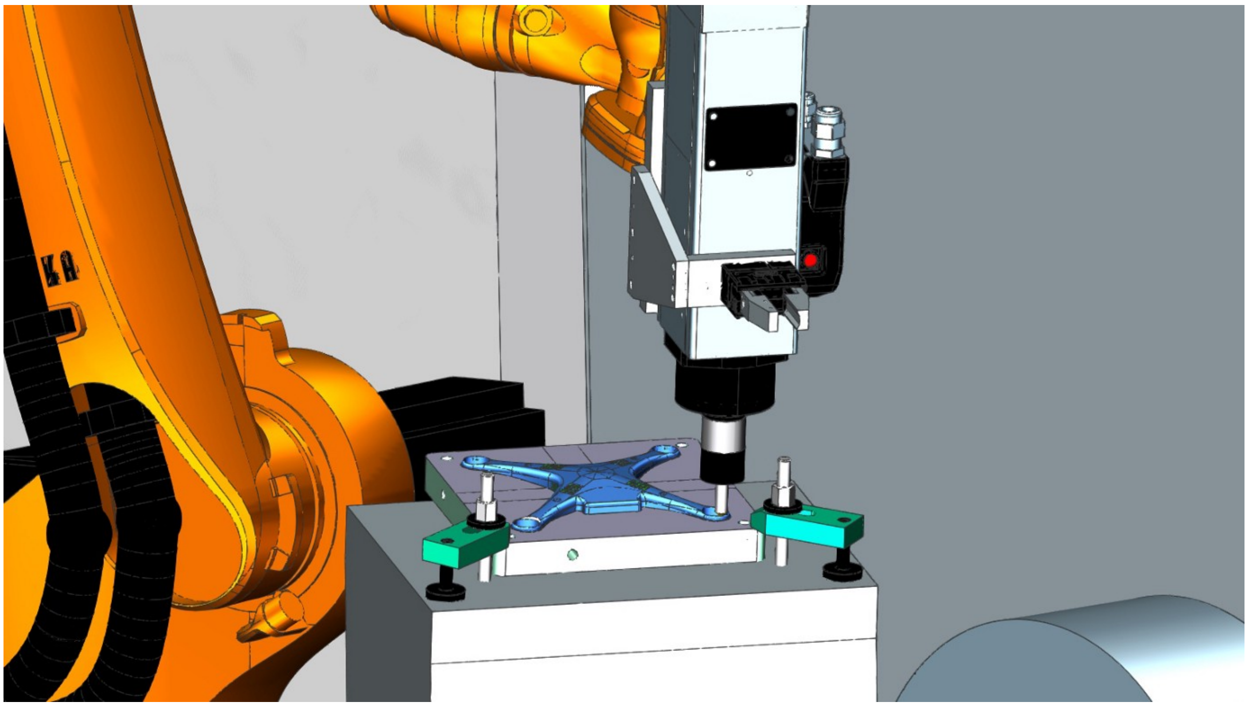
Towards a cloud-based CAM: A review of tools working with G-code

Nikita Letov ⋮⋮ 2021/7/13

Computer-Aided Manufacturing (CAM) software is the logical step after Computer-Aided Design (CAD): geometry of a CAD model (whether 2D or 3D) is provided as an input to CAM. The purpose of CAM is to 1) generate a Computer Numerical Control (CNC) script to be uploaded directly into a CNC machining tool; and to 2) simulate the actual manufacturing process. In other words, it is a necessary link that brings the digital twin of the designed part into the real world. This article attempts to review CAM tools that can potentionally be implemented on the web.

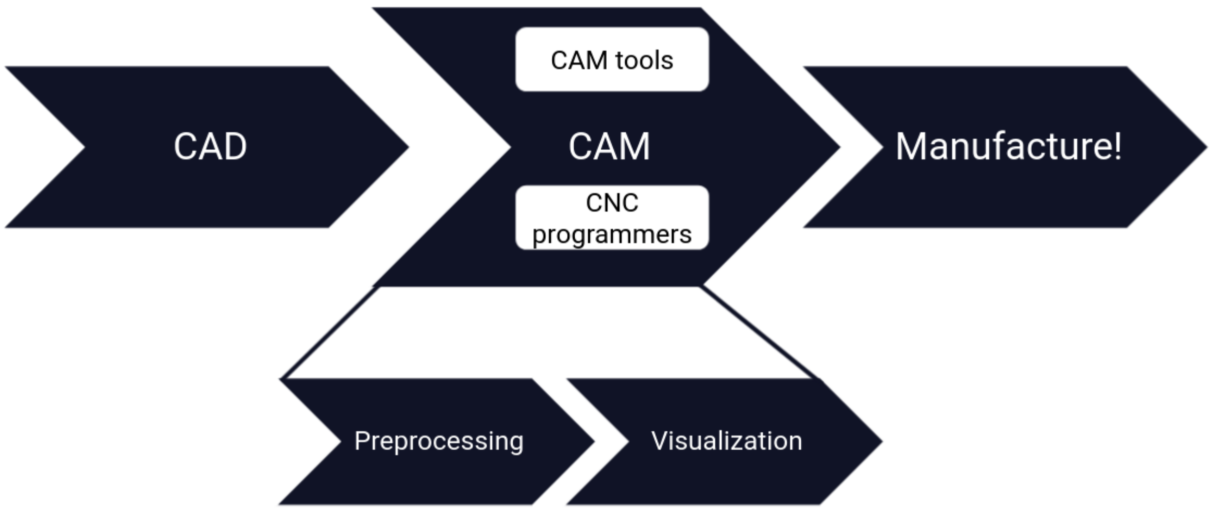
There are numerous proprietary CAM systems such as *Siemens NX for Manufacturing* in picture below. **However, this review focuses only on Free and Open-Source Software (FOSS).** Firstly, because propriatory CAMs are expensive (for business purposes even more) and it would make less sense paying for those while a manufacturing shop is still fresh and new. And secondly, because proprietory CAM tools are extremely hard to integrate in the web as they never were meant to be web-based. It also could make legal problems: people would be using *SolidWorks CAM* without paying for it, ***Dassault Systèmes*** — developers of ***SolidWorks*** — might not be very happy about it. FOSS solutions often loose to propitiatory analogs in terms of performance and quality, yet they are free and relatively quick to integrate in the web. Moreover, they are numerous, and that means that we are free to combine several of them into a single CAM taking only the best out of them.



Siemens NX for Manufacturing // © Siemens Digital Industries Software

There are different CNC programming languages that are used in CNC machine tools and that CAM operate with. However, out of them all, **G-code is the most widely spread** (even though it is more than a half-century old). Thus, **this review focuses on FOSS CAM operating with G-code**. A while ago, when G-code has already been invented but CAM systems were not yet, CNC programmers were coding G-code scripts based on the drawings provided by the design branches of entreprises. Nowadays, CAM systems usually handle this task automatically.

It was decided to categorize CAM tools by their focus on (1) preprocessing tools and (2) visualization tools. Both categories are covered below with notable examples of the FOSS tools.



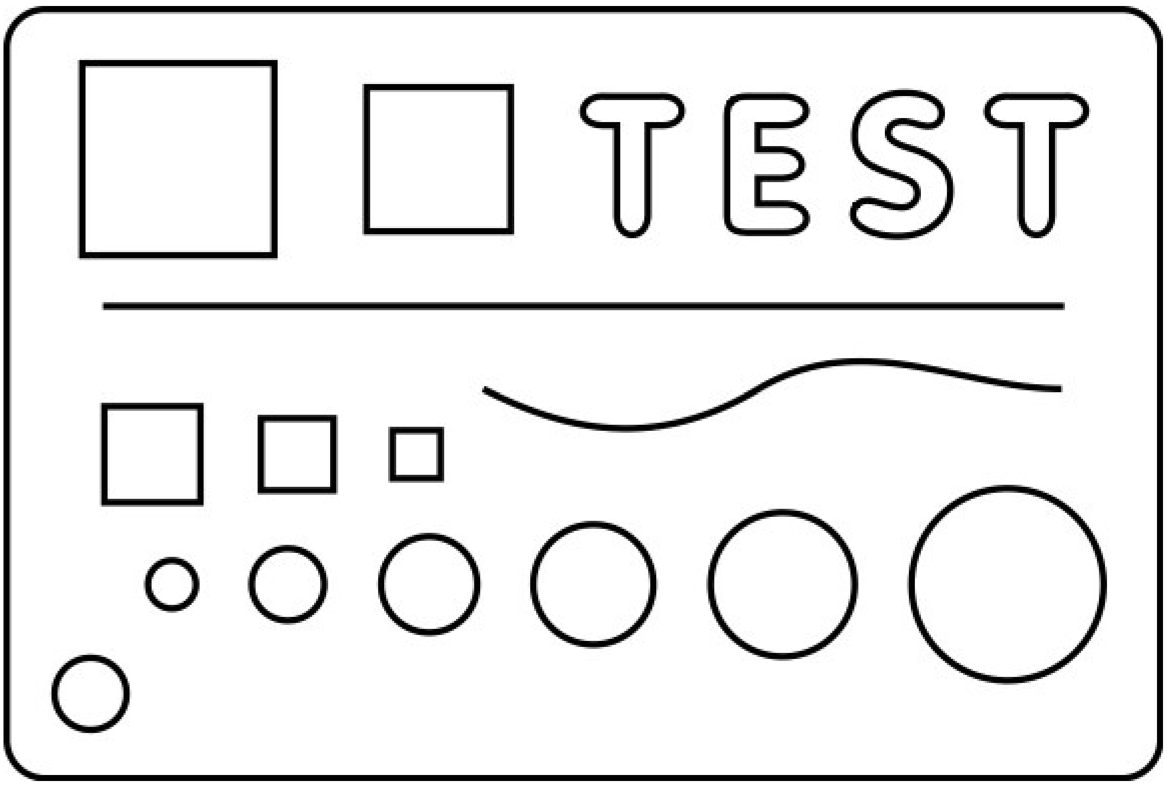
1. Preprocessing

Preprocessing is a necessary step before the visualization of the manufacturing process. Preprocessing allows G-code generation and preparation for use in a CNC machine. It is done either manually by a CNC programmer, or automatically by a CAM software in use at site (often with a sanity check by a CNC programmer).

1.1. DXF2GCODE

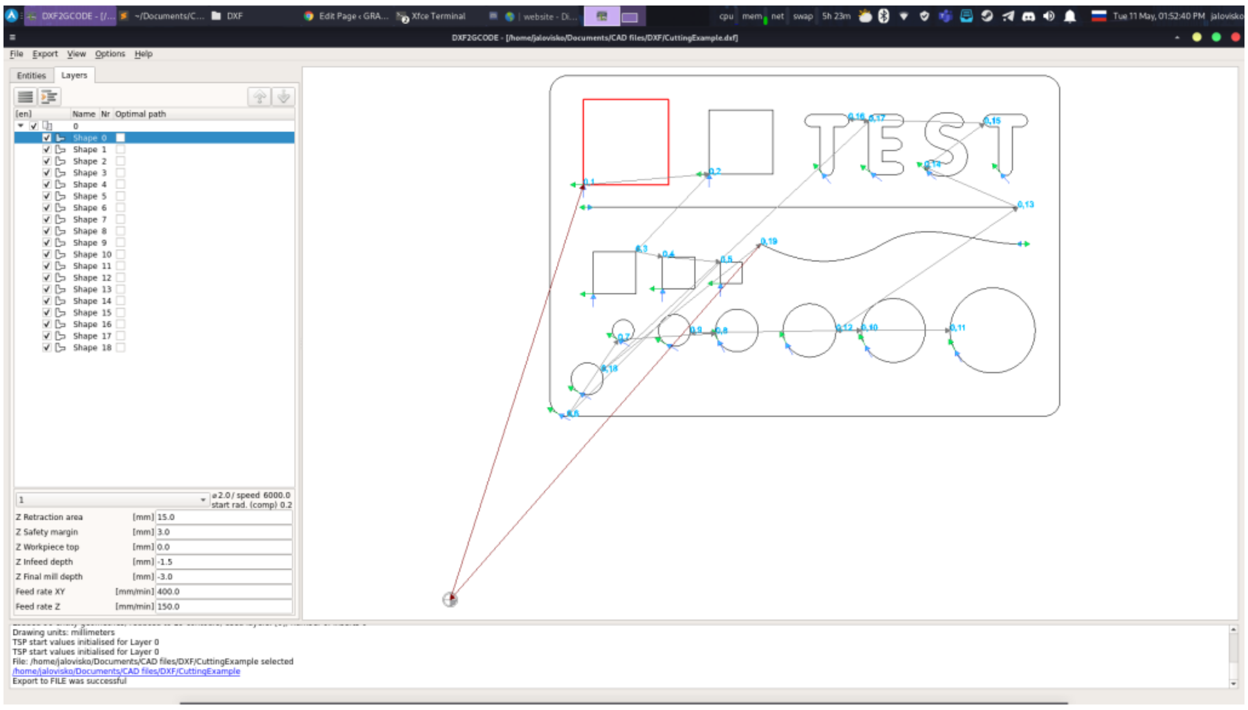
DXF2GCODE is a FOSS (GPLv3) 2D (dxf, pdf, ps) drawings to CNC machine compatible GCode.

Let us consider a DXF file illustrated in the picture below for example.



A preview of a DXF file used for testing // © CNC Plasma Info

DXF2GCODE allows convering this DXF file to a CNC-compatible G-code script. Moreover, it has a simplistic but yet present visualization as can be seen in the figure below (note that it is a GUI application).


DXF2GCODE - [/home/jalovisko/Documents/CAD files/DXF/CuttingExample.dxf]
File
Export
View
Options
Help
Entities
Layers
Name
Nr
Optimal path

```
1                                   ⌀2.0 / speed 6000.0
                                    start rad. (comp) 0.2
Z Retraction area        [mm]       15.0
Z Safety margin          [mm]       3.0
Z Workpiece top          [mm]       0.0
Z Infeed depth           [mm]       -1.5
Z Final mill depth       [mm]       -3.0
Feed rate XY             [mm/min]   400.0
Feed rate Z              [mm/min]   150.0
```

```
Drawing units: millimeters
TSP start values initialised for Layer 0
TSP start values initialised for Layer 0
File: /home/jalovisko/Documents/CAD files/DXF/CuttingExample selected
/home/jalovisko/Documents/CAD files/DXF/CuttingExample
Export to FILE was successful
```

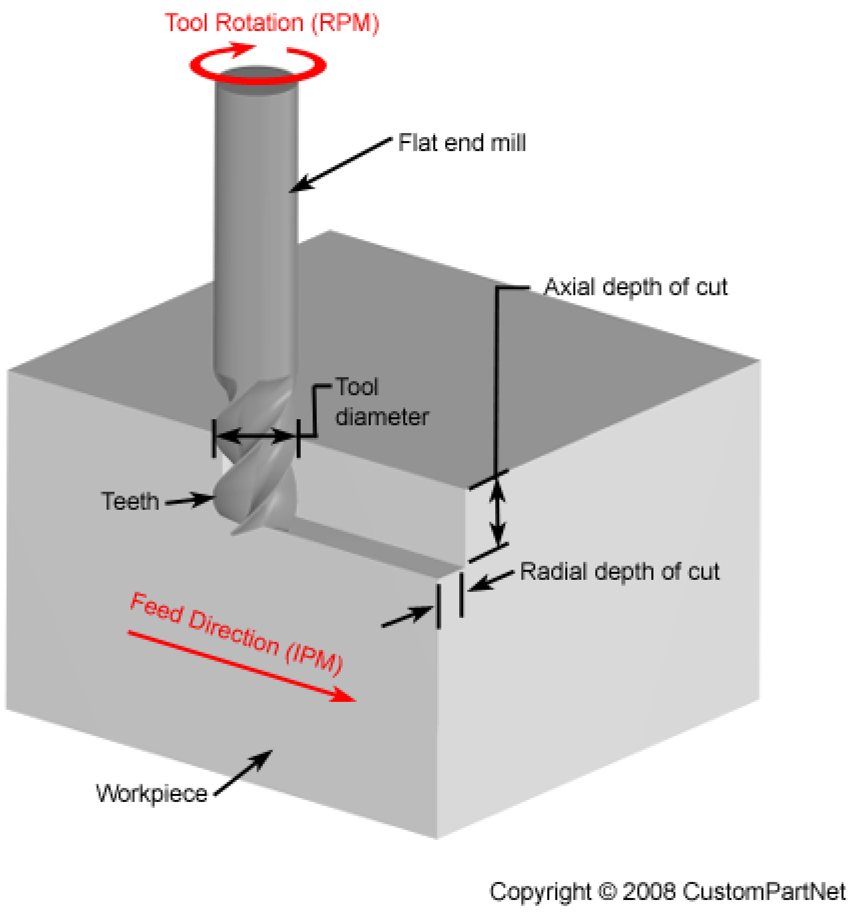
A view on the *DXF2GCODE* GUI

The generated G-code of, for example, the red square in the image above looks as follows and, in a nutshell, is just a set of coordinates that sets the path of the machine tool:

```
(* SHAPE Nr: 0 *)
G0 X  39.824 Y 123.775
M3 M8
G0 Z   3.000
F150
G1 Z  -1.500
F400
G1 X  39.824 Y 149.174
G1 X  65.224 Y 149.174
G1 X  65.224 Y 123.775
G1 X  39.824 Y 123.775
F150
G1 Z  -3.000
F400
G1 X  39.824 Y 149.174
G1 X  65.224 Y 149.174
G1 X  65.224 Y 123.775
G1 X  39.824 Y 123.775
F150
G1 Z   3.000
G0 Z  15.000
M9 M5
```

However, this CNC machining path does not take into the account that the machining happens in 3D. This is fine for laser cutting and other CNC cutting processes not requiring the tool moving vertically. However, it does not fit well milling and other similar CNC cutting processes because of that.

Another thing to notice lies in manual setting of the CNC process parameters such as the milling diameter (2mm in the example) and rotational speed
(6 000 RPM).

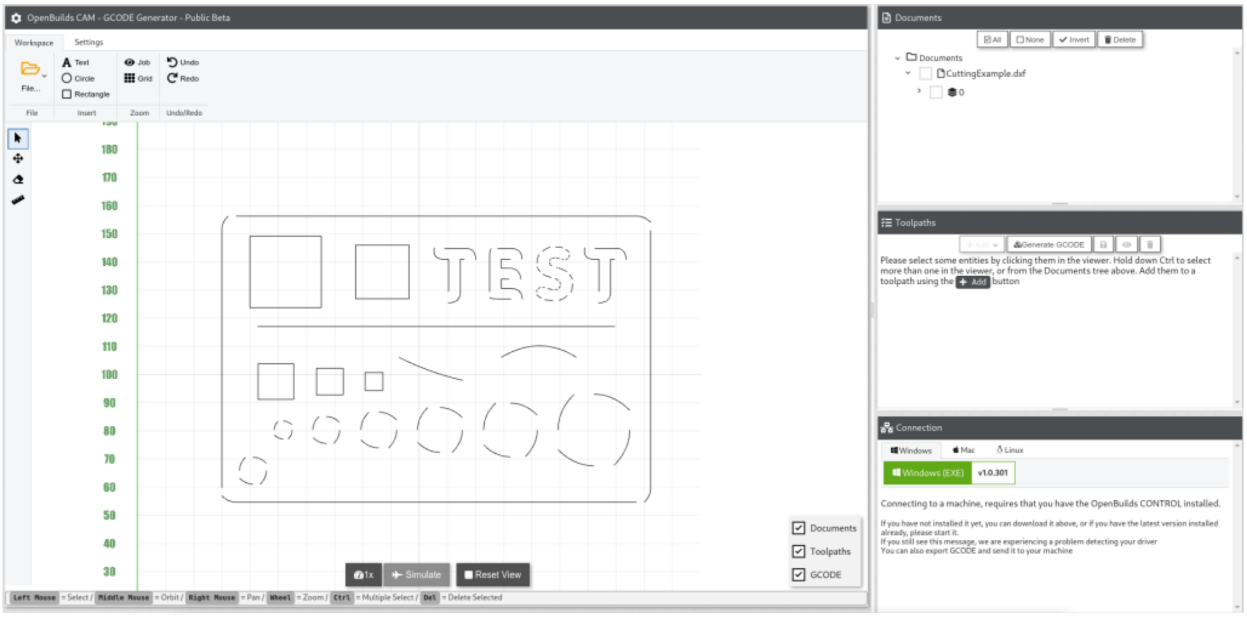


Some of the milling parameters // © Manufacturing Cost Estimation

1.2. OpenBuilds CAM

OpenBuilds CAM provides functionality similar to *DXF2GCODE* but on the web which is a major difference. Also, it is licenced under AGPLv3 which can be troublesome to be used in the web applications in some cases.

However, for the same sample DXF file from above, it fails to recognize some features and thus the generated G-code is not completely correct. The reason lies in failure to recognize DXF features other than lines and splines such as arcs. This issue is quite common for open-source CAD tools such as *Dia* for 2D CAD drawings, for example.



Please select some entities by clicking them in the viewer. Hold down Ctrl to select more than one in the viewer, or from the Documents tree above. Add them to a toolpath using the Add button

Connecting to a machine, requires that you have the OpenBuilds CONTROL installed.

If you have not installed it yet, you can download it above, or if you have the latest version installed already, please start it.
If you still see this message, we are experiencing a problem detecting your driver
You can also export GCODE and send it to your machine

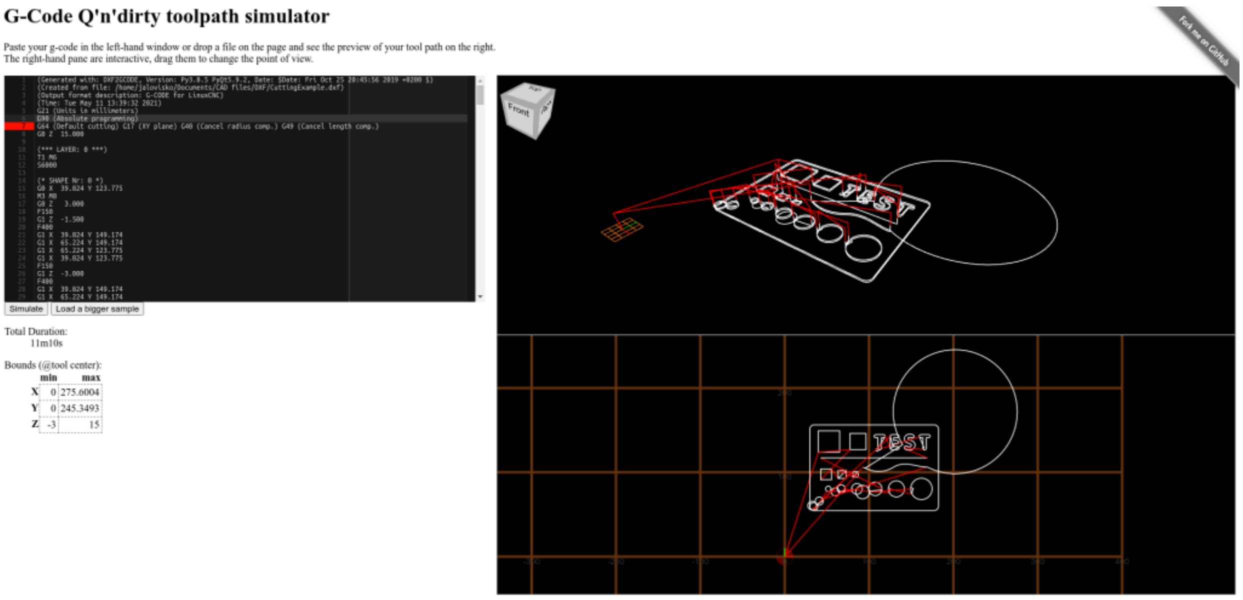
OpenBuilds CAM main window

2. Visualization

Visualization in CAM is done by means of simulation of the resulted G-code being run by a CNC machine tool. It serves as a check of whether the G-code script actually makes sense and moves exactly where it is supposed to.

2.1. Webgcode

Webgcode distributed under both MIT and AGPL licencses is a front-end tool for viewing G-code which, unlike *DXF2GCODE*, takes 3D into the account. For example, the G-code generated by *DXF2GCODE* in section 1.1 passes the instructions to a CNC machine which should move as follows according to the output of *Webgcode*:



G-Code Q'n'dirty toolpath simulator

Paste your g-code in the left-hand window or drop a file on the page and see the preview of your tool path on the right.
The right-hand pane are interactive, drag them to change the point of view.

```
(Generated with: DXF2GCODE, Version: Py3.8.5 PyQt5.9.2, Date: $Date: Fri Oct 25 20:45:56 2019 +0200 $)
(Created from file: /home/jalovisko/Documents/CAD files/DXF/CuttingExample.dxf)
(Output format description: G-CODE for LinuxCNC)
(Time: Tue May 11 13:39:32 2021)
G21 (Units in millimeters)
G90 (Absolute programming)
G64 (Default cutting) G17 (XY plane) G40 (Cancel radius comp.) G49 (Cancel length comp.)
G0 Z  15.000

(*** LAYER: 0 ***)
T1 M6
S6000

(* SHAPE Nr: 0 *)
G0 X  39.824 Y 123.775
M3 M8
G0 Z   3.000
F150
G1 Z  -1.500
F400
G1 X  39.824 Y 149.174
G1 X  65.224 Y 149.174
G1 X  65.224 Y 123.775
G1 X  39.824 Y 123.775
F150
G1 Z  -3.000
F400
G1 X  39.824 Y 149.174
G1 X  65.224 Y 149.174
```

Total Duration:
11m10s

Bounds (@tool center):

	min	max
X	0	275.6004
Y	0	245.3493
Z	-3	15

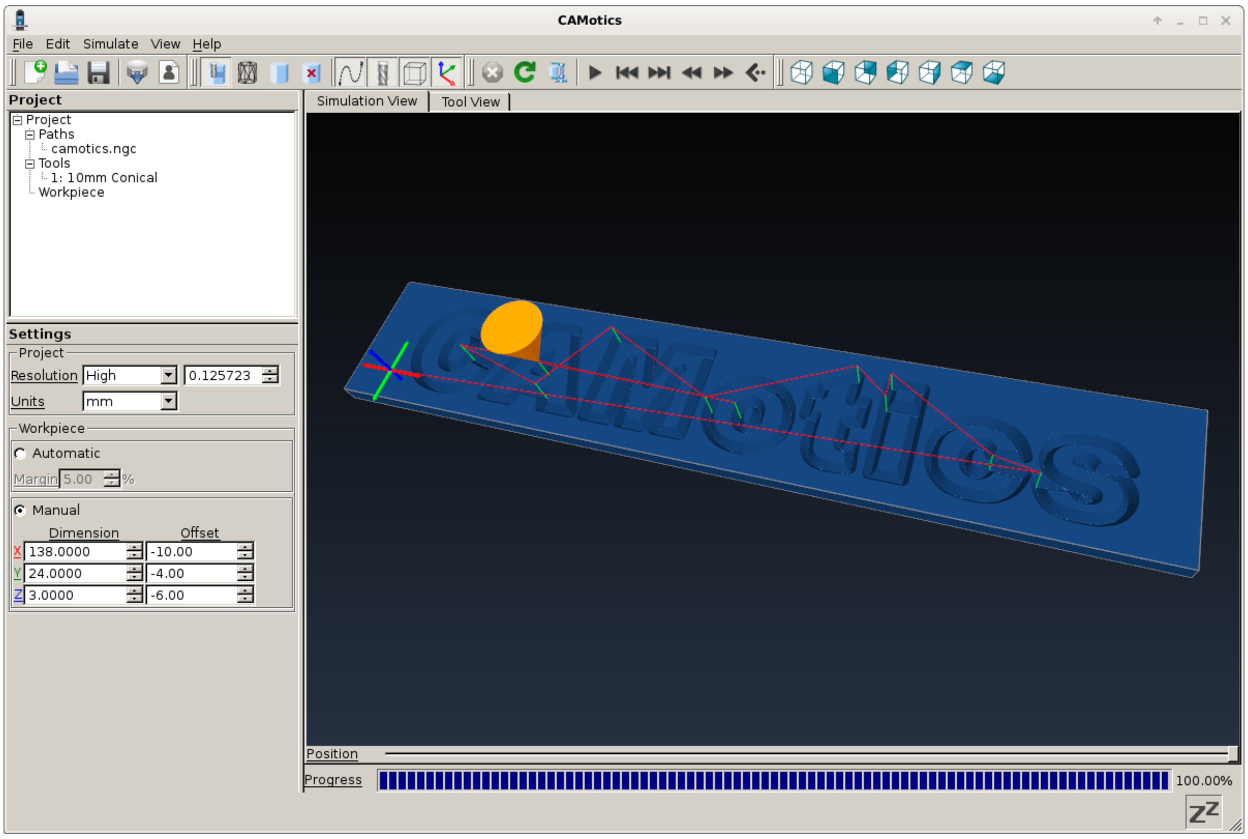
Webgcode applied to G-code generated by DXF2GCODE

Note that itis also estimates the total duration for manufacturing which sounds like a useful feature for a manufacturing (in manufacturing, it is a common practice to estimate costs in minutes of a machine tool runtime). It also calculates bounds which is also useful for suppliers, as it allows understanding if their CNC machine can produce a part of the given size.

It was however noticed that some G-code commands were not recognized by Webgcode, mostly because there are numerous commands which are specific to some CNC machines.

2.2. CAMotics

CAMotics is a GPLv2 licensed CAM visualization tool which, however, is tricky to run on Linux and is mainly supposed to be run on Windows (the Windows version seems to work nicely as I tested). It has nice features though: a 3D model of the resulting part is shown together with the paths required to produce it. Moreover, the cutting tool is customizable.



Settings

Project

Resolution High 0.125723

Units mm

Workpiece

Automatic

Margin 5.00 %

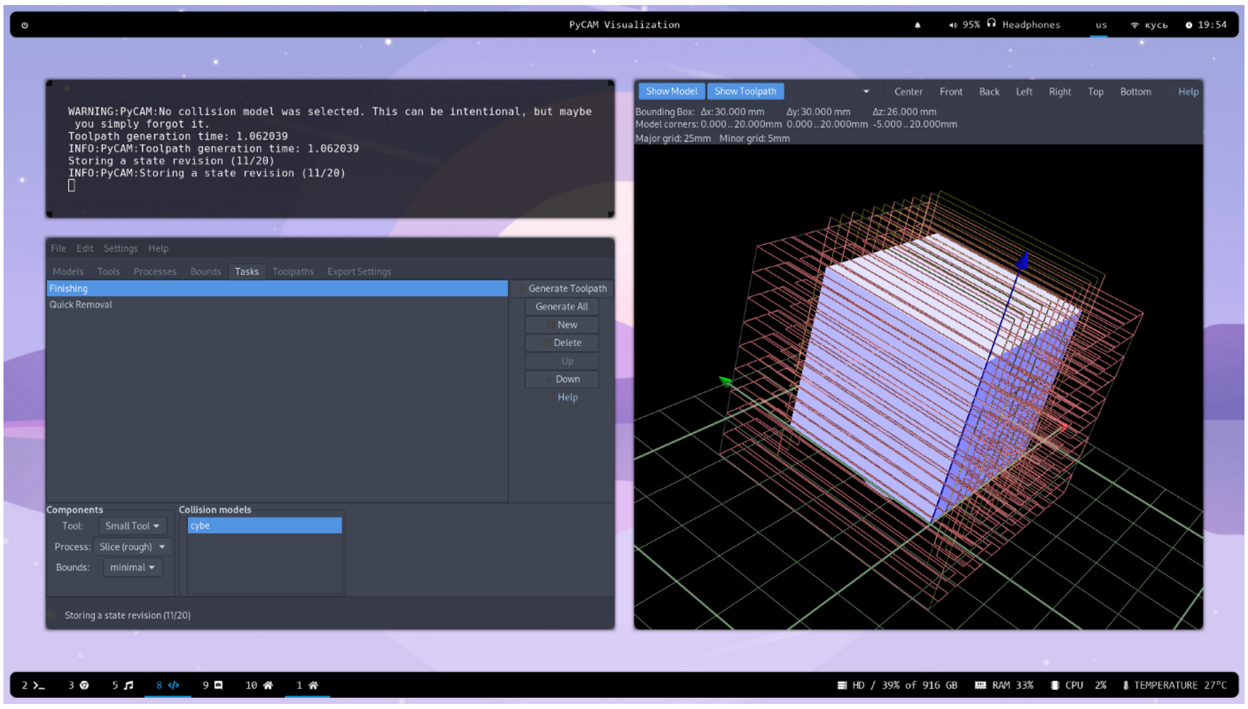
Manual

	Dimension	Offset
X	138.0000	-10.00
Y	24.0000	-4.00
Z	3.0000	-6.00

CAMotics main window

2.3. PyCAM

PyCAM is a GPLv3 licensed tool similar in its functionality to *CAMotics*. A neat diffirence however is having level-sets visualized in *PyCAM* which are seen as pink lines in the screenshot below. It also generates G-code based on SVG or DXF files uploaded. There are no versions of it for macOS and the Windows version is a decade old. It can also be ran as a CLI app, though quite a few parameters need to be configured for that. By the way, it generates G-code too! So PyCAM can actually be placed in section 1.3 of this review!



```
WARNING:PyCAM:No collision model was selected. This can be intentional, but maybe
 you simply forgot it.
Toolpath generation time: 1.062039
INFO:PyCAM:Toolpath generation time: 1.062039
Storing a state revision (11/20)
INFO:PyCAM:Storing a state revision (11/20)
```

Finishing
Quick Removal

Components
Tool: Small Tool
Process: Slice (rough)
Bounds: minimal

Collision models
cybe

Storing a state revision (11/20)

Bounding Box: Δx: 30.000 mm Δy: 30.000 mm Δz: 26.000 mm
Model corners: 0.000 .. 20.000mm 0.000 .. 20.000mm -5.000 .. 20.000mm
Major grid: 25mm Minor grid: 5mm

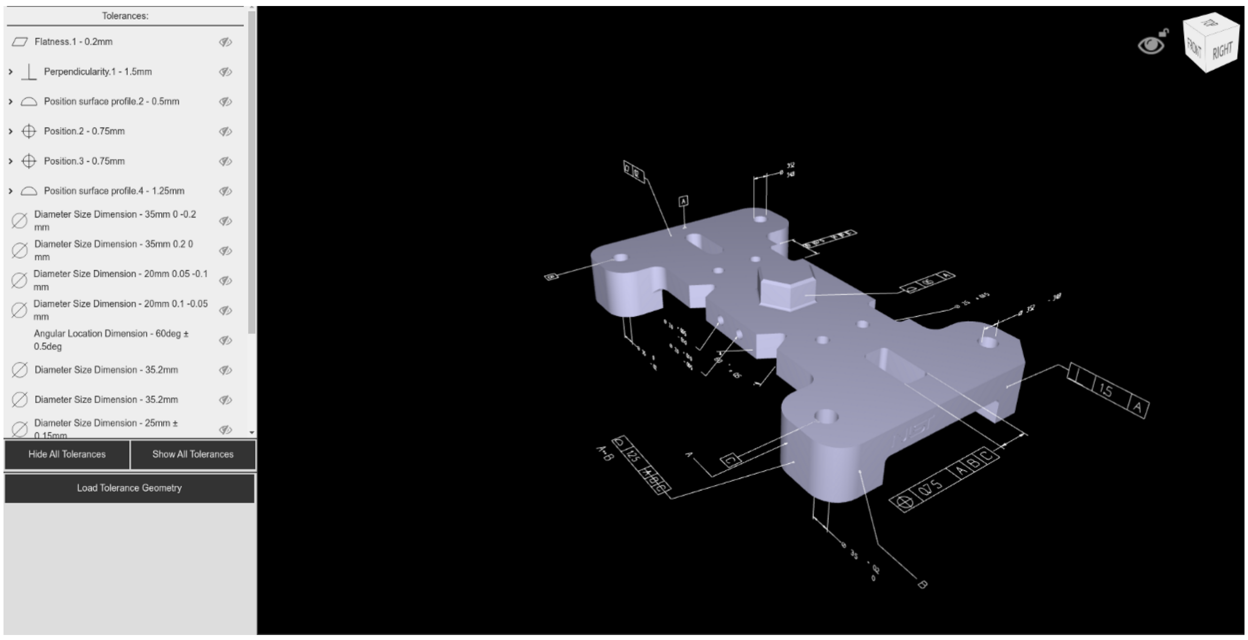
PyCAM interface

Licensing

Note that every tool has its own licensing and almost none of the tools reviewed have a permissive license. Thus, we must thread carefully the FOSS CAM soil. GPLv3 licensed tools such as *DXF2GCODE* and PyCAM cannot be used in the software distributed to the end users without distributing the whole source code. GPLv2 licensed tools such as *CAMotics* are similar, but the GPLv2 license is a bit more vague compared to GPLv3. For example, it is illegal to put GPLv3 components inside a program licensed under GPLv2. AGPL licensed tools such as *OpenBuilds CAM* are completely forbidden from being used even on the web, without sharing the whole source code. The permissive licenses such as MIT and Apache applied to *Webgcode* (which is dual-licensed with AGPL which is an interesting use-case), allow using the applicable software everywhere with close to none restrictions.

Further steps

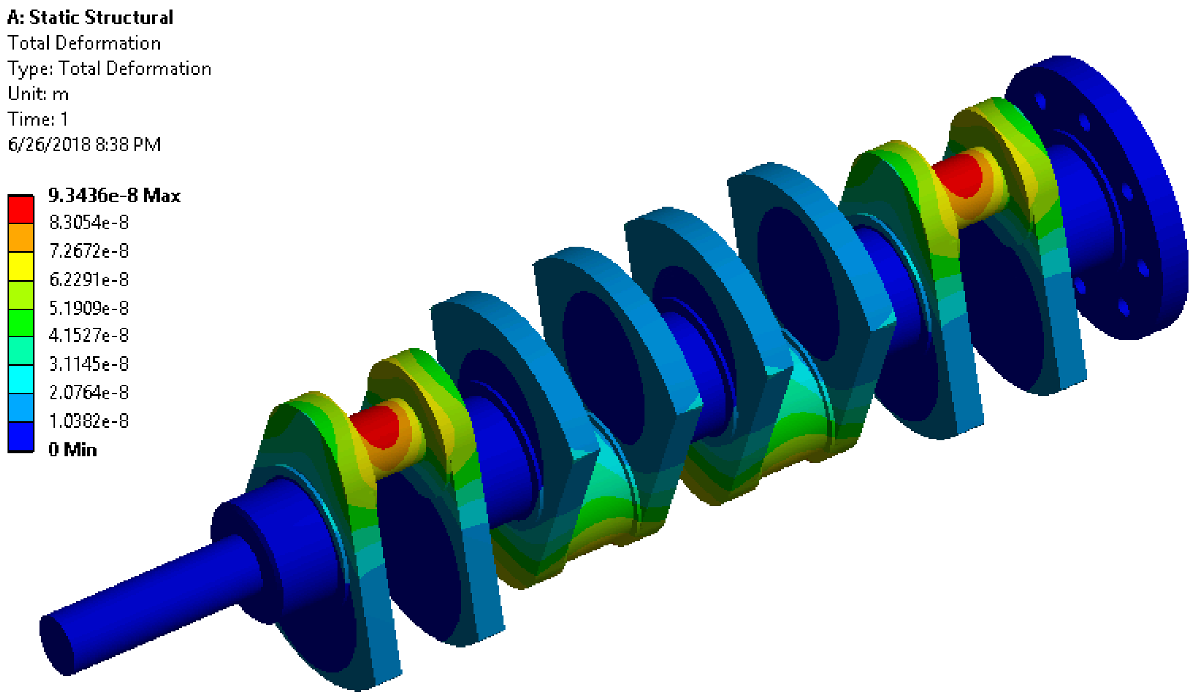
In this review we identified that all FOSS CAM tools have pros and cons. **The possible solution to mitigate the cons would be simply combining several reviewed tools into one.** For instance, *DXF2GCODE* can import a DXF file and output a G-code script, which in turn can be fed to *Webgcode*. Having tolerances and dimensions visible in 3D just like in *NC.js* can also aid with the manufacturing process, thus contributing more to a better cloud-based CAM system. Showing both the path of CNC machining and the 3D model of the result in the same viewer can provide useful visual feedback to CAM users.


Tolerances:
Flatness.1 - 0.2mm
Perpendicularity.1 - 1.5mm
Position surface profile.2 - 0.5mm
Position.2 - 0.75mm
Position.3 - 0.75mm
Position surface profile.4 - 1.25mm
Diameter Size Dimension - 35mm 0 -0.2 mm
Diameter Size Dimension - 35mm 0.2 0 mm
Diameter Size Dimension - 20mm 0.05 -0.1 mm
Diameter Size Dimension - 20mm 0.1 -0.05 mm
Angular Location Dimension - 60deg ± 0.5deg
Diameter Size Dimension - 35.2mm
Diameter Size Dimension - 35.2mm
Diameter Size Dimension - 25mm ± 0.15mm
Hide All Tolerances
Show All Tolerances
Load Tolerance Geometry
FRONT
RIGHT

An example of NC.js in use. You can play with this and other examples here

This review also **ignores CNC programmings languages other than G-code**, such as M-code and STEP-NC. This is done due to G-code having a wider spread in the manufacturing sector. However, STEP-NC is a rather modern CNC programming language that attempts to replace G-code and throw it from its pedestal of popularity. It would be interesting to see FOSS tools dealing with STEP-NC in the future!

As was mentioned in the begining, CAM is a logical continuation of CAD. However, before the actual manufactiring, a **Computer-Aided Engineering (CAE) simluation is needed which allows performing simulations on a 3D model**, for example, to see whether it breaks or not under certain conditions



ANSYS is a classical CAE system that allows visualizing loads distributed within a part which helps with identifying weak points // © Rescale Inc.

CAD/CAM/CAE are tools that form the fondation of any Product Lifecycle Management (PLM) system such as *Siemens Teamcenter*. Yet, **no PLM system has ever made it to the web**. PLM systems are often found at the core of possible routes reaching Industry 4.0 which is a rather hot topic in industry and academia.

Still, there are limitations in the world of open-source CAM. And they are mostly related to the general lack of open-source engineering solutions. *Open Cascade* remains the single most used open-source geometric modelling kernel, while there are many more proprietary ones with a much more well-developed functionality.

FreeCAD is a good example of what the open-source community can give to the engineering community. Yet, it still feels like a drop in the engineering ocean. But, hopefuly, this drop is the harbinger of rain!