1.5in 0.6in 1.0in 0.8in 20pt 0.25in 9pt 0.3in

University of Buea

Faculty of Science

Department of Computer Science

# *Implementation Of A Heart-shaped Primitive*

*A thesis submitted to the Faculty of Science*
*in partial fulfilment of the requirements for the degree of*
*Master of Science in Computer Science*
*By*
*Isaac Kamga Mkounga*
*( SC09B676 )*

*Supervisor:*
Prof. Emmanuel Kamgnia

*Date:*
July 2015

# Declaration

I, Isaac Kamga Mkounga, Matriculation Number (SC09B676) declare that this thesis titled, *Implementation Of A Heart-shaped Primitive* and the work presented in it are my own. I confirm that this work was done wholly while in candidature for a Master of Science in Computer Science at the University of Buea. Where I consulted the published work of others, this has been clearly acknowledged. This work has not been previously submitted for a degree or any other qualification at this University or any other institution.

Signed: ______________________ Date: ______________________

Isaac Kamga Mkounga
(SC09B676)

# Dedication

***To My Family***

# Certification

This is to certify that this research project titled *Implementation Of A Heart-shaped Primitive* is the original work of Isaac Kamga Mkounga, Matriculation Number (SC09B676), a Master of Science in Computer Science student of the Department of Computer Science in the Faculty of Science at the University of Buea.

Supervisor: ______________________ Date: ______________________

Prof. Emmanuel Kamgnia

HOD/CSC: ______________________ Date: ______________________

Dr Denis Nkweteyim

Dean/FS: ______________________ Date: ______________________

Prof Ayonghe Samuel

# Acknowledgement

Firstly, I thank our Lord Jesus Christ for giving me life and good health during my stay at University Of Buea and for granting me the patience to go through this program.

I am grateful to the Vice Chancellor of the University Of Buea and her collaborators for maintaining an enabling environment where graduate students can learn and express their utmost research potential. I am especially indebted to Professor Joyce B.M. Endeley, currently the Director of the Higher Technical Teachers' Training College of the University of Buea in Kumba and Professor Theresa NkuoAkenji, the Deputy Vice Chancellor for Internal Control and Evaluation who authorized the connection of internet facilities to the Postgraduate Computer Science Laboratory during the implementation of this project. I wish to sincerely thank all the staff of the Department of Computer Science for their advice and stimulating conversations during the writing of this thesis. Special thanks go to Professor Emmanuel Kamgnia and Dr. Denis Nkweteyim for meticulously proofreading through and correcting this work while making their wide spectrum of research experience and books on graphics available to me.

My appreciation goes to the Google Open Source Programs Office (OSPO) – Mary Radomile, Carols Smith, Cat Allman and Stephanie Taylor and the United States Army Research Laboratory BRL-CAD team for their sponsorship of this project under the auspices of the 2013 Google Summer Of Code program. Special thanks go to Mr. Christopher Sean Morrison and Mr. Erik Greenwald for their valuable advice and mentorship during the implementation of this project. I am also indebted to *Nominal Animal* and the mathematician Dr. Titus Piezas for tips on ray-tracing the heart surface.

Many thanks go to my family for their assistance and encouragement throughout this program.This thesis is dedicated to them. I also thank the friends I made during the course of this program whether from University of Buea or the Developers community, for bringing all the fun they came along with– I appreciate your company and I'm happy to know I was not alone.

# *Abstract*

*In this thesis titled Implementation Of A Heart-shaped Primitive, we aimed at demonstrating the engineering of a heart-shaped primitive within BRL-CAD package.Using a case study approach, we designed the heart-shaped primitive's data structure, wrote necessary callback functions and tested them using BRL-CAD's testing infrastructure. We showed that the Laguerre-based root solver is indeed a sure-fire iterative method for finding roots of polynomials and ascertained it's stability on sextic equations. This work provides a guideline for the development of primitives within Computer-Aided Design (CAD) software by highlighting the implementation of geometrically-useful properties for any primitive within BRL-CAD.*

# Contents

# List of Figures

# List of Tables

# *Abbreviations*

| | |
|---|---|
| **CAD** | **C**omputer **A**ided **D**esign |
| **BRL-CAD** | **B**allistic **R**esearch **L**aboratory **C**omputer **A**ided **D**esign |
| **CSG** | **C**onstructive **S**olid **G**eometry |
| **NURBS** | **N**on-**U**niform **R**ational **B-S**plines |
| **NMG** | **N**on - **M**anifold **G**eometry |

# Chapter 1

# INTRODUCTION

## 1.1 Historical Background

Throughout our long history, we humans have always sought for means to express our creativity–ways to communicate our ideas through writing, sculpting, painting, carving, architecture and drawing.As a matter of fact, paleolithic cave representations of animals at least 32,000 years ago in Southern France, ink drawings and paintings of human figures as well as writings in hieroglyphics on papyrus in the pyramids of ancient Egypt is indicative of the fact that our need to express our individuality goes back to antiquity.Before the renaissance, drawing was treated as a preparatory stage for painting and sculpting.The wide availability of drawing instruments such as pens and pencils and most especially paper made master draftsmen like Leonardo Da Vinci, Raphael and Michelangelo around the world to lift drawing to an art in its own right.Thus, drawing stood out as the most popular and fundamental means of public expression in human history and is one of the simplest and most efficient means of communicating visual ideas.[1]

The drawing board era where paper, pens, pencils, rulers and ink prevailed has been relegated to the background in this information age which is powered by ubiquitous computer technology. Craving the unity of science and art, this essentially binary-sequenced revolutionary device called the computer married the artistic and engineering forms of drawing into an androgynous one called ComputerAided Geometric Design or geometric modeling for short. Geometric modeling currently involves the use of computers to aid in the creation, manipulation, maintenance and analysis of representations of the geometric shapes of two and threedimensional objects [2]. It is the outgrowth of convergent motivations and developments from several works of life as outlined below.

- In the 1950s, the need to automate the engineering drawing process led to electronic drawings which could be archived and modified more easily, could be easily verified and errors could be eliminated from mechanical designs without introducing new ones.These computer drafting systems allowed designers to produce drawings of objects by projecting threedimensional objects unto twodimensional surfaces.

- Then in the 1960s, there was a pressing need for software in the automobile, shipbuilding and aircraft industries to produce computer-compatible descriptions of geometric shapes which can be machined from wood and steel into stamps and dies for the manufacturing and assembling of car parts,ship hulls as well as wings and fuselages using computer numerically controlled tools.

- Later in the 1970s, the growing need for computers to render realistic images of objects as well as animate solid objects pushed research institutes like Xerox Palo Alto Research Center (Xerox PARC) and Apple Computers to make significant contributions to graphical user interface design and computer graphics.

These needs and problems could only be solved by research in fields such as graphics, animation and applications from algebraic geometry. The work of various computer scientists and mathematicians lead to the active development of several commercial packages sponsored by companies such as

Renault, Citroen, Ford and Boeing who could afford the computers capable of performing such lengthy calculations.

Today,geometric modeling is also referred to as Computer Aided Design (CAD), is pronounced "cad" and is routinely used in the design and manufacturing of engineering and architectural structures such as buildings, car parts, ship hulls and aircraft artillery as well as to specify special effects in cartoon movies, music videos and television shows. Indeed, CAD packages provide facilities for designing shapes of solid physical objects and specifying their motion in a way that art and science can unite to create cool designs.

Even though significant progress had been made in basic research and the functionality of commercially available solid modelers like Apple Computer's RenderMan, many solid modelers especially within the open source community are still limited in their geometric features. The open source community is a selforganizing collaborative social network of programmers driven by a passion to solve problems using computers. It has several thousands of its projects on sites that offer services like bug tracking, mailing lists and version control viz Github and Source Forge.These projects are constantly being improved upon by thousands of programmers putting in time and effort to write and debug software without direct monetary pay.

In this thesis, we document the process of developing a heart-shaped primitive, a set of callback functions and procedures which compute geometrically useful properties of solids such as wireframe plotting, database importation and exportation,ray tracing, bounding box calculations, just to name a few, within the Ballistic Research Laboratory Computer Aided Design (BRL-CAD) software package.

BRL-CAD was initiated by the United States Army Research Laboratory in 1983, the same agency which created the E.N.I.A.C., the world's first general purpose computer in the 1940s, to model military systems for the United States government. According to [3], BRL-CAD became born again in 2004 when it joined the open source community with portions of its source code licensed under the Lesser General Public License (LGPL) and Berkeley

Software Distributions (BSD) licenses and has been credited as being the oldest open source repository in continuous development. It supports a wide variety of geometric representations including an extensive set of traditional implicit primitive shapes as well as explicit primitives made from collections of uniform Bspline surfaces, Nonuniform Rational Bspline (NURBS) surfaces, Nonmanifold geometry (NMG) and purely faceted polygonal mesh geometry.

BRL-CAD also focuses on solid modeling aspects of Computer Aided Design. Figure 1.1 below shows a threedimensional model of a Goliath tracked mine, a German engineered remote controlled vehicle used during World War II. This model was created by students new to BRL-CAD in the span of about 2 weeks, starting from actual measurements in a museum.

## 1.2 Importance Of This Work

This work is significant to several stakeholders for several reasons ;

- By raytracing the heart's surface, it demonstrates to the scientific community that the Laguerre zerofinder is indeed a surefire iterative method for finding roots of polynomials and that Laguerre-based root solvers are stable on sextic equations.
- This work incorporates more geometric modeling functionality into the free and open source software community through BRL-CAD, the oldest open source repository in continuous development[3] by going beyond traditional CSG primitives shapes such as tori, spheres, boxes and ellipsoids towards the more complex heart-shape based on a sextic equation. This work provides a guideline for the development of primitives within open source CAD software.
- Given that BRL-CAD is used within governments to model military artillery and for engineering and analysis purposes within academia, this heart-shaped primitive gives BRL-CAD a more loving aura – an

environment where artists can produce cartoon animations as well as design cards, royal seals and banners, gifts and presents for family and communal celebrations such as weddings, family reunions and Valentine's day for entertainment purposes.

## 1.3 Thesis Organisation

This thesis is divided into five (5) chapters. Chapter 1 introduces the study and chapter 2 reviews the literature in the field of geometric modeling. In Chapter 3, we state the problems we intend to solve and our project design. In chapter 4, we discuss the interesting results which we obtained. Finally, in chapter 5, we state the contribution of our work and give possible research directions which can proceed from it.

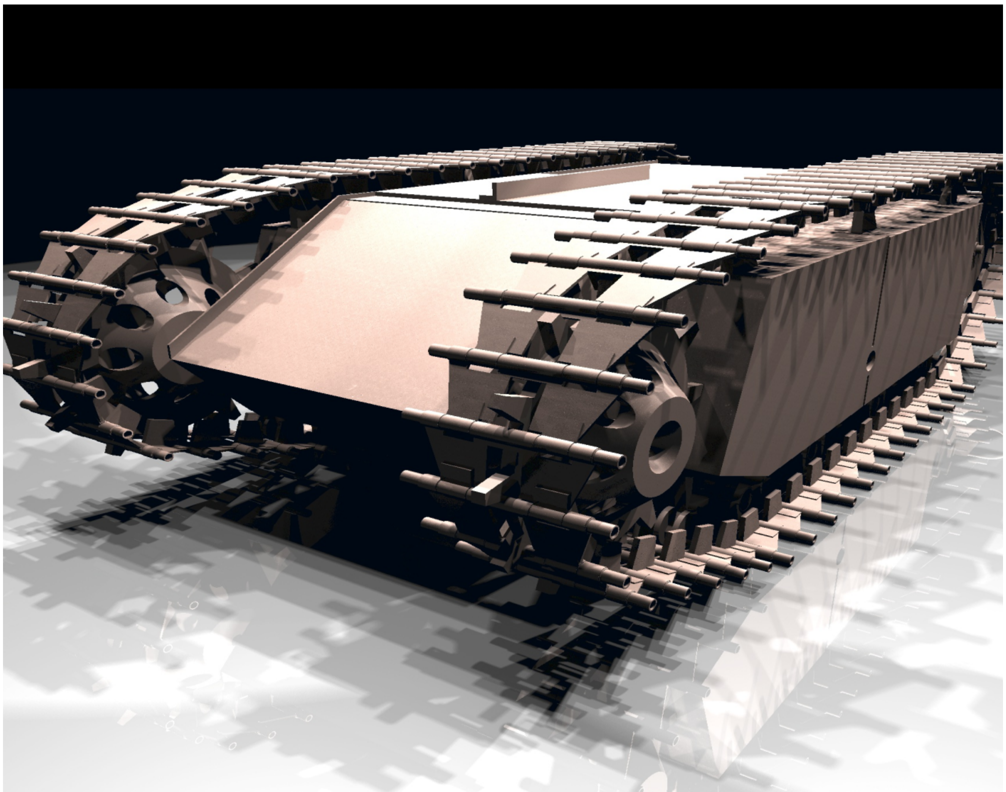

FIGURE 1.1: Model of a Goliath tracked mine

### 1.3.1 A (not so short) Introduction to LaTeX

If you are new to LaTeX, there is a very good eBook – freely available online as a PDF file – called, "The Not So Short Introduction to LaTeX". The book's title is typically shortened to just "lshort". You can download the latest version (as it is occasionally updated) from here:
`http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf`

It is also available in several other languages. Find yours from the list on this page:
`http://www.ctan.org/tex-archive/info/lshort/`

It is recommended to take a little time out to learn how to use LaTeX by creating several, small 'test' documents. Making the effort now means you're not stuck learning the system when what you *really* need to be doing is writing your thesis.

### 1.3.2 A Short Math Guide for LaTeX

If you are writing a technical or mathematical thesis, then you may want to read the document by the AMS (American Mathematical Society) called, "A Short Math Guide for LaTeX". It can be found online here:
`http://www.ams.org/tex/amslatex.html`
under the "Additional Documentation" section towards the bottom of the page.

### 1.3.3 Common LaTeX Math Symbols

There are a multitude of mathematical symbols available for LaTeX and it would take a great effort to learn the commands for them all. The most common ones you are likely to use are shown on this page:
`http://www.sunilpatel.co.uk/latexsymbols.html`

You can use this page as a reference or crib sheet, the symbols are rendered as large, high quality images so you can quickly find the LaTeX command for the symbol you need.

### 1.3.4 LaTeX on a Mac

The LaTeX package is available for many systems including Windows, Linux and Mac OS X. The package for OS X is called MacTeX and it contains all the applications you need – bundled together and pre-customised – for a fully working LaTeX environment and workflow.

MacTeX includes a dedicated LaTeX IDE (Integrated Development Environment) called “TeXShop” for writing your ‘`.tex`’ files and “BibDesk”: a program to manage your references and create your bibliography section just as easily as managing songs and creating playlists in iTunes.

## 1.4 Getting Started with this Template

If you are familiar with LaTeX, then you can familiarise yourself with the contents of the Zip file and the directory structure and then place your own information into the ‘`Thesis.cls`’ file. Section 1.6 on page 11 tells you how to do this. Make sure you read section 1.8 about thesis conventions to get the most out of this template and then get started with the ‘`Thesis.tex`’ file straightaway.

If you are new to LaTeX it is recommended that you carry on reading through the rest of the information in this document.

### 1.4.1 About this Template

This LaTeX Thesis Template is originally based and created around a LaTeX style file created by Steve R. Gunn from the University of Southampton (UK), department of Electronics and Computer Science. You can find his

original thesis style file at his site, here:
`http://www.ecs.soton.ac.uk/~srg/softwaretools/document/templates/`

My thesis originally used the '`ecsthesis.cls`' from his list of styles. However, I knew LaTeX could still format better. To get the look I wanted, I modified his style and also created a skeleton framework and folder structure to place the thesis files in.

This Thesis Template consists of that modified style, the framework and the folder structure. All the work that has gone into the preparation and groundwork means that all you have to bother about is the writing.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution's recommendations.

## 1.5 What this Template Includes

### 1.5.1 Folders

This template comes as a single Zip file that expands out to many files and folders. The folder names are mostly self-explanatory:

**Appendices** – this is the folder where you put the appendices. Each appendix should go into its own separate '`.tex`' file. A template is included in the directory.

**Chapters** – this is the folder where you put the thesis chapters. A thesis usually has about seven chapters, though there is no hard rule on this. Each chapter should go in its own separate '`.tex`' file and they usually are split as:

- Chapter 1: Introduction to the thesis topic

- Chapter 2: Background information and theory
- Chapter 3: (Laboratory) experimental setup
- Chapter 4: Details of experiment 1
- Chapter 5: Details of experiment 2
- Chapter 6: Discussion of the experimental results
- Chapter 7: Conclusion and future directions

This chapter layout is specialised for the experimental sciences.

**Figures** – this folder contains all figures for the thesis. These are the final images that will go into the thesis document.

**Primitives** – this is the folder that contains scraps, particularly because one final image in the 'Figures' folder may be made from many separate images and photos, these source images go here. This keeps the intermediate files separate from the final thesis figures.

### 1.5.2 Files

Included are also several files, most of them are plain text and you can see their contents in a text editor. Luckily, many of them are auxiliary files created by LaTeX or BibTeX and which you don't need to bother about:

**Bibliography.bib** – this is an important file that contains all the bibliographic information and references that you will be citing in the thesis for use with BibTeX. You can write it manually, but there are reference manager programs available that will create and manage it for you. Bibliographies in LaTeX are a large subject and you may need to read about BibTeX before starting with this.

**Thesis.cls** – this is an important file. It is the style file that tells LaTeX how to format the thesis. You will also need to open this file in a text editor and fill in your own information (such as name, department, institution). Luckily, this is not too difficult and is explained in section 1.6 on page 11.

**Thesis.pdf** – this is your beautifully typeset thesis (in the PDF file format) created by LaTeX.

**Thesis.tex** – this is an important file. This is the file that you tell LaTeX to compile to produce your thesis as a PDF file. It contains the framework and constructs that tell LaTeX how to layout the thesis. It is heavily commented so you can read exactly what each line of code does and why it is there. After you put your own information into the '`Thesis.cls`' file, go to this file and begin filling it in – you have now started your thesis!

**vector.sty** – this is a LaTeX package, it tells LaTeX how to typeset mathematical vectors. Using this package is very easy and you can read the documentation on the site (you just need to look at the '`vector.pdf`' file): `http://www.ctan.org/tex-archive/macros/latex/contrib/vector/`

**lstpatch.sty** – this is a LaTeX package required by this LaTeX template and is included as not all TeX distributions have it installed by default. You do not need to modify this file.

Files that are *not* included, but are created by LaTeX as auxiliary files include:

**Thesis.aux** – this is an auxiliary file generated by LaTeX, if it is deleted LaTeX simply regenerates it when you run the main '`.tex`' file.

**Thesis.bbl** – this is an auxiliary file generated by BibTeX, if it is deleted, BibTeX simply regenerates it when you run the main tex file. Whereas the '`.bib`' file contains all the references you have, this '`.bbl`' file contains the references you have actually cited in the thesis and is used to build the bibliography section of the thesis.

**Thesis.blg** – this is an auxiliary file generated by BibTeX, if it is deleted BibTeX simply regenerates it when you run the main '`.tex`' file.

**Thesis.lof** – this is an auxiliary file generated by LaTeX, if it is deleted LaTeX simply regenerates it when you run the main '`.tex`' file. It tells LaTeX how to build the 'List of Figures' section.

**Thesis.log** – this is an auxiliary file generated by LaTeX, if it is deleted LaTeX simply regenerates it when you run the main '`.tex`' file. It contains messages from LaTeX, if you receive errors and warnings from LaTeX, they will be in this '`.log`' file.

**Thesis.lot** – this is an auxiliary file generated by LaTeX, if it is deleted LaTeX simply regenerates it when you run the main '`.tex`' file. It tells LaTeX how to build the 'List of Tables' section.

**Thesis.out** – this is an auxiliary file generated by LaTeX, if it is deleted LaTeX simply regenerates it when you run the main '`.tex`' file.

So from this long list, only the files with the '`.sty`', '`.bib`', '`.cls`' and '`.tex`' extensions are the most important ones. The other auxiliary files can be ignored or deleted as LaTeX and BibTeX will regenerate them.

## 1.6 Filling in the '`Thesis.cls`' File

You will need to personalise the thesis template and make it your own by filling in your own information. This is done by editing the '`Thesis.cls`' file in a text editor.

Open the file and scroll down, past all the '`\newcommand`...' items until you see the entries for '`University Name`', '`Department Name`', etc....

Fill out the information about your group and institution and ensure you keep to block capitals where it asks you to. You can also insert web links, if you do, make sure you use the full URL, including the '`http://`' for this.

The last item you should need to fill in is the Faculty Name (in block capitals). When you have done this, save the file and recompile '`Thesis.tex`'.

All the information you filled in should now be in the PDF, complete with web links. You can now begin your thesis proper!

## 1.7 The 'Thesis.tex' File Explained

The `Thesis.tex` file contains the structure of the thesis. There are plenty of written comments that explain what pages, sections and formatting the LaTeX code is creating. Initially there seems to be a lot of LaTeX code, but this is all formatting, and it has all been taken care of so you don't have to do it.

Begin by checking that your information on the title page is correct. For the thesis declaration, your institution may insist on something different than the text given. If this is the case, just replace what you see with what is required.

Then comes a page which contains a funny quote. You can put your own, or quote your favourite scientist, author, person, etc... Make sure to put the name of the person who you took the quote from.

Next comes the acknowledgements. On this page, write about all the people who you wish to thank (not forgetting parents, partners and your advisor/-supervisor).

The contents pages, list of figures and tables are all taken care of for you and do not need to be manually created or edited. The next set of pages are optional and can be deleted since they are for a more technical thesis: insert a list of abbreviations you have used in the thesis, then a list of the physical constants and numbers you refer to and finally, a list of mathematical symbols used in any formulae. Making the effort to fill these tables means the reader has a one-stop place to refer to instead of searching the internet and references to try and find out what you meant by certain abbreviations or symbols.

The list of symbols is split into the Roman and Greek alphabets. Whereas the abbreviations and symbols ought to be listed in alphabetical order (and this is *not* done automatically for you) the list of physical constants should be grouped into similar themes.

The next page contains a one line dedication. Who will you dedicate your thesis to?

Finally, there is the section where the chapters are included. Uncomment the lines (delete the '`%`' character) as you write the chapters. Each chapter should be written in its own file and put into the 'Chapters' folder and named '`Chapter1`', '`Chapter2`, etc... Similarly for the appendices, uncomment the lines as you need them. Each appendix should go into its own file and placed in the 'Appendices' folder.

After the preamble, chapters and appendices finally comes the bibliography. The bibliography style (called '`unsrtnat`') is used for the bibliography and is a fully featured style that will even include links to where the referenced paper can be found online. Do not under estimate how grateful you reader will be to find that a reference to a paper is just a click away. Of course, this relies on you putting the URL information into the BibTeX file in the first place.

## 1.8 Thesis Features and Conventions

To get the best out of this template, there are a few conventions that you may want to follow.

One of the most important (and most difficult) things to keep track of in such a long document as a thesis is consistency. Using certain conventions and ways of doing things (such as using a Todo list) makes the job easier. Of course, all of these are optional and you can adopt your own method.

### 1.8.1 Printing Format

This thesis template is designed for single sided printing as most theses are printed and bound this way. This means that the left margin is always wider than the right (for binding). Four out of five people will now judge the margins by eye and think, "I never noticed that before.".

The headers for the pages contain the page number on the right side (so it is easy to flick through to the page you want) and the chapter name on the left side.

The text is set to 11 point and a line spacing of 1.3. Generally, it is much more readable to have a smaller text size and wider gap between the lines than it is to have a larger text size and smaller gap. Again, you can tune the text size and spacing should you want or need to. The text size can be set in the options for the '`\documentclass`' command at the top of the '`Thesis.tex`' file and the spacing can be changed by setting a different value in the '`\setstretch`' commands (scattered throughout the '`Thesis.tex`' file).

### 1.8.2 Using US Letter Paper

The paper size used in the template is A4, which is a common – if not standard – size in Europe. If you are using this thesis template elsewhere and particularly in the United States, then you may have to change the A4 paper size to the US Letter size. Unfortunately, this is not as simple as replacing instances of '`a4paper`' with '`letterpaper`'.

This is because the final PDF file is created directly from the LaTeX source using a program called '`pdfTeX`' and in certain conditions, paper size commands are ignored and all documents are created with the paper size set to the size stated in the configuration file for pdfTeX (called '`pdftex.cfg`').

What needs to be done is to change the paper size in the configuration file for `pdfTeX` to reflect the letter size. There is an excellent tutorial on how to

do this here:
`http://www.physics.wm.edu/~norman/latexhints/pdf_papersize.html`

It may be sufficient just to replace the dimensions of the A4 paper size with the US Letter size in the `pdftex.cfg` file. Due to the differences in the paper size, the resulting margins may be different to what you like or require (as it is common for Institutions to dictate certain margin sizes). If this is the case, then the margin sizes can be tweaked by opening up the `Thesis.cls` file and searching for the line beginning with, '`\setmarginsrb`' (not very far down from the top), there you will see the margins specified. Simply change those values to what you need (or what looks good) and save. Now your document should be set up for US Letter paper size with suitable margins.

### 1.8.3 References

The '`natbib`' package is used to format the bibliography and inserts references such as this one [**?** ]. The options used in the '`Thesis.tex`' file mean that the references are listed in numerical order as they appear in the text. Multiple references are rearranged in numerical order (e.g. [**? ?** ]) and multiple, sequential references become reformatted to a reference range (e.g. [**? ? ?** ]). This is done automatically for you. To see how you use references, have a look at the '`Chapter1.tex`' source file. Many reference managers allow you to simply drag the reference into the document as you type.

Scientific references should come *before* the punctuation mark if there is one (such as a comma or period). The same goes for footnotes[1]. You can change this but the most important thing is to keep the convention consistent throughout the thesis. Footnotes themselves should be full, descriptive sentences (beginning with a capital letter and ending with a full stop).

To see how LaTeX typesets the bibliography, have a look at the very end of this document (or just click on the reference number links).

[1]Such as this footnote, here down at the bottom of the page.

### 1.8.4 Figures

There will hopefully be many figures in your thesis (that should be placed in the ‘Figures’ folder). The way to insert figures into your thesis is to use a code template like this:

```
\begin{figure}[htbp]
  \centering
    \includegraphics{Figures/Electron.pdf}
    \rule{35em}{0.5pt}
  \caption[An Electron]{An electron (artist’s impression).}
  \label{fig:Electron}
\end{figure}
```

Also look in the source file. Putting this code into the source file produces the picture of the electron that you can see in the figure below.

Sometimes figures don’t always appear where you write them in the source. The placement depends on how much space there is on the page for the figure. Sometimes there is not enough room to fit a figure directly where it should go (in relation to the text) and so LaTeX puts it at the top of the next page. Positioning figures is the job of LaTeX and so you should only worry about making them look good!

Figures usually should have labels just in case you need to refer to them (such as in Figure 1.2). The ‘`\caption`’ command contains two parts, the first part, inside the square brackets is the title that will appear in the ‘List of Figures’, and so should be short. The second part in the curly brackets should contain the longer and more descriptive caption text.

The ‘`\rule`’ command is optional and simply puts an aesthetic horizontal line below the image. If you do this for one image, do it for all of them.

The LaTeX Thesis Template is able to use figures that are either in the PDF or JPEG file format.



Figure 1.2: An electron (artist's impression).

### 1.8.5 Typesetting mathematics

If your thesis is going to contain heavy mathematical content, be sure that LaTeX will make it look beautiful, even though it won't be able to solve the equations for you.

The "Not So Short Introduction to LaTeX" (available here) should tell you everything you need to know for most cases of typesetting mathematics. If you need more information, a much more thorough mathematical guide is available from the AMS called, "A Short Math Guide to LaTeX" and can be downloaded from:
`ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf`

There are many different LaTeX symbols to remember, luckily you can find the most common symbols here. You can use the web page as a quick reference or crib sheet and because the symbols are grouped and rendered as high quality images (each with a downloadable PDF), finding the symbol you need is quick and easy.

You can write an equation, which is automatically given an equation number by LaTeX like this:

```
\begin{equation}
E = mc^{2}
  \label{eqn:Einstein}
\end{equation}
```

This will produce Einstein's famous energy-matter equivalence equation:

$$E = mc^2 \tag{1.1}$$

All equations you write (which are not in the middle of paragraph text) are automatically given equation numbers by LaTeX. If you don't want a particular equation numbered, just put the command, '`\nonumber`' immediately after the equation.

## 1.9 Sectioning and Subsectioning

You should break your thesis up into nice, bite-sized sections and subsections. LaTeX automatically builds a table of Contents by looking at all the '`\chapter{}`', '`\section{}`' and '`\subsection{}`' commands you write in the source.

The table of Contents should only list the sections to three (3) levels. A '`\chapter{}`' is level one (1). A '`\section{}`' is level two (2) and so a '`\subsection{}`' is level three (3). In your thesis it is likely that you will even use a '`\subsubsection{}`', which is level four (4). Adding all these will create an unnecessarily cluttered table of Contents and so you should use the '`\subsubsection*{}`' command instead (note the asterisk). The asterisk (*) tells LaTeX to omit listing the subsubsection in the Contents, keeping it clean and tidy.

## 1.10 In Closing

You have reached the end of this mini-guide. You can now rename or overwrite this pdf file and begin writing your own '`Chapter1.tex`' and the rest of your thesis. The easy work of setting up the structure and framework has been taken care of for you. It's now your job to fill it out!

Good luck and have lots of fun!

Guide written by —
Sunil Patel: www.sunilpatel.co.uk

# Chapter 2

# LITERATURE REVIEW OF GEOMETRIC MODELING

With the advent of computers which could perform millions of floating point operations in unit time and which are still growing faster, researchers who believed computers could aid the processes of mechanical design and manufacturing were faced with a critical issue – how to represent physical reality using computer software. They sought the best data structures to represent this reality and the most appropriate algorithms to manipulate these representations.

BRL-CAD supports a wide variety of geometric representations including an extensive set of traditional implicit primitive shapes as well as explicit primitives made from collections of uniform Bspline surfaces, nonuniform rational Bspline (NURBS) surfaces, non-manifold geometry (NMG) and purely faceted polygonal mesh geometry. Consequently, in this chapter, we review the existing work done by scholars in the field of geometric modeling which have been applied to the development of BRL-CAD. First of all, it introduces the issue of representation and the notion of representation schemes.Then, it summarizes developments in wireframe modeling, surface modeling, solid modeling and non-manifold modeling (aka nonmanifold geometry or nmg for short) with a keen eye on the algorithms underlying them.

As we progress in our literature review from older forms of geometric modeling to newer ones, we will discover that representation schemes were closely linked to algorithmic efficiency and that it has always been normal to expect designers to switch to newer ones in response to the improvements in algorithmic performance.Despite these enhancements in algorithmic efficiency within the designer community, we cannot say with complete certainty whether traditional representation schemes can be relegated to the background. We can only conclude that old and new representation paradigms coexist and that research led to representation schemes which supplemented the repertoire of geometric modeling.

## 2.1 Representation Schemes

A representation **R** of a solid or representation for short is a subset of three-dimensional Euclidean space denoted $\mathbb{E}^3$ which models a physical solid. According to [5], Requicha and Tilove stated that point set topology provided a formal language for describing the geometric properties of solids and they also threw more light on the mathematical characteristics of solids such as a solid's interior, boundary, complement, closure, boundedness and regularity. Requicha [4] insisted that to be computationally useful, a representation should formally capture the following properties ;

- ***Rigidity:*** Representations should have an invariant configuration irrespective of their location and orientation.
- ***Homogeneity:*** A representation should have an interior.
- ***Finiteness:*** A representation must occupy a finite amount of space.
- ***Boundary determinism:*** A representation must unambiguously determine the interior of that solid.

- ***Closure***: Representations of solids which are manipulated by rigid motions and regularized boolean operations should produce representations of solids too.

These formal characteristics leave representations no choice than to be bounded, closed, regularized and semianalytic, hence their coinage rsets according to [5]. An ***r-set*** is simply a regular and bounded set in $\mathbb{E}^3$.

A representation scheme is simply a relation between physical solids and their representations which can be characterized by the following properties;

- ***Domain***: A representation scheme must represent quite a number of useful geometric solids.
- ***Unambiguity***: A representation scheme should produce representations which intuitively capture the properties of the physical solid so that it can be easily distinguished from other representations.
- ***Uniqueness***: A representation scheme should uniquely represent a solid object within a software's database.
- ***Validity***: Representation schemes should yield representations of solids which do not exist or are valid.
- ***Closure***: A Representation scheme which transforms (reflects, scales, rotates) a representation should yield other representations too.
- ***Compactness***: Representation schemes should yield representations which save space and allow efficient algorithms to determine desirable physical characteristics.

## 2.2 Wireframe Modeling

For rectilinear objects whose edges are straight lines and whose faces are planar, the ordered pair of vertices $\mathbf{V} \in \mathbb{E}^3$ and edges $\mathbf{E} \in \mathbb{E}^3$ denoted by $(\mathbf{V}, \mathbf{E})$ is the object's wireframe. In a practical sense, it is the skeleton of an object wherein joints are vertices and bones are edges. In [6], a six step algorithm to generate an object's wireframe was developed wherein an object's wireframe was represented by a vertex table and an edge table. Although the work in [6] had drawbacks such as not checking the validity of input data, wireframe modeling has always provided designers with a chance to experiment with the final result of their models through sketching and it is frequently used to preview complex models. However, the use of only edge information left wireframe models ambiguous on rectilinear polyhedra talk less of topological ones. Figure 2.1 below shows the wireframe of a sphere in greyscale.

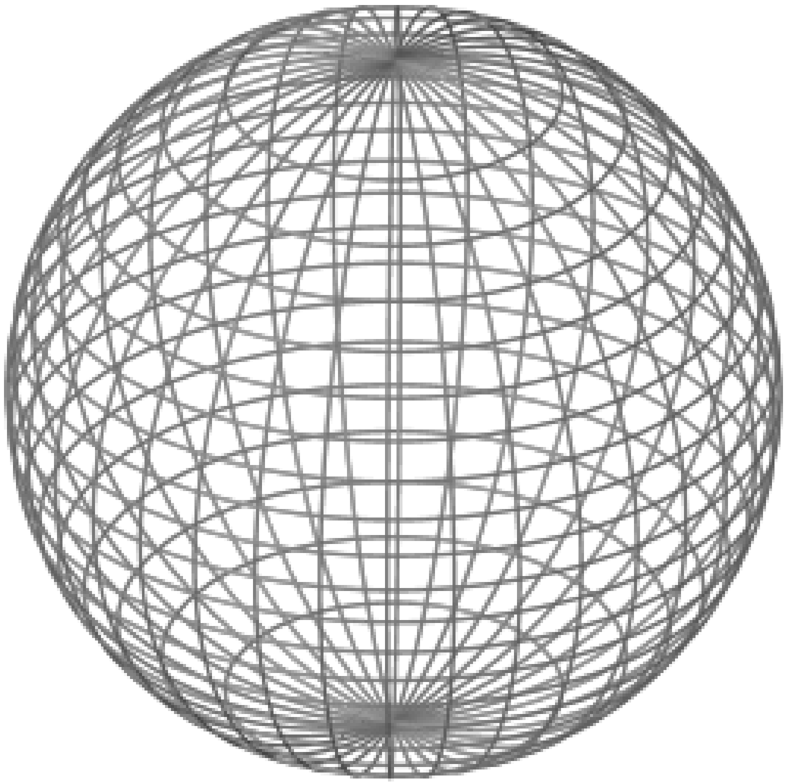

FIGURE 2.1: A wireframe of a sphere

## 2.3 Surface Modeling

After breakthroughs in wireframe modeling, research efforts in geometric modeling were directed towards extending the geometric coverage of CAD packages by incorporating complex freeform surfaces and curves. In this section, we emphasize on algebraic surfaces and curves used within BRL-CAD as it is the basis for Bezier surfaces and NURBS.

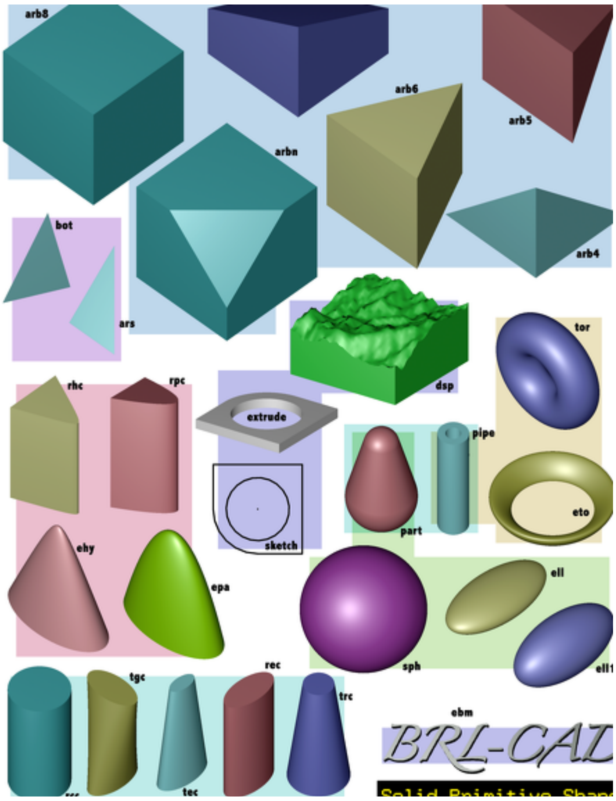


FIGURE 2.2: BRLCAD Solid Primitive Shapes

Figure 2.2 above shows a collection of some primitives used within the BRL-CAD package before the heart-shaped primitive was developed – several of which are implicitly and/or parameterically represented by algebraic equations.On the BRL-CAD's ideas page[7], there is a list of primitives which have not yet been implemented such as the Steiner surface, the ring cyclide surface, the quartoid, Wallis' conical edge solid, etc.

### 2.3.1 Implicit Representation

An algebraic surface in $\mathbb{E}^3$ is expressed as the set of points satisfying an irreducible polynomial equation

$$\mathbf{g(x,y,z) = 0}$$

in the unknowns x,y and z.
A polynomial ***f(x,y,z)*** over a field **F** is said to be irreducible over **F** if the degree of ***f(x,y,z)*** is positive and its only factors are $c$ and *cf(x,y,z)* where $c$ is a nonzero constant in **F**.

The requirement of irreducibility is so that a surface represented by an equation should not be decomposed into two separate surfaces, each of which can be described by an implicit equation.

### 2.3.2 Parametric Representation

Some algebraic surfaces possess a parametric representation which consists of a system of equations similar to the ones listed in (1) below;

$\mathrm{x} = \mathrm{h}_1(u, v)$
$y = h_2(u, v)$
$z = h_3(u, v)$
where $h_i$ are rational functions and, u and v are restricted to particular closed intervals in $\mathbb{R}$.

As an example, the unit sphere given implicitly by $x^2 + y^2 + z^2 – 1 = 0$ can be parameterized by equation (2) viz

$\mathrm{x} = (1 - \mathrm{s}^2 - t^2)/(1 + s^2 + t^2)$
$y = 2s/(1 + s^2 + t^2)$
$z = 2t/(1 + s^2 + t^2)$

Also, some algebraic curves possess parametric forms. A parameterization of the unit circle is given by the system of equations in (3) below;

$$\mathrm{x} = (1 - \mathrm{t}^2)/(1 + t^2)$$
$$y = 2t/(1 + t^2)$$

When the parametric representation is employed, it is easier to generate points on an algebraic surface or curve as compared to the implicit representation. Also, parametric equations are useful for interactive design since changes in their polynomial coefficients alter the surface's shape in an intuitive manner.

Lots of geometric operations could become faster if both aforementioned representations are made available within CAD packages. Thus, the problem of how to convert from one representation to the other is of great practical importance.

# Appendix A

# Appendix Title Here

Write your Appendix content here.