AutoCAD application for LPS, grounding and EMC problems

D.S.Shishigin [1], S.L.Shishigin[1], N.V.Korovkin[2]

[1] Vologda State University, Vologda, Russia, shishigind@yandex.ru

[2] Saint-Petersburg State Polytechnical University, Saint-Petersburg, Russia

Abstract—A software program ZYM (AutoCAD application) for calculating of LPS, grounding and EMC problems of power substations has been developed. The program consists of a calculation kernel and an AutoCAD-bound interface (COM-technology). Advanced methods and programming technologies are used. High performance is reached with the use of Intel Math Kernel Library. The results of calculation get visualized by 3D plots, and dynamic processes are animated. The example of LPS, grounding and EMC calculations of electric power substation is given.

Keywords—LPS, grounding, shielding, AutoCAD, ZYM, computation electromagnetics, electric power substations.
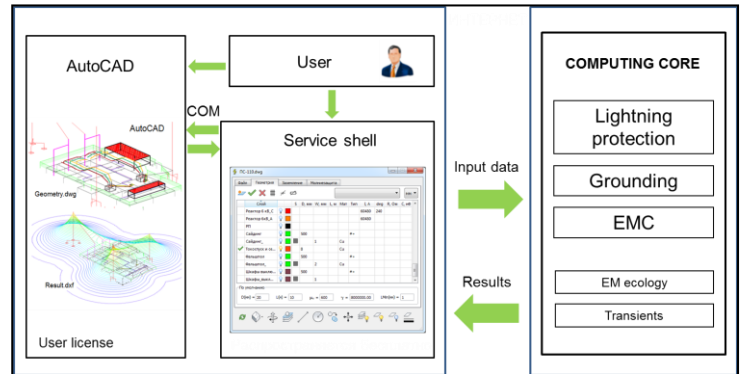
I. INTRODUCTION

Grounding, shielding and lightning protection systems ensure a reliable operation of electric equipment and microprocessors based control systems of electrical power substations and provide the necessary electrical safety requirements. While designing a power substation computer simulation of emergencies due to lightning strokes, short circuits and electrical equipment switching is performed. The designer must find technical solutions ensuring reliable operation of the substation. Hence, there are strict requirements to the software.

While developing the computer program we aimed to implement new methods [1-3], to increase calculation performance and to provide the designer with familiar environment for geometrical modeling in AutoCAD.

II. PROGRAM STRUCTURE

ZYM program includes (fig.1): a computing core and a service shell (interface). The core is presented in the form of a dll-library set. Intel Math Kernel Library (Intel MKL) package procedures are used for matrix operations acceleration. AutoCAD control is carried out using the object model (COM technology). Plots are preliminary drawn in dxf-format and are automatically loaded in AutoCAD. Basic commands are duplicated in the program for user's convenience.

Fig. 1 ZYM program structure

III. COMPUTING CORE MODELS AND METHODS

A. Grounding

Calculation of grounding devices lies in defining its resistance, currents and voltages of its elements (as in the circuit theory), and also potential distribution and electromagnetic field strength (as in the EM field theory). These EM-circuit problems and their solution require two interrelated models – EM and circuit models (a hybrid model). An EM-model is required to calculate electromagnetic characteristics of the grounding elements which will be used afterwards in a circuit model to calculate element currents. Once currents are specified, the distribution of electromagnetic field strength shall be calculated again on the EM-model. Such calculation methodology is typical for the antenna theory as well.

The grounding device is divided into elements of the small length $l<\lambda/10$ (λ – EM wave length), in order to use the static approximation for parameters calculation. Elements are described by the square matrices of leakage conductivity **G**, inductance **M**, capacity **C** and also by the vectors of own resistance **r** and own inductance **L**. Two-layer, multy-layer 1D models and 3D model, based on the integral equation method, are used for grounding resistance calculation.

The nodal solution is applied for steady-state conditions. The discrete coordinate method should be used to calculate transients. All calculation details are described in [1-2].

B. EM screens modeling

The grounding device model has commonality, which allows solving electromagnetic screening problems. The design of a grid screen is similar to that one of ground grids; that is why the calculations of grid screens should be performed together with the grounding by using common programs. It is desirable to use a similar procedure for bulk screens by replacing them with grid models. Unfortunately there is no Hertzian wave attenuation process in a grid model.

A real steel screen may be replaced by a curtain model (without wave attenuation at frequencies above some kHz) with infinite conductivity, whose internal resistance is zero, without capacitance currents. The screen shall be replaced by a bar grid. At selecting the bar diameter, the condition of invariable external inductance of a plate should be met. Thus, bulk screens may also be simulated by grids and the calculations of any screens can be performed with the grounding. The calculation of canonical screens proves it (fig.2).
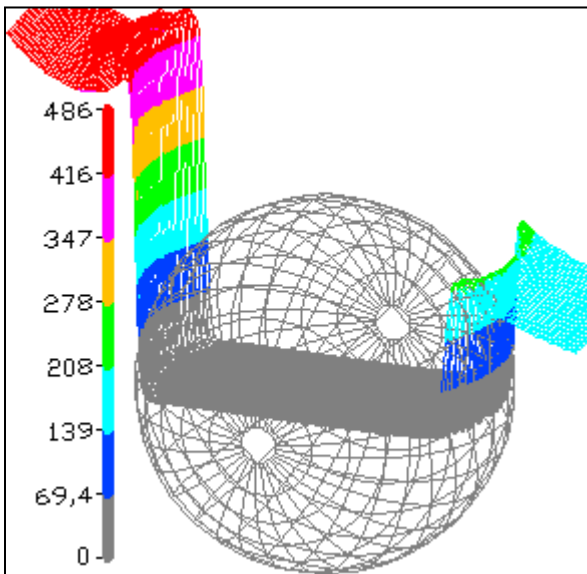

Fig. 2   A wire magnetic field with a spherical screen

C. LPS designing

The rolling sphere method (RSM) and protection angle method (PAM) are the main engineering approaches for LPS design according to IEC 62305-3. Russian standards use PAM, but it specifies different angles for single and double lightning rods, and protection zone can be located below the top of the rod. Calculation practice has shown that current methods are not agreed and lead to different results. We consider that the main RSM drawback is the way capture zone (the geometrical position of rolling sphere centers) is constructing. The lightning leader appearance at some points of the zone is highly unlikely, and the protection from non-existent threats results in unseasonable solutions. This drawback is excluded in our version of RSM [3]. The capture zone is constructing with the condition of equal probability of lightning leader appearance, the rolling sphere radius is variable.

All methods of LPS design are implemented in ZYM program, it is up to a designer to choose any of them.
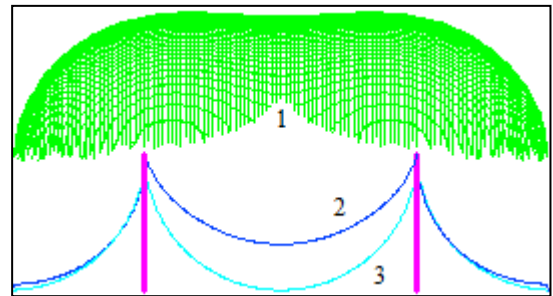

Fig.3   Capture zone (1), protection zone (2); RSM protection zone (3)

The comparison of this method with RSM has shown that the protection zones of a single rod are nearly the same (fig. 3, outside the lightning conductors). Double rods protection zones are considerably wider when using the new method, which is explained by the lightning rods mutual influence.

IV.   ZYM INTEGRATION INTO AUTOCAD

A. Why AutoCAD

Computer programs, that manage complex 3D geometric models, should be developed as AutoCAD (or other CAD-system) applications. There is no need to develop your own graphic editor when AutoCAD has an open access to all methods, properties and events with geometrical primitives. We can read 3D geometric models, plot 3D graphs or make drawings in AutoCAD based on the data from our own program.

B. The problem and it's solution

The problem is that AutoCAD is not destined for scientific calculations and does not possess ready-made components for visualization of calculation results (as Mathcad does). But one can make a 3D-graph component, because a graph is a set of lines, which can be drawn with the use of the *AddLine*-function. However, this obvious route turned to be ineffective for plotting high dimensional graphs due to its low performance.

As the basis for the task solution we used the following idea: data arrays for plotting and animating of 3D-graphs in AutoCAD should be stored in dxf–format (data exchange format). Actually the graph is preliminary drawn in dxf-codes, and for digitization of the graph the numerical values are stored in extended data (XDATA) bound to each line. Dimension of a dxf-file may exceed million strings, but it can be quickly written and loaded in AutoCAD. In a dynamic mode we make a sequence of time frames each of which is stored in a separate layer, one layer is made visible. Switching between the layers creates the effect of animation similar to viewing avi-files. Frame navigation system

in the form of player allows frame viewing of dynamic processes in manual mode and animated viewing when automatic mode of switching frames is applied. Thus, plotting of 3D-graphs in dxf-codes is considerably faster for data arrays than the standard method with the use of the *AddLine*-function.

C. Geometrical modeling support

AutoCAD contains all necessary tools for constructing complex geometrical objects. Nevertheless, scripts that automatize typical geometrical constructions are created (fig.4).
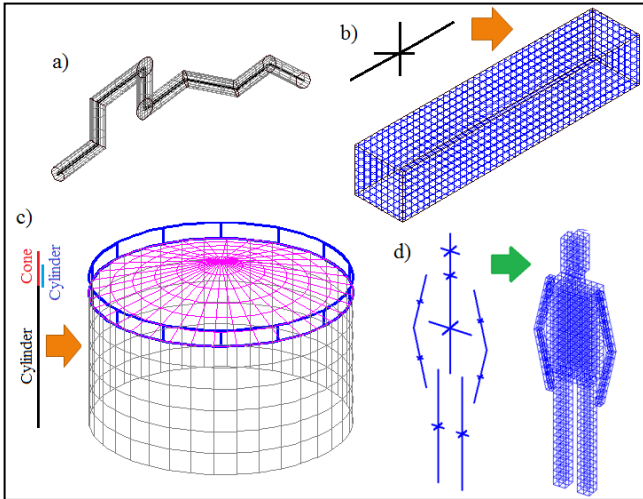

Fig. 4   Grid models of: a control cable (a), an EM-screen (b), a tank (c), a person (d)

V. CODE PERFORMANCE

A. The problem

Calculation performance is one of the main features of a computer program, necessary condition of its competitiveness. High performance is achieved by a careful choice of mathematical models and methods, therefore, it appears as an indirect feature of built-in algorithms effectiveness. It is necessary to aim to use vector equations notation, which is not only more compact, but also more efficient.

B. Algorithmic code optimization

The first step on the way to calculation acceleration is the algorithmic code optimization. The basic resource of performance growth is a decrease of calls to a slow RAM memory and an increase of calls to a fast cache memory. Cache stores last used cycles and data, so that cycles and operations with arrays will be executed faster. When an implementing device needs data, and it is not in a cache memory, this is called a cache miss: a CPU must address to an external memory for data fetching. If requested data is in a cache, this is a cache hit: access to an external memory is not needed. For example, when multiplying matrices there is a multiplication of one matrix row by another matrix column

(fig.5a). Matrix row consecutive elements are located one after another, and access to it will be the fastest possible, because prefetch strategy (reading RAM data block, that equals cache line) of elements in a cache fully justifies itself. On the other hand, matrix column consecutive elements are at a distance of $n * sizeof (n)$ bytes from one another and access to it will be the slowest possible (because a cache memory line is much less than the distance between elements). In the mathematical library *Alglib* the matrix multiplication algorithm is written in a non-standard way: matrix elements multiplication is executed by rows (fig.5b). Therefore, the number of cache misses is come to minimum, which leads to the matrix multiplication operations acceleration in several times (in 7 times for square matrices with dimensionality 2000) (table 1). Here we also see that a compiler choice practically does not affect calculation performance. So, data structure and algorithms should be considered carefully in order to reduce cache misses count.
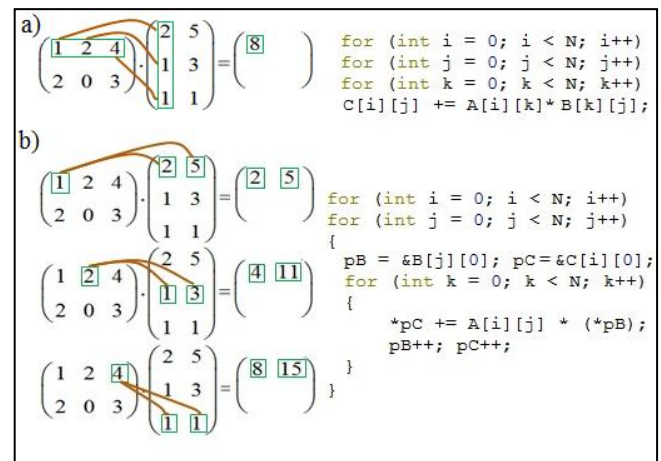

Fig. 5   Matrix multiplication: rows by columns (a), rows by rows (b)

TABLE I.   COMPARISON OF STANDARD AND OPTIMAL MATRIX MULTIPLICATION ALGORITHMS

Compiler	Standard algorithm	"By row" multiplication
MS Visual C++	94.3 s	15.8 s
Intel C++	117.9 s	13.2 s
Delphi XE3	96.1 s	15.6 s
Matrix dimensionality – 2000. Notebook: Windows 7 64-bit, Intel Core i7 4x2.2 GHz, RAM 6 Gb.		

C. Matrix operations acceleration

An independent research was carried out on matrix operations acceleration that is the most time-consuming part of an algorithm. Performance capabilities of the freeware package *Alglib* (was mentioned earlier) with algorithmic code optimization and the commercial library Intel Math Kernel Library (Intel MKL) with low-level optimization and multithreading was compared. Note that Mathcad and Matlab users owe Intel MKL for matrix operations high performance. Results are shown in table 2.

Matrix operation	AlgLib	Intel MKL	Comparison
A*A	11.8 s	0.6 s	19 times
linear system A*X=B	3.9 s	0.3 s	13 times
A^{-1}	28.8 s	1.1 s	26 times
linear system C*Y=D	46 s	0.9 s	51 times
C^{-1}	125 s	3.6 s	34 times
N=2000, A[N,N] double, C[N,N] complex, X,B[N] double, Y,D[N] complex. Notebook: see Table 1.			

So, Intel MKL significantly increases matrix operations performance compared to Alglib package and its analogs. When developing commercial software with matrix operations it is certainly worth the money spent.

Today there is a tendency to transition from calculations on central processors (CPU) to calculations on graphics processors (GPU). Brief tests with NVIDIA CUDA allowed us to accelerate matrix multiplications in 2 times compared to Intel MKL. In general, we managed to increase matrix operations performance in 100-300 times compared to "student" algorithms. So, there are great resources of calculation performance increase that should be implemented in professional programs.

VI.   CODE SIMPLICITY

Following the example of mathematical packages the core is presented as a set of procedures in dll-libraries with a set of test problems. The core is developed by specialists, many of which are far from programming. We ensured that if we debug the solution in Mathcad on test problems using matrix operations then we get a simple, understandable code that is the key to the long life cycle of the program.

VII.   PROGRAM VERIFICATION

The testing starts with the problems, which have analytical solution. For instance, the impedance of a copper rod with a length of l=20 m, radius R=1 cm in the ground with ρ=100 Ω·m at frequency f=25 kHz is Z=5.89+j1.52 Ω if long line theory is used and Z=5.91+j1.44 Ω using ZYM program. Error is 1%.

Grounding calculation problems from scientific papers of other authors have been solved. Satisfactory match in results was achieved in all cases. We would also like to offer a test problem (fig.6).

VIII.   SIMULATION RESULTS

Let's design LPS and grounding device of the electric power substation (fig.7). First, we analyze short-circuit modes (fig.8). If maximum potential, current or touch voltage values exceed the standard values, changes are made in the grounding device design. Then we analyze lightning protection reliability. It is determined that lightning protection rods intercept lightning strokes (fig.9, 10). Then we simulate a lightning stroke to each lightning rod and find potential and current distribution. Animation allows examining the transients in dynamics. Voltage maximum on
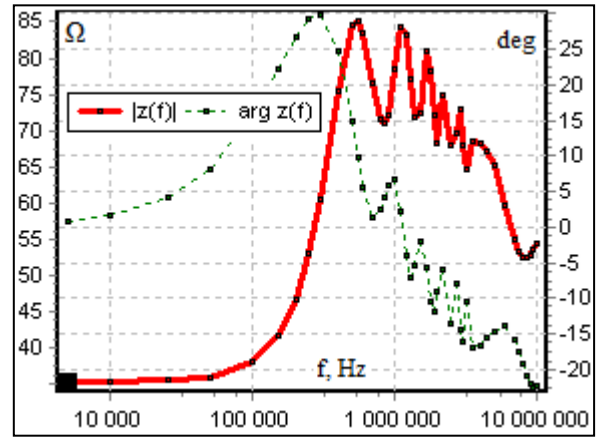

Fig. 6   Impedance frequency response of a copper horizontal rod with a length of 50 m, diameter of 20 mm, burial depth of 1 m in soil with ρ=1000 Ω·m, ε=30

the control cable screen should not exceed permissible value. Magnetic field strength in microprocessor equipment locations should not exceed permissible level (taking into account shielding effect of screens and metal constructions) (fig.11). Otherwise we adjust rods location, intensify shielding effect and use equipment with higher noise immunity.

IX.   COMPARISON WITH SES-SOFTWARE

SES-company is a "world leader in grounding, lightning and electromagnetic interference" (sestech.com). If so then with the use of advanced mathematical models and methods, AutoCAD and Intel MKL we can compete. Similar to SES-software our program is aimed at a complex solution of grounding, lightning and EMC problems. Unlike SES we pay much attention to external LPS.

X.   CONCLUSIONS

Computer programs which operate with complex geometric models should be developed as AutoCAD (or other CAD-system) applications. High performance of matrix operations is reached with the use of Intel MKL. The code simplicity is achieved by researching and developing of modules in Mathcad. The ZYM program, that includes modern mathematical models and methods, user-friendly interface, is a useful tool for a LPS and grounding design.
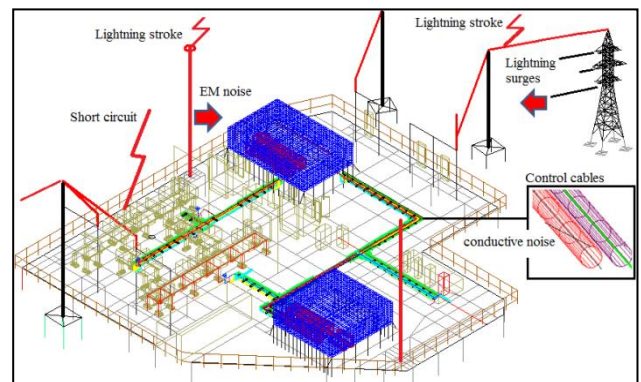

Fig. 7   Electric power substation model

REFERENCES

[1] N.V. Korovkin, O.V. Frolov, S.L. Shishigin, D.S. Shishigin "Computing methods for building earthing devices in problems related to EMC of substations," EMC Europe 2012.

[2] N.V. Korovkin, O.V. Frolov, E.A. Ivanova and S.L. Shishigin, "Grounding and shielding in EMC problems of electric power substations" EMC Europe 2013.

[3] S.L. Shishigin, V.E. Meshcheryakov. "Lightning protection calculation method based on the electrostatic model of lightning leader attraction", Technical Physics Letters, March 2015, Volume 41, Issue 3 , pp 273-276, DOI: 10.1134/S106378501503027X.
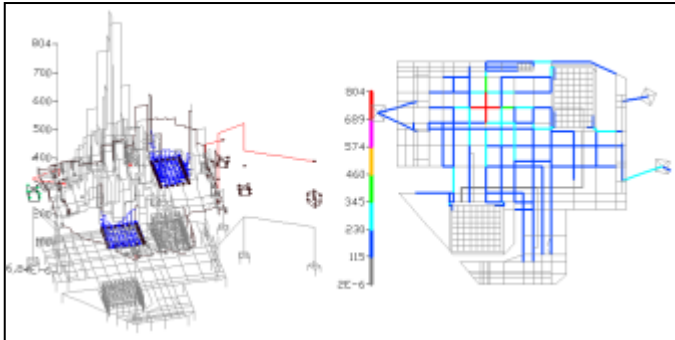
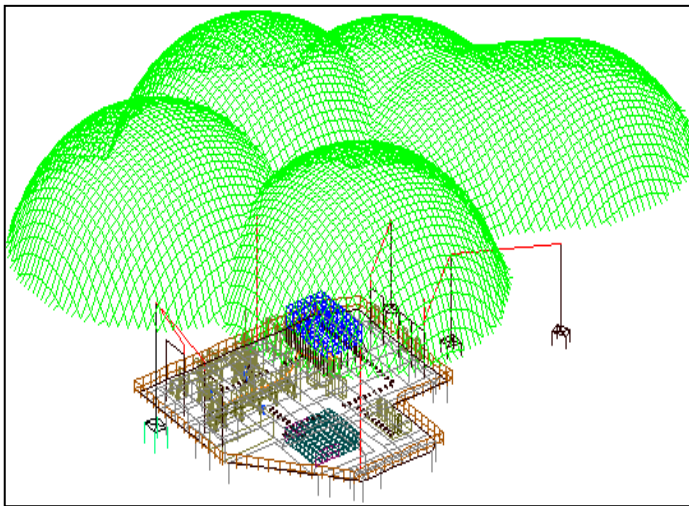Fig. 8   Short-circuit currents in 3D, 2D views

Fig. 9   Rolling sphere method: lightning-capture zone
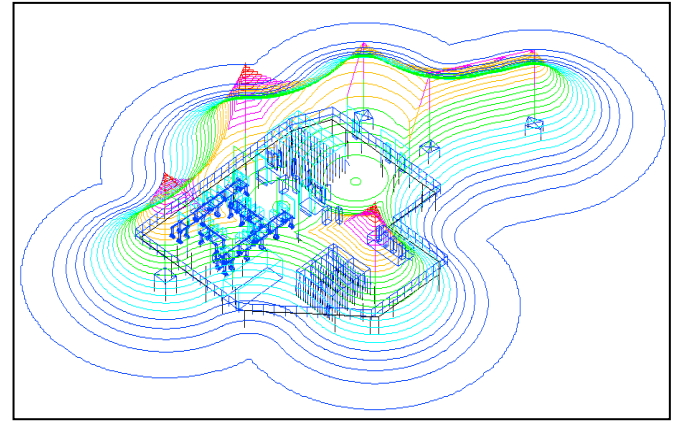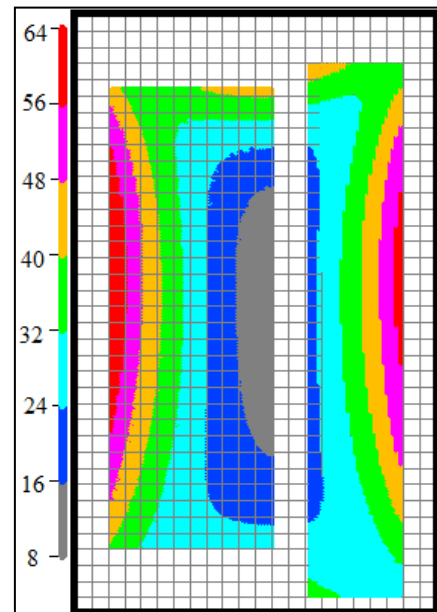

Fig. 10   Rolling sphere method: protection zone

Fig. 11   Magnetic field strength