

Die Nutzung von Matlab zur STL-Datenverarbeitung bis zur G-Code Maschinensteuerung

Prof. Dr. Tim Lüth
TU München

Grundlegendes zu Matlab

- Matlab ist eine prozedurale Programmiersprache ähnlich zu C/C++ mit der sehr komplexe Programme (u.a. objektorientiert) entwickelt und evaluiert und zertifiziert werden können.
- Die Basis-Einzelplatzlizenz (Mathworks) kostet 5000 € im Jahr. Der Code ist frei von Lizenzen.
- Die Softwareentwicklung erfolgt meist im Interpreter-Modus.
- Funktionen können als p-code vorcompiliert (beschleunigt) und gleichzeitig für Dritte unlesbar geschützt werden. Die Dokumentation bleibt jedoch auf Wunsch vollständig lesbar.
- Es können Stand-Alone-Applikationen für PC und MAC generiert werden.
- Selbst erstellte Funktionen können außerordentlich gut dokumentiert und evaluiert werden. Der Code ist wiederverwendbar.
- Eigene und Matlab-Funktionen können als C-Quellcode exportiert werden.
- Es existieren umfangreiche mathematische Bibliotheken um komplexe Algorithmen auf vielen Gebieten (Computergrafik, Geometrie, Numerik, Listenverwaltung, strukturierte Datentypen usw.) umzusetzen.
- Es können auch Schnittstellen angesprochen werden und echtzeitfähige Steuerungen als embedded Systems auf ARM-Prozessoren (Raspberry Pie, Beagle Bone Black, usw.)
- Es gibt umfangreiche Funktionen für die Realisierung von mausbasierten Benutzeroberflächen einschließlich der Handhabung von gerendertem dreidimensionalen Körpern.

STL-Verarbeitung mit Matlab (SG-Lib)

- STL-Daten bestehen aus Punkten (Vertices) im Raum und Dreiecksflächen (Facets) zwischen diesen Punkten
- Matlab unterstützt direkt diese Datenstrukturen bei der grafischen Ausgabe als "Vertex List" und "Facet List".
- STL-Slices bestehen aus verschachtelten geschlossen Konturen
- Matlab unterstützt direkt diese Datenstrukturen als "Closed Polygon Lines"

```
A=SGui
```

```
A =
```

```
struct with fields:
```

```
VL: [7073x3 double]
```

```
FL: [14142x3 double]
```

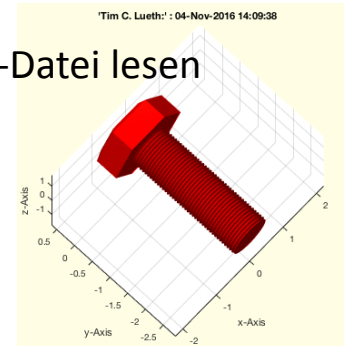
```
SGplot(A)
```

```
B=CPLofSGslice(A,0.01);
```

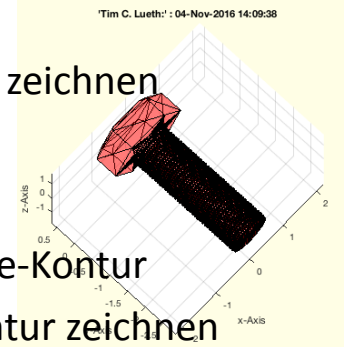
```
CPLplot(B);
```

```
CPLwriteSVG(B*20)
```

```
% STL-Datei lesen
```

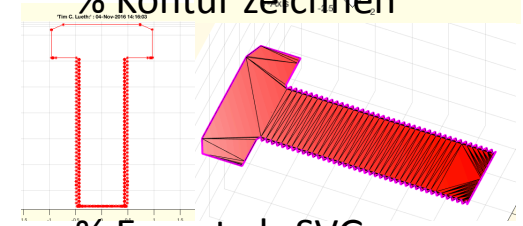


```
% STL zeichnen
```



```
% Slice-Kontur
```

```
% Kontur zeichnen
```



```
% Export als SVG
```

SG-Lib Beispielfunktion CPLarea(B)

```
function [ASUM,A]=CPLarea(CPL)

[PL,FL]=PLFLoFCPLdelaunay(CPL);
if nargin==0;
    VLFLarea(PL,FL); view(-30,30);
else
    [ASUM,A]=VLFLarea(PL,FL);
end
```

```
function [XPL,XFL,XEL]=PLFLoFCPLdelaunay(CPL,varargin)
ADDVL=[]; if nargin>=2; ADDVL=varargin{1}; end;
[PL,EL]=PLELoFCPL(CPL);
dtC=DeLaunayTri([PL;ADDVL],EL);
in=inOutStatus(dtC);
XPL=dtC.X;
XFL=dtC.Triangulation(in,:);
XEL=freeBoundary(dtC);

if nargin==0; SGfigure; VLFLplot(XPL,XFL); show; end
```