Rotorcraft Fuselage Sizing Methods in the Open-source Framework Pandora

D. B. Schwinn, M. Petsch, D. Kohlgrüber, J. Heubischl, C. Leon Munoz
(German Aerospace Center (DLR), Institute of Structures and Design, Germany)

P. Weiand, M. Buchwald
(German Aerospace Center (DLR), Institute of Flight Systems, Germany)

Abstract

Rural depopulation resulting in altered hospital coverage, new challenges for medical evacuation during military operations, and increased off-shore activities of energy suppliers, lead to changed requirements of helicopter emergency medical services (HEMS). Recently, interest has significantly increased to overcome the traditional physical limitation of flight speed by providing helicopters with auxiliary propulsive devices, so-called compound rotorcraft. In order to assess these novel rotorcraft concepts, an integrated, multidisciplinary, and automated design procedure has been established at the German Aerospace Center (DLR) using the data model CPACS (Common Parametric Aircraft Configuration Schema).

The design processes of rotary- and fixed-wing aircraft highly resemble each other: In the first stage of a typical aircraft design process, the conceptual stage, basic characteristics are established that typically consist of e.g. outer dimensions (i.e. its aerodynamic shape), flight performance, mass breakdown, etc. At this stage of the design process mostly fast, analytical, and statistical methods are applied featuring many simplifications. In the subsequent preliminary design phase the detail level increases. The continuously growing computational power has enabled design engineers to integrate higher fidelity methods at this design stage. At the DLR Institute of Structures and Design, tools have been developed in the last couple of years that use finite element (FE) methods to size aeronautical fuselage structures according to static load cases to allow a more precise prediction of the structural mass, and thus in turn to a different maximum take-off mass which is considered as a major design parameter.

Although based on the same framework approach, these FE based tools developed for preliminary sizing of rotary- and fixed-wing fuselages diverged over the years due to different project requirements, such as specific modeling aspects, different syntax for the involved FE solvers, or different design emphases. These issues resulted in different tools to generate FE meshes and to conduct analyses with some inconsistencies between the individual tools. In order to unify the tools, the development of the software framework

PANDORA (Parametric Numerical Design and Optimization Routines for Aircraft) has been started at DLR in 2016 from scratch using the Python programming language. The key idea behind PANDORA is to generate one common software framework to model, analyze, and size both fixed- and rotary-wing fuselage structures. Particular focus in the development of PANDORA lies in the use of dedicated open-source packages and the interchangeability of different commercial and open-source FE solvers.

This paper first shows the approach of the PANDORA toolbox for fixed-wing aircraft. Then, the process of adapting respectively integrating specific modeling and analysis methods for rotorcraft fuselages into the new framework is shown. Concluding this article an outlook of new enhancements into PANDORA is given highlighting its benefits in the context of preliminary structural analysis of novel rotorcraft concepts.

1. Introduction

A common phenomenon among most industrialized countries these days is rural depopulation. Young people often leave their home and move to major cities for education and/or work. As a result the tax income in these rural and sparely populated regions decreases. Tax income in turn is used in most countries as funding to maintain its infrastructure, such as roads and public institutions, for instance hospitals and medical emergency centers. Lowered tax income paired with a reduced population often leads reduced funding and eventually to closing of the corresponding medical facilities. This circumstance results in longer distances to cover for medical emergency units to arrive at an accident site. Many countries have laws that guarantee the arrival of a medical unit within a certain time after the emergency call. In order to still match those times, emergency vehicles have to be operated with higher speed due to the increased distances.

A similar situation can be observed in the energy industry offshore. Increasing need for fossil energy, such as oil and gas, has led the energy suppliers to explore farther offshore. The approach to use wind as renewable energy has led to the installation of huge wind parks offshore. In order to not impede ship traffic too much, new wind parks are to be installed farther offshore as well. Both circumstances increase the distance for the operational and maintenance crew, not only for transportation but also in case of emergency.

Rotorcraft feature an important branch in aeronautics due to their unique abilities of very slow flight, hovering, taking-off and landing vertically. They are used in a wide variety of operations, such as search and rescue missions, especially in areas that are difficult to access, such as mountains and offshore. Hence one major and very important operational field for rotorcraft is the use as helicopter emergency medical service (HEMS).

In contrast, helicopters feature limited capabilities in maximum flight speed and range. Moreover, especially in slow decent flight they are very noisy due to the blade vortex interaction. Considering the aforementioned problem of increasing distances for medical services and thus the maximum velocity, there is a need for novel rotorcraft concepts that overcome the physical limitations of speed.

The limitation in flight speed is schematically shown in Fig. 1. The figure on the left shows a main rotor (seen from top) during hover rotating counter-clockwise with the angular velocity ω. Due to the complicated dynamic reaction of the blades the angular velocity is kept constant for conventional configurations. The flow velocity u is a function of the rotor radius r:

$$u(r) = \omega \times r \tag{1}$$

In case the helicopter tilts its rotor tip plane forward, the lift vector changes its direction and the rotorcraft starts to fly forward. A constant flow field v_∞ is shown in blue in the centered figure indicating the freestream velocity induced by the horizontal flight. The combination of both velocities u and v_∞ is shown in green in the right figure. On the advancing blade side the velocities are added leading to an increased velocity while on the retreating blade side the velocities have different directions, therefore they are subtracted.

The addition of both velocities adds the dependancy of the rotor azimuth angle ψ (exemplary shown in Fig. 1, center):

$$u(r) + v_\infty = u(r, \psi) \tag{2}$$

It can be seen that on the advancing blade side the blade tip experiences much higher velocities which eventually lead to transonic flow problems. On the retreating side the velocities are comparably lower so that stall problems can arise. It shall be noted that Fig. 1 shows a simplified approach without flapping and lagging effects. However, the general problem of speed limitation for rotorcraft can be demonstrated in principle.

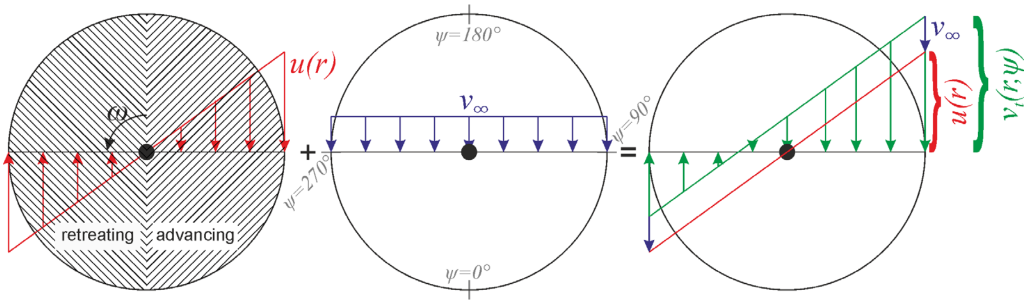

Figure 1: Flight speed limitation of rotorcraft

One solution to this problem of limited speed lies in the use of so-called compound helicopters. These feature wings that generate additional lift in forward flight so that the main rotor(s) can be off-loaded and slowed down. Modern design approaches will allow to slow down the main rotor in predictable limits without encountering dynamic response. In order to overcome drag, especially at higher flight velocities, additional thrust needs to be generated by one (or more) propeller(s). This principle is sensible to increase the forward flight velocity v_∞ (exemplary shown is the Airbus X^3 in Fig. 2 on the left). Another possible solution is the use of tilt-rotor configurations. These rotorcraft feature two main rotors mounted at a central wing that can be tilted about the lateral axis so that they can be operated like a fixed-wing aircraft (exemplary shown is the Bell/Boeing V-22 in Fig. 2 on the right). Even for a higher disc loading in comparison to the conventional main rotor the side-by-side arrangement of the rotors requires an increased amount of space. During take-off and landing this circumstance may complicate their use, for example when trying to land in narrow locations, such as forest glades or canyons.

Figure 2: Compound and tilt-rotor configurations (sources: airbus.com and bellflight.com)

Two new demonstrators that are currently under development within the Future Vertical Lift (FVL) program by the US Armed Forces are the SB>1 Defiant by Sikorsky and Boeing and the V-280 Valor by Bell. It shall be noted that the Defiant does not feature wings but uses the Advancing Blade Concept (ABC, [1, 2]). One of the specified design requirements is to achieve flight speeds of at least $v = 430$ km/h.

An overview of the maximum flight speeds of several helicopters is shown in Fig. 3. It can be observed that the typical 'standard' configurations (one main rotor and an anti-torque mechanism, coaxial helicopters, tandem helicopters, highlighted in black) feature a flight speed limit of slightly below 350 km/h. Compound helicopters (highlighted in blue) show superior flight speeds almost reaching 500 km/h. The highest flight speeds for rotorcraft are achieved by using tilt-rotor concepts (highlighted in red). But these configurations are also the most complex and most expensive ones in order to transport a specified payload. Therefore, the solution of the initially stated problem of increased flight speed for HEMS can not only be seen in the development of novel configurations. Furthermore, the comparison of different virtual configurations is required to assess their economic sense. Additionally, it shall be mentioned that the increase in maximum flight speed is at the expense of the typical rotary-wing characteristics, such as hovering, side- and rearward flight.

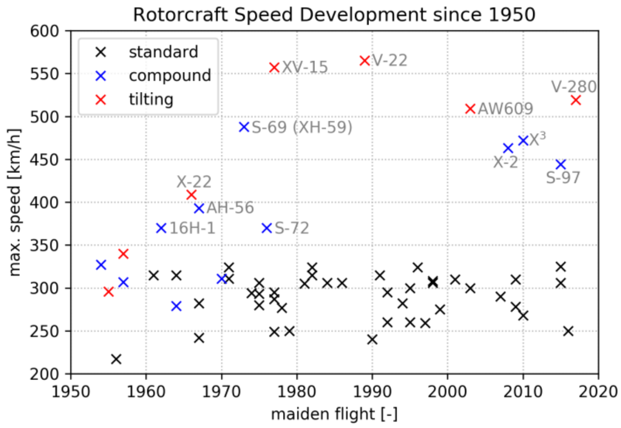

Figure 3: Rotorcraft speed development since 1950 (sources: wikipedia.com)

In general, aeronautical predesign activities widely use data of existing air- and rotorcraft for the early design stage of the conceptual design. The manufacturers can use a huge database considering flight performance, maintenance and repair, construction, aircraft life, as well as cost estimations - just to mention some few. It is understandable that this approach becomes difficult when a novel aircraft configuration is desired by the customer and/or designer for which the aforementioned data is not available.

Compound and tilting configurations are not a new approach by the aeronautical industry. Early concepts were already developed in the 1950s. But the complexity of the aerodynamical, rotor dynamical, and flight mechanical problems made the prediction of the systems behavior and response difficult and without sophisticated computer codes even impossible. However, manufacturers had mostly been working on the enhancement of standard helicopters, e.g. on reducing costs and improving efficiency. As a consequence, reliable data on these high-speed configurations is rather little. Therefore, in order to deal with compound, tilting, or completely new and unconventional configurations, an approach that is rather based on physical modelling instead of statistics becomes feasible.

Increasing computational power nowadays allows design engineers worldwide to couple higher fidelity tools from various disciplines in order to assess mutual influences and effects at very early design phases.

A couple of years ago, the set-up of a design environment had been started at the German Aerospace Center (DLR) in order to assess new rotorcraft concepts. This approach eventually resulted in the new development of the integrated and distributed design environment for rotorcraft IRIS (Integrated Rotorcraft Initial Design, [3]). This approach follows the multidisciplinary design environment for aircraft which had been previously initiated at DLR during the internal project TIVA (Technology Integration for the Virtual Aircraft, [4]). These design environments use the data model CPACS (Common Parametric Aircraft Design, [5]) to describe the air- and rotorcraft system. The network based simulation tool RCE (Remote Component Environment,[6]) is used to set up workflows to generate and assess generic concepts according to user-specified top level aircraft requirements (TLARs), such as range, payload, cruise speed, cabin volume, and more.

2. Design process and software framework

Since the design processes for air- and rotorcraft highly resemble each other, the term aircraft design is used subsequently to describe both processes. However, it shall be noted that even though the phases resemble each other the design extent for rotorcraft is noticeable higher [7]. In general, the design process is subdivided into three consecutive phases, as shown in Fig. 4. In the first design phase an initial concept is evaluated that covers the specified TLARs. In general, analytical and/or statistical methods are used and many simplifications are made to allow fast analyses enabling trade-offs. The automation level in this phase is generally high. At the end of the conceptual phase the outer loft has been determined and a mass estimation has been conducted that are required for basic flight performance calculations. Subsequently, in the preliminary phase, the primary structure within the loft is distributed. Therefore, structural analyses can be conducted allowing a first sizing of the airframe. The tools that are required in this phase feature less simplifications compared to the conceptual design stage. They require more input to set up the models and also more effort to process the results. Therefore, automation in this phase is difficult to achieve. The detailed design phase finalizes the design process, i.e. major changes on the global design are, in general, not expected and will not be performed. This phase is conducted with particular focus on manufacturing aspects resulting in technical drawings for production. The tools used in this phase require the highest computational power and simplifications are the exception, not the rule. The pre- and post-processing of the model respectively of the results requires an enormous manual effort. Therefore, automation in this phase is almost impossible.

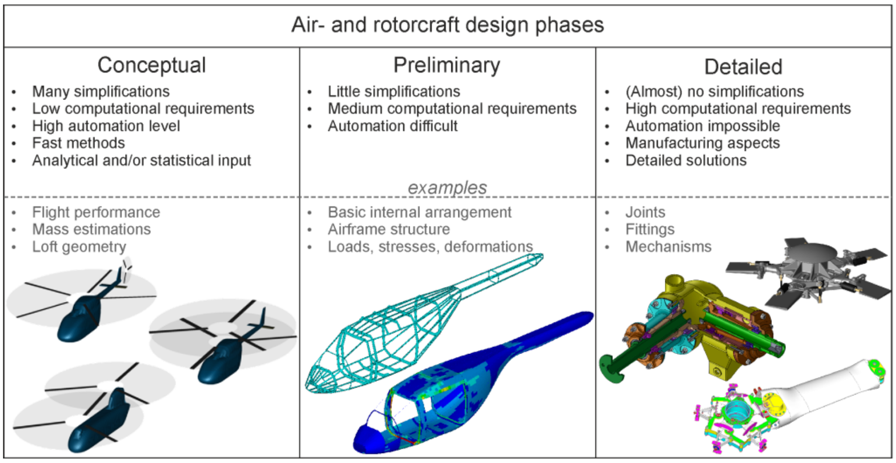

Figure 4: Design phases

2.1. Software framework

One main aspect to benefit from a multidisciplinary and integrated aircraft design process is the flawless connection and communication of the involved tools. They need to interact on two levels, namely data transfer and software processes.

CPACS is a data model in the .xml[1] format and used at DLR for multidisciplinary design activities. It can be considered as a key component for the communication and data exchange between the individual tools and users. Its advantages are the hierarchical structure, easy access and readability. During the design process the CPACS file that describes the aircraft is filled step-by-step with information gained by the individual design tools. It is used to store the results, not only for the assessment of the analysis but also as potential input since results from one tool may serve as input for the subsequent ones. It shall be noted that the use of CPACS also reduces the amount of interfaces n_i for the involved tools n_t from

$$n_i = n_t(n_t - 1) \tag{3}$$

to

$$n_i = 2 \cdot n_t \tag{4}$$

as schematically shown in Fig. 5 resulting in an easier maintenance of the design tools.

[1] Extensible markup language

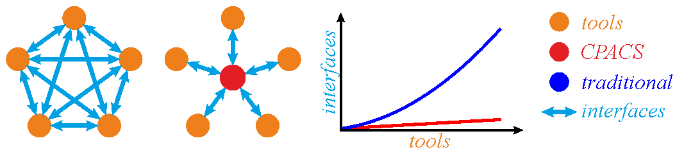

Figure 5: Tool interface reduction

Since many institutes (located at different cities) are involved in the presented design process contributing with a variety of computational tools it was decided to use the software RCE to set-up the workflows and access the design tools. RCE allows the corresponding disciplinary specialists to develop and maintain their tools which are installed on locally separated servers. The tools are published to the participating partners allowing them to execute the program but denying the access to the source code, thus protecting the knowledge of the developing partner. Data is transferred via an internal network, i.e. a CPACS file (serving as input) is sent from the local user to the server where the corresponding tool is installed and where the computation is executed. Subsequent to the computation, the CPACS file is updated with the calculated information and sent back to the local user or to the next tool (respectively server) for the next computation. Optionally, additional results (such as diagrams, etc.) can be returned as well.

2.2. Structural analyses tools

The fuselage structure is statically analyzed and sized at preliminary design level at the Institute of Structures and Design using finite element methods (FEM) with the objective to obtain a more precise estimation of the fuselage mass compared to the estimations of the conceptual design stage. For this purpose a fuselage model generator in the Python programming language was developed that uses the program language APDL (ANSYS Parametric Design Language) to actually generate the FE models. The tools TRAFUMO (Transport Aircraft Fuselage Model, [8]), ROFUMA (Rotorcraft Fuselage Mass Assessment, [9]) and AC-CRASH (Aircraft Crash, [10]) were developed using the aforementioned fuselage modeller to generate FE models and the corresponding input cards (boundary conditions, material models, etc.). Sizing of the fuselage structure was conducted with the tool S-BOT+ (Sizing Robot, [8]) which had been originally developed to size wing structures [11].

Typical for the detail level at the preliminary design phase, the FE model is of global FEM quality (GFEM), i.e. each two-frame and two-stringer bay comprises one skin element. Reinforcements are in general modeled using beam elements while the skin panels are discretized using shell elements. Rotorcraft specific modelling aspects in the presented design process comprise cut-outs, stage modeling [12], and frames that are modeled as extruded shell profiles as proposed by Hunter [13]. Figure 6 shows a rotorcraft FE model with cut-outs and the stage modeling approach in order to model a more realistic stiffness distribution at the tail boom.

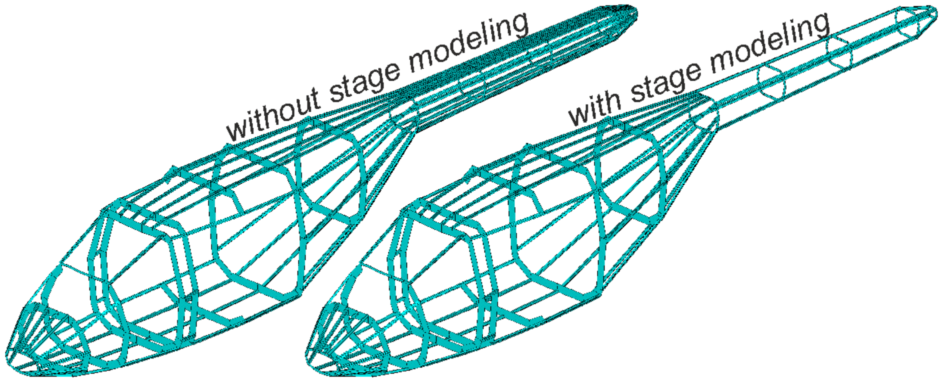

Figure 6: Rotorcraft specific modeling features (ROFUMA) [14]

2.3. Lessons learned

Even though it was intended to keep these tools integer, the development of these tools diverged over the years due to different project requirements and other issues which will be listed first and subsequently explained in further detail:

- General differences between air- and rotorcraft in CPACS
- Model requirements
- Discretization approach
- Requested FE solver
- Time aspects
- Project requirements
- Fluctuation of staff
- Different programming skills
- Lack of software testing methods

General differences between air- and rotorcraft with regard to the fuselage description in CPACS are small except the model node. However, external loads for aircraft are introduced into the structure using so-called dynamic aircraft model (DAM) points which are nodes on the aircraft reference axis. External loads are calculated before the structural analysis and condensed on the DAM points. In contrast, the external loads for rotorcraft are introduced at the rotors. Therefore, the generation of the load introduction for both types of aircraft required a different approach for the reading of the corresponding CPACS data and for the model generation.

Although both, air- and rotorcraft, are based on CPACS they feature a fundamentally different structural construction. Aircraft feature a wingbox and, in general, have two floor levels (cargo and passenger) while rotorcraft do not.

Therefore, several structural members, crossbeams and vertical struts for each floor, have to be modeled for aircraft only. Aircraft feature a more or less smooth splining of the fuselage while most helicopters feature a rough transition from the cabin to the tail boom which requires different boundary conditions for splining.

Static analyses at preliminary design level are modeled using the GFEM approach. Emergency landings require a detailed FEM (DFEM) approach. Therefore, the reinforcements have to be modeled with small shell elements with nonlinear effects, such as plasticity for isotropic materials (at least in the area of interest) and to provide a suitable time step for the dynamic solution. AC-CRASH for example features the integration of so-called detailed regions where the impact area is modeled in detail using a stepwise decrease of detail level up to the global model.

Different projects were concerned with different features. Moreover, the collaboration with different partners (using different solvers) complicated the comparison of the results and integration of sub-models (for instance the addition of wings to the fuselage). Rotorcraft projects in contrast focussed on the automated generation of cut-outs and the distribution of primary structure within the fuselage, especially around the cut-outs, and a realistic stiffness distribution at the tail boom. Another important issue concerning the solver choice are license costs and availability (e.g. the license server is down or a license pool where the user has to share the licenses with colleagues).

Time aspects during sizing were critical for aircraft due to the size of the fuselage. This aspect could be neglected for helicopters which are small in comparison.

Fluctuation of personnel that was involved in the programming is a critical aspect in general. It seems as an inevitable characteristic in programming that no matter how detailed one writes a report or comments the code, it is very difficult for new personnel to fully comprehend the logic understanding of the predecessor who started to write the code. Even “simple” problems like different naming conventions for data structures, such as parameters, lists, dictionaries, etc. can lead to an erroneous execution of the code. In the worst case, the code does not throw an error but still computes something wrong, or at least something the programmer does not want it to do.

Different programming skills and different programming approaches also led to divergence of the initially closely related tools. These circumstances are not owed to intentional misconduct but to the commitment to reach the individual objectives as prescribed in the different projects, and can be seen as a general problem in the engineering branch nowadays.

Moreover, tough schedules contributed to an excessive use of the *try-except* rules in Python to cater for a fast solution working for the corresponding tool only. Project deadlines can be matched indeed with this approach but the logical architecture of the global programming approach starts to crack.

Using repositories for software development is a good idea in general. However, since it was not possible to use each tool for the generation of each type of air- and rotorcraft, common functions (for air- and rotorcraft) produced slightly different results, even though they worked perfectly for each individual type. As an example, an arbitrary function was thought to produce the correct results but weeks later it was discovered that the function did not work well for both types.

Due to a lack of testing routines errors appeared sometimes weeks after the responsible software code was committed to the repository. The error was not detected at the right time since the code was not tested for all possible air- and rotorcraft. The task to find an error weeks after it has been programmed can turn into a time-consuming and frustrating challenge.

Most engineers, regardless whether from academia or from industry, will be familiar with – at least – one of those aforementioned problems.

To resolve these issues, it was decided at the Institute of Structures and Design in 2016 to create a "lessons learned" list in order to develop a new software framework from scratch unifying the individual functionalities of the previous tools.

3. PANDORA as open-source approach

PANDORA (Parametric Numerical Design and Optimization Routines for Aircraft, [15]) is intended to be one tool that features different modules (which resume the functions of the previous tools). The key idea behind PANDORA is to generate one common framework in order to model, analyze, and size both fixed- and rotary wing fuselages. It was decided to use Python as programming language because of its simple and logic design. Moreover, Python is largely spread in the research community and features many modules for scientific work. Particular focus in the development of PANDORA lies in the use of dedicated open-source packages and the interchangeability of different commercial and open-source FE solvers. In order not to make the same mistakes again, testing routines were integrated into PANDORA from the very first beginning. The idea behind software testing was to realize mistakes as early as possible. Ultimate objective with testing is to set-up test methods for automated testing during night.

A schematic overview of the functionalities in PANDORA is given in Fig. 7 and the modules will be introduced in the following sub-sections. The user can access PANDORA either via batch mode to conduct structural analyses or via a graphical user interface (GUI). During pre-processing models can be generated using an external CPACS file or a model that has already been generated outside PANDORA (note that the model has to be available in an ASCII file of the supported FE solver). The FE converter is used to write the correct file format to either start a single analysis or a sizing process. For FE computation the model is send to the solver and as soon as the computation has finished and the results have been written in ASCII format it is returned to PANDORA where the converter reads the results for subsequent post-processing in the GUI.

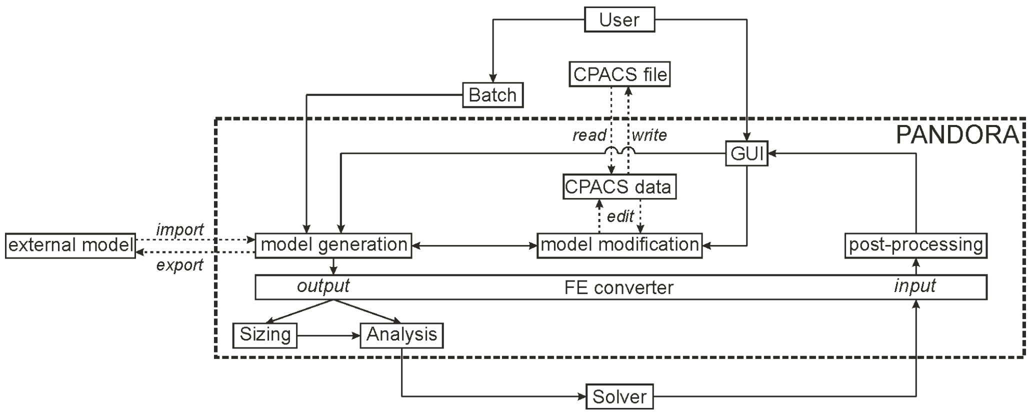

Figure 7: PANDORA scheme

3.1. Graphical User Interface

A GUI was integrated in PANDORA to allow easy access and handling for colleagues without major programming experience. The circumstance that the engineer actually sees what he (or she) is doing leads to a faster and larger acceptance of the tool. A wider spread (and thus use of a) software tool allows more and thus better feedback to the programmers which in turn leads to improvements and therefore, a better performance.

It was mentioned earlier that CPACS serves as central data model for all predesign activities at the Institute of Structures and Design. Therefore, the integration of a CPACS viewer to the GUI was considered as an essential feature of PANDORA. CPACS data can be read and edited in the GUI with a direct visual impact on the edited data. Figure 8 shows the CPACS data of the frames of an aircraft fuselage and the full aircraft geometry.

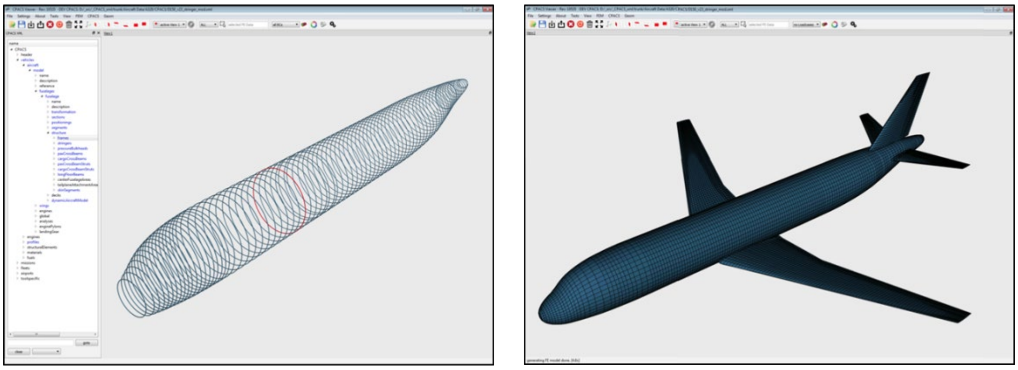

Figure 8: CPACS fuselage data visualization and aircraft model in PANDORA

3.2. FE converter

One of the main features of PANDORA is an in-house developed FE converter. The intention of the converter was to create a *convert-everything-to-everything* tool in order to be able to become less independent from one single FE solver, as schematically shown in Fig. 9.

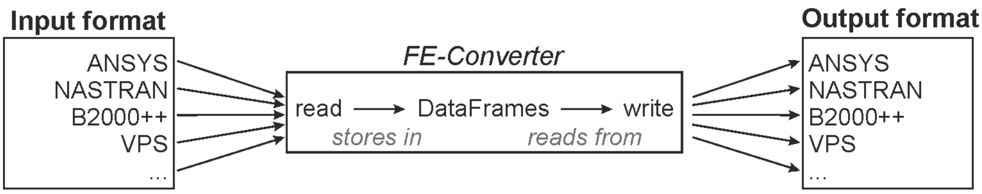

Figure 9: FE converter scheme

The FE converter allows to read and to write input files in ASCII[2] format for ANSYS [16], NASTRAN [17], and B2000++ [18]. Currently the FE solvers VPS [19], ABAQUS [20], and LS-DYNA [21] are being integrated. The FE converter is able to read and write:

- Nodes
- Elements (shells, beams, point elements)
- Materials (isotropic, orthotropic)
- Properties (thickness, layered stack-ups, engineering constants)
- DOF coupling elements
- Coordinate systems
- External loads (forces, moments, pressure, accelerations)
- Constraints

[2] American Standard Code for Information Interchange

The converter is written in Python and uses the *pandas* module [22]. The FE data within PANDORA is stored in so-called *pandas DataFrames* in a format heavily based on the NASTRAN format. In addition, data can easily be stored in the HDF5 format.

Figure 10 shows a generic utility transport helicopter (UTH) imported into the PANDORA pre-processor module. The model has been generated in ANSYS using ROFUMA: During the model generation process a .cdb file in the ASCII format has been written with all the relevant data for subsequent computation. The model features shell elements (blue) for the fuselage skin and the frames. Beam elements are used for the stringers (red). Single masses (such as systems, or additional structure like for instance the alighting gear) are discretized as lumped masses (red) being coupled to user-specified areas to avoid large local stress concentrations. The model features rotorcraft typical modeling aspects as implemented in ROFUMA, such as cut-outs and stage modeling at the tail boom.

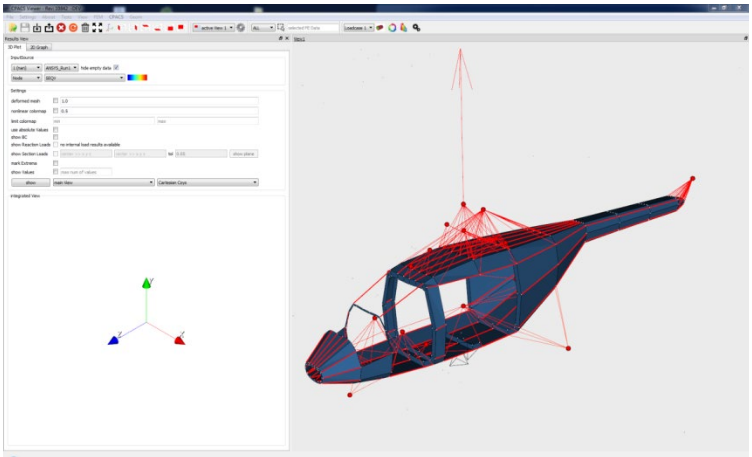

Figure 10: Imported rotorcraft model in the PANDORA GUI

3.3. Model Generation

In order to dispense the use of the earlier tools TRAFUMO and ROFUMA, it was necessary to integrate a module into PANDORA to generate fuselage models. For this purpose the Python OCC (Open Cascade, [23]) geometry kernel was integrated and is used to generate the fuselage geometry for the subsequent meshing with shell elements. The surface comprises several individual profiles which are generated by applying B-Splines to the corresponding profile points. Aircraft frames and stringers as well as floor structure consisting of crossbeams and struts are discretized using beam elements.

Simultaneously, additional data from the CPACS file is read (such as materials, profiles, etc.) using the *lxml* package for Python and stored in *pandas*

DataFrames, so that the input cards for the FE model can be generated automatically using the write-functions of the FE converter.

An aircraft airframe is shown in Fig. 11 in the PANDORA pre-processor module. The airframe has been generated in PANDORA using CPACS fuselage data as described in [24].

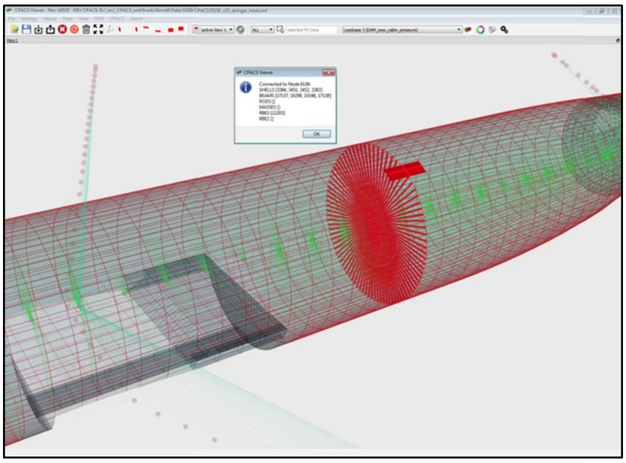

Figure 11: Aircraft airframe in the PANDORA pre-processor

Currently, the model generation module does not allow the aforementioned rotorcraft specific features. Therefore, rotorcraft models have to be imported after generation with ROFUMA.

3.4. Exemplary analysis of a rotorcraft airframe

The equivalent stress of a generic utility and transport helicopter (UTH) during hovering calculated using ROFUMA is shown in Fig. 12.

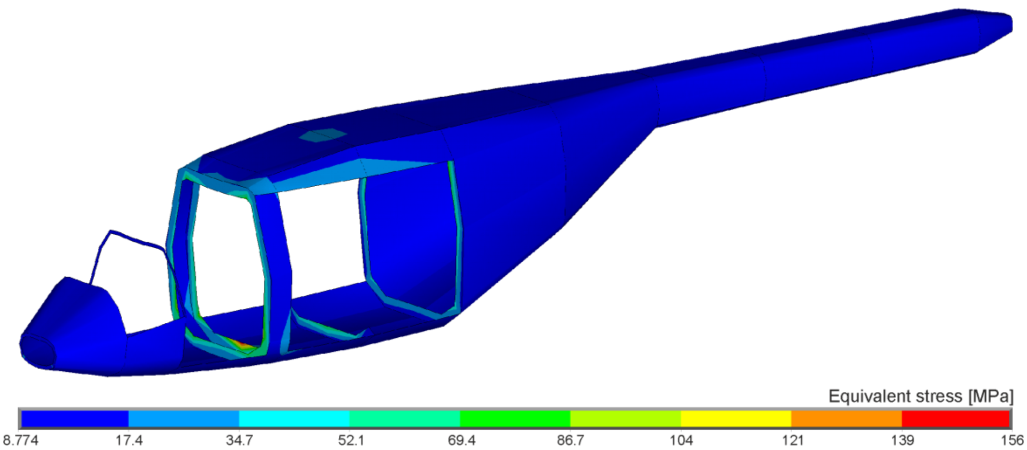

Figure 12: Equivalent stress of an UTH during hover (ROFUMA / ANSYS)

The ROFUMA model was exported into the .cdb[3] format during the model generation process, imported via the FE converter into PANDORA and calculated using the ANSYS solver. The resulting equivalent stress distribution of the computation is shown in Fig. 13. A good agreement can be observed with minor deviations caused by the different color scheme on the one hand and a slightly different description of the beam properties. The beam sections within ROFUMA are calculated according to the profile description in CPACS internally in ANSYS while the beam properties within PANDORA are calculated in Python and reduced to the engineering constants (area, center of gravity, moments of inertia). This slight simplification is accepted due to the detail level of the preliminary design stage and due to the benefit of reduced computational time.

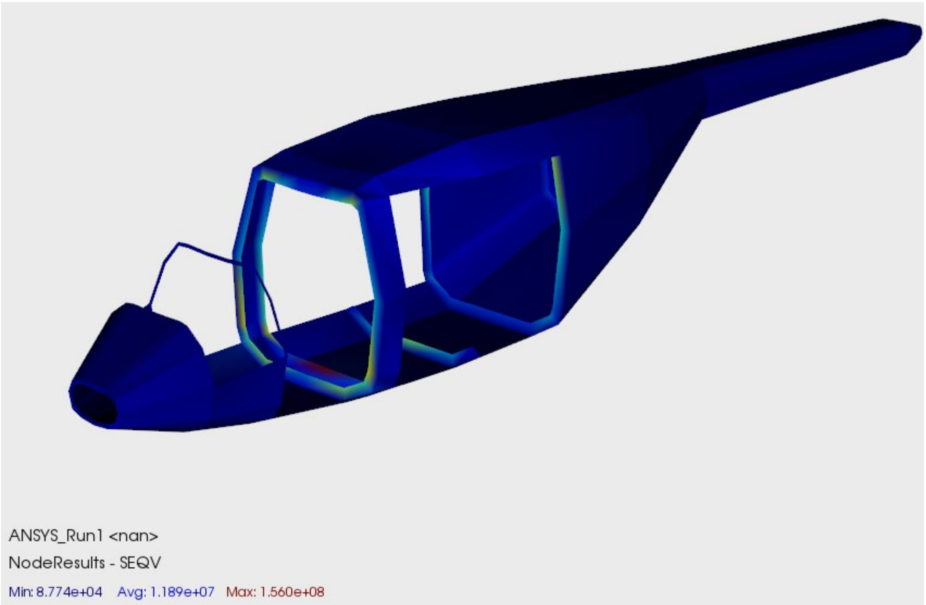

Figure 13: Equivalent stress of an UTH during hover (PANDORA / ANSYS)

3.5. Sizing validation

The sizing module *fe_sizer* is currently under development allowing structural sizing against maximum strength, stability, and fatigue. Stability is evaluated for pure longitudinal and shear compression as well as combined loading as proposed by Bruhn and Niu [25–27].

For validation purposes a barrel is used due to its high resemblance to aircraft fuselages (Fig. 14). The model consists of 180 circumferentially equally distributed stringers and 18 frames that are equally distributed along the

[3] Coded Database

longitudinal axis. The stringers and frames are modelled using beam elements while the fuselage skin is modelled using shell elements. The radius of the barrel is r = 4.0 m and the length is l = 10.2 m. At the one end all nodes are fixed in their translational and rotational degrees of freedom (DOFs). Load is introduced at a single node located in the center of the barrel at its opposite end. This node is connected to the barrel with a rigid body to avoid any deformation of the barrel edge. A ring of 180 shell elements is used for a detailed comparison of the results. The location was chosen to be distant enough from the clamping to avoid any significant influences. The material used for the validation is an isotropic aluminum alloy, typically used in current metallic aircraft.

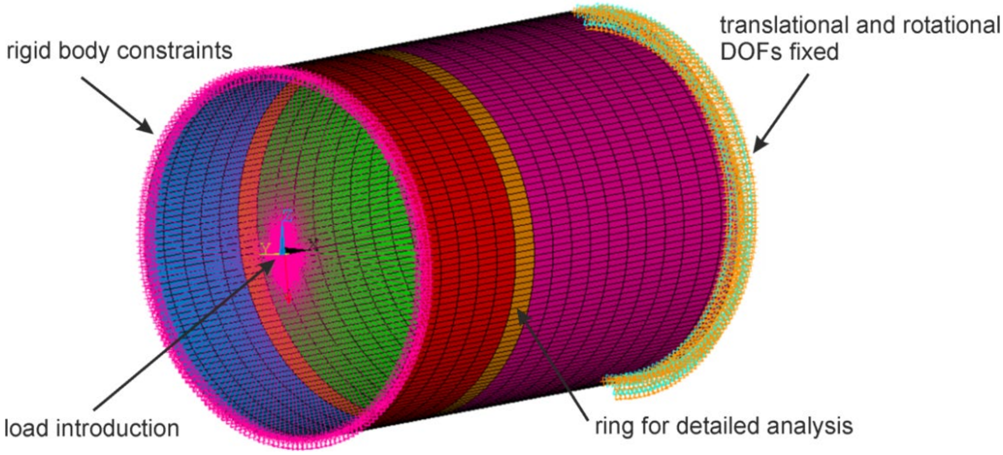

Figure 14: Validation barrel

The validation process comprised two parts: First, it had to be ensured that a single static analysis with PANDORA produces the same results as using S-BOT+. Therefore, several load combinations of forces and moments introduced at the load introduction node were applied and the results compared. Exemplary, the longitudinal stress, the shear stress, and the equivalent stress in the ring (see Fig. 14) for a load case with a single lateral force, a bending moment and a torsional moment are shown in Fig. 15. It can be observed that the results show a very good agreement, so that the first validation step could be concluded successfully. This circumstance also lets conclude that the FE converter works satisfactorily for ANSYS. The computations were compared to an analytical tool. However, as the analytical tool does not update the position of the neutral axis as iterative FE solutions do, a comparison with the analytical tool for a sizing process is merely suited to the first iteration and will therefore be relinquished in this paper.

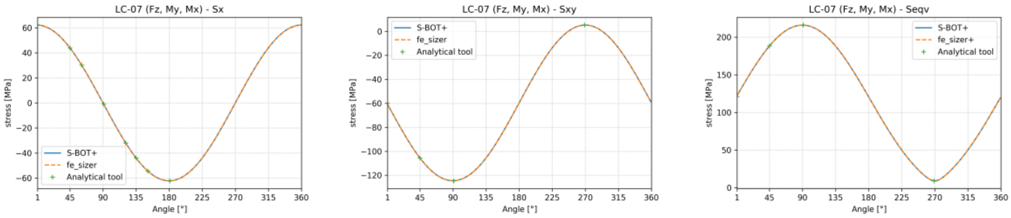

Figure 15: Basic validation results (Fz, My, Mx) with the ANSYS solver

Since it is intended to use several FE solvers within PANDORA, it was necessary to compare other solvers to each other. Table 1 shows a comparison of the three solvers ANSYS, NASTRAN, and B2000++ for the same load case that was used for the computations shown in Fig. 15. The results that were compared are the minimum and maximum displacement of the global model as well as the vertical displacement and the rotations about the x- and the y- axis for the node that is used for the load introduction (see Fig. 14).

Table 1: Comparison of different solvers

Maximum displacement	**ANSYS**	**NASTRAN**	**B2000++**
u_x [mm]	-11.72 / 11.72	-11.72 / 11.72	-11.70 / 11.70
u_y [mm]	-22.27 / 22.27	-22.30 / 22.30	-22.28 / 22.28
u_z [mm]	-64.79 / 0.0	-64.84 / 0.0	-64.75 / 0.0
Nodal displacement			
u_z [mm]	-42.518	-42.540	-42.473
rot_x	.55684e-02	.55740e-02	.55688e-02
rot_y	-.29306e-02	-.29295e-02	-.29262e-02

It can be seen that the three presented solvers show a good agreement to each other. As reverse conclusion it can be stated that the FE converter also works well with NASTRAN and B2000++.

Figure 16 and 17 show the final thicknesses of the validation barrel after sizing against maximum strength and stability under a combined loading of F_z, M_y, and M_x. Convergence with *fe_sizer* was reached after 15 iterations. It can be observed that – globally seen - both sizing processes show a good agreement of the thickness contours. It shall be mentioned that deviations in the thickness color plot results from a slightly different color scheme scaling on the one hand and on the smeared visualization of the PANDORA model and the elementwise visualization of the S-BOT+ model on the other hand. The thicknesses are given in [mm]. Both models show a lower boundary for the thicknesses of about $t_{min} \approx 1.0$ mm and a maximum thickness of about $t_{max} \approx 5.1$ [mm]. The average shell thickness calculated with *fe_sizer* is $t_{avg} \approx 3.1$ [mm].

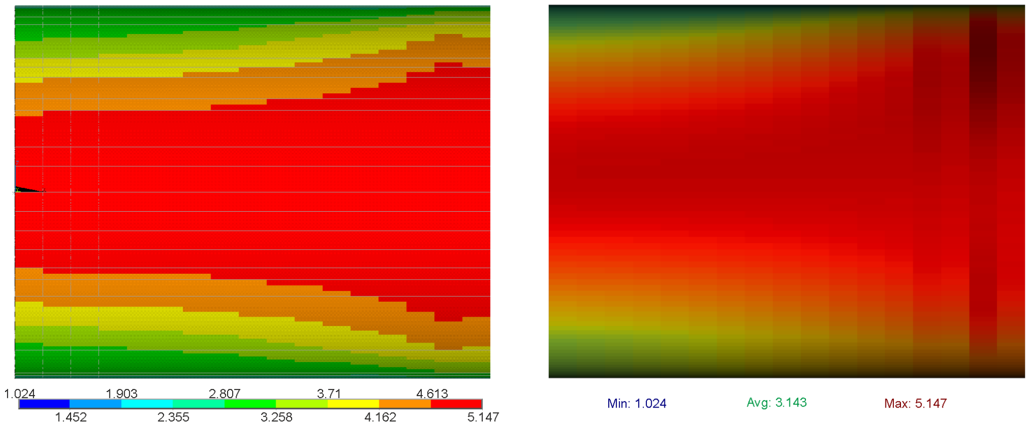

Figure 16: Validation barrel after sizing (F_z, M_y, M_x) – side view (left: S-BOT+, right fe_sizer)

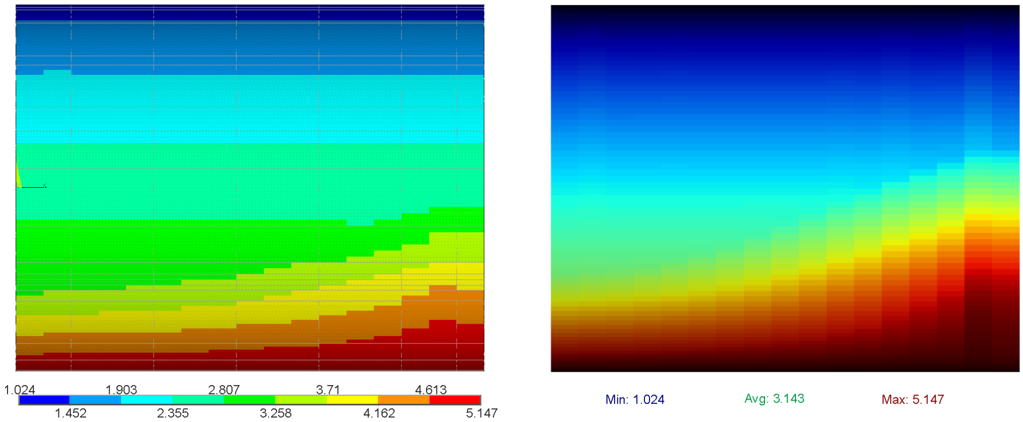

Figure 17: Validation barrel after sizing (Fz, My, Mx) –top view (left: S-BOT+, right fe_sizer)

The thickness distribution within the ring is shown in Fig. 18. The dashed orange curve shows the thickness distribution in the ring sized with *fe_sizer* while the blue curve shows the thickness distribution of the ring calculated with S-BOT+. A good agreement between both tools can be observed.

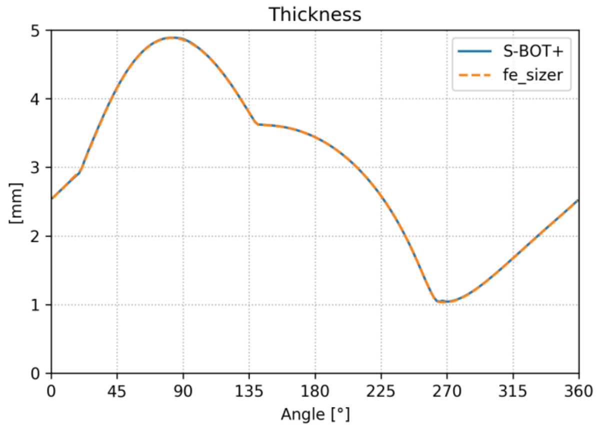

Figure 18: Thickness distribution within the ring after 15 iterations

The safety factors for the individual criteria are shown in Fig. 19. A safety factor (SF) of SF = 1 means, that the element is fully loaded and no reserves respectively margin of safety (MS) is left. The relation between the safety factor and the margin of safety is defined as

$$MS = SF - 1. \tag{5}$$

The presented results in Fig. 18 and 19 already feature a safety factor of SF = 1.5 which is typical for aeronautical vehicles. It has been included in the material limits, i.e. the yield stress σ_{yield} has been reduced according to

$$\sigma_{yield} = \frac{\sigma_{yield,original}}{1.5}. \tag{6}$$

Therefore, a safety factor of SF = 1 already includes a safety of 50% against failure and the structure can be considered as optimum designed for the given load case.

It can be observed from Fig. 19, that the barrel is partly sized against the maximum strength criterion (orange curve). This criterion shows responsible for angles in the range of about 20° < α < 135°. In this area, the structure is loaded in tension, so that buckling is not a relevant criterion. In the range of about 135° < α < 20, the critical load condition is stability against combined loading of longitudinal compression and shear (denoted by the blue line).

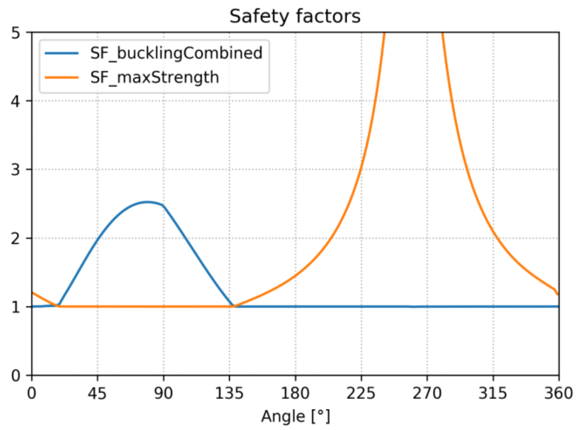

Figure 19: Safety factors within the ring after 15 iterations

The safety factor against maximum strength SF_{ms} is calculated using the equivalent stress compared to the yield strength:

$$SF_{ms} = \frac{\sigma_{yield}}{\sigma_{eqv}} \tag{7}$$

Stability against combined loading (shear and compression) is calculated as

$$R_s^2 + R_c \leq 1.0 \tag{8}$$

as shown in Fig. 20 with the stress ratios

$$R_s = \frac{\tau_{xy}}{\tau_{critical}} \tag{9}$$

and

$$R_c = \frac{\sigma_x}{\sigma_{critical}}. \tag{10}$$

Figure 20 shows the interaction curve for longitudinal compression and shear loading. The blue line denotes MS = 0. The point B lies on the critical line, i.e. the stress combination in B leads to buckling of the panel. Point A is within the boundaries and thus features a margin of safety against combined loading. The distance between the points A and B quantitatively describes the margin of safety.

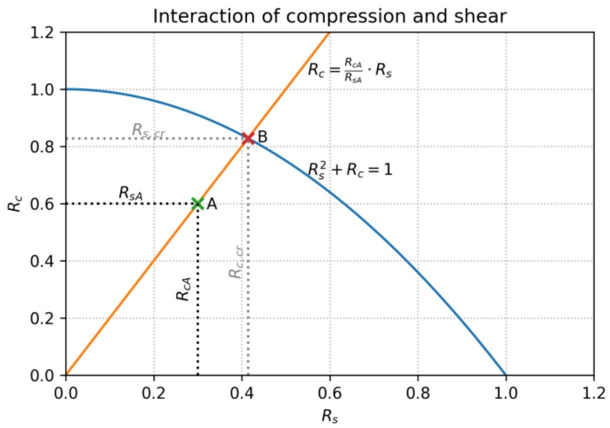

Figure 20: Interaction curve for combined shear and compression loading

3.6. Post-Processing

In order to facilitate the handling of PANDORA for the user during daily work, a post-processor based on the *Visualization toolkit* (VTK [28]) was added to PANDORA to relieve the user from switching between a multitude of different programs. VTK allows the visualization of 3D data. Thus, the user can stay within PANDORA and directly view the results of an FE computation without manually transferring data. Figure 21 shows the validation barrel after the sizing process. On the right side the thickness distribution is displayed while the graph on the left shows a 3D-graph of the thickness over the iterations and (circumferential) angle of the barrel.

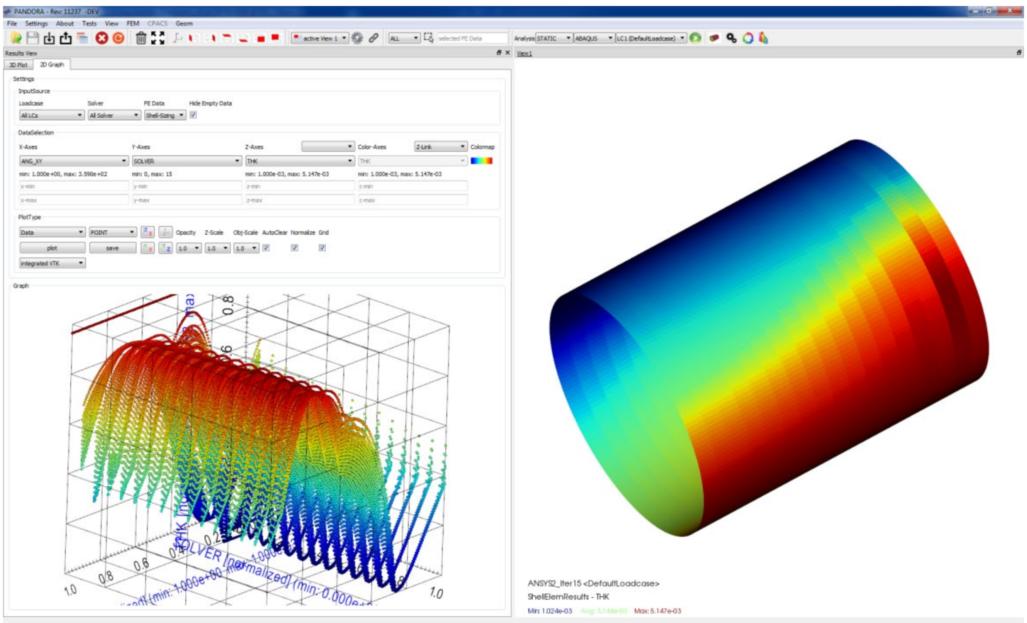

Figure 21: Exemplary sizing of the barrel using the PANDORA post-processor

4. Summary and outlook

In this paper the software framework PANDORA was presented. PANDORA is a software framework based on open-source codes. It allows the pre-processing of air- and rotorcraft fuselage FE models based on the CPACS data model. Due to an internal FE converter it is possible to calculate a fuselage model with different solvers, commercial but also open-source ones. A post-processing module allows the subsequent visualization and assessment of the results.

Several applications of PANDORA were presented in this paper including air- and rotorcraft fuselage modelling and analysis. Sizing methods against maximum strength and stability have been introduced showing parts of the validation process.

The use of CPACS, RCE, and PANDORA marks a powerful combination for structural analysis and offers new potential partners an easy to handle entry for collaboration. Additionally, partners may profit from using the solver of their choice without having to get familiar with new solvers.

Current work on PANDORA is conducted with regard to a finer discretization of the structure as implemented in AC-CRASH for emergency crash analysis, such as ditching for instance. Additionally, it is planned to enhance the model generation with the rotorcraft specific features of stage modelling so that rotorcraft fuselage models can be directly generated within PANDORA. Another focus lies in the sizing of composite structures, e.g. altering of ply lay-ups (number of plies and their orientation) and sizing of structures that are discretized with beam elements. Moreover, the aforementioned simplification of reducing beam profiles to engineering constants will be resolved: The geometry definition will be deposited in *pandas* so that the reduction to engineering constants will be available only for the FE Codes that do not feature a geometric profile definition.

5. References

A. J. Ruddell, *Advancing Blade Concept (ABC™) Development,* Journal of the American Helicopter Society 22 (1977), pp. 13-23 .

A. Bagai, *Aerodynamic Design of the X2 Technology Demonstrator Main Rotor Blade*, in *American Helicopter Society 64th Annual Forum*, Montréal, Canada, 2008.

P. Weiand, M. Schmid, M. Buchwald und D. B. Schwinn, *A distributed design environment for rotorcraft*, in *Deutscher Luft- und Raumfahrt Kongress*, Friedrichshafen, Germany, 2018.

C. M. Liersch und M. Hepperle, *A distributed toolbox for multidisciplinary preliminary aircraft design*, CEAS Aeronautical Journal

2 (2011), pp. 57-68.

B. Nagel, D. Böhnke, V. Gollnick, P. Schmollgruber, A. Rizzi, G. La Rocca und J. J. Alonso, *Communication in Aircraft Design: Can We Establish a Common Language?*, in *28th International Congress of the Aeronautical Sciences*, Brisbane, Australia, 2012.

D. Seider, M. Litz, A. Schreiber und A. Gerndt, *Open Source Software Framework for Applications in Aeronautics and Space*, in *IEEE Aerospace Conference*, Big Sky, MT, USA, 2012.

D. P. Raymer, *Aircraft Design - A Conceptual Approach*, 4th ed., American Institute of Aeronautics and Astronautics, 2006.

J. Scherer, D. Kohlgrüber, F. Dorbath und M. Sorour, *A Finite Element Based Tool Chain for Structural Sizing of Transport Aircraft in Preliminary Aircraft Design*, in *Deutscher Luft- und Raumfahrtkongress*, Stuttgart, Germany, 2013.

D. B. Schwinn, P. Weiand und M. Schmid, *Structural Analysis of a Rotorcraft Fuselage in a Multidisciplinary Environment*, in *NAFEMS World Congress*, Stockholm, Sweden, 2017.

D. B. Schwinn, *Parametrised fuselage modelling to evaluate aircraft crash behaviour in early design stages*, International Journal of Crashworthiness 20 (2015), pp. 413-430.

B. Nagel, M. Kintscher und T. Streit, *Active and Passive Structural Measures For Aeroelastic Winglet Design*, in *26th International Congress of the Aeronautical Sciences*, Anchorage, AK, USA, 2008.

D. B. Schwinn, *Applied parametrized and automated airframe modeling methods in the preliminary design phase*, International Journal of Modeling, Simulation, and Scientific Computing 6 (2015), p. 1550037.

E. W. Hunter, *A Process to Enable High Fidelity Airframe Sizing and Optimization for Conceptual Design,* in *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Schaumburg, IL, USA, 2008.

D. B. Schwinn, P. Weiand, M. Buchwald und M. Schmid, *Rotorcraft fuselage weight assessment in early design stages*, in *Deutscher Luft- und Raumfahrtkongress*, Friedrichshafen, Germany, 2018.

M. Petsch, D. Kohlgrüber und J. Heubischl, *PANDORA – A python based framework for modelling and structural sizing of transport aircraft*, in *8th EASN-CEAS International Workshop*, Glasgow, Scotland, 2018.

ANSYS (homepage), *https://www.ansys.com*. [Accessed 15th Feb 2019].

NASTRAN (homepage), *http://www.mscsoftware.com/product/msc-nastran*. [Accessed 15th Feb 2019].

B2000++ (homepage), *https://www.smr.ch/products/b2000*. [Accessed 15th Feb 2019].

VPS (homepage), *https://www.esi-group.com/software-solutions/virtual-performance/virtual-performance-solution*. [Accessed 15th Feb 2019].

ABAQUS (homepage), *https://www.3ds.com/products-services/simulia/products/abaqus*. [Accessed 15th Feb 2019].

LS-DYNA (homepage), *http://www.lstc.com/products/ls-dyna*. [Accessed 15th Feb 2019].

pandas (Python Data Analysis Library) (homepage), *https://pandas.pydata.org*. [Accessed 15th Feb 2019].

pythonOCC - 3D CAD for python (homepage), *http://www.pythonocc.org*. [Accessed 15th Feb 2019].

J. Scherer und D. Kohlgrüber, *Fuselage structures within the CPACS data format*, Aircraft Engineering and Aerospace Technology 88 (2016), pp. 294-302.

E. F. Bruhn, *Analysis and Design of Flight Vehicle Structures*, Tri-State Offset Company, 1973.

M. C.-Y. Niu, *Airframe Structural Design*, 2nd ed., Hong Kong Conmilit Press Ltd., 1999.

M. C.-Y. Niu, *Airframe Stress Analysis and Sizing*, 2nd ed., Hong Kong Conmilit Press Ltd., 1999.

VTK (homepage), *https://pypi.org/project/vtk*. [Accessed 15th Feb 2019].