Chapter 1

Introduction

This document descibes the installation, configuration, execution and usage of the Comprehensive Inner Magnetosphere Ionosphere (CIMI) model.

1.1 System Requirements

In order to install and run CIMI the following minimum system requirements apply.

- The CIMI model runs only under the UNIX/Linux operating systems. This now includes Macintosh system 10.x because it is based on BSD UNIX. The CIMI model does not run under any Microsoft Windows operating system.
- A FORTRAN 90 or greater compiler must be installed.
- The Perl interpreter must be installed.
- A version of the Message Passing Interface (MPI) library must be installed.
- Small tests such as single field-line simulations can be done on a single processor
- Very large runs require many more processors.
- In order to generate the documentation you must have LaTex installed on your system. The PDF generation requires the `dvips` and `ps2pdf` utilities.

In addition to the above requirements, the CIMI output is designed to be visualized using IDL. You may be able to visualize the output with other packages, but formats and scripts have been designed for only this visualization software.

Chapter 2

Quick Start

2.1 A Brief Description of the CIMI Distribution

The top level directory contains the following subdirectories:

- `src` - Source code of the main code
- `srcSAMI3` - Source code of SAMI3 model
- `Scripts` - shell and Perl scripts
- `bin` - location of executable
- `doc` - the documentation directory
- `share` - shared scripts and source code
- `util` - general utilities such as TIMING and NOMPI

and the following files

- `README` - a short instruction on installation and usage
- `Makefile` - the main makefile
- `Config.pl` - Perl script for (un)installation and configuration

Note that some of these directories are not distributed with CIMI and need to separately downloaded.

2.2 General Hints

Getting help with scripts and the Makefile

The configuration command and be ellaborated with the `-h` flag. For example,

```
Config.pl -h
```

This will provide a detailed listing of the options and capabilities of the `Config.pl` script. In addition, you can find all the possible targets that can be built by typing

```
make help
```

Input commands: PARAM.XML

A very useful set of files to become familiar with are the `PARAM.XML` files.

This file contains a complete list of all input commands for the component as well as the allowed ranges for each of the input parameters. Although the XML format makes the files a little hard to read, they are extremely useful. A typical usage is to cut and paste commands out of the PARAM.XML file into the PARAM.in file for a run.

2.3 Installing the Code

The CIMI model needs to know what architecture you are running the code on and what FORTRAN compiler will be used. For most platforms and compilers, it can figure this out all by itself. To install CIMI run the command:

```
Config.pl -install
```

in the main directory. This creates `Makefile.def` with the correct absolute path to the base directory and `Makefile.conf` which contains the operating system and compiler specific part of the Makefile. If the compiler is not the default one for a given platform then the compiler must be specified explicitly with the `-compiler` flag. If the MPI header file is not the default one, it can be specified with the `-mpi` flag. For example on a Mac machine one can select the gfortran as follows

```
Config.pl -install -compiler=gfortran
```

To uninstall CIMI type

```
Config.pl -uninstall
```

If the uninstallation fails (this can happen if some makefiles are missing) do reinstallation with

```
Config.pl -install
```

and then try uninstalling the code again. To get a complete description of the `Config.pl` script type

```
Config.pl -h
```

2.4 Creating Documentation

The documentation for CIMI can be generated from the distribution by the command

```
make PDF
```

which creates the user manual

```
doc/CIMI.pdf
```

In order for this to work you must have LaTex installed on your system (and dvips and ps2pdf).

To clean the intermediate files type

```
cd doc/Tex
make clean
```

To remove all the created documentation type

```
cd doc/Tex
make cleanall
```

2.5 Building and Running an Executable

Compilation flags, such as the precision and optimization level are stored in `Makefile.conf`. This file is created on installation of CIMI and has defaults which are appropriate for your system architecture. The precision of reals can be changed to single precision (for example) by typing

```
Config.pl -single
```

while the compiler flags can be modified with

```
Config.pl -debug -O0
```

to debug the code with 0 optimization level, and

```
Config.pl -nodebug -O4
```

to run the code at maximum optimization level and without the debugging flags.

To build the executable **bin/CIMI.exe**, type:

```
make
```

Depending on the configuration, the compiler settings and the machine that you are compiling on, this can take from 1 to up to 5 minutes.

The `CIMI.exe` executable should be run in a sub-directory, since a large number of files are created in each run. To create this directory use thecommand:

```
make rundir
```

This command creates a directory called `run`. You can either leave this directory as named, or `mv` it to a different name. It is best to leave it in the same CIMI directory, since keeping track of the code version associated with each run is quite important. The `run` directory will contain links to the codes which were created in the previous step as well as subdirectories where input and output will reside.

Here we assume that the `run` directory is still called `run`:

```
cd run
```

In order to run CIMI you must have the input files: PARAM.in. The PARAM.in file contains the detailed commands for controlling what you want the code to do during the run. The default PARAM.in file in the run directory suitable to perform a simple test.

To run CIMI interactively on four processors:

```
cd run
mpirun -np 4 CIMI.exe
```

To recompile the executable with different compiler settings you have to use the command

```
make clean
```

before recompiling the executables. It is possible to recompile only a component or just one subdirectory if the `make clean` command is issued in the appropriate directory.

2.6 Restarting a Run

There are several reasons for restarting a run. A run may fail due to a run time error, due to hardware failure, due to software failure (e.g. the machine crashes) or because the queue limits are exceeded. In such a case the run can be continued from the last saved state of CIMI.

The restart files are saved at the frequency determined in the PARAM.in file. Normally the restart files are saved into the IM/restartOUT directory. The files in that directory need only be copied into the PW/restartIN directory inorder to prepare a restarted run.

Chapter 3

The Basics

3.1 Configuration of CIMI

Configuration refers to several different ways of controlling how the CIMI model is compiled and run. The most obvious is the setting of compiler flags specific to the machine and version of FORTRAN compiler. The other methods refer to the grid (default vs expanded) and species.

3.1.1 Setting compiler flags in Makefile.conf and with Config.pl

The compiler flags can be modified by editing

```
Makefile.conf
```

This makefile is created during installation, and it contains the platform and compiler specific part of the makefile system. The most usual changes can be easily done with the Config.pl script. The precision of real numbers can be set with

```
Config.pl -single
```

or

```
Config.pl -double
```

If the precision is modified, the script will execute 'make clean', so that all the files are compiled with the new real precision.

The debugging flags can be switched on and off with

```
Config.pl -debug
```

and

```
Config.pl -nodebug
```

respectively. The maximum optimization level can be set to -O2 with

```
Config.pl -O2
```

The minimum level is 0, the maximum is 4 (Note that not all compilers support level 4 optimization).

3.1.2 Grid and Species Configuration

The CIMI model is set to have a very general core of code,

```
Config.pl -GridExpanded
```

to configure for H

```
Config.pl -EarthH
```

3.2 PARAM.in

The input parameters for CIMI are read from the `PARAM.in` file which must be located in the run directory.

We refer to the lines starting with a # character as commands.

For example if the command string

```
#END
```

is present, it indicates the end of the run and lines following this command are ignored. If the #END command is not present, the end of the file indicates the end of the run.

There are several features of the input parameter file syntax that allow the user to easily run the code in a variety of modes while at the same time being able to keep a library of useful parameter files that can be used again and again.

The syntax and the content of the input parameter files is defined in the PARAM.XML files.

This file can be read (and edited) in a normal editor. The same files are used to produce much of this manual with the aid of the `share/Scripts/XmlToTex.pl` script. The `Scripts/TestParam.pl` script also uses these files to check the PARAM.in file. Copying small segments of the `PARAM.XML` files into `PARAM.in` can speed up the creation or modification of a parameter file.

3.2.1 Commands, Parameters, and Comments

The commands in the PARAM.in file are necessary for instucting the CIMI to carry out the simulation. The following is an example of a parameter.

```
#TIMEACCURATE
F                       DoTimeAccurate
```

In this example, the parameter is #TIMEACCURATE and the command is DoTimeAccurate.

3.2.2 The Order of Commands

In essence, the order of parameter commands within a session is arbitrary, but there are some important restrictions. We should note that the order of the parameters following the command is not arbitrary and must exactly match what the code requires.

3.2.3 Iterations, Time Steps and Time

In several commands the frequency or 'time' of some action has to be defined. This is usually done with a pair of parameters. The first defines the frequency or time in terms of the number of time steps, and the second in terms of the simulation time. A negative value for the frequency means that it should not be taken into account. For example, in time accurate mode,

```
#SAVEPLOT
-1              DtSavePlot
2000            DnSavePlot
T               SaveFirst
```

means that a plot file should be saved after every 2000th time step, while

```
#SAVEPLOT
100             DtSavePlot
-1              DnSavePlot
T               SaveFirst
```

means that it should be saved every 100 seconds in terms of physical time.

Chapter 4

Using the CIMI as the IM component of the SWMF

The ring current can have a significant impact on the solution in the magnetosphere. It controls the energy density in the inner magnetosphere and is important for the overall solution. Because of the importance of including such connections in our model, we now turn to the issue of how to use the CIMI model as a component of the SWMF.

4.1 Configuring the SWMF to use CIMI

Configuring the SWMF to use the CIMI is straight forward. Simply type:

```
Config.pl -v=IM/CIMI
```

Doing so tells the SWMF that the CIMI will represent the IM component of the SWMF. This configuration allows the user to execute the CIMI model through the SWMF without any other components.

Should the user wish to include, say, a global magnetosphere or the ionosphere electrodynamics the configuration is simply:

```
Config.pl -v=IM/CIMI,GM/BATSRUS,IE/Ridley_serial
```

Coupling with these other components is described in greater detail in the next section.

4.2 Coupling the IM component with other components

The CIMI model represents the IM component of the SWMF. It can be directly coupled to the Global Magnetosphere (GM) and Ionosphere Electrodynamics (IE) components by using the

```
#COUPLE1
```

or

```
#COUPLE2
```

commands (See the SWMF manual for details). The IE component is coupled in a unidirectional manner with the IM component, by which the polar cap potential is passed to IM but no information flows in the opposite direction. (actually there is a feedback of currents provided, but it is not recommended).

The IM-GM coupling is more complicated. The GM component can be coupled unidirectionally, with the IM component setting the densities and temperature at the IM outer boundary, or it can be set bidirectionally where the magnetospheric pressure and density is set from the IM model.

There are several options for how the coupling between GM and IM can proceed. They all, however, start with the same command in the PARAM file:

```
#COUPLE2
GM              COMP1
IM              COMP2
10.0            DTcoupling
```

Chapter 5

Complete List of Input Commands

The content of this chapter is generated from the Param/PARAM.XML file. The XML file can be read with an editor and can be used for creating PARAM.in files by copying small parts from them.

The transformation of the XML format into LaTex is done with the share/Scripts/XmlToTex.pl script. This script generates index terms for all commands, which are used to create an alphabetical index at the end of this chapter.

5.1 Input Commands for the CIMI: IM Component

List of IM commands used in the PARAM.in file

5.1.1 General commands

#ECHO command

```
#ECHO
T                          DoEcho
```

If the DoEcho variable is true, the input parameters are echoed back. The echoing either goes to the standard output. The default value for DoEcho is .false., but it is a good idea to set it to true at the beginning of the PARAM.in file.

#INCLUDE command

```
#INCLUDE
IM/restartIN/restart.H          NameCIMIIncludeFile
```

The NameCIMIIncludeFile parameter contains the name of the file to be included. The file name may be followed with a trailing comment if it is separated with at least 3 spaces or one TAB character. The #INCLUDE command can be used anywhere in the parameter file, even in the sections which contain the component specific parameters. For example the information in the run/IM/restartIN/restart.H file or parameters specific to a component can be included.

#END command

```
#END
```

The #END command signals the end of the included file or the end of the PARAM.in file. Lines following the #END command are ignored. It is not required to use the #END command. The end of the included file or PARAM.in file is equivalent with an #END command in the last line.

5.1.2 Timing control

#TIMESIMULATION command

```
#TIMESIMULATION
0.0                     TimeSimulation
```

This command specifies the current simulation time. It is typically read from the IM/restartIN/restart.H file. Default value is 0.

#STOP command

```
#STOP
1 hour                  TimeSimulation
```

This command specifies the simulation stop time.

#STARTTIME command

```
#STARTTIME
2012                    iYear
11                      iMonth
12                      iDay
18                      iHour
00                      iMinute
00                      iSecond
```

The #STARTTIME command sets the initial date and time for the simulation in Greenwich Mean Time (GMT) or Universal Time (UT) in stand alone mode. This time is stored in the CIMI restart header file. CIMI has no default values.

#IMTIMESTEP command

```
#IMTIMESTEP
1.                      IMDeltaT [s]
1.                      IMDeltaTMax [s]
```

The IMTIMESTEP command controls for the time stepping of CIMI, but does not control the sub-cycling based on the courant condition. Useful for setting large time-steps for basic tests. Defaults shown.

5.1.3 Restart control

#RESTART command

```
#RESTART
T                       IsRestart
F                       DoReadRestartSatellite (read if IsRestart=T)
```

If IsRestart is true, read in restart files from a previous run. If DoReadRestartSatellite is also true, read the satellite buffer file found in IM/restartIN/restart.sat for tracing satellites. This command is usually read from the IM/restartIN/restart.H file. The default is to start the simulation from scratch, so IsRestart is false.

#SAVERESTART command

```
#SAVERESTART
100.0                   DtSaveRestart
```

Command controls how often restart files are saved. This only works when CIMI is in standalone mode.

#PRERUNFIELD command

```
#PRERUNFIELD
F                       DoWritePrerun
F                       UsePrerun
60.                     DtRead
```

Command controls the saving or reading of the Prerun magnetic field (IM/PrerunField_********.dat) and ionospheric potential (IM/PrerunIE_********.dat) files, where ******** is the simulation time in seconds from the simulation start time. While coupled with GM, the user can save the calculated field line traces to IM/. As the field line traces can be the most computationally expensive part routine in CIMI, this allows for quick reconfiguration of a GM-coupled run using CIMI in standalone while still using the BATSRUS fields. Default values shown.

****NOTE****

Files are only saved when DoWritePrerun=.true. in the GM-coupled run; UsePrerun and DtReadSat are NOT read while DoWritePrerun=.true. When performing a restart, set UsePrerun=.true. with the time cadence given by DtRead. Consult input/testfiles/PARAM.in.test.Prerun usage of the PrerunSat files in conducting the Prerun test.

#PRERUNSAT command

```
#PRERUNSAT
F                       DoWritePrerunSat
F                       UsePrerunSat
60.                     DtReadSat
```

Command controls the saving or reading of the Prerun satellite trace files ((IM/PrerunSat_********.dat)), where ******** is the simulation time in seconds from the simulation start time. While coupled with GM, the user can save the calculated satellite traces to IM/. As the satellite traces can be quite computationally expensive as the number of satellites increases, this allows for quick reconfiguration of a GM-coupled run using CIMI in standalone while still using the BATSRUS fields. Default values shown.

****NOTE****

Files are only saved when DoWritePrerunSat=.true. in the GM-coupled run; UsePrerunSat and DtReadSat are NOT read while DoWritePrerunSat=.true. When performing a restart, set UsePrerunSat=.true. with the time cadence given by DtReadSat. Consult input/testfiles/PARAM.in.test.Prerun for usage of the PrerunSat files in the Prerun test.

5.1.4 Numerical scheme

#LIMITER command

```
#LIMITER
```

```
F                       UseMcLimiter
2                       BetaLimiter
```

Set whether or not the MC limiter is used. If it is not, the super bee limiter is used. Also set the Beta parameter for the MC limiter. The default value is shown.

#DRIFTSCHEME command

```
#DRIFTSCHEME
2               iOrderLat
2               iOrderLon
```

Specifies the spatial advection scheme. If iOrderLat = 7 and iOrderLon = 7, then construct latitude/longitude inter flux in 7th order scheme, using ULTIMATE advection scheme (Lagrangian interpolation + Universal Limiter).

If iOrderLat and iOrderLon are both equal to 2, then use the default 2nd order scheme and Superbee limiter.

Default values shown.

NOTES:

1. Users wanting to use the 7th order scheme in latitude should compile CIMI with the GridUniformL option.
2. iOrderLat and iOrderLon can take integer values of [1-7], but have only been tested and developed with iOrderLat=iOrderLon=2 and iOrderLat=iOrderLon=7.

#HIGHERORDERDRIFT command

```
#HIGHERORDERDRIFT
F                       UseHigherOrder
2                       iOrderLat
2                       iOrderLon
```

This command has the same functionality as the updated #DRIFTSCHEME, but is included for backwards compatibility. Default values shown.

#STRICTDRIFT command

```
#STRICTDRIFT
F                       IsStrictDrift
```

This command specifies whether to force phase space density to be ≥ 0. If set to TRUE and phase space density goes negative for any species, the code will immediately stop. Default shown.

5.1.5 Initial and boundary conditions

#INITIALF2 command

```
#INITIALF2
F                       IsEmptyInitial
F                       IsGmInitial
T                       IsDataInitial
F                       IsRBSPData (read if IsDataInitial=T)
```

Determines whether to fill the fluxes in the simulation domain based on a Maxwellian determined from MHD quantities (IsGmInitial=T) or to set the initial fluxes in the simulation domain to zero (IsEmptyInitial=T, not recommended) or to set the initial fluxes in the simulation to values from AMPTE/CCE data (IsDataInitial=T). One can also put RBSP observed fluxes into the same format as the AMPTE/CCE data to initialize CIMI with RBSP observations which is controlled with the IsRBSPData logical, which is only read if IsDataInitial=T.

The default IsGmInitial=T and all others are false.

#INITIALLSTAR command

```
#INITIALLSTAR
F                       DoLstarInitialization
```

Determines whether to initialize the PSD array from dipolar L values or CIMI calculated Lstar. Default is shown.

#PLASMASHEET command

```
#PLASMASHEET
F                       UseYoungEtAl
T                       UseBoundaryEbihara
```

Command determines if the empirical boundary conditions for ions to be applied at the plasmasheet boundary. Variable UseBoundaryEbihara is only read if CIMI is compiled in stand-alone mode. If UseBoundaryEbihara is .true., uses Ebihara & Ejiri, 2000 and Borovsky et al., 1998 models for boundary density and temperatures, respectively. When false, uses the Tsyganenko-Mukai plasmasphere model. Default shown.

#BMODEL command

```
#BMODEL
mhd                     NameModel
T                       UseCorotation
T                       UsePotential
```

Specify the magnetic field model that CIMI will use. Acceptable values for NameModel are:

- 'dip' - Static Dipolar magnetic field. (Stand-alone)
- 't96' - Tsyganenko 1996 magnetic field model (Stand-alone)
- 't04' - Tsyganenko-Sitnov 2004 storm-time magnetic field model. (Stand-alone)
- 'mhd' - MHD calculated magnetic fields. (Coupled)

Variables UseCorotation and UsePotential are only read when the dipole field is requested. Defaults shown.

#IEMODEL command

```
#IEMODEL
F                       UseWeimer
```

Command sets whether the Weimer ionospheric potential is used. Default shown.

#NGDC_INDICES command

```
#NGDC_INDICES
Indices.dat             NameNGDCFile
```

Command points to the file containing the DST quick look Index and the F10.7 to be used during the run. Command is used only in stand-alone runs. Example file can be seen in input/testfiles/Indices.dat Default shown.

#MHD_INDICES command

```
#MHD_INDICES
imf.dat                 UpstreamFile
```

Command specifies the file containing the upstream solar wind parameters to be used during the run. Formatting is identical to the solar wind input files used in BATSRUS/SWMF, but an example can be found in input/testfiles/imf.dat. Command is used only in stand-alone runs. Default shown.

#KYOTO_DST command

```
#KYOTO_DST
T               UseDstKyoto
Dst_Kyoto.dat   NameDstFile
```

Command points to the file containing the DST Index from the Kyoto website in the IAGA2002 format.

#KYOTO_AE command

```
#KYOTO_AE
T               UseAeKyoto
Ae_Kyoto.dat   NameAeFile
```

Command points to the file containing the AE Index from the Kyoto website (https://wdc.kugi.kyoto-u.ac.jp/aeasy/index.html) in the IAGA2002 format.

#POTSDAM_KP_AP_F107 command

```
#POTSDAM_KP_AP_F107
T               UseKpApF107IndicesFile
```

Command tells CIMI to use the KP,AP,F107 values from https://www.gfz-potsdam.de/en/section/geomagnetism/data products-services/geomagnetic-kp-index. Note that this file contines data from 1937 to the present, but must be periodically updated with the latest values.

#SOLARWIND command

```
#SOLARWIND
5.0                     DensitySW
400.0                   VelSW
0.0                     BxSW
0.0                     BySW
-5.0                    BzSW
```

Command sets the constant solar wind values to be used through out the simulation. Command is used only in stand-alone runs. VelSW specifies the x-component of the solar wind velocity.

#SMOOTH command

```
#SMOOTH
F                       UseSmooth
60.                     SmoothWindow
```

Command sets the box car averaging smooth window [in seconds] of the input solar wind file. Default is UseSmooth=.false., but SmoothWindow variable is only read if UseSmooth=.true.

5.1.6 Output parameters

#SAVEPLOT command

```
#SAVEPLOT
8                       nCIMIFileType
fls ions                StringPlot
300.                    DtOutput
F                       DoSaveSeparateFiles
fls e                   StringPlot
60.                     DtOutput
T                       DoSaveSeparateFiles
psd all                 StringPlot
60.                     DtOutput
F                       DoSaveSeparateFiles
vl H                    StringPlot
60.                     DtOutput
T                       DoSaveSeparateFiles
vpdrift ions            StringPlot
60.                     DtOutput
F                       DoSaveSeparateFiles
preci all               StringPlot
60.                     DtOutput
F                       DoSaveSeparateFiles
2d both                 StringPlot
60.                     DtOutput
2d lstar                StringPlot
60.                     DtOutput
F                       DoSaveSeparateFiles
```

The #SAVEPLOT command determines the number, type, and frequency of output from CIMI.

The nCIMIPlotType sets the number of plot types to be read in and configured. For each plot type, the StringPlot parameters define the content of each file and species or domain to be plot.

StringPlot must contain 2 parts:

```
PlotType PlotOption
```

PlotType can take values:

```
'fls'           - PARTICLE flux information. Also accepts 'flux'
'psd'           - PARTICLE Phase Space Density (PSD).
'vl'            - PARTICLE radial drift. Also accepts 'vldrift'
'vp'            - PARTICLE poloidal drift. Also accepts 'vpdrift'
'preci'         - PARTICLE precipitation to the ionosphere.
                  Also accepts 'precipitation' or 'precip'
'2d'            - VARIABLE Data on the 2d simulation plane
```

PlotOption controls species information where relevant or controls for domain of output (in the case of ‘2d’). PlotOption takes on values for the aforementioned PARTICLE output types:

```
‘all’                   - Output all particle species.
‘ions’                  - Output only the ion species’ information.
‘e’ or ‘electrons’      - Output only the electron’s information.
‘h’                     - Output only the hydrogen’s information.
‘o’                     - Output only the oxygen’s information.
                          (Only available if CIMI is compiled with ‘-EarthHO’ option.)
‘he’                    - Output only the helium’s information.
                          (Only available if CIMI is compiled with ‘-EarthHOHe’ option.)
```

For 2d output types, PlotOption specifies which VARIABLE output is to be saved. PlotOption can take on values:

```
‘lstar’ or ‘l*’         - Output of VARIABLE L*, the adiabatic drift shell.
‘equator’ or ‘eq’       - Output calculated VARIABLES on the minimum B surface.
‘ionosphere’ or ‘iono’  - Output calculated VARIABLES in the ionosphere.
‘both’ or ‘all’         - Output calculated VARIABLES at both the ionospher and minimum B surface.
```

Plots are saved in IM/plots. PARTICLE flux, PSD, raidal drift, poloidal drift, precipitation files are saved with extensions ‘.fls’, ‘.psd’, ‘.vl’, ‘.vp’, ‘.preci’, respectively. Lstar VARIABLE output is saved with extension ‘.lstar’. IDL scripts for reading these files can be found in CIMI/tools.

VARIABLE output at the equator or ionosphere have extension ‘.outs’ and can be read with SWMF IDL visualization scripts.

DtOutput is required to be read in for ALL output file types. The minimum output is currently set to 60 seconds simulation time.

DoSaveSeparateFiles is a logical check to save individual output files for each time step; otherwise a single appended file is saved. DoSaveSeparateFiles is required to be read for all PARTICLE output types and for Lstar VARIABLE output. Files contain the same information as their appended counterparts and can read with the IDL scripts found in CIMI/tools. Individual PARTICLE files are saved with name format YYYYMMDD_HHMMSS_PARTICLE.EXTENSION where particle currently can be ‘{h,o,he,e}’ for hydrogen, oxygen, helium and electrons, respectively; values for EXTENSION are detailed above. Separate files for the Lstar VARIABLE are output as YYYYMMDD_HHMMSS.lstar.

Default is nCIMIPlotType=0 so no plot files are saved.

#VERBOSELSTAR command

```
#VERBOSELSTAR
F                       DoVerboseLstar
```

Command controls for the output of the Lstar calculation to the screen, including information about magnetic island locations (those locations where B is not monotonically decreasing) and the maximum Lstar values for each value of the second adiabatic invariant, K.

Default is DoVerboseLstar=F.

#VERBOSELATGRID command

```
#VERBOSELATGRID
F                       DoVerboseLatGrid
```

Prints to screen latitude and equatorial grid information.

Default is DoVerboseLatGrid=F.

#SAVELOG command

```
#SAVELOG
10                      DtSaveLog
```

When this command is set, a log file for CIMI is written out. The log file saves the change in ring current energy content for each species resulting from each operator. A new entry in the log is written out every DtSaveLog seconds of simulation time.

#TYPEBOUNDARY command

```
#TYPEBOUNDARY
ellipse                 TypeBoundary
```

Determines if the IM outer boundary is an 'ellipse' or 'circle.' Default value is shown.

#SETBOUNDARYPARAMS command

```
#SETBOUNDARYPARAMS
2.0                     DeltaRmax   [Re]
2.0                     DeltaMLTmax [hour]
```

The CIMI grid is based in the ionosphere and we trace the field from those footpoints through the magnetosphere. CIMI then does a number of checks to set its domain. First, we check for open-closed boundary, and the CIMI domain must be inside that. We then check for multiple off equator magnetic field minima. This usually occurs on the dayside under northward IMF and strong pressure. This gives the "Shebansky orbits", which cannot currently be captured by CIMI so those field lines are treated as open as well.

If the spacing between two successive minB points is more than DeltaRmax, the line is considered open to avoid excessive deformation. Range is 1 to 3 Re.

If the MLT of the minB changes more than DeltaMLTmax from the footpoint, the line is considered open to avoid excessive warping. Range is 1 to 4 hours.

Default values are shown.

#MINIMUMPRESSURETOGM command

```
#MINIMUMPRESSURETOGM
1e-2                    MinimumPressureToGM
```

Sets minimum pressure passed to GM.

#DTSATOUT command

```
#DTSATOUT
60.0                    DtSatOut
```

Sets the time cadence, in seconds, that the particle fluxes are output. Default shown. do I print this?

5.1.7 Gridding parameters

#TYPEBOUNDARY command

```
#TYPEBOUNDARY
Ellipse                 TypeBoundary
```

Determines if the IM outer boundary is an 'Ellipse' or 'Circle.' Default shown.

#ENERGYGRID command

```
#ENERGYGRID
  0.10000               MinIonEnergy (in keV)
316.22777               MaxIonEnergy (in keV)
```

DEPRECATED - Command sets the minimum and maximum of the ion energy grid for the output fluxes. MinIonEnergy and MaxIonEnergy are the bounds for the energy array in keV; electron energies are multiplied by 10. Results in a 15 element grid per species that is evenly spaced logarithmically. Parameter included for backwards compatibility. For more direct control over the grid, use #SETENERGYGRID command.

#SETENERGYGRID command

```
#SETENERGYGRID
15                      neng
T                       UseLogEGrid
  0.10000               MinIonEnergy (in keV)
316.22777               MaxIonEnergy (in keV)
```

Command provides user with direct control over the size, spacing, and energy extent of the ion energy grid for the output fluxes. neng is the number of elements in the energy grid per species. UseLogEGrid=T sets logarithmic spacing to the grid; setting UseLogEGrid=F changes it to linearly spaced. MinIonEnergy and MaxIonEnergy are the bounds for the energy array in keV; electron energies are multiplied by 10. The values displayed here result in CIMI's default energy grid regardless of #SETENERGYGRID being specified in PARAM.in.

#RBSPENERGYGRID command

```
#RBSPENERGYGRID
F                       UseRBSPGrid
```

Command sets the output energy grid to exactly the energy centroids of the Van Allen Probes' MagEIS and REPT instruments. If this parameter is set, the #ENERGYGRID and #SETENERGYGRID commands are ignored. Default shown.

#LATITUDINALGRID command

```
#LATITUDINALGRID
T                       DoDefineVarNpower
2.                      varNpower
72.4356255492731975             xlatmax
```

If DoDefineVarNpower is true, calculate ionospheric latitude grid by xlat = acos(1./varL)**(1./varNpower) varL is uniformly spacing from varLmin to varLmax. xlatmin = acos(1./varLmin)**(1./varNpower) xlatmax = acos(1./varLmax)**(1./varNpower) Note. If LATITUDINALGRID is not turned on, default varNpower is 2. parameter xlatmax is in a unit of degrees.

5.1.8 Physics parameters

#STRONGDIFFUSION command

```
#STRONGDIFFUSION
F                       UseStrongDiff
```

Applies very effective exponential decay of ring current and radiation belt electron populations. Diffusion time is first (mu) and second (K) adiabatic invariant dependent. Can result in electrons being lost by up to 50%.

Default is UseStrongDiff is false.

#DIAGONALIZEDDIFFUSION command

```
#DIAGONALIZEDDIFFUSION
T                       UseDiagDiffusion
```

If UseDiagDiffusion is true, use (Q1,Q2) coordinates instead of (a0,E) or (M,K) during diffusion calculation, where Q1 = K and Q2 is defined to be Q2_min = E and Q2 is obtaind from constant Q2 curve (dQ2 = 0) in (a0,E), integrating dE/da0 = DaE/Daa Note. UseWaveDiffusion must be true to use this option.

#DECAY command

```
#DECAY
T                       UseDecay
10 hours                DecayTimescale in seconds
```

If UseDecay is true, adds exponential decay to ring current ion populations, so that if there are no other effects, ion phase space density (PSD) decays proportional to

```
exp(-(deltaT/DecayTimescale))
```

This ad hoc decay can improve agreement with observed recovery after magnetic storms. The default DecayTimescale value of 10 hours as above seems to be close to optimal. The decay term is NOT applied to the electron PSD since both ring current and radiation belt electrons are represented. Rapid loss of electron PSD is controlled with the #STRONGDIFFUSION routine.

The default is UseDecay false.

#FLC command

```
#FLC
T                       UseFLC
```

If UseFLC is true, adds exponential decay to ring current populations, due to field line curvature scattering. This is based on the formulation in Young et al 2002 and 2008.

The default is UseFLC false.

5.1.9 Testing parameters

#DIAGDIFFUSIONTEST command

```
#DIAGDIFFUSIONTEST
F                       UsePitchAngleDiffusionTest
F                       UseEnergyDiffusionTest
```

If either of the parameters is true, calculate diffusion and compare with analytical solution. NOTE: UseWaveDiffusion and UseDiagDiffusion must be true to use this option. Only UsePitchAngleDiffusionTest OR UseEnergyDiffusionTest can be TRUE at a time; the other must be false.

#COMPOSITION command

```
#COMPOSITION
FIXED                   NameCompModel
0.85                    DensityFraction Hp
0.15                    DensityFraction Op
1.0                     DensityFraction e
```

When CIMI is coupled with a single fluid MHD code, the boundary conditions obtained from GM do not determine the composition of the plasma. This command sets the assumed fraction of H+ and O+. The combined global-inner magnetospheric dynamics strongly depends on these values. NameCompModel selects the model used to set the composition. Currently only "FIXED" is implemented. The fraction of O+, H+ and electrons are given by the next three parameters. The first two should add up to 1, and the electron number density should be 1, so in fact only the first parameter is adjustable, the rest is provided for testing purposes only.

Default values are shown.

#WAVES command

```
#WAVES
T                           UseWaveDiffusion
T                           UseHiss
T                           UseChorus
D_hiss_UCLA.dat             HissWavesD
D_LBchorus_merge3.dat       ChorusWavesD
T                           UseKpIndex
```

This command defines the wave diffusion in CIMI. By default the waves are off so this command must be added to turn on waves. Other commands include whether to use use plasmaspheric hiss and lower band Chorus. The data files for the wave diffusion coeficients that should be in the rundir/IM/ directory. Note that waves can be driven by KP or AE. When KP is used and GM coupling is true we take the KP passed from GM (note that GEOMAGINDICES should be turned on in BATSRUS for this to work. Note, if not using Kp you should make sure to read an AE file using #KYOTO_AE.

Index