NATIONAL RADIO ASTRONOMY OBSERVATORY
SOCORRO, NEW MEXICO
VERY LARGE ARRAY PROGRAM

VLA COMPUTER MEMORANDUM NO. 152

AN IMPLEMENTATION OF CLEAN

B. G. Clark

December 1979

1. THE ALGORITHM

The CLEAN algorithm is a nonlinear iterative method of deconvolving the instrumental point source response from the observed data. The method consists of the following steps:

1) locate the maximum in the image (the nonlinear operation in the process),
2) generate a "component", a spike at this location, and of intensity some fraction G of the maximum in the image,
3) calculate the convolution of this component with the instrumental point source response function,
4) subtract this instrumental response function from the map,
5) repeat steps 1-4 until the remaining map is satisfactorily small,
6) generate and add back into the map the whole collection of components, convolved, not with an instrumental point source response, but with an aesthetically pleasing function such as a Gaussian.

The main computational effort of CLEAN, which is considerable, lies in the steps 1-4 above. Ignoring the nonlinear step, the effect of the repeated application of steps 2-4 is the convolution of the collection of spikes ("source components") with the instrumental response function. It is well-known that the computationally efficient way of doing a convolution with an extended function is to Fourier

transform, multiply and inverse transform. In an attempt to capture this computational advantage, I have implemented, on the 11/70 and array processor, a modified version of the CLEAN algorithm.

The requisite for doing this is to separate the component location step from the convolve-subtract step. This is done by providing an approximate algorithm for the component location step, rather than using the full CLEAN algorithm.

The algorithm I have implemented is to consider the high points only of the map, and to subtract a beam from them only if they are within a certain distance (the "beam patch") of the maximum. This vastly decreases the computational effort of CLEAN, and removes it almost entirely to the convolution step. I shall explain below the details of how the high points and beam patch are chosen.

The procedure is divided into major and minor cycles. A major cycle begins by constructing a histogram of map values, and a plot of the maximum sidelobe outside a centered square of a given size. It is then possible to choose a beam patch size and a map limiting value such that

1) The largest map point less than the limiting value is the same fraction of the map peak as the largest beam value outside the beam patch.
2) Both the map points above the limiting value and the beam patch will fit in the available main memory.

For no very good reason I have constrained the beam patch size to be between 21x41 elements and 64x127. [The small end requires a small part of main memory (which is 32 k words) and the large end is rarely encountered in the pursuit of the above criterion and was imposed as a matter of programming convenience.]

The map points are stored as a value (floating point) and a location (two 13-bit integers stored in a single word of main memory).

A minor iteration cycle then consists of locating the maximum value among those stored in main memory, examining all the map points in main memory and, from those which fall within one beam patch width of this maximum point, subtracting the CLEAN gain times

the flux of the component times the beam level, determined by a table lookup in the beam patch area.

A major iteration cycle consists of the histogram computation, an appropriate number of minor cycles and a convolution/subtraction step. For convenience and speed, the transform of the beam (the "transfer function") is kept, and multiplied by the transform of the components. For reasons discussed in Section 3, this transfer function is also the input to the algorithm rather than the beam.

The transform of the component spikes is done by a standard discrete transform in the Y direction followed by a FFT in the X direction. This technique (suggested to me by Walter Jaffe) eliminates the long transpose step at the cost of little or no extra computation. The component transform is multiplied by the weight file (producing the transform of the spikes convolved with the beam) generating a new mapmaker input file. The map generated by the standard mapmaker routines can then be subtracted from the dirty map to yield a new residual map. Obviously, once one has the transform of the components one can, for negligible extra effort, multiply it also by the transform of the Gaussian "clean beam" and add it back into the grid file. This has the disadvantage that the clean beam in fact used is not the specified Gaussian, but that Gaussian convolved with the sinc function which is the transform of the field of view itself. By the time you are up to four points per beam, though, where all careful CLEAN's should take place, the difference is negligible.

The rule for deciding when a major iteration should end is not well-defined. Obviously, it should not be pushed to the point where the maximum point in the map present is smaller than some point which was ignored when the map was scanned for those points to reside in main memory. In fact, a slightly more conservative rule would seem appropriate. I have used the following, with apparent success:

> A major iteration is terminated at minor iteration N if the N^{th} map maximum is smaller than S_{Max} * F(M,N) where M is

the minor iteration number at which the major cycle began, and $F(M,N) = 1 + \sum_{n=M}^{N} 1/n$.

This rule seems in practice to result in the first few components of a new major iteration being larger than the last few of the previous cycle, but falling to that level in a few percent of the major cycle length.

2. SOME REMARKS ON FULL-FIELD CLEANING

The well-known corrections for the effects of gridding the data interact with CLEAN in obscure fashion. A brief exposition is given below for the case of one-cell boxcar convolution.

A word on notation: I shall use capital and lower case letters to indicate a Fourier transform pair. A centered dot will indicate multiplication and an asterisk, convolution. The bed-of-nails function, $\amalg$, is its own Fourier transform.

$I(x,y)$ is the sky brightness

$b(u,v)$ is the VLA sampling function.

The observed visibilities are therefore

$$t = i \cdot b$$

Let $s(\Delta u,\Delta v)$ be a one-cell u,v sampling boxcar.

$$S(x,y) = \mathrm{sinc}(x)\cdot\mathrm{sinc}(y)$$

Then the gridded data is

$$m = \amalg \cdot (s*t)$$

The transform is the map

$$M = \amalg * (S\cdot T)$$
$$= \amalg * [S\cdot(I*B)]$$

Similarly, we have the same relation for the beam. The computed beam $\underline{b}'$ is

$$b' = \amalg \cdot (s*b)$$
$$B' = \amalg * (s\cdot B)$$

If we can, at this point, ignore the replication operator $\amalg$, we can proceed further. This assumption is, in effect stating that there

is no appreciable flux outside the field of view and that the sidelobes of the true beam, B, are unimportant outside the field of view. Then,

$$B' = S \cdot B$$

$$M = S \cdot (I * B)$$

or $$M/S = I * (B'/S).$$

Therefore we should use CLEAN to subtract gridding corrected beams (B'/S) from the gridding corrected map (M/S). This is in fact implemented by use of the MAPTYPE 'XFR' in MAKMAP to generate the transform of (B'/S), which is the appropriate transfer function.

Although the above assumption sounds entirely reasonable, one should be a little wary. It is easy to find cases which sound inherently reasonable in which the above assumption is violated. Perhaps the most egregious is the case in which each filled cell is well-filled, that is, uniformly filled with datapoints throughout its extent. Then, after gridding, the response to a point source near the edge of the field is, everywhere in the u,v plane, simply multiplied by the appropriate sinc functions. Its sidelobe response in the map M is therefore a translated B'. In the gridding corrected map, M/S, the sidelobe pattern will be distorted, and will not clean up by subtracting B'/S at the location of the source. (This case also conflicts with Greisen's numerical experiments of VLA Scientific Memorandum No. 123.) The practical importance of such special cases is not at all clear to me, but we should let their existence be a constant caution to us.

3. SOME REMARKS ABOUT DYNAMIC RANGE

Westerbork reductions have encountered an unexpected problem with the number of bits required to express the beam for use in CLEAN; 18 bits did not seem to be enough. This is surprising in that 18 bits can express a ratio of 130000:1, whereas the dynamic range of the maps being cleaned was only of order 1000:1.

To see qualitatively how this comes about let us consider the total flux. Suppose we are cleaning an N*N map. If the beam has an inherent accuracy of E at each point, the the total flux, the sum

of all beam points, will have an rms uncertainty of NE. Since we would like (and indeed require, to fit the observations of a very extended source) a similar accuracy in the total flux as is present in the short spacing measurements, perhaps 1%, we need beam accuracies somewhat better than .01/N, which can be smaller than 10^{-5}. This unhappily conflicts with our desire to store beams as 16-bit scaled integers.

However, there appears to be no corresponding problem in the u,v plane. The required accuracy in the u,v plane is apparently only a few times the dynamic range to be achieved in the output map. Since only the wildest optimists expect a dynamic range as great as 3000:1 to be achieved, the 15 bit + sign representation of the data should provide adequate dynamic range. We therefore store beam information in the form of transfer functions (from MAPPER say MAPTYPE 'XFR') for use by CLEAN.