A Scalable Graph Representation of Terrain for Watershed Delineation

Joseph Buchholz and Alan Qiu

December 9, 2025

1 Abstract

This project investigates whether triangulated irregular networks (TINs), specifically right triangulated irregular networks (RTINs), can serve as a scalable and accurate alternative to traditional raster-based digital elevation models (DEMs) for hydrological analysis. Using Pymartini to generate multi-resolution RTINs, we implemented TIN-based algorithms to compute lines of the steepest descent, drainage networks, and watershed boundaries. We then compared these TIN-derived watersheds to those produced by standard raster methods such as D8 and D∞ by evaluating both visual and quantitative similarity. While RTINs performed well in some cases and offer advantages in geometric accuracy and data reduction, the overall Jaccard similarity scores and RMSE values highlight key limitations in mesh error and algorithm robustness. Regardless, the results show that scalable TIN representations remain a promising direction for improving hydrologic modeling efficiency and adaptability.

2 Introduction

Modeling how water moves across terrain is a fundamental problem in hydrology. It has close ties with flood risk analysis, water resource management, and water quality assessment. Traditionally, most approaches rely on digital elevation models (DEMs) represented as a rectangular grid of pixels. Flow algorithms have been developed to compute water flow direction over these grids. However, two problems can arise when working with raster DEMs. First, the uniform, gridded nature of raster DEMs is not found in natural landscapes, making them ill-suited for modeling terrain. Additionally, this means higher-resolution DEMs are typically needed for better accuracy, driving up computational costs. So, the core goal of the project is to find a method that computes flow paths, drainage networks, and watershed boundaries from DEMs in a way that is both geometrically accurate and computationally efficient.

The goal is important because of how water flow dictates not only the occurrence of floods, but also the transportation of pollutants which is important in determining water quality. Inaccurate modeling can lead to poor predictions, but more accurate models tend to indicate worse performance.

The basic approach that we followed was to use a triangulated irregular network (TIN) representation of terrain instead of a typical raster grid. Starting from the raster DEM, we generated a multiresolution TIN to compute flow paths as lines of steepest descent across the triangular faces. Unlike the grid-based methods, TIN methods are able to leverage the more continuous characteristic of TIN surfaces to reduce directional bias allowing for a better representation of real-world terrain. TIN methods are also able to use fewer data points for less detailed terrain unlike their uniform grid counterparts. Additionally, we made use of a hierarchy of TIN approximations, allowing for different choices of scale depending on the particular application. In short, this project explores in which ways TINs can be better alternatives to raster DEMs for hydrological analysis.

3 Problem Definition

This work focuses on the problem of accurately modeling surface water flow, drainage networks, and watershed boundaries from digital elevation models in a way that is more computationally efficient and geometrically accurate compared to traditional raster-based methods. Many other approaches are based on regular grids leading to water flow that is restricted to a small set of predefined directions which can cause artifacts and reduced accuracy in complex terrain. As a result, water movement across landscapes can be misrepresented leading to unreliable and inaccurate results. This is where TIN-based methods can be useful.

TINs have been successfully used by a variety of studies in relation to hydrological applications (Jones et al., 1990; Freitas et al. 2016; Chen and Zhou, 2013). This work evaluates some of these previously developed methods. One of the questions that this work strives to answer is whether specifically a right-triangulated irregular network (RTIN), which is just a TIN consisting only of right triangles, can be a good DEM for hydrological applications. If so, then an algorithm developed by Evans et al. (2001), which allows for the efficient generation of RTINs, could be used instead of other TIN generation methods. This work also takes a look at scalability to evaluate how different approximation levels affect performance.

Solving the problem is important because of its wide application. Flow routing and watershed delineation are useful for many reasons including water resources management and runoff estimation. The use of machine learning for flood prediction is a particularly new area (Sarkar et al., 2025), and it could potentially benefit from a scalable graph terrain representation rather than a raster DEM. A machine learning model would not need all the detail that is present in a raster DEM, and thus a TIN which approximates the original DEM in a fewer number of data points could significantly improve performance.

4 Algorithms

A variety of different existing algorithms were used in conjunction to solve the problem. First, an open-source library called Pymartini was used to generate a triangulated irregular network from a raster-based digital elevation model (Barron et al., 2025). Pymartini is based on an algorithm developed by Evans et al. (2001) which allows for a multi-scale generation of a RTIN. The algorithm generates a hierarchy of meshes at different approximation levels which can then be chosen according to the desired level of accuracy. The scalability of this TIN allows one to pick the best level for a particular application that gives the best speed vs accuracy trade-off.

Once a TIN is generated from a DEM, there are various algorithms that have been developed to solve various hydrological problems. Jones et al. (1990) was one of the first to develop TIN-based algorithms for creating a drainage network and delineating watersheds. We implemented the algorithms developed by Jones et al. (e.g. calculating the line of steepest descent and creating a drainage network) while also implementing some improvements to drainage network calculation outlined by Freitas et al. (2016).

Calculating the line of steepest descent is needed in order to predict where water will flow next. Unlike in a raster-based method where there is a discrete, finite number of choices for the next grid-cell to move to, a TIN has a more continuous nature which requires a more sophisticated technique. This contrasts with the well-known D8 algorithm which just checks a grid-cell's neighbors and picks the lowest one (O'Callaghan and Mark, 1984). When at a triangle, we can calculate the gradient to find the next point that leads us in the direction of steepest descent. The gradient can be easily found by doing some simple manipulation to the equation for the plane defined by the three vertices of a given triangle (Jones et al., 1990). The following is the gradient equation:

$$-\nabla f = \frac{A}{B}\hat{i} + \frac{B}{C}\hat{j}$$

In which A, B, and C are three of the coefficients of the triangle's plane equation:

$$Ax + By + Cz + D = 0$$

We can then take the intersection between this gradient and the triangle's edges to find the next point with which to continue the descent line (Jones et al., 1990). Then at this next point some simple linear algebra can be used to determine whether the line of steepest descent should continue along an edge (channel flow) or across another triangle (overland flow). The line stops once an edge of the TIN is reached or a pit is reached, where a pit is defined as a vertex where all incident edges are sloped toward it.

Once we can calculate the line of steepest descent from a point, we can use that to produce a drainage network (Freitas et al., 2016). This can be done by calculating the line of steepest descent from the centroid of each triangle.

When any triangle is traversed during this process, that triangle is marked as visited and an outlet point is associated with that triangle. During the process of calculating the line of steepest descent, if a triangle is found that has already been visited then the current line of steepest descent is connected to the outlet point of that triangle and the algorithm for the steepest descent line stops (the final line would end up continuing through the outlet point of the visited triangle which has already been computed). After doing this for each triangle, multiple disconnected drainage networks are produced each with an outlet point that represents where all the upstream points drain to.

Once a drainage network is computed, it is easy to delineate the watershed associated with that network. The watershed is just the set of triangles which the drainage network covers. This set of triangles can then be used to calculate the watershed's area, construct a single polygon that represents the watershed, or be used for any other desired application.

5 Implementation

For this project we used the HydroSHEDS version 1.1 core data set (Lehner et al., 2008). HydroSHEDS provides DEMs for the surface of the entire earth, in addition to many other hydrologically oriented data products. For this project we used one of the provided pre-conditioned DEMs at 3 arc-second resolution. Using a pre-conditioned DEM left us with many of the preprocessing steps already completed, and so we were able to focus more on the core algorithms rather than the preprocessing. Though some preprocessing still needed to be done after the TIN generation.

QGIS was used to crop the DEM into smaller testing regions. The main region that was used spanned an area intersecting Canada, Montana, and Idaho (see Figure 1 and Figure 2). After the dataset was put in place, a TIN could be generated.

We used the Python programming language to implement everything. While not as performant as some other languages, it was chosen for its ease of use and the plethora of libraries it provides. The graph terrain representation was generated using Pymarini (Barron et al., 2025). To use Pymartini, the DEM needed to be resized to a square of size $2^k + 1$ (where k is an integer). This is one of the limitations of the algorithm, however the return for this is in the algorithm's speed. Pymarini generates a hierarchy of approximations for the TIN. We tested with error levels of none, 10, 30, and 90 (the error is in the units of the original DEM's pixels, in this case 1 unit $\approx$ 75 meters). Figure 3 depicts the generated TIN at a high error level.

After the TIN is generated, some preprocessing needs to be done before any hydrological algorithms can be performed. While the preconditioned DEM does have some preprocessing done already, it is primarily in preparation for raster-based methods. The generation of a TIN can introduce some additional problems that need to be addressed. The existence of flat triangles is one such problem (Freitas et al., 2016).

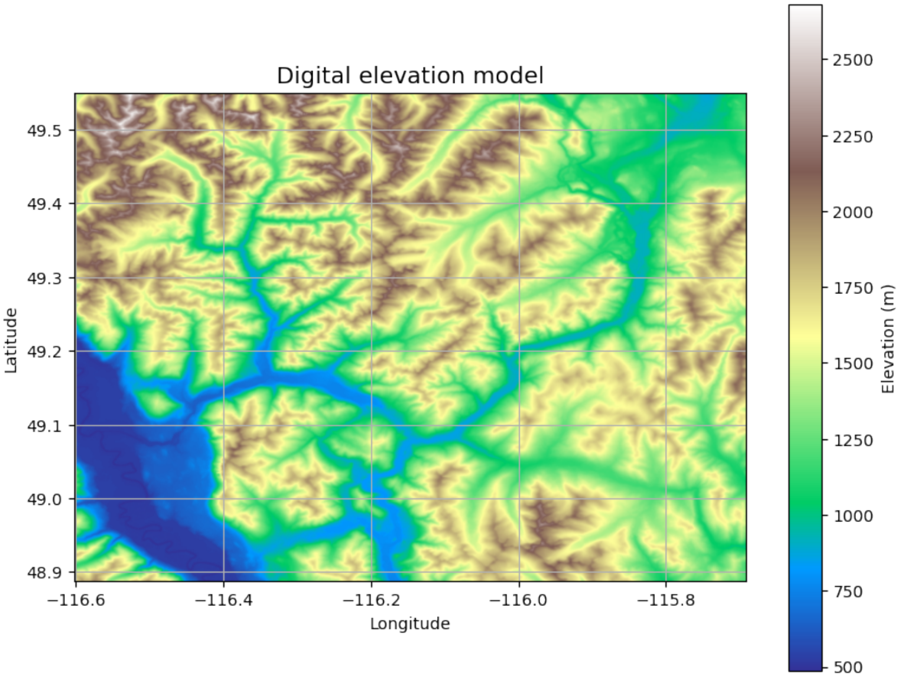


Figure 1: HydroSHEDS digital elevation model

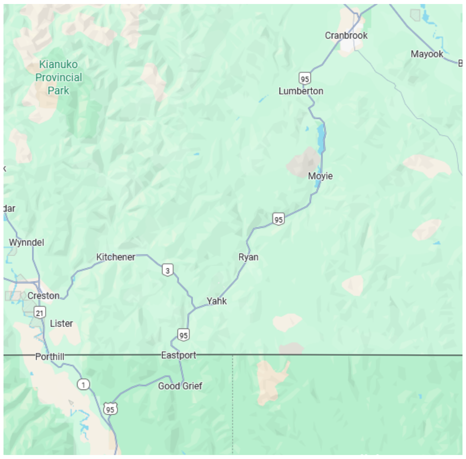


Figure 2: A Google Maps view of the DEM's location

With a preprocessed TIN generated, the lines of steepest descent can be calculated, and these lines can be connected to form a drainage network. The implementation of these algorithms follows the work of Jones et al. (1990)

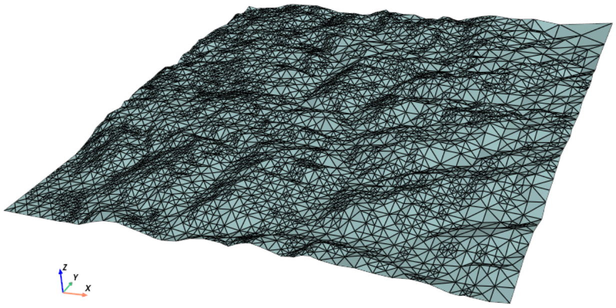


Figure 3: A 3D view of the generated TIN generated at an error level of 90

for lines of steepest descent and Freitas et al. (2016) for drainage network calculation. Only a few modifications and assumptions needed to be made.

Figure 4 depicts the lines of steepest descent from each triangle's centroid on a small section of the generated TIN. The lines end either at a pit or the edge of the TIN section.

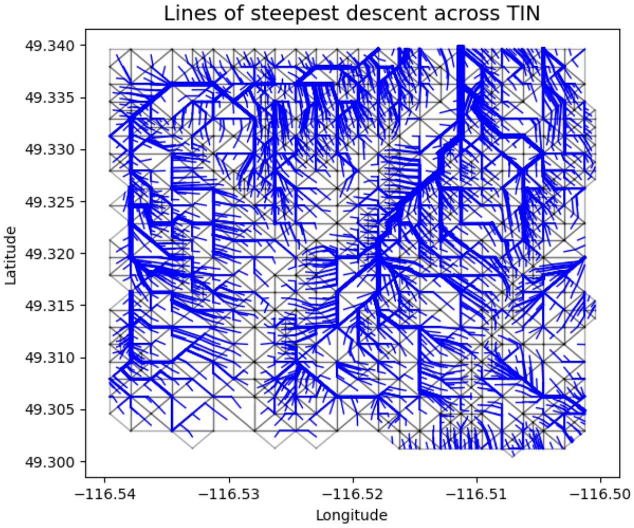


Figure 4: Lines of steepest descent on a small section the generated TIN. Thicker line segments represent the base of longer flow lines.

Connecting the lines of steepest descent together form drainage networks which can be seen in Figure 5. Each color represents a different drainage network where all the lines flow to a single outlet point. With a drainage network calculated, it is easy to then delineate a watershed by finding all the triangles that the drainage network intersects (see Figure 6).

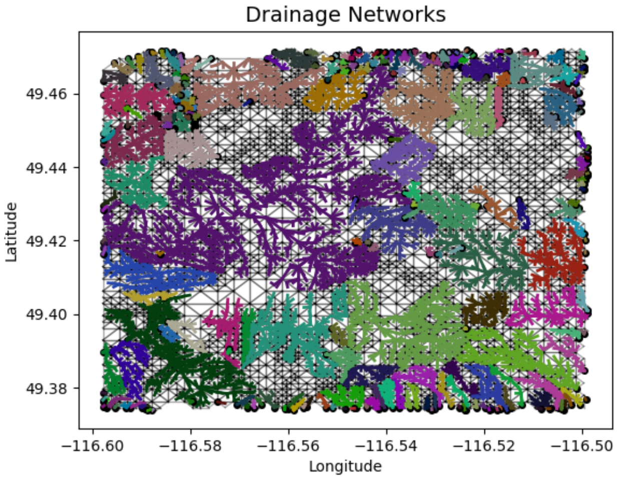


Figure 5: Drainage networks

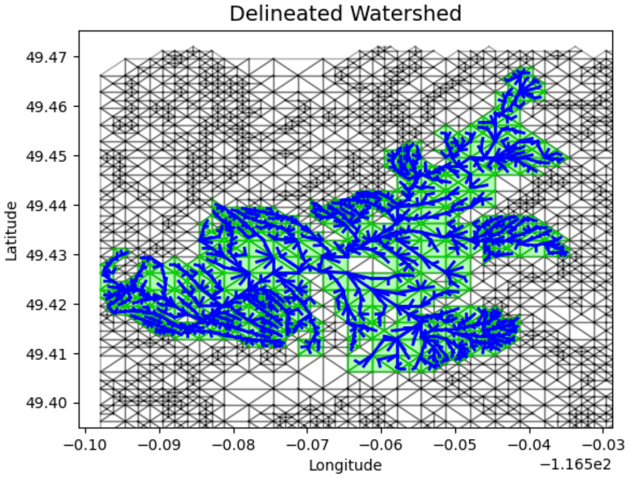
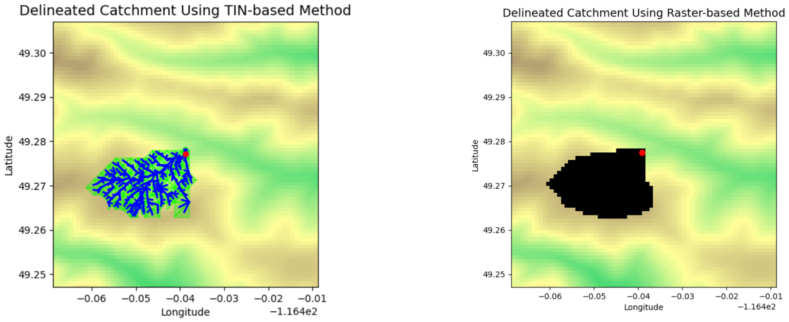


Figure 6: Watershed

To evaluate our methods, we compared our delineated watersheds against



(a) TIN method at error level 10

(b) Raster method

Figure 7: Comparison of TIN and raster watershed delineation methods with the DEM visualized in the background. Jaccard index: 0.84

those generated via a raster-based method. For this we used the open source library Pysheds which implements both D8 and D∞ (Bartos, 2020). We used D8. Comparing the two methods side by side in Figure 7, visually one can see that our method does a pretty good approximation of the raster-based watershed (which is assumed to correspond to the real watershed). In addition to visual checks, we evaluated our method quantitatively by using a few metrics such as the Jaccard index and taking benchmarks to evaluate the algorithm's performance.

6 Results and Discussion

Figure 8 shows the TIN method in green overlaid with the raster based method in black (the red dots represent the outlet points). Visually, one can see that in this case the watersheds coincide very nicely, indicating that the method is performing well. For a quantitative measure of how similar the two methods are, we used the Jaccard index. The Jaccard index for Figure 8 is 0.77, which is pretty good. Other watershed pairs do not always have this good of a score.

To further evaluate the average performance of our methods, we ran the algorithm to delineate multiple watersheds over different TIN mesh error levels. Table 1 shows the results of this evaluation. The average Jaccard index shows how well the TIN methods work on average. For reasonable mesh error levels (i.e. < 30), the Jaccard index hovers from about 0.2 to 0.3. While this does show that there is some similarity, it is not ideal. It shows that there is quite a large discrepancy between the watersheds generated with a TIN and those generated using D8.

RMSE tells a similar story here. It is the error in the area of the watershed (in sq. miles) generated by the TIN method versus the raster method (i.e. the raster method is treated as the true value and the TIN is treated as the predicted

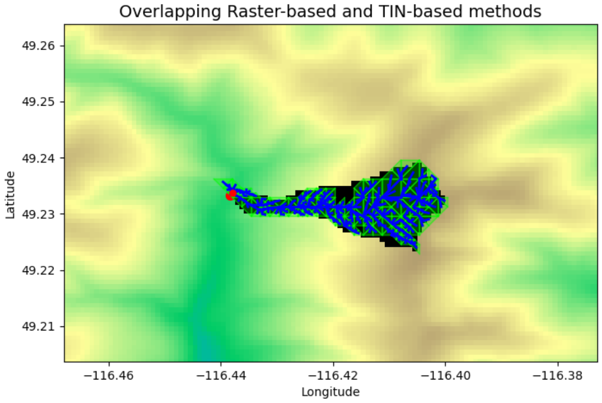


Figure 8: Overlapping raster-based and TIN-based methods for visual comparison. TIN mesh error level: 10. Jaccard index: 0.77

Table 1: Watershed delineation results for 4 different error levels over 100 watersheds. Only watersheds containing a minimum of 30 triangles were kept (e.g. "Runs" is less than 100). "Runs" is the number of watersheds that were averaged over. "RMSE Area" is the RMSE for watershed area in sq. miles.

Mesh Error Level	Runs	Average Jaccard Index	RMSE Area
0	12	0.297	0.877
10	17	0.267	58.420
30	36	0.234	40.026
90	31	0.074	211.748

value). While not bad for a mesh with no error level, the RMSE increases dramatically when the mesh error level increases. This is not too surprising, however. For example, with the mesh error level of 90, it corresponds to an error of approximately 4.2 miles. Squaring that gives 17.6 sq. miles which is comparable in magnitude to 211.7 sq. miles. Thus, these error levels are not all that unexpected (though of course they could still be much better).

There are a few potential reasons for the low Jaccard index. For one, there is at least one problem with the algorithm's implementation that leaves spots uncovered by the drainage network (see Figure 5). Unfortunately, this implementation problem could not be solved in the time available. This could potentially be the issue with watersheds such as the one in Figure 9. Errors this big can of course drastically offset the average.

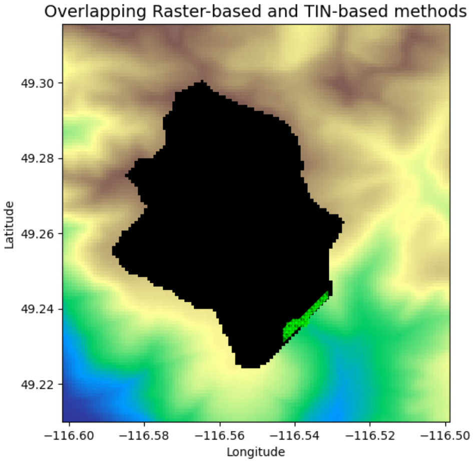


Figure 9: Seemingly failed watershed delineation.

Preprocessing is an important step to ensure that watersheds are delineated correctly. So, this is another potential problem with our methods, as we did only some of the minimal preprocessing required and did not implement some of the more sophisticated preprocessing techniques such as TIN pit filling outlined by Freitas et al. (2016). Pits can be a big problem as they will often make flow lines stop early. While some pits are desirable because they do actually affect the real flow directions, any pits that are too small or of artificial origin should ideally be removed. Since we did not implement any pit filling algorithms, this may be one of the reasons why we see some of these low Jaccard values.

Another thing to consider is that there are most certainly inconsistencies between the watersheds delineated by D8 and the actual real watersheds (which would need to be verified manually). Thus, it is important to realize that the output of Pysheds is not a perfect evaluation method. Ideally, our TIN method would be tested against manually verified delineated watersheds. Finally, of course much of the error observed is due to the error inherent in generating the mesh.

We also measured the performance of our methods. The average time for running the drainage network calculation on the generated TIN was 123.4 seconds for a mesh containing 44052 triangles. This result was determined by averaging over 5 trials. This may seem much slower than the Pysheds counterpart which executes almost instantaneously (and admittedly it is), but there are a few things to consider here. First, our method essentially precomputes all the watersheds at the time of creating the drainage networks. On the other hand, Pysheds only computes a watershed when one is desired. Thus, to give a fair comparison one would have to run Pyshed's watershed delineation for each watershed in the data set (and no attempt was made to do this).

Additionally, there are likely many optimizations that we overlooked which could be a cause of the slowness. But, in the end it is clear that of course the TIN method would be at least somewhat slower than D8 because of the complexity involved in TIN drainage network calculation. Thus, it is important to consider the tradeoff between datapoint density and performance. This is why having multiple approximation levels is useful. Since as the error in approximation increases, performance also increases. This consideration is of course heavily dependent on the particular application.

Overall, this does show that RTINs are a viable alternative to representing terrain within a hydrological framework. Whether this method can be improved enough to make it a desirable alternative is still an open question that needs more work. Another thing to be observed is that for mesh error levels 30 or under it seems that the Jaccard index is relatively stable. This may be an indicator that only a few important data points are really needed to effectively represent terrain within a hydrological framework. This lends cadence to the idea of using TINs as an effective data reduction mechanism while maintaining accuracy.

7 Related Work

A few different kinds of DEMs exist. While some experimental DEMs, such as hexagonal grids, have been experimented with (Liao et al., 2020), the primary ones in use are rectangular pixel grids and TINs. Algorithms for calculating flow paths on rectangular grid DEMs have been studied the most. First, O'Callaghan and Mark (1984) introduced their D8 algorithm. After the initial D8 algorithm, many others followed up and made many improvements (Jenson and Domingue, 1988; Tarboton, 1997; Paik, 2008). Notably, Tarboton (1997) introduced D∞ which expanded upon D8 by allowing more than just the 8 neighbor directions. This was done by translating the 8 neighbor pixels into 8 right triangles, and then dispersing the flow based on the triangle's slopes. In a way, this is similar to how flow paths are calculated on a TIN in this project.

While the calculation of flow paths on a TIN DEM has not been studied as extensively, there has still been a large amount of work done in the area. Jones et al. (1990) pioneered the area by developing algorithms for calculating lines of steepest descent, channel networks, and watersheds on TINs. Freitas et al. (2016) made some improvements on the methods presented there for calculating drainage networks and improved the preprocessing methods. Zhou and Chen (2011) have experimented with creating a TIN crafted precisely for hydrological calculations. Additionally, scalability similar to that of Pymartini was added in a follow up paper (Chen and Zhou, 2013). It too allows for different levels of approximation to be picked for a particular application. Though Chen and Zhou used a different method than the one used in Pymartini and tailored their solution specifically to retain hydrologically important terrain features.

The approach that we took to the problem is slightly different from these others in a few ways. For one, we specifically used an RTIN as our terrain model. Using the algorithm by Evans et al. (2001), this brings in some speed in generating the TIN, but it also yields some unique challenges such as restrictions on DEM size and looser approximations. Additionally, we evaluated the use of different approximation levels in watershed delineation.

8 Conclusion

This work explored the use of triangulated irregular networks as an alternative terrain representation for hydrological analysis, focusing on flow routing, drainage network extraction, and watershed delineation. Our results show that TIN-based methods can approximate raster-derived watersheds while offering clear advantages in scalability and data reduction, but the overall quantitative performance reveals some limitations. Lower Jaccard similarity scores and higher RMSE values indicate that mesh approximation errors, incomplete preprocessing, and implementation issues have significantly impacted accuracy. Despite these challenges, the findings demonstrate that scalable TIN representations remain a promising alternative terrain representation for hydrologic analysis. Future work could address the current shortcomings by implementing more sophisticated preprocessing (for example pit filling), refining the drainage network algorithm, optimizing performance, and validating results against manually verified watershed boundaries or higher quality datasets.

9 Disclosure

During the course of this project, we made use of generative AI tools to facilitate the code writing process. All generated code was checked for correctness and revised when necessary. Code completion was the primary tool used to speed up development. Mass code generation (i.e. anything more than a few lines) was rarely used.

10 Bibliography

Barron, K., Chemla J., & Brochart, D. (2025). *Pymartini* [Source code]. https://github.com/kylebarron/pymartini

Bartos, M. (2020). *Pysheds: simple and fast watershed delineation in Python* [Source code]. https://github.com/mdbartos/pysheds

Chen, Y., & Zhou, Q. (2013). A scale-adaptive DEM for multi-scale terrain analysis. *International Journal of Geographical Information Science*, *27*(7), 1329-1348. https://doi.org/10.1080/13658816.2012.739690

Evans, W., Kirkpatrick, D., & Townsend, G. (2001). Right-triangulated irregular networks. *Algorithmica*, *30*(2), 264-286. https://doi.org/10.1007/s00453-001-0006-x

Freitas, H. R. de A., Freitas, C. da C., Rosim, S., & Oliveira, J. R. de F. (2016). Drainage networks and watersheds delineation derived from

TIN-based digital elevation models. *Computers & Geosciences*, *92*, 21-37. https://doi.org/10.1016/j.cageo.2016.04.003

Jenson, S. K., & Domingue, J. O. (1988). Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing*, *54*(11), 1593-1600. https://pubs.usgs.gov/publication/70142175

Jones, N. L., Wright, S. G., & Maidment, D. R. (1990). Watershed delineation with triangle-based terrain models. *Journal of Hydraulic Engineering*, *116*(10), 1232-1251. https://doi.org/10.1061/(asce)0733-9429(1990)116:10(1232)

Lehner, B., Verdin, K., & Jarvis, A. (2008). New global hydrography derived from spaceborne elevation data. *Eos, Transactions American Geophysical Union*, *89*(10), 93-94. https://doi.org/10.1029/2008eo100001

Liao, C., Tesfa, T., Duan, Z., & Leung, L. R. (2020). Watershed delineation on a hexagonal mesh grid. *Environmental Modelling & Software*, *128*, 104702. https://doi.org/10.1016/j.envsoft.2020.104702

O'Callaghan, J. F., & Mark, D. M. (1984). The extraction of drainage networks from digital elevation data. *Computer Vision, Graphics, and Image Processing*, *28*(3), 323-344. https://doi.org/10.1016/s0734-189x(84)80011-0

Paik, K. (2008). Global search algorithm for nondispersive flow path extraction. *Journal of Geophysical Research: Earth Surface*, *113*(F4). https://doi.org/10.1029/2007jf000964

Sarkar, A., Hakimi, A., Chen, X., Huang, H., Lu, C., Demir, I., & Jannesari, A. (2025). HydroGAT: distributed heterogeneous graph attention transformer for spatiotemporal flood prediction. *The 33rd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '25).* https://doi.org/10.1145/3748636.3764172

Tarboton, D. G. (1997). A new method for the determination of flow directions and upslope areas in grid digital elevation models. *Water Resources Research*, *33*(2), 309-319. https://doi.org/10.1029/96wr03137

Zhou, Q., & Chen, Y. (2011). Generalization of DEM for terrain analysis using a compound method. *ISPRS Journal of Photogrammetry and Remote Sensing*, *66*(1), 38-45. https://doi.org/10.1016/j.isprsjprs.2010.08.005

11 Appendix

The code for this project can be found in a GitHub repository at the following link: https://github.com/JosephBuchholz/cpts-475-watershed-delineation