Parallel computing in finite element analysis of geomagnetic vertical profiling

O.V. Shapovalov¹, V.V. Getmanskiy¹, D.I. Kryzhanovskiy¹, V.S. Chalyshev¹, S.E. Larin², P.S. Pavlov²

¹Singularis Lab Ltd. (http://www.singularis-lab.com/en.html), Russia, Volgograd
²GEOFISTECHNIKA Ltd. (http://geoft.ru), Russia, Saratov

oleg.shapovalov@singularis-lab.com, victor.getmanskiy@singularis-lab.com,
dmitry.kryzhanovsky@singularis-lab.com, vladimir.chalyshev@singularis-lab.com, geoft.office@gmail.com,
geoft.office@gmail.com

Abstract. The present paper is devoted to application of the finite element analysis for solving electrostatics problems. The job appears when one reconstructs the rock structure by geomagnetic probing. The 3D axisymmetric problems are computed using the first and second order finite elements. The free library Triangle is used for triangulation; the libraries MKL, CUSP and the technologies OpenMP, MPI are used for parallel computing. The proposed implementation of the method operates faster than COMSOL’s one with the same mesh. The maximum solution error is not higher than 1 %.

Keywords
Finite element analysis, geomagnetic profiling, antisymmetric electrostatic problem.

1 Introduction

The method of electric coring is one of the modern geophysics research trends. It consists in downhole geophysical survey and further research of electromagnetic field in ground adjoining to the hole wall. Coring measurements elaboration is impractical without high-performance hardware and software implementation of numerical models for simulation of electromagnetic field in media. There are several commercial and freeware packages with similar capabilities (ANSYS, COMSOL, Elmer), but in many cases the problem specific requirements make it hard to achieve a quality solution in acceptable time using common software packages. It is determined by increasing solution quality and performance requirements caused by fast development of high-performance hardware capabilities in the recent years.

It is evident that for achievement of the quality software solution it is required to utilize all computing resources with distribution computations to CPUs and GPUs in heterogeneous computing nodes.

So the tasks of the present research are:

1. Adaptation of the existing algorithms and solutions for a parallel implementation working in heterogeneous computing hardware with CPUs and GPUs.
2. Development of the software library for 3D axial symmetrical and asymmetrical electrostatic field simulation problems providing with support for COMSOL models format.

2 Theoretical Part

2.1 Solving direct electrostatic problem with finite element method

Exploring horizontally imbedded rocks, one encounters the problem how to determine location and thickness of the rock strata. The mathematical apparatus applied here comes to multi-objective optimization task with the given initial approximation. The exact estimation is performed using a 3D model of the explored space. But for the preliminary estimation it’s enough to use an axisymmetric model of the explored space and multivariant calculation with varying the strata thickness.
In this work the forward electrostatics task is solved for finding out the distribution of potentials inside the geophysical cut. The appropriate mathematical model based on inhomogeneous Poisson’s equation, given below in cylindrical coordinates:

\[
\frac{1}{r} \frac{\partial}{\partial r} \left( r \sigma (r, \varphi, z) \frac{\partial u}{\partial r} \right) + \frac{1}{r^{2}} \frac{\partial}{\partial \varphi} \left( \sigma (r, \varphi, z) \frac{\partial u}{\partial \varphi} \right) + \frac{\partial}{\partial z} \left( \sigma (r, \varphi, z) \frac{\partial u}{\partial z} \right) = f(r, \varphi, z),
\]  
(1)

where \( \sigma \) is conductivity, \( u \) is the potential distribution function, \( f(r, \varphi, z) \) is electric charge distribution function.

On the assumption of axisymmetry the derivative in \( \varphi \) is equal to 0 and the equation (1) takes the form

\[
\frac{1}{r} \frac{\partial}{\partial r} \left( r \sigma (r, z) \frac{\partial u}{\partial r} \right) + \frac{\partial}{\partial z} \left( \sigma (r, z) \frac{\partial u}{\partial z} \right) = f(r, z),
\]  
(2)

For discretization of the equation (2) the finite element method (FEM) [2] with a triangular adaptive mesh was selected since it showed itself to good advantage for such tasks. The linear and quadratic triangular FEs (finite elements) are used in the proposed implementation.

This implementation is composed of several stages: specifying the calculation scheme, triangulation, calculation with finite element method, postprocessing.

2.2 Specifying the calculation scheme

At this stage one adds geometric objects, sets the materials, boundary conditions and sampling lines. Arbitrary closed and non-closed polygons can be used as the geometric objects. The materials are set for closed areas. The boundary conditions can be set for lines and closed regions. The samples can be performed only along the lines at the moment.

2.3 Triangulation

The free library Triangle [1] is applied for triangulation. It works with Delaunay triangulation and splits the space as a planar graph with direct edges.

Though the library supports both the linear and the quadratic finite element mesh generation, the quadratic case takes much time for the generation, about 40 – 50 % of the total computation time. That is why the mesh is always generated since it showed itself to good advantage for such tasks. The linear and quadratic triangular FEs (finite elements) are used in the proposed implementation.

2.4 Calculation with finite element method

Calculation with finite element method consists of the following stages: building the stiffness matrix for all the FEs, assembling the global stiffness matrix, solving system of linear equations (SLE).

On the assumption of axisymmetry the stiffness matrix for a triangular element takes the form

\[
\left[ k^{(e)} \right] = \frac{\pi R \sigma}{2 A} M
\]  
(3)

where \( \sigma \) is permittivity, \( A \) is the triangular element area, \( R \) is the distance between the triangle mass center and the symmetry axis.

The matrix \( M \) for the first order elements has the form

\[
M = \begin{bmatrix}
    b_{1} b_{1} + c_{1} c_{1} & b_{1} b_{j} + c_{1} c_{j} & b_{1} b_{k} + c_{1} c_{k} \\
    b_{j} b_{1} + c_{j} c_{1} & b_{j} b_{j} + c_{j} c_{j} & b_{j} b_{k} + c_{j} c_{k} \\
    b_{k} b_{1} + c_{k} c_{1} & b_{k} b_{j} + c_{k} c_{j} & b_{k} b_{k} + c_{k} c_{k}
\end{bmatrix}
\]  
(4)

where \( b \) and \( c \) are the coefficients depended on the triangle vertices coordinates as follows:

\[
b_{1} = Y_{2} - Y_{3}; \quad c_{1} = X_{3} - X_{2};
b_{2} = Y_{1} - Y_{3}; \quad c_{2} = X_{3} - X_{1};
b_{3} = Y_{1} - Y_{2}; \quad c_{3} = X_{2} - X_{1};
\]  
(5)

where \((X_{a}, Y_{a})\) are the coordinates of the \( a \)-th triangle vertex.
The matrix $M$ for the second order elements has the form

$$M = \begin{bmatrix}
  k_{1}^2 + c_1^2 & -k_{1} b_{2} c_{1} c_{2} & 0 & 4(b_{1} b_{2} + c_{1} c_{2}) \\
  -k_{1} b_{2} c_{1} c_{2} & k_{2}^2 + c_2^2 & 0 & 4(b_{1} b_{2} + c_{1} c_{2}) \\
  0 & -k_{2} b_{1} c_{1} c_{3} & 0 & 4(b_{1} b_{2} + c_{1} c_{3}) \\
  -k_{2} b_{1} c_{1} c_{3} & 0 & k_{3}^2 + c_3^2 & 4(b_{1} b_{2} + c_{1} c_{3}) \\
  4(b_{1} b_{2} + c_{1} c_{2}) & 4(b_{1} b_{2} + c_{1} c_{3}) & 4(b_{1} b_{2} + c_{1} c_{3}) & 4(b_{1} b_{2} + c_{1} c_{2}) \\
  4(b_{1} b_{2} + c_{1} c_{2}) & 4(b_{1} b_{2} + c_{1} c_{3}) & 4(b_{1} b_{2} + c_{1} c_{3}) & 4(b_{1} b_{2} + c_{1} c_{2}) \\
  4(b_{1} b_{2} + c_{1} c_{2}) & 4(b_{1} b_{2} + c_{1} c_{3}) & 4(b_{1} b_{2} + c_{1} c_{3}) & 4(b_{1} b_{2} + c_{1} c_{2}) \\
  4(b_{1} b_{2} + c_{1} c_{2}) & 4(b_{1} b_{2} + c_{1} c_{3}) & 4(b_{1} b_{2} + c_{1} c_{3}) & 4(b_{1} b_{2} + c_{1} c_{2})
\end{bmatrix}$$

(6)

where $f(p,q) = p^2 + pq + q^2$, $g_p = p_1 + p_2 + p_3$.

The global stiffness matrix represents the sum of single FEs contributions. To assembly the global matrix one need just to sum the single FEs stiffness matrices for the corresponding vertices.

2.5 Postprocessing

Postprocessing consists in the final treatment of the results gained at the calculation stage. At the moment the following features are implemented: comparison with the previously saved calculations in COMSOL, computing the current through the selected surface, saving the results in VTK format for further viewing in Paraview.

3 Practical Part and Experiment

3.1 Description of the software library

The software being developing is a dynamic library which allows to perform finite element simulation of electrostatic field in axisymmetrical geologic horizon models based on the data received from well-logging device. Library consists of the objects registry based on abstract factory. It provides with necessary interfaces for manipulation of different objects required for simulation such as geometrical primitives, materials, boundary conditions and measurements for direct problem and also variable voltage and restrictions in addition for dynamic focusing problem.

The calculation schemes including the well-logging device model is specified by polygons. The materials, boundary conditions, measurement surfaces etc. are assigned to the polygons.

To convert the model from overlapping polygons to a planar graph we coded the algorithms for splitting the overlapping and intersecting polygon edges. The uniqueness of the points is provided with the rule of sorting by coordinates. To sort the points correctly we use an additional collection of the points given with the fixed precision corresponding the float numbers comparison error. The sorted collection is used for a quick search when adding new points to avoid ones having the same coordinates.

After triangulation the materials, boundary conditions and measurements are assigned to the triangles and points, that is the triangles and the points are matched to arbitrary geometrical regions.

The initial implementation of preprocessing took about 90% of the time because of the slow collections in the standard C++ library and the linear search algorithms. To speed up the geometric algorithms that detect belonging of the points to various regions we use a preliminary detection of belonging of these items to the rectangles bounding boxes. The linear search algorithms were replaced with the binary ones, and the collections of the type map and the triangles to various regions we use a preliminary detection of belonging of these items to the rectangles bounding boxes. The collections are optimized and take less than a tenth of percent of the total simulation time.

To provide the compatibility with COMSOL format a parser was developed; it works with m-file representation of COMSOL model and extracts the information for creating the model. Also COMSOL-like logic was developed for properly numbering of model objects. In particular supernodes based contour searching algorithms were used for splitting the domain into polygons. It is also implemented COMSOL-like sorting of objects. All parsing algorithms are optimized and take less than a tenth of percent of the total simulation time.
3.2 Experiment

Testing of the performance was done on the stand with Intel Core i5 2310 and GeForce GTX560. The general speed-up and the speed-up of solving SLE was measured and compared with the serial calculation.

The CPU speed-up with use of the developed library compared with COMSOL, also using parallel libraries to solve SLE, makes 1.5 – 2 times. To verify the solution we used the data gained with COMSOL v. 4.2. To control the precision we specified the surface which the total current was computed through. The results are displayed in the table 1.

<table>
<thead>
<tr>
<th>Environment conductivity</th>
<th>COMSOL</th>
<th>Computation with the 1st order FEs</th>
<th>Computation with the 2nd order FEs</th>
<th>$\varepsilon_1$</th>
<th>$\varepsilon_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>0.7441</td>
<td>0.741</td>
<td>0.734</td>
<td>0.0031</td>
<td>0.0101</td>
</tr>
<tr>
<td>3.4</td>
<td>1.6361</td>
<td>1.648</td>
<td>1.623</td>
<td>0.0119</td>
<td>0.0131</td>
</tr>
<tr>
<td>5.6</td>
<td>2.2708</td>
<td>2.278</td>
<td>2.238</td>
<td>0.0272</td>
<td>0.0128</td>
</tr>
<tr>
<td>7.8</td>
<td>2.7000</td>
<td>2.742</td>
<td>2.688</td>
<td>0.0420</td>
<td>0.0120</td>
</tr>
<tr>
<td>10</td>
<td>3.0426</td>
<td>3.098</td>
<td>3.031</td>
<td>0.0554</td>
<td>0.0116</td>
</tr>
</tbody>
</table>

In the presented model dynamic focusing method is used to calculate the unknown potential of the middle electrode using the known potentials of the upper and the lower electrodes and potential equality constraint for middle electrodes separated by dielectrics.

As one can see, the parallel implementation speed-up of single simulation is far from an ideal speed-up. The main running mode of the developed software is a parametric multivariant computation. So the solution can be effectively used both for computing cluster system and for multicore hardware with OpenMP and MPI libraries. The parallel simulation is performed with even distribution of parametric variants on cluster nodes. The results are presented in the table 2.

<table>
<thead>
<tr>
<th>Triangles count</th>
<th>COMSOL</th>
<th>Serial</th>
<th>OpenMP</th>
<th>MPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 000</td>
<td>219</td>
<td>169.4</td>
<td>59.3</td>
<td>57.7</td>
</tr>
<tr>
<td>50 000</td>
<td>527</td>
<td>305.7</td>
<td>106.1</td>
<td>100</td>
</tr>
<tr>
<td>100 000</td>
<td>1377</td>
<td>712</td>
<td>232</td>
<td>205.3</td>
</tr>
</tbody>
</table>

4 Conclusion

The developed software library is designed and suitable for numerical solution of geomagnetic probing. It allows to solve direct electrostatic field simulation problem, dynamic focusing problem and multivariant computation. Model can be specified directly using library programming interface or imported from COMSOL. Current implementation provides with high precision and speed of simulation. Parallel computing is realized with optimization for multi-threading MKL PARDISO solver and GPGPU CUSP library solver. The parallel multivariant solution is implemented using OpenMP and MPI.

References
