The Empirical Canadian High Arctic Ionospheric Model (E-CHAIM): Primer

David R. Themens, Anthony M. McCaffrey, Karim Meziane, Benjamin Reid, P.T. Jayachandran, Joey Bernard

Version: 3.0

Release Date: September 1st, 2018
Updated: Aug 28th, 2020

Table of Contents

1 Overview

E-CHAIM is designed as an empirical electron density model of the high latitude ionosphere intended to replace the use of the IRI at these latitudes. To this end, the model represents ionospheric electron density in the region above 50°N geomagnetic latitude. The model is composed of several sub-models, each representing a key feature in the ionospheric electron density profile. Like the IRI, NmF2 and hmF2 are chosen as the anchor point of the profile, with all other components representing characteristics with respect to the F2 peak density and height. Each of these sub-models feature a spherical cap harmonic expansion in the new Altitude-Adjusted Corrected Geomagnetic (AACGM) coordinates of Shepherd [2014], calculated at 350km altitude, for the representation of the horizontal structure of the modelled parameter. All further references to geomagnetic coordinates in this text refer to these 350km AACGM coordinates. These spherical cap harmonics are similar to those used in Weimer [1996], Liu et al. [2014], and Yamazaki et al. [2015]. The order and degree of this expansion is determined experimentally, based on the amount, distribution, and quality of available data. The seasonal variability is modelled by a Fourier expansion and solar cycle variability is modelled via a function of solar F10.7 cm flux and IG ionospheric index. The model parameterizations for the F2 peak and topside have been published in Themens et al. [2017 and 2018] with the remaining components expected to be published in the coming months.

References:

Liu, J., R. Chen, J. An, Z. Wang, and J. Hyyppa, (2014), Spherical cap harmonic analysis of the Arctic ionospheric TEC for one solar cycle, J. Geophys. Res. Space Physics, 119, 601-619, doi:10.1002/2013JA019501

Shepherd, S.G., (2014), Altitude-Adjusted Corrected Geomagnetic Coordinates: Definition and Functional Approximations, J. Geophys. Res., 119, doi:10.1002/2014JA020264

Themens, D.R., P.T. Jayachandran, I. Galkin, and C. Hall (2017). The Empirical Canadian High Arctic Ionospheric Model (E-CHAIM): NmF2 and hmF2, J. Geophys. Res. Space Physics, doi: 10.1002/2017JA024398

Themens, D.R., et al. (2018), Topside Electron Density Representations for Middle and High Latitudes: A Topside Parameterization for E-CHAIM Based on the NeQuick, Journal of Geophysical Research: Space Physics, 123, 2, 1603-1617.

Weimer, D. R. (1996), A flexible IMF dependent model of high-latitude electric potentials having "space weather" applications, Geophys. Res. Let., 23, 2549-2553.

Yamazaki, Y., M.J. Kosch, and E.K. Sutton (2015), A model of high-latitude thermospheric density. J. Geophys. Res. Space Phys., 120, doi:10.1002/2015JA021371

2 License

The Empirical Canadian High Arctic Ionospheric Model (E-CHAIM) License

Copyright and License Information

In general, we ask users to provide a valid email address to echaim@unb.ca, which will be used to contact users in the event of model updates and bug fixes. If you have acquired the code through the CHAIN website and created an account, your information has already been recorded.

AACGM License

Copyright and License Information

4201 Wilson Blvd,
Arlington, VA 22230
email: kbaker@nsf.gov

3 Limitations

The following list details the current limitations of the E-CHAIM model code:

1) The lower boundary of the model is 50°N geomagnetic latitude. The current code allows the model to be used between 45°N and 50°N geomagnetic latitude in order to provide a transition region for users interested in merging the model with other global models, such as the IRI. E-CHAIM results from this region should not be used for any other purpose. Each version of the E-CHAIM code (IDL, Matlab, C) includes a warning flag to assist users in identifying this transition region. Any requests below 45°N geomagnetic latitude will return NaNs.
2) This model uses AE index as a primary driving index; however, due to the slow updating of AE index, we have developed a model of AE index based on PC index for use during periods when AE index is not available. This allows the model to be run as close to real time as possible. During any period where both AE and PC index are not available, the all-time average PC index is used. Similarly, the all-time average ap and Dst indices are used for the E-CHAIM storm model if either of these indices are not available. Each version of the code includes warning flags to assist the user in identifying what Dst, ap, AE, or PC index is being used.
3) F10.7 solar activity indices are available at latencies of less than a day from the Canadian Government ftp, as such, the model database is updated daily on the CHAIN website for the user's convenience. If a period within 40 days of the last F10.7 record is requested, a NOAA F10.7 forecast is appended to the F10.7 timeseries for calculation of the required 81-day smoothed F10.7 flux.
4) If the user requests a period beyond the last IG index record, forecasted IG12 is used in place of monthly IG index.
5) The model does not have an explicit upper or lower altitude boundary. That said, use outside of the altitude range of the initial fitting (80km to 3500km) may produce unexpected results.

4 Recommendations

1) When available, it is always recommended that the Storm NmF2 model is used in place of the quiet NmF2 model. This is not included as the default option in the model, as AE, Dst, ap, and PC index may not always be available.

5 Formulation

The formulation of the E-CHAIM model has been published in Themens et al. [2017 and 2018]. We encourage users to consult these publications prior to their use of E-CHAIM.

In the development of the E-CHAIM code distribution a few changes have been made to the model to facilitate a simpler cross-platform implementation of the model code and improve the computational performance of the model. We shall here list these changes:

1) The E-CHAIM quiet NmF2 model now uses geomagnetic longitude and latitude as its horizontal coordinates, in place of the previous geomagnetic local time and latitude. This change was made to avoid having to calculate magnetic coordinates twice in the interpolation process between hours. This modification had little to no impact on the performance of the NmF2 model.
2) The E-CHAIM quiet NmF2 model now uses a 4-3 spherical cap harmonic expansion instead of the previous 5-4 expansion. This reduced expansion produces electron densities more comparable to the IRI at the model's lower boundary, which will facilitate a smoother transition when attempting to merge both models at the lower boundary. This change only resulted in a ~1% loss in model fitting performance.
3) The E-CHAIM topside model now uses g = 0.18 in place of g = 0.2024. This g value corresponds to the best correlation, rather than the previous value, which optimized RMS error. This value better captures the high altitude topside, which contributes relatively little to overall RMS fitting error.

The E-CHAIM bottomside, hmF2, and storm NmF2 models remain unchanged from those published in Themens et al. [2017, 2018].

6 Matlab Distribution

There are two versions of the Matlab E-CHAIM code available:

1) A version featuring a self-updating database. In this version, if a date is requested beyond the last record within the database, the model will automatically search the various index ftp sites and self-update. This process will occur a maximum of once per day to prevent repeated, potentially long, update cycles.
2) A version that uses an identical database to that of the C code (in .mat file format) but does not self-update. This version is ideal for users who have ftp access restrictions. Database files for this version are updated daily and provided on the E-CHAIM website.

6.1 *System Requirements*

The Matlab E-CHAIM distribution presently only works with Microsoft Windows and Linux/Unix platforms. Extension to Mac platforms is in beta. The code has been tested with Matlab versions 2017a, 2017b, 2019a, 2019b, and 2020a.

The Matlab model's cross-platform capabilities are limited by the AACGM shared library. Shared libraries built on Windows (Windows 10 64-bit) and Unix (Ubuntu 18.04) are provided and this should allow for the model to run on both Windows and Unix machines. If the model does not work, instructions to implement an AACGM shared library (built on their machine) into the model are provided below:

- First the user must create a shared library (named aacgm) from the AACGM raw C code. This is located in aacgm/rawFiles
    - For OSX we expect the compilation command to look something like:
        - gcc -dynamiclib -o aacgm.dylib *.c
- Next, place the library file in the aacgm > sharedlibrary folder in the Matlab project folder.
- In Matlab, CD to the aacgm > sharedlibrary folder.
- In Matlab, enter loadlibrary('aacgm','aacgm.h','mfilename','aacgmproto')
- Note that this will overwrite the current aacgmproto file. This is fine but means it will now be specific to the user's system.
- The Matlab code should now use the new shared library.
- Note:

6.2 *Package Contents*

The E-CHAIM Matlab distribution includes all necessary source code, data files, and libraries needed to run E-CHAIM immediately. Note that the source code is encrypted (Matlab P code) to eliminate possible errors by accidental editing of the E-CHAIM source.

6.3 Installation

1. Unpack the model code into a directory of your choosing, referred to as the project directory.
2. Add the root directory and all subfolders to your Matlab path.
3. Open ECHAIMConfig.m, located in the UserConfig folder within the project directory.
4. Replace the hardcoded value for the output.projectFolder variable to the full project directory path as a character array or string data type.
5. Run ECHAIM('test') to run an installation test. Compare the output from this to the expectedOutput.txt file located within the Installation Test folder.
6. If there is an error with the test, run ECHAIM('debug'). If this does not solve the issue, send the newly created ECHAIMDebug.zip file to ECHAIM@unb.ca with details of your issue.

6.4 How To Run The Model

The main model code is ECHAIM.m. The code features three modes: the default mode that functions identically to those of the IDL and C codes and generates profiles at matching sets of locations and times, a mode that determines the electron density along a prescribed path, called Satellite Mode, and a mode that provides 4D electron density fields for a prescribed domain and set of times, called Map Mode. The Matlab version of the model also features a GUI for easy visualization of model output.

6.4.1 Default Mode

In the default operation of the Matlab E-CHAIM code mode the user provides vectors of latitudes, longitudes, times, and altitudes to ECHAIM.m, for which a structure is returned containing electron density profiles at the altitudes requested for each triplet of latitude, longitude, and time. These latitude, longitude, and time vectors must be the same length, while the altitudes vector can be of any length. The latitudes and longitudes should be in geographic degrees, the times must be a datetime array, created using the Matlab datetime function and should be in UTC, and the altitudes must be in kilometers.

For example: If we wanted the electron density profiles between 60 km and 1000 km at 1km resolution at 55°N, 69.5°E on January 23rd, 2013, at 11:24:50UTC and 15:47:20UTC, we could use the following code to call E-CHAIM:

```
dates_use = datetime([2013 2013], [1 1], [23 23], [11 15], [24 47], [50 20]);
lat_use = [55 55];
lon_use = [69.5 69.5];
alts_use = 60:1:1000;
YY = ECHAIM (lat_use, lon_use, dates_use, alts_use);
```

YY is then a structure with fieldnames given as

NmF2, HmF2, HmF1, NmE, NmF1, dens, HF1, HBot, HTop, HE, lon, lat, mlon, mlat, mlt, dates

Electron densities corresponding to the two input times/locations can then be accessed by using the following

Electron_densities = YY.dens

which has the dimension N_times x N_altitudes. Other output variables are accessed using their corresponding structure fieldnames and are vectors of dimension N_times. Note that the NmF2, hmF2, hmF1, AACGM latitude (mlat), AACGM longitude (mlon), and AACGM magnetic local time (mlt) are all also returned in this call and correspond to the inputted latitudes, longitudes, and times triplet. Electron densities are provided in units of e/m^3 while all heights are provided in kilometers.

6.4.2 Satellite Mode

In Satellite mode, the user provides vectors of latitudes, longitudes, times, and altitudes to ECHAIM.m, for which a structure is returned containing the corresponding electron densities at those locations and times. These latitude, longitude, time, and altitude vectors must be the same length. The latitudes and longitudes should be in geographic degrees, the times must be a datetime array, created using the Matlab datetime function and should be in UTC, and the altitudes must be in kilometers.

For example: If we wanted the electron density at 55^{o}N, 69.5^{o}E, and 1100km altitude on January 23rd, 2013, at 11:24:50UTC and 15:47:20UTC, we could use the following code to call E-CHAIM:

dates_use = datetime([2013 2013], [1 1], [23 23], [11 15], [24 47], [50 20]);
lat_use = [55 55];
lon_use = [69.5 69.5];
alts_use = [1100 1100];
YY = ECHAIM (lat_use, lon_use, dates_use, alts_use, ‘sat’);

YY is then a structure with fieldnames given as

NmF2, HmF2, HmF1, NmE, NmF1, dens, HF1, HBot, HTop, HE, lon, lat, mlon, mlat, mlt, dates

Electron densities corresponding to the two inputs can then be accessed by using the following

Electron_densities = YY.dens

which has the same length as dates_use (and all the other input vectors).

Note that the NmF2, hmF2, hmF1, AACGM latitude (mlat), AACGM longitude (mlon), and AACGM magnetic local time (mlt) are all also returned in this call and correspond to the inputted latitudes, longitudes, and times. Electron densities are provided in units of e/m^3 while all heights are provided in kilometers.

6.4.3 Storm, D region, and Precipitation Keywords

E-CHAIM includes a precipitation module to account for ionization due to auroral precipitation into the E-region. If the storm NmF2 model (and/or the precipitation density model) is desired, one must simply add the ‘storm’ or ‘precip’ keywords to the ECHAIM.m call. The keywords can be used together, separately, or in combination with other keywords.

E-CHAIM also includes a D-Region model, which can be used with the ‘dregion’ keyword.

YY = ECHAIM(lat_use, lon_use, dates_use, alts_in, ‘storm’,’precip’,’dregion’);

6.4.4 Map Mode

The second operational mode of the Matlab code is the Map Mode. This mode is designed to produce three- or four-dimensional electron density maps. In this mode, the user supplies vectors of latitudes, longitudes, times, and altitudes, which can each be of different length. The code will iterate through all combinations of these inputs to produce a 4D electron density output.

For example: If one wishes to generate maps of electron density for latitudes between 60oN and 65oN at 1o resolution, longitudes between 200oE and 220oE at 2o resolution, and altitudes between 100km and 600km at 5km resolution for three times (February 3rd, 2010, at 06:09:34UTC; June 12th, 2011, at 13:06:53UTC; and June 12th, 2011, at 03:06:29UTC) one could simply use the following set of commands:

dates_use = datetime([2010 2011 2011], [2 6 6], [3 12 12], [6 13 3], [9 6 6], [34 53 29]);

lats_use = 60:1:65;
lons_use = 200:2:220;
alts_use = 100:5:600;
YY = ECHAIM (lats_use, lons_use, dates_use, alts_use, ‘map’);

Note the use of the ‘map’ keyword. The resulting electron densities can be accessed using
Electron_Densities = YY.dens;

This Electron_Densities variable will have dimensions of N_times x N_lats x N_lons x N_alts (3x6x11x101 in this case). As with the Satellite Mode, the NmF2, hmF2, etc… corresponding to each latitude, longitude, and time is also returned and can be accessed using the corresponding fieldnames. These parameters will have dimension N_times x N_lats x N_lons (3x6x11 in this case).

6.4.5 Database Updating

Keywords have been added to allow the user the choice of whether the model will self update the local database file. By default, the model will not attempt to update the database. If the user supplies the ‘update’ keyword, the model will check the local database to see if it needs to be updated, if so, the latest database file will be downloaded from the E-CHAIM website and will overwrite the current local database file. If the user provided the keywords ‘force update’, the model will download and overwrite the local database file with the latest file on the E-CHAIM website without checking the local file first.

6.4.6 *E-CHAIM GUI*

To launch the GUI in the Matlab prompt run

```
>> guiTest4
```

A window will pop up [see Figure 1 below] so you can select the desired output parameter. Single profiles or Contours are the two types of output. The single profiles selection includes vertical density profiles and time/latitude/longitude series of NmF2, hmF2 and hmF1. The contours selection includes various possible contours based on two selected input variables; for example, Longitude-Latitude NmF2 contours, Latitude-Altitude density contours, can be constructed.

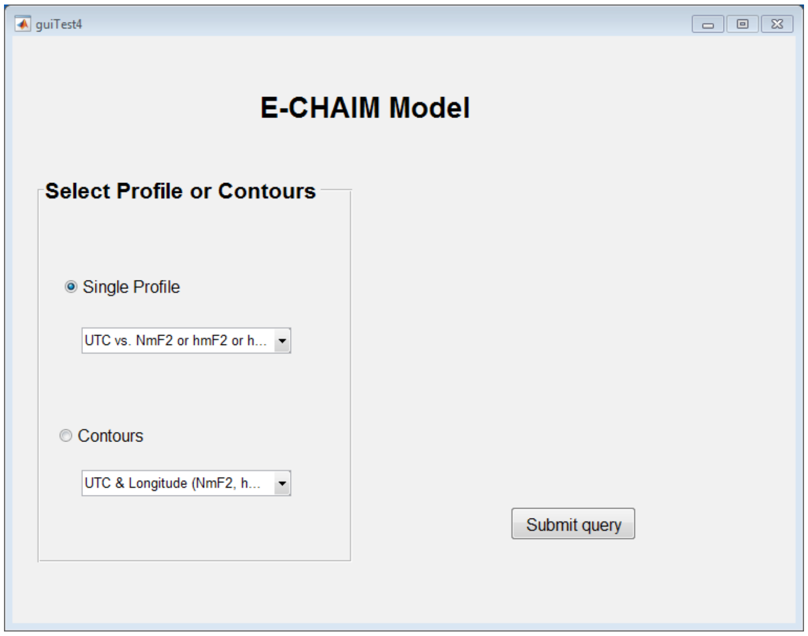
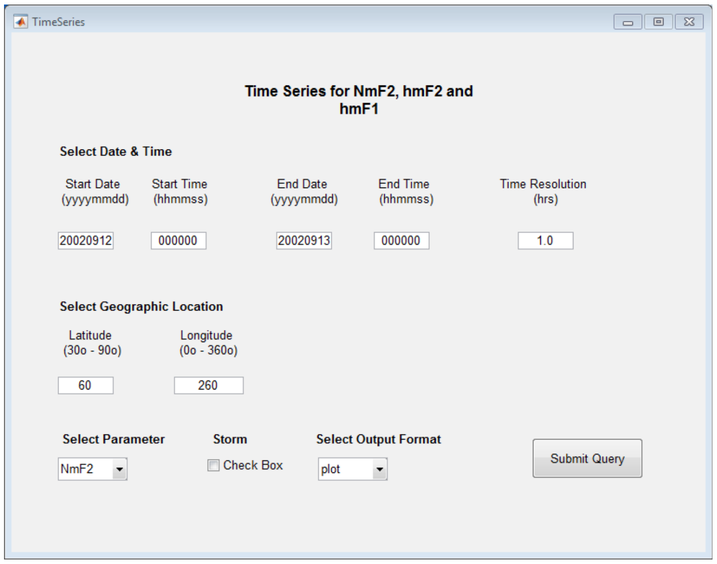

guiTest4
E-CHAIM Model
Select Profile or Contours
Single Profile
UTC vs. NmF2 or hmF2 or h...
Contours
UTC & Longitude (NmF2, h...
Submit query

Figure-1

Once a type of output and the desired parameter are selected, a second windows pops up. For example, if a profile of *UTC vs. NmF2 or hmF2 or hmF1* is selected the below figure pops up.


TimeSeries

Time Series for NmF2, hmF2 and hmF1

Select Date & Time

Start Date (yyyymmdd)	Start Time (hhmmss)	End Date (yyyymmdd)	End Time (hhmmss)	Time Resolution (hrs)
20020912	000000	20020913	000000	1.0

Select Geographic Location

Latitude (30o - 90o)	Longitude (0o - 360o)
60	260

Select Parameter: NmF2

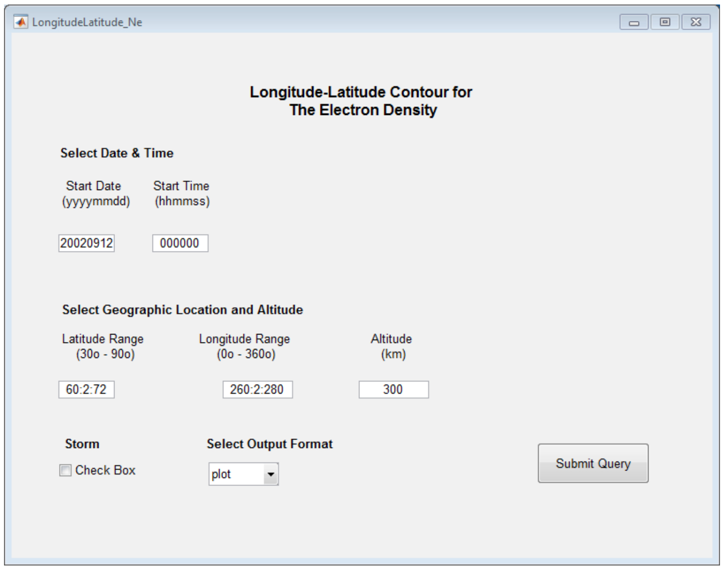
Storm: Check Box

Select Output Format: plot

Figure-2

Input the various variables needed to obtain the output: date, time and latitude and longitude according to the format indicated on the window. For the time series, the time resolution (in hours) is also needed. Also select which parameter is desired (NmF2, hmF2 or hmF1), select whether the NmF2 storm model output is desired (storm check box), and select the format of the output (plot, list, or ascii file). If the *ascii file* option is selected, a file is created in in the *pub folder* sub-directory within the E-CHAIM project directory.

If a Contours option is selected a new window will pop up. For example, if Longitude-Latitude density contour is selected, a window as shown on Figure-3 will be displayed.


LongitudeLatitude_Ne

Longitude-Latitude Contour for The Electron Density

Select Date & Time

Start Date (yyyymmdd)	Start Time (hhmmss)
20020912	000000

Select Geographic Location and Altitude

Latitude Range (30o - 90o)	Longitude Range (0o - 360o)	Altitude (km)
60:2:72	260:2:280	300

Storm: Check Box

Select Output Format: plot

Figure-3

Input the instant (date and time) of interest and the altitude at which the density contour should be calculated. Input the latitude and longitude ranges of interest. As indicated on the pop up window, both latitude and longitude are 1D-array and should be entered according to Matlab format: Lat_{MIN}:°Lat:Lat_{MAX} and Lon_{MIN}:°Lon:Lon_{MAX} where °Lat and °Lon are the latitude and longitude resolution, (Lat_{MIN},Lat_{MAX}) and (Lon_{MIN},Lon_{MAX}) define the geographic region of interest in terms geographic coordinates.

6.5 Known Issues

- On some machines, Matlab has been known to have permission issues when writing to folders. Running Matlab as administrator resolves this issue.

6.6 Reporting Bugs or Other Issues

If you have an error while running E-CHAIM, run E-CHAIM with the debug flag. If you have a constant problem, try:

ECHAIM(‘debug’);

If your problem is with a specific set of inputs, run echaim with these inputs and add the debug flag:

ECHAIM(lat,lon,dates,alts,otherflags,’debug’);

If the debug tool does not fix your problem, it will create a debug file (ECHAIMDebug.zip) in your project directory. Send this file, along with a full description of your issue to ECHAIM@unb.ca.

7 C Distribution

The C code is provided as an API, allowing for the calculation of NmF2, hmF2, electron density profiles, and the density along prescribed paths (ex: following a satellite trajectory). A separate function is provided for each of the above outputs, as well as their ‘storm perturbation’ counterparts. Each of these functions require location (geographic latitude and longitude), altitude (in kms, and where applicable), and time inputs.

A ’front end’ is provided for users who do not wish to use the API. This will read a configuration file that contains the required inputs (time, location, altitudes, etc.) and will print the desired output to a file. The API provides more flexibility and functionality than the front end and thus the API is recommended.

Error code retrieval functions are provided as an optional part of the API and are NOT available when using the front end.

7.1 System Requirements

7.2 Package Contents

The E-CHAIM C distribution includes the following:

1) All source code (.c) and header files (.h)
2) The database of indices (CHAIM_DB.db)
3) The database of model coefficients (COEFS_DB.db)
4) The database of precipitation model coefficients (PRECIP_DB.db)
5) AACGM coefficient files (aacgm_coeffs-12-???.asc, igrf12coeffs.txt)

7.3 Installation

The C code can be used either by manually compiling all of the source with the users code, or a library can be compiled.
For those familiar with CMake, a CMakeLists file is provided. This will create the shared library (placed in the CMake folder, names libECHAIM.so) and compile the front end (which will be located in the parent directory, named ECHAIM_User. The example configuration file (configECHAIM.dat) will be edited to list the etc folder located in the parent directory. See ‘First Time Setup’ for more information on the configuration file.

Running the compiled front end from CMake requires moving the shared library to the same directory as the executable and copying the inputs file (ECHAIMInputs.dat) from the User folder to the etc folder.

The aforementioned functions are declared in the ECHAIM.h header file (located in the lib directory), so this file is necessary with the library.

Depending on the compiler being used, some dependencies will need to be specified when compiling. These are listed below with the flag needed for GCC:

- Math library : -lm
- Dynamic linking library : -ldl
- Pthreads library : -lpthread
- libcurl library : -lcurl
- C99 standard : -std=c99

7.4 First time setup

The model requires a configuration file in the current working directory (the directory in which the executable is run). This file must contain the full directory path where the required files are located. The required files are everything from the 'etc' folder and the ECHAIMinputs file from the 'User' folder only if the front end is being used. The configuration file is to be named configECHAIM.dat.

The first time the model is run, and any time it is run and does not find the configuration file, the user will be prompted to enter the path to the required files. This path will be written to a new configuration file in the current working directory.

7.5 Examples of Use

Note that there are other options in compiling and using the provided code. This is just a couple examples for those that need it.

7.5.1 Create a shared object file (UNIX):

In the directory with all the source code (.c) and header files (.h) for the E-CHAIM model itself only

```
gcc –fPIC –c *.c
gcc –shared –o libECHAIM.so *.o -lcurl
```

This will create object files for all of the source code, then compile all of the object files into a shared object (libECHAIM.so).

7.5.2 To use the shared object file:

Place the file (libECHAIM.so) and the E-CHAIM header file (EHCAIM.h) in the same directory as the code which uses the model routines (examples.c for example). Compile the code with the shared object file:

```
gcc examples.c libECHAIM.so –o output
```

Where ‘output’ is the desired name for the executable.

7.5.3 Compiling the source code directly with your code or the front end (UNIX, MinGW [Windows])

Alternatively, you may choose compile all of the source code with your code that uses the model. In a directory with all the model source code (.c) and header files (.h) and your code OR the front end file (ECHAIM.c, located in the ‘User’ folder):

```
gcc *.c –o output -lcurl
```

Where ‘output’ is the desired name for the executable.

7.6 Error Codes

Public functions are provided in the errorCodes.h header file. These functions are used to retrieve a character array which contains possible error codes associated with the run of the model. The public model functions include an option to log these error codes as the final input to the functions. If the input (integer) is set to 1, the model will log these codes, if set to 0, the model will not log the codes. If logging of the codes is set, the function getErrors must be used after the model function call. getErrors takes no inputs and outputs a 2D character array of the size [11xN] where N is the length of the arrays provided to the relevant model function. See the examples file (examples.c) for an example of obtaining and printing the error codes. Note that each character in the array is reserved for a specific error code, e.g index 0 will always contain either code ‘A’, a dash, or be blank. If the character is present, the warning has occurred, if the character is blank, the warning has not occurred, and if the character is a dash, that warning is not applicable to the current output. Descriptions of each error code are provided in Section 9.

7.7 Other Examples

An example file is provided with the source code (examples.c). It contains calls to each of the aforementioned functions and contains comments outlining the output and the required inputs. Example inputs are declared at the top and commented blocks with calls to each of the public functions below.

This can be compiled and run, but it is provided as a reference. Note that the examples.c code uses the E-CHAIM API, and the error codes API, thus the ECHAIM.h header and the errorCodes.h header are both required to compile examples.c (the header files are located in the lib directory).

7.8 Database

As part of the C version of the E-CHAIM model, a database file is provided to the user that was last updated August 29, 2018. Updated versions of this database file are produced daily and are provided on the E-CHAIM website. As part of the API, a function to update the database file automatically is provided (updateLocalDB). See the examples.c code for more information.

7.9 Using the Front-End

When running the front-end executable, the model expects an input file (ECHAIMinputs.dat) located in the directory indicated by the ECHAIMconfig file. An example of the input file is provided in the ‘User’ folder. The header of the example file outlines how it is used.

In short, the file requires a number of inputs: Latitude, longitude, altitude, time, and desired output. Location and time inputs can be provided as a space delimited list or as an interval range (see the header for more information).

The desired outputs include: NmF2 (with or without the storm perturbation), hmF2, Density profile (with or without storm perturbation), density path (with or without storm perturbation). The output will be written to the directory specified in the configuration file. The output file (ECHAIMoutput.dat) will contain the desired outputs, separated by line breaks for each inputted time/location (as applicable).

When multiple outputs are associated with a given location/time (like density profiles) the output will space delimited.

Note: The first line of the output file is the expected dimensions of the data. For example, if a density profile was requested for 3 times/locations and at a resolution of 100 altitude points, the first line would read: 3 100, indicating there should be 3 lines of 100 space delimited data.

7.10 Known Issues

When compiling on Linux, a warning for the implicit declaration of ‘tzset’ may occur. This can be ignored. This is a known issue with using the C99 standard.

8 IDL Distribution

There is also an IDL distribution of the model available; however, as the IDL code is our research code, it does not provide the same level of functionality as the other two model versions. While its output is in agreement with that of the other versions, it does not include many of the features that have been included in the C and Matlab distributions.

8.1 Installation

Installation of the IDL version of E-CHAIM is virtually identical to that for the Matlab version. To install, follow steps 1 and 2 of Section 6.3. Note that the environment variables for the AACGM need to be set for the IDL version of E-CHAIM to run:

AACGM_v2_DAT_PREFIX # Variable Name
The path and prefix of the AACGM coefficient files.
(ex: …\Release_IDL\AACGM COEFS\New\aacgm_coeffs-13-) # Variable Value

IGRF_COEFFS
The location and name of the igrf coefficients file.
(ex: …\Release_IDL\AACGM COEFS\idl_aacgm\igrf13coeffs.txt)

The IDL version requires compiling the AACGM files at startup. A startup file (startup.pro) is provided. Either set this file as your startup file or copy the contents of the file to any previous startup file being used.

It is also necessary to add the model's root folder and subfolders to the IDL and DLM paths. You can do this by either using the IDL paths menu under preferences, or by using the following commands (examples only):

Under Windows:
PREF_SET, 'IDL_DLM_PATH', '…\Release_IDL\NRLMSISE; <IDL_DEFAULT>', /COMMIT

Under UNIX:
PREF_SET, 'IDL_DLM_PATH', '…/Release_IDL/NRLMSISE:<IDL_DEFAULT>', /COMMIT

8.2 Differences from the Matlab Distribution

The IDL distribution does not include the following features that were included with the Matlab distribution:

1) The IDL distribution has not been cleaned, fully commented, or optimized for performance.

2) The IDL distribution does not include a "Map" mode, instead opting for a "Profile" mode, where N vertical profiles are generated for a set of N-length input vectors of Julian time, geographic latitude, and geographic longitude. The profiles each have length corresponding to an M-length altitude vector.
3) The IDL distribution includes optional user inputs of hmF2 and NmF2. These must be vectors of the same length as the input times vector (N).

8.3 Features available in the IDL version

1) Despite the lack of some features, the IDL version of the E-CHAIM model does provide the same default and 'sat' modes of operation as provided by the Matlab and C codes.
2) The IDL version of the code allows for user input of hmF2 or NmF2 (similar to the IRI).

Please contact the project lead at david.themens@gmail.com for assistance using the IDL distribution.

8.4 Examples of Use

Interfacing with the IDL code is virtually identical to that for the Matlab code but with slight differences due to differences in language syntax; as such, please see the Matlab section prior to proceeding with the following.

8.4.1 Using the Default Mode

For the default mode, one may use the following call:

```
YY = ECHAIM(latitudes, longitudes, julday(month,day,year,hour,minute,second), altitude)
```

In this case, latitude, longitude, month, day, year, hour, minute, and second are all vectors of the same length. Altitude is a vector of desired output profile altitudes of separate length. The output of the function is a structure with the following keywords:

MLON: AACGM Longitude
MLAT: AACGM Latitude
NmF2:NmF2
LTs: Local Time
MLT: Magnetic Local Time
hmF2:hmF2_out_arr,
F107_27: 27-day smoothed F10.7 flux
hmF1: hmF1 peak scale thickness
HBot: F2-layer peak scale thickness
HF1:F1-layer peak scale thickness
HE: E layer peak scale thickness
DENS: Densities array
ALTS: Altitudes array
JULS: Julian dates array

HTop: Topside semi-Epstein layer thickness
FLAGS: Error codes (see Section 9)
AURORAL_Flux: Mean Auroral Flux used by the precipitation module (units of ergs/s/cm^2)
AURORAL_E_0: Mean Auroral Energy used by the precipitation module (units of keV)

The DENS, ALTS, and JULS keywords correspond to arrays of size N_times x N_altitudes. The altitudes and densities here are provided in an array format simply for ease of contour plotting. All of the remaining keywords correspond to vectors of the same length as the input time/latitude/longitude vector.

8.4.2 Using the Satellite Mode

For the satellite mode, one may use the following call:

YY = ECHAIM(latitudes, longitudes, julday(month,day,year,hour,minute,second), altitude, /satellite)

In this case, latitude, longitude, month, day, year, hour, minute, second, and altitude are all vectors of the same length. The output of the function is a structure with the following keywords:

MLON: AACGM Longitude
MLAT: AACGM Latitude
NmF2:NmF2
LTs: Local Time
MLT: Magnetic Local Time
hmF2:hmF2_out_arr,
F107_27: 27-day smoothed F10.7 flux
hmF1: hmF1 peak scale thickness
HBot: F2-layer peak scale thickness
HF1:F1-layer peak scale thickness
HE: E layer peak scale thickness
DENS: Densities vector
ALTS: Altitudes vector (redundant in the satellite call)
JULS: Julian dates vector (redundant in the satellite call)
HTop: Topside semi-Epstein layer thickness
FLAGS: Error codes (see Section 9)
AURORAL_Flux: Mean Auroral Flux used by the precipitation module (units of ergs/s/cm^2)
AURORAL_E_0: Mean Auroral Energy used by the precipitation module (units of keV)

All of these keywords correspond to vectors of the same length as the input time/latitude/longitude/altitude vector.

8.4.3 Using the Storm_NmF2 Option

To use the NmF2 storm model in place of the quiet time model, simply add the /storm_nmF2 keyword to the function call.

8.4.4 *Using the Precipitation and D Region Options*

To use the precipitation density model, simply add the /precip keyword to the function call. Note: the /precip keyword will only affect the electron density profile outputted by the model under structure key “DENS”. It will not affect any other output parameters. For example, this means that the F2 peak density (NmF2) from the model output may not be the same as the maximum of the electron density profile outputted under “DENS”.

The D region model can be includes using the /dregion flag.

8.4.5 *Database Updating*

Keywords have been added to allow the user the choice of whether the model will self update the local database file. By default, the model will not attempt to update the database. If the user supplies the ‘/update’ keyword, the model will check the local database to see if it needs to be updated, if so, the latest database file will be downloaded from the E-CHAIM website and will overwrite the current local database file. If the user provided the keywords ‘/forceUpdate’, the model will download and overwrite the local database file with the latest file on the E-CHAIM website without checking the local file first.

8.5 Known Issues

1) There is a known problem where some users may be force-exited from their IDL session when using the /update or /forceupdate keywords. This is a problem in the IDLNetURL tools and we are currently unaware of a way to fix the issue; as such, we have modified the database update code to first attempt to use a local version of WGET before attempting to use the IDLNetURL-based wget.pro routine. If you experience this forced-exit problem, we recommend that you install a version of WGET (https://www.gnu.org/software/wget/) on your machine and add the path to that utility in your system’s Path Environmental Variable. Once this is complete, the model code will automatically bypass the use of the IDLNetURL utility and use your native WGET instead.

9 Error Flags

All three versions of the E-CHAIM code provide some form of error flag record. In the case of the IDL and Matlab codes, this is simply an output structure tag that contains a string vector of all relevant flags for each requested time/location.

A: Measured AE index was not available for at least one requested input. If the storm model was requested, a synthetic AE index, based on PC Index was used. If the precipitation model was requested, a PC index-based model has been used in place of the standard AE-based model.

B: PC index was not available for at least one requested input. Synthetic AE was generated using PC = 0.

C: At least one input is beyond the available IG indices. Forecasted IG12 has been used for these calculations.

D: At least one input is beyond the available forecasted F10.7 indices. E-CHAIM output has been set to NaN.

E: At least one input is beyond the available F10.7 indices. NOAA Forecasted F10.7 has been used for these calculations instead.

F: At least one input is beyond the available forecasted IG12 indices. E-CHAIM output has been set to NaN.

G: At least on input is below the recommended boundary of the model (50 degrees geomagnetic latitude). That output should be used with caution.

H: At least one input is below the absolute lower boundary of the model (45 degrees geomagnetic latitude). That output has been set to NaN.

I: At least one input is outside the available range of the AP index.\nAP has been set to the all time median value (9.1) for this calculation.

J: At least one input is outside the available range of the DST index.\nDST has been set to the all time median value (-10.2) for this calculation.

10 Rules of the Road

When using the Empirical-Canadian High Arctic Ionospheric Model (E-CHAIM) for scientific purposes, we ask users to adhere to the following guidelines:

1) List the version number for the version of the E-CHAIM code you are using (ex: v2.1.0).
2) While we do not ask to be included as co-authors to your publications using the model, we do ask that you let us know after you have published a paper that uses the model. This goes a long way to helping us justify further funding support to keep the model available and free.
3) Please include the following acknowledgement in publications using the model: "E-CHAIM development was supported under Defence Research and Development Canada contract number W7714-186507/001/SS and is maintained by the Canadian High Arctic Ionospheric Network (CHAIN) with operations support from the Canadian Space Agency."