Run CST codes (Coherent Structure Tracking)

Martine Chane-Yook (IAS)
&
Thierry Roudier (IRAP)

March 4, 2020

What do we need ?

1. A computing cluster

What do we need ?

1. A computing cluster
2. CST codes from MEDOC/Tools, then CST codes

What do we need ?

1. A computing cluster
2. CST codes from MEDOC/Tools, then CST codes
3. HMI/SDO 45s-cadence intensity and Doppler data series from JSOC

What do we need ?

1. A computing cluster
2. CST codes from MEDOC/Tools, then CST codes
3. HMI/SDO 45s-cadence intensity and Doppler data series from JSOC
4. Training exercise : running CST codes for 30 min of SDO/HMI observation (data) on November, 29, 2018

What do we need ?

1. A computing cluster
2. CST codes from MEDOC/Tools, then CST codes
3. HMI/SDO 45s-cadence intensity and Doppler data series from JSOC
4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data)
5. How many observation days to run ? from 1 to 6 days

Plan

1. A computing cluster
2. CST codes from MEDOC/Tools, then CST codes
3. HMI/SDO 45s-cadence intensity and Doppler data series from JSOC
4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data)
5. How many observation days to run ? from 1 to 6 days
6. Frequent troubles encountered

Plan

1. A computing cluster
2. CST codes from MEDOC/Tools, then CST codes
3. HMI/SDO 45s-cadence intensity and Doppler data series from JSOC
4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data)
5. How many observation days to run ? from 1 to 6 days
6. Frequent troubles encountered
7. Work in progress

Plan

1. A computing cluster
2. CST codes from MEDOC/Tools, then CST codes
3. HMI/SDO 45s-cadence intensity and Doppler data series from JSOC
4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data)
5. How many observation days to run ? from 1 to 6 days
6. Frequent troubles encountered
7. Work in progress
8. Contacts (SAV)

1. Computing cluster

- CST codes are written in Fortran 90 and IDL
- We need
  - ifort fortran compiler
  - SolarSoft (SSWIDL) software
  - slurm workload manager (for example)
- Morning of workshop 2nd day will be devoted to CST codes installation on your computing cluster

2. CST codes (CST_V 1.0)

- Documentation and Quick start guide are available.

2. CST codes (CST_V 1.0)

- Documentation and Quick start guide are available.
  → *A section devoted to the description of IDL and Fortran subroutines is missing : work is in progress (online soon)*

2. CST codes (CST_V 1.0)

- Documentation and Quick start guide available.
  → *A section dedicated to the description of IDL and Fortran subroutines is missing : work is in progress*
- CST_V1.0 directory contains :
  - a test case CST_TEST_30min using 30 min HMI observation data (Nov, 29, 2018)

2. CST codes (CST_V 1.0)

- Documentation and Quick start guide available.
  → *A section dedicated to the description of IDL and Fortran subroutines is missing : work is in progress*
- CST_V1.0 directory contains :
  - a test case CST_TEST_30min using 30 min HMI observation data (Nov, 29, 2018) → special treatment
  - source codes to deal with 1 to 6 HMI observation days CST_1_TO_6_DAYS

2. CST codes (CST_V 1.0)

- Documentation and Quick start guide available.
  → *A section dedicated to the description of IDL and Fortran subroutines is missing : work is in progress*
- CST_V1.0 directory contains :
  - a test case CST_TEST_30min using 30 min HMI observation data (Nov, 29, 2018) → special treatment
  - source codes to deal with 1 to 6 HMI observation days CST_1_TO_6_DAYS
- each folder contains Fortran files (codes_CST_Fortran directory corresponding to part 2 of CST algorithm) and IDL files (codes_CST_IDL directory corresponding to parts 1 and 3 of CST algorithm)

2. Scheme of CST codes

- STEP 1 (IDL) : codes_CST_IDL directory
  Files : - reduction_doppler_intensity_all_days_apres.pro,
  - reduction_doppler_intensity_all_days_avant.pro
  Input : intensity and Doppler HMI data
  Output/input in STEP 2 :
  dayX_apres_int_derot_NNNN.fits, ...,
  dayX_avant_int_derot_NNNN.fits, ...
  ...

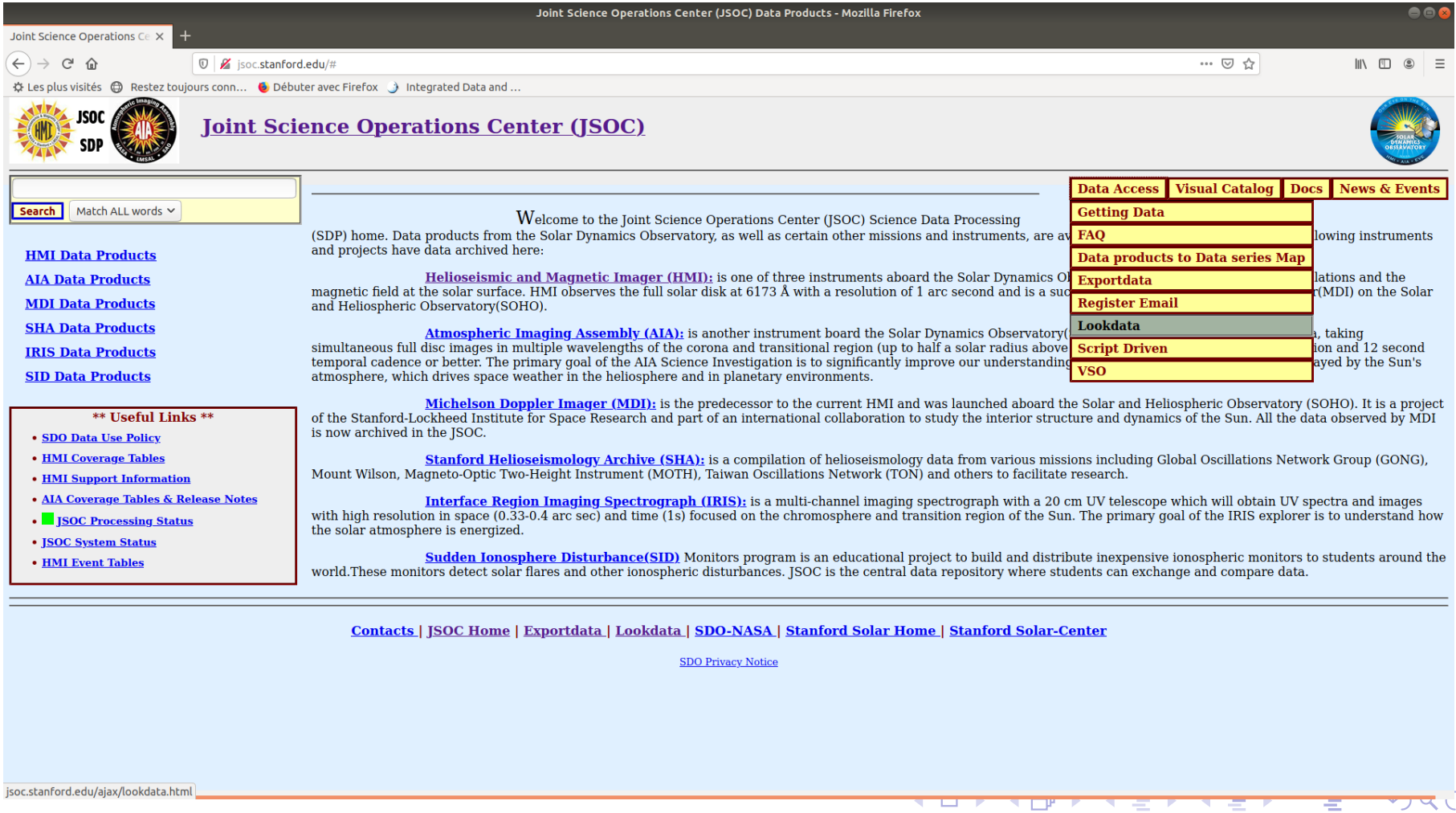
- STEP 2 (FORTRAN) : codes_CST_Fortran directory
  Files : cst_labv7_FS_2017.f90 (main program), Makefile, ...
  Output/input in STEP 3 : ux_ZZZ, ..., uy_ZZZ, ...

2. Scheme of CST codes

- STEP 3 (IDL) : codes_CST_IDL directory
  Files : - transform_uxuy_vxvy_all_dates_step_30min.pro,
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro,
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_avant.pro,
  - transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro
  Output : Vr_Vtheta_Vphi_derot_dayX_ap.dat,
  Vr_Vtheta_Vphi_derot_dayX_av.dat, ...

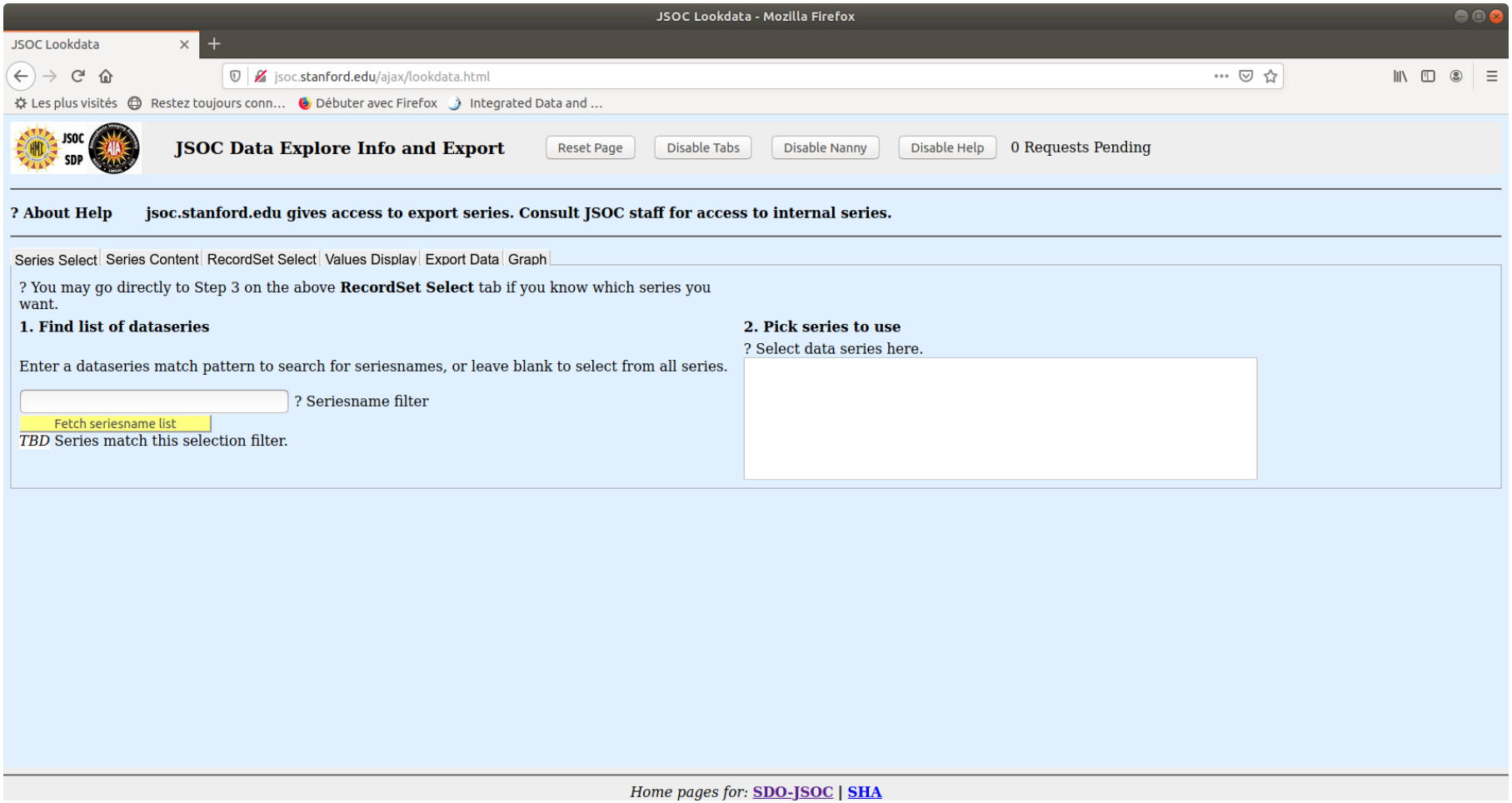
3. Procedure to take HMI/SDO 45s-cadence intensity and Doppler data from JSOC

Click on Data Access then on Lookdata



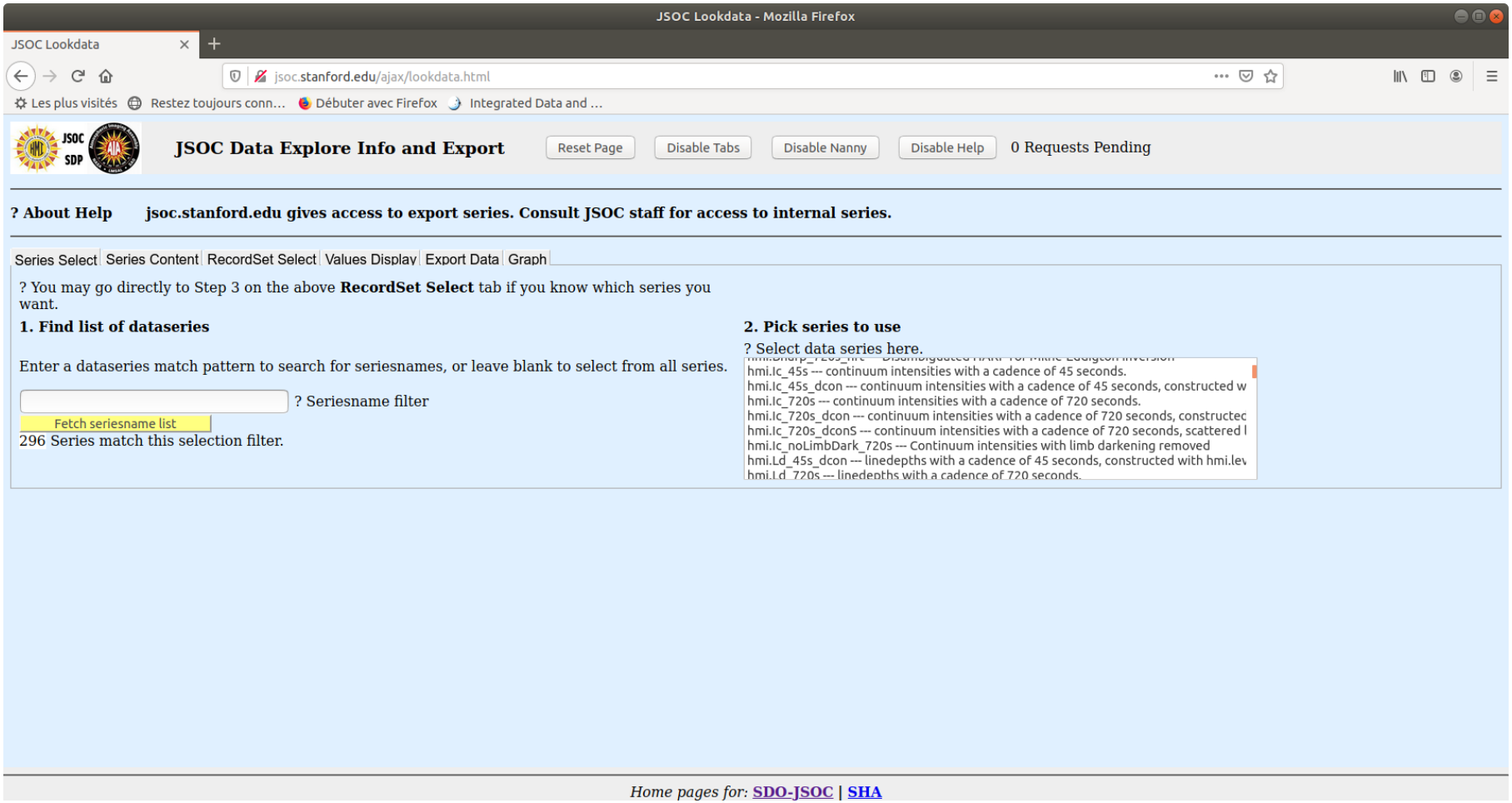
3. Procedure to take HMI/SDO 45s-cadence intensity and Doppler data from JSOC

Click on Fetch seriesname list in yellow or type (on left) Ic_45s



3. Procedure to take HMI/SDO 45s-cadence intensity

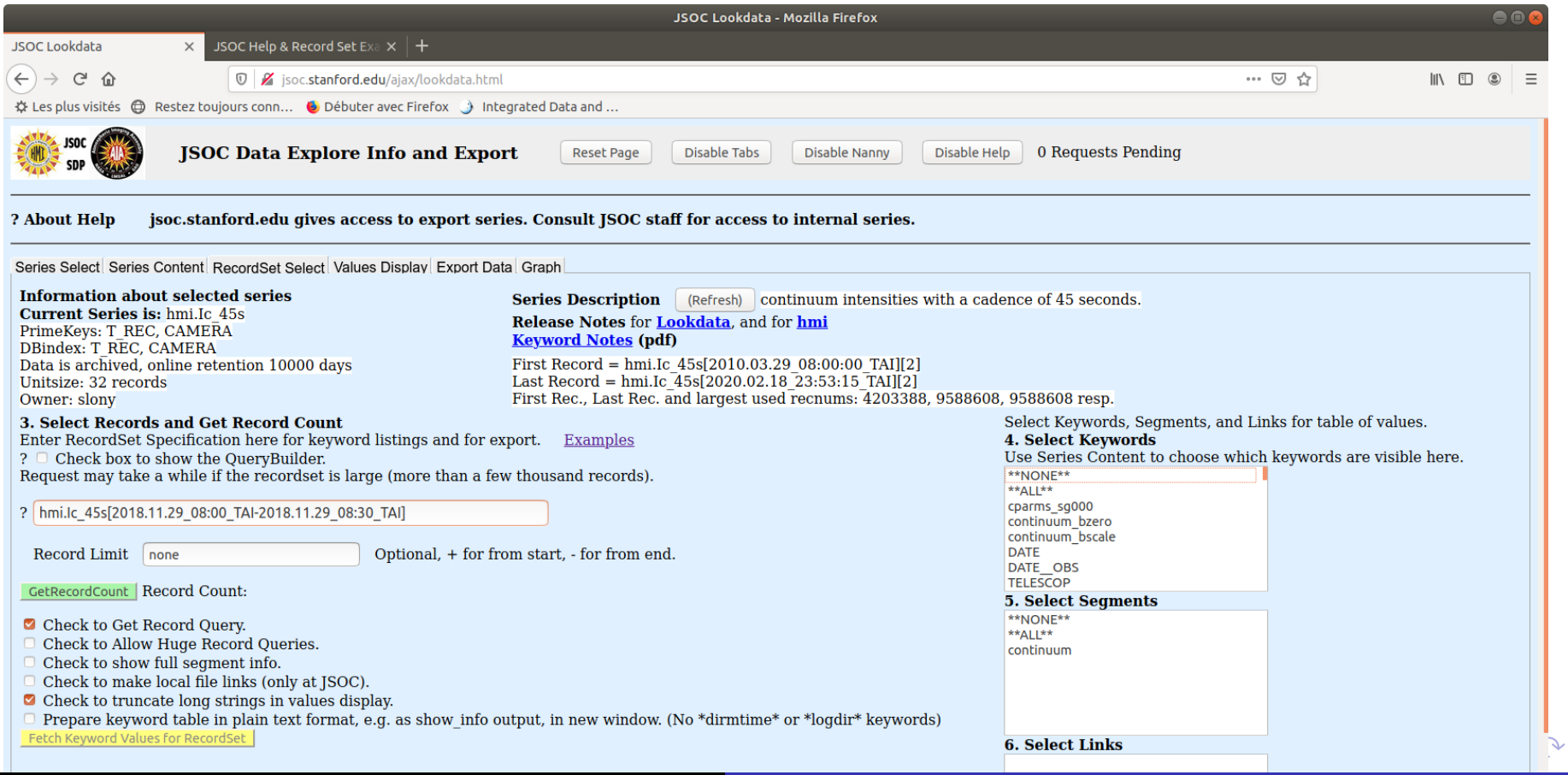
Several series appear on the right window. Select
hmi.Ic_45s–continuum intensities with a cadence of 45 seconds



3. Procedure to take HMI/SDO 45s-cadence intensity

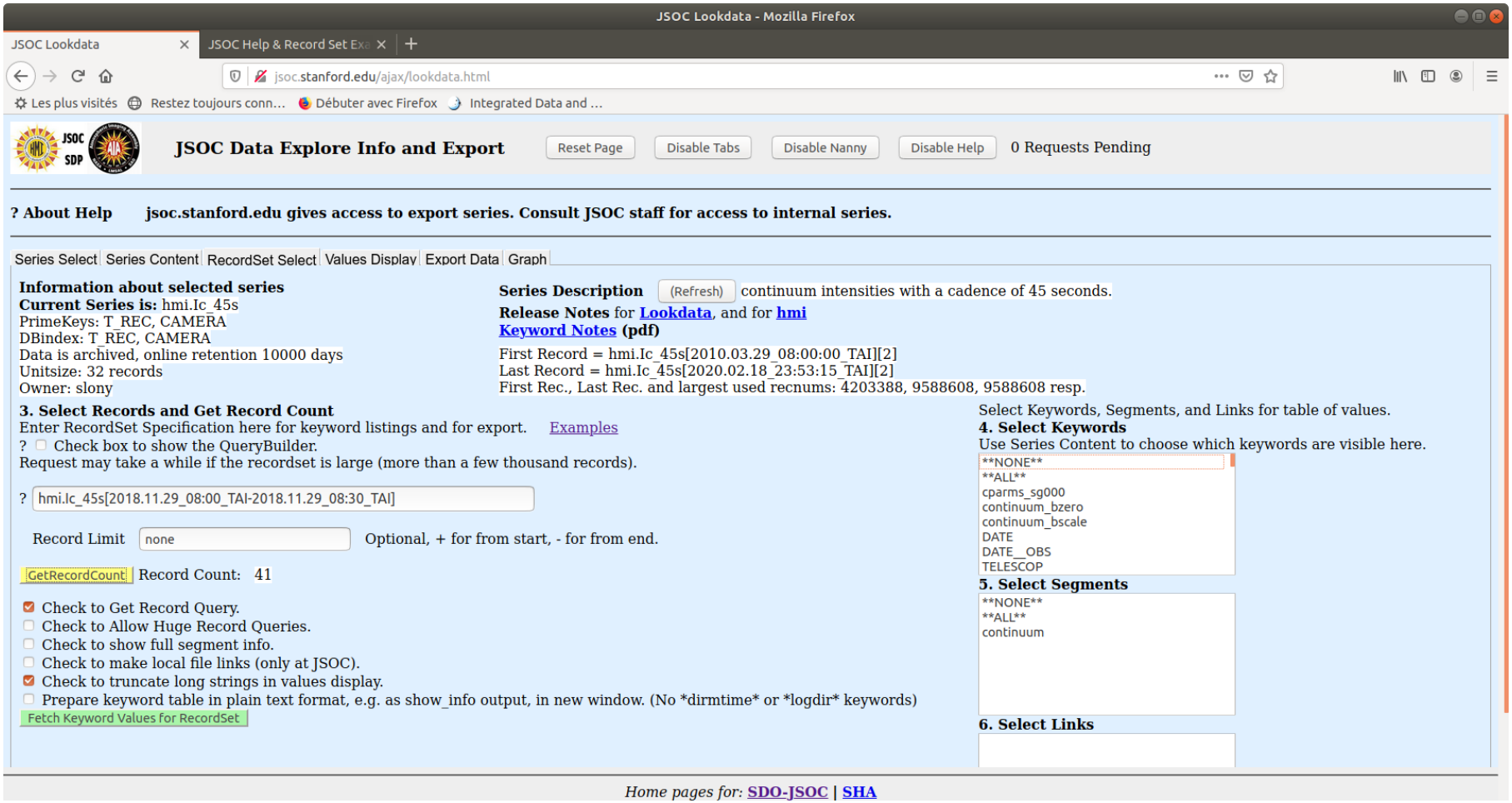
Write for example 30 mins on November, 29, 2018 (see Examples for syntax) :
hmi.Ic_45s[2018.11.29_08:00_TAI-2018.11.29_08:30_TAI]

→ For 1 day : hmi.Ic_45s[2018.11.29/1d]



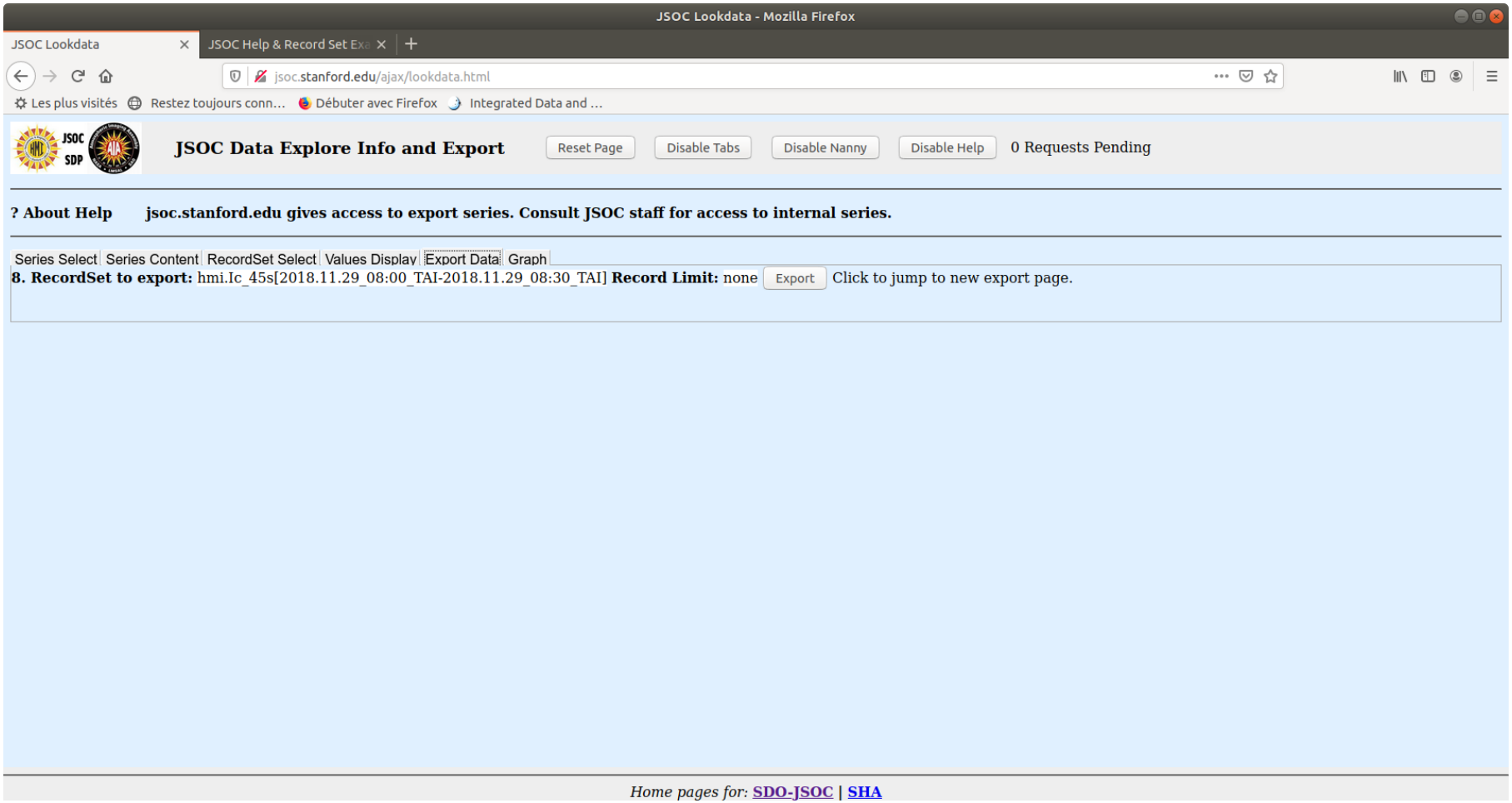
3. Procedure to take HMI/SDO 45s-cadence intensity

Click on GetRecordCount in yellow to obtain the number of files requested. For our example of 30 mins of observation, we have 41 files



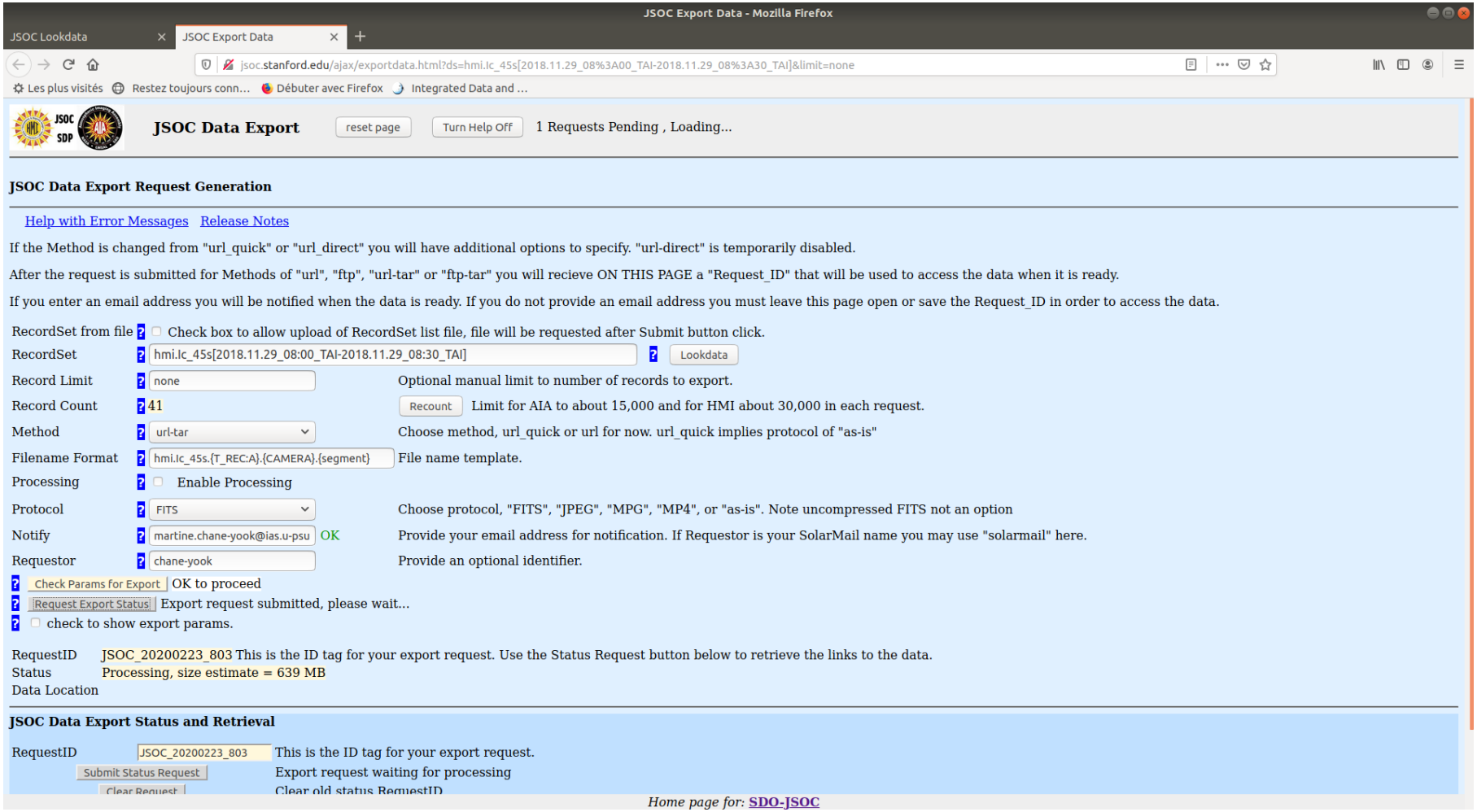
3. Procedure to take HMI/SDO 45s-cadence intensity

Click on Export Data at the top of the page, then on Export on the right



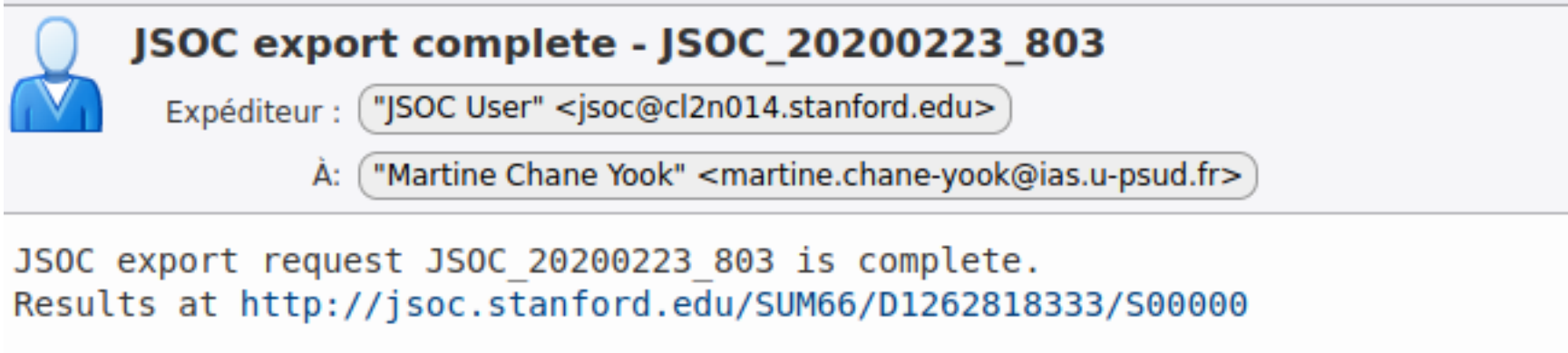
3. Procedure to take HMI/SDO 45s-cadence intensity

Select get url-tar, fill your email adress, your name, click on Check Params for Export, when green click on Request Export Status



3. Procedure to take HMI/SDO 45s-cadence intensity

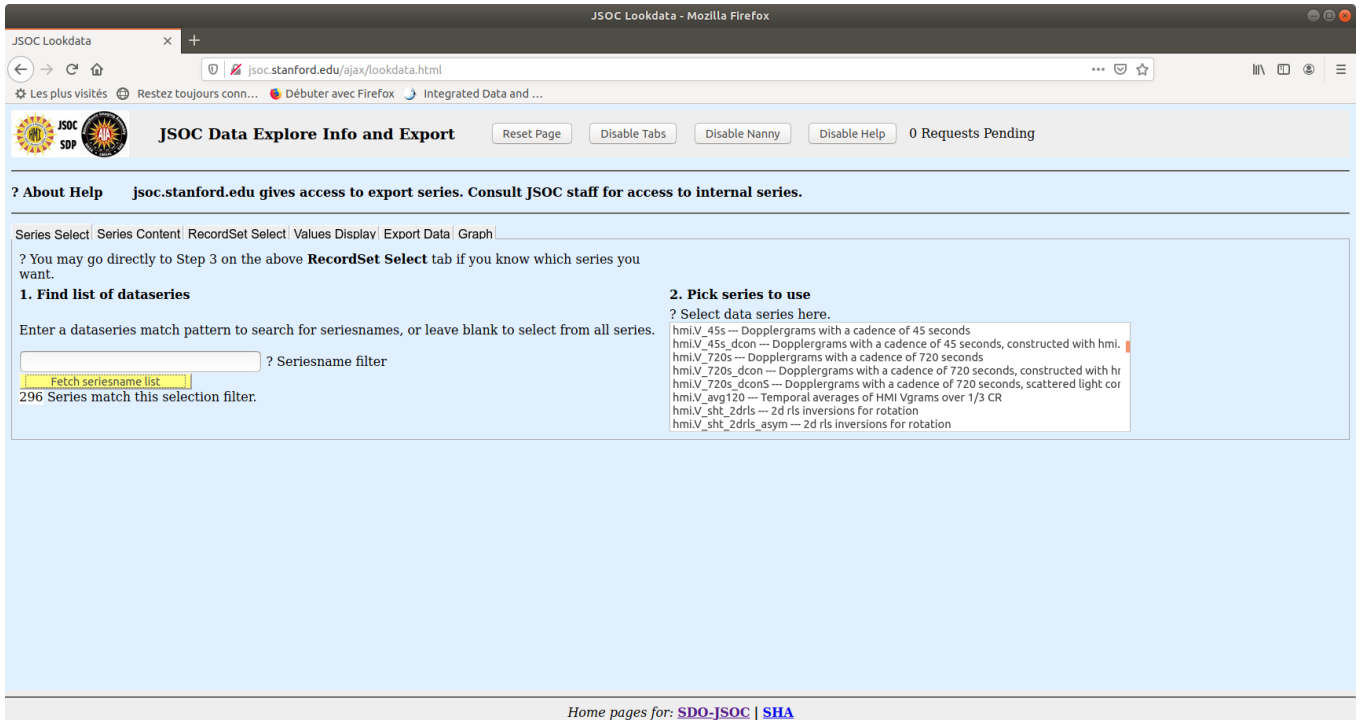
- An email has been sent to you with a link



- To get files from a server, type the following linux commands :
  - wget http://jsoc.stanford.edu/SUM66/D1262818333/S00000/JSOC_20200223_803.tar
  - tar -xvf JSOC_20200223_803.tar
- You get 41 intensity files from hmi.lc_45s.20181129_080000_TAI.2.continuum.fits up to hmi.lc_45s.20181129_083000_TAI.2.continuum.fits

3. Procedure to take HMI/SDO 45s-cadence Doppler

- Repeat the procedure to get Doppler data : select hmi.V_45s–Dopplergrams with a cadence of 45 seconds



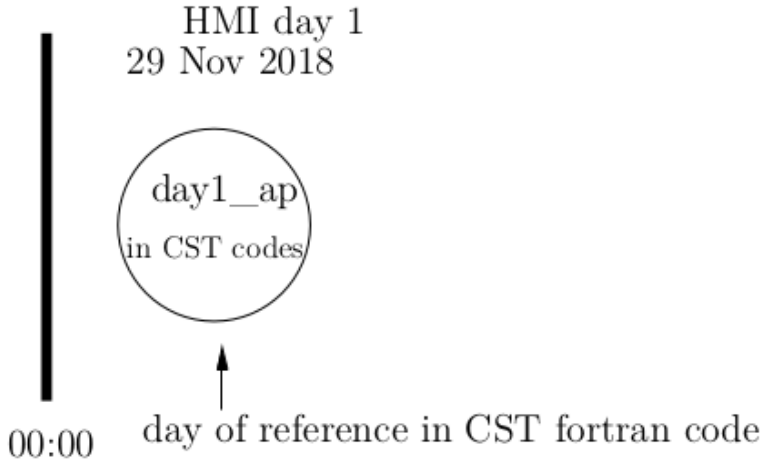
- You get 41 Dopplergram files
- 30 mins HMI data (intensity and Doppler) → 2,7 G
  → see 4. Example of running CST codes (training exercise)
- 1 day HMI data (intensity and Doppler) → 63 G

4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data) - Connection to IAS computing cluster

- login : cst2020-1 → cst2020-5
- password : cst2020@MEDOC!
- Hereafter, all linux commands in this presentation are written in orange
- connection to the cluster :
  ssh -X -l login ias-ssh.ias.u-psud.fr
  ssh -X cluster-r730-2 : connection to a cluster node for running SSWIDL
  ssh -X cluster-head : for running FORTRAN
- Copy CST_TEST_30min directory in your home :
  cp -R /data/cluster/workshop_CST2020/cst2020/CST_TEST_30min .

4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data) - STEP 1 (IDL)

- It's the same case as for 1 day (reference day) according to the figure below



- Create day1_apres directory and put HMI data (intensity and Doppler) inside : done !
  → /data/cluster/workshop_CST2020/day1_apres
- Create result directory treated_day1_ap in your home (/home/cst2020-1:5)
- cd CST_TEST_30min

4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data) - STEP 1 (IDL)

- cd codes_CST_IDL
- Reduction IDL file :
  reduction_doppler_intensity_all_days_apres.pro
- Adapt the 4 following parameters :
  - path : path to HMI data (intensity and Doppler) directory (day1_apres)
  - path_out : path to the result directory (treated_day1_ap)
  - rota=0 (for standard rotation, quiet Sun) or rota=1 (rotation is measured directly on the data)
  - spawn,'rm /tmp/HMI*' (for IAS cluster)
- Limitation of the code : if 2 users read HMI files at the same time on a same cluster node, they try to write the same temporary file, which has always the same name ! It is due to a bad way of managing temporary file with SSWIDL.
  → IDL parts will be written in Python

4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data) - STEP 1 (IDL)

- ssh -X cluster-r730-2 (connection to a node of the cluster)
  cd codes_CST_IDL
  ssw hmi (for IAS cluster)
  .r reduction_doppler_intensity_all_days_apres.pro
- Optimal condition for CPU Time : 8 min
- The result directory (treated_day1_ap) contains :
  - co_latitude_HMI_4096.fits
  - co_latitude_HMI_586.fits
  - day1_apres_int_derot_0001.fits → 0041.fits : input files in STEP 2
  - donnees_correction_CST.dat
  - Doppler_derot_30mn.dat
  - Doppler_derot_raw_0001.fits → 0041.fits
  - Doppler_derot_smooth_0001.fits → 0041.fits
  - Doppler_limb4096_0001.fits → 0041.fits
  - Doppler_raw_0001.fits → 0041.fits

4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data) - STEP 1 (IDL)

- The result directory (treated_day1_ap) contains :
  - Doppler_smooth_0001.fits → 0041.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 → 0041.fits
  - SDO_seq_doppler_0001.fits → 0041.fits
  - SDO_seq_int_avec_rot_0001.fits → 0041.fits
- If there are not 41 files, it is necessary to complete from the last (copy the last file)

4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data) - STEP 2 (FORTRAN)

- ssh -X cluster-head
- cd codes_CST_Fortran
- Files to modify are :
  - param_seq_29nov2018_EOS_30mn : prefix (path to day1_apres_int_derot_ files), input_file, arcsec, pixel
  - script_29nov2018.sh : adapt 4 times the name of param_seq_29nov2018_EOS_30mn, NINDEX=0, while [$NINDEX -le 0] (for 30 min HMI data)
- sbatch script_29nov2018.sh
- There is a defect in the Makefile which obliges to run the sbatch 2 times or more (in order to get executable file "cst_labv7_FS_2017").
  → Makefile will be reviewed and improved to avoid this problem
- Optimal condition for CPU time : $\sim$ 1h - 1h20 (#SBATCH –cpus-per-task=18)

4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data) - STEP 2 (FORTRAN)

- Different files are created : slurm-XXX.out in current folder, output_XXX.log in JOB_XXX directory
- The results are in JOB_XXX/results directory
- Copy the following files from JOB_XXX/results to treated_day1_ap directory :
  - ux_b_0000
  - ux_h_0000
  - ux_k_0000
  - ux_l_0000
  - ux_m_0000
  - uy_b_0000
  - uy_k_0000
  - uy_l_0000
  - uy_m_0000

4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data) - STEP 3 (IDL)

- Reminder : 2 users can't run STEP 3 IDL codes at the same time on a same cluster node because of tempory file problem (see STEP 1)
- Adapt the following parameters :
  - transform_uxuy_vxvy_all_dates_step_30min.pro : path, path_out
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro : path, path_out, path_vit, spawn,'rm /tmp/HMI*'
  - transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro : path1, path, path_out
- ssh -X cluster-r730-2 (connection to a node of the cluster)
  cd codes_CST_IDL
  ssw hmi
  .r transform_uxuy_vxvy_all_dates_step_30min.pro
  .r correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro
  .r transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro

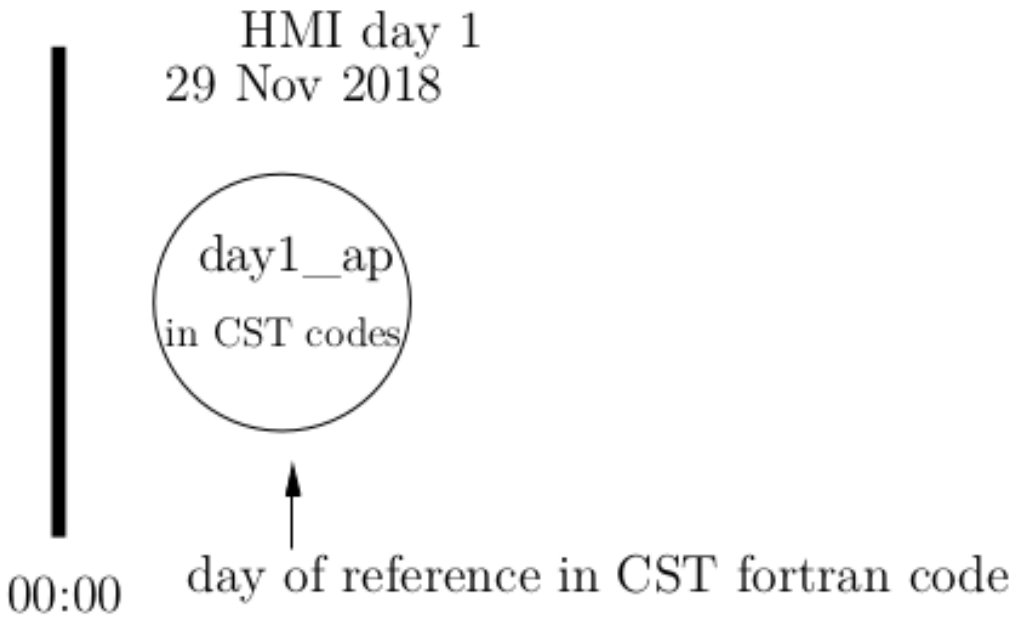
4. Training exercise : running CST codes for 30 mins of SDO/HMI observation (data) - STEP 3 (IDL)

- CPU time : fast $\sim$ 2 min
- The output files in treated_day1_ap directory are :
  - vit_cent_18coeurs.dat
  - Vr_Vtheta_Vphi_derot_day1_ap.dat
  - Vx_h_moy30mn_18coeurs_cor_0001.fits
  - Vx_moy30mn_derot_18coeurs_0001.fits
  - vx_vy_h_derot_moy30mn_18coeurs.sav
  - Vy_h_moy30mn_18coeurs_cor_0001.fits
  - Vy_moy30mn_derot_18coeurs_0001.fits
- treated_day1_ap directory : 19 G

5. How many observation days to run ?

- CST codes can be run up to 6 days and the reference day is the 4th day
- The number of days is very important for the preparation of the directories which will contain the HMI data and the results of CST codes
- We will see different cases (1 day, 2 days, ..., 6 days) and directories to create before running different parts of CST codes

5. Running CST codes for 1 observation day - STEP 1



- Create day1_apres directory and put HMI data (intensity and Doppler) inside
  → 1920 files : 63 G
- Create treated_day1_ap result directory
- cd CST_1_TO_6_DAYS/codes_CST_IDL
- Reduction IDL file :
  reduction_doppler_intensity_all_days_apres.pro

5. Running CST codes for 1 observation day - STEP 1

- Adapt the 4 following parameters :
  - path : path to HMI data (intensity and Doppler) directory (day1_apres)
  - path_out : path to the result directory (treated_day1_ap)
  - rota=0 (for standard rotation, quiet Sun) or rota=1 (rotation is measured directly on the data)
  - spawn,'rm /tmp/HMI*' (for IAS cluster)
- ssh -X cluster-r730-2 (connection to a node of the cluster)
  cd codes_CST_IDL
  ssw hmi
  .r reduction_doppler_intensity_all_days_apres.pro
  reduction_doppler_intensity_all_days_apres,1,rota
- The result directory (treated_day1_ap) contains :
  - ..., day1_apres_int_derot_0001.fits $\rightarrow$ 1909.fits, ...
- CPU time : $\sim$ 8h - 10h

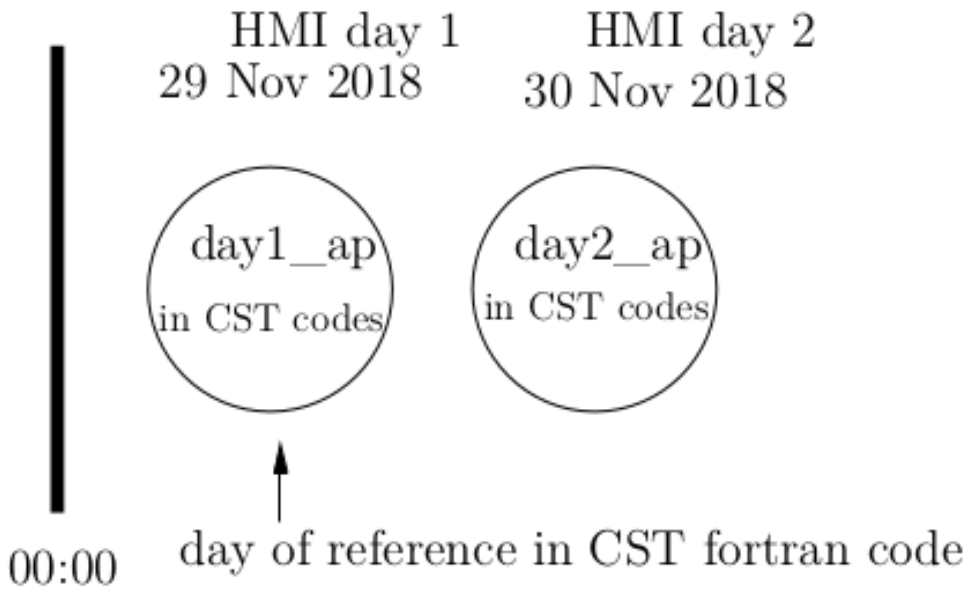
5. Running CST codes for 1 observation day - STEP 2

- If there are not 1920 files (corresponding to 1 day), it is necessary to complete from the last (copy the last file)
- ssh -X cluster-head
- cd codes_CST_Fortran
- Files to modify are :
  - param_seq_29nov2018_EOS_30mn : prefix (path to day1_apres_int_derot_ files), input_file, arcsec, pixel
  - script_29nov2018.sh : adapt 4 times the name of param_seq_29nov2018_EOS_30mn
- sbatch script_29nov2018.sh (several times if necessary - check slurm-XXX.out file)
- The results are in JOB_XXX/results directory
- Copy all ux and uy files from JOB_XXX/results directory to treated_day1_ap directory
- CPU time : $\sim$ 20h - 24h

5. Running CST codes for 1 observation day - STEP 3

- Adapt the following parameters :
  - transform_uxuy_vxvy_all_dates_step_30min.pro : path, path_out (for day=1)
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro : path, path_out, path_vit, spawn,'rm /tmp/HMI*' (for day=1)
  - transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro : path1, path, path_out (for day=1)
- ssh -X cluster-r730-2 (connection to a node cluster)
  cd codes_CST_IDL
  ssw hmi
  .r transform_uxuy_vxvy_all_dates_step_30min.pro
  .r correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro
  .r transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro
- CPU time : fast $\sim$ 2 min - 5 min
- treated_day1_ap directory : 854 G
- JOB_XXX directory : 11 G (in codes_CST_Fortran directory)

5. Running CST codes for 2 observation days



- Create day1_apres directory and put HMI data (intensity and Doppler) on November, 29, 2018 inside : 1920 files (63G)
- Create day2_apres directory and put HMI data (intensity and Doppler) on November, 30, 2018 inside : 1920 files (63G)
- Create result directory : treated_day1_ap (29 Nov 2018)
- Create result directory : treated_day2_ap (30 Nov 2018)

5. Running CST codes for 2 observation days - STEP 1

- cd CST_1_TO_6_DAYS/codes_CST_IDL
- IDL file : reduction_doppler_intensity_all_days_apres.pro
- Adapt the 4 following parameters :
  - path : path to HMI data (intensity and Doppler) directory (day1_apres)
  - path_out : path to the result directory (treated_day1_ap)
  - rota=0 (for standard rotation, quiet Sun) or rota=1 (rotation is measured directly on the data)
  - spawn,'rm /tmp/HMI*' (for IAS cluster)
- ssh -X cluster-r730-2 (connection to a node of the cluster)
  cd codes_CST_IDL
  ssw hmi
  .r reduction_doppler_intensity_all_days_apres.pro
  reduction_doppler_intensity_all_days_apres,2,rota
- Result directories (treated_day1_ap and treated_day2_ap) contain respectively : ..., day1_apres_int_derot_0001.fits → 1909.fits, ...,day2_apres_int_derot_0001.fits → 1909.fits,...

5. Running CST codes for 2 observation days - STEP 2

- If there are not 1920 files (corresponding to 1 day), it is necessary to complete from the last (copy the last file)
- ssh -X cluster-head
- cd codes_CST_Fortran
- Files to create and to modify are :
  - param_seq_29nov2018_EOS_30mn : prefix (path to day1_apres_int_derot_ files), input_file, arcsec, pixel
  - param_seq_30nov2018_EOS_30mn : prefix (path to day2_apres_int_derot_ files), input_file, arcsec, pixel
  - script_29nov2018.sh : adapt 4 times the name of param_seq_29nov2018_EOS_30mn
  - script_30nov2018.sh : adapt 4 times the name of param_seq_30nov2018_EOS_30mn

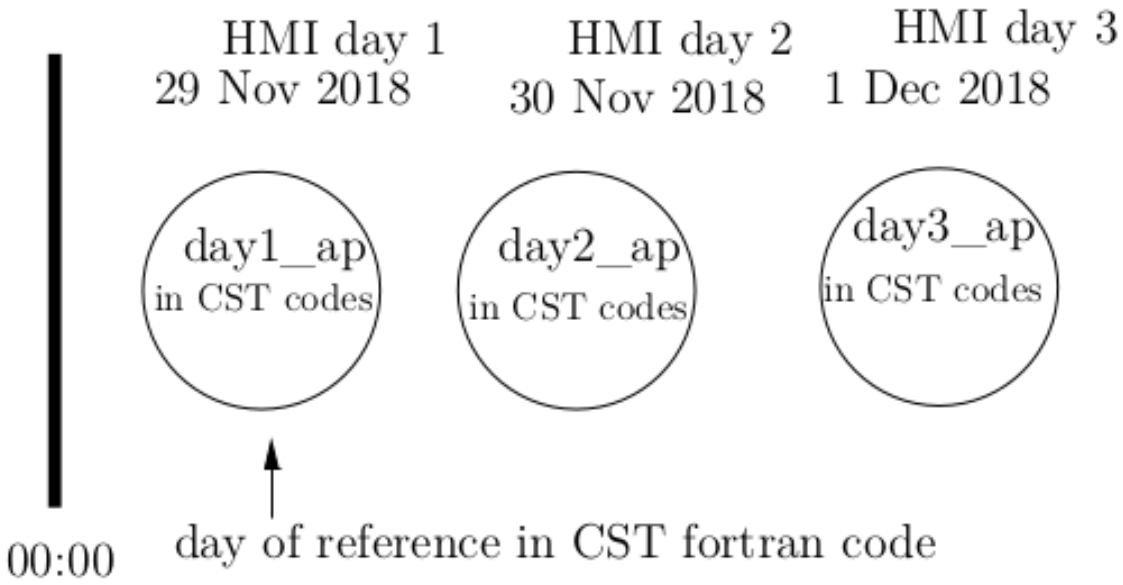
5. Running CST codes for 2 observation days - STEP 2

- FORTRAN jobs can be run in parallel
- sbatch script_29nov2018.sh (several times if necessary - check slurm-XXX.out file)
- sbatch script_30nov2018.sh (several times if necessary - check slurm-ZZZ.out file)
- The results are in JOB_XXX/results directory for the 1st day
- The results are in JOB_ZZZ/results directory for the 2nd day
- Copy all ux and uy files from JOB_XXX/results directory to treated_day1_ap directory
- Copy all ux and uy files from JOB_ZZZ/results directory to treated_day2_ap directory

5. Running CST codes for 2 observation days - STEP 3

- Adapt the following parameters in :
  - transform_uxuy_vxvy_all_dates_step_30min.pro : path, path_out (for day=1 and day=2)
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro : path, path_out, path_vit, spawn,'rm /tmp/HMI*' (for day=1 and day=2)
  - transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro : path1, path, path_out (for day=1 and day=2)
- ssh -X cluster-r730-2 (connection to a node cluster)
  cd codes_CST_IDL
  ssw hmi
  .r transform_uxuy_vxvy_all_dates_step_30min.pro
  .r correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro
  .r transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro
- Results are in treated_day1_ap and treated_day2_ap directories

5. Running CST codes for 3 observation days



- Create day1_apres, day2_apres, day3_apres directories and put respectively HMI data inside (Nov 29, Nov 30, Dec 1, 2018)
- Create result directories : treated_day1_ap (Nov 29), treated_day2_ap (Nov 30), treated_day3_ap (Dec 1)

5. Running CST codes for 3 observation days - STEP 1

- cd CST_1_TO_6_DAYS/codes_CST_IDL
- Adapt paths and parameters in :
  reduction_doppler_intensity_all_days_apres.pro
- ssh -X cluster-r730-2 (connection to a node of the cluster)
  cd codes_CST_IDL
  ssw hmi
  .r reduction_doppler_intensity_all_days_apres.pro
  reduction_doppler_intensity_all_days_apres,3,rota
- Result directories (treated_day1_ap, treated_day2_ap, treated_day3_ap) contain respectively : ...,
  day1_apres_int_derot_0001.fits → 1909.fits, ...,
  day2_apres_int_derot_0001.fits → 1909.fits, ...,
  day3_apres_int_derot_0001.fits → 1909.fits, ...

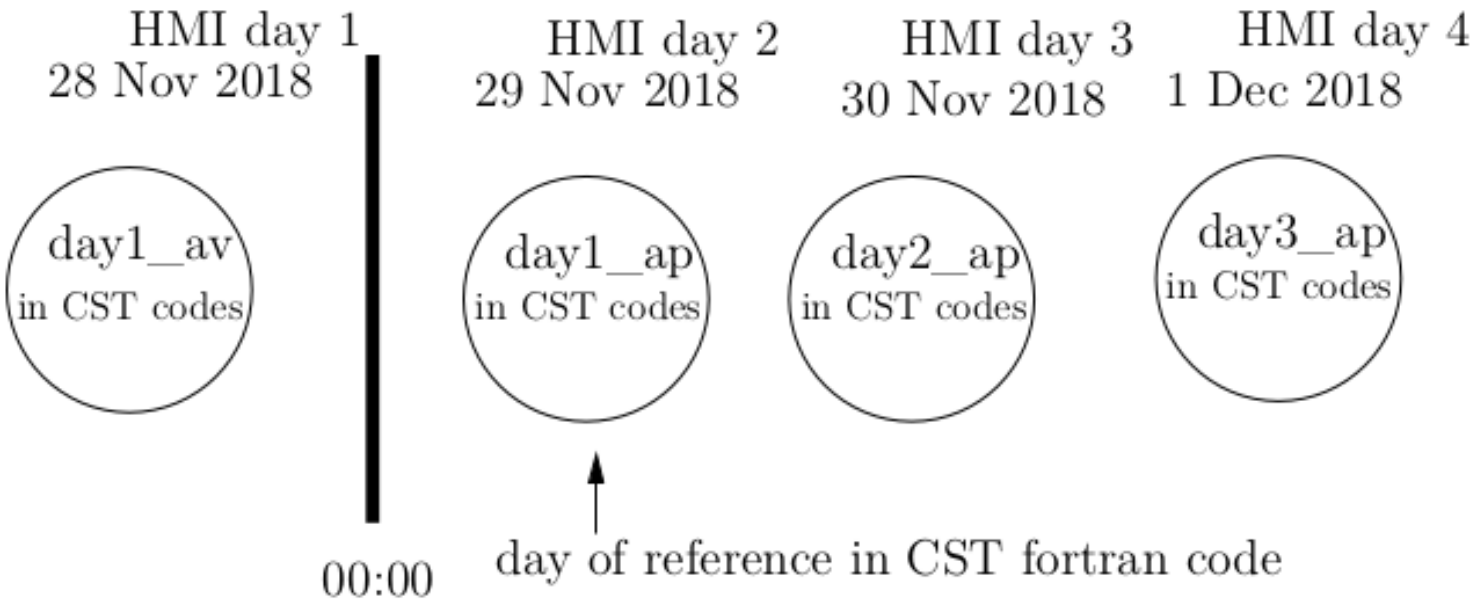
5. Running CST codes for 3 observation days - STEP 2

- If there are not 1920 files for each day, complete by copying
- ssh -X cluster-head
- cd codes_CST_Fortran
- Files to create and to modify are :
  - param_seq_29nov2018_EOS_30mn
  - param_seq_30nov2018_EOS_30mn
  - param_seq_1dec2018_EOS_30mn
  - script_29nov2018.sh
  - script_30nov2018.sh
  - script_1dec2018.sh
- Run in parallel theses 3 jobs (several times if necessary) :
  sbatch script_29nov2018.sh → JOB_AAA
  sbatch script_30nov2018.sh → JOB_BBB
  sbatch script_1dec2018.sh → JOB_CCC
- Copy all ux and uy files from each JOB_XXX/results folder to treated_day1_ap, treated_day2_ap and treated_day3_ap folders respectively

5. Running CST codes for 3 observation days - STEP 3

- Adapt paths in IDL files for day=1, day=2, day=3 :
  - transform_uxuy_vxvy_all_dates_step_30min.pro
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro
  - transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro
- ssh -X cluster-r730-2 (connection to a node cluster)
  cd codes_CST_IDL
  ssw hmi
  .r transform_uxuy_vxvy_all_dates_step_30min.pro
  .r correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro
  .r transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro
- Results are in treated_day1_ap, treated_day2_ap and treated_day3_ap directories

5. Running CST codes for 4 observation days



- Create day1_avant, day1_apres, day2_apres, day3_apres directories and put respectively HMI data inside (Nov 28, Nov 29, Nov 30, Dec 1, 2018)
- Create result directories : treated_day1_av (Nov 28), treated_day1_ap (Nov 29), treated_day2_ap (Nov 30), treated_day3_ap (Dec 1)

5. Running CST codes for 4 observation days - STEP 1

- cd CST_1_TO_6_DAYS/codes_CST_IDL
- Adapt paths/parameters in :
  reduction_doppler_intensity_all_days_apres.pro
  reduction_doppler_intensity_all_days_avant.pro
- ssh -X cluster-r730-2 (connection to a node of the cluster)
  cd codes_CST_IDL
  ssw hmi (Respect the order of runs)
  .r reduction_doppler_intensity_all_days_apres.pro
  reduction_doppler_intensity_all_days_apres,3,rota
  .r reduction_doppler_intensity_all_days_avant.pro
  reduction_doppler_intensity_all_days_avant,1,rota
- Result directories (treated_day1_av, treated_day1_ap, treated_day2_ap, treated_day3_ap) contain respectively : ...,
  day1_avant_int_derot_0001.fits → 1909.fits, ...,
  day1_apres_int_derot_0001.fits → 1909.fits, ...,
  day2_apres_int_derot_0001.fits → 1909.fits, ...,
  day3_apres_int_derot_0001.fits → 1909.fits, ...

5. Running CST codes for 4 observation days - STEP 2

- If there are not 1920 files for each day, complete by copying
- ssh -X cluster-head
- cd codes_CST_Fortran
- Files to create and to modify are :
  - param_seq_28nov2018_EOS_30mn
  - param_seq_29nov2018_EOS_30mn
  - param_seq_30nov2018_EOS_30mn
  - param_seq_1dec2018_EOS_30mn
  - script_28nov2018.sh
  - script_29nov2018.sh
  - script_30nov2018.sh
  - script_1dec2018.sh

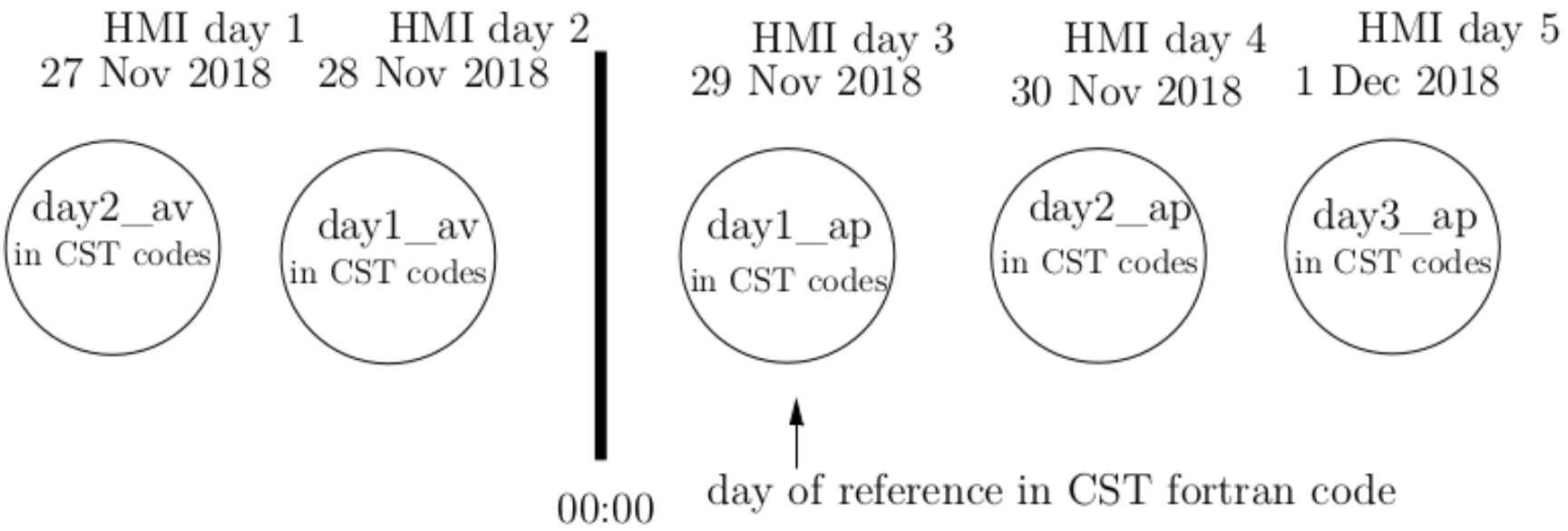
5. Running CST codes for 4 observation days - STEP 2

- Run in parallel theses 4 jobs (several times if necessary) :
  sbatch script_28nov2018.sh → JOB_DDD
  sbatch script_29nov2018.sh → JOB_AAA
  sbatch script_30nov2018.sh → JOB_BBB
  sbatch script_1dec2018.sh → JOB_CCC
- Copy all ux and uy files from each JOB_XXX/results folder to treated_day1_av, treated_day1_ap, treated_day2_ap and treated_day3_ap folders respectively

5. Running CST codes for 4 observation days - STEP 3

- Adapt paths in IDL files for day=1, day=2, day=3, day=4 :
  - transform_uxuy_vxvy_all_dates_step_30min.pro
  - transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro
- Adapt paths in IDL file for day=1, day=2, day=3 :
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro
- Adapt paths in IDL file for day=1 :
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_avant.pro
- ssh -X cluster-r730-2 (connection to a node cluster)
  cd codes_CST_IDL
  ssw hmi (Respect the order of runs)
  .r transform_uxuy_vxvy_all_dates_step_30min.pro
  .r correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro
  .r correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_avant.pro
  .r transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro
- Results : treated_day1_av, treated_day1_ap, treated_day2_ap and treated_day3_ap directories

5. Running CST codes for 5 observation days



- Create day2_avant, day1_avant, day1_apres, day2_apres, day3_apres directories and put respectively HMI data inside (Nov 27, Nov 28, Nov 29, Nov 30, Dec 1, 2018)
- Create result directories treated_day2_av (Nov 27), treated_day1_av (Nov 28), treated_day1_ap (Nov 29), treated_day2_ap (Nov 30), treated_day3_ap (Dec 1)

5. Running CST codes for 5 observation days - STEP 1

- cd CST_1_TO_6_DAYS/codes_CST_IDL
- Adapt paths/parameters in :
  reduction_doppler_intensity_all_days_apres.pro
  reduction_doppler_intensity_all_days_avant.pro
- ssh -X cluster-r730-2 (connection to a node of the cluster)
  cd codes_CST_IDL
  ssw hmi (Respect the order of runs)
  .r reduction_doppler_intensity_all_days_apres.pro
  reduction_doppler_intensity_all_days_apres,3,rota
  .r reduction_doppler_intensity_all_days_avant.pro
  reduction_doppler_intensity_all_days_avant,2,rota
- Results respectively in each treated_day_av/ap directories :
  day2_avant_int_derot_0001.fits → 1909.fits, ...,
  day1_avant_int_derot_0001.fits → 1909.fits, ...,
  day1_apres_int_derot_0001.fits → 1909.fits, ..., day2_apres_, ...,
  day3_apres_int_derot_0001.fits → 1909.fits, ...

5. Running CST codes for 5 observation days - STEP 2

- If there are not 1920 files for each day, complete by copying
- ssh -X cluster-head
- cd codes_CST_Fortran
- Files to create and to modify are :
  - param_seq_27nov2018_EOS_30mn
  - param_seq_28nov2018_EOS_30mn
  - param_seq_29nov2018_EOS_30mn
  - param_seq_30nov2018_EOS_30mn
  - param_seq_1dec2018_EOS_30mn
  - script_27nov2018.sh
  - script_28nov2018.sh
  - script_29nov2018.sh
  - script_30nov2018.sh
  - script_1dec2018.sh

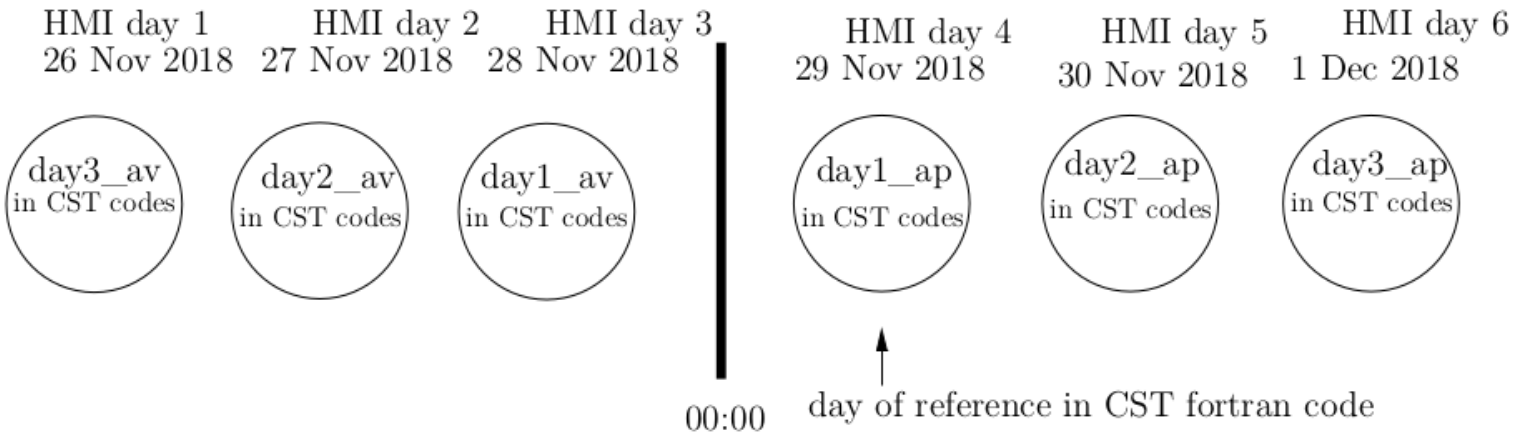
5. Running CST codes for 5 observation days - STEP 2

- Run in parallel theses 5 jobs (several times if necessary) :
  sbatch script_27nov2018.sh → JOB_EEE
  sbatch script_28nov2018.sh → JOB_DDD
  sbatch script_29nov2018.sh → JOB_AAA
  sbatch script_30nov2018.sh → JOB_BBB
  sbatch script_1dec2018.sh → JOB_CCC
- Copy all ux and uy files from each JOB_XXX/results folder to treated_day2_av, treated_day1_av, treated_day1_ap, treated_day2_ap and treated_day3_ap folders respectively

5. Running CST codes for 5 observation days - STEP 3

- Adapt paths in IDL files for day=1, day=2, day=3, day=4, day=5 :
  - transform_uxuy_vxvy_all_dates_step_30min.pro
  - transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro
- Adapt paths in IDL file for day=1, day=2, day=3 :
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro
- Adapt paths in IDL file for day=1, day=2 :
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_avant.pro
- ssh -X cluster-r730-2 (connection to a node cluster)
  cd codes_CST_IDL
  ssw hmi (Respect the order of runs)
  .r transform_uxuy_vxvy_all_dates_step_30min.pro
  .r correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro
  .r correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_avant.pro
  .r transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro
- Results : treated_day2_av, treated_day1_av, treated_day1_ap, treated_day2_ap, treated_day3_ap

5. Running CST codes for 6 observation days



- Create day3_avant, day2_avant, day1_avant, day1_apres, day2_apres, day3_apres directories and put respectively HMI data inside (Nov 26, Nov 27, Nov 28, Nov 29, Nov 30, Dec 1, 2018)
- Create result directories : treated_day3_av (Nov 26), treated_day2_av (Nov 27), treated_day1_av (Nov 28), treated_day1_ap (Nov 29), treated_day2_ap (Nov 30), treated_day3_ap (Dec 1)

5. Running CST codes for 6 observation days - STEP 1

- cd CST_1_TO_6_DAYS/codes_CST_IDL
- Adapt paths/parameters, for day=1 to day=3, in :
  reduction_doppler_intensity_all_days_apres.pro
  reduction_doppler_intensity_all_days_avant.pro
- ssh -X cluster-r730-2 (connection to a node of the cluster)
  cd codes_CST_IDL
  ssw hmi (Respect the order of runs)
  .r reduction_doppler_intensity_all_days_apres.pro
  reduction_doppler_intensity_all_days_apres,3,rota
  .r reduction_doppler_intensity_all_days_avant.pro
  reduction_doppler_intensity_all_days_avant,3,rota
- Results are respectively in each treated_dayN_av/ap directory :
  day1_avant_int_derot_0001.fits → 1909.fits, ..., day2_avant_, ...,
  day3_avant_int_derot_0001.fits → 1909.fits, ...
  day1_apres_int_derot_0001.fits → 1909.fits, ..., day2_apres_, ...,
  day3_apres_int_derot_0001.fits → 1909.fits, ...

5. Running CST codes for 6 observation days - STEP 2

- If there are not 1920 files for each day, complete by copying
- ssh -X cluster-head
- cd codes_CST_Fortran
- Files to create and to modify are :
  - param_seq_26nov2018_EOS_30mn
  - param_seq_27nov2018_EOS_30mn
  - param_seq_28nov2018_EOS_30mn
  - param_seq_29nov2018_EOS_30mn
  - param_seq_30nov2018_EOS_30mn
  - param_seq_1dec2018_EOS_30mn
  - script_26nov2018.sh
  - script_27nov2018.sh
  - script_28nov2018.sh
  - script_29nov2018.sh
  - script_30nov2018.sh
  - script_1dec2018.sh

5. Running CST codes for 6 observation days - STEP 2

- Run in parallel theses 6 jobs (several times if necessary) :
  sbatch script_26nov2018.sh → JOB_FFF
  sbatch script_27nov2018.sh → JOB_EEE
  sbatch script_28nov2018.sh → JOB_DDD
  sbatch script_29nov2018.sh → JOB_AAA
  sbatch script_30nov2018.sh → JOB_BBB
  sbatch script_1dec2018.sh → JOB_CCC
- Copy all ux and uy files from each JOB_XXX/results folder to treated_day3_av, treated_day2_av, treated_day1_av, treated_day1_ap, treated_day2_ap and treated_day3_ap folders respectively

5. Running CST codes for 6 observation days - STEP 3

- Adapt paths in IDL files for day=1 to day=6 :
  - transform_uxuy_vxvy_all_dates_step_30min.pro
  - transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro
- Adapt paths in IDL file for day=1, day=2, day=3 :
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro
- Adapt paths in IDL file for day=1, day=2, day=3 :
  - correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_avant.pro
- ssh -X cluster-r730-2 (connection to a node cluster)
  cd codes_CST_IDL
  ssw hmi (Respect the order of runs)
  .r transform_uxuy_vxvy_all_dates_step_30min.pro
  .r correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_apres.pro
  .r correct_mvtSDO_sur_Vx_et_Vy_precis_all_dates_avant.pro
  .r transfor_Vx_Vy_Vdop_vthe_vphi_vr_all_dates_step_30min.pro
- Results are in treated_day3_av, treated_day2_av, treated_day1_av, treated_day1_ap, treated_day2_ap, treated_day3_ap directories

6. Frequent troubles encountered

- IDL parts : IDL codes of STEP 1 and 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
- The first time you run CST Fortran codes, you need to make sbatch several times because of a link editing problem in the Makefile.

6. 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 precendent 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_DDMMMYYY.sh", is switched into NINDEX=20 (if ux_0020 is not computed) and we also modify the line :
  while [$NINDEX -le 47] to while [$NINDEX -le 20] .

7. Work in progess

- CST_V1.1 online soon : in STEP 3, there is only 1 IDL file instead of 3
  → step3_CST_IDL_apres.pro
  → step3_CST_IDL_avant.pro (depending of the treated case)
- Add to user guide a section on description of the codes
- IDL parts will be written in Python
- Makefile in FORTRAN part will be improved to avoid several sbatch for the compilation
- ...
- Next workshop in 2021 !

8. Contacts (SAV)

Don't hesitate to contact us if you have any problem

Thierry Roudier : thierry.roudier@irap.omp.eu

Martine Chane-Yook : martine.chane-yook@ias.u-psud.fr


MERCI POUR VOTRE
ATTENTION