Knowledge-based automatic Airframe Design using CPACS

J.-N. Walther
German Aerospace Center (DLR), Air Transportation Systems (LY)
Research Associate
Blohmstraße 20, 21079 Hamburg, Germany
jan-niclas.walther@dlr.de

P. D. Ciampa
German Aerospace Center (DLR), Air Transportation Systems (LY)
Research Associate and Team Lead MDO Group

ABSTRACT
The CPACS data format [1, 2] has long been established as the primary means of data exchange in preliminary aircraft design projects within DLR. As described by Scherer et al. [3], it contains a wide range of options for describing the structural layout of a design including frames and stringers, floors, bulkheads, etc. Based on these descriptions, several finite element model generators comparable to the one described by Walther et al. [4] have been implemented, which can provide detailed computational structural models of a given design. However, all model generators require the information on the structural layout to be available in CPACS upfront.
Within a larger aircraft design context, this necessitates the augmentation of the description of the structure to a given plain aircraft geometry. So far, this has been accomplished through a manual process, which not only results in an increased risk of errors, but also prohibits the exposure of parameters to a larger multidisciplinary optimization.
In the presented paper, a newly developed knowledge-based airframe augmentation module will be introduced. Implemented in Python, it provides methods to automatically initialize a full structural layout on a given CPACS geometry, based on a manageable number of control parameters. In addition to an outline of the governing design rules, several application cases will also be given.

KEYWORDS: CPACS; preliminary aircraft design; fuselage structures; knowledge based design

1 INTRODUCTION

Over the past years, the CPACS (Common Parametric Aircraft Configuration Schema) data format [1, 2] developed at the German Aerospace Center (DLR) has enjoyed growing popularity as a means of data exchange in collaborative aircraft design processes. It enables the integration of knowledge across disciplines, as well as levels of detail ranging from simple statistical methods to computationally expensive high-fidelity methods involving advanced computational fluid dynamics (CFD) and computational structure mechanics (CSM) analyses.
In most cases, the transition from a lower to a higher level of detail also requires more detailed information on the product to be available in order to obtain a reliable solution. This also applies in the field of structure mechanics, where global finite element models (GFEM), i.e. computational models of the primary structure, where the entire airframe is represented using beam and shell elements, play a fundamental role in the mid- to high-fidelity stages of the preliminary design process.
Consequently, several CPACS-based GFEM model generators have been implemented:

- TraFuMo (DLR Institute of Structures and Design) [4, 5]: Fuselage
- DELIS (DLR Institute of Composite Structures and Adaptive Systems) [6]: Wings
- ELWIS (DLR Air Transportation Systems) [7]: Wings
- PyGFEM (Polytecnico Milano) [8]: Overall aircraft
- Descartes (Airbus Defense and Space) [9]: Overall aircraft

Despite their many capabilities, all of the above tools require information on the structural layout to be readily available in CPACS at the beginning of the model generation, in order to produce a proper GFEM model including reinforcements and structural parts. During development, this information has often been augmented by hand to models provided by lower level tools, which then served as specific test cases for the implementation of GFEM generation algorithms.

This procedure is, however, not suited for a production environment, where results from lower level tools must be analyzed or potentially optimized within automated MDO workflows, which are being setup within multiple projects at DLR [10, 11, 12]. To partly bridge this gap, Scherer et al. [3] introduced an automatic tool named F-DESIGN, which could automatically adjust a design's mainframe position to match changes in the wing structure. It was also proposed to implement knowledge based design rules to generate a structural layout from scratch.
Based on this idea, a structural augmentation module has now been developed at the DLR Institute of Air Transportation Systems. It aims to provide a simple yet powerful interface for adding an initial structural layout to a given CPACS dataset, based on a manageable set of parameters.

2 STRUCTURE DEFINITION IN CPACS

The CPACS data format provides descriptions for a number of structural elements for both the fuselage and wing structure, which have been collected by Scherer et al. [3]. This section will give a short overview of the key concepts.

2.1 Fuselage Structure Components

A typical layout of a transport aircraft fuselage as described by Niu [13] is a thin-walled structure, which consists of skin panels reinforced by frames in circumferential direction and stringers in longitudinal direction. The pressurized fuselage section is capped by two pressure bulkheads. The floors consist of crossbeams and support posts, which are attached to the reinforcements. They may also contain further longitudinal beams.
In CPACS, this building technique is mirrored by the fuselage/structure branch. Sub-branches exist for frames and stringers, but also floor component definitions, bulkhead placement and skin segmentation. Some further elements such as center fuselage areas and tail plane attachment areas exist, but are beyond the scope of this work.
In the description scheme, frame and stringer definitions are of particular importance, since they define a topologically two-dimensional grid on the fuselage, which serves as a reference for the placement of all other components. Both are created by extruding a structural profile along a given path, which is in turn defined by the intersection of the fuselage surface and a definition plane.
CPACS defines such planes using lists of stringerFramePositionType XML elements, which are essentially definitions of vectors in space. The vector origin is given by the positionX, referenceY and referenceZ parameters, all of which are defined in the global coordinate system of the aircraft. The orientation is given via the referenceAngle parameter, which is always measured around the aircraft's longitudinal axis, starting at the top. A simple example for a stringer/frame position definition at a given position along the x-axis is shown in Fig. 1.

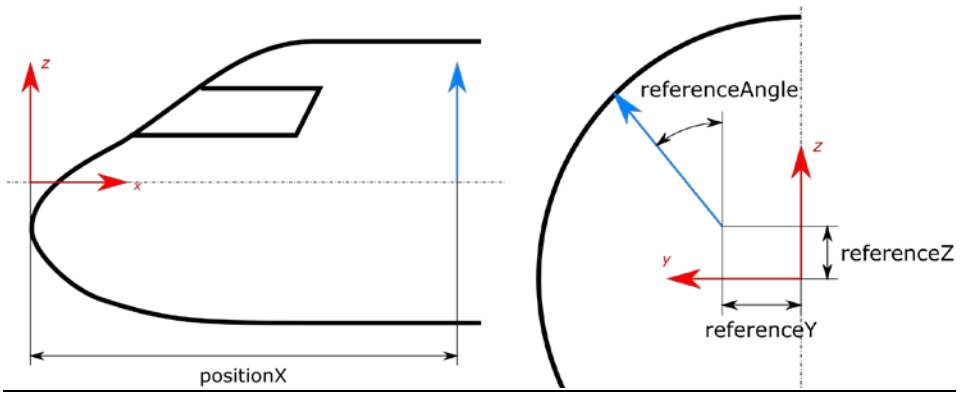

Figure 1: Stringer/Frame positioning parameters in CPACS

Based on a set of position definitions, planes can be extruded. For frames, it is assumed that the extrusion is performed in circumferential direction of the fuselage. Thus, one positioning element is sufficient for defining a plane orthogonal to the longitudinal axis. However, more positions may be given if more complex intersection planes must be defined.
For stringers, there is no implicit extrusion direction in the original CPACS specification. Therefore, at least two positions must be given. The interpolation between them is controlled by the continuity

parameter, which can be set to zero for linear interpolation or two for a smooth interpolation with continuous curvature. In practice, it may, however, be necessary to deviate from the original assumptions for stringer extrusion. A common condition for the stringer extrusion paths is for them to lie on a flat plane along the longitudinal axis for production reasons. This can only be realized by introducing an additional continuity -1, implying a piecewise constant interpolation scheme. In this case an extrusion path along the longitudinal axis is assumed. The different extrusion approaches are illustrated for a simple example in Fig. 2.

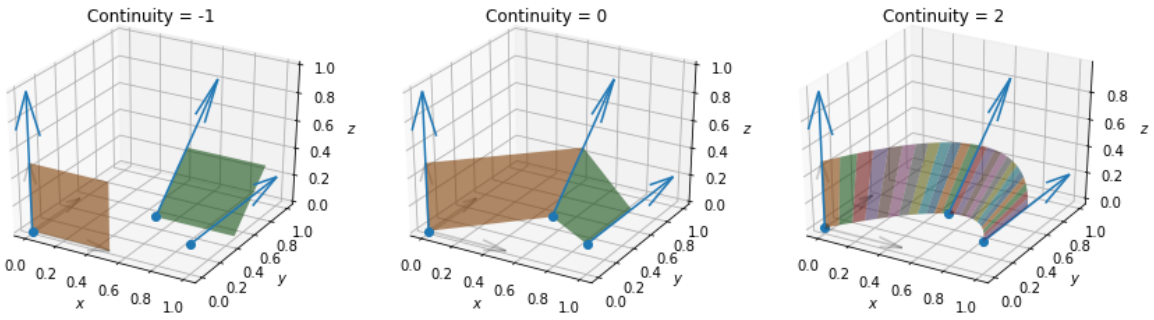

Figure 2: Effect of continuity parameter on stringer plane extrusion

Unique identifiers (uIDs), which enable references from one object to another, play a central role in CPACS. Therefore, any frame or stringer is assigned its own uID. Among other things, the presence of uIDs is required due to the bulkhead definition. Modeling bulkheads is essential to ensure an airtight model if the pressurization of the fuselage is meant to be considered in the structural analysis. The definition in the fuselage structure branch is implemented using two uIDs. The first refers to the frame, where the bulkhead will be placed, while the second points to a more detailed description of the bulkhead geometry and its properties in the structural elements branch. In the scope of the augmentation the second part is, however, neglected and only a flat default bulkhead element is considered.

2.2 Wing Structure Components

Like the fuselage, a typical transport aircraft wing as described by Niu [13] is also a lightweight structure consisting of reinforced skin panels. Nonetheless, there are slight differences in topology and terminology. The wing is connected to the fuselage by at least two spars, which run from the root all the way to the tip. The spars among themselves are connected by ribs, forming a grid. The skin covers the upper and lower side of the grid. It is typically reinforced by stringers to avoid skin buckling.

In CPACS, the definition of the wing structure follows a different paradigm compared to the fuselage structure. Instead of using global coordinate system, a local two-dimensional coordinate system on the wing chord surface is employed. The plane is parametrized from 0 to 1 in span- and chordwise direction by the η- and ξ-coordinates respectively. Then, planes in the global coordinate frame are created based on the normal vector of the η-ξ-plane at each position.

Within the scope of this work, only the spar placement will be considered.

2.3 Floor Components

Following a typical fuselage structure design as described by Niu [13], floors in CPACS consist of three types of structural components. The defining components of all floors are the crossbeams which are extruded within frame planes along the global y-axis. They are described by a frame uID and a z-position. Further parameters are a reference to the structural element defining the cross-section of the crossbeam and some further alignment parameters to modify the positioning of the cross-section. Crossbeams are the only element in the floor definition that is required under all circumstances. All further elements are optional. During the generation of the GFEM model, the z-position cannot be strictly enforced, since the crossbeams have to be connected to a stringer/frame intersection point, due to the GFEM topology. These do not necessarily coincide with the desired z-position, so the closest intersection point will be picked at each frame instead.

In addition to the crossbeams, longitudinal floor beams may be defined as well. They run perpendicular to the crossbeams along the length of the fuselage and are defined by a succession of at least two longFloorBeamPosition elements. Each position is composed of a cross beam uID and a y-coordinate defining a point on the crossbeam. Furthermore, a continuity parameter comparable to the stringer definition describes the behavior of the function between positions. Like the crossbeam elements, longitudinal beams also possess parameters defining the cross-section.
Finally, floor support posts (labeled crossbeam struts in CPACS) can also be defined. As for longitudinal beam positions, a crossbeam uID and a y-position specify a point on the cross beam. An additional angle in the global yz-plane is also given to describe the extrusion direction. The parameters at a given frame position are illustrated in Fig. 3.

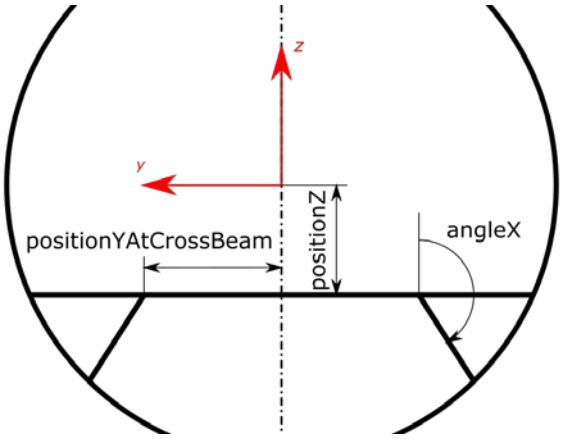

Figure 3: Floor description at a given frame

Note that in its current version (2.3) CPACS differentiates between passenger and cargo floors for crossbeams and crossbeam struts. However, the underlying data structure in both cases is the same. Furthermore, CPACS provides a branch for floor panels, which is not taken into account, since they usually do not carry significant loads.

3 FUSELAGE DESIGN RULES

The structural augmentation is driven by a simplified structural description stored in a newly developed tool-specific branch in the CPACS data format. It reduces the description of the structural layout to a few parameters, which serve as inputs to the underlying knowledge-based design rules employed. The design rules and the corresponding parameters will be described in detail in the following.

3.1 Supplementary Source File

Aside from the parameters given explicitly in the description, the model augmentation requires further information, which is not directly dependent on the present aircraft geometry. For instance, the structural element definitions need not be recreated from scratch for each design and can instead be considered semi-finished products. As such, individual profiles may be re-used same across airplane designs. Therefore, these inputs are taken from a supplementary source file, the path to which is given as an additional augmentation parameter.

3.2 Frame Placement

The frame placement constitutes the basis of the structural augmentation of the fuselage. It is thus dependent on a number of parameters concerning other elements. Therefore, the general placement procedure consists of two steps: data assembly and interpolation.
In the first step, data on the positioning of the main frames is assembled, not only from the augmentation input values, but also from other existing structural data in the CPACS data set.
Currently, main frames are positioned at

- Start/end of structural range
- Bulkhead positions

- Wing/tail plane spar positions at $\xi = 0$
- Start/end of floor range

The respective data items are merged in the given order, with the existing data taking precedence over the newly added. This means, that if a new data point is within a certain tolerance of another data point, no additional main frame position will be added. As a simplification, all frame planes are assumed to be perpendicular to the longitudinal axis, which makes each positioning a single value along the x-coordinate.

In the simplified structural description, frame positions can be given in three ways: as an absolute value along the coordinate axis, as a relative position along the fuselage, where 0 is the nose and 1 the tail, or as an offset value from the respective boundary. The positioning scheme used is specified by the augmentationPositionType argument, which can be passed along with any positioning node.

With the main frames in place, the remaining frame positions are computed by interpolating the distances between the main frames. The interpolation is bounded by two parameters, the frame reference spacing Δx_{ref} and the maximum number of frames $n_{f,max}$. In a typical case, the number of frames in between two mainframe positions $x_{mfr,i}$ and $x_{mfr,i+1}$ is given by

$$n_{sections,i} = \left\lfloor \frac{x_{mfr,i+1} - x_{mfr,i}}{\Delta x_{ref}} \right\rfloor + 1. \tag{1}$$

The position of the regular frames is then given by

$$x_{f,ij} = x_{mfr,i} + \frac{x_{mfr,i+1} - x_{mfr,i}}{n_{sections,i}} \cdot j, \qquad \{j \epsilon \mathbb{N}_0 : 0 \leq j < n_{sections,i}\}. \tag{2}$$

This results in equal distances between frames within each segment, which are at most as wide as the reference spacing. The result of the distribution algorithm for an example configuration is shown in Fig. 4. Note that the mainframe position for front spar of the vertical tail plane has been merged with the rear pressure bulkhead due to the tolerance criterion.

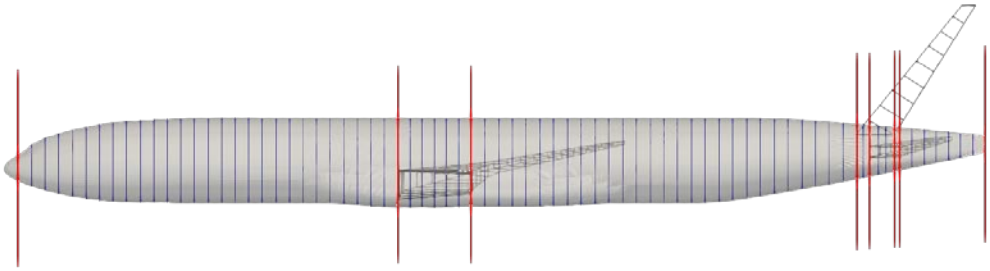

Figure 4: Mainframe positions and resulting frame distribution

The maximum number of frames $n_{f,max}$, acts as a safeguard to prevent the creation of designs with an unreasonably high number of frames, which could cause very high program runtimes. If the number of frames computed using equations 1 and 2 exceeds $n_{f,max}$, the frame spacing Δx_{ref} will be adjusted.

Furthermore, separate structural element uIDs can be given for mainframes and regular frames.

Instead of using linear spacing, a nonlinear approach can be employed. It assumes for the space between two stringers to be proportionate to the cross-sectional area $A_{xsec}(x)$ at the fuselage position. This yields a distribution function

$$f(x) = \frac{A_{xsec}(x)}{\int_{x_0}^{x_n} A_{xsec}(x)dx}. \tag{3}$$

The cumulative distribution functions for $f(x)$ then provides the scaled x-coordinate

$$x = F(x') = x_0 + (x_n - x_0) \int_{x_0}^{t} f(t)dt\,, \qquad \{t \epsilon \mathbb{R} | x_0 \leq x' \leq x_n\}\,, \tag{4}$$

which then needs to be solved for the reference frame positions. The new reference positions serve as input for the above interpolation algorithm to compute a distribution on the scaled coordinate, which is fed back into the cumulative distribution function to yield the final positions.

The distribution can be modified by applying an exponent to the areas. For instance, assuming an approximately circular cross-section, the scaling would be proportionate to the circumference, and

thus the stringer spacing, for an exponent of 0.5. This potentially yields a superior structure, since the deviation of aspect ratios between different buckling fields is reduced.

3.3 Stringer Placement

As outlined in chapter 2, stringer placement may become a very complex task, depending on the boundary conditions. In the augmentation module, various cases are supported. For all of them, some basic knowledge on the fuselage shape should be available.

- Centroid position over fuselage length $\mathbf{p}_{\mathbf{cent}}(x)$
- Cross-sectional area over fuselage length $A_{xsec}(x)$
- Minima and maxima in z-direction $z_{min}(x)$, $z_{max}(x)$

Furthermore, the number of stringers and the uIDs of the left and right structural element are given in the augmentation tree.
In the following, two cases shall be considered in more detail, the simple linear interpolation case and the stepwise constant case, where all stinger planes are extruded along the global x-axis.

Linear interpolation case
If the application of a linear stringer interpolation scheme is allowed, the placement of stringers may be kept relatively simple. Given a strictly convex fuselage surface, a simple stringer distribution can be computed based on the given number of stringers:

$$\varphi_i = \frac{i}{n_{stringers}} * 2\pi, \qquad \{i \epsilon \mathbb{N}_0 | 0 \leq i < n_{stringers}\}. \tag{5}$$

This distribution is applied at the nose and tail points of the fuselage, yielding the plane distribution shown in Fig. 5.

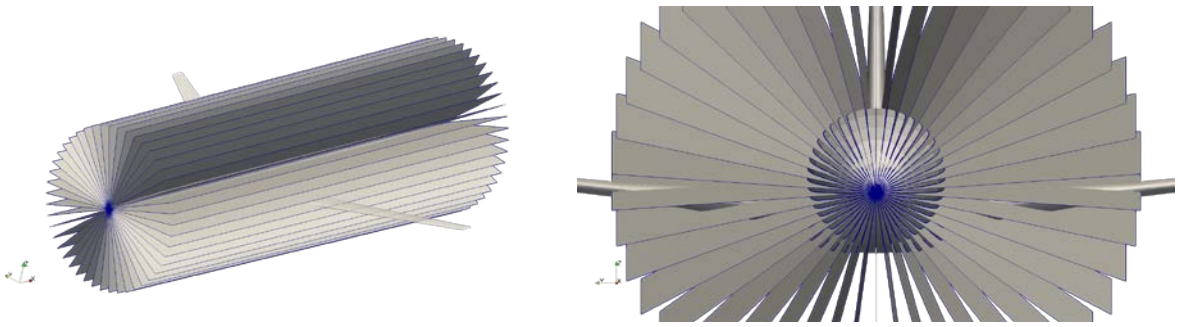

Figure 5: Stringer planes for linear interpolation scheme

Intersection of the stringer planes and the fuselage surface yields the stringer extrusion paths shown in Fig. 6.

Figure 6: Stringer extrusion paths for linear interpolation scheme

Adapting for non-point shaped fuselage ends
While the above approach will function for many designs, the assumption of a strictly convex fuselage surface is false for almost all of them. Thus, the above method must be expanded to cover several exceptions.
In the case of a strictly convex fuselage shape, the cross-sections at the beginning and the end of the fuselage must be points, which is not always the case for real designs. Therefore, reference points

may have to be computed based on the evaluation cross-section in order to avoid stringer agglomerations. This can be accomplished by analyzing the cross section in question.

The basic approach is to approximate the cross section shape with an ellipse, whose main axes are parallel to the global y- and z-axes. The lengths of the major and minor axes are computed by subtracting the maximum and minimum coordinate values in y- and z-direction at the x-position in question. Using these values, a linear eccentricity can be computed. Depending on the orientation of the ellipse, the vector origins will be spaced out either along the y- or z-axis depending on the angle to at most the linear eccentricity parameter using the equations below:

$$p'_y = p_y - c \cdot \sin\varphi \text{ if } \Delta y > \Delta z,$$
$$p'_z = p_z + c \cdot \cos\varphi \text{ otherwise.} \quad (6)$$

The result of this adaptation scheme for a line-shaped fuselage end is shown in Fig. 7.

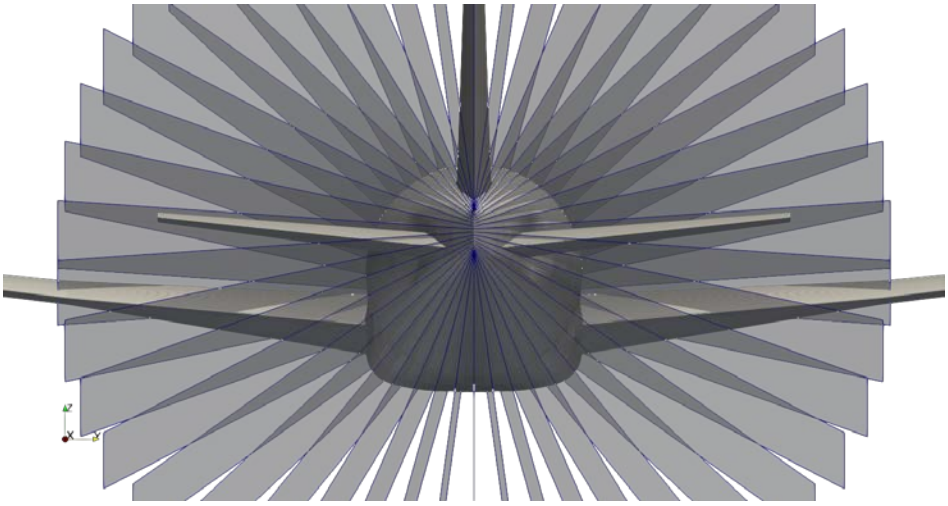

Figure 7: Adapted source points fuselage tail ending in a line

Improving the stringer distribution by adding evaluation points

For fuselage shapes with large concave sections, it is possible for the connecting line of the evaluation points to intersect the fuselage surface. Assuming the fuselage to be symmetric, the intersection points can easily be determined by comparing the connecting line to the profile extrema in z-direction. In case of intersection, further evaluation points must be added along the length of the fuselage. As before, the evaluation point is placed at the centroid of the respective cross section.

While the intersections between the surface and the connecting line rarely occur in practice, the algorithm for introducing additional evaluation points is still very useful in a different respect, namely the treatment of cylindrical sections. As can be seen in Fig. 6, the stringer extrusion paths are not parallel to the fuselage length axis within the cylindrical section, if just one definition segment is used. This is not typical for real-life aircraft designs, which are also subject to manufacturing constraints.

By adding evaluation points at the beginning and the end of each cylindrical section as shown in Fig. 8, this issue can be overcome. The respective positions along the x-axis can be approximated by analyzing the area function. A cylindrical section is found everywhere, where the function stays within a given tolerance over a given length.

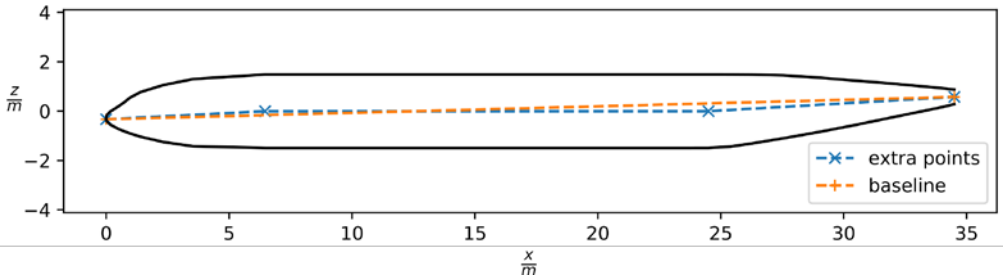

Figure 8: Insertion of additional points at the cylindrical section

Piecewise constant interpolation case

As mentioned briefly in the previous chapter, the linear interpolation scheme may not be applicable under certain circumstances, where the stringer definition planes are required to be orthogonal to the y-z-plane. In this case, it is necessary to enforce the consistency between sections, i.e. that stringer planes meet on the fuselage surface at the transition location.

The simplest solution to this problem is to determine a reference cross-section position at which the angle distribution is given. For a classical fuselage, the position, where the cross-sectional area $A_{xsec}(x)$ reaches its maximum is a good choice, because it usually lies within the cylindrical section. Based on the angles, the intersection points on the fuselage surface can be computed. The extruded planes must meet at these points for the extruded planes to be consistent.

Now, two evaluation points are picked, up to where the planes shall be extruded. In our simple example, these are the centroids at the nose and the tail. Since these points might have a different position in the y-z-plane than the centroid at the reference position, the angles must be adjusted to meet the pre-calculated intersection points on the reference cross-section using the arctan2 function

$$\varphi_{ij} = -\arctan2\left(\Delta p_{y,ij}, \Delta p_{z,ij}\right), \tag{7}$$

where $\Delta p_{d,ij} = p_{ref,d,i} - p_{ev,d,ij}$ with the coordinate direction d, the circumferential index i and the evaluation position index j. Note, that the order of the coordinate directions in the arctan2-function is reversed with respect to convention. In conjunction with the negative sign, this will place the origin of the circumferential coordinate at the top of the fuselage. An exemplary set of definition planes is shown in Fig. 9.

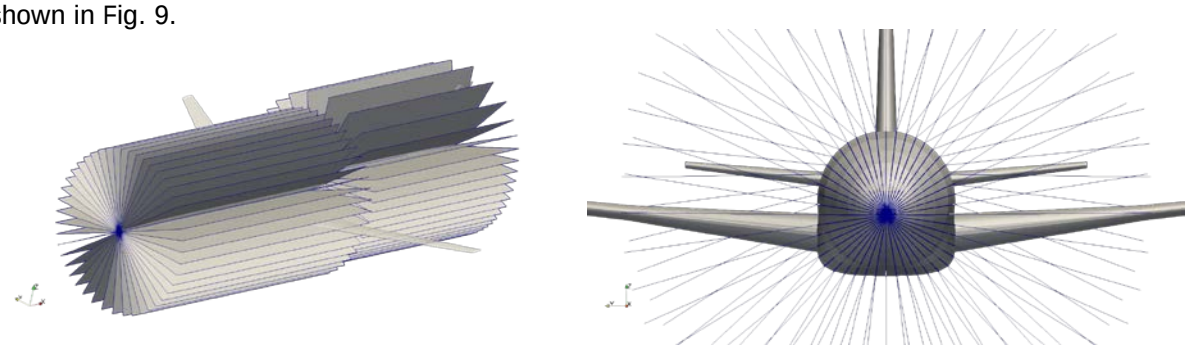

Figure 9: Stringer planes for piecewise constant interpolation scheme

The resulting planes can be seen in Fig. 10. Unlike the linear approach, the piecewise continuous interpolation yields a constant stringer distribution across the entire cylindrical section without the addition of evaluation points.

Figure 10: Stringer extrusion paths for piecewise constant interpolation scheme

3.4 Bulkhead Placement

As outlined above, the bulkhead positions are an important input for the frame distribution. An arbitrary number of bulkheads can be set. They will be spaced evenly in the range given by a start position and end position parameter along the x-direction. They will be assigned the sheet element specified in the sheetUID node.

3.5 Floor Placement

The simplified structural description provides the possibility, to define an arbitrary number of floors in a comparatively concise manner. In this respect, its capabilities reach beyond those of the original CPACS definitions, where floor structures are quite rigidly separated into cargo and passenger floors.

The associated branch allows for the specification of an arbitrary number of floor elements, each containing a mandatory branch for cross beam definitions and optional branches for long beams and struts.
The cross beam definition only requires two parameters, namely a z-position and a structural element uID. The former places the floor in the fuselage, while the latter sets the shape of the cross section. Optionally, two x-positions can be given as well, specifying the elongation across the fuselage length. If no positions are given, the floor will be bracketed by the two outermost bulkheads.
The long beam definition also requires two parameters. Once more, a structural element uID is required for describing the cross section. Furthermore, the number of beams can be specified. The beams will be distributed equally across the longest crossbeam. It is assumed that the longitudinal beam runs across the full length of a given floor, i.e. from the first to the last crossbeam.
Similarly, the cross beam struts are defined across the whole floor length. It is possible to define an arbitrary number of strut positions, each of which is made up of a global y-variable on the cross beam and a global orientation angle. Furthermore, a symmetry parameter can be set to mirror the struts at the xz-plane.

3.6 Skin Segment Distribution

So far only, very basic rules for the instantiation of skin segments have been implemented. For the augmentation, the number of segments along the length and the circumference can be given. The augmentation tool will automatically create the given number of segments, while trying to make them approximately equal in size. In addition, the Boolean parameter startAtZeroDegrees is provided. If true, a skin segment boundary will be placed at the top of the fuselage. Otherwise, the top segment will be centered at the 0° position. Since the placement of the skin segments is highly dependent on the stringer and frame distributions; the specifications can only approximately be fulfilled in many cases.

3.7 Dynamic Aircraft Model Points

Several of the model generators mentioned in section 1 use the dynamic aircraft model (DAM) points from CPACS as load introduction points. During the augmentation, a set of DAM points can be generated along the fuselage. It is only required to give the number of points. They will be spaced evenly across the length of the fuselage. For demonstration purposes, a set of dummy loads is also added.

4 APPLICATIONS

As an application case, the augmentation module will be used to instantiate a structural design on two given geometries. In addition, a GFEM model of the fuselage is created and analyzed using the methods described by Walther et al. [4].

4.1 AGILE DC-2 Reference Configuration

The first example case is the Agile DC-2, which is a reference design considered in the AGILE project [11]. The DC-2 is a conventional aircraft configuration with a long cylindrical section capped by a cockpit and tail area.
The model is augmented with a frame spacing of 0.5m and 61 stringers using the linear frame spacing and piecewise constant stringer interpolation design rules. Therefore, the resulting stringer and frame distributions are analogous to Figs. 4 and 10. The skin is divided into 6 lengthwise and 4 circumferential segments. Furthermore, two bulkheads are created, as well as a passenger and a cargo floor, where only the former has longitudinal beams. A view of the interior is given in Fig. 11. A dummy load distribution is applied at 42 DAM points.

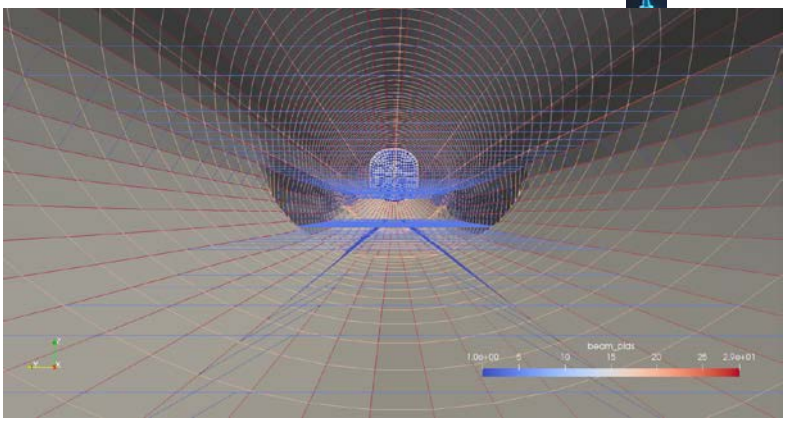

Figure 11: Interior of GFEM fuselage model of Agile DC-2 based on augmentation

Using the Conspyre package [4], the model is written to Nastran format [14], where the stress distribution and displacements shown in Fig. 12 are computed in a linear static analysis.

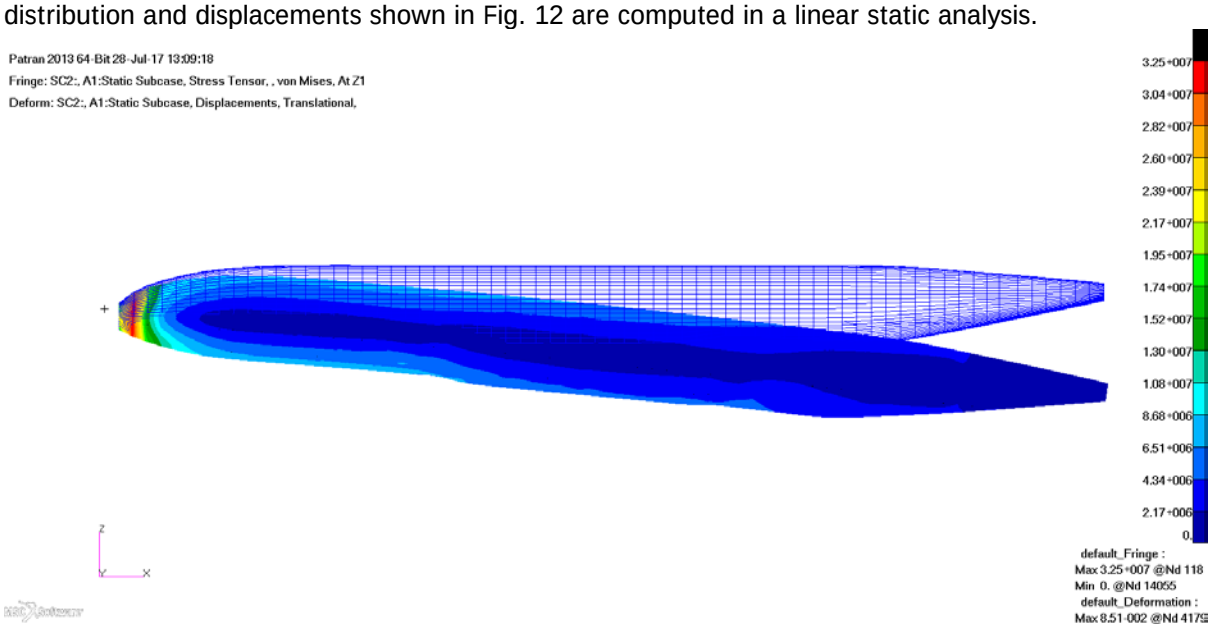

Figure 12: Nastran solution of Agile DC-2 fuselage model for dummy loads

4.2 X-31

As a second example case, the X-31 fighter design is picked. Featuring a nonconvex fuselage surface and non-elliptic cross-sectional profiles, it is a more challenging example case, compared to the Agile DC-2. In order to achieve a somewhat realistic design, the augmentation parameters have been adjusted slightly. A total of five bulkheads have been defined to model the mount points for wings and engine. Furthermore, only one short floor segment is modeled between the second and third frame to emulate a cockpit area.
The resulting model is shown in Fig. 13. The design rules for frames and stringers remain the same as in the previous example. It can be seen, that the rules still yield a good structural layout for the given geometry.

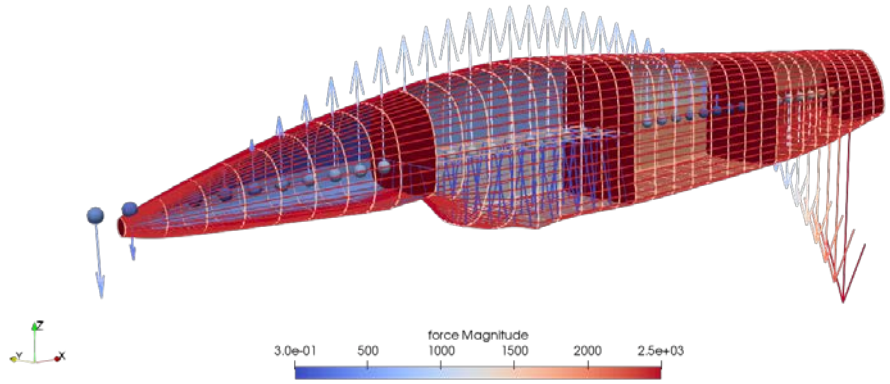

Figure 13: GFEM fuselage model of X-31 based on augmentation

As in the previous case, a linear static analysis of the model is performed using Nastran. The results are shown in Fig. 14.

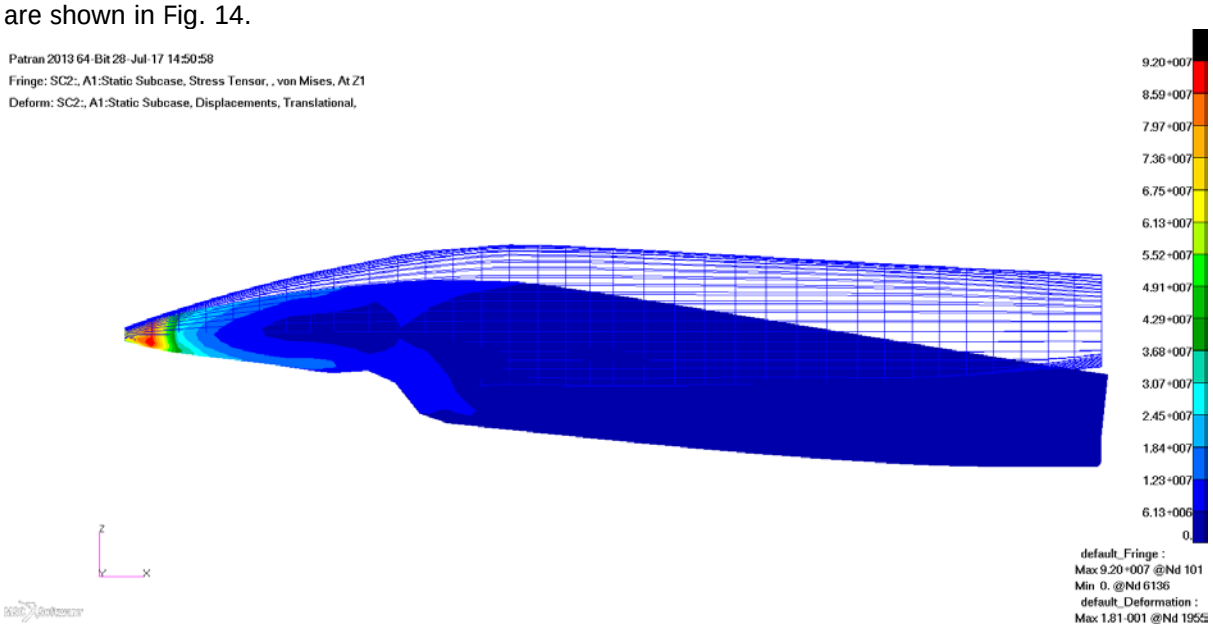

Figure 14: Nastran solution of X-31 fuselage model for dummy loads

The analysis succeeds in both cases. It is shown, that the augmentation module is sufficiently flexible to handle not only classical tube-shaped fuselages, but also more disruptive designs.

5 CONCLUSION AND OUTLOOK

This paper introduces a knowledge-based structural design augmentation module for CPACS, which takes very basic parameters and a set of design rules as an input to create a detailed structural layout in CPACS. It has been shown that the chosen rules and parameters not only cover common transport aircraft designs, but also translate nicely to fighter designs. Furthermore, it could be demonstrated, that the CPACS datasets from the augmentation provide valid input for a GFEM model generator.
However, work on the augmentation module is ongoing, as several key features remain to be implemented. Most obviously, augmentation of wing structures is yet to be implemented. Furthermore, implementing more detailed knowledge to connect the individual structural parts is necessary. For instance, the wing box shape might be considered for the floor placement, which could in turn influence the stringer distribution. Another interesting aspect might be to vary the running length of stringers along the fuselage, i.e. letting stringers run out as the cross section decreases. This is commonly done in real-life aircraft designs and prevents small elements in the GFEM model. Additionally, the implementation of load introduction regions such as the center fuselage area and the tail plane attachment area must be considered.

Finally, the augmentation module must be integrated into a larger aircraft design process, where more information from other disciplines, such as proper loads, will be available for the analysis which will improve the results. On the other hand, the parameters provided in the simplified structural description can also serve as control variables for an optimization.

ACKNOWLEDGEMENTS

The authors would like to thank Dieter Kohlgrüber and Julian Scherer from the DLR Institute of Structures and Design for their input and the fruitful discussions.

REFERENCES

1. B. Nagel, D. Böhnke, V. Gollnick, P. Schmollgruber, A. Rizzi, G. La Rocca, J. J. Alonso; 2012; „Communication in Aircraft Design: Can We Establish a Common Language?"; 28th Congress of the International Council of the Aeronautical Sciences (ICAS); Brisbane, Australia, 23-28 September
2. CPACS Homepage; 2017; http://www.cpacs.de (retrieved 27-07-2017)
3. J. Scherer, D. Kohlgrüber; 2016; "Fuselage structures within the CPACS data format"; Aircraft Engineering and Aerospace Technology; **88**(2); pp. 294–302
4. J.-N. Walther, M. Petsch, D. Kohlgrüber; 2017; "Modeling of CPACS-based fuselage structures using Python"; Aircraft Engineering and Aerospace Technology; DOI: 10.1108/AEAT-01-2017-0028.R1
5. J. Scherer, D. Kohlgrüber, F. Dorbath, M. Sorour; 2013; "Finite element based tool chain for sizing of transport aircraft in the preliminary aircraft design phase"; 62. DLRK; Stuttgart, Germany
6. T. Führer, C. Willberg, S. Freund, F. Heinecke; 2014; "Parametric Model Generation and Automated Sizing Process for the Analysis of Aircraft Structures using the Design Environment DELiS"; 4th EASN Workshop on Flight Physics & Aircraft Design; Aachen, Germany; 27-29 October
7. F. Dorbath; 2014; "A Flexible Wing Modeling and Physical Mass Estimation System for Early Aircraft Design Stages"; Ph.D. Thesis; Technical University Hamburg-Harburg; Hamburg, Germany
8. L. Travaglini; 2016; "PyPAD: A Framework for Multidisciplinary Aircraft Design"; Ph.D. Thesis; Polytecnico Milano; Milan, Italy
9. R. Maierl, Ö. Petersson, F. Daoud; 2013; "Automated creation of aeroelastic optimization models from a parameterized geometry"; International Forum on Aeroelasticity and Structural Dynamics IFASD; Bristol, UK; 24-26 June; pp. 1417-1431
10. S. Görtz, C. Ilic, M. Abu-Zurayk, R. Liepelt, J. Jepsen, T. Führer, R. Becker, J. Scherer, T. Kier, M. Siggel; 2016; „Collaborative multi-level MDO process development and application to long-range transport aircraft", 30th Congress of the International Council of the Aeronautical Sciences, Daejeon, Korea; 25-30 September
11. P. D. Ciampa, B. Nagel; 2017; "The AGILE Paradigm: the next generation of collaborative MDO", 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference; Denver, Colorado; 5-9 June
12. AGILE Project Homepage; 2017; http://www.agile-project.eu (retrieved 27-07-2017)
13. M. C. Y. Niu; 1988; *Airframe Structural Design*; Conmilit Press Ltd.; Hong Kong
14. MSC Software Corporation; 2014; *MSC Nastran 2013.1.1 Quick Reference Guide*