This electronic thesis or dissertation has been downloaded from Explore Bristol Research, http://research-information.bristol.ac.uk

Author:
Kucera, Jakub

Title:
Automating “design for manufacture” of aerospace composite components

General rights
Access to the thesis is subject to the Creative Commons Attribution - NonCommercial-No Derivatives 4.0 International Public License. A copy of this may be found at https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode This license sets out your rights and the restrictions that apply to your access to the thesis so it is important you read this before proceeding.

Take down policy
Some pages of this thesis may have been removed for copyright restrictions prior to having it been deposited in Explore Bristol Research. However, if you have discovered material within the thesis that you consider to be unlawful e.g. breaches of copyright (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please contact collections-metadata@bristol.ac.uk and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline nature of the complaint

Your claim will be investigated and, where appropriate, the item in question will be removed from public view as soon as possible.

Automating “design for manufacture” of aerospace composite components

Jakub Kučera

14/11/2021

Academic supervisors: James Kratz, Thomas Rendall

Industrial supervisor: Amit Visrolia

Word count: 51117

A dissertation submitted to the University of Bristol in accordance with the requirements for award of the degree of Engineering Doctorate (EngD) in composites manufacture in the Faculty of Engineering

Author’s declaration:

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: Jakub Kucera DATE: 14/11/2021

Abstract:

Design of composite components is inherently multi-disciplinary problem; the current linear design method is not very well suited for holistic design optimization and frequent iteration of design. A novel methodology for comprehensive design of a composite component is suggested in this thesis. The methodology utilizes Python scripting to create an interconnected system of simulations (SySi), including CAD, basic aerodynamic analysis, meshing, structural FE, kinematic braiding simulation, and resin infusion simulation. An SQL database is used to manage and trace iteration data. The models are self-generated with given iterated variables, to allow for a multi-simulation optimisation and fast component rework.

A demonstrator part was developed to demonstrate the suggested methodology. A small UAV braided spar was selected. It is sufficiently simple for the demonstrator development, yet it retains some of the complexities of typical composite component.

All the modules were successfully automated, providing variety of standalone benefits. A Latin hypercube sampling method proved highly successful in initial sampling of the design space. It also supports troubleshooting, as it generates all allowed combinations of parameters. It was also used as an input for surrogate models. Non-surrogate optimisations proved too computationally expensive. With surrogate model this problem was alleviated to some extent. Surrogate model was successfully created with minimal errors present in the validation. However, significant number of datapoints was required to create high quality surrogate. The subsequent optimisation was not particularly useful for the demonstrator part, as the optimised values could be obtained from plotting the sampling data.

The alternative, highly automated design process was successfully demonstrated. Some of the envisioned benefits are clear from the demonstrator, while others require further study.

The demonstrator system with its scripts is available on https://github.com/Ellutze/sysi .

Abbreviations

API	application programming interface
CAD	Computer aided design
CLA	classical laminate analysis
COTS	commercial of the shelf software
DFM	design for manufacturing
FE	finite element
FEA	finite element analysis
IMD	initial mandrel distance
MS	mandrel speed
LHS	Latin hypercube sampling
LLT	lift line theory
OSS	open-source software
QFD	quality function diagram
RTM	resin transfer moulding
SQL	structured query language
SySi	system of simulations, referring to the design methodology described in the thesis
UAV	unmanned aerial vehicle
$\boldsymbol{V_f}$	volume fraction
VLM	vortex lattice method

Any variables used in the referenced scripts are defined in the annotations.

Acknowledgments

I would like to thank Dr. James Kratz for his supervision throughout the majority of my EngD, and mainly through its final stages. I would also like to thank my industrial supervisor Dr. Amit Visrolia and my secondary academic supervisor Dr. Thomas Rendall for the additional guidance.

I would also like to thank my initial supervisors Prof. Kevin Potter and Miroslav Stojkovic for the initial guidance.

I would like to thank researchers at the Purdue CMSC for facilitating the visit to their centre for collaboration, namely Dr. Michael Bogdanor, Dr. Rebecca Cutting, and Prof. Byron Pipes.

I would also like to thank the IDC management team for their support, which allows us EngDs to spend more time on technical problems instead of bureaucracy.

Lastly, I would like to thank the engineers from the unnamed company that supported the industrial feasibility review presented in chapter 5. For the sake of confidentiality this shall be a nameless acknowledgment.

Contents

List of figures:

List of tables:

1. Introduction

1.1 Background and motivation

The design of engineering components is constantly improving, usually due to newly available technologies. Through iterative change, engineers have come a long way from a pencil drawn components to highly optimised simulations. However, the process of design seems to remain fixed in many ways.

The process is sequential. Product specification goes to a design engineer who produces a basic geometry and suggests materials. The design is then passed to the structural analysis department where the performance of the part is analysed, often using finite element (FE) models. Once the structural integrity has been verified, the part is sent to the manufacturing department.

At each stage of the design process, there is a chance for the part to be sent back one step for rework or adjustments. This is working very well for low-volume, high-value parts that are made in the aerospace industry. Using this approach, responsibilities are clearly distributed. Everybody in the design process knows what their task is, and it is quite easy to evaluate individual work. For metallic parts that require minimal optimisation, and use traditional material removal manufacturing processes, this might even be adequate going forwards. However, problems arise if the sequential design approach is applied to complex composite structures that are made using layer by layer additive manufacturing methods.

The first complication that arises with composite structures is the material layup or fibre orientation, which significantly influences all three groups: design, analysis, and manufacturing. Designers would like the layup and thickness to conform to the assembly requirements and shape constraints. Analysts will likely want the fibres positioned along the main loading direction. Manufacturing engineers would like a layup which allows for a robust process with minimal material alterations (e.g. darting, shearing, etc...). Although the constraints discussed above may be a large oversimplification and will depend on the manufacturing process, issues like this pose a couple of questions: Who has the final say? What is the most important aspect of design?

The second complication of the sequential workflow in the design of composite structures is the inherent imbalance. Designers come first and lock certain parameters in place. These could be radii, joggles, thickness, mounting points, etc... Structural engineers then lock the layup as it defines the loading paths by specifying another set of parameters. Materials and Processes team then finish the design by fixing the parameters that are only relevant for them. A problem with the sequential design approach is that the radii in certain situations might be more important for manufacturing than for design, but the decision in current design process usually lies with the design department. The manufacturing team might have to request changes for the radii to be suitable. Radius is just an example and various other issues related the process imbalance may manifest depending on the part in question.

The third problem with the sequential approach for composites is often quoted in the "design for manufacturing" literature, which report that about 75% of product costs are decided at the conceptual design stage(1–6). While the numbers might differ, the main idea stands: decisions at the conceptual stage can affect the component to such a degree, that any further optimisations might seem futile. This is because deciding between several manufacturing options and many different configurations can be overwhelming. Some of these aspects of design are therefore fixed early. For instance, the manufacturing process might be decided before the structural implication of it are fully explored, or before the shape is fully defined. The incompatibility of the shape with the manufacturing process can become very costly.

Beyond the sequential aspect, another drawback of the current design methodology is its separation. All three listed departments will generate their own models with their own objectives. Numerical simulation is where a significant amount of progress has been achieved in recent years. The capability to reliably predict structural properties or manufacturing are increasing due to a large amount of software development, ever-growing computational capacities, and expanding userbase. However, the capabilities of these software only rarely reach beyond one engineering department.

The first downside of the separated design approach is that each department will choose the ideal software for their purpose. This very often leads to incompatibilities or loss of details when passing models between departments. It also makes any rework quite difficult. Although parametric models are becoming common, it is still very likely that changes in one department will require additional work in another to update the model.

Many "design for manufacturing" (DFM) strategies are trying to improve the balance between performance and manufacturing costs. However, the sequential and modular aspects of the design process are barriers to implementing new DFM strategies. The issues described above are inherent to the design process and therefore can be mitigated but not removed.

1.2 Aim and Objectives

The aim of this EngD is to improve the design process for aerospace composite parts, with the use of scripting and good data management.

The core objectives are:

1. Establish a robust scriptable data exchange between simulation software and a database
2. Automate simulation software involved in the design process
3. Connect the automated software and database using scripting
4. Iterate through the design of an example part
5. Evaluate optimisation options for such a system of simulations
6. Evaluate the suitability of envisioned design process for industrial application

1.4 Novelty

The main novelty comes from the design process proposed. Instead of a standard process where optimisation of design happens at final stages, this process promotes design for optimisation, where the multi-simulation optimisation itself can become integral part of the initial design. The design is also highly data driven because of its close tie with the database. This provides a full traceability of design, as all inputs and outputs are recorded automatically, and robust naming system is implemented. All the models require full automation, allowing for recreation of models based on pre-defined variables. This is a step forward from parametric models as it allows for greater variety of shape parameters. Unlike some of the software integration provided by software companies, the novel approach allows for heterogenous software requirement. Each software can be from a different vendor or open source, as long as it is scriptable.

The solution outlined in this thesis utilises several modern tools, amending the design process accordingly. The tools in question are numeric simulations, scripting automation, automated data collection and complex optimisations.

The proposed design process relies heavily on automatic generation of models and requires some scripting knowledge of the engineers. The comparison between the standard and envisioned processes is shown in Figure 1.1.

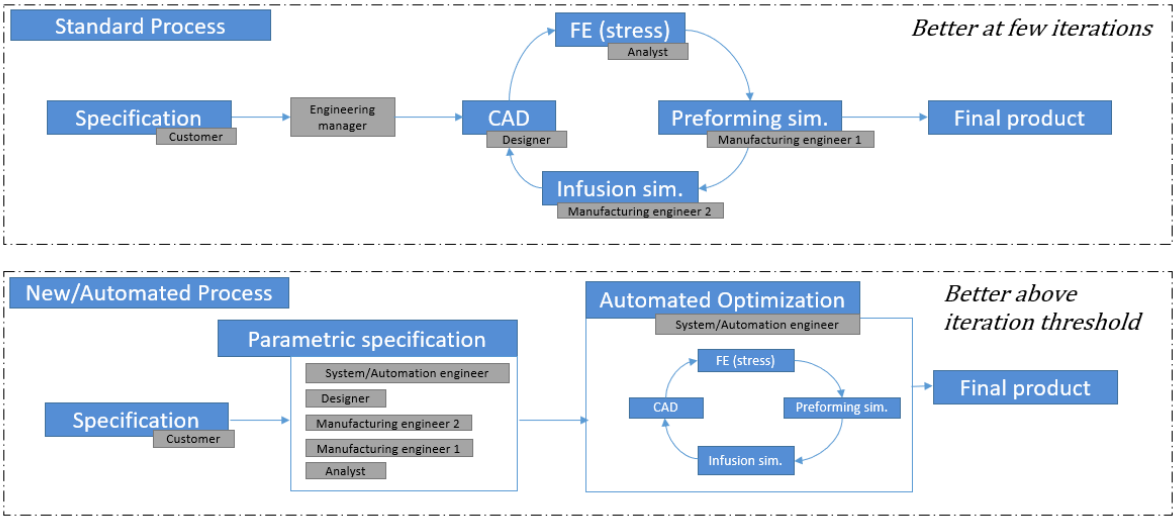


Figure 1.1 – the news design process compared to the standard design process

The new process is based on creating a set of connected and automatically generated models. The main difference is that for the new process, the first iteration of the design will take longer to develop. This is because parametric automated models are required, and the interaction between models must be considered. However, for all subsequent iterations of the design, the engineering time required should be negligible. This is clearly more beneficial for parts that require a higher degree of optimisation and thus more iterations. The layout of the new process solves the first two issues of the conventional process: the question of priority and the imbalance problem. The priorities are defined by the specification and directly translate to objective function. Therefore, all variables and their effects on results are evaluated with respect to the overall customer requirements.

The novel process also allows for seamless extensions, such as costing of the part or material selection modules. These can be integrated into the optimisation loop, and their weight can be easily reflected in the fitness functions used. For high value component lines, it might be worth extending the system of simulation to model all aspects of the component lifecycle.

1.2.1 Who should be developing the envisioned solution?

On the first glance, this appears to be a programmer’s job, not an engineer’s job. However, due to the product structure of software companies this is difficult. Each of the software companies specialize in one piece of software. If they provide other software from the design chain, it is likely that it will be outcompeted in terms of performance by more specialized software. Every company’s interest is to make the software work together, their software that is. There is little monetary motivation for each software company to work towards integration with another company’s software. In practice, in industry each department is going to use the software that best suits their needs. This will result in multiple software providers supporting different aspects of design. In this environment, it is difficult to provide complete integrated solutions, even while such attempts are more and more common.

Secondary issue is that the software companies do not always appreciate all the complexities that come from dealing with specialized materials, such as composites. An engineer is better positioned to envision how the flow of data works and what connections need to be established. This is because of higher understanding of the purpose of individual software solution, with regards to the specific design process employed at the company, for their specific line of products. Also, each novel product might have bespoke requirements that the software companies cannot predict, requiring the engineers to develop their own modifications, typically with scripts.

In effect, it is not cost effective for the software companies to provide the solution outlined above. The solution is to some extend bespoke for each potential customer, in terms of requirements, software used, products developed etc…

1.6 Thesis Structure

Following this chapter is the literature review, which pragmatically assesses the key topics needed to inform the developments required to deliver the envisioned solution. Demonstration of the solution has been developed with an example part, this is described in Chapter 3. The core demonstrator includes automated shape generation, braiding simulation, and structural analysis. These are linked and managed using python. The data generated by the system is stored in SQL database. While Chapter 3 focuses on how the simulations were developed and automated, Chapter 4 outlines the iteration through the demonstrator system. This addresses how the iterations are specified, initiated, managed, and describes various optimisation methods trialled. Chapter 5 outlines additional developments added to the core demonstrator. These demonstrate the modularity of the system and present the procedure for extending the system. The chapter also includes an assessment of the potential to introduce this system to the industry.

A diagram of the objectives and their location in the thesis chapters is shown as Figure 1.2.

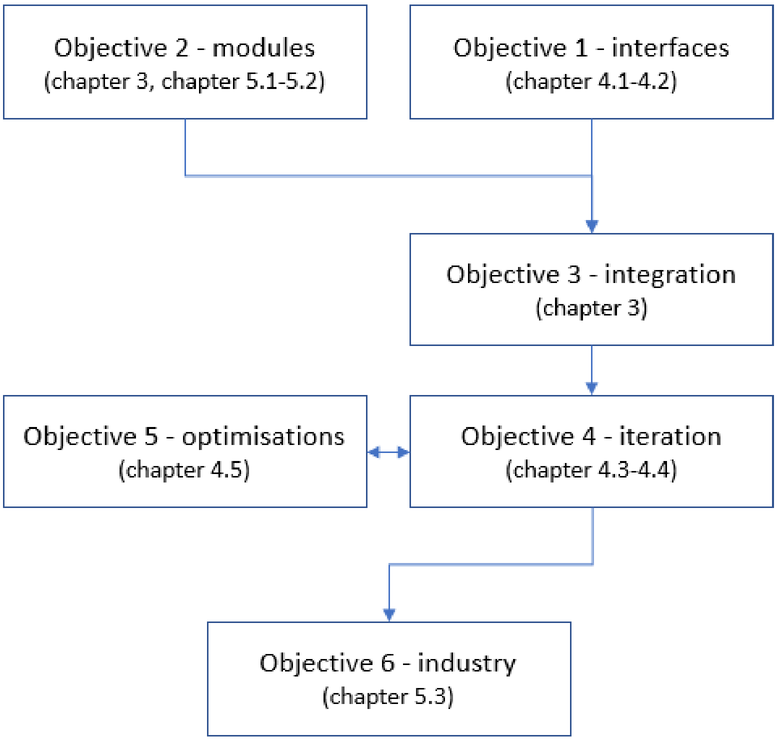


Figure 1.2 – objectives and chapters

References

1. Ullman DG. The Mechanical Design Process. 4th ed. 2003.
2. Saravi ME, Newnes LB, Mileham AR, Goh YM. Estimating cost at the conceptual design stage to optimize design in terms of performance and cost.
3. Siedlak DJL, Schlais PR, Pinon OJ, Mavris DN. Supporting affordability-based design decisions in presence of demand variablility. In: ASME 2015 International Manufacturing Science and Engineering Conference.
4. Karbharil VM, Burns JS, Wilkins DJ. Total quality design (TQD) approach for composites. In: International SAMPE Technical Conference. 1991. p. 1097–111.
5. Newnes LB, Mileham AR, Cheung WM, Marsh R, Lanham JD, Saravi ME, et al. Predicting the whole-life cost of a product at conceptual design stage. J Eng Des. 2008;19(2):99–112.
6. Roy R, Kelvesjo S, Forsberg S, Rush C. Quantitative and Qualitative Cost Estimating for Engineering Design. [cited 2018 Jan 3]; Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.3282&rep=rep1&type=pdf
7. Kucera J. Sysi [Internet]. 2020. Available from: https://github.com/Ellutze/sysi
8. Moza S. LHS-MDU [Internet]. GitHub repository; Available from: https://github.com/sahilm89/lhsmdu
9. Deutsch JL, Deutsch C V. Latin hypercube sampling with multidimensional uniformity. J Stat Plan Inference [Internet]. 2012 Mar 1 [cited 2019 Nov 19];142(3):763–72. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0378375811003776?via%3Dihub
10. Bouhel MA, Bartoli N, Otsmane A, Morlier J. Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction. In: Open Archive Toulouse Archive Ouverte [Internet]. 2017. Available from: https://hal.archives-ouvertes.fr/hal-01598259/document
11. Awad M, Khanna R. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers. Apress open. 2015.

2. Literature Review

Several different topics were deemed relevant to this thesis. Some of these are relevant for specific modules of the “system of simulations” (SySi), some are focusing on methods similar to the holistic design and optimisation proposed here. Several niche sections are quite concise and exist for later reference in the appropriate chapters. To avoid excessive scope creep of this review, mainly the findings that were later used in development of SySi are mentioned here.

2.1 Aerospace needs and outlook

The push for digitization is clear in Aerospace industry for quite a while. The ATI reports from 2016-2018 put digital twin and other similar concepts high up on the priority list, only superseded in visibility by the push for highly integrated structures (1–3). System of simulations does not address all the issues presented, but it is a process which goes well with the broad digitization outlook.

At the start of this project an initial literature review was executed to find the gaps in design and simulation of composite components. The main finding was that composites specific processes and methods are being researched, they have clear path to being solved and simulated. What we do not have is a framework where all of the findings will lead to the optimal easily adjustable design.

The fact that large companies such as Dassault, Siemens, or ESI are trying to offer more complete platforms is also sign that there is a requirement in the industry for projects like this. These are further discussed in the main body of the literature review below.

2.2 Design software automation and scripting

Both commercial and open-source CAD software were reviewed for use in SySi. Ideally, open-source software would be used for easier shareability and access. Several open-source CAD software were tried. The most promising was FreeCAD, as it has a python interface, and a small community of people developing modules. However, similarly to every other open-source CAD tried, it lacks robustness. The functions output frequent errors that are difficult to predict, which adds a layer of difficulty to any automation. Therefore, commercial software was deemed more suitable for initial implementation.

Out of the commercial software 3 different providers were briefly considered: Autodesk software (Fusion, AutoCAD, Inventor), CATIA and SolidWorks by Dassault Systemes, and NX by Siemens.

No evidence of people successfully automating Autodesk software was found, although the software website suggests the automation should be possible. Small community of CATIA users who use VBA to automate some tasks exists. The automation in CATIA is usually done using the macro functions and VBA, but some people have used python(4). The automation of SolidWorks follows the same methodology and is supported with similar mechanisms to CATIA. However, it appears that the number of engineers automating SolidWorks is very small. Similarly to CATIA, python API exists but it is more difficult to use (5). NX is rarely automated, but it can be done through various high level programming languages and Microsoft Visual Studio. (6)

The table 2.1 summarises the most notable CAD software. This is by far not an exhaustive list, but other software solutions were deemed clearly unsuitable for the envisioned work.

Table 2.1 – summary table of most notable CAD software

	Required shape generation	Used for engineering	Scripting	Costs	Robustness	Provider
CATIA	yes	Yes (high-end)	Yes (VBA, python API)	High. No free student licences.	High	Dassault Systemes
NX	yes	Yes (high-end)	C++, Visual Basic, Java..	High, student licence available free.	High	Siemens
SolidWorks	yes	yes	Less common (VBA, C, python)	High. No free student licences.	High	Dassault Systemes
Fusion	yes	yes	Uncommon	Low, student licence available free.	N/A[1]	Autodesk
Inventor	Yes	yes	Uncommon	Medium, student licence available free.	Medium	Autodesk
AutoCAD	Yes	Architecture	Uncommon	Medium, student licence available free.	N/A	Autodesk
FreeCAD	Difficult	Rarely	Yes (python)	Open source	Low-medium	Community Developed
PythonCAD (Linux)	Difficult	Rarely	Yes (python)	Open source	N/A	Community Developed
Blender	No	No	Yes (python)	Open source	N/A	Community Developed

---

[1] N/A at robustness means author did not have a chance to personally test and does not have a reliable reference.

2.2.1 CATIA automation

CATIA automation is not as common as one would expect considering its extended scripting capability. However, there are few research areas where it is sometimes used. One of these is aircraft configuration. Yang et.al. focused on assembly and positioning of various aircraft components onto an airframe, such as fuel tanks, landing gears and control surfaces. The automation was coupled with genetic algorithm (GA) to provide the optimal arrangement. Patran was used to verify structural integrity of the iterated options(7). Multiple examples exist, where the researchers have focused on the overall configuration for aerodynamic purposes(8,9). These have different levels of automation and flexibility, some focus on specific questions, others develop flexible solutions for everyone to use. One recent example of this is De Marco et.al. (9) who have developed a CAD environment in which automatic generation of aircraft main geometries is possible. The environment is developed using Java on open-source foundations called OpenCascade, the same ones CATIA was originally developed from.

All of these are potential extensions of automated system of simulations. Once the key components are automatically optimised and analysed, full digital twin of the final product, e.g. aircraft, becomes possible.

One of the reasons why only limited number of researchers and engineers use CATIA automation is the lack of learning materials. The only good quality tutorial for CATIA scripting found is a lecture material from Schor (4), from Brno University of Technology. The documentation provided by Dassault is useful for troubleshooting but is not suitable for initial learning.

2.3 Structural finite element models

Structural optimisations in Abaqus, Ansys, Patran, Hyperworks and many others are quite well documented and often used, even when it comes to the more complex composite optimisations, for example dealing with layups(10–15). When it comes to script-ability Abaqus has a significant advantage, due to a well-developed and documented python interface. Many people have used python to aid with optimisations in Abaqus(11), which can be apparent even just from the Q&A on various engineering forums.

A major difference exists between the modelling of static stiffness and more complex phenomena such as failure, local damage, stresses, and fatigue. The more complex structural problems are still subject to extensive research, therefore for the SySi only stiffness analysis shall be considered. This is due to the calculations of material properties, which are considered in separate section of this literature review.

2.4 Meshing

Meshing is an important part of any simulation. Coarse mesh might lead to imprecise results, while having an excessively fine mesh extends the simulation runtime, sometimes even preventing effective optimisation(16).

Meshing is often done within the simulation software, which is why meshing is not addressed at all in most simulation papers. Meshing is usually difficult to automate reliably and can be very time consuming to adjust manually. Also, it might be practical, when running multiple simulations, to use similar mesh for all of them.

Other option is to mesh in a dedicated meshing software. These are not frequent in commercial space, as in that case they would be integrated within the software provided by the same company. Hypermesh by Altair is one of the few meshing software often used for exporting meshes even to competitor's software, such as PAM-RTM by ESI.

In the open source space, the most notable meshing software is probably GMesh, which is sometimes being used for meshing input to engineering simulations(17).

Another potentially useful meshing technique is mesh morphing. Mesh morphing is in effect repositioning of individual nodes in 3D space. This is done instead of generating a new mesh for next iteration of a part. This is useful mainly for simulations where the part itself changes shape in a complex manner(18), such as where second aerodynamic simulation is run with deflected part.(19–21)

One of the benefits of mesh morphing is that the effect of parameter change is not affected by any change new mesh could cause. Mesh morphing also typically decreases re-meshing runtime, as it is usually faster to change 3D coordinates of nodes than to re-create the complete mesh(19).

Several modern software include some sort of mesh morphing algorithm, for example Altairs Hyperworks, or Ansys Fluent(19). There is also open-source software that has the capability(18).

2.5 Infusion simulations

Infusion simulation is quite common in academia and generally as a research topic. However, only recently there is evidence of commercial companies adopting infusion simulation as part of the design for manufacturing process. The simulation of simpler LCM methods, such as VARTM, has been validated many times, and recently even more complex behaviours such as compaction are sometimes simulated(22).

Infusion simulation is usually used to verify an infusion setup to minimize any trial-and-error manufacturing. The inlets, outlets, temperatures, and other parameters are adjusted until the simulation suggests complete infusion. Often optimisation of these parameters is done to achieve fastest infusion, to minimize the manufacturing time(23–28).

The optimisations related to the infusion simulation are usually done in terms of the cure profile, rather than using all the parameters used in the simulation software. For instance, Struzziero et.al. optimised the cure cycle based on dwell and ramp times and temperatures, while the size parameters were based on literature(23). The gate location is kept constant within the optimisation run. The PAM-RTM is iterated for that optimisation by modifying the input file and re-running the simulation (23).

Some attempt of optimisations in PAM-RTM and LIMS have been reported(28–30), other software is rarely used(29). These optimisations are usually done by a manual re-run(31) of simulations or by scripts generated directly for that particular software. These scripts are usually designed to adjust input file(23), rather than to control the software itself. The iteration is then usually triggered by the command line. This leads to various limitations, especially when parametrized/variable part input is required. Further integration with other aspects of component design are rare. An attempt for CAD and infusion simulation integration exists, Dassault and ESI have developed a CATIA PAM-RTM workbench (32). Author did not find any evidence of this being used in industry or academia.

Sometimes parameters are fed back to simulation from the manufacturing. This allows adjustment of flow rates for a more robust infusion. This is a good example of moving toward utilization of digital twin for manufacturing. Some attempts to map geometric and material data to connect FE models has been done(30,31).This will be more closely discussed in the material property prediction section of this literature review.

Both LIMS and PAM-RTM are scriptable. For either, limited number of examples is available. Although some papers report utilization of such script, these are not widely shared and the documentation allowing for scripting development is minimal. This is made even more difficult in case of LIMS as the scripting is done in lesser known scripting language, LBASIC(33). PAM-RTM is scriptable using python, but minimal guidance exists.

2.6 Cure simulation

Cure simulation is a common input to, or component of, the infusion simulation. Cure simulations have been around for a significant amount of time, as evidenced by papers from as early as 1983(34). Nowadays, cure simulations are undertaken for variety of reasons: to minimize cure time, manage temperature overshoot in the curing process, as an input to infusion simulation, or to predict the residual stresses in the part(23,35,36). Most of the optimizations that include cure simulation focus on optimizing the cure profile, which is quite specific for each resin system(23,37,38).

In the System of Simulations, cure simulation could most likely be paired with infusion simulation to broaden the scope and provide more detail to any multi-simulation optimisation.

Of-the-shelf software is available for 3D cure simulation, for example LMAT which interfaces with both Abaqus and LS-DYNA(35), or COMPRO that interfaces with Abaqus and Ansys(39). 1D cure simulation is also quite common, and made simple by readily available simulation tools such as “SimpleCure” developed at the University of Bristol and NCC (40,41). 1D cure simulation is also likely more suitable for implementation in SySi like projects.

2.7 Braiding simulations

There are two main groups of braiding simulations: kinematic and final element.

Kinematic simulation is based on position of the spool and last known position of mandrel contact. Through iterations and basic vector operations single yarn is projected on the surface of the mandrel. As only one yarn is typically considered, the kinematic simulation inherently ignores interactions between yarns. Friction between yarns and guide ring is also neglected. Yarns are usually assumed to be fixed after first contact with the mandrel, although effect of yarn slip can be included in kinematic braiding simulation. Trajectories of deposited lines are continuous. (42–44)

Finite element analysis attempts to model all the interactions, by creating a complete model of the system, including all the yarns and all relevant machine parts. This allows for considerations of friction, both yarn-yarn and yarn-mandrel. However, FE braiding simulations are very expensive in terms of computation time. The comparison between various types of braiding simulations is shown in Figure 2.1. (42,45,46)

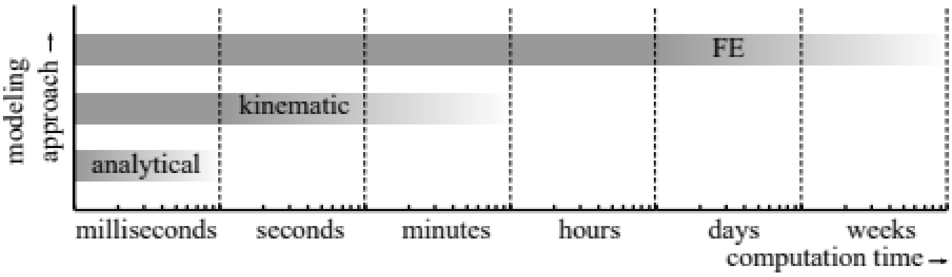


Figure 2.1 - braiding simulation comparison (42)

For the purposes of SySi kinematic simulation is clearly more suitable. The looped design does not aim to provide high fidelity solution to all the individual simulations but is required to run many times to optimise the parameters. Running a full FE simulation is therefore entirely unfeasible. In a design office using SySi, FE simulation might still be used at the end of design process to verify the results, or to tweak the machine settings. However, when FE simulation is run most of the parameters already need to be settled as it is inefficient running it many times.

2.7.1 Kinematic braiding simulations

In terms of commercially available kinematic simulation software, the most notable is the CATIA CBX module. It is a CATIA plugin that provides basic braid model and information for mandrels of reasonable complexity. This software still has several technical issues but should be sufficient for parameter selection before manufacturing. However, the main problem is the lack of scriptability. None of the methods are fully disclosed by CATIA, and none of the simulation can be automated, preventing it from being used for the SySi. The simulation is generally not suitable for optimisations.

For circular mandrels with constant cross sections, open source GitHub project exists(47). It does allow for modelling of various braid patterns. However, due to the lack of capability regarding more complex geometries this is not particularly suitable for SySi either.

It appears none of the publicly available solutions fully satisfy the requirements for system of simulations; i.e. allow for variety of mandrel shapes, and are scriptable and sufficiently robust for iterations.

Quite a lot of resources exist, that describe the braiding process or that outline the advances in FE analysis(48–50). However, minimal resources exist that would provide practical information required for the development of a kinematic simulation. The most comprehensive information about kinematic braiding is probably provided by Ravenhorst, in his PhD thesis (42). It provides a good review of previous models, and in detail describes the development of kinematic braiding simulation. Outtakes of this work will be referenced in braiding simulation section, as it informs the development of a simple kinematic simulation used in SySi.

2.8 Material property prediction

To utilize the benefits of connecting the various simulations, the data from manufacturing simulations must be transferred into the structural analysis and between each other. The two most relevant translations, for the version of system of simulation presented in following sections, were related to braiding. Permeability must be predicted, based on braid angles and pitch between yarns, to inform the infusion simulation. The structural properties must be also estimated from the braiding simulation data, to provide structural analysis for each iteration of design.

2.8.1 Mechanical Properties

The prediction of mechanical properties for any composite manufacturing process is quite complex, and for most processes subject to ongoing research. In general stiffness predictions can be quite reliable, while failure criteria are usually significantly more complex(49,51).

When it comes to braiding, the research into both of these areas is not as advanced as for some of the more common manufacturing processes. However, the research follows similar pattern as for woven fabrics. Multi-scale modelling is usually required(46,50–56). First, unit cell of the yarn structure is modelled, and analysed using finite element methods. Then the obtained material properties are applied to homogenous regions on macroscale, where the whole part is considered. Many researchers use weave structure modelling software TexGen(57), developed at University of Nottingham(58). Another common software is WiseTex(59), developed at Katholieke Universiteit Leuven (60). Both TexGen and WiseTex have growing community of people outside of their original universities that develop on top of the core software. One example of this is the “Composites Dream” which is University of Osaka’s software that uses WiseTex for the yarn structure modelling(56,61). Other yarn modelling methods are also reported, but usually on individual basis; for example standard version of CATIA has been used to model yarn level microstructure(62). All of these methods can be quite computationally demanding (51) and require more research to improve their reliability and precision. There are other similar methods that follow the same logic, but without the use of the textile modelling software(63–65).

Alternatively, classical laminate analysis based method has been reported to correlate well with experimental results(49,66–68). However, this was done only for very simple geometry and can only predict tensile and compressive stiffness as it neglects the weave of the structure and approximates the composite as layered. This is clearly a very basic method that requires a lot of assumptions but can be quite useful for an initial estimate required for SySi, as it will require minimal computational time.

Neglecting the weave makes it quite impossible to predict the poison ratio, which is significantly dependent on the type of weave, and the current elongation(69). Therefore, it is important to check how sensitive the particular model is to changes in poison ratio. If the model is sensitive it might be better to create a lookup table from existing poison ratio experiments, such data is provided for example by Shahabi et.al.(69).

2.8.2 Permeability

The main benefit of infusion simulation is typically the prediction of issues that might prevent complete infusion. This is particularly important for larger, more complex parts, where simple intuition about the flow will not suffice when designing the infusion strategy.

One of the most important parameters for the infusion simulation is the permeability. On a curved, layered or in any other way non-simple part, the permeabilities will differ across the preform. This is also the case for braiding. The permeability will differ in all 3 directions based on the braid angle, the local cross section, and the pitch of the fibres.

The typical approach for predicting permeability is very similar to the predictions of mechanical properties. The weave is first modelled on yarn-by-yarn basis, and the model is then analysed in flow simulation, typically a finite element model(70,71). Commonly used software is WiseTex, with combination of various flow simulations such as FlowTex(72), Ansys(72,73) or OpenFoam(31). TexGen is also being used to construct models leading to prediction of permeabilities(71,72,74). For instance, Swery et.al. have developed a tool that predicts permeabilities from TexGen model. However, this model requires several intermediate steps involving Matlab, Abaqus, Ansys CFX and HyperMesh(74). This research has provided insights into what aspects of braided component significantly affect permeability. However, the precision of the models, and hence their usefulness for analysis of production components, is not reported on.

It is typical that intra-tow permeability is neglected(70).

The reliability of braiding simulations is also still in question. Therefore, in some cases it might be more practical to collect actual data from manufacturing and run the infusion simulation based on that. Several researchers have already looked at obtaining braid angle information from braided preforms using a camera image processing methods(75,76). Varying levels of light intensity can be used to track the local direction of fibre. These methods could prove quite useful if corresponding method of automatic permeability assignments is developed in the infusion simulation environment. The measured permeability data could then be coupled with the data from braiding assessment. Multiple researchers have collected permeability data from manufacturing(77,78).

It is difficult to predict if the data-based approach or simulation-based approach will turn out better. Combination of these two approaches has been attempted in various related fields.

Some attempts for analytical predictions of permeability of woven fabrics exists (79), some even specifically for braided composites. Closest to the requirement of SySi is Endruweit et al. who presented a method for calculating the permeability of a triaxially braided composite(80). This takes in account multilayer effects and compression.

In summary, nothing off the shelf is currently suitable for SySi, but several streams of research are being worked on.

2.9 Testing in Aerospace

Testing in the aerospace industry makes sure that the designed solutions are safe. However, testing is very expensive. Therefore, the testing procedures are aiming to provide the most cost-effective way of guaranteeing safety.

The most common approach is referred to as the testing pyramid or a building block approach. This is most characteristic for the aerospace industry, but it is being adopted for composites outside of aerospace as well(81,82). Testing composites has some inherent randomness in it. Hence, to make sure material properties measured are usable, many samples should be collected. On the other hand, the final product, an aircraft, is very expensive and therefore testing the final product to failure repetitively is not an option.

The testing pyramid generates the confidence iteratively. First a small material sample is created and tested, this provides the basic material properties, for the given constituents and material process. This is relatively cheap, therefore larger number of samples can be created to have a statistical distribution of the material properties. Guidelines exists on how to turn the testing data into usable material properties, accounting for statistical variation(83). The material properties are then applied when modelling a specific shape that is to be manufactured. Few of these can be created and tested with representative loads. The testing data are then used to validate the created part models. This progresses in a similar manner through subassemblies all the way to final product, where typically one full aircraft structure is tested to failure in fatigue. Throughout the pyramid more and more comprehensive models can be created, collecting the confidence from smaller more specialized models.

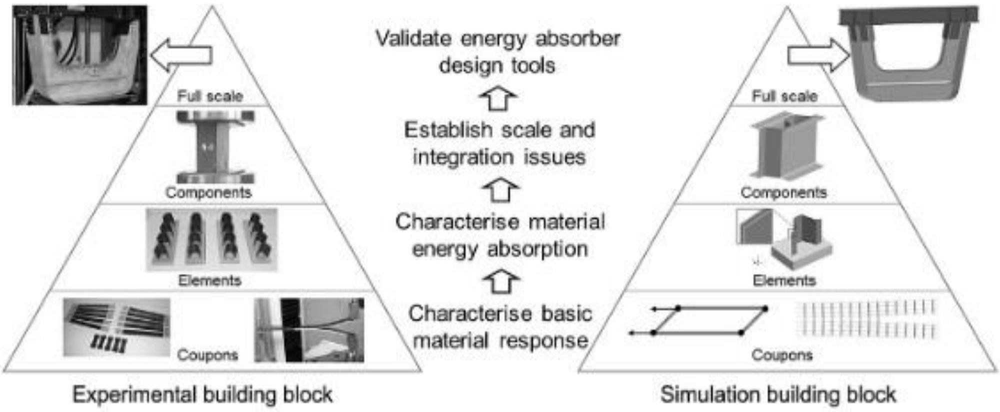


Figure 2.2 – testing pyramid (84)

The need for large amount of testing is addressed by a variety of efforts, most notably by the CMH-17 Composite Material Handbook, which aims to standardise the engineering data required(85,86). Some of the newer methods and efforts look at leveraging data and modern statistical methods to decrease the amount of testing required(87).

The proposed solution in this Thesis does not invalidate these approaches. Each of the models involved in an industrial SySi still has to be validated in this way to generate confidence. However, the collection of comprehensive well controlled data might allow for some cost savings. The bottom of the testing pyramid might be the same for multiple solutions. If the data is collected in well traced reliable manner, this data might be reused to support test pyramid for the next part or solution. For example if the matrix and fibres are the same, the coupon testing might still be relevant for a new part.

In chapter 4.1.3 a methodology of simplifying data collection from ASTM standard testing is outlined. This could allow for standardisation of data formats and provide additional incentive to begin trading testing data within the industry, some of which is already happening through NCAMP(88).

2.10 Defects

Some manufacturing defects are inherent aspect of composites. Some defects are unexpected, such as a piece of backing paper between plies; these defects need to be avoided by good process design and cannot really be addressed by simulation solutions. However, there are also defects that are inherent to chosen material process and design.These will always exist in some size and frequency, including for example porosities and voids, fibre waviness, or out of plane wrinkles. This is due to the inherent variability of the reinforcement material and the complexity of the manufacturing processes.

It is impossible to design parts with no defects being allowed. The usual approach is to specify the defect size or quantity that is allowed. During inspection there are going to be specific defects in specific locations that will lead to part scraps.

The work described in this Thesis does not directly address defect issues but could eventually support implementation of some of the solutions being developed. For instance automated defect identification methods are being developed, using image recognition algorithms(89–91). These could be integrated with the toolchain. Due to the automated models, it would be quite easy to map any identified defects back to the structural analysis. The focus on data collection also goes hand-in-hand with this effort. Similarly in service defect inspection can be used as an adjustment input for any structural analysis models.

The properties that should be attributed to these defects is an active area of research as well(92). One of the options is to create representative volume elements similar to what is described in "material property prediction" section of this literature review. Except this time larger number of options needs to be created, accommodating for the variety of defects. Alternatively, specialized testing can be used to evaluate the effects of the defects, which can be turned into knock-down factors(87,93).

Identifying common defects and linking it to some design features, could also lead to informing future designs, minimizing the defect occurrence by changing the design of the part itself(94).

2.11 Digital Twin

Various aspects of SySi overlaps with various efforts referred to as "digital twin". This section briefly outlines the core aspects of what is typically called "digital twin", and specifies its relation to SySi.

The most commonly cited ***benefits*** of Digital Twins are: optimized MRO (95), minimize manufacturing errors and improve quality(95,96), warranty/fatigue information(96,97), better tractability of work(96), faster prototyping and new product development(96,98), improved identification of bottlenecks (96), virtual training(99), factory flexibility(98). One of the sub-topics are the 4.0 factories, the main aspects of these are automation, physical system offloading, mass data collection, processing, usage, and commercialization(96,98–102).

Main ***barriers to entry*** are probably the costly collection of data and lack of data cohesion. Lack of relevant skills in the workforce might also become an issue(101,103,104).

A possible ***approach*** to digital manufacturing is to focus on description of modules through interfaces. Modules allow for identification of high yield improvements. Interface considerations make sure that the modules can be designed with integration in mind. Interface development allows for sharing of data and high efficiency of its usage. It can be determined which modules are crucial for overall development; interdependence along with immediate effect evaluation can help determine which modules should be targeted first. Current manufacturing practices, such as lean or preventative maintenance, are good intermediate stepping stones, or prerequisites, for gradually implementing digital manufacturing.(101,105,106)

Several ***roadmaps*** towards digital manufacturing have been produced. The most comprehensive and extensive seems to be the "German standardization roadmap, Industry 4.0"(107). It outlines the requirements for standardisation with relevance to industry 4.0. However, this roadmap is quite high-level government policy-making outlook, looking at concepts rather than required steps and industrial examples. Other attempts are typically even less specific. Step-by-step practical, technical, guide for implementing a digital factory or other digitization efforts is still missing. Although many of these provide additional motivations to the development outlined in the thesis, there is minimal information that could aid with the development of the technical solution itself.

Many different definitions of Digital Twin exist, all of them focus on how new digital models can improve manufacturing, design, or simulation, with varying degrees of complexity and integration.

The SySi fits somewhere under Digital Twin, as a subtopic. The cited benefits reveal that many of the goals are essentially the same: faster prototyping and product development, better traceability of work, minimized manufacturing errors, etc.

2.11.1 Integrating Simulations

Integration of different simulations can be considered as a separate topic or a subtopic of digital twin, as it is complimentary to the digital twin ideas. This section focuses on the research efforts that are closest to the SySi itself.

Composite manufacturing is inherently a multi-simulation problem, where structural analysis results will be influenced by the fibre positioning which is often computed by a different simulation. Fully interconnected system of simulations could therefore provide various benefits. Multi-objective optimizations are easier to execute, as optimization loop can be executed with multiple simulations connected, sharing parameters. Redesign of the component can be largely automated. The digital twin idea can be reinforced by real-time manufacturing data feeding back into the simulation which allows for property prediction for each component manufactured.

Currently most research on integration of simulations has more specific goals, such as including lifecycle costs in design evaluation(108,109), combining meso and macro scales models to improve permeability predictions(110), using draping information to reinforce the infusion simulation(111), supporting the design of geometry with manufacturing and stress models(112–114), or using the material simulations to inform the structural analysis(51). Outside of the composites engineering, people have integrated the aerodynamic and the structural optimisations, or the assembly with the structural analysis(7). Multiple German institutions cooperate on project ARENA2036, part of Industry 4.0 initiative. DigiPro, subproject of ARENA2036, aims to create a close loop process chain simulation. However, the reports available suggest that a lot of the individual efforts focus on integration of 2-3 models in quite significant detail, not outlining the intended overarching architecture(115–117). Another German effort worth mentioning is the Agile 4.0 project, which seems to be tackling the problems described in this thesis(118). Similarly, NASA has produced “Roadmap for Integrated, Multiscale Modelling and Simulation of Materials and Systems”(119) which aims to provide a comprehensive list of steps required for an integrated simulation system. The study has been developed through surveys, workshops, and a validation exercise, but so far seems to lack practical use. The main aspects of simulations that the roadmaps highlight are modularity, flexibility, accessibility of data, reusability of test data, interdepartmental cooperation, and inclusion of uncertainty in simulations.

CDS (composite design and simulation) from university of Delaware is addressing the issue of disconnected software. The package supposedly offers calculations of effective properties and response of composite laminates, conducts micromechanics calculations, as well as virtual process simulation and optimization. This seems like detailed analysis of a particular laminate, but without the design aspect and the automation (120,121). Another similar project is the already mentioned COMPOSITES-DREAM. This brings together couple of textile analysis software from KU-Leuven (Wise-Tex, MeshTex), with some Osaka University's software and LS-Dyna and Abaqus(56). These efforts do work towards creating a chain of simulations, but they do not seem to offer a pathway towards integrating more aspects into their software, or for external participation.

There are several companies which are clearly good at providing specific high-quality software. The Dassault's software (Catia, Abaqus, Enovia) seem to be regarded as the most robust and are often used for high-end engineering. However, although these software work very well together they have two major issues. Firstly, the software is very expensive, making it less accessible to smaller companies. Secondly, various aspects of the software and their functionalities are proprietary and cannot be easily reviewed. This causes problems with integrating other software and with attempting to improve on specific aspects of some simulations. Also, Dassault is unlikely to ever focus sufficiently on specific composites modules due to limited demand. This is not just specific for Dassault Systemes, but applies to most other software providers.(122,123)

All the software companies are aware of the value of integrating various simulations, this can be particularly evident from all the different PLMs. Dassault houses their software under 3DExperience platform, which pulls together all their software, aiming towards full Digital Twin capability(122). ESI houses several manufacturing simulation tools under PAM-Composites, allowing for seamless transition between these(124). Siemens also seems to be putting a lot of effort in developing holistic digital solutions(106,125). This is not an exhaustive list(126). However, the focus of these PLMs is still on selling their software, so there seems to be a lack of incentives for integrating software made by others. There are some exceptions were Dassault has integrated Siemens's Fibresim and made a brief attempt at integrating PAM-RTM by ESI(32). Another downside of these commercial solutions is the lack of scriptability. In the effort to conceal as much of their proprietary code as possible it is often difficult to plug in bespoke modules and control certain software from outside scripts. This is quite important for full integration solutions as no generic commercial software will ever satisfy all specialized needs of a company at the forefront of technology.

It is likely that some large engineering corporations have developed their own software, but these are usually not disclosed. Airbus have outlined the basic idea of their Lagrange project, a multi-simulation optimisation tool, with thousands of design parameters aimed to optimise wing structure for both aerodynamics and structural efficiency(127).

In summary all these companies understand the requirement for interconnectedness and flow of simulations. However, they all seem to believe that their company will provide all the solutions, which currently does not appear practical.

Table 2.2 – integration of simulations efforts

	Institution /company	Software included	Main purpose
3D Experience	Dassault Systemes	Catia, Abaqus, Enovia, Isight, …	PLM, optimisations
PAM-composites	ESI	PAM-RTM, PAM-FORM, PAM-DIST…	Simulation of complete manufacturing
Siemens PLM	Siemens	NX, Fibresim, Heeds	PLM
MindSphere	Siemens	N/A	IoT operating system, data management
MSC Apex	MSC (Hexagon)	Digimat, Patran, Nastran…	Unified environment for virtual product development
HyperWorks	Altair	HyperMesh, OptiStruct, ESAComp…	Unified environment for virtual product development
CDS	University of Delaware	LS-Dyna, …	Laminate analysis, software integration, optimisation
COMPOSITES-DREAM	Osaka University	LS-Dyna, Abaqus, MeshTex, WiseTex	Combining textile analysis with structural models
Lagrange	Airbus	Not disclosed	Combining aerodynamics and structures, optimisation

2.12 Optimisations

There are many different optimisation algorithms available, the table 2.3 summarizes the most common ones, that were considered for SySi.

Table 2.3 – most common engineering optimisation algorithms

	Discrete variables	Continuous variables	Escaping local minima	Complex problems	Convergence	Available library
Genetic Algorithm (GA)	Yes	No	yes	sometimes	slow	Easily programmed
Gradient descent	no	yes	no	rarely	fast	Easily programmed
Ant Colony optimisation	Yes	No	Yes	sometimes	medium	ACO-Pants
Differential evolution	no	yes	yes	yes	medium	SciPy

The algorithms in the table are provided as an overview, the columns are only true in their basic form. Naturally, many of these have many iterations that significantly change how these would be classified in the table.

2.12.1 Sampling methods

Significant amount of understanding can be obtained purely by sampling a design space. This sample can then be used to build surrogate model or provide population for first iteration of genetic algorithm (GA)(128). The quality of the sample can have large effect on any following surrogate optimisations(129).

Probably the most common design space sampling method is Latin Hypercube Sampling method (LHS). It provides a randomised sample of different variable combination, while maintaining high level of representation from all areas of design space. Functioning LHS project was found on GitHub.(130,131)

Sometimes LHS method may not always sufficiently fill the design space. Sequential sampling approaches such as “multi-fidelity competitive sampling” can be employed to improve this(129).

2.12.2 Gradient optimisations

Gradient optimisations are quite common in various engineering fields. These can be quite efficient at finding minima with few continuously defined parameters(132). However, workarounds are required for working with discrete variables and these might struggle to find global minimum in complex design space.

When used for composite optimisation it is usually coupled with another optimisation. Gradient optimisation can, for instance, be used to further improve on a resulting set of parameters from genetic optimisation(133).

2.12.3 Genetic Algorithm (GA)

Genetic algorithm is one of the most popular algorithms used in recent years, for quite varied list of tasks. GAs work very well especially for discrete variables.

GA mimics the passing of genes between individuals in population. It consists of repeating steps of evaluation of individuals, selection of individuals, genetic crossover of selected individuals, and optional step of mutation. It is particularly good algorithm where non-continuous variables are used.

Various different versions of GAs exist, specifically tailored for the particular task. Momentum offspring method can decrease the time to reach optimum. Constant range mutations allows simulations to escape local minima by preventing immediate disappearance of most mutated individuals(134).

In composites it has been used to optimise various manufacturing processes and multiple different aspects of component design.

Laminate analysis with many parameters was optimised using a GA by Fontecha et.al.(135,136), multi-scale modelling and equivalent single layer method were used. GA is often used for optimisation of inlet positions for RTM simulation, typically in LIMS or PAM-RTM software(24,27,137). Multiple researchers use GA for optimising cure cycle, in order to support an infusion optimisation(23,24). Park et.al., uses GA for multi-objective optimisation simultaneously investigating mechanical performance and costs of a component. Bisagni et.al., for post-buckling optimisation of composite panels(15). The list of GA uses for composites optimisations is quite inexhaustible. GAs are also often used in conjunction with other methods to act as an initial sampling model(15,24).

Genetic algorithms are typically slow due to the large number of required evaluations(134). However for very specific tasks GAs can have comparable efficiency to other algorithms(23).

2.12.4 Differential evolution

Differential evolution is a stochastic method that is good at searching large design space for global minima, but usually requires more iteration than gradient based methods. It is available as part of SciPy Python library.(138)

It has also been used in some composite optimisations(139).

2.12.5 Surrogate modelling

Surrogate models are very popular nowadays for addressing complex multi-variable problems, which are difficult to understand. These depend on robust sampling method, such as the LHS, to generate initial dataset. Then prediction function is generated which best fits the sample set. This function is then used for optimisations, usually in combination with another optimisation algorithm(128). Computationally expensive multi-disciplinary optimisations can become viable, through surrogate models(140).

In the field of composites, surrogate model has been used for example in optimising RTM infusion parameters(141), or to optimise stacking sequence (129). Along with the optimisation usage, surrogate models are also used for other computationally demanding tasks such as predicting bond strength, or inter-ply shear strength, in dependence to process parameters(142,143).

The scikit-learn is an open-source library for python with good documentation and robust implementation of various surrogate models. The library includes several regression algorithms; SGD regressor for smaller samples, Lasso method for large sample with few important features, and SVR for large samples with many features.(144,145)

2.12.6 Other relevant optimisation notes

Other optimisations that are not that common in composites but might be worth reviewing for the purposes of SySi, are for instance particle swarm optimisation (PSO), or ant colony optimisation (ACO).

Traceability and reusability of data is of significant importance in modern optimisations(146).

Several commercial optimisers exist, typically coming with other engineering software. For instance, Isight is Dassault Systemes's optimiser software focusing on passing parameters between their software, further promoting connectivity and multi-simulation optimisation. Hexagon, the owner of MSC software, also has their optimisation software "Odysee" which is used by Audi. Heeds is a relatively new addition by Siemens. Various lesser-known integration software from other companies are also available. It is difficult to assess the quality and usefulness of these commercial software without trying each of them.

2.13 Aerodynamic analysis software for optimisations

Standard aerospace optimisations can be quite time expensive. Therefore, the focus here is to find aerodynamic optimisation options that can be run fast and therefore be part of the initial design iteration loop.

There are several degrees of aerodynamic analysis. The most comprehensive, which provides most complete output, is a full computation fluid dynamics model, CFD. However, these models are usually complex and take a long time to run. For any aerodynamic part this will likely be used at some point during development. However, this will likely be either stand-alone optimisation, or just a one analysis based on already selected parameters.

For early design stage analysis, less complex models are usually used to provide estimates of various aerodynamic parameters. The most notable ones are coefficients of lift and drag, or the lift and drag distribution, depending on the purpose of the analysis. Most common methods are vortex lattice method (VLM)(147), and lift line theory (LLT). The VLM estimates lift and induced drag, ignoring thickness and viscosity. The LLT takes in account the aerofoil slices and their interaction. It is debatable which one is better for what type of application. However, both are suitable for implementation in the SySi.

Topology optimisations are also often done. These can be coupled with structural analysis to assess bend-twist coupling interactions or similar. However, this will likely be also unnecessarily complicated for the demonstrator addressed in this thesis.(146)

Large number of open-source software exists for these basic aerodynamic assessments, mainly thanks to the model aircraft community. These software include: XFLR5, Open Vogel, AeroPy, AeroSandbox, OpenFoam, Su2, PyFR, SUAVE, Code Saturn, HiFiles, OpenVSP, AVL, ADflow, Tornado, VSAero, Matrix-V, Q3D. These have various levels of quality and reliability which are difficult to assess purely from literature. Many of these lack any kind of scriptability which makes them unsuitable for SySi.(146–153)

XFLR5 for instance would be ideal with its functionality but fully functional python API does not appear to exist yet(148,154,155), although some people have attempted it’s development (156).

OpenVSP also seems to be working quite well for many people, and based on its documentation it has a python API (157).

AVL is used in variety of optimisation works, or as initial estimate for decision making in concept stage of product development (147,150,158,159).

SUAVE is a conceptual design tool that can be connected to different optimisation packages, and even to some other aerodynamic analysis tools such as AVL. It can be used for other than aerodynamic optimisation, such as mission planning, potentially allowing for even more holistic optimisation. (152,153,155,160)

Xfoil is also a potential candidate as it has python API developed (151).

Assessment of the selected software will be discussed in later sections of this report, as these are subject to trials. The literature review itself is not sufficient to make a decision on the software to be used.

Commercial software used for aerodynamic optimisations usually employs CFD, this can be done for instance in Ansys FLUENT (161). It appears that because there is large number of open-source software for the lower-level aerodynamic analysis there is no space for commercial software.

2.14 Literature Review Summary

Simulation is becoming a major topic in all aspects of composite component design. Hence large number of papers is continuously being produced on both process simulation and performance simulation. Some of these simulations, such as structural stiffness-based analysis, are well developed and can be used as a plug-in into the SySi. However, other simulations, such as braiding, are less developed and will require a significant development both in terms of capability and scripting. Improvements to all of these are constantly being developed at several institutions at once, with various approaches.

What is being addressed to lesser extend is the interconnectedness of these, the overall design process and how it corresponds to customer specifications. This can include efficient ways of translating material properties between simulations, collection and reuse of data, automation of different software or the process of design itself.

Large number of institutions and, above all, commercial software providers talk about integration, digital twin, and digitization as a whole. However, evidence of practical attempts to resolve some of the issues are less common.

Some ideas about which software should be used for the SySi has been gathered in this literature review, but further testing might be required to establish the suitability of each software. It is unclear if the commercial or the open-source way is more suitable.

References

1. ATI. Technology strategy and portfolio update 2016. 2016.

2. ATI, Fathom. Raising Ambition for UK Industrial Strategy, Lessons from Innovation in Aerospace. 2016.

3. ATI. New application process for UK Aerospace Research and Technology Funding - Aerospace Technology Institute [Internet]. 2018. [cited 2018 Apr 12]. Available from: http://www.ati.org.uk/2018/01/new-application-process-uk-aerospace-research-technology-funding/

4. Schor P. CATIA V5 scripting with python and Win32API [Internet]. Brno; 2012. Available from: http://www.lu.fme.vutbr.cz/~schor/data/CatiaScripting1.pdf

5. Redstone J. Solidworks Macros via Python [Internet]. 2015 [cited 2021 Apr 29]. Available from: https://joshuaredstone.blogspot.com/2015/02/solidworks-macros-via-python.html

6. Zbiciak M, Grabowik C, Janik W. An automation of design and modelling tasks in NX Siemens environment with original software - generator module. In: IOP Conference Series Materials Science and Engineering [Internet]. 2015. Available from: https://www.researchgate.net/publication/283634416_An_automation_of_design_and_modelling_tasks_in_NX_Siemens_environment_with_original_software_-_generator_module

7. Yang W, Yue Z, Li L, Wang P. Aircraft wing structural design optimization based on automated finite element modelling and ground structure approach. Eng Optim [Internet]. 2016 Jan 2 [cited 2019 Mar 20];48(1):94–114. Available from: http://www.tandfonline.com/doi/full/10.1080/0305215X.2014.995175

8. La Rocca G, L van Tooren MJ. Knowledge-Based Engineering Approach to Support Aircraft Multidisciplinary Design and Optimization.

9. De Marco A, Di Stasio M, Della Vecchia P, Trifari V, Nicolosi F. Automatic modeling of aircraft external geometries for preliminary design workflows. Aerosp Sci Technol [Internet]. 2020 Mar [cited 2020 Jan 28];98:105667. Available from: https://linkinghub.elsevier.com/retrieve/pii/S1270963819316943

10. Fontana M, Araujo AL, Madeira JFA. Optimization of thin-walled composite crash absorber. Thin-Walled Struct. 2020;155.

11. Kaveh A, Dadras EA, Malek GN, Ansari R. An open-source computational framework for optimization of laminated composite plates. Acta Mech. 2020;231.

12. Imram M, Shi D, Tong L, Waqas HM, Muhammad R, Uddin M, et al. Design optimization and non-linear buckling analysis of spherical composite submersible pressure hull. Materials (Basel). 2020;13.

13. Zhou L, Xue Y, Gao S. Theoretical Analysis on the Flexural Properties of Spliced Composite Slim Beam. IOP Conf Ser Earth Environ Sci. 2020;455.

14. Sousa CS, Camanho PP, Suleman A. Analysis of multistable variable stiffness composite plates. Compos Struct [Internet]. 2013 Apr 1 [cited 2018 Mar 9];98:34–46. Available from: https://www-sciencedirect-com.bris.idm.oclc.org/science/article/pii/S0263822312005545#f0005

15. Bisagni C, Lanzi L. Post-buckling optimisation of composite stiffened panels using neural networks [Internet]. [cited 2020 Feb 4]. Available from: www.elsevier.com/locate/compstruct

16. Dong C. Injection Gate Definition for Improving the Accuracy of Liquid Composite Molding Process Simulation. J Compos Mater [Internet]. 2007 Aug 27 [cited 2020 Feb 18];41(15):1851–70. Available from: http://journals.sagepub.com/doi/10.1177/0021998307069898

17. Dobri A, Wang Y, Papathanasiou TD. Transietn heat transfer in fibrous multi-scale composites: A semi-analytical model and its numerical validation. Numer Heat Transf Part A Appl. 2020;77(9).

18. Ertl F-J, Dhondt G, Bletzinger K-U. Vertex assigned morphing for parameter free shape optimization of 3-dimensional solid structures. Comput Method Appl Mech Eng. 2019;353:86–106.

19. Baghdadi M, Elkoush S, Akle B, Elkhoury M. Dynamic shape optimization of a vertical-axis wind turbine via blade morphing technique. Renew Energy. 2020;154.

20. Wang W, Wu Z, Wang D, Zhang W, Zhau K, Okolo PN, et al. Hypersonic vehicle aerodynamic design using modified sequential approximate optimization. Adv Sp Res. 2019;63(11):3706–20.

21. Wei D, Seymour MJS, Kareem A, Jemcov A. A structured mesh boundary motion approach for simulating wind effects on bluff bodies with changing boundaries. J Wind Eng Ind Aerodyn. 2014;126:119–31.

22. Rubino F, Carlone P, Rubino F, Carlone P. A Semi-Analytical Model to Predict Infusion Time and Reinforcement Thickness in VARTM and SCRIMP Processes. Polymers (Basel) [Internet]. 2018 Dec 24 [cited 2019 Jun 26];11(1):20. Available from:

http://www.mdpi.com/2073-4360/11/1/20

23. Struzziero G, Skordos AA. Multi-objective optimization of Resin Infusion. Adv Manuf Polym Compos Sci [Internet]. 2019 Jan 2 [cited 2019 Aug 20];5(1):17–28. Available from: https://www.tandfonline.com/doi/full/10.1080/20550340.2019.1565648

24. Gupta A, Kelly PA, Ehrgott M, Bickerton S. A surrogate model based evolutionary game-theoretic approach for optimizing non-isothermal compression RTM processes. Compos Sci Technol [Internet]. 2013 Jul 29 [cited 2019 Jun 18];84:92–100. Available from: https://www.sciencedirect.com/science/article/pii/S0266353813002121

25. Baskaran M, Ortiz De Mendibil I, Sarrionandia M, Aurrekoetxea J, Acosta J, Argarate U, et al. MANUFACTURING COST COMPARISON OF RTM, HP-RTM AND CRTM FOR AN AUTOMOTIVE ROOF. [cited 2017 Nov 14]; Available from: http://www.escm.eu.org/eccm16/assets/0643.pdf

26. Dong C. Injection Gate Definition for Improving the Accuracy of Liquid Composite Molding Process Simulation. J Compos Mater. 2007;

27. Okabe T, Oya Y, Yamamoto G, Sato J, Matsumiya T, Matsuzaki R, et al. Multi-objective optimization for resin transfer molding process. Compos Part A Appl Sci Manuf [Internet]. 2017 Jan 1 [cited 2019 Oct 17];92:1–9. Available from: https://www.sciencedirect.com/science/article/pii/S1359835X1630327X

28. Gokce A, Advani SG. Vent Location Optimization Using Map-Based Exhaustive Search in Liquid Composite Molding Processes. Mater Manuf Process [Internet]. 2004 Dec 28 [cited 2019 Aug 20];19(3):523–48. Available from: http://www.tandfonline.com/doi/abs/10.1081/AMP-120038659

29. Grössing H, Stadlmajer N, Fauster E, Fleischmann M, Schledjewski R. Flow front advancement during composite processing: predictions from numerical filling simulation tools in comparison with real-world experiments. Polym Compos [Internet]. 2016 Sep 1 [cited 2018 Jan 4];37(9):2782–93. Available from: http://doi.wiley.com/10.1002/pc.23474

30. Marquette P, Dereims A, Pickett A. Simulation Based Solution for Industrial Manufacture of Large Infusion Composite Parts. In 2014.

31. Kärger L, Bernath A, Fritz F, Galkin S, Magagnato D, Oeckerath A, et al. Development and validation of a CAE chain for unidirectional fibre reinforced composite components. Compos Struct [Internet]. 2015 Nov 15 [cited 2019 Jan 31];132:350–8. Available from: https://www.sciencedirect.com/science/article/pii/S0263822315004213

32. Group E. PAM-RTM for CATIA V5 demo: Injection of wind blade [Internet]. www.youtube.com; 2009. Available from: https://www.youtube.com/watch?v=amE2c92f_Fk&ab_channel=ESIGroup

33. Liquid Injection Molding Simulation (LIMS) [Internet]. University of Delaware. Available from: https://sites.udel.edu/lims/

34. Loos AC, Springer GS. Curing of Epoxy Matrix Composite. J Compos Mater [Internet]. 1983;17:135–69. Available from: https://journals.sagepub.com/doi/pdf/10.1177/002199838301700204

35. Zhang W, Xu Y, Hui X, Zhang W. A multi-dwell temperature profile design for the cure of thick CFRP composite laminates. Int J Adv Manuf Technol [Internet]. 2021;117(3–4):1133–46. Available from: 10.1007/s00170-021-07765-1

36. Fernlund G, Rahman N, Courdji R, Bresslauer M, Poursartip A, Willden K, et al. Experimental and numerical study of the effect of cure cycle, tool surface, geometry, and lay-up on the dimensional fidelity of autoclave-processed composite parts. Compos Part A Appl Sci Manuf. 2002;33(3):341–51.

37. Pantelelis N, Vrouvakis T, Spentzas K. Cure cycle design for composite materials using computer simulation and optimisation tools. Forsch im Ingenieurwes. 2003;67.

38. Carlone PP., Palazzo GS. A simulation Based Metaheuristic Optimization of the Thermal Cure Cycle of Carbon-Epoxy Composite Laminate. AIP Conf Proc. 2011;5.

39. COMPRO Simulation Software [Internet]. Convergent Simulation Technologies. Available from: https://www.convergent.ca/products/compro-simulation-software

40. Mesogitis T, Kratz J, Skordos AA. SimpleCure [Internet]. Advanced Composites Collaboration for Innovation & Science. Available from: accis.github.io/archive/simplecure

41. Mesogitis T, Kratz J, Skordos AA. Heat transfer simulation of the cure of thermoplastic particle interleaf carbon fibre epoxy prepregs. J Compos Mater [Internet]. 2018;53(15). Available from: 10.1007/s00170-021-07765-1

42. Van Ravenhorst JH. DESIGN TOOLS FOR CIRCULAR OVERBRAIDING OF COMPLEX MANDRELS [Internet]. [cited 2019 Feb 21]. Available from: https://ris.utwente.nl/ws/portalfiles/portal/46645249/PhD_thesis_Johan_van_Ravenhorst.pdf

43. Ravenhorst, J.H., Akkerman R. Circular Braiding Process Simulation For a Pressure Vessel. ASME, Press Vessel Pip Conf. 2014;

44. Akkerman R, Hadir B, Rodríguez V. Braiding Simulation and Slip Evaluation for Arbitrary

Mandrels [Internet]. 2007 [cited 2019 Mar 13]. Available from: https://ris.utwente.nl/ws/portalfiles/portal/5404032/Akkerman-Esaform2007-Braiding_Simulation_and_slip_evaluation.pdf

45. Razavi S, Iannucci L. 5 th International LS-DYNA ® Users Conference Simulation of the Braiding Process in LS-DYNA ® [Internet]. [cited 2019 Feb 21]. Available from: https://www.dynalook.com/conferences/15th-international-ls-dyna-conference/composites/simulation-of-the-braiding-process-in-ls-dyna-r

46. Hans T, Hinterhoelzl R. A NUMERICAL APPROACH MODELING THE BRAIDING PROCESS FOR ARBITRARY MANDREL SHAPES TO CALCULATE PREFORM PROPERTIES [Internet]. [cited 2019 Jul 18]. Available from: http://www.lcc.mw.tum.de

47. Melenka G. BraidedCompositeDesign [Internet]. GitHub repository; 2020. Available from: https://github.com/gmelenka/BraidedCompositeDesign

48. Du GW, Popper P. Analysis of a Circular Braiding Process for Complex Shapes. J Text Inst [Internet]. 1994 [cited 2020 Jan 31];316–37. Available from: https://www.tandfonline.com/doi/pdf/10.1080/00405009408631277?needAccess=true&instName=University+of+Bristol

49. Cheung BKO, Carey JP. Micromechanics for braided composites. In: Handbook of Advances in Braided Composite Materials: theory, production, testing and application [Internet]. Woodhead Publishing; 2017 [cited 2019 Oct 31]. p. 239–57. Available from: https://www.sciencedirect.com/science/article/pii/B9780081003695000064?via%3Dihub

50. Xu L. xComposites - MultiScale Analysis tool [Internet]. Available from: https://www.xcomposites.com/

51. Pickett AK, Sirtautas J, Erber A. Braiding Simulation and Prediction of Mechanical Properties. Appl Compos Mater. 2009;16:345–64.

52. Wang C. Strength prediction for bi-axial braided composites by a multi-scale modelling approach. J Mater Sci [Internet]. 2016 [cited 2019 Apr 30];51(12):6002–18. Available from: http://dx.doi.org/10.1007/s10853-016-9901-z.

53. Lomov S., Verpoest I, Cichosz J, Hahn C, Ivanov DS, Verleye B. Open Data Formats and Scripting in Integrated Meso-Level Textile Composite Simulations. 19 TH Int Conf Compos Mater. 2014;

54. Xu L, Jin CZ, Ha SK. Ultimate strength prediction of braided textile composite using a multi-scale approach. J Compos Mater [Internet]. 2014 [cited 2020 Jan 28]; Available from:

https://journals.sagepub.com/doi/pdf/10.1177/0021998314521062

55. Dong J, Huo N. A two-scale method for predicting the mechanical properties of 3D braided composites with internal defects. Compos Struct. 2016;152:1–10.

56. Sakakibara T, Yamamoto T, Yamada G, Imaoku A, Ohtagaki R, Zako M. Recent developments in "COMPOSITES DREAM" for meso-FE modelling of advanced materials. In: 13th International Conference on Textile Composites [Internet]. 2018 [cited 2020 Jan 30]. Available from: https://iopscience.iop.org/article/10.1088/1757-899X/406/1/012027/pdf

57. Kim KJ, Yu WR, Lee JS. Multi-scale modelling of 3D multi-layerd braided composite tubes. Soul Natl Univerisity-Department Mater Sci Eng Yeungmnam Univ Text [Internet]. 2008 [cited 2019 Jan 29]; Available from: https://link-springer-com.bris.idm.oclc.org/content/pdf/10.1007%2Fs12289-008-0237-3.pdf

58. Brown LP, Endruweit A, Long A, Jones IA. Characterisation and modelling of complex textile geometries using TexGen. 13th Int Conf Text Compos [Internet]. 2018 [cited 2019 Mar 20]; Available from: https://iopscience-iop-org.bris.idm.oclc.org/article/10.1088/1757-899X/406/1/012024/pdf

59. Lomov S., Verpoest I, Cichosz J, Hahn C, Ivanov DS, Verleye B. Open Data Formats and Scripting in Integrated Meso-Level Textile Composite Simulations. 19 TH Int Conf Compos Mater [Internet]. 2014 [cited 2019 May 1]; Available from: https://www.mtm.kuleuven.be/Onderzoek/Composites/Research/meso-macro/textile_composites_map/textile_modelling/downloads/textile-modelling-paper-2014.pdf

60. Verpoest I, Lomov S V. Virtual textile composites software WiseTex: Integration with micro-mechanical, permeability and structural analysis. Compos Sci Technol [Internet]. 2005 [cited 2020 Jan 28]; Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.554.5493&rep=rep1&type=pdf

61. Yamamoto T, Imaoku A, Sakakibara T, Zako M. A proposal of conventional FE-modeling for layered braided composites: comparison of numerical results with experimental results. IOP Conf Ser Mater Sci Eng [Internet]. 2018 Sep 21 [cited 2019 Oct 31];406(1):012031. Available from: http://stacks.iop.org/1757-899X/406/i=1/a=012031?key=crossref.d84a3be4c9c1afefcff44d31c9f44000

62. Gideon RK, Zhou H, Wu X, Sun B, Gu B. Finite element analysis of 3D circular braided composites tube damage based on three unit cell models under axial compression loading. Int J Damage Mech [Internet]. 2016 May 9 [cited 2019 Nov 20];25(4):574–607. Available from:

http://journals.sagepub.com/doi/10.1177/1056789515605568

63. Pitz E, Miron M-C, Kállai I, Major Z. NUMERICAL PREDICTIONS AND MECHANICAL TESTING OF BRAIDED COMPOSITE STRUCTURES UTILISING DIGITAL IMAGE CORRELATION. Acta Polytech CTU Proc [Internet]. 2016 Dec 9 [cited 2019 Apr 30];7:43. Available from: https://ojs.cvut.cz/ojs/index.php/APP/article/view/3963

64. Hong Y, Yan Y, Guo F, Li X, Tian Z. Predicting the Elastic Properties of 3D N-Directional Braided Composites Via a Theoretical Method. Mech Compos Mater [Internet]. 2019 Mar 15 [cited 2019 Oct 30];55(1):95–106. Available from: http://link.springer.com/10.1007/s11029-019-09795-z

65. Robitaille F, Clayton BR, Long AC, Rudd CD. Geometric modelling of industrial preforms: woven and braided textiles. 1999.

66. Srikanth L, Rao RMVGK. Strength and stiffness behaviour of braided and filament wound glass epoxy composites - Simultaneous studies and comparison. J Compos Mater. 2014 Feb;48(4):407–14.

67. Byun J-H, Whitney T., Du GW, Chou TW. Analytical Characterization of Two-Step Braided Composites [Internet]. 1990 [cited 2019 May 1]. Available from: https://journals.sagepub.com/doi/pdf/10.1177/002199839102501203

68. Shokrieh MM, Mazloomi MS. An analytical method for calculating stiffness of two-dimensional tri-axial braided composites. Compos Struct. 2010 Nov;92(12):2901–5.

69. Shahabi NE, Saharkhiz S, Mohammad S, Varkiyani H. Effect of Fabric Structure and Weft Density on the Poisson’s Ratio of Worsted Fabric [Internet]. Vol. 63, Journal of Engineered Fibers and Fabrics. [cited 2019 Jun 6]. Available from: http://www.jeffjournal.org

70. Song YS, Chung K, Kang TJ, Youn JR. Prediction of permeability tensor for three dimensional circular braided preform by applying a finite volume method to a unit cell. Compos Sci Technol [Internet]. 2004 Aug 1 [cited 2019 Oct 30];64(10–11):1629–36. Available from: https://www.sciencedirect.com/science/article/pii/S0266353803004378

71. Wong CC, Long AC, Sherburn M, Robitaille F, Harrison P, Rudd CD. Comparison of novel and efficient approaches for permeability prediction based on the fabric architecture. Compos Part A Appl Sci Eng. 2006;37(6):847–57.

72. Swery EE, Meier R, Lomov S V, Drechsler K, Kelly P. Predicting permeability based on flow simulations and textile modelling techniques: Comparison with experimental values and verification of FlowTex solver using Ansys CFX. J Compos Mater [Internet]. 2016 Mar 9

[cited 2020 Jan 24];50(5):601–15. Available from: http://journals.sagepub.com/doi/10.1177/0021998315579927

73. Swery EE, Hans T, Bultez M, Wijaya W, Kelly P, Hinterhölzl R, et al. Complete simulation process chain for the manufacturing of braided composite parts. Compos Part A Appl Sci Manuf [Internet]. 2017 Nov [cited 2018 Jul 30];102:378–90. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1359835X17303111

74. Swery EE, Hans T, Bultez M, Wijaya W, Kelly P, Hinterhölzl R, et al. Complete simulation process chain for the manufacturing of braided composite parts. Compos Part A Appl Sci Manuf [Internet]. 2017 Nov 1 [cited 2019 Jun 25];102:378–90. Available from: https://www.sciencedirect.com/science/article/pii/S1359835X17303111

75. Lian B, Jiang L, McGrath JJ, Jaranson J. Quantitative determination of morphological features of triaxially braided composites by the use of machine vision. Compos Sci Technol [Internet]. 2000 Feb 1 [cited 2019 Nov 5];60(2):159–66. Available from: https://www.sciencedirect.com/science/article/pii/S0266353899001153

76. Hunt AJ, Carey JP. A machine vision system for the braid angle measurement of tubular braided structures. Text Res J [Internet]. 2019 Jul 24 [cited 2019 Oct 30];89(14):2919–37. Available from: http://journals.sagepub.com/doi/10.1177/0040517518803792

77. Schillfahrt C, Fauster E, Schledjewski R. An experimental method for characterizing unsaturated 1D flow in tubular braided preforms during bladder-assisted resin transfer molding. Polym Compos [Internet]. 2018 Dec 16 [cited 2019 Oct 25];39(12):4666–77. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/pc.24585

78. Endruweit A, McGregor P, Long AC, Johnson MS. Influence of the fabric architecture on the variations in experimentally determined in-plane permeability values. Compos Sci Technol. 2006;66(11–12):1778–92.

79. Yu B, Lee J. A Simplified In-Plane Permeability Model for Textile Fabrics [Internet]. Available from: https://onlinelibrary.wiley.com/doi/epdf/10.1002/pc.10221?saml_referrer

80. Endruweit A, Long AC. A model for the in-plane permeability of triaxially braided reinforcements. Compos Part A Appl Sci Manuf [Internet]. 2011 Feb 1 [cited 2019 Oct 25];42(2):165–72. Available from: https://www.sciencedirect.com/science/article/pii/S1359835X10002897

81. Adams D. Composites testing as part of a building block approach, Part 1: Coupon-level testing. CompositesWorld [Internet]. 2021; Available from: https://www.compositesworld.com/articles/composites-testing-as-part-of-a-building-block-

approach-part-1-coupon-level-testing

82. Adams D. Composites testing as part of a building block approach, Part 2: Upper building block levels. CompositesWorld [Internet]. 2021; Available from: https://www.compositesworld.com/articles/composites-testing-as-part-of-a-building-block-approach-part-2-upper-building-block-levels

83. DoD. Composite Materials Handbook, Volume 1. Polymer Matrix Composites Guidelines for Characterization of Structural Materials. Department of Defense, USA; 2002.

84. Johnson AF, Thomson RS, David M, Joosten MW. 10 - Design and testing of crashworthy aerospace composite components. Polym Compost Aerosp Ind. 2015;261–93.

85. CMH-17 Composite Materials Handbook [Internet]. Wichita State University. 2013. Available from: https://www.cmh17.org/

86. SAE International, Wichita State University, Research NI for A. Composite materials handbook. SAE International on behalf of CMH-17, a division of Wichita State University; 2012.

87. Thomsen O, Butler R, Hallett S, Barton J. Certification for Design: Reshaping the testing pyramid [Internet]. EPSRC. Available from: https://www.composites-certest.com/

88. NIAR. National Center for Advanced Materials Performance [Internet]. Wichita State University. 2022. Available from: https://www.wichita.edu/industry_and_defense/NIAR/Research/ncamp.php

89. Elen M, Aishwarya N, Vadlamudi V, Raihan R. Detection and prediction of defects in composite materials using di-electric characterization and neural networks. In: Institute for Predictive Performance and Methodologies [Internet]. The University of Texas at Arlington Research Institute; Available from: https://www.researchgate.net/publication/326040660_DETECTION_AND_PREDICTION_OF_DEFECTS_IN_COMPOSITE_MATERIALS_USING_DI-ELECTRIC_CHARACTERIZATION_AND_NEURAL_NETWORKS

90. Schillemans L, Hemlrijck d. Van, Roey, F D, Wilde WPD, Cardon AH, Anastassopoulos AA. Defect-identification in composite materials using pattern recognition techniques on ultrasonic data. Adv Compos Mater. 1993;

91. Orazio TD, Leo M, Guaragnella C, Distante A. Analysis of Image Sequences for Defect Detection in Composite Materials. Int Conf Adv Concepts Intell Vis Syst [Internet]. 2007;855–64. Available from: https://link.springer.com/chapter/10.1007/978-3-540-74607-2_78

92. Elsaid B, Hallett S. Multiscale surrogate modelling of the elastic response of thick composite structures with embedded defects and features. Compos Struct. 2018;200:781–98.

93. Ali HT, Jones MI, Kawashita LF, Hallett SR. Experimental study of laminated composites containing manufacturing defects under combined stress states. In: Technical Conference of the American Society for Composites [Internet]. Advanced Composites Centre for Innovation and Science (ACCIS); 2016. Available from: https://www.scopus.com/record/display.uri?eid=2-s2.0-85013985497&origin=resultslist&sort=cp-f&src=s&st1=material+AND+properties+AND+composites+AND+defect+AND+knock-down+wrinkle&sid=2f6d391474225bca17842466ec5492e9&sot=b&sdt=b&sl=87&s=TITLE-ABS-KEY%28mater

94. Potter K, Khan B, Wisnom M, Bell T, Stevens J. Variability, fibre waviness and misalignment in the determination of the properties of composite materials and structures. Compos Part A Appl Sci Manuf [Internet]. 2008; Available from: https://www.scopus.com/record/display.uri?eid=2-s2.0-50049125317&origin=reflist&sort=cp-f&src=s&st1=material+AND+properties+AND+composites+AND+defect+AND+knock-down+AND+waviness&sid=f5864a969d0646049f251d3d4a037c91&sot=b&sdt=b&sl=92&s=TITLE-ABS-KEY%28mate

95. Kraft EM. The Air Force Digital Thread/Digital Twin - Life Cycle Integration and Use of Computational and Experimental Knowledge. In: 54th AIAA Aerospace Sciences Meeting [Internet]. Reston, Virginia: American Institute of Aeronautics and Astronautics; 2016 [cited 2018 Apr 18]. Available from: http://arc.aiaa.org/doi/10.2514/6.2016-0897

96. Deloitte. Industry 4.0 and the digital twin Manufacturing meets its match. [cited 2018 Apr 17]; Available from: https://www2.deloitte.com/content/dam/Deloitte/cn/Documents/cip/deloitte-cn-cip-industry-4-0-digital-twin-technology-en-171215.pdf

97. Nikishkov Y, Seon G, Makeev A. Structural analysis of composites with porosity defects based on X-ray computed tomography. J Compos Mater [Internet]. 2014 Jul 15 [cited 2018 Apr 19];48(17):2131–44. Available from: http://journals.sagepub.com/doi/10.1177/0021998313494917

98. Thomas H.-J. Uhlemann, Christoph Schock, Christian Lehmann, Stefan Freinberger, Rolf Steinhilper. The Digital Twin: Demonstrating the potential of real time data acquisition in production systems. Procedia Manuf [Internet]. 2017 [cited 2018 Apr 23];9:113–20. Available from: https://ac.els-cdn.com/S2351978917301610/1-s2.0-S2351978917301610-main.pdf?_tid=292c49df-ed65-468d-8a5a-

3814827fecbb&acdnat=1524472724_232412058911dbb96c54dbb9c27de4b8

99. Rodič B. Industry 4.0 and the New Simulation Modelling Paradigm. De Gruyter [Internet]. 2017 [cited 2018 Apr 23];50(3). Available from: https://www.degruyter.com/downloadpdf/j/orga.2017.50.issue-3/orga-2017-0017/orga-2017-0017.pdf

100. Siedlak DJL, Pinon OJ, Schlais PR, Schmidt TM, Mavris DN. A digital thread approach to support manufacturing-influenced conceptual aircraft design. Res Eng Des [Internet]. 2018 Apr 13 [cited 2018 Jun 6];29(2):285–308. Available from: http://link.springer.com/10.1007/s00163-017-0269-0

101. Geissbauer R, Schrauf S, Berttram P, Cheraghi F. Digital Factories 2020: Shaping the future of manufacturing. PWC. 2017;

102. TwinControl. Project overview | Twin-Control [Internet]. [cited 2018 Apr 24]. Available from: http://twincontrol.eu/project-overview/

103. MTC. From Industry 4.0 to Digitising Manufacturing An End User Perspective Conference Report [Internet]. [cited 2018 Jul 31]. Available from: http://www.the-mtc.org/pdf/Industry-4-Report-2016-e.pdf

104. Dekhtiar J, Durupt A, Bricogne M, Eynard B, Rowson H, Kiritsis D. Deep learning for big data applications in CAD and PLM – Research review, opportunities and case study. Comput Ind [Internet]. 2018 Sep 1 [cited 2018 Jun 6];100:227–43. Available from: https://www-sciencedirect-com.bris.idm.oclc.org/science/article/pii/S0166361517305560

105. Deborah Sherry. If We Want to Build the Factory of the Future, We Need to Go Beyond Industry 4.0 | GE Digital [Internet]. www.ge.com. [cited 2018 Apr 23]. Available from: https://www.ge.com/digital/blog/if-we-want-build-factory-future-we-need-go-beyond-industry-40

106. Siemense. Digital Enterprise - on the way to Industrie 4.0 - Digital Twin. Advance [Internet]. 2015 [cited 2018 Apr 23];(2). Available from: https://www.siemens.com/content/dam/internet/siemens-com/customer-magazine/old-mam-assets/print-archiv/advance/adv152-en-screen.pdf

107. Deutsche Kommission Elektrotechnik D. DIN/DKE - Roadmap, German Standardization Roadmap, Industrie 4.0. 2018 [cited 2018 Jun 27]; Available from: https://www.din.de/blob/65354/57218767bd6da1927b181b9f2a0d5b39/roadmap-i4-0-e-data.pdf

108. McEwan W, Butterfield J. A Digital Methodology for the Design Process of Aerospace Assemblies with Sustainable Composite Processes & Manufacture. In: AIP Conference Proceedings [Internet]. American Institute of Physics; 2011 [cited 2018 Aug 17]. p. 1683–8. Available from: http://aip.scitation.org/doi/abs/10.1063/1.3589758

109. Calado EA, Leite M, Silva A. Selecting composite materials considering cost and environmental impact in the early phases of aircraft structure design. J Clean Prod [Internet]. 2018 Jun 10 [cited 2018 Aug 30];186:113–22. Available from: https://www.sciencedirect.com/science/article/pii/S0959652618303548

110. Verleye B, Lomov SV, Long A, Verpoest I, Roose D. Permeability prediction for the meso–macro coupling in the simulation of the impregnation stage of Resin Transfer Moulding. Compos Part A Appl Sci Manuf [Internet]. 2010 Jan 1 [cited 2018 Aug 6];41(1):29–35. Available from: https://www.sciencedirect.com/science/article/pii/S1359835X09001869

111. Pierce RS, Falzon BG, Thompson MC. A multi-physics process model for simulating the manufacture of resin-infused composite aerostructures. Compos Sci Technol [Internet]. 2017 Sep [cited 2018 Aug 6];149:269–79. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0266353817307911

112. Kärger L, Kling A. As-built FE simulation of advanced fibre placement structures based on manufacturing data. Compos Struct [Internet]. 2013 Jun 1 [cited 2018 Jan 4];100:104–12. Available from: http://www.sciencedirect.com/science/article/pii/S0263822312006009

113. Seeger J, Wolf K. Multi-objective design of complex aircraft structures using evolutionary algorithms. Proc Inst Mech Eng Part G J Aerosp Eng [Internet]. 2011 Oct 15 [cited 2018 Aug 20];225(10):1153–64. Available from: http://journals.sagepub.com/doi/10.1177/0954410011411384

114. Fengler B, Hrymak A, Karger L. Multi-objective CoFRP patch optimization with consideration of manufacturing constraints and integrated warpage simulation. Compos Struct [Internet]. 2019;221. Available from: https://doi.org/10.1016/j.compstruct.2019.04.033

115. Böhler P, Dittmann J, Michaelis D, Middendorf P, Liebold C. MANUFACTURING SIMULATION AS PART OF THE DIGITAL PROTOTYPE [Internet]. 2017 [cited 2018 Aug 21]. Available from: https://www.dynalook.com/11th-european-ls-dyna-conference/materials-arena-2036/manufacturing-simulation-as-part-of-the-digital-prototype

116. Vinot M, Holzapfel M, Jemmali R. Numerical Investigation of Carbon Braided Composites at the Mesoscale: using Computer Tomography as a Validation Tool [Internet]. 2015 [cited 2018 Aug 21]. Available from: https://www.dynalook.com/10th-european-ls-dyna-conference/4

Materials II - Endless Fibers/03-Vinot-GermanAerospaceCenter-P.pdf

117. Böhler P, Dittmann J, Michaelis D, Middendorf P. Process Simulation as Part of Industry 4.0. Light Des [Internet]. 2016 Jun 14 [cited 2018 Aug 22];9(S3):6–11. Available from: http://link.springer.com/10.1007/s35725-016-0070-4

118. Ciampa PD, Prakasha PS. Agile 4.0 [Internet]. Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR). Available from: https://www.agile4.eu/

119. Liu X, Furrer D, Kosters J, Holmes J. Vision 2040: A Roadmap for Integrated, Multiscale Modeling and Simulation of Materials and Systems [Internet]. 2018 [cited 2019 Jul 1]. Available from: https://ntrs.nasa.gov/search.jsp?R=20180002010

120. Tierney JJ, Gillespie JW. Composite Design and Simulation (CDS) software: A comprehensive toolkit for real-time design of composite processes and structures. In: International SAMPE Symposium and Exhibition (Proceedings). 2009.

121. Delaware U of. CDS Composite Design and Simulation Software [Internet]. 2011. Available from: http://nwp.engr.udel.edu/ccm/CDS/

122. Dassault Systemes. The 3DExperience platform, a game changer for business and innovation [Internet]. [cited 2020 Sep 28]. Available from: https://www.3ds.com/3dexperience/

123. Indermuehle K, Prior A. Design and Analysis Workflows for Advanced Simulation of Composite Structures. Dassault Syst Simulia [Internet]. [cited 2020 Jan 20]; Available from: http://iccm-central.org/Proceedings/ICCM17proceedings/Themes/Behaviour/STRUCTURAL ANALY & OPTIM/INT - STRUCT ANALYSIS & OPTIM/IF22.5 Prior.pdf

124. ESI. ESI Launches PAM-Composites 2016. 2016; Available from: https://www.esi-group.com/company/news/esi-launches-pam-composites-2016

125. Siemens. MindSphere: The cloud-based, open IoT operating system for digital tranformation. 2017; Available from: https://iiot-world.com/wp-content/uploads/2017/03/Siemens_MindSphere_Whitepaper.pdf

126. MSC_Software. Hexagon/MSC Software - Composites [Internet]. Available from: https://www.mscsoftware.com/application/composites

127. Schuhmacher G, Daoud F, Petersson Ö, Wagner M. MULTIDISCIPLINARY AIRFRAME DESIGN OPTIMISATION [Internet]. [cited 2018 Nov 1]. Available from: http://www.icas.org/ICAS_ARCHIVE/ICAS2012/PAPERS/994.PDF

128. Forrester A, Sobester A, Keane A. Engineering Design via Surrogate Modelling. John Wiley & Sons Ltd.; 2008.

129. Tian K, Ma X, Li Z, Lin S, Wang B, Waas AM. A multi-fidelity competitive sampling method for surrogte-based stacking sequence optimization of composite shell with multiple cutouts. Int J Solids Struct. 2020;

130. Moza S. LHS-MDU [Internet]. GitHub repository; Available from: https://github.com/sahilm89/lhsmdu

131. Deutsch JL, Deutsch C V. Latin hypercube sampling with multidimensional uniformity. J Stat Plan Inference [Internet]. 2012 Mar 1 [cited 2019 Nov 19];142(3):763–72. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0378375811003776?via%3Dihub

132. Dababneh O, Kipouros T, Whidborne JF. Application of an Efficient Gradient-Based Optimization Strategy for Aircraft Wing Structures. Aerospace. 2018;5(1).

133. Tutum CC., Baran I, Deb K. Optimum design of pultrusion process via evolutionary multi-objective optimization. Int J Adv Manuf Technol. 2014;72(9–12):1205–17.

134. Kinjo H, Nakazono · K, Sam DC, Maeshiro · M, Yamamoto T, Kinjo H, et al. Solution searching for multivariable optimisation problem by a momentum genetic algorithm. 2008 [cited 2019 Apr 2]; Available from: https://link.springer.com/content/pdf/10.1007%2Fs10015-007-0467-3.pdf

135. Fontecha Dulcey G, Fischer X, Joyot P, Fadel G. Support for Decision Making in Design of Composite Laminated Structures. Part 2: Reduced Parametric Model-Based Optimization. Appl Compos Mater [Internet]. 2019 Apr 15 [cited 2019 Jul 31];26(2):663–81. Available from: http://link.springer.com/10.1007/s10443-018-9742-9

136. Fontecha Dulcey G. Parametric, reduced and multiscale model for the interactive optimization of laminated composite structures [Internet]. [cited 2019 Jul 31]. Available from: https://tel.archives-ouvertes.fr/tel-01985817

137. Mathur R, Advani SG, Fink BK. Optimization of gate and vent locations for resin infusion processes using genetic algorithms. In: Proceedings of the 1998 American Control Conference ACC (IEEE Cat No98CH36207) [Internet]. IEEE; 1998 [cited 2019 Jun 27]. p. 2176–80 vol.4. Available from: http://ieeexplore.ieee.org/document/703013/

138. Scipy. scipy.optimise.differential.evolution [Internet]. 2020. Available from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html

139. Li H, Wu T, Gao Z, Wang X, Ma H, Han Q, et al. An iterative method for identification of temperature and amplitude dependent material parameters of fiber-reinforced polymer composites. Int J Mech Sci. 2020;184T.

140. Dubreuil S, Bartoli N, Gogu C, Lefebvre T. Towards an efficient global multidisciplinary design optimization algorithm. Struct Multidiscip Optim [Internet]. 2020;62:1739–65. Available from: https://link.springer.com/content/pdf/10.1007/s00158-020-02514-6.pdf

141. Bouhel MA, Bartoli N, Otsmane A, Morlier J. Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction. In: Open Archive Toulouse Archive Ouverte [Internet]. 2017. Available from: https://hal.archives-ouvertes.fr/hal-01598259/document

142. Hurkamp A, Gellrich S, Ossowski T, Beuscher J, Thiede S, Herrmann C, et al. Combining simulation and machine learing as digital twin for manufacturing of overmolded thermoplastic composites. J Manuf Mater Process [Internet]. 2020;4(3). Available from: https://www.mdpi.com/2504-4494/4/3/92

143. Tullu A, Lee B, Hwang H. Surrogate Model Based Analysis of Inter-Ply Shear Stress in Fiber Reinforced Thermoplastic Composite Sheet Press Forming. Appl Sci. 2020;10.

144. scikit-learn: Machine Learning in Python [Internet]. Available from: https://scikit-learn.org/stable/

145. Scikit-learn. Choosing the right estimator [Internet]. Available from: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

146. Mader CA, Kenway GKW, Yildirim A, Martins JRRA. ADflow: An open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimisation. J Aerosp Inf Syst. 2020;17(9):508–27.

147. Budziak K. Aerodynamic Analysis with Athena Vortex Lattice (AVL) [Internet]. Hamburg University of Applied Sciences; 2015. Available from: https://core.ac.uk/download/pdf/288024095.pdf

148. XFLR5 Analysis of foils and wings operating at low Reynolds numbers [Internet]. 2019. Available from: https://engineering.purdue.edu/~aerodyn/AAE333/FALL10/HOMEWORKS/HW13/XFLR5_v6.01_Beta_Win32%282%29/Release/Guidelines.pdf

149. Sharp P. AeroSandbox [Internet]. Available from: https://peterdsharpe.github.io/AeroSandbox/%0A

150. Drela M. AVL [Internet]. MIT; Available from: http://web.mit.edu/drela/Public/web/avl/

151. Launchpad-McQ. Aileron-design [Internet]. GitHub repository; Available from: https://github.com/Launchpad-McQ/Aileron-design%0A

152. SUAVE 2.2: New Supersonic Vortex Lattice [Internet]. Stanford Aerospace Design Lab. 2020 [cited 2020 Sep 22]. Available from: https://suave.stanford.edu/

153. Lukaczyk T, Botero E, Mungia B. SUAVE: An Aerospace Vehicle Environment for Designing Future Aircraft [Internet]. GitHub repository; Available from: https://github.com/suavecode

154. Vogeltanz T. Comparison of open-source CFD software for aerodynamic analysis of mini-UAV. In: AIAA/IEEE Digital Avionics Systems Conference - Proceedings. Institute of Electrical and Electronics Engineers Inc.; 2015. p. 5E31–315.

155. Wendorf AD, Botero E, Alonso JJ. Comparing Different Off-the-Shelf Optimizers' Performance in Conceptual Aircraft Design. Am Inst Aeronaut Astronaut [Internet]. Available from: https://suave.stanford.edu/publications/opt_compare.pdf%0A

156. O-gent. PyXFLR [Internet]. GitHub repository; 2020. Available from: https://github.com/o-gent/PyXFLR

157. OpenVSP [Internet]. GitHub repository; Available from: https://github.com/OpenVSP/OpenVSP

158. Azabi Y, Savvaris A, Kipouros T. The Interactive Design Approach for Aerodynamic Shape Design Optimisation of the Aegis UAV. Sch Aerospace, Transp Manuf Cranf Univ. 2019;

159. Azabi Y, Savvaris A, Kipouros T. Initial Investigation of Aerodynamic Shape Design Optimisation for the Aegis UAV. In: 6th CEAS AIR & SPACE CONFERENCE AEROSAPCE EUROPE 2017. 2017.

160. Vegh MJ, Alonso JJ, Orra TH, Illario da Silva CR. Flight Path and Wing Optimization of Lithium-Air Battery Powered Passanger Aircraft. In: American Institute of Aeronautics and Astronautics [Internet]. Available from: http://adl.stanford.edu/papers/AIAA-2015-1674.pdf

161. Benaouali A, Kachel S. Multidisciplinary design optimisation of aircraft wing using commercial software integration. Aerosp Sci Technol. 2019;92:766–76.

3. Development of the System of Simulation demonstrator

3.1 Introduction

The literature review in Chapter 2 identified the key software/simulations required to design a composite part and manufacturing process. This chapter will show how the Python model automation, robust data management, and simulation connectivity lead to holistic optimisation and rapid design changes. To establish how well the envisioned benefits of the system of simulations fare against the reality of part design, a demonstrator of the system of simulations was developed as a case study.

Python scripting is used because Python allows for bespoke optimisation algorithms, data processing, and can be interfaced with the majority of commercial and open-source software. The python scripts are designed to re-create all models required for the complete design, with parameters which differ for each iteration. The set of iterable parameters will be specified at the start of the project so that each model can accommodate for the variability. Thus, the importance of clearly defining the specification at the beginning of design increases for this new process.

A flow map of variables should always be created to aid with the design of the individual models. This way work can be distributed, while overall, the chain will not be negatively impacted by the division of work. An example of a variable flow map, used in a case study, is shown below in Figure 1.2. This is a simplified schematic with unspecified variables, the full map is available on GitHub as part of the documentation(1).

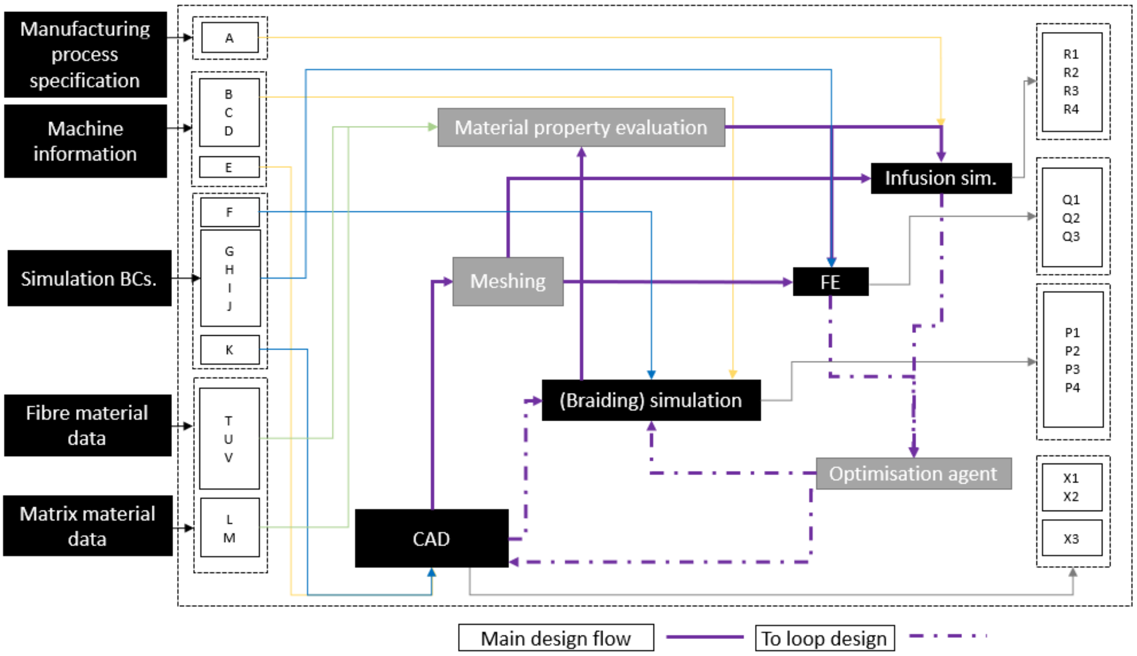


Figure 3.1 – a simplified variable flow map used for system of simulation development(1)

3.2 The demonstrator part – module development

On one hand, the demonstrator component must be simple enough so that all the required modules could be combined without an excessive amount of technical development with each simulation. On the other hand, the component needs to have basic complexity associated with composite components. For instance, the manufacturing aspects must affect the material properties. The usefulness of manufacturing simulation is also beneficial.

With these requirements in mind, a small unmanned aircraft spar was selected to be used as a demonstrator for the system of simulation, see Figure 3.2 for illustration. The blue segment of the wing corresponds to the braided spar. The complete part can be described by several airfoil point sets adjusted by standard aerospace parameters: dihedral, sweep, twist, taper etc… The radii of the spar corners can also be adjusted, along with the location along the chord and percentage of chord taken by the spar. The local braiding parameters in each segment can affect the structural properties. Therefore, to achieve high quality part these parameters can be optimised. It is also a part that would benefit from the new approach due to the high likelihood of design iterations. For instance, with the same design and only altered parameters multiple vastly different UAV designs could be produced. These could range from small toy like UAVS, through high endurance surveillance drones, to large transport drones. Following the braiding simulation, resin infusion simulation can also be considered.

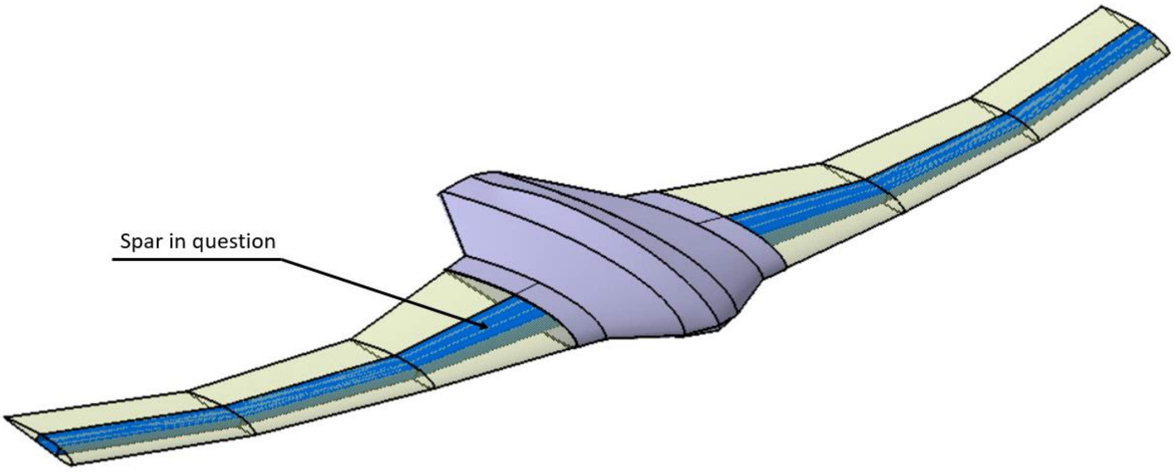


Figure 3.2 – generic UAV shape with a spar highlighted

Manufacturing methods other than braiding were considered. However, for a part of this shape the braiding is quite ideal. Braiding is also highly automated, hence repeatable, process (2). Therefore, simulations of this process will suffer less from randomness than other methods, such as hand-layup (3). Additional benefit of selecting braiding is the initiated research program at authors host organisation, which can lead to some cross-project collaboration.

The part uses a set of parameters that would be present in other aerospace parts. The main difference from other aerospace parts is the relative simplicity of construction. The part is to be made of a single type of fibres and one resin system. The shape of the part strongly suggests braiding manufacturing method followed by a resin infusion method.

The assembly considerations were ignored for the demonstrator, as that would require a specific UAV to be considered. However, these would not be difficult to include either as boundary conditions in the structural analysis or as parameters in the CAD module. Along with the shape parameters each module described in following sections introduces additional parameters for the part. These parameters either must be controlled or must become part of the holistic optimisation.

The methods described here are available on GitHub in their most current form(1).

3.3 CAD module

This section describes the development of the CATIA automation module. The CATIA software has been selected for reasons outlined in the literature review, namely robustness and scriptability.

The following sections describe the development of the scripts used, summarizes the learning, and outlines potential further uses of these tools, in that order.

3.3.1 development of scripts in CATIA

As outlined in the literature review, there are only few resources available for scripting CATIA. The main resource are the community discussion forums that can help with specific problems. Therefore, the scripting development initially consisted of recording macros within CATIA. The recorded macros help with understanding how the CATIA functions are structured. With basic understanding of VBA syntax, these recordings can be used as a template for new macros.

These VBA macros were then turned in Python scripts. Python can be used to control CATIA through the use of “win32com” library. There are only minor differences between the VBA macro and the Python scripts in terms of syntax. Mainly, care needs to be taken with indents and case sensitivity as neither of these are used in VBA.

These Python scripts can then be adjusted to create loops, gated function and more, to programmatically adjust the part created.

The demonstrator part is generated using cross-sections. Each cross-section is created by point cloud for the top and bottom section based on local air foil definition. Splines are created in those regions. These are then connected by vertical lines. Radius is applied where the straight lines come in contact with the aerofoil. To finalize the cross section all the components are joined using “join” function.

Lastly, a loft function is used to create the part surface. Figure 3.3 shows an example part in CATIA with the standard tree structure.

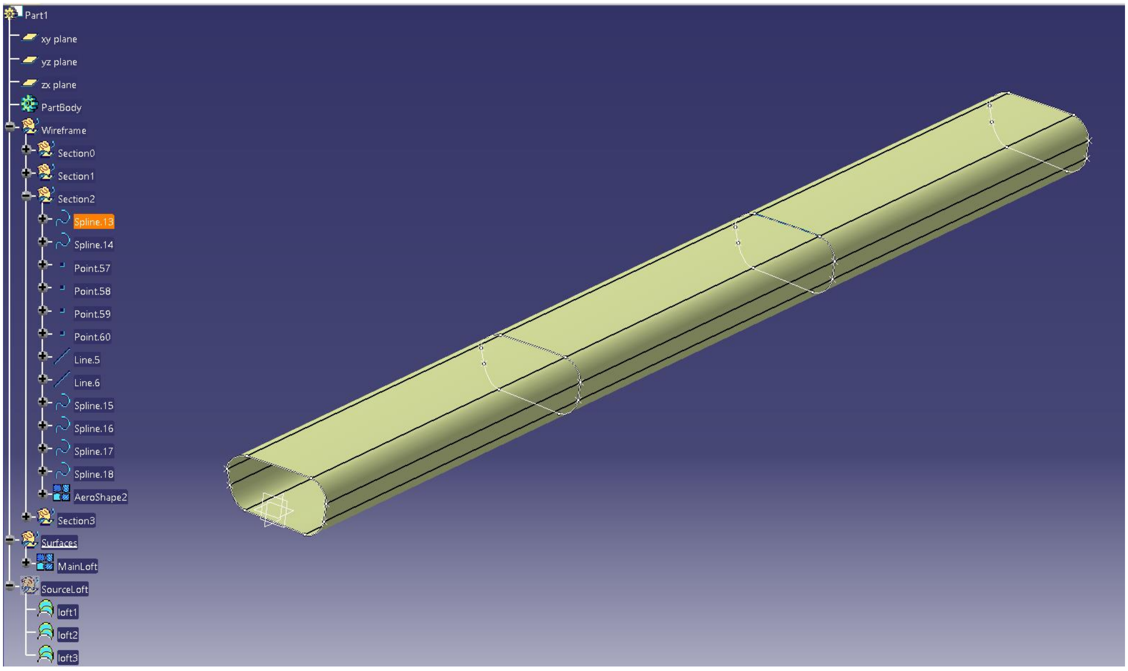


Figure 3.3 – example part after the initial shape defining script

Figure 3.3 is a relatively simple part. The whole script that creates the part in an empty CATIA runs for about 2 seconds. However, with build-up of geometry the efficiency of scripted CATIA drops exponentially, as will be further outlined in the braiding and meshing sections.

The Figure 3.4 shows some parameter variation possibilities. These are few shapes that have been generated automatically, any reasonable measure of twist, sweep and taper can be implemented. The span and cord-wise length of spar can also be adjusted by parametric variations.

Figure 3.4 – exemplar shapes created by the script

3.3.3 Learning from scripting

Generation of automated scripts in CATIA is not imposing too much additional burden on the designer. CATIA allows for recording of scripts and amendments that allow for automated replication of work. With few loops and relatively simple syntax the design of created component can be automatically recreated with amended parameters. However, some guidelines need to be followed for this to be a seamless process. If these are not followed, the automation becomes more complex than the design itself. Some of these guidelines are required for the automation, but most of them are general good practices that are beneficial even in standard design work.

First suggestion is to repeat same function as much as possible. For example, it is better to create all source points for multiple splines, rather than creating points then spline and repeating this. In standard design it does not matter, as it is a personal convenience decision: make 2 splines 1 surface and then repeat the process or creates all splines first for all surfaces. However, in automation it is easier to loop through the same function first, then move on. It makes the script more divisible and easier to read. It will also make the script marginally faster due to minimized re-referencing, as the reference geometries (parent geometries) are likely to overlap for any group of similar geometries.

Naming geometries is quite important. Reopening CATIA will refresh the default numbering system. The scripts require the geometry numbering for referencing. Therefore, it is important to care when user opens and closes CATIA. Alternatively, one can get into habit of including naming standards in the scripts.

Grouping geometries in systematic manner is important for any design work. However, in scripted design it has few more benefits. An example of this will be discussed in braiding simulation section where extracting normal depends heavily on good management of visible geometries. Hiding geometries is generally good idea, as it speeds up loading and updating process. This can be simplified by moving geometries into dedicated hidden geometry sets.

“CATIA.RefreshDisplay = False” can prevent some problems with running the CATIA in the background. It is good practice to always include “CATIA.RefreshDisplay = True” at the end of the same script to prevent visualisation issues.

Using hybrid design is not recommended by the author. This is to prevent mixing of surface modelling with part modelling, which make the recorded macros more difficult to understand.

Similarly to naming of geometries, naming of saved files also needs a system. Opening and closing of files through scripting relies on predictable file names for any geometry or part crated. Of course, this also helps with standard design methods as the time to identify a specific file on a shared drive is greatly reduced. It also allows other users to understand the purpose of a file without having to open it.

When recording macros CATIA choses directions randomly where two are available, for instance along a spline. When a similar spline is created with the script the direction usually remains the same. However, with complex geometries author found the direction can be reversed, causing major errors. The direction used in many CATIA functions can be created manually. This is not possible to do within the user interface, and hence it cannot be recorded as VBA code. The function is “dir1 = HSF1.AddNewDirectionByCoord(0, 1, 0)”, where the 3 numbers in the bracket denote the x, y, z vector. In this case line is used to create direction in terms of vector; this direction will not show in the interface. The direction can then be used as a reference input into any of the geometric functions. This was found to be one of key functions that enable robust automated process. This is an example of a difficulty, stemming from lack of specialized guides and other teaching materials.

Author would recommend the approach of using Python to maintain the scripts and run them in sequence. Although translation from VBA is required, the file management and connectivity are much easier. For general scripting this suggestion is optional, for interconnected systems of simulation this is a requirement.

3.3.4 Downsides and Limitations of CATIA scripting

Some functions within CATIA are not scriptable. One example is the measuring of distances. There are two issues with this; the output formats don’t really support easy extractions of obtained variables, and the VBA recording function does not record this particular activity. There is no documentation that would suggest this would be somehow possible without the recording functionality. There are workarounds where the above mentioned vector extraction can be used to create new geometry, and calculate distances outside CATIA.

The material database has similar issues. The recording ignores material library operations. Also, adjusting the material database through outside scripts seems difficult. Other users have tried this, and the community generally agrees this is not possible. Some documentation suggests importing pre-defined materials should be possible, but author was unable to replicate the process and confirm this. (2–4) Therefore material weights have to be calculated elsewhere if required.

Other tools might also not be scriptable. The author suspects that anything not directly related to geometry creation will be rather difficult to automate.

Another issue with the CATIA scripting is the scalability. Even with updating turned off, CATIA seems to be significantly affected by the number of geometric features already existent in a part or in an assembly. This issue will be further discussed in meshing section of this thesis.

The two most common laminate tools used withing CATIA are the internal Dassault option (CBA) and Fibresim. Neither of these is scriptable. There is no documentation that would suggest this would be possible and macro recording ignores anything done in these modules.

The assembly also seems to be limited in terms of scripting. For instance, author did not succeed at using the “isolate command” programmatically. Therefore, the build-up of geometries makes it impossible to develop large assembly models. However, author has not fully explored the options for automating assemblies in CATIA.

3.3.5 Other uses

These are few useful things that might be possible with CATIA automation that are beyond the scope of this work but could be quite useful if explored.

Assembling in CATIA can normally be quite a tedious process, as all parts need to be well oriented and assembly method chosen. However, to an experienced CATIA user it is obvious throughout the process which assembly method will be used. Therefore, instead of opening the assembly, he can move geometry relevant for the assembly to a specified geometrical set. The automated assembly could then pull parts based the naming of files and geometrical sets. This is highly dependent on robust referencing system.

The automation can also be used for easy manipulation of parameters already used within CATIA, this would result in a hybrid between parametric model and automated model.

The main standalone benefit of CATIA automated shape generation is visualisation of the part and of the assembly situations. Typically to assess assembly, surface contacts etc., new shape is created manually, or in best case scenario with the aid of parametrically setup model. With the automated tool multiple assessments of assembly options can be done quickly and painlessly. The same goes for concept development. When various concepts are being discussed, having them adjust the visual aids with minor parameter adjustments could be very useful for transferring or sharing information.

Better organisation of stored files is another benefit. With the SQL saving the iteration parameters, it is not necessary to store any CAD files, as these can be recreated from easily searchable parameters in a matter of seconds. This decreases the storage size requirements and forces a good organisation of information. This is especially useful where more people work on the same project.

The details of the shapes can also be reviewed outside CATIA, potentially saving on company-wide licence requirements.

Overall, automation of CATIA for shape generation can be an inexpensive method to promote and support good design practices such as knowledge management.

3 other potential uses will be outlined in the other section of this thesis: braiding simulation, meshing, weave modelling.

3.4 Meshing

In general, meshing is very important for all simulations. A bad mesh can significantly influence the results of the simulation, undermining its functionality. Mesh needs to be fine enough as not to neglect any geometrical features. Yet making the mesh excessively fine will result in unacceptably long runtimes.

The aspect ratio, mesh continuity and other features are also important, so that the mesh itself does not influence results of the simulation. Various non-standard features in the mesh might crash any simulation due to the built-in calculations; the simulation software usually works only with certain mesh standards.

In system of simulations multiple software use meshes to iterate over the part analysed. It is therefore of high importance to find robust methods to provide good quality meshes automatically, to any combination of parameters chosen for the part in question. As with all the other modules in system of simulation, UAV spar was used as an example for the development of the meshing scripts.

3.4.1 Outline of meshing options

It is a standard practice to generate mesh within the software environment it is subsequently used in, usually after importing solid or surface from CAD. This is often quite a manual process that requires an expertise to do in reasonable timeframes. This is especially true for more complex components. Meshing usually involves creation of further geometries within the FE software. This is done to better define mesh-seeds, number of nodes on each edge, and vary how fine the mesh is in different regions. FE software is usually less robust in geometry creation than CAD software. Also, this mesh adjustment would introduce unnecessary level of complexity into the system of simulation, making it almost impossible to automate. Therefore, alternative options for mesh generation will be explored. There are 4 major meshing strategies that have been identified and considered.

Option 1 is mimicking the normal approach of meshing within the FE, while automating the process.

Option 2 is to explore specialized meshing software; importing CAD into them and exporting ready to use mesh to the FEA and other simulations.

Option 3 is to use robust geometry creation of CAD systems, namely Catia, to automatically create segments of the main surface that can easily be translated into a mesh. Segment in Catia would be one element of mesh in FE software.

Option 4 is to mesh purely mathematically. Populate a virtual surface by nodes, connect those to form elements, and export the collection into the FE software.

The options are explored in several ways. The robustness is perhaps the most important requirement for the system of simulations, followed closely by quality and run-time. Secondary requirement is to minimize the level of development required from the engineers, so that the method can be used in industrial setting.

3.4.2 Option 1 – Meshing within the software used for analysis

This would be an ideal option if it was robust. If the software came with robust methods to automatically generate functional meshes, it would make the development of chained simulations significantly easier. Implementation of this method was attempted.

For this option to be suitable, all software in the chain would have to fulfil this requirement individually. The Abaqus was selected as the trial software, as it is quite frequently used for generation of meshes. Various settings have been tried in Abaqus for the mesh. The aim was to import the surface of the spar and automatically generate the mesh. With only 1 imported surface the success rate of automatically generated mesh that could be used without adjustments was at best below 20%. Usually, a random segment of the part would end up with elements with aspect ratios that cannot be accepted by the solver.

The surface was segmented into almost flat sections and corners, to attempt meshes at simplified geometries. This improved the resulting meshes visually but did not prevent the issue of faulty element generation.

Brief test of automated mesh generation in PAM-RTM resulted in mostly usable meshes. However, the meshes did not seem much better visually. It is likely that the main difference is that PAM-RTM simulation has lower requirements for the quality of the mesh than Abaqus.

Overall, the conclusion is that current meshing capability of these software is suitable only if manual adjustment is available, not for completely automated chain with varying shape parameters.

An option that lies somewhere between option 1 and option 3 is using this software to automatically generate meshes in various sections separately with different setting that adjust based on parameters. However, options 2-4 are likely to produce more flexible meshes and it would be inefficient developing this method for every software used separately.

3.4.3 Option 2 – Specialized meshing software

The most commonly used specialized meshing software is probably Gmesh; Gmesh is a lightweight Python scripting enabled meshing software. It has been briefly tried out by the author, and comparable amount of development would be required to make this work as for option 3 and 4. However, due to authors previous experience with CATIA and Python those 2 options were preferred. Therefore, author refrains from comparing this option with the limited review. From the brief review of tutorials and for basic meshing and scripting author did not find immediate motivations to choose this route over the next two options discussed.

Hypermesh is also sometimes used specifically for meshing. The files are then imported into other software for analysis. However, all that applies for Gmesh applies here as well. Also, the Hypermesh requires additional licences and hence costs. It uses quite specific scripting language; therefore, additional learning is required for using Hypermesh for automated meshing. On the other hand, because it is widely used commercial software it is likely more robust than Gmesh.

3.4.4 Option 3 – Catia meshing

This option is the least common, author is not aware of anybody else attempting this method. The main benefit is a good control of the mesh, while the development is relatively straight forward, compared to option 4.

Previously developed automatically generated spar shape was used for the meshing study. Building blocks of the meshing codes were developed using the VBA recording available within Catia. This was later translated into Python code. The development was similar to that of automated shape generation discussed in previous section of this chapter.

Various approaches were tested out in CATIA, usually consisting of splitting the main surface by spanwise plane cuts and generation of equidistant points along each of the cross sections. The building blocks used for the mesh are planes, intersections, points, splines, cuts, and “multi-section surfaces” respectively. Sweeps and fills were also considered for the final element surface generation, but the “multi-section surfaces” proved more reliable than either, and faster than sweeps.

The construction geometry of the final version is shown in Figure 3.5.

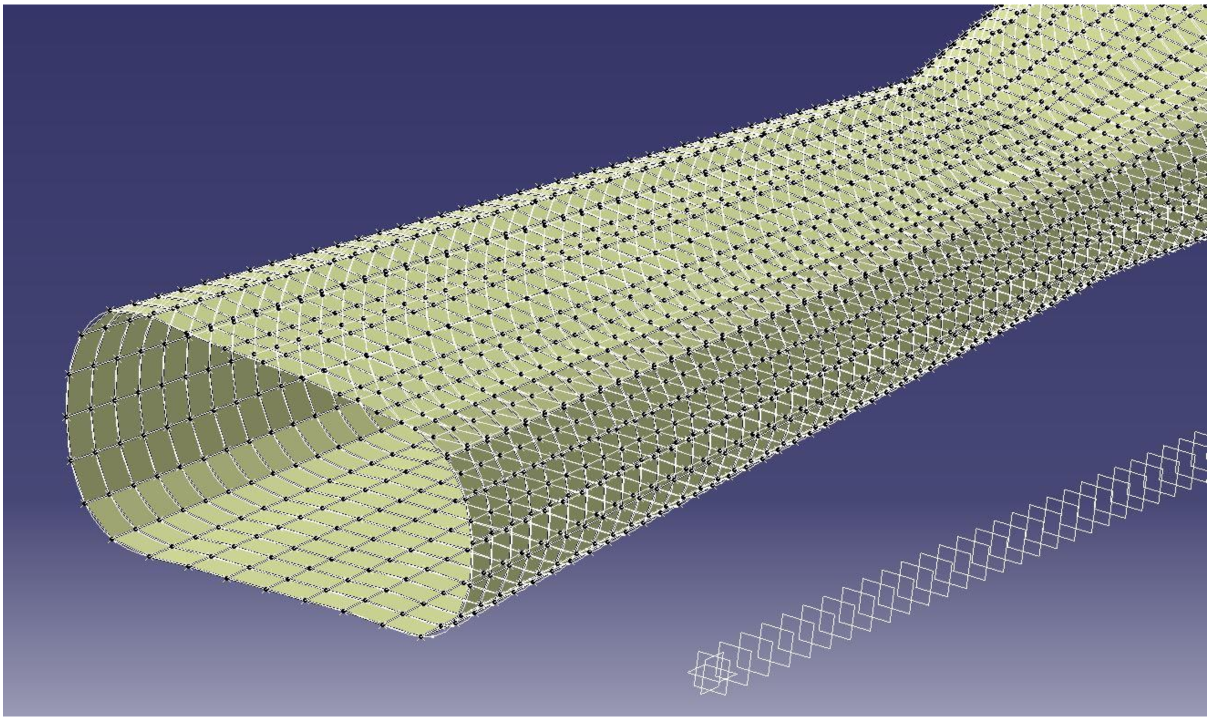


Figure 3.5 – construction geometry for CATIA mesh

The intersections are parallel with the root of the blade, this could be a limitation at high angles of wing sweep. For this part’s geometry this procedure is quite robust, for more complex geometries more functionalities would have to be added. The spanwise segmentation is performed based on variable elements size, corresponding to local cross-section; this is further explained in next section, as the same method is used for numerical meshing.

The CATIA macro recording does not record all the necessary features required for complete control of the geometry generation. For instance, the default direction of cutting, point placement, etc., is not recorded. This means that upon rerun of the same script with different parameters or on different pc it is likely that different default direction is chosen. However, it is possible to define the direction as a geometry feature and then use this to specify the orientation of other geometries. For this knowledge of the specialized CATIA scripting functions is required. To the best of author's knowledge there is no comprehensive library, but various scripts are available online to adapt and learn from. This would be likely considered a disadvantage, if this method was considered in industry for new product line.

The main disadvantage of this meshing method is likely the running time. For fine meshes this procedure creates a lot of geometry in Catia. This increases runtime exponentially as CATIA checks previous geometry for issues when creating a new one. There are various things like "screen-updating" and geometry "update" that can be switched off to speed up the script. However, although some improvement is achieved the runtime still increases with each new geometry, disqualifying this method for large parts or very fine meshes. The comparison of runtimes is shown in numerical mesh section later.

The benefit of this meshing method is a good control of the mesh used. Another benefit is that the meshing method is quite independent of FE software, which is ideal, as remeshing is only required where new geometry is involved.

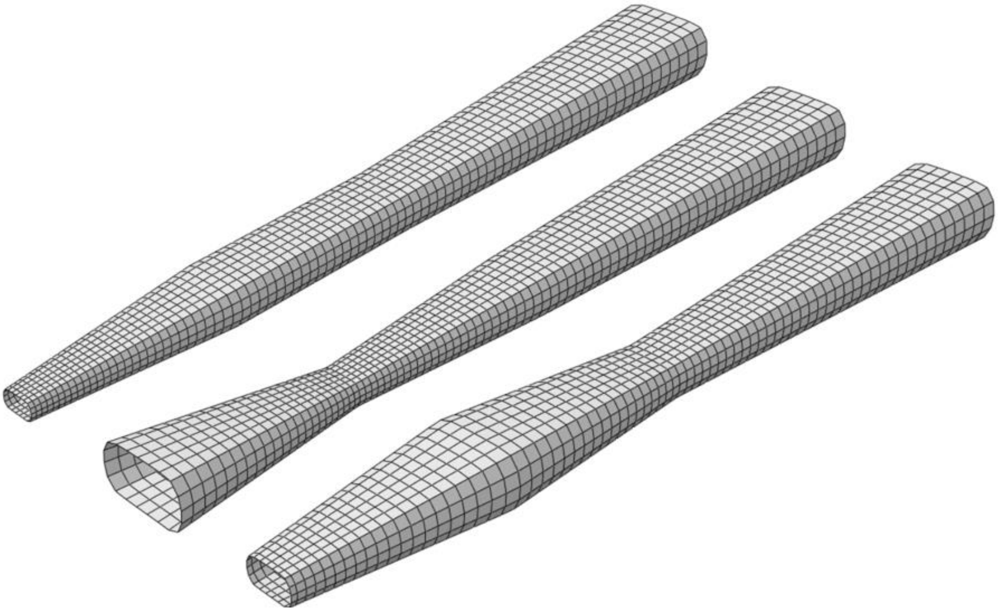


Figure 3.6 - example meshes

Originally this method was supposed to provide a pre-mesh for Abaqus rather than a full mesh. Pre-mesh would be segmented main surface, while retaining all curvature. The envisioned result would be geometry split into very simple shapes that could be easily meshed within the Abaqus, not losing any fidelity. This would also allow for different mesh sizes based on local requirements in automated manner. However, this method was abandoned due to the volatility of imports into the Abaqus. For some unknown reason additional nodes are generated in various spots, making referencing problematic and automatic meshing in Abaqus almost impossible. This issue was raised with Dassault support, but response was never obtained. Therefore, the above-described method of complete meshing in CATIA was adopted instead. However, with future versions of Abaqus, if the import reliability is improved, this could be a method which achieves lower runtimes and is quite fast to setup. It was verified that this was an Abaqus issue as the same file could successfully, without additional geometry, be imported into Patran and PAM-RTM.

3.4.5 Option 4 – Analytical mesh

The 4th option is to develop a bespoke scripted meshing algorithm. This is the fastest method in terms of runtime, but slowest in terms of development required. For simple shapes that are easy to describe mathematically this is not too much of a problem. For shapes which were otherwise defined by complex surface operations in advanced CAD software, this could become an equivalent of designing a new CAD package, which is not recommended.

There are also less obvious benefits and downsides of this approach. For instance, the analytical mesh approach is less, if at all, affected by software or operating system updates. While CAD meshing script will be subject to changes whenever either Windows or CATIA make a significant update, these cannot impact purely numerical mesh. On the other hand, anything that will require change in the definition of geometry, for example adding a degree of freedom, will require additional work. This might be difficult to predict when deciding the mesh type for a specific component line.

The part used as a demonstrator is somewhere between the two. It contains multiple curvatures, but all of them can be described by combination of airfoil point clouds, interpolations and arcs. Therefore, it has been deemed necessary to try out this method.

3.4.5.1 Development

The spar is defined by airfoil at various positions along span, by chord length of spar, and by radii connecting the sidewalls and airfoil sections.

Each of the defined cross sections is first turned into point defined shape. Top and bottom surfaces of the spar are constructed from interpolation between airfoil points. The mesh size, which is stored in global variable matrix, is used to find the distance between two neighbouring points. Based on the required distance along the section edge, the two nearest airfoil points are found. The precise position of the new mesh point is than obtained through interpolation between the two points. The start and end point of the segment are defined by the “chord_min” and “chord_max” variables, which are the percentage chord locations of the front and aft wall respectively.

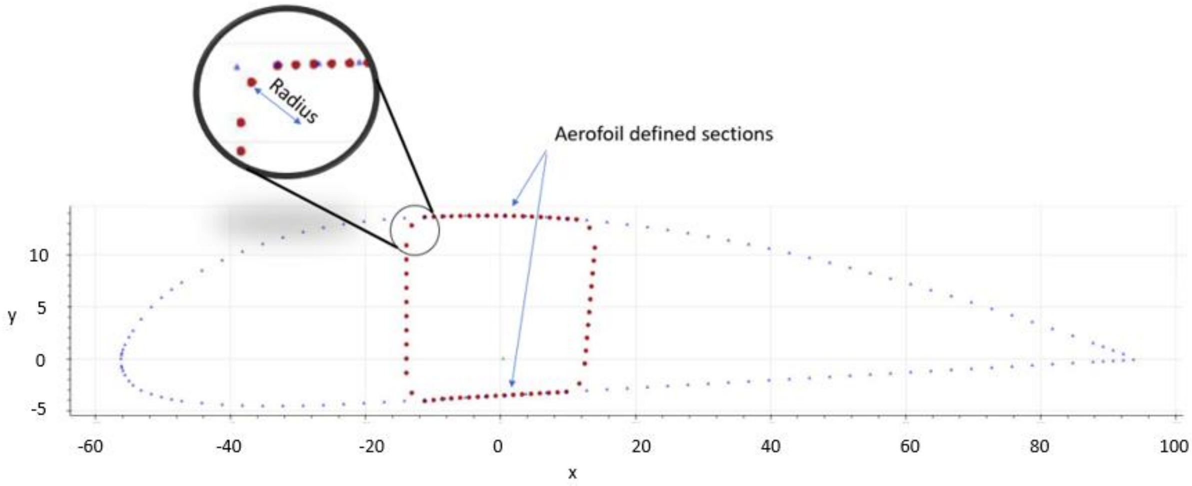


Figure 3.7 – cross section of a spar mesh

To define corners, start and end points of the arc have to be defined first. The points are where cross section intersects the dotted lines in Figure 3.8. The mesh propagates clockwise, therefore corner 1 and 3 start with the end points of airfoil section. The corners 2 and 4 start with wall end points. The start and end wall points are defined by radii. The wall edge points are radius in x and radius in y away from the airfoil edge point.

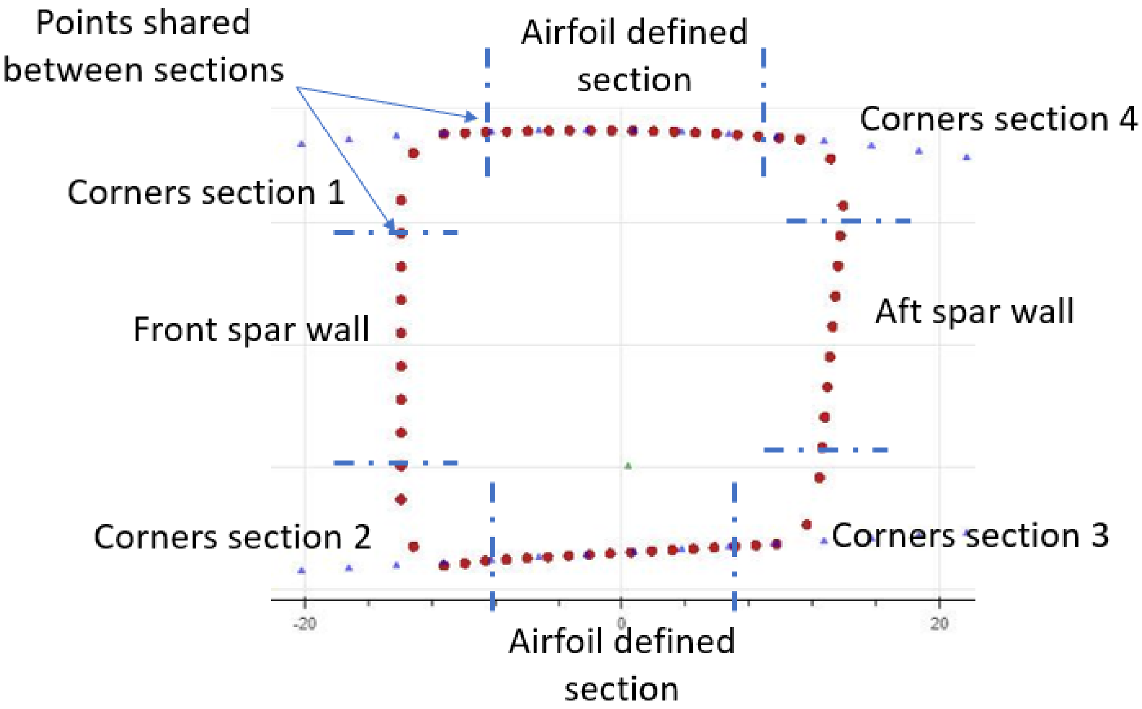


Figure 3.8 – segments of cross section

After the corner end and start points have been established the arc between them is produced. The segmentation is done by angle, using polar transformation from x and y to element angle. Requirement for at least two elements in corner is enforced, no-matter what the shape and size parameters are. This prevents small radii from being replaced by very sharp corners. For other parts corner mesh-sensitivity analysis might be required to select the meshing rules for corners.

First cross-section is segmented based on prescribed mesh size. All other sections need to maintain the same number of points around cross section. This way rectangular mesh is maintained with reasonable aspect ratio. All sections specified directly by airfoil are segmented as per above. The number of elements for segment in the following cross-sections is defined by the root segment. In other words, only the root section uses the mesh-size parameters.

To produce a spanwise mesh with roughly square elements, the element spanwise size has to be adjusted according to local perimeter. The element size is calculated based on the logic shown in Figure 3.9.

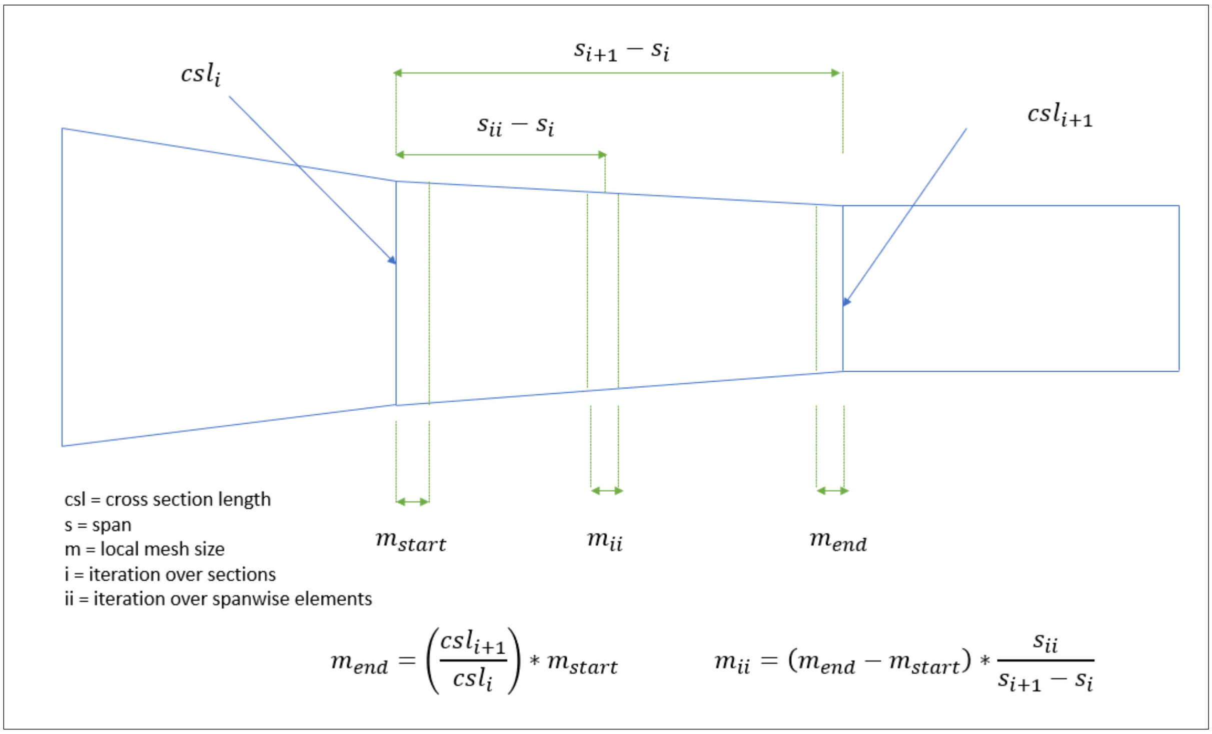


Figure 3.9 – spanwise element sizing

Once the matrix of mesh sizes has been created the interpolation between main sections, those defined by airfoil, provides all the remaining mesh points. When the difference between two main sections perimeters is small, the mesh is not adjusted at all, as it is found to yield less issues than very small adjustments. With the formulas given it sometimes happens that the last element before the airfoil section is too small spanwise. In such case the mesh sizes between the relevant two airfoils are adjusted to prevent this. All this mesh standardisation is mainly required because of the Abaqus methods used to match elements with segments, which is done by spherical definition of element location. This is described further in FE structural analysis section.

The mesh is exported as an .inp file accompanied by .npy file with spanwise size of elements, which aids with the assignment in Abaqus. The .npy export uses standard numpy library. The .inp file export required a development of small bespoke script to construct the .inp file out of 3D coordinates. This script can be useful standalone, for other simulation projects.

3.3.5.2 The positives and negatives of purely numerical meshing

Due to a complete control over how the mesh is generated, it is possible to increase mesh density in key areas. For instance, in Figure 3.10 the corners have smaller element sizes than the rest of the cross section. This is quite easily implemented within the meshing script. However, it does pose additional requirement when implementing the Abaqus element assignment.

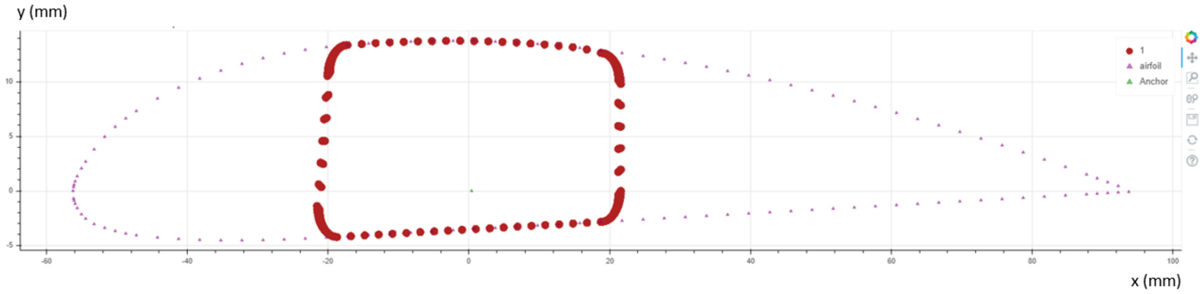


Figure 3.10 – finer mesh in corners

Another benefit is the runtime. Figure 3.11 shows the runtimes of the option 3 for comparison, the only other fully developed meshing method. Two separate datasets are shown, each run on different machine. It is clear that the increase in mesh-time with increasing element count is more than linear. The run-times start at about 5 minutes but can increase up to almost 100 minutes for close to 6000 elements.

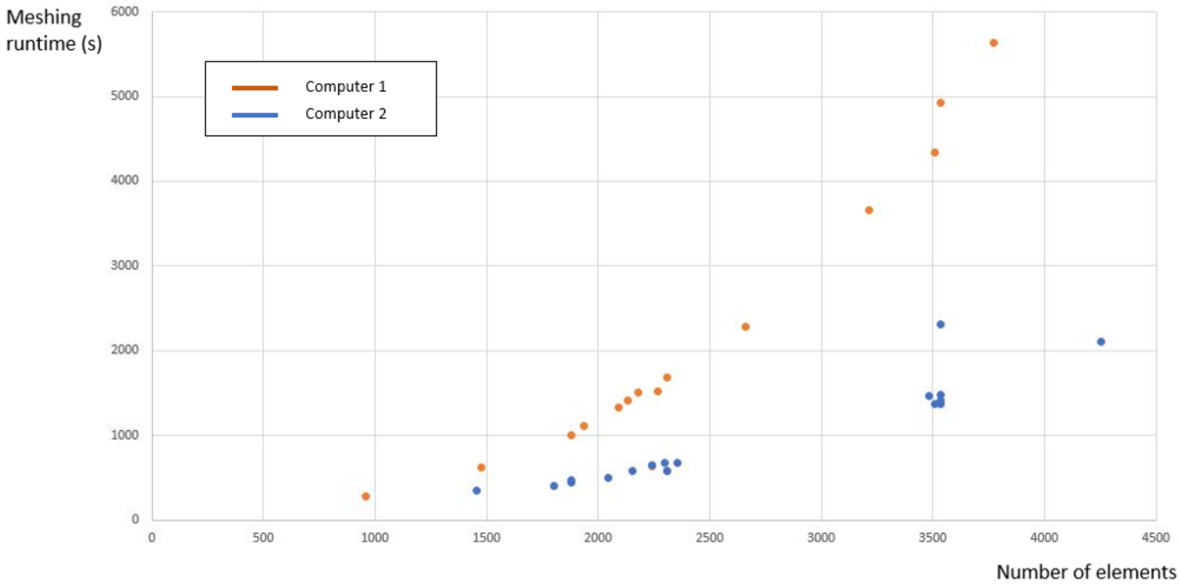


Figure 3.11 – CATIA mesh run-times

Figure 3.12 shows the run-times of numerical mesh. The graphs are separate due to the order of magnitude difference. Due to various loops used to fix certain issues, and closer dependence on simulation variables, the variance in run-time results is quite large. The sample shown on the graph is taken over various parameter combinations. Even with the variations, it is clear that most results adhere to linear increase of mesh run-time with the increased number of elements. Longer runs were also executed to establish the limitations of this, but these are not displayed on the graph for clarity. 300 thousand elements took approximately 2 minutes to run, therefore the runtime of the numerical meshing is unlikely to ever become an issue. The runtimes include the generation of .inp file.

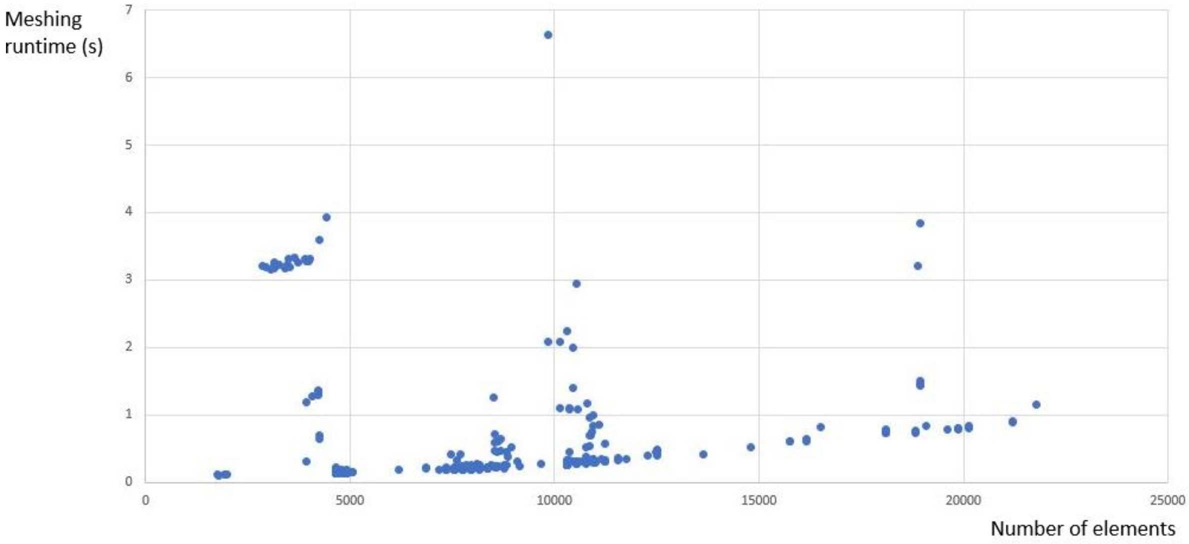


Figure 3.12 – numerical mesh run-times

However, the numerical meshing requires minor adjustment of the Abaqus import. This adjustment makes the analysis 5-20% slower with equivalent mesh size. Of course, with decreased element size, which is possible with the numerical mesh, analysis run-times can further increase. This is further discussed in dedicated FE structural analysis section.

Less obvious benefit of the numerical simulation is the independence of any commercial software. Not only does the same script run easily on any version of any operating system, but also it does not require any commercial software. Being only dependent on Python libraries makes this method less prone to bugs caused by updates or transfers between machines, operating systems, or software versions. Throughout the development of the system of simulation considerable amount of time was spent on rework due to such changes, making this benefit of perhaps surprising importance.

The main downside of numerical meshing is that it takes probably longest to develop, out of the reviewed methods. The development time will likely increase exponentially with the increase of part complexity. Where the 3rd meshing option can outsource a lot of geometrical calculation to the CAD software, any complexity in this method has to be thoroughly understood and described by mathematical equations. The part in this study has relatively simple spanwise iteration, other shapes might not have easily standardized method for the iteration. It is quite possible that with some shapes development of robust iteration would be the most time-consuming task.

Any misconceptions about how the part might need to vary throughout its development will also become more expensive. Variation of an additional geometry will be significantly more difficult to implement than for the other methods.

To troubleshoot this method, additional scripts will be required, but due to the nature of system of simulation, this is more-or-less true for all 4 methods considered.

To summarize, for the UAV spar this is by far the most suitable method, as the mathematics are relatively simple, and runtime is of utmost importance. However, the author would be more inclined towards other methods with more complex geometries.

3.4.6 Summary and recommendations

For the system of simulations, the analytical mesh is being used. It has very low run-times and is quite easily troubleshooted. However, it does make the Abaqus module slightly slower and slightly more difficult. For more complex part this method might not be suitable.

CATIA meshing was also developed but the significantly longer run-times make it difficult to iterate over the design using this method.

Other options might still benefit from closer examination.

The summary of the key differences between meshing methods is shown on table 3.1.

Table 3.1 – meshing methods

Meshing method	**Development**	**Run-time**	**Scalability**	**Interoperability**	**Update sensitivity**
1. Within every software	Largely in the hands of software provider	Medium	Yes	bad	medium
2. Specialized meshing software	Unexplored	Likely fast	Yes	Unexplored	Medium-high
3. CATIA mesh	Fast and easy	Grows exponentially with mesh	No	medium	high
4. Analytical	Depends on complexity	Fast	Yes	good	low

3.5 Braiding module

This section describes the development of kinematic braiding simulation tool. It uses mandrel shape and braiding parameters to calculate local braid angles and pitch information.

The explored methods mirror the meshing section. First an automated method was developed using bespoke scripts and CATIA geometry. Later, an alternative purely numeric method was implemented, which was only possible due to the development of numerical meshing module. The CATIA CBX module was also tried, but it is not suitable for SySi due to the lack of scripting functionality.

Both of the developed methods rely on equations provided in the thesis by Ravenhorst (5).

3.5.1 Description of the current meshing method

The main input required for the braiding simulation is the mesh. The input mesh is spanwise segmented by z distance and has the same number of nodes for each cross-section. The details of the mesh are presented in section 3.3 above. If other mesh was used, some data management or adjustments to the braiding scripts would have to be done. For this part rectangular mesh with approximately square elements is suitable. For more complex part triangular mesh might be better. Triangular mesh would likely accommodate for unconventional shapes better but would make the script a bit more complicated.

Other inputs required are the braiding parameters. These include mandrel speed, number of spools, number of modelled spools, initial spool distance from initial fell point (IMD), and guide ring diameter (gd). The spool travel speed is kept constant, as any ratio of mandrel speed to spool rotation can be achieved by varying only one of these parameters. The absolute values are less relevant as the manufacturing time is currently not an output parameter. If it was required, the spool speed would have to become an input variable.

The script is structured into two levels. Simulation setup level is the outside of the box in Figure 3.13. Yarn level and analysis level is the inside of the box in Figure B1. The setup script (Braid_CMD_S.py) is run from the system of simulations. This iterates through yarns and stores the data. For each yarn it executes the yarn level, where a main braiding function is run (Braid_main_S.py). In this function all fell points for a single yarn are obtained.

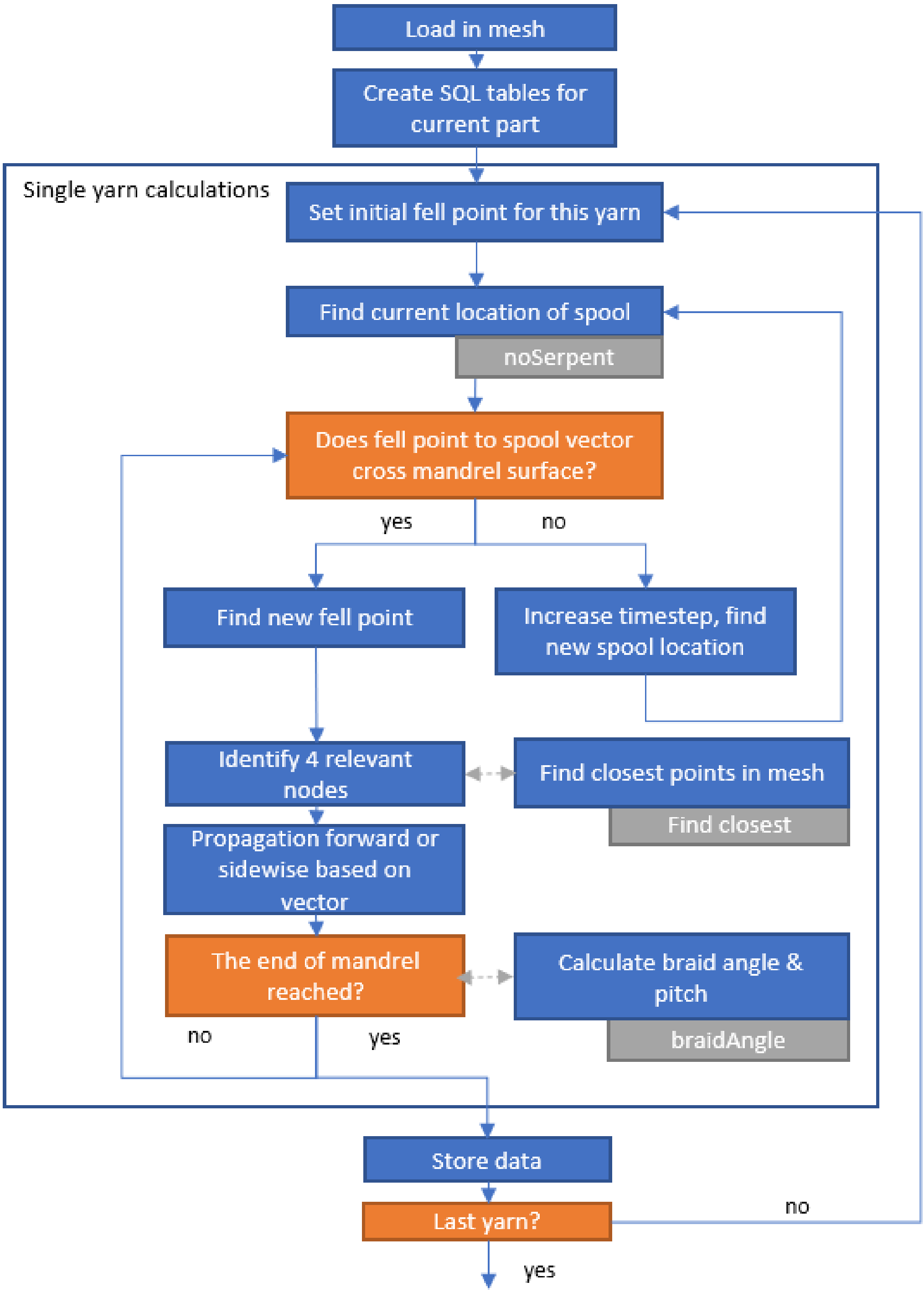


Figure 3.13 – braiding simulation script structure

Multiple different propagation methods were tried. The one used in the end consists of two main parts. First, the position of spool relative to fell point is checked. Second, the new mesh element to which yarn propagates is established. The position of spool relative to fell point is checked, by considering the relative vector. This is shown in Figure 3.14. Fell point is defined as the last point of contact between yarn and mandrel. Guide ring position is used instead of serpentine motion, continuous constant rotation is assumed.

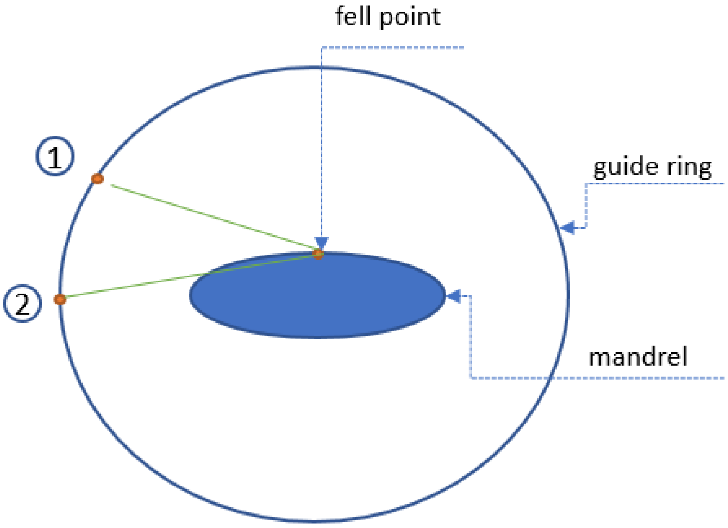


Figure 3.14 – fell point to spool position vector check

If this vector crosses the mandrel, the next stage of propagation algorithm is initiated; this is shown by position 2 in Figure 3.14. This is because the yarn is in contact with the mandrel, and therefore can be assumed that it will not move freely. Yarn slip is currently not implemented but should be feasible to include if necessary. If the yarn does not cross the mandrel, it can still pivot around the last known fell point. Therefore, in such case the spool position is iterated, maintaining the fell point. This means that the fell points are not iterated with time. The spool position is iterated with time, therefore the mandrel speed is also taken in account, as relative spool position. The fell points are iterated only when the geometric position requires it. When the fell point is iterated, the spool position remains the same, as it is possible that multiple fell points will be created once the spool moves beyond a horizon.

The propagation of fell point is depicted in Figure 3.14. Once fell point propagation is required, the element to which the yarn propagates is identified. The 3 options are depicted by “e2”,”e3” and “e4” elements in Figure 3.15. The direction is obtained by considering the lines p2-p4 and p4-p3. Loop through points in all positions on these two lines is created, this will be called the crossing point (CP). At each location CP, the total distance of yarn is calculated. This distance consists of CP➔Yarn and

CP➔ original fell point distances. The shortest total distance obtained is where the new crossing point is. Based on this, the next element is selected.

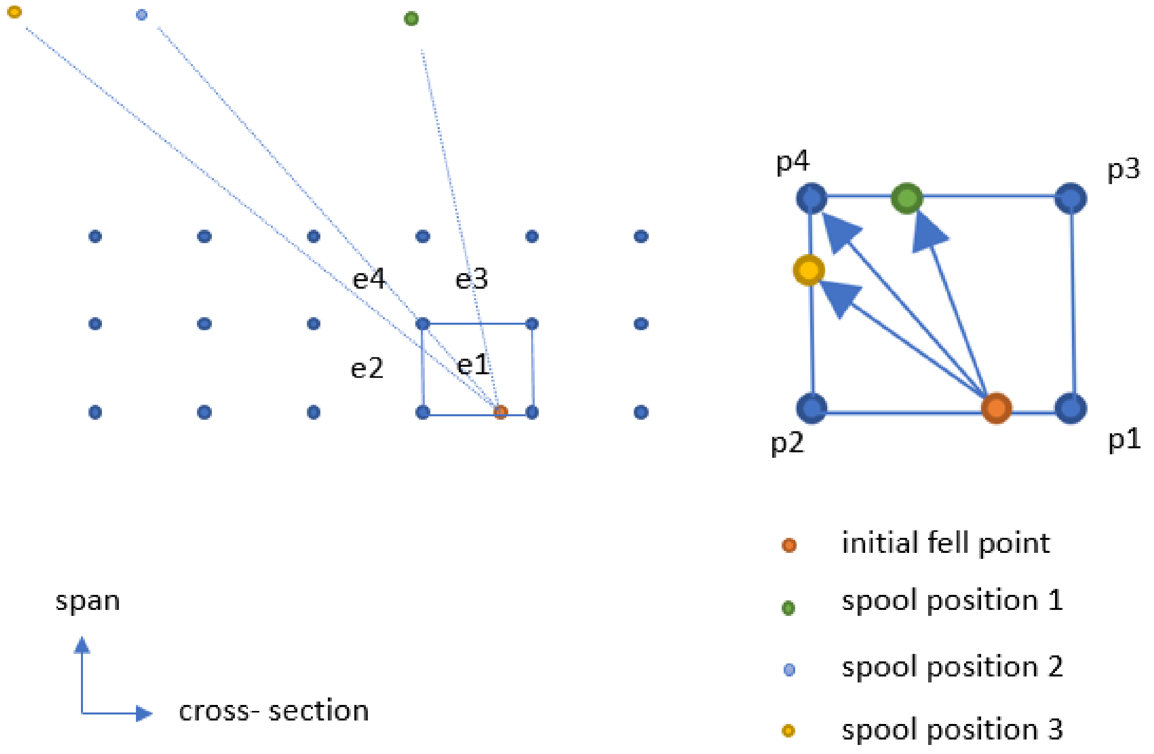


Figure 3.15 – braiding simulation fell point propagation

The initial distance of guide ring from initial fell point does not appear to have much effect over the majority of the mandrel. However, well selected initial mandrel distance minimizes the run-in distance, where the braid angles stabilise.

With current standardised method the propagation to next element is quite easy. If the propagation is required spanwise, the same node numbers in the next cross section are used. If the propagation is required in cross-section direction, the next nodes are used on the same cross sections. If the propagation is perfectly diagonal, both propagations are used. Originally, this was done by 3D proximity of nodes, rather than standardised cross sections. For various tested mandrels this was less robust, and more calculations were required, making the troubleshooting more difficult. However, for complex parts where it is impossible to maintain number of nodes for each cross section, the 3D distance search method would have to be re-implemented.

Initial 2 nodes that define starting fell point are found by 3D distance measurements. A starting point is selected based on direction Warp/Weft and number of simulated spools. This is done so that the distribution of simulated yarns is uniform over the whole circumference. Then, distance between each node at z=0 and the starting point is calculated. The two closest points are p1 and p2, as per Figure 3.16.

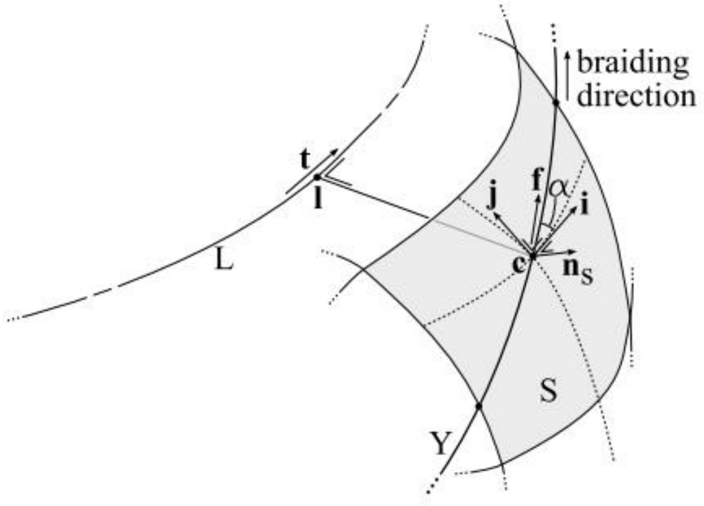


Figure 3.16 – braid angle calculation (5)

At each fell point the braid angle is calculated by the following formula(5):

$$\alpha = arctan\left|\frac{f \cdot j}{f \cdot i}\right|$$

where

$$j = \pm\frac{n_s \times i}{\|n_s \times i\|}$$

Vectorized form is used as it is easier to work with for complex mandrels. The i is the unit projection of t. Where t is the unit tangent of the centreline (L). n_s is the normal at fell point. This is approximated as the normal of the element that contains the fell point, which can be easily calculated from the 3D positions of the closest nodes.

Pitch of the fibres is calculated using simple trigonometry, with the local perimeter and local braid angles.

After all necessary yarns are calculated the pitch and braid angle information is stored in SQL-server database. Each point is specified by 3D position and yarn number.

Only representative number of yarns is modelled. The numerical braid simulation allows for modelling significantly more yarns than the original CATIA one described below. Therefore, about 60 yarns are usually modelled. This gives sufficient information for all sections used in follow up software.

Further details of the braiding simulation can be obtained by reviewing the annotated scripts on GitHub. (1)

The braiding simulation can be used separately from the complete SySi. This is because some potential users expressed interest in specifically this aspect of the work described in this text. To facilitate this, altered version of the simulation is available on GitHub(6). This version can be used on manually created CATIA files. This version includes visual display of yarns in CATIA.

The braid of one yarn on the SySi part takes about 1 second with this method, about a minute for a typical simulation. Even though this simulation is order of magnitude faster than CATIA based one described below, some speeding up measures were still implemented. For instance, obtaining mandrel cross-section circumference at every section, by calculating distances between all the points, was taking too long. Therefore, this calculation is executed only on 4 cross sections, for the other circumference values interpolation is used.

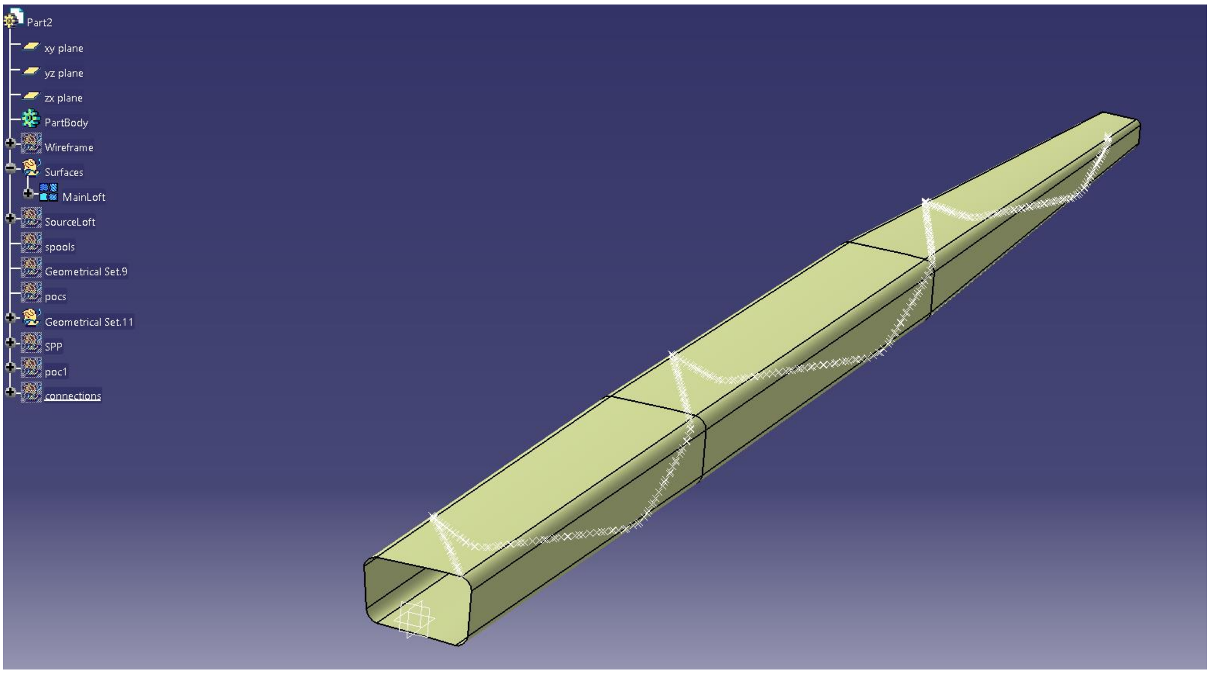


Figure 3.17 – example part, fell points for one yarn

3.5.2 Differences with CATIA based braiding simulation

The main problem of braiding simulation stems from the complexity of the shapes and their inherent effect on the simulation. Geometrically, all the mathematical formulas involved are quite simple. However, obtaining various vectors from points on a complex mandrel is a problem. In this braid simulation method CATIA is used for the study (Braid_CMD_C.py). The development of this followed similar procedure to CATIA scripting described in CAD module section.

The propagation in this version is done geometrically. A line is created between the fell point and the spool point. An intersection is used to produce a point on that line, which is then projected on the surface of the mandrel.

The main downside, and reason why this simulation was replaced, is the runtime: 1-6min/yarn depending on mandrel and parameters. To make this simulation accurate and reliable, small timesteps are required. However, with large number of geometries created in single CATIA file the simulation slows down significantly. This was improved by segmenting the simulation. The CATIA would close, then open and continue simulation based on the last recorded fell point and spool point without the previous geometry present. The restarting of CATIA prevented it from slowing down throughout the simulation, but it added some flat run-time per restart.

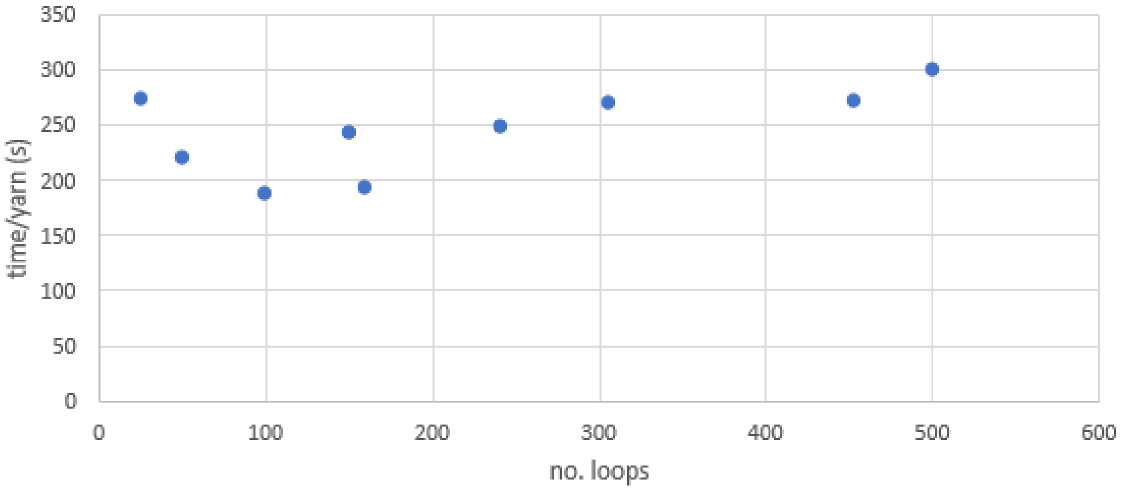


Figure 3.18 – optimising CATIA restarts

Figure 3.18 shows the time per yarn required with different closing strategies. When the closing was too frequent the time increases due to the restarting times introduced. In this trial 500 iterations completed the simulation; without closing, the time per yarn was highest. More data could be generated, but certain amount of randomness exist so the exact optimum might depend on too many factors. However, it was assumed that closing after 100 loops leads to shortest runtimes. 1 loop corresponds to one fell point propagation.

The restarting also outlined one of the issues with using commercial software for the integrated simulation system. After transition from windows 7 to windows 10, the restart times for CATIA have significantly increased. This changed the optimal amount of geometry kept in one file before restarting. It also made this simulation even less suitable for high number of component iterations. It is difficult to assess what was the specific reason for this, but it is clear that updates to both operating system and commercial software can have large impact on any bespoke scripts developed.

Another theoretical option to speed up the CATIA based braiding simulation is to delete source geometry. However, the isolate command is required so that CATIA does not run into errors. The isolate command does not record using the macro recording functionality and author did not find any evidence of it being scriptable. Therefore, this methodology was considered currently unfeasible.

The original CATIA braiding simulation did not require a mesh to run, as the propagation was based on the geometric functions and the mandrel surface.

Further details of this simulation will not be discussed, as it was replaced by the numerical version. The last version of this is still available withing the GitHub repository, in case it turned out to be useful, or some of its aspects needed to be reused.

For example, this method required vectors and point positions to be extracted from CATIA, for calculations. There is no built-in function in CATIA that would facilitate this. Neither could author find readily available solution on the forums. This was solved by a specialized script (vecEX2_C.py). The method consists of hiding all geometry in CATIA except the point or a vector and saving the CATIA file as .wrl file. This file is than interrogated by the script and the 3D location can be extracted. This can be useful outside of the braiding simulation as well.

In one of the versions of this simulation, varying mandrel speed across span was used (Braid_CMD_S.py). It is iteratively adjusted for the changes in cross section to allow for reasonably constant braid angle. This was implemented as the optimisation would come up with shapes and mandrel speed combinations which would often result in impossible local braid angles.

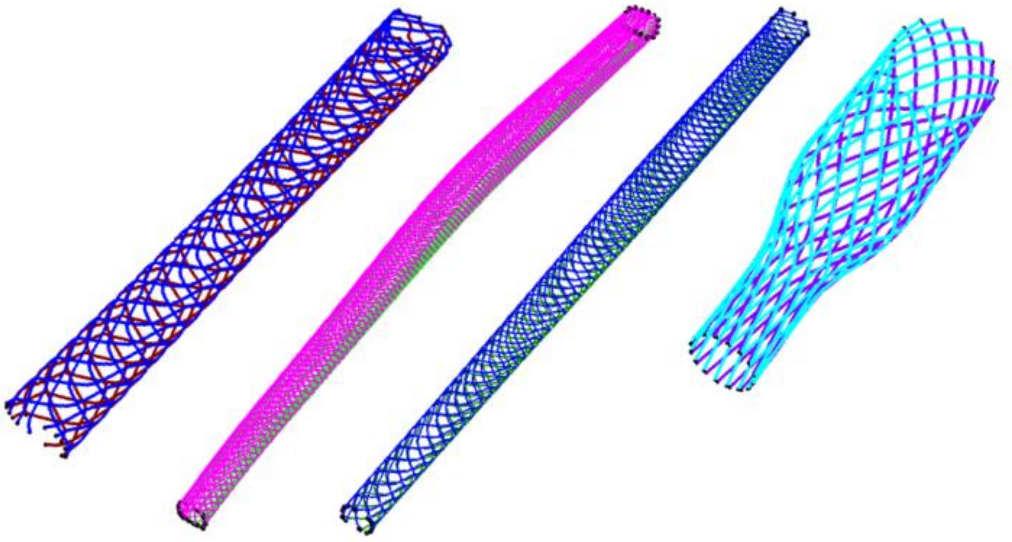


Figure 3.19 – example runs of CATIA based braiding simulation

3.4.2.1 Testing against CBX

To verify that the kinematic simulations developed are reasonably accurate, the initial braid simulation using CATIA geometry was compared to CATIA CBX braiding module. The CATIA CBX module was previously assessed at NCC and it was considered accurate after the run-in region for circular mandrels.

Multiple shapes were compared, the cross sections were kept circular, as no investigation of quality of corner predictions or similar was available for CATIA CBX. Also, CATIA CBX only outputs averaged braid angle at positions along span, so additional work on extracting more localised braid angles would be required.

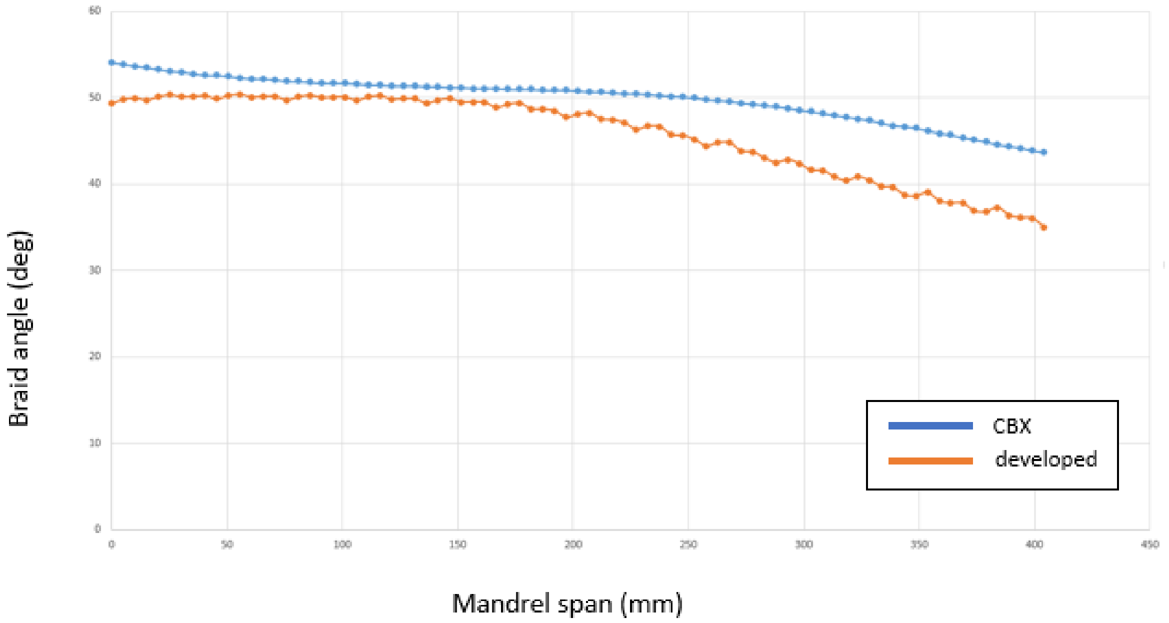


Figure 3.20 – comparison of CBX and developed braiding simulation

Figure 3.20 is a good representative of the comparison trials. Clearly, there is some discrepancy between the two simulations. On circular regions, both simulations typically agreed with differences of less than 3 degrees. However, on tapered regions and on transitions between segments the developed simulations changed braid angles more rapidly than the CBX. The source of the discrepancy is difficult to assess. One of the reasons for this is the lack of visibility into Dassault's algorithms.

On a mandrel which is completely circular and un-tapered along the whole span almost perfect match is achieved, maximum of ±0.5 degrees difference.

Based on this trial, the simulation was suitable for usage in the SySi demonstrator. The precision does not need to be perfect; the braid angles just need to reasonably follow trends so that changes to material properties can be propagated into other simulations.

If this simulation was to be used on a commercial project, or any project where the part would be manufactured, the simulation would need a proper validation using manufacturing trials. This would likely need to be followed by tweaking of the code to solve any discovered issues.

This was not considered as a priority during the development of the SySi as the changes to the simulations were continuous. An example of these changes is the transition from CATIA geometry-based simulation to completely numerical one.

3.5.3 Troubleshooting

Troubleshooting is an important aspect of any system of simulation or optimisation. The importance is further increased by the bespoke development of braiding simulation, which is going to lack robustness due to lack of users randomly encountering issues.

The CATIA version of the simulation was quite good for troubleshooting. When error occurs the CATIA stops, but the part can be viewed to see where the error occurred. The CATIA also has a decent library of errors that are quite informative in terms of what the problem is. However, even here few additional troubleshooting measures were implemented. Most common measures are print outs of key variables that indicate potential issues. This includes for instance values for the surface normal, as wrong direction can easily be spotted and 0 values means it was not correctly retrieved from CATIA.

For the numerical meshing this is more important, as all troubleshooting methods have to be developed. Variety of messages helping to track the script progress are implemented. For instance, if fell point wasn’t iterated over 100 motions of spool, it is likely not tracking the direction correctly. Therefore, the normal is printed out, this way user can spot if the braid angle is just low and the simulations should be kept running or if the user should interrupt simulation as something went wrong.

Good tracking of typical errors and potentially problematic variables can speed up the development of script like this significantly.

3.5.4 Standalone benefits

The standalone benefits assume further development of the simulation, and extensive verification, supported by manufacturing trials. Although some comparison with manufactured components have been done, author assumes that the precision of the simulation could be improved, and special cases assessed.

Considering the limited development of freely available braiding simulations, this relatively simple simulation could serve as a building block for simulations in industry, for smaller companies lacking the resources to procure and help develop commercial software. Kinematic simulation of braiding could significantly reduce the amount of trial and error required, which is how braiding parameters for a braided part are typically developed.

The storage of braid data in MySQL can help inform future braiding work even without re-running the simulation. The data can be extrapolated to provide initial parameter estimates.

3.5.5 Catia weave modelling

This section focused on modelling local braiding setup with greater detail.

This exercise was executed for 2 reasons. First one was to establish how viable it would be to expand the CATIA braiding simulation by detailed weave model. Secondly it was another good way to demonstrate CAD/CATIA scripting.

The idea was, that since the splines are already available in the correct position from braid simulation, it should not be too difficult to add the undulation and create good weave model. Similar thing is currently done in TexGen or WiseTex. In ideal case the segment modelled, with all undulations and yarn interactions, would then be exported into FE (fluid/solid). This would then be used to provide local permeability and structural material properties.

Into the system of simulation this could potentially fit as the last check for the final design, as clearly this would be very computationally expensive. Alternatively, library could be built of previously modelled segments, that could be used to estimate material properties by means of interpolation. This method could be fast enough to fit into the Sysi.

Weave patterns were generated with relative ease using same techniques as for the other CATIA scripting work. The surfaces generated are shown in Figure 3.21. Solids can also quite easily be generated if that is required for a follow up software.

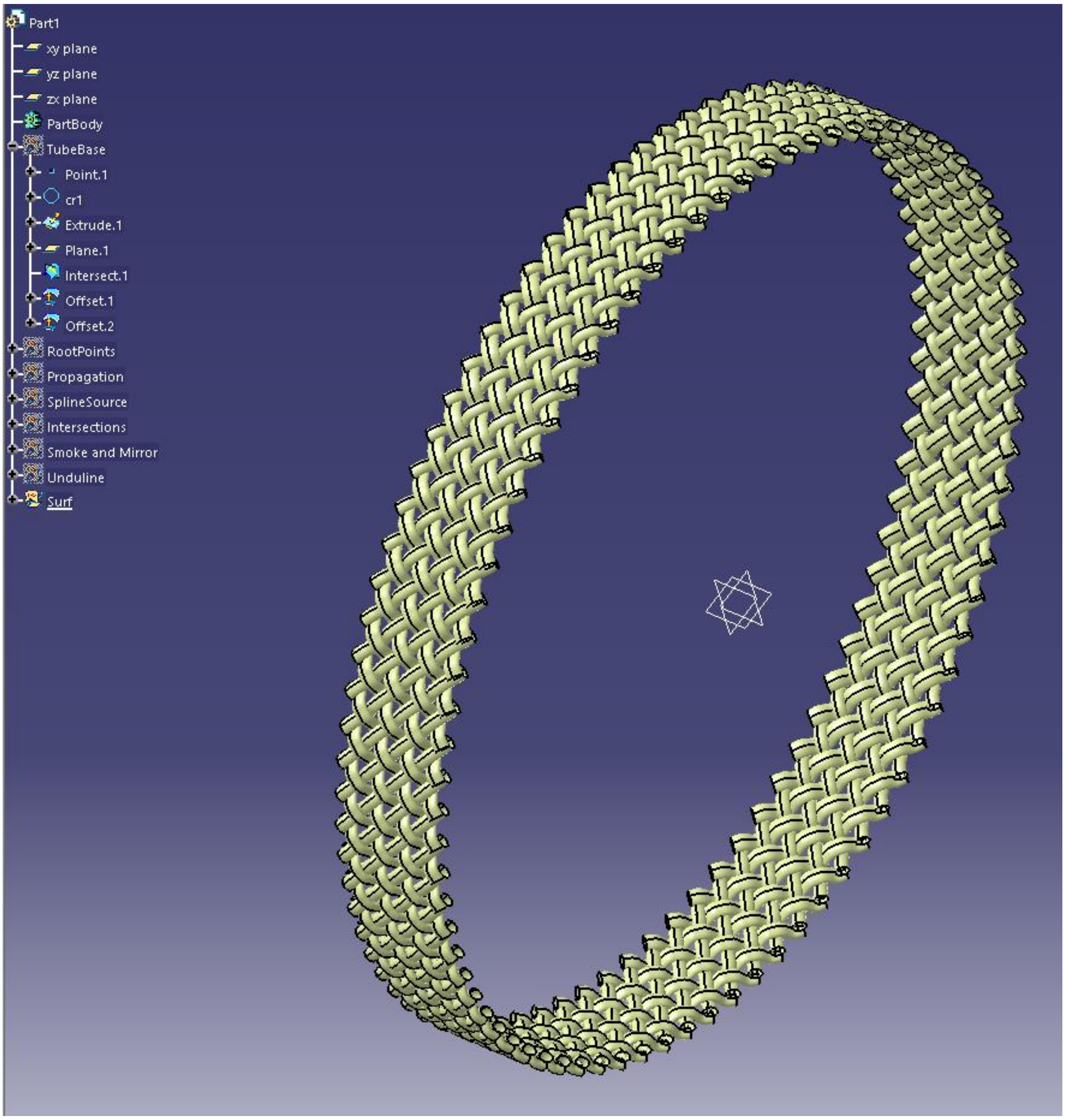


Figure 3.21 – weave modelling sample

As expected, the model turned out to be very slow. This problem is exaggerated as most of the geometry is being actively used for the display of the final shapes. Therefore, it is not viable to split the model into sections and then only assemble resulting solids in the final file.

Visually, on first glance, it appears very good. The sweep function shows good approximation of the yarn surface. The cross section of the yarn can be quite easily altered for the whole yarn, allowing for inclusion of yarn deformation effects in any following analysis. However, it would be significantly more difficult to consider the yarn deformation changing along the yarn, which would probably be more precise in many scenarios. The undulation is likely to cause the yarn to have varying cross section along its length, with reference to its central axis.

Catia does not have any tools to replace material among clashing. Therefore, the yarns have to be either modelled very precisely, or with a gap. If modelled with a gap, it might be possible to use Abaqus to create a deformed model, obtaining localised yarn cross-section deformation information. However, this is purely speculative as this experiment was not deemed particularly successful and hence the follow up work is unlikely to be revisited.

3.5.6 Braiding simulation summary

The major benefit of developing bespoke simulation of braiding is the adaptability of it for the system. Also having access to all the code building the simulation, allows for easy implementation of changes later in the project. Having said that, if a robust kinematic simulation is available commercially, it could save significant amount of development time for a system like this.

The presented braiding simulation works sufficiently for the demonstrator. With some additional work this simulation could likely be used on commercial projects. Bespoke developments would probably be needed.

3.6 Finite element structural analysis

For structural analysis Abaqus software was selected. It has a built-in Python library of functions, it is commonly automated, and frequently used for composite analysis.

The objective of Abaqus finite element model is to find the maximum deflection of the spar at a given load. The main development aspect for this module was the mapping of previously generated data onto the elements in the model.

The main challenge here was to create a robust automated analysis, which will accommodate for all possible shape change options and corresponding change in material properties assigned.

The units used within Abaqus in this project are ton for mass, mm for length, s for time, N for force, MPa for stress.

3.6.1 Material data processing

The data from the braiding simulation is imported, as a point cloud where each point is assigned braid angle and pitch value. These points are grouped into corresponding segments. Spanwise segments are simple due to the nature of the part, defined by spanwise distance. The cross-section segmentation is depicted in Figure 3.22.

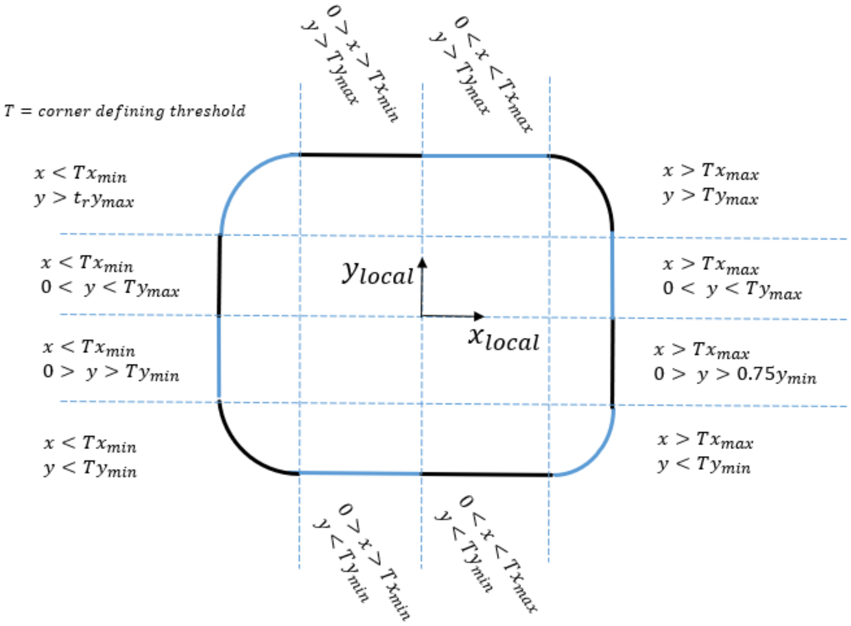


Figure 3.22 - cross-sectional segments of the spar

The script “segmentation.py” is responsible for allocating the data to the correct segment. In the first function, each segment is defined in terms of local coordinate system, which is defined by the centreline direction. The local coordinate system is also rotated by twist in the x-y plane so that corners are always well defined.

The main “segmentation.py” function then allocates every data point to the correct segment and averages the braid parameters. The output is a matrix of segments with braid and pitch data allocated. The boundary condition matrix is also outputted, which is later used to define segments in Abaqus.

Then “lam_tools.py” is used to translate the braiding variables into material properties. Due to a lack of better readily available alternatives the approximation is done using rule of mixtures and classical laminate theory. This neglects the effects of weave. However, considering the structural analysis only deals with deflection and does not deal with failure, this is satisfactory for the demonstrator.

It is assumed that in the industry this would be replaced by test data that could be interpolated for more accurate results. Alternatively, library of modelled braid structures from TexGen, WiseTex, or similar could be used.

3.6.2 Abaqus scripting

The Abaqus functions cannot be used as part of user Python library. Also, external libraries cannot be imported into the Abaqus library. Therefore, the Abaqus functions must be used through command line, which is controlled by the user Python library. The Figure 3.23 summarizes the logic.

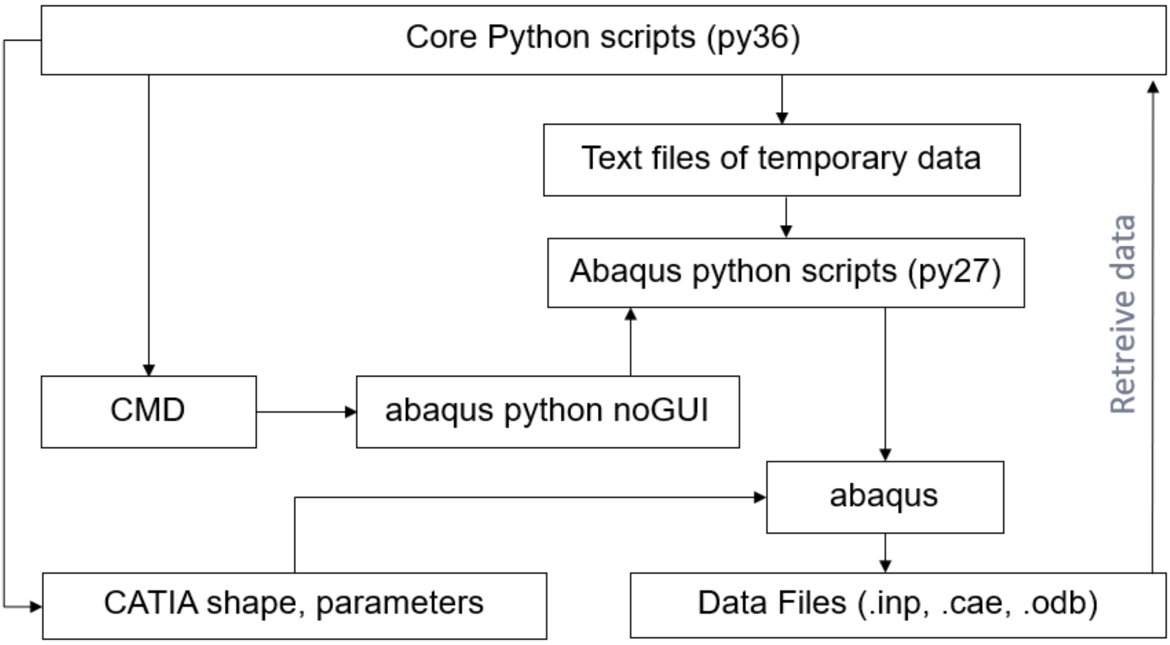


Figure 3.23 – logic of scripting Abaqus

A minor benefit of the command line running of these scripts is the potential for parallel processing. Since it is already run as a subprocess it would not take too much additional effort to assign this work to a specific workstation, while the other simulations can proceed for the next iteration of the optimisation. This could be done to make the scripts more time efficient, or to save up on licence requirements, as this way the Abaqus licence is used continuously. However, this was not deemed necessary for the work here due to the scale of the demonstrator project.

The complexity of having to run Abaqus scripts through command line is compensated by a very good documentation. The Python functions available in Abaqus are all reasonably well explained in the Abaqus documentation. The only difficulty is the terminology, knowing what different aspects of model generation are called in Abaqus greatly helps with interrogation of the documentation.

Recording capability is also available, similar to CATIA. Therefore, the simulation can be set-up manually to provide a backbone for the script. Of course, because of the complicated assignment of variables etc, significantly more additional scripting is required here than in case of CATIA.

The first part of the script deals with the import of data. Because of the asynchronous Python library usage, the data from SQL database cannot be imported directly from the Abaqus scripts. Therefore, the data is first translated from SQL to .txt files, before the internal Abaqus script is run. Then the data is imported from the text files at the start of the Abaqus script. The content in the text file is separated by designated string sections ("---"), allowing for easy isolation of variables using Python split function.

The Abaqus FE model was originally developed with the use of CATIA pre-mesh method. It was kept as an option. The script now identifies if the mesh came from "numimesh.py" option or from CATIA and creates elements accordingly. Programmatically the numimesh, that generates a mesh file at the end, is significantly simpler. 6 lines of difficult to understand code, were replaced by a single line.

Then the number of layers is assigned. In terms of the Abaqus work, it would be reasonably easy to add multiple layers with varying material properties. However, this would make the optimisation much more complicated due to the number of variables.

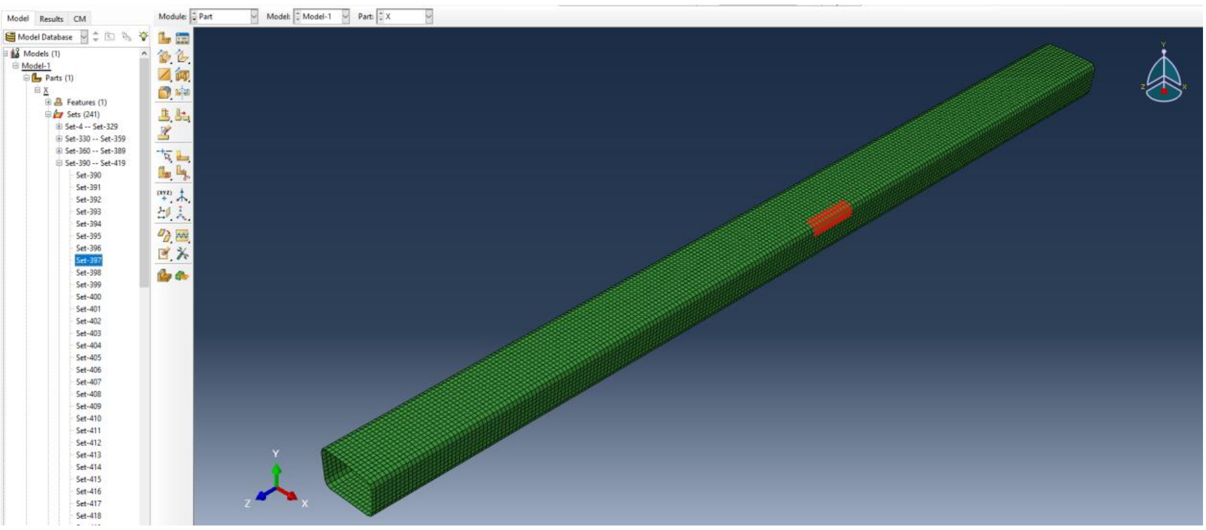
Boundary conditions for the demonstrator part are relatively simple, as these are applied at all nodes at z=0. The imported node list is simply iterated through to create a smaller list to apply boundary conditions on.

For the initial iteration, the forces were applied in similar manner, as a force applied at nodes at maximum distance from z =0. This will be re-addressed in chapter 5, where aerodynamic input is provided to this simulation.

To assign the material properties all the spanwise and cross-section wise segments are iterated through. Sub-loop than iterates through all elements available. Based on the boundary conditions paired with each segment, the elements are assigned to current segment. The assignment is done using "getByBoundingSphere" method, as it was found to be least error prone method, which can be guaranteed to assign the right element and no other. This is only possible because of the good control of the mesh achieved by the bespoke meshing algorithm, which dictates the size restrictions for an element. "getByBoundingBox" method was also tried, as in theory it should allow for assignment of all elements within a box area in one go. However, for multiple reasons this proved to be quite unreliable. "getClosest" and "getAt" functions were also tried, and were also deemed unsuitable for the application.

Once an element is assigned, it is deleted from the temporary element list. This way no element is assigned twice, and the list can easily be checked for any stray elements at the end. Once all elements belonging to a segment were identified, a set is created. Previously calculated materials are assigned to this set.

An example of assigned segment with specific material properties is shown in Figure 3.24.



Module: Part

Model: Model-1

Part: X

Figure 3.24 – example corner segment with assigned local material

At the end of the model setup, the mass of the component is exported. Then the job is started.

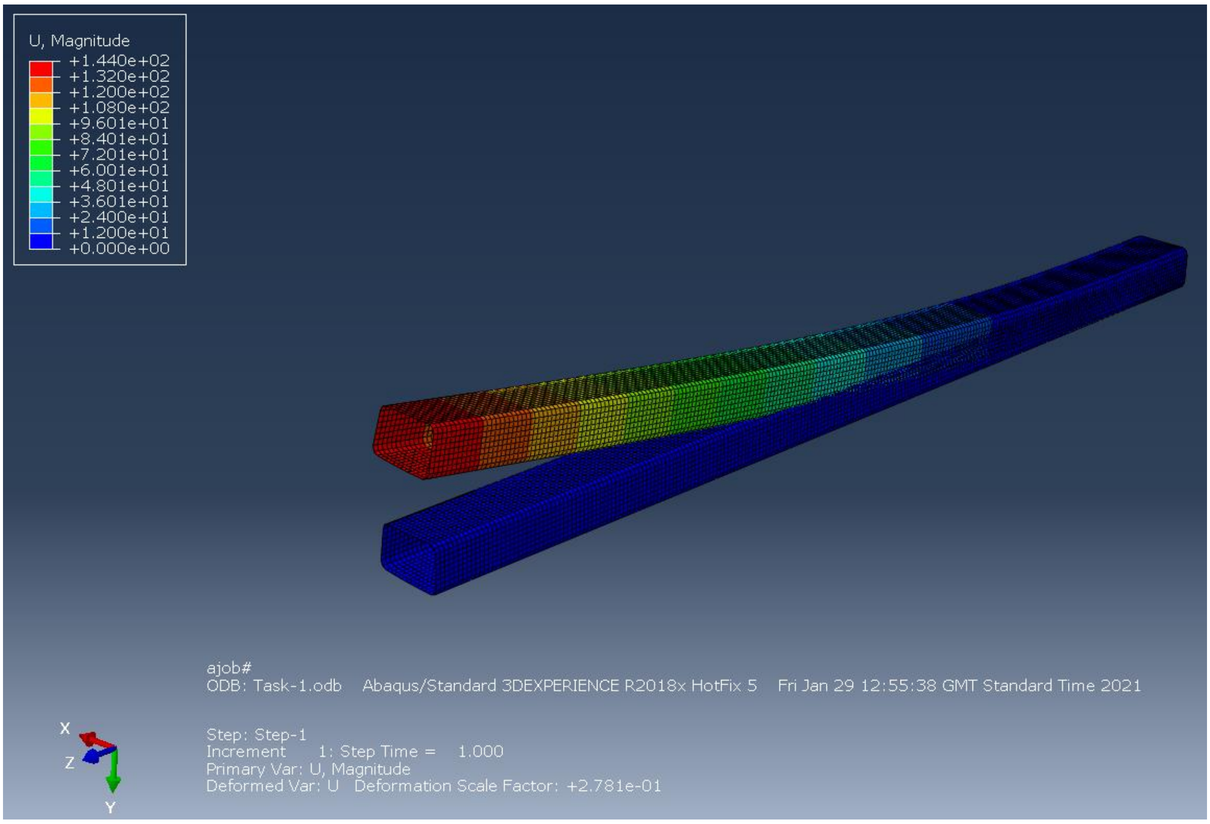
Initially, obtaining the mass using CATIA was attempted. However, materials cannot be imported into CATIA programmatically, or at least not without excessive difficulty.

A separate script is used for the post processing (abaqus_postProc.py). Here the documentation was less clear than with setting up the model, but was luckily well supplemented by existing discussions on various forums. Currently the main post processing consists of obtaining the deflection data and identifying the most deflected node. This maximum deflection is then used within the objective function to establish the quality of the current iteration of the part.

Other outputs, such as stresses, could easily be extracted. However, due to the lack of fidelity of the material models that feed into this simulation, this would likely be very crude estimate.

The runtime of the Abaqus simulation depends on the number of elements. The reasonably fine mesh shown in Figure 3.24 above takes 10-15 minutes to run.

During the development the Abaqus simulation was very prone to errors. The most common ones consisted of assigning elements multiple times, or not assigning an element. This is caused by Abaqus sometimes creating additional elements on top of those imported. The source of this error was not identified. However, all the errors are currently counteracted by various methods in the script, making sure that it runs, and everything is set as expected.



```
ajob#
ODB: Task-1.odb    Abaqus/Standard 3DEXPERIENCE R2018x HotFix 5    Fri Jan 29 12:55:38 GMT Standard Time 2021

Step: Step-1
Increment      1: Step Time =    1.000
Primary Var: U, Magnitude
Deformed Var: U   Deformation Scale Factor: +2.781e-01
```

Figure 3.25 – deflected part example

3.6.3 Standalone benefits

The standalone benefits of automating Abaqus are quite clear. Automated pre-processing of FE is quite commonly done to minimize the work put into running multiple simulations iterations. The automation of post-processing is mainly required to close the loop. Generally, structural analysis automation allows for structural optimisation and for the changes to the design to be easily assessed.

3.7 Summary of module developments

All the core modules needed to describe the geometry and mechanical performance of a braided composite part were successfully automated. The inputs and outputs of each module are well controlled, making the system highly modular. This was very useful for instance in replacing the meshing method, but it would likely be a useful feature even in industrial setting. It also allows for inclusion of any modules that become available, as will be shown in chapter 5.

The only programming language required is Python, making it quite possible to develop the modules, for engineers otherwise working on the simulations. It is difficult to assess the time it took to generate these scripts, but now that all the mechanisms are developed it would be relatively easy to create this system for new part time of similar difficulty. As a rule of thumb author estimates the automated simulation to take 3-4 times longer to develop than a conventional design of the same component.

Throughout the development the runtimes were continuously decreased. Now it is apparent that all the runtimes can be made almost negligible compared to the FE structural analysis.

The next section will discuss the iteration through the system and the various optimisation methods used.

References

1. Kucera J. Sysi [Internet]. 2020. Available from: https://github.com/Ellutze/sysi
2. Macro to assign materials, how? Community of experts of dassault systemes solutions. [Internet]. 2006. Available from: http://www.coe.org/p/fo/et/thread=14644
3. CATIA and SIMULIA Scripting and Automation with Python [Internet]. if curious: then learn. 2015. Available from: http://ifcuriousthenlearn.com/blog/2015/06/10/catia-and-simulia-scripting-and-automation-with-python/
4. Material Properties. Available from: http://catiadoc.free.fr/online/anlug_C2/anlugfempropmaterial.htm
5. Van Ravenhorst JH. DESIGN TOOLS FOR CIRCULAR OVERBRAIDING OF COMPLEX MANDRELS [Internet]. [cited 2019 Feb 21]. Available from: https://ris.utwente.nl/ws/portalfiles/portal/46645249/PhD_thesis_Johan_van_Ravenhorst.pdf
6. Kucera J. kiBraid [Internet]. GitHub. Available from: https://github.com/Ellutze/kiBraid

4. Data management and Optimisation

The more automated the design process becomes, the more it relies on a good data management. With large amount of collected and strategically stored data, the optimisation can become integral part of the design process. On top of obtaining the best combination of parameters, the large number of collected data can support any required changes to the specification.

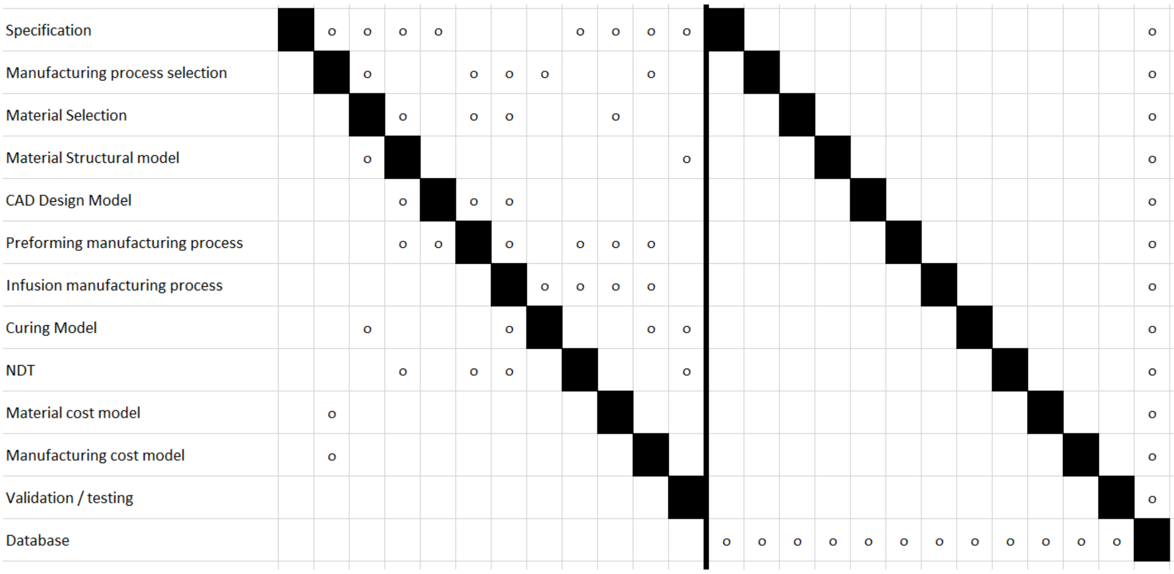
This chapter first outlines some of the data management aspects that are important for SySi. Then the iteration of the SySi is discussed; this includes discussion about the sampling method used and outlines the user interface, developed to facilitate the use of SySi. The main part of this section discusses the various optimisation methods used, their benefits, downsides, and potential improvements.

4.1 Data management – SQL

4.1.1 SQL data storage

The key to successful iterative improvements is good data collection and processing. Forgetting details of one design prevents learning for the next. Understanding how each variable involved in the design affects the resulting products can be of immense value. There are various ways to store data. However, in terms of structured data collection, databases stand out. A database forces a good standardisation of data and can prevent omission of important details. It also forces the user to think about how the variables interact, as such evaluation is important for good database design. Although data sharing between companies is currently not common, databases could facilitate potential data trading in the future. This could for example minimize the amount of material testing done in general or streamline design data flow through the product supply chain.

Storing data in a database also allows for better linking of different parts of design. Instead of developing methods of passing all the iteration data between all software, only one method for passing data to and from database is required. The benefit of this is visualized by the N^2 diagram shown in Figure 4.1 below. The “o” marks represent flow of information between different aspects/modules of component design. By passing all the information through the database, and tracking all the important information, the flow of data is significantly simplified.



Specification	■	o	o	o	o				o	o	o	o	■												o
Manufacturing process selection		■	o			o	o	o			o			■											o
Material Selection			■	o		o	o			o					■										o
Material Structural model			o	■								o				■									o
CAD Design Model				o	■	o	o										■								o
Preforming manufacturing process				o	o	■	o		o	o	o							■							o
Infusion manufacturing process							■	o	o	o	o								■						o
Curing Model			o				o	■			o	o								■					o
NDT				o		o	o		■			o									■				o
Material cost model		o								■												■			o
Manufacturing cost model		o									■												■		o
Validation / testing												■												■	o
Database													o	o	o	o	o	o	o	o	o	o	o	o	■

Figure 4.1 – data tracking between design modules, direct passing of information left, database used as continuous variable tracking tool right.

Even without generation of fully integrated simulation loop, large amount of benefit can be obtained from using a database. When any minor change is done to a component, a new structural analysis is likely required, then new part is typically re-analysed manually. However, if the initial analysis was done using parameters from the SQL, the next iteration can be run with minimal changes. This also allows other engineers in the design loop to understand which variables they can change, or suggest changing, without significantly affecting the part. Good practice now is that meetings are called about changes to parts, this is tedious and can result in difficult conversations about how much work is each party creating for the others. Systematically defined variables which anyone can change, without causing significant rework for others, could result in much more fluid collaboration.

Database storage of design details also makes it easier to review past designs and see how different variables affected the design. This can be useful for troubleshooting or simply as inspiration for development of new parts.

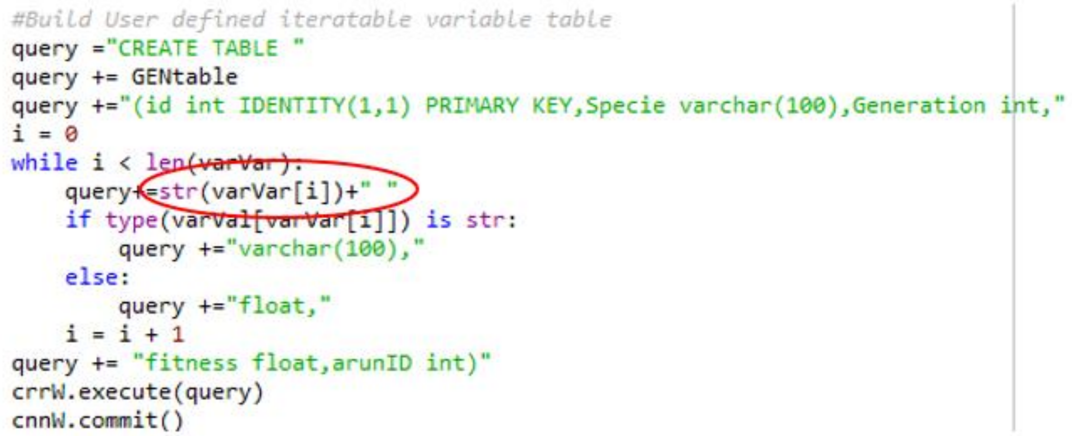
The main requirement for this system is to be easily interrogated by a scripting language so that it can be interfaced with the simulation modules. Due to the suspected amount of data extraction and input required the solution needs to be computationally efficient. MySQL software operated through Python scripts was identified as a solution to evaluate initially.

The development now lies in database architecture. Continuous development lies in facilitating transfer of data between all interfaces within the simulation system. The details of the database architecture, the system of tables, are omitted here. Good amount of information is available online on good database design practices, and every project is going to face slightly different issues.

In industrial settings it would be recommended to implement an automated data management system, which identifies old irrelevant data, or at given intervals interacts with a user to identify old irrelevant data. The growth of the database without these mechanisms can be quite enormous. This was not that much of an issue, as none of the data used in the demonstrator system have industrial value. However, presence of flawed or outdated information in the database could make it more difficult to isolate the truly useful data.

4.1.2 SQL within Python scripts

The SQL queries are used through pre-existent libraries in Python. The query is created as an assembly of multiple strings and then submitted to the database, see example in Figure 4.2.



```python
#Build User defined iteratable variable table
query ="CREATE TABLE "
query += GENtable
query +="(id int IDENTITY(1,1) PRIMARY KEY,Specie varchar(100),Generation int,"
i = 0
while i < len(varVar):
    query+=str(varVar[i])+" "
    if type(varVal[varVar[i]]) is str:
        query +="varchar(100),"
    else:
        query +="float,"
    i = i + 1
query += "fitness float,arunID int)"
crrW.execute(query)
cnnW.commit()
```

Figure 4.2 – example query built inside Python

This is only slightly more difficult than saving results in .npy or similar formats. Once each query type is used once, the same code can easily be reused with adjusted variables. However, the SQL storage greatly simplifies any future usage. The queries can be used to merge multiple datasets and to pick out specific information. This is commonly for constructing surrogates described in section 4.5.3.

The syntax of the queries is quite intuitive and therefore does not take too long to familiarize oneself with. Both MySQL and SQL-server were used at different stages of the project. In practice there is very little difference. Syntax differs slightly, requiring some script rework when transitioning between the two. Another database software could probably also be used, depending on the Python application programming interface (API).

4.1.3 Embedding SQL into a company process

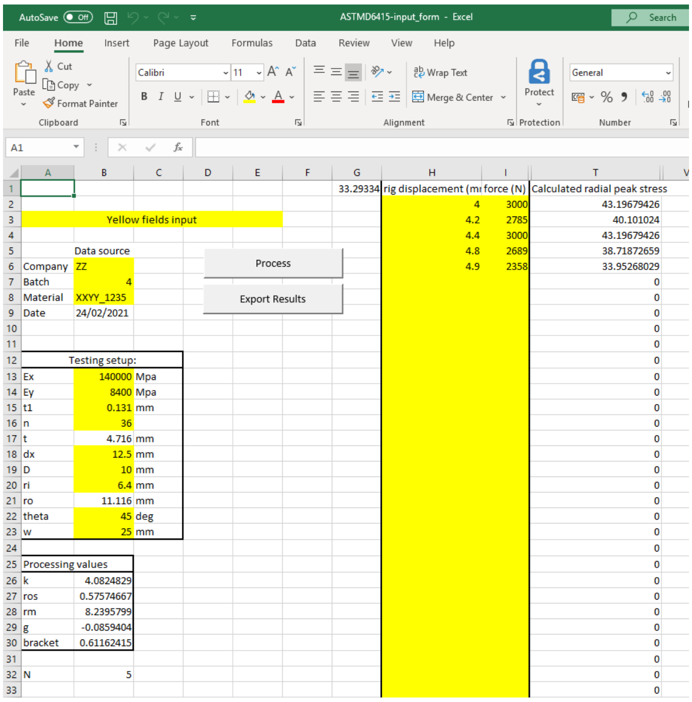
For SQL to be used within simulation loops such as this, it would be very convenient for the input data to also exist in the same storage space. The input data can include for example tested material properties or details about procurement of these. The more information is initially provided about all aspects of the design process, the more can be used to adjust the component through optimisation. Some information is difficult to store in database, such as specification. However, even specification can be stored in simplified version by the means of objective function definition.

Currently, management of data from testing is often uncontrolled. Material properties are usually tested for a specific project. This will likely result in an excel file, or even worse, an output pdf. Not only are these formats difficult to search for, but also the translation of data from there is often difficult. Generally, no consideration is given to the automation following the data storage.

Moving this data to SQL would require manual efforts that are difficult to assign to anyone within typical company structure. Data management is rarely a strong focus point, in best case all the testing data is kept in the same folder.

The testing is usually well standardised. SQL storage can create a good, standardised environment for maintaining data. Therefore, the only missing link is the act of storing the data. Expecting all engineers who collect data to know SQL would be too optimistic. However, the standardisation of the source, testing procedure, allows for generation of input forms which can be used by anybody.

Example set of input forms has been generated as a side-project by the author. The input forms are simple excel sheets with only few editable fields and automated file naming system. These can be used to collect key variables from various standard tests. The material properties are automatically calculated and stored within the SQL, the calculation and storage are done through Python script run at defined intervals. The Python script checks a folder with collected input forms and propagates the information wherever necessary, so that it can be later reviewed or used within simulations.



	A	B	C	G	H	I	T
1				33.29334	rig displacement (m	force (N)	Calculated radial peak stress
2					4	3000	43.19679426
3	Yellow fields input				4.2	2785	40.101024
4					4.4	3000	43.19679426
5		Data source			4.8	2689	38.71872659
6	Company	ZZ			4.9	2358	33.95268029
7	Batch	4					0
8	Material	XXYY_1235					0
9	Date	24/02/2021					0
10							0
11							0
12	Testing setup:						0
13	Ex	140000	Mpa				0
14	Ey	8400	Mpa				0
15	t1	0.131	mm				0
16	n	36					0
17	t	4.716	mm				0
18	dx	12.5	mm				0
19	D	10	mm				0
20	ri	6.4	mm				0
21	ro	11.116	mm				0
22	theta	45	deg				0
23	w	25	mm				0
24							0
25	Processing values						0
26	k	4.0824829					0
27	ros	0.57574667					0
28	rm	8.2395799					0
29	g	-0.0859404					0
30	bracket	0.61162415					0
31							0
32	N	5					0
33							0

Figure 4.3 -example form for ASTMD-6415 standard

Similar methodology is likely already employed in companies that specialize in testing materials or perform similar tests repeatedly. However, the sharing of this data with customer is most likely be through PDFs. Company that does its own tests might not repeat similar type of test multiple times. Therefore, generations of these forms might not necessarily be worth it. For this to become part of industrial standards, these forms would have to be available for all engineers and it would have to become standard industrial practice to use them. Then their creation for all ASTM and similar standards would be worth it. This could allow for merging multiple datasets for generation of surrogate models.

Sharing of data could be largely facilitated by the establishment of industrial standard input forms. The motivation for sharing data could be monetary, as many smaller companies would likely benefit from obtaining reliable, tested material data from some of the larger companies. The larger company would have another source of revenue, covering for some of the material testing expenditure. In short, with the greater standardisation the data could be used as a commodity, which is increasingly the case in various industries.

A robust and reasonable way of pricing different types of data would need to be developed, based on its source and other considerations. However, these are likely minor hurdles; as long as there is monetary incentive for sharing data and the costs of data purchase are lower than that of re-testing.

The sharing would in turn increase the amount of data available to any single company, minimizing the issues caused by uncertainty in composite manufacturing. For critical simulations a stochastic analysis can be done to assess the effect of the variation in the input data. This would go well with the SySi; for the specific analysis smaller iteration loop can be established to feed in the variable data, assessing the effect of the variation on the results. This can then be used in the main optimisation loop to disqualify combinations of parameters that have high likelihood of becoming unsatisfactory solutions due to the variation in manufacturing.

In summary this sub-project shows a potential way of standardising the material inputs that are integral part of projects relying on automated simulation.

4.2 Git management – script version control

Script management system is required for this project. Git was selected, with GitHub as the cloud storage.

This has been implemented at later stages of the project, when multiple different branches were tested. Different versions of each simulation, 2 database options and iterative improvements emphasized the requirement for version control.

Git is very good at tracking changes, therefore version before implementing certain problematic feature can be found and loaded. Even more importantly, the branching available on the platform is very useful. Branching allows for temporary work on an aspect of the program, which might not turn out to be useful or working. Therefore, without impacting the functioning system, new versions can be easily trialled. Pulling and pushing specific branches also allows for working on multiple versions on one workstation, without cross-contamination.

The GitHub is also good for sharing. The main GitHub page for SySi is made public. This allows people to have a look at the construction of scripts. They can also try the demonstrator, but due to the scripts being designed for a specific part this is unlikely to be used. The scripts are mainly to be used as an inspiration for similar projects.

The scripts are annotated to make it easier to take out any pieces other engineers might find useful. The guidance on demonstrator installation is also available. The downside of the GitHub page is the additional requirement to keep track of the documentation, making sure that the instructions still work with new versions.

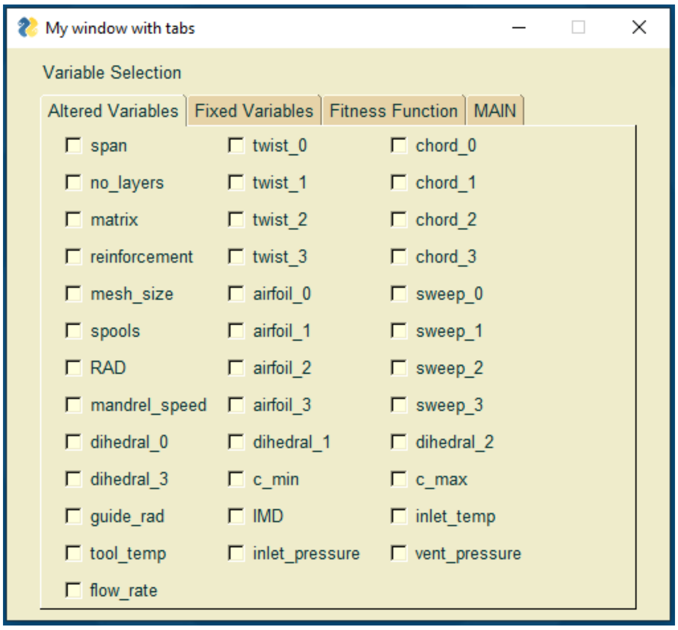
The usage of SQL makes the sharing of demonstrator more difficult. An additional script that generates the default tables is also provided. However, a database needs to be set-up ahead of time and the database details must be imputed into the script.

Main project: https://github.com/Ellutze/sysi

Braiding isolated: https://github.com/Ellutze/kiBraid

4.3 Iteration – user interface

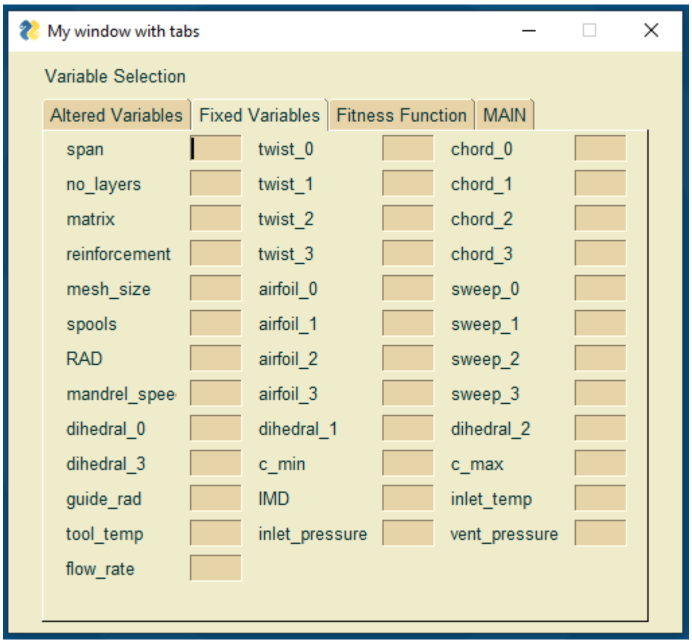
User interface was created using the Python PySimpleGUI library. First tab, in Figure 4.4 shows the variables that can iterated in the default version of SySi. Any combination of these variables can be selected.



☐ span	☐ twist_0	☐ chord_0
☐ no_layers	☐ twist_1	☐ chord_1
☐ matrix	☐ twist_2	☐ chord_2
☐ reinforcement	☐ twist_3	☐ chord_3
☐ mesh_size	☐ airfoil_0	☐ sweep_0
☐ spools	☐ airfoil_1	☐ sweep_1
☐ RAD	☐ airfoil_2	☐ sweep_2
☐ mandrel_speed	☐ airfoil_3	☐ sweep_3
☐ dihedral_0	☐ dihedral_1	☐ dihedral_2
☐ dihedral_3	☐ c_min	☐ c_max
☐ guide_rad	☐ IMD	☐ inlet_temp
☐ tool_temp	☐ inlet_pressure	☐ vent_pressure
☐ flow_rate		

Figure 4.4 – iterated variables tab GUI

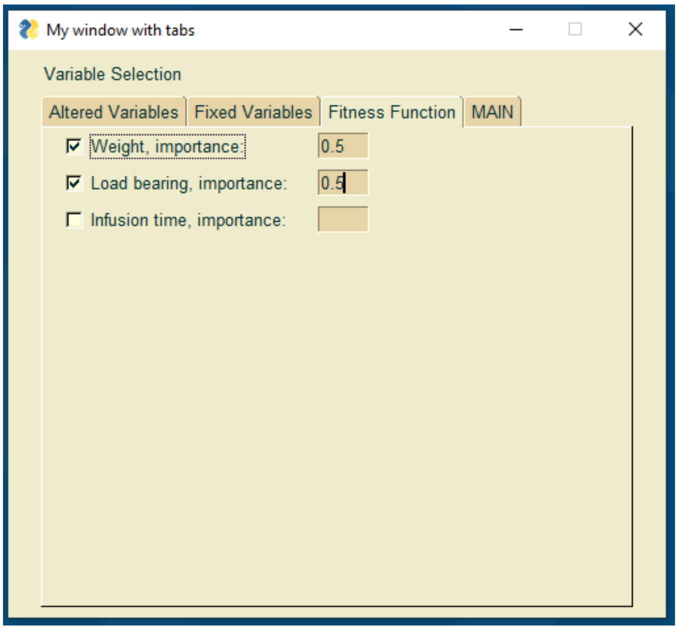
Second tab allows for specifying a value for any of the none-iterated variables, Figure 4.5.



span		twist_0		chord_0	
no_layers		twist_1		chord_1	
matrix		twist_2		chord_2	
reinforcement		twist_3		chord_3	
mesh_size		airfoil_0		sweep_0	
spools		airfoil_1		sweep_1	
RAD		airfoil_2		sweep_2	
mandrel_spee		airfoil_3		sweep_3	
dihedral_0		dihedral_1		dihedral_2	
dihedral_3		c_min		c_max	
guide_rad		IMD		inlet_temp	
tool_temp		inlet_pressure		vent_pressure	
flow_rate					

Figure 4.5 – fixed variables with none-default value

The user interface was also designed to adjust the objective function. However, without a specification of the aircraft it is difficult to make a realistic objective function composed of the available variables. Therefore, this functionality was not really used, as will be clear from the objective function defined in section 4.4.1.



☑ Weight, importance: 0.5
☑ Load bearing, importance: 0.5
☐ Infusion time, importance:

Figure 4.6 – objective function variation

The fourth tab is perhaps the most useful. Originally it was meant to select from various optimisation methods. However, since most runs end up starting with Latin hypercube sample it is mostly used just to select which run is to be continued. When a run is interrupted or an error occurs it can be continued here, without creating new sample.

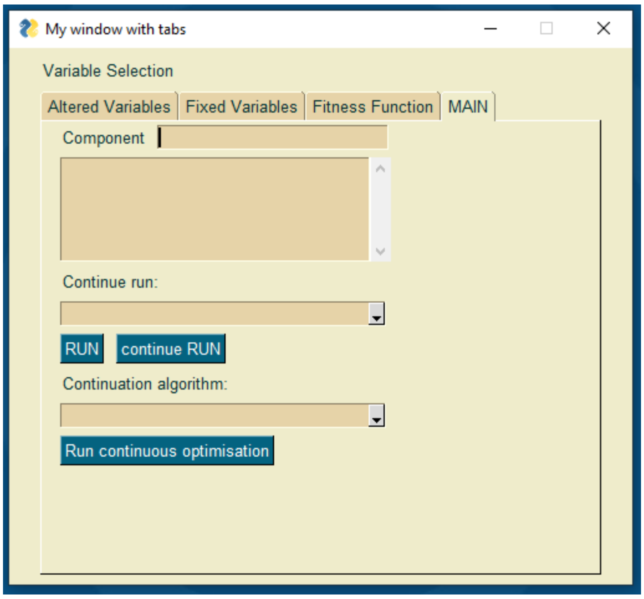


Figure 4.7 – selecting run to continue or to initiate

In summary, the GUI makes the running and troubleshooting easier and allows for more visual demonstration of the system.

4.4 Sampling design space

Sampling the data is a crucial aspect of this project for 3 main reasons: understanding design space, troubleshooting, and feeding surrogate models.

The design space can be better understood with plots of variables against each other. This can reveal that some variables have a lesser effect than others or can point at previously unacknowledged interactions of variables.

Troubleshooting is a crucial aspect of any complex project that includes some amount of newly written code. A good sampling method allows for running various combinations of parameters, some of which will inevitably cause errors. This allows for isolating the problem at hand and making the code more robust.

Any surrogate model requires a sample as an input. This will be further discussed in optimisation section below.

The sampling method selected here is the Latin Hypercube Sampling (LHS). A few versions were tried, including a bespoke scripted one. The one that proved to be most robust, and very fast was the one provided by Moza (1) on GitHub, which is based on paper by Deutsch (2).

Figure 4.8 shows an example of sampling 5 variables. The example variables are number of layers, mandrel speed, and 3 variables defining the scaling of airfoil at 3 different locations.

The figure only shows the uniform, but random, distribution of variables. The reason why the LHS is later only used for 3 or less variables, is that any following optimisations with higher number of variables become too computationally expensive.

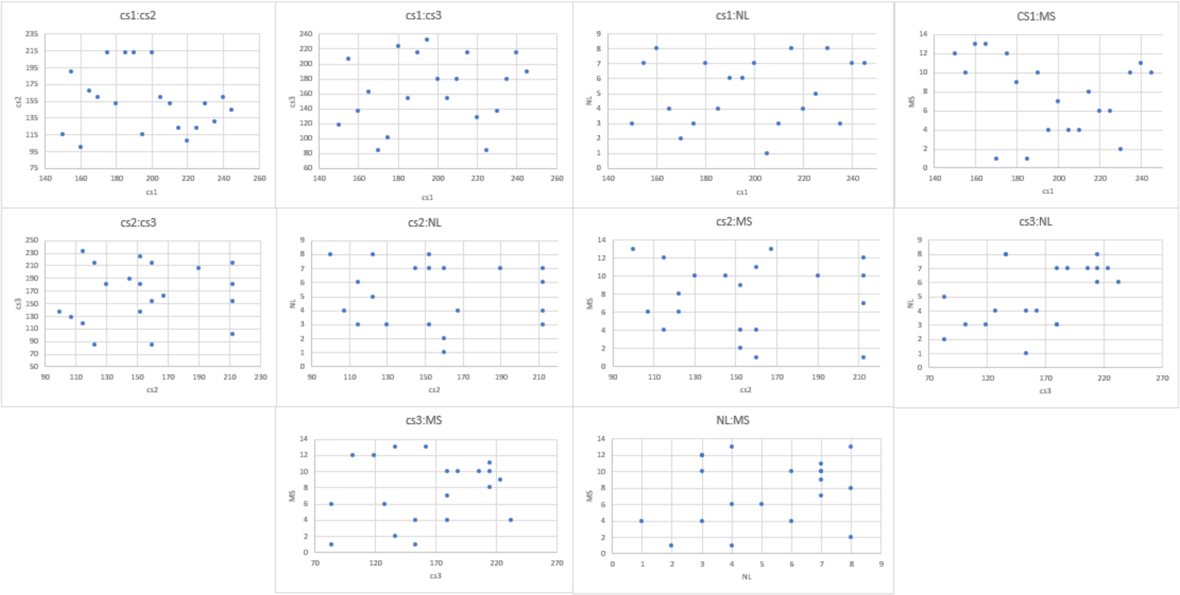


Figure 4.8 – example sampled design space with five iterated variables

The LHS sample, such as above, might be slightly too complex for optimisations. However, it can still be very informative, to run a wide LHS. For example, from the results it will be very clear which variables had negligible effect compared to the rest. The rather negligible variable can then be set to a reasonable value, removing it from the optimisation.

4.4.1 Objective function

Each iteration of the part is evaluated using an objective function. The objective function should correspond to the specification for the part. For most iterations in this study a simple two-part objective function was used.

The function is defined so that in ideal case f≅1, where half of the function comes from minimizing weight and the other half from minimizing deflection. In the function below the weight in metric tons and deflection in mm are used.

$$f = 0.5 * (0.5^{w*20000}) + 0.5 * (0.5^{d/10})$$

f=fitness function ;   w=part weight ; d=deflection under pre-defined load

In industry, failure modes would likely be more important than deflection. However, since the part does not reflect a real specification, this objective function only serves to combine more than one optimisation objective. Deflection is typically only of concern for aerodynamic designers, but failure mechanisms are currently very difficult to implement and are considered beyond the scope of this work.

Initially the objective function was supposed to have a 3rd component, infusion time or infusion probability. This was simply planned to demonstrate an objective function that is composed of aspects of multiple simulations, highlighting the benefits of SySi. However, the infusion of the simple part proved to be relatively easy, the part always infused with minimal changes to the infusion time. Therefore, adding this component to the objective function would be meaningless, without artificially making the infusion difficult by adjusting the inputs.

The objective function is quite simple, but it still requires all aspects of the system. The structural analysis that outputs deflection is heavily dependent on manufacturing simulation outputs. Real objective functions would likely be more complex, but this needs to come from the component specification. The benefit of an industrial example part is discussed in future work section.

4.4.2 Notable parameter combinations

Although the part is quite simple, some interesting parameter combinations and their effects have been observed.

The focus in this thesis is on the mechanisms of the optimisations and design rather than on the assessment of the specific part. Therefore, the following graphs are not all from the same version of the system of simulations, so there are differences in minor aspects such as volume fraction calculations.

The plots do not display the absolute values, but rather percentage of available design space. For example, 0 number of layers corresponds to the minimum 1 layer, and value of 1 displayed on graph corresponds to the maximum number of layers allowed, 15 in case of Figure 4.9.

The three parameters shown in Figure 4.9 were used for several optimisation runs, as all three have a considerable impact on the fitness function. From the Figure 4.9 mandrel speed has no apparent direct influence on fitness function. However, this is only the case because the other 2 variables have more significant impact. When multiple variables are iterated together, the effect of one variable might not be visible.

To show the effect of mandrel speed, a new run was executed, only varying mandrel speed. This is plotted in Figure 4.10.

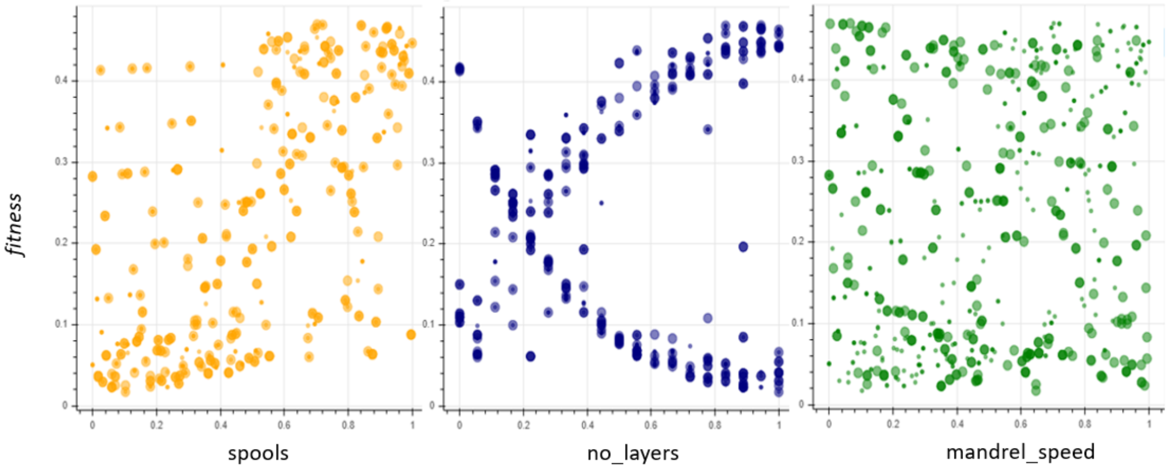


Figure 4.9 – LHS with mandrel speed, no. of spools, and no. of layers variables

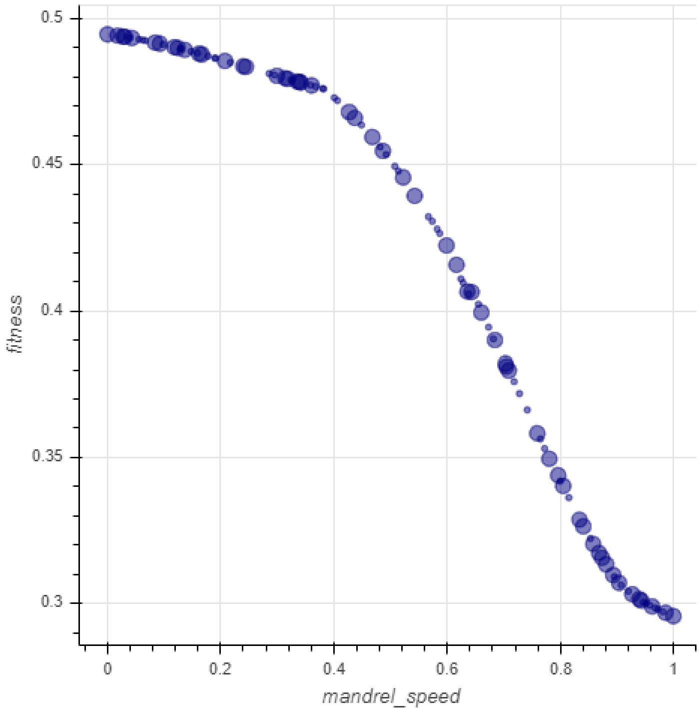


Figure 4.10 – mandrel speed impact on fitness

Clearly, mandrel speed has a significant effect on the fitness value, when it is not obscured by the effects of other variables. Higher mandrel speed translates to lower braid angle. The main cause of decreasing fitness function with increasing mandrel speed is the volume fraction. Because the number of spools is kept constant, the fibres don't cover the mandrel fully at low braid angles. This effect would be significantly less pronounced if nesting was accounted for. Alternatively, allowing the number of spools to vary, makes sure that coverage is possible even at low braid angles. This is part of the reason why the Figure 4.9 suggests mandrel speed has minimal effect; the spool variations can obscure the Vf effects caused by mandrel speed. The Figure 4.10 also shows lower absolute gradient at low mandrel speed values. This is because the effect of fibre alignment becomes more pronounced. At very high braid angles the fibres lose a significant amount of their load bearing capacity.

Returning to Figure 4.9, there is a general tendency for higher fitness values to correspond to higher number of spools. But the most interesting observation is the relationship between number of layers and fitness. On the same graph two separate effects can be observed, which depend on the setup of the other two iterated parameters. When the other parameters are set well, the number of layers increases the fitness value. This is because the additional weight is efficiently compensated by the decreased deflection. This appears to have diminishing returns as expected. However, when the other two parameters are set badly, the number of layers only decreases the fitness value. In this case the weight is increased without decreasing deflection sufficiently. This is most likely the case when the number of spools and mandrel speed result in low volume fractions.

The difference between the two trends is so significant that the best fitness values are obtained either at the minimum 1 layer, or above 8 layers.

These data were collected with first iteration of volume fraction (V_f) calculation, which was non-linear. The original V_f calculation was gated using if function. If the spools fit side by side the V_f was calculated based on geometry of the cross sections. When the spools number would result in clashes of fibres the calculation of V_f is capped, therefore the calculation is difficult to describe by a single mathematical function. Hence it is difficult for a surrogate model to follow these trends. This will further be described in surrogate modelling section, where another V_f prediction method is tested.

The combination of parameters shown above is also used to test out surrogate models.

Similar dual-effects of number of layers on the fitness value is observed with other parameters, given the other parameters don’t have significantly stronger influence. In Figure 4.11, mandrel speed has been replaced by spar radius.

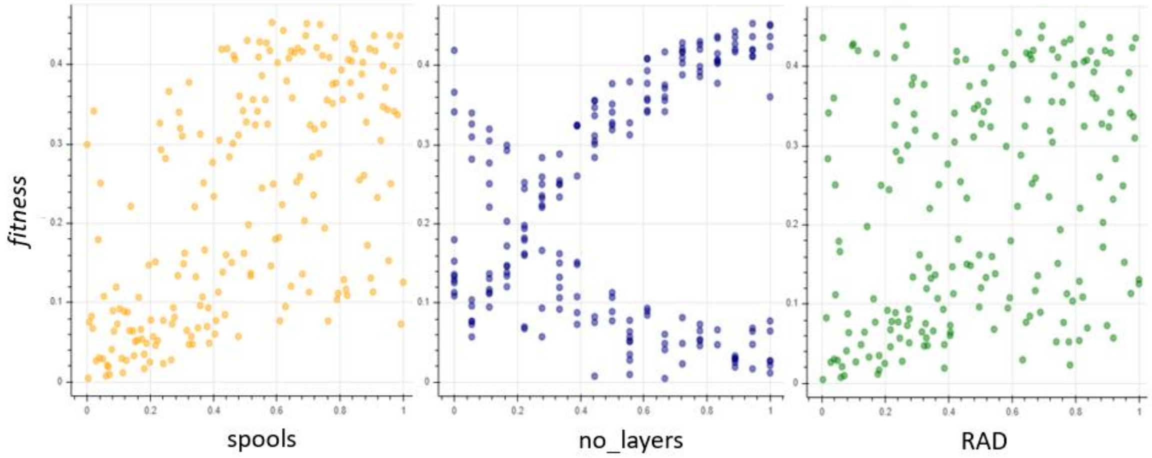


Figure 4.11 - LHS with corner radius, no. of spools, and no. of layers variables

Some of the variables originally considered for iteration cause problems when iterated. The size of a cross-section of a spar is defined mainly by the chord length. If more than one of the chord length positions is iterated large negative and positive tapers can be created. When this happens the braiding, simulation tends to run into errors. The major issues are related to the propagation mechanism employed in the simulation. If large expanding/negative taper is used the fell point is always over horizon from perspective of the spool location, leading to repeated fell point propagation in one direction. The reverse happens with large positive taper where the horizon is never reached, as most of the spool rotation circle is in line of sight of the fell point. This can be solved to some extend by adjusting the guide ring radius variable. This comes with two issues.

Firstly, an automated adjustment algorithm would have to be designed to accommodate for variety of parameter variations, requiring very high level of understanding of all the possible errors.

Secondly, a problem with adjusting the guide ring is that in real manufacturing the excessive tapers would likely not cause this type of problem. Accommodating for this would be solving a simulation related problem, rather than improving how close the simulation is to physical manufacturing. Also, negative taper is very unlikely to be useful for any kind of spar. Therefore, to include multiple varying chord values should be accommodated for by linking these variables together, making sure subsequent chord values are always smaller. In this way likely scenarios of part in question are covered, while the limitations of the braiding simulation do not cause any major issues.

The question then becomes: when is this adjustment done? There are two main options. Either the variable is adjusted within the Latin Hypercube Sample, or the chord values are decreased when necessary after the sampling. The first case requires additional coding within the Latin Hypercube Sampling. The latter option makes optimisations and other analysis of the results more difficult as the adjustments of subsequent chord values have to be taken into account. In case of both options, surrogate modelling would be difficult as the design space will not be sampled in completely uniform manner.

Retrospectively, these problems are the cause of bad specification of variables. The assumption should have been that only positive tapers exist, where the chord is smallest at the tip. For example, the variables could instead be defined as: the initial chord, location of the tapering, and the end chord. This would account for the majority of standard taper setups. More complex setups, such as varying degree of taper in various sections would not be possible. However, although this is possible with the setup currently used, due to the above described issues, these variables are not very well suitable for optimisations. It is a difficult decision, as simplifying the variables would also prevent variation of the airfoil along the span, which can be very useful in many aerospace applications. Any complexity makes mathematical operations, optimisations and surrogate models more problematic, but also allow for better designs. This should be taken as learning for future projects.

Other variables that could have been designed better were the location of spar front and aft wall relative to chord. Instead of two location values, as percentage of chord, only the front wall should have been defined. The second variable would be better defined as length of the spar in direction of the chord. This way the second variable would be more suitable for optimisations. The spar length would have more consistent effect than location of aft wall, which is heavily dependent on the location of front wall. The design of the two variables was improved by if clauses preventing unsuitable scenarios.

Changing these variables is quite difficult once the system of simulations is running as it would require changes in every module developed so far.

The initial mandrel distance (IMD) is a variable that mainly has effect on the run-in distance of the braiding, the distance it takes for the braid to stabilise. Too short IMD is shown in Figure 4.12, where no parameter is changed during the braid, but the braid clearly takes majority of the part to stabilise. When the IMD is set close to the stabilised distance from fell point, almost no run-in distance is required. This would be the most realistic setup as the run-in would in practice be covered by extended mandrel. This is how CATIA CBX solves stabilisation, by requiring user to provide extended mandrel.

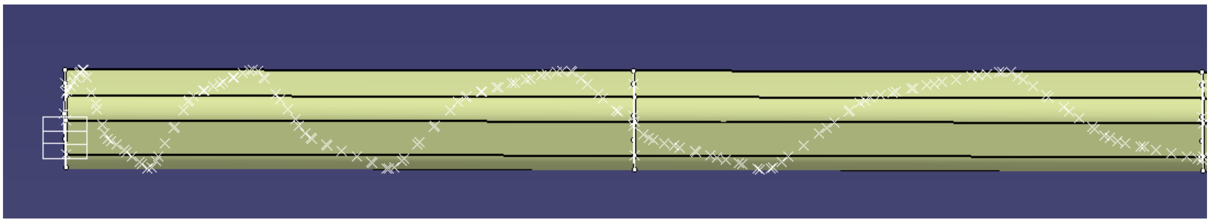


Figure 4.12 – badly selected IMD parameter

Another variable that had to be restricted is the corner radius. The corner radius must be maximum half the thickness of airfoil. This might not always be the case with the standard boundary conditions, as taper or change of airfoil can cause the thickness of spar to be quite small. In such cases an if function is used to prevent radius from being too small. Of course, this again poses the danger of unevenly sampled design space. Secondary issue with this is the manual review of results is more problematic, as it would unnecessarily complicate the variable storage process, if it would account for the practical automatic adjustments of variables. However, this occurs in only small number of cases.

In general shape changes are more prone to causing errors than other variables. These are hard to predict, especially because all the different combinations of variables available. Most commonly the errors stem from the braiding simulation. It is developed from scratch and hence the only troubleshooting and testing it had was within the SySi development. The Abaqus simulation is the second most likely, but most of these are prevented in the latest version of SySi by the bespoke meshing algorithm.

These are only the most notable issues caused by parameter interaction. On one hand this outlines the difficulty of setting up an optimiser system such as this one, which is simple compared to real industrial problems. On the other hand, the troubleshooting of the issues caused by these variables helps understanding the interaction of variables, which is probably more important benefit in complex parts.

4.5 Optimisations

The ultimate goal is to be able to optimise for any parameters within the design loop, while varying any number of selected parameters. The initial investigation here focuses on implementation of few common optimisation methods to outline the benefits of the integrated closed loop. Initially only few parameters were selected to make the initial implementation simple, but same steps presented here could be taken if the number of parameters used is increased. The main obstacle that comes with large number of variables is the computational expense.

4.5.1 Optimisation methods

Variety of available optimisation methods and libraries is large. Multiple were tested to establish which ones are most suitable for SySi, the most notable ones are outlined in this section. The main requirement is that the optimisation needs to be able to work with multiple parameters with varying impact on the resulting fitness function. Ideally the optimisation should be suitable for both discrete and continuous variables, but this proved to be very difficult, and hence was considered a soft requirement. The optimisation should also include a mechanism for escaping local minima, as the topology of design space can be quite complex for SySi and similar systems. The convergence should be as fast as possible, due to the rather large overall simulation runtime.

4.5.1.2 Ant colony inspired algorithm

Standard ant colony optimisation, ACO, mimics the ant search for food. Where food is present the ants release pheromones to persuade ants to take that route. Algorithm inspired by ant colony optimisation is implemented here. First all points from LHS strategy are evaluated using a fitness function. This is only to create the initial datapoints, 10-20 were considered reasonable. New datapoint in the optimisation is created by first randomising every iterated variable value, within the boundary conditions. Each of the randomly generated variables is then adjusted by previous top results. Each variable value is then adjusted by the top 10% results, the adjustment scales with the fitness value. The adjustment function is as follows:

$$X = R + \sum_{i=1}^{n} (T_i - R) * \left(1 - \frac{i}{n}\right)^2 * 0.2$$

Where X is k dimensional vector; k being the number of iterated variables. R is the randomly selected k dimensional vector. T_i is the top results of previous runs. $i = 0$, corresponds to the top result, $i = n$ correspond to the last result in top 10%. The 0.2 value is used as a convergence adjustment. Higher convergent adjustment would allow the previous results to have large effect on the new guess and vice versa.

However, the function still requires tweaking as too strong adjustment is likely to lead to local minima, rather than global minima. Conversely, too small adjustments will result in slow convergence of the optimisation.

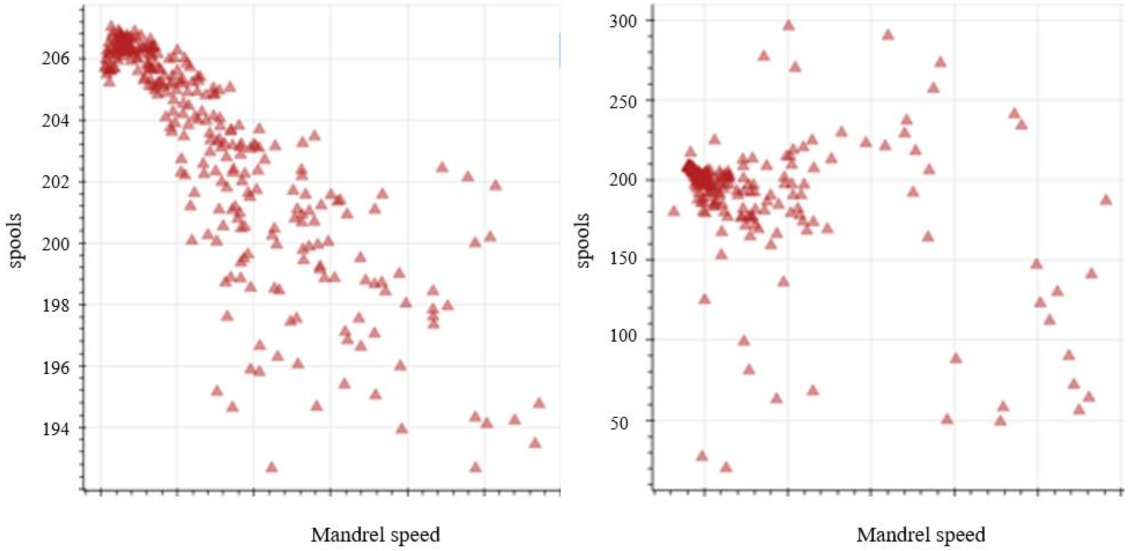


Figure 4.13 – ant colony optimisation trial

Figure 4.13 shows this algorithm working with 3 variables: mandrel speed, number of spools and number of layers. The data has been plotted to show the convergence of the optimisation. 2D plot was used for representation, similar convergence can be observed with the other 2 variable combinations. First run of this optimisation, left graph, shown minimum at the edge of allowed design space. Therefore, it was suspected that the boundary conditions could affect the result. Therefore, rerun with extended boundary conditions was done. This re-run is shown on the right, confirming the same minimum. The plot of the left graph is obsolete since the new dataset was generated, but it was included to show the danger of setting up too narrow boundary conditions.

This is quite a simple example, but it shows a good convergence to a global minimum. However, it still took about 70-100 runs to achieve convincing convergence, resulting in a very long runtime that would increase when the number of variables is increased. Also, this does not include previous runs that were required to tweak the convergence parameters.

Therefore, this method might be suitable with a large amount of computing capacity, but it is not very practical for increased number of variables. In runs with 5 or more variables convergence was not achieved in first 400 runs.

This optimisation only works with continuous variables. Therefore, variables such as airfoil, must be selected separately.

Due to the simplicity of the component and time required for each run, the ability of this algorithm to escape local minima was not verified.

Because of the pull mechanism of top results certain minima might be unfairly prioritized against others, if boundary conditions are too limiting. If a minimum is close to boundary condition it will be hard to achieve, as all the results are pulled away from it, only based on the LHS initial datapoint distribution. The minima in the middle of design space will have initially more pull than edge minima, therefore it is important the simulation runs until only one minimum remains. When analysing the results of this optimisations, the possibility of this occurring should be considered.

4.5.2 Genetic algorithm

Genetic algorithms (GAs) are very popular. These are mainly useful where the user cannot make a reliable guess of where the global minima is, as the algorithm should be relatively good at getting out of a local minima. The usefulness of such algorithm would be more pronounced when more variables are involved, and the intuition of the analysis becomes less reliable. However, general problem of GAs is that they require many iterations to reach the final result. This increases with the number of variables.

Also, most of the variables in SySi can change continuously, rather than by discrete steps, which requires the GA to have finely tweaked mutation algorithm. When the optimum is found using discretised variables, some sort of additional optimisation would be required to find the best values of the continuous variables.

A GA script was created for testing. The script first creates a random initial population within design space. Then all members of the population are evaluated using a fitness function. The highest rated individuals are crossed, each offspring taking each gene/variable value from a random of the two assigned parents. Mutation is then implemented, changing a random value in the population. The percentage rate of mutation is crucial for successful GA. Too frequent mutation could prevent results from reaching any minimum, while too low mutation rate would prevent the optimisation from escaping local minima.

As expected, with the iteration runtime of 20 minutes, no useful results were obtained using the GA as the number of iterations required far exceeds the computational capacities available. The iteration runtime was significantly decreased since the time of testing but it is deemed unlikely that the decrease was sufficient for implementation of GA.

However, the general GA script is working and it might be possible to use it later in combination with other optimisation methods.

For instance, the GA could employ several local gradient optimisations which would adjust values within individual modules within the system. This would lend itself nicely for parallel processing. However, at the moment this is considered as potential future development.

Alternatively, an optimisation with multiple steps could also utilise GA. If both discrete and continuous variables are involved, two separate optimisations can be run. First, all continuous variables are kept fixed at reasonable values. The discrete variables are then optimised using GAs, hopefully with decreased number of iterations required. Once reasonable values have been obtained for discrete variables, another optimisation type is used for continuous variables, while keeping the discrete constant. This optimisation combination would not account very well for the interaction of the discrete and continuous variables. Niche ideas such as this would have to be a good fit with the specification of the component in question, and would have to be evaluated for the particular problem.

4.5.1.4 Other standard optimisations

Differential evolution and gradient optimisations were tested, but will be discussed further in surrogate modelling section, as these were found to be more useful when working with a surrogate model.

Other optimisation methods were also briefly considered, but insufficient number of trials was run for it to lead to meaningful conclusions, other than not being particularly suitable for SySi.

4.5.3 Surrogate modelling

The LHS sampling was already quite useful in understanding the design space and troubleshooting the SySi. The optimisation methods tried before the surrogates were mainly limited by runtime related issues. Therefore, surrogate modelling is very fitting tool, the computationally expensive runs are already provided from the LHS and the optimisations can then be done in an efficient way.

The relation between simulation, surrogate model and the optimisation is visualized in Figure 4.14.

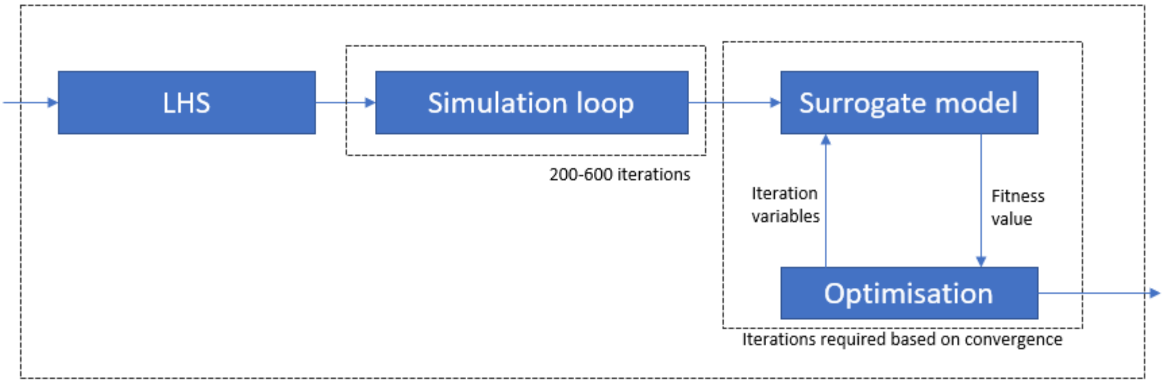


Figure 4.14 – surrogate model enabling high iteration optimisation

A popular machine learning toolkit scikit-learn was used. It is a Python library with variety of surrogate modelling tools. The map in Figure 4.15 outlines all the different options the library provides along with a decision tree helping with the method selection process. pyKriging library was also considered but the sci-kit learn proved easier to use.

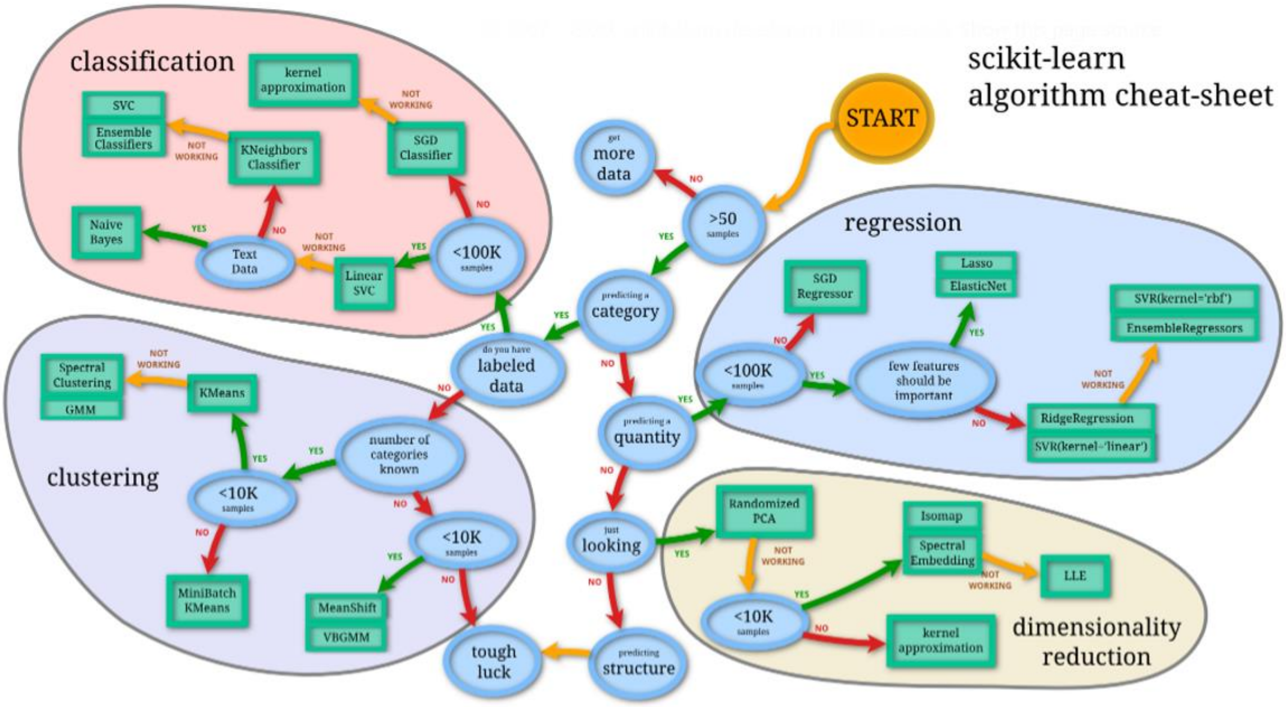


Figure 4.15 – Scikit library with method selection tree (3)

Based on this map and additional research, the Support Vector Regression (SVR) was deemed most suitable to construct the surrogate model. Multiple kernels were used, including most notably radial basis function (RBF), linear, and polynomial functions. The data presented in LHS section were initially used, these included 3 separate LHS samples of 200 runs each, amounting to 600 datapoints.

These datapoints were randomly split into teaching and validation data, this is shown in Figure 4.16 below.

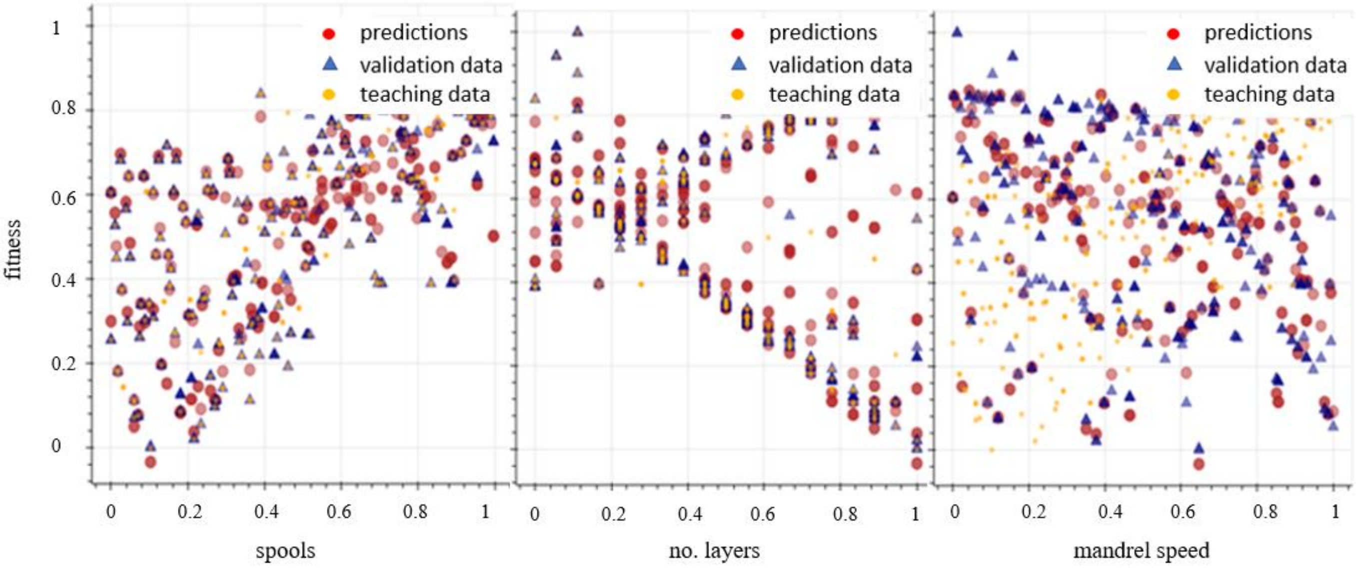


Figure 4.16 – dataset split and validation calculations

It is clear that there is some noise in terms of how the predictions from the generated surrogate model compare to the validation dataset. To better visualize this, the design space was plotted with errors displayed. In the plotted version the SVR uses 7th order polynomial kernel.

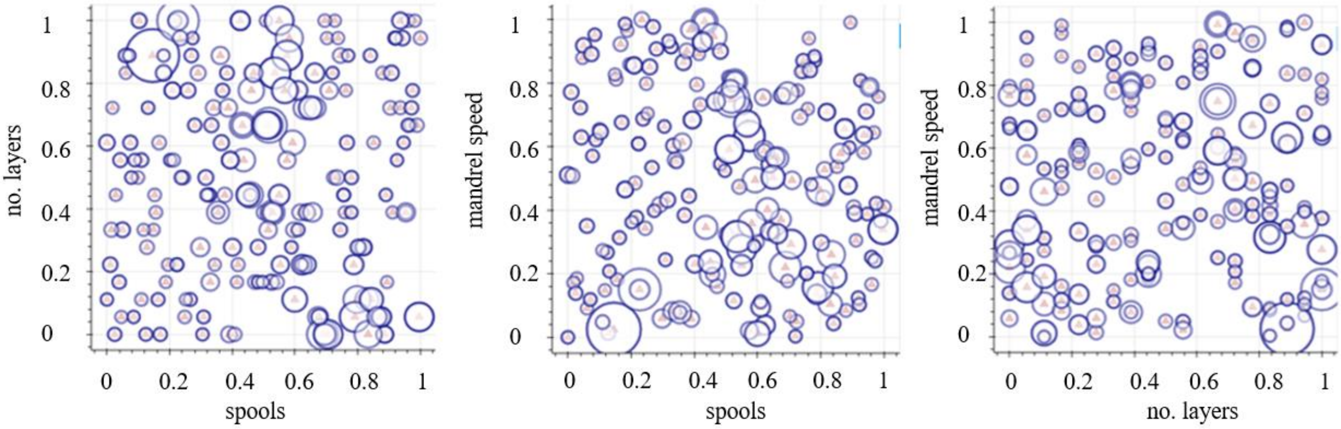


Figure 4.17 – visualized errors of the surrogate model

The circle size denotes the error between validation data and the predicted value for the same set of variables. The circles are not to scale, they are to denote where the largest errors are. The prediction success is then measured in average percentage error, which is 6% here. It is clear that the errors occur mostly in specific areas in the design space. Other kernels are shown in table below for comparison. In terms of localization of high error regions, different kernels were quite similar. This suggests that the error is caused by complexity of the generated data, rather than insufficiency of the kernels.

Table 4.1 – notable SVR runs

Kernel	Degree	Datapoints	Av. error
RBF	N/A	600	0.041
Linear	N/A	600	0.124
Poly	2	600	0.084
Poly	3	600	0.0665
Poly	4	600	0.0696
Poly	5	600	0.0639
Poly	6	600	0.0564
Poly	7	600	0.0593
Poly	8	600	0.0589

Other parameters such as the regularization parameter (C) or training loss parameter (ε) were kept constant at values which tended to yield good results: C = 20, $\varepsilon = 1 * 10^{-7}$. The values shown on the table can vary quite a bit. The average error displayed is the average error of 3 separate runs. The additional randomness comes from splitting of the dataset.

The RBF proved the best in terms of error, at 4%. This shows that approximate prediction is usually possible, but the errors are still too large to completely rely on, when selecting new design parameters. The Figure 4.17 hints at the possible sources of error.

The split of data into teaching and validation data is done by a function that randomizes the order of datapoints and splits the data in half. Theoretically better surrogate should be created if two separate LHS datasets are used, as both will have good design space variable distribution. This was tested using the 2 out of the 3 LHS sets used for the above table. The 400 datapoints RBF yielded average error of 4.5% when randomly shuffled, this was an average of 3 shuffle runs. The error was 4.15% when separate LHS run was used for teaching and validation data. Therefore, there is a small difference. This suggests that if data is generated using multiple surrogates it is better to split into the teaching and validation data by the source LHS run. This difference is likely to be decreased with larger number of datapoints as the split will be less affected by randomness. However, this effect is small and should not override other practical requirements.

There are clearly some identifiable areas where larger errors are present, clearly the mathematics governing the behaviour of the variables in that area is problematic to follow with the surrogate. It is difficult to identify exactly where these errors come from. Some will be just a general randomness of sample, which could be improved by more datapoints. Other will be due to interactions of the simulations which creates complex behaviour which is difficult to model with the surrogate. And lastly, some source of error is going to be minor unsolved issues with scripting, that might affect results in the less tested parts of the loop, such as braiding simulation.

4.5.2.1 Iterative improvements, effect on surrogate model

The results presented above are from the original development of the SySi. However, since the initial development many small and few large improvements were added. This has taken effect between initial optimisation through SySi, and writing of the thesis, additional improvements are likely to come from any further development.

Probably the most significant change was the introduction of AVL as an input for force distribution on the part, replacing the concentrated load at the tip of the spar. This addition is further discussed in the next chapter. The meshing was also significantly improved after the initial surrogate modelling, as was discussed in chapter 3.

Along with meshing the braiding was also reworked and some of its problems have been solved or mitigated.

One of the suspected problems of the original surrogate is the volume fraction calculation. The geometric predictions do not take into nesting or undulations, and therefore result in unrealistically high values. This is limited by a V_f cap value. However, when the cap is reached too often, it prevents the surrogate from making correct predictions. In other words, the result is sometimes caused by logic function rather than mathematical one, which is impossible to predict by the surrogate. The volume fraction is also influenced by the mandrel speed, which causes the high error band to be wider than if only the spools number was adjusted. It likely corresponds to some of the high error regions in Figure 4.17.

When the volume fraction algorithm was developed it was assumed sufficient for rough estimation, while the objective is to demonstrate the process of system of simulations. However, due to the way the assumptions were resolved in the script, i.e. the discontinuous calculation of volume fraction, the quality of the surrogate model is limited. To prevent this issue from occurring in system of simulations gateway methods (if, else) should be avoided in construction of any functions that affect the results of any simulation. These are harder to mimic using purely mathematical function. As the V_f was considered a significant source of errors in the surrogate model, an alternative method of calculating volume fraction was introduced.

New volume fraction method separates the weave into smaller section, based on local interaction of individual yarns. This method is a version of what has been outlined by Endruweit at.al. (4), for triaxially braided composite. Similar approach was selected here but for a simpler weave, therefore the calculations for each of the regions are simplified. The new calculation is detailed in the appendix.

There are also many minor changes that could have effect on the success of the surrogate, for instance the boundary conditions for used variable have been changed a few times. Fitness function was also adjusted to better reflect both requirements, low weight and low deflection. The objective function presented in 4.4.1 is the final objective function.

Figure 4.18 shows the error graph from above with new dataset, generated after all the listed improvements above have been implemented. The error circles are scaled so that they are comparative to the ones shown in Figure 4.16.

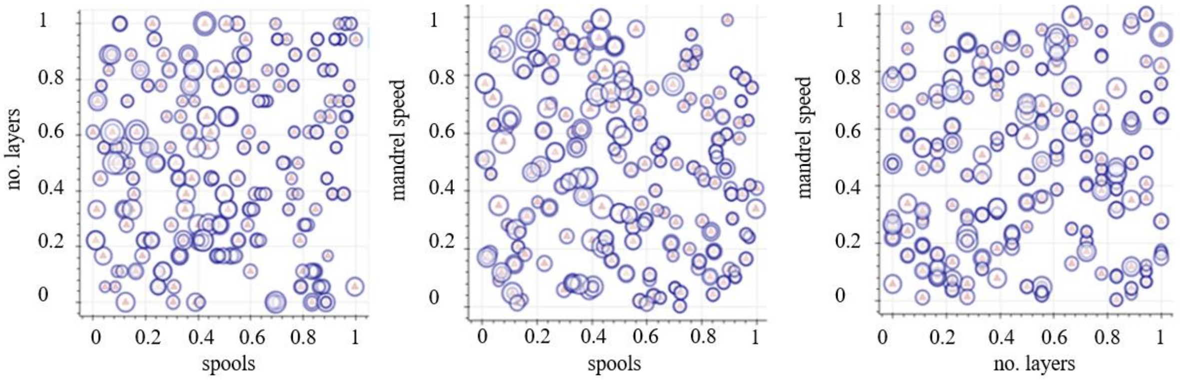


Figure 4.18 – visualized errors of the surrogate model with new dataset

The average error became 1.7%. Also, the error seems to be relatively evenly spread out over the design space. This suggests that majority of this error is caused by randomness, rather than any underlying issues preventing the surrogate from working.

It is quite difficult to assess the effect of individual changes listed above. As each change would require generation of large number of datapoints to assess, few days of uninterrupted run of the simulations. However, author suspects the main improvements were achieved through new meshing method and different V_f calculation. This is not because the second V_f calculation would necessarily be more precise, but because it is more suitable for the surrogate modelling, as it avoids gated functions. This is an example of situation where the optimisation requirements dictate certain type of methodology.

4.5.2.2 Optimisation of the surrogate

To use the surrogate for the optimisation, differential evolution from SciPy library was used. The differential evolution algorithm was run using the surrogate created function. Each of the variables was translated into values 0 to 1, based on how close they are to their respective boundary conditions. The result was then verified by running the obtained optimum parameters through the full loop.

The optimisation, using differential evolution, has been run using the latest surrogate model. For the 3 parameters discussed above, the optimal values were [0.761 0.809 0.364], with 0.49 fitness function value. This translates to 243 spools, 15 layers, and 4.99m mandrel speed. Mandrel speed is expressed in meters, as it represents the ratio of mandrel speed to spool rotation. Rerunning this through the full simulation gives fitness function of 0.51.

The difference between 0.49 and 0.51 points to a persisting error in the surrogate generation. However, this difference was about 5 times worse before the improvements described in previous section were implemented. This shows that tailoring some of the aspects of simulation for this type of optimisation is important for SySi type projects.

The Figure 4.19, representing the input LHS sample, agrees with the result of the optimisation, i.e. the maximum is not particularly clear but can be at the values suggested by the optimisation. This suggests that the surrogate created would be suitable for optimisations and could be trusted.

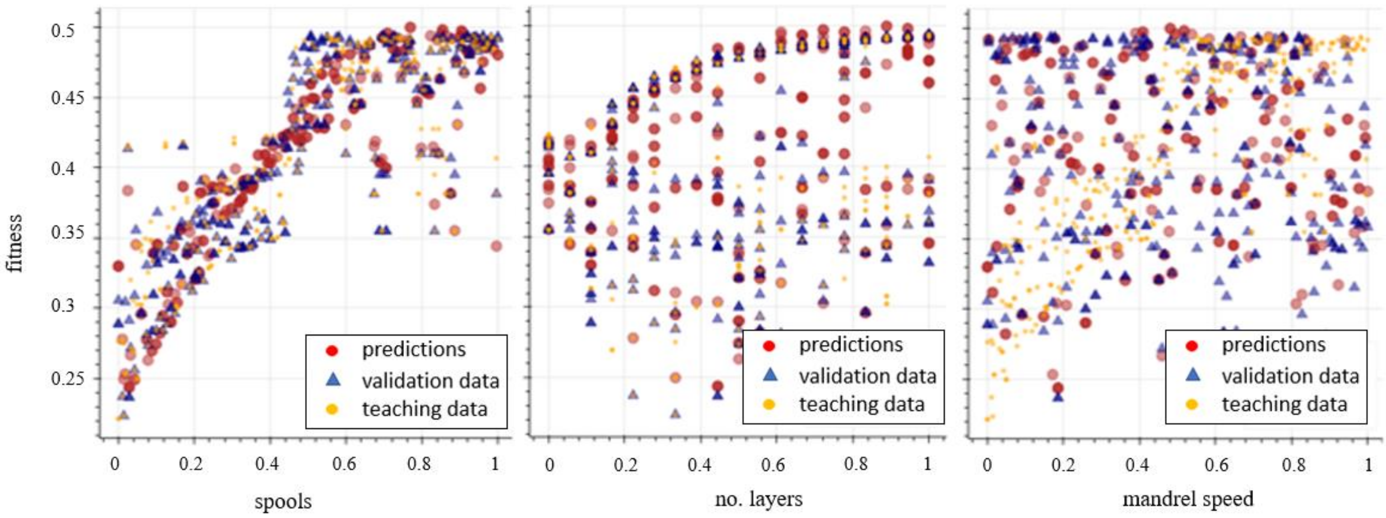


Figure 4.19 – 3 variable LHS with updated SySi

The relationship between variables is simplified compared to the of original SySi results, the difference is likely the V_f calculation which has now clearer effect on the fitness values. This shows that with sufficient data the optimisations are possible. However, when this amount of data is generated, for a specific question the optimisation is not very useful, as it will unlikely improve the result. In very complex scenario and with many different variables, the optimisation might retain its use, but that is difficult to demonstrate using the simple part in question.

The initial surrogate, with higher error levels, was less useful for optimisations. Usually, the optimisation would return high value result, belonging to top 5% of the results from LHS. However, it would not be able to locate the minimum. This shows that the surrogate optimisation has high sensitivity to the errors present in the surrogate.

The next step was to test out, what is the minimum number of input datapoints that lead to this optimal value. However, because of the region of high-end results is quite large this is difficult to assess. Even with 200 datapoints, the optimum point is found in that region. Because of the slightly increased error this is probably further from the optima, but with minimal effect on the actual fitness function. Therefore, for this set of parameters on this part the number of datapoints and the optimisation were both quite excessive.

The methodology has been trialled and more complex part with manufacturing trials would probably be required to further evaluate the potential of this.

The surrogate can also be used to generate general understanding of the relationship between some variables. This can be used at the conceptual design stage for any similar parts.

4.5.2.2 Recycling iteration data in simulations

The success of surrogate modelling of SySi depends largely on the number of iterations that can be run.

Figure 4.20 outlines a potential way of alleviating the processing times by the use of multiple related optimisations.

Optimizing the system for parameters A,B,C generates x number of iterations. Optimizing for D,E,F generates another y number of iterations. SQL can be used to store all this iteration in relatively light-weight manner. All subsequent optimizations can then use pre-existing iterations where relevant, which saves the number of re-runs. At some point, when optimizing for parameters A,D,G majority of required iterations would already be available, and only small number of iterations should be required to identify the minima. The amount of time savings that can be generated by this depends on the particular situation.

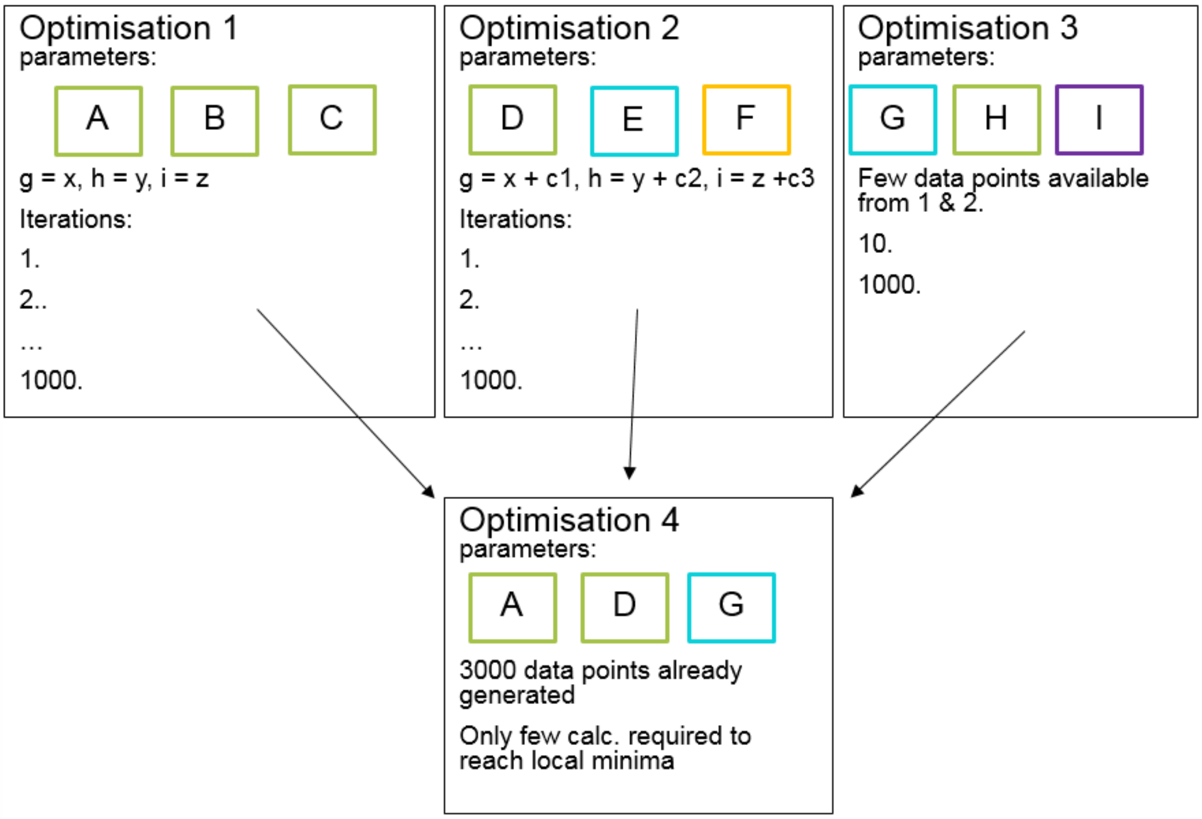


Figure 4.20 – originally envisioned combination of iterations

To assess this idea the run of tree parameters was split into 3 different runs of 2 parameters, to see how it affects the results when surrogate model is generated from this data. The mandrel speed, number of layers and number of spools variables were used as comparison can be done to the runs presented above. This assessment was done before majority of the SySi iterative improvements, so the original compared errors are as presented on table 4.1.

A script was developed that checks all variables changed over multiple datasets and merges the dataset in a way that can be used for SVR. Effect of each variable requires enough data to be reflected in the surrogate model. In this case 3 datasets were added, one for each combination of two variables. The total number of datapoints was increased from 600 to 1576, the errors are displayed in Figure 4.21.

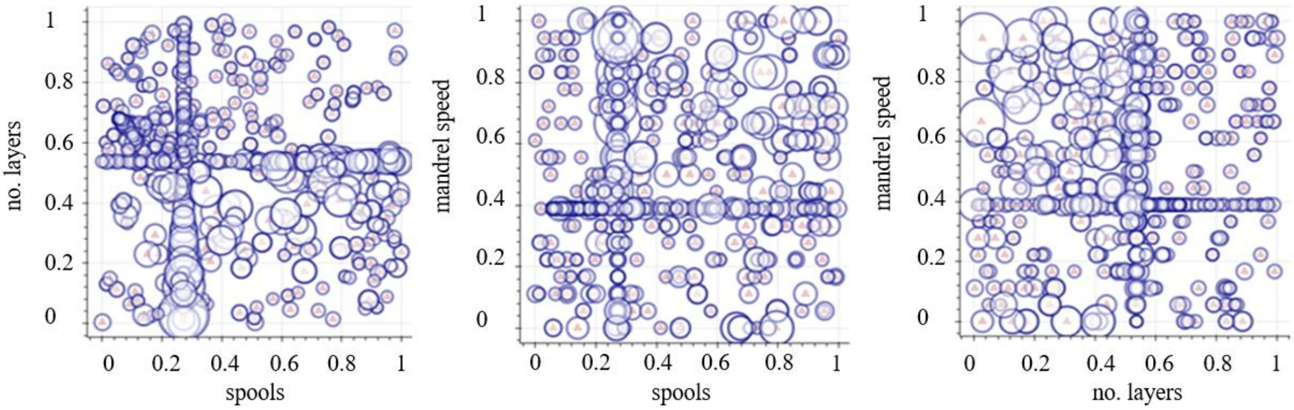


Figure 4.21 – SVR errors with merged datasets

The first thing to note on those graphs is the large concentration of results at a specific variable, this happens when that particular variable is kept constant while the dataset only generates Latin hypercube sample for the other two variables. The average error has not changed in significant manner, and remained at 4.1%, with slightly smaller variation than the original RBF run with 600 results. Therefore, although more data points are available, this is approximately balanced by the decreased uniformity of data distribution.

There are methods that should help to correct the imbalance in sampled data(5). However, at this stage this was considered as future work considering this is hypothetical part, with imperfect background simulation.

Another problem with merging datasets is that all the fixed variables need to be fixed at the same value. If two values are present for any non-iterated variable, new dataset has to be generated to link the fixed variables to the varied ones in specialized run.

With this combined dataset it is more difficult to explore the cause of the error. The newly introduced error is caused by overfitting the teaching data at the specific fixed variable values. Because more data is concentrated there, it has larger effect on the surrogate model. This mixes with any calculation discontinuity errors, such as the volume fraction calculation described above.

In short, this exercise showed that the theoretical merging of datasets is a very difficult task that is far beyond the scope of this project, and undertaking it requires a very specific expertise.

4.6 Chapter summary and key learning

Data collection is major aspect of the design process. The SQL database option proved to be well suited for this type of project. Other options would probably also work, for instance via some form of dedicated file format stored in traditional manner in folders. However, the SQL has several perks that are great for this type of project. It allows for easy troubleshooting of current run via the user interface. It prevents users from storing wrong types of variable, which can help spot errors early in design. It can also be accessed through multiple machines simultaneously, which would likely to be a requirement if this methodology was employed in the industry. The main downside of the SQL storage is the additional software requirement and the database maintenance. It can be affected by software updates which might in turn affect runtimes of the iterations.

Given that the SySi is a small step from design towards software development, a few aspects were brought in. This mainly includes the script storing and version management system, Git. Developing GUI using freely available library turned out to be quite practical as well, although this is an optional module which depends on the variety of expected users for the system.

Sampling the design space was the most successful aspect of the demonstrator. Even though the part was very simple multiple iterations could be run to come up with interesting outcomes in relatively efficient manner. Specific variables can be picked out and their interaction and overall effect on the part can be established. Latin hypercube sampling proved quite suitable for this task, although non-continuous variables have to be temporarily translated into numbers for it to work. This can also help generate understanding of importance of different variables, supporting further design and learning.

Surrogate modelling was proved useful and necessary for optimisation of complex problems. This might not be the case if the runtime of iteration was decreased significantly, by order of magnitude. With the simple demonstrator part, the results obtained through subsequent optimisation are hardly surprising and do not demonstrate the need for the follow up optimisations.

Combining multiple samples, with different variable combinations proved to be highly problematic. With complex management of data this might still be possible, but currently is deemed unsuitable for inclusion in the design process. The difficulties related to implementing this likely outweigh any potential runtime savings.

The surrogate modelling and design space sampling highlighted that certain variables were not defined in ideal manner. More thought should be given to this at the initial stage of systems planning. Variables should be self-sufficient in their effects on optimisation. The selection of shape parameters should reflect the importance of various features, and their effect on fitness function. Variation of airfoil requires continuity or it can easily break the simulation. In general, it is difficult to include multiple airfoil in the same optimisations, especially when combined with continuous variable iteration. Mesh morphing could be one of the unexplored options to solve this problem, this is currently considered as potential future work.

The above problem is especially the case for shape defining variables, as these are also more prone to causing errors, both in the bespoke scripts and in the used software.

There are several things that would be great in retrospect but would be excessively difficult introduce at later stages.

For instance, version control should be synchronized with SQL storage, so that it is clear which version of the system produced what data. This is one aspect of the design that was not captured in the SQL, simply due to lack of authors foresight. Continuous improvement was required more than expected. This creates problems when trying to understand the data and the potential sources of error. It is quite difficult to understand the potential issues in different data samples if version cannot be reloaded to explore. The version control using Git was also introduced at later stages, making it even more difficult to retrace initial changes.

References

1. Moza S. LHS-MDU [Internet]. GitHub repository; Available from: https://github.com/sahilm89/lhsmdu

2. Deutsch JL, Deutsch C V. Latin hypercube sampling with multidimensional uniformity. J Stat Plan Inference [Internet]. 2012 Mar 1 [cited 2019 Nov 19];142(3):763–72. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0378375811003776?via%3Dihub

3. Scikit-learn. Choosing the right estimator [Internet]. Available from: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

4. Endruweit A, Long AC. A model for the in-plane permeability of triaxially braided reinforcements. Compos Part A Appl Sci Manuf [Internet]. 2011 Feb 1 [cited 2019 Oct 25];42(2):165–72. Available from: https://www.sciencedirect.com/science/article/pii/S1359835X10002897

5. Torgo L, Branco P, Ribeiro RP, Pfahringer B. Resampling strategies for regression. Expert Syst [Internet]. 2015;32. Available from: https://web-b-ebscohost-com.bris.idm.oclc.org/ehost/pdfviewer/pdfviewer?vid=1&sid=593aff21-59dc-4ef0-b56c-600b1c328b6d%40pdc-v-sessmgr05

5. Additional developments

This chapter discusses the additional development done on top of the core concept demonstrator. These by no means the only extensions that can be done. The main purpose of this is to demonstrate the modularity of the system, and adjustability for further development. Firstly, two additional modules are discussed: the aerodynamics analysis and resin infusion simulation. This is followed by an assessment of potential commercialization of the system, evaluating the difficulties of introducing this to the industry.

5.1 Infusion simulation

This section of the thesis outlines the expansion of the simulation chain by adding infusion simulation. Infusion simulation is closely integrated within the overall design cycle of composite components. The geometry of the part can significantly impact the manufacturability of the part when it comes to infusion. The local braiding parameters can also significantly impact the permeability of the preform. This might lead to resin race-tracking and generation of voids, it could prevent resin from infusing a specific area, or it could prevent a successful infusion completely, due to overall low permeability. The inclusion of the infusion simulation in the system will allow for verifying the manufacturability and an estimate of the manufacturing time. Both can become part of the fitness function for future holistic optimizations.

The options regarding infusion software are considerably more limited than for structural analysis or CAD design. There are two software that are commonly used; these are the PAM-RTM module of PAM-COMPOSITES by ESI, and the LIMS developed by University of Delaware.

The work presented in this chapter consisted of evaluating the scripting options provided by ESI’s PAM-RTM and then developing self-generating models which complement the system of simulations. The aim was to create a completely self-generating model, where the inputs are parameters and outputs of previous models. Methods for accessing and processing the results of the simulation are developed to allow for automated iteration of the simulation. In that regard, reliability and speed of the model generation and simulation are particularly important. Both will greatly impact the usefulness of the model.

The utility of the developed PAM-RTM Python scripts is demonstrated. The last section demonstrates how the scripts can be useful even outside of the system of simulations.

5.1.1 Methodology for scripting development

When scripting commercial software there tend to be two very helpful aspects. Firstly, complete documentation outlining all the script-enabled functions can be of great help. Secondly, online community of engineers facing the same scripting problem. PAM-RTM does not offer scripting related documentation, and the community of people scripting this software is either very small or non-existent. However, PAM-RTM does have a tool which records the actions within the user interface. All the actions are recorded as Python commands which further improves the utility of the tool.

This is not something that one can easily learn through documentation or general usage of ESI. Upon contacting ESI support I was pointed to the recording location of each session: “C:\Users\user_name\Documents\VE\14_5\vsession_XXXXXX_XXXXXX_XXXXX.py”. Information on how to run the script was also provided: "C:\Program Files\ESI Group\Visual-Environment\14.5\Windows-x64\VEBatch" -activeconfig Trade:CompositesandPlastics -activeapp VisualRTM -sessionrun "script.py". However, the scripts usually need adjustment for re-runs and it is easier to run these using the subprocess library, in the same way the Abaqus Python scripts are run.

Therefore, the method for development of the PAM-RTM scripts consists of manual setup of a model, followed by tailoring of the recorded function to create the automated model. This way an understanding of the functions used is also developed. The downside of this approach is that the functions used are only the ones available in the GUI, which are optimized for manual operation, rather than for fast and computationally inexpensive iteration. Also, significant amount of trial and error is required.

To develop automated extraction of results from PAM-RTM, all possible export formats and files generated during simulation have been assessed. The main file is the log-file, which can be interrogated with bespoke script. All the other results, that are normally displayed in “visual-viewer”, can be also exported as text files automatically. However, each is standardized for that specific export, which means that for every type of information programmatically exported new script for interrogation of the text file must be developed. This could be simplified significantly if ESI provided a library of post-processing tools, similarly to Dassault’s documentation for the Abaqus FE. However, to the best of author’s knowledge such library doesn’t exist and hence this is likely the most efficient approach.

5.1.2 Key findings about the scriptability of PAM-RTM

Table 5.1 shows the main tasks done when building typical infusion simulation in Visual-RTM. Each of the tasks was automated, then all tasks were run in a loop. The looping adjustments are discussed in later sections. Complexity in the table refers to how easy (1) or problematic (5) it was to automate the task in the ESI software. This is to some extend subjective rating, but it is mainly caused by how the functions are structured in the ESI's UI, how they are structured by recorded script and how easily these two can be matched and understood.

Runtimes displayed are for orientation and do depend on the part, mesh, and other aspects. The times presented correspond to the part used in SySi and CATIA pre-mesh discussed in the meshing section. Due to the simplicity of the demonstrator part, the table very well highlights which aspects are slow simply because of the PAM-RTM programming or interface.

Table 5.1 – automated PAM-RTM tasks

No.	**Function/task**	**Complexity**	**Average Runtime**
1	Import of pre-meshed part	1	Negligible
2	Surface referencing	4	20min
3	Surface to part assignments	2	1 min
4	Mesh refinement	1	2 min
5	Importing resin data	Not automated	Not automated
6	Infusion parameters setting	1	Negligible
7	Rosette definition	2	2min
8	Define reinforcement material for each segment	3	5 min
9	Layup manager (matching materials and parts)	5	25 min
10	Assign rosettes	3	3min
11	Create regions (for BCs etc.)	3	1 min
12	Flow rate definition	4	4 min
13	Vent definition	1	Negligible
14	Inlet definition	1	Negligible
15	Tool temperature definition	1	Negligible
16	Saving and running the simulation	1	5-10 min
17	Recording fill time and fill factor	2	3 min
18	Recording fill at specific time (or similar)	3	3 min
19	Post processing the simulation data (various)	3-5	1-8min

Most functions are quite straight forward and can be recorded from working with the GUI, requiring only minor script adjustments. However, few can be very problematic. For instance, the importing of various lookup tables to define resin characteristics was not resolved programmatically. The recording does not specify the table that is being adjusted, therefore only one table can be used for a material, preventing the use of multiple lookup tables corresponding to different temperatures. Author has tried various standard referencing systems used by Python that would fit the typical ESI function syntax, but with no success. Therefore, the current advice regarding this, is to import lookup tables ahead of time and only use scripting to select the materials from already imported library. This would not be suitable if cure kinetics were part of the holistic loop, which could quite easily be a requirement for certain parts. In such case the only way this could be addressed from users' side, is to adjust the files corresponding to materials stored. These files are not very intuitive, and it would take a significant amount of string alteration to be able to change material properties that way. It might be easier to persuade ESI to change this, as it would probably be significantly less effort to adjust this from their end, for the next version of PAM-COMPOSITES.

In terms of run-time, two major bottlenecks were identified. First one is the surface referencing. The only way the author managed to reference surfaces in controlled manner was to output each surface, recording the reference number, averaging all relevant nodes to find approximate 3D position. The 3D positions were then used to allocate the surfaces to respective parts. Due to repeated export of .inp files the runtime is quite large. This is required because the surface referencing PAM-RTM creates upon loading a part is highly inconsistent; two very similar parts will end up with differently ordered surface numbers.

The layup manager bottleneck just seems to take a lot of time even when manually adjusted in the GUI. The script replicates what is being done in GUI, rather than just importing the information in bulk. Hence, when adjusting 200+ segments the runtime is large. The recording requires accepting changes line by line, this can be streamlined by inputting all changes and then accepting them for multiple rows. However, this improvement only decreased the runtime by about 10%, only slightly mitigating the problem.

These two bottlenecks are problematic independent of mesh-size, additional problems are created when attempting to significantly increase number of elements.

It should be noted that it is quite likely that author did not find the most efficient methods to do these tasks, but the best attempt has been made, considering available documentation.

5.1.3 System of simulations

The part developed in core SySi is considered here, as implementation of PAM-RTM automation into it is the main reason for this module. The cross section, and other airfoil related characteristics, are completely adjustable. The permeability prediction requires further development, but the aim is to have a predicted permeability for each segment based on the outputs of braiding simulation. Currently the permeability is predicted based on volume fraction and local braid angles, which are assumed to behave approximately like sheared woven preform. This method is probably very imprecise but can be replaced by plugging in a ready to use tool in the future. For now, the focus is placed on demonstration of the automatically generated and interconnected models.

Previously generated data is imported from SQL, as input for the infusion simulation. The main feature of the data is the cross-section and spanwise segmentation, as per Figure 5.1 (right). Each of these segments has properties specifically calculated from the braiding process. The Figure 5.1 (left) shows the predicted paths of yarns on the mandrel, it only shows representative sample of the number of yarns, data points required for the missing gaps is filled by interpolation from neighboring yarns to avoid excessive runtimes.

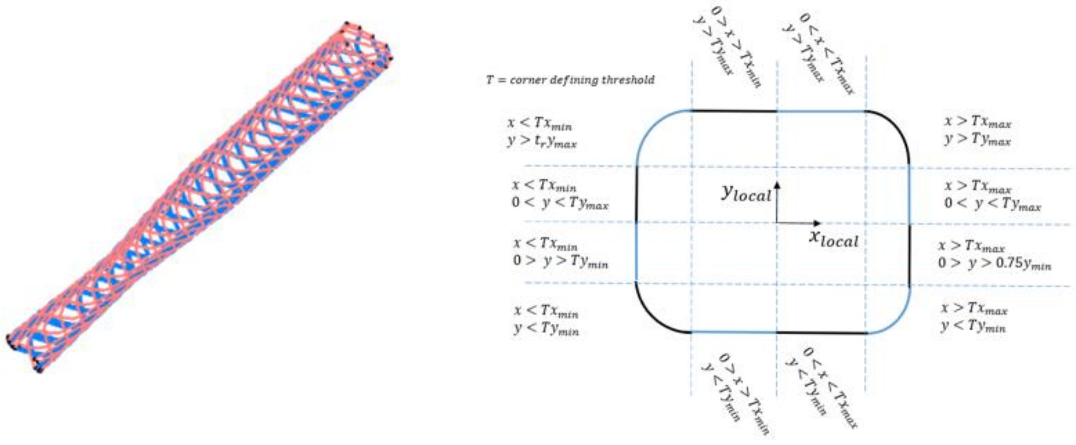


Figure 5.1 – the part with portion of braided yarns for illustration (left), cross-section segmentation for material properties calculation and assignment (right)

The segmentation shown is reflected in the automated RTM simulation. This created requirement of referencing based on imported surfaces, as highlighted in previous section. Each of the segments has a bespoke material created during the automated simulation. The material is then matched to simulation elements based on location in 3D space. Translations between global and local coordinates are used to accommodate for twist; spar twist is one of the global parameters and therefore must be accommodated for. The boundary conditions are applied on all root/tip nodes available based on the z-coordinate.

The simulation takes .iges file created in CATIA as an input. This .iges is a set of approximately square surfaces that represent the rough mesh of the part. The reason for the controlled mesh import is mainly prevention of PAM-RTM errors, leading to robust iteration.

Multiple meshing methods were developed for the system of simulations. It is difficult to create one set of scripts that would for example take both .iges with surfaces and orphan mesh, and run the infusion simulation. The current approach to automating inclusion of various mesh types, would be to create a gate function, and having a part of script designed specifically for the particular mesh. This is quite impractical, as it would significantly increase the initial work required to generate the system of simulations for a component. With current tools in industrial settings, this should probably be overcome by specifying a robust methodology for design. Rigorous selection of the most suitable meshing method should be done at the start of the project, minimizing the changes required. However, with the global interest in digital twin and similar concepts, it is likely that commercial software will soon work well with any standardised file format provided.

The whole RTM model generation is automated, from importing .iges file to extraction of results following the simulation run. The results extracted are the infusion time and infusion percentage. Other data can be extracted, but for the purposes of iterating through this simulation these two are the most important. The model generation and simulation time is also recorded for further assessment of the script’s performance.

5.1.4 Infusion Improvement Strategy

The first clear benefit of the automated model is being able to iterate through itself and adjust available input parameters to improve the expected infusion. Two methods for this were developed, and many others were considered.

Method 1

The first method was to add flow-mesh ahead of segments that do not infuse. In the simulation this was done by increasing the permeability before problematic section. Problematic sections were identified by region where nodes have fill factor between 0.01 and 0.95 at the end of infusion. However due to the very small part considered, the automated simulation typically suggested so much flow-mesh, that it would be more practical to apply it across the whole part. Any flow mesh added would likely create a void on the other side of the part, until the requirement for flow-mesh was identified almost everywhere. To summarize, the capability to automatically iterate this strategy within the software was created but was found impractical for the demonstrator part. On the other hand, the demonstrator part is unlikely to need this methodology anyway. The testing was done on artificially created issues, where the permeability differences were exaggerated.

Method 2

Second option that was considered was flow rate adjustments. With continuous flow front, voids and weld lines are less likely. The probability of successful infusion can also be improved. For this method the flow rate definition was changed from individual values to a matrix, which provides flow rate for each segment, out of the 12 root segments, at any point in time. Due to the simplicity of the part that many injection points would probably be excessive manufacturing strategy. However, as a proof of concept for the infusion improvement it is still valid.

Firstly, the initial infusion simulation is run with constant values across the matrix. The result files are then checked for disruption in the flow front. This is done by looking at the nodal filling factors at a point in time. If the filling factors that correspond to flow front (0.05< fill factor < 0.95) are occurring at points more than specific distance in z direction from the average flow front, it is considered flow front disruption. To adjust for it the flowrates before this point in time are adjusted. The sections that are lagging receive higher flow rate and vice versa. A minimum and maximum flow rates are specified. If the flow rate adjustments do not improve that point in time, the z direction is adjusted to allow for larger flow front disruptions and the adjustments proceed along the span. This loop is repeated either until the flow front is uniform all the way to the end, or the flow adjustment cannot achieve better results with the given limit values. Figure 5.2 shows an example of reasonably successful improvement in flow front over 3 iterations of the flow rate adjustment script; the two pictures are taken at equal infusion time.

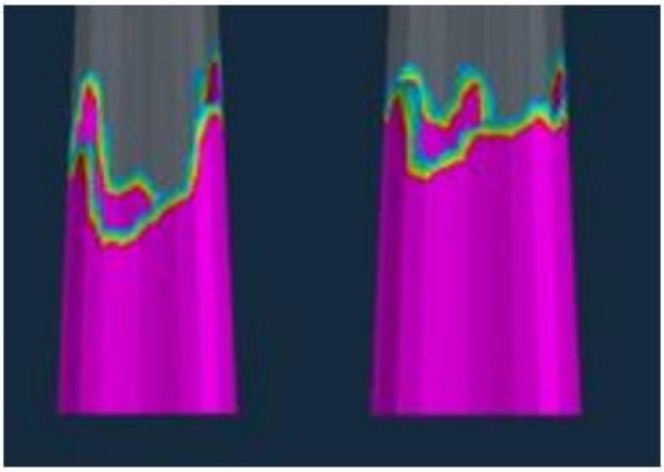


Figure 5.2 – Adjustment of flow rates to improve flow front

It was found that in specific circumstances this approach can help. The method is quite good at keeping the flow front uniform close to the inlets, but as the flow progresses beyond half of the spar the adjustments to individual inlets have very little effect on flow front due to transverse permeabilities. Also, it works better where transverse permeabilities are significantly lower than spanwise permeabilities. This is for two reasons. Firstly, the problems with flow front are more likely occurring in that scenario. Where the permeabilities are relatively constant there is no point in running the script, as the flow front is likely good already. Secondly with high transverse permeabilities the flow traversing sideways rather than spanwise is more likely, making the flow less dependent on the closest inlet.

5.1.5 Integration options

There are two main ways to utilize the simulation as part of the simulation chain. First one is to use it as verification only. This accounts for comparatively long runtime of the infusion simulation, when compared to the other models currently in the system. Also, the part is relatively small, therefore it is unlikely that infusion time would be a significant design performance metric. In this approach the simulation chain runs through many iterations, and only the top 5% based on fitness function are also run through the infusion simulation. This allows for the chain to rule out otherwise high-quality optimisation runs, that are not predicted to fully infuse.

The second option is probably more suitable for larger components. However, for the sake of completeness it might be reviewed in the future. In this scenario the infusion time forms part of the fitness function of the whole system of simulations. In this case for every loop of the system of simulation, the RTM simulation is also executed. The runtime will increase, but for large complex parts the infusion time might be quite important feature for evaluating different design options. Similarly, high volume parts might also significantly benefit from minimizing runtime.

As the model generation of the PAM-RTM simulation is currently the most computationally expensive part of the system of simulation, the first option is currently employed. The fitness function comes mainly from deflection analysis, using the weight and deflection as the only evaluation parameters.

The simulation iterates through the core loop and once high enough fitness value was reached that part is checked for successful infusion in PAM-RTM, see Figure 5.3.

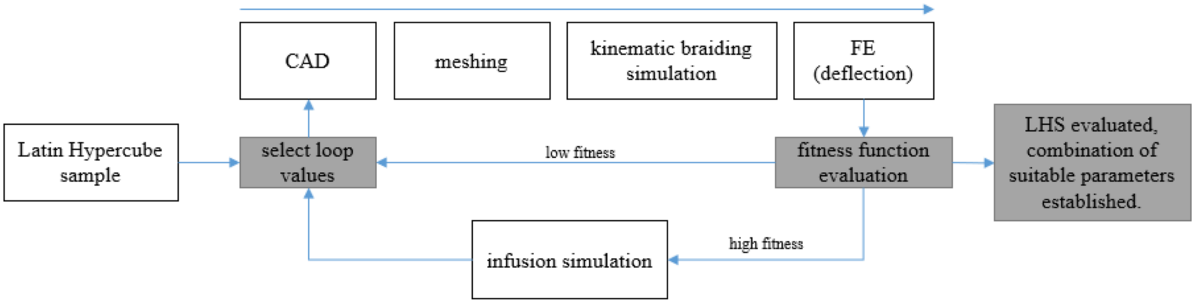


Figure 5.3 - simulation loop with infusion included

Unfortunately, without altering input parameters the demonstrator part never has issues with infusing. Therefore, this module is not an important part of the SySi loop. However, for other parts this could be quite important, and because of the potential permeability inputs it could be a module that benefit the most from the integration. The automated scripts were adjusted and used on other parts to verify it utility.

5.1.6 Blade previously analysed at Purdue CMSC

To show the automation benefits in industrial setting, a blade previously analyzed at Purdue’s Composites Manufacturing Simulation Centre (CMSC) was considered, the part and the tool is displayed in Figure 5.4.

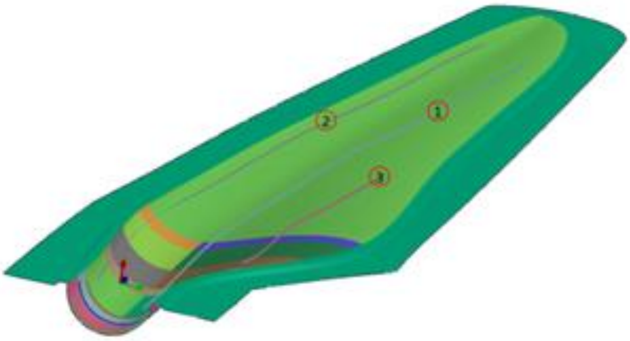


Figure 5.4 - previously analysed blade

The part was previously analysed and parameters for successful infusion have been obtained. However, 10 different inlet points were used. It was not an issue for this part, but for larger more complex parts it might be desirable to keep the number of inlets at minimum. In order to achieve that objective, the inlet positions have to be optimised.

To demonstrate this on the selected part the number of inlets was decreased to 3, one on each of the 3 span-wise resin channels shown in Figure V4. The previously generated model was used with all its other settings. The position parameters were defined as proportion of channel spanwise distance.

Latin Hypercube Sampling(1,2) was used to iterate through the design space. The 4 parameters adjusted were the resin inlet temperature and the 3 positions of inlets, as percentage span.

Figure 5.5 shows the results of sampling the design space. Figure 5.5 left shows the infusion percentages along with inlet positions. It is difficult to see any general trend, in terms of a specific position on either of the three inlets. Therefore, it is likely that the interactions between the different variables is quite significant. The effect of temperature, shown in Figure V5 right, is as expected. With higher temperature the successful infusions take less time, but also at very high temperature the successful infusions are less likely. This is likely due to fast curing of the resin.

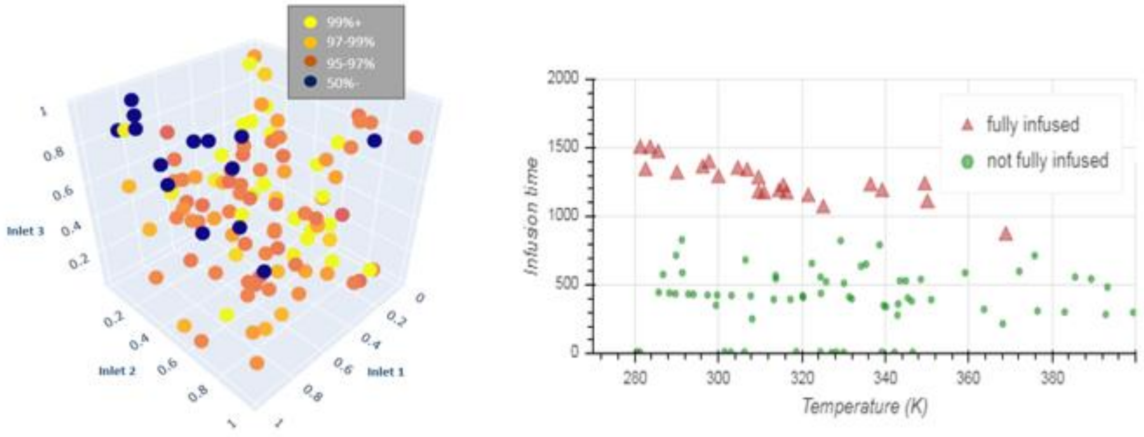


Figure 5.5 - sampling the design space

Only about 15% of parameter combinations lead to complete infusion. The most common issue was the tip of the blade not filling completely. Therefore, for trial and error the expectation for first successful positioning is 7 attempts, assuming random selection of parameters. In case of manual assignment of parameters it is likely that engineering judgment would decrease that expected number of runs required.

The amount of engineering time saved by automating the model setup will depend on the engineer and the part. With this part author assumes that for about 10 runs it is worth automating the model rather than manually editing it between runs. With complexity of part, and practice automating the PAM-RTM, this threshold could decrease further.

5.1.7 NCC’s braiding material properties project

The developed PAM-RTM automation was also used to support a research project at NCC. The project required assignment of permeabilities in localized fashion over quite large mesh. Using the scripts 1000 different permeabilities were assigned to different section of the part, allowing for the simulation to closely resemble manufactured trial parts.

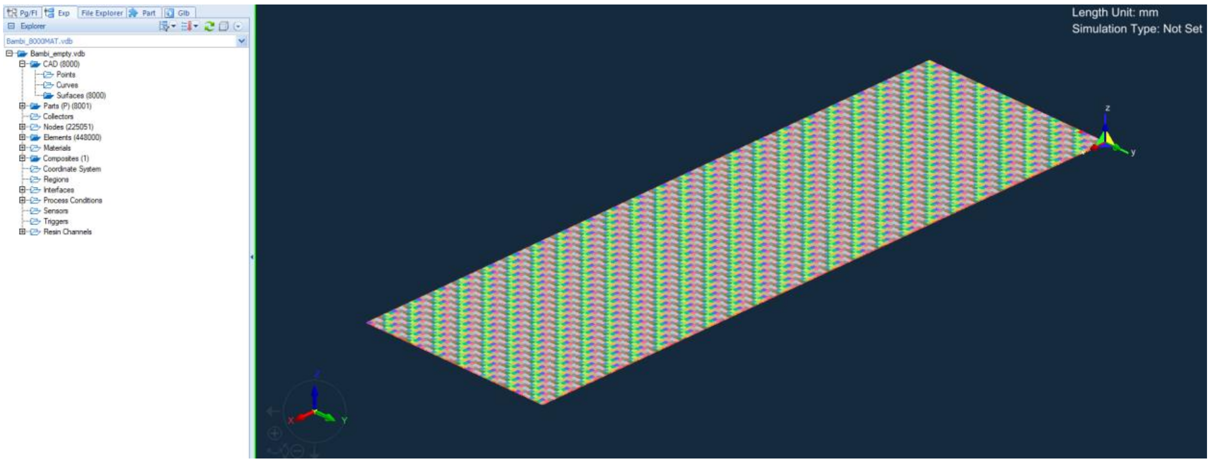


Figure 5.6 – segmenting flat plate to 8 thousand separate areas

Limitations of PAM-RTM automation were also discovered during this project. Attempting to segment the part into 8000 material zones was unsuccessful. The methodology was the same as for 1000, but the computational requirement increases more than linearly with more materials involved. Author suspects this is because the script still controls the user interface, while the user interface used for matching materials and zones in PAM-RTM is quite slow and error prone even with less values involved. Figure 5.6 shows 8000 surface created; when the materials are being assigned to this part, the software crashes.

5.1.8 RTM module Summary

Tools for automating the model generation and simulation within PAM-RTM software were developed. These tools were used to increase the scope of system of simulation, which aims to analyze composite parts in more complete manner. Accounting for interactions between various models could increase the quality of the infusion model.

The scripts were also applied on previously analyzed part to demonstrate their effectiveness in standard process simulation workflow.

To further exploit the benefits of automation and connectivity it is crucial to develop reliable methods for predicting material properties, such as permeability, from braiding parameters.

All the parts automated within this study had low complexity, making the automation inefficient from the perspective of engineering time. However, it is quite easy to see how large part with multiple inlets would benefit from the automation and optimization.

The scripts are available in the SySi GitHub repository. However, for the simple demonstrator part this module is not integral part of the loop. Therefore, this module integration might not be up to date with the current state of SySi at all times.

5.2 Aerodynamics analysis

This module was included to take aerodynamics into account. The main goal is to inform the structural finite element analysis, to input the force distribution better corresponding to aerodynamic forces. The inclusion of simple aerodynamic assessment will also open the path for including the aerodynamic requirements in the objective function. These could be requirements for lift to drag ratio, total lift or similar. However, because the focus here is on the spar rather than on the whole assembly this aspect is less relevant.

There are many open-source software options for low fidelity aerodynamic simulations, as has been outlined in the literature review. Several software have been trialled. AVL has been selected, the main deciding factors were the ease of use and availability of teaching examples(3). YouTube based guides on how to use basic AVL are available (4). AVL uses extended vortex lattice method. This can be used to find the lift distribution.

5.2.1 AVL module development

There is limited information on how to run the AVL from command line. SUAVE was reviewed to improve understanding of how this could be done. SUAVE uses Python scripts to run AVL, method analogous to what is required for SySi.

After reviewing SUAVE and the AVL tutorials key files were identified and understanding of their function was developed. The most important file is the ".avl" file which specify the shape of the aircraft in its entirety, for standard shapes this is easily understood. However, for non-typical configurations it might be difficult to specify all shape details with this format. The ".deck" file specifies all command that need to be run within the AVL. The commands required were first run within the AVL command line, to verify each command does what is expected. The manually tested sequence is then turned into the ".deck" file, where each line represents a command. The ".run" file specifies any other aerodynamic parameters required.

Subprocess method was used to run AVL from command line using Python, method analogous to the RTM or Abaqus work. For practical reasons, the command run through subprocess is done using anaconda installation of Python. The script "avl_cmd.py" would have to be adjusted if basic Python installation was used.

Because this module introduced significant changes to the core SySi scripts, most notably the Abaqus module, new branch on GitHub was started. This module was not originally planned for the SySi and therefore the variables available are far from optimal for any potential optimisations using aerodynamic parameters.

The scripting of AVL was split into three relatively small scripts: AVL_inputs.py, avl_cmd.py and AVL_PostProc.py. The avl_cmd.py runs the other two scripts by the subprocess method; it also saves all relevant parameters in .npy format so that these can be accessed from the subprocess. The AVL_inputs.py, as the name suggests, sets up the inputs required for AVL analysis. Most notably the .avl file is adjusted; default template of the .avl file is adjusted to fit the shape parameters used in an iteration. This script works to some extend as interface between the SQL and the aerodynamic module. The script simply considers the text file as a string. Markers were introduced into the default .avl file so snippets can easily be extracted and replaced by the required variables. The variables definition used in SySi is highly compatible with the typical definition of aerodynamic shapes in AVL. Both use cross-section based definition, were the airfoil and other parameters are defined at various locations along the span. The AVL_inputs.py script also runs the analysis.

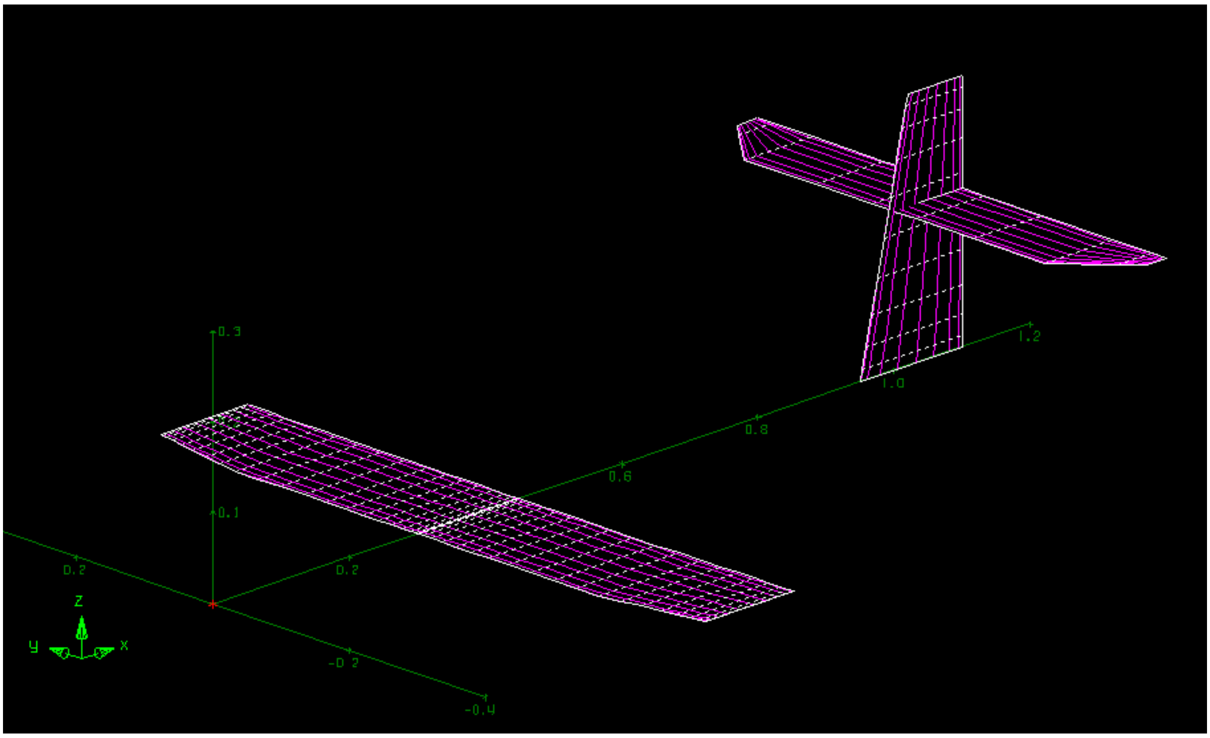


Figure 5.7 – default AVL configuration visualized

Figure 5.7 shows the visualization of default .avl file. The configuration is not particularly important here. The only section that changes according to the parameters of SySi is the main wing.

The AVL_PostProc.py script is used to interrogate the output files of AVL. The output file is a standardised text document with all relevant outputs listed. Currently string manipulation is used to extract lift distribution, but any other values can be taken out if necessary. The output lift coefficient is translated to lift using the lift coefficient formula:

$$L = C_l * \frac{A * \rho * v^2}{2}$$

With these two scripts the AVL can be run with the SySi parameters. The integration within SySi requires small addition to the MASTER.py script that controls when each module is run. The main edit that was required is the adjustment of abaqus_inst.py which deals with the pre-processing of Abaqus.

The forces were then mapped onto the node sets in the same way materials were mapped, as per chapter 3. Therefore, the rest of the integration consisted of reusing already existent tools and troubleshooting any errors that were caused. Figure 5.8 illustrates the distribution of the forces, highlighting one segment. Each of the 20 spanwise segments correspond to the two bottom areas used to allocate material properties.

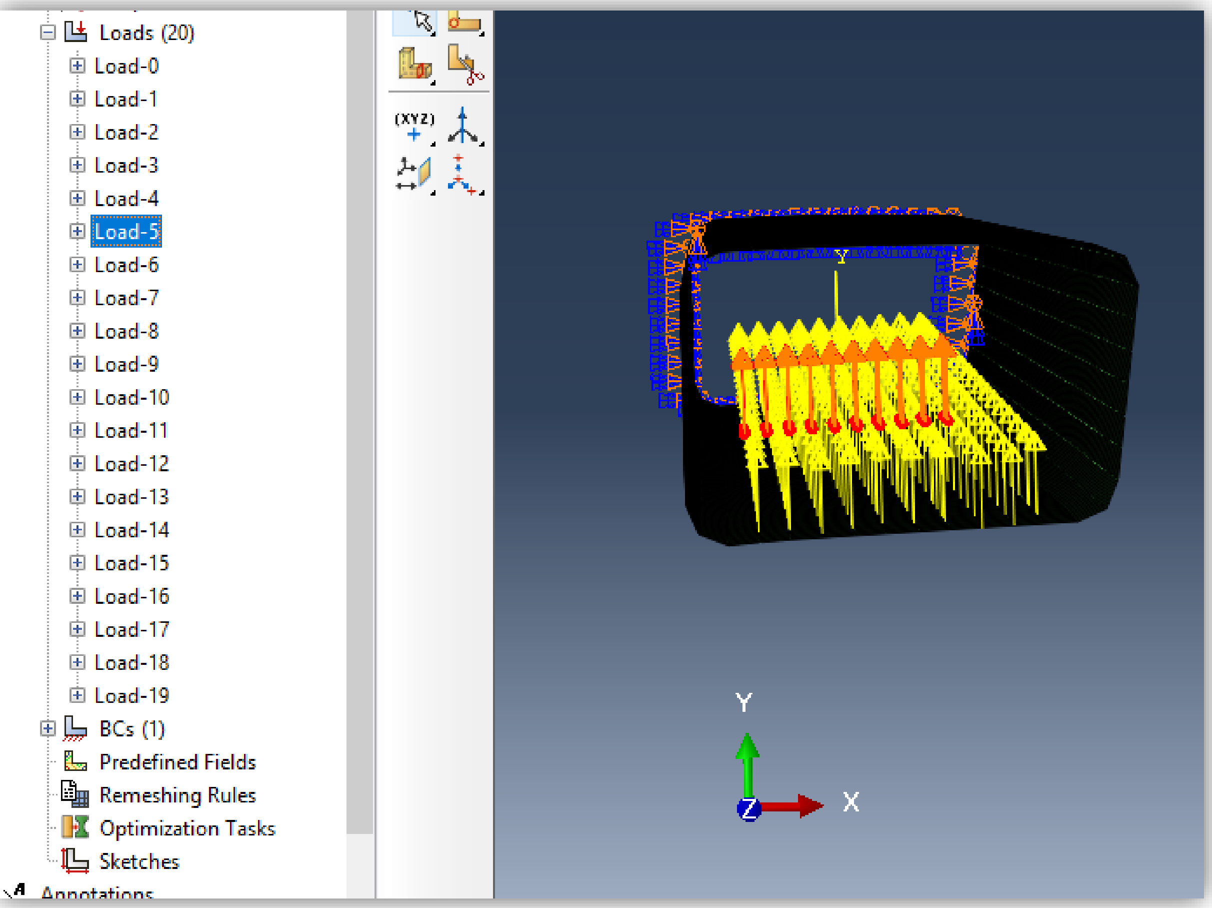


Figure 5.8 – force distribution applied in Abaqus

The runtime of the AVL related scripts and the AVL analysis is almost negligible compared to other modules.

Although the variables used in AVL are very good for the core task, informing the structural analysis, they do not support easy implementation of AVL centred optimisation. First problem is the mixture of numerical and string variables, as airfoils are specified by pre-defined point clouds. This makes it difficult for potential optimisation algorithms to move through design space and obtain ideal solution. Second, possibly even more difficult problem, is the interlinking of variables. Any aerospace engineer has a good idea of the relationship between the various parameters. For instance, the wing will likely be tapered from root to wing tip. The airfoil is also unlikely going to change too much along the span of the wing. These and many more ideas embedded in engineer’s knowledge would have to be embedded in the optimisation as boundary conditions or parameter-linking mechanisms. Without this the design space for the optimisations with its variety of combinations of variables poses too complex of a problem.

Embedding other outputs of AVL, such as L/D, into the objective function is a viable future extension of this work. For that it would be highly beneficial to consider industrial part with more realistic and better-defined specification.

5.2.2 AVL Module Summary

The integration of the AVL module to inform force distribution for structural analysis was performed successfully. The most difficult aspect was the identification of the right software and the methods used to automate it, to make it suitable for SySi. The scripts to pre-process, post-process and integrate the module were comparatively simple to create.

The selection of the parameters is of a particular importance in this module. The parameters need to fit the software used and allow for the type of optimisation required. These considerations should be weighted at the start of the project when the complete system of simulations is scoped and planned.

Interestingly, the parameters selected at the start of SySi were quite suitable for AVL. The same parameters that cause the problems with optimisations described in chapter 4, are the ones that make the implementation of this module easy. The mesh-morphing that could be considered as potential future work, would make implementation of this module much more difficult. To some extent this reflects the discontinuity present in the design process of today.

There are various pathways that could be taken to improve on this work. However, it would be highly dependent the part in question. Therefore, it is not particularly useful to explore these using the very simple demonstrator part.

5.3. Introducing SySi to industry

This chapter assesses various aspect of bringing a version of SySi to the industry, mainly focusing on the design methodology. The novel methodology is defined by the automated scripts connecting various simulations within the design chain of composite component. The envisioned benefits are decreased rework, holistic optimisation, and better understanding of complete product design. At the time of this study the demonstrator has already been developed. It has been shown that connectivity of the various software can be achieved using Python scripting, leading to a complete optimisation, where the same parameters are used in all simulations involved. The main downside is the additional scripting work required.

The overall aim of this study is therefore to assess the suitability of SySi for industrial applications.

This chapter is based on review done as part of a "Study Tour" unit that is part of the EngD programme.

The first objective of this study is to assess current design process in an aerospace company, and any potential requirements for improvements. The second objective is to establish the difference between the company's envisioned improvements and the SySi, in terms of tackled issues and proposed solutions. Then, individual modules of the SySi are evaluated in terms of their usefulness in their current state. Pathways of implementing SySi in industrial settings are then outlined, based on the findings, with some technical notes on work that is still required to make SySi suitable for industrial application. Business strategy is then considered, and its effect on commercialization of the novel process.

Modules from SySi that could be used within the company's envisioned process are listed.

Two main methods for implementing the SySi methodology are then discussed, internal implementation and external experts hiring. Other variations of the potential implementation are briefly discussed.

The cost savings achieved by suggested methodology are assessed. The cost savings only consider the engineering time required for standard process currently implemented against the engineering time required for SySi implementation.

Introduction of SySi through a consultancy type company is discussed, focusing on the business aspects. This includes PESTEL analysis reviewing the aerospace industry. Business model canvas is used to define the envisioned consultancy. Porter's 5 forces method is used to outline the market position. Market size is also estimated. Business strategy is then considered, and its effect on commercialization of the novel process.

The study is based on series of conversations with engineers from an aerospace company. First, author presented to them about the SySi work, outlining the outstanding questions regarding their design process and development outlook. Second, the person in charge of "digitization" and development of their design process presented about their current practice, their work towards digitization and their future outlook. This was followed by a discussion, structured as an informal interview with open ended questions. The discussions inform this report and guided some of the latest SySi developments.

5.3.1 Summary of the design process within the studied company

An aerospace company was considered for this study, it will be referred to as “the company” for confidentiality reasons. The company produces components that use braiding and manual layup for relatively complex shapes. The parts produced are assessed aerodynamically and are load bearing components. The information here is based on general knowledge of the company’s product and discussions with the company engineers in charge of improving design process, through digitization.

5.3.1.1 Current design process

The design process of the company currently consists of core modules and secondary modules. Core modules are the ones that are subject to high amount of analysis before completion. The core modules are aerodynamics & performance, CAD design, and structural analysis. These modules are usually iterated over. However minimal optimisation through all three is done due to time constraints. The secondary modules consist mainly of manufacturing and controls teams. These two departments should ideally not send design back to core modules. The engineers working on core modules accommodate for key requirements of the secondary modules based on experience. The experience or practice does decrease the chance of rework but does not remove it entirely.

Most key parameters have to do with aerodynamics and performance, with the performance dictating the outer shape of the part. CAD is mainly used as the reference file for what is the current setup, but CATIA is also used for draping analysis. The finite element model is then used to verify that the part satisfies all structural requirements. Minimization of vibration induced noise, and maximization of efficiency are commonly the main evaluation parameters. While the efficiency is composed of how well the design fits all requirements for operation.

The typical manufacturing consists of hand layup, braiding and infusion. The layup is defined and analysed in CATIA within the main process. However, minimal optimisation of layup takes place.

Usually, data from same or similar layup sequences is used. Analytical braiding simulation is done using spreadsheets, with the main objective to specify braiding parameters for the machine. The infusion simulation is rarely necessary, as infusions are rarely problematic. However, LIMS software is sometimes used to verify infusion and to design the tool.

The core modules are usually iterated through about 4 times. This is not by design and ideally this would only be done once, as this process is very expensive. Once the product is finished and used on flying aircraft, there are never any changes or iterations to the product. This is because any potential improvements are outweighed by the certification costs.

Manufacturing of trials is often not done for parts that are not confirmed to go into production. This means that iterations of simulations exist, which are unused unless learned from directly.

One of the reasons for iterations being expensive is quite inefficient data transfer. The parameters are passed between teams using series of forms and the master design is held by the CAD file. This makes generation of parametric models or simulations quite difficult.

Another difficulty perceived by the engineers is the engineering downtime, caused by waiting for other aspects of design to be finished. This is due to the way the design process is structured and the way information is passed between departments.

This might not be the case of the studied company, but in general aerospace companies often struggle with access to CAD files, as the number of CATIA licenses is usually based on engineers working on design. However, when CAD is used as the main definition of design many other people will likely need to access the models.

5.3.1.2 Internally envisioned changes

The Company is aware of some of the issues in current design workflow. There is an internal project looking at reworking the process, digitizing it. This is in many ways similar to SySi. The main objective is multi-simulation optimisation, better data management and utilization of modern optimisation tools. This is achieved with the use of parametric models rather than automatically generated ones.

In a typical optimisation, objective function of the part is composed of various parameters corresponding to performance requirements. Only a select few parameters are adjusted at the same time, as it is understood that the difficulty of producing high quality surrogate model scales exponentially with number of parameters.

All data from optimisation iterations is stored for later use, for example in creating further surrogates.

There is a large focus on distinguishing between high fidelity simulations and low fidelity estimates. The goal is to run optimisations only based on low fidelity models, supported by surrogates built from high fidelity models. Current optimisation mainly focused on aerodynamics, performance and structure. The CFD model is only used to verify a simpler model which is based on data from previous runs. The part is segmented, and each segment is assigned aerodynamic coefficient based on previously run CFD. In place of structural FE, a simple bending analysis is executed. The segments are defined by airfoils which are specified by Bezier curves, 4 points used to fully describe the curve of the camber.

Along with the iteration and simulation integration, the improvements efforts are also focused on better utilization of materials. For instance, the layup sequences are rarely optimised due to time constraints. Similarly, braiding simulation is mostly used to define machine parameters, while it could be used for optimizing braid angles to better suit the material requirements. For this kind of optimisation the material properties from braiding have to be efficiently turned into inputs for structural models.

Improvements to high fidelity models, such as fatigue, are also envisioned.

So far, the development of these improvements is a study mainly developed by dedicated individuals. Therefore, no manuals of how to work towards this exist for engineers. Therefore, this work also faces the difficulty of being introduced into a running commercial process.

5.3.1.3 Contrast between internally envisioned changes and system of simulations (SySi)

The main difference to point out, is that due to large number of available data many of these models do not actually need to provide analytical predictions. For instance, instead of predicting material properties using laminate analysis with complex adjustments, a lookup table with data for similar layups can be interrogated. This is probably the largest contrast between the novel process under development at the Company and SySi.

This is something that significantly simplifies the development of fully integrated system of simulations. One of the drawbacks identified during generation of SySi was the lack of reliable analytical methods predicting material properties, such as strength or permeability, from braiding parameters. The only methods that are reported to have a good reliability involve an FE model, which makes them unsuitable for iteration. Supplementing this with large amount of data and lookup tables will likely be sufficient for initial design iterations.

Another noticeable difference is the optimisation focus. SySi focused on the involvement of manufacturing simulations. The Company’s approach focuses mainly on integrating already existent core modules, and only later bringing in the secondary modules into optimisations. For example, the infusion simulation is considered of lower priority even for envisioned system. This stems from the customer requirements focused on aerodynamic efficiency. Therefore, this will likely differ based on the product in question. This is quite surprising considering the complex manufacturing methods employed at the company, where the material processing has likely high impact on the final product, especially the weight.

The utilization of existing designs for development of the integrated simulation chain is a major benefit of the in-house system. The optimisation loop and the connectivity between modules can be designed specifically so that iteration of the most important variables is supported.

The similarity in the two systems is that the development is somewhat disconnected from the actual manufacturing process. It will be difficult to replace the current design process standards with a completely new process. Replacing it one module at a time would be more practical, but might make it more problematic to demonstrate the benefits during the transition.

5.3.2 Modules of SySi that could support the company envisioned framework

This section discusses how modules of SySi can support the envisioned framework developed at the Company.

The kinematic braiding simulation used in SySi might be useful for the Company. Mainly it can be used to predict braid angles and pitch in any given region of the part. This could be coupled with existent data available within the company to provide decent estimate of material properties based on the manufacturing details. The visualisation of different braid options might also be useful.

The main improvements that are still required are an option for variable mandrel speed, and reliability over complex shapes. The variable mandrel speed could help with optimisation of design. Complex shapes sometimes cause errors to occur within current version of simulation, it is unclear if these shapes would be problematic for braiding or if the issues are purely within the simulation. This is due to the lack of practical manufacturing trials to verify the simulation.

The reverse solution, where mandrel speeds are provided based on required braid angles, should also be provided as an option. This would address practical requirements for setting up the braiding machine.

A method for using and testing braiding simulation outside of the simulation chain has been developed. It is unclear how the SySi braiding simulation compares to the current spreadsheet method used at the company.

The SQL storage of data is a reliable method for saving and retrieving parameters. It prevents pollution of data and allows a good, centralized control over the data. The methods used in SySi could be used as Python templates to develop an automated storage system with minimal effort. However, for each component type different tables are required. Therefore, some in house development would be required.

Similarly, Catia automation has to be bespoke for a component. Due to this, only the methodology of the CATIA automation can be taken from SySi, not the complete solution. However, it has some benefits over parametric models. For instance, shape type options can be implemented for optimisations.

5.3.3 Suggested implementation procedure

This section describes the potential methods for implementing SySi in an already running company such as the one reviewed. Generally, there are two main options to be considered. One is implementing the methodology from within. Senior management assign engineers already working for the company to develop set of methodologies analogous to SySi for their component or production line. The second one is external implementation. In this scenario the company hires external engineers to setup first iteration of SySi. The subcontractors would also provide manuals for further development and teach key personnel at the company how to use and develop the implemented system of simulations.

5.3.3.1 Option 1 – Internal implementation

This option largely benefits from the inherent internal knowledge of the specific problem. It is much easier to map out the important variables for engineers who have been working on similar products for the same customers for a while.

The most obvious downside of this is that during the initial stages of development engineers will have to perform their usual tasks and develop new methodologies on top of that. The only alternative would be to hire more people, but this would diminish the benefits of having internal knowledge of the problem.

Analogous to implementation of tools such as system of simulation today, is implementation of CAD/CAM systems few decades ago. From the implementation back then, various learning outcomes can be used. Introducing these solutions is easier from bottom up rather than the standard company change which comes top down (5). This is because the engineers can take ownership of the development/introduction. It is more important that the users believe in the technology than that the management does. This is possible with option 1, as all the changes are driven by the same engineers whose work it is supposed to support.

The likelihood of success of option 1 depends mainly on the buy-in of the engineers. If they believe the automation can result in better work outcomes with decreased effort this has a good chance of success. If the engineers are sceptical from the get-go, it will very likely fail.

5.3.3.2 Option 2 – Consultancy-based implementation

The option number 2 is an external implementation. This would consist of hiring sub-contractors, ideally from a specialized consultancy, to implement the new methodology. This is quite a standard approach for specific project work in aerospace. For instance, when developing a new aircraft, an aerospace company can expand the employee numbers for short term by hiring engineering subcontractors. This incurs minimal costs associated with returning to normal workload capacity afterwards.

This method could have varying degree of involvement. At the very least the sub-contractors would be required to develop methodologies every engineer in the company can follow, to create an automated version of their model. The overall methodology would also be developed for an integrator running the project. The opposite end of the involvement spectrum would be the sub-contractors developing the new system and then running a full example system and holistic optimisation for a new part. This could be even done in parallel of current work to make a complete comparison of the two methods.

The main benefit of the option 2 is that the current projects are not at risk due to the new process implementation. The downside is that significantly more investment is likely required to implement the new methodology.

The risk of the project failing is also decreased due to the implied experience of the sub-contractors in similar integration projects.

However, to the best of author’s knowledge a consultancy specializing in this particular work does not exist. This will be further discussed in the “potential business case” section.

5.3.3.3 Other variations

Both options can be implemented as a complete system, or systematically by individual modules. The modular approach allows for minimizing the risks involved, staggers investments, and prevents a project failure due to failed implementation. This is because the standard projects at the company can always use previous methods if a single module takes long to transition. On the other hand, the complete implementation of the procedure makes the final process more comprehensive and should be overall more efficient. Staggered implementation would cause some additional rework to the process, as each module implemented will be different when included in original workflow than when part of the complete new system.

Also, for companies that already work towards digital integration goals, it might be worth simply providing modules that support their envisioned framework.

5.3.3.4 Notes applicable for all options

Olsson et.al. (6) describe the move from agile to continuous development as an important topic in software development. Agile development is the attempt to not only respond to customer needs – agile, but also work with the customer to develop according to emerging needs. This is very appropriate for the topic discussed here as software development is involved and responding to customer needs is a major objective. Many parallels can be found between SySi and the software development described in the paper. In the case of SySi the developer works with the product, allowing for faster troubleshooting than would be the case in standard software development. As is true in either case, proper planning is required that allows for continuous development within the introduction plan/framework. Olsson at.al recommends that the new technology is implemented on incremental basis.(6)

This suggests it would be a good idea to focus on specific project, for initial implementation. Firstly, it allows for complete explanation of all the benefits to the people involved. Secondly, the segments that are more relevant for the given project can be prioritized, leading to incremental implementation.

The normal approach for companies is to buy commercial of the shelf software (COTS) and use it according to the manual. However, in advanced engineering this significantly limits the possibilities of the software. The software companies rarely permit any visibility into how their product is coded and how exactly it works. The suggested system of simulation would not only be different for each customer but would require significant amount of development from the engineers in any company that decides to use it.

Using open source software (OSS) might limit the size of the company, as large bureaucratic companies will be using COTS and will be against large implementation of OSS. On the other hand, large companies would probably be developing similar systems themselves and would have little use of the services provided (7).

The success or failure of first contribution to open source project typically defines individual’s willingness for further participation (8). When introducing engineers to system of simulation, it is therefore important that their first task is well documented and predictable.

5.3.4 Estimation of cost savings

Hard data regarding engineering work used on projects at the example company are not available. However, with the experience of developing SySi, author attempts to estimate the time required for standard process at the company and time for a version of the integrated system at the company.

Author estimates this by adjusting the development time required for the SySi demonstrator by the complexity of the company component. Four iterations over the design are considered as baseline. These are estimated to take roughly 33% of initial model generation time, adjusted by specific considerations.

Optimisation is not part of the original loop within company and therefore the optimisation development is not compared here. The optimisation deserves separate comparison where its benefits are compared to the extra development time. However, this would require very close visibility to part requirements and achieved part parameters.

This comparison deals with implementing SySi, with regards to the company requirements. There are many different combinations of ideas that would result in different comparisons. For instance, it could be argued that for the optimisation FE structural model is excessive. For simplicity the current version of SySi is used as default version.

The comparison is summarized in table 5.2.

Table 5.2 – engineering time required

Estimated time in hours	SySi	Example company part standard process (hrs)	Example company part – automated/integrated (hrs)
CAD model	2	24	28
CAD model iteration	0	8*3	0
CAD automation	16	0	24
Structural FE modelling	8	32	40
Structural FE iteration	0	12*3	0
Structural FE automation	32	0	48
Aero/performance model	N/A	32	36
Aero iteration	N/A	10*3	0
Aero/performance automation	N/A	0	32
Kinematic braiding simulation	40	48	48
Kinematic braiding iteration	0	0	0
Kinematic braiding simulation automation	0	0	0
RTM simulation	8	24	26
RTM iteration	0	0	0
RTM automation	24	0	28
Data management setup	8	4	12
Data management within iterations	0	6*3	0
Material property translation automation	16	0	20
Material property passing	0	6*4	0
Total	**154**	**296**	**342**

These estimates suggest it is not economic, based on only saving engineering time, to develop a version of SySi at the company. The total time required to achieve the final design with current process is estimated as 296 hours, while implementing the automated integrated system is estimated as 342 hours. The two main reasons are the low number of iterations currently done for each part, and the lack of iteration of manufacturing simulations.

However, this comparison does not consider engineering downtime, where engineers wait for others to finish their tasks. This aspect would be very difficult to quantify, as it will also depend on overlap of projects and the details of each specific task in the design process.

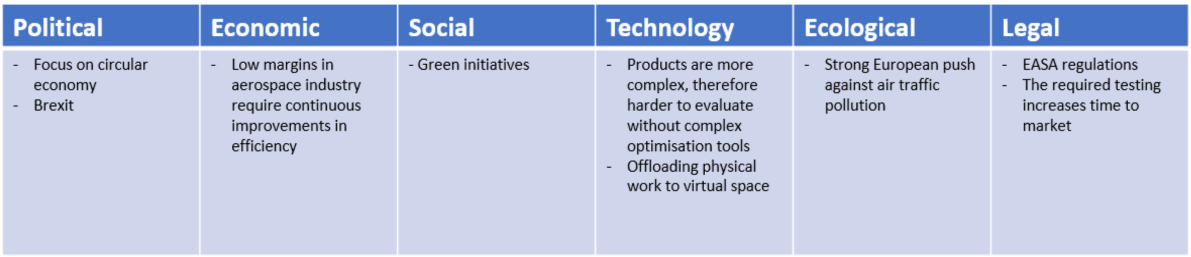
With a requirement for optimisation or for frequent post-production rework or design iteration, this would significantly improve. This is quite likely to be the case in the future at the company reviewed. Once the holistic optimisation is used in the company, it is also more likely that the braiding simulation will become more important. In such case the iteration of manufacturing would also be required. It is unlikely that the infusion simulation would ever need iteration as previous experience suggest infusions are rarely problematic.

5.3.5 Consultancy business case and strategy

As has been already outlined, one of the means of introducing this technology to the industry is through an engineering consultancy. One of the significant benefits of this method is that there are no requirements for engineer's scripting capability at the company. Since a consultancy specializing in this work does not exist yet, the business aspects of creating such a company are discussed here.

5.3.5.1 PESTEL analysis

To review the aerospace industry, the main operating space for the business, PESTEL analysis was undertaken. The take-outs in Figure 5.9 show the aerospace related aspects that relate to the proposed business.



Political	**Economic**	**Social**	**Technology**	**Ecological**	**Legal**
- Focus on circular economy - Brexit	- Low margins in aerospace industry require continuous improvements in efficiency	- Green initiatives	- Products are more complex, therefore harder to evaluate without complex optimisation tools - Offloading physical work to virtual space	- Strong European push against air traffic pollution	- EASA regulations - The required testing increases time to market

Figure 5.9 – PESTEL, points relevant to the consultancy business in aerospace industry

The focus on circular economy is specified in reports such as the ATI's "Technology and Strategy portfolio" (9–11). The system of simulation allows for better analysis of aspects relating to complete product lifecycle.

On the other hand, Brexit might limit any potential growth of Aerospace focused company outside of the UK.

The notoriously low profit margins in the aerospace industry force companies to strive for efficiency and minimization of costs wherever possible(12). The focus is rarely on minimization of excessive engineering rework, something that system of simulation aims to support.

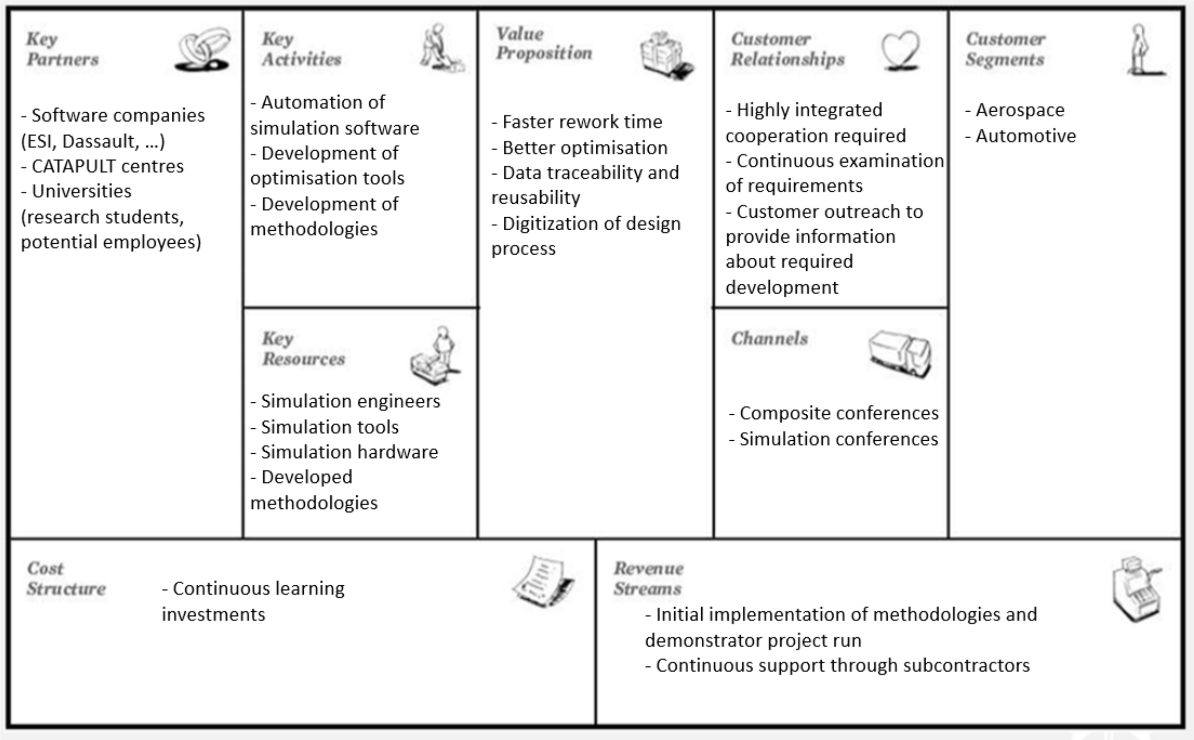
The technology, such as composite materials, make products increasingly complex. Therefore, it is becoming more difficult for even experienced engineers to fully assess quality of a product iteration. The proposed system of simulation can help assign numerical values to product iterations based on customer specification.

The structural optimisations inherently present in the Sysi will help minimize the weight of composite components. This will in turn decrease the fuel consumption of any aircraft using these components, decreasing the emissions.

The legal requirements for testing components form a significant part of the aircraft development costs. These costs can be minimized with modern tools by correctly reusing data from previous tests. This is heavily supported by the integrated data tracking and management within system of simulations.

5.3.5.2 The envisioned consultancy company

To give overview of the suggested consultancy business, a business model canvas was created, see Figure 5.10.



Key Partners

- Software companies (ESI, Dassault, …)
- CATAPULT centres
- Universities (research students, potential employees)

Key Activities

- Automation of simulation software
- Development of optimisation tools
- Development of methodologies

Key Resources

- Simulation engineers
- Simulation tools
- Simulation hardware
- Developed methodologies

Value Proposition

- Faster rework time
- Better optimisation
- Data traceability and reusability
- Digitization of design process

Customer Relationships

- Highly integrated cooperation required
- Continuous examination of requirements
- Customer outreach to provide information about required development

Channels

- Composite conferences
- Simulation conferences

Customer Segments

- Aerospace
- Automotive

Cost Structure

- Continuous learning investments

Revenue Streams

- Initial implementation of methodologies and demonstrator project run
- Continuous support through subcontractors

Figure 5.10 – business model canvas

The key partners are mainly the software companies. The cooperation with them will improve the service provided by the consultancy and will make the implementation of the system of simulations easier. It also allows for the feedback regarding the software used, which can lead to simulation improvements in general. Collaboration with catapult centres, and other research centres, could also prove beneficial. Firstly, the work usually given to catapult centres could be supported by any integrated simulation workflow. Also, the customers seeking improvements to their work through collaboration with CATAPULT centres, are likely to be interested in improvements to their design workflow. The collaboration with universities can also be useful. Research projects that would fill gaps in the workflow can be suggested. Finishing students can then be brought into the consultancy.

The simulation engineers are definitely the key resource. As these employees are required to continuously learn in a field where limited amount of resources are available to start with. Second key resource are the simulation tools. This not only includes licences to variety of commercial software, but also a library of currently available open source projects to use, or to develop on top of.

The key processes have to do with the simulation chain development, but also with identifying and categorizing customer needs to design the bespoke simulation systems. A strong systems-engineering methodology should be in place to aid with this.

The revenue of the consultancy comes from services connected to system of simulations. The main revenue should come from introducing the new methodology to the company. This would include the training of employees, the development of simulation modules and the introduction of the manuals and the methodologies bespoke for the company. This should be followed by additional revenue coming from retraining, maintenance and other support associated with further use of the implemented system.

The customer value proposition is two-fold. The engineering work to achieve certain level of optimisation can be decreased. The optimisation complexity can be increased, due to the simultaneous consideration of multiple design areas.

In summary it is key to maintain high level of specialised knowledge, and continuously improve the services available through learning on successful projects. The learning should be recorded in manuals to reduce the risk of losing business edge due to employee rotation.

5.3.5.3 The entered market

Porter's five forces diagram is used to evaluate the market position of the discussed consultancy, Figure 5.11.

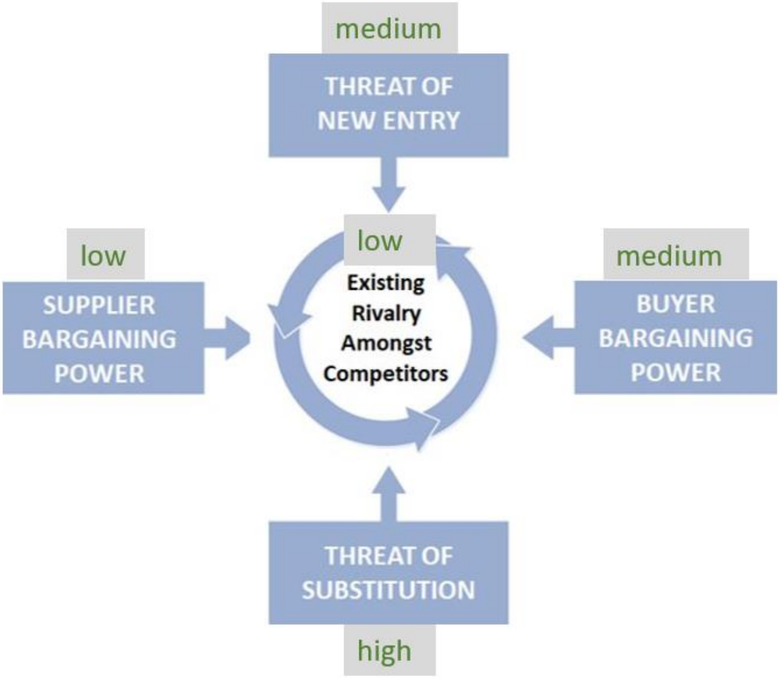


Figure 5.11 – Porter's five forces

The current rivalry in the market being entered is considered low, as no other company operates in this niche business area.

The threat of entry is medium; there are many consultancies that might want to enter, especially if the work turns out to be profitable. However, quite specialized skills are required for entry, therefore the investments to include this business in an existent portfolio might be too high. The main barriers to entry are the same for the envisioned consultancy as for others: lack of people with expertise in this type of work and lack of reference projects.

The buyers bargaining power is also medium. Although there is lack of direct competition, the companies can develop their own internal alternatives.

The threat of substitution is quite high as the software companies are already trying to develop a complete solution to the same problem. However, it is to some extend mitigated by the lack of cooperation between the software companies.

The supplier bargaining power is considered low. The only suppliers involved are the software companies. There is usually decent amount of competition among these. Also, once the consultancy has established itself with few projects, the software companies will significantly benefit from the consultancy using their software. This could even lead to free software being provided, with the agreement that the consultancy will promote this software with its customers.

The consultancy should wrap its initial strategy around being the innovators or first adopters. Developing experience early is key, as it is likely that similar methodologies are going to be attractive to more and more engineering design companies. It is crucial to already have wealth of experience when the market reaches its peak size. New customers are likely to select a consultancy with a track record, which provides a long-term revenue from follow up support. If a company joins this market late, it is quite likely that others will already have provided this kind of service. In this market it will be very unlikely that customers will switch away from the company who first provided the service.

The early stage of the venture lifecycle where revenue does not yet come in, but expenses exist is often called the "valley of death". In this case, this stage might be longer than usual due to the required initial projects that demonstrate the profitability of the simulation system to customers. The main benefit for a customer to employ you is your extensive experience, this is only true after implementing this at several companies. Hence there need to be additional incentives for initial customers, such as discounts, which might make the work less profitable. Risk sharing offers could also be provided as an alternative incentive. Risk sharing would allow for profitability earlier but increases the risk of complete failure of the venture.

The company's work is focused on setup of bespoke methodologies to improved design process at an engineering design company. Therefore, it is assumed to be unnecessarily costly and ineffective to try to protect IP with patents or similar. On the contrary, the successful examples should be publicly visible wherever customers allow, to support the intake of new customers in early stages of the consultancy's existence. The software components would probably be offered for collaboration as open source. This allows for more testing and development with minimal investment. The revenue is generated by the bespoke implementation at a company.

5.3.5.4 Market size

The market size for this consultancy is quite limited. The customer must design a product which requires high level of optimisation, or a significant amount of engineering rework. The manufacturing should be quite important part of the product specification. Composite components benefit from this more, but there is a chance that non-composite component manufacturers might also be interested. Majority of the design and simulation must be done in-house. The company must be willing and capable of amending their design practices. The customer must be willing to invest in process improvements.

The market size was assessed, although in a very crude manner. The main target would be aerospace companies due to high levels of optimisations and complexity of components. To review how many aerospace companies in the UK might become potential customers, a list from Wikipedia was used as a sample (13). Where company from the list no longer exists its successors was considered, where applicable. Out of the 44 companies reviewed 20 were considered as potential customers. Out of the 20, 2 are likely too large and would have developed their own alternative. However, this might still allow them to use some of the services of the envisioned consultancy. The list of aerospace companies is not exhaustive but likely lists the major players, the complete list of potential customers from aerospace industry is unlikely to be over 30.

A composites company specializing in complete bespoke design would also be a potential customer. The composites UK website lists 372 companies working in composites(14). 20 were randomly selected from the list to create a sample. These 20 were reviewed, only 1 was deemed a likely customer, with another 2 considered unlikely customers due to the size of the company, although their work fits the profile. Extrapolating, it can be estimated that there are around 19 companies in the UK that might become a customer. Other industries, such as Automotive or Wind might also provide some potential customers. However, the Aerospace companies, and Aerospace suppliers are more likely initial customers as these companies benefits from the integrated optimisations due to complex specifications. It would also be difficult for the author of this text to assess the likelihood of automotive manufacturers to require this system, having minimal knowledge of the typical automotive requirements.

It is difficult to assess how many of the companies identified as potential customers would actually be willing to consider the proposed design process change. Also, this is something that is likely to change as more engineers become well versed in automation, scripting and other digitization tools.

Generally, it appears that the market size is quite small and uncertain. On the other hand, it is unlikely that competition within the UK would likely spring up due to this fact. The potential customers identified might be sufficient for the establishment of the company. However, to make profit it is important to look for growth elsewhere. Based on the number of engineers employed in other European countries (15), it can be assumed that about the same customer base can be expected in France, about 100% larger in Germany, and about equivalent to that of Germany in the rest of the Europe. Of course, these are very crude estimates, not taking account the variability of engineering jobs and the different distribution of engineers per industry in differ countries. The main point is that the expansion to Europe should be considered as a precursor to success. It is difficult to assess all potential competition in that market, especially due to the time scales.

Mainly due to the market size, it seems the risks of starting such consultancy significantly outweigh the potential revenues. Therefore, the most likely implementation of this kind of work is through iterative expansion of capabilities of current engineering consultancies, or from internal isolated efforts.

5.3.7 Summary

The aim of this study was to establish the viability of introducing SySi to the industry. This was assessed based on discussions with engineers from an example company, that was selected as a potential user of the suggested methodology.

The discussion with the Company shed some light on what it would involve bringing SySi into the industry. On one hand the Company realizes the need to move in that direction, on the other it has its own ideas and strategies of how to get there.

The main difference between SySi and the envisioned system at the company is utilization of lookup tables to substitute missing material prediction models with data interpolation. Similarly to SySi, their current efforts are currently in the hands of an individual trying to demonstrate the benefits, rather than implementing a completely new design process.

In terms of introducing SySi to industry in general, two main methods were identified. One is from the inside, where the transition project is defined and run by existing employees. The other one is by outsourcing the transition project to a specialized consultancy style company.

The idea of a specialized consultancy was considered in greater detail. There appears to be clear demand from the industry, but it is unclear how many companies would be willing to entrust changes to their design methodologies to outsiders. This would likely be improved once a track record of successful implementation exists. Based on the example of the Company it is not sufficient for the value proposition to be centred around decreased engineering work, as the current number of iterations does not make that economic. Instead, attention should be paid to the benefits of the holistic optimisation facilitated by the new integrated system.

The technical development required for SySi will depend on the specific component in question. However, addition of aerodynamic considerations to the loop is a likely requirement.

References

1. Deutsch JL, Deutsch C V. Latin hypercube sampling with multidimensional uniformity. J Stat Plan Inference [Internet]. 2012 Mar 1 [cited 2019 Nov 19];142(3):763–72. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0378375811003776?via%3Dihub

2. Moza S. LHS-MDU [Internet]. GitHub repository; Available from: https://github.com/sahilm89/lhsmdu

3. Drela M. AVL [Internet]. MIT; Available from: http://web.mit.edu/drela/Public/web/avl/

4. Kelm B. AVL Tutorial (1-4) [Internet]. youtube.com; 2020. Available from: https://www.youtube.com/channel/UCSJMNygUkTkEsXzP3_SDiXQ

5. Esan AO, Khan MK, Naylor C, Qi HS. An integrated approach to Lean Systems and CADCAM methodology deployment in SMME. Univ Bradford Institutional Repos [Internet]. [cited 2020 Jan 29]; Available from: https://bradscholars.brad.ac.uk/bitstream/handle/10454/8022/fo-105-1.pdf?sequence=3&isAllowed=y

6. Olsson HH, Alahyari H, Bosch J. Climbing the Stairway to Heaven; A Mulitiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. In: 2012 38th Euromicro Conference on Software Engineering and Advanced Applications [Internet]. IEEE; 2012 [cited 2020 Jan 29]. p. 392–9. Available from: http://ieeexplore.ieee.org/document/6328180/

7. Holck J, Larsen MH, Pedersen MK. Managerial and Technical Barriers to the Adoption of Open Source Software. COST-Based Softw Syst [Internet]. [cited 2020 Jan 29]; Available from: https://link.springer.com/content/pdf/10.1007%2Fb105900.pdf

8. Steinmacher I, Graciotto Silva MA, Gerosa MA, Redmiles DF. A systematic literature review on the barriers faced by newcomers to open source software projects. Inf Softw Technol [Internet]. 2015 Mar [cited 2020 Jan 29];59:67–85. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0950584914002390

9. ATI. Technology strategy and portfolio update 2016. 2016.

10. ATI. New application process for UK Aerospace Research and Technology Funding - Aerospace Technology Institute [Internet]. 2018. [cited 2018 Apr 12]. Available from: http://www.ati.org.uk/2018/01/new-application-process-uk-aerospace-research-technology-funding/

11. ATI, Fathom. Raising Ambition for UK Industrial Strategy, Lessons from Innovation in Aerospace. 2016.

12. Top ten most profitable companies in the aerospace industry by pre-tax profit margin 1999/2000. Aircr Eng Aerosp Technol [Internet]. 2002;74. Available from: https://www.emerald.com/insight/content/doi/10.1108/aeat.2002.12774aaf.005/full/html

13. Category:Aerospace companies of the United Kingdom. In: wikipedia [Internet]. 2017. Available from: https://en.wikipedia.org/wiki/Category:Aerospace_companies_of_the_United_Kingdom

14. CompositesUK, current members [Internet]. [cited 2020 Dec 4]. Available from: https://compositesuk.co.uk/join-us/current-members

15. Manske HD, Erdmann V. European Engineering Report [Internet]. 2009. Available from: https://engineerseurope.eu/sites/default/files/European_Engineering_Report.pdf

16. Lomov S., Verpoest I, Cichosz J, Hahn C, Ivanov DS, Verleye B. Open Data Formats and Scripting in Integrated Meso-Level Textile Composite Simulations. 19 TH Int Conf Compos Mater [Internet]. 2014 [cited 2019 May 1]; Available from: https://www.mtm.kuleuven.be/Onderzoek/Composites/Research/meso-macro/textile_composites_map/textile_modelling/downloads/textile-modelling-paper-2014.pdf

17. Endruweit A, Long AC. A model for the in-plane permeability of triaxially braided reinforcements. Compos Part A Appl Sci Manuf [Internet]. 2011 Feb 1 [cited 2019 Oct 25];42(2):165–72. Available from: https://www.sciencedirect.com/science/article/pii/S1359835X10002897

18. Swery EE, Hans T, Bultez M, Wijaya W, Kelly P, Hinterhölzl R, et al. Complete simulation process chain for the manufacturing of braided composite parts. Compos Part A Appl Sci Manuf [Internet]. 2017 Nov [cited 2018 Jul 30];102:378–90. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1359835X17303111

6. Conclusions

A novel methodology for design of composite components was developed. It consists of the system of simulations (SySi) connected by Python scripts and supported by an SQL database. This approach was motivated by the challenges facing the current sequential design process, which suffers from disconnectedness of departments and the related re-work and optimisation difficulties. The envisioned system decreases the re-work engineering time and allows for holistic optimisations, by considering the propagation of variables through the design process.

A demonstrator system has been developed, with an example braided UAV spar to showcase SySi. The part was selected for relative simplicity while retaining some of the complexity associated with structural composite components.

The main contributions are as follows:

1. **CAD Automation**: A CAD module was successfully automated, allowing for relatively painless development of automatically generated CAD parts within CATIA. Automating CAD can bring standalone benefits, such as fast prototyping with visualization. The CAD automation also supported initial method for mesh generation. However, the CAD based mesh generation proved to be inefficient for large number of iterations. Therefore, a numerical meshing method using only Python was later developed, improving the meshing algorithm, and simplifying the follow up braid simulation.
2. **Braiding simulation**: The braiding simulation module had to be developed from scratch, as no open-source or commercial software that would suit SySi was readily available. The development was also improved by moving away from CATIA based simulation to completely numerical one, with only visualisation available in CATIA. The simulation lacks proper verification against a physical prototype, but was compared to kinematic braiding simulation in CATIA CBX. The comparison provides sufficient results for the demonstrator. The braid angles and pitch of fibres is outputted for each fell point, allowing for highly localized material property predictions. The currently used material properties predictions are crude but can be easily replaced by alternative methods if available.
3. **Linking aerodynamics to composite design for manufacture**: The most recently developed module is the aerodynamic analysis in AVL. This has been developed primarily to inform the load distribution used in structural analysis. It could potentially be used to form other part of the fitness function, to consider a requirement for lift/drag ratio or similar.
4. **Integration of structural analysis**: The structural analysis module takes into account the material properties and works with the shape imported from CAD. The latest version also

used the lift distribution input. Although structural analysis is often automated, seamless integration with other modules of design is still very rare. Abaqus software was used; it is a software commonly used for analysis of composite components and it can be scripted with Python. The development of Abaqus automation benefited from high quality documentation, and large community of user’s collaboration through online forums, which was especially useful for post-processing automation.

5. **Scripting PAM-RTM**: During a 3-month visit to Purdue CMSC, a PAM-RTM infusion module was also developed, with complete automation. PAM-RTM was selected as it can be automated with Python, although there is no documentation for this and there is no community of people to learn from, as it is the case for Abaqus. The whole process of setting up the simulation and processing the results was automated and implemented into the SySi. Due to the simplicity of the part, inclusion of infusion simulation in the core loop proved non-essential. However, the automation of PAM-RTM is highly useful even on its own for optimising more complex components, such as a large wind turbine blade, or mass manufactured propellers.
6. **Iteration and optimisation through complete design was achieved**.

   To facilitate the iteration through the SySi a simple GUI was developed. There are several Python libraries that allow for quick development of GUIs, with good documentation. The GUI allows for easy specification of iterations and helps with visual representation of the inputs to the simulation. This greatly simplifies running of new iterations or continuing interrupted ones.

   Latin Hypercube Sampling method was crucial when running the SySi for any combination of parameters. It supports troubleshooting as many variable combinations can be tested quickly. It helps the user to understand the relationships between different variables, which would be especially useful for more complex components. Finally, it facilitates the optimisation by feeding into surrogate models.

   Optimisations without a surrogate model were first tried, including genetic algorithm, ant colony inspired optimisation, and few other algorithms available in open-source Python libraries. However, the common problem was the large number of iterations required, coupled with high runtimes. The runtimes were decreased significantly since the first version of SySi, but the problem still exists. Hence, surrogate modelling was used. This way the design space can be first sampled using LHS, already providing high value information. Then the already obtained data can be used to form a surrogate which can be used for optimisation.

   With the use of the surrogate model, optimisation was possible. After a variety of SySi improvements the surrogate model now predicts the fitness function for a combination of variables with reasonable accuracy. Typically, about 600 datapoints were used for surrogate generation and validation, 300 datapoints each. Due to the simplicity of the part, it was quite easy to see where the optimum is by just look at the plotted results of the sampling. This severely diminished the benefits of the following optimisation. With more complex parts this would likely be more useful.

It has been highly practical, that all the simulations in the chain were scripted by Python. It shows that the knowledge of one quite simple programming language is sufficient to link all the aspects of design. Similarly, the SQL is a robust data management tool, that allows for review of previous runs in very efficient manner. It is important to design the database well at the start of the project, but even with minimal experience of the author this proved to be a very minor issue.

A study of potentially industrial application has been undertaken. This was done in cooperation with a relevant aerospace company. The company considered in the industrial study did share similar motivation to develop an integrated conceptual design loop. One of the outcomes of the study was that the in-house iterative development is the most likely path towards implementation of such a system. This is mainly because these systems need to be largely bespoke, it is important for the people developing it to have an intimate knowledge of the problem and the part.

In general, many small automation tasks have been demonstrated that can easily be implemented into a commercial design process. With a stable data management these small automation related improvements can be slowly transitioned into a tool like SySi. It will probably take a while before such a system is planned for at the start of every design project. Initially there might be quite high chance of the project being discontinued (before any prototyping), at which point this would just cause higher investments. However, the costs of this system will likely decrease significantly, as more and more automation tools become involved in the design process. This will qualify engineers involved in the design to have the necessary skills for the development of the overall system.

The usefulness of the envisioned system depends significantly on the part details. The main question is: how much does the part in question benefits from the re-design and holistic optimisations brought by the system? The part also needs to a have high requirement for simulations, ideally both in terms of performance and manufacturing. Sufficient complexity is required, so that it is difficult for engineers to find perfect solutions without automated optimisation. However, the complexity must not be too high, as that exponentially increases the difficulty of designing the optimisation chain. Similarly, the shape also needs to be suitable, the variation of the shape must be predictable to some extent. Lastly, it needs to be a high value part as the time investment into creating this system is quite high. To give an idea of how suitable different parts are based on the criteria listed, table 6.1 was created.

Table 6.1 – suitability of example components for SySi

	Re-design	*Optimisation*	*Simulations*	*Complexity*	*Shape*	*Value*	*Total*
UAV spar	4	2	2	1	4	1	**14**
propeller	5	5	4	5	4	3	**26**
airliner wing	3	5	5	1	4	5	**23**
small aircraft wing	2	4	3	2	4	4	**19**
helicopter blade	4	5	5	4	4	4	**26**
complete aircraft	3	5	2	0	1	5	**16**
car pillars	2	3	2	3	2	2	**14**
car bumper	2	4	4	3	4	2	**19**
wind turbine blade	5	4	2	4	4	3	**22**
bicycle frame	4	2	1	4	1	2	**14**

Each of the parts was evaluated in terms of the criteria to give indication of suitability. All factors are considered equal weight, although in company's decision-making that might not be the case. Higher values mean more suitable. In terms of complexity, numbers can be low for the reasons of too low or too high complexity. The numbers that are caused by too high complexity are underlined. It clearly shows that propellers, wind turbines and wings are ideal problems. These are complex but with predictable geometry and require high amount of optimisation. There are many versions of these produced and they are quite high value and high volume. Some of the parts rated as too complex for suggested process are more suitable for systems developed elsewhere, for example the Lagrange Airbus project or Pacelab APD, both mentioned in literature review.

The demonstrator system developed in this study is available on GitHub, https://github.com/Ellutze/sysi. This library is mostly provided for inspiration, and possibly to take small aspects of the code and use these for other solutions. The overall solution, in the presented version, is only suitable for the specific part.

In summary, this thesis proposed an alternative designed process and suggested a potential path towards the development of the required system of simulations. The current barriers to implementation were identified, all of which are likely to be diminished with time.

7. Future Work

This section could include variety of smaller and larger pieces of work that would move the methodology presented closer to a widespread industrial application. This includes introduction of various bespoke modules, more robust test cases, expanding on the optimisation review, or implementing various improvements to the existent modules. In this section, the two areas considered to be the most crucial to enable industrial application, development, and funding are discussed.

7.1 Creating an industrial test case

At this stage all the baseline aspects of SySi have been demonstrated. It has been shown that vast variety of software can be connected, suitability of Python and SQL database has been demonstrated, and iterations through the whole system were proven possible. However, there are several limitations to the demonstrator part. The part is relatively simple, allowing for identifying ideal parameters by visual inspection of sampled design space. This limits the need for more complex optimisations, decreasing the incentives for SySi implementation. The part does not fit within larger product. The lack of commercial product makes the specification and requirements quite arbitrary. This projects into the objective function, which was highly simplistic for the studied part.

These limitations could be alleviated by a test case on actual industrial project. Part in industry will have clearer requirements, allowing for better definition of an objective function. The realistic objective function will have two aspects to it, hard constraints and iterable variables.

Hard constrains or requirements will come mainly from certification bodies, imposing mainly safety requirements. The optimiser would need to check that hard requirements are satisfied before computing the fitness function for an iteration. The minimization of weight and cost would probably be a key metric in the objective function, but it is likely that there would be many minor requirements present in the objective function such as manufacturing time. These objectives are likely going to be clear for engineers experienced in designing the particular component.

It is important that the product of industrial part is in line with suggested products in conclusion, table 6.1. Only when the part itself is highly suitable for SySi, can the benefits be clearly highlighted. The complexity must warrant optimisation, yet the shape parameters must be well understood. Wind-turbine blades, helicopter blades, or propellers would likely be ideal candidates. These products are also quite similar across multiple implementations, therefore only minor amendments to the SySi can make it serviceable for new products.

This will not only address the demonstration of the benefits of a SySi but will force a practical resolution of smaller problems. An example of this, is the material property prediction. Through the application of test pyramids, any company dealing with composites will have some amount of test data already available. Therefore, algorithms interpolating between available data can be implemented in SySi optimisation instead of analytical material prediction models. Once this is in place, it can be further improved by data from manufacturing back to the simulation, analysing the part as manufactured. It would require a library of defects, and corresponding material properties. This would allow for more practical simulation and would better inform scrap related decisions.

Also, other aspects of the design process such as version control, specification management, and change management are important aspects that will have to be addressed. These are better reviewed in industrial settings as many of the practical implications are difficult to imagine. This might also bring to light some of the positive externalities of the robust SQL management.

Other practical considerations include licencing of the different software used, and collaboration between different companies on a single design solution. These are potential problems that need to be resolved for industrial application of SySi.

Couple of projects that might be suitable for the application of SySi are on the horizon at the NCC. It will be crucial to manage expectations of the customers, when it comes to SySi, and to make sure that the development costs are shared for the first few projects, as some development is currently required.

7.1.2 Accomodating for uncertainty

The SySi in it’s current state does not account for uncertainty. Accomodating for uncertainty does make the optimisation through the toolchain likely too complex, with respect to the typically available computational resources and optimisation algorithms. However, there is a way to use an existent toolchain and known uncertainty to inform desing.

If sufficient testing is done to obtain material properties, a distribution of variables can created. A separate SQL table can be created to contain the defining parameters of the probabilistic distributions. The toolchain can then be used to iterate through design, according to the distribution. For instance, the best case scenario and worst case scenario can be computed, accounting for variations of multiple material properties. Also, similar to aircraft certification, probabilistic limits can be imposed. If the requirement is, that the part sustains certain loads in 99,9% of manufacturing scenarios, the appropriate threshold values can be obtained from the probabilistic distribution and the toolchain can be re-run.

Good quality and sufficient quantitity of data is required to make this possible. Also, toolchain should already exist, as in most cases it would be difficult to justify the development only to accommodate for uncertainty. This is why this future section is considered subsection of 7.1, as industrial test case would likely be required, and this might provide additional business case for it’s development.

7.2 Costing module

There are several modules that gain large benefit from the integrated optimisation. This is especially the case for modules which are affected by many different aspects of the product lifecycle, such as for example carbon footprint assessment. However, in reality carbon footprint optimisation is unlikely to incentivise additional investments in design process, but minimization of costs might. The business case is relatively simple, with additional module implemented in SySi the part design will take into account parts cost, hence improving profitability.

There are various sources of costs in design and manufacturing of a component. This is typically split into recurring and non-recurring costs. The non-recurring costs are going to be mainly dependent on the conceptual design stage; based on chosen manufacturing method or part configuration. A SySi type approach could be applied at concept design stage. However, it would be more difficult, as iteration of configuration and manufacturing method would have to be implemented, adding a significant layer of complexity.

Therefore, currently it can be assumed that costing module within SySi is more suitable to assess the recurring costs. The simplest aspect of this are the material costs, where the company’s procurement data can be used to feed into the calculation. The manufacturing costs are slightly more complex. The costs of personnel, machine runtime, energy, etc. need to all be considered here. This would likely work well with a manufacturing simulation which provides the manufacturing time for the part in question.

The costing module can include vast number of aspects, and it would depend on the company and product in question which would be considered important. Additional parameters would likely be brought in, and new interesting relationships could be explored. For example, an inspection interval in service could be brought in as a parameter. Varying this parameter would create a trade-off between having to impose larger structural safety factors to decrease inspections required. If the inspection is difficult, it might be worth causing additional degree of overdesign in the part. These, and many other, trade-offs could be evaluated with the introduction of the costing module. The details and benefits of costing module would depend on the application.

The downside of this module is the large reliance on high quality data. Therefore, it is probably a module that is better attempted in an industrial setting, nicely complimenting the idea of moving towards industry-based demonstrator. Without a good quality of costs data from many sources across the company, the reliability of results might not be sufficient for it to figure in the optimisation.

X. Appendix

X.1 Alternative volume fraction calculation

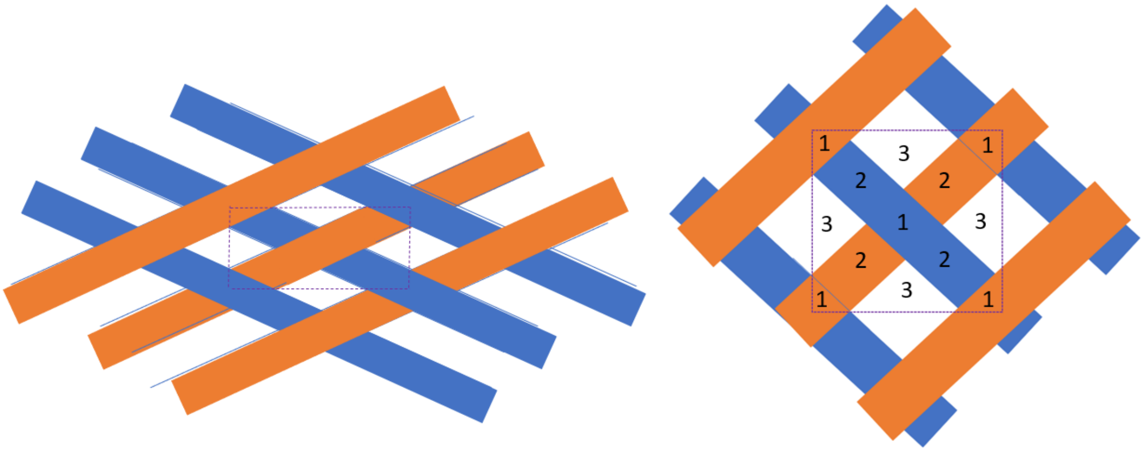


Figure x1 – unit cell for V_f *segmentation*

The volume fraction calculation is based on segmentation of unit cell of the braided preform. The dotted square and rectangle are the unit cells of each of the two pictures. The right-side picture shows default braid angles of 45 degrees and the numbered regions. For each region number a separate Vf calculation is provided. The left side just visualizes the change in the shape of unit cell with braid angle.

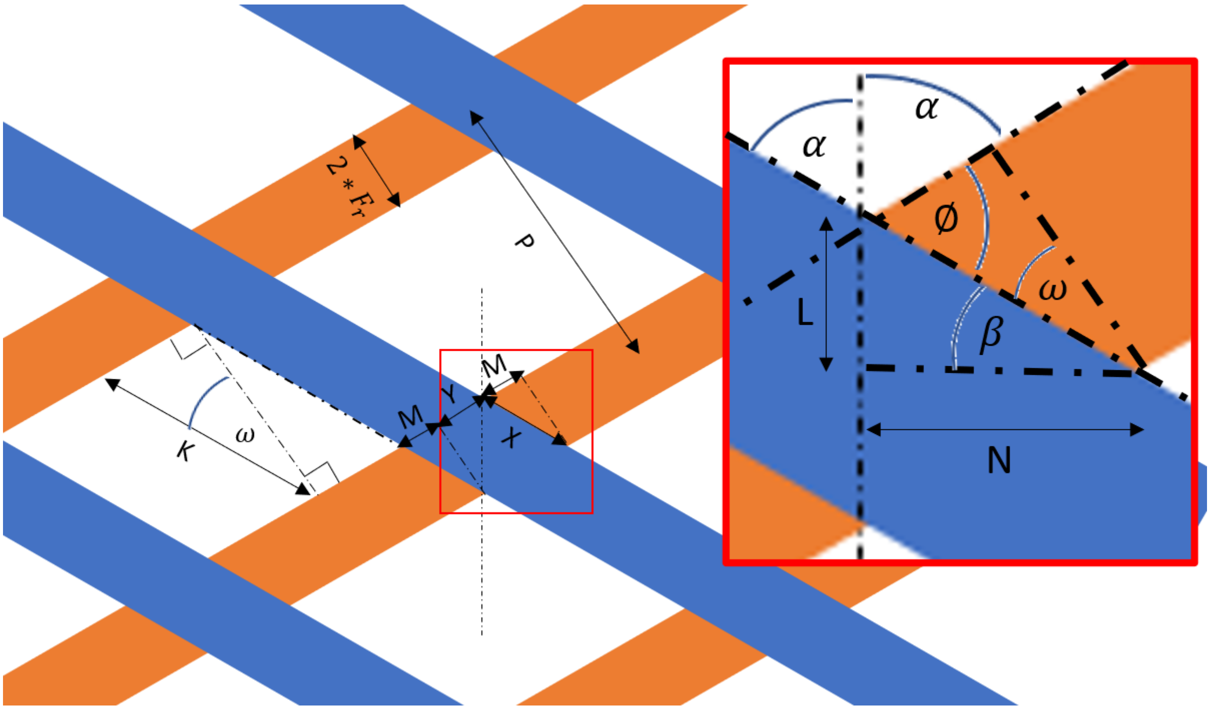


Figure x2 – calculation of V_f

Figure x2 shows the trigonometry used for the calculations below. P denotes the pitch between fibres, distance from the yarn centreline to the neighbouring yarn centreline. F_r is the fibre radius.

$$\emptyset = 180 - 2\alpha$$

$$\beta = 90 - \alpha$$

$$\omega = 90 - \emptyset$$

$$X = \frac{2F_r}{\cos(w)}$$

$$L = X\sin(\beta)$$

$$N = Xcos(\beta)$$

$$M = 2F_r \tan(\omega)$$

$$Y = \sqrt{(2L)^2 - (2F_r)^2}$$

$$A_1 = 4F_r(Y + M)$$

The A_1 corresponds to middle diamond and the 4 corners, which add to the same diamond. Hence the 4x multiplier, 2x multiplier corresponds to one diamond.

$$K = (P - 2F_r)/\cos(w)$$

$$A_2 = 8F_r K$$

$$H = 2P sin(\beta)$$

$$W = 2P cos(\beta)$$

Where W and H are the width and height of unit cell respectively.

$$A_{tot} = W * H$$

$$A_3 = A_{tot} - A_1 - A_2$$

Later F_r in all above has bee replaced by b to account for elliptical yarn. An arbitrary ratio for the elliptical yarn was chosen. This can be adjusted if data about the typical ellipticity of yarn is obtained.

To maintain the same cross section area, new thickness of the yarn has to be calculated, n_t. Same as b, t_n replaces radius, full thickness of yarn being 2*t_n.

$$YarnArea = \pi * Fr^2$$

$$t_n = YarnArea/(\pi b)$$

$$V_{f1} = YarnArea/(4b * t_n)$$

D is the distance taken of yarn through diagonal of open space in A2.

$$D = \sqrt{(P - 2b)^2 + t_n^2}$$

$$V_{f2} = (\pi D b t_n)/(4A_2 t_n)$$

$$V_{f3} = 0$$

$$V_f = \left(\frac{A_3}{A_{tot}}\right) * V_{f3} + \left(\frac{A_2}{A_{tot}}\right) * V_{f2} + \left(\frac{A_1}{A_{tot}}\right) * V_{f1}$$

Because this method is only used to obtain volume fraction, an asymmetric braid angles of warp and weft yarns are treated as symmetric through averaging the braid angles. For volume fraction calculations 35 degrees and 65 degrees are the same, the diagram has been created for above 45 degrees, hence lower angles than 45 are translated to their above 90 equivalents.

If the pitch is smaller than 2b this method currently doesn't work, as nesting has not been addressed.

X.2 Scripts list

This is an extract of the script list that was mostly maintained throughout the development. In the full list additional information was stored: date started, storage location of latest script (before transition to GIT), annotations (yes/no), active in core loop (yes/no), related scripts, etc. This was quite important to manage the development. In industry management of scripts and system versions would be absolutely crucial.

Now this list can be used for navigation, when reviewing the GitHub repository. Portions of scripts can be extracted and reused if necessary.

Table X.1 – Scripts list

	SCRIPT NAME	**WORK-PACKAGE**	**DESCRIPTION**
1	catia_XX001	CATIA	initial CATIA wing iteration
2	catia	CATIA	current version of CATIA integration
3	abaqus_inst	Abaqus	passes instructions to Abaqus upon being triggered from Abaqus
4	abaqus_cmd	Abaqus	operates Abaqus through command line
5	optis	management	looping the established simulations
6	MySQL_utils	MySQL	small MySQL tasks such as storing sim data
7	python_mysql_dbconfig	MySQL	reads configuration file
8	abaqus_intro	Abaqus	initial trial script, use from Abaqus
9	catia_utils	CATIA	general repeatable CATIA utilities: parameter creation and change, measure of a line/curve…
10	AnyInput	SQL server	imports any standardised testing data, creates tables etc. as well
11	utilities	general	general repeatable tasks
12	excursor	SQL server	creates connection with SQL server
13	SQLserverconfig	SQL server	reads configuration file
14	MySQL_importing	MySQL	importing data to MySQL
15	general_utils	general	general repeatable tasks (airfoil…)

16	SQLS_ASTDM3039	SQL server	relevant to post-import-processing of ASTDM3039 data
17	SQLS_ASTDM6415	SQL server	relevant to post-import-processing of ASTDM6415 data
18	SQL_utilities	SQL server	general SQL server utilities, such as storing sim. Data
19	cmd_utils	general	command line utilities
20	catia_mesh	CATIA	creates a mesh in Catia that can later be used in Abaqus (or other)
21	maths_mesh	Pure mesh	(under development - meshing based maths only)
22	LOOPS	management	(not made yet, list of optimisation loops including different combination of software…)
23	abaqus_postProc	Abaqus	processing ODB file
24	catia_mesh_OLD	CATIA	older mesh generation, including curved surfaces, to be further meshed elsewhere
25	vecEX	CATIA	exports vector values for line in CATIA
26	Braid_setup	Braiding	creates centreline and other braiding related geometries
27	Braid_main	Braiding	Simulation modules
28	Braid_proc	Braiding	Collection of functions to be used for post-processing of braiding simulation
29	Braid_visuals	Braiding	collection of visual representations for the braiding simulation
30	Braid_SQL	Braiding	Collects output from the braiding simulation
31	Braid_CMD	Braiding	master script for braiding sim
32	vecEX2	CATIA	exports vector values for line in CATIA
33	abaqus_main	Abaqus	passes Abaqus scripts, stores and retrieves data
34	lam_tools	general	laminate analysis tools to obtain material properties from braiding info
35	Segmentation	general	pairs calculated material values with segments of the part

36	cnn_main	general	bridge between braiding and FE, uses lam_tools and segmentation
37	practical	miscellaneous	used for practical purposes, e.g. running simulation only during off hours
38	math_utils	general	math functions, eg. Coordinate translations
39	MASTER	general	runs the full loop, to be replaced by the Dojo/Ribosomes/Helix…
40	IDP_agents	general	Major optimisation tools, GA,ACO
41	IDP_assistants	general	Supporting optimisation tools
42	LHS	general	Latin hypercube sampling, standalone
43	IDP_GUI	general	creating user interface for the loop
44	MySQL_setup	MySQL	creates the MySQL structure on a new machine/database
45	RTM_toolbox	RTM	main set of PAM-RTM automation tools
46	RTM_postProc	RTM	obtaining results from the PAM-RTM results files
47	RTM_cleanup	RTM	moves files after the simulation run into bin/archive
48	RTM_cmd	RTM	uses command line to pass scripts to , PAM-RTM, same as abaqus_cmd
49	RTM_surfaces	RTM	uses .inp files to find approximate position of imported surfaces
50	RTM_main	RTM	main script running the commands for RTM sim.
51	RTM_PPcmd	RTM	the post processing scripts that require cmd line interaction
52	RTM_WB_iterations	RTM	separate demonstration of RTM iteration
53	RTM_WB_inlet	RTM	the visual-RTM script supporting the above
54	RTM_WB_nodegen	RTM	the initial node collection supporting the above
55	RTM_cmd2	RTM	adjusted RTM_cmd for better subprocess control, used by all except PostProc to avoid circular imports

56	RTM_lil_toolbox	RTM	only the sections of toolbox relevant to flow-rate adjustment, used for looping
57	RTM_package_check	RTM	Imports required Python libraries, to test installation of all outside modules
58	IDP_D3_main	RTM	runs Demo no.3, adjusting temperature of unit 5 panel with difficult resin
59	IDP_D3_temp	RTM	is run through cmd_2, adjusts temperature of the panel and re-runs sim
60	plotD3	RTM	bokeh plot the results
61	IDP_cheats	general	workarounds, eg. toggle VPN by mouse manipulation
62	IDP_databases	general	establishes connection with appropriate database (because I switch between them way too often)
63	Bambi_run	RTM	initiates Bambi related automation
64	Bambi_main	RTM	NCC project (BAMBI) to use my automation,
65	default_var_dict	general	gives default values to all variables
66	IDP_install	general	allows for installation of scripts on new machine, link renaming, folder creation…
67	numimesh	Mesh	entirely numerical meshing script to provide .inp file to abaqus
68	IDP_inpGen	Mesh	creates .inp file from 3d matrix of points
69	IDP_geometry	braiding	analytically obtained vector from surface defined by point cloud
70	Braid_main_P	braiding	purely numerical braiding simulation main calculations
71	Bradi_cmd_P	braiding	initiation of purely numerical braiding simulation
72	data_proc_utils	optimisations	used to collect data from various tables, provides alternative fitness functions etc,
73	SVR_testing	optimisations	testing SVR and other sci-kit learn modules
74	DiffEvo	optimisations	testing differential evolution
75	plot_DE	optimisations	plotting the results of differential evolution

76	pathing	general	used to adjust location references for non-folder based python scripts (Abaqus, PAM-RTM)
77	AVL_inputs	AVL	adjusts the input files required for AVL analysis
78	AVL_postProc	AVL	string manipulations to extract information out of AVL output
79	avl_cmd	AVL	controls the AVL module, run from master
80	Braid_main_S	braiding	initial separated braiding simulation
81	Bradi_cmd_S	braiding	initial separated braiding simulation
82	OtherAlgo_testing	optimisations	version of DiffEvo
83	braid_CATIA	braiding	used to braid general CATIA shape, separate repo
84	braid_check	braiding	visualization of yarns in CATIA to check simulation
85	braid_data	braiding	data management for standalone braiding simulation
86	mesh_anyshape	braiding	meshing for separated braiding simulation
87	settings_GUI	general	allows for selecting settings (eg. Software) for SySi, not used much

X.3 Optimisation loop flow chart

Figure X3 on next page presents a flow chart that visualizes the optimisation. The flow chart does not reflect all additions to the loop but outlines the core scripts that were used for the initial optimisation. The core loop has not changed too much since its inception.

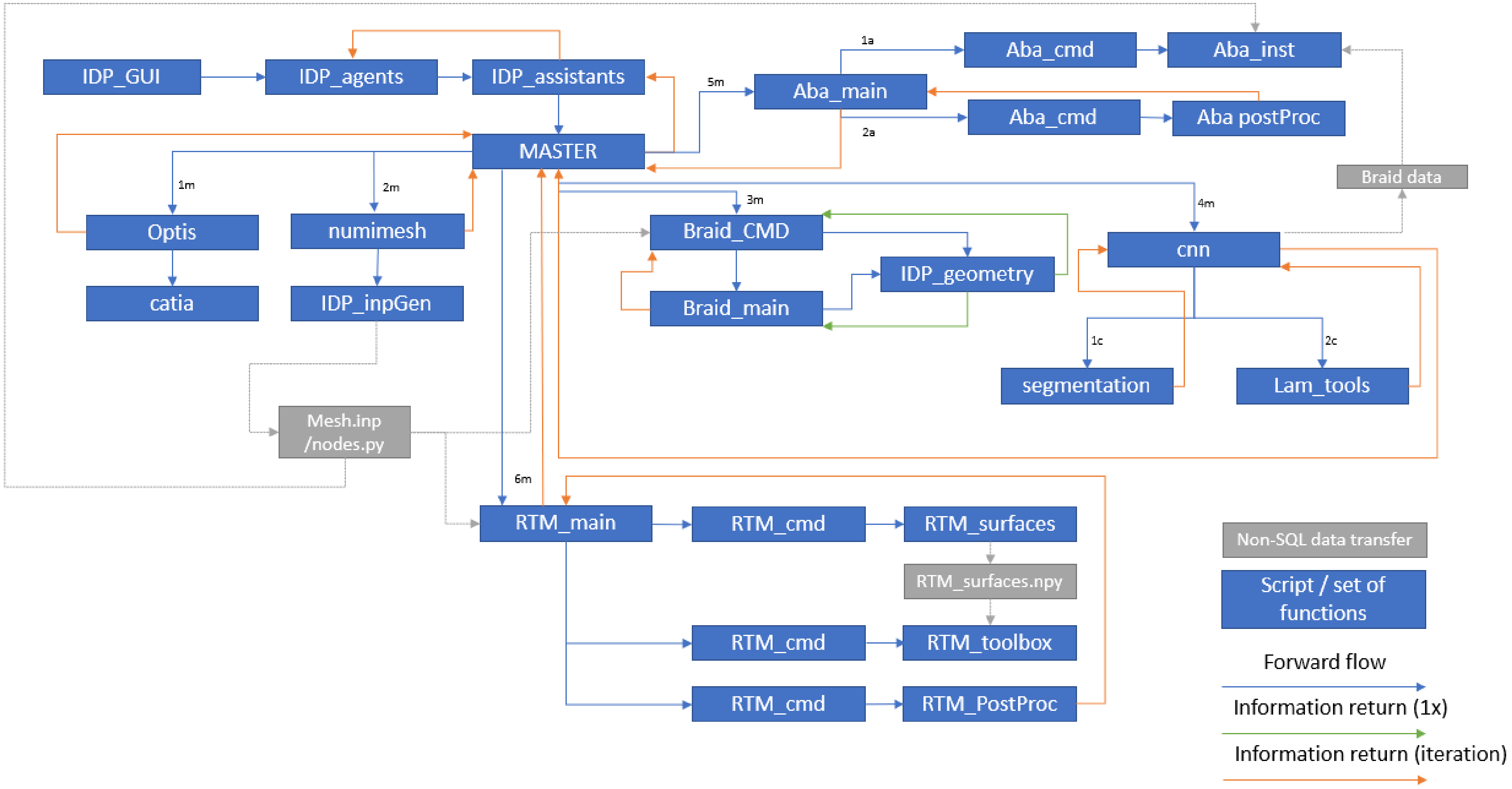


Figure X3 – Optimisation loop flow chart, highlighting core scripts

X.4 Optimising CATIA braiding simulation

As outlined in chapter 3, braiding simulation based on CATIA geometry suffered from large runtimes. To minimize these CATIA was closed after certain number of loops. Figure X4 shows the effect the closing had on runtime.

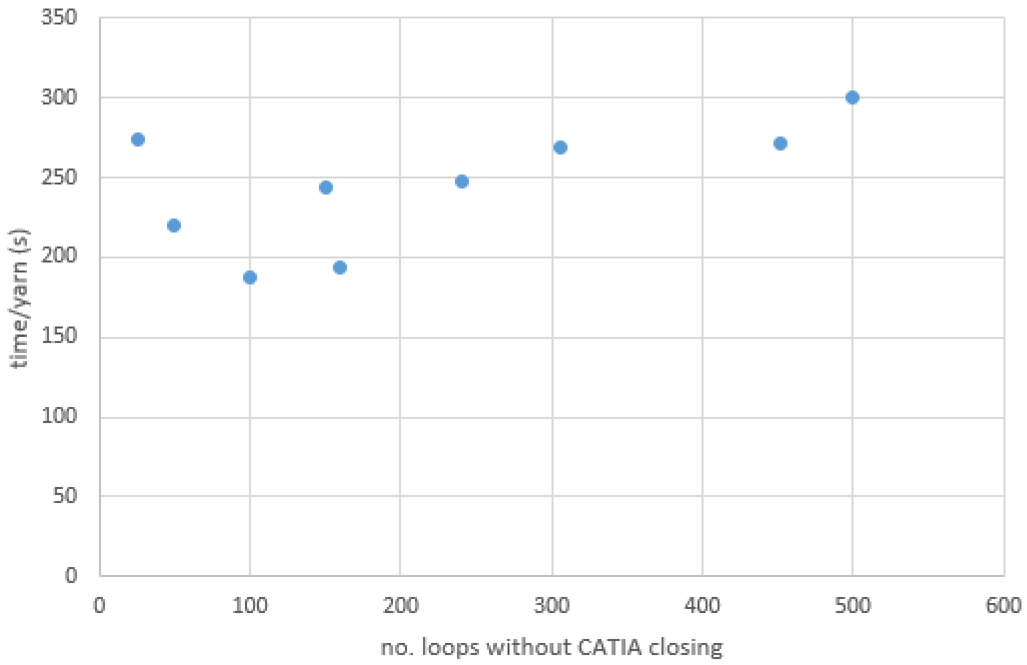


Figure X4 – optimising CATIA braiding simulation

Not closing resulted in largest runtimes, while closing CATIA too much suffered from too high restart time. The ideal number of iterations for the tested part was about 100. It is likely that this will be similar for other parts as well, as this is based on the restart times and number of geometrical features in CATIA.