S3D Parameters that affect Extrusion amount
Scott Bozeman, Oregon State University, 2021

- E command in gcode is how many mm of wire to extrude – directly controls rate of material deposition

- **Filament diameter** – reduces extrusion amount by a factor of 2
- **Extrusion width** – linearly scales extrusion amount
  - Large extrusion width values create poor resolution
- **Layer height** – linearly scales extrusion amount
- **Nozzle diameter** – no effect
- **Extrusion multiplier** – linearly scales extrusion amount
- **Travel Distance** – linearly scales extrusion amount
- **Travel Speed** – no effect

```
G1 X67.500 Y113.500 E95.8842 F600
G1 X68.500 Y113.500 E1.6822
G1 X68.500 Y56.500 E95.8842
G1 X69.500 Y56.500 E1.6822
```

*Build plate moves 1 mm and extrudes 1.6822 mm of wire*
Filament Diameter

- Filament (wire) diameter directly affects how extrusion values are calculated
  - See next slide for eqn
- Should be 0.89 mm, not 0.87 mm, but minor difference
- Meltio uses default filament diameter and “corrects” with their extrusion multiplier
For this example gcode, $M_E = 1.0$, $w_E = 1.0$, $z = 1.0$, $x = 1.0$, $d_w = 0.87$ mm

$E \text{ (mm)} = \frac{M_E \cdot w_E \cdot z \cdot x}{\frac{\pi}{4} (d_w)^2}$

$1.6822 = \frac{1 \cdot 1 \cdot 1 \cdot 1}{\frac{\pi}{4} (0.87)^2}$

*This is all set in S3D – the true bead height and extrusion width will be different!!!

Build plate moves 1 mm and extrudes 1.6822 mm of wire
Correcting Extrusion Width & Layer Height

1. Print single tracks with a constant E value, varying scan speed and laser power
2. Measure extrusion (bead) width and layer (bead) height
3. Put correct values into slicer for a given speed and power combination. Adjust with the extrusion multiplier to get the same E values as before. Now your slicer should resemble / match the physical parameters of your wire.

\[ E (mm) = \frac{M_E \cdot w_E \cdot z \cdot x}{\frac{\pi}{4} (d_w)^2} \]

- Extrusion Multiplier
- Extrusion (bead) Width (mm)
- Layer (bead) Height (mm)
- Travel Distance (mm) (for that G1 line)
- Wire (Filament) Diameter (mm)

mm of wire extruded amount (E in gcode)
Energy Densities

- 2 ways to calculate energy density for wire-DED
  - Used for separate comparisons

1. Heat Input
   - Amount of energy per volume in a deposited line of material
   - Best for comparing single tracks

2. Bulk Energy Density
   - Amount of energy per volume in a bulk part
   - Only works for comparing builds with a constant feed rate
   - Best for dense, multilayered samples
   - Similar to PBF bulk energy density

\[
\rho_{E,PBF} = \frac{P_L}{v \cdot t \cdot h_s}
\]
Volumetric Feed Rate

\[ V \left( \frac{mm^3}{s} \right) = \frac{E \cdot \pi (d_w)^2 \cdot v}{x} \]

- mm of wire extruded amount (E in gcode)
- Wire (Filament) Diameter (mm)
- Travel Speed (mm/s) (F command in gcode)
- Travel Distance (mm) (for that G1 line)

\[ \rho_{E,PBF} = \frac{p_L}{v \cdot t \cdot h_s} \]

*This is all should be the same in S3D and reality if printing single tracks
Bead heights and widths will be different, but same amount of material extruded
1. Heat Inputs (Single track)

- Desirable to compare different parameter sets
  - Ex: Energy / power density in PBF
- We can do the same thing here:

\[
\rho_{E,PBF} = \frac{P_L}{v \cdot t \cdot h_s}
\]

\[
\rho_E = \frac{P_L}{V} = \frac{P_L \cdot x}{E \cdot \frac{\pi}{4} (d_w)^2 \cdot v} \quad (J/mm^3) \quad \text{– an energy density}
\]

- This is best for single tracks, doesn’t incorporate successive layers or for hatching
2. Bulk Energy Density

- Basically the same as in PBF, but with a constant feed rate (extrusion value).
- This is best used after single tracks have been optimized (parts are dense).
- Line spacing is related to extrusion width, also similar to hatch spacing in PBF.

\[
\rho_{E,DED} = \frac{P_L}{v \cdot t \cdot s}
\]

\[
\rho_{E,PBF} = \frac{P_L}{v \cdot t \cdot hs}
\]