

**Measure of Solar surface velocities from granule tracking:
Coherent Structure Tracking (CST) algorithm.**

VERSION 1.1

Thierry Roudier (IRAP)
thierry.roudier@irap.omp.eu

Martine Chane-Yook (IAS)
martine.chane-yook@universite-paris-saclay.fr

April 23, 2020

Contents

1	Introduction	2
2	Description of CST codes	2
3	CST algorithm	3
4	Procedure to take SDO/HMI data from JSOC	11
5	Running CST	12
6	Description of input file	18
7	Example: CST codes applied to 30 minutes HMI observations	19
8	Frequent troubles encountered	20
9	Important note about the usage of CST codes	21
10	Other applications of CST codes	21
11	CST MEDOC WORKSHOP	21
12	Acknowledgements	21
	References	22

1 Introduction

Coherent **Structure Tracking (CST)** algorithm (Roudier et al. (2012); Švanda et al. (2013); Roudier et al. (2013); Rincon et al. (2017); Roudier et al. (2018)) is a set of codes written in IDL and Fortran 90, computing the horizontal velocity field on the Sun surface by using solar granules as tracers. CST codes take as input **HMI/SDO** intensity images (hmi.Ic_45s), with a time step of 45s. Each record includes a list of keywords and one image of the Sun (2D, 4096x4096) in FITS format. We use also the HMI/SDO Dopplergrams (hmi.V_45s), to derive spherical velocities on the Sun surface. The V_x and V_y (in km/s) are computed at a cadence of 30 min with a spatial window of 7 pixels, equivalent to 3.5 arcsec, around 2.5 Megameters (Mm) (Rieutord et al., 2001). HMI/SDO data have the North at the bottom and East on the right of the image; the images are rotated in order to have the North on top and East on the left. To read and treat HMI/SDO data, we use **SSWIDL** and ifort compiler. The Fortran part is parallelized with OpenMP.

Version 1.1 of CST codes is available from the **MEDOC** website. What is new compared to previous release is that IDL codes from step 3 are combined into one single IDL file. Furthermore, input file (param_seq_) and step 1 to 3 describing CST codes are explained in details (see section 6 and 3). An automatic way to take HMI data from JSOC is proposed (see section 4).

In the following sections, we explain in details CST codes: downloading of HMI intensity and Doppler data from **JSOC** (HMI 45s-cadence data series are not available from **MEDOC**); data reduction; algorithm; etc.

2 Description of CST codes

Nature of the physical problem: Measure of Solar surface velocities

Method of solution: Granule tracking, Daubechies wavelets

Other relevant information: IDL step 3 is reduced to a single file instead of 3 files in the previous version of CST codes (V1.0)

Authors: Th. Roudier, M. Rieutord, N. Meunier, F. Rincon, S. Roques, N. Renon

Program available from: <https://idoc.osups.universite-paris-saclay.fr/medoc/tools/cst-codes/>

Version: **1.1**

Computer(s) on which program has been tested: IAS server

Operating System(s) for which version of program has been tested: Linux (Debian 10)

Programming language used: IDL (with **SSWIDL** software) and Fortran 90 (with **ifort** com-

piler)

Status: Stable

Accessibility: open ([MEDOC](#))

Nb. of code lines in combined program and test deck: 3111 lines for IDL files and 4018 lines for Fortran files (libraries not included)

Typical running time: for 30 min HMI observations, IDL parts take 15 min and Fortran part takes 1h16 on IAS server with the following characteristics (CPU: 2 x Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz 24 cores, Memory: 256Go)

3 CST algorithm

Figure 1 describes the scheme of CST codes.

They take as input HMI/SDO intensity images and Dopplergrams, with a time step of 45s (4096 x 4096 pixels). The procedure to take HMI intensity and Doppler data from [JSOC](#) is described in sections 4 and 5. CST codes can be run up to 6 HMI observation days. The reference day is the 4th observation day (for more details, see remark 3.2).

Remark 3.1 Definition of derotation

We often use the term “derotation” : this term indicates that we have applied an algorithm that corrects natural differential rotation of the Sun and brings back longitudes related to solar surface at the same locations for each time deviation from the origin of the first HMI image. In other words, points of the longitude 0 degrees at $t=0$ which are at different longitudes are brought back to $t= 10$ min for example at longitude 0 degrees for this time. Thus we can follow the evolution in time of the structures at their original longitude.

CST algorithm is divided into 3 steps :

- step 1 : CST IDL part
- step 2 : CST Fortran part
- step 3 : CST IDL part

We describe below each step of CST codes.

Step 1 : CST IDL part

The first step concerns data preparation under IDL using SSWIDL as input for CST Fortran part (step 2). More precisely, it is data reduction with recentring and (with or not) derotating of the differential rotation of the Sun. It summarized as follow :

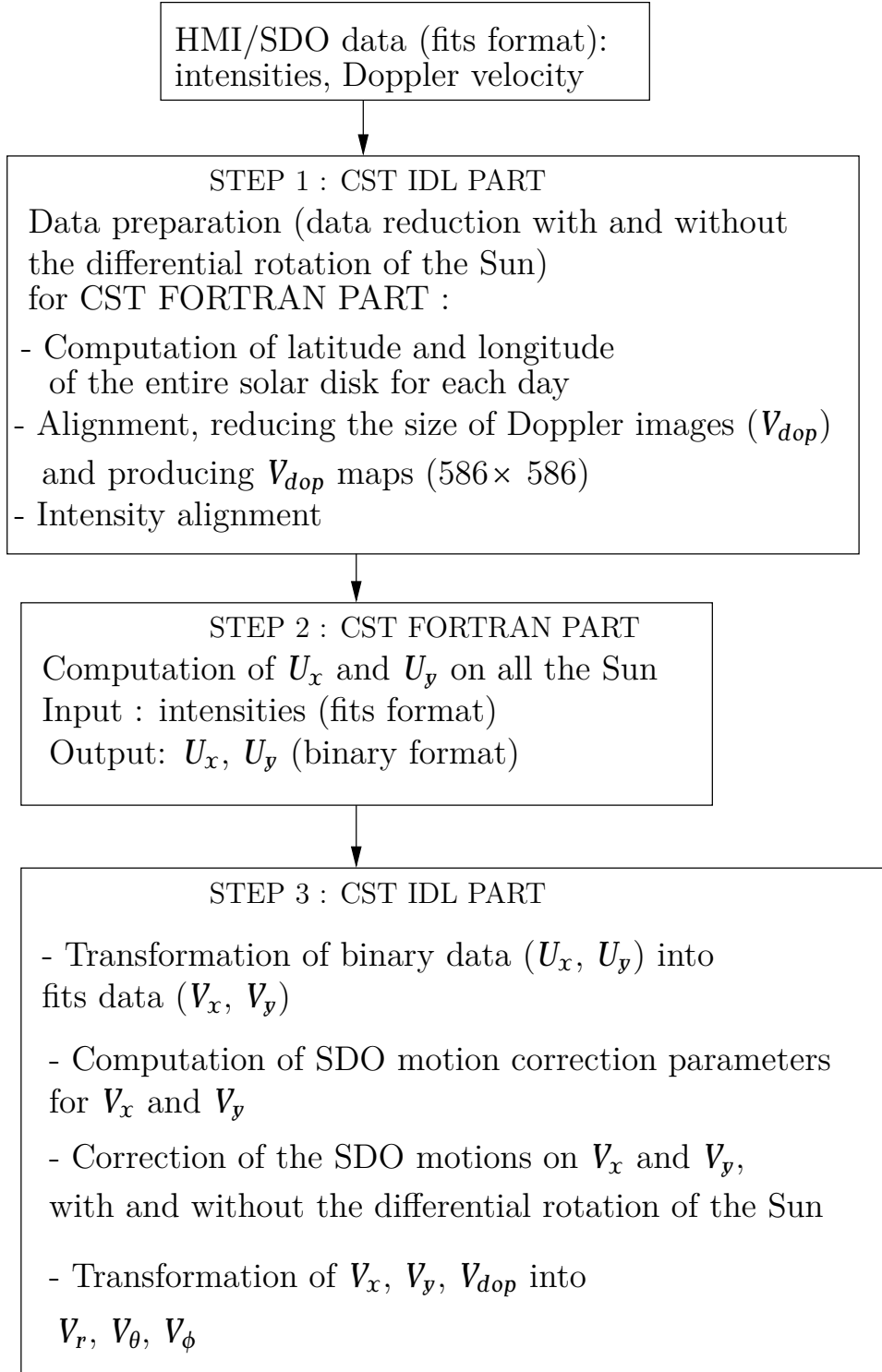


Figure 1: Scheme of CST codes

- computation of latitude and longitude of the entire solar disk for each observation day
- alignment and resizement of the Doppler data. In output files, North is at the bottom and file size is 4096 x 4096 pixels
- SDO motion correction : we correct satellite motion on Doppler data. In output files, North is at the top and the file size is 4096 x 4096 pixels
- limbshift correction : we measure the limbshift; we calculate circular average and we correct the Doppler data for each pixel of the Sun. In output files, North is at the top and file sizes are 586 x 586 pixels and 4096 x 4096 pixels
- average over 30 min of Doppler data: available file V_{dop} with solar rotation. In output files, North is at the top and file size is 586 x 586 pixels
- measure of the solar rotation from Doppler data (or use the standard solar rotation)
- derotation of Doppler data (see figure 2). In output files, North is at the top and file size is 586 x 586 pixels
- average over 30 min of Doppler data: we produce V_{dop} maps without solar rotation. In output files, North is at the top and file size is 586 x 586 pixels
- intensity alignment : recentering and derotating in the same way as before for Doppler data,
 - ★ with rotation (aligned and resized, North top)
 - ★ derotation of the intensity (North top)

The output file size is 4096 x 4096 pixels.

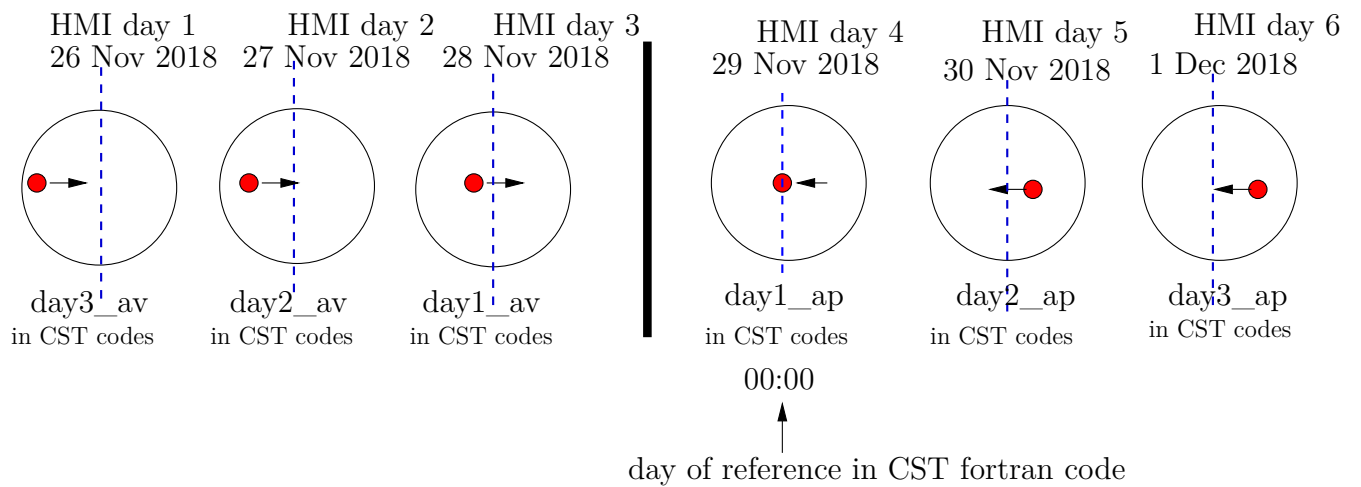


Figure 2: Derotation of Doppler data.

Remark 3.2 *CST codes can be run up to 6 (HMI observation) days : the reference day for CST Fortran code is the 4th observation day (the reference of the center of the Sun and the solar radius are taken at 00:00 of the 4th observation day). This remark has a sense when running CST codes (after step 1, data reduction). For more details, see section 5. Figure 8 illustrates, as example, 6 days from November 26 to December 1, 2018. To run CST codes for 4 observation days, the 2nd observation day is the reference day. Figure 6 illustrates, as example, 4 days from November 28 to December 1, 2018. Figures 7, 5, 4 and 3 illustrate respectively, as example, 5 days, 3 days, 2 days and 1 day.*

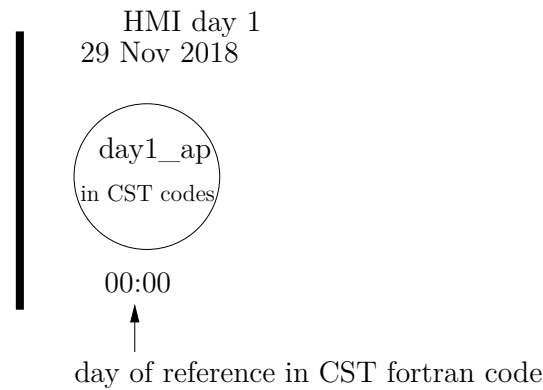


Figure 3: CST run for 1 observation day. The observation day (November 29, 2018) is the reference day.

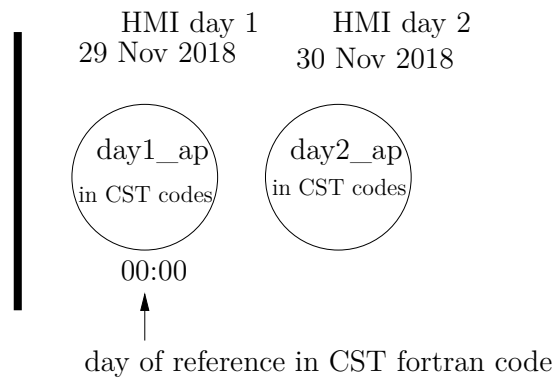


Figure 4: CST run for 2 observation days. The 1st day (November 29, 2018) is the reference day.

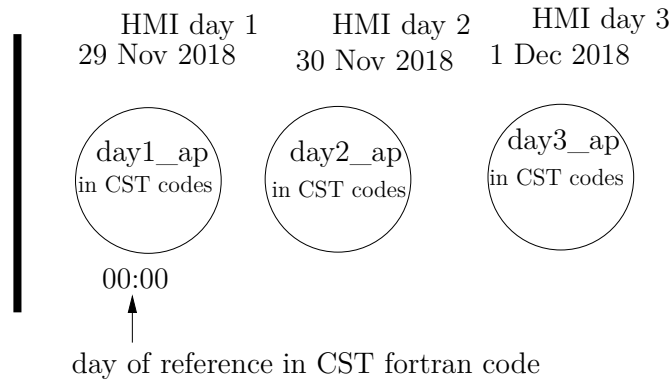


Figure 5: CST run for 3 observation days. The 1st day (November 29, 2018) is the reference day.

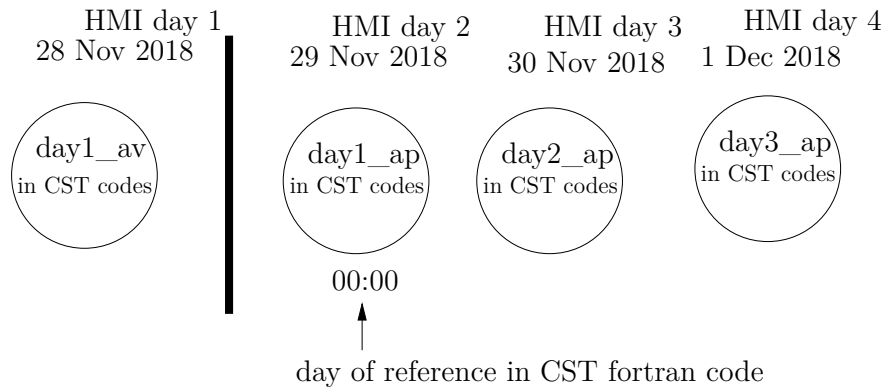


Figure 6: CST run for 4 observation days. The 2nd day (November 29, 2018) is the reference day.

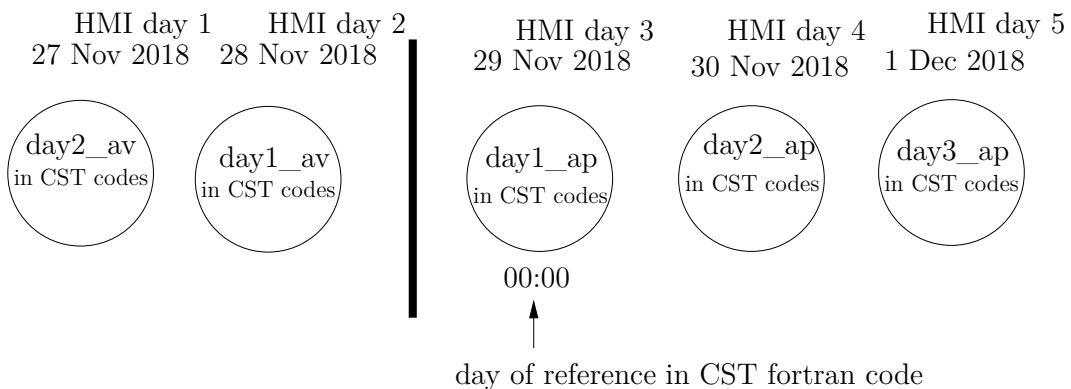


Figure 7: CST run for 5 observation days. The 3rd day (November 29, 2018) is the reference day.

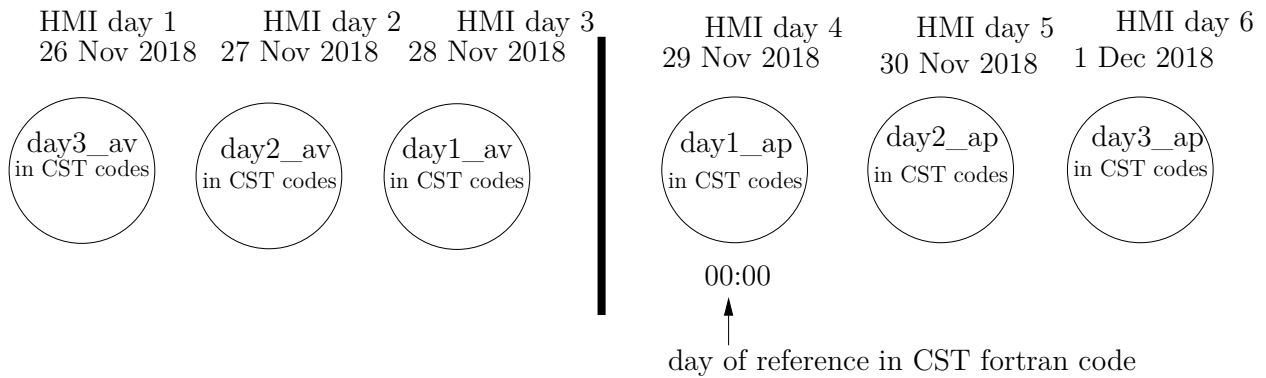


Figure 8: CST run for 6 observation days. The 4th day (November 29, 2018) is the reference day.

Step 2 : CST Fortran part

The second step computes the velocities (U_x and U_y) on all the Sun for a temporal window of 30 min from intensity images. It uses CST Fortran code. Two possible choices : data with solar rotation or data without solar rotation.

The principle of this step (of the CST) is to follow granules in order to define their lifetime and coherence in time. More specifically :

- assume that granules are objects advected by an underlying flow which we wish to measure
- the lifetime of a coherent object (granule) is defined between its appearance and disappearance
- if granule splits : the life is stopped and children = new granules (see t_3 and t_4 on figure 9)
- if granules merge : the lives of granules that merge are stopped and the new granule issued from merging = new granule (see t_1 and t_2 on figure 9).

We can follow coherent structures between their birth and death (see figure 9).

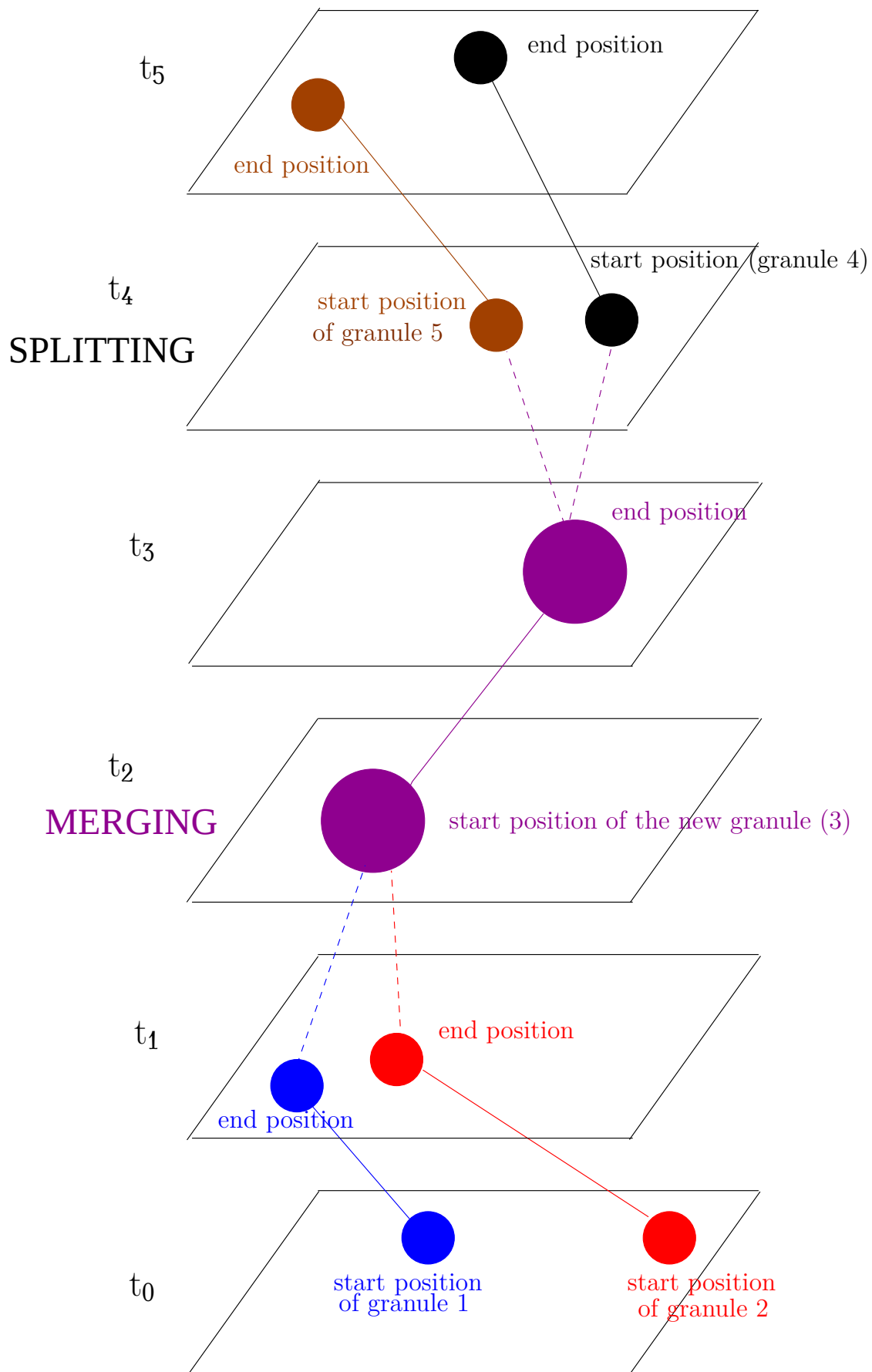


Figure 9: Step 2 (CST) principle.

During their lifetime, granules can split or merge into multiple objects. Disappearance or appearance between two successive frames must be taken into account. Then, the life of coherent objects (i.e granules) is defined between its appearance and disappearance if the granule does not split or merge. When the granule splits, the life of granule is stopped and its children are considered as new granules. In the same way, when granules merge, the lives of the granules that merge are stopped and the new granule issued from the merging is considered as a new granule. Thus, we can follow a coherent structure between their birth and death.

For each detected granule, velocity between its start position and end position is measured. Velocities which are in spatial windows $2,5 \text{ Mm} \times 2,5 \text{ Mm}$ and over 30 minutes (temporal windows) are averaged (Roudier et al. (1999), Roudier et al. (2012)).

Step 3 : CST IDL part

The third step uses IDL codes to transform velocity binary data (U_x and U_y) into fits data (V_x and V_y); to compute SDO motion correction parameters on V_x and V_y ; to correct SDO motions on V_x and V_y with and without the differential rotation of Sun; to transform V_x , V_y , V_{dop} into V_r , V_{phi} , V_{theta} (with and without rotation).

Figure 10 is an example of results provided by step 3 of CST codes. It represents the module of (V_ϕ, V_θ) obtained for three hours of HMI data, where supergranulation is visible.

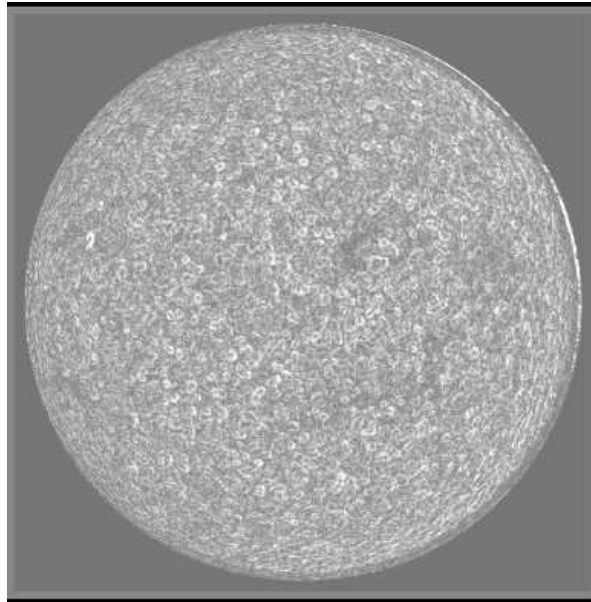


Figure 10: Module of (V_ϕ, V_θ) for three hours of HMI data (step 3).

Formulas used in IDL and Fortran codes can be found in Rincon et al. (2017).

In section 4, we explain how to take HMI/SDO data from JSOC. In section 5, we explain how to run CST software. In section 6, we describe in details input file to modify. In section 7, we give an example of CST software run applied to 30 min HMI observations.

4 Procedure to take SDO/HMI data from JSOC

45s-cadence HMI files are not available at [MEDOC](#).

An automatic procedure to take SDO/HMI data from [JSOC](#) is proposed by T. Corbard (OCA). "codes_data_JSOC" directory contains 3 codes (exportfile.csh, url_escape.pl, gethmi_cst.pro) to run in order to download automatically SDO/HMI data (29 November, 2018, 8:00 to 8:30, for example).

In order to get the HMI data in the directory /data/cluster/workshop_CST2020/day1_apres/, type the following commands in the directory which contains the three codes (exportfile.csh, url_escape.pl, gethmi_cst.pro):

1. `chmod +777 exportfile.csh`
2. Then with IDL :

```
> .r gethmi_cst.pro  
> gethmi_cst,'your_registered_JSOC_email_adress',date_s=[11,29,2018,8,0],  
date_e=[11,29,2018,8,30],out_dir='/data/cluster/workshop_CST2020/  
day1_apres/',/exec
```

For the brave ones, here is the manual procedure to take HMI data directly from JSOC website. But you can see that automatic procedure introduced above is certainly the fastest one.

1. On the [JSOC](#) webpage, click on data access and then in look data (top of the page)
2. click on Fetch seriesname list
3. select hmi.lc - continuum intensities with a cadence of 45s
4. write for example 30 min on November, 29, 2018:

```
hmi.lc_45s[2018.11.29_08:00_TAI-2018.11.29_08:30_TAI]
```

or for one day: `hmi.lc_45s [2018.11.29/1d]`.

See examples in the web page for different requests

5. click on GetRecordCount (this gives you the number of files requested)
6. click on Export Data (top of the page)
7. click on Export (on the right)
8. click on method and get url-tar

-
9. fill the notify with your email address. An email will be sent to you to confirm your registration
 10. fill Requestor with your name
 11. click on Check Params for Export
 12. when green, click on Submit Export Request
 13. an email will be sent to you with the link for downloading the data in tar format: JSOC_20191104_695.tar. Click on tar file link to download on your PC or type the linux command to get results from a server, for example:

```
wget http://jsoc.stanford.edu/SUM85/D1231699647/S00000/JSOC_20191104_695.tar
```

Type the following linux command:
`tar -xvf JSOC_20191104_695.tar`

You get 41 files from hmi.lc_45s.20181129_080000_TAI.2.continuum.fits
up to hmi.lc_45s.20181129_083000_TAI.2.continuum.fits.

For 1 (HMI observation) day, we get 1920 files (one file each 45 seconds)

14. repeat operation at point 4 to download Doppler hmi.V_45s - Dopplergrams with a cadence of 45s.

Data size for 30 min HMI observations (intensity and Doppler) is 2.7 G.
Data size for 1 day HMI observations (intensity and Doppler) is 63 G.

5 Running CST

- Download the package source file CST-V1.1.tgz from [MEDOC](#) webpage
- ifort compiler and SSIDL software are required. Type the following linux commands:
- `tar -xvzf CST-V1.1.tgz`
- `cd CST-V1.1`
- The directory contains 1 directory (codes_data_JSOC) for taking automatically HMI data from JSOC and 2 directories corresponding to 2 different cases : a test case (29 Nov 2018 HMI observations of 30 min) to be able to check if your results are good ("CST_TEST_30min" directory), source codes to treat from 1 to 6 observation days ("CST_1_TO_6_DAYS" directory, adapted for the day 29 Nov 2018 HMI observations).
- Each of these directories (corresponding to the 2 different cases above) contains the following files:

-
1. Fortran files ("codes_CST_Fortran" directory):
 - cst_labv7_FS_2017.f90 (main program)
 - Makefile.common (used to compile librairies that are in the Lib directory)
 - deriw2d.f90
 - detect.f90
 - div_curl.f90
 - mres2d.f90
 - opt_scale.f90
 - the_name.f90 (contains thename module)
 - segment.f90 (contains segmentation module)
 - cg.f90
 - interpol2d.f90
 - interpol2dth.f90
 - interpol2d_routine.f90 (contains interpolationroutine module)
 - interpol2d_routine_short.f90
 - centre_gravite.f90
 - input.f90 (contains input subroutine)
 - sub_coef_r.f90
 - fftw3.inc, form_num.inc, Makefile.inc
 - "lib" directory contains librairies needed for compilation
 - Makefile (file to modify): variables CFLAGS and CFLAGS_LIB are to be adapted according to your server (-I/usr/include/x86_64-linux-gnu/c++/8)
 - script_29nov2018.sh (file to modify): according to your server, you need to add or not the following commands (see section 6 for more details) :
"source /opt/intel/compilers_and_libraries_2018.3.222/..." : path to Fortran ifort compiler
"ulimit -s unlimited"
 - param_seq_29nov2018_EOS_30mn (file to modify) :
the following parameters are to be adapted according to the HMI/SDO data:
prefix, input_file, arcsec, pixel (see section 6 for more details)

2. IDL files (codes_CST_idl directory):
 - reduction_doppler_intensity_all_days_apres.pro
 - reduction_doppler_intensity_all_days_avant.pro
 - step3_CST_IDL_apres.pro
 - step3_CST_IDL_avant.pro
 - image_cont.pro : to read the binary file "image_cont" (see STEP 2)

- take HMI intensity and Doppler data from JSOC (45s-cadence HMI files are not available at MEDOC). For more details, see section 4.
- CST codes can be run up to 6 days (cf "CST_1_TO_6_DAYS" directory): the reference day is the 4th day (the reference of the center of the Sun and the Solar radius are taken at 00:00 of the 4th day). More precisely, for 6 days, transfer the intensity and Doppler data in each directory corresponding to the day of the sequence:

-
- “day3_avant” directory: 1st day data
 - “day2_avant” directory: 2nd day data
 - “day1_avant” directory: 3rd day data
 - “day1_apres” directory: 4th day data (reference day)
 - “day2_apres” directory: 5th day data
 - “day3_apres” directory: 6th day data

First, you must create these directories and second, put data inside.

Note : To run CST codes for 5, 4, 3, 2 and 1 days, see remark [3.2](#)

- For the test case of 30 min HMI observations, you must create “day1_apres” directory and put the HMI data (intensity and Doppler) inside.
- Create the result directories “treated_day1_ap” for 1 observation day, “treated_day1_ap” and “treated_day2_ap” for 2 observation days, ...
- Adapt the following parameters in reduction IDL files (file name starting with “reduction”):
 path, path_out, nombre_j, rota (=0 for standard rotation (quiet Sun) or =1 for rotation measured directly on the data), path for tmp (temporary directory) and check that tmp directory has free space in memory

▪ **STEP 1: CST IDL PART**

To prepare input data for CST Fortran program, type the following commands with SSWIDL for the test case (30 min HMI observations), in “codes_CST_IDL” directory:

1. `.r reduction_doppler_intensity_all_days_apres.pro`
2. `.r reduction_doppler_intensity_all_days_avant.pro` (depending of the treated case)

For 1 to 6 days HMI, type the following commands with SSWIDL:

1. `.r reduction_doppler_intensity_all_days_apres.pro`
2. `reduction_doppler_intensity_all_days_apres.pro,nombre_j,rota`
3. `.r reduction_doppler_intensity_all_days_avant.pro` (depending of the treated case)
4. `reduction_doppler_intensity_all_days_avant.pro,nombre_j,rota` (depending of the treated case)

The order in which the IDL codes are run is important: the reference day is chosen in the first code “reduction_doppler_intensity_all_days_apres.pro”.

For example, for 1 observation day, the output files are (in FITS format) in “treated_day1_ap directory”:

- co_latitude_HMI_4096.fits
- co_latitude_HMI_586.fits
- day1_apres_int_derot_000.fits, ..., day1_apres_int_derot_1909.fits (inputs for CST Fortran code)
- Doppler_derot_30mn.dat
- Doppler_derot_raw_0001.fits, ..., Doppler_derot_raw_1909.fits
- Doppler_derot_smooth_0001.fits, ..., Doppler_derot_smooth_1910.fits
- Doppler_limb4096_0001.fits, ..., Doppler_limb4096_1910.fits
- Doppler_raw_0001.fits, ..., Doppler_raw_1910.fits
- Doppler_smooth_0001.fits, ..., Doppler_smooth_1910.fits
- Doppler_with_rotation.dat
- latitude_HMI_4096.fits
- latitude_HMI_586.fits
- longitude_HMI_4096.fits
- longitude_HMI_586.fits
- SDO_Dop_cormvt_0001.fits, ..., SDO_Dop_cormvt_1910.fits
- SDO_seq_doppler_0001.fits, ..., SDO_seq_doppler_1910.fits
- SDO_seq_int_avec_rot_0001.fits, ..., SDO_seq_int_avec_rot_1909.fits

If there are not 1920 files (corresponding to 1 day), it is necessary to complete from the last (copy the last file). So we have a measure of the last half hour with a lower amplitude.

For 30 min HMI observations, we have:

- ...
- day1_apres_int_derot_000.fits, ..., day1_apres_int_derot_41.fits
- ...

For 6 days HMI observations, we have:

- ...
- day1_apres_int_derot_000.fits, ..., day1_apres_int_derot_1909.fits
- day2_apres_int_derot_000.fits, ..., day2_apres_int_derot_1909.fits
- day3_apres_int_derot_000.fits, ..., day3_apres_int_derot_1909.fits
- day1_avant_int_derot_000.fits, ..., day1_avant_int_derot_1909.fits
- day2_avant_int_derot_000.fits, ..., day2_avant_int_derot_1909.fits
- day3_avant_int_derot_000.fits, ..., day3_avant_int_derot_1909.fits
- ...

If there are not 1920 files for each day, it is necessary to complete from the last (copy the last file).

Warning: Data reduction uses the same “temporary” directory (/tmp) when reading SDO data. So you should not run the IDL code for 2 data sets at the same time, otherwise the SDO decompression files (in /tmp directory) will be destroyed by the 2nd IDL run.

▪ STEP 2: CST FORTRAN PART

Choose 18 threads for an optimal run. To compile and execute Fortran files, here are the commands for SLURM, in “codes_CST_Fortran” directory:

```
sbatch script_20nov2018.sh
```

The output files are for each 30 mn (in “JOB_XXXX/results” directory) :

- output.log : to check if CST code runs well
- image_cont: last segmented image of the Sun in binary format (visualized by an IDL program “image_cont.pro”(with SSWIDL : .r image_cont.pro))
- param_seq_ddmmyyyy_EOS_30mn : parameters used
- traject_11_0000: trajectories of all selected granules. The second number of the first line is the total number of treated granules. Column 1 is the granule number, column 2 is x_cent (gravity center), column 3 is y_cent, column 4 is the number of the image where the granule is born, column 5 is the number of the image where the granule dies, column 6 is the lifetime of the granule in second, column 7 is the velocity ux (in km/s), column 8 is the velocity uy (in km/s)
- nb_gran_0000: pixel size chosen for the spatial window, 1 arcsec in km, pixel size in arcsec, treatment threshold of CST code
- ux_b_0000, ux_h_0000, ux_l_0000, ux_m_0000, ux_k_0000
- uy_b_0000, uy_h_0000, uy_l_0000, uy_m_0000, uy_k_0000
- div_b_0000, div_h_0000, div_l_0000, div_m_0000, div_k_0000
- rot_b_0000, rot_h_0000, rot_l_0000, rot_m_0000, rot_k_0000
- err_b_0000, err_h_0000, err_l_0000, err_m_0000, err_k_0000
- ux,uy,div,rot, sampled on a regular grid , traj contains the trajectories of each granule.

ux_b_0000 file is for the first 30mn. The last 0 refers to time in “time window unit” (and following ux_b_0001 for the following 30min etc. The last 1 refers to time in “time window unit”).

- raw: b (brut) raw ux and uy from CST (586 x 586 pixels)
- high: h: first high resolution wavelet filter (Daubechies)
- middle: m second wavelet filtering (Daubechies) half resolution of “h”
- large: l third filtering by wavelet (Daubechies) resolution half of “m”
- extra-large: k fourth filtering by wavelet (Daubechies) resolution half “l”

For example the most common with the SDO data the resolution will be (if we take $\text{bin_sp} = 7$ (pixels) $h = 2.5$ Megameters (Mm), $m = 5.0$ Mm, $l = 10$ Mm, $k = 20$ Mm output file speeds (u_x, u_y) on a size of 586×586 pixels, latitude, longitude and Doppler files are also set to this size and directly superimposable.

In the Fortran main program (`cst_labv7_FS_2017.f90`) which calculates horizontal speeds, a circular mask is used to properly zero the data outside the Sun. This mask is fixed once and for all in the program with the following parameters in pixels:

$R_{sol}=1873$ $x_{cent}=2054.47$ $y_{cent}=2048.21$.

As the diameter of the Sun changes with time, it is possible that in some times the mask is a little bit smaller than the solar disk. We can so modify R_{sol} , x_{cent} , y_{cent} if this needed. Anyway even with a mask smaller, the program output V_x and V_y speeds are surimposable to Doppler and latitude and longitude grids in 586×586 pixels.

▪ STEP 3: CST IDL PART

1. Copy all u_x and u_y files from the "JOB_XXXX/results" directory to the "treated_day1_ap" directory. These files are needed as input data for IDL codes above
2. Adapt the following parameters in IDL files (in "codes_CST_IDL" directory):
 - `step3_CST_IDL_apres.pro` : `path` (path to original fits HMI files), `path_out` (path to result directory : `treated_dayX_ap`), `path_vit` (directory where are velocities u_x, u_y with and without rotation. Generally, it is the same path as `path_out`), `derot` (`=1` : velocities without solar rotation; `= 0` : velocities with solar rotation), `nombre_j` (number of days)
 - `step3_CST_IDL_avant.pro` : `path` (path to original fits HMI files), `path_out` (path to result directory : `treated_dayX_ap`), `path_vit` (directory where are velocities u_x, u_y with and without rotation. Generally, it is the same path as `path_out`), `derot` (`=1` : velocities without solar rotation; `= 0` : velocities with solar rotation), `nombre_j` (number of days)
3. With SSWIDL, type the following commands:
 - `.r step3_CST_IDL_apres.pro`
 - `.r step3_CST_IDL_avant.pro` (depending of the treated case)

For example for 6 days HMI observations, the output files are for each day after and before :

- `Vr_Vtheta_Vphi_derot_day1_ap.dat`
- `Vr_Vtheta_Vphi_derot_day2_ap.dat`
- `Vr_Vtheta_Vphi_derot_day3_ap.dat`
- `Vr_Vtheta_Vphi_derot_day1_av.dat`
- `Vr_Vtheta_Vphi_derot_day2_av.dat`

- Vr_Vtheta_Vphi_derot_day3_av.dat

Warning: IDL codes of step 3 use the same “temporary” directory (/tmp) when reading SDO data. So you should not run the IDL code for 2 data sets at the same time, otherwise the SDO decompression files (in /tmp directory) will be destroyed by the 2nd IDL run.

For example, “treated_day1_ap” directory (for 30 min HMI observations) size is 19 G and “JOB_XXXX” directory size is 590 M (in “codes_CST_Fortran” directory).

For example, “treated_day1_ap” directory (for 1 day HMI observations) size is 854 G and “JOB_4174” directory size is 11 G (in “codes_CST_Fortran” directory).

As the volume of the result data is huge (“treated_day1_ap” directory), we will give only one result file of step 2 (Fortran part) corresponding to 30 min HMI observations: “output.log” in “JOB_XXXX” directory.

6 Description of input file

We describe below variables used in param_seq_29nov2018_EOS_30mn file corresponding to “29 November, 2018” HMI data. For example, we have 40 HMI files (one file each 45 seconds) for 30 minutes of data.

The following variables are common to all HMI data :

- nconv : convolution window for smoothing starting HMI images (before segmentation). nconv = 1 (default value) is the best signal-to-noise ratio
- option = vitesse : default value
- istep = 1 (fixed value) : increment step for HMI images i.e. temporal step between 2 images (all images are taken into account)
- nb_series = 1 : default value. We have only 1 sequence of images (here 30 minutes of HMI data)
- ngmax0 = 900000 : maximum number of granules per image
- ngmax = 25000000 : maximum number of granules tracked during the 30-minute sequence (i.e. 40 images)
- np = 40 : number of images corresponding to 30 minutes of HMI data
- n_ech = 40 : default value. np = n_ech
- n_life = 60 : maximum granule lifetime. A granule is tracked on maximum 60 images

-
- `bin_sp = 7` (default value) : space bin in pixel (i.e. 2.5 Megameters)
 - `v_thres = 7.0` (fixed value) : speed of sound in photosphere (maximum speed in km/s considered here)
 - `t_thres = 180.` : minimum granule lifetime in seconds. 180 s correspond to 4 HMI images. If a granule is only tracked in 3 images, then it is not taken into account. `t_thres = 180` is the best signal-to-noise ratio
 - `seuil_min = 0.01` (default value) : we consider here all pixels
 - `segmentation_type = strouss` (default value) : we don't use Strauss segmentation method here but CST segmentation method
 - `suffix = .fits` : file format
 - `verbose = 1` : computation details printed in `output_XXXX.log` file (in `JOB_XXXX` directory). If `verbose = 0`, there is no `output.log`

The following variables are corresponding to 29 November, 2018 HMI data :

- `prefix` : path to "day1_apres_int_derot_" FITS files (results of IDL Step 1)
- `input_file = param_seq_29nov2018_EOS_30mn`
- `arcsec = 715.403` : value of 1 arcsec in km which is obtained by the following way :
 $715.403 = R_S(\text{km}) / R_S(\text{arcsec})$,
 where $R_S(\text{km}) = 696000$ is the solar radius in km, $R_S(\text{arcsec}) = 972.877991$ is the solar radius in arcsec (provided by the header of the 1st HMI image (in fits format) of sequence for each day (`RSUN_OBS=INDEX.RSUN_OBS`))
- `timestep = 45.` : time (cadence) in seconds between 2 HMI images
- `pixel = 0.504008` : pixel size in arcsec provided by the header of the 1st HMI image of sequence for each day (`pix=INDEX.CDELTA1`)

7 Example: CST codes applied to 30 minutes HMI observations

Source codes are in "CST_TEST_30min" directory. You can run CST codes for any half hour. We consider here the following case : November 29, 2018 from 8:00 to 8:30. There are 41 intensity files and 41 Doppler files (provided by JSOC), in FITS format. These data are to be put in a directory named "day1_apres" (2.7 G).

The first step prepares input data for CST Fortran program (data reduction) by first running with SSWIDL "reduction_doppler_intensity_all_days_apres.pro" (then "reduction_doppler_intensity_all_days_avant.pro"), in "codes_CST_IDL" directory.

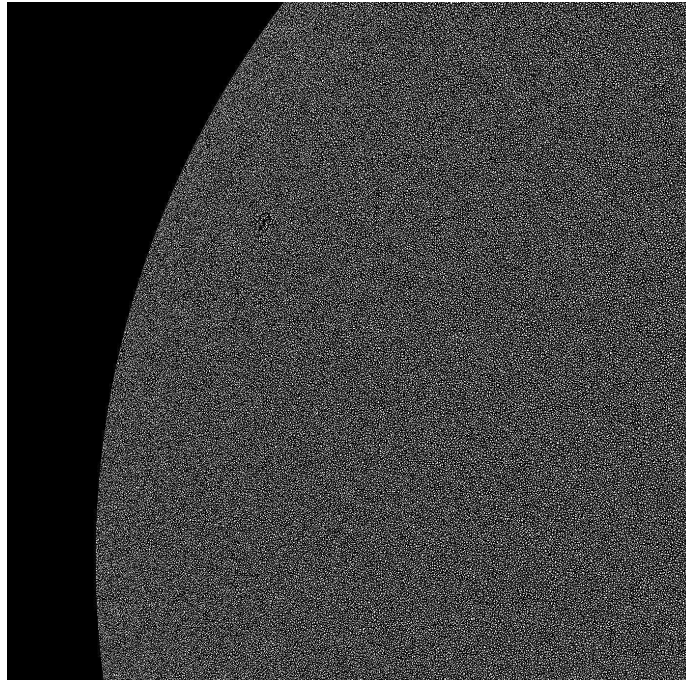


Figure 11: Solar segmented granules close to the limb - observations on October 21, 2010

The results (in FITS format) are in “treated_day1_ap” directory (19 G).

You can run now the second step i.e. CST Fortran part by typing the following command, in “codes_CST_Fortran” directory, for example with SLURM :

```
sbatch script_29nov2018.sh
```

A SLURM file (“slurm-5617.out”) corresponding to the JOB number (5617 in our case) is created, as well as “JOB_5617” directory which contains the results (see section 5 for more details on each result files) and “output_5617.log” file. You can check the progress of the code in “output_5617.log”.

When the run of step 2 is finished, copy all ux and uy files “from JOB_5617/results” directory to “treated_day1_ap” directory.

Then, run the third step “step3_CST_IDL_apres.pro” with SSWIDL. The results are in “treated_day1_ap” directory.

Figure 11 is the visualization of the binary file “image_cont”.

8 Frequent troubles encountered

When you apply CST codes to 1 day HMI observations or more, sometimes it can happen that CST Fortran part (Step 2) does not calculate all the velocities ux and uy in JOB_XXX/results. Generally, one FITS file (day1_apres_int_derot_xxxx.fits) is corrupted (we don’t know why) but you can overcome that problem by replacing the corrupted file

by the precedent FITS file (which is good) of the series. In order to know which FITS file is corrupted (day1_apres_int_derot_xxxx.fits), you must locate it in “output_XXX.log” file. Then, you run CST Fortran program (step 2) to calculate the missing u_x and u_y . In that case, like for example $NINDEX=0$, in “script_DDMMYYYY.sh”, is switched into $NINDEX=20$ (if ux_0020 is not computed) and we also modify the line :

while [$\$NINDEX \leq 47$] to while [$\$NINDEX \leq 20$] .

9 Important note about the usage of CST codes

We strongly encourage interested scientists to use our codes. However, we shall ask them to comply with the following rules:

1. the first refereed-journal publication of any new user should include both the names of Th. Roudier (with affiliation: Université Paul Sabatier, Observatoire Midi-Pyrénées, Cnrs, Cnes, IRAP, F-31400 Toulouse) and M. Chane-Yook (with affiliation: IAS, Université Paris-Saclay, Cnrs, F-91400 Orsay) among the co-authors;
2. the MEDOC service at IAS (Orsay, France) which distributes these resources should also be properly acknowledged;
3. further publications should be mentioned to thierry.roudier@irap.omp.eu

10 Other applications of CST codes

- Measurement of cell displacement (biology)
- Measurement of any small structure displacement on a surface
- ...

11 CST MEDOC WORKSHOP

A workshop was dedicated to the training and use of the Coherent Structure Tracking (CST) software. It took place on March 3-4, 2020 at **IAS** in Orsay, France. This workshop is open to all researchers and students of the international solar community.

Part of the workshop was devoted to the explanation of the installation of this CST code on their own servers as well as the various precautions to take.

Workshop website: <https://cst2020-medoc.sciencesconf.org/>

Presentations are available on [CST webpage](#) in Workshop section.

12 Acknowledgements

We thank MEDOC, in particular the MEDOC technical team and SDO/HMI team.

References

- M. Rieutord, T. Roudier, H. G. Ludwig, Å. Nordlund, and R. Stein. Are granules good tracers of solar surface velocity fields? *Astron. Astrophys.*, 377:L14–L17, Oct 2001. doi: 10.1051/0004-6361:20011160.
- M. Rieutord, T. Roudier, S. Roques, and C. Ducottet. Tracking granules on the Sun’s surface and reconstructing velocity fields. I. The CST algorithm. *Astron. Astrophys.*, 471(2):687–694, Aug 2007. doi: 10.1051/0004-6361:20066491.
- F. Rincon, T. Roudier, A. A. Schekochihin, and M. Rieutord. Supergranulation and multiscale flows in the solar photosphere. Global observations vs. a theory of anisotropic turbulent convection. *Astron. Astrophys.*, 599:A69, Mar 2017. doi: 10.1051/0004-6361/201629747.
- Th. Roudier, M. Rieutord, J. M. Malherbe, and J. Vigneau. Determination of horizontal velocity fields at the sun’s surface with high spatial and temporal resolution. *Astron. Astrophys.*, 349:301–311, Sep 1999.
- Th. Roudier, M. Rieutord, J. M. Malherbe, N. Renon, T. Berger, Z. Frank, V. Prat, L. Gizon, and M. Švanda. Quasi full-disk maps of solar horizontal velocities using SDO/HMI data. *Astron. Astrophys.*, 540:A88, Apr 2012. doi: 10.1051/0004-6361/201118678.
- Th. Roudier, M. Rieutord, V. Prat, J. M. Malherbe, N. Renon, Z. Frank, M. Švanda, T. Berger, R. Burston, and L. Gizon. Comparison of solar horizontal velocity fields from SDO/HMI and Hinode data. *Astron. Astrophys.*, 552:A113, Apr 2013. doi: 10.1051/0004-6361/201220867.
- Th. Roudier, M. Švanda, J. Ballot, J. M. Malherbe, and M. Rieutord. Large-scale photospheric motions determined from granule tracking and helioseismology from SDO/HMI data. *Astron. Astrophys.*, 611:A92, Apr 2018. doi: 10.1051/0004-6361/201732014.
- R. Tkaczuk, M. Rieutord, N. Meunier, and T. Roudier. Tracking granules on the Sun’s surface and reconstructing velocity fields. II. Error analysis. *Astron. Astrophys.*, 471(2):695–703, Aug 2007. doi: 10.1051/0004-6361:20066492.
- Michal Švanda, Thierry Roudier, Michel Rieutord, Raymond Burston, and Laurent Gizon. Comparison of Solar Surface Flows Inferred from Time-Distance Helioseismology and Coherent Structure Tracking Using HMI/SDO Observations. *Astrophys. J.*, 771(1):32, Jul 2013. doi: 10.1088/0004-637X/771/1/32.