

Documentation for the Sudden Stratospheric Warming Compendium data set

API 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	4
Reading this document	4
Top-level procedures	4
SSWC_outFields_calcClim	4
SSWC_outFields_calcDist	5
SSWC_outFields_varAnom	6
SSWC_outFields_varDerive	6
SSWC_outFields_varFull	7
SSWC_raFields_calcAnnMode	8
SSWC_raFields_calcFluxes	9
SSWC_raFields_calcTCO3	9
SSWC_raFields_calcTheta	10
SSWC_raFields_readGrid	11
SSWC_raFields_readRaw	11
Supporting procedures	12
decompProduct	12
DOY	12
DOY_leapYr	13
gauss_lats	13
gridBounds_set	13
hdfFiles_read	14
ncFiles_read	15
next_leapYr	16
raFiles_find	16
raFiles_read	16
readEvDates	18
SSWC_constants_define	18
SSWC_dir_search	18
SSWC_error_mesg	19
SSWC_events_mod	19
SSWC_file_search	20
SSWC_gridData_read	20
SSWC_misc_mod	20
SSWC_ncFiles_write	21
SSWC_ncWrite_cellMethods	21
SSWC_ncWrite_define	22
SSWC_ncWrite_validAttributes	22
SSWC_outFields_readDist	23
SSWC_outFields_readFiles	23
SSWC_outFields_removeClim	24
SSWC_pro_init	24
SSWC_raFields_rawTheta	24
SSWC_read_mod	25
SSWC_readEvents	25
SSWC_runVersion	25
SSWC_setTimeJul	25

SSWC_setTimeUnits.....	26
SSWC_varAnom_init.....	26
SSWC_varDerive_init.....	26
SSWC_varFull_init.....	26
SSWC_varGrid_init.....	27
SSWC_vorticity_mod.....	27
tseriesClimIdxs.....	27
tseriesElcount.....	27
tseriesExtrema.....	28
tseriesFull.....	28
tseriesReversal.....	29
tseriesVorticityCalc.....	29
External procedures.....	30
Dobson.....	30
inv_spherical_transform.....	30
match.....	31
mod_struct.....	31
spherical_transform.....	32
undefine.....	32
weights_legendre.....	33
Changelog.....	33

Purpose

This document provides a guide to the application program interface (API) for the Sudden Stratospheric Warming Compendium (SSWC). Details on the location, purpose, and user interaction with each procedure and function are given.

Reading this document

This document contains formatting to enable ease of reading.

Commands are presented as indented paragraphs with light gray shading. 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
```

Top-level procedures

SSWC_outFields_calcClim

File	./runCode/SSWC_outFields_calcClim.pro		
Purpose	Calculates climatological means and standard deviations of a field. Output are stored in CF-compliant netCDF-4 files.		
Description	The climatological mean values at each day and point are calculated from the entire dataset, unless specified otherwise. Retaining only the first four Fourier components from a temporal FFT results in smoothed means. These smoothed climatological means are used to calculate corresponding climatological standard deviations.		
Requires	DOY, mod_struct, ncFiles_read, next_leapYr, SSWC_error_mesg, SSWC_ncWrite_cellMethods, SSWC_ncFiles_write, SSWC_runVersion, SSWC_setTimeJul, SSWC_varGrid_init, undefine		
Inputs	raSet	String	Name of the reanalysis dataset.

	raFields	String	Single element or vector name(s) of field(s). Names are defined in the user input file.
	RAYRS	Integer	(OPTIONAL, default is all years) Vector of years for which to calculate the statistics.
	INPUTFILE	String	(OPTIONAL, default is raSet+'_raFields_userInput.pro') Name of user input file.
Outputs		File	SSWC_v{##}_{outName}ClimMean_{raSet}_s{sDate}_e{eDate}_c{cDate}.nc in folder raInStr.climDir.
		File	SSWC_v{##}_{outName}ClimStds_{raSet}_s{sDate}_e{eDate}_c{cDate}.nc in folder raInStr.climDir.
Usage	IDL> SSWC_outFields_calcClim, 'MERRA2', 'vwnd'		

SSWC_outFields_calcDist

File	./runCode/SSWC_outFields_calcDist.pro		
Purpose	Calculates climatological distribution values of a field at given percentages of the distribution. Output are stored in CF-compliant netCDF-4 files.		
Description	All values at all points are firstly read into memory. The distribution values for a certain percentage are calculated following Eq. (1) of Zhang et al. (2005).		
Requires	DOY, mod_struct, ncFiles_read, next_leapyr, SSWC_error_mesg, SSWC_ncFiles_write, SSWC_ncWrite_cellMethods, SSWC_runVersion, SSWC_setTimeJul, SSWC_varGrid_init		
Inputs	raSet	String	Name of the reanalysis dataset.
	raFields	String	Single element or vector name(s) of field(s). Names are defined in the user input file.
	RAYRS	Integer	(OPTIONAL, default is all years) Vector of years for which to calculate the statistics.
	RAPCTS	Double	(OPTIONAL, default is [0.05D, 0.10D, 0.90D, 0.95D]) Single element or vector specifying the percentage values of the distribution for which to calculate.

	INPUTFILE	String	(OPTIONAL, default is raSet+'_raFields_userInput.pro') Name of user input file.
Outputs		File	SSWC_v{##}_{outName}Pct {raPcts}_{raSet}_s{sDate}_ e{eDate}_c{cDate}.nc in folder raInStr.distDir.
Usage	IDL> SSWC_outFields_calcDist, 'MERRA2', 'tsfcMin'		

SSWC_outFields_varAnom

File	./runCode/SSWC_outFields_varAnom.pro		
Purpose	Extracts and stores anomaly fields for the ± 60 days around a specified event.		
Description	Full fields and climatologies are read into memory. Anomalies are then calculated. Event dates are parsed from the input event list. Fields from years with events are read into memory. Anomalies surrounding the events are extracted and stored in a CF-compliant netCDF-4 file.		
Requires	SSWC_constants_define, SSWC_error_mesg, SSWC_ncFiles_write, SSWC_outFields_readFiles, SSWC_outFields_removeClim, SSWC_readEvents, SSWC_runVersion, SSWC_setTimeUnits, SSWC_varAnom_init, SSWC_varGrid_init, tseriesFull, undefine		
Inputs	raSet	String	Name of the reanalysis dataset.
	evName	String	Name of the input event list.
	RAYRS	Integer	(OPTIONAL, default is all years) Vector of years for which to extract event-based data.
	INPUTFILE	String	(OPTIONAL, default is raSet+'_raFields_userInput.pro') Name of user input file.
	NCMODIFIER	String	(OPTIONAL, default is '') Modifier of output file name(s).
Outputs		File	SSWC_v{##}_varAnom_ {ncModifier}{raSet}_d{dDate}_ s{sDate}_e{eDate}_c{cDate}.nc in folder raInStr.outDir.
Usage	IDL> SSWC_outFields_varAnom, 'MERRA2', 'evList_MERRA2'		

SSWC_outFields_varDerive

File	./runCode/SSWC_outFields_varDerive.pro		
Purpose	Extracts and stores derived fields for the ± 60 days around a specified event.		

Description	Full fields and climatologies are read into memory. Anomalies and derived fields are then calculated. Event dates are parsed from the input event list. Fields from years with events are read into memory. Derived fields surrounding the events are extracted and stored in a CF-compliant netCDF-4 file.		
Requires	SSWC_constants_define, SSWC_error_mesg, SSWC_ncFiles_write, SSWC_outFields_readDist, SSWC_outFields_readFiles, SSWC_readEvents, SSWC_outFields_removeClim, SSWC_runVersion, SSWC_setTimeUnits, SSWC_varDerive_init, SSWC_varGrid_init, tseriesClimIdxs, tseriesElcount, tseriesExtrema, tseriesFull, tseriesReversal, tseriesVorticity_calc, undefine		
Inputs	raSet	String	Name of the reanalysis dataset.
	evName	String	Name of the input event list.
	RAYRS	Integer	(OPTIONAL, default is all years) Vector of years for which to extract event-based data.
	INPUTFILE	String	(OPTIONAL, default is raSet+'_raFields_userInput.pro') Name of user input file.
	NCMODIFIER	String	(OPTIONAL, default is "") Modifier of output file name(s).
Outputs		File	SSWC_v{##}_varDerive_{ncModifier}{raSet}_d{dDate}_s{sDate}_e{eDate}_c{cDate}.nc in folder raInStr.outDir.
Usage	IDL> SSWC_outFields_varDerive, 'MERRA2', 'evList_MERRA2'		

SSWC_outFields_varFull

File	./runCode/SSWC_outFields_varFull.pro		
Purpose	Extracts and stores full fields for the ± 60 days around a specified event.		
Description	Full fields are read into memory. Event dates are parsed from the input event list. Fields from years with events are read into memory. Full fields surrounding the events are extracted and stored in a CF-compliant netCDF-4 file.		
Requires	SSWC_constants_define, SSWC_error_mesg, SSWC_ncFiles_write, SSWC_outFields_readFiles, SSWC_readEvents, SSWC_runVersion, SSWC_setTimeUnits, SSWC_varFull_init, SSWC_varGrid_init, tseriesFull, undefine		
Inputs	raSet	String	Name of the reanalysis dataset.
	evName	String	Name of the input event list.
	RAYRS	Integer	(OPTIONAL, default is all years) Vector of years for which to extract event-based data.

	INPUTFILE	String	(OPTIONAL, default is raSet+'_raFields_userInput.pro') Name of user input file.
	NCMODIFIER	String	(OPTIONAL, default is "") Modifier of output file name(s).
Outputs		File	SSWC_v{##}_varFull_ {ncModifier}{raSet}_d{dDate}_ s{sDate}_e{eDate}_c{cDate}.nc in folder raInStr.outDir.
Usage	IDL> SSWC_outFields_varFull, 'MERRA2', 'evList_MERRA2'		

SSWC_raFields_calcAnnMode

File	./runCode/SSWC_raFields_calcAnnMode.pro		
Purpose	Calculates the annular mode from geopotential height fields.		
Description	<p>Calculates the first principal component (PC) time series of the height-dependent, zonal-mean empirical orthogonal function (EOF). The method is described in Baldwin and Thompson (2009). Reads in all geopotential heights and calculates anomalies. After zonal averaging and detrending, the EOF's and PC's are calculated. Results are stored in netCDF-4 files.</p> <p>Note that this code does not account for missing data from the interior of the data record! Continuity is extremely important for doing principal component analysis. When dealing with reanalysis data, the only missing data should be from days which fall outside the range of the product. The code is designed to handle these instances.</p>		
Requires	doy_leapyr, mod_struct ncFiles_read, SSWC_error_mesg, SSWC_ncFiles_write, SSWC_ncWrite_cellMethods, SSWC_varGrid_init, undefine		
Inputs	raSet	String	Name of the reanalysis dataset.
	NAM	Boolean	(OPTIONAL, default is True) Flag for setting calculation to the Northern Annular Mode.
	SAM	Boolean	(OPTIONAL, default is False) Flag for setting calculation to the Southern Annular Mode.
	RAYRS	Integer	(OPTIONAL, default is all years) Vector of years for which to calculate the annular mode.
	INPUTFILE	String	(OPTIONAL, default is raSet+'_raFields_userInput.pro') Name of user input file.

Outputs	File	SSWC_{outName}_{raSet}_s{sDate}_e{eDate}.nc in folder raInStr.fieldsDir.
Usage	IDL> SSWC_raFields_calcAnnMode, 'MERRA2', /NAM	

SSWC_raFields_calcFluxes

File	./runCode/SSWC_raFields_calcFluxes.pro		
Purpose	Calculates the zonal covariance or eddy flux of two fields.		
Description	Calculates the zonal mean of the product of two zonal eddy fields. This is done using the procedure 'decompProduct.' Results are stored in netCDF-4 files.		
Requires	decompProduct, mod_struct, ncFiles_read, SSWC_error_mesg, SSWC_ncWrite_cellMethods, SSWC_ncFiles_write, SSWC_varGrid_init		
Inputs	raSet	String	Name of the reanalysis dataset.
	waveFields	String	Single element or vector name(s) of the flux field(s) to calculate. Names are defined in the user input file.
	RAYRS	Integer	(OPTIONAL, default is all years) Vector of years for which to calculate the fluxes.
	INPUTFILE	String	(OPTIONAL, default is raSet+'_raFields_userInput.pro') Name of user input file.
Outputs	File	SSWC_{outName}_{raSet}_s{sDate}_e{eDate}.nc in folder raInStr.fieldsDir.	
Usage	IDL> SSWC_raFields_calcFluxes, 'MERRA2', 'vt'		

SSWC_raFields_calcTCO3

File	./runCode/SSWC_raFields_calcTCO3.pro		
Purpose	Calculates total column ozone given ozone mixing ratio.		
Description	Calculates total column ozone in Dobson units. Ozone mixing ratios are to be provided on pressure levels for the function 'Dobson.'		
Requires	Dobson, mod_struct, ncFiles_read, SSWC_error_mesg, SSWC_ncFiles_write, SSWC_ncWrite_cellMethods, SSWC_varGrid_init, undefine		
Inputs	raSet	String	Name of the reanalysis dataset.
	raFields	String	Single element name of the ozone mixing ratio. Names are defined in the user input file.

	RAYRS	Integer	(OPTIONAL, default is all years) Vector of years for which to calculate the fluxes.
	VOLUME_MIX	Boolean	(OPTIONAL, default is False) Flag for whether passed ozone mixing ratio is volume mixing ratio.
	MASS_MIX	Boolean	(OPTIONAL, default is True) Flag for whether passed ozone mixing ratio is mass mixing ratio.
	INPUTFILE	String	(OPTIONAL, default is raSet+'_raFields_userInput.pro') Name of user input file.
Outputs		File	SSWC_{outName}_{raSet}_s{sDate}_e{eDate}.nc in folder raInStr.fieldsDir.
Usage	IDL> SSWC_raFields_calcTCO3, 'MERRA2', 'o3', /MASS_MIX		

SSWC_raFields_calcTheta

File	./runCode/SSWC_raFields_calcTheta.pro		
Purpose	Interpolates fields to isentropic surfaces.		
Description	Linearly interpolates fields to isentropic (theta) surfaces that are either provided in the data or calculated from temperature and pressure fields. Preference is for provided theta surfaces. The output theta levels are set in 'SSWC_raFields_readGrid.' Data are only interpolated to output theta surfaces that are within the range of either provided or calculated theta surfaces.		
Requires	mod_struct, ncFiles_read, SSWC_error_mesg, SSWC_ncFiles_write, SSWC_ncWrite_cellMethods, SSWC_setTimeJul, SSWC_varGrid_init, undefine		
Inputs	raSet	String	Name of the reanalysis dataset.
	raFields	String	Single element or vector name(s) of the field(s) to interpolate to theta surfaces. Names are defined in the user input file.
	RAYRS	Integer	(OPTIONAL, default is all years) Vector of years for which to calculate the fluxes.
	INPUTFILE	String	(OPTIONAL, default is raSet+'_raFields_userInput.pro') Name of user input file.
Outputs		File	SSWC_{outName}Theta_{raSet}_s{sDate}_e{eDate}.nc in folder raInStr.fieldsDir.
Usage	IDL> SSWC_raFields_calcTheta, 'MERRA2', 'o3'		

SSWC_raFields_readGrid

File	./runCode/SSWC_raFields_readGrid.pro		
Purpose	Creates an IDL save file with horizontal and vertical grid values.		
Description	Saves original horizontal grid values, regridded horizontal grid values, and vertical grid values. This save file is routinely accessed by other procedures in the SSWC.		
Requires	raFiles_find, raFiles_read, SSWC_error_mesg, SSWC_runVersion		
Inputs	raSet	String	Name of the reanalysis dataset.
	varPres	String	Single element name for a pressure level field. Names are defined in the user input file.
	varSfc	String	Single element name for a surface level field. Names are defined in the user input file.
	INPUTFILE	String	(OPTIONAL, default is raSet+'_raFields_userInput.pro') Name of user input file.
Outputs	File	{raSet}Grid.sav in folder userInput.	
Usage	IDL> SSWC_raFields_readGrid, 'MERRA2', 'vwnd', 'tsfcMin'		

SSWC_raFields_readRaw

File	./runCode/SSWC_raFields_readRaw.pro		
Purpose	Reads raw reanalysis data and saves into intermediate files.		
Description	Reads raw reanalysis data. Performs horizontal interpolation if requested for a given field. Chunks data into July 01-June 30 annual segments. Saves output to netCDF-4 files.		
Requires	mod_struct, raFiles_read, SSWC_error_mesg, SSWC_ncWrite_cellMethods, SSWC_ncFiles_write, SSWC_setTimeJul, SSWC_varGrid_init, undefine		
Inputs	raSet	String	Name of the reanalysis dataset.
	raFields	String	Single element or vector of name(s) for the fields to read. Names are defined in the user input file.
	RAYRS	Integer	(OPTIONAL, default is all years) Vector of years for which to calculate the fluxes.
	INPUTFILE	String	(OPTIONAL, default is raSet+'_raFields_userInput.pro') Name of user input file.
Outputs	File	SSWC_{outName}_{raSet}_s{sDate}_e{eDate}.nc in folder raInStr.fieldsDir.	

Usage	IDL> SSWC_raFields_readRaw, 'MERRA2', 'vwnd'
-------	--

Supporting procedures

decompProduct

File	./procedures/decompProduct.pro		
Purpose	Computes the product of Fourier wavenumber decompositions on two fields.		
Description	This function is used to calculate wave fluxes or zonal covariances of two fields. A fast Fourier transform (FFT) is performed on both fields. The product is taken between the FFT coefficients of one field and the conjugate FFT coefficients of the other. The real part of this is the zonal covariance field. The function may return individual wavenumbers or sums over ranges of wavenumbers.		
Requires			
Inputs	inArr1	Double	Array of values.
	inArr2	Double	Array of values of same size as inArr1.
	freqLo	Integer	Single element or vector of values giving lower frequency bounds.
	freqHi	Integer	Single element or vector of values with same size as freqLo giving upper frequency bounds.
	DESTROY	Boolean	(OPTIONAL, default is False) Deallocates the input arrays if set to True.
	DIMENSION	Integer	(OPTIONAL) Not currently utilized.
Outputs	waveVals	Double	Array of size [N_ELEMENTS(freqLo), (SIZE(inArr1,/DIM))[1:*]].
Usage	<pre>IDL> inArr1 = RANDOMU(seed,100,100)*50.D IDL> inArr2 = RANDOMU(seed,100,100)*50.D IDL> waveVals = decompProduct(inArr1, inArr2, [0,1,2,0], [0,1,2,2], /DESTROY)</pre>		

DOY

File	./procedures/SSWC_misc_mod.pro		
Purpose	Calculates the day of the year.		
Description	Returns the day of the year, where Jan 01 is 1, Jan 02 is 2, etc. Dec 31 is 365 in non-leap years and 366 in leap years.		
Requires	SSWC_pro_init		

Inputs	jdate	Double	Single element or vector of Julian dates.
Outputs	doyR	Double	Single element or vector of same size as jdate containing days of the year.
Usage	IDL> doyr = DOY(JULDAY(01,01,1999)+DINDGEN(365))		

DOY_leapYr

File	./procedures/SSWC_misc_mod.pro		
Purpose	Calculates the day of the year relative to a leap year.		
Description	Returns the day of the year, where Jan 01 is 1, Jan 02 is 2, etc. Dec 31 is always 366.		
Requires	SSWC_pro_init		
Inputs	jdate	Double	Single element or vector of Julian dates.
Outputs	doyR	Double	Single element or vector of same size as jdate containing days of the year.
Usage	IDL> doyr = DOY_leapYr(JULDAY(01,01,1999)+DINDGEN(365))		

gauss_lats

File	./procedures/SSWC_vorticity_mod.pro		
Purpose	Calculates the Gaussian latitudes for a given number of latitudes.		
Description	This code was adapted to IDL from the Guass2lats MATLAB code written by Tom Holt.		
Requires	SSWC_pro_init		
Inputs	nLats	Integer	Single element number of latitudes.
Outputs	gaus_lat	Double	Vector of Gaussian latitudes.
Usage	IDL> gaus_lat = gauss_lats(73)		

gridBounds_set

File	./procedures/SSWC_misc_mod.pro		
Purpose	Calculates the boundaries of grids.		
Description	Grid boundaries are needed for CF-compliant netCDF files.		
Requires	SSWC_pro_init		
Inputs	gridVals	Double	Vector of grid values.
	GRIDMIN	Double	(OPTIONAL, default is MIN(gridVals)) Minimum value of the grid.
	GRIDMAX	Double	(OPTIONAL, default is MAX(gridVals)) Maximum value of the grid.

	GRIDWRAP	Boolean	(OPTIONAL, default is False) Flag for whether the grid wraps (as it does for longitude) or not.
Outputs	gridBounds	Double	Array of grid bounds with size [SIZE(grid Vals,/DIM),2].
Usage	IDL> gridVals = DINDGEN(144)*2.5D - 180.D IDL> gridBounds = gridBounds_set(grid Vals, /GRIDWRAP)		

hdfFiles_read

File	./procedures/SSWC_read_mod.pro		
Purpose	Reads a variable and dimensions from a Hierarchical Data Format (HDF) file.		
Description			
Requires	SSWC_pro_init		
Inputs	raFName	String	Name of the HDF file.
	LONNAME	String	(OPTIONAL) Name of the longitude dimension.
	LATNAME	String	(OPTIONAL) Name of the latitude dimension.
	LEVNAME	String	(OPTIONAL) Name of the pressure dimension.
	THENAME	String	(OPTIONAL) Name of the theta dimension.
	TIMENAME	String	(OPTIONAL) Name of the time dimension.
	VARNAME	String	(OPTIONAL) Name of the variable.
Outputs	LONVALS	Double	(OPTIONAL) Output longitude values, if LONNAME is properly set.
	LATVALS	Double	(OPTIONAL) Output latitude values, if LATNAME is properly set.
	LEVVALS	Double	(OPTIONAL) Output pressure values, if LEVNAME is properly set.
	THEVALS	Double	(OPTIONAL) Output theta values, if THENAME is properly set.
	TIMEVALS	Double	(OPTIONAL) Output time values, if TIMENAME is properly set.
	VARVALS	Double	(OPTIONAL) Output variable values, if VARNAME is properly set.
Usage	IDL> hdfFiles read, 'hdfFile.hdf', \$ LONNAME='lon', VARNAME='uwnd' LONVALS=lons, VARVALS=uwnd		

ncFiles_read

File	./procedures/SSWC_read_mod.pro		
Purpose	Reads a variable and dimensions from a Network Common Data Form (netCDF or NC) file.		
Description			
Requires	SSWC_pro_init		
Inputs	raFName	String	Name of the netCDF file.
	LONNAME	String	(OPTIONAL) Name of the longitude dimension.
	LATNAME	String	(OPTIONAL) Name of the latitude dimension.
	LEVNAME	String	(OPTIONAL) Name of the pressure dimension.
	THENAME	String	(OPTIONAL) Name of the theta dimension.
	TIMENAME	String	(OPTIONAL) Name of the time dimension.
	VARNAME	String	(OPTIONAL) Name of the variable.
Outputs	LONVALS	Double	(OPTIONAL) Output longitude values, if LONNAME is properly set.
	LATVALS	Double	(OPTIONAL) Output latitude values, if LATNAME is properly set.
	LEVVALS	Double	(OPTIONAL) Output pressure values, if LEVNAME is properly set.
	THEVALS	Double	(OPTIONAL) Output theta values, if THENAME is properly set.
	TIMEVALS	Double	(OPTIONAL) Output time values, if TIMENAME is properly set.
	VARVALS	Double	(OPTIONAL) Output variable values, if VARNAME is properly set.
	VARCOORDS	String	(OPTIONAL) Returns string containing the coordinate name(s) for the variable VARNAME.
	TINTERVAL	Double	(OPTIONAL) Returns the global attribute 't_interval' from the NetCDF file, if it exists.
	XINTERVAL	Double	(OPTIONAL) Returns the global attribute 'x_interval' from the NetCDF file, if it exists.
	YINTERVAL	Double	(OPTIONAL) Returns the global attribute 'y_interval' from the NetCDF file, if it exists.
Usage	<pre>IDL> ncFiles read, 'ncFile.nc', \$ LONNAME='lon', VARNAME='uwnd' LONVALS=lons, VARVALS=uwnd</pre>		

next_leapYr

File	./procedures/SSWC_misc_mod.pro		
Purpose	Returns the next leap year in the forwards or backwards direction.		
Description			
Requires	SSWC_pro_init		
Inputs	inYr	Double	Single element year value.
	PRIOR	Boolean	(OPTIONAL, default is FALSE) Flag for direction of search. True searches to prior years.
Outputs	nextLeapYr	Double	Single element year value.
Usage	IDL> nextLeapYr = next_leapYr(1998)		

raFiles_find

File	./procedures/SSWC_read_mod.pro		
Purpose	Finds the files for a given dataset. Often used to find subsets of the files, such as those falling between a range of years and/or months.		
Description			
Requires	SSWC_pro_init		
Inputs	raStr	Structure	The raInStr returned by the userInput procedure.
	varStr	Structure	A single variable structure from the varInStr structure returned by the userInput procedure.
	RAYRS	Integer	(OPTIONAL) Single element or vector of years to span.
	RAMONS	Integer	(OPTIONAL) Single element or vector of months to span.
Outputs	fNames	String	Vector of file names.
	SUBYRS	Integer	(OPTIONAL) Vector of years that the files span.
Usage	IDL> fNames = raFiles_find(raInStr, varInStr)		

raFiles_read

File	./procedures/SSWC_read_mod.pro		
Purpose	Reads input data files, removes missing points, limits to desired time bounds, and performs time averaging or statistics.		
Description			
Requires	hdfFiles_read, ncFiles_read, SSWC_error_mesg, SSWC_pro_init, undefine		
Inputs	raStr	Structure	The raInStr returned by the userInput procedure.

	varStr	Structure	A single variable structure from the <code>varInStr</code> returned by the <code>userInput</code> procedure.
	RAYRS	Integer	(OPTIONAL) Single element or vector of years over which to read.
	RAMONS	Integer	(OPTIONAL) Single elements or vector of months over which to read.
	FTYPE	String	(OPTIONAL) Type of file. Valid types: 'nc,' 'hdf,' and 'grb.' Grib reading is not currently enabled, however.
	REGRIDLONS	Double	(OPTIONAL) Longitude values to which to regrid the data.
	REGRIDLATS	Double	(OPTIONAL) Latitude values to which to regrid the data.
	TIMEBND	Double	(OPTIONAL) Vector of times over which to find bounds. The lower and upper time bounds are calculated from the minimum and maximum, respectively, of <code>TIMEBND</code> .
	TIMEADJUST	Boolean	(OPTIONAL) Flag for whether to apply time adjustments set in <code>raInStr</code> (1) or not (0).
Outputs	VARVALS	Double	(OPTIONAL) Array of variable values.
	LONGITUDE	Double	(OPTIONAL) Vector of longitude values.
	LATITUDE	Double	(OPTIONAL) Vector of latitude values.
	PRESSURE	Double	(OPTIONAL) Vector of pressure values.
	THETA	Double	(OPTIONAL) Vector of theta values.
	TIMEVALS	Double	(OPTIONAL) Vector of time values.
	TIMEINTERVAL	Double	(OPTIONAL) Returns the global attribute 't_interval' from the netCDF file, if it exists.
	LONINTERVAL	Double	(OPTIONAL) Returns the global attribute 'x_interval' from the netCDF file, if it exists.
	LATINTERVAL	Double	(OPTIONAL) Returns the global attribute 'y_interval' from the netCDF file, if it exists.

Usage	IDL> raFiles_read, raInStr, varStr, \$ RAYRS=[1999, 2000], \$ TIMEBnds=[JULDAY(07,01,1999),JULDAY(06,30,2000)], \$ LONGITUDE=lons, VARVALS=vwnd
-------	--

readEvDates

File	./procedures/SSWC_events_mod.pro		
Purpose	Reads the event list and creates the needed event date types.		
Description	Returns a structure containing event dates in the following formats: event dates Julian dates, start dates Julian dates, end dates Julian dates, event dates as 'YYYYMMDD,' event dates as 'YYYY-MM-DD,' and event years as 'YYYY.'		
Requires	SSWC_error_mesg, SSWC_pro_init		
Inputs	inputDir	String	Name of directory, with path included.
	evFName	Integer	Name of the event list.
	raRange	Boolean	(OPTIONAL, default is False) If set to true, will continue if no match is found.
Outputs	evStr	Structure	Structure containing a number of instances of the event dates in different formats.
Usage	IDL> evStr = readEvDates('./userInput/', 'evList_MERRA2', [JULDAY(07,01,1999), JULDAY(06,30,2000)]) IDL> print, evStr.evDates_iso		

SSWC_constants_define

File	./procedures/SSWC_misc_mod.pro		
Purpose	Defines useful constants.		
Description	Creates a common block for later referencing of useful constants.		
Requires	SSWC_pro_init		
Inputs			
Outputs	CONSTANTS	Common	Common block containing useful constants.
Usage	IDL> !NULL = SSWC_constants_define()		

SSWC_dir_search

File	./procedures/SSWC_misc_mod.pro		
Purpose	Wrapper for IDL intrinsic FILE_SEARCH searching for directories.		
Description			
Requires	SSWC_error_mesg, SSWC_pro_init		

Inputs	dName	String	Name of directory, with path included.
	CONTINUE	Boolean	(OPTIONAL, default is False) If set to true, will continue if no match is found.
	_REF_EXTRA		(OPTIONAL) Additional keyword inputs to FILE_SEARCH.
Outputs	outDirs	String	String vector of length equal to the number of matched directories.
	COUNT	Integer	(OPTIONAL) Returns number of matches.
Usage	IDL> outDirs = SSWC_dir_search('/opt/local/exelis/idl85/bin/', /CONTINUE)		

SSWC_error_mesg

File	./procedures/SSWC_error_mesg.pro		
Purpose	Creates an error message with traceback after an error has occurred.		
Description			
Requires			
Inputs	message_text	String	Single element or vector of error message(s) including traceback.
	FATAL	Boolean	(OPTIONAL) Flag for whether to halt execution (1) or not (0).
	CONTINUE	Boolean	(OPTIONAL, default is 1) Flag for whether to continue (1) or not (0).
	CAUGHT	Boolean	(OPTIONAL) Flag for whether error was caught (1) or initiated by SSWC checks (0).
Outputs			
Usage	IDL> SSWC_error_mesg, 'There was a problem here.', /FATAL		

SSWC_events_mod

File	./procedures/SSWC_events_mod.pro		
Purpose	Empty procedure used for initializing other helpful procedures.		
Description			
Requires	SSWC_pro_init		
Inputs			
Outputs			
Usage	IDL> RESOLVE ROUTINE, 'SSWC_events_mod', /EITHER, /COMPILE_FULL_FILE		

SSWC_file_search

File	./procedures/SSWC_misc_mod.pro		
Purpose	Wrapper for IDL intrinsic FILE_SEARCH searching for files.		
Description			
Requires	SSWC_error_mesg, SSWC_pro_init		
Inputs	fName	String	Name of file, with path included.
	CONTINUE	Boolean	(OPTIONAL, default is False) If set to true, will continue if no match is found.
	_REF_EXTRA		(OPTIONAL) Additional keyword inputs to FILE_SEARCH.
Outputs	outFiles	String	String vector of length equal to the number of matched files.
	COUNT	Integer	(OPTIONAL) Returns number of matches.
Usage	IDL> outDirs = SSWC_file_search('/opt/local/exelis/idl85/bin/idl', /CONTINUE)		

SSWC_gridData_read

File	./procedures/SSWC_gridData_read.pro		
Purpose	Simple procedure for restoring the grid data IDL save file.		
Description			
Requires	SSWC_pro_init		
Inputs	dirInput	String	Directory containing the grid IDL save file.
	gridFName	String	Name of the IDL save file.
Outputs	outLons	Double	Longitude values.
	outLats	Double	Latitude values.
	outPres	Double	Pressure values.
	outTheta	Double	Theta values.
	outSLons	Double	Surface longitude values.
	outSLats	Double	Surface latitude values.
Usage	IDL> SSWC_gridData_read, './userInput/', 'MERRA2Grid.sav', outLons, outLats, outPres, outTheta, outSLons, outSLats		

SSWC_misc_mod

File	./procedures/SSWC_misc_mod.pro		
Purpose	Empty procedure used for initializing other helpful procedures.		
Description			
Requires	SSWC_pro_init		
Inputs			
Outputs			

Usage	IDL> RESOLVE_ROUTINE, 'SSWC_misc_mod', /EITHER, /COMPILE_FULL_FILE
-------	---

SSWC_ncFiles_write

File	./procedures/SSWC_ncFiles_write.pro		
Purpose	Writes out a netCDF-4 file.		
Description	Output data are expected to be defined as structures in the scope of the calling routine with a common block named 'OUT_FIELDS'. This procedure will set a number of standard global attributes that are required for CF-compliance. Other attributes, such as extra globals and variable attributes, are expected to be defined and passed properly (i.e. send globals to the keyword parameter and variable attributes with the structures containing the variables). Compression via short integer packing is enabled by default.		
Requires	SSWC_dir_search, SSWC_error_mesg, SSWC_pro_init		
Inputs	outNames	String	Single element or vector of names of the structures which to write to output. Should be defined in a common block 'OUT_FIELDS'.
	ncName	String	Name of the output netCDF file. Full path expected.
	SURFACE	Boolean	(OPTIONAL, deprecated)
	GLOBAL	String	(OPTIONAL) Single element or vector of extra global attributes. Should be in the form [ATTRIB_NAME, ATTRIB_VAL].
Outputs			
Usage	<pre>IDL> COMMON OUT_FIELDS, vwnd IDL> vwnd = {vwnd, values:RANDOMU(seed,720,361,10,365)} IDL> SSWC_ncFiles_write, 'vwnd', 'vwnd_out.nc', \$ GLOBAL=['Test attribute', 'Test value']</pre>		

SSWC_ncWrite_cellMethods

File	./procedures/SSWC_misc_mod.pro		
Purpose	Produces CF-compliant cell_methods attributes.		
Description			
Requires	SSWC_pro_init		
Inputs	tInterval	Double	Original time interval of the data.
	xInterval	Double	Original x-dimension interval of the data.
	yInterval	Double	Original y-dimension interval of the data.

	ZNAME	String	(OPTIONAL, default is 'pres') Name of the vertical dimension.
	TIMENAME	String	(OPTIONAL, default is 'timeEv60') Name of the time dimension.
	OVERMETHOD	String	(OPTIONAL) Comments to overwrite a specific descriptor in cell_methods. May contain comments for multiple descriptors. These are to be semi-colon separated and must begin with 'AREAMETHOD:', 'XMETHOD:', 'YMETHOD:', 'ZMETHOD:', or 'TIMEMETHOD:'.
	ADDMETHOD	String	(OPTIONAL) Comments to be concatenated to a specific descriptor in cell_methods. Format must be the same as OVERMETHOD.
Outputs	cellMethods	String	String containing the attribute cell_methods.
Usage	<pre>IDL> cellMethods = SSWC_ncWrite_cellMethods(0.25, 1.5, 1.5, \$ IDL> TIMENAME='timeJul', \$ IDL> OVERMETHOD= 'TIMEMETHOD:timeEv30: minimum over days (anomaly from climatology);'</pre>		

SSWC_ncWrite_define

File	./procedures/SSWC_misc_mod.pro
Purpose	Initializes structures that are used for writing SSWC outputs.
Description	The output netCDF writing procedure SSWC_ncFiles_write makes use of IDL structures, which may contain a number of keys with different data types. Structures allow for easy passing and maintenance of output data. The most common output attributes for fields and dimensions are defined here in the NCWrite and NCDimWrite structures, respectively.
Requires	SSWC_pro_init
Inputs	
Outputs	
Usage	IDL> SSWC_ncWrite_define

SSWC_ncWrite_validAttributes

File	./procedures/SSWC_misc_mod.pro
Purpose	Defines valid attributes for CF-compliant netCDF files.

Description	Creates a common block containing an array of valid attributes. If an attribute is passed to the output procedure <code>SSWC_ncFiles_write</code> and it is not a valid attribute, then it is not saved.
Requires	<code>SSWC_pro_init</code>
Inputs	
Outputs	
Usage	<code>IDL> SSWC_ncWrite_validAttributes</code>

SSWC_outFields_readDist

File	<code>./procedures/SSWC_outFields_readDist.pro</code>
Purpose	Reads climatological distribution files and places the contents into variables defined in a new common block 'EI_VALS'.
Description	
Requires	<code>ncFiles_read</code> , <code>SSWC_error_mesg</code> , <code>SSWC_pro_init</code>
Inputs	<code>dataStr</code> Structure The <code>varInStr</code> structure returned by the <code>userInput</code> procedure.
Outputs	
Usage	<code>IDL> SSWC_outFields_readDist, varInStr</code>

SSWC_outFields_readFiles

File	<code>./procedures/SSWC_outFields_readFiles.pro</code>
Purpose	Reads the annually-packed intermediate data.
Description	The dataset structures and information, as well as the years over which to read, are to be passed. Variables with the same names as the <code>varInStr.varName</code> names are expected to be already defined in common block 'DATA_VALS'. Returns structures containing the data for those variables which are defined in 'DATA_VALS'.
Requires	<code>ncFiles_read</code> , <code>SSWC_error_mesg</code> , <code>SSWC_pro_init</code> , <code>SSWC_setTimeJul</code>
Inputs	<code>raInStr</code> Structure The <code>raInStr</code> structure returned by the <code>userInput</code> procedure.
	<code>dataStr</code> Structure The <code>varInStr</code> structure returned by the <code>userInput</code> procedure.
	<code>raSet</code> String Name of the dataset.
	<code>inYrs</code> String String of years for the data, passed in the form 'YYYY_YYYY'. The sets of 'YYYY' are not expected to be the same.
Outputs	
Usage	<code>IDL> SSWC_outFields_readFiles, raInStr, varInStr, 'MERRA2', \$ '1999_2000'</code>

SSWC_outFields_removeClim

File	./procedures/SSWC_outFields_removeClim.pro		
Purpose	Calculates anomalies of a field.		
Description	Reads climatology files and aligns the climatologies with the data fields. Calculates the anomalies and, if requested, the standardized anomalies. Returns the anomalies in structures if the 'Anom' variable of a field is defined in 'DATA_VALS'.		
Requires	DOY_leapYr, ncFiles_read, SSWC_error_mesg, SSWC_pro_init, SSWC_setTimeJul, Undefine		
Inputs	dataStr	Structure	The <code>varInStr</code> structure returned by the <code>userInput</code> procedure.
	STANDARDIZE	Boolean	Flag for whether to return standardized anomalies (1) or not (0). Standardized anomalies are anomalies divided by their standard deviation.
Outputs			
Usage	IDL> SSWC_outFields_removeClim, dataStr		

SSWC_pro_init

File	./procedures/SSWC_pro_init.pro		
Purpose	Batch file for procedure initialization. Sets compile options, error handling, and version control.		
Description			
Requires	SSWC_error_mesg, SSWC_runVersion		
Inputs			
Outputs			
Usage	IDL> @SSWC_pro_init		

SSWC_raFields_rawTheta

File	./procedures/SSWC_misc_mod.pro		
Purpose	Updates <code>gridTheta</code> structure with values and bounds.		
Description	Used for reading raw data on isentropic (theta) surfaces, this allows the <code>SSWC_raFields_readRaw</code> procedure to store the dataset's native theta surfaces. Data on native theta surfaces is later interpolated to standard isentropic surfaces.		
Requires	SSWC_pro_init, gridBounds_set		
Inputs	rawTheta	Double	Vector of native theta surface grid values.
Outputs			
Usage	IDL> thetaVals = [850.D,550.D,450.D,350.D] IDL> !NULL = SSWC_raFields_rawTheta(thetaVals)		

SSWC_read_mod

File	./procedures/SSWC_read_mod.pro
Purpose	Empty procedure used for initializing other helpful procedures.
Description	
Requires	SSWC_pro_init
Inputs	
Outputs	
Usage	IDL> RESOLVE_ROUTINE, 'SSWC_read_mod', /EITHER, \$ /COMPILE_FULL_FILE

SSWC_readEvents

File	./procedures/SSWC_events_mod.pro
Purpose	Defines a common block containing event dates.
Description	Defines the EVENTS common block.
Requires	readEvDates, SSWC_pro_init
Inputs	raRange Double Two element vector containing the range of possible event dates. Type is Julian date.
Outputs	
Usage	IDL> SSWC_readEvents, \$ [JULDAY(07,01,1999),JULDAY(06,30,2000)]

SSWC_runVersion

File	./procedures/SSWC_runVersion.pro
Purpose	Function where the SSWC version is set and returned.
Description	If the version changes, this is where to actually adjust the number.
Requires	
Inputs	
Outputs	runVersion String Version of the SSWC.
Usage	IDL> runVersion = SSWC_runVersion()

SSWC_setTimeJul

File	./procedures/SSWC_misc_mod.pro
Purpose	Updates timeJul structure with values and bounds.
Description	Used for reading raw data, this allows the SSWC_raFields_readRaw procedure to store the dataset's Julian dates.
Requires	SSWC_pro_init
Inputs	timeJulV Double Vector of Julian dates.
Outputs	
Usage	IDL> timeJulV = JULDAY(07,01,1999) + DINDGEN(366) IDL> !NULL = SSWC_setTimeJul(timeJulV)

SSWC_setTimeUnits

File	./procedures/SSWC_misc_mod.pro		
Purpose	Updates time variable's 'units' attributes to be CF-compliant.		
Description	Sets the units as 'days since YYYY-MM-DD' where 'YYYY-MM-DD' is set to be the relevant date.		
Requires	SSWC_pro_init		
Inputs	timeVar	String	Name of the time variable.
	timeUnitAdd	String	ISO formatted string date.
Outputs			
Usage	IDL> timeUnitAdd = '1999/07/01' IDL> SSWC_setTimeUnits, 'timeJulV', timeUnitAdd		

SSWC_varAnom_init

File	./procedures/SSWC_varAnom_init.pro		
Purpose	Initializes the structures for the varAnom output files.		
Description			
Requires	SSWC_pro_init		
Inputs			
Outputs			
Usage	IDL> SSWC_varAnom_init		

SSWC_varDerive_init

File	./procedures/SSWC_varDerive_init.pro		
Purpose	Initializes the structures for the varDerive output files.		
Description			
Requires	SSWC_pro_init		
Inputs			
Outputs			
Usage	IDL> SSWC_varDerive_init		

SSWC_varFull_init

File	./procedures/SSWC_varFull_init.pro		
Purpose	Initializes the structures for the varFull output files.		
Description			
Requires	SSWC_pro_init		
Inputs			
Outputs			
Usage	IDL> SSWC_varFull_init		

SSWC_varGrid_init

File	./procedures/SSWC_varGrid_init.pro
Purpose	Initializes the dimensional structures for grid variables.
Description	
Requires	gridBounds_set, SSWC_gridData_read, SSWC_ncWrite_define, SSWC_ncWrite_validAttributes, SSWC_pro_init
Inputs	
Outputs	
Usage	IDL> SSWC_varGrid_init

SSWC_vorticity_mod

File	./procedures/SSWC_vorticity_mod.pro
Purpose	Empty procedure used for initializing other helpful procedures.
Description	
Requires	SSWC_pro_init
Inputs	
Outputs	
Usage	IDL> RESOLVE ROUTINE, 'SSWC_vorticity_mod', /EITHER, /COMPILE_FULL_FILE

tseriesClimIdxs

File	./procedures/tseriesClimIdxs.pro									
Purpose	Read raw climate index files, interpolates to a daily time grid, and save into structures.									
Description										
Requires	SSWC_error_mesg, SSWC_pro_init, undefine									
Inputs	<table border="0"> <tr> <td>raInStr</td> <td>Structure</td> <td>The raInStr structure returned by the userInput procedure.</td> </tr> <tr> <td>climInStr</td> <td>Structure</td> <td>The climInStr structure returned by the userInput procedure.</td> </tr> <tr> <td>evIdx</td> <td>Integer</td> <td>Integer index of the event.</td> </tr> </table>	raInStr	Structure	The raInStr structure returned by the userInput procedure.	climInStr	Structure	The climInStr structure returned by the userInput procedure.	evIdx	Integer	Integer index of the event.
raInStr	Structure	The raInStr structure returned by the userInput procedure.								
climInStr	Structure	The climInStr structure returned by the userInput procedure.								
evIdx	Integer	Integer index of the event.								
Outputs										
Usage	IDL> tseriesClimIdxs, raInStr, climInStr, evIdx									

tseriesElcount

File	./procedures/tseriesElcount.pro
Purpose	Determines if points in space and time are more than or less than (the default) the given distribution values.
Description	Returns values in structures that have been initialized. The data type is integer, with 1 indicating that the field was outside of the distribution value.

Requires	DOY_leapYr, SSWC_error_mesg, SSWC_ncWrite_cellMethods, SSWC_pro_init, undefine		
Inputs	inFieldName	String	Name of the field variable.
	inDist	String	Name of the distribution variable.
	outName	String	Name of the output variable.
	evIdx	Integer	Integer index of the event.
	MORE	Boolean	(OPTIONAL, default is 0) If set, finds instances of exceedance which are larger than the distribution values.
Outputs			
Usage	IDL> tseriesElcount, 'tsfcMin', 'tsfcMin_90', 'eiTN90P', \$ evIdx, /MORE		

tseriesExtrema

File	./procedures/tseriesExtrema.pro		
Purpose	Finds the locations and values of extrema (both minima and maxima) for a given data field between set latitude and pressure bounds.		
Description	For a given field, will determine time series of either the minima or maxima and the locations of these. Spatial locations include position in height, pressure, longitude, and latitude. The extrema are found in space bounded by 30N to 90N, and by 300 hPa to the uppermost pressure level.		
Requires	SSWC_error_mesg, SSWC_ncWrite_cellMethods, SSWC_pro_init, undefine		
Inputs	inFieldName	String	Name of the field variable.
	hgtFieldName	String	Name of the geopotential height variable.
	evIdx	Integer	Integer index of the event.
	MIN	Boolean	(OPTIONAL, default is 0) If set, calculates the minima.
	MAX	Boolean	(OPTIONAL, default is 1) If set, calculates the maxima.
Outputs			
Usage	IDL> tseriesExtrema, 'uwnd', 'geop', evIdx, /MIN		

tseriesFull

File	./procedures/tseriesFull.pro		
Purpose	Extracts the ± 60 days of data for a field around a given date.		
Description	For a given field, will extract the 60 days surrounding a given event and at a given level (if requested).		

Requires	SSWC_error_mesg, SSWC_ncWrite_cellMethods, SSWC_pro_init, undefine		
Inputs	inFieldName	String	Name of the field variable.
	outName	String	Name of the output variable.
	evIdx	Integer	Integer index of the event.
	PLVL	String	(OPTIONAL, default is 'All') Can be set to 'All', 'SFC', or any pressure level value. If 'SFC' or pressure level value, only saves data on a single latitude-longitude level.
Outputs			
Usage	IDL> tseriesFull, 'uwnd', 'uwndFull_TS', evIdx		

tseriesReversal

File	./procedures/tseriesReversal.pro		
Purpose	Calculates the dates and pressures at which a field's zonal mean reverses from positive to negative.		
Description	Used mainly for calculating reversals of zonal mean zonal winds, this will produce two output fields: a (nLatitude,nPressure) array holding the dates of reversal, and a (nLatitude,nDay) array holding the pressure levels of reversal.		
Requires	SSWC_error_mesg, SSWC_ncWrite_cellMethods, SSWC_pro_init, undefine		
Inputs	inFieldName	String	Name of the field variable.
	evIdx	Integer	Integer index of the event.
Outputs			
Usage	IDL> tseriesReversal, 'uwnd', evIdx		

tseriesVorticityCalc

File	./procedures/SSWC_vorticity_mod.pro		
Purpose	Calculates absolute vorticity from input wind fields.		
Description	Requires zonal wind and meridional wind on the same grid. This code is specific to the ± 60 day window around events. Calculates and returns both unsmoothed and smoothed absolute vorticity.		
Requires	gauss_lats, inv_spherical_transform, spherical_transform, SSWC_error_mesg, SSWC_ncWrite_cellMethods, SSWC_pro_init		
Inputs	uFieldName	String	Name of the zonal wind field.
	vFieldName	String	Name of the meridional wind field
	outFull	String	Name of the output unsmoothed vorticity field.
	outFilt	String	Name of the output smoothed vorticity field.
	evIdx	Integer	Integer index of the event.

	LMAX	Integer	(OPTIONAL, default is 11) Spherical harmonic at which to perform triangular truncation. Higher numbers result in less smoothing.
Outputs			
Usage	IDL> evIdx = 3 IDL> tseriesVorticity_calc, 'uwnd', 'vwnd', 'vortFull_TS', 'vortFilt_TS', evIdx, LMAX=11		

External procedures

Dobson

File	./procedures/Dobson.pro		
Purpose	Calculates total column ozone provided pressure and ozone mass mixing ratio on those pressure levels.		
Description	Written by Mark Hervig, edited for multi-dimensional inputs by Sean Davis.		
Requires			
Inputs	P	Double	Vector or array of pressure levels in hPa.
	o3	Double	Vector or array (same size as p) of ozone mass mixing ratio in units kg/kg.
Outputs	dobs	Double	Vector or array (same size as p) of total column ozone in units Dobsons.
Usage	IDL> dobs = Dobson(pres,o3)		

inv_spherical_transform

File	./procedures/SSWC_vorticity_mod.pro		
Purpose	Performs an inverse spherical harmonic transform on a 2-D array.		
Description	Code written by M. DeRosa.		
Requires	SSWC_pro_init		
Inputs	B	Complex	Array to be transformed.
	Cp	Double	Cosine of theta collocation points.
	Lmax	Long	Maximum zonal wavenumber l.
	Mrange	Long	Maximum meridional wavenumber m.
	Period	Double	Periodicity factor in longitude phi.

	Phirange	Double	(OPTIONAL) Single element or two element vector with range of longitude phi in radians.
	Cprange	Double	(OPTIONAL) Single element or two element vector with range of cp.
	Thindex	Double	(OPTIONAL) Custom array of colatitude coordinates in radians.
	Phindex	Double	(OPTIONAL) Custom array of longitude coordinates in radians.
Outputs	A	Double	Transformed array.
Usage	<pre>IDL> nLons = 144 IDL> lons = DINDGEN(nLons)*2.5D - 180.D IDL> gaus_lat = 90.D + REVERSE(gaus_lats(73)) IDL> A = inv_spherical_transform(B, COS(gaus_lat*!DPI/180.D), LMAX=11, PHINDEX=SHIFT(lons,- nLons/2)*!DPI/180.D, THINDEX=(gaus_lat*!DPI/180.D))</pre>		

match

File	./procedures/match.pro		
Purpose	Finds indices where two vectors have matching values within a given range epsilon.		
Description	Written by D. Lindler, W. Landsman, and Kim Tolbert.		
Requires			
Inputs	a	Any	Vector of values.
	b	Any	Vector of values.
	SORT	Boolean	(OPTIONAL, default is 0) Selects an alternative sorting algorithm.
	EPSILON	Double	(OPTIONAL, default is 0) Values within epsilon are considered a match.
Outputs	suba	Integer	Vector of matching indices in vector a.
	subb	Integer	Vector of matching indices in vector b.
	COUNT	Integer	Total number of matching elements.
Usage	IDL> match, [4,5,6], [5,6,7], matchA, matchB, COUNT=count		

mod_struct

File	./procedures/mod_struct.pro		
Purpose	Modifies and existing IDL structure by changing, adding, or deleting a tag.		
Description	Written by Konrad R. W. Tristram.		

Requires			
Inputs	oldstr	Structure	Input structure.
	tagnam	String	Name of the tag to modify.
	tagval	Any	Value of the tag to modify.
	POSITI	Integer	(OPTIONAL) Position of the modified tag.
	DELETE	Boolean	(OPTIONAL, default is 0) If true (1) then the tag is deleted.
Outputs	newstr	Structure	Output structure with the modified tag.
Usage	<pre>IDL> origStr = {origStr,values:DINDGEN(21)+5.D,name: 'Example'}</pre> <pre>IDL> newStr = mod_struct(origStr, 'name', 'Altered example')</pre>		

spherical_transform

File	./procedures/SSWC_vorticity_mod.pro		
Purpose	Performs a spherical harmonic transform on a 2-D array.		
Description	Code written by M. DeRosa.		
Requires	SSWC_pro_init		
Inputs	A	Double	Array to be transformed.
	Cp	Double	Cosine of theta collocation points.
	Lmax	Long	Maximum zonal wavenumber l.
	Period	Double	Periodicity factor in longitude phi.
Outputs	B	Complex	Transformed array.
Usage	<pre>IDL> gaus_lat = 90.D + REVERSE(gaus_lats(73))</pre> <pre>IDL> B = inv_spherical_transform(A,</pre> <pre> COS(gaus_lat*!DPI/180.D)</pre>		

undefine

File	./procedures/undefine.pro		
Purpose	Undefines a variable within an IDL session or program.		
Description	Written and copyrighted by David W. Fanning.		
Requires			
Inputs	variable	Any	The variable to be deleted. Up to 10 variables may be specified.
	DEFINE	Boolean	(OPTIONAL, default is 0) If set, returns the variables as type NULL rather than fully undefined. Useful for defining variables which will be used later.
Outputs			

Usage	<pre>IDL> UNDEFINE,/DEFINE,variable IDL> variable = DINDGEN(45)*5.D + 1.D IDL> origStr = {origStr,values:variable,name: 'Example'} IDL> UNDEFINE, variable, origStr</pre>
-------	---

weights_legendre

File	./procedures/SSWC_vorticity_mod.pro		
Purpose	Returns the Legendre integration weights given an array of collocation points x.		
Description	Code written by M. DeRosa.		
Requires	SSWC_pro_init		
Inputs	x	Double	Array of collocation points.
Outputs	weights	Double	Array of Legendre integration weights.
Usage	<pre>IDL> gaus_lat = 90.D + REVERSE(gaus_lats(73)) IDL> weights = weights_legendre(COS(gaus_lat*!DPI/180.D))</pre>		

Changelog

v1.0 : First official release.