Title

MCLip

APPLICATION MANUAL

	Name and Function	Date	Signature
Prepared by	**C. THEROUDE**	24/09/2009	
Verified by	**U. POLVERINI**	24/09/2009	
Approved by			
Authorized by			
Application authorized by			

Document type	Nb WBS	Keywords

SUMMARY

This document is the application manual of the MCLip module. This module is dedicated to the computation of the gaseous flow at the thruster lip. It is integrated in the PLUMFLOW procedure.

Chapter 3 presents the MCLIP module, chapter 4 describes in detail the input and output of the module and chapter 5 gives an example of application.

Document controlled by

DOCUMENT CHANGE LOG

Issue/ Revision	Date	Modification Nb	Modified pages	Observations
01/00	19/08/09	1	All	Format, minor corrections
02/00	25/09/09	2	All	Add page 1

PAGE ISSUE RECORD

Issue of this document comprises the following pages at the issue shown

Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.
All	01/00										
All	02/00										

MCLip V3.2 – Application Manual

This Manual contains task-oriented instructions that show you how to use the MCLip module.

Document issue : 2.0

Software version : MCLip Version 3.2

The information contained in this document is subject to change without notice and should not be construed as a commitment by EADS ASTRIUM SAS. EADS ASTRIUM SAS assumes no responsibility for any errors that may appear in this document.

The software described is licensed for installation according to the following conditions:

- The software shall not be distributed to third parties,
- Modification of the software is not permitted.

All errors and suggestions concerning the software or this documentation should be sent to:

C. Theroude at:
EADS ASTRIUM
31 avenue des Cosmonautes
Z.I. du Palays
31402 TOULOUSE CEDEX 4
France

Page left intentionally blank

TABLE OF CONTENTS

1 REFERENCE DOCUMENTS

[RD1] *Spécification d'intégration : PLUMFLOW, PLUME, CONTAMINE dans ESABASE*. P. Chèoux-Damas, C. Theroude. Doc. Astrium: S413/RT/17.94. 12/01/95.

[RD2] *PLUME V3.2 - Interface files definition*. P. Chèoux-Damas. Doc. Astrium: S413/RT/41.97. 24/10/97.

2 INTRODUCTION

The objective of the MCLIP module is to compute the thruster flow field at the vicinity of the nozzle lip. In this region the flow passes rapidly from a continuum regime inside the nozzle to a free molecular regime outside. Moreover, the flow around the lip has a strong influence on the backflow and so on the contamination of surfaces located in this region.

The methods classically used to compute the continuum flow field (Navier-Stokes, method of characteristics) are not adapted to simulate the flow field in transition regime. That is why, a specific module dedicated to the computation of the nozzle lip flow field has been developed. It is based on the Direct Monte-Carlo Simulation (DSMC) method that is well adapted to model the transitional regimes.

The MCLIP module is integrated inside the PLUMFLOW software and is interfaced with the ODE, CEC, NAVIER and TRAJET modules.

This document is the application manual of the MCLIP software. Chapter 3 presents the MCLIP module, chapter 4 describes in detail the input and output of the module and chapter 5 gives an example of application and some advises to user.

3 MCLIP PRESENTATION

The Direct Simulation Monte-Carlo (DSMC) method originally developed by G.A. Bird is based on the simulation of the large number of molecules of a real gas by a "small number" of representative molecules. The typical number of representative molecules is a few tens thousands of molecules.

Although this method has no limitation, its classical application is the modelling of the transitional medium. It allows to model a lot of physical phenomena like exchanges between internal energy and kinetic energy, chemical reactions, surfaces interactions, etc.

The computation domain is divided into a set of cells and the time moves by discrete time steps. The displacement of molecules and the collisions are uncoupled.

A schematic presenting the Monte-Carlo method at the nozzle lip is presented at the Figure 4.1-1.

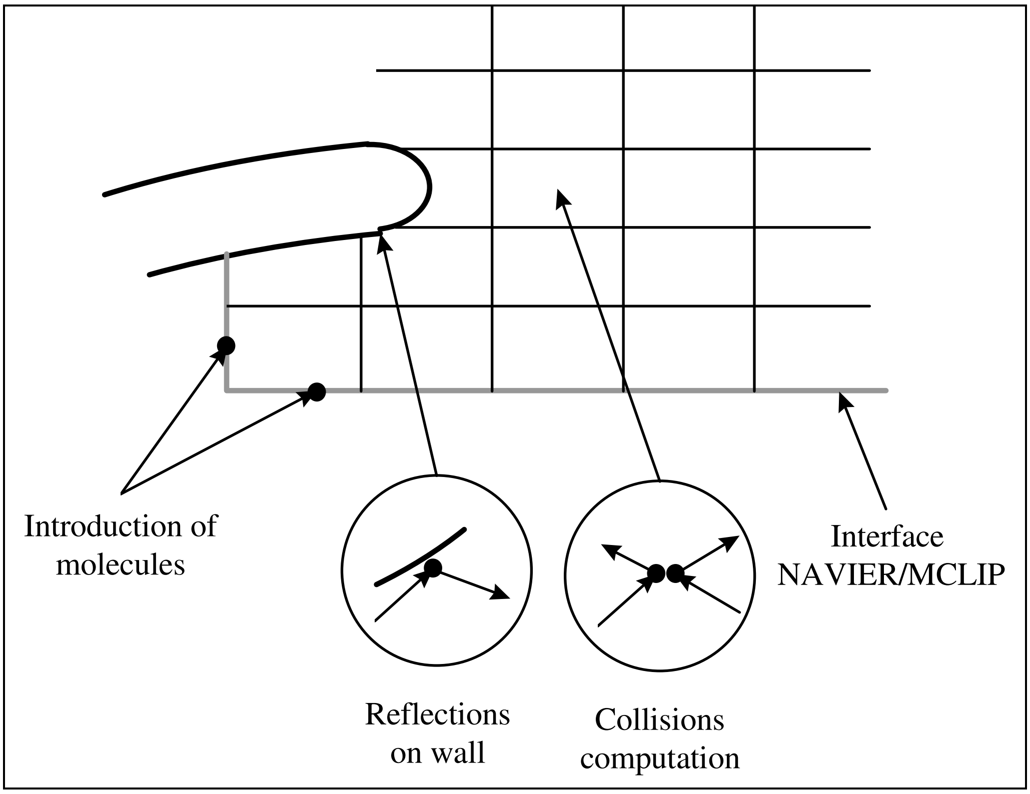


Figure 4.1-1: Monte-Carlo calculation at the nozzle lip

The process is iterative, and during each Δt time step, the following operations are repeated:

- Molecules are introduced at the boundaries of the computation domain. The number of introduced molecules depends on the velocity and the temperature. The initial characteristics of the molecules are randomly determined taking into account the distribution functions at the boundaries.
- The molecules are displaced by a distance depending on their velocity and on the computation time step. The possible reflections on the thruster walls are computed.

- A representative number of collisions, corresponding to Δt is computed among the molecules of each cell. The colliding molecules pairs are randomly sampled among all the molecules of a cell. The characteristics after collision (velocity, internal energy) are modified taking into account the cross sections.
- The significant properties of the flow (density, velocity, temperature, etc) are sampled in each cell.

4 MCLIP INPUT/OUTPUT

4.1 MCLIP ARCHITECTURE

The input files of the MCLIP software are the following:

- The file definition (PLUMFLOW.SYSINPUT),
- The run parameters file (.MLI),
- The flow field file (.FLOW) generated by the NAVIER module,
- The thermodynamic file (.T07) generated by ODE or CEC.

These files are presented at the Figure 4.1-1 and detailed in the paragraph 4.2.

All these files are formatted in ASCII to allow the portability on different operating systems (HP, SUN, ULTRIX, etc).

The output files of the MCLIP software are:

- The listing file (.MLO) resuming the run parameters and the progress of the run,
- The flow field file containing the characteristics of the flow calculated by MCLIP in the computation domain (.FLOW). This file can be visualised using the TRAJET module.
- The restart file (.REP) allowing to restart a simulation from the results of a previous calculation.
- The Ideas format file (.unv) to allow visualisation of the results using the ideas post-processor.

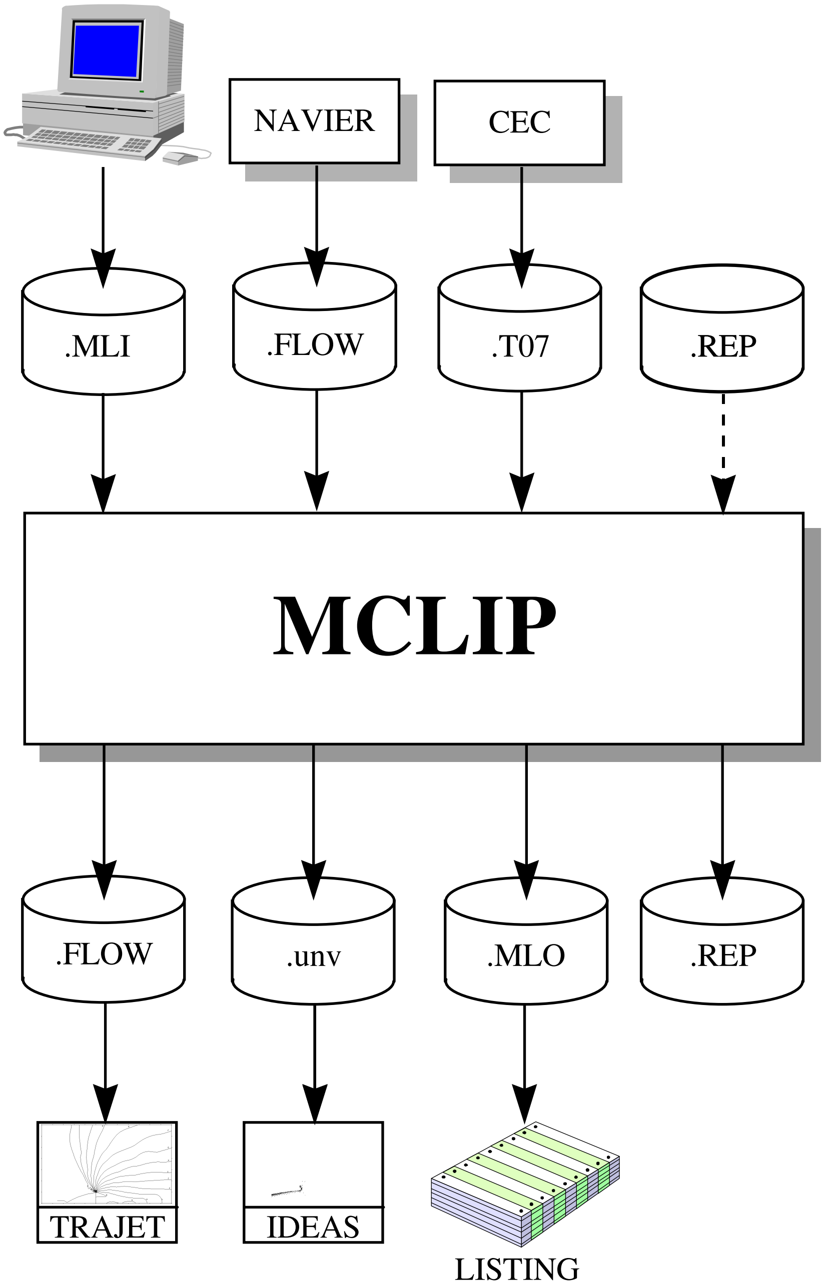


Figure 4.1-1: Architecture of the MCLIP module

4.2 MCLIP INPUT FILES DESCRIPTION

4.2.1 The definition file: PLUMFLOW.SYSINPUT

This file contains only the name of the thruster (e.g. NAME). This name allows MCLIP to open the different files (NAME.MLI, NAME.FLOW, etc) at the beginning of the MCLIP run. This file is created by the PLUMFLOW framework and has not to be created by the user excepted if he wants to run MCLIP outside the PLUMFLOW framework.

4.2.2 The run parameters file: .MLI

This file contains the run parameters of the MCLIP software. It is characterised by its extension (.MLI). The structure of the run parameters file is based on a set of namelist.

The first namelist, called $MESH is used to define the computation domain and its meshing.

The second namelist, called $RUNPAR is used to define the run control parameters of MCLIP.

The content of the file is described hereafter.

Format of the file:

```
$MESH
ZINFL      ZSUPL      RINFL      RSUPL
ZINFE      ZSUPE      RSUPE
LIPTYPE    THICKLIP
NBZ1       NBR1       NBZ2       NBR2       NBZ3       NBR3       NBR4
NBR5       NBZ6       NBR6       NBZ7       NBR7       NBZ8       NBR8
NBZ9       NBR9
QFZ1       QFR1       QFZ2       QFR2       QFZ3       QFR3       QFR4
QFR5       QFZ6       QFR6       QFZ7       QFR7       QFZ8       QFR8
QFZ9       QFR9
$END
$RUNPAR
NMOLEC
NITER      NITER0     REP        NREP       REPSTAT
ISEED      ETA
VISUMESH   VISUNV     NPRINT
COLLI      MUSU
TWINT      TWLIP      TWEXT
$END
```

Description

The mesh geometry and the main dimensions of the domain are presented at the Figure 4.2-2. The computation domain is split in two main areas. The first one concerns the zone close to the lip and is divided in six blocks. The second one concerns the zone far from the lip and is divided in three blocks. The mesh parameters of the nine blocks can be defined by the user. The block numbers are identified by hooped figures.

$MESH part

ZINFL: Minimum abscissa of the zone close to the lip (see Figure 4.2-2). This value is positive and assumes that the zero abscissa is at the thruster exit.

Type: real

Default: 0.5

Unit: Adimensioned by the thruster exit radius.

ZSUPL: Maximum abscissa of the zone close to the lip (see Figure 4.2-2). This value is positive and assumes that the zero abscissa is at the thruster exit.

Type: real

Default: 1.

Unit: Adimensioned by the thruster exit radius.

RINFL: Minimum radius of the zone close to the lip (see Figure 4.2-2). This value is positive and assumes that the zero abscissa is at the thruster exit.

Type: real

Default: 0.2

Unit: Adimensioned by the thruster exit radius.

RSUPL: Maximum radius of the zone close to the lip (see Figure 4.2-2). This value is positive and assumes that the zero abscissa is at the thruster exit.

Type: real

Default: 1.

Unit: Adimensioned by the thruster exit radius.

ZINFE: Minimum abscissa of the zone close far from the lip (see Figure 4.2-2 This value is positive and assumes that the zero abscissa is at the thruster exit.

Type: real

Default: 5.

Unit: Adimensioned by the thruster exit radius.

ZSUPE: Maximum abscissa of the zone far from the lip (see Figure 4.2-2). This value is positive and assumes that the zero abscissa is at the thruster exit.

Type: real

Default: 10.5

Unit: Adimensioned by the thruster exit radius.

RSUPE: Maximum radius of the zone far from the lip (see Figure 4.2-2). This value is positive and assumes that the zero abscissa is at the thruster exit.

Type: real

Default: 8.5

Unit: Adimensioned by the thruster exit radius.

LIPTYPE: Defines the shape of the lip (see Figure 4.2-1).

Range: 1: The lip has a circular shape.

2: The lip has a square shape.

Type: integer

Default: 1

THICKLIP: Defines the thickness of the lip (see Figure 4.2-1).

Type: real

Default: 0.2

Unit: Adimensioned by the thruster exit radius.

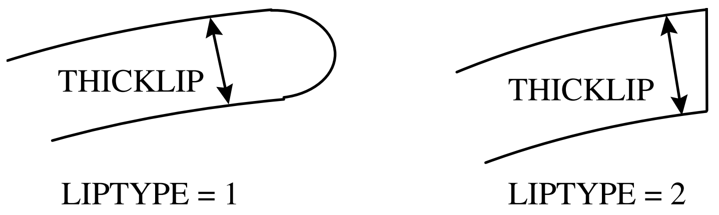


Figure 4.2-1 : Description of the lip

NBZ1: Number of mesh in the Z direction for the block 1.

Type: Integer

Default: 20

NBR1: Number of mesh in the R direction for the block 1.

Type: Integer

Default: 30

NBZ2: Number of mesh in the Z direction for the block 2.

Type: Integer

Default: 30

NBR2: Number of mesh in the R direction for the block 2.

Type: Integer

Default: 15

NBZ3: Number of mesh in the Z direction for the block 3.

Type: Integer

Default: 30

Remark: this parameter is used only if LIPTYPE = 2.

NBR3: Number of mesh in the R direction for the block 3.

Type: Integer

Default: 20

Remark: if LIPTYPE = 1, this parameter means the number of concentric mesh around the circular lip.

NBR4: Number of mesh in the R direction for the block 4.

Type: Integer

Default: 20

NBR5: Number of mesh in the R direction for the block 5.

Type: Integer

Default: 15

NBZ6: Number of mesh in the Z direction for the block 6.

Type: Integer

Default: 10

NBR6: Number of mesh in the R direction for the block 6.

Type: Integer

Default: 15

NBZ7: Number of mesh in the Z direction for the block 7.

Type: Integer

Default: 30

NBR7: Number of mesh in the R direction for the block 7.

Type: Integer

Default: 25

NBZ8: Number of mesh in the Z direction for the block 8.

Type: Integer

Default: 10

NBR8: Number of mesh in the R direction for the block 8.

Type: Integer

Default: 20

NBZ9: Number of mesh in the Z direction for the block 9.

Type: Integer

Default: 10

NBR9: Number of mesh in the R direction for the block 9.

Type: Integer

Default: 15

QFZ1: Geometrical progression of the mesh in the Z direction for the block 1.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 1.

QFR1: Geometrical progression of the mesh in the R direction for the block 1.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.95

QFZ2: Geometrical progression of the mesh in the Z direction for the block 2.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.95

QFR2: Geometrical progression of the mesh in the R direction for the block 1.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.95

QFZ3: Geometrical progression of the mesh in the Z direction for the block 3.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 1.

QFR3: Geometrical progression of the mesh in the R direction for the block 3.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.9

QFR4: Geometrical progression of the mesh in the R direction for the block 4.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 1.

QFR5: Geometrical progression of the mesh in the R direction for the block 5.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.95

QFZ6: Geometrical progression of the mesh in the Z direction for the block 6.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.9

QFR6: Geometrical progression of the mesh in the R direction for the block 6.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.9

QFZ7: Geometrical progression of the mesh in the Z direction for the block 7.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.94

QFR7: Geometrical progression of the mesh in the R direction for the block 7.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.94

QFZ8: Geometrical progression of the mesh in the Z direction for the block 8.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.9

QFR8: Geometrical progression of the mesh in the R direction for the block 8.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.94

QFZ9: Geometrical progression of the mesh in the Z direction for the block 9.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.95

QFR9: Geometrical progression of the mesh in the R direction for the block 9.

Range: ≥ 0. If it is less than 1, the progression is such that, the farther from the thruster lip, the larger the meshes.

Type: Real

Default: 0.95

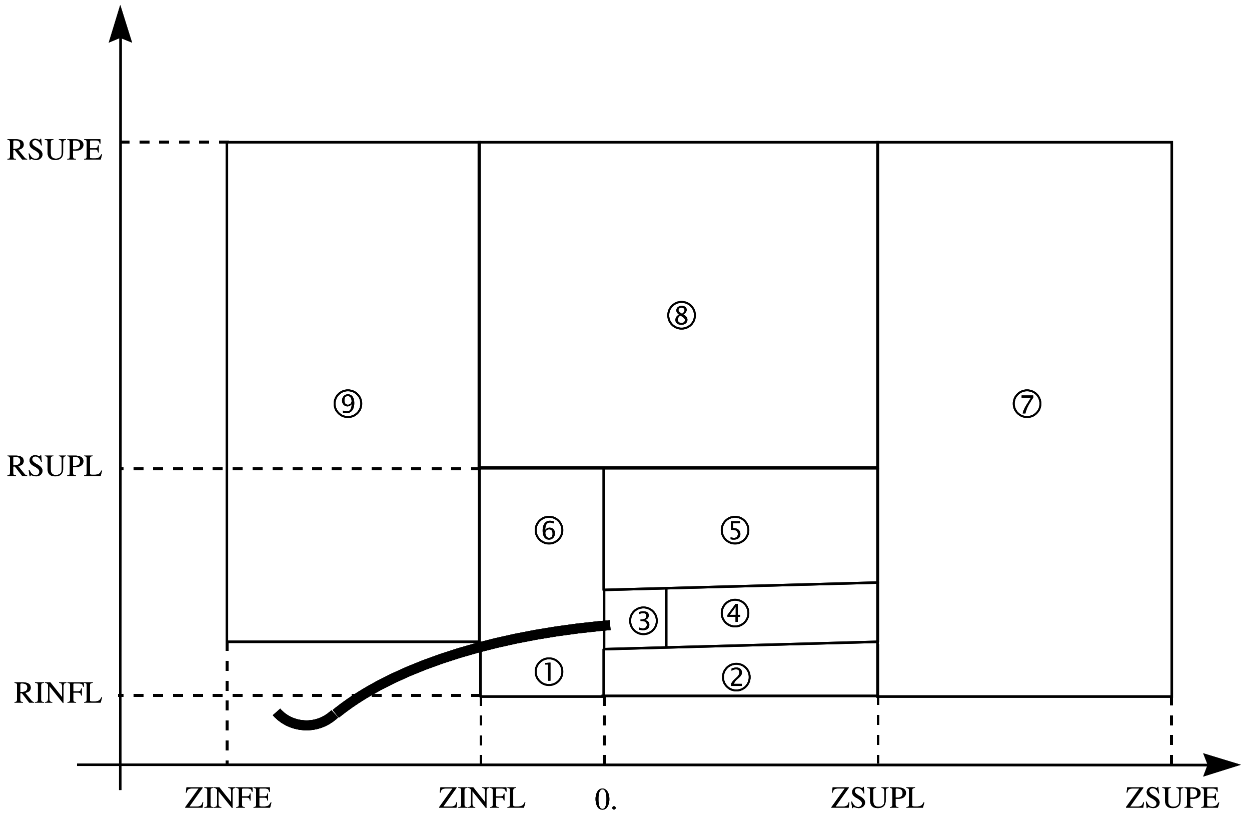


Figure 4.2-2: Description of the mesh geometry

$RUNPAR part

NMOLEC: Target number of molecules in the computation domain.

Type: integer

Range: ≥ 0

Default: 100

NITER: Total number of iterations to be done.

Type: integer

Default: 500

NITER0: Number if iterations to be done by MCLIP without sampling the flow parameters.

Type: integer

Default: 0

REP: Parameter used to specify whether the user wants to continue or not a previous run.

Type: integer

Range: 0: The current run starts from scratch.

1: The current run starts from a previous run using the .REP file as initial conditions.

Default: 0

NREP: Number of iterations between two writings of the restart file.

Type: integer

Default: 100

REPSTAT: Parameter used to specify whether the user wants to continue or not a previous run using the sampling results cumulated during the previous run. This parameter is taken into account only if REP = 1.

Type: integer

Range: 0: The current run takes into account the sampling cumulated during previous runs.

1: The current run starts a new sampling without taking into account previous sampling.

Default: 0

ISEED: Initial random seed value used and propagated by the random number generator. A variation of its initial value modifies the sequence of random number.

Type: integer

Default: 4764

ETA: Multiplication coefficient of the time step during a MCLIP computation. The reference time step is automatically computed from the characteristics of the flow.

Type: real

Default: 1.

VISUMESH: Parameter to specify whether the user wants to perform only the generation of the mesh or not. If VISUMESH equal 1, MCLIP generates only a .unv file for mesh visualisation.

Type: integer

Range: 0: MCLIP performs a calculation of molecules transport.

1: MCLIP generates a .unv file without making any calculations.

Default: 0

VISUNV: Parameter to specify whether the user wants to generate a .unv file containing the results of the calculation.

Type: integer

Range: 0: MCLIP do not generate a .unv file.

1: MCLIP generates a .unv file containing the results.

Default: 0

NPRINT: Number of iterations between two writings of the run characteristics in the .MLO file.

Type: integer

Default: 10

COLLI: Parameter to specify which type of collisions computation, the user wants to use.

Type: integer

Range: 1: MCLIP evaluates the maximum of CXS*VR at each iteration. It is the more accurate but the less efficient method.

2: MCLIP evaluates the maximum of CXS*VR every 50 iterations.

3: MCLIP uses the No Time Counter procedure. It is the less accurate but the more efficient method.

Default: 1

MUSU: Parameter to specify if the computation is performed with or without the multiplication / suppression procedure in the computation domain.

Type: integer

Range: 0: No multiplication / suppression is performed.

1: Multiplication / suppression is performed.

Default: 1

TWINT: Parameter to specify the temperature of the internal part of the thruster wall.

Type: real

Range: ≥ 0

Unit: Kelvin

Default: 600 K

TWLIP: Parameter to specify the temperature of the lip part of the thruster wall.

Type: real

Range: ≥ 0

Unit: Kelvin

Default: 600 K

TWEXT: Parameter to specify the temperature of the external part of the thruster wall.

Type: real

Range: ≥ 0

Unit: Kelvin

Default: 600 K

4.2.3 The flow field file: .FLOW

This file contains the characteristics (density, temperature, velocity) of the flow field inside and in the vicinity of the thruster. It is used to initialise the Monte-Carlo calculation at the limit of the computation domain. This file shall be created by the NAVIER module. The complete description of the .FLOW file is presented in the RD2.

4.2.4 The thermodynamic file: .T07

This file contains the thermodynamic characteristics (mass, molar fraction, cross section, number of degrees of freedom) of each group of species. It can be created the CEC module or by the ODE module (nevertheless, the normal way is to use the CEC module to control the contents of the .T07 file). The number of group of species defined in the .T07 shall not exceed 10. The complete description of the .T07 file is presented in the RD2.

4.2.5 The restart file: .REP

This contains all the information necessary to restart a run. That is to say, the characteristics of the molecules included in the computation domain and the characteristics of the flow already cumulated during the previous run. This file is created by the MCLIP module at the end of each run and is read by MCLIP if the user wants to perform a restart (REP = 1). The characteristics of the flow are used if the user wants to restart the statistics (REPSTAT = 1). This file is a binary file and can be relatively large (some tens of kbytes).

4.3 THE MC3D OUTPUT FILES

4.3.1 The listing file: .MLO

This file is a listing file containing a summary of the input parameters and some information concerning the progress of the run.

An example of such a file is given hereafter. It contains:

- The description of the run parameters,

- A summary of the .T07 file,
- A summary of the .FLOW file,
- The main characteristics of the mesh,
- Every NPRINT iteration, the theoretical and simulated upstream mass flow rate, the theoretical and simulated lateral mass flow rate, the total mass flow rate and the ratio between the input and output mass flow rate,
- At the end of the calculation, the theoretical and simulated upstream mass flow rate, the theoretical and simulated lateral mass flow rate, the total mass flow rate and the ratio between the input and output mass flow rate.

```
RUN DATA :


          COMMAND FILE   :                    mbb10_new.MLI
          THRUSTER FILE  :                    mbb10_new.FLOW
          SPECIES FILE   :                    mbb10_new.T07

     RUN PARAMETERS DESCRIPTION :

     Total number of iterations                  :   10
     Number of iterations before integration     :    0
     Frequency of creation of the Restart file   :   50
     Restart calculation                         :    1
     Number of molecules                         : 3000
     Collision model                             :    1
     Multiplication / suppression of molecules :    1
     Time factor in the first block              :0.300E+00
     Random seed                                 : 4747
     Restart of the statistics                   :    1

.T07 FILE DESCRIPTION :

     Reference temperature :  300.0000000

     Number of groups :   1

     Group name    Molar fraction    Molar mass    Diameter       Degrees of freedom
         MOY       .1000000E+01     .3394118E-25  .4045486E-09   2.3729661

     FLOW-FIELD FILE HEADER :   CALCUL NAVIER-STOKES DE LA TUYERE MBB10

          THROAT RADIUS      = 0.1425000E-02 M
          THRUST             = 0.9722190E+01 N
          MASS FLOW RATE     = 0.3176800E-02 KG/S
          EXIT RADIUS        = 0.9473700E+01 R*
          THROAT/EXIT LENGTH = 0.2667000E+02 R*
          IDEAL GAS                                      NGAS =    3

          NUMBER OF POINTS BY STREAMLINE                  NPT = 145
          NUMBER OF STREAMLINES                           NSL =  49
          NUMBER OF VARIABLES OF THE FLOW-FIELD FILE      NVP =  10

     MESH CHARACTERISTICS

     NUMBER OF BLOCKS :   9

     BLOCK      NUMBER OF CELLS          TIME STEP
       1           1050               .2396646E-07
       2            300               .2508981E-07
       3            800               .2891570E-08
```

```
   4           480                 .3567296E-07
   5           450                 .6214461E-07
   6           150                 .6205887E-07
   7           750                 .2499293E-06
   8           200                 .3133571E-06
   9           150                 .3528315E-06

 TOTAL NUMBER OF CELLS :   4330

 THE PROGRAM IS READING THE RESTART FILE

 Number of molecules read in the restart file :   50145

 SUMMARY AT THE ITERATION :     5
 ------------------------------

 Number of molecules :  50291

 Upstream mass flow rate
 Theoretical : .6404438E-03
 Input : .6675566E-03 Output : .2393633E-04 Net : .6436203E-03

 Lateral mass flow rate
 Theoretical : .3093770E-02
 Input : .4186990E-02 Output : .5354359E-03 Net : .3651554E-02

 Total mass flow rate
 Input : .4854546E-02 Output : .3393432E-02 Ratio : .6990213E+00

 SUMMARY AT THE END OF CALCULATION :
 -----------------------------------

 Number of molecules :  50426

 Upstream mass flow rate
 Theoretical : .6404438E-03
 Input : .6658757E-03 Output : .2008899E-04 Net : .6457867E-03

 Lateral mass flow rate
 Theoretical : .3093770E-02
 Input : .3725448E-02 Output : .4341648E-03 Net : .3291284E-02

 Total mass flow rate
 Input : .4391324E-02 Output : .3465633E-02 Ratio : .7892000E+00
```

Extract of the .MLO file

4.3.2 The restart file: .REP

See § 4.2.5.

4.3.3 The flow field file: The .FLOW file

This file contains the characteristics (density, temperature, velocity) of the flow field inside and in the vicinity of the thruster computed by MCLIP. It is interpolated on an adapted mesh from the results computed on the real mesh. It allows to perform the visualisation of the flow field using the TRAJET module.

This file zaps the .FLOW file created by the NAVIER module and used as input of the MCLIP module.

4.3.4 The .unv file

This file contains the characteristics (density, temperature, velocity) of the flow field inside and in the vicinity of the thruster computed by MCLIP. It is calculated on the mesh used to perform the MCLIP calculation. It allows to perform the visualisation of the flow field using adapted software like Ideas.

5 HOW TO USE MCLIP

The goal of this chapter is to present the use of the MCLIP module on a real case and to give to the user some advises.

5.1 APPLICATION CASE

In this paragraph, a complete case of MCLIP application is presented.

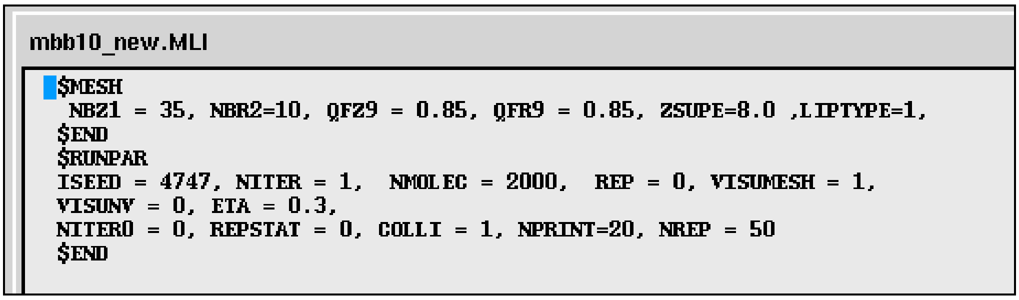
5.1.1 External input file

To perform this application case, the user shall have in the current directory:

- A mbb10.FLOW file computed by the NAVIER module,
- A mbb10.T07 file computed by CEC. An example of the generation of the .T07 file is given in the CEC application manual.

5.1.2 Generation of the mesh

First of all, the user shall generate the .MLI file containing the mesh and the run parameters. To do that, the user can edit the .MLI file using the PLUMFLOW interface. In order to edit the input file, the user has to click on ***Edit input file*** and then on ***MCLIP***. Using the editor, the user can enter the following file.



mbb10_new.MLI

```
$MESH
  NBZ1 = 35, NBR2=10, QFZ9 = 0.85, QFR9 = 0.85, ZSUPE=8.0 ,LIPTYPE=1,
 $END
 $RUNPAR
 ISEED = 4747, NITER = 1,  NMOLEC = 2000,  REP = 0, VISUMESH = 1,
 VISUNV = 0, ETA = 0.3,
 NITER0 = 0, REPSTAT = 0, COLLI = 1, NPRINT=20, NREP = 50
 $END
```

As described in the § 4.2.2, the ZSUPE defines the maximum distance of the domain from the thruster exit, the LIPTYPE defines the circular shape of the lip and the NBZ1, NBR2, QFZ9 and QFR9 define the mesh of the blocks 1, 2 and 9. The other mesh parameters are the default parameters. The $RUNPAR section will be described below except for the VISUMESH parameter defining that a .unv file will be generated without performing any calculation.

To perform the calculation, the user has to select ***MCLIP*** and then on ***OK***. The MCLIP program is then executed and after few minutes the user can observe that a mbb10_new.MLO and a mbb10_new.unv files have been created.

After completion of the run the user can edit the .MLO file using the PLUMFLOW integrated editor. To do that the user has to click on ***Edit input file*** and then on ***MCLIP.*** The informations included in the

.MLO files allows to verify that the number of species groups is three and the total number of cells is 4330.

The .unv file can be used to visualise the mesh using as example the Ideas tool. The generated mesh of the above case is presented at the Figure 5.1-1.

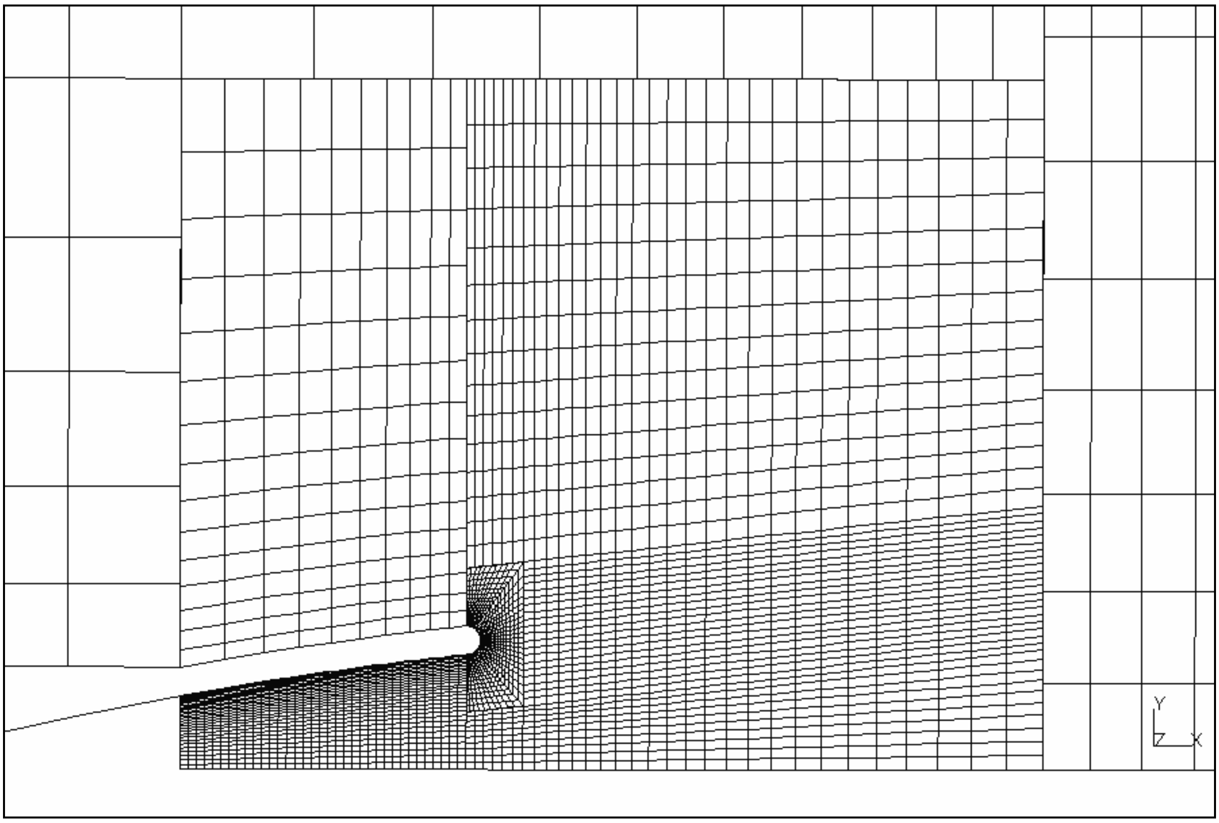
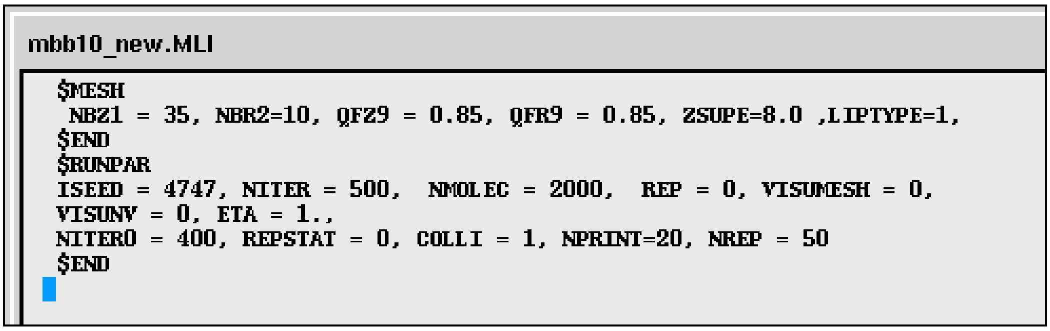


Figure 5.1-1: Mesh generated by MCLIP

5.1.3 Preliminary run execution

Once the mesh has been generated, the user has to define the run parameters in order to start the MCLIP computation. The .MLI file is presented below.



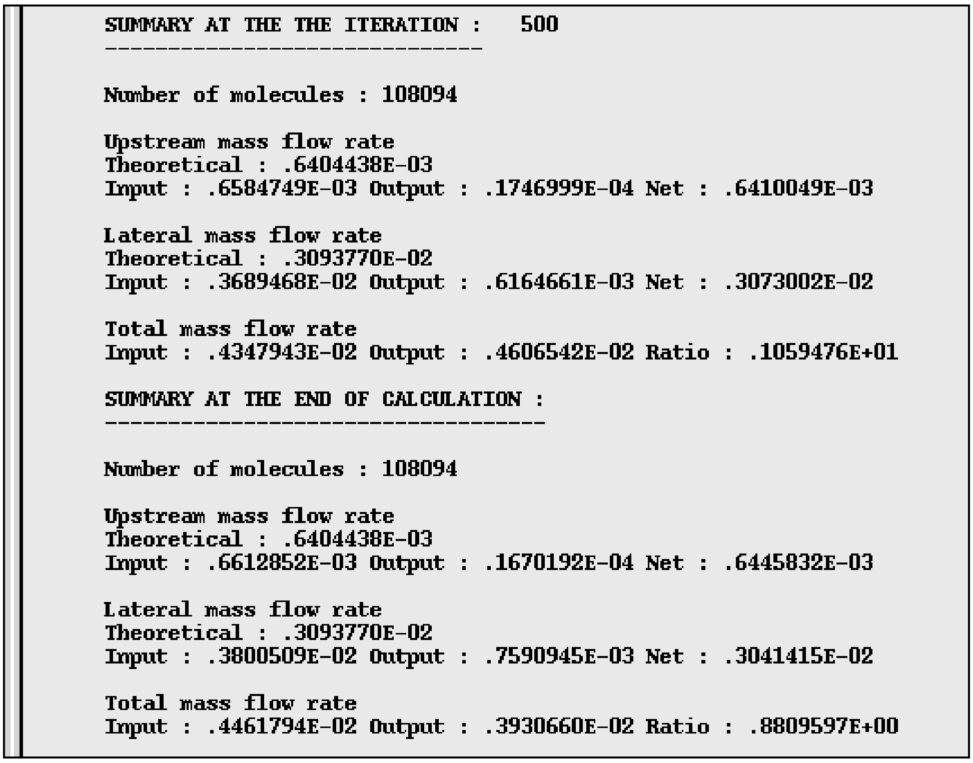
mbb10_new.MLI

```
 $MESH
  NBZ1 = 35, NBR2=10, QFZ9 = 0.85, QFR9 = 0.85, ZSUPE=8.0 ,LIPTYPE=1,
 $END
 $RUNPAR
 ISEED = 4747, NITER = 500,  NMOLEC = 2000,  REP = 0, VISUMESH = 0,
 VISUNV = 0, ETA = 1.,
 NITERO = 400, REPSTAT = 0, COLLI = 1, NPRINT=20, NREP = 50
 $END
```

The mesh parameters are the same as in the §5.1.2. On the other hand, the run parameters have been modified in order to start the flow field computation. The objective of this preliminary run is to fill the computation domain before starting the sampling of the results.

Thus, the time step is equal to 1 that is too large from the accuracy point of view but allows quickly filling the computation domain. The total number of iterations (NITER) is equal to 500 that should be sufficient to completely fill the mesh and the number of iterations (NITER0) without sampling is equal to 400, that means the results are sampled during the last 100 iterations. The target number of molecules is equal to 200 that leads to around 110 000 molecules in the domain that is compliant with the objective to have around ten molecules by cell and by species.

After completion of the run the user can edit the .MLO file using the PLUMFLOW integrated editor. An extract of the .MLO file is given below.



```
SUMMARY AT THE THE ITERATION :    500
------------------------------

Number of molecules : 108094

Upstream mass flow rate
Theoretical : .6404438E-03
Input : .6584749E-03 Output : .1746999E-04 Net : .6410049E-03

Lateral mass flow rate
Theoretical : .3093770E-02
Input : .3689468E-02 Output : .6164661E-03 Net : .3073002E-02

Total mass flow rate
Input : .4347943E-02 Output : .4606542E-02 Ratio : .1059476E+01

SUMMARY AT THE END OF CALCULATION :
-----------------------------------

Number of molecules : 108094

Upstream mass flow rate
Theoretical : .6404438E-03
Input : .6612852E-03 Output : .1670192E-04 Net : .6445832E-03

Lateral mass flow rate
Theoretical : .3093770E-02
Input : .3800509E-02 Output : .7590945E-03 Net : .3041415E-02

Total mass flow rate
Input : .4461794E-02 Output : .3930660E-02 Ratio : .8809597E+00
```

At the end of the .MLO file, the summary at the end of calculation gives the total number of molecules, comparisons between the theoretical and the computed flow rate, and the ratio between the input and output flow rate. The mass flow ratio is equal to 0.88 proving that the convergence was not completely achieved during the last 100 iterations.

5.1.4 Second run execution

A new calculation is then necessary. To do that, the results of the previous calculation are used as initial condition (so REP = 1), the time step is equal to 0.3 in order to improve the quality of the collisions computation, the NITER parameter is equal to 200 and the NITER0 parameter equal to 100. A new run is executed. After completion, the user can edit the .MLO file and note that the ratio of flow rate is equal to 1.03 and so that the computation has reached a stationary state.

The results can be easily visualised using the TRAJET module. To do that, the user has to select ***Isocontours plot*** in the PLUFMLOW main menu. The TRAJET module allows to visualise the iso-density contours calculated by MCLIP. The results are presented at the Figure 5.1-2.

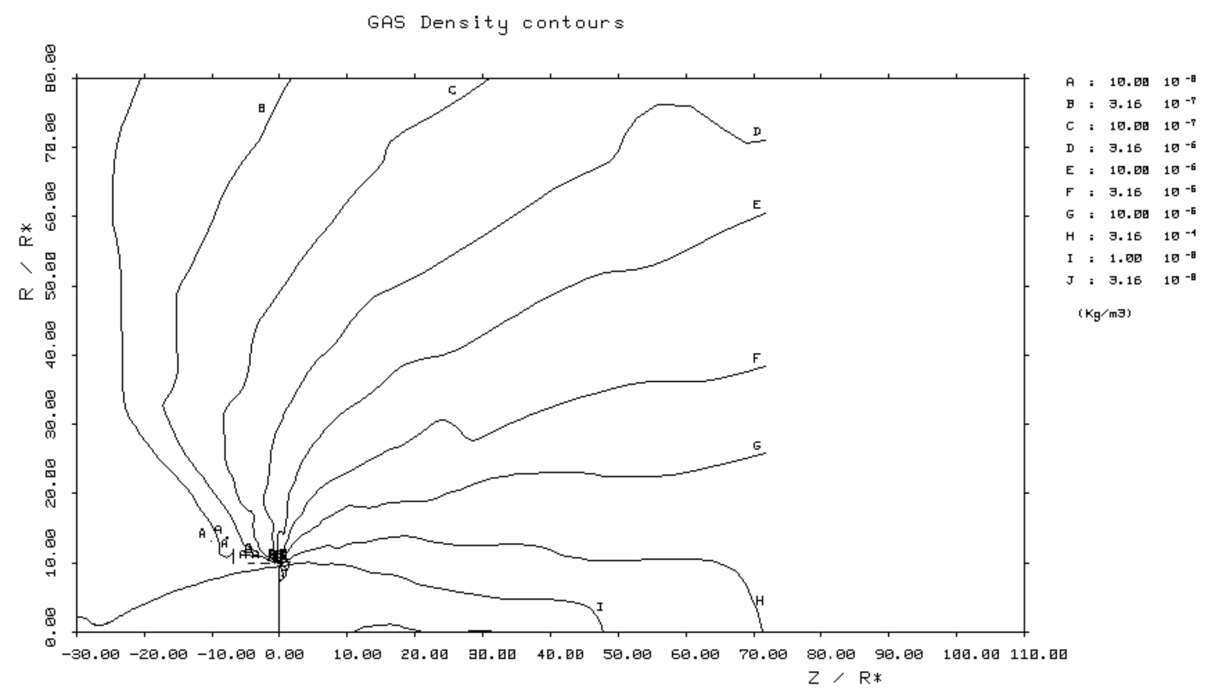


Figure 5.1-2: Density after the second run execution

On the above figure, on can remark that, the results are still slightly fluctuated in the DSMC region.

5.1.5 Accuracy improvement run

The accuracy of the results can be improved by performing a new run. In this case, it is interesting to use the sampling results cumulated during the previous run. So REPSTAT is equal to 1. NITER is taken equal to 300 and ITER0 to 0. So, the results will be evaluated from the last 400 iterations (the 100 iterations of the previous run and the 300 of the current one). The density visualised by the TRAJET module is presented below.

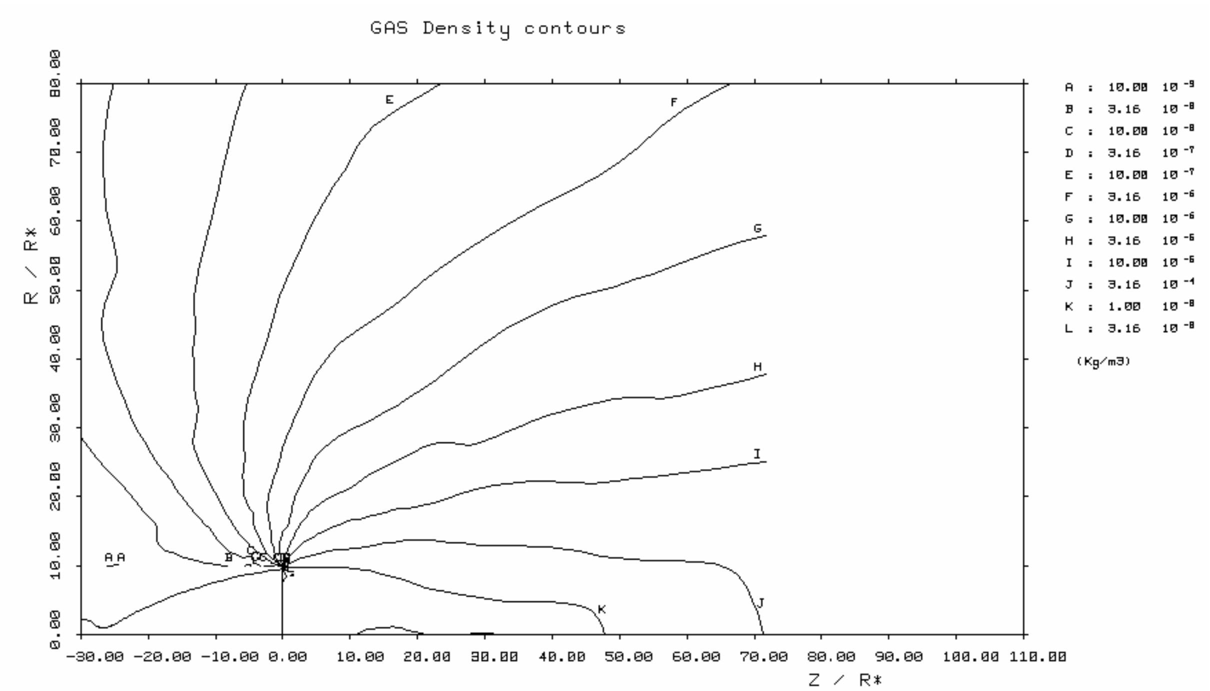


Figure 5.1-3: Density after the third run

On can remark, on the above figure, that the obtained results are more regular than in the previous run. The obtained .FLOW file can be used to perform further analysis.

5.2 ADVISES TO THE USER

5.2.1 On the meshing

The default parameters are generally adapted to mesh properly the computation domain. Nevertheless, the user can extend the mesh dimensions or refine certain regions. To properly model the flow field, the user shall verify that the mean free path of the flow is greater than the mesh dimension.

Moreover, the mesh dimensions shall be compatible with the initial Navier-Stokes calculation. Indeed the initial conditions of the flow at the limits of the domains are interpolated from the results of the NAVIER module. In particular, the internal limit included between ZINFL and ZSUPE shall be completely included in the Navier-Stokes mesh. Otherwise, the run cannot be executed.

5.2.2 On the number of molecules

To perform an accurate calculation of the collisions between molecules, it is currently recommended to have around ten molecules per cell and per species. The number of molecules is controlled using the NMOLEC parameter. The total number of molecules inside the computation domain is proportional to this parameter. If it appears that the total number of molecules is not sufficient to meet the target of ten molecules per cell, the NMOLEC parameter can be multiplied by the appropriate factor.

5.2.3 On the flow field file

As stated above, the initial conditions of flow are computed from the .FLOW initial file. Thus, this file shall contain the characteristics of the flow inside the nozzle. So, the .FLOW file shall be imperatively created by the NAVIER module and shall not be prolonged by the MATFLOW module.

The .FLOW file created by the MCLIP module zaps the already existing file created by NAVIER. So, if the user wants to perform a restart, he has to replace the MCLIP .FLOW file by the initial NAVIER .FLOW file.

5.2.4 On the collision model

Three collision models are available in the MCLIP module. They can be selected using the COLLI parameter.

- COLLI = 1: The model is a time counter method. The maximum of the relative velocity is evaluated at each time step. It is the most accurate method.
- COLLI = 2: The model is a time counter method. The maximum of the relative velocity is every 50 time steps.
- COLLI = 3: The model is a No Time Counter (NTC) method. In terms of CPU time, it is the most efficient of the three methods.

The methods 2 and 3 can be used to perform the filling of the mesh, however the method 1 is recommended during the sampling of the results.

5.2.5 On the multiplication / suppression

Molecules are multiplied / suppressed when they are changing of block. This method has been implemented in order to increase the accuracy of the results in the low density regions (backflow). So, it is recommended to perform the calculations with MUSU = 1.

APPENDIX A - THEORETICAL ASPECTS

A-1 - INTRODUCTION

This appendix presents the main theoretical aspects of the Direct Simulation Monte-Carlo method. The general principles have been presented in the chapter 3.

Let us recall the basic ideas. The displacement of molecules and the collisions are uncoupled. The process is iterative, and during each Δt time step, the following sequence is repeated:

- Molecules are introduced at the boundaries of the computation domain.
- The molecules are displaced by a distance depending on their velocity and on the computation time step. The possible reflections on the thruster walls are computed.
- A representative number of collisions, corresponding to Δt is computed among the molecules of each cell. The characteristics after collision (velocity, internal energy) are modified taking into account the cross sections.
- The significant properties of the flow (density, velocity, temperature, etc) are sampled in each cell.

A-2 - MESHING

The meshing is one of the most important aspects of a DSMC code. Indeed, the computation time depends strongly on the number of cells used for the calculation (to obtain a good statistic during the calculation of collisions it is necessary to have at least 10 molecules per cell). In addition, in order to obtain satisfactory results, it is recommended that the gradient of the flow properties varies not much in one cell.

The computation domain being axisymetric, the mesh has a cylindrical symmetry around the thruster axis. An example of the MCLIP meshing is given at the Figure 5.1-1.

A-3 - STATISTICAL WEIGHT OF MOLECULES

The statistical weight ω of one computation molecule refers to the number of real molecules simulated by this molecule.

The results are obtained from the calculation of statistical means in each cell. So, in order to limit the statistical fluctuations, it is to be hoped that the number of molecules is large enough (this number is usually around 10). In order to cope with these requirements, the statistical weight of the molecules may vary in time and in space.

In that way, it could be efficient to introduce more molecules in some regions of the mesh. In order to introduce the real flow rate, a different statistical weight is attributed to each molecule.

Moreover, in order to control the number of molecules per cell (to maintain a minimum number of molecules in each cell or to increase the accuracy in some regions of the mesh) molecules can be multiplied or suppressed when they change of cell.

To do that, a weighting factor (*Wf*) is attributed to each cell. When a molecule passes from a cell where the weighting factor if Wf_i to a cell where it is equal to Wf_f, the weight of the molecule is multiplied by Wf_f/Wf_i. In order not to distort the results, a multiplication/suppression method is used. Each molecule is transformed, on average, into Wf_i/Wf_f molecules. So, the number of real molecules remains, in average, unchanged.

This method has been applied in the MCLIP software. Indeed, different weighting factors have been attributed to the different blocks. In particular, a high weighting factor has been attributed to the backflow region where the number of molecules is low.

A-4 - MOLECULES INTRODUCTION

The number of molecules to introduce through one boundary of the computation domain is given by the formula:

$$N = \frac{S.D_{amb}.\Delta t}{2\pi^{1/2}\beta}\exp\left(-s_n^2\right) + \pi^{1/2}.s_n\left(1 + erf\left(s_n\right)\right)$$

Where β is the reciprocal of the most probable molecular thermal velocity:

$$\beta = \left(\frac{m}{2kT}\right)^{-1/2}$$

Where $s_n = \beta.V_{amb}.cos\theta$ is the normal component of the product $\beta.V_{amb}$, with θ equal to the angle between the velocity and the normal.

Where S is the area of the surface, D_{amb} is the density of the input flow and Δt the time step.

The velocity of an introduced molecule is given by the following formula:

$$\vec{V}_{intro} = \vec{V}_{amb} + \vec{U}_{n,th} + \vec{V}_{th} + \vec{W}_{th}$$

The velocities V_{th} and W_{th} are determined from the following formula:

$$f_{V_{th}W_{th}} = \frac{\beta^2}{\pi}\exp\left(-\beta^2\left(V_{th}^2 + W_{th}^2\right)\right)$$

The normal component of the velocity $U_{n,th}$ is determined from the following formula:

$$f_{U_{n,th}} = \frac{1}{\pi^{1/2}}\left(\beta.U_{n,th} + \beta.V_{n,amb}\right)\exp\left(-\beta^2.U_{n,th}^2\right)$$

A-5 - MOLECULES MOTION

If the motion in 3D geometry is obvious, this is not the case in axisymetric geometry. Indeed, the molecules characteristics are its coordinates x and r and the three components of the velocity v_x, v_r, v_θ. The method of computation is the following:

- Computation of the new coordinates:

$$x' = x + v_x.\Delta t$$
$$r' = \left(\left(r + v_r.\Delta t\right)^2 + \left(v_\theta.\Delta t\right)^2\right)^{1/2}$$

- Computation of the new radial and tangential velocities :

$$v_r' = \frac{\left(v_r\left(r+v_r\Delta t\right)+\left(v_\theta \Delta t\right)\right)}{r'}$$
$$v_\theta' = \frac{\left(v_\theta\left(r+v_r\Delta t\right)+\left(v_r v_\theta \Delta t\right)\right)}{r'}$$

A-6 - REFLECTIONS ON WALLS

The used method allows to model the reflections on the thruster walls or on any others surfaces.

The model contains a fraction of diffuse reflection and a fraction of specular reflection. It allows to handle the specular reflection, the diffuse reflection and the intermediate cases.

- Specular reflection: it is a perfectly elastic process. The normal component of the velocity is reversed while the tangential components remain unchanged.
- Diffuse reflection: in this case the velocity of the molecule after reflection is independent of the initial velocity. Nevertheless, the reflected molecules follow a Maxwell distribution for a wall temperature equal to T_w.

  So, the distribution function of the normal components is:

  $$f_{u_n} = u_n \exp\left(-\beta^2 u_n^2\right)$$

  The distribution function of the tangential components is maxwellian:

  $$f_{v,w} = \exp\left(-\beta^2\left(v^2+w^2\right)\right)$$

Let σ the proportionality coefficient between specular reflection and diffuse reflection. The method of calculation is the following:

- The software determines if the molecule impinges a surface.
- A random number is generated. If it is above σ the molecule is specularly reflected, if it is less than σ the molecule is diffusely reflected.
- The molecule is displaced as function of the remaining time and its velocity after reflection.

A-7 - CALCULATION OF COLLISIONS

The computation of collisions is the crucial point of the Monte-Carlo calculation. The kinetic and internal energies are modified during the collisions.

The computation of an appropriate number of collisions N_c during Δt is very important for the simulation. In principle, it is necessary to evaluate the collision frequency. Nevertheless, the evaluation of the collision frequency if very expensive in terms of CPU time (it is proportional to the square of the number of molecules in the cell). So, more efficient approaches are usually used like time counter or no time counter methods.

In addition, the collisions computation requires the knowledge of the collision cross section that allows to evaluate the probability of a shock between two molecules. The model commonly used is the VHS model (variable hard sphere) that allows to simulate the variation of the viscosity with the temperature.

The molecules are generally poly-atomic. At the usual temperatures of simulation, the involved energy during a collision is such that only the rotational energy has to be taken into account. This process is simulated using the Larsen-Borgnakke model.

Time counter method

The time counter method developed by Bird can be used to simulate the collisions. But it is known that this method can generate significant fluctuations if the number of computed collisions is too small. So, this method has been improved by W.L. Hermina in order to compute a minimum number of collisions in each cell and at each time step. This method has been implemented has is described below.

The collision frequency within a cell is given by the following formula:

$$\nu = \frac{1}{2V}\sum_{j=1}^{N}\sum_{j=1}^{N}\omega_i\omega_j\sigma_{ij}V_{r,ij} \quad (1)$$

Where V is the volume of the cell, N the total of molecules in the cell, ω_i the statistical weight of the molecule i, $V_{r,ij}$ the relative velocity of the pair ij and s_{ij} the cross section of the pair ij.

By noting n the total number of real molecules within a cell:

$$n = \sum_{j=1}^{N}\omega_i \quad (2)$$

We can rewrite the formula (1):

$$\nu = \frac{n^2\sum_{i=1}^{N}\sum_{j=1}^{N}\omega_i\omega_j V_{r,ij}}{2V\sum_{i=1}^{N}\sum_{j=1}^{N}\omega_i\omega_j} \quad (3)$$

The summation over all collision pairs can now be replaced by a summation over a sampling index k, where for each sample k two collision partners are chosen at random within the cell. It should be noted that in the limit of large k, the formula remains true.

$$\nu \approx \frac{n^2\sum_k \omega_{i(k)}\omega_{j(k)}\sigma_{i(k)}V_{r,i(k)j(k)}}{2V\sum_k \omega_{i(k)}\omega_{j(k)}} \quad (4)$$

We can now choose the collisions number increment to be proportional to the summand in the numerator and the time increment to be proportional to the summand in the denominator.

$$\Delta n_{c,k} = \beta_k\omega_{i(k)}\omega_{j(k)}\sigma_{i(k)j(k)}V_{r,i(k)j(k)} \quad (5)$$

$$\Delta t_{c,k} = \beta_k\frac{2V}{n^2}\omega_{i(k)}\omega_{j(k)} \quad (6)$$

Equation (4) can be then recovered as the sum of the collisions number increment divided by the sum of the time increment:

$$v=\frac{\sum_{k}\Delta n_{c,k}}{\sum_{k}\Delta t_{c,k}}=\frac{n^{2}\bar{\beta}\sum_{k}\omega_{i(k)}\omega_{j(k)}\sigma_{i(k)j(k)}V_{r,i(k)j(k)}}{2V\bar{\beta}\sum_{k}\omega_{i(k)}\omega_{j(k)}} \quad (7)$$

A maximum value is imposed on the collisions number increment, by the constraint that it cannot exceed the highest statistical weight of the collision partners. Consequently, β_k is defined by the following equation where a_k is between 0 and 1.

$$\beta_k=\frac{a_k}{\left(\omega_i\sigma_{ij}V_{r,ij}\right)_{\max}} \quad (8)$$

The resulting expression for the equations (5) and (6) are:

$$\Delta n_{c,k}=\frac{a_k\sigma_{ij}V_{r,ij}\omega_i\omega_j}{\left(\omega_i\sigma_{ij}V_{r,ij}\right)_{\max}} \quad (9)$$

$$\Delta t_{c,k}=\frac{a_k 2V\omega_i\omega_j}{\left(\omega_i\sigma_{ij}V_{r,ij}\right)_{\max}} \quad (10)$$

The parameter a_k is determined in order to limit the maximum time increment to a fraction of the computation time step Δt.

$$a_k=\min\left(\frac{n^{2}\left(\omega_i\sigma_{ij}V_{r,ij}\right)_{\max}}{p2V\left(\omega_i\omega_j\right)_{\max}},1\right) \quad (11)$$

Where p is a parameter greater than 1, corresponding to the minimum number of collisions to compute at each time step. Usually p is around 10.

The process of collision computation is the following:

- Estimation of $\left(\omega_i\sigma_{ij}V_{r,ij}\right)_{\max}$
- Computation of a_k
- Selection of two collision partners at random
- Computation of the collisions number increment and time increment
- The probability that the collision is taken into account is:

$$p_i=\frac{\Delta n_{c,k}}{\omega_i}$$

Due to the choice of a_k and β_k, p_i is always below 1.

This procedure is repeated for each cell until the time step is exceeded.

Variable Hard Sphere model (VHS)

The VHS model allows to model the variations of the cross sections with the relative velocity. We consider that the molecules are hard spheres with a variable diameter. The cross section is proportional to the kinetic energy of the collision to the power of -*ω*.

$$\sigma \approx \left(\frac{1}{2} m_r v_r\right)^{-\omega}$$

The exponent w is related to the exponent of the Sutherland law by the formula $\omega = 0.5 + a$. That allows to recover the variation of the viscosity with the temperature.

So, for one sort of collision, we have: $\sigma = C_{col} V_r^{-2\omega}$

For the reference temperature, the reference cross section can be obtained from the viscosity. The relation between the viscosity and the cross section is:

$$\mu_{ref} = \frac{15}{8} \frac{\left(\pi m k T_{ref}\right)^{1/2}}{\left(\Gamma(4-a)\right)(2-a)^{\omega} \sigma_{ref}}$$

Consequently, the coherence between the macroscopic properties (viscosity) and the microscopic properties (cross section) are preserved.

Larsen-Borgnakke model

The Larsen-Borgnakke model allows to handle the exchange between kinetic energy and internal energy.

Each species has ζ degrees of freedom. ζ can be deduced from the γ of each species. An internal energy E_{int} is attributed to each molecule.

Each collision has a probability p_{inel} to be inelastic. When the collision is inelastic, the total energy is redistributed between the internal energy and the kinetic energy as function of their distribution at equilibrium, knowing that the total energy is preserved.

The computation procedure is the following:

- The kinetic energy of the collision of the collision is computed :

$$E_{trans} = \frac{1}{2} m_r v_r^2$$

- A random number is chosen between 0 and 1. If this number is below p_{inel}, the collision is inelastic, otherwise the collision is elastic.
- It the collision is inelastic:
  - The total energy of the collision is computed:

$$E_{tot} = E_{trans} + E_{i,1} + E_{i,2}$$

  - The total energy is redistributed between internal post-collision energy E_{int}^{*} and kinetic post-collision energy E_{kin}^{*} using an acceptation/rejection method. The normalised distribution function is:

$$f'\left(\frac{E_{kin}^{*}}{E_{tot}^{*}}\right)=\left[\left(\bar{\zeta}-\omega\right)\frac{E_{kin}^{*}}{E_{tot}^{*}\left(1-\omega\right)}\right]^{1-\omega}\frac{1-\dfrac{E_{kin}^{*}}{E_{tot}^{*}}}{\left(1-\dfrac{1-\omega}{\bar{\zeta}-\omega}\right)^{\bar{\zeta}-1}}$$

Where $E_{tot}^{*}=E_{tot}=E_{\mathrm{int}}^{*}+E_{kin}^{*}$

And $\bar{\zeta}$ represents the mean degrees of freedom of the two partners.

- The internal post-collision energy E_{int}^{*} is redistributed between the two partners using an acceptation/rejection method. The normalised distribution function is :

$$f'\left(\frac{E_{i,1}^{*}}{E_{\mathrm{int}}^{*}}\right)=\frac{\left(\bar{\zeta}-2\right)^{\bar{\zeta}-2}}{\left(\dfrac{\zeta_{1}}{2}-2\right)^{\frac{\zeta_{1}}{2}-1}\left(\dfrac{\zeta_{2}}{2}-2\right)^{\frac{\zeta_{2}}{2}-1}}\left(\frac{E_{i,1}^{*}}{E_{\mathrm{int}}^{*}}\right)^{\frac{\zeta_{1}}{2}-1}\left(1-\frac{E_{i,1}^{*}}{E_{\mathrm{int}}^{*}}\right)^{\frac{\zeta_{2}}{2}-1}$$

In addition: $E_{\mathrm{int}}^{*}=E_{i,1}^{*}+E_{i,2}^{*}$

- If the collision is elastic:

$$E_{kin}^{*}=E_{kin}$$
$$E_{i,1}^{*}=E_{i,1}$$
$$E_{i,2}^{*}=E_{i,2}$$

DISTRIBUTION LIST

	Overall document		Summary
	Action	Information	

NbCars 36064
NbWords 7376
FileName Mclip_AM_V2.doc