A Discussion: Parameter Estimation

In Sec. 4.4, we conducted a parameter sensitivity analysis on the two datasets to determine λ, running a sequence of exponentially increasing λ values. This analysis provides a systematic approach for identifying the optimal λ. In this section, we further discuss criteria to help users estimate an appropriate value for λ, focusing in particular on determining its minimum value.

Estimating the minimum value of λ. We begin by defining the terms used in the discussion. Let N denote the total data volume and R the number of ranks. The parameter λ determines which attachment points are exchanged: specifically, only those whose interior forest volume exceeds λ are included. Let α be the number of attachment points received from other ranks. An upper bound on α for each rank is

$$\frac{N - N/R}{\lambda + 1},$$

where $N - N/R$ represents the maximum external data volume for a rank, and $\lambda + 1$ specifies the minimum interior forest volume of attachment points to be exchanged. This bound is loose, as part of the data volume is already represented in the shared contour tree structure.

The first criterion for determining λ, referred to as the *memory criterion*, ensures that sufficient memory is available for analytical computations. Based on this criterion, we estimate the minimum value of λ through two runs on a subvolume of the data, followed by a test run on the full dataset.

For example, to estimate the minimum λ satisfying the memory criterion for the 2048^3 volume of the Nyx dataset, we first run the framework on a 1024^3 subvolume with two values of λ: 0 and 100. The run without pre-simplification ($\lambda = 0$) consumes 574.66 GiB of memory, whereas the run with $\lambda = 100$ uses 439.17 GiB. In parallel, the number of attachment points drops from $697{,}320{,}285$ to $1{,}288{,}810$. This reduction implies that processing approximately 6.96×10^8 attachment points requires about 135.49 GiB of memory, or roughly 209.02 bytes per attachment point.

Next, we perform a test run on the full 2048^3 volume using a large λ value (e.g., $\lambda = 10^5$) to eliminate most attachment points and measure memory consumption. If this run fails due to insufficient memory, the hardware configuration is unlikely to be viable for the dataset, regardless of λ. If successful, we record the peak memory usage for any single rank—in our case, 133.26 GiB (note that this is not the total memory usage across all ranks). The remaining available memory must then be sufficient to handle attachment point computations.

In this example, $N = 2048^3$, $R = 16$, and each rank (i.e., compute node) has 512 GB of available memory. Since the upper bound on the number of attachment points is $\frac{N - N/R}{\lambda+1}$, and the memory required for attachment point computation is 209.02 bytes per voxel on average, we require

$$209.02\,\alpha < 209.02 \frac{N - N/R}{\lambda + 1} < 512 \times 10^9 - 133.26 \times 1024^3.$$

From this, we estimate the minimum λ for this example to be 4.

The second criterion concerns communication overhead, which we refer to as the *communication criterion*. Pre-simplification reduces the number of attachment points exchanged during communication, thereby mitigating scalability limits. As shown in Sec. 4.4, both the attachment points and the shared contour tree structure contribute to this overhead. With increasing λ, the number of attachment points decreases, and eventually the shared contour tree structure becomes the dominant factor. For optimal scalability, our goal is to reduce the number of attachment points to be comparable to, or smaller than, the size of the shared contour tree, which has been shown [26, 7, 29] to be bounded by $O(N^{2/3})$ for 3D data. This implies that λ should be on the order of $\Omega(N^{1/3})$ for optimal scalability.

Limitation. We conclude by discussing the limitations of our approach for estimating λ. First, the memory criterion requires recording statistics such as the number of attachment points and the memory usage for each rank. Although our implementation includes logging functionality, this method still entails additional effort. Second, the estimated minimum λ for the first criterion is likely higher than the true minimum, as it is based on the worst-case distribution of attachment points. Third, for the communication criterion, constant factors in the computation make it difficult to determine a precise minimum value of λ.

Parameter choices. While pre-simplification significantly reduces communication overhead and enables the processing of much larger datasets, it also removes some small-volume features that may be important in certain tasks or data contexts. To preserve such features, we aim to choose λ as small as possible, subject to satisfying the memory criterion (and optionally the communication criterion), and ensuring that $\lambda < \Lambda$. However, if the estimated λ is substantially larger than the expected size of the smallest relevant features, pre-simplification may not be suitable for the application.

Currently, our implementation supports only a single, global λ for pre-simplification as a means of reducing the number of attachment points in the computation. Since the augmentation step can be performed on arbitrary subsets of attachment points [29], it would be possible to customize λ for different subareas or subvolumes of the data, depending on the features of interest, or even to selectively preserve specific features—an extension we leave for future work.