PyIRTAM: A New Module of PyIRI for IRTAM Coefficients

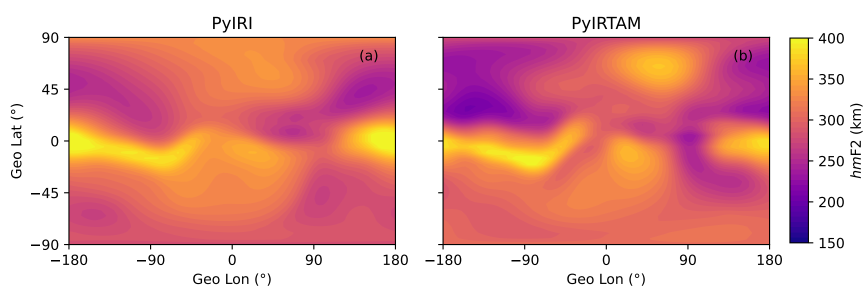
Victoriya Forsythe[1], Ivan A Galkin[2], Sarah E McDonald[3], Kenneth F. Dymond[3], Bruce Aaron Fritz[1], Angeline Gail Burrell[4], Katherine Anne Zawdie[3], Douglas Drob[5], Peter F. Caffrey[3], and Dustin A Hickey[6]

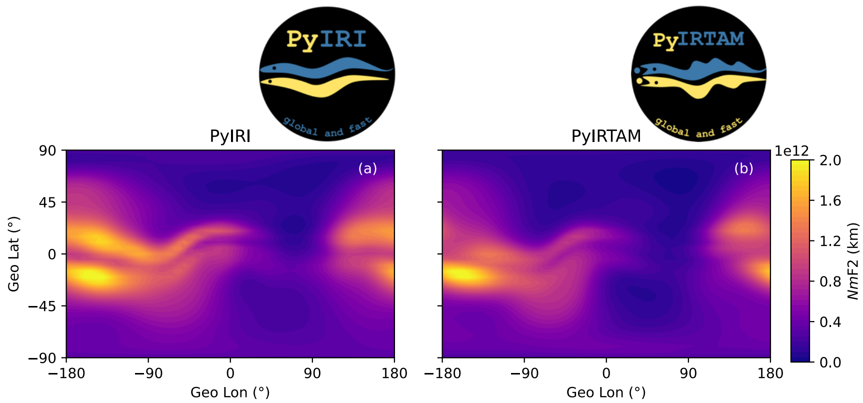
[1]U.S. Naval Research Laboratory
[2]University of Massachusetts Lowell
[3]Naval Research Laboratory
[4]US Naval Research Laboratory
[5]Naval Research Lab
[6]United States Naval Research Laboratory

March 25, 2024

Abstract

A novel model called PyIRI was recently developed. It offers a fully Python alternative to the widely used FORTRAN International Reference Ionosphere (IRI) model to construct the ionospheric electron density for the entire day and on the entire global grid in one computation, which has a significantly lower computational overhead. PyIRI introduced a novel approach to the computation of the global and diurnal functions and their matrix multiplication with Consultative Committee on International Radio (CCIR) coefficients, that enabled this global approach for the density specification. Since the IRI-based Real-Time Assimilative Model (IRTAM) produces coefficients in a similar format as CCIR coefficients, the PyIRI software was extended to be able to work with IRTAM coefficients using the same computationally efficient approach. This technical note describes the PyIRTAM software and shows examples of how to use it. PyIRTAM is 21 times more efficient than the classical IRTAM. PyIRTAM tool is made publicly available.

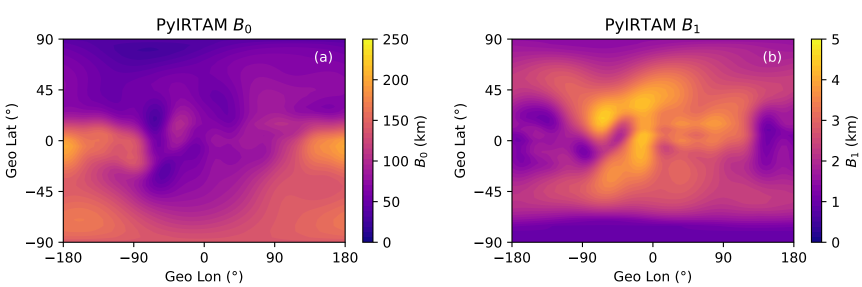


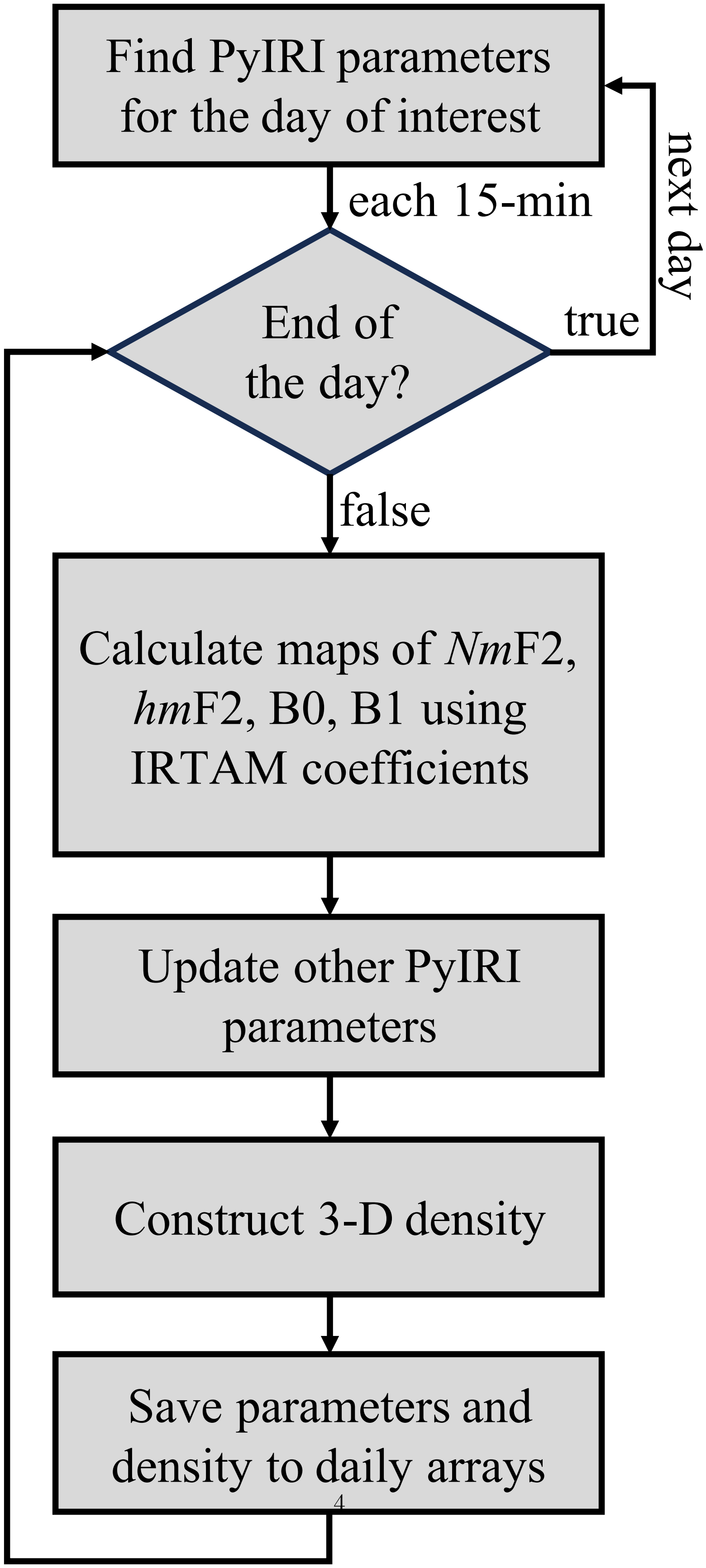


PyIRTAM Input Parameters					
Name of parameter	Explanation	Type	Specification	Units	Size
year	Year of interes	Integer	E.g. 2022		
month	Month of interest	Integer	E.g. 1		
day	Day of the month of interest	Integer	E.g. 1		
ahr	Time	1-D Numpy array	Can be regular or irregular array, but has to have 15-min resolution. E.g. [0, 0.25, 0.5, 0.75, 1, 1.25, ..., 23.75]	hours	[Nt]
alon	Geographic longitude	1-D Numpy array	Regular or irregular flattened grid array	degrees	[Ng]
alat	Geographic latitude	1-D Numpy array	Regular or irregular flattened grid array	degrees	[Ng]
aalt	Altitude	1-D Numpy array	Regular or irregular array	km	[Nv]
F107	F10.7 solar flux index	Float	E.g. 98.2	SFU	
irtam_dir	Place on your local machine where the IRTAM coefficients are located	String	'/Users/Documents/IRTAM/'		

PyIRTAM Output Parameters

Name of parameter	Explanation	Type	Keys	Units	Size
f2_iri	Dictionary with PyIRI parameters for F2 layer	Dictionary	'Nm' is peak density in m-3 'fo' is critical frequesncy in MHz 'M3000' is the obliquity factor for a distance of 3,000 km 'hm' is peak height in km 'B_top' is thickness of the topside in km 'B_bot' is thickness of the bottomside in km	m-3 MHz unitless km km km	[Nt, Ng]
f1_iri	Dictionary with PyIRI parameters for F1 layer	Dictionary	'Nm' is peak density in m-3 'fo' is critical frequesncy in MHz 'P' is probability density for occurrence of F1 layer 'hm ' is peak height in km 'B_bot' is thickness of the bottomside in km	m-3 MHz unitless km km	[Nt, Ng]
e_iri	Dictionary with PyIRI parameters for E layer	Dictionary	'Nm' is peak density in m-3 'fo' is critical frequesncy in MHz 'hm ' is peak height in km 'B_top' is thickness of the topside in km 'B_bot' is thickness of the bottomside in km	m-3 MHz km km km	[Nt, Ng]
es_iri	Dictionary with PyIRI parameters for sporadic E layer	Dictionary	'Nm' is peak density in m-3 'fo' is critical frequesncy in MHz 'hm ' is peak height in km 'B_top' is thickness of the topside in km 'B_bot' is thickness of the bottomside in km	m-3 MHz km km km	[Nt, Ng]
sun	Dictionary with the subsolar	Dictionary	'lon' is longitude of subsolar point in degrees 'lat' is latitude of subsolar point in degrees	degrees degrees	[Nt]
mag	Dictionary with magnetic parameters	Dictionary	'inc' is inclination of magnetic field in degrees 'modip' is modified dip anle in degrees 'mag_dip_lat' magnetic dip latitude in degrees	degrees degrees degrees	[Ng]
edp_iri	PyIRI 3-D electron density	Numpy array		m-3	[Nt, Nv, Ng]
f2_irtam	Dictionary with PyIRTAM parameters for F2 layer	Dictionary	'Nm' is peak density in m-3 'hm ' is peak height in km 'B0' thickness of the bottom side in km 'B1' shape parameter in km 'B_top' is thickness of the topside in km	m-3 km km km km	[Nt, Ng]
f1_irtam	Dictionary with PyIRTAM parameters for F1 layer	Dictionary	'Nm' is peak density in m-3 'hm ' is peak height in km 'B_bot' is thickness of the bottomside in km 'P' is probability density for occurrence of F1 layer	m-3 km km unitless	[Nt, Ng]
e_irtam	Dictionary with PyIRTAM parameters for E layer	Dictionary	'Nm' is peak density in m-3 'hm ' is peak height in km 'B_top' is thickness of the topside in km 'B_bot' is thickness of the bottomside in km	m-3 km km km	[Nt, Ng]
es_irtam	Dictionary with PyIRTAM parameters for sporadic E layer	Dictionary	'Nm' is peak density in m-3 'hm ' is peak height in km 'B_top' is thickness of the topside in km 'B_bot' is thickness of the bottomside in km	m-3 km km km	[Nt, Ng]
edp_irtam	PyIRTAM 3-D electron density	Numpy array		m-3	[Nt, Nv, Ng]




Find PyIRI parameters for the day of interest
each 15-min
End of the day?
true
next day
false
Calculate maps of *Nm*F2, *hm*F2, B0, B1 using IRTAM coefficients
Update other PyIRI parameters
Construct 3-D density
Save parameters and density to daily arrays

PyIRTAM: A New Module of PyIRI for IRTAM Coefficients

Victoriya V. Forsythe[1], Ivan A. Galkin[2], Sarah E. McDonald[1], Kenneth F. Dymond[1], Bruce A. Fritz[1], Angeline G. Burrell[1], Katherine A. Zawdie[1], Douglas P. Drob[1], Peter F. Caffrey[1], Dustin A. Hickey[1]

[1]U.S. Naval Research Laboratory, Washington, DC, USA
[2]University of Massachusetts Lowell, Space Science Laboratory, Lowell, United States of America

Distribution Statement

Key Points:

- Global approach for the IRTAM coefficients implemented in Python
- Python tool for making rapid global ionospheric electron density estimates
- It takes only 3 minutes to calculate 24-hour global electron density for high temporal and spacial resolution grid.

Corresponding author: Victoriya V. Forsythe, `victoriya.makarevich@nrl.navy.mil`

Abstract

A novel model called PyIRI was recently developed. It offers a fully Python alternative to the widely used FORTRAN International Reference Ionosphere (IRI) model to construct the ionospheric electron density for the entire day and on the entire global grid in one computation, which has a significantly lower computational overhead. PyIRI introduced a novel approach to the computation of the global and diurnal functions and their matrix multiplication with Consultative Committee on International Radio (CCIR) coefficients, that enabled this global approach for the density specification. Since the IRI-based Real-Time Assimilative Model (IRTAM) produces coefficients in a similar format as CCIR coefficients, the PyIRI software was extended to be able to work with IRTAM coefficients using the same computationally efficient approach. This technical note describes the PyIRTAM software and shows examples of how to use it. PyIRTAM is 21 times more efficient than the classical IRTAM. PyIRTAM tool is made publicly available.

1 Introduction

The International Reference Ionosphere (IRI) empirical model estimates the electron density in the ionosphere based on a climatological analysis of the ionospheric electron density profiles (EDPs) over several years. IRI is the gold standard for the ionospheric community. The International Standardization Organization (ISO), the International Union of Radio Science (URSI), the Committee on Space Research, and the European Cooperation for Space Standardization have all recognized IRI as the official standard for the Earth's ionosphere (ISO 16457: https://www.iso.org/standard/61556.html). A recent review paper by Bilitza et al. (2022) describes the current state of the IRI model, its history, and recent developments.

Recently, a novel tool called PyIRI was developed (Forsythe et al., 2024). PyIRI is a Python tool that modernized the core IRI components to take advantage of current matrix programming frameworks. It presented a novel approach for empirical ionospheric modeling that allows the evaluation of the model parameters simultaneously for global and Universal Time (UT) grid. It currently incorporates both the Consultative Committee on International Radio (CCIR) and URSI coefficients.

The IRI-based Real-Time Assimilative Model (IRTAM) (Galkin et al., 2020) is an operational ionospheric weather model based on the low-latency sensor inputs from the

Global Ionosphere Radio Observatory (GIRO) (Reinisch & Galkin, 2011). It provides a 3-D quiet-time climatology of the ionospheric plasma density by adjusting IRI definitions into a better match with the available measurements and geospace activity indicators. Every 15 minutes, IRTAM provides four files with coefficients that specify the electron density peak NmF2, the height of the peak hmF2, and two parameters (B_0 and B_1) that describe the shape of the bottom side of the EDP. The format of those files is very similar to the CCIR coefficients. Since PyIRI was developed for the rapid construction of the parameters and electron density using CCIR coefficients, it is beneficial to apply its approach for the processing of IRTAM coefficients. Applying the improved matrix handling methods used to create PyIRI to IRTAM led to the creation of a new ionospheric tool, PyIRTAM.

The rest of the paper describes the IRTAM formalism, describes the PyIRTAM tool and its approach to the EDP construction, and provides examples of how to use it.

2 CCIR and IRTAM Formalism

The CCIR coefficients were obtained in the pioneering studies conducted by Jones and Gallet (1962, 1965) and Jones et al. (1966). They analyzed the monthly medians of the critical frequency of the F2 layer, foF2, for minimum and maximum levels of solar activity. First they found the 13 coefficients for the Fourier time series to represent the diurnal trends of foF2 at about 150 ionosonde stations using a Least Squares minimization. They then found the coefficients for a special set of 76 geographic functions (similar to surface waves) to describe the variation of the diurnal Fourier coefficients with geographic location. As a result of their work, the diurnal and geographic variations of the monthly median foF2 measurements are described for two levels of solar activity using sets of monthly coefficients. The same approach was taken for the propagation factor M(3000)F2 from which the hmF2 can be derived. The derivation of B_0 and B_1 thickness parameters is described in Bilitza et al. (2022). A detailed explanation of the diurnal and global functions can be found in Forsythe et al. (2024).

Galkin et al. (2020) developed the IRTAM (http://giro.uml.edu/RTAM) system, which extracts four parameters (which are foF2, hmF2, B_0, and B_1) from the GIRO ionosonde data and assimilates them into IRI. In the assimilation process a set of updated diurnal functions is found from the time series analysis at each ionosonde station separately.

It uses a low-pass temporal filter as a part of its diurnal harmonics analysis to smooth out the data outliers and the low-confidence values. IRTAM starts with CCIR coefficients as an initial guess to represent the difference between foF2 data and the model. It also includes a linear trend term that accounts for potential day-to-day changes. Further, for each 15 minute time frame the global CCIR coefficients are updated to connect the measurements at the individual ionosonde stations and to obtain a global distribution. As a result, IRTAM generates four files that are intended to be updated in real time, every 15 minutes.

The IRTAM coefficients follow the CCIR coefficient format, but have an additional set of 76 coefficients for an additional diurnal term and omit the solar activity dependence. Since IRTAM relies on standard FORTRAN IRI code to reconstruct the global distributions of the parameters, it is beneficial to extend the PyIRI software to produce a 3-D electron density from the IRTAM coefficient files with the same efficiency as PyIRI.

Consider the differences between the IRTAM and IRI mathematical formalism using an example of the critical ionospheric frequency foF2. IRTAM expresses foF2 in terms of the diurnal variations at a geographic North latitude ϕ, East longitude θ, and at a particular time of the day in universal time (UT) t expressed as angle time from π to $-\pi$:

$$foF2(\phi, \theta, t) = a_0(\phi, \theta) + b_0(\phi, \theta)t_d + \sum_{i=1}^{M}[a_{2i-1}(\phi, \theta)\cos(it) + a_{2i}(\phi, \theta)\sin(it)], \qquad (1)$$

whereas the IRI definition does not include the $b_0(\phi, \theta)t_d$ term. In this additional term, t_d is defined as

$$t_d/\text{min} = 720/\text{min} + (UT/\text{hour} - TOV/\text{hour}) * 60/\text{min}, \qquad (2)$$

where UT is Universal Time in hours and TOV is Time of Validity. The b_0 slope is defined in 1/min units, and is applied to the 24 hour window centered in the middle (which explains 720 minute offset, since it is equal to 12 hours). An additional difference between the UT and TOV (converted to minutes) enables the use of IRTAM for the forecast. In case one is interested in nowcasting historical runs, the TOV should be the same as UT, and should also match with the time stamp in the name of the IRTAM coefficient files. The $(UT - TOV) * 60$ term disappears for the nowcast runs. However, if one is interested in the forecast mode the UT can go past the TOV into the future. Further, the peak of the electron density NmF2 can be derived from the foF2 as

$$NmF2/\text{m}^{-3} = 0.124 \times 10^{11}(foF2/\text{MHz})^2. \qquad (3)$$

105 Figures 1 and 2 compare the PyIRI and PyIRTAM $NmF2$ and $hmF2$ parameters,
106 respectively, during 1 Jan 2022, 10:00 - 10:15 UT.

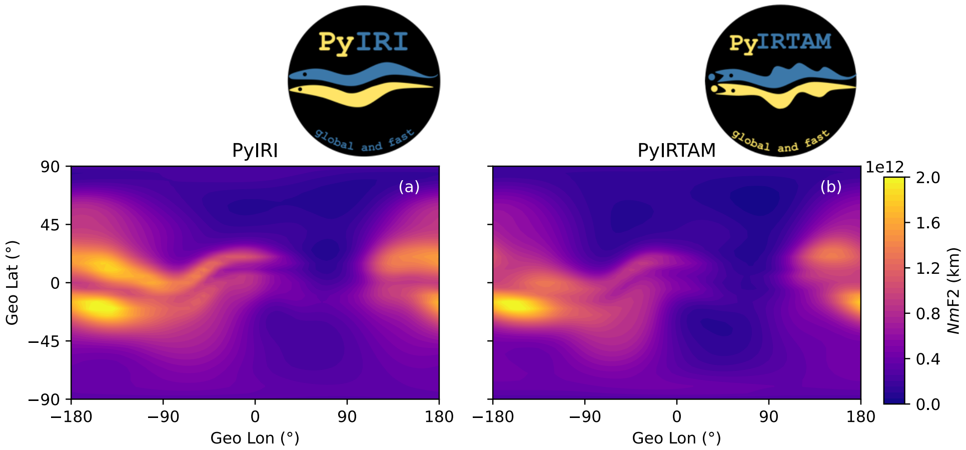


Figure 1: $NmF2$ for PyIRI (a) and PyIRTAM (b) on 1 Jan 2022, 10:00 - 10:15 UT.

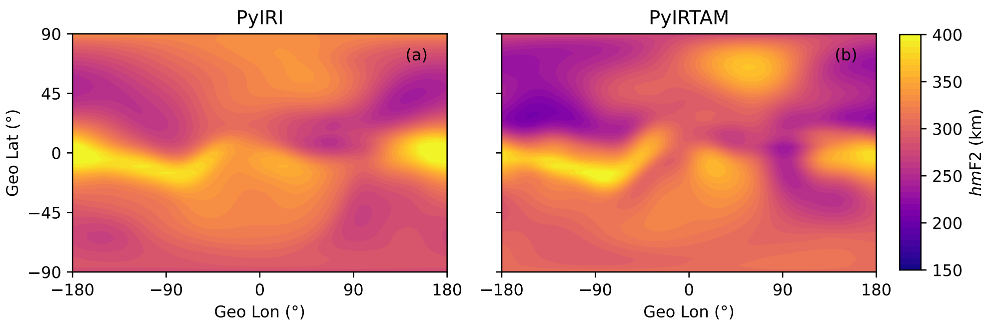


Figure 2: $hmF2$ for PyIRI (a) and PyIRTAM (b) on 1 Jan 2022, 10:00 - 10:15 UT.

107 The differences between panels (a) and (b) in Figures 1 and 2 are caused by the
108 ingested ionosonde data.

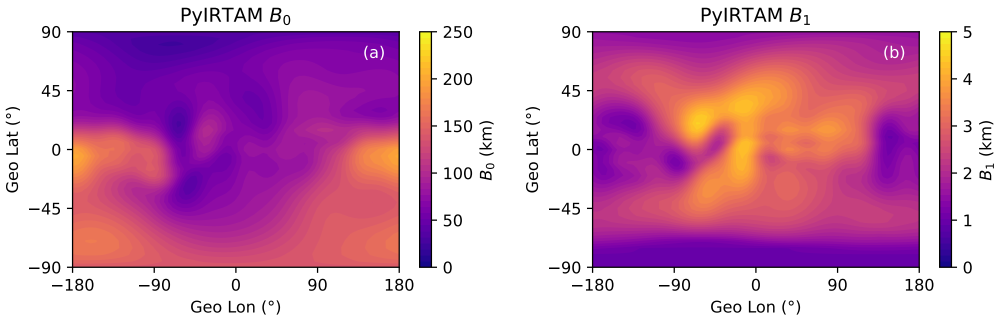


Figure 3: PyIRTAM B_0 and B_1 thickness parameters for 1 Jan 2022, 10:00 - 10:15 UT.

Additionally, Figure 3 shows F2 layer bottom side shape parameters B_0 and B_1. The comparison with PyIRI is not shown because PyIRI uses a different approach to the construction of the bottom side of F2 region, using a single thickness parameter B_{bot}^{F2} an td the Epstein function.

3 PyIRTAM Software

IRTAM coefficients are provided every 15 min. Therefore, this time resolution should be used to obtain the daily electron density distribution. However, since PyIRI can calculate electron density in one operation for the entire day, it needs to be executed only once in the beginning of each day of interest and not every 15 min, unless this is desired operationally.

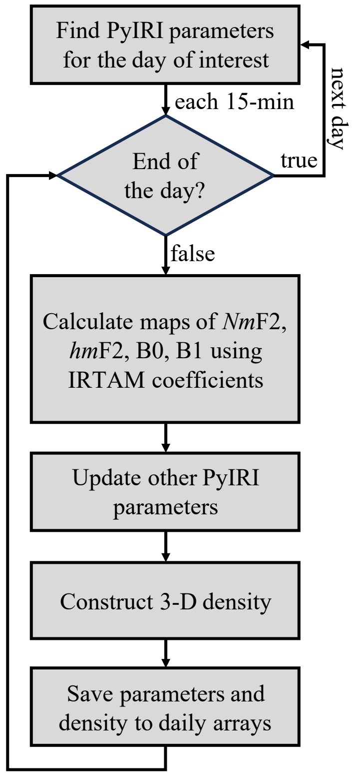


Figure 4: Flow chart to obtain daily distribution of electron density using PyIRTAM.

Figure 4 shows a flow chart of PyIRTAM. After the daily PyIRI parameters are found, the code starts a for-loop. For each 15 min time frame, the PyIRTAM (modified part of PyIRI) calculates the maps of NmF2, hmF2, B_0, and B_1 parameters using IRTAM coefficients. This modified routine includes an additional diurnal term and adds TOV as an additional input parameter. Next, PyIRI parameters that depend on the updated parameters are modified and the 3-D electron density is constructed. Recall that PyIRI uses just one thickness parameter for the F2 bottom side, while PyIRTAM uses different equations with two thickness parameters to construct the bottom side of the F2 layer ast described in Bilitza et al. (2022). Finally, the results are saved to the daily arrays of parameters and electron density.

PyIRTAM depends on Numpy and PyIRI (Forsythe & Burrell, 2023) .

PyIRTAM can be obtained either from GitHub or PyPi. PyPi installation is recommended to ensure all dependencies are installed.

The following submodules need to be installed:

```
import PyIRTAM.main_library as irtam_main
```

To obtain the ionospheric parameters and the electron density for a particular day, the following command in Python can be used:

```
f2_iri, f1_iri, e_iri, es_iri, sun, mag, edp_iri, f2_irtam, f1_irtam,
e_irtam, es_irtam, edp_irtam = irtam_main.run_PyIRTAM(year, month, day, ahr,
alon, alat, aalt, f107, irtam_dir)
```

where the inputs are explained in Table 1 and the outputs are explain in Table 2.

PyIRTAM Input Parameters					
Name of parameter	**Explanation**	**Type**	**Specification**	**Units**	**Size**
year	Year of interes	Integer	E.g. 2022		
month	Month of interest	Integer	E.g. 1		
day	Day of the month of interest	Integer	E.g. 1		
ahr	Time	1-D Numpy array	Can be regular or irregular array, but has to have 15-min resolution. E.g. [0, 0.25, 0.5, 0.75, 1, 1.25, ..., 23.75]	hours	[Nt]
alon	Geographic longitude	1-D Numpy array	Regular or irregular flattened grid array	degrees	[Ng]
alat	Geographic latitude	1-D Numpy array	Regular or irregular flattened grid array	degrees	[Ng]
aalt	Altitude	1-D Numpy array	Regular or irregular array	km	[Nv]
F107	F10.7 solar flux index	Float	E.g. 98.2	SFU	
irtam_dir	Place on your local machine where the IRTAM coefficients are located	String	'/Users/Documents/IRTAM/'		

Table 1: PyIRTAM input parameters.

PyIRTAM Output Parameters					
Name of parameter	**Explanation**	**Type**	**Keys**	**Units**	**Size**
f2_iri	Dictionary with PyIRI parameters for F2 layer	Dictionary	'Nm' is peak density in m-3 'fo' is critical frequesncy in MHz 'M3000' is the obliquity factor for a distance of 3,000 km 'hm' is peak height in km 'B_top' is thickness of the topside in km 'B_bot' is thickness of the bottomside in km	m-3 MHz unitless km km km	[Nt, Ng]
f1_iri	Dictionary with PyIRI parameters for F1 layer	Dictionary	'Nm' is peak density in m-3 'fo' is critical frequesncy in MHz 'P' is probability density for occurrence of F1 layer 'hm ' is peak height in km 'B_bot' is thickness of the bottomside in km	m-3 MHz unitless km km	[Nt, Ng]
e_iri	Dictionary with PyIRI parameters for E layer	Dictionary	'Nm' is peak density in m-3 'fo' is critical frequesncy in MHz 'hm ' is peak height in km 'B_top' is thickness of the topside in km 'B_bot' is thickness of the bottomside in km	m-3 MHz km km km	[Nt, Ng]
es_iri	Dictionary with PyIRI parameters for sporadic E layer	Dictionary	'Nm' is peak density in m-3 'fo' is critical frequesncy in MHz 'hm ' is peak height in km 'B_top' is thickness of the topside in km 'B_bot' is thickness of the bottomside in km	m-3 MHz km km km	[Nt, Ng]
sun	Dictionary with the subsolar	Dictionary	'lon' is longitude of subsolar point in degrees 'lat' is latitude of subsolar point in degrees	degrees degrees	[Nt]
mag	Dictionary with magnetic parameters	Dictionary	'inc' is inclination of magnetic field in degrees 'modip' is modified dip anle in degrees 'mag_dip_lat' magnetic dip latitude in degrees	degrees degrees degrees	[Ng]
edp_iri	PyIRI 3-D electron density	Numpy array		m-3	[Nt, Nv, Ng]
f2_irtam	Dictionary with PyIRTAM parameters for F2 layer	Dictionary	'Nm' is peak density in m-3 'hm ' is peak height in km 'B0' thickness of the bottom side in km 'B1' shape parameter in km 'B_top' is thickness of the topside in km	m-3 km km km km	[Nt, Ng]
f1_irtam	Dictionary with PyIRTAM parameters for F1 layer	Dictionary	'Nm' is peak density in m-3 'hm ' is peak height in km 'B_bot' is thickness of the bottomside in km 'P' is probability density for occurrence of F1 layer	m-3 km km unitless	[Nt, Ng]
e_irtam	Dictionary with PyIRTAM parameters for E layer	Dictionary	'Nm' is peak density in m-3 'hm ' is peak height in km 'B_top' is thickness of the topside in km 'B_bot' is thickness of the bottomside in km	m-3 km km km	[Nt, Ng]
es_irtam	Dictionary with PyIRTAM parameters for sporadic E layer	Dictionary	'Nm' is peak density in m-3 'hm ' is peak height in km 'B_top' is thickness of the topside in km 'B_bot' is thickness of the bottomside in km	m-3 km km km	[Nt, Ng]
edp_irtam	PyIRTAM 3-D electron density	Numpy array		m-3	[Nt, Nv, Ng]

Table 2: PyIRTAM output parameters.

The most important output parameter is `edp_irtam` which contains 3-D electron density distribution with the size [Nt, Nv, Ng], where Nt is the size of the input time array `ahr`, Nv is the size of the input altitude array `aalt`, and Ng is the size of the flattened grid arrays `alon` and `alat`. `edp_iri` output contains the PyIRI density output, whereas the other output parameters describe the ionospheric layers.

Finally, to quantify the computational efficiency of PyIRTAM, both IRI-based IRTAM and PyIRTAM were run on the same system, which is a single processor Linux system Intel(R) Xeon(R) CPU E5-2695 v3 @ 2.30GHz. IRTAM was compiled with `gfortran`. Both runs were completed for the horizontal grid of 16110 cells (2° horizontal resolution), vertical grid of 101 levels, and the 96-element time array (15-min time resolution). It took 62.88 min and 3.16 min for IRTAM and PyIRTAM, respectively. Which means that IRTAM is 21 times more efficient.

4 Conclusion

This paper presents a new, efficient application of IRTAM, PyIRTAM. It applies a computationally efficient approach to coefficient handling developed for PyIRI to the IRTAM coefficients. It takes only 15 sec on a regular PC to process a day of IRTAM coefficients and to construct 3-D electron density. PyIRTAM software is available to the community at GitHub (citation when available).

5 Open Research

PyIRTAM software is available to the community at GitHub (Forsythe & Burrell, 2024).

Acknowledgments

This work is sponsored by the Office of Naval Research.

References

Bilitza, D., Pezzopane, M., Truhlik, V., Altadill, D., Reinisch, B. W., & Pignalberi, A. (2022). The International Reference Ionosphere model: a review and description of an ionospheric benchmark. *Rev. Geophys.*, *60*(4). doi: 10.1029/2022RG000792

Forsythe, V. V., Bilitza, D., Burrell, A. G., Dymond, K. F., Fritz, B. A., & McDonald, S. E. (2024). PyIRI: Whole-Globe Approach to the International Reference Ionosphere Modeling Implemented in Python. *Space Weather*. doi: 10.1029/2023SW003739

Forsythe, V. V., & Burrell, A. G. (2023). victoriyaforsythe/PyIRI: v0.0.2. *GitHub*. doi: 10.5281/zenodo.10139334

Forsythe, V. V., & Burrell, A. G. (2024). victoriyaforsythe/PyIRTAM: v0.0.2a. *GitHub*. doi: 10.5281/zenodo.10844522

Galkin, I. A., Reinisch, B. W., Vesnin, A. M., Bilitza, D., Fridman, S., Habarulema, J. B., & Veliz, O. (2020). Assimilation of sparse continuous near-Earth weather measurements by NECTAR model morphing. *Space Weather*, *18*. doi: https://doi.org/10.1029/2020SW002463

Jones, W. B., & Gallet, R. M. (1962). Representation of diurnal and geographical variations of ionospheric data by numerical methods. *Journal of Research of the National Bureau of Standards*, *66*(4), 129–147. doi: 10.6028/jres.066D.043

Jones, W. B., & Gallet, R. M. (1965). Representation of diurnal and geographic variations of ionospheric data by numerical methods, II. Control of instability. *ITU Telecommunication Journal*, *32*(1), 18–28.

Jones, W. B., Graham, R. P., & Leftin, M. (1966). Advances in ionospheric mapping by numerical methods. *ESSA Technical Report ERL 107-ITS75, U.S. Department of Commerce*.

Reinisch, B. W., & Galkin, I. A. (2011). Global ionospheric radio observatory (GIRO). *Earth, Planets and Space*, *63*, 377–381. doi: 10.5047/eps.2011.03.001