Simulation and Optimization of Aircraft Assembly Process Using Supercomputer Technologies

Tatiana Pogarskaia✉ [0000-0003-0314-7369], Maria Churilova [0000-0002-3709-4886], Margarita Petukhova [0000-0002-9357-4809] and Evgeniy Petukhov [0000-0003-0152-5573]

Peter the Great St.Petersburg Polytechnic University, 195251, Saint Petersburg, Russia
pogarskaya.t@gmail.com
m_churilova@mail.ru
margarita@lamm.spbstu.ru
eugene@lamm.spbstu.ru

Abstract. Airframe assembly is mainly based on the riveting of large-scale aircraft parts, and manufacturers are highly concerned about acceleration of this process. Simulation of riveting emerges the necessity for contact problem solving in order to prevent the penetration of parts under the loads from fastening elements (fasteners). Specialized methodology is elaborated that allows reducing the dimension and transforming the original problem into quadratic programming one with input data provided by disposition of fasteners and initial gap field between considered parts.

While optimization of a manufacturing process the detailed analysis of the assembly has to be done. This leads to series of similar computations that differ only in input data sets provided by the variations of gap and fastener locations. Thus, task parallelism can be exploited, and the problem can be efficiently solved by means of supercomputer.

The paper is devoted to the cluster version of software complex developed for aircraft assembly simulation in the terms of the joint project between Peter the Great St.Petersburg Polytechnic University and Airbus SAS. The main features of the complex are described, and application cases are considered.

Keywords: Aircraft Assembly · Optimization · Supercomputing · Task Parallelism · Quadratic Programming.

1 Introduction

During the assembly process, it is important to control both gaps between joined parts and stresses caused by installed fastening elements. On the one hand, tight contact between parts should be achieved; and on the other hand, engineers should avoid cracks, composite layer delamination, and part damage.

The main goal of the presented work is to develop a special tool that allows performing simulations in order to evaluate displacements and stresses of aircraft parts on the assembly line. For this purpose, specialized software complex ASRP (Assembly Simulation of Riveting Process) is developed for contact problem solving. As a result, we determine the deformed stress state of the assembly loaded by the forces from fastening elements.

This contact problem has following peculiar properties to be taken into account in order to derive efficient algorithm:

1. The contact may occur only in junction area that is known a priori. Thus, there is no need to implement complicated procedures for detection the zone of possible contact.
2. The installed fasteners and rivets restrict relative tangential displacements of assembled parts in the junction area. Therefore, the relative tangential displacements in junction area are negligible in comparison with normal ones. This special feature of the problem justifies implementation of node-to-node contact model that is much simpler than general surface-to-surface model.
3. Loads from fastening elements are applied inside junction area.
4. Only the stationary solution of the problem is of interest.
5. Friction forces between assembled parts in the contact zone do not play significant role due to small relative tangential displacements. So the friction can be omitted from consideration.
6. Stress state of each part in the assembly is described by the linear theory of elasticity.

Solving of considered contact problem comes down to the variation simulation that is used to predict the final assembly state taking into account the part variations. These variations arise from the manufacturing tolerances and can be provided by measurement data or statistical models (see [1], [2]).

The simplest approach is rigid variation simulation when the part deformations are excluded from consideration as in [3]. Consequently, the results are far from the reality. If the mechanical behavior of assembled parts is involved in simulation, then there is a need for finite element analysis (FEA). FEA is used in number of studies and is implemented in specialized commercial software for tolerance analysis [4-6].

Direct application of FEA in variation simulation is inefficient, as even one FEA run may take considerable time for real aircraft models. To overcome this problem, the Method of Influence Coefficients (MIC) is introduced in [1]. The MIC approach establishes linear relationship between a part variation and corresponding assembly variation via sensitivity matrix calculated by FEA. However, possible contact interaction of parts is neglected. Authors of [7] combined MIC with contact modeling for variation simulation in automotive industry.

This paper presents the approach developed in [8] that is similar to MIC to some extent. Reduced stiffness matrix is computed (like sensitivity matrix of influence coefficients), and contact problem is transformed into Quadratic Programming Problem (QPP). Then efficient algorithms are derived for QPP solving. Studies [9], [10] suggest the analogous ideas.

2 The Basics of Numerical Algorithm

We would give here only the main idea of the method, details of numerical algorithms, as well as validation tests, are described in [8].

Let us consider artificial finite element model of the upper wing-to-fuselage junction that is shown in Fig. 1 and Fig. 2 (fragment). There are two parts in the assembly: the first part is the wing (light blue) and the second one is the fragment of center wing box (yellow).

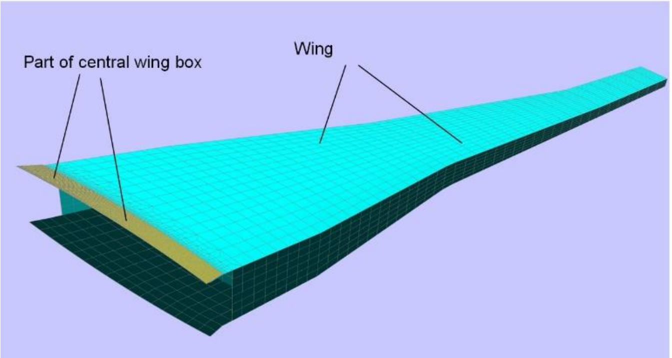

Fig. 1. Finite element model of an artificial wing-to-fuselage junction.

Green points in Fig. 2 mark the nodes of possible contact (the nodes in junction area). We denote these nodes as **computational** ones. The set of all computational nodes is referred as **calculation net**.

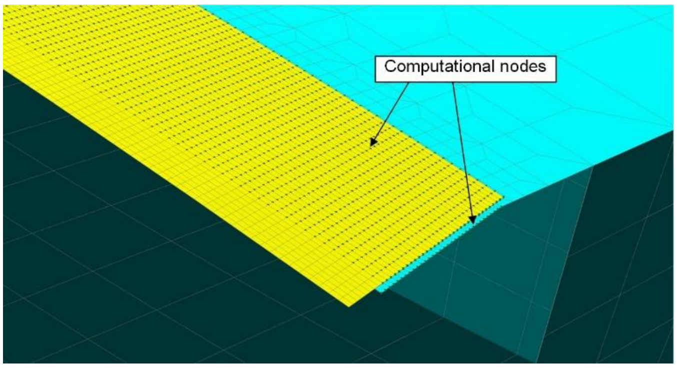

Fig. 2. The nodes of junction area.

Using the standard finite element modeling technique, we formulate the contact problem in discrete variation form [11]:

$$\min_{U \in S_h} \left(\frac{1}{2} U^T \cdot K \cdot U - F^T \cdot U \right) \quad (1)$$

Here U is the displacement vector of finite element nodes, K is the stiffness matrix of finite element system, F is the vector of applied loads, S_h is the admissible set that is determined with regard to boundary and non-penetration conditions.

Typically, number of nodes in a finite element model of airframe junction (e.g. wing-to-fuselage junction, as it is shown in the figures above) is much bigger than number of nodes in the junction area. Therefore, the elimination of displacements outside the junction area reduces the problem dimension dramatically. However, we have to divide all the finite element nodes into two groups containing nodes in junction area (green points in Fig. 2) and all the rest. Then the displacement vector can be written as follows

$$U = \begin{pmatrix} U_C \\ U_R \end{pmatrix},$$

where U_C is the vector of node displacements in junction area and U_R is the vector of displacements in other finite element nodes. The same procedure can be implemented for matrix $K = \begin{pmatrix} K_{CC} & K_{CR} \\ K_{CR}^T & K_{RR} \end{pmatrix}$ and vector $F = \begin{pmatrix} F_C \\ F_R \end{pmatrix}$.

Calculating Schur complement of K_{RR}, we get reduced stiffness matrix K_C from the formula $K_C = K_{CC} - K_{CR} \cdot K_{RR}^{-1} \cdot K_{CR}^T$. Now it is possible to derive the reduced QPP:

$$\min_{N \cdot U_C \le G} \left(\frac{1}{2} U_C^T \cdot K_C \cdot U_C - F_C \cdot U_C \right) \qquad (2)$$

where N is the linear operator which defines normal to contact surface, G is the initial gap vector in the junction area.

In addition, we mention that U_C contains only the normal components of node displacements. This simplification is possible due to the smallness of tangential displacements regarding normal ones.

Thus, we reduce the initial contact problem to the quadratic programming problem. Moreover, the dimension of the reduced problem is much smaller (e.g. the finite element model depictured in Fig. 1 and Fig. 2 has around 130 000 degrees of freedom and the reduced problem (2) for this model has only 16 000 unknowns). This approach is known as substructuring in finite element modeling.

Similar approaches to contact problem solving can be found in [9, 10, 12].

3 Software Overview

3.1 ASRP Desktop Version

The desktop version of the software for assembly simulation is divided into three modules: Preprocessor, Simulator and Postprocessor in order to fully separate the data preparation from the assembly simulation process and subsequent detailed stress analysis, see [13].

ASRP Preprocessor.

Preprocessor is designed to prepare models for Simulator on the base of imported finite element model of the assembly in MSC Nastran format.

Preprocessor generates all data structures required for Simulator:

- Reduced stiffness matrix describing mechanical properties of assembled parts. The matrix is computed using MSC Nastran as external finite element solver;
- Geometry for visualization created from finite element mesh;
- Positions of points used for determination of initial gap in ASRP Simulator;
- Positions and diameters of holes for fastening elements in every part.

ASRP Simulator.

Simulator is the central part of ASRP software complex. It is designed for the riveting process simulation. This tool permits calculating gaps between assembled parts, absolute displacements, reaction forces caused by contact in junction area, loads in fastening elements needed to achieve contact. In addition, Simulator provides great variety of extra tools for simulation of riveting process and optimization of assembly technology:

- Capabilities for statistical analysis using sets (clouds) of random gaps. For example, user can compute the percentage of examined points with resulting gap within given range (e.g. less than 0.2 mm) for predefined arrangement of fasteners;
- Automatic positioning of fastening elements in order to minimize gap by given number of fasteners. In doing so user can consider either determined initial gap or the cloud of random gaps with given roughness and deviation;
- Powerful tools for editing and visualization of fastening elements (including work with groups of fastening elements, a special library of standard fastening elements etc.);
- Different options for visualization of simulation results;
- Automation of simulation process using script files.

The Simulator is a standalone application that does not need any external software (like MSC Nastran) but can exchange data with other ASRP parts and third party software (e.g. import of measured initial gap or export the instructions for fitting machining).

ASRP Postprocessor.

ASRP Postprocessor is aimed at computing the stresses caused by the riveting process.

The main purpose of this module is to evaluate the stresses arising during the assembly process of aircraft junction without solving the contact problem by standard means of finite element analysis but using the results of ASRP simulation instead. Thus, the input data for Postprocessor are the finite element model of junction appropriate for stress computations (in MSC Nastran format) and the file with computation results exported from ASRP Simulator. Postprocessor makes it possible to apply the results imported from ASRP Simulator to the finite element model as the boundary conditions for subsequent static stress analysis.

3.2 ASRP Cluster Version

In order to obtain robust and reliable results, the assembly should be thoroughly analyzed over the wide range of input data that may include the initial gap measurements from the final assembly line or the information about geometric tolerances. Even if the input (i.e. initial gap) is undefined during manufacturing stage, it can be generated in ASRP Simulator using statistical methods and certain gaps properties [14].

According to the assembly technology, the aircraft parts are temporary connected with fasteners installed in about 50% of all holes. These fasteners are called temporary ones and their main objective is to provide contact between the parts for further technological operations. The challenge we face is to find the best temporary fastener positions (fastener pattern) using minimum possible number of elements that still provides sufficient quality of the assembly or to rearrange existing fasteners to minimize all the gaps.

Thus, we have to deal with the clouds of initial data that may account several hundreds of entities that obviously cause significant increase of computations. To overcome this difficulty, the cluster version of ASRP Simulator is developed.

ASRP Cluster Version (CV) is a console application written in C++ that uses MPI for process communication. It does not require any external libraries for executing. Fig. 3 illustrates the flowchart of ASRP CV.

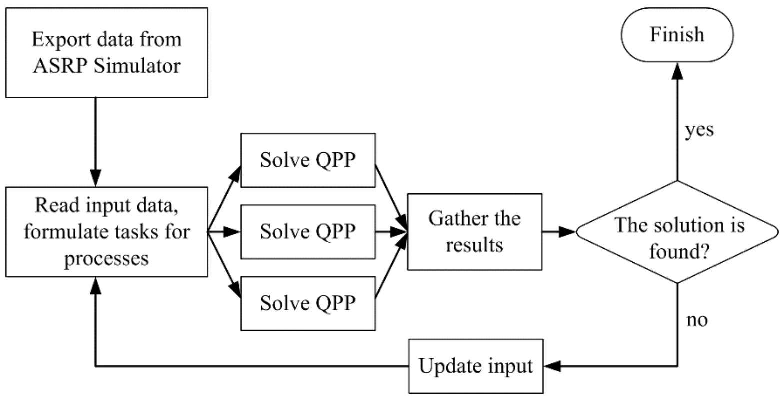

Fig. 3. ASRP CV flowchart

The computations with different input data are so-called task parallel as they can be done independently. The aim is to state the correct QPP problem for each parallel process and then to gather the resulting data at the root process.

The core function in the flowchart is solving QPP (2). Three solvers that are based on the most common methods for such kind of problems are implemented in ASRP CV:

1. Active Set method [15-16] and its adaptation to the features of given problem [8];
2. Interior Point method [17];
3. Projected Gradient method [18] and its adaptation.

Depending on the considered model user may choose the most appropriate and the fastest solver.

Let us consider the application issues of ASRP CV.

Verification of Current Fastener Configuration.

The gap values computed under the loads from the fastener configuration are checked against some predefined value for each initial gap from the cloud. The percentage of "closed" (relatively small) gaps is calculated.

Each process receives its own initial gap field, QPP is solved, and then the root process gathers the statistics for computed gap values. Verification is done in one pass, no update input in Fig. 3 is needed.

Fastener Initial Positioning.

Sometimes it is necessary to install a fixed number of new fastening elements. This process is iterative and starts having no new fasteners installed. Each process solves the QPP with the specific gap and identifies the hole for the next fastener according to some criteria. Then the root process gathers data from all the processes, chooses the most suitable hole for fastener installation, and broadcasts its index to all the processes. The algorithm continues until the required number of installed fasteners is reached.

Optimization of Fastener Positions.

Due to the time-consuming calculation of the objective functions (calculating resulting gaps with each pattern modification for hundreds of initial gaps) and impossibility to calculate its derivatives, the local variations' method is applied. The optimization procedure is an iterative exhaustive search of optimal position for each fastener one-by-one among predefined holes.

The local variations' algorithm for minimizing function $F(P)$, where P is a vector of hole numbers where fasteners are installed and P_0 is the initial pattern, is as follows:

Initialization: $P := P_0$, Iteration := 1.
Repeat
  Set Progress := false;
  For each hole i with installed fastener
    For each empty hole j;
      Obtain pattern P^* by moving fastener from hole i to hole j;
      For each initial gap of the cloud
        Calculate the resulting gap with fastener pattern P^*;
      End for
      Evaluate $\Delta F = F(P^*) - F(P)$;
      If $\Delta F < 0$, keep the new pattern $P = P^*$ and set Progress := true;

End for
End for
Iteration := Iteration +1;
Until Progress = false (no *F(P)* improvement in one iteration).

The local variations' algorithm is implemented in ASRP CV. Iterations continue until the algorithm has converged to some local optimum. The benefit of this approach is that only fastener patterns that improve the goal function are accepted and therefore the optimization algorithm can be stopped at any moment.

4 Application Example

As an application example of the described methodology, we consider a problem of a real aircraft[1] when it is necessary to improve current temporary fastener pattern for a wing-to-fuselage junction. The gap between the parts should be reduced to a given critical value by rearranging the constant number of fasteners.

While assembly, the wing is positioned slightly below the Central Wing Box (CWB), and initial gap between the parts is measured in several predefined points along the junction area (see Fig. 4). The optimized temporary fastener pattern has to be suitable for any similar parts of one aircraft series what makes the procedure difficult as the number of measured initial gaps is very limited. In order to avoid this problem and guarantee fastener pattern quality, the optimization is performed over a set of artificial initial gaps modeled on available real measurements.

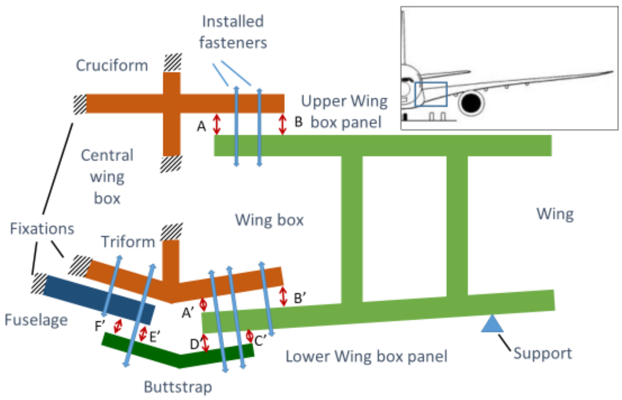

Fig. 4. Mechanical scheme of wing-to-fuselage junction

For the considered model, the set of 209 measured initial gaps was provided (initial gap cloud). The wing-to-fuselage junction model consists of two independent junction areas. The first one includes the upper wing panel and the part of CWB with 7308 computational nodes in junction area and the second one includes a lower wing panel and the part of CWB with 6452 computational nodes and has more complex geometry.

[1] Due to confidentiality reasons, the model details could not be provided in the paper.

Temporary fasteners on these two junctions have to be rearranged. Thus, the objective function is chosen as the percent of computational nodes where the gap between connected parts exceeds the desired value for all initial gaps in the cloud.

Results for the Upper Panel.
The optimization steps are described in Table 1. After three iterations, the optimization procedure stops due to no further improvement. The total number of gap computations is about 430 000 what would take nearly 3.5 years of computations on a personal computer without parallelization.

Table 1. Optimization steps for upper panel junction.

Operation	Computational time	Goal function, %
Computation of the goal function value	4.6 min	0.466 (7127 nodes)
Optimization, 1st iteration	53.2 h	0.263 (4013 nodes)
Optimization, 2nd iteration	52.1 h	0.244 (3727 nodes)
Optimization, 3rd iteration	52.1 h	0.242 (3694 nodes)
Total computational time	6.6 days	

The Fig. 5 illustrates the results of a new (optimized) pattern validation for four different gap clouds. The gap cloud on which the optimization is done is denoted as №3. The different gap clouds were obtained by adding local roughness to the measurements in order to simulate part variations. The methodology of initial gap generation is described in detail in [14].

The percent of computational nodes with gap less than X mm is plotted on the vertical axis in the Fig. 5. Thus, the percent of nodes where the gap is less than 0.2 mm for all gaps in this cloud is around 97% for gap cloud №4 and the initial fastener pattern. For all gap clouds with the optimized pattern, the plot lines (dashed) are located above the lines, corresponding to the initial pattern (solid) which means that the resulting gaps are decreased after fastener rearrangement.

Results for the Lower Panel.
The optimization steps for a lower panel are described in Table 2. One gap computation for this model is almost two times longer than for the previous one because of more complex geometry. The total number of gap computations is about 554 000 what would take nearly 7 years of computations on a personal computer without parallelization.

Table 2. Optimization steps for lower panel junction.

Operation	Computational time	Goal function, %
Computation of the goal function value	7.6 min	0.706 (9529 nodes)
Optimization, 1st iteration	115.1 h	0.567 (7647 nodes)
Optimization, 2nd iteration	105.7 h	0.563 (7597 nodes)

Optimization, 3rd iteration	106.2 h	0.563(7593 nodes)
Total computational time	13.6 days	

The Fig. 6 illustrates the results of a new (optimized) pattern validation for three different gap clouds. Optimization is done for gap cloud №2. The figure shows that the initial temporary fastener pattern eliminates gaps almost everywhere. Therefore, the optimization provides only slight improvement.

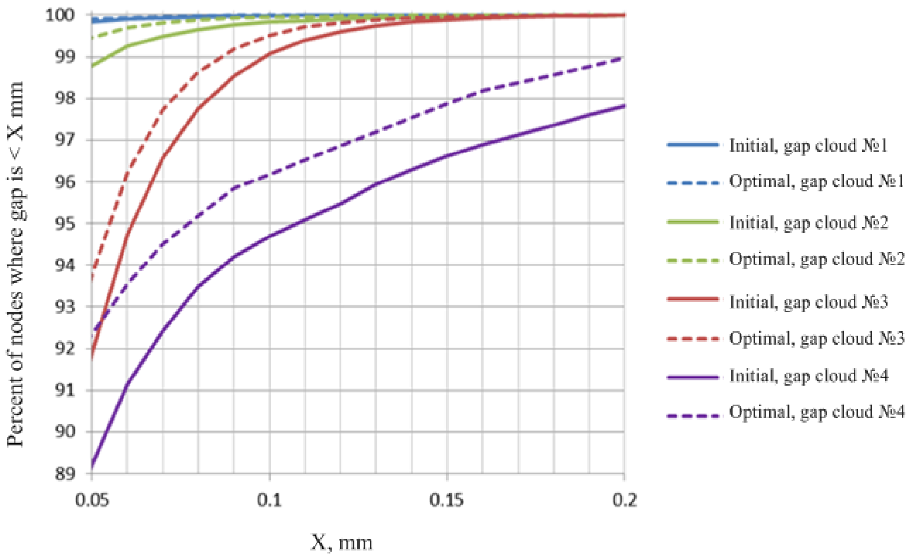

Fig. 5. Validation of optimized fastener pattern for upper panel

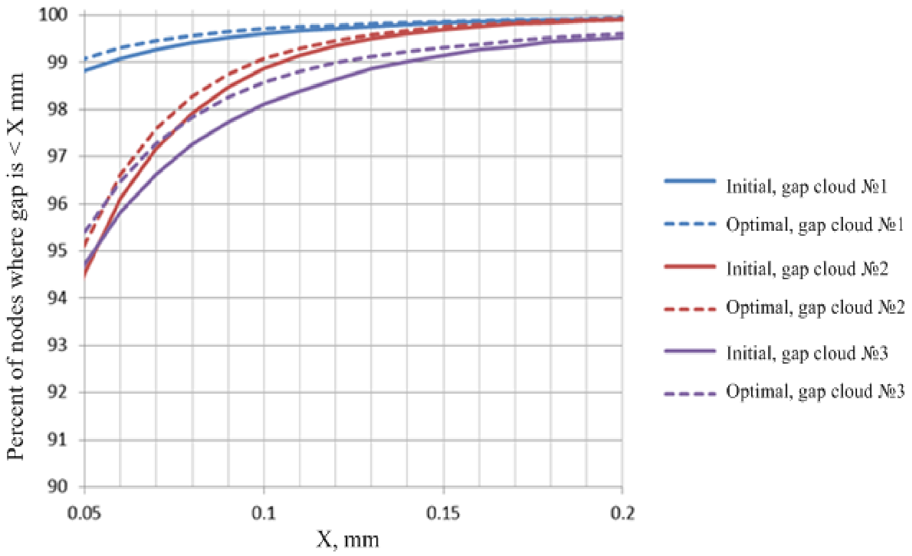

Fig. 6. Validation of optimized fastener pattern for lower panel

5 Conclusion

ASRP complex is developed for diverse but very specific engineering challenges. Some of these problems can be solved using commercial software, such as MSC Nastran, ANSYS etc., but ASRP application results in gain of time and a better quality. It can be explained by the fact that specialized algorithms and data saving strategies are implemented for solving of the narrow contact problem class. The key feature of ASRP optimization and verification methodology is assembly analysis over a cloud of initial gaps that involves series of similar computations for different initial gaps. This makes possible to parallelize the main optimization procedure and to use high-performance computers for executing simultaneous processes.

This software is successfully applied both to modification of existing fastener pattern for wing-to-fuselage junction and to the validation of new pattern against the old one. According to the obtained results, assembly engineers can update the technology at the final assembly line.

We expect the further investigations will be aimed at improvement of ASRP CV, investigation of parallel computing technologies for better performance.

The results of the work are obtained using computational resources of Peter the Great St.Petersburg Polytechnic University Supercomputing Center (www.spbstu.ru).

References

1. Liu, S.C., Hu, S.J.: Variation simulation for deformable sheet metal assemblies using finite element methods. ASME J. Manuf. Sci. Eng. 119 (3), 368–374 (1997).

2. Dahlström, S.: Variation simulation of sheet metal assemblies for geometrical quality: parameter modeling and analysis. Ph.D Thesis. Göteborg, Sweden: Chalmers University of Technology (2005).
3. Marguet, B., Chevassus, N., Falgarone, H., Bourdet, P.: Geometrical behavior laws for computer aided tolerancing: AnaTole a tool for structural assembly tolerance analysis. In: 8th CIRP Seminar on Computer-Aided Tolerancing, pp. 124-131, Charlotte (2003).
4. Warmefjord, K., Söderberg, R., Lindau, B., Lindkvist, L., Lorin, S.: Joining in nonrigid variation simulation. Computer-aided Technologies, Intech (2016).
5. Wang, H., Ding, X.: Identifying sources of variation in horizontal stabilizer assembly using finite element analysis and principal component analysis. Assembly Automation 13(1), 86–96 (2013).
6. Dimensional Control System (DCS) Homepage, http://www.3dcs.com, last accessed 2017/05/21.
7. Dahlström, S., Lindkvist, L.: Variation simulation of sheet metal assemblies using the method of influence coefficients with contact modeling. Journal of Manufacturing Science and Engineering 129(3), 615-622 (2007).
8. Petukhova, M., Lupuleac, S., Shinder, Y., Smirnov, A., Yakunin, S., Bretagnol, B.: Numerical approach for airframe assembly simulation. Journal of Mathematics in Industry 4 (8), 1-12 (2014).
9. Lindau, B., Lorin, S., Lindkvist, L., Söderberg, R.: Efficient contact modeling in nonrigid variation simulation. J. Comput. Inf. Sci. Eng 16 (1), (2016).
10. Yang, D., Qu, W., Ke, Y.: Evaluation of residual clearance after pre-joining and pre-joining scheme optimization in aircraft panel assembly. Assembly Automation 36(4), 376–387 (2016).
11. Wriggers, P.: Computational contact mechanics. 2nd edn. Springer, Berlin Heidelberg (2006).
12. Pedersen, P.: A direct analysis of elastic contact using super elements. Computational Mechanics 37 (3), 221–231 (2006).
13. Lupuleac, S., Petukhova, M., Shinder, J., Smirnov, A. et al.: Software Complex for Simulation of Riveting Process: Concept and Applications. In: SAE Technical Papers, 2016-01-2090 (2016).
14. Lupuleac, S., Zaitseva, N., Petukhova, M., Shinder, J. et al.: Combination of Experimental and Computational Approaches to A320 Wing Assembly. In: SAE Technical Papers, 2017-01-2085 (2017).
15. Goldfarb, D., Idnani, A.: A numerically stable dual method for solving strictly quadratic programs. Mathematical Programming 27 (1), 1–33 (1983).
16. Powell, M. J. D.: On the quadratic programming algorithm of Goldfarb and Idnani. In: Mathematical programming essays in honor of George B. Dantzig, Part II, pp. 46–61. Vol. 25 of the Springer Mathematical Programming Studies book series (1985).
17. Stefanova, M., Yakunin, S., Petukhova, M., Lupuleac, S. & Kokkolaras, M: An interior-point method-based solver for simulation of aircraft parts riveting. Engineering Optimization 50(5), 781-796 (2017).
18. Bertsekas, D.: Projected Newton methods for optimization problems with simple constraints. SIAM Journal on Control and Optimization 20(2), 221-246 (1982).