Software for reading VIPIR raw data files

Terry Bullett[1,2]

26 December 2015

1. University of Colorado, CIRES, Boulder, CO
2. National Geophysical Data Center, NOAA/NGDC, Boulder, CO

Abstract

This document is a user manual for the readriq suite of programs that read raw data files from the Vertical Incidence Pulsed Ionosphere Radar (VIPIR), also known as the Scion HF radar. This software is a combination of shell scripts and FORTRAN95 code developed and run on Linux. The purpose of the software is to provide visualization and basic data analysis tools to the scientific user that are flexible. The software is open source and only relies on other open source packages. The user is highly encouraged to make improvements and send the improved code back to the author for integration into future releases.

1 Introduction

The Vertical Incidence Pulsed Ionospheric Radar (VIPIR) was developed under a Small Business Innovative Research grant from the US Air Force Research Laboratory to Scion Associates. The first installation was at the NASA Wallops Island Flight facility in 2008. There have since been a total of 15 instruments installed. A Version 2 of the radar system was developed by Scion Associates and the first operation of that instrument was in 2015. This software suite can be used on both generations of instruments. A technical descrition of the radar is given by Grubb et all [ref].

2 RIQ Files

The data output of the VIPIR is a Raw In-phase and Quadrature (RIQ) file that contains a number of range gate samples of the output of the Digital Down Converter for each of the 8 radar receive channels. There is an in-phase and quadrature sample for each range gate and receiver. In addition to the raw data blocks, each RIQ file contains meta-data records Sounding Control Table (SCT) and a Pulse Control Table (PCT) for each transmitted pulse. These define the instrument mode of operation and the site specific information such as station location and antenna configuration. The RIQ file is a binary file with a specific, custom format. The details of the data format are best described by C and FORTRAN structures, the listings for which are in Appendix **??**. Some attempt is made to document the details of each element in the RIQ file in Appendix B.**??**

3 readriq

The readriq package contains a number of shell scripts and executable codes. Shell scripts are always designated by a file name extension of .sh at the end of the command name. Executable code does not have such an extension. This document describes the code version 2.08 which is under development as of the end of 2015. Release versions of the code are available by FTP download at `ftp://ftp.ngdc.noaa.gov/ionosonde/software/VIPIR/` and development versions are available by request to the author.

The executable programs to be described are outlined in **??**

- dump_headers
- riq_ionogram

All program in the suite feature help by running the command with a **-h** help parameter. This gives information on how to use the program. The general format is **command options inputfile** Some shell commands generate output files based on the input file name, and these are reported to stdout to allow the calling script to manage the files. The information in the program's help should be considered difinitive over this document as the help information will surely be more up to date than this document. As always, the Code is King.

4 dump_headers

The simplest program in the suite is the **dump_headers** program. This program is written in FORTRAN95 using the g95 compiler under linux. Like all programs in the suite, it is command line code. The purpose of this code is to read the SCT and PCT of the RIQ file and display the variable names and the contents in text format on standard output. It also provides the option to create a binary file containing only the SCT and PCT. The primary use of **dump_headers** is to allow the user to view the contents of these meta-data structures. A secondary use is to allow shell scripts to have text-based access to the binary data in the SCT. This would generally be done by using a pipe of the standard output to programs such as **grep** and **sed** for selection of parameters.

The code requires one command line value, which is the name of the RIQ file to process. The command line options are shown in Listing 1. The output of dump-headers is long and a sample is given in Appendix 9.

To build this program from source code, there is a makefile. From the top level of the package, type:

`make -f Makefile.dump_headers clean`
`make -f Makefile.dump_headers`
will produce a fresh executable

Listing 1: **dump_headers** help

```
useage: dump_headers {options} filename
Version:    2.080
 Dump the contents of the Sounding Control Table (SCT) and
 Pulse Control Table (PCT) from the RIQ format file

  -b binary -- generate binary files of SCT and PCT, default=no
  -n Number of PCT blocks to read.  -1=all, 0=none,default=1
  Filename must be provided on the command line.
```

5 riq_ionogram

The heart of the readriq software suite is the riq_ionogram program. This program is written in FORTRAN95 using the g95 compiler under linux. The purpose of this code is turn the Level 0 raw data RIQ files into Level 1 ionogram data in various formats. The meaning of 'ionogram' is defined a two-dimensional gridded array of receiver output as a function of the 2 independent axes of an ionogram, these being transmit frequency on the horizontal axis and time delay on the vertical axis. The time delay is calibrated in km of virtual range using half the speed of as a scale factor:$R = c/2 * t$. Receiver output can be raw receiver output or raw data subject to several signal processing steps. Not all outputs are ionograms, as the code has numerous modes as defined in this document and the **-h** option. The code also allows access to the metadata in the file.

To build this program from source code, there is a makefile. From the top level of the package, type:

`make clean`
`make`
will produce a fresh executable. This is the only code in the suite that uses the default Makefile. Note there may be a symbolic link to a linux distribution based Makefile, due to the troubles caused by netCDF. See the section on netCDF for details.

The help information for **riq_ionogram** is extensive and shown in Listing 2.

Listing 2: **riw_ionogram** help

```
  useage: riq_ionogram {} filename

Version:    2.080
 Decode VIPIR HF Radar Raw I/Q data file into ionogram image data

  -U URSICODE Set (over-ride) the station URSI code.
  -r Raw plot of A-scans between N1 and N2. Ignores shuffle settings
  -s Stacked raw plot of A-scans between N1 and N2.  Follows shuffle settings
  -R RTI mode processing. Integrate over pulsets only
  -S {n} Shuffle-mode data with {n} interleaves
  -P {i} Pick the {i}th shuffled sequence to process.  i=1,2..n
  -G {n} Select only range gate n (raw modes)
  -I Enable Simple Phase Interferometry (SPID). Re-defines raw mode.
  -D Enable dP/dt Doppler calculations.  Forces -R RTI mode
  -N {filename} Create netCDF file
  -R0 {Range0} Set range gate offset
  -n1 First Number
  -n2 Second Number
  -t {x} Set Threshold to {x}
  -cf {configuraton_file}  Read metadata from this file
  -pr {chan} Set phase reference, relative given {chan}
  ------------------
  -p Peak Detect (for raw data)
  -i Ionogram (incoherent)
  -c Ionogram (coherent)
  -sa Spectrum Analyzer mode.  -c or -i needs to preceed!
  -pdf output the signal amplitude probability distribution function
  -d Debug info to stdout
```

```
-dd Dump intermediate results to stdout
-cal {method} ; Current methods are:
    0 - No calibration (default)
    1 - Simple cable length difference phase adjustment
        Others TBD
-NG {n} Set # of range gates = n, for corrupt v0 files <<- switch change
-m {n} Set Rx Mask to n
-n {x} Set NoiseMod to x
-a {n} Select single receiver n
-fp {n} Mask for flipping phase by 180 degrees (mis-wired antennas)
-f1 Lower frequency select band [kHz]
-f2 Upper frequency select band [kHz]
-r1 Lower range for statistics and raw output Default=30 [km]
-r2 Upper range for statistics and raw output  [km]
    Negative values are relative to the maximum range in the a-scan
-l  Limit range gate output to r1:r2 above, but as integer range gates
-ba Blank between each Antenna or A-scan
-bp Blank between eack PRI
-bf Blank between each Frequency
-bi Blank between each Ionogram
-v Verbose
Filename must be provided on the command line.

Returns exit status as follows:
 0:  No Problems
-1:  Problems
```

The **riq_ionogram** program has several major modes of operation, and numerous command line switches and parameters which control how these modes operate on the data, and also control the data output format. In general, the data output format is designed to be easily read by the `gnuplot` program for visualization. Data are by default sent to the standard output, so it is normal to pipe or re-direct this handle to another program or to a file.

- Ionogram – Create an array of amplitude vs frequency and range.
- Raw – Output raw I/Q and Amplitude/Phase data.
- RTI – Treat the data as Range-Time-Intensity, normally for fixed frequency data.
- Spectrum Analyzer – Reduce the data in the range dimension to give signal vs frequency.
- PDF – Generate Probability Distribution Functions of data.
- Echo Detector – Detect echos from the raw data. Under development.
- Interferometry – Determine arrival angles and Doppler of radio waves. Under development.

Each of these modes are described.

The remaining command line switches are ones which are useful and necessary to control the function of the various routines.

5.1 Debugging switches

The switches -v (verbodse) -d (debug) and -dd (dump) provide 3 independent controls on the internals of the program. Since both routine outout and this extra information both go to the standard output, these switches are only turned on when the user is looking at the code internal functions and trying to fix the code. Most of the output is labled, and there are many references to internal variables in the code.

5.2 Format Control

The `gnuplot` program uses blank lines between columns of numbers as data separators for both 2D and 3D plots. The **riq_ionogram** code allows control over the use of blank lines as data seprators. These are relatively self-explanatory.

- -ba Blank between each Antenna or A-scan. An a-scan is a group of range bins from a single pulse and single receiver.
- -bp Blank between eack PRI. A Pulse Repetition Interval is all receivers taken simultaneously for a single transmitted pulse.
- -bf Blank between each Frequency. A Frequency is the nominal frequency in the measurement and includes all pulses from a pulset or a ramo pr repeat taken at that frequency. Offests in the pulset frequency vector are ignored.
- -bi Blank between each Ionogram. The RIQ file can contain any number of ionograms repeated without any delay between them. Note this code is generally not capable of handling such data sets

Note also that all headers start with the "#" character, which is ignored by gnuplot as a comment line.

5.3 Data Selection

These command line parameters control the selection of data to be processed or displayed.

- -U URSI – Define the URSI code unstead of using the one in the SCT. Required for old Version 0 format data
- -S n Shuffle-mode data with n interleaves. Normally a shuffle mode ionogram can be detected and this is determined from the SCT, but in cases where the SCT is wrong, this can over-ride the SCT value

- -P i Pick the ith shuffled sequence to process. i=1,2..n. This allows to select one of the shuffled sequences to process.
- -G n Select only range gate n (raw modes). Only output one range gate for each A-scan
- -n1 First Number Generally used for raw data extraction, to select which the first PRI to print.
- -n2 Second Number Similar to n1
- -f1 Lower frequency select band [kHz] Sets the lower boundry of data processing in the frequency domain
- -f2 Upper frequency select band [kHz] Sets the upper boundry of data processing in the frequency domain
- -r1 Lower range for statistics and raw output Default=30 [km] Sets the lower range boundry for computing noise and peak values in ionogram modes, and sets the lower range gate for data output in raw modes.
- -r2 Upper range for statistics and raw output [km] Sets the upper range boundry for computing noise and peak values in ionogram modes, and sets the upper range gate for data output in raw modes. A negative r2 is relative to the maximum range in the a-scan. This is useful for removing the transmit pulse at low ranges and the calibrataion pulse at high ranges.
- -l Limit range gate output to r1:r2 above, but as integer range gates. This implements the range boundries for raw modes. It's not sufficient to just define them for raw data output

5.4 Processing Control

These switches control how the data are processed.

- -m n Set Rx Mask to n. This is a binary bitmask, where a 1 indicates a receiver is to be used and a 0 it is to be excluded. But this has to be passed to the code as an integer. The least significnat bit is Channel 1. So the binary value 2%00000011 = 3 would pnly use channels 1 and 2 in the processing. The default is 2%11111111 = 255 which is process all channels.
- -a n Select single receiver n A shorthand way to select a single receive channel. This is converted into a Rx Mask inside the program.
- -fp n Mask for flipping phase by 180 degrees (mis-wired antennas). This is a bitmask similar to Rx Mask, but a 1 in the binary number indicates a 180° phase change for that antenna. Useful where stations have mis-wired twinaxial cables, preamps or rotated antennas. The default is 2%00000000 = 0 no phase flips.
- -R0 Range0 Set range gate offset. This sets the true range of the imaginary 0^{th} range gate to be 0 km time of flight. See the section on Range Calibration
- -cal method ; Current methods are:
  0 - No calibration (default)
  1 - Simple cable length difference phase adjustment
  This refers to use of the calibration pulse in the VIPIR data and is still preliminary, experimental, and not recommended for general use. Neither of these options actually use the calibration data. Option 1 applies a simple cable length difference that needs to be programmed into the source code. This feature will be developed further in the future.

5.5 Range Calibration

The range calibration for the VIPIR is determined experimentally from looking at multiple hop sporadic-E layers. There are default values of this set into some of the codes. Since the received waveform often occupies multiple range gates, usually 6 to 8 depending on the For the VIPIR, it is convention that the peak of the receiver impulse response is "the" range of the echo. Also note there are deficiencies in this version of the code that does not correctly take into account the holdoff and start range settings in either the Version 1 and especially not the Version 2 VIPIR. Some skill and caution is necessary to use this control properly

Include a diagram explaining this.

5.6 Noise

The noise algorithm in **riq_ionogram** is based on statsitics of environmental noise, man-made interference, and ionosphere echos, and generally works well for these instances. The standard noise algorithm performs probability distribution functions (PDF) on a selected data set. That selection is defined by the mode of operation and can be raw data or data that are integrated and averaged for all PRI's at the same nominal frequency, or just integration over a pulset, as with the RTI mode. In general, these data are amplitude samples as a function of range gate. A PDF is defined with default bins of 1dB each over a range of amplitudes that exceed the dynamic range of the radar, default is -20 dB to 160 dB, with 0dB being defined as the senstiivity threshold of the Version 1 VIPIR receiver. This version of the software scales the Version 2 VIPIR receiver to the same approximate scale. For this version, none of the analog front end settings are considered and a calibration to an absoulute standard is not provided. Each dB amplitude value is sorted into it's corresponding bin, and the number of samples in each bin is accumulated. The PDF is then the output of these counts as a function of amplitude. The default noise value is then the mode of this distribution, often called the Most Probable Amplitude (MPA). The **-n x** parameter allows a a modification of the MPA based on the data in the PDF. The value of **x** can take on values from > 0 to ± 1 and works in the following way. Instead of the peak of the PDF, a noise value of $|x| * MPA$ is selected. Positive values of **x** require the noise value be greater than MPA and search the PDF in the increasing amplitude direction and increase the noise value, while negative values of **x** search the PDF below the MPA to provide lower estimates of the noise. Values of 0 are not recommended.

This noise model produces less than desireable results in the presence of pulsed interference, such as nearby radar systems or impulsive interference from arcing high voltage power systems. It also fails during instances of range spreading in the ionogram where the number of range gates containing signal from the ionosphere plasma exceed the number of range gates containing noise. New algorithms are required for such conditions.

5.7 Ionogram

In the ionogram mode, the I/Q data from all receivers is summed either coherently (-c) in a vector sum or incoherently (-i) by summing the powers of all receivers.

The output is independent for each range bin and nominal frequency. For modes with multiple pulses at each nominal frequency, which is most modes, summation over time is an incoherent power average. In all cases, the inchoerent "average" is averaged amplitude in dB, so mathematiclly it is the geometric mean of the data samples over time and receivers.

An Ordinary/eXtraordinary (O/X) mode is included with either -c or -i mode, and the O/X mode is by definition a phase coherent summation. For this computational mode, the even numbered radar channels are phase shifted both plus 90° and −90° and then coherently added with the data from the odd numbered channels. This is for circularly polarized O/X radio waves more than a few degrees off the magnetic equator. For a magnetic equator station, O and X modes are linearly polarized parallel and perpendicular to the magnetic field direction, so a vector sum and difference is done without a phase shift. Time averages are again the average amplitudes of the vector sum, averaged in dB.

A sample of the text format output for ionogram modes is given in Listing 3. It is a column format, with these comumns. Additional columns can and will be added in the future.

1. IFRQ – Index of the nominal frequency in the ionogram.
2. IRNG – Range Gate Number.
3. UT_Time – The start time of the ionogram, in seconds, relative to the start of the UT day.
4. dt – Time in seconds from the start of the ionogram to the first PRI in the summation.
5. Freq – Nominal sounding frequency, in MHz
6. Range – Calibrated range, in km. Note the negative numbers and refer to the RANGE0 parameter
7. TPower – Total integrated power, in dB relative to receiver threshold
8. TSNR – Total Signal to Noise Ratio, in dB This has the noise subtracted. See the Noise section
9. OPower – Ordinary mode power, in dB.
10. OSNR – Ordinary mode Signal to Noise Ratio, in dB
11. Xpower – eXtraordiary mode power, in dB
12. XSNR – eXtraordianry mode Signal to Noise Ratio, in dB
13. DPDT – Change of phase with time, a gateway to Doppler shifts *[UNDER DEVELOPMENT]*
14. DPDTv – Doppler Velocity from DPDT *[UNDER DEVELOPMENT]*

Listing 3: Ionogram mode output

```
\#   IFRQ    IRNG       UT_Time           dt          Freq         Range  TPower     TSNR  OPower     OSNR  XPower     XSNR
DPDT    DPDTv
        1       1   54424.01953      0.00000      1.00000      -4.50104   92.22    64.22   97.53    64.53   91.98    60.98
0.00     0.00
        1       2   54424.01953      0.00000      1.00000      -3.00208   92.40    64.40   97.32    64.32   92.57    61.57
0.00     0.00
        1       3   54424.01953      0.00000      1.00000      -1.50311   92.43    64.43   97.27    64.27   92.68    61.68
0.00     0.00
        1       4   54424.01953      0.00000      1.00000      -0.00415   92.42    64.42   97.28    64.28   92.67    61.67
0.00     0.00
        1       5   54424.01953      0.00000      1.00000       1.49481   92.34    64.34   97.38    64.38   92.64    61.64
0.00     0.00
        1       6   54424.01953      0.00000      1.00000       2.99377   92.06    64.06   97.68    64.68   91.98    60.98
0.00     0.00
        1       7   54424.01953      0.00000      1.00000       4.49274   90.50    62.50   97.04    64.04   87.33    56.33
0.00     0.00
        1       8   54424.01953      0.00000      1.00000       5.99170   80.07    52.07   86.60    53.60   70.23    39.23
0.00     0.00
        1       9   54424.01953      0.00000      1.00000       7.49066   59.08    31.08   65.44    32.44   47.16    16.16
0.00     0.00
        1      10   54424.01953      0.00000      1.00000       8.98962   27.93    -0.07   32.68    -0.32   30.65    -0.35
0.00     0.00
        1      11   54424.01953      0.00000      1.00000      10.48858   29.62     1.62   37.37     4.37   26.81    -4.19
0.00     0.00
        1      12   54424.01953      0.00000      1.00000      11.98755   28.77     0.77   35.08     2.08   31.27     0.27
0.00     0.00
        1      13   54424.01953      0.00000      1.00000      13.48651   29.61     1.61   33.13     0.13   34.35     3.35
0.00     0.00
        1      14   54424.01953      0.00000      1.00000      14.98547   27.91    -0.09   31.96    -1.04   31.17     0.17
0.00     0.00
        1      15   54424.01953      0.00000      1.00000      16.48443   28.79     0.79   31.55    -1.45   33.72     2.72
0.00     0.00
```

The -N *[filename]* flag causes the code to generate a NetCDF Gridded Ionogram (.NGI) format output file with the required *[filename]*. This flag is only meaningful for ionogram modes of data processing.

6 netCDF

In order to comply with NOAA standards for data archiving and metadata content, a netCDF version of ionogram data was created. In theory this self-documenting universal data format should be readable by any software package that understands netCDF. In practice, these files are little used. The inclusion of netCDF and the libraries required for FORTRAN have required substantial effort when moving to different versions and releases of Linux. Various methods of handling this are included in the numerous generations of the Makefiles for each release. The 2.08 version Makefile works under Debian 8 "jessie" architecture amd64 and is not backward compatible with Debian 7 "wheezy". A separate Makefile is maintained for that target platform. For backward compatibility, it may be necessary to adapt the Makefiles from older versions of the code.

4.

Listing 4: Tabulated Frequency Example

```
 <FrequencyStep Units="MHz" Type="tabulated" Num="7">
1.00,1.50,2.75,3.85,6.67,8.83,12.45
</FrequencyStep>
```

APPENDICIES

A Sounding Control Table

The Sounding Control Table is a C and FORTRAN structure. Both structure formats are supported, and produce nearly idential files. The exception is for the user-defined text strings, where C prodices a null filled character string and FORTRAN produces a space filled character string. Both methods are supported. For 64 bit C code, it is necessary to define the structure as “packed”

Listing 5: SCT Structure, C

```
#define VERSION 1.20            // update version
#define MAGIC 0x51495200        // from "null""RIQ"
#define SYSREF 120000000        // system clock reference, MHz
#define PATH   64               // path name length for various files
#define USER   128              // user defined area, each substructure
#define RXANTMX 32              // maximum receive antenna count per station
#define FREQMX 8192             // maximum number of base frequencies (not total frequencies)
#define BAUDMX 1024             // maximum baud count in data- and cal waveform strings
#define PULSEMX 256             // maximum pulseset element count
#define QUIETMX 64              // maximum number of quiet bands
#define MEMMX 24320             // maximum size OCRAM


typedef struct                                  // sct...
{
   int            magic;                // magic number for VIPIR
   int            sounding_table_size;  // bytes in sounder configuration structure (this file)
   int            pulse_table_size;        // bytes in pulse configuration structure
   int            raw_data_size;        // bytes in raw data block (one PRI)
   float               struct_version;          // per #define above
   int            start_year;           // start time of ionogram
   int            start_daynumber;              // start time of ionogram
   int            start_month;          // start time of ionogram
   int            start_day;            // start time of ionogram
   int            start_hour;           // start time of ionogram
   int            start_minute;         // start time of ionogram
   int            start_second;         // start time of ionogram
   unsigned int  start_epoch;           // start time of ionogram
//   unsigned int  ref_usec;                    // real time clock read at start
   char                readme[128];             // information or operator comments
   int            decimation_method;    // if processed, 0 = no process
   float               decimation_threshold;    // if processed
   char                user[128];               // user defined (spare)


   struct                                       // sct.station...
   {
      char       file_id[PATH];                 // name of station settings file
      char       ursi[8];                       // station URSI designation (input)
      char       rx_name[32];                   // station name (input)
      float      rx_latitude;                   // array reference latitude deg (input)
      float      rx_longitude;                  // array reference longitude deg (input)
      float      rx_altitude;                   // array reference altitude m (input)
      int   rx_count;                   // antennas at station (input, up to RXANTMX)
      char       rx_antenna_type[RXANTMX][32];  // antenna NAME text descriptors (input)
      float      rx_position[RXANTMX][3];       // antenna placement x, y, z (input)
      float      rx_direction[RXANTMX][3];      // antenna element direction (input)
      float      rx_height[RXANTMX];            // antenna height above reference ground m (input)
      float      rx_cable_length[RXANTMX];      // physical length of receive cables (input)
      float      frontend_atten;                // frontend attenuator setting (input)
      char       tx_name[32];                   // station name (input)
      float      tx_latitude;                   // transmit antenna latitude deg (input)
      float      tx_longitude;                  // transmit antenna longitude deg (input)
      float      tx_altitude;                   // transmit antenna altitude m (input)
      char       tx_antenna_type[32];           // antenna NAME descriptors (input)
      float      tx_vector[3];                  // antenna vector (input)
      float      tx_height;                     // antenna height above reference ground m (input)
      float      tx_cable_length;               // physical length of receive cables (input)
      int   drive_band_count;           // number of blocked bands (derived)
//      float      drive_band_bounds[QUIETMX][2];        // antenna drive bands (input)
      float      drive_band_bounds[QUIETMX];    // antenna drive bands (input)
      float      drive_band_atten[QUIETMX];     // antenna drive atteunuation in dB (input)
      int   rf_control;                    // -1 = none, 0 = drive/quiet, 1 = full, 2 = only quiet, 3 = only atten
      int   lpf_freq_count;             // number of LPF switches (derived)
      float      lpf_freq_switch[10];           // LPF switch frequencies (input)
      char       ref_type[32];                  // OCXO or bistatic PLO (input)
      char       clock_type[32];                // NTP, GPSD, etc. (input)
//      float      clock_frequency;             // reference sample rate
      char       user[340];                     // user defined (spare)
   } station;

   struct                                       // sct.timing....
   {
```

```
   char       file_id[PATH];                  // name of time settings file
   float      pri;                            // PRI period us (input)
   int   pri_count;                   // total number of PRIs in ionogram (dervied)
   int   ionogram_count;              // repeat count for ionogram within same data file (input)
   float      holdoff;                        // time between GPS 1 pps and start (fixed)
   float      range_gate_offset;              // true range to gate 0
   int   gate_count;                  // number of gates, adjusted up for USB blocks (adjusted)
   float      gate_start;                     // start gate placement us, adjusted (adjusted)
   float      gate_end;                       // end gate placement us, adjusted (adjusted)
   float      gate_step;                      // range delta us (derived)
   float      data_start;                     // data pulse range placement start us (input)
   float      data_width;                     // data pulse baud width us (input)
   int   data_baud_count;             // data pulse baud count (input)
   char       data_wave_file[PATH];           // alternative baud pattern file name
   float      data_baud[BAUDMX][2];           // data waveform baud pattern  X, Y (input or from file)
   int   data_pairs;                  // number of IQ pairs in waveform memory (derived)
   float      cal_start;                      // cal range placement start us (input)
   float      cal_width;                      // cal pulse baud width us (input)
   int   cal_baud_count;              // data pulse baud count (input)
   char       cal_wave_file[PATH];            // alternative baud pattern file name
   float      cal_baud[BAUDMX][2];            // calibration waveform baud pattern X, Y (input or from file)
   int   cal_pairs;                   // number of IQ pairs in waveform (derived)
   char       user[128];                      // user defined (spare)
} timing;


struct                                        //sct.frequency...
{
   char       file_id[PATH];                  // name of frequency settings file (input)
   float      base_start;                     // start frequency for log or linear (input)
   float      base_end;                       // end frequency for log or linear (input)
   int   base_steps;                  // computed for log, linear or from table read (derived)
   int   tune_type;                   // 0 = fixed, 1 = log, 2 = linear, 3 = table
   float      base_table[FREQMX];             // base frequencies pre-pulseset, and action (derived or loaded)
   float      linear_step;                    // currently in kHz (input)
   float      log_step;                       // currently in percent (input)
   char       freq_table_id[PATH];            // manual tuning table name (input)
   int   tune_steps;                          // all frequencies pre-ramp repeats (derived)
   int   pulse_count;                         // pulse set (input)
   int   pulse_pattern[PULSEMX];      // pulse set (input)
   float      pulse_offset;                   // pulse set offset kHz (input)
   int   ramp_steps;                  // pulse set count before repeat (input)
   int   ramp_repeats;                        // repeat count of pulse set steps (nput)
   float      drive_table[FREQMX];            // base frequencies attenuation/silent table (derived)
   char       user[128];                      // user defined (spare)
} frequency;

struct                                        // sct.receiver...
{
   char        file_id[PATH];                 // name of DDC settings file
   int    rx_chan;                    // number of channels being used (input)
   int    rx_map[16];                         // channel-to-antenna mapping (input)
   int         word_format;                   // 0 = big endian fixed, 1 = little endian, 2 = float (input)
   int    cic2_dec;                   // DDC filter block (input)
   int    cic2_interp;                        // DDC filter block (input)
   int    cic2_scale;                         // DDC filter block (input)
   int    cic5_dec;                   // DDC filter block (input)
   int    cic5_scale;                         // DDC filter block (input)
   char        rcf_type[32];                  // text descriptor of FIR filter block (input)
   int    rcf_dec;                    // decimation factor for FIR filter block (input)
   int    rcf_taps;                   // number of taps in FIR filter block (input)
   int    coefficients[160];          // FIR filter coefficients (input)
   float       analog_delay;                  // analog delay of receiver, us
   char        user[128];                     // user defined (spare)
} receiver;

struct                                        // sct.exciter...
{
   char        file_id[PATH];                 // name of DDC settings file
   int    cic_scale;                  // DUC filter block (input)
   int    cic2_dec;                   // DUC filter block (input)
   int    cic2_interp;                        // DUC filter block (input)
   int    cic5_interp;                        // DUC filter block (input)
   char        rcf_type[32];                  // text descriptor of FIR filter block (input)
   int    rcf_taps;                   // number of taps in FIR filter block (input)
   int    rcf_taps_phase;             // number of taps in FIR filter block (input)
   int    coefficients[256];          // FIR filter coefficients (input)
   float       analog_delay;                  // analog delay of exciter, us
   char        user[128];                     // user defined (spare)
  } exciter;
struct                                        //sct.monitor...
{
   int    balun_currents[8];                  // as read prior to ionogram
   int    balun_status[8];            // as read prior to ionogram
   int    front_end_status[8];                // as read prior to ionogram
   int    receiver_status[8];                 // as read prior to ionogram
```

```
      int    exciter_status[2];         // as read prior to ionogram
      char        user[512];                    // user defined (spare)
   } monitor;

} SCT;
```

Listing 6: SCT Structure, FORTRAN

```
!
! Define the Sounding Configuration Table (SCT) structure for Version 1.2 of the
! Raw Inphase & Quadrature (RIQ) file format for the Scion HF Radar (VIPIR)
!
!
! T. Bullett    04Nov08   g95
! Adapted from:
! R. Livingston  sct.h    gcc
!
! The sct element, Version 1.2 of RIQ data file.
! This contains information about the station of observation.  Relatively static.
!
! Version 1.2  04Nov08
! Version 2.0  07Dec15

TYPE :: STATIONtype
CHARACTER(64)   :: file_id      ! name of station settings file
CHARACTER( 8)   :: ursi_id      ! URSI standard station ID code
CHARACTER(32)   :: rx_name      ! Receiver Station Name
REAL(KIND=4)    :: rx_latitude  ! Position of the Receive array reference point [degrees North]
REAL(KIND=4)    :: rx_longitude !                                               [degrees East]
REAL(KIND=4)    :: rx_altitude  ! meters above mean sea level
INTEGER(KIND=4) :: rx_count     ! Number of defined receive antennas
CHARACTER(32),DIMENSION(32)  :: rx_antenna_type ! Rx antenna type text descriptors
REAL(KIND=4),DIMENSION(3,32) :: rx_position     ! X,Y,Z = (East,North,Up) Positon [m] of each Rx
REAL(KIND=4),DIMENSION(3,32) :: rx_direction    ! X,Y,Z = (East,North,Up) Direction of each Rx
REAL(KIND=4),DIMENSION(32)   :: rx_height       ! Height above ground [m]
REAL(KIND=4),DIMENSION(32)   :: rx_cable_length ! physical length of receive cables [m]
REAL(KIND=4)    :: frontend_atten  ! Front End attenuator setting
CHARACTER(32)   :: tx_name         ! Transmitter Station Name
REAL(KIND=4)    :: tx_latitude     ! Position of the Transmit Antenna reference point [degrees North]
REAL(KIND=4)    :: tx_longitude    !                                                   [degrees East]
REAL(KIND=4)    :: tx_altitude     ! meters above mean sea level
CHARACTER(32)   :: tx_antenna_type ! Tx antenna type text descriptors
REAL(KIND=4),DIMENSION(3) :: tx_vector ! tx antenna direction vector [m]
REAL(KIND=4)    :: tx_height           ! antenna height above reference ground [m]
REAL(KIND=4)    :: tx_cable_length     ! physical length of transmit cables [m]
INTEGER(KIND=4) :: drive_band_count    ! Number of antenna drive bands
REAL(KIND=4),DIMENSION(2,64) :: drive_band_bounds ! drive bands start/stop in kHz
REAL(KIND=4),DIMENSION(64)   :: drive_band_atten  ! antenna drive atteunuation in dB
INTEGER(KIND=4) :: rf_control   ! -1 = none, 0 = drive/quiet, 1 = full, 2 = only quiet, 3 = only atten
CHARACTER(32)   :: ref_type     ! Type of reference oscillator
CHARACTER(32)   :: clock_type   ! Source of absoulte UT timing
CHARACTER(128)  :: user         ! Spare space for user-defined information
END TYPE STATIONtype

! Timing of the measurement
TYPE :: TIMINGtype   ! Time values are in microseonds unless otherwise indicated
CHARACTER(64)   :: file_id           ! Name of the timing settings file
REAL(KIND=4)    :: pri               ! Pulse Repetition Interval (PRI) (microseconds)
INTEGER(KIND=4) :: pri_count         ! number of PRI's␣in␣the␣measurement
INTEGER(KIND=4)␣::␣ionogram_count␣␣␣␣!␣repeat␣count␣for␣ionogram␣within␣same␣data␣file
REAL(KIND=4)␣␣␣␣::␣holdoff␣␣␣␣␣␣␣␣␣␣␣!␣time␣between␣GPS␣1␣pps␣and␣start
REAL(KIND=4)␣␣␣␣::␣range_gate_offset␣!␣true␣range␣to␣gate␣0
INTEGER(KIND=4)␣::␣gate_count␣␣␣␣␣␣␣␣!␣Number␣of␣range␣gates,␣adjusted␣up␣for␣USB␣blocks
REAL(KIND=4)␣␣␣␣::␣gate_start␣␣␣␣␣␣␣␣!␣start␣gate␣placement␣[us],␣adjusted
REAL(KIND=4)␣␣␣␣::␣gate_end␣␣␣␣␣␣␣␣␣␣!␣end␣gate␣placement␣[us],␣adjusted
REAL(KIND=4)␣␣␣␣::␣gate_step␣␣␣␣␣␣␣␣␣!␣range␣delta␣[us]
REAL(KIND=4)␣␣␣␣::␣data_start␣␣␣␣␣␣␣␣!␣data␣range␣placement␣start␣[us]
REAL(KIND=4)␣␣␣␣::␣data_width␣␣␣␣␣␣␣␣!␣data␣pulse␣baud␣width␣[us]
INTEGER(KIND=4)␣::␣data_baud_count␣␣␣!␣data␣pulse␣baud␣count
CHARACTER(64)␣␣␣::␣data_wave_file␣␣␣␣!␣data␣baud␣pattern␣file␣name
COMPLEX(KIND=4),DIMENSION(1024)␣::␣data_baud␣!␣data␣waveform␣baud␣pattern
INTEGER(KIND=4)␣::␣data_pairs␣␣␣␣␣␣␣␣!␣number␣of␣IQ␣pairs␣in␣waveform␣memory
REAL(KIND=4)␣␣␣␣::␣cal_start␣␣␣␣␣␣␣␣!␣cal␣range␣placement␣start␣[us]
REAL(KIND=4)␣␣␣␣::␣cal_width␣␣␣␣␣␣␣␣!␣cal␣pulse␣baud␣width␣[us]
INTEGER(KIND=4)␣::␣cal_baud_count␣␣␣!␣cal␣pulse␣baud␣count
CHARACTER(64)␣␣␣::␣cal_wave_file␣␣␣␣!␣alternative␣baud␣pattern␣file␣name
COMPLEX(KIND=4),DIMENSION(1024)␣::␣cal_baud␣!␣cal␣waveform␣baud␣pattern
INTEGER(KIND=4)␣::␣cal_pairs␣␣␣␣␣␣␣␣!␣number␣of␣IQ␣pairs␣in␣waveform␣memory
CHARACTER(128)␣␣::␣user␣␣␣␣␣␣␣␣␣␣␣␣␣␣!␣Spare␣space␣for␣user-defined␣information
END␣TYPE␣TIMINGtype
!
!␣Frequency␣information␣about␣the␣measurement
!
TYPE␣::␣FREQUENCYtype␣␣!␣Values␣are␣in␣kilohertz␣unless␣otherwise␣indicated
CHARACTER(64)␣␣::␣file_id␣␣␣␣␣␣!␣Frequency␣settings␣file
```

```
REAL(KIND=4)   :: base_start   ! Initial base frequency
REAL(KIND=4)   :: base_end     ! Final base frequency
INTEGER(KIND=4):: base_steps   ! Number of base frequencies
INTEGER(KIND=4):: tune_type    ! Tuning type flag:  1=log, 2=linear, 3=table, 4=Log+Fixed ShuffleMode
REAL(KIND=4),DIMENSION(8192) :: base_table ! Nominal or Base frequency table
REAL(KIND=4)   :: linear_step   ! Linear frequency step [kHz]
REAL(KIND=4)   :: log_step      ! Log frequency step, [percent]
CHARACTER(64)  :: freq_table_id ! Manual tuning table filename
INTEGER(KIND=4):: tune_steps    ! all frequencies pre-ramp repeats
INTEGER(KIND=4):: pulse_count   ! pulset frequency vector length
INTEGER(KIND=4),DIMENSION(256) :: pulse_pattern ! pulset frequency vector
REAL(KIND=4)   :: pulse_offset  ! pulset offset [kHz]
INTEGER(KIND=4):: ramp_steps    ! pulsets per B-mode ramp (ramp length, base freqs per B-block)
INTEGER(KIND=4):: ramp_repeats  ! repeat count of B-mode ramps
REAL(KIND=4),DIMENSION(8192):: drive_table ! base frequencies attenuation/silent table
CHARACTER(128) :: user          ! Spare space for user-defined information
END TYPE FREQUENCYtype
!
!
! Receiver Settings
TYPE :: RECEIVERtype
CHARACTER(64)   :: file_id       ! Frequency settings file
INTEGER(KIND=4) :: rx_chan       ! Number of receivers
INTEGER(KIND=4),DIMENSION(16) :: rx_map ! receiver-to-antenna mapping
INTEGER(KIND=4) :: word_format   ! 0 = big endian fixed, 1 = little endian, 2 = floating_point, 3=32 bit little endian integer
INTEGER(KIND=4) :: cic2_dec      ! DDC filter block
INTEGER(KIND=4) :: cic2_interp   ! DDC filter block
INTEGER(KIND=4) :: cic2_scale    ! DDC filter block
INTEGER(KIND=4) :: cic5_dec      ! DDC filter block
INTEGER(KIND=4) :: cic5_scale    ! DDC filter block
CHARACTER(32)   :: rcf_type      ! text descriptor of FIR filter block
INTEGER(KIND=4) :: rcf_dec       ! decimation factor for FIR filter block
INTEGER(KIND=4) :: rcf_taps      ! number of taps in FIR filter block
INTEGER(KIND=4), DIMENSION(160) :: coefficients   ! Receiver filter coefficients
REAL(KIND=4)    :: analog_delay  ! analog delay of receiver, us
CHARACTER(128)  :: user          ! Spare space for user-defined information
END TYPE RECEIVERtype
!
!
! Exciter Settings
TYPE :: EXCITERtype
CHARACTER(64)   :: file_id        ! Frequency settings file
INTEGER(KIND=4) :: cic_scale      ! DUC filter block
INTEGER(KIND=4) :: cic2_dec       ! DUC filter block
INTEGER(KIND=4) :: cic2_interp    ! DUC filter block
INTEGER(KIND=4) :: cic5_interp    ! DUC filter block
CHARACTER(32)   :: rcf_type       ! text descriptor of FIR filter block
INTEGER(KIND=4) :: rcf_taps       ! number of taps in FIR filter block
INTEGER(KIND=4) :: rcf_taps_phase ! number of taps in FIR filter block
INTEGER(KIND=4), DIMENSION(256) :: coefficients   ! Receiver filter coefficients
REAL(KIND=4)    :: analog_delay   ! analog delay of exciter/transmitter, us
CHARACTER(128)  :: user           ! Spare space for user-defined information
END TYPE EXCITERtype
!
!
! System status and Built-In-Test info
TYPE :: MONITORtype
INTEGER(KIND=4),DIMENSION(8) :: balun_currents  ! As read prior to ionogram
INTEGER(KIND=4),DIMENSION(8) :: balun_status    ! As read prior to ionogram
INTEGER(KIND=4),DIMENSION(8) :: front_end_status! As read prior to ionogram
INTEGER(KIND=4),DIMENSION(8) :: receiver_status ! As read prior to ionogram
INTEGER(KIND=4),DIMENSION(2) :: exciter_status  ! As read prior to ionogram
CHARACTER(512) :: user                            ! Spare space for user-defined information
END TYPE MONITORtype
!
! Top level Sounding Configuration Table, Version 1.2
!
TYPE :: SCTtype
INTEGER(KIND=4) :: magic                 ! magic number 0x51495200 (/nullRIQ) {POSSIBLY BYTE REVERSED}
INTEGER(KIND=4) :: sounding_table_size   ! bytes in sounder configuration structure (this file)
INTEGER(KIND=4) :: pulse_table_size      ! bytes in pulse configuration structure
INTEGER(KIND=4) :: raw_data_size         ! bytes in raw data block (one PRI)
REAL(KIND=4)    :: struct_version        ! Format Version Number.  Currently 1.2
INTEGER(KIND=4) :: start_year            ! Start Time Elements of the ionogram (Universal Time)
INTEGER(KIND=4) :: start_daynumber       !
INTEGER(KIND=4) :: start_month           !
INTEGER(KIND=4) :: start_day             !
INTEGER(KIND=4) :: start_hour            !
INTEGER(KIND=4) :: start_minute          !
INTEGER(KIND=4) :: start_second          !
INTEGER(KIND=4) :: start_epoch           ! epoch time of the measurement start.
CHARACTER(128)  :: readme                ! Operator comment on this measurement
INTEGER(KIND=4) :: decimation_method     ! If processed, 0=no process (raw data)
REAL(KIND=4)    :: decimation_threshold  ! If processed, the treshold value for the given method
CHARACTER(128)  :: user                  ! user-defined
```

```
TYPE(STATIONtype)␣␣␣::␣station␣␣␣␣␣␣␣!␣Station␣info␣substructure
TYPE(TIMINGtype)␣␣␣␣::␣timing␣␣␣␣␣␣␣␣!␣Radar␣timing␣substruture
TYPE(FREQUENCYtype)␣::␣frequency␣␣␣␣␣!␣Frequency␣sweep␣substructure
TYPE(RECEIVERtype)␣␣::␣receiver␣␣␣␣␣␣!␣Receiver␣settings␣substructure
TYPE(EXCITERtype)␣␣␣::␣exciter␣␣␣␣␣␣␣!␣Exciter␣settings␣substructure
TYPE(MONITORtype)␣␣␣::␣monitor␣␣␣␣␣␣␣!␣Built␣In␣Test␣values␣substructure
END␣TYPE␣SCTtype
```

B Pulse Control Table

The Pulse Control Table is a small structure containing values that are unique to each transmitted pulse, such as it's time, frequency and sequence in the ionogram.

Listing 7: PCT Structure, C

```
typedef struct
{
   long int        record_id;           // pri counter
   double          pri_ut;              // nominal per PRI duration
   double          pri_time_offset;     // read from clock, not accurate short term
   long int        base_id;             // base frequency number this PRI
   long int        pulse_id;            // pulse set element number this PRI
   long int        step_id;             // ramp count this PRI
   long int        repeat_id;           // ramp repeat count this PRI
   long int        loop_id;             // outer loop repeat this PRI
   float           frequency;           // frequency this PRI, kHz
   long int        nco_tune_word;       // NCO tune word applied this PRI
   float           drive_attenuation;   // drive attenuation applied this PRI
   long int        pa_flags;            // power amplifier status
   float           pa_forward_power;    // power amplifier measured forward power
   float           pa_reflected_power;  // power amplifier measured reverse power
   float           pa_vswr;             // power amplifier measured vswr
   float           pa_temperature;      // power amplifier measured temperature
   long int        proc_range_count;    // number of range gates kept this PRI [processed]
   float           proc_noise_level;    // estimated noise level for this PRI [processed]
   char            user[64];            // user spare
 } PCT;
```

Listing 8: PCT Structure, FORTRAN

```
!
! Define the Pulse Configuration Table (PCT) structure for Version 1.2 of the
! Raw Inphase & Quadrature (RIQ) file format for the Scion HF Radar (VIPIR)
!
!
! T. Bullett    04 November 2008   g95
! Adapted from:
! R. Livingston           gcc
!
!
! Pulse Configuration Table   Version 1.2
TYPE :: PCTtype
INTEGER(KIND=4) :: record_id            ! Sequence number of this PCT
REAL(KIND=8)    :: pri_ut               ! UT of this pulse
REAL(KIND=8)    :: pri_time_offset      ! Time read from system clock, not precise.
INTEGER(KIND=4) :: base_id              ! Base Frequency counter
INTEGER(KIND=4) :: pulse_id             ! pulse set element for this PRI
INTEGER(KIND=4) :: ramp_id              ! ramp set element for this PRI
INTEGER(KIND=4) :: repeat_id            ! ramp repeat element for this PRI
INTEGER(KIND=4) :: loop_id              ! Outer loop element for this PRI
REAL(KIND=4)    :: frequency            ! Frequency of observation (kHz)
INTEGER(KIND=4) :: nco_tune_word        ! Tuning word sent to the receiver
REAL(KIND=4)    :: drive_attenuation    ! Low-level drive attenuation [dB]
INTEGER(KIND=4) :: pa_flags             ! Status flags from amplifier
REAL(KIND=4)    :: pa_forward_power     ! Forward power from amplifier
REAL(KIND=4)    :: pa_reflected_power   ! Reflected power from amplifier
REAL(KIND=4)    :: pa_vswr              ! Voltage Standing Wave Ratio from amplifier
REAL(KIND=4)    :: pa_temperature       ! Amplifier temperature
INTEGER(KIND=4) :: proc_range_count     ! Number of range gates kept this PRI
REAL(KIND=4)    :: proc_noise_level     ! Estimated noise level for this PRI
CHARACTER(64)   :: user                 ! Spare space for user-defined information
END TYPE PCTtype
```

Listing 9: **dump_headers** output

```
#
# GENERAL:
# sct.magic:                     0x51495200
# sct.sounding_table_size:            90076
# sct.pulse_table_size:                 144
# sct.raw_data_size:                  16384
```

```
# sct.struct_version:                  1.20
# sct.start_year:                      2011
# sct.start_daynumber:                  146
# sct.start_month:                        5
# sct.start_day:                         26
# sct.start_hour:                        15
# sct.start_minute:                       7
# sct.start_second:                       4
# sct.start_epoch:               1306422424
# sct.readme:
# sct.user:
#
# STATION:
# sct.station.file_id          ../run/fastsweep/station_settings.txt
# sct.station.ursi_id                 WI937
# sct.station.rx_name          WallopsIsland
# sct.station.rx_latitude             37.94
# sct.station.rx_longitude           -75.48
# sct.station.rx_altitude             10.00
# sct.station.rx_count                    8
# sct.station.
# rx_antenna_type       rx_position X Y Z       rx_direction X Y Z   rx_height rx_cable_length
#         dipole    0.00   50.00    0.00    0.00    1.83    0.00        4.25          144.25
#         dipole   50.00    0.00    0.00    1.83    0.00    0.00        4.25          144.25
#         dipole    0.00  -50.00    0.00    0.00    1.83    0.00        4.25          144.32
#         dipole  -50.00    0.00    0.00    1.83    0.00    0.00        4.25          144.27
#         dipole    0.00    8.25    0.00    0.00    1.83    0.00        4.25          144.30
#         dipole    8.25    0.00    0.00    1.83    0.00    0.00        4.25          144.26
#         dipole    0.00   -8.25    0.00    0.00    1.83    0.00        4.25          144.31
#         dipole   -8.25    0.00    0.00    1.83    0.00    0.00        4.25          144.33
# sct.station.frontend_atten          10.00
# sct.station.tx_name          WallopsIsland
# sct.station.tx_latitude             37.94
# sct.station.tx_longitude           -75.48
# sct.station.tx_altitude             10.00
# sct.station.tx_vector
#         LPA120  250.00   20.00   10.00
# sct.station.tx_height                1.00
# sct.station.tx_cable_length         77.00
# sct.station.drive_band_count            1
# sct.station.
#   drive_band_bounds   drive_band_bounds    drive_band_atten
#         1000.000000        15000.000000            0.000000
# sct.station.rf_control                  0
# sct.station.ref_type         OCXO
# sct.station.clock_type       NTP
# sct.station.user
#
# TIMING:
# sct.timing.file_id           ../run/fastsweep/time_settings.txt
# sct.timing.pri                   10000.00
# sct.timing.pri_count                 3272
# sct.timing.ionogram_count               1
# sct.timing.holdoff                  11.00
# sct.timing.range_gate_offset        41.00
# sct.timing.gate_count                 512
# sct.timing.gate_start               88.00
# sct.timing.gate_end               5208.00
# sct.timing.gate_step                10.00
# sct.timing.data_start                0.00
# sct.timing.data_width               60.00
# sct.timing.data_baud_count              1
# sct.timing.data_wave_file    ../run/waveform_table.txt
# sct.timing.data_baud
#    1 (  1.00,  1.00 )
# sct.timing.data_pairs                  30
# sct.timing.cal_start               120.00
# sct.timing.cal_width                60.00
# sct.timing.cal_baud_count               1
# sct.timing.cal_wave_file     ../run/waveform_table.txt
# sct.timing.cal_baud
#    1 (  1.00,  1.00 )
# sct.timing.cal_pairs                   30
# sct.timing.user
#
# FREQUENCY:
# sct.frequency.file_id        ../run/fastsweep/freq_settings.txt
# sct.frequency.base_start          1000.00
# sct.frequency.base_end           26000.00
# sct.frequency.base_steps              818
# sct.frequency.tune_type                 1
# sct.frequency.base_table
#    1    1000.000000
#    2    1004.000000
#    3    1008.015991
#    4    1012.047974
```

```
#      5     1016.096008
#      6     1020.159973
#      7     1024.240967
#      8     1028.338013
#      9     1032.451050
#     10     1036.581055
#     11     1040.727051
#     12     1044.890015
#     13     1049.069946
#     14     1053.265991
#     15     1057.479004
#     16     1061.708984
#     17     1065.956055
#     18     1070.219971
#     19     1074.500977
#     20     1078.798950
#     21     1083.114014
#     22     1087.446045
#     23     1091.796021
#     24     1096.162964
#     25     1100.547974
#     26     1104.949951
#     27     1109.369995
#     28     1113.807007
#     29     1118.262939
#     30     1122.735962
#     31     1127.227051
#     32     1131.734985
#     33     1136.261963
#     34     1140.807007
#     35     1145.370972
#     36     1149.952026
#     37     1154.552002
#     38     1159.170044
#     39     1163.807007
#     40     1168.462036
#     41     1173.135986
#     42     1177.828003
#     43     1182.540039
#     44     1187.270020
#     45     1192.019043
#     46     1196.786987
#     47     1201.573975
#     48     1206.380981
#     49     1211.206055
#     50     1216.051025
#     51     1220.915039
#     52     1225.798950
#     53     1230.702026
#     54     1235.625000
#     55     1240.567017
#     56     1245.530029
#     57     1250.511963
#     58     1255.514038
#     59     1260.536011
#     60     1265.578003
#     61     1270.640015
#     62     1275.723022
#     63     1280.826050
#     64     1285.948975
#     65     1291.093018
#     66     1296.256958
#     67     1301.442017
#     68     1306.647949
#     69     1311.875000
#     70     1317.121948
#     71     1322.390991
#     72     1327.680054
#     73     1332.990967
#     74     1338.322998
#     75     1343.676025
#     76     1349.051025
#     77     1354.447021
#     78     1359.864990
#     79     1365.303955
#     80     1370.765015
#     81     1376.249023
#     82     1381.754028
#     83     1387.281006
#     84     1392.829956
#     85     1398.401001
#     86     1403.994995
#     87     1409.610962
#     88     1415.249023
#     89     1420.910034
#     90     1426.593994
```

```
#    91    1432.300049
#    92    1438.029053
#    93    1443.781006
#    94    1449.556030
#    95    1455.354980
#    96    1461.176025
#    97    1467.020996
#    98    1472.889038
#    99    1478.780029
#   100    1484.696045
#   101    1490.634033
#   102    1496.597046
#   103    1502.583008
#   104    1508.593994
#   105    1514.628052
#   106    1520.687012
#   107    1526.769043
#   108    1532.875977
#   109    1539.008057
#   110    1545.163940
#   111    1551.344971
#   112    1557.550049
#   113    1563.780029
#   114    1570.035034
#   115    1576.314941
#   116    1582.620972
#   117    1588.951050
#   118    1595.307007
#   119    1601.687988
#   120    1608.094971
#   121    1614.526978
#   122    1620.984985
#   123    1627.468994
#   124    1633.979004
#   125    1640.515015
#   126    1647.077026
#   127    1653.666016
#   128    1660.280029
#   129    1666.921021
#   130    1673.588989
#   131    1680.282959
#   132    1687.005005
#   133    1693.753052
#   134    1700.527954
#   135    1707.329956
#   136    1714.159058
#   137    1721.015991
#   138    1727.900024
#   139    1734.811035
#   140    1741.750977
#   141    1748.718018
#   142    1755.712036
#   143    1762.734985
#   144    1769.786011
#   145    1776.864990
#   146    1783.973022
#   147    1791.109009
#   148    1798.272949
#   149    1805.465942
#   150    1812.687988
#   151    1819.938965
#   152    1827.218994
#   153    1834.527954
#   154    1841.865967
#   155    1849.233032
#   156    1856.630005
#   157    1864.057007
#   158    1871.512939
#   159    1878.999023
#   160    1886.515015
#   161    1894.061035
#   162    1901.636963
#   163    1909.244019
#   164    1916.880981
#   165    1924.547974
#   166    1932.245972
#   167    1939.974976
#   168    1947.734985
#   169    1955.526001
#   170    1963.348022
#   171    1971.202026
#   172    1979.087036
#   173    1987.003052
#   174    1994.951050
#   175    2002.931030
#   176    2010.942017
```

```
#   177   2018.985962
#   178   2027.062012
#   179   2035.170044
#   180   2043.311035
#   181   2051.483887
#   182   2059.689941
#   183   2067.928955
#   184   2076.200928
#   185   2084.506104
#   186   2092.843994
#   187   2101.215088
#   188   2109.620117
#   189   2118.058105
#   190   2126.531006
#   191   2135.037109
#   192   2143.576904
#   193   2152.150879
#   194   2160.760010
#   195   2169.403076
#   196   2178.080078
#   197   2186.792969
#   198   2195.540039
#   199   2204.322021
#   200   2213.138916
#   201   2221.991943
#   202   2230.879883
#   203   2239.802979
#   204   2248.762939
#   205   2257.758057
#   206   2266.789062
#   207   2275.855957
#   208   2284.958984
#   209   2294.099121
#   210   2303.275879
#   211   2312.489014
#   212   2321.739014
#   213   2331.025879
#   214   2340.350098
#   215   2349.710938
#   216   2359.110107
#   217   2368.545898
#   218   2378.020020
#   219   2387.532959
#   220   2397.083008
#   221   2406.670898
#   222   2416.298096
#   223   2425.962891
#   224   2435.666992
#   225   2445.408936
#   226   2455.190918
#   227   2465.011963
#   228   2474.872070
#   229   2484.770996
#   230   2494.709961
#   231   2504.688965
#   232   2514.708008
#   233   2524.767090
#   234   2534.865967
#   235   2545.004883
#   236   2555.185059
#   237   2565.406006
#   238   2575.667969
#   239   2585.970947
#   240   2596.313965
#   241   2606.699951
#   242   2617.125977
#   243   2627.594971
#   244   2638.104980
#   245   2648.657959
#   246   2659.251953
#   247   2669.888916
#   248   2680.569092
#   249   2691.291016
#   250   2702.055908
#   251   2712.864990
#   252   2723.716064
#   253   2734.611084
#   254   2745.549072
#   255   2756.531982
#   256   2767.558105
#   257   2778.627930
#   258   2789.742920
#   259   2800.900879
#   260   2812.104980
#   261   2823.354004
#   262   2834.646973
```

```
#   263   2845.986084
#   264   2857.368896
#   265   2868.799072
#   266   2880.273926
#   267   2891.794922
#   268   2903.362061
#   269   2914.976074
#   270   2926.635986
#   271   2938.342041
#   272   2950.095947
#   273   2961.895996
#   274   2973.743896
#   275   2985.638916
#   276   2997.581055
#   277   3009.572021
#   278   3021.610107
#   279   3033.696045
#   280   3045.831055
#   281   3058.013916
#   282   3070.246094
#   283   3082.527100
#   284   3094.857910
#   285   3107.237061
#   286   3119.666016
#   287   3132.145020
#   288   3144.673096
#   289   3157.251953
#   290   3169.881104
#   291   3182.560059
#   292   3195.291016
#   293   3208.072021
#   294   3220.904053
#   295   3233.788086
#   296   3246.722900
#   297   3259.709961
#   298   3272.749023
#   299   3285.840088
#   300   3298.982910
#   301   3312.178955
#   302   3325.427979
#   303   3338.729004
#   304   3352.083984
#   305   3365.492920
#   306   3378.955078
#   307   3392.469971
#   308   3406.040039
#   309   3419.664062
#   310   3433.343018
#   311   3447.075928
#   312   3460.864990
#   313   3474.708008
#   314   3488.606934
#   315   3502.561035
#   316   3516.572021
#   317   3530.637939
#   318   3544.760986
#   319   3558.939941
#   320   3573.175049
#   321   3587.468018
#   322   3601.818115
#   323   3616.225098
#   324   3630.689941
#   325   3645.212891
#   326   3659.793945
#   327   3674.433105
#   328   3689.131104
#   329   3703.886963
#   330   3718.702881
#   331   3733.577881
#   332   3748.511963
#   333   3763.506104
#   334   3778.560059
#   335   3793.674072
#   336   3808.849121
#   337   3824.083984
#   338   3839.381104
#   339   3854.738037
#   340   3870.156982
#   341   3885.637939
#   342   3901.179932
#   343   3916.784912
#   344   3932.451904
#   345   3948.181885
#   346   3963.975098
#   347   3979.831055
#   348   3995.750000
```

```
#   349   4011.732910
#   350   4027.780029
#   351   4043.891113
#   352   4060.066895
#   353   4076.306885
#   354   4092.612061
#   355   4108.981934
#   356   4125.417969
#   357   4141.919922
#   358   4158.487793
#   359   4175.122070
#   360   4191.821777
#   361   4208.589844
#   362   4225.423828
#   363   4242.326172
#   364   4259.294922
#   365   4276.332031
#   366   4293.437012
#   367   4310.610840
#   368   4327.854004
#   369   4345.165039
#   370   4362.545898
#   371   4379.996094
#   372   4397.516113
#   373   4415.105957
#   374   4432.766113
#   375   4450.497070
#   376   4468.298828
#   377   4486.172852
#   378   4504.117188
#   379   4522.133789
#   380   4540.222168
#   381   4558.382812
#   382   4576.617188
#   383   4594.922852
#   384   4613.303223
#   385   4631.755859
#   386   4650.283203
#   387   4668.883789
#   388   4687.560059
#   389   4706.310059
#   390   4725.134766
#   391   4744.036133
#   392   4763.012207
#   393   4782.063965
#   394   4801.191895
#   395   4820.396973
#   396   4839.679199
#   397   4859.037109
#   398   4878.474121
#   399   4897.986816
#   400   4917.579102
#   401   4937.250000
#   402   4956.999023
#   403   4976.827148
#   404   4996.733887
#   405   5016.721191
#   406   5036.788086
#   407   5056.935059
#   408   5077.163086
#   409   5097.471191
#   410   5117.860840
#   411   5138.333008
#   412   5158.886230
#   413   5179.521973
#   414   5200.240234
#   415   5221.041016
#   416   5241.924805
#   417   5262.892090
#   418   5283.943848
#   419   5305.080078
#   420   5326.299805
#   421   5347.604980
#   422   5368.996094
#   423   5390.472168
#   424   5412.034180
#   425   5433.682129
#   426   5455.416992
#   427   5477.237793
#   428   5499.146973
#   429   5521.144043
#   430   5543.228027
#   431   5565.400879
#   432   5587.663086
#   433   5610.013184
#   434   5632.454102
```

```
#   435   5654.982910
#   436   5677.603027
#   437   5700.313965
#   438   5723.115234
#   439   5746.006836
#   440   5768.991211
#   441   5792.066895
#   442   5815.235840
#   443   5838.497070
#   444   5861.851074
#   445   5885.297852
#   446   5908.838867
#   447   5932.475098
#   448   5956.205078
#   449   5980.028809
#   450   6003.949219
#   451   6027.964844
#   452   6052.077148
#   453   6076.285156
#   454   6100.590820
#   455   6124.993164
#   456   6149.493164
#   457   6174.090820
#   458   6198.787109
#   459   6223.582031
#   460   6248.477051
#   461   6273.471191
#   462   6298.564941
#   463   6323.758789
#   464   6349.054199
#   465   6374.450195
#   466   6399.948242
#   467   6425.547852
#   468   6451.250000
#   469   6477.055176
#   470   6502.962891
#   471   6528.975098
#   472   6555.090820
#   473   6581.311035
#   474   6607.636230
#   475   6634.066895
#   476   6660.603027
#   477   6687.246094
#   478   6713.995117
#   479   6740.851074
#   480   6767.813965
#   481   6794.884766
#   482   6822.064941
#   483   6849.353027
#   484   6876.750000
#   485   6904.257812
#   486   6931.875000
#   487   6959.602051
#   488   6987.439941
#   489   7015.390137
#   490   7043.452148
#   491   7071.625977
#   492   7099.912109
#   493   7128.312012
#   494   7156.825195
#   495   7185.452148
#   496   7214.193848
#   497   7243.050781
#   498   7272.022949
#   499   7301.110840
#   500   7330.315918
#   501   7359.637207
#   502   7389.075195
#   503   7418.631836
#   504   7448.306152
#   505   7478.100098
#   506   7508.012207
#   507   7538.043945
#   508   7568.195801
#   509   7598.469238
#   510   7628.862793
#   511   7659.377930
#   512   7690.016113
#   513   7720.775879
#   514   7751.659180
#   515   7782.666016
#   516   7813.795898
#   517   7845.050781
#   518   7876.432129
#   519   7907.937012
#   520   7939.568848
```

#	521	7971.327148
#	522	8003.212891
#	523	8035.226074
#	524	8067.366211
#	525	8099.636230
#	526	8132.034180
#	527	8164.562988
#	528	8197.220703
#	529	8230.009766
#	530	8262.929688
#	531	8295.982422
#	532	8329.165039
#	533	8362.482422
#	534	8395.931641
#	535	8429.515625
#	536	8463.234375
#	537	8497.086914
#	538	8531.075195
#	539	8565.199219
#	540	8599.459961
#	541	8633.858398
#	542	8668.393555
#	543	8703.067383
#	544	8737.878906
#	545	8772.831055
#	546	8807.921875
#	547	8843.154297
#	548	8878.527344
#	549	8914.041016
#	550	8949.697266
#	551	8985.496094
#	552	9021.438477
#	553	9057.523438
#	554	9093.752930
#	555	9130.127930
#	556	9166.649414
#	557	9203.316406
#	558	9240.128906
#	559	9277.088867
#	560	9314.198242
#	561	9351.454102
#	562	9388.860352
#	563	9426.416016
#	564	9464.121094
#	565	9501.977539
#	566	9539.986328
#	567	9578.146484
#	568	9616.458008
#	569	9654.923828
#	570	9693.543945
#	571	9732.318359
#	572	9771.247070
#	573	9810.332031
#	574	9849.574219
#	575	9888.971680
#	576	9928.528320
#	577	9968.242188
#	578	10008.115234
#	579	10048.147461
#	580	10088.339844
#	581	10128.693359
#	582	10169.208008
#	583	10209.884766
#	584	10250.723633
#	585	10291.726562
#	586	10332.893555
#	587	10374.225586
#	588	10415.722656
#	589	10457.385742
#	590	10499.214844
#	591	10541.211914
#	592	10583.376953
#	593	10625.709961
#	594	10668.212891
#	595	10710.885742
#	596	10753.730469
#	597	10796.745117
#	598	10839.931641
#	599	10883.291016
#	600	10926.825195
#	601	10970.532227
#	602	11014.414062
#	603	11058.471680
#	604	11102.705078
#	605	11147.116211
#	606	11191.705078

```
#    607    11236.471680
#    608    11281.416992
#    609    11326.542969
#    610    11371.848633
#    611    11417.336914
#    612    11463.005859
#    613    11508.858398
#    614    11554.893555
#    615    11601.113281
#    616    11647.517578
#    617    11694.108398
#    618    11740.883789
#    619    11787.847656
#    620    11834.999023
#    621    11882.338867
#    622    11929.868164
#    623    11977.587891
#    624    12025.498047
#    625    12073.599609
#    626    12121.894531
#    627    12170.381836
#    628    12219.064453
#    629    12267.940430
#    630    12317.011719
#    631    12366.280273
#    632    12415.745117
#    633    12465.408203
#    634    12515.269531
#    635    12565.331055
#    636    12615.591797
#    637    12666.053711
#    638    12716.718750
#    639    12767.584961
#    640    12818.656250
#    641    12869.929688
#    642    12921.410156
#    643    12973.095703
#    644    13024.988281
#    645    13077.087891
#    646    13129.396484
#    647    13181.914062
#    648    13234.641602
#    649    13287.580078
#    650    13340.731445
#    651    13394.093750
#    652    13447.669922
#    653    13501.460938
#    654    13555.465820
#    655    13609.688477
#    656    13664.126953
#    657    13718.784180
#    658    13773.659180
#    659    13828.752930
#    660    13884.068359
#    661    13939.605469
#    662    13995.363281
#    663    14051.343750
#    664    14107.549805
#    665    14163.980469
#    666    14220.635742
#    667    14277.518555
#    668    14334.628906
#    669    14391.966797
#    670    14449.535156
#    671    14507.333008
#    672    14565.362305
#    673    14623.624023
#    674    14682.118164
#    675    14740.846680
#    676    14799.809570
#    677    14859.009766
#    678    14918.446289
#    679    14978.119141
#    680    15038.032227
#    681    15098.183594
#    682    15158.577148
#    683    15219.210938
#    684    15280.087891
#    685    15341.208008
#    686    15402.573242
#    687    15464.182617
#    688    15526.040039
#    689    15588.143555
#    690    15650.497070
#    691    15713.098633
#    692    15775.951172
```

```
#   693   15839.054688
#   694   15902.411133
#   695   15966.021484
#   696   16029.884766
#   697   16094.004883
#   698   16158.380859
#   699   16223.013672
#   700   16287.906250
#   701   16353.057617
#   702   16418.470703
#   703   16484.144531
#   704   16550.080078
#   705   16616.281250
#   706   16682.746094
#   707   16749.476562
#   708   16816.474609
#   709   16883.740234
#   710   16951.275391
#   711   17019.080078
#   712   17087.156250
#   713   17155.505859
#   714   17224.128906
#   715   17293.023438
#   716   17362.195312
#   717   17431.644531
#   718   17501.371094
#   719   17571.376953
#   720   17641.662109
#   721   17712.228516
#   722   17783.078125
#   723   17854.210938
#   724   17925.628906
#   725   17997.330078
#   726   18069.318359
#   727   18141.597656
#   728   18214.162109
#   729   18287.019531
#   730   18360.167969
#   731   18433.607422
#   732   18507.343750
#   733   18581.371094
#   734   18655.697266
#   735   18730.320312
#   736   18805.242188
#   737   18880.462891
#   738   18955.984375
#   739   19031.808594
#   740   19107.935547
#   741   19184.367188
#   742   19261.105469
#   743   19338.148438
#   744   19415.501953
#   745   19493.164062
#   746   19571.136719
#   747   19649.421875
#   748   19728.019531
#   749   19806.931641
#   750   19886.158203
#   751   19965.703125
#   752   20045.566406
#   753   20125.748047
#   754   20206.251953
#   755   20287.076172
#   756   20368.224609
#   757   20449.697266
#   758   20531.496094
#   759   20613.621094
#   760   20696.076172
#   761   20778.861328
#   762   20861.976562
#   763   20945.425781
#   764   21029.205078
#   765   21113.322266
#   766   21197.775391
#   767   21282.568359
#   768   21367.697266
#   769   21453.169922
#   770   21538.980469
#   771   21625.136719
#   772   21711.638672
#   773   21798.484375
#   774   21885.677734
#   775   21973.220703
#   776   22061.113281
#   777   22149.357422
#   778   22237.955078
```

```
#   779   22326.908203
#   780   22416.214844
#   781   22505.880859
#   782   22595.904297
#   783   22686.287109
#   784   22777.031250
#   785   22868.140625
#   786   22959.613281
#   787   23051.451172
#   788   23143.656250
#   789   23236.232422
#   790   23329.177734
#   791   23422.494141
#   792   23516.183594
#   793   23610.248047
#   794   23704.689453
#   795   23799.507812
#   796   23894.705078
#   797   23990.285156
#   798   24086.246094
#   799   24182.591797
#   800   24279.322266
#   801   24376.439453
#   802   24473.945312
#   803   24571.839844
#   804   24670.128906
#   805   24768.808594
#   806   24867.882812
#   807   24967.355469
#   808   25067.224609
#   809   25167.492188
#   810   25268.162109
#   811   25369.236328
#   812   25470.712891
#   813   25572.595703
#   814   25674.886719
#   815   25777.585938
#   816   25880.695312
#   817   25984.218750
#   818   26088.156250
# sct.frequency.linear_step          10.00
# sct.frequency.log_step              0.40
# sct.frequency.freq_table_id ../run/fastsweep/frequency_table.txt
# sct.frequency.tune_steps            3272
# sct.frequency.pulse_count              4
# sct.frequency.pulse_pattern
#     1     0
#     2     0
#     3     0
#     4     0
# sct.frequency.pulse_offset          0.00
# sct.frequency.ramp_steps               1
# sct.frequency.ramp_repeats             1
# sct.frequency.drive_table
#     1   0.0
#     2   0.0
#     3   0.0
#     4   0.0
#     5   0.0
#     6   0.0
#     7   0.0
#     8   0.0
#     9   0.0
#    10   0.0
#    11   0.0
#    12   0.0
#    13   0.0
#    14   0.0
#    15   0.0
#    16   0.0
#    17   0.0
#    18   0.0
#    19   0.0
#    20   0.0
#    21   0.0
#    22   0.0
#    23   0.0
#    24   0.0
#    25   0.0
#    26   0.0
#    27   0.0
#    28   0.0
#    29   0.0
#    30   0.0
#    31   0.0
#    32   0.0
```

```
#    33   0.0
#    34   0.0
#    35   0.0
#    36   0.0
#    37   0.0
#    38   0.0
#    39   0.0
#    40   0.0
#    41   0.0
#    42   0.0
#    43   0.0
#    44   0.0
#    45   0.0
#    46   0.0
#    47   0.0
#    48   0.0
#    49   0.0
#    50   0.0
#    51   0.0
#    52   0.0
#    53   0.0
#    54   0.0
#    55   0.0
#    56   0.0
#    57   0.0
#    58   0.0
#    59   0.0
#    60   0.0
#    61   0.0
#    62   0.0
#    63   0.0
#    64   0.0
#    65   0.0
#    66   0.0
#    67   0.0
#    68   0.0
#    69   0.0
#    70   0.0
#    71   0.0
#    72   0.0
#    73   0.0
#    74   0.0
#    75   0.0
#    76   0.0
#    77   0.0
#    78   0.0
#    79   0.0
#    80   0.0
#    81   0.0
#    82   0.0
#    83   0.0
#    84   0.0
#    85   0.0
#    86   0.0
#    87   0.0
#    88   0.0
#    89   0.0
#    90   0.0
#    91   0.0
#    92   0.0
#    93   0.0
#    94   0.0
#    95   0.0
#    96   0.0
#    97   0.0
#    98   0.0
#    99   0.0
#   100   0.0
#   101   0.0
#   102   0.0
#   103   0.0
#   104   0.0
#   105   0.0
#   106   0.0
#   107   0.0
#   108   0.0
#   109   0.0
#   110   0.0
#   111   0.0
#   112   0.0
#   113   0.0
#   114   0.0
#   115   0.0
#   116   0.0
#   117   0.0
#   118   0.0
```

```
#   119   0.0
#   120   0.0
#   121   0.0
#   122   0.0
#   123   0.0
#   124   0.0
#   125   0.0
#   126   0.0
#   127   0.0
#   128   0.0
#   129   0.0
#   130   0.0
#   131   0.0
#   132   0.0
#   133   0.0
#   134   0.0
#   135   0.0
#   136   0.0
#   137   0.0
#   138   0.0
#   139   0.0
#   140   0.0
#   141   0.0
#   142   0.0
#   143   0.0
#   144   0.0
#   145   0.0
#   146   0.0
#   147   0.0
#   148   0.0
#   149   0.0
#   150   0.0
#   151   0.0
#   152   0.0
#   153   0.0
#   154   0.0
#   155   0.0
#   156   0.0
#   157   0.0
#   158   0.0
#   159   0.0
#   160   0.0
#   161   0.0
#   162   0.0
#   163   0.0
#   164   0.0
#   165   0.0
#   166   0.0
#   167   0.0
#   168   0.0
#   169   0.0
#   170   0.0
#   171   0.0
#   172   0.0
#   173   0.0
#   174   0.0
#   175   0.0
#   176   0.0
#   177   0.0
#   178   0.0
#   179   0.0
#   180   0.0
#   181   0.0
#   182   0.0
#   183   0.0
#   184   0.0
#   185   0.0
#   186   0.0
#   187   0.0
#   188   0.0
#   189   0.0
#   190   0.0
#   191   0.0
#   192   0.0
#   193   0.0
#   194   0.0
#   195   0.0
#   196   0.0
#   197   0.0
#   198   0.0
#   199   0.0
#   200   0.0
#   201   0.0
#   202   0.0
#   203   0.0
#   204   0.0
```

```
#   205   0.0
#   206   0.0
#   207   0.0
#   208   0.0
#   209   0.0
#   210   0.0
#   211   0.0
#   212   0.0
#   213   0.0
#   214   0.0
#   215   0.0
#   216   0.0
#   217   0.0
#   218   0.0
#   219   0.0
#   220   0.0
#   221   0.0
#   222   0.0
#   223   0.0
#   224   0.0
#   225   0.0
#   226   0.0
#   227   0.0
#   228   0.0
#   229   0.0
#   230   0.0
#   231   0.0
#   232   0.0
#   233   0.0
#   234   0.0
#   235   0.0
#   236   0.0
#   237   0.0
#   238   0.0
#   239   0.0
#   240   0.0
#   241   0.0
#   242   0.0
#   243   0.0
#   244   0.0
#   245   0.0
#   246   0.0
#   247   0.0
#   248   0.0
#   249   0.0
#   250   0.0
#   251   0.0
#   252   0.0
#   253   0.0
#   254   0.0
#   255   0.0
#   256   0.0
#   257   0.0
#   258   0.0
#   259   0.0
#   260   0.0
#   261   0.0
#   262   0.0
#   263   0.0
#   264   0.0
#   265   0.0
#   266   0.0
#   267   0.0
#   268   0.0
#   269   0.0
#   270   0.0
#   271   0.0
#   272   0.0
#   273   0.0
#   274   0.0
#   275   0.0
#   276   0.0
#   277   0.0
#   278   0.0
#   279   0.0
#   280   0.0
#   281   0.0
#   282   0.0
#   283   0.0
#   284   0.0
#   285   0.0
#   286   0.0
#   287   0.0
#   288   0.0
#   289   0.0
#   290   0.0
```

```
#   291   0.0
#   292   0.0
#   293   0.0
#   294   0.0
#   295   0.0
#   296   0.0
#   297   0.0
#   298   0.0
#   299   0.0
#   300   0.0
#   301   0.0
#   302   0.0
#   303   0.0
#   304   0.0
#   305   0.0
#   306   0.0
#   307   0.0
#   308   0.0
#   309   0.0
#   310   0.0
#   311   0.0
#   312   0.0
#   313   0.0
#   314   0.0
#   315   0.0
#   316   0.0
#   317   0.0
#   318   0.0
#   319   0.0
#   320   0.0
#   321   0.0
#   322   0.0
#   323   0.0
#   324   0.0
#   325   0.0
#   326   0.0
#   327   0.0
#   328   0.0
#   329   0.0
#   330   0.0
#   331   0.0
#   332   0.0
#   333   0.0
#   334   0.0
#   335   0.0
#   336   0.0
#   337   0.0
#   338   0.0
#   339   0.0
#   340   0.0
#   341   0.0
#   342   0.0
#   343   0.0
#   344   0.0
#   345   0.0
#   346   0.0
#   347   0.0
#   348   0.0
#   349   0.0
#   350   0.0
#   351   0.0
#   352   0.0
#   353   0.0
#   354   0.0
#   355   0.0
#   356   0.0
#   357   0.0
#   358   0.0
#   359   0.0
#   360   0.0
#   361   0.0
#   362   0.0
#   363   0.0
#   364   0.0
#   365   0.0
#   366   0.0
#   367   0.0
#   368   0.0
#   369   0.0
#   370   0.0
#   371   0.0
#   372   0.0
#   373   0.0
#   374   0.0
#   375   0.0
#   376   0.0
```

```
#    377    0.0
#    378    0.0
#    379    0.0
#    380    0.0
#    381    0.0
#    382    0.0
#    383    0.0
#    384    0.0
#    385    0.0
#    386    0.0
#    387    0.0
#    388    0.0
#    389    0.0
#    390    0.0
#    391    0.0
#    392    0.0
#    393    0.0
#    394    0.0
#    395    0.0
#    396    0.0
#    397    0.0
#    398    0.0
#    399    0.0
#    400    0.0
#    401    0.0
#    402    0.0
#    403    0.0
#    404    0.0
#    405    0.0
#    406    0.0
#    407    0.0
#    408    0.0
#    409    0.0
#    410    0.0
#    411    0.0
#    412    0.0
#    413    0.0
#    414    0.0
#    415    0.0
#    416    0.0
#    417    0.0
#    418    0.0
#    419    0.0
#    420    0.0
#    421    0.0
#    422    0.0
#    423    0.0
#    424    0.0
#    425    0.0
#    426    0.0
#    427    0.0
#    428    0.0
#    429    0.0
#    430    0.0
#    431    0.0
#    432    0.0
#    433    0.0
#    434    0.0
#    435    0.0
#    436    0.0
#    437    0.0
#    438    0.0
#    439    0.0
#    440    0.0
#    441    0.0
#    442    0.0
#    443    0.0
#    444    0.0
#    445    0.0
#    446    0.0
#    447    0.0
#    448    0.0
#    449    0.0
#    450    0.0
#    451    0.0
#    452    0.0
#    453    0.0
#    454    0.0
#    455    0.0
#    456    0.0
#    457    0.0
#    458    0.0
#    459    0.0
#    460    0.0
#    461    0.0
#    462    0.0
```

```
#   463   0.0
#   464   0.0
#   465   0.0
#   466   0.0
#   467   0.0
#   468   0.0
#   469   0.0
#   470   0.0
#   471   0.0
#   472   0.0
#   473   0.0
#   474   0.0
#   475   0.0
#   476   0.0
#   477   0.0
#   478   0.0
#   479   0.0
#   480   0.0
#   481   0.0
#   482   0.0
#   483   0.0
#   484   0.0
#   485   0.0
#   486   0.0
#   487   0.0
#   488   0.0
#   489   0.0
#   490   0.0
#   491   0.0
#   492   0.0
#   493   0.0
#   494   0.0
#   495   0.0
#   496   0.0
#   497   0.0
#   498   0.0
#   499   0.0
#   500   0.0
#   501   0.0
#   502   0.0
#   503   0.0
#   504   0.0
#   505   0.0
#   506   0.0
#   507   0.0
#   508   0.0
#   509   0.0
#   510   0.0
#   511   0.0
#   512   0.0
#   513   0.0
#   514   0.0
#   515   0.0
#   516   0.0
#   517   0.0
#   518   0.0
#   519   0.0
#   520   0.0
#   521   0.0
#   522   0.0
#   523   0.0
#   524   0.0
#   525   0.0
#   526   0.0
#   527   0.0
#   528   0.0
#   529   0.0
#   530   0.0
#   531   0.0
#   532   0.0
#   533   0.0
#   534   0.0
#   535   0.0
#   536   0.0
#   537   0.0
#   538   0.0
#   539   0.0
#   540   0.0
#   541   0.0
#   542   0.0
#   543   0.0
#   544   0.0
#   545   0.0
#   546   0.0
#   547   0.0
#   548   0.0
```

```
#   549   0.0
#   550   0.0
#   551   0.0
#   552   0.0
#   553   0.0
#   554   0.0
#   555   0.0
#   556   0.0
#   557   0.0
#   558   0.0
#   559   0.0
#   560   0.0
#   561   0.0
#   562   0.0
#   563   0.0
#   564   0.0
#   565   0.0
#   566   0.0
#   567   0.0
#   568   0.0
#   569   0.0
#   570   0.0
#   571   0.0
#   572   0.0
#   573   0.0
#   574   0.0
#   575   0.0
#   576   0.0
#   577   0.0
#   578   0.0
#   579   0.0
#   580   0.0
#   581   0.0
#   582   0.0
#   583   0.0
#   584   0.0
#   585   0.0
#   586   0.0
#   587   0.0
#   588   0.0
#   589   0.0
#   590   0.0
#   591   0.0
#   592   0.0
#   593   0.0
#   594   0.0
#   595   0.0
#   596   0.0
#   597   0.0
#   598   0.0
#   599   0.0
#   600   0.0
#   601   0.0
#   602   0.0
#   603   0.0
#   604   0.0
#   605   0.0
#   606   0.0
#   607   0.0
#   608   0.0
#   609   0.0
#   610   0.0
#   611   0.0
#   612   0.0
#   613   0.0
#   614   0.0
#   615   0.0
#   616   0.0
#   617   0.0
#   618   0.0
#   619   0.0
#   620   0.0
#   621   0.0
#   622   0.0
#   623   0.0
#   624   0.0
#   625   0.0
#   626   0.0
#   627   0.0
#   628   0.0
#   629   0.0
#   630   0.0
#   631   0.0
#   632   0.0
#   633   0.0
#   634   0.0
```

```
#   635   0.0
#   636   0.0
#   637   0.0
#   638   0.0
#   639   0.0
#   640   0.0
#   641   0.0
#   642   0.0
#   643   0.0
#   644   0.0
#   645   0.0
#   646   0.0
#   647   0.0
#   648   0.0
#   649   0.0
#   650   0.0
#   651   0.0
#   652   0.0
#   653   0.0
#   654   0.0
#   655   0.0
#   656   0.0
#   657   0.0
#   658   0.0
#   659   0.0
#   660   0.0
#   661   0.0
#   662   0.0
#   663   0.0
#   664   0.0
#   665   0.0
#   666   0.0
#   667   0.0
#   668   0.0
#   669   0.0
#   670   0.0
#   671   0.0
#   672   0.0
#   673   0.0
#   674   0.0
#   675   0.0
#   676   0.0
#   677   0.0
#   678   0.0
#   679   0.0
#   680   0.0
#   681   0.0
#   682   0.0
#   683   0.0
#   684   0.0
#   685   0.0
#   686   0.0
#   687   0.0
#   688   0.0
#   689   0.0
#   690   0.0
#   691   0.0
#   692   0.0
#   693   0.0
#   694   0.0
#   695   0.0
#   696   0.0
#   697   0.0
#   698   0.0
#   699   0.0
#   700   0.0
#   701   0.0
#   702   0.0
#   703   0.0
#   704   0.0
#   705   0.0
#   706   0.0
#   707   0.0
#   708   0.0
#   709   0.0
#   710   0.0
#   711   0.0
#   712   0.0
#   713   0.0
#   714   0.0
#   715   0.0
#   716   0.0
#   717   0.0
#   718   0.0
#   719   0.0
#   720   0.0
```

```
#   721   0.0
#   722   0.0
#   723   0.0
#   724   0.0
#   725   0.0
#   726   0.0
#   727   0.0
#   728   0.0
#   729   0.0
#   730   0.0
#   731   0.0
#   732   0.0
#   733   0.0
#   734   0.0
#   735   0.0
#   736   0.0
#   737   0.0
#   738   0.0
#   739   0.0
#   740   0.0
#   741   0.0
#   742   0.0
#   743   0.0
#   744   0.0
#   745   0.0
#   746   0.0
#   747   0.0
#   748   0.0
#   749   0.0
#   750   0.0
#   751   0.0
#   752   0.0
#   753   0.0
#   754   0.0
#   755   0.0
#   756   0.0
#   757   0.0
#   758   0.0
#   759   0.0
#   760   0.0
#   761   0.0
#   762   0.0
#   763   0.0
#   764   0.0
#   765   0.0
#   766   0.0
#   767   0.0
#   768   0.0
#   769   0.0
#   770   0.0
#   771   0.0
#   772   0.0
#   773   0.0
#   774   0.0
#   775   0.0
#   776   0.0
#   777   0.0
#   778   0.0
#   779   0.0
#   780   0.0
#   781   0.0
#   782   0.0
#   783   0.0
#   784   0.0
#   785   0.0
#   786   0.0
#   787   0.0
#   788   0.0
#   789   0.0
#   790   0.0
#   791   0.0
#   792   0.0
#   793   0.0
#   794   0.0
#   795   0.0
#   796   0.0
#   797   0.0
#   798   0.0
#   799   0.0
#   800   0.0
#   801   0.0
#   802   0.0
#   803   0.0
#   804   0.0
#   805   0.0
#   806   0.0
```

```
#   807   0.0
#   808   0.0
#   809   0.0
#   810   0.0
#   811   0.0
#   812   0.0
#   813   0.0
#   814   0.0
#   815   0.0
#   816   0.0
#   817   0.0
#   818   0.0
# sct.frequency.user
#
# RECEIVER:
# sct.receiver.file_id         ../run/fastsweep/ddc_settings.txt
# sct.receiver.rx_chan                    8
# sct.receiver.rx_map
#     1     0
#     2     2
#     3     3
#     4     4
#     5     5
#     6     6
#     7     7
#     8     8
# sct.receiver.word_format                1
# sct.receiver.cic2_dec                  25
# sct.receiver.cic2_interp                1
# sct.receiver.cic2_scale                10
# sct.receiver.cic5_dec                   2
# sct.receiver.cic5_scale                10
# sct.receiver.rcf_type        RG-COS2-0704
# sct.receiver.rcf_dec                   16
# sct.receiver.rcf_taps                 160
# sct.receiver.coefficients
#     1     0
#     2     0
#     3     0
#     4     0
#     5     0
#     6     0
#     7     0
#     8     0
#     9     0
#    10     0
#    11     0
#    12     0
#    13     0
#    14     0
#    15     0
#    16     0
#    17     1
#    18     1
#    19     2
#    20     3
#    21     5
#    22     8
#    23    11
#    24    16
#    25    22
#    26    30
#    27    41
#    28    53
#    29    70
#    30    90
#    31   114
#    32   143
#    33   179
#    34   221
#    35   271
#    36   329
#    37   397
#    38   476
#    39   565
#    40   668
#    41   783
#    42   913
#    43  1059
#    44  1220
#    45  1398
#    46  1594
#    47  1807
#    48  2039
#    49  2290
#    50  2559
```

```
#     51   2846
#     52   3152
#     53   3475
#     54   3815
#     55   4171
#     56   4542
#     57   4926
#     58   5321
#     59   5726
#     60   6139
#     61   6558
#     62   6979
#     63   7401
#     64   7821
#     65   8236
#     66   8644
#     67   9041
#     68   9424
#     69   9791
#     70  10139
#     71  10465
#     72  10766
#     73  11041
#     74  11287
#     75  11501
#     76  11683
#     77  11830
#     78  11941
#     79  12016
#     80  12054
#     81  12054
#     82  12016
#     83  11941
#     84  11830
#     85  11683
#     86  11501
#     87  11287
#     88  11041
#     89  10766
#     90  10465
#     91  10139
#     92   9791
#     93   9424
#     94   9041
#     95   8644
#     96   8236
#     97   7821
#     98   7401
#     99   6979
#    100   6558
#    101   6139
#    102   5726
#    103   5321
#    104   4926
#    105   4542
#    106   4171
#    107   3815
#    108   3475
#    109   3152
#    110   2846
#    111   2559
#    112   2290
#    113   2039
#    114   1807
#    115   1594
#    116   1398
#    117   1220
#    118   1059
#    119    913
#    120    783
#    121    668
#    122    565
#    123    476
#    124    397
#    125    329
#    126    271
#    127    221
#    128    179
#    129    143
#    130    114
#    131     90
#    132     70
#    133     53
#    134     41
#    135     30
#    136     22
```

```
#   137    16
#   138    11
#   139     8
#   140     5
#   141     3
#   142     2
#   143     1
#   144     1
#   145     0
#   146     0
#   147     0
#   148     0
#   149     0
#   150     0
#   151     0
#   152     0
#   153     0
#   154     0
#   155     0
#   156     0
#   157     0
#   158     0
#   159     0
#   160     0
# sct.receiver.user
#
# EXCITER:
# sct.exciter.file_id          ../run/fastsweep/duc_settings.txt
# sct.exciter.cic_scale                  11
# sct.exciter.cic2_dec                   96
# sct.exciter.cic2_interp               625
# sct.exciter.cic5_interp                 2
# sct.exciter.rcf_type         ADToolsExample
# sct.exciter.rcf_taps                   72
# sct.exciter.rcf_taps_phase              6
# sct.exciter.coefficients
#     1  -226
#     2    13
#     3   806
#     4  1306
#     5   732
#     6   -11
#     7    47
#     8    62
#     9   880
#    10  1297
#    11   652
#    12   -52
#    13  -169
#    14   100
#    15   948
#    16  1279
#    17   576
#    18   -59
#    19   -46
#    20   159
#    21  1014
#    22  1252
#    23   496
#    24   -96
#    25  -141
#    26   212
#    27  1073
#    28  1218
#    29   422
#    30   -84
#    31   -82
#    32   280
#    33  1128
#    34  1176
#    35   345
#    36  -124
#    37  -124
#    38   345
#    39  1176
#    40  1128
#    41   280
#    42   -82
#    43   -84
#    44   422
#    45  1218
#    46  1073
#    47   212
#    48  -141
#    49   -96
#    50   496
```

```
#    51  1252
#    52  1014
#    53   159
#    54   -46
#    55   -59
#    56   576
#    57  1279
#    58   948
#    59   100
#    60  -169
#    61   -52
#    62   652
#    63  1297
#    64   880
#    65    62
#    66    47
#    67   -11
#    68   732
#    69  1306
#    70   806
#    71    13
#    72  -226
# sct.exciter.user
#
# MONITOR:
# sct.monitor.balun_currents    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
# sct.monitor.balun_status      00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
# sct.monitor.front_end_status 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
# sct.monitor.receiver_status  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
# sct.monitor.exciter_status    00000000 00000000
# sct.monitor.user
          1
#
# PCT:
# pct.record_id                          1
# pct.pri_ut                      54424.00
# pct.pri_time_offset                 0.00
# pct.base_id                            1
# pct.pulse_id                           1
# pct.ramp_id                            1
# pct.repeat_id                          1
# pct.frequency                    1000.00  0.100000E+04
# pct.nco_tune_word             0x03333333      53687091      1000.000
# pct.drive_attenuation               0.00
# pct.pa_flags                  0x00000000
# pct.pa_forward_power                0.00
# pct.pa_reflected_power              0.00
# pct.pa_vswr                         0.00
# pct.pa_temperature                  0.00
# pct.procq_range_count                  0
# pct.proc_noise_level                0.00
# pct.user:
```