

Problem Statement:

The “main.cpp” does the following using the classes and data structures that you have been provided:

1. Call `setupModel()`
 - this function initializes a grid (nodes and elements) using the data structure that you were provided. The main argument used is a global variable `DX_GLOBAL` which specifies the uniform grid resolution. (`DX_GLOBAL` = 50 for the first figure above and `DX_GLOBAL`=5 for the second figure)
2. Call `AllocateArrays()`
 - dynamically allocate the required arrays;
3. Call `PopulateArrays()`
 - populates a two dimensional array (called *area*) based upon the area and node numbers of each triangular element;
 - populates an array called *X* based upon the triangle element normal and node numbers of each triangular element
4. Call `MatrixMatrix()`
 - performs matrix multiplication $ATA = (\text{transpose of area}) * \text{area} = \text{area}^T * \text{area}$;
5. Call `MatrixVector()`
 - calculates $Y = ATA * X$;
6. Calculate the l2-norm (length) of the Y vector.
 - Calculates $\text{norm} = \sqrt{Y(1)*Y(1)+Y(2)*Y(2)+...+Y(N)*Y(N)}$
7. Print the l2-norm and the time it takes the entire program to execute.

Objective:

Your main objective is to populate the table below:

Simulation #	DX_GLOBAL	L2-norm	Execution Time
1	50		
2	20		
3	10		
4	5		
5	1		

You will need to compile and run the code 5 times for different `DX_GLOBAL` values and record the results here. Review the existing code to make this program execute as fast as possible. You can choose to make small changes or completely destroy the entire code and data structures. You are free to use internet resources to search for ideas and/or use any profiling / performance tools that you wish.