|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Team 40 | Name | st. number | Name | st. number |
| S. Nolet | 4535677 | H.S. van Manen | 4549465 |
| Generation of a KBE app to support the design of automotive rear spoilers | | | | |
| Describe briefly your design case and make sure to address the following crucial questions:   * What is the design challenge your KBE app is supposed to tackle? * Why the use of KBE is supposed to be a good means to address such challenge? Thus, what are the characteristics of the problem at hand that match the strengths of KBE technology?     *The KBE app is designed to create a rear spoiler for a sports car. As the amount of downforce that is created is largely dependent on geometry, the use of KBE knowledge can benefit fast design changes that are necessary in the design process. This way, the amount of downforce and the amount of drag of the spoiler can be assessed for different designs and operating conditions.*  *There are two options for the use case of the application:*   1. *The user can specify a total planform geometry of a rear spoiler and calculate the downforce and the drag force.* 2. *The user can specify a desired downforce, together with a partial planform of a rear spoiler. For instance omitting the incidence angle, and consequently walking through an array of incidence angles to calculate at which angle the desired amount of downforce is obtained. Subsequently also calculation the drag of the spoiler.* | | | | |

|  |
| --- |
| Rule based parametric model requirements |
| Describe here the main functionalities of the rule based parametric model, which will be the core of your KBE application   * What systems components/features will be included in the parametric model? * What are the main parameters used to define and control the various components/features? * What are the main (engineering) rules governing the definition/interface of the various model components? Identify the main sources you will use to capture knowledge * How will your app deal with rules violation? (warnings, automatic corrections, change suggestions)   *The KBE application will focus on the generation of an automotive rear spoiler. Such a spoiler consists of the* ***main spoiler plate*** *and its* ***struts*** *with which it would be attached to a car. Additionally, it can be chosen to also put* ***endplates*** *to the tips of the spoiler. All components will be modelled as having the same material, in order to give weight estimations and perform stress calculations. For this reason, the user must specify which type of material will be used. This can either be done from a database in which several materials will be predescribed, or the user can implement several crucial material properties of a different material.*  *The* ***main spoiler plate*** *will be modelled as an upside down wing as its main functionality is to provide additional downforce on the car. It will be defined from the centerline of the car to one tip of the spoiler, after which it will be mirrored to get the full span. For the definition of the half-span, at least two sections have to be defined (one on the centerline and one on the tip, more sections can be defined in between these points). These sections will be interpolated to get to the spoiler geometry. The parameters that influence the geometry of the spoiler plate are:*   * *airfoil type (fixed for the whole spoiler)* * *span* * *[x, y, z] coordinate of the leading edge (per section)* * *chord (per section)* * *thickness to chord distribution (per section)* * *incidence angle (per section)*   *The* ***struts*** *are generated to fix the spoiler onto any car. It is chosen to fix the amount of struts to 2, as this is most common in automotive design. The struts will be modelled as thin plates, since it is desired to have a minimum drag penalty. The strut will be based on the interpolation of two sections: the point of application to the spoiler and the point of application to the car. Again, only one strut is modelled, after which it will be mirrored around the centerline of the spoiler to receive the second strut. The parameters that influence the geometry of the strut are:*   * *strut lateral location as percentage of the half-span of the spoiler* * *chord (per section)* * *height* * *sweepback of leading edge* * *cant angle (angle around the x-axis)*   *It is optional to fix* ***endplates*** *to the tips of the spoiler. This might be desirable to lower the drag of the spoiler, as the endplates resemble winglets. These endplates will be modelled as flat plates with a variable geometry. Once again only one of the endplates will be defined after which it will be mirrored about the centerline. The parameters that influence the geometry of the endplates are:*   * *point of application to the spoiler (defined as percentage of height)* * *height* * *chord* * *thickness* * *sweepback* * *cant angle*   **Main engineering rules:**   * *Aerodynamics calculated based on the vortex lattice method. Simple first-order estimation of the aerodynamic forces based on the design of the spoiler.* * *Weight estimations and structural calculations (deformation, shear stresses) based on structural analysis literature taught in the bachelor.* * *Specific spoiler design methods will be retrieved from automotive literature and sources found on the internet*   **Rules violations:**   * *In the case when a specific amount of downforce is given and it is not possible to achieve this (referring to option 2 of the application), an error message will pop up saying that the requirement cannot be fulfilled with the current design.* * *In the case it is physically not possible to create the geometry of the spoiler, for example when the distance between the struts is larger than the spoiler span, it will automatically be corrected to the maximum value that is possible. In the case of the struts, the distance between the struts will be set equal to the spoiler span.* * *When the forces in the spoiler become too high and the structural analysis determines that the structure will break, the application will warn the user and ask if they want to lower the amount of downforce or change the material.* |
| Internal analysis capabilities |
| * What analysis modules will be implemented **inside** your KBE application, thus coded in ParaPy. At least one internal analysis module should be present.   **Internal analysis modules:**   * *The application will be able to calculate the reference area and total wetted area of each of the mentioned components for the inputted spoiler geometry.* * *The application will be able to calculate the stresses and deflections in the structure based on the force distribution that is provided by AVL. This is done by assuming the spoiler can be modelled as a beam and using the Euler-Bernoulli beam theory.* * *The application will be able to make a weight estimation for the inputted spoiler geometry. The material properties are used to determine the different part weights and the application computes the total weight.* * *The application will provide a first-order estimate of the additional normal force that is exerted on the wings. Input for this is the longitudinal position of the wheel axes and the longitudinal position of the spoiler.* * *Based on the additional normal force on the wheels, the application will provide a first-order estimate of the additional friction drag of the wheels. Together with the drag of the spoiler, the total amount of extra drag can be calculated by the application.* |
| Link(s) with external analysis module(s) |
| * What **external** analysis module will be connected to your KBE app? How will your app interact with such applications? At least one external analysis tool/module should be present.   ***Link with AVL***  *The application will have an automatic link with the open source vortex lattice code AVL. The application will be able to implement the 3D spoiler geometry into the AVL code, in order to:*   * *calculate the spanwise lift and drag distributions.* * *calculate the total lift force.* * *calculate the total drag force.*   *The application will extract the calculation results from AVL and will be able to use these in further internal calculations.*  ***Link with XFOIL***  *The application will have an automatic link with XFOIL. The application will let the user specify a spanwise position along the span of the spoiler plate for which the XFOIL code will output the sectional Cl-alpha plot. This can be useful to calculate a new incidence angle if the current design gives too much/too little downforce.* |

|  |
| --- |
| Input data handling capabilities |
| * What data sets will be provided as input to your KBE app? * In which format will the input data sets be defined? * Which data (sub)set will be interactively editable in the ParaPy GUI?   *The* ***design parameters*** *will be first implemented as a list of design inputs for the different components as described above (airfoil, incidence angle, etc.). The format in which the data will be implemented will be either two or three DAT-files (depending on whether the spoiler has endplates or not), which contains the geometry parameters of each component.*  *The* ***external flow conditions*** *will also be implemented as a list. This list will contain all parameters of influence on the flow (velocity, temperature, etc.). Again, the format for the input will be a DAT-file.*  *The list of design variables will be interactively editable in ParaPy. This means that in the ParaPy GUI, for instance the initial design span, chord, etc. can be changed and automatically updated. From this,new values of the downforce and drag can be calculated.*  *From the flow conditions data, the velocity of the free-stream, which is equivalent to the car speed, can be altered inside ParaPy. This can be done to quickly calculate the forces on the spoiler in ‘off-design’ conditions.* |
| Output data reporting capabilities |
| What output files is the KBE app supposed to generate and in what format?  At least one STEP(or IGES) file and one output file containing results from the analysis modules.  *The main outputs of the application are performance and geometry data:*   * *It will export DAT-files from AVL with both spanwise and chordwise force distribution. Furthermore, it will export a file that gives the total values of the spoiler (downforce, drag)* * *It can export Cl-alpha XFOIL data as both a raw DAT-file or a plot (using matplotlib). The user can specify which section they want to output.* * *The tool will generate a PDF-file containing a top, front and side view of the 3D geometry. This will be in the format of a technical drawing* * *The tool will export a STEP-file in order to visualise the spoiler configuration in CATIA (or other CAD programs).* * *The stress distribution in the spoiler and the deflection can also be outputted. The application can give a short summary of the most critical section and the highest deflections.* |