SFE detection program

Gastón Creci Keinbaum

Introduction

An sfe (*solar flare effect*), also known as *crochet*, is a daylight geomagnetic variation produced by a chromospheric eruption. The main characteristics that distinguish sfes from other magnetic variations are (*Parkinson,* 1983):

1. Fast growing and an exponential return to the normal value. The complete phenomena last less than an hour.
2. It occurs simultaneously in the entire illuminated hemisphere.
3. It's an increment of the diurnal variation wherever it occurs.
4. Its magnitude does not increase in aural zones.
5. Start with exact simultaneity with the vision of the solar flare activity.
6. A typical value of sfe's amplitude at medium latitudes is about 10 to 20 nT.

The program, based on those characteristics, has the goal of recognize an sfe.

1. -Theoretical basis

To recognize sfes the program is based on the variation of the X and Y components of the magnetic field vector. The electric current circuit in the ionosphere that produces the magnetic variations has an approximate circular form, i.e. quasi-elliptic. Due that the magnetic field vector has a 90º phase respect the electric current vector, the northern and southern observatories vary at the X component and the eastern and western ones vary at the Y component in a symmetric way as shown at Fig 1.
[Introduir figura 1]

2. - Metodology

To know the sfe focus position I have used the data from *Curto* (1995). That data gives an approximate position of the sfe focus at 12:00 UT depending of the season. The position of the sfe focus changes as subsolar point does, so I have used the position at given hour and make it move simultaneously with subsolar point along each day.

	Sfe focus position		
Latitude	37,0 º	42,0 º	43,1 º
Longitude	-12,6 º	-2,0 º	-4,6 º
Season	Winter	Equinoxes	Summer

Table 1. - Position of the SFE focus at 12:00 UT.

Once determined the sfe focus, I have taken the position of all the observatories of the *Intermagnet* network (144 around the world) and done a rotation from equatorial coordinates to "sfe coordinates", i.e., spherical coordinates which origin is the sfe focus. To make that rotation I've solved the following spherical triangle:
[Imatge del triangle esfèric]
(1.0)

The resolution of the spherical triangle can be done as a rotation of the X-axis:

$$(\varphi,\xi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_{SFE}) & -\sin(\phi_{SFE}) \\ 0 & \sin(\phi_{SFE}) & \cos(\phi_{SFE}) \end{pmatrix} (\lambda,\phi) \qquad (1.1)$$

Where:

$$(\varphi,\xi) = \begin{pmatrix} x = \cos(\varphi)\sin(\xi) \\ y = \sin(\varphi)\sin(\xi) \\ z = \cos(\xi) \end{pmatrix} \text{ and } (\lambda,\phi) = \begin{pmatrix} x' = \sin(\lambda_{SFE} - \lambda)\sin(\phi_P) \\ y' = \cos(\lambda_{SFE} - \lambda)\sin(\phi_P) \\ z' = \cos(\phi_P) \end{pmatrix} \qquad (1.2)$$

Thus:

$$\varphi = \arctan\left(\frac{y}{x}\right) \text{ and } \xi = \arccos(z) \qquad (1.3)$$

Are the longitude and colatitude in sfe coordinates.

Now is possible to take a specific range of colatitude ξ from the focus of the sfe, which allow us to select the observatories of interest.

To classify the observatories in north, south, east and west, I have used the following reasoning:
[Imatge dels increments]

$$\text{North: } \begin{cases} \lambda_{SFE} - \Delta\lambda < \lambda_{OBS} < \lambda_{SFE} + \Delta\lambda \\ \phi_{OBS} < \phi_{SFE} - \Delta\phi \end{cases} \quad \text{South: } \begin{cases} \lambda_{SFE} - \Delta\lambda < \lambda_{OBS} < \lambda_{SFE} + \Delta\lambda \\ \phi_{OBS} > \phi_{SFE} + \Delta\phi \end{cases}$$

$$\text{East: } \begin{cases} \lambda_{OBS} > \lambda_{SFE} + \Delta\lambda \\ \phi_{OBS} - \Delta\phi < \phi_{OBS} < \phi_{OBS} + \Delta\phi \end{cases} \quad \text{West: } \begin{cases} \lambda_{OBS} < \lambda_{SFE} - \Delta\lambda \\ \phi_{OBS} - \Delta\phi < \phi_{OBS} < \phi_{OBS} + \Delta\phi \end{cases}$$

Once determined the classification of the observatories, the program applies the algorithm.

The algorithm, as said, is based on the rate of change of the X and Y components of the magnetic field vector. It takes the derivate of the magnetic field in each component for each observatory in the same location and makes the average value of the data.

Let δ_n be the average value of the "n" cardinal point. The main idea is to multiply each δ_n by a sign function and the peak of the derivate will predominate, letting us know that an sfe may have occur. Theoretically the noise from δ_S will be compensated by the δ_N one, and the same with δ_W and δ_E. Thus:

$$\delta_{TOTAL} = \delta_S - \delta_N + \delta_W - \delta_E$$

δ_{TOTAL} is a function which peak, if is greater than a certain noise value, will indicate a possible sfe.

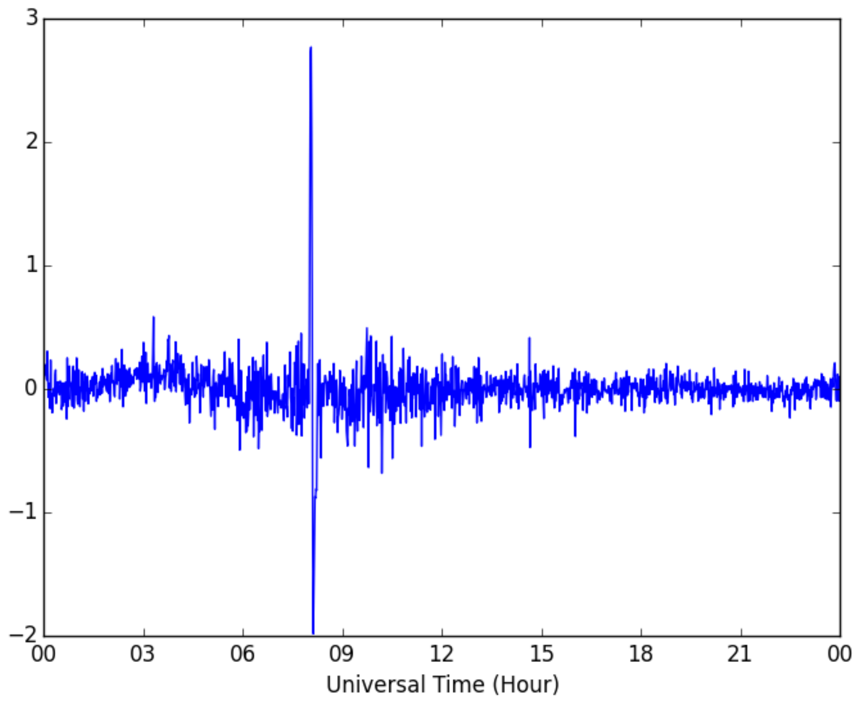

Fig 2. - δ_{TOTAL} representation. The peak indicates an sfe. In this case is an sfe of the day 09/08/2011 at 08:03 UT.

Program

First of all, to run the program we must have the numpy, pylab and pyephem libraries. They can be downloaded for free at:

- http://rhodesmill.org/pyephem/ - Pyephem
- http://www.scipy.org/scipylib/download.html - Numpy
- http://matplotlib.org/users/installing.html - Pylab

The program has been made in python 2.7.

<u>Functions</u>
In this section we are going to take a look at the functions of the program:

- **Open_data(**nom_arxiu**)**
  It takes a data called "nom_arxiu", opens and reads it, returning the desired data in lists. The particularity of this function is that deletes all the string format of the data.

  When there are no measurements, the corresponding point of the data is 99999.0. We use the function "extrapolation(A)" to substitute these points for a new ones that are the medium value of the next and last points.

- **extrapolation(**A**)**
  Takes a list (A) and substitutes the point 99999.0 for the medium value of the next and last point. The possibilities that the last and next point could be 99999.0 are contemplated in this function.

- **Time_funct()**
  This function makes two lists, one consists of 1440 elements, each element is the time in hours at each minute of the day and the other is the same as the first, but in string format (HH:MM:SS.000). These lists are useful for certain operations.

- **plot data(**X,Y,Z,F,T,title,ax1label,ax2label,ax3label,ax4label)
  Makes the plot of the data X,Y,Z,F in function of T with the corresponding labels: ax1label, ax2label,ax3label and ax4label and the title.

- **magnetograma(**X,Y,Z,F,T,NamesObs_str,num)
  It makes the magnetogram of the data X,Y,X,F in funciton of T for the observatory number "num" located at the list NamesObs.

- **derivate(**X,Y,T**)**
  Returns the lists DX,DY with the derivate of X and Y respect T. The list Tg is the list T without the increments of T.

- **subsolar_point(**Date,Timefloat)
  This function uses the pyephem library to know where the subsolar point is and returns the longitude and declination in radians. The arguments are related to internal formats of the pyephem library.

- **Observat()**
  Takes the file "Observatoris.csv" with all the observatories longitudes and declinatons. It returns the lists colatitude, eastlongitudes and NamesObs (with all the observatories names).

- **arguments(**Date,TimeFloat**)**
  Takes the angles and changes them arguments to [0,π].

- **rotation(**phi_pss,longitude,colatitude,longitude_solar)

This function makes the rotation described at the upper section.

- **coord_change(**Date**,**Date_str,TimeFloat**)**
  Makes the change of coordinates using the rotation function.
- **focus_sfe(**Date**,**Date_str,TimeFloat**)**
  It returns the position of the sfe focus at each hour depending of the epoch of the year.
- **rang_colat(**position,colat_sfe_array,NamesObs,rang_colat_min,rang_colat_max)
  Returns the observatories that are in a certain range of colatitude. For a time greater that 15:00 the range is fixed between 20 and 40 degrees because is the optimum range. At this time the subsolar point enters into the ocean and the number of observatories are minimum.

- **rang_colat_long_eq(**position,colat_sfe_array,NamesObs,Time_str,sfe_focus_colat,sfe_focus_long,Date_str,rang_colat_min,rang_colat_max)
  This function uses the rang_colat to place the observatories at them respective cardinal points, returning the lists N_sfe, S_sfe, E_sfe and W_sfe.

- **namesobs(**NamesObs**)**
  Changes the string form of the observatories from *ABS* to abs.

- **Suma_obs(**num,Date_file,Carpeta,NamesObs)
  From an input path it opens the data systematically using the format name of it. It goes along the list ('d','v','p','q') and the observatories names to search for the desired data.

- **SFE_set(**Date_file,Carpeta,Time,Time_str,colat_sfe_array,NamesObs,sfe_focus_colat,sfe_focus_long,Date_str,rang_colat_min,rang_colat_max)
  Applies the algorithm and returns the delta function, the time and the observatories that could have detected the sfe.

- **Suma_magnetograma(**num,Date_file,Carpeta,NamesObs)
  Opens the desired data and makes the magnetogram.

- **SFE_magnetograma(**n,Date_file,Carpeta,Time,Time_str,colat_sfe_array, NamesObs,sfe_focus_colat,sfe_focus_long,Date_str,rang_colat_min,rang_colat_max)
  This function makes the magnetogram of the possible detected sfe. If there’s no data available of that observatory it launches a message.

The other functions such as day(), month(), year() and day_magnetograma() are related to the menu() function. The menu() function makes the menu of the program and, depending of the chosen option, it goes to the day, month, year or day_magnetograma.

The day_magnetograma() function makes the same as day() function but it also returns the plot of the delta function and the magnetograms of the observatories that could have detected the SFE. This function has the aim to prevent the user from false detections. The user can visualize the magnetogram and see if it is or not an sfe.

Upgrades

Here's a list of some of the upgrades and changes that the program needs:

- A better structuration of the functions.
- Solve a problem in the coordinate change that made me to select the cardinal points in the way is described instead of selecting the cardinal points with the longitudes of the SFE focus system.
- A deeper study of the noise of the delta function to have a better accuracy at the detecting process.