Special Software Development to Customize ANSYS for Specific Applications

Alexey I. Borovkov
Denis V. Shevchenko
Oleg A. Zakirov
Computational Mechanics Laboratory, St.Petersburg State Polytechnical University, Russia

Abstract

The ANSYS software has powerful engineering analysis capabilities in many application fields some applications require the user to have a high level of knowledge. Industrial companies may not have enough specialists of such qualifications. In this circumstance the development of special software (shells) seems to be topical. Such software should allow engineers not having sufficient knowledge in up-to-date computational mechanics to use powerful ANSYS capabilities.

The ANSYS Parametric Design Language (APDL) enables engineers to create parametric models – this is the kernel for the shell software. A parametric model can be modified to efficiently solve problems of a specific type. For this quasi-optimal model, a number of settings can be changed. These settings include required parts of the model, material properties and boundary conditions, finite element mesh parameters, and so on.

Specialized shell software makes it convenient to solve multivariant and multiparametric problems of structural mechanics. The user-friendly interface enables the user to specify parameters, perform computations, display analysis results, and automatically create a report in required format.

In the current article, the description of two developed shells is presented.

1. Shell software to compensate rotor body stiffness dissymmetry - CRSD software. A totally parametric model of a rotor is developed that takes into account all main parts of the rotor: end parts, rotor body, windings and wedges. As a result of analysis performed, 60 main parameters were chosen to describe correctly the geometry of the rotor. The shell parametric model was specially adjusted to solve these specific problems. The quasi-optimal parametric model dramatically reduced solution CPU time in comparison with the initial model (from 50 hours to 30 minutes). The user can choose material properties for rotor elements, define stiffness of the bearings and loading (gravity, angular velocity).

2. Shell software to carry out steady-state 3D thermal analysis of radiator with about 30 fuel elements. A totally parametric model was developed that enables the user to vary the number of fuel elements and include thermal isolation disks. The user can define 6 geometrical parameters of the radiator, material properties of the radiator and fuel elements, boundary conditions at all surfaces (air-cooling, convective heat transfer to environment).

After analysis, both software shells automatically generate reports in MS Word that contain the main results of the analysis.

Introduction

The development of problem-oriented customized software includes the development of a parametric model that contains debugged and adjusted computational models for efficient solution of specific problems. It also includes development of shell software that has a user-friendly interface and enables engineer to perform calculations. In the present article, two developed problem-oriented shells are presented: software to eliminate turbogenerator rotor body stiffness anisotropy, “Compensation for Rotor body Stiffness Dissymmetry” (CRSD Software) and software to perform 3D steady-state thermal analysis of electronic radiator devices (Radiator Software).
**CRSD Customizing Software**

**Description of the Problem**

Steam-turbine electric current generators (turbogenerators) are being used at most power plants. Design engineering of turbogenerators of any power capacity is a complex and consuming task. It is implemented on a modern level by large design and technological departments that have significant experience in designing, manufacturing and implementing these machines.

The rotor or inductor is a critical part of turbogenerator. It consists of a shaft, excitation windings, windings isolation, and wedges holding windings. The rotor is the most loaded (mechanically as well as thermally) part. The rotating rotor is exposed to vibrations that can originate for following reasons:

a) out-of-balance of mass of rotor shaft as well as the windings;
b) unbalanced clearance between stator (stable part of the turbogenerator) and rotor;
c) closing between coil loops in the excitation winding;
d) poor center adjustment of turbogenerator rotors and turbine or dog leg appearance during operation;
e) weakening of the shroud joint, ventilators, and other elements fixed on the rotor;
f) thermal unbalance of the rotor expressed in cooling asymmetry of detached zones of rotor body.

Of special interest is the vibration with double rotational frequency that can be observed in general in rotors with significant relation of the rotor body length to its diameter. This vibration is the consequence of different stiffnesses in two main axes – the pole axis and the neutral axis perpendicular to the pole axis. Such vibration cannot be eliminated by installation of balance weights of the rotor shaft, and the only way to eliminate it is maximum equalization of rotor body stiffness in two directions.

One of the methods to lower rotor body anisotropy is implementation of cross slots, called Laffun’s slots, on the rotor body. Variations of slot depth lead to changes in rotor body cross section moments of inertia, which in turn leads to variations of rotor body anisotropy.

In the rotor there is complex contact interaction: rotor body interacts with winding, winding interacts with wedges and isolation, wedges interact with rotor body. Modeling of all contact interactions requires significant computational resources. For example, using a Intel Pentium-IV 1.9 GHz machine with 1 Gb RAM, the solution of a 3D contact problem with consideration of all structural elements of the rotor (FE model contains ~ 600,000 degrees of freedom and ~ 30,000 contact elements) takes about 50 hours.

To obtain reliable results in a reasonable time, a special engineering approach was developed. The solution of a series of 3D elastic problems with contact interaction revealed that the model with contact interactions can be replaced by a model without contact interaction. The interactions between wedges and other rotor parts can be neglected, and the interactions of windings and wedges can be replaced with rigid constraints. This method enables lower solution time to 30 minutes. The difference between this engineering approach and the solution with all contact interactions is 0.5%.

**CRSD Software**

Realization of rotor body stiffness dissymmetry compensation implies changing of model and new solution at every step. Rotors of up-to-date turbogenerators possess complex geometries (Figure 1 from UG), and it takes several days for a CAD professional to create the new model.

![Figure 1. Geometric layout of the turbogenerator rotor](image-url)
Modification of created models (for example, variation of number of cross slots – See Figure 2 from ANSYS) also takes several days. So, elimination of stiffness anisotropy is a very time consuming job.

Figure 2. Cross slots and winding slots of the turbogenerator rotor

To achieve labor savings for the procedure of stiffness anisotropy compensation, a totally parametric model was developed in ANSYS. It contains 62 parameters (2 angles, 38 lengths, 19 diameters, radius of cross slots, number of windings, number of cross slots) and enables users to perform FE analysis of rotors of practically any shape (see Figures 3-4). In the Appendix an APDL fragment of the parametric model is presented. It illustrates building of the rotor body cross section.

Figure 3. Rotor with 6 cross slots and 4 windings

Figure 4. Rotor with 10 cross slots and 8 windings
For convenient usage of the developed parametric model, the problem-oriented shell software (CRSD Software) was created.

The basic goal of creating the CRSD Software was to provide a complete tool that enables engineers to quickly and efficiently utilize ANSYS and APDL capabilities with the developed rotor parametric model.

The CRSD Software, working on the basis of the developed parametric model, enables the user to:

- Create new designs of a rotor (all 62 parameters of the new rotor should be set);
- Change existing rotor designs, which enables them to be used as initial approximation for new designs;
- Specify material properties for rotor structural elements, consider rotor bearings stiffness;
- Analyze deformed state of the two-pole turbogenerator rotor and eliminate rotor body stiffness anisotropy;
- Carry out analysis for various loadings (gravity body force, angular velocity);
- Output (after solution) a graph illustrating dependency of rotor axial movement on longitudinal coordinate. This allows one to evaluate rotor body stiffness anisotropy based on the solutions of two problems;
- Generate report in Microsoft Word.

**Rotor Body Stiffness Equalization**

The CRSD Software enables users to perform rotor body stiffness anisotropy compensation. After the solution, the CRSD Software creates a report containing a graph illustrating dependency of rotor axial movement on longitudinal coordinate. This helps to evaluate rotor body stiffness anisotropy on the basis of maximum static sag values. It is convenient to use as an estimation the relation between the difference of maximum static sag and minor values as described below. (1):

$$\delta = \frac{|U_{\text{max}1} - U_{\text{max}2}|}{\inf\{|U_{\text{max}1}|, |U_{\text{max}2}|\}} \times 100\% \quad (1)$$

**Algorithm**

1. The problem is solved for the rotor under a gravity force acting in the first plane of symmetry. On the basis of the resulting graphs, maximum axial displacement $U_{\text{max}1}$ is defined;
2. The problem is solved for the rotor under a gravity force acting in the second plane of symmetry. On the basis of the resulting graphs, maximum axial displacement $U_{\text{max}2}$ is defined;
3. On the basis of two maximum static sag values, stiffness anisotropy is evaluated and changes in rotor geometry are made (for example, depths of Laffun’s slots is varied). After this geometry change, the computation is repeated for both loading directions, and stiffness anisotropy is evaluated once more. If the value of stiffness anisotropy becomes less then the predefined value, the process is stopped.

**CRSD Interface**

CRSD Software is created with use of code development system Free Pascal. Object-oriented library is also developed (analogous to Microsoft Foundational Classes in Microsoft Visual C++ and Visual Component Library in Borland Delphi) that enables one to create Graphical User Interface controls in a convenient manner without accessing Windows API functions.

CRSD software Graphical User Interface is presented in Figure 5. Using the menu system and toolbar engineers can create, open, and edit rotor designs, save created designs, perform computations, and generate reports after the solution.
The **Main Window** contains menus for access to all software functions and a toolbar to access the most frequently used commands.

In the **Project Window**, a menu tree structure allows the engineer to input data for the rotor design.

The **Navigation Window** is used to output additional information in the current program context (specification of the current parameter on the drawing, information about loadings, values of parameters, etc.).

The **Project window** (Figure 6) is used to input and edit design data. Section **Geometry** is used to set geometrical parameters of the current design. Section **Materials** is used to set material properties. Section **Boundary conditions** is used to set type of the bearings (rigid, flexible with specified stiffness). Section **Load** is used to choose the type and orientation of loadings.

After input of all parameters the computation is carried out. After solution the report is automatically generated. The CRSD Software creates the report in Microsoft Word (see Figure 7). The report contains the drawings, values of geometrical parameters, and material properties. A graph illustrating dependency of rotor axial movement on longitudinal coordinate is included in the report.
Radiators are utilized in electronic devices where significant emission of power in components occurs, and it is necessary to consider cooling to prevent element failure. Direct FE modeling can be used to analyze thermal fields originating in the radiator and evaluate the temperature drop for fuel elements installed on the radiator.

### 3D Steady-State Thermal Analysis of Radiator

When analyzing the thermal state of the radiator and fuel elements, the transient process of temperature redistribution is of no importance while the steady thermal state is of interest. This fact allows considering just the steady-state problem of heat transfer.

In the heat transfer problem, the following types of conditions exist:

- Geometrical data that characterize shape and dimensions of the body where the heat transfer process is to be analyzed;
- Physical conditions that characterize thermo-physical properties of the materials: thermal conductivity, density, specific heat and internal sources of heat;
- Boundary conditions that characterize thermal interaction of the body surface with surroundings;
- Initial conditions that are used to specify temperature distribution in the body at initial time. These conditions are essential when modeling transient processes.

Consider the thermal state of the radiator with three installed fuel elements and isolating gaps between the radiator and elements. The radiator material of radiator is duralumin; the isolating gaps material is mica. Fuel elements are simulated by setting thermal fluxes at the surface of the isolating gaps.

Figure 8 shows a 3D FE model of a radiator with isolating gaps.
On the surface of the isolating gaps, thermal fluxes are set. On the surfaces of the radiator, various heat-flux densities are set (see Figure 9) as well as various heat transfer factors:

\[
\alpha = \begin{cases} 
\alpha_1, & \text{on the top surface} \\
\alpha_2, & \text{on the left surface} \\
\alpha_3, & \text{on the right surface} \\
\alpha_4, & \text{on the inner surface} \\
\alpha_5, & \text{on the back surface} \\
\alpha_6, & \text{on the front surface} \\
\alpha_7, & \text{on the bottom surface}
\end{cases}
\]

and various temperatures on the surfaces of radiator:

\[
T_\infty = \begin{cases} 
T_\infty^1, & \text{on the top surface} \\
T_\infty^2, & \text{on the inner surfaces} \\
T_\infty^3, & \text{on the right surface} \\
T_\infty^4, & \text{on the left surface} \\
T_\infty^5, & \text{on the back surface} \\
T_\infty^6, & \text{on the front surface} \\
T_\infty^7, & \text{on the bottom surface}
\end{cases}
\]

The steady thermal state is considered, so initial conditions are not specified. The resulting temperature field is presented in Figure 9.
Radiator Software

Radiators often have the same shape and differ only by their sizes. During the design of similar radiators, engineers waste time creating new radiator models. To lower labor costs of radiator thermal analysis, a totally parametric model of the finned radiator was developed with use of APDL.

This parametric model enables users to analyze the 3D thermal state of the radiator and set the following parameters:

- 7 parameters to describe the radiator: 5 geometrical sizes (see Figure 10), number of fins, thermal conductivity of radiator material;
- 8 parameters from every fuel element: element shape (rectangle or round); presence of isolating gap; thickness and thermal conductivity coefficient of the gap; power emission; positions of fuel elements and their shape and sizes (length, width or radius);
- Conditions of convective heat transfer with environment; temperature for every surface of radiator; type of surface cooling (natural air cooling or forced air cooling). For the case of forced air cooling the velocity of air is to be set.

On the basis of the created parametric model, the shell Radiator Software was created. It enables the user to:

- Create new projects of radiator analysis (in this case all radiator parameters are to be specified);
- Open and edit existing projects of radiator analysis;
- Perform analysis of the 3D steady thermal state of the radiator with various number of fins and various number of fuel elements installed on the working surface of the radiator;
- Plot resulting temperature fields;
- Output average temperature of fuel elements;
- Output average temperature of radiator surfaces;
- Calculate aerodynamic resistance;
- Generate a report in Microsoft Word;

The interface of the Radiator Software is presented in Figure 11.

![Figure 11. Main windows of Radiator Software](Image)

The **Main Window** includes menus and toolbar with buttons corresponding to main functions of the software.

The **Project Window** is used to select and edit design parameters such as geometry of the radiator and fuel elements, and boundary conditions.

The **Model Window** displays the geometry of radiator and fuel elements. A prompt is also shown to specify correct boundary conditions on the surfaces of the radiator.

**Algorithm**

To carry out research using the Radiator Software, the user must:

1. Specify radiator parameters (after this model geometry will be shown in the **Model Window**);
2. Define fuel elements and specify their properties;
3. Define conditions of convective heat transfer for all surfaces of radiator.

After specification of all essential parameters, the solution is performed. After the solution, the user can use the **Control Points Window** (Figure 12) to define Control Points and evaluate temperature at these points. The Radiator Software allows the user to create a report in Microsoft Word (see Figure 13).
Conclusions

Two problem-oriented shells were developed: CRSD Software to carry out rotor body stiffness anisotropy compensation (Compensation for Rotor body Stiffness Dissymmetry) and Radiator Software to carry out 3D steady-state thermal analysis of electronic radiator devices.

In the process of software development, two ANSYS parametric models were created with use of APDL:

1. A totally parametric model of turbogenerator considering such rotor elements as rotor body, windings, wedges. The model contains 62 parameters and enables the user to analyze the deformed state of rotors of various shapes and designs.
An engineering approach was developed that enables the user to replace a 3D contact problem with all structural elements of the rotor with a linear solution, which reduces computation time from 50 hours to 30 minutes.

2. A parametric model of a radiator that the user allows to carry out 3D steady-state thermal analysis of a radiator with installed fuel elements.

The developed software shells are easy-to-use engineering tools that allow engineers not familiar with advanced FEA to carry out computations and use powerful ANSYS capabilities.

The following customization algorithm is suggested:

1. Creation and tuning of the parametrical model for the chosen object of analysis. At this step the mechanical behavior of the object is analyzed, critical parameters of the model and FE mesh are chosen.

2. With use of approachable code development software create user-friendly graphical interface and automatic report generation block according to corporate standards of the customer. This will enable engineers to input data post-process the analysis in a convenient manner and generate a report.

For the development of user interface of the shell software one can use both free code development utilities such as Free Pascal, and commercial packages developed by world-leading companies like Microsoft Corporation (Visual Basic, Visual C++) and Borland Software Corporation (C++ Builder, Delphi).

Developed in 2002 special software ANSYS Workbench Environment also gives vast opportunities to create engineering applications.

**Appendix**

Fragment of the APDL parametrical model building rotor body cross section:

```plaintext
CSYS,1
LGEN, ,ALL, , ,,-A1, , , ,1
*do,i,1,n/2-1,1
LGEN,2,ALL, , ,,-A2*i, , , ,0
LSEL,S,,,1
*do,j,2,NumberLines,1
LSEL,A,,,j
*enddo
*enddo
ALLSEL,ALL
K,1000001,0,0,0,
K,1000002,R10,90,0,
K,1000003,R10,90-(A1-asin(R20/R10))/2,0,
LARC,1,1000002,1000003
K,1000004,R10,0,0,
K,1000005,R10,(A2-2*asin(R20/R10))/4,0,
LARC,(n-1)*NumberKPoint+1,1000004,1000005
*GET,coord4,KP,4,LOC,X
*GET,theta4,KP,4,LOC,Y
K,1000006,coord4,90,0,
K,1000007,coord4,90-(A1-asin(R20/R10))/2,0,
LARC,4,1000006,1000007
```
K,1000008,coord4,0,0,
K,1000009,coord4,(A2-2*asin(R20/R10))/4,0,
LARC,(n-1)*NumberKPoint+4,1000008,1000009
L,1000008,1000004
NUMCMP,LINE
*GET,coord8,KP,8,LOC,X
*GET,theta8,KP,8,LOC,Y
K,1000010,coord8,90,0,
K,1000011,coord8,90-(A1-asin(R20/R10))/2,0,
LARC,8,1000010,1000011
K,1000012,coord8,0,0,
K,1000013,coord8,(A2-2*asin(R20/R10))/4,0,
LARC,(n-1)*NumberKPoint+8,1000012,1000013
L,1000012,1000008
CSYS,0
K,1000014,0,R10-L26,0,
K,1000015,R1,R10-L26+R1,0,
K,1000016,R1/2,R10-L26+R1-sqrt(R1*R1-(R1/2)*(R1/2)),0,
LARC,1000014,1000015,1000016
CSYS,1
KSEL,S,,1000002
LSLK,S
*GET,number1,LINE,0,NUM,MAX
ALLSEL,ALL
KSEL,S,,1000006
LSLK,S
*GET,number2,LINE,0,NUM,MAX
ALLSEL,ALL
KSEL,S,,1000014
LSLK,S
*GET,number3,LINE,0,NUM,MAX
ALLSEL,ALL
LSBL,number3,number1
NUMCMP,LINE
KSEL,S,,1000015
LSLK,S
*GET,number1,LINE,0,NUM,MAX
LDELE,number1
ALLSEL,ALL
KSEL,S,,1000006
LSLK,S
*GET,number2,LINE,0[NUM,MAX
ALLSEL,ALL
KSEL,S,,1000014
LSLK,S
*GET,number3,LINE,0[NUM,MAX
ALLSEL,ALL
LSBL,number3,number2
NUMCMP,LINE
*GET,theta1,KP,NumberKPoint*n+1,LOC,Y
*GET,theta2,KP,1,LOC,Y
K,1000002,R10,90,0,
K,1000017,R10,theta1-(theta1-theta2)/2,0,
K,1000018,R10,90-(90-theta1)/2,0,
L,1000001,1000012
L,1000001,1000010
L,1000010,1000014
L,1000014,1000002
LARC,1,NumberKPoint*n+1,1000017
LARC,1000002,NumberKPoint*n+1,1000018
*GET,theta1,KP,NumberKPoint*n+2,LOC,Y
*GET,theta2,KP,4,LOC,Y
K,1000019,coord4,theta1-(theta1-theta2)/2,0,
LARC,4,NumberKPoint*n+2,1000019
KDELE,1000019
KDELE,1000018
K,1000019,coord4,90-(90-theta1)/2,0,
K,1000018,coord4,90+1,0,
LARC,1000018,NumberKPoint*n+2,1000019
*GET,number2,LINE,0[NUM,MAX
*GET,coord14,KP,1000014,LOC,X
LSEL,S,LOC,X,coord14-0.01,R10
LSEL,R,LOC,Y,90-0.01,90+0.01
*GET,number1,LINE,0[NUM,MAX
ALLSEL,ALL
LSBL,number1,number2
NUMCMP,LINE
LARC,NumberKPoint*n+3,NumberKPoint*n+2,1000019
ALLSEL,ALL
delta = 90-A1+A2/2
*do,i,1,n/2-1,1
K,600+i,R10,delta-A2*i,0,
K,600+n/2-1+i,coord4,delta-A2*i,0,
K,600+2*(n/2-1)+i,coord8,delta-A2*i,0,
*enddo
*do,i,1,n/2-1,1
LARC,2*NumberKPoint*i-12,2*NumberKPoint*i+1,600+i
LARC,2*NumberKPoint*i-9,2*NumberKPoint*i+4,600+n/2-1+i
LARC,2*NumberKPoint*i-5,2*NumberKPoint*i+8,600+2*(n/2-1)+i
*enddo
ALLSEL,ALL

References
1) Ansys APDL Programmer’s Guide. SAS IP, Inc.