# Table of Contents

1 OVERVIEW ................................................................. ........................................... .............................. 1

1.1 STATEMENT OF PURPOSE ................................................................. ........................................... .............................. 1
1.2 PURPOSE OF DOCUMENT ................................................................. ........................................... .............................. 1

2 RELATED DOCUMENTS ................................................................. ........................................... .............................. 1

2.1 PARENT DOCUMENT ................................................................. ........................................... .............................. 1
2.2 APPLICABLE DOCUMENTS ................................................................. ........................................... .............................. 1

3 PRODUCT ARCHITECTURE ................................................................. ........................................... .............................. 2

3.1 SUMMARY OF ARCHITECTURE ................................................................. ........................................... .............................. 2

4 PRODUCT DESIGN ................................................................. ........................................... .............................. 5

4.1 PROCESS CALLING HIERARCHY FOR THE EXECUTIVE SCRIPT ................................................................. ........................................... .............................. 5
4.2 SUBROUTINE CALLING HIERARCHY FOR THE DATA PREPARATION PROCESS ................................................................. ........................................... .............................. 6
4.3 SUBROUTINE CALLING HIERARCHY FOR THE REPORT PROCESS ................................................................. ........................................... .............................. 7
4.4 SUBROUTINE CALLING HIERARCHY FOR THE SOURCE1 PROCESS ................................................................. ........................................... .............................. 8
4.5 SUBROUTINE CALLING HIERARCHY FOR THE SOLWIND PROCESS ................................................................. ........................................... .............................. 9
4.6 SUBROUTINE CALLING HIERARCHY FOR THE EC PLOT PROCESS ................................................................. ........................................... .............................. 10
4.7 SUBROUTINE CALLING HIERARCHY FOR THE EPHEMERIS SUBROUTINE WESHIRE ................................................................. ........................................... .............................. 11
4.8 SUBROUTINE CALLING HIERARCHY FOR THE IMF VECTOR SUBROUTINE SBT ................................................................. ........................................... .............................. 12
4.9 SUBROUTINE CALLING HIERARCHY FOR THE R-T CURVE SUBROUTINE SRT AND THE SOURCE ................................................................. ........................................... .............................. 13
4.10 SUBROUTINE CALLING HIERARCHY FOR THE TRIANGLE AREA SUBROUTINE TRIA2 AND THE COORDINATE TRANFORMATION SUBROUTINE GSEQTOGSM ................................................................. ........................................... .............................. 14
4.11 PROGRAM CONSTANTS (PARAMETERS) ................................................................. ........................................... .............................. 15
4.12 PROGRAM GLOBALS (COMMON BLOCKS) ................................................................. ........................................... .............................. 19

5 OBJECT DESCRIPTIONS ................................................................. ........................................... .............................. 22

5.1 FORTRAN OBJECTS ................................................................. ........................................... .............................. 22
5.1.1 FUNCTION asine x(x) ................................................................. ........................................... .............................. 22
5.1.2 FUNCTION atan2deg(y, x) ................................................................. ........................................... .............................. 22
5.1.3 SUBROUTINE bmaxmin(max, jmx) ................................................................. ........................................... .............................. 23
5.1.4 SUBROUTINE calcv ................................................................. ........................................... .............................. 24
5.1.5 SUBROUTINE carrot(zjd, icrn) ................................................................. ........................................... .............................. 25
5.1.6 FUNCTION cosdeg(x) ................................................................. ........................................... .............................. 25
5.1.7 FUNCTION cotan(x) ................................................................. ........................................... .............................. 26
5.1.8 SUBROUTINE cross(x1, x2, x3) ................................................................. ........................................... .............................. 27
5.1.9 SUBROUTINE ctos(vin, r, t, p) ................................................................. ........................................... .............................. 27
5.1.10 SUBROUTINE ctosv(theta, phi, win, uvs, thetab, phib) ................................................................. ........................................... .............................. 28
5.1.11 SUBROUTINE dot(x1, x2, x3) ................................................................. ........................................... .............................. 28
5.1.12 PROGRAM ecplot ................................................................. ........................................... .............................. 29
5.1.13 SUBROUTINE geitogsm(s, g, s, v, vout) ................................................................. ........................................... .............................. 32
5.1.14 SUBROUTINE geotoge(gst, vin, vout) ................................................................. ........................................... .............................. 33
5.1.15 SUBROUTINE getu (iyr, imo, id, ihr, imin, secs, iutc, jd) ................................................................. ........................................... .............................. 34
5.1.16 SUBROUTINE gseqtoget(s, vin, vout) ................................................................. ........................................... .............................. 35
5.1.17 SUBROUTINE gseqto gsm(s, g, s, v, vout) ................................................................. ........................................... .............................. 36
5.1.18 SUBROUTINE helico (iyr, zlatec, zlon e, zlateq, zlo neq) ................................................................. ........................................... .............................. 36
5.1.19 SUBROUTINE helio(zjd, 10, b0, p, eclon, sun) ................................................................. ........................................... .............................. 37
5.1.20 SUBROUTINE helhec(iyr, zlatec, zlon e, zlateq, zlo neq) ................................................................. ........................................... .............................. 38
5.1.21 FUNCTION iday(iyr, mo, id) ................................................................. ........................................... .............................. 39
5.1.22 SUBROUTINE interp3(x, y, z) ................................................................. ........................................... .............................. 39
6.4.1 Software
1 Overview

1.1 Statement of Purpose
The primary purpose of the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model software is to provide AFWA the capability to quantitatively predict solar wind conditions (speed, density and interplanetary magnetic field) at the Earth based upon observations of the Sun. Specifically, it provides shock propagation and temporal profiles of the solar wind speed, density, and interplanetary magnetic field (IMF) at the L1 Sun-Earth libration point using background solar wind and solar event data. This information can be used to predict the time and severity of terrestrial disturbances following solar events. Additionally, the HAF code produces chronological sequences of ecliptic cross-section plots of the IMF. These plots may be displayed with an existing IDL program (‘ecimf_c.pro’).

The main outputs from the HAF code are a solar wind forecast table for L1, a Shock Arrival Time (SAT) prediction, and IMF plot data. The inputs to the system are the solar source surface synoptic maps and solar event data. The current synoptic maps of the magnetic field and solar wind velocity at the source surface are available from a UAF web site. The AFWA forecaster will provide the solar event data from presently available event reports (solar optical, radio and GOES X-ray data sources).

1.2 Purpose of Document
The purpose of this document is to provide for the customer’s developer and maintainer a description of the application and its software components, describe the architecture of the software and to identify the components that comprise the application.

2 Related Documents

2.1 Parent Document
The parent document is the requirements document from which this software design document is derived.


2.2 Applicable Documents
The following documents are referenced within this software design document or are directly applicable. The scientific papers document the theoretical basis for the HAF software.

- F19628-86-K0030
3 Product architecture

3.1 Summary of Architecture

Figure 1 shows a context diagram of the HAF software. The two primary inputs are the synoptic maps of the solar wind speed and magnetic field and the reports of solar transient events (flares and CME’s). The synoptic maps are derived from observations of the large-scale photospheric magnetic field. They are the inferred configuration of the coronal magnetic field at the source surface, defined as a surface around the sun where the bulk-flow energy becomes dominant over the magnetic field energy. The HAF software uses the maps to establish boundary conditions on the inner boundary of the computational domain. There are also empirical algorithms in the software to represent the time-varying enhancements to the coronal magnetic field on the source surface resulting from solar transients derived from the Event Reports.

The HAF software is capable of providing IMF data in Heliocentric and Geocentric Solar Equatorial coordinates as well as Geocentric Solar Magnetic (GSM) coordinates. It must also convert the ingested solar observation data to the Heliocentric Equatorial (HEQ) coordinates that are used by the model. The ephemeris routine is used to perform these coordinate transformations and to locate the Earth in the various coordinate frames.

Figure 1. Block diagram showing the components of the Hakamada Akasofu-Fry Solar Wind Model (HAF Code).
The primary outputs from the HAF software, the SAT prediction, IMF plots, and IMF forecasts at L1, are all stored to ASCII text output files. Several plot files are normally produced, one for each sample time requested.

The data flow of the HAF software is shown in Figure 2. The components shown in this figure map into the software units delivered. An executive script initiated by the AFWA scheduler manages the environment and the execution of the software according to AFWA requirements.

**Kinematic Solar Wind Model**

**On Demand**

![Diagram](diag.png)

**Figure 2a. Kinematic Solar Wind Model Data Flow (part 1)**

As shown in the data flow diagram, the HAF software consists of two pairs of programs communicating through a set of intermediate files. The first two programs are ingest preparation processes. The first process, REPORT, calculates the Carrington Rotation and translates the observed flare characteristics from observatory reports. The SOURCE1 process extracts the corresponding Carrington source field data. SOLWIND uses the source surface data to simulate the IMF, solar wind velocity, and density. ECPLOT creates ecliptic cross-section 'snap-shot' files, at given time increments, of the IMF pattern due to the inner boundary source. These plots may be displayed with an existing IDL program ('ecimf_c.pro').
Figure 2b. Kinematic Solar Wind Model Data Flow (part 2)

In addition to the processes discussed above, an executive script will retrieve the current synoptic maps from UAF and coordinate the execution of the processes. The running of the script will be scheduled in accordance with AFWA requirements. There is no custom graphical user interface. AFWA may use existing software or commercial packages to display the content of the output products.
4 Product design

The following structure charts depict the organization of the HAF code. The various modules are represented by rectangles. Dependencies between modules as well common (block)data stores, displayed as hexagons, are shown. “Home plates” are off-page connectors to modules that are further decomposed in subsequent figures.

4.1 Process Calling Hierarchy for the Executive Script

![Diagram of HAF Executive Script Architecture]

Figure 3: HAF Executive Script Architecture
4.2 Subroutine Calling Hierarchy for the Data Preparation Process

Figure 4: Data Preparation Process
4.3 Subroutine Calling Hierarchy for the Report Process

![Diagram of subroutine calling hierarchy](image)

Figure 5: Report Process Architecture
4.4 Subroutine Calling Hierarchy for the Source1 Process

Figure 6: Source1 Process Architecture
4.5 Subroutine Calling Hierarchy for the Solwind Process

Figure 7: Solwind Process Architecture
4.6 Subroutine Calling Hierarchy for the Ecplot Process

Figure 8: Ecplot Process Architecture
4.7 Subroutine Calling Hierarchy for the Ephemeris Subroutine Weshere

Figure 9: Weshere Subroutine Architecture
4.8 Subroutine Calling Hierarchy for the IMF Vector Subroutine SBT

Figure 10: SBT Subroutine Architecture
4.9 Subroutine Calling Hierarchy for the R-T Curve Subroutine SRT and the Source Surface Field Subroutine SRCFLD

![Subroutine Calling Hierarchy Diagram]

Figure 11: SRT and SRCFLD Subroutine Architecture
4.10 Subroutine Calling Hierarchy for the Triangle Area Subroutine TRIA2 and the Coordinate Transformation Subroutine GSEQTOGSM

Figure 12: TRIA2 and GSEQTOGSM Subroutines
### 4.11 Program constants (Parameters)

The HAF software has several constants that are defined in and shared through the params.f file. These constants fall into two general categories. One set are the software control parameters that set certain characteristics of the software such as the maximum size of some data arrays and selection of specific modes for the programs to run in. Several of these FORTRAN parameters have a limited set of valid values they may be set to. Typically only one of those values permits the HAF code to function in accordance with the documented requirements. Those values are identified in **bold**.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Used in (routines)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFSUBDR</td>
<td>Subdirectory of the data directory to read and write files. Used only in the</td>
<td>report sbt</td>
</tr>
<tr>
<td></td>
<td>‘SCHED’ and ‘FIXED’ STARTYPE modes.</td>
<td>solwind</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ecplot</td>
</tr>
<tr>
<td>IDIAGFLG</td>
<td>1 = Enable diagnostic messages and output files. 0 = Diagnostics disabled.</td>
<td>sbt sbt</td>
</tr>
<tr>
<td>IMAX1 or IMAX2</td>
<td>Array dimensions of sourceb.par and sourcev.par data files.</td>
<td>calcv blkbfld</td>
</tr>
<tr>
<td>JMAX</td>
<td></td>
<td>ecplot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ldbfld</td>
</tr>
<tr>
<td></td>
<td></td>
<td>prtparam</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solwind</td>
</tr>
<tr>
<td></td>
<td></td>
<td>source1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>srcfld srcvel</td>
</tr>
<tr>
<td>IMAX1 KMAX</td>
<td>Array dimensions of Wilcox-type source field data files.</td>
<td>calcv blktemp</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ecplot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ldbfld</td>
</tr>
<tr>
<td></td>
<td></td>
<td>params</td>
</tr>
<tr>
<td></td>
<td></td>
<td>prtparam</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solwind</td>
</tr>
<tr>
<td></td>
<td></td>
<td>source1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>srcfld srcvel</td>
</tr>
<tr>
<td>IMAX2 KMAX</td>
<td>Array dimensions of SEC-type source field data files.</td>
<td>blkbfld blktemp</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ecplot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ldbfld</td>
</tr>
<tr>
<td></td>
<td></td>
<td>params</td>
</tr>
<tr>
<td></td>
<td></td>
<td>prtparam</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solwind</td>
</tr>
<tr>
<td></td>
<td></td>
<td>source1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>srcfld srcvel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>srf srf</td>
</tr>
<tr>
<td>Table Title</td>
<td>Description</td>
<td>Related Parameters</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ISRCTYPE</td>
<td>Selects expected format of ingested source surface data.</td>
<td>1 = Wilcox-type data, 2 = SEC-type data</td>
</tr>
<tr>
<td>IOUTTYPE</td>
<td>1 = Perform simulation with provided source surface solar wind. 2 = In addition, perform simulation with +/- 200 km/sec source surface solar wind, output to earth_1.dat and earth_2.dat, respectively.</td>
<td>prtparam, solwind</td>
</tr>
<tr>
<td>JEX</td>
<td>Maximum number of particle samples handled in each R-T curves.</td>
<td>blkhlat, blkvel, ecplot, prtparam, sbt, solwind, srt, svrt</td>
</tr>
<tr>
<td>JOBS</td>
<td>Source of SEC data: 1 = NSO, 2 = WSO, 3 = MWO</td>
<td>srf</td>
</tr>
<tr>
<td>LATN LONN</td>
<td>“Maximum number of latitude and longitude values.” Note: Parameters are unused except for diagnostic print out.</td>
<td>prtparam</td>
</tr>
<tr>
<td>LOCATION</td>
<td>Location to forecast solar wind parameters: L1 or EARTH</td>
<td>prtparam, solwind, report</td>
</tr>
<tr>
<td>MAXFLR</td>
<td>Maximum number of flare events HAF is capable of handling.</td>
<td>ecplot, prtparam, report, solwind, source1, srf</td>
</tr>
<tr>
<td>ROOT</td>
<td>Name of the directory containing the input and output file subdirectory.</td>
<td>ecplot, prtparam, report, solwind, source1, srf</td>
</tr>
<tr>
<td>STARTYPE</td>
<td>Determines how the start time is derived: ‘SCHED’ = start time is at the nearest 12-hour period to the run time, input and output files accessed from the subdirectory named in DEFSUBDR. ‘EVENT’ = start time is at 00 UT nearest the start of the first flare event, input and output files accessed from the subdirectory identified in the haf.par file. ‘OTHER’ = start time is read from the startime.par file, input and output files accessed from the subdirectory identified in the haf.par file. ‘FIXED’ = start time is read from the startime.par file, input and output files accessed from the subdirectory named in DEFSUBDR.</td>
<td>ecplot, prtparam, report, solwind, source1, srf</td>
</tr>
</tbody>
</table>
The other category of constants are various physical and mathematical constants, algorithm coefficients and sampling rates.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Used in (routines)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1, B1, C1</td>
<td>Coefficients for calculating a weighted solar wind speed (vco) from the minimum and maximum wind speeds at the source surface.</td>
<td>calcv, ecplot, prtparam, solwind</td>
</tr>
<tr>
<td>AMINSSI</td>
<td>The effective shock search index detection threshold.</td>
<td>ssindex</td>
</tr>
<tr>
<td>AMP, BASE, POWER, AMPC, BASEC, POWERC</td>
<td>Solar wind algorithm (exponential) coefficients.</td>
<td>prtparam, srt</td>
</tr>
<tr>
<td>AMPF, BASEF</td>
<td>Exponential coefficients used to calculate impact of the flares on the solar wind velocity.</td>
<td>prtparam, srt</td>
</tr>
<tr>
<td>ARBIFAC</td>
<td>Flare event decay constant to flare time constant.</td>
<td>report</td>
</tr>
<tr>
<td>AU</td>
<td>Astronomical Unit (kilometers)</td>
<td>ecplot, prtparam, sbdt, solwind</td>
</tr>
<tr>
<td>AUINV</td>
<td>1/AU (km⁻¹)</td>
<td>prtparam, srt</td>
</tr>
<tr>
<td>DRPOINT</td>
<td>Radial distance increment for velocity computations (AU).</td>
<td>prtparam, sbdt</td>
</tr>
<tr>
<td>DTI</td>
<td>Time increment between R-T curves used to calculate solar wind speed (hours).</td>
<td>params, prtparam, solwind</td>
</tr>
<tr>
<td>FACT</td>
<td>Conversion factor from radians to degrees (degrees/radian).</td>
<td>prtparam, sbt</td>
</tr>
<tr>
<td>FCCTTI</td>
<td>Solar wind speed coefficient (km/sec).</td>
<td>prtparam, srt</td>
</tr>
<tr>
<td>FWIDTH</td>
<td>Forward shock length parameter.</td>
<td>prtparam, srt</td>
</tr>
<tr>
<td>NDTI</td>
<td>Number of R-T curve (UTIME) sampling times in the DTI period.</td>
<td>params, prtparam, svrt</td>
</tr>
<tr>
<td>OMEGA</td>
<td>Solar rotation rate (degrees/hour).</td>
<td>ecplot, params, prtparam, sbdt, solwind, srt</td>
</tr>
<tr>
<td>PF</td>
<td>Flare deceleration constant (hours).</td>
<td>report</td>
</tr>
<tr>
<td>PI</td>
<td>π</td>
<td>params, prtparam, sbdt</td>
</tr>
<tr>
<td>RAD</td>
<td>Conversion factor from degrees to radians (radians/degree).</td>
<td>ecplot, params, prtparam, sbt</td>
</tr>
<tr>
<td>RSS</td>
<td>Source surface radius (km).</td>
<td>params</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Data Type</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-----------</td>
</tr>
<tr>
<td>RSSAU</td>
<td>Source surface radius (AU).</td>
<td>sbt sbt srt srt</td>
</tr>
<tr>
<td>RSUN</td>
<td>Solar radius (km).</td>
<td>params srt</td>
</tr>
<tr>
<td>SHOCKF</td>
<td>Coefficients used in determining the position of forward shocks.</td>
<td>prtparam srt</td>
</tr>
<tr>
<td>SLANT</td>
<td>Coefficients used in determining the position of reverse shocks.</td>
<td>prtparam srt</td>
</tr>
<tr>
<td>SINTR</td>
<td>Coefficient used in determining the position of reverse shocks.</td>
<td>prtparam srt</td>
</tr>
<tr>
<td>TPERI</td>
<td>Solar rotation period (hours).</td>
<td>params prtparam srt</td>
</tr>
<tr>
<td>TWIDTH</td>
<td>Coefficient used in determining shock width.</td>
<td>prtparam srt</td>
</tr>
<tr>
<td>UNITP</td>
<td>Azimuthal sampling rate for R-T curves around the sun (degrees).</td>
<td>ecplot params prtparam sbdt sbt solwind srt</td>
</tr>
<tr>
<td>UPHI</td>
<td>Azimuthal sampling rate for R-T curves around the sun (radians).</td>
<td>prtparam sbdt</td>
</tr>
<tr>
<td>UTIME</td>
<td>R-T curve sampling rate in terms of suns rotation (hours).</td>
<td>ecplot params prtparam sbt solwind srt</td>
</tr>
</tbody>
</table>

Most parameters are not explicitly defined, so the data types are determined by F77 default typing, with the floating point parameters typed as REAL. Even when these parameters are promoted to eight byte floating point using compiler options, the value provided in the parameter declaration may be of insufficient precision to take advantage of the increased precision of the parameter itself.
### 4.12 Program globals (Common Blocks)

The HAF software uses many global variables to communicate data between subroutines. While these variables are organized into several common blocks, all of these common blocks are defined in a single (commons.f) file, so that any subroutine including this file has access to all of the global variables. The functional relationship of the common blocks to the HAF routines is displayed with arrowed connectors in the calling hierarchy diagrams, above. The arrows on the connectors display the direction of data flow.

The following table lists the global variables used by the HAF programs, the common block they are located in, the purpose of each variable, and the subroutines that use them. Each subroutine or program that sets a global variable has its name followed by an (O). Those that only read a global have their name followed by an (I). Routines that use the globals in only a local capacity have their names followed by neither character.

<table>
<thead>
<tr>
<th>Global</th>
<th>Common Block</th>
<th>Description</th>
<th>Used in (routines)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFLD</td>
<td>BLKBFLD</td>
<td>The magnetic field value on the source surface array ($\mu$T).</td>
<td>bmaxmin(I), calcv(I), ecplot(O), ldbfld(O), solwind(O), source1(I), srcfld(I)</td>
</tr>
<tr>
<td>BMAX</td>
<td>BLKBMAX</td>
<td>The maximum magnetic field at the source surface ($\mu$T).</td>
<td>bmaxmin(O), calcv(I), ecplot(O,I), solwind(O,I), source1(I), srcfld(I)</td>
</tr>
<tr>
<td>BMIN</td>
<td>BLKBMIN</td>
<td>The minimum magnetic field at the source surface ($\mu$T).</td>
<td>bmaxmin(O), calcv(I), ecplot(O,I), solwind(O,I), source1(I), srcfld(I)</td>
</tr>
<tr>
<td>BTEMP1</td>
<td>BLKTEMP</td>
<td>The magnetic field value on the source surface array ($\mu$T), read from Wilcox type source data (BTEMP1) or SEC type source data (BTEMP2).</td>
<td>ldbfld(I), srf(O)</td>
</tr>
<tr>
<td>BTEMP2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLRDEC</td>
<td>BLKFLR2</td>
<td>The decay time of each flare’s speed (hours), and the standard deviations of each flare’s width (degrees).</td>
<td>ecplot(O), report solwind(O,I), srt(I)</td>
</tr>
<tr>
<td>FLRSTD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLRLAT</td>
<td>BLKFLR1</td>
<td>Each flare’s HEQ latitude (degrees), longitude (degrees), and start time (hours from simulation start time).</td>
<td>ecplot(O), report solwind(O,I), srt(I)</td>
</tr>
<tr>
<td>FLRLON</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLTIME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLRSPD</td>
<td>BLKFLR2</td>
<td>Each flare’s maximum velocity (km/sec) and deceleration constant (hours).</td>
<td>ecplot(O), report solwind(O,I), srt(I)</td>
</tr>
<tr>
<td>POWERF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HFLD</td>
<td>BLKHLAT</td>
<td>R-T curve particle’s originating (source surface) magnetic field value (nT).</td>
<td>ecplot(I), solwind(I), srt(O)</td>
</tr>
<tr>
<td>Global</td>
<td>Common Block</td>
<td>Description</td>
<td>Used in (routines)</td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>HLAT</td>
<td>BLKHLAT</td>
<td>R-T curve particle’s originating (source surface) magnetic latitude (degrees).</td>
<td>sbdt(I) sbt(I) srt(O)</td>
</tr>
<tr>
<td>HLON</td>
<td>BLKHLAT</td>
<td>R-T curve particle’s originating (source surface) HEQ longitude (degrees).</td>
<td>sbt(I) solwind(I) srt(O)</td>
</tr>
<tr>
<td>IRDIAG</td>
<td>DIAG</td>
<td>Output diagnostic R-T curves, if set.</td>
<td>solwind(O) srt(I)</td>
</tr>
<tr>
<td>JEND</td>
<td>BLKRAD</td>
<td>Number of particles processed in each R-T curve.</td>
<td>ecplot(O,I) sbdt(I) sbt(I) solwind(O,I) srt(I) svrt(I)</td>
</tr>
<tr>
<td>JFD</td>
<td>JBLK</td>
<td>R-T particle index closest to the observation point and the interpolation factor to the observation point between the two adjacent R-T particles.</td>
<td>sbt(O,I) solwind(I)</td>
</tr>
<tr>
<td>NOFLR</td>
<td>BLKRAD</td>
<td>Number of flare records ingested.</td>
<td>ecplot(O,I) report solwind(O,I) srt(I)</td>
</tr>
<tr>
<td>NSLOPE1</td>
<td>NSLOPE2</td>
<td>BLKSRC</td>
<td>Slopes of the northern and southern magnetic field lines</td>
</tr>
<tr>
<td>SUBDR</td>
<td>BLKSUBDR</td>
<td>Name of the subdirectory containing the data files. The ROOT parameter provides the path to the subdirectory.</td>
<td>ecplot prtparam(I) report solwind(O,I) source1(O,I) srf(I)</td>
</tr>
<tr>
<td>VA</td>
<td>VB</td>
<td>BLKSRC</td>
<td>The maximum and minimum background solar wind speed.</td>
</tr>
<tr>
<td>VCO</td>
<td>VD</td>
<td>BLKSRC</td>
<td>A weighted solar wind speed and the solar wind speed range both derived from the minimum and maximum wind speeds at the source surface.</td>
</tr>
<tr>
<td>VEL</td>
<td>BLKVEL</td>
<td>Solar wind speed estimated at each point on an R-T curve (km/sec).</td>
<td>svrt</td>
</tr>
<tr>
<td>VFL</td>
<td>BLKVEL</td>
<td>Speed of flare particle (km/hour).</td>
<td>srt</td>
</tr>
<tr>
<td>VFLD</td>
<td>BLKFLD</td>
<td>The solar wind speed on the source surface array.</td>
<td>calcv(O) ecplot(O) ldfld(O) solwind(O) source1(I) srcvel(I) vmaxmin(I)</td>
</tr>
<tr>
<td>VMAX</td>
<td>BLKBMAX</td>
<td>The maximum solar wind speed at the source surface (km/sec).</td>
<td>ecplot solwind source1(I) vmaxmin(O)</td>
</tr>
<tr>
<td>Global</td>
<td>Common Block</td>
<td>Description</td>
<td>Used in (routines)</td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>VMIN</td>
<td>BLKBMIN</td>
<td>The minimum solar wind speed at the source surface (km/sec).</td>
<td>ecplot</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>solwind</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>source1(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>vmaxmin(O)</td>
</tr>
<tr>
<td>VSO</td>
<td>BLKVEL</td>
<td>Speed of the background solar wind particles at each point on an R-T curve (km/hr).</td>
<td>sbdt(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>srt(O,I)</td>
</tr>
<tr>
<td>VSW</td>
<td>BLKVEL</td>
<td>Average speed of an R-T curve particle from the source surface (km/sec).</td>
<td>sbdt(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>srt(O)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>svrt(I)</td>
</tr>
<tr>
<td>VTEMP2</td>
<td>BLKTEMP</td>
<td>The solar wind speed on the source surface array (km/sec).</td>
<td>ldbfld(I)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>srf(O)</td>
</tr>
</tbody>
</table>
5 Object descriptions

5.1 Fortran Objects

5.1.1 FUNCTION asindeg(x)

The asindeg function returns the principal value of the arc sine of x. The result lies in the range –90 to 90 degrees. A domain error occurs if |x| > 1.

Input argument:
   x – input argument to arc sine.

Returns: arc sine of x in degrees.

Called by:
   calcv
   hecheq
   heghec
   srcfld

Comments:
   • The function, its argument, and parameters are not explicitly defined and default to type REAL. Special compiler options must be invoked to take advantage of eight-byte processing.
   • Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
   • Uses private copies of the PI and FACT parameters instead of the ones available in params.f.

5.1.2 FUNCTION atan2deg(y,x)

The atan2deg function returns the principal value of the arc tangent of y/x, using the signs of both arguments to determine the quadrant of the return value. Viewed in terms of a Cartesian coordinate system, the result is the angle between the positive x-axis and a line drawn from the origin through point (x,y). The result lies in the range –180 to 180 degrees.

Input arguments:
   x – x coordinate argument to arc tangent
   y – y coordinate argument to arc tangent

Returns: arc tangent of y/x in degrees.
Called by:
  ctosve
  hecheq
  heqhec

Comments:

• The function, its arguments, and parameters are not explicitly defined and default to type REAL. Special compiler options must be invoked to take advantage of eight-byte processing.

• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Uses private copies of the PI and FACT parameters instead of the ones available in params.f.

5.1.3 SUBROUTINE bmaxmin(imax, jmx)

The bmaxmin subroutine searches the bfld array for the maximum and minimum magnetic field values.

Input arguments:
  imax, jmax – maximum array indices of the bfld array to search to.

Input global:
  BFLD – The magnetic field magnitude on the source surface array (μT).

Output globals:
  BMAX, BMIN – The maximum and minimum magnetic field found on the source surface (μT).

Called by: source1

Comments:

• The subroutine’s input arguments are not checked for valid range.

• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Uses non-ANSI F77 standard DO loop format that may complicate porting to other platforms and compilers.

• The algorithm assumes bmax >= 0 and bmin <= 0. One of these globals is set to 0 should the entire source surface field fall outside one of these ranges. But such a field is expected only in simulated data.
5.1.4 SUBROUTINE calcv

The calcv subroutine calculates the solar wind speed at the source surface from the magnetic field at the source surface and a basic solar wind model.

Input globals:
- BFLD – The magnetic field value on the source surface array (μT).
- BMAX, BMIN – The maximum and minimum magnetic field found on the source surface (μT).
- VA, VB – The maximum and minimum background solar wind speed (km/sec).

Output globals:
- VFLD – The solar wind speed on the source surface array (km/sec).

Calls:
- asindeg
- cosdeg

Called by: source1

Comments:

- The calcv subroutine is linked to the source1 program, but never called (unless the ISRCTYPE parameter is set to 1).
- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
- Uses non-ANSI F77 standard DO loop format that may complicate porting to other platforms and compilers.
- Divide-by-zero will occur when BMAX = 0.
- Computations are performed in single precision unless compiler options are used to promote REAL type variables and parameters to 8-byte floating point.
5.1.5 SUBROUTINE carrot(zjd, icrn)

The carrot subroutine determines the solar Carrington rotation number from the julian day provided. Reference Practical Astronomy with your calculator, 2nd Edition, Peter Duffett-Smith, Cambridge University Press, Pg 76.

Input argument:
  zjd – julian date.

Output argument:
  icrn – Carrington rotation number.

Called by: weshere

Comments:

- Missing 'IMPLICIT NONE' statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.

- Solar rotation parameters should be placed in shared params.f file.

5.1.6 FUNCTION cosdeg(x)

The cosdeg function returns the cosine of x, where x is measured in degrees.

Input argument:
  x – input argument to cosine (degrees).

Returns: cosine of x.

Called by:
  calcv
ctosve
ecplot
geotogei
hecheq
heqhec
srt
stoc
Comments:

- The function, its argument, and parameters are not explicitly defined and default to type REAL. Special compiler options must be invoked to take advantage of eight-byte processing.

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Uses private copies of the PI and degree to radian conversion parameters instead of the ones available in params.f.

5.1.7 FUNCTION cotan(x)

The cotan function returns the cotangent of \( x \), where \( x \) is measured in radians.

Input argument:

\[ x \] - input argument to cotangent (radians).

Returns: cotangent of \( x \).

Called by: sun

Comments:

- The function and its argument are not explicitly defined and default to type REAL. Special compiler options must be invoked to take advantage of eight-byte processing.

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Because of the way cotan is implemented, the function will overflow for \( x = (n + \frac{1}{2})\pi \) as well as for \( x = n\pi \).
5.1.8 SUBROUTINE cross(x1, x2, x3)

The cross subroutine computes the cross product of two vectors: 

\[ x3 = x1 \times x2. \]

Input arguments:

x1, x2 – the vectors to perform the cross product on.

Output argument:

x3 – the cross product of x1 and x2.

Called by:

geitogsm
gseqtogeigetriia2

Comments:

• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.

5.1.9 SUBROUTINE ctos(vin, r, t, p)

The ctos subroutine converts a cartesian vector into spherical coordinates, \( r, \pi/2-\theta, \) and \( \phi. \)

Input arguments:

vin – Cartesian vector.

Output arguments:

r,t,p – spherical coordinates \( r, \) latitude(radians), and longitude(radians) respectively.

Called by: sbt

Comments:

• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.

• Should use shared PI and other parameters placed in shared params.f file instead of private copies.
• Should not be possible to enter IF (t.LT.0.) block. If it were, p (longitude) would be left in the range [0 - 2π].

5.1.10 SUBROUTINE ctosve(theta,phi,uvc,uvs,thetab,phib)

The ctosve subroutine converts a Cartesian vector from heliocentric equatorial (HEQ) coordinates to Geo-Solar equatorial (GSEQ) coordinates, where the x-axis is aligned with the sun-earth line and the y-axis is in the plane of the solar equator.

Input arguments:
theta – HEQ latitude of observation point (degrees).
phi – HEQ longitude of observation point (degrees).
uvc – HEQ coordinate input vector.

Output arguments:
uvs – GSEQ coordinate output vector.
thetab – GSEQ spherical coordinate θ of uvs vector: magnetic field latitude angle (degrees).
phib – GSEQ spherical coordinate φ of uvs vector: magnetic field azimuth angle from sunward direction (degrees).

Called by: sbt

Comments:
• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

5.1.11 SUBROUTINE dot(x1,x2,x3)

The dot subroutine computes the dot (inner) product of two vectors: x3 = x1 • x2.

Input arguments:
x1, x2 – the vectors to perform the cross product on.

Output argument:
x3 – the dot (scalar) product of x1 and x2.

Called by: sbt

Comments:
• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
• Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.

5.1.12 PROGRAM ecplot

The ecplot program creates plot files of the ecliptic cross section of the IMF pattern due to the inner boundary source surface and as modified by flare propagation.

Input files:

`starttime.par` - The start time file containing the simulation start time. The start time file format is specified in the *Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document*. IYR,IMO,IDA,IHR,IMI - Start time of simulation.

`haf.par` - The control parameter file containing output control parameters. The control parameter file format is specified in the *Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document*. AUL - The radial distance to plot out to (AU). SUBDR - The subdirectory name containing the input and output files. JTS - The hour of the first plot generated relative to the simulation start time. JTE - The hour of the last plot generated. JTINC - The hour sampling increment between JTS and JTE.

`sorceb.par` - The source field file containing the SEC-type magnetic field map at the source surface. The source field file format is specified in the *Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document*. BMAX - The maximum magnetic field (µT). BMIN - The minimum magnetic field (µT). BFLD(IMAX2,JMAX) - The magnetic field on the source surface (µT).

`sorcev.par` - The source velocity file containing the solar wind speed map at the source surface. The source velocity file format is specified in the *Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document*. VMAX - The maximum solar wind speed (km/sec). VMIN - The minimum solar wind speed (km/sec). VFLD(IMAX2,JMAX) - The solar wind speed on the source surface (km/sec).
flare.par – The flare parameter file containing solar flare parameter records. The flare parameter file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

NOFLR – The number of flare records.
FLTIME(NOFLR) – Start time of flare (hours).
FLRLON(NOFLR) – Flare’s HEQ longitude (degrees).
FLRLAT(NOFLR) – Flare’s HEQ latitude (degrees).
FLRSPD(NOFLR) – Flare’s maximum speed (km/sec).
FLRDEC(NOFLR) – Flare’s time constant (hours).
FLRSTD(NOFLR) – Flare’s spatial extent (degrees).
POWERF(NOFLR) – Deceleration constant (hours).

sourcefld.par – The source field file containing the Wilcox-type magnetic field map at the source surface.
BMAX – The maximum magnetic field (µT).
BMIN – The minimum magnetic field (µT).
BFLD(IMAX1,JMAX) – The magnetic field on the source surface (µT).

Output files:
ecplotnnn.dat – The ecliptic plot files for each sample time. The ecliptic plot file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.
IAU – The radial distance of the plot out from the sun (AU).
IYRJ, MOJ, IDJ, IHJ – Plot sample time.
EX,EY – HEQ coordinates of the Earth in the plot (AU).
X1, Y1, X, Y – Coordinates of line segment endpoints depicting the IMF pattern (AU).
HFLD – Line segment’s originating (source surface) magnetic field value (nT).

Input globals:
HFLD – R-T curve particle’s originating (source surface) magnetic field value (nT).
Output globals:

- **BFLD** - The magnetic field magnitude on the source surface array (μT).
- **BMAX, BMIN** - The maximum and minimum magnetic field found on the source surface (μT).
- **FLRDEC** - The decay time of each flare’s speed (hours).
- **FLRLAT, FLRLON** - Each flare’s HEQ latitude and longitude (degrees).
- **FLRSPD** - Each flare’s maximum velocity (km/sec).
- **FLRSTD** - The standard deviations of each flare’s width (degrees).
- **FLTIME** - Each flare’s start time (hours from simulation start time).
- **JEND** - Number of particles processed in each R-T curve.
- **NOFLR** - Number of flare records ingested.
- **POWERF** - Each flare’s deceleration constant (hours).
- **VCO** - A weighted solar wind speed derived from the minimum and maximum wind speeds at the source surface (km/hr).
- **VFLD** - The solar wind speed on the source surface array (km/hr).

Calls:

- `cosdeg`
- `heqhec`
- `jdymd`
- `sindeg`
- `srt`
- `weshere`
- `ymdjd`

Called by: `runhaf`

Comments:

- The ecplot program must read either the sourceb.par or srcfld.par file. The srcfld.par file is never read (unless the ISRCTYPE parameter is set to 1).

- Extraneous variables: BMAX2, BMIN2, filename4, filename5, mswitch, s, VD, VMAX2 and VMIN2.

- Global variables only used locally: SUBDR, VA, VB, VMAX, and VMIN.

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Uses non-ANSI F77 standard DO loop format that may complicate porting to other platforms and compilers.

- Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.
• Relies on the REPORT program to ensure NOFLR <= MAXFLR so that flare parameter array bounds are not violated.

• Field line computations actually performed in HEQ coordinates, using the heqhec subroutine to estimate the HEQ latitude of the ecliptic for each R-T curve.

• No protection from attempting to open more than 999 ecplotnnn.dat files.

• Computation of ipstep truncated rather than rounded.

• Computations are performed in single precision unless compiler options are used to promote REAL type variables and parameters to 8-byte floating point.

5.1.13 SUBROUTINE geitogsm(s, gst, vin, vout)

The geitogsm subroutine converts a Cartesian vector from geocentric equatorial inertial (GEI) coordinates to Geocentric Solar magnetic (GSM) coordinates, where the x-axis is aligned with the sun-earth line and the y-axis is perpendicular to the Earth’s magnetic dipole. References: Kivelson, M. G. and C. T. Russell, Introduction to Space Physics, Cambridge Univ. Press, Appendix 3, 1996 and http://www.spenvis.oma.be/spenvis/Help/background/coortran/coortran.html.

Input arguments:
  s – Vector pointing from earth to sun in GEI coordinates obtained with a call to the sun subroutine. This is the X-axis.
  gst – Greenwich sidereal Time (degrees).
  vin – GEI coordinate input vector.

Output arguments:
  vout – GSM coordinate output vector.

Calls:
  cross
  geotogei
  uvect

Called by: gseqtogsm

Comments:

• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.
• The dipole axis vector should be updated every five years using the latest IGRF.

• Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.

• Specification statement placed after data statement.

• The s argument is assumed a unit vector, but no checks or preparation are performed.

5.1.14 SUBROUTINE geotogei(gst, vin, vout)

The geotogei subroutine converts a Cartesian vector from geographic (GEO) coordinates to geocentric equatorial inertial (GEI) coordinates, where the x-axis is aligned with the first point of Aries and the z-axis is parallel to the Earth’s axis. References: Kivelson, M. G. and C. T. Russell, Introduction to Space Physics, Cambridge Univ. Press, Appendix 3, 1996 and http://www.spenvis.oma.be/spenvis/Help/background/coortran/coortran.html.

Input arguments:
   gst – Greenwich sidereal Time (degrees).
   vin – GEO coordinate input vector.

Output arguments:
   vout – GEI coordinate output vector.

Calls:
   cosdeg
   sindeg

Called by: geitogsm

Comments:

• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.
5.1.15 SUBROUTINE getutc(iyr, imo, id, ihr, imin, secs, iutc, jd)

The getutc subroutine calls DATE_AND_TIME to get the UT date, time, and julian date.

Output arguments:
- iyr - year.
- imo - month.
- id - day of the month.
- ih - hour of the day.
- imin - minutes of the hour.
- secs - fractional seconds of the minute.
- iutc - hours to add to get UTC.
- jd - julian date.

Calls:
- jdymd
- ymdjd

Called by:
- prtparam
- report
- solwind

Comments:
- Calls DATE_AND_TIME FORTRAN 90 subroutine.
- No validity check of outputs from DATE_AND_TIME.
- Missing 'IMPLICIT NONE' statement to promote explicit data typing and protect against using mistyped variables and parameters.
- Leap second events may change number of seconds (of the minute) between local and Greenwich.
- Computation of dayfrac is performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point. Otherwise up to millisecond precision is lost.
- One minute errors are possible because imin is recalculated after call to jdymd, but secs are not.
5.1.16 SUBROUTINE gseqtogeis(vin, vout)

The gseqtogeis subroutine converts a Cartesian vector from geocentric solar equatorial (GSEQ) coordinates to geocentric equatorial inertial (GEI) coordinates, where the x-axis is aligned with the first point of Aries and the z-axis is parallel to the Earth’s axis. References: Kivelson, M. G. and C. T. Russell, Introduction to Space Physics, Cambridge Univ. Press, Appendix 3, 1996 and http://www.spenvis.oma.be/spenvis/Help/background/coortran/coortran.html.

Input arguments:
- vin – GEO coordinate input vector.

Output arguments:
- vout – GEI coordinate output vector.

Calls:
- cross
- uvect

Called by: gseqtogsm

Comments:
- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
- Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.
- Is (private) solar rotation axis consistent with rest of model?
- The s argument is assumed a unit vector, but no checks or preparation are performed.
5.1.17 SUBROUTINE gseqtogsm(s, gst, vin, vout)


Input arguments:
- s – Vector pointing from earth to sun in GEI coordinates obtained with a call to the sun subroutine. This is the X-axis.
- gst – Greenwich sidereal Time (degrees).
- vin – GSEQ coordinate input vector.

Output arguments:
- vout – GSM coordinate output vector.

Calls:
- geitogsm
- gseqtogei

Called by: sbt

Comment:
- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

5.1.18 SUBROUTINE hecheq(iyr, zlatec, zlonec, zlateq, zloneq)

The hecheq subroutine transforms latitude and longitude in Heliocentric ecliptic coordinates (HEC) to heliocentric equatorial coordinates (HEQ).

Input arguments:
- iyr – 4-digit year (needed to get alpha).
- zlatec, zlonec – HEC latitude and longitude to transform (degrees).

Output arguments:
- zlateq, zloneq – HEQ latitude and longitude (degrees).

Calls:
- asindeg
- atan2deg
- cosdeg
- sindeg
Called by: weshere

Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Uses private copies of solar coordinate parameters instead of ones available in params.f.

- Uses obsolete solar inclination measurement.

- Precision of equinox precession term is about 0.004 of a degree.

- Input arguments not checked for valid range.

- Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.

5.1.19 SUBROUTINE helio(zjd, l0, b0, p, eclon, sunr)

The helio subroutine returns the Carrington latitude and longitude of the Earth, and P angle for the input Julian date.

Input arguments:
  zjd – fractional Julian day.

Output arguments:
  l0 – The Earth’s Carrington longitude (degrees).
  b0 – Solar latitude of the Earth (degrees).
  p – Angle between the Sun’s rotational axis and the Earth’s (degrees). The Earth’s ecliptic longitude from the Sun is eclon + 180.
  sunr – Distance from the sun to the Earth (AU).

Calls:
  sunloc

Called by: weshere

Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Uses non-ANSI F77 standard DO WHILE loop format that may complicate porting to other platforms and compilers.

- Parameters pi3over2 and piover2 are not used.
• Uses private copies of PI and other parameters instead of shared parameters in the params.f file.

• Uses obsolete solar inclination measurement.

• 'DO WHILE (MP .ge. 360D0)' loop never entered.

### 5.1.20 SUBROUTINE heqhec(iyr, zlateq, zloneq, zlatec, zlonec)

The heqhec subroutine transforms latitude and longitude in heliocentric equatorial coordinates (HEQ) to heliocentric ecliptic coordinates (HEC).

**Input arguments:**
- iyr – 4-digit year (needed to get alpha).
- zlateq, zloneq – HEQ latitude and longitude to transform (degrees).

**Output arguments:**
- zlatec, zlonec – HEC latitude and longitude (degrees).

**Calls:**
- asindeg
- atandeg
- cosdeg
- sindeg

**Called by:** ecplot

**Comments:**

• Missing 'IMPLICIT NONE' statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Uses private copies of solar coordinate parameters instead of ones available in params.f.

• Uses obsolete solar inclination measurement.

• Precision of equinox precession term is about 0.004 of a degree.

• Input arguments not checked for valid range.

• Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.
5.1.21 FUNCTION iday(iyr, mo, id)

The iday function calculates the number of the day in the input year.

Input arguments:
   iyr – 4-digit year.
   mo – month (1 - 12)
   id – day of month.

Returns: day number (starts at 1 for Jan. 1).

Called by:
   sbt
   weshere

Comments:

• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Input arguments not checked for valid range (to prevent out of array bounds access, in particular).

• Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.

5.1.22 SUBROUTINE interp3(x,y,z)

The interp3 subroutine interpolates z(5) at point [x(5),y(5)] interior to a coordinate-aligned rectangle formed by the four corners [x(i),y(i)], i = 1-4, with values z(i). The point [x(5), y(5)] must be interior, or on the border of, the rectangle. No error if out of bounds, but bad values result.

Input arguments:
   x(1-4), y(1-4) – The coordinates of the four corners.
   z(1-4) – The values at the four corners.
   x(5), y(5) – The coordinates of the evaluation point.

Output argument:
   z(5) – The bilinear interpolated value at [x(5), y(5)].

Called by:
   interplon
   srcfld
   srcvel

Comments:
• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• The input criteria are over specified, given that the algorithm assumes bilinear interpolation from a coordinate-aligned rectangle. The components, x(2), x(3), y(3), and y(4) are neither used nor checked for conformity with a coordinate-aligned rectangle.

• Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.

• Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.

• Protection from zero-divide, but not overflow.

5.1.23 SUBROUTINE interplon(long,win,wout)

The interplon subroutine interpolates an array to the nearest 5-degree longitude increment greater than the input longitude. Interplon inputs the array win(36,72) and the longitude long, and outputs wout(36,73) shifted to the right by mod(long,5) degrees, and wrapped around at the 360/0 degree longitude boundary. Interplon calls interp3 to perform the bilinear interpolation.

Input arguments:
  long – Starting longitude of 72 row array
  win - Input array (72 longitudes from long to long+355 degrees and 36 latitudes from +87.5 to -87.5 degrees). Note the array may overlap two Carrington rotations (this one and the last one).

Output argument:
  wout – The interpolated array.

Called by:
  sourcesec

Comments:
  • Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
  • Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.
  • Uses non-ANSI F77 standard DO loop format that may complicate porting to other platforms and compilers.
5.1.24 SUBROUTINE jdymd(jd, iyr, month, day)


Input argument:
   jd – Julian date.

Output arguments:
   iyr – Year.
   mo – Month.
   day – fractional day of the month.

Called by:
   ecplot
   getutc
   report
   sbt
   solwind

Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
- Variable b not set if jd < 2299160.5, (in which case variable a is set but never used).
- Arguments and most variables are not explicitly defined and floating point variables default to type REAL. Most computations are performed in single precision and are of insufficient precision (in the case of computing d) unless compiler options are used to promote REAL type variables and parameters to 8-byte floating point. The variables a, b, c, d, e, and g would have greater precision as 4-byte integers than as 4-byte reals.

5.1.25 SUBROUTINE ldbfld(irot,zl0)

The ldbfld subroutine loads up the source field and velocity arrays from Stanford data. The counterpart to this routine are the srcfld and srcvel which read the value from bfld(i,j) and vfld(i,j) using bilinear interpolation.

bfld and vfld are arrays containing three rotations of data. This ought to be enough for the simulations of current sheets from Stanford data.

HEQ LON = -355  …  0.0  …  360.0  …  720.0

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>j =</td>
<td>1</td>
<td>72</td>
</tr>
</tbody>
</table>

Input arguments:
irot - Carrington rotation number of HEQ zero longitude at start time.

z10 - Carrington longitude corresponding to zero HEQ longitude.

Input globals:
BTEMP1, BTEMP2 - The magnetic field value on the source surface array (μT), read from Wilcox type source data (BTEMP1) or SEC type source data (BTEMP2).
VTEMP2 - The solar wind speed on the source surface array (km/sec).

Output globals:
VFLD - The solar wind speed on the source surface array (km/sec).
BFLD - The magnetic field value on the source surface array (μT).

Calls:
srf

Called by: source1

Comments:

• Missing 'IMPLICIT NONE' statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Deceptive printout of nrot (= irot - 1) which is subsequently set to irot when ingesting SEC data.

• Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.

• Incrementing of nrot overridden when ingesting SEC data.

• Range checking excludes z10 of 0.0 and 360.0.

5.1.26 SUBROUTINE prtparam

The prtparam subroutine prints out the values of most of the parameters set in the params.f file.

Input global:
subdr - Name of the subdirectory containing the data files.

Calls:
getutc

Called by: solwind

Comments:
• The prtparam subroutine is linked to the solwind program, but never called (unless the IDIAGFLG parameter is set to 1).

• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• subdr not set when printed.

5.1.27 SUBROUTINE r2(e, m, ea)

The r2 subroutine finds the solution to Kepler’s equation \( ea - e \cdot \sin(ea) = M \), using an iterative process.

Input arguments:
- \( e \) – orbit eccentricity.
- \( m \) – mean anomaly (radians).

Output arguments:
- \( ea \) – eccentric anomaly (radians).

Constant:
- \( \epsilon s \) – the error allowed between \( ea \) and analytic \( ea \).

Called by: sunloc

Comments:

• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Uses non-ANSI F77 standard DO loop format that may complicate porting to other platforms and compilers.

• No check of inputs for valid values.

• Uses private copies of PI and other parameters instead of shared parameters in the params.f file.

5.1.28 SUBROUTINE rdreprec(filename, ievnum, iyrb, mob, idb, ihb, mib, latf, lonf, vs, tau, vsw, sigma, ihm, mim, ihe, mie, imp, bril, xray, stn, sat, ierr)

The rdreprec subroutine reads a flare event record from the solar event file, parsing the 15 space-delimited fields. Rdreprec performs superficial (domain) input validation for most of the fields in each record. An invalid event number field is associated with an end of data (EOF). Other invalid fields will result in an IERR of -2 and incomplete processing. Each record is expected to have at least the first six fields (event number,
date, time, latitude, longitude, and shock speed). The remaining nine may be identified as missing with an 'XXX' field or left blank.

Input arguments:
- filename – Path/file name of the solar event file to open and read records from.

Input files:
- IEVNUM – Event number.
- IYRB, MOB, IDB – Event start year, month, and day: YYYYMMDD.
- IHB, MIB – Event start hour and minutes: HHMM.
- SLAT, LATF – Source heliographic latitude (N or S, degrees).
- SLON, LONF – Source heliographic longitude (E or W, degrees).
- VS – Type-II shock speed (km/s).
- TAUHM – Event decay constant (HHMM).
- VSW – Background solar wind speed (km/s).
- SIGMA – Area extent of event source (deg).
- IHM, MIM – Event max time (HHMM).
- IHE, MIE – Event end time (HHMM).
- IMP, BRIL – Optical Flare Classification, importance and brilliance.
- XRAY – X-ray event class and flux.
- STN – Station ID of source of optical data.
- SAT – Satellite ID of source of X-ray data.
Output arguments:

IEVNUM – Event number.
IYRB, MOB, IDB – Event start year, month, and day: YYYYMMDD.
IHB, MIB – Event start hour and minutes: HHMM.
SLAT, LATF – Source heliographic latitude (N or S, degrees).
SLON, LONF – Source heliographic longitude (E or W, degrees).
VS – Type-II shock speed (km/s).
TAU – Event decay constant (hours).
VSW – Background solar wind speed (km/s).
SIGMA – Area extent of event source (deg).
IHM, MIM – Event max time (HHMM).
IHE, MIE – Event end time (HHMM).
IMP, BRIL – Optical Flare Classification, importance and brilliance.
XRAY – X-ray event class and flux.
STN – Station ID of source of optical data.
SAT – Satellite ID of source of X-ray data.
IERR – Error Status:
   =0    Record read successfully, all fields set
   =1    EOF no more records
   =-1   File Error, no data available
   =-2    Record Error, record data incomplete
   = 128 (2^7) TAU set to 'XXX'
   = 256  VSW set to 'XXX'
   = 512  SIGMA set to 'XXX'
   = 1024 IHM/MIM set to 'XXX'
   = 2048 IHE/MIE set to 'XXX'
   = 4096 IMP/BRIL set to 'XXX'
   = 8192 XRAY set to 'XXX'
   = 16384 STN set to 'XXX'
   = 32768 SAT set to 'XXX'

Called by: report

5.1.29 PROGRAM report

The report program reads in flare reports to prepare the flare parameters used by the solwind and ecplot programs to simulate the solar wind using the Hakamada-Akasofu-Fry kinematic method. It also calculates the Carrington Rotation number (CR), the heliographical longitude, and the Earth’s location in heliocentric equatorial coordinates.

Input files:

haf.par – The control parameter file containing output control parameters. The control parameter file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.
SUBDR – The subdirectory name containing the input and output files.
Output files:

```
startime.par – The start time file containing the
simulation start time. The start time is set to the
nearest 00 UT to the first event record in the
report.dat file. The start time file format is
specified in the Hakamada-Akasofu-Fry (HAF) Kinematic
Solar Wind Model Functional Requirements Document.
IYR,IMO,IDA,IHR,IMI - Start time of simulation.

flare.par – The flare parameter file containing solar flare
parameter records. The flare parameter file format
is specified in the Hakamada-Akasofu-Fry (HAF)
Kinematic Solar Wind Model Functional Requirements
Document.
NOFLR - The number of flare records.
FLTIME(NOFLR) - Start time of flare (hours).
FLRLON(NOFLR) - Flare’s HEQ longitude (degrees).
FLRLAT(NOFLR) - Flare’s HEQ latitude (degrees).
FLRSPD(NOFLR) - Flare’s maximum speed (km/sec).
FLRDEC(NOFLR) - Flare’s time constant (hours).
FLRSTD(NOFLR) - Flare’s spatial extent (degrees).
POWERF(NOFLR) - Deceleration constant (hours).

earth.par – The Earth Location file for the specified start
time. The Earth Location file format is specified in
the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind
Model Functional Requirements Document.
IROT - The Carrington rotation number of the HEQ zero
longitude at the simulation start time.
ZL0 - Carrington longitude corresponding to HEQ zero
longitude (degrees).
ELAT, ELON - Earth’s HEQ Latitude and Longitude
(degrees).
SUNR - Distance from the sun to the Earth (AU).
```

Calls:
```
getutc
jdymd
weshere
ymdjd
```

Called by: runhaf

Comments:
```
• The srcvel.par file is never written (unless the ISRCTYPE
parameter is set to 1).

• Extraneous variable: isd.

• Global variables only used locally: SUBDR, VA, VB, FLRDEC,
FLRLAT, FLRLON, FLRSPD, FLRSTD, FLTIME, NOFLR, NSLOPE1,
NSLOPE2, and POWERF.
```
• Missing ‘IMPLICIT NONE’ statement to promote explicit data
typing and protect against using mistyped variables and
parameters.

• Uses non-ANSI F77 standard DO loop format that may
complicate porting to other platforms and compilers.

• Uses non-ANSI F77 standard (inline) comments that may
complicate porting to other platforms and compilers.

• flrstd is set to an arbitrary (non-zero) value when it
cannot be determined from the input data for a flare.

5.1.30 SUBROUTINE sbdt(rpoint, r, bm, de)

The sbdt subroutine calculates the magnetic field magnitude and
density at the observation point along the R-T curve.

Input arguments:
  rpoint – Radial distance of observation point (AU).
  r – R-T curve containing the particle array along a
      radial(AU).

Output arguments:
  bm – Calculated magnetic field magnitude.
  de – Calculated particle density (cm$^{-3}$).

Input globals:
  HLAT – R-T curve particle’s originating (source surface)
          magnetic latitude (degrees).
  JEND – Number of particles processed in each R-T curve.
  VSO – Speed of the background solar wind particles at each
         point on an R-T curve (km/hr).
  VSW – Average speed of an R-T curve particle from the
         source surface (km/sec).

Calls:
  sindeg

Called by: solwind

Comments:

• Missing ‘IMPLICIT NONE’ statement to promote explicit data
typing and protect against using mistyped variables and
parameters.

• Computations are performed in single precision unless
  compiler options are used to promote REAL type variables
  and arguments to 8-byte floating point.

• Relies on the solwind program to set JEND within the array
  bounds of R.
• J2 index set greater than JEND when RPOINT > R(JEND-1). J2 then indexes into arrays outside the bounds of processed data or potentially outside the bounds of the arrays themselves.

• Calculated magnetic field magnitude not used by solwind.

• Uses data from VSW array without converting to km/hr (only impacts magnetic field magnitude output).

5.1.31 SUBROUTINE sbt(zjd, ztime, aul, zlat1, zlong1, itype, bvect)

The sbt subroutine calculates the interplanetary magnetic field vector at time ZTIME and location [aul, zlat1, zlong1]. A triangular flux tube is used in the calculation. Refer to Ghee Fry’s thesis, Chapter 2, for the details of the algorithm.

Input arguments:
  zjd – Fractional Julian date of calculated effect at observation point.
  ztime – Time (from simulation start time) of calculated effect at observation point (hours).
  zlat1, zlong1 – HEQ latitude and longitude of the observation point (degrees).
  itype – IMF output coordinate type:
      = 0 then Bx, By, Bz (heliocentric cartesian)
      = 1 then Bx, By, Bz, /B/, Bφ (GSEQ)
      = 2 then Bx, By, Bz, /B/, Bφ (GSM)
      = 3 then don't calculate B field

Output arguments:
  bvect – Six-element magnetic field vector (nT).

Input globals:
  HLAT – R-T curve particle’s originating (source surface) magnetic latitude (degrees).
  HLON – R-T curve particle’s originating (source surface) HEQ longitude (degrees).
  JEND – Number of particles processed in each R-T curve.

Output globals:
  JFD – R-T particle index closest to the observation point.
  RFAC – The interpolation factor to the observation point between the two adjacent R-T particles.
Calls:
  ctos
  ctosve
dot
qseqtogsm
iday
jdymd
srcfld
srt
stoc
sun
tria2
uvect
vect

Called by: solwind

Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Uses non-ANSI F77 standard DO loop format that may complicate porting to other platforms and compilers.

- Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.

- BF and SMLAT printed before being set (if IWFLAG = 1).

- Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.

5.1.32 FUNCTION sindeg(x)

The sindeg function returns the sine of x, where x is measured in degrees.

Input argument:
  x - input argument to sine (degrees).

Returns: sine of x.
Called by:
calcv
ctosve
ecpot
geotogei
hecheq
heqhec
sbdx
srcfld
srcvel
srt
stoc

Comments:

- The function, its argument, and parameters are not explicitly defined and default to type REAL. Special compiler options must be invoked to take advantage of eight-byte processing.

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Uses private copies of the PI and degree to radian conversion parameters instead of the ones available in params.f.

5.1.33 PROGRAM solwind

The solwind program uses the magnetic field data of the source surface to simulate the interplanetary magnetic field, solar wind speed, and density using the Hakamada and Akasofu kinematic method.

Input files:

starttime.par – The start time file containing the simulation start time. The start time file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document. IYR, MO, ID, IH, IMI – Start time of simulation.


IYR, MO, ID, IH, IMI – Start time of simulation.

IHRS – Start hour of output, relative to the simulation start time (hours).

NHRS – Number of hours from IHRS to output at an hourly sampling rate.

SUBDR – The subdirectory name containing the input and output files.
sourceb.par – The source field file containing the SEC-type magnetic field map at the source surface. The source field file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

BMAX – The maximum magnetic field (μT).
BMIN – The minimum magnetic field (μT).
BFLD(IMAX2,JMAX) – The magnetic field on the source surface (μT).

sourcev.par – The source velocity file containing the solar wind speed map at the source surface. The source velocity file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

VMAX – The maximum solar wind speed (km/sec).
VMIN – The minimum solar wind speed (km/sec).
VFLD(IMAX2,JMAX) – The solar wind speed on the source surface (km/sec).

flare.par – The flare parameter file containing solar flare parameter records. The flare parameter file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

NOFLR – The number of flare records.
FLTIME(NOFLR) – Start time of flare (hours).
FLRLON(NOFLR) – Flare’s HEQ longitude (degrees).
FLRLAT(NOFLR) – Flare’s HEQ latitude (degrees).
FLRSPD(NOFLR) – Flare’s maximum speed (km/sec).
FLRDEC(NOFLR) – Flare’s time constant (hours).
FLRSTD(NOFLR) – Flare’s spatial extent (degrees).
POWERF(NOFLR) – Deceleration constant (hours).

sourcefld.par – The source field file containing the Wilcox-type magnetic field map at the source surface.

BMAX – The maximum magnetic field (μT).
BMIN – The minimum magnetic field (μT).
BFLD(IMAX1,JMAX) – The magnetic field on the source surface (μT).
Output files:

**earth_0.dat** - The simulated field and velocity file contains an IMF forecast record for the L1 observation point at each sample time. The simulated field and velocity file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

**NHRS** - Number of hours from IHRS to output at an hourly sampling rate.

**IYR, MO, ID, IH** - Start time of the simulation.

**IROT** - The Carrington Rotation number at the simulation start time.

**AUL** - The distance of the L1 observation point from the sun (AU).

**ITYPE** - The BIMF output coordinate type:
- 0 then Bx, By, Bz (heliocentric cartesian)
- 1 then Bx, By, Bz, /B/, Bθ, BΦ (GSEQ)
- 2 then Bx, By, Bz, /B/, Bθ, BΦ (GSM)
- 3 then don't calculate B field

**ZTIME** - The hour, from the simulation start time, of the forecast record (hours).

**WSPD** - The solar wind speed (km/sec).

**DENS** - Particle density (cm⁻³).

**BIMF** - Magnetic field components as determined by ITYPE (nT).

**ZLAT1, ZLONG1** - HEQ latitude and longitude of the Earth (degrees).

**ssi_0.dat** - The shock file contains the forecast shock arrival time at the L1 observation point and the dynamic pressure and shock strength index at each sample time. The shock file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

**SAT** - The forecast shock arrival time relative to the simulation start time (hours).

**IYRJ, MOJ, IDJ, IHJ** - The year, month, day, and hour of the forecast shock.

**PRESSU** - The forecast dynamic pressure at the L1 observation point at each simulation sample time (nPa).

**SSI** - Shock strength index at the L1 observation point at each simulation sample time.

Input globals:

**HFLD** - R-T curve particle’s originating (source surface) magnetic field value (nT).

**HLON** - R-T curve particle’s originating (source surface) HEQ longitude (degrees).

**JFD** - R-T particle index closest to the observation point.

**RFAC** - The interpolation factor to the observation point between the two adjacent R-T particles.
Output globals:

- **BFLD** - The magnetic field magnitude on the source surface array (μT).
- **BMAX, BMIN** - The maximum and minimum magnetic field found on the source surface (μT).
- **FLRDEC** - The decay time of each flare’s speed (hours).
- **FLRLAT, FLRLON** - Each flare’s HEQ latitude and longitude (degrees).
- **FLRSPD** - Each flare’s maximum velocity (km/sec).
- **FLRSTD** - The standard deviations of each flare’s width (degrees).
- **FLTIME** - Each flare’s start time (hours from simulation start time).
- **IRDIAG** - Controls output of R-T curves in diagnostic mode.
- **JEND** - Number of particles processed in each R-T curve.
- **NOFLR** - Number of flare records ingested.
- **POWERF** - Each flare’s deceleration constant (hours).
- **VCO** - A weighted solar wind speed derived from the minimum and maximum wind speeds at the source surface (km/hr).
- **VFLD** - The solar wind speed on the source surface array (km/hr).

Calls:

- `getutc`
- `jdymd`
- `prtparam`
- `sbdt`
- `sbt`
- `srt`
- `ssindex`
- `svrt`
- `weshere`
- `ymdjd`

Called by: *runhaf*

Comments:

- The solwind program must read either the sourceb.par or srcfld.par file. The srcfld.par file is never read (unless the ISRCTYPE parameter is set to 1).

- Global variables only used locally: `SUBDR`, `VA`, `VB`, `VD`, `VMAX`, `VMIN`, and `ZL00`.

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Uses non-ANSI F77 standard DO loop format that may complicate porting to other platforms and compilers.

- Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.
- Relies on the REPORT program to ensure NOFLR <= MAXFLR so that flare parameter array bounds are not violated.
- No protection from NHRS being set to more than JEX sample times and causing indexing outside array boundaries.

### 5.1.34 PROGRAM source1

The source1 program prepares the magnetic source field and wind speed arrays for the SOLWIND and ECPLT programs to use. It wraps the source data to cover three rotations, from -355 to 720 degrees longitude, shifting the data from Carrington coordinates to Heliospheric Equatorial coordinates.

**Input files:**

- `haf.par` - The control parameter file containing output control parameters. The control parameter file format is specified in the *Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document*.
- `SUBDR` - The subdirectory name containing the input and output files.
- `earth.par` - The Earth Location file for the specified start time. The Earth Location file format is specified in the *Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document*.
- `IROT` - The Carrington rotation number of the HEQ zero longitude at the simulation start time.
- `ZL0` - Carrington longitude corresponding to HEQ zero longitude (degrees).

**Output files:**

- `sourceb.par` - The source field file containing the SEC-type magnetic field map at the source surface. The source field file format is specified in the *Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document*.
  - `BMAX` - The maximum magnetic field (μT).
  - `BMIN` - The minimum magnetic field (μT).
  - `BFLD(IMAX2,JMAX)` - The magnetic field on the source surface (μT).
- `sourcev.par` - The source velocity file containing the solar wind speed map at the source surface. The source velocity file format is specified in the *Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document*.
  - `VMAX` - The maximum solar wind speed (km/sec).
  - `VMIN` - The minimum solar wind speed (km/sec).
  - `VFLD(IMAX2,JMAX)` - The solar wind speed on the source surface (km/sec).
sourcefld.par - The source field file containing the Wilcox-type magnetic field map at the source surface. 
BMAX - The maximum magnetic field (\( \mu T \)). 
BMIN - The minimum magnetic field (\( \mu T \)). 
BFLD(IMAX1,JMAX) - The magnetic field on the source surface (\( \mu T \)).

Input globals:
BFLD - The magnetic field magnitude on the source surface array (\( \mu T \)).
BMAX, BMIN - The maximum and minimum magnetic field found on the source surface (\( \mu T \)).
VFLD - The solar wind speed on the source surface array (km/sec).
VMAX, VMIN - The maximum and minimum solar wind speed found on the source surface (km/sec).

Output globals:
NSLOPE1, NSLOPE2 - Slopes of the northern and southern magnetic field lines.
SUBDR - The subdirectory name containing the input and output files.
VA, VB - The maximum and minimum background solar wind speed (km/sec).

Calls:
bmaxmin
calcv
ldbfld
vmaxmin

Called by: runhaf

Comments:

- The solwind program must read either the sourceb.par or srcflld.par file. The srcflld.par file is never read (unless the ISRCTYPE parameter is set to 1).

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.

5.1.35 PROGRAM sourcesec

The sourcesec program prepares the source surface magnetic field and velocity data from NOAA/SEC. Sourcesec reads SEC/RPC files and load output files with solar surface magnetic field (B) and velocity (V) data, which provide boundary conditions for the HAF model. The input array does not necessarily begin at a Carrington rotation boundary so the end of the array is copied to the beginning so that the data is arranged as follows:
array(i,j) i=1 (lon=0) to i=72 (lon=355) in 5 degree increments
j=1 (lat=+87.5) to j=36 (lat=-87.5) in 5 degree increments
The longitudes are interpolated to the nearest 5 degree longitude increment.

Input files:

- haf.par - The control parameter file containing output control parameters. The control parameter file format is specified in the **Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document**.
- SUBDR - The subdirectory name containing the input and output files.
- MAPTYPE - The model used to generate the input SEC source surface map data.
- sourcefilelist.txt - The SEC Map File List containing the name(s) of the SEC source surface map file(s). The format of the SEC Map File List file is specified in the **Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document**.
- FILENAME2 - The name(s) of the SEC data file(s) that will be processed by sourcesec.

SEC data files - The SEC source surface map files listed in sourcefilelist.txt. The format of the SEC map files is specified in the **Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document**.
- ICRNUM - The Carrington rotation covered by the file, parsed from the file name.
- ICRLON - The Carrington longitude, parsed from the file name.
- JOBS - The observatory name, parsed from the file name.
- NCARR - Carrington rotation covered by the file.
- LONG - Carrington longitude.
- BSS - Source Surface B field (Gauss).
- FEXP - Flux-tube expansion factor evaluated at the source surface.
- THETA - The colatitude (radians).
- PHI - The longitude (radians).
- BPHOT - The photospheric field (Gauss) for every 5 degrees of longitude and latitude at the photosphere.

Output files:

- bfield.dat - The B field map file containing the SEC-type magnetic field map at the source surface. The B field map file format is specified in the **Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document**.
- ICRNUM - The Carrington rotation number covered by the map file.
ICRLON – Carrington longitude corresponding to zero HEQ longitude.
JOBS – The observatory name.
[index value] – Carrington longitude of each record of map data (degrees).
BB – The magnetic field on the source surface (µT).

velfield.dat – The velocity map file containing the solar wind speed map at the source surface. The velocity map file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.
ICRNUM – The Carrington rotation number covered by the map file.
ICRLON – Carrington longitude corresponding to zero HEQ longitude.
ILON – Carrington longitude of each record of map data (degrees).
V2(IMAX2,KMAX) – The solar wind speed at the source surface (km/sec).

b_xxx.dat – The B field map file covering one frame, or one SEC source surface data file.
ICRNUM – The Carrington rotation number covered by the map file.
ICRLON – Carrington longitude corresponding to zero HEQ longitude.
BB – The magnetic field on the source surface (µT).

v_xxx.dat – The velocity map file covering one frame, or one SEC source surface data file.
ICRNUM – The Carrington rotation number covered by the map file.
ICRLON – Carrington longitude corresponding to zero HEQ longitude.
V2(IMAX2,KMAX) – The solar wind speed at the source surface (km/sec).

Calls:
interplon

Called by:
runhaf

Comments:

• The solwind program must read either the sourceb.par or srcfld.par file. The srcfld.par file is never read (unless the ISRCTYPE parameter is set to 1).

• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.
• Uses non-ANSI F77 standard DO loop format that may complicate porting to other platforms and compilers.

• Global parameters STARTYPE, ROOT, IDIAGFLG, and DEFSUBDR, defined in params.f, are redefined and reinitialized here. The default STARTYPE value of ‘OTHER’ should not be changed. Code referring to a STARTYPE value different than the default will be removed in HAF v3.0.

• The movie frame output files b_xxx.dat and vel_xxx.dat contain the same data as the files bfield.dat and velfield.dat when used after a call to fusionshell.dat.

5.1.36 SUBROUTINE srcfld(timem, zlat, zlong, bmag, smlat)

The srcfld subroutine determines the source surface magnetic field strength and magnetic latitude of a point at HEQ coordinates [zlat, zlon] on the source surface at time timem.

Input arguments:
- timem – Time (from simulation start time) of calculated source surface magnetic field (hours).
- zlat, zlong – HEQ latitude and longitude on the source surface (degrees).

Output arguments:
- bmag – Value of magnetic field (μT).
- smlat – Magnetic latitude (degrees).

Input globals:
- BFLD - The magnetic field magnitude on the source surface array (μT).
- BMAX, BMIN - The maximum and minimum magnetic field found on the source surface (μT).
- FLRDEC - The decay time of each flare’s speed (hours).
- FLRLAT, FLRLON - Each flare’s HEQ latitude and longitude (degrees).
- FLRSPD - Each flare’s maximum velocity (km/sec).
- FLRSTD - The standard deviations of each flare’s width (degrees).
- FLTIME - Each flare’s start time (hours from simulation start time).
- NOFLR - Number of flare records ingested.

Calls:
- asindeg
- interp3
- sindeg

Called by:
- sbt
- srt

Comments:
• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.

• Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.

• Leaves output arguments unchanged if unable to calculate them (e.g., a latitude or longitude input is out of range).

5.1.37 SUBROUTINE srcvel(zlat, zlong, vmag)

The srcvel subroutine interpolates the source surface solar wind speed of a point at HEQ coordinates [zlat, zlon] on the source surface at time timem.

Input arguments:
   zlat, zlong – HEQ latitude and longitude on the source surface (degrees).

Output argument:
   vmag – The solar wind speed (km/hr).

Input global:
   VFLD – The solar wind speed on the source surface array (km/hr).

Calls:
   interp3
   sindeg

Called by:
   srt

Comments:

• Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

• Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.

• Unlike srcfld, srcvel does not include flare perturbations in its source surface calculation.
• Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.

• Leaves output arguments unchanged if unable to calculate them (e.g., a latitude or longitude input is out of range).

5.1.38 SUBROUTINE srf(irot)

The srf subroutine reads the source surface magnetic field and solar wind speed maps from the B field map file and velocity map file, respectively.

Input arguments:
- irot – Carrington rotation number of HEQ zero longitude at start time.

Input files:
- bfield.dat – The B field map file containing the SEC-type magnetic field map at the source surface. The B field map file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.
  - ICRNUM – The Carrington rotation number covered by the map file.
  - ICRLON – Carrington longitude corresponding to zero HEQ longitude.
  - ILON – Carrington longitude of each record of map data (degrees).
  - BTEMP2(IMAX2,KMAX) – The magnetic field on the source surface (μT).
- velfield.dat – The velocity map file containing the solar wind speed map at the source surface. The velocity map file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.
  - ICRNUM – The Carrington rotation number covered by the map file.
  - ICRLON – Carrington longitude corresponding to zero HEQ longitude.
  - ILON – Carrington longitude of each record of map data (degrees).
  - VTEMP2(IMAX2,KMAX) – The solar wind speed at the source surface (km/sec).
- ctnnnn.dat – The B field map file containing the Wilcox-type magnetic field map at the source surface. nnnn specifies the Carrington rotation number of the map.
  - LONG – Carrington longitude of each record of map data (degrees).
  - BTEMP1(IMAX1,KMAX) – The magnetic field on the source surface (μT).

Input globals:
- SUBDR – The subdirectory name containing the input and output files.
Output globals:

- BTEMP1, BTEMP2 – The magnetic field value on the source surface array (μT), read from Wilcox type source data (BTEMP1) or SEC type source data (BTEMP2).
- VTEMP2 – The solar wind speed on the source surface array (km/sec).

Called by: ldbfld

Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
- Uses non-ANSI F77 standard DO loop format that may complicate porting to other platforms and compilers.
- The srf subroutine reads either the SEC-type data bfield.dat and velfield.dat files or a Wilcox-type data ct_?????.dat file. The Wilcox-type files are never read (unless the ISRCTYPE parameter is set to 1).

5.1.39 SUBROUTINE srt(ztime, zlati, zlong, r)

The srt subroutine calculates the radial distances of particles on a radial line from the sun along [zlati, zlong]. It is modified from Hakamada’s FPUEQ Solar wind code to allow calculations along a radial line for any latitude and longitude.

Input arguments:

- ztime – Time (from simulation start time) of calculated effects along R-T curve (hours).
- zlati, zlong – HEQ latitude and longitude of the observation point (degrees).

Output arguments:

- r – R-T array of the radial distances of particles (AU).

Input globals:

- FLRDEC – The decay time of each flare’s speed (hours).
- FLRLAT, FLRLON – Each flare’s HEQ latitude and longitude (degrees).
- FLRSPD – Each flare’s maximum velocity (km/sec).
- FLRSTD – The standard deviations of each flare’s width (degrees).
- FLTIME – Each flare’s start time (hours from simulation start time).
- IRDIAG – Controls output of R-T curves in diagnostic mode.
- JEND – Number of particles processed in each R-T curve.
- NOFLR – Number of flare records ingested.
- POWERF – Each flare’s deceleration constant (hours).
- VCO – A weighted solar wind speed derived from the minimum and maximum wind speeds at the source surface (km/hr).
Output globals:
- **HFLD** – R-T curve particle’s originating (source surface) magnetic field value (nT).
- **HLAT** – R-T curve particle’s originating (source surface) magnetic latitude (degrees).
- **HLON** – R-T curve particle’s originating (source surface) longitude (degrees).
- **VSO** – Speed of the background solar wind particles at each point on an R-T curve (km/hr).
- **VSW** – Average speed of an R-T curve particle from the source surface (km/sec).

Calls:
- cosdeg
- sindeg
- srcfld
- srcvel

Called by:
- sbt
- solwind

Comments:
- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
- Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.
- VFL is a global array but it is only used locally.
- Flare wind computation (VFL) appears to be lagged by one UTIME period.

**5.1.40 SUBROUTINE ssindex(pressu, ssi, jt, sat)**

The ssindex subroutine calculates the Shock Strength Index (SSI) and Shock Arrival Time (SAT).

Input arguments:
- pressu – dynamic pressure of the solar wind (nPa).
- jt – number of consecutive, hourly observations in pressu.

Output arguments:
- ssi – shock strength index.
- sat – shock arrival time, relative to solwind’s output start time, ihrs (hours).

Called by: solwind
Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- No protection from jt being set to more than JEX sample times and causing indexing outside array boundaries.

- Dimensioning of pressu and ssi arrays set with magic number rather than with JEX parameter.

- sat output may never be set if dynamic pressure monotonically decreases (no shocks).

- Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.

- The condition “SAT .LT. -0.5” is never true unless sat is set external to ssindex. This prevents sat from ever being set to the default 9999.

5.1.41 The step1.f, step2.f, step3.f, and step5.f Include Files.

The step1.f, step2.f, step3.f and step5.f files contain lists of INCLUDE statements that effectively concatenate all of the source code that must be compiled to build the report, source1, solwind, and ecplot programs, respectively. While these unorthodox constructs simplify the commands to build the HAF programs, they apparently hinder the operation of the programs in debug mode, probably because dbx has no knowledge of the source code file names.

5.1.42 SUBROUTINE stoc(r, theta, phi, x, y, z)

The stoc subroutine converts a position in latitude and longitude to Cartesian coordinates. X-axis is defined as phi=0.0 and X-Y plane as theta=0.0

Input arguments:
- r - radial distance from origin.
- theta - latitude North (degrees).
- phi - longitude East (degrees).

Output arguments:
- x, y, z - Cartesian coordinates of location, in same units as r.

Calls:
- cosdeg
- sindeg
Called by: sbt

Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data
typing and protect against using mistyped variables and
parameters.

- Computations are performed in single precision unless
compiler options are used to promote REAL type variables
and arguments to 8-byte floating point.

- No check for valid inputs.

5.1.43 SUBROUTINE sun(iyr, iday, secs, gst, slong, srasn, sdec)

The sun subroutine calculates the sidereal time and position of
the sun. It is good for years 1901 through 2099 with an accuracy
of 0.006 degrees.

Input arguments:
  iyr – 4-digit year.
  iday – day of year.
  secs – fractional seconds of the day.

Output arguments:
  gst – Greenwich mean sidereal time (degrees).
  slong – The Sun’s longitude along the ecliptic (degrees).
  srasn, sdec – The Sun’s apparent Right Ascension and
  Declination (degrees).

Calls:
  cotan

Called by:
  sbt
  weshere

Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data
typing and protect against using mistyped variables and
parameters.

- High-precision constants are not identified as double
precision.

- Trigonometric computation of SRASN and SDEC can be
simplified.

- Uses private copies of the degree to radian conversion
  factor and other parameters instead of shared parameters in
  the params.f file.
5.1.44 SUBROUTINE sunloc(zjd, eclon, dearth)


Input argument:
- zjd – julian date.

Output argument:
- elon – The Sun’s ecliptic longitude as viewed from the Earth (degrees).
- dearth – Distance from the Sun to the Earth (AU).

Calls:
- r2

Called by: helio

Comments:
- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
- Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.
- Uses non-ANSI F77 standard DO WHILE loop format that may complicate porting to other platforms and compilers.
- Uses private copies of PI and other parameters instead of shared parameters in the params.f file.

5.1.45 SUBROUTINE svrt(rpoint, r1, r2, r3, swspeed)

The svrt subroutine calculates the solar wind speed at the observation point along the r2 R-T curve by using three successive R-T curves at the time increment of NDTI*UTIME apart.

Input arguments:
- rpoint – Radial distance of observation point (AU).
- r1, r2, r3 – R-T curves containing the particle array along a radial at three consecutive time steps (AU).

Output argument:
- swspeed – Calculated solar wind speed at rpoint (km/sec).

Input globals:
- JEND – Number of particles processed in each R-T curve.
- VSW – Average speed of an R-T curve particle from the source surface (km/sec).

Called by: solwind
Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- VEL is a global array but it is only used locally.

5.1.46 SUBROUTINE tria2(pt1, pt2, pt3, area, anorm)

The tria2 subroutine calculates the area in the part of the plane enclosed by three Cartesian points. Uses half area of trapezoid by cross product.

Input arguments:
  pt1, pt2, pt3 – the three Cartesian points.

Output arguments:
  area – area enclosed by the three Cartesian points.
  anorm – unit vector normal to the area plane.

Called by: sbt

Calls:
  cross
  uvect
  vect

Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.

5.1.47 SUBROUTINE uvect(dx, ux, anrm)

The uvect subroutine normalizes a Cartesian vector.

Input arguments:
  dx – the vector to extract the unit vector from.

Output argument:
  ux – the unit vector parallel to dx.
  anrm – magnitude of dx.

Called by:
  geitogsm
  gseqtogei
  sbt
  tria2
Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
- No protection from overflow caused by roundoff when processing extremely small vectors.
- Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.

5.1.48 SUBROUTINE vect(x1, x2, dx)

The vect subroutine returns the difference of two vectors:

\[ dx = x2 - x1. \]

Input arguments:
- x1, x2 – the vectors to difference.

Output argument:
- dx – the difference vector.

Called by:
- sbt
- tria2

Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
- Computations are performed in single precision unless compiler options are used to promote REAL type variables and arguments to 8-byte floating point.

5.1.49 SUBROUTINE vmaxmin(imax, jmx)

The vmaxmin subroutine searches the vfld array for the maximum and minimum velocity field values.

Input arguments:
- imax, jmax – maximum array indices of the vfld array to search to.

Input global:
- VFLD – The magnetic field magnitude on the source surface array (km/sec).

Output globals:
- VMAX, VMIN – The maximum and minimum magnetic field found on the source surface (km/sec).

Called by: source1
Comments:

- The subroutine’s input arguments are not checked for valid range.
- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.
- Uses non-ANSI F77 standard DO loop format that may complicate porting to other platforms and compilers.
- The algorithm assumes $b_{min} \leq 10^7$.

5.1.50 SUBROUTINE weshere(iyr, mo, id, ih, zlateq, zloneq, sunr, irot, zl0)

The weshere subroutine locates the Earth in Heliocentric Equatorial (HEQ) coordinates. Since the Earth moves, the routine needs the time (UT) to determine the HEQ latitude and longitude at that time. Also returned are the Carrington rotation number and Carrington longitude of the HEQ zero longitude line.


Input arguments:
- iyr – 4-digit year.
- mo – month (1 – 12).
- id – day of month.
- ih – hour of the day (UT).

Output arguments:
- zlateq, zloneq – HEQ latitude and longitude (degrees).
- sunr – Distance from the sun to the Earth (AU).
- irot – The Carrington rotation number of the HEQ zero longitude.
- zl0 – The Carrington longitude corresponding to HEQ zero longitude (degrees).

Calls:
- carrot
- hecheq
- helio
- iday
- sun
- ymdjd

Called by:
- ecplot
- report
- solwind
Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- Uses non-ANSI F77 standard (inline) comments that may complicate porting to other platforms and compilers.

- No valid range check of input arguments.

5.1.51 SUBROUTINE ymdjd (iyr, month, day, jd)


Input arguments:
- iyr - Year.
- mo - Month.
- day - fractional day of the month.

Output argument:
- jd - Julian date.

Called by:
- ecplot
- getutc
- report
- solwind
- weshere

Comments:

- Missing ‘IMPLICIT NONE’ statement to promote explicit data typing and protect against using mistyped variables and parameters.

- No valid range check of input arguments.

- Arguments and most variables are not explicitly defined and floating point variables default to type REAL. Most computations are performed in single precision unless compiler options are used to promote REAL type variables and parameters to 8-byte floating point. The variables a, b, c, and dd would have greater precision as 4-byte integers than as 4-byte reals.
5.2 Executive Script Objects

5.2.1 runhaf [-o] [-t “YYYYMMDD [HHMM]”]

The runhaf shell script sequences the running the HAF programs, including the retrieval of the current B field map and velocity map files.

Options:
- f - retrieve SEC source surface files.
- o - prevents retrieval of source surface maps from UAF.
- s - run the Data Preparation Process (HAF step0).
- t - “YYYYMMDD HHMM” - specify simulation start time.

Calls:
- gse-get.pl
- step1.out
- step2.out
- step3.out
- step5.out
5.2.2 **gse-get.pl [-f fl] [-l lf] [-v]**

The gse-get perl script retrieves the specified files from the University of Alaska’s Geophysical Institute’s ‘Geoeffectiveness of Solar Events’ web page (gse.gi.alaska.edu). HAF uses this routine to retrieve the current B field map and velocity map files of the source surface. gse-get.pl calls ‘get’ in the LWP perl library.

**Options:**

- **-f fl** – Identifies the file that contains the list of files to retrieve from the ‘Geoeffectiveness of Solar Events’ web page. The default file is gse-get.config.

- **-l lf** – Write file retrieval report records to the lf log file.

- **-v** – Verbose mode. Reports each processing step to standard error.

**Input files:**

- **haf.par** – The control parameter file containing output control parameters. The control parameter file format is specified in the *Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document*.

- **SUBDR** – The name of the subdirectory to place the retrieved files in. The HAF routines will access files in the ./data/$SUBDR subdirectory.

- **gse-get.config** – Contains the list of files to retrieve from the ‘Geoeffectiveness of Solar Events’ web page (e.g., bfield.dat and velfield.dat).

**Called by:** runhaf

**Returns:** nonzero if it cannot access its input files, write the retrieved files to the ./data/$SUBDR subdirectory, or write to the logfile (if requested). Returns 0 otherwise.

**Comment:**

- The command line option facility is not robust. There is no protection from mal-formed command lines such as missing option arguments. Invalid options are ignored. Command line parser should be replaced with call to getopt.
5.2.3 fusionshell.dat

The fusionshell shell script controls retrieval and preparation of the specified files from the Space Environment Center’s ftp site (ftp://fusion.sec.noaa.gov/dist/ss_maps/). HAF uses this routine to call the perl scripts fusion-get.pl and fusion-sort.pl and to copy a text file containing a list of retrieved files to the appropriate directory.

Output file:

sourcefilelist.txt – The SEC Map File List containing the name(s) of the SEC source surface map file(s). The format of the SEC Map File List file is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

Called by: runhaf

Comments:

- The variable ‘obs’ contains the name of the observatory from which the SEC data originated and must be appropriately set in this file. Possible values are ‘mwo’, ‘nso’, and ‘wmo’.

- This script does not check for the existence of the directory to which it copies the text file containing the list of retrieved files.

5.2.4 fusion-get.pl

The fusion-get perl script retrieves the most recent solar data file(s) from the Space Environment Center’s ftp site (ftp://fusion.sec.noaa.gov/dist/ss_maps/). Specifically, it retrieves data newer than 5 days old. HAF uses this routine to retrieve the current B field map and velocity map files of the source surface. fusion-get.pl calls ‘get’ in the NET perl library. The structure of the directories to which the retrieved files are written mimics the directory structure of the SEC ftp site. If the directories do not exist, fusion-get.pl creates them.

Output files:

hafpid.log – Log file containing the process ID, program name, and date and time at which the program was run.

fusion-log.txt – Log file containing the name of the downloaded file(s) and the date and time of download.

Called by: fusionshell.dat

Returns: nonzero if it cannot write the retrieved files to the ./data/sec subdirectory, open the ftp connection, change
remote directory, or write to the log files. Returns 0 otherwise.

Comments:

- The script will retrieve files newer than the number of days set by the ‘daysold’ variable. If ‘daysold’ is set to 0, the script will retrieve all files. The current value of ‘daysold’ is 5.

- This file differs from gse-get.pl in that it retrieves the pre-processed source surface data files directly from SEC, whereas the files that gse-get.pl retrieves have already been processed and are ready for ingestion by HAF. That processing is done using HAF step0 (Data Manipulation Process).

- This script appends to the log files hafpid.log and fusion-log.txt rather than overwriting them. The log files may grow unchecked.

- This script retrieves data for the observatories specified by the ‘@dirs’ variable, currently set to retrieve WSO and MWO data. The script retrieves both MWO Potential Field Source Surface model data and MWO Source Surface Current Sheet model data (into the mwo/cs directory).

5.2.5 fusion-sort.pl

The fusion-sort perl script organizes all of the files in the data/sec/[obs] directory, where [obs] is the name of the observatory from which the data originated, into a numerically descending list.

Output files:

- hafpid.log – Log file containing the process ID, program name, and date and time at which the program was run.
- [obs]filelist.txt – The ordered list of SEC source surface map files contained in the directory data/sec/[obs].
- ORDERED[x] – The name of the ordered source surface file.
- [obs]latestfile.txt – The most recent SEC source surface map file in the directory data/sec/[obs].

Called by: fusionshell.dat

Returns: nonzero if it cannot open the /data/sec/[obs] subdirectory or write to its output files. Returns 0 otherwise.

Comment:
• This script processes data from the observatories specified by the variable ‘@dirs’, currently set to process only MWO data.

• This script can also sort the data in reverse order by appropriately setting the ‘direction’ variable.
5.2.6 startdate.pl

The startdate perl script is a utility used to print the date at 00:00 hours a specified number of days before or after the current date in the format required by the startime.par HAF file. Output may be piped directly to startime.par.

Options:
- `h` - prints usage and examples

Input argument:
- `n` - an integer specifying the number of days before or after the current date. Use a negative integer for days before the current date.

Returns: The date n days ago, in the format YYYYMMDD HHMM where HHMM is 0000.

Comment:
- Where available, the UNIX `date` function may also be used. For example, to print the date 3 days ago, execute the following at the command line:
  
  ```bash
date --date '3 days ago'
  ```

5.3  Visualization Objects

5.3.1 aceshell.dat

The aceshell shell script runs ace-get.pl, usually executed as a cron job.

Output files:

- ace-log.txt – Output listing of all downloaded files. Ace-log.txt contains a subset of the information contained in ace-run-log.txt.

- ace-run-log.txt – Verbose output of call to ace-get.pl. This file is empty if the ace-get.pl variable named ‘verbose’ is set to 0. Else, it contains ftp host name and remote directory name, remote directory contents, retrieved files, local directory to which files are written, and file transfer mode.

5.3.2 ace-get.pl

The ace-get perl script retrieves ACE SWEPAM and magnetometer instrument data files from the Space Environment Center’s ftp site (ftp.sec.noaa.gov/pub/lists/ace2). Ace-get.pl calls ‘get’ in the NET perl library. The HAF plotting software (programs vdb.s.pro and vdp.pro) uses this data to compare solar wind parameters as measured by ACE with those predicted by HAF.

Output files:

- hafpid.log – Log file containing the process ID, program name, and date and time at which the program was run.

- ace-log.txt – Log file containing the name of the downloaded file(s) and the date and time of download.

Called by: aceshell.dat

Returns: nonzero if it cannot write the retrieved files to the ./data/ace subdirectory, open the ftp connection, change remote directory, or write to the log files. Returns 0 otherwise.

Comments:

- This script appends to the log files hafpid.log and ace-log.txt rather than overwriting them. The log files may grow unchecked.

- This script appends to the log files hafpid.log and ace-log.txt rather than overwriting them. The log files may grow unchecked.
• If the ‘verbose’ variable is set to 1, verbose output is sent to stdout.

5.3.3 acebuild.pl

The acebuild perl script builds recent ACE files for ingestion by the plotting routines v dbs.pro and vdp.pro. Acebuild uses startime.par to determine the simulation start time, and haf.par to determine the simulation duration. The script uses ACE magnetometer and SWEPAM instrument data.

Input files:

startime.par – The start time file containing the simulation start time. The start time file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.
SYEAR,SMONTH,SDAY – Start date of simulation.

haf.par - The control parameter file containing output control parameters. The control parameter file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.
IHRS - start hour of output (on start date)
NHOURS - number of hours of output
SUBDR - The name of the subdirectory to place the processed files in.

[YYYYMM]_ace_swepam_1h.txt – Data file containing ACE SWEPAM instrument data covering the year and month specified in the file name.

[YYYYMM]_ace_mag_1h.txt – Data file containing ACE magnetometer data covering the year and month specified in the file name.

Output files:

hafpid.log – Log file containing the process ID, program name, and date and time at which the program was run.

[SUBDR]_ace_swepam_1h.txt – Data file containing ACE SWEPAM instrument data covering the simulation time range.

[SUBDR]_ace_mag_1h.txt – Data file containing ACE magnetometer data covering the simulation time range.

Returns: nonzero if it cannot open the ACE data files, haf.par, or startime.par. Returns 0 otherwise.

Comment:

• This script appends to the log file hafpid.log rather than overwriting it. The log file may grow unchecked.

• Prints verbose output to stdout.
5.3.4 ecimf_c.pro

The ecimf_c.pro IDL program plots line segments to create ecliptic plane plots of the IMF out to 2 AU.

Input files:

haf.par - The control parameter file containing output control parameters. The control parameter file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

AU - Radial distance along the sun-earth line
SUBDR - The subdirectory name containing the input and output files.
JST - The hour (relative to start time) of the first ECPLOT generated
JEND - The hour of the last plot generated
JINC - The hour increment between JTS and JTE

eplotnnn.dat - The ecliptic plot files for each sample time. The ecliptic plot file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

AU - The radial distance of the plot out from the sun (AU).
IY, IM, ID, IH - Plot sample time.
EX, EY - HEQ coordinates of the Earth in the plot (AU).
X0, Y0, X1, Y1 - Coordinates of line segment endpoints depicting the IMF pattern (AU).
H1 - Line segment’s originating (source surface) magnetic field value (nT).

Output files:

ecimf_nnn.gif - The ecliptic plane IMF plot.

5.3.5 mapsrsrc.pro

The mapsrsrc.pro IDL program generates contour plots of SEC B field and velocity data.

Input files:

haf.par - The control parameter file containing output control parameters. The control parameter file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

SUBDR - The subdirectory name containing the input and output files.

earth.par - The Earth Location file for the specified start time. The Earth Location file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.
L0 – Carrington longitude corresponding to HEQ zero longitude (degrees).
ELAT, ELON – Earth’s HEQ Latitude and Longitude (degrees).

flare.par – The flare parameter file containing solar flare parameter records. The flare parameter file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.
NOFLR – The number of flare records.
FLON(NOFLR) – Flare’s HEQ longitude (degrees).
FLAT(NOFLR) – Flare’s HEQ latitude (degrees).

NUM – Event number.

b_001.dat – File containing a subset bfield.dat, written by program sourcesec.
IROT – The Carrington rotation number covered by the map file.
ILON – Carrington longitude corresponding to zero HEQ longitude.
LON – Carrington longitude of each record of map data (degrees).
IN1 – The magnetic field on the source surface (\textmu T).

vel_001.dat – File containing a subset of velfield.dat, written by program sourcesec.
IROT – The Carrington rotation number covered by the map file.
ILON – Carrington longitude corresponding to zero HEQ longitude.
LON – Carrington longitude of each record of map data (degrees).
IN1 – The velocity field on the source surface.

Output files:
b_001.gif – The source surface magnetic field contour plot.
vel_001.gif – The source surface velocity field contour plot.

5.3.6 vdb.s.pro

The vdb.s.pro IDL program plots solar wind speed, density, IMF magnitude, theta and phi angles as a function of time.

Input files:
haf.par – The control parameter file containing output control parameters. The control parameter file format is specified in the Hakamada-Akasofu-Fry (HAF)
Kinematic Solar Wind Model Functional Requirements Document.

SHR - Start hour of output, relative to the simulation start time (hours).

NHR - Number of hours from SHR to output at an hourly sampling rate.

SUBDR - The subdirectory name containing the input and output files.

earth_0.dat - The simulated field and velocity file contains an IMF forecast record for the L1 observation point at each sample time. The simulated field and velocity file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

NP - Number of hours from SHR to output at an hourly sampling rate.

DAY - Start day of the simulation.

A - The hour, from the simulation start time, of the forecast record (hours).

B - The solar wind speed (km/sec).

C - Particle density (cm$^{-3}$).

earth_q.dat - The simulated field and velocity file contains an IMF forecast record for the L1 observation point at each sample time for quiet (no flare) conditions. The simulated field and velocity file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

NPQ - Number of hours from SHR to output at an hourly sampling rate.

DAYQ - Start day of the simulation.

IYRQ, MOQ, DAYQ, HRQ - Start time of the simulation.

A - The hour, from the simulation start time, of the forecast record (hours).

B - The solar wind speed (km/sec).

C - Particle density (cm$^{-3}$).

ssi_0.dat - The shock file contains the forecast shock arrival time at the L1 observation point and the dynamic pressure and shock strength index at each sample time. The shock file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.

STOA0 - The forecast shock arrival time relative to the simulation start time (hours).

SDATE - The year, month, and day of the forecast shock.

SHR - The hour of the forecast shock.

A - The forecast dynamic pressure at the L1 observation point at each simulation sample time (nPa).

B - Shock strength index at the L1 observation point at each simulation sample time.

[SUBDR]_ace_sweepam_1h.txt - Data file containing ACE SWEPAM instrument data covering the simulation time range.
Output file:
  vdfs_[SUBDR].ps – The velocity, density, IMF plot.

5.3.7 vdp.pro

The vdp.pro IDL program plots solar wind speed, density, dynamic pressure, and shock searching index (SSI) as a function of time. The SSI is a running calculation of $\log(DP/ P_{min})$ at each time step in the temporal profile of the data or model output.

Input files:
  haf.par - The control parameter file containing output control parameters. The control parameter file format is specified in the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model Functional Requirements Document.
  SHR - Start hour of output, relative to the simulation start time (hours).
  NHR - Number of hours from SHR to output at an hourly sampling rate.
  SUBDR - The subdirectory name containing the input and output files.

Output file:
  vdp_[SUBDR].ps – The velocity, density, pressure plot.

6 External Interfaces

6.1 Data

The following files must be available to run the HAF code:
  gse-get.config, haf.par, bfield.dat, velfield.dat, and report.dat. The report.dat flare report file will typically be updated with the most recent observatory data prior to each run. HAF automatically retrieves the current bfield.dat and velfield.dat files from UAF.

When the HAF code programs are executed, they create a start time (startime.par) file, a flare parameter (flare.par) file, an Earth location (earth.par) file, a source field (sourceb.par) file, a source velocity (sourcev.par) file, a simulated field and velocity (earth_0.dat) file, a shock (ssi_0.dat) file, and ecliptic plot (ecplot_nnn.dat) files. Processing progress is reported to standard output. The latter three files are the actual outputs from the HAF code, the former files conveying data between the separate programs. Each run of the HAF code normally consumes less than 1M of disk space.

6.2 Operating system services

The scripts and routines of the HAF code execute under the Sun Solaris 7 operating system on a Sun Ultra 60 Model 2360 computer.
The software was developed for use on a Sparc Solaris platform. For installation and operation of the software, access to a Sparc platform with a minimum of 10M of disk space is required.

6.3 Outputs

When runhaf is executed, it creates a simulated field and velocity file, a shock file and ecliptic plot files which are stored in the specified subdirectory of the ./data directory.

6.4 Other software programs or libraries

6.4.1 Software

The software was developed under Sun Solaris 7. An ANSI-standard FORTRAN 77 compiler should be present on the target system.
### Appendix A. ACROYNMS AND ABBREVIATIONS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACE</td>
<td>Advanced Composition Explorer</td>
</tr>
<tr>
<td>AFCCC</td>
<td>Air Force Combat Climatology Center</td>
</tr>
<tr>
<td>AFOSR</td>
<td>Air Force Office of Scientific Research</td>
</tr>
<tr>
<td>AFRL</td>
<td>Air Force Research Laboratory</td>
</tr>
<tr>
<td>AFSCN</td>
<td>Air Force Satellite Control Network</td>
</tr>
<tr>
<td>AFSPACECOM</td>
<td>Air Force Space Command</td>
</tr>
<tr>
<td>AFSWC</td>
<td>Air Force Space Weather Center</td>
</tr>
<tr>
<td>AFWA</td>
<td>Air Force Weather Agency</td>
</tr>
<tr>
<td>AFWIN</td>
<td>Air Force Weather Information Network</td>
</tr>
<tr>
<td>AF/XOW</td>
<td>Air Force Director of Weather</td>
</tr>
<tr>
<td>APL</td>
<td>Applied Physics Laboratory of Johns Hopkins University</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>AU</td>
<td>Astronomical Units</td>
</tr>
<tr>
<td>AVHRR</td>
<td>Advanced Very High Resolution Radiometer</td>
</tr>
<tr>
<td>AVN</td>
<td>Aviation Model</td>
</tr>
<tr>
<td>B</td>
<td>Magnetic Flux Density</td>
</tr>
<tr>
<td>CIR</td>
<td>Co-rotating Interaction Region</td>
</tr>
<tr>
<td>CME</td>
<td>Coronal Mass Ejection</td>
</tr>
<tr>
<td>COE</td>
<td>Common Operating Environment</td>
</tr>
<tr>
<td>DII</td>
<td>Defense Information Infrastructure</td>
</tr>
<tr>
<td>DMSP</td>
<td>Defense Meteorological Satellite Program</td>
</tr>
<tr>
<td>ECMWF</td>
<td>European Center for Medium-Range Weather Forecasts</td>
</tr>
<tr>
<td>EIT</td>
<td>Extreme ultraviolet Imaging Telescope on the SOHO spacecraft.</td>
</tr>
<tr>
<td>EXPI</td>
<td>Exploration Physics International, Inc.</td>
</tr>
<tr>
<td>FIP</td>
<td>Fault Isolation Procedure</td>
</tr>
<tr>
<td>FNMOC</td>
<td>Fleet Numerical Meteorology and Oceanography Center</td>
</tr>
<tr>
<td>FSL</td>
<td>Forecast Systems Laboratory</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GI</td>
<td>Geophysical Institute</td>
</tr>
<tr>
<td>GIF</td>
<td>Graphic Interchange Format</td>
</tr>
<tr>
<td>GMT</td>
<td>Generic Mapping Tools</td>
</tr>
<tr>
<td>GOES</td>
<td>Geostationary Operational Environmental Satellite</td>
</tr>
<tr>
<td>GOLD</td>
<td>Geophysical On-Line Data</td>
</tr>
<tr>
<td>GRIB</td>
<td>Gridded Binary</td>
</tr>
<tr>
<td>GSEQ</td>
<td>Geocentric Solar Equatorial Coordinates</td>
</tr>
<tr>
<td>GSM</td>
<td>Geocentric Solar Magnetic Coordinates</td>
</tr>
<tr>
<td>HAF</td>
<td>Hakamada-Akasofu-Fry</td>
</tr>
<tr>
<td>HEC</td>
<td>Heliocentric Ecliptic Coordinates</td>
</tr>
<tr>
<td>HEQ</td>
<td>Heliocentric Equatorial Coordinates</td>
</tr>
<tr>
<td>ICME</td>
<td>Interplanetary Coronal Mass Ejection</td>
</tr>
<tr>
<td>IDL</td>
<td>Interactive Data Language</td>
</tr>
<tr>
<td>IGRF</td>
<td>International Geomagnetic Reference Field</td>
</tr>
</tbody>
</table>
IMF  Interplanetary Magnetic Field
JHU  Johns Hopkins University
L1  Sun-Earth libration point
LAN  Local Area Network
LAPS  Local Analysis and Prediction System
LASCO  Large Angle and Spectrometric Coronagraph experiment on the SOHO spacecraft.
MHD  Magnetohydrodynamics
MM5  Fifth Generation Mesoscale Model
MWO  Mount Wilson Solar Observatory
NCAR  National Center for Atmospheric Research
NCEP  National Centers for Environmental Prediction
netCDF  Network Common Data Form
NGDC  National Geophysical Data Center
NGM  Nested Grid Forecast Model
NOAA  National Oceanic and Atmospheric Administration
NOGAPS  Navy Operational Global Atmospheric Prediction System
nPa  Nano-Pascal
NSO  National Solar Observatory
nT  Nano-Tesla
NWP  Numerical Weather Prediction
OWS  Operational Weather Squadron
PACE  Polar Anglo-American Conjugate Experiment
PBL  Planetary Boundary Layer
PCA  Polar Cap Absorption
PFRR  Poker Flat Research Range
PFSS  Potential Field Source Surface
POS  Plane-of-sky
RPC  NOAA/SEC Rapid Prototyping Center
SABER  Sounding of the Atmosphere using Broadband Emission Radiometry
SAT  Shock Arrival Time
SD  Space Department of the Applied Physics Laboratory
SDP  Software Development Plan
SEC  Space Environment Center
SEE  Solar EUV Experiment
SEON  Solar Electro-optical Observing Network
SFIR  Swept Frequency Interferometric Radiometer
SFOC  Spaceflight Operations Center
SOON  Solar Observing Optical Network
SOHO  Solar and Heliospheric Observatory
SRBL  Solar Radio Burst Locator
SRS  Solar Radio Spectrograph
SSCS  Source Surface Current Sheet
SSI  Shock Searching Index
STP  Solar Terrestrial Physics
SWEPAM  Solar Wind Electron, Proton, and Alpha Monitor
SWOC  Space Weather Operations Center (Offutt)
SWXS  Space Weather Squadron
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tcl</td>
<td>Tool Command Language</td>
</tr>
<tr>
<td>Tk</td>
<td>Toolkit</td>
</tr>
<tr>
<td>Tix</td>
<td>Tk Interface Extension</td>
</tr>
<tr>
<td>UAF</td>
<td>University of Alaska, Fairbanks</td>
</tr>
<tr>
<td>UCAR</td>
<td>University Corporation for Atmospheric Research</td>
</tr>
<tr>
<td>UPOS</td>
<td>University Partnering for Operational Support</td>
</tr>
<tr>
<td>WDC</td>
<td>World Data Center</td>
</tr>
<tr>
<td>WF</td>
<td>Weather Flight</td>
</tr>
<tr>
<td>WMO</td>
<td>World Meteorological Organization</td>
</tr>
<tr>
<td>WSO</td>
<td>Wilcox Solar Observatory</td>
</tr>
<tr>
<td>XDR</td>
<td>External Data Representation</td>
</tr>
</tbody>
</table>