Project #1

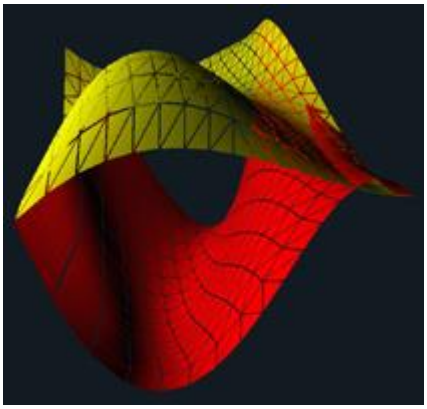
OpenMP: Numeric Integration with OpenMP

100 Points

Due: April 20

Introduction

Bézier surfaces are a way of sculpting shapes. Because they are analytically defined, they can be computed (and rendered) to any precision you want. In this project, we are going to take these two Bézier surfaces:



and use numerical techniques to find the volume between the bottom surface (red) and the top surface (yellow).

Using some number of subdivisions in both X and Y, NUMNODESxNUMNODES, take $NUMNODES^2$ height samples.

We will think of each height sample as sitting on a 2D tile. That really makes that height sample act as a volume where the tile is extruded vertically from the bottom to the top.

The tiles in the middle of the floor are full-sized tiles. Tiles along the edges are half-sized. Tiles in the corners are quarter-sized. The volume contribution of each extruded height tile needs to be weighted accordingly. The logic of this is for you to figure out.

Requirements

- Using OpenMP, compute the total volume betwen the two surfaces.
- Use a variety of number of subdivisions (NUMNODES). Pick at least 8 different ones.
- Use a variety of number of threads (NUMT). You must use at least 1, 2, and 4.
- Record the data in units of something that gets larger as speed increases. Joe Parallel used "MegaHeights Computed Per Second", but you can use anything that makes sense.
- From the speed-up that you are seeing, use the "Inverse Amdahl's Law" to determine the Parallel Fraction for this application.
- From the Parallel Fraction, determine what maximum speed-up you could *ever* get, even with a million cores.
- Your commentary write-up (turned in as a PDF file) should include:

    1. Tell what machine you ran this on
    2. What do you think the actual volume is?
    3. Show the performances you achieved in tables and graphs as a function of NUMNODES and NUMT
    4. What patterns are you seeing in the speeds?
    5. Why do you think it is behaving this way?
    6. What is the Parallel Fraction for this application, using the Inverse Amdahl equation?
    7. Given that Parallel Fraction, what is the maximum speed-up you could *ever* get?