Les Cahiers du GERAD ISSN: 0711–2440

An interior-point method-based solver for simulation of aircraft parts riveting

M. Stefanova, S. Yakunin,
M. Petukhova, S. Lupuleac, M. Kokkolaras

G–2017–62

July 2017

CITATION ORIGINALE / ORIGINAL CITATION

Stefanova, Maria; Yakunin, Sergey; Petukhova, Margarita; Lupuleac, Sergey; Kokkolaras, Micheal, An interior-point method-based solver for simulation of aircraft parts riveting, Engineering Optimization, 1–16, 2017, http://dx.doi.org/10.1080/0305215X.2017.1355367.

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

An interior-point method-based solver for simulation of aircraft parts riveting

Maria Stefanova [a]
Sergey Yakunin [b]
Margarita Petukhova [a]
Sergey Lupuleac [a]
Michael Kokkolaras [d]

[a] *Department of Applied Mathematics, Peter the Great St.Petersburg Polytechnic University, Saint-Petersburg, Russia*

[b] *Department of Materials Science and Engineering and Chemical Engineering, Charles III University of Madrid, Madrid, Spain*

[c] *GERAD and Department of Mechanical Engineering, McGill University, Montréal (Québec), Canada*

stefanova.m@list.ru
michael.kokkolaras@mcgill.ca

July 2017

Les Cahiers du GERAD
G–2017–62

Abstract: The particularities of the aircraft parts riveting process simulation necessitate the solution of a large amount of contact problems. We propose a primal-dual interior point method-based solver for solving such problems efficiently. The proposed method features a worst case polynomial complexity bound $O(\sqrt{n}\ln\frac{1}{\epsilon})$ on the number of iterations, where n is the dimension of the problem and ϵ is a threshold related to desired accuracy. In practice, the convergence is often faster than this worst case bound, which makes the method applicable to large-scale problems. The computational challenge is solving the system of linear equations because the associated matrix is ill-conditioned. To that end, we introduce a preconditioner and a strategy for determining effective initial guesses based on the physics of the problem. We compare numerical results to ones obtained using the Goldfarb-Idnani algorithm. The results demonstrate the efficiency of the proposed method.

Keywords: Interior point methods, primal-dual methods, quadratic programming, initial guesses, preconditioning

Acknowledgments: This work was inspired by a collaborative project with Airbus SAS. The authors would like to thank Mrs. Elodie Bonhomme, Mr. Jacques Bouriquet, Mr. Ludovic Gervier and Mr. Dmitry Bondarenko for their continuous support. This research was motivated by a discussion with Professor Vladimir Chepyzhov, whose involvement is also kindly appreciated.

1 Introduction

Accelerating the airframe assembly process is one of the most important challenges for aircraft original equipment manufacturers (OEMs). Long order lists for popular models and limited number of assembly lines motivate OEMs to reduce the time required for the manufacturing process. The aircraft final assembly line (FAL) includes several time-consuming operations such as joining large-size airplane sections (fuselage, wing, etc.) by means of riveting. This process is based mainly on manual labor and thus remains one of the bottlenecks of aircraft production rate.

One of the possible remedies to this problem is a global simulation of the assembly process so that numerical methods can be used to optimize its most time-consuming stages. Specialized software has been developed within a framework of cooperation between Airbus and Saint Petersburg Polytechnic University (Lupuleac et al., 2013). This tool allows engineers to solve contact problems arising when aircraft parts are being joined in order to evaluate gaps between riveted parts as well as displacements and stresses caused by the assembly (see Figure 1).

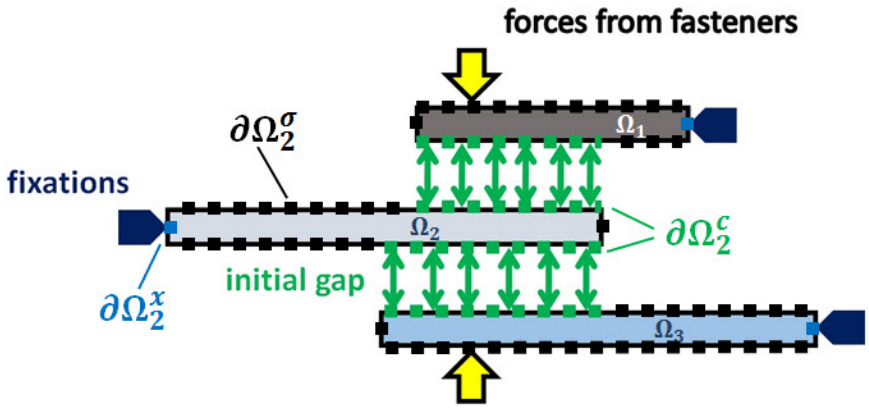

Figure 1: Schematic of aircraft parts riveting.

This contact problem exhibits specific features that can be exploited to implement specialized solution techniques (Petukhova et al., 2014):

1. The junction area is known a priori (marked by square dots in Figure 1). The simulation aims at evaluating of gap between parts during the assembly process. In this regard, only junction area displacements are of interest. Typically, the number of nodes in the junction area is much smaller than the total number of nodes, which reduces the dimension of problem.
2. The installed fasteners and rivets restrict relative tangential displacements of assembled parts in the junction area; thus tangential displacements can be omitted from consideration. This feature of the problem justifies the implementation of a node-to-node contact model.
3. Loads from fastening elements (denoted by thick arrows in Figure 1) are applied at the junction area.
4. Friction forces between assembled parts in the contact zone do not play significant role due to very low relative tangential displacements, and can thus be neglected.
5. The theory of linear elasticity is applicable.

Consider p elastic bodies, occupying bounded domains $\Omega_i \subset \mathbb{R}^N$, where $i = 1, 2, ..., p$ is the body index and $N \in \{2, 3\}$ is the dimension. The boundary of each body Ω_i is partitioned in three disjoint parts: $\partial\Omega_i^x$ is the part of boundary, where the body is constrained, $\partial\Omega_i^\sigma$ is a free surface and $\partial\Omega_i^c$ is a junction area. For each junction area, a gap $G_n^{i,j}$ is defined between the i-th and j-th bodies, and forces $\boldsymbol{F}^i$ are defined for i-th body. The contact problem for elastic bodies is formulated as the unilateral contact problem (Eck, Janusek, and Krbec, 2005). Its variation formulation is disscused in (Hlavacek et al., 1998; Kinderlehrer and

Stampacchia, 2000) and under assumptions 1–5 it can be written using the energy functional:

$$\min_{\boldsymbol{X}\in S} \frac{1}{2}\sum_{i=1}^{p}\left(\int_{\Omega_i} \boldsymbol{\epsilon}\left(\boldsymbol{X}^i\right):\ \mathbf{C}^i:\boldsymbol{\epsilon}\left(\boldsymbol{X}^i\right)d\Omega_i - \int_{\Omega_i} \boldsymbol{F}^i\cdot\boldsymbol{X}^i d\Omega_i\right), \tag{1}$$

$$S=\{\boldsymbol{X}\in[W_2^1(\Omega)]^N|\boldsymbol{X}^i=\boldsymbol{X}_0^i \text{ on } \partial\Omega_i^x,\ X_n^i-X_n^j\leq G_n^{i,j} \text{ on } \partial\Omega_i^c, i,j=1..p, i\neq j\},$$

where $\boldsymbol{X}$ are displacements, $W_2^1(\Omega)$ is Sobolev space (see Kinderlehrer and Stampacchia (2000)), $\mathbf{C}^i$ is Hooke's tensor of i-th body that is symmetric coercive, and $\boldsymbol{\epsilon}$ is a strain tensor.

After discretizing the variation formulation and using a node-to-node contact approach, the problem can be formulated as a quadratic programming problem subject to linear non-penetration conditions (Lupuleac et al., 2011):

$$\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T\mathbf{K}\boldsymbol{x}-\boldsymbol{f}^T\boldsymbol{x} \tag{2}$$

$$\text{subject to } \mathbf{A}^T\boldsymbol{x}-\boldsymbol{g}\leqslant 0,$$

where $\boldsymbol{x}\in\mathbb{R}^n$ is the vector of normal displacements at the nodes of the junction area; $\boldsymbol{f}\in\mathbb{R}^n$ is the vector of normal loads in the junction area; $\mathbf{K}\in\mathbb{R}^n\times\mathbb{R}^n$ is the reduced stiffness matrix computed using super element technique (Pedersen, 2006); $\mathbf{A}\in\mathbb{R}^n\times\mathbb{R}^m$ is a linear operator that defines the normal to the contact surface, and $\boldsymbol{g}\in\mathbb{R}^m$ is the initial gap vector in the junction area. Necessary conditions of optimality are given by Karush-Kuhn-Tucker theorem (KKT). Applied to contact problems such as Problem (2), KKT conditions have physical interpretation. Stationarity condition:

$$\mathbf{K}\boldsymbol{x}-\boldsymbol{f}+\mathbf{A}\boldsymbol{\lambda}=0, \tag{3}$$

is responsible for equality between internal and external forces. Primal feasibility condition:

$$\mathbf{A}^T\boldsymbol{x}-\boldsymbol{g}\leqslant 0, \tag{4}$$

can be regarded as non penetration condition. Dual feasibility condition:

$$\boldsymbol{\lambda}\geqslant 0, \tag{5}$$

corresponds to absence of tensile stress during contact. The complementarity condition:

$$(\boldsymbol{a_j^T}\boldsymbol{x}-g_j)\lambda_j=0. \tag{6}$$

yields that a contact force λ_j can appear only if a contact at the nodes tied by the j-th constraint occurs, which means that $y_j=-\boldsymbol{a_j^T}\boldsymbol{x}+g_j$ has to be equal to zero. On the opposite, if y_j is greater than zero there is no contact at the nodes tied by the j-th constraint, and the contact force is thus equal to zero.

When choosing an algorithm for solving Problem (2), we should keep in mind that $\mathbf{K}$ is ill-conditioned and $\mathbf{A}$ is sparse. Moreover, the technology of riveting process requires performing a series of similar computations under different initial gaps and forces in fastening elements (variable vectors $\boldsymbol{g}$ and $\boldsymbol{f}$).

The most common algorithms for solving Problem (2) are active set methods and interior point methods. Active set aim at determining constraints that become active at the optimum. An initial set of active constraints, called the working set, is assumed, which is then modified iteratively by adding or removing one constraint at each iteration. Thus, these methods are efficient only for moderately sized problems as the number of iterations grows with number of active constraints at the optimum (contact occurs almost everywhere in the junction area). To deal with ill-conditioned problems, we choose the Goldfarb–Idnani dual active set algorithm with Powell modification (Goldfarb and Idnani, 1983; Powell, 1985). It is implemented in well-known mathematical libraries such as IMSL and SciLab and proved to be robust and efficient. Its advantages include simplicity in building initial approximations and easy adaptation to problem features such as sparsity. However, its algorithm complexity is $O(mn^2)$, where m is the number of iterations and n is the problem size. As mentioned above, in case of contact over the entire junction area the algorithm complexity

tends to $O(n^3)$. It should also be mentioned that intermediate computations are memory-consuming, as it is necessary to store several fully populated matrices of size that is equal to the problem dimension.

Interior point methods (IPMs) reformulate Problem (2) to a sequence of unconstrained minimization problems:

$$\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T\mathbf{K}\boldsymbol{x} - \boldsymbol{f}^T\boldsymbol{x} - \mu\psi(\boldsymbol{x}), \tag{7}$$

where μ is a parameter characterizing the closeness to solution and $\psi(\boldsymbol{x})$ is a barrier function. In this case, the complexity of interior point method is polynomial (Nesterov and Nemirovski, 1994). The use of interior point methods is not limited by its application to the discrete form of the energy functional (2) (Tanoh, Renard and Noll, 2004). Mazorche et al. (2007) present two formulations of the contact problem as nonlinear complementarity problems as well as an implementation of an IPM for their solution. As opposed to active set methods, the complexity does not depend on the number of active constraints. On the other hand, it may be difficult to find appropriate initial approximations, which heavily depends on the problem at hand. Infeasible IPM algorithms allows starting from an infeasible point; only inequality constraints need to be satisfied.This simplifies the search process considerably. Popular starting point selection methods include Mehrotra (1992), D'Apuzzo, De Simone and di Serafino (2010),Vanderbei (1998), and Meszaros (2011). The most time-consuming step for interior point methods is the solution of a linear system of equations. A challenging numerical issue is that the condition number grows as the optimal solution is approached. To solve the linear system of equations, one can use direct or iterative approaches. Direct methods are based on decomposition techniques (Gondzio and Grothey, 2005; Meszaros, 2009) and can be efficient when parallelization is possible. A good preconditioner is required in the case of using iterative methods. To improve global convergence of interior point methods for nonlinear programming it is proposed to use filter line search strategies (Wang, Liu and Zhang, 2017; Costa and Fernandes, 2011).

In this paper, we present a feasible IPM that introduces a conjugate gradients preconditioner for solving the required system of equations and a strategy for determining effective initial approximations.

2 Peculiarities of quadratic programming problem

Considering riveting process of p parts, the vector $\boldsymbol{x}$ consists of the node displacements at each part independent of each other:

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{u_1} \\ \vdots \\ \boldsymbol{u_i} \\ \vdots \\ \boldsymbol{u_p} \end{pmatrix},$$

where $\boldsymbol{u_i}$ is the vector of normal displacements at the junction area of the i-th part. For such a vector of displacements the stiffness matrix $\mathbf{K}$ has the following form:

$$\begin{pmatrix} \mathbf{K}_1 & \dots & 0 & \dots & 0 \\ & & \vdots & & \\ 0 & \dots & \mathbf{K}_i & \dots & 0 \\ & & \vdots & & \\ 0 & \dots & 0 & \dots & \mathbf{K}_p \end{pmatrix}$$

where $\mathbf{K}_i$ is the stiffness matrix of i-th part. The stiffness matrix $\mathbf{K}$ reflects dependence of nodes on each other in the finite element model. Said another way, the element $\mathbf{K}^{lj}$ is stress that will appear in the j-th node if non-zero displacements at the l-th node are applied as boundary conditions and all other nodes are fixed in the model. If there is not dependence between the l-th and j-th nodes, the element $\mathbf{K}^{lj}$ will be 0. Assembling parts are independent on each other before fastening, however nodes inside one part are dependent. These properties of considered problem lead to block-diagonal structure of $\mathbf{K}$. While $\mathbf{K}$ is

symmetric and positive-definite, it is ill-conditioned, each block $\mathbf{K}_i$ is fully populated. Recalling that the Problem (2) is posed for unknown normal displacements only in the junction area, the average number of unknowns for realistic problems is between 1000 and 10000. The matrix $\mathbf{A}$ has a simple and sparse structure: each row contains one or two non-zero elements due to the fact that node-to-node contact is considered.

The dual of Problem (2) can also be formulated as a quadratic program:

$$\max_{\boldsymbol{\lambda}} \quad -\frac{1}{2}\boldsymbol{\lambda}^T\mathbf{Q}\boldsymbol{\lambda} + \boldsymbol{p}^T\boldsymbol{\lambda} + s \tag{8}$$

$$\text{subject to } \boldsymbol{\lambda} \geqslant 0,$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of Lagrange multipliers corresponding to contact forces in a contact area, $\mathbf{Q} = \mathbf{A^T K^{-1} A}$ is a symmetric positive-definite matrix (in our case the columns of matrix $\mathbf{A}$ are linearly independent), $\boldsymbol{p} = \mathbf{A}^T\mathbf{K}^{-1}\boldsymbol{f} - \boldsymbol{g}$ and $s = -\frac{1}{2}\boldsymbol{f}^T\mathbf{K}^{-1}\boldsymbol{f}$.

In summary, the particular features of Problems (2) and (8):

1. $\mathbf{K}$ is symmetric, positive-definite, ill-conditioned with block diagonal structure. Each block is fully-populated;
2. $\mathbf{Q}$ is symmetric, positive-definite, ill-conditioned, fully-populated;
3. $\mathbf{A}$ is sparse;
4. Usually during the simulation and optimization of aircraft assembly process, matrices $\mathbf{K}$, $\mathbf{A}$ and $\mathbf{Q}$ are the same for many computations with different $\boldsymbol{f}$ and $\boldsymbol{g}$, $\boldsymbol{p}$ and s.

3 Goldfarb–Idnani algorithm

It is sufficient to solve the system given by Equations (3)–(6) to find a solution to Problem (2). Equations (3)–(6) form a nonlinear system that is very hard to solve directly; most methods replace this system by a sequence of simpler problems. This can be achieved by either slackening the primal feasibility condition, leading to active set methods or slackening the complementarity condition leading to interior point methods.

The active set method we consider in this work is the one implemented the Goldfarb–Idnani algorithm with Powell modification (Powell, 1985), described in Algorithm 1. It can be observed that stationarity and complementarity conditions are satisfied automatically at step 6 and dual complementarity condition will be satisfied at the inner loop. The algorithm terminates when the primal feasibility condition is satisfied. Thus,

Algorithm 1 Goldfarb-Idnani algorithm

1: Denote the entire set of constraints $\boldsymbol{Y} = \{y_j | j = 1, ..., m\}$, where $y_j = \boldsymbol{a_j^T x} - g_j$; if $y_j > 0$ then $y^+ = y_j$ and if $y_j < 0$ then $y^- = y_j$;
2: Initialize $\boldsymbol{x_0} = \mathbf{K}^{-1}\boldsymbol{f}$ as an unconstraint optimum, $Y_0 = \emptyset$, $k = 0$, $\boldsymbol{\lambda_0} = 0$;
3: **while** violated constraint $y^+(\boldsymbol{x_k})$ exists **do**
4: $Y_{k+1} = Y_k \cup y^+$
5: **repeat**
6: Find $\boldsymbol{x_{k+1}}$, $\boldsymbol{\lambda_{k+1}}$ as a solution of optimization problem:

$$\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T\mathbf{K}\boldsymbol{x} - \boldsymbol{f}^T\boldsymbol{x} \tag{9}$$

$$\text{subject to } y(\boldsymbol{x}) = 0, \forall y \in Y_{k+1}$$

7: **if** $\lambda^- < 0$ corresponding to $y^- \in Y_{k+1}$ exists **then**
8: $Y_{k+1} \leftarrow Y_{k+1} \setminus \{y^-\}$
9: **end if**
10: **until** negative Lagrange multiplier $\lambda^-_{k+1} < 0$ exists
11: $k \leftarrow k + 1$
12: **end while**
13: **return** $\boldsymbol{x_k}$

all KKT conditions are satisfied, which means that a stationary point has been found. Since the problem is convex, this KKT point is a global minimizer.

Step 6 of the algorithm is the most challenging. Goldfarb and Idnani (1983) proposed a way of updating $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ using solution of (9) for the set of active constraints C_k from the previous iteration. The updating is based on Cholesky factorization of $\mathbf{K}$ and its inversion. The updating procedure requires $O(n^2)$ operations and $O(n^2)$ additional memory for matrices storage. The feature 4 of Problem (2) allows performing factorization of $\mathbf{K}$ with $O(n^3)$ operations once for several computations with different external forces $\boldsymbol{f}$ and gaps $\boldsymbol{g}$. The theorem affirming that Algorithm 1 terminates after a finite number of steps was proved by Goldfarb and Idnani (1983). The number of steps can be estimated by the number of constraints n_{act} that are active at the optimum, i.e., the Goldfarb-Idnani algorithm finds a solution after $O(n_{act}n^2)$ computational operations.

Several modifications that decrease the working time of the Goldfarb-Idnani algorithm for a considered class of problems were developed in previous work by Petukhova et al. (2014). We will use modified Goldfrab-Idnani-Powell algorithm to compare results.

4 Primal-dual interior point method

Interior point methods date back to the Karmarkar algorithm (1984). It is a polynomial-time algorithm for linear programming problems, and a competitor to the simplex method. Equivalence of the Karmarkar algorithm to a barrier Newton method was shown by Gill et al. (1986). We use this result to describe ideas of interior point methods for quadratic programming problems. In barrier method, the problem with constraints is replaced by a sequence of unconstrained optimization problems:

$$\min_{\boldsymbol{x}} F_\mu(\boldsymbol{x}) = \min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T\mathbf{K}\boldsymbol{x} - \boldsymbol{f}^T\boldsymbol{x} - \mu\sum_{j=1}^{m}\ln(-\boldsymbol{a_j^T}\boldsymbol{x} + g_j), \tag{10}$$

where μ is the duality gap. The solution of Problem (10) converges to the solution of Problem (2) when μ goes to zero. The Newton method is used for solving the unconstrained problems of the sequence.

The main challenges of the barrier methods include choosing a type of barrier function and a way of updating the duality gap. We use a logarithmic barrier function because of a very attractive property: the gradient of the function $F_\mu(\boldsymbol{x})$ resembles the KKT conditions of Problem (2). This feature inspires the development of the most efficient interior point method: a primal-dual method that can be interpreted as a simultaneous application of the barrier Newton method to the primal and dual problems. Information about dual variables gives to the primal-dual method a key for updating the duality gap. It decreases dramatically the number of unconstrained problems that have to be solved to obtain the solution of Problem (2) for a given accuracy.

The primal-dual method solves a sequence of nonlinear systems:

$$S_\mu(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda}) = \begin{pmatrix} \mathbf{K}\boldsymbol{x} - \boldsymbol{f} + \mathbf{A}\boldsymbol{\lambda} \\ \mathbf{A}^T\boldsymbol{x} - \boldsymbol{g} + \boldsymbol{y} \\ \boldsymbol{\Lambda}\mathbf{Y}\mathbf{e} - \mu\mathbf{e} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{y} \geqslant 0, \boldsymbol{\lambda} \geqslant 0, \tag{11}$$

where $\boldsymbol{\Lambda}$ and $\mathbf{Y}$ are diagonal matrices with $\boldsymbol{\lambda}$ and $\boldsymbol{y}$ on their diagonals and $\boldsymbol{e}$ is a vector of ones. The first and second equations of System (11), together with the inequalities, resemble the stationarity and primal-dual feasibility of KKT conditions. The third equation is a violated complementarity condition and it tends to be satisfied when the duality gap tends to zero. The primal-dual algorithm is described in Algorithm 2.

Algorithm 2 Basic iteration of primal-dual algorithm

Input: centering parameter $\sigma \in [0;1]$, duality gap μ, point $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda})$, boundary closeness parameter $\theta = (0;1)$;
Output: updated duality gap μ, updated point $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda})$;

1: Calculate direction using Newton method:

$$\nabla S_\mu(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) \begin{pmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta y} \\ \boldsymbol{\Delta \lambda} \end{pmatrix} = -S_\mu(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) \Rightarrow$$

$$\begin{pmatrix} \mathbf{K} & \mathbf{0} & \mathbf{A} \\ \mathbf{A^T} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda} & \mathbf{Y} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta y} \\ \boldsymbol{\Delta \lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{s_1} \\ \boldsymbol{s_2} \\ \boldsymbol{s_3} \end{pmatrix}, \tag{12}$$

where $\mathbf{I}$ is an identity matrix, $\boldsymbol{s_1} = -(\mathbf{K}\boldsymbol{x} - \boldsymbol{f} + \mathbf{A}\boldsymbol{\lambda})$, $\boldsymbol{s_2} = -(\mathbf{A}^T\boldsymbol{x} - \boldsymbol{g} + \boldsymbol{y})$, $\boldsymbol{s_3} = -(\mathbf{\Lambda Y e} - \sigma\mu\mathbf{e})$;

2: Calculate step length which holds feasibility:

$$\alpha = \min\{\theta \min_{\Delta y_i < 0} \left(-\frac{y_i}{\Delta y_i}\right), \theta \min_{\Delta \lambda_i < 0} \left(-\frac{\lambda_i}{\Delta \lambda_i}\right), 1\};$$

3: Update variables:

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{\lambda} \end{pmatrix} \leftarrow \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{\lambda} \end{pmatrix} + \alpha \begin{pmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta y} \\ \boldsymbol{\Delta \lambda} \end{pmatrix};$$

4: Update duality gap:

$$\mu \leftarrow \frac{\boldsymbol{\lambda^T y}}{m}.$$

5 Application of IPM to special class of contact problems

5.1 Complexity analysis

Definition 1 *We call the point* $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ *feasible (or lying in a feasible set* F_0*) if the following conditions are satisfied:*

$$\mathbf{Kx} - \mathbf{f} + \mathbf{A\lambda} = \mathbf{0},\ \mathbf{A^T x} - \mathbf{g} + \mathbf{y} = \mathbf{0},\ \boldsymbol{\lambda} \geqslant \mathbf{0},\ \mathbf{y} \geqslant \mathbf{0}. \tag{13}$$

In other words, the point is feasible if all KKT conditions except complementarity are satisfied.

Definition 2 *Central path or central trajectory is a set of feasible points such that a slack complementarity condition* $\lambda_i y_i = \mu \geqslant 0$ *holds for a set of solutions of Problem (11) for all values of* $\mu \geqslant 0$.

If a point is feasible and belongs to the central path neighborhood $N_2(k, \mu) = \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) \in F_0 | \parallel \mathbf{\Lambda Y e} - \mu\mathbf{e} \parallel_{\mathbf{2}} \leqslant k\mu\}$ with $k \in (0, 1)$ at every iteration of a primal-dual algorithm, then this method is a path-following method and the following theorem can be proven (Gondzio, 2012):

Theorem 1 *If an initial point* $(\boldsymbol{x_0}, \boldsymbol{y_0}, \boldsymbol{\lambda_0})$ *is in a central path neighborhood* $N_2(0.1, \mu_0)$ *and satisfies for given* $\epsilon_\mu > 0$ *the relation* $\mu_0 = \frac{\boldsymbol{\lambda_0^T y_0}}{m}$*, where* $\mu_0 < \frac{1}{\epsilon_\mu^i}$ *for some positive constant* i*, then there exists an index* $L = O(\sqrt{m}\ln(\frac{1}{\epsilon_\mu}))$ *such that* $\mu_l \leqslant \epsilon_\mu, \forall l \geqslant L$*, where* μ_l *is duality gap at the* l*-th iteration.*

We will use this complexity result for theoretical comparison of the primal-dual method and Goldfarb–Idnani despite its worst-case complexity. We describe next an implementation of the primal-dual IPM that yields better practical complexity results.

5.2 Predictor-corrector scheme

The main disadvantage of path-following methods is the short step necessary for holding closeness to a central path, which leads to a high number of necessary steps. Choosing a central parameter close to zero decreases the number of required iterations but may prevent convergence. To reduce the number of steps while guaranteeing convergence, a predictor-corrector scheme is used. The most efficient is the Mehrotra scheme (1992) with adaptive choice of centering parameter (see Algorithm 3). We use this algorithm as a foundation for development of our solver.

Algorithm 3 Primal-dual algorithm (predictor-corrector scheme)

1: Choose accuracy ϵ_μ and boundary closeness parameter $\theta \in (0, 1)$;
2: Choose initial point $(\boldsymbol{x_0}, \boldsymbol{y_0}, \boldsymbol{\lambda_0})$;
3: $\mu_0 = \frac{\boldsymbol{\lambda_0^T y_0}}{m}$;
4: **while** $\mu_k > \epsilon_\mu$ **do**
5: Compute $\mu^{(p)}$, $(\boldsymbol{x}^{(p)}, \boldsymbol{y}^{(p)}, \boldsymbol{\lambda}^{(p)})$ by Algorithm 2 with input $\sigma = 0$, μ_k, $(\boldsymbol{x_k}, \boldsymbol{y_k}, \boldsymbol{\lambda_k})$, θ;
6: Compute μ_{k+1}, $(\boldsymbol{x_{k+1}}, \boldsymbol{y_{k+1}}, \boldsymbol{\lambda_{k+1}})$ by Algorithm 2 with input $\sigma = (\frac{\mu^{(p)}}{\mu_k})^3$, $\mu^{(p)}$, $(\boldsymbol{x}^{(p)}, \boldsymbol{y}^{(p)}, \boldsymbol{\lambda}^{(p)})$, θ;
7: $k \leftarrow k + 1$
8: **end while**
9: **return** $\boldsymbol{x}$

5.3 Linear system solver

Consider the computationally-intense stage of the primal-dual algorithm, i.e., the solution of the linear System (12) necessary to find the Newton step direction. System (12) can be rewritten in simpler form:

$$\begin{pmatrix} \mathbf{H} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta y} \\ \boldsymbol{\Delta \lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{s_1} - \mathbf{A Y^{-1} s_3} + \mathbf{A Y^{-1} \Lambda s_2} \\ -\mathbf{A^T \Delta x} - \mathbf{s_2} \\ \boldsymbol{Y^{-1}(s_3 - \Lambda \Delta y)} \end{pmatrix}, \tag{14}$$

where $\mathbf{H} = \mathbf{K} + \mathbf{A Y^{-1} \Lambda A^T}$ is symmetric and positive definite. The only way to solve linear system faster than in $O(n^3)$ operations is by means of iterative methods. Because of the properties of $\mathbf{H}$, we use a conjugate gradient method.

Consider the case where $\mathbf{A}$ is the identity matrix. $\mathbf{H}$ can be partitioned as $\mathbf{H} = \mathbf{K} + \mathbf{D}$, where $\mathbf{D} = \mathbf{Y^{-1}\Lambda}$ is a diagonal matrix. $\mathbf{H}$ becomes ill-conditioned because there are very large elements on the diagonal of $\mathbf{D}$ (corresponding to close to zero values of $\boldsymbol{y}$) as well as small elements (corresponding to small values of $\boldsymbol{\lambda}$). Iterative methods are not efficient for ill-conditioned problems without using a preconditioners. The most common approach to this type of problems consists in building a splitting preconditioner that divides high and low values of $\mathbf{D}$ to independent blocks (Oliveira and Sorensen, 2005). We use a preconditioner based on Cholesky factorization of $\mathbf{K}$, which is quite efficient and easier to build than a splitting preconditioner. Solve a preconditioned system $\widetilde{\mathbf{H}}\widetilde{\boldsymbol{\Delta x}} = \widetilde{\mathbf{f_1}}$ instead of $\mathbf{H\Delta x} = \mathbf{f_1}$, where $\widetilde{\mathbf{H}} = \mathbf{V^{-T} H V^{-1}}$, $\widetilde{\boldsymbol{\Delta x}} = \mathbf{V\Delta x}$, $\widetilde{\boldsymbol{f_1}} = \mathbf{V^{-T} f_1}$ and $\mathbf{V}$ is the preconditioner. One way is to use just a Cholesky factorization of $\mathbf{K} = \mathbf{U^T U}$ as a preconditioner that can be calculated once in the stage of data preprocessing. However, $\mathbf{U}$ does not contain any information about the diagonal $\mathbf{D}$ that has a strong influence on the condition number of $\mathbf{H}$. To take this information into account we use:

$$\mathbf{V} = \mathbf{U} + \sqrt{\mathbf{D}}. \tag{15}$$

$\mathbf{V}$ is upper triangular and any system with such a matrix can be solved easily.

Table 1 reports the number of conjugate gradient method iterations. Usage of the preconditioner $\mathbf{V}$ decreases the number of iterations needed for solving the linear system to a few instead of hundreds or thousands. This example shows that $\mathbf{V}$ is an appropriate and effective preconditioner, and that its usage can decrease dramatically the number of computational operations.

If $\mathbf{A}$ is not the identity matrix, we solve the dual problem. The matrix of the constraints for dual problem is always equal to the identity matrix, therefore, the technique described above is applicable.

5.4 Initial guess choice

The working time of a primal-dual interior-point method depends on the initial guess. It can be quite complicated to find a feasible initial guess. Luckily, in the contact problems under consideration, it can be made straightforwardly using physical interpretation.The initial guess should satisfy Conditions (13). This means that bodies are in equilibrium, the gap $\boldsymbol{y}$ between them is not negative, and contact forces $\boldsymbol{\lambda}$ are positive. However it may happen that both gap and contact force are positive at the same node. Contact

Table 1: CG iterations for a matrix size of 4386×4386 ($cond(\mathbf{K}) = \mathbf{2.63 * 10^6}$).

Number of iteration	$\mathbf{V = I}$ predictor	$\mathbf{V = I}$ corrector	$\mathbf{V = U + \sqrt{D}}$ predictor	$\mathbf{V = U + \sqrt{D}}$ corrector
1	18395	19636	20	22
2	32527	33338	17	18
3	28273	28405	17	18
4	20770	21127	17	18
5	15354	15449	17	18
6	8234	8250	17	17
7	3919	3893	17	17
8	1610	1562	17	17
9	579	545	17	16
10	183	168	16	15
11	118	107	15	14
12	89	81	14	14
13	71	64	13	13
14	58	52	13	12
15	49	44	12	11
16	42	37	11	11
17	36	32	11	10
18	29	26	11	10
19	25	21	11	9
20	30	25	11	10
21	54	43	14	11
22	102	96	15	13
23	188	165	17	14
24	318	226	17	13
25	370	233	15	11
26	546	89	14	13
27	595	211	14	13
28	532	109	14	13
29	940	242	14	13
30	1962	549	17	10
31	2540	407	19	12
32	3607	1850	17	10
33	6092	3169	23	9
34	6406	3690	23	9
35	18079	3189		
36	26399	3987		

forces $\boldsymbol{\lambda}$, if they are not in equilibrium with external forces $\boldsymbol{f}$, lead to increasing the gap $\boldsymbol{y}$ between parts. In fact, high contact forces can in the considered types of problems produce positive gaps between parts, and can be used for finding initial guesses. This approach is presented for primal (Algorithm 4) and dual problems (Algorithm 5). Note that both algorithms are equivalent because $\mathbf{Q\lambda - p = A^T K^{-1} A\lambda - A^T K^{-1} f + g}$.

Algorithm 4 Algorithm for determining initial guess for primal problem

Choose starting value for contact force $\boldsymbol{\lambda} > 0$ such that $\mathbf{A^T K^{-1} A\lambda > 0}$ and multiplier $mult > 1$;
while *true* **do**
  Compute displacements $\boldsymbol{x}$ from system of equations: $\mathbf{Kx = f - A\lambda}$;
  Compute gap: $\boldsymbol{y} = -\mathbf{A^T x + g}$;
  if $\boldsymbol{y} \geqslant 0$ **then**
    break
  end if
  $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} * mult$;
end while
return initial guess $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda})$

Algorithm 5 Algorithm for determining initial guess for dual problem

1: Choose starting value for contact force $\boldsymbol{\lambda} > 0$ such that $\mathbf{Q}\boldsymbol{\lambda} > \mathbf{0}$ and multiplier $mult > 1$;
2: **while** *true* **do**
3: Compute gap: $\boldsymbol{y} = \mathbf{Q}\boldsymbol{\lambda} - \mathbf{p}$;
4: **if** $\boldsymbol{y} \geqslant 0$ **then**
5: **break**
6: **end if**
7: $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} * mult$;
8: **end while**
9: Set $\boldsymbol{w} = \boldsymbol{\lambda}$;
10: **return** initial guess $(\boldsymbol{\lambda}, \boldsymbol{w}, \boldsymbol{y})$

Algorithms terminate when $\boldsymbol{\lambda}$ satisfies:

$$\boldsymbol{\lambda} > 0, \mathbf{Q}\boldsymbol{\lambda} > \mathbf{0}. \tag{16}$$

Indeed, increasing of contact force $\boldsymbol{\lambda}$ leads to increasing of gap between parts $\boldsymbol{y}$ (steps 3–4 in Algorithm 4 and step 3 in Algorithm 5). For such matrices $\mathbf{Q} = \begin{pmatrix} \boldsymbol{q_1^T} \\ \dots \\ \boldsymbol{q_m^T} \end{pmatrix}$ that have positive sum of elements for each row it is sufficient to pose $\boldsymbol{\lambda} = \lambda \boldsymbol{e}$ as starting value to satisfy (16). In general, if sum $\boldsymbol{q_i^T}\boldsymbol{\lambda} < 0$ for some i then it can be corrected by modification of λ_i: $\lambda_i \leftarrow -\frac{\sum_{j=1, j\neq i}^{m} q_{ij}\lambda_j}{q_{ii}} + \epsilon$, where ϵ is small value. This is possible due to positiveness of all diagonal elements q_{ii}. In practice, this process converges to $\boldsymbol{\lambda}$ that satisfies (16) and does not require many corrections.

Table 2 presents working time and number of IPM iterations for the problem with a matrix $\mathbf{K}$ of size 4386 and for different sets of external loads $\boldsymbol{f}$. The initial point obtained using Algorithm 5 was compared to starting points that are obtained using strategies STP1 and STP2 from (D'Apuzzo, De Simone and di Serafino, 2010). Approaches STP1 and STP2 give infeasible starting point that belongs to the central path, however infeasibility is kept small. Algorithm 5 requires fewer iterations relative to other approaches.

Table 2: Comparison of different starting points for a matrix size of 4386×4386. (iter. - number of IPM iterations, T - working time in sec.).

Number of active constraints	STP1		STP2		Algorithm 5	
	iter.	T	iter.	T	iter.	T
52	24	45.117	26	50.036	25	47.323
107	24	73.511	24	72.134	25	74.857
418	24	119.645	24	121.226	24	121.673
795	24	109.433	22	104.378	22	101.324
863	23	98.536	22	101.202	21	89.793
1074	23	88.546	22	89.891	21	82.979
1128	24	85.498	21	81.601	21	78.217
1347	25	75.866	21	71.506	20	64.038
1471	24	69.343	21	66.267	21	64.558
1592	24	64.895	24	71.203	21	59.902

6 Results

We first discuss a few general properties of the developed primal-dual method and previously used Goldfarb-Idnani algorithm (Table 3). Data preparation time for both algorithms has the same behavior for large n, however, the Goldfarb-Idnani algorithm does not only require a Cholesky factorizations but also matrix inversions. The main advantage of the primal-dual method is its lower memory requirement, which makes it attractive for solving large-scale problems. We compare working time as a function of the number of active constraints at the optimum for contact problems arising in wing-to-fuselage assembly simulations (Table 4) that are described below.

Table 3: Comparison of algorithms.

	Primal-Dual	Goldfarb-Idnani
Time for data preparation	$O(n^3)$	$O(n^3)$
Working time	$O(n^{2,5} n_{it} \ln(\frac{1}{\epsilon_\mu}))$, n_{it} is number of conjugate gradient method iterations	$O(n_{act} n^2)$, $0 \leqslant n_{act} \leqslant m$
Required memory	$O(\sum_{i=1}^{p} n_i^2)$, p is number of parts, n_i number of nodes for i-th part (Cholesky decomposition matrix)	$O(n^2) = O((\sum_{i=1}^{p} n_i)^2)$ (Cholesky decomposition matrix +its inversion+memory required for step 6 in Algorithm 1)

Table 4: Test problems.

N^o	Number of nodes	Number of constraints	Condition number of K
1	4386	2193	$2.63 * 10^6$
2	6112	3056	$1.11 * 10^7$
3	11308	5654	$1.77 * 10^7$

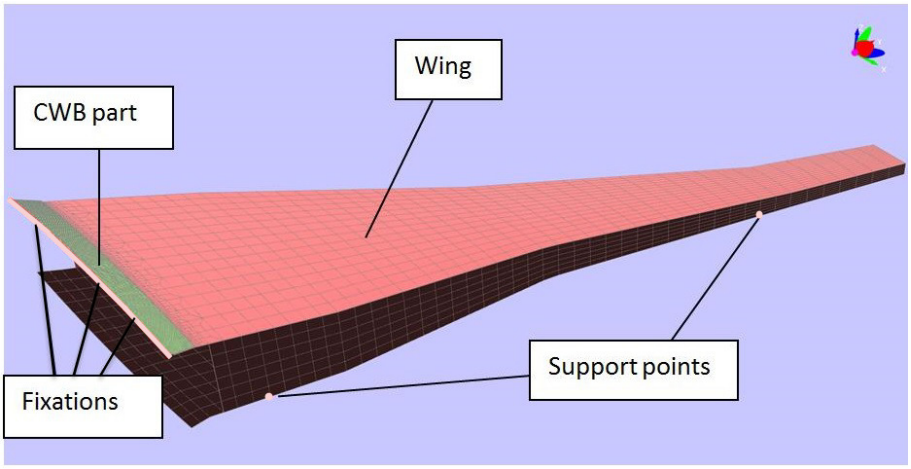

Figure 2: Finite element model of wing-to-fuselage junction.

Let us consider a finite element model of upper wing-to-fuselage junction, shown in Figure 2. There are two parts in the assembly: the first is the wing (pink) and second (green) is the fragment of Center Wing Box (CWB). The width of the junction area along the longest direction is about 3–5 m. The wing is supported in four points at lower surface and CWB fragment is fixed along the left edge. The only difference between problems 1 and 3 is the density of nodes in junction area (correspondingly 4386 and 11308, see Figure 3). The main task of the simulation is to provide the distribution of gap between parts during assembly. The example of such distribution (problem 3 with constant 0.6 mm initial gap and three installed fasteners) is depicted in Figure 4. The test problem 2 (6112 nodes) is similar to the other two problems, but it is based on a finite element model of wing-to-fuselage junction of real commercial aircraft that can not be presented for proprietary reasons.

Comparison of working time for modified Goldfarb–Idnani–Powell algorithm (GI) and primal-dual method (IPM) with the proposed initial gap and preconditioner for primal and dual problem is shown in Figures 5–7. Working time in the plots does not include time of data preparation, i.e. Cholesky factorization, inversion of Cholesky factorization matrix and building of matrix $\mathbf{Q}$ for dual problem. Due to specificity of application it is necessary to solve many problems with different initial gaps $\boldsymbol{g}$ and loads $\boldsymbol{f}$ for the same stiffness matrix $\mathbf{K}$. That fact allows picking out the step of data preparation. The initial points of the algorithms are different. The unconstrained minimum was used as starting point for the Goldfarb-Idnani algorithm (Goldfarb and Idnani, 1983). It is important to note that the algorithm takes advantage of such starting point. Algorithm 5 was used to find the starting point for IPM. Difference between obtained displacements by both the two

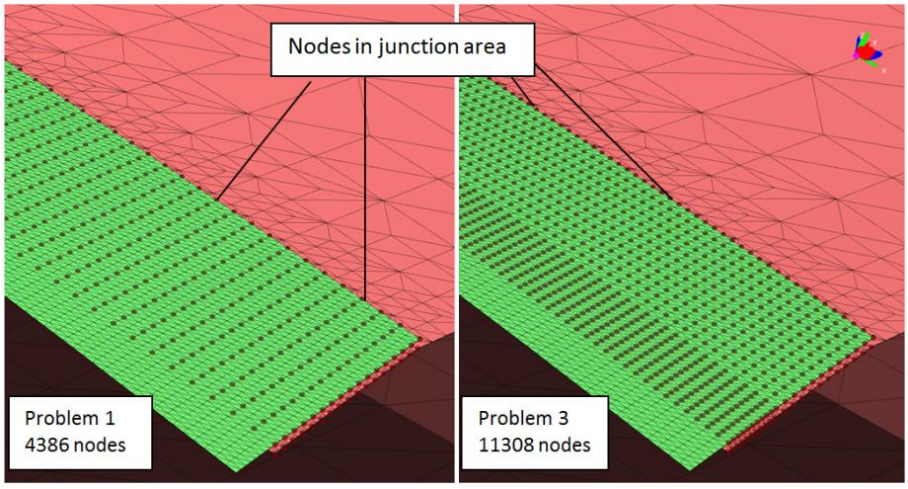

Figure 3: Grids of nodes in junction area for problems 1 and 3 (fragment).

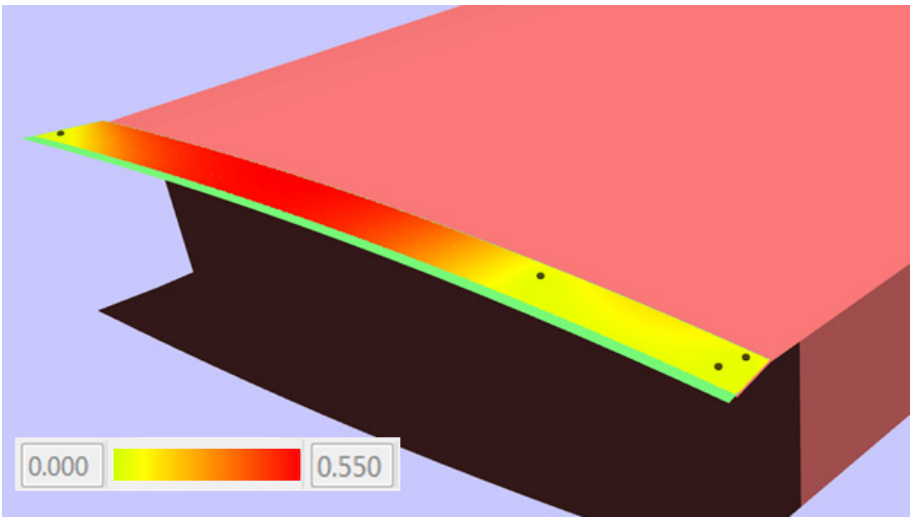

Figure 4: Distribution of gap (in mm) between parts with installed fasteners.

algorithms does not exceed 10e-7 mm. All computations were done on an Intel Core i5 CPU 3.30 GHz computer with 16 GB of RAM.

It can be observed that the Goldfarb-Idnani algorithm for the primal problem takes longer for larger numbers of active constraints in the primal problem or for smaller numbers of active constraints for the dual problem. This can be attributed to the structure of the Goldfarb–Idnani algorithm and contact problem. Inequality constraints in primal problem represent non-penetration conditions. Those constraints become active when the gap between parts is zero. The larger contact zone is, the bigger the number of active constraints is at the optimum for the primal problem. In the same time contact forces are equal to zero if the gap is not zero. Therefore the larger contact zone is the smaller the number of active constraints is at the optimum for the dual problem. The Goldfarb–Idnani algorithm looks over all violated constraints starting from the unconstrained optimum, increasing or decreasing the working set of active constraints at each step of the algorithm that is why the computational time depends on the number of active constraints at the optimum. On the contrary, the working time of the primal-dual algorithm is not influenced by the number of active constraints. The results of working time show slight influence on number of active constraints coming from quality of initial point choosing for different configuration of loads.

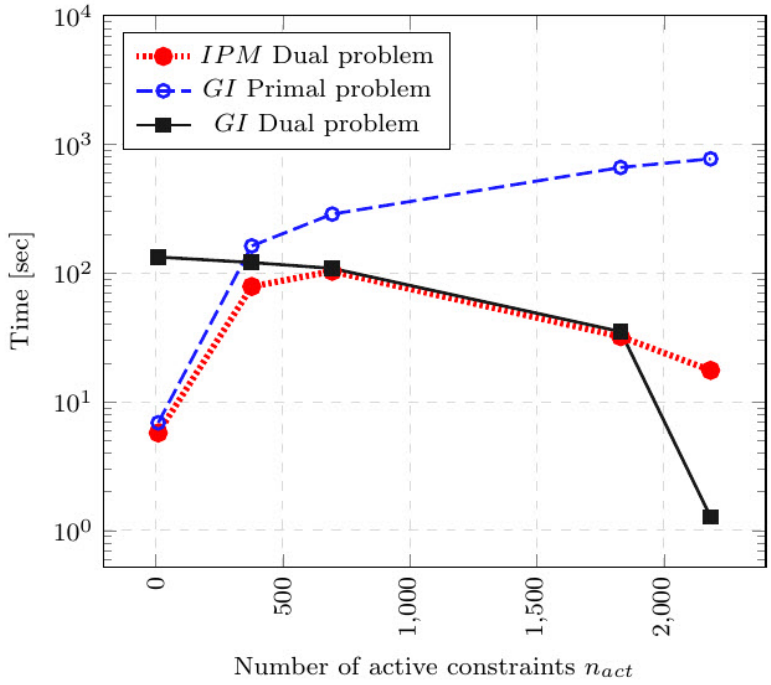

Figure 5: Working time of algorithms for problem with 4386 nodes and two parts.

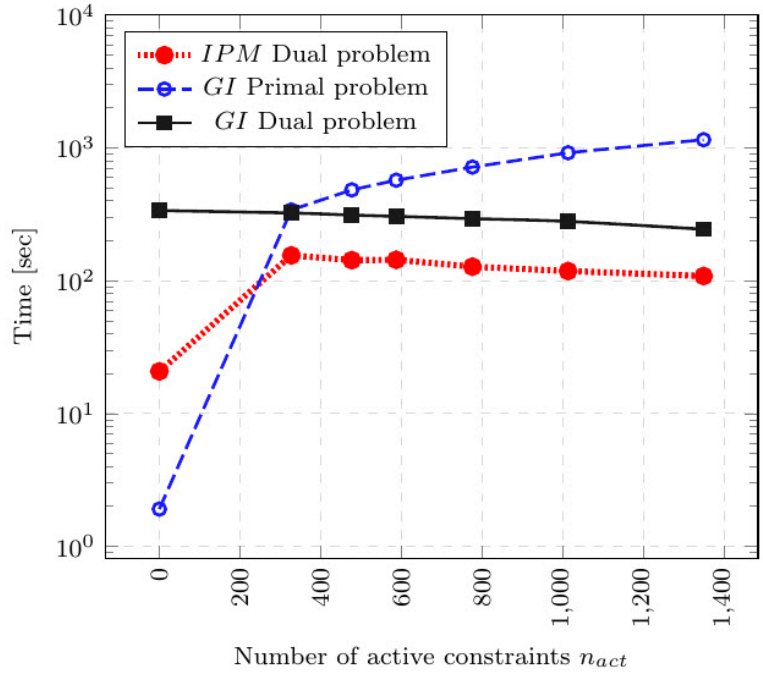

Figure 6: Working time of algorithms for problem with 6112 nodes and two parts.

The largest observed working time of the primal-dual algorithm in its worst case is always smaller than Goldfarb-Idnani algorithm in its worst case (Figures 5–7) for large-scale problems. This does not contradict asymptotic estimates listed in Table 3.

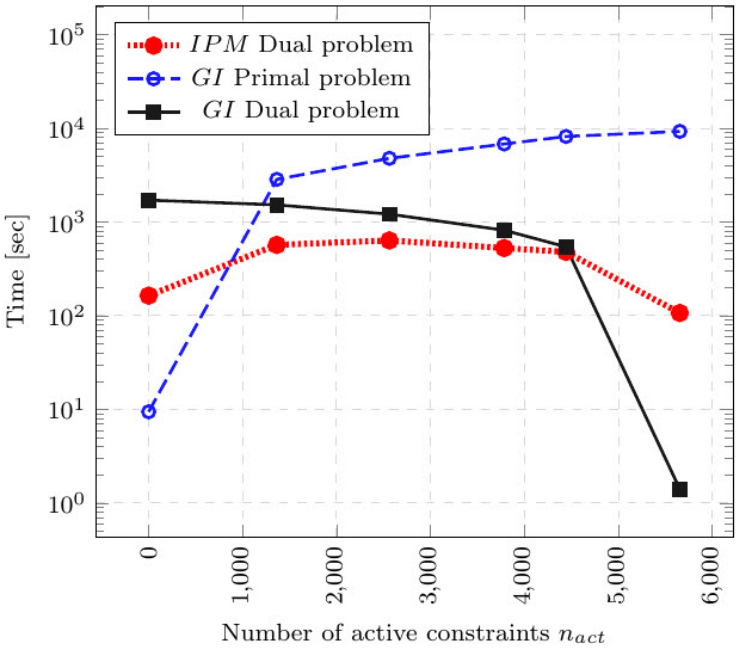

Figure 7: Working time of algorithms for problem with 11308 nodes and two parts.

The Goldfarb–Idnani algorithm can be applied efficiently when we have sufficient information about size of the contact area. When the contact occurs in half of the nodes, the primal-dual method is always faster than the Goldfarb–Idnani algorithm applied to the primal problem, and for some problems also to the dual one.

7 Conclusion

This paper focuses on two aspects of applying the primal-dual interior point method to solve special class of contact problems: preconditioning and determining initial guess. The specific features of the considered problems allow us to build an effective preconditioner with straightforward and rapid updating. Both the proposed preconditioner and the strategy for choosing initial guess help contribute greatly to reducing the working time of the primal-dual method for quadratic problems such as (2).

In conclusion, primal-dual method appears to be more suitable than Goldfarb-Idnani algorithm for large-scale problems arising during the airframe assembly. Such problems usually occur when computing stresses caused by assembly or shimming prediction for large aircraft parts.

References

Costa, M. F. P., Fernandes, E. M. G. P. 2011. “On Minimizing Objective and KKT Error in a Filter Line Search Strategy for an Interior Point Method.” Computational Science and Its Applications - ICCSA 2011 231–244

D’Apuzzo, M., De Simone, V., and di Serafino, D. 2010. “Starting-point strategies for an infeasible potential reduction method.” Optimization Letters, 4(1): 131–146

Eck C., Janusek J., and Krbec M. 2005. Unilateral contact problems. Variational methods and existing theorems. Hoboken, NJ : Taylor and Francis

Gill, P. E., Murray, W., Saunders, M. A., Tomlin, J. A., Wright, M. H. 1986. “On the projected Newton barrier methods for linear programming and an equivalence to Karmarkar’s projective method.” Mathematical Programming, 36: 183–209

Goldfarb D., and Idnani A. 1983. “A numerically stable dual method for solving strictly quadratic programs.” Mathematical programming, 27(1): 1–33

Gondzio, J. 2012. “Interior Point Methods 25 Years Later.” European Journal of Operation Research, 587–601

Gondzio, J., and Grothey, A. 2005. "Direct Solution of Linear Systems of Size 10^9 Arising in Optimization with Interior Point Methods." Proceeding PPAM'05 Proceedings of the 6th international conference on Parallel Processing and Applied Mathematics, Poland, September, 11–14

Hlavacek, I., Haslinger, J., Necas, J., Lovisek, J. 1998. Solution of Variational Inequalities in Mechanics. New York: Springer-Verlag

Karmarkar, N. 1984. "A new polymomial-time algorithm for linear programming." Combinatorica, 4: 373–395

Kinderlehrer, D., and Stampacchia, G. 2000. An Introduction to Variational Inequalities and Their Applications. Philadelphia, PA: SIAM

Lupuleac, S. V., Petukhova, M. V., Shinder, Y. K., Bretagnol, B. 2011. "Methodology for Solving Contact Problem during Riveting Process." SAE Int. J. Aerosp., 4(2): 952–957

Lupuleac, S. V., Shinder, Y. K., Petukhova, M. V., Yakunin, S. A., Smirnov, A. B., Bondarenko, D. 2013. "Development of Numerical Methods for Simulation of Airframe Assembly Process." SAE Int. J. of Aerosp., 6(1): 101–105,

Mazorche, S. R., Herskovits, J., Canelas, A., Guerra, G. M. 2007. "Solution of contact problems in linear elasticity using a feasible interior point algorithm for nonlinear complementarity problems." Proceedings of the Seventh World Congress on Structural and Multidisciplinary Optimization, COEX, Seoul, May, 21–25

Mehrotra, S. 1992. "On the Implementation of a Primal-Dual Interior Point Method." SIAM J. Optimization, 2(4): 575–601

Meszaros, C. 2009. "The bpmpd interior point solver for convex quadratically constrained quadratic programming problems"Proceeding LSSC'09 Proceedings of the 7th nternational conference on Large-Scale Scientific Computing, Bulgaria, June, 04–08

Meszaros, C. 2011. "Solving quadratically constrained convex optimization problems with an interior-point method."Journal Optimization Methods and Software, 26(3): 421–429

Nesterov, Y., and Nemirovski, A. 1994. "Interior-point polynomial algorithms in convex programming." SIAM Studies in Applied Mathematics.

Oliveira, A. R. L., and Sorensen, D. C. 2005. "A new class of preconditioners for large-scale linear systems from interior point methods for linear programming." Linear Algebra and its Applications, 394: 1–24

Pedersen P. 2006. "A direct analysis of elastic contact using super elements."Computational Mechanics 37(3): 221–231

Petukhova, M. V., Lupuleac, S. V., Shinder, Y. K., Smirnov, A. B., Yakunin, S. A., Bretagnol, B. 2014. "Numerical approach for airframe assembly simulation." Journal of Mathematics in Industry, 4:8

Powell, M. J. D. 1985. "On the quadratic programming algorithm of Goldfarb and Idnani." Mahematical Programming Study, 25: 46–61

Tanoh, G., Renard, Y., and Noll, D. 2004. "Computational experience with an interior point algorithm for large scale contact problems." Optimization Online, http://www.optimization-online.org/DB_HTML/2004/12/1012.html

Vanderbei, R. J. 1998. "LOQO: An interior point code for quadratic programming."Optimization methods and software 11(1–4): 451–484

Wang, L., Liu, X., and Zhang, Z. 2017. "An efficient interior-point algorithm with new non-monotone line search filter method for nonlinear constrained programming."Engineering Optimization 49(2): 290–310