VLA SCIENTIFIC REPORT NO. 110

COMMENTS ON GRIDDING IN THE (u,v) PLANE

Eric W. Greisen

PURPOSE: In VLA Scientific Report 105, R. M. Hjellming examined the consequences of the gridding of measured data in the (u,v) plane prior to Fourier transformation. This note is intended to point out that the large "phase errors" discussed by Hjellming cause no difficulties in the mapping process and that the existence of periodicities in maps produced from gridded data does not have to be a serious problem.

THEORY: Let V(u,v) be the continuous complex visibility function. Since the integration times and the number of baselines are finite, we actually measure a sampled visibility V' given by

$$V'(u,v) = V(u,v) \cdot S(u,v)$$

where $$S(u,v) = \sum_{i=1}^{I} w_i \, {}^2\delta(u-u_i, v-v_i)$$

is the sampling function and the w_i are the weights assigned to the data points. In order to avail ourselves of the FFT it is necessary to smooth the data to a grid. This process may be written as

$$V''(u,v) = III(u,v) \cdot \{C(u,v) * (V \cdot S)\}$$

where $$III(u,v) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} {}^2\delta(u-j\Delta u, v-k\Delta v)$$

and C is a "convolving function". For completeness, I should mention that a few benighted observers follow the above steps with a "division by the sum of the weights" within each cell to obtain

$$V''' = III \cdot R \cdot [C * (v-S)] \qquad (1)$$

where $R(u,v) = 1/(C*S)$ $\quad (C*S \neq 0)$

$R(u,v) = ?$ $\quad (C*S \approx 0).$

When we Fourier transform this mess we obtain

$$T'''(x,y) = \overline{III} * \overline{R} * [\overline{C} \cdot (T * B_D)] \qquad (2)$$

where $T = \overline{V}$

$B_D = \overline{S}$

and the superscript bar denotes a Fourier transform. In order to determine a synthesized beam pattern, we note that V(u,v) = 1 everywhere for a point source and obtain

$$T_p''' = \overline{III} * \overline{R} * (\overline{C} \cdot B_D). \qquad (3)$$

DISCUSSION:

(1) The R term is determined by the data and is only known at grid points. For this reason, we know very little about $\overline{R}$ except that it messes up our maps. All weighting should be accomplished with the w's and we should <u>never</u> introduce the function R into our maps.

The effect of the R term is to assign several weights to each data point depending on the grid points to which they are smoothed. In the case where C is a simple pill-box function, it is computationally convenient to compute the w_i as if we were determining an R function. However, in such cases we are really computing the w_i and should not confuse the conceptual simplicity of equations (1) - (3) with our computer-oriented tricks.

(2) The C term in the equations is the sole direct effect of the smoothing operation. It is true that, if we regard V'' as the true visibility function, we are making a relatively serious error in phase and/or amplitude. However, this error is precisely known and can be corrected without any errors other than round-off problems in the computer. Assuming, for the moment, that the $\overline{III}$ term may be ignored, we may rewrite equations (2) and (3) as

$$\frac{T''}{\overline{C}} = T * \frac{T_p''}{\overline{C}} \quad . \qquad (4)$$

Thus, the maps and synthesized beam patterns can be corrected for the smoothing and are left in a form which is both familiar and suitable for use of "cleaning" procedures.

(3) The $\overline{III}$ term causes the map to be aliased onto itself at boundaries determined by Δu and Δv. However, the part of the map which is aliased into the principal map area is multiplied by the factor

$$F = \frac{\overline{C}\ (x_a,\ y_a)}{\overline{C}\ (x_m,\ y_m)}$$

where the a refers to the map coordinates before aliasing and the m to the coordinates after aliasing. The ideal case, of course, is $F \equiv 0$. Unfortunately, this requires that the data be convolved in the (u,v) plane with a sinc function. Because of the slow decay of the sinc function, such a convolution is prohibitively expensive in computer time. For small amounts of data, C is usually taken as a Gaussian function. Such a choice is moderately expensive in computer time, but produces low values for F nearly everywhere. The Gaussian

has the advantage that its width is not directly tied to the choice of Δu and Δv. Thus a wide Gaussian can be used in the (u,v) plane to produce a narrow Gaussian and low values of F in the (x,y) plane. Of course, the wider the Gaussian, the more computer time is required for the convolution. The use of a Gaussian convolution function for VLA data should not be rejected without serious consideration.

The least expensive convolution function is the pill-box function:

$$C(u,v) = 1/(\Delta u\ \Delta v) \qquad \left(|u| \leq \frac{\Delta u}{2},\ |v| \leq \frac{\Delta v}{2}\right) \qquad (5)$$

$$C(u,v) = 0 \qquad \text{(elsewhere)}.$$

This function has the transform

$$\overline{C}(x,y) = \frac{\sin(\pi x/4x_o)}{(\pi x/4x_o)} \quad \frac{\sin(\pi y/4y_o)}{(\pi y/4y_o)} \qquad (6)$$

where x_o, y_o determine the size of the map and will be defined in §5 below. Such a convolving function provides a simple method for computing the weights w_i in a fashion such as suggested by Hjellming. Unfortunately, with this function F can be as high as 0.33 or so and falls off very slowly as (x_a, y_a) increase. Perhaps a two-stage pill-box procedure can be devised so that weights may be determined using the C of equation 5, but resulting in a final C covering $|u| \leq \Delta u$, $|v| \leq \Delta v$. In this case

$$\overline{C} = \text{sinc}\ (\pi x/2x_o)\ \text{sinc}\ (\pi y/2y_o),$$

and F will fall more rapidly although it retains the same maximum value in the principal map area. More complicated two-stage procedures should also be considered.

(4) The weights w_i should be chosen to produce a B_D with the least side-lobes. With small amounts of data, the weights are often taken proportional to the local density of data points at each data point. However, this is a lengthy computing procedure. Depending on the source and quality of data, there are two simple choices for the weights:

$$w_i = w_t\ (u_c,\ v_c)$$
$$w_i = w_t\ (u_c,\ v_c)/N_c$$

where w_t is a tapering function evaluated at the center of cell c and N_c is the number of data points in cell c.

(5) To determine Δu, Δv and the array sizes, let us assume (1) that the source is contained in an area somewhat smaller than $|x| < x_o$, $|y| < y_o$, (2) that the data lie entirely within $|u| < u_T$, $|v| < v_T$, and (3) that we wish m points per beamwidth in the x-direction and n points per beamwidth in the y-direction. To define a synthesized beam for cleaning purposes, we must actually compute the map over an area $4x_o \times 4y_o$. Plugging in the FFT relationships we find that

$$\Delta u = \frac{1}{4x_o} \qquad \Delta v = \frac{1}{4y_o}$$
$$\Delta x \leq \frac{1}{2mU_T} \qquad \Delta y \leq \frac{1}{2n\ v_T} \qquad (7)$$
$$M \geq 8\ x_o\ u_T\ m \qquad N \geq 8\ y_o\ v_T\ n,$$

where M, N are the x,y dimensions of the transform and the inequality arises from the necessity to have M, N integer powers of 2. Let us look at some numbers. For smooth, cleanable maps, m and n are ≥ 4. For the full continuum bandwidth, we may want x_o, $y_o \sim 1$ minute of arc.

If u_T, v_T are around 10^6 wavelengths then M and N are 8192! For a field encompasing the primary beam (x_o, y_o ~ 15'), our transform would have to be 131072 x 131072!!! These numbers suggest, among other things, that the requirements for the asynchronous computer must be reconsidered. Hjellming's arguments concerning the use of brute-force Fourier transforms also would suggest a reconsideration of the computer requirements.