CS 484: Project Milestone

Goal

The project milestone is used to make sure students are making satisfactory forward progress with their project. At the project milestone, students are expected to have the sequential implementations of their chosen algorithm working. The functional correctness of the implementation should be demonstrated for different input data (of a size that is at least larger than the last level cache) of varying sizes.

Description

In this project milestone report, a sequential artifact together with the results is presented. We will avoid physics and mathematics for now. The details of physics and mathematics will be given in the final write-up.

The sequential program consists of the following parts:

- main.cpp
  - In the project entry, we first read the geometry information in the text files generated by Matlab using the function “ReadSubdomainInfo()”.
  - Then we do the finite element assembly to form the linear system. Both the Dirichlet boundary condition and the source (electric charge) are enforced in this procedure.
  - Next, we solve this linear system using the serial Conjugated Gradient method. A non-DDM version is solved in the total domain first and then we use the Additive Schwarz DDM to solve this problem in each of the four subdomains.
  - Finally, the solution of the coefficients will be written to a text file, which will be visualized using Matlab script ‘draw_script.m’ in the post-processing procedure.
- Serial_CGM.cpp
  - This source file includes the sequential Conjugated Gradient solver.

In the sequential artifact, neither the DDM nor the Conjugated Gradient solver is parallelized. These two will be parallelized in the final submission. The former uses MPI and the latter uses OpenMP. Besides, in the assembly procedure, some operations can also be accelerated using OpenMP pragmas.

Sequential Artifact

Sequential implementation is in the ‘serial’ folder of Gitlab Project:
https://gitlab.engr.illinois.edu/sp21-cs484/turnin/jundaf2/fe-ddm-asm-parallel-cg.git

The structure of this self-determined project is very similar to the machine problems and pre-determined project, as can be seen from the following screenshot of the project folder.

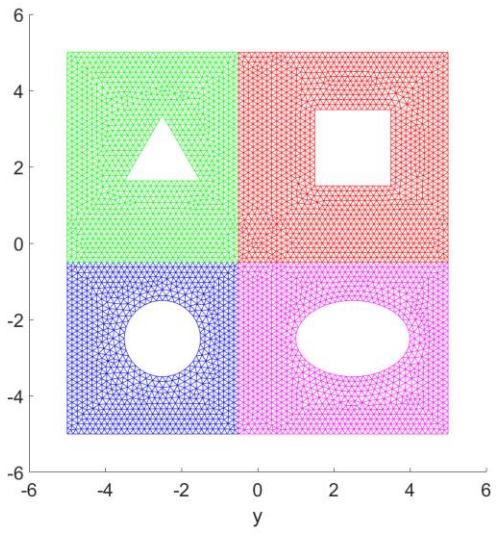
build	2021/4/18 22:17	File folder	
cmake	2021/4/17 13:38	File folder	
parallel	2021/4/17 13:38	File folder	
scripts	2021/4/18 21:43	File folder	
serial	2021/4/18 22:17	File folder	
writeup	2021/4/18 21:51	File folder	
.cproject	2021/4/16 22:01	CPROJECT File	5 KB
.project	2021/4/15 13:34	PROJECT File	1 KB
CMakeLists.txt	2021/4/18 22:13	Text Document	3 KB
README.md	2021/4/17 13:38	MD File	1 KB

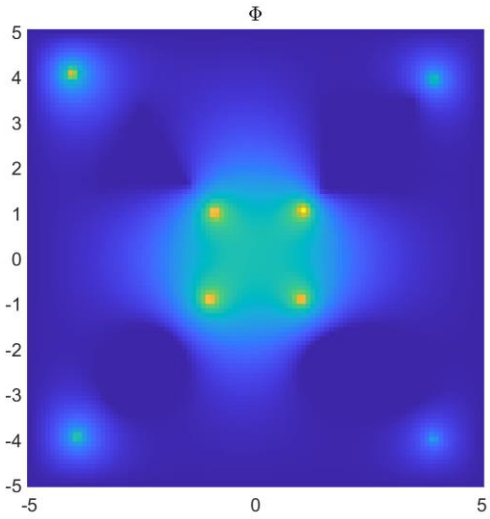
The ‘parallel’ folder and the ‘serial’ folder respectively contains the sequential version and the parallel version. Other folders are inherited from the tradition of the machine problems.

Functional correctness

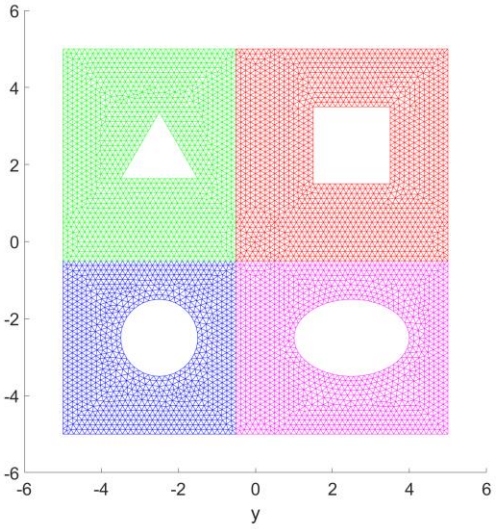
Functional correctness is demonstrated as follows.
There are 8 groups of figures in total. In each group, the first figure is the finite element mesh used in the test case. The denser the mesh, the more the unknowns, the larger the problem. The figures are enumerated in ascending order of the problem size. The second figure is the interpolated results using basis functions defined as nodes and the coefficients generated by solving linear systems. The interpolation procedure is done by the Matlab script ‘draw_script.m’.

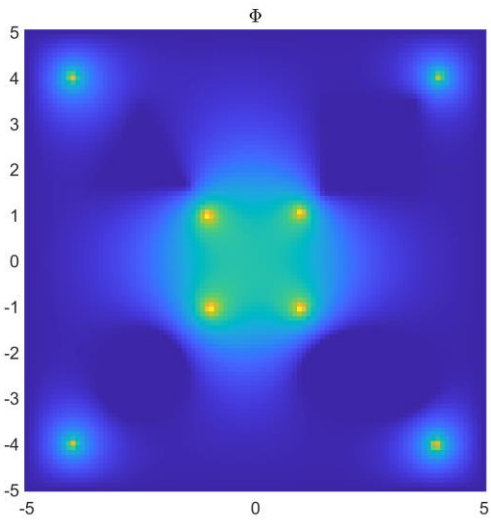
- 4580 unknowns



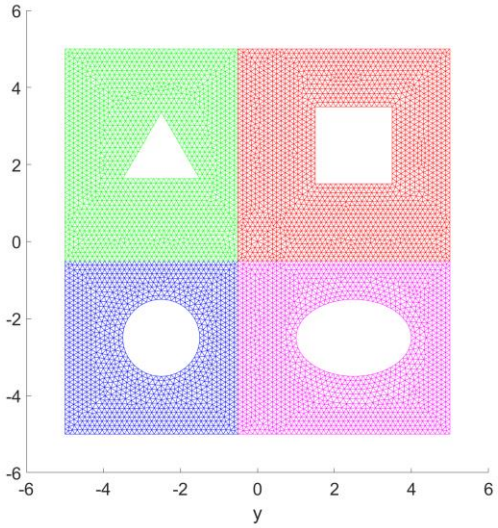


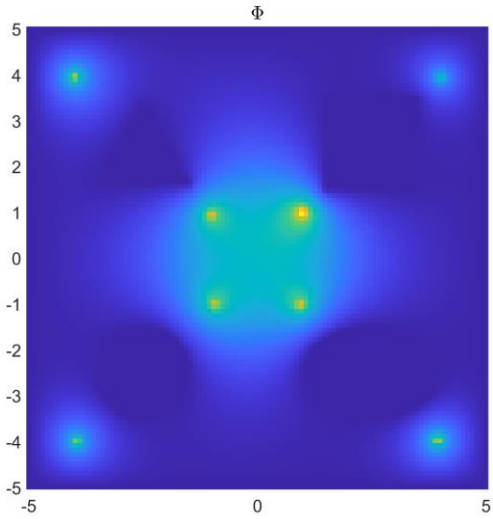
- 5116 unknowns



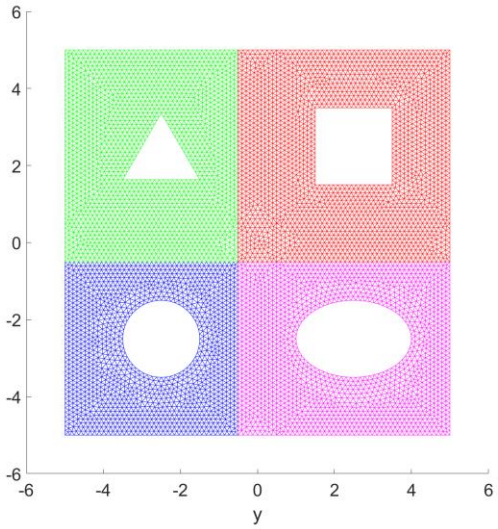


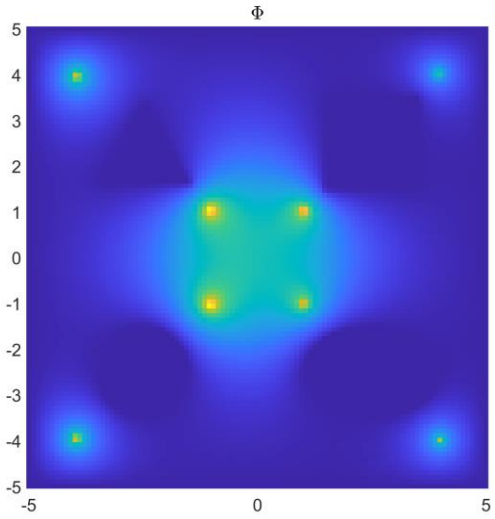
- 6094 unknowns



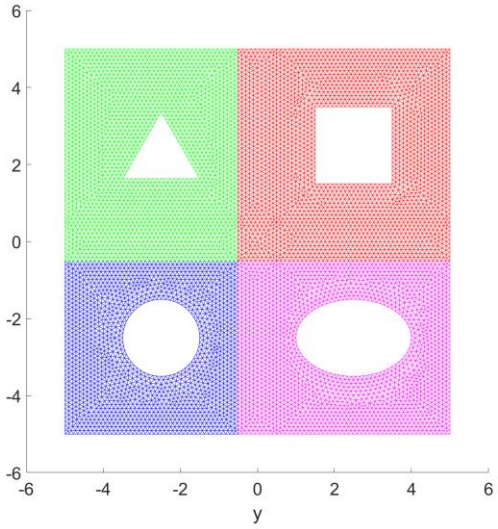


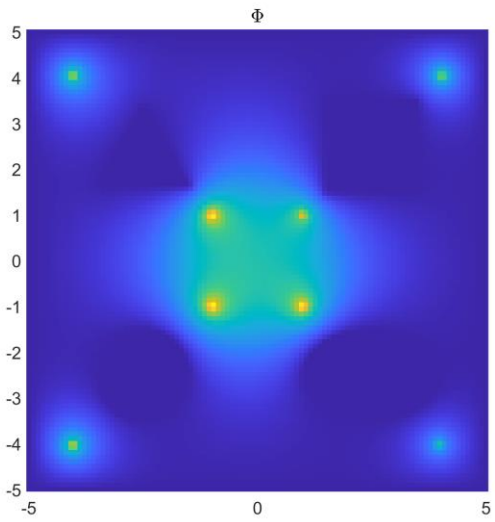
- 7191 unknowns



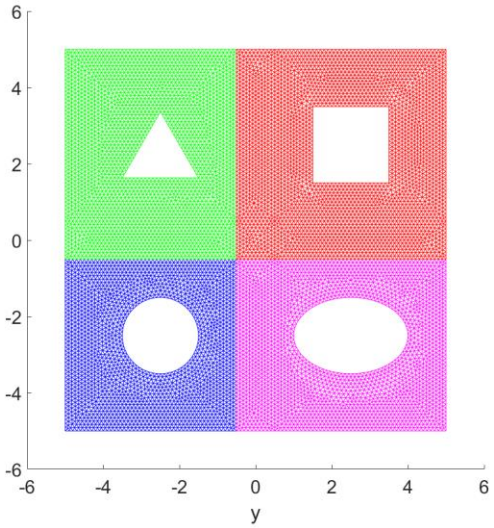


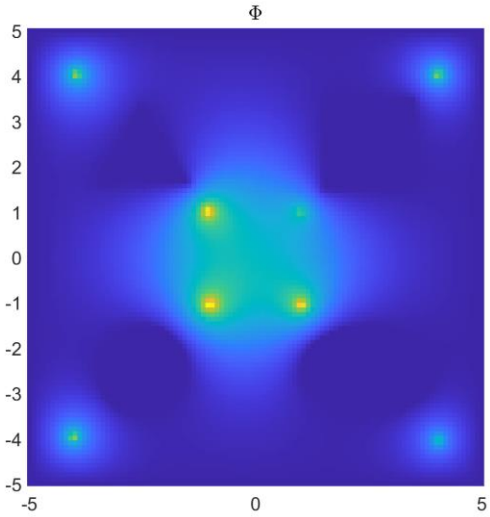
- 8294 unknowns



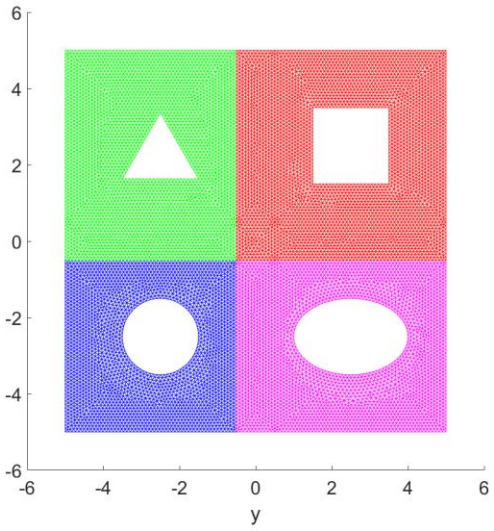


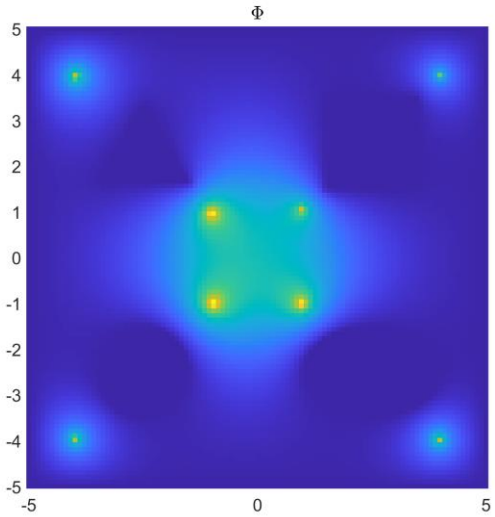
- 10179 unknowns



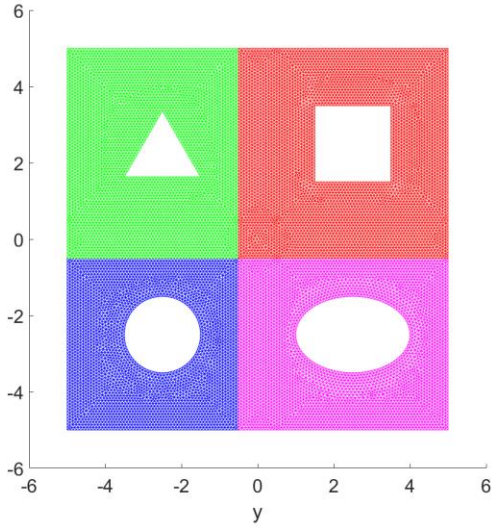


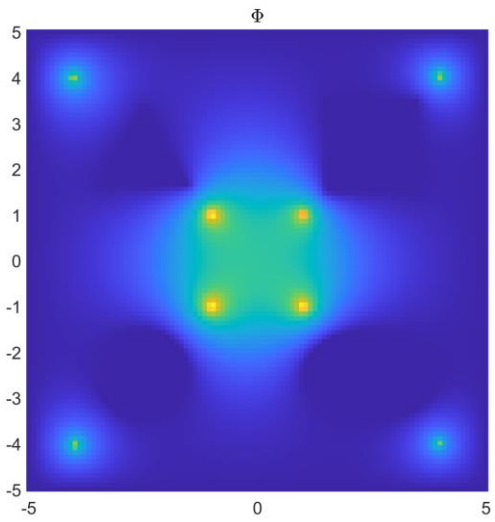
- 12370 unknowns



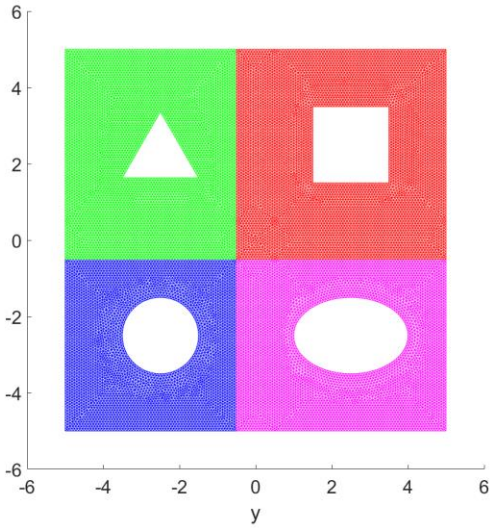


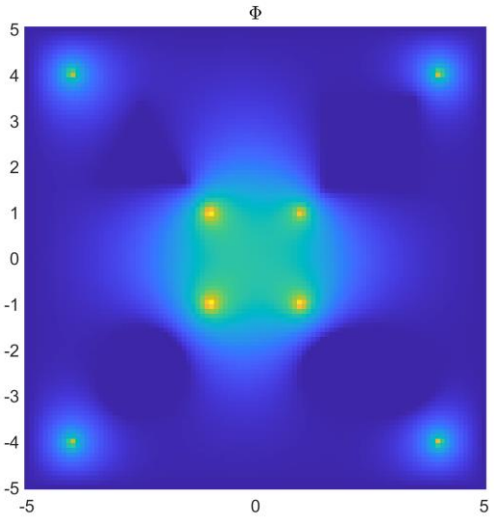
- 15587 unknowns



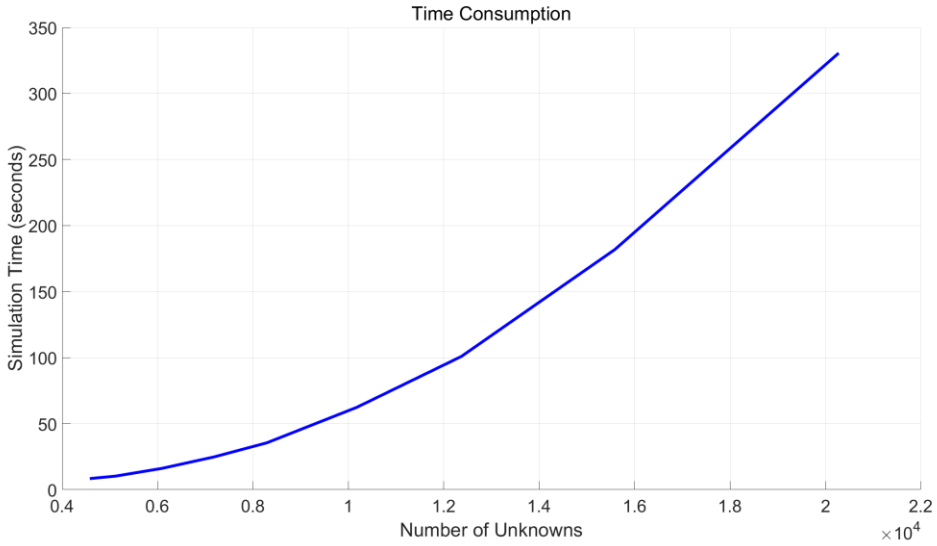


- 20277 unknowns


6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6
y


Φ
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
0
5

Time consumption vs different input size



As we can see, the time consumption of this electrostatic 2D finite element method increases almost exponentially as the number of unknowns grows.

Appendix: the command line output of the executable program

./bin/feddm_serial_cg 0.15
... read sequence has no problem ...
Num_Nodes 4580
Num_Elements 8714
coords size 9160
Element2NodeList size 26142
... traditional serial fem costs time 2.67642 s ...
... SubArray success ...
... SubMatrix success ...
... serial variables define success ...
... ASM serial version costs time 8.36184 s ...

./bin/feddm_serial_cg 0.14
... read sequence has no problem ...

Num_Nodes 5116
Num_Elements 9754
coords size 10232
Element2NodeList size 29262
  ... traditional serial fem costs time 3.37738 s ...
  ... SubArray success ...
  ... SubMatrix success ...
  ... serial variables define success ...
  ... ASM serial version costs time 10.2035 s ...

./bin/feddm_serial_cg 0.13
... read sequence has no problem ...
Num_Nodes 6094
Num_Elements 11668
coords size 12188
Element2NodeList size 35004
  ... traditional serial fem costs time 5.24424 s ...
  ... SubArray success ...
  ... SubMatrix success ...
  ... serial variables define success ...
  ... ASM serial version costs time 16.1075 s ...

./bin/feddm_serial_cg 0.12
... read sequence has no problem ...
Num_Nodes 7191
Num_Elements 13815
coords size 14382
Element2NodeList size 41445
  ... traditional serial fem costs time 7.98685 s ...
  ... SubArray success ...
  ... SubMatrix success ...
  ... serial variables define success ...
  ... ASM serial version costs time 24.8173 s ...

./bin/feddm_serial_cg 0.11
... read sequence has no problem ...
Num_Nodes 8294
Num_Elements 15975
coords size 16588
Element2NodeList size 47925
  ... traditional serial fem costs time 11.3816 s ...
  ... SubArray success ...
  ... SubMatrix success ...
  ... serial variables define success ...

... ASM serial version costs time 35.4795 s ...

./bin/feddm_serial_cg 0.1
... read sequence has no problem ...
Num_Nodes 10179
Num_Elements 19681
coords size 20358
Element2NodeList size 59043
... traditional serial fem costs time 19.5402 s ...
... SubArray success ...
... SubMatrix success ...
... serial variables define success ...
... ASM serial version costs time 62.3584 s ...

./bin/feddm_serial_cg 0.09
... read sequence has no problem ...
Num_Nodes 12370
Num_Elements 23993
coords size 24740
Element2NodeList size 71979
... traditional serial fem costs time 32.2924 s ...
... SubArray success ...
... SubMatrix success ...
... serial variables define success ...
... ASM serial version costs time 100.883 s ...

./bin/feddm_serial_cg 0.08
... read sequence has no problem ...
Num_Nodes 15587
Num_Elements 30326
coords size 31174
Element2NodeList size 90978
... traditional serial fem costs time 58.1756 s ...
... SubArray success ...
... SubMatrix success ...
... serial variables define success ...
... ASM serial version costs time 181.851 s ...

./bin/feddm_serial_cg 0.07
... read sequence has no problem ...
Num_Nodes 20277
Num_Elements 39589
coords size 40554
Element2NodeList size 118767

... traditional serial fem costs time 106.259 s ...
... SubArray success ...
... SubMatrix success ...
... serial variables define success ...
... ASM serial version costs time 330.56 s ...