

Funcities in CNC Core 1.6b

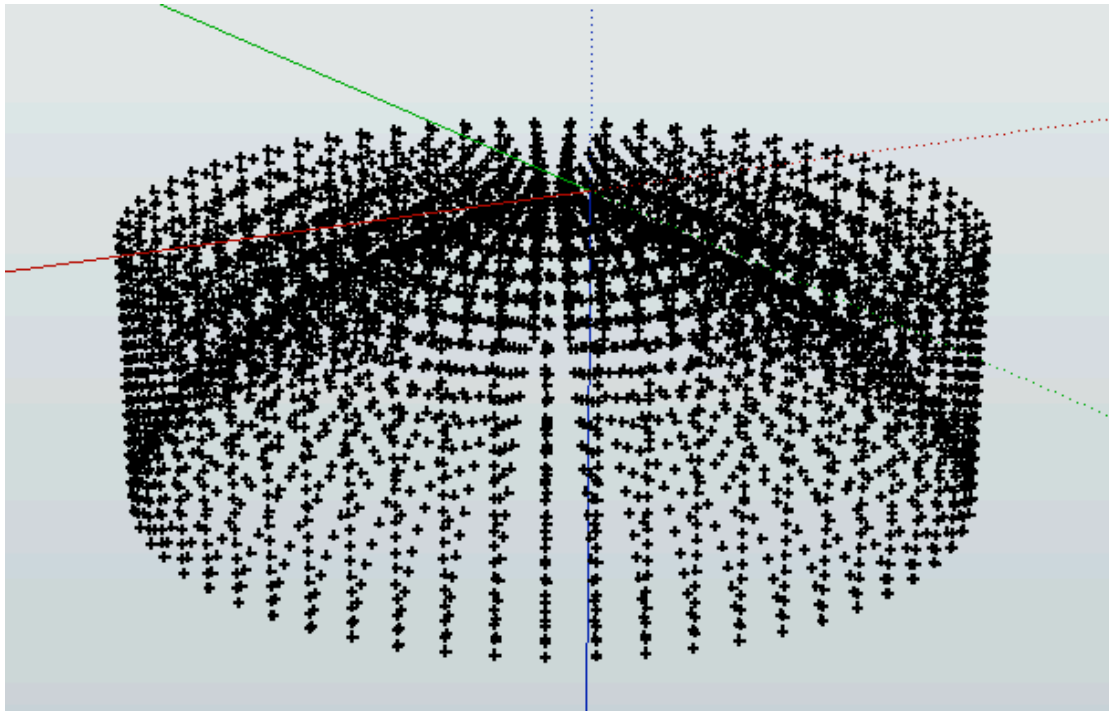
Een toelichting op de open source-functies voor 3D-frezen via Arduino. Hiermee kan gespeeld en gecombineerd worden, tot complexere beschrijvingen van mogelijke driedimensionale figuren.

Bij deze een hartelijke uitnodiging om deze functies te delen, samen met het programma en de aanwijzingen erbij, en om aanvullende functies te schrijven en te delen.

Rijk Willemse

www.rijkwillemse.nl/arduino.html

September 2013



Inhoud

1	Vooraf	2
2	Beschikbare functies	3
	Frees_box: een vierhoekig lichaam frezen	3
	Een driehoekig lichaam frezen	6
	Een cilindervormig lichaam frezen	7
	Een kegelvormig lichaam frezen	8
	Een piramidevormig lichaam frezen	9
3	De basic functies	11

1 Vooraf

Hardware

- Arduino, met Sparkfun EasyDriver v4.4 per steppermotor
- Eigen, 'tinker'-mill, die wacht op verbetering en vervanging
- Zie Fritzing-schema op www.rijkwillemse.nl/arduino.html voor de pins
- Motoren: sm-428YG011-25, 1,8 (= 200 steps per revolutie)
- Verzet: M8 draadstaal voor transmissie naar de 'tafels'
- Controle: Not On-schakelaars als limit-/eindswitches

Libraries

- Stepperdrivers: www.airspayce.com/mikem/arduino/AccelStepper/
- Debouncing: www.pjrc.com/teensy/td_libs/Bounce.html

Testplots

- In Terminal van Mac OSX, zie www.gnuplot.info
- Google Sketchup met points cloud plugin, zie <http://rhin.crai.archi.fr>

Programma

Code te downloaden op www.rijkwillemse.nl/arduino.html, met dank aan www.arduino.cc en allen die hun ideeën deelden. Deel ook!

Ik deel deze programmatuur graag, maar neem geen verantwoordelijkheid voor de werking en eventuele gevolgen van de foutieve werking. Opmerkingen en suggesties zijn altijd welkom!

Copyright

Creative Commons van toepassing

Opmerkingen

3D-milling is een vak apart. Lees bijvoorbeeld het 'CNC Programming Handbook' van Peter Smid. Dat maakt duidelijk dat 3D-frezen iets anders is dan 3D-printing... Centraal staat het denken 'van boven naar beneden', wat soms veel vraagt van het ruimtelijke voorstellingsvermogen.

Rijk Willemse

www.rijkwillemse.nl/arduino.html

Arnhem, Nederland, september 2013

2 Beschikbare functies

CNC Core 1.6b bevat een aantal *basic* functies, zoals de functie voor het frezen van een lijn in een 3D-ruimte, de functies voor het frezen van een vierhoek, een cirkel, een ellips, een spiraal... Op basis van deze functies is in CNC Core 1.6b een aantal gebruikersfuncties geschreven, die voor uitbreiding en (stuur je suggesties op!) voor verbetering in aanmerking komen.

We bespreken deze '*higher level*'-functies hier kort naar hun werking en hun resultaat. Voor nadere informatie over de betreffende functies zij verwezen naar de code van CNC Core 1.6b, zoals deze beschikbaar is op www.rijkwillemse.nl/arduino.html

In de bespreking van de eerste functie (*frees_box*) wordt ook de handling besproken van de output van CNC Core: naar de testomgeving en naar de frees-machine. Deze uitleg wordt bij de bespreking van de overige figuren voor bekend aangenomen.

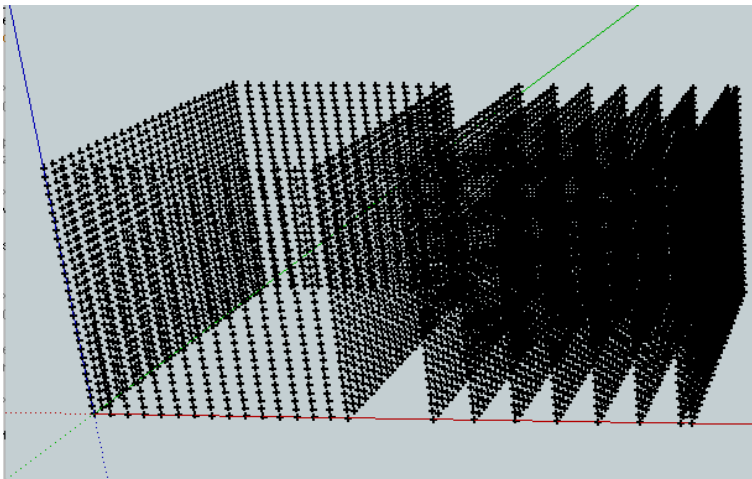
Frees_box: een vierhoekig lichaam frezen

In CNC Core 1.6b is een aantal mogelijkheden beschikbaar om een vierhoekig lichaam te frezen. De functie *frees_box* biedt twee mogelijkheden: 1) de omtrek van een vierhoekig lichaam, 2) de volledige uitfrezing van een vierhoekig lichaam. Beide functies worden hieronder achtereenvolgens in hun 'call' in CNC Core en hun grafische representatie van het resultaat met behulp van Sketchup weergegeven.

De calls

```
frees_box(0.0, 0.0, 0.0, BOL, 15.0, 25.0, 15.0);  
frees_box(20.0, 0.0, 0.0, HOL, 15.0, 25.0, 15.0);
```

De grafische representatie



Syntaxis

De syntaxis van de calls is als volgt. De eerste drie argumenten van de functie `frees_box` geven de x-, y- en z-coördinaten van het lichaam aan. Het volgende argument (HOL of BOL) geeft het ruimtelijke beslag van het lichaam weer. BOL staat hier voor de omtrek alleen, HOL staat voor het volledig uitfrezen van de inhoud van het lichaam.

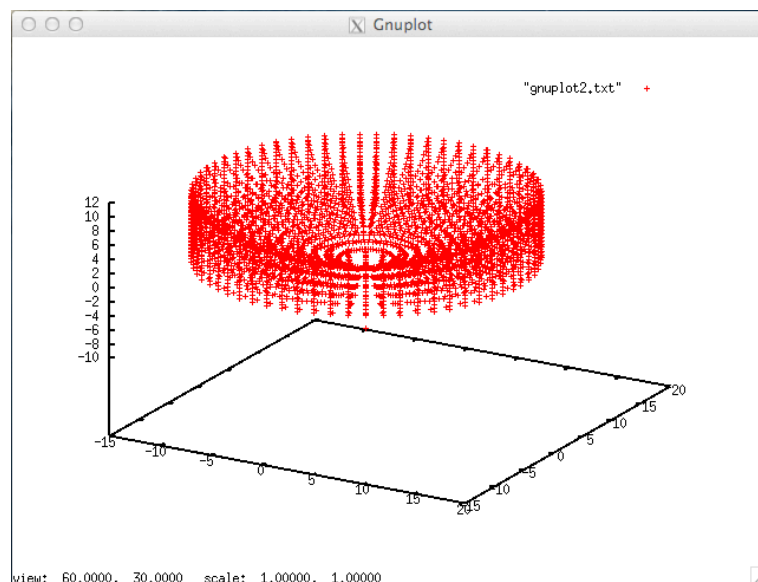
De laatste drie argumenten staan voor de lengte van de ribben van het lichaam, langs respectievelijk de x-, de y- en de z-as.

Toelichting

De calls zijn eenvoudig in het programma van CNC Core te doen, door ze te plaatsen in de functie `vector_P` van het programma. Met de globale variabele `GNUPLOT` op `true` of `false` (in het gebruikersgedeelte van het programma aan te passen) is output naar respectievelijk de serial monitor (voor preview met bijvoorbeeld `GNUPLOT` of `Sketchup`) en de frees-machine te bewerkstelligen.

De functie `vector_P` kan worden aangeroepen in de serial monitor (hit `CMD SHIFT M`) en daarna 'p' (van plot). Afhankelijk van de instellingen van de globale variabele `GNUPLOT` gaat de output van de functies die de gebruiker in deze functie plaatst naar de serial monitor (kopieer deze output van de serial monitor met knip en plak naar een bestand voor analyse) of naar de frees zelf.

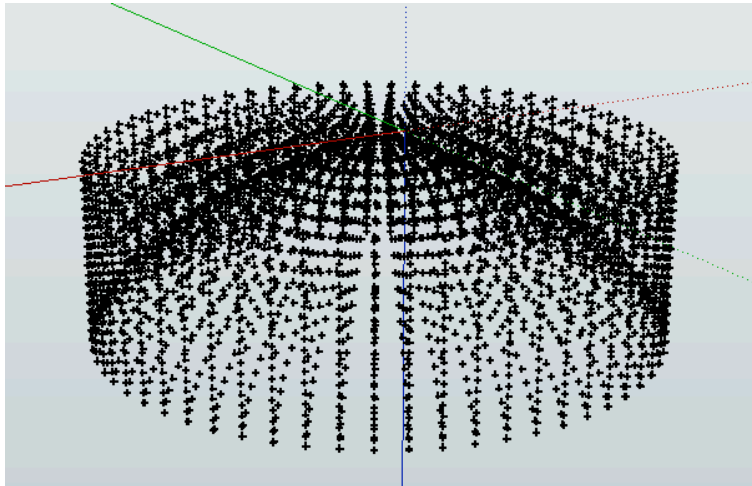
In de volgende afbeelding zie je een voorbeeld van een analyse van de functie die een bolvorm (zie verderop in deze in deze toelichting voor meer details over de corresponderende functie) uitfreest via `GNUPLOT`.



In de grafische afbeelding van de figuren vormt de z-as (de staande as) de ruimtelijke oriëntatie in de hoogte, waarbij in aanmerking moet worden genomen dat de hoogte op de z-as correspondeert met de diepte van de frees. Dus: hoe hoger op de z-as, hoe dieper de frees. In gedachten moet de grafische

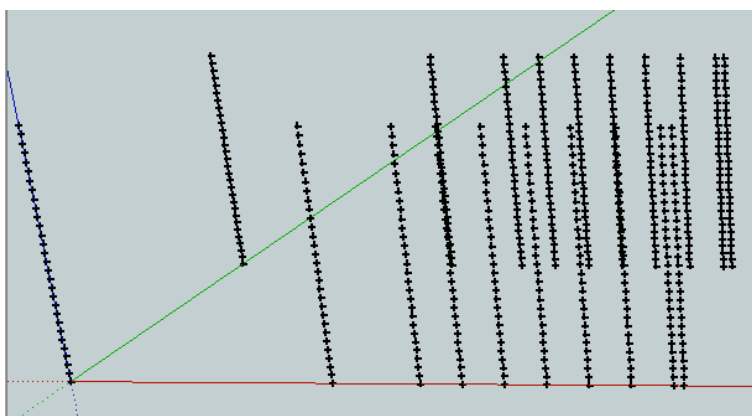
afbeelding dus om de x-as gespiegeld worden om het te frezen lichaam te beschouwen.

In de volgende afbeelding zie je een voorbeeld van de analyse van dezelfde bolvorm via Sketchup, met gebruikmaking van de plugin voor points clouds. (Deze figuur is gekanteld over de x-as, om de betreffende vorm te accentueren.)



De lijnen tussen de punten in de grafische representatie worden afgehandeld met de frees. Met de globale variabele GROF op *true* of *false* (in het gebruikersgedeelte van het programma aan te passen) wordt de output van het programma respectievelijk geschikt voor snelle(re) afhandeling met de frees en voor inzichtelijke aanschouwing via GNU PLOT of Sketchup.

Wanneer de instelling van de globale variabele GROF op *true* staat, is frezen passend. De frees kan dan in de grootst mogelijk halen van het ene punt naar het andere. Wanneer we de grafische representatie van deze instelling (GROF = *true*) bij de twee bovenstaande calls van de functie `frees_box` bijvoorbeeld naar de analysesoftware sturen (in dit geval Sketchup), krijgen we het volgende beeld.



Hierin zien we telkens de start- en eindpunten afgebeeld voor de frees die tussen beide punten reist. Beide bovenstaande figuren corresponderen met de grafische representatie van beide calls naar de functie `frees_box` die we hierboven gaven. Het mag duidelijk zijn dat er meer inzicht in het ruimtelijke beslag van de

betreffende functies ontstaat bij analyse met de globale variabele GROF op *false*, en dat frezen het liefst plaatsvindt onder GROF op *true*.

Een driehoekig lichaam frezen

In CNC Core 1.6b is een aantal mogelijkheden beschikbaar om een driehoekig lichaam te frezen. De functie *frees_tribox* biedt vier mogelijkheden: 1) een driehoekig lichaam in de breedte met de hoge zijde aan de rechterkant, 2) een driehoekig lichaam in de breedte met de hoge zijde aan de linkerkant, 3) een driehoekig lichaam in de lengte met de hoge zijde aan de rechterkant, en 4) een driehoekig lichaam in de lengte met de hoge zijde aan de linkerkant.

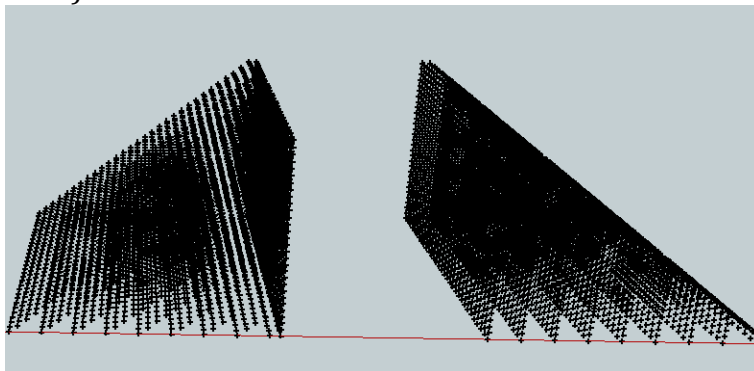
Deze functies worden hieronder achtereenvolgens in hun 'call' in CNC Core en hun grafische representatie van het resultaat met behulp van Sketchup weergegeven.

De calls

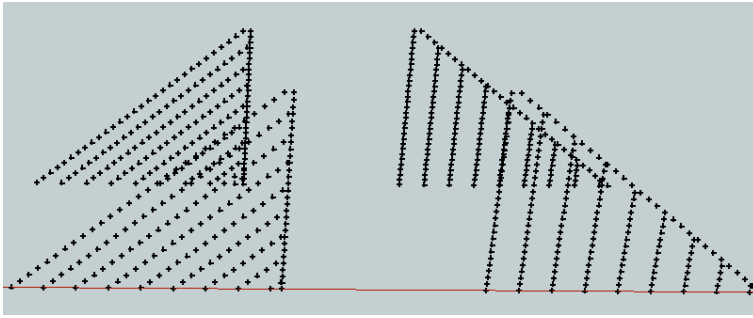
```
frees_tribox(40.0, 0.0, 0.0, RIGHT, OBLONG, 20.0, 30.0, 15.0);  
frees_tribox(75.0, 0.0, 0.0, LEFT, OBLONG, 20.0, 30.0, 15.0);
```

De grafische representatie

De onderstaande figuur is gemaakt met GROF op *false*, waarbij de stapjes op de te frezen lijnen goed zichtbaar zijn. (NB ook deze figuur gebruikt de z-as als aangeduid: deze lichamen worden naar beneden toe uitgefreesd van breed naar smal.)



De volgende figuur 'freest' dezelfde figuren uit, maar nu staat de globale variabele GROF op *true*. Deze figuren verbeelden dus de aansturing van de frees van de bovenstaande figuren.



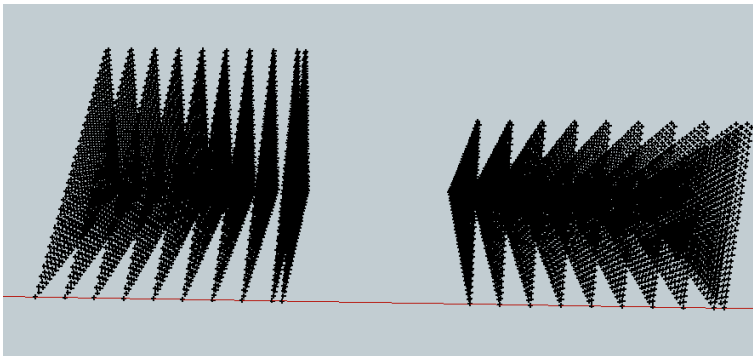
De syntaxis

De syntaxis van de calls is als volgt. De eerste drie argumenten van de functie `frees_box` geven de x-, y- en z-coördinaten van het lichaam aan. Het volgende argument (RIGHT of LEFT) geeft de plaats van de hoge zijde van het lichaam weer. OBLONG staat een afbeelding van het lichaam in de lengte langs de y-as.

De laatste drie argumenten staan voor de lengte van de ribben van het lichaam, langs respectievelijk de x-, de y- en de z-as.

In plaats van OBLONG kan deze figuur in beide versies ook in de lengte langs de x-as geprojecteerd worden. Kijk maar naar deze twee voorbeelden.

```
frees_tribox(110.0, 0.0, 0.0, RIGHT, LONG, 20.0, 30.0, 15.0);
frees_tribox(145.0, 0.0, 0.0, LEFT, LONG, 20.0, 30.0, 15.0);
```



Een cilindervormig lichaam frezen

In CNC Core 1.6b is een aantal mogelijkheden beschikbaar om een cilindervormig lichaam te frezen. De functie `frees_cilinder` biedt twee mogelijkheden: 1) de omtrek van een cilindervormig lichaam, 2) de volledige uitfrezing van een cilindervormig lichaam.

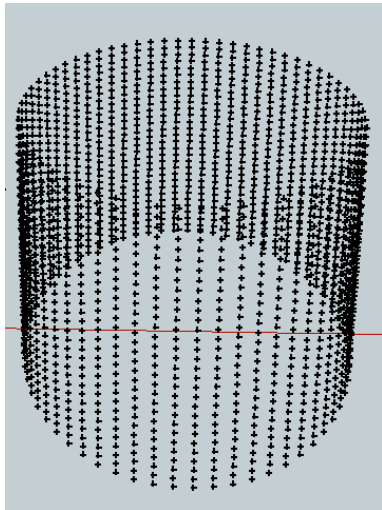
Beide functies worden hieronder achtereenvolgens in hun 'call' in CNC Core en hun grafische representatie van het resultaat met behulp van Sketchup weergegeven.

De calls

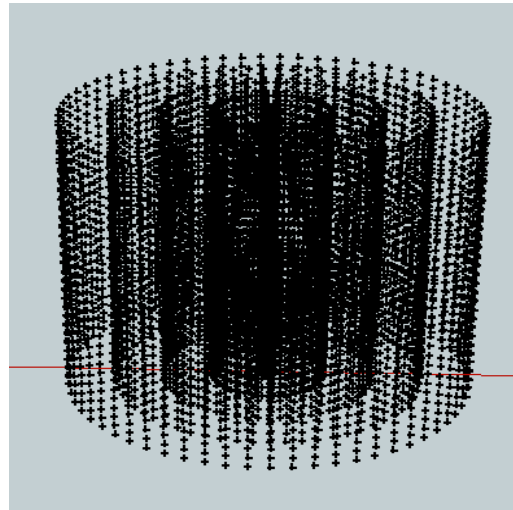
```
frees_cilinder(180.0, 0.0, 0.0, BOL, 10.0, 20.0, 15.0);
frees_cilinder(230.0, 0.0, 0.0, HOL, 10.0, 10.0, 15.0);
```

De grafische representatie

Omdat cirkelvormige figuren door CNC Core liefst zo precies mogelijk worden uitgefreesd, is er in het programma geen onderscheid tussen de twee waarden van de globale variabele GROF: zowel bij het frezen als bij het testen worden de kleinst mogelijke stapjes gezet, die overigens wel afhankelijk zijn van de globale variabele 'resolutie'.



BOL



HOL

De syntaxis

De syntaxis van de calls is als volgt. De eerste drie argumenten van de functie *frees_box* geven de x-, y- en z-coördinaten van het lichaam aan. Het volgende argument (HOL of BOL) geeft het ruimtelijke beslag van het lichaam weer. HOL staat voor de omtrek alleen, BOL staat voor het volledig uitfreesen van de inhoud van het lichaam.

De laatste twee argumenten staan voor de lengte van de straal van de cilinder zowel langs de x-as als langs de y-as bezien (wat elliptische vormen mogelijk maakt). Het laatste argument betreft de z-as: de hoogte van de cilinder.

Een kegelvormig lichaam frezen

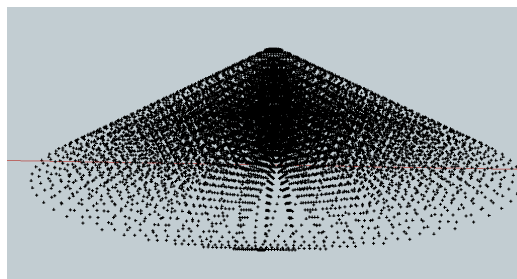
In CNC Core 1.6b is een aantal mogelijkheden beschikbaar om een kegelvormig lichaam te frezen. De functie *frees_kegel* biedt hiervoor vier mogelijkheden: 1) een kegelvormig lichaam (BOL, RECHT), 2) een koepelvormig lichaam (BOL, DOME), 3) een kegelvormig uitgesneden lichaam (HOL, RECHT), en 4) een koepelvormig uitgesneden lichaam (HOL, DOME).

De laatste van deze functies worden hieronder achtereenvolgens in hun 'call' in CNC Core en hun grafische representatie van het resultaat met behulp van Sketchup weergegeven. De eerste figuren zijn spiegelbeeldig hieraan, langs de x-as.

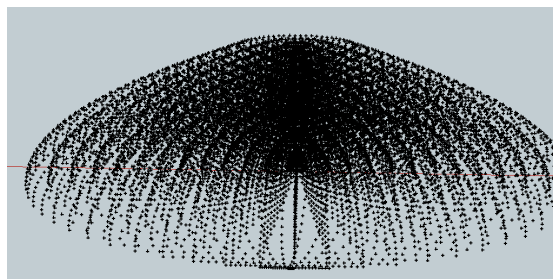
De calls

```
frees_kegel(300.0, 0.0, 0.0, HOL, RECHT, 30.0, 20.0, 15.0);  
frees_kegel(370.0, 0.0, 0.0, HOL, DOME, 30.0, 20.0, 15.0);
```

De grafische representatie



RECHT



DOME

De syntaxis

De syntaxis van de calls is als volgt. De eerste drie argumenten van de functie *frees_box* geven de x-, y- en z-coördinaten van het lichaam aan. Het volgende argument (HOL of BOL) geeft het ruimtelijke beslag van het lichaam weer. HOL staat in dit geval voor een uitgesneden lichaam, BOL staat voor een 'staand' lichaam, waarbij de omgeving volledig uitgefreesd wordt zodat de kegel overblijft.

De laatste twee argumenten staan voor de lengte van de straal van de cilinder zowel langs de x-as als langs de y-as bezien (wat elliptische vormen mogelijk maakt). Het laatste argument betreft de z-as: de hoogte van de kegel.

NB ook deze figuren gebruiken de z-as als aangeduid: deze lichamen worden naar beneden toe uitgefreesd van breed naar smal, van lage z, naar hogere z.

Een piramidevormig lichaam frezen

In CNC Core 1.6b is een aantal mogelijkheden beschikbaar om een piramidevormig lichaam te frezen. De functie *frees_piramide* biedt twee mogelijkheden: 1) een staande piramide (BOL), 2) een piramidevormige uitgesneden vorm (HOL).

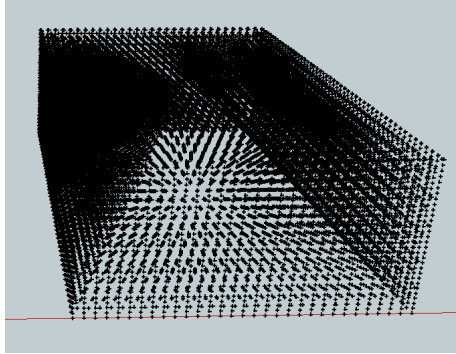
Beide functies worden hieronder achtereenvolgens in hun 'call' in CNC Core en hun grafische representatie van het resultaat met behulp van Sketchup weergegeven.

De calls

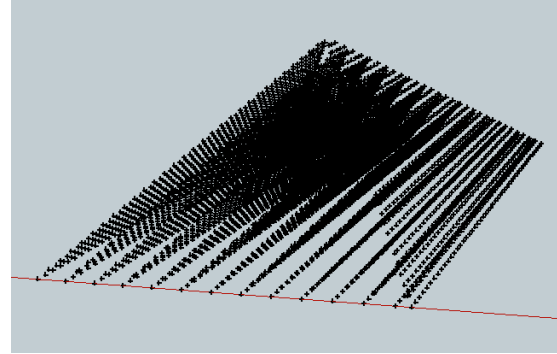
```
frees_piramide(440.0, 0.0, 0.0, BOL, 30.0, 50.0, 15.0);  
frees_piramide(510.0, 0.0, 0.0, HOL, 30.0, 50.0, 15.0);
```

De grafische representatie

Met ruime resolutie ($GROF = false$, voor preview)

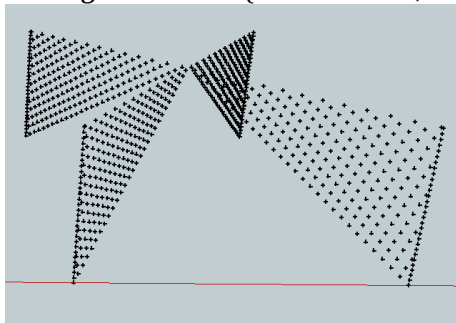


BOL

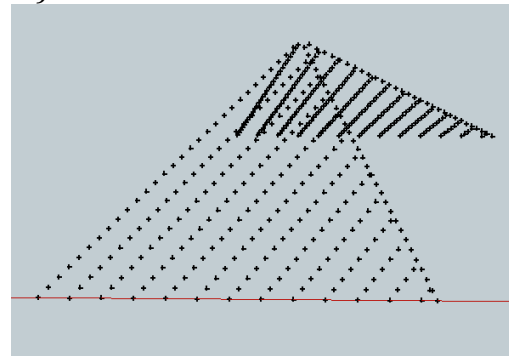


HOL

Met lage resolutie ($GROF = true$, voor de frees)



BOL



HOL

NB ook deze figuren gebruiken de z-as als aangeduid: deze lichamen worden naar beneden toe uitgefreesd van breed naar smal, van lage z, naar hogere z.

Syntaxis

De syntaxis van de calls is als volgt. De eerste drie argumenten van de functie `frees_box` geven de x-, y- en z-coördinaten van het lichaam aan. Het volgende argument (HOL of BOL) geeft het ruimtelijke beslag van het lichaam weer. HOL staat in dit geval voor een uitgesneden lichaam, BOL staat voor een 'staand' lichaam, waarbij de omgeving volledig uitgefreesd wordt zodat de piramide overblijft.

De laatste drie argumenten staan voor de lengte van de ribben van het lichaam, langs respectievelijk de x-, de y- en de z-as.

3 De basic functies

Voor een beter begrip van de gebruikersfuncties is een overzicht van de primitieve functies van CNC Core nuttig. Maar ook voor de gevorderde die wil bijdragen aan het functierepertoire is dit overzicht van belang. Neem met dat doel voor ogen gerust de code van CNC Core onder ogen. En, wanneer iemand mogelijke verbeteringen ziet in dit repertoire, die zijn altijd welkom!

Hieronder een summier overzicht van de vier basisfuncties van CNC Core, in de vorm van hun definitie in C.

```
frees_cirkel(float start_x, float start_y, float start_z,  
            float boog,  
            int spiraal,  
            float straal_x, float straal_y)
```

```
frees_vierhoek(float start_x, float start_y, float start_z,  
              int hol_of_bol,  
              float rib_x, float rib_y)
```

```
frees_lijn(float start_x, float start_y, float start_z,  
          float eind_x, float eind_y, float eind_z)
```

```
ga_naar(float x, float y, float z)
```