PMIP4 SATIRE-M solar forcing data

The files contain Solar Spectral Irradiance (SSI) reconstruction based on **14C** or **10Be** and the SATIRE-M model, with and without an adaptation of the spectral irradiance to the CMIP6 *historical* forcing.

- 14C data for the last 9000 years (6754.5 BC to December 2015)
- 10Be data for the years 885 CE to December 2015
- In both cases, the data is daily, starting on January 1st 1850, and yearly before

The **14C-based data set** scaled to the CMIP6 historical forcing is the **recommended forcing for the PMIP4-CMIP6 *tier-1* past1000 experiment**.

Data files

- file names = `SSI_<type>_cycle_yearly_cmip_v20160613_<scaling>.txt`
  - `type` = 14C or 10Be
  - `scaling` =
    - `fc`: adaptation of the spectral irradiance to the CMIP6 *historical* forcing
    - `nfc`: no re-scaling

Data format

Be careful when working with the time axis, because the *float year values* do not really follow time axis and calendar conventions!

The original data files are provided in simple text format, and we also provide the data in netCDF format. The text files' structure is as follows:

- 1st array: **wavelength array** in [nm], listing the center of each wavelength bin `[ 115.5, 116.5, 117.5, 118.5, 119.5, […] 100000.0078125 , 120000.0234375 , 140000.015625 , 160000.015625 ]`
- 2nd array: **wavelength bin** in [nm], listing the bin width of each wavelength bin `[ 1., 1., 1., 1. […] 40., 5010., 14990.00488281, 20000.00585938, 20000., 20000., 20000.00585938, 20000.00585938, 20000., 20000. ]`
- 3rd array: **time** in [year] (floating numbers)
  - 69235 time steps for 14C: `[-6754.5, -6753.5, -6752.5, -6751.5, […] 2015.98632812, 2015.98901367, 2015.99182129, 2015.99450684, 2015.99731445]`
  - 61595 time steps for 10Be: `[885.5, 886.5, 887.5, 888.5, […] 2015.986, 2015.989, 2015.992, 2015.995, 2015.997]`
- 4th array: **SSI reconstruction** in [W m-2 nm-1]. SSI is average SSI in corresponding bin.

We provide the following IDL and python code to read the `.txt` file and calculate TSI:

```
;=======================================================
;N=69235 for 14C reconstruction
;N=61595 for 10Be reconstruction
header=strarr(12)
satire_wl=dblarr(1070)
satire_dwl=dblarr(1070)
time=dblarr(N)
SSI=dblarr(1070,N)

openr,1,filename
 readf,1,header
 readf,1,satire_wl
 readf,1,satire_dwl
 readf,1,time
 readf,1,SSI
close,1

TSI=dblarr(N)
FOR i=0L,N-1 DO TSI[i]=TOTAL(satire_dwl*SSI[*,i])
;=======================================================
```

- The following python code shows how to deal with the original compressed text data. You can also check the full python script that was used to generate the netCDF files

```
# Get directly the data from the bz2 compressed file
file_in = bz2.BZ2File(input_full_path)

# Print the comments at the beginning of the file
print '\nData file header:'
for skip_header in range(nb_header_lines):
  hdr = file_in.readline().strip()
  print 'HEADER =>', hdr

# Get the wavelength data (on a single line) and store it in a
numpy
# array
wl_str = file_in.readline()
wl = np.array(map(float, wl_str.strip().split()),
dtype=np.float32)

# Get the wavelength bins data (on a single line) and store it in
a
# numpy array
wl_bin_str = file_in.readline()
wl_bin = np.array(map(float, wl_bin_str.strip().split()),
dtype=np.float32)

# Get the years (on a single line) and store them in a numpy array
year_str = file_in.readline()
year = np.array(map(float, year_str.strip().split()),
```

```
dtype=np.float32)

# The rest of the data is the ssi data, with one line for each
time
# steps! Read it in an array
ssi = np.loadtxt(file_in, dtype=np.float32)

file_in.close()

print '\nSize of the SSI matrix =', ssi.shape

# Compute the TSI
tsi = np.dot(ssi, wl_bin)
```

- The following python example shows how to determine the indices of specific years, before Jan 1850 1st and after. Once you know the indices of Jan 1st and Dec 31st, and if a year is a leap year or not

```
 # 1 value per day, AFTER (and including) Jan 1st 1850
>>> Jan_01_1850_idx = np.argwhere(year == 1850)[0, 0]
>>> Jan_01_1851_idx = np.argwhere(year == 1851)[0, 0]
>>> Jan_01_1850_idx, Jan_01_1851_idx
(8605, 8970)
>>> Jan_01_1851_idx - Jan_01_1850_idx
365
>>> Dec_31_1850 = Jan_01_1851_idx - 1
>>> year[Jan_01_1850_idx], year[Dec_31_1850_idx],
year[Jan_01_1851_idx]
(1850.0, 1850.9973, 1851.0)

# 1 value per year, strictly BEFORE Jan 1st 1850
>>> year_50_idx = np.argwhere(year == 50.5)[0, 0] # DO NOT FORGET
to use 'NNN.5' for the year value
>>> year_50_idx, year[year_50_idx]
(6805, 50.5)
```

- You can use the following if you want to do a weighted average over the time axis

```
>>> time_weights = np.ones((69235,))
>>> np.argwhere(year == 1850)
array([[8605]])
>>> tt[8605]
1850.0
>>> year[8604]
1849.5
# Note: it would be nicer to assign 366 to leap years below...
>>> time_weights[:8605] = 365 # Assign 365 to all time steps up to
(but excluding) step 8605
>>> time_weights[8600:8620]
array([ 365.,  365.,  365.,  365.,  365.,    1.,    1.,    1.,
1.,
          1.,    1.,    1.,    1.,    1.,    1.,    1.,    1.,
```

```
1.,
          1.,    1.])
>>> ssi_average_weighted = np.average(ssi, axis=0,
weights=time_weights)
```

References

- Baroni, M., and ASTER Team (2015), **A new 10Be record recovered from an Antarctic ice core: validity and limitations to record the solar activity**, Geophysical Research Abstracts 17, EGU2015-6357
- Vieira, L.E.A. et al. (2011), **Evolution of the solar irradiance during the Holocene**, Astron. Astroph., 531, A6, doi:10.1051/0004-6361/201015843
- Usoskin, I.G. et al. (2014), **Evidence for distinct modes of solar activity**, Astron. Astrophys., 562, L10, doi:10.1051/004-6361/201423391
- Usoskin, I.G. et al. (2016), **Solar activity during the Holocene: the Hallstatt cycle and its consequence for grand minima and maxima**, Astron. Astroph., 587, A150, doi:10.1051/0004-6361/201527295

Download

Once you have downloaded the compressed text data file, you can use the command `bzmore` to have a quick look at its content without having to uncompress it!

```
 > bzmore
SSI_14C_cycle_yearly_cmip_v20160613_fc.txt.bz
2
------>
SSI_14C_cycle_yearly_cmip_v20160613_fc.txt.bz
2 <------
;
;Solar Spectral Irradiance for last 9
millennia (added solar cycle), 6754.5 BC ~
 12.31.2015
;
;File structure
;1. wavelength: array[1070],nm
;2. wavelength bin: array[1070],nm
;3. year: array[69235], floating number
;   6754.5 BC~1849.5 AD, yearly cadence
;   1,1,1850 ~ 12,31,2015, daily cadence
;4. SSI(wavelength,date): array[1070,69235],
W m-2 nm-1
;
; Note: SSI adjusted to CMIP
     115.500     116.500     117.500
```

```
118.500     119.500     120.500     121
.500     122.500     123.500 [...]
```

You will find below a table with all the available data files, and their md5sum checksum (if you want to check that you download was OK, you can just type `md5sum file.nc` and compare the result to what is displayed in the table).

If you want to download a file, click on the PMIP4 SATIRE-M solar forcing data download link and then on the file you need. The files are currently protected by a password. Get in touch with Jean-Yves Peterschmitt if you need to access them.

md5sum output	**Data file**	**Size**
636519aa89b472a04748893d6f3ff1b3	**SSI_14C_cycle_yearly_cmip_v20160613_fc.txt.bz2** recommended forcing for the PMIP4-CMIP6 *tier-1* past1000 experiment	506 Mb
af4d1c36647f094b38fd6d9bc8e6617b	**SSI_14C_cycle_yearly_cmip_v20160613_fc.nc** recommended forcing for the PMIP4-CMIP6 *tier-1* past1000 experiment	283 Mb
86bae35d2cd4d7f2c3dde9ee567d5a87	SSI_14C_cycle_yearly_cmip_v20160613_nfc.txt.bz2	506 Mb
b5639df4fab6cac7d55972d7a066e2b9	SSI_14C_cycle_yearly_cmip_v20160613_nfc.nc	283 Mb
96cdeb6a561f0be6b83e1b45a809f8ad	SSI_10Be_cycle_yearly_cmip_v20160613_fc.txt.bz2	450 Mb
1f9075a93e58173281ee11731bdb97e5	SSI_10Be_cycle_yearly_cmip_v20160613_fc.nc	252 Mb
0516a6a073c25365674a004034392130	SSI_10Be_cycle_yearly_cmip_v20160613_nfc.txt.bz2	450 Mb
c2922684c81e839fb07de7017ed0d1f3	SSI_10Be_cycle_yearly_cmip_v20160613_nfc.nc	252 Mb

From:
https://pmip4.lsce.ipsl.fr/ - **PMIP4**

Permanent link:
https://pmip4.lsce.ipsl.fr/doku.php/data:solar_satire?rev=1470753753

Last update: **2016/08/09 16:42**