Replies to Reviewer 1

1) I think it would be easier to understand the procedure with very simple examples, for example with the initial volume given by an ellipsoid, a sphere or a cube. Looking at figure 3, for example, it is not clear how the conversion from STL to spheres works, also because it seems that there are isolated spheres (on the right in panel 3B). I was thinking that all the spheres should be connected/touching.

Reply: We agree with the reviewer that we were not clear in specifying how spheres are placed inside the STL shape. And indeed there can be isolated spheres, the ones that are in touch with the external STL surface. We propose to modify the text as follows, but we would prefer to leave a more complex shape than ellipsoids or cubes because we think it is more appropriate for the kind of use the code is designed for.
The modified paragraph is:
Line 157: "*The N_v vertices are points distributed along the surface of the STL shape (Fig.3a, 3d). Faces are instead described by a matrix of three columns and N_v/3 N_v/3 rows, where each row contains three integers reporting the corresponding vertices involved in the creation of the face.*
fromStlToSpheres generates a random point P_r inside the 3D surface using the Matlabthe MATLAB built-in function inpolyhedron. This operation is repeated until P_r is generated outside an existing sphere. Then it we finds the closest point P_ns among all the vertices of the triangles (see Fig.3a, 3d) andor the random points already placedthe centers of already placed spheres. If P_n∈N_v, the This information allows the center and the radius of the new placed sphere to beis computed(P_r P_n ⁀; on the contrary if P_n is one of the already placed spheres, the radius is (P_r P_n ⁀ minus the radius of the sphere whose center is P_n, once verified that the random point has not been generated inside an existing sphere. In conclusion, the new placed sphere will be tangent or to the STL surface or to another sphere.".

***

2) I have found that in some part of the paper a more quantitative analysis would be important. In particular, I think that a metric to quantify the accuracy of some steps of the procedure are needed. For example, when a given 3D shape is approximated with a set of not overlapping spheres, I think it is important to quantify how much this approximation is close to the original shape, both in terms of volume and surface. Is it possible to quantify the accuracy of the approximation?

Reply: This is an interesting observation. We agree that it would be an improvement to provide some quantitative information on the degree of accuracy between the initial STL representation and the sphere composite representation. Following your suggestion, we added a line of code where we compute the fraction of the actual volume of the STL object that is covered by the spherical representation. This is then given as an info in the new "datalog" structure that has been added as an output for "fromStlToSpheres". For what concerns the surface, it is not that easy to quantify how much of the actual surface of the STL object is described in terms of the spheres. Because this is a quite complex result of the sphere geometry packing and their orientation towards the STL surface. A compromise would be to compute the external surface that contains the spheres (such as the convex hull, see later

comments on that) and then compare this to the actual surface of the STL file. But we are not sure that this would help the user to rigorously quantify the approximation, since the internal surface is in turn a secondary approximation.

***

3) The external volume of an aggregate is approximated by the convex hull outlined by the most outer points of its internal spheres. This choice is not clear to me, because the aggregate can be very far from being convex, and this would lead to a significant over-estimation of its volume. The volume of the components of the aggregate is computed in a different way, and this can lead to strange results in Eqs. 1-3. For example, when the equations are applied to one single component (i.e. without aggregation), rho_aggis different from rho_p. In addition, as the authors write, in this way the external volume of the aggregated is "approximated" by the volume of the convex hull. When using an approximation, as in my comment #2, I think that an estimation of the accuracy of the approximation is needed, otherwise it is difficult to analyze the subsequent results (porosity of aggregates).

Reply: We totally agree with the reviewer that the topic of the porosity and external volume deserves more space throughout the text. For what concerns the volume of a single particle, it is true that in case of a single particle the actual volume can be slightly different from the one determined by its sphere composite representation. Especially in those cases where the sphere composite does not properly describe the original shape. However, since SCARLET is meant to be used to reproduce aggregates with more than one particle, this error was considered of second order importance with respect to the convex hull approximation, where the error can be of greater importance. In those cases where the porosity is negative, as sometimes it happens for single particles or PC2 objects when the coating is negligible, the porosity can be considered null. On the other hand, for what concerns the convex hull approximation, we also agree that a much better quantification should be provided in the text. Even if a complete constraint on how good the convex hull assumption is for an external volume is really dependent on the STL files under analysis, the number of spheres used, their size, and also the scientific application in which the "porosity" is defined (for a fixed shape), we propose to add further text and investigations on the topic as shown as follows. But without the sake of being complete. More than a validation, this is a discussion.

We added the following text at the end of section 2 (Model description):
"*The calculation of the aggregate porosity requires some additional clarification here because it can vary according to the definition of V_ext. In SCARLET-1.0, the determination the aggregate porosity (Eq. 1) is done under the assumption that the external volume V_ext is of the aggregate is well approximated by the convex hull outlined formed by the most outer points of its internal spheressphere-composite representation. This choice is a compromise between what has been observed in nature for PC3 aggregates (Bagheri et al., 2016; Gabellini et al., 2020) and the aim for a reduced complexity in the algorithm. All the The volume V_int^i of each component of the aggregate isare directly calculated directly from the surface of the scaled STL shape, using the divergence theorem (Suresh, 2021see the K. Suresh algorithm on MathWorks File Exchange). In SCARLET-1.0 all the inner components of the aggregate are characterized by*

Aa unique density ρ_p is assigned to the monomers, that can be modified by the user by means of the variable closet.core_density. The particle packing τ is evaluated according to Eq.1:
The aggregate porosity ϕ_agg and density ρ_agg can then be easily quantified as:

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.43).

Where V_int^i is the volume of the i-th particle and V_ext is the global volume of the aggregate.The algorithm takes advantage of the MATLAB built-in function convexHull, which is applied to the set of most external points among those describing the sphere-composition representation of the aggregate. The choice of using points belonging to the sphere composite representation, instead of the STL file, is determined by the need of increasing the number of points generally used to define the external surface of objects involved in the aggregate. For a single STL characterized by a large number of facets and points, the use of a sphere-composite representation can lead to larger approximations in the determination of the porosity (or density). However, the code has not been designed for single particles and the error gets relatively less important for aggregates. In fact, in these cases, what matters is how well the convex surface assumption describes the actual overall bulk volume of the object, more than the error on the single component. In any case, a proper use of fromStlToSpheres is always preferred to obtain the desired sphere-composite representation of the irregular shape."

In addition, we added a completely new section at line 295 with a new figure (Fig.9), in which we try to show what is a reasonable error on the porosity for some shapes where the porosity is a-priori known:
"*3.1.1 Porosity evaluation using the convex hull approximation: a comparison with analytical results*
As outlined in section 2, the porosity of an aggregate is always dependent on the surface that is used to define the external volume of the object. In SCARLET-1.0 the use of the convex-hull approximation is suitable for central collisional processes that result in roughly spherical aggregates, such as the PC1 and PC3 samples recently observed in the field (Bagheri et al., 2016; Gabellini et al., 2020). However, it can lead to an overestimation of the porosity for structures that are poorly approximated by a convex geometry, such as fractal-like aggregates. In all these cases the porosity should be considered as an upper-bound limit.
In order to investigate the accuracy, we compared the porosity computed by the algorithm with those belonging to particular aggregate configurations or single objects for which analytical results are given in literature. In Fig.9a and Fig.9b the comparison is made with respect to a classical configuration of sphere packing (e.g. the cannonball problem (Lucas, 1883)). For these convex shapes the porosity difference is about 7%-12%, with a dependency on the number of inner spheres used to describe the shapes. In general, the higher the number of spheres, the higher the evaluated porosity. This is a consequence of a more accurate representation of the original structure. However, for a fixed number of spheres, the approximation is also dependent on how many points define the STL triangulation and from Niter (i.e. how large are the spheres). In Fig.9c and Fig9d we calculate the porosity of the fractal shape known as "Menger's sponge", respectively obtained with n=2 and n=3 recursive iterations, for which the porosity can be

exactly determined as a function of the recursive step n (Sergeyev, 2009). For n=2 and n=3 we found respectively a porosity larger than 14% and 4% .
Lastly, Fig.9e shows a non-convex L-shaped aggregate made of 8 spheres. If the user-defined external volume is the one related to the void filling the space between two close spheres, this is equal to Fig.9a (i.e. 48%). In this case SCARLET considers the convex surface that contains the spheres, which is close to the one defined by the triangular surface as base and one sphere diameter as height."

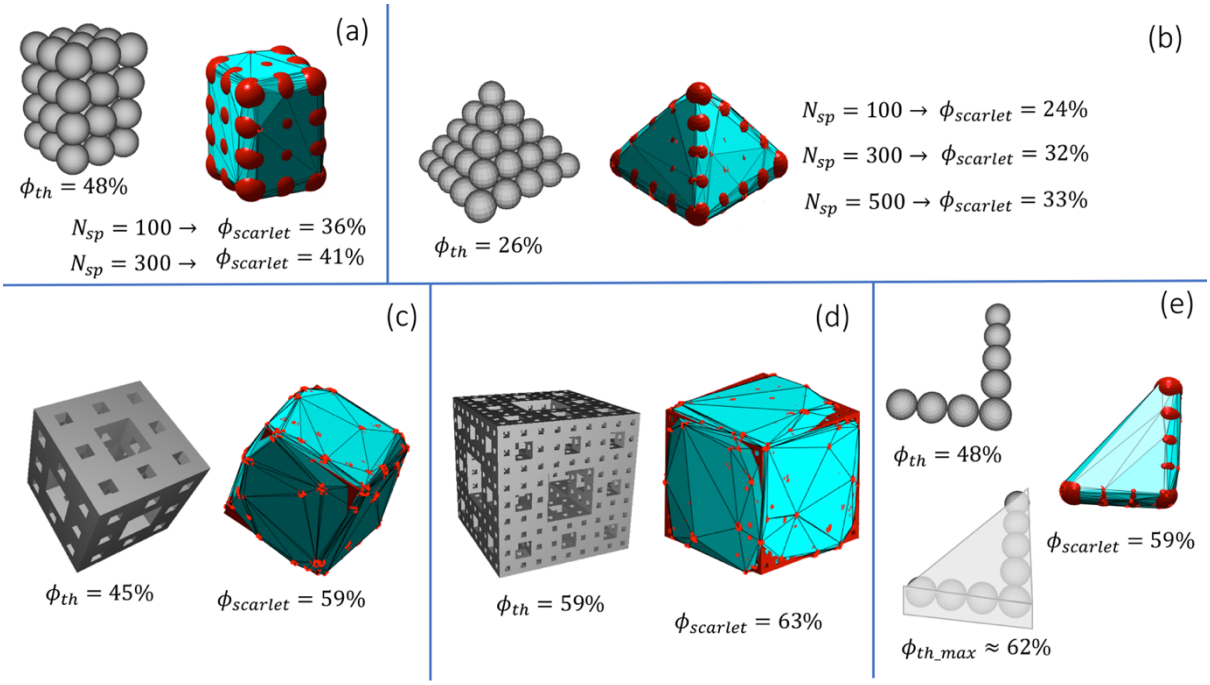

Figure 1 Evaluation of the accuracy in the determination of the porosity using the convex-hull surface of the most external points of the sphere-composite representation of structures for which the porosity is given. (a) Spheres packed with a theoretical 48% of porosity; (b) Spheres packed with a 26% of theoretical porosity; (c) – (d) Examples of Menger's sponge obtained respectively with 2 and 3 recursive processes; (e) L-shaped deposition of spheres for which the porosity is the same as in (a). The maximum theoretical porosity is evaluated considering the solid whose base is the L-shaped rectangular triangle and the height one sphere diameter.

We also added a caveat on this:
"The determination of the aggregate porosity is based on the assumption that the external surface that circumscribes the inner components is convex. This choice is motivated from the aggregate morphology associated to PC type aggregates in volcanology. Also in other applications this can be the case but the user must be a-priori aware of this. In fact, this approach may lead to an overestimation of the porosity in case, for example, of fractal-like aggregates for which the overall fractal dimension is less from three. In those cases the convex hull approximation can be seen as an upper limit for the maximum porosity, which converges to the actual porosity for aggregates that are well described by a convex geometry."

***

4) Section 3.2.1 is devoted to the analysis of the porosity of the union of two ellipsoid.It is not clear to me what you mean here with porosity, because there are no internal voids in this

configuration. So, I think that it is important here to give a clear definition of porosity. If porosity is simply defined by Eq. 1, does this definition coincide with that used in volcanology when measuring porosity of volcanic samples? I think this is an important point, because otherwise the analysis of results, and a comparison natural samples, are difficult to understand.

Reply: We agree with your comment because when dealing with porosity, it is always better to state clearly to which external volume we refer to. In fact, porosity can change a lot in this application, from zero (if we consider the external volume as the one defined by the two object assuming zero vesiculation in the ellipsoids) to the maximum one, defined by the convex surface that contains the two shapes regardless of their orientation. Here we are interested in the second one. The application came to our mind as a curious mathematical investigation, more than a real application in volcanology. However, we think that especially for the study of multidimensional population balances in the theoretical description of ash aggregation, this could be of some interest in the future.
In order to fully address your comment we propose to add the following sentence to sec.3.2.1 and to improve Fig.11 (the old Fig.10) adding two sketches about what we mean with external volume in this analysis.
"*In this application we are interested in studying how the porosity ϕ_agg changes as a function of particle size ratios and their orientation in space. Here, such as in rest of the paper, the external volume for the calculation of ϕ_agg is defined by the convex-hull surface that bounds the two ellipsoids involved in the collision (Fig. 11b, 11c). This is equivalent to study the maximum porosity that can exist between two single and not-vesiculated ellipsoids.*"

5) As a final point, I think that the computational time required to run the package should be discussed a little bit more. A table with the times of the simulations reported in some of the examples could be useful.

Reply: We agree with the reviewer that the computational time requires some more attention. We propose here to add some caveats at the end of the paper on this aspect, more than a rigorous quantification in the main body. This because we couln't perform massive simulations on a real multi-core cluster during the revision process and because at this stage of the project our attention is mostly dedicated to the virtual reconstruction of ash aggregates and the validation/verification with observations. The improvement of the computational efficiency will be probably the main goal of the future release(SCARLET-v2.0), in which a special attention will be dedicated in speeding up both the sphere-composite representation and the while loop that is in charge of the outward movement of the particle (this is the main bottleneck at the moment). Moreover, we will introduce the option to first first rotate the shapes and then use them in the investigation cones (this allows for two parfor in sequence). This reduces the indepence of each collision but it speeds up the code.

We propose to add in the caveats the following paragraph:
"*4.3.2 Computational efficiency*
The computational efficiency of the package in the release v1 is mostly dependent on four main factors: i) The step-size used to detect the single collision; ii) The number of particles in the coating; iii) the number of spheres used in the sphere-composite representation; iv) the number

of cores available for the parallelization of the rotations. The tests performed using a laptop with processor i7-4600U CPU @ 2.10 GHz x4 (2 threads in MATLAB parfor) revealed that among all the above mentioned factors the most critical parameter is f_c2, that controls the outward movement of the i-th particle. It shows a non-linear increase of the computation time in reducing the iteration step. On the other hand, the computational time increases linearly with respect to the number of spheres used in the sphere-composite representation of the STL, the number of rays in the investigation cone and the number of rotations."

Here we attach a list of tables where we report the computation times obtained with different setups with a HP laptop *i7-4600U CPU @ 2.10 GHz x4*, that corresponds to 2 workers in MATLAB parfor (basically no parallelization). This is not a rigorous investigation of the computational time (as previously mentioned), but just some simulations to capture the bulk behavior of the code in varying some of the parameters. The trends reported in the caveats are derived from this table.

For all the simulations we used the same ellipsoid.

Table 1: example of computational times measured during ash aggregate simulations

Number of particles	Rays	Rotations	Inward step-size movement (coarse)	Outwards step-size movement (fine tuning)	Time (s)
50	10	1	0.05	0.001	198
50	10	10	0.05	0.001	345
50	10	30	0.05	0.001	637
50	10	60	0.05	0.001	998
50	50	50	0.05	0.001	1126
50	100	1	0.05	0.001	441
50	100	10	0.05	0.001	2081
50	100	30	0.05	0.001	5617
50	100	60	0.05	0.001	10422
100	10	1	0.05	0.001	550
100	10	10	0.05	0.001	1058
100	10	30	0.05	0.001	2161
100	10	60	0.05	0.001	3690

Table2: Test varying the number of spheres

Number of particles	Number of spheres	Rays	Rotations	Inward step-size movement (coarse)	Outwards step-size movement (fine tuning)	Time (s)
100	10	10	10	0.05	0.005	204
100	100	10	10	0.05	0.005	1728
100	200	10	10	0.05	0.005	2049
100	300	10	10	0.05	0.005	6140

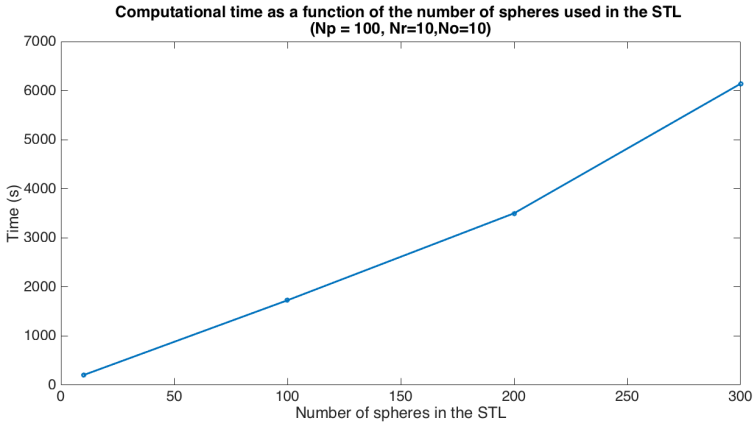

Table3: test on the effect of the step size on a single particle in the coating (200 spheres in the sphere-composite)

Number of particles	Rays	Rotations	Inward step-size movement (coarse)	Outwards step-size movement (fine tuning)	Time (s)
1	1	1	0.05	0.05	0.10
1	1	1	0.05	0.005	0.13
1	1	1	0.05	0.0005	0.24
1	1	1	0.05	0.00005	0.42
1	1	1	0.05	0.000005	1.36

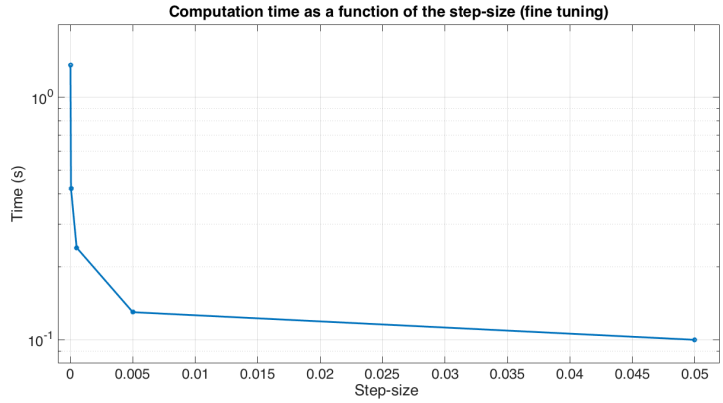

Table4: test on the effect of fine tuning on a single particle in the coating (200 spheres in the sphere-composite)

Number of particles	Rays	Rotations	Inward step-size movement (coarse)	Outwards step-size movement (fine tuning)	Time (s)
1	30	1	0.05	0.05	4.2
1	30	10	0.05	0.005	24.3
1	30	30	0.05	0.0005	79.6
1	30	60	0.05	0.00005	141.8

Table5: Test on the effect of fine tuning on a single particle in the coating (100 spheres in the sphere-composite)

Number of particles	Rays	Rotations	Inward step-size movement (coarse)	Outwards step-size movement (fine tuning)	Time (s)
1	30	1	0.05	0.05	4.2
1	30	10	0.05	0.005	24.3
1	30	30	0.05	0.0005	79.6
1	30	60	0.05	0.00005	141.8
10	30	1	0.05	0.05	34
10	30	10	0.05	0.005	230
10	30	30	0.05	0.0005	647
10	30	60	0.05	0.00005	1250

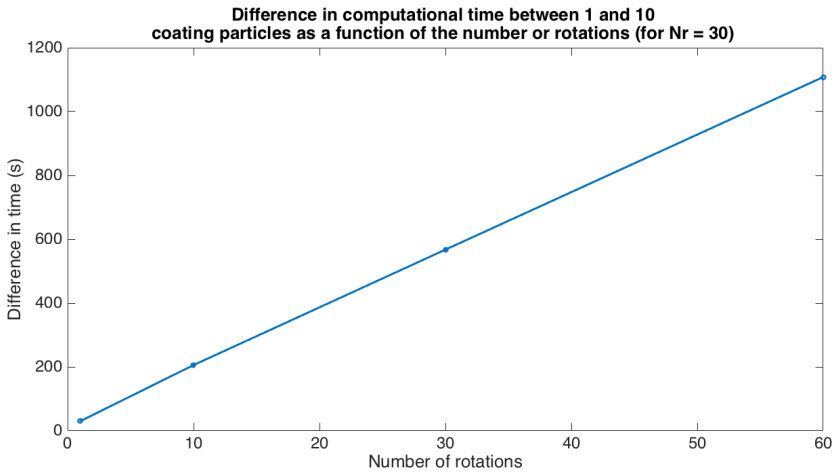

Table6: Test on the number of rays (100 spheres in the sphere-composite)

Number of particles	Rays	Rotations	Inward step-size movement (coarse)	Outwards step-size movement (fine tuning)	Time (s)
10	1	1	0.05	0.005	1.1
10	10	1	0.05	0.005	10.1
10	100	1	0.05	0.005	104.1
10	1000	1	0.05	0.005	1139.2

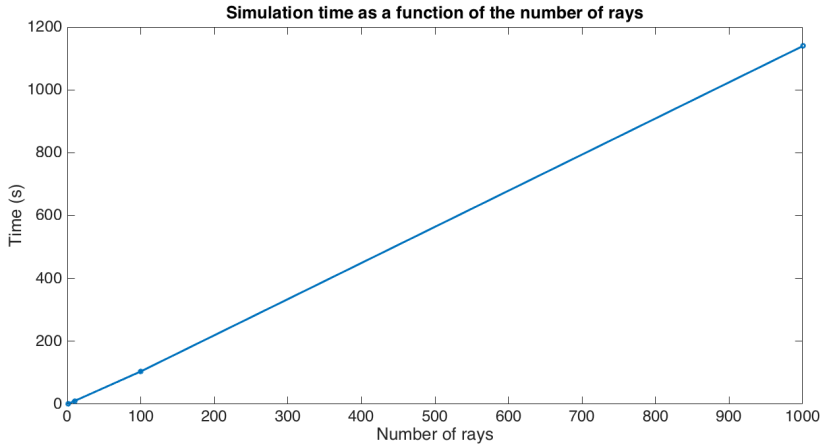

Table7: test on the number of the rotations

Number of particles	Rays	Rotations	Inward step-size movement (coarse)	Outwards step-size movement (fine tuning)	Time (s)
10	1	1	0.05	0.005	1.17
10	1	10	0.05	0.005	5.74
10	1	100	0.05	0.005	54.5
10	1	1000	0.05	0.005	569.3