}-1) \right]} \, dldm \]

\[ V(u, v, w) = \int \frac{I(l, m)}{\sqrt{1-l^2-m^2}} G(l, m, w) e^{-2\pi i \left[ ul + \bar{v}m \right]} \, dldm \]

where

\[ G(l, m, w) = e^{-2\pi i \left[ w(\sqrt{1-l^2-m^2}-1) \right]} \]
Usual equation - different view

- Or using convolution theorem

\[ V(u, v, w) = V(u, v, 0) * \tilde{G}(u, v, w) \]

\[ \tilde{G}(u, v, w) = FT(G(l, m, w)) \]
Griding-Convolution

- We use FFT and we resample (grid/degrid) the visibilities in the process of making images or predicting the visibilities.
- Usually use spheroidal function to reduce aliasing due to sampling on the grid.
- Effectively doing/undoing a convolution to/from the grid.
griding revisited

- $V(u,v,w)$ – locate $u,v$
- Use $w$ to get right $\tilde{G}(u,v,w)$
- Use this along with spheroidal function to grid and degrid
- Note each $u,v,w$ sample need its own $\tilde{G}(u,v,w)$
Implementation in AIPS++

- User specify number of planes
- Determine range of $w$
- Steps in $\sqrt{w}$ (empirically found)
- Calculate a cube of $G(l, m, w)$ with $w$ as 3rd axis
- FT it and keep it in memory (one time computing cost)
Performance
Example on real data