# Processing of VDFs observed by the SWA-PAS instrument

Tereza Ďurovcová, Lubomír Přech

June 10th, 2021

## Contents



# 1 Introduction

This document describes the structure and dataset produced by two alternative routines
A: `solo_l2proc_batch_pasep_noimf` and B: `solo_l2proc_batch_pasep`. Both these procedures use nonlinear fitting to estimate the basic parameters (density, velocity and temperature) of the proton and alpha solar wind components from 3D velocity distribution functions (VDFs) measured by the PAS instrument (L2 data) on-board the Solar Orbiter. The first of them (procedure A) does not use magnetic field measurements and therefore provides the isotropic temperature only. The second (procedure B) uses magnetic field measurements (MAG) to determine the temperature anizotropy. The inputs of both routines are the same: `sTime` and `eTime` are the Modified Julian start and end times of the required L2 data processing, `dt` is the final data resolution in days (0 means no averaging), and `magframe` specifies the coordinate system for magnetic field input data ('srf' or 'rtn', procedure B only). Routines produce results in the .sav files for each day in the entered time interval.

The following description is related to the V1.01 of the procedures as implemented on the *rosina1.irap.omp.eu* server.

# 2 Description of the fitting method

## 2.1 Data preparation and restriction of the angular domain

- Read full energy, elevation, and azimuth tables
  These tables are usually read only once from the first PAS L2 VDF file. They are updated if

`Calibration_version` attribute changes in following L2 files.

- Read L2 VDFs and L2 energy fluxes and IMF normal-mode data (procedure B only)
- MAG normal-mode RTN/SRF data are linearly interpolated to the PAS measurement times. Time differences between two interpolating MAG points are not checked in the current version and possible long MAG data gaps may impact the results.
- Transformation of the MAG data into the PAS frame (procedure B only) given by:

$$\mathbf{A_{PAS}} = m2pas\mathbf{A},$$

where $m2pas$ is transformation matrix from the RTN/SRF frame to the PAS frame.

RTN: $m2pas = \texttt{PAS\_to\_RTN}^{-1}$, `PAS_to_RTN` is provided in the L2 VDF data files,

SRF: $m2pas = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

- If the input value `dt` is greater than 0, the averaged VDF, energy flux, and magnetic field vector components are computed for each `dt` interval.
- If the input value `dt` is 0, the correction of elevation (provided in the L2 VDF data files) is applied
- Velocity components in the PAS frame are computed as follows:

$$\mathbf{V_{PAS}} = [\,-v_r\cos(\alpha)\cos(\varepsilon), -v_r\sin(\alpha)\cos(\varepsilon), v_r\sin(\varepsilon)\,],$$

where $\alpha$ is azimuth and $\varepsilon$ is elevation.

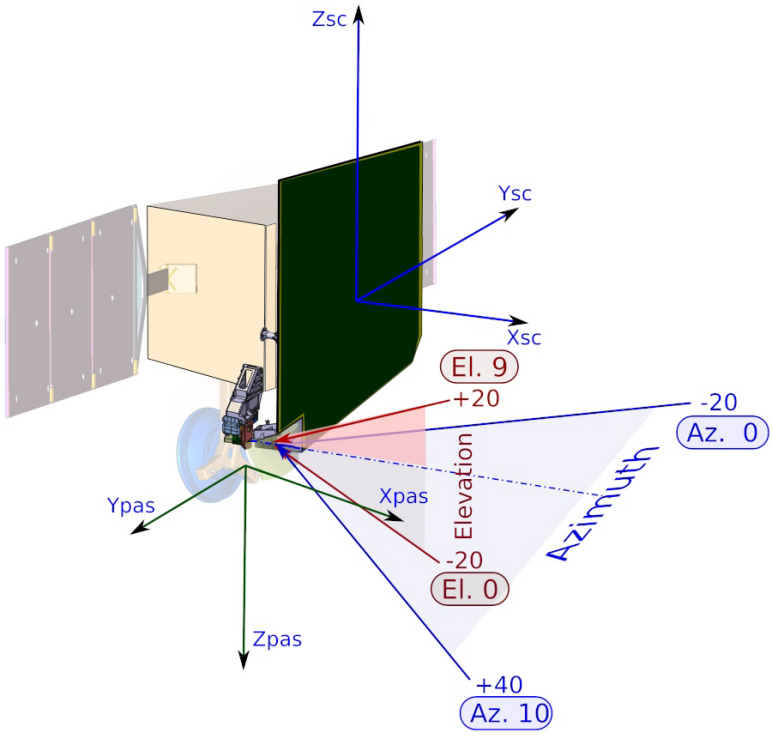


Figure 1: Elevation and azimuth bins in the Solar Orbiter frame.

- The angular velocity space is restricted to an area with significant count levels (count level greater than 1) in order to reduce the impact of noise in the outlying channels

## 2.2 Estimation of the helium-dominant part of the VDF

Separation of proton and alpha parts of the measured VDF is identical for both procedures, A and B. The following procedure is used to find an estimate of the boundary between them:

1. Find the global maximum of the energy flux $F(E_{max})$ and corresponding energy $E_{max}$
2. Find all local minima in the energy range from $E_{max}$ to $2E_{max}$
3. As the boundary between the proton and alpha parts, we choose the minimum that satisfies the following two conditions: it has the lowest energy flux of all local minima and it is lower than 0.2 $F(E_{max})$

When no boundary is found, momentum calculation and fitting are not performed.

## 2.3 Moment calculation

The moments of the proton-dominant and helium-dominant parts are calculated for both procedures as follows:

$$N = \sum_{v_r,\varepsilon,\alpha} f(\mathbf{V_{PAS}})\mathrm{d}\Omega = \sum_{v_r,\varepsilon,\alpha} f(\mathbf{V_{PAS}})\, v_r^2 \cos(\varepsilon)\Delta v_r \Delta\varepsilon\Delta\alpha$$

$$\mathbf{U_{PAS}} = \mathbf{V_{max,PAS}} + \frac{1}{N}\sum_{v_r,\varepsilon,\alpha} (\mathbf{V_{PAS}} - \mathbf{V_{max,PAS}})\, f(\mathbf{V_{PAS}})\mathrm{d}\Omega$$

$$w_{tot}^2 = \frac{1}{N}\sum_{v_r,\varepsilon,\alpha} (\mathbf{V_{pas}} - \mathbf{U_{PAS}})^2\, f(\mathbf{V_{PAS}})\mathrm{d}\Omega$$

where $N$ is the number density, $f(\mathbf{V_{PAS}})$ is the velocity distribution function, $\mathbf{U_{PAS}}$ is the bulk speed in the PAS frame, $\mathbf{V_{max,PAS}}$ is the speed corresponding to the maximum of the measured VDF and $w_{tot}$ is the total thermal speed.

## 2.4 Fitting using the Levenberg-Marquardt algorithm

### 2.4.1 Procedure A

All components are approximated by a Maxwellian distribution, which is defined for a species, j as:

$$f_j = \frac{N_j}{(2\pi)^{\frac{3}{2}} w_{iso\,j}^3} \exp\left(-\frac{(v_x-u_{xj})^2}{2w_{iso\,j}^2} - \frac{(v_y-u_{yj})^2}{2w_{iso\,j}^2} - \frac{(v_z-u_{zj})^2}{2w_{iso\,j}^2}\right),$$

where $v_i$ ($i = \{x, y, z\}$) are velocities in the PAS frame, $N_j$ is the number density, $u_{ij}$ are components of the bulk velocity, and $w_{iso}$ is isotropic thermal speed. Free parameters of the proton and alpha fits are: $N_j$, $\mathbf{u_{ij}}$, $w_{iso\,j}$. The proton core fit (obtained first) is subtracted from the measured VDF. Then, the VDF alpha-particle dominant part is fitted.

### 2.4.2 Procedure B

Before fitting, the VDF is rotated into the IMF aligned frame. This rotation is determined by the rotation axis $\mathbf{k} = \dfrac{e_{xPAS} \times e_b}{|e_{xPAS} \times e_b|}$, and angle $\measuredangle(e_b, e_{xPAS})$, where $e_{xPAS} = [1, 0, 0]$ is an unit vector in the direction of the X axis in the PAS frame, $e_b = \dfrac{\mathbf{B_{pas}}}{|\mathbf{B}|}$ is an unit vector in the direction of IMF.

All components are approximated by a bi-Maxwellian distribution, which is defined for a species, j as:

$$f_j = \frac{N_j}{(2\pi)^{\frac{3}{2}} w_{||j} w_{\perp j}^2} \exp\left(-\frac{(v_x-u_{xj})^2}{2w_{||j}^2} - \frac{(v_y-u_{yj})^2}{2w_{\perp j}^2} - \frac{(v_z-u_{zj})^2}{2w_{\perp j}^2}\right),$$

where $v_i$ ($i = \{x, y, z\}$) are velocities in the IMF aligned frame, $N_j$ is the number density, $u_{ij}$ are components of the bulk velocity, and $w_{||}, w_{\perp}$ are parallel and perpendicular thermal speeds. Free parameters of the proton and alpha fits are: $N_j$, $\mathbf{u_{ij}}$ (for alpha we fit only the $u_x$ component that is along the IMF vector), $w_j$, $WA$ (thermal speed and anizotropy factor defining $w_{||j} = w_j(1 + WA)$, $w_{\perp j} = w_j(1 - WA)$). The proton core fit (obtained first) is subtracted from the measured VDF. Then, the VDF alpha-particle dominant part is fitted.

## 2.5 Suspicious fits removal

### 2.5.1 Procedure A

From the resulting dataset the alpha particle fits are removed (replaced by $NaN$ values) when any of the following tentative conditions is met:

- The relative alpha-particle abundance

$$A_{He}[\%] = \frac{N_a}{N_p} \times 100$$

  is not in the range $\langle 0.5, 10 \rangle$).

- The proton-to-alpha speed ratio

$$\frac{|u_p|}{|u_a|}$$

  is not in the range $\langle 0.9, 1.2 \rangle$.

- The alpha-to-proton temperature ratio

$$4 \times \frac{w_{iso\,a}^2}{w_{iso\,p}^2}$$

  is not in the range $\langle 0.5, 10 \rangle$.

### 2.5.2 Procedure B

From the resulting dataset the alpha particle fits are removed (replaced by $NaN$ values) when any of the following tentative conditions is met:

- The relative alpha-particle abundance

$$A_{He}[\%] = \frac{N_a}{N_p} \times 100$$

  is not in the range $\langle 0.5, 10 \rangle$).

- The proton-to-alpha speed ratio

$$\frac{|u_p|}{|u_a|}$$

  is not in the range $\langle 0.9, 1.2 \rangle$.

- The alpha-to-proton temperature ratio

$$4 \times \frac{\left(w_{||a}^2 + 2w_{\perp a}^2\right)}{\left(w_{||p}^2 + 2w_{\perp p}^2\right)}$$

  is not in the range $\langle 0.5, 10 \rangle$.

# 3 Description of the produced dataset

Table 1 summarizes the elements of the resulting data structure. We add four data flags:

- The first flag indicates the coefficient of determination of the fit of a VDF proton part and it is defined as:

$$R^2 = 1 - \frac{\sum_i (f_i - f_{i,fit})^2}{\sum_i (f_i - \langle f_i \rangle)^2},$$

  where $f_i$, $f_{i,fit}$ are the values of the measured and fitted VDFs at given $v_x$, $v_y$, $v_z$ and the angle brackets indicate the mean value. This flag therefore characterizes the quality of the proton fit.

- The second flag gives the same, but for the alpha part of the measured VDF.

- The third flag indicates the ratio between the maximum of the fitted alpha distribution and the maximum $f_i$ at the proton-alpha boundary. This flag is used to estimate the overlap of the proton and alpha parts of the measured VDF. Currently (V1.01), if the maximum $f_i$ is 0 at the proton-alpha boundary, the flag is set to an arbitrary value 1000.

- The fourth flag gives the ratio between the proton density obtained from the fitting and calculated as a moment. It can thus serve as a rough estimate of the abundance of non-thermal parts of the proton VDF.

| Parameter | Structure tag name | Units |
|---|---|---|
| Modified Julian time | MJD | |
| Proton density | $Np$ | [cm$^{-3}$] |
| Proton velocity | $vVp$ | [km/s] |
| Proton isotropic thermal speed | $Wp\_iso$ | [km/s] |
| Proton parallel thermal speed | $Wp\_par$ | [km/s] |
| Proton perpendicular thermal speed | $Wp\_perp$ | [km/s] |
| Alpha density | $Na$ | [cm$^{-3}$] |
| Alpha velocity | $vVa$ | [km/s] |
| Alpha isotropic thermal speed | $Wa\_iso$ | [km/s] |
| Alpha parallel thermal speed | $Wa\_par$ | [km/s] |
| Alpha perpendicular thermal speed | $Wa\_perp$ | [km/s] |
| Magnetic field in RTN frame | $vB\_rtn$ | [nT] |
| Magnetic field in SRF frame | $vB\_srf$ | [nT] |
| flag 1 | flag_1 | |
| flag 2 | flag_2 | |
| flag 3 | flag_3 | |
| flag 4 | flag_4 | |

Table 1: Elements of the resulting data structure.

# 4 Implementation of the code - rosina1 server

## 4.1 Folder assignment

IDL code folder: /home/prech/work/IDLCOM/swa-pas_TD_L3
PAS L2 data input folder: /DATA/SOLAR/DATA/
MAG L2 data input folder: /DATA/SOLAR/DATA/MAG/L2/
MAG L2 data (cdag) folder: /DATA/SOLAR/CDAG/MAG/ (not used in the current version)
output folder: /home/prech/work/L3/ (default value)

journal folder: /home/prech/work/L3/ (default value)

### 4.2 Calling sequence

Prior any calls all IDL source routinse have to precompiled:
$ cd /home/prech/work/IDLCOM/swa-pas_TD_L3
$ idl
IDL>@_compile.pro

Procedure A:
SOLO_L2proc_batch_pase_noimf, sTime, eTime, dt, journal=journal, quiet=quiet, outf=outf

; INPUTS
; sTime ... Modified Julian Time for a start time
; eTime ... Modified Julian Time for a end time
; dt ... final data resolution (0 means no averaging), in fraction of day

; KEYWORDS
; journal ... create log file
; quiet ... reduce debug output
; outf ... output folder (data and journal)

; CALL EXAMPLE
; SOLO_L2proc_batch_pasep_noimf, JD(2020,9,6,0,0,0, /m), JD(2020,9,7,0,0,0, /m), 0, /j,
outf='/rosina_b1/solar/data/l3/'
; SOLO_L2proc_batch_pasep_noimf, JD(2020,9,6,0,0,0, /m), JD(2020,9,7,0,0,0, /m), 30d/24/3600, /j

Procedure B:
SOLO_L2proc_batch_pasep, sTime, eTime, dt, magframe, journal=journal, quiet=quiet, outf=outf

; INPUTS
; sTime ... Modified Julian Time for a start time
; eTime ... Modified Julian Time for a end time
; dt ... final data resolution (0 means no averaging), in fraction of day
; magframe ... coordinate system for magnetic field data ( 'srf' or 'rtn' )

; KEYWORDS
; journal ... create log file
; quiet ... reduce debug output
; outf ... output folder (data and journal)

; CALL EXAMPLE
; SOLO_L2proc_batch_pasep, JD(2020,9,6,0,0,0, /m), JD(2020,9,7,0,0,0, /m), 0, 'rtn', /j,
outf='/rosina_b1/solar/data/l3/'
; SOLO_L2proc_batch_pasep, JD(2020,9,6,0,0,0, /m), JD(2020,9,7,0,0,0, /m), 30d/24/3600, 'rtn', /j

### 4.3 Output

Procedure A:

Data file as IDL save file named AYYYY_DOY.sav or AYYYY_DOY_xxxs.sav (when averaging over xxx seconds used).

Internal structure:

data = replicate( {swa_pas_L2fit_noimf, MJD:0d, $
Np:0d, vVp:dblarr(3), Wp_iso:0d, Na:0d, vVa:dblarr(3), Wa_iso:0d, $
flag_1:0d, flag_2:0d, flag_3:0d, flag_4:0d }, num )

Optional journal file logA_YYYY_MM_DD_HHMMSS.txt

Procedure B:

Data file as IDL save file named BYYYY_DOY.sav or AYYYY_DOY_xxxs.sav (when averaging over xxx seconds used).

Internal structure:
data = replicate( {swa_pas_L2fit, MJD:0d, $
Np:0d, vVp:dblarr(3), Wp_par:0d, Wp_perp:0d, Na:0d, vVa:dblarr(3), Wa_par:0d, Wa_perp:0d, $
vB_rtn:dblarr(3), vB_srf:dblarr(3), $
flag_1:0d, flag_2:0d, flag_3:0d, flag_4:0d }, num )

Optional journal file logB_YYYY_MM_DD_HHMMSS.txt

IDL procedures/functions in the /home/prech/work/IDLCOM/swa-pas_TD_L3/Tools/MJD folder can be used for manipulations with the Modified Julian Time MJD, used in Procedures A,B input parameters and output .sav files (e.g., jd.pro, jultime.pro, str2mjd.pro, mjd2str.pro, caltime.pro), including conversion to/from the CDF Epoch format (cdf_time.pro).