Analysis and optimization of conceptual aircraft landing gear designs

Ritik Kothari[1]
Supervised by Dr. Craig A. Steeves

[1] *Graduate Department of Aerospace Science and Engineering*
University of Toronto

Abstract

In this paper, we present the development of a finite element method (FEM) component for Safran Landing Systems' George multidisciplinary optimization tool, specifically aimed at the analysis and optimization of conceptual aircraft landing gear designs. The study addresses the challenges posed by indeterminate beam structures and enhances George's functionality by integrating FEM techniques. Implementation of the FEM focuses on deriving stiffness matrices, computing nodal displacements, and evaluating internal joint forces for beam structures. A comparison between George's existing SFrame optimization method and the developing FEM module highlights successes in optimizing determinate structures and limitations encountered in more complex indeterminate designs. Further development and refinement of the FEM code are required to address discrepancies observed in the optimization of indeterminate structures, with future work set to explore the addition of joint release spring elements and improved computational efficiency through a Guyan stiffness matrix reduction technique. The project's overall goal is to enable George to handle a wider range of landing gear structures, ultimately reducing the time and cost associated with the design of landing gear.

1. INTRODUCTION

The design process of an aircraft's landing gear is often quite long and tedious. Design teams must work closely with stress teams to ensure the development of a landing gear strong enough to withstand the loads it will face in flight, and also light enough to avoid high costs of manufacturing. In an effort to help accelerate the design process, the Research and Technology department at Safran Landing Systems has been developing a multidisciplinary landing gear design and optimization tool, referred to as George, for the preliminary conceptual design phase (Ellis 2023). This end-to-end research effort seeks to develop software that can be used to explore the design space, allowing engineers and technicians to make more informed decisions earlier in the design process (Ellis 2023). Furthermore, the creation of conceptual landing gear models using the George tool is expected to reduce the number of hours spent during the design process by a significant amount, allowing for the opportunity to allocate these hours towards other parts of manufacturing.

As an optimization process, the overall objective of the George tool is the minimization of a conceptual landing gear's structural mass (Ellis 2023). While the existing process, known as SFrame, works well in solving for the nodal forces acting on determinate beam structures, a key drawback of the George tool is its limited ability in solving for an indeterminate structure's nodal forces. This limitation motivates the development and integration of a sophisticated finite element method that aims to accurately solve determinate and indeterminate structures, via beam discretizations and the formation of stiffness matrices.

In this paper, we discuss modifications made to the George tool that improve its functionality and efficiency in regards to generating an optimized indeterminate landing gear structure. Beginning with a literature review of the finite element method (FEM) and structural mechanics, an initial computational approach using Python was developed to determine the stiffness matrix of a simple two dimensional beam frame (Section 2.a). We also outline the modification of a Safran employee's existing FEA code using stiffness matrix calculations from the literature review, which was done in order to solve for the nodal forces of a Safran proprietary landing gear design (in two and three dimensions). In Section 2.b, we discuss connections between stiffness matrices, nodal displacements, and internal beam forces. Section 2.c details work performed for the integration of a complete FEA method into the George tool's codebase, through collaboration with another member of Safran's George Development Team. Kinematic results comparing the accuracy of the FEA method to George's SFrame method are presented in Section 3, along with a discussion of their significance in Section 4. The paper concludes with a summary of key findings and possibilities for future study in Section 5.

2. METHODOLOGY

In this section, we give a brief overview of the project's progress, from literature review to final implementation of the FEM into the George tool.

2.a. *FEM Literature Review and Initial Attempt*

At the start of the project, the primary objective was to develop an understanding of structural mechanics and the FEM. A critical reference for much of the work during this stage (and subsequent ones) was the fifth edition of Daryl L. Logan's "A First Course in the Finite Element Method", particularly Chapters 4 and 5. In the project's first two weeks, code was developed from scratch in order to calculate the global stiffness matrix of a simple 2D plane frame (shown in Figure 1), following the method outlined in Chapter 4 of Logan (2012).

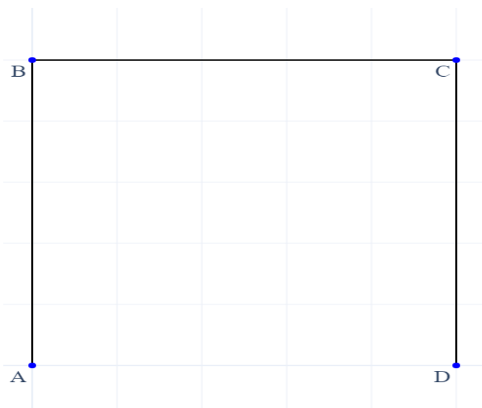


Figure 1: A simple two dimensional planar frame. Following the stiffness matrix computation method from Logan (2012), the displacements of joints A, B, C, D were solved for given an applied load at B. The nodal forces acting on each joint were also found.

For the frame in Figure 1, the local stiffness matrices of each beam element were found, from which the global stiffness matrix of the frame was found via superposition. As this solution was coded specifically to solve this beam arrangement, there were several limitations. In particular, the written code did not account for the addition of support beams. To address this issue, a Safran employee's code (referred to as "Andrew's code" henceforth) was provided as a more general way of solving the plane frame. With this new foundation, the stiffness matrix calculations from the original code were implemented into Andrew's code to develop the local matrices for the plane frame.

Following this exercise, Andrew's code was used to solve a 2D model of the Global7000, a Safran proprietary landing gear. After validating the results, the functionality of Andrew's code was extended to 3D systems as well, using Logan (2012) as a reference. This work led to the development of the FEM for the Global7000 assembly, for which beam members were discretized into multiple nodes, and reconstructed as "sliced" beam elements. The global stiffness matrix of the assembly was then found by superposing sliced element matrices.

2.b. *Stiffness Matrices, Nodal Displacements, and Internal Forces*

With the successful implementation of stiffness matrix calculations into Andrew's code, it was found that the calculation of nodal displacements and internal forces needed to be modified to effectively solve three dimensional models. Starting with the simple frame from Figure 1, we attempted to solve for its nodal displacements using the equation (Logan 2012),

$$L = K \times d_{\text{global}} \implies d_{\text{global}} = K^{-1} \times L, \tag{1}$$

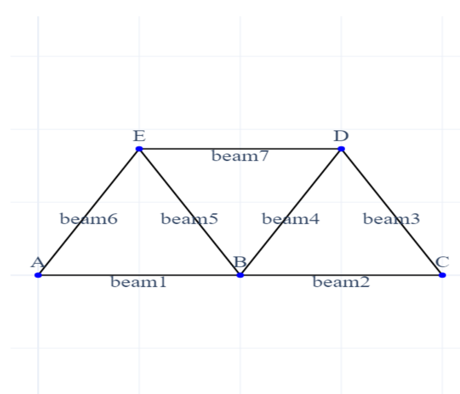
where L is a load vector, K is the (invertible and constrained) global stiffness matrix, and d_{global} is the global nodal displacement vector. Given a horizontal load of 1000 N at node B in Figure 1, the nodal displacements at A, B were found using Python code and Equation 1.

Having obtained the nodal displacements, the internal forces acting on the beam member spanned over A, B were found using the equation (Logan 2012),

$$F = K \times d_{\text{local}}, \tag{2}$$

where F is a force vector and d_{local} is a displacement vector that has been transformed to the beam member's local frame via a transformation matrix. This procedure was repeated for the other beam members, as well as a two dimensional model of the Global7000 landing gear frame.

Comparing the previously obtained results with those obtained from the George tool for the same frames, it was apparent that the results were not in mechanical equilibrium. Consequently, tests were ran using simple trusses in order to try and deduce where the error(s) would be.



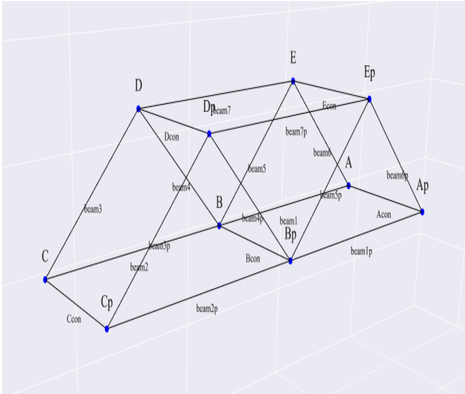


Figure 2: (Top panel) A simple two dimensional truss and (bottom panel) a simple three dimensional truss. For each frame, a load of 1000 N was applied at joint E. The results of solving each frame were used to debug and solve errors in the developing FEM code.

2.c. *Implementation of FEM into George Codebase*

Convex optimization software often involves solving a problem of the form

$$\begin{aligned} \min_{x \in X} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \quad \text{for } i \in \mathbb{N}, \end{aligned} \tag{3}$$

where $f(x)$ is an objective function, $X \subseteq \mathbb{R}^n$ is the set of feasible solutions, and $g_i(x)$ is the set of constraints. The objective function is usually differentiable, and one or more of its derivatives (or gradient for multivariate functions) are used in the optimization process. This gradient-based approach is in essence how the George tool functions.

Safran's George tool has been developed with the idea of multidisciplinary optimization in mind, in order to optimize various parts of the preliminary design process simultaneously. As the George tool is primarily a Python-based software optimization tool, the Python open source framework `OpenMDAO` forms the foundation of the optimization process (Gray et al. 2019).

The `OpenMDAO` framework relies on the definition of `IndepVarComp`, `ImplicitComponents`, and `ExplicitComponents` in place of the traditional dependent and independent variables used in optimization problems. These systems take in inputs and produce outputs that are used interchangeably throughout the optimization process, leading to the minimization of the objective function (Gray et al. 2019). At a high level, the objective of George's optimization process is to minimize the mass (and thus cost) of a landing gear design, which is subject to constraints such as beam member dimensions and structural material strength, among others (Ellis 2023).

Before fully implementing FEA work into the George codebase while adhering to the `OpenMDAO` architecture, there were several steps that needed to be taken. Use of preliminary FEA code often resulted in large stiffness matrices, which grew in size proportional to the number of beam member discretizations. These large matrices are computationally expensive to work with, so adding a dimensionality reduction method (referenced from Ellis (2023)) was prioritized.

Considering a global stiffness matrix K_g, which contains the degrees of freedom (DoFs) of all nodes present in the structure, we can treat the DoFs of discretizing nodes as secondary and the DoFs of main joints as primary. Using this approach, K_g can be partitioned in the following way (Ellis 2023):

$$K_g = \left[\begin{array}{c|c} K_{pp} & K_{sp} \\ \hline K_{ps} & K_{ss} \end{array} \right] \tag{4}$$

where the subscript p refers to the primary DoFs and the subscript s refers to the secondary DoFs. Simply partitioning the global stiffness matrix in this way does not reduce its dimensions however, this is done via the Guyan Reduction method (Ellis 2023):

$$K_r = K_{pp} - K_{ps} K_{ss}^{-1} K_{sp} \tag{5}$$

where K_r is the reduced version of K_g, matching the dimensions of K_{pp}. Employing the Guyan Reduction method allows for information from K_g to be preserved, while also reducing computational times for operations requiring the global stiffness matrix.

In addition, the FEA code needed to take into account the formation of a structure's global reduced stiffness matrix K_r using the local reduced stiffness matrices of each beam member, the accommodation of multiple joint types, and the orientation or type of beam members with respect to the global frame.

These considerations were implemented into the George codebase with the assistance of a member of the George Development Team, while adhering to the `OpenMDAO` architecture. A significant challenge during this process was the construction of analytic partial derivatives for each `OpenMDAO` component, which are extremely important for reducing the overall optimization runtime. This part of the project is still in development, and will be continuously improved by the George Development Team.

3. RESULTS

We present here several tables that compare the results of an optimization run using the FEM and George's existing SFrame methodology. The comparison was run for both an indeterminate and a determinate beam structure, particularly a simply supported truss (shown in Figure 3) and a Safran proprietary landing gear, which cannot be shown due to confidentiality reasons. The first important result from our tests is the relative error we find between the FEM and George's SFrame optimization results for a determinate beam structure, seen in Tables 1 and 2. For brevity, we only include results for the beam member attached to the load application joint **O** (Trailing Arm). For our second comparison, we give the relative error between the FEM and George's SFrame optimization results for an indeterminate beam structure, shown in Tables 3, 4, 5, and 6. In this test, we only include results for the beam connected to the load application joint **O**, and the beam protruding out of the truss' plane. Note that for both tests, we apply a load vector

$$\vec{L} = [f_1, f_2, f_3, m_1, m_2, m_3],$$

where $f_i = 500\,\mathrm{N}$ and $m_i = 10^6\,\mathrm{kN{*}mm}$, for $i \in \{1, 2, 3\}$.

3.a. *Comparison Results for Determinate Beam Structure*

In the below tables, we present the relative error between the FEM's optimization results and those obtained using SFrame, for a single beam member of a determinate Safran proprietary landing gear. The Trailing Arm was chosen in particular since it is the beam member attached to the load application point **O**, which experiences a variety of forces upon landing. Note that SFrame's results are assumed to be correct, due to exhaustive testing done by other members of the George Development Team. Small relative error values would imply a successful comparison test for the FEM.

Joint	Fu [kN]	Fv [kN]	Fw [kN]
E	0	0	0
F	0	0	0
O	0	0	0

Table 1: Trailing Arm relative error results for a FEM and SFrame optimization run. This table only includes comparison results for the internal translational forces acting on each joint. Here, u, v, w describe the beam member's local x, y, z axes.

The data displayed in Tables 1 and 2 shows that the FEM optimization results match very closely with those obtained using SFrame. For reference, the relative error formula used to generate these tables is

$$\delta = \left| \frac{v_a - v_e}{v_e} \right| \tag{6}$$

where $0 \le \delta \le 1$ is the relative error, v_a is the SFrame value, and v_e is the FEM value.

Joint	Mu [kN*mm]	Mv [kN*mm]	Mw [kN*mm]
E	0	0	0
F	0	0	0
O	0	0	0

Table 2: Trailing Arm relative error results for a FEM and SFrame optimization run. This table only includes comparison results for the internal rotational forces acting on each joint.

The scale on which we evaluated our relative error results is also presented below:

$$\begin{cases} \text{Pass} & \delta \le 0.0001 \\ \text{Concern} & 0.0001 \le \delta \le 0.05 \\ \text{Fail} & \delta \ge 0.05 \end{cases} \tag{7}$$

While not included, we obtained similar relative error values for the other beam members of the landing gear.

3.b. *Comparison Results for Indeterminate Beam Structure*

In this section, we present the relative error values between the FEM's optimization results and those obtained using SFrame, for two beam members of an indeterminate beam assembly, pictured below:

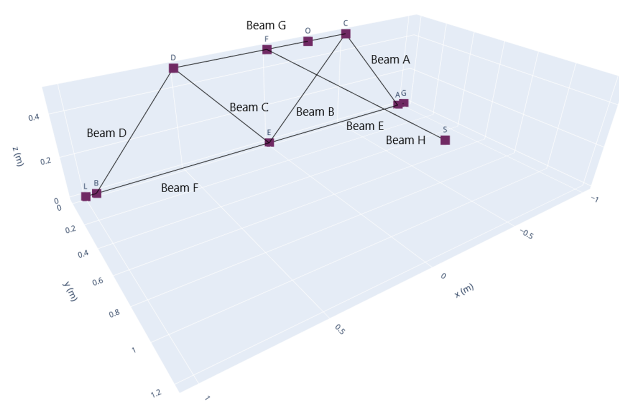


Figure 3: A planar truss attached to a protruding beam, with joints **G**, **L**, **S** fixed. A load vector $\vec{L}$ is applied at joint **O**, which is connected as part of Beam G. Joints **G** and **L** are attached to the main structure via support beams connected to joints **A** and **B** (not labeled).

The FEM and SFrame comparison results for Beam G in the above figure are presented in Tables 3 and 4 below. Note that while indeterminate Safran landing gears do exist, a simple truss was chosen to run this test due to the relative simplicity for debugging, should the need arise.

Joint	Fu [kN]	Fv [kN]	Fw [kN]
C	0.8071	-0.8571	0.9216
O	0	0	0
F	1.1025	-1.1451	1.1193
D	-0.8694	0.9995	-0.9967

Table 3: Beam G relative error results for a FEM and SFrame optimization run. This table only includes comparison results for the internal translational forces acting on each joint.

Joint	Mu [kN*mm]	Mv [kN*mm]	Mw [kN*mm]
C	-1.5692	3.3128	1.5811
O	0	0	0
F	-0.9959	-0.131	1.019
D	-0.9997	-0.9947	1.0141

Table 4: Beam G relative error results for a FEM and SFrame optimization run. This table only includes comparison results for the internal rotational forces acting on each joint.

According to our evaluation criteria, the vast majority of the results in both tables imply failure for our comparison test. An explanation as to why this might be is provided in Section 4.

The following tables display the relative error values for the protruding Beam H of Figure 3.

Joint	Fu [kN]	Fv [kN]	Fw [kN]
F	-1.1451	-1.1193	1.1025
S	1.1451	1.1193	-1.1025

Table 5: Beam H relative error results for a FEM and SFrame optimization run. This table only includes comparison results for the internal translational forces acting on each joint.

Joint	Mu [kN*mm]	Mv [kN*mm]	Mw [kN*mm]
F	-0.131	-1.019	-0.9959
S	0.131	-1.2693	-1.4399

Table 6: Beam H relative error results for a FEM and SFrame optimization run. This table only includes comparison results for the internal rotational forces acting on each joint.

Similar to the results for Beam G, these results also imply failure for our comparison test. Possible reasons for these results, as well as an explanation of why a determinate structure was solved perfectly, are provided in the following section.

4. DISCUSSION

In this section, we discuss the significance and meaning of the results from Section 3.

Regarding the tables in Section 3.a, we can immediately infer that the FEM and SFrame comparison test is successful, according to our evaluation criteria given by Equation 7. This success supports our correct implementation of FEM into the George codebase, as both the FEM and SFrame need to optimize a given structural design and obtain the same results. We also obtained similar relative error results when testing other determinate beam structures, such as a simple cantilever beam fixed on one end (not included for brevity).

In contrast, the relative errors shown in Section 3.b show a large difference between FEM and SFrame internal joint force values. While the exact cause of these discrepancies is still under investigation, there are several working theories being explored by the George Development Team.

The first of these theories involves a feature present in the SFrame code that is not part of the FEM. To apply local joint releases, that is to say, translational or rotational restrictions to a joint's DoFs in its local frame, SFrame incorporates spring elements between beam members. Spring elements are simple rigid beams that overlap at joint intersections between beam members, and they are used to avoid having to release joints in more than one local frame. Our implementation of the FEA method into the George codebase does not use spring elements to release joints, but rather releases them directly within their associated beam member's local frame. As a result of this difference in implementation, the global structural stiffness matrices generated by SFrame and the FEM differ slightly, possibly leading to different internal joint forces obtained by each method.

On the other hand, a significant difference between the two optimization methods is the use of Guyan reduction in the FEM, and the lack of it in SFrame. To reiterate, the FEM generates stiffness matrices for a structure's members via joint discretizations along a member's axis. A large number of discretizations would result in large stiffness matrices, which can be computationally expensive to work with for future calculations. As a solution to this problem, Guyan reduction was implemented into the FEM in order to reduce the size of the stiffness matrices, making computations involving them more robust. More research needs to be conducted in order to explain the discrepancies due to Guyan reduction, an idea of which is outlined in Section 5.

5. CONCLUSIONS AND FUTURE WORK

In this study, we begin development on a new finite element method (FEM) component for Safran Landing Systems' George multidisciplinary optimization tool, specifically for the analysis and optimization of indeterminate landing gear designs. Referencing the insights for the three dimensional finite element method from Logan (2012) and Ellis (2023), our FEM development represents a significant step toward improving George's capabilities in handling more complex structural systems in the preliminary design phase.

The implementation of the FEM is achieved by first analyzing literature, particularly Logan (2012) and Ellis (2023), to design a foundation for a two dimensional FEA code. This is later extended to a more comprehensive 3D FEM approach that allows for the analysis of realistic landing gear designs. Throughout development, particular emphasis was placed on ensuring that the FEM component integrated smoothly within the George tool's existing architecture, which relies heavily on the OpenMDAO framework for optimization.

The results obtained during our comparative analysis showed that the FEM-based optimization aligns well with George's original SFrame methodology when applied to determinate structures. In these cases, the optimization results exhibited minimal relative error, confirming that the FEM approach is valid for simpler structures. This consistency also provides evidence of the successful incorporation of FEM techniques into George, providing Safran engineers with a valuable tool for early-stage design decision-making.

However, the study also revealed key challenges when applying the FEM method to indeterminate structures, with the relative errors observed in these cases suggesting that further development is needed. A primary source of error appears to stem from the differences in how joint releases are handled between the SFrame and FEM methods. While SFrame utilizes spring elements for joint releases, the FEM method developed here applies direct joint releases within each beam's local frame. This difference likely affects the global structural stiffness matrices, leading to discrepancies in the internal force calculations. Additionally, the use of Guyan reduction for stiffness matrix dimensionality reduction in the FEM method introduces further complexities, as this process is not employed in the SFrame method.

Moving forward, several areas of future work are identified to improve the FEM implementation within the George tool. First, the method of joint release in the FEM framework must be revisited to ensure it closely mirrors that of SFrame's. Second, more comprehensive testing of the FEM method on a variety of indeterminate structures is necessary to better understand the limitations of the current approach and to refine it accordingly. A detailed investigation into the impact of Guyan reduction on the optimization process will also be important in ensuring that computational efficiency is not achieved at the expense of accuracy.

Ultimately, the integration of a refined FEM component into the George tool holds great promise for improving the efficiency of the preliminary landing gear design process. By addressing the limitations identified in this study, George will be better equipped to optimize both determinate and indeterminate structures, contributing to a reduction in design time and manufacturing costs. The future development of this tool will not only enhance Safran Landing Systems' design capabilities, but also serve as a valuable case study in the application of FEM in aerospace design optimization.

REFERENCES

Ellis, A. 2023, Integrating Dynamics into Landing Gear Multidisciplinary Optimization

Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., & Naylor, B. A. 2019, Structural and Multidisciplinary Optimization, 59, 1075, doi: 10.1007/s00158-019-02211-z

Logan, D. L. 2012, A First Course in the Finite Element Method (Toronto: Nelson)