DOI: 10.13009/EUCASS2023-350
Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS

# A cloud-based multi-fidelity solution for debris re-entry prediction

*Thibault Bridel-Bertomeu*[⋆†] *and Agnes Chan*[⋆]
[⋆]*Re CAE*
*Bassins, Switzerland*
thibault.bridelbertomeu@re-cae.com · agnes.bridelbertomeu@re-cae.com
[†]Corresponding author

## Abstract

The proliferation of space debris has become an escalating crisis, yearning an urgent need for comprehensive mitigation strategies. One prominent strategy for space debris mitigation involves the deliberate planning of post-mission disposal ensuring re-entry of spacecrafts into Earth's atmosphere. To ensure effective disposal, re-entry predictions through advanced atmospheric modeling and improved technologies need to be enhanced. In this work, we will present refined models on a spacecraft-oriented platform developed by Re CAE that allows to run multi-fidelity exo- and endo-atmospheric assessments. This cloud-based solution offers the possibility to run from a web browser on any computer, orbital propagation and collision risk assessment of a spacecraft during its operational phase. The second part of the tool encapsulates a six degree of freedom analysis to predict re-entry trajectory and a higher fidelity coupled aero-thermal CFD simulation of a space debris. This suite is designed to leverage CPU instances found on the cloud via Kubernetes to offer more flexibility and accessibility. Beyond the classical application of such a suite to the assessment of debris re-entry, the design choices selected here also open the door to browser-based uncertainty quantification campaigns for re-entering vehicles. Validation computations have been carried out on wide variety of objects and results show good consistency with data found in the literature.

## 1. Introduction

Since the launch of the first artificial satellite, Sputnik 1, in 1957, the proliferation of space debris has become a growing concern for space missions and satellite operations. The increasing number of satellites and other objects launched into space has led to a cluttered environment in Earth's orbit.[13] This accumulation of space debris, which includes defunct satellites, spent rocket stages, and fragments from disintegration and collisions, poses a significant threat to operational spacecrafts, operational aircrafts if the debris re-enter the atmosphere and even to human lives on Earth. Worst, the density of space debris increases the likelihood of collisions, which in turn generates more debris, creating a cascade effect known as the Kessler Syndrome.[21]

In response to the escalating issue of space debris, there is a pressing need for innovative software and strategies to limit its proliferation. One such approach is the development of tools that can act as digital twins for the lifecycle of spacecraft, aiding in the design, operation, and eventual controlled re-entry or disposal of these objects. The platform put forward by Re CAE enters that category and allows any user, from satellite manufacturers to space agencies, to perform a wide range of analyses on their spacecrafts, from the design phase to the operational phase to the end-of-life phase.

The three main software components of the platform, **Re·Propagate**, **Re·Entry** and **Re·CFD**, are described in the following paragraphs with a focus on the underlying numerical methods, the performance of the software components and an illustration of the results. The platform is designed to be completely hosted on the cloud, hence the discussion ends with the presentation of the chosen cloud architecture for Re CAE platform.

## 2. Exo-atmospheric flight

Re CAE platform offers a wide range of tools to support the life cycle of space objects, the first of which covers the exo-atmospheric flight phase. For *e.g.* an artificial satellite, this phase starts as soon as the satellite is positioned and spans the entire duration of its mission. It therefore encompasses the propagation of the satellite orbit, as well as the always present collision risks assessment.

DOI: 10.13009/EUCASS2023-350

The component of the platform that ensures all exo-atmospheric computations is **Re·Propagate**, a software developed by Re CAE as a Python shell around a high-performance low-level kernel. It leverages both the latest CPU-intrinsic parallelisation capabilities as well as the high-level multiprocessing capabilities offered by Ray[29] to ensure optimal performance.

### 2.1 Orbital Propagation

Orbital propagation is the process of computing the position and velocity of a spacecraft at a given time. Intrinsically, it is based on Newton's second law, but in practice, the implementation of such a method can go from accounting only for the gravitational potential of the Earth to including the gravitational potential of the Sun, the Moon, and other planets, as well as the atmospheric drag, the solar radiation pressure, and other perturbations.[35]

Mathematically, the problem amounts to solving the set of ordinary differential equations described below in equation (1) that characterizes the motion of a solid with 3 degrees of freedom (d.o.f.). Note that by nature, as **Re·Propagate** is meant to compute stable or slowly-decaying orbits around the Earth, the equations of motion are given in an inertial frame in the Geocentric Celestial Reference System (GCRS).[20]

$$\frac{d\vec{x}_{\mathrm{ECI}}}{dt} = \vec{V}_{\mathrm{ECI}}\,, \quad m_{\mathrm{vehicle}}\frac{d\vec{V}_{\mathrm{ECI}}}{dt} = \vec{F}_{\mathrm{grav}} + \vec{F}_{\mathrm{atmosphere}} + \vec{F}_{\mathrm{radiation}}. \tag{1}$$

As a first approximation, only the Moon, the Sun and Jupiter are taken into account as gravitationally perturbating third bodies and only the Sun is taken into account as a source of radiation pressure.

The gravitational pull of the Earth on the spacecraft is modeled in **Re·Propagate** by the latest XGM2019e (see Zingerle *et al.*[41]) model. Its formulation in terms of spherical harmonics (see equation (2)) allows to go from a simple central gravity field to a highly detailed model with an accuracy of about 4km:

$$\vec{F}_{\mathrm{grav}} = -m_{\mathrm{vehicle}}\nabla_{\mathrm{ECI}}\mathcal{U} \tag{2}$$

where

$$\mathcal{U} = -\frac{\mu}{a}\sum_{n=0}^{\infty}\sum_{k=0}^{n} C_{nk}\mathfrak{V}_{nk} + S_{nk}\mathfrak{W}_{nk}, \tag{3}$$

with $\mu$ the gravitational parameter of the Earth, $a$ the semi-major axis of the planet ellipsoid, $C_{nk}$ and $S_{nk}$ the spherical harmonic coefficients, and $\mathfrak{V}_{nk}$ and $\mathfrak{W}_{nk}$ the fully normalized associated Legendre functions of degree $n$ and order $k$ (see Vallado[35] for more details). For long-time integration and to obtain a preliminary solution, such a formulation can also be reduced to a simple zonal gravity field where only the $J_2 = -C_{20}$ and $J_3 = -C_{30}$ perturbation terms are considered. Naturally, the logic behind the implementation of such a model holds in the possibility to also run simulations with a more detailed model, to account for the finest drift in the orbit of the spacecraft.

As for $\vec{F}_{\mathrm{atmosphere}}$ and $\vec{F}_{\mathrm{radiation}}$, more complex models will be detailed when discussing atmospheric re-entry, but for orbital propagation we follow Vallado,[35] Curtis,[10] Bate,[4] etc. and assume simplistic formulations. The atmospheric drag force is modeled using a balistic approximation wherein the atmospheric density is obtained with the simplified US Standard Atmosphere of 1976[3] which is justified by the low variability of density with altitude above the 100km line (see section 3.1.3 for a more detailed discussion about atmospheric models):

$$\vec{F}_{\mathrm{atmosphere}} = -\frac{1}{2}\rho_{\mathrm{COESA76}}||\vec{V}_{\mathrm{ECI}}||^2 C_D A_{\mathrm{atmos}}\frac{\vec{V}_{\mathrm{ECI}}}{||\vec{V}_{\mathrm{ECI}}||} \tag{4}$$

where $\rho_{\mathrm{COESA76}}$ is the density obtained with the US76 model, $C_D$ the drag coefficient and $A_{\mathrm{atmos}}$ the cross-sectional area of the spacecraft exposed to atmospheric effects. The perturbing force due to the solar radiation pressure is modeled following the cannonball model:

$$\vec{F}_{\mathrm{radiation}} = -\nu P_{\mathrm{rad}} C_R A_{\mathrm{rad}}\hat{\mathbf{u}}, \tag{5}$$

where $\nu$ is the shadow switch, $P_{\mathrm{rad}}$ is the solar radiation pressure at 1 AU, $C_R$ is the radiation pressure coefficient, $A_{\mathrm{rad}}$ is the cross-sectional area of the spacecraft exposed to solar radiation pressure and $\hat{\mathbf{u}}$ is the unit vector pointing from the spacecraft to the Sun. Note that if one tries to equate equation (4) and equation (5) and solve for the altitude, one obtains the altitude of the transition from the dominance of the perturbative effect of atmospheric drag to that of solar radiation pressure - about 625 km - which is the altitude below which the computation of the solar radiation pressure perturbation is disabled in **Re·Propagate**.

DOI: 10.13009/EUCASS2023-350

An illustration of the capabilities of **Re·Propagate** is given on figure 1. The orbit of the ENVISAT satellite is propagated over the span of 200 years, from 2023 to 2223, using different levels of modeling complexity: a comparison between the simple Keplerian orbit, an orbit affected by the oblateness of the Earth, an orbit affected by the lunar attraction and an orbit affected by the atmospheric drag is given. For the latter, the comparison is refined further and three different possible values of the drag coefficient are investigated, from $C_D = 5.0$ to $C_D = 10.0$, based on a loose interpretation of the results presented by Weigel *et al.*[38] The (expected) results show that the atmospheric drag is the dominant perturbation over the span of 200 years, and that the drag coefficient has a significant impact on the orbital decay rate, with a predicted lithobrake event happening in 2214 for $C_D = 5.0$ and in 2118 for $C_D = 10.0$.
The restitution times of the different models are given in table 1.

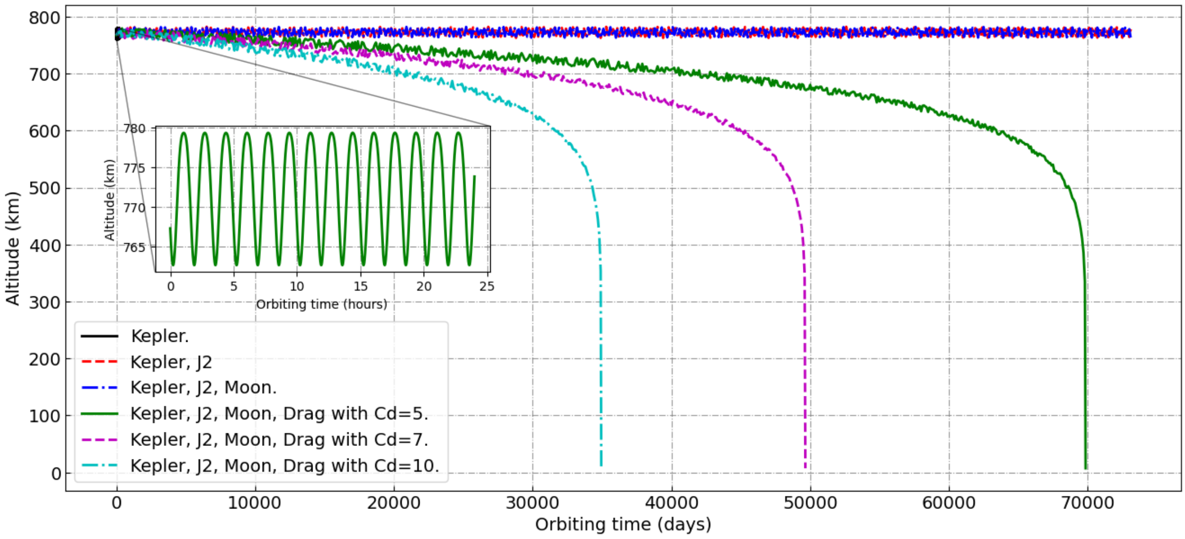


Figure 1: Orbital propagation of the ENVISAT satellite over the span of 200 years. Multiple models are compared, from the simple Keplerian orbit to a more complex modeling that accounts for the oblateness of the Earth, the gravitational pull of the Moon and the atmospheric drag. To better illustrate the effect of the atmospheric drag, the drag coefficient is varied from 5.0 to 10.0 (based on a loose interpretation of Weigen *et al.*[38]).

| Model | CPU time [s] | Performance [$\mu$s/it] | Lithobrake year |
|---|---|---|---|
| Kepler | 347 | 2.2 | *None* |
| Kepler + $J_2$ | 405 | 2.4 | *None* |
| Kepler + $J_2$ + Moon | 476 | 3.0 | *None* |
| Kepler + $J_2$ + Moon + drag ($C_D = 10$) | 234 | 3.1 | 2118 |
| Kepler + $J_2$ + Moon + drag ($C_D = 7$) | 331 | 3.1 | 2159 |
| Kepler + $J_2$ + Moon + drag ($C_D = 5$) | 486 | 3.16 | 2214 |

Table 1: Summary of the results for the orbital propagation comparative study of the ENVISAT satellite. The CPU time is given for a single processor of an Apple M1 Pro chip from 2021. For reference, the computation "Kepler + $J_2$ + Moon + drag ($C_D = 5$)" conducted using the public library `poliastro`[32] took about 20 hours to complete on the same machine.

### 2.2 Collision risk assessment

The first level of space debris mitigation comes during the lifecycle of the spacecrafts. The first step is indeed to ensure that the spacecrafts are not put in a situation where they could collide with other objects, and therefore it is paramount to be able to assess the collision risk of a given object with other objects.
To do so, the **Re·Propagate** kernel is also equipped to propagate the uncertainty on the position and velocity of the spacecrafts during their orbits, which is a key feature to assess the collision risk between objects. The uncertainty propagation is marred with the same numerical error as the propagation of the orbit itself, because it is based on the same numerical scheme: only the vector of unknowns and the formulation of the system of differential equations changes. As a reminder, let us re-write in a state formulation the equations of motion:

DOI: 10.13009/EUCASS2023-350

$$\frac{d}{dt}\vec{S}_{\mathrm{ECI}} = \vec{F}(\vec{S}_{\mathrm{ECI}}, t) \tag{6}$$

where $\vec{S}_{\mathrm{ECI}} = [\vec{X}_{\mathrm{ECI}}, \vec{V}_{\mathrm{ECI}}]$ and $\vec{F}$ represents the sum of the external forces acting on the spacecraft. It is also possible to introduce the notion of Jacobian matrix of the system, $\mathbb{J}_{\mathrm{ECI}} = d\vec{F}_{\mathrm{ECI}}/d\vec{S}_{\mathrm{ECI}}$, which is the $6 \times 6$ matrix of partial derivatives of the system of differential equations with respect to the state vector.

If one then considers the uncertainty on the position and the velocity of the spacecraft in the ECI frame at epoch $t_0$, it can be represented by a $6 \times 6$ covariance matrix, say $\mathbb{C}_{\mathrm{ECI}}(t_0)$. To propagate that matrix, one can use the state transition theory[18,35] that states that the covariance matrix at epoch $t_1$ is given by:

$$\mathbb{C}_{\mathrm{ECI}}(t_1) = \mathbb{S}_{\mathrm{ECI}}(t_1)\mathbb{C}_{\mathrm{ECI}}(t_0)\mathbb{S}^T_{\mathrm{ECI}}(t_1), \tag{7}$$

where $\mathbb{S}$ is the state transition matrix defined as the differential evolution of the state between two epochs, *i.e.* here between $t_0$ and $t_1$ we have $\mathbb{S}_{\mathrm{ECI}}(t_1) = d\vec{S}_{\mathrm{ECI}}(t_1)/d\vec{S}_{\mathrm{ECI}}(t_0)$.

Injecting equation (6) into equation (7) and using the notion of Jacobian matrix, it comes that the state transition matrix can be also integrated numerically as an initial-value problem decribed by the following system of differential equations:

$$\frac{d}{dt}\mathbb{S}_{\mathrm{ECI}} = \mathbb{J}_{\mathrm{ECI}}(t)\mathbb{S}_{\mathrm{ECI}}(t) \tag{8}$$

and $\mathbb{S}_{\mathrm{ECI}}(t_0) = \mathbb{I}$, the identity matrix. To use the same numerical strategy as for the state alone, the matrix system of differential equations is flattened and a new vector of unknowns of size 42 is considered - on average, the overhead of propagating numerically the state transition matrix along with the state is around 500%.

Propagating the orbits of all the spacecrafts potentially watched for collision events allows to determine the TCA between each pair of spacecrafts. An illustration of the propagation of the orbits of the ENVISAT satellite and COSMOS 1269 debris is given in figure 2 around 19:14:10 UTC on 2004-09-02, which is the time of closest approach of the two objects in that period (this result obtained with **Re·Propagate** matches the data published by Klinkrad *et al.*[23]).

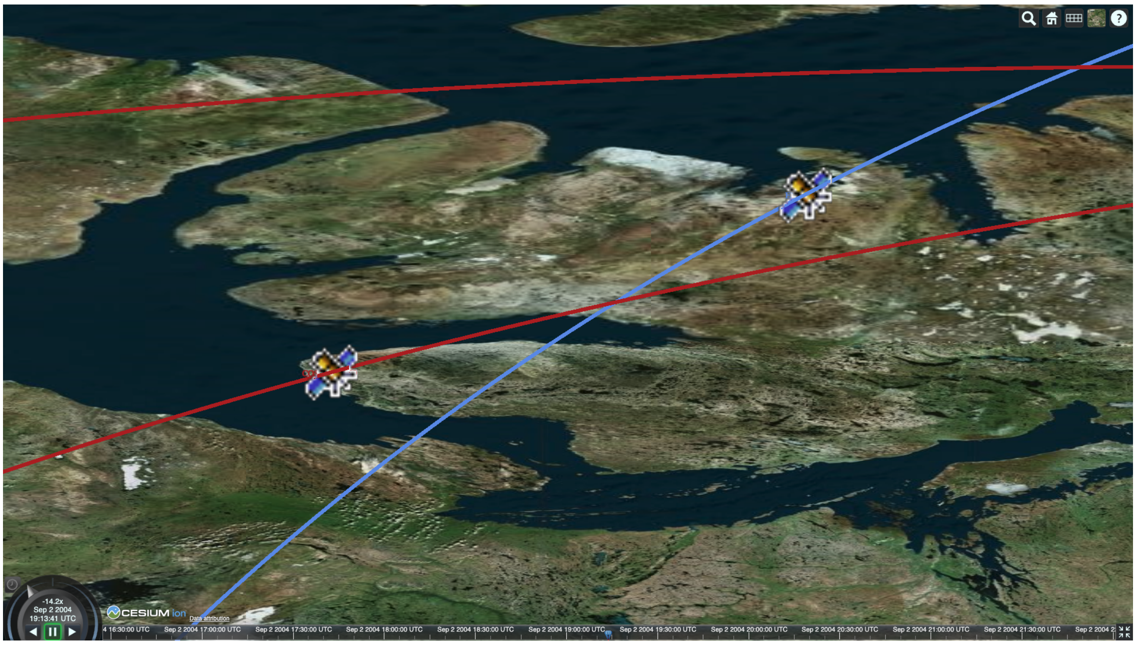
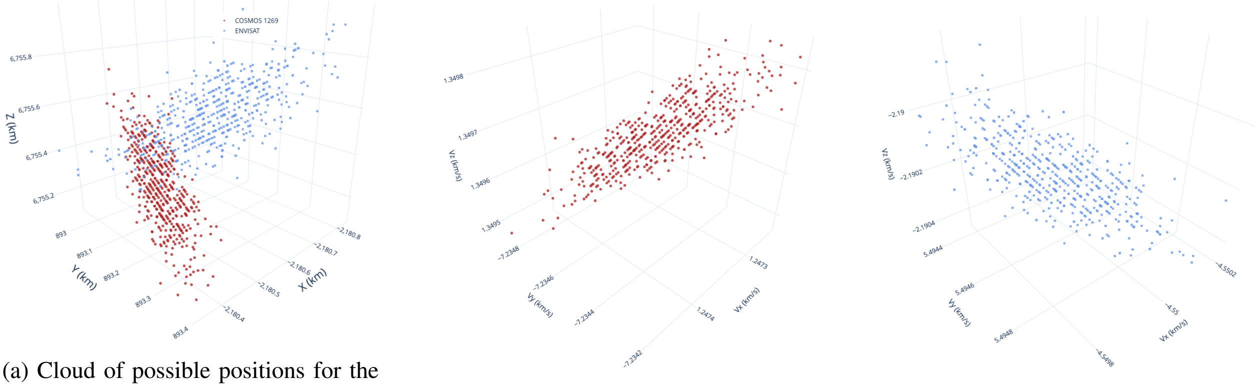


Figure 2: Orbit traces of the ENVISAT satellite (blue line) and the COSMOS 1269 debris (brickred line) on 2004-09-02 around the time of closest approach between the two objects in that time period. The TCA was estimated with **Re·Propagate** to occur at 19:14:10 UTC on that day, which matches the data published by Klinkrad *et al.*[23]

Since the covariance matrix has also been propagated, **Re·Propagate** gives at the TCA of each pair of spacecrafts the position and velocity of the spacecrafts and the uncertainty thereof. This information is then used to perform a Monte Carlo simulation of both the primary and secondary object states at the TCA and to statistically determine the probablity of collision based on the number of trials that violate a predetermined proximity threshold. Some trials of

DOI: 10.13009/EUCASS2023-350

the Monte Carlo analysis performed for the conjunction of the ENVISAT satellite and the COSMOS 1269 debris on 2004-09-02 are given in figure 3 - this small analysis yield a collision probability of the order of $9 \times 10^{-5}$ which is consistent with the literature about that specific conjunction.



(a) Cloud of possible positions for the ENVISAT satellite and COSMOS 1269 debris at TCA.

(b) Cloud of possible velocityies for the COSMOS 1269 debris at TCA.

(c) Cloud of possible velocities for the ENVISAT satellite at TCA.

Figure 3: Samples of the Monte Carlo simulation of the position and velocity of the COSMOS 1269 debris and the ENVISAT satellite at the time of closest approach between the two objects on 2004-09-02.

## 3. Endo-atmospheric prediction

To support the end-of-life phase of satellites, Re CAE has developed two components, **Re·Entry** and **Re·CFD**.

The former is a Python shell around a high-performance low-level kernel that handles the prediction of the re-entry trajectory of satellites, including ablation & fragmentation and that is capable of yielding the number of fragments as well as their landing footprint - see section 3.1.

On the other hand, **Re·CFD** is a high-fidelity multi-physics Computational Fluid Dynamics (CFD) simulation software written purely in a low-level language. It leverages both shared and distributed memory parallelization technologies in a hybrid manner to ensure an optimal performance and a reduced time-to-solution - see section 3.2.

### 3.1 Re-entry trajectory

Compared to an orbital propagation, the actual atmospheric flight time of the vehicle is often only in the hundreds to thousands of seconds, making much more detailed models and closures for the forces acting on the vehicle numerically tractable. Due to the strong dynamics involved in the re-entry phase, those models are also actually necessary to obtain a reliable prediction of the trajectory. In addition to solving for the 3 d.o.f. motion of the vehicle, it also becomes computationally affordable and, once again, a pre-requisite for accuracy, to consider the evolutions of the attitude and of the rotation of the vehicle, *i.e.* to solve for the 6 d.o.f. motion of the vehicle (refer *e.g.* to Landau & Lifshitz[25] or Greenwood[16] for a detailed discussion on the subject of solid mechanics). As will be discussed below, the source terms for the evolutions of attitude and rotation are easily expressed in a frame attached to the vehicle, which is more easily related to an Earth-Centered Earth-Fixed (ECEF) frame of reference *via* the vehicle attitude quaternion. Instead of the inertial reference frame used in **Re·Propagate**, the equations of motion for the re-entry trajectory are therefore given in a non-inertial ECEF frame of reference, specifically the International Terrestrial Reference Frame (ITRF) (see *e.g.* Altamimi *et al.*[1]).

Mathematically, the problem can therefore be expressed as the following set of ordinary differential equations:

DOI: 10.13009/EUCASS2023-350

$$\begin{cases} \dfrac{d\vec{x}_{\text{ECEF}}}{dt} & = \vec{V}_{\text{ECEF}}, \\ m_{\text{vehicle}} \dfrac{d\vec{V}_{\text{ECEF}}}{dt} & = \underbrace{\vec{F}_{\text{grav}} + \vec{F}_{\text{inertial}}}_{\vec{F}_{\text{grav,ECEF}}} + \vec{F}_{\text{aerodynamic}}, \\ \dfrac{d\vec{\Omega}}{dt} & = \mathbb{I}_{\text{vehicle}}^{-1} \left[ \vec{M}_{\text{aerodynamic}} - \vec{\Omega} \times \left( \mathbb{I}_{\text{vehicle}} \cdot \vec{\Omega} \right) \right], \\ \dfrac{d\mathfrak{q}}{dt} & = \dfrac{1}{2} \underline{\underline{W}}(\mathfrak{q})^{\dagger} \left( \vec{\Omega} - \underline{\underline{R}}(\mathfrak{q}) \cdot \vec{\Omega}_{\oplus} \right) \end{cases} \tag{9}$$

where the two first equations are simply expressed in the aforementioned ECEF frame of reference, whereas it is more physically relevant to express the two last equations in the non-inertial frame of reference attached to the vehicle. The nomenclature is straightforward, with perhaps the exception of $\mathfrak{q}$ which is the attitude quaternion[11] of the vehicle relating the ECEF and the vehicle frames of reference.

### 3.1.1 Numerical details

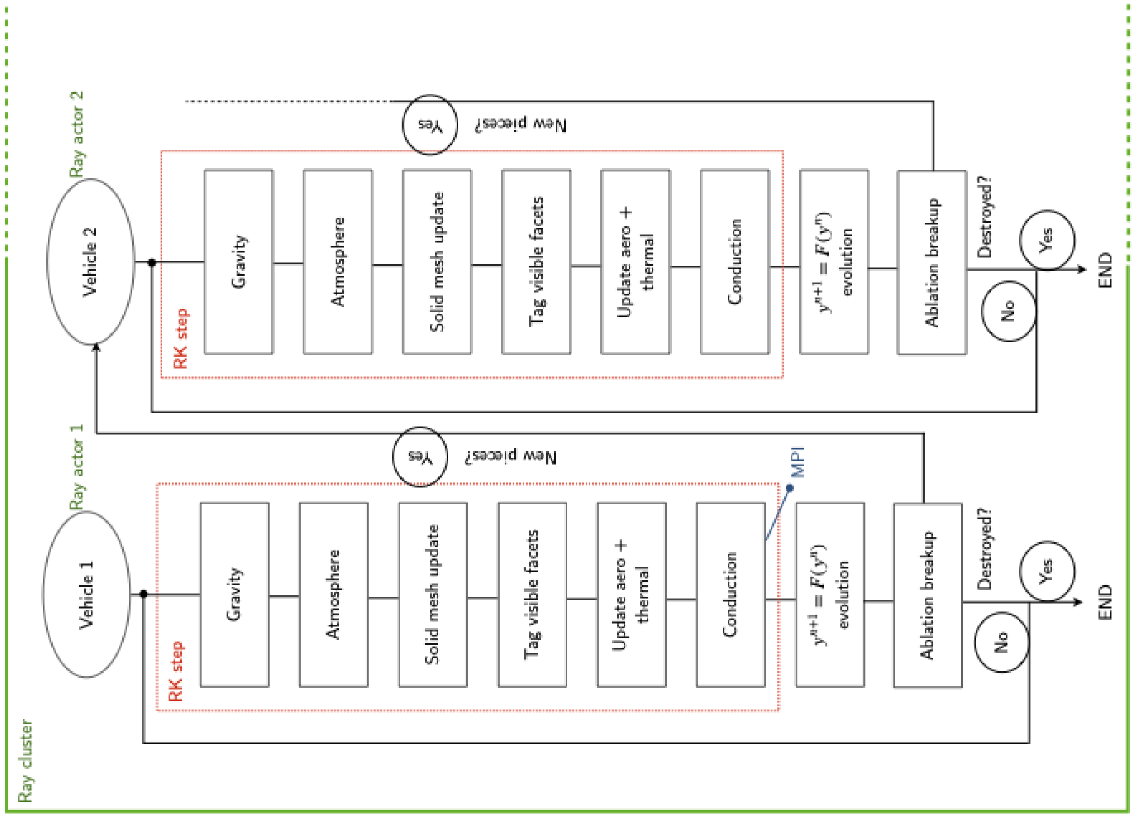


Figure 4: Complete flowchart of **Re·Entry** describing the different steps of the re-entry trajectory prediction as well as the strategy to handle multiple debris/body re-entering.

**Re·Entry** is designed with a modular architecture that allows to easily solve for the re-entry trajectory of multiple vehicles at once. The vehicles can either be actual spacecrafts that it would be interesting to study together, or more likely, the different fragments of a single spacecraft that has broken up during re-entry. To each debris is associated a Runge-Kutta method with adaptive timestepping that solves for the set of equations (9): to offer the best time-to-solution and no overhead when a new debris is created during breakup, each solver is a Ray[29] actor that lives in its own process and can therefore be spawn and killed independently of the others. The spawing and destruction process of each actor is exhibited on the entire flowchart of **Re·Entry** shown in figure 4.

The lifecycle of each actor is centered around the ensemble of steps taken to integrate in time the equations of motion. Each step is a succession of updates of each object state (gravity field, atmospheric model, etc.) that leads to the

DOI: 10.13009/EUCASS2023-350

evolution of the state of the object to the next timestep - detailed discussions about the models will be given in the following paragraphs 3.1.2-3.1.4. Once the new state of vehicle is known, it becomes possible to proceed to a breakup assessment and to spawn new actors if necessary.

### 3.1.2 Influence of the planet

Before moving on to an in-depth discussion about the aerodynamic modeling and the strategy for ablation and break-up, let us first detail the closures of the terms of equation (9) that are related to the planet.

The gravitational pull of the Earth on the vehicle is modeled in **Re·Entry** by the latest XGM2019e (see Zingerle *et al.*[41]) model, similarly to what is done in **Re·Propagate** - see section 2.1, equations (2). As shown on figure 5, it is however not necessary, in general, to use the full set of harmonics of the model. That figure presents the deltas on the ECEF coordinates of a vehicle during a re-entry flight of 1000 seconds in the Earth's atmosphere, computed with the XGM2019e model truncated at different degrees and orders and compared with the degree 2 truncation - the trajectory itself does not matter, it is representative of the trajectories computed by **Re·Entry**. It shows that beyond degree and order 10, the differences obtained are small, hence the choice of truncating the model at degree and order 10 by default, although the decision is ultimately left to the user.

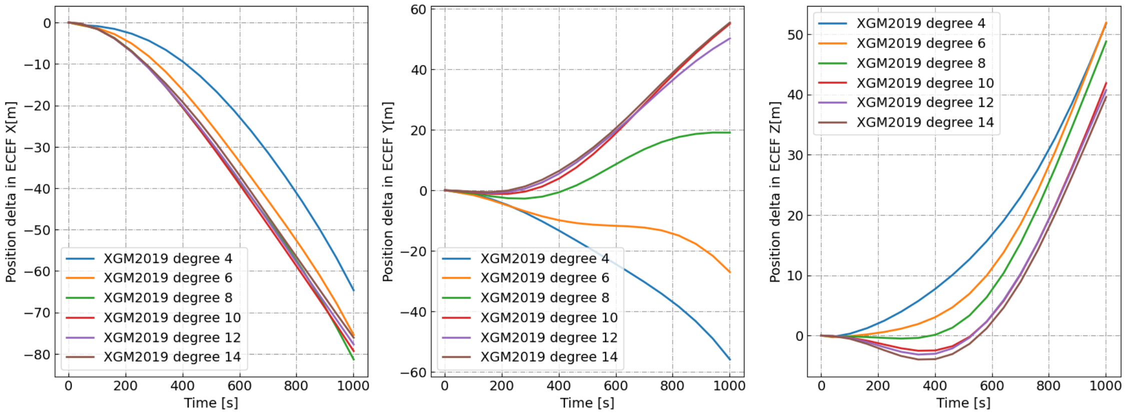


Figure 5: Deltas on the ECEF coordinates of a vehicle during a re-entry flight of 1000 seconds in the Earth's atmosphere, computed with the XGM2019e model truncated at different degrees and orders and compared with the degree 2 truncation.

The inertial force vector $\vec{F}_{\text{inertial}}$ is composed of the very classical centrifugal and Coriolis forces,

$$\vec{F}_{\text{inertial}} = \vec{F}_{\text{centrifugal}} + \vec{F}_{\text{Coriolis}} = m_{\text{vehicle}}\left(\vec{\Omega}_{\oplus} \times \left(\vec{x}_{\text{ECEF}} \times \vec{\Omega}_{\oplus}\right) + 2\vec{V}_{\text{ECEF}} \times \vec{\Omega}_{\oplus}\right). \tag{10}$$

Note that the extra inertial force due to the unsteadiness of the Earth's rotation, $m_{\text{vehicle}}\vec{x}_{\text{ECEF}} \times d\vec{\Omega}/dt$, is not present in equation (10) as the evolution of Earth's rotation rate over such short reentry times is negligible. This comment also holds when closing the term $\vec{\Omega}_{\oplus}$ in equation (9) which is considered as constant over the reentry time and contributing only along the polar axis of the considered ECEF frame, *i.e.* $\vec{\Omega}_{\oplus} = [0, \quad 0, \quad \Omega_{\oplus}]$.

### 3.1.3 Aerodynamic forces and moments

The degree of accuracy of **Re·Entry** in terms of atmospheric modeling is higher than that of **Re·Propagate** because the variability of the atmosphere is much higher at lower altitudes, as illustrated on figure 6 - therefore **Re·Entry** relies on the semi-empirical NRL MSIS v2 model (see Emmert *et al.*[12]) of the neutral atmosphere from ground to space.

The model is leveraged to compute the composition of the atmosphere as a function of the altitude, the latitude, the longitude, the time and the solar activity and coupled to a real-gas thermochemistry mechanism it yields a high-fidelity representation of the density, the pressure, the temperature and the specific heat coefficients of the atmosphere at any point during the trajectory of the re-entering vehicle.

It also yields the Knuden number: for very low Knudsen numbers, the flow is said to be a continuum whereas for very high values, the flow is said to be rarefied - see *e.g.* Bird[5] for a detailed discussion on the subject. Each regime has its own physics and as a consequence **Re·Entry** leverages different semi-analytical models to compute the aerodynamic forces and moments depending on the Knudsen number. Also, to clarify the following discussion, note that each

DOI: 10.13009/EUCASS2023-350

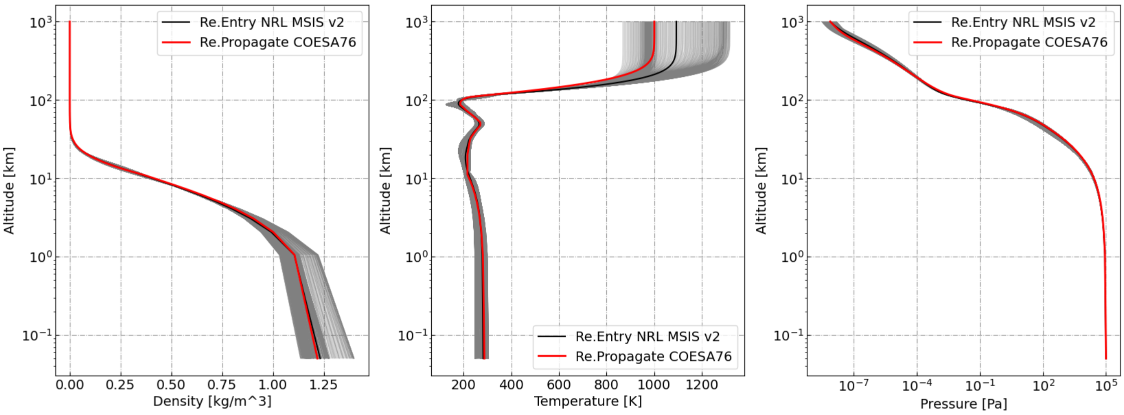


Figure 6: Comparison of the COESA76[3] model used in **Re·Propagate** and of the NRL MSIS v2[12] model used in **Re·Entry**. A random selection of NRL MSIS v2 atmospheric profiles has been generated to exhibit the variability of the model (thin grey lines). The average of the random selection is shown in thick black line, to be compared to the COESA76 model (thick red line).

vehicle is numerically represented with a three dimensional tetrahedralized volume mesh that therefore presents a set of triangular facets to the incoming flow - see *e.g.* figure 7.

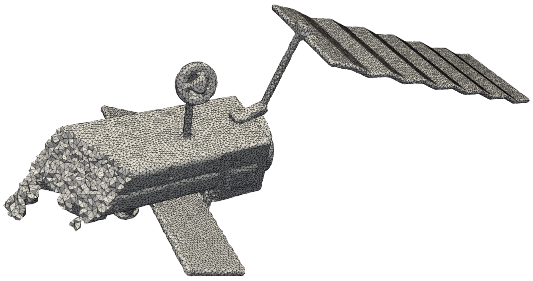


Figure 7: Illustration of the three dimensional tetrahedralized volume mesh used to represent the ENVISAT satellite in **Re·Entry**.

To compute the global aerodynamic forces and moments acting on the vehicle, independently of the flow regime, the pressure and friction coefficients are computed for each facet and then integrated over the entire surface of the vehicle. As such, the formulae that will be presented in the following paragraphs are valid for a single facet of the vehicle, the integration over the entire surface being performed numerically afterwards and before the update of the vehicle state (see section 3.1.1 and figure 4 in particular).

In the rarefied regime, the work by Schaaf & Chambré[8] allows to obtain near-analytical solutions for the pressure and the friction coefficients, based on the free molecular theory and the assumption that both incident and reflected particles have Maxwellian distributions of velocity (see *e.g.* Vincenti & Kruger[37] for more details on the subject, the expressions will not be reproduced here.)

On the other hand, when the medium surrounding the re-entering vehicle becomes a continuum, the Newtonian impact theory is adopted (see *e.g.* Lees *et al.*[26]). By nature, it only applies to the normal component of the flow impinging the vehicle and therefore only the pressure coefficient is non-zero and given by $C_p = C_{p,\max} \sin(\delta)^2$ where $C_{p,\max}$ is the maximum pressure coefficient that can be obtained overall for an assumed-blunt vehicle, *i.e.* right behind the normal part of the detached shock (see Anderson[2] for more details, the expression is not reproduced here.)

Finally, the global aerodynamic force and moment acting on the vehicle are obtained by integrating the pressure and friction coefficients over the entire surface of the vehicle:

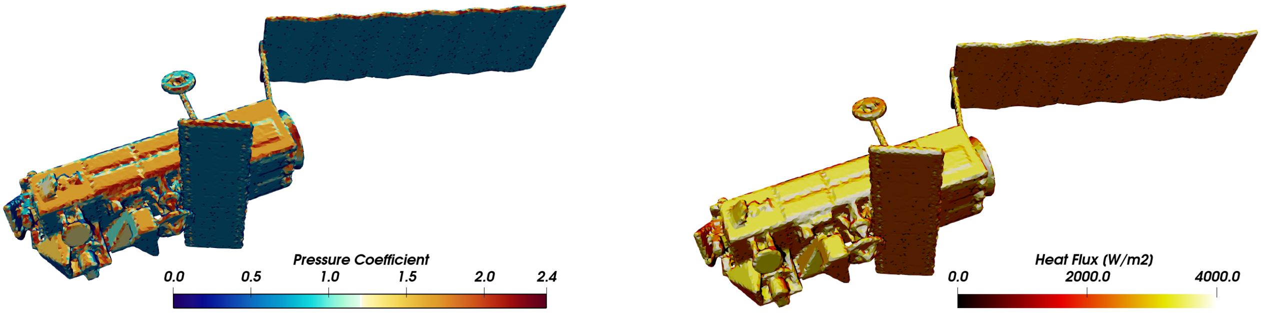
$$\underline{\underline{R}}(\mathrm{q})^{\dagger} \vec{F}_{\text{aerodynamic}} = \frac{1}{2}\rho\|\vec{V}_{\text{vehicle}}\|^2 A_{\text{atmos}} \begin{bmatrix} C_A \\ C_Y \\ C_N \end{bmatrix}, \quad \text{and} \quad \vec{M}_{\text{aerodynamic}} = \frac{1}{2}\rho\|\vec{V}_{\text{vehicle}}\|^2 A_{\text{atmos}} L_{\text{atmos}} \begin{bmatrix} M_{x,G} \\ M_{y,G} \\ M_{z,G} \end{bmatrix}, \tag{11}$$

where $C_A$, $C_Y$ and $C_N$ represent the force coefficients in the vehicle frame of reference and $M_{x,G}$, $M_{y,G}$ and $M_{z,G}$ represent the moment coefficients in the vehicle frame of reference with respect to the center of gravity of the vehicle, expressed from the aforementioned pressure and friction coefficients:

$$\begin{bmatrix} C_A \\ C_Y \\ C_N \end{bmatrix} = \frac{1}{A_{\text{atmos}}} \sum_{k \in \text{facets}} A_k (C_f \vec{t}_k - C_p \vec{n}_k), \quad \text{and} \quad \begin{bmatrix} M_{x,G} \\ M_{y,G} \\ M_{z,G} \end{bmatrix} = \frac{(\vec{x}_k - \vec{x}_G)}{A_{\text{atmos}} L_{\text{atmos}}} \wedge \sum_{k \in \text{facets}} A_k (C_f \vec{t}_k - C_p \vec{n}_k), \tag{12}$$

DOI: 10.13009/EUCASS2023-350

where $A_k$ is the area of the $k$-th facet, $\vec{x}_k$ is the position of the center of the $k$-th facet, $\vec{x}_G$ is the position of the center of gravity of the vehicle, $\vec{n}_k$ is the outwards pointing normal vector of the $k$-th facet and $\vec{t}_k$ is the tangential direction of the flow to the $k$-th facet.



(a) Example of pressure coefficient distribution on the ENVISAT satellite during a re-entry trajectory.

(b) Example of heat flux distribution on the ENVISAT satellite during a re-entry trajectory around 100 km of altitude.

Figure 8: Illustrations of pressure coefficient and heat flux calculations on the ENVISAT satellite.

### 3.1.4 Evolution and integrity of the vehicle

To complete the set of closures for the terms of equation (9), it is necessary to discuss the evolution of the vehicle 6 d.o.f. characteristics, *i.e.* its inertia matrix, mass, center of gravity, centroid, etc. As mentioned in section 3.1.3, the vehicle is numerically represented with a three dimensional tetrahedralized volume mesh upon which is defined a piecewise-constant field representing the material of each cell. That information is paramount to determine the tensor of inertia of the vehicle $\mathbb{I}_{\text{vehicle}}$ during the computation - note that given the computational complexity of evaluating the tensor, it is however not updated at each timestep but rather only when an event that changes the mass distribution of the vehicle occurs.
The characterization of the ablation of the external layers of the vehicle first passes by the computation of the heat flux received by the vehicle. For the same reasons as for the computation of the aerodynamic forces and moments, the heat flux is computed for each facet of the vehicle and then integrated over the entire surface of the vehicle, with different formulae depending on the flow regime (see section 3.1.3 for more details).
In the rarefied regime, using the same assumptions of Maxwellian distributions of velocity for both incident and reflected particles, Schaaf & Chambré[8] showed that the heat flux can also be obtained quasi-analytically (not reproduced here, please refer to the original paper for more details.).
On the other hand, in the continuum regime, the heat flux is computed based on the stagnation point theory developed by Fay & Riddell[14] and extended to any point on the vehicle using a strategy similar to the Newtonian impact theory:[24]

$$q = q_{\text{stag}} \left(0.1 + 0.9 \sin(\delta)\right), \tag{13}$$

where $q_{\text{stag}}$ is evaluated using Fay & Riddell theory and assuming a non-catalytic wall:

$$q_{\text{stag}} = \varphi Pr_{\text{wall}}^{-0.6} (\rho_{\text{ext}}\mu_{\text{ext}})^{0.4} (\rho_{\text{wall}}\mu_{\text{wall}})^{0.1} \sqrt{\nabla V} (h_{\text{ext}} - h_{\text{wall}}) \left(1.0 - \frac{h_{\text{form}}}{h_{\text{ext}}}\right), \tag{14}$$

where $\varphi$ is a shape correction factor, $Pr_{\text{wall}}$ is the Prandtl number at the wall, $\rho_{\text{ext}}$ is the density of the external flow, $\mu_{\text{ext}}$ is the viscosity of the external flow, $\rho_{\text{wall}}$ is the density at the wall, $\mu_{\text{wall}}$ is the viscosity at the wall, $\nabla V$ is the velocity gradient at the wall, $h_{\text{ext}}$ is the enthalpy of the external flow, $h_{\text{wall}}$ is the enthalpy at the wall and $h_{\text{form}}$ is the enthalpy of formation at the wall. An illustration of the results obtained with the two aforementioned theories is given on figure 8(b).

From the incoming heat flux, and assuming that the external wall of the vehicle radiates heat *per* the black body theory, *i.e.* $q_{\text{rad}} = \sigma T_{\text{wall}}^4$, it becomes possible to compute the increment of temperature of the external wall of the vehicle during the current timestep:

$$\forall k \in \text{facets}, \quad \Delta T_k = (q_k - q_{\text{rad},k}) \frac{\Delta t}{\rho_k c_{p,k} V_c^k}, \tag{15}$$

DOI: 10.13009/EUCASS2023-350

where $q$ and $q_{\text{rad}}$ have been re-indexed with the facet index for clarity purposes and $V_c^k$ is the volume of the cell to which facet $k$ belongs. A comparison with the melting temperature of the material the cell is made of then allows to determine whether the facet is ablated - in **Re·Entry**, the first level of modeling inmplies that if ever the melting temperature of the cell is reached, the whole cell is considered ablated and it is removed from the mesh.
As the material is slowly deteriorated by the incoming heat flux, and some parts of the debris are completely ablated (*e.g* solar panels support rods, antennae, etc.), it becomes necessary to account for the independence of the multiple pieces. As shown in figure 4, the strategy adopted in **Re·Entry** is to spawn a new actor for each new debris that is created during the re-entry and the computation goes on, tracking the trajectory of each debris independently.

### 3.2 High-fidelity hypersonic computations

As discussed in lengths in the previous section 3.1, the **Re·Entry** component offered by Re CAE platform is a fast tool that is meant to be used in a pre-design loop or an optimization framework. Once the preliminary design phase is over and the main features of the re-entry vehicle have been defined, it is always necessary to perform a more detailed analysis of the vehicle's response to the atmospheric environment, to ensure, in the context of this paper, that the vehicle does not survive re-entry and leaves zero debris behind.
This is where the **Re·CFD** component comes into play. It is a high-fidelity multi-physics simulation code that is capable of yielding accurate results about the flow around a re-entering object as well as the three-dimensional thermal conduction inside the vehicle. Altogether, **Re·CFD** can be seen either as an on-the-spot validation tool for the results obtained with **Re·Entry** or as a standalone tool for the detailed design of the re-entry vehicle, in terms of external aerodynamics and thermal protection system.

The design of the component is thought in such a way that the co-operability with the other components and in particular with **Re·Entry** is as seamless as possible. To this end, the geometries and three-dimensional tetrahedral internal meshes of the vehicles & debris used and generated on-the-fly in **Re·Entry** can be directly used in **Re·CFD** without any modification.
Debris geometries can however be quite random and complex by nature, making the process of creating an adjusted unstructured mesh around them quite tedious and time consuming. That is the main reason why **Re·CFD** leverages simple irregular Cartesian meshes generated automatically and without any input from the user and uses the immersed boundary condition (IBC) paradigm (see *e.g.* Mittal *et al.*[28] for a detailed review of the technique) to account for the presence of the debris in the flow.

#### 3.2.1 A quick word about the external aerodynamics

Before discussing in details the implementation of the sharp immersed boundary method in **Re·CFD** (see paragraph 3.2.2) to account for the presence of the debris in the flow, let us first describe the mathematical formulation used for the external aerodynamics.
**Re·CFD** solves the three-dimensional compressible Navier-Stokes equations in a dimensionless formulation used to minimize the stiffness of the system in high-Mach & high-Reynolds numbers regimes. Mathematically, the system can be written as follows:[27]

$$\overbrace{\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{bmatrix}_t}^{\mathbf{U}} + \overbrace{\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ \rho uE+pu\end{bmatrix}_x}^{\mathbf{F}} + \overbrace{\begin{bmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vw\\ \rho vE+pv\end{bmatrix}_y}^{\mathbf{G}} + \overbrace{\begin{bmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho w^2+p\\ \rho wE+pw\end{bmatrix}_z}^{\mathbf{H}} = \overbrace{\begin{bmatrix}0\\ \tau_{xx,x}+\tau_{xy,y}+\tau_{xz,z}\\ \tau_{xy,x}+\tau_{yy,y}+\tau_{yz,z}\\ \tau_{zx,x}+\tau_{zy,y}+\tau_{zz,z}\\ \nabla\cdot(\tau\cdot\mathbf{u}-\mathbf{q})\end{bmatrix}}^{\mathbf{E}_x^{v,x}+\mathbf{E}_y^{v,y}+\mathbf{E}_z^{v,z}}, \tag{16}$$

where the subscripts indicate differentiation, $\mathbf{U}$ is the vector of conservative dimensionless variables and $\mathbf{F}$ ($\mathbf{E}^{v,x}$), $\mathbf{G}$ ($\mathbf{E}^{v,y}$) and $\mathbf{H}$ ($\mathbf{E}^{v,z}$) represent the inviscid (viscous) fluxes in $x-$, $y-$ and $z-$direction respectively.
In the above dimensionless expressions, $t$ denotes the time and $x$, $y$ and $z$ are the Cartesian coordinates, $\rho$ denotes density, $u$, $v$ and $w$ denote the $x-$, $y-$ and $z-$direction velocity components respectively, $E$ denotes the specific total energy, $p$ denotes the static pressure, $\tau = \hat{\mu}\left[-\frac{2}{3}(\nabla\cdot\mathbf{v})\mathbb{I} + \nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathrm{t}}\right]$ is the viscous stress tensor and $\mathbf{q} = -\frac{\gamma\hat{\mu}}{(\gamma-1)Pr}\nabla\left(\frac{p}{\rho}\right)$ is the heat flux vector. Here the equation of state cannot be that of a perfect gas because of the high compressibility factor, enthalpies and pressures reached in the shock layer, hence the polynomial approximations for a real-gas dry air proposed by Hansen[17] are used instead. As for the non-dimensionalization, we rely on the speed of sound as reference scale for the velocity as is often done for high-Mach number compressible flow solvers:

DOI: 10.13009/EUCASS2023-350

$$\rho = \frac{\rho^*}{\rho^*_\infty},\quad p = \frac{p^*}{\rho^*_\infty a^{*,2}_\infty},\quad [u,v,w] = \left[\frac{u^*}{a^*_\infty}, \frac{v^*}{a^*_\infty}, \frac{w^*}{a^*_\infty}\right],\quad [x,y,z] = \left[\frac{x^*}{L}, \frac{y^*}{L}, \frac{z^*}{L}\right],\quad t = \frac{t^*}{L/a^*_\infty}, \tag{17}$$

and

$$\hat{\mu} = \frac{\text{Ma}\,\mu}{\text{Re}}, \tag{18}$$

is the dimensionless dynamic viscosity with Ma the Mach number of the freestream, Re its Reynolds number and $\mu$ evaluated using the Sutherland's law.

In terms of numerical methods, **Re·CFD** relies on a mixed finite-volume flux-balance formulation for the hyperbolic terms ($\mathbf{F}_x$, $\mathbf{G}_y$ and $\mathbf{H}_z$) and a finite-difference formulation of the gradients involved in the parabolic terms ($\mathbf{E}^v_{x,y,z}$). The flux-balance is obtained with numerical fluxes computed at each face of each cell using an approximate Riemann solver[34] relying on left- and right-interpolated values from the neighboring cells - the interested reader is referred to Nauleau *et al.*[30] and Bridel-Bertomeu[7] for the mathematical details of the formulation. The discrete time integration can be performed using either a third-order strong stability preserving Runge-Kutta scheme[15] for time-resolved simulations or a first-order backward Euler scheme for steady-state simulations, relying on a version of the LU-SGS implicit solver.[6]

### 3.2.2 The immersed boundary condition paradigm

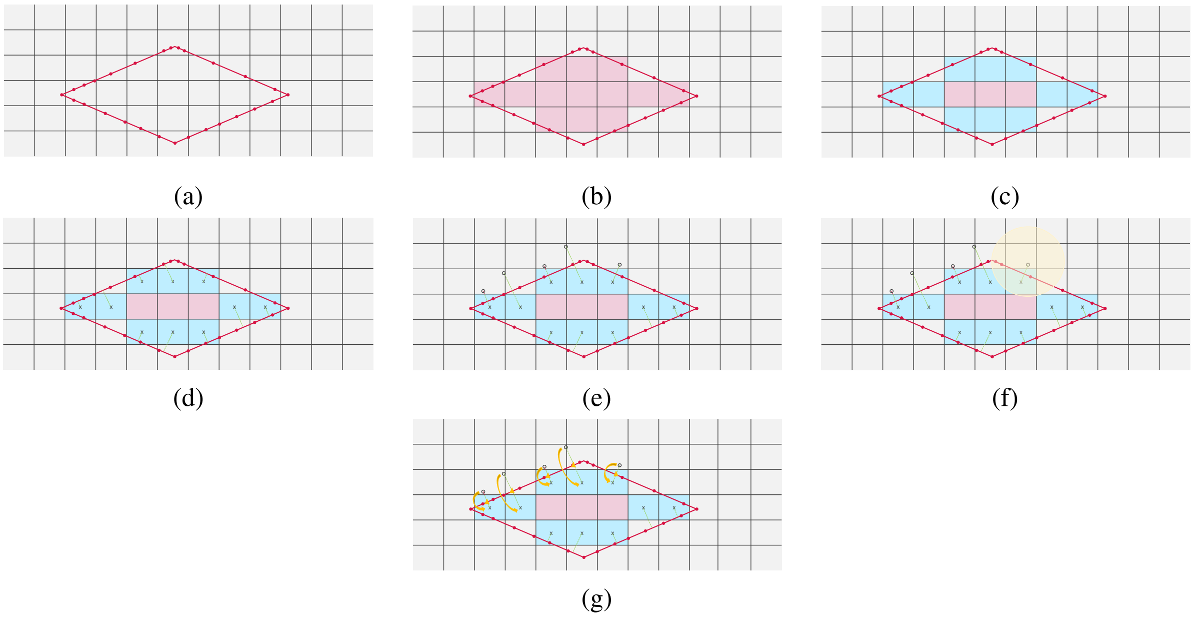


Figure 9: Immersed boundary method workflow - diagrams are shown in two dimensions but the extension to three dimensions in straightforward. (a) Introduction of a tesselated object in the Cartesian mesh. (b) Detection of the immersed cells (red), *i.e.* solid cells, at initialization. (c) Detection of the immersed boundary cells (blue), *i.e.* solid cells with at least one fluid neighbor within the extent of the stencil, at initialization. (d) Detection of the nearest facet to each immersed boundary cell, at initialization (e) Creation of the image points in the direction normal to the nearest facet for each immersed boundary cell, at initialization. (f) Illustration of the neighborhood (yellow) where fluid cells are queried for information to interpolate the values of the fields at the image points, at each iteration of the fluid solver. (g) Illustration of the linear extrapolation from the image points to the immersed boundary cells to fill the values allowing for the enforcement of the boundary condition at the wall of the immersed object, at each iteration of the fluid solver.

To handle the presence of debris in the compressible flows, **Re·CFD** uses a sharp immersed boundary method (IBM).[28] As mentioned in many papers,[22,31] the sharp interface method proves to be well suited for compressible flows because the boundary conditions at the immersed boundary are taken into account directly rather than being computed indirectly *via* a forcing term or smoothed with a distribution function. In the particular version implemented in **Re·CFD**, the IBM formulation is based on ghost cells so there is no need to modify the original flow solver or degenerate the reconstruction stencils at the edges.

Overall, the workflow (presented on figure 9) is fairly classical[9,22,28,31,39,40] but the implementation relies on novel algorithms that guarantee robustness and short time-to-solution:[30]

- a fast rasterization algorithm based on video games rendering techniques co-developed with D. Eberly,[33]
- a massively parallel migratable tasks-based algorithm for the interpolation and reconstruction phase - figure 9(f),
- a non-oscillatory reconstruction method robust to the presence of strong discontinuities in the flow.[7]

These three items make the core of **Re·CFD** and are the main reason why the code is capable of yielding accurate results in a short time - an illustration of the flow around the ENVISAT satellite is given in figure 10 for arbitrary reentry conditions.

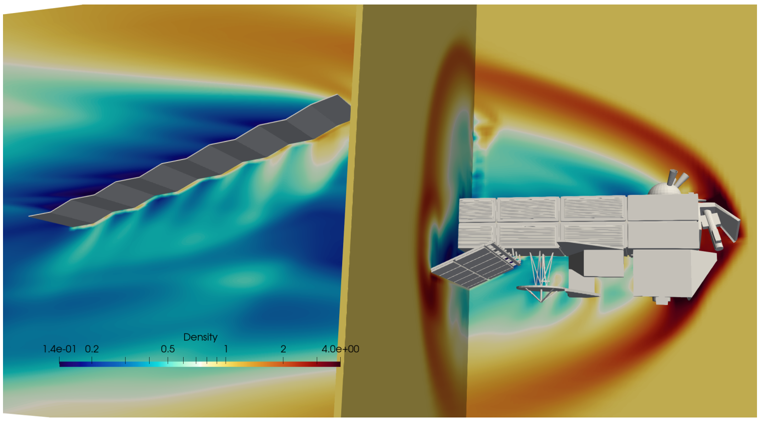


Figure 10: Illustration of the flow around the ENVISAT satellite at Mach 5.

### 3.2.3 Internal thermal behaviour

Along with the external aerodynamic of the debris, **Re·CFD** is also capable of computing the heat diffusion inside them. As mentioned in paragraphs 3.1.4 and 3.2, each debris is represented with a three dimensional tetrahedralized volume mesh upon which is defined, among others, a piecewise-constant field representing the material of each cell. Since the aerodynamic solver in **Re·CFD** is decoupled from the unstructured internal mesh of the debris, the latter can be used to support the resolution of the heat diffusion equation, knowing the heat flux brought upon the external wall by the hypersonic flow and the characteristics of the materials of the derbis. Also, since the physics are decoupled, it becomes possible to not rely also on finite-volumes to solve for the heat equation but rather use the finite-element method, well known to be more suited to elliptic problems than the former.
The heat diffusion equation solved in **Re·CFD** is the following:

$$\rho C_p \frac{\partial T}{\partial t} = \nabla \cdot (\kappa \nabla T) + \dot{q}, \tag{19}$$

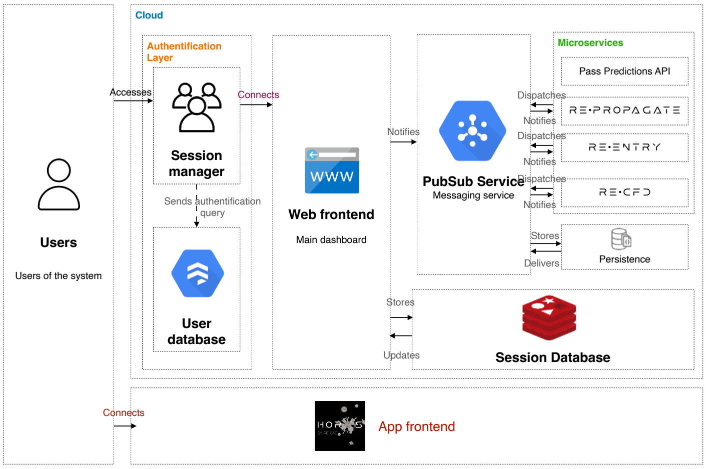
where $\dot{q}$ is the potential heat generation per unit volume, $\kappa$ is the thermal conductivity, $C_p$ is the specific heat capacity at constant pressure and $\rho$ is the density of the material and assuming a constant density of the materials (*i.e* no pyrolisis) and constant specific heat capacity in the range of temperatures encountered during re-entry. Using the finite-element method, the equation is discretized and put in the following matrix form:

$$\mathbb{M}\frac{\partial T}{\partial t} + \mathbb{K}T = \mathbb{F}, \;\text{ with }\; \mathbb{M}_{ij} = \int_V N_i N_j \rho C_p dV,\; \mathbb{K}_{ij} = \int_V \nabla N_i \cdot \kappa \nabla N_j dV, \;\text{ and }\; \mathbb{F}_i = \int_V N_i \dot{q} dV, \tag{20}$$

where $\mathbb{M}$ is the mass matrix, $\mathbb{K}$ is the stiffness matrix, $\mathbb{F}$ is the vector of nodal heat sources and $N_i$ and $N_j$ are the Lagrangian shape functions. Using the heat flux coming from the fluid solver as boundary condition at the external wall of the debris, the matrix system (20) is solved using a first-order backward Euler scheme for time integration and the biconjugate gradient stabilized iterative method (BiCGSTAB) for the linear system.[36] In the context of **Re·CFD**, solving the unsteady heat diffusion equation during a flow timestep allows to update the temperature of the wall of the debris and therefore to assess whether some parts get ablated and to refine the computation of the incoming heat flux at the next timestep.

DOI: 10.13009/EUCASS2023-350

## 4. Cloud architecture



The cloud architecture of the system is designed to be robust, scalable, and secure, leveraging a variety of technologies to ensure optimal performance and user experience.

Users interact with the system through two primary interfaces: a web interface and a dedicated iOS/Android application. The web interface, a multi-page dashboard developed using Dash,[19] provides a comprehensive access to the system's microservices, from orbital propagation to high-fidelity CFD simulations. The mobile application, developed natively for iOS and Android platforms, offers a more pedagogical experience mostly focusing on raising awareness about the space debris problem.

The system's backend is hosted on a cloud architecture, specifically a Kubernetes cluster, which houses all the microservices offered to the users. The first layer of security is the Authentication Layer, which is implemented using Flask for session management. User authentication queries are sent to a Google Cloud Platform (GCP) Firestore database, which stores user information.

The Web Frontend, the main dashboard for most applications, is connected with a Session Manager built on top of an in-memory Redis database which holds session information for connected users. The entries in the database are designed to have a relatively short lifetime to ensure security. The files uploaded through the Web Frontend are stored on user-specific Blob Storages to ensure data privacy. These storage units are mounted both on the Web Frontend workers and on the microservices pods, allowing files to be manipulated directly without further data exchange, enhancing the system's efficiency.

Communication between the frontends and backend microservices is handled by a Redis PubSub service, which acts as a messaging service and allows to implement a loosely coupled architecture with nonetheless a high level of responsiveness. The frontends post messages on the different channels of the service, which in turn dispatches notifications to the concerned backend microservice once a request is ready. To ensure the persistence of messages received by the PubSub service while the request is not complete, a NoSQL MongoDB database is used that stores chunks of messages and concatenates them into the request once it is ready.

The backend microservices layer hosts a variety of services. The Pass Predictions API, a FastAPI, provides information about future passes of a given space object over a given location - this service is used by the smartphone applications but can also be accessed over HTTP, as is. **Re·Propagate** is a Python-wrapped kernel that handles orbital propagation and collision risk assessment. **Re·Entry** is another Python-wrapped kernel that handles low-fidelity reentry simulations. **Re·CFD** is a high-fidelity Computational Fluid Dynamics (CFD) code that performs RANS computations and multi-physics CFD simulations.

This architecture is designed to be modular and scalable, allowing for the addition of new services and capabilities as required. The use of cloud-based services and microservices architecture ensures high availability and fault tolerance, providing a reliable and efficient system for users.

## 5. Perspectives

This paper introduced the platform developed by Re CAE to support the life cycle of space objects, from the design phase to the end-of-life phase. The platform is designed to be modular and scalable, and is therefore capable of supporting a wide range of applications - articulated in three main components, it is indeed capable of propagating orbits as well as the position and velocity uncertainties related thereto, assessing quickly the re-entry of an object with its break up and the landing footprint of the fragments, and finally performing high-fidelity CFD simulations of the re-entry of an object for design purposes.

The next steps in the development of the platform are plural. The design of the collision assessment capabilities embedded in **Re·Propagate** will be overhauled with a novel game-changing algorithm that will remove the need to perform expensive Monte Carlo simulations. The predictions of **Re·Entry** will be equipped with structural analyses to refine the break up of the object and the landing footprint of the fragments. The uncertainty propagation proposed

DOI: 10.13009/EUCASS2023-350

in **Re·Propagate** will also be extended to the re-entry phase to offer a better qualification of the re-entry flights of demiseable spacecrafts. Finally, new reduced order models shall be developed to offer an even better accuracy during re-entry while keeping the computational cost low.

## References

[1] Zuheir Altamimi, Patrick Sillard, and Claude Boucher. Itrf2000: A new release of the international terrestrial reference frame for earth science applications. Journal of Geophysical Research: Solid Earth, 107(B10):ETG–2, 2002.

[2] John David Anderson. Hypersonic and high temperature gas dynamics. Aiaa, 1989.

[3] US Standard Atmosphere. US standard atmosphere. National Oceanic and Atmospheric Administration, 1976.

[4] Roger R Bate, Donald D Mueller, Jerry E White, and William W Saylor. Fundamentals of astrodynamics. Courier Dover Publications, 2020.

[5] G.A. Bird. Molecular Gas Dynamics and the Direct Simulation of Gas Flows. Number Bd. 1 in Molecular Gas Dynamics and the Direct Simulation of Gas Flows. Clarendon Press, 1994.

[6] Jiri Blazek. Computational fluid dynamics: principles and applications. Butterworth-Heinemann, 2015.

[7] Thibault Bridel-Bertomeu. Immersed boundary conditions for hypersonic flows using eno-like least-square reconstruction. Computers & Fluids, 215:104794, 2021.

[8] Paul A Chambre and Samuel A Schaaf. Flow of rarefied gases, volume 5064. Princeton University Press, 2017.

[9] Cheng Chi, Bok Jik Lee, and Hong G Im. An improved ghost-cell immersed boundary method for compressible flow simulations. International Journal for Numerical Methods in Fluids, 83(2):132–148, 2017.

[10] Howard D Curtis. Orbital mechanics for engineering students. Butterworth-Heinemann, 2013.

[11] James Diebel et al. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Matrix, 58(15-16):1–35, 2006.

[12] J. T. Emmert, D. P. Drob, J. M. Picone, D. E. Siskind, M. Jones Jr., M. G. Mlynczak, P. F. Bernath, X. Chu, E. Doornbos, B. Funke, L. P. Goncharenko, M. E. Hervig, M. J. Schwartz, P. E. Sheese, F. Vargas, B. P. Williams, and T. Yuan. Nrlmsis 2.0: A whole-atmosphere empirical model of temperature and neutral species densities. Earth and Space Science, 8(3), 2021.

[13] ESA. Space debris by the numbers, 2023.

[14] James A Fay and Frederick R Riddell. Theory of stagnation point heat transfer in dissociated air. Journal of the Aerospace Sciences, 25(2):73–85, 1958.

[15] Sigal Gottlieb and Chi-Wang Shu. Total variation diminishing runge-kutta schemes. Mathematics of computation, 67(221):73–85, 1998.

[16] Donald T Greenwood. Principles of dynamics, volume 21. Prentice-Hall Englewood Cliffs, NJ, 1988.

[17] C Frederick Hansen. Approximations for the thermodynamic and transport properties of high-temperature air. Number 4150. National Advisory Committee for Aeronautics, 1958.

[18] Samuel Herrick. Astrodynamics. vol. 2: Orbit correction, perturbation theory, integration. London: Van Nostrand Reinhold, 1972.

[19] Shammamah Hossain, C Calloway, D Lippa, D Niederhut, and D Shupe. Visualization of bioinformatics data with dash bio. In Proceedings of the 18th Python in Science Conference, volume 126, page 133. SciPy, Austin, Texas, pp. 126–133, 2019. Available at dash.plotly.com.

[20] George H. Kaplan. The iau resolutions on astronomical reference systems, time scales, and earth rotation models, 2006.

DOI: 10.13009/EUCASS2023-350

[21] Donald J. Kessler and Burton G. Cour-Palais. Collision frequency of artificial satellites: The creation of a debris belt. Journal of Geophysical Research: Space Physics, 83(A6):2637–2646, 1978.

[22] M Ehsan Khalili, Martin Larsson, and Bernhard Müller. High-order ghost-point immersed boundary method for viscous compressible flows based on summation-by-parts operators. International Journal for Numerical Methods in Fluids, 89(7):256–282, 2019.

[23] H Klinkrad, JR Alarcon, and N Sanchez. Collision avoidance for operational esa satellites. In 4th European Conference on Space Debris, volume 587, page 509, 2005.

[24] Georg Koppenwallner, Bent Fritsche, Tobias Lips, and Heiner Klinkrad. Scarab-a multi-disciplinary code for destruction analysis of space-craft during re-entry. In Fifth European Symposium on Aerothermodynamics for Space Vehicles, volume 563, page 281, 2005.

[25] L.D. Landau and E.M. Lifshitz. Mechanics: Volume 1. Number Bd. 1. Elsevier Science, 1982.

[26] Lester Lees. Hypersonic flow. In Proceedings of the Fifth International Aeronautical Conference, Institute of the Aeronautical Sciences, pages 241–76, 1955.

[27] Katate Masatsuka. I do Like CFD, vol. 1, volume 1. Lulu. com, 2013.

[28] Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. Annu. Rev. Fluid Mech., 37:239–261, 2005.

[29] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18), pages 561–577, 2018.

[30] F Nauleau, T Bridel-Bertomeu, F Vivodtzev, and H Beaugendre. Immersed boundaries in hypersonic flows with considerations about high-fidelity and massive parallelism. In 8th European Congress on Computational Methods in Applied Sciences and Engineering. CIMNE, 2022.

[31] Yegao Qu, Ruchao Shi, and Romesh C Batra. An immersed boundary formulation for simulating high-speed compressible viscous flows with moving solids. Journal of Computational Physics, 354:672–691, 2018.

[32] Juan Luis Cano Rodriguez, Yash Gondhalekar, Antonio Hidalgo, Shreyas Bapat, Nikita Astrakhantsev, Chatziargyriou Eleftheria, Kevin Charls, Meu, Dani, Abhishek Chaurasia, Alberto Lorenzo Marquez, Dhruv Sondhi, Tomek Mrugalski, Emily Selwood, Manuel Lopez-Ibanez, Orestis Ousoultzoglou, Pablo Rodriguez Robles, Greg Lindahl, Syed Osama Hussain, andrea carballo, Andrej Rode, Helge Eichhorn, Anish, sme, Himanshu Garg, Hrishikesh Goyal, Ian DesJardin, Matthew Feickert, and Ole Streicher. poliastro/poliastro: poliastro 0.17.0 (SciPy US '22 edition), July 2022.

[33] Philip Schneider and David H Eberly. Geometric tools for computer graphics. Elsevier, 2002.

[34] Eleuterio F Toro. Riemann solvers and numerical methods for fluid dynamics: a practical introduction. Springer Science & Business Media, 2013.

[35] David A Vallado. Fundamentals of astrodynamics and applications, volume 12. Springer Science & Business Media, 2001.

[36] Henk A Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. SIAM Journal on scientific and Statistical Computing, 13(2):631–644, 1992.

[37] Walter Guido Vincenti. Introduction to physical gas dynamic. 1975.

[38] M Weigel, G Kirchner, and H Fiedler. Slr tracking of envisat past its end of life. In 18th International Workshop on Laser Ranging, 13-Po40, November 2013, 2013.

[39] Mehrdad Yousefzadeh and Ilenia Battiato. High order ghost-cell immersed boundary method for generalized boundary conditions. International Journal of Heat and Mass Transfer, 137:585–598, 2019.

[40] Yang Zhang, Xinglong Fang, Jianfeng Zou, Xing Shi, Zhenhai Ma, and Yao Zheng. Numerical simulations of shock/obstacle interactions using an improved ghost-cell immersed boundary method. Computers & Fluids, 182:128–143, 2019.

[41] P Zingerle, R Pail, T Gruber, and X Oikonomidou. The combined global gravity field model xgm2019e. Journal of Geodesy, 94:1–12, 2020.