LibLathe Documentation

Release 0.0.1

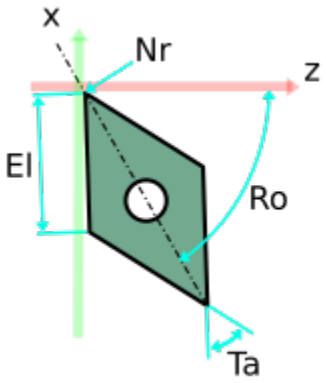
Daniel Wood

Aug 06, 2023

Contents

CHAPTER 1

LibLathe Tooling



Parameters:
- Edge Length (El)
- Nose Radius (Nr)
- Tip Angle (Ta)
- Rotation (Ro)

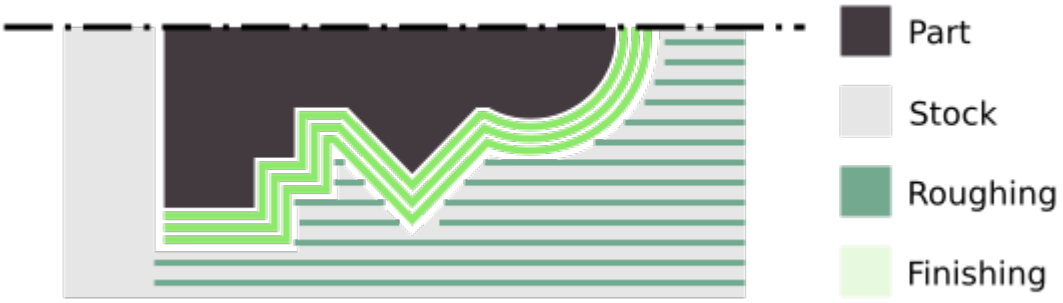
1.1 Turning inserts

LibLathe tooling are based around turning inserts for more information refer to turning_inserts.

CHAPTER 2

Geometry

LibLathe uses a simplified geometry representation which consists of a single type; the segment. This geometry is used throughout LibLathe to hold the part and path representations.



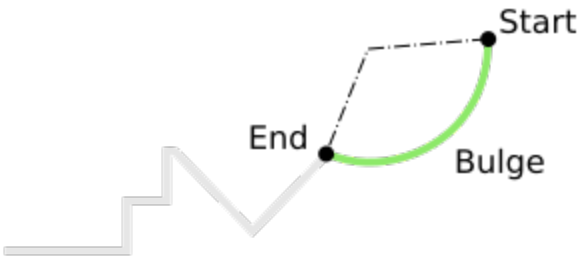
2.1 The Segment

The segment is the fundamental representation of LibLathe geometry. The segment can represent one of two basic forms; A line segment or an arc segment.

Grouping the geometry into a single type is possible as both lines and arcs can be defined by three components:

- Start Point
- End Point
- Bulge

This method of segment representation is inspired by a technical paper titled **An offset algorithm for polyline curves** by Xu-Zheng Liu et al, ISBN: 0166-3615



2.1.1 Line Segments

Line segments are represented by two points; a start point and an end point, the buldge value is always equal to zero for line segments.

2.1.2 Arc Segments

Arc segments are more complicated than line segments, however using the segment method arcs can be simplified to a three component representation. As with lines arcs also need start and end points, arcs however require a non-zero bulge value

refer to *Bulge* for more information.

2.2 The Segment Group

The segment group is a container object that holds a collection of segments, usually representing a 'pass'. The segment group can perform a set of operations on the segments such as:

- Offsetting
- Converting to gcode
- Analysis for use with a selected tool

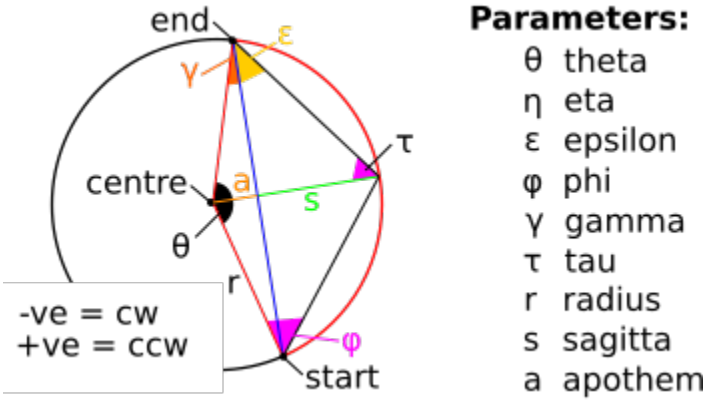
CHAPTER 3

Bulge

LibLathe uses a simplified geometry representation which consists of a single type; the segment. Segments consist of 3 bit of data:

- Start Point
- End Point
- Bulge

These 3 bits of data give Liblathe all the information required to calculate the position, size and direction of lines and arcs.



Bulge values are calculated:

```
bulge = tan(theta/4)
```

theta is the central arc angle between the start and end points.

Direction:

`bulge > 0 = CCW` and `bulge < 0 = CW`

Positive bulge values represent arcs with a counter clockwise direction
Negative bulge values represent arcs with a clockwise direction
Segments where the bulge is equal to zero represent a line

3.1 Parameters:

Name	Description
theta	included angle
eta	half included angle
epsilon	quarter included angle
phi	pi - epsilon
gamma	pi - eta
tau	see phi
radius	distance between start and end / 2
sagitta	arc height
apothem	radius - sagitta

CHAPTER 4

GCode

The fundamental purpose of LibLathe is to generate a program code or tool path commonly referred to as GCode.

GCode is typically a sequential list of commands that explain to the machine where to move the current tool in order to create the desired shape.

Common Lathe GCodes	
G0	Rapid Motion
G1	Linear Motion
G2	Clockwise Arc
G3	Anti-Clockwise Arc
G18	Reference Plane XZ
G20	Inch Units
G21	Metric (mm) Units
G28	Return to Home Position
G32	Constant Lead Threading Cycle
G70	Canned Finishing Cycle
G71	Canned Roughing Cycle
G72	Canned Facing Cycle
G73	Canned Pattern Cycle
G74	Canned Peck Drilling
G75	Canned Grooving Cycle
G76	Canned Threading Cycle
G98	Feedrate/Minute
G99	Feedrate/Cycle

Canned Cycle: Canned cycles are preprogrammed operations that are created from variables. Canned cycles usually perform repeative tasks such as peck drilling or threading where a number of repeat passes are made.

Example GCode File:

CHAPTER 5

Lathe Operations

- Roughing
- Facing
- Profile
- Grooving
- Boring
- Chamfer
- Threading
- Drilling

CHAPTER 6

About LibLathe

LibLathe is an opensource python library for generating CNC machining paths and outputting GCode

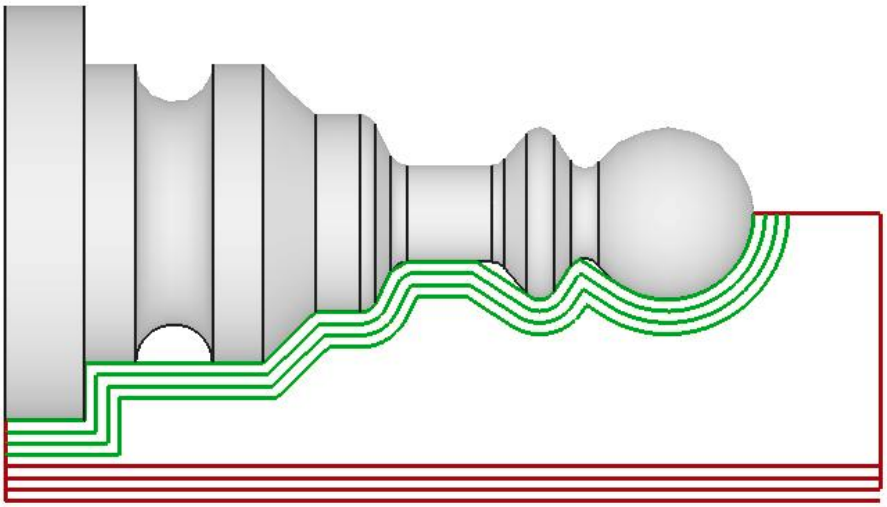


Fig. 1: LibLathe profile operation generated in FreeCAD

6.1 Getting LibLathe

LibLathe is opensource making it free to use, change and redistribute. LibLathe sourcecode can be found at GitHub