PROJECT PLAN

BACKGROUND

Space-filling curves are a favourable method of partitioning multi-dimensional data for parallelisation algorithms. When measuring their surface to volume ratio on regular grids, it has analytically been shown that space-filling curves provide a partition as efficient as an n-sphere besides a constant. For the rest of the paper, the following conditions are set as:

Grid: In this project, adaptive Cartesian grids will be used as the underlying space for the space-filling curves to be generated upon.

Partition: A partitioning algorithm will be used to fragment the grids containing the space-filling curves into three segments.

Surface-to-volume ratio: The average length of the boundary of a segment over the number of cells it contains. The lower the ratio, the better the partition. In a two-dimensional case, this would be the average length of the edge touching another segment over the area of the segment.

OBJECTIVES

Currently, a quantitative formula for calculating such ratios on adaptive grids is non-existent to the best of my knowledge. Through a brute-force approach, this project aims to:

- Find out whether space-filling curve partition algorithms are quasi-optimal on adaptive grids.
- Formulate meaningful estimates for the surface to volume ratio of partitions on adaptive grids.
- Derive tighter bounds and averages for the surface to volume ratio of partitions on adaptive grids.

METHOD

A brute force space-filling curve partition algorithm will be used to calculate the surface to volume ratio on adaptive grids. The algorithm will start on a basic grid and develop into exploring finer and finer meshes. However, as grids become increasingly intricate, the number of possible partitions intensifies. Therefore, the algorithm will require ‘short-cutting’ methods to simplifying the brute force algorithm using patterns such as symmetry in order to improve the runtime performance. By increasing the efficiency of the algorithm, finer meshes containing masses of cells will also be to return their surface to volume ratios.

RESULTS

The results of all partitions should be stored in a table and graph. By investigating finer meshes, these results should assist in outlining any correlations or patterns within the data and help solve the objectives of the project.

CONCLUSION

Upon succession, the data acquired could be used in future research directed towards finding a quantitative formula for calculating such a constant in the following formula: $s \leq C_{part} \cdot v^{\frac{d-1}{d}}$, where s, v and d represent the surface, volume and dimension respectively.

PRELIMINARY PREPARATION

- Develop a solid understanding of current formulae and theorems behind the surface to volume ratio of space-filling curve partitioning.
- Understand the algorithms behind generating space-filling curves on an adaptive grid.
- Create an algorithm that can produce quadtree grids in a suitable programming language

DELIVERABLES

BASIC

- Retrieving the surface to volume ratio of at least one two-dimensional space-filling curve partition algorithm on an adaptive Cartesian grid using brute force such that the algorithm:
    - can provide a result for the most effective partition split across 3 processors within reasonable time for a given quadtree grid containing fewer than 64 cells.
    - can theoretically provide results for any quadtree grid given.
    - can generate a graph comparing the surface to volume ratios produced by various quadtree grids.

INTERMEDIATE

- Improve the efficiency of the brute force algorithm by introducing techniques to reduce the computational time of calculating the surface to volume ratio (e.g. using symmetry and recursion properties of the quadtree traversals produced) and use this to provide solutions to the objectives.

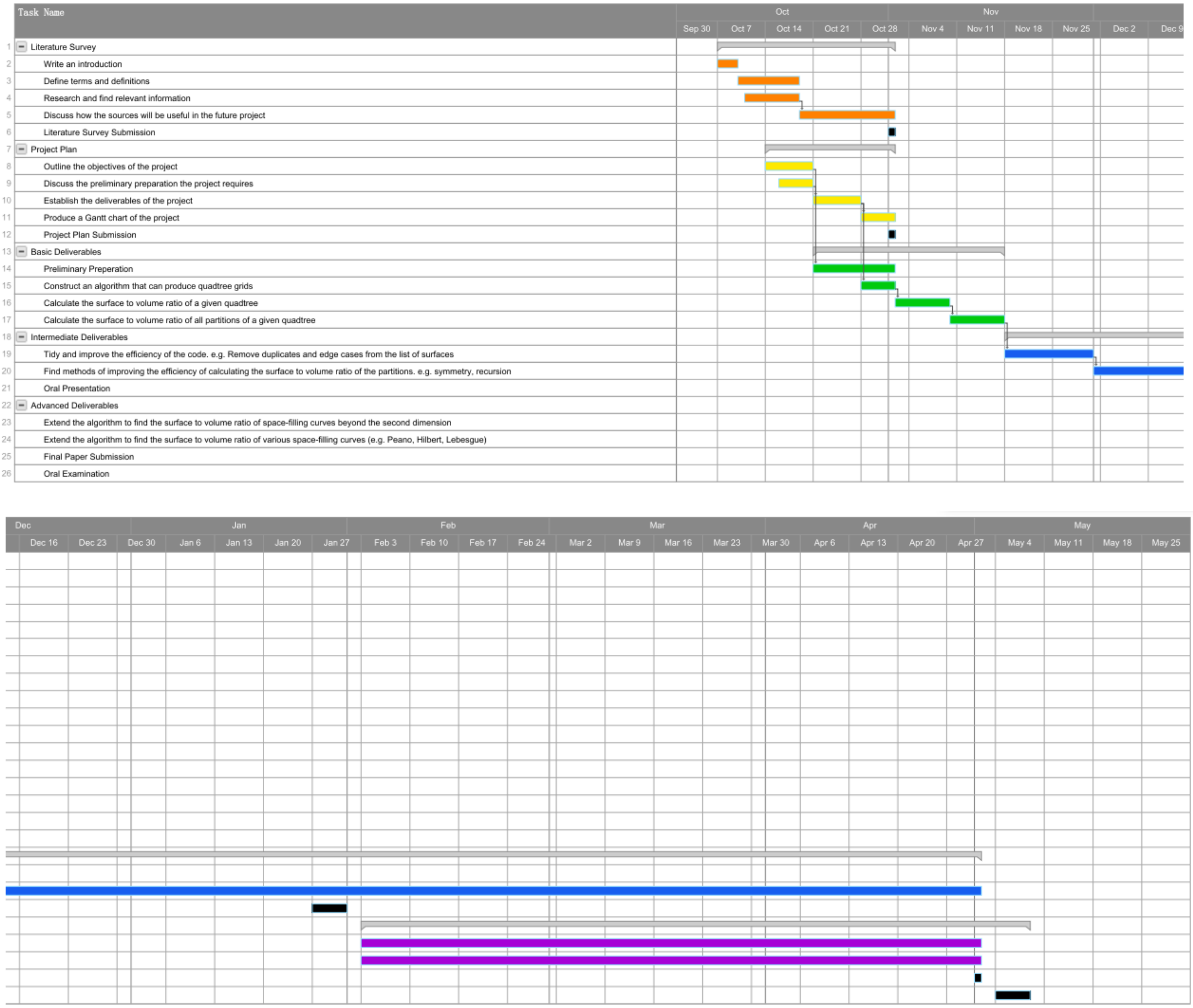
ADVANCED

- Extend the application to incorporate space-filling curve partitions beyond the second dimension.
- Extend the application to find the surface to volume ratio on adaptive grid partitions based on different types of space-filling curves.

SOFTWARE DEVELOPMENT METHODOLOGY

Since the method of the project is to find trends in space-filling partitions on increasingly detailed meshes, one of the main focuses of the project is to reduce the runtime of the partitioning algorithm. Aside from it being difficult to approximate the timeframe of implementing ‘short cuts’ to the algorithm, there should be little scope for change, making it suitable to take a plan-driven approach. In particular, the spiral model will be adopted. This process will allow for multiple prototype algorithms to be built. Presumably, the results yielded from testing the prototypes will help find patterns and develop new ideas for increasing the efficiency of the algorithm.

GANTT CHART


Task Name
Oct
Nov
Sep 30
Oct 7
Oct 14
Oct 21
Oct 28
Nov 4
Nov 11
Nov 18
Nov 25
Dec 2
Dec 9
1 Literature Survey
2 Write an introduction
3 Define terms and definitions
4 Research and find relevant information
5 Discuss how the sources will be useful in the future project
6 Literature Survey Submission
7 Project Plan
8 Outline the objectives of the project
9 Discuss the preliminary preparation the project requires
10 Establish the deliverables of the project
11 Produce a Gantt chart of the project
12 Project Plan Submission
13 Basic Deliverables
14 Preliminary Preperation
15 Construct an algorithm that can produce quadtree grids
16 Calculate the surface to volume ratio of a given quadtree
17 Calculate the surface to volume ratio of all partitions of a given quadtree
18 Intermediate Deliverables
19 Tidy and improve the efficiency of the code. e.g. Remove duplicates and edge cases from the list of surfaces
20 Find methods of improving the efficiency of calculating the surface to volume ratio of the partitions. e.g. symmetry, recursion
21 Oral Presentation
22 Advanced Deliverables
23 Extend the algorithm to find the surface to volume ratio of space-filling curves beyond the second dimension
24 Extend the algorithm to find the surface to volume ratio of various space-filling curves (e.g. Peano, Hilbert, Lebesgue)
25 Final Paper Submission
26 Oral Examination
Dec
Jan
Feb
Mar
Apr
May
Dec 16
Dec 23
Dec 30
Jan 6
Jan 13
Jan 20
Jan 27
Feb 3
Feb 10
Feb 17
Feb 24
Mar 2
Mar 9
Mar 16
Mar 23
Mar 30
Apr 6
Apr 13
Apr 20
Apr 27
May 4
May 11
May 18
May 25