FloodA5: An Open-Source Framework for Flood Modelling on an Equal-Area Pentagonal DGGS

Matthew D. Wilson[1, 2]

[1] Geospatial Research Institute Toi Hangarau, University of Canterbury - Christchurch, New Zealand
[2] School of Earth and Environment, University of Canterbury - Christchurch, New Zealand - matthew.wilson@canterbury.ac.nz

Keywords: DGGS, flood modelling, A5, open-source software, GIS, sub-grid modelling

Abstract

Discrete Global Grid Systems (DGGS) provide a hierarchical alternative to conventional raster and vector spatial representations, offering globally consistent indexing, equal-area tessellations, and native support for multi-resolution analysis. Although DGGS have been widely applied in Earth observation and geospatial analytics, comparatively little work has explored their use as the primary computational mesh for hydrodynamic flood simulation. This paper presents FloodA5, an open-source flood modelling framework built on the A5 equal-area pentagonal DGGS. FloodA5 integrates DGGS-native mesh generation, terrain processing, hydrodynamic simulation, sub-grid terrain representation, data storage, and visualisation within a unified workflow implemented in Julia and Python. The framework employs an inertial formulation of the shallow-water equations adapted for operation on a non-orthogonal pentagonal mesh and includes an optional sub-grid sampling (SGS) approach based on pre-computed hypsometric storage relationships. FloodA5 is evaluated using synthetic benchmark problems and the Carlisle 2005 flood event. Results demonstrate that physically plausible flood simulations can be produced on an equal-area pentagonal DGGS and show reasonable agreement with a LISFLOOD-FP reference simulation, with the best-performing configuration achieving an intersection-over-union score of 0.67 and a depth RMSE of 1.17 m. Synthetic benchmarks reveal directional routing artefacts associated with the current treatment of non-orthogonal gradients and limitations in representing internal topographic barriers within the SGS formulation. The results demonstrate the feasibility of DGGS-native flood modelling while highlighting the importance of numerical schemes specifically designed for non-orthogonal polygonal meshes. FloodA5 provides an open-source platform for future research into hydrodynamic simulation on hierarchical global grid systems.

1. Introduction

Flood inundation models are widely used for flood hazard assessment, emergency management, infrastructure planning, and climate adaptation. Most contemporary two-dimensional flood models discretise the landscape using structured raster grids or unstructured meshes and solve different forms of the shallow-water equations, with different levels of simplification, to simulate the movement of water across the terrain (Bates et al., 2010; Neal et al., 2012).

Discrete Global Grid Systems (DGGS) have emerged as an alternative spatial framework for representing and analysing geospatial information. The Open Geospatial Consortium (OGC) DGGS standard defines a hierarchical spatial reference system based on recursively subdivided cells that provide consistent indexing and support multi-resolution analysis (Gibb et al., 2022). DGGS have been applied to Earth observation, environmental monitoring, spatial indexing, and global-scale geospatial analytics (Robertson et al., 2020), but comparatively little work has investigated their use as the primary computational mesh for hydrodynamic simulation.

One attraction of DGGS for hydrodynamic modelling is the ability to maintain a single spatial framework throughout the modelling workflow. Conventional approaches typically require conversion between raster data structures, computational meshes, and output grids. In contrast, a DGGS-native workflow can use a common spatial representation for mesh generation, terrain processing, simulation, storage, and visualisation. Equal-area DGGS frameworks are particularly attractive because water storage calculations depend directly on cell area and water depth.

This paper presents FloodA5, an open-source flood modelling framework built on the A5 equal-area pentagonal DGGS and implemented using the A5 ecosystem, including the pya5 library (Palmer and contributors, 2026). FloodA5 integrates DGGS-native mesh generation, terrain processing, hydrodynamic simulation, and visualisation within a single workflow. The objectives of this paper are to:

1. Present the architecture and implementation of FloodA5;
2. Evaluate the feasibility of hydrodynamic simulation on an equal-area pentagonal DGGS;
3. Assess model behaviour using synthetic benchmark problems and a real-world flood case study; and
4. Identify methodological challenges requiring further investigation.

2. FloodA5 Framework

2.1 A5 Discrete Global Grid System

FloodA5 is built upon the A5 Discrete Global Grid System (DGGS), an equal-area hierarchical tessellation of the Earth's surface composed of pentagonal cells (Palmer and contributors, 2026). At each resolution level, all cells possess identical area, while refinement follows a strict parent–child hierarchy in which each cell subdivides into five children at the next resolution level. Higher resolution levels yield finer cells: at resolution 14, cells have a geodetic plan area of approximately $2\,\mathrm{km}^2$ and a characteristic spacing of $\sim 1.4\,\mathrm{km}$ (Table 1).

Unlike conventional raster grids, interior A5 cells possess a uniform five-neighbour topology. Cell identifiers are represented using compact hierarchical indices, enabling efficient storage, spatial querying, and future support for adaptive mesh refinement. The equal-area nature of the grid is particularly attractive for hydrodynamic applications because water storage can be calculated directly from cell volume and depth without requiring latitude-dependent area corrections.

Level	Cell area	Spacing ($\approx \sqrt{A}$)
8	$518\,\text{km}^2$	22.8 km
10	$32.4\,\text{km}^2$	5.7 km
12	$2.02\,\text{km}^2$	1.42 km
14	12.6 ha	356 m
16	$7902\,\text{m}^2$	89 m
18	$494\,\text{m}^2$	22 m
20	$31\,\text{m}^2$	5.6 m

Table 1. A5 resolution levels and cell properties at a mid-latitude reference point (Christchurch, NZ; lon 172.6°, lat −43.5°). Area computed via the Shoelace formula on a local equirectangular projection; spacing $\approx \sqrt{A}$.

No native Julia library for A5 exists; all topological operations use the Python package `pya5` (Palmer and contributors, 2026), called via a persistent subprocess bridge (`a5_bridge.py`). Cell IDs are normalised to 16-character zero-padded hexadecimal strings throughout both the Julia and Python components to prevent padding-length mismatches between the two environments. FloodA5 currently operates on a single-resolution mesh, although the hierarchical structure of A5 is intended to support future adaptive multi-resolution simulations.

2.2 System Architecture

FloodA5 is implemented primarily in Julia (Bezanson et al., 2017), with Python providing access to the `pya5` DGGS library through a lightweight interoperability layer, `a5_bridge.py`. The framework was designed to provide a complete DGGS-native flood-modelling workflow, integrating mesh generation, terrain processing, hydrodynamic simulation, visualisation, and data storage within a common spatial framework. Figure 1 illustrates the overall pipeline for the generation of a simulation-ready flow state for a provided AoI and DEM. Key dependencies are listed in Table 2.

The modelling workflow consists of five stages: (i) Generation of an A5 mesh from a user-defined area of interest; (ii) Sampling and processing of digital elevation model (DEM) data; (iii) Construction of optional sub-grid terrain representations; (iv) Hydrodynamic simulation; and (v) Output, storage, and visualisation. Mesh and terrain products are stored using GeoParquet, while simulation outputs are written to HDF5. FloodA5 includes both desktop visualisation using GLMakie and browser-based visualisation using CesiumJS. A key design objective is the maintenance of a single DGGS-native representation throughout the modelling workflow. Unlike conventional approaches that frequently require conversion between raster, mesh, and vector data structures, FloodA5 uses the DGGS as the primary spatial framework from mesh generation through to simulation output.

2.3 AOI Discretisation

Mesh generation begins with a user-supplied AOI as a GeoJSON polygon in EPSG:4326. Cell discovery proceeds in two passes.

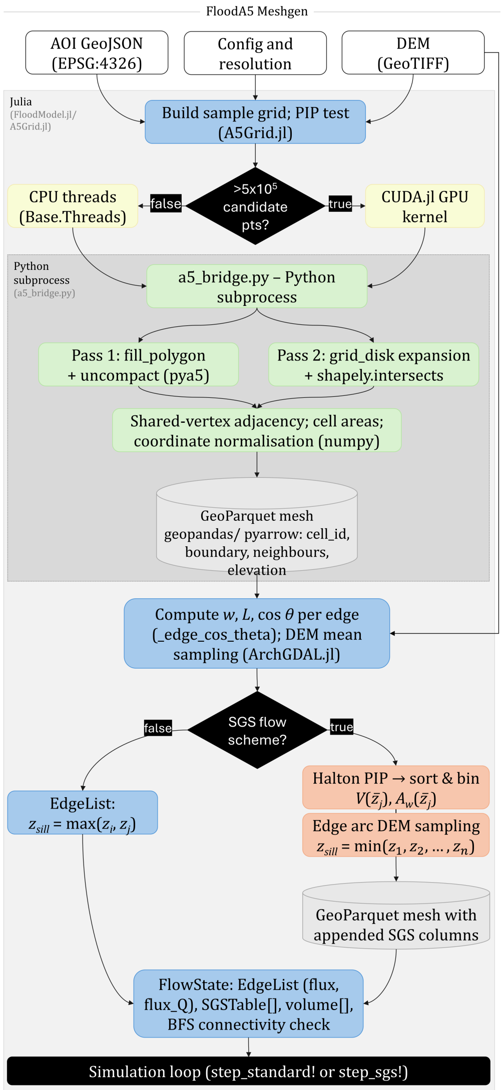

Figure 1. FloodA5 mesh generation and flow-model initialisation pipeline. Blue nodes are Julia components; green nodes execute in the Python subprocess via `a5_bridge.py`; orange nodes are the additional SGS pre-processing steps, bypassed for the standard flow pathway. The GPU/CPU branch applies to the point-in-polygon test: the `CUDA.jl` kernel is invoked when the total candidate point count exceeds 5×10^5, otherwise multithreaded CPU evaluation is used. The edge sill sampling loop is executed serially because the `ArchGDAL` coordinate transform uses `PyCall.jl`, whose garbage-collector finaliser is not thread-safe on non-main threads. Both pathways converge to a `FlowState` struct that drives the simulation loop.

Pass 1 (seed cells). `pya5.fill_polygon` returns a compact, multi-resolution representation of all A5 cells covering the AOI; `pya5.uncompact` expands this to a flat set of cells at the target resolution. However, at every A5 resolution level, pentagons tile in two geometrically complementary *sublattices*: cells whose IDs begin with the hex prefix `8e...` and those beginning with `9e...` (at resolution 14). Every cell in one sublattice is surrounded exclusively by cells of the other. The `fill_polygon` function returns cells from only one sublattice, so its output has zero topological edges — every neighbour of every returned cell belongs to the absent sublattice, and no inter-cell flux would be computed.

Package	Language	Role
`pya5`	Python	A5 cell indexing, topology
`shapely`	Python	AOI intersection tests
`geopandas`	Python	GeoParquet I/O
`pyarrow`	Python	Arrow list columns (SGS tables)
`numpy`	Python	Coordinate normalisation
`ArchGDAL.jl`	Julia	DEM I/O, CRS reprojection
`CUDA.jl`	Julia	GPU point-in-polygon kernel
`HDF5.jl`	Julia	Simulation output
`Arrow.jl`	Julia	GeoParquet read (SGS columns)
`JSON3.jl`	Julia	AOI and mesh serialisation
`Oxygen.jl`	Julia	HTTP/WebSocket visualisation server
`GLMakie.jl`	Julia	Native desktop visualiser

Table 2. Key software packages used in FloodA5.

Pass 2 (expansion). To recover both sublattices, an iterative expansion is applied: for each cell in the current set, `pya5.grid_disk(cell, 1)` retrieves first-ring neighbours. Because `grid_disk` returns a *compact* result that may include cells at different resolution levels, results are expanded with `pya5.uncompact` at the target resolution and each candidate is checked with `pya5.get_resolution` before inclusion. Candidates whose polygons intersect the AOI (via `shapely.intersects`) are added to the cell set. The expansion iterates until convergence, after which a final intersection filter removes any cell whose polygon does not overlap the AOI.

For each cell, boundary vertices and the geodetic centre are retrieved from `pya5`. Longitudes are normalised to $[-180°, 180°]$ and cell areas are computed using the Shoelace formula on a local equirectangular projection centred on each cell's centroid (error $< 0.1\%$ at resolution 14).

Adjacency. Two cells are defined as edge-sharing neighbours if and only if their boundaries share exactly two vertices. This shared-vertex criterion is geometrically exact and avoids the compaction artefacts that cause `grid_disk` to occasionally return fewer than five neighbours for boundary cells. The adjacency is stored as a `neighbours` list column in the output GeoParquet file, eliminating the need for any `pya5` calls at simulation run-time.

2.4 Terrain Representation

Terrain data are obtained from raster DEMs and transferred to the DGGS through point-in-polygon aggregation. DEM samples falling within each cell are identified and aggregated to produce a representative cell elevation. For the standard solver, the mean elevation is retained and used as the computational bed elevation.

Terrain information is derived from user-supplied digital elevation models. Elevation values are sampled onto A5 cells using point-in-polygon methods, with optional GPU acceleration available for large domains and high-resolution meshes. For the standard solver, each cell is represented by a single mean elevation value. Shared edge elevations are derived from neighbouring cell elevations and are used to determine hydraulic connectivity.

FloodA5 includes an optional Sub-Grid Sampling (SGS) formulation intended to represent terrain variability below the computational mesh resolution. Rather than storing a single elevation value per cell, the SGS approach constructs hypsometric relationships from DEM samples contained within each DGGS cell.

For each cell, pre-processing generates: (i) volume–elevation relationships; (ii) wetted area–elevation relationships; (iii) edge conveyance relationships; (iv) wetted perimeter relationships; and (v) edge sill elevations.

Figure 2 illustrates the construction of SGS relationships for a representative cell for the Carlisle case study.

During simulation, water storage is tracked as volume and converted to water-surface elevation through inversion of the volume–elevation relationship. Edge flux calculations utilise hydraulic properties derived from the SGS tables rather than assuming uniform cell geometry. This approach allows partial wetting of cells and representation of channels that are substantially narrower than the computational mesh resolution while avoiding the computational cost associated with mesh refinement.

2.5 DEM Sampling — Standard Pathway

Bed elevation is assigned to each cell by one of two methods selected at run-time.

Centroid method. The DEM is sampled at the cell centre via bilinear interpolation (`ArchGDAL`; Tan and contributors 2026). Fast and sufficient for coarse resolution or smooth terrain.

Cell-mean method. A 2-D Halton quasi-random sequence (Halton, 1960) in bases 2 and 3 generates N_s candidate points (default $N_s = 256$) within the cell bounding box. A point-in-polygon (PIP) test retains only interior candidates; the cell mean elevation is the average of the valid DEM samples. The PIP test is executed on the GPU (`CUDA.jl`) when the total candidate count exceeds 5×10^5, and on CPU threads otherwise. CRS reprojection from EPSG:4326 to the native DEM CRS is performed by `ArchGDAL` for each cell batch.

2.6 SGS Pre-processing — Hypsometric Tables

Rather than storing a single representative elevation, SGS pre-processing constructs a series of monotonic lookup tables describing sub-cell topographic structure. Elevation samples are sorted and converted into cumulative volume–elevation and wetted area–elevation relationships. Additional edge-based tables describing sill elevations, flow area, wetted perimeter and hydraulic radius are generated from DEM samples intersecting shared cell boundaries.

The sub-grid sampling (SGS) pathway replaces a single mean bed elevation per cell with pre-computed lookup tables that map water-surface elevation (WSE) to hydraulic geometry. Two sets of tables are constructed: one for each *cell* (describing inundation storage) and one for each *edge* (describing the cross-sectional geometry of the flow path between adjacent cells). This resolves partial inundation, sub-cell channels, and concentrated channel flow without refining the mesh.

Cell hypsometric curves. For each cell, a Halton sequence with $N_s = 512$ candidates is generated and filtered to interior points (CPU multithreaded). The N_{in} interior sample elevations $\{z_k\}$ are sorted and used to construct $N_b = 100$ quantile-spaced WSE levels $\bar{z}_1 < \cdots < \bar{z}_{N_b}$ spanning $[z_{\min}, z_{\max}]$. For each level, the cumulative stored volume and wetted plan area are:

$$V(\bar{z}_j) = \frac{A_{\text{cell}}}{N_{\text{in}}} \sum_{k:\, z_k \le \bar{z}_j} (\bar{z}_j - z_k), \qquad A_w(\bar{z}_j) = \frac{A_{\text{cell}}}{N_{\text{in}}} \sum_{k:\, z_k \le \bar{z}_j} 1 \tag{1}$$

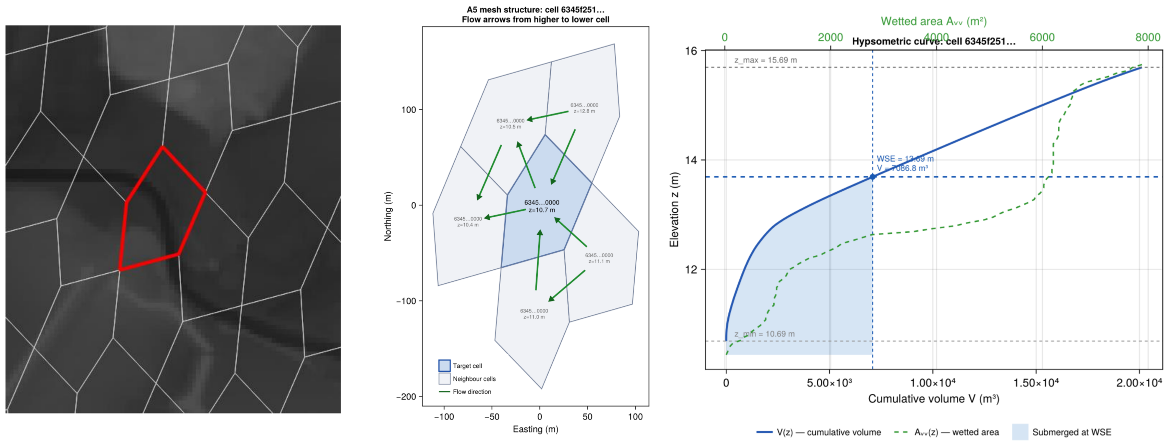

Figure 2. The A5 mesh structure and generation of hyposometric curves: left shows a representative resolution-16 cell (ID: 6345f25178000000) for a location with sub-cell topographic variability, illustrated using the underlying DEM (lower elevations are darker); the middle plot shows a single cell and its neighbours, with flow direction for shallow water indicated; right shows the hyposometric curve for this cell, with a representative water surface elevation (WSE) indicated.

where A_{cell} is the geodetic plan area. At run-time, WSE is recovered from stored volume by inverse linear interpolation of $V(\bar{z}_j)$. When the stored volume exceeds $V(\bar{z}_{N_b})$ (cell terrain ceiling), WSE is extrapolated linearly as:

$$\text{WSE} = z_{\max} + \frac{V - V(\bar{z}_{N_b})}{A_{\text{cell}}} \tag{2}$$

preserving a positive driving head between overfull upstream cells and allowing continuous downstream routing above the terrain ceiling.

Edge hydraulic curves. For each pair of adjacent cells, the DEM is sampled at regular intervals along the shared edge arc (spacing equal to half the DEM pixel size), producing a sorted elevation profile $\{e_m\}_{m=1}^{M}$ of the flow cross-section. Using the same N_b WSE knot levels $\bar{z}_j$ as the cell curves, the cumulative cross-sectional flow area and wetted perimeter are built incrementally:

$$A(\bar{z}_j) = \delta x \sum_{m:\, e_m \le \bar{z}_j} (\bar{z}_j - e_m), \qquad P(\bar{z}_j) = \delta x \sum_{m:\, e_m \le \bar{z}_j} 1 \tag{3}$$

where $\delta x = W/M$ is the width element per sample point and W is the haversine edge length. The hydraulic radius is $R(\bar{z}_j) = A(\bar{z}_j)/P(\bar{z}_j)$. These per-edge curves replace the wide-channel approximation inherent in the Bates (2010) formulation and provide a physically accurate friction term for concentrated channel flow at any resolution (Section 3).

The minimum sampled elevation is retained as the flow sill z_{sill} for each edge, unchanged from the earlier implementation. The edge arc sampling loop is executed serially because the `ArchGDAL` coordinate transform uses `PyCall.jl`, whose garbage-collector finaliser is not thread-safe on non-main threads.

All cell and edge tables are stored as Apache Arrow list columns in the output GeoParquet file: `sgs_elev_bins`, `sgs_vol_curve`, `sgs_area_curve`, `sgs_edge_sills`, `sgs_edge_area_curve`, and `sgs_edge_perim_curve`. The edge curve columns are stored as flat arrays of length $N_b \times 5$ per row (five adjacency slots per cell), reshaped to an $(N_b \times 5)$ matrix on load. Meshes generated without the edge curve columns (pre-existing files) load successfully and fall back to the Bates SGS kernel with a logged warning.

2.7 Flow Model Initialisation

At the start of a simulation, the GeoParquet mesh is loaded into Julia and an undirected `EdgeList` is constructed in which each edge appears exactly once, with the lower-index cell assigned to i and the higher-index cell to j. For each edge, the shared edge length w (haversine between the two shared boundary vertices), the centre-to-centre distance L, the non-orthogonality cosine $\cos\theta$ (Section 3.1), and the flow sill z_{sill} are computed and stored. SGS meshes additionally load the cell and edge hypsometric tables into a `SGSTable` struct per cell; these contain the $A(\bar{z}_j)$ and $P(\bar{z}_j)$ curves (Eq. 3) indexed by adjacency slot for efficient per-edge lookups at run-time. The `EdgeList` carries two flux arrays: `flux` (q, $\text{m}^2\,\text{s}^{-1}$, unit discharge for the standard solver) and `flux_Q` (Q, $\text{m}^3\,\text{s}^{-1}$, volumetric discharge for the SGS Manning R-A solver).

A breadth-first search connectivity check warns if the mesh contains isolated components, which can occur for small AOIs at fine resolution where the two A5 sublattices fail to overlap sufficiently; a minimum AOI area of $5\,\text{km}^2$ ensures full connectivity at resolution 14.

2.8 Software Availability

FloodA5 is released as open-source software under the Apache 2.0 licence. Source code, documentation, test cases, and example workflows are available through a public GitHub repository (Wilson, 2026). The framework includes command-line tools for mesh generation, simulation execution, visualisation, and post-processing. Input formats include GeoTIFF elevation models, GeoJSON areas of interest, and LISFLOOD-FP boundary condition files, facilitating interoperability with existing flood-modelling workflows.

3. Hydrodynamic Formulation

FloodA5 currently implements standard solver (section 3.2) and a sub-grid solver (SGS) (section 3.3). Both advance the stored water volume V_i in each cell as the primary state variable. The water-surface elevation (WSE) is derived from V_i at each timestep: for the standard solver, $\mathrm{WSE}_i = z_i + V_i/A_{\mathrm{cell}}$; for the SGS solver, by inverse interpolation of the cell hypsometric curve (Eq. 2).

3.1 Non-Orthogonal Edge Geometry

A5 cells form a non-orthogonal mesh. FloodA5 accounts for this geometry through a first-order correction based on the angle between the cell-centre connection vector and the edge normal:

$$L_{\mathrm{eff}} = L \cos\theta \tag{4}$$

where L is the centre-to-centre distance and θ is the non-orthogonality angle. This approach provides a simple correction for skewed cell geometries but does not fully reconstruct the gradient field. More sophisticated approaches based on least-squares gradient reconstruction and over-relaxed decomposition have been proposed for polygonal meshes (Weller, 2014).

3.2 Standard Flow — Bates (2010) Inertial Formulation

The standard solver applies the inertial shallow-water formulation of Bates et al. (2010) to each edge of the `EdgeList`. For edge e connecting cells i and j, the unit discharge ($\mathrm{m^2\,s^{-1}}$) at time t is:

$$q^t = \frac{q^{t-\Delta t} - g\, h_{\mathrm{flow}}\, \Delta t\, \dfrac{\Delta \mathrm{WSE}}{L_{\mathrm{eff}}}}{1 + g\,\Delta t\, n^2\, |q^{t-\Delta t}|\, h_{\mathrm{flow}}^{-7/3}} \tag{5}$$

where $h_{\mathrm{flow}} = \max(\mathrm{WSE}_i, \mathrm{WSE}_j) - z_{\mathrm{sill}}$ is the flow depth above the edge sill (set to $z_{\mathrm{sill}} = \max(z_i, z_j)$ for the standard solver), $\Delta\mathrm{WSE} = \mathrm{WSE}_i - \mathrm{WSE}_j$ is the driving head, L_{eff} is the corrected gradient length (Eq. 4), n is Manning's roughness (arithmetic mean of the two adjacent cells), and $g = 9.81\,\mathrm{m\,s^{-2}}$. The sign convention follows the `EdgeList` canonical ordering: $q < 0$ when $\mathrm{WSE}_i > \mathrm{WSE}_j$ (flow from i to j); the volumetric flux is $Q = q \cdot w$ ($\mathrm{m^3\,s^{-1}}$), where w is the shared edge length. Cell volumes are updated as $\Delta V_i = Q\,\Delta t$ and $\Delta V_j = -Q\,\Delta t$.

Three stability measures are applied in `_bates_flux_limited`, informed by a line-by-line comparison with the LISFLOOD-FP ACC solver: (i) a Froude limiter caps $|q|$ at $h_{\mathrm{flow}}\sqrt{g\,h_{\mathrm{flow}}} \times 0.8$, preventing supercritical discharge on the pentagonal mesh where five independently evolving q_{prev} values per cell lack the directional damping present on rectangular grids; (ii) a per-edge volume limiter constrains the transfer to $h_{\mathrm{donor}} \cdot w/(5\,\Delta t)$; and (iii) the post-limiting unit discharge q_{stored} is written back to `edges.flux` so the momentum state carried into the next step is consistent with what was physically transferred.

The CFL timestep uses the wave-speed criterion:

$$\Delta t_{\mathrm{CFL}} = C\, \frac{\Delta x_{\min}}{\sqrt{g\, h_{99}}} \tag{6}$$

with Courant number $C = 0.7$, $\Delta x_{\min} = \sqrt{\min(A_{\mathrm{cell}})}$, and h_{99} the 99th-percentile wet-cell depth (to prevent a single deep channel cell from forcing unnecessarily small timesteps across the whole domain).

3.3 SGS Flow — Manning R-A Formulation

The SGS solver replaces the standard depth-area relationship with storage volumes derived from hypsometric curves. Water surface elevations are obtained by inversion of the volume-elevation relationship. Flow routing uses pre-computed edge hydraulic properties and a Manning hydraulic radius formulation, allowing conveyance to vary according to sub-cell topographic structure.

At fine resolution (e.g. resolution 18, $\Delta x \approx 22\,\mathrm{m}$), the wide-channel approximation in Eq. (5) underestimates friction for concentrated channel flow: when the wetted width P is much smaller than the cell width w, using h_{flow} in place of the hydraulic radius $R = A/P$ overestimates the effective depth and therefore over-delivers momentum. The SGS solver replaces the Bates unit-discharge kernel with a volumetric formulation using the per-edge cross-sectional curves $A(\bar{z}_j)$ and $P(\bar{z}_j)$ computed during pre-processing (Eq. 1):

$$Q^t = \frac{Q^{t-\Delta t} - g\,A\,\Delta t\, \dfrac{\Delta\mathrm{WSE}}{L_{\mathrm{eff}}}}{1 + g\,\Delta t\, n^2\, |Q^{t-\Delta t}|\, (R^{4/3}\, A)^{-1}} \tag{7}$$

where Q ($\mathrm{m^3\,s^{-1}}$) is the volumetric discharge, A ($\mathrm{m^2}$) and R (m) are the cross-sectional flow area and hydraulic radius at $\mathrm{WSE} = \max(\mathrm{WSE}_i^{\mathrm{eff}}, \mathrm{WSE}_j^{\mathrm{eff}})$, averaged symmetrically from both cells' perspectives ($A = \frac{1}{2}(A_i + A_j)$, $R = \frac{1}{2}(R_i + R_j)$). The $R^{4/3}\,A$ denominator term provides physically correct friction scaling: as a channel fills, both A and R grow, making the denominator self-stabilising without requiring an explicit Froude or volume limiter in the flux kernel. This mirrors the LISFLOOD-FP SGC formulation (Shaw et al., 2021), adapted to the hypsometric pre-processing approach of FloodA5 in place of parametric cross-sections.

To ensure that a high-elevation dry cell cannot drive spurious flux through a low sill into an adjacent basin, a dry cell's effective WSE is set to $\max(z_{\mathrm{sill}},\, z_{\min,i})$ rather than $z_{\min,i}$ alone. The post-limiting volumetric discharge Q_{stored} is written back to `edges.flux_Q` (for consistency, analogous to the standard solver). A last-resort per-edge donor cap ($|\Delta V| \leq V_{\mathrm{donor}}/10$) is retained as a mass-conservation guard.

3.4 Two-Solver Architecture

Table 3 summarises the key differences between the two solvers. Both share the same `EdgeList`, CFL criterion, volume update functions, and velocity computation. The standard solver is computationally cheaper; the SGS solver provides aims to represent terrain heterogeneity within cells, at the cost of mesh pre-processing time and the additional memory for the edge hydraulic tables.

4. Evaluation

Two synthetic benchmark problems were used to evaluate model behaviour. The first benchmark consists of a point-source injection on a flat domain. Under isotropic conditions, the resulting inundation pattern should be circular, providing a simple test of directional bias. The second benchmark consists of a planar slope intersected by a perpendicular embankment. This test evaluates routing behaviour under a known flow direction and assesses the ability of the SGS formulation to represent sub-cell topographic barriers. Results from both benchmark problems are presented and discussed in Section 5.

FloodA5 was also evaluated using the January 2005 flood event in Carlisle, following the domain configuration and inflow hydrographs presented by Wilson and Coulthard (2023) (Figure 3). Three FloodA5 simulations were performed: (i) resolution 18

Property	**Standard**	**SGS**
Flux kernel	Bates (2010) eq. 9, q (m^2 s^{-1})	Manning R-A, Q (m^3 s^{-1})
WSE from volume	$z_i + V_i/A_{\text{cell}}$	Hypsometric inverse interpolation
Edge sill	$\max(z_i, z_j)$	$\min$ DEM along shared edge arc
Friction term	$h_{\text{flow}}^{7/3}$ (wide channel)	$R^{4/3} \cdot A$ (hydraulic radius)
Stability	Froude + volume limiter in kernel	Self-stabilising via R–A denominator
Momentum state	`edges.flux` (q)	`edges.flux_Q` (Q)
Pre-processing	Cell mean elevation only	Cell + edge hypsometric tables

Table 3. Comparison of the two FloodA5 flow solvers.

standard solver (∼22 m cell spacing); (ii) resolution 20 standard solver (∼5.6 m cell spacing); and (iii) resolution 18 SGS solver. Model outputs were compared with a 5 m LISFLOOD-FP simulation, with agreement assessed using maximum inundation extent intersection-over-union (IoU) and depth RMSE metrics.

5. Results

5.1 Synthetic Benchmarks

Figure 4 presents results from the point-source benchmark. The simulation produced a near-circular inundation pattern with a Polsby–Popper circularity score of 0.96. However, there was a preferential propagation axis aligned approximately northwest–southeast, indicating directional routing bias within the current implementation.

Figure 5 presents results from the planar slope and embankment benchmark. Rather than propagating directly downslope, the flood wave exhibited a systematic deviation of approximately 30–60° from the expected flow direction. This result suggests limitations in the current treatment of non-orthogonal gradients.

The SGS simulation successfully represented sub-cell terrain variability but failed to reproduce the hydraulic effect of an internal embankment located entirely within individual cells. Water was able to pass through the barrier because the current SGS formulation stores elevation distributions but does not explicitly preserve the spatial arrangement of topographic features within cells.

5.2 Carlisle Simulation

Table 4 summarises agreement between FloodA5 and the LISFLOOD-FP reference simulation and Figure 6 illustrates the flood peak depth. The Resolution 18 standard simulation produced the closest correspondence with the LISFLOOD-FP reference simulation, achieving an IoU of 0.67 and RMSE of 1.17 m. While these results indicate reasonable agreement between the two models, all FloodA5 simulations over-predicted inundation extent relative to the reference solution.

Model	IoU	RMSE (m)
Resolution 18 Standard	0.668	1.174
Resolution 20 Standard	0.593	1.684
Resolution 18 SGS	0.553	1.711

Table 4. Comparison with LISFLOOD-FP maximum inundation depth.

The SGS simulation exhibited the greatest inundation over-prediction despite representing sub-cell topography. This behaviour appears to be associated with increased hydraulic connectivity introduced by the SGS formulation, but requires further investigation.

5.3 Computational Performance

Table 5 summarises mesh generation and simulation runtimes for the evaluated FloodA5 configurations. All performance measurements were obtained on an AMD Ryzen Threadripper 9970X workstation (32 cores).

Configuration	Cells	Mesh Generation	Simulation Runtime
Resolution 18 Standard	29,902	2 min	39 min
Resolution 18 SGS	29,902	2 min	71 min
Resolution 20 Standard	472,681	17.8 min	12.2 h

Table 5. FloodA5 computational performance.

For the Resolution 18 configurations, DEM sampling required approximately 14 seconds, while the Resolution 20 mesh required approximately 110 seconds due to the substantially larger number of cells. The SGS preprocessing stage was performed during mesh generation and therefore did not contribute additional runtime during simulation.

Simulation runtimes increased substantially with mesh resolution. Increasing resolution from Level 18 to Level 20 increased the number of computational cells by a factor of approximately 16, from 29,902 to 472,681 cells, and increased simulation runtime from 39 minutes to approximately 12.2 hours. This behaviour is broadly consistent with the increased computational cost associated with evaluating fluxes across a larger number of cells and edges.

The SGS solver incurred additional computational cost relative to the standard solver due to repeated evaluation of hypsometric relationships and hydraulic property tables during each timestep. For the Carlisle case study, the Resolution 18 SGS simulation required approximately 1.8 times longer to complete than the equivalent standard simulation.

The results indicate that DGGS-native flood simulations can be performed on workstation-class hardware without specialised high-performance computing resources. The Resolution 18 configuration, which also produced the closest correspondence with the LISFLOOD-FP reference simulation, completed a 120-hour simulation in 39 minutes, equivalent to approximately 184 simulated hours per wall-clock hour.

6. Discussion

FloodA5 demonstrates that a complete flood-modelling workflow can be implemented within a DGGS-native spatial framework. The system integrates mesh generation, terrain processing, hydrodynamic simulation, data storage, and visualisation without requiring conversion to conventional raster computational grids.

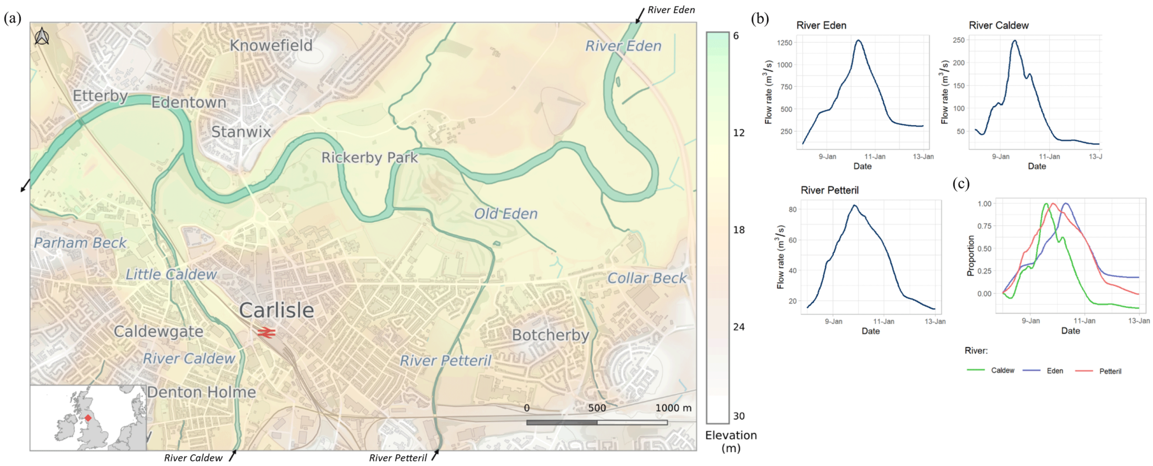

Figure 3. Test case site at Carlisle, UK, at the confluence of the rivers Caldew, Petteril, and Eden, with river flows for the January 2005 flood event simulated shown in (b). Reproduced from Wilson and Coulthard (2023) under CC-BY-4.0 licence. The figure contains OS data © Crown copyright and database right (2020).

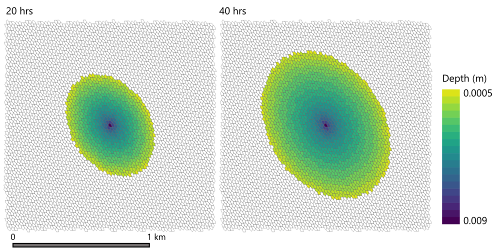

Figure 4. Results from the point spread test, showing depth for a flat DEM with constant inflow to the centre (standard flow), after 20 and 40 hours.

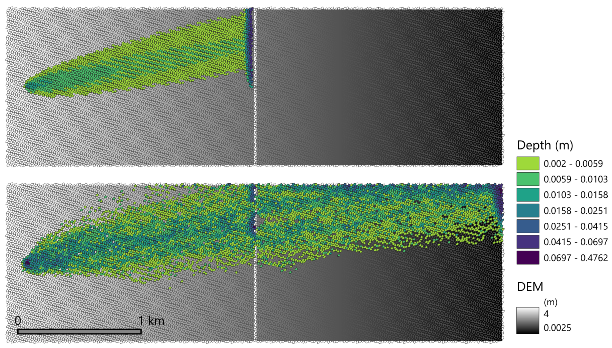

Figure 5. Results from simulation for the planar slope with embankment (closed domain), showing depth after 20 hours of constant input of 0.1 $m^3 s^{-1}$. Top is the standard flow model, which is blocked by the embankment as the cells contain sufficient representation of its height; bottom shows the SGS model where the embankment is represented as sub-cell topography, but it does not block flow due to the absence of spatial information to the flow solver.

The results demonstrate that hydrodynamic flood simulation can be implemented successfully on an equal-area pentagonal DGGS. The framework produced physically plausible flood inundation patterns in both synthetic and real-world test cases and achieved reasonable agreement with an established LISFLOOD-FP reference simulation.

However, the synthetic benchmarks revealed important limitations. The directional bias observed in both the point-source and planar slope experiments indicates that the current first-order non-orthogonality correction is insufficient for fully accurate routing on the A5 mesh. Although the DGGS topology itself provides uniform connectivity, accurate simulation on non-orthogonal polygonal meshes requires numerical treatments specifically designed for such geometries. Future work will investigate least-squares gradient reconstruction and over-relaxed non-orthogonal correction methods following Weller (2014).

The SGS formulation also revealed limitations. While the approach successfully represents sub-cell elevation variability, it does not preserve the internal spatial arrangement of topographic features. Consequently, barriers located entirely within cells may not be represented correctly. Similar challenges have been reported in other storage-cell and sub-grid flood modelling approaches (Neal et al., 2012; Shustikova et al., 2019).

The challenges identified in this study should not be interpreted as limitations of DGGS-based modelling in general. Rather, they highlight the importance of numerical formulations specifically designed for non-orthogonal polygonal meshes. The synthetic benchmark results suggest that future development effort should focus on gradient reconstruction and flux discretisation schemes, while the Carlisle case study demonstrates that useful flood simulations can nevertheless be obtained using the current implementation.

Beyond flood modelling, FloodA5 demonstrates the potential for DGGS-native environmental simulation frameworks. Many environmental models continue to rely on workflows that repeatedly transform data between raster, vector, and computational mesh representations, introducing complexity and potential sources of error (Liao et al., 2025). By maintaining a single hierarchical spatial framework throughout mesh generation, terrain processing, simulation, storage, and visualisation, DGGS-based approaches offer a promising alternative. Although significant numerical challenges remain, particularly for hydrodynamic routing on non-orthogonal meshes, the results suggest that DGGS may provide a useful foundation for future environmental digital twins, multi-scale modelling systems, and integrated geospatial analysis workflows.

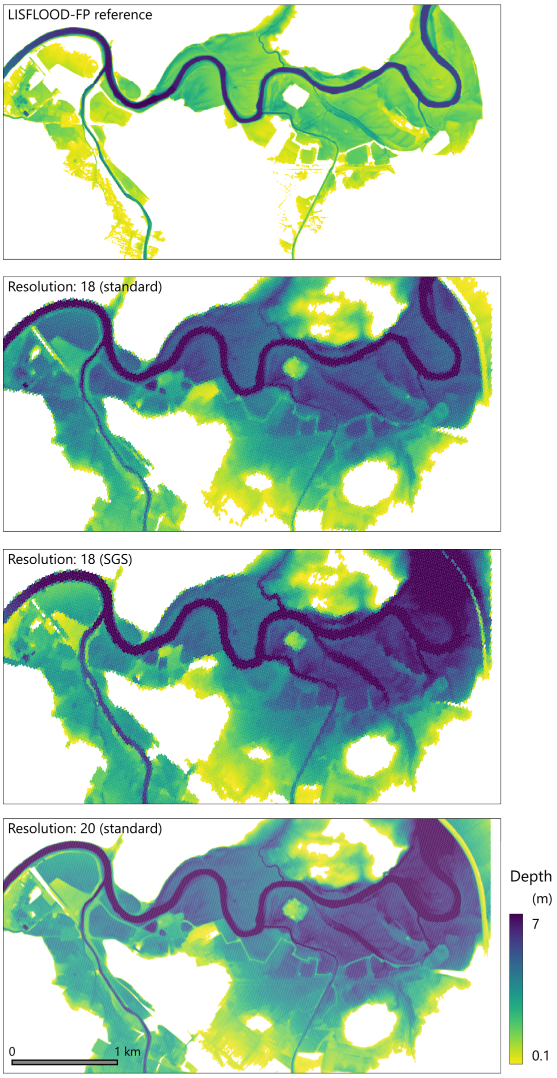

Figure 6. Predicted depth at flood peak for the reference simulation (LISFLOOD-FP) and each model. Note that depth for the SGS model shows maximum cell depth, whereas the reference and standard simulations are mean cell depth.

7. Conclusions

This paper presented FloodA5, an open-source flood modelling framework built on the A5 equal-area pentagonal DGGS. The framework integrates DGGS-native mesh generation, terrain processing, hydrodynamic simulation, and visualisation within a unified workflow. Evaluation using synthetic benchmark problems and the Carlisle 2005 flood event demonstrated that physically plausible flood simulations can be produced on an equal-area pentagonal DGGS and that model outputs show reasonable agreement with a LISFLOOD-FP reference simulation.

The experiments also revealed two important methodological challenges. First, the current treatment of non-orthogonal gradients introduces directional routing bias. Second, the SGS formulation does not adequately represent internal topographic barriers. These findings identify key priorities for future development and highlight the importance of numerical schemes specifically designed for non-orthogonal polygonal meshes. FloodA5 demonstrates the feasibility of DGGS-native flood modelling while providing a platform for future research into hydrodynamic simulation on hierarchical global grid systems.

References

Bates, P. D., Horritt, M. S., Fewtrell, T. J., 2010. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology*, 387(1–2), 33–45.

Bezanson, J., Edelman, A., Karpinski, S., Shah, V. B., 2017. Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1), 65–98.

Gibb, R. G., Purss, M. B., Sabeur, Z., Strobl, P., Qu, T., 2022. Global reference grids for big earth data. *Big Earth Data*, 6(3), 251–255.

Halton, J. H., 1960. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1), 84–90.

Liao, C., Engwirda, D., Cooper, M. G., Li, M., Fang, Y., 2025. Discrete global grid system flow routing datasets in the Amazon and Yukon basins. *Earth System Science Data*, 17(5), 2035–2062.

Neal, J. C., Villanueva, I., Wright, N., Willis, T., Fewtrell, T., Bates, P. D., 2012. How much physical complexity is needed to model flood inundation? *Hydrological Processes*, 26(15), 2264–2282.

Palmer, F., contributors, 2026. A5Geo: Global, equal-area, millimeter-accurate geospatial index. https://github.com/felixpalmer/a5, accessed:2026-06-15.

Robertson, C., Chaudhuri, C., Hojati, M., Roberts, S. A., 2020. An integrated environmental analytics system (IDEAS) based on a DGGS. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162, 214–228.

Shaw, J., Kesserwani, G., Neal, J. C., Bates, P. D., Sharifian, M. K., 2021. LISFLOOD-FP 8.0: the new discontinuous Galerkin solver for model acceleration by code parallelisation. *Geoscientific Model Development*, 14(6), 3577–3602.

Shustikova, I., Domeneghetti, A., Neal, J. C., Bates, P., Castellarin, A., 2019. Comparing 2D capabilities of HEC-RAS and LISFLOOD-FP on complex topography. *Hydrological Sciences Journal*, 64(14), 1769–1782.

Tan, Y., contributors, 2026. ArchGDAL.jl: A Julia wrapper for GDAL. https://github.com/yeesian/ArchGDAL.jl, accessed: 2026-06-15.

Weller, H., 2014. Non-orthogonal version of the arbitrary polygonal C-grid and a new diamond grid. *Geoscientific Model Development*, 7(3), 779–797.

Wilson, M., 2026. FloodA5: A 2D surface-water flood model built on the A5 pentagonal DGGS. https://github.com/MatthewDWilson/FloodA5/, accessed:2026-06-15.

Wilson, M. D., Coulthard, T. J., 2023. Tracing and visualisation of contributing water sources in the LISFLOOD-FP model of flood inundation (within CAESAR-Lisflood version 1.9 j-WS). *Geoscientific Model Development*, 16(9), 2415–2436.