Computational notebooks for accessing and analyzing data at the Space Physics Data Facility

2023 Data, Analysis, and Software in Heliophysics (DASH) Meeting

Eric Grimes and Bernie Harris, October 10, 2023
Space Physics Data Facility at Goddard Space Flight Center
eric.grimes@nasa.gov

Overview

- Notebooks showing how to access data at CDAWeb / SSCWeb
- Accessing / Analyzing data at SPDF using PySPEDAS

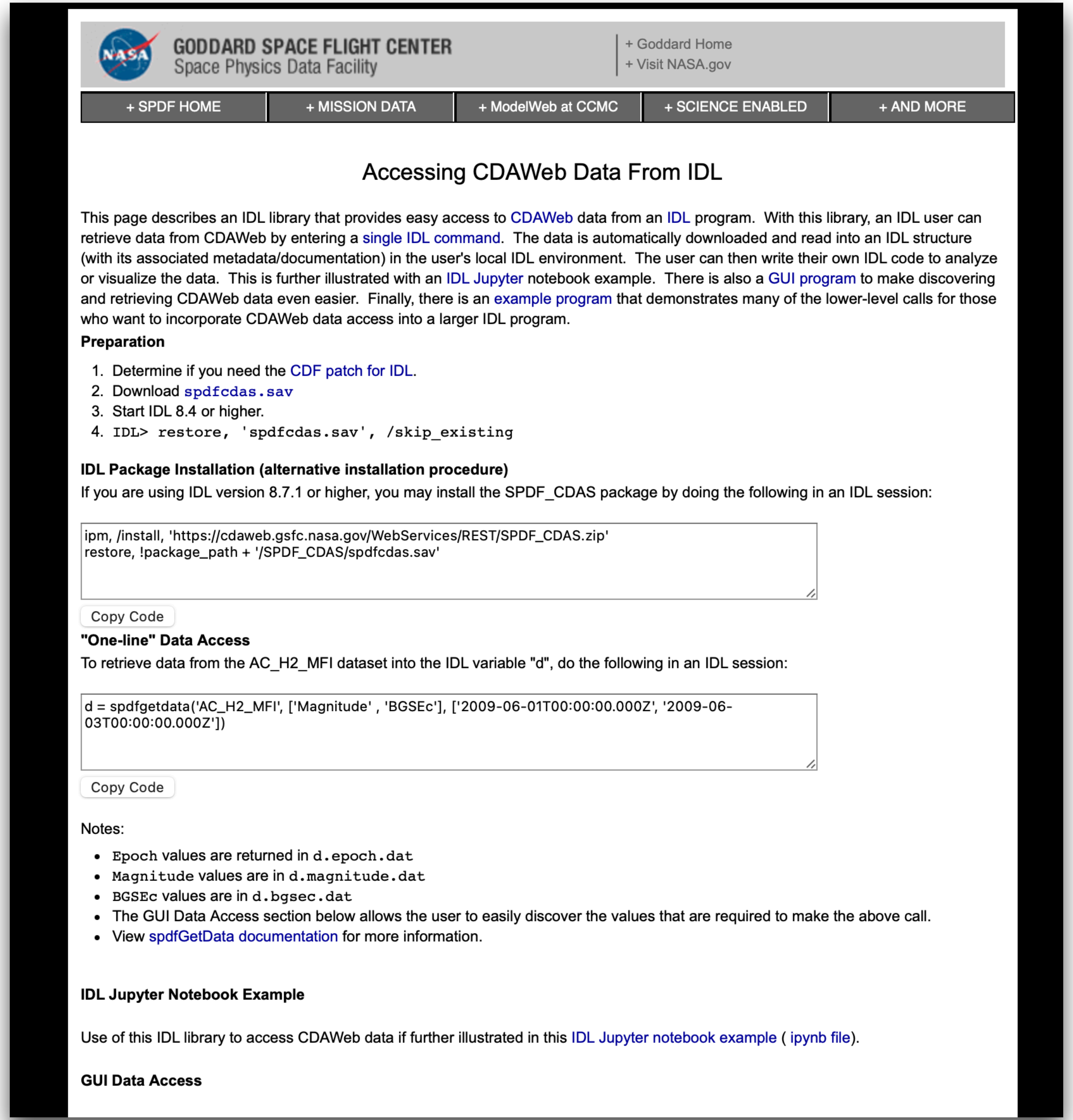
Accessing Data at CDAWeb, SSCWeb & HDP

- Jupyter notebooks are available for IDL and Python
- IDL:
  - https://cdaweb.gsfc.nasa.gov/WebServices/REST/CdasIdlLibrary.html
- Python:
  - https://cdaweb.gsfc.nasa.gov/WebServices/REST/#Jupyter_Notebook_Examples
  - https://sscweb.gsfc.nasa.gov/WebServices/REST/#Jupyter_Notebook_Examples
  - https://heliophysicsdata.gsfc.nasa.gov/WebServices/#Jupyter_Notebook_Examples

IDL Examples

Installation instructions

Jupyter notebook



GODDARD SPACE FLIGHT CENTER
Space Physics Data Facility

+ Goddard Home
+ Visit NASA.gov

+ SPDF HOME | + MISSION DATA | + ModelWeb at CCMC | + SCIENCE ENABLED | + AND MORE

Accessing CDAWeb Data From IDL

This page describes an IDL library that provides easy access to CDAWeb data from an IDL program. With this library, an IDL user can retrieve data from CDAWeb by entering a single IDL command. The data is automatically downloaded and read into an IDL structure (with its associated metadata/documentation) in the user's local IDL environment. The user can then write their own IDL code to analyze or visualize the data. This is further illustrated with an IDL Jupyter notebook example. There is also a GUI program to make discovering and retrieving CDAWeb data even easier. Finally, there is an example program that demonstrates many of the lower-level calls for those who want to incorporate CDAWeb data access into a larger IDL program.

Preparation

1. Determine if you need the CDF patch for IDL.
2. Download `spdfcdas.sav`
3. Start IDL 8.4 or higher.
4. `IDL> restore, 'spdfcdas.sav', /skip_existing`

IDL Package Installation (alternative installation procedure)

If you are using IDL version 8.7.1 or higher, you may install the SPDF_CDAS package by doing the following in an IDL session:

```
ipm, /install, 'https://cdaweb.gsfc.nasa.gov/WebServices/REST/SPDF_CDAS.zip'
restore, !package_path + '/SPDF_CDAS/spdfcdas.sav'
```

Copy Code

"One-line" Data Access

To retrieve data from the AC_H2_MFI dataset into the IDL variable "d", do the following in an IDL session:

```
d = spdfgetdata('AC_H2_MFI', ['Magnitude' , 'BGSEc'], ['2009-06-01T00:00:00.000Z', '2009-06-03T00:00:00.000Z'])
```

Copy Code

Notes:

- `Epoch` values are returned in `d.epoch.dat`
- `Magnitude` values are in `d.magnitude.dat`
- `BGSEc` values are in `d.bgsec.dat`
- The GUI Data Access section below allows the user to easily discover the values that are required to make the above call.
- View spdfGetData documentation for more information.

IDL Jupyter Notebook Example

Use of this IDL library to access CDAWeb data if further illustrated in this IDL Jupyter notebook example (ipynb file).

GUI Data Access

https://cdaweb.gsfc.nasa.gov/WebServices/REST/CdasIdlLibrary.html

IDL

cdasws Example IDL Jupyter Notebook

This Jupyter notebook demonstates using the cdasws IDL library to access data from cdaweb in the IDL programming language.

Note: This notebook is for the IDL version of cdasws. Jupyter notebooks for the Python version of cdasws is available at python cdasws notebooks. This notebook contains the following sections:

Installation

The following contains the procedure to install the cdasws IDL library into your IDL environment. There are different procedures for different versions of IDL.

IDL 8.7.1 and higher

If you have an old version of the SPDF_CDAS package already installed, remove the old version.

In [0]:
```
ipm, /remove, 'SPDF_CDAS'
```

```
Package "SPDF_CDAS" was removed
```

If the lastest version of the SPDF_CDAS package is not already installed, install it as shown below.

In [1]:
```
ipm, /install, 'https://cdaweb.gsfc.nasa.gov/WebServices/REST/SPDF_CDAS.zip'
```

```
Package: SPDF_CDAS, Version: 1.7.44 installed
```

You only need to install a particular version of the package once. You will need to restore the package everytime you restart your IDL session. Restore the package as shown below.

In [2]:
```
restore, !package_path + '/SPDF_CDAS/spdfcdas.sav'
```

IDL 8.4.0 and newer

https://cdaweb.gsfc.nasa.gov/WebServices/REST/CdasIdlLibrary.html

IDL

Setup

Create an SpdfCdas object that will be used in the code that follows.

In [3]:
```
cdas = obj_new('SpdfCdas')
```

Get Observatory Groups

The following code demonstrates how to get the mission/observatory groups supported by cdaweb.

In [5]:
```
groups = cdas.getObservatoryGroups()
foreach group, groups[0:3] do print, group.getName()
print, '...'
```

```
ACE
AIM
AMPTE
ARTEMIS
...
```

Get Intrument Types

The following code demonstrates how to get the intrument types supported by cdaweb.

In [6]:
```
instrTypes = cdas.getInstrumentTypes()
foreach instrType, instrTypes do print, instrType.getName()
```

```
Activity Indices
Electric Fields (space)
Electron Precipitation Bremsstrahlung
Energetic Particle Detector
Engineering
Ephemeris/Attitude/Ancillary
Gamma and X-Rays
Ground-Based HF-Radars
Ground-Based Imagers
Ground-Based Magnetometers, Riometers, Sounders
Ground-Based VLF/ELF/ULF, Photometers
Housekeeping
Imagers (space)
Imaging and Remote Sensing (ITM)
Imaging and Remote Sensing (ITM/Earth)
Imaging and Remote Sensing (Magnetosphere/Earth)
Imaging and Remote Sensing (Sun)
Magnetic Fields (Balloon)
Magnetic Fields (space)
Particles (space)
Plasma and Solar Wind
```

https://cdaweb.gsfc.nasa.gov/WebServices/REST/CdasIdlLibrary.html

IDL

Get Variable Names

The following code demonstrates how to a dataset's variable names.

In [10]:
```
names = cdas.getVariableNames(datasets[-1].getId())
print, names
```

```
Magnitude BGSEc BGSM dBrms SC_pos_GSE SC_pos_GSM
```

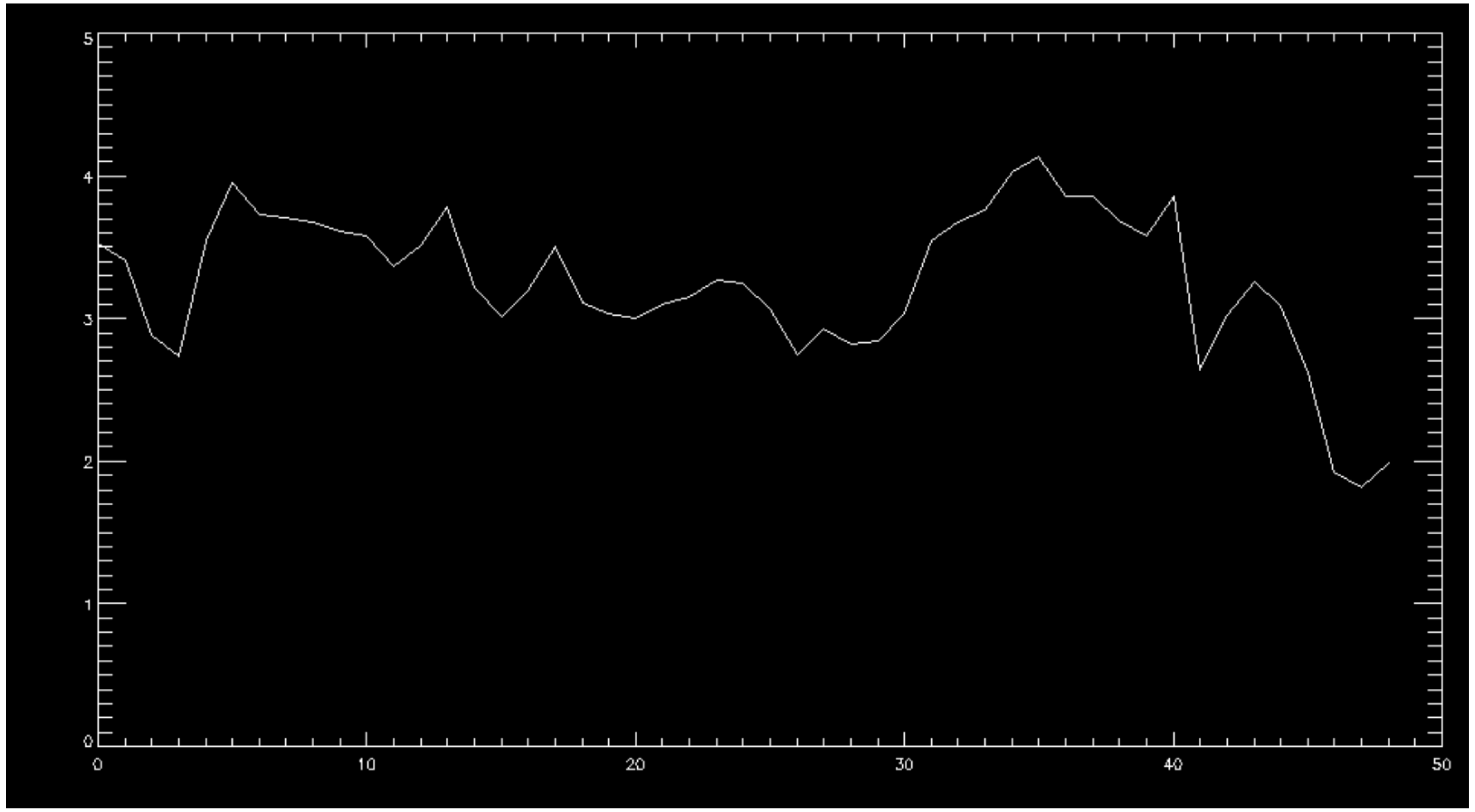
Get Data

The following code demonstrates how to access magnetic field measurements from the ACE mission dataset.

In [11]:
```
d = spdfgetdata('AC_H2_MFI', ['Magnitude' , 'BGSEc'], ['2009-06-01T00:00:00.000Z', '2009-06-03T00:00:00.000Z'])
```

Use the standard IDL PLOT procedure to display the data.

In [12]:
```
plot, d.magnitude.dat
```



Print the values.

In [13]:
```
print, d.magnitude.dat
```

https://cdaweb.gsfc.nasa.gov/WebServices/REST/CdasIdlLibrary.html

Python Examples

Several example notebooks are available showing how to access data from the Coordinated Data Analysis System Web Services (CDASWS)

- python™
  - cdasws (FAQ)
    - cdasws at the PyPI respository.
    - cdasws at the conda-forge respository.
    - Jupyter Notebook Examples: launch binder
      - Basic Example (with SpacePy DataModel results) (ipynb file).
      - Basic Example (with xarray.Dataset results) (ipynb file).
        launch binder
      - NetCDF Example (with xarray.Dataset results) (ipynb file).
        launch binder
      - CDAWlib Graph Example (ipynb file).
        launch binder
      - Magnetic Field Line Conjunction Example (ipynb file) with related data retrieval/plotting using cdasws.
        launch binder
      - CDAWlib Audio Example (ipynb file).
        launch binder
    - These web services themselves can produce example client Python and IDL source code. For example, here is example client code to get data from the AC_H2_MFI dataset.
  - ai.cdas
- NV5 Geospatial Interactive Data Language (IDL)
  - SPDF CDAS IDL Web Service Library. This library makes it easy to call these web services from IDL code. The library contains several complete example clients (e.g., spdfwsexample, spdfcdawebchooser, spdfgetdata, and an IDL Jupyter notebook).
  - IDLnetURL
  - IDLffXMLDOM
- Unix Shell Script
  - cURL
  - xsltproc
  - Shell Script Example. This example should run on most Unix-like (e.g., macOS, Linux, Solaris, Windows Subsystem for Linux, etc.) platforms that have `curl`, `xsltproc`, and `xmllint`. To install the prerequisite software on debian/ubuntu based Linux distributions, do
    - `$ sudo apt install libxml2-utils`
    - `$ sudo apt install xsltproc`

    To install the prerequisite software on rpm based Linux distributions, do
    - `$ sudo dnf install libxml2`
    - `$ sudo dnf install libxslt`

    Here is the output from running the example.
- JavaScript
  - XMLHttpRequest
  - jQuery.ajax()
  - JavaScript Example. This JavaScript/jQuery example demonstrates how to request and recieve data from CDAS.
  - Node.js Example. This example is the Node.js equivalent of the preceding JavaScript example. Simply download the example and run
    - `node CdasNodeExample.js`
- XQuery
  - EXPath HTTP Client
  - cdasws.xql

https://cdaweb.gsfc.nasa.gov/WebServices/REST/#Jupyter_Notebook_Examples

Python

cdasws Example Jupyter Notebook

This Jupyter notebook demonstrates using the cdasws Python package to access data from cdaweb with the data returned in the SpasePy data model. Alternatively, it is possible to have the data returned in an xarray.Dataset. For xarray.Dataset results, see this notebook (ipynb file). This notebook contains the following sections:

Prerequisites

Install the prerequisite software from CDF and the Python Package Index.

1. Install CDF.
2. pip install spacepy
3. pip install cdasws

Setup

Execute some preliminary code that is necessary before the code that follows.

In [1]:
```
from cdasws import CdasWs
import matplotlib.pyplot as plt
cdas = CdasWs()
```

Get Datasets

The following code demontrates how to get a list of datasets.

In [2]:
```
datasets = cdas.get_datasets(observatoryGroup='ACE',
                             instrumentType='Magnetic Fields (space)')
for index, dataset in enumerate(datasets):
```

https://cdaweb.gsfc.nasa.gov/WebServices/REST/#Jupyter_Notebook_Examples

Python

Get Datasets

The following code demontrates how to get a list of datasets.

In [2]:
```
datasets = cdas.get_datasets(observatoryGroup='ACE',
                             instrumentType='Magnetic Fields (space)')
for index, dataset in enumerate(datasets):
    print(dataset['Id'], dataset['Label'])
    if index == 5:
        print('...')
        break
```

```
AC_AT_DEF ACE Hourly RTN, GSE and J2000 GCI Attitude direction cosines - E. C. Stone (California Institute of Techno
logy)
AC_H0_MFI H0 - ACE Magnetic Field 16-Second Level 2 Data - N. Ness (Bartol Research Institute)
AC_H1_MFI H1 - ACE Magnetic Field 4-Minute Level 2 Data - N. Ness (Bartol Research Institute)
AC_H2_MFI H2 - ACE Magnetic Field 1-Hour Level 2 Data - N. Ness (Bartol Research Institute)
AC_H3_MFI H3 - ACE Magnetic Field 1-Second Level 2 Data - N. Ness (Bartol Research Institute)
AC_K0_GIFWALK Links to ACE KP pre-generated survey and other plots - Polar-Wind-Geotail Ground System (NASA GSFC)
...
```

Get Dataset Variables

The following code demonstrates how to get a dataset's variables.

In [3]:
```
variables = cdas.get_variables('AC_H1_MFI')
for variable in variables:
    print(variable['Name'], variable['LongDescription'])
```

```
Magnitude B-field magnitude
BGSEc Magnetic Field Vector in GSE Cartesian coordinates (4 min)
BGSM Magnetic field vector in GSM coordinates (4 min)
SC_pos_GSE ACE s/c position, 3 comp. in GSE coord.
SC_pos_GSM ACE s/c position, 3 comp. in GSM coord.
```

Get Data

The following code demonstrates how to access magnetic field measurements from the ACE AC_H1_MFI dataset.

In [4]:
```
data = cdas.get_data('AC_H1_MFI', ['Magnitude', 'BGSEc'],
                     '2009-06-01T00:00:00Z', '2009-06-01T00:10:00Z')[1]
print(data)
```

```
{'Epoch': VarCopy([datetime.datetime(2009, 6, 1, 0, 0),
         datetime.datetime(2009, 6, 1, 0, 4),
         datetime.datetime(2009, 6, 1, 0, 8)], dtype=object), 'Magnitude': VarCopy([3.495, 3.474, 3.477], dtype=floa
t32), 'BGSEc': VarCopy([[-0.106,  2.521, -2.391],
         [-0.412,  2.402, -2.449],
         [-0.094,  2.309, -2.587]], dtype=float32), 'cartesian': VarCopy(['x_component', 'y_component', 'z_componen
```

https://cdaweb.gsfc.nasa.gov/WebServices/REST/#Jupyter_Notebook_Examples

Python

Display Metadata

The following code displays metadata for the Magnitude variable.

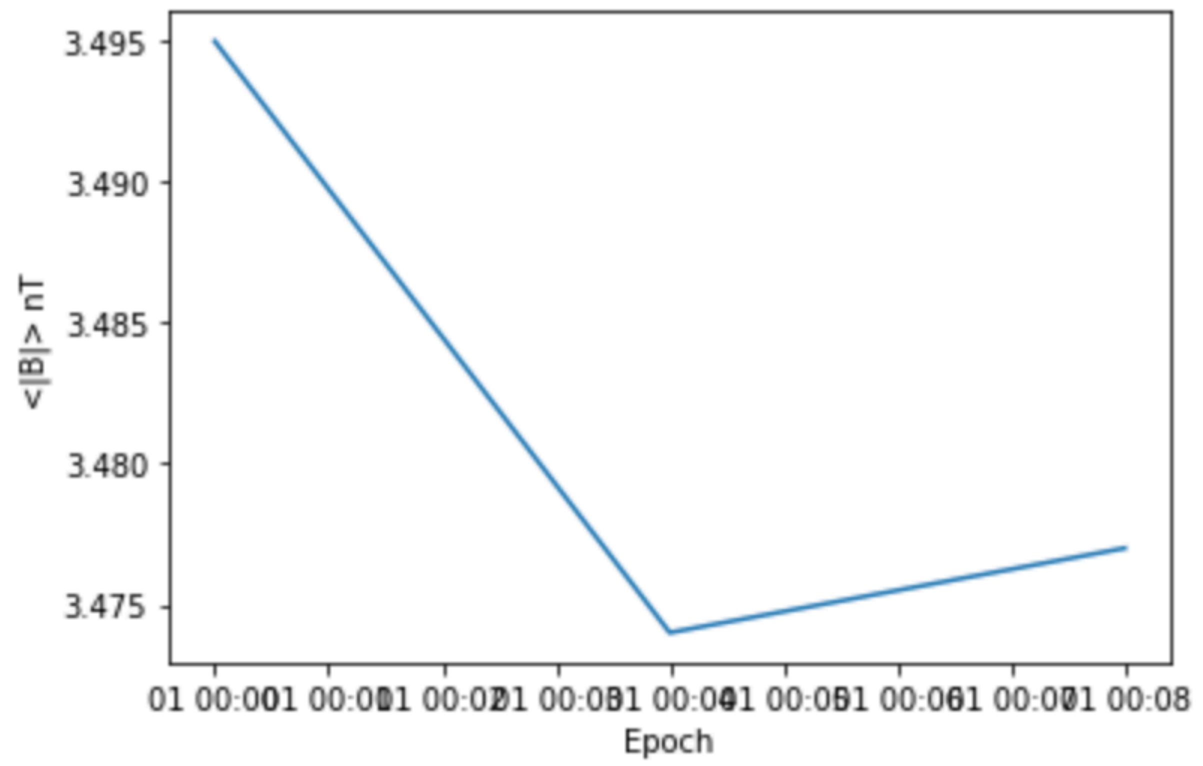
```
In [5]: print(data['Magnitude'].attrs)
```

{'FIELDNAM': 'B-field magnitude', 'VALIDMIN': 0.0, 'VALIDMAX': 500.0, 'SCALEMIN': 0.0, 'SCALEMAX': 10.0, 'UNITS': 'nT', 'FORMAT': 'F8.3', 'VAR_TYPE': 'data', 'DICT_KEY': 'magnetic_field>magnitude', 'FILLVAL': -1e+31, 'DEPEND_0': 'Epoch', 'CATDESC': 'B-field magnitude', 'LABLAXIS': '<|B|>', 'DISPLAY_TYPE': 'time_series', 'DIM_SIZES': 0}

Plot Values

The following code plots the Magnitude variable's values using the label values from the metadata.

```
In [6]: plt.plot(data['Epoch'], data['Magnitude'])
plt.xlabel(data['Epoch'].attrs['LABLAXIS'])
plt.ylabel(data['Magnitude'].attrs['LABLAXIS'] + ' ' +
           data['Magnitude'].attrs['UNITS'])
plt.show()
```



Binning Example

For analysis, it is often useful to place two datasets that have different timestamps on the same time grid (with optional spike removal). The following demonstrates doing this with cdasws and the datasets AC_H0_SWE and AC_H2_SWE. For more information on binning, see binning in cdaweb.

Display Original Data

The following code gets and displays the original, unbinned data.

https://cdaweb.gsfc.nasa.gov/WebServices/REST/#Jupyter_Notebook_Examples

Python Examples

Several example notebooks are available showing how to access data from the Satellite Situation Center Web Services (SSCWS)

- python
  - sscws
    - Getting started with the `sscws` client library.
    - sscws at the PyPI respository.
    - sscws at the conda-forge respository.
    - Jupyter Notebook Examples: launch binder
      - Simple Location Example (ipynb file).
        launch binder
      - Complex Location Example (ipynb file) requesting many values including magnetic field line tracing. This example also demonstrates how to diagnose a problem with an invalid request.
        launch binder
      - Radial Conjunction With Ground Location (ipynb file).
        launch binder
      - Magnetic Field Line Conjunction Example (ipynb file) with related data retrieval/plotting using cdasws.
        launch binder
    - These web services themselves can produce example client Python and IDL source code. For example, here is example client code to get International Space Station (ISS) orbit information.
- L3Harris Interactive Data Language (IDL)
  - SPDF SSC IDL Web Services Library. This library makes it easy to call these web services from IDL code. The library contains several complete example clients (e.g., spdfsscwsexample) and an IDL Jupyter notebook).
  - IDLnetURL
  - IDLffXMLDOM
- Unix Shell Script
  - cURL
  - xsltproc
  - jq
  - Shell Script Examples. These examples should run on most Unix-like (e.g., macOS, Linux, Solaris, Windows Subsystem for Linux, etc.) platforms that have `curl`, `xsltproc`, and `xmllint`. To install the prerequisite software on debian/ubuntu based Linux distributions, do
    - `$ sudo apt install libxml2-utils`
    - `$ sudo apt install xsltproc`

    To install the prerequisite software on rpm based Linux distributions, do
    - `$ sudo dnf install libxml2`
    - `$ sudo dnf install libxslt`
- JavaScript
  - Fetch API
  - jQuery .ajax()
  - XMLHttpRequest
- jQuery Examples. These examples should run on most modern browsers.
  - Get Graphs example.
  - Get Locations example.
- Java
  - `java.net.http.HttpClient`
  - `java.net.HttpURLConnection`
  - JAX-RS 2 client API

https://sscweb.gsfc.nasa.gov/WebServices/REST/#Jupyter_Notebook_Examples

Python

sscws Example Jupyter Notebook

This Jupyter notebook demonstrates using the sscws Python package to access satellite location and (modeled) magnetic field information. This notebook contains the following sections:

Prerequisites

Install the prerequisite software from the Python Package Index if it is not already installed.

In [ ]:
```
#%pip install sscws
```

Setup

Execute some preliminary code that is necessary before the code that follows.

In [1]:
```
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
from packaging import version

from sscws.sscws import SscWs
ssc = SscWs()
```

Get Observatories

The following code demonstrates how to get the list of available observatories.

In [2]:
```
result = ssc.get_observatories()
for observatory in result['Observatory'][:5]:
    print('{:15s} {:20.20s} {:25s}'.format(observatory['Id'], observatory['Name'], observatory['StartTime'].isoformat(
print('...')
```

```
ace              ACE                   1997-08-25T17:48:00+00:00
active           Active                1989-09-29T00:00:00+00:00
aec              AE-C                  1973-12-17T08:01:00+00:00
aed              AE-D                  1975-10-17T00:00:00+00:00
aee              AE-E                  1975-11-20T21:04:00+00:00
...
```

Get Ground Stations

https://sscweb.gsfc.nasa.gov/WebServices/REST/#Jupyter_Notebook_Examples

Python

```
SPL   Siple               -76.00  -84.00
HBA   Halley Bay          -75.52  -26.60
...
```

Get Locations

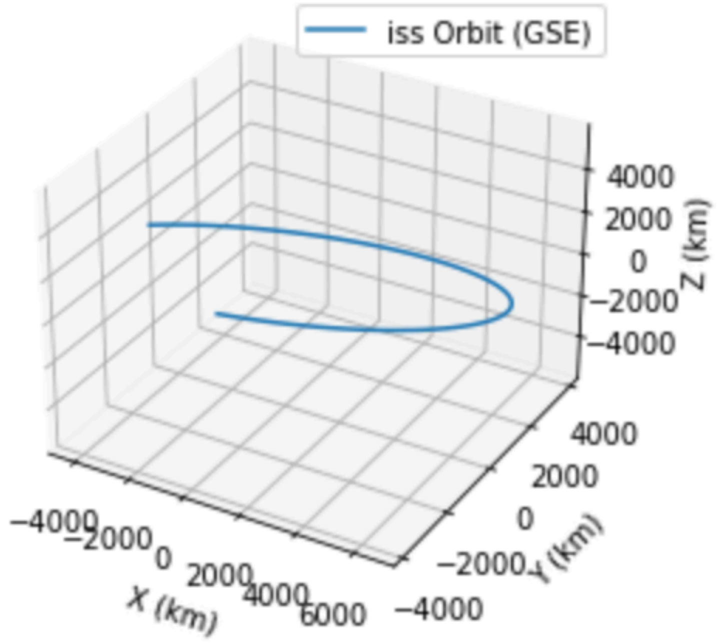
The following code gets location information for the International Space Station (ISS) spacecraft and prints a few X coordinate values.

In [4]:
```
result = ssc.get_locations(['iss'],
                            ['2020-01-01T00:00:00Z',
                             '2020-01-01T01:00:00Z'])
data = result['Data'][0]
coords = data['Coordinates'][0]
print(coords['X'][:5])
```

```
[-1473.76002814 -1035.57214121  -592.61584877  -146.91796194
   299.48049963]
```

And the following code plots the ISS location information.

In [5]:
```
fig = plt.figure()
if version.parse(mpl.__version__) < version.parse('3.4'):
    ax = fig.gca(projection='3d')
else:
    ax = Axes3D(fig, auto_add_to_figure=False)
    fig.add_axes(ax)
ax.set_xlabel('X (km)')
ax.set_ylabel('Y (km)')
ax.set_zlabel('Z (km)')
title = data['Id'] + ' Orbit (' + \
        coords['CoordinateSystem'].value.upper() + ')'
ax.plot(coords['X'], coords['Y'], coords['Z'], label=title)
ax.legend()
plt.show()
```



Additional Documentation

View the sscws API for additonal features. Additional notebook examples are also available.

https://sscweb.gsfc.nasa.gov/WebServices/REST/#Jupyter_Notebook_Examples

Python

sscws Conjunction Example Jupyter Notebook

This Jupyter notebook demonstrates using the sscws Python package to find satellite conjunctions. Additionally, it demonstrates using cdasws Python package to get data from cdaweb for the satellite/ground stations involved in the conjunction. This notebook contains the following sections:

1. Prerequisites
2. Setup
3. Define Conjunction Query
4. Run Conjunction Query
5. Display the results
6. Setup For Access To CDAWeb Data
7. Get Data During Conjunctions
8. Additional Documentation

Prerequisites

For just satellite conjunctions, install the following software from the Python Package Index if it is not already installed.

In [1]:
```
#%pip install sscws
```

To also get data from cdaweb, install the following additional software from the Python Package Index.

In [2]:
```
#%pip install xarray
#%pip install cdflib
#%pip install cdasws
```

Setup

Execute some preliminary code that is necessary before the code that follows.

In [3]:
```
import numpy as np
from sscws.sscws import SscWs
from sscws.conjunctions import BoxConjunctionArea, ConditionOperator,\
    GroundStationCondition, GroundStationConjunction,\
    Satellite, SatelliteCondition, TraceCoordinateSystem
from sscws.coordinates import CoordinateComponent, CoordinateSystem,\
    SurfaceGeographicCoordinates
from sscws.request import DataRequest, QueryRequest, SatelliteSpecification
from sscws.timeinterval import TimeInterval
from sscws.tracing import BFieldTraceDirection, TraceType
ssc = SscWs()
```

Define Conjunction Query

The following code defines a query to find magnetic field line conjunctions of at least two THEMIS satellites with one of four THEMIS ground

https://sscweb.gsfc.nasa.gov/WebServices/REST/#Jupyter_Notebook_Examples

Python

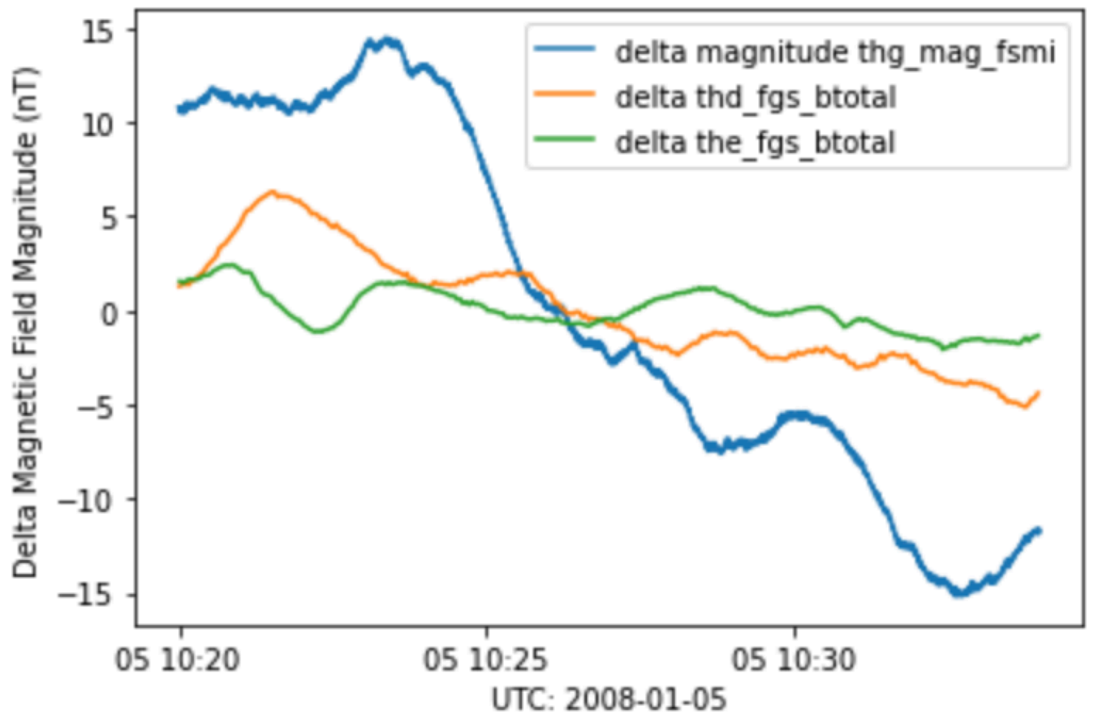
Get Data During Conjunctions

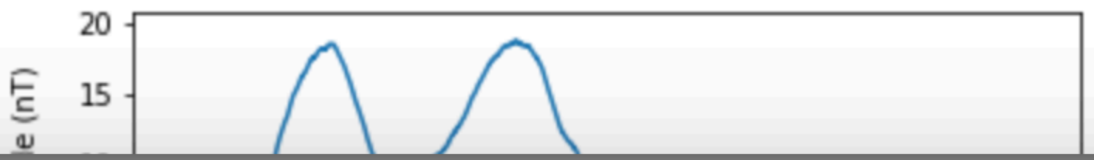
The following code gets data from cdaweb during the conjunctions and plots it.

In [8]:

```
for conjunction in result['Conjunction']:
    time_interval = conjunction['TimeInterval']
    start = time_interval['Start']
    end = time_interval['End']
    sats = []
    for sat_des in conjunction['SatelliteDescription']:
        sats.append(sat_des['Satellite'])
    sat = conjunction['SatelliteDescription'][0]['Satellite']
    gs = conjunction['SatelliteDescription'][0]['Description'][0]['TraceDescription']['Target']['GroundStation']
    datasets = [get_cdaweb_ds(gs)]
    for sat in sats:
        datasets.append(get_cdaweb_ds(sat))
    for ds in datasets:
        var_name = get_a_mag_var_name(ds, cdas.get_variable_names(ds))
        status, data = cdas.get_data(ds, [var_name], start, end,
                                     dataRepresentation = DataRepresentation.XARRAY)
        var_data = data[var_name]
        epoch_var_name = var_data.attrs['DEPEND_0']
        if var_data.ndim == 2:
            var_data = np.linalg.norm(var_data, axis = 1)
            data_label = 'delta magnitude ' + var_name
        else:
            data_label = 'delta ' + var_name
        mean = var_data.mean(axis = 0)
        delta_data = var_data - mean
        plt.plot(data[epoch_var_name], delta_data, label = data_label)

    plt.xlabel('UTC: ' + start.date().isoformat())
    plt.ylabel('Delta Magnetic Field Magnitude (nT)')
    plt.legend()
    plt.show()
```

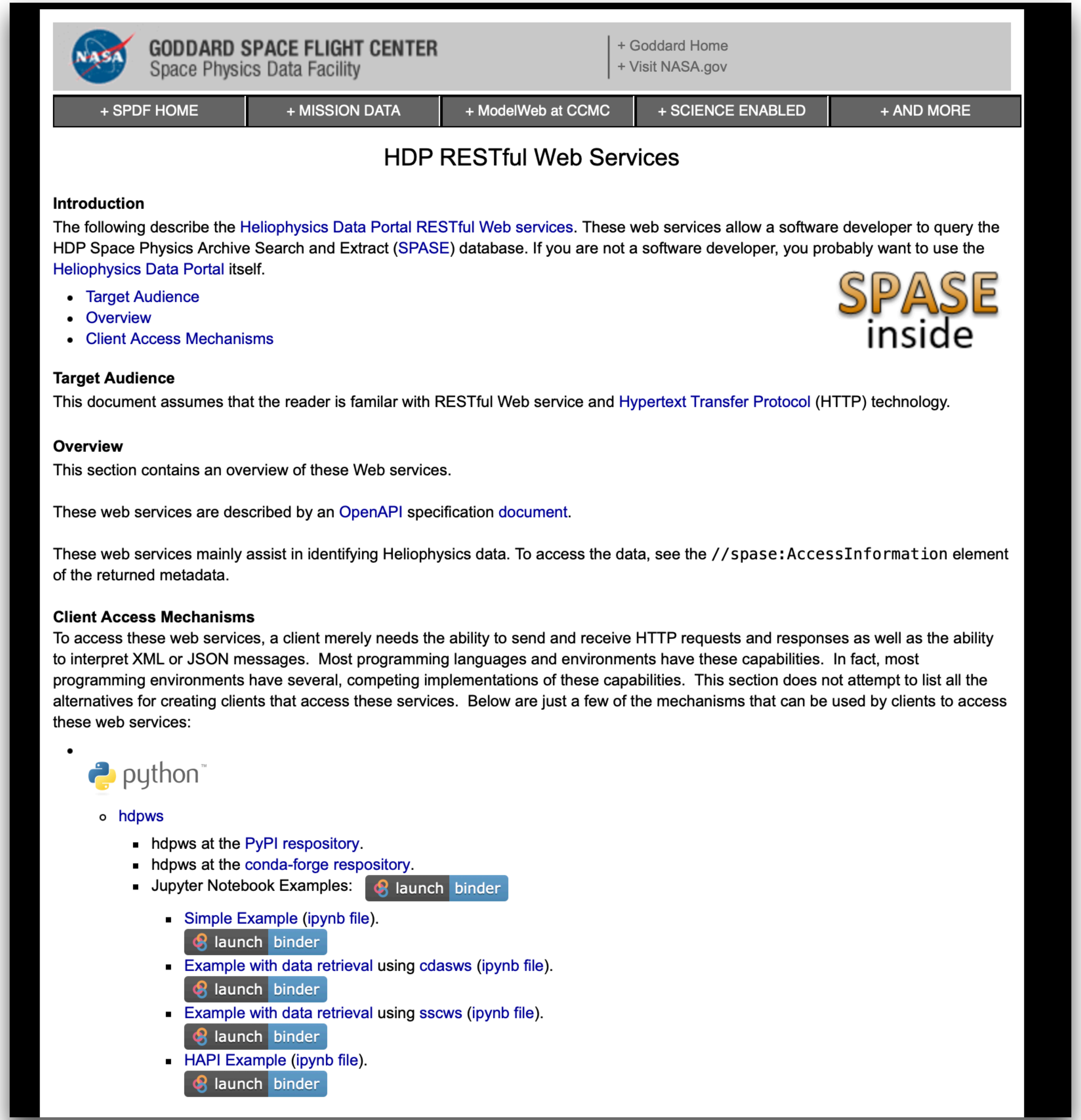




https://sscweb.gsfc.nasa.gov/WebServices/REST/#Jupyter_Notebook_Examples

Python Examples

Several example notebooks are available showing how to access the Heliophysics Data Portal (HDP)


GODDARD SPACE FLIGHT CENTER
Space Physics Data Facility

+ Goddard Home
+ Visit NASA.gov

+ SPDF HOME | + MISSION DATA | + ModelWeb at CCMC | + SCIENCE ENABLED | + AND MORE

HDP RESTful Web Services

Introduction

The following describe the Heliophysics Data Portal RESTful Web services. These web services allow a software developer to query the HDP Space Physics Archive Search and Extract (SPASE) database. If you are not a software developer, you probably want to use the Heliophysics Data Portal itself.

- Target Audience
- Overview
- Client Access Mechanisms

SPASE inside

Target Audience

This document assumes that the reader is familar with RESTful Web service and Hypertext Transfer Protocol (HTTP) technology.

Overview

This section contains an overview of these Web services.

These web services are described by an OpenAPI specification document.

These web services mainly assist in identifying Heliophysics data. To access the data, see the `//spase:AccessInformation` element of the returned metadata.

Client Access Mechanisms

To access these web services, a client merely needs the ability to send and receive HTTP requests and responses as well as the ability to interpret XML or JSON messages. Most programming languages and environments have these capabilities. In fact, most programming environments have several, competing implementations of these capabilities. This section does not attempt to list all the alternatives for creating clients that access these services. Below are just a few of the mechanisms that can be used by clients to access these web services:

- python™
  - hdpws
    - hdpws at the PyPI respository.
    - hdpws at the conda-forge respository.
    - Jupyter Notebook Examples: launch binder
      - Simple Example (ipynb file). launch binder
      - Example with data retrieval using cdasws (ipynb file). launch binder
      - Example with data retrieval using sscws (ipynb file). launch binder
      - HAPI Example (ipynb file). launch binder

https://heliophysicsdata.gsfc.nasa.gov/WebServices/#Jupyter_Notebook_Examples

Python

hdpws Example Jupyter Notebook

This Jupyter notebook demonstrates using the hdpws Python package to access Space Physics Archive Search and Extract (SPASE) metadata documents from the Heliophysics Data Portal (HDP). It assumes some familarity with the SPASE data model. This notebook contains the following sections:

Prerequisites

Install the prerequisite software from Python Package Index (PyPI) software repository.

1. pip install hdpws

Setup

Execute some preliminary code that is necessary before the code that follows.

[3]:

```
from hdpws.hdpws import HdpWs
from hdpws import NAMESPACES as NS
from hdpws.resourcetype import ResourceType as rt
from hdpws.spase import AccessURL, HapiAccessURL

from IPython.core.display import HTML
```

https://heliophysicsdata.gsfc.nasa.gov/WebServices/#Jupyter_Notebook_Examples

Python

Get ObservedRegions

The following code demonstrates how to get the list of available /Spase/ObservedRegion values.

In [7]:
```
result = hdp.get_observed_regions()
observed_regions = result['ObservedRegion']
print(f'{len(observed_regions)} HDP Observed Regions:')
for value in observed_regions[0:9]:
    print(f'    {value}')
print('    ...')
```

```
122 HDP Observed Regions:
    Asteroid
    Comet
    Earth
    Earth.Magnetosheath
    Earth.Magnetosphere
    Earth.Magnetosphere.Magnetotail
    Earth.Magnetosphere.Main
    Earth.Magnetosphere.Plasmasphere
    Earth.Magnetosphere.Polar
    ...
```

Get ObservatoryIDs

The following code demonstrates how to get the list of available /Spase/Observatory/ResourceID values.

In [8]:
```
result = hdp.get_observatory_ids()
observatory_ids = result['ObservatoryID']
print(f'{len(observatory_ids)} HDP ObservatoryIDs:')
for value in observatory_ids[0:9]:
    print(f'    {value}')
print('    ...')
```

```
2936 HDP ObservatoryIDs:
    spase://SMWG/Observatory/AE-D
    spase://SMWG/Observatory/Helios1
    spase://SMWG/Observatory/DynamicsExplorer1
    spase://SMWG/Observatory/SolarOrbiter
    spase://SMWG/Observatory/IMP8
    spase://SMWG/Observatory/MarsExpress
    spase://SMWG/Observatory/GIRO
    spase://SMWG/Observatory/Interball-Tail
    spase://SMWG/Observatory/IRIS
    ...
```

Get Numerical/Display Data

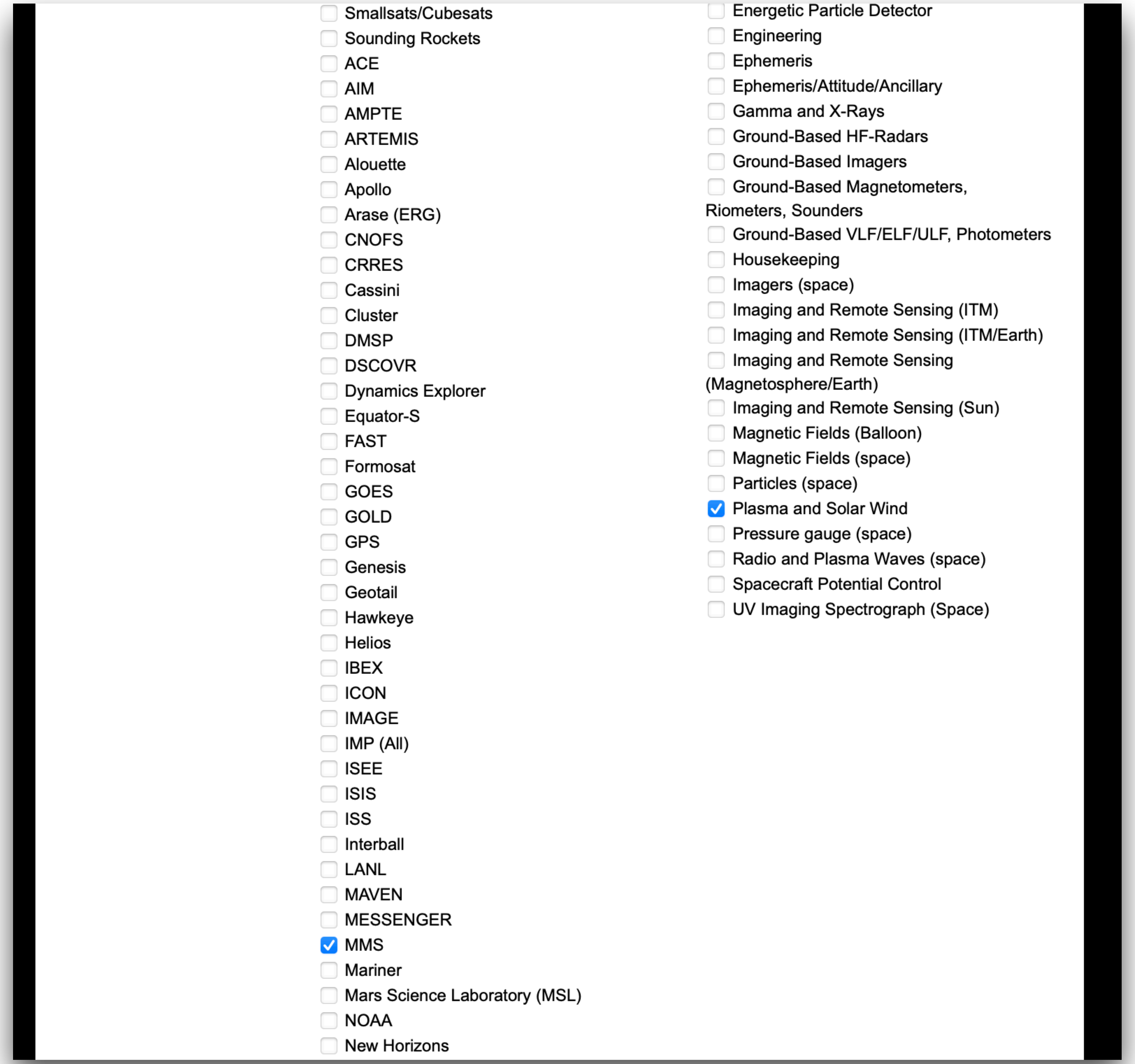
The following code demonstrates how to get SPASE Numerical/Display data documents matching the specified search criteria.

https://heliophysicsdata.gsfc.nasa.gov/WebServices/#Jupyter_Notebook_Examples

More Examples at CDAWeb

- We also have example notebooks for each dataset, available at:

https://cdaweb.gsfc.nasa.gov/



[Available Time Range: 2015/07/15 15:27:04 - 2023/08/12 05:23:32] Info Metadata

- [ ] **MMS2_FPI_BRST_L2_DIS-DIST:** MMS Satellite Number 2 Dual Ion Spectrometer Burst-resolution instrument distributions - J. Burch, C. Pollock (SwRI, NASA/GSFC)
  [Available Time Range: 2015/06/22 17:56:25 - 2023/08/12 05:23:32] Info Metadata
- [ ] **MMS3_FPI_BRST_L2_DIS-DIST:** MMS Satellite Number 3 Dual Ion Spectrometer Burst-resolution instrument distributions - J. Burch, C. Pollock (SwRI, NASA/GSFC)
  [Available Time Range: 2015/05/06 00:00:00 - 2023/08/12 05:23:32] Info Metadata
- [ ] **MMS4_FPI_BRST_L2_DIS-DIST:** MMS Satellite Number 4 Dual Ion Spectrometer Burst-resolution instrument distributions - J. Burch, C. Pollock (SwRI, NASA/GSFC)
  [Available Time Range: 2015/07/27 15:30:54 - 2023/08/12 05:23:32] Info Metadata
- [ ] **MMS1_FPI_FAST_L2_DES-DIST:** MMS Satellite Number 1 Dual Electron Spectrometer Fast Survey-resolution instrument distributions - J. Burch; C. Pollock and B. Giles (SwRI; NASA/GSFC)
  [Available Time Range: 2015/07/15 15:11:27 - 2023/08/31 23:59:56] Info Metadata
- [ ] **MMS2_FPI_FAST_L2_DES-DIST:** MMS Satellite Number 2 Dual Electron Spectrometer Fast Survey-resolution instrument distributions - J. Burch; C. Pollock and B. Giles (SwRI; NASA/GSFC)
  [Available Time Range: 2015/06/22 17:46:20 - 2023/08/31 23:59:58] Info Metadata
- [ ] **MMS3_FPI_FAST_L2_DES-DIST:** MMS Satellite Number 3 Dual Electron Spectrometer Fast Survey-resolution instrument distributions - J. Burch; C. Pollock and B. Giles (SwRI; NASA/GSFC)
  [Available Time Range: 2015/05/06 00:00:01 - 2023/08/31 23:59:59] Info Metadata
- [ ] **MMS4_FPI_FAST_L2_DES-DIST:** MMS Satellite Number 4 Dual Electron Spectrometer Fast Survey-resolution instrument distributions - J. Burch; C. Pollock and B. Giles (SwRI; NASA/GSFC)
  [Available Time Range: 2015/07/28 00:00:02 - 2023/08/31 23:59:56] Info Metadata
- [ ] **MMS1_FPI_FAST_L2_DIS-DIST:** MMS Satellite Number 1 Dual Ion Spectrometer Fast Survey-resolution instrument distributions - J. Burch; C. Pollock and B. Giles (SwRI; NASA/GSFC)
  [Available Time Range: 2015/07/15 15:11:27 - 2023/08/31 23:59:56] Info Metadata
- [ ] **MMS2_FPI_FAST_L2_DIS-DIST:** MMS Satellite Number 2 Dual Ion Spectrometer Fast Survey-resolution instrument distributions - J. Burch; C. Pollock and B. Giles (SwRI; NASA/GSFC)
  [Available Time Range: 2015/06/22 17:46:20 - 2023/08/31 23:59:58] Info Metadata
- [ ] **MMS3_FPI_FAST_L2_DIS-DIST:** MMS Satellite Number 3 Dual Ion Spectrometer Fast Survey-resolution instrument distributions - J. Burch; C. Pollock and B. Giles (SwRI; NASA/GSFC)
  [Available Time Range: 2015/05/06 00:00:01 - 2023/08/31 23:59:59] Info Metadata
- [ ] **MMS4_FPI_FAST_L2_DIS-DIST:** MMS Satellite Number 4 Dual Ion Spectrometer Fast Survey-resolution instrument distributions - J. Burch; C. Pollock and B. Giles (SwRI; NASA/GSFC)
  [Available Time Range: 2015/07/28 00:00:02 - 2023/08/31 23:59:56] Info Metadata
- [x] **MMS1_FPI_FAST_L2_DIS-MOMS:** MMS Satellite Number Dual Ion Spectrometer Survey-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  ] Info Metadata
- [ ] **MMS2_FPI_FAST_L2_DIS-MOMS:** MMS Satellite Number Dual Ion Spectrometer Survey-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/06/22 17:46:20 - 2023/08/31 23:59:58] Info Metadata
- [ ] **MMS3_FPI_FAST_L2_DIS-MOMS:** MMS Satellite Number 3 Dual Ion Spectrometer Survey-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/05/06 00:00:01 - 2023/08/31 23:59:59] Info Metadata
- [ ] **MMS4_FPI_FAST_L2_DIS-MOMS:** MMS Satellite Number 4 Dual Ion Spectrometer Survey-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/07/28 00:00:02 - 2023/08/31 23:59:56] Info Metadata
- [ ] **MMS1_FPI_BRST_L2_DIS-MOMS:** MMS Satellite Number 1 Dual Ion Spectrometer Burst-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/07/15 15:27:04 - 2023/08/12 05:23:32] Info Metadata
- [ ] **MMS2_FPI_BRST_L2_DIS-MOMS:** MMS Satellite Number 2 Dual Ion Spectrometer Burst-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/06/22 17:56:25 - 2023/08/12 05:23:32] Info Metadata
- [ ] **MMS3_FPI_BRST_L2_DIS-MOMS:** MMS Satellite Number 3 Dual Ion Spectrometer Burst-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/05/06 00:00:00 - 2023/08/12 05:23:32] Info Metadata
- [ ] **MMS4_FPI_BRST_L2_DIS-MOMS:** MMS Satellite Number 4 Dual Ion Spectrometer Burst-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/07/27 15:30:54 - 2023/08/12 05:23:32] Info Metadata
- [ ] **MMS1_FPI_FAST_L2_DES-MOMS:** MMS Satellite Number 1 Dual Electron Spectrometer Survey-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/07/15 15:11:27 - 2023/08/31 23:59:56] Info Metadata
- [ ] **MMS2_FPI_FAST_L2_DES-MOMS:** MMS Satellite Number 2 Dual Electron Spectrometer Survey-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/06/22 17:46:20 - 2023/08/31 23:59:58] Info Metadata
- [ ] **MMS3_FPI_FAST_L2_DES-MOMS:** MMS Satellite Number 3 Dual Electron Spectrometer Survey-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/05/06 00:00:01 - 2023/08/31 23:59:59] Info Metadata
- [ ] **MMS4_FPI_FAST_L2_DES-MOMS:** MMS Satellite Number 4 Dual Electron Spectrometer Survey-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/07/28 00:00:02 - 2018/06/07 23:59:59] Info Metadata
- [ ] **MMS1_FPI_BRST_L2_DES-MOMS:** MMS Satellite Number 1 Dual Electron Spectrometer Burst-resolution distribution moments - J.Burch; C.Pollock, B.Giles (SwRI, NASA/GSFC)
  [Available Time Range: 2015/07/15 15:27:04 - 2023/08/12 05:23:32] Info Metadata

Click "Info" for the dataset you're interested in

https://cdaweb.gsfc.nasa.gov/

bulk velocity to obtain the corrected velocity vector.

---> (no error bars displayed) MMS1 FPI/DIS ion bulk-velocity GSE vector during this survey [mms1_dis_bulkv_gse_fast_noerr]

MMS1 FPI/DIS ion bulk-velocity spintone vector in DBCS during this survey [mms1_dis_bulkv_spin_dbcs_fast]

Estimated error in spin-plane bulk velocity (km/s) due to imperfect sensor suite flat-fielding

MMS1 FPI/DIS ion bulk-velocity spintone vector in GSE during this survey [mms1_dis_bulkv_spin_gse_fast]

Estimated error in spin-plane bulk velocity (km/s) due to imperfect sensor suite flat-fielding

MMS1 FPI/DIS ion pressure tensor DBCS matrix during this survey [mms1_dis_prestensor_dbcs_fast]

The 3x3 matrix is represented as: Row1) Pxx Pxy Pxz, Row2) Pyx Pyy Pyz, Row3) Pzx Pzy Pzz. Note that Pij=Pji.

MMS1 FPI/DIS ion pressure tensor GSE matrix during this survey [mms1_dis_prestensor_gse_fast]

The 3x3 matrix is represented as: Row1) Pxx Pxy Pxz, Row2) Pyx Pyy Pyz, Row3) Pzx Pzy Pzz. Note that Pij=Pji.

MMS1 FPI/DIS ion background pressure during this survey [mms1_dis_pres_bg_fast]

MMS1 FPI/DIS ion temperature tensor DBCS matrix during this survey [mms1_dis_temptensor_dbcs_fast]

The 3x3 matrix is represented as: Row1) Txx Txy Txz, Row2) Tyx Tyy Tyz, Row3) Tzx Tzy Tzz. Note that Tij=Tji.

MMS1 FPI/DIS ion temperature tensor GSE matrix during this survey [mms1_dis_temptensor_gse_fast]

The 3x3 matrix is represented as: Row1) Txx Txy Txz, Row2) Tyx Tyy Tyz, Row3) Tzx Tzy Tzz. Note that Tij=Tji.

MMS1 FPI/DIS ion heat-flux DBCS vector during this survey [mms1_dis_heatq_dbcs_fast]

MMS1 FPI/DIS ion heat-flux GSE vector during this survey [mms1_dis_heatq_gse_fast]

MMS1 FPI/DIS ion parallel temperature during this BP [mms1_dis_temppara_fast]

MMS1 FPI/DIS ion perpendicular temperature during this BP [mms1_dis_tempperp_fast]

Click "Data Access Code Examples"

Dataset in CDAWeb

Data Access Code Examples

Back to top

MMS1_FPI_FAST_L2_DIS-PARTMOMS (spase://NASA/NumericalData/MMS/1/FastPlasmaInvestigation/DIS/Fast/Level2/PartialMoments/PT4.5S)

Description

FPI usually operates in Fast Survey (FS) Mode in the MMS Region Of Interest (ROI) for the current Mission Phase. Data are taken at burst (30/150 ms for DES/DIS) resolution in this mode. Data are also made available at survey (4.5 s) resolution. Per mission design, not all burst-resolution data are downlinked, but all survey data are downlinked. Planning around calibration activities, avoidance of Earth radiation belts, etc, when possible, FPI usually operates in Slow Survey (SS) Mode (60 s resolution) outside of ROI. This product contains partial moments that come from performing the standard moment integrals over a limited portion of velocity space. The resulting quantities are named similarly to their corresponding standard moments, but are decorated with 'part' to differentiate. For example, density_part is the density moment integrated from a

https://cdaweb.gsfc.nasa.gov/

Launch notebook in Binder showing how to access data for your dataset

CDAS Web Service Client Code Examples

The following web service client code examples demonstrates how to access data from the MMS1_FPI_FAST_L2_DIS-MOMS dataset from particular programming environments.

Jupyter Notebook on Binder

The following link launches a Python Jupyter Notebook that demonstrates using the cdasws library to access MMS1_FPI_FAST_L2_DIS-MOMS data in a Jupyter Notebook. It is merely an example and does not show all the capabilities of the library. You should edit the code to suit your needs.

launch binder

cdasws Python Library

The following code demonstrates using the cdasws library to access MMS1_FPI_FAST_L2_DIS-MOMS data in Python. It is merely an example and does not show all the capabilities of the library. You should edit the code to suit your needs.

```
#  Install these prerequisites once before executing the example code:
#  Option 1.
#    Install CDF from https://cdf.gsfc.nasa.gov/
#    pip install -U spacepy
#    pip install -U cdasws
#  Option 2.
#    pip install -U xarray
#    pip install -U cdflib
#    pip install -U cdasws

from cdasws import CdasWs
cdas = CdasWs()

#  Edit the following vars, time variables, and printing to suit your environment
# (spacepy or cdflib) and needs.
vars =
['mms1_dis_errorflags_fast','mms1_dis_compressionloss_fast','mms1_dis_startdelphi_count_fast','mms1_dis_
startdelphi_angle_fast','mms1_dis_energyspectr_px_fast','mms1_dis_energyspectr_mx_fast','mms1_dis_ener
gyspectr_py_fast','mms1_dis_energyspectr_my_fast','mms1_dis_energyspectr_pz_fast','mms1_dis_energyspe
ctr_mz_fast','mms1_dis_energyspectr_omni_fast','mms1_dis_spectr_bg_fast','mms1_dis_numberdensity_bg_f
ast','mms1_dis_numberdensity_fast','mms1_dis_numberdensity_fast_noerr','mms1_dis_numberdensity_err_fa
st','mms1_dis_densityextrapolation_low_fast','mms1_dis_densityextrapolation_high_fast','mms1_dis_bulkv_db
cs_fast','mms1_dis_bulkv_dbcs_fast_noerr','mms1_dis_bulkv_spintone_dbcs_fast','mms1_dis_bulkv_gse_fast'
,'mms1_dis_bulkv_spintone_gse_fast','mms1_dis_bulkv_gse_fast_noerr','mms1_dis_bulkv_spin_dbcs_fast','m
ms1_dis_bulkv_spin_gse_fast','mms1_dis_prestensor_dbcs_fast','mms1_dis_prestensor_gse_fast','mms1_dis_
pres_bg_fast','mms1_dis_temptensor_dbcs_fast','mms1_dis_temptensor_gse_fast','mms1_dis_heatq_dbcs_fa
st','mms1_dis_heatq_gse_fast','mms1_dis_temppara_fast','mms1_dis_tempperp_fast']
time = ['2023-08-31T21:59:56.000Z', '2023-08-31T23:59:56.000Z']
status, data = cdas.get_data('MMS1_FPI_FAST_L2_DIS-MOMS', vars, time[0], time[1])

# If spacepy was installed
print(data['mms1_dis_errorflags_fast'])
print(data['mms1_dis_errorflags_fast'].attrs)
```

Copy code to clipboard | Download code

More information about using this library is available from the following:

- PyPI description cdasws
- Jupyter Python notebook examples
- Application Programming Interface description API

https://cdaweb.gsfc.nasa.gov/

Accessing / Analyzing data at SPDF using PySPEDAS

- Numerous general examples are available showing how to access and analyze data at SPDF using PySPEDAS:

  https://github.com/spedas/pyspedas_examples

- Even more examples can be found in mission-specific repositories, e.g.:

  https://github.com/spedas/mms-examples

- Many missions in PySPEDAS access the data from SPDF by default, and others (e.g., MMS) support SPDF access via a keyword (spdf=True) in the load routines

- PySPEDAS also supports loading data from CDAWeb via HAPI master

pyspedas_examples / pyspedas_examples / **notebooks** /

Go to file t | Add file

jameswilburlewis Updated text to reflect current pytplot convention, that times are st... 7cb97cc · 4 months ago History

Name	Last commit message	Last commit date
..		
Cluster_CIS_data_from_CSA.ipynb	Adding example of loading/plotting Cluster CIS data from CSA	9 months ago
Coordinate_transformations_with_OMNI_data.ipynb	adding cotrans notebook with OMNI data	last year
Exploring_the_Heliosphere_with_Python.ipynb	Move basemap install to beginning of notebook (to avoid having to ...	4 months ago
Introduction_to_PyTplot.ipynb	adding pytplot basics notebook	9 months ago
PySPEDAS_PyTplot_for_Solar_Physicists.ipynb	Adding 2 new notebooks	last year
PySPEDAS_PyTplot_timeseries_data.ipynb	minor update	9 months ago
PySPEDAS_loading_data_from_HAPI_servers.ipynb	Adding notebooks created at the PyHC spring hackathon	last year
PyTplot_annotations.ipynb	Adding examples showing greek letters, superscripts and subscripts	6 months ago
PyTplot_error_bars.ipynb	Install pytplot from pypi rather than github zip file	4 months ago
PyTplot_highlight_intervals_and_vertical_bars.ipynb	Install pytplot from pypi rather than github zip file	4 months ago
PyTplot_legend_options.ipynb	Install pytplot from pypi rather than github zip file	4 months ago
PyTplot_markers_and_symbols.ipynb	Install pytplot from pypi rather than github zip file	4 months ago
PyTplot_pseudo_variables.ipynb	Adding two new PyTplot notebooks	last year
PyTplot_range_options.ipynb	Adding two new PyTplot notebooks	last year
PyTplot_spectrogram_options.ipynb	Install pytplot from pypi rather than github zip file	4 months ago
RBSP_RBSPICE_examples.ipynb	Adding RBSP RBSPICE examples	5 months ago
Swarm_data_in_PySPEDAS.ipynb	Updated with the latest version of PySPEDAS/PyTplot	last year

https://github.com/spedas/pyspedas_examples

To see the available datasets, set the `catalog` keyword:

In [5]:

```
pyspedas.hapi.hapi(trange=['2015-10-16', '2015-10-17'],
                   server='https://cdaweb.gsfc.nasa.gov/hapi',
                   catalog=True)
```

```
18-May-22 16:46:24: /usr/local/lib/python3.7/dist-packages/urllib3/connectionpool.py:847: InsecureRequestWarnin
g: Unverified HTTPS request is being made. Adding certificate verification is strongly advised. See: https://url
lib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
  InsecureRequestWarning)
```

```
Available datasets:
A1_K0_MPA
A2_K0_MPA
AC_AT_DEF
AC_H0_MFI
AC_H0_SWE
AC_H1_EPM
AC_H1_MFI
AC_H1_SIS
AC_H2_CRIS
AC_H2_EPM
AC_H2_MFI
AC_H2_SEP
AC_H2_SIS
AC_H2_SWE
AC_H2_SWI
AC_H2_ULE
AC_H3_CRIS
AC_H3_EPM
AC_H3_MFI
AC_H3_SW2
AC_H3_SWI
AC_H4_SWI
AC_H5_SWI
AC_H6_SWI
AC_K0_EPM
AC_K0_MFI
AC_K0_SIS
AC_K0_SWE
AC_K1_EPM
AC_K1_MFI
AC_K1_SWE
AC_K2_MFI
AC_OR_DEF
AC_OR_SSC
AEROCUBE-6-A_DOSIMETER_L2
AEROCUBE-6-B_DOSIMETER_L2
ALOUETTE2_AV_LIM
```

https://github.com/spedas/pyspedas_examples/blob/master/pyspedas_examples/notebooks/PySPEDAS_loading_data_from_HAPI_servers.ipynb master

mms-examples / advanced
/ Generate_2D_slices_of_FPI_and_HPCA_data.ipynb

Go to file

ericthewizard Adding notebook showing 2D slices of FPI and HPCA distribution functi... 9aa48dc · last year History

Preview Code Blame 1872 lines (1872 loc) · 292 KB Raw

This notebook shows how to create 2D slices of 3D particle data from FPI and HPCA using PySPEDAS

Note: these routines are still being beta tested; please report potential issues to egrimes@igpp.ucla.edu

In [ ]:
```
!pip install pyspedas
```

In [2]:
```
from pyspedas.mms.particles.mms_part_slice2d import mms_part_slice2d
```

Bi-directional field-aligned beam of 0-300 eV ions observed by FPI and HPCA

In [3]:
```
time = '2017-09-10/09:32:20'
```

FPI ions with geometric interpolation

The data are rotated such that the x axis is parallel to B field and the bulk velocity defines the x-y plane, and plotted using geometric interpolation (each point on the plot is given the value of the bin it intersects)

```
mms_part_slice2d(time=time, instrument='fpi', species='i', rotation='bv', erange=[0, 300])
```

```
SDC username (blank for public access):
30-Jun-22 17:24:14: Downloading mms1_fpi_fast_l2_dis-dist_20170910080000_v3.3.0.cdf to pydata/mms1/fpi/fast/l2/d
is-dist/2017/09
Time clip was applied to: mms1_dis_errorflags_fast
Time clip was applied to: mms1_dis_startdelphi_count_fast
Time clip was applied to: mms1_dis_startdelphi_angle_fast
Time clip was applied to: mms1_dis_dist_fast
Time clip was applied to: mms1_dis_disterr_fast
Time clip was applied to: mms1_dis_energy_fast
Time clip was applied to: mms1_dis_energy_delta_fast
That name is currently not in pytplot
That name is currently not in pytplot
That name is currently not in pytplot
```

Note: some of the analysis tools, e.g., MMS particle tools, also support the `spdf` keyword, to load the data from SPDF instead of the MMS SDC

https://github.com/spedas/mms-examples/blob/master/advanced/Generate_2D_slices_of_FPI_and_HPCA_data.ipynb