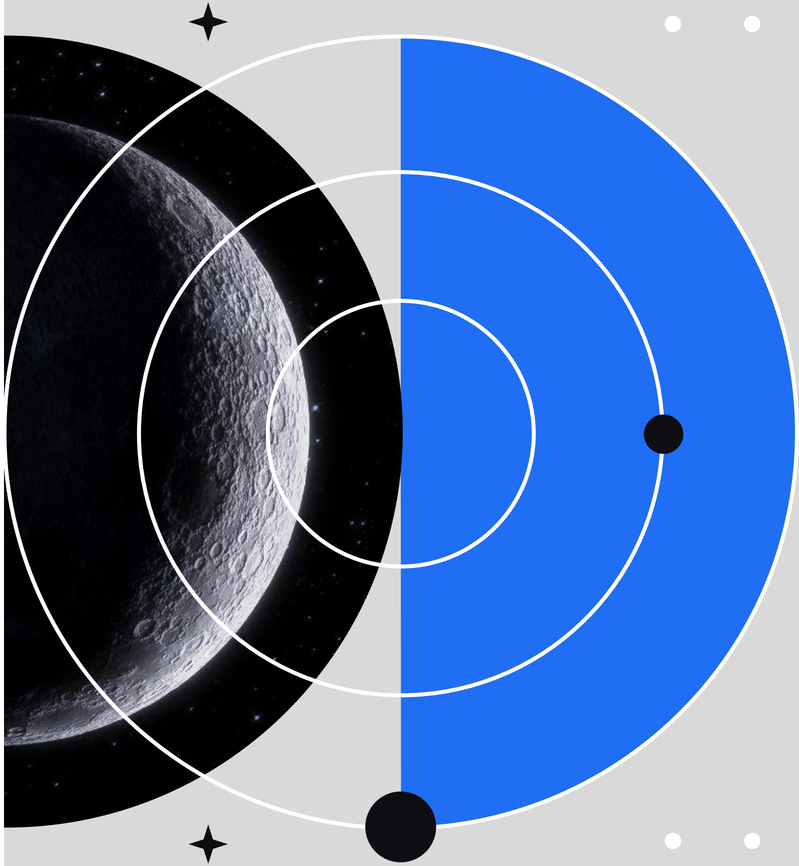


Computational tools for processing data from THEMIS instruments

Prepared by Saniya Danenova

TABLE OF CONTENTS

01 Problem Statement

02 Computational Tools

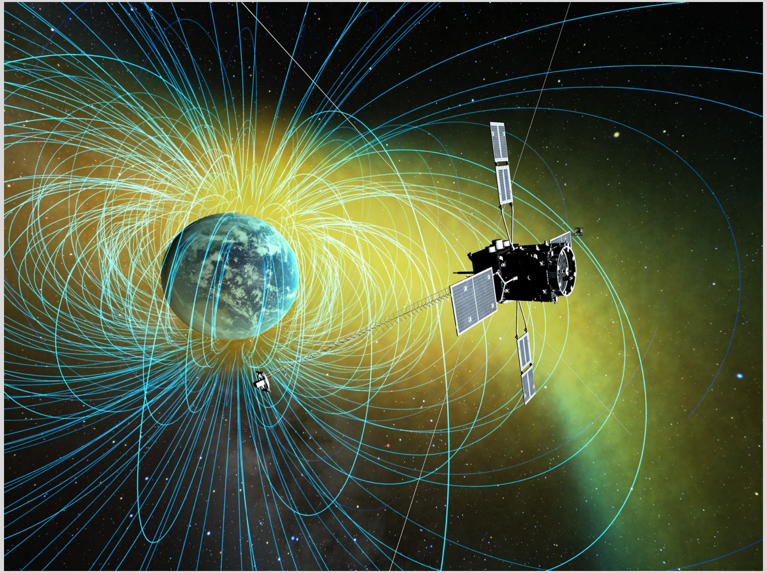
03 Demonstration of how tools work

04 Future Development

Brief Introduction

THEMIS mission
THA, THD, THE satellite
Data from measuring instruments:

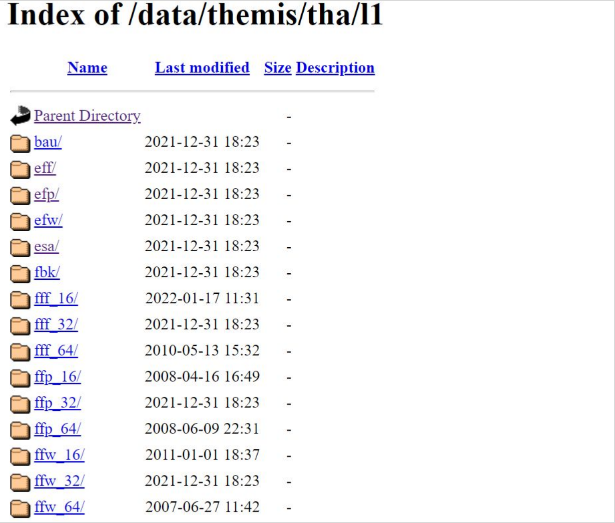
- Electric field instruments (EFI)
- Fluxgate magnetometers (FGM)
- Search coil magnetometers (SCM)
- Electrostatic analyzers (ESA)
- Solid state telescopes (SST)



Problem Statement

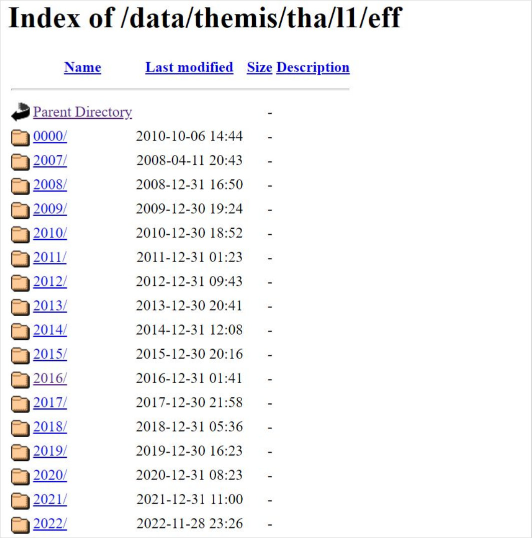
- How to retrieve datafiles more effectively?
- How to easily determine the event of plasma injection?
- How to manipulate the data effectively and perform computations for further analysis?
- How to increase the accuracy of the computations?

Retrieving DataFiles



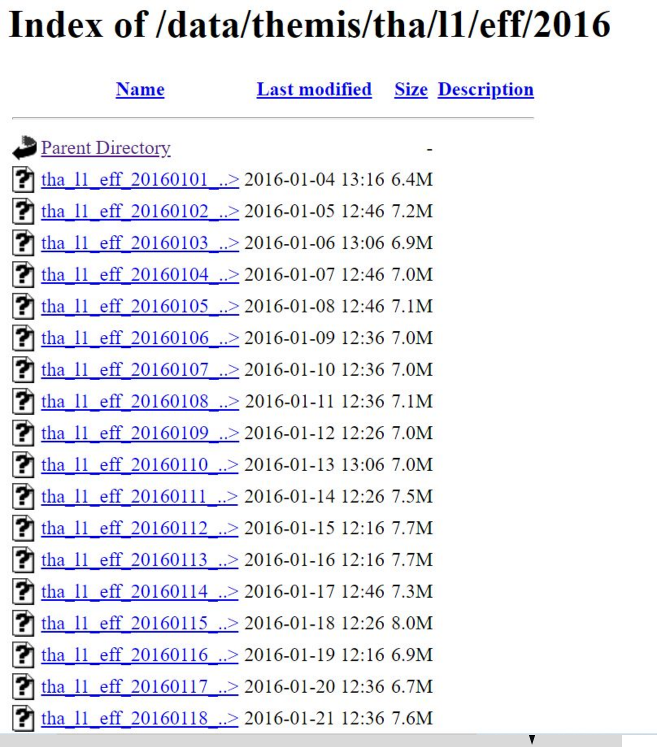
Index of /data/themis/tha/l1

Name	Last modified	Size	Description
Parent Directory		-	
bau/	2021-12-31 18:23	-	
eff/	2021-12-31 18:23	-	
efp/	2021-12-31 18:23	-	
efw/	2021-12-31 18:23	-	
esa/	2021-12-31 18:23	-	
fbk/	2021-12-31 18:23	-	
fff_16/	2022-01-17 11:31	-	
fff_32/	2021-12-31 18:23	-	
fff_64/	2010-05-13 15:32	-	
ffp_16/	2008-04-16 16:49	-	
ffp_32/	2021-12-31 18:23	-	
ffp_64/	2008-06-09 22:31	-	
ffw_16/	2011-01-01 18:37	-	
ffw_32/	2021-12-31 18:23	-	
ffw_64/	2007-06-27 11:42	-	



Index of /data/themis/tha/l1/eff

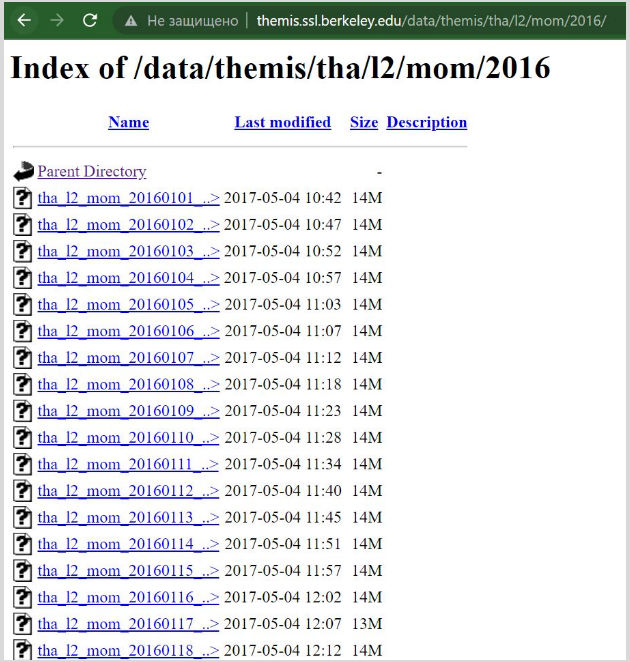
Name	Last modified	Size	Description
Parent Directory		-	
0000/	2010-10-06 14:44	-	
2007/	2008-04-11 20:43	-	
2008/	2008-12-31 16:50	-	
2009/	2009-12-30 19:24	-	
2010/	2010-12-30 18:52	-	
2011/	2011-12-31 01:23	-	
2012/	2012-12-31 09:43	-	
2013/	2013-12-30 20:41	-	
2014/	2014-12-31 12:08	-	
2015/	2015-12-30 20:16	-	
2016/	2016-12-31 01:41	-	
2017/	2017-12-30 21:58	-	
2018/	2018-12-31 05:36	-	
2019/	2019-12-30 16:23	-	
2020/	2020-12-31 08:23	-	
2021/	2021-12-31 11:00	-	
2022/	2022-11-28 23:26	-	



Index of /data/themis/tha/l1/eff/2016

Name	Last modified	Size	Description
Parent Directory		-	
tha_l1_eff_20160101_..>	2016-01-04 13:16	6.4M	
tha_l1_eff_20160102_..>	2016-01-05 12:46	7.2M	
tha_l1_eff_20160103_..>	2016-01-06 13:06	6.9M	
tha_l1_eff_20160104_..>	2016-01-07 12:46	7.0M	
tha_l1_eff_20160105_..>	2016-01-08 12:46	7.1M	
tha_l1_eff_20160106_..>	2016-01-09 12:36	7.0M	
tha_l1_eff_20160107_..>	2016-01-10 12:36	7.0M	
tha_l1_eff_20160108_..>	2016-01-11 12:36	7.1M	
tha_l1_eff_20160109_..>	2016-01-12 12:26	7.0M	
tha_l1_eff_20160110_..>	2016-01-13 13:06	7.0M	
tha_l1_eff_20160111_..>	2016-01-14 12:26	7.5M	
tha_l1_eff_20160112_..>	2016-01-15 12:16	7.7M	
tha_l1_eff_20160113_..>	2016-01-16 12:16	7.7M	
tha_l1_eff_20160114_..>	2016-01-17 12:46	7.3M	
tha_l1_eff_20160115_..>	2016-01-18 12:26	8.0M	
tha_l1_eff_20160116_..>	2016-01-19 12:16	6.9M	
tha_l1_eff_20160117_..>	2016-01-20 12:36	6.7M	
tha_l1_eff_20160118_..>	2016-01-21 12:36	7.6M	

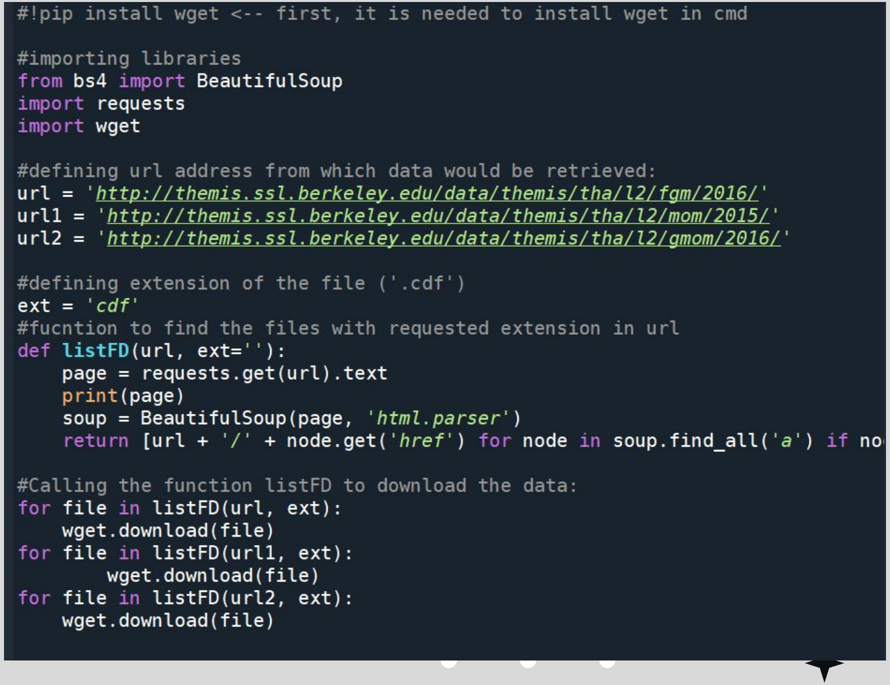
Retrieving DataFiles



Не защищено | themis.ssl.berkeley.edu/data/themis/tha/l2/mom/2016/

Index of /data/themis/tha/l2/mom/2016

Name	Last modified	Size	Description
Parent Directory		-	
tha_l2_mom_20160101_..>	2017-05-04 10:42	14M	
tha_l2_mom_20160102_..>	2017-05-04 10:47	14M	
tha_l2_mom_20160103_..>	2017-05-04 10:52	14M	
tha_l2_mom_20160104_..>	2017-05-04 10:57	14M	
tha_l2_mom_20160105_..>	2017-05-04 11:03	14M	
tha_l2_mom_20160106_..>	2017-05-04 11:07	14M	
tha_l2_mom_20160107_..>	2017-05-04 11:12	14M	
tha_l2_mom_20160108_..>	2017-05-04 11:18	14M	
tha_l2_mom_20160109_..>	2017-05-04 11:23	14M	
tha_l2_mom_20160110_..>	2017-05-04 11:28	14M	
tha_l2_mom_20160111_..>	2017-05-04 11:34	14M	
tha_l2_mom_20160112_..>	2017-05-04 11:40	14M	
tha_l2_mom_20160113_..>	2017-05-04 11:45	14M	
tha_l2_mom_20160114_..>	2017-05-04 11:51	14M	
tha_l2_mom_20160115_..>	2017-05-04 11:57	14M	
tha_l2_mom_20160116_..>	2017-05-04 12:02	14M	
tha_l2_mom_20160117_..>	2017-05-04 12:07	13M	
tha_l2_mom_20160118_..>	2017-05-04 12:12	14M	

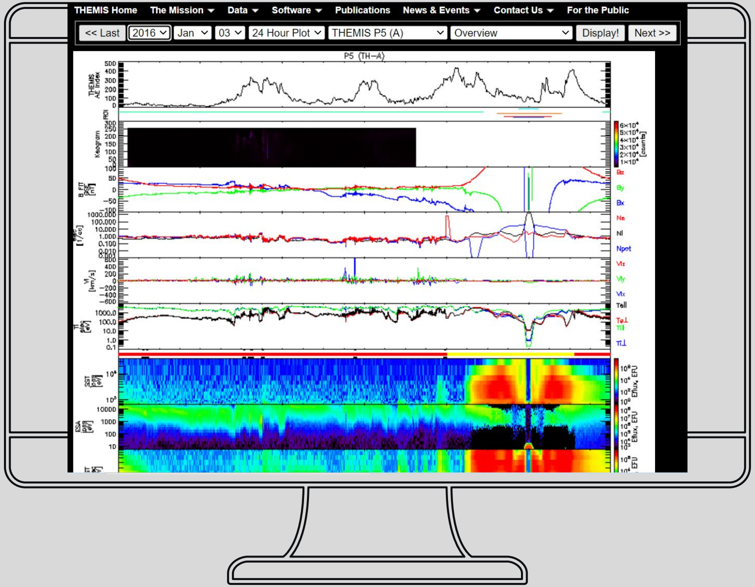


```
#!pip install wget <-- first, it is needed to install wget in cmd

#importing libraries
from bs4 import BeautifulSoup
import requests
import wget

#defining url address from which data would be retrieved:
url = 'http://themis.ssl.berkeley.edu/data/themis/tha/l2/fgm/2016/'
url1 = 'http://themis.ssl.berkeley.edu/data/themis/tha/l2/mom/2015/'
url2 = 'http://themis.ssl.berkeley.edu/data/themis/tha/l2/gmom/2016/'

#defining extension of the file ('.cdf')
ext = 'cdf'
#fucntion to find the files with requested extension in url
def listFD(url, ext=''):
    page = requests.get(url).text
    print(page)
    soup = BeautifulSoup(page, 'html.parser')
    return [url + '/' + node.get('href') for node in soup.find_all('a') if no

#Calling the function listFD to download the data:
for file in listFD(url, ext):
    wget.download(file)
for file in listFD(url1, ext):
        wget.download(file)
for file in listFD(url2, ext):
    wget.download(file)
```

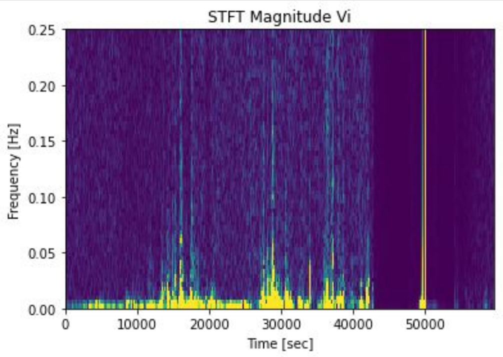
Identifying Plasma Injection events

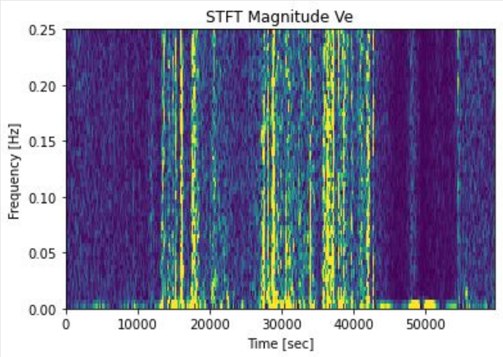
Criteria features for plasma injection events:

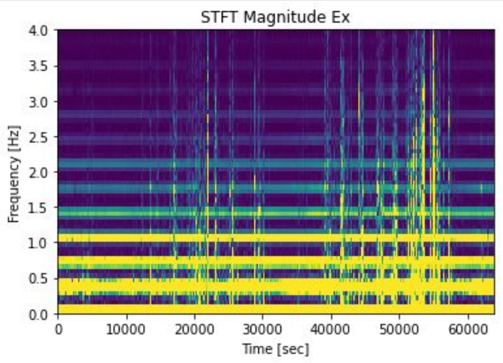
- ★ sharp increase in Bz, along with sharp decrease in Bx
- ★ increase in the velocity values for Vx
- ★ sharp increase in the activity of Electric field

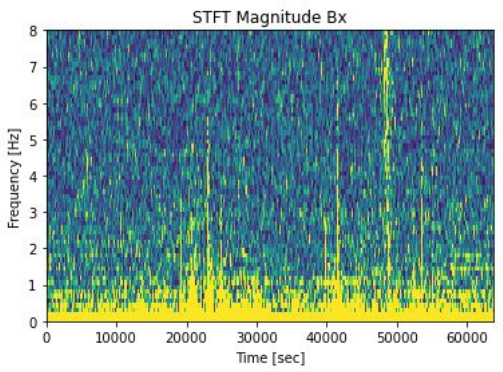


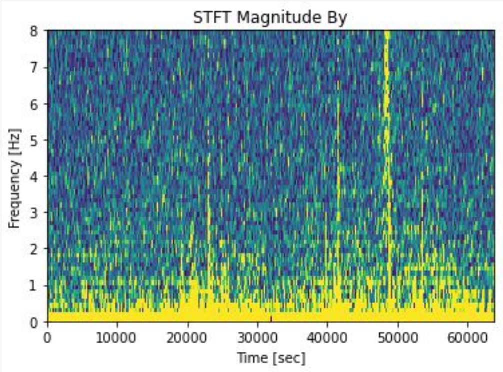
Performing FFT on data and facing new challenges

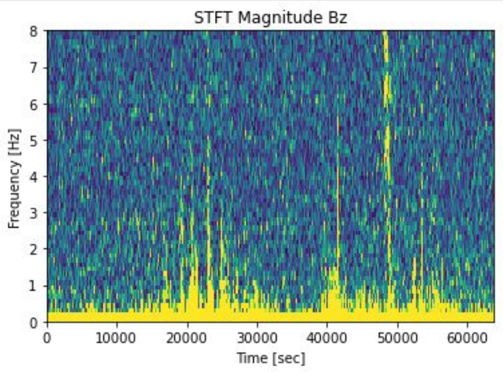




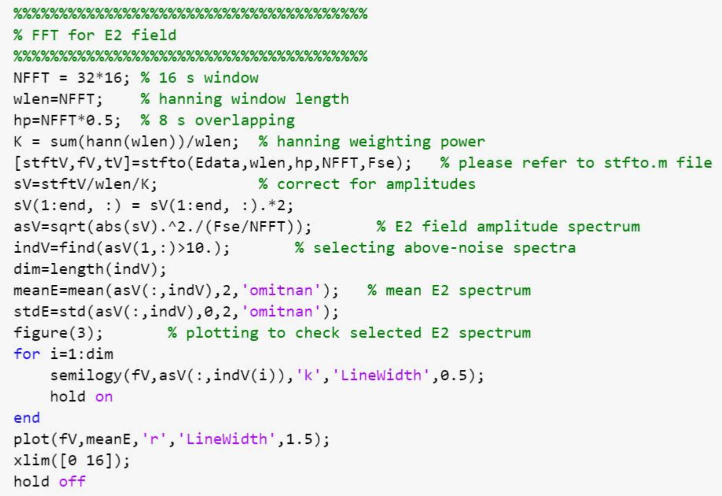




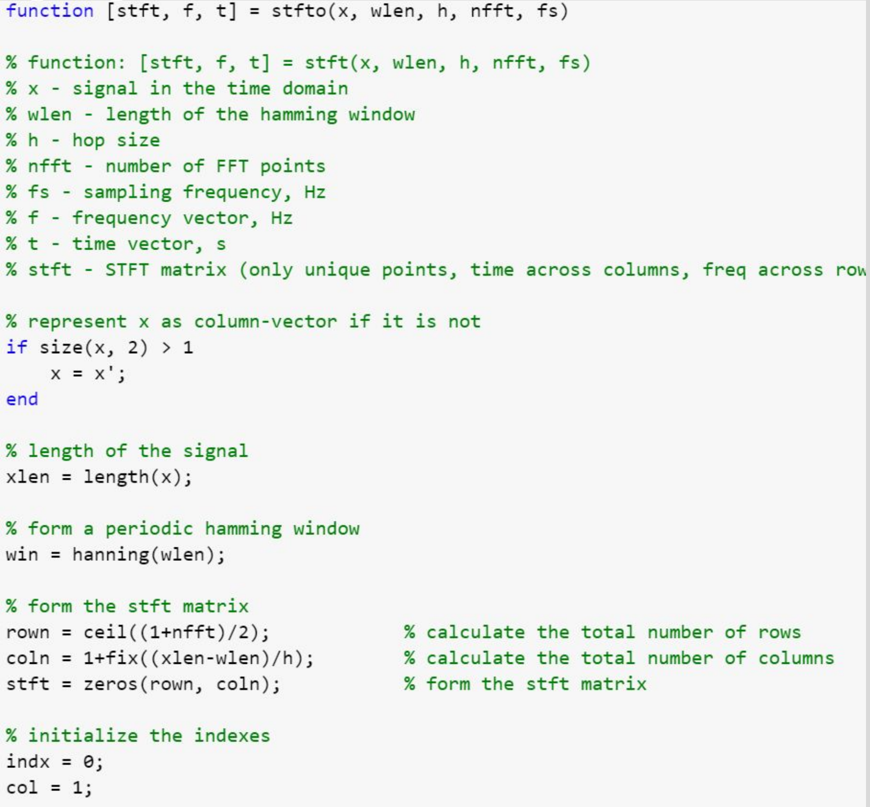




Evaluating the method with subroutine function

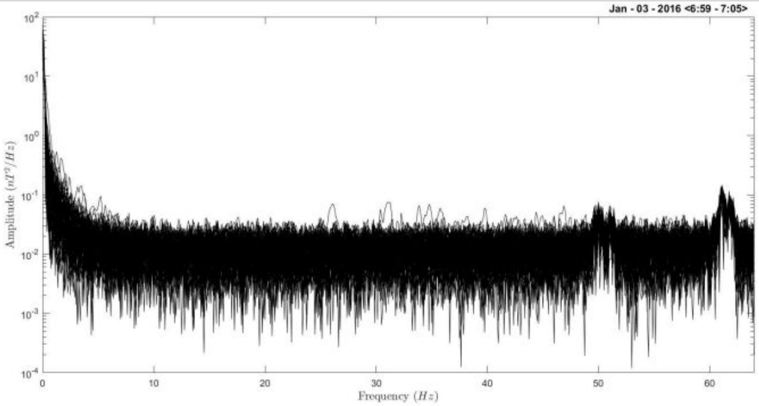


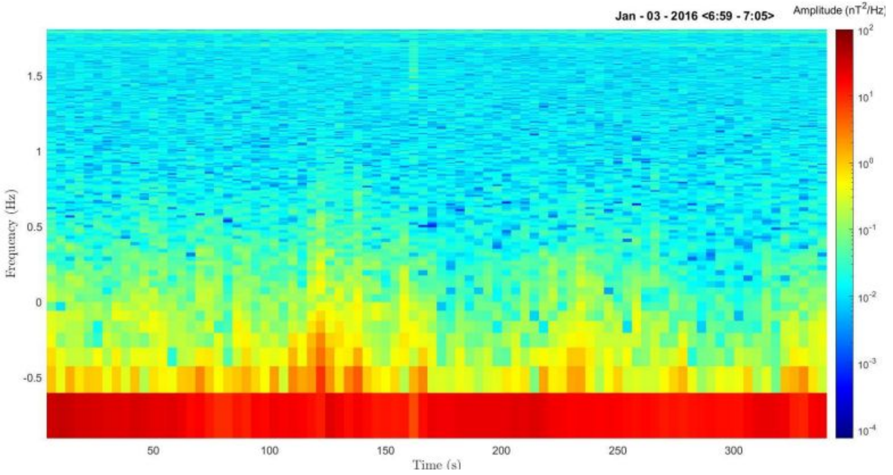
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT for E2 field
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
NFFT = 32*16; % 16 s window
wlen=NFFT;    % hanning window length
hp=NFFT*0.5;  % 8 s overlapping
K = sum(hann(wlen))/wlen;  % hanning weighting power
[stftV,fV,tV]=stfto(Edata,wlen,hp,NFFT,Fse);   % please refer to stfto.m file
sV=stftV/wlen/K;           % correct for amplitudes
sV(1:end, :) = sV(1:end, :).*2;
asV=sqrt(abs(sV).^2./(Fse/NFFT));       % E2 field amplitude spectrum
indV=find(asV(1,:)>10.);       % selecting above-noise spectra
dim=length(indV);
meanE=mean(asV(:,indV),2,'omitnan');   % mean E2 spectrum
stdE=std(asV(:,indV),0,2,'omitnan');
figure(3);       % plotting to check selected E2 spectrum
for i=1:dim
    semilogy(fV,asV(:,indV(i)),'k','LineWidth',0.5);
    hold on
end
plot(fV,meanE,'r','LineWidth',1.5);
xlim([0 16]);
hold off
```



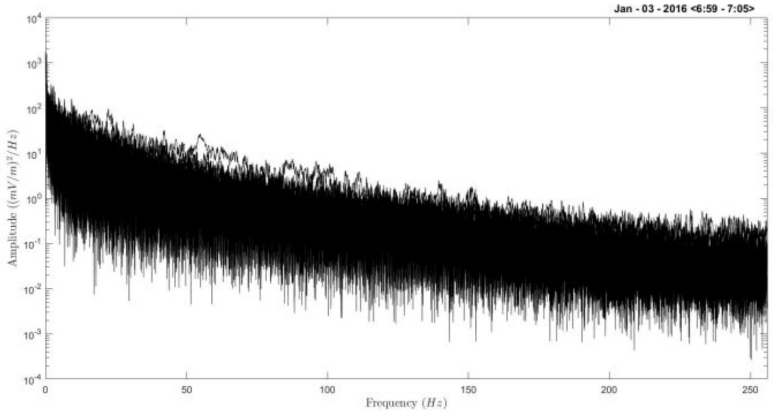
```
function [stft, f, t] = stfto(x, wlen, h, nfft, fs)

% function: [stft, f, t] = stft(x, wlen, h, nfft, fs)
% x - signal in the time domain
% wlen - length of the hamming window
% h - hop size
% nfft - number of FFT points
% fs - sampling frequency, Hz
% f - frequency vector, Hz
% t - time vector, s
% stft - STFT matrix (only unique points, time across columns, freq across row

% represent x as column-vector if it is not
if size(x, 2) > 1
    x = x';
end

% length of the signal
xlen = length(x);

% form a periodic hamming window
win = hanning(wlen);

% form the stft matrix
rown = ceil((1+nfft)/2);           % calculate the total number of rows
coln = 1+fix((xlen-wlen)/h);       % calculate the total number of columns
stft = zeros(rown, coln);          % form the stft matrix

% initialize the indexes
indx = 0;
col = 1;
```

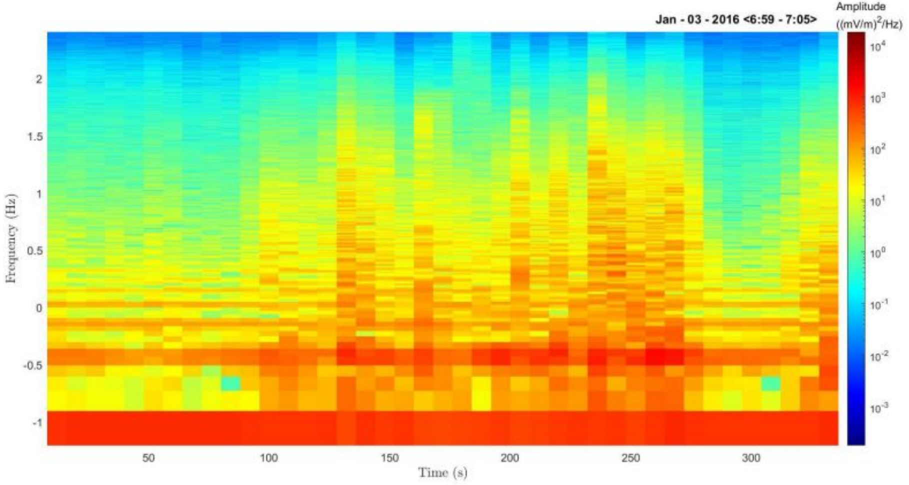
BField FFT





EField FFT





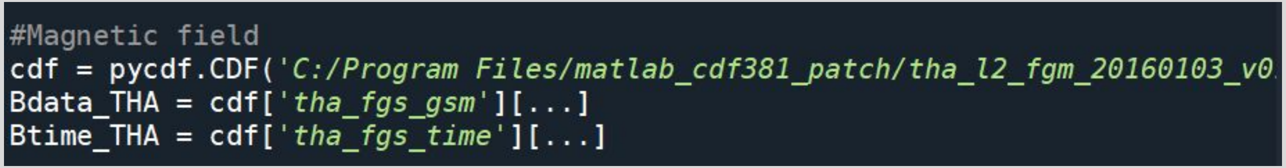
Analysis of Single Plasma injection Event

1 Entering Data (‘.cdf’)

2 Interpolating Date Series

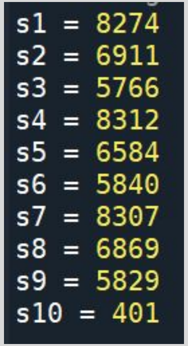
3 Applying equations and performing calculations

Entering and loading data

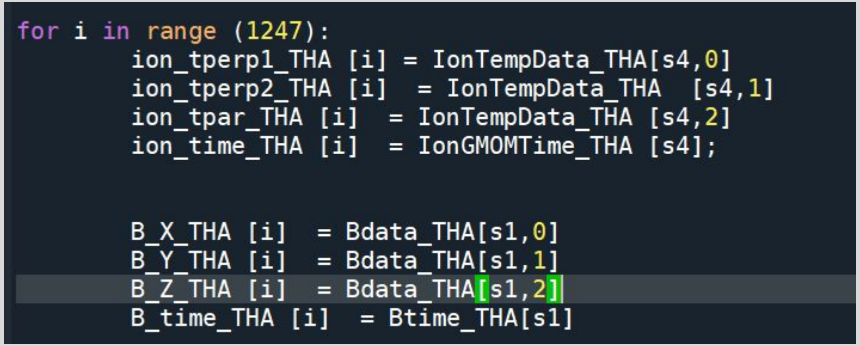


```
#Magnetic field
cdf = pycdf.CDF('C:/Program Files/matlab_cdf381_patch/tha_l2_fgm_20160103_v0.
Bdata_THA = cdf['tha_fgs_gsm'][...]
Btime_THA = cdf['tha_fgs_time'][...]
```

Entering time constraints

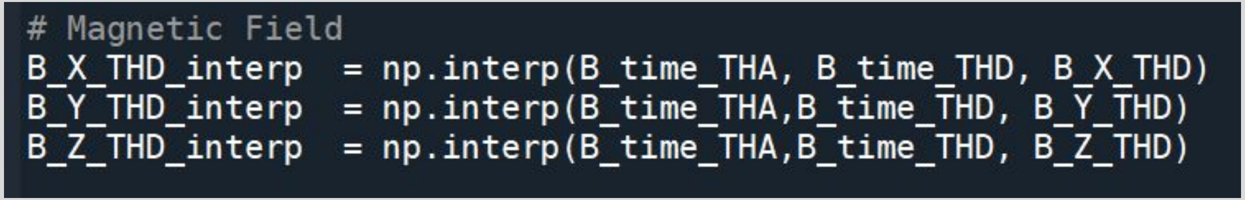


```
s1 = 8274
s2 = 6911
s3 = 5766
s4 = 8312
s5 = 6584
s6 = 5840
s7 = 8307
s8 = 6869
s9 = 5829
s10 = 401
```

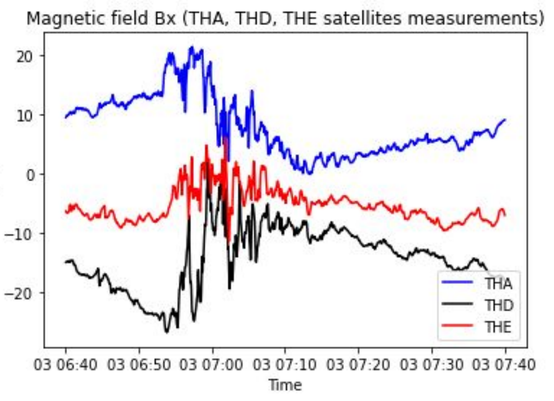


```
for i in range (1247):
        ion_tperp1_THA [i] = IonTempData_THA[s4,0]
        ion_tperp2_THA [i]  = IonTempData_THA  [s4,1]
        ion_tpar_THA [i]  = IonTempData_THA [s4,2]
        ion_time_THA [i]  = IonGMOMTime_THA [s4];


        B_X_THA [i]  = Bdata_THA[s1,0]
        B_Y_THA [i]  = Bdata_THA[s1,1]
        B_Z_THA [i]  = Bdata_THA[s1,2]
        B_time_THA [i]  = Btime_THA[s1]
```

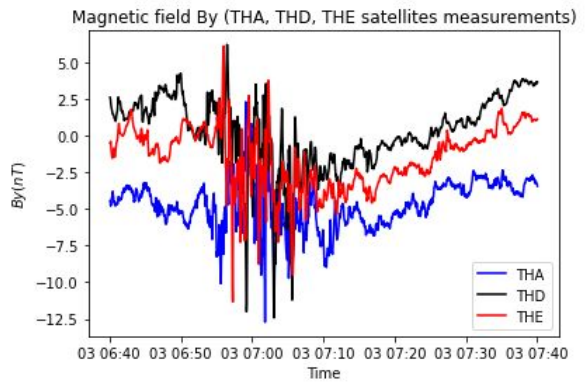
Performing Time Series interpolation



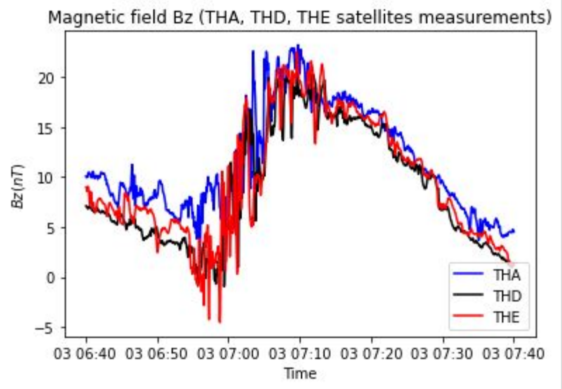
```
# Magnetic Field
B_X_THD_interp  = np.interp(B_time_THA, B_time_THD, B_X_THD)
B_Y_THD_interp  = np.interp(B_time_THA,B_time_THD, B_Y_THD)
B_Z_THD_interp  = np.interp(B_time_THA,B_time_THD, B_Z_THD)
```



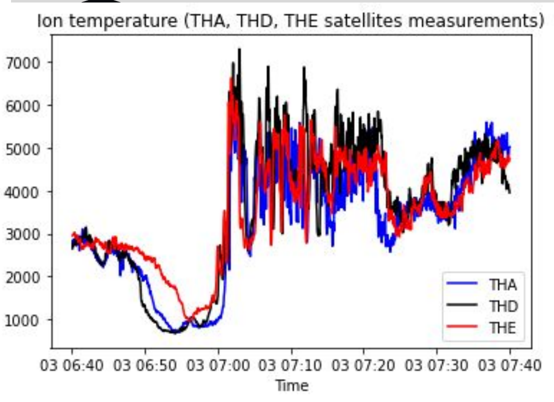
Magnetic field Bx (THA, THD, THE satellites measurements)



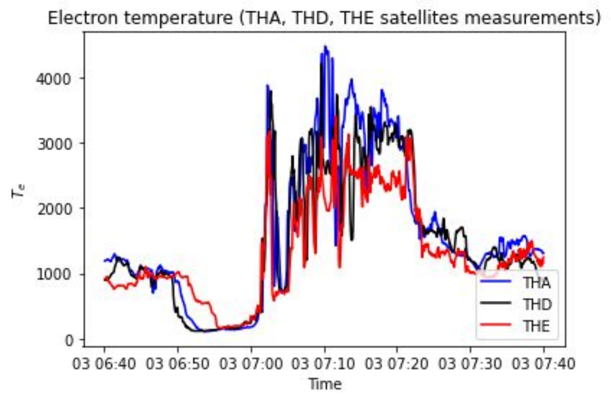
Magnetic field By (THA, THD, THE satellites measurements)



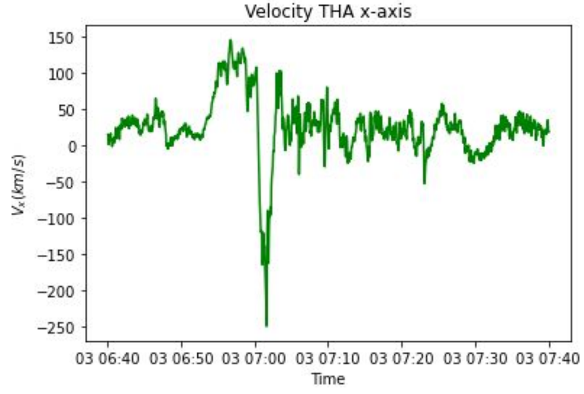
Magnetic field Bz (THA, THD, THE satellites measurements)



Ion temperature (THA, THD, THE satellites measurements)

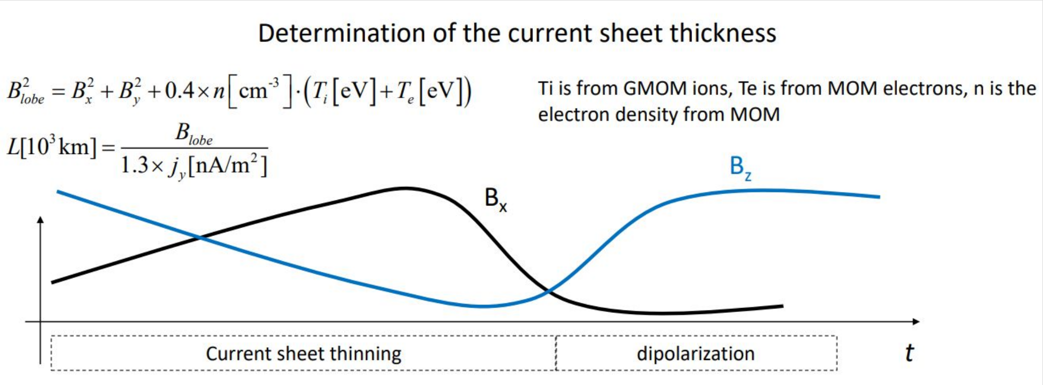


Electron temperature (THA, THD, THE satellites measurements)



Velocity THA x-axis

Equations for calculations



Determination of the current sheet thickness

$$B_{lobe}^2 = B_x^2 + B_y^2 + 0.4\times n\left[\mathrm{cm}^{-3}\right]\cdot\left(T_i\left[\mathrm{eV}\right]+T_e\left[\mathrm{eV}\right]\right)$$

Ti is from GMOM ions, Te is from MOM electrons, n is the electron density from MOM

$$L[10^3\,\mathrm{km}] = \frac{B_{lobe}}{1.3\times j_y[\mathrm{nA/m^2}]}$$

$$B_x^A - B_x^D \approx \frac{\partial B_x}{\partial x}dx^{AD} + \frac{\partial B_x}{\partial z}dz^{AD}$$

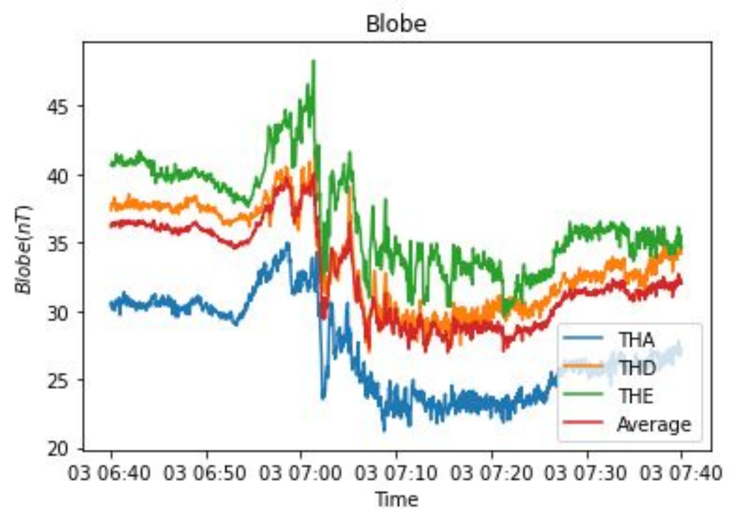
$$B_x^A - B_x^E \approx \frac{\partial B_x}{\partial x}dx^{AE} + \frac{\partial B_x}{\partial z}dz^{AE}$$

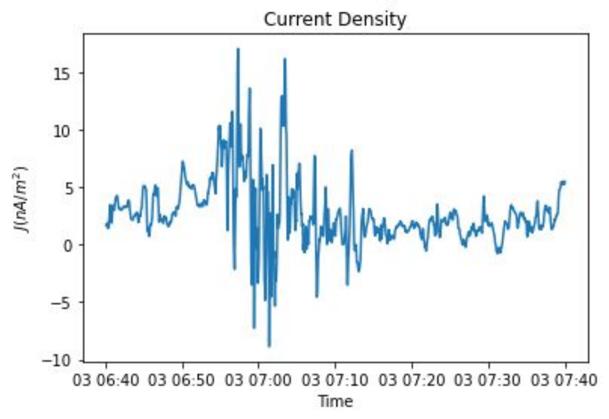
$$B_z^A - B_z^D \approx \frac{\partial B_z}{\partial x}dx^{AD} + \frac{\partial B_z}{\partial z}dz^{AD}$$

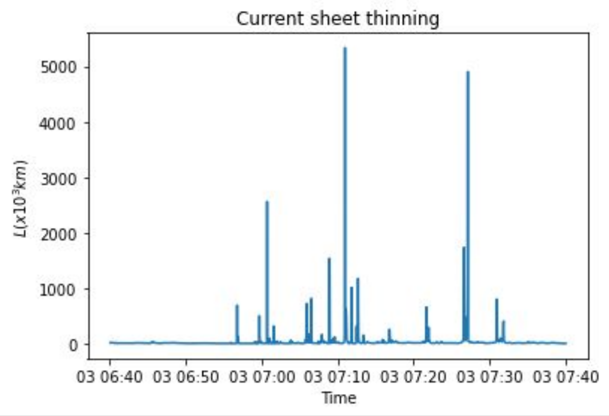
$$B_z^A - B_z^E \approx \frac{\partial B_z}{\partial x}dx^{AE} + \frac{\partial B_z}{\partial z}dz^{AE}$$

$$J = \bar{\nabla}\times\bar{B} = \hat{j}\,\left(\frac{\partial B_x}{\partial z} - \frac{\partial B_z}{\partial x}\right)$$

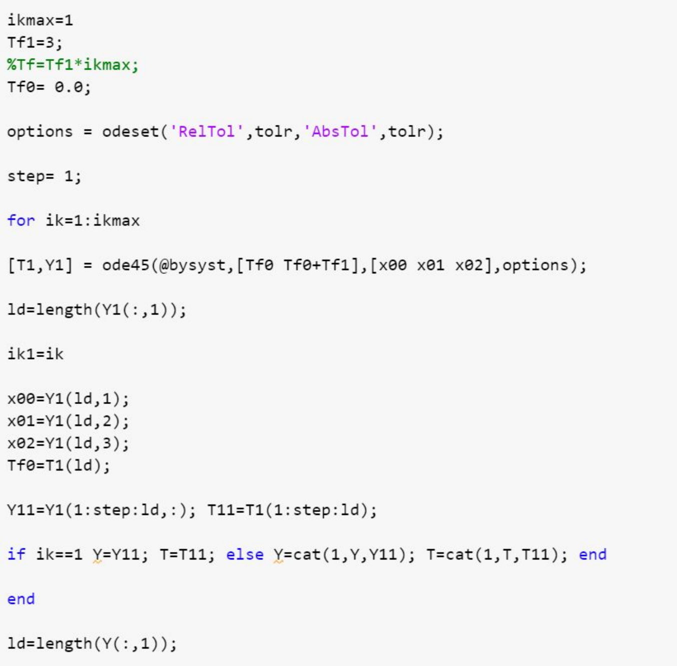
Results from calculations



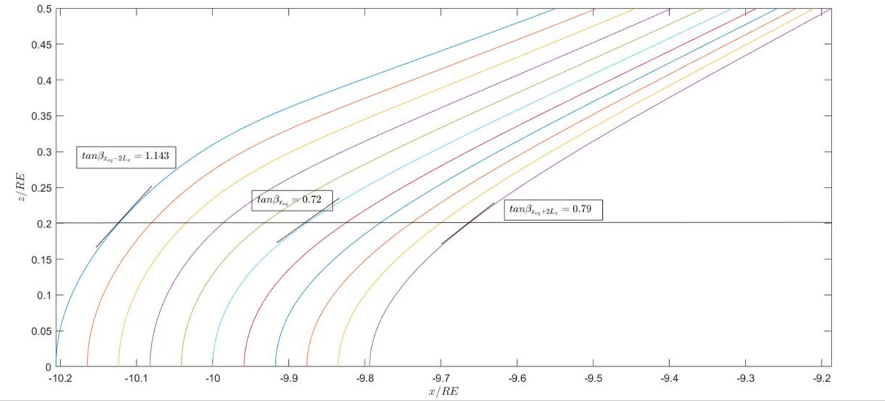


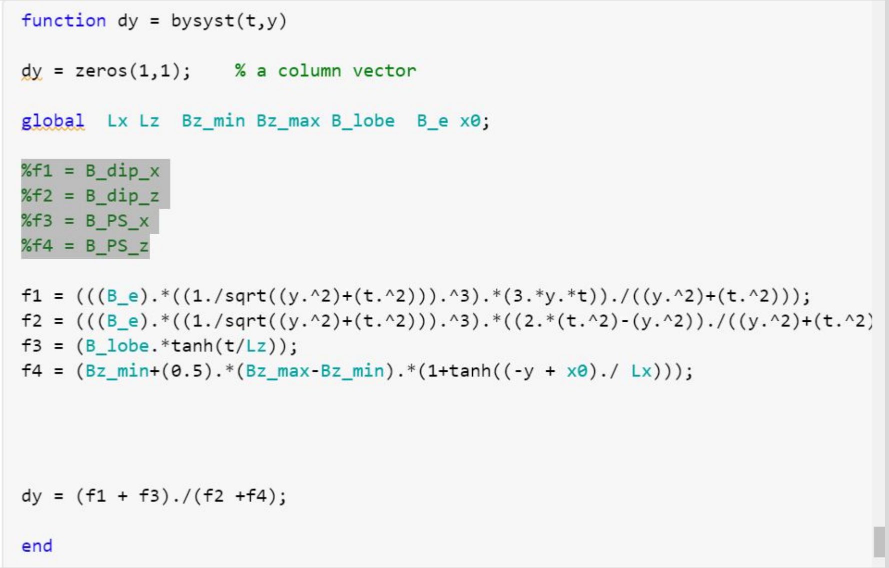


Computational Tool to solve ODE and and visualise magnetic field lines



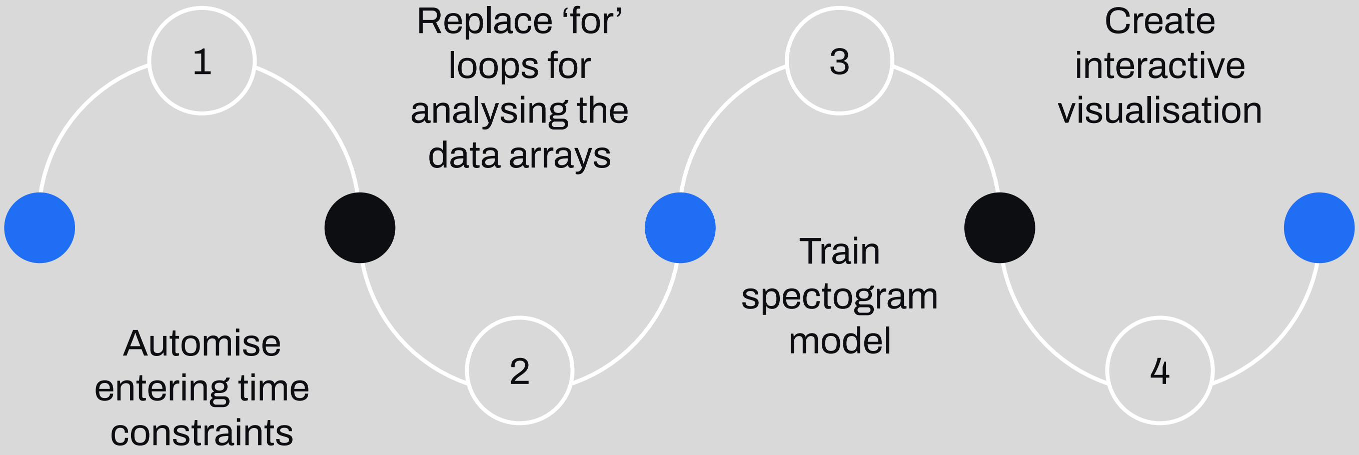
```
ikmax=1
Tf1=3;
%Tf=Tf1*ikmax;
Tf0= 0.0;

options = odeset('RelTol',tolr,'AbsTol',tolr);

step= 1;

for ik=1:ikmax

[T1,Y1] = ode45(@bysyst,[Tf0 Tf0+Tf1],[x00 x01 x02],options);

ld=length(Y1(:,1));

ik1=ik

x00=Y1(ld,1);
x01=Y1(ld,2);
x02=Y1(ld,3);
Tf0=T1(ld);

Y11=Y1(1:step:ld,:); T11=T1(1:step:ld);

if ik==1 Y=Y11; T=T11; else Y=cat(1,Y,Y11); T=cat(1,T,T11); end

end

ld=length(Y(:,1));
```





```
function dy = bysyst(t,y)

dy = zeros(1,1);    % a column vector

global  Lx Lz  Bz_min Bz_max B_lobe  B_e x0;

%f1 = B_dip_x
%f2 = B_dip_z
%f3 = B_PS_x
%f4 = B_PS_z

f1 = (((B_e).*((1./sqrt((y.^2)+(t.^2))).^3).*(3.*y.*t))./((y.^2)+(t.^2)));
f2 = (((B_e).*((1./sqrt((y.^2)+(t.^2))).^3).*((2.*(t.^2)-(y.^2))./((y.^2)+(t.^2)
f3 = (B_lobe.*tanh(t/Lz));
f4 = (Bz_min+(0.5).*(Bz_max-Bz_min).*(1+tanh((-y + x0)./ Lx)));




dy = (f1 + f3)./(f2 +f4);

end
```

Future Development Roadmap



1

Automise entering time constraints

2

Replace ‘for’ loops for analysing the data arrays

3

Train spectogram model

4

Create interactive visualisation

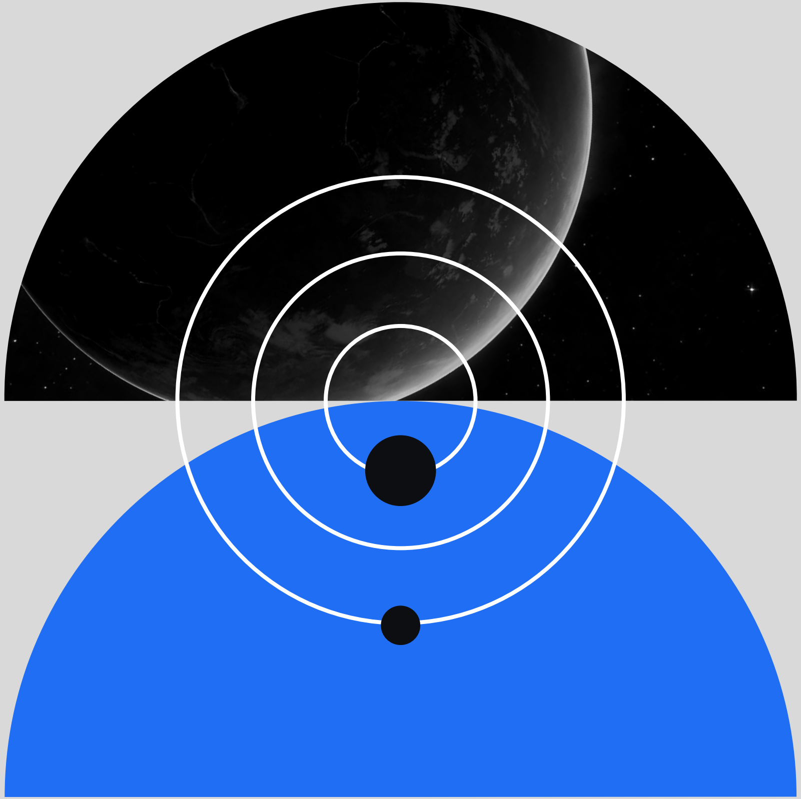
Citations

Themis. Accessed November 29, 2022. http://themis.ssl.berkeley.edu/summary.php?year=2016&month=01&day=03&hour=0024&sumType=tha&type=overview.

“Cdflib¶.” cdflib. Accessed November 29, 2022. https://cdflib.readthedocs.io/en/latest/.

“How to Download and Upload Files in FTP Server Using Python?” GeeksforGeeks, January 12, 2022. https://www.geeksforgeeks.org/how-to-download-and-upload-files-in-ftp-server-using-python/.

“Pycdf - Python Interface to CDF Files¶.” pycdf - Python interface to CDF files - SpacePy v0.4.1 Manual. Accessed November 29, 2022. https://spacepy.github.io/pycdf.html.



THANKS!