Olga Rodionova

(Department of Applied Mathematics,

St.Petersburg Polytechnical University)

Numerical simulation of the curved surface junction

Content

Introduction

The fast numerical algorithm for calculation the gap between the two elastic bodies in the process of their deformation under the external forces applied was developed in the Laboratory of Applied Mathematics and Mechanics of Saint Petersburg State Polytechnical University. On the base of this algorithm the software complex was created. This software complex allows simulating bolted junctions of the models with different geometrical and mechanicals properties.

During the work on this project the verification of the developed complex was performed, including comparing the results of the computations with the commercial codes and physical experiments. For the models with flat junction surfaces the relative difference of these results doesn't exceed 5%, what is quite satisfactory.

The object of the present research is the application of the developed algorithm to the junction of the curved shaped bodies. We restrict ourselves with the case when contact surfaces have the constant curvature. In particular, the aims of research are, as follows:

- Find out how the computation results depend on the curvature of the junction surfaces.
- Appraise the curvatures where computational results are not satisfactory.
- Find out how this complex can be modified to be applicable to wider range of different geometries of the models.

The description of the numerical algorithm

The interaction of the rigid body and fixed obstacle

Let us consider the elastic body fixed in several points (Fig. 1). This body is to be riveted to another elastic body or rigid obstacle. The region on the body surface where the contact is possible we will call junction area. In the junction area we choose the system of computational nodes $CN=\left(cn_1, cn_2, \cdots, cn_n\right)$ (see Fig. 1), i.e. the points where the body displacements will be computed. We will call this system of computational nodes by the calculation net.

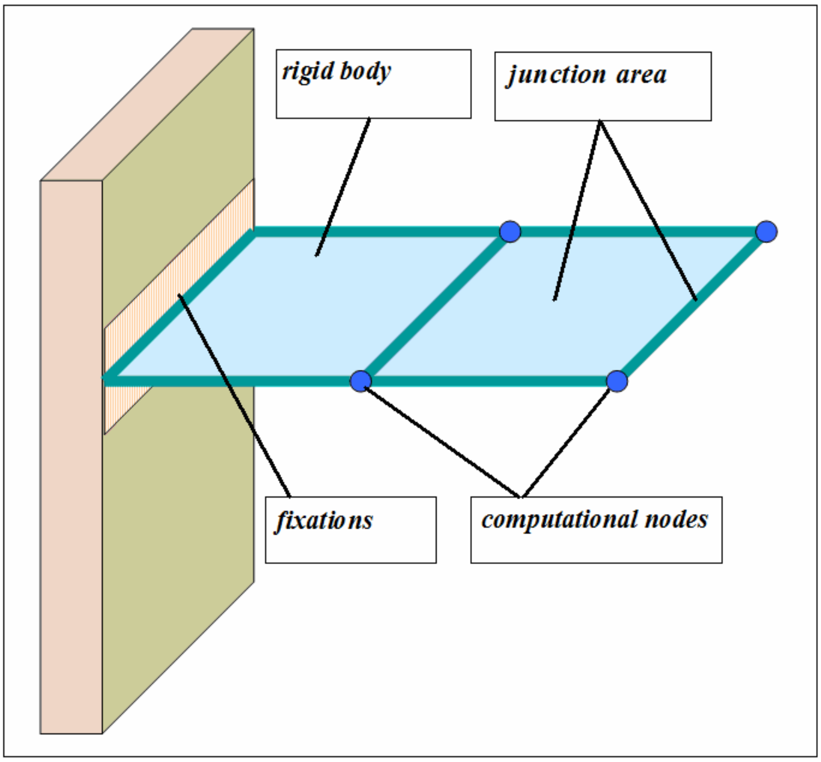

Fig. 1. An example of junction area and calculation net

Our task is to find the displacements of the computational nodes when the elastic body is under the influence of the specified system of external forces, taking into account the obstacle which restricts body displacements.

Let us consider P^T as the vector of loads applied to the computational nodes. These loads provide the displacements of computational nodes U^T. On this step the obstacle is not taken into account. Forces P and displacements U are connected with the formulas:

$$P = K \cdot U \, , \tag{1}$$

$$U = R \cdot P \, , \tag{2}$$

where K is the rigidity matrix of the system, R is the flexibility matrix, which is inverse to the rigidity matrix: $K = R^{-1}$. Flexibility matrix is used for calculating the rigidity matrix.

The rigidity matrix characterizes the ability of the body to react on the applied external loads. It depends on the body geometry, on the strength characteristics of the materials of the body, and on the positions of the computational nodes $CN = (cn_1, cn_2, \cdots, cn_n)$. So, changing any of these parameters demands the new computation of the rigidity matrix. The procedure of computation will be described below.

With the help of rigidity matrix we abstract away from the real body and replace it with the system of computational nodes, where forces and displacements are connected with the formulas (1-2). We are interested just in the displacements in the junction area, that's why the computational nodes are to be chosen only in this region.

The developed algorithm implements the approach to solution of the contact problem, based on the minimization of the quadratic functional with the linear restrictions. Due to this approach the fast speed of computations is achieved.

Let us write down the formula for potential energy of the system:

$$W(U) = \frac{1}{2} U^T \cdot K \cdot U - P^T \cdot U \,. \tag{3}$$

Here K is the rigidity matrix of the system, U is the vector of displacements of computational nodes, P is the vector of external loads.

From the mechanical law it's known that the system is in the state of equilibrium, when its potential energy reaches its minimum.

The presence of the obstacle put the extra restrictions on the body displacements. These restrictions can be written in the following way:

$$AU \leq \Delta, \tag{4}$$

where matrix A and vector Δ define respectively the direction and distance to the obstacle. More detailed description of matrix A and vector Δ will be given below.

Thus, the task of finding the computational node displacements comes to the task of finding the minimum of quadratic functional (3) with the restrictions on the displacements (4). To make the numerical algorithm fast it's necessary to provide the linearity of the restrictions.

We pay attention that the same algorithm with some modifications can be applied for simulation of interaction of the two elastic bodies (in other words the obstacle can be considered not rigid but elastic too) (see Fig. 2). In this case the rigidity matrix is calculated as the sum of flexibility matrices of two bodies, and the computed vector of displacements represents the relative displacements of two bodies.

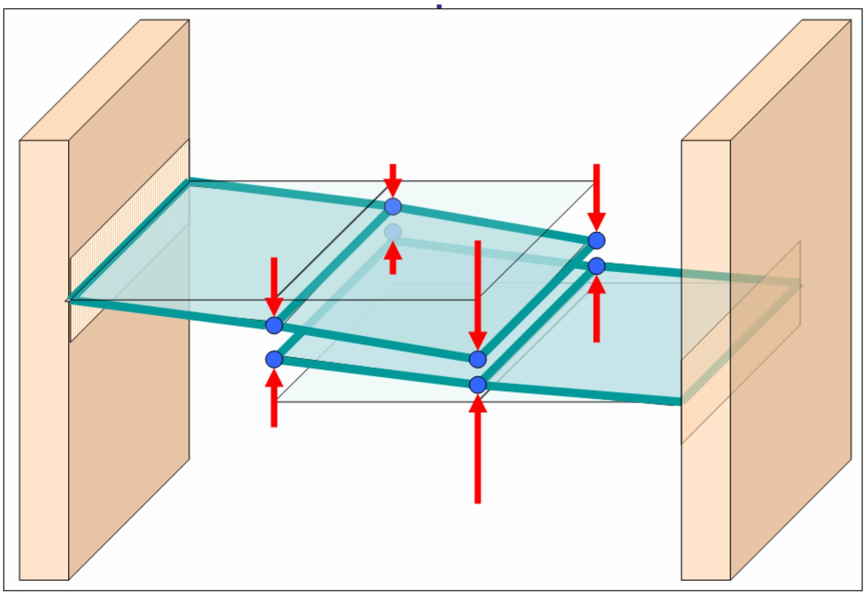

Fig. 2. An example of contact interaction of two elastic bodies

Now the realization in the algorithm of the computations of the values from formulas (3) and (4) will be explained.

Computation of the displacements in normal direction

The basic algorithm developed in the laboratory allowed to apply the forces in the computational nodes in the directions, normal to the junction surfaces, and to take into account only the normal displacements of the body.

In this algorithm $P^T = (p_1, p_2, \cdots, p_n)$ is vector of loads applied to the computational nodes, where p_i is the load, applied to the *i*-node in the direction, normal to body surface in this point. $U^T = (u_1, u_2, \cdots, u_n)$ is vector of normal displacements, provided by these loads (u_i is the displacement of the node cn_i under the influence of the forces *P* in the direction, normal to the surface in this point).

The rigidity matrix *K* is computed by the following way. In the first computational node in the direction normal to the surface the unit force of the value 1 N is applied. The values of the displacements of all computational nodes in three directions *X, Y* и *Z* under the influence of applied load are computed (the computation can be done, for example, in the finite element complexes ANSYS, NASTRAN, etc.). The obtained vectors of displacements are to be projected to the vectors of normals in the corresponding nodes. In this way, we get the first row of the flexibility matrix *R*.

Repeating the described procedure for each computational node we fill the flexibility matrix R. Thus, the flexibility matrix has the dimension $n \times n$, where n is the number of computational nodes. Inversing the flexibility matrix we obtain the required rigidity matrix $K = R^{-1}$.

Formula (4) in this case will have the following form:

$$U \le \Delta,$$

where Δ is the vector of the initial gap, in other words, the distance between the computational nodes and the obstacle.

Computation of the displacements in all directions

Now let us consider the modification of the described above algorithm, which allows computing the displacements of computational nodes in all directions *X, Y, Z*, and applying the loads in arbitrary directions.

As the vector of the loads applied to the computational nodes we consider vector $P^T = \left(p_{1x}, p_{1y}, p_{1z}, p_{2x}, p_{2y}, p_{2z}, \cdots, p_{nx}, p_{ny}, p_{nz}\right)$, where p_{ix} is the load applied to *i*-node in the direction of axis *X*, p_{iy} is the load applied to *i*-node in the direction of axis Y, and so on. Forces provide the displacements of computational nodes $U^T = \left(u_{1x}, u_{1y}, u_{1z}, u_{2x}, u_{2y}, u_{2z}, \cdots, u_{nx}, u_{ny}, u_{nz}\right)$ (u_{ix} is the displacement of the node cn_i in the direction of axis X).

In this case *K* is computed by the next way. In the first computational node in the direction of axis *X* the unit force of the value 1 N is applied. The values of the displacements of all computational nodes in three directions *X, Y* и *Z* under the influence of applied load are computed. The obtained vector becomes the first row of the flexibility matrix *R*. Then in the first computational node the unit force is applied in *Y* direction, and the values of the displacements of computational nodes in all directions are computed. In this way the second row of the flexibility matrix *R* is filled. Similarly, by applying the unit force in *Z* direction, the third row of matrix R is computed.

Repeating the described procedure for each computational node we fill the flexibility matrix R. Thus, the flexibility matrix has the dimension $3n \times 3n$, where n is the number of computational nodes.

In this case in formula (4) Δ is also the vector of initial gap, or the distance from the computational nodes to the obstacle. This distance is defined for normals to the body surface, thus, matrix A in this case is the matrix of projection to the normals, which has the dimension $n \times 3n$ and the following form:

$$\begin{pmatrix} -N_{1x} & -N_{1y} & -N_{1z} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & -N_{2x} & -N_{2y} & -N_{2z} & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & -N_{nx} & -N_{ny} & -N_{nz} \end{pmatrix},$$

where $\left(N_{ix} \quad N_{iy} \quad N_{iz}\right)$ is the vector of normal in *i*-node.

This method of setting the restriction also takes into account just the normal displacements of the body. But the new approach to the rigidity matrix computation provides the possibility to take into account the tangential displacements as well. Particularly, these displacements also can be restricted.

Investigation of the correctness of the developed numerical algorithm when applying to the simulation of the junction of surfaces with big curvature

Test model description

As the test model we took the junction of two parts with equal thickness, which contact surfaces had the constant curvature radius in one direction and infinite curvature radius in another direction (in other words, the parts of cylinder surfaces). The model of this junction is shown in the Figures 3 and 4. The ends of the parts were fixed. The values of curvature angle α (see Fig 3) were considered equal to 5^0 and 90^0.

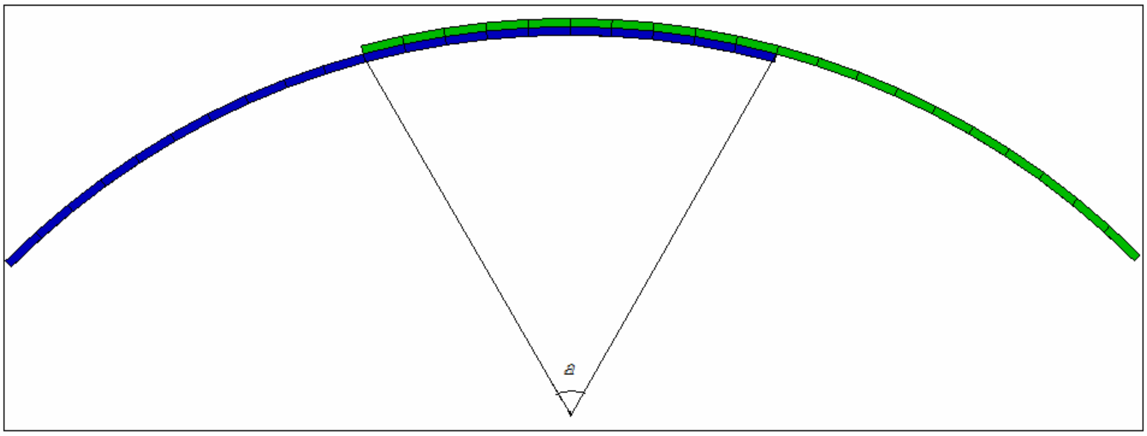

Fig. 3. Test model with curved contact surface

In the developed software complex the parts of the model are not physically separated and setting of the initial gap is provided by numerical setting its value in each computational node. In this work the features of initial gap setting are not described, but it's necessary to pay attention that the precision of the set values have the significant influence on the obtained results, as it was found out during our research.

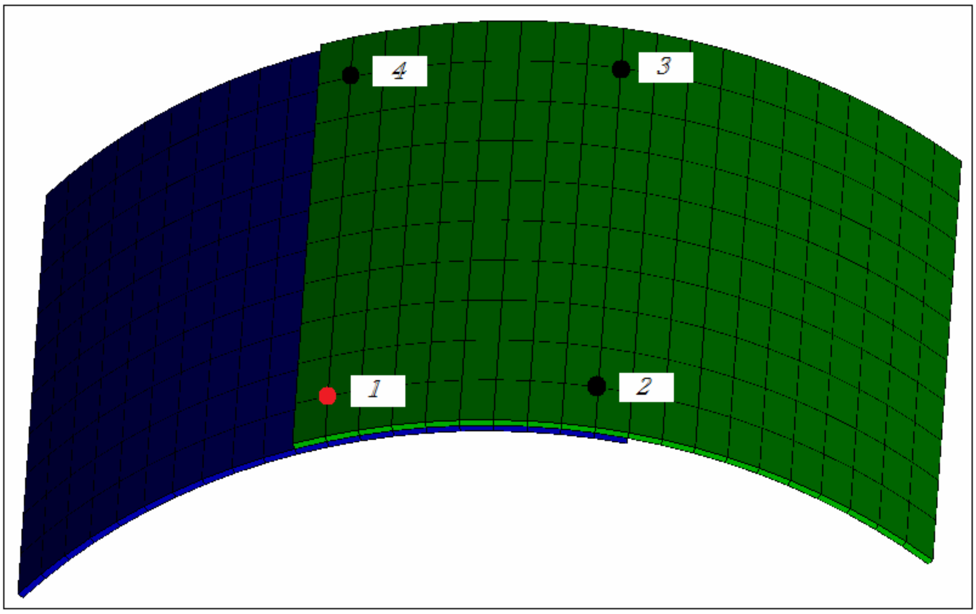

Fig. 4. An example of interaction of rigid bodies

The main idea of the research was to apply different values of the force in one computational node (in our case it was node #1, marked with the red point in the Figure 4) and to examine the gaps provided with this force in several nodes of the calculation net (nodes 1, 2, 3, 4 in the Fig. 4).

The gap is calculated as the difference between the initial gap and the value of displacements. The displacements are obtained as the result of the contact problem solution, according to the two approaches described above.

Comparing the two approaches to the displacement calculation

Under the influence of forces applied in normal direction to the contact surfaces the computational nodes can get the displacements in different directions. In the basic approach just the normal displacements are taken into account. It's expected that for the surfaces of slight curvature the tangential displacements will be insignificant even under the strong loads, while for the surfaces of large curvature they can even exceed the normal ones.

This assumption is proved with the results of the research. For the model with the angle 5^0 the ratio of the tangential displacements to the normal ones is about 2% - 5%. At the same time, for large curvature angles (e.g. for $\alpha=90^0$) the tangential displacements become comparable with the normal.

Thus, in the gap computation with our algorithm we neglect the values equal to the values taken into account. This conclusion provides the questions of admissibility of the obtained results and of the applicability of the basic algorithm to the surfaces of large curvature. To answer these questions we carried out the comparing of the basic algorithm with the second, which allows the computation of displacements in all directions and seemed to be more satisfactory for simulations for the surfaces with large curvature.

In the result of this comparing it was found out, that the difference in the gap computation with the two approaches is almost absent. The relative difference was less than 1% for the model with angle 5^0 , and (what is more important) with the angle 90^0 as well.

The explanation for this result lies in the definition of the gap as the distance between the bodies, which, according to our assumption, is computed in the direction normal to the surfaces. For this, in the basic algorithm as well as in its modification we apply the projection on the normals; the difference between the algorithms is just in the order of the operations (the displacement computation and the projection).

Thus, the research results show that the two approaches are equally applicable to the simulation the junction of curved surfaces.

Comparing of the two approaches to the gap computation

The researches described above were based on the definition of the gap as the difference between the initial gap and the values of normal displacements. With this assumption it's clear that the tangential displacements don't influence on the value of the calculated gap, what was demonstrated in the previous paragraph. But for the real physical models this assumption is not always correct.

For example, in the Figure 5, the upper part moves just in the tangential direction, so in the computations the gap value will not change. But from the picture it can be seen that the real gap increases, as the normal to the surface for the new node position has different direction than the initial one.

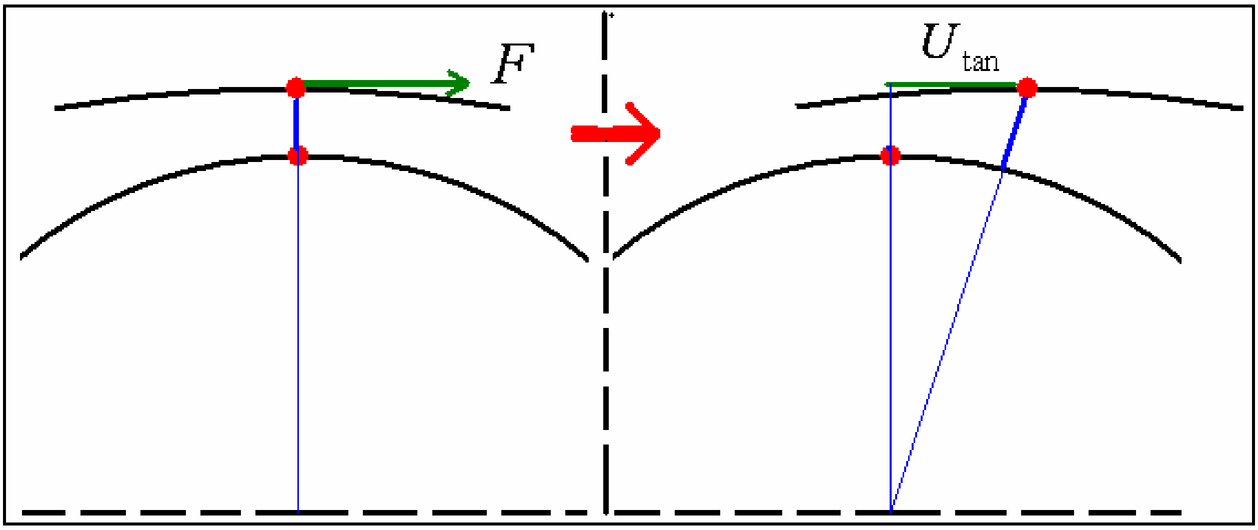

Fig. 5. Changing of the gap due to tangential displacements

Now we will try to take this fact into account for our particular models. The main idea is to find the new positions of the nodes under the influence of applied forces according to the displacements computed with the modification of the developed algorithm, to determine the normals to the contact surface in these positions, and to compute the distance between the nodes and the surface in these normal directions. For our particular models these calculations become quite easy, if we fix the lower part of the model and allow the just movement of the upper part. The formulas for the computations come from the simple geometrical considerations (see Fig. 6).

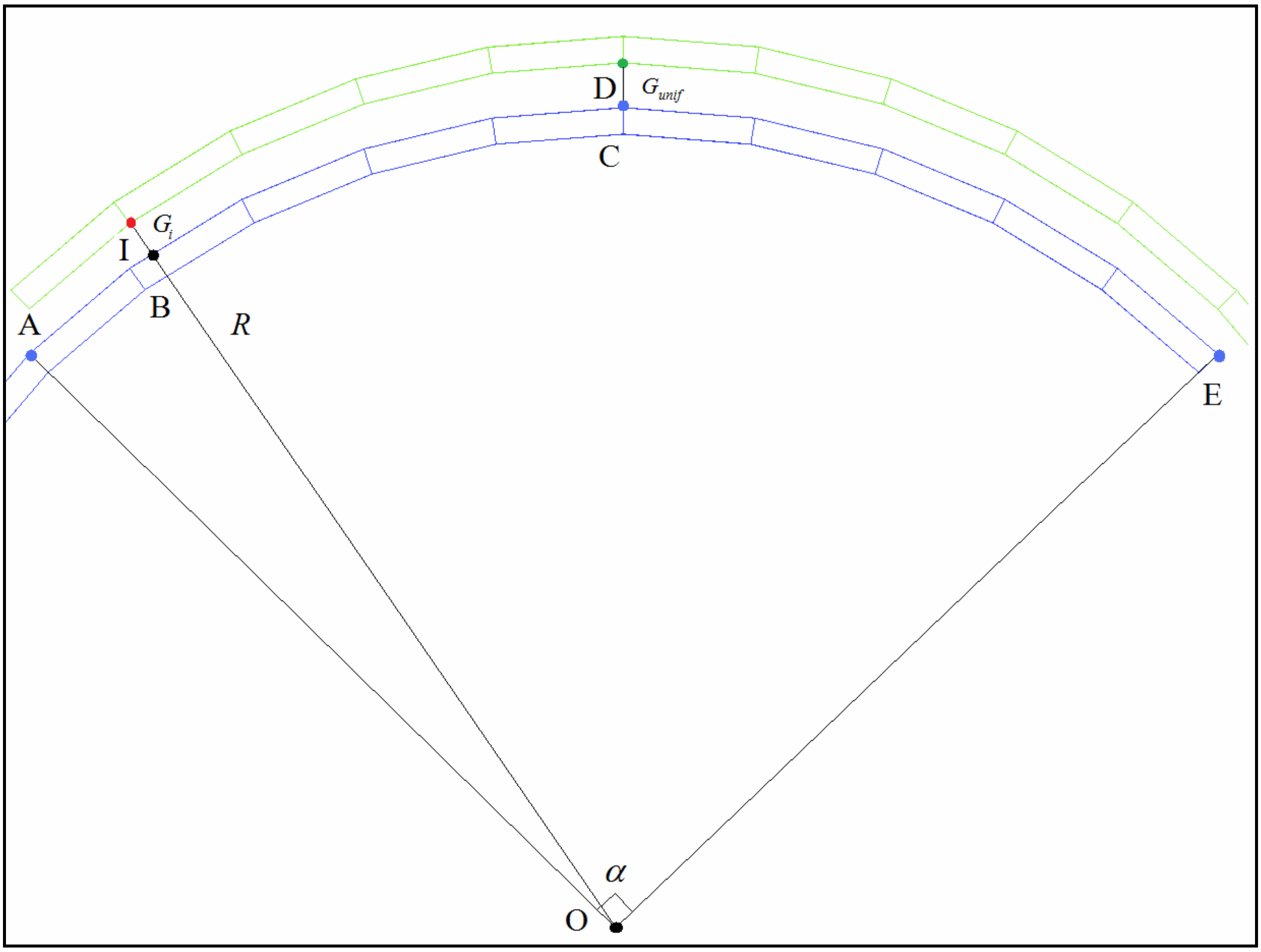

Fig. 6. Gap calculation

We choose the node *I* (marked with the red color on the picture) on the upper part and drop a perpendicular from this node on the surface of the second part. As this surface, according to the construction of the model, is an arc *AE* of a circle with the center *O* and radius *R*, the perpendicular on it is the line though the points *I* and *O*, which intersect the arc *AE* in the point *B*. The length of the segment *IB* is considered as the value of the gap G_i in the point *I*:

$$G_i = |IB| = |IO| - |BO| = |IO| - R\,, \tag{5}$$

$$R = \frac{|\cup AE|}{\alpha}. \tag{6}$$

If we consider the axis *X* to have the horizontal direction and the axis *Z* to have the vertical direction in the Figure 6, the length of the segment *IO* can be computed with the following formula:

$$|IO| = \sqrt{(x_i - x_0)^2 + (z_i - z_0)^2}\ , \tag{7}$$

where the coordinated of point *O* are determined with the formulas:

$$\begin{aligned} x_0 &= x_A + R \cdot \sin(\alpha / 2) \\ z_0 &= z_A - R \cdot \cos(\alpha / 2) \end{aligned}. \tag{8}$$

We make a note that this procedure determines the gap value quite precisely, thus, can be implemented for setting the initial gap, what was indeed done in all the researches.

Here we compare this precise method of gap calculation with standard method, that is implemented in our algorithm (where gap is calculated as the difference between the initial gap and the displacements). This comparison shows that the difference in the gap calculation with the two methods for the models of slight curvature is almost absent (e.g. for the model with angle 5^0 the relative difference doesn't exceed 0.01%). And what is more important, even for the surfaces of large curvature it is also insignificant (e.g. for the model with angle 90^0 it is in the range of 1%).

To explain this fact, we list the restrictions that are considered to be fulfilled in the models used. First of all, we use the models with realistic surfaces, where the gap variation is not big and the gap changing is smooth. Secondly, the value of the gap is small comparing with the linear sizes of the contact surfaces. Thirdly, the angle between these contact surfaces is also comparably small. The last assumption allows us to neglect the difference between the normal directions to the two contact surfaces and to consider the normals to be collinear in the corresponding computational nodes. Thus, for realistic displacement in each node the differences in the initial gap values and normal directions are negligible.

Thus, it's possible to make the conclusion that the simplifications, implemented in our algorithm for gap calculation, in practice doesn't have the influence on the results. This fact also votes for the correctness of the developed algorithm.

Conclusion

The object of the work was to find out the boundaries of the application of the developed complex and to modify the complex in order to widen these boundaries. The fulfilled research discloses that this algorithm can be applied for simulation of the junction of the models with curved contact surfaces, as well as with the flat ones.