Appendix 4 SAMURAI_1.0 User Manual

Robert L. Eastwood[1] and Thomas R. Anderson[2]

1 1942 Rosebank Ave, Halifax, Nova Scotia B3H 4C7, Canada

2 Marine Systems Modelling, National Oceanography Centre, Southampton SO14 3ZH, United Kingdom

Citation

Eastwood, R.L. and Anderson, T.R. (2022). SAMURAI_1.0: Surface Area Modelling Using Rubik As Inspiration, version 1.0. GitHub *** [provide URL once published]

Introduction

SAMURAI (Surface Area Modelling Using Rubik As Inspiration) version 1.0 is a model that simulates the progressive degradation of individual detritus particles that are represented as a matrix of aligned sub-units akin to the well-known Rubik cube. The seminal publication is:

Anderson, T.R., Gentleman, W.C., Cael, B.B., Hirschi, J.M., Eastwood, R.L. and Mayor, D.J. (2022). Modelling surface area as a dynamic regulator of particle remineralisation in the ocean. Biogeochemistry *** [vol, pp., doi.]

1. Rubik structure and degradation

The matrix is specified in x-y-z dimensions and can be entirely filled with cube-shaped sub-units (each with six faces) from the outset, generating a solid cuboid, or can alternatively be populated to give other chosen shapes. This release of SAMURAI provides options for either a cuboid (a rectangular hexahedron made up of three pairs of rectangles whose dimensions x, y and z that define the width, depth and height of the object) or an ellipsoid (a deformed sphere obtained by scaling its dimensions in x, y and z that describe the diameter of each of the respective axes). Note that the published article (Biogeochemistry) considers spheres that are defined by a grid resolution (GR), e.g. GR 100× is a sphere with diameter 100 sub-units.

As well as opting for a solid matrix (entirely filled with sub-units), the user may choose to simulate aggregate particles where the main shape volume is populated with spherical sub-particles made up of sub-units on the same grid. For example, a 100× spherical aggregate would have an outer bound as in a solid 100× particle, but could instead be populated by sub-particles of 31× diameter (as shown in Figure 1, which illustrates the User Interface). Such a particle is constructed as follows. A notional empty spherical container is first created with dimensions 100×100×100. The aggregate is first seeded with a single sub-particle at the geometric centre of the container. Further sub- particles are then added by (i) randomly selecting a point anywhere within the volume of the container, (ii) randomly selecting an existing sub-particle within the aggregate (iii) creating a vector from that point to the centre of the selected sub-particle, (iv) bring in the new candidate sub- particle along the vector until

it touches the main particle ensuring that, at the last stage of the process, the sub-particle aligns with the grid, (v) fuse the new sub- particle providing two conditions are met, that not exceed the container boundary and that it does not overlap any existing sub-particles at the contact point (it is discarded in either instance). This procedure is repeated until, as far as possible, the container is filled.

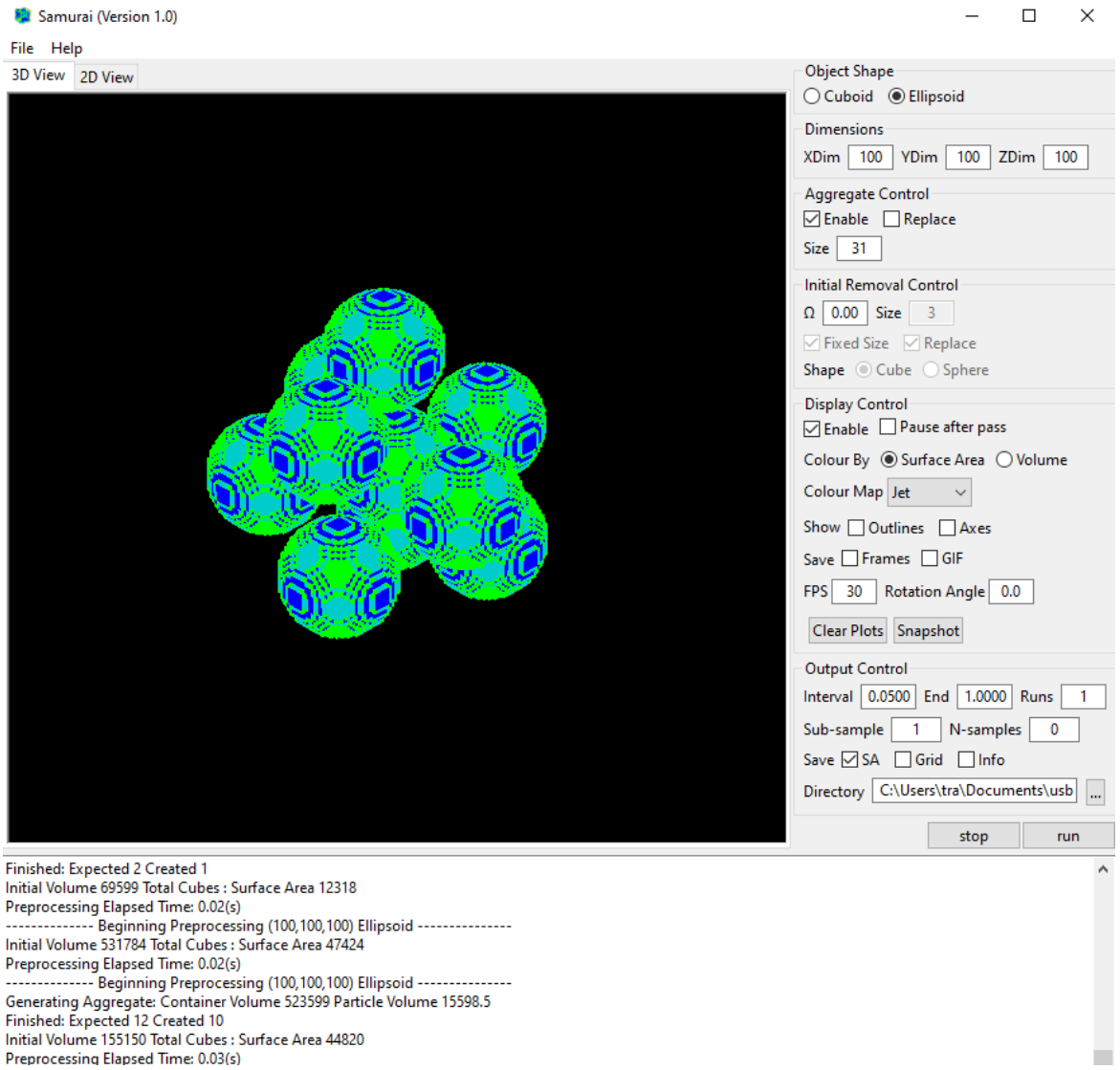


Figure 1. User interface, here shown for a spherical aggregate with initial dimensions 100×100×100, packed with sub-units or dimensions 31×31×31. Sub-units are coloured by surface area (number of exposed faces).

An alternative means of generating heterogeneity (pores and cavities) into the matrix at initialisation is provided, but which applies to non-aggregate particles only. A fraction of total volume to be removed is specified, Ω, e.g., $\Omega = 0.3$ requires an elimination of 30% of the otherwise solid Rubik. This is carried out by removing the corresponding number of sub-units from the chosen Rubik shape, one at a time, using random selection. In order to ensure that the particle remains as a single entity, a check is made after the desired number of sub-units have been removed to see if the matrix has fragmented, i.e., separated into two or more fragments. If this is so, then the user has the option to eliminate all except the largest

fragment and add the removed sub-units back on to the largest fragment at random locations, ensuring that these are within the bounds of the shape initialisation. In this way, the initialisation generates a single particle, without the possibility of multiple fragments.

Particle degradation is represented by sequential removal of sub-units within the matrix based on exposed surface area (i.e., outward facing, thereby excluding internal cavities that may be generated when $\Omega > 0$). At each step in the simulation, the computer randomly selects an exposed face and the associated that sub-unit is removed from the matrix, exposing new faces associated with adjacent sub-units. Sub-unit removal continues until the matrix is left empty, at which point degradation is complete.

2. User options

The various user options associated with the User Interface (Figure 1) are described in this section. Note that when making alterations to text boxes, users should type a carriage return to ensure that the new settings are enacted.

Options 1: View

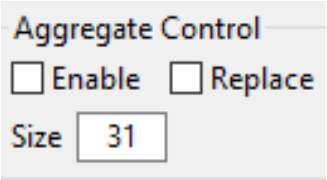
The chosen interface display can be either 3D or 2D: buttons at top left, **3D View**, **2D View**.

Options 2: Object Shape and Dimensions

Either **Cuboid** or **Ellipsoid** may be selected, with dimensions in x, y and z, **XDIM**, **YDIM**, **ZDIM**, e.g., 100, 100, 100 as in Figure 1.

The maximum dimensions that can be handled by the model, as coded, are constrained by the product of XDIM, YDIM, ZDIM. The limit is 2^{33}-1. While this is the programmed maximum, the functional maximum is dictated by available system memory. Each sub-unit requires a minimum of 8 bytes of memory, e.g. a 1000×1000×1000 cuboid object would require at least 8 Gb of memory.

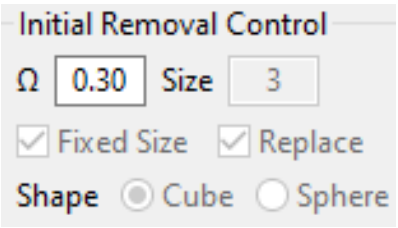
Options 3: Aggregate control



Enable create aggregates of specified Size.

Replace It is the case that when aggregates are formed, a very small number (typically <10 out of millions) of sub-units may overlap (merge) during the process of sub-particle fusion. This option permits the user to add these sub-units back to the aggregate particle, following the same rules for porosity generation in Options 4 (below). Note that the number of overlapping sub-units can be minimised by making sub-particle diameter an odd-number of sub-units.

Options 4: Initial Removal Control



Ω is the fraction of sub-units within the matrix to be removed at initialisation.

Size is the diameter, with each sub-unit counting 1, of the pore or cavity (sub-unit volume) to be removed, and can be either cube- (default) or sphere-shaped.

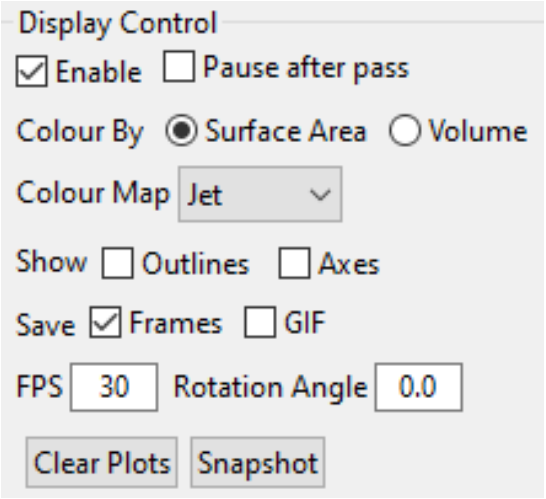
Fixed Size. This checkbox indicates whether the pore size should be fixed (checkbox ticked) or vary randomly between 1 and the size indicated by the **Size** input.

Replaced removed. During removal at initialisation when Ω > 0, fragmentation may occur.

This checkbox indicates that sub-units associated with all but the largest fragment should be randomly added back to the object so that initial volume conditions are maintained. Default is checked.

Shape Allows pores to be cuboid or spheroid in shape

Options 5: Display Control

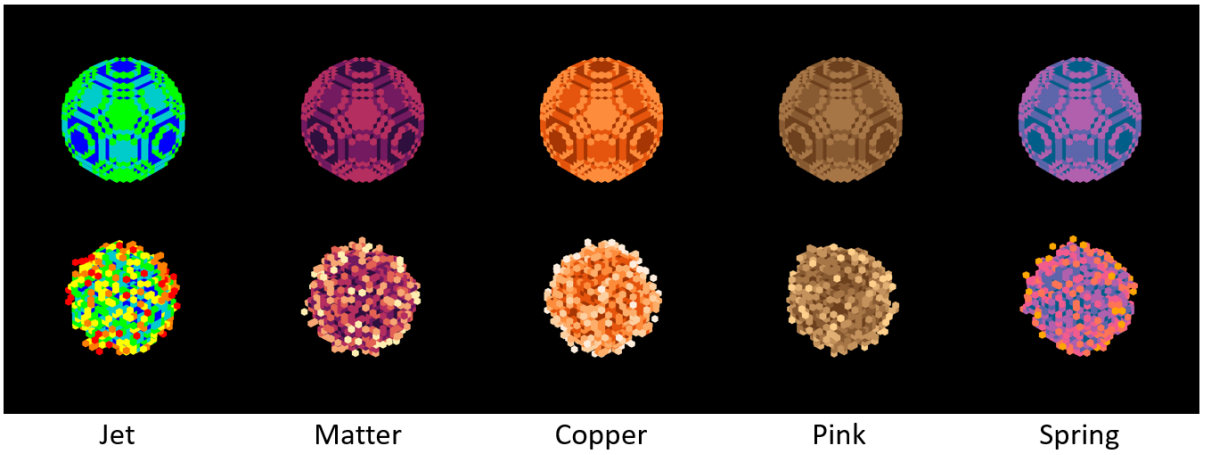


Enable. This checkbox enables the 3D and 2D views. Disabling this feature can greatly increase processing speed.

Pause after pass. This checkbox causes the simulation to be paused after a specified output interval (see Output Control options, below). Pressing the **Resume** button resumes processing until the next interval is reached. This option is useful for visually monitoring a simulation in steps as it proceeds, and for taking snapshots. Especially for small grid sizes, the simulation may otherwise pass by too quickly to permit effective visual inspection.

Colour by. The Rubik can be coloured by either its surface area or volume, including any fragments that may have formed during the simulation. **Surface area** colouring represents the number of exposed faces associated with each sub-unit. **Volume** colouration represents $\log_{10}$ the number of sub-units of the Rubik, including fragments.

Colour Map. A drop-down menu allows the user to select between five colour schemes, Jet, Matter, Copper, Pink, and Spring:



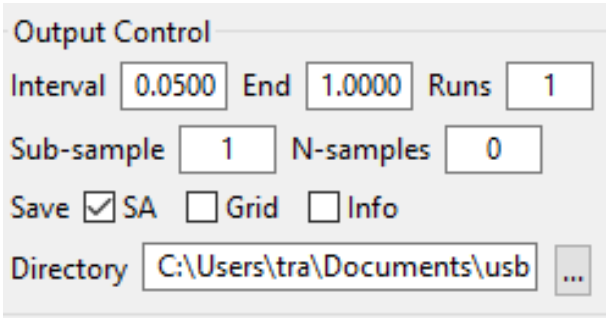
Show has two associated checkboxes. **Outlines** enables each face of the sub-units to be outlined with a white frame. **Axes** enables the x, y, and z axis to be displayed with the Rubik at the centre.

Save has four associated checkboxes. **Frames** enables saving the display at each output interval as PNG files. **GIF** enables the saving of the display at each output interval as a GIF animation. **FPS** is the frames per second that the GIF animation will run. **Rotation Angle** is the amount that the object will rotate about the z-axis at each output interval.

Clear Plots. This button clears the 2D View display.

Snapshot. This button saves the current 2D and 3D view displays.

Options 5: Output Control



Interval. This is the update interval which is the fraction of model simulation to be carried out before each update. If Display Control is enabled (Options 3), the visual display will be updated as per the specified Interval. An Interval of 1.0 means that all the processing is completed in one go. Updates allow both display and status feedback to inform the user of what fraction of the model simulation has been completed.

End. This enables the model simulation to be terminated before 100% of sub-unit removal, if desired. For example, a value of 0.25 will result in removal of 25% of the sub-units. Default is 1.0 (process all sub-units).

Runs. This setting allows users to run ensembles of simulations, each with the same configuration. Default is 1 (a single model run). Output files for each simulation are saved to sub-directories, named Run###, under the main Directory.

Sub-sample. The frequency at which model results are written to output files. For example, a setting of 100 means that every 100th position in the sequence of sub-unit removal is written as output, noting that initial and final positions are always included. This setting provides a means to reducing the size of output files which may otherwise be very large when using high grid resolution. Default is 1 (each step in the sequence is saved).

N-samples. The total number of samples that will be written to the output file. This provides a means to reduce file output size. Default 0 (All samples saved); "1". **Sub-sample** is automatically adjusted to match the desired number **N-samples**. To re-set the default, type "1" into **Sub-sample**.

Save. The **SA** checkbox enables the saving of a comma-separated output text file (.txt) that has two columns: (1) the number of sub-units removed (starting at 0 and completing with the initial number of sub- units in the matrix); (2) and number of exposed faces (starting at the initial condition and ending at 0). The **Grid** checkbox enables the saving of all sub-unit

information necessary to reconstruct the 3D display, generating an output text file (.txt) that has four columns: the first three represent the x, y, z position of each sub-unit in the matrix, while the fourth column provides the number of exposed faces associated with that sub-unit. The **Info** checkbox stores basic information about the run including initial volume and surface area, number of sub-particles (in the case of aggregates) and porosity.

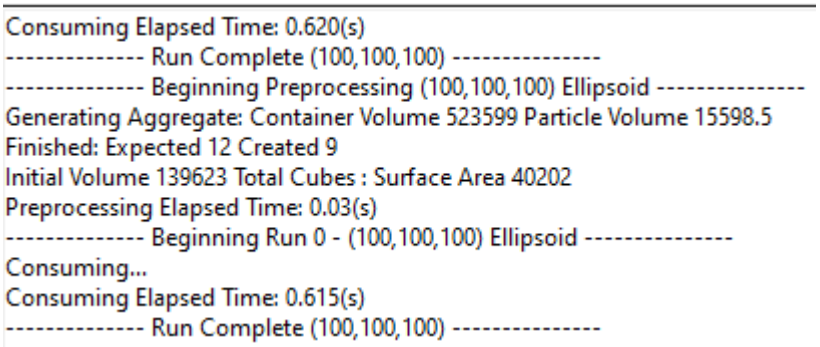
Directory. Permits the user to specify a full path to a chosen directory for output. The directory can alternately be selected by browsing the computer desktop using the **…** button.

Buttons

stop. This button halts the model simulation, which cannot then be resumed.

run/**pause**/**resume**. This button starts the model simulation (**run**), pauses processing (**pause**) and resumes processing (**resume**). At each update interval, if **Pause after pass** is enabled, the button will change from **pause** to **resume**. The user can then press **resume** to continue processing until the next interval is reached.

Message Box



The message box provides configuration information and current processing feedback.

4. XML file

Running the SAMURAI_1.0 model generates an XML file, “SAMURAI.xml” which stores the input settings. Rather than using the User Interface, the user can instead make alterations by directly editing this file. In case of the settings shown in Figure 1, the associated XML file shows the following when opened:

```
<?xml version="1.0" encoding="utf-8" ?>
<settings>
      <app w="880" h="820" x="0" y="0" sashpos="623" />
      <object isCuboid="0" />
```

```
     <dimensions x="100" y="100" z="100" />
     <porectrl porosity="0.00000" size="3" fixed="1" isCuboid="1"
replace="1" />
     <aggctrl enable="1" size="31" replace="0"/>
     <outputctrl inc="0.05000" end="1.00000" runs="1" subsamp="1"
nsamps="0"
dir="C:\Users\tra\Documents\usb180321star\aaW_projects190307\CNpro
filemodel210922\model_Bob211016\SAMURAIagg230223\test230223"
save="1" grid="0" info="1" />
     <displayopts enable="1" cmindex="0" faces="1" outline="0"
axes="0" pause="0" save="0" frames="0" cubeview="1" />
     <displayctrl fps="30" rotangle="0.00000" zoom="0.00000"
xangle="0.00000" yangle="0.00000" zangle="0.00000" />
</settings>
```