GENERATING PARTICLES DIAMETER ACCORDING WITH A SIZE DISTRIBUTION FUNCTION

SILVA, A.C.[1], GALERY, R.[2], PERES, A.E.C.[3]
[1]Mining Engineering Department and Geography Post Graduation Program, Geography Department, Goiás Federal University, Catalão, Brazil. andrecarlos@catalao.ufg.br
[2]Mining Engineering Department, Minas Gerais Federal University, Belo Horizonte, Brazil. rgalery@demin.ufmg.br
[3]Metallurgical Engineering Department, Minas Gerais Federal University, Belo Horizonte, Brazil. aecperes@demet.ufmg.br

ABSTRACT

The size of the elements in a particles population, as many others variables, tends to follow a well-defined mathematical law to describes its size distribution, being this a subject not only theoretical, but applied as well, once the data handling is easier if it could be described mathematically. Many size distribution functions has been proposed to satisfactory model the granulometric distribution of a material, such as Log-Normal, Gates-Gaudin-Schumann, Rosin-Rammler-Bennett-Sperling, Gaudin-Meloy, Roller and Svensson. To simulate granular material is often necessary to establish many parameters of the simulated particles, like the diameter, sphericity, roughness, restitution coefficient, amount many others. This paper shows an algorithm to generate the diameter of each particle of a finite particle population characterized by a size distribution function. The results found indicated the algorithm is able to generate the size of a particle population in a fast and precise way, providing a reliable input to DEM simulations.

KEYWORDS: simulation; computational; granulometry; distribution.

1. INTRODUCTION

According to King (2001) the distribution function for a particular property defines quantitatively how the values of that property are distributed among the particles in the entire population. Perhaps the best known and most widely used distribution function is the particle size distribution function $P(d_p)$ defined by $P(d_p)$ = mass fraction of that portion of the population that consists of particles with size less than or equal to d_p. The symbol d_p is used throughout this book to represent the size of a particle. The function $P(d_p)$ has several important general properties:

(a) $P(0) = 0$;
(b) $P(\infty) = 1$;
(c) $P(d_p)$ increases monotonically from 0 to 1 as d_p increases from 0 to ∞.

Properties (a) and (b) are obvious because no particle in the population can have a size less than or equal to 0 and all the particles have a size less than infinity. Property (c) reflects the fact that the fraction of the population having size less than or equal to d_{p1} must contain at least all those particles of size d_{p2} or smaller, if $d_{p2} < d_{p1}$.

King (2001) affirms that the concept of particle size is ambiguous, since particles that are of interest in mineral processing do not have regular definable shapes such as spheres and cubes. The size of a spherical particle can be unambiguously defined as the diameter. Likewise the size of a cube can be defined unambiguously as the length of a side but another dimension could be equally well used such as the longest diagonal. Particle size clearly does not have a unique meaning even for particles with regular shapes. In mineral processing technology an indirect measure of size is used. In mineral processing the size of a particle is defined as the smallest hole opening in a square-mesh screen through which the particle will fall. Sometimes it is necessary to work with particles that are too small to measure size conveniently by means of screening. Then other appropriate indirect measures are used such as the terminal falling velocity in a fluid of specified viscosity and density. In practical applications it is convenient and often essential to make use of a discrete partitioning of the length scale so that the particle population is divided conceptually into groups each identified by the smallest and largest size in the group and the value of P can be measured experimentally at a number of fixed sizes that correspond to the mesh sizes of the set of sieves that are available in the laboratory.

According to Macías-García *et al.* (2004) many methods of varying complexity have been developed to determine the size distribution of particulates. Particle size is probably the most important single physical characteristic of solids. It influences the combustion efficiency of pulverized coal, the setting time of cements, the flow characteristics of granular materials, the compacting and sintering behaviour of metallurgical powders, and the masking power of paint pigment. These examples illustrate the intimate involvement of particle size in energy generation, industrial processes, resource utilization, and many other phenomena.

Wills and Napier-Munn (2006) indicate that many curves of cumulative oversize or undersize against particle size are S-shaped, leading to congested plots at the extremities of the graph. More than a dozen methods of plotting in order to proportion the ordinate are known. The two most common methods, which are often applied to comminution studies, where non-uniform size distributions are obtained, are the Gates-Gaudin-Schuhmann (Schuhmann, 1940) and the Rosin-Rammler (Rosin and Rammler, 1933) methods. Both methods are derived from attempts to represent particle size distribution curves by means of equations, which results in scales which, relative to a linear scale, are expanded in some regions and contracted in others.

According to Wills and Napier-Munn (2006) the Rosin-Rammler method is often used for representing the results of sieve analyses performed on material which has been ground in ball mills. Such products have been found to obey the following relationship:

$$Y = 100e^{-\left(\frac{x}{b}\right)^n} \qquad (1)$$

Where Y is the cumulative undersize in per cent, b is a constant, x is the particle size, and n is a constant.

The same authors pointed that the Gates-Gaudin-Schuhmann plot is often preferred to the Rosin-Rammler method in mineral processing applications, the latter being more often used in coal-preparation studies, for which it was originally developed, and is given by the relationship:

$$Y = 100.\left(\frac{x}{b}\right)^n \qquad (2)$$

In this paper an algorithm to generate the diameter of each particle of a finite particle population characterized by a size distribution function is presented. It is easily possible to change the parameters and the size distribution function to simulate a particle population similar to a real one. The results found indicated the algorithm is able to generate the size of a particle population in a fast and precise way, providing a reliable input to DEM simulations.

2. MATERIAL AND METHOD

To simulate the diameter of a particle population is necessary first to determine the value of the constants *b* and *n* used to describe the material, which can be done by following the steps:

1. Perform a granulometric analysis, such as a screen analysis, of the real particle population which is desired to simulate;
2. Plot the screen analysis graph, particle size versus the cumulative weight percentage undersize;
3. From the graph is possible to calculate the *b* and *n*;
4. The calculation of the diameter of the particles following the Gates-Gaudin-Schuhmann (or GGS) equation, for example, is possible by:
5. Create a two-dimensional matrix (*p*) to store the sieve size and probability of this sieve to contains particles, which size must agree with the number of number of sieves considered;
6. Define the number of sieves (*NumberOfSieves*), the size of the first sieve (*Si*) and the last one (*Sf*);
7. The code hereafter shows how to calculate the probability of each sieve to contain particles retained on it:

```
dx := sqrt(2);
for i := 0 to NumberOfSieves do
begin
  p[i].x := Si * power(dx,i);
  p[i].Y := Min(Power(p[i].x/b, n),1);
end;
```

8. To calculate the diameter of each particle is necessary to create a function (here named as *SearchProbability*) to search the matrix *p* trying to find the index of the sieve which has the nearest probability around a given one. This function is given by the code bellow:

```
function SearchProbability (p1: Extended): integer;
var
  i: integer;
```

```
begin
  Result := 0;
  for i := 0 to Length(p)-1 do
  begin
    if p1 < p[i].Y then
    begin
      Result := i;
      break;
    end;
  end;
end;
```

9. Create an array (*a*) to store the particles diameters, which size must agree with the number of particles (*NumberOfParticles*) to be created;
10. The size of each particle is calculated using a uniform distribution by code bellow:

```
for i := 0 to NumberOfParticles do
begin
  repeat x := random*dx until (x >= 1) and (x <= dx);
  a[i] := x * p[SearchProbability (random)].X;
end;
```

It´s possible to notice that is necessary to randomly select two different uniform distributed numbers. The first number is used to calculate how bigger the particle is than the sieve, since the function *random* generates uniform distributed pseudorandom numbers between 0 and 1 ($0 \leq$ *random* < 1). The second number is used to generate a probability within the probability matrix and, doing so, generated elements which still follows the desired distribution.

3. RESULTS

The proposed algorithm was tested against data obtained from quartz sand screen analysis, as shown in table 1, which also presents the results the results obtained from the use of the GGS equation and the proposed algorithm. The figure 1, plotted using the data in table I, shows the log of the particle size versus the log of the cumulative weight percentage undersize, used to determine the constants in the GGS equation. It´s possible to note that the quartz sands adjust very well to the GGS model (R^2 ~ 98%) and with the linear regression was possible to determine the constants *b* and *n* for the quartz sand.

Table I. Results from quartz sand screen analysis.

Sieve		Sieve fractions		Cumulative %		GGS	Algorithm
#	µm	wt (g)	wt %	oversize	undersize		
35	419	30	11.11	11.11	88.89	98.02	99.99
48	296	40	14.81	25.93	74.07	73.49	75.64
65	209	40	14.81	40.74	59.26	55.10	56.91
100	148	50	18.52	59.26	40.74	41.31	43.08
150	105	25	9.26	68.52	31.48	30.97	33.17
200	74	15	5.56	74.07	25.93	23.22	25.47
270	52	20	7.41	81.48	18.52	17.41	19.50
400	37	20	7.41	88.89	11.11	13.05	15.16
- 37		30	11.11	100.00	0.00	0.00	0.00
TOTAL		270	100.00	-	-	-	-

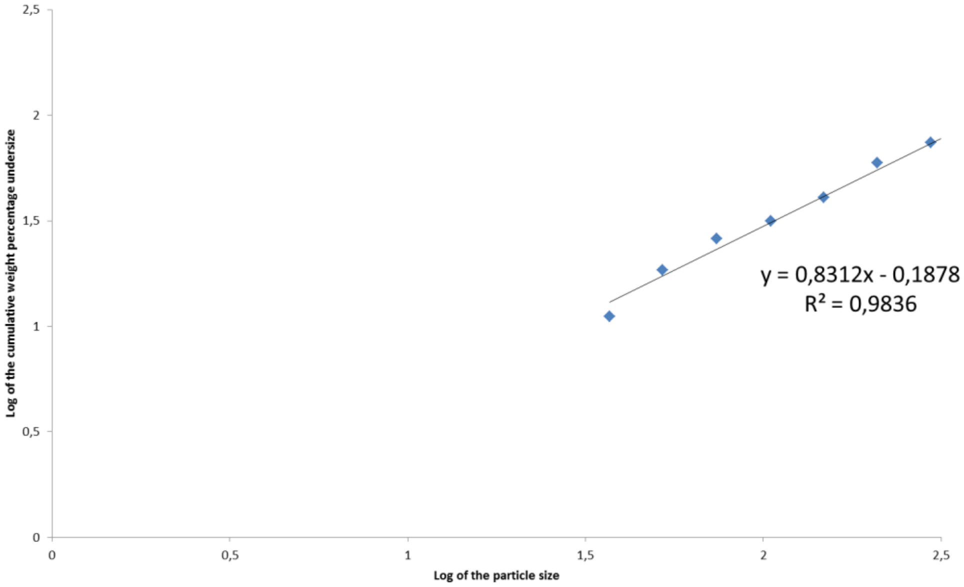

Figure 1. Log of the particle size versus the log of the cumulative weight percentage undersize, used to determine the constants in the GGS equation.

The figure 2 shows the results of the granulometric analysis using the data from table I for the quartz sand sample, as well as the granulometric analysis expected form the GGS model and the proposed algorithm. Both GGS model and the algorithm led to overestimation of the first sieve (419 μm) and the last one (37 μm). Figure 3 shows the correlation coefficient between the quartz sand sample and the GGS equation, figure 4 the correlation coefficient between the quartz sand sample and the proposed algorithm and figure 5 the correlation coefficient between the proposed algorithm and the GGS equation.

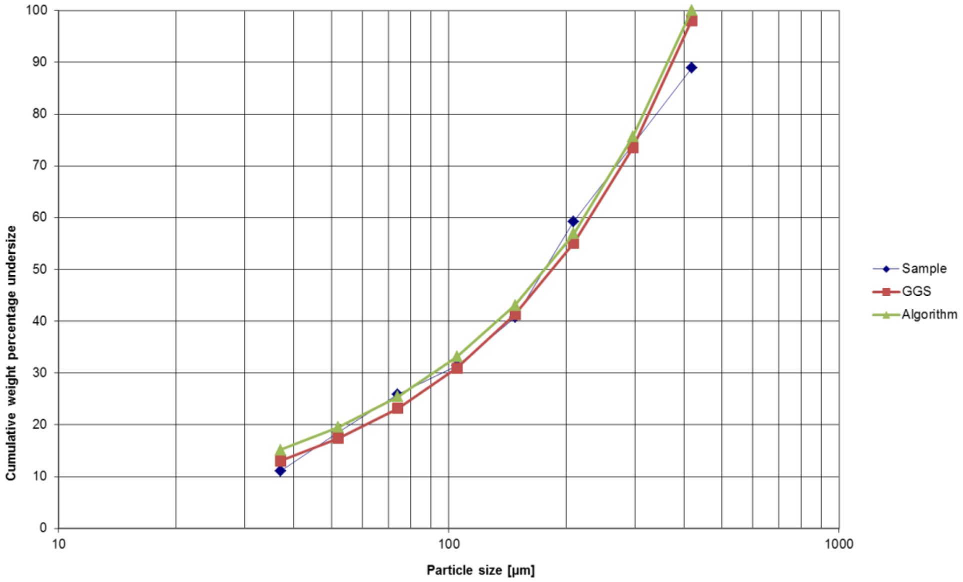

Figure 2. Expected granulometric analysis from the quartz sample, GGS model and the proposed algorithm.

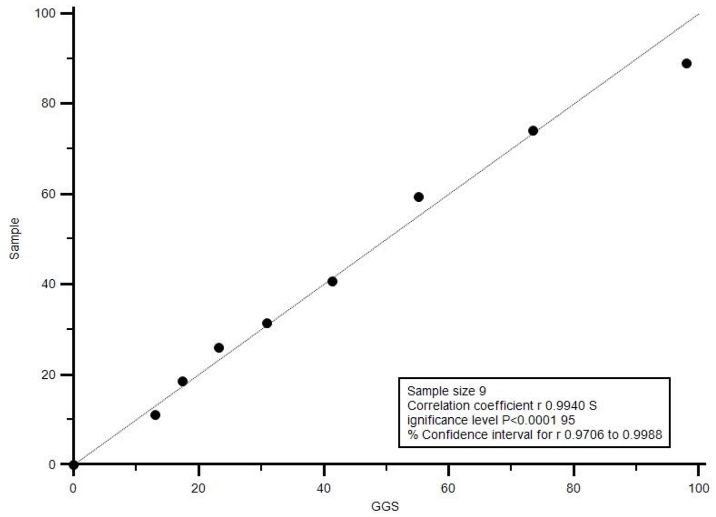

Figure 3. Correlation coefficient between the quartz sand sample and the GGS equation.

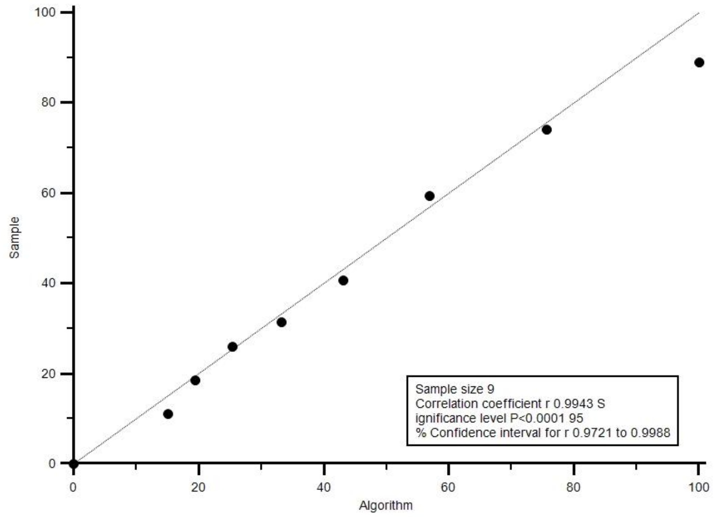

Figure 4. Correlation coefficient between the quartz sand sample and the proposed algorithm.

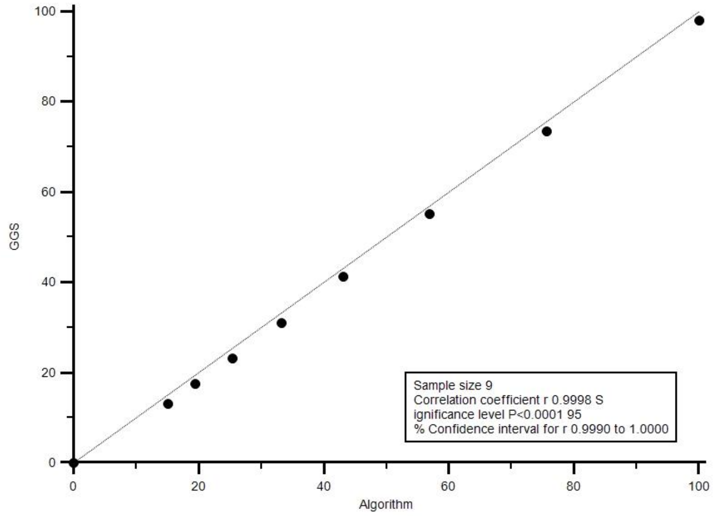

Figure 5 – Correlation coefficient between the GGS equation and the proposed algorithm.

The correlation coefficient obtained between the quartz sample and the GGS equation was equals to 0.9940, which indicates a very good agreement and 0.9943 between the quartz sand sample and the proposed algorithm. When comparing the equation and the proposed algorithm the found correlation coefficient was 0.9998, which indicates that was possible to generate particles diameter with a very strong correlation with the GGS equation.

4. CONCLUSIONS

In this paper we tried to generate the diameter of particles using a size distribution function. This procedure is necessary when the simulation of particles systems is desired and the size of the particles is not the same, or don´t follows a simple statistic distribution. The found results indicate that the algorithm works as good as the size distribution function, once the proposed algorithm depends of the size distribution function and its agreement with the particle system.

5. ACKNOWLEDGES

The authors thank financial support from the Brazilian agencies CNPq, CAPES, FAPEG and FUNAPE and boot universities Federal University of Minas Gerais and Federal University of Goiás, for support and allow the research.

6. REFERENCES

King, R.P. Modeling and Simulation of Mineral Processing Systems. Butterworth-Heinemann, 1st edition, 2001, 416 p.

Macías-García, A., Cuerda-Correa, E.M., Díaz-Díez, M.A. Application of the Rosin–Rammler and Gates–Gaudin–Schuhmann models to the particle size distribution analysis of agglomerated cork. Materials Characterization, 52(2), p. 159–164, 2004.

Rosin, P., Rammler, E. The laws governing the fineness of powdered coal. Journal of the Institute of Fuel, 7, 29, 1933.

Schuhmann, R., Jr. Principles of comminution, 1 - Size distribution and surface calculation, Tech. Publs. AIME, no. 1189, 11, 1940.

Wills, B.A., Napier-Munn, T.J. Wills' Mineral Processing Technology, 7th Edition: An Introduction to the Practical Aspects of Ore Treatment and Mineral Recovery, p. 456, 2006.