ICES User Manual

Code Version 2.0

Kenneth Hansen, Martin Rubin, Nicolas Fougere

November 11, 2011

Contents

Chapter 1

Introduction

This document describes the Inner Coma Environment Simulation tools (ICES). The tool development began with the efforts of our ISSI comet modeling team and have continued as a part of our work with the ESA/Rosetta mission. Out original intent was provide the comet community with access to results from a comprehensive set of models of the cometary coma ranging from the surface out to the interaction with the solar wind and covering dust, gas and plasma phases. We have now expanded the tool to include nucleus models.

Our initial target comet is 67P/Churyumov-Gerasimenko with specific application to the Rosetta mission. Our intent is that the tool will be useful for mission and science planning as well as for data interpretation.

The ICES tool does not run the individual models, but rather extracts data from data files provided with the tool. Currently the following models are available:

Coma Models

Abbrev.	Description	Physics
ICES_c	Inner Coma Environment Simulator	User interface & tool
DSMC	Direct Simulation Monte Carlo	Dust and neutral gas
ELEC	Electron	Field aligned electrons
HYB	Hybrid	Plasma: kinetic ions, fluid electrons
MHD	Magnetohydrodynamics	Plasma: single fluid

Nucleus Models

Abbrev.	Description	Physics
ICES_n	Inner Coma Environment Simulator	User interface & tool
K3D	Nucleus Evolution Model	Coma Sources, Surface and Subsurface

1.1 Acknowledgments

The ICES tool is the result of the work of several different research groups who have developed state of the art numerical models of different parts of the cometary environment. On behalf of the ICES effort, we have coordinated the use of these models to address the cometary environment in a self-consistent and coupled manner and have produced a tool which provides the community easy access to model results.

The physics models were developed by the following research groups:

Abbrev.	Institution	Location	Responsible Persons
ICES	University of Michigan	Ann Arbor, MI	K.C. Hansen
DSMC	University of Michigan	Ann Arbor, MI	Michael Combi, Valeriy Tenishev
ELEC	University of Kansas	Lawrence, Kansas	Tom Cravens, Ina Robertson
HYB	Institute for Theoretical Physics, Technical University of Braunschweig	Braunschweig, Germany	Uwe Motschmann, Nikolaos Gortsas
MHD	University of Michigan	Ann Arbor, MI	K.C. Hansen, Martin Rubin, Tamas Gombosi

The majority of work done to develop the ICES tool was supported by the following funding agencies, institutions and individuals.

- **ISSI comet modeling team**
  The ISSI comet modeling team was organized and is directed by Tamas Gombosi (University of Michigan). Team meetings have been held at the International Space Science Institute (ISSI) in Bern, Switzerland with funding provided by ISSI.

- **ICES**
  The ICES tool, the user interface and documentation, and the coordination of tool issues was carried out by the Center for Space Environment Modeling (CSEM) of the University of Michigan under the funding from JPL as part of US Rosetta project. The main development work has been carried out by KC Hansen and Martin Rubin.

- **DSMC**

- **Electrons**

- **Hybrid** The comet Hybrid model was developed at the Institute of Theoretical Physics of the Technical University of Braunschweig under project MO539/10 of the Deutsche Forschungsgemeinschaft (DFG). Enhancements of the Hybrid model are supported by the Institute of Planetary Research of the German Aerospace Center (DLR).

- **MHD** The comet MHD model is part of the Space Weather Modeling Framework (SWMF) which was developed at the Center for Space Environment Modeling (CSEM) of the University of Michigan under the NASA Earth Science Technology Office (ESTO) Computational Technologies (CT) Project (NASA CAN NCC5-614). The main work has been carried out by KC Hansen and Martin Rubin.

- **K3D**

1.2 Contact

The ICES tool can be download from

http://ices.engin.umich.edu/ICES/Download.php/

Details of the current distribution, documentation and other resources can be found at this location.

For questions regarding the ICES tool or any of the physics modules please contact ices-admin@umich.edu.

Additional information about the ISSI comet modeling team, the individual teams, and the Rosetta mission can be found under the following links:

http://www.issibern.ch (International Space Science Institute (ISSI), Bern, Switzerland)
http://csem.engin.umich.edu/ISSI_Comet/(ISSI comet home)
http://www2.ku.edu/ kuspace/ (Electron model, University of Kansas)
http://www.tu-braunschweig.de/theophys/research/plasma (Hybrid model, University of Braunschweig)
http://csem.engin.umich.edu (MHD model, University of Michigan)
http://www.esa.int/export/SPECIALS/Rosetta/index.html (ESA Rosetta page)
http://rosetta.jpl.nasa.gov/ (JPL Rosetta page)

1.3 System Requirements

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

- ICES runs only under UNIX/Linux operating systems. This now includes Macintosh system 10.x because it is based on BSD UNIX. ICES does not run under any Microsoft Windows operating system.
- A FORTRAN compiler must be installed to extract data from the MHD and Electron models.
- A C++ compiler is needed to extract data from the DSMC and the Hybrid simulations.
- Python must be instaled to extract data from the K3D model.
- The Perl Interpreter must be installed.
- A minimum of 1GB of RAM is needed to run the ICES tool.
- Roughly 50 MB of hard disk space is needed to install the ICES tool - not including the simulation data files.
- Each simulation case requires an additional 1 GB of hard disk space.
- In order to generate the documentation you must have LaTeX installed. The PDF generation requires the dvips and ps2pdf utilities. To generate the HTML version you also must install the latex2html package.

Please note that C++, Perl and Python general come automatically installed on many UXIX/Linux systems.

1.4 Documentation

The ICES tool is delivered with both a PDF version and an HTML version of the documentation. The PDF version is located at Doc/ICES_manual.pdf and can be open with Adobe Reader or a similar product. The HTML version can be accessed by opening the file Doc/HTML/index.html in your favorite web browser.

You can also access the user manual online by pointing your browser to:
http://ices.engin.umich.edu/ICES/

It is also possible to access the source LaTex files for the documentation and to build new version of the PDf and HTML versions. The LaTex source is found in Doc/Tex/ and the new version can be built by typing.

```
make PDF
make HTML
```

in the main directory of the distribution. Output files from these two commands are located in the Doc/Tex/ directory.

Chapter 2

Installing and Running ICES

The ICES tools are delivered as a tar-gzip file of the source code. This code must be compiled before it can be used. The tool also makes use of data files which are installed separately from the tool because the data files are very large in size.

Basics for using the tool are found in this chapter. The next chapter addresses problems that may arise or more complex ways of interacting with the tool.

The top level directory of the ICES tool contains the following subdirectories and files:

```
Data/           Directory containing the data files needed by each model
Doc/            Documentation
ICES_c.pl       Main script, runs data extraction for COMA models
ICES_n.pl       Main script, runs data extraction for NUCLEUS models
Makefile        Utility for compiling source codes
Makefile.conf   Configuration file for Fortran and C compilers
Models          Directory containing source code for all models
README          Short description of how to install and run the code
Scripts/        Useful scripts
Tests/          Contains sample files and files for comparison tests
```

2.1 Installing the code

The first step in installing the code is untarring the distribution. Change directories to the location at which you would like the ICES tool to reside.

```
cd <path_to_desired_installation_location>
```

Next, untar the code

```
tar xzf <path_to_file>/ICES_v<version_number>.tgz
```

This will unpack in a new directory ICES/. If the tar program does not know about the -z flag, you will have to open the distribution in two steps

```
gunzip <path_to_file>/ICES_v<version_number>.tgz
tar xf <path_to_file>/ICES_v<version_number>.tar
```

The main directory ICES and its content as described above will be generated. Installation of data files is described in the next section.

You should now be able to find additional information about the ICES utilities by typing

```
make help
ICES_c.pl -h
ICES_n.pl -h
```

2.2 Installing coma data files

Data files are distributed separately from the code because they are quite large in size. You can find a list of all available cases for which data file exist at

http://ices.engin.umich.edu/ICES/Download.php

as well as in Chapter 6. Note that coma and nucleus data directory structures are quite different due to the differences in the way the models function.

In order to install the data files, begin by downloading a gzipped tar file of a case you are interested in from the above URL. For the sake of discussion in this manual we will assume that you downloaded the 67P/CG case which corresponds to the test case described below in Section: 2.6

```
CG_1.3_au_00.tgz
```

Once you have a file, move to the Data/Coma/ directory

```
cd Data/Coma
```

and unpack the gzipped tar file

```
tar xzf <path_to_file>/CG_1.3_au_00.tgz
```

This will make a new directory in the Data/Coma/ directory that will contain all the necessary data files to extract information from each of the physics models.

If your tar program does not know about the -z flag, you will have to install the data files in two steps

```
gunzip <path_to_file>/CG_1.3_au_00.tgz
tar xf <path_to_file>/CG_1.3_au_00.tar
```

Other data files can be download from the web site and installed in a similar manner.

The data is ordered by case first and then by the model. After installation of the case CG_1.3_au_00 the Data/Coma/ folder should look like the following.

```
Data/Coma/
    CG_1.3_au_00/
        DETAILS
        DSMC/
        Electron/
        HYB/
        MHD/
        SHORT_DESCRIPTION
        VERSION
```

Every sub-folder in the case directory contains the data of the corresponding model. Each case contains a DETAILS file which provides some details about the simulation and physical parameters for that case. In addition, each case contains a VERSION file which give a version number so that the user can make sure they have the most up to date data files. Moreover, a SHORT_DESCRIPTION file provides a brief description of each case. For more information about the individual models (e.g. the size of the simulated domain), see Chapter 7.

2.3 Installing nucleus data files

Data files are distributed separately from the code because they are quite large in size. You can find a list of all available cases for which data file exist at

http://ices.engin.umich.edu/ICES/Download.php

as well as in Chapter 6. Note that coma and nucleus data directory structures are quite different due to the differences in the way the models function.

In order to install the data files, begin by downloading a gzipped tar file of a case you are interested in from the above URL. For the sake of discussion in this manual we will assume that you downloaded the 67P/CG case which corresponds to the test case described below in Section: 2.6

```
K3D_CG_01.tgz
```

Once you have a file, move to the Data/Nucleus/ directory

```
cd Data/Nucleus
```

and unpack the gzipped tar file

```
tar xzf <path_to_file>/K3D_CG_01.tgz
```

This will make a new K3D directory in the Data/Nucleus if it does not exist year and will add to it the CG_01 directory that will contain all the necessary data files to extract information for this case.

If your tar program does not know about the -z flag, you will have to install the data files in two steps

```
gunzip <path_to_file>/K3D_CG_01.tgz
tar xf <path_to_file>/K3D_CG_01.tar
```

Other data files can be download from the web site and installed in a similar manner.

The data is ordered by case first and then by the orbit (inbound or outbound) and finally by the heliocentric distance. After installation of the case K3D_CG_01 the Data/Nucleus/K3D folder should look like the following.

```
Data/Nucleus/K3D/
    CG_01/
        CG_01.cgf
        CG_01_01i_5.7au.dat
        CG_01_02i_1.3au.dat
        CG_01_01i_1.4au.dat
        CG_01_01i_1.5au.dat
        ...
        DETAILS
        SHORT_DESCRIPTION
        VERSION
```

Each case contains a DETAILS file which provides some details about the simulation and physical parameters for that case. In addition, each case contains a VERSION file which give a version number so that the user can make sure they have the most up to date data files. Moreover, a SHORT_DESCRIPTION file provides a brief description of each case. For more information about the individual models (e.g. the size of the simulated domain), see Chapter 7.

2.4 Changing the default settings

As outlined in the system requirements (Section 1.3), the ICES code can be run only under UNIX/Linux (this includes Mac, version 10.x). Because the tools is provided as source code, it must be compiled before it can be used. Before compiling the source code you must make sure that you have a Fortran 77 and 90/95 compiler as well as a C++ compiler installed on your system.

The codes in the ICES distribution have been tested on the following systems with the listed compilers:

Abbrev.	Source code type	Operating System	Compiler Name (Provider)
DSMC	C++	Mac Os X	g++ (GNU)
		SGI Altix	g++ (GNU)
ELEC	Fortran 77	Mac OS X	gfortran (GNU), f95 (NAG)
		SGI Altix	ifort (Intel)
HYB	C++	Mac OS X	g++ (GNU)
		SGI Altix	g++ (GNU)
MHD	Fortran 77	Mac OS X	gfortran (GNU), f95 (NAG)
		SGI Altix	ifort (Intel)
K3D	Python	Mac OS X	python

If your compiler does not appear, it may still work. To find out how, read this section as well as the sections in the next chapter on installing compilers (Sections 4.1 and 4.2).

In order to compile the code you must edit the Makefile.conf file in order to set the default compilers for your system. You must make a selection for the Fortran 90/95 (F90), Fortran 77 (F77) and the C++ compilers. No selection is necessary for Python. A sample entry from the Makefile.conf file is below:

```
#-------------------------------------------------
# Select a compiler for FORTRAN 90/95 codes
#-------------------------------------------------
#GNU Compiler
#    Known to work on:  MAC OSX
COMPILE.f90     = gfortran

#NAG Compiler
#    Known to work on: MAC OSX
#COMPILE.f90     = f95

#INTEL Compiler
#    Known to work on:  SGI Altix
#COMPILE.f90     = ifort
```

In this example the GNU gfortran compiler is selected for compiling F90 files. The selected compiler should not have a # in front of it. All other compilers should be commented out using the # symbol.

The F77 compiler should typically be set the same as the F90 compiler.

```
#-------------------------------------------------
# Select a compiler for FORTRAN 77 codes
#    (In most cases should be set the same as for
#     the FORTRAN 90/95 codes)
#-------------------------------------------------
#Set the same as for FORTRAN 90/95
```

```
COMPILE.f77      = ${COMPILE.f90}

#Set different than for FORTRAN 90/95
#
#GNU Compiler
#COMPILE.f77     = gfortran
```

In this example the Makefile.conf has the F77 compiler set equal to the F90 compiler. If the "COMPILE.f77 = gfortran" then the F77 compiler could be set to something different than the F90 compiler.

The C++ compiler is set similarly.

You can check to see if the tool can located the compilers that you have selected by typing

```
make status
```

This command lists the current state of compilation of the different physics models as well as the status of finding the different compilers. A sample output from this command might be:

```
Running ICES_v1.0b
OS: Darwin Kernel Version 8.11.1: Wed Oct 10 18:23:28 PDT 2007; root:xnu-792.25.20~1/RELEASE_I386

MHD:      Not yet compiled
Hybrid:   Not yet compiled
Electron: Not yet compiled
DSMC:     Not yet compiled
K3D:      Not yet compiled
F90 compiler: NOT FOUND (ifort)
F77 compiler: NOT FOUND (ifort)
C++ compiler: found (/usr/bin/g++)
Py compiler: found (/usr/bin/python)
One or more compilers were not found.  Edit Makefile.conf!
```

First note that this command tells you the version of ICES that you are running as well as giving you information about the computer system that you are using. Following this, the results would indicate that you have not yet compiled any of the codes (this is described in the next section). In addition, it indicates that the tool cannot find the ifort (Intel) compiler which you have selected to compile the Fortran codes. It did, however, find the g++ compiler for the C++ codes.

If your compiler does not appear in the list of compliers in the Makefile.conf file then you will have to add it. Because the code has not been tested with your compiler the codes may not work without some editing of the source code. See the later sections on installing compilers (Sections 4.1 and 4.2) in order understand how you can make the codes work with your compiler.

If you do not have a Fortran or C++ compiler you may try to install the free GNU f95 and g++ compilers. See the later sections on installing compilers.

2.5 Compilation

Before you can use the ICES_c.pl or ICES_n.pl tools to extract data from the simulation results you must compile the extraction codes for each physics model you are interested in. In order to compile all models type the following commands into the terminal:

```
make ALL
```

Or, to install selected components only (e.g. the Direct Simulation Monte Carlo and Hybrid models), type:

```
make DSMC HYB
```

The user can check the status of which models have been compiled and are ready to be used by typing:

```
make status
```

In the case where all the models are installed, the output should look like:

```
MHD:      Compiled
Hybrid:   Compiled
Electron: Compiled
DSMC:     Compiled
K3D:      Not yet compiled
F90 compiler: found (/usr/local/bin/f95)
F77 compiler: found (/usr/local/bin/f95)
C++ compiler: found (/usr/bin/g++)
Py compiler: found (/usr/bin/python)
```

Note that the compilers are also looked for as discussed earlier in Section 2.4.

2.6 Testing the code

2.6.1 Testing the ICES_c.pl coma code

The ICES tool ships with a very simple way of testing to make sure that the code is working as it should. The test looks for the compilers set in Makefile.conf, compiles the code, runs the ICES_c.pl tool using a sample trajectory file (Tests/traj_xaxis_example.dat) for all the models for the CG_1.3_au case and then compares the output with output files shipped with the distribution. The test can be run by simply typing

```
make test
```

Assuming that you have the compilers set right and that the test works as it should, the output should look like:

```
DSMC_CG_1.3_au_00.CO.dat: identical (within tolerance: 0.001, zeros= 1e-05)
DSMC_CG_1.3_au_00.Dust.dat: identical (within tolerance: 0.001, zeros= 1e-05)
DSMC_CG_1.3_au_00.H2O.dat: identical (within tolerance: 0.001, zeros= 1e-05)
Electron_CG_1.3_au_00.dat:   identical (within tolerance: 0.001, zeros= 1e-05)
Hybrid_CG_1.3_au_00.dat: identical (within tolerance: 0.001, zeros= 1e-05)
MHD_CG_1.3_au_00.dat: identical (within tolerance: 0.001, zeros= 1e-05)
```

This indicates that the values in the files created by the test are identical to the files provided in the distribution to within .1% (0.001). If there are larger differences in the files various error messages will appear. Identical tests indicate that your code is compiled and working properly. Note that the test compares to files that are located in the Tests/ directory.

If the tests are not identical, the script will produce an error message to let the user know what the difference between the two files is. The script will warn about the first difference that it finds.

Possible errors include

- Missing file: A file can be missing for several reasons including
  - The Hybrid model files are not compatible with Mac computers running Power PC processors so they are missing. On a PPC Mac there will be an error associate with this.

  - The MHD model files require different formats on Mac computers running Power PC processors. See the FAQ.
  - If the code fails for some other reason then file will be missing. You will have to look back at the output for a reason for the failure or consult the FAQ.

- Different number of lines in the two files. See the FAQ.
- Difference exceeds tolerance: This error may mean that there is a real problem or it may just mean that because you are using a different machine, with different compilers that the numbers are slightly different, but nothing to worry about. See the FAQ for a detailed description of how to tell the difference.

It is possible to compare the results of runs on various machines for the same case using the Scripts/TestCompare.pl script. This script is responsible for the comparison done in the test outlined above. Type

```
Scripts/TestCompare.pl -h
```

to learn more about this script and it's uses. You can, for example, change the tolerance which determines if the files are identical or not. In case you want to further investigate the output of the "make test" read also "Difference exceeds tolerance" in the FAQ section to learn where the corresponding files can be found.

2.6.2 Testing the ICES_n.pl nucleus code

This test has not yet been implemented. It is coming.

2.7 Running the code and examples

2.7.1 The ICES_c.pl script

In order to run the code to extract coma information, use the Perl script, ICES_c.pl, provided in the main directory. As with all scripts provided in the ICES tool, you can get help about running the script by typing:

```
ICES_c.pl -h
```

Additionally, you can find out which data files have been installed and which are ready to be used by typing

```
ICES_c.pl -s
```

This command looks in the Data/Coma/ directory for available cases and lists them as well as which models are available for this case. Also indicated is the version number of the files so that the users can check to make sure that they have the most up to date data files. Typical output might look like:

```
Current cases available in Data/Coma/:

CG_1.3_au_00     --      MHD    Hybrid      DSMC  Electron      (Original ISSI Case) (V1.4)
CG_2.0_au_00     --      MHD    Hybrid      DSMC  Electron      (Original ISSI Case) (V1.4)
CG_2.7_au_00     --      MHD    Hybrid      DSMC                (Original ISSI Case) (V1.4)
CG_2.7_au_01     --      MHD                DSMC                (H2O Driven) (V1.4)
CG_2.7_au_02     --                         DSMC                (H2O Driven) (V1.4)
CG_2.7_au_03     --      MHD                DSMC                (CO/CO2 Driven) (V1.4)
CG_2.7_au_04     --                         DSMC                (CO/CO2 Driven) (V1.4)
CG_2.7_au_05     --      MHD                DSMC                (H2O Driven/Jet) (V1.4)
CG_2.7_au_06     --      MHD                DSMC                (CO/CO2 Driven/Jet) (V1.4)
CG_3.0_au_01     --      MHD                                    (H2O Driven) (V1.4)
```

```
CG_3.0_au_02     --                          DSMC              (H2O Driven) (V1.4)
CG_3.0_au_03     --      MHD                 DSMC              (CO/CO2 Driven) (V1.4)
CG_3.0_au_04     --                          DSMC              (CO/CO2 Driven) (V1.4)
CG_3.0_au_05     --      MHD                                   (H2O Driven/Jet) (V1.4)
CG_3.0_au_06     --      MHD                 DSMC              (CO/CO2 Driven/Jet) (V1.4)
CG_3.0_au_07     --                          DSMC              (H2O Driven) (V1.5)
CG_3.0_au_08     --                          DSMC              (CO driven) (V1.5)
CG_3.0_au_09     --                          DSMC              (CG_3.0_au_07 with Q +20%) (V1.0)
CG_3.0_au_10     --                          DSMC              (CG_3.0_au_07 with Q -20%) (V1.0)
CG_3.0_au_11     --                          DSMC              (CG_3.0_au_07 with Q +50%) (V1.0)
CG_3.0_au_12     --                          DSMC              (CG_3.0_au_07 with Q -50%) (V1.0)
CG_3.3_au_00     --      MHD    Hybrid       DSMC              (Original ISSI Case) (V1.4)
```

In addition, you can find out the different cases available for the DSMC model by typing:

```
ICES_c.pl -ds
```

This command looks in the Data/Coma/ directory and, for each installed case, looks in its DSMC/ directory and lists all the different particles which are simulated by the model. Moreover, it gives a short description of the case including its dimension and the version number of the files, so that the users can check to make sure that they have the most up to date data files. Typical output might look like:

```
Current cases available for the DSMC model/:

CG_1.3_au_00     -- (2d) CO Dust H2O    (Original ISSI Case) (V1.4)
CG_2.0_au_00     -- (2d) CO Dust H2O    (Original ISSI Case) (V1.4)
CG_2.7_au_00     -- (2d) CO Dust H2O    (Original ISSI Case) (V1.4)
CG_2.7_au_01     -- (2d) CO CO2 Dust H2O        (H2O Driven) (V1.4)
CG_2.7_au_02     -- (2d) CO CO2 Dust H2O        (H2O Driven) (V1.4)
CG_2.7_au_03     -- (2d) CO CO2 Dust H2O        (CO/CO2 Driven) (V1.4)
CG_2.7_au_04     -- (2d) CO CO2 Dust H2O        (CO/CO2 Driven) (V1.4)
CG_2.7_au_05     -- (2d) CO CO2 Dust H2O        (H2O Driven/Jet) (V1.4)
CG_2.7_au_06     -- (2d) CO CO2 Dust H2O        (CO/CO2 Driven/Jet) (V1.4)
CG_3.0_au_01     -- (2d) CO CO2 Dust H2O        (H2O Driven) (V1.4)
CG_3.0_au_02     -- (2d) CO CO2 Dust H2O        (H2O Driven) (V1.4)
CG_3.0_au_03     -- (2d) CO CO2 Dust H2O        (CO/CO2 Driven) (V1.4)
CG_3.0_au_04     -- (2d) CO CO2 Dust H2O        (CO/CO2 Driven) (V1.4)
CG_3.0_au_05     -- (2d) CO CO2 Dust H2O        (H2O Driven/Jet) (V1.4)
CG_3.0_au_06     -- (2d) CO CO2 Dust H2O        (CO/CO2 Driven/Jet) (V1.4)
CG_3.0_au_07     -- (1d) C CO Dust H H2 H2O O OH        (H2O Driven) (V1.5)
CG_3.0_au_08     -- (1d) C CO Dust H H2 H2O O OH        (CO Driven) (V1.5)
CG_3.0_au_09     -- (1d) C CO Dust H H2 H2O O OH        (CG_3.0_au_07 with Q +20%) (V1.0)
CG_3.0_au_10     -- (1d) C CO Dust H H2 H2O O OH        (CG_3.0_au_07 with Q -20%) (V1.0)
CG_3.0_au_11     -- (1d) C CO Dust H H2 H2O O OH        (CG_3.0_au_07 with Q +50%) (V1.0)
CG_3.0_au_12     -- (1d) C CO Dust H H2 H2O O OH        (CG_3.0_au_07 with Q -50%) (V1.0)
CG_3.3_au_00     -- (2d) CO Dust H2O    (Original ISSI Case) (V1.4)
```

To extract data for modules you use ICES_c.pl with 4 arguments in the command line. The command we use to carry out the test (make test) is

```
ICES_c.pl Tests/traj_xaxis_example.dat CG_1.3_au_00 test_temp ALL
```

The arguments to the scripts are as follows

1. A trajectory file (Test/traj_xaxis_example.dat)
2. A case for which data is available (CG_1.3_au_00)
3. The directory where you want output to go (test_temp)
4. Which module you want to extract data from (1 or more items can be listed)

The format for the trajectory file is addressed in the following chapter. As outlined above, the case argument should be the name of one of the directories in the Data/Coma/ directory.

Note that the output directory must exist before the tool runs. If it does not exist you will loose the output. You can send the output to your current directory by using a command something like:

```
ICES_c.pl Tests/traj_xaxis_example.dat CG_1.3_au_00 . ALL
```

where here the "." indicates send output to the current directory.

The final argument is the list of modules that you wish to extract data from. Available options include:

Argument	Model
DSMC	Direct Simulation Monte Carlo model
ELEC	Electron model
HYB	Hybrid model
MHD	Magnetohydrodynamics model
ALL	Extract data from each of the above models

Arguments can also be typed in lowercase.

An output file is created for the output from each model selected and for each particle run by the DSMC. Output files are named with the name of the model followed by the name of the data input file case used and the name of the particle for the DSMC model. From the above test command the following output files would be generated.

```
DSMC_CG_1.3_au_00.CO.dat
DSMC_CG_1.3_au_00.CO2.dat
DSMC_CG_1.3_au_00.Dust.dat
DSMC_CG_1.3_au_00.H2O.dat
Electron_CG_1.3_au_00.dat
Hybrid_CG_1.3_au_00.dat
MHD_CG_1.3_au_00.dat
```

2.7.2 Coma Examples

Here we give a few examples of ways to run the ICES_c.pl tool. As outlined above, the following command is the one we use in the "make test" command.

```
ICES_c.pl Tests/traj_xaxis_example.dat CG_1.3_au_00 test_temp ALL
```

If you wish to use only the plasma models (MHD and Hybrid) you could use

```
ICES_c.pl Tests/traj_xaxis_example.dat CG_1.3_au_00 test_temp MHD HYB
```

If you wanted to place the output in the current working directory and only wanted to know what the neutral distribution was you could use only the DSMC by typing:

```
ICES_c.pl Tests/traj_xaxis_example.dat CG_1.3_au_00 . DSMC
```

For information of the format of the extracted data see the next chapter.

2.7.3 The ICES_n.pl script

In script order to run the code to extract nucleus information, use the Perl script, ICES_n.pl, provided in the main directory. As with all scripts provided in the ICES tool, you can get help about running the script by typing:

```
ICES_n.pl -h
```

Additionally, you can find out which data files have been installed and which are ready to be used by typing

```
ICES_n.pl -s
```

This command looks in the Data/Nucleus/ directory for available cases and lists them as well as which models are available for this case. Also indicated is the version number of the files so that the users can check to make sure that they have the most up to date data files. Typical output might look like:

```
Current nucleus models available:        K3D

============================================================

 Model   Case    Version         Description
------------------------------------------------------------
 K3D     CG_01   1.0             Nominal K3D model
```

In addition, you can find out the different orbits and heliocentric distances available for each case by typing:

```
ICES_n.pl -sv
```

Note that this command will produce a very lengthy output. Typical output might look like:

```
Current nucleus models available:        K3D

============================================================

 Model   Case    Orbit   Heliocentric distances
------------------------------------------------------------
 K3D     CG_01   01i     5.7
                 02i     1.3 1.4 1.5 1.6 1.7 1.8 1.9  ... 5.4 5.5 5.6 5.7 ap ph
                 02o     1.3 1.4 1.5 1.6 1.7 1.8 1.9  ... 5.4 5.5 5.6 5.7 ap
                 03i     1.3 1.4 1.5 1.6 1.7 1.8 1.9  ... 5.4 5.5 5.6 5.7 ap ph
                 03o     1.3 1.4 1.5 1.6 1.7 1.8 1.9  ... 5.4 5.5 5.6 5.7 ap
                 04i     1.3 1.4 1.5 1.6 1.7 1.8 1.9  ... 5.4 5.5 5.6 5.7 ap ph
                 04o     1.3 1.4 1.5 1.6 1.7 1.8 1.9  ... 5.4 5.5 5.6 5.7 ap
                 05i     1.3 1.4 1.5 1.6 1.7 1.8 1.9  ... 5.4 5.5 5.6 5.7 ap ph
                 05o     1.3 1.4 1.5 1.6 1.7 1.8 1.9  ... 5.4 5.5 5.6 5.7 ap
```

To extract data for modules you use ICES_n.pl with 6 arguments in the command line. The command we use to carry out the test (make test) is

```
ICES_n.pl -model=K3D Tests/nucleus_input_example.dat  CG_01 03i 2.5 test_temp
```

The arguments to the scripts (in required order) are as follows

1. The model we wish to use (-model=K3D)
2. A trajectory file (Test/nucleus_input_example.dat)

3. A case for which data is available (CG_01)
4. The orbit from which you want to extract data (03i)
5. The heliocentric distance you want to consider (2.5)
6. The directory where you want output to go (test_temp)

The format for the trajectory file is addressed in the following chapter. As outlined above, the case argument should be the name of one of the directories in the Data/Nucleus/ directory.

Note that the output directory must exist before the tool runs. If it does not exist you will loose the output. You can send the output to your current directory by using a command something like:

```
ICES_n.pl -model=K3D Tests/nucleus_input_example.dat  CG_01 03i 2.5 .
```

where here the "." indicates send output to the current directory.
Arguments can also be typed in lowercase.

An output file is created for the parameters you have selected. Output files are named with the name of the model followed by the name of the data input file case. From the above test command the following output file would be generated.

```
K3D__CG_01_03i_2.5au.dat
```

For information of the format of the extracted data see the next chapter.

2.8 Recompilation and clean up

Anytime you wish to change the compiler or are trying to recover from an error you probably should clean out any intermediate files that were created. To clean up files (.o, .mod, *∼, ...) which were used to build the executable and the documentation type

```
make clean
```

To remove the some additional files related to installation and any executable files that exist you can type:

```
make distclean
```

Once you have make a clean or a distclean the code be recompiled as outlined above.

Chapter 3

Input and Output Files

All input and output in the ICES tool should be in SI units (with few exceptions):

Quantity	Unit
position	m
density	$\#/m^3$
velocity	m/s
magnetic field	T (Tesla)
pressure	Pa (Pascal)
temperature	K (Kelvin)
energy	eV

3.1 ICES_c.pl and Coma Models

3.1.1 Coordinate system

The coordinate system used in each of the models is defined as follows:

- X: Points from the comet to the sun
- Y: Perpendicular to the X-axis lying essentially in the Solar Ecliptic plane. The vector then points in a sense that is retrograde to planetary motion. More importantly, this plane is the one which the models define as containing the solar wind magnetic field when a simple Parker Spiral field is used.
- Z: Perpendicular to the X and Y axes and positive in a direction toward ecliptic north.

3.1.2 Input trajectory file

Input trajectory files should contain a list of points at which the user wishes to extract data from the models. In general we might typically think of this file containing a list of points along a spacecraft trajectory. However, this is not a requirement. It is possible, for example, to ask for a three dimensional grid of regularly spaced points or any other list of points that might be desirable.

The trajectory files should have three columns in it corresponding to the three dimensional location of the desired points (X Y Z). Columns should be separated by spaces only. Note that the locations should be given in meters. For the downloadable version, data should start after a line containing the text "#START". Anything in the file before this command is ignored by the processing codes. This conveniently allows the user to add notes or other information about the trajectory file in the header.

A sample trajectory file which represents points along the x-axis of the models can be found in Tests/traj_xaxis_example.dat

```
This is a sample file which contains points
along the x axis.

#START
3.0e9 0.0 0.0
2.0e9 0.0 0.0
1.0e9 0.0 0.0
9.0e8 0.0 0.0
8.0e8 0.0 0.0
```

The format of the numbers in the individual columns should not matter.

For the GUI version, the user does not have to type the text "#START". Thus, in order to use the same sample file which contains points along the x axis with the GUI version, the user has to enter the trajectory as follow:

```
3.0e9 0.0 0.0
2.0e9 0.0 0.0
1.0e9 0.0 0.0
9.0e8 0.0 0.0
8.0e8 0.0 0.0
```

This file contains point which will be out of range for some of the physics models. In this case the physics model detects this and returns files that flag the error. See the next section.

3.1.3 Output files

Output files are written into the directory listed in the ICES_*c*.pl command. A unique output file is created for the output from each model selected. Output files are named with the name of the model followed by the name of the data input file case used. A typical list of output files might be:

```
DSMC_CG_1.3_au_00.CO.dat
DSMC_CG_1.3_au_00.CO2.dat
DSMC_CG_1.3_au_00.Dust.dat
DSMC_CG_1.3_au_00.H2O.dat
Electron_CG_1.3_au_00.dat
Hybrid_CG_1.3_au_00.dat
MHD_CG_1.3_au_00.dat
```

Most output files have similar formats (DSMC, Hybrid, MHD). In these files interpolated values follow the "#START" line the file. Above this line is header information which may be unique the model. In each file there will be a list of variables in the following format:

```
#VARIABLES = Status, X, Y, Z, n, Vx, Vy, Vz, Bx, By, Bz, p
```

There may be additional information in the header as well. Each quantity is listed in a column.

The status variable is used to indicate errors in the processing. Possible values for this quantity are:

Status Value	Meaning
0	Value is fine
1	Position of point was outside the simulation domain

Additional values may be added in the future.

The format for the electron model output is quite different than the other models due to number of data values at each location. The format of the file is described below.

MHD

MHD output files contain the standard MHD variables.

Variable	Physical Quantity
Status	Indicates the validity of this data point, outlined above
X, Y, Z	Position
n	Ion (and electron) mass density (see discussion below)
Vx, Vy, Vz	Velocity components
Bx, By, Bz	Magnetic field components
p	Thermal pressure in Pascals (p=nkT)

We need to note several unique aspects of the MHD files. First, because the MHD simulations are single species, there is no way of knowing the ion composition. Near the comet the plasma would have an average mass of $\tilde{1}8$ amu (H_2O) and far upstream of the comet the average mass would be $\tilde{1}$ amu (H). In regions where ions are mixed it is not possible to determine the average mass. For this reason the mass density for the comet is given in amu/m^{-3}. Near the comet or in the far upstream solar wind the number density can be determined. In mixing regions it cannot.

The second important aspect of the MHD output to note is that the pressure is provided rather than the temperature. The two are obviously related

$$P = n_e K T_e + \sum_s n_s K T_s \tag{3.1}$$

where n_e, T_e refer to the electron and n_s, T_s refer to individual ion species. From the above discussion it should be clear that it is not always possible to determine n_e and n_s. Assuming that we are in a region where we know the average mass and also assuming that the temperature of all ion species is the same (T_i) yields

$$P = \frac{\rho}{< m >} K(Te + T_i) \tag{3.2}$$

where i indicates a sum over all the ions, ρ is the mass density and $< m >$ is the average mass. In this region we can determine the sum of the ion and electron temperatures but cannot determine either individually because we do not know the ratio T_i/T_e.

Hybrid

The hybrid model can generated distribution functions of ions. However, in this version of ICES we are only able to make available moments of the distribution function. Available quantities are:

Variable	Physical Quantity
x, y, z	Position
Bx, By, Bz	Magnetic field components
Ex, Ey, Ez	Electric field components
rhosw	Solar wind ion number density
uswx, uswy, uswz	Velocity components of the solar wind ions
rhohi	Oxygen ion number density
uhix, uhiy, uhiz	Velocity components of Oxygen ions

Electron

The data format of the Electron data files are quite a bit more complicated than the data products from the other models. This is because the files contain the electron flux in a list of energy bins at each point. There are 185 energy bins and in each we give both the upward and downward electron flux. The file contains the

data at each point in blocks. The first block in the files is a header which indicates the energy values (in keV) of each energy bin. The first lines of this block look like:

```
Energy Bins
  1- 10      0.25      0.75      1.25      1.75      2.25      2.75     ...
 11- 20      5.25      5.75      6.25      6.75      7.25      7.75     ...
 21- 30     10.50     11.50     12.50     13.50     14.50     15.50     ...
```

where the first entry is the bin number, with ten bins per row. In that row then follow the energy values for each bin.

The data begins with a "#START" line as in the other data files.

What follows is two blocks of data for each point in the trajectory file. The first block of data includes the upward flux of electrons while the next block contains the downward flux of electrons. The format is as follows:

```
 9.000E+07 0.000E+00 0.000E+00 X, Y, Z, Upward Flux
  1- 10 7.363E+04 8.687E+04 7.363E+04 5.409E+04 1.220E+05 1.899E+05     ...
 11- 20 5.207E+05 5.439E+05 5.956E+05 6.573E+05 6.766E+05 7.287E+05     ...
 21- 30 9.293E+05 9.478E+05 9.869E+05 1.044E+06 1.069E+06 1.123E+06     ...
...
 9.000E+07 0.000E+00 0.000E+00 X, Y, Z, Downward Flux
  1- 10 7.363E+04 5.409E+04 3.889E-04 1.133E-05 1.129E-05 1.124E-05     ...
 11- 20 1.098E-05 1.093E-05 1.088E-05 1.082E-05 1.077E-05 1.072E-05     ...
 21- 30 1.044E-05 1.038E-05 1.033E-05 1.027E-05 1.021E-05 1.016E-05     ...
```

where here the first block indicates that this contains the upward flux for the listed location X,Y, Z. Each line of the block starts with the bin range of that row followed by the fluxes in each bin. The second block lists the same X, Y, Z but now contains the downward flux.

DSMC

Files for the DSMC model follow the format of the MHD and Hybrid models. As with the Hybrid model, the DSMC model is capable of computing distribution functions of the neutral molecules. In addition, the DSMC model can compute distribution functions of the dust particles. However, at this time we are only able to provide the moments of the distribution function.

Currently available quantities in the DSMC files for the neutrals are:

Variable	Physical Quantity
Status	Indicates the validity of this data point, outlined above
x, y, z	Position
Vx, Vy, Vz	Neutral gas velocity components
T	Neutral gas thermal temperature
N	Neutral gas number density

The DSMC dust output file contains the following variables:

Variable	Physical Quantity
Status	Indicates the validity of this data point, outlined above
x, y, z	Position
Vx(R=a), Vy(R=a), Vz(R=a)	Dust particles of radius a velocity components
NUMBERDENSITY(R=a)	Dust particles of radius a number density
MASSDENSITY(R=a)	Dust particles of radius a mass density

3.2 ICES_n.pl and Nucleus Models

3.2.1 Coordinate system

The coordinate system used is the nucleus models in ICES is a sun-fixed coordinate system. All angles are measured from the Sun-Comet line (the sub-solar point).

- Latitude (in degrees)
- Solar Hour angle (SHA, in degrees): This is an angle similar to longitude, except sun-fixed. 0 degrees corresponds to the sub-solar point (noon) while 180 degrees corresponds to midnight. 90 and 180 degrees correspond to west and east (left and right) when looking from the sun to the comet.
- Depth (in meters): The surface is given by using "0". Other positive numbers indicate the depth below the surface.

3.2.2 Input trajectory file

Input trajectory files should contain a list of points at which the user wishes to extract data from the models. The may wish to use a list of points at fixed LAT, SHA and then scan across depth. Or perhaps use depth "0" for all points and grid the surface. However, in general there is no restriction on the way the points are selected or ordered.

The trajectory files should have three columns in it corresponding to the three dimensional location of the desired points (LAT, SHA, D). Columns should be separated by spaces only. Note that the locations should be given in meters. For the downloadable version, data should start after a line containing the text "#START". Anything in the file before this command is ignored by the processing codes. This conveniently allows the user to add notes or other information about the trajectory file in the header.

A sample trajectory file can be found in Tests/nucleus_input_example.dat

```
#MODEL CG1
# Latitude[deg] Hour-Angle[deg] Depth[m]
#START
90.0    0.0     -1.0
80.0    20.0    0.0
70.0    40.0    0.01
60.0    60.0    0.02
50.0    80.0    0.0
40.0    100.0   0.06
30.0    120.0   0.07
20.0    140.0   0.08
10.0    160.0   0.09
0.0     180.0   0.1
-10.0   200.0   0.5
-20.0   220.0   1.0
-30.0   240.0   2.0
-40.0   260.0   5.0
-50.0   280.0   10.0
-60.0   300.0   20.0
-70.0   -60.0   50.0
-80.0   -40.0   70.0
-90.0   -20.0   99.9
```

The format of the numbers in the individual columns should not matter.

For the GUI version, the user does not have to type the text "#START". Thus, in order to use the same sample file which contains points along the x axis with the GUI version, the user has to enter the trajectory as follow:

```
90.0    0.0     -1.0
80.0    20.0    0.0
70.0    40.0    0.01
60.0    60.0    0.02
50.0    80.0    0.0
40.0    100.0   0.06
30.0    120.0   0.07
20.0    140.0   0.08
10.0    160.0   0.09
0.0     180.0   0.1
-10.0   200.0   0.5
-20.0   220.0   1.0
-30.0   240.0   2.0
-40.0   260.0   5.0
-50.0   280.0   10.0
-60.0   300.0   20.0
-70.0   -60.0   50.0
-80.0   -40.0   70.0
-90.0   -20.0   99.9
```

This file contains point which will be out of range for some of the physics models. In this case the physics model detects this and returns files that flag the error. See the next section.

3.2.3 Output files

Output files are written into the directory listed in the ICES_n.pl command. A unique output file is created for the output from each model selected. Output files are named with the name of the model followed by the name of the data input file case used. A typical output files might be:

```
K3D_CG_01_03i_2.5au.dat
```

Most output files have similar formats. In these files interpolated values follow the "#START" line the file. Above this line is header information which may be unique the model. In each file there will be a list of variables. There may be additional information in the header as well. Each quantity is listed in a column.

K3D

K3D output files contain the following variables:

Column	Physical Quantity	Valid Depth	Unit
1	Status	All	
2	Latitude	All	[deg]
3	Solar hour angle	All	[deg]
4	Depth	All	[m]
5	Flag	All	
6	Temperature	All	[K]
7	Mass Density	Sub-surface	[Kg m^{-3}]
8	Porosity	Sub-surface	
9	Pore size	Sub-surface	[m]
10	Amorphous Ice MF	Sub-surface	
11	Crystalline Ice MF	Sub-surface	
12	Permanent dust MF	Sub-surface	
13	Dust 1um MF	Sub-surface	
14	Dust 10um MF	Sub-surface	
15	Dust 100um MF	Sub-surface	
16	Dust 200um MF	Sub-surface	
17	Water flux	Surface	[molecule $sec^{-}1$ m^{-2}]
18	1um dust flux	Surface	[Kg sec^{-1} m^{-2}]
19	10um dust flux	Surface	[Kg sec^{-1} m^{-2}]
20	100um dust flux	Surface	[Kg sec^{-1} m^{-2}]
21	200um dust flux	Surface	[Kg sec^{-1} m^{-2}]

Chapter 4

Troubleshooting

4.1 Installing compilers

The free GNU Fortran compiler (gfortran) can be downloaded from

http://gcc.gnu.org/wiki/GFortran

We recommend that you install the most recent stable version for your platform. You will download a gzipped tar file that you will have to unpack. Once you have unpacked the file, follow the instructions in the included PDF file to install the compiler.

4.2 Using your unlisted compiler

If you are using a compiler that is not already listed in the Makefile.conf file you will have to make a new entry in that file. You can test that the tool can find your compiler by typing

```
make status
```

One the tool can locate your compiler, you may still have to edit the source code or Makefiles of the individual models in order to get the compiler flags to work correctly or to overcome any compiler specific bugs that may still be in the code or in the compiler itself. From experience, we know that since we have ported this code to several different machines and compilers that for most platforms using a new compiler should be fairly simple however there may still be some issues. For each model we list below the locations of Makefile which you may need to modify.

- DSMC: DSMC/makefile
- Electron: Electron/Makefile
- MHD: MHD/src/Makefile
- Hybrid: Hybrid/Makefile

For additional problems you may have to contact us to see if we are able to help you with you individual problem.

4.3 FAQ

4.3.1 MHD: Unable to read from opened MHD file

The MHD data files are binary and they are machine specific. Apple Macintosh computers with Power PC (PPC = G5 processor or before) use a special format. All other machines (including Apple computers with Intel processors) use the more common format.

The following error

```
read_1block_file: unable to read from opened MHD file named : MHDRestart/blk00001.rst
```

generally indicates that you have downloaded the incorrect data files for your computer's processor. You can check the type of your machine's processor by typing the following command

```
uname -p
```

If the result is "powerpc" then you should use the version of the files on the ICES web page labeled "Version for MACs with PPC Processors (G5 or before)." Otherwise you download the files with the label "Version for most systems."

4.3.2 The last line of the output file is missing

Due to the implementation of the gcc compiler, the input trajectory file must end with at least one carriage return to a blank line. If this is not the case in your file the last line of the trajectory file will not be interpolated to and the line will be missing from the output file of some or all of the different models depending on the selected compilers. You can easily fix this by adding one or more blank lines to the end of the trajectory file.

4.3.3 Errors when running "make test"

See the section on Testing the Code.

4.3.4 Difference exceeds tolerance in "make test"

There are various reasons why you might receive this error message. Typically, the reason is dependent on what machine and/or compiler you are using. Most commonly, the error is a result of how floating point mathematics (when doing interpolations on your trajectory file) and the associated round-off errors are handled. These round-off errors can lead to slightly different results on different machines. As part of "make test" the output files in the distribution are compared with the output files created on your machine to check for differences. The error message indicates the first point in a trajectory file where the percent difference between the two data files exceeds a given tolerance (for ICES currently 0.1 %). Some times these errors are significant but often they are not, even when highlighted by the code. In order to investigate if your installation is working properly and if these errors are significant or not we provide some hints for things to look for:

- Open the two files containing the results of the corresponding code. The baseline file (the "good" file), which is shipped with the ICES distribution, can be found in the following ICES subfolder

  ```
  /Tests/Test_CG_1.3_au_00/
  ```

  and the corresponding output file created on your machine during the "make test" procedure in

  ```
  /test_temp/
  ```

For example, the files of the MHD code can be found at:

```
/Tests/Test_CG_1.3_au_00/MHD_CG_1.3_au_00.dat
/test_temp/MHD_CG_1.3_au_00.dat
```

- You will need to look at the two files in detail starting with the specific line and variable called out by the script. Since the code only locates the first error there may be more, possibly larger, differences that you should also examine. Graphical differencing tools are great for this exercise (tkdiff, xdiff). Another simple examination method is to import both files into a spreadsheet.
- You want to determine if the differences you find are significant or not. Often differences are not significant because they represent fluctuations around zero. In the MHD simulation, for instance, the upstream boundary conditions for the solar wind velocity are (u_x=400 km/s,u_y=0,u_z=0). Far upstream however, due to roundoff errors, uy_{sw} and uz_{sw} are very small but not exactly zero. Since the small, non-zero value is likely machine dependent, the numbers in the two files may be very different (often more than an order of magnitude). However, compared to the value of u_x they are orders of magnitude smaller and the errors are physically insignificant. When errors like this occur, they can be ignored.
- If the values with errors do not seem to be near zero and are therefore physically significant, a large error probably indicates that you have a problem with the installation and/or the compiler. You may need to contact the ICES administrators.

Chapter 5

Available Cases for the Coma

The ICES was develop as part of an ISSI comet modeling project that concentrated on the Rosetta mission target comet 67P/Churyumov-Gerasimenko. The first cases that are available for use with the tool therefore are for comet 67P/CG. Other comets and cases will be added periodically.

5.1 67P/Churyumov-Gerasimenko

Comet 67P/CG has been observed on a number previous perihelion passages, but not intensively. Once it was identified as the Rosetta target - well after the most recent perihelion passage - a more concerted effort was begun to monitor at least its outgoing leg. The ISSI comet modeling team chose to model the comet at four representative heliocentric radial distances that represent either significant times of the mission or serve to span the range of plasma environments that can be expected during the mission. These locations and their relation to the mission milestones have changed during the time that these simulations were developed, but the cases are still representative of the range of environments that the comet will experience.

In order to model the comet for the cases listed above, we must choose representative solar wind conditions. So that the models will be reasonably self consistent, we have based the solar wind conditions on nominal values at 1AU. At that location we use u = 400 km/s, $n = 10cm^{-3}$, $T_e = 1 \times 10^5$ K, $T_i = 5 \times 10^4$ K and $B = 7$nT confined to the ecliptic plane, making an angle representative of the Parker spiral at 1AU. We then use an extrapolation to obtain the conditions for each of the cases. The above Table also gives values in the upstream solar wind for density (n), velocity (v), magnetic field magnitude (B) and Parker spiral angle, the Alfvén speed (v_a), and the electric field (E). Clearly these values are only representative of the conditions that comet 67P/CG may encounter during the Rosetta mission, however as average values they will give a good idea of the type and scale of the expected interaction between the comet and the solar wind.

Below an additional table indicates the values of the cometary neutral production rate which are applied in the models. Both the gas production rate, Q, and the ionization scale length, λ, depend on heliocentric distance. The table indicates the values for these parameters that we have adopted.

Note that the MHD and hybrid models use a spherically symmetric source while the DSMC mode uses an axisymmetric source terms with day night asymmetry. More detail about the parameters used to run each of the models can be found in the following papers:

Case Name	Heliocentric Distance	Afρ	Dust Distribution	Nucleus Density	MHD Mean Ion Mass	Additional Description
	AU	cm	r=10^{-7}...10^{-2} m	kg/m^3	amu	
CG_1.3_au_00	1.3	100 to 500	-4	300	17	Perihelion: Original IS
CG_1.3_au_01	1.3		x	300	17	min perihelion C
CG_1.3_au_02	1.3		x	300	17	max perihelion C
CG_2.0_au_00	2.0	40 to 300	-4	300	17	Original ISSI Ca
CG_2.0_au_01	2.0		-3.5	300	17	Detailed Chemis
CG_2.0_au_02	2.0		x	300	17	Min Case
CG_2.0_au_03	2.0	40 to 300	x	300	17	Max Case
CG_2.7_au_00	2.7	25 to 250	-4	300	17	Original ISSI Ca
CG_2.7_au_01	2.7	20	-4	300	17	H2O Driven
CG_2.7_au_02	2.7	10	-4	300	17	H2O Driven
CG_2.7_au_03	2.7	250	Knee*	300	35	CO/CO2 Drive
CG_2.7_au_04	2.7	25	Knee*	300	35	CO/CO2 Drive
CG_2.7_au_05	2.7	20	-4	300	17	H2O Driven/Je
CG_2.7_au_06	2.7	250	Knee*	300	35	CO/CO2 Driven/
CG_3.0_au_01	3.0	10	-4	300	17	H2O Driven
CG_3.0_au_02	3.0	5	-4	300	17	H2O Driven
CG_3.0_au_03	3.0	200	Knee*	300	35	CO/CO2 Drive
CG_3.0_au_04	3.0	20	Knee	300	35	CO/CO2 Drive
CG_3.0_au_05	3.0	10	-4	300	17	H2O Driven/Je
CG_3.0_au_06	3.0	200	Knee*	300	35	CO/CO2 Driven/
CG_3.0_au_07	3.0		-3.5	300	17	H2O Driven
CG_3.0_au_08	3.0		-3.5	300	17	CO Driven
CG_3.0_au_09	3.0		-3.5	300	17	CG_3.0_au_07 with Q
CG_3.0_au_10	3.0		-3.5	300	17	CG_3.0_au_07 with C
CG_3.0_au_11	3.0		-3.5	300	17	CG_3.0_au_07 with Q
CG_3.0_au_12	3.0		-3.5	300	17	CG_3.0_au_07 with C
CG_3.0_au_13	3.0		-3.5	300	17	CG_3.0_au_07 with Knudsen layer
CG_3.0_au_14	3.0		-3.5	300	17	CG_3.0_au_07 with Knudsen layer
CG_3.0_au_15	3.0		x	300	17	Min Case
CG_3.0_au_16	3.0		x	300	17	Max Case
CG_3.3_au_00	3.3	5 to 150	-4	300	17	Rendezvous: Original I
CG_3.5_au_01	3.5	5 to 150	-3.5	300	17	Nominal Case
CG_3.5_au_02	3.5	5 to 150	-3.5	300	17	CO/CO2 Drive
CG_3.5_au_03	3.5	5 to 150	-3.5	300	17	H2O Driven,Low produc

** Knee distribution: -3 for dust particles with radius up to 1mm and -4 beyond*

Case Name	Dimension DSMC	Production Rate Total	H_2O	CO	CO_2	Dust
		molecules/s				kg/s
CG_1.3_au_00	2d	5×10^{27}	4.76×10^{27}	2.4×10^{26}	0	114
CG_1.3_au_01	2d	4.14×10^{27}	4×10^{27}	4×10^{25}	1×10^{26}	0
CG_1.3_au_02	2d	1.13×10^{28}	1×10^{28}	5×10^{26}	8×10^{26}	0
CG_2.0_au_00	2d	8×10^{26}	7.62×10^{26}	3.8×10^{25}	0	23
CG_2.0_au_01	1d	8.7×10^{26}	7.62×10^{26}	3.8×10^{25}	1.9×10^{25}	23
CG_2.0_au_02	2d	4.14×10^{26}	4×10^{26}	4×10^{24}	1×10^{25}	0
CG_2.0_au_03	2d	2.4×10^{27}	2×10^{27}	2×10^{26}	2×10^{26}	0
CG_2.7_au_00	2d	8×10^{25}	7.62×10^{25}	3.8×10^{24}	0	2
CG_2.7_au_01*	2d	2.4×10^{26}	2×10^{26}	2×10^{25}	2×10^{25}	10
CG_2.7_au_02	2d	2.4×10^{26}	2×10^{26}	2×10^{25}	2×10^{25}	5
CG_2.7_au_03	2d	2.2×10^{27}	2×10^{26}	1×10^{27}	1×10^{27}	80
CG_2.7_au_04	2d	2.2×10^{27}	2×10^{26}	1×10^{27}	1×10^{27}	8
CG_2.7_au_05**	2d	2.4×10^{26}	2×10^{26}	2×10^{25}	2×10^{25}	10
CG_2.7_au_06***	2d	2.98×10^{27}	2×10^{26}	1.39×10^{27}	1.39×10^{27}	80
CG_3.0_au_01	2d	6×10^{25}	5×10^{25}	5×10^{24}	5×10^{24}	5
CG_3.0_au_02	2d	6×10^{25}	5×10^{25}	5×10^{24}	5×10^{24}	2
CG_3.0_au_03	2d	2.1×10^{27}	5×10^{25}	1×10^{27}	1×10^{27}	60
CG_3.0_au_04	2d	2.1×10^{27}	5×10^{25}	1×10^{27}	1×10^{27}	6
CG_3.0_au_05**	2d	6×10^{25}	5×10^{25}	5×10^{24}	5×10^{24}	35
CG_3.0_au_06***	2d	2.98×10^{27}	5×10^{25}	1.47×10^{27}	1.47×10^{27}	60
CG_3.0_au_07	1d	5.5×10^{25}	5×10^{25}	5×10^{24}	0	60
CG_3.0_au_08	1d	1.1×10^{27}	1×10^{26}	1×10^{27}	0	60
CG_3.0_au_09	1d	6.6×10^{25}	6×10^{25}	6×10^{24}	0	72
CG_3.0_au_10	1d	4.4×10^{25}	4×10^{25}	4×10^{24}	0	48
CG_3.0_au_11	1d	8.25×10^{25}	7.5×10^{25}	7.5×10^{24}	0	90
CG_3.0_au_12	1d	2.75×10^{25}	2.5×10^{25}	2.5×10^{24}	0	30
CG_3.0_au_13	1d	5.5×10^{25}	5×10^{25}	5×10^{24}	0	60
CG_3.0_au_14	1d	5.5×10^{25}	5×10^{25}	5×10^{24}	0	60
CG_3.0_au_15	2d	2.08×10^{24}	2×10^{24}	2×10^{22}	6×10^{22}	0
CG_3.0_au_16	2d	3.6×10^{26}	3×10^{26}	3×10^{25}	3×10^{25}	0
CG_3.3_au_00	2d	1×10^{24}	9.52×10^{23}	4.8×10^{22}	0	0.03
CG_3.5_au_01	1d	6×10^{26}	5×10^{26}	5×10^{25}	5×10^{25}	21
CG_3.5_au_02	1d	1.05×10^{28}	5×10^{26}	5×10^{27}	5×10^{27}	61
CG_3.5_au_03	1d	2×10^{23}	1.67×10^{23}	1.67×10^{22}	1.67×10^{22}	0.007

** The rest of the total production rate comes from others parent species*

*** 50% of productions as in the "non-jet" case and the other 50% of productions are due to the jet in an area centered on the sub-solar point which represents 10% of the total surface of the nucleus.*

**** The size of the jet is adjusted so that the right dust size is lifted off, that is to say 1 cm particles.*

Location & Status				Solar Wind				
Case	Rosetta Status	R(AU)	n(cm^{-3})	v(km/s)	v_A(km/s)	B(nT)	Angle	E(mV/m)
a	Instruments on	3.25	0.9	400	36.8	1.6	73	0.61
b		3.0	1.13	400	36.9	1.9	71	0.67
c	Lander Down	2.7	1.4	400	36.9	2.0	70	0.75
d		2.0	2.5	400	38.7	2.8	63	1.00
e	Perihelion	1.3	6.0	400	43.7	4.9	52	1.54

Location & Status			Neutral Coma	
Rosetta Status	R (AU)	Case	Q (s^{-1})	λ (km)
Instruments on	3.25	Original ISSI	1×10^{24}	1.1×10^{7}
	3.0	H_2O driven	6×10^{25}	8.9×10^{6}
	3.0	CO/CO_2 driven	2.1×10^{27}	8.9×10^{6}
Lander Down	2.7	Original ISSI and H_2O driven	8×10^{25}	7.3×10^{6}
Lander Down	2.7	CO/CO_2 driven	2.2×10^{27}	7.3×10^{6}
	2.0	Original ISSI	8×10^{26}	4.0×10^{6}
Perihelion	1.3	Original ISSI	5×10^{27}	1.7×10^{6}

Chapter 6

Available Cases for the Nucleus

Chapter 7

Physic Module Details

7.1 MHD

Please see the following papers for details about the MHD model.

- Hansen, K.C., T. Bagdonat, U. Motschmann, C. Alexander, M.R. Combi, T.E. Cravens, T.I. Gombosi, Y.-D. Jia, and I.P. Robertson, The plasma environment of Comet 67P/Churyumov-Gerasimenko throughout the Rosetta main mission (2007), *Space Sci. Rev.*, *70*, 133-166, doi:10.1007/s11214-006-9142-6.
- Gombosi, T.I., K.C. Hansen, D.L. De Zeeuw, M.R. Combi, and K.G. Powell (1997), MHD simulation of comets: The plasma environment of comet Hale-Bopp, *Earth, Moon and Planets*, *79* 179-207, doi:10.1023/A:1006289418660.
- Gombosi, T.I, D.L. De Zeeuw, R.M. Häberli and K. G. Powell (1996), Three-dimensional multiscale MHD model of cometary plasma environments, *J. Geophys. Res."*, *101*, 15233–15253.

The magnetohydrodynamics (MHD) treats the cometary plasma environment in the fluid approximation using the global, 3D magnetohydrodynamic (MHD) code BATSRUS (Block Adaptive Tree Solarwind Roe-Type Upwind Scheme). The code solves the governing equations of ideal magnetohydrodynamics. In the MHD limit, the plasma is treated as a single species plasma. Effects related to finite gyroradius, resistivity and other kinetic effects are neglected. However, we are able to describe some non-MHD effects by describing these deviations from ideal MHD (sources and sinks of mass, momentum and energy, or resistive and diffusive processes) through appropriate source terms. Here we present a few details about the aspects of BATSRUS that are relevant to the solar wind–comet interaction and refer the reader to an extensive literature related to the numerical algorithms used in the code. A review of many of the fundamental aspects of the model algorithms and their application to comets can be found in papers listed above an references contained in them.

One of the most important features of BATSRUS is the ability to easily adapt the grid to resolve specific regions of space. BATSRUS utilizes the approach of adaptive blocks. Adaptive blocks partition space into regions, each of which is a regular Cartesian grid of cells, called a block. If the region needs to be refined, then the block is replaced by 8 child sub-blocks, each of which contains the same number of cells as the parent block. The use of cells of varying sizes allows us to resolve features of vastly different length scales. In the simulations presented in this paper, we are able to resolve the body, the diamagnetic cavity and the solar wind–neutral cloud interaction even though these occur at vastly different length scales Note that the adaptive mesh allows us to use an extremely large simulation box and therefore self-consistently model the mass loading and subsequent slowing of the plasma upstream of the bow shock.

In the model all processes associated with mass loading (ionization, charge exchange, recombination, etc.) are handled through appropriate sources terms. These are implemented in the same manner as the

source terms used in our previous study of the comet Hale-Bopp Differences from that paper only require adjusting the gas production rate to that of comet 67P/CG and adjusting the photoionization scale length for heliocentric distance.

7.2 Hybrid

Nikolaos Gortsas, Uwe Motschmann, Ekkehard K uhrt
Technical University of Braunschweig and German Aerospace Center (DLR)
Contact: nikolaos.gortsas@dlr.de, u.motschmann@tu-bs.de

A fully three-dimensional hybrid plasma simulation code developed since the year 2000 at the Technical University of Braunschweig[1;2] is applied to the study of the plasma environment of Comet 67P/Churyomuv-Gerasimenko - the target comet of ESA's Rosetta Mission.

The hybrid model is characterized by a semi-kinetic treatment of the particle dynamics. In the presented calculations we consider two different species: solar wind protons and the much heavier cometary ions. The particle dynamics of each species is described by Newtonian equations of motions with the Lorentz force and a collision force modeling interactions with the neutral gas. The electrons are assumed to be a massless, charge neutralizing fluid which is described by conservation laws. An equation for the electric field is derived from the electron model while the time evolution of the magnetic field is obtained from Faraday's law. This model resolves physical structures down to a time scale below the inverse lower hybrid frequency.

The code solves the spatial part of the Maxwell equations on an arbitrary curvilinear, hexahedral, ordered grid by applying finite difference schemes. The time integration is done by a cyclic leapfrog scheme with a modified current advancement method [3]. In addition, in order to achieve a higher resolution of dispersive effects the magnetic field time evolution is obtained from a subcycling leapfrog method[1].

We conduct a series of hybrid plasma simulations for four different heliocentric distances covering the range between 3.25 AU and 1.30 AU, which corresponds to important stages of the Rosetta mission. The used physical parameters are described in [4].

1. Bagdonat T. and Motschmann U.: 3d hybrid simulation code using curvilinear coordinates. J. Comp. Phys., 183, p. 470-485, 2002

2. Bagdonat T. and Motschmann U.: From a weak to a strong comet - 3d global hybrid simulation studies. Earth, Moon and Planets, 90, p. 305-321, 2002

3. Matthews, A. P., Current advance method and cyclic leapfrog for 2D multispecies hybrid plasma simulations, J. Comp. Phys., 112, p. 102-116, 1994

4. Hansen K. C., Bagdonat T., Motschmann U., Alexander C., Combi M. R., Cravens T.E., Gombosi, T.I., Jia Y.-D. and Robertson I.P. : The Plasma Environment of Comet 67P/Churyumov-Gerasimenko Throughout the Rosetta Main Mission, Space Science Reviews, 128, p. 133-166, 2007

7.3 Electrons

The electron component of the cometary plasma is not a simple Maxwellian due to the several electron sources including the shocked and unshocked solar wind, secondary electrons produced by the impact of solar wind electrons and photoelectrons. In each case, collisions with the neutral gas of the cometary coma modifies the electrons as they travel along magnetic field lines.

We model the electron population using methods described by Gan et al. (1990). Gan constructed a model of the spatial and energy distributions of electrons in the vicinity of comet Halley. The method involves the calculation of suprathermal electron fluxes using a two-stream transport technique which included electron impact cross sections for rotational, vibrational, and electronic excitation of water molecules. Their model

also calculated electron temperatures for the thermal electron population and found that cold, "ionospheric" electrons should indeed be present for cometocentric distances less than a few thousand km.

The Rosetta target comet, 67P/Churyumov-Gerasimenko, even near perihelion, is a considerably weaker comet than Halley was in 1986. Nonetheless, we have undertaken electron modeling for this comet near perihelion using the methods described in Gan et al. (1990). In our model, the electron transport is constrained by the magnetic field topology which we have extracted from the perihelion results of the MHD model discussed above. For the results presented here, the field line shape is approximately parabolic over the region of the coma where the calculation occurs. In principle, the model can be used for any field line which threads the coma, however below we present results from a representative field line which crosses the Sun-comet line at 50km upstream of the nucleus.

- Gan, L. and Cravens, T. E. 1990, J. Geophys. Res., 95, 6285

7.4 DSMC

Please see the following papers for details about the DSMC model:

- Rubin, M., V.M. Tenishev, M.R. Combi, K.C. Hansen, T.I. Gombosi, K. Altwegg and H. Balsiger (2011), Monte Carlo modeling of neutral gas and dust in the coma of comet 1P/Halley. *Icarus*, in press, doi:10.1016/j.icarus.2011.04.006.

- Tenishev, V.M., M.R. Combi, and M. Rubin (2011), Numerical simulation of dust in a cometary coma: Application to Comet Churyumov-Gerasimenko. *Astrophys. J.*, 732:104 (17pp), doi:10.1088/0004-637X/732/2/104.

- Tenishev, V.M., M.R. Combi, and B.J.R. Davidsson (2008), A global kinetic model for cometary comae: The evolution of the coma of the Rosetta target comet Churyumov-Gerasimenko throughout the mission. *Astrophys. J.*, 685, 659-677.¡/li¿¡/ul¿

- Combi, M.R. , W.M. Harris and W.H. Smyth (2004), Gas dynamics and kinetics in the cometary coma: Theory and observations. In Comets II (M.C. Festou, H. U. Keller, H. A. Weaver, eds.), *U. Arizona Press*, Tucson, p. 523-552.

- Combi, M.R. (1996), Time-dependent gas kinetics in tenuous planetary atmospheres: The cometary coma. *Icarus*, 123, 207-226.

Kinetic approaches to modeling the cometary coma recognize that in much of the coma, collisions occur between molecules, but not enough to maintain thermal equilibrium in large and sometimes critical regions. Fully kinetic models based on the Direct Simulation Monte Carlo (DSMC) method have been applied to the cometary coma as a practical method to solving the general collisional Boltzmann equation. DSMC was developed to simulate the transition regime, where the mean free path of molecules is too large for continuum hydrodynamics to be applicable. Therefore, gas molecules are modeled as a sample of individual particles and simulated as they move around within a grid, colliding with other particles and with solid objects (if any). Macroscopic properties, such as density, velocity and temperature are computed by performing the standard kinetic theory averaging of particle masses, locations, velocities, and internal energies over the particle distribution function. DSMC is based on the rarefied-gas assumption that over a short time interval or 'step' the molecular motion and the intermolecular collisions are uncoupled and therefore can be calculated independently. Molecules are moved over the distances appropriate for this time step, followed by the calculation of a representative set of collisions. The time step is small compared to the mean collision time, and the results are independent of its actual value. DSMC will produce a hydrodynamic-like solution in the collisional fluid limit.

We have developed a gas kinetic dusty-gas DSMC comet model code. It is capable of being run for 1D-spherical, 2D axisymmetric and fully 3D problems. The photochemistry and photochemical kinetics for

the following parent molecules are now available: H2O, CO2, CO, CH3OH, H2CO, C2H6, C2H4, C2H2, HCN, NH3, and CH4. The model subsequently tracks the following dissociation products: H, H2, O, OH, C, CH, CH2, CH3, N, NH, NH2, C2, C2H, C2H5, CN, and HCO. Refractory dust particles are included also as many simulation particles where the particle density and size distribution can be specified.

The fundamental descriptions and various applications of our DSMC comet model are found the papers listed above.

7.5 K3D

Please see the following papers for details about the K3D nucleus model:

- Rosenberg, E.D. and D. Prialnik (2009), Fully 3-dimensional calculations of dust mantle formation for a model of Comet 67P/Churyumov-Gerasimenko, *Icarus*, *201*, 740-749, doi:10.1016/j.icarus.2009.01.028.

K3D is a fully 3-dimensional model used to study dust mantle formation. The model is essentially a thermal model, taking into account conduction, absorption of solar radiation, emission of thermal radiation, as well as heat released in crystallization of amorphous ice and absorbed in sublimation of crystalline ice. While we consider a porous structure and allow for internal sublimation, we do not compute the gas flow rate, but rather assume rapid transfer to the surface, which is acceptable for shallow depths, large porosities and steady-state conditions. We also consider dust drag, representing the dust grain size distribution by 5 discrete categories, differing in grain size. For each azimuth, depth and time, we compute the critical grain size above which grains can be dragged by gas and leave the nucleus. Those left behind form, eventually, an ice-depleted permanent dust mantle. The rate of growth of this mantle decreases considerably with repeated or bital revolutions; therefore, the results obtained here at the end of 5 orbits may be taken as representative.

The thickness of this mantle varies significantly over the nucleus surface. Depending on latitude and on thermal conductivity (but discarding as unphysical very high conductivities), the dust mantle may be as thin as 1 cm, and as thick as 10 cm. This range is significant because the former is smaller than the diurnal skin depth, while the latter exceeds it. This means that on the nucleus surface there will be areas that show diurnal variation of activity (rapid response to insolation), but also areas of more constant activity. The depth of unaltered material, marked here by the crystallization front, ranges between 1 m and several meters. The size distribution of dust grains changes along the orbit, be- ing steeper at large distances. In all cases, it is steeper than that assumed for the nucleus.