HiSST: 2nd International Conference on
High-Speed Vehicle Science & Technology
11–15 September 2022, Bruges, Belgium

Approximation models for drag and heat flux of random tumbling objects in the transitional regime

Pierre Van Hauwaert[1]

Abstract

A Direct Simulation Monte Carlo (DSMC) code is presented with its validation and its applications. The code uses a cut cell approach combined with an on-the-fly Cartesian grid adaptation. The simulation is distributed using the MPI protocol and has the ability to rebalance the computation loads through the simulation. Validation of the code is demonstrated. Over 3400 simulations are carried out on 105 geometries in order to compute the random tumbling drag and heat rate in the transitional regime. The simulation setup to obtain reliable DSMC results in an automated way is outlined. Simulation results are compared with an approximation model. In order to compute the dimensionless random tumbling coefficients characterizing both the drag force and the heat rate, the random tumbling average of the projected shape is selected as reference surface. A formulation that can be applied to various topologies of object is used as a basis for the characteristic length used to define the average Knudsen number and for the equivalent radius used in the approximation model that computes the heat rate. For the 105 cones and cone-segments investigated, parameters can be selected such that the average difference between the approximation model and the DSMC results are below 1% for both drag and heat rate.

Keywords: *Approximation model for drag and heat rate, Atmosphere reentry, Debris, Direct Simulation Monte Carlo, Random tumbling, Transitional regime,*

Nomenclature

Latin

A – Apothem of a profile of cone or cone-segment
AoA – Angle of Attack
Ch – Cell size over mean collision distance
Ct – Time step over mean collision time
C_p – Thermal capacity per unit of mass
Cp – Pressure coefficient
Cf – Friction coefficient
Cq – Heat rate coefficient
CD – Drag coefficient
CFD – Computational Fluid Dynamics
CZ – Cell Zero or Zero Cell: abstraction layer grouping other cells together.
$DSMC$ – Direct Simulation Monte Carlo
e – Thickness of a profile of cone or cone-segment
$FNUM$ – Ratio between the real number of particles and the number of simulation particles
h – Enthalpy
K – Shape factor
Kn – Knudsen number
$Lref$ – Reference Length
N_A – Avogadro Number
P_∞ – Free-stream Pressure
q – Heat rate per unit of surface, in (W/m2)
Q – Heat rate, in (W)
QOI – Quantity of interest: CD and St
Req – Equivalent radius
R – Perfect gas constant
s – Speed ratio
$Sadim$ – Ratio between S_{ref} and $S_{outer}/4$
SM – Smallest scaled Median
S_{outer} – External surface of an object
S_{ref} – Reference surface
St – Ratio between the heat rate and the heat rate in the free molecular regime
T_∞ – Free-stream temperature
V_∞ – Free-stream velocity
VF – View factor
VHS – Variable Hard Sphere
w_i – Weight associated with an attitude i

Greek

[1]*R.Tech Engineering NL, Baarsjesweg 243-1, 1058 AA Amsterdam, The Netherlands, pierre@rtech-engineering.nl*

λ – Mean collision path
θ – Revolution angle for cone-segment
ρ_∞ – Free-stream air density
γ – Heat capacity ratio of a gas

Superscripts

CW – Cold wall
HW – Hot wall

Subscripts

∞ – Free-stream conditions
$cont$ – Continuum regime
FM – Free Molecular
i – relative to an attitude i
$pred$ – Prediction or interpolation scheme
ref – Reference
stg – Stagnation

1. Introduction

Risk estimation of object re-entering the atmosphere largely depends on drag force and heating rate. The computation of those two quantities with high fidelity methods, such as Computation Fluid Dynamics (CFD) or Direct Simulation Monte Carlo (DSMC), cannot be done within a reasonable time frame for a full earth re-entry. An alternative is to used correlation models where objects are assumed to have a specific tumbling motion. The most common type of tumbling motion is the *random* tumbling motion. An object moving with a *random* tumbling motion means that, in average, the free-stream direction is distributed uniformly over all possible attitudes and that the heat rate and drag forces are averaged accordingly. From a numerical standpoint, this can be modeled by a discrete set of attitudes as shown in Fig. 1. Correlation models are typically used in object oriented tools in order to assess numerically the survivability of vehicles reentering the atmosphere. In such tools, the vehicle geometry is reduced to an ensemble of simple shapes (box, cylinder, etc). Correlations to compute drag forces and heat loads are used for each shape, assuming a specific tumbling motion. From a defined break-up altitude, the trajectory, the thermal state and the possible ablation processes are computed for each fragment. Examples of such object oriented tools include Debrisk [1], Drama [2], Orsat [3] or SAM [4]. However, correlation models have often been derived in the 60s for non-destructive reentry [5] [6]. Therefore such correlations are often not conservative for destructive reentry [7]. Also, they are not made general to different (or complex) shapes since they are mainly based on experiments and observations.

In [8], an alternative methodology to compute the heat rate and the drag force in the continuum regime for tumbling objects is presented. That model reduces the dimensionality of the problem thanks to a normalization with respect to free-stream conditions and wall temperature. A further normalization of the problem over different shapes is done thanks to geometric quantities associated with each shape and the type of tumbling. Those normalizations result in error that can be up to the order of 25 % for the drag forces and up to a factor 2 for the heat rate. In order to mitigate those differences, a database depending on the geometry of the shapes is helping to deal with the deviation from the normalized behavior. This database is fed with CFD computations. One of the objective of the current paper is to extend the method presented in [8] to the transitional regime using bridging functions between the continuum regime and the free molecular regime. Therefore, the correlations in the continuum regime presented in [8] are used in the bridging function. Even the model presented in current paper used well known bridging functions, the combination of the various numerical method put together is new.

The final goal of this work is to assess the accuracy of the approximation model to predict drag and heat rate of object reentering the atmosphere with a tumbling motion in the higher part of the transitional regime, i.e. for Knudsen number between 1 and 10. The approximation model is compared with numerical results obtained from DSMC simulations for 105 geometries. Normalizing the data for all geometries is achieved thanks to specific geometric quantities. In the first part of this work, the DSMC code that is used to carry out the simulation is presented. The Direct Simulation Monte Carlo method developed by Bird [9] is one common method to simulate transitional flow. It consists in decoupling the collisions process of the particles from their motion. Only a fraction of the particles are modeled. An accurate solution can be obtained as long as the collision candidates to be selected at random are within the range of the local mean free path. Other criteria need to be respected such as having enough candidates for a collision partner and a time step small enough [9]. The implementation of the DSMC method in the code is described in the first part, focusing on the automation of the computations which is required to carry out a large number of simulations on different geometries. The second part

demonstrates the validation of the code. It is performed through comparisons with respect to the theory for the DSMC routines as well as cross comparisons with other codes or experiments. The goal of the last part is to describe the approximation model and quantify its accuracy thanks to comparison with data obtained from DSMC. Over 3400 simulations on 105 geometries are carried out and the simulation setup to obtain reliable results in an automated way is outlined. The approximation model is then described. The bridging function between the free molecular and the continuum regime are taken from the literature but the geometry quantities involved in the formulations are chosen in order to minimize the variation between shapes. The difference between the approximation model and the DSMC results are presented and discussed.

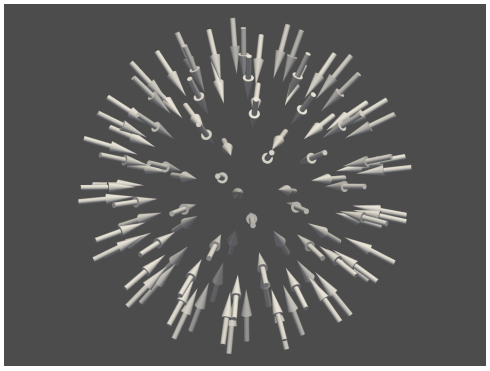


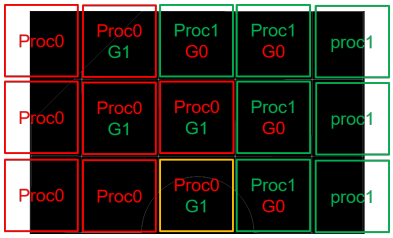
Fig 1. Representation of a set of 100 attitudes distributed uniformly and used to model numerically a random tumbling. The discretisation follows the Fibonacci sphere distribution.

2. DSMC Code description

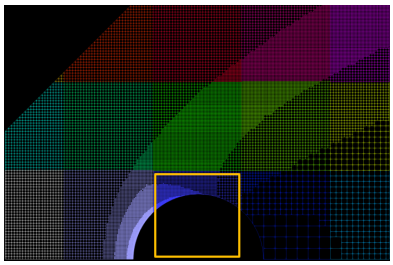
The motivation to develop a Direct Simulation Monte Carlo code goes back in 2014 with the interest in carrying out simulations on any kind of geometry, including thin objects and large – industrial – cases. Freely available codes such as DSXV codes [10] or the open source code OpenFOAM [11] were considered at the time but considered unsatisfactory. Therefore a new DSMC code called Mistral-DSMC was developed. The SPARTA [12] code had not been identified at the time. The key reasons to develop a new code were to automate the simulation workflow as much as possible while ensuring reliable results. Mistral-DSMC has been used for internal studies, for the elaboration of new models in an upcoming version of DEBRISK [7], or for design of demise studies and comparison with Spacecraft oriented demise codes [13, 14] . The code is written in C++. It can run 2D and 3D simulations. The Cartesian structure of the code and its parallel implementation are first outlined. Geometry, grid refinement and collision models implemented are then described.

2.1. Grid structure

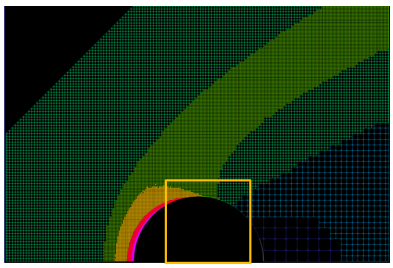
The structure of the code is based on Cartesian grids. At first, the simulation domain is discretized into a uniform Cartesian grid made of abstract cells whose length is equal to a number of time the mean free path of the free-stream conditions. Those abstract cells are referred to as CZ for cell of *level zero* or *zero cell*. Each of them is assigned to a processor. The direction of the Cartesian grid is not aligned with the direction of the inflow but with the Cartesian frame used to define of the geometry. Some of them also are flagged as ghost cells to be used for parallel communication. This is sketched in Fig. 2a. A *zero cell* is an abstraction object that groups geometric cells together as sketched in Fig. 2b where cells that belongs to the same *zero cell* have an identical color. Within each *zero cell*, a multi-level cell structure with partial non-conforming connectivity is used to discretize the space. The cell size can be adapted to match requirements of the DSMC method as illustrated Fig. 2c where cells of identical size have identical color. Each cell stores information related to the simulation particles, the geometry (if it contains any), and the local sampling quantities.



(a) Diagram of 3x5 *zero cells* that discretize the domain. Here, each *zero cell* is either assigned to proc0 or proc1. Ghost cell for adjacent proc are label with the letter "G"



(b) Diagram of 3x5 *zero cells* that discretize the domain. Each colour represents a different *zero cell*. The wireframe represents the grid.



(c) Multi level grid structure. The wireframe represents the grid and is colored by grid refinement (depth) level.

Fig 2. Grid layouts.

2.2. Load balancing and communication

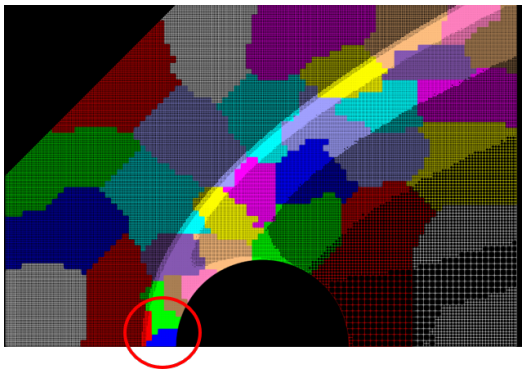
The load balancing is carried out using the METIS library [15]. It aims at partitioning the *zero cell* between the available processors. After load balancing, the area of communication between each processor domain is minimized as sketched Fig. 3a or Fig. 3b. As a results, the number of ghost cells is minimized as well. This does not necessarily minimize the communication (number of particles) between processors because the partition boundaries are not aligned with the free-stream direction. After each *zero cell* has been assigned to a processor, at least one layer of *zero cell* adjacent to each processor domain are defined as ghost *zero cell*, see Fig. 2a. The number of adjacent layers of ghost *zero cell* depends on the triple product of the size of the *zero cell*, the timestep of the simulation and one user defined parameter set in order to keep the memory overhead low while ensuring all simulation particles can be transferred during the communicating step. If a simulation particle cannot be transferred during the communication step because the number of layers of ghost cells was too small, the simulation particle is destroyed and a counter of destroyed simulation particle is incremented so it can be monitored and the user defined parameter changed if that ever happens.

During a simulation run, a rebalancing of the simulation is attempted only after a specific number of iterations, typically 500, and is only carried out if the simulation is found to be unbalanced within a threshold. Rebalancing is then attempted at a fixed frequency, typically 500 to 2,000 iterations depending on the time step and the configuration. No rebalancing is carried out after another specific user defined number of iterations. This is to avoid overhead due to rebalancing the simulation at the point it cannot be effectively rebalanced. Rebalancing becomes more useful as differences in density within the domain gets larger and is associated with a larger number of iterations for the simulation to reach steady state. For a typical object reentering the atmosphere at constant speed, the usefulness of rebalancing increases as altitude decreases. During the rebalancing of the simulation, the geometry refinement layout, the sampling quantities as well as the simulation particles are transferred.

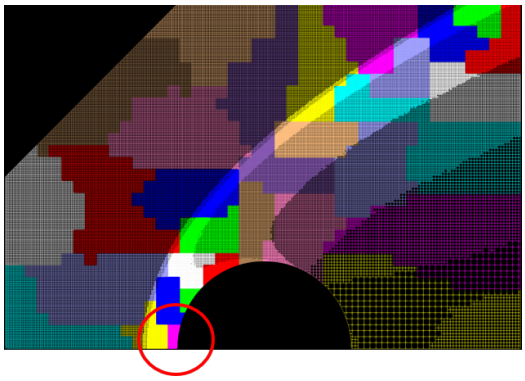
Communication is carried through MPI calls once per iteration to move the particle across the cells using the ghost cells for communication. Additional communication such as post-processing from the previous iteration, e.g. to integrate the wall quantities (e.g. heat rate) over the triangulated geometry, are carried out just after the "move" operation.

The definition of the size of the *zero cell* impacts the scalability of the code. A smaller size of *zero cell* – e.g. Fig. 3a compared to Fig. 3b – allows for a better load balancing as well as less memory overhead because less ghost cells are needed. A drawback of using a small *zero cell* is that the maximum cell size may not be large enough to have enough simulation particles which can break DSMC best practices [16]. For example, in the wake of an object (e.g. right side of Fig. 3b), area of low density exist and the size of the cell required to have enough simulation particles can be larger than the size of the *zero cell* chosen. By default the size of the *zero cell* is set to a multiple (integer) of 0.9 times the targeted

ratio of the local cell size over the local mean free path - this quantity referred as "Ch" later in this work is typically set to 1 or less - in order to avoid the back and forth grid refinement oscillations in the free-stream region when adapting the grid.



(a) Grid around a 2D cylinder colored by assignment to one of the 32 processors. The size of the *zero cell* is $0.9 * 2$ times the mean free path of the free-stream conditions



(b) Grid around a 2D cylinder colored by assignment to one of the 32 processors. The size of the *zero cell* is $0.9 * 4$ times the mean free path of the free-stream conditions

Fig 3. Comparison of load balancing using different *zero cell* size.

2.3. Geometry, cut cell and move operation

Geometries are defined through segments of line (2D) or triangular surface elements (3D). In order to detect which side of the surface shall be considered as the "fluid" side, the orientable property of triangular closed surfaces in combination with their boundary condition – inflow or wall – are used. A similar approach is used in 2D. A Cartesian grid is automatically generated in the "fluid" domain. Cartesian cells that intersect with the geometry are called a "cut cell" and are also generated. Such configuration can results in one, see Fig. 4, or multiple segregated volumes, see Fig. 5. The second configuration is called a "split cell", see Fig. 5.

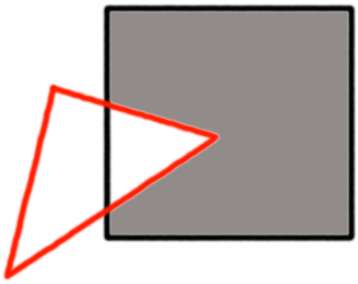


Fig 4. A cut cell is created when a the geometry (red) intersects the Cartesian cell (black). The resulting volume (grey) shall be computed for DSMC procedures.

Since the value of the volume is required for the DSMC collision procedure, in the case of split cell, each segregated volume must be identified and its value computed numerically. To identify cut cells, each Cartesian cell is checked for intersection with the geometry. Next, the segregated volumes are identified with an approach similar to [17]. However, each face of the cut cell or split cell are triangulated. As

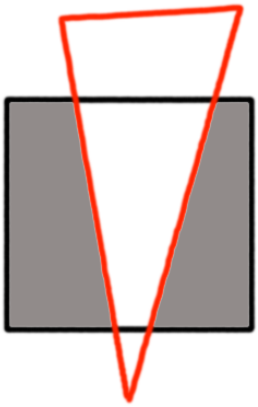


Fig 5. The geometry (red) intersects the Cartesian cell (black) such that 2 disjoint flow volumes (grey areas) are created.

a results, each cell is an orientable closed surface made of triangle. It is watertight and each normal can be oriented toward the ”fluid” side. Consequently the volume of each cell is computed by numerical integration looping over each triangle. There are several reasons for using a cut cell approach. It allows to model thin objects such as in Fig. 6. Indeed the geometry discretization and the size of the volume grid required to group the simulation particles together (which is a requirement for the DSMC simulation to converge) can be decoupled [18]. The trajectory of each simulation particle is modeled as a straight line. So the computation of the position when a particle move from a Cartesian cell to another is straightforward. However, within each cut cell and split cell, collision with the wall needs to be accounted for and the intersection between the particle direction vector and each triangle of the segregated volume the particle belongs to must be tested (ray tracing).

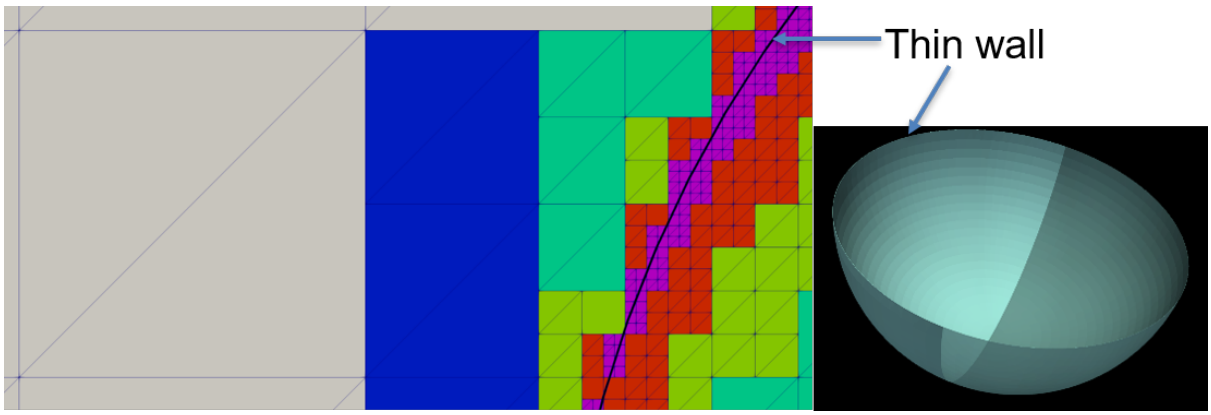
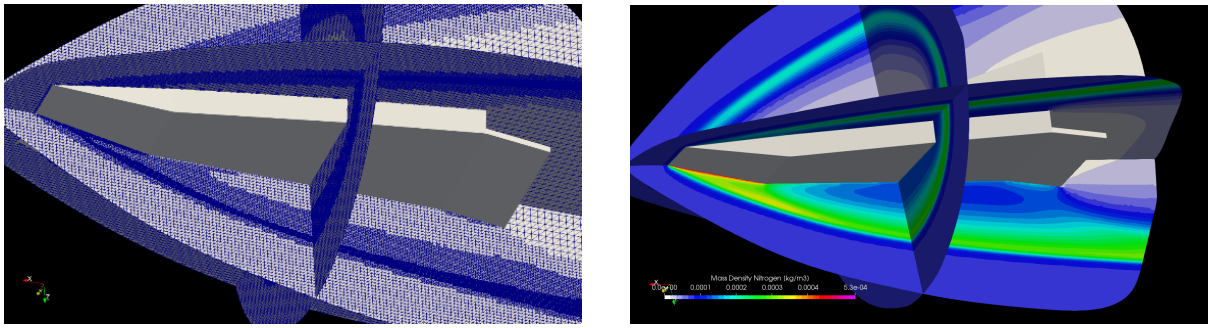


Fig 6. Half hollow sphere (right) and cut through a 3D grid (left) used for a DSMC simulation picturing split cells.

2.4. Grid adaptation

The grid adaptation algorithm sets each cell to a size lower than a user defined fraction of the local mean free path – Ch – usually 1 or lower. The value of 0.5 is used in this work when no further information is given. In order to compute the local mean free path in each cell, the number of collisions between each couple of species as well as the corresponding product of the total collision cross-section with the relative speed are sampled. Using other sampled quantities (density, velocity components, etc), the mean collision distance is computed from eq 1.41 of [9] which is then used for grid adaptation.

The grid adaptation is correlated with the mass density as illustrated by the couple Fig. 7a and Fig. 7b. In the case not enough simulation particles have been sampled inside one cell, no refinement is commanded for that cell which results in a group of cells having that same characteristic to be merged together unless the maximum size (*zero cell*) has been reached already. Finally, a minimal level of grid refinement can be imposed at the walls.



(a) Grid wireframe. **(b)** Mass density.

Fig 7. Mistral-DSMC simulation of generic sharp edge re-entry vehicle. ATD3 workshop 2017

The grid adaptation is carried out on one *zero cell* at the time. With the mesh adaptation, the different quantities are recomputed from the previous mesh based on the ratio of the volume of the cells. In order for the interpolation to be conservative, it must be possible to evaluate which split cell (or cut cell) is included inside another one. In order to do so, a point inside each segregate volume of each split cell (and cut cell) is generated. Then it is possible to identify if a segregate volume is contained by another one by testing if the point inside the first segregated volume is also inside the second one. That test is achieved by counting the number of intersections of a ray (starting from the generated point) with the triangles that define the second segregate volume as illustrated Fig. 8. If the number of intersection is uneven, the point is inside to watertight volume.

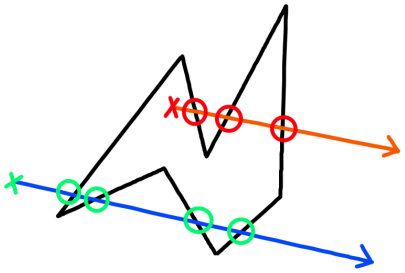


Fig 8. Illustration of detection if a point is inside or outside a polygon/polyhedron depending on if the number of intersection is even (outside) or uneven (inside).

That test is also carry out in order to re-assign each simulation particle to the correct segregated volume in case of split cells, during the mesh adaptation process. The grid adaptation can be carried out even if the sampling quantities are not reset because the sampling values as well as collision reminders are conserved during the mesh adaptation. After the grid adaptation, the ghost cells of *zero cell* assigned to a different processor needs to be updated. Therefore, the grid structure within each *zero cell* is broadcast to processors they are a ghost *zero cell* of.

2.5. Collision models

The NTC model [9] is used to simulate the collision rate. Selecting collision partners that are close improves the convergence of the DSMC method [16]. But instead of looking for the nearest neighbor

for each simulation particle, e.g. virtual sub-cells [19], sub-cells with a span-out-approach, as illustrated Fig. 9, are used. The downside of using the nearest neighbor search is that it gets inefficient if a large amount (>30) of simulation particles is used.

Here, sub cells are created on-the-fly within each cell to decrease the collision distance between tested collision pairs. With the span-out-approach, an array stores the simulation particles based on their Cartesian coordinates. That is also the case for split cell but a different array is used for each segregated volume within a split cell. For example in Fig. 5, two arrays shall be used, one for each of the two grey volumes. Simulation particles are assigned to a sub cell by computing the sub cell index directly from the simulation particle position. The collision partner is selected at random in the current (red) sub cell and if none can be found the selection domain is extended (green, then purple, then yellow) until at least one collision partner can be found. Collision between previously collided partners is forbidden [9]. Fig. 9 illustrates a drawback of this method which is a configuration where the collision partner found is not the closest one. The time penalty is limited since most operations are on average linear with the number of simulation particles. In this work 8 sub cells are used in each direction for a total of 512 sub cells per cell in 3D and 64 sub cells per cell in 2D.

The Larsen-Borgnakke from [9] which allows double relaxation is implemented. The variable hard sphere (VHS) model from [9] is also implemented.

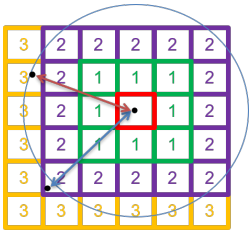


Fig 9. Illustration of the sub cell selection.

3. Code validation

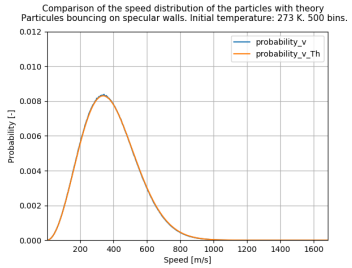
Validation of the collision routine is presented and are followed by 2D and 3D cross-code validation including some experimental data. All simulations use the VHS model with the parameters described in Table 5.

3.1. Validation of collision routine

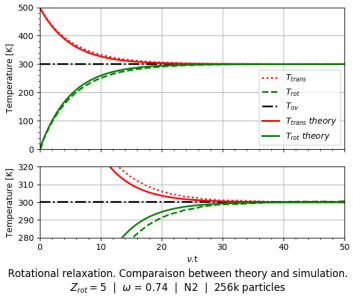
In order to validate the ability of the code to reproduce the speed distribution at equilibrium, a 4.56 meters diameter sphere with specular wall is filled with Argon. The Argon properties are taken from Table 5. The initial temperature is 260 [K], the density is 5e-8 [kg/m^3] and the velocity is zero. 12788 simulation particles are simulated. The time step used for the simulation is 0.00209 [s]. The simulation is carried out for 5,000 iterations during which the velocities are binned. The velocity bins are compared with theory in Fig. 10a which shows that the probability distributions are identical.

In order to validate the implementation of the serial Larsen-Borgnakke model, a test case from [9] is reproduced. A 0.02 meters width cube is filled with N2 whose numerical parameters are defined in Table 5. The translation temperature is set to 500 [K] and the rotation energy is initially set to zero. The particle density is 1e20 particles per cubic meters and 256,000 simulation particles are used. The simulation is run for 200 iterations with a time step of 5.607e-6 [s] which is a quarter of the mean collision time. The rotational collision number is set to 5. Comparison with the analytical solution in Fig. 10b shows satisfactory results.

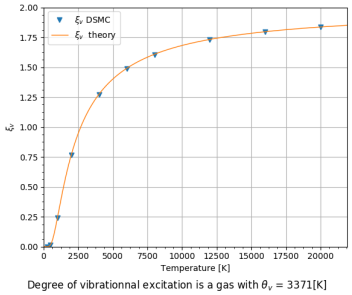
Finally, another test case from [9] is reproduced in order to validate the ability of the code to reproduce the degree of vibrational excitation at equilibrium as shown in Fig. 10c.



(a) Comparison of speed distribution with the analytical Boltzmann distribution



(b) Rotational relaxation in a simple gas



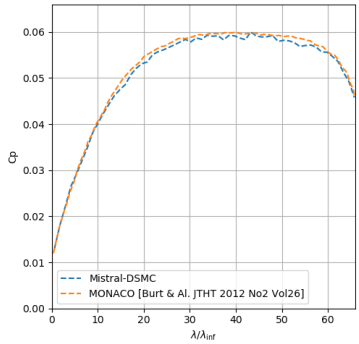
(c) Degree of vibrational excitation.

Fig 10. Comparison with analytical results.

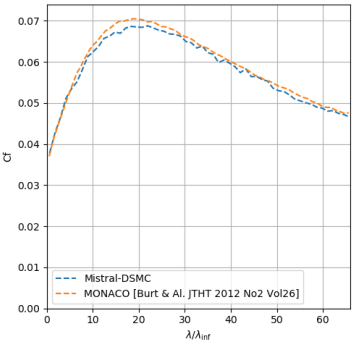
3.2. 2D validation

A cross-code comparison with MONACO [20] on a 0.112-meter-long flat plate is carried out. The free-stream conditions "M20.2 plate" from Table 6 are used. A rectangular domain of $0.048 * (0.112 + 0.03)$ square meters is used for the simulation with a 0.03 meters buffer between the inflow and the beginning of the flat plate. Diffuse reflection with fully thermal accommodation at the temperature of 290 [K] is assumed for the flat plate. N2 specie with properties from Table 5 is used. Only rotation mode with a fixed rotational collision number of 5 is considered. The cell size target for the mean collision distance is set to 0.5. At steady state 265,000 simulation particles are simulated. The time step used is 3.01e-7 [s]. The simulation is run for 20,000 iterations. The samples are reset at 1,000 iterations and at 2,000 iterations. The grid adaptation is carried out every 1,000 iterations. 64 sub cells per cell are used. The simulation took 625 sec on 4 cores. Results presented in Fig. 11a, Fig. 11b and Fig. 11c show good agreements. A larger amount of simulation particles only impacts the results by making the curves smoother but does not change the local averaged value.

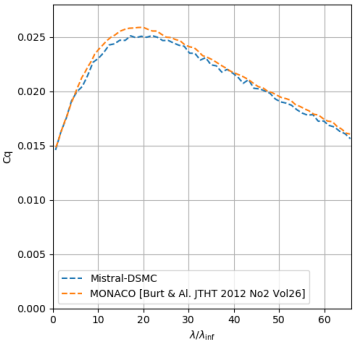
In order to evaluate the performance of the code, the same simulation but with 4 times as much simulation particles is carried out with various processors configurations. The results presented Fig. 12 reveal a decent scaling. The code performs better when the load is spread across multiple nodes which suggests the scaling is limited by the memory bandwidth of the machines used.



(a) Comparison of pressure coefficient with MONACO



(b) Comparison of friction coefficient with MONACO



(c) Comparison of heat rate coefficient with MONACO

Fig 11. Comparison with analytical results.

Other 2D code comparisons such as with MONACO [21] on a 12-inches cylinder for different conditions

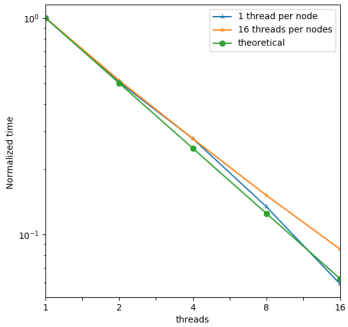


Fig 12. Scaling test on the flat plate.

have been carried out for the first spacecraft demise workshop – SCDW – at Lisbon in 2015 and are not repeated here.

3.3. 3D validation

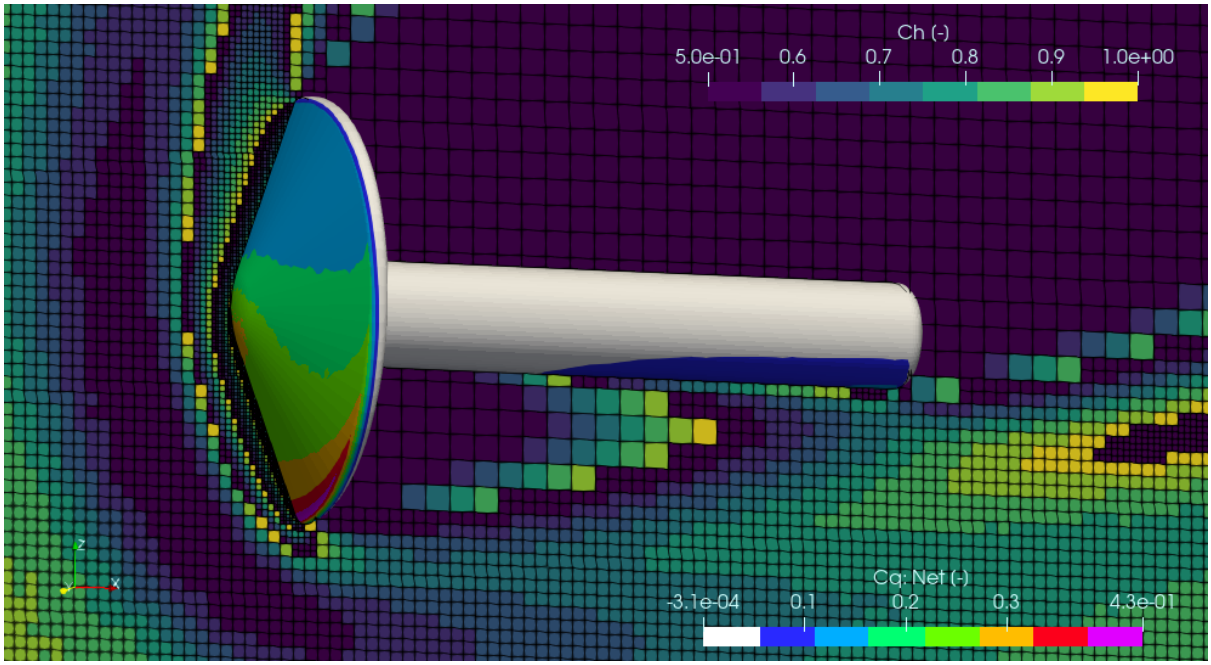


Fig 13. Adapted grid in the plane of symmetry colored Ch (cell size over mean collision distance) and surface grid of the probe colored by heat rate. Angle of attack: 30°

Simulations on a 70° blunt cone is carry out for 4 AoAs (angles of attack) and results are compared with experimental data from [22] as well as with the MGDS code [23]. The thermocouples positions are labeled from 1 to 8 such as described in [22]. The free-stream conditions "M20.2 planetary probe" from Table 6 are used. A spherical domain is used for the simulation. Diffuse reflection with fully thermal accommodation at the temperature of 290 [K] is assumed for the probe. N2 molecule with properties from Table 5 is used. Only rotation mode with a fixed rotational collision number of 5 is considered. The target for the cell size of the mean collision distance is set to 1. The time step used is 1.24e-7 [s], which is 0.008 times the mean collision time of the free-stream conditions. The simulation is run for 100,000 iterations. The samples are reset every 1,000 iterations until 12,000 iterations. The grid adaptation is carried out every 1,000 iterations and is stopped at 30,000 iterations. The final grid for a 30° AoA is

shown Fig. 13. The number of simulations particles at steady state reached 157 millions for that same case. The simulation took 56 hours on 64 cores.

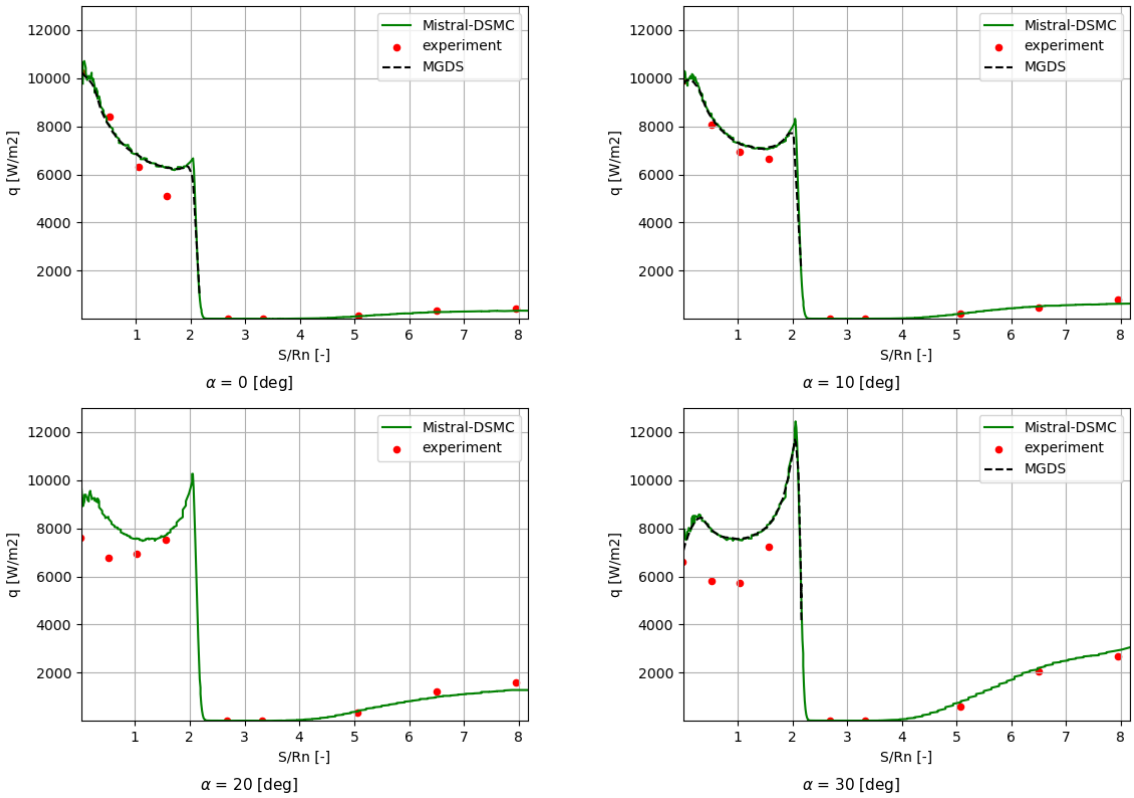
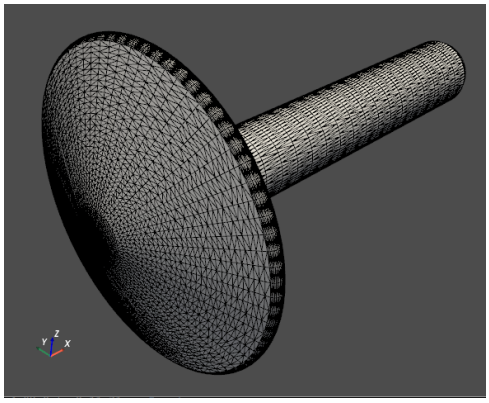


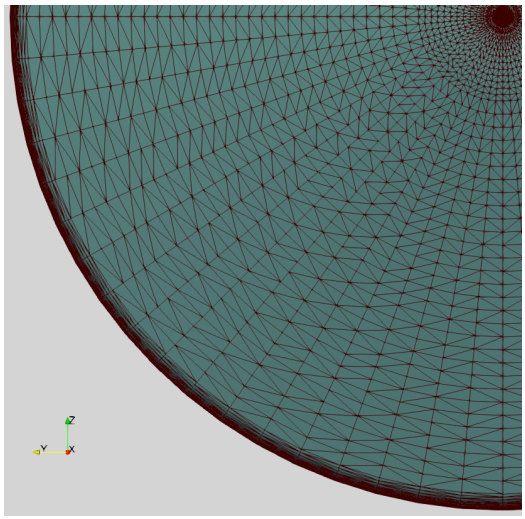
Fig 14. Comparison of Mistral-DSMC heat flux predictions with experiment and MGDS.

The results are presented Fig. 14. Good overall agreement is obtained with MGDS on the windward side. Significant differences with experimental data exist on the windward side, while there is a very good match with the experiment data behind the probe for the last 5 thermocouples. [23] suggest that the differences with the experiment on the windward side could come from the inflow conditions that were not uniform even that was accounted for in the experimental data with an experimental free-run. The largest difference with MGDS are located near the shoulder. A possible explanation is the difference in surface discretization to model the probe geometry. The MGDS simulations use a coarser mesh than in this work at the shoulder according to the screenshot of the triangulated surface available in [23]. In this work the surface grid is based on an axisymmetric topology presented in Fig. 15a and Fig. 15b with a larger grid resolution at the shoulder where the peak in heat rate occurs, as illustrated Fig. 15c. Finally, a higher surface discretization adds computational cost associated with the higher number of cells to be checked during the ray tracing procedures. And the relative error on triangular elements decreases with the inverse of the square root of the number of simulation particles colliding the triangle. So using smaller elements required to increase the number of iterations to reach the same error level. The surface grid at the nose of the probe displayed at the top right corner of Fig. 15b illustrates an area where more than one hundred vertices converges and where the local convergence is more difficult – as illustrated in Fig. 15d – for the small triangles, area where scattering in the results is observed.

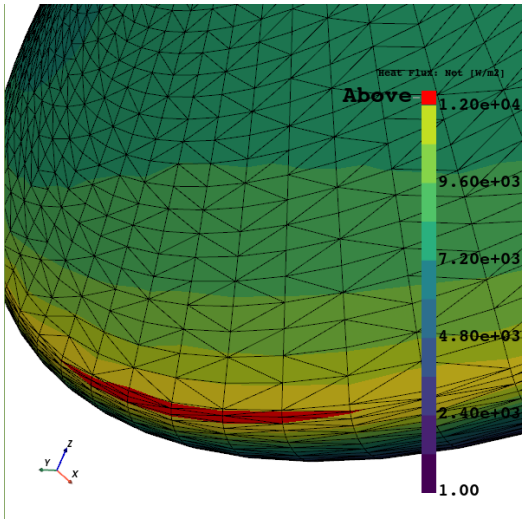
With the exception of the differences at the shoulder and at the nose – both attributed to a different surface mesh – very good agreement is obtained with MGDS. And very good agreement is obtained with experimental data on the leeward parts for all attitudes.



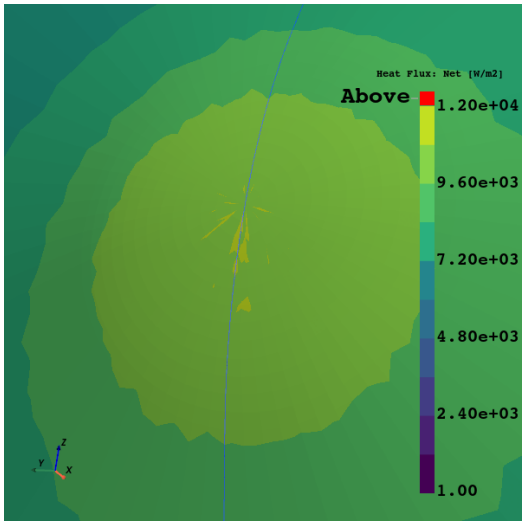
(a) Surface mesh from afar



(b) Front side of the surface mesh



(c) Local increase in heat rate at the shoulder junction



(d) Local oscillations in heat rate over the blue line (y=0) due to under sampling for very small triangles.

Fig 15. Surface mesh and DSMC results for the planetary probe.

4. Comparison between the approximation model and results obtained from DSMC

This section aims at evaluating how accurate the prediction in drag and heat rate of random tumbling objects re-entering the atmosphere in the transitional regime are. The predictions are made with an approximation model. However, even the model is an approximation based on empirical comparison, it can be called ”analytical” by opposition to ”numerical” because it can be evaluated only using formula as long as the reference surface can be computed analytically. DSMC simulations are used to generate a large amount of data to be compared with the approximation model used for the prediction. The large amount of DSMC simulations requires to automate the simulations setup while ensuring reliable results that are used to compute the random average quantities. This is what is explained after describing the geometries. The models to predict the heat rate and drag forces are then described. The bridging formulations are formulated through non-dimensional quantities. For the heat rate, it is expressed through the ratio between the heat rate and the heat rate in the free molecular regime: the St number. For the drag, it is expressed through the drag coefficient: CD. Geometric quantities involved in the formula of the approximation model are described. Choosing specific geometry parameters to compute the normalized parameter CD and St helps improving the accuracy of the model. Finally the DSMC results are compared with the approximation model and are discussed.

4.1. Geometry specification

The geometries studied are cones and cone-segments, described by a set of five variables. The corresponding notations are presented Fig. 16. For all geometries, the largest diameter “DG” is fixed to 1 meter. The four other parameters “h”, “e”, “DP” as well as “θ” the angle of revolution of the profile vary. The geometry matrix is presented in Fig. 27 and Fig. 28. 60 axisymetrical shapes (cones) and 45 cone-segments with an angle of revolution of 180° are picked resulting in a total of 105 shapes.

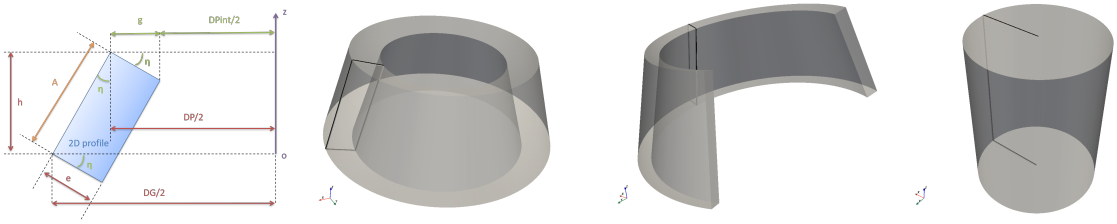


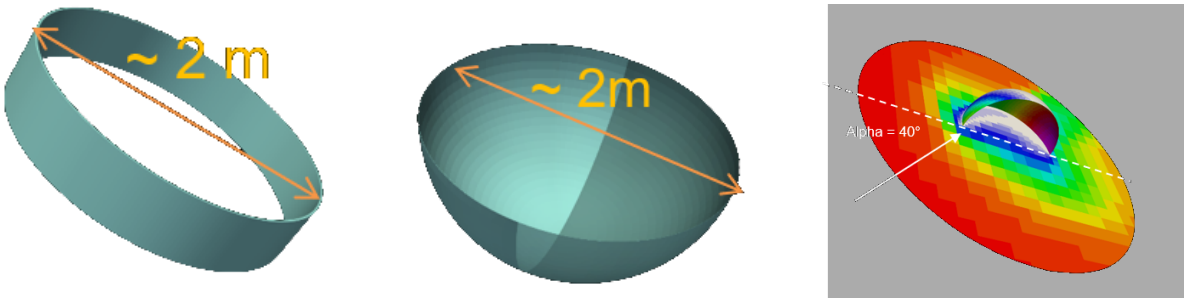
Fig 16. Definition of the geometry parameters and overview of the geometries. From left to right: 2D profile of a cone, cone, cone-segments with a revolution angle “θ”, cylinder.

Contrary to ”simple” shapes such as spheres, boxes or cylinders, cones and cone-segments are generally not convex, and therefore exhibit shadowing effects. For some of them, the small gap create local densities that are several order of magnitude larger than the freestream density and beyond what would be seen for convex shapes. The combination of those two effects makes the set of geometries described here particularly interesting when comparing DSMC results with the approximation model, especially because the model discussed further aim at being as general as possible.

4.2. Numerical setup and convergence of DSMC simulations

The simulations have been carried out for an altitude of 110 km – ML110 in Table 6 – which is not numerically challenging. However, a large number of simulations need to be carried out, so it matters that a good compromise between speed and accuracy can be found, even for the critical cases. The objective is to set a single set of numerical parameters such that the simulations can be carried out for all geometries. The effects of the time step, of the size of the inflow condition as well as the number of simulation particles are investigated. This has been done on a selection of geometries presented Fig. 17a and Fig. 17b.

For all simulations, the VHS model is used and no chemical reactions are considered since it will not impact the results for this specific free-stream conditions as demonstrated by [24]. Constant time steps and constant weight – ”FNUM” using the notations of [9] – for the simulation particles are used. The Weight, FNUM, is the ratio between the real number of particles and the number of simulation particles used in the DSMC simulation. The smaller is the weight, the larger is the number of simulation particles.



(a) Tube

(b) Half hollow sphere

(c) Velocity magnitude in the plane of symmetry and pressure coefficient at the wall.

Fig 17. Geometry investigated for numerical convergence and illustration of the angle of attack.

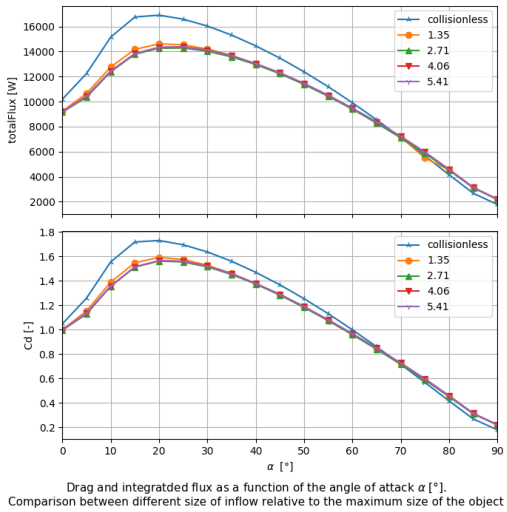
Simulations have been run for 40,000 iterations. Every 2,000 iterations mesh adaptation has been carried out. Samples are reset every 2,000 iterations until 20,000 iterations.

The effect of the size of the inflow has been investigated first. The size of the inflow impacts the volume of the domain which is proportional to the number of particles for a fixed particle weight. Therefore, minimizing it decreases the time needed to run the computations and decreases the memory requirements. The inflow is modeled by a sphere of radius whose diameter is proportional to the size of the object (twice the largest distance from its center of mass) multiplied by a parameter larger than one. Simulations are run with a time step of 0.008 times the mean collision time of the free-stream conditions, and FNUM is set to 5.08e13. Even though the shock is relatively diffuse with those conditions, the impact of that ratio on the results is small (below 4%) as illustrated by the comparison of the drag coefficient and heat rate in Fig. 18a and Fig. 18b. This could change for a ratio closer to one but has not been investigated. A ratio of 3 is chosen to minimize the effects on the drag forces and the heat flux. This ratio is valid for geometry with a maximum size of approximately 2 meters large. It will decrease if the Knudsen number decreases (the shock is less diffuse), which happens if the altitude decreases for a fixed object size, or if the size of the object increases for a fixed altitude, and vice-et-versa.

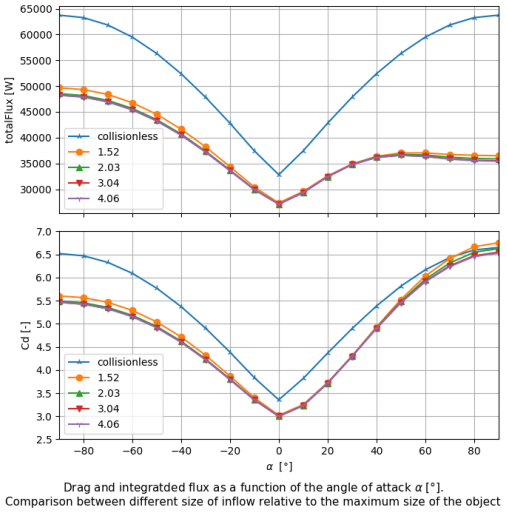
Using the inflow size previously defined, the effect of the number of simulation particles – the weight FNUM assigned to each simulation particle – on the results has been investigated. The results presented Fig. 18c and Fig. 18d show a stronger effect on the half hollow sphere. For positive angle of attack, the difference increases with the angle of attack up to 90°. Fig. 17c shows the definition of the the angle of attack. For simulations with the lowest number of simulation particles, the cells with large density – e.g. at the wall – do not have enough simulation particles for the collision rate to be modeled accurately: there are not enough simulation particles to be selected as collision partner in the cells. The heat rate are getting closer to those of a "collisionless" simulation as the number of simulation particle decreases. Selecting a weight – FNUM – corresponding to 5 millions simulations particles, or more, has no effects on the results as illustrated in Fig. 18d. The 5 millions simulations particles – purple line in Fig. 18d – corresponds to a particle weight of 5.08e13. Consequently a particle weight of 5.08e13 is selected for the computations. The use of a lower time step – $Ct = 0.004$ – has been investigated but has no significant effect on the results for the two cases. This concludes the initial convergence study.

For cones with small holes – see Fig. 24 – the density is sometimes larger than those observed for the half hollow sphere case presented above for the same conditions. To filter which simulations shall have its computational parameter adjusted, three quantities are retrieved for all the cells in the y=0 plane:

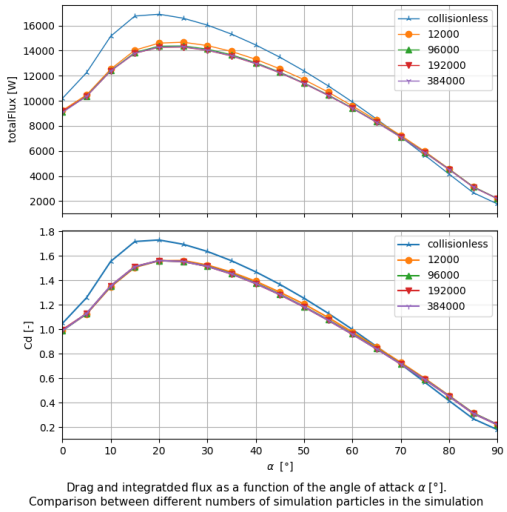
- the ratio of the mean collision time over the time step: Ct
- the ratio of the mean collision distance over the cell size: Ch



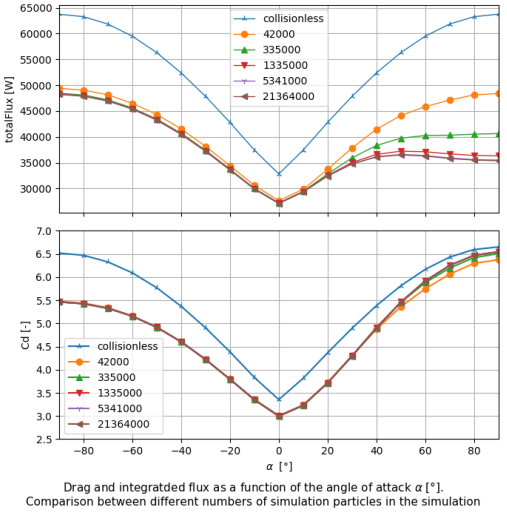
(a) Effect of the size of the inflow. Tube. S_{ref}=1.



(b) Effect of the size of the inflow. Half hollow sphere. S_{ref}=1.



(c) Effect of the number of particles. Tube. S_{ref}=1.



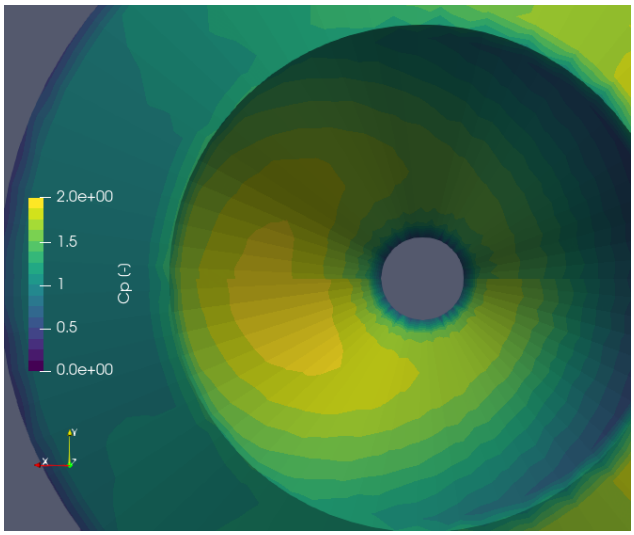
(d) Effect of the number of particles. Half hollow sphere. S_{ref}=1.

Fig 18. Effect of the size of the inflow and the number of particles on a tube and a half hollow sphere.

- the number of simulation particles

If more than five cells do not fit the best practices of DSMC, the geometry is re-run for the full set of attitudes with 8 times more particles and a time step four times smaller. The effect of using the new parameters is presented for the geometries showing the largest difference in Fig. 24. No cone-segments are flagged to be re-run as they are ”open” so the local increase in density is much lower. The effect is smaller than 2% on the integrated contributions (drag and heat rate) when considering any attitudes-geometries combinations. In order to obtain the random average of drag force and heat rate, a weighted average of the drag forces and heat rates over the attitudes is carried out, see paragraph 4.4.3. As a consequence, the error on the random average values is necessarily lower than the maximum error. Indeed the error of the contribution of each attitude can either be positive or negative thus they can compensate each other. Addtionnally, not all errors are equal to the maximum error. Still, significant local differences are observed for some specific configurations such as displayed Fig. 19a, for which the pressure coefficients for both set of parameters are compared in Fig. 19b. Noticeable changes in pressure coefficient are observed. Not all differences in pressure distribution translate into drag coefficient differences because the normal of the wall at the location where differences occur mainly are perpendicular to the free-stream direction. It is clear that the initial set of parameters is not able to accurately handle those cases. Even it would have been interesting to continue the convergence study for a few other cases – not discussed here – it is stopped because of the small impact on the integrated coefficients, especially because it concerns AoA close to +/- 90° which have less impacts on the random tumbling drag forces and heat rates for axisymmetric shape. Indeed, when considering the random tumbling of an axisymmetric shape, such as in Fig. 17c, the AoA of +/- 90° is the one with the less probability, see paragraph 4.4.3.

(a) Cone with small hole and "large" (45°) half cone angle "η".



(b) Comparison of pressure coefficient between the initial set of parameters (upper half) and the new one (lower half). AoA=70°.

Fig 19. Example of geometry where the convergence is more difficult

Between a million of simulation particles and a couple of tenths of millions of simulation particles have been used depending on the geometry and the weight of the simulation particles. Simulation of the initial batch use 1.7 million particles per simulations on average. Simulation of the second – more computationally expensive one – use 14 millions particles per simulation on average. The total of the two batches were run in about 2 weeks using 588 cores of our cluster on average. The main simulation parameters are summed-up in Table 1

Table 1. Sum up of the simulation parameters for both initial and second simulation batches

	Number of simulations	Average wall time, in hours	Average number of simulation particle, in millions	Cores	Ct
Initial batch	3210	5.0	1.7	4	0.008
Second batch	239	12.3	14.2	8	0.002

4.3. Geometric quantities for normalization

Existing correlations for computing the drag forces and heat rate usually involve geometry dependent normalization parameters, such as a reference surface, a reference length or an equivalent radius [25], [26]. Indeed, correlations based on experimental data or high fidelity numerical code can be used to calibrate the models. However, the models need to be applicable to a wide range of geometries so they can be used for realistic configurations where objects can be of any size. Moreover during the destructive re-entry of a random tumbling object, ablation occurs and the geometry of the object changes over time. Thus, choosing the right geometry quantities for normalization so that the correlations stay accurate for a wide range of geometry parameters has a direct impact on the accuracy of object oriented code such as [1]. The reference surface, the reference length and the equivalent radius used in this work to achieve the goal explained above are defined in paragraphs 4.3.1, 4.3.2 and 4.3.3 respectively.

4.3.1. Reference surface, S_{ref}

The surface of reference S_{ref} selected is the motion random average of the projected shape. For convex shape, Cauchy [27] demonstrated in the 19th century that it is a quarter of the surface of the geometry S_{outer}, as defined by Eq.1.

$$S_{ref} = \frac{S_{outer}}{4} \tag{1}$$

For some non-convex shapes, it is possible to compute S_{ref} analytically. In [28] the primitives are broken down into the convex and concave parts. For the convex part the area of S_{ref} is taken as $S_{outer}/4$. For the concave parts the view factor (VF) to the outside world is computed analytically and the area is taken as $VFS_{outer}/4$. Applying this method for example to a tube (external radius "R", height "L", thickness "e") results in the expressions described in Eq.2 to Eq.6.

$$S_{ref}^{tube} = \frac{\left(S^{out} + S^{top} + S_{bot} + S^{in}\right)}{4} \tag{2}$$

$$S^{top} = S^{bot} = \pi(R^2 - (R-e)^2) \tag{3}$$

$$S^{out} = 2\pi RL \tag{4}$$

$$S^{in} = \pi(R-e)L(\sqrt{N^2+4} - N) \tag{5}$$

$$N = \frac{L}{R-e} \tag{6}$$

For concave shapes for which no analytical formulation has been found in literature, S_{ref} has been computed numerically from a triangulated surface mesh identical to the one used for the DSMC simulations. A numerical code that uses ray tracing has been developed to carry out this task. The random average is modeled by a discrete number of direction such as shown in Fig. 1. For each direction, the area of

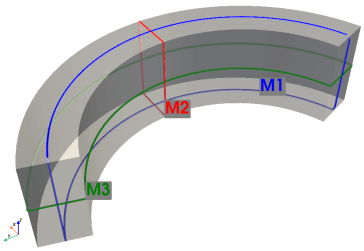


Fig 20. Sector of cone and its 3 medians lines M1, M2, M3.

the projected shape is computed. The numerical random averages S_{ref} is computed by averaging that projecting area over a large number of direction (attitude). In order to compute the projected area of the shape in one direction, rays are launched from the surface of the object toward that direction. For each triangles, 100 rays are launched. And the origin of the rays are distributed uniformly over the surface of the triangle . By identifying the rays that does not interest the surface, the projected surface toward the specific direction can be computed. The OptiX library from NVidia is used for Ray Tracing. The numerical random averages S_{ref} is computed by averaging over a large number of attiudes following the Fibonacci sphere distribution .

4.3.2. Reference length, Lref

A reference length – "Lref" – is required to compute a tumbling average Knudsen number. The reference length proposed in this work is abbreviated "SM" for *Smallest scaled Median*. SM is defined by taking median lines of the object built from connecting the medians of the faces together, taking the smallest of them, and divide it by π so it matches the diameter of a sphere when the definition is applied to a sphere. An example for a cone-segment is presented Fig. 20 where Lref_{SM} is the length of the "M2" line divided by π. Using the notations from Fig. 16, the expression for a cone is given by Eq.7 and the expression for a cone-segment given by Eq.8. The idea behind this definition is that the flow tends, when considering a tumbling average, to go around the object "choosing" the shortest path. The definition could arguably be applied to any geometries that have "faces" but brings arbitrary choices to be made if a face does not have 4 edges.

$$Lref_{SM}^{cone} = \frac{min(med1, med2)}{\pi} \tag{7}$$

$$Lref_{SM}^{cone-segment} = \frac{min(med1, med3, med4)}{\pi} \tag{8}$$

With:

$$med1 = A + e \tag{9}$$
$$med2 = R_{med} + e\,\cos(\eta)/2 \tag{10}$$
$$med3 = \theta R_{med} + e \tag{11}$$
$$med4 = \theta R_{med} + A \tag{12}$$
$$R_{med} = \frac{0.5(DG + DP) - e\,\cos(\eta)}{2} \tag{13}$$
$$\tag{14}$$

4.3.3. Equivalent radius, Req

The equivalent radius Req used in this work is computed from the reference length, see 4.3.2. It is defined by Eq.15 for all geometries. As a results, for a sphere, it is equal to the radius of the sphere.

$$Req = Lref_{SM}/2 \tag{15}$$

4.4. Approximation model and bridging

The results are presented using drag coefficients for the drag forces and the St number (the ratio between the heat rate and the heat rate in the free molecular regime) for heat rate. Formula for predicting both drag force and heat rate are described in this section. Formulations in the continuum regime, the free molecular regime as well as bridging functions are used. The selected bridging functions are calibrated for spheres because both the reference length and the equivalent radius are calibrated for a sphere. The formula to compute the tumbling average quantities are also described.

4.4.1. Drag force and drag coefficient

The drag coefficient of an object is defined by Eq.16 with $\vec{Force}$ the drag force experienced by the object, ρ_∞ the free-stream density and $\vec{V_\infty}$ the free-stream velocity.

$$CD = \frac{\vec{Force} \cdot \frac{\vec{V_\infty}}{|\vec{V_\infty}|}}{\frac{1}{2}\rho_\infty V_\infty^2 S_{ref}} \tag{16}$$

The bridging function of Blanchard [26], given by Eq.17, is used to compute the drag coefficient of an object CD_{pred} based on both the continuum and the free molecular expressions.

$$CD_{pred} = CD_{cont} + \sin(\pi\frac{3 + log_{10}(Kn)}{10})^{\frac{7-log_{10}(Kn)}{5}}(CD_{FM} - CD_{cont}) \tag{17}$$

The Knudsen number, Kn, is computed from Eq 18 using the mean collision distance of the *hard sphere* model, see Eq.19, where σ_{air} is the mean collision diameter of air. A value of 3.65E-10 meters is used for σ_{air}.

$$Kn = \frac{Lref_{SM}}{\lambda_{HS}} \tag{18}$$

$$\lambda_{HS} = \frac{RT_\infty}{\sqrt{2}\pi P_\infty N_A {\sigma_{air}}^2} \tag{19}$$

Drag coefficient in the free molecular regime CD_{FM}

In the present work, the formula of the drag coefficient for a sphere using the kinetic theory is used. It is taken from [9] Eq 7.71 and copied here for comprehension. The formula is given in Eq.20, where s is the speed ratio (Eq.30), T_{wall} is the temperature of the wall, T_∞ is the temperature of the free-stream conditions and α_{acc} is the accommodation coefficient. In this work, we use $\alpha_{acc} = 1$. Free-stream conditions at an altitude of 110 km ("ML110" in Table 5) are used to evaluate 20 which results in Eq.21.

$$CD_{FM} = \frac{2s^2 + 1}{\sqrt{(\pi)}s^3}e^{-s^2} + \frac{4s^4 + 4s^2 - 1}{2s^4}erf(s) + \frac{2\alpha_T\sqrt{\pi}}{3s}\sqrt{\frac{T_{wall}}{T_\infty}} \tag{20}$$

$$CD_{FM} = 2.005 \tag{21}$$

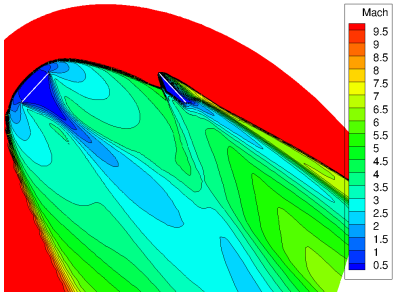


Fig 21. Shock-shock iteration around 30° angle of attack in the continuum regime.

Drag coefficient in the continuum regime CD_{cont}

In the current study a simplification is made for the continuum regime. In literature the drag coefficients for a sphere in hypersonic flow is often cited as 0.92. For random tumbling objects, by selecting the reference surface as the averaged projected surface as described in section 4.3.1, the drag coefficient in random tumbling stays in the order of 1 for all objects [8], except for cases where important auto-shock impingements are observed, see Fig. 21. Such shock impingements cause a significant increase of drag for the given attitude of the object. Once again, such impingements with shocks induced by the object itself, occur only for selected angles. And therefore the impact on the random tumbling average is reduced. The detail of the drag on general shapes in the continuum hypersonic regime is outside the scope of the present work but is presented in [8]. As a results, here, the assumption Eq.22 is used as a baseline for cone and cone-segment.

$$CD_{cont} = 1.0 \tag{22}$$

4.4.2. Heat rate and "St" number

In the present work, for all regimes, the St number is defined by Eq.23 where Q is the heat rate received by the object in Watts and Q_{FM} is an analytical expression of the heat rate received by the object in the free molecular regime. In the literature the Stanton number has an expression similar to Eq.23.But $1/2\rho V^3$ or $\rho C_p V_\infty$ is used for the normalization instead of Q_{FM}. So the St number defined here cannot be called the Stanton number because Q_{FM} is close to but not necessarily equal to $1/2\rho V^3$.

$$St = \frac{Q}{Q_{FM}} \tag{23}$$

The expression of the heat rate in the transitional regime Q_{pred} received by the object in Watts is computed with the function of Matting [25], see Eq. 24, where Q_{FM} is an analytical expression of the heat rate received by the objected in the free molecular regime and Q_{cont} is an analytical expression of the heat rate received by the objected in the continuum regime. The associated St_{pred} number is defined Eq.25.

$$Q_{pred} = Q_{cont}\left(1 - e^{-\frac{|Q_{FM}|}{Q_{cont}}}\right) \tag{24}$$

$$St_{pred} = \frac{Q_{pred}}{Q_{FM}} \tag{25}$$

Heat rate in the free molecular regime Q_{FM}

The heat rate in the free molecular regime Q_{FM} received by the object uses the expression for a sphere from [29] Eq.2.15 and is re-written as Eq.26 because the surface of reference in this work is different. Eq.27 to Eq.30 are also copied here for clarification.

$$Q_{FM} = 4Y \left[\frac{\gamma+1}{2(\gamma-1)}(G+F)\frac{T_{wall}}{T_\infty} - (s^2 + \frac{\gamma}{\gamma-1})(G+F) + \frac{G}{2} \right] S_{ref} \tag{26}$$

$$G = \frac{erf(s)}{4s^2} \tag{27}$$

$$F = \frac{1}{4}\left[1 + \frac{ierfc(s)}{s} - \frac{erf(s)}{2s^2}\right] \tag{28}$$

$$Y = -\alpha_{acc} N k T_\infty V_\infty \tag{29}$$

$$s = \frac{V_\infty}{\sqrt{2RT_\infty}} \tag{30}$$

Heat rate in the continuum regime Q_{cont}

The heat rate received in the continuum regime is based on an analytical model for a sphere. The heat rate per unit of surface at the stagnation point for a "cold wall" sphere in the continuum regime is computed using a DKR – Detra Kemp Riddel – formula [30]. It is defined by Eq.31 with $\rho_{sl} = 1.225(kg/m^3)$ and $V_c = 7802.88(m/s)$.

$$q_{stg}^{CW} = \frac{1.10285E8}{\sqrt{Req}} \sqrt{\frac{\rho_\infty}{\rho_{sl}}} \left(\frac{V_\infty}{V_c}\right)^{3.15} \tag{31}$$

In order to get the heat rate per unit of surface at the stagnation point for a "hot wall" (Eq.32), the previous formula is multiplied by a "hot-wall" correction coefficient that can vary between 0 and 1. It is defined by Eq.33 and Eq.34 with a reference temperature $T_{ref} = 300(K)$ and a fixed thermal capacity of $C_p = 1009(J/K/kg)$ because a perfect gas assumption is used.

$$q_{stg}^{HW} = q_{stg}^{CW} HW_{corr} \tag{32}$$

$$HW_{corr} = \begin{cases} \frac{h_{stg} - C_p(T_{wall})T_{wall}}{(h_{stg} - C_p(T_{ref})T_{Ref})} & \text{if } T_{wall} > T_{ref} \\ 1 & \text{if } T_{wall} \leq T_{ref} \end{cases} \tag{33}$$

$$h_{stg} = V^2/2 + C_p T_\infty \tag{34}$$

The heat rate received in the continuum regime by a random tumbling object is computed by multiplying the previous formula (the heat rate at the stagnation point for a "hot wall" sphere) by 2 quantities that account for the shape effect: the reference surface S_{ref} (see section 4.3.1) and the shape factor K. It results in Eq.35. The "shape factor" K depends on the shape. Similarly to the drag coefficient in the continuum regime, thanks to the specific selection of Req and S_{ref}, the "shape factor" K stays in the order of 1 [8] except for cases where important auto-shock impingements are observed, see Fig. 21. Such shock impingements cause a drastic increase of heat rate for the given attitude of the object. The dependency of the shape factor" K with the geometry is significantly larger than for the drag coefficient.

The details of the heat rate in the continuum hypersonic regime is outside the scope of the present work but is presented in [8]. Eq.36 is used as baseline for cone and cone-segment in the present work.

$$Q_{cont} = q_{stg}^{HW} S_{ref} K \tag{35}$$

$$K = 1.1 \tag{36}$$

4.4.3. Random tumbling coefficients

For each geometry, the random tumbling coefficients are computed by averaging the results of DSMC computations for different attitudes. For objects without symmetry, the Fibonacci sphere distribution is chosen (see Fig. 1.) to discretize the random tumbling motion over 50 attitudes and the 50 DSMC simulations are run accordingly. For axisymmetric shapes, 19 attitudes, using a variation of 10° in angle of attack are considered. Axisymmetric shapes with an additional plane of symmetry perpendicular to the axis of symmetry – e.g. tube – have their number of attitudes reduced to 10. The random tumbling coefficients are computed accounting for each attitude contribution based on the appropriate weight depending on the attitude probability. The tumbling average drag coefficient is computed from Eq.37 where CD_i are computed for each attitude i from Eq.16 and w_i are the associated weights.

$$CD_{tumbling} = \sum_{i \in attitudes} CD_i w_i \tag{37}$$

Similarly, the tumbling average St number is computed with Eq.38 where St_i are computed for each attitude i from Eq.23 and w_i are the associated weights.

$$St_{tumbling} = \sum_{i \in attitudes} St_i w_i \tag{38}$$

For objects without symmetry, a value of 0.02 is used for each of the 50 weights. For axisymmetric configurations the associated weights are not constant but can be computed analytically. The drag force or heat rate are assumed to be constant over the range $[AoA - 5^\circ, AoA + 5^\circ]$ The resulting weight for each angle of attack (AoA) are given by Eq.39 to Eq.41 with $d = 5^\circ$ (half the 10° resolution). Numerical values are given Table 7.

$$w_i = (sin(b_i) - sin(a_i))/2 \tag{39}$$

$$b_i = \begin{cases} AoA_i + d & \text{if } AoA_i \neq \pi/2 \\ AoA_i & \text{if } AoA_i = \pi/2 \end{cases} \tag{40}$$

$$a_i = \begin{cases} AoA_i + d & \text{if } AoA_i \neq -\pi/2 \\ AoA_i & \text{if } AoA_i = -\pi/2 \end{cases} \tag{41}$$

4.5. Results

After explaining how to obtain heat rate and the drag force for random tumbling object from the approximation model, the comparison between the approximation model and the DSMC data for the 105 geometries is presented.

4.5.1. Methodology to obtain heat rate and the drag force of random tumbling object

In order to obtain the heat rate and the drag force of a random tumbling object that has the temperature T_{wall} and experiencing the free-stream conditions $(V_\infty, \rho_\infty, T_\infty, P_\infty)$, the geometric quantities S_{ref}, $Lref_{SM}$ and Req of the object of interest shall first be computed as described in 4.3.1, 4.3.2 and 4.3.3. The expression Q_{pred} for the heat rate received by the object is computed from Eq.24. Both Q_{FM} and Q_{cont} are computed using the same free-stream conditions $(V_\infty, \rho_\infty, T_\infty, P_\infty)$ and the wall temperature T_{wall}. The expression of the drag coefficient CD_{pred} for the same object is computed from Eq.17. Using the values CD_{FM} and CD_{cont} for a sphere, i.e. Eq.22 and Eq.21 respectively is a good first approximation. All the formulation described here can be applied to more simple shapes such as cylinder or box provided that the geometry quantities (reference surface, reference length and equivalent radius) are changed accordingly. Indeed, similar results have been observed for box and cylinder but are not reported here.

4.5.2. Comparison between DSMC and the approximation model

The comparison for random tumbling objects are presented Fig. 22 for drag coefficient and Fig. 23a for St number. The two numbers St and CD are referred as QOI (Quantity Of Interest) in this section and are referring to the random tumbling quantities. A close up is available Fig. 25 and Fig. 26 respectively where the nomenclature to label each geometry matches the description of Fig27 and Fig28. On Fig. 22, Fig. 23a, Fig. 25 and Fig. 26, the data associated with each geometry is represented by a point. Each bridging functions is represented by a line. Fig. 25 and Fig. 26 illustrate how the 2 QOIs tend to be close to the bridging functions of the literature even cones and cone-segments are not blunt bodies. Both QOIs get closer to the free molecular values for geometries that have the highest Knudsen number. But none of the QOIs reach a values higher than the free molecular limit. The choice of S_{ref} influences the vertical position of the value of the 2 QOIs, while the definition of $Lref$ influence their horizontal position in the plots. The specific choices of the geometric reference quantities help narrowing down the scattering of the QOIs for different geometries.

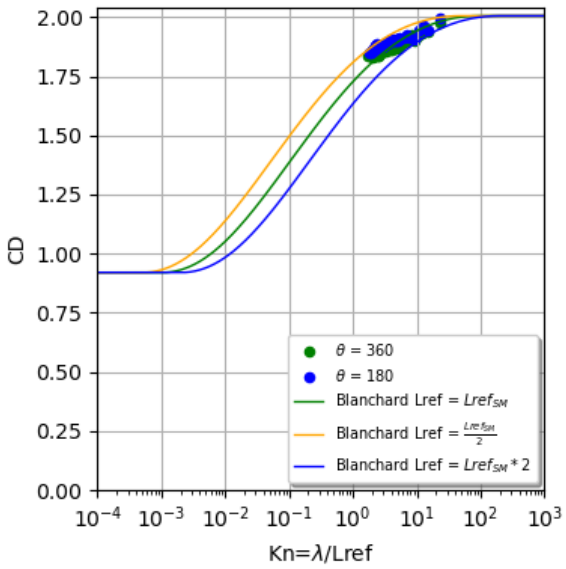


Fig 22. Drag coefficient as a function of the Knudsen number. DSMC data is represented by points and approximation models are represented by lines

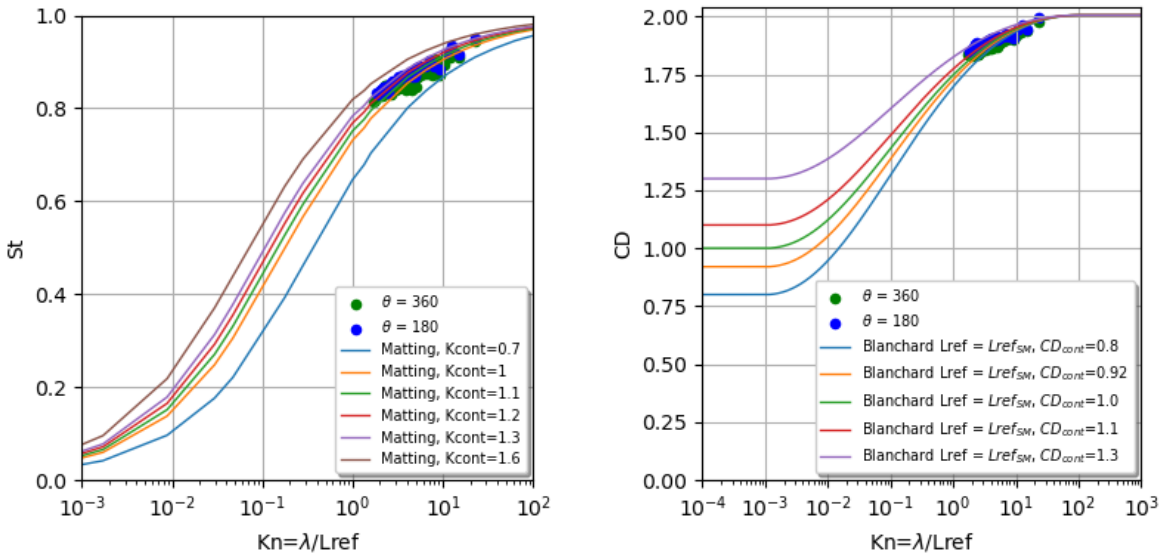
Regarding the drag coefficient presented Fig.22 and Fig.25, for random tumbling objects, the cloud of point closely follows the green line and the difference between the DSMC data (points) and the approximation model (lines) is quantified Table 2. The average of the relative difference between the prediction (Eq.17) of the proposed model ($Lref = Lref_{SM}$, green line) and the DSMC data is -0.6%

with a standard deviation of 1.27%. Multiplying or dividing the reference length by a factor two changes the average error prediction by around 3%.

Table 2. Relative difference between the 105 predictions of the approximation model and the DSMC data: $\delta = (CD_{pred} - CD_{DSMC})/CD_{DSMC}$, for different values of $Lref$

	$Lref = Lref_{SM}$	$Lref = Lref_{SM}/2$	$Lref = 2Lref_{SM}$
mean	-0.6	2.2	-4.2
std	1.3	0.9	1.8
min	-3.1	0.3	-7.3
max	1.8	3.8	-0.7
max-min	4.9	3.6	6.6

Regarding the St number presented Fig.23a and Fig.26 for random tumbling objects, the cloud of point follows less closely the lines when compared to what is observed for the drag coefficient CD. The difference between the DSMC data (points) and the approximation model (lines) is quantified in Table 4. The range of K values investigated in the table is representative of values computed through CFD [8]. The choice $K = 1.1$ (green line) minimize the differences between the predictions (Eq.25) and the DSMC data. But it shall be noted than any values from $K = 1.0$ until $K = 1.3$ give a prediction error with a maximum amplitude of 2.1% and a maximum standard deviation of 2.3%.



(a) Random tumbling St number as a function of the Knudsen number

(b) Random tumbling CD as a function of the Knudsen number

Fig 23. Random tumbling coefficient as a function of the Knudsen. DSMC data is represented by points and approximation models are represented by lines

4.6. Discussions

The impact of the shape factor K and of the drag coefficient in the continuum regime CD_{cont} on the accuracy of the approximation models are discussed before enumerating some of the uncertainties of the method.

4.6.1. Impact of CD_{cont}

The effect of the value of CD_{cont} on the prediction of the drag coefficient in random tumbling CD_{pred} decreases with an increase in Knudsen number as illustrated by Fig.23b. The impact of the value of CD_{cont} on the prediction is summarized Table 3. The range of CD_{cont} values investigated in that table is representative of values computed through CFD [8]. For example, switching from $CD_{cont} = 0.92$ to $CD_{cont} = 1.1$ ($+20$% in CD in the continuum regime) shifts the average difference of 1.2%, about 16 times less.

Table 3. Relative difference between the 105 predictions of the approximation model and the DSMC data : $\delta = (CD_{pred} - CD_{DSMC})/CD_{DSMC}$, for different values of CD_{cont}

	$CD_{cont} = 0.8$	$CD_{cont} = 0.92$	$CD_{cont} = 1.0$	$CD_{cont} = 1.1$	$CD_{cont} = 1.3$
mean	-1.4	-0.6	-0.1	0.6	1.9
std	1.5	1.3	1.1	1.0	0.8
min	-4.2	-3.1	-2.4	-1.5	-0.2
max	1.4	1.8	2.0	2.4	3.4
max-min	5.6	4.9	4.5	3.9	3.5

4.6.2. Impact of the shape factor K

The value of K linearly impacts the heat rate in the continuum regime Q_{cont} (Eq.35), however its impact on the accuracy of the prediction of the heat rate of random tumbling object in the transitional regime is less severe as demonstrated Table 4. For example, switching from $K = 1$ to $K = 1.6$ ($+60$% in heat rate in the continuum regime) shifts the average difference of 6%, a 10-fold reduction. Obviously the impact of the error of K on the error prediction of the heat rate will decrease with an increase in Kundsen number.

Table 4. Relative difference between the 105 predictions of the approximation model and the DSMC data: $\delta = (St_{pred} - St_{DSMC})/St_{DSMC}$, for different values of K.

	K=0.7	**K=1.0**	**K=1.1**	**K=1.2**	**K=1.3**	**K=1.6**
mean	-7.8	-1.5	-0.1	1.1	2.1	4.5
std	3.6	2.3	2.0	1.8	1.6	1.5
min	-14.5	-5.9	-3.9	-2.3	-0.9	1.4
max	-1.1	3.5	4.3	5.3	6.2	8.6
max-min	13.4	9.4	8.2	7.6	7.1	7.2

4.6.3. Discussion on the uncertainties

Different types of uncertainties exist. First, the interaction of each particle with the wall may not be modeled properly. Arguably, using a more realistic accommodation coefficient, $\alpha_{acc} = 0.8$ instead of $\alpha_{acc} = 1$, could change the results. However, this can also be accounted for in Eq.20 and in Eq.26. Regarding the numerical uncertainties, numerical convergence of each DSMC computation is demonstrated in 4.2 and is highlighted Fig.24. Uncertainties are below 1%. Finally, only 50 attitudes (or 19 or 10 depending symmetries) are used to model the random tumbling of the object. Even this uncertainty has not been quantified for DSMC computation it has been investigated in the continuum regime [8]. The corresponding uncertainty in the transitional regime is expected to be lower than in the continuum regime because the variation of the QOIs with the attitude are more significant in the continuum regime than in the higher part of the transitional regime. This is highlight in Fig. 24 where the variation of drag and heat rate with the attitude are relatively smooth. A possible explanation is that the effect of auto-shock impingements that affects the continuum regime (see Fig. 21) will be spread out as shocks become more diffuse with the rarefaction of air.

5. Conclusion

A DSMC code is described along with its validation. It is used to carry out thousands of simulations on cones and cone-segments across a large range of geometry parameters. The reference surface and the reference length used to compute the non-dimensional quantities – the St number and the drag coefficient – lead to an average prediction error of the random tumbling heat rate and drag force of less than 1%. This is achieved by calibrating the bridging function with a shape factor $K = 1.1$ in the continuum regime and with a drag coefficient of $CD_{cont} = 1.0$ in the continuum regime. However the error on the prediction can be more significant if shape specific value for calibrating the bridging function in the continuum regime are used. Those values (K and CD_{cont}) are needed so prediction errors can be quantified accurately. The formulations presented are not specific to cones and cone-segments and can be applied to various shapes. Future work should be extended to other shapes, other kind of tumbling, variation in accommodation coefficient and lower altitude with/without reacting flow while it is significantly more numerically challenging.

Acknowledgments

The development of the DSMC solver has been done thanks to WBSO program from the Netherlands in 2014 and 2015. Part of the work on the determination of the reference quantities has been financed under contract n° 170990 from CNES.

References

[1] P. Omaly, C. Magnin Vella, and S. Galera, "DEBRISK, CNES Tool for Re-Entry Survivability Prediction: Validation and Sensitivity Analysis," in *Safety is Not an Option, Proceedings of the 6th IAASS Conference* (L. Ouwehand, ed.), vol. 715 of *ESA Special Publication*, p. 76, Sept. 2013.

[2] e. a. R. Kanzler, "Upgrade of drama's spacecraft entry survival analysis codes, 4th international workshop on space debris reentry," tech. rep., Darmstadt, 2018.

[3] C. S. C. Ostrom, "Orsat modelling and assessment, 4 th international workshop on space debris reentry," tech. rep., Darmstadt, 2018.

[4] J. Beck, I. Holbrough, J. Merrifield, N. Joiner, and S. Bainbridge, "Progress in hybrid spacecraft/object oriented destructive re-entry modelling using the sam code," in *7th European conference on space debris, Darmstadt*, 2017.

[5] R. D. Klett, "Drag coefficients and heating ratios for right circular cylinders in free-molecular and continuum flow from mach 10 to 30," tech. rep., Sandia Corp., Albuquerque, N. Mex., 1964.

[6] L. Cropp, "Analytical methods used in predicting the re-entry ablation of spherical and cylindrical bodies," tech. rep., Sandia Corp., Albuquerque, N. Mex., 1965.

[7] J. Annaloro, S. Galera, C. Thiebaut, and P. Omaly, "Debrisk v3: new generation of object-oriented tools," in *8th European Conference on Space Debris*, ESA Space Debris Office, 2021.

[8] P. V. Hauwaert, M. Spel, S. Galera, J. Annaloro, and P. Omaly, "Numerical methodology for analytic formulation of drag and heat rates of different objects re-entering the earth atmosphere in a prescribed motion," 2nd International Conference on High-Speed Vehicle Science Technology (HISST), 2022.

[9] G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Oxford University Press, 1994.

[10] G. A. Bird, "The DS2v/3v program suite for DSMC calculations," in *AIP Conference Proceedings*, AIP, 2005.

[11] C. White, M. Borg, T. Scanlon, S. Longshaw, B. John, D. Emerson, and J. Reese, "dsmcFoam: An OpenFOAM based direct simulation monte carlo solver," *Computer Physics Communications*, vol. 224, pp. 22–43, Mar. 2018.

[12] S. J. Plimpton, S. G. Moore, A. Borner, A. K. Stagg, T. P. Koehler, J. R. Torczynski, and M. A. Gallis, "Direct simulation monte carlo on petaflop supercomputers and beyond," *Physics of Fluids*, vol. 31, p. 086101, Aug. 2019.

[13] D. J., C. E., S. M., L. T., K. R., P. A., B. M., H. G., S. C., S. S., L. S., and A. J., "Rebuild and data exploitation of the avum re-entry event for break-up model development," in *2nd International Conference on Flight Vehicles, Aerothermodynamics and Re-entry Missions Engineering (FAR)*, 2022.

[14] P. V. Hauwaert, E. Constant, M. I., M. Spel, P. Makowski, V. Ledermann, J. Dumon, S. Galera, and J. Annaloro, "Pampero v3, a spacecraft-oriented reentry analysis code," in *2nd International Conference on Flight Vehicles, Aerothermodynamics and Re-entry Missions Engineering (FAR)*, 2022.

[15] V. K. G. Karypis, "Metis: A software package for partitioning unstructured graphs. partitioning meshes, and computing fill-reducing orderings of sparse matrices, version 4.0," tech. rep., University of Minnesota, Department of Computer Science / Army HPC Research Center Minneapolis, MN 55455, The address of the publisher, 1999.

[16] G. A. Bird, "Sophisticated dsmc", notes prepared for a short course at the dsmc07 meeting santa fe," tech. rep., 2007.

[17] D. Gao, C. Zhang, and T. Schwartzentruber, "A three-level cartesian geometry-based implementation of the DSMC method," in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, Jan. 2010.

[18] G. LeBeau, "A parallel implementation of the direct simulation monte carlo method," *Computer Methods in Applied Mechanics and Engineering*, vol. 174, pp. 319–337, May 1999.

[19] G. LeBeau, K. Jacikas, and F. Lumpkin, "Virtual sub-cells for the direct simulation monte carlo method," in *41st Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, Jan. 2003.

[20] J. M. Burt, E. Josyula, and I. D. Boyd, "Novel cartesian implementation of the direct simulation monte carlo method," *Journal of Thermophysics and Heat Transfer*, vol. 26, pp. 258–270, Apr. 2012.

[21] A. Lofthouse, *Non-equilibrium hypersonic aerothermodynamics using the Direct Simulation Monte Carlo and Navier-Stokes model*. PhD thesis, University of Michigan, 2008.

[22] J. Allegre, D. Bisch, and J. C. Lengrand, "Experimental rarefied aerodynamic forces at hypersonic conditions over 70-degree blunted cone," *Journal of Spacecraft and Rockets*, vol. 34, pp. 719–723, Nov. 1997.

[23] D. Gao, C. Zhang, and T. E. Schwartzentruber, "Particle simulations of planetary probe flows employing automated mesh refinement," *Journal of Spacecraft and Rockets*, vol. 48, pp. 397–405, May 2011.

[24] T. D. Holman, *Numerical investigation of the effect of Continuum Breakdown on Hypersonic Vehicle Surface Properties*. PhD thesis, University of Michigan, 2010.

[25] F. W. MATTING, "Approximate bridging relations in the transitional regime between continuum and free- molecule flows," *Journal of Spacecraft and Rockets*, vol. 8, pp. 35–40, Jan. 1971.

[26] R. C. Blanchard, "Rarefied-flow aerodynamics measurement experiment on the aeroassist flight experiment vehicle," *Journal of Spacecraft and Rockets*, vol. 28, pp. 368–375, July 1991.

[27] B. MELTZER, "Shadow area of convex bodies," *Nature*, vol. 163, pp. 220–220, Feb. 1949.

[28] C. P. Jr. Albert, "Configuration factor for exchange of radiant energy between axisymmmetrical section of cylinder, cones, and hemispheres and their bases," tech. rep., NASA Technical Note D-944, Langley Research Center, 1961.

[29] A. K. OPPENHEIM, "Generalized theory of convective heat transfer in a free-molecule flow," *Journal of the Aeronautical Sciences*, vol. 20, pp. 49–58, Jan. 1953.

[30] F. R. Detra R.W., Kemp N.H., "Addendum to 'heat transfer to satellite vehicles re-entering the atmosphere," *Journal of Jet Propulsion*, vol. 27, no. 12, pp. 1256–1257, 1957.

A. Annexes

Table 5. Species parameters. Viscosities are given for a reference temperature $T_{ref} = 273$ K

Name	Symbol	Mass [kg]	Diameter at T_{ref} [m]	Viscosity [N/s/m2]	Viscosity index
Argon	Ar	6.63e-26	4.11e-10	2.12e-5	0.81
Nitrogen	N2	4.65e-26	4.17e-10	1.66e-5	0.74
Oxygen	O2	5.31e-26	4.07e-10	1.81e-5	0.77
Atomic oxygen	O	2.65e-26	2.73e-10	2.96e-5	0.699

Table 6. Free-stream conditions and wall temperature

Name	Speed [m/s]	Temperature [K]	Density [kg/m3]	Wall Temperature [K]
ML110	7500	247	9.64e-8	350
M20.2 plate	1502	13.3	1.73e-5	290
M20.2 Planetary probe	1502	13.3	1.73e-5	300

Table 7. Weight for axisymmetric configuration as a function of the angle of attack

AoA (degree)	w_i
-90	0.0019
-80	0.0151
-70	0.0298
-60	0.0436
-50	0.0560
-40	0.0668
-30	0.0755
-20	0.0819
-10	0.0858
0	0.0872
10	0.0858
20	0.0819
30	0.0755
40	0.0668
50	0.0560
60	0.0436
70	0.0298
80	0.0151
90	0.0019

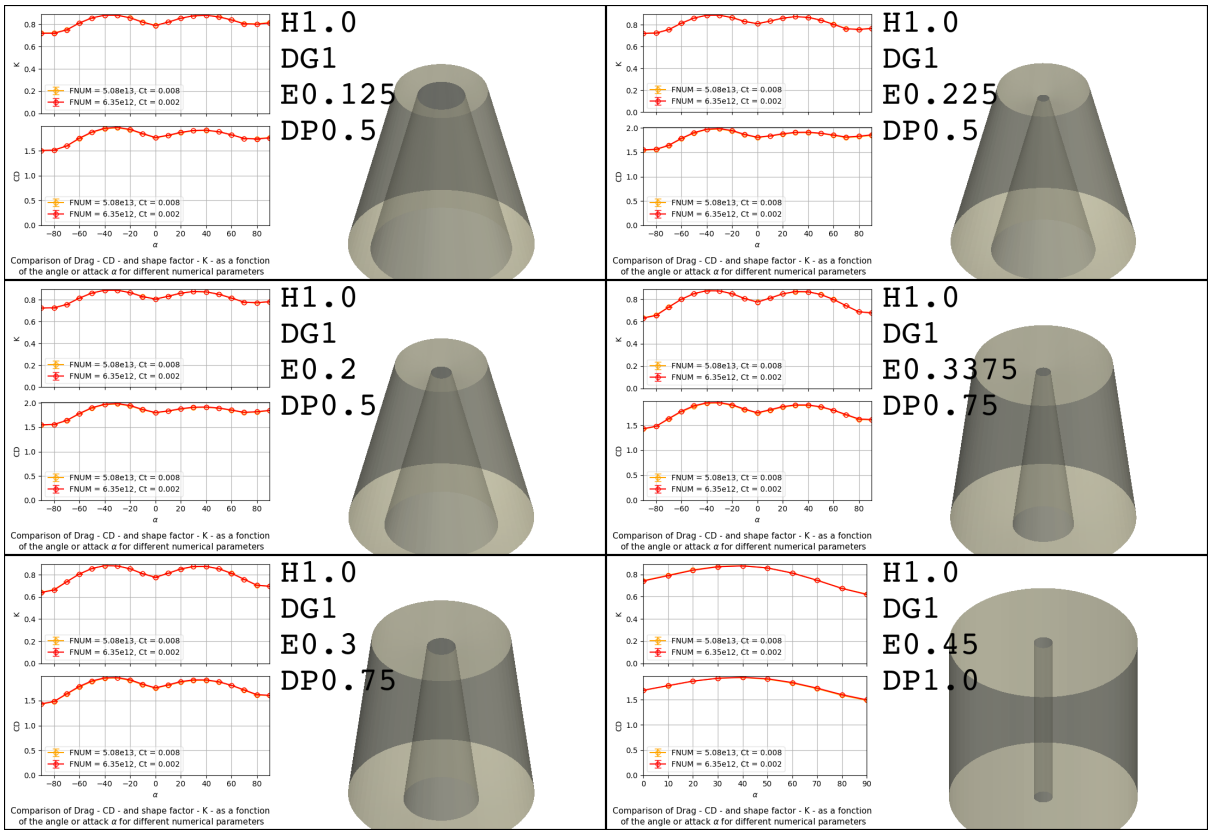


Fig 24. Variation in shape factor and drag coefficient due to the number of simulation particles and the time step.

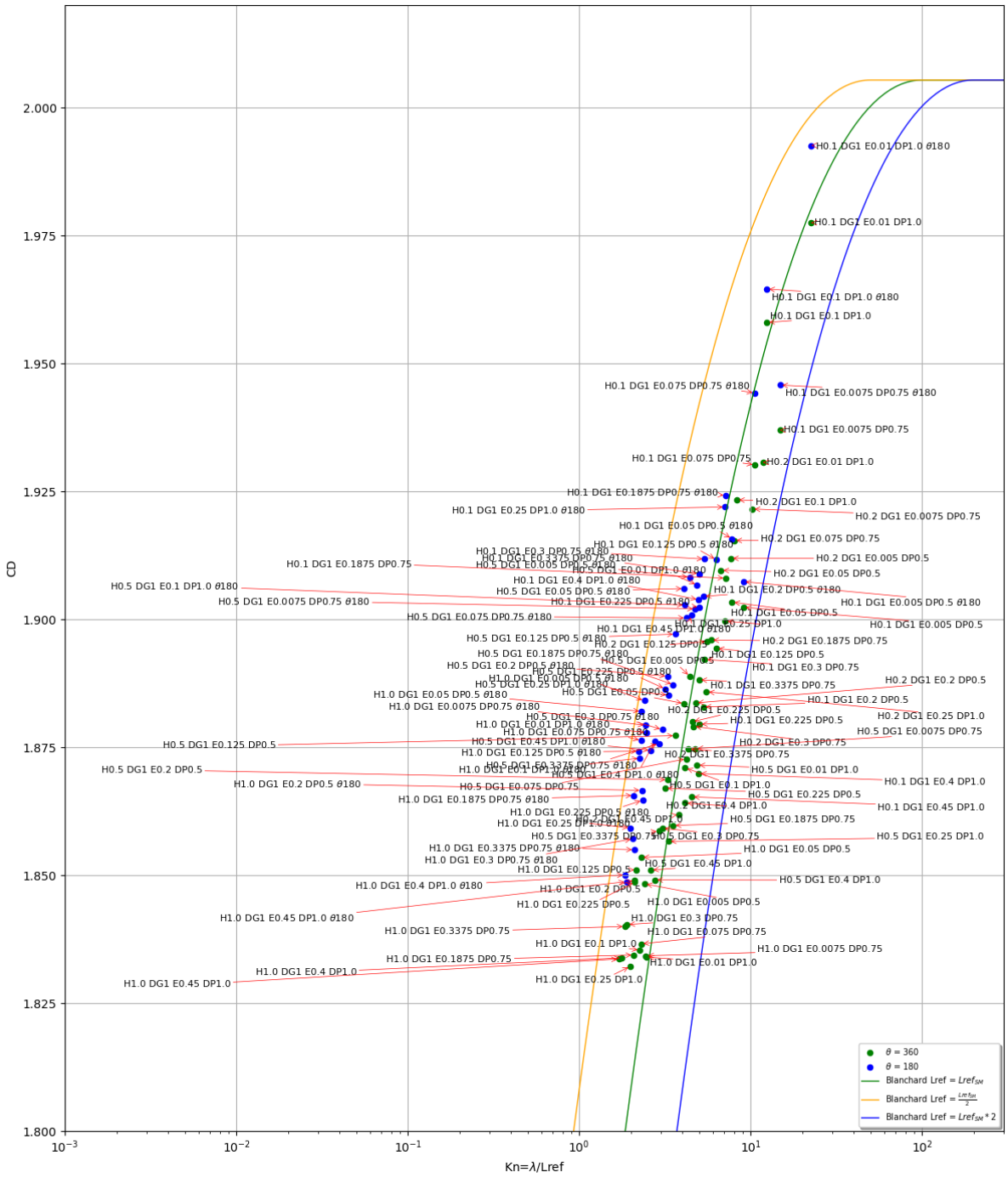


Fig 25. Drag coefficient as a function of the Knudsen number, including geometry labels. DSMC data is represented by points and approximation models are represented by lines

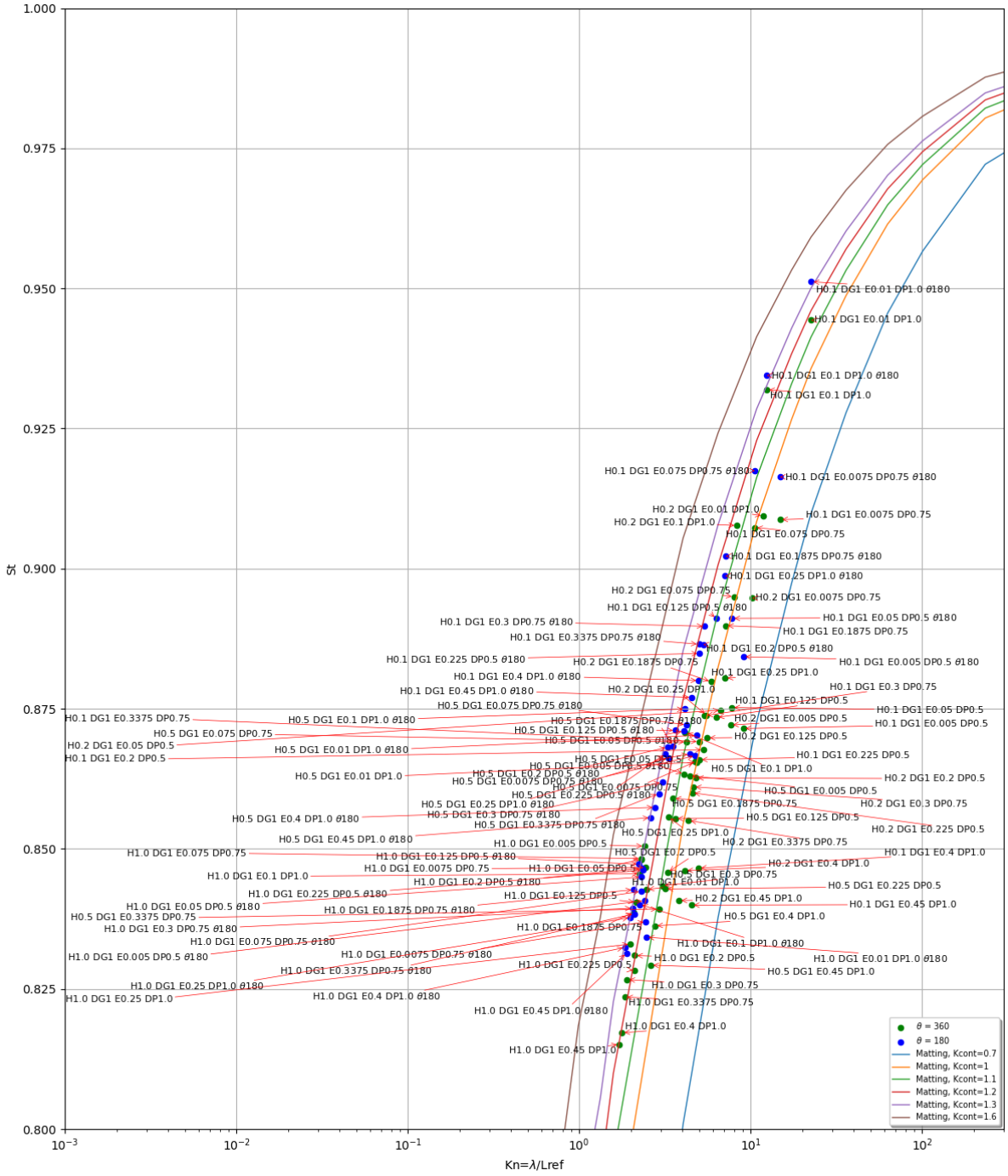


Fig 26. St number as a function of the Knudsen number, including geometry labels. DSMC data is represented by points and approximation models are represented by lines

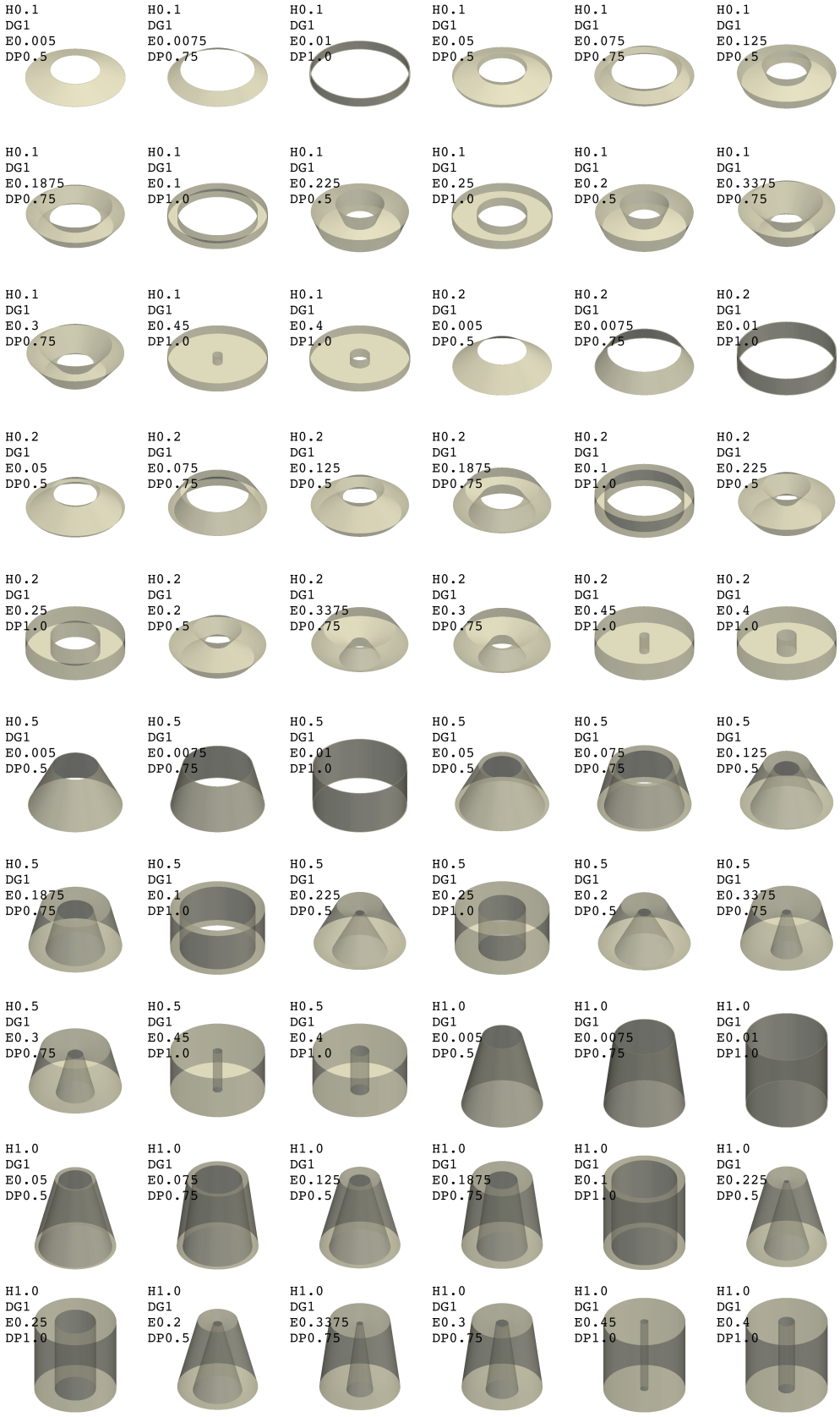


Fig 27. Drawing of cones

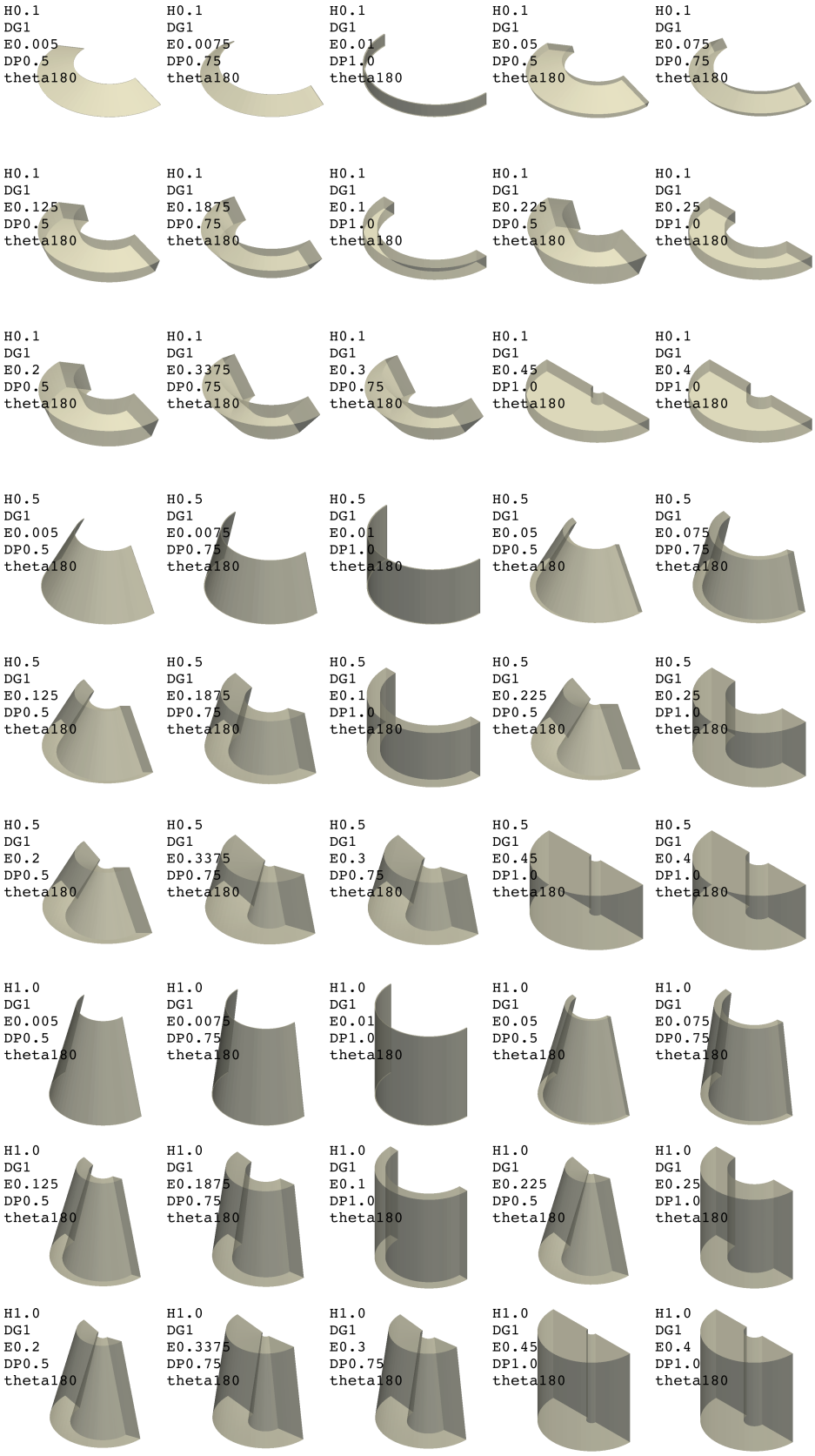


Fig 28. Drawing of cone-segments.