Using Python language for analysing measurements from SABER instrument on TIMED satellite

P. Hoffmann, Ch. Jacobi, S. Gimeno-Garcia

Summary

The practical handling and analysis of satellite data is outlined using the programming language Python. The limb sounding technique of the SABER instrument on board of the TIMED satellite delivers vertical profiles of kinematic temperature from the stratosphere (∼30 km) up to the lower thermosphere (∼120 km).
The procedure may be summarised as follow: In the first step the level 2 data for one month are extracted from the netCDF format and arranged into a new altitude-latitude grid for the ascending and descending orbits. The longitudinal structure is rearranged applying the decomposition into zonal harmonics. Various cross sections of the data give a good overview of the thermal structure and dynamics of the atmosphere up to 120 km. The monthly values of the zonal averaged temperature are compared to the available data from stratospheric reanalyses up to 60 km as well as the initialized background climatology of general circulation models for the middle atmosphere.

Zusammenfassung

In diesem Artikel soll der praktische Umgang mit Satellitendaten und deren Auswertung unter Verwendung der Programmiersprache Python skizziert werden. Auf der Basis der Horizontsondierungen des SABER Instruments auf dem TIMED Satelliten werden vertikale Profile wie die kinetischen Temperatur von der Stratosphäre (∼30 km) bis zur unteren Thermosphäre (∼120 km) gewonnen.
Die Arbeitsschritte bei der Analyse lassen sich wie folgt gliedern: Als erstes werden die Level 2 Produkte eines Monats aus dem netCDF Format extrahiert und an ein neues Höhen-Breiten Gitter für jeden auf- und absteigenden Orbit angepasst. Die Längenstruktur wird mit Hilfe einer Zerlegung in harmonische Funktionen regularisiert. Diverse Querschnitte der Daten geben ein guten Überblick über die thermischen Struktur und Dynamik der Atmosphäre bis 120 km. Die Monatswerte des Zonalmittels der Temperatur werden mit denen aus operationellen Reanalysedaten (∼60 km) sowie der Hintergrundklimatologie von Zirkulationsmodellen der mittleren Atmosphäre verglichen.

1. Introduction

1.1 Motivation

The knowledge of a modern interpreted progamming language is necessary in todays science. These offer many useful numerical and visualisation facilities for analysing atmospheric data. The Interactive Data Language (IDL) and Matlab are well-known commercial environments of this kind. A freely available language is the Python interpreter and the extensive standard library can be downloaded for all major platforms from the Python Web site, *http://www.python.org/*.

It’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in different fields (Perez et al. 2007). The interactive environment of Python is performed here showing an example of application for middle atmosphere analysing satellite measurements from SABER instrument on the TIMED satellite. Several useful python modules are imported to extract and prepare data calculating daily averaged temperature fields and zonal harmonics. These steps are necessary to obtain information of planetary waves, tides and also gravity waves from satellite measurements. Some extracts from the main Python script *saber.py* declared as *Algorithm* are given in the following sections.

1.2 Getting Started with Python

The Python programming language and several useful modules are installed on the Linux Compute-Server (passat) of the Leipzig Institute for Meteorology (LIM).
The following statements given on Algorithm 1 describe the first steps starting with the interactive nature of Python, how to import the numerical module *numpy* and running Python programs:

Algorithm 1 *First steps for getting started with Python. ($\curvearrowright$ typing enter button, $\rightleftarrows$ typing tabulator button)*

```
> ipython ↷                        # starting Python interpreter
In[1]:                             # Python prompt
In[1]:  import ⇄                   # lists all callable modules
In[2]:  import numpy as N          # imports the numerical module numpy
In[3]:  N.⇄                        # shows all numerical functions
In[4]:  N.zeros?  ↷                # zeros(shape, dtype=float, order='C')
In[5]:  a = N.zeros(10,float)      # generating an array with zeros as elements
> python saber.py ↷.               # running a python script
> pydoc -w saber.py ↷              # generating a html- documentation
> pydoc -g ↷                       # open the python documentation server
```

The language can be started using *python* for executing python scripts and for interactive typing *ipython* which features a lot of preferences e.g. viewing at data, testing new ideas, combining algorithms and directly evaluating of results.
The structure of more complex programs should be divided into *class* and *functions* as given in Algorithm 2. The Python script *saber.py* includes one *class* and several *functions* starting with the *def* statement. At the beginning of each program all necessary modules for numerical operations *numpy* and visualisation *matplotlib* have to be imported. The function *__init__()* is a kind of main program where all other functions are step-wise called such as for reading filelist, reading the scientific data format netCDF, for harmonic analysis and for plotting results. The *self* statement, see Algorithm 2, is used to define global variable (a) as follow: *self.a* .
Two more application of the Python language are given in appendix on Fig.8 and Fig.9 showing at first a generated html-documentation of the main Python script *saber.py* using *pydoc*. This document lists all *Modules*, *Functions* and *Data* which are used as well as *Comments* which are written in the program. The second example presents a user interface for analysing different atmospheric data based on *wxpython.*

Algorithm 2 *Programm structure of the Python script saber.py including functions for reading netCDF data, analysis and visualisation.*

```
class saber(object):
      def __init__(self):
      def reading_filelist(self):
      def reading_netcdf(self):
      def zonal_harmonics(self):
      def plotting_orbits(self):
      def plotting_latlev(self):
saber = saber()
```

2. SABER on TIMED Satellite

The TIMED satellite was launched on 7. December 2001 into a 625 km orbit of 74.1° inclination to investigate the Thermosphere, Ionosphere, Mesosphere and Energetics Dynamics. The SABER instrument on board of the spacecraft began making observations in late January 2002 (Mlynczak, 1997). By step-scanning the atmospheric limb SABER measures height profiles of temperatures and selected chemical species from 10-180 km altitude with a horizontal resolution along track of about 400 km. The multispectral radiometer operates in the near to mid-infrared over the range 1.27 µm to 17 µm (7865 cm-1 to 650 cm-1). It measures CO_2 infrared limb radiance from approximately 20-120 km altitude and the kinetic temperature profiles are retrieved over this heights using a full non-LTE inversion (Mertens et al. 2004). The used SABER L2A data (version 1.07) were downloaded from the web site: *http://saber.gats-inc.com* including all observed geophysical parameter.
The SABER latitude coverage extends from about 52° of one hemisphere to 83° of the other. This latitude range turn back after a 60-day period while the satellite change its orientation.
For investigating dynamics at mesosphere/lower thermosphere (MLT) region atmospheric tides are the most dominate features. Tides are global-scale waves (e.g., in temperature, winds, and density) with periods that are subharmonics of a solar day with 24 (diurnal) and 12 (semidiurnal) hours corresponding to the longitudinal structure of the first two zonal harmonics. These can be of migrating and nonmigrating nature. Migrating tides propagate westward with the apparent motion of the sun. In contrast to that, nonmigrating tides are non-sun-synchronous components. They can propagate westward, eastward, or remain standing with any zonal wavenumbers except for those from the migrating tides. Thus, tidal waves play an important role in the interpretation of satellite measurements at MLT region (Oberheide et al., 2003).
A new method for analysis of satellite data is presented in Pancheva et al. (2008). They use derived temperature fields from SABER to extract the migrating and nonmigrating tidal components as well as all stationary and travelling planetary waves simultaneously. The comparison between the altitude and latitude structure of the SABER and UKMO planetary waves of the northern hemispheric stratosphere indicates a good agreement.

2.1 Reading Netcdf Data

The SABER L2A data are available per orbit in the *Network Common Data Form* (netCDF) of about 10 MByte size. This data format is self-describing and portable including information of defining data. Special libraries are necessary to extract the content. A monthly data coverage is obtained processing 30×14 orbits in the batch mode producing a list of all file names typing: `> ls -a *.nc > filelist.txt`. The extraction of the netCDF data using Python is described

Variable(dimensions)	units	Long name
Event(event)		event number for current file
date(event)	yyyyddd	date [yyyyddd]
time	msec	time since midnight
earth_sun(event)	km	earth-sun distance
tpDN(event)		0=day 1=night
tpAD(event)		0=ascending 1=descending
tpSolarZen(event)	degrees	tangent-point solar-zenith angle
tplatitude(event,altitude)	degrees	tangent-point latitude
tpaltitude(event, altitude)	km	tangent-point altitude
tplongitude(event, altitude)	degrees	tangent-point longitude
tpSolarLT	msec	tangent-point local solar time
sclatitude(event,altitude)	degrees	spacecraft latitude
scaltitude(event, altitude)	km	spacecraft altitude
sclongitude(event, altitude)	degrees	spacecraft longitude

Table 2: *Geolocation data from the SABER measurements. The statements in brackets give the dimension of the data array.*

in Algorithm 3 importing the module *Scientific.IO.NetCDF*. The list of variables can be printed out enter: `In[6]:  ncfile.variables` ↷.
The level 2A data contains geolocation listed on Tab.2 and retrieval products listed on Tab.5. The variables that are dimensioned use the integer: *Altitude*, *Event*. The *Altitude* dimension has 400 elements to cover the maximum altitude range, but not all 400 elements will have data. The *Event* dimension depends on the number of events in the netCDF file. The geolocation data give information about geometry between sun-earth-satellite system. The spacecraft location is declared in geografic coordinates (*sclongitude, sclatitude, scaltitude*). The tangent point coordinates for the limb sounding observation technique are indicated with the prefix (*tp*). A reconstructed picture for one point in time is shown on Fig.1 (left panel). The solar zenith angle at the tangent point (*tpSolarZen*) indicates the angle between the source of solar radiation and the observation point. The variable *tpAD* is later used in this section for separating orbits into ascending (*tpAD=0*) and descending (*tp_AD=1*) data.
The level 2A products of essential meteorological parameter (e.g. T, ρ, p) and concentration of atmospheric species (CO_2,O_3, NO, H_2O) retrieved from SABER instument are listed on Tab.5. The right panel on Fig.1 represents one vertical profile of temperature at 26.6°N on 1/1/2005 retrieved by the limb sounding technique.

Algorithm 3 *Excluding SABER L2A data with variable name 'ktemp' from netcdf file and converting as a numpy array.*

```
In[1]:  from Scientific.IO.NetCDF import NetCDFFile as Dataset
In[2]:  import numpy as N

In[3]:  ncfile = NetCDFFile('filename.nc','r')
In[4]:  tem = ncfile.variables['ktemp']
In[5]:  tem = tem.getValue()
In[6]:  tem = N.array(tem,float)
```

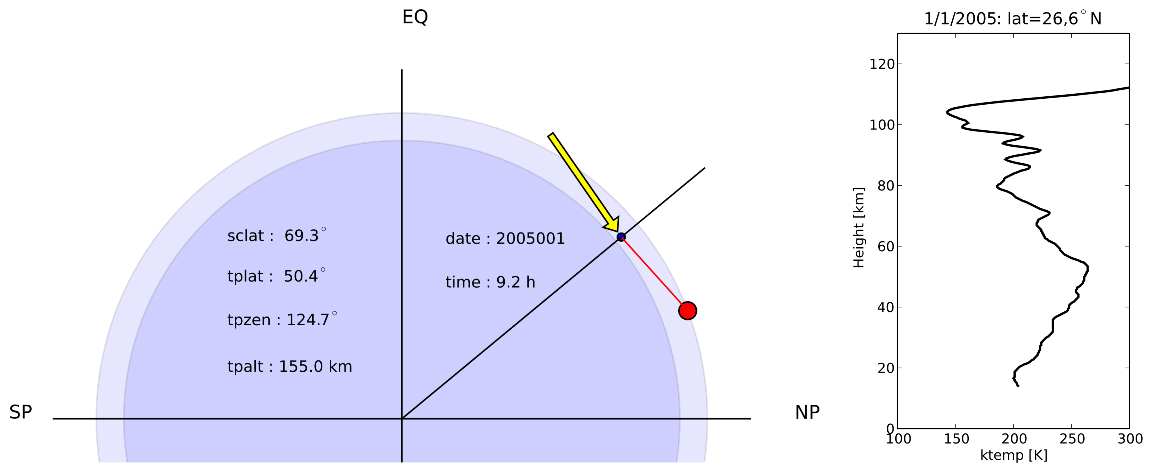


Figure 1: *left panel: The geometry of the tangent point at 50.4°N and 9.2 hours UT is reconstructed using the geolocation data which are included in the L2 netCDF file: (spacecraft: big circle, tangent point: small circle, arrow: solar ray path). right panel: Single vertical profile of temperature at 26.6°N on 1/1/2005.*

Variable(dimensions)	**units**	**Long name**
ktemp(event, altitude)	K	kinetic temperature (merge)
density(event, altitude)	1/cm3	atmospheric density
pressure(event, altitude)	mbar	pressure (merge)
tpgpaltitude(event, altitude)	km	tangent-point geopot. altitude
O3_96(event, altitude)	mixing ratio	O3 mixing ratio 9.6μm (merge)
H2O(event, altitude)	mixing ratio	H2O mixing ratio (merge)
CO2(event, altitude)	mixing ratio	CO2 mixing ratio

Table 5: *Derived level 2 products of meteorological parameters and selected chemical species. The statements in brackets give the dimension of the data array.*

2.2 Satellite Orbits

The daily global projection of satellite orbits separated into ascending (asc) and descending (dsc) nodes is presented in Fig.2 for the 1st January 2005. Ascending (open circles) orbit nodes are the instrument footprints when the satellite moves from south to north and descending (solid circles) orbit nodes are the footprints for north-south movement. Alltogether 14 orbits covering the globe having a longitudinal resolution of about 25°. The Python script for mapping the TIMED satellite orbits is given on Algorithm 4 using the *matplotlib toolkits basemap*. Every circle includes about 98 events of 400 altitude elements and is defined in time by the *universal time* (t_{UT}) and the *solar local time* (t_{LT}). The daily t_{LT} variation for a given orbit node and latitude band is about 12 minutes and the difference between the maximum asc/dsc t_{LT} at equator amounts to 9 hours.

Algorithm 4 *Global projection of satellite orbits using the scatter plot function.*

```
In[1]:  from pylab import *
In[2]:  from mpl_toolkits.basemap import Basemap,shiftgrid

In[3]:  m = Basemap(projection='moll',resolution='c',area_thresh=10000.,/
lat_0=30.,lon_0=0.)
In[4]:  x,y = m(*meshgrid(self.lon,self.lat))
In[5]:  m.scatter(x[:,0],y[:,0],c='w')
In[6]:  m.drawcoastlines()
In[7]:  m.drawmapboundary()
In[8]:  title('SABER on TIMED (tangent points):  1/1/2005')
In[9]:  savefig('orbits.pdf')
```

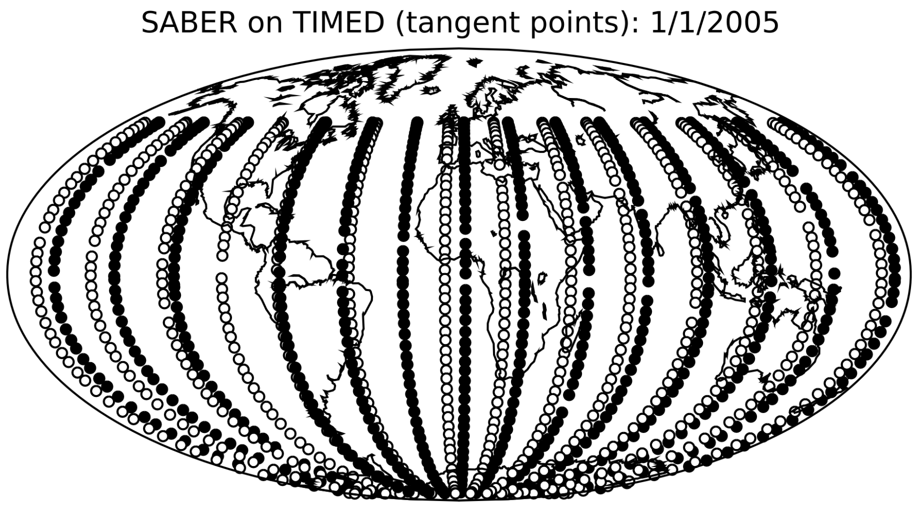


Figure 2: *Mollweide projection of the tangent points for each satellite orbit on 1/1/2005 (right panel): day-time (open circles) and night-time (solid circles) measurements.*

2.3 Daily Projection of Zonal Mean Temperature Profils

The irregular daily SABER observations at tangent points for the 1st January 2005 are arranged to a new regular height-latitude (*lev, lat*) grid for the ascending and descending orbits separately, as shown (Algorithm 4). The latitude dimension (*ny*) is divided into 10 bins of 10° starting from -45° to 45° as the central point. The new vertical dimension (*nz*) has 50 layers of 2 km ranges from 31 to 129 km. The *numpy* function *masked_where* is applied to regrid the data by masking values which lie outside of the given range and to mask out the invalid values (-999). All data within one grid of 10° and 2 km are averaged. This procedure is applied for all orbits of day defined by the dimension (*no*).

From the new gridded data a daily height-latitude picture of the zonal mean temperature ($\overline{T}_{asc}$) and the zonal standard deviation (T'_{asc}) for the ascending data are calculated obtaining the thermal structure and dynamics of the atmosphere up to the lower thermosphere (120 km). The left panel on Fig.3 reveals the warm stratopause (50 km, 270 K) and cold mesopause (85 km, 180 K) on the summer hemisphere (-45°) as well as the transition to the thermosphere. On the right panel, the standard deviation of the longitudinal temperature variation displays some

Algorithm 5 *Making a latitude-altitude grid of temperature data for the ascending orbits with horizontal bins of 10°.*

```
In[1]:  no = 14     #number of orbits per day
In[2]:  ny = 10; dy = 10.
In[3]:  nz = 50; dz = 2.

In[4]:  lat = N.arange(-45.,-45.+ny*dy,dy,float)
In[5]:  lev = N.arange(31.,31.+nz*dz,dz,float)
In[6]:  tem = N.zeros((nz,ny,no),float)

In[7]:  for o in range(no):
......        for z in range(nz):
......              for y in range(ny):
......                    dum = N.ma.masked_where(la<(lat[y]-5.0),asc)
......                    dum = N.ma.masked_where(la>(lat[y]+5.0),dum)
......                    dum = N.ma.masked_where(le<(lev[z]-1.0),dum)
......                    dum = N.ma.masked_where(le>(lev[z]+1.0),dum)
......                    dum = N.ma.masked_where(dum<-999,dum)
......                    dum = dum.compressed()
......                    temp[z,y,o] = mean(dum)
```

dynamical aspects caused by tidal waves above 80 km and in the winter hemisphere (+45°), also at lower altitudes, the latter caused by planetary waves activity, especially the stationary part.

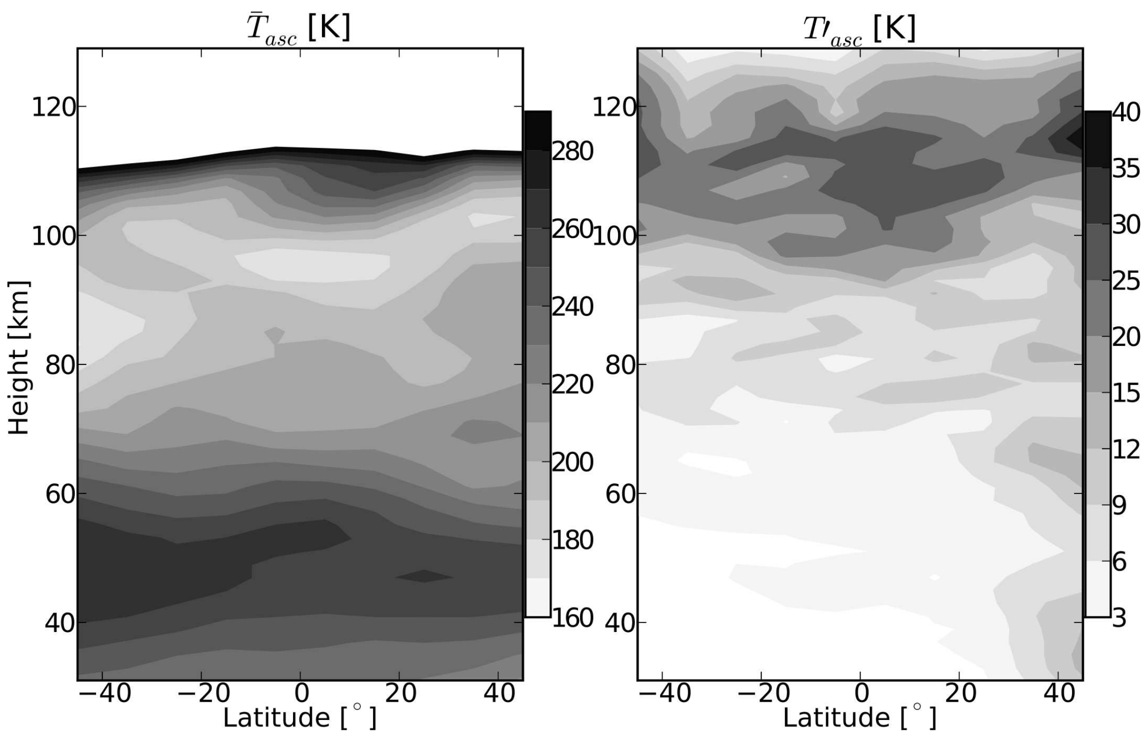


Figure 3: *Height-latitude cross-section of the zonal mean temperature (left) and zonal standard deviation (right) on 1/1/2005. (Only the ascending data are used!)*

2.4 Harmonic Decomposition

The irregular longitude samples are rearranged using the *tangent point longitude* (x_{tp}) information for sorting the ascending (asc) and descending (dsc) satellite orbit nodes. The decomposition of the ascending temperature variation (T_{asc}) along longitude into the first three zonal harmonics ($k = 1,2,3$) and the zonal mean ($k = 0$)

$$T_{asc}(z,y,x_{tp}) = A_0(z,y) + \sum_{k=1}^{3} A_k(z,y) \cdot cos[k \cdot x_{tp} - \phi_k(z,y)] \tag{1}$$

is applied using least-squares approximation. This inverse problem is solved by singular value decomposition (SVD). This method is given in a Fortran subroutine *zonal.f* which is translated into a Python module using: `> f2py -m -c zonal zonal.f`. This *self-made* library *zonal.so* can be imported into the Python program. To this aim, simply some additional information about *in/output arrays* must be append in the Fortran code, for example: *cf2py intent(in): a*. These Python statements are regarded as comments in the Fortran syntax.
Several pictures of temperature disturbances give an overview of the dynamics of the middle atmosphere. A height-latitude image of temperature amplitudes for the first $A_{asc}(k = 1)$ and second $A_{asc}(k = 2)$ zonal harmonics is depicted on Fig.4. The left panel shows the amplitude of the diurnal tide at higher altitudes. Though, the westward moving diurnal migrating tide cannot be observed from sun-synchron satellite measurements due to its similar phase speed. The observed tidal signal must be of non-migrating nature. At lower altitudes the slower planetary waves with periods of several days and zonal wavenumber 1 are dominant in the winter hemisphere. The right panel shows the signature of tidal and planetary waves having zonal wavenumber 2. The picture of the mesosphere/lower thermosphere dynamics is somewhat inaccurate due to the tidal effects which cannot be exactly analysed from orbiting satellite observations.

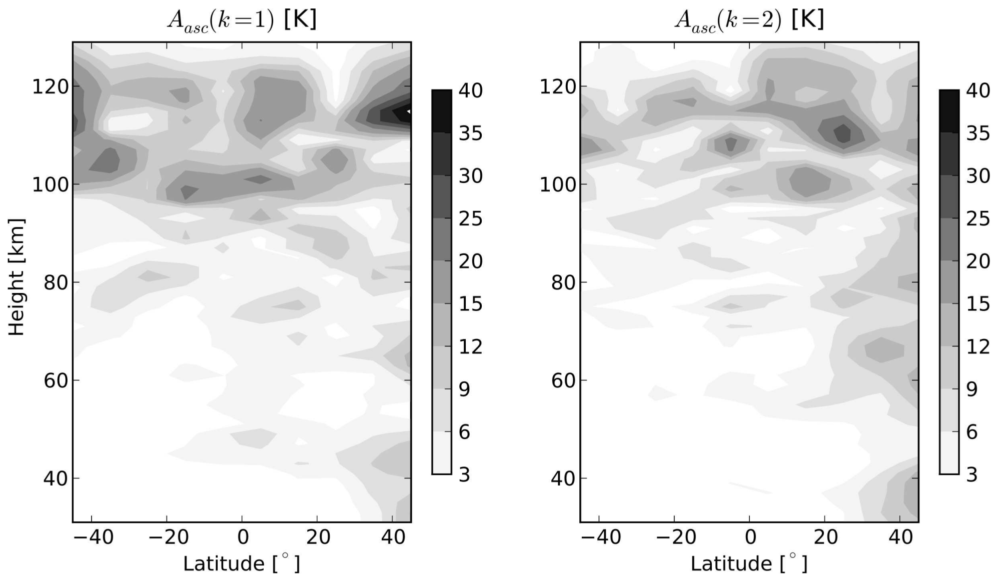


Figure 4: *Height-latitude cross-section of zonal wavenumber amplitudes for k=1 (left) and k=2 (right) on 1/1/2005.*

The reconstruction of the longitudinal structure using amplitude and phase from the first two zonal harmonics is applied producing height-longitude plots in Fig.5 at mid-latitudes (left panel) and at equator (right panel). These depict the vertical structure of the zonal harmonics. The signatures of planetary waves at stratospheric height are weak in contrast to the diurnal and semidiurnal tidal waves. At 80 km the zonal wavenumber 1 increases. Above 90 km the zonal wavenumber 2 is dominant which means that two maxima and minima are visible in the zonal wave structure. This seems to correspond to the behaviour of the diurnal and semidiurnal tides. The semidiurnal component is the dominant feature at MLT region. A view of the horizontal structure of the temperature disturbances at several heights (51 km, 71 km, 91 km, 111 km) depicted on Fig.6 in a latitude-longitude cross-section shows a similar behaviour. At stratospheric heights (51 km) the upward propagating planetary waves in the winter hemisphere (+45°) are dominant having amplitudes of about 9 K. At mesophere (71 km) the planerary wave activity decreases and the equatorial forced tidal waves become important. In the MLT region the amplitudes of tides can amount more than 20 K.

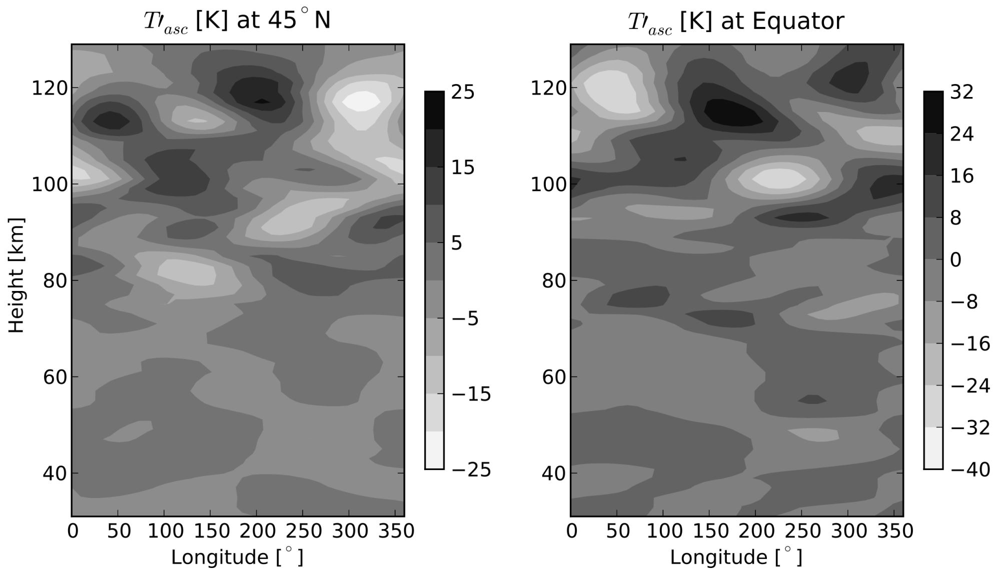


Figure 5: *Longitude-height cross-section of the temperature disturbances at about 45°N (left) and near the equator (right) on 1/1/2005.*

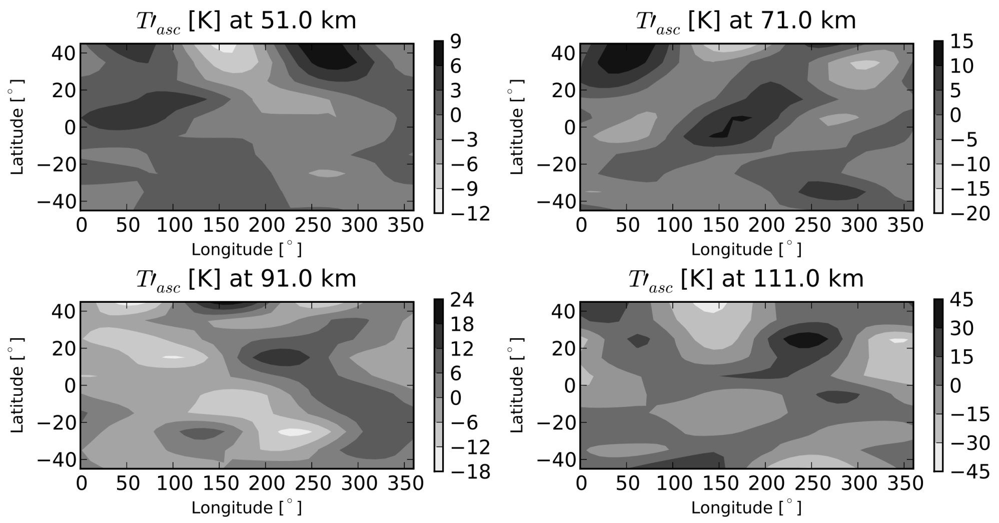


Figure 6: *Horizontal contour plots of temperature disturbances at 51 km, 71 km, 91 km and 111 km height on 1/1/2005.*

2.5 Monthly Mean Data

The temperature data derived from the SABER instrument for January 2005 are used to obtain a monthly mean picture, see Fig.7 (black contours). In comparison to that the greyscaling presents the background climatology for January covering an altitude range from the troposphere to the lower thermosphere generated by the general circulation model COMMA. The white contour lines show the January mean of 2005 taken from the UK Met Office stratospheric reanalysis data. These are regularly produced up to 0.1 hPa (about 60 km). The log-pressure height used as the vertical coordinate of the COMMA model was interpolated into geometric height. This approves a comparison between SABER and COMMA of the mesopause region height and temperature. Small differences can be detected, because the circulation model uses climatological data e.g. surface temperature and stationary planetary waves to generate the background wind and temperature fields. The SABER temperature data are only averaged over the January 2005. Thus, there are some variances of the summer mesopause region height (< 5 km) and temperature (< 10 km).

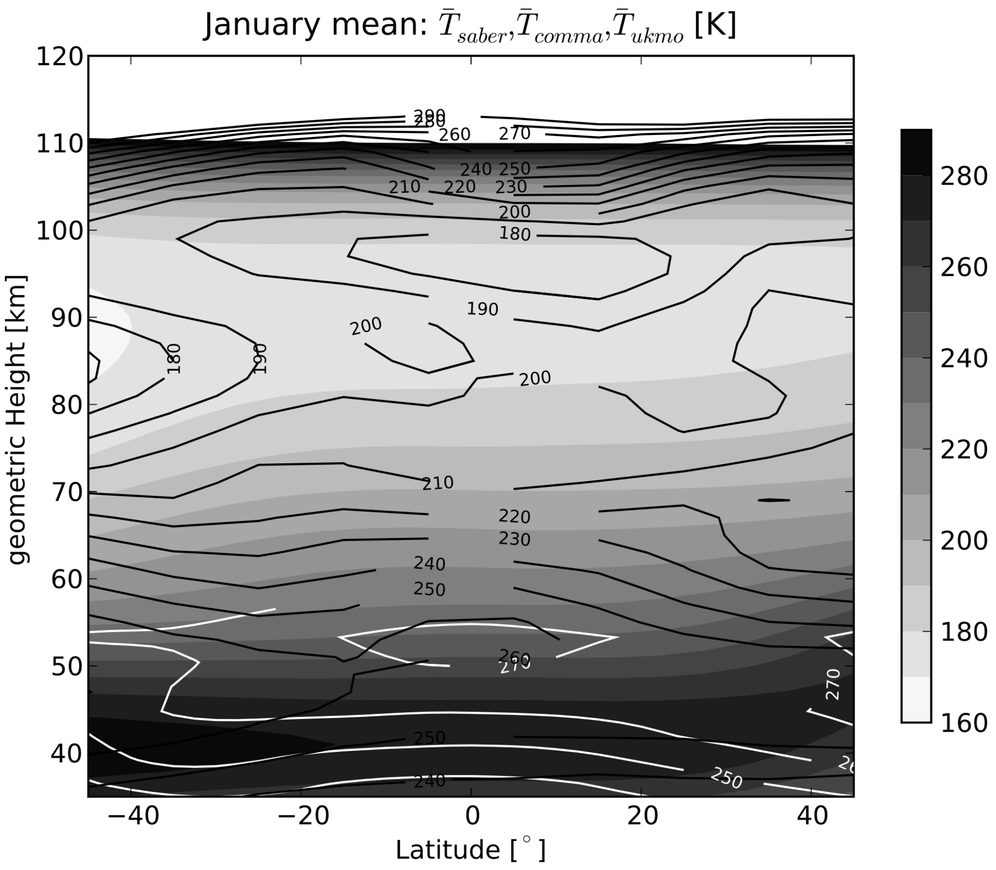


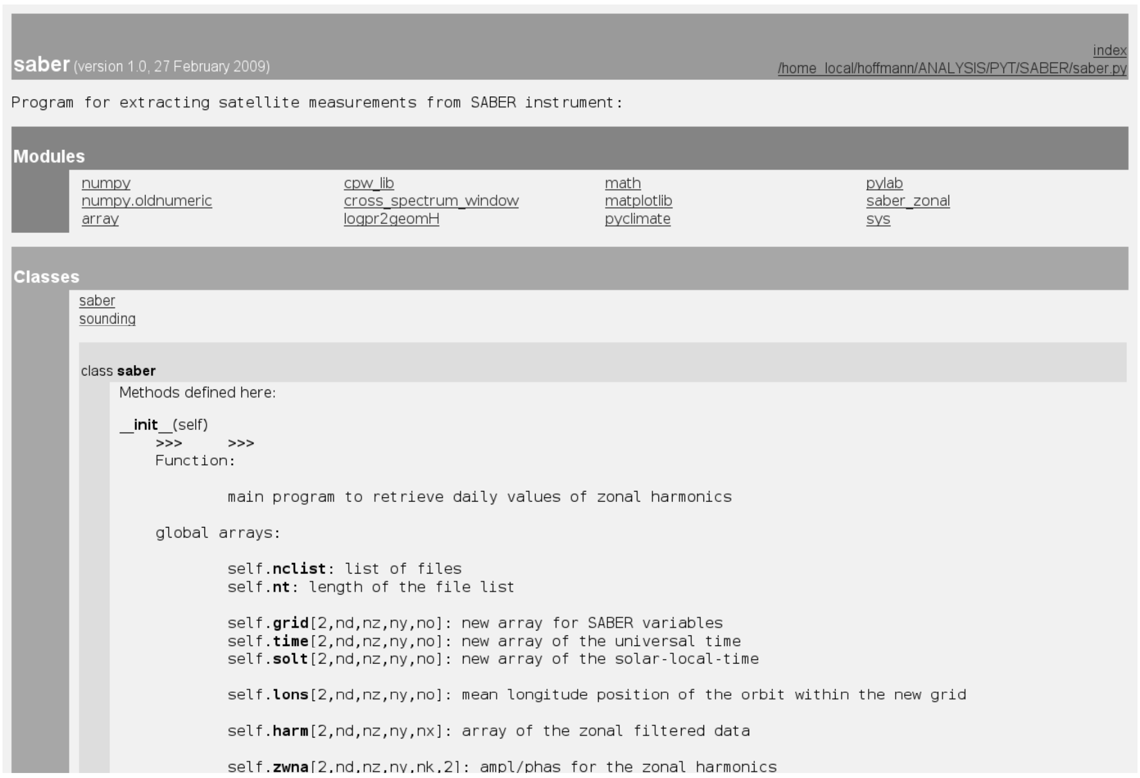
Figure 7: *Height-latitude cross-section of monthly mean temperature data: shaded: COMMA (Jan), black contour: UKMO (Jan05), white contour: SABER (Jan05)*

3. Discussion and Outlook

The SABER limb sounding of the stratosphere-mesosphere-lower thermosphere on board the TIMED satellite delivers temperature profiles covering the latitude region of 50°N to 50°S. The analysis of these data was presented in this work using the programming language Python.

The procedure performed here shows the preliminary steps for investigating satellite measurements with respect to global scale waves (e.g. planetary waves, tides) by producing daily maps of zonal averaged temperature profils and its superposed harmonics. The monthly mean backgound temperature fields for January 2005 was calulated in comparison to the middle atmosphere model background temperature and to stratospheric reanalysed temperature data from UK Met Office (UKMO) model up to 60 km.
In the future, the SABER data may be assimilated into a mechanistic models (e.g., the MUAM, Pogoreltsev et al., 2007) for generating an extended reanalyses up to the lower thermosphere. The planetary wave analysis from UKMO could delivers the spectra of wave signals, which can be forced in the models.
Information about gravity wave activity from satellite measurements can be derived filtering the vertical wavelength (λ_z < 10 km) for each temperature profile. The application of the procedure presented here can be used to illustrate gravity wave fields and to study modulation effects by planetary waves.

Appendix



saber (version 1.0, 27 February 2009)

index
/home_local/hoffmann/ANALYSIS/PYT/SABER/saber.py

Program for extracting satellite measurements from SABER instrument:

Modules

numpy	cpw_lib	math	pylab
numpy.oldnumeric	cross_spectrum_window	matplotlib	saber_zonal
array	logpr2geomH	pyclimate	sys

Classes

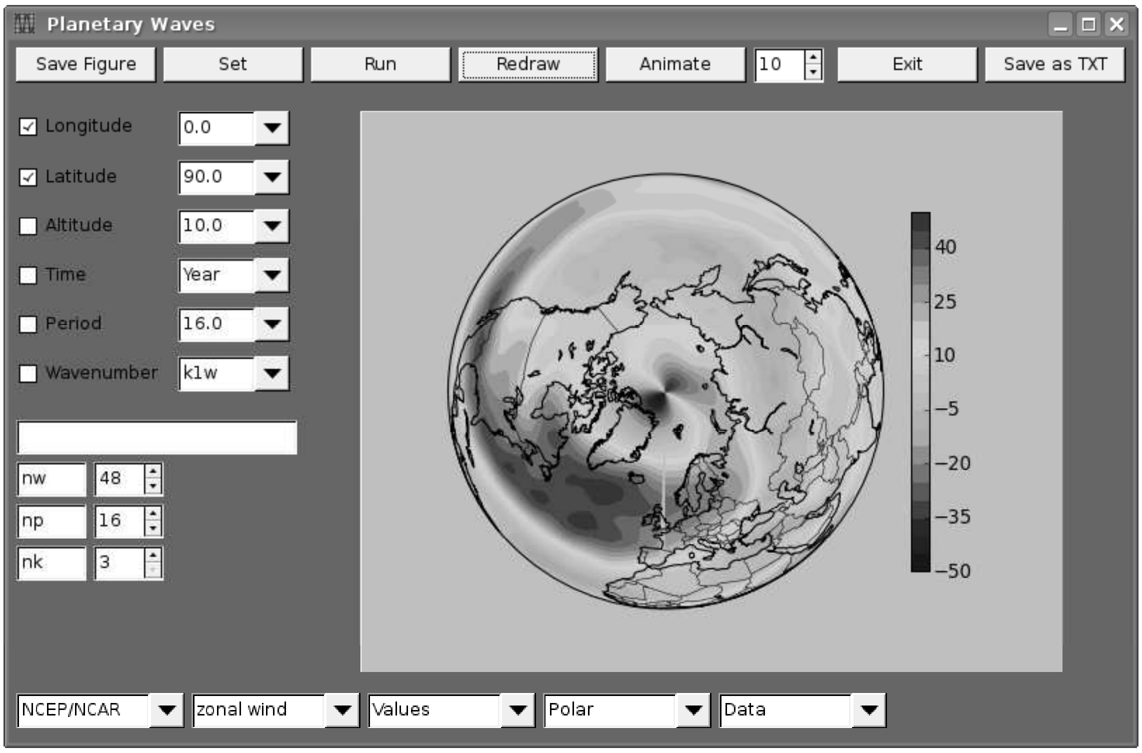
saber
sounding

class **saber**
Methods defined here:

__init__(self)

```
>>>     >>>
Function:

        main program to retrieve daily values of zonal harmonics

global arrays:

        self.nclist: list of files
        self.nt: length of the file list

        self.grid[2,nd,nz,ny,no]: new array for SABER variables
        self.time[2,nd,nz,ny,no]: new array of the universal time
        self.solt[2,nd,nz,ny,no]: new array of the solar-local-time

        self.lons[2,nd,nz,ny,no]: mean longitude position of the orbit within the new grid

        self.harm[2,nd,nz,ny,nx]: array of the zonal filtered data

        self.zwna[2,nd,nz,ny,nk,2]: ampl/phas for the zonal harmonics
```

Figure 8: *Snapshot of the Python-script documentation which is automatically generated using* pydoc. *The HTML-document includes the directory, used modules and functions as well the defined data arrays. All comments appear in this documentation for describing variables and functions.*



Animate 10

☑ Longitude 0.0
☑ Latitude 90.0
☐ Altitude 10.0
☐ Time Year
☐ Period 16.0
☐ Wavenumber k1w

nw 48
np 16
nk 3

NCEP/NCAR | zonal wind | Values | Polar | Data

Figure 9: *Snapshot of a Python generated interface using the module* wx. *This user interface is build for investigation of the middle- and upper atmosphere dynamics. There are many options for selecting data base (e.g. UKMO), analysing planetary waves and long-term trends.*

Aknowledgements

This work is supported by DFG under grand JA 836/22-1. The authors thanks to the NASA organization providing the SABER data and the Python community for using the open-source software tools as well Falk Kaiser for his techniqual support.

References

[1] Pancheva, D., Mukhtarov, P. , Andonov, P., Mitchell, N. J., Forbes, J. M., 2008, Planetary waves observed by TIMED/SABER in coupling the stratosphere-mesosphere-lower thermosphere during the winter of 2003/2004: Part 1-Comparison with the UKMO temperature results, J. Atm. Sol.-Terr. Phys., in press.

[2] Mertens, C. J., et al., 2004, SABER observations of mesospheric temperatures and comparisons with falling sphere measurements taken during the 2002 summer MaCWAVE campaign, Geophys. Res. Lett., 31, L03105, doi:10.1029/2003GL018605.

[3] Perez, F., Granger, B. E., 2007, IPython: A System for Interactive Scientific Computing, Computing in Science & Engineering, Vol. 9, Issue 3, pp. 21 - 29.

[4] Hunter, J.D., 2007, Matplotlib: A 2D Graphics Environment Hunter, Computing in Science & Engineering Publication Date: May-June 2007, Vol. 9, Issue 3, pp. 90-95.

[5] Mlynczak, M. G., Marshall, T. B., Martin-Torres, J. F., Russell , J. M., Thompson, E. R., Remsberg, E. E., Gordley, L. L., 2007: Sounding of the Atmosphere using Broadband Emission Radiometry observations of daytime mesospheric O2(1D) 1.27 mm emission and derivation of ozone, atomic oxygen, and solar and chemical energy deposition rates, J. Geophys. R. Lett., Vol. 112, D15306, doi:10.1029/2006JD008355.

[6] Mlynczak, M. G., 1997, Energetics of the mesosphere and lower thermosphere and the SABER Experiment, Adv. Space. Res., Vol. 20, pp. 1177 – 1183.

[7] Oberheide, J., Hagan, M. E., Roble, R. G., 2003, Tidal signatures and aliasing in temperature data from slowly precessing satellites, J. Geophys. Res., Vol. 108(A2), doi:10.1029/2002JA009585.

[8] Pogoreltsev, A. I., Vlasov, A. A., Fröhlich, K., Jacobi, Ch., 2007, Planetary waves in coupling the lower and upper atmosphere. J. Atmos. Solar-Terr. Phys. 69, 2083-2101.

Address of the Authors:

Peter Hoffmann, Christoph Jacobi, Institute for Meteorology, University of Leipzig, Stephanstr. 3, 04103 Leipzig, phoffmann@uni-leipzig.de.

Sebastian Gimeno-Garcia, Institut für Methodik der Fernerkundung (IMF), Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Oberpfaffenhofen, Postfach 1116, 82234 Wessling.