



**ANNUAL**  
CONFERENCE & EXPO 2026

# FusionCut: B-Rep Based and Cloud-Ready Cutter–Workpiece Engagement for Virtual Machining

IISE 2026 Annual Conference | Abstract ID: 16549



Hasan Sinan Bank<sup>1</sup> Nevzat Bircan Bugdayci<sup>2</sup>

<sup>1</sup>Department of Systems Engineering, Colorado State University

<sup>2</sup>Department of Mechanical Engineering, Michigan State University

IISE Annual Conference & Expo 2026



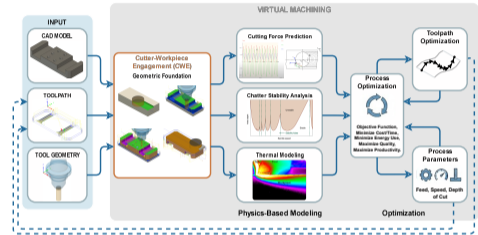
[iise.org/annual](https://iise.org/annual)

[#IISEANNUAL2026](https://twitter.com/IISEANNUAL2026)

Section 1

# Introduction

- **CWE** = instantaneous contact geometry between tool and in-process workpiece (IPW)
- Critical link between geometric modeling and physical process prediction
- Directly drives downstream models for:
  - ▶ cutting force prediction
  - ▶ chatter stability analysis
  - ▶ thermal effects
- An error here compromises the entire predictive chain.



Two prevailing paths in CWE modeling — and a forced choice.

## B-Rep solid modeling

- + Mathematically exact, “ground truth” geometry
- + Same kernel as commercial CAD/CAM
- – Locked in proprietary kernels (e.g., Parasolid)
- – Reproducibility crisis

## Discrete methods (voxel, dexel, mesh)

- + Open libraries, GPU-friendly, algorithmically general
- + Object-oriented implementations
- – Approximate — staircase artifacts, resolution-dependent engagement angles
- – Errors propagate into downstream physics models

## ? Research question

Can high-fidelity B-Rep CWE be *feasible*, *accessible*, and *reproducible* — without proprietary-kernel lock-in?

## Research gap

No common, **accessible** platform that delivers **high-fidelity B-Rep CWE** with the reproducibility needed for collaborative research.

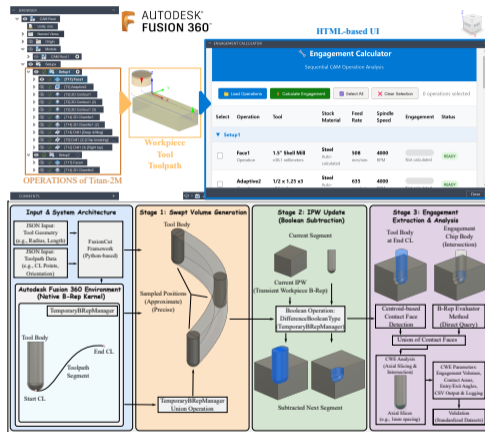
### **Our contribution** — **FusionCut:**

- Reproducible B-Rep CWE framework built on a modern, freely accessible CAD/CAM API (Autodesk Fusion 360)
- Validated end-to-end on **public** datasets (Titans of CNC Academy “Titan-2M”)
- Establishes a new baseline for what is computationally feasible *and* scientifically reproducible for high-fidelity virtual machining

Section 2

# Method

- **Decoupled** frontend / backend
- Frontend: tool + toolpath as JSON (web-accessible)
- Backend: Python script driving the **Fusion 360 API**
- Output: per-CL CWE parameters as CSV
- JSON-in / CSV-out backend runs on any machine — local or cloud-hosted; *containerizable* for headless deployment in future scalable rollouts



Per toolpath segment ( $CL_i \rightarrow CL_{i+1}$ ):

1. **Swept volume generation** — sample tool positions along the segment, union via TemporaryBRepManager
2. **IPW update** — Boolean Difference of swept volume from the running in-process workpiece body
3. **Engagement extraction** — intersect tool body at  $CL_{i+1}$  with the updated IPW; extract entry/exit angles by axial slicing

## Sampling step

$$\Delta s = \frac{R_{\text{tool}}}{k}, \quad k = 2 \text{ (default)}, \quad n_{\text{samples}} = \lceil L_{\text{seg}} / \Delta s \rceil + 1$$

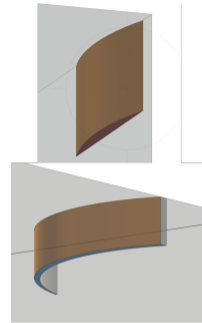
Denser sampling ( $k=10$ ) for non-cylindrical tools (chamfer, ball, drill).

## Method 1 — centroid-based

- Compares engagement-chip face centroids against IPW face centroids
- Workpiece-origin faces excluded by match
- **Fails when** Boolean ops split/merge faces  $\Rightarrow$  centroid drifts

## Method 2 — B-Rep evaluator

- Queries each face's `pointOnFace` against the tool body's surface evaluator
- **Fails when** the *infinite* surface extension matches a non-contact face, or near edges/vertices



Two cases for two methods. **Red**: detected by both. **Orange**: centroid-only. **Cyan**: evaluator-only.

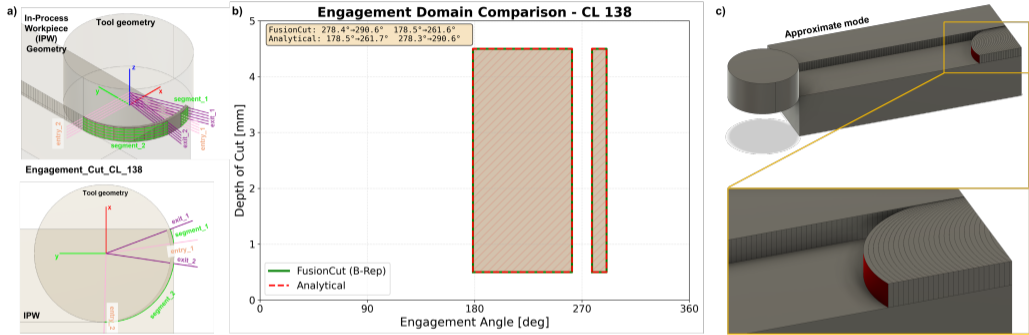
Section 3

# Results

Titan-2M Setup 1 — 1 mm vs 0.2 mm axial sampling

| Operation                      | Proc. / Sched. CLs | Total time (s) | Avg / proc. CL (ms) |
|--------------------------------|--------------------|----------------|---------------------|
| <i>Axial sampling = 1 mm</i>   |                    |                |                     |
| Setup 1 (total)                | 19,460 / 36,695    | 5,912.9        | 303.85              |
| Face1                          | 352 / 361          | 59.1           | 168.37              |
| Adaptive2                      | 18,320 / 35,097    | 5,580          | 304.60              |
| 2D Contour1                    | 388 / 451          | 139            | 308.89              |
| <i>Axial sampling = 0.2 mm</i> |                    |                |                     |
| Setup 1 (total)                | 19,460 / 36,695    | 10,073         | 517.62              |
| Face1                          | 352 / 361          | 131            | 373.21              |
| Adaptive2                      | 18,320 / 35,097    | 9,480          | 517.50              |
| 2D Contour1                    | 388 / 451          | 270            | 600.00              |

- Same order of magnitude as Boz et al. (2015)'s dexelfield ( $\sim 528$  ms/CL at 0.2 mm)
- 0.2 mm vs 1 mm:  $\sim 1.7\times$  runtime for  $5\times$  finer resolution (negligible accuracy gain on tested geometry)
- Skipped CLs are non-cutting moves (rapids, retracts) — no IPW change, no geometric error



Titan-2M face milling: (top) 3D iso view, tool + IPW + engagement segments; (bottom) 2D top view with engagement angles on the tool circumference.

Section 4

# Conclusion

## Key result

The fidelity-vs-accessibility dichotomy is **not fundamental** — it is a consequence of implementation choices.

- **Geometric fidelity** of traditional B-Rep methods, validated against an analytical ground truth ( $\Delta\theta < 0.25^\circ$ )
- **Practical computational performance** — same order of magnitude as discrete dexelfield methods
- **Fully accessible, fully reproducible** (Fusion 360 API + open Titan-2M dataset)
- **Cloud-ready by construction** — decoupled JSON-in / CSV-out backend

## Current Limitations

- Bound to the **Fusion 360** platform — no kernel-agnostic backend yet
- Sequential per-segment processing
- Python + API overhead vs direct kernel access
- Cross-method comparison constrained by non-identical literature benchmarks

## Future Work

1. Standardized open benchmark suite for cross-method comparison
2. C++ implementation to reduce API overhead
3. Experimental validation via cutting-force measurements
4. 5-axis toolpaths and complex tool geometries (adaptive sampling on high-curvature paths)



**ANNUAL**  
CONFERENCE & EXPO 2026

**Q&A**

[sinan.bank@colostate.edu](mailto:sinan.bank@colostate.edu)

*Remember to complete your evaluation for this session within the app!*

[iise.org/annual](https://iise.org/annual) #IISEANNUAL2026