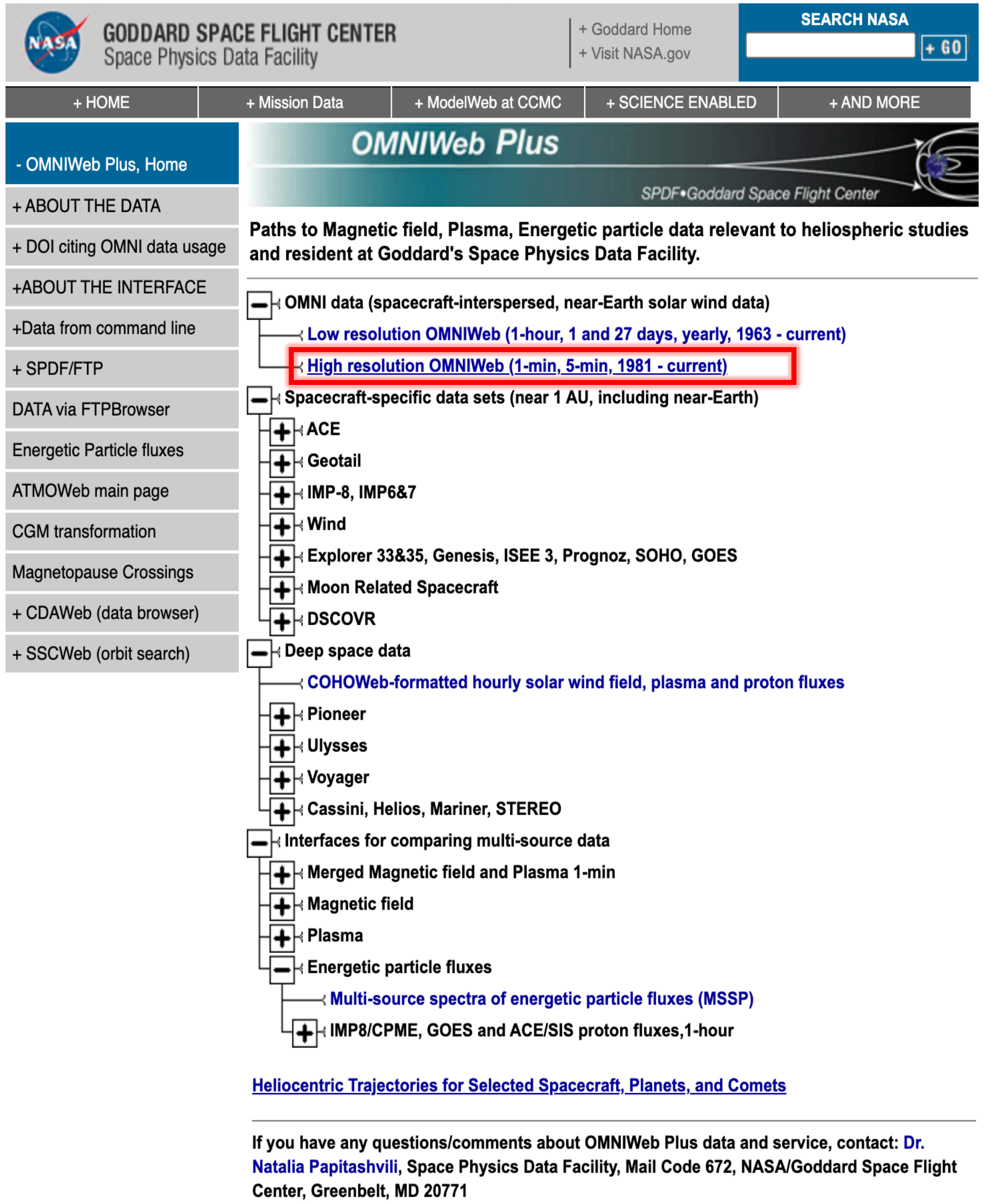
Read ASCII File from NASA Space Physics Data Facility -- OMNI Data

Chen Wu, Aaron Ridley/University of Michigan

https://omniweb.gsfc.nasa.gov/

OMNI Data

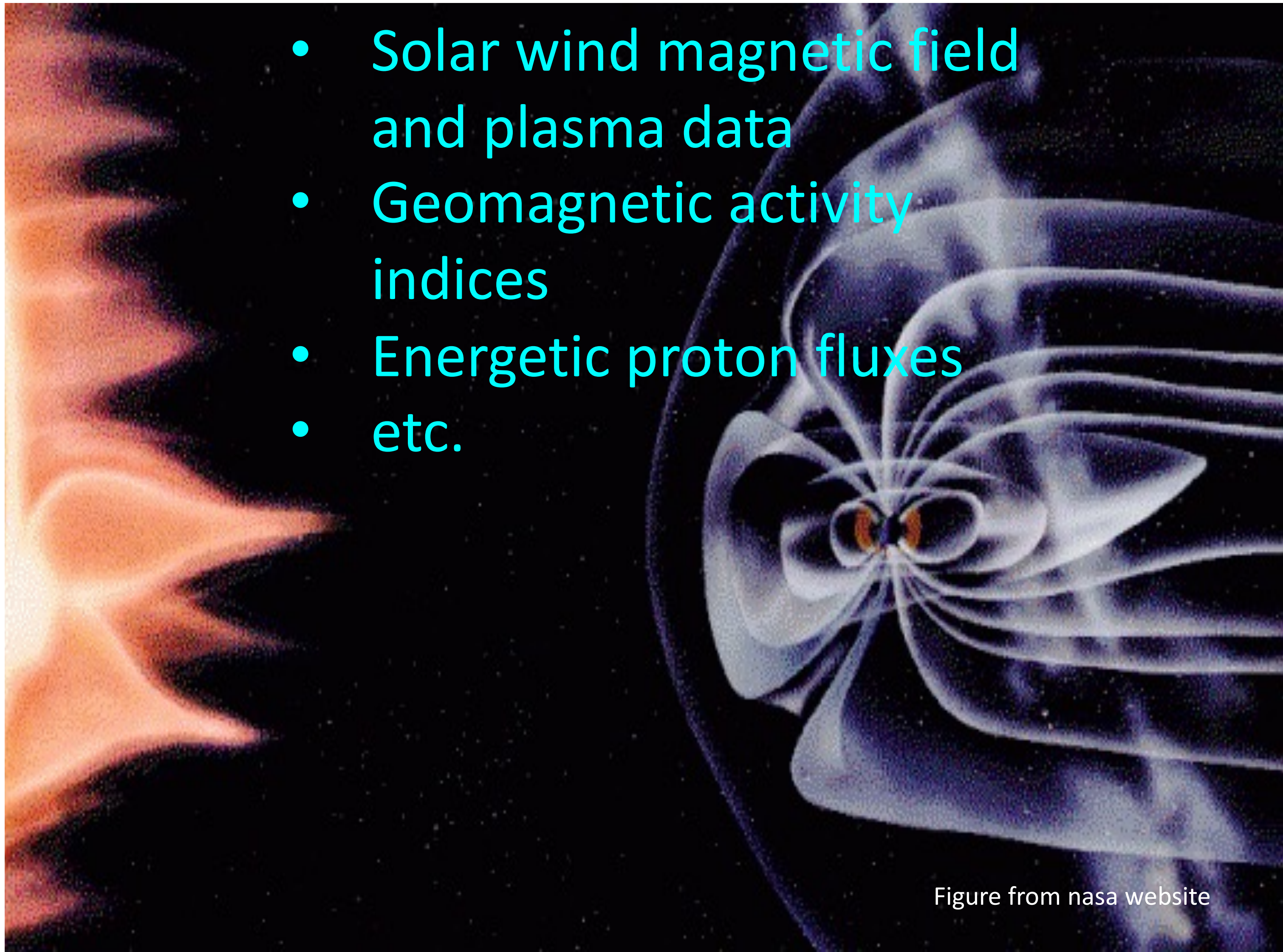


Paths to Magnetic field, Plasma, Energetic particle data relevant to heliospheric studies and resident at Goddard's Space Physics Data Facility.

- OMNI data (spacecraft-interspersed, near-Earth solar wind data)
  - Low resolution OMNIWeb (1-hour, 1 and 27 days, yearly, 1963 - current)
  - High resolution OMNIWeb (1-min, 5-min, 1981 - current)
- Spacecraft-specific data sets (near 1 AU, including near-Earth)
  - ACE
  - Geotail
  - IMP-8, IMP6&7
  - Wind
  - Explorer 33&35, Genesis, ISEE 3, Prognoz, SOHO, GOES
  - Moon Related Spacecraft
  - DSCOVR
- Deep space data
  - COHOWeb-formatted hourly solar wind field, plasma and proton fluxes
  - Pioneer
  - Ulysses
  - Voyager
  - Cassini, Helios, Mariner, STEREO
- Interfaces for comparing multi-source data
  - Merged Magnetic field and Plasma 1-min
  - Magnetic field
  - Plasma
  - Energetic particle fluxes
    - Multi-source spectra of energetic particle fluxes (MSSP)
    - IMP8/CPME, GOES and ACE/SIS proton fluxes,1-hour

Heliocentric Trajectories for Selected Spacecraft, Planets, and Comets

If you have any questions/comments about OMNIWeb Plus data and service, contact: Dr. Natalia Papitashvili, Space Physics Data Facility, Mail Code 672, NASA/Goddard Space Flight Center, Greenbelt, MD 20771



- Solar wind magnetic field and plasma data
- Geomagnetic activity indices
- Energetic proton fluxes
- etc.

Figure from nasa website

open()

- open file and return a file object
- open(filename, mode, encoding=None)

❑ filename: gives the pathname of the file to be opened.

Example 1:

```
f = open("omni_test.lst”)
f.close()
```

Example 2:

```
with open("omni_test.lst") as f:
   # perform file operations
```

1. pathname of file
- data and code in the same directory
  "omni_min_test.lst”,
  ”./omni_min_test.lst”,
- different directory:
  ”../data/omni_min_test.lst”,
  “/Users/chenwum/Documents/summer_school/data/omni_min_test.lst”

2. close the opened file

open()

- open file and return a corresponding file object
- open(filename, mode, encoding=None)

❑ mode: an optional string that specifies the mode in which the file is opened.

- ‘r’, ‘w’, ‘x’, ’a’, ‘b’, ‘t’, ‘+’

Character	Meaning
'r'	open for reading (default)
'w'	open for writing,
'x'	open for exclusive creation, failing if the file already exists
'a'	open for writing, appending to the end of file if it exists
'b'	binary mode
't'	text mode (default)
'+'	open for updating (reading and writing)

❑ encoding: name of the encoding used to decode or encode the file. The default encoding is platform dependent.

- e.g, encoding="utf-8”

```
import locale
locale.getpreferredencoding(False)
```

More details:
https://docs.python.org/3/library/functions.html#open

read each line of the file

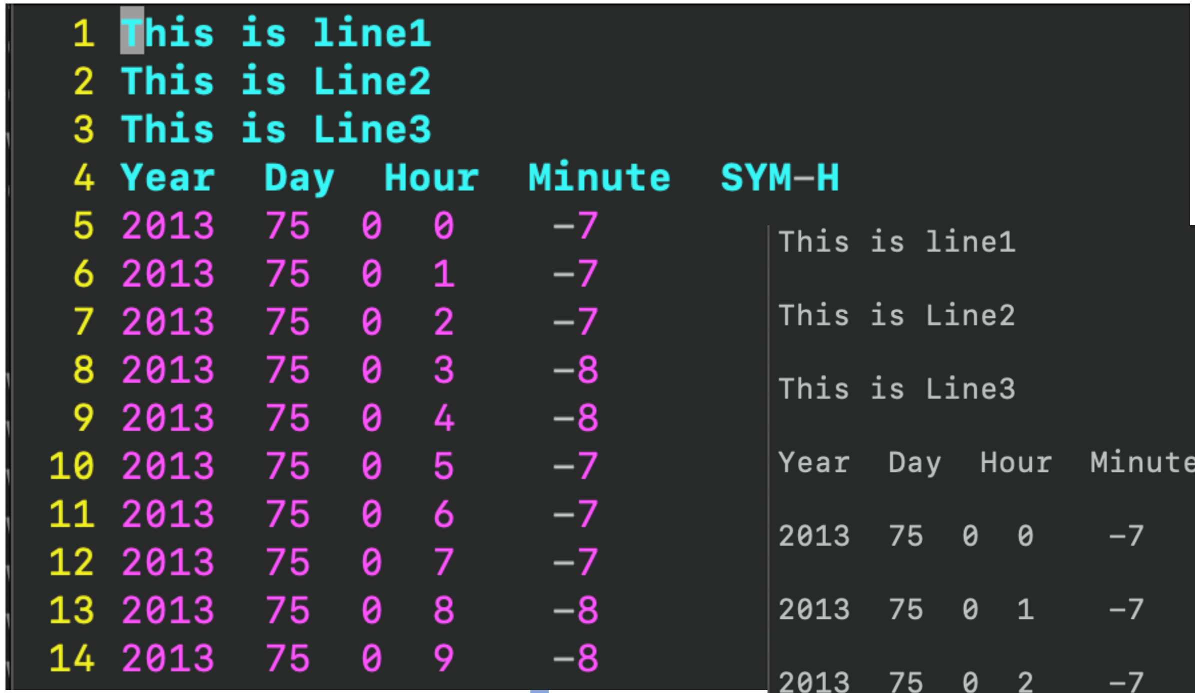
```
Example 1:
with open("omni_test.lst") as f:
    f.readline()            # read one line
    f.readline()            # read the second line

    nLines = 3
    # or read more lines with for loop
    for iLine in range(nLines):
        tmp = f.readline()
```

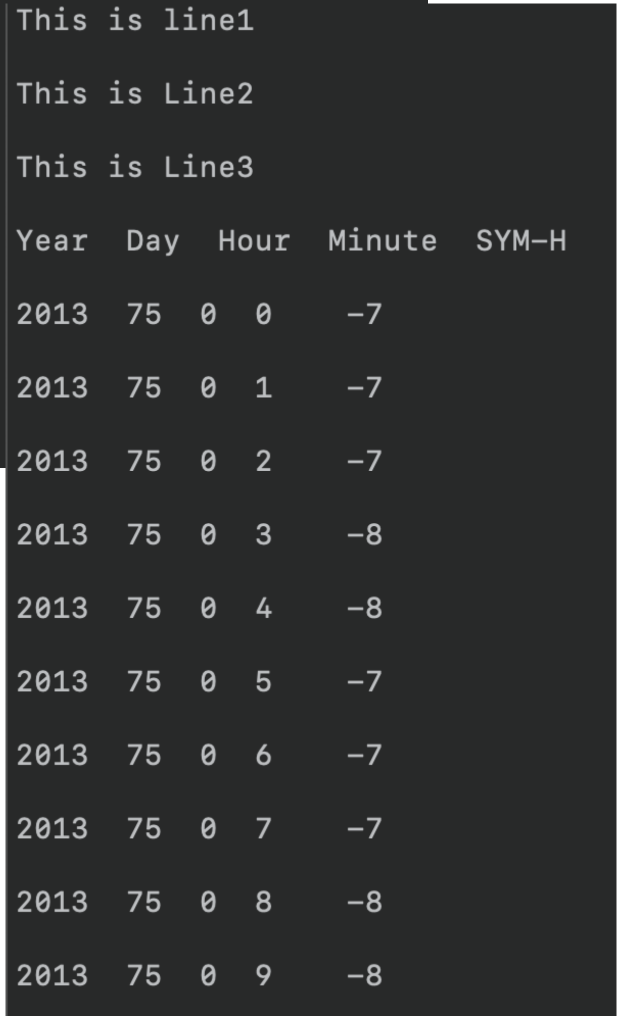
```
Example 2:
with open("omni_test.lst") as f:
    for line in f:
        print(line)
```

Output is string

read each line of the file



```
 1 This is line1
 2 This is Line2
 3 This is Line3
 4 Year  Day  Hour  Minute  SYM-H
 5 2013  75  0  0    -7
 6 2013  75  0  1    -7
 7 2013  75  0  2    -7
 8 2013  75  0  3    -8
 9 2013  75  0  4    -8
10 2013  75  0  5    -7
11 2013  75  0  6    -7
12 2013  75  0  7    -7
13 2013  75  0  8    -8
14 2013  75  0  9    -8
```



```
This is line1

This is Line2

This is Line3

Year  Day  Hour  Minute  SYM-H

2013  75  0  0    -7

2013  75  0  1    -7

2013  75  0  2    -7

2013  75  0  3    -8

2013  75  0  4    -8

2013  75  0  5    -7

2013  75  0  6    -7

2013  75  0  7    -7

2013  75  0  8    -8

2013  75  0  9    -8
```

Output is string

1. For **description** of the file (lines 1-3)
- Skip

2. For **header** of the file (line 4)
- variable names

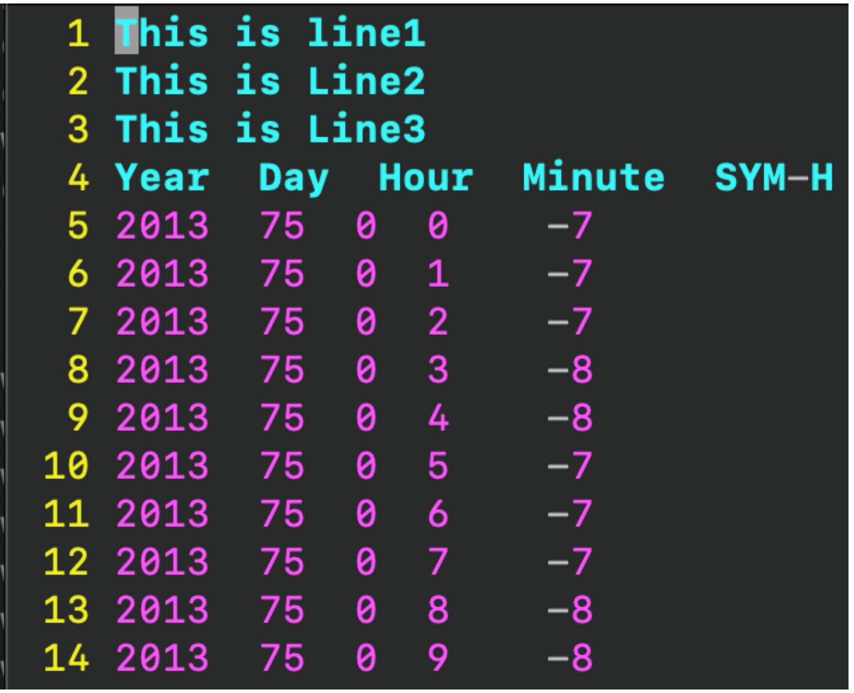
3. For **data** of the file
- Convert to numerical values

Split string

Example 1:
```
with open("omni_test.lst") as f:
    line = f.readline()          # read one line
    tmp = line.split()
```

Example 2:
```
with open("omni_test.lst") as f:
    for line in f:
        tmp = line.split()
```

Output is substring array



```
This is line1
This is Line2
This is Line3
Year  Day  Hour  Minute  SYM-H
2013  75   0   0      -7
2013  75   0   1      -7
2013  75   0   2      -7
2013  75   0   3      -8
2013  75   0   4      -8
2013  75   0   5      -7
2013  75   0   6      -7
2013  75   0   7      -7
2013  75   0   8      -8
2013  75   0   9      -8
```

If convert a string to a float data, use float()!

```
with open("omni_test.lst") as f:
    # skip 3 lines doing nothing
    for iLine in range(nLines):
        tmp = f.readline()

    # line 4: read in variables line and
    # convert to variable names
    header = f.readline()
    vars = header.split()

    # read in data line, convert to numerical values
    for line in f:
        tmp = line.split()
        year.append(int(tmp[0]))    # convert a string to an integer
        day.append(int(tmp[1]))
        hour.append(int(tmp[2]))
        minute.append(int(tmp[3]))
        symh.append(int(tmp[-1]))
```

```
with open("omni_min_def_5rROITB6pD.lst") as f:
    year = []
    day = []
    hour = []
    minute = []
    symh = []

    for line in f:
        tmp = line.split()
        year.append(int(tmp[0]))      # convert a string to an integer
        day.append(int(tmp[1]))
        hour.append(int(tmp[2]))
        minute.append(int(tmp[3]))
        symh.append(int(tmp[-1]))
```

# Create empty lists for variables that we want to extract

# add each numerical value to the corresponding list in each step of the loop

However, we can try use datetime and dictionary now!

https://docs.python.org/3/library/datetime.html

datetime

- supplies classes for manipulating dates and times.

Some important objects:

datetime.datetime(*year*, *month*, *day*, *hour=0*, *minute=0*, *second=0*)
datetime.timedelta(*days=0*, *seconds=0*, *microseconds=0*, *milliseconds=0*, *mi
nutes=0*, *hours=0*, *weeks=0*)

```
Example:
import datetime as dt
# Create with Year, Month, Day, Hour, Minute, Second
time1 = dt.datetime(2013,1,3, 10,12,30)
# Create with Year, Day of Year, Hour, Minute, Second
time2 = dt.datetime(2013,1,1,10,12,30) + dt.timedelta(days = 2)
print(time1.date())

lp = time1 == time2       # output is true
lp = time1>dt.datetime(2013,1,5, 0,0,0)  #output is false
```

```
with open("omni_test.lst") as f:
   # skip 3 lines doing nothing
   for iLine in range(nLines):
      tmp = f.readline()

    # line 4: read in variables line and
    header = f.readline()
    vars = header.split()

    data_dict = {“time”:[]}              # create dictionary
    for var in vars:
        data_dict[var] = []              # add variables to dictionary:

   for line in f:
      tmp = line.split()

       # create datetime with year, day, hour, minute
       time0 =dt.datetime(int(tmp[0]),1,1,int(tmp[2]),int(tmp[3]),0)\
                    + dt.timedelta(days=int(tmp[1])-1)
       data_dict[“time”].append(time0)
       for iVar, var in enumerate(vars):
          data_dict[var].append(int(tmp[iVar]))
```