UNCLASSIFIED

esac
European Space Astronomy Centre
28691 Villanueva de la Cañada
Madrid, Spain
T +34 91 8131 100
F +34 91 8131 139
www.esa.int

CSA Technical Note

The Curlometer technique: a beginner's guide

Prepared by	Helen Middleton and Arnaud Masson, with thanks to Denise Perrone, Malcolm Dunlop and Stein Haaland
Issue	1.1
Date	13 December 2016
Reference	ESDC-CSA-TN-0001

European Space Agency
Agence Spatiale Europeénne

Contents

1 INTRODUCTION

The curlometer technique is a calculation that can be performed, under certain conditions and circumstances, to directly estimate the current density vector inside the tetrahedron formed by the four spacecraft of Cluster.

This technique enables the estimation of the electric current density in various regions of the magnetosphere including

- magnetopause currents
- currents in flux transfer events
- current at the cusp boundaries
- field aligned currents (FAC) near the polar cap
- magnetotail current sheet
- currents in the plasma sheet boundary layer
- ring current

This 'beginner's guide' technical note will explain the theory in simple terms (Section 3), and explain the entire actual calculation of the current density (Section 4) and the quality parameters (Sections 3.2 and 5) that can help in determining the validity of the result. Section 6 gives indications of the order of current density that have been found in different regions of the magnetosphere.

Section 4 also includes original Python 3 code calculating step by step the current density and quality parameters from Cluster magnetic field data at a particular point in time. In the appendix, the full code is made available that enables you to

- load Cluster magnetic field data with 0.2s time resolution for any time period (using CEFLIB, `http://ceflib.irap.omp.eu`),
- calculate the current density and quality parameters with a user defined time resolution,
- generate a simple ASCII output file
- plot the result.

This code has been successfully compared with current densities published in the literature.

A link to the CEFLIB package is also available from the software page of the Cluster Science Archive: `http://www.cosmos.esa.int/web/csa/software`

Two other software packages linked from that page enable the calculation of the current density using the curlometer technique: the IRFU-MATLAB package (`https://github.com/irfu/irfu-matlab`) and the QSAS software (`http://www.sp.ph.ic.ac.uk/csc-web/QSAS/`). Please note that at the time of writing, the latter does not yet work on a Mac with Yosemite or El Capitan installed.

The curlometer technique has been used in many near regions of space since the launch of Cluster; see for instance Dunlop and Eastwood (2008) for a review and Dunlop et al. (2016) for a summary of the practical experience gained in its use with Cluster data.

2 CONDITIONS FOR USE

The technique requires that:

- the tetrahedron configuration of the spacecraft must be reasonably regular (see Section 5)
- there is a linear field gradient between the spacecraft (see Section 3.2). In the inner magnetosphere, the geomagnetic field contributes a non-linear field and so this needs to be subtracted using a geomagnetic field model before making the calculation.
- all measurement points are inside the same current sheet

These points will be explained below.

3 THEORY

3.1 $J_{average}$

Maxwell-Ampère's law states that

$$\mu_0 \mathbf{J} = \nabla \times \mathbf{B} - \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} \tag{1}$$

where $\mathbf{J}$ represents the current density and $\mathbf{B}$ the magnetic field. The term relating to the rate of change of the electric field describes the displacement current, which is nearly always negligible in space plasmas. [1] So, using all possible equivalent notations (as you may have learned a different notation):

$$\mathbf{J} = \frac{1}{\mu_0} \nabla \times \mathbf{B} = \frac{1}{\mu_0} \mathrm{rot}\, \mathbf{B} = \frac{1}{\mu_0} \mathrm{curl}\, \mathbf{B} \tag{2}$$

The integral definition of $\mathrm{curl}\, \mathbf{B}$ from Stokes' theorem gives

$$\iint \mathbf{J} \cdot ds = \frac{1}{\mu_0} \oint \mathbf{B} \cdot dl \tag{3}$$

Dunlop et al. (1988) derives the average approximation of the current density from this:

$$\mathbf{J}_{average} \cdot (\Delta \mathbf{r}_i \times \Delta \mathbf{r}_j) = \frac{1}{\mu_0} \Delta \mathbf{B}_i \cdot \Delta \mathbf{r}_j - \Delta \mathbf{B}_j \cdot \Delta \mathbf{r}_i \tag{4}$$

where $\mathbf{J}_{average}$ represents the measured mean current density over the tetrahedron volume, and $\Delta \mathbf{r}_i \equiv \mathbf{r}_i - \mathbf{r}_{ref}$ and similarly, $\Delta \mathbf{B}_j \equiv \mathbf{B}_j - \mathbf{B}_{ref}$.

Using one spacecraft as the reference, this can be applied to three independent faces of the tetrahedron, leading to a set of three equations for the independent components of the average current in the directions of the normals of the three faces around the reference spacecraft. The central assumption in this estimate

[1] The characteristic velocities within the plasma are much less than the speed of light, and quasi-neutrality applies (Russell, Luhmann, and Strangeway, 2016). For example, typical values for $\frac{\partial \mathbf{E}}{\partial t}$ rarely exceed 10 mV m^{-1} s^{-1} so dividing both sides by μ_0 and rounding up ε_0 to 10^{-11}, this term would be of the order of 10^{-14}A m^{-2} or 10^{-5}nA m^{-2}, while usual values of $\mathbf{J}$ are around 10s of nA m^{-2}.

is a linear field variation (i.e. linear gradient) between spacecraft so that $\mathbf{J}$ is constant over the spacecraft volume. Remember that if you are investigating the inner magnetosphere, you will need to subtract the geomagnetic field using a model, in order to remove the non-linear contribution from the static field.

3.2 Quality indicators: $\mathrm{div}\,\mathbf{B}$ and $|\mathrm{div}\,\mathbf{B}|/|\mathrm{curl}\,\mathbf{B}|$

According to **Gauss' law for magnetism**, $\mathrm{div}\,\mathbf{B}$ = 0. From Eq. (3), we can also estimate $\mathrm{div}\,\mathbf{B}$ using four point measurements in space, with:

$$\langle \mathrm{div}\,\mathbf{B} \rangle_{av} |\Delta \mathbf{r}_i \cdot (\Delta \mathbf{r}_j \times \Delta \mathbf{r}_k)| = \left| \sum_{cyclic} \Delta \mathbf{B}_i \cdot (\Delta \mathbf{r}_j \times \Delta \mathbf{r}_k) \right| \tag{5}$$

where $\langle \mathrm{div}\,\mathbf{B} \rangle_{av}$ is the differential estimate of $\mathrm{div}\,\mathbf{B}$ for the tetrahedron (Dunlop et al., 1988; Dunlop et al., 2002).

Such an estimation of $\mathrm{div}\,\mathbf{B}$ with real data produces non-zero values as a consequence of

- nonlinear spatial gradients neglected in its estimate (e.g. the geomagnetic field)
- the effect of timing and measurement errors

This estimation of $\mathrm{div}\,\mathbf{B}$ can provide a useful indicator of the quality of the estimate of $\mathbf{J}$, at least when the tetrahedron is regular (Dunlop et al., 2002), although it is not proportional to the error on $\mathbf{J}$. Dunlop and Eastwood (2008) state that the $\mathrm{div}\,\mathbf{B}$ estimator can only indicate when the current estimate may be bad.

An indication of the uncertainty in the current estimation is given by the ratio $|\mathrm{div}\,\mathbf{B}|/|\mathrm{curl}\,\mathbf{B}|$, which is related to the ratio $\nabla \mathrm{J}/\mathrm{J}$ (Robert et al., 1998a). This yields a dimensionless quantity where values $\ll 1$ are desirable (Haaland et al., 2004), which may then be expressed as a percentage deviation from zero.

To calculate it, once you have $\mathbf{J}_{average}$, multiply it by μ_0 to obtain $\mathrm{curl}\,\mathbf{B}$ (Equation (2)) and then calculate

$$|\mathrm{div}\,\mathbf{B}|/|\mathrm{curl}\,\mathbf{B}| \tag{6}$$

Dunlop et al. (2002) and Zhang et al. (2011) disregard the $\mathbf{J}_{average}$ when $|\mathrm{div}\,\mathbf{B}|/|\mathrm{curl}\,\mathbf{B}|$ is over 50%; Dunlop et al. (2015) uses 30%.

4 CALCULATION

The only CSA dataset required for the magnetic field and spacecraft position is the FGM magnetic field, as the position components are included. For a meaningful calculation, 4s resolution (SPIN) is enough, however, the times for these data points are all slightly different. The 5VPS (vectors per second, C[n]_CP_FGM_5VPS) data files contain already-interpolated data such that the same times are in all four files, which makes data analysis much simpler. The magnetic field components are given, along with the position components, in nT and km (respectively), in GSE coordinates.

In your calculation, remember to convert the magnetic field and position vectors from nT and km (respectively) into SI units. For clarity, $|x|$ refers to either the magnitude or the absolute value, depending on whether x is a vector or a scalar.

In order to calculate $\mathbf{J}_{average}$, we need Equation (4) for three sets of three spacecraft each to resolve the full vector $\mathbf{J}_{average}$. For example, the calculation below is one of the three necessary, for spacecraft C1, C2 and C4, where C4 is the reference spacecraft (to match Figure 1). If the tetrahedron is regular, it doesn't make much difference which spacecraft is used as the reference (see, for instance, Section 4.2 of Vallat et al., 2005), but since the calculation provides some redundancy (in the calculation that doesn't include the reference spacecraft), the quality of the estimation can also be checked this way.

Remember that $\Delta\mathbf{r}_i \equiv \mathbf{r}_i - \mathbf{r}_{ref}$, so $\Delta\mathbf{r}_{14} \equiv \mathbf{r}_1 - \mathbf{r}_4$, and similarly for $\Delta\mathbf{B}_i$. The right hand side of the calculation below will be the average current normal to the surface formed by spacecraft C1, C2 and C4 (that which is illustrated in Figure 1 by the lowest arrow, $\mathbf{J}_{124}$, which comes towards you, out of the page):

$$\mathbf{J}_{average} \cdot (\Delta\mathbf{r}_{14} \times \Delta\mathbf{r}_{24}) = \frac{1}{\mu_0}\Delta\mathbf{B}_{14} \cdot \Delta\mathbf{r}_{24} - \Delta\mathbf{B}_{24} \cdot \Delta\mathbf{r}_{14}$$

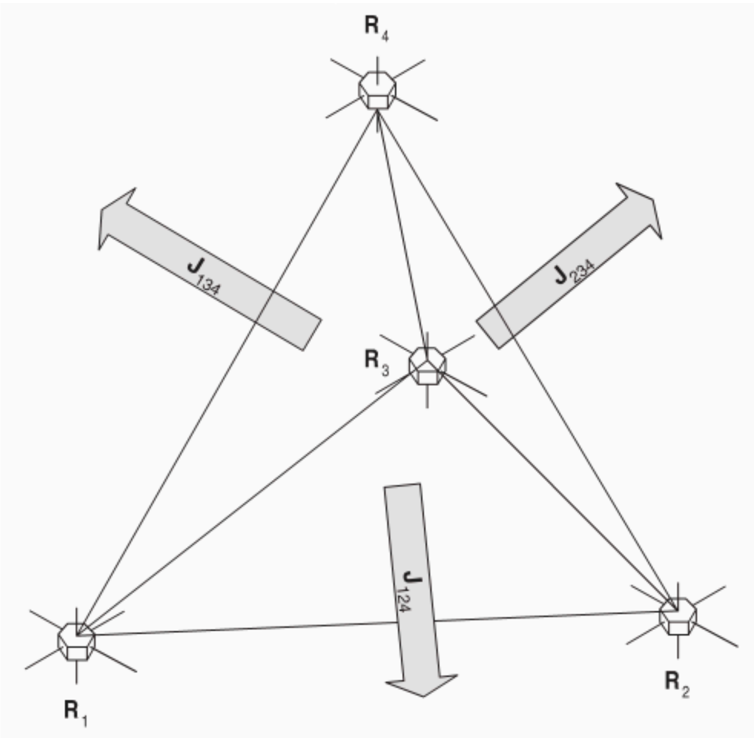

Figure 1: Illustration of the Curlometer estimate, where the spacecraft R_3 is furthest away (into the page). After Dunlop et al. (2002).

Doing this for the three sets of three spacecraft with the same reference spacecraft (C4), and calculating e.g., $(\Delta\mathbf{r}_{14} \times \Delta\mathbf{r}_{24})$ as $[Rx_{124}, Ry_{124}, Rz_{124}]$ gives you a matrix equation:

$$\begin{bmatrix} Rx_{124} & Ry_{124} & Rz_{124} \\ Rx_{234} & Rx_{234} & Rx_{234} \\ Rx_{134} & Rx_{134} & Rx_{134} \end{bmatrix} \begin{bmatrix} J_{average(x)} \\ J_{average(y)} \\ J_{average(z)} \end{bmatrix} = \frac{1}{\mu_0} \begin{bmatrix} I_{average(124)} \\ I_{average(234)} \\ I_{average(134)} \end{bmatrix}$$

And since for

$$Ar = k \tag{7}$$

to find r, one would use

$$r = A^{-1}k \tag{8}$$

we invert our coordinate transformation matrix to obtain the full $\mathbf{J}_{average}$ in Cartesian coordinates.

$$\begin{bmatrix} J_{average(x)} \\ J_{average(y)} \\ J_{average(z)} \end{bmatrix} = \frac{1}{\mu_0} \begin{bmatrix} Rx_{124} & Ry_{124} & Rz_{124} \\ Rx_{234} & Rx_{234} & Rx_{234} \\ Rx_{134} & Rx_{134} & Rx_{134} \end{bmatrix}^{-1} \begin{bmatrix} I_{average(124)} \\ I_{average(234)} \\ I_{average(134)} \end{bmatrix}$$

To be absolutely clear, Equation (5) for $\operatorname{div}\mathbf{B}$ for our example with C4 as the reference spacecraft, is calculated thus:

$$\langle\nabla\cdot\mathbf{B}\rangle_{av}|\Delta\mathbf{r}_{14}\cdot(\Delta\mathbf{r}_{24}\times\Delta\mathbf{r}_{34})| = |\Delta\mathbf{B}_{14}\cdot(\Delta\mathbf{r}_{24}\times\Delta\mathbf{r}_{34}) + \Delta\mathbf{B}_{24}\cdot(\Delta\mathbf{r}_{34}\times\Delta\mathbf{r}_{14}) + \Delta\mathbf{B}_{34}\cdot(\Delta\mathbf{r}_{14}\times\Delta\mathbf{r}_{24})|$$

Then, as mentioned before, $\operatorname{div}\mathbf{B}/|\operatorname{curl}\mathbf{B}|$ needs $\mathbf{J}_{average}$, multiplied by μ_0 to obtain $\operatorname{curl}\mathbf{B}$, take the magnitude of this and divide $\langle\operatorname{div}\mathbf{B}\rangle_{av}$ by the result.

These calculations are both in the example script (in Python 3) below. This script is only to test the mathematics within the code. If you re-write this in a language you are more comfortable with, ensure that you obtain the same results. The data point is the first from the period used by Dunlop and Balogh (2005) for their Figure 2a; from 11th June 2001, 20:05-20:23; 20:05:00.1. This figure's calculation was done using 5VPS data, but is presented in their paper in GSM coordinates, and so it is very slightly different from the reproduction in this document (in GSE). However, ensure that given the example data below, you get the same results:

```python
import numpy as np
import math

def delta(ref, i):
    delrefi = i - ref
    return delrefi

# Constants and conversions
mu0 = (4*math.pi)*1e-7
km2m = 1e3
nT2T = 1e-9

# Data example in GSE (2001-06-11T20:05:00.100Z)
C1B = np.array([-0.074, -4.924, -1.178])*nT2T
C1R = np.array([-17048.9, -121511.3, 12814.3])*km2m
C2B = np.array([0.575, -0.864, 0.440])*nT2T
C2R = np.array([-18907.9, -122063.9, 13353.4])*km2m
C3B = np.array([-0.239, -2.992, 0.959])*nT2T
C3R = np.array([-18363.2, -121961.3, 11480.5])*km2m
C4B = np.array([0.670, -3.871, -0.616])*nT2T
C4R = np.array([-17470.0, -123201.5, 12519.8])*km2m

delR14 = delta(C4R, C1R)
delR24 = delta(C4R, C2R)
delR34 = delta(C4R, C3R)
delB14 = delta(C4B, C1B)
delB24 = delta(C4B, C2B)
delB34 = delta(C4B, C3B)
```

```

# Calculate J

# Have to 'convert' this to a matrix to be able to get the inverse.
R = np.matrix(([np.cross(delR14, delR24), np.cross(delR24, delR34),
        np.cross(delR14, delR34)]))
# The inverse:
Rinv = R.I

# I(average) matrix (note the shape):
Iave = ([np.dot(delB14, delR24) - np.dot(delB24, delR14)],
       [np.dot(delB24, delR34) - np.dot(delB34, delR24)],
       [np.dot(delB14, delR34) - np.dot(delB34, delR14)])

# The dot is equivalent to * here
JJ = np.dot(Rinv,Iave)/mu0
print(JJ)

# Calculate div B

lhs = np.dot(delR14, np.cross(delR24, delR34))

rhs = np.dot(delB14, np.cross(delR24, delR34)) + \
    np.dot(delB24, np.cross(delR34, delR14)) + \
    np.dot(delB34, np.cross(delR14, delR24))

divB = rhs/lhs
print(divB)


curlB = JJ*mu0
magcurlB = math.sqrt(curlB[0]**2 + curlB[1]**2 + curlB[2]**2)
divBbycurlB = abs(divB)/magcurlB
print(divBbycurlB)
```

For the example data given, the answers should be:

$$\mathbf{J}_{average} = \begin{bmatrix} -4.184e-10 \\ 1.136e-09 \\ -1.171e-09 \end{bmatrix}$$

$$\langle \mathrm{div}\mathbf{B} \rangle_{\mathrm{av}} = 9.084\mathrm{e}-16$$

$$\frac{\langle \mathrm{div}\mathbf{B} \rangle_{\mathrm{av}}}{|\mathrm{curl}\mathbf{B}|} = 0.429$$

5 TETRAHEDRON PARAMETERS

The tetrahedron must be as close to a regular tetrahedron as possible, i.e. all the distances between all four spacecraft are the same. If the spacecraft all lie in the same plane or even in a 'string-of-pearls' configuration, clearly there is no volume in which to calculate the average current density. There are 1D and 2D parameters which are in the Cluster Science Archive that tell you about the 'regular-ness' of the tetrahedron. The relationship and merits of both the 1D and 2D parameters are discussed in Robert et al. (1998b).

5.1 One dimensional parameters

These are the Glassmeier, Q_G, and Robert/Roux, Q_R parameters (Daly, 1994). The Glassmeier parameter is defined as:

$$Q_G = \frac{\text{True Vol.}}{\text{Ideal Vol.}} + \frac{\text{True Surf.}}{\text{Ideal Surf.}} + 1 \tag{9}$$

where it can take a value between 1 and 3, where 3 is a regular tetrahedron.

> "QGM takes values between 1 and 3, and attempts to describe the 'fractional dimension' of the tetrahedron: a value of 1 indicates that the four spacecraft are in a line, while a value equal to 3 indicates that the tetrahedron is regular. There is nevertheless some difficulty with this interpretation: it is perfectly possible to deform a regular (QGM = 3) tetrahedron continuously until it resembles a straight line (QGM = 1) without it resembling a plane at any time; therefore QGM = 2 is not a sufficient condition for planarity." Robert et al. (1998b)

The Robert/Roux parameter uses a sphere that circumscribes the tetrahedron (all four points on its surface) and is defined as:

$$Q_R = \left(\frac{9\pi}{2\sqrt{3}} . \frac{\text{True Vol.}}{\text{Sphere Vol.}} \right)^{\frac{1}{3}} \tag{10}$$

where $\frac{9\pi}{2\sqrt{3}}$ is a normalization factor to make Q_R = 1 for a regular tetrahedron. The range of values is between 0 and 1.

> "This parameter was selected from many on the basis of its usefulness in estimating the error in the determination of the spatial gradient of the magnetic field." Robert et al. (1998b)

The two 1D parameters can be found under 'Auxiliary, MAARBLE and ECLAT support data', Ancillary/General/Auxiliary Data -> sc_config_QG and sc_config_QR. This dataset has a time resolution of 1 minute, so the values of Q_G and Q_R for 11th June 2001, 20:05:30.0, are Q_G = 2.98331 and Q_R = 0.992108.

5.2 Two dimensional parameters

There are alternatives for defining the shape. Where a, b and c are the major, middle and minor semiaxes of the pseudo-ellipsoid which would surround the tetrahedron, the shape can be defined by the elongation and the planarity, as follows:

$$\begin{aligned} \text{Elongation, } E &= 1 - (b/a) \\ \text{Planarity, } P &= 1 - (c/b) \end{aligned} \tag{11}$$

These 2D parameters can be found under 'Auxiliary, MAARBLE and ECLAT support data', Ancillary/General/Auxiliary Data -> sc_geom_elong and sc_geom_planarity. As mentioned above, this dataset has a time resolution of 1 minute, so the values of E and P for 11th June 2001, 20:05:30.0, are E = 0.0850134 and P = 0.085342. See Robert et al., 1998a for an analysis of the effects of E and P on the accuracy of the current density estimate.

6 TYPICAL CURRENT DENSITY VALUES IN THE MAGNETOSPHERE

Feature/Region	Typical values for $\mathbf{J}$
magnetopause currents	~10 nA m^{-2} (Dunlop and Eastwood, 2008), up to 50 nA m^{-2} (Phan et al., 2004), see also Dunlop and Balogh, 2005
currents in flux transfer events	~1 nA m^{-2} (Dunlop and Eastwood, 2008) up to 10 nA m^{-2} (Pu et al., 2005)
current at the cusp boundaries	~20 nA m^{-2} (Dunlop et al., 2002) sect 3.4
field aligned currents (FAC) near the polar cap	~200 nA m^{-2} at 3.3 R_E altitude (Dunlop et al., 2002) sect 3.3, while Dunlop et al., 2015 reports ~2 μA m^{-2} at 500km altitude and ~20 nA m^{-2} at 2.5 R_E altitude
magnetotail current sheet	up to ~30 nA m^{-2} (Runov, Nakamura, and Baumjohann, 2006)
currents in the plasma sheet boundary layer	~10 nA m^{-2}, (Nakamura et al., 2004)
ring current	9-27 nA m^{-2} at 4-4.5 R_E, (Zhang et al., 2011)

Many papers have used this technique with Cluster and other multi-spacecraft combinations, see Dunlop and Eastwood (2008) for a review of work and Dunlop et al. (2016) for a summary of practical experience of using the curlometer technique.

References

Daly, P W (1994). “The Tetrahedron Quality Factors”. In: *Max-Planck-Institut für Aeronomie*.

Dunlop, M W and A Balogh (2005). “Magnetospheric current as seen by Cluster”. In: *Annales Geophysicae* 23, pp. 901–907.

Dunlop, M W and J P Eastwood (2008). “The Curlometer and Other Gradient Based Methods”. In: *Multi-Spacecraft Analysis Methods Revisited*. Ed. by G Paschmann and P W Daly. Vol. SR-008. ISSI Scientific Report. ISSI/ESA. Chap. 2, pp. 17–25.

Dunlop, M W et al. (1988). “Analysis of multipoint magnetometer data”. In: *Advances in Space Research* 8.9-10, pp. 273–277.

Dunlop, M W et al. (2016). “Commentary on accessing 3-D currents in space: Experiences from Cluster”. In: *Journal of Geophysical Research* 121. doi:10.1002/2016JA022668, pp. 7881–7886.

Dunlop, M W et al. (2002). “Four-point Cluster application of magnetic field analysis tools: The Curlometer”. In: *Journal of Geophysical Research* 107.A11, pp. 1384–1398.

Dunlop, M W et al. (2015). “Simultaneous field-aligned currents at Swarm and Cluster satellites”. In: *Geophysical Research Letters* 42, pp. 3683–3691.

Haaland, S et al. (2004). “Orientation and motion of a discontinuity from Cluster curlometer capability: Minimum variance of current density”. In: *Geophysical Research Letters* 31.L10804. doi:10.1029/2004GL020001.

Nakamura, R et al. (2004). “Flow shear near the boundary of the plasma sheet observed by Cluster and Geotail”. In: *Journal of Geophysical Research* 109.A05204.

Phan, T D et al. (2004). “Cluster observations of continuous reconnection at the magnetopause under steady interplanetary magnetic field conditions”. In: *Annales Geophysicae* 22, pp. 2355–2367.

Pu, Z Y et al. (2005). “Multiple flux rope events at the high-latitude magnetopause: Cluster/RAPID observation On 26 January, 2001”. In: *Surveys in Geophysics* 26, pp. 193–214.

Robert, P et al. (1998a). “Accuracy of Current Density Determination”. In: *Analysis Methods for Multi-Spacecraft Data*. Ed. by G Paschmann and P W Daly. Vol. SR-001. ISSI Scientific Report. ISSI/ESA. Chap. 16, pp. 395–418.

Robert, P et al. (1998b). “Tetrahedron Geometric Factors”. In: *Analysis Methods for Multi-Spacecraft Data*. Ed. by G Paschmann and P W Daly. Vol. SR-001. ISSI Scientific Report. ISSI/ESA. Chap. 13, pp. 323–348.

Runov, A, R Nakamura, and W Baumjohann (2006). “Multi-point study of the magnetotail current sheet”. In: *Advances in Space Research* 38, pp. 85–92.

Russell, C T, J G Luhmann, and R J Strangeway (2016). *Space Physics: An Introduction*. 978-1107098824. Cambridge University Press.

Vallat, C et al. (2005). “First current density measurements in the ring current region using simultaneous multi-spacecraft CLUSTER-FGM data”. In: *Annales Geophysicae* 23, pp. 1849–1865.

Zhang, Q-H et al. (2011). “The distribution of the ring current: Cluster observations”. In: *Annales Geophysicae* 29, pp. 1655–1662.

A CURLOMETER PYTHON CODE

This code, in Python 3, uses the IRAP CEFLIB library (http://ceflib.irap.omp.eu) to read in the FGM 5VPS data from each Cluster spacecraft for 20:05-20:23 on 11th June 2001, in order to reproduce Figure 2a from Dunlop and Balogh (2005), although there are slight differences due to the figure being in GSM and the reproduction here being in GSE. There is a very small gap in the data for C1 and so the code aligns the data according to the millisecond time tag. The figure presents data at 5 vectors per second resolution but the code is able to take one point every [window] seconds and calculates the current density from these data if a lower resolution is wanted. It then writes these results to file and plots them in a series of graphs, which are saved as a .png file. Lastly, the auxiliary data is read in (for the tetrahedron shape-quality parameters) and plotted to a .png file. The results can be found after the code.

```
#!/usr/bin/env python

import numpy as np # for matrix calculations
import math # for pi and sqrt
import glob # for sensible listdir()
import ceflib # for reading CEF files
from copy import deepcopy # for obtaining variables in CEF files
import matplotlib.pyplot as plt # for plotting
import datetime as dt # for dates
from matplotlib import dates # for formatting axes

# User-defined variables:

# Path to data:
path = r'../Data/FGM_5VPS__20010611_200500_20010611_202300/'

# Filename for output current density:
outfile = '/Users/Hmiddleton/Documents/Cluster/Projects/\
InProgress/Curlometer/Data/200106112005-2023_J.txt'

# Plot filenames:
BJQFileName = '20010611_2005-2023_BJQ.png'
GeomFileName = '20010611_2005-2023_Geom.png'

# X-axis labels:
XAxisLabel = 'Time on 11th June 2001'

# Desired resolution of data for the curlometer calculation
window = 0.2 # in seconds, 0.2 = minimum


'''The Curlometer Function'''
def delta(ref, i):
    delrefi = i - ref
    return delrefi

def curlometer(d1, d2, d3, d4):

    km2m = 1e3
    nT2T = 1e-9
    mu0 = (4*math.pi)*1e-7

    C1R = np.array([d1[3], d1[4], d1[5]])*km2m
    C1B = np.array([d1[0], d1[1], d1[2]])*nT2T
    C2R = np.array([d2[3], d2[4], d2[5]])*km2m
    C2B = np.array([d2[0], d2[1], d2[2]])*nT2T
    C3R = np.array([d3[3], d3[4], d3[5]])*km2m
    C3B = np.array([d3[0], d3[1], d3[2]])*nT2T
    C4R = np.array([d4[3], d4[4], d4[5]])*km2m
```

```
    C4B = np.array([d4[0], d4[1], d4[2]])*nT2T

    delB14 = delta(C4B, C1B)
    delB24 = delta(C4B, C2B)
    delB34 = delta(C4B, C3B)
    delR14 = delta(C4R, C1R)
    delR24 = delta(C4R, C2R)
    delR34 = delta(C4R, C3R)

# J

    # Have to 'convert' this to a matrix to be able to get the inverse.
    R = np.matrix(([np.cross(delR14, delR24), np.cross(delR24, delR34),
         np.cross(delR14, delR34)]))
    Rinv = R.I

    # I(average) matrix:
    Iave = ([np.dot(delB14, delR24) - np.dot(delB24, delR14)],
        [np.dot(delB24, delR34) - np.dot(delB34, delR24)],
        [np.dot(delB14, delR34) - np.dot(delB34, delR14)])

    JJ = (Rinv*Iave)/mu0

# div B
    lhs = np.dot(delR14, np.cross(delR24, delR34))

    rhs = np.dot(delB14, np.cross(delR24, delR34)) + \
        np.dot(delB24, np.cross(delR34, delR14)) + \
        np.dot(delB34, np.cross(delR14, delR24))

    divB = abs(rhs)/abs(lhs)

# div B / curl B
    curlB = JJ*mu0
    magcurlB = math.sqrt(curlB[0]**2 + curlB[1]**2 + curlB[2]**2)
    divBbycurlB = divB/magcurlB

    return [JJ, divB, divBbycurlB]
# End of curlometer function


'''Read in all the data using CEFLIB.read '''

cluster = ['C'+str(x) for x in range(1,5)]

time = {}
B = {}
pos = {}

for c in cluster:
    folder = c+'_CP_FGM_5VPS/*.cef'
    filename = glob.glob(path+folder)
    print(filename)
    ceflib.read(filename[0])
    time[c] = deepcopy(ceflib.var('time_tags')) # in milli-seconds
    B[c] = deepcopy(ceflib.var('B_vec_xyz_gse'))
    pos[c] = deepcopy(ceflib.var('sc_pos_xyz_gse'))
    ceflib.close()

'''Align all the data with the time by using a dictionary with
the time in milliseconds as the key'''
clean = {}
for c in cluster:
    for i,p in enumerate(time[c]):
```

```
        if p not in clean.keys():
            clean[int(p)] = {}
        clean[p][c] = [B[c][i][0],
                       B[c][i][1],
                       B[c][i][2],
                       pos[c][i][0],
                       pos[c][i][1],
                       pos[c][i][2]]

mintime, maxtime = min(clean.keys()), max(clean.keys())

# Time array (min, max, step)
tarr = range(mintime, maxtime, int(window*1000))
nwin = len(tarr)

Jave = np.zeros(nwin, dtype = [('time', float),('Jx', float),
                               ('Jy', float),('Jz', float),
                               ('divB', float),
                               ('divBcurlB', float)])

for i,t in enumerate(tarr):

    if len(clean[t]) == 4:
        onej = curlometer(clean[t]['C1'],clean[t]['C2'],
                          clean[t]['C3'],clean[t]['C4'])

        Jave['time'][i] = t/1000
        Jave['Jx'][i] = onej[0][0]
        Jave['Jy'][i] = onej[0][1]
        Jave['Jz'][i] = onej[0][2]
        Jave['divB'][i] = onej[1]
        Jave['divBcurlB'][i] = onej[2]
    else:
        Jave['time'][i] = t/1000
        Jave['Jx'][i] = np.nan
        Jave['Jy'][i] = np.nan
        Jave['Jz'][i] = np.nan
        Jave['divB'][i] = np.nan
        Jave['divBcurlB'][i] = np.nan

'''Write all results out to file, tarr is already sorted'''

with open(outfile, 'w') as f:
    for j in Jave:
        outstring = str(dt.datetime.utcfromtimestamp(j['time'])) + \
        ', ' + str(j['Jx']) + ', ' + str(j['Jy']) + \
        ', ' + str(j['Jz']) +', '+ str(j['divBcurlB'])+'\n'
        f.write(outstring)

'''Pull out the mag field used for the calculation'''
Magnpt = {}
for c in cluster:
    Bx, By, Bz, Bmag = [], [], [], []
    for p in tarr:
        if c in clean[p].keys():
            Bx.append(clean[p][c][0])
            By.append(clean[p][c][1])
            Bz.append(clean[p][c][2])
            Bmag.append(math.sqrt(clean[p][c][0]**2 + clean[p][c][1]**2 + clean[p][c/
                ][2]**2))
        else:
            Bx.append(np.nan)
            By.append(np.nan)
            Bz.append(np.nan)
```

```
            Bmag.append(np.nan)
    Magnpt[c] = [Bx, By, Bz, Bmag]

'''Take times and put as date into list'''
tdate = []
for t in tarr:
    tdate.append(dates.date2num(dt.datetime.utcfromtimestamp(t/1000)))


'''Plot the B field and the current density and divB/curlB'''
fig = plt.figure(figsize=(8.5, 12))

hfmt = dates.DateFormatter('%H:%M')
minutes = dates.MinuteLocator(interval=2)

sub1=fig.add_subplot(811)
plt.plot(tdate, Magnpt['C1'][2], color='black', linestyle='-', label = 'C1')
plt.plot(tdate, Magnpt['C2'][2], color='red', linestyle='-', label = 'C2')
plt.plot(tdate, Magnpt['C3'][2], color='green', linestyle='-', label = 'C3')
plt.plot(tdate, Magnpt['C4'][2], color='blue', linestyle='-', label = 'C4')
plt.ylim(-20, 30)
plt.yticks([-20,-10,0,10,20,30])
plt.ylabel('Bz (nT)')
sub1.xaxis.set_major_locator(minutes)
sub1.xaxis.set_major_formatter(hfmt)
plt.legend(bbox_to_anchor=(1.02, 1), loc=2, borderaxespad=0., fontsize='small')

sub2=fig.add_subplot(812)
plt.plot(tdate, Magnpt['C1'][1], color='black', linestyle='-' )
plt.plot(tdate, Magnpt['C2'][1], color='red', linestyle='-' )
plt.plot(tdate, Magnpt['C3'][1], color='green', linestyle='-' )
plt.plot(tdate, Magnpt['C4'][1], color='blue', linestyle='-' )
plt.ylim(-20, 20)
plt.yticks([-20, -10, 0, 10, 20])
plt.ylabel('By (nT)')
sub2.xaxis.set_major_locator(minutes)
sub2.xaxis.set_major_formatter(hfmt)

sub3=fig.add_subplot(813)
plt.plot(tdate, Magnpt['C1'][0], color='black', linestyle='-' )
plt.plot(tdate, Magnpt['C2'][0], color='red', linestyle='-' )
plt.plot(tdate, Magnpt['C3'][0], color='green', linestyle='-' )
plt.plot(tdate, Magnpt['C4'][0], color='blue', linestyle='-' )
plt.ylim(-20, 20)
plt.yticks([-20, -10, 0, 10, 20])
plt.ylabel('Bx (nT)')
sub3.xaxis.set_major_locator(minutes)
sub3.xaxis.set_major_formatter(hfmt)

sub4=fig.add_subplot(814)
plt.plot(tdate, Magnpt['C1'][3], color='black', linestyle='-' )
plt.plot(tdate, Magnpt['C2'][3], color='red', linestyle='-' )
plt.plot(tdate, Magnpt['C3'][3], color='green', linestyle='-' )
plt.plot(tdate, Magnpt['C4'][3], color='blue', linestyle='-' )
plt.ylim(0, 30)
plt.yticks([0, 10, 20, 30])
plt.ylabel('|B| (nT)')
sub4.xaxis.set_major_locator(minutes)
sub4.xaxis.set_major_formatter(hfmt)

sub5=fig.add_subplot(815)
plt.plot(tdate, Jave['Jz']*1e9, color='black', linestyle='-' )
plt.ylim(-15, 10)
plt.yticks([-15, -10, -5, 0, 5, 10])
```

```
plt.ylabel(r'$\mathbf{J}_Z (nA/m^2)$')
sub5.xaxis.set_major_locator(minutes)
sub5.xaxis.set_major_formatter(hfmt)

sub6=fig.add_subplot(816)
plt.plot(tdate, Jave['Jy']*1e9, color='red', linestyle='-' )
plt.ylim(-20, 20)
plt.yticks([-20, -10, 0, 10, 20])
plt.ylabel(r'$\mathbf{J}_Y (nA/m^2)$')
sub6.xaxis.set_major_locator(minutes)
sub6.xaxis.set_major_formatter(hfmt)

sub7=fig.add_subplot(817)
plt.plot(tdate, Jave['Jx']*1e9, color='green', linestyle='-' )
plt.ylim(-10, 20)
plt.yticks([-10, 0, 10, 20])
plt.ylabel(r'$\mathbf{J}_X (nA/m^2)$')
sub7.xaxis.set_major_locator(minutes)
sub7.xaxis.set_major_formatter(hfmt)

sub8=fig.add_subplot(818)
plt.plot(tdate, Jave['divBcurlB'], color='blue', linestyle='-' )
plt.ylim(0, 2)
plt.yticks([0, 1, 2])
plt.ylabel(r'$|{\rm {div}}\,\mathbf{B}|/|{\rm {curl}}\,\mathbf{B}|$')
sub8.xaxis.set_major_locator(minutes)
sub8.xaxis.set_major_formatter(hfmt)
plt.xlabel(XAxisLabel)
plt.savefig(BJQFileName, dpi=300)
#plt.show()
plt.close()

'''Read in the geometry data'''

all_data = {}
folder = 'CL_SP_AUX/*.cef'
filename = glob.glob(path+folder)
ceflib.read(filename[0])

time = deepcopy(ceflib.var('time_tags')) # in milli-seconds
QG = deepcopy(ceflib.var('sc_config_QG'))
QR = deepcopy(ceflib.var('sc_config_QR'))
E = deepcopy(ceflib.var('sc_geom_elong'))
P = deepcopy(ceflib.var('sc_geom_planarity'))
ceflib.close()


tgdate = []
for t in time:
    tgdate.append(dates.date2num(dt.datetime.utcfromtimestamp(t/1000)))

gminutes = dates.MinuteLocator(interval=2)

'''Plot the geometrical parameters'''
fig = plt.figure(figsize=(8, 5))

gsub1 = fig.add_subplot(211)
plt.plot(tgdate, P, label = 'Planarity')
plt.plot(tgdate, E, label = 'Elongation')
plt.legend(bbox_to_anchor=(0.4, 0.8), loc=2, borderaxespad=0., fontsize='small')
gsub1.xaxis.set_major_locator(gminutes)
gsub1.xaxis.set_major_formatter(hfmt)

gsub2 = fig.add_subplot(212)
```

```
plt.plot(tgdate, QR, label = r'$Q_R$')
plt.plot(tgdate, QG, label = r'$Q_G$')
plt.xlabel('Time on 4th February 2001')
plt.ylim(0, 3.5)
gsub2.xaxis.set_major_locator(gminutes)
gsub2.xaxis.set_major_formatter(hfmt)
plt.legend(bbox_to_anchor=(0.4, 0.75), loc=2, borderaxespad=0., fontsize='small', /
      ncol=2)
plt.savefig(GeomFileName, dpi=300)

#plt.show()
plt.close()
```

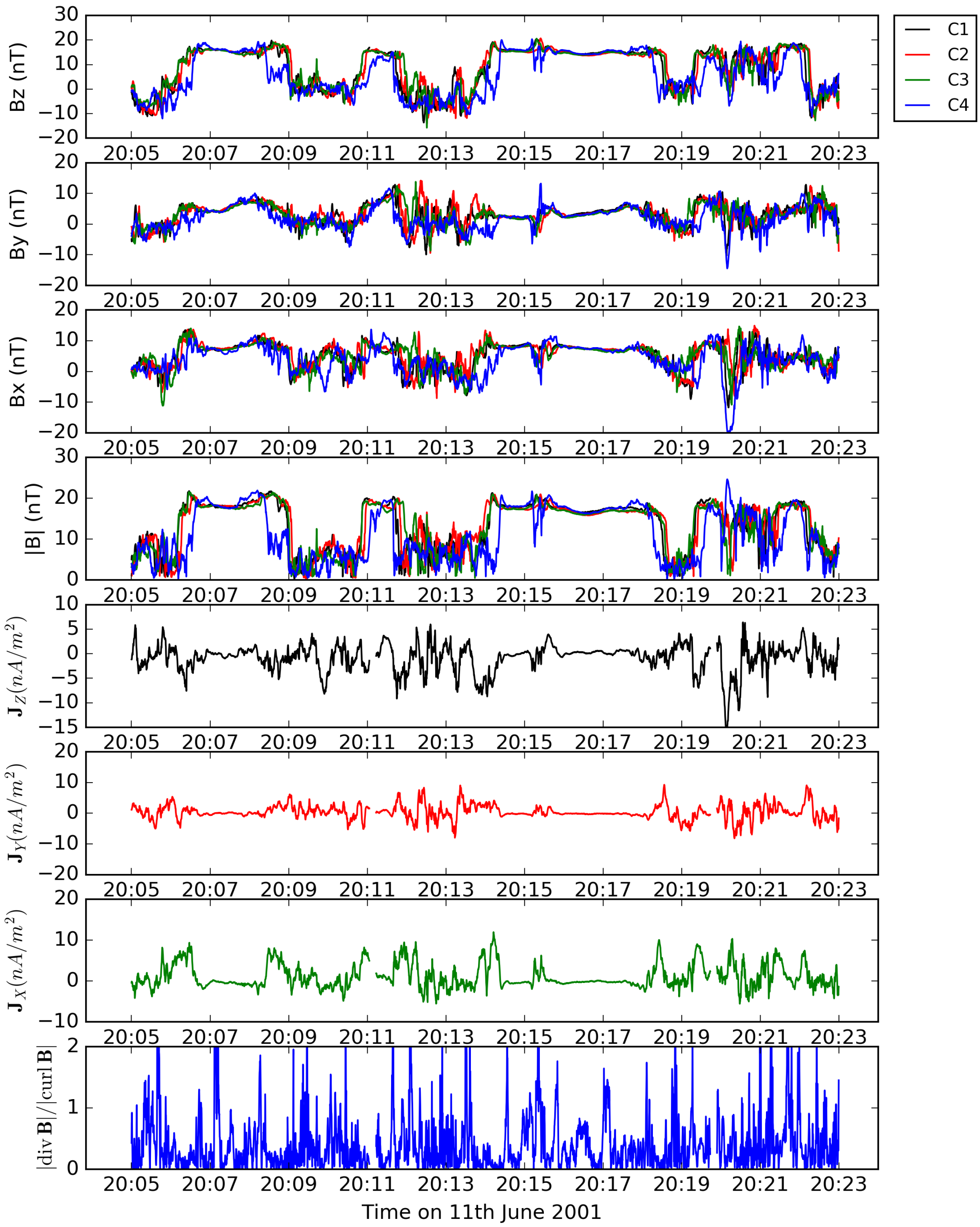

Figure 2: Reproduction of Dunlop and Balogh (2005) Figure 2a.

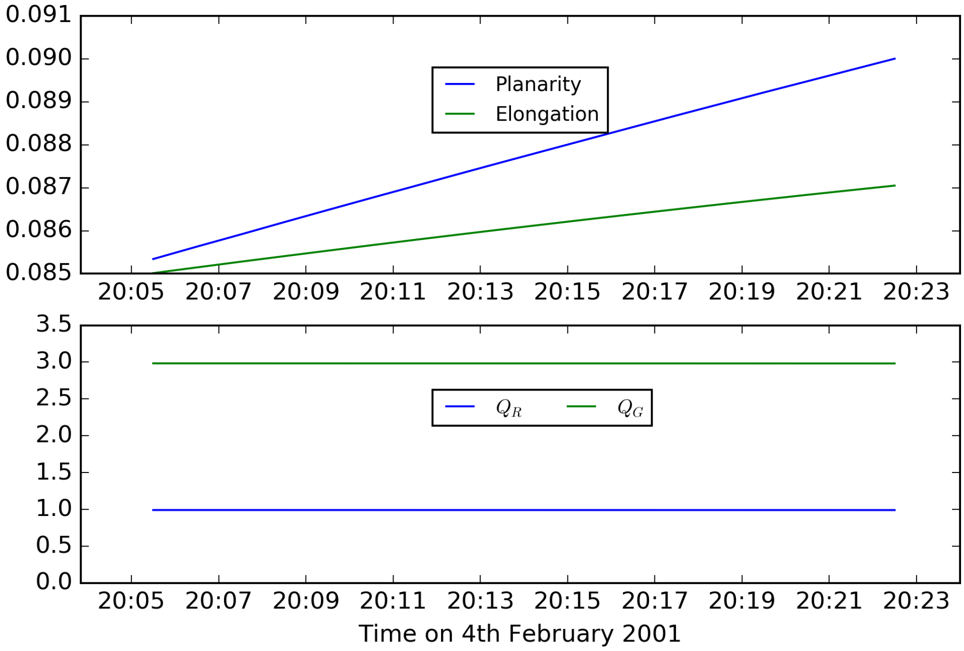

Figure 3: Geometric parameters for data in Dunlop and Balogh (2005) Figure 2a.