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

VLA COMPUTER MEMORANDUM NO. 157

REMARKS ON THE GRIDDER DEVELOPMENT

W. N. Brouw

February 1981

GRIDDER consists of six parts:

(1) Read the data from SORTER.

(2) Supply gain corrections and delete data.

(3) Grid the data onto rectangular grid.

(4) Do first pass Fourier transform.

(5) Do second pass Fourier transform.

(6) Write maps onto storage medium.

This note will mainly be on steps 3, 4, and 5, but some remarks on the other steps will be made at the end.

Assumptions

In writing the program for step 3, the following assumptions have been made on the data:

(1) The data is sorted into pigeon holes with a known width ΔU_p nsec, according to its $|U|$-value.

(2) The data format is, roughly, comparable to the DEC-10 format; i.e. each data packet is assumed to consist of at least the (U,V,W)-coordinates and a number (≥ 1) of complex data values.

(3) For each database the maximum $|U|$ ($U_{p,max}$ nsec) observed is known. (This is not essential, but can save considerable computer time.)

(4) For each database the maximum and minimum V observed ($V_{p,max}$ and $V_{p,min}$ nsec) are known. (Again, not essential, but helpful in reducing computer time.)

(5) Data can be read from SORTER in <u>decreasing</u> order of pigeon hole $|U|$.

(6) Gain corrections and data deletion has been applied.

<u>Limitations</u>

(1) The available AP's are limited to Fourier transforms of 8192 points. Although the AP's can be changed to support larger Fourier transforms, I will assume this to be the maximum size of any map in any dimemsion to be made by the pipeline.

(2) The available transpose memory has 4M words. This supports one output map of 2048 x 2048 points (or other sizes, see below) if $U_{p,max}$ is not known before hand, or, if $U_{p,max}$ is known and a reasonable number of points per synthesized beam has been specified, of 2048 x 4096 points. For any size beyond this, the map has to be made in 2 or more steps. As Barry Clark has already pointed out, it is, in general, faster to reread the original database, rather than adding maps together later. I have programmed with this in mind.

(3) The currently planned amount of memory for the Gridding AP will not be able to support gridding in one step for an 8192 point transform with a correlation function of more than 1 grid point

wide. If no more core will be available (see for details of the necessary amount later) the gridding and subsequent transforms have to be done in more than one step; i.e. the database has to be read more than once.

(4) The "Third dimension" necessitates the generation of more than one 2-dimensional map per output map. This may, again, necessitate the rereading of the database.

(5) The program supports an unlimited number of simultaneous maps from the one database. The restrictions on AP gridding memory size and on the transpose memory size, will make rereading of the database necessary in many instances.

Definitions

ΔU_p (nsec): Width of a pigeon hole.

$U_{i,max}$(nsec): Maximum $|U|$ in pigeon hole i. (i = 0,1,...). i.e. $U_{i,max} = (i + 1).\ \Delta U_p$.

$U_{p,max}$(nsec): Maximum $U_{i,max}$ in input database (or combination of databases).

$V_{p,max}$(nsec): Maximum V in input database(s).

$V_{p,min}$(nsec): Minimum V in input database(s).

ν_i (GHz): Frequencies of map i to be made (i = 0,1,...)

F_ℓ(radians): Fieldwidth in ℓ-(R.A.) direction of transformed map.

F_m(radians): Fieldwidth in m-direction.

F_n(radians): Fieldwidth in n-direction. NOTE: F is the same for all maps made simultaneously.

L_F(-): Number of points in ℓ-direction transform (L power of 2, $4 < L \leq 8192$).

M_F(-): Number of points in m-direction transform (M power of 2, $4 \leq M \leq 8192$).

N_F(-): Number of points in n-direction transform ($N \geq 1$).

L_o(-): Number of output points per map in ℓ-direction ($0 < L \leq L_F$, even).

M_o(-): Number of output points per map in m-direction ($0 < M \leq M_F$, even).

$W_{c,u}$(gridpoints): Width of convolving function in U-direction ($W > 0$).

$W_{c,v}$(gridpoints): Width of convolving function in V-direction ($W > 0$).

NOTE: W is a real number. Often a 2.5 point convolution function is as good as a 3 point, and is 30% faster.

$C_u(U)$(-): Convolution function in U-direction.

$C_v(V)$(-): Convolution function in V-direction.

NOTE: The convolution functions should be given as a table with $n_{c,uv}$ points per unit increment in U or V. (n_c power of 2 [positive or negative]).

$W_{w,u}$(nsec): Width of area for uniform weight calculation in U-direction.

$W_{w,v}$(nsec): Width of area for uniform weight calculation in V-direction.

$B_u(U)$(-): Uniform weight distribution function in U-direction.

$B_v(V)$(-): Uniform weight distribution in V-direction.

NOTE: Distribution function assumed seperable; function given as table with $n_{w,uv}$ points ($n_w > 0$, real) per unit increment in U or V.

NOTE: The widths are given in nsec, hence will vary from map to map at different frequencies.

G_{max}(words): Core available for gridded data.

T_{max}(words): Size of transpose memory.

$(\ell,m,n)_s$(gridpoints): Shift output map center to $(\ell,m,n)_s$.

$B_i(\ell,m,n)$(gridpoints): Brightness of pointsource to be subtracted at position (ℓ,m,n).

N_{map}(-): Number of simultaneous maps to be made.

$N_{T,map}$(-): Total number of simultaneous 2-dimensional maps; i.e. $N_{T,map} = N_{map} \times N_F$.

Inputs

The input to the gridding/transform programs consists basically of the above defined parameters. In addition the following inputs are required:

type = 1: Antenna pattern produced

2: Map produced

3: Antenna pattern and map produced

weight = 0: Natural weighting

1: Uniform weighting

NOTE: As long as part of the flagging is done in the database, uniform weighting requires two passes through the database. If all flagging is done in (a) seperate flagging file(s), step 2 of the program could calculate the U,V,W coordinates for the first pass.

buffer element: A buffer element consists of:

0 - U (nsec)

1 - V (nsec)

2 - W (nsec)

3 - delete pattern

4 - weight of this element

5 - complex value 0

6 - complex value 1

.

. complex value n.

e.g. for current data n = 3, and value's are AA, AC, CC, and CA; for 2 IF's n = 7; for spectral line work n could be anything up to 511.

map description: For each simultaneous map (i.e. a total of N_{map}):

0: Correlation buffer pointer

1: Uniform weight buffer pointer

2: Pointer to a source list.

source list(s): Lists of sources to be subtracted before gridding the data:

0: B_o

1: ℓ_o

2: m_o

3: n_o

4: B_1

.....

value description: Description of correlation wanted.

0: map descriptor pointer

1: $\nu \times F_n$: Scale W

2: $\nu \times F_\ell$: Scale U

3: $\nu \times F_m$: Scale ν

4: weight : multiplication factor

5: point pattern: To match delete pattern

6: ± buffer element offset: Add (+) or subtract (-) data at given offset in buffer element

7: ± offset

.....

m: ℓ_s : shift coordinates

m + 1: m_s

m + 2: n_s

m + 3: indicator : 0 : no shift

≠0 : shift

Examples:

If the buffer element consists of 8 values (2 IF's), one map could be obtained by two value descriptors:

First:

0: Map description 1

1: $\nu_1 \times F_n$: ν_1 = First IF frequency

2: $\nu_1 \times F_\ell$

3: $\nu_1 \times F_m$

4: 0.25

5: 15.

6: +5

7: +7

8: +9

9: +11

10: 0

Second:

0: Map description 1

1: $\nu_2 \times F_n$: ν_2 = Second IF frequency

.

.

.

6: 13

7: 15

8: 17

9: 19

.

.

To produce all 4 polarization maps, the value descriptions could be:

1: As above

2: As above

3: 0: Map 2

1: $\nu_1 \times F_n$

6: +5

7: -7

9: +9

11: -11

etc., a total of 8 descriptors.

NOTE: With the separation of value and map descriptions the possibility exists to average before or after gridding for all spectral line combinations wanted.

Buffers: 2-buffers for overlapped I/O and gridding.

Action

From the input data, calculate:

N_{map}: Total number of input maps

$N_{T,map} = N_{map} \times N_F$: Total number of correlation maps.

NOTE: CEIL(x) = Smallest integer not less than x

FLOOR(x) = Largest integer not greater than x.

$NP_2(x)$ = Smallest integer for which $x \leq 2^n$

$$V_{min} = CEIL(MIN_i(MAX(V_{p,min} \times \nu_i \times F_m - W_{c,v}/2, -M_F/2))) \quad i = 0,1,---,N_{map}-1$$

$$V_{max} = FLOOR(MAX_i(MIN(V_{p,max} \times \nu_i \times F_m + W_{c,v}/2, M_F/2-1))) \quad i = 0,1,---$$

$$U_{max} = FLOOR(MAX_i(MIN(U_{p,max} \times \nu_i \times F_\ell + W_{c,u}/2, L_F/2-1))) \quad i = 0,1,---$$

$$U_{min} = CEIL(MIN_i(MAX((U_{p,max}-2.\Delta U_p)x\nu_i xF_\ell - W_{c,u}/2, -W_{c,u}/2))) \quad i = 0,1,---$$

$$V_n = V_{max} - V_{min} + 1$$

$$V_N = V_n$$

$$U_n = U_{max} - U_{min} + 1$$

$$U_N = U_{max} - CEIL\ (-W_{c,u}/2) + 1$$

i.e.: The v-width can contain all points with overlap for the correlation function. The size in u is determined by the correlation width plus twice the pigeonhole width for double buffer output to the first phase transform AP.

In the case of uniform weighting, we also define:

$$V_{w,n} = V_n + 2\ FLOOR\ (W_{c,u}/2)$$

$U_{w,n} = U_{max} - MIN(U_{min}, MAX_i((U_{p,max} - 2\Delta U_p - U_{w,u}) \times \nu_i \times F_\ell))$. i.e.: The uniform weight should "look ahead".

Example

We want to make one 1024 x 1024 map at 21 cm for the A array with 0°5 diameter, 3.5 x 3.5 correlation and a uniform weight covering 75 nsec (25m). If we further assume that there are 512 pigeonholes up to a maximum U of 75000 nsec.

In that case we have:	For 0°1 map:
$\Delta U_p \approx 150$ nsec	
$U_{p,min} \approx 75000$ nsec	
$V_{p,max} \approx 75000$ nsec (assumed)	
$V_{p,min} \approx -75000$ nsec (assumed) $\nu \approx 1.4$ GHz	
$F_\ell \approx 0.01$ radians	
$F_m \approx 0.01$ radians	
$F_n = 10$ radians	
$L_F = 1024$	
$M_F = 1024$	
$N_F = 1$	
$W_{c,u} = 3.5$ gridpoints	
$W_{c,v} = 3.5$ gridpoints	
$W_{w,u} = 75$ nsec ≈ 1.1 gridpoints	
$W_{w,v} = 75$ nsec ≈ 1.1 gridpoints	
$v_{min} = 512$ (NOTE: U,V do not fit completely)	$v_{min} = -106$
$v_{max} = 511$	$v_{max} = 106$
$u_{max} = 511$	$u_{max} = 106$
$v_n = 1024$	$v_n = 213$
$v_{w,n} = 1026$	$v_{w,n} = 215$
$u_n = 6$	$u_n = 6$
$u_{w,n} = 7$	$u_{w,n} = 7$

Core needed

Transpose memory needed:

$N_{map} \times N_F \times M_F \times NP_2(u_n) \times \text{type} = T_m$ words

Gridding memory:

$N_{map} \times N_F \times v_N \times u_n \times \text{type} + N_{map} \times v_{w,n} = G_m$ words

The number of passes through the input database are:

$CEIL(T_m/T_{max})$ x $CEIL(G_m/G_{max})$.

If we want to be able to do one 8000 x 8000 transform in at least one gridding step, the minimum amount of core for a reasonable (4x4) correlation and 512 pigeonholes, will be $\approx$ 256 k words.

Status

A. The gridding step works, including:

- natural/uniform weighting
- double buffered input and output
- any correlation size/function
- any uniform weighting area/function (separable)
- any, separable, grading function
- pre-gridding averaging
- post-gridding averaging
- multiple maps and/or antenna patterns
- source subtraction
- third dimension
- field center shift

Timing: 80 sec for 1 M points for 4 x 4 correlation and natural weighting

+ 2.5 sec for field shift for 1M points.

+ 2 sec per subtracted source for 1M points.

+ $\approx$60 sec for 1M points for uniform weighting.

B. The first pass Fourier transform has worked, and is being rewritten at the moment. It will include:

- 1.5 buffer input/output (core limitations)
- third dimension

- multiple maps/antenna patterns
- source subtraction using the calculated transfer function
- output to user of gridded data and/or transfer function
- 2 versions: Separate AP or combined with gridding function
- output map size less than transform size

C. The second pass Fourier transform has still to be written. It will probably include:

- third dimension correction
- choice of output between all data per map, or a cut at constant right ascention ("declination-frequency")
- correction for convolution "envelope"
- multiple maps/antenna patterns
- automatic handling of large maps
- output map size less than transform size

D. The 11/40 programs and the gain correction have not been thought about yet. Its details will depend on the definition of the new Flagger file.

Conclusion

The above is not meant as a detailed description, but serves to give a general idea of how my thoughts have developed.

Appendix A

Possible Extensions

1. To support a map size greater than 8192 on a side, hardware changes have to be made to the AP's and some small software changes to the Fourier transform and field shift/source subtract routines. Furthermore, additional core for the FT AP's (at least 4 x size words, and preferably double this amount for double buffering). If the core size extends beyond 64 k words, additional software changes for multiple pages are necessary.

2. Extended addressing capabilities of AP (i.e. the 64-bit AP), will decrease central loop of gridding by 30%!

3. Optimally I think the AP's should have the following memory sizes:

Gain corrector/gridder:	$\gtrsim$ 384 k words
First phase transformer:	64 k words
Second phase transformer:	64 k words
Transpose memory:	64 M words (needs special interface instead of IOP's for 26 bits addressing).

With the above 8192 x 8192 transforms can be done optimally. To include the third dimension for this size (say 4 points), the gridder should have $\geq$ 512 k words, and the transpose memory 120 M words (assuming a "reasonable" number of points per beam).

4. The U, V, W-coordinates in the database are, probably, not precise enough (16 nsec precision?) for source subtraction and/or large

field mapping (16 nsec precision will be equal to a 16 nsec "box" convolution). If they are not precise enough, recalculation during the gain correction phase with the aid of the time and antenna file should be no large task ($\lesssim$10 sec for 1M points?).

5. Uniform weighting requires two passes through the database. If all flagging were done in separate flagfiles, on antenna and IF basis, and maybe even on an interferometer and spectral line channel basis, the first pass could be replaced by AP computations, greatly reducing the, probably limiting, I/O load.

81.02.12

Thoughts/Proposed Gridder/Mapper (80.10.29)

I will assume that corrected data is available as a set of from 1 to 8 complex numbers, stamped with U, V, W (in nsec) and weight (in essence the number of 10 sec samples averaged). The 1 to 8 values could be 1-8 frequency channels, or continuum channels, or both.

The data is "sorted" per frequency group in $|U|$, and put into pigeon holes of width ΔU_p nsec. Each frequency group has its own pigeon holes. The boundaries of the pigeon holes are at $i.\Delta U_p$ (i = 0, 1, ...). I will, furthermore, assume that the data will be fed into the AP starting with data in pigeon hole number 0.

The object will be to generate one or more maps and/or antenna patterns from this (and other) data.

1. I will consider the following possibilities:

   a. Pre-gridding averaging

      From the 1-8 datapoints (D_i) 1-8 values (V_i) can be obtained by adding/subtracting a subset of the datapoints.

      E.g. $D_0 + D_1 \ldots\ldots D_7$

      or $D_0, D_1 \ldots\ldots D_7$

      or $D_0 + D_1 + D_2 + D_3 - D_4 - D_5 - D_6 - D_7$

      or $D_0 - D_1 + D_2 - D_3, D_4 - d_5 + D_6 - D_7$

      or $D_0 + D_1 + D_2 + D_3 + D_4 + D_5 - D_6 - D_7, - D_6 - D_7,$ $- D_6 - D_7$ (with gridding with source grid).

   b. Post-gridding averaging

      The V_i can be gridded onto 1-8 different grids, which are identical in size (on sky) and dimensions, but vary in frequency.

c. Post-map averaging

Can be done in 2 ways: a. Just after all the maps are made, outside the AP's

b. As post-gridding averaging.

In the latter case, after having processed all data in a certain frequency group and |U| pigeon hole, a different frequency group in the same pigeon hole can be handled.

d. "Third" dimensions

The third dimension will be handled if asked for.

2. Assumptions

a. I will assume (as pointed out in Memo..... by Barry Clark) that it takes more time to read a map than to read the original data for large maps. Therefore, in the case of, for instance, large maps with a third dimension, the map will be processed piece wise, rather than per third dimension plane.

b. I will assume that flagged data has already been deleted from the gridder database in the gain correction phase immediately preceding the gridding process.

c. For the gridding phase 1 page of 64 k of AP data memory is available. I will, however, allow for extension to multiple page memory in the program.

d. Each of the two Fourier transform phases has 1 page of 32 k of AP memory available.

e. Maps produced will start at the top left-hand corner. If 0 is at field center, there will be one less point in the top and right-hand half than in the bottom and left-hand half.

Q: Is this the definition now?

3. Layout Data

a. General

$|V|_{max}$ (nsec) - The maximum $|V|$ in the set of data to be used. It would be nice if SORTER could supply this value. In general it will save half the gridding memory requirements.

Phase (0,1,...) - The phase number for multiple passes through the data.

Mapsize (radians) - The x,y,z-size of the map(s) to be
(X_r, Y_r, Z_r) produced. z is allowed to be zero.

Map dimensions - The number of points in the x,y,z-directions
(X_m, Y_m, Z_m) to be produced. X and Y must be powers of 2.

Output dimensions - The number of points in the x,y-directions
(X_o, Y_o, Z_o) to be finally output. X and Y must be $\leq$ the corresponding map dimensions.

Switches - Not fully defined, but will include:
- Natural/uniform weighting
- Restore for gridding function influence
- Force minimum number of phases by limiting U,V coverage.

Gridding widths - Width of gridding function to be used in
(U_g, V_g) points. Can be a fractional number. I think 3.5 will be a good number in the proper functions. U_g and V_g may be different.

Gridding functions - The U and V gridding functions to be used in $2^{-n}g$ increments. The U and V functions may be the same, but don't have to be. n_g is the same for the U and V functions.

Note: The W-gridding will be a 1-cell box gridding.

Gridding restore functions - The U, V and W gridding restore functions.

Taper functions - The U and V (W?) taper functions. They may be identical.

Pigeon hole width (nsec) - The width of one $|U|$ pigeon hole. (U_p) Pigeon hole boundaries will be assumed at $i.U_p$ ($i = 0,1,2,...$).

Q: Will the pigeon hole size be configuration dependent?

$|U|_{max}$(nsec) - Maximum $|U|$ in the set of data to be used.

b. Value Definitions (changed)

N_v(Up to 8) entries defining:

Data combination - 16 bits for 8 input data points. 1 bit defining skip/no-skip for the data point; 1 bit defining add/subtract this data point.

Grid number - Number of grid (map) (0,1,...7) onto which this value should be gridded.

c. Map Definitions (changed)

N_g (Up to 8) entries, defining:

Map frequency (GHz) - Nominal frequency to be used in producing this grid (map).

Map/AP - 16 bits defining whether map, antenna pattern or both are wanted. These bits specify a total of N_m ($N_g \leq N_m \leq 2N_g$) maps to be made.

(d. Source Definitions to be thought about.)

A list of source coordinates (x,y in map points) and intensities (negative allowed). Depending on a switch these sources will be subtracted from the values V_i before gridding takes place; or they will be added (in this case, if no input data is specified it will cover the CLEAN-program). (NOTE: Needs probably better U,V definitions than standard precision of 16 nsec.)

4. Limitations

The description will be based on "normal" operation. For special users they will be different.

a. Gridder

Tables in AP-memory:

- Gridding functions: 3.5 x 64 + 1 = 225 words.
  i.e. identical functions for U,V, $n_g = 6$.
- Value table: 8 x 6 + 1 = 49
- Map table: 8 x 3 + 1 = 25
- (- Source list: n x 3 + 1)
- Various: < 101 words

400 words.

Remaining for gridded data: 65136 words.

Gridded Data:

- Each grid point will use up to 4 memory words:
  - Weighted, convolved real part
  - Weighted, convolved imaginary part
  - Convolved weights: to obtain antenna pattern and sending
  - Weights: to get natural weighting
- Per grid U_l lines should be in memory. If V_{max} is the maximum frequency in GHz, in the set of maps wanted; V_{min} the minimum frequency, then:

  $$U_l = IP\{U_p.v_{max}.X_r + U^1(V_{max} - V_{min}).X_r + U_g + 1\}$$

  Where $U^1 = |U|_{max}$ if given

  $= x_m/2/(v_{max}-V_{min})$ otherwise.

  For line-work at 21 cm with $u_g = 3.5$, and 1024 pigeon holes at A configuration;

  0°.5 field, $x_m = 2048$ (U_{max}: $U_l \approx 5 \approx 512$)

  For 2-band continuum ($V_{max} - V_{min} \sim .1$ GHz),

  $U_g = 6$, 1024 pigeon holes at A; 0°.15 field,

  $X_m = 2048$ $\qquad U_l \approx 11$

- In each line there will be V_l points:

  $$V_l = MAX[2*IP\{V^1v_{max}.Y_r+v_g+1\}, Y_m]$$

  where $V^1 = |V|_{max}$ if given

  $= Y_m/2/v_{max}/y_r$ otherwise.

  For above examples: $V_l \approx 1030$

  $V_l = 2048$

Each grid (map) takes: $4.U_1.V_1$ memory points.

In above examples ≈21000 resp. 90112 memory points. The last one is too large, and should be done in 2 passes. (If V_{max} was given changes are that only 45000 were necessary.) The total number of points necessary for the mapping is:

$$P = z_m * N_g * 4 * U_1 + V_1$$

(3rd dimension *# grids *...)

If P>65000 more than 1 pass is necessary. If the switch tells to force U,V plane coverage restriction, V_1 will be made equal to $\approx 65000/(z_m * N_g * 4 * U_1)$ if other restrictions do not work for multiple passes.

If $|V|_{max}$ is given, it follows from the above that, in general, for 3.5 point gridding about 3 frequency maps can be gridded simultaneously of 2048 * 2048. The transpose memory restricts it to one at a time.

b. Transpose Memory

The transpose memory limits the simultaneous throughput to:

$$Z_m * N_m * X^1 * Y_0$$

3rd dimension # maps/AP $|U|_{max}$ output size

i.e. in normal case, with U_{max} defined: 2 maps of 2048 x 2048
or 16 maps of 1024 x 512
or .....

c. Transform mechanisms

The 32k memory in the AP limits the maximum size (x_m, Y_m) in any direction to 8192 points (the remainder of the memory is for taper-resp. restoring function(s)).

5. Action if Limitations Found

a. X_m and/or Y_m > 8192: should not happen

b. If $X_m * N_m * X^1 * Y_0 > 4M$,

define a Y_0^1 so that $Z_m * N_m * X^{11} \leq 4M$

(for $N_m \leq 16$, $x^1 \leq 4096$ this limits Z_m to 64 to find a solution). A total of n_{p1} = CEIL (Y_0^1/Y_0) passes through the data are now necessary.

c. If the gridded data does not fit:

i.e. if $Z_m * N_g * 4 * U_1 * V_1 > 65136$

either - force a smaller V_1 (if told to do so by switch).

or - define a V_1^1 such that

$$Z_m * N_g * 4 * U_1 * V_1^1 \leq 65136$$

and make n_{p2} = CEIL (V_1/V_1^1) passes through input data.

6. Output (changed)

The final transform AP will output the map/antenna pattern to the 11/40 in an order:

map 0 - top line

map 1 - top line

2

.

.

n - top line

map 0 - next line

.

.

.

map n - bottom line

This has been chosen for:

a. Easy reference of transposing memory

b. Easy implementation of 3rd dimension

c. 11/40 won't notice any multiple passes through data.

7. Conclusions (changed)

The above proposal will produce a maximum of 8 simultaneous maps and/or antenna patterns of a maximum size of 8192 x 8192 points.

8. Speed

Convolution step: (double)

$60 + 5U_g + 10U_gV_g$ cycles/input point. (1 cycle = 1/6 M sec).

or, for $U_g = V_g = 3.5$: $\approx$ 33 M sec.

For 1 12-hr, 10 sec integration, 351 integers: 51 sec.

For 12 hr, 10 sec integration, 351 integers, 256 channels:

3.6 hr = 30% of real time. (1 pass).

Reading input data from disk

12 hr, 10 sec integration, 351 integers, 256 channels, 8 channels per input group, 12½% overhead = 1.7×10^8 bytes

speed $\approx$ 2000 bytes/30 nsec? = 7.3 hr!

$\approx$ 65000 bytes/sec?

Transforms (per AP), no 3rd dimension:

$2048 \times 2048 \lesssim 2048 \times 30$ M sec $\approx$ 63 sec

256 channels: $\approx$ 4.5 hr x 37% of real time (but in parallel with gridding)

Map writing

2048 x 2048 = 1.7 x 107 bytes

256 channels = 4.3 x 10^9 bytes

(66 kb/sec!) $\approx$ 17.9 hr! ($\approx$ 150% of realtime!)