Documentation for the Sudden Stratospheric Warming Compendium data set

Production Guide

Revision 1.0
2016/07/06

Amy H. Butler and Jeremiah P. Sjoberg
sswcompendium@gmail.com
NOAA/CIRES ESRL Chemical Sciences Division

Table of Contents

Purpose

This document provides a basic production guide to the tools and code for generating the Sudden Stratospheric Warming Compendium (SSWC). Covered topics include an overview of the production algorithm, system setup, user inputs, and coding workflow.

Reading this document

This document contains formatting to enable ease of reading.

Directory trees or commands are presented as indented paragraphs with light gray shading. An example is

```
Primary folder
 |----Subfolder
         |----Nested folder 1
         |----Nested folder 2
 |----Second subfolder
         |----Nested folder 3
```

where the path to 'Nested folder 3' would be './Primary folder/Second subfolder/Nested folder 3.'

An example command is

```
IDL> t = FLTARR(45)
```

References to procedures, both with and without the '.pro' suffix, and to code variables will be presented as inline text with light gray shading. For example, the above command would be written inline as FLTARR(45).

We will use object-oriented referencing styles for structures and their tags. For example, accessing the 'dir' tag from the 'raInStr' structure is written here as raInStr.dir for which the equivalent command to access this tag in IDL is

```
IDL> raInStr.dir
```

Production algorithm

An informal description of the SSWC production algorithm follows. This is intended to give an end-user a basic knowledge of how the SSWC is generated from the input data.

Note: more detailed API guide is available for using the Compendium code itself.

Input data spanning the entire desired record are firstly accumulated. The production user then provides overview details of these data, such as time span, directory, and generalized file names, to a userInput script. Details of the individual variables are provided at this time as well. Production users also define an event list giving the dates used for producing the event-based data.

The input data are then read by the Compendium and stored as the field variable data described in "Output data description." In this process, the input data are interpolated to the desired horizontal grid and chunked into yearly segments spanning July 01 through June 30 of the next year. These yearly chunks of data are stored in CF-compliant netCDF-4 files to increase accessibility to production users. This portion of the algorithm also requires some side computations, such as calculation of wave fluxes and interpolation of select fields to isentropic surfaces.

Climatological statistics are then calculated from these field variable data. Any field variables that rely on climatological statistics, such as annular mode indices, are then computed and stored alongside the other field variable data.

The event-based data are then compiled using combinations of the field variable data and the climatological statistics. Climate indices are incorporated here as well. Dates of these events are defined by the event list passed by the production user to the code. Together with the climatological statistics, this constitutes the primary output of the SSWC.

System setup

Production of the SSWC requires a licensed Interactive Data Language (IDL) installation accessible by the user. The IDL version is expected to be 8.5 or later. NetCDF library version 4.3.3.1 or later must be installed.

Minimum disk storage space and virtual memory capacity for running the SSWC are 300 GB and 32 GB, respectively. However, we recommended storage space of 1 TB and virtual memory capacity of 64 GB.

User inputs

Directory dependencies

The provided directory tree of the SSWC is as follows

```
SSWC
 |----v#.#
        |----procedures
        |----runCode
        |----userInput
```

It is expected that these directories and the contained files remain as-is. The exception to this is the user is required to alter input procedure scripts within the ‘userInput’ directory. This is discussed in “Preliminary scripting.” Installation of the SSWC code simply requires placing the directory tree in a desired location.

Users must determine directories for intermediate and output files. It is recommended that users create separate directories for intermediate files, for climatological files, and for event-based files. It is further recommended that users create these three directories for each input dataset. As an example, for MERRA-2 data, one may create the directory structure

```
/Path/to/data/files
            |----MERRA2
                  |----SSWC_intFiles
                  |----SSWC_climFiles
                  |----SSWC_evFiles
```

where ‘SSWC_intFiles,’ ‘SSWC_climFiles,’ and ‘SSWC_evFiles’ are designated for intermediate files, climatological files, and event-based files, respectively.

Preliminary scripting

To utilize the SSWC code, users must fill in IDL structure variables with appropriate values to allow the code to find and process input data, and to instruct the code where to store output files. This is performed in the RASET_raFields_userInput.pro procedure file contained in the ‘userInput’ directory of the SSWC hierarchy. As provided, this file is intended as a template. Users should make a copy of this template and replace all instances of ‘RASET’ with a useful identifier for the input dataset. See the API documentation for more detailed documentation of how to properly modify the structures in the procedure.

Users firstly set overarching parameters for a given dataset or hereafter, ‘raSet,’ by modifying default values for the structure raInStr. Contained in this structure are the input and output directories, timespan of the dataset, and standard time adjustments. Note that the output directory setting here is an overview directory meant to simplify directory inputs in the varInStr structures discussed below.

Individual structures for each input geophysical field are then set using the predefined structure varInStr. Contained in this structure are the field’s input and output names, file name and directory, coordinate names, processing parameters, and metadata. After setting individually named structures for each input geophysical field, the user must redefine the output varInStr as an array of each of these structures.

Individual structures for each climate index are finally set using the predefined structure climInStr. Contained in this structure are the output names, file name and directory, and reading parameters. After setting individually named structures for each input geophysical field, the user must redefine the output climInStr as an array of each of these structures.

Finally, production users create an event list file in the 'userInput' directory. This file lists one event per line with the format 'YYYY/MM/DD.'

Note: A detailed, working example file MERRA2_raFields_userInput.pro is provided with the distribution. An example event list is also included.

Input data

It is expected that the users have input data stored in a directory tree with consistent file naming. In particular, the input files are expected to have year identifiers that enable systematic identification of all data contained between two annual time bounds. This allows the SSWC code to correctly identify input data files and be able to sequentially process the data into intermediate files.

As an example, MERRA-2 data files are provided in a directory tree as follows

```
MERRA2
 |----${DATASET}
        |----${YYYY}
               |----${MM}
```

with standard files name following

```
MERRA2_${STREAM}.${MODIFIER}.${YYYY}${MM}${DD}.nc4  .
```

Here, ${DATASET}, ${STREAM}, and ${MODIFIER} are variable values which can be appropriately defined within RASET_raFields_userInput.pro so as to allow the code to find the files in the directory tree. The date variables ${YYYY}, ${MM}, and ${DD} meanwhile ensure that the code will be able to properly process annual chunks of data into intermediate files.

Details of the input data used in the currently available version of the SSWC are contained in the User's Guide Documentation.

Workflow

Once the user has installed the necessary libraries and SSWC code, created the desired directories, and fully filled out the RASET_raFields_userInput.pro procedure for a given input dataset or raSet, production may begin. All commands that follow are run from the 'runCode' directory.

Field variable data

As a preliminary step, the user runs the command

```
IDL> SSWC_raFields_readGrid, raSet, fieldString1, fieldString2
```

where raSet, fieldString1, and fieldString2 are string values. The latter two reference specific structure names from the varInStr output from RASET_raFields_userInput.pro, where fieldString1 is a pressure-level field and fieldString2 is a surface field. This command produces an IDL SAVE file containing 1) the longitude and latitude values on both the interpolated horizontal grid and the surface grid; and 2) the pressure and isentropic grid values. This SAVE file is utilized throughout the SSWC code.

The user will then process the input data for each geophysical field. To do so, the user will run the command SSWC_raFields_readRaw from the runCode directory. This is done by

```
IDL> SSWC_raFields_readRaw, raSet, fieldString
```

This command produces July 01-June 30 files for each year of processed data and places these intermediate outputs into the raInStr.fieldsDir directory.

Following this, secondary processing begins. Flux fields, those that are products of zonal variations of two fields, are calculated by

```
IDL> SSWC_raFields_calcFluxes, raSet, fieldString
```

See the provided example user input procedure for how to properly define a structure in varInStr for calculating fluxes.

Fields that are provided on isentropic surfaces are interpolated by

```
IDL> SSWC_raFields_calcTheta, raSet, fieldString
```

Total column ozone is then calculated for those input datasets that do not provide it[1]. This is done by

```
IDL> SSWC_raFields_calcTCo3, raSet, fieldString
```

where fieldString refers to the ozone mixing ratio structure from varInStr.

Climatological statistics

Production users then calculate climatological statistics.

Climatological means and standard deviations should be produced for each geophysical field. This is done by

```
IDL> SSWC_outFields_calcClim, raSet, fieldString
```

where raSet and fieldString are as before.

For those fields which distribution percentages are required, users run the command

```
IDL> SSWC_outFields_calcDist, raSet, fieldString
```

[1] This is currently a step specifically for MERRA-2.

Both commands produce output files that are stored in the raInStr.climDir and raInStr.distDir directories, respectively. The span of years used in both commands may be adjusted.

Note: the distribution percentage calculation is currently highly memory intensive for high resolution, long-term input datasets. For 57 years of JRA-55 data, this command requires roughly 25 GB of virtual memory to complete.

An additional field variable is calculated now: annular mode time series. These time series are dependent on the climatological statistics. To calculate these time series, use the two commands

```
IDL> SSWC_raFields_calcAnnMode, raSet, /NAM
IDL> SSWC_raFields_calcAnnMode, raSet, /SAM
```

where the first command calculates the Northern Annular Mode (NAM) and the second calculates the Southern Annular Mode (SAM).

Event-based data

The final step in production is to produce the event-based data. This is done in three steps.

The full field data, anomaly field data, and derived field data are extracted and stored by

```
IDL> SSWC_outFields_varFull, raSet, eventList
IDL> SSWC_outFields_varAnom, raSet, eventList
IDL> SSWC_outFields_varFull, raSet, eventList
```

where raSet is as before and eventList is a string for the event list to use from the 'userInput' directory. No directory prefix should be provided for eventList.

Example

A brief example using MERRA-2 data is provided here. This example makes use of the provided 'MERRA2_raFields_userInput.pro' and 'evList_MERRA2' files. Note that IDL output has been truncated here.

```
IDL>;; Create grid file and intermediate files.
IDL>SSWC_raFields_readGrid, 'MERRA2', 'vwnd', 'tsfcMin'
IDL>SSWC_raFields_readRaw, 'MERRA2', 'vwnd'
IDL>SSWC_raFields_readRaw, 'MERRA2', 'temp'
IDL>SSWC_raFields_readRaw, 'MERRA2', 'geop'
IDL>SSWC_raFields_readRaw, 'MERRA2', 'o3'
IDL>SSWC_raFields_readRaw, 'MERRA2', 'tsfcMin'
IDL>
IDL>;; Calculate fluxes.
IDL>SSWC_raFields_calcFluxes, 'MERRA2', 'vt'
IDL>
IDL>;; Interpolate to isentropic surfaces.
IDL>SSWC_raFields_calcTheta, ‘MERRA2’, ‘o3’
IDL>
IDL>;; Calculate total column ozone.
IDL>SSWC_raFields_calcTCo3, ‘MERRA2’, ‘o3’
IDL>
IDL>;; Produce climatologies.
IDL>SSWC_outFields_calcClim, ‘MERRA2’, ‘vwnd’
IDL>SSWC_outFields_calcClim, ‘MERRA2’, ‘temp’
IDL>SSWC_outFields_calcClim, ‘MERRA2’, ‘geop’
IDL>SSWC_outFields_calcClim, ‘MERRA2’, ‘o3’
IDL>SSWC_outFields_calcClim, ‘MERRA2’, ‘tsfcMin’
IDL>SSWC_outFields_calcClim, ‘MERRA2’, ‘vt’
IDL>
IDL>;; Produce distributions.
IDL>SSWC_outFields_calcDist, ‘MERRA2’, ‘tsfcMin’
IDL>
IDL>;; Calculate annular modes.
IDL>SSWC_raFields_calcAnnMode, ‘MERRA2’, /NAM
IDL>SSWC_raFields_calcAnnMode, ‘MERRA2’, /SAM
IDL>
IDL>;; Calculate event-based data.
IDL>SSWC_outFields_varFull, ‘MERRA2’, ‘evList_MERRA2’
IDL>SSWC_outFields_varAnom, ‘MERRA2’, ‘evList_MERRA2’
IDL>SSWC_outFields_varDerive, ‘MERRA2’, ‘evList_MERRA2’
IDL>
```

Changelog

v1.0 : First official release.