Additive Build Advisor — Technical Report

Seymur Hasanov

Contents

1 Technical Report: Additive Build Advisor **1**
1.1 Overview . . . 1
1.2 System architecture . . . 2
1.3 Geometry and the voxelization engine . . . 2
1.3.1 Engine validation . . . 4
1.4 Orientation: rest on a flat face . . . 4
1.5 Build simulation . . . 4
1.6 Warpage FEA: the thermal-contraction method . . . 6
1.6.1 Process focus and basis . . . 8
1.6.2 What the number means: on-bed vs after-removal . . . 8
1.6.3 FEA validation . . . 8
1.7 Cross-process comparison . . . 8
1.8 DfAM checks . . . 9
1.9 Inspection planning and process capability . . . 9
1.10 The release gate . . . 9
1.11 Validation summary . . . 11
1.12 Limitations . . . 11
1.13 Where I'd take it next . . . 11

1 Technical Report: Additive Build Advisor

1.1 Overview

The Additive Build Advisor is a teaching demo I built for **ES 51, Computer-Aided Machine Design**, the course I teach at Harvard SEAS, where students design a part in CAD, FFF-print it, and then machine features on the lathe and mill. It walks the **design-to-inspection digital thread** for additive manufacturing end to end, with **fused filament fabrication (FFF)** as the home process. It ingests a part as an STL mesh, recovers a watertight geometry, chooses a build orientation by resting the part on candidate flat faces, simulates the build on a voxel model, solves a finite-element **warpage analysis** (the thermal-contraction curl that lifts FFF parts off the bed), checks manufacturability (DfAM), turns the part's tolerances into an inspection plan, and assembles a single machine-readable record with an explicit release gate.

The aim is not a production build processor. It is to make each decision in the design-to-make loop transparent for students — implemented from first principles where that aids understanding and with an established library where that is the right tool — and to be a clean base I can keep extending.

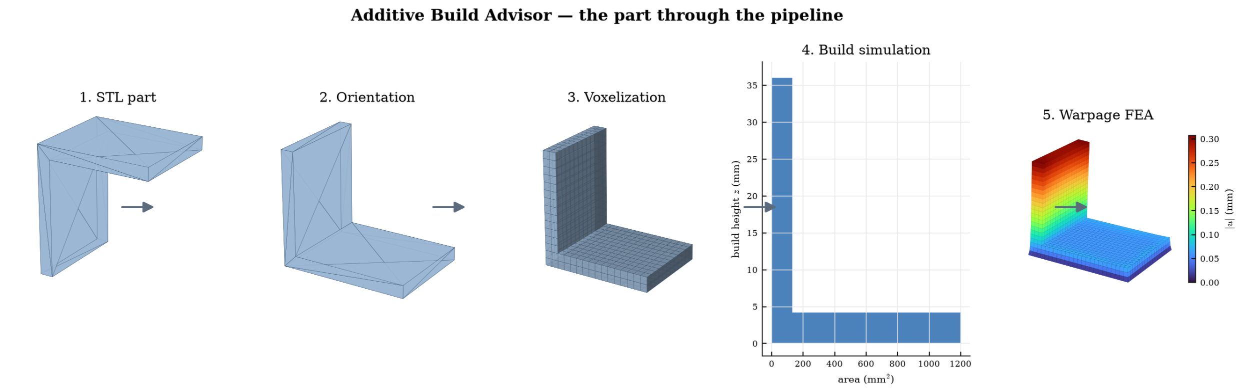

Figure 1: The part through the pipeline

The STL parser, geometry kernel, voxelizer, orientation search, and build simulation are written from scratch on `numpy`. The distortion FEA is assembled and solved with **scikit-fem** (on `scipy`); `matplotlib` renders the figures.

1.2 System architecture

The pipeline is: geometry recovery → orientation screening → voxelization → three analyses on the voxel model (build simulation, distortion FEA, DfAM) plus an inspection plan from the tolerances → a release gate → a digital-thread record that can hand off to a runtime monitoring twin.

1.3 Geometry and the voxelization engine

The STL is parsed (binary and ASCII), normals are recomputed from vertex winding rather than trusted from the file, and the mesh volume is computed analytically from the divergence theorem as a sum of signed tetrahedra over the F facets,

$$V = \frac{1}{6}\left|\sum_{f=1}^{F} \mathbf{a}_f \cdot (\mathbf{b}_f \times \mathbf{c}_f)\right|,$$

where $\mathbf{a}_f, \mathbf{b}_f, \mathbf{c}_f$ are the vertices of facet f. A watertight/manifold check welds coincident vertices and confirms every edge is shared by exactly two facets.

A triangle soup is not a solid model, so the advisor builds an occupancy grid by *ray stabbing*: for each (x, y) column it shoots a vertical ray, collects the z heights where the ray crosses the mesh, sorts them, and fills voxels between entry/exit pairs (the even–odd / crossing-number rule — a point is inside when a ray to infinity crosses the surface an odd number of times). Per-axis grid jitter keeps sample points off face diagonals, coincident crossings are de-duplicated so the pairing stays correct on shared edges, and z is sampled at voxel centres so the discretised volume is unbiased. The grid drives the support, thin-wall, trapped-void and per-layer cross-section estimates, and supplies the element mesh for the FEA.

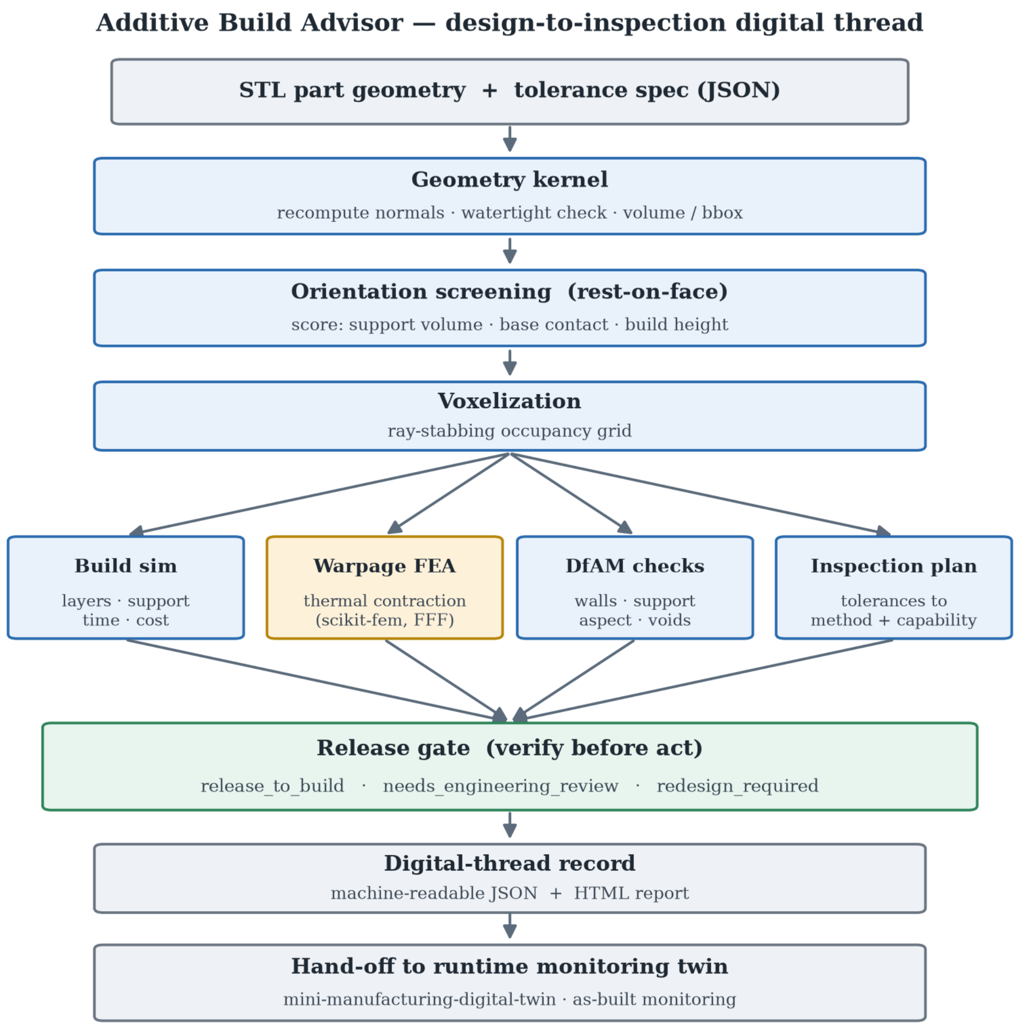

Figure 2: System diagram

1.3.1 Engine validation

The voxel/mesh volume error is reported on every run. For an axis-aligned cube (10 mm, analytic 1000 mm^3) the discretisation is exact:

grid_n	voxel volume (mm^3)	error
16	1000.00	0.00%
32	1000.00	0.00%
64	1000.00	0.00%

For an off-axis rotated bracket (analytic 8640 mm^3) it converges as the grid refines, staying under ~0.4% even when coarse (+0.18%, +0.03%, −0.02% for grid_n 24, 48, 96). A solid cube reports zero trapped volume and zero support; a hollow box with a 216 mm^3 internal void recovers ~220 mm^3 by flood fill.

1.4 Orientation: rest on a flat face

Orientation is the highest-leverage additive decision. Candidates are generated the way a machinist chooses — *which face goes down* — by clustering the mesh's facet normals into its significant flat faces and adding the six bounding-box directions as a fallback, so every candidate rests a real face on the plate.

Each candidate is screened on a coarse voxelization and scored by a normalised, weighted objective (lower is better),

$$\text{score} \;=\; w_S\,\tilde{S} \;+\; w_c\,(1-\tilde{c}) \;+\; w_H\,\tilde{H} \;+\; P, \qquad (w_S, w_c, w_H) = (0.45,\; 0.35,\; 0.20),$$

where $\tilde{S}$ is the min–max normalised **support material volume**, $\tilde{c}$ the **base-contact fraction** (footprint actually touching the plate), $\tilde{H}$ the **build height**, and P a hard penalty for orientations that exceed the build volume. On the sample bracket the winner rests on the large flat back face — full base contact and zero support — versus alternatives that need more than 5 cm^3 of support on a tiny contact patch.

A note on a pitfall I hit: an earlier version scored orientation by overhang *facet area* alone and chose a 45° edge-balanced tilt. It drove the overhang metric to zero but was physically wrong — no flat base and large support. Scoring the quantity that actually costs (support volume) plus base contact fixed it. A metric that is cheap to game is the wrong metric.

1.5 Build simulation

From the oriented mesh and grid, the simulation estimates the layer count and build time,

$$N_{\text{layers}} = \left\lceil \frac{H}{h} \right\rceil, \qquad t \;=\; \frac{V_{\text{part}} + V_{\text{support}}}{\dot{V}} \;+\; N_{\text{layers}}\, t_{\text{recoat}},$$

with H the build height, h the layer height, $\dot{V}$ the effective deposition/fusion rate, and t_{recoat} the per-layer overhead; cost is material mass times unit cost plus machine time times machine rate.

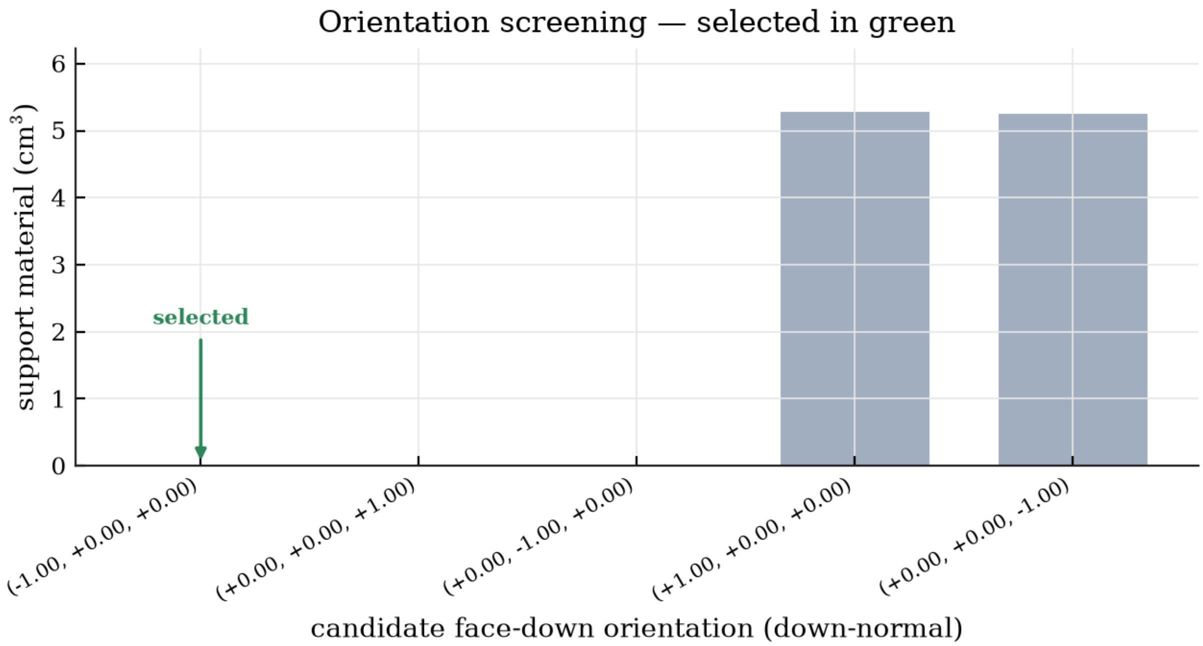

Figure 3: Orientation screening

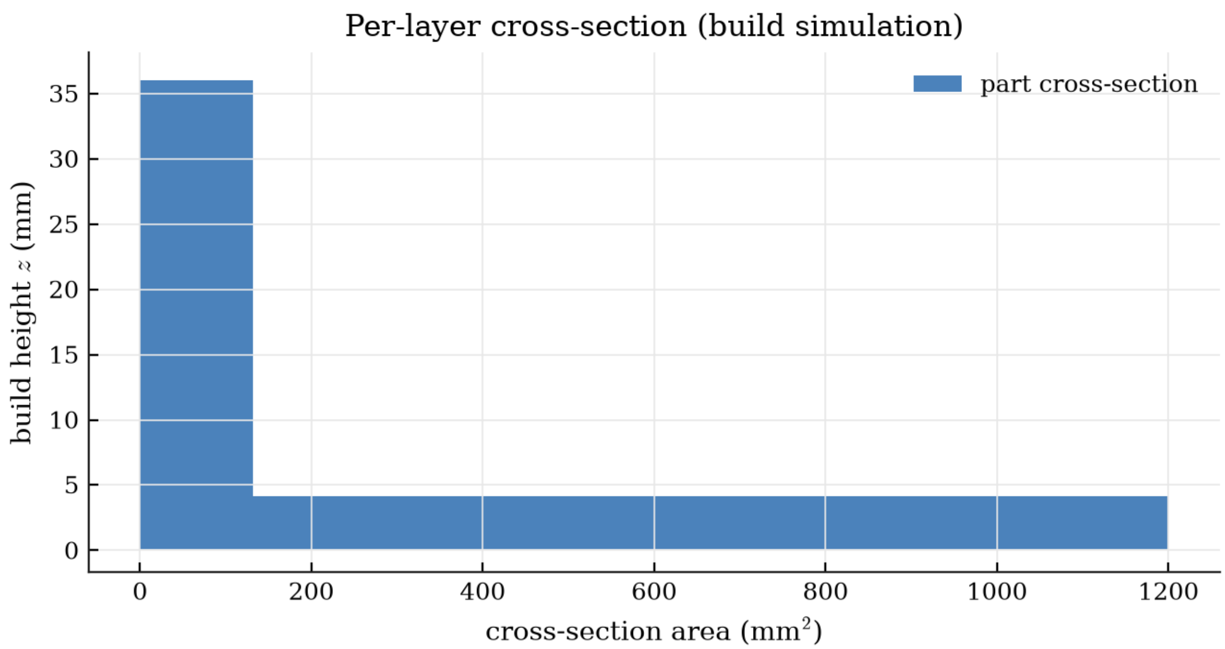

Figure 4: Per-layer cross-section

Support material is the support infill fraction times the empty volume that sits beneath solid. The per-layer cross-section comes straight from the voxel grid:

These scale correctly across processes: the same bracket is ~180 layers and a few dollars on FFF but ~720 layers and roughly $18 on SLA, driven by the finer layer height and the higher machine rate.

1.6 Warpage FEA: the thermal-contraction method

Warpage is predicted with a genuine finite-element solve, assembled and solved with **scikit-fem** on a hexahedral mesh of 8-node trilinear elements (good for bending, unlike linear tetrahedra). The constitutive law carries an eigenstrain ε^* (the thermal-contraction strain that lumps the part's cooling shrinkage, $\varepsilon^* \approx -\alpha\,\Delta T$),

$$\sigma = \mathbf{C} : (\varepsilon - \varepsilon^*)\,, \qquad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}.$$

The weak form is: find $\mathbf{u}$ such that for all test fields $\mathbf{v}$,

$$\int_\Omega \varepsilon(\mathbf{v}) : \mathbf{C} : \varepsilon(\mathbf{u})\, dV \;=\; \int_\Omega \varepsilon(\mathbf{v}) : \mathbf{C} : \varepsilon^*\, dV.$$

For an isotropic eigenstrain $\varepsilon^* = \varepsilon^*\mathbf{I}$ the eigenstress is hydrostatic and the consistent load reduces to a divergence term,

$$\sigma_0 = (3\lambda + 2\mu)\,\varepsilon^*, \qquad \mathbf{f} = \int_\Omega \sigma_0\, \nabla\cdot\mathbf{v}\, dV,$$

giving the discrete system $\mathbf{K}\mathbf{u} = \mathbf{f}$, with the first layer clamped to the bed ($\mathbf{u} = \mathbf{0}$ at $z = 0$ — the bed-adhesion constraint) and the sparse solve performed by SciPy. The displacement field is the predicted distortion; clamping the base while the bulk contracts reproduces the corner-lift that warps FFF parts off the bed. The element von Mises stress,

$$\sigma_{vM} = \sqrt{\tfrac{1}{2}\left[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2\right] + 3\left(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2\right)},$$

is recovered for context. The result is drawn as a deformed element mesh (exaggerated for visibility), contour-coloured by displacement magnitude:

A useful property: for an eigenstrain-only load (no external force) both $\mathbf{K}$ and $\mathbf{f}$ scale linearly with E, so the displacement $\mathbf{u} = \mathbf{K}^{-1}\mathbf{f}$ is **independent of Young's modulus** — distortion is governed by geometry, the contraction strain, and Poisson's ratio. E only enters the stress. That peak stress is linear-elastic and *indicative* only: with no plasticity it can exceed yield, where a real part would yield and stress-relieve. The distortion field is the meaningful output.

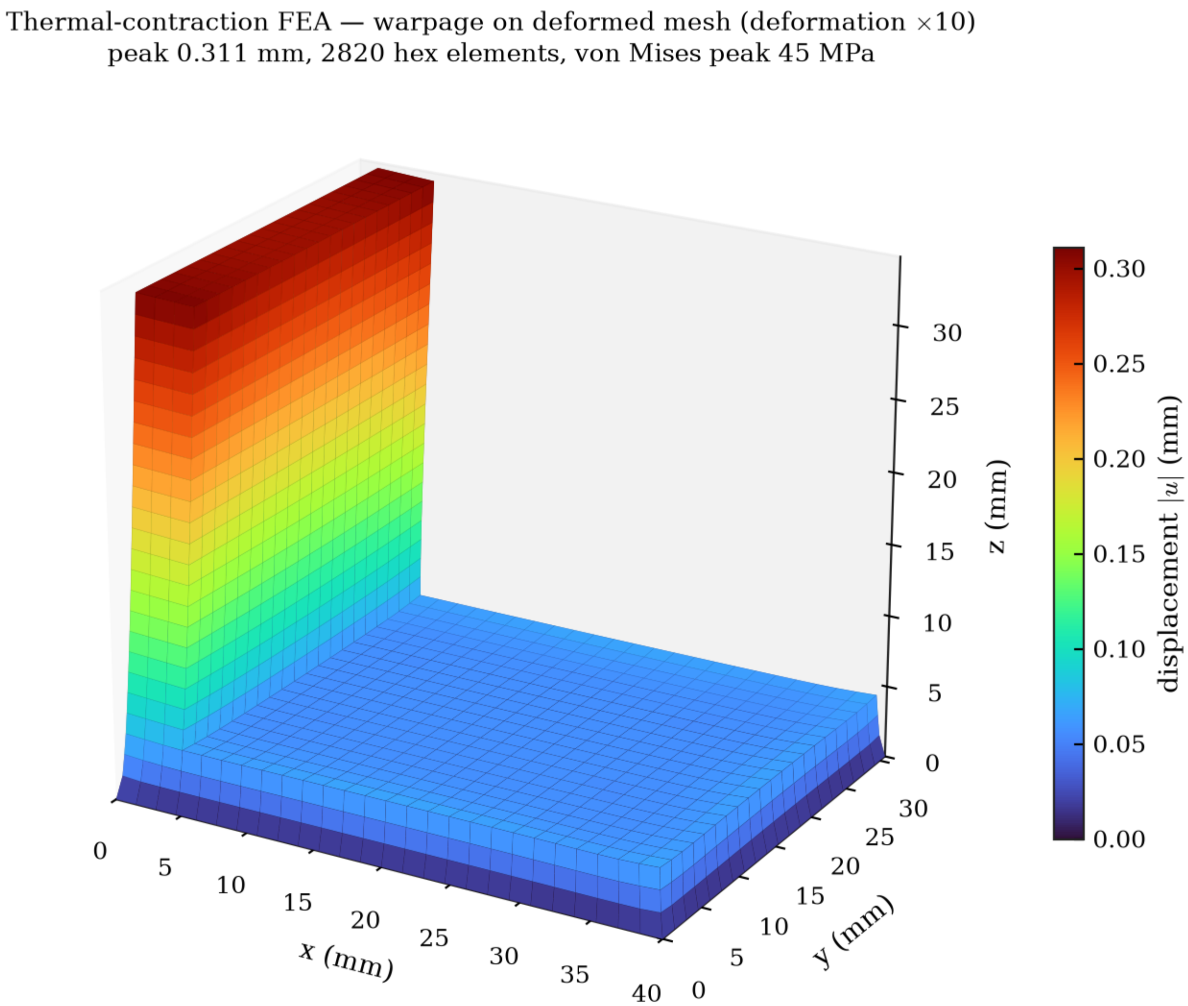

Figure 5: Thermal-contraction warpage FEA

1.6.1 Process focus and basis

The home process is **fused filament fabrication (FFF)**, where warping off the bed is the dominant geometric defect: as each extruded road cools from the printing temperature it contracts, the already-solid material below resists it, residual stress builds, and the part curls up at its corners (worst on large flat footprints, and far worse for ABS than PLA). The reduced-order recipe — lump the cooling into one effective contraction strain $\varepsilon^* \approx -\alpha\,\Delta T$ and apply it as a static eigenstrain load to a part-scale linear-elastic FEA — is a standard way to screen build warpage without a full transient thermo-mechanical solve. The repo's `cantilever_benchmark` (a long, flat ABS bar) is the worst-case warp geometry it is exercised on.

1.6.2 What the number means: on-bed vs after-removal

This model holds the part **bonded to the bed** and reports the *on-bed* distortion field — the warpage that has built up while the part is still stuck down. That is deliberately *not* the same as the **after-removal spring-back**: once the part is peeled off the bed, stored residual stress relaxes and the free part curls further. The on-bed field is a useful *relative* warpage screen (which geometries and materials warp more), not a prediction of the final free-part deflection. Capturing that needs a bed-release step and a contraction strain calibrated to a measured cooling history — see "Where I'd take it next."

1.6.3 FEA validation

A prismatic bar clamped at the base under a uniform eigenstrain has an analytical top displacement $u_{\mathrm{top}} = |\varepsilon^*|\,H$. Refining the mesh, the FEA converges toward it:

voxel pitch (mm)	FEA peak distortion (mm)	error
4.0	0.4213	+5.3%
2.0	0.4161	+4.0%
1.0	0.4141	+3.5%
0.5	0.4132	+3.3%

(analytic axial value 0.400 mm for $\varepsilon^* = -0.01$, $H = 40$ mm). The residual few percent is the lateral corner motion: the reported metric is the peak displacement *magnitude*, which at the top corner adds in-plane shrinkage to the axial component; the axial component itself converges to 0.400 mm. Running the same bracket across every process confirms the displacement is linear in the eigenstrain and independent of E — exactly what linear elasticity requires for an eigenstrain-only load:

1.7 Cross-process comparison

The build simulation, cost/time, DfAM, and warpage FEA run natively for every process profile. FFF is the home process; SLA and SLS are shown as a cross-process comparison. Running the same bracket through three additive processes:

Process	Build time	Cost	Layers	Warpage FEA
FFF (PLA)	0.80 h	\$4.24	180	0.326 mm
SLA (resin)	2.12 h	\$18.15	720	0.166 mm

Process	Build time	Cost	Layers	Warpage FEA
SLS (PA12)	1.38 h	$35.02	360	0.221 mm

FFF is the fastest and cheapest; SLA gives the finest layers; SLS is the priciest here (PA12 powder plus machine rate). The warpage ordering follows each process's representative contraction strain (and is E-independent) — so among these PLA warps the most and SLA the least, and ABS (not shown) would warp more still, as it does in practice.

1.8 DfAM checks

Transparent, severity-ranked manufacturability rules read from the same grid, simulation, and FEA as the headline numbers:

Check	Method
Build-volume fit	bounding box vs machine envelope
Thin walls	morphological opening at the min-wall radius
Support burden	support material as a fraction of part volume
Aspect ratio	height / min footprint dimension
Trapped volume	flood fill from outside; unreachable voids are enclosed
Distortion	FEA peak displacement $\delta_{\max}$ as a fraction $r = \delta_{\max}/\max(\ell_x, \ell_y, \ell_z)$

1.9 Inspection planning and process capability

The planner turns each toleranced dimension, GD&T control, and surface-finish requirement into an inspection step, choosing a method by how tight the tolerance is (calipers → micrometer → CMM → CT) and checking each tolerance against the process's **as-built capability**. A tolerance below as-built capability is flagged as needing post-machining, because inspecting to an impossible tolerance only confirms a guaranteed failure. The tolerance spec is plain JSON, so the design data stays CAD-neutral (a Fusion or STEP exporter would populate the same fields).

1.10 The release gate

The advisor never silently approves a build. It returns one of three decisions with reasons and blocking findings attached:

Decision	When
`release_to_build`	DfAM clean, tolerances within capability, simulation validated
`needs_engineering_review`	warnings, tolerances needing post-machining, or low simulation confidence (unsealed mesh / coarse-grid volume error)
`redesign_required`	a critical DfAM finding (does not fit, traps resin/powder, severe thin walls)

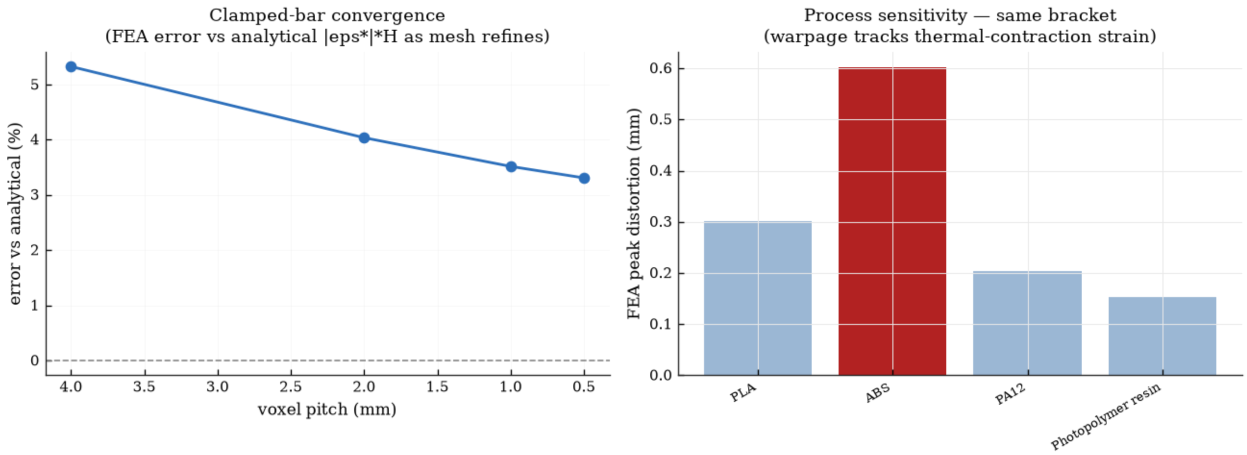

Figure 6: FEA validation

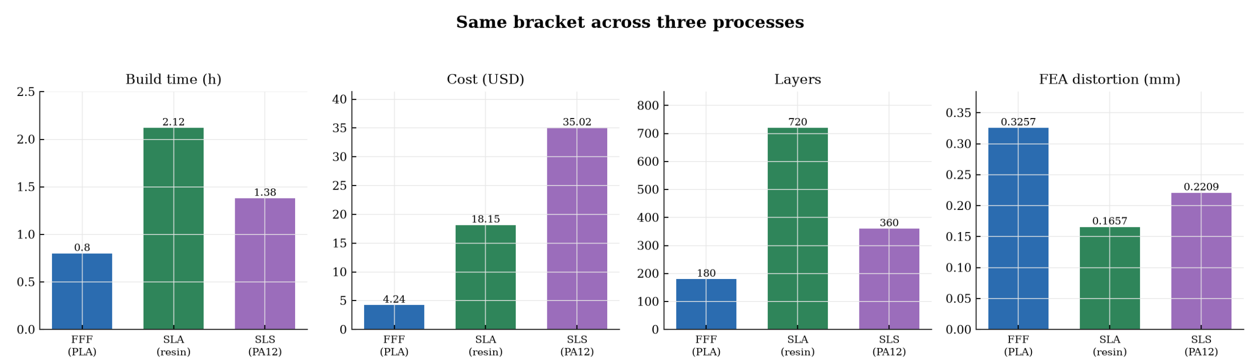

Figure 7: Cross-process comparison

A model may recommend, but a physical action is gated on evidence, on the confidence of the model, and on a human-review path when anything is uncertain. The record carries an explicit hand-off block so design intent can flow into as-built monitoring. Worked outcomes from the sample run: **calibration_cube / FFF** → `release_to_build`; **gantry_bracket / FFF** → `needs_engineering_review` (±0.05 mm and 3.2 µm are below FFF capability → route to machining); **hollow_housing / SLA** → `redesign_required` (enclosed cavity traps resin → add drain holes).

1.11 Validation summary

```
pytest                              # 10 tests
python examples/validate_fea.py     # regenerates the FEA validation figure
```

The tests cover: sample parts are watertight; STL round-trips; the cube discretises exactly; an off-axis part's volume converges; trapped volume is detected; the chosen orientation rests on a real face and is not the highest-support candidate; the FEA matches the analytical clamped-bar solution within 10%; distortion appears in the record; all three gate outcomes fire; and the record is JSON-clean.

1.12 Limitations

Stated plainly, because the value here is the workflow and the engineering, not production fidelity. The tool does **not** include:

- A true slicer / toolpath generator (build time is volume- and layer-based).
- A calibrated transient thermo-mechanical solve. The FEA is a real linear-elastic solve, but the contraction strain is a representative per-process value, not fit to a measured cooling history.
- OEM-qualified material/machine profiles (numbers are representative defaults).
- A real CAD/CAM integration (geometry is STL; tolerances are JSON).
- Lattice/infill modeling, multi-part nesting, or layer-by-layer activation in the FEA (the eigenstrain is applied to the whole part at once).
- Validation against measured build outcomes.

1.13 Where I'd take it next

1. A real slicer with toolpath-length-based time and proper support generation.
2. Layer-by-layer element activation and a contraction strain calibrated to measured cooling (or a transient thermo-mechanical solve), validated against measured warpage on printed parts, including a bed-release step so the after-removal spring-back can be predicted, not just the on-bed field.
3. Qualified per-machine process profiles and measured-vs-predicted calibration.
4. CAD/CAM integration (Fusion / STEP) to pull features, datums and PMI directly into the same record schema.
5. Closed-loop feedback: as-built inspection and in-situ monitoring updating the capability data and the eigenstrain calibration.
6. A persisted thread store linking design revision → build → inspection → part.