Bed Mesh

The Bed Mesh module may be used to compensate for bed surface irregularities to achieve a better first layer across the entire bed. It should be noted that software based correction will not achieve perfect results, it can only approximate the shape of the bed. Bed Mesh also cannot compensate for mechanical and electrical issues. If an axis is skewed or a probe is not accurate then the bed_mesh module will not receive accurate results from the probing process.

Prior to Mesh Calibration you will need to be sure that your Probe's Z-Offset is calibrated. If using an endstop for Z homing it will need to be calibrated as well. See Probe Calibrate and Z_ENDSTOP_CALIBRATE in Manual Level for more information.

Basic Configuration

Rectangular Beds

This example assumes a printer with a 250 mm x 220 mm rectangular bed and a probe with an x-offset of 24 mm and y-offset of 5 mm.

```
[bed_mesh]
speed: 120
horizontal_move_z: 5
mesh_min: 35, 6
mesh_max: 240, 198
probe_count: 5, 3
```

- `speed: 120`
  Default Value: 50
  The speed in which the tool moves between points.

- `horizontal_move_z: 5`
  Default Value: 5
  The Z coordinate the probe rises to prior to traveling between points.

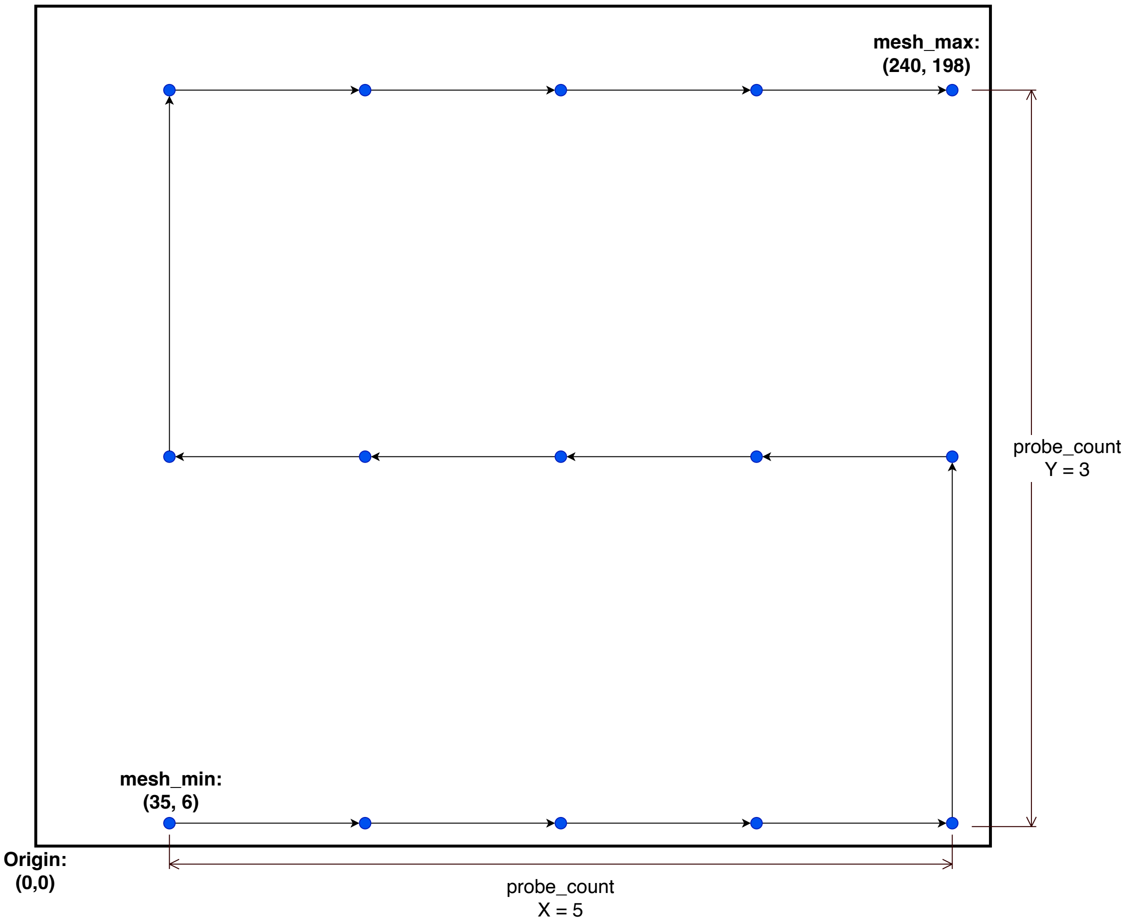
- `mesh_min: 35, 6`
  Required
  The first probed coordinate, nearest to the origin. This coordinate is relative to the probe's location.

- `mesh_max: 240, 198`
  Required

The probed coordinate farthest farthest from the origin. This is not necessarily the last point probed, as the probing process occurs in a zig-zag fashion. As with `mesh_min`, this coordiante is relative to the probe's location.

- `probe_count: 5, 3`
  Default Value: 3, 3
  The number of points to probe on each axis, specified as X, Y integer values. In this example 5 points will be probed along the X axis, with 3 points along the Y axis, for a total of 15 probed points. Note that if you wanted a square grid, for example 3x3, this could be specified as a single integer value that is used for both axes, ie `probe_count: 3`. Note that a mesh requires a minimum probe_count of 3 along each axis.

The illustration below demonstrates how the `mesh_min`, `mesh_max`, and `probe_count` options are used to generate probe points. The arrows indicate the direction of the probing procedure, beginning at `mesh_min`. For reference, when the probe is at `mesh_min` the nozzle will be at (11, 1), and when the probe is at `mesh_max`, the nozzle will be at (206, 193).



Round beds

This example assumes a printer equipped with a round bed radius of 100mm. We will use the same probe offsets as the rectangular example, 24 mm on X and 5 mm on Y.

```
[bed_mesh]
speed: 120
horizontal_move_z: 5
mesh_radius: 75
mesh_origin: 0, 0
round_probe_count: 5
```

- `mesh_radius: 75`

  Required

  The radius of the probed mesh in mm, relative to the `mesh_origin`. Note that the probe's offsets limit the size of the mesh radius. In this example, a radius larger than 76 would move the tool beyond the range of the printer.

- `mesh_origin: 0, 0`

  Default Value: 0, 0

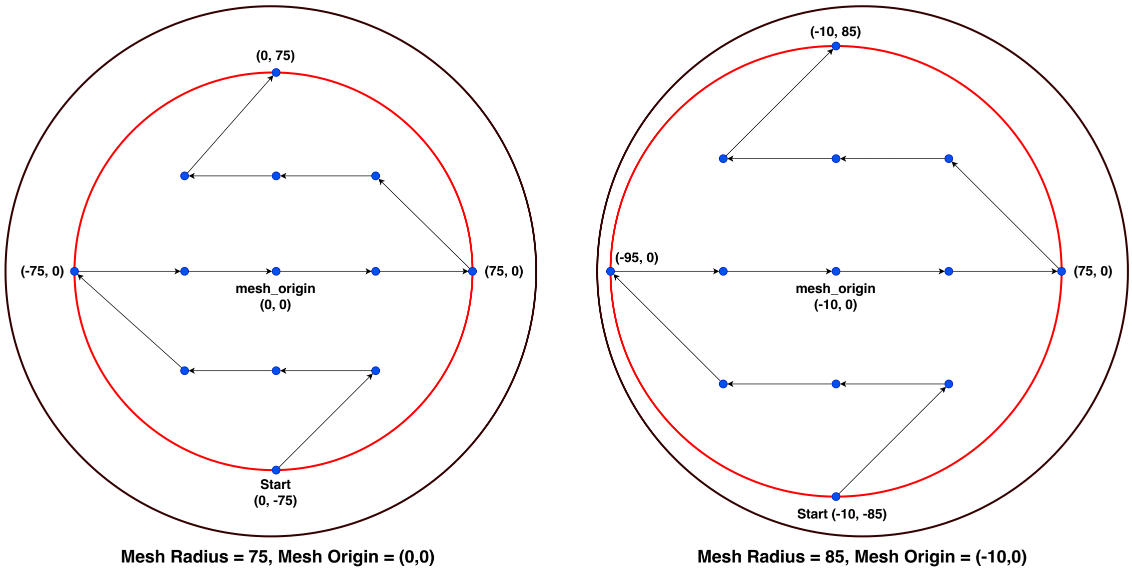
  The center point of the mesh. This coordinate is relative to the probe's location. While the default is 0, 0, it may be useful to adjust the origin in an effort to probe a larger portion of the bed. See the illustration below.

- `round_probe_count: 5`

  Default Value: 5

  This is an integer value that defines the maximum number of probed points along the X and Y axes. By "maximum", we mean the number of points probed along the mesh origin. This value must be an odd number, as it is required that the center of the mesh is probed.

The illustration below shows how the probed points are generated. As you can see, setting the `mesh_origin` to (-10, 0) allows us to specifiy a larger mesh radius of 85.



Advanced Configuration

Below the more advanced configuration options are explained in detail. Each example will build upon the basic rectangular bed configuration shown above. Each of the advanced options apply to round beds in the same manner.

Mesh Interpolation

While its possible to sample the probed matrix directly using simple bilinear interpolation to determine the Z-Values between probed points, it is often useful to interpolate extra points using more advanced interpolation algorithms to increase mesh density. These algorithms add curvature to the mesh, attempting to simulate the material properties of the bed. Bed Mesh offers lagrange and bicubic interpolation to accomplish this.

```
[bed_mesh]
speed: 120
horizontal_move_z: 5
mesh_min: 35, 6
mesh_max: 240, 198
probe_count: 5, 3
mesh_pps: 2, 3
algorithm: bicubic
bicubic_tension: 0.2
```

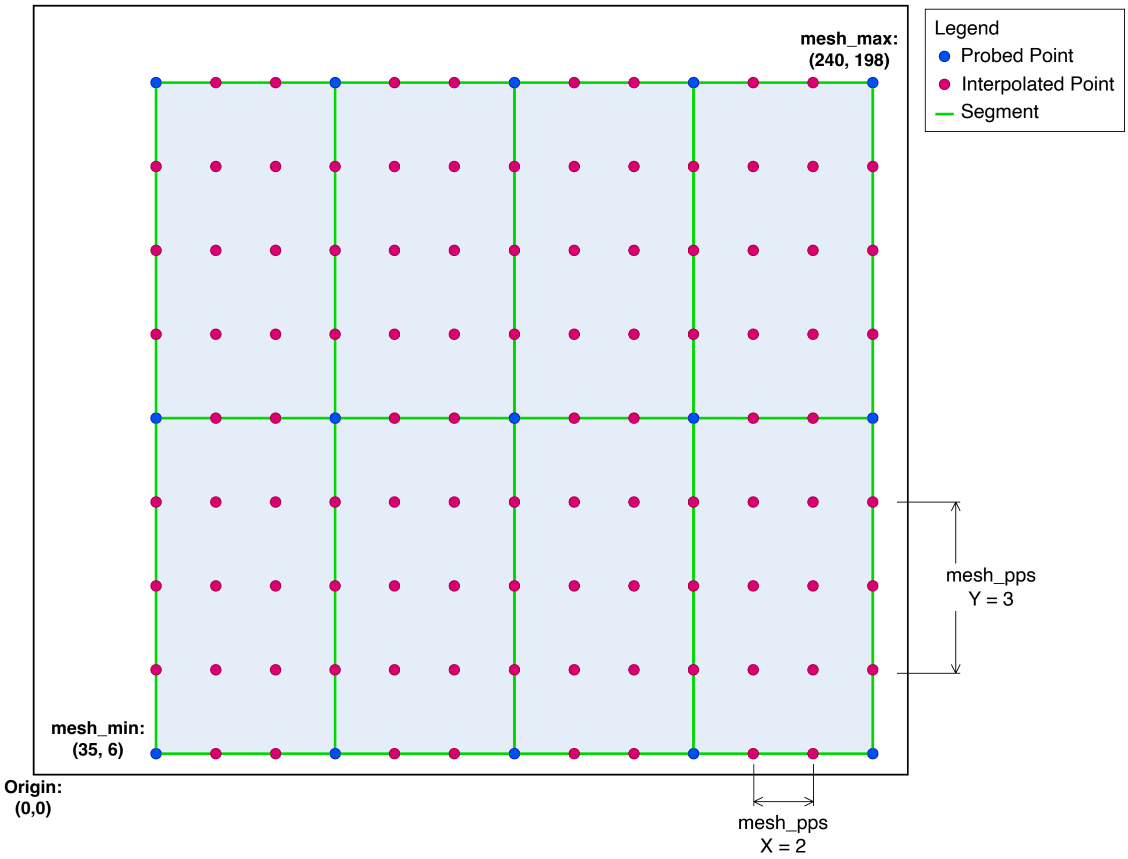
- `mesh_pps: 2, 3`
  Default Value: 2, 2
  The `mesh_pps` option is shorthand for Mesh Points Per Segment. This option specifies how many points to interpolate for each segment along the X and Y axes. Consider a 'segment' to be the space between each probed point. Like `probe_count`, `mesh_pps` is specified as an X, Y integer pair, and also may be specified a single integer that is applied to both axes. In this example there are 4 segments along the X axis and 2 segments along the Y axis. This evaluates to 8 interpolated points along X, 6 interpolated points along Y, which results in a 13x8 mesh. Note that if mesh_pps is set to 0 then mesh interpolation is disabled and the probed matrix will be sampled directly.

- `algorithm: lagrange`
  Default Value: lagrange
  The algorithm used to interpolate the mesh. May be `lagrange` or `bicubic`. Lagrange interpolation is capped at 6 probed points as oscillation tends to occur with a larger number of samples. Bicubic interpolation requires a minimum of 4 probed points along each axis, if less than 4 points are specified then lagrange sampling is forced. If `mesh_pps` is set to 0 then this value is ignored as no mesh interpolation is done.

- `bicubic_tension: 0.2`
  Default Value: 0.2
  If the `algorithm` option is set to bicubic it is possible to specify the tension value. The higher the tension the more slope is interpolated. Be careful when adjusting this, as higher

values also create more overshoot, which will result in interpolated values higher or lower than your probed points.

The illustration below shows how the options above are used to generate an interpolated mesh.



Move Splitting

Bed Mesh works by intercepting gcode move commands and applying a transform to their Z coordinate. Long moves must be split into smaller moves to correctly follow the shape of the bed. The options below control the splitting behavior.

```
[bed_mesh]
speed: 120
horizontal_move_z: 5
mesh_min: 35, 6
mesh_max: 240, 198
probe_count: 5, 3
move_check_distance: 5
split_delta_z: .025
```

- `move_check_distance: 5`
  Default Value: 5
  The minimum distance to check for the desired change in Z before performing a split. In

this example, a move longer than 5mm will be traversed by the algorithm. Each 5mm a mesh Z lookup will occur, comparing it with the Z value of the previous move. If the delta meets the threshold set by `split_delta_z`, the move will be split and traversal will continue. This process repeats until the end of the move is reached, where a final adjustment will be applied. Moves shorter than the `move_check_distance` have the correct Z adjustment applied directly to the move without traversal or splitting.

- `split_delta_z: .025`
  Default Value: .025
  As mentioned above, this is the minimum deviation required to trigger a move split. In this example, any Z value with a deviation +/- .025mm will trigger a split.

Generally the default values for these options are sufficient, in fact the default value of 5mm for the `move_check_distance` may be overkill. However an advanced user may wish to experiment with these options in an effort to squeeze out the optimial first layer.

Mesh Fade

When "fade" is enabled Z adjustment is phased out over a distance defined by the configuration. This is accomplished by applying small adjustments to the layer height, either increasing or decreasing depending on the shape of the bed. When fade has completed, Z adjustment is no longer applied, allowing the top of the print to be flat rather than mirror the shape of the bed. Fade also may have some undesirable traits, if you fade too quickly it can result in visible artifacts on the print. Also, if your bed is significantly warped, fade can shrink or stretch the Z height of the print. As such, fade is disabled by default.

```
[bed_mesh]
speed: 120
horizontal_move_z: 5
mesh_min: 35, 6
mesh_max: 240, 198
probe_count: 5, 3
fade_start: 1
fade_end: 10
fade_target: 0
```

- `fade_start: 1`
  Default Value: 1
  The Z height in which to start phasing out adjustment. It is a good idea to get a few layers down before starting the fade process.

- `fade_end: 10`
  Default Value: 0
  The Z height in which fade should complete. If this value is lower than `fade_start` then fade is disabled. This value may be adjusted depending on how warped the print surface is. A significantly warped surface should fade out over a longer distance. A near flat

surface may be able to reduce this value to phase out more quickly. 10mm is a sane value to begin with if using the default value of 1 for `fade_start`.

- `fade_target: 0`
  Default Value: The average Z value of the mesh
  The `fade_target` can be thought of as an additional Z offset applied to the entire bed after fade completes. Generally speaking we would like this value to be 0, however there are circumstances where it should not be. For example, lets assume your homing position on the bed is an outlier, its .2 mm lower than the average probed height of the bed. If the `fade_target` is 0, fade will shrink the print by an average of .2 mm across the bed. By setting the `fade_target` to .2, the homed area will expand by .2 mm, however the rest of the bed will have an accurately sized. Generally its a good idea to leave `fade_target` out of the configuration so the average height of the mesh is used, however it may be desirable to manually adjust the fade target if one wants to print on a specific portion of the bed.

The Relative Reference Index

Most probes are suceptible to drift, ie: inaccuracies in probing introduced by heat or interference. This can make calculating the probe's z-offset challenging, particuarly at different bed temperatures. As such, some printers use an endstop for homing the Z axis, and a probe for calibrating the mesh. These printers can benefit from configuring the relative reference index.

```
[bed_mesh]
speed: 120
horizontal_move_z: 5
mesh_min: 35, 6
mesh_max: 240, 198
probe_count: 5, 3
relative_reference_index: 7
```

- `relative_reference_index: 7`
  Default Value: None (disabled)
  When the probed points are generated they are each assigned an index. You can look up this index in klippy.log or by using BED_MESH_OUTPUT (see the section on Bed Mesh GCodes below for more information). If you assign an index to the `relative_reference_index` option, the value probed at this coordinate will replace the probe's z_offset. This effectively makes this coordinate the "zero" reference for the mesh.

When using the relative reference index, you should choose the index nearest to the spot on the bed where Z endstop calibration was done. Note that when looking up the index using the log or BED_MESH_OUTPUT, you should use the coordinates listed under the "Probe" header to find the correct index.

Faulty Regions

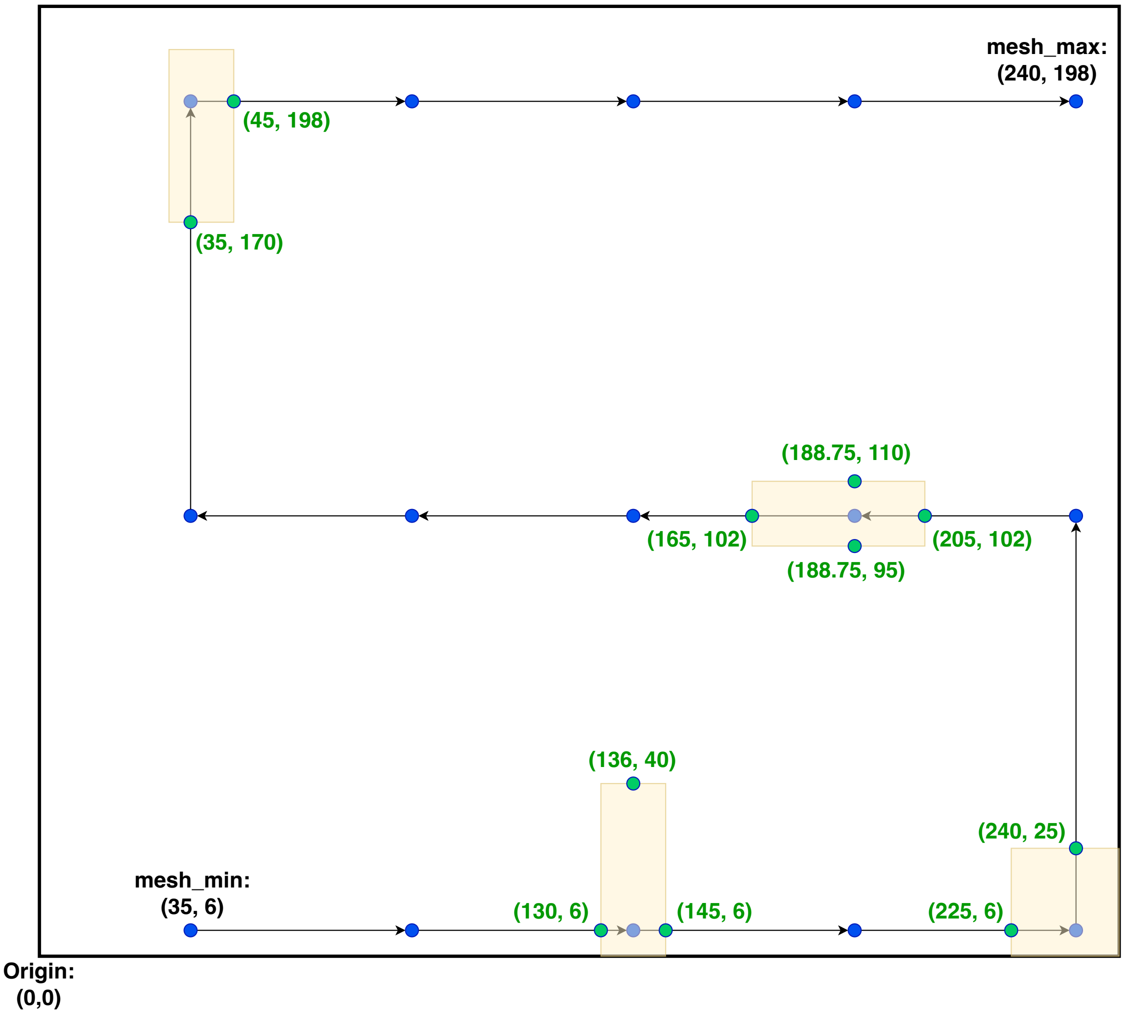
It is possible for some areas of a bed to report inaccurate results when probing due to a "fault" at specific locations. The best example of this are beds with series of integrated magnets used to retain removable steel sheets. The magnetic field at and around these magnets may cause an inductive probe to trigger at a distance higher or lower than it would otherwise, resulting in a mesh that does not accurately represent the surface at these locations. **Note: This should not be confused with probe location bias, which produces inaccurate results across the entire bed.**

The `faulty_region` options may be configured to compensate for this affect. If a generated point lies within a faulty region bed mesh will attempt to probe up to 4 points at the boundaries of this region. These probed values will be averaged and inserted in the mesh as the Z value at the generated (X, Y) coordinate.

```
[bed_mesh]
speed: 120
horizontal_move_z: 5
mesh_min: 35, 6
mesh_max: 240, 198
probe_count: 5, 3
faulty_region_1_min: 130.0, 0.0
faulty_region_1_max: 145.0, 40.0
faulty_region_2_min: 225.0, 0.0
faulty_region_2_max: 250.0, 25.0
faulty_region_3_min: 165.0, 95.0
faulty_region_3_max: 205.0, 110.0
faulty_region_4_min: 30.0, 170.0
faulty_region_4_max: 45.0, 210.0
```

- `faulty_region_{1...99}_min`
  `faulty_region_{1..99}_max`
  Default Value: None (disabled)
  Faulty Regions are defined in a way similar to that of mesh itself, where minimum and maximum (X, Y) coordinates must be specified for each region. A faulty region may extend outside of a mesh, however the alternate points generated will always be within the mesh boundary. No two regions may overlap.

The image below illustrates how replacement points are generated when a generated point lies within a faulty region. The regions shown match those in the sample config above. The replacement points and their coordinates are identified in green.



Bed Mesh Gcodes

Calibration

`BED_MESH_CALIBRATE PROFILE=<name> METHOD=[manual | automatic] [<probe_parameter>=<value>]`
` [<mesh_parameter>=<value>]`
Default Profile: default
Default Method: automatic if a probe is detected, otherwise manual

Initiates the probing procedure for Bed Mesh Calibration.

The mesh will be saved into a profile specified by the `PROFILE` parameter, or `default` if unspecified. If `METHOD=manual` is selected then manual probing will occur. When switching between automatic and manual probing the generated mesh points will automatically be adjusted.

It is possible to specify mesh parameters to modify the probed area. The following parameters are available:

- Rectangular beds (cartesian):
  - `MESH_MIN`
  - `MESH_MAX`
  - `PROBE_COUNT`
- Round beds (delta):
  - `MESH_RADIUS`
  - `MESH_ORIGIN`
  - `ROUND_PROBE_COUNT`
- All beds:
  - `RELATIVE_REFERNCE_INDEX`
  - `ALGORITHM`

See the configuration documentation above for details on how each parameter applies to the mesh.

Profiles

`BED_MESH_PROFILE SAVE=<name> LOAD=<name> REMOVE=<name>`

After a BED_MESH_CALIBRATE has been performed, it is possible to save the current mesh state into a named profile. This makes it possible to load a mesh without re-probing the bed. After a profile has been saved using `BED_MESH_PROFILE SAVE=<name>` the `SAVE_CONFIG` gcode may be executed to write the profile to printer.cfg.

Profiles can be loaded by executing `BED_MESH_PROFILE LOAD=<name>`.

It should be noted that each time a BED_MESH_CALIBRATE occurs, the current state is automatically saved to the *default* profile. If this profile exists it is automatically loaded when Klipper starts. If this behavior is not desirable the *default* profile can be removed as follows:

`BED_MESH_PROFILE REMOVE=default`

Any other saved profile can be removed in the same fashion, replacing *default* with the named profile you wish to remove.

Output

`BED_MESH_OUTPUT PGP=[0 | 1]`

Outputs the current mesh state to the terminal. Note that the mesh itself is output

The PGP parameter is shorthand for "Print Generated Points". If `PGP=1` is set, the generated probed points will be output to the terminal:

```
// bed_mesh: generated points
// Index | Tool Adjusted | Probe
// 0 | (11.0, 1.0) | (35.0, 6.0)
// 1 | (62.2, 1.0) | (86.2, 6.0)
// 2 | (113.5, 1.0) | (137.5, 6.0)
// 3 | (164.8, 1.0) | (188.8, 6.0)
// 4 | (216.0, 1.0) | (240.0, 6.0)
// 5 | (216.0, 97.0) | (240.0, 102.0)
// 6 | (164.8, 97.0) | (188.8, 102.0)
// 7 | (113.5, 97.0) | (137.5, 102.0)
// 8 | (62.2, 97.0) | (86.2, 102.0)
// 9 | (11.0, 97.0) | (35.0, 102.0)
// 10 | (11.0, 193.0) | (35.0, 198.0)
// 11 | (62.2, 193.0) | (86.2, 198.0)
// 12 | (113.5, 193.0) | (137.5, 198.0)
// 13 | (164.8, 193.0) | (188.8, 198.0)
// 14 | (216.0, 193.0) | (240.0, 198.0)
```

The "Tool Adjusted" points refer to the nozzle location for each point, and the "Probe" points refer to the probe location. Note that when manually probing the "Probe" points will refer to both the tool and nozzle locations.

Clear Mesh State

`BED_MESH_CLEAR`

This gcode may be used to clear the internal mesh state.

Apply X/Y offsets

`BED_MESH_OFFSET [X=<value>] [Y=<value>]`

This is useful for printers with multiple independent extruders, as an offset is necessary to produce correct Z adjustment after a tool change. Offsets should be specified relative to the primary extruder. That is, a positive X offset should be specified if the secondary extruder is mounted to the right of the primary extruder, and a positive Y offset should be specified if the secondary extruder is mounted "behind" the primary extruder.