SCARLET-1.0: SpheriCal Approximation for viRtuaL aggrEgaTes

Eduardo Rossi[1], Costanza Bonadonna[1]

[1]Department of Earth Sciences, University of Geneva, Geneva, 1203, Switzerland

Correspondence to: Eduardo Rossi (Eduardo.Rossi@unige.ch)

5 **Abstract.**

Aggregation of particles occurs in a large variety of settings, and, therefore, it is the focus of many disciplines, e.g. Earth and environmental sciences, astronomy, meteorology, pharmacy, food industry. It is thus very important to provide a full description of the outcome of an aggregation process, starting from its basic features such as the position in space of its components and the overall porosity of the final object. We present SCARLET-1.0, a Matlab package specifically created to

10 provide a 3D virtual reconstruction of aggregates in central oriented collisions that can be later 3D printed. With aggregates in central oriented collisions we refer to aggregates that build up their own structure around the first particle (*the core*) acting as a seed. SCARLET-1.0 belongs to the class of *sphere-composite algorithms*, a family of algorithms that approximate 3D complex shapes in terms of not-overlapping spheres. The conversion of a 3D surface in its equivalent spherical approximation allows an analytical computation of their intersections. Thus, provided a vector of sizes and shapes, SCARLET-1.0 places

15 each element in the vector around the core minimizing the distances between their centers of mass. The user can play with three main parameters that are in charge of controlling the minimization process, namely the solid angle of the cone of investigation (Ω), the number of rays per cone (N_r), and the number of orientations of the object (N_o). All the 3D shapes are described using the STL format, the nowadays standard for 3D printing. This is one of the key features of SCARLET-1.0, which results in an unlimited range of application of the package. The main outcome of the code is the virtual representation

20 of the object, its size, porosity, density and the associated STL file. As an example, here SCARLET-1.0 has been applied to the investigation of ellipsoid-ellipsoid collisions and to a more specific analysis of volcanic ash aggregation. In the first application we show that the final porosity of two colliding ellipsoids is less than 20% if flatness and elongation are greater than or equal to 0.5. Higher values of porosities (up to 40-50%) can be, instead, found for ellipsoids with needle-like or extremely flat shapes. In the second application, we reconstruct the evolution in time of the porosity of two different aggregates

25 characterized by different inner structures. We find that aggregates whose population of particles is characterized by a narrow distribution of sizes tend to rapidly reach a plateau in the porosity. In addition, to reproduce the observed densities, almost no minimization is necessary in SCARLET-1.0; a result that suggests how these objects are quite far from the maximum packing condition often investigated in literature.

1 Introduction

The formation of aggregates from an initial set of individual monomers is a common topic in science, such as in planetary formation, granulation processes, food industry, meteorology, pollution, and Earth sciences (Brauer et al., 2001; Dominik et

35   al., 2006; Poon et al., 2008; Brown et al., 2012; Cuq et al., 2013; Shi, 2015; Dacanal et al., 2016; Pumir and Wilkinson, 2016; Imaeda and Ebisuzaki, 2017; Ohno et al., 2020;). However, the study of aggregation phenomena with experimental setups or direct observations in natural environments is often a challenging - if not impossible - task. For this reason, the use of virtual reality has been considered a valuable alternative to the direct investigation (Lumme and Rahola, 1994; Filippov, 2000; Min et al., 2007).

40   Several algorithms have been dedicated to the numerical solution of theoretical problems concerning aggregation phenomena, such as the study of the maximum packing of geometrical shapes within fixed boundary condition (Conway and Sloane, 1998; Weaire and Aste, 2000; Williams and Jia, 2003; Donev et al., 2004; Hales, 2005; Man et al., 2005); fewer codes have been specifically written for the investigation of aggregation in natural contexts; among them, we only mention those algorithms aimed to study aggregates in protoplanetary disks (Ormel et al, 2007; Yurkin and Hoekstra, 2011), in snowflakes formation

45   (Kessler et al., 1984; Maruyama and Fujiyoshi, 2005; Reiter, 2005; Ning and Reiter, 2007; Gravner and Griffeath., 2009) and in other environmental applications, such as water treatment (Chopard et al., 2006). Unfortunately, not all the available codes are suitable for the great variety of applications encountered in natural sciences. For example, several codes treat aggregation as a fractal process (Nguyen et al., 2003; 2004). However, this is not always the case, as it has been proven for volcanic ash, where recent field observations have revealed that in many cases volcanic ash aggregates are composed of a big particle of

50   about 200-1000 μm (named *the core*) at the center of the structure and a large variety of smaller sizes typically less than 100 μm around it (Bagheri et al., 2016). These two elements poorly fit with a typical monomer-like description at the base of fractal theory, in which the characteristic length of the aggregate, R, is related to the number of monomers involved, N, by means of the power law $N \propto R^{Df}$ (Jacobson, 2005). For those cases where the presence of an inner large particle is present, volcanic ash aggregates are better simulated with a central collisional process, in which a sequence of particles collides towards the

55   inner core, considered as a pole of attraction.

Another important aspect concerning aggregation algorithms is the capability of accurately describe the shapes of the particles involved in the process. In some circumstances the morphology of the colliding objects is sufficiently well described in terms of equivalent spheres. In other cases, a more accurate description is needed, as for example in the study of particle packing or in all those cases where particle interlocking is the leading mechanism for aggregation, such as in snowflakes formation

60   (Gravner and Griffeath., 2009). In these problems the user would benefit from an algorithm that offers the possibility of dealing with arbitrary shapes in the simplest way possible.

All these aspects motivated us to create SCARLET-1.0 (*SpheriCal Approximation for viRtuaL aggrEgaTes*), a Matlab package designed for the study of virtual aggregates generated by central collisional processes of particles with arbitrary 3D shapes. SCARLET-1.0 simulates the binary collision between the core and a vector of N_p particles that are supposed to interact with

65 the growing object one after the other. The algorithm then follows a Monte Carlo approach to investigate the final positions of the *i-th* particle and the already placed aggregate, minimizing the distance between the center of mass of the two bodies. Once that the *i-th* particle has been placed, it is considered fixed within the aggregate.

One of the main problems related to the aggregation of complex surfaces is the detection of the intersections between them. SCARLET-1.0 belongs to the so called *sphere-composite algorithms* (Evans and Ferrar, 1989; Nolan and Kavanagh, 1995),
70 in which each single shape is seen as the superposition of N_{sph} not-overlapping spheres. Describing a given shape in terms of a set of spheres leads to a pure analytical solution to the intersection problem. On the other hand, a large number of spheres is required for those collisions where a high degree of accuracy is needed (Jia and Williams, 2001).

The key feature of SCARLET-1.0 is the use of the *Standard Triangulation Language* (STL) to describe the 3D shapes involved in the collisions. STL is nowadays one of the most common formats for 3D printing, 3D scanning and design (Szilvasi-Nagy
75 and Matyasi, 2003). Therefore, a large amount of software packages – opensource or under license - are now available to create or manipulate this kind of files, such as OpenSCAD (www.openscad.org). This guarantees a great flexibility to create virtual aggregates with SCARLET-1.0 starting from a set of completely arbitrary shapes and the appealing possibility of 3D printing the virtual aggregate. As a demonstration of this, in the main-body of the paper and in the appendix we show virtual aggregates made of a great variety of shapes: from scientific ones, such as volcanic particles, snowflakes, cones, ellipsoids and spheres,
80 to the most creative ones, as a t-rex and a LEGO character.

The manuscript is structured as follows: in section 2 we introduce the technical details of the algorithm, with a particular focus to the fundamental routines *fromStlToSpheres* and *mainSCARLET*; in section 3.1 the algorithm is tested using the problem of maximum packing as a benchmark; in section 3.2 we present an application of SCARLET-1.0 to both the study of generic ellipsoids and to the study of the evolution in time of volcanic ash aggregates.

2 Model description

SCARLET-1.0 is written in Matlab 2015 and it simulates the aggregation process of N_p arbitrary objects, here named "*the coating particles*", around a central one, here referred to as "*the core*". Interestingly, shapes are provided in the STL standard, nowadays one of the most popular formats for 3D printing and prototyping: this means that a limitless range of shapes can be potentially investigated, making SCARLET-1.0 a flexible tool of investigation for science and several other applications, such
90 as visual arts or education.

SCARLET-1.0 is a *sphere-composite algorithm*, which describes any 3D shape in terms of a set of not overlapping spheres. The main advantage of these algorithms consists in reducing the intersection between complex shapes into a simplified problem between spheres, whose solution is purely analytical. This approach is worthwhile in those situations where the particular shapes involved may affect the packing of the final object, such as for volcanic aggregates or snowflakes. On the other hand,
95 two main drawbacks exist: firstly, a large number of spheres might be needed in those cases where particularly complex shapes are involved, affecting the computational efficiency; secondly, a preprocessing code is needed in order to transform a shape in its sphere-equivalent body. SCARLET-1.0 uses a dedicated script named "*fromStlToSpheres*", which is in charge of describing

the STL shape in terms of a user defined number of spheres (see section 2.1). The information is then stored in a specific structure (i.e. "*out_st*" in Fig.1) that represents the typical input for SCARLET-1.0.

100 For what concerns the strategy for particle positioning, SCARLET-1.0 is part of the wide family of *random placement algorithms*, in which particles from a given vector of sizes and shapes are placed in random locations on the surface of the growing aggregate, according some user-defined constraints for the spanned solid angle and the 3D orientation of the object. In SCARLET-1.0 the initial particle, i.e. the core, is placed at coordinate (0,0,0) of a virtual 3D Cartesian space. It represents the seed of the aggregate, around which all the vectors of the coating particles will be aggregated. This process emulates binary

105 collisions, a condition that is usually assumed in the theoretical description of aggregate formation in various natural contexts (e.g. meteorology, Earth sciences). In the next sections we present first a general overview of the code, followed by two section 2.2 and 2.3 dedicated to the details of the preprocessing code (i.e. *fromStlToSpheres*) and the main code (i.e. *mainSCARLET*).

2.1 General overview of the code

In this section we present the main structure behind SCARLET-1.0, while the details of the subroutines are presented in

110 sections 2.2 and 2.3. A dedicated user-guide is present at the Github link: https://github.com/EduardoRossiScience/SCARLET (and also at the Zenodo repository https://doi.org/10.5281/zenodo.4307995), where the package can be downloaded and the code will be constantly updated. As mentioned earlier, SCARLET-1.0 requires a preprocessing code to describe each STL shape in terms of spheres. The code *fromStlToSpheres* is in charge of this and each STL file must be preprocessed singularly. Once the desired shapes have been treated, new Matlab structures will appear as an output in the Workspace, such as *out_st_1*,

115 *fv_1*, *out_st_2*, *fv_2* in Fig.1. The external user can now merge the treated shapes in a new structure (i.e. *input_st* in Fig.1) that will be used as input by SCARLET.

Let us assume for example that we want to create a virtual aggregate made of three different shapes: a T-Rex shape for the core, a cone and an ellipse for the coating (see Fig.2). Therefore, in this example, the preprocessing code must be executed three times, one for each STL file. *fromStlToSpheres* also requires some input parameters, such as the STL filename, the

120 number of spheres and other user-defined variables that will be discussed in sec. 2.2. For this example we type in the Command Window:

$$[\text{core\_spheres, core\_fv}] = \text{fromStlToSpheres}(\textit{'volcanic\_particle.stl'}, 300, 3, 0, 1);$$
$$[\text{coating1\_spheres, coating1\_fv}] = \text{fromStlToSpheres}(\textit{ellipse.stl'}, 300, 3, 0, 1);$$
$$[\text{coating2\_spheres, coating2\_fv}] = \text{fromStlToSpheres}(\textit{'cone.stl'}, 300, 3, 0,1);$$

125

Each time that *fromStlToSpheres* is executed two distinct structures are produced as output in the workspace. In the first structure (i.e. "*core_spheres*", "*coating1_spheres*", "*coating2_spheres*" in the example above), it is stored the information relative to the spheres. In the second one (i.e. "*core_fv*", "*coating1_fv*", "*coating2_fv*") the information relative to the STL file, such as the faces and the vertices of the 3D surface. It is worth noticing that this operation needs to be done only once for

130 each shape: all the structures can be saved for further simulations.

The next step in the procedure is to assemble the shapes in a single structure, named "*input_struct*" in the example, that will be used as input by SCARLET. This operation is done in the Command Window, typing:

input_struct (1).fv = core_fv;

input_struct (1).sphere_struct = core_spheres;

135   input_struct (2).fv = coating1_fv;

input_struct (2).sphere_struct = coating1_spheres;

input_struct (3).fv = coating2_fv;

input_struct (3).sphere_struct = coating2_spheres;

140   It is mandatory to assign to the first element of *input_struct* the shape that we want to assign to the central core. The other shapes will be assigned randomly to the coating particles. The same shape can appear both in the core and in the coating. It is important to notice that the fields "*fv*" and "*sphere_struct*" in the structure must be written rigorously as presented here. On the other hand, no constraint is posed on the name of the structure. The structure *input_struct* is the unique external input required by SCARLET-1.0 that is now ready to be used. However, some internal parameters can be adjusted by the external

145   user in order to setup the desired simulation, such as the number of particles involved, their sizes, the solid angle to be spanned by the algorithm, the number of rotations of each particles, etc. These parameters will be discussed in detail in section 2.3. Once the simulation is over, SCARLET-1.0 produces an output structure (i.e. *SCARLET_output_st* in Fig.1) with some fundamental features of the aggregate, such as the porosity, density and overall size, and a 3D Matlab figure as shown in Fig.2 for the example discussed above. At the end of this section it is worth stressing that the number of shapes and the number of

150   particles are not coincident. In the example discussed here 20 coating particles have only three shapes, even if of different sizes.

2.2 The preprocessing code *fromStlToSpheres*

The preprocessing code *fromStlToSpheres* is in charge of describing a given 3D shape into a random set of not overlapping

155   spheres. To do so, the algorithm takes advantage of the internal structure of the STL file. As a matter of fact, the STL standard describes any closed surface as a set of triangles, which are uniquely described by the coordinates of their *vertices* and *faces*. *Vertices* are expressed as a matrix with three columns and a number of rows equal to the number of vertices N_v. *Faces* are instead described by a matrix of three columns and $\frac{N_v}{3}$ rows, where each row contains three integers reporting the corresponding vertices involved in the creation of the face.

160   *fromStlToSpheres* generates a random point inside the surface using the Matlab built-in function *inpolyhedron*. Then it finds the closest points among all the vertices of the triangles (see Fig.3a, 3d) or the random points already placed. This information

allows the center and the radius of the new placed sphere to be computed, once verified that the random point has not been generated inside an existing sphere.

Referring to Fig.1, *fromStlToSpheres* requires five inputs: i) the STL filename; ii) the number of spheres N_{sph} that we want to use to describe the object; iii) a Boolean variable O_{bary}, set equal to one if we want to place the initial sphere at the barycenter of the shape; iv) a number of random attempts N_{iter} that are associated with the positioning of each sphere; v) a Boolean variable O_{ext} set equal to one if the external user wants to discard the most internal spheres (i.e. those spheres that are completely surrounded by other spheres).

N_{iter} has been introduced to let the user decide about the size of the spheres. In fact, a single sphere that is placed in the STL shape is actually the one with the largest radius among N_{iter} trials. Higher values of N_{iter} thus guarantee that the original surface will be filled with larger spheres respect to small values of N_{iter}.

On the other hand, $O_{ext} = 1$ eliminates those spheres that are completely surrounded by other spheres in the STL file. This operation is based on the built-in Matlab function *boundary*. This can largely speed-up the intersection process, as it will be discussed in sec 2.2, resulting in a general increase of the computational efficiency of SCARLET. However, not all the shapes can be accurately described by their external spheres and $O_{ext} = 1$ should be used only in combination with $N_{iter} = 1$ or for all those problems where a lower degree of accuracy in the overlapping of the shapes is accepted.

As an example, in Fig.3 we show the application of *fromStlToSpheres* to two different volcanic particles: Fig. 3a, 3d illustrate the vertices of the triangles of the STL files. In fig. 3b, 3e, both volcanic particles have been filled with $N_{sph}=300$ spheres. In Fig. 3c, 3f the reduction of the spheres obtained by setting $O_{ext} = 1$.

Fig. 4 shows some applications of the pre-processing routine to different shapes, each of them characterized by a different degree of complexity, such as the presence of convex or not convex hulls. It is worth noticing that the size of the object produced by *fromStlToSpheres* is the same as the one contained in the STL file. Therefore, no modification is done at this stage. It will be the function *mainSCARLET* to be in charge of rescaling the core and the coating particles according to the vector of sizes desired by the external user.

2.3 *mainSCARLET*

mainSCARLET is in charge of creating virtual aggregates from the available set of shapes contained in the input structure of sec. 2.2. However, the user can set up specific internal parameters at the top of code in order to simulate the desired aggregation process, such as the sizes of the particles involved, the degree of packing of the aggregate, etc. (see Fig.1). In the following we illustrate the steps for particle placing, leaving to the user guide for the details of the internal parameters that can be modified by the user.

mainSCARLET is composed of two distinct logical blocks (Fig.5): the first one assigns a 3D shape to the *i-th* object of the vector of N_p coating particles; then it scales the shape and the inner spheres to the characteristic size d_{eq}^i of the *i-th* object. It

is worth stressing that after the scaling process the maximum length of the STL shape is made equal to d_{eq}^i. The second block
195 places the *i-th* object around the inner core, whose Center of Mass (CM) is always made coincident with the origin of the axes.
The first shape of the *input_struct* is assigned by default to the core (see sec. 2.1) and the scaling assumes that the maximum
dimension of the STL file corresponds to the size provided by the user.

The algorithm for particle placing is based on five main loops that are described in detail here:
200   - *Loop 1* (for i=1 to N_p): for each coating particle the code generates a cone of aperture Ω around the unit vector $\vec{v}$,
whose components $\vec{v} = [v_1, v_2, v_3]$ are randomly oriented under the constraint $\sqrt{v_1^2 + v_2^2 + v_3^2} = 1$. The center of the
cone is placed at the origin of the axes (the user can modify Ω by means of *closet.cone_aperture_degree* in the internal
parameters). Large values of Ω produce a wider exploration of the surface of the aggregate before placing the coating
particle; on the contrary, small values of Ω generally produce loose aggregates.
205   - *Loop 2* (for j=1 to N_r): N_r rays are generated within the *i-th* cone (the user controls this parameter by means of
closet.N_raysXSA). The code computes the most external intersection point P_e of each ray with the spheres already
placed in the aggregate.
     - *Loop 3* (parallelized, for k=1 to N_o): the shape associated with the *i-th* particle of size D_i is randomly rotated N₀ times
in the space along three angles $[\varphi, \theta, \psi]$, according to the Euler rotation matrix. This loop is parallelized using *parfor*
210 (the user controls this parameter by means of *closet.N_Euler_triplets*).
     - *Loop 4*: this is a while loop that moves inwards the *i-th* particle starting from the position $P_e + D_i$ along the direction
of the *j-th* ray. It stops when at least one intersection is counted between the spheres of the coating particle and the
aggregate. In the algorithm, the inward motion is thought to be a coarse and fast displacement of the particle,
preprocessing to the fine tuning of loop 5. The step-size of this movement is defined as a fraction f_{c1} of the particle
215 size, whereby default $f_{c1} = 0.05$. However this parameter can be modified by the user by means of the internal
variable *closet.delta*.
     - *Loop 5*: this is a while loop that moves outwards the *i-th* particle from the last location reached in loop 4. It stops
when it is empty the intersection of the spheres belonging to the aggregate and the coating particle. The step-size f_{c2}
of the outward movement is finer than f_{c1} (by default $f_{c2} = 0.00005$). f_{c2} can be varied by means of *closet.delta_2*.
220

For a given ray within the cone, SCARLET determines the rotated particle that has the minimum distance between the CM of
the coating particle and the aggregate (Fig.8). Finally, it selects the coating particle that among all the rays has the minimum
distance respect to the CM of the aggregate. Therefore, the *i-th* coating particle will be placed after two minimization processes:
the first one over all the rotations per each ray; the second one over all the rays.
225

mainSCARLET produces two types of outputs: i) three figures showing respectively the 3D image of the final aggregate, the external volume used for the calculation of the porosity and the points used to determine the external volume itself; ii) the structure *output_st,* containing two substructures *output_st.packing_info* and *output_st.aggregate*. The first one, *output_st.packing_info*, reports quantitative descriptors of the aggregate, such as its mass, density, porosity, size, the external

230 volume and the sum of the inner volumes of its components. The second structure, *output_st.aggregate*, contains all the information relative to the location in space of the core and the coating particles.

In SCARLET-1.0, the determination the aggregate porosity (Eq. 1) is done under the assumption that the external volume V_{ext} of the aggregate is well approximated by the convex hull outlined by the most outer points of its internal spheres. The volume V_{int}^i of each component of the aggregate is calculated directly from the surface of the scaled STL shape, using the divergence

235 theorem (see the K. Suresh algorithm on MathWorks File Exchange). In SCARLET-1.0 all the inner components of the aggregate are characterized by a unique density ρ_p, that can be modified by the user by means of the variable *closet.core_density*. Porosity ϕ_{agg}, aggregate density ρ_{agg} and aggregate packing τ are related as shown in Eq.2 . Finally, the characteristic size D_{agg} assigned to the aggregate is th e *sphere-equivalent diameter*, calculated as the diameter of a sphere with the same external volume of the aggregate (Eq.3).

$$\phi_{agg} = 1 - \frac{\sum_{i=1}^{N_p+1} V_{int}^i}{V_{ext}} = 1 - \tau \qquad \text{(Eq. 1)}$$

240

$$\rho_{agg} = \rho_p \cdot (1 - \phi_{agg}) \qquad \text{(Eq. 2)}$$

$$D_{agg} = \left(\frac{6}{\pi} \cdot V_{ext}\right)^{1/3} \qquad \text{(Eq. 3)}$$

where $\boldsymbol{V_{int}^i}$ is the volume of the *i-th* particle and $\boldsymbol{V_{ext}}$ is the global volume of the aggregate.

3 Results

245 ### 3.1 Testing the model

SCARLET-1.0 has been tested evaluating the packing τ of spheres and ellipsoids with axes ratios (L=1.25):(I=1):(S=0.8), for which theoretical values of maximum packing are available in literature (see Fig.2 of Man et al., 2005). This particular choice of the largest axis (L), the intermediate (I) and the smallest one (S) guarantees an equal value for flatness ($\boldsymbol{f = \frac{S}{I} = 0.8}$) and elongation ($\boldsymbol{e = \frac{I}{L} = 0.8}$). The goal of the test is to relate the computed values of $\boldsymbol{\tau}$ with the initial setup adopted in each

250 simulation for solid angles, number of rays and number of rotations respectively, i.e. ($\boldsymbol{\Omega}$, $\boldsymbol{N_r}$, $\boldsymbol{N_o}$). Virtual aggregates created in this test are made of particles with the same size. Packing is displayed in Fig.9 as a function of the dimensionless parameter $\boldsymbol{\gamma = \bar{a}/R}$, where $\boldsymbol{\bar{a}}$ is the characteristic size of the ellipsoid or the sphere, and R is the radius of a sphere with same external

volume of the aggregate. More in detail, $\bar{a}$ is equal to the radius when spheres are adopted as particles; $\bar{a} = (a_1 \cdot a_2 \cdot a_3)^{1/3}$ for an ellipsoid with principal axes $2a_1$, $2a_2$, $2a_3$. The use of ellipsoids with equal values for flatness and elongation results

255 in a simplified value for $\bar{a} = \frac{L \cdot e}{2}$. A variable number of particles, comprised between a minimum of 10 and a maximum of 1000, has been used in all the simulations in order to achieve different values of γ. The results presented in Fig.9 show that the condition of maximum packing for ellipsoids can be easily reached only for values of $\gamma > 0.3$. For values of $\gamma < 0.3$ a larger number of rays must be adopted to reach the same degree of packing. For spheres and for the conditions analyzed in this example, the maximum packing is reached for $\gamma > 0.1$. As expected and it is clearly shown for spheres, the increase in number

260 of rays is not linearly related neither to the decrease of γ no the increase of τ. In the search of the maximum packing, it seems that a large number of rays can have larger benefits respect to the increase of the number of rotations, at least for the shape analyzed in this example. In addition, we notice that for spheres the minimum value for packing τ is about 20%; a value that correspond to a maximum porosity of 80% (see Eq.1). This threshold can be considered as a good estimation of the maximum value for porosity reachable by means of the SCARLET-1.0 algorithm.

265   3.2 Application of SCARLET-1.0 to packing problems

In this section two examples of possible applications of SCARLET-1.0 to different scientific problems are discussed. In the first example the attention is focused on the study of binary collisions of generic ellipsoids, i.e. aggregates made of just two particles. In particular the goal here is to derive how the porosity of the final product depends on the flatness and elongation of the ellipsoids and their relative sizes. Ellipsoids are often used to approximate complex shapes; this application is, thus, of

270 general interest for different scientific topics, such as particle sedimentation, particulate transport, etc. In the second example, SCARLET-1.0 is instead applied to the investigation of how the porosity evolves in time for volcanic ash aggregates. In this application, parameters of real volcanic aggregates are used to constrain the free parameters of the algorithm (Ω, N_r, N_o) and to draw important conclusions on the evolution of these objects.

3.2.1 Porosity for particle-particle collision of two generic ellipsoids

275 A collision of two distinct bodies is always the initial seed for binary aggregation processes (Jacobson, 2005). The aggregate will then evolve in time building up its own structure collision after collision, particle by particle. It is thus worth understanding the details of this initial stage, especially for what concerns how the 3D space is filled respectively by matter and voids, i.e. the porosity ϕ_{agg} (Eq.1). SCARLET-1.0 can be easily adapted to this problem and an arbitrary number of colliding shapes can be investigated in theory. However, in this section we limit our attention to particle-particle collisions of ellipsoids, due to

280 their vast application in science as an approximation of much more irregular shapes (Bagheri et al., 2015).

An ellipsoid is fully characterized by the knowledge of its three principal axes L, I, S (where $L \geq I \geq S$). Here we describe the intermediate and smallest axes, I and S respectively, in terms of flatness f and elongation e ($I = L \cdot e$ and $S = L \cdot e \cdot f$). This formalism allows describing the shape of any given ellipsoid as a single point in the "*flatness-elongation plane*" (FE-

plane), as shown in Fig. 10, where 16 different ellipsoids have been created with OpenSCAD (https://www.openscad.org/) to

285 map the FE-plane. When two ellipsoids collide, the resulting porosity ϕ_{agg} is also affected by the relative dimensions of the

two objects, i.e. the *size ratio* r_L, here described in terms of the ratio between the major principal axes $\left(r_L = \frac{L_c}{L_p}\right)$, where the

subscript letters c and p indicate the central object and the colliding particle respectively. It is worth noticing that in general r_L

is related to the ratio of the equivalent diameters $d_{eq} = (L\,I\,S)^{1/3}$ as follows: $r_d = \frac{d^c_{eq}}{d^p_{eq}} = r_L \cdot \left(\frac{e_c^2\,f_c}{e_p^2\,f_p}\right)^{1/3}$, that simplifies to $r_d = r_L$ in case of particles with the same flatness and elongation.

290 In this section we limit our attention to ellipsoids with the same shape. Two different packings have been investigated for five times: a loose packing, *setup 1*, for which ($\Omega = 1$, $N_r = 1$, $N_o = 1$) (Fig.11); a much denser packing, *setup 2*, for which ($\Omega = 270$, $N_r = 30$, $N_o = 10$) (Fig.12). In Fig.11 and Fig.12 the pair of numbers in brackets refer to the red labels in Fig.10, used to identify the shapes involved in the collisions.

Few comments can be made to the results reported in Fig.11 and Fig.12. Firstly, the loose packing generally shows significantly

295 more variability than the dense one (shadow areas in the figures), as it is expected given that *setup 1* completely depends on the initial random orientations of the bodies and the initial direction of collisions. Increasing the size ratios of the two objects or reaching the limiting case of a sphere ($f \to 1$, $e \to 1$), the variability goes to zero. Secondly, we notice that for collisions of particles of the same size ratio, where $f \geq 0.5$ and $e \geq 0.5$, the final porosity of the aggregate approaches $\phi_{agg} \approx 20\%$, which corresponds to the analytical solution of two contacting spheres of the same size (as also confirmed by the simulations (16;16)).

300 Interestingly, if the shape of the objects is not characterized by extreme low values of elongation or flatness (e.g. less than 0.5) the threshold of 20% represents a maximum value for porosities, regardless of the size ratios involved in the collision, as clearly shown in Fig. 11 and Fig. 12. Scenarios become more complicated if flatness or elongation (or both) are less than 0.5 (i.e. needle-like or extremely flat objects): in all these cases we notice a dependence from the initial setups used in SCARLET-1.0 and, in general, higher values of the porosities with respect to the previous cases. As an example, for shapes labelled as

305 (1;1) and $r_L = 1$, the setup 1 gives a porosity of $\phi_{agg} = 0.52 \pm 0.13$, compared to $\phi_{agg} = 0.42 \pm 0.09$ of setup 2. This last value is consistent with $\phi_{agg} = 0.44 \pm 0.08$ obtained averaging 30 runs with the setup ($\Omega = 270$, $N_r = 60$, $N_o = 30$).

3.2.2 Time evolution of porosities for volcanic ash aggregates

The term *volcanic ash aggregation* refers to the formation in the atmosphere or within a volcanic plume or cloud of

310 agglomerates of about 200-1000 μm in size due to the collision of smaller particles with a diameter generally less than 100 μm (Brown et al., 2012). As recently observed during the 2010 Eyjafjallajökull eruption (Iceland) (Bonadonna et al., 2011), volcanic ash aggregation plays a major role in ash sedimentation, as it affects the particle residence time in the atmosphere and it potentially leads to an overestimation of ash concentrations in the atmosphere if not correctly taken into account by the forecast models (Durant, 2015). Unfortunately, despite its importance, many questions still remain open due to the complexity

315   of the processes involved (Durant, 2015). A key factor is, for example, the porosity of volcanic ash aggregates, that strongly affects the sedimentation process and the residence time in the atmosphere. This parameter can be derived from field observations of the terminal velocity of the object before the impact on the ground. However, field observations are difficult to obtain and, in any case, they can only provide the porosity of the final structure of the aggregate, with no information about its evolution in time. In this section we apply SCARLET-1.0 to study the porosity evolution of volcanic ash aggregates,

320   expressed as a function of time, shapes and the grainsize distributions involved in the process.

Here we apply our algorithm to the reconstruction of two specific aggregates associated with two different eruptions: the sample labelled as *27Sk21* belongs to the eruption occurred on 3rd of August 2013 at Sakurajima volcano, (Japan) (Bagheri et al, 2016); the sample *EJ15* belongs to the eruption occurred on 5th of May 2010 at Eyjafjallajökull volcano (Iceland) (Bonadonna et al., 2011). Sample *27Sk21* was observed with a high-speed camera and was collected on an adhesive tape later

325   analyzed with a Scanning Electron Microscope (SEM); as a result, size and density could be derived before the impact on the adhesive tape from High-Speed videos, while the grainsize distribution could be derived from image analysis of the material collected on the adhesive tape (Bagheri et al, 2016). On the other hand, no High-Speed video is available for aggregate EJ15; as a result, only the grainsize distribution and a rough estimation of its size (about 200 μm) could be derived from the material left on the adhesive tape (Bonadonna et al., 2011).

330   These samples EJ15 and 27Sk21 were characterized as cored cluster (i.e. Particle Cluster 3 – PC3) and ash cluster (i.e. Particle Cluster 1 – PC1), respectively (Bagheri et al., 2016; Bonadonna et al., 2011). PC1 are clusters of ash particles with roughly all the same sizes of about 20-50 μm (Brown et al., 2012), whereas PC3 are characterized by the presence of an inner single object larger than 200 μm that has been coated by hundreds of smaller ash particles (Bagheri et al., 2016).

SCARLET-1.0 requires a vector of sizes as an input to locate the coating particles around the core, i.e. the detailed grainsize

335   distribution forming the aggregate to be reconstructed. However, large uncertainties are expected to affect the measured grainsize distribution, due to i) the mass lost in the air after the impact of the aggregate with the adhesive tape; ii) the mass lost during the transport of the samples from the field location to the laboratory; iii) the overlapping of particles in the SEM images of the coating, which lead to a loss of information deriving from the 2D representation of a 3D structure.

For aggregate 27Sk21, for which a constraint about size and density is available from the High-Speed video, we make the

340   assumption that all the losses in material only affect the total mass contained in the original coating population, but not the relative proportion of particles in each size-bin. This is equivalent to consider the measured grainsize distribution as fully equivalent to the original one, normalized for an unknown constant. The normalization constant is then derived matching the size and density of the virtual aggregate with those derived from field observations.

Volcanic ash particles are characterized by a variety of irregular shapes for which the STL representation of the surfaces is

345   usually not available (an example of an aggregate made of real 3D scanned volcanic particles is presented in Fig.A1 of the Appendix). However, Bagheri and Bonadonna (2016) demonstrated that the dynamical properties of irregular particles can be sufficiently well described in terms of equivalent ellipsoids. In this work we thus follow the simplified approach of classifying

the coating population as two distinct classes, i.e. coarse (between 63 μm and 2000 μm) and fine (≤63 μm) ash, each of them characterized by a single specific ellipsoid as displayed in Fig.13.

350 The two ellipsoids for coarse and fine ash are defined in terms of the three principal axes, derived averaging the L, I, S values of particles with size ≤63 μm and between 63 μm and 2000 μm from ash samples of the 2010 Eyjafjallajökull (Iceland) (Table 2). These ellipsoids are used here to represent coarse and fine ash for both samples EJ15 and 27sk21. The actual size of the *i-th* particle, $d_{eq}^i = (L_i I_i S_i)^{1/3}$, is used to scale the STL shape and to calculate the maximum length of the *i-th* ellipsoid $L = \alpha \beta d_{eq}^i$ (where $\alpha = L/I$ and $\beta = L/S$) as required by SCARLET-1.0 (see section 2.3).

355 In this virtual reconstruction we aim to reproduce the observed diameter of the aggregate (675 μm) and its final density (310 kg/m³) (Table 1), under the assumption that the measured grainsize of Fig.14b can be replicated N_A times in order to compensate for the expected mass loss. In our reconstruction particles below 32 μm in size are not taken into account in order to optimize the computational efficiency without affecting the final result. As a matter of fact, given that the volume scales as the third power of diameter, smaller particles add second order contributions to the final packing. Several combinations of the 360 solid angle Ω, number of rays N_r and number of Euler rotations N_o have been tested. However, only a loose packing (e.g. Ω = 1, N_r = 1, N_o ≤ 5) and N_A = 6 can approximatively reproduce the observed features, characterized by a porosity larger than 80% and an aggregate to core ratio of ≈ 3 (see table 1). In Fig.15 the variation of aggregate porosity is displayed as a function of collisions, for four different scenarios: ellipsoids whose sizes are randomly picked from the grainsize distribution (i.e. random displacement; red continuous line); ellipsoids sequentially located from the largest size to the smallest one (i.e. 365 sequential displacement; red dotted line); spheres whose sizes are randomly picked from the grainsize distribution (black continuous line); spheres sequentially located from the largest size to the smallest one (i.e. sequential displacement; black dotted line). Each line is the average of 5 repetitions and the final size of the virtual aggregate is ≈ 600 μm.

The second object to be reconstructed is the PC1 aggregate EJ15, that was collected on an adhesive tape during the 2010 Eyjafjallajökull eruption (Iceland) (Bonadonna et al. 2011). The reconstruction is strongly affected by the lack of High-Speed 370 video that does not allow to characterize the aggregate size and density before the impact with the adhesive paper. Therefore, in this application we simply focus on the time evolution of the porosity for different configurations of Ω, N_r, N_o, keeping the grainsize distribution equivalent to the one observed by Bonadonna et al., (2011) (i.e. no assumption has been made on the mass loss). In addition, the same ellipsoids reported in Table 2 have been used also for this application. The results of different simulations are reported in Fig.16, together with the 3D visualization provided by SCARLET-1.0. In the following we refer to 375 the setup (Ω = 50, N_r = 30, N_o = 10) as "tight packing" and to the setup (Ω = 1, N_r = 1, N_o = 1) as "loose packing".

The range of final porosities spans from a maximum value of about 73-76% for the loosest packing down to 30% for the tight packing with Ω = 50, N_r = 30, N_o = 10. If we compare the curves of Fig.16 with values of porosities reported in literature

for PC1 aggregates, comprised between ≈80-97% (Brown et al., 2012; Gabellini et al., 2020; Table 1), we can deduce that only the loose packing can realistically reproduce the observed features, such is the case for PC3 aggregate.

380 Finally, it is important to notice how the plateau in porosity for EJ15 is reached roughly after the same number of collisions (≈50-70 collisions) both for the tight and loose packing.

4. Discussion

4.1 SCARLET-1.0: general comments on the operational use

385 SCARLET-1.0 has been specifically designed for the simulation of aggregation of complex shapes. In order to treat the intersection problem between colliding objects, the algorithm takes advantage of the sphere composite representation of a body. This means that if a given shape is described in terms of N_{sph} spheres and if N_p particles collide, the final aggregate will be described as $N_{sph} \times N_p$ spheres. Given that the intersection is evaluated for each step in the inward and outward movement before placing each single particle (see Fig.5), the use of a large number of spheres combined with a large number of particles

390 can significantly increase the computational time of the package. As a general advice, we recommend an appropriate use of the preprocessing code *fromStlToSpheres* and the Boolean variable O_{ext} to find a good balance between an accurate description of the shape and a reduced number of spheres, i.e. faster computational times.

An important point that the external user should always consider is how the parallelization of the algorithm works. The parallelization of the code is based on the execution of simultaneous for-loop iterations on the different available workers

395 (*parfor*). However, Matlab does not allow for nested parallel loops, which results in a parallelization that can only be performed at one level. As shown in Fig.5 in the present release of SCARLET-1.0 this occurs inside the loop dedicated to describing particle rotations along a single ray. This means that if few rotations are set in the simulation the parallelization will not substantially speed up the code.

As outlined in the introduction, SCARLET-1.0 has been created to simulate central collisional processes, in which particles

400 collide from random directions and find their final location after spanning a given solid angle Ω. This occurs in different natural phenomena as the ones discussed in the previous examples and in the introduction. According to the setup imposed by the external user, the package can simulate various degrees of packing by simply changing the solid angle of investigation (Ω), the number of rays (N_r) and the number of rotations (N_o). For what concerns the maximum packing, Fig. 9 shows that objects characterized by a particle-to-aggregate size smaller than 0.17-0.25 require more computational efforts to match the maximum

405 packing condition than aggregates with smaller sizes or larger components (at least for the shapes investigated). In these cases, the use of multiple processors allows having a large number of rotations tested, which can finally result in a better packing of the object in a lower computational time. However, in several applications of interest, such as the one reported in section 3.2.2, the maximum packing is not the final aim of the virtual reconstruction and a much less dense packing is required to correctly mimic the phenomenon under analysis.

410

4.2 SCARLET-1.0 applied to the study of porosity for volcanic ash aggregates

In section 3.2.2 we presented SCARLET-1.0 applied to the study of porosity in volcanic ash aggregates. This is an interesting example of how the algorithm can provide answers not only about the packing of shapes but also on its evolution in time, an

415 information that is usually missing in field or lab observations. The study has been focused on the virtual reconstruction of two specific aggregate structures: PC3 type (sample 27sk21) ,made of a central object five to ten times larger than the particle sizes involved in the coating; PC1 type (sample EJ15), made of particles with roughly the same size.

The reconstruction of the PC3 aggregate (Fig. 15) has shown that the modelled aggregate porosity ($\approx$73%) is almost 10% smaller than the observed one, for the same size of the aggregate. This discrepancy can be due to the approximations made in

420 the description of particle shapes. An alternative explanation can suggest the presence of liquid bridges in the original structure that alter the contact condition between particles and finally creates a looser packing of the coating than (Gilbert and Lane, 1994). As a matter of fact, the setup used in the simulation that corresponds to *loose packing* (e.g. $\Omega = 1$, $N_r = 1$, $N_o \leq 5$) describes a particle coming from a random direction, randomly oriented, that it is going to be located on the surface of the core immediately after the first contact. In this configuration no minimization is done on the packing of the aggregate and it should

425 thus reproduce a condition of maximum porosity for the object. However, the fact that the observed porosity is even larger than the simulated one could be interpreted as a violation of the contact condition among particles as assumed by SCARLET-1.0. This violation is theoretically possible if liquid bonds act as a sticking medium between particles.

The importance of loose packing for the typologies under analysis is also confirmed by the PC1 sample (EJ15), for which the computed porosity of 73-76% for the loosest configuration matches with the lower bound of porosity reported in literature

430 (e.g. 80%). This work thus suggests that the condition of maximum packing should not be seen as the natural tendency in the context of volcanic ash aggregation, at least for PC1 and PC3 types.

A second important observation concerning the virtual reconstruction of sample 27sk21 is the use of $N_A = 6$ to reproduce the observed features of the aggregate. This practically means that the number of particles analyzed at the SEM is underestimated of about 1/6 with respect to the original coating. This implies that a realistic quantification of the number of particles stuck

435 over the surface of the PC3 aggregate under analysis was probably of the order of a few thousands.

For what concerns the differences between random or sequential displacement of particles, we notice from Fig. 15 and Fig. 16 that random displacement initially produces lower porosities with respect to a sequential displacement of particles from large to small sizes. This can be explained thinking of the large number of voids that is created when particles of similar sizes are packed together. On the contrary, smaller particles can easily fill the voids resulting in lower porosities. However, a non-trivial

440 observation from Fig. 15 and Fig. 16 is that the final value of the aggregate porosity weakly depends on the "*history*" of the packing, as long as the number of coating particles is wide enough. In addition, Fig. 15 suggests that the use of ellipsoids does

not produce remarkable differences in the packing with respect to spheres (about $\approx 5\%$ in the porosity). As expected, in case of a loose packing (i.e. $\Omega = 1$, $N_r = 1$, $N_o \leq 5$) the use of ellipsoids produces slightly higher porosities with respect to spheres. An interesting result emerges when both PC1 and PC3 aggregates are compared on the same plot for the same configuration of (Ω, N_r, N_o (Fig.17). It is evident how the PC1 aggregate reaches the plateau in porosity much faster than the PC3 one. This can be explained noticing that the presence of a large core inside the PC3 somehow slows down the increase in porosity of the object, which takes more time and more collisions to reach comparable levels of densities. The consequence of this observation is not only that PC1 objects can reach low densities faster than PC3 aggregates, but that if there are enough time and collisions to let a PC3 aggregate grow, its final density can be as low as a PC1. In other words, the presence of an inner large particle (the core) at its center is diluted in time by the increasing coating, that finally drops the overall density down to much lower values than that of the core.

5. Conclusions

In this work we presented SCARLET-1.0, a MATLAB package aimed to simulate aggregate formation in central collision processes, starting from an arbitrary population of sizes and shapes. 3D bodies are approximated by a set of non-overlapping spheres, resulting in an analytical description of their intersections. The code has been thought for all those scientific applications where packing strongly depends on the complex shapes of the objects involved, such as planetary formation, volcanic ash sedimentation, formation of snowflakes and environmental pollution. The main output of the code is the final porosity of the aggregate, calculated as the ratio of the inner voids and the external volume of the agglomerate.

In terms of novelty of the code we can conclude that:

1) One of the most appealing features of the code is the use of the STL format to import 3D shapes. The STL standard is nowadays one of the most widespread formats in the community, due to its strong connection to the world of 3D printing. This results in a large availability of software to design arbitrary shapes and in the possibility to import 3D scanned surfaces from real samples.

2) An interesting aspect of the package is the production as an output of the STL file on the modelled virtual aggregate. This practically means that the final object can be 3D printed and used in real contexts, such as laboratory investigations.

3) The external user can easily control the basic aspects of the algorithm simply playing on few parameters, such as (Ω, N_r, N_o) for the minimization process or Boolean quantities for the computation of porosity in time.

4) The algorithm can be easily modified to the study of different problems, such as the study of aggregate-aggregate collision.

The examples analyzed in the manuscript show the versatility of the code and its potential. In terms of the two specific applications of the code to aggregation processes we can conclude that:

475   1) The resulting porosity of an ellipsoid-ellipsoid collision has a maximum of 20% for objects of the same size and with flatness and elongation larger or equal to 0.5. Needle-like or flat bodies characterized by flatness and elongation less than 0.5 can result in higher porosities.

2) The virtual reconstruction of volcanic ash aggregates shows how PC1 and PC3 types are better represented in terms of a loose packing in which particles stop their relative motion after touching. This can be explained by a very efficient

480   binding mechanism or a reduced impact velocities of the colliding objects.

3) PC1 aggregates can reach high porosities faster than PC3, i.e. after a lower number of collisions. However, if a large number of particles can stick to the central object, the final porosity of PC3 type tends to be similar to the PC1 sample.

4) Random and sequential packing produce differences of about 10% in the porosity in the analyzed samples.

485   To sum up, SCARLET-1.0 an opportunity to simulate aggregates formation in central collision processes, regardless of the nature of the objects and the complexity of the shapes involved.

Appendix

490 Examples of virtual aggregates created with SCARLET-1.0 with different shapes. The goal is to show the potential use of the algorithm to a large variety of cases.

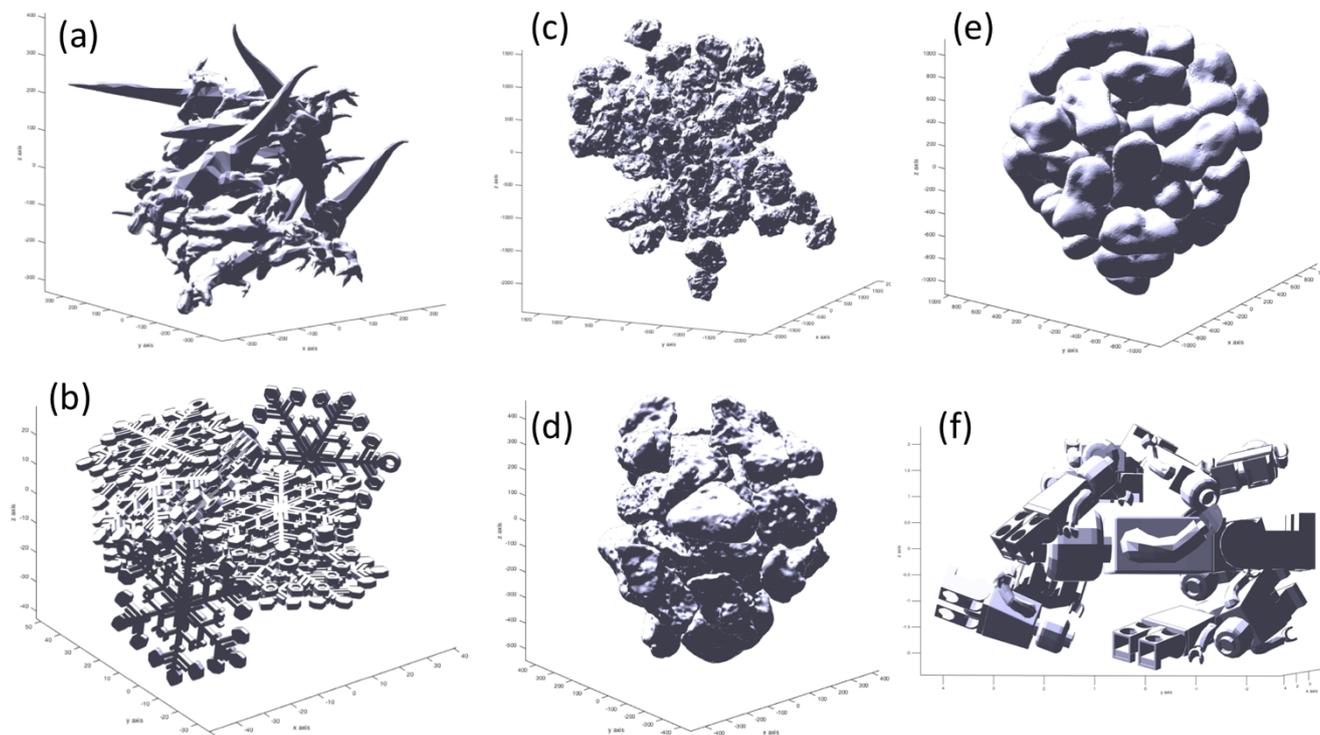



Figure A1. Virtual aggregates made of different initial stl files. (a) aggregate made from a stl file of a t-rex; (b) aggregate made from
495 **a stl file of snowflakes; (c) aggregate made of real volcanic particles, scanned with a 3D laser scanning machine (loose packing); (d) the same shapes and objects of (c) but with a more tight packing; (e) aggregate made using as a stl file the asteroid Castalia; (f) aggregate made of LEGO characters.**

500

505

Author contribution

510

- Eduardo Rossi: he developed the algorithm and wrote the first version of the paper and the user guide
- Costanza Bonadonna: she provided the dataset for section 3.2.2 and 4.2, revised the manuscript and improved the writing of the manuscript

515 **Code availability**

The code has been developed for scientific purposes and it can be freely used if the user has a Matlab license. The code is made available at the following Zenodo link for the revision process of GMD (https://doi.org/10.5281/zenodo.4307995).
In future, new updates of the code will be released at the Github page https://github.com/EduardoRossiScience/SCARLET.

520

Competing interests

The authors declare that they have no conflict of interest.

Acknowledgements

525

We would like to thank Marco Pistolesi and Raffaello Cioni for their help in reconstructing the grain size distribution of the aggregate EJ15. This work was supported by the Swiss National ScienceFoundation (Grant No. 200021_156255).

References

Bagheri, G. H., Bonadonna, C., Manzella, I., and Vonlanthen, P.: On the characterization of size and shape of irregular
530 particles, Powder Technol., 270, 141-153, 2015.

Bagheri, G. and Bonadonna, C.: On the drag of freely falling non-spherical particles, Powder Technol., 301, 526-544, 2016.

Bagheri, G., Rossi, E., Biass, S., and Bonadonna, C.: Timing and nature of volcanic particle clusters based on field and
535 numerical investigations, J. Volcanol. Geotherm. Res., 327, 520-530, 2016.

Bonadonna, C., Genco, R., Gouhier, M., Pistolesi, M., Cioni, R., Alfano, F., Hoskuldsson, A., and Ripepe, M.: Tephra sedimentation during the 2010 Eyjafjallajokull eruption (Iceland) from deposit, radar, and satellite observations, J. Geophys. Res.-Solid Earth, 116, 20, 2011.

540

Brauer, M., Avila-Casado, C., Fortoul, T. I., Vedal, S., Stevens, B., and Churg, A.: Air pollution and retained particles in the lung, Environ. Health Perspect., 109, 1039-1043, 2001.

Brown, R. J., Bonadonna, C., and Durant, A. J.: A review of volcanic ash aggregation, Phys. Chem. Earth, 45-46, 65-78, 2012.

545

Chopard, B., Nguyen, H., and Stoll, S.: A lattice Boltzmann study of the hydrodynamic properties of 3D fractal aggregates, Math. Comput. Simul., 72, 103-107, 2006.

Conway, J. and Sloane, NJA, Sphere Packings, Lattices and Groups, Springer New York, 1998.

550

Cuq, B., Mandato S., Jeantet R., Saleh K., Ruiz T., Agglomeration/granulation in food powder production, Woodhead Publishing, 150-177, 2013

Dacanal, G. C., Feltre, G., Thomazi, M. G., and Menegalli, F. C.: Effects of pulsating air flow in fluid bed agglomeration of
555 starch particles, J. Food Eng., 181, 67-83, 2016.

Dominik, C., Blum, J., Cuzzi, J., and Wurm, G.: Growth of Dust as the Initial Step Toward Planet Formation, 2006.

Donev, A., Cisse, I., Sachs, D., Variano, E., Stillinger, F. H., Connelly, R., Torquato, S., and Chaikin, P. M.: Improving the
560 density of jammed disordered packings using ellipsoids, Science, 303, 990-993, 2004.

Durant, A. J.: Toward a realistic formulation of fine-ash lifetime in volcanic clouds, Geology, 43, 271-272, 2015.

Evans, K. E. and Ferrar, M. D. A. A.: The packing of thick fibres, Journal of Physics D Applied Physics, 22, 354, 1989.

565

Filippov, A. V., Zurita, M., and Rosner, D. E.: Fractal-like aggregates: Relation between morphology and physical properties, J. Colloid Interface Sci., 229, 261-273, 2000.

Gabellini P., Rossi E., Bonadonna C., Pistolesi M., Bagheri G., Cioni R.,, Physical and aerodynamic characterization of particle
570 clusters at Sakurajima volcano (Japan). Frontiers in Earth Science, doi: 10.3389, 2020

Gilbert, J. S. and Lane, S. J.: The origin of accretionary lapilli, Bulletin of Volcanology, 56, 398-411, 1994.

Gravner, J. and Griffeath, D.: Modeling snow-crystal growth: A three-dimensional mesoscopic approach, Phys. Rev. E, 79, 575   18, 2009.

Hales, T. C.: A proof of the Kepler conjecture, Ann. Math., 162, 1065-1185, 2005.

Imaeda, Y. and Ebisuzaki, T.: Tandem planet formation for solar system-like planetary systems, Geosci. Front., 8, 223-231, 580   2017.

Ishizaka, M. (1993). An accurate measurement of densities of snowflakes using 3-D microphotographs. Annals of Glaciology, 18, 92-96. doi:10.3189/S0260305500011319

585   Jacobson, M. Z. U.: Fundamentals of Atmospheric Modeling, Cambridge University Press, 2005.

Jia, X. and Williams, R. A.: A packing algorithm for particles of arbitrary shapes, Powder Technol., 120, 175-186, 2001.

Kessler, D. A., Koplik, J., and Levine, H.: Numerical simulation of two dimensional snowflake, Phys. Rev. A, 30, 2820-2823, 590   1984.

Lumme, K. and Rahola, J.: Light scattering by porous dust particles in the discrete-dipole approximation, Astrophys. J., 425, 653-667, 1994.

595   Man, W. N., Donev, A., Stillinger, F. H., Sullivan, M. T., Russel, W. B., Heeger, D., Inati, S., Torquato, S., and Chaikin, P. M.: Experiments on random packings of ellipsoids, Phys. Rev. Lett., 94, 4, 2005.

Maruyama, K. and Fujiyoshi, Y.: Monte Carlo simulation of the formation of snowflakes, J. Atmos. Sci., 62, 1529-1544, 2005.

600   Min, M., Waters, L., de Koter, A., Hovenier, J. W., Keller, L. P., and Markwick-Kemper, F.: The shape and composition of interstellar silicate grains, Astronomy & Astrophysics, 462, 667-676, 2007.

Nguyen, H. P., Chopard, B., and Stoll, S.: Lattice Boltzmann method to study hydrodynamic properties of 2D fractal aggregates. In: Computational Science - Iccs 2003, Pt I, Proceedings, Sloot, M. A. P., Abramson, D., Bogdanov, A. V., 605   Dongarra, J. J., Zomaya, A. Y., and Gorbachev, Y. E. (Eds.), Lecture Notes in Computer Science, Springer-Verlag Berlin, Berlin, 2003.

Nguyen, H. P., Chopard, B., and Stoll, S.: Hydrodynamic properties of fractal aggregates in 2D using Lattice Boltzmann simulation, Futur. Gener. Comp. Syst., 20, 981-991, 2004.

610

Ning, C. and Reiter, C. A.: A cellular model for three-dimensional snow crystallization, Comput. Graph.-UK, 31, 668-677, 2007.

Nolan, G. T. and Kavanagh, P. E.: Random packing of nonspherical, Powder Technol., 84, 199-205, 1995.

615

Ohno, K., Okuzumi, S., and Tazaki, R.: Clouds of Fluffy Aggregates: How They Form in Exoplanetary Atmospheres and Influence Transmission Spectra, Astrophys. J., 891, 19, 2020.

Ormel, C. W., Spaans, M., and Tielens, A.: Dust coagulation in protoplanetary disks: porosity matters, Astronomy & Astrophysics, 461, 215-232, 2007.

620

Poon, J. M. H., Immanuel, C. D., Doyle, F. J., and Litster, J. D.: A three-dimensional population balance model of granulation with a mechanistic representation of the nucleation and aggregation phenomena, Chem. Eng. Sci., 63, 1315-1329, 2008.

625 Pumir, A. and Wilkinson, M.: Collisional Aggregation Due to Turbulence. In: Annual Review of Condensed Matter Physics, Vol 7, Marchetti, M. C. and Sachdev, S. (Eds.), Annual Review of Condensed Matter Physics, Annual Reviews, Palo Alto, 2016.

Reiter, C. A.: A local cellular model for snow crystal growth, Chaos Solitons Fractals, 23, 1111-1119, 2005.

630

Shi, Y. Y., Ji, Y. F., Sun, H., Hui, F., Hu, J. C., Wu, Y. X., Fang, J. L., Lin, H., Wang, J. X., Duan, H. L., and Lanza, M.: Nanoscale characterization of PM2.5 airborne pollutants reveals high adhesiveness and aggregation capability of soot particles, Sci Rep, 5, 10, 2015.

635 Szilvasi-Nagy, M. and Matyasi, G.: Analysis of STL files, Math. Comput. Model., 38, 945-960, 2003.

Weaire, D. and Aste: The Pursuit of Perfect Packing, CRC Press, 2000.

Williams, R. A. and Jia, X. D.: A new method for prediction of bulk particle packing behavior for arbitrary-shaped particles
640 in containers of any shape, Part. Sci. Technol., 21, 195-205, 2003.

Yurkin, M. A. and Hoekstra, A. G.: The discrete-dipole-approximation code ADDA: Capabilities and known limitations, J. Quant. Spectrosc. Radiat. Transf., 112, 2234-2247, 2011.

645

Figures

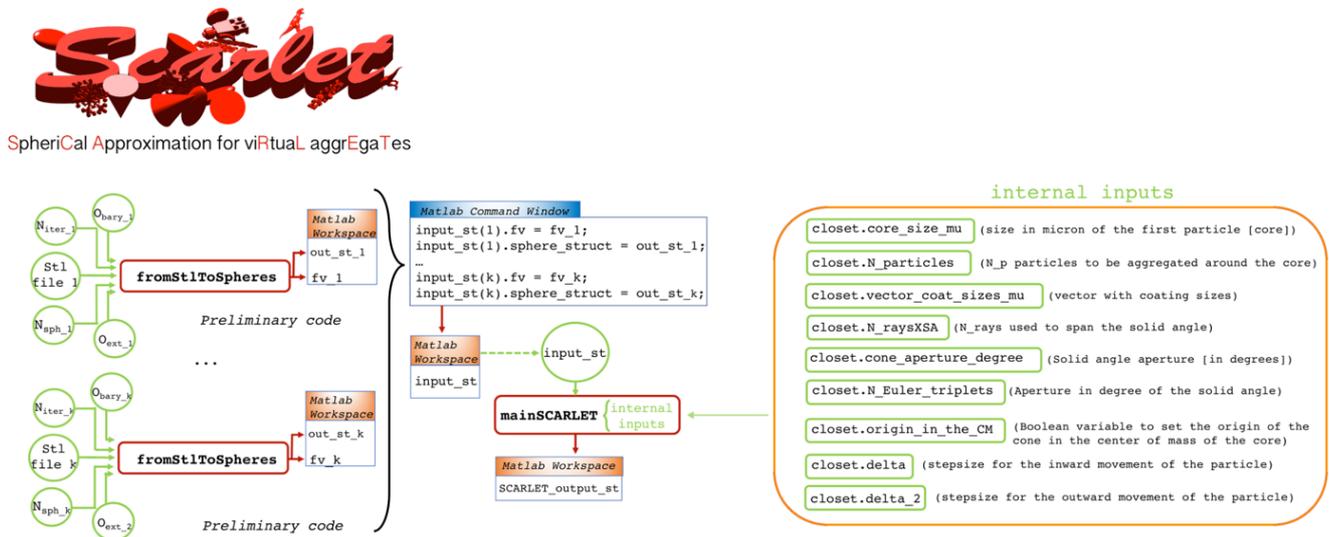



Figure 1. General scheme of SCARLET-1.0: on the left, the preprocessing code "*fromStlToSpheres*". On the right and in the center, the main function "*mainSCARLET*". The preprocessing code is needed to convert each 3D shape into a set of not overlapping spheres. The function *fromStlToSpheres* produces two output structures for each STL file, *out_st* and *fv*, which must be assembled into a new structure, *input_st*, as shown here. *input_st* represents the unique external input for *mainSCARLET*. A set of internal inputs control the details of each simulation, such as the number of coating particles involved, their sizes and the degree of packing.

655

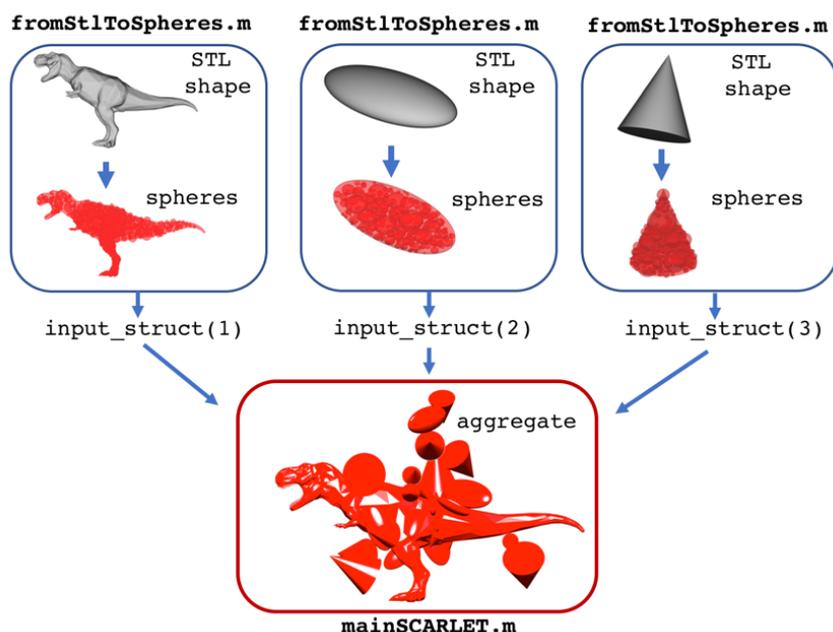



Figure 2. Example showing how SCARLET-1.0 creates a loosely packed aggregate made of 20 ellipsoids and cones of different sizes around a central object (a T-rex).

660

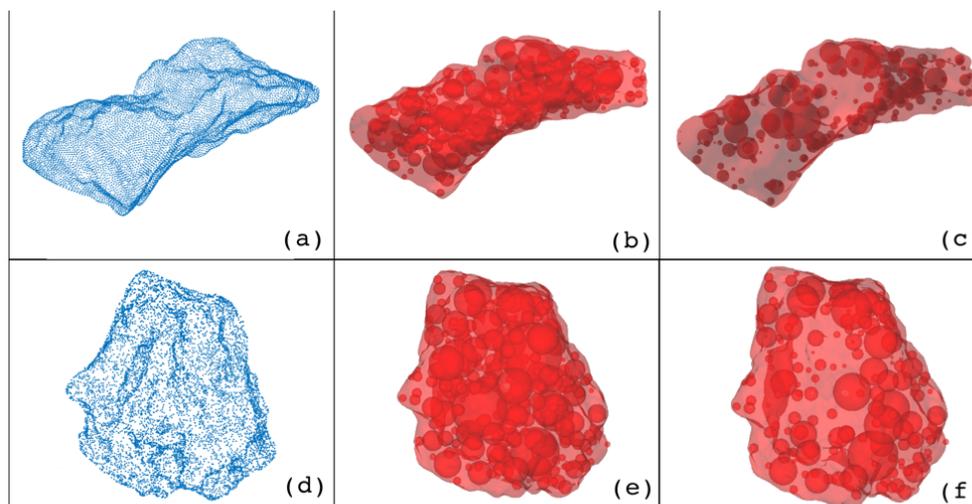



Figure 3. Application of the pre-processing routine *fromStlToSpheres* to two different volcanic particles. Blue points in (a) and (d) represent the vertices of the triangles used to define the 3D surfaces, as present in the original STL files. (b) and (e) illustrate the final process of filling the volumes with 300 spheres, placed randomly inside the objects. (c) and (f) show the most external spheres obtained using the built-in Matlab function *boundary*.

665

670

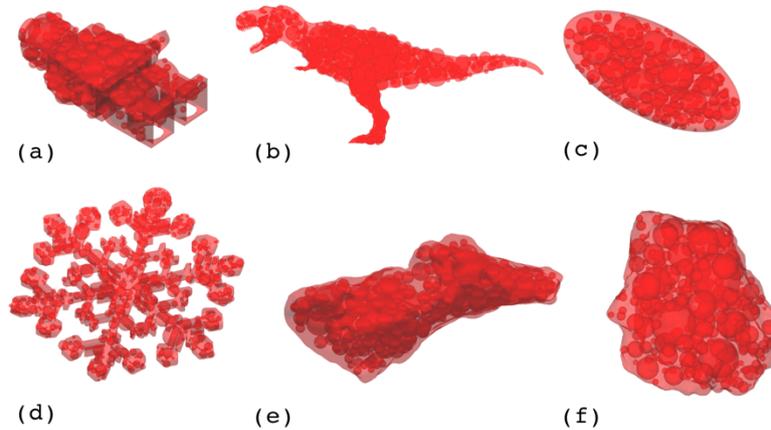
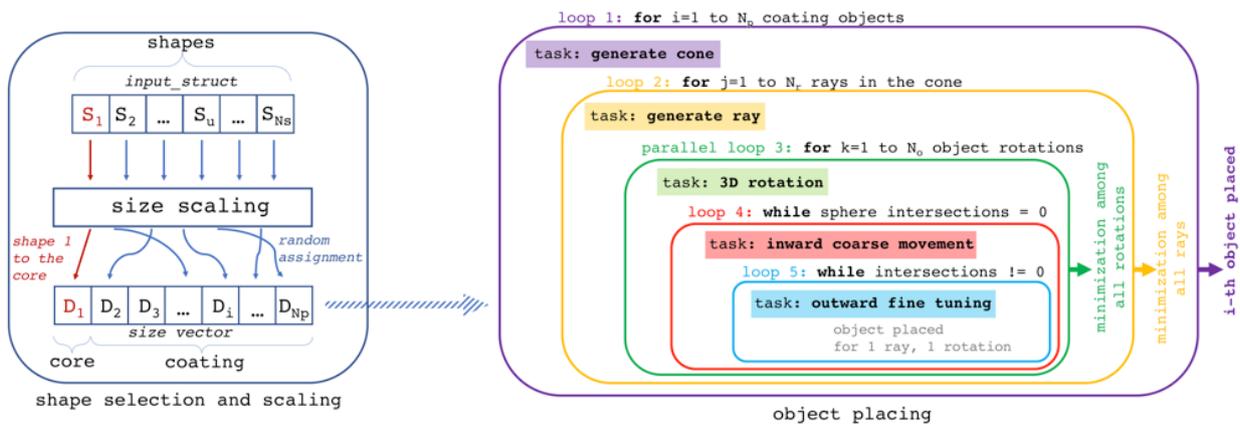



Figure 4. Application of the pre-processing routine *fromStlToSpheres* to objects characterized with different surfaces (i.e. convex hulls or not) that will be used in this paper. In (a) a LEGO toy, in (b) a T-rex, in (c) an ellipsoid, in (d) a snowflake, in (e) and (f) two different volcanic particles. No scale has been reported for this image since it is not important for the pre-processing. In fact, all the
675 **objects will be rescaled by the mainSCARLET function.**

680 **Figure 5. Scheme followed by SCARLET-1.0 for particle placing. On the left, the first part of the algorithm where shapes are randomly assigned to each coating particle and scaled to the actual particle size is shown. On the right, the series of five nested loops that are in charge of particle placing are presented.**

685

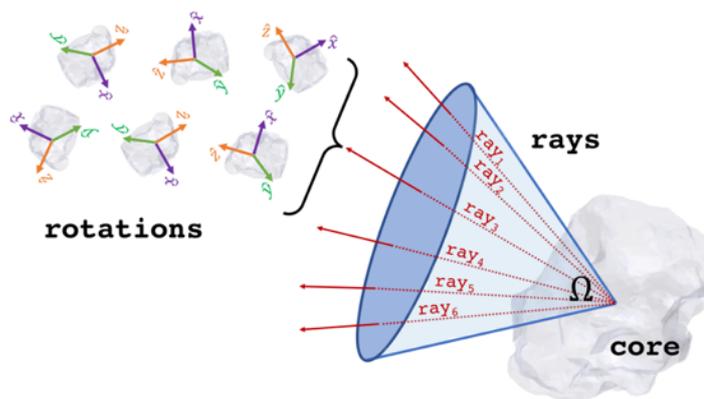



Figure 6. Cone, rays and rotations associated with the placing of one single coating particle.

690

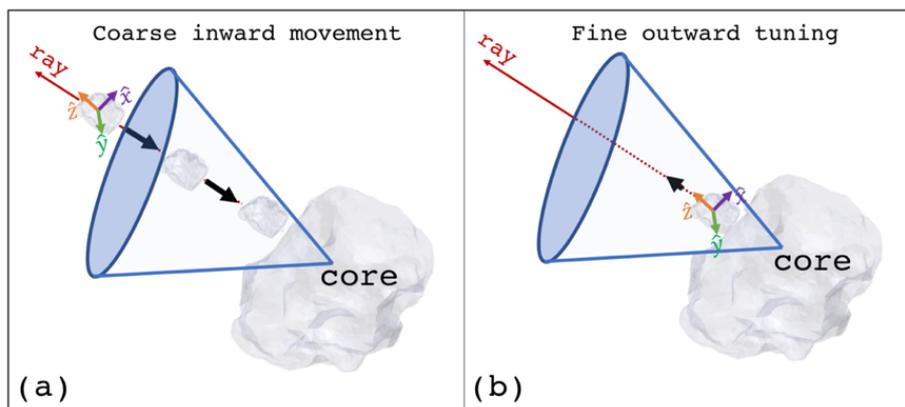



Figure 7. Coarse inward movement and fine outward tuning associated with loop 4 and loop 5 respectively.

695

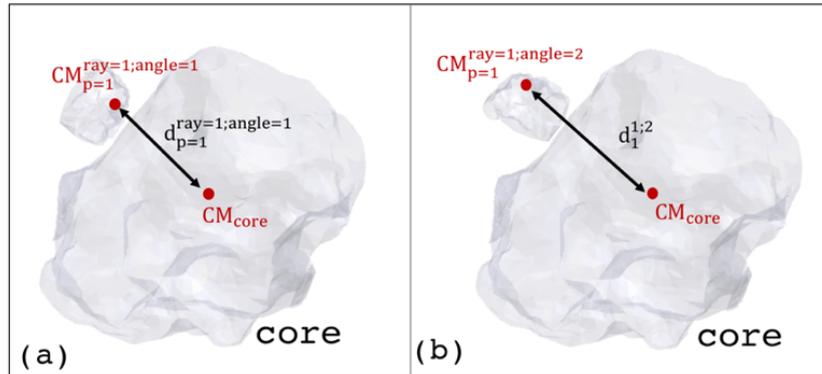



Figure 8. Example of minimization process over two rotated particles. The selected particle is the one with minimum distance with respect to the center of mass of the core.

700

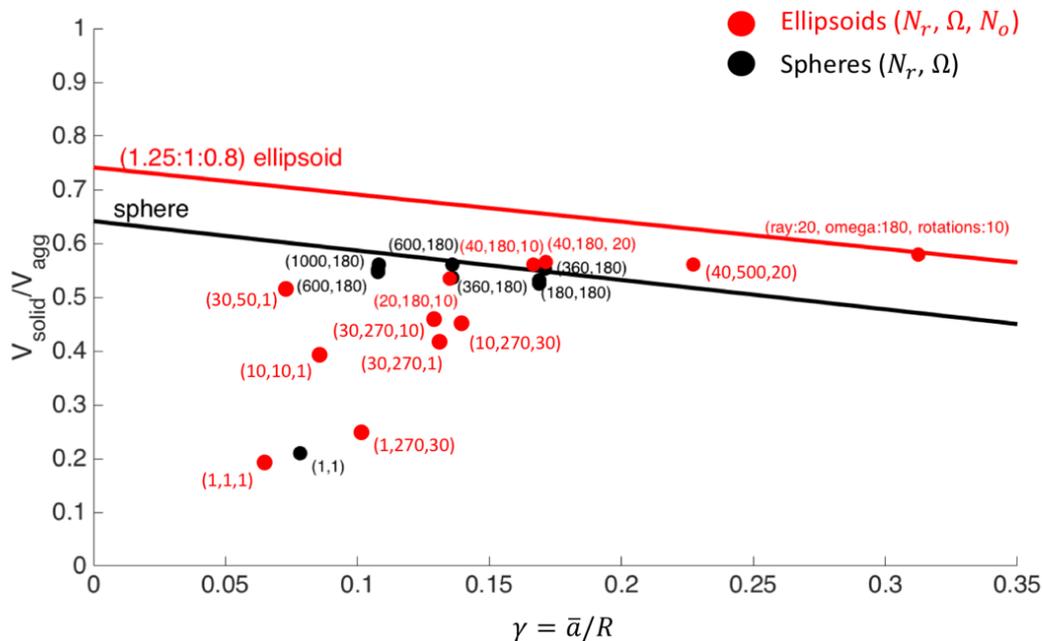



Figure 9. Packing $\left(\tau = \frac{V_{solid}}{V_{agg}}\right)$ for spheres and ellipsoids for different values of solid angles, number of rays and number of rotations respectively, i.e. (Ω, N_r, N_o). The red line indicates the maximum packing for ellipsoids with axes with a ratio of (1.25:1:0.8). The black line indicates the maximum theoretical packing for spheres. Red dots are simulations made for ellipsoids, black dots for spheres. The red brackets show the values of (Ω, N_r, N_o) used for a single simulation; the black brackets show the values of (Ω, N_r),
705 **since no rotation of the shape is investigated for spheres. All the values are expressed as a function of a/R, where a is the average radius of the ellipsoid/sphere and R is the radius of the sphere with same external volume of the aggregate.**

710

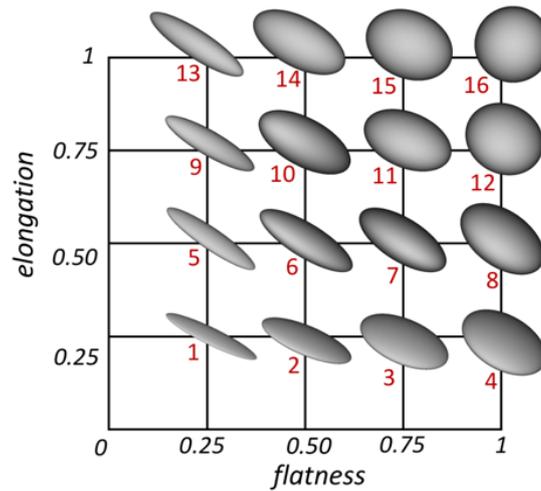



Figure 10 The *flatness-elongation plane* (FE-plane) and the associated shape of the investigated ellipsoid. The red labels in the figure are introduced to characterize each particle-particle collision univocally.

715

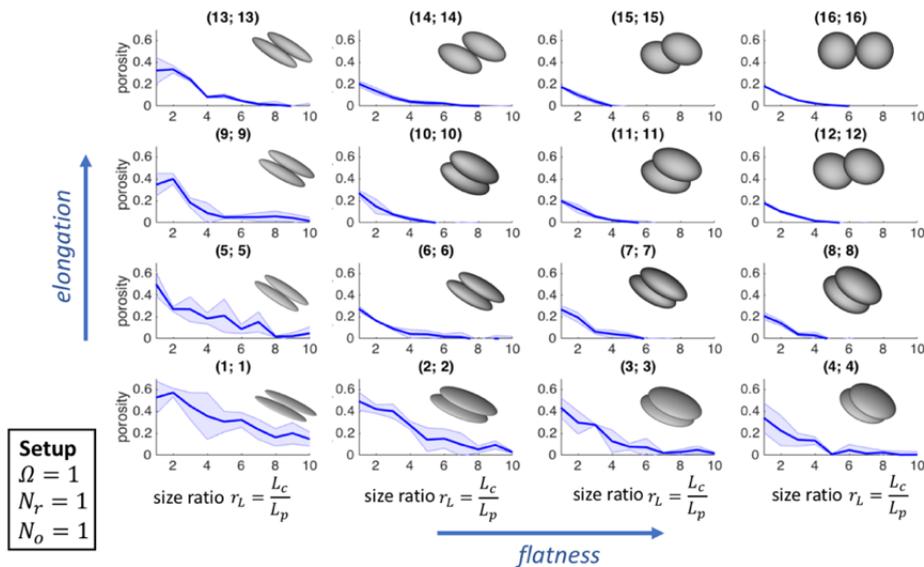



Figure 11 Porosity of two colliding ellipsoids of same shape as a function of the size ratio r_L. The setup used in SCARLET-1.0 for these simulations is ($\Omega = 1$, $N_r = 1$, $N_o = 1$). The pair of numbers in the brackets refer to the shapes labelled in Fig. 10 The
720 **shadowed area is the outcome of five simulations.**

Figure 12 Porosity of two colliding ellipsoids ofsame shape as a function of the size ratio r_L. The setup used in SCARLET-1.0 for these simulations is ($\Omega = 270, N_r = 30, N_o = 10$). The pair of numbers in the brackets refer to the shapes labelled in Fig. 10 The shadowed area is the outcome of five simulations.

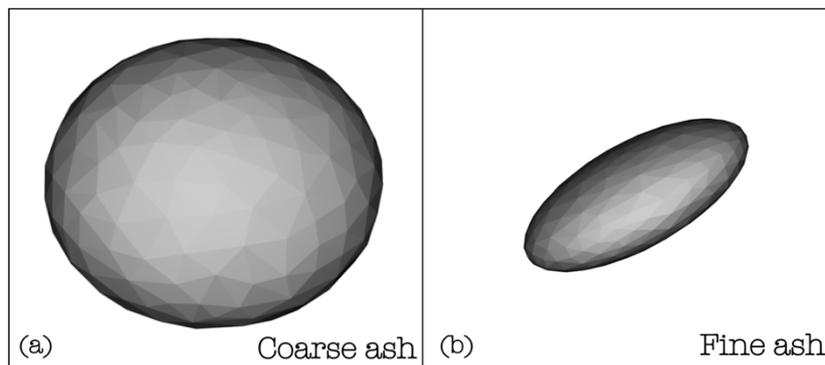



Figure 13. Ellipsoids used to characterize coarse (a) and fine (b) ash in the virtual reconstruction of the aggregates 27Sk21, 21Sk21, 22Sk20, 19Sk19 observed at Sakurajima Volcano (Japan) the 2nd and 3rd of August 2013.

(a)   (b)

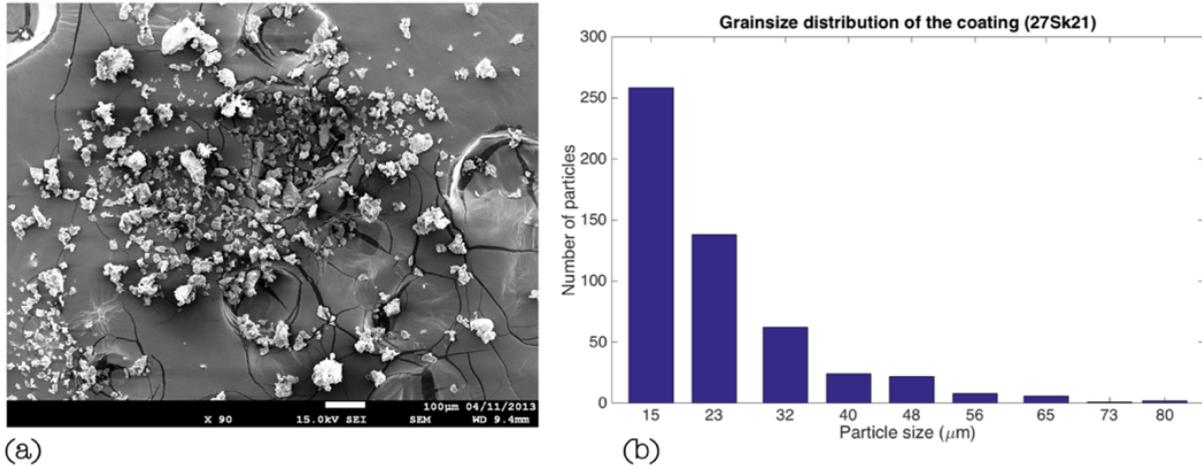



Figure 14. (a) SEM image of the coating particles relative to the aggregate 27sk21 (the big core escaped from the tape); (b) Grainsize distribution derived from the SEM image (Bagheri et al. 2016). In the present work we did not use particles below 32 μm in size: these allows optimizing the computational efficiency without affecting the final result (since the volume scales as the third power of diameter, smaller particles add second order contributions to the final volume).

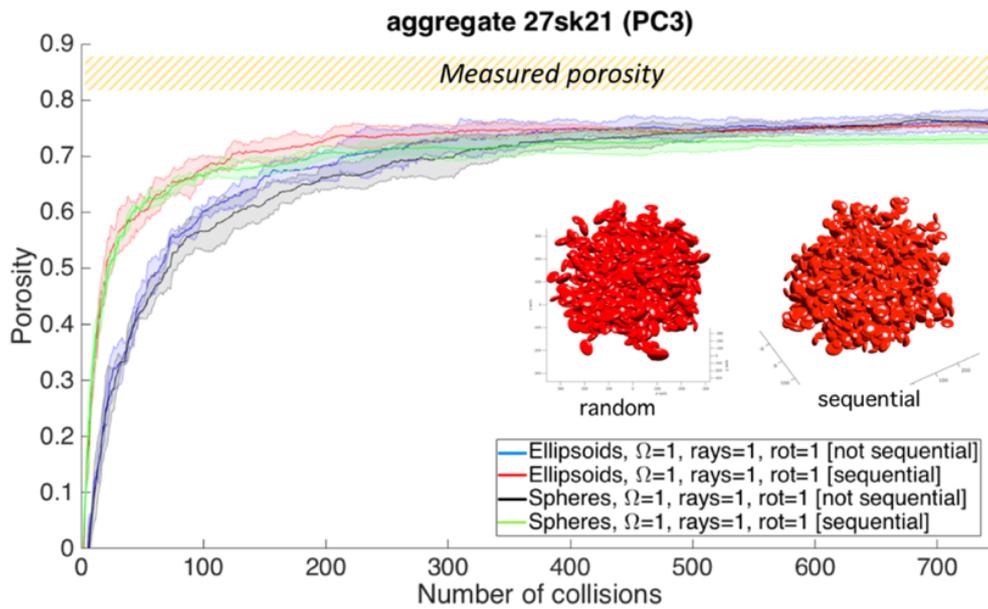



Figure 15. Evolution in time of the porosity for the aggregate 27Sk21, classified as PC3 from field observations (Bagheri et al. 2016). The final reconstruction of the aggregate is reported in red within the frame of the figure, both for a sequential and a random displacement of the coating particles. The core size – not visible in the figure – is 270 μm wide. The final size of the object matches the observed one (≈670 μm). The shaded area represents the 68% of confidence around the mean over five repetitions.

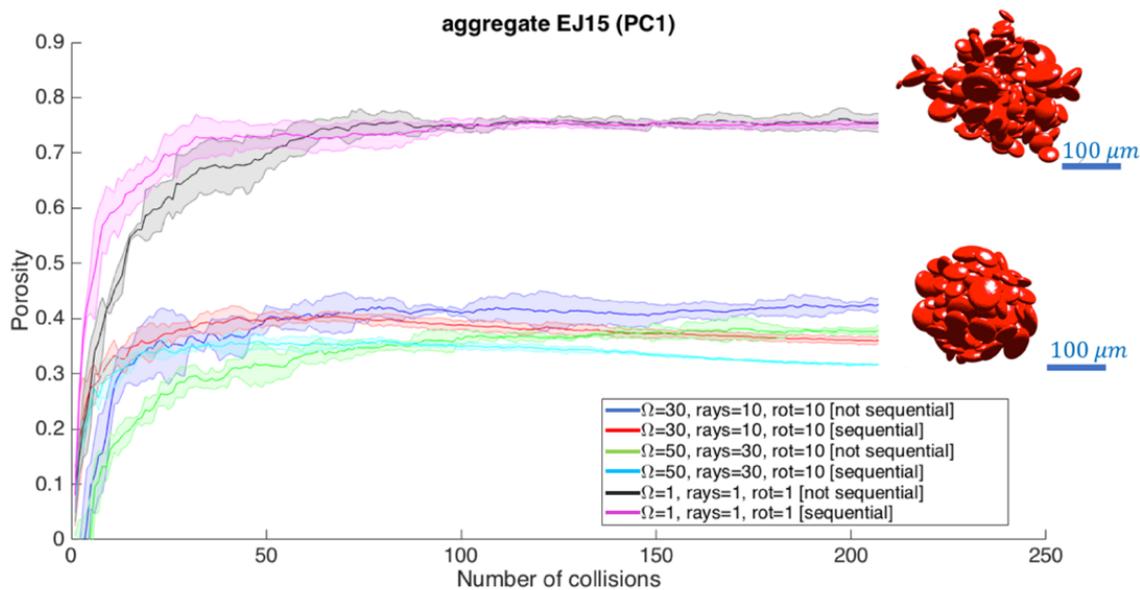



Figure 16 Time evolution of the porosity for aggregate EJ15 (PC1), collected during the 2010 Eyjafjallajökull eruption (Iceland) (Bonadonna et al. 2011). The virtual reconstructions are based on the same ellipsoids of Fig.13. The shaded areas represent the 68% of confidence around the mean over five repetitions.

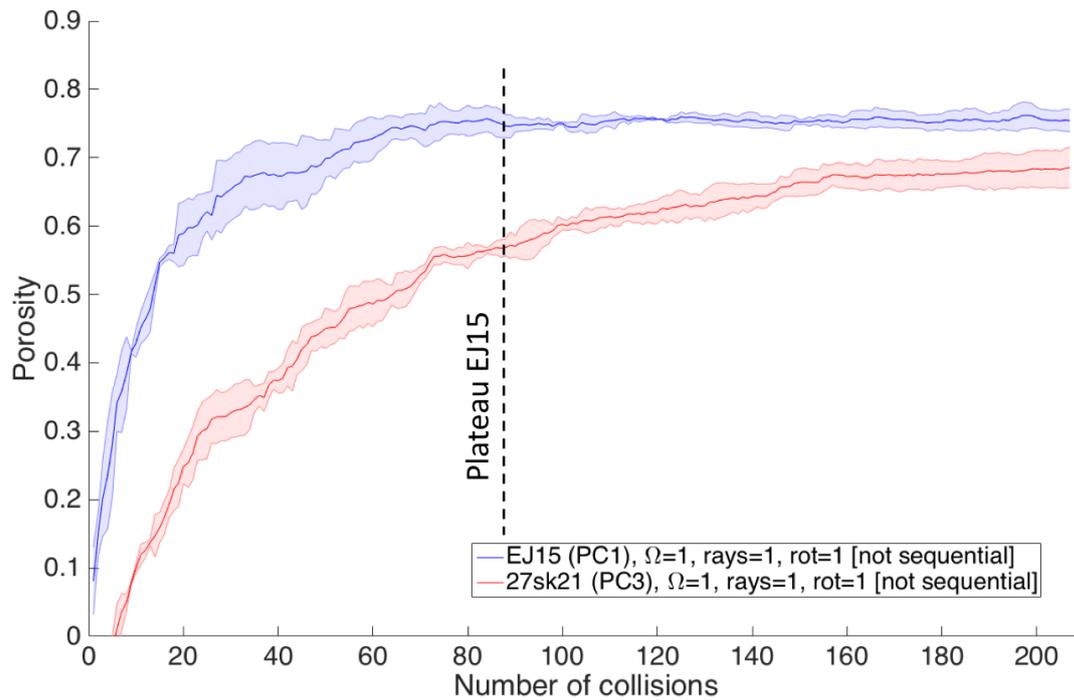



Figure 17 Evolution of the porosity for aggregates EJ15 (PC1) and 27sk21 (PC3) for the same configuration of SCARLET-1.0 (Ω = 1, N_r = 1, N_o = 1). The shaded area represents the 68% of confidence around the mean over five repetitions.

760

765

770

775 **Tables**

Aggregate name	Aggregate Type	Aggregate size (μm)	Seed size (i.e. the core) (μm)	Aggregate density (kg/m³)	Core density (kg/m³)	Aggregate porosity	Grainsize distribution
27Sk21	PC3	635 ± 50	270	310 ± 50	2500	0.86 ± 0.04	Yes
EJ15	PC1	≈ 200	≈ 40	60-500 (from literature)	2528	0.80-0.97	Yes

Table 1. Observed features for aggregates 27Sk21 (Sakurajima, Japan) and EJ15 (Eyjafjallajökull, Iceland). For the 27Sk21 sample, aggregate and core size are observed from high-speed video, the aggregate density is derived based on settling velocity and size from high-speed video, aggregate porosity is estimated based on measured particle density
780 **and derived aggregate density, core density is measured with a water pycnometer and grainsize distribution is derived from analysis of SEM images (Bagheri et al. 2016). For sample EJ15, the aggregate size, core size and grainsize distribution are derived from analysis of SEM images (Bonadonna et al. 2011), while the aggregate density and porosity are assumed based on literature (Brown et al., 2012; Gabellini et al., 2020), and the core density is estimated based on a combination of helium pycnometer and a theoretical relation with size (the reported value is the average for sizes less**
785 **than 40 μm; Bonadonna et al. 2011).**

	L axis (μm)	I axis (μm)	S axis (μm)	$\alpha = \frac{L}{I}$	$\beta = \frac{L}{S}$
Coarse ash ellipsoid	158.0	138.7	116.4	1.1	1.4
Fine ash ellipsoid	28.9	24.0	11.1	1.2	2.6

790 **Table 2. Geometrical features of the ellipsoids used to simulate fine and coarse ash. L, I, S are the average of 2010 Eyjafjallajökull (Iceland) samples for particles with size ≤63 μm (fine ash) and** between 63 μm and 2000 **(coarse ash).**