

4-5 AMATERASU (テラヘルツ大気放射伝達モデル) の新機能 : SIW と SMILES-2 ミッションのための GPU による多偏波放射伝達計算の高速化

4-5 *New Features in AMATERASU: GPU Accelerated Polarized Radiative Transfer for SIW and SMILES-2 Studies*

バロン フィリップ

Philippe BARON

AMATERASU (テラヘルツ大気放射伝達モデル) は、大気リモートセンシングや通信のテラヘルツシステム的设计や解析に利用するために開発された。現在、中層・上層大気の観測を目的とする衛星ミッション、スウェーデンの SIW と日本の SMILES-2 の検討に用いられている。この2つの衛星ミッションはどちらも、中層大気の風を宇宙から初めて観測することを目的としている。この検討のために、AMATERASU の新たなモジュールを開発した。そのモジュールでは、オープンソースの TensorFlow ライブラリを使うことで GPU による計算を高速化し、さらに、地磁気によるゼーマン効果によって生じるテラヘルツ波大気伝搬の偏波依存性を解析することを可能とした。この論文では、開発内容とともに、現在進行中の衛星ミッションについてのこのモジュールに関連した検討状況を報告する。

The Advanced Model for Atmospheric TeraHertz Analysis and Simulation (AMATERASU) has been developed for designing and analyzing THz systems for atmospheric remote sensing or wireless communication. It is currently used to study two satellite missions for the middle and upper-atmosphere: the Swedish SIW and the Japanese SMILES-2. They both aim at measuring, among other, middle-atmospheric wind for the first time from space. To carry out these studies, new AMATERASU modules have been implemented. They allow us to accelerate the computation on GPUs using the open source library TensorFlow, and to consider the radiation polarization and the Zeeman effect induced by the geomagnetic field. In this paper, we describe these new developments within the context of the on-going satellite mission studies.

1 Introduction

The Advanced Model for Atmospheric TeraHertz Analysis and Simulation (AMATERASU) [1] is a Python program for analyzing THz systems used for atmospheric remote-sensing or wireless communication. Given atmospheric conditions, it simulates how the radiation propagates through the atmosphere and the level of attenuation. It has been used to various types of research. For instance, it helped to verify a model proposed for correcting the troposphere induced delays on the 2 and 14 GHz communication channels used for the ESA's Atomic Clock Experiment in Space (ACES) [2]. It has also been used to analyze atmospheric observations near 0.6 THz with spaceborne instruments [3]-[5], and to process the CO₂ mea-

surements of the coherent infrared Lidar (2 μm) built in the National Institute of Information and Communications Technology (NICT) [6]. It was the core of the NICT processing chain for the Superconducting Submillimeter-Wave Limb-Emission Sounder (SMILES) which operated during the winter 2009-2010 from the Japanese Experiment Module (JEM) onboard the International Space Station [5]. The mission was jointly led by the Japan Aerospace Exploration Agency (JAXA) and NICT with scientific PI from Kyoto University. It measured the atmospheric radiation spectra near the ozone (O₃) line at 625 GHz. Using the small Doppler-shift signal in this line, we have retrieved winds in the altitude range between 35 and 80 km where no global observations are available [7]. Such data are expected to improve meteorological and climate models as,

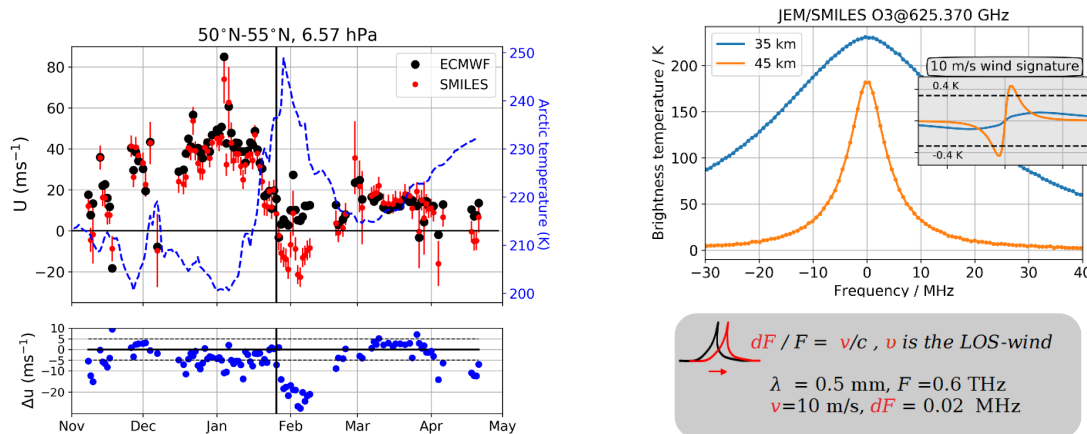


Fig. 1 Left Panel: Daily average of quasi-zonal winds retrieved from JEM/SMILES near 35 km between 50N-55N [7]. The black dots show forecasts from the European Centre for Medium-Range Weather Forecasts (ECMWF, version cy23r4). Right panel: SMILES O3 line and LOS wind signature. The black-dashed lines show the 1- σ noise level on a single spectrum (4-K cooled radiometer with 300 K SSB system-noise temperature).

for example, surface long-term weather forecast. This result broadens the interest for such missions, leading to the study of the Stratospheric Inferred Winds (SIW) [8] and SMILES-2 [9]-[11].

New features have been implemented in AMATERASU for conducting the simulation studies needed to design these missions. The large increase of the spectral information needed for wind measurements pushed us to accelerate the computations using Graphical Processing Units (GPUs). Also, the radiation polarization has been included to consider the Zeeman effect on the atomic and molecular oxygen lines targeted with SMILES-2, i.e., the change of the spectra due to the geomagnetic field. These features have been implemented with TensorFlow (TF), the opensource library developed by Google for machine learning (<https://www.tensorflow.org/>).

In this paper, we will first provide a quick overview of SIW and SMILES-2, and of the simulation studies. The new AMATERASU features will be presented and we will discuss the computation performances obtained with TF.

2 Context

2.1 SIW and SMILES-2 missions

Global wind observations from the surface to the upper atmosphere are needed to improve the meteorological and climate prediction models which are being extended above 80 km [10]. The two past decades are considered as the golden age for the spatial observation of Earth, but wind data are barely available below 80 km [7]. To fill in the lack of measurements, NICT is studying a space-borne coherent

Lidar to measure winds below 20 km, and passive THz heterodyne radiometers for the middle and upper-atmosphere (30–160 km). Figure 1 shows the wind component along the line-of-sight (LOS wind) obtained near 35 km with JEM/SMILES. This demonstrates the potential for such a technique to measure wind above about 30 km. This is roughly the lower limit for measuring wind because the wind induced Doppler shift is too small compared to the linewidth (the line is broadened by pressure).

The missions SIW and SMILES-2 are designed for retrieving horizontal winds, molecular composition and temperature using similar measurements. The observation method is shown in Fig. 2. The 2-d horizontal wind vector will be derived from perpendicular observations of the same air-mass using two antennas.

SIW is a low-cost Swedish mission that will be launched in 2022/23. Chalmers Technical University (Sweden) leads the project and NICT is involved in the definition of the mission concept and the simulation studies [8]. The scientific payload is a small instrument (17 kg, 47 W) with the capability for measuring winds between 35 and 80 km (Fig. 3). It uses a double-side band (DSB) radiometer near 0.64 THz passively cooled (70 K below ambient temperature). About 15 strong O₃ lines will be measured for obtaining a wind signal to noise ratio larger than that of the 4-K cooled JEM/SMILES.

SMILES-2 is a larger instrument (220 kg, ~300 W) equipped with 3 radiometers at 0.64, 0.76, and 2 THz, all cooled to 4.8-K [9][11]. The 0.64 THz band is the same as the SIW one. The mission could sense the altitude range between 20–160 km with a very high precision (130-180 K

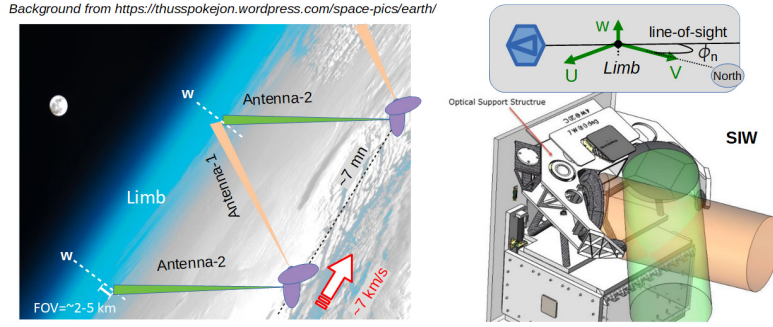


Fig. 2 Limb sounding representation for SIW and SMILES-2. On the right-side, a representative view of SIW with its two antennas (30 cm diameter) is shown (Omnisys Instruments).

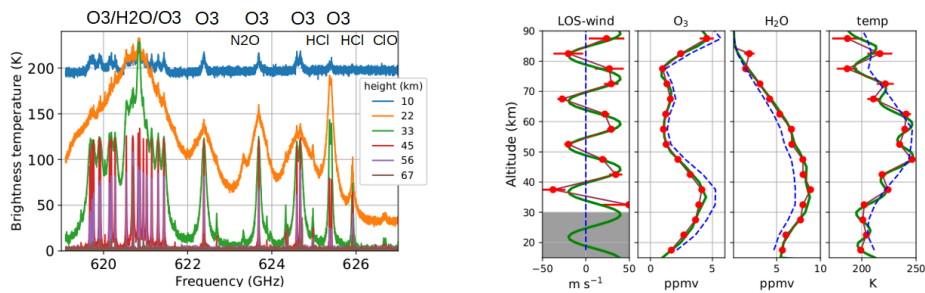


Fig. 3 Left panel: Simulated DSB Spectra for SIW. The frequency is that of the lower sideband (local oscillator 638.075 GHz, 1 MHz resolution, 1200 K DSB system-noise temperature). Right panel: Retrieval of LOS wind, O₃, H₂O and temperature (red lines). The true solution is shown in green and the blue lines show the profiles used to initialize the calculation. The horizontal lines show the 2- σ error bars. Figure taken from [8].

DSB system-noise temperature near 0.7 GHz) [9]. The project is led by Kyoto University and involves NICT, JAXA, and several other universities. NICT contributes to the mission concept definition, instrument design, and simulation studies. The realization of the mission is still under consideration.

A Model describing the propagation of the radiation through the atmosphere, such as AMATERASU, is essential for analyzing remote measurements. The model provides the relationship between the radiances measured with the instrument and the atmospheric state. AMATERASU is called the forward model (FM) and retrieving atmospheric information from the radiance is done by the inverse model. The quality of the retrievals depends on that of the measured radiances and on the FM accuracy. In addition, the FM must be fast enough to allow the processing of the large amount of data acquired with a satellite mission.

2.2 Simulation studies

Simulation studies are a prerequisite when defining a new instrument. They provide quantitative information to atmospheric and instrumental scientists helping them to converge toward the best compromise between scientific requirements and technological feasibility.

The forward model is the core of the simulator. It computes the outgoing radiation from line-of-sights passing through the limb at different altitudes (Fig. 4). The ray-path is divided in small segments (length <5 km) in which the atmosphere is assumed homogenous. The algorithms for unpolarized radiation are described in [1][12] and the polarized one in [13]. In short, the radiance exiting a small segment with length ds is:

$$I(s + ds) = (U - \exp^{-A(s)ds})B(T) + \exp^{-A(s)ds} I(s),$$

Equation 1

where $I(s)=[i(s), q(s), u(s), v(s)]^T$ is the Stokes vector describing the 4 polarization states of the radiation at the position s , B is the Stokes vector of the unpolarized black-body radiation at the temperature T , A is the 4x4 absorp-

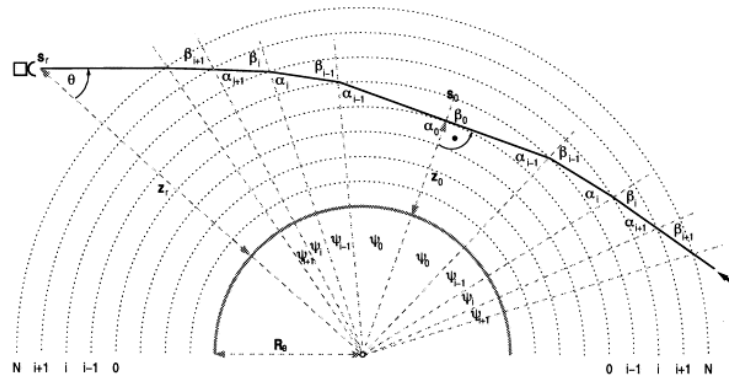


Fig. 4 Discretization of the line-of-sight into homogeneous small ranges. Figure taken from [12].

tion matrix and U is the identity matrix. \mathbf{A} is computed by summing up the lines with a significant contribution (line-by-line model). The line shape is the Voigt function, its complex form including dispersion effects is used for assessing the Zeeman effects [13]–[15]. The matrix exponential is the computation bottleneck of the model when a general algorithm is used. A fast function has then been implemented using the Caley-Hamilton development and the analytical solution of \mathbf{A} eigenvalues (2 independent values) [13]. Radiances are computed over fine tangent-height and frequency grids and convolved with the instrumental functions (e.g., antenna pattern or spectrometer channels). The case of unpolarized radiation is computed using a similar equation but with scalar parameters instead of matrices and vectors.

The standard approach for retrieving the forward model (FM) unknown parameters (atmospheric and observation parameters) from a measurement y can be expressed as:

$$\mathbf{x} = (\mathbf{K}^T \mathbf{K})^{-1} \mathbf{K}^T \mathbf{y} \tag{Equation 2}$$

Where \mathbf{x} is a vector (size= n) representing all the unknown parameters in the forward model, \mathbf{y} (size= m) contains all the spectra measured during a vertical scan, and \mathbf{K} is the Jacobian matrix $d\mathbf{y}/d\mathbf{x}$ (size= $[n, m]$). This equation gives the most probable vector \mathbf{x} for which the FM matches the noisy measurement. Usually iterations are performed to deal with non-linearities in the forward model, \mathbf{K} would be weighted with the measurement errors and an a priori knowledge of \mathbf{x} would be used as a constraint. Here, these issues have been omitted for simplification but are described in [4]. The matrix \mathbf{K} allows us to infer the information content of the measurement and to map into the x -space the measurement error described in the y -space. We characterize the measurement perfor-

mances by the precision and the spatial resolution of the retrieved geophysical parameters [8][9][13].

3 Quick tour of AMATERASU

The first version of AMATERASU (V1) uses functions derived from the code MOLIERE [12] to compute the absorption coefficient, the radiative transfer for unpolarized radiation and the instrumental effects. Each function, implemented in Fortran2000, are compiled into a DLL (Windows) or shared library (Linux, MacOS), and wrapped into python functions using the standard foreign-function library ctypes and the scientific library Numpy (<https://numpy.org/>). The model is a set of python functions, allowing a quick implementation of tools that match the experience specificities (e.g., atmospheric conditions, observation characteristics, retrieval algorithms). Calculations can be performed dynamically in a command line interface or using standalone scripts. The rich set of scientific and plotting libraries available in the python environment can be exploited. Additional functions are also provided to handle atmospheric and observation characteristics (e.g., instrumental effect, spectroscopic parameters, ray-path refraction, hydrostatic equilibrium) as well as the inverse problem.

The new functions implemented with TF can be used together with V1 ones. The main motivation for using TF was to use GPUs to boost the calculation runtimes. Figure 5 shows a calculation of the molecular oxygen (O_2) line at 773.84 GHz selected for SMILES-2. The LOS tangent height is 100 km and it is measured with a polarized instrument. The molecule of O_2 is a magnetic dipole that interacts with the geomagnetic field (B). The spectral transition is split into the so-called σ and π Zeeman lines (right panel). These lines are polarized, and their frequency separation depends

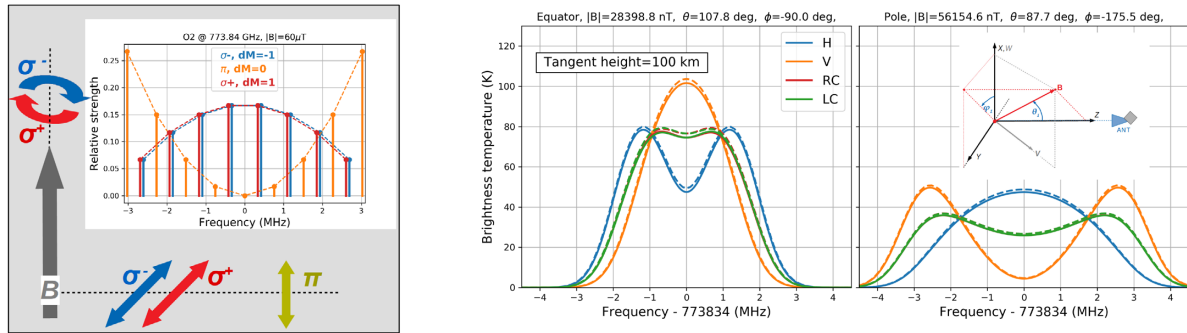


Fig. 5 Right panel: Zeeman lines of the O₂ transition at 773.834 GHz ($|B|=0.6$ Gauss) and polarization states of the lines with respect to the relative orientation of the LOS with the magnetic field. Right panel: O₂ spectra simulated over the equator (left-side plots) and over the North pole (right-side plots) considering the Zeeman. Results are shown for radiometers with horizontal (H), vertical (V) and circular (C) polarizations. The full and dashed lines are calculations with AMATERASU-TF and ARTS, respectively

Table 1 Spectroscopic parameters

Notation	Definition	Position
SPEISO	Isotopologue tag using HITRAN rules	1-2 (i2)
FRE	Frequency of the transition [MHz]	4-18 (f15.4)
ERR	Frequency error [MHz]	19-26 (f8.4)
FSH	Pressure induced frequency-shift at T0 [MHz/Torr]	27-34 (f8.3)
STG	Log10 of the line strength at T0 [MHz·nm ²]	39-42 (f8.4)
ELO	Energy of the lowest transition level [cm ⁻¹]	43-52 (f10.4)
AGA, SGA	air- and self-collisional broadening parameters [MHz/Torr] at T0	53-62 (f5.2)
N, NS	air and self broadening parameters temperature dependence [-]	63-70 (f4.2)
NF	Frequency-shift temperature dependence parameter [-]	73-76 (f4.2)
T ₀	Reference temperature [K]	77-83 (f7.2)
Q _{0,...,3}	Partition function coefficients (I)	84-127 (f11.4)
FSHS	Self frequency-shift parameter [MHz/Torr]	128-135 (f8.3)
NFS	Frequency-shift parameter temperature dependence	136-139 (f4.2)
(1)	$Q(T) = Q_0 + Q_1 (T-T_0) + Q_2 (T-T_0)^2 + Q_3 (T-T_0)^3$	

on the strength of B. These effects have been implemented in the new AMATERASU functions. Comparisons with calculations done with the Atmospheric Radiative Transfer Simulator (ARTS) [15] are shown. The differences between both models are less than 2% which is acceptable considering that the atmosphere and the spectroscopic parameters were not fixed. Further validations will be conducted including comparisons with the O₂ lines measured at 488 GHz from the Odin satellite [15].

In this section, we illustrate how to use both versions of AMATERASU with simple examples. We do not discuss the algorithms behind the functions since they are described in the references given in the previous section.

3.1 Spectroscopy

AMATERASU provides tools to handle standard spectroscopic databases (e.g., [16]) and to convert them into its own formats. For version V1, the parameters of a spectroscopic transition are stored in a string variable of 139 characters. The parameters describing a transition are

Table 2 Handling spectroscopic parameters with AMATERASU-V1. Comments are indicated with '#'.

```
import numpy as np #Scientific PYTHON library (www.numpy.org)
import amaterasu.common.spectro as AS #AMATERASU spectroscopic module
f_ghz = np.array([770, 780]) # Frequency range containing the O2 line at 773.84 GHz
# Read the line parameters in the HITRAN catalogue (use wavenumber (cm^-1)).
wavenum = AS.cm2mhz(f_ghz*1.e3, inv=1) #Wavenumber [cm^-1] (HITRAN unit)
#Select lines for O2, O3, H2O, HDO, N2O, 13-CO, HO2 lines between 770-780 GHz
h = AS.hitran().select(fre=wavenum, speiso=[71, 11,14,31,41,52,331])
outCabs_amalines = h.saveasama(None) # AMATERASU-V1 inputs
print(outCabs_amalines) # " 71 773839.7019...\n"
#List of spectroscopic and molecular parameters. One element per isopologue (zip variable)
hitranZip = h.getIso( size=20 ) #zip spectroscopy (ama format) and molecule dictionaries.
diclines_o2 = [lines for lines,mol in hitranZip if mol["name"] == "O2" ][0]
print(hitranZip[0][0]) # {'SPEISO': array([11, 11]), 'FRE': array([775502.9805, 774855.9684]), ...}
print(hitranZip[0][1]) # { 'name': 'H2O', 'iso': 161, 'riso': 0.997317, 'molarm': 18.010565}
```

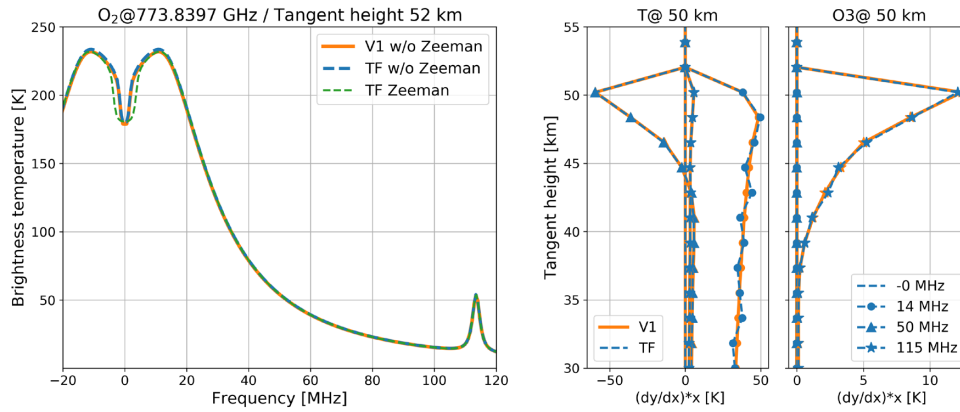


Fig. 6 SMILES-2 O₂ line without and with Zeeman effect. A line from O₃ is seen near 115 MHz. The Jacobian is computed without Zeeman effect and 1.5 km vertical resolution for the profiles in the vector x (Eq. 2).

Table 3 Definition of the atmospheric state with AMATERASU as well as the frequencies (10000 points near the O₂ line) and the LOS tangent heights (50 points between 30 and 120 km). We also create a magnetic field profile B with spherical coordinates shown in Fig. 5 (right-most panel).

```
import amaterasu.atm.atm as AA
dicAtm = AA.Atm().getProf( zkm= np.linspace(20,130,151))
print (dicAtm.keys()) #p, T, z, l, 2, 3, ...
nalt = dicAtm["z"].size #150, Number of altitudes
B = np.array( [[60.e-6,0.,0.]*nalt ] #array [nalt,3]: |B| (T), theta and phi (deg)
FGHZ=np.linspace(773.834-3, 773.834+3, 10000) #Frequencies (GHz), nF = 10000
TANH= np.linspace(30,120,50) #Tangent-heights (km), nTanh=50
```

Table 4 Unpolarized radiative transfer with AMATERASU-V1

```
import amaterasu.common.main as AM
ama = AM.Amaterasu()
ama.inCatm["zabs"] = dicAtm["z"]
ama.inCabs["f"] = FGHZ
ama.inCrt["flag"]=1 #Radiative transfer computation
ama.inCrt["unit"] = "rj" #Radiance unit in Rayleigh-Jeans brightness temperature
ama.inCrt["ObsHeight"] = [600.] #Observation altitude (needed for LOS elevation angles)
ama.inCrt["RadiusEarth"] = 6370. #Earth radius (km)
ama.inCrt["LimbStep_km"] = 3. #Integration segment size along LOS
ama.inCrt["ObsTanH"] = TANH
ama.setCout() #Initiatialize the model with the given setting
ama.outCabs["lines"] = outCabs_ama.replace("\n", "")
ama.outCatm = dicAtm #replace the default atmosphere with that created previously
ama.setAabs() #Absorption coefficient -> ama.outAabs["abscoef"], array [nF,nalt,1]
ama.setArt() #Radiative transfer
radv1 =ama.outArd["rad"][:,0] #Numpy array [nTanh,nF,1]
```

shown in Table 1. Several transitions are concatenated and separated by the UNIX end-of-line character “\n”.

The commands to select the parameters of the O₂ line selected for SMILES-2 (isotopologue tag=71) and those of nearby lines are shown in Table 2. The HITRAN database [16] is used and the data are stored per isotopologue into Python dictionaries (hitranZip). We also create the string

variable (outCabs_lines) to be used with AMATERASU-V1.

3.2 Radiative transfer with no polarization

Both versions of AMATERASU are used to compute outgoing radiation near the SMILES-2 O₂ line without considering the Zeeman effects. Figure 6 shows spectra computed for a LOS with a tangent height of 52 km and

Table 5 Same as Tab. 4 but with AMATERASU-TF

```

%env AMADEVICE=gpu:0
import amaterasu.p3d.zrt as ZRT
atf = ZRT.Scan(dicAtm ["z"]) #Layer altitudes (km) above the geoid.
atf.Surfacekm = 6370. #Geoid radius (km) below the tangent point
atf.setAtm(dicAtm, dicAtm ["p"], dicAtm ["t"], B=B);
[atf.setLine(ama, mol) for ama, mol in hitranZip ] #See Tab. 2
atf.setAbs(ama.outCfr["f"]) #Absorption coefficient (m^-1), atf.tf_abs_m [nalt,nF]
rad = atf.setTanh(ama.outCscan["tanh"], nranges=250).rad() #Tensor [nTanh, nF]
import pylab as P #Plot AMA-V1 and -TF spectra at tangent height 52 km (Fig. 6)
P.plot( (ama.outCfr["f"]-773.8397)*1.e3, ama.outArd["rad"][12,:;0]) #freq. are MHz
P.plot( (ama.outCfr["f"]-773.8397)*1.e3, rad.numpy()[12,:]); P.xlim(-25,120)

```

Table 6 Commands for Jacobian computations. The Jacobian matrix size is (5e5 x 600)

V1 / commands set before ama.setCout (Tab. 3)	TF / commands replacing atf.setAbs & atf.rad (Tab. 4)
ama.inCwf["flag"]=1	w = ZRT.Wf(atf, zbin=2, epsz=1.e-9) #75 altitudes
ama.inCwf["atm_z"]=np.linspace(20,130,76)	#Initialize for 6 species, temperature and wind (93)
ama.inCwf["atm_zup"]=None	s0 = w.scan0(ama.outCfr["f"], ama.outCscan["tanh"], atags=["t", 93])
Tags = [1,3,4,5,7,33,99,93]	Tags = s0.AbsTag.keys() #8 atmospheric parameters
ama.inCwf["atm"] = [[tag, 0., 0., 0.] for tag in Tags]	K= [w.jac(s0, tag, batch=10)["j"] for tag in Tags]
...	#Tensor [8,75,ntanh,nfre]
K =ama.outArd["rad"][:,:,1:] #Array [ntanh,nfre,75*8]	

Table 7 Same as Table 5 but with Zeeman O2 lines (use the same atmosphere and LOS heights).

```

atf.new() # Initialize the TF model used in the previous calculation
fghz = np.linspace(dicama_o2["FRE"][0]-150, dicama_o2["FRE"][0]+150, 1000)*1.e-3
#Spectroscopic data for all the lines but O2. Use of the Line object.
[atf.setLine( lines,mol ) for lines,mol in hitranZip if mol["name"] != "O2" ]
atf.setAbs(fghz) #Compute the absorption coefficient [nalt,nF] for all species but O2
#Set the O2 transition and magnetic field for Zeeman calculation
[atf.setZline( lines,mol ) for lines,mol in hitranZip if mol["name"] == "O2" ]
atf.Atm["B"] = B #Magnetic field vector profile created in Tab. 3.
atf.setZabs(fghz) # atf.tf_zabs_m[7,nalt,nF] (m^-1)
Stokes = atf.zrad() #TensorFlow tensor [nTanh, nF, 4] containing the 4 Stokes radiances

```

the Jacobians for O₃ and temperature at 50 km. The double peak near the O₂ line center (± 15 MHz) is due to saturation effects because the atmosphere is opaque above the tangent height. In AMATERASU-TF, the Jacobian is computed by evaluating the difference between $y(x+dx)$ and $y(x)$ where dx is a small perturbation on the FM parameter x . The results are consistent with the analytical computation used in AMATERASU-V1 [1][12].

The preamble of the calculation is the definition of the atmosphere. We need the profiles of temperature, pressure, wind, molecular volume mixing ratios and the magnetic field vector. The latter is defined with spherical coordinates in the frame shows in Fig. 5. Table 3 shows the way to create AMATERASU inputs between 20 and 130 km with

150 intermediate levels. The frequencies and LOS tangent heights considered in the calculations are also defined. We consider 10000 frequencies over 6 GHz bandwidth centered on the O₂ transition frequency and 50 tangent heights between 30 and 120 km.

Table 4 shows the AMATERASU-v1 commands to compute the radiation spectra. The absorption coefficients and radiances are stored in 3d NumPy arrays, the third dimension is used when the Jacobian matrix is computed. The same calculation with AMATERASU-TF is shown in Table 5. It is run on a GPU. The LOSs are divided into 250 ranges between the tangent point and the top of the atmosphere which corresponds to $ds=3$ km (4 km) for the tangent height of 87 km (52 km). Table 6 shows how to

compute the Jacobian matrix with respect to the 6 chemical species profiles considered in Table 2 as well as those of temperature and LOS wind. They are defined for 1.5 km thickness layers from 20 to 130 km ($x = 75 \times 8$ parameters).

3.3 Radiative transfer with Zeeman effect and polarization

Table 7 shows the commands to include the Zeeman effect on the O₂ line. The same setting as that in the previous section is used except for the frequency grid. It is reduced to 300 MHz and 1000 samples. The Zeeman lines of the O₂ transition are determined with the method “setZabs” and the polarized radiative transfer is applied (method “zrad” instead of “rad”). The spectrum of the total radiance at 50 km is shown in Fig. 5. Differences with non-Zeeman calculations (previous section) are only seen near the line center.

4 Performances

4.1 Benchmarks

Runtime improvements when using GPU are illustrated with the benchmarks presented in Table 8. The computations for unpolarized radiations presented in Sect. 3.2 are considered. The absorption coefficient (150 altitudes, 10000 frequencies) includes 105 spectroscopic lines. The Voigt line-shape is used and computed with the Hui algorithm taken from the model GARLIC [14]. The radiances are computed for 50 tangent heights. All calculations are done in double precision (FLT64) on a working station equipped with 2 CPUs Intel(R) Xeon(R) Silver 4108 @ 1.80GHz (32 threads) and a GPU NVIDIA TITAN V.

Table 8 Runtimes for the unpolarized radiative transfer calculations in Sect. 3.2.

	CPU - 1 thread	CPU - 32 threads	GPU
Absorption coefficient TF / V1	36 s / 1.1 s	7 s / x	0.24 s / x
Radiative transfer TF / V1	25 s / 13 s	7.2 s / x	0.09 s / x

Poor performances are found for the CPU computation of the TF absorption coefficient compared to V1 one. Our implementation of the Voigt function with TF has been identified as the bottleneck. This will be improved. When using the CPU with 32 threads, the full runtime (absorption coefficient and radiative transfer) are similar as that with V1 (14 s) but takes less than 0.35 s on the GPU. The Jacobian computations (Table 6) take about 50 s with both

AMATERASU-TF (GPU) and -V1 (not shown). The runtimes are similar though the perturbation method used in the TF version is much less computationally efficient than the analytical one in V1. The GPU runtimes for the absorption coefficients presented in Sect 3.3 (150 altitudes, 1000 frequencies) are 44.1 ms and 11 ms for the non-Zeeman transitions (104 lines) and for the O₂ transition (25 Zeeman lines, 7 coefficients), respectively. The polarized radiative transfer is computed in 2.4 s (4 Stokes parameters) which is 300 times slower than the calculation done without polarization (7 ms). This should be improved.

4.2 TensorFlow

Though it is developed for implementing codes for machine learning, TF can be used for other applications where large arrays are computed and handled. The model has been implemented in a relatively short time as we could expect from using a high-level language, allowing us to proceed smoothly in our studies. This must be underlined since we had very little knowledge of TF when we started and, likely, we can still improve the way the model has been implemented. Using the GPU clearly helps us to get satisfactory computational performances. The overall result is very positive.

We have chosen TF because it was the most used library when we started this project and a large user community provides supports on internet. Other similar tools (e.g., <https://pytorch.org/> or <https://cupy.chainer.org/>) are likely as valuable choices as TF and moving to another framework is still under investigation. The main appealing aspects in TF (version 1.12) are:

- The same code can be executed on GPUs and/or CPUs
- It is Implemented on top of Numpy and provides a relatively comprehensive linear algebra library
- The Eager mode allows us to use TF dynamically together with AMATERASU-v1
- It is portable to the main operating systems (AMATERASU-v1/TF tested on Linux, MacOS and Windows10)
- Automatic differentiation tools are available

We obviously must deal with issues. Some of them may be improved in future versions of TF:

- Assignment such as “*tensor[i]= ...*” is not directly possible with TF constant under Eager mode.
- Selecting a tensor element (e.g. “*y = tensor[k]*” or similar functions as gather or slice) has poor perfor-

mances.

- Sparse matrix functions are not implemented on the GPU for TF64 computations.
- Automatic differentiation cannot be used for computing the Jacobian matrix (it is designed for scalar output).

5 Conclusion

New AMATERASU modules have been implemented for studying the new THz sounders for middle and upper atmospheric, namely SIW and SMILES-2. The new modules allow us to accelerate the calculations on GPUs, to derive the Zeeman spectra of the molecular and atomic oxygen lines, and to consider the polarization of the radiation. They are implemented with TensorFlow and can be used together with AMATERASU-v1. The latter provides additional functions for modeling the instrumental effects, computing the inverse problem or interfacing standard continua absorption models (Liebe for the millimeter wavelengths and MT-CKD for the THz/infrared regions). AMATERASU-TF will be improved in the future with more efficient algorithms (e.g., for Jacobian computations or polarized radiative transfer). TensorFlow is also regularly updated which should likely bring improvements. An automatic differentiation method for computing the full Jacobian matrix would be very valuable. Migrating from TF to another framework is also investigated. Further studies about SIW and SMILES-2 will aim for the definition of fast retrieval algorithms. Deep machine-learning techniques will be investigated, and the use of TF or similar tools in AMATERASU make such studies relatively easier.

Acknowledgments

SMILES-2 is led by Prof. Masato Shiotani (Kyoto University). In NICT, the project is led by Satoshi Ochiai (Remote Sensing Group) who is designing the instrument. SIW is led by the Chalmers Technical University (Sweden) team of Prof. Donal Murtagh. We like to thank Franz Schreier (DLR, Germany) for providing the Voigt functions that he implemented in the model GARLIC [14]. We are grateful to Donal Murtagh and Joakim Möller (Molflow, Sweden) for their help. Many thanks for Richard Larsson (Max Planck institute, Germany) for providing the ARTS calculations. A special thought for Joachim Urban (Chalmers Technical University) who sadly passed away when we started these studies.

References

- 1 P. Baron, J. Mendrok, Y. Kasai, S. Ochiai, T. Seto, K. Sagi, K. Suzuki, H. Sagawa, and J. Urban, "AMATERASU: Advanced Model for Atmospheric Terahertz Radiation Analysis and Simulation," *Journal of the National Institute of Information and Communications Technology*, 55 (1):pp.109-121, 2008.
- 2 T. Hobiger, D. Piester, and P. Baron, "A correction model of dispersive troposphere delays for the ACES microwave link," *Radio Science*, 48(2):pp.131-142, 2013.
- 3 P. Baron, E. Dupuy, J. Urban, D. P. Murtagh, P. Eriksson, and Y. Kasai, "HO2 measurements in the stratosphere and the mesosphere from the Sub-Millimetre limb Sounder Odin/SMR," *International Journal of Remote Sensing*, 30(15-16):pp.4195-4208, 2009.
- 4 P. Baron, J. Urban, H. Sagawa, J. Möller, D. P. Murtagh, J. Mendrok, E. Dupuy, T. O. Sato, S. Ochiai, K. Suzuki, T. Manabe, T. Nishibori, K. Kikuchi, R. Sato, M. Takayanagi, Y. Murayama, M. Shiotani, and Y. Kasai, "The level-2 research product algorithms for the Superconducting Submillimeter-Wave Limb-Emission Sounder (SMILES)," *Atmospheric Measurement Techniques*, 4(10):pp.2105-2124, 2011.
- 5 O. Nyström, D. Murtagh, and V. Belitsky, "PHOCUS radiometer," *Atmospheric Measurement Techniques*, 5(6):pp.1359-1373, 2012.
- 6 S. Ishii, K. Mizutani, P. Baron, H. Iwai, R. Oda, T. Itabe, H. Fukuoka, T. Ishikawa, M. Koyama, T. Tanaka, I. Morino, O. Uchino, A. Sato, and K. Asai, "Partial CO2 column-averaged dry-air mixing ratio from measurements by coherent 2 μ m differential absorption and wind lidar with laser frequency offset locking," *J. Atmos. Oceanic Technol.*, 2012.
- 7 P. Baron, D. P. Murtagh, J. Urban, H. Sagawa, S. Ochiai, Y. Kasai, K. Kikuchi, F. Khosrawi, H. Körnich, S. Mizobuchi, K. Sagi, and M. Yasui, "Observation of horizontal winds in the middle-atmosphere between 30°S and 55°N during the northern winter 2009–2010," *Atmos. Chem. Phys.*, 13, pp.6049-6064, 2013.
- 8 P. Baron, D. Murtagh, P. Eriksson, J. Mendrok, S. Ochiai, K. Pérot, H. Sagawa, and M. Suzuki, "Simulation study for the stratospheric inferred winds (SIW) sub-millimeter limb sounder," *Atmospheric Measurement Techniques*, 11(7):pp.4545-4566, 2018.
- 9 S. Ochiai, P. Baron, T. Nishibori, Y. Irimajiri, Y. Uzawa, T. Manabe, H. Maezawa, A. Mizuno, T. Nagahama, H. Sagawa, M. Suzuki, and M. Shiotani, "SMILES-2 mission for temperature, wind, and composition in the whole atmosphere," *SOLA*, 13A:pp.13-18, 2017.
- 10 M. Shiotani, A. Saito, T. Sakazaki, S. Ochiai, P. Baron, T. Nishibori, M. Suzuki, T. Abe, H. Maezawa, S. Oyama, "A proposal for satellite observation of the whole atmosphere - Superconducting Submillimeter-Wave Limb-Emission Sounder-2 (SMILES-2)," *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2019.
- 11 S. Ochiai, "SMILES-2, a satellite observation plan of the whole atmosphere -- objectives and key technologies," *Journal of the National Institute of Information and Communications Technology*, this issue, 2019.
- 12 J. Urban, P. Baron, N. Lautié, N. Schneider, K. Dassas, P. Ricaud, and J. de La Noë, "MOLIERE (v5): a versatile forward- and inversion model for the millimeter and sub-millimeter wavelength range," *Journal of Quantitative Spectroscopy and Radiative Transfer*, 83(3-4):pp.529-554, 2004.
- 13 P. Baron, S. Ochiai, E. Dupuy, R. Larsson, H. Liu, D. Murtagh, S. Oyama, H. Sagawa, A. Saito, T. Sakazaki, M. Shiotani, and M. Suzuki, "The measurement of MLT wind, temperature, density and geomagnetic field with Superconducting Submillimeter-Wave Limb-Emission Sounder-2 (SMILES-2)," to be submitted to *Atmospheric Measurement Techniques*, 2019.
- 14 F. Schreier, S. G. García, P. Hedelt, M. Hess, J. Mendrok, M. Vasquez, J. Xu, "GARLIC — A general purpose atmospheric radiative transfer line-by-line infrared-microwave code: Implementation and evaluation," *Journal of Quantitative Spectroscopy and Radiative Transfer*, 137, pp.29-50, 2014.
- 15 R. Larsson, S. A. Buehler, P. Eriksson, J. Mendrok, "A treatment of the Zeeman effect using Stokes formalism and its implementation in the Atmospheric Radiative Transfer Simulator (ARTS)," *Journal of Quantitative Spectroscopy and Radiative Transfer*, 133, pp.445-453, 2014.
- 16 L.S. Rothman, I.E. Gordon, A. Barbe, et al., "The HITRAN-2008 molecular spectroscopic database," *Journal of Quantitative Spectroscopy and Radiative Transfer*, 110(910):pp.533-572, 2009.



Philippe Baron

Senior Researcher,
Remote Sensing Laboratory,
Applied Electromagnetic Research Institute
Ph.D
Atmospheric remote sensing,
Radiative transfer, Data scientist