

SUPPORTING INFORMATION FOR *A physics-based modeling framework to assess the cost scaling of additive manufacturing, with application to laser powder bed fusion*

Kaitlyn Gee, Suh In Kim, Haden Quinlan, A. John Hart

Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA

Appendix 1. Digital file slicing and contour generation

The code for importing, manipulating, and slicing mesh geometry was developed in-house with the Julia scientific programming language. Visual representation of the graphics processing workflow and the contour generation algorithm are shown in Figures A.1 and A.2, respectively.

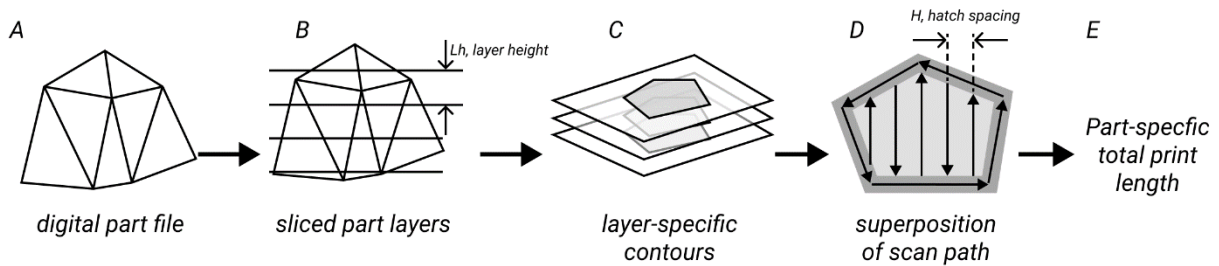


Fig A.1. Graphics processing workflow for the physics-based rate model, which simulates part slicing and, conceptually, translation to machine-level print paths. (A) Digital part file is imported as an .stl mesh, (B) mesh is sliced by taking triangle-plane intersections at each print plane, (C) for each print plane, closed-loop contours are found, (D) a representative print strategy (i.e. laser scan paths for the skin and core) are superimposed onto the contours, (E) a part-specific total print length is output. For LPBF, total print length represents the total length of laser scanning required to print a given part.

The slicing operation results in a list of unordered intersection edges for each slice plane. Assuming that the mesh is watertight and no errors have occurred in slicing, this list of unordered edges can be structured into closed-loop paths that represent the perimeter of material to be printed on a given layer. If the mesh is not watertight, the algorithm will attempt to match the remaining vertex to its closest neighbor within a tolerance limit (i.e. if the distance between two vertices are lower than $1\text{E-}6$ mm, for example); if no neighbor is found, the algorithm will throw an exception. Details are shown in Figure A.2.

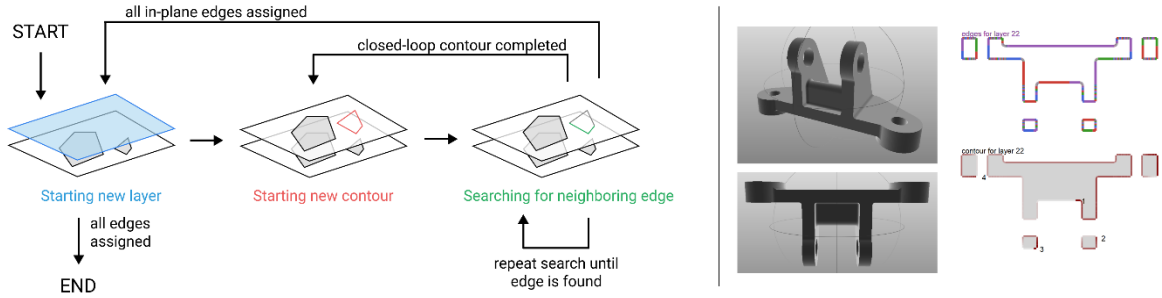


Figure A.2. (Left) State machine-based algorithm for contour generation. The algorithm begins by initializing the data structures to store the edges for a given slice plane. An edge is selected to begin the first contour, and the algorithm iterates through the remaining edges in the slice plane until a neighboring edge is found. If additional edges remain in the slice plane, a subsequent contour is started. The algorithm finishes when all edges for all layers have been assigned to contours. (Right) Raw output from the slicing (top) and contour generation (bottom) algorithms, shown against input part geometry. The area bounded by the closed-loop contour is shown in grey in the contour output example.

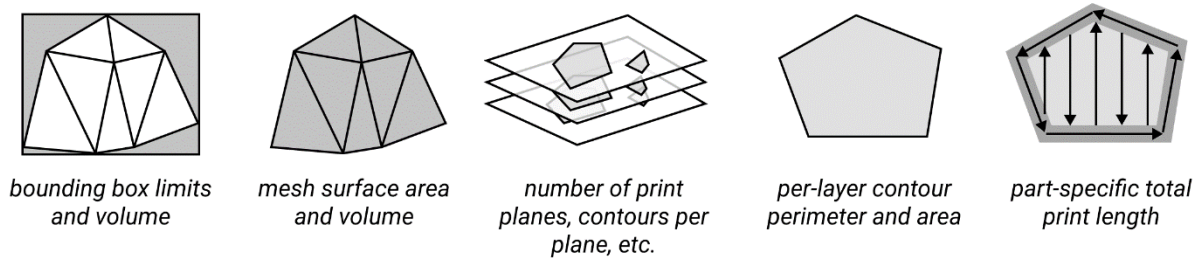


Figure A.3. From directly interrogating the part geometry, we extract additional geometric parameters: mesh volume, mesh surface area, bounding box limits and volume, area per print layer, perimeter per print layer, and number of contours per print layer. This computational workflow therefore allows a more nuanced, part-specific estimation of to-print length (and therefore volume).

Appendix 2. Parameterized, activity-based cost model for LPBF

This section describes the formulation of our physics- and activity-based cost estimator, which estimates the per-part cost of additively manufacturing a single component. To do so, we begin with a production run of size N identical parts. The underlying assumptions are accordingly:

- The run comprises identical parts, and thus costs associated with the full production run can be evenly distributed across each part. This scenario does not consider mixed-part builds as is done in industry to minimize cost.
- The costs of build failure are negligible, i.e. no overhead is applied to account for non-perfect yield.
- The major contributor for production cost lies in the printing step; heat treatment, depowdering and post-processing costs are not considered in the model. However, we note that post-processing costs can be high and therefore an important consideration when assessing cost for LPBF.

Per-part cost is decomposed into four categories, and costs are assigned based on the corresponding activities and expenses to each category (Baumers *et al.*, 2012). The four categories considered are: material cost, machine usage cost, consumables cost (i.e. inert gas, build plate), and labor cost. Thus, the estimated cost per part can be expressed as

$$C_{total} = C_m + C_e + C_c + C_l$$

where C_m , C_e , C_c , C_l are the costs due to material usage, equipment/machine utilization, consumables, and labor, respectively.

First, we translate the production size n into the total number of builds required N . Assuming that the part footprint A_p (i.e. the width and length of the part's bounding box) and build chamber area A_c are known, the number of parts per layer $nPartsPerLayer$ is estimated as:

$$nPartsPerLayer = \lfloor A_c/A_p \rfloor$$

where $\lfloor x \rfloor$ is the floor operator and denotes the greatest integer less than or equal to x . The part footprint A_p is found in the graphics processing workflow described in the main body text. The number of layers per build $nLayersPerBuild$ is set to one.

In polymer SLS or other printing techniques where volumetric packing is used, the $nLayersPerBuild$ could be calculated as:

$$nLayersPerBuild = \lfloor H_c/H_p \rfloor$$

where H_c and H_p are the absolute heights of the build chamber and part, respectively. Visualization of the simplified single-layer packing scheme is shown in Figure 4 of the main text.

The number of parts per build is:

$$NperBuild = nPartsPerLayer * nLayersPerBuild$$

and the total number of builds N is

$$N = \lceil n/NperBuild \rceil$$

where $\lceil x \rceil$ is the ceiling function and denotes the lowest integer greater than or equal to x .

The total volume of all print layers V_p is therefore

$$V_p = nLayers A_c H_p$$

where $A_c H_p$ represents the volume of a single part layer. The total number of part layers $nLayers$ for the production run is

$$nLayers = \lceil N / N_{perBuild} \rceil$$

This approach simulates a simple 2d-packing approach, sensitive to the aspect ratio of the part's bounding box. Specifically, we implement single-layer (i.e. planar) packing. Volumetric packing (multiple part layers per print) is used in other processes, such as polymer SLS and in general the choice of packing depends on process, part geometry, and support structure design. We assume that the inclusion of supports modifies the print mass, but not the part's bounding box dimensions, and that parts can be stacked vertically. Distinguishing between number of part layers and number of builds allows more accurate estimations of activity-based cost (i.e. some operations are layer-based such as recoating while others are build-based such as gas flushing).

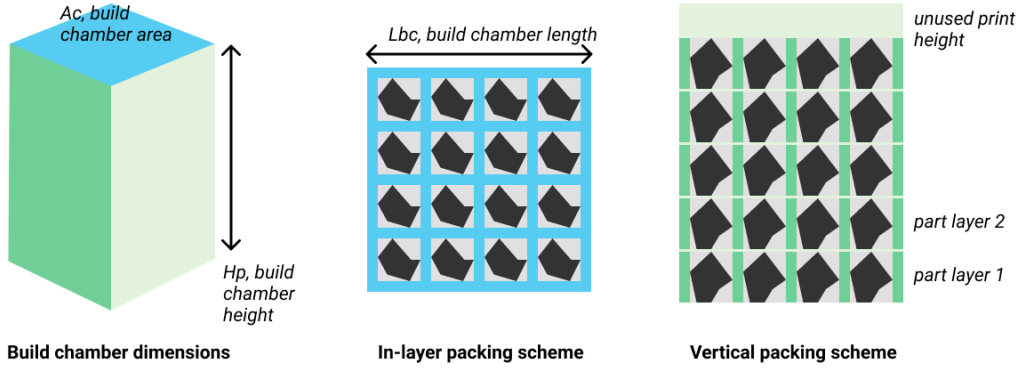


Figure A.4. Schematic representation of volumetric packing (i.e. multiple part layers) for other AM processes such as polymer SLS.

The following sections outline the analytical models for each categorical cost.

A2.1 Material Cost

Material cost has three components: (a) the cost of powder to produce the finished part, (b) the cost of powder to produce part supports, and (c) the cost of surrounding powder that cannot be recycled.

For a part of volume V and final density ρ , the material cost to produce the finished part C_m^a is:

$$C_m^a = V \rho C P$$

where CP is the cost of powder in \$/kg. Part volume V is defined as the volume bounded by the digital part file and is calculated using the graphics processing workflow described in the main text. The cost of powder to produce the supports C_m^b is:

$$C_m^b = 0.2 V \rho C P$$

i.e. we assume that the mass of the supports is 20% of the part mass. Because the part and support are the same material, this implies that $V_{support} = 0.2V$, i.e. that support volume is 20 % of part volume. Finally the cost of unrecycled powder C_m^c is:

$$C_m^c = \frac{(1 - \gamma)\rho V_u P nLayers}{n}$$

where γ is the recycling fraction, i.e. the portion of unused powder than can be recycled. For the purposes of this analysis, recycling fraction is 0.95 for all materials. V_u is the volume of unused powder in a single part layer:

$$V_u = A_c H_p - (V + 0.2V) nPartsPerLayer$$

where $A_c H_p$ is the volume of a single part layer, V is the volume occupied by the part, and $0.2 V$ is the volume occupied by the supports. This approach assumes that the volume decreases from unsolidified to melted powder is negligible for estimating material cost. Therefore, we expect our estimate of material cost to be a slight underestimate.

Total per-part material cost is:

$$C_m = C_m^a + C_m^b + C_m^c$$

A2.2 Machine Cost

Machine cost considers the capital investment required for the printer itself, its supporting infrastructure requirements, as well as ongoing maintenance costs. We use the subscript e for equipment. The cost of the machine is expressed as a yearly cost, from which an hourly utilization cost is derived.

The yearly cost of machine ownership is approximated by the yearly depreciation cost in the first year of ownership:

$$C_{e,yr} = (C_{machine} + C_i) \frac{r}{1 - (1 + r)^L} + C_{maint} * C_{machine}$$

where $C_{machine}$ is the up-front purchase cost of the machine, C_i is the infrastructure cost, r is the discount rate, L is the predicted lifetime of the machine in years, and C_{maint} is the maintenance rate defined as the portion of capital cost. The underlying depreciation model is shown below.

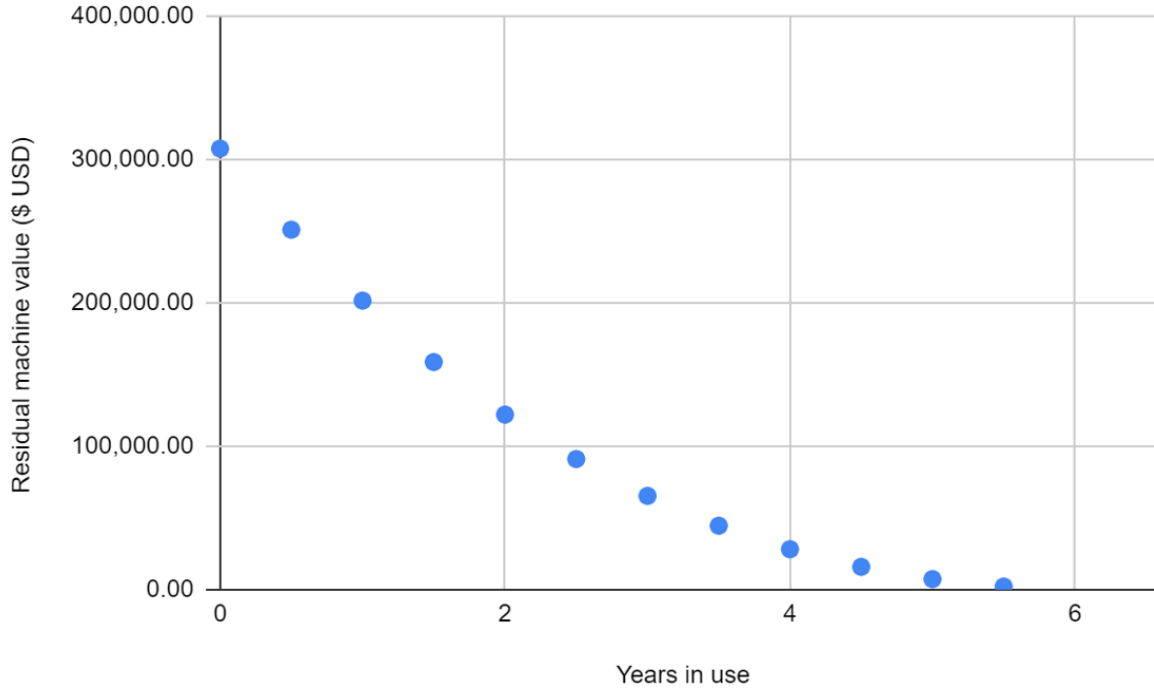


Figure A.5. Illustration of the residual value of the machine versus time in present-day dollars over time. In this example, capital machine cost is 248,000 USD and infrastructure costs are 124,000 USD (i.e. half of capital machine cost). The discount rate r is 0.05 and the expected lifetime L is 7 years. The yearly machine depreciation, and therefore yearly cost of machine ownership not including maintenance, is estimated to be 64,000 USD for the first year of machine ownership in this instance.

From this, the cost of machine utilization per hour can be expressed as:

$$C_{e,hr} = \frac{C_{e,yr}}{h_{yru}}$$

where h_{yr} is the number of operating hours per year and u is uptime, i.e. the fraction of operating hours wherein the machine is functional and utilized. A value of u lower than 1 simulates the machine being offline for maintenance or repairs.

Next, we estimate the time associated with printing a single part, which is equivalent to the machine utilization time. A full build sequence includes: (a) build preparation wherein print parameters are selected and an engineer configures the physical layout of parts in the build chamber, (b) build chamber warm-up and gas flush, (c) the print itself, (d) cool-down, and (e) build plate removal and build exchange. Step (a) occurs only once per production run, and does not require machine time.

The per-part time associated with build chamber warm-up and gas flush is:

$$t_b = t_w N / n$$

where t_w is a constant warm-up time for a machine. This approach amortizes the time associated with warming up N times across the n production parts. A similar approach is applied for cool-down (i.e. $t_d = t_{cool} N / n$) and print exchange (i.e. $t_e = t_{exch} N / n$) where t_{cool} is the time in hours required for machine cool-down and t_{exch} is the time in hours required for build exchange. The time to print is estimated as:

$$t_c = t_{scan} / N_L + t_r / nPartsPerLayer$$

where t_{scan} is the estimated total scanning time in hours required to produce a single part, N_L is the number of lasers of the machine and t_r is the total recoating time in hours associated with a given part layer (assuming that recoating time is independent of the number of parts per layer). Recoating time is therefore evenly distributed amongst all parts in a layer and is calculated as:

$$t_r = (L - 1)(1.4L_{bc}(1/v_f + 1/v_b) + t_v)$$

where L is the number of print layers required to produce a part layer, L_{bc} is the length of the build chamber in the direction of recoat (i.e. the length over which the recoater travels), v_f is the forward velocity of the recoater, v_b is the backwards velocity of the recoater, and t_v is the time taken for vertical travel of the build plate. We assume that the recoater travel length exceeds the length of the build chamber by a factor of 1.4 (i.e. such that edge effects of recoater travel does not impact the print area). This granularization of recoating time therefore is sensitive to print quality (i.e. print resolution as quantified by layer height) as well as machine capability (i.e. recoating velocity) and physical size.

Note that t_{scan} is a complex function of machine capability (laser power, spot size), print parameters (overmelt, hatch spacing, layer height) and machine properties. Therefore, the time to print t_c is process- and machine-specific, and driven not by empirical data of machine-specific build rate but their base specifications. This approach enables us to estimate cost for a wider range of machine capabilities. The total, per-part production time is:

$$t_p = t_b + t_c + t_d + t_e$$

$$t_p = t_{scan}/N_L + t_r/nPartsPerLayer + (t_w + t_{cool} + t_{exch})N/n$$

Taken in total, the cost of machine usage per part can therefore be expressed as:

$$C_e = t_p C_{e,hr}$$

	m100	m290	m400	m400-4	
Laser power	200	400	1000	400	W, nominal laser wattage
Laser spot size	40	100	90	90	μm
A_chamber	25π	25^2	40^2	40^2	cm^2 , area of build plate

H_chamber	9.5	32.5	40	50	cm, vertical height of build chamber
C_printer	2.48E+05	5.00E+05	1.00E+06	1.76E+06	\$ USD, capital cost of printer
Lifetime	7	7	7	7	Years, overall lifetime of printer
Discount rate	0.05	0.05	0.05	0.05	Discount rate used in yearly cash flow analysis
Maintenance rate	0.05	0.1	0.1	0.1	Maintenance rate of machine cost per year
Uptime	0.9	0.9	0.9	0.9	Fraction of operating hours that yield productive output
Infrastructure rate	0.5	0.5	0.25	0.25	One-time capital cost for printer-related infrastructure, in the form of a multiple of C_printer
nLasers	1	1	1	4	Number of lasers on printer
NhoursPerYear	7884	7884	7884	7884	Hrs, total operational hours per year
Gas flow rate	10	3	3	3	L/hr, Volumetric gas flow, defined as a multiple of the build chamber volume flow per unit hour

Table A.1. Machine-specific inputs to activity-based cost.

A2.3 Consumables Cost

For LPBF, the primary build consumables are (i) gas flow used to inert the build chamber before and during print, and (ii) the build plate, (iii) energy consumption of the machine. While energy consumption may have large implications for the environmental impact of LPBF, we argue that the cost of electricity is negligible in relation to other per-part costs.

To approximate (i), we consider an initial gas flush at the beginning of every build, as well as continuous gas flow during print. The cost of the initial gas flush c_f is

$$c_f = qV_{bc}N/nC_{gas}$$

Where V_{bc} is the volume of the build chamber and C_{gas} is the per-volume cost of inert gas. We assume that volume of gas used to flush the chamber sufficiently is q times the build chamber volume. Gas flush will occur N times per production run, normalized by the production quantity n . The cost of gas flow during the print c_{gp} is

$$c_{gp} = \dot{V}_g(t_b + t_c + t_d)C_{gas}$$

where \dot{V}_g is the volumetric gas flow rate, t_b is per-part warm-up time, t_c is per-part print time, and t_d is per-part cool-down time. We assume that \dot{V}_g is a function of build chamber volume.

To approximate (ii), we consider a refurbishment model wherein build plates are reused a finite number of times. The amortized, per-part cost of build plate use is:

$$c_{bp} = (C_{plate}/P + C_{refurb})N/n$$

where C_{plate} is the initial cost of the plate, P is the number of times the plate can be reused and C_{refurb} is the cost to refurbish a plate.

Combining (i) and (ii), i.e. the cost of gas flush, gas flow, and build plate use, gives:

$$C_c = c_f + c_{gp} + c_{bp}$$

Substituting and rearranging:

$$C_c = \dot{V}_g(t_b + t_c + t_d)C_{gas} + (10V_{bc}C_{gas} + C_{plate}/P + C_{refurb})N/n$$

Therefore, consumable cost is a function of machine capability (through t_c) as well as size (through V_{bc} and \dot{V}_g).

A2.4 Labor Cost

Our last category is labor, accounting for the work of engineers and build operators in setting up, supervising and exchanging builds. We consider the labor associated with the physical build itself, not including the labor costs of part and process design in the product development phase.

For this model, we assume that an engineer will configure the print (i.e. select appropriate build parameters and physically lay out the components in the print volume) and a build operator will handle printer warm-up (including gas purging), print supervision, printer cool-down and build exchange. For the printer warm-up, print supervision, and printer-cool down stages, we assume that the operator is only 10% dedicated to the print, i.e. that 90% of their labor costs can be attributed to other tasks the operator performs in parallel.

The labor cost for setup C_{setup} is

$$C_{setup} = t_a C_{eng}/n$$

where t_a is the time required to set up the print, and C_{eng} is the hourly wage of the engineer. Setup costs are distributed among the n parts of the production run.

The warm-up, print, cool-down, and exchange costs are:

$$C_{warm} = 0.1t_b C_{op}$$

$$C_{print} = 0.1t_c C_{op}$$

$$C_{cool} = 0.1t_d C_{op}$$

$$C_{exch} = t_e C_{op}$$

where t_b , t_c , t_d and t_e are the per-part times associated with machine warm-up, print, machine cool-down, and build exchange, respectively. C_{op} is the hourly wage of the operator. Total per-part cost of labor C_l is:

$$C_l = C_{setup} + C_{warm} + C_{print} + C_{cool} + C_{exch}$$

Cost description

Parameter		Value
P	Number of times a build plate can be refurbished	10
C_refurb	Cost of a single build plate refurbishment	25 \$ USD
C_gas	Volumetric cost of inert gas	0.018 \$ USD /L
C_operator	Fully burdened hourly wage for AM operator	40 \$/hr
C_engineer	Fully burdened hourly wage for AM engineer	60 \$/hr
FTE_run	Fraction of attention required for overseeing the print (i.e. one operator could supervise 5 machines at once)	0.2

Table A.2. Inputs to consumable and labor-related costs.

A2.4 Total Production Cost

The total per-part production cost is estimated as

$$C_{total} = C_m + C_e + C_c + C_l$$

i.e. the linear sum of costs incurred from material, machine, consumables and labor expenses.

Appendix 3. Re-melt analysis

An analytical solution was developed to calculate the number of times each point in the build is re-melted, with simplifying assumptions. The approach is based on discretizing a given unit volume of a print based on in-layer overlap between adjacent melt tracks (i.e. horizontal) and inter-layer overlap between print layers (i.e. vertical).

A3.1 Analytical form of a single melt track

Our model assumes that the user will specify layer height L_h and overmelt ϕ (i.e. the degree to which melt track extends past the layer height, defined as a multiple of layer height. In realistic printing conditions, overmelt will always be greater than 1. Therefore, the full depth of the melt track d is

$$d = \phi L_h$$

The width of the parabola w is an input parameter of this analysis. In integration with our rate model, width will be a dependent variable based on melt track depth d , laser scanning parameters and material thermodynamic properties. Assigning the minimum of the melt track parabola to the origin of a coordinate reference frame, we apply the constraint that the parabola would intersect with $z(-w/2) = z(w/2) = \phi L_h$ and $z(x = 0) = 0$ where the x -coordinate corresponds to melt track width (i.e. horizontal direction) and the z -coordinate corresponds to the melt track depth (i.e. vertical direction)

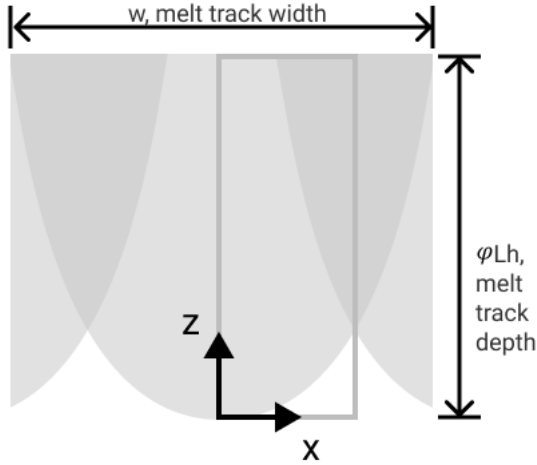


Figure A.6. Melt track geometry of width w and depth ϕL_h . Grey window denotes unit cell.

The analytical equation expressing the parabolic melt track is therefore:

$$z = \phi L_h \frac{4}{w^2} x^2$$

This equation is specific to a melt track with a non-rotated scan direction, i.e. the laser scan speed is parallel to the y -axis and perpendicular to the x - y plane.

A3.2 Analytical form of a rotated melt track

To account for realistic scan strategy, the analytical form of a rotated melt track (i.e. melt track produced by a laser scanning in a rotated direction) is also required. Rotation occurs about the z -axis (Figure A.7).

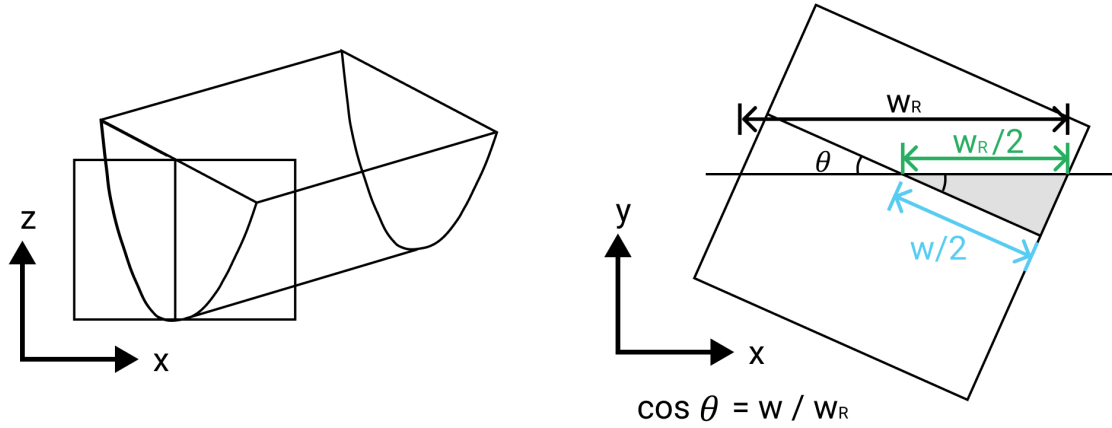


Figure A.7. Accounting for width of rotated melt tracks. (Left) Rotation of a melt track about the z-axis. (Right) The width of the rotated parabolic prism in the xz plane is given by $(w \cos(\theta))$.

We can then write from rotational symmetry that:

$$w_R = w / \cos$$

The analytical equation expressing the rotated, parabolic melt track is therefore:

$$z = \phi L_h \frac{4}{w_R^2 x^2}$$

$$z = \phi L_h \cos^2 \theta \frac{4}{w^2} x^2$$

A3.3 Mathematical definition of hatch spacing

The ultimate goal is to quantify how the parabolas overlap, as overlap represents re-melting during print. With the analytical equations of the parabolas defined (i.e. their shape and curvature), the next step is to express how the parabolas are laid out in space (i.e. their horizontal placement within a print layer).

We first define the intersection depth, which is the vertical depth at which two adjacent melt tracks intersect (Figure 2A of main text). Here, β is the intersection parameter and drives the depth at which two adjacent parabolas intersect:

$$d_{intersection} = \beta L_h$$

Note that $d_{intersection}$ is defined relative to the bottom of the melt track, such that a lower value of $d_{intersection}$ (and therefore of β) corresponds to greater overlap.

Taking the analytical equation for a rotated, parabolic melt track and substituting intersection depth for z yields:

$$\beta L_h = \frac{4\phi L_h \cos^2 \theta}{w^2} x_i^2$$

We solve for x :

$$x_i = \sqrt{\frac{\beta L_h w^2}{4\phi L_h \cos^2 \theta}}$$

Simplify:

$$x_i = \frac{w}{2 \cos \theta} \sqrt{\frac{\beta}{\phi}}$$

Here, x is the distance between the centerline of the parabola to the intersection point and is therefore equal to half of the hatch spacing. Hatch spacing is therefore:

$$H = \frac{w}{\cos \theta \sqrt{\frac{\beta}{\phi}}}$$

With the analytical forms of non-rotated and rotated parabolas established, along with parametric equations for melt pool dimensions and hatch spacing, the full horizontal layout of melt tracks can be defined as functions of w , L_h , ϕ , and θ as shown in Figure A8. The layout can be represented by a repeating width, i.e. a "slice" of the horizontal layout that captures the overlapping behavior of a parabola and its neighbors. The unit width is shown below in Figure A.8C (as the orange area). The left boundary of the width is $x = 0$, i.e. the center of a parabola. The right boundary of the width is $x = H/2$, i.e. the centerline of the intersection between the parabola and its neighbor. The re-melt profile of the region between $0 \leq x < H/2$ therefore represents the re-melt profile of the entire part.

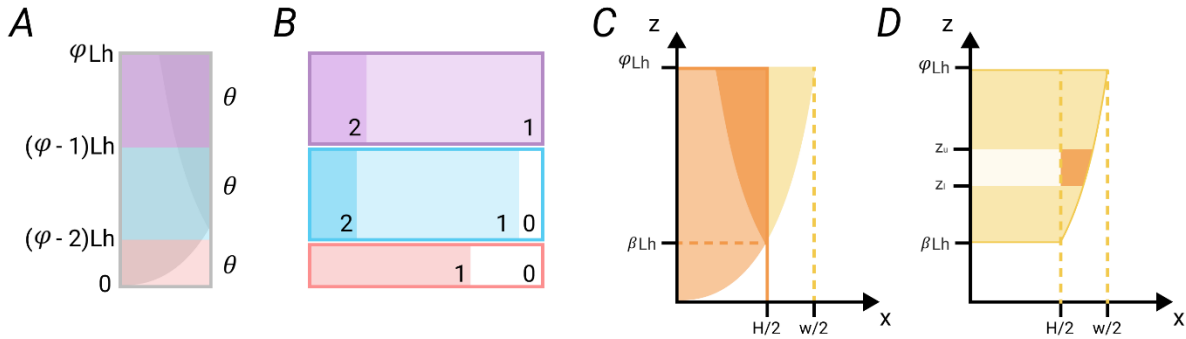


Fig A.8. (A) Output of vertical discretization is a unit-width melt track discretized into layer heights. (B) Proportion of area re-melted zero, one or two times for each layer increment, from evaluating area integrals. (C) Unit cell for area integral evaluation. Two integration regimes are required: one for evaluating above $z = \beta L_h$ and one for evaluating below $z = \beta L_h$. (D) Illustration of integration between z -bounds. The upper integration regime relies on a symmetric boundary condition at $x = H/2$.

A3.4 Vertical discretization

Next we must consider the vertical dimension. A single melt track may span multiple layers, depending on its depth to layer height ratio (i.e. ϕ) (Figure A.9). For a given layer, the full re-melt profile must consider subsequent layers that overlap -- from Figure A.9, the full-re-melt profile must consider the contribution of layer $i = 0$, as well as layers $i = 1, 2$, and 3 (labelled here as 'partial'). An equivalent approach is discretizing a single melt track profile into layer height increments (wherein each increment represents the contribution of a layer to the melt track profile), and then superimposing them, thus accounting for all overlapping layers. We follow the latter approach.

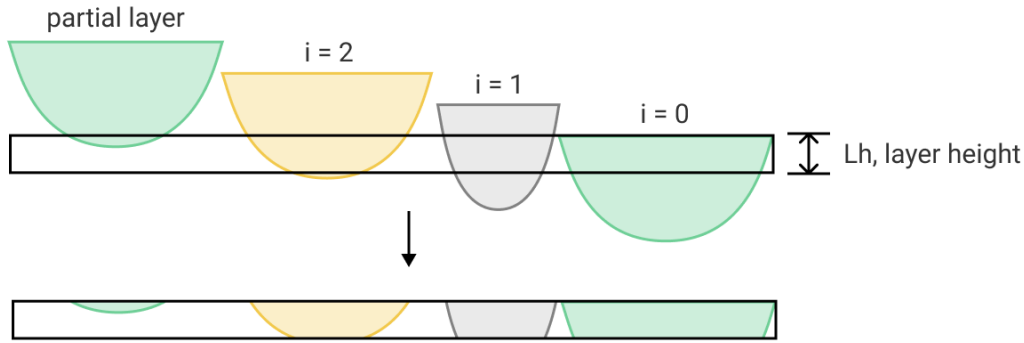


Fig A.9. Decomposition of overlapping melt tracks for a given layer height (i.e., black box). Since each melt track may span multiple layers, the number of times the material within a single layer is re-melted must consider the superposition of all intersecting melt tracks. The current layer is ($i = 0$) and the black box denotes a single layer height. Subsequent layers contribute to the melt profile of the layer height, and have individual melt profiles based on rotation.

The unit cell of interest has the unit width described above (i.e. $0 \leq x < H/2$) and spans the depth of the melt track (i.e. $0 \leq z < \phi L_h$). The number of full layer heights encompasses in the unit cell is:

$$n_{Full} = \lfloor \phi \rfloor$$

and the total number of layers to consider is:

$$n_{Total} = \lfloor \phi \rfloor$$

Therefore, the absolute z -bounds of each full layer are:

$$z_{l,i} = (\phi - i)L_h$$

$$z_{u,i} = (\phi - i + 1)L_h$$

for each i th layer with $i \leq nFull$. The bounds for the partial layer, if $ntotal > nFull$, will be $0 < z < (\phi - nFull) * L_h$.

Therefore, each layer increment is defined by x-bounds of $0 \leq x < H/2$ and z-bounds as described above. The next step is to compute the area proportions associated with each layer increment, i.e. the proportion of material within this window that is re-melted once, twice or zero times.

A3.4 Computation of area proportions

For each layer increment, regions are melted once (i.e., bounded by a single parabola), melted twice (i.e., bounded by the intersection of two parabolas) or not melted (i.e., at the bottom) of the view. Since the equations of the parabolas and their placements are known, precise areas can be calculated from computing integrals. Two integration regimes are required -- one to account for above the intersection and one for below (Figure A.8C and Figure A.8D).

A3.5 Upper integral regime

We begin with the upper integral regime, i.e. for $\beta L_h \leq z \leq \phi L_h$. The goal of this analysis is to integrate the analytical equation for parabolas and to quantify the area re-melted twice, once, or not at all within the layer increment. Since the width of our integration area is fixed at $H/2$ and the z-heights vary based on the layer, we choose to integrate over dz .

For bounds of integration $z = z_l$ to $z = z_u$, we observe that no area within the upper integral regime can be re-melted zero times from inspection:

$$A_0(z_l, z_u) = 0$$

Next, we consider the area re-melted twice, i.e. the dark orange region of Figure A.8C. The area between $z = 0$ and the yellow curve (which has the equation of our base parabola) is:

$$a_2(z_l, z_u) = \int_{z_l}^{z_u} \frac{w}{2\cos\theta} \sqrt{\frac{z}{\phi L_h}} dz$$

We subtract the area of a rectangle bounded by $H/2$ and $(z_u - z_l)$ (i.e. the area of a layer increment):

$$A_2(z_l, z_u) = a_2 - (z_u - z_l) \frac{H}{2}$$

Evaluating the integral and simplifying gives:

$$A_2(z_l, z_u) = \frac{w}{2 \cos \theta} \left[\frac{2}{3\sqrt{\phi l_h}} (z_u^{3/2} - z_l^{3/2}) - \sqrt{\beta/\phi} (z_u - z_l) \right]$$

which is the absolute area between $z = z_l$ and $z = z_u$ that would be re-melted twice. The area re-melted once is the remaining area in the layer increment:

$$A_1(z_l, z_u) = \frac{w}{2 \cos \theta} \sqrt{\frac{\beta}{\phi}} (z_u - z_l) - A_2$$

A3.6 Lower integral regime

The same analysis applies to the area underneath $z = \beta L_h$. First, observe that no area can be re-melted twice below the intersection point, so the area re-melted twice for the lower integral regime B_2 must be zero:

$$B_2(z_l, z_u) = 0$$

The area re-melted zero times within this reference frame b_0 is simply the area under the $x = \frac{w \cos \theta}{2} \sqrt{z \phi L_h}$ curve:

$$b_0(z_l, z_u) = \int_{z_l}^{z_u} \frac{w}{2 \cos \theta} \sqrt{\frac{z}{\phi L_h}} dz$$

Evaluating gives the area re-melted zero times aka B_0 :

$$B_0(z_l, z_u) = \frac{w}{3 \cos \theta} \sqrt{\frac{\beta}{\phi}} (z_u^{3/2} - z_l^{3/2})$$

This region of zero melting occurs only when considering a single layer increment and does not imply that any area of the printed part will be insufficiently melted. By superimposing the contributions of print layers, the final re-melt profile of the unit cell, assuming proper print parameter selection, will not contain any un-melted area.

Finally, the area re-melted once is the unassigned, remaining area of the layer increment:

$$B_1(z_l, z_u) = \frac{w}{2 \cos \theta} \left[\sqrt{\beta/\phi} (z_u - z_l) - \frac{2}{3\sqrt{\phi l_h}} (z_u^{3/2} - z_l^{3/2}) \right]$$

A3.7 Area proportions

The total area considered per increment has the width of $H/2$ and a height of $z_u - z_l$:

$$A_T = \frac{w}{2\cos\theta} \sqrt{\frac{\beta}{\phi}} (z_u - z_l)$$

Therefore, for an area remelted n times A_n , the corresponding area proportion k_n is:

$$k_n = A_n/A_T$$

The remainder of the analysis proceeds with area proportions instead of absolute area.

A3.8 Superposition of layer increments

The final step of the workflow is to superimpose the area proportions of each layer increment, with the end goal being the overall proportions of area re-melted 0 to n times within the region, with n being the maximum number of overlapping parabolas possible.

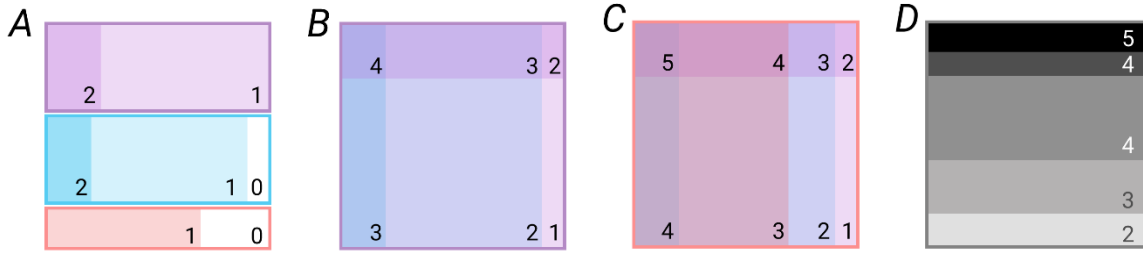


Fig A.10. Superposition of layer increments to compute total melt profile. (A) Output of area proportion evaluation is the region melted 0, 1, or 2 times within each layer increment. (B) Superposition of two layers (i.e. blue and purple) yields the areas melted 0, 1, 2, 3, or 4 times. Note that the area melted zero times is zero once two layers are considered in this example. (C) Superposition of three layers. The algorithm will recursively combine all layer increments. (D) Final output of the model. The area proportions apply to a unit cell of width $H/2$ and height L_h .

Starting with the simplest case, the overlap between two layer increments can produce area that is melted 4 to 0 times, from a strictly mathematical sense. Consider two vectors k and q representing the area proportions for two layers:

$$k = [k_0 \ k_1 \ k_2]$$

$$q = [q_0 \ q_1 \ q_2]$$

where k_0 corresponds to the area proportion re-melted zero times for layer k , etc. Next, take the product of the two vectors W :

$$k^T \cdot q = W = \begin{bmatrix} k_0 * q_0 & k_0 * q_1 & k_0 * q_2 \\ k_1 * q_0 & k_1 * q_1 & k_1 * q_2 \\ k_2 * q_0 & k_2 * q_1 & k_2 * q_2 \end{bmatrix}$$

The resulting area proportions can be found from summing the antidiagonal elements of W . For example the area proportion re-melted twice \dot{A}_2 is:

$$\dot{A}_2 = W[3,1] + W[2,2] + W[1,3]$$

This process is shown graphically in Figure A.10, and proceeds recursively until all layers (i.e. $nTotal$ layers) are considered. Pseudocode for computing the area proportions is:

```
for i = 1:nTotal
    define rotation* and z-bounds
    calculate hatch
    compute area proportions
    save area proportions
end
```

*Rotation is defined as rotation angle between the y-axis and the scan direction. For example, the standard sequence $\theta = 0, \pi/3, -\pi/3$ is common industry practice.

The end result of the re-melt analysis is a $nTotal$ by 3 matrix (i.e. called *areas*), with each row of the matrix corresponding to a layer, and each column corresponding to the area proportion melted zero, one or two times respectively. The layers are then superimposed using a recursive function:

```
function recursiveCombine(areas)
    if first row is longer than the second, lengthen second row with zeros**
    compute dot product
    total area proportions = sum along antidiagonals
    remove first and second row from areas
    if more rows remain in areas
        add total area proportions to top of areas
        recursiveCombine(areas)
    end
```

**This step accounts for the fact that the possible number of re-melted areas increases with each combination. For example, two layers superimposed could have a re-melt profile ranging from 0 to 8 times. Three layers superimposed could have a re-melt profile ranging from 0 to 16 times re-melted. However, the recursive algorithm requires that the two rows have the same number of elements.

The final output of this workflow is a vector wherein each n th element is the proportion of area within a unit cell of width $H/2$ and height L_h remelted n times. This re-melt profile is parameterized by L_h, β, ϕ, θ and w . This unit cell is the minimum repeating window representing the re-melt profile of the entire print, such that this final vector represents the local number of times area is re-melted within a part.

References

Baumers, M., Tuck, C., Wildman, R., Ashcroft, I., Rosamond, E. and Hague, R. (2012), “Combined build-time, energy consumption and cost estimation for direct metal laser sintering”, *23rd Annual International Solid Freeform Fabrication Symposium - An Additive Manufacturing Conference, SFF 2012*, No. January, pp. 932–944.