USERS MANUAL TO THE FAST INFRARED RADIATIVE TRANSFER MODEL

VERSION 1.0

TOM GREENWALD (AWG PROXY DATA PROJECT MANAGER & SCIENCE LEAD)
ALLEN HUANG (AWG PROJECT PI)
ERIK OLSON
MAT GUNSHOR
RUPALI MAHAJAN

COOPERATIVE INSTITUTE FOR METEOROLOGICAL SATELLITE STUDIES
UNIVERSITY OF WISCONSIN - MADISON

31 JANUARY 2007

TABLE OF CONTENTS

LIST OF FIGURES **3**

LIST OF TABLES **4**

LIST OF ACRONYMS **5**

1 INTRODUCTION **6**

1.1 PURPOSE 6
1.2 WHO SHOULD USE THIS MANUAL? 6
1.3 SECTION SUMMARIES 6

2 FAST MODEL OVERVIEW **6**

2.1 REQUIREMENTS 6
2.2 DEVELOPMENT AND IMPLEMENTATION TEAM 6
2.3 ALGORITHM 8
2.4 MODEL DEVELOPMENT HISTORY 11

3 FAST MODEL IMPLEMENTATION **12**

3.1 CODE AND DATA ACCESS 12
3.2 DATA FILE DESCRIPTION 12
3.2.1 Input Data 12
3.2.2 Ancillary Data 13
3.2.3 Look-up Tables 13
3.2.4 Outputs 14
3.3 CODE FEATURES AND REQUIREMENTS 15
3.4 EXAMPLE 17
3.5 USAGE IN CLUSTER ENVIRONMENT 18

4 REFERENCES **19**

APPENDIX A: EMISSIVITY MODEL FOR WATER SURFACES **20**

APPENDIX B: ABI SRFS FOR CHANNELS 8-16 .... **20**

APPENDIX C: STANDARD PRESSURE LEVELS FOR FPLOD **21**

APPENDIX D: SCRIPT FILE TO RUN FAST MODEL ON A CLUSTER **22**

LIST OF FIGURES

Figure 1. Top level flow diagram for the IR RTM in producing a GIFTS spectrum. NWP model input data include temperature (T), water vapor mixing ratio (q), cloud mixing ratio (q_c) and particle effective diameter (D_e).

Figure 2. Top level flow diagram for the production of TOA infrared radiance datasets.

Figure 3. Spectral variation of emissivity for a calm water surface.

Figure 4. Spectral response functions defined for the ABI.

LIST OF TABLES

Table 1. Possible cloud configurations in FIRTM2.

Table 2. Characteristics of ABI channels simulated in this work (adapted from Schmit et al. 2005).

Table 3. Characteristics of FPLOD coefficient files.

Table 4. Variables contained in the simulated GIFTS TOA radiance output file.

Table 5. Variables contained in the ABI TOA radiance output file.

Table 6. Brief description of selected routines and modules in IR RTM code.

LIST OF ACRONYMS

ABI	Advanced Baseline Imager
ATBD	Algorithm Theoretical Basis Document
AWG	Algorithm Working Group
CIMSS	Cooperative Institute for Meteorological Satellite Studies
CRTM	Community Radiative Transfer Model
DISORT	DIScrete Ordinate Radiative Transfer
FIRTM2	Fast Infrared Radiative Transfer Model 2
FOV	Field Of View
FPLOD	Fast Pressure Layered Optical Depth
GEO	Geosynchronous Earth Orbit
GIFTS	Geosynchronous Imaging Fourier Transform Spectrometer
GOES-R	Geostationary Operational Environmental Satellite R series
IR	Infrared
IRSSEM	Infrared Sea Surface Emissivity Model
LEO	Low Earth Orbit
MODIS	MODerate Resolution Imaging Spectroradiometer
netCDF	network Common Data Format
NWP	Numerical Weather Prediction
OPTRAN	Optical Path TRANsmittance
RTM	Radiative transfer model
SRF	Spectral response function
TOA	Top of Atmosphere
UTC	Coordinated Universal Time
WRF	Weather Research and Forecasting

1 INTRODUCTION

1.1 *Purpose*

The purpose of this manual is to familiarize users with the fast IR forward model developed at CIMSS for GOES-R AWG Proxy Data Set activities. Specifically, the manual will enable the user to acquire the model code, understand how it functions, and execute the code for producing data sets of TOA high-resolution IR radiance spectra (587.3 cm^{-1} to 2349.5 cm^{-1}) and TOA radiances for ABI channels 8-16 from 3D NWP model output.

1.2 *Who Should Use This Manual?*

This manual is intended for members of the GOES-R AWG who will use simulated ABI radiance datasets for testing algorithms and products.

1.3 *Section Summaries*

The next section provides a summary of the IR fast forward model and includes a brief history of its development. Section 3 describes in detail the fast model code and its associated data sets and shows how to set up and execute the code.

2 FAST MODEL OVERVIEW

2.1 *Requirements*

The goal of the Proxy Data Set project is the creation and distribution of simulated datasets to support the AWG. The requirements for this project are coupled to the ongoing GOES-R Risk Reduction efforts at CIMSS from which the software tools and databases used in the current effort were initially developed. Elements of the Proxy Data Set requirements relevant to the forward model include

- Simulation of ABI measurements
- Enhancement of the fast IR forward model
- Development of cloud property databases
- Production of IR land surface emissivity databases

2.2 *Development and Implementation Team*

The following is a list of the team members and their roles:

Dr. Tom Greenwald
CIMSS, University of Wisconsin
Proxy Data program manager
Science team lead
tom.greenwald@ssec.wisc.edu
Tel: 608-263-3629

Dr. Hung-Lung (Allen) Huang
CIMSS, University of Wisconsin
AWG Principal Investigator
allen.huang@ssec.wisc.edu
Tel: 608-263-5283

Dr. Ping Yang
Department of Atmospheric Sciences, Texas A&M University
Radiative transfer model support
pyang@ariel.met.tamu.edu
Tel: 979-845-4923

Zhibo Zhang
Department of Atmospheric Sciences, Texas A&M University
Radiative transfer model development
zzbatmos@ariel.met.tamu.edu
Tel: 979-845-0177

Jianguo Niu
Department of Atmospheric Sciences, Texas A&M University
Radiative transfer model development
niu@ariel.met.tamu.edu

Erik Olson
CIMSS, University of Wisconsin
Implementation, optimization and testing
erik.olson@ssec.wisc.edu
Tel: 608-262-9538

Mat Gunshor
CIMSS, University of Wisconsin
Forward model support
mat.gunshor@ssec.wisc.edu
Tel: 608-263-1146

Leslie Moy
CIMSS, University of Wisconsin
Forward model support
leslie.moy@ssec.wisc.edu
Tel: 608-262-7307

Dr. Bryan Baum
SSEC, University of Wisconsin
Cloud optical properties support
bryan.baum@ssec.wisc.edu
Tel: 608-263-3898

Dr. Rupali Mahajan
CIMSS, University of Wisconsin
Documentation development
rupali.mahajan@ssec.wisc.edu
Tel: 608-265-2335

2.3 *Algorithm*

The fast IR RTM was designed specifically to calculate TOA radiances for the 3074 channels of the GIFTS, although certain parts of the model can be used in more general applications. The GIFTS spectrum is composed of two separate bands: 587.406 – 1174.812 cm^{-1} (1025 channels) and 1174.812 – 2349.624 cm^{-1} (2049 channels). Because two of the channels have identical central wavenumbers, the first channel in band 2 is ignored and thus radiances are computed for a total of 3073 channels.

The IR RTM itself is constructed of several different components (see Figure 1). The first is FPLOD, which is a statistical-regression-type model that predicts the polychromatic optical depth due to gases (water vapor, ozone, oxygen) for each atmospheric layer and is very similar in concept to OPTRAN (McMillin et al. 1995). Like OPTRAN, FPLOD assumes 101 fixed pressure levels (see Appendix A), which means all atmospheric NWP model variables must be interpolated to these levels. Another important consideration is that FPLOD requires a minimum water vapor mixing ratio of 10^{-3} g/kg since that is the lowest value in the training sets used for generating the statistical coefficients. Using smaller mixing ratios can cause unpredictable behavior.

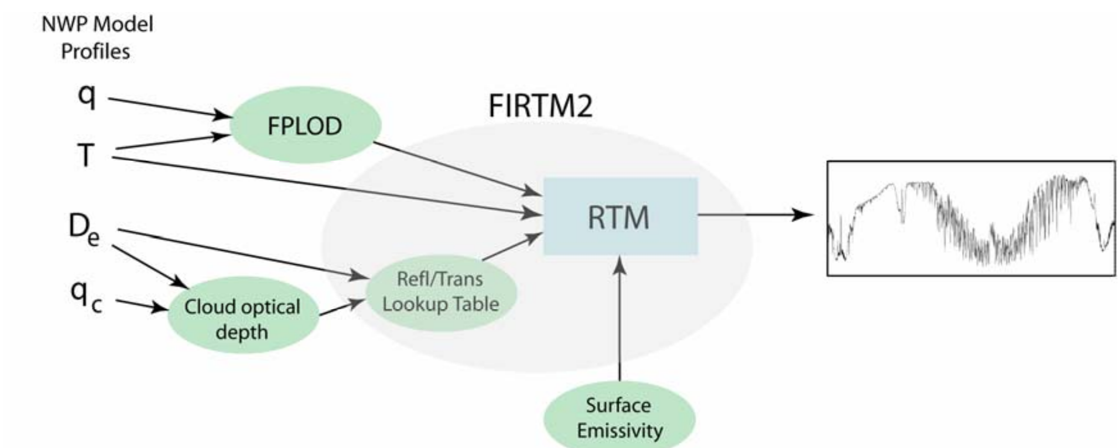

Fig. 1. Top level flow diagram for the IR RTM in producing a GIFTS spectrum. NWP model input data include temperature (T), water vapor mixing ratio (q), cloud mixing ratio (q_c) and particle effective diameter (D_e).

The second major component of the fast model is FIRTM2, which computes the radiative transfer to obtain the TOA radiance for either clear sky or cloudy sky conditions (Niu et al. 2006). The main features of the model are:

- Total of 5 atmospheric layers
- Up to two cloud layers are allowed
- Cloud reflectance and transmittance are parameterized

The cloud radiative properties are computed beforehand based on DISORT calculations using scattering properties for ice and liquid (Baum et al. 2005). These calculations are stored in separate lookup tables for ice and liquid phases (see Section 3.2.3 for more details.) More information on FPLOD, FIRTM2 and other aspects of the fast IR RTM can be found in the GOES-R Risk Reduction ATBD at http://cimss.ssec.wisc.edu/goes_r/risk-reduction.html (access is currently restricted.)

Because FIRTM2 allows only a limited number of cloud layers, an input atmospheric profile must be preprocessed before it can be used by FIRTM2. Cloud layers are assigned as either liquid or ice; layers with mixed phase conditions are not currently supported (see Table 1). The process begins by assigning each hydrometeor type (i.e., liquid cloud, ice cloud, rain, graupel, snow) in each layer a value of 1 for liquid or 2 for ice (the exception is rain, which is assigned 1.5.) These values are weighted by the mass (i.e., mixing ratio) of the hydrometeor type and if the total value is greater than 1.5 it is classified as an ice cloud, otherwise it is a liquid cloud. Next, the cloud layer is converted to visible optical depth (different formula exist for ice and liquid). Finally, the cloud layers are reorganized into one or two layers and the total optical depths and representative particle effective diameters are computed.

Table 1. Possible cloud configurations in FIRTM2.

Cloud category index	Description
0	Clear sky
1	One ice cloud layer
2	One liquid cloud layer
3	Ice cloud over liquid cloud
4	Ice cloud over ice cloud
5	Liquid cloud over liquid cloud
6	Liquid cloud over ice cloud

One refinement made to FIRTM2 (which is currently an option) is in how the downward radiance ($I_\downarrow$) at the surface is computed for clear sky conditions. Assuming a Lambertian surface, $I_\downarrow$ at angle μ_o can be expressed as

$$I_\downarrow(\mu_o) = \Im(\mu_o) F_\downarrow r ,$$

where $\Im$ is the atmospheric transmittance from the surface to the TOA, r is the surface reflectance and μ is the cosine of the zenith angle. The improvement here is in approximating the downward flux $F_\downarrow$ by two Gaussian quadrature points:

$$F_\downarrow = 2\pi \int_0^1 \mu I(\mu) d\mu \cong 2\pi\left[0.2I(\mu = 0.3584) + 0.3I(\mu = 0.8480)\right].$$

Surface radiative properties are needed by FIRTM2 in order to compute TOA radiance. For simplicity, the land surface is assumed to be Lambertian so there is no angular dependence of the assigned surface emissivity. Our source for the land surface emissivity is the global IR database developed at CIMSS (see http://cimss.ssec.wisc.edu/iremis.) The database has 0.05^{o} spatial resolution and is derived from a combination of high spectral resolution laboratory measurements of selected materials and MODIS observed land surface emissivities at six wavenumbers ranging from 699.3 to 2702.7 cm^{-1}. More information on this database is available in Section 3.2.1.

The model describing the emissivity of water surfaces is only a function of wavenumber and is shown in Appendix A. The emissivity of water surfaces, however, also depends on zenith angle and wind speed; therefore, there are plans to utilize a more sophisticated model. One model being considered is IRSSEM, which is currently part of NOAA's CRTM.

Generating TOA radiances for the ABI thermal IR channels (see Table 2) involves integrating radiance across the GIFTS TOA radiance spectrum while utilizing the SRF represented for each individual ABI channel. The SRFs used are shown in Appendix B. The steps in the process can be summarized as follows:

1. Resample SRF to spectral resolution of GIFTS data. This is done using linear regression and the central wavenumber of each high spectral band. For these purposes, high spectral bands are considered to be monochromatic radiance measurements and their own spectral responses are ignored.

2. Convolve High Spectral Radiance data with the multi-spectral SRF. The High Spectral Radiance *R* at wavenumber *v* is convolved with the multi-spectral SRF *S* as:

$$L_i = \frac{\int_{v_1}^{v_2} R(v) S_i(v) dv}{\int_{v_1}^{v_2} S_i(v) dv}$$

where *i* is ABI channel number (which can be arbitrary) and *L* the convolved broadband (simulated) radiance. This process has come to be known as a convolution, or "convolving high spectral data with a broadband SRF." It is perhaps more correct mathematically to call it a weighted average, where the SRF values interpolated at the high spectral resolution frequency index are the weights and the denominator in that equation is a normalization factor (Tobin et al., 2006).

3. The outcome of this step is a single radiance value (milliWatts/meter^2/steradian/centimeter^{-1}) per instrument FOV. Details are described in the ATBD (http://cimss.ssec.wisc.edu/goes_r/risk-reduction.html).

Table 2. Characteristics of ABI channels simulated in this work (adapted from Schmit et al. 2005).

Channel	Wavelength range (μm)	Wavenumber range (cm^{-1})	Sample use
8	5.77-6.6	1515.15-1733.1	High-level water vapor, winds
9	6.75-7.15	1398.6-1481.5	Mid-level water vapor, winds
10	7.24-7.44	1344.09-1381.2	Low-level water vapor and SO_2
11	8.3-8.7	1149.4-1204.8	Stability, cloud phase, dust, SO_2, rainfall
12	9.42-9.8	1020.4-1061.6	Ozone, turbulence, winds
13	10.1-10.6	943.40-990.10	Surface and clouds
14	10.8-11.6	862.07-925.93	Imagery, SST, clouds, rainfall
15	11.8-12.8	781.25-847.46	Total water, ash, SST
16	13.0-13.6	735.29-769.23	Air temperature, cloud heights and amounts

2.4 *Model Development History*

Jim Davies developed most of the code for the original fast IR RTM, which was written in Fortran 77. Jianguo Niu supplied the FIRTM2 code, while Hal Woolf and Leslie Moy developed the FPLOD code.

In late 2005 the complete set of code was overhauled and converted to Fortran 95. Due to earlier hardware limitations, it was not possible to load the input NWP model data and the FIRTM2 lookup tables into memory. The F95 version of the code loads both data sets into memory. In addition, the new code was restructured for greater flexibility.

Significant developments in both the IR RTM and its code include:

- Improved calculation of the thermal source functions in FIRTM2 (Nov 2005)
- Included ability to generate output TOA radiance data files in netCDF (March 2006)
- Utilized the latest ice scattering properties (Baum et al. 2005) and regenerated lookup tables for FIRTM2 (April 2006)
- Incorporated improved calculation of downward radiance at surface in FIRTM2 (April 2006)
- Added ability to read IR land surface emissivity database (May 2006)
- Included option for simulating an instrument in LEO (Oct 2006)
- Added option to ignore clouds in the NWP model data and compute TOA radiance (Nov 2006)
- Added two more possible cloud categories (index numbers 5 and 6) (Dec 2006)

Current model input options are:

- TOA radiance calculations can be done in normal mode (i.e., with clouds) or in clear sky mode (i.e., clouds are ignored).
- Improved calculation of the downward radiance at the surface may be performed (*Caveat: the code is currently not optimized and will slow the calculations down significantly*)
- Can simulate a satellite instrument in either GEO or LEO (see Section 3.2.1 for details).

3 FAST MODEL IMPLEMENTATION

3.1 *Code and Data Access*

The code and data for the IR RTM and ABI channel generation will soon be available at the CIMSS AWG activities website (http://cimss.ssec.wisc.edu/goes_r/awg.html).

3.2 *Data File Description*

3.2.1 Input Data

The primary input data for the fast IR RTM is 3D NWP model data from the WRF model. The data files are provided in netCDF. Model variables included in these files are described in the *Fulldisk Atmospheric Profile Dataset Description and Documentation*. If users intend to utilize other NWP model data, the model variables must be interpolated to the standard 101 pressure levels required by FPLOD (see Appendix C.)

Coefficient files needed by FPLOD are also required input data. These files are binary and direct access (see Table 3). Each record corresponds to one GIFTS channel and contains the coefficients for 100 atmospheric layers. Separate files are provided for each of the GIFTS two bands (band 1 has 1025 channels and band 2 has 2049 channels.)

Another input file is the IR land surface global emissivity database. These files are also binary and direct access. Each record contains a single latitude/longitude point along with slope and intercept data (7 points each) for computing surface emissivity between 8 spectral inflection points (699.3007, 925.9259, 1075.269, 1204.819,1315.790, 2000.000, 2325.581, 2702.703 cm^{-1}). The file record length is 64 bytes. Because the spatial resolution of the databases is 0.05^{o}, the files can become quite large (≈1 GB). However, version 2.0 of the database is now available in netCDF and it does not save slopes and intercepts but instead utilizes the actual emissivity values at 10 inflection points. This significantly reduces the size of the databases (see http://cimss.ssec.wisc.edu/iremis.)

Table 3. Characteristics of FPLOD coefficient files.

Filename	No. coeffs	Record length (bytes)	Description
giftsfr1.dry	9	3600	Oxygen – band 1
giftsfr1.wco	5	2000	Water vapor continuum – band 1
giftsfr1.wtl	3	1200	Water vapor line – band 1
giftsfr1.wts	12	4800	Water vapor shape – band 1
giftsfr1.ozo	10	4000	Ozone – band 1
giftsfr2.dry	9	3600	Oxygen – band 2
giftsfr2.wco	5	2000	Water vapor continuum – band 2
giftsfr2.wtl	3	1200	Water vapor line – band 2
giftsfr2.wts	12	4800	Water vapor shape – band 2
giftsfr2.ozo	10	4000	Ozone – band 2

An available option for the IR RTM is the ability to compute TOA radiances for an instrument in LEO. However, this requires an input file that supplies the observation zenith angle information for the instrument. The only requirements for the file is that it is in netCDF and at a minimum has the following attributes and variables:

- Global attribute called "OUTPUT_TIME" that corresponds to the time of the satellite overpass and which has the same format as the input WRF model data file.
- Dimensions for the size of the satellite swath are called "acrosstrack" (the number of samples or pixels in a scan line) and "alongtrack" (the number of scan lines).
- Three variable grids named "Lat", "Lon", and "szen" (contains observation zenith angles).

See routine *read_LEOdat_ncdf* in module *ncdf_util* for additional information.

The generation of ABI radiances requires SRF functions, which are contained in the input file *dasrfabi.716*. The file has a 3-line header and is in ASCII format.

3.2.2 Ancillary Data

IR RTM requires two ancillary files. The first is the file, *gifts_chans.dat*, which contains the central wavenumbers for the GIFTS channels. The file is in ASCII format. The second file is *RefProf1.dat*, which contains the reference temperature and water vapor profiles needed by FPLOD. This file is also in ASCII format.

3.2.3 Look-up Tables

Two look-up tables are needed by FIRTM2, each of which contain the precomputed cloud reflectances and transmittances for ice particles and water droples. The file for ice particles, i.e., *FIRTM2_baum05.dat*, is binary and direct access. The file record length is determined by the number of spectral points in the table (variable NUM_ICE) and the number of observation zenith angle points (variable THETMAX) and includes both the reflectance and transmittance data. Since NUM_ICE = 3151 and THETMAX = 9, the file record length is 226872 bytes. The spectral range of this table is 100 to 3250 cm^{-1} with an interval of 1 cm^{-1}. The observation zenith angle data range from 0^o to 80^o in 10^o increments. The number of records in *FIRTM2_baum05.dat* is determined by the number of visible optical depth values in the table (TAUMAX_ICE) and the number of D_e points in the table (DMAX_ICE). Since TAUMAX_ICE = 25 and DMAX_ICE = 18, the total number of records is 450. For ice particles, D_e ranges from 10 μm to 180 μm and is in 10 μm increments. On the other hand, the cloud optical depth values range between 0.01 and 10 at irregular intervals.

The second table of reflectances and transmittances for water droplets is contained in file *unf_water_c.dat*. This file is also binary and direct access. The spectral range of the table is 587.30 to 2349.50 cm^{-1} with an interval of 0.6 cm^{-1} (NUM_LIQ = 2938.) The record size for this file is therefore 211536 bytes. Since TAUMAX_LIQ = 22 and DMAX_LIQ = 13, the total number of records is 286. For ice particles, D_e ranges from 2 μm to 100 μm in irregular increments. The cloud optical depth values range between 0.5 and 110 at irregular intervals.

3.2.4 Outputs

Two netCDF output files are created by the forward model system. The first file contains simulated GIFTS TOA radiance spectra for each grid point of the 2D WRF model domain. For fulldisk WRF simulations the filenames have the following convention:

TOA_*yyyy_ddmm_HHMM_xx_yy*.cdf

where *yyyy* is year, *dd* is day of the month, *mm* is month of the year, *HHMM* is in UTC, and *xx* and *yy* are the column and row indices for an individual “data cube.” Table 4 provides a summary of the output variables. A more complete description can be found in the *Simulated Top-of-Atmosphere Radiance Data Set Description*.

The second file contains the simulated TOA radiances for ABI channels 8-16 and has a similar filename convention:

sim_abi_*yyyy_ddmm_HHMM*utc.nc

Because only 9 channels are included in ABI files, the separate data cubes generated for the fulldisk simulation are combined into one file. Table 5 gives a summary of the output variables.

Table 4. Variables contained in the simulated GIFTS TOA radiance output file.

Variable name	Description	Units
wvn	Wavenumber	1/cm
Altitude	Terrain height	Meters
lat	Latitude	degrees north
lon	Longitude	degrees east
CLD_INDT	Cloud indicator	Category
CP4	Upper cloud top indicator	hPa
CTau4	Upper cloud visible optical depth	Dimensionless
CDe4	Upper cloud particle effective diameter	Microns
CP2	Lower cloud top indicator	hPa
CTau2	Lower cloud visible optical depth	Dimensionless
CDe2	Lower cloud particle effective diameter	Microns
theta	Observation zenith angle	Degrees
radiance	Spectral TOA radiance	mW/(m^{2}steradian cm^{-1})

Table 5. Variables contained in the ABI TOA radiance output file.

Variable name	Description	Units
lat	Latitude	Degrees north
lon	Longitude	Degrees east
altitude	Terrain height	Meters
landmask	Land classification	Category
CLD_INDT	Cloud indicator	Category
ABI_rad_band_08	Band 8 Radiance	mW/(m^{2}steradian cm^{-1})
ABI_rad_band_09	Band 9 Radiance	mW/(m^{2}steradian cm^{-1})
ABI_rad_band_10	Band 10 Radiance	mW/(m^{2}steradian cm^{-1})
ABI_rad_band_11	Band 11 Radiance	mW/(m^{2}steradian cm^{-1})
ABI_rad_band_12	Band 12 Radiance	mW/(m^{2}steradian cm^{-1})
ABI_rad_band_13	Band 13 Radiance	mW/(m^{2}steradian cm^{-1})
ABI_rad_band_14	Band 14 Radiance	mW/(m^{2}steradian cm^{-1})
ABI_rad_band_15	Band 15 Radiance	mW/(m^{2}steradian cm^{-1})
ABI_rad_band_16	Band 16 Radiance	mW/(m^{2}steradian cm^{-1})

3.3 *Code Features and Requirements*

The IR RTM code and the code used to generate ABI TOA radiances are two separate programs. However, there are future plans to merge the two programs. Figure 2 illustrates the general flow of the process. The main program for the IR RTM code is contained in file *irfrtm.f90*, while the main program for the ABI radiance generation is contained in file *goesrad.f90*. Other important routines and their locations and descriptions are given in Table 6.

Primary testing for the IR RTM code was done in a Linux environment on a 64-bit Opteron cluster and a dual-processor Xeon system. To maximize its portability, the code was tested with various compilers. It was successfully compiled and executed using the Intel F90 (v9.1) and Gnu F95 (v.4.0.3) compilers on the Opteron system and using the Portland Group F90 (v5.2-1) and Gnu F95 compilers on the Xeon system. Scripts to build executables for the Intel F90, Gnu F95, and Portland Group F90 compilers are provided in the files *build_ifort*, *build_g95*, and *build_pgf90*, respectively. It should be mentioned that because compilers sometimes use different keywords for the file format in the Fortran OPEN statement (e.g., 'UNFORMATTED' vs. 'BINARY'), the correct keyword for a given compiler must be set in the file *open_file_format.f90* before compilation.

Scripts to build an executable for the ABI code will also be provided as a makefile (*makefile.linux*). Both sets of code require the netCDF libraries be installed on the user's system. Version 3.6.1 of the netCDF libraries was used in our tests. In addition, users can set the precision of floating point numbers used in both sets of code in the module *type_kinds* by altering the floating point kind index IFP. This module also allows one to alter the integer precision but this is not supported in the current version of the IR RTM code.

Memory requirements for the IR RTM code will depend entirely on the size of the input NWP model data set to be processed. For a 3D WRF model grid with 128 x 128 horizontal grid points and 101 vertical levels, the amount of resident memory needed can be expected to be approximately 500 MB.

Table 6. Brief description of selected routines and modules in IR RTM code.

File name	Routine or module name	Description
irfrtm.f90	irfrtm	The main driver for the IR RTM. Loads the necessary data sets, computes GIFTS TOA radiances and outputs the data into a netCDF file.
forward_rtm.f90	GIFTSrad_fwd	Computes GIFTS TOA radiances for each grid point in the NWP model domain.
fplod.f90	fplod_gifts	Predicts layer gas optical depths for the GIFTS channels based on the FPLOD algorithm.
firtm2.f90	firtm2	Computes TOA radiance and brightness temperature at a single wavenumber for a scattering atmosphere with at most two cloud layers.
cloud_properties.f90	getCloudsMM5	Obtains cloud optical and cloud top properties from the input NWP microphysical profile and determines how to vertically redistribute cloud properties.
geometry.f90	observation_zenith_angle_leo	Assigns observation zenith angle for an input NWP model grid point given a 2D grid of latitudes, longitudes, and observation zenith angles from an instrument in LEO. Current algorithm searches for nearest neighbor.
open_file_format.f90	open_file_format	Assigns format type for binary input files according to compiler
Type_Kinds.f90	Type_Kinds	Holds specification kinds for variable declaration.

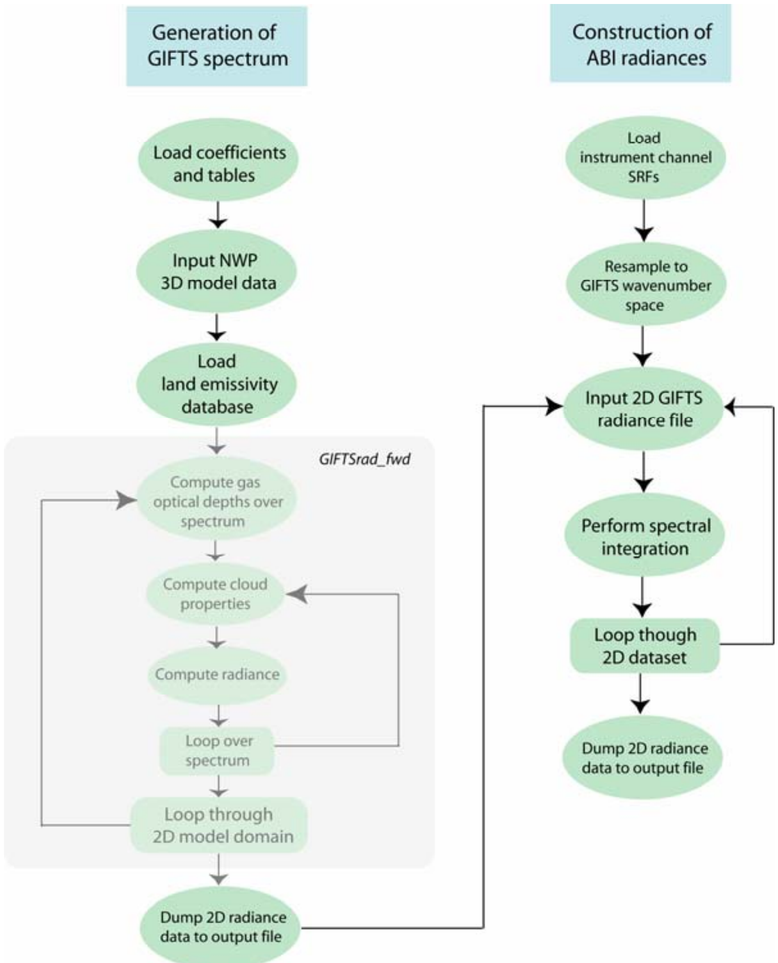

Fig. 2. Top level flow diagram for the production of TOA infrared radiance datasets.

3.4 *Example*

A specific example is provided for building executables for the IR RTM code and ABI TOA radiance generation code and running the executables using input list files. These files are in ASCII format and supply input/output filenames and option settings to the program through standard input.

On a Linux system, for example, the code can be compiled using the Intel F90 compiler and executed by typing the following at the command line:

```
./build_ifort
./irfm.e < input.lis
```

The contents of a typical `input.lis` file might look like this:

```
/home/username/data/giftsfr1.dry
/home/username/data/giftsfr2.dry
/home/username/data/giftsfr1.ozo
```

```
/home/username/data/giftsfr2.ozo
/home/username/data/giftsfr1.wco
/home/username/data/giftsfr2.wco
/home/username/data/giftsfr1.wtl
/home/username/data/giftsfr2.wtl
/home/username/data/giftsfr1.wts
/home/username/data/giftsfr2.wts
/home/username/data/RefProf1.dat
/home/username/data/FIRTM2_baum05.dat
/home/username/data/unf_water_c.dat
/home/username/data/gifts_chans.dat
/home/username/input/0625_2003_1200utc_2_2.nc
/scratch/Emissivity_data/global_emis_slopeInterc_MYD11C3.A2003335_wvnu
m.bin.le
GEO      ! Satellite type - GEO or LEO. For GEO supply satellite
           subpoint longitude on next line; for LEO supply filename
           and full path for netcdf file
-89.0
NORMAL   ! Mode of radiance calculations: NORMAL is for all-sky
           radiances and CLEAR is for clear sky only
N        ! Option for improved calculation of downwelling radiance at
           surface (clear sky only); Y - Yes, N - No
/home/username/output/TOA_0625_2003_1200_2_2.cdf
```

To generate the ABI TOA radiance file, one runs the makefile provided to build the executable. The executable can be run using a different input list file in the same manner described above. A typical input list file for processing an entire fulldisk simulation (12 x 14 data files) might look like this:

`/scratch/2003_0625_0000_TOA/`	Input file directory
`/scratch/2003_0625_0000_ABI/`	Output file directory
`2003`	Year
`0625`	Month/Day
`0000`	UTC
`TOA`	File prefix
`1 12`	Start and end columns
`1 14`	Start and end rows
`0`	Empty variable
`10`	No. of ABI bands
`abirad_12x14_0000.log`	Output log file
`ABI data from simulated GIFTS radiances`	Comment for output file
`dasrfabi.716`	File containing ABI SRFs

3.5 *Usage in Cluster Environment*

A sample script is given in Appendix D for executing the fast model on a computer cluster.

4 REFERENCES

Baum, B. A., A. J. Heymsfield, P. Yang, and S. T. Bedka, 2005: Bulk scattering properties for the remote sensing of clouds. Part I: Microphysical Data and Models. *J. Appl. Meteor.*, **44**, 1885-1895.

McMillin L. M., L. J. Crone, M. D. Goldberg, and T. J. Kleespies, 1995: Atmospheric transmittance of an absorbing gas, 4. OPTRAN: A computationally fast and accurate transmittance model for absorbing gases with fixed and variable mixing ratios at variable viewing angles. *Appl. Opt*., **34**, 6269–6274.

Niu, J., P. Yang, H.-L. Huang, J. E. Davies, J. Li, B. A. Baum, and Y. X. Hu, 2006: A fast infrared radiative transfer model for overlapping clouds. *J. Quant. Spect. Radiat. Transfer,* in press.

Schmit, T. J., M. M. Gunshor, W. P. Menzel, J. J. Gurka, J. Li, and A. S. Bachmeier, 2005: Introducing the next-generation advanced baseline imager on GOES-R. *Bull. Amer. Meteor. Soc*., **8**, 1079-1096.

Tobin, D. C., H. E. Revercomb, C. C. Moeller, and T. S. Pagano, 2006: Use of Atmospheric Infrared Sounder high-spectral resolution spectra to assess the calibration to Moderate resolution Imaging Spectroradiometer on EOS Aqua. *J. Geophys. Res*., **111**, D09S05.

APPENDIX A: EMISSIVITY MODEL FOR WATER SURFACES

Specification of surface emissivity for ocean and fresh water surfaces is done using a simple model shown in Figure 3.

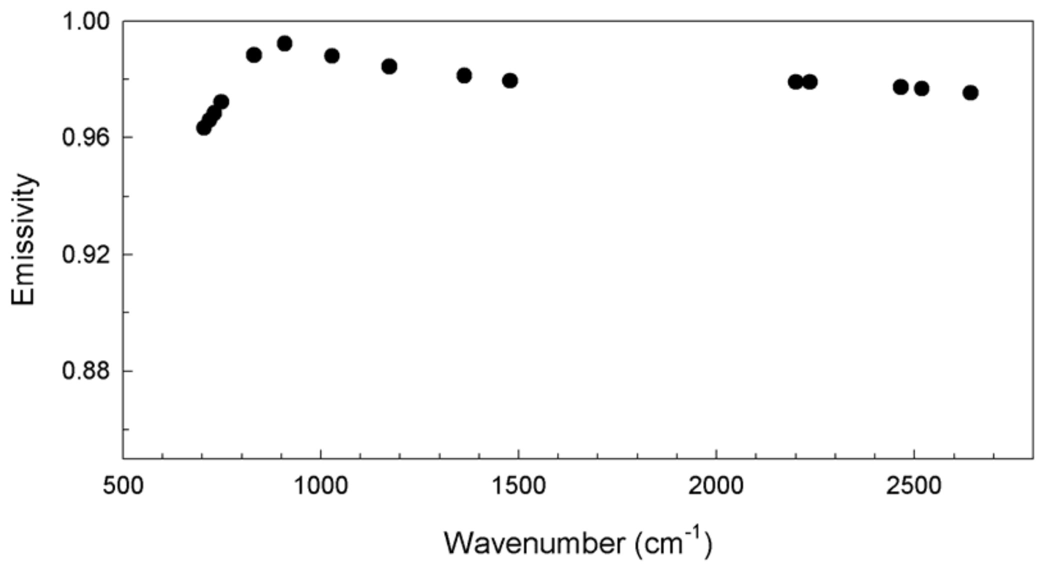

Fig. 3. Spectral variation of emissivity for a calm water surface.

APPENDIX B: ABI SRFS FOR CHANNELS 8-16

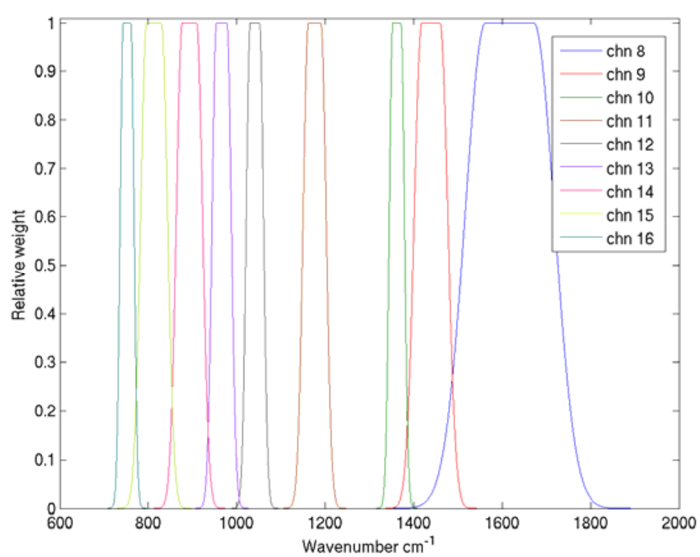

Fig. 4. Spectral response functions defined for the ABI.

APPENDIX C: STANDARD PRESSURE LEVELS FOR FPLOD

Level no.	Pressure (hPa)	Level no.	Pressure (hPa)	Level no.	Pressure (hPa)	Level no.	Pressure (hPa)
1	0.0050	31	32.2744	61	259.9691	91	827.3713
2	0.0161	32	35.6505	62	272.9191	92	852.7880
3	0.0384	33	39.2566	63	286.2617	93	878.6201
4	0.0769	34	43.1001	64	300.0000	94	904.8659
5	0.1370	35	47.1882	65	314.1369	95	931.5236
6	0.2244	36	51.5278	66	328.6753	96	958.5911
7	0.3454	37	56.1260	67	343.6176	97	986.0666
8	0.5064	38	60.9895	68	358.9665	98	1013.9476
9	0.7140	39	66.1253	69	374.7241	99	1042.2319
10	0.9753	40	71.5398	70	390.8926	100	1070.9170
11	1.2972	41	77.2396	71	407.4738	101	1100.0
12	1.6872	42	83.2310	72	424.4698		
13	2.1526	43	89.5204	73	441.8819		
14	2.7009	44	96.1138	74	459.7118		
15	3.3398	45	103.0172	75	477.9607		
16	4.0770	46	110.2366	76	496.6298		
17	4.9204	47	117.7775	77	515.7200		
18	5.8776	48	125.6456	78	535.2322		
19	6.9567	49	133.8462	79	555.1669		
20	8.1655	50	142.3848	80	575.5248		
21	9.5119	51	151.2664	81	596.3062		
22	11.0038	52	160.4959	82	617.5112		
23	12.6492	53	170.0784	83	639.1398		
24	14.4559	54	180.0183	84	661.1920		
25	16.4318	55	190.3203	85	683.6673		
26	18.5847	56	200.9887	86	706.5654		
27	20.9224	57	212.0277	87	729.8857		
28	23.4526	58	223.4415	88	753.6275		
29	26.1829	59	235.2338	89	777.7897		
30	29.1210	60	247.4085	90	802.3714		

APPENDIX D: SCRIPT FILE TO RUN FAST MODEL ON A CLUSTER

The following script was used to run the fast model for the 12 by 14 array of input profile cubes that make up the fulldisk data set. The input and output file locations will necessarily be different on individual systems. To run this code on a particular cluster, change the path names to the appropriate locations, comment out the cvs commands, and manually spread the metadata and executable program to the processing nodes.

```
#!/bin/bash
#
#  Run the new G95 version of the fast model
#
#  Version 2.0
#
#  Version 2.0  adds control for running a model by cvs tag with logging

#  Set appropriate paths for input and output

 CVS_TAG=Release_Jan07_C
 RUNCOLLECT=/home/$LOGNAME/FD8/
 SANDS=sands
 PROFILES=serenity:/tank/pub/PROXY_DATASETS/FULLDISK8km/PROFILES/
 CODE=devel/Model/gfm_f90/

 YSTR=2003;
 DSTR=0624;
 HOUR=2120;

#--------------------------------------------------------------------
#  Check out code from cvs, compile, and spread to cluster nodes.
#--------------------------------------------------------------------
if true; then
#if false; then
 COMPILE_DIR=cvstest
 if [ ! -d "COMPILE_DIR" ]; then
   mkdir ${COMPILE_DIR}
   if [ ! -d "${COMPILE_DIR}" ]; then
     echo 'cvstest directory creation unsuccessful, exiting'
     exit
   fi
 fi
 cd ${COMPILE_DIR}
 cvs checkout -t ${CVS_TAG} ${CODE}
 cvs checkout ${CODE}
 cd ${CODE}
 ./build_ifort
 cp irfm.e ~/fm_static_data/.
 cluster-fork mkdir /scratch/fm_static_data
 cluster-fork rsync -uvL zara:/home/eriko/fm_static_data/*
/scratch/fm_static_data/.
fi

 EOFSTR=EOF;
 H='`';
 D='$';

 c_x=('01' '02' '03' '04' '05' '06' '07' '08' '09' '10' '11' '12')
 c_x_in=('1' '2' '3' '4' '5' '6' '7' '8' '9' '10' '11' '12')
 c_y=('01' '02' '03' '04' '05' '06' '07' '08' '09' '10' '11' '12' '13' '14')
 c_y_in=('1' '2' '3' '4' '5' '6' '7' '8' '9' '10' '11' '12' '13' '14')
```

```
#---------------------------------------------------------------------------
#  Loop through each cube to write run script and submit it to the queue
#---------------------------------------------------------------------------
 let cx=0
 while [ $cx -le 11 ]; do
  let cy=0
   while [ $cy -le 13 ]; do
     cube=${c_x[$cx]}_${c_y[$cy]}
     cube_in=${c_x_in[$cx]}_${c_y_in[$cy]}

#-------------------------------------------------------------
#  Create a single run file that does everything needed
#-------------------------------------------------------------
      runfile=${RUNCOLLECT}FMrun/run_fm_${HOUR}_${cube}.scr
      cat > ${runfile} << EOF
#!/bin/sh

#  Define location of various inputs
 TEMP_SPACE=/scratch/
# FINAL_DEST=${TOA}/${DAYS}${tstep[$tt]}/
 PROF=8km_${DSTR}_${YSTR}.${HOUR}utc_${cube_in}.nc
 CODE=${CODE}

#  Create temporary directory name, check, and make
 RUN_DATE=${H}date '+%Y%m%d'${H}
 RUN_TIME=${H}date '+%H%M%S%Z'${H}

RUN_DIR=${D}{TEMP_SPACE}'fm_run_'`whoami`'_'${D}{RUN_DATE}'_'${D}{RUN_TIME}
 RUN_DIR=${D}{RUN_DIR}_${HOUR}_${cube}
 if [ ! -d "${D}RUN_DIR" ]; then
   mkdir ${D}{RUN_DIR}
   if [ ! -d "${D}RUN_DIR" ]; then
     echo 'run directory creation unsuccessful, exiting'
     exit
   fi
 else
   echo ${D}{RUN_DIR}' already exists, exiting'
   exit
 fi

 cd ${D}{RUN_DIR}

 rsync -uc ${PROFILES}${YSTR}_${DSTR}_${HOUR}/${D}{PROF} .

 cat > input.lis << EOF
/scratch/fm_static_data/giftsfr1.dry
/scratch/fm_static_data/giftsfr2.dry
/scratch/fm_static_data/giftsfr1.ozo
/scratch/fm_static_data/giftsfr2.ozo
/scratch/fm_static_data/giftsfr1.wco
/scratch/fm_static_data/giftsfr2.wco
/scratch/fm_static_data/giftsfr1.wtl
/scratch/fm_static_data/giftsfr2.wtl
/scratch/fm_static_data/giftsfr1.wts
/scratch/fm_static_data/giftsfr2.wts
/scratch/fm_static_data/RefProf1.dat
/scratch/fm_static_data/FIRTM2_baum05.dat
/scratch/fm_static_data/unf_water_c.dat
/scratch/fm_static_data/gifts_chans.dat
${D}{PROF}
/scratch/fm_static_data/global_emis_jun.bin
GEO
-89
NORMAL
```

```
N
output_temp.cdf
${EOFSTR}

 /scratch/fm_static_data/irfm.e < input.lis > frte.out

#------------------
#  Create run log
#------------------
 LFNAME=${HOUR}_${cube}_runlog.txt
 cat > ${D}{LFNAME} << EOF

  Run log for ${D}{RUN_DIR}

  Code tag is: ${CVS_TAG}

 Input profile file:
 ${D}{PROF}

${EOFSTR}
  echo 'Forward Model input file:' >> ${D}{LFNAME}
  cat input.lis >> ${D}{LFNAME}
  echo '' >> ${D}{FINAL_DEST}${D}{LFNAME}

#  Clean up
 rsync -uc ${D}{LFNAME}
${SANDS}::incoming/Fulldisk8km/Jan07/test${YSTR}_${DSTR}_${HOUR}/.
 rsync -uc output_temp.cdf
${SANDS}::incoming/Fulldisk8km/Jan07/test${YSTR}_${DSTR}_${HOUR}/TOA_${YSTR}_$
{DSTR}_${HOUR}_${cube}.cdf
 rm -fr ${D}{RUN_DIR}
EOF

     chmod u+x ${runfile}

     qsub -V -cwd -p -15 -S /bin/sh -e /home/$LOGNAME/FD8/Error -o
/home/$LOGNAME/FD8/Output $runfile
     sleep 15

     let cy=$(($cy + 1))
   done # cy loop end
   let cx=$(($cx + 1))
 done # cx loop end
```