The VIPER Code
Voyager **I**on **P**LS **E**xperiment **R**esponse

RE-ANALYSIS OF VOYAGER PLS DATA AT JUPITER
LASP - University of Colorado - Boulder
10 February 2017

Logan Dougherty
Kaleb Bodisch
Working with Fran Bagenal (bagenal@colorado.edu)
With help from Rob Wilson

http://lasp.colorado.edu/home/mop/missions/voyager/

Table of Contents

1. Introduction

Motivated by the arrival of the Juno mission to Jupiter in 2016, we decided to look back at the Voyager data obtained at Jupiter.

In order to preserve, archive and re-analyze data obtained at Jupiter by the Plasma Science (PLS) instruments, we decided to develop new code rather than try to run the MIT code. We decided to use IDL (rather than the original Fortran) because it was the programming language learned by CU undergraduates a few years ago. But the code could probably be translated to other similar languages. The intention was to document everything so that anyone could fairly easily re-analyze the data at any point in the future.

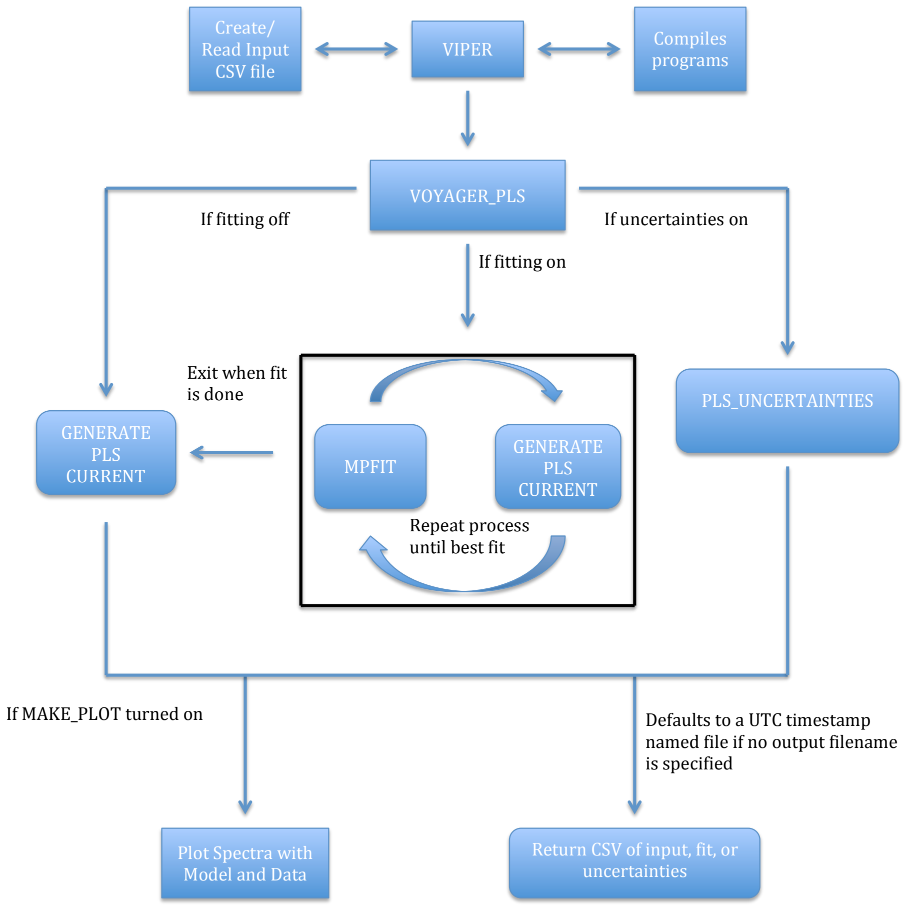
The code described in the manual analyzes data from the PLS instrument on the Voyager 1 and 2 spacecraft using a numerical response function for the Faraday cups. The response function is based on the analysis done by Alan Barnett in his thesis work. The source code for the programs was originally in FORTRAN but has been translated into IDL for present use. This document is an operation manual for the code and serves as documentation for the method of fitting the PLS data at Jupiter.

To access the Voyager 1 & 2 data from Jupiter, the code and a ReadMe file that walks the user through how to run the code, go to http://lasp.colorado.edu/home/mop/missions/voyager/.

The current version of the code is VIPER 2.

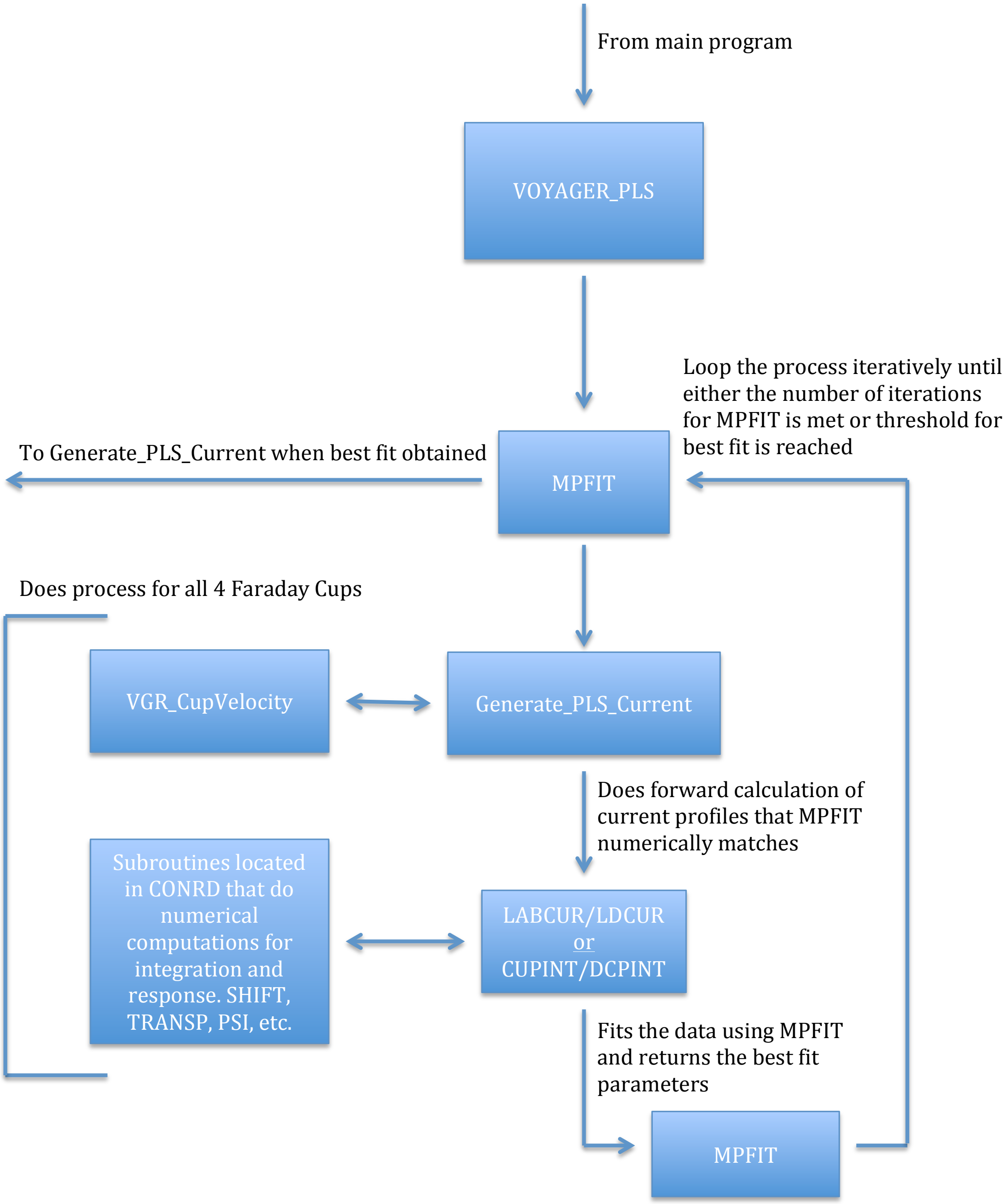
A. Block Diagram

Block Diagram for Voyager PLS Analysis


Create/ Read Input CSV file
VIPER
Compiles programs
VOYAGER_PLS
If fitting off
If uncertainties on
If fitting on
Exit when fit is done
GENERATE PLS CURRENT
MPFIT
GENERATE PLS CURRENT
Repeat process until best fit
PLS_UNCERTAINTIES
If MAKE_PLOT turned on
Defaults to a UTC timestamp named file if no output filename is specified
Plot Spectra with Model and Data
Return CSV of input, fit, or uncertainties

Fitting procedure in greater detail on the next page

Block Diagram for Voyager PLS Analysis Fitting


VOYAGER_PLS
MPFIT
VGR_CupVelocity
Generate_PLS_Current
Subroutines located in CONRD that do numerical computations for integration and response. SHIFT, TRANSP, PSI, etc.
LABCUR/LDCUR or CUPINT/DCPINT
MPFIT

From main program

Loop the process iteratively until either the number of iterations for MPFIT is met or threshold for best fit is reached

To Generate_PLS_Current when best fit obtained

Does process for all 4 Faraday Cups

Does forward calculation of current profiles that MPFIT numerically matches

Fits the data using MPFIT and returns the best fit parameters

2. VIPER 2.2

A. Organization

The VIPER 2.0 ZIP file includes all code and data used in the analysis. Inside the ZIP file there are four subdirectories. CSV_Files contains all of the .csv files that are used for deciding what time stamp to display and what parameters should be fit during the fitting process. Data contains Voyager PLS data for the Jupiter encounter for both Voyager 1 and Voyager 2, chemistry ratios from analysis from Delamere et al. (2005), information about the spacecraft trajectories, including spacecraft pointing information from the SPICE database, constants for numerical integration for the instrument, and the responses of the Faraday cups to a cold corotating beam of plasma in the jovian system. The miscellaneous directory was used for various purposes in obtaining lookup tables for the chemistry ratios or for early analysis by the JUNO JADE science team. The output folder is where plots of spectra will be saved and where .txt files of reduced chi-squared calculations are stored.

B. Setting up the Code

1. Download the .zip file from the MOP website
2. Unzip the file
3. Set the IDL directory to that folder
4. The code is ready to run by calling VIPER

More detail on running the code and all programs are described starting in section 4.

3. Necessary Files

The following section describes all text files that are necessary for this program to be run for the analysis of PLS data during the Voyager 1 and 2 flybys of Jupiter. The name and required location in the VIPER file system are given. All files are included in the downloadable zip file, but in the instance that a file is either changed or deleted, these files are located as a separate download.

A. mjs.out.txt

This text file contains lookup tables for the numerical integration performed in the LABCUR/LDCUR and CUPINT/DCPINT programs for the computation of the response function for the PLS instrument on the Voyager 1 spacecraft. It is a large file that contains 121,991 numerical values with 6 digits of precision total.

This file comes from MJSANL based on the work in Alan Barnett's thesis. This file is located in /Data/Instrument_Constants

B. v1jup.ssedr.62-65.79_rob.txt

This file contains trajectory information for the Voyager 1 spacecraft between days 62 and 65 of the year 1979. It contains the position and velocity of the spacecraft in both the System III coordinate system of Jupiter (IAU Jupiter) and the ECL50 coordinate system based off of the ecliptic plane in 1950, but centered around Jupiter. It also contains a rotation matrix from the ECL50 coordinate system to the spacecraft pointing coordinate system. Example output on the next page.

This file is located in /Data/Jupiter/Trajectories

C. v2jup.ssedr.rob.txt

This file contains trajectory information for the Voyager 2 spacecraft between days 183 and 215 of the year 1979. There are time gaps in the data in this file. It contains the position and velocity of the spacecraft in both the System III coordinate system of Jupiter (IAU Jupiter) and the ECL50 coordinate system based off of the ecliptic plane in 1950, but centered around Jupiter. It also contains a rotation matrix from the ECL50 coordinate system to the spacecraft pointing coordinate system. This file has the same format as the Voyager 1 SSEDR file above, however it seems as if there was a slight error with the creation of this text file. For many of the time stamps, there are either 0 or 1000 milliseconds (trivially just 1 second). This precision may not matter but if using SPICE UTC to Ephemeris time , or other similar tools, this may need to be accounted for because it will not correctly read millisecond data to more than 3 numbers and will return an error.

This file is located in /Data/Jupiter/Trajectories

Output from SSEDR file - using Voyager Memo 168 revised April 1994 by Adam Szabo

```
1-6     Time    1979  62  6  0  35  970
7-12    S/Cs3  -2.24571E+06   2.22237E+06   109583.   398.241   383.075  -0.781218
```

This is spacecraft position and velocity in TPMED (true prime meridian and equator of date)
This is System III for Jupiter.
Position in km translates to (-31.5,31.1,1.5)= 44.3 Rj
Velocity is in km/s - magnitude of 500 km/s on approach to Jupiter
- note this is Jupiter's rotating frame.

```
13-18   ECL50 1.23683E+06   -2.90926E+06   24721.7   -2.38935    13.7913  -0.327090
```

This is s/c position and velocity in ECL50 - ecliptic coordinates.
Velocity in this Jupiter's centered (non-rotating) frame is 14 km/s.

```
19-30          Io        420814.   -14762.9    60.3238
TPMED          Euro    -574900.    356376.  -4574.32
               Gany    -790897.   -723218.   -4183.84
               Call     995241.   -1.60692E+06   -3067.89
```

Satellite locations in System III.

```
31-33
ECL50   Call   -313632.    1.86304E+06    58492.5
```

Callisto location in ECL50

34-39	Point time	1979 62 5 59 47 984
40-48	Matrix	-0.682896 -0.538747 -0.493359 -0.380669 -0.313975 0.869775 -0.623492 0.781774 9.32826E-03

This matrix rotates a vector from ECL50 into s/c coordinates.
Note this is Fortran ordering of the matrix.

For comparison - here is the output from the Iowa ephemeris tool
http://cassini.physics.uiowa.edu/das/ephem5

```
                              Geographic Coordinate System
                                    1 Rj = 71492.0 km

    SCET (UT)          W Lon (deg)     Lat (deg)    MLat (deg)      LT (hrs)        R (Rj)  Io phase (deg)
---------------------    -----------     ----------    ----------    ----------    ----------  --------------
1979 062 06 00 35.000       224.108          1.982         10.870         11.094         44.226      226.672
1979 062 06 05 35.000       227.114          1.980         10.665         11.095         44.169      227.363

                              Ecliptic Coordinate System

    SCET (UT)              X (Rj)        Y (Rj)        Z (Rj)        R (Rj)   1 Rj = 71492.0 km
---------------------    ----------    ----------    ----------    ----------
1979 062 06 00 35.000         42.974       -10.432          0.572        44.226
1979 062 06 05 35.000         42.922       -10.406          0.571        44.169

                            Equatorial Coordinate System

    SCET (UT)              X (Rj)        Y (Rj)        Z (Rj)        R (Rj)   1 Rj = 71492.0 km
---------------------    ----------    ----------    ----------    ----------
1979 062 06 00 35.000         42.962       -10.384          1.530        44.226
1979 062 06 05 35.000         42.910       -10.358          1.526        44.169
```

D. SpiceRotations-vgr1.txt

Note: If you use this file, you do not need to utilize JPL's SPICE database.

This file contains data for 6x6 rotation matrices for state vectors from System III to ECL50 at the same timestamps as the data in the Voyager 1 SSEDR file above for the spacecraft trajectory timestamps. These are columns 1-6 in the SSEDR file. This file was generated to validate the consistency of the SPICE kernel data with the trajectory information in the SSEDR and is used in the conversion of the plasma flow speed to velocity in the frames of the Faraday Cups. Due to the nature of the System III coordinate system and how it is body fixed and thus rotates compared to the ECL50 coordinate system, this transformation matrix is different for each different time stamp. This text file was generated from use of the SPICE database as described in the SPICE section.

This file is located in /Data/Jupiter/Trajectories

E. SpiceRotations-vgr2.txt

This is the same as section D above, but for Voyager 2.

F. day62.txt

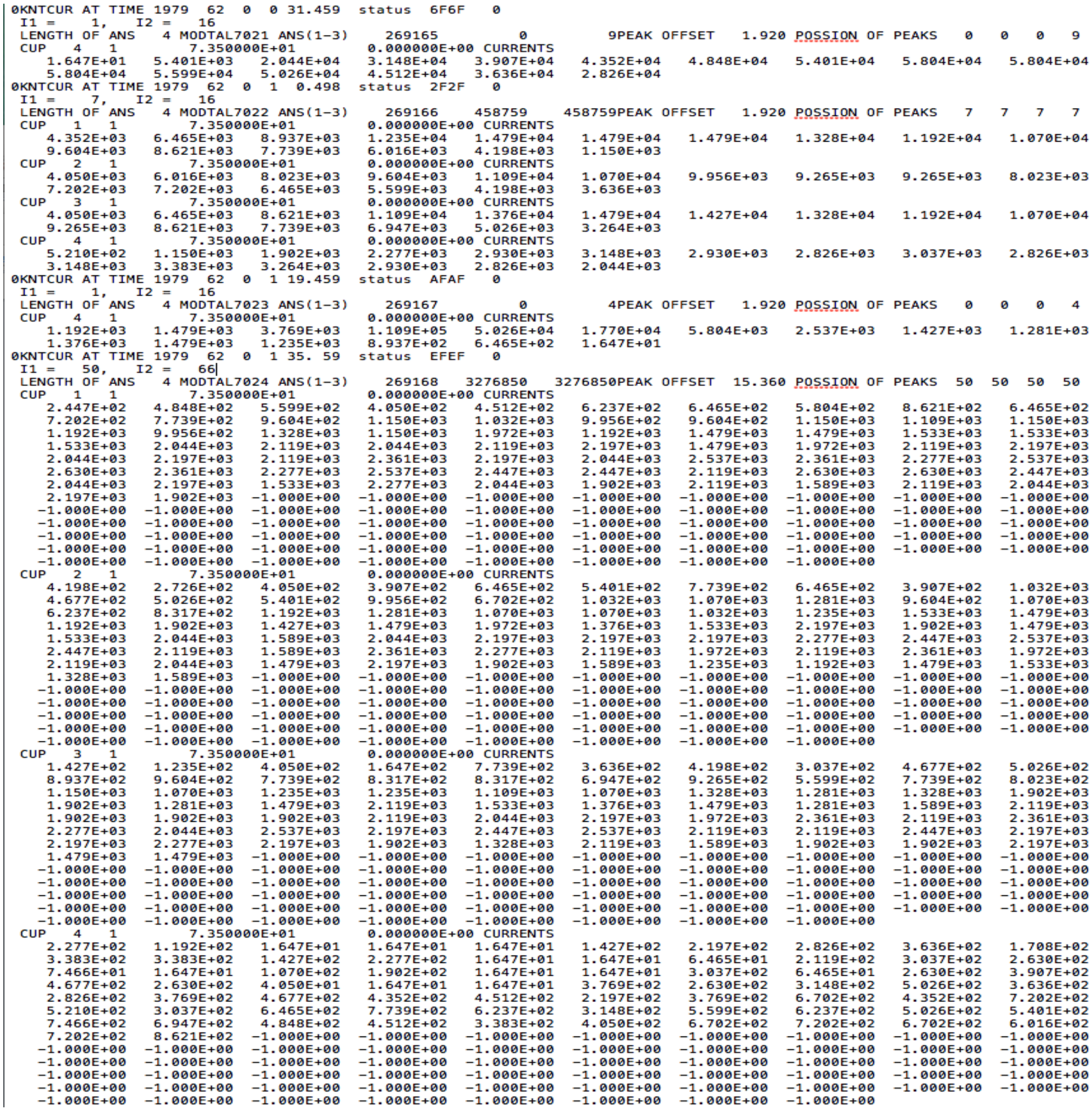
Contains PLS data for day 62 of the year 1979. Data was taken every 96 seconds and this file includes both high resolution (M-mode) and low resolution (L-mode) PLS data for all 4 Faraday cups on the Voyager spacecraft. The data is in a difficult format for IDL to read in, so the IDLVoyager.pro exists to read the data. On the next page is the data for a single timestamp.

This file is located in /Data/Jupiter/PLS_Data

G. day___.txt

Same as above but for different days of the year. Data exists for days 59-81 for Voyager 1 and for days 183-215 for Voyager 2 around Jupiter.

Included in the VIPER zip is data for DOY 62, 63, and 64 for Voyager 1 and 189 and 190 for Voyager 2. These are the text files in which our analysis was done, where the plasma had a considerable signal and was solar wind data. Due to the large size of these files, the remaining files were not included in the VIPER zip but are included on the website as a separate zipped download if desired.



```
0KNTCUR AT TIME 1979  62  0  0 31.459  status  6F6F   0
 I1 =    1,   I2 =   16
 LENGTH OF ANS   4 MODTAL7021 ANS(1-3)    269165         0        9PEAK OFFSET   1.920 POSSION OF PEAKS   0   0   0   9
 CUP   4   1         7.350000E+01         0.000000E+00 CURRENTS
    1.647E+01   5.401E+03   2.044E+04   3.148E+04   3.907E+04   4.352E+04   4.848E+04   5.401E+04   5.804E+04   5.804E+04
    5.804E+04   5.599E+04   5.026E+04   4.512E+04   3.636E+04   2.826E+04
0KNTCUR AT TIME 1979  62  0  1  0.498  status  2F2F   0
 I1 =    7,   I2 =   16
 LENGTH OF ANS   4 MODTAL7022 ANS(1-3)    269166    458759    458759PEAK OFFSET   1.920 POSSION OF PEAKS   7   7   7   7
 CUP   1   1         7.350000E+01         0.000000E+00 CURRENTS
    4.352E+03   6.465E+03   8.937E+03   1.235E+04   1.479E+04   1.479E+04   1.479E+04   1.328E+04   1.192E+04   1.070E+04
    9.604E+03   8.621E+03   7.739E+03   6.016E+03   4.198E+03   1.150E+03
 CUP   2   1         7.350000E+01         0.000000E+00 CURRENTS
    4.050E+03   6.016E+03   8.023E+03   9.604E+03   1.109E+04   1.070E+04   9.956E+03   9.265E+03   9.265E+03   8.023E+03
    7.202E+03   7.202E+03   6.465E+03   5.599E+03   4.198E+03   3.636E+03
 CUP   3   1         7.350000E+01         0.000000E+00 CURRENTS
    4.050E+03   6.465E+03   8.621E+03   1.109E+04   1.376E+04   1.479E+04   1.427E+04   1.328E+04   1.192E+04   1.070E+04
    9.265E+03   8.621E+03   7.739E+03   6.947E+03   5.026E+03   3.264E+03
 CUP   4   1         7.350000E+01         0.000000E+00 CURRENTS
    5.210E+02   1.150E+03   1.902E+03   2.277E+03   2.930E+03   3.148E+03   2.930E+03   2.826E+03   3.037E+03   2.826E+03
    3.148E+03   3.383E+03   3.264E+03   2.930E+03   2.826E+03   2.044E+03
0KNTCUR AT TIME 1979  62  0  1 19.459  status  AFAF   0
 I1 =    1,   I2 =   16
 LENGTH OF ANS   4 MODTAL7023 ANS(1-3)    269167         0        4PEAK OFFSET   1.920 POSSION OF PEAKS   0   0   0   4
 CUP   4   1         7.350000E+01         0.000000E+00 CURRENTS
    1.192E+03   1.479E+03   3.769E+03   1.109E+05   5.026E+04   1.770E+04   5.804E+03   2.537E+03   1.427E+03   1.281E+03
    1.376E+03   1.479E+03   1.235E+03   8.937E+02   6.465E+02   1.647E+01
0KNTCUR AT TIME 1979  62  0  1 35. 59  status  EFEF   0
 I1 =   50,   I2 =   66
 LENGTH OF ANS   4 MODTAL7024 ANS(1-3)    269168   3276850   3276850PEAK OFFSET  15.360 POSSION OF PEAKS  50  50  50  50
 CUP   1   1         7.350000E+01         0.000000E+00 CURRENTS
    2.447E+02   4.848E+02   5.599E+02   4.050E+02   4.512E+02   6.237E+02   6.465E+02   5.804E+02   8.621E+02   6.465E+02
    7.202E+02   7.739E+02   9.604E+02   1.150E+03   1.032E+03   9.956E+02   9.604E+02   1.150E+03   1.109E+03   1.150E+03
    1.192E+03   9.956E+02   1.328E+03   1.150E+03   1.972E+03   1.192E+03   1.479E+03   1.479E+03   1.533E+03   1.533E+03
    1.533E+03   2.044E+03   2.119E+03   2.044E+03   2.119E+03   2.197E+03   1.479E+03   1.972E+03   2.119E+03   2.197E+03
    2.044E+03   2.197E+03   2.119E+03   2.361E+03   2.197E+03   2.044E+03   2.537E+03   2.361E+03   2.277E+03   2.537E+03
    2.630E+03   2.361E+03   2.277E+03   2.537E+03   2.447E+03   2.447E+03   2.119E+03   2.630E+03   2.630E+03   2.447E+03
    2.044E+03   2.197E+03   1.533E+03   2.277E+03   2.044E+03   1.902E+03   2.119E+03   1.589E+03   2.119E+03   2.044E+03
    2.197E+03   1.902E+03  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
 CUP   2   1         7.350000E+01         0.000000E+00 CURRENTS
    4.198E+02   2.726E+02   4.050E+02   3.907E+02   6.465E+02   5.401E+02   7.739E+02   6.465E+02   3.907E+02   1.032E+03
    4.677E+02   5.026E+02   5.401E+02   9.956E+02   6.702E+02   1.032E+03   1.070E+03   1.281E+03   9.604E+02   1.070E+03
    6.237E+02   8.317E+02   1.192E+03   1.281E+03   1.070E+03   1.070E+03   1.032E+03   1.235E+03   1.533E+03   1.479E+03
    1.192E+03   1.902E+03   1.427E+03   1.479E+03   1.972E+03   1.376E+03   1.533E+03   2.197E+03   1.902E+03   1.479E+03
    1.533E+03   2.044E+03   1.589E+03   2.044E+03   2.197E+03   2.197E+03   2.197E+03   2.277E+03   2.447E+03   2.537E+03
    2.447E+03   2.119E+03   1.589E+03   2.361E+03   2.277E+03   2.119E+03   1.972E+03   2.119E+03   2.361E+03   1.972E+03
    2.119E+03   2.044E+03   1.479E+03   2.197E+03   1.902E+03   1.589E+03   1.235E+03   1.192E+03   1.479E+03   1.533E+03
    1.328E+03   1.589E+03  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
 CUP   3   1         7.350000E+01         0.000000E+00 CURRENTS
    1.427E+02   1.235E+02   4.050E+02   1.647E+02   7.739E+02   3.636E+02   4.198E+02   3.037E+02   4.677E+02   5.026E+02
    8.937E+02   9.604E+02   7.739E+02   8.317E+02   8.317E+02   6.947E+02   9.265E+02   5.599E+02   7.739E+02   8.023E+02
    1.150E+03   1.070E+03   1.235E+03   1.235E+03   1.109E+03   1.070E+03   1.328E+03   1.281E+03   1.328E+03   1.902E+03
    1.902E+03   1.281E+03   1.479E+03   2.119E+03   1.533E+03   1.376E+03   1.479E+03   1.281E+03   1.589E+03   2.119E+03
    1.902E+03   1.902E+03   1.902E+03   2.119E+03   2.044E+03   2.197E+03   1.972E+03   2.361E+03   2.119E+03   2.361E+03
    2.277E+03   2.044E+03   2.537E+03   2.197E+03   2.447E+03   2.537E+03   2.119E+03   2.119E+03   2.447E+03   2.197E+03
    2.197E+03   2.277E+03   2.197E+03   1.902E+03   1.328E+03   2.119E+03   1.589E+03   1.902E+03   1.902E+03   2.197E+03
    1.479E+03   1.479E+03  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
 CUP   4   1         7.350000E+01         0.000000E+00 CURRENTS
    2.277E+02   1.192E+02   1.647E+01   1.647E+01   1.647E+01   1.427E+02   2.197E+02   2.826E+02   3.636E+02   1.708E+02
    3.383E+02   3.383E+02   1.427E+02   2.277E+02   1.647E+01   1.647E+01   6.465E+01   2.119E+02   3.037E+02   2.630E+02
    7.466E+01   1.647E+01   1.070E+02   1.902E+02   1.647E+01   1.647E+01   3.037E+02   6.465E+01   2.630E+02   3.907E+02
    4.677E+02   2.630E+02   4.050E+01   1.647E+01   1.647E+01   3.769E+02   2.630E+02   3.148E+02   5.026E+02   3.636E+02
    2.826E+02   3.769E+02   4.677E+02   4.352E+02   4.512E+02   2.197E+02   3.769E+02   6.702E+02   4.352E+02   7.202E+02
    5.210E+02   3.037E+02   6.465E+02   7.739E+02   6.237E+02   3.148E+02   5.599E+02   6.237E+02   5.026E+02   5.401E+02
    7.466E+02   6.947E+02   4.848E+02   4.512E+02   3.383E+02   4.050E+02   6.702E+02   7.202E+02   6.702E+02   6.016E+02
    7.202E+02   8.621E+02  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
   -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00  -1.000E+00
```

F. Delamere_Ratios.txt

This file contains ratios of the ion species between 6 and 9 Rj based off of analysis from Delamere et al. (2005). The columns of the text file are in the following order: Rj, nO+/nO+, nO++/nO+, nS+++/nO+, nS++/nO+, nS+/nO+. These ratios will be used for whatever species the user specifies if the Delamere ratios tag is turned on.

4. Additional But Not Necessary Files

- Cassini_UVIS_Ratios.txt
    - Located in the /Data/Jupiter/Chemistry_Ratios folder, it does the same thing as the Delamere_Ratios.txt file but was based on few Cassini UVIS measurements. It was not used for any of the fitting routines. VIPER still has a keyword argument to use it if desired, but the accuracy of those fits would be questionable.
- V1_Cup_Responses.txt
    - Located in the /Data/Instrument_Constants folder, this text file, as well as the similarly named V2 file, are the flowspeeds in the Faraday cup coordinate frame based off of a cold corotating beam.

5. VIPER

VIPER begins by compiling all necessary procedures and functions as well as building common blocks that contain the instrument look up tables. Once all functions are compiled and common blocks built, VIPER sets up an IDL structure format to process the data. To run the program, all that one needs to do is type VIPER, ‘j’ into the command line in IDL, or call it from another program.

During our analysis, an input filename was always provided with the keyword argument *INPUT_FILENAME* defined below. The input for this is a string of the .csv filename that is in the CSV_Files directory (including the .csv extension in the string argument). The code is also set to run without an input filename. In this instance, it will generate a csv file and place it on the screen. This csv can be edited and saved before returning to the program. By continuing the program, it will analyze the input that was just provided. This, however, is limited to just one timestamp of PLS data for analysis, so the preferred method was to input a csv filename that contained several different spectral plots for analysis. VIPER saves the csv information as a structure that can be passed through multiple programs for complete analysis.

The csv format is shown below with a description of every column and what each setting does. Below that, keyword arguments are included for the VIPER program. This program is written so that not many keywords are needed. Most of the keywords simply make modifications to the csv file for quick use at the command line.

Inputs

PlanetName

String that determines what planet the spacecraft is around (Currently only works for *Jupiter*: 'J', ‘JUP’, or ‘JUPITER’ either capitalized or lowercase).

Other options can be input but the code will not do any calculations. These values below were only programmed in so that eventually a pipeline for other planetary systems could be included. It would use almost all of the same functions, just different lookup tables for trajectories.

‘S’, ‘SAT’, or ‘SATURN’ will have the code print a 2
‘U’, ‘UR’, ‘URA’, or ‘URANUS’ will have the code print a 3
‘N’, ‘NEP’, or ‘NEPTUNE’ will have the code print a 4
If nothing that matches is entered, it will produce an error message.

Keyword Arguments

MAKE_PLOT

Keyword argument that plots the data. *Additional* argument in the CSV, or keyword below, to save the plot automatically

UNCERTAINTIES

Keyword argument that when set will calculate the uncertainties of the parameters that are being fit

INPUT_FILENAME

Keyword argument for the input file that should be used for running the code

CSV_INDICES

Keyword argument that will tell the program what rows of the CSV file to use. *Note* that CSVs starts on row 2

FULL_CSV

Keyword argument to process the full csv. *This* is the default option if the above isn't specified

NO_CSV_RETURN

Keyword argument to prevent the return of a csv file - Useful for just plotting the data. Note that this suppresses the output of the program completely

OUTPUT_FILENAME

Keyword argument to specify the output name of the csv file being returned

CLEAR

Keyword argument that will clear any existing graphic windows

FORCE_SAVE_JPG

Keyword argument that overrides what is written in the CSV and automatically saves a JPG of the plot. *May* not plot on screen

NO_FIT

Keyword argument to explicitly turn off fitting procedure - Useful for some batch jobs to manipulate CSV files from console

FIT

Keyword argument to explicitly turn on fitting procedure

CASSINI_COMPOSITION

Keyword argument to use the CASSINI UVIS observations for constraints on the composition of ion species. *Obtained* from analysis by Edward Nerney

COMPUTE_CHI_SPACE

Keyword argument to calculate chi space for the given parameters

N_CHI_PTS

Keyword argument for number of points to use (Creates a square array of that size)

N_CHI_SIG

Keyword argument for number of sigma to deviate from best fit

INPUT_UNCERTAINTY_FILENAME

Keyword argument with name of uncertainty file. *For* use with the chi space computation since it needs the best fit value and the 1 sigma value. *If* no name is entered, then it will assume the autosaved filename for *uncertainties*: 'filename_Unc.csv'

PLOT_QUIETLY

Keyword argument that produces and saves the plots but does so without popping them up on the computer screen. The code does this by adding 100 to the Save_Plot structure tag that the code then understands

DELAMERE_COMPOSITION

Keyword argument to use the composition from Delamere's 2005 Paper. Roughly 30 points were extrapolated from his plots of mixing ratios and then interpolated to give our tables. These values are in Data/Jupiter/Chemistry_Ratios: 'Delamere_Ratios.txt'. *Columns* are in the following *order*: Rj, nO+/nO+, nO++/nO+, nS+++/nO+, nS++/nO+, nS+/nO+. The chemistry model is valid for 6-9 Rj. *Before* this point, the calculation has not been done. *Past* this point, it is assumed that mixing has stopped and that the density ratios are the same as the values at 9 Rj.

NO_PLOT_SAVE

Keyword argument to turn off plot saving

POD

Keyword argument to specify Plot Output Directory (POD)

ANALYTICAL_VPHI

Keyword argument that specifies an analytical equation to use for the Vphi profile. *It* defaults to a 1, which is a hyperbolic tangent model

CUP[A-D]

Keyword argument if you would only like to plot a single cup A-D after running a spectra through Viper

CENTRIFUGAL_EQUATOR

Keyword argument that tells Viper to walk along modeled field lines at the location of the spacecraft until reaching the centrifugal equator

ANISOTROPIC

Keyword argument to over ride the assumption of an isotropic system in favor of the value input by the user in this keyword

Note: In order to run Viper you simply have to run a call to Viper in the IDL command line twice. On the first run, the call to Viper will fail due to build order, the second time viper is called it will successfully compile all programs and execute the function call.

All of the functions/procedures in the sections following mainly just handle data and how it is processed. Section 6 describes the model and the mathematics behind calculating the model in detail.

A. CONRD

This program contains the common blocks that are used to store the data for numerical integration for the PLS instrument response. The program uses common blocks to primarily store the lookup tables, given by mjs.out.txt, for the numerical integration and calculation of the response function in LABCUR and LDCUR. CUPINT and DCPINT use a few values from the common blocks, but rely heavily on an analytical calculation. These common blocks exist so that the lookup tables can easily be called in several different programs.

The common blocks are also used for storing global variables in the IDL procedure Save_Structures. These global variables are just the structures used in the fitting process since mpfit passes data in a different way than our code was setup and this avoided rewriting large parts of the IDL code. Also stored in common blocks are information about the trajectory of the spacecraft, including its pointing information from the SPICE kernels to reduce computation time from reading in those files.

B. VOYAGER_PLS

This program is the overarching function that processes the data. It reads in the PLS data for the corresponding timestamp to the one provided. From there, the code enters into one of 3 modes. The first is a fitting mode. In this mode, the code takes the structure provided and fits the parameters that were specified as free parameters by the user. It uses a chi-squared minimization technique called mpfit. The other mode is to calculate uncertainties. Uncertainties are calculated by finding the curvature matrix of the data in n-dimensions, where n is the number of free parameters. From this matrix, it is possible to determine the uncertainty in the fit parameter. If the parameter is not at a minimum in chi-squared space, these uncertainty values could be returned as NaNs. More information on the methods can be found in Wilson (2015). The third method is calculating chi-squared space. This requires large numbers of model calculations and uses a Matlab code to plot the results. This was used for our analysis to show goodness of fit. The results can be found on the website for all 772 Voyager 1 fits and 103 Voyager 2 spectra fits. These methods require many calculations from the Generate_PLS_Currents function described in part B.

From here, if specified, the code will enter into the plotting routine where it will make a plot of the both the PLS data and the modeled spectrum. The current of each ion species will be shown so that it is possible to see each contribution to the total current. If the plot is set to save, it will be saved to the Plot Output Directory (POD).

Inputs

Structure

A structure containing information on how to handle all parameters in the fit

POD

Directory name for where to save the plot outputs. It creates an output folder of this name if one does not already exist. If not specified by the VIPER keyword, then it defaults to '/Output/Plots'.

CUPSS

Argument 0-4 denoting which cup plots to output at the completion of the code. Default value is 0 unless changed with keyword argument in call to viper all 4 cups will be plotted. Otherwise an argument of 1 = A cup, 2 = B cup, 3 = C cup, and 4 = D cup.

Keyword Arguments

MAKE_PLOT

Keyword argument that plots the data. *Additional* argument in the CSV, or keyword below, to save the plot automatically

UNCERTAINTIES

Keyword argument that determines whether uncertainties are being calculate, gets pulled from Viper and is superseded by tag in CSV

Structure_Unc

Keyword argument in the form of a structure which contain the results of the uncertainties

N_CHI_PTS

Keyword argument for number of points to use

N_CHI_SIG

Keyword argument for number of sigma to deviate from best fit during reduced chi square calculations

COMPUTE_CHI_SPACE

Keyword argument to calculate chi space for the given parameters

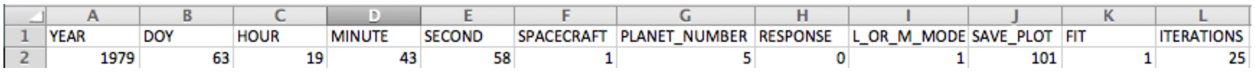
CEq

ANISOTROPIC

Keyword argument to over ride the assumption of an isotropic system in favor of the value input by the user in this keyword

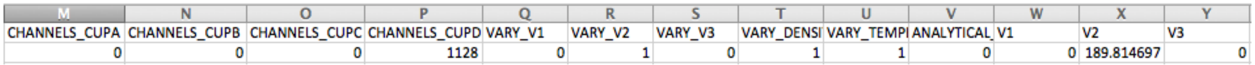
Input/Output

CSV file- put into structure format within the code



	A	B	C	D	E	F	G	H	I	J	K	L
1	YEAR	DOY	HOUR	MINUTE	SECOND	SPACECRAFT	PLANET_NUMBER	RESPONSE	L_OR_M_MODE	SAVE_PLOT	FIT	ITERATIONS
2	1979	63	19	43	58	1	5	0	1	101	1	25

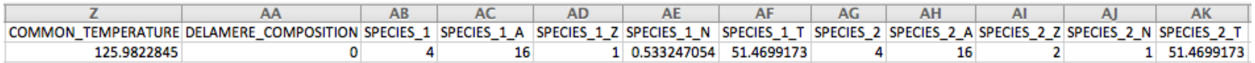
- Year (1)
- Day of Year (2)
- Hour, Minute, Second (3-5)
- Spacecraft (6)
- Planet Number (7)
- Response (8)
    - 0 for CUPINT/DCPINT (default)
    - 1 for LABCUR/LDCUR
- L or M-Mode (9)
    - 1 for L-Mode
    - 2 for M-Mode (default)
- Save Plot (10)
    - 0 to not save plot output (default)
    - 1 to save plot output
- Fit (11)
    - 0 for no fitting routine (default)
    - 1 for custom fitting routine
    - 2 to fit with all parameters as free variables
- Iterations (12)
    - Insert number of iterations for MPFIT to go through before stopping fit (default =25)



M	N	O	P	Q	R	S	T	U	V	W	X	Y
CHANNELS_CUPA	CHANNELS_CUPB	CHANNELS_CUPC	CHANNELS_CUPD	VARY_V1	VARY_V2	VARY_V3	VARY_DENSI	VARY_TEMPI	ANALYTICAL_	V1	V2	V3
0	0	0	1128	0	1	0	1	1	0	0	189.814697	0

- Channels [cup A – cup D] (13-16)
    - 0 do not fit channels in the cup
    - 1 fit all channels (default)
    - six digit number to specify channel range to fit
    - Example: to fit channels 40-116 Input 040116
        - Excel will remove leading 0's (IDL code takes care of that)
        - If inappropriate channel values entered defaults to full range
- Vary Flow Speed (17-19
    - 0 to hold constant

    - 1 to vary
- Vary Density (20)
    - 0 to keep all densities as constants
    - 1 to allow densities to vary (will vary if species tag designates given species as a free parameter)
- Vary Temperature (20)
    - 0 to keep all Temperatures as constants
    - 1 to allow Temperatures to vary (will vary if species tag designates given species as a free parameter)
- Analytical Vphi (21)
    - 0 if input Vphi value to be used
    - 1 if you want to use derived hyperbolic curve to Voyager Data
    - 2 if you want to use strict corotation to the planet spin
- V1 (22)
    - Set to V1 flow speed (for Voyager Vr)
- V2 (23)
    - Set to V2 flow speed (for Voyager Vphi)
- V3 (24)
    - Set to V3 flow speed (for Voyager Vz)



Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK
COMMON_TEMPERATURE	DELAMERE_COMPOSITION	SPECIES_1	SPECIES_1_A	SPECIES_1_Z	SPECIES_1_N	SPECIES_1_T	SPECIES_2	SPECIES_2_A	SPECIES_2_Z	SPECIES_2_N	SPECIES_2_T
125.9822845	0	4	16	1	0.533247054	51.4699173	4	16	2	1	51.4699173

- Common Temperature (25)
    - Set this value to the temperature (eV) that the ion species with appropriate species tags will follow
- Delamere Composition (26) - (O+, O++, S+, S++, S+++)
    - 0 to allow species densities defined by delamere composition to follow the fitting routine as given by there species tag (see Species 1)
    - 1 to force the species defined by delamere composition to follow delamere composition
- Species 1
    - 0 – This species is not in parameter space and is a constant of the model. It will follow the common temperature value if it is greater than 0 and the species temperature will be neglected.
    - 1 – This species is not in parameter space and does not follow the common temperature value, but follows its own temperature value.
    - 2 – This species is in parameter space with its density and temperature as a parameter. IF common temperature is greater than 0, then the species temperature will be ignored.
    - 3 – This species is in parameter space but does not follow the common temperature value, eve if specified to be above 0.
    - 4 – This species is taken indirectly into parameter space. This value means that the
- Species 1 A
    - Mass of ion species (1 for H+, 16 for O+ and O++, etc.)

- Species 1 Z
    - o Charge of ion species (1 for O+, 2 for O++, etc.)
- Species 1 N
    - o Density of ion species, given in number/cc, unless argument of 4 is given as species tag. If 4 is given and it is the first 4 it is a density, if it is not the first species tag of 4 it is a fractional density to be multiplied by the first species density with a species tag of 4.
- Species 1 T
    - o SPECIES_1_T – Temperature in eV, of Species. A common temperature will generally supersede this value, see species tags for details.

- Species X
    - o Same as above where X is the number of the species

- Species X A
    - o Same as above where X is the number of the species

- Species X Z
    - o Same as above where X is the number of the species

- Species X N
    - o Same as above where X is the number of the species

- Species X T
    - o Same as above where X is the number of the species

C. FIT_PARAMETERS

Fits Parameters reads in both the input parameters and the Voyager Current data. Following reading in the necessary data, fit_parameters uses parinfo to fix and/or put limits on each of the given parameters. For instance it limits the densities and thermal speeds to be above zero, can limit the values to a certain range of answers, or even fix a variable and not allow for MPFIT to vary that variable. Once each parameter has appropriate fit conditions, the fitting procedure, MPFITFUN, is now run. MPFITFUN works by communicating with generate_spectra_fit as it creates current distributions, from the parameters that are compared through least chi squared method to the given data.

Inputs

X
P
Structure
P_Locations

Keyword Arguments

err=err

Outputs

Total_Current

D. Generate_PLS_Current

Generate Spectra is used near the end of the program as a forward analysis. It either uses the best fit parameters from the MPFIT fitting routine or the input parameters if the fitting routine is turned off to produce current profiles for the different species of ions. First it calls on Vgr1_CupVelocity to transform the flow speed into individual cup velocities. It then utilizes the LABCUR and LDCUR or CUPINT and DCPINT functions to calculate the response of the PLS instrument and return the currents in the Faraday cup due to the numerical model. Generate Spectra is run through a FOR loop to calculate the total current in each of the individual Faraday cups.

GENERATE_SPECTRA_FIT

This function is similar to the above function and uses the same procedures to compute the response function for the Faraday cups. Generate Spectra Fit calculates the current profiles for all 4 of the Faraday cups and combines them into one array. This function is linked to the MPFIT function that calls on it. Its output is compared to the actual data and MPFIT calculates a new set of parameters that mathematically fit the data better. This new set of parameters is then iteratively passed through Generate Spectra Fit again and MPFIT finds a new set of parameters that best fit the data.

This iterative process in MPFIT is done until MPFIT reaches a set tolerance level or until it reaches the number of desired iterations at which point the Generate Spectra Fit function is no longer used.

Inputs

Structure

Keyword Arguments

RETURN_TOTAL_CURRENT
SET_FLOAT
RETURN_PLASMA_PROPERTIES

Outputs

Total_Current
Plasma_Properties

E. PLS_UNCERTAINTIES

This function uses PLS data in conjunction with the best fit parameters in order to calculate the Chi-Squared values for the parameters. The program steps to each side of the best-fit using either an absolute or relative step size, following, assuming a parabolic shape, the function then calculates the Hessian and Covariance matrix to find the formal 1 Sigma deviations from best fit values.

Inputs

Structure
P_Locations

PLS_Data_Fit
PLS_Error
Model_Name
index
Delta

Keyword Arguments
RELATIVE

Outputs
S (Covariance Matrix)

F. CHI_ARRAYS

Computes the Chi-Squared space for the given parameters in the model based off of the central location given in the input structure.

Inputs
Structure
Structure_Unc
P_Locations
PLS_Data_Fit
PLS_Error
Model_Name
index
N_Chi_Steps
N_Sigma
str_date

Outputs
Chi_Array
Multiple Different Files

G. PLS_CURRENT_DATA (Amber Hall, LASP)

This program reads in the data from the day62.txt and subsequent files. The data in the files is in a difficult format for IDL to process with a simple read command and contains information for both the low and high resolution data. This program allows these text files to be read in for the specified time stamp and for the current distribution to be returned in a structure so that it can then be accessed for use in the fitting routine of the code.

Inputs
Structure
Data_Filename
Time
Cup
STR_DIR

Outputs
Data Array

H. MPFIT (Craig B. Markwardt, NASA/GSFC)

MPFIT utilizes the Levenberg-Marquardt method for least-squares minimization. This program provides a best fit series of parameters to a set of data for a given model. In the case of this Voyager PLS analysis, the model comes from the LABCUR and LDCUR or CUPINT and DCPINT programs described below, while the data is taken from the measurements aboard the Voyager 1 spacecraft. MPFIT can constrain the parameters to certain ranges or even fix certain parameters so that they do not affect the fitting procedure. Thus it is possible to fix different parameters depending on the fit being run.

Full documentation on MPFIT and MPFITFUN can be found at the following links.
MPFIT:
https://www.physics.wisc.edu/~craigm/idl/down/mpfit.pro
MPFITFUN:
https://www.physics.wisc.edu/~craigm/idl/down/mpfitfun.pro

I. Miscellaneous Functions

- PLS_DATA_READ
- read_file
  - reads current data from txt file
- read_info
  - reads ssedr trajectory information
- conversionM
  - channelarray = [0:127]   (length 128 array)
  - energyarray = 60. * 10. ^(channelarray / 64.) - 50
- conversionLE2
  - channelarray = [0:15]   (length 16 array)
  - energyarray = 60. * 10. ^(channelarray / 8.) - 50

- conversionE1
  - channelarray = [0:15]   (length 16 array)
  - energyarray = 60. * 10. ^(channelarray / 32.) - 50
- convert_times
  - converts times to decimal Day of Year

6. ADDITIONAL_PLS_FUNCTIONS

A. VGR_CupVelocity

This function reads in the flow of the plasma in a non-rotating Jovicentric coordinate frame. The velocity is read in cylindrical coordinates, R, ϕ, and Z, used for historical reasons. The program also reads in the cup for which to calculate the velocity and then the decimal date of 1979 for which to determine the velocity.

1. Converts the cylindrical plasma flow into a rotating, System III, frame

2. Converts the plasma flow into Cartesian coordinates
3. Transforms both the plasma and the spacecraft state vectors from System III to ECL50 using the SpiceRotations.txt file
4. Flux of the plasma hitting the spacecraft is calculated
5. Converts flux from ECL50 frame to spacecraft pointing frame using the SSEDR
6. Converts the flux into the coordinate frame for the Faraday cups from Fran Bagenal's geometric derivation
7. Function returns this velocity for use in LABCUR/LDCUR or CUPINT/DCPINT.

Below is a more detailed version of the above bullet points.

The function first converts the cylindrical flow of plasma in the non-rotating system to a cylindrical flow in the rotating frame, System III, by subtracting a corotating frame from the velocity. Then the velocity is converted into Cartesian coordinates. Using the SSEDR the state vector, position and velocity, of both the plasma and the spacecraft are determined. Using the SpiceRotations.txt file, both of these state vectors are converted from System III state vectors to ECL50 state vectors.

The flux of the plasma hitting the spacecraft is then calculated. Using the SSEDR, the flux of plasma is converted from the ECL50 frame into the spacecraft pointing frame for the given time. The time given in the input is matched with the closest index value in the SSEDR, so the rotation matrix from the ECL50 frame to the spacecraft pointing frame is as accurate as possible.

Next, a rotation matrix from the spacecraft pointing frame to the X, Y, and Z axes for the specified Faraday cup is used. These rotation matrices for the individual cups are static in time since the instrument was connected to the spacecraft and always rotated with the spacecraft. The exact numbers for the rotation matrices were derived from measurements done in Voyager memos #20 and #29.

This function then returns the flow of the plasma in the Faraday cup's coordinates. For reference, the Z velocity is positive when the flow of the plasma is going into the cup directly. These flow speeds are then used for the calculation of the current below in LABCUR, LDCUR, CUPINT, and DCPINT.

This function was written at LASP in IDL.

Functions associated with VGR_CupVelocity

- eV_ThermalSpeed – Converts ion temperature in eV to a speed in km/s since responses use flowseeds of km/s
- closest – Finds closest value of array with single value for determining time stamp
- VPHI_PROFILE

B. CUPINT/DCPINT

This function simulates the response function for the Voyager PLS instrument to a cold, Mach > 5, Convected Maxwellian plasma. It uses primarily an analytical expression for the distribution function and utilizes the Simpson's rule for numerical integration. The integration limit is set to when the argument of the Maxwellian is large enough that its contribution is below the threshold of any measurable accuracy. CUPINT is for the analysis of Faraday cups A-C. DCPINT is for the analysis of the D cup. Exact details of the functional form and calculations can be seen in the code or in Alan Barnett's 1983 thesis in which the same calculation was used.

This function was written by Alan Barnett in FORTRAN and translated to IDL at LASP. Original notes from Alan Barnett can be found at the beginning of the code.

Functions associated with CUPINT/DCPINT

- AREAOV
- CLAROV
- TRANSP
- TRNSPD
- SHIFT

C. LABCUR/LDCUR

This function simulates the response function for the Voyager PLS instrument to a Convected Maxwellian plasma. Unlike CUPINT and DCPINT, it is not restricted to a cold, high Mach, plasma. It uses a numerical integration technique that utilizes the lookup tables provided by the mjs.out-2.txt file that account for the plasma being warm. LABCUR is for the analysis of Faraday cups A-C. LDCUR is for the analysis of the D cup. Exact details of the functional form and calculations can be seen in the code or in Alan Barnett's 1983 thesis in which the same calculation was used.

This function was written by MJSANL in FORTRAN and translated to IDL at LASP. Original notes from Alan Barnett can be found at the beginning of the code:

Functions associated with LABCUR/LDCUR

- TDERF
- TDERFC
- DPHI
- DPSI
- DPHIc
- PHI
- XBARP

D. DELAMERE_RATIOS

Reads in the structure used in the Viper program and if Delamere ratios are requested, the program accordingly obtains correct species rations for the given timestamp. Ratios span from 6.5 to 9 Rj and are locked in outside of 9 Rj

Inputs

Structure_In

Uncertainty_Flag

E. Miscellaneous Functions

- Euler – Returns rotation matrix from the three Euler angles.
- CupRotations – Used as a sanity check for the rotation from S/C bus to Faraday Cup coordinates using the above Euler function
- PLOT_RANGES_LOG – Reads in the array of current values and creates a plot range that is 4 orders of magnitude, with the top being above the maximum value of the array. Limits the low end to 100, even if it is not a full 4 orders of magnitude.
- Convert_Eddie_Ratios – Converts mixing ratios from Edward Nerney's analysis of Cassini UVIS ion composition. Cassini composition was not used in our analysis.
- CASSINI_RATIOS – Uses the output from the above program to get the relative densities of the various ion species. Again, this was not used for our analysis.
- VIPER_ST_JOIN – Combines two structures of original format. Code originally written by Rob Wilson of LASP and adopted for our use.
- CSV_COMBINE – Combines 2 of the VIPER csv files using the above function.
- VGR_CUP_FLOW_DIRECTIONS
- VGR_CUP_RESPONSE_PLOT

7. Output

A Plot is returned that plots the original data as well as the fit data. Individual colored lines denote different species as well as one line that is the sum of the current contributions from all species. This summed line is what is fit to the original data.

In the plot window, the time, flow speed in Jovicentric cylindrical coordinates, charge, mass, density, and temperature of the species are printed.

If the save output button is activated then the program will save a csv file and a jpg image. The csv file will indicate the original input into the change_parameters.csv file used earlier. The jpg will contain the image of the plots that appear in a window during the fitting routine. All 3 of these files will have the same naming scheme, only the extension varying and will be saved to the current directory that IDL is being run in. The naming scheme includes the UTC timestamp when the simulation finished, the mode* that the simulation is using, and the timestamp that is being fit. While lengthy, this will avoid accidentally overwriting any previous save files. An example filename is located below:

Date_Jun08_2015_2217_25_CUPINT_FIT_OF_064_1016_30

This fit was run on June 8, 2015 at 22:17:25 in UTC time (Hour, minute, second) and used the CUPINT and DCPINT responses to fit the data on day 64 of 1979 at 10:16:30. The timestamp at the end matches the timestamp in the data files (day62.txt, day63.txt, etc.).

*The filename will either include LABCUR or CUPINT. If it is LABCUR, the LDCUR will be run for the D Faraday cup. Similarly, if it is CUPINT, then DCPINT will be run for the D Faraday cup. This is not included in the save file to cut down on the length of the name.

8. Methods of Fitting

A. Overview: A Few Words

Through use of the csv file, it is possible to set many different sets of parameters for the fitting routine. In the jovian magnetosphere, the plasma behaves very differently in many different regions. In the very inner magnetosphere, within Io's orbit, roughly 6 R_J, the plasma is cold and corotates with the planet due to the magnetic field. The ion temperatures drop as low as 0.8 eV and the total electron density can be up to about 2000 per cubic centimeter. In this region, the plasma produced very well resolved peaks in Current vs. Energy/Charge. These peaks are easily distinguishable for different M/Q values and thus, apart from an ambiguity in O+ and S++, a unique fit can be determined for all 3 directions of the flow, the temperature of the plasma, and the densities of individual species.

Further out in the magnetosphere, the plasma becomes much more tenuous, densities dropping to a few per cc, or even lower, the plasma heats up as it radially diffuses outwards, and the plasma deviates from strict corotation. When the plasma heats up, the width of the peak increases, turning from a nice Maxwellian Gaussian looking distribution to a broad peak that fills up most of the Faraday cup. The different ions become very indistinguishable from one another as they begin to overlap in energy space. This therefore provides for a non-unique solution and attempts at fitting it result in failure when all parameters are set to vary.

For these regions, it is possible to use a chemistry calculation by Delamere (NEED CITATION) to get a ratio of what the densities of the various ion species should be in regards to one species, such as O+. This allows for the fitting routine to only fit the density of O+, effectively reducing 5 parameters for density into one proxy parameter for the total density. In some regions, this is enough to get a unique fit for the total density, the temperature of the ion species, and for the azimuthal direction of the flow speeds. Unfortunately, there is not enough resolution in the instrument to get a strong signal from flow speeds in the radial or vertical direction (r and z in cylindrical coordinates) in the jovicentric coordinate frame; this is due to the instrument being pointed mainly along the azimuthal flow direction of the plasma.

In some regions, the total peak is not very well defined and the code will again fail in fitting. In these regions it is necessary to also constrain the flow speed of the plasma. This can be constrained to corotational speeds if it is close to the planet, but since corotation eventually begins to break down, a different analytical equation for the flow speed was derived. Using values of the azimuthal flow from the fits, a hyperbolic tangent profile was established. This profile is linear for low values of R_J, fitting the corotational approximation. However, at large values, it asymptotes to a specific value and therefore acts as a simple model for the flow speed. Based off of all of the fits to the Voyager PLS data, this hyperbolic tangent model is not an accurate model the middle magnetosphere, but it does work for both the inner and outer. It appears that in the middle region, of about 10-20 R_J, the flow speed deviates from corotation, dipping down a little before returning to close to corotation values and then asymptoting in the range of 200-250 km/s.

Further out in the magnetosphere, amidst the hot tenuous plasma, there are small pockets of cold and dense plasma called cold blobs. In these regions, the peak resolution is similar to that of the cold torus region and it is therefore possible to get much more unique fits. These spectra use a combination of Delamere Ratios and fitting for density to obtain the most robust analysis of the plasma properties in this region.

B. Selection of Channel Values

For the M-Mode spectra, each Faraday cup had 128 values of energy/charge along the x-axis, ranging from 10-5950 Volts. Based off of the speed of the plasma hitting the spacecraft, there would not be signal in every single channel number for every cup. Where there were large and defined peaks, it was simple to determine the range in channel numbers that should be fit. Where the signal was not at a large value, it would drop orders of magnitude lower and be noise that could be ignored so those channels could be ignored for the fit.

In the warmer region, it became more difficult. As a general guideline, for our analysis, we fit the regions where it appeared that there was a high signal to noise, often cutting off the low end of the lowest M/Q ion species and the high end of the highest M/Q ion species so that the code did not attempt to fit large amounts of noise as opposed to actual signal.

C. Method I: All Parameters Vary

-Set Vphi and all densities and temperatures to vary independently.

D. Method II: Delamere Ratios

-Tie the densities strictly to Delamere Composition

E. Method III: Interpolate between Delamere and Cold Torus Measurements

-Use a composition determined by an interpolation between Cold torus measurements and delamere composition at 6 Rj.

F. Method IV: Fixed Speed with Delamere Composition

-Densities are tied to Delamere species while the V_{phi} flow speed is tied to strict corotation or a fit to corotation determined during this analysis.

G. Method V: Modified Delamere Composition

-In various regions of the magnetosphere plasma properties seemed to deviate from Delamere composition. In these locations, users are able to modify the values of Delamere ratio to more accurately fit the data.

G. Inclusion of a Hot Ion Species: A Proxy for Kappa Distributions

In reality, the plasma does not behave with a cold convected Maxwellian distribution. Therefore, that assumption in the model will fail to model the higher energies where warm plasma tails can dominate. In most space plasmas, Kappa distributions with a power law tail of the distribution seem to be prevalent. As it stands, VIPER is not written to include Kappa distributions. In future work, it will be possible to include the Voyager PLS measurement with the LECP data to include even higher energy ranges and fit a Kappa distribution.

For our analysis, a single hot species of O+ was included. This acts as a proxy for Kappa distributions of multiple ion species, only serving the purpose of filling in the higher energy channels with a reasonable constant so that the fitting process can do the best job possible. The values for the density and temperatures of this hot ion species do not reflect real quantifiable data, but allow for data analysis of the regions where we expect there to be data.

9. SPICE

Note: This section is for reference. Access to JPL's SPICE database is not needed due to the presence of the SpiceRotation text files; following these steps will reproduce those files if needed.

In our analysis we utilize the SPICE database to derive a rotation matrix between state vectors from System III to ECL50. The SSEDR file has a transformation matrix between ECL50 and the spacecraft pointing and geometrical analysis provides the transformation into the frame of the Faraday cup. It makes sense to specify the plasma flow in a coordinate frame that is centered around the spin axis of Jupiter. Therefore, the plasma flow is easily converted to the body fixed frame of System III, a flow speed of 0 km/s if the plasma is purely corotating, and then the SPICE database is used to derive a rotation matrix between System III and ECL50 so that the data can be analyzed.

The SPICE commands and procedures are included in the section below although they are not necessary for this program to run. As long as the SpiceRotations.txt file is included in the IDL workspace, it is not necessary to use any kernels. The SpiceRotations.txt file includes data for all, and only, the timestamps present in the SSEDR file. In the instance that there is data for more timestamps than present in the SSEDR then the SPICE commands will need to be run to get an accurate transformation matrix between the System III and ECL50 coordinate frames for those desired times.

In IDL, the following SPICE command was used to create the rotation matrices between System III and ECL50 for the n points in the SSEDR file.

```
Rotation = Make_array(n)
FOR i = 0, n-1 DO BEGIN
cspice_sxform, ‘IAU_Jupiter’, ‘ECLIPB1950’, et(i), xform
Rotation(i) = xform
ENDFOR
```

‘IAU_Jupiter’ is the body-fixed frame (System III) and ECLIPB1950 is the ECL50 frame. An input of ephemeris time is required through the et parameter. The output is the 6x6 rotation matrix between the state vectors of the two frames. Unfortunately, the cspice_sxform command did not work with an array of ephemeris times and only seems to work with a single ephemeris time input so it uses a FOR loop to calculate values for all of the times.

For calculating the ephemeris time it was helpful to use the built in SPICE command to convert a string time in UTC to ET. The below code will be for the SSEDR file, but a similar format can be used for other formats with a similar structure. Documentation about the function cspice_str2et can be found in the SPICE toolkits. Data is the array that contains all of the SSEDR information.

```
UTC1 = Make_array(n,/STRING)
FOR i = 0, n-1 DO BEGIN
Year = Data(0,i)
DOY = Data(1,i)
Hour = Data(2,i)
Minute = Data(3,i)
Second = Data(4,i)
Millisecond = Data(5,i)

; Format for entry should be yyyy-dddTHH:MM:SS.FFF
UTC1(i) = STRING(Year, ‘-’, DOY, ‘T’, Hour, ‘:’, Minute, ‘:’, Second, ‘.’, Millisecond, $
                        FORMAT = ‘(I04,A,I03,A,I02,A,I02,A,I02,A,I03)’
ENDFOR

CSPICE_STR2ET, UTC1, ET
```

The above code will return an array of the same dimensions as UTC1 and will provide the ephemeris time from the data if there is not another way to easily calculate the ephemeris times from the SSEDR library.

A. Necessary SPICE Kernels

The following list shows all of the SPICE kernels that were used for the calculation and cross reference of the SSEDR. These SPICE kernels include more data than is necessary for just the calculation of the SpiceRotations.txt file but offer trajectory information for the various coordinate frames to test the validity of the data in the SSEDR file.

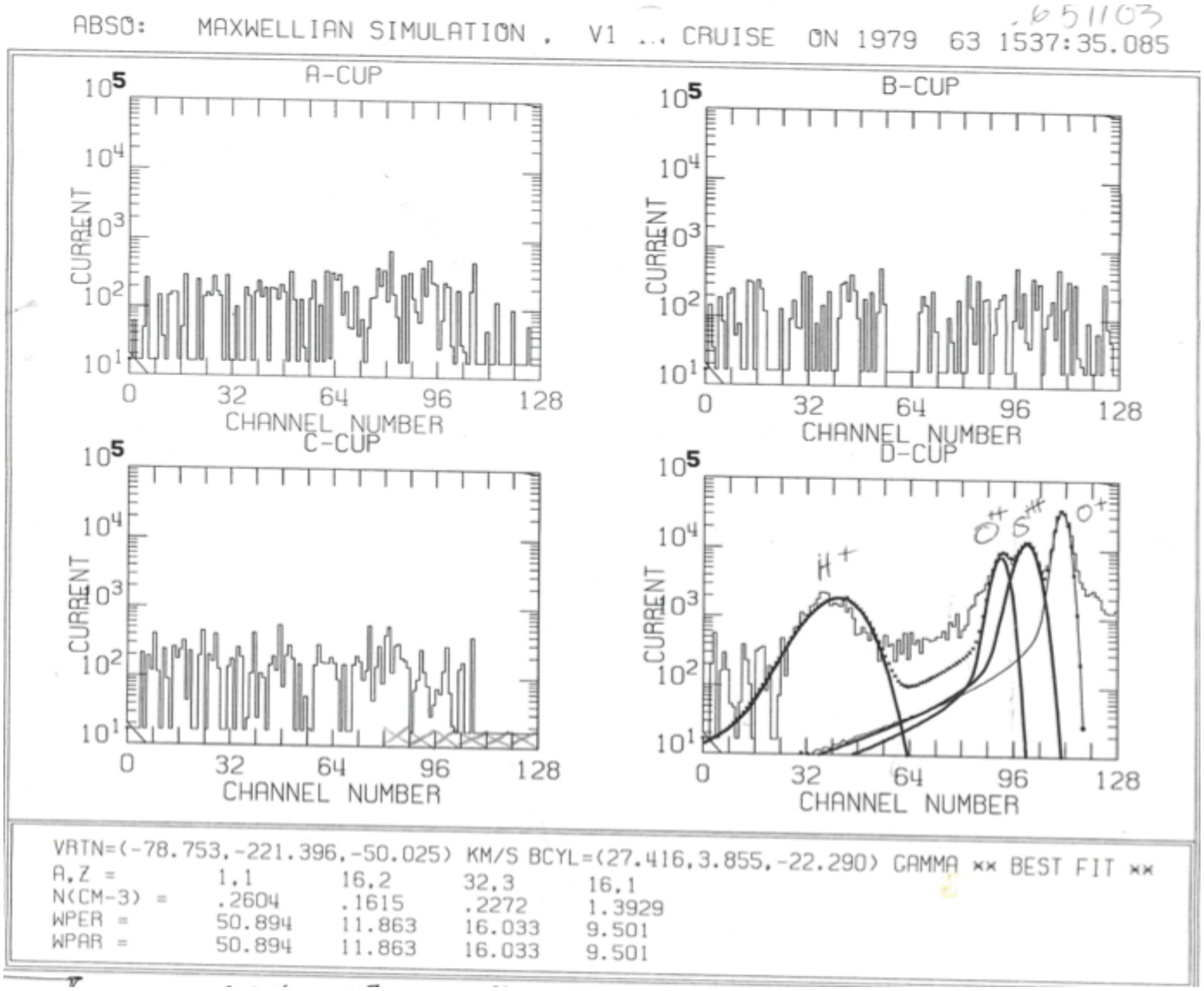
- naif0010.tls
- pck00010.tpc
- vg1_jup_version1_type1_iss_sedr.bx
- VG1_V02.tf.txt

- vg100019.tsc.txt
- vgr1_jup230-2.bsp
- vgr1_super.bc
- Voyager_1.a54206u_V0.2_merged.bsp
- voyager_1.ST+1991_a54418u.merged.bsp
- voyager_2.ST+1992_m05208u.merged.bsp

10. Method Comparisons

This section contains comparisons to the original analysis of data by MJSANL with the code from LASP to verify that it works.

CASE 1 - Day 63 15:37:35
Original
Note: This day has a slightly different velocity flow in the original plot using VRTN, so the plots from the re-analysis will not match exactly.

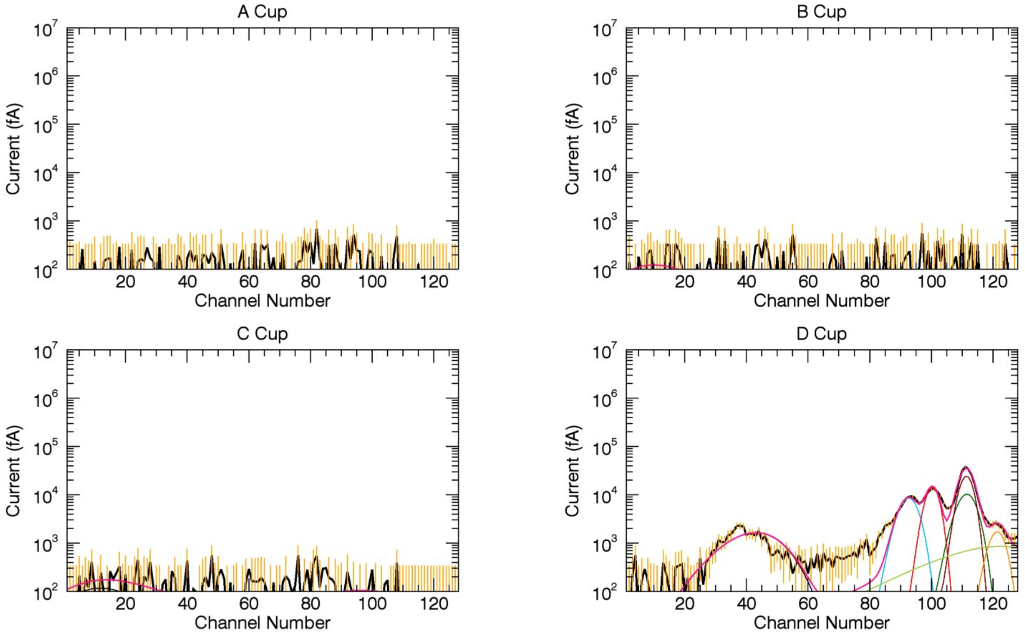


```
ABSO:   MAXWELLIAN SIMULATION .  V1 ... CRUISE  ON 1979  63 1537:35.085
```

```
VRTN=(-78.753,-221.396,-50.025) KM/S BCYL=(27.416,3.855,-22.290) GAMMA ** BEST FIT **
A,Z =          1,1       16,2      32,3      16,1
N(CM-3) =      .2604     .1615     .2272     1.3929
WPER =         50.894    11.863    16.033    9.501
WPAR =         50.894    11.863    16.033    9.501
```

CUPINT Response

Mode:CUPINT/DCPINT Year: 1979 Date (DOY, Hour, Min, Second): 63 15 37 35 RJ: 19.762

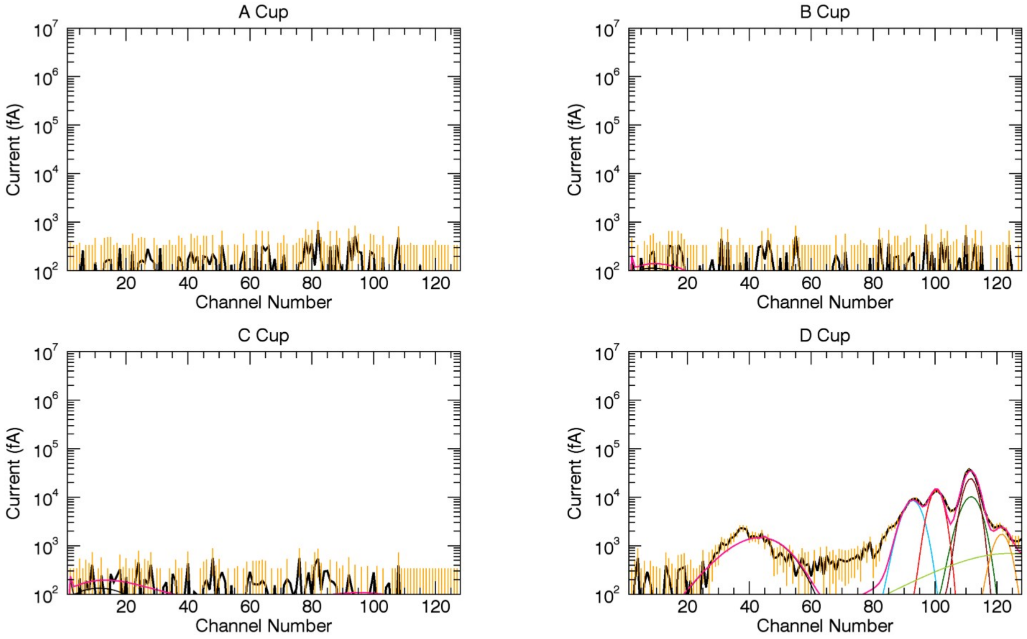


Cyl Vel(V_r, V_φ, V_z): 0.00 203.97 0.00

A (amu), Z (q):	16, 1	16, 2	32, 3	32, 2	1, 1	16, 1	23, 1
n (cm^{-3}):	0.43	0.18	0.14	0.36	0.21	0.20	0.06
T (eV):	18.30	18.30	18.30	18.30	18.30	900.00	18.30

LABCUR Response

Mode:LABCUR/LDCUR Year: 1979 Date (DOY, Hour, Min, Second): 63 15 37 35 RJ: 19.762

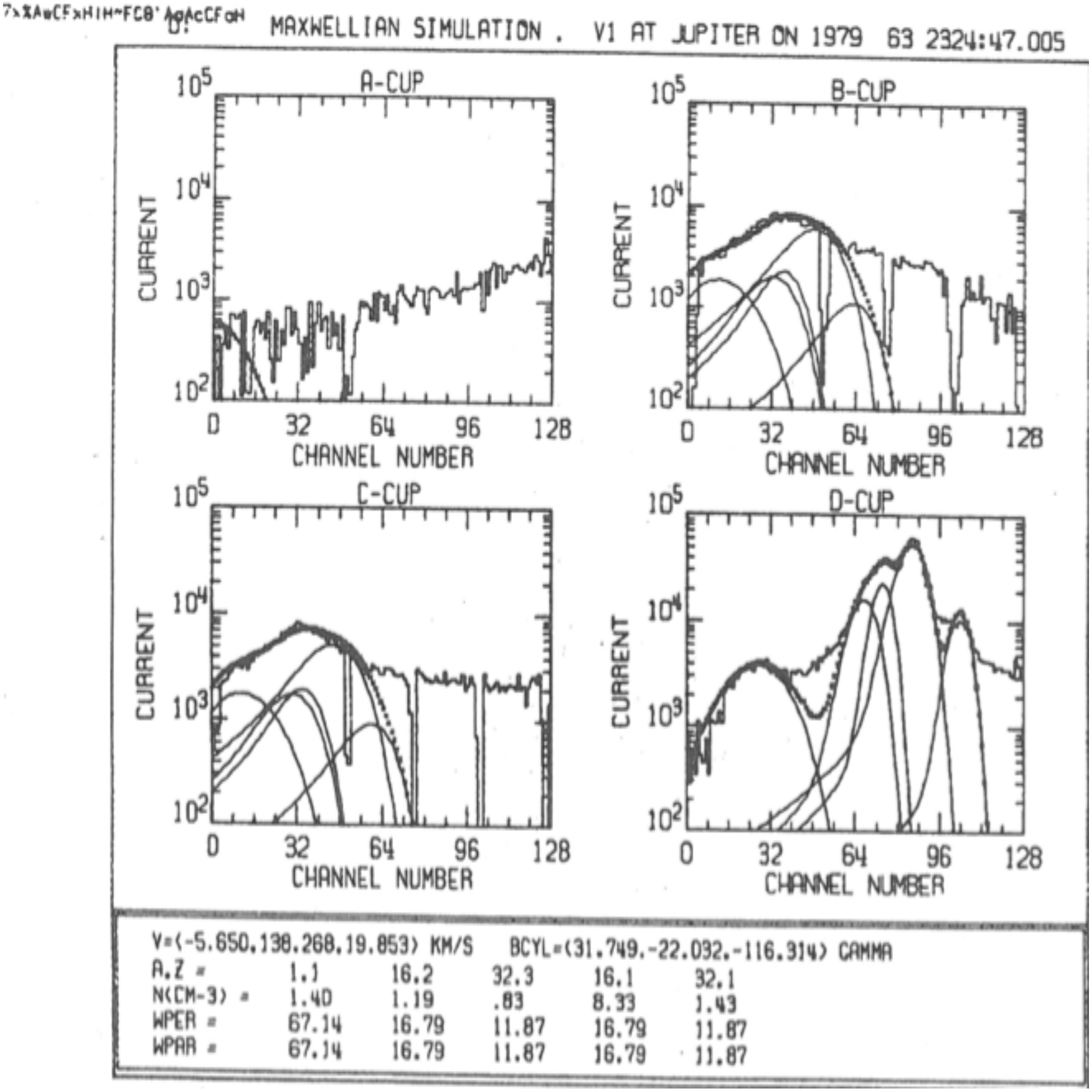


Cyl Vel(V_r, V_φ, V_z): 0.00 203.97 0.00

A (amu), Z (q):	16, 1	16, 2	32, 3	32, 2	1, 1	16, 1	23, 1
n (cm^{-3}):	0.43	0.18	0.14	0.36	0.21	0.20	0.06
T (eV):	18.30	18.30	18.30	18.30	18.30	900.00	18.30

CASE 2 - Day 63 23:24:47

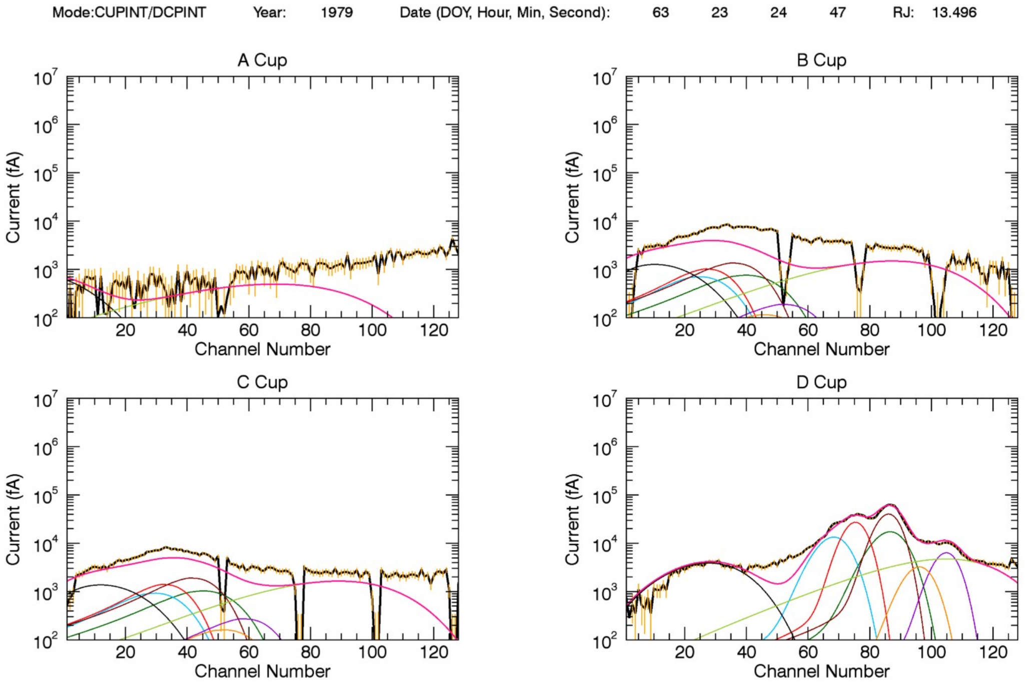
Original



```
7x%AwCFxHIH~FCB'AgAcCFoH D:  MAXWELLIAN SIMULATION .  V1 AT JUPITER ON 1979  63 2324:47.005
```

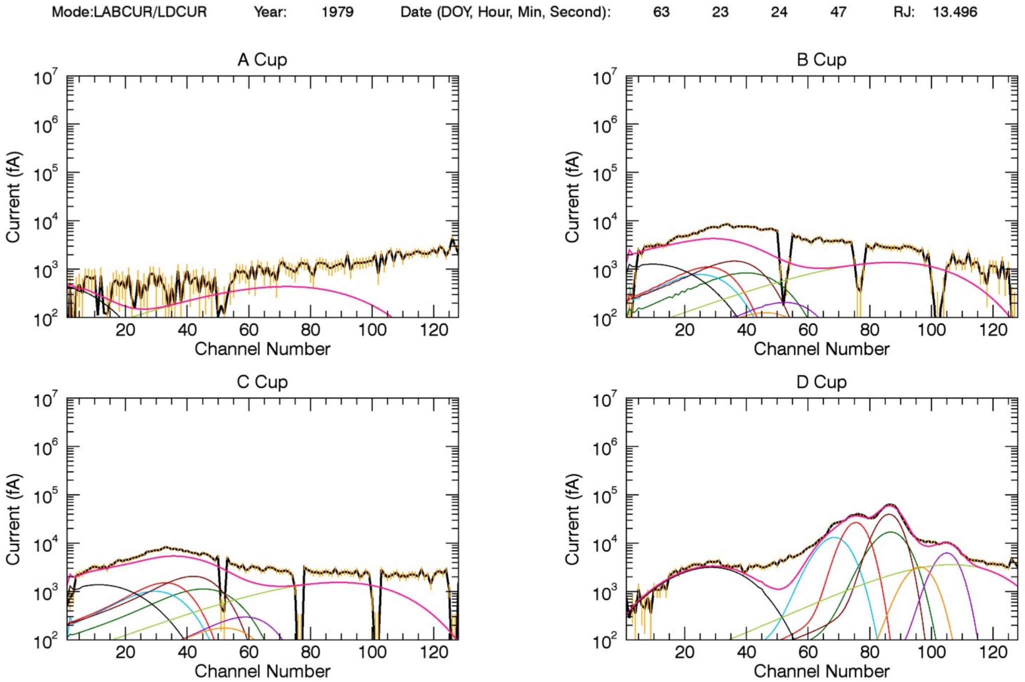
```
V=(-5.650,138.268,19.853) KM/S    BCYL=(31.749,-22.032,-116.314) GAMMA
A,Z =          1.1       16.2       32.3       16.1       32.1
N(CM-3) =      1.40      1.19       .83        8.33       1.43
WPER =         67.14     16.79      11.87      16.79      11.87
WPAR =         67.14     16.79      11.87      16.79      11.87
```

CUPINT Response



Cyl Vel(V_r, V_φ, V_z):		0.00	134.80	0.00				
A (amu), Z (q):	16, 1	16, 2	32, 3	32, 2	32, 1	1, 1	16, 1	23, 1
n (cm^{-3}):	2.24	0.84	0.83	1.87	0.60	1.12	2.20	0.36
T (eV):	24.91	24.91	24.91	24.91	24.91	24.91	750.00	24.91

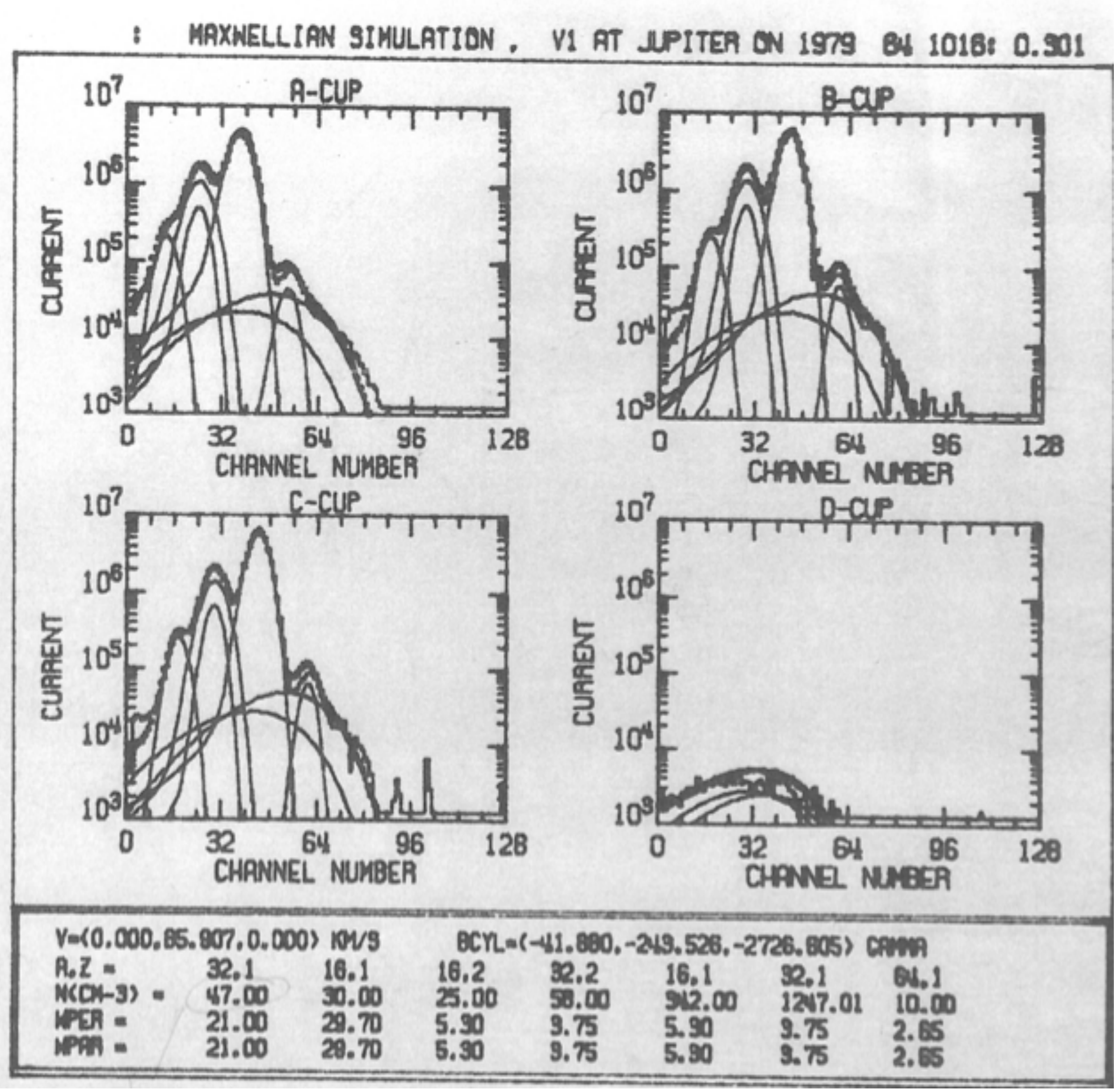
LABCUR Response



Cyl Vel(V_r, V_φ, V_z):		0.00	134.80	0.00				
A (amu), Z (q):	16, 1	16, 2	32, 3	32, 2	32, 1	1, 1	16, 1	23, 1
n (cm^{-3}):	2.24	0.84	0.83	1.87	0.60	1.12	2.20	0.36
T (eV):	24.91	24.91	24.91	24.91	24.91	24.91	750.00	24.91

CASE 3 - Day 64 10:16

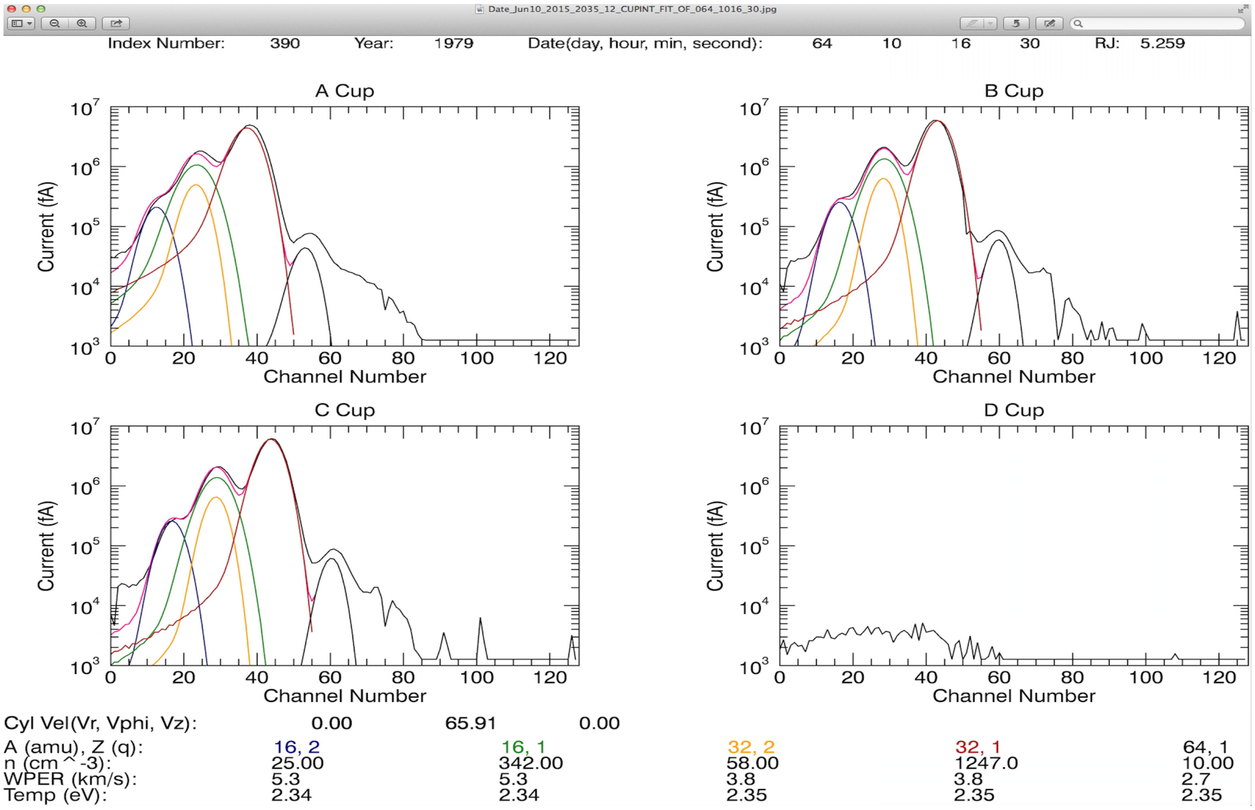
Original



```
:   MAXWELLIAN SIMULATION ,  V1 AT JUPITER ON 1979  64 1016: 0.301
```

```
V=(0.000,85.807,0.000) KM/S          BCYL=(-41.880,-249.526,-2726.805) GAMMA
A,Z =          32.1      16.1      16.2      32.2      16.1      32.1      64.1
N(CM-3) =      47.00     30.00     25.00     58.00     942.00    1247.01   10.00
WPER =         21.00     29.70     5.30      9.75      5.90      9.75      2.65
WPAR =         21.00     29.70     5.30      9.75      5.90      9.75      2.65
```

CUPINT Response

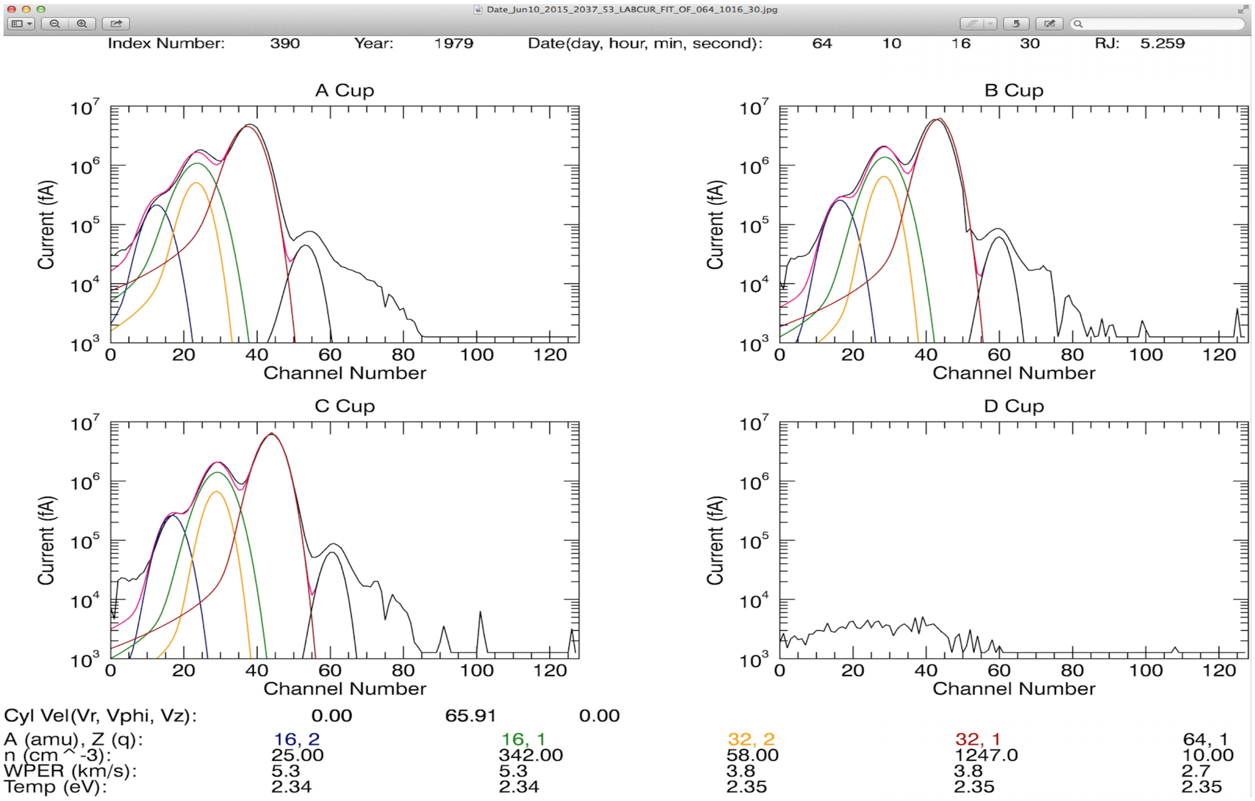


Index Number: 390 Year: 1979 Date(day, hour, min, second): 64 10 16 30 RJ: 5.259

Cyl Vel(Vr, Vphi, Vz): 0.00 65.91 0.00

A (amu), Z (q):	16, 2	16, 1	32, 2	32, 1	64, 1
n (cm ^-3):	25.00	342.00	58.00	1247.0	10.00
WPER (km/s):	5.3	5.3	3.8	3.8	2.7
Temp (eV):	2.34	2.34	2.35	2.35	2.35

LABCUR Response



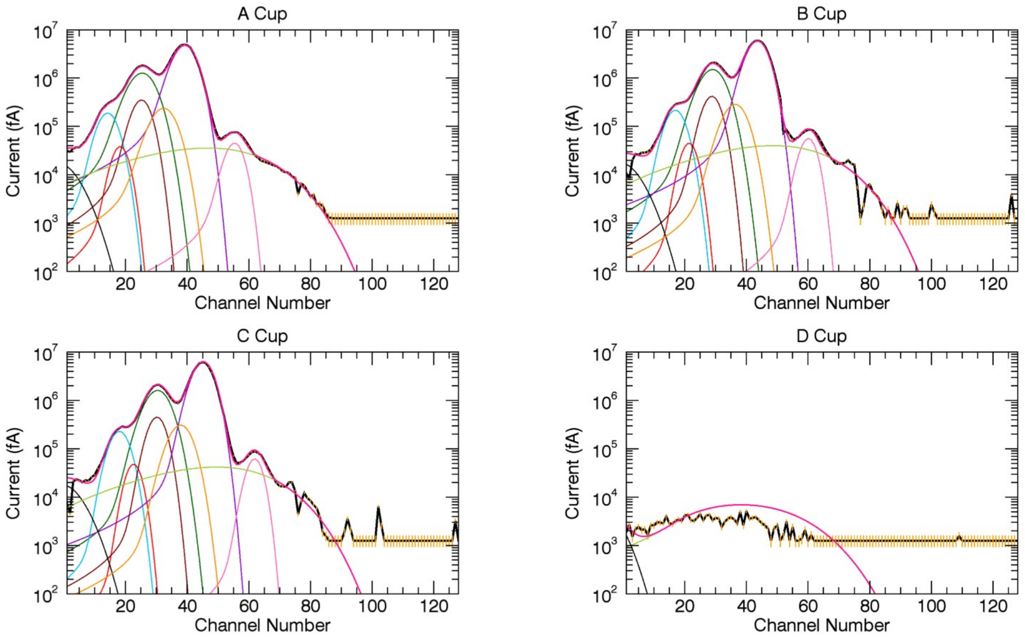
Index Number: 390 Year: 1979 Date(day, hour, min, second): 64 10 16 30 RJ: 5.259

Cyl Vel(Vr, Vphi, Vz): 0.00 65.91 0.00

A (amu), Z (q):	16, 2	16, 1	32, 2	32, 1	64, 1
n (cm ^-3):	25.00	342.00	58.00	1247.0	10.00
WPER (km/s):	5.3	5.3	3.8	3.8	2.7
Temp (eV):	2.34	2.34	2.35	2.35	2.35

CUPINT Response – With added hot species

Mode:CUPINT/DCPINT Year: 1979 Date (DOY, Hour, Min, Second): 64 10 16 0 RJ: 5.269

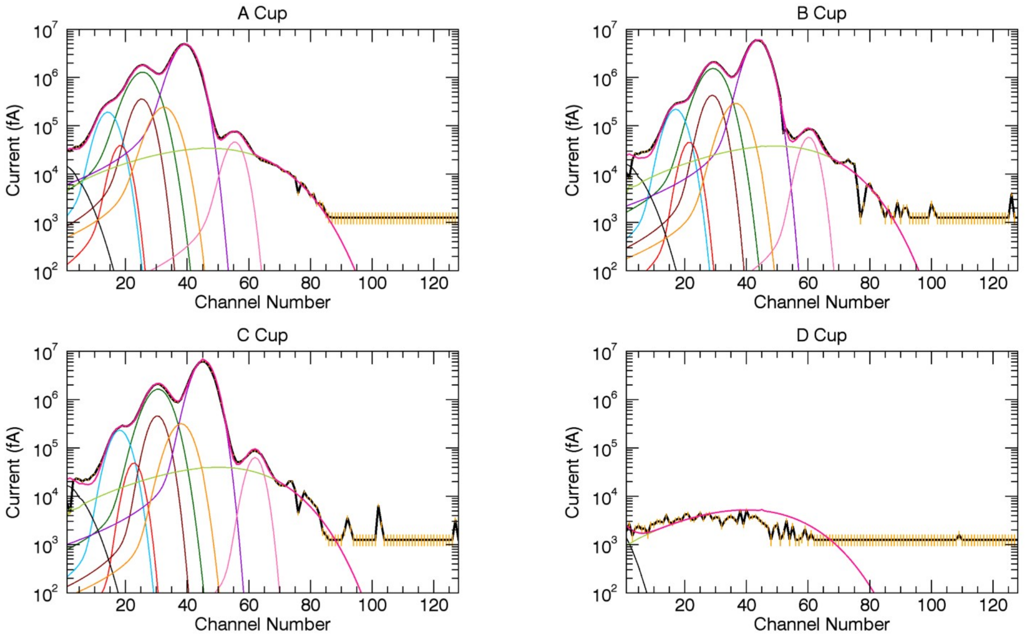


Cyl Vel(V_r, V_φ, V_z): -1.60 66.09 -1.58

A (amu), Z (q):	16, 1	16, 2	32, 3	32, 2	32, 1	1, 1	64, 1	16, 1	23, 1
n (cm^{-3}):	376.63	20.93	2.35	37.66	1226.70	4.30	9.34	50.00	66.41
T (eV):	2.17	2.17	2.17	2.17	2.17	4.39	2.17	127.00	2.17

LABCUR Response – With added hot species

Mode:LABCUR/LDCUR Year: 1979 Date (DOY, Hour, Min, Second): 64 10 16 0 RJ: 5.269



Cyl Vel(V_r, V_φ, V_z): -1.60 66.09 -1.58

A (amu), Z (q):	16, 1	16, 2	32, 3	32, 2	32, 1	1, 1	64, 1	16, 1	23, 1
n (cm^{-3}):	376.63	20.93	2.35	37.66	1226.70	4.30	9.34	50.00	66.41
T (eV):	2.17	2.17	2.17	2.17	2.17	4.39	2.17	127.00	2.17

Acknowledgements

We are very grateful to our colleagues at MIT John Belcher and John Richardson, plus Ralph McNutt (JHU APL) and Alan Barnett, for help in resurrecting the Voyager PLS analysis code. We also thank Rob Wilson, Frank Crary and Amber Hall at LASP.

References

For full list of references on Voyager PLS data at Jupiter see

http://lasp.colorado.edu/home/mop/bibliographies/references/

Delamere, P., Bagenal, F., & Steffl, A. (2005). Radial variations in the Io plasma torus during the Cassini era. *J. Geophys. Res., 110,* A12223.

Markwardt, C. B. (2009), Non-linear least squares fitting in IDL with MPFIT, in Proceedings of Astronomical Data Analysis Software and Systems XVIII, Quebec, Canada, ASP Conference Series, vol. 411, D. Bohlender, P. Dowler, and D. Durand (eds), pp. 251–254, Astron. Soc. of the Pacific, San Francisco, CA

Wilson, R. J. (2015) Error analysis for numerical estimates of space plasma parameters, Earth Space Sci., 2, 201–222, doi:10.1002/2014EA000090.