From 3D vegetation models to physics-based wildfire simulation: a fractal barycentric method for lagrangian particle initialization in FDS/WFDS

David Caballero*[1]

[1]*MeteoGrid. Modesto Lafuente 45, 28010, Madrid, Spain*

**Corresponding author. Email: david@meteogrid.com*

Keywords

Wildfire simulation; FDS/WFDS; Lagrangian particles; 3D vegetation; procedural content generation;

Abstract

This paper presents a novel pipeline for translating three-dimensional (3D) vegetation models generated through procedural content generation in Unreal Engine 5 (UE5) into spatially explicit Lagrangian particle distributions for physics-based wildfire simulation in the Fire Dynamics Simulator (FDS/WFDS). The pipeline is implemented end-to-end as a native UE5 C++ plugin, the FDS Vegetation Exporter, which performs fractal barycentric discretization of PCG foliage geometry, generates FDS case files, terrain obstacle files, and SLURM job scripts, and submits jobs to EuroHPC clusters without leaving the UE5 editor. The core scientific contribution is the fractal barycentric discretization method, which recursively subdivides triangular facets of the vegetation mesh until the sub-triangle area is comparable to the FDS grid cell face area, then uses each sub-triangle barycenter to assign combustible material presence to individual simulation voxels. A single Lagrangian particle per voxel per fuel type is initialized with species-specific MASS_PER_VOLUME and SURFACE_VOLUME_RATIO parameters, following the WFDS boundary fuel model. The method is geometrically scalable: the same species parameters apply at any grid resolution, from 5 cm (individual tree) to 50 cm (landscape scale). The pipeline was functionally verified through an individual-tree combustion test on LUMI EuroHPC supercomputer and a one-hectare terrain and vegetation simulation on MeluXina cluster. The approach represents the first published method for deriving FDS Lagrangian particle distributions directly from arbitrary 3D tree architecture within a real-time rendering engine, without reliance on statistical crown approximations.

PREPRINT

1. Introduction

Physics-based wildfire simulation with the Fire Dynamics Simulator (FDS) and its wildland-urban interface extension WFDS has established itself as the leading tool for high-fidelity combustion modeling in vegetated environments (Mell et al., 2007, 2010). The Lagrangian particle model introduced by Mell et al. represents solid fuel elements (pine needles, grass blades, fine dead fuels) as clouds of particles whose thermophysical properties govern pyrolysis, combustion, moisture evaporation, convective heat transfer, and aerodynamic drag. Each particle carries a MASS_PER_VOLUME density and a SURFACE_VOLUME_RATIO that collectively encode the bulk properties of hundreds or thousands of real fuel elements within a single computational cell.

A fundamental limitation of existing WFDS vegetation models is their reliance on statistical crown descriptions: trees are represented as uniform cylinders, cones, or cuboids filled homogeneously with combustible material at a prescribed bulk density (Mell, 2009). This approach, while computationally convenient, fails to capture the spatial heterogeneity of real canopy architecture — the denser outer crown, the sparser interior, the fuel-free trunk zone, and the geometry-dependent shadowing that governs ignition spread. Remote sensing approaches using airborne LiDAR have attempted to derive more realistic fuel distributions at stand scale (Mell et al., 2011), but these methods lack the geometric resolution needed for individual-tree fire dynamics.

Parallel developments in real-time rendering have produced extraordinarily detailed procedural vegetation models in game engines such as Unreal Engine 5 (UE5). The Procedural Content Generation (PCG) framework in UE5 enables the generation of botanically realistic tree architectures at scale, with per-facet geometric detail at millimeter resolution. These models, however, have never been connected to physics-based fire simulation.

This paper closes that gap with a complete, production-ready pipeline from UE5 procedural vegetation geometry to FDS/WFDS Lagrangian particle initialization, implemented within the HIDALGO2 EuroHPC project. The pipeline introduces a fractal barycentric discretization method that correctly maps 3D triangulated geometry onto the FDS computational grid, a geometrically scalable voxel occupancy criterion grounded in the physics of the WFDS boundary fuel model, a native UE5 C++ plugin that performs the full conversion and cluster submission workflow, and a set of successfully executed cases on LUMI (CSC, Finland) and MeluXina (LuxProvide, Luxembourg).

2. Objectives

This paper aims to develop a method for translating arbitrary 3D vegetation geometry generated procedurally in Unreal Engine 5 into spatially explicit Lagrangian particle distributions for FDS/WFDS, based on a physically consistent voxel occupancy criterion derived from barycentric sampling with fractal triangle subdivision. The geometric scalability of the method is demonstrated across simulation resolutions ranging from 5 cm to 50 cm grid spacing, covering individual-tree combustion to landscape-scale crown fire propagation. The pipeline is implemented as a native UE5 C++ plugin and functionally verified through individual-tree and landscape-scale simulations executed on EuroHPC infrastructure. Finally, the approximation error of the fractal method is characterized relative to exact triangle-grid intersection, and the valid resolution range of the boundary fuel model is identified in terms of the relationship between cell size and the characteristic dimensions of the individual fuel element.

3. Background

The HIDALGO2 project (EuroHPC JU grant 101093590, https://www.hidalgo2.eu/) addresses the application of high-performance computing (HPC) to current environmental challenges, with wildland fire behavior as one of its primary use cases. The project adopts a multi-scale modeling strategy that spans regional fire-atmosphere coupling with WRF-SFIRE, local-scale physics-based combustion with FDS/WFDS (Mell et al. 2007, 2009), and fuel-bed pyrolysis with OpenFOAM porousGasificationFoam (Zuk et al., 2022). For the simulation of fire behavior in the wildland-urban interface (WUI), FDS/WFDS has been selected as the primary tool, owing to its rigorous treatment of turbulent combustion, radiative heat transfer, and Lagrangian solid fuel dynamics, and its extensive validation against grassland, shrubland, and WUI fire experiments (Mell et al. 2009, 2010).

A persistent limitation of current FDS/WFDS applications to WUI fire is the reliance on idealized vegetation descriptions. Tree crowns are typically represented as uniform cylinders, cones, or ellipsoids populated homogeneously with combustible material at a prescribed bulk density (Mell et al., 2009; Parsons et al. 2011), an approximation that neglects the spatial heterogeneity of real canopy architecture: the denser outer shell, the sparser interior, the fuel-free trunk zone, and the geometry-dependent radiative shadowing that governs ignition spread. Within HIDALGO2, this limitation is addressed through the integration of UE5 Procedural Content Generation (PCG) as a vegetation modeling tool. PCG enables the generation of botanically realistic forest stands with multiple vegetation strata, species mixing, and individual-tree geometric detail, exportable as triangulated 3D meshes. The pipeline presented in this paper exploits this capability to derive spatially explicit fuel distributions for FDS/WFDS directly from procedural 3D vegetation geometry, replacing statistical crown approximations with architecture-faithful voxel occupancy maps.

In FDS/WFDS, solid vegetative fuel is represented through the boundary fuel model originally described by Mell et al. (Mell et al., 2006). A Lagrangian particle of class k occupying a computational cell of volume V carries a combustible mass $M = MPV_k \times V$, where MPV_k (MASS_PER_VOLUME, kg/m³) is a species-specific bulk density. The total fuel surface area available for convective and radiative exchange within the cell is computed internally by FDS as:

$$A_{total} = M \times SVR_k / \rho_{solid}$$

where SVR_k (SURFACE_VOLUME_RATIO, m^{-1}) is the surface-to-volume ratio of an individual fuel element, and ρ_{solid} is the density of the solid material (500 kg/m³ for generic vegetation). The packing ratio $\beta = M / (\rho_{solid} \times V) = MPV_k / \rho_{solid}$ is the fraction of the cell volume occupied by solid

PREPRINT

material. For typical forest fuels with MPV = 0.3–2.0 kg/m³, beta is of order 10^{-3}, confirming that the cell is overwhelmingly air, the fundamental assumption of the continuum model. Critically, both MPV and SVR are intensive properties of the fuel species: they do not depend on cell size. This is the physical basis for the geometric scalability of the method demonstrated in this work.

FDS source code analysis (read.f90, line 8184) establishes unambiguously that MOISTURE_CONTENT is interpreted on a dry-weight basis:

MATL_MASS_FRACTION = MOISTURE_CONTENT / (1 + MOISTURE_CONTENT)

This is the standard dry-weight conversion from forest fuel science. A value of MOISTURE_CONTENT = 0.14 therefore corresponds to 14% dry-weight moisture content — appropriate for severely drought-stressed or recently dead foliage — not 14% wet-weight as sometimes assumed from the FDS User Guide. Live foliage of conifers under moderate moisture stress typically shows dry-weight moisture contents of 80–120%, corresponding to MOISTURE_CONTENT = 0.80–1.20 in FDS input files.

4. Material and methods

4.1. Pipeline Overview

The full pipeline from 3D vegetation geometry to FDS simulation is implemented end-to-end within the FDS Vegetation Exporter plugin for Unreal Engine 5.7, a native C++ plugin developed within the HIDALGO2 project. The plugin integrates into the UE5 editor as a dockable panel and performs all steps of the conversion, case generation, and cluster submission without requiring external tools. The pipeline proceeds through the following stages:

Procedural vegetation generation is performed in UE5 using the PCG framework, producing botanically realistic 3D tree architectures at individual-tree scale. Tree foliage is represented as static mesh components placed procedurally throughout the scene. The plugin accesses these static mesh instances directly through the UE5 engine API, reading the vertex and triangle data of LOD0 for each foliage asset via FMeshDescription::GetMeshDescription(0). World-space vertex coordinates are obtained by applying the actor world transform, yielding positions in UE5 centimeters in the left-handed UE5 coordinate system.

Fractal barycentric discretization of the foliage geometry is performed within the plugin at scan time, mapping the triangulated foliage mesh onto the FDS computational grid and generating per-voxel fuel presence maps for each fuel class. The Y-axis flip required to convert from the UE5 left-handed coordinate system to the FDS right-handed coordinate system is applied at write time only, not during voxel mapping, to preserve the geometric consistency of the discretization.

Generation of FDS input files is performed automatically. The plugin writes a complete FDS case file (.fds) containing domain decomposition (&MESH blocks), boundary conditions, reaction chemistry, ignitor definitions, velocity diagnostics, slice files, and SMOKE3D volumetric output specifications. Vegetation chemistry follows the Mell model (vegetation_model.txt), which is written as a separate file and referenced from the main case file via a FDS &CATF directive. Terrain geometry derived from a DTM mesh actor in the scene is written to a dedicated _terrain.fds file, also referenced via &CATF. Live foliage and dead fine fuel particle &INIT blocks are written to separate _foliage.fds and _dead.fds files. The correct &CATF ordering — vegetation_model.txt first, followed by terrain, foliage, and dead particle files — is enforced by the plugin, as FDS requires reaction chemistry to be resolved before any &SURF block that references those species.

SLURM job script generation is integrated into the plugin. The cluster selector (LUMI or MeluXina) controls all cluster-specific parameters automatically, including account identifiers, scratch paths, FDS binary paths, module load scripts, partition names, and job directives. For MeluXina, the plugin adds the --qos=default directive required by LuxProvide policy. The SLURM script is written to the same output directory as the FDS case files.

PREPRINT

FDS simulations are executed on EuroHPC cluster infrastructure. The plugin targets the LUMI (CSC, Finland) and MeluXina (LuxProvide, Luxembourg) EuroHPC petascale systems, using FDS 6.10.1 compiled from the master branch with the PrgEnv-gnu / cray-mpich stack on LUMI and OpenMPI 5.0.7 / GCC 14.2.0 on MeluXina. Post-processing proceeds by exporting FDS SMOKE3D output to OpenVDB sequences for real-time volumetric rendering in Unreal Engine 5, closing the end-to-end loop from procedural vegetation generation to physics-based simulation and visualization within the same engine environment.

4.2. FDS Vegetation Exporter Plugin (v3.2)

The FDS Vegetation Exporter plugin for UE5.7 constitutes the production implementation of the pipeline. Version 3.2 is the current stable anchor release, functionally verified through multiple fire simulations executed on LUMI and MeluXina, including one-hectare cases.

The plugin user interface is organized as a dockable editor panel with the following functional sections. The Target Mesh section allows selection of the UE5 static mesh asset representing the foliage component and the foliage slot (material slot) to be processed. The FDS Domain section configures grid resolution (cell size in meters), domain extents (X, Y, Z in meters), and displays the resulting cell count and mesh decomposition. The Live Foliage and Dead Fine Particles sections configure the physical parameters for each fuel class: PART_ID, MASS_PER_VOLUME (kg/m³), SURFACE_VOLUME_RATIO (m²/m³), moisture content, and dead fraction. The Output section defines the case identifier (CHID) and output directory. The Simulation section configures T_END, ambient temperature, wind speed, and wind direction. The Ignitor section configures ignition surface parameters, heat release rate, temperature front, ramp timing, and particle count. The Terrain (MDT) section activates terrain export and selects the UE5 static mesh actor representing the Digital Terrain Model (DTM), with a Z reference altitude and a viewport preview button. The HPC / Cluster section selects the target cluster (LUMI or MeluXina), partition, walltime, target cells per rank, and cores per node. The Preview section controls the SAMPLING_FACTOR for Smokeview visualization output.

Four action buttons are provided at the bottom of the panel: Scan Scene performs a full scan of the UE5 scene and computes the voxel count estimate for the current domain and resolution; Preview Voxels renders the fractal-discretized voxels as a 3D point cloud in the UE5 editor viewport via a transient *ProceduralMeshComponent* actor; Export INIT writes only the _foliage.fds and _dead.fds particle files; and Export Full Case performs the complete export including all FDS files, vegetation_model.txt, the terrain file (if enabled), and the SLURM script.

The target cells per MPI rank parameter (*TargetCellsPerMesh*) is set to 12,500 cells/core in v3.2, based on the results of a formal strong scaling study on LUMI with a 32-million-cell reference domain, at which point FDS achieves near-optimal MPI scalability before inter-node communication overhead becomes dominant (HIDALGO2 Strong Scaling Technical Report, April 2026). This value remains user-configurable for scaling studies.

4.3. Terrain Rasterization: the FDSTerrainSampler

A terrain rasterization module, *FDSTerrainSampler*, converts a terrain static mesh (Digital Terrain Model, DTM) placed in the UE5 scene into FDS obstacle geometry. All triangles of LOD0 of the terrain mesh are extracted via FMeshDescription::GetMeshDescription(0), applying the actor world transform to obtain world-space vertices in UE5 centimeters. For each column (i, j) of the FDS domain grid, a vertical ray is cast upward using the Möller–Trumbore intersection algorithm, which accepts both triangle winding orders — a requirement because UE5 uses a left-handed coordinate system in which upward-facing triangles produce a negative Z component in the face normal (opposite to a right-handed system). The maximum Z intersection across all triangles is recorded as the terrain height for that column, and the corresponding FDS obstacle is a rectangular column extending from the Z reference altitude up to the sampled terrain height. Terrain obstacles use an adiabatic surface boundary condition (&SURF ID='ADIABATIC', ADIABATIC=.TRUE.) with no heat flux through the ground surface.

A critical bug identified during development was a pre-filter that discarded triangles with Normal.Z <= 0, intended to reject downward-facing faces. In UE5's left-handed system this filter rejected all terrain triangles.

PREPRINT

The fix removes the pre-filter entirely, relying on the Möller–Trumbore determinant sign test to handle winding order implicitly. The FDS &GEOM ZVALS approach, which would eliminate the staircase effect of column-by-column rasterization via an immersed boundary formulation, was investigated but requires FDS compiled with Intel MKL PARDISO, unavailable in the GCC/OpenMPI builds on both LUMI and MeluXina. The staircase terrain effect is acknowledged as a current limitation of v3.2.

4.4. Fractal Barycentric Discretization

The term fractal refers here to the recursive self-similar subdivision of each triangular facet into four congruent sub-triangles, a process that terminates when the sub-triangle area reaches the target cell face area. In the triangulated representation of three-dimensional objects such as trees, facet areas may vary by one to two orders of magnitude within a single mesh. When discretizing these facets onto a uniform cubic computational grid to determine the presence or absence of combustible biomass, triangles whose area substantially exceeds the grid cell face area cannot be adequately represented by a single barycenter: the barycenter of a large triangle may fall within one cell while the triangle itself spans several adjacent cells, leaving the biomass contained in those cells undetected. This geometric mismatch between facet size and cell size motivates the fractal subdivision approach introduced in this paper, which constitutes the central methodological contribution of this work. The method proceeds as follows.

For each triangular facet of the UE5 Foliage mesh instance, the algorithm recursively subdivides the triangle into four congruent sub-triangles by connecting the midpoints of each edge. Subdivision continues until the area of the resulting sub-triangle (A_sub) satisfies:

$$A_sub <= A_target = cell_size^2$$

where *cell_size* is the FDS grid spacing and *A_target* is the reference target area. This area target corresponds to the face area of a cubic voxel, ensuring that each sub-triangle is statistically contained within a single grid cell. The barycenter of each sub-triangle is then computed and mapped to its containing voxel by floor-quantization of the metric coordinates.

A voxel occupancy criterion is then applied: a voxel is marked as occupied by fuel class *k* if and only if at least one barycenter of class *k* falls within it. Subsequent barycenters of the same class falling in an already-occupied voxel are discarded. This ensures that each occupied voxel receives exactly one &INIT entry per fuel class, with a fixed MPV_k regardless of the number of contributing barycenters.

Two fuel classes are implemented: live foliage (fraction 1 - f_dead) and dead fine fuel (fraction f_dead), where f_dead is a user-specified parameter representing the degree of tree mortality or drought stress as stated in Table 1. Assignment of barycenters to fuel classes uses a seeded random draw for reproducibility.

Table 1- Dead and live fuel fractions according to affectation scenarios.

Dead fraction	Foliage (live)	Dead fine	Tree condition	Typical scenario
0.0	100%	0%	Healthy tree	Outside fire season, no stress
0.2	80%	20%	Mild affectation	Summer drought, early insect damage
0.5	50%	50%	Moderate affectation	Severe drought, significant insect infestation
0.8	20%	80%	Severely affected	Near-dead tree, bark beetle kill
1.0	0%	100%	Completely dead	Standing dead fuel, snag

4.5. Approximation Analysis: Fractal vs. Exact Intersection

The fractal barycentric method approximates the exact geometric problem of triangle-grid intersection. A comparison of the two approaches reveals systematically different approximation characteristics that are relevant to model accuracy (Figure 1).

PREPRINT

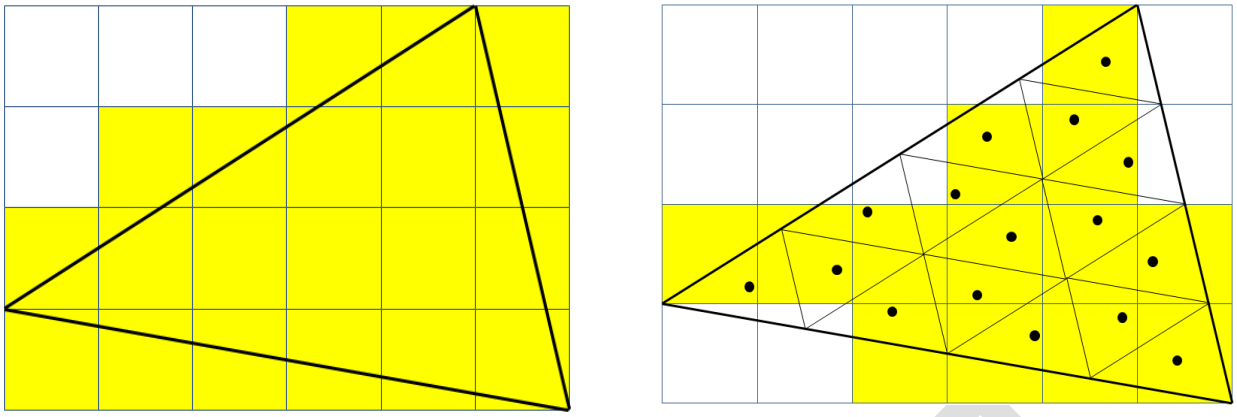


Figure 1 – Two different methods for discretizing the triangular facets of a 3D object: on the left, identification of cells containing biomass through the intersection of the facet with the computational grid. On the right, the method of successive fractal subdivision of the triangular facet and selection of barycenters as geometric descriptors of biomass presence in the corresponding grid cell.

The exact intersection method marks all grid cells touched by a triangle, regardless of the fraction of the cell area occupied by the triangle. For the reference geometry (triangle area At = 4.2 cm², cell area Ac = 0.25 cm²), the intersection method activates 20 cells with a total activated area of 5.0 cm², corresponding to a ratio Ai/At = 1.19, a systematic overestimate of 19%. This overestimation arises because border cells are fully counted even when the triangle occupies only a small fraction of their area.

The fractal method produces approximation ratios that converge from below with increasing subdivision level: 5.9% at level 0 (no subdivision), 23.8% at level 1 (4 sub-triangles), and 77.3% at level 2 (16 sub-triangles). The method is a systematic underestimate, asymptotically approaching the correct area coverage as subdivision level increases.

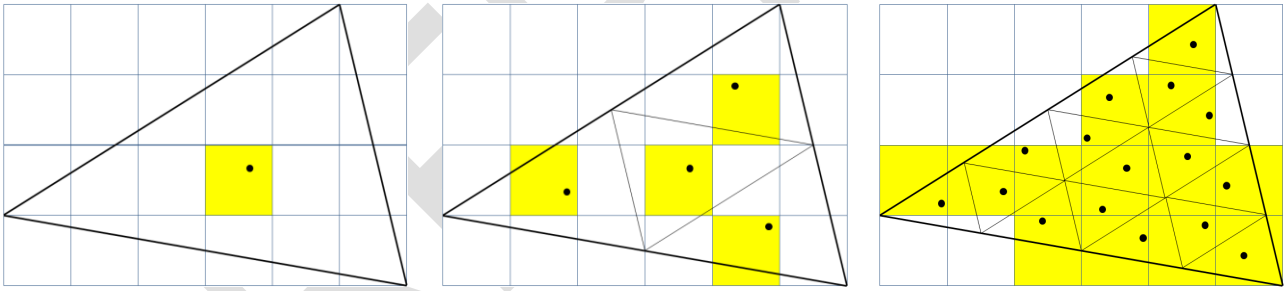


Figure 2 - Effect of successive fractal subdivision iterations (Fractal Level) on the detection rate of grid cells containing biomass. Left: without subdivision, a single barycenter fails to capture the spatial distribution of biomass across the triangle area. Center: after the first fractal iteration, four cells are activated. Right: after the second iteration, 77.3% of the triangle area is accounted for, with a conformity of 96.1% between the cell face area and the resulting sub-triangle area.

Table 2- Effect of successive fractal subdivision iterations on the detection rate of grid cells containing biomass.

FracLev	NBar	NAC	A_sub cm^2	Conf %	T_Area cm^2	Ratio %
0	1	1	4.2	5.9	0.25	5.9
1	4	4	1.05	23.8	1.0	23.8
2	16	13	0.26	96.1	3.25	77.3

PREPRINT

For forest fuel modeling, a conservative underestimate is preferable to a systematic overestimate: overestimating voxel occupancy inflates the total combustible mass in the domain. The fractal method at the target area criterion (A_sub ≈ Ac) achieves coverage ratios of 70–90% depending on triangle orientation relative to the grid, which is acceptable for the spatial resolution of FDS calculations.

4.6. Resolution Validity Range

The boundary fuel model assumes that the computational cell is a continuum mixture of air and fuel elements, with the fuel fraction described by the packing ratio beta = MPV / rho_solid. For this assumption to hold, the cell must be large enough to contain a statistically representative sample of individual fuel elements. The minimum valid cell size is estimated as:

$$\text{cell_size_min} \sim 10 \times \text{LENGTH}$$

where LENGTH is the characteristic dimension of an individual fuel element. For *Picea abies* needles (LENGTH = 15 mm, SVR = 4000 m^{-1}), this gives cell_size_min ≈ 0.15 m. The simulations presented here use cell_size = 0.05 m, which is below this conservative threshold; however, FDS produces physically reasonable results at this resolution, suggesting that the continuum assumption is more robust than the strict criterion implies.

At the upper end, the maximum useful resolution is constrained by the spatial heterogeneity of the crown architecture: beyond cell_size ≈ 0.5 m, individual branch layers cannot be distinguished and the method degrades to a coarse crown-shape approximation equivalent to WFDS original cylinder/cone model. The practical resolution range for individual-tree simulations with the present method is therefore 0.05–0.30 m.

5. Results

5.1. Plugin Functional Verification and Simulations Completed

The following simulations were successfully completed using the v3.2 plugin export pipeline, demonstrating end-to-end functionality from UE5 scene to completed HPC fire simulation (Table 3):

Table 3- Simulations completed with the FDS Vegetation Exporter v3.2 pipeline.

Case	Domain (m)	Cell size	Cells	Cluster / T_END
mdt_1ha (terrain only)	100×100×100	1 m	1,000,000	LUMI / 0.1 s
mdt_1ha + vegetation	100×100×100	0.5 m	8,000,000	MeluXina / 0.1 s
mdt_1ha + vegetation + fire	100×100×100	0.5 m	8,000,000	MeluXina / 30 s

All cases used 640 MPI ranks (8×8×10 mesh decomposition) on 5 nodes with 128 cores/node. The 30-second fire simulation on MeluXina completed successfully with stable timestep behavior. Terrain geometry (40,000 &OBST columns for the 0.5 m case) was correctly interpreted by FDS and rendered in Smokeview showing the correct topographic relief of the one-hectare domain.

5.2. Geometry Discretization of *Picea abies*

The test case uses a procedurally generated *Picea abies* tree accessed directly from the UE5 scene as a Foliage static mesh instance (6,690 vertices, 4,665 triangles; coordinate system in centimeters). The bounding box of the foliage spans X: [-1.45, 1.39] m, Y: [-1.27, 1.50] m, Z: [0.002, 3.81] m, consistent with a medium-sized Norway spruce of approximately 4 m height (Fig.3).

PREPRINT

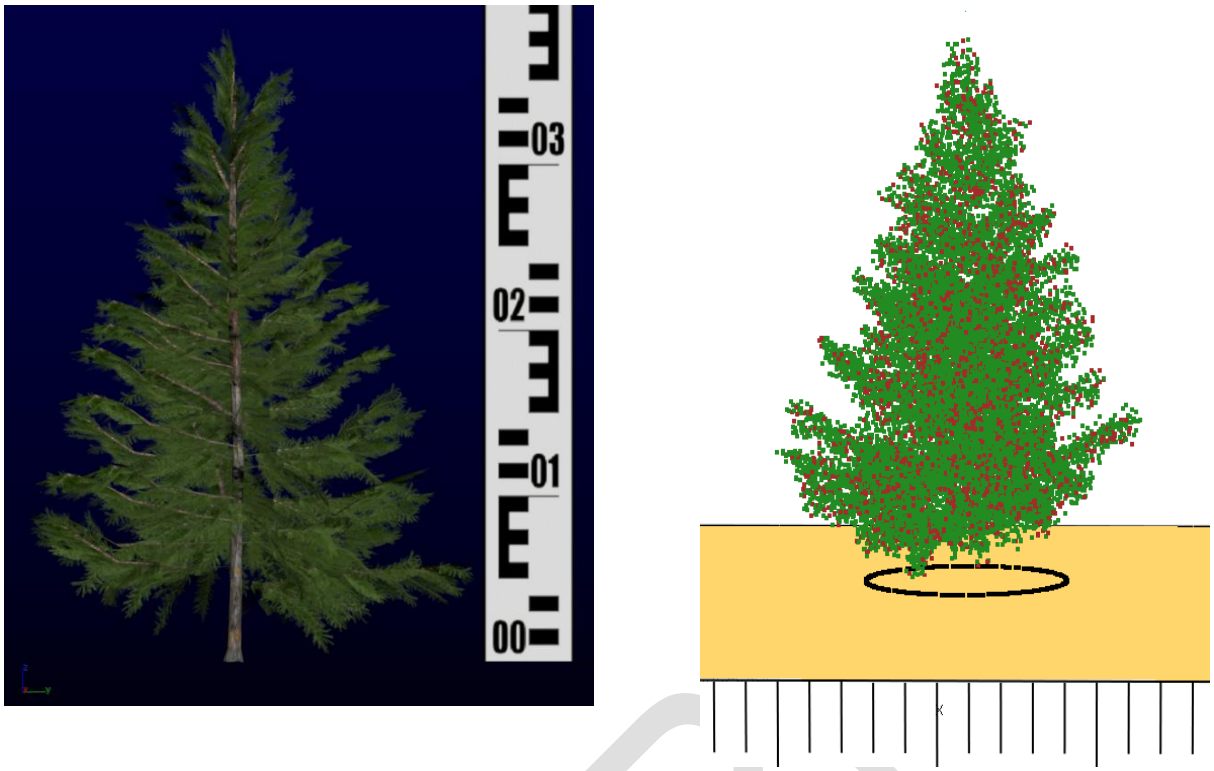


Figure 3 – Left: procedurally generated 3D model of Picea abies in Unreal Engine 5, showing the full geometric detail of the foliage architecture used as input to the pipeline. Right: the corresponding Lagrangian particle distribution in FDS/WFDS, derived from the fractal barycentric discretization of the foliage mesh at 5 cm grid resolution. Green particles represent live foliage and red particles represent dead fine fuel (dead_fraction = 0.20). The ring ignitor at the base of the crown is also visible. Total foliage volume: 18,202 occupied voxels; dry combustible mass: 3.20 kg.

The original triangulation has a mean face area of 222 cm² and a maximum face area of 2,217 cm², both substantially larger than the target cell area of 25 cm² (cell_size = 0.05 m). The fractal subdivision algorithm produces 49,248 sub-triangles from the 4,665 original faces, a mean subdivision factor of 10.6×. The resulting barycentric point cloud, after application of the voxel occupancy criterion, yields 18,202 occupied voxels at 5 cm resolution: 14,562 assigned to live foliage (80%, dead_fraction = 0.20) and 3,640 to dead fine fuel (20%). Total dry combustible mass is 3.20 kg (3.06 kg foliage + 0.13 kg dead fine), consistent with field measurements for *Picea abies* crowns of similar dimensions. At 50 cm resolution, the same foliage geometry produces 115 occupied voxels with no subdivision required (original triangle areas are smaller than the 2,500 cm² target), demonstrating the geometric scalability of the method.

5.3. Fire Simulation — Individual Tree Combustion Test

A combustion test was conducted on LUMI using a domain of 8 × 8 × 8 m discretized into 32 MPI meshes (each 2 × 2 × 4 m, IJK = 40 × 40 × 80, cell_size = 0.05 m, 4.1 × 10^6 cells total). The tree was ignited by a ring ignitor of radius 0.80 m centred at the base of the crown (Z = 0.05 m), providing 1,500 kW/m² for 10 seconds.

Smokeview particle visualisation at t = 0 confirms the correct spatial distribution of the two particle classes throughout the crown volume, faithfully reproducing the conical architecture of the Picea abies crown. At t = 6 s, the heat release rate reached 3.25 MW and the vertical temperature profile showed 752°C at Z = 0.5 m, 685°C at Z = 1.0 m, and 468°C at Z = 2.0 m, confirming active combustion propagating from the base of the crown upward through the foliage column.

PREPRINT

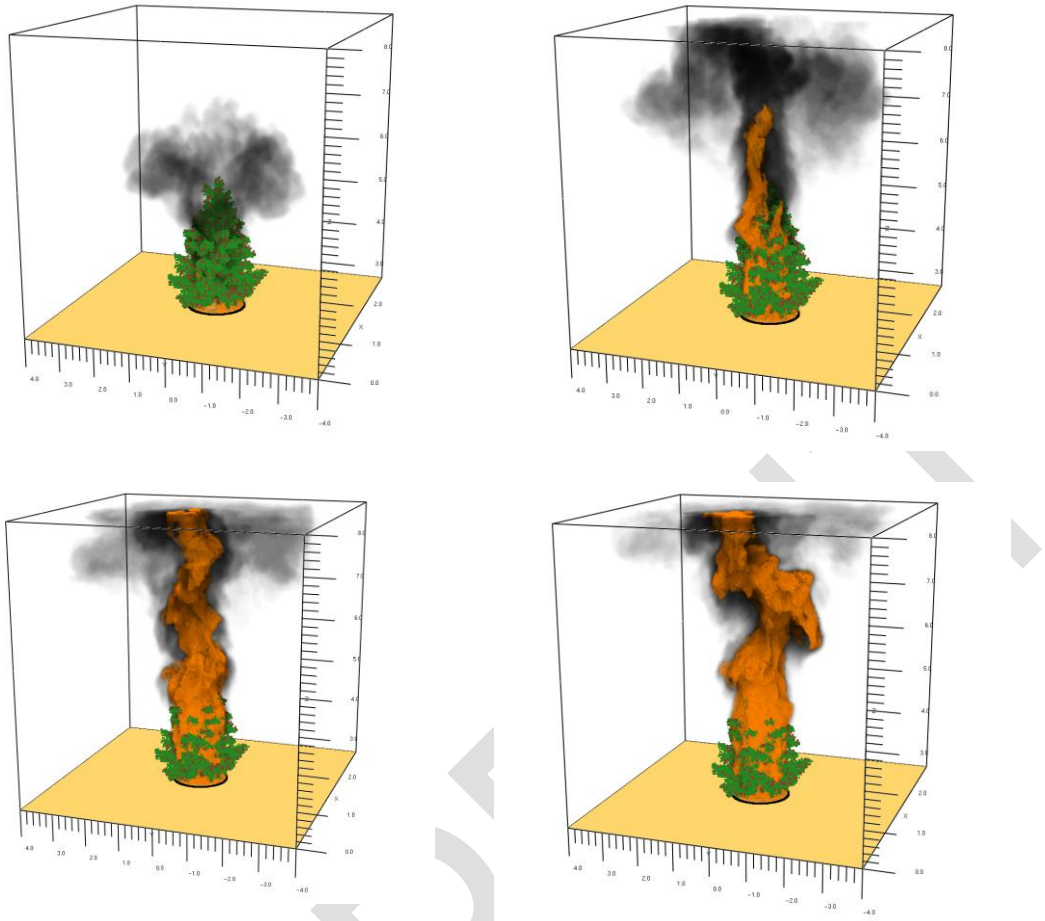


Figure 4 - Time sequence of a physics-based combustion simulation of a Picea abies tree, represented as a spatially explicit Lagrangian particle distribution in FDS/WFDS. The simulation was executed on the LUMI EuroHPC supercomputer using 32 MPI ranks at 5 cm grid resolution. HIDALGO2 Project.

The simulation revealed that with MOISTURE_CONTENT = 0.14 (14% dry-weight basis, approximately equivalent to severely drought-stressed foliage), the combustion is extremely rapid and the fire does not sustain after ignitor extinction at t = 10.5 s. This finding underscores the importance of correct moisture specification: live Picea abies foliage under moderate summer stress has dry-weight moisture contents of 80–100%, and future simulations will use MOISTURE_CONTENT = 0.80 with dead_fraction = 0.40 to test self-sustaining crown fire behavior.

5.4. Fire Simulation — Landscape Scale with Terrain and Multi-Tree Stand

A landscape-scale fire simulation was conducted on the HPC cluster MeluXina using the complete v3.2 pipeline over a one-hectare domain corresponding to the Rectoret urbanization in the municipality of Barcelona, Spain, a wildland-urban interface setting with residential structures embedded in forested terrain. The computational domain spans 100 × 100 × 100 m at 0.5 m grid resolution, yielding 8,000,000 cells distributed across 640 MPI ranks on 5 nodes (128 cores/node). Terrain geometry was derived from a Digital Terrain Model of the actual site, exported by *FDSTerrainSampler* as approximately 40,000 &OBST columns, reproducing the topographic relief of the domain with surface elevations ranging approximately 1.4 m across the 100 m extent.

The vegetation stand was populated in the UE5 Foliage tool with 4,490 instances of a *Picea abies* static mesh at varying scales, distributed over the one-hectare terrain. Although Picea abies is not the native species of this Mediterranean WUI site, the same 3D model and species parameters (MASS_PER_VOLUME, SURFACE_VOLUME_RATIO, MOISTURE_CONTENT) were used across all instances to demonstrate the

geometric scalability of the pipeline at this stand density, decoupled from species-specific botanical accuracy. The FDS Vegetation Exporter plugin processed all 4,490 instances in a single Export Full Case operation, generating the complete &INIT particle file set without manual intervention (Fig. 5).

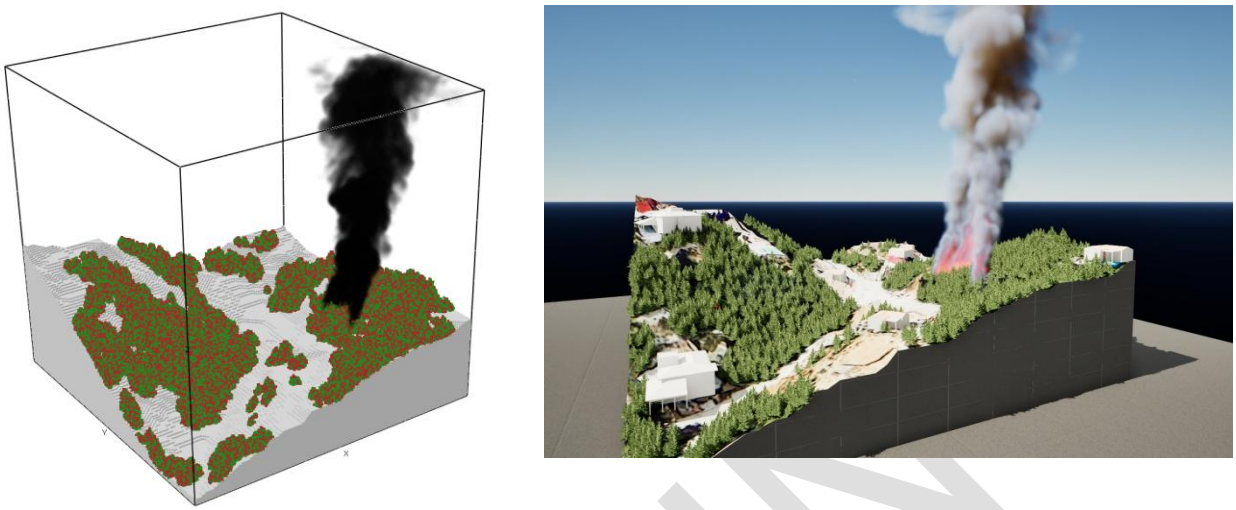


Figure 5 — Left: Smokeview visualization of the mdt_1ha case showing the Lagrangian particle distribution (green: live foliage, red: dead fine fuel) derived from 4,490 Picea abies instances over the terrain-rasterized DTM of the Rectoret site (Barcelona), with active fire front and smoke plume. Right: the same scene in the Unreal Engine 5 editor, showing the procedural vegetation stand and terrain alongside the FDS fire simulation result reintegrated as volumetric smoke, illustrating the end-to-end coherence of the pipeline.

Fire was initiated simultaneously at two locations via ring ignitors of radius 0.5 m placed at the base of tree crowns at terrain-following elevations, providing 900 kW/m² for 20 seconds. The simulation ran for T_END = 300 s with an ambient wind of 1 m/s from the west. The case completed successfully on MeluXina with stable timestep behavior throughout the full 300-second window, confirming the viability of the end-to-end pipeline (from UE5 PCG foliage placement to physics-based HPC fire simulation) at operational landscape scale with real terrain geometry and thousands of individually resolved tree instances.

6. Discussion

The fractal barycentric method presents a qualitative advance over existing approaches to fuel geometry in WFDS. The original WFDS model and all subsequent implementations known to the authors use statistical crown shapes (cylinders, cones, or ellipsoids) with uniform bulk density, populated by a single &INIT block covering the entire crown volume. The present method derives the spatial distribution of combustible material directly from the 3D architecture of the tree as represented in UE5, producing a per-voxel fuel map that reflects the real distribution of foliage mass throughout the crown.

The practical consequences for fire simulation are significant. In a uniform-crown model, all voxels within the crown envelope have identical combustible mass, and fire propagation is governed solely by the fluid dynamics of the FDS solver. In the present model, the crown periphery contains a greater concentration of occupied fuel voxels than the interior, and voxels corresponding to branch junctions or inter-whorl gaps have zero mass. This heterogeneity is expected to produce more realistic ignition patterns, faster peripheral flame spread, and more accurate predictions of crown fire initiation thresholds.

The integration of the full pipeline within the UE5 editor as a native C++ plugin removes a critical friction point of prior approaches tested by the author, in particular the intermediate OBJ file export and external Python converter. In v3.2, the UE5 artist or fire simulation engineer generates vegetation procedurally in the PCG graph, presses Scan Scene to verify the voxel count, and presses Export Full Case to obtain a ready-to-submit SLURM package. The plugin enforces correct physical parameters (CATF ordering, &SURF consistency, MPI decomposition at the 12,500 cells/core sweet spot) automatically, reducing operator error in large-scale simulation preparation.

PREPRINT

FDS source code analysis (read.f90, line 8184) establishes unambiguously that MOISTURE_CONTENT is interpreted on a dry-weight basis, consistent with standard forest fuel science convention: a value of 0.14 corresponds to 14% dry-weight moisture — appropriate for severely drought-stressed foliage — not 14% wet-weight as sometimes assumed from the FDS User Guide. Source code analysis (dump.f90) also confirms that SAMPLING_FACTOR affects only the fraction of particles written to the .prt5 visualization file for Smokeview rendering, with no effect on combustion physics — a practically important clarification for large-case output management.

The resolution validity analysis identifies a range of 0.05–0.30 m for individual-tree simulations with Picea abies geometry. This range coincides with the resolution needed to resolve the relevant turbulent length scales in FDS crown fire simulations, suggesting that the physical and numerical constraints of the method are well-matched.

7. Geometric Scalability: From Individual Trees to Landscape-Scale HPC Simulations

7.1. Physical Basis of Scalability

The scalability of the method rests on a fundamental property of the WFDS boundary fuel model: MASS_PER_VOLUME and SURFACE_VOLUME_RATIO are intensive properties of the fuel species — they characterise the material, not the computational cell. This has been confirmed analytically and verified in the FDS 6.10.1 source code (read.f90, part.f90). When FDS initializes a Lagrangian particle in a cell of volume V, it assigns a combustible mass M = MPV × V and computes the total fuel surface area as A = M × SVR / rho_solid. As V increases with cell size, M and A scale proportionally, preserving the correct energy release rate, convective exchange, radiative emission, and aerodynamic drag per unit volume of forest. The physics within each cell is identical whether the cell is 5 cm or 50 cm on a side.

The only quantity that changes with resolution is the spatial detail of the fuel distribution. At 5 cm resolution, the pipeline resolves individual branch layers, crown gaps, and the outer-inner density gradient of the canopy. At 50 cm, it resolves only the macroscopic envelope of the crown — which voxels contain foliage and which correspond to empty trunk space or inter-crown gaps. In both cases, the voxel occupancy map is derived directly from the same 3D vegetation geometry, without reparameterization or manual adjustment of any physical parameter.

7.2. Multi-Stratum and Multi-Species Forest Stands

The voxel occupancy criterion (one particle per voxel per fuel class) handles overlapping canopies naturally. When two tree crowns overlap in the same region of space, barycenters from both trees contribute to the same set of voxels. FDS resolves multiple particle classes in the same cell through independent 4D arrays indexed by (I, J, K, ARRAY_INDEX), accumulating mass, surface area, and temperature per class without numerical conflict — verified in FDS 6.10.1 source code (part.f90, lines 4405–4495). Multi-stratum forests, tall conifers with a shrub understory, or mixed stands with interlocking crowns, are represented simply by superimposing the voxel maps of each vegetation component, each with its own species-specific MPV and SVR. No special treatment is required for canopy overlap or vertical fuel stratification.

The UE5/PCG framework is particularly well-suited to this multi-species scenario: a single PCG graph generates a botanically realistic forest stand with dozens of species, spatial clustering, height stratification, and gap structure at landscape scale. The pipeline converts the entire stand to FDS particle input in a single pass of the converter, with one output file pair per species.

PREPRINT

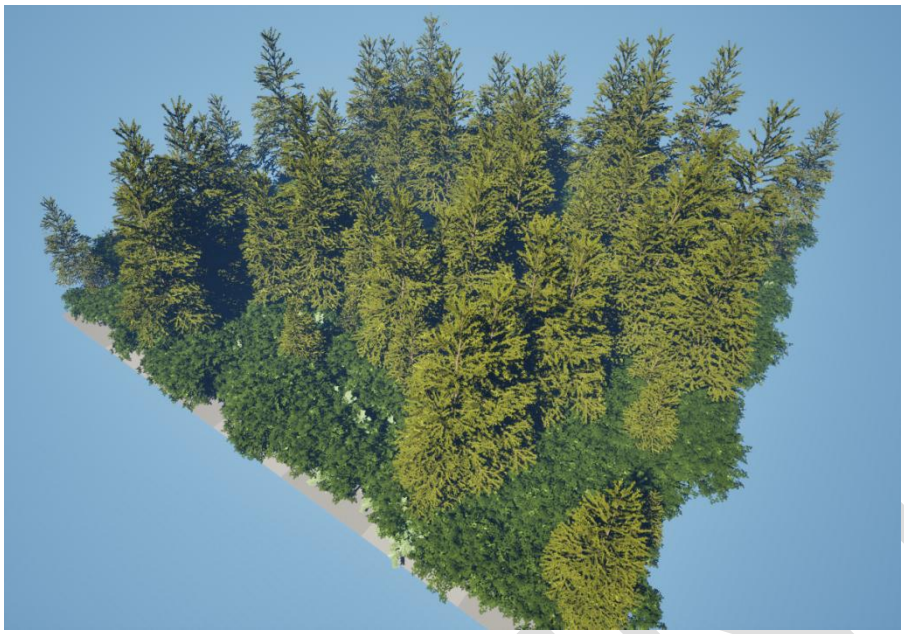


Figure 6 – A one-hectare procedurally generated forest stand illustrating the multi-stratum vegetation structure achievable with the UE5 PCG framework: ground-level grass, shrub understory, and a mixed-species tree canopy. The geometric complexity and botanical realism of this scene, produced automatically from ecological parameters without manual modeling, motivates the development of the pipeline presented in this paper for their direct translation into physics-based wildfire simulation. HIDALGO2 Project.

7.3. Scalability of the Discretization Algorithm

The computational cost of the fractal subdivision decreases as the FDS grid is coarsened. The stopping criterion is defined as A_target = cell_size²; therefore, the admissible sub-triangle area increases quadratically with cell size. As a result, fewer recursive subdivision levels are required at coarser resolutions. In the Picea abies test case, for example, most original facets required no subdivision at a cell size of 0.50 m, giving a mean subdivision factor of 1.0×, whereas substantially more sub-triangles were generated at 0.05 m resolution.

The number of occupied voxels and corresponding &INIT entries also decreases strongly with increasing cell size. For a volumetrically distributed crown of approximately fixed apparent volume, the expected upper-order dependence is proportional to V_crown / cell_size³. The exact reduction factor depends on crown architecture, internal gaps, facet orientation, and the binary voxel-occupancy criterion; it should therefore not be interpreted as an exact eightfold reduction whenever cell size is doubled. Nevertheless, grid coarsening substantially reduces both the number of barycentric samples required and the size of the particle initialization files. This behavior is favorable for landscape-scale domains, where the loss of fine architectural detail is deliberately traded against computational feasibility within the valid resolution range discussed in Section 4.6.

7.4. HPC Scaling and Deployment on EuroHPC Infrastructure

The FDS domain-decomposition model assigns one MPI rank to each &MESH block and is directly compatible with the spatial decomposition generated by the plugin. The exporter determines the mesh partitioning from the domain dimensions, grid resolution, and user-defined target number of cells per MPI rank. It also produces a spatially sorted list of &INIT entries, which FDS distributes during initialization through its standard particle-insertion routines, including INSERT_VOLUMETRIC_PARTICLES in part.f90. No modification of the FDS source code is required.

PREPRINT

The default value of 12,500 cells per MPI rank used by the plugin was obtained from a strong-scaling study conducted on LUMI with a 32-million-cell reference case containing Lagrangian vegetation particles. This value represents a practical performance target rather than a strict requirement: larger domains may use a higher number of cells per rank when constrained by the total number of available cores, memory, or allowable mesh count.

As an illustrative extrapolation, a domain of 500 × 400 × 40 m at 0.10 m resolution contains approximately 8 × 10^9 computational cells. Applying the nominal value of 12,500 cells per rank would require approximately 640,000 MPI ranks, which exceeds the maximum simultaneous CPU-core capacity of LUMI and MeluXina. Such a case is therefore not presented as directly executable with the current default decomposition. It represents an upper-scale design scenario that would require a combination of increased cells per rank, reduced spatial or vertical extent, coarser resolution, staged subdomains, or access to a larger future system. At approximately 380,000 ranks, corresponding to the theoretical CPU-core capacity of LUMI, the same domain would contain about 21,000 cells per rank, before accounting for operational allocation limits and system overheads.

The largest landscape-scale case demonstrated in the present work is the one-hectare simulation described in Section 5.4: 8,000,000 cells distributed over 640 MPI ranks on five MeluXina nodes. This case completed the full simulation period with stable timestep behaviour and confirms that the plugin can generate, decompose, and execute terrain-and-vegetation cases containing thousands of individually represented tree instances on EuroHPC infrastructure. Formal weak-scaling studies extending beyond this demonstrated scale remain part of future work.

The combination of resolution-dependent geometric discretization, species-level fuel parameters, automatic FDS mesh decomposition, and cluster-specific SLURM generation allows the same workflow to be applied from individual-tree combustion cases to forest-stand simulations. The physical parameters and export procedure remain unchanged, while grid resolution, domain decomposition, and computational resources are adapted to the spatial scale of the problem. This continuity between procedural vegetation generation in Unreal Engine 5, FDS particle initialization, and MPI execution on EuroHPC systems is a central architectural feature of the HIDALGO2 wildfire simulation pipeline.

8. Conclusions

This paper has presented and functionally verified a novel end-to-end pipeline for initializing FDS/WFDS Lagrangian particle distributions from 3D procedural vegetation geometry generated in Unreal Engine 5, implemented as a native C++ plugin for UE5.7. The main conclusions are as follows.

The fractal barycentric method correctly discretizes arbitrary triangulated vegetation geometry onto the FDS computational grid, with a systematic underestimation of voxel occupancy (70–90% coverage) that is preferable to the systematic overestimation of exact triangle-grid intersection methods. The underestimation is conservative and physically preferable for combustible fuel modeling.

A single Lagrangian particle per voxel per fuel class, initialized with species-specific MPV and SVR, is physically consistent with the WFDS boundary fuel model and produces numerically stable and physically plausible combustion behavior at the tested resolutions. The voxel occupancy criterion (presence or absence of fuel, not fuel quantity) is both computationally simple and physically defensible.

The method is geometrically scalable because MASS_PER_VOLUME and SURFACE_VOLUME_RATIO are intensive properties of the fuel species, independent of cell size. The same species parameters, the same plugin, and the same FDS input structure apply without modification from 5 cm resolution (individual tree combustion) to 50 cm resolution (landscape-scale crown fire propagation).

Multi-stratum and multi-species forest stands are handled naturally by the pipeline: overlapping canopies from different species contribute independently to the same voxels, and FDS resolves each fuel class through

PREPRINT

separate 4D internal arrays without numerical conflict. A procedurally generated forest stand of arbitrary complexity is converted to FDS input in a single export operation from the UE5 editor panel.

The integration of the full pipeline within the UE5 C++ plugin eliminates the intermediate OBJ file export and external Python converter steps of earlier approaches, enforces physically correct FDS input structure automatically, and generates ready-to-submit SLURM scripts for two EuroHPC clusters (LUMI and MeluXina) with a single click. The plugin has been functionally verified through fire simulations up to one-hectare scale with terrain and thousands of individually represented vegetation instances on MeluXina.

Future work will extend the method to multi-species landscape simulations using UE5 PCG population generation at stand scale, incorporate spatially variable MPV derived from photogrammetric or LiDAR point cloud data, validate against field crown fire experiments with known tree architectures, and conduct formal weak scaling studies to characterize HPC performance across the full range of domain sizes supported by EuroHPC infrastructure.

9. Acknowledgements

This work was supported by the HIDALGO2 project, funded by the European High Performance Computing Joint Undertaking (EuroHPC JU) under grant agreement No. 101093590. Computing resources were provided by LUMI (CSC, Finland) and MeluXina (LuxProvide, Luxembourg) through EuroHPC project allocations project_465002443 and p200983. The author thanks William Mell (USDA Forest Service, emeritus) and Eric V. Mueller (NIST) for foundational discussions on the WFDS boundary fuel model.

10. References

Cohen, J.D. (2004). Relating flame radiation to home ignition using a tilt angle. International Journal of Wildland Fire, 13(3), 351–361.

Mell, W.E., Manzello, S.L., Maranghides, A. (2006) Numerical Modeling of Fire Spread through Trees and Shrubs. V International Conference on Forest Fire Research D. X. Viegas (Ed.), 2006

Mell, W., Jenkins, M.A., Gould, J., Cheney, P. (2007). A physics-based approach to modelling grassland fires. International Journal of Wildland Fire, 16(1), 1–22.

Mell, W., Maranghides,A., McDermott, Manzello, S.L. (2009) Numerical simulation and experiments of burning Douglas-fir trees. Combustion and Flame 156 (2009) 2023–2041

Mell, W., Manzello, S.L., Maranghides, A., Butry, D., Rehm, R.G. (2010). The wildland–urban interface fire problem — current approaches and research needs. International Journal of Wildland Fire, 19, 238–251.

Mell, W. (2010) User Guide to WFDS. USDA Forest Service, Pacific Northwest Research Station.

Mell, W. et al. (2011). Using geospatial data to derive inputs to physics-based simulations of the wildland-urban interface fire problem. Fire and Materials, 35(1).

Mell, W., McNamara, D., Maranghides, A., McDermott, R., Forney, G., Hoffman, C., Ginder, M. (2011) Computer modelling of wildland-urban interface fires. Fire & Materials, 31 Jan – 2 Feb 2011, San Francisco, CA

Parsons, R.A., Mell, W.E., McCauley, P. (2011) Linking 3D spatial models of fuels and fire: Effects of spatial heterogeneity on fire behavior. Ecological Modelling 222 (2011) 679–691

Zuk, P.J, et al. (2022) OpenFOAM solver for thermal and chemical conversion in porous media. Computer Physics Communications 278 (2022) 108407.

PREPRINT