Fifth LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2007)
"Developing Entrepreneurial Engineers for the Sustainable Growth of Latin America and the Caribbean:
Education, Innovation, Technology and Practice"
29 May – 1 June 2007, Tampico, México.

Some STL File Generation Issues in Rapid Prototyping

Boppana V. Chowdary
The University of the West Indies
Department of Mechanical and Manufacturing Engineering
St Augustine, Trinidad,
chowdary@eng.uwi.tt

Divesh R. Sahatoo
The University of the West Indies
Department of Mechanical and Manufacturing Engineering
St Augustine, Trinidad,
drsahatoo@yahoo.com

Raj Bhatti
University of Greenwich
Medway School of Engineering
Chatham Maritime
Kent, UK,
r.bhatti@gre.ac.uk

Abstract

The stereolithography (STL) file format was introduced by 3D Systems and has become the de facto industry standard. However, this is not an ideal format due to the existence of some inherent problems. Due to the non-robustness of commercial CAD systems, the choice of tolerance is a major issue when saving STL files. These tolerances will affect the file size, the number of triangular facets and the number of STL file errors. The primary focus of this paper is to outline the relationship between tolerance and these issues in the conversion of CAD models to STL files. To meet the aim of this study, the file size and the number of triangular facets were recorded after STL file generation while the number of file errors was obtained after scanning. Some interesting relationships were noticed and are documented. It is noted that the number of errors decreases with increasing tolerance and that in general there are more errors in the ASCII file format than in the binary equivalent. This paper is intended to provide to the practitioner an idea of how the different factors mentioned are affected by tolerance and allow them to choose the best values based on the findings.

Keywords: Rapid Prototyping, STL File Generation, Software Errors, Tolerance, File Type.

1. Introduction

Rapid Prototyping (RP) is a technique for direct conversion of 3D computer-aided design (CAD) data into a physical prototype using the STL file format. In the STL file format, the object's entire geometry is divided into triangles with three vectors representing the vertices of each triangle and a normal pointing outward. This was introduced by 3D Systems and has become the *de facto* industry standard. However, this is not an ideal format and none of the alternative innovative formats has been accepted yet as a replacement for STL. The main advantage of STL is its simplicity and independence of specific CAD modeling methods (Zhang et al., 2003).

Unfortunately this file, although widely used, has serious drawbacks associated with it. This file is notoriously large and is known to be prone to structural and geometrical errors. Also, the storage required for this file format is very high (Niekerk and Ehlers, 2000). It is therefore important to understand where the STL file errors stem from and what are the issues associated with the conversion of the CAD model to the STL format (Fadel and Kirschman, 1996). In the tessellation of the CAD model to its STL file equivalent, there are a number of issues which will influence the choice of tolerance and hence the accuracy of RP output.

This paper outlines the relationship between tolerance and a host of issues encountered in the conversion of CAD models to the RP compatible STL file format. To meet the aim of the research, factors such as the STL file size, the number of triangular facets and the number of errors in the STL file are considered. This paper is intended to provide to the practitioner an idea of how the different factors mentioned are affected by tolerance and allow them to choose the best values based on the findings.

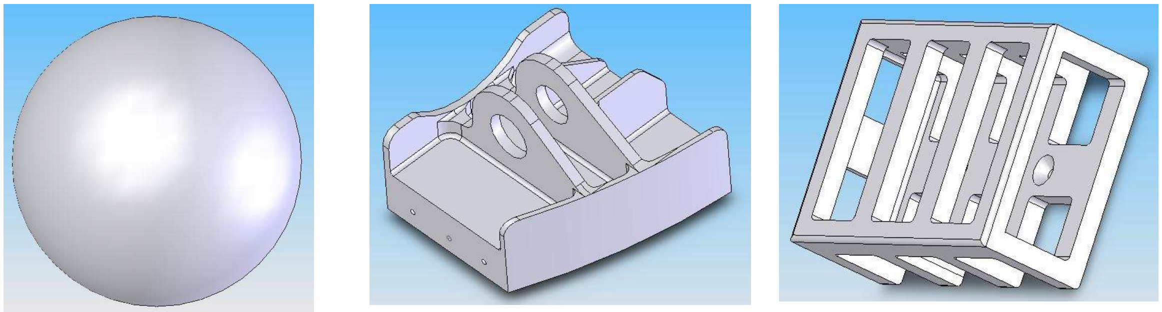
2. Background and Motivation of the Study

The focus of this study is mainly on tessellated files, primarily in STL format. There are many published items of literature and research which have dealt to some extent with the issue of accuracy and hence tolerance in RP. It is noted however that the tolerance issue is not addressed adequately in the past. A few research papers give only a superficial view on the matter without an in-depth analysis. Rock and Wozny (1991) proposed a flexible file format for solid freeform fabrication. Guduri et al. (1993) generate contour files directly from the CAD representation, by passing the triangulation step used by most users. Vuyuru et al. (1994) also describe bypassing the STL step and generating directly non uniform rational basis spline (NURBS) contours that are then segmented for processing. Frank and Fadel (1994) study the orientation issues to reduce errors. Fadel and Kirschman (1996) discuss other formats used to bridge the CAD to RP gap. Koc et al. (2000) present a new approach to reconstruct the slicing contour data to improve the accuracy of the STL file while reducing the necessary data file size. Dimitrov et al. (2006) also considered the accuracy issue especially in the context of 3D printing (3DP) process. Their emphasis in particular was on how the material (powder) used in the 3DP process influences the accuracy. Further, the authors outlined the measurement of tolerance and the associated errors and deviations. More details of the various factors such as the STL file size, the number of triangular facets and the number of STL file errors are available in Fadel and Kirschman (1996), Niekerk and Ehlers (2000), Zhang et al. (2003) and Noorani (2006).

Many attempts at controlling the CAD software related errors have been published in the past. In order to keep control over the design process, a proper understanding of the various errors and the effect of each error on the RP output is needed. In the past also, the effect of tolerance on STL file generation was dealt to some extent. However, there was less emphasis on the relationship between tolerance and a host of factors encountered in the conversion of CAD models to the STL file format. Based on the state of the art, an attempt has been made in this study, to provide the practitioner with a better understanding of the CAD to the RP machine compatible file generation issues such as how the STL file size, the number of triangular facets and the number of errors are affected by tolerance. It can allow for decisions to be made on what will be affected if tolerance has to be sacrificed because of file size or other factors. It is noted that very few studies have dealt with the factors in relation to tolerance (Noorani, 2006; Fadel and Kirschman, 1996; Dimitrov et al. 2006). In particular, emphasis has been placed on the analysis and presentation of software related errors, which occur due to the conversion of the CAD model to the STL format as well as the relationships between tolerance and the factors mentioned previously.

3. Model Building and Experimental Setup

To meet the aim of the study, three different objects (Sphere, Alblock and Battery Carrier) of varying geometrical complexity were selected to model and perform tolerance tests for determination of the relationships with the file size, the number of triangular facets and the number of errors in the object. The three objects used are shown in Figures 1 (a)-(c).



(a) Sphere (b) Alblock (c) Battery Carrier

Figure 1: The three objects selected for the study

The three models used in the current study for testing was created using the SolidWorks 2006 package. To test the effect of tolerance on the factors selected (file size, the number of triangular facets and the number of errors), various STL files were generated from the CAD models by varying the file type (binary or ASCII) and also by varying the tolerance values (available in the software package) by using values available for each object and also values which were common to all three objects.

For each of the three objects, a total of 50 STL files were created. Five values of deviation tolerance were used with two of them being common to all three objects (0.05 and 0.09) in the interest of comparison. These values were used with five angular tolerance values for both ASCII and binary file types. For instance, 5 deviation tolerance values for an angular tolerance of 10°, binary and ASCII each, making it 10 files. This was also done for the other five angular tolerance values (10 files for each angular tolerance value) and hence 50 STL files were created (for each model) and a sample list is presented in Table 1.

The reason for using the specified number of STL files is because the testing procedures are very cumbersome and time consuming and hence the selected number would be ideal for reasonable testing within a specific time period and would give enough evidence to be used for inference and decision making. The following attributes were recorded for each generated STL file: deviation and angular tolerance; file type, STL and CAD file size; number of triangular facets; number of face and gap errors. The results were quite interesting and are documented in the next section.

Table 1 Number of STL files generated

File Number	Angular Tolerance (degrees)	STL Type	Deviation Tolerance (mm)	File Number	Angular Tolerance (degrees)	STL Type	Deviation Tolerance (mm)
1	5	Binary	0.05	26	15	ASCII	0.05
2	5	Binary	0.07	27	15	ASCII	0.07
3	5	Binary	0.09	28	15	ASCII	0.09
4	5	Binary	0.10	29	15	ASCII	0.10
5	5	Binary	0.20	30	15	ASCII	0.20
6	5	ASCII	0.05	31	20	Binary	0.05
7	5	ASCII	0.07	32	20	Binary	0.07
8	5	ASCII	0.09	33	20	Binary	0.09
9	5	ASCII	0.10	34	20	Binary	0.10
10	5	ASCII	0.20	35	20	Binary	0.20
11	10	Binary	0.05	36	20	ASCII	0.05
12	10	Binary	0.07	37	20	ASCII	0.07
13	10	Binary	0.09	38	20	ASCII	0.09
14	10	Binary	0.10	39	20	ASCII	0.10
15	10	Binary	0.20	40	20	ASCII	0.20
16	10	ASCII	0.05	41	25	Binary	0.05
17	10	ASCII	0.07	42	25	Binary	0.07
18	10	ASCII	0.09	43	25	Binary	0.09
19	10	ASCII	0.10	44	25	Binary	0.10
20	10	ASCII	0.20	45	25	Binary	0.20
21	15	Binary	0.05	46	25	ASCII	0.05
22	15	Binary	0.07	47	25	ASCII	0.07
23	15	Binary	0.09	48	25	ASCII	0.09
24	15	Binary	0.10	49	25	ASCII	0.10
25	15	Binary	0.20	50	25	ASCII	0.20

4. Results

Several graphs relating the file size, the number of triangular facets and the number of errors with deviation and angular tolerances for the three objects of varying shapes and sizes were plotted. The relationships between the file size and the deviation tolerance for all three objects were found somewhat similar in shape as shown in Figure 2. However, the relationships for the “Sphere” file size with angular tolerance are straight horizontal lines (constant file size with increasing angular tolerance) while the relationships for the “Alblock” and “Battery Carrier” are similar in shape with a decrease in file size with increasing angular tolerance (refer Figure 3). It is also noted that the ASCII version of the STL file is larger than the binary equivalent. The number of triangular facets decreases with increasing deviation tolerance and are similar in shape as shown in Figure 4. However, with regard to the “Sphere” a dissimilar relationship was seen as the number of triangular facets remains constant with increasing angular tolerance as shown in Figure 5.

For all the three objects it was noticed that the number of errors decreases with increasing deviation tolerance for the “Sphere” and “Alblock” while there are little or no errors present in the “Battery Carrier” STL file (refer Figure 6). However, with increasing angular tolerance, there is a decrease in errors for the “Alblock” and “Battery Carrier” to a certain point after which the number of errors remains fairly constant. For the “Sphere”, however, the number of errors remains constant with changing angular tolerance as shown in Figure 7.

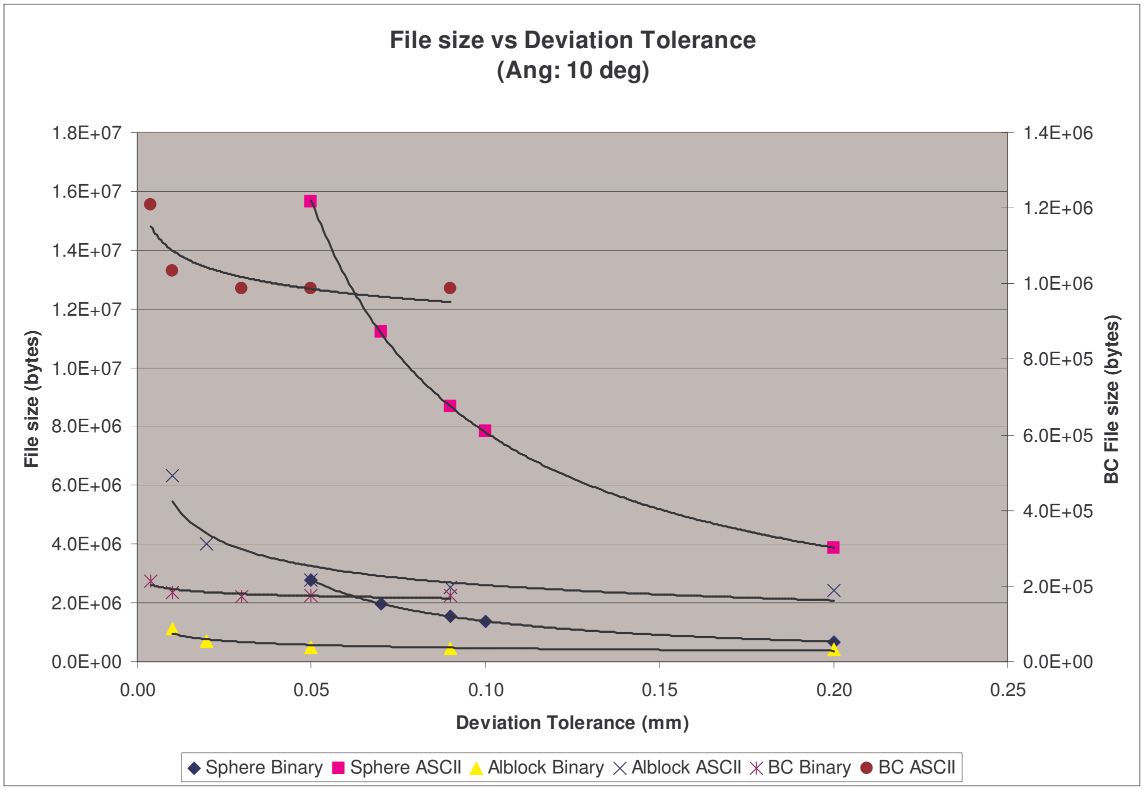


Figure 2: File size versus deviation tolerance (ang. tol. = 10 degrees)

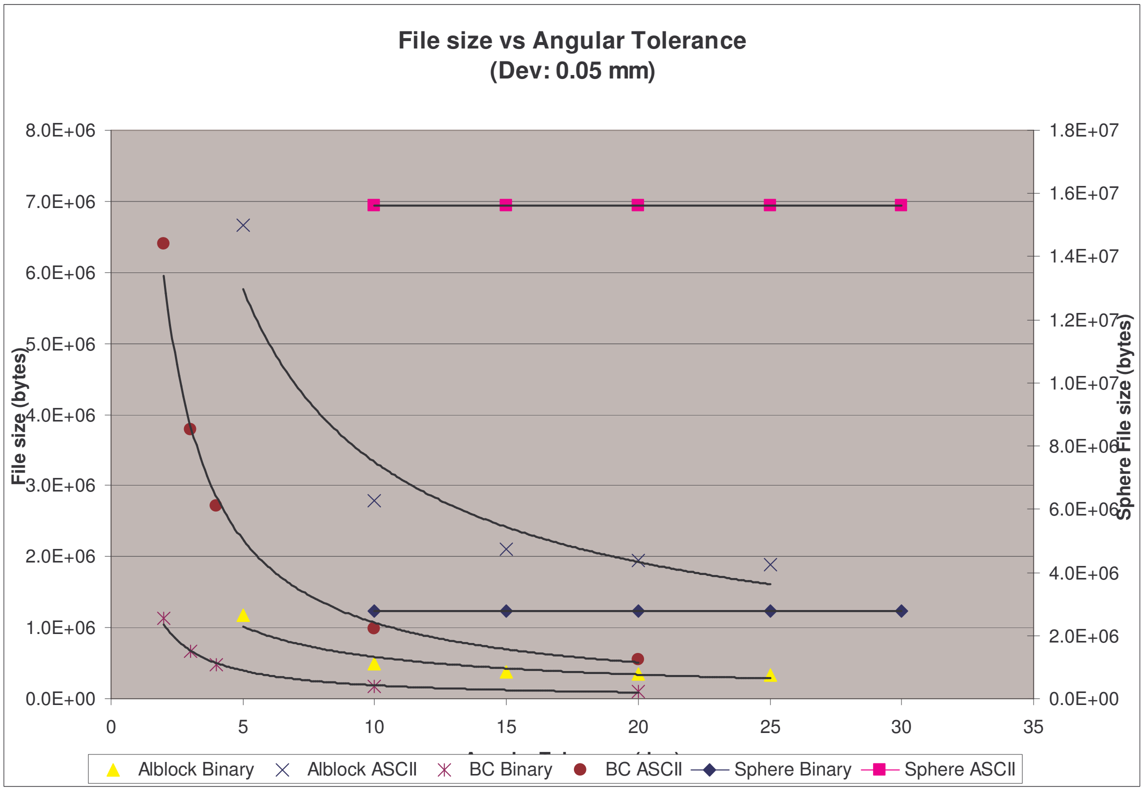


Figure 3: File size versus angular tolerance (dev. tol. = 0.05 mm)

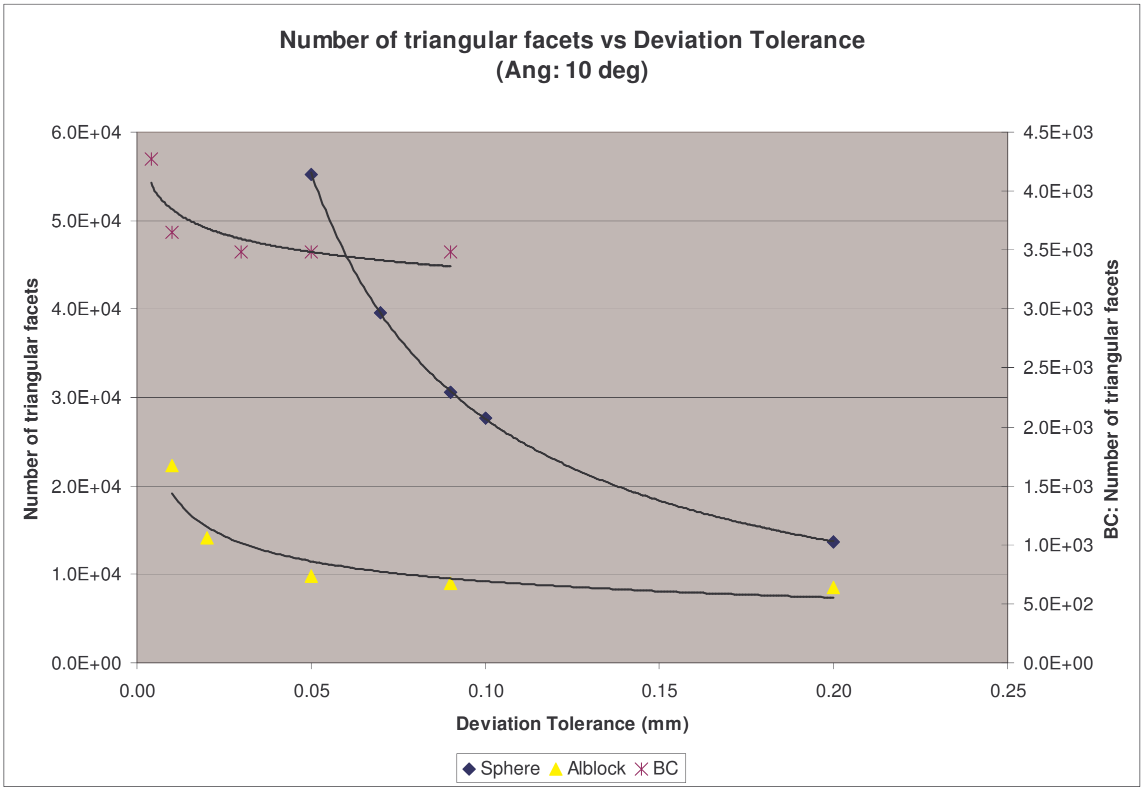


Figure 4: Number of triangular facets versus deviation tolerance (ang tol. = 10 degrees)

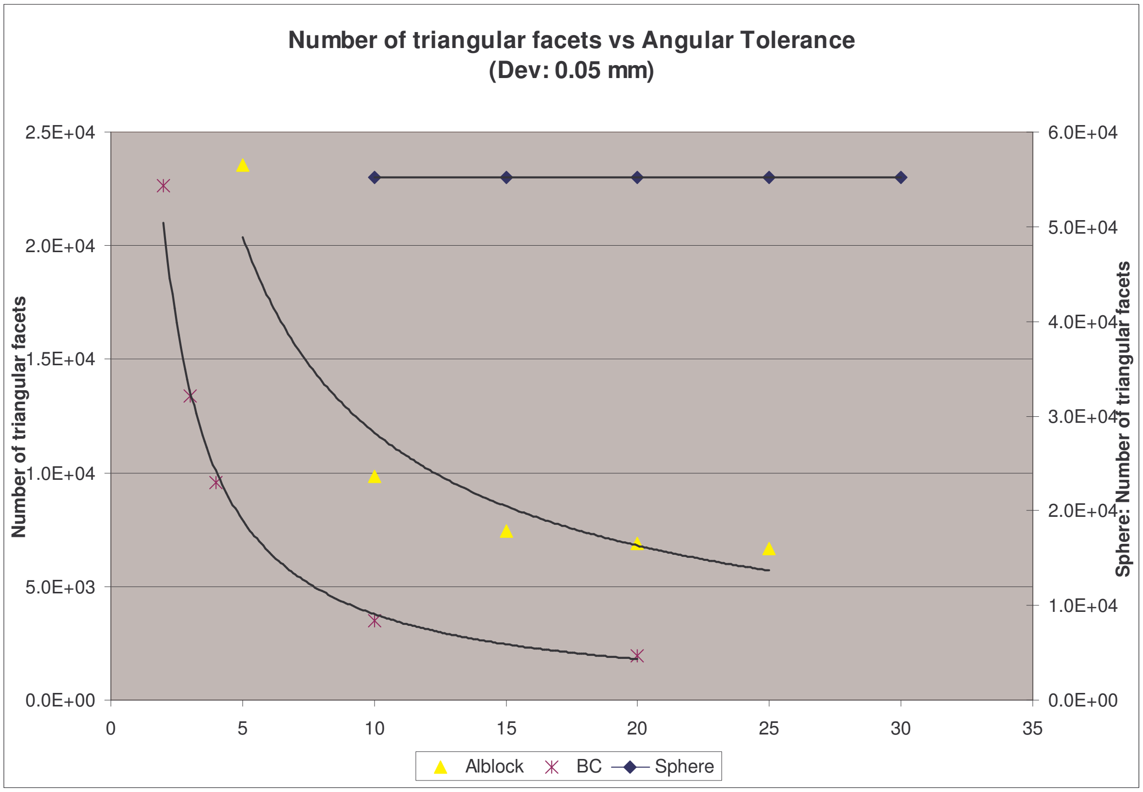


Figure 5: Number of triangular facets versus angular tolerance (dev. tol. = 0.05 mm)

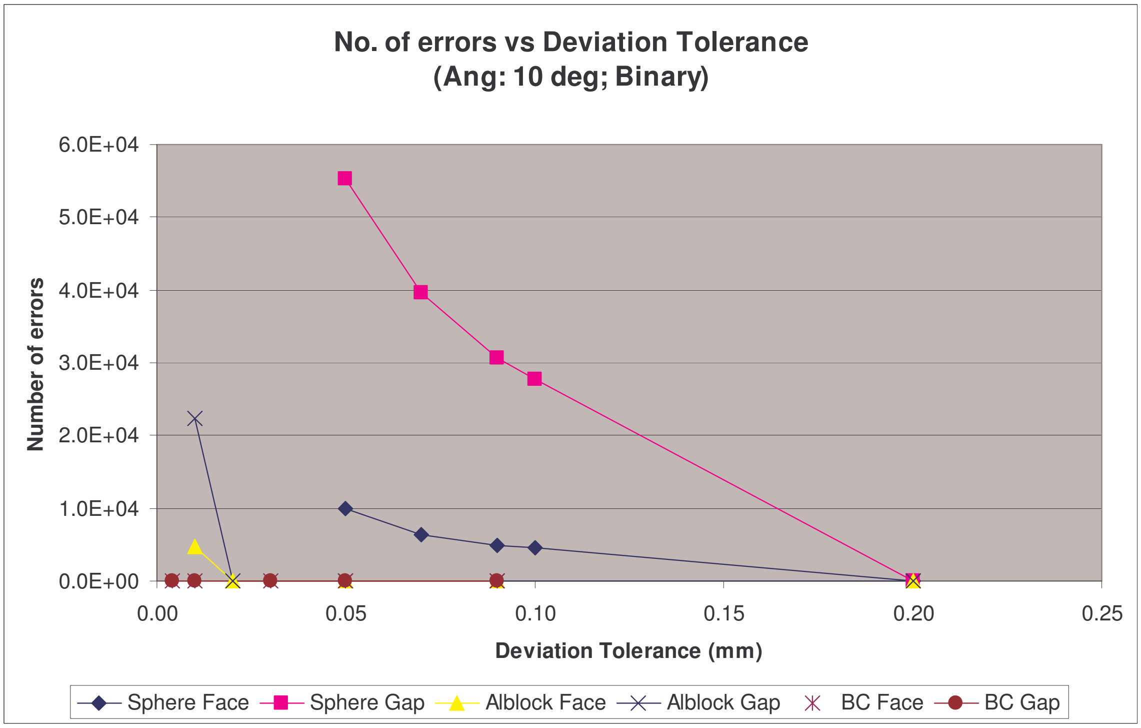


Figure 6: Number of errors versus tolerance (ang. tol. = 10 degrees, binary)

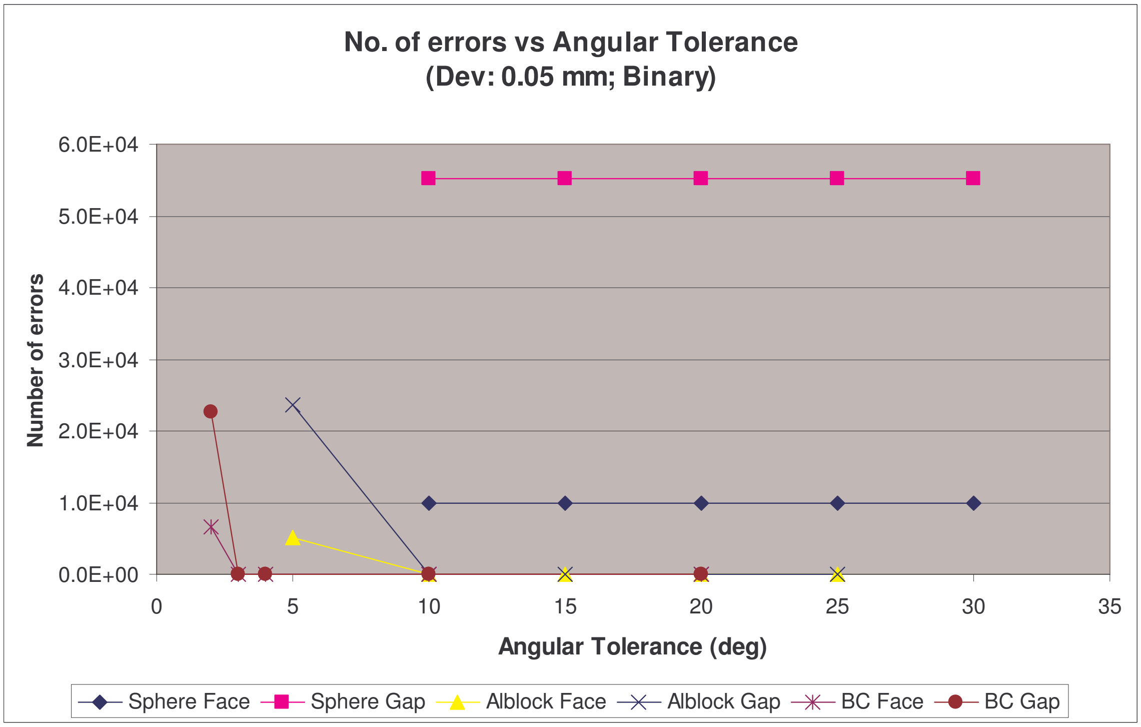


Figure 7: Number of errors versus tolerance (dev. tol. = 0.05 mm, binary)

5. Discussion

From the results shown, the file size decreases with increasing deviation tolerance (refer Figure 2). It can also be seen that for the respective binary or ASCII curve, the “Sphere” has the largest values and the “Battery Carrier” has the smallest. Since the “Sphere” has the most curved surfaces on it and the “Battery Carrier” has the least, this in fact shows that deviation tolerance takes a greater effect on objects with more curved surfaces.

Experimental results show a decrease in file size with increasing angular tolerance for the “Alblock” and the “Battery Carrier” while the file size for the “Sphere” remains constant (Figure 3). Since the surface curvature for the “Sphere” is constant unlike the other two objects, this would suggest that the angular tolerance value would have no effect on the file size of the “Sphere” STL or any object which has constant surface curvature.

A decrease in the number of triangular facets with increasing deviation tolerance for all the three selected objects is exhibited in this study (Figure 4). Furthermore, it can be seen that the “Sphere” has the most number of triangular facets while the “Battery Carrier” has the least. This again demonstrates the fact that deviation tolerance would have a greater effect on objects which have more curved surfaces. It also demonstrates that the “Sphere” is the most complicated of the three objects since it has the most triangular facets (in curved surfaces) while the “Battery Carrier” is the least complicated in this regard.

From the experimental results, interesting relationships were observed. There is a decrease in the number of triangular facets with increasing angular tolerance for the “Alblock” and the “Battery Carrier”. The graph concerning the “Sphere” however shows a constant number of triangular facets with increasing angular tolerance (Figure 5). This again supports the fact that angular tolerance has no effect on objects (like Sphere) that have constant surface curvature.

Figures 6-7 show that the “Sphere” has the most errors while the “Battery Carrier” has the least. The number of errors for the “Sphere” and “Alblock” decrease with increasing angular tolerance while the “Battery Carrier” has very little or no errors (Figure 6). Since the “Sphere” has the most triangular facets while the “Battery Carrier” has the least, this would suggest that if more triangular facets are used to represent the surface then it provides an increased risk of these developing errors. Hence, objects with more curved surfaces are more prone to developing STL errors. It can also be noted that the ASCII type files have more errors than their binary equivalent. This is more so in objects with more curved surfaces.

A decrease in the number of errors with increasing angular tolerance was found for the “Alblock” and the “Battery Carrier”. The number of errors for the “Sphere” with increasing angular tolerance is constant (Figure 7). This again reinforces the fact that angular tolerance has no effect on an object or parts of an object with constant surface curvature.

6. Conclusions

The choice of tolerance plays a vital role when converting a CAD model to the STL file format. From the current study it is clear that the part complexity has a direct bearing over the file size hence the “Sphere” is the most complex followed by the “Alblock” then the “Battery Carrier”. Furthermore, it can be concluded that the more complex the object is (means more curved surfaces), the larger the STL file size would be. This is as expected since a more complicated object would require more data to store its geometry and hence a greater file size. In this study it was also found that the more complex the object i.e. the “Sphere”, the larger the number of triangular facets. But the number of triangular facets for the “Sphere” remains constant with angular tolerance variation proving that varying angular tolerance has no effect on parts of an object with constant surface curvature.

From the results obtained, it can be seen that the most complex object i.e. the “Sphere”, had the most number of STL errors. In general, the number of errors decreased with increasing tolerance. In all the three objects it was seen that there were more errors in the ASCII type file than in the binary equivalent, although in some cases the difference was not significant. So as shown in the current study, tolerance is a very important factor in the generation of STL files and can result in not only either the acceptance or rejection of the RP model, but more critically its subsequent use in, for example, final tooling and investment casting. Work in this direction is in progress.

References

3D Systems, Inc. (1989), *SLA-250 Users Reference Manual, 3D Systems Publications*, Valencia, CA.

Choi, S.H. and Kwok, K.T. (2002), “A tolerant slicing algorithm for layered manufacturing”, *Rapid Prototyping Journal*, Vol 8, No 3, pp. 161-179.

Dimitrov, D. and van Wijck, W., Schreve, K., de Beer, N. (2006), “Investigating the achievable accuracy of three dimensional printing”, *Rapid Prototyping Journal*, Vol 12, No 1, pp. 42-52.

Fadel, G.M. and Kirschman, C. (1996), “Accuracy issues in CAD to RP translations”, *Rapid Prototyping Journal*, Vol 2, No 2, pp. 4-17.

Frank, D. and Fadel, G.M. (1994), “Preferred direction of build for rapid prototyping processes”, *Fifth International Conference on Rapid Prototyping, Dayton, OH*, pp. 191-200.

Gibson, I., Cheung, L.K., Chow, S.P., Cheung, W.L., Beh, S.L., Savalani, M. and Lee, S.H. (2006), “The use of rapid prototyping to assist medical applications”, *Rapid Prototyping Journal,* Vol 12 No 1, pp. 53–58.

Gopakumar, S. (2004), “RP in medicine: a case study in cranial reconstructive surgery”, *Rapid Prototyping Journal,* Vol 10, No 3, pp. 207–211.

Guduri, S., Crawford, R.H. and Beaman, J.J. (1993), “Direct generation of contour files from constructive solid geometry representations”, *Solid Freeform Fabrication Proceedings, Austin, TX*, pp. 291-302.

Hearn, E.J. (2003), “Mechanics of Materials I: An Introduction to the Mechanics of Elastic and Plastic Deformation of Solids and Structural Materials”, Butterworth-Heinemann, Jordan Hill, Oxford.

Kai, C.C., Jacob, G.G.K., and Mei, T. (1997), “Interface between CAD and rapid prototyping systems. Part 1: A study of existing interfaces”, *Rapid Prototyping Journal,* Vol 13, No 8, pp. 566–570.

Koc, B., Ma, Y. and Lee, Y-S. (2000), “Smoothing STL files by Max-Fit biarc curves for rapid prototyping”, *Rapid Prototyping Journal*, Vol 6, No 3, pp. 186 – 203.

Lanzetta, M. and Sachs, E. (2003), “Improved surface finish in 3D printing using bimodal powder distribution”, Rapid Prototyping Journal, Vol 9, No 3, pp. 157-166.

Noorani, R. (2006), “Rapid Prototyping: Principles and Applications”, Wiley, U.S.A.

Rao, P.N., Lerner, Y. and Kouznetsov, V. (2003), “Rapid Prototyping Applications in Metal Casting”, *Institution of Engineers Journal, Malaysia*, Vol. 64, No.3, pp. 1-7.

Rock, S.J. and Wozny, M.J. (1991), “A flexible file format for solid freeform fabrication”, *Solid Freeform Fabrication Proceedings, Austin, TX*, pp. 1-12.

SolidWorks Corporation (1995-2005), *SolidWorks 2006 Online User’s Guide,* 1995-2005.

Tata, K., Fadel, G., Bagchi, A. & Aziz, N. (1998), “Efficient slicing for layered manufacturing”, *Rapid Prototyping Journal*, Vol 4, No 4, pp. 151–167.

Van Niekerk, G.J. and Ehlers, E.M. (2000), “Intelligent Stereolithography File Correction”, *Intelligent Systems in Design and Manufacturing III, Proceedings of SPIE, Vol 4192, pp. 54 – 62.*

Vuyyuru, P., Kirschman, C., Fadel, G.M., Bagchi, A. and Jara-Almonte, C. (1994), “A NURBS based approach for rapid prototyping realization”, *Fifth International Conference on Rapid Prototyping, Dayton, OH*, pp. 229-40.

Zhang, L.C., Han, M. and Huang, S.H. (2003), “CS file - An improved interface between CAD and rapid prototyping systems”, *International Journal of Advanced Manufacturing Technology*, Vol 21, No 1, pp. 15-19.