SUAVE: An Open-Source Environment for Multi-Fidelity Conceptual Vehicle Design

Trent Lukaczyk[*], Andrew D. Wendorff[†],

Emilio Botero[‡], Timothy MacDonald[‡], Timothy Momose[‡], Anil Variyar[‡], J. Michael Vegh[‡],
Michael Colonno[§], Thomas D. Economon[¶], and Juan J. Alonso[‖]

Stanford University, Stanford, CA 94305, USA

Tarik H. Orra,[**] Carlos Ilario da Silva[††]

EMBRAER, São José dos Campos, SP, 12277-901, Brazil.

SUAVE, a conceptual level aircraft design environment, incorporates multiple information sources to analyze unconventional configurations. Developing the capability of producing credible conceptual level design conclusions for futuristic aircraft with advanced technologies is a primary directive. Many software tools for aircraft conceptual design rely upon empirical correlations and other handbook approximations. SUAVE proposes a way to design aircraft featuring advanced technologies by augmenting relevant correlations with physics-based methods.

SUAVE is constructed as a modular set of analysis tools written compactly and evaluated with minimal programming effort. Additional capabilities can be incorporated using extensible interfaces and prototyped with a top-level script. The flexibility of the environment allows the creation of arbitrary mission profiles, unconventional propulsion networks, and right-fidelity at right-time discipline analyses.

This article will first explain how SUAVE's analysis capabilities are organized to enable flexibility. Then, it will summarize the analysis strategies for the various disciplines required to evaluate a mission. Of particular interest will be the construction of unconventional energy networks necessary to evaluate configurations such as hybrid-electric commercial transports and solar-electric unmanned aerial vehicles (UAVs). Finally, verification and validation studies will be presented to demonstrate the capabilities of SUAVE, including cases for conventional and unconventional vehicles. While some of these cases will be optimized results, discussion of SUAVE's interface with optimization will be reserved for a future publication.

I.   Introduction

Surveying the conceptual design landscape of tools used in creating new aerospace vehicle configurations, there appears a need for analysis capabilities for non-tube-and-wing configurations. Many current software tools for aircraft conceptual design, such as PASS, EDS, TASOPT, and FLOPS, rely upon empirical correlations and other low-fidelity approximations for the propulsion and power system as well as the architecture of the airframe. In most cases, the propulsion system is modeled using traditional jet fuel and the airframe is modeled with a fuselage and wing, or in some cases certain types of blended wing-bodies. On the other hand, designing unconventional aircraft configurations integrated with game-changing technologies will require the

---

[*]Graduate Student, Department of Aeronautics and Astronautics, AIAA Student Member.
[†]Graduate Student, Department of Aeronautics and Astronautics, AIAA Student Member.
[‡]Graduate Students (authors in alphabetical order), Department of Aeronautics and Astronautics, AIAA Student Members.
[§]Engineering Research Associate, Department of Aeronautics and Astronautics, AIAA Member.
[¶]Post-doctoral Researcher, Department of Aeronautics and Astronautics, AIAA Senior Member.
[‖]Professor, Department of Aeronautics and Astronautics, AIAA Associate Fellow.
[**]Aircraft Conceptual Design Engineer - Embraer, AIAA Senior Member.
[††]Technology Development Engineer - Embraer, AIAA Member.

ability to perform conceptual design using both physics-based predictions (higher-fidelity) as well as relevant correlations (lower-fidelity) based on available historical trends. In short, a design tool is required that allows for an arbitrary aerospace vehicle to be designed with an arbitrary level of fidelity in its supporting data. To allow for these more exotic configurations, SUAVE (Stanford University Aerospace Vehicle Environment) has been designed and developed from the beginning to be a multi-fidelity conceptual design environment for future aircraft incorporating advanced technologies. SUAVE is an open-source environment[1] written in Python and designed as an easy-to-use application programming interface (API) with clearly-understandable syntax accessible to those with minimal Python experience.

SUAVE was initially built under the NASA LEARN Phase I program focused on green aviation systems. Commercial aviation is one of the fastest growing sources of greenhouse gas emissions and yet a critical component of the global economic infrastructure. SUAVE is contributing to the effort to find new vehicle designs that can mitigate this environmental impact.

Moving forward, game-changing leaps in aircraft performance will be found by integrating revolutionary technology, such as new energy and propulsion sources, while adapting the rest of the configuration to take advantage of synergistic interaction at the system level. For instance, an all-electric aircraft with distributed propulsion might simultaneously enable a cleaner wing design and free internal wing volume to house an active laminar flow device rather than liquid fuel. SUAVE will enable this by providing higher-fidelity analysis where necessary to achieve accurate predictions particularly for unconventional configurations with radical technologies, all within a flexible framework for synthesizing analyses at variable fidelities while driving design optimization at the vehicle level.

These goals are achieved through (a) an energy-source-agnostic formulation that makes the integration of both traditional and carbon-free propulsion systems possible, (b) an object-oriented architecture that enables arbitrary aircraft and propulsion system topologies, and (c) an infrastructure that permits the incorporation of multi-fidelity analysis modules: *the right level of fidelity at the right time.*

SUAVE's core is a modular set of components than can be assembled and analyzed with minimal programming effort. `Vehicles()`, a top-level data structure, are assembled from components that include wings, fuselages, propulsors, etc. These components themselves can be specified at a level chosen by the user to match the fidelity of the available analysis methods. For instance, a component can be described by only the minimal information necessary for completing a low-fidelity approximation of its performance, or directly with high-fidelity data and geometry. Vehicles are not limited to aircraft: the class structure in SUAVE has been created with complete generality. For example, during the construction of the class structure, rotorcraft, UAVs, and even launch vehicles were considered and could, potentially, be analyzed in the same way. Aerodynamic information can be generated from simple models within SUAVE or easily imported from external sources like CFD or wind tunnel results. Propulsors (which include anything turning stored energy into a force on a vehicle) are constructed from elements that include compressors, turbines, burners, motors, fans, and so on. Similarly, missions are assembled from segments, which consist of a variety of intrinsic types including cruise, climb, descent, glide, or launch vehicle trajectories. Evaluation of the performance of a given vehicle through a given mission is one of SUAVE's primary functionalities. In order to support arbitrary levels of fidelity, this is done by integrating the relevant equations of motion directly, making the simulation of a mission independent of the level of fidelity of the supporting data. This is performed on a segment-by-segment basis via a pseudospectral collocation method and relies on the user-provided or internally calculated mass properties, aerodynamic information, and propulsion system data.

Overall, SUAVE is constructed to allow aircraft designers of the future, including all ranges of engineers from students learning about aircraft configurations to veteran professionals with years of experience, to integrate useful tools while exploring new concepts. We believe offering the versatility to create radical configurations while integrating advanced analysis techniques from aerodynamics, propulsion, structures, and energy storage, among other disciplines, differentiates SUAVE from other conceptual codes. Through these additional capabilities, we hope to provide an environment where different optimization and uncertainty quantification (UQ) methodologies can be connected to create realizable vehicles that could not be analyzed before.

This paper is organized as follows: Section II describes the detailed code architecture, programming language layout, multi-fidelity integration, and optimization approach. Section III contains a detailed breakdown of the current analysis techniques integrated into SUAVE. Section IV describes case studies of both the basic functionalities on traditional configurations and more advanced methods on exotic designs. Finally, a summary of current capabilities and outline of future development is located in Section V.

American Institute of Aeronautics and Astronautics

II.   Architecture

To accomplish the functional goals of SUAVE, a significant effort was given to developing its code architecture. This section will review the decisions made while laying out the interfaces and structures of the environment. First, high level topics including design principles, choice of programming language, and method-attribute orthogonality will be covered. We will then describe the more functional aspects of the architecture, including its major attributes and methods, how it is organized for multi-fidelity, and how SUAVE can interface with optimizers in the future.

A.   Code Design Principles

This section consists of a discussion on code design, separate from the topic of vehicle design. A code that can examine unconventional configurations at variable fidelities requires a high degree of *flexibility*, *composability* and *extensibility*. We must plan for a wide range of model requirements, and especially models that forgo the traditional analysis structure for the performance of tube-and-wing vehicles.

1.   Language

We have chosen to implement the above principles in Python. As with many scripting projects, Python's combination of object oriented programming, duck typing, concise language, portability in open source, and community standard as glueware has driven our decision. Python's concept of duck typing is especially used by SUAVE to enable rapid prototyping of new analyses, as will be discussed in further sections. The flexibility of Python comes with a notable penalty in speed, which needs to be managed carefully by vectorizing where possible and minimizing overrides to fundamental functions like attribute access which are called very often.

There were other languages considered in SUAVE's early planning stages. C++, Java, and Go are comfortably object-oriented and fast, but their strong typing is not suited for a scripted code like this as it becomes too cumbersome. Go is a notable language for its concurrent computing structures, but at the start of SUAVE's construction two years ago, Go was not sufficiently integrated with scientific computing packages.

Julia is also another notable language for its strong inclusion of scientific computing especially automatic differentiation, its flexibility, and speed that can match that of C++. At present it is still a young language, but with more development it will become an important option to consider for design optimization.

2.   Flexibility

While many definitions of code flexibility exist,[2,3] we settle on the following definition:

Definition 1. *A system is flexible if modifications that are known to be necessary can be locally confined.*

Programmatically, this means that a user does not need to dig in the base classes, or implement a class extension (this would be extensibility) to make a change.

SUAVE implements flexibility by building a data structure based on the Python `dict()` dictionary. The base data class in SUAVE is named `Data()`. SUAVE `Data()` dictionaries are mutable like the base Python dictionary, meaning objects can be added, deleted, or modified. For convenience, the traditional Python dictionary syntax with square brackets has been overloaded with the dot syntax traditionally reserved for attribute access. This is a strong expression of the code design principle that almost every object in SUAVE is data, which in turn is most easily managed by a mutable mapping like a Python dictionary.

To document what is nominally expected as inputs by SUAVE modules, these data structures include the concept of defaults. These are intelligent values that are defined inside the data class and are set at object instantiation.

For an illustrating example, consider a notional wing data structure:

```python
class Wing(Data):
    def __defaults__(self):
        self.sweep = 0.0
        self.span = 1.0
```

American Institute of Aeronautics and Astronautics

Here a `Wing()` class is declared extending the base `Data()` object. The `__defaults__()` class-method provides the opportunity to assign defaults to `self`, which in this case would be the instantiated wing object. The wing would then be instantiated with:

```
wing = Wing()
```

which creates an object with attributes that are pre-assigned with the default values:

```
print wing.sweep
>>> 0.0
print wing.span
>>> 1.0
```

These attributes can be modified,

```
wing.sweep = 10.0
```

and new attributes can be added, even if they were not defined as a default,

```
wing.twist = 2.2
```

This flexibility shows its power when paired with SUAVE's composable analysis structure, which lets users patch in new functions prototyped in their test scripts, preventing the need to work inside of the SUAVE package in early stages of implementing new models.

With this flexibility comes the risk of an unmanageable naming structure. There are many analyses requiring various colors of data. To manage this, SUAVE's style guide includes the convention that all field names needed while setting up the vehicle and analyses should be named verbosely and be closely tied to the physical quantity they represent. In the above example, `wing.sweep` was used, instead of `wing.lambda`. In sticking to this convention we can also mitigate the requirements of extensive variable defining documentation by assuming that variables are named in a discoverable way. This means that based on the context in which they are defined it is obvious what the fields do and what information they contain.

3. Composability

The mutability of the SUAVE `Data()` dictionary enables another key code design element of composability. We define composability with the following:

Definition 2. *A system is composable, if functionality of a given object is largely derived by assembling one or more other objects.*

The act of assembling one or more objects into a composed object is also referred to as *marshalling*.[2] By using composition, and exposing the ability to marshal new objects, it is possible to construct arbitrary vehicles of arbitrary fidelity. In turn, this makes analyses self-contained, and their functionality transparent.

An example of how composability is used is shown in a keystone SUAVE data structure called `Conditions()`. This structure, the details of which are provided later in the Attributes Section II.B, contains the conditions, in the broadest sense of the word, of the vehicle as it flies in a mission. It contains for example the freestream pressure, and velocity, as well as vehicle positions and orientations.

The most powerful use of `Conditions()` is found by appending new data as desired inside an analysis. The `Conditions()` data is passed to each analysis in the mission. Analyses are allowed to modify the conditions data, and they may append additional data, which could include for example components of drag from an incremental drag analysis.

In this way the composition pattern permeates even into the way data is manipulated during vehicle analysis. Thus SUAVE easily liberates data for visualization or optimization, which would otherwise be hidden inside of low level analyses.

American Institute of Aeronautics and Astronautics

4. Extensibility

If flexibility allows SUAVE to adapt, extensibility allows SUAVE to grow. We define extensibility with this:[3]

Definition 3. *A system is extensible, if functionality can be added while making small or no changes to the system's internal structure and data flow.*

SUAVE implements this with traditional object-oriented principles like class inheritance, as well as the principle of common interfaces. Higher level analyses like missions expect standardized interfaces of inputs and outputs. Often inputs and outputs are contained in a single dictionary-like data structure to further reduce assumptions on the extent of the interface.

By setting common interfaces we can create whole categories of methods that would be useful. This is important for managing the different types of analyses that are needed for different unconventional aircraft.

An illustrating example of this is provided here, by showing three ways of casting an aerodynamics function needed by the mission solver.

The mission solver runs the vehicle through a set of segments, for example climb, cruise and descent. In evaluating these, it expects an aerodynamics object. The object when called need only receive conditions as an input, and return lift and drag as an output. How this is accomplished is up to the aerodynamics object, enabling an exceptional amount of extensibility.

The first way of implementing such an aerodynamics object for SUAVE can be accomplished by implementing a Python method.

```python
def aerodynamics(conditions):
    AoA = conditions.freestream.angle_of_attack
    CL = pi * AoA
    CD = pi * AoA**2
    return Results(
        lift_coefficient = CL ,
        drag_coefficient = CD )
```

This is easily marshalled into a mission analysis, notionally shown with this syntax:

```python
mission.analyses.aerodynamics = aerodynamics
```

A second way of implementing the aerodynamics object is by writing a callable class:

```python
class Aerodynamics(object):
    def __defaults__(self):
        self.lift_slope = pi
    def __call__(self,conditions):
        AoA = conditions.freestream.angle_of_attack
        CL = self.lift_slope * AoA
        CD = self.lift_slope * AoA**2
        return Results(
            lift_coefficient = CL ,
            drag_coefficient = CD )
```

This class can be instantiated and then marshalled into the mission object in a similar manner:

```python
mission.aerodynamics = Aerodynamics(lift_slope = 0.9*pi)
```

Finally, an aerodynamics class can be extended in the traditional object-oriented manner. In this example we extend a high-lift aerodynamics model by adding a lift increment:

American Institute of Aeronautics and Astronautics

```python
class High_Lift(Aerodynamics):
    def __defaults__(self):
        self.lift_increment = 0.1
    def __call__(self,conditions):
        # call the parent class
        results = Aerodynamics.__call__(self,conditions)
        # apply the increment
        results.lift_coefficient += self.lift_increment
        return results
```

Note that this class relies on the parent's aerodynamics class to calculate a first estimate of lift and drag. This is an example of the important principle of code *extensibility*.

5. *Attribute-Method Orthogonality*

An essential design principle in SUAVE is that of attribute-method orthogonality. This design principle is what enables the multi-fidelity aspect of SUAVE.

The premise is that vehicle and mission data, or more properly in programming vocabulary, "attributes", are responsible only for describing what things are, not how they work. A follow-on definition of this is that Attributes do not express assumptions of fidelity. To emphasize the separation of attributes, they are stored in one branch of the SUAVE package: SUAVE.Attributes. The fields contained in an Attribute could be high-level data, like sweep and span, or detailed data, like span-wise twist angles. In being specific about the name of the field and grouping detailed data into sub-fields we have so far been able to express a vehicle without conflicting field names.

Methods are responsible for managing how the attributes are manipulated or processed to produce results. They are stored in a separate arm of the SUAVE package: SUAVE.Methods. Some methods are actually callable classes, and given the differentiating name SUAVE.Analyses. Python enables this with the special function `__call__()`. In this case the data that is contained in these callable objects are only related to configuring the analyses itself.

The construction of attribute-method orthogonality enables multiple fidelity levels in SUAVE because analyses are swappable, assuming they share a common interface (this is an important result of extensible code design as described earlier). In the simplest implementation, this appears in practice as the ability to run vehicle analysis twice, each time with a different aerodynamics method. Extending this implementation, a true multi-fidelity analysis object can be constructed by composing multiple analysis sub-objects, each of different fidelity, to be managed and merged by the parent object according to a chosen procedure.

B. Attributes

1. *Vehicle*

The `Vehicle()` data structure stores information defining the geometrical layout and physical properties of the aircraft. It features a hierarchical set of data containers for various categories of components and properties. Vehicle data is typically specified by the user in a Python script. The user creates individual aircraft components such as the fuselage, main wing, and tails from existing SUAVE component classes, and appends them to a `Vehicle()` data structure. The `Vehicle()` object has a built-in append function that recognizes the types of the components and organizes them into the appropriate containers in the geometry hierarchy. The physical components are categorized into the following groups: wings, fuselages, propulsors, energy, systems, and payloads. The `Vehicle()` data structure also includes overall aircraft properties such as mass properties, cost, and envelope limits.

A separate overarching `Configs()` data structure contains configuration objects that reference the baseline vehicle and specify perturbations to the geometry or properties. For example, the user can define a climb configuration, a cruise configuration, and a descent configuration, each with different flap settings for climb, cruise, and descent, respectively. Together, the `Vehicle()` and `Configs()` objects provide all the aircraft-level parameters required to perform analyses in different stages of the mission.

The components of a `Vehicle()` object are meant to be initialized without assuming any particular method of analysis. Analysis parameters and settings are specified in `Analysis()` objects rather than in the

`Vehicle()` itself. In general, the analysis object receives the `Vehicle()` as one of its function arguments and analyzes the `Vehicle()` performance without altering the `Vehicle()` object or its contents. There are, however, some exceptions. For example, user-defined sizing and preprocessing functions may alter `Vehicle()` parameters such as weights or engine dimensions.

2. Missions

Missions are built to be extremely flexible. The mission contains an ordered dictionary of segments. Each segment contains a data structure of conditions, to be described shortly. It also contains a data structure of analyses, such as aerodynamic analysis, orientation calculations, atmospheric conditions determination, and propulsion evaluation. These are called by the mission solver, to be described later.

The mission is defined by the user in the top level setup script. This is done by composing a new object, and not by extending a new class. In practice, this involves instantiating an empty mission, then instantiating the desired segments necessary to create the desired mission and appending the segments to the mission's segment list. Segments are implemented as classes and are pulled from the SUAVE Python package.

This allows SUAVE to minimize assumptions made on the composition of the mission. Arbitrary combinations of climb, cruise and descent segments can be composed, without requiring changes to how the mission is solved. Additional segments types can also be extended from existing segments. Composability in this way allows an exceptional amount of code reuse.

3. Conditions

The `Conditions()` data structure is used to store information regarding the state of the aircraft and its aerodynamic environment. `Conditions()` data is generally stored in two-dimensional column arrays. `Conditions()` records the aircraft's position, orientation, velocity, acceleration, and applied forces in the relevant frames of reference. Furthermore, `Conditions()` stores aerodynamic coefficient results and drag breakdowns for the individual components of the aircraft. The purpose of a `Conditions()` object is to exchange up-to-date data within analysis objects and then later present them to the user. Rather than passing long lists of input variables, SUAVE analyses can receive, pass on, and return a `Conditions()` object that is progressively updated with new results and flight conditions.

The use of the `Conditions()` data structure can be demonstrated by the process of segment analysis. When a SUAVE mission segment object is instantiated, it creates its own associated `Conditions()` data structure to hold data relating to the analysis. The contents of the `Conditions()` are initialized with data including atmospheric conditions and aircraft initial position provided by the segment. Most variables in the `Conditions()` are initialized to zero and later updated with analysis results. After initialization, the segment expands the `Conditions()` variables to some number of control points – typically 16 points by default. With these control points in place, a segment solver can modify specific free variables iteratively to push the segment residuals to zero and solve for the state of the aircraft throughout the segment. This solved state – including aircraft position and orientation, forces on the aircraft, aircraft weights, propulsion state, and freestream conditions – is stored in the `Conditions()`. The `Conditions()` object is finally put through post-processing and returned for further analysis or data presentation.

4. Energy Networks

In addition to modeling standard turbofan and turbojet-powered aircraft, SUAVE can handle a variety of different energy and power sources, including batteries, solar panels, and fuel cells. Integrating the collection of energy sources into an "energy network," vehicles with arbitrary energy resources can be used by the mission solver. The level of fidelity varies from incorporating component-level efficiencies that may be altered by the user (not only as static efficiency factors, but arbitrary curves and functions as well) to physics based modeling. The created energy network takes in the mission solver values along with inputs and outputs from different analysis components, and returns the thrust, vehicle mass change rate, and other properties to the mission as necessary. Conservation of mass, energy, and current are maintained by the energy network object.

As an example, energy networks for a standard turbojet, as well as hybrid electric turbojet can be seen in Figures 1 and 2, respectively.

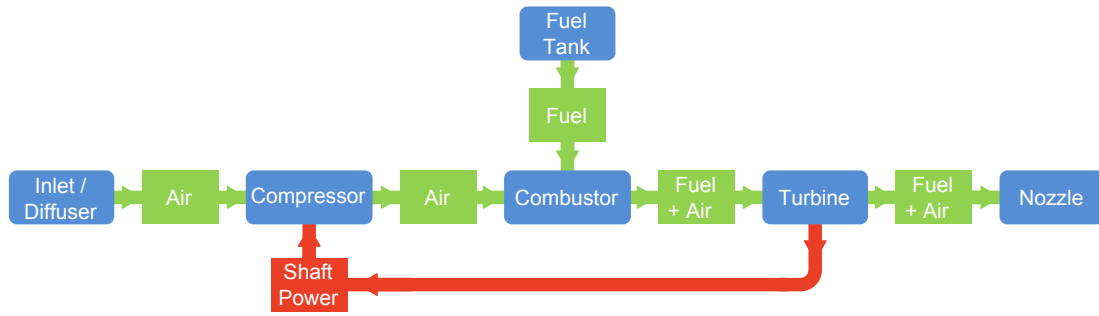American Institute of Aeronautics and Astronautics

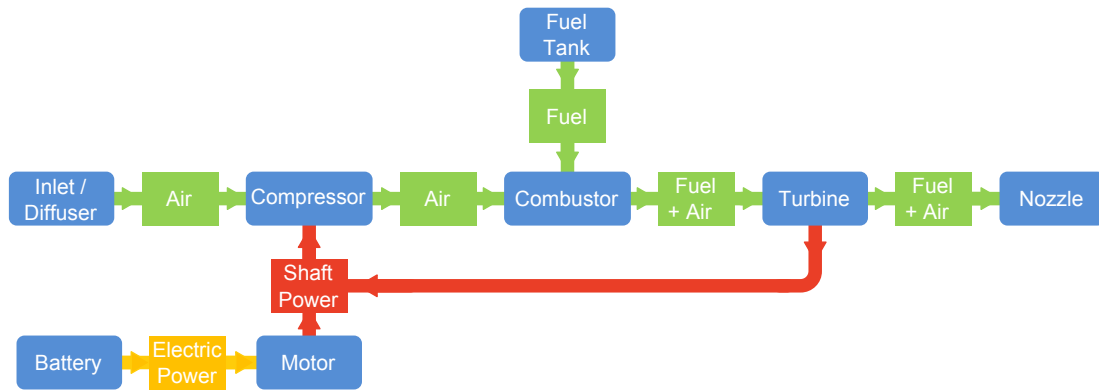Figure 1: Standard Turbojet Power and Mass Flows

Figure 2: Hybrid Turbojet Power and Mass Flows

Both Figures 1 and 2 depict turbojet systems which use fuel as an energy source to power the compressor/turbine system, and thus, produce thrust. Figure 2 differs from Figure 1, in that it uses a battery as an energy source to power a motor, which converts the shaft power to assist in powering the compressor. Figure 2 is not a standardized SUAVE Network object as of yet. It is, however, easily constructed from the described set of propulsor and energy components. Note that in Figures 1 and 2, boxes with rounded corners represent components, and boxes with sharp corners represent various forms of energy transfer between components. Additional "energy networks" added to SUAVE require the user to handle conversion between data types within the network object. Advanced networks can be built as "networks of networks" in the case of exotic propulsion systems.

A network for a battery, connected to a motor, which is in turn connected to a ducted fan can be seen in Figure 3.
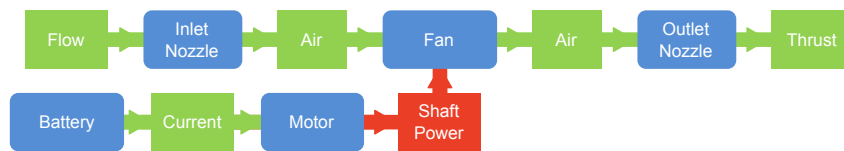

Figure 3: Battery-Motor-Ducted Fan Energy Network

To elaborate, Figure 3 depicts the physical components in blue boxes with rounded corners, while the different flow properties (e.g. current, power, mass) are in the green rectangular corners. The battery acts as a (finite) current reservoir, converted by the motor into shaft power, used by the ducted fan to produce

American Institute of Aeronautics and Astronautics

thrust, the desired mission object input.

Note the differing levels of fidelity in the electric system between Figures 2 and 3. Figure 3 models current/voltage interactions between the motor and battery system, whereas Figure 2 simply assumes a certain amount of energy in the battery and motor losses are modeled via a constant efficiency factor. A version of the battery network from Figure 3 can be just as easily defined that is agnostic to current, with the battery acting as an energy, rather than current reservoir. Similarly, the hybrid network from Figure 2 can be defined to evaluate current/voltage relations between the battery and motor. Interaction between any set of components is handled by the energy network, with the user prescribed connections determining the properties and variables (e.g. current, mass flow, energy) that need to be evaluated.

C. Methods

Methods perform actions on SUAVE attributes. By maintaining this separation, extensibility is permitted in the number of fidelity levels available in SUAVE. This hierarchy of fidelities needs to be organized in a useful way. Information sources are organized by discipline. They are given a descriptive name where possible. Additionally, they require common interfaces, which vary by discipline. However, they often must, at minimum, be able to accept the conditions data and manipulate it appropriately. Multiple fidelities can be composed into higher level analyses. For example, an aerodynamics analysis object is required by a mission segment, and a mission segment analysis object is required by a mission. This composition can be expressed in the top level run script, which enables rapid prototyping of functionality.

Detailed discussion of the SUAVE Analysis methods are found in Section III.

D. Multi-Fidelity

Within the SUAVE methods section, there are multiple fidelity levels for each of the different component analyses. Since the beginning, defining different fidelity levels has been one of the driving factors in the perceived necessity of SUAVE. While incorporating different fidelity levels offers the ability of the user to set a desired level of accuracy, multi-fidelity analysis must balance this accuracy level with the computational resources and time available. If a new configuration is being designed, a basic drag build-up might sufficiently describe the cruise segment, but this same methodology would not likely work as well in more extreme environments, such as takeoff and landing. In SUAVE, the user could combine the drag build-up for cruise with CFD data, in the form of SU2[4] or another solver, to increase fidelity where necessary, but not at the expense of increasing fidelity everywhere.

To effectively combine different fidelity levels, SUAVE uses response surfaces to map data over the entire domain of interest. A low fidelity analysis can then be corrected to the high fidelity using a differential or multiplicative offset. For example, aerodynamics data is used in one case to create a response surface over all angles of attack to be seen in the mission. This not only speeds up the evaluation of the mission, but also allows higher fidelity data to improve the accuracy of the low fidelity model using a small number of high fidelity points. In the simplest case, if a low fidelity model has a fixed offset over the entire domain, one higher fidelity data point will correct the offset and improve the global accuracy.

Multi-fidelity capabilities offer significant reward when considering more exotic designs. While most of SUAVE's initial analyses were first constructed for tube-and-wing configurations, understanding when these capabilities can be extended to radical configurations is important. This will allow engineers the ability to gain knowledge of the design space. Considering a large number of methods are built on the underlying physics of the environment, it seems likely that some of the basic methods can be extended, provided engineering judgment is used. For example, trying to use a wing weight estimation of a tube-and-wing aircraft for a BWB would not be suggested, but the drag build up could be extended and then corrected with additional factors as necessary. Overall, being able to support connections between different analysis capabilities providing the same information is necessary for any new conceptual design environment.

E. Units

Depending on discipline and even type of aircraft there are different sets of customary units for aerospace analysis. SUAVE provides support for unit conversion through a native port of the `pint`[5] dimension management package. Internally, all interfaces to functions and objects, including inputs, outputs and defaults,

are expected to be expressed in Standard Internationale (SI) with base units of meters, kilograms, seconds, and radians etc. Conversion into units needed for handbook methods or plotting is accomplished with the SUAVE.Units package. In an example, a number is converted into SUAVE-compliant SI with the multiplication operator,

```
span = 50. * Units.feet
```

where in this example the value of 50 US feet would be converted into meters and stored into the span variable. Similarly, a number is converted out of the SUAVE-compliant SI units with the division operator,

```
distance_miles = distance / Units.miles
```

where in this example the value of distance in meters would be converted into US miles.

In order to provide this particular flavor of syntax sugar, some of pint's functionality has been modified. Thanks to pint's clever construction of a units database, a large selection of units are available including weights and time, and can transparently handle combinations like torque, areas, and volumes.

F. Optimization

As designs become more and more complicated and move further away from traditional configurations, it becomes unreasonable to expect aircraft designers to be able to identify how all parameters correlate with one another. With this thought, optimization capabilities are of utmost importance when developing SUAVE. To retain as flexible an environment as possible, the analysis capabilities of SUAVE run as a black-box function where inputs are specified and outputs of interest generated in a manner where any optimizer can be incorporated. SUAVE is designed to handle the sizing and optimization of a number of disparate cases, with the mission solver in particular constructed in a manner to return even highly unreasonable, yet continuously varying results. Future publications will detail how optimization capabilities are integrated with SUAVE.

III. Discipline Analyses

Within SUAVE, each different discipline (Aerodynamics, Structures, Stability & Control, Propulsion, etc.) has its own methods that are used to assess the combined vehicle and mission of interest. These disciplines are broken down by fidelity level, discipline, and even component, such as drag and lift for aerodynamics.

A. Aerodynamics

Computing the aerodynamics of the aircraft is essential for the prediction of aircraft performance. To solve the mission composed of the specified segments, the lift and drag of the aircraft are calculated at a set number of control points. Moreover, for the computation of the stability characteristics of the aircraft, lift, drag and moment coefficients are required. Generally for conceptual design, simple inexpensive correlation-based methods are used to compute aircraft properties. However, in SUAVE, the aerodynamics module is set up to permit multiple fidelity levels of analysis. The choice of the aerodynamic class is made using the Aerodynamics attributes that is a callable class that helps assign the appropriate aerodynamics module to a configuration. Thus, based on the level of fidelity of the aerodynamics desired, the appropriate callable class is selected. Currently, the lowest level of fidelity models available for use are the Fidelity Zero models. These are described below. Work is ongoing on connecting higher fidelity models directly to SUAVE, but response surfaces are available to incorporate higher fidelity lift and drag data for analysis purposes.

1. Fidelity Zero

Currently, the lowest level of fidelity in the aerodynamics module is the fidelity zero methods which uses a vortex lattice[6] for the prediction of lift and the correlations for the prediction of drag for the subsonic case.

LIFT

SUAVE has capabilities for both subsonic and supersonic lift.

SUBSONIC LIFT

A vortex lattice method is used for computation of the inviscid lift for subsonic flows.[6] When the aerodynamics module is assigned to a configuration, the vortex lattice is run for a few angles of attack to obtain a set of C_L. Then, a polynomial is fit through the data to obtain a surrogate for the inviscid wing C_L vs angle of attack. During the mission analysis, this surrogate is used to compute the inviscid wing C_L for all flight conditions. Factor-based corrections to account for the fuselage, compressibility, and viscous effects are applied to the inviscid wing C_L to obtain the aircraft C_L.[7]

Computation of C_L increments associated with the high lift devices is performed using correlation based methods.[7,8] The effect of the wing slat on wing $C_{L_{max}}$ is computed using Equation 1 where $\Delta C_{L_{slat}}$ is the increment in C_L due to the slat, Λ is the wing sweep and sa is the slat angle.

$$\Delta C_{L_{slat}} = \frac{sa}{23} \cos(\Lambda)^{1.4} \cos(sa)^2 \tag{1}$$

The effect of flaps on $C_{L_{max}}$ is computed using the Equation 2 where K_c is a flap chord correction factor, K_m is the flap motion correction factor, K_{sw} is a sweep correction factor and $d_{max_{ref}}$ is an increment (in two dimensions) in $C_{L_{max}}$ for the 25% chord flaps at the 50 degrees landing flap angle. The computation of these parameters is explained in detailed in the text by Shevell[8] and Kroo.[7]

$$\Delta C_{L_{flaps}} = K_c K_m K_{sw} d_{max_{ref}} \tag{2}$$

The $C_{L_{max}}$ is then computed using Equation 3 where $C_{L_{max,wing}}$ is the lift coefficient associated with the wing due to the effect of Reynolds Number, Mach Number and the stall speed. Computation of $C_{L_{max,wing}}$ is described in the text by Shevell[8] and Kroo.[7]

$$C_{L_{max}} = C_{L_{max,wing}} + \Delta C_{L_{slat}} + \Delta C_{L_{flaps}} \tag{3}$$

SUPERSONIC LIFT

Linear supersonic theory[8] is used with strip theory for computation of lift in supersonic flow. As with the subsonic case, this is run for a few angles of attack and a model is created to capture and apply the results. Equation 4 calculates the airfoil lift coefficient using

$$c_l = \frac{\alpha}{\sqrt{M^2 - 1}}, \tag{4}$$

where c_l indicates chordwise lift coefficient. This is integrated spanwise to find the total C_L.

Since many supersonic aircraft rely on vortex lift in subsonic conditions, vortex lift is computed for relevant aircraft. The Polhamus Suction Analogy[7] is used to determine C_L in this instance as shown in Equation 5 with angle of attack denoted by α. The sweep here is the sweep of the leading edge, shown as Λ.

$$C_{L_{vortex}} = \frac{\pi AR}{2} \sin\alpha \cos\alpha \left(\cos\alpha + \frac{\sin\alpha \cos\alpha}{\cos\Lambda} - \frac{\sin\alpha}{2\cos\Lambda} \right) \tag{5}$$

This is combined with the standard subsonic calculations to create a surrogate model for low speed flight.

DRAG

SUAVE has capabilities for subsonic and supersonic drag.

SUBSONIC DRAG

Subsonic drag associated with the aircraft is predicted by first computing the different subcomponents of drag and then adding them up them to get overall drag value. The aircraft drag is divided into the parasite drag, lift-induced drag, compressibility drag, and miscellaneous drag.

PARASITE DRAG

American Institute of Aeronautics and Astronautics

Parasite drag of the aircraft is the drag associated with the skin friction and pressure drag of the aircraft. The parasite drag for the aircraft is computed by adding the contributions of the fuselages, wings, and nacelles. Both the wing and the fuselage use Equation 6 to compute parasite drag. The skin friction coefficients are computed assuming the approximate compressible flat plate equations calculate the skin friction coefficient associated with the fuselage, wing, and nacelle.

$$C_{D_p} = k \; C_f \; S_{ref} \tag{6}$$

Equation 6 denotes C_{D_p} as the parasite drag, k as the form factor, C_f as the skin friction coefficient, and S_{ref} as the wing reference area.

The form factors for the wing and the fuselage are computed differently. For the fuselage, the relation based on Shevell's book[8] is used as shown in Equation 7,

$$k = (1 + C \; \delta u_{max})^2, \tag{7}$$

where C is a user defined factor normally set to 2.3 and δu_{max} is the maximum velocity increase on an ellipsoid of revolution. Parasite drag on propulsors is similar, but if the nacelle diameter, inlet area, and exit area are available, k will may be calculated more precisely with a better estimate of the largest change in velocity.

To compute the form factor for the wing, Equation 8 also from Shevell[8] is used where Λ is the wing sweep, M is the cruise mach number, C is a wing form factor, and t/c is the thickness-to-chord ratio of the wing.

$$k = 1 + \frac{2C(t/c \cos(\Lambda)^2)}{\sqrt{(1 - M^2(\cos(\Lambda))^2)}} + \frac{C^2 \cos(\Lambda)^2 (t/c)^2 (1 + 5\cos(\Lambda)^2)}{2(1 - (M\cos(\Lambda))^2)} \tag{8}$$

LIFT INDUCED DRAG

Another significant component of the drag breakdown is the lift induced drag. This is computed by using a correlation based on the aircraft lift and the inviscid drag computed by a vortex lattice method (or any other inviscid method).

The lift induced drag contains a viscous and an inviscid component as shown in Equation 9.

$$C_{D_i} = C_{D_{inviscid}} + C_{D_{viscous}} \tag{9}$$

For the lowest level of fidelity, Equation 11 is used. The variable e is not the span efficiency factor. Instead it is computed using equation 10 where $e_{inviscid}$ is the span-efficiency factor, AR is the wing aspect ratio, K is a scaling factor determined from flight test data and C_{D_p} is the parasite drag coefficient. This formula allows us to take into account the viscous contribution to lift induced drag as well. C_L is the aircraft lift and C_{D_i} is the induced drag.

$$e = \frac{1}{\frac{1}{e_{inviscid}} + \pi \cdot AR \cdot K \cdot C_{D_p}} \tag{10}$$

$$C_{D_i} = \frac{C_L^2}{\pi ARe} \tag{11}$$

COMPRESSIBILITY DRAG

To model the effects of compressibility, corrections to the drag are computed by finding the crest critical Mach number (M_{cc}) and then using the correlations in Equation 14 to arrive at an estimate for the compressibility drag associated with the aircraft configuration. The effective thickness to chord ratio and effective C_L are computed as shown in Equations 12 and 13. From these results, the crest critical and divergence Mach numbers are computed using equations obtained by regressing plots shown in Shevell's book.[8] Finally, the compressibility drag is calculated.

$$t/c_{corrected} = \frac{t/c}{\cos(\Lambda)} \tag{12}$$

$$C_{L_{corrected}} = \frac{C_L}{\cos(\Lambda)^2} \tag{13}$$

$$C_{D_c} = 0.0019 \left(\frac{M}{M_{cc}}\right)^{14.641} \cos(\Lambda)^3 \tag{14}$$

MISCELLANEOUS DRAG

In addition to the components described above, the contribution of drag due to fuselage upsweep, control surface gaps, air conditioning systems etc. is not modeled explicitly. These are instead lumped into a miscellaneous drag component. Accounting for these small but important factors is essential for accurate drag estimation.[7,8]

SUPERSONIC DRAG

In the supersonic case, drag is computed as above with some important changes and additions where appropriate.

PARASITE DRAG

Above Mach 0.95, the form factor of the wing is set to 1, while compressibility effects are removed from the form factor for the fuselage. This correction, which is expected to slightly underestimate parasite drag in this regime, accounts for the inability of standard correlation methods to capture supersonic flow effects. Mach effects are still considered in the skin friction coefficient.

LIFT INDUCED DRAG

In supersonic flow span efficiency factor is no longer a concern. Therefore a modified equation is used neglecting this correction.[7]

$$C_{D_i} = \frac{C_L^2}{\pi AR} \tag{15}$$

COMPRESSIBILITY DRAG

Unlike in low Mach subsonic flow, wave drag must be accounted for during supersonic flight. This includes both volume and lift dependent wave drag. First the wings are considered. Equations 16 and 17 are volume and lift wave drag coefficients respectively for wings. Equation 19 calculates the lengthwise aspect ratio with L denoting the total length of the wing.

$$C_{D_{w_v}} = \frac{4(t/c)^2(\beta^2 + 2x^2)}{(\beta^2 + x^2)^{1.5}} \tag{16}$$

$$C_{D_{w_l}} = \frac{C_L^2 x(\sqrt{1 + \beta^2/x^2} - 1)}{4} \tag{17}$$

$$x = \frac{\pi AR_L}{4} \tag{18}$$

$$AR_L = \frac{L^2}{S_{ref}} \tag{19}$$

$$\beta = \sqrt{M^2 - 1} \tag{20}$$

Wave drag for bodies of revolution is then calculated with L representing the length of the fuselage and R_{max} representing the maximum radius.

$$C_{D_{wv}} = \frac{9\pi^3 R_{max}^4}{2L^2 S_{ref}} \tag{21}$$

These results assume ideally shaped long slender bodies, so a correction term of 15% is added.[7] Outside of wave drag, special consideration is also given to the transonic region since all supersonic aircraft must pass through this regime. Additional corrections are used here depending the type of aircraft. Aircraft with high subsonic cruise Mach numbers are assumed not to follow the standard correlations given above.

Since no low fidelity model is available very close to sonic conditions, a linear fit is created that transitions from empirical compressibility drag computations at Mach 0.99 to analytical wave drag at Mach 1.05.

Additionally, delta wing configurations will not follow the empirical trends for the crest critical Mach number (M_{cc}) and drag divergence Mach number. Because of this, these values are simply set to 0.93 and 0.95 respectively. These numbers are based on subsonic cruise values for Concorde.[9]

MISCELLANEOUS DRAG

This module is unchanged from the subsonic case.

B.   Weights

Within the SUAVE methodology, the maximum takeoff weight, W_{max} and maximum zero fuel weight W_{zf} must entered as estimates. After all the other components' masses are estimated, the remaining mass is assigned to fuel. SUAVE is designed to be used for many different types of configurations, however, at present only tube-and-wing, human powered, and blended wing body (BWB) mass estimates are available.

1.   Tube-and-Wing

For the tube-and-wing configuration, the methodology stems from the work of Shevell[8] at Douglas Aircraft and Kroo.[7] As such, all of these equation use English engineering units, in pounds and feet, unless otherwise specified. All of these correlations are based on metal aircraft and at present, there are no corrections for advanced materials. The weight of the wing corresponds to the fully-stress bending weight of the wing box. This is a correlation based method from considering 15 transport aircraft and results in

$$W_w = 4.22S_w + 1.642 \cdot 10^{-6} \frac{N_{ult}b^3 \sqrt{W_{max}W_{zf}}(1+2\lambda)}{(t/c)_{avg}cos^2\Lambda_{c/4}S_w(1+\lambda)} \tag{22}$$

where N_{ult} is ultimate design load factor for the aircraft, usually equal to 2.5. Beyond the wing, other lifting surface weights are of interest to be calculated. The horizontal tail wing weight index, which accounts for the weight of the elevator assuming the elevator is 25% of the horizontal tail area, results in a weight estimation of

$$W_h = 5.25S_h + 0.8 \cdot 10^{-6} \frac{N_{ult}b_h^3 W_{max}c_w \sqrt{S_h}}{(t/c)_{avg}cos^2\Lambda_h l_h S_h^{1.5}} \tag{23}$$

denoting c_w as the mean aerodynamic chord of the wing. While the weight of the horizontal tail accounts for the weight of the elevator, the weight of the vertical tail and rudder are calculated separately even with S_v accounting for the area with rudder included. The weight of just the vertical tail is calculated from

$$W_v = 2.62S_v + 1.5 \cdot 10^{-5} \frac{N_{ult}b_v^3 \left(8.0 + 0.44\frac{W_{max}}{S_w}\right)}{(t/c)_{avg}cos^2\Lambda_v} \tag{24}$$

with the rudder assumed to be 25% of the vertical tail area and weighing 60% more per unit area for a conventional tail or 25% greater than the conventional tail if instead a T-tail is used. To account for the weight of the actuators to operate the control surfaces of the aircraft, the following equation is used.

$$W_{sc} = I_{sc} \cdot (S_h + S_v) \tag{25}$$

I_{sc} is a scaling factor for the surface controls where 3.5 is used for fully-power controls, 2.5 for partially-powered systems, and 1.7 for fully aerodynamic controls. Beyond estimating the weights of the aerodynamic

American Institute of Aeronautics and Astronautics

surfaces, the fuselage weight is the other important estimate necessary. In SUAVE, the fuselage weight is based on the gross fuselage wetted area and a pressure-bending load parameter. The pressure and bending indices are calculated using

$$I_p = 1.5 \cdot 10^{-3} \Delta P_f \cdot w_f \qquad I_b = 1.91 \cdot 10^{-4} N_{lim}(W_{zf} - W_w - W_{w,p}) \cdot \frac{l_{f,e}}{h_f^2} \qquad (26)$$

where ΔP_f is the maximum pressure differential of the fuselage, $l_{f,e}$ is the effective fuselage length, and $W_{w,p}$ is the weight of the wing-mounted engines, nacelles, and pylons. The effective fuselage length is the fuselage length minus the wing chord root divided by two to account for the distributed support provided by the wing. If the fuselage is pressure-dominated, $I_p > I_b$, $I_f = I_p$. Otherwise, $I_f = \frac{I_p^2 + I_b^2}{2I_b}$. The fuselage index is then used in

$$W_f = (1.051 + 0.102 I_f) S_{f,w} \qquad (27)$$

where $S_{f,w}$ is the fuselage wetted area.

The weight of furnishings is accounted by using

$$W_{furn} = (43.7 - 0.037 N_{seat}) N_{seat} + 46 N_{seat} \qquad (28)$$

for aircraft with 300 or fewer seats, or

$$W_{furn} = (43.7 - 0.037 \cdot 300) N_{seat} + 46 N_{seat} \qquad (29)$$

for aircraft over 300 seats. In the case of over water and long range aircraft, an additional 23 lbs per seat is added.

The landing gear weight of the aircraft is simply 4% of W_{max} while the dry weight of the engine is regressed from collection of engines as

$$W_{p,dry} = 0.4054 T_{sls}^{0.9255} \qquad (30)$$

where T_{sls} is the sea-level static thrust of the engine, in lbs thrust. The entire propulsion system weight is W_p is 160% of the dry engine weight. Other components (such as air conditioning, operating items, flight crew, APU, electrical and additional systems) are also calculated following Kroo.[7] Finally, for all of SUAVE, we follow the FAA guidelines estimating each passenger weighing 225 lbs, flight crew weighing 240 lbs, and flight attendants weighing 210 lbs.

2. Blended Wing Body

SUAVE uses methods developed by Bradley of George Washington University to obtain low-fidelity weight estimates for blended wing body (BWB) centerbodies.[10] The method assumes a composite sandwich structure in which the cabin segment of the centerbody is designed to carry both pressure and bending loads. The aft centerbody is assumed to be unpressurized and is modeled as a horizontal tail, with corrections for the structure supporting aft-mounted engines. The aft centerbody is assumed to have control surfaces. The equations for the cabin and aft-centerbody weights were obtained by fitting to FEA results and verified against Boeing BWB design study results.

$$W_{cabin} = 0.316422 K_s (W_{max})^{0.166552} (S_{cabin})^{1.061158} \qquad (31)$$

$$W_{aft} = (1 + 0.05 N_{eng}) 0.53 S_{aft} (W_{max})^{0.2} (\lambda_{aft} + 0.5) \qquad (32)$$

The scaling factor, K_s, is 5.698865. Bradley confirmed that these equations yield results that agree with Boeing BWB design studies. Care must be taken to account for the reduction in wing loading due to the lift of the centerbody. This affects the BWB outer wing weight calculation, and is modeled in low fidelity by calculating the wing weight for a gross weight adjusted according to the ratio of the centerbody and outer wing reference areas. This weight estimation method is demonstrated in the blended wing body case study in Section IV.B.1.

3. Solar/Human Powered

One of the new configurations being considered in SUAVE is a Solar UAV. This vehicle is designed for long endurance flight. Lightly loaded, high aspect ratio, carbon fiber, and composite construction is considered critical to a successful vehicle. When looking for vehicles constructed in a similar manner, the MIT Daedalus project matches these characteristics.[11] There are several useful correlations available from this project. They are incorporated within SUAVE for the wing, tail, and fuselage. All equations in this section are evaluated using metric units. The weight of the wing is broken down by cantilevered wing spar, ribs, end ribs, leading edge, trailing edge, and coverings. These are calculated using

$$W_{w,s} = (b_w \cdot 1.7 \cdot 10^{-1} + 1.1 \cdot 10^{-2} \cdot b_w^2)\left(1 + \frac{\frac{N_{ult}W_{max}}{100} - 2}{4}\right) \tag{33}$$

$$W_{w,r} = N_{ribs}(5.5 \cdot 10^{-2}c_{w,avg}^2(t/c)_w + 1.91 \cdot 10^{-3}c_{w,avg}) \tag{34}$$

$$W_{w,er} = N_{endribs}(6.62 \cdot 10^{-1}c_{w,avg}^2(t/c)_w + 6.57 \cdot 10^{-3}c_{w,avg}) \tag{35}$$

$$W_{w,le} = 0.456\frac{S_w^2\Delta_w^{4/3}}{b_w} \tag{36}$$

$$W_{w,te} = 2.77 \cdot 10^{-2}b_w \tag{37}$$

$$W_{w,c} = 3.08 \cdot 10^{-2}S_w \tag{38}$$

where Δ_w is the average rib spacing to average chord ratio, $c_{w,avg}$ is the average wing chord, N_{ribs} is the number of wing ribs, and $N_{endribs} = 2N_{Ind.wingpanels} - 2$ is the number of wing end ribs. The wing weight is calculated by summing all these weights together. To calculate the tail weight of the aircraft, the component weight for tail spar, ribs, leading edge, and covering need to be estimated. The methods employed to approximate these different components are

$$W_{t,s} = (4.15 \cdot 10^{-2}b_h + 3.91 \cdot 10^{-3}b_h^3)\left(1 + \frac{\frac{q_m \, S_h}{78.5} - 1}{12}\right) \tag{39}$$

$$W_{t,r} = N_{t,r}(1.16 \cdot 10^{-1}c_{t,avg}^2(t/c)_t + 4.01 \cdot 10^{-3}c_{t,avg}) \tag{40}$$

$$W_{t,le} = 0.174\frac{S_t^2\Delta_t^{4/3}}{b_h} \tag{41}$$

$$W_{t,c} = 1.93 \cdot 10^{-2}S_t \tag{42}$$

where q_m is the dynamic pressure at maneuvering speed, Δ_t is the average rib spacing to average chord ratio for the tail, and $N_{t,r} = \frac{b_t}{\Delta_t S_t}$ is the number of tail surface ribs. Finally, the fuselage can be estimated from

$$W_f = (1.14 \cdot 10^{-1}l_{tb} + 1.96 \cdot 10^{-2}l_{tb}^2)\left(1 + \frac{q_m \, S_t 78.5 - 1}{2}\right) \tag{43}$$

where l_{tb} is the length of the tail boom. All of these are estimates shown as only a basic estimate that we believe will be similar to a solar aircraft as a result of similar construction techniques and mission requirements.

4. Other Components

Motors may either be sized using an input mass, or using empirical trends of power vs. weight. This includes regressions for both air-cooled as well as high temperature super conductor motors of overall weight vs. shaft power. A plot of both of these trends, along with some of the data can be seen in Figure 6.[12]
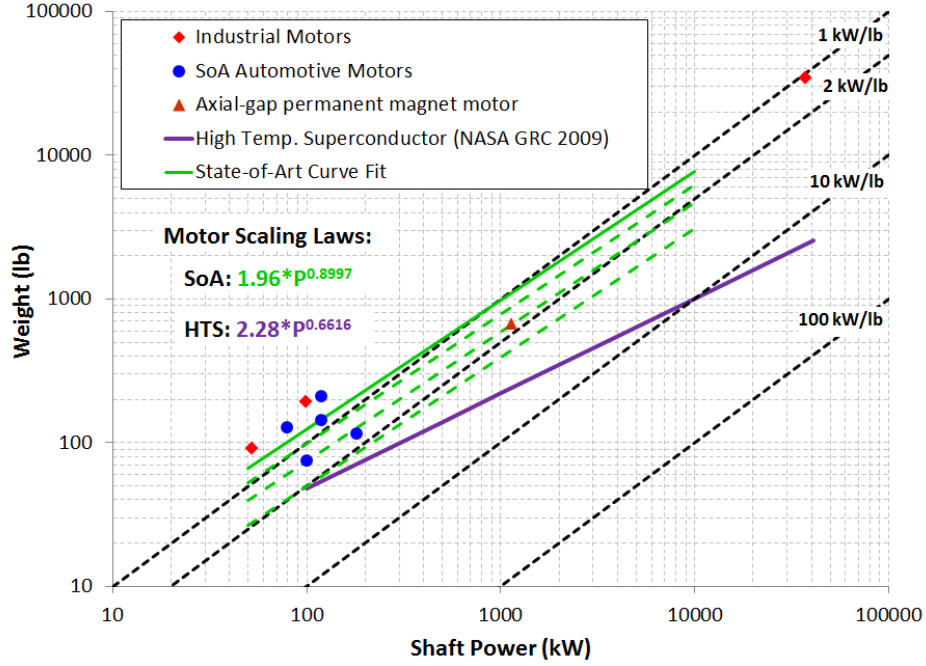
Figure 4: Electric Motor Weight Scaling Correlation

C. Stability and Control

1. Static Stability

SUAVE uses correlation methods described by Roskam[13] to analyze the static longitudinal stability (C_{m_α}) and static directional stability (C_{n_β}). These methods are meant for the analysis of subsonic and transonic tube-and-wing aircraft. The longitudinal stability analysis accounts for the effect of all horizontal lifting surfaces and the fuselage. The contribution of the ith lifting surface to longitudinal static stability is calculated by the equation below.[8]

$$C_{m_\alpha,i} = -\frac{l_i S_i}{\bar{c} S_w} C_{L_\alpha,i} \eta_{hi} \left(1 - \frac{\partial \epsilon}{\partial \alpha}\right) \tag{44}$$

where $\frac{d\epsilon}{d\alpha}$ can be calculated from

$$\frac{d\epsilon}{d\alpha} = \frac{2 C_{L_\alpha}}{\pi AR} \tag{45}$$

The lift curve slope of the lifting surface is computed by the DATCOM formula, shown below.[13] In the DATCOM formula, η is the ratio of the section lift curve slope to the theoretical slope of 2π.

$$C_{L_\alpha} = \frac{2\pi AR}{2 + \sqrt{(AR/\eta)^2(1 + \tan^2 \Lambda - M^2) + 4}} \tag{46}$$

The fuselage contribution to longitudinal stability is calculated as shown below, where K_f is a factor accounting for the effect of wing position along the longitudinal axis.[14]

$$\frac{\partial C_{m_{fuse}}}{\partial C_L} = \frac{K_f w_f^2 L_f}{S_w \bar{c} C_{L_\alpha w}} \tag{47}$$

The directional stability analysis accounts for the effect of the vertical tail and the fuselage, with corrections for Reynolds number. Accurate evaluation of the vertical tail contribution requires the specification of an effective aspect ratio of the vertical tail. This effective aspect ratio accounts for the end plate effect of

American Institute of Aeronautics and Astronautics

the fuselage and horizontal tail. The fuselage contribution $(C_{n_\beta f})$ and vertical tail contribution $(C_{n_\beta v})$ to directional stability are calculated by the equations below.[13]

$$C_{n_{\beta_f}} = -57.3 K_N K_{R_l} \frac{S_{f_s} l_f}{Sb} \tag{48}$$

$$C_{n_{\beta_v}} = -C_{y_{\beta_v}} \left(\frac{l_v \cos(\alpha) + z_v \sin(\alpha)}{b} \right) \tag{49}$$

In these equations, K_N and K_{R_l} are factors accounting for the effects of wing-fuselage interference and Reynolds number on the fuselage contribution. S_{f_s} is the projected side area of the fuselage and l_f is the total fuselage length.

In the vertical tail equation, l_v and z_v specify the position of the vertical tail aerodynamic center relative to the aircraft's center of gravity. The sideslip derivative of sideforce, $C_{y_{\beta_v}}$ is calculated with corrections for fuselage-tail interactions by k_v, wing-tail interactions by $\left(1 + \frac{d\sigma}{d\beta}\right)\eta_v$, and vertical tail properties by $C_{L_{\alpha_v}}$ and S_v.[13]

$$C_{y_{\beta_v}} = -k_v C_{L_{\alpha_v}} \left(1 + \frac{d\sigma}{d\beta}\right) \eta_v \left(\frac{S_v}{S} \right) \tag{50}$$

2. Dynamic Stability

In the Dynamic Stability section, there are two different levels of fidelity available to the user: Approximations and Full Linearized Equations. Both employ methods taken from Blakelock[15] with the objective to quantify the short period and phugoid damping (ζ) and natural frequency (ω_n) in the longitudinal direction and the roll rate (τ_{roll}), spiral rate (τ_{spiral}) and dutch roll damping and natural frequency. In order for SUAVE to calculate the dynamic stability characteristics, the vehicle's moment of inertia must be specified. Functions are also provided to create aerodynamic stability derivatives not known immediately to the designer for desired stability characteristics. All methods that we consider at present in SUAVE require the aircraft to be symmetric along the X-Z axis plane of symmetry so the longitudinal and lateral-directional equation of motion will decouple. Other assumptions for any of these methods to be valid are listed in Blakelock[15] such as constant mass of the aircraft, aircraft being a rigid body, quasi-steady flow, and perturbations from equilibrium being small. Throughout this section, p, q, and r refer to the perturbation quantities related to the total angular velocity of the body with respect to inertial space.

FULL LINEARIZED EQUATIONS OF MOTION

Constructing the full linearized equations, we must assume there is no external inputs, zero initial conditions and neglect $C_{x_{\dot{\alpha}}}$, C_{x_q}, and C_{m_u}. Taking the Laplace transform of the general equations of motion to shift from the time domain (t) to the frequency domain (s), the resulting system of equations shown in matrix form is:

$$\begin{pmatrix} \frac{mV}{Sq}s - C_{x_u} & -C_{x_\alpha} & -C_w \cos\Theta \\ -C_{z_u} & \left(\frac{mV}{Sq} - \frac{c}{2V}C_{z_{\dot{\alpha}}}\right)s - C_{z_\alpha} & \left(-\frac{mV}{Sq} - \frac{c}{2V}C_{z_q}\right)s - C_w \sin\Theta \\ 0 & -\frac{c}{2V}C_{m_{\dot{\alpha}}}s - C_{m_\alpha} & \frac{I_{yy}}{Sqc}s^2 - \frac{c}{2V}C_{m_q}s \end{pmatrix} \begin{pmatrix} u'(s) \\ \alpha'(s) \\ \theta(s) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{51}$$

where the matrix contains the physics of the problem. In Equation 51, Θ is the trimmed pitch angle of the aircraft and $\theta(s)$ is the perturbation of the pitch angle from the trimmed value. $u'(s)$ and $\alpha'(s)$ are the perturbations from the trimmed forward velocity (V) and angle of attack (α), respectively. The short period and phugoid mode characteristics can be calculated by taking the determinant of that matrix. A quartic equation is created with two sets of complex roots, the phugoid and short period modes. Solving those equations to match $s^2 + 2\zeta\omega_n s + \omega_n^2$ will result in the damping and naturals frequency for both longitudinal modes.

For the lateral equations of motion, the same methodology is followed. The C_{y_p} and C_{y_r} terms are neglected and after the Laplace transform is taken of the general equations, the resulting matrix of equations is:

$$
\begin{pmatrix}
\frac{I_{xx}}{Sqb}s^2 - \frac{b}{2V}C_{l_p}s & -\frac{J_{xz}}{Sqb}s^2 - \frac{b}{2V}C_{l_r}s & -C_{l_\beta} \\
-\frac{J_{xz}}{Sqb}s^2 - \frac{b}{2V}C_{n_p}s & \frac{I_{zz}}{Sqb}s^2 - \frac{b}{2V}C_{n_r}s & -C_{n_\beta} \\
-C_{y_\phi} & \frac{mU}{Sq}s - C_{y_\psi} & \frac{mU}{Sq}s - C_{y_\beta}
\end{pmatrix}
\begin{pmatrix}
\phi(s) \\
\psi(s) \\
\beta(s)
\end{pmatrix}
=
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}
\tag{52}
$$

Taking the determinant of this matrix results in a fifth order polynomial, but one root is zero. This root is the heading stability meaning there is no restoring moment to move an aircraft back to its original heading. The reduced fourth order polynomial results in two real poles, the roll subsidence τ_{roll} and the spiral divergence τ_{spiral}. A third set of complex poles, the dutch roll mode is also created. In SUAVE, beyond requiring the moments of inertia for the aircraft to be known, the full linearized equations of motion require C_{y_β}, C_{l_p}, and C_{l_β} be known also. At present, capabilities are not included to estimate these parameters. Other stability derivatives are calculated using the Estimated Stability Derivatives section equations. If this data is not available, but dynamic stability derivatives are desired, then the approximation methods are recommended.

APPROXIMATION METHODS

If there is a desire for dynamic stability calculations, but all of the aerodynamic derivatives necessary to construct the full linearized equations of motion are not known, approximation methods will be used. These are the default dynamic simulation capabilities. We have implemented checks in the code to determine if certain derivatives are known from the aerodynamic analysis. If the necessary aerodynamic derivatives are not known, then estimation methods are available.

Once all the necessary stability calculations are created, through aerodynamic analysis, user input or another method, then stability characteristics can be determined. To calculate the approximate short period characteristics, a constant forward speed and neglecting forces in the x-direction, along with $C_{z_{\dot{\alpha}}}$ and C_{z_q} results in the following two equations to calculate $\omega_{n,sp}$ and ζ_{sp}, respectively.

$$
\omega_{n,sp} = \frac{V\rho Sc}{2}\left(\frac{\frac{C_{m_q}C_{z_\alpha}}{2} - \frac{2m}{\rho Sc}C_{m_\alpha}}{I_{yy}m}\right)^{1/2}
\tag{53}
$$

$$
\zeta_{sp} = -\frac{1}{4}\left(C_{m_q} + C_{m_{\dot{\alpha}}} + \frac{2I_{yy}}{mc^2}C_{z_\alpha}\right)\left(\frac{mc^2}{I_{yy}\left(\frac{C_{m_q}C_{z_\alpha}}{2} - \frac{2mC_{m_\alpha}}{\rho Sc}\right)}\right)^{1/2}
\tag{54}
$$

In the phugoid approximation method, the angle of attack is almost constant and θ is varying slowly. These assumptions result in the inertia forces being neglected and the natural frequency and damping equations are

$$
\omega_{n,p} = \frac{g}{V}\sqrt{2} \qquad \zeta_p = \frac{D}{W\sqrt{2}}
\tag{55}
$$

For the lateral modes, it is assumed that the major effect of rudder deflection is the generation of the Dutch roll mode. This results in the Dutch roll mode consisting of sideslip and yaw so ϕ and any of its derivatives are assumed zero. If $\Theta = 0$ and neglecting C_{y_r}, the $\omega_{n,dr}$ and ζ_{dr} are calculated using the approximations below.

$$
\omega_{n,dr} = V\left(\frac{C_{n_\beta}S\rho b}{2I_{zz}}\right)^{1/2} \qquad \zeta_{dr} = \frac{-C_{n_r}}{8}\left(\frac{2S\rho b^3}{I_{zz}C_{n_\beta}}\right)^{1/2}
\tag{56}
$$

To determine the roll time constant, the major equation of interest is the roll equation. By neglecting the sideslip and yaw angle, these quantities can be set to zero and the resulting equation for roll time constant is

$$
\tau_{roll} = -\frac{4I_{xx}}{S\rho Vb^2 C_{l_p}}
\tag{57}
$$

While longitudinal, dutch roll, and spiral modes follow a large number of assumptions to reduce the equations of motion to the approximation methods. Blakelock states, "it proves difficult to evaluate the spiral divergence time constant",[15] but provides the equation below to estimate the spiral divergence.

American Institute of Aeronautics and Astronautics

$$\tau_{spiral} = \frac{mV}{Sq} \left[\frac{C_{l_p} C_{n_\beta}}{C_{y_\phi}(C_{l_\beta} C_{n_r} - C_{n_\beta} C_{l_r})} \right] \tag{58}$$

It should be noted that these are approximation methods with a significant number of assumptions being made. If possible, it is better to use the full equations of motion.

ESTIMATED STABILITY DERIVATIVES

When trying to evaluate the dynamic stability characteristics of a vehicle, there are probably stability derivatives not known. SUAVE has added approximation methods for some of the easier to estimate derivatives based on the methodology listed below. All of these method come from Blakelock.[15]

If we assume that the thrust is constant with change in the perturbational x velocity and $\frac{\partial C_D}{\partial u} = 0$, then

$$C_{x_u} = -2C_D \tag{59}$$

If there is a transonic Mach number, then this will not be accurate. Neglecting the thrust in the z-direction, the change in force in the Z-direction with respect to velocity can be calculated as

$$C_{z_u} = -2 \, C_L - V \, \frac{\partial C_L}{\partial u} \tag{60}$$

where this equation will account for transonic effects. When determining how a disturbance in the flow changes the forces, since the axes are fixed to the aircraft, the resulting coefficients for change in forces in the x-direction and z-direction are

$$C_{x_\alpha} = C_L - \frac{\partial C_D}{\partial \alpha} \tag{61}$$

$$C_{z_\alpha} = -C_D - \frac{\partial C_L}{\partial \alpha} \tag{62}$$

To account for the time lag it takes for the downwash change due to change in lift on the main wing to affect the horizontal tail, $C_{m_{\dot{\alpha}}}$ is necessary. This parameter can be estimated using

$$C_{m_{\dot{\alpha}}} = \frac{2C_{m_i} \frac{d\epsilon}{d\alpha} l_t}{c} \tag{63}$$

The same time lag affecting the z-force necessitates $C_{z_{\dot{\alpha}}}$ to be calculated with the following equation.

$$C_{z_{\dot{\alpha}}} = 2C_{m_i} \frac{d\epsilon}{d\alpha} \tag{64}$$

The pitching velocity results from the curvature of the flight path causing a change in the angle of attack of the tail. This effect will affect the pitching moment and force in the z-direction on the body. These quantities are estimated with

$$C_{m_q} = 2 \frac{1.1 C_{m_i} l_t}{c} \tag{65}$$

$$C_{z_q} = 2 \cdot 1.1 C_{m_i} \tag{66}$$

While there are a large amount of different approximations available for estimating different longitudinal stability derivatives, the availability of methods for lateral-directional method is significantly less. When the aircraft is rolling, the lift vector is tilted forward on the down-going wing and rearward on the up-going wing. This results in a negative yawing moment for a positive roll rate. If there is an elliptic lift distribution, then C_{n_p} can be approximated using

$$C_{n_p} = -\frac{C_{L_w}}{8} \left(1 - \frac{d\epsilon}{d\alpha} \right) \tag{67}$$

When an aircraft is yawing, the relative flow over the left and right wings is non-symmetric. This generates a lift imbalance and by integrating over the wing and assuming an elliptic lift distribution results in the following approximation.

American Institute of Aeronautics and Astronautics

$$C_{l_r} = \frac{C_{L_w}}{4} \tag{68}$$

C_{n_r} also stems from the same phenomenon along with lift generated by the vertical stabilizer. η_v accounts for the loss of efficiency due to interference between the vertical tail and the fuselage. With these characteristics in mind, C_{n_r} is estimated by:

$$C_{n_r} = \frac{C_{D_w}}{4} - 2\frac{S_v}{S}\left(\frac{l_v}{b}\right)^2 C_{L_{v_\alpha}} \eta_v \tag{69}$$

To understand how the force in the y-direction varies with changing angles of interest for the lateral case, the methods available are

$$C_{y_\phi} = C_L \tag{70}$$

$$C_{y_\psi} = C_L \ tan\theta \tag{71}$$

It should be remembered that these are all approximation methods with different assumptions and valid configurations. Most of these methods should only be used for tube-and-wing configurations with some of these equations requiring even more restrictive assumptions.

D.   Propulsion

Computation of the propulsion performance such as thrust and fuel consumption for different aircraft is essential for analyzing the performance of aircraft. A variety of different systems are required to model the different aircraft existing today. In SUAVE this is accomplished in a general form by composing "energy networks". This allows the construction of models not only for gas turbine engines like turbofans, turboprops, and turbojets, but also for battery-electric, solar-electric and hybrid systems. The sections below describe the models in detail.

1.   Gas-Turbine

Gas turbine engines are used today by most aircraft. The energy network framework is used to construct gas turbine networks to model conventional propulsion systems like turbofans, turbojets and ducted fans. The components that are required to build gas turbine networks are modeled individually, and then assembled into a network that calculates thrust and fuel consumption rate. A selection of components are available, including compressors, nozzles, turbines, fans, and combustors. These contain evaluation methods to compute the energy transfer across that component. Currently, the one dimensional flow equations are solved across each component.[16,17] The model for each component will be described, and then will be shown as part of a gas turbine network.

COMPRESSOR AND COMPRESSION NOZZLE

The stagnation pressure and temperature changes across a compression nozzle, compressor, and fans are modeled using the Equations 72 and 73 where $P_{t_{in}}$ and $T_{t_{in}}$ are the stagnation pressure and temperature of the working fluid coming into the component, $P_{t_{out}}$ and $T_{t_{out}}$ are the stagnation pressure and temperature of the working fluid coming out of the component, respectively. π_d is the component pressure ratio, γ is the ratio of specific heats, and η_p is the polytropic efficiency of the component.

$$P_{t_{out}} = P_{t_{in}} \ \pi_d \tag{72}$$

$$T_{t_{out}} = T_{t_{in}} \ \pi_d^{((\gamma-1)/(\gamma \ \eta_p))} \tag{73}$$

EXPANSION NOZZLE

Expansion nozzles use the same equation as compression nozzles to compute the stagnation pressure drop across the component. For the temperature drop however the computation is slightly different. This is shown in Equation 74 where $T_{t_{in}}$ is the stagnation temperature of the working fluid coming into the component, $T_{t_{out}}$ are the stagnation temperature of the working fluid coming out of the component, π_d is the component pressure ratio, γ is the heat capacity ratio, and η_p is the polytropic efficiency of the component.

$$T_{t_{out}} = T_{t_{in}} \ \pi_d^{((\gamma-1) \ \eta_p/(\gamma))} \tag{74}$$

TURBINE

For turbines, the energy drop across the turbine is computed based on the work done by the associated compressors/fans that the turbine powers. This energy drop is then used to compute the change in the stagnation quantities of the working fluid as shown in Equations 75, 76 and 77. In the equations, Δh_t is the work done across the turbine, f is the fuel-to-air ratio, η_{mech} is the shaft mechanical efficiency, $W_{compressor}$, and W_{fan} are the work done across the compressor and the fan, C_p is the specific heat at constant pressure and α is the bypass ratio of the gas turbine engine.

$$\Delta h_t = -\frac{1}{(1+f)} \ \frac{1}{\eta_{mech}} \ ((W_{compressor}) + \alpha \ (W_{fan})) \tag{75}$$

$$T_{t_{out}} = T_{t_{in}} + \Delta h_t/C_p \tag{76}$$

$$P_{t_{out}} = P_{t_{in}} \left(\frac{T_{t_{out}}}{T_{t_{in}}}\right)^{(\gamma/(\gamma-1)\eta_p)} \tag{77}$$

COMBUSTOR

The flow across the combustor uses the specific heat of the fuel selected, the specified turbine inlet temperature as well as the combustor inlet temperature to compute the fuel-to-air ratio of the engine. Equation 78 illustrates how the fuel-to-air ratio is computed in the combustor. In Equation 78, h_{t_4} is the stagnation enthalpy of the working fluid at the exit of the combustor (normally turbine inlet) computed from the turbine inlet temperature, $h - t_{in}$ is the inlet stagnation enthalpy of the working fluid, η_b is the burner efficiency, h_{t_f} is the specific energy of the fuel and f is the fuel-to-air ratio.

$$f = \frac{h_{t_4} - h_{t_{in}}}{\eta_b \ h_{t_f} - h_{t_4}} \tag{78}$$

The exit stagnation pressure is computed using the inlet stagnation pressure and burner pressure ratio using the same equation used to compute the exit stagnation pressure across the compressor.

GAS TURBINE NETWORKS

The different network components are coupled appropriately to build accurate models of the gas turbine propulsion system.

TURBOFAN

For the turbofan engine, an inlet nozzle, low and high pressure compressors, a fan, combustor, low and high pressure turbines and exit nozzles are used to generate a model. The network Figure 5 indicates how the different components are connected to each other. The data flow is very similar to the turbojet (Subsection 1 described above. The difference here is that from the inlet nozzle, a significant quantity of the air flow is directed to the fan and bypasses the core completely. The stagnation quantities at the exit of the inlet nozzle are updated for a fraction of the air in the fan component. These are then passed on to the fan nozzle component where again they are updated. The stagnation quantities at the exit of the fan nozzle and the core nozzle along with the fuel-to-air ratio computed at the combustor are used to compute the thrust of the engine.
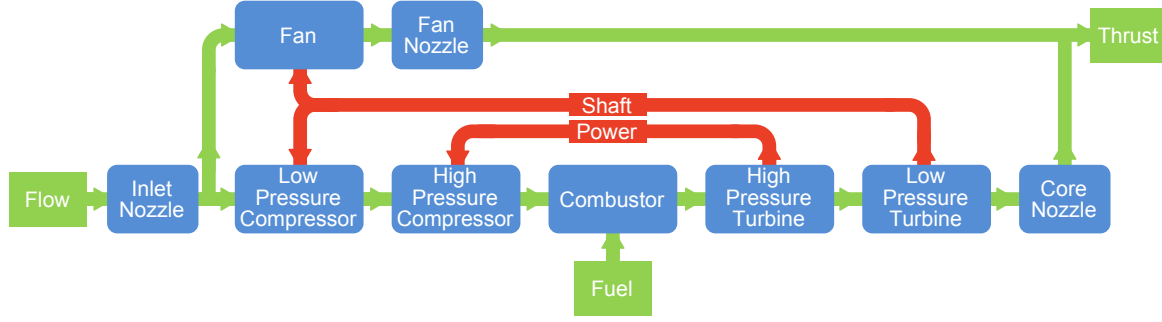
TURBOJET

Figure 5: Turbofan Network

A turbojet model shown originally in Figure 1 is similar to the turbofan except that that the fan and fan exit nozzle are absent from the network. The freestream data is input into the inlet diffuser component. The stagnation quantities are computed and the passed to the low pressure compressor and then the high pressure compressor. The change in the stagnation quantities and the work done by the compressors are computed by these components. Then the stagnation quantities are passed in to the combustor. Here based on the turbine inlet temperature specified by the user, the fuel-to-air ratio is computed and the stagnation flow quantities are updated. These are then passed into the turbine. The work done by the compressors are also passed in to the turbine and the stagnation quantities at the turbine exit are computed and these are passed on to the exit nozzle. Finally the inlet and exit stagnation quantities of the network and the fuel-to-air ratio are used to compute the engine performance characteristics.

DUCTED FAN

The ducted fan model consists of just a fan with its inlet and exit nozzle as originally shown in 3. The freestream conditions are inserted into the inlet nozzle of the propulsor where the stagnation quantities at the inlet and exit are computed. Then the exit stagnation quantities are fed into the fan from where they then go to the outlet nozzle. The stagnation quantities are then used to compute the thrust generated by the propulsor. The shaft power required to run the fan can be generated by an internal combustion engine, a combustor turbine combination (as in the case of the turbofan), or by a battery motor combination.

2. *Electric Networks*

The development of an analysis tool for arbitrary energy networks allows the composition of unconventional systems. This section will describe the relationships and networks to model electric propulsion systems.

ELECTRIC MOTOR

If necessary RPM and torque for a given motor are known, efficiency may be estimated with a motor map. The motor map included in Figure 6 uses several user selected parameters to create a fit for motor electrical efficiency. Equation 79 is used for the fit. Equation 80 is manipulated to arrive at C_i values.[18] Here the caret denotes the value at maximum efficiency, with Ω as RPM, Q as torque, η as efficiency, and k_0 as the parasite loss ratio.

$$P_L = C_0 + C_1\Omega + C_2\Omega^3 + C_3Q^2 \tag{79}$$

$$\eta = \frac{\Omega Q}{\Omega Q + P_L} \tag{80}$$

$$C_0 = \frac{k_0\hat{\Omega}\hat{Q}(1-\hat{\eta})}{6\hat{\eta}} \tag{81}$$

$$C_1 = \frac{-3C_0}{2\hat{\Omega}} + \frac{\hat{Q}(1-\hat{\eta})}{4\hat{\eta}} \tag{82}$$

American Institute of Aeronautics and Astronautics

$$C_2 = \frac{C_0}{2\hat{\Omega}^3} + \frac{\hat{Q}(1 - \hat{\eta})}{4\hat{\eta}\hat{\Omega}^2} \tag{83}$$

$$C_3 = \frac{\hat{\Omega}(1 - \hat{\eta})}{2\hat{Q}\hat{\eta}} \tag{84}$$
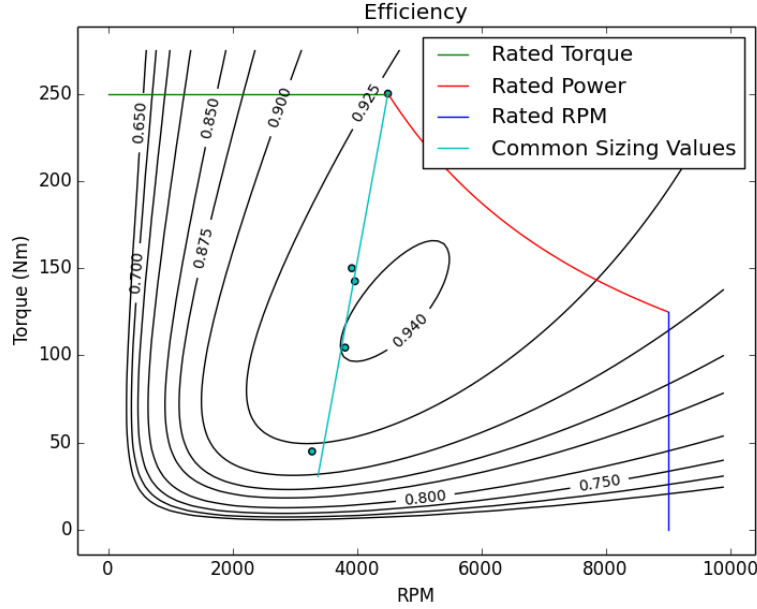

Figure 6: Sample Motor Map

Fuel Cell

Multiple fuel cell models are currently implemented within SUAVE. The first simply models applies as an efficiency factor, which scales the input power requirements, then calculates the mass flow rate of the "fuel" (for instance Hydrogen). The following equation[19] is applied,

$$\dot{m} = \frac{MW_{H_2}}{2 \cdot V_{0_{H_2O}} \cdot F} \cdot P \tag{85}$$

where $V_{0_{H_2O}}$ is the reaction potential of water, MW_{H_2} is the molar weight of hydrogen, F is Faraday's constant, P is the power requirement. The factor 2 corresponds to the number of moles of hydrogen per mole of water in the reaction,

$$2H_2 + O_2 \rightarrow 2H_2O. \tag{86}$$

Higher-fidelity fuel cell models are also included, which outline the power required in Equation 85 in more detail. Voltage is calculated according to the method below,[19]

$$V = E_{0c} + ir - A \cdot \ln(i) + m \cdot \exp(i \cdot n), \tag{87}$$

where i is the current density in A/m^2, r is the resistance times the area, and m, n, and A are constants derived from empirical data.[19] E_{0c} is further defined with,

$$E_{0c} = E + A \cdot \ln(i_0), \tag{88}$$

American Institute of Aeronautics and Astronautics

where E is the open current voltage of the cell (1.2V for a hydrogen fuel cell), and i_0 is the exchange current density. The power is then determined from the well known power-voltage relation,

$$P = IV. \tag{89}$$

Fuel cell weight may either be calculated from an input specific power, or can be initialized using the following algorithm: a maximum power output is prescribed by the user, and divided by the maximum power output for a single cell determined from Equation 89. The volume of the fuel cell stack is then calculated from,

$$Vol = N_{cell} \cdot t_{wall} \cdot A_{interface}, \tag{90}$$

where N_{cell} is the number of cells, t_{wall} is the wall thickness, and $A_{interface}$ is the area of the cell interface (t_{wall} and $A_{interface}$ are prescribed). The total fuel cell weight is then calculated with,

$$W_{fuelcell} = Vol \cdot \rho \cdot \zeta, \tag{91}$$

where ρ is the cell density and ζ is the cell porosity.

BATTERY DISCHARGE

As of this writing, one battery discharge loss model exists within SUAVE, although the code is designed such that a new discharge model can be easily defined and assigned to the battery object. Of note is that the current model is only valid for lithium ion-like batteries. Batteries possessing a significantly different Peukart constant require an alternative approach to approximating the discharge losses. The default discharge model is empirically based. A basic formulation is repeated below for reference.[20]

$$f = 1 - \exp(-20x) - \exp(-20(1-x)) \tag{92}$$

$$R = R_0(1 + C \cdot f) \tag{93}$$

$$P_{discharge} = I^2 \cdot R \tag{94}$$

In the above, x is the state of charge of the battery, R_0 is a baseline internal resistance, and C is the discharge rate of the battery. Furthermore, different battery chemistries and technology levels tend to have different characteristic curves of specific power vs. specific energy (called Ragone Plots) that can be modeled to determine the "optimal" (from a mass perspective) battery for a given mission, knowing the maximum power and energy requirements from the corresponding mission.

A sample of different Ragone plots for different battery types can be seen in Figure 7. Note that additional characteristic curves may be easily defined by the user depending on the data available.

MASS GAINING BATTERIES

In addition to discharge losses, certain kinds of batteries (e.g. metal air) increase in mass as they discharge as oxygen particles accumulate on the cathode. While the specifics of individual batteries vary (due to varying battery chemistry), the results for a lithium oxygen battery are shown below.[21]

$$2Li + O_2 \Rightarrow Li_2O_2 \tag{95}$$

$$\dot{m} = \frac{MW_{O_2}}{V_{0_{Li_2O_2}} \cdot F} \cdot P \tag{96}$$

This basic approach is used to model the mass accumulation rate for other battery chemistries by replacing the factor,

$$\frac{MW_{O_2}}{V_{0_{Li_2O_2}} \cdot F}, \tag{97}$$

with a simple "mass gain factor," which accounts for material accumulated by any variable-mass battery from the atmosphere. Other materials that react with the battery (e.g. the external H_2O required for an aluminum air battery), that must be carried by the aircraft, are modeled as functions of the individual battery class itself. Currently, SUAVE includes models for lithium-air and aluminum-air batteries.
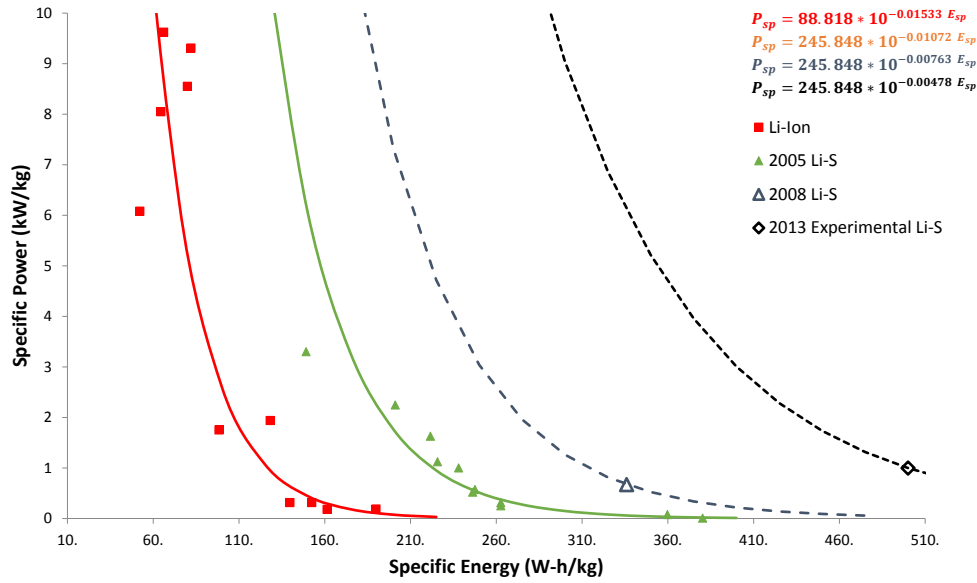
American Institute of Aeronautics and Astronautics

$P_{sp} = 88.818 * 10^{-0.01533\ E_{sp}}$
$P_{sp} = 245.848 * 10^{-0.01072\ E_{sp}}$
$P_{sp} = 245.848 * 10^{-0.00763\ E_{sp}}$
$P_{sp} = 245.848 * 10^{-0.00478\ E_{sp}}$

■ Li-Ion
▲ 2005 Li-S
△ 2008 Li-S
◇ 2013 Experimental Li-S

Figure 7: Ragone Plots

SOLAR

For solar aircraft the power generation was broken into two steps, first was the determination of the solar radiation and the second was the calculation of the solar energy captured by the solar panels. The solar radiation model incorporates a homogeneous spherical model with an elevator observer for the establishment of air mass effects. This method sets the solar radiation to be 1367 W/m^2 at the edge of the atmosphere. The position and orientation of the vehicle relative to the Sun is accounted for in determining the radiation.[22] This includes pitch and roll of the vehicle.[23] Additionally, the time of day and the day of the year are accounted for in determining the orientation of the Sun relative to the vehicle. The solar panels operate through efficiency and areas to calculate power outputs. The power is routed through some logic and a Maximum Power Point Tracker (MPPT). For low levels of fidelity the MPPT are modeled by an efficiency factor.

PROPELLER

Two different levels of fidelity exist for propeller models. The lowest level of fidelity is purely an efficiency applied to the shaft power to make thrust power and thus thrust. The higher level of fidelity incorporates a blade element momentum routine for the calculation of thrust. The methodology is based on the methodology presented by Theodorsen, and is very similar to the approach from Mark Drela's Qprop.[24] Airfoil data can be provided and is scaled for Reynolds and Mach number effects.

Additionally, there is a method for designing propellers based on the methodology presented by Adkins and Liebeck.[25] This routine varies slightly from the Qprop methodology, but produces similar results and provides a reassuring verification. The Reynolds and Mach scaling for the airfoil data is also integrated into this design routine.

SOLAR NETWORK

A network of components was created for solar powered vehicles. The most critical component is the combined MPPT and Logic component. This organizes the flow of power throughout the network. The linkages of components are illustrated below in a diagram.
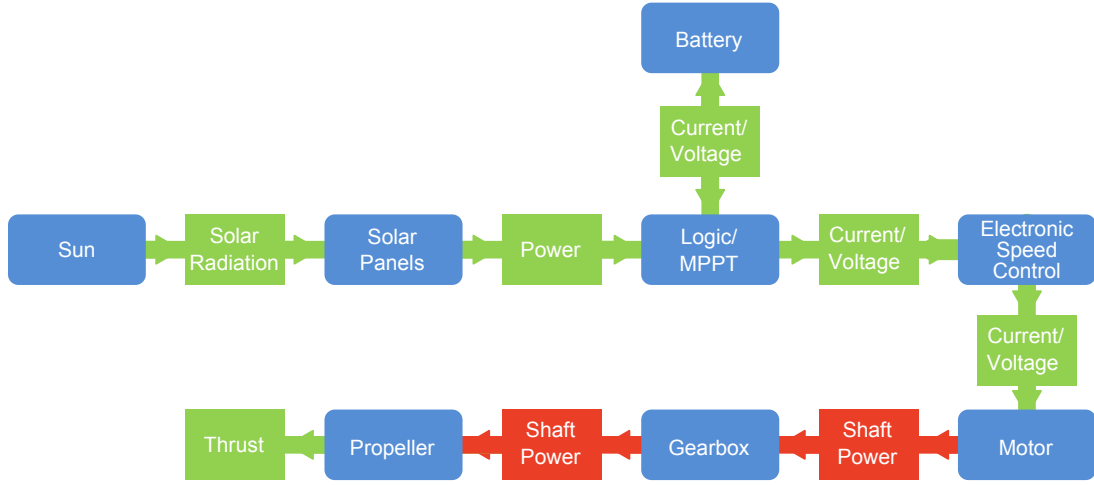
American Institute of Aeronautics and Astronautics

Figure 8: Solar Powered Aircraft Network

E. Numerics

SUAVE implements a pseudospectral collocation method based on Chebyshev polynomials and differentiation. This is particularly needed for solving the equations of motion for a mission segment, though is implemented generally and available for use elsewhere in the code.

Pseudospectral collocation is well-suited to general boundary-value problems and flexible enough to remain robust with any well-posed set of governing equations and boundary (or initial) conditions. The core physics can range from simple constraining the forces on the vehicle to be constant (for example, in a level cruise Segment) to an integration of the general three-dimensional equations of motion within the same solver framework. Pseudospectral methods also allow for high accuracy and spectral convergence rates with a small number of control points, a key savings in computational cost when complex physical simulations are required to produce the supporting data.

SUAVE's pseudospectral collocation method is based upon the formation of generic differentiation D and integration I matrices for a given number of control points, N, in a mission segment. It is worth emphasizing that the process of forming D and I are only executed once in a pre-processing step with negligible computational expense. It can be shown that using cosine-spaced control points produces optimal accuracy and convergence rates when Chebyshev polynomials are used as basis functions. D and I are both fully-dense and, in SUAVE, are scaled to an independent variable range of zero to one. With this formulation, D and I can be used to differentiate or integrate any cosine-spaced variable via the proper scaling, most commonly temporal or spatial operations, in SUAVE. Differentiation in physical units, therefore, is $D/(x_f - x_0)$ while integration is $I \cdot (x_f - x_0)$, where x_f and x_0 represent the final and initial values of the independent variable, respectively. Since these operators are stored in a mission segment they can be leveraged by users when adding functionality that requires differentiation or integration.

The operators above provide a robust means to define the relevant equations of motion for a Segment, transforming the governing equations to a discrete and (generally) nonlinear $N \times N$ system that requires solution. This is solved via SciPy's nonlinear solution algorithm, a wrapper of MINPACK's "hybrid" algorithm[26] implemented in Fortran. An exact Jacobian is provided via complex step differentiation of the Segment's equations of motion. The authors note that this comes at a price: while efficient and robust within the confines of SUAVE, calls to user-defined functions or those which deal with more sophisticated physics may not accept complex arguments.

F. Frames of Reference

There are three frames that are needed to make dynamics calculations reasonably general yet convenient – Inertial, Body, and Wind. These frames are all centered at the aircraft's center of gravity

American Institute of Aeronautics and Astronautics

The inertial frame is chosen such that the **x-y** plane is oriented parallel to a flat ground plane, and the z-axis points downward. Two vector quantities are conveniently defined in this frame. First is the weight vector, which points in the +**z**-direction with the direction of gravity. Second is the vehicle's velocity vector.
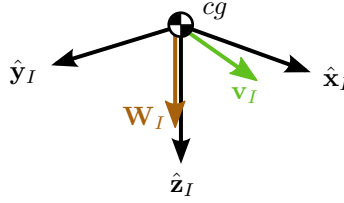

Figure 9: Inertial Frame

It is convenient to define the velocity vector in the inertial frame because a combination of the velocity vector and the body frame orientation will later define the orientation of the wind frame.

The body frame is aligned with the vehicle body, usually with the +**x**-direction pointing out the nose, the +**y**-direction pointing out the starboard wing, and the +**z**-direction pointing down.
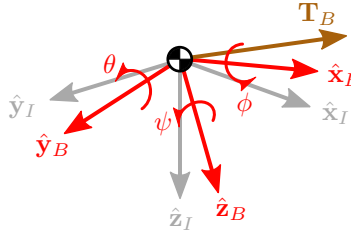

Figure 10: Body Frame

The thrust of the aircraft is conveniently described in this frame since the propulsion system usually rotates with the aircraft. The thrust's line of action may point in any direction in this frame.

The orientation of this axis can be described relative to the inertial frame with Euler-angle rotations $\psi - \theta - \phi$, which describe successive yaw-pitch-roll rotations around the $\mathbf{z} - \mathbf{y} - \mathbf{x}$ axes respectively. These rotations can be encoded in a 3×3 transformation matrix, and can be composed of the matrix products of component rotations around each axis. These component rotations are built as,

$$
\boldsymbol{T_\mathbf{x}}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}, \quad \boldsymbol{T_\mathbf{y}}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}, \quad \boldsymbol{T_\mathbf{z}}(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.
$$

$$(98)$$

Given the yaw-pitch-roll angles of the body frame within the inertial frame, we can build a transformation matrix that takes any vector in the inertial frame and finds its equivalent coordinates in the rotated body frame. This matrix is built from right to left,

$$
\boldsymbol{T}_{B/I} = \boldsymbol{T_\mathbf{x}}(\phi)\boldsymbol{T_\mathbf{y}}(\theta)\boldsymbol{T_\mathbf{z}}(\psi). \tag{99}
$$

We need a transformation matrix that will find parameterizations for thrust in the inertial frame. This is found by taking the transpose of the $\boldsymbol{T}_{B/I}$ to find the transformation from the body to the inertial frame,

$$
\boldsymbol{T}_{I/B} = \boldsymbol{T}_{B/I}^\top. \tag{100}
$$

The last frame we need is the wind frame, which is found using information from both the inertial and body frames. It is typically constructed because the lift and drag aerodynamic force vectors rotate with this frame and the freestream velocity direction.

The wind frame is rotated from the body frame with two rotations - side slip angle, followed by angle of attack. The frame is constructed by first aligning the +**x**-direction parallel to the vehicle velocity vector.
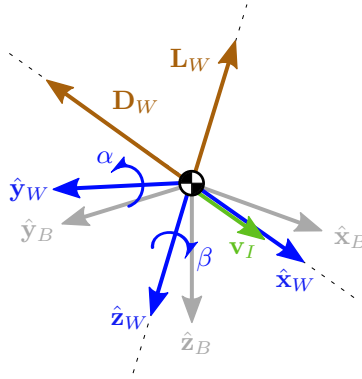
Figure 11: Wind Frame

The velocity vector is then projected into the $\mathbf{x} - \mathbf{z}$ plane of the body axis. Angle of attack is found as the angle subtended from the body frame's $+\mathbf{x}$-direction to the projected velocity vector. Side slip angle is found as the angle subtended from the projected velocity vector to the original velocity vector. Note that this is a negative angle in the right hand rule on the wind frame $\mathbf{z}$-axis shown in Figure 11. Knowing these rotations, we can finally construct a transformation matrix from the wind frame to the body frame,

$$\boldsymbol{T}_{B/W} = \boldsymbol{T}_{\mathbf{y}}(\alpha)\boldsymbol{T}_{\mathbf{z}}(-\beta) \tag{101}$$

We need a transformation matrix that will find parameterizations of the lift and drag vectors in the inertial frame. This is done by combining the wind-body transformation with the body-inertial transformation,

$$\boldsymbol{T}_{I/W} = \boldsymbol{T}_{I/B}\boldsymbol{T}_{B/W} \tag{102}$$

With this we have transformation matrices that can take forces either the body or wind frames and return their parameterization in the inertial frame, where we will resolve the equations of motion.

G. Mission

1. Mission Solving

The mission is solved by running a root-finding algorithm from the numerical Python package (numpy) called `f_solve`. This solves the equations of motion by driving residuals of the problem to zero,

$$f(\mathbf{x}) = \boldsymbol{R} = [0]. \tag{103}$$

There are three major steps in solution of a mission. First is initialization. During this step, segment inputs are unpacked, condition vectors are sized to the number control points of the segment, and then conditions are precomputed using much information as can be known before encountering open equations. Second is iteration. During this step analyses are called with given inputs $\mathbf{x}$ to calculate the residuals $\boldsymbol{R}$. Only the needed analyses for the residuals should be included here to increase speed. This step is iterated until residuals converge to zero. The third step is post-processing. This step takes the converged conditions found in the iteration step, and performs any additional analyses needed to produce useful results.

Between segments, final data can be mapped to initial data. For example the mass, time, and position at the end of the climb segment is automatically used at the initial mass, time, and position of the cruise segment.

2. Cruise

Two cruise segments are available: Constant-Speed Constant-Altitude, and Constant-Mach Constant-Altitude. Because the Constant-Mach cruise segment is at constant altitude, it is a simple extension of the constant speed segment using the speed of sound estimated by an atmospheric model. These segments are discretized in time. Currently SUAVE assumes this segment is 2-dimensional.

American Institute of Aeronautics and Astronautics

The inputs for the cruise segments are horizontal velocity or Mach number, altitude, horizontal range, initial mass, and initial time. Final time is easily calculated from range and velocity, and distance traveled at each time point is calculated by integrating velocity. We can assume accelerations are zero in this segment. Freestream conditions are calculated from the atmospheric model knowing the constant altitude.

The unknowns for this segment are the aircraft body pitch in radians and propulsion throttle ratio in fractions of 1. These are guessed and converged during the iteration step by passing them in as vector **x**. Using these inputs we can work the free equations. First using the given body pitch angle, the angle of attack is observed as identical, since flight is level. This is used to calculate the aerodynamic lift and drag using the segment's aerodynamic model. The throttle is used to calculate the thrust force and the fuel consumption rate. The fuel rate is integrated from the initial mass to yield the mass, and then the weight under gravity, at each time point in the segment.

The residuals of this segment are the total horizontal and vertical forces, which should equal zero in this steady case.

3. Climb and Descent

There are a few types of climb segments, that vary by the information available, and in turn change the way the segment is solved. It turns out that in terms of how climb and descent segments are solved, they can share identical strategies, and only vary by a sign in the change in altitude, and by extension the climb rate or angle.

The inputs, general for all climb and descent segments, are the start and end altitudes, as well as the initial mass and time. These segments are discretized in altitude in order to make certain condition variables precomputable. Each segment has different requirements in terms of of which variables are precomputable vary by segment type.

The different segment types are Constant-Speed Constant-Climb-Rate, Constant-Mach Constant-Climb-Rate, Constant-Mach Constant-Climb-Angle, and Constant-Throttle Constant-Airspeed.

All segments share a common unknown condition at each time point, the vehicle body pitch angle. For all segments except for constant throttle, the throttle fraction is unknown. For the constant throttle segment, the aircraft wind pitch angle is unknown (angle of attack in 2D flight is taken as the body angle minus the wind angle).

All segments share common residuals, the sum of forces should equal mass times acceleration.

4. Constant-Airspeed Constant-Climb-Rate

Additional inputs for this segment are the climb rate, which is the vertical velocity component, and airspeed, which is the trajectory's tangential velocity component. These two components define a wind angle. Discretizing on altitude allows this segment to pre-compute the static atmospheric conditions like pressure, temperature and speed of sound. Accelerations are defined as zero in this constant velocity segment.

During segment iteration, throttle and body angle unknowns. These are used to solve the free equations. The angle of attack is calculated as the body angle minus the wind angle. The angle of attack and the precomputed static atmospheric conditions are used by the aerodynamics model to estimate lift and drag. The throttle is used by the propulsion model to estimate thrust and the rate of change in vehicle weight. Like cruise the weight is integrated in time, but in this case an extra chain rule step is required to account for the segment's discretization in altitude. With this information the sum of forces can be calculated, which then inform the residuals.

5. Constant-Mach Constant-Climb-Rate

In this segment, instead of providing a constant airspeed, one can provide a constant freestream Mach number (in the wind frame). This segment is nearly identical to the constant-speed constant-climb-rate, with the exception of how the airspeed is precomputed. Since this segment is discretized in altitude, the atmospheric conditions including the speed of sound can be precomputed. With this, one can determine the airspeed, which now varies with a segment's progress. One can also precompute the vehicle's accelerations. From here the segment is solved identically to the constant-speed constant-climb rate segment.

6. Constant-Mach Constant-Climb-Angle

This segment is functionally a combination of the previous two segments. Again the precomputed speed of sound is used to find vehicle velocity, but now the climb angle is used to find the vertical and horizontal components. The segment proceeds from here in the same fashion as the constant-airspeed constant-climb-rate Segment.

7. Constant-Throttle Constant-Airspeed

This segment is the most representative of how a passenger jet typically climbs. It is convenient for a pilot to be able to leave the throttle in a set position. This segment takes for input the throttle fraction, as well as the airspeed. As a result, the constant-throttle segment requires special treatment. Two angles are used as unknowns, the body angle and wind angle. During iteration, the body angle is used to find the inertial velocity vector, and the wind angle is used to find the angle of attack, which allows the aerodynamics model to estimate lift and drag. Throttle is constant, but atmospheric conditions vary. To find the atmospheric conditions, we must integrate the vertical component of the inertial velocity to find how the vehicle's altitude changes with segment progress. With this information we can estimate thrust with the propulsion model, as well as vehicle weight change. Accelerations are found by differentiating on inertial velocity. With all of this information we now can inform the residuals with a force balance.

8. Takeoff and Landing

In addition to the correlation-based takeoff and landing field length analysis (see Subsection I. 1), SUAVE supports a physics-based analysis method that solves the equations of motion for the aircraft on the ground. The analysis accounts for rolling friction in addition to aerodynamic forces. This method carries a higher computational expense than the correlation method, but it is better suited to the analysis of unconventional aircraft configurations.

The takeoff and landing segments are both subclasses of a Ground Segment class that varies the aircraft velocity and rolling time to enforce residuals requiring force balance and a final velocity requirement. In the second residual, the final velocity requirement is set to the rotation speed for takeoff, or zero for landing.

H. Noise

The methodology for calculating aircraft noise in SUAVE is done using a reference state (101 dB for one, 25,000 lb thrust jet engine at 1,000 ft distance). Corrections are then made for thrust level, distance, ground attenuation, and phase of flight (whether take-off or landing). These corrections come from correlations based on data from Kroo[7]

$$\Delta Noise_{thrust}[dB] = -2.193 \cdot 10^{-4} \left(\frac{T_{SL}[lbs]}{25000[lbs]} \cdot 100 \right)^2 + 0.09454 \left(\frac{T_{SL}[lbs]}{25000[lbs]} \cdot 100 \right) - 7.30116 \qquad (104)$$

$$\Delta Noise_{landing}[dB] = -1.5766 \cdot 10^{-3} \left(\frac{T_L}{T_{SL}} \cdot 100 \right)^2 + 0.34882 \left(\frac{T_L}{T_{SL}} \cdot 100 \right) - 19.2569 \qquad (105)$$

where all thrusts are specified in lbs. The effect of the thrust level and number of engines on noise is calculated by scaling the sound intensity by the ratio of thrust to reference thrust (applied to both engine size and number of engines). Noise at takeoff, sideline, and landing are all calculated in this manner for the jet aircraft. A reduction in noise is calculated using the distance from the vehicle to the measuring station as the attenuation distance: -4 dB for takeoff, -6 dB for side line, and 9.1 dB for landing based on known distance from the aircraft to the microphone. Summing all these different corrections results in the noise value for each different flight regime required for certification. Airframe landing noise is also calculated using a relation to the log of the aircraft weight.

$$Noise_{airframe}[dB] = 40 + 10log(W[lbs]) \qquad (106)$$

The landing noise is then the combination of the engine landing noise and airframe noise both in dB. These estimates are rough approximations that work for jet aircraft that are similar to the configuration

baseline, in our case a DC-10. Noise estimates for other configurations and power sources are being considered in the future.

I. Performance

Aircraft performance includes many aspects of the airplane operation and contains the key metrics to be used for the design evaluation and optimization. Several design constraints are related to airplane performance, in order to meet both customer requirements and "14 CFR Part 25" regulation requirements. SUAVE current capabilities address the most important performance measures, including airfield and cruise performance. The following sections describe the methodologies used for level zero-fidelity estimation of the main performance characteristics of a given airplane.

1. Takeoff Field Length

For the level zero-fidelity analyses the Takeoff Field Length (TOFL) is estimated using a correlation method that relates the TOFL to parameters that have influence during the takeoff run, namely: wing area, maximum lift coefficient, takeoff weight, and engine thrust. These parameters are grouped together as proposed by Loftin,[27] but a quadratic correlation is consider instead of a linear one. This method presents different coefficients for 2, 3 and 4 engine models.[7,8] The parameterization in SUAVE was slightly modified, as presented in Equation 107, in order to allow the user to evaluate different V_2 speeds, as well as any atmospheric condition required, such as hot and/or high conditions. The resulting coefficients are presented in Table 1 and they are implemented as default values inside SUAVE. Alternatively, the user may inform the coefficients to be used to the fit, and this fit can be of any order.

$$TOFL = \sum_{i=0}^{n} k_i \left[\frac{V_2^2}{T/W} \right]^i \tag{107}$$

where V_2 is velocity in m/s, T thrust in Newtons and W is weight in kg.

Table 1: Takeoff coefficients as a function of the number of engines

Engines	k_0	k_1	k_2
2	857.4	2.476	1.40e-4
3	667.9	2.343	9.30e-5
4	486.7	2.282	7.05e-5

2. Landing Field Length

The same correlation approach is used for the Landing Field Length (LFL) estimation, where the principal parameter considered here is the approach speed, as showed in Equation 108. The method used is the one proposed in Torenbeek,[28] that uses a quadratic correlation between LFL and V_{app}, and presents different coefficients for two-wheel and four-wheel trucks. By default, SUAVE uses the two-wheel trucks coefficients, since this is the most common type of landing gear. Similar to the takeoff module, the default values are presented in Table 2, and the user is allowed to input the most applicable coefficient to a given design, as well as the order of the fit.

$$LFL = \sum_{i=0}^{n} k_i \left[V_{app}^2 \right]^i \tag{108}$$

where V_{app} is the approach velocity in m/s.

American Institute of Aeronautics and Astronautics

Table 2: Landing coefficients as a function of the number of wheels truck

Wheel Trucks	k_0	k_1	k_2
2	250	0.0	0.2533
4	250	0.0	0.3030

3. Payload Range Diagram

One key feature to evaluate an airplane cruise performance is the payload range diagram, which presents the trade-off relationship between the payload carried on-board and the maximum possible range for that payload. A typical payload range diagram is illustrated in Figure 12, and is composed by three elementary points calculated as described in Table 3:

Figure 12: Payload Range Diagram

Table 3: Three Points in a Payload Range Diagram

Point	Payload	TOW	Fuel
1	MaxPLD	MTOW	MTOW - OEW - MaxPLD
2	MTOW-OEW-MaxFuel	MTOW	MaxFuel
3	0.0	OEW + MaxFuel	MaxFuel

Since SUAVE's mission profile is flexible and can be defined by the user, in order to compute the payload range diagram the user must inform which segment shall be changed in order to match the fuel consumption of each point of the diagram. The same convergence approach presented in Section III.I.4 is used.

4. Fuel Consumption Sizing Loop

Due to SUAVE modular capability the in mission profile, as described in Section III.G, the mission convergence is made in a single segment base. Therefore, to valuate a mission with a desired total range one must iterate the mission evaluation by changing a given segment (usually a cruise segment) in order to have the

American Institute of Aeronautics and Astronautics

sum of the mission ranges of all segments equal to the desired total range. The same approach is used for mission based in total fuel burn.

SUAVE solves the input mission profile and it checks the desired range or fuel against the total range or fuel resulting from the mission evaluation. Then, a mission segment informed by the user is changed based in the cruise specific range and the residual between target and current range or fuel, and the iteration goes until minimum residual criteria is achieved.

5. Second Segment Climb Gradient

The correlation method used to determine the TOFL does not take into account the required climb gradient imposed by "14 CFR Part 25" requirements. In cases of low climb capability, such as in hot and high airfields, the airplane may not be able to comply with this requirement using the most deflected takeoff flaps. To check airplane compliance to this requirement, one has to estimate the climb gradient of the airplane in the V_2 speed, based in the Equation 109 below:

$$\gamma = \frac{T - D}{W} = \frac{T}{W} - \frac{1}{L/D} \tag{109}$$

The engine thrust is obtained from the engine model described in Section III.D, and the weight is an input variable. The estimation of the L/D ratio in the takeoff configuration may be a very complex task if done bottom-up. The SUAVE approach starts from the estimation of the L/D in the takeoff configuration as a function of AR and $C_{L_{V2}}$, based on the data presented by Obert,[29] modified to the following equation:

$$\frac{L}{D} = 7.265\sqrt{AR} - 6.464 \cdot C_L \tag{110}$$

This L/D does not account for the engine failure associated drag, thus one must add windmilling and asymmetry trim drag. The windmilling drag[30] is estimated based in nacelle wetted area according to Equation 111:

$$C_{D_{WM}} = \frac{7.274 \cdot 10^{-3} \ S_{wet-NAC}}{S_{ref}} \tag{111}$$

For the asymmetry trim drag, Equation 112 is applied from Shevell:[8]

$$\Delta C_{D_{trim}} = \frac{y_{eng}^2 (T + D_{windmill})^2}{q \pi h_V^2 l_V^2} \tag{112}$$

Finally, the L/D at the V_2 speed is calculated through the sum of equations 111 and 112 into Equation 110. After computing the climb gradient using Equation 109 for a given condition, the resulting value is compared with "14 CFR 25.121" requirement based in airplane number of engines.

American Institute of Aeronautics and Astronautics

IV. Case Studies

When building a new conceptual tool, it is important that its models match existing tools as well as available real-world data. This section presents a Verification and Validation (V&V) study with six test cases, including three conventional designs modeled primarily with correlation-based (zero-fidelity) methods, and three unconventional designs modeled in some cases with physics-based methods.

A. Canonical Zero-Fidelity

1. Medium Range Jet - Boeing 737-800

A representative model for the Boeing 737-800 is created in SUAVE based on publicly available data[31] and its flight performance is simulated. Some of the design parameters for the aircraft are shown in Table 4. The performance estimates obtained from SUAVE are then compared with published data and data from other aircraft design tools obtained during a project (FAA Partner Project 43) that two of the authors were apart.

Table 4: B737-800 design parameters

Design Parameter	Value
Wing reference area	1334 ft^2
Wing span	117 ft
Wing sweep	25°
Cruise Mach number	0.78
MTOW	174200 lb

For this particular test case, the flight of the B737-800 over a range 2950 nautical miles at a cruise Mach number of 0.78 is simulated. In order to ensure that the mission is accurately modeled, the mission is constructed using four climb segments, one cruise segment and five descent segments. The B737-800 mission uses the fidelity-zero aerodynamics module, the network based gas turbine for its propulsion, the weights module, the take-off and landing field length module, and the noise module.

Some of the important mission parameters are shown in Table 5. The fuel burn for the mission predicted by SUAVE is within 4% of the value predicted by the other tools. The value of the lift coefficient C_L and the lift to drag ratio (L/D) also lie within a few percent of the values predicted by the other tools. Table 5 also shows that the engine parameters, cruise thrust, and the specific fuel consumption (sfc) are predicted accurately by SUAVE with the predicted sfc within 3% of the value predicted by the other tools.

Table 5: Comparison of performance data for the B737-800

Variable	Design Tool 1	Design Tool 2	Design Tool 3	SUAVE
Fuel Burn (lb)	38200	39500	38400	39400
L/D (beginning of cruise)	18.8	16.2	17.9	16.5
Cruise C_L	0.6	0.57	0.57	0.58
Cruise thrust per engine (lbf)	4530	5190	6070	5000
Cruise sfc (lb/lbf-hr)	0.65	0.61	0.62	0.63

Next, we compare the weight breakdown of the aircraft computed by SUAVE with data from the other tools as shown in Table 6. The maximum takeoff gross weight (MTOW) and the operating empty weight (OEW) are within 4% of the values predicted by the other tools which clearly indicates that the weight prediction for conventional aircraft is accurate. Moreover, individual component weights (wing, fuselage and engine) presented in 6 are also within a few percent of the other tools' predictions.

Subplot 1 of Figure 13 shows how the aircraft mission in SUAVE is modeled. The aircraft climbs to 35000 ft where it cruises at constant altitude. We also see in Figure 13 subplot 2 how the weight of the aircraft decreases over the course of the mission as a result of the fuel being burnt. The specific fuel consumption

American Institute of Aeronautics and Astronautics

Table 6: Comparision of the Weight breakdown for the B737-800

Variable	Design Tool 1	Design Tool 2	Design Tool 3	SUAVE
MTOW (lb)	174,000	171,500	174,200	174,200
OEW (lb)	91,700	87,100	91,100	88,100
Wing Weight (lb)	14,300	17,100	16,800	16,900
Fuselage Weight (lb)	16,900	18,900	15,800	15,000
Engine Weight (lb)	16,500	12,200	16,800	15,100

over the course of the mission varies significantly. At takeoff and climb, the sfc values are low owing to the higher density and mass flow of air as shown in Figure 13 subplot 3. The significantly higher thrust as shown in Figure 14 subplot 2 , however contributes to the large fuel burn. The plot of throttle with time in Figure 14 subplot 3 clearly verifies this with the throttle maximized for the climb segments.

During cruise, the sfc is much higher indicating poorer engine performance but the small cruise thrust requirement ensures a lower fuel burn rate. The lower cruise thrust requirement is because at cruise, the thrust only needs to stay equal to the vehicle drag as shown in 14 subplots 1 and 2. Once the aircraft starts descending, the thrust drops down significantly while the drag increases.

The lift coefficient plotted in Figure 15 subplot 1 clearly shows that the C_L increases significantly at the takeoff segment and then remains high during the climb segment. The angle of attack (shown in 15 subplot 3) during the climb segments is extremely high as the mission solver generates the required aircraft lift by increasing the angle of attack. This is also reflected in the drag breakdown for the aircraft as shown in Figure 16 where for the climb segment, lift induced drag is the major contributor to the total vehicle drag. As the climb segments progress and the mission nears the cruise segment, the parasite drag becomes dominant. The compressibility drag of the aircraft also increases at cruise.

At the end of the final climb segment, the throttle command becomes greater than 100%. This is a feature that is built into SUAVE to permit smooth quantities of interest. If the throttle was clamped at 100% for example, there is the potential that the segment would not be able to find a solution for the equations of motion, which would compromise the solution of the rest of the mission, and return a ambiguous or discontinuous result to the optimizer. Care for constructing smooth behaviors, by permitting infeasible results, is a major component of SUAVE that will support the optimization results to be presented in future publications.

During the cruise segment, the C_L and C_D do not vary much as shown in Figures 15 and 16. For the descent segment, again the C_L is shown to increase corresponding to an increase in the angle of attack. This is because, the effect of spoilers are not currently modeled and so in order to increase the drag and permit descent at the specified rate of descent, the solver increases the vehicle angle of attack thus increasing the induced drag.

Overall, we see that SUAVE is able to reasonably model the different segments of the mission and with remarkable detail for a conceptual design environment.
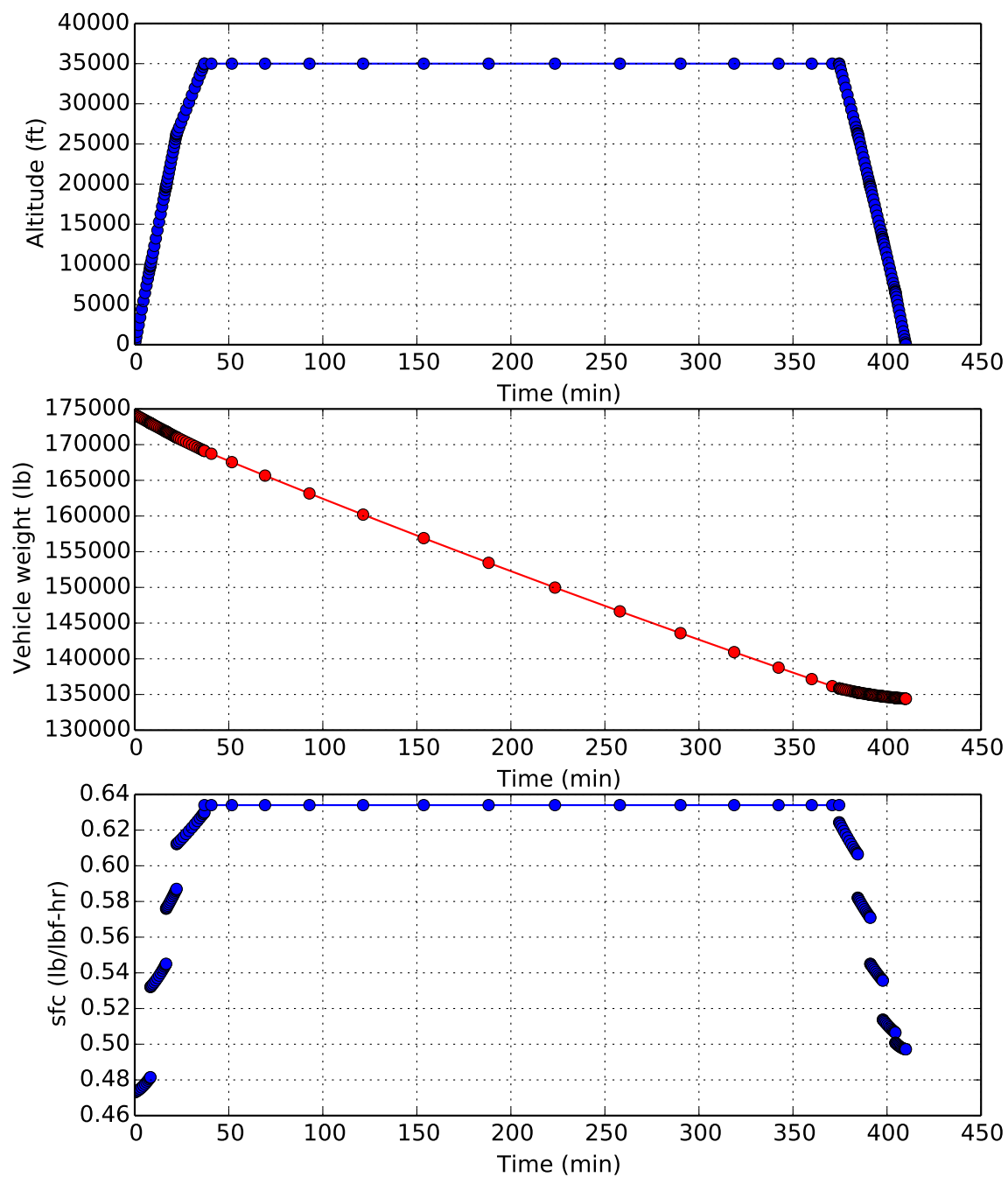
American Institute of Aeronautics and Astronautics

Figure 13: Mission parameters for the B737-800

Figure 14: Force Breakdown for the B737-800

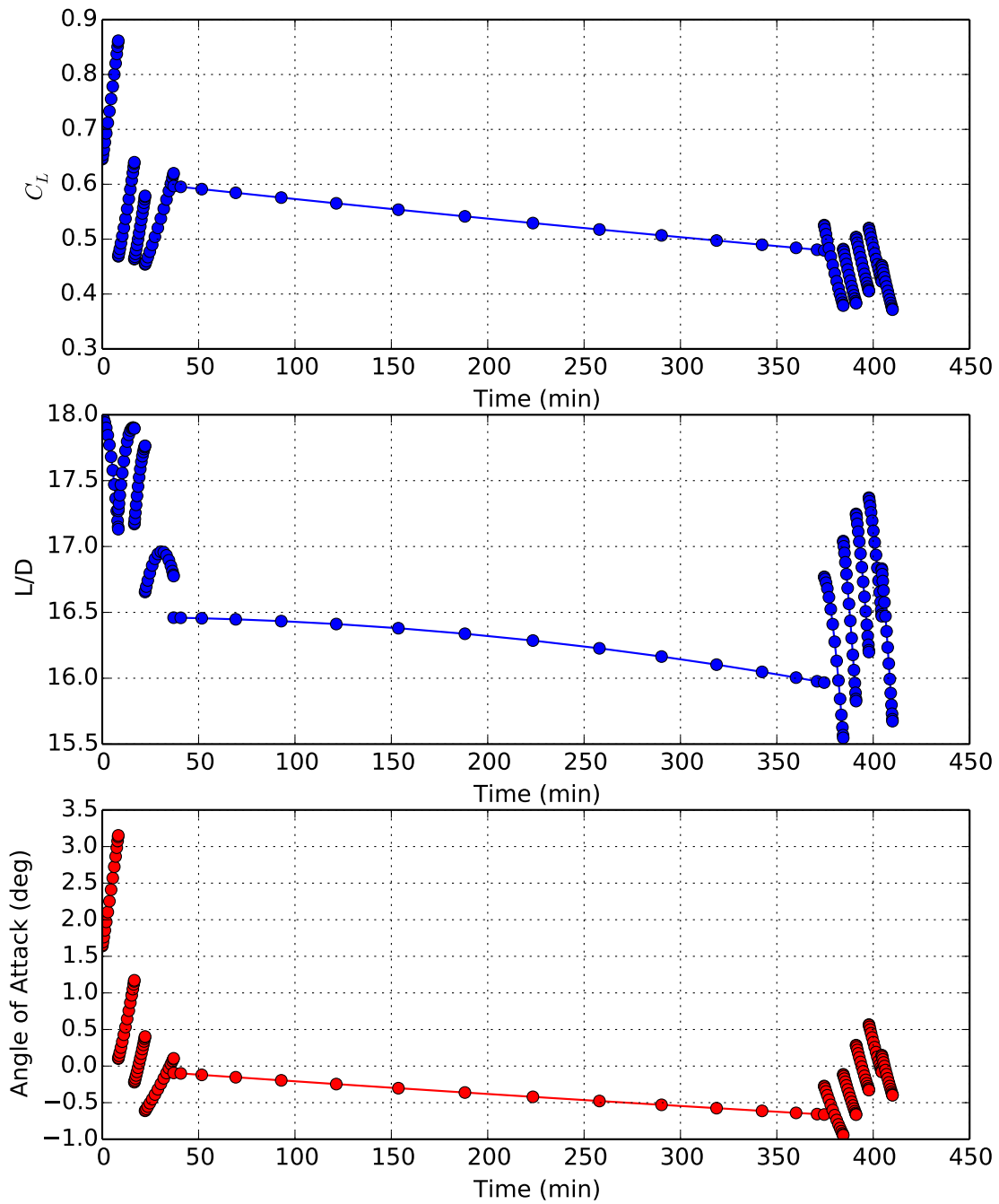American Institute of Aeronautics and Astronautics
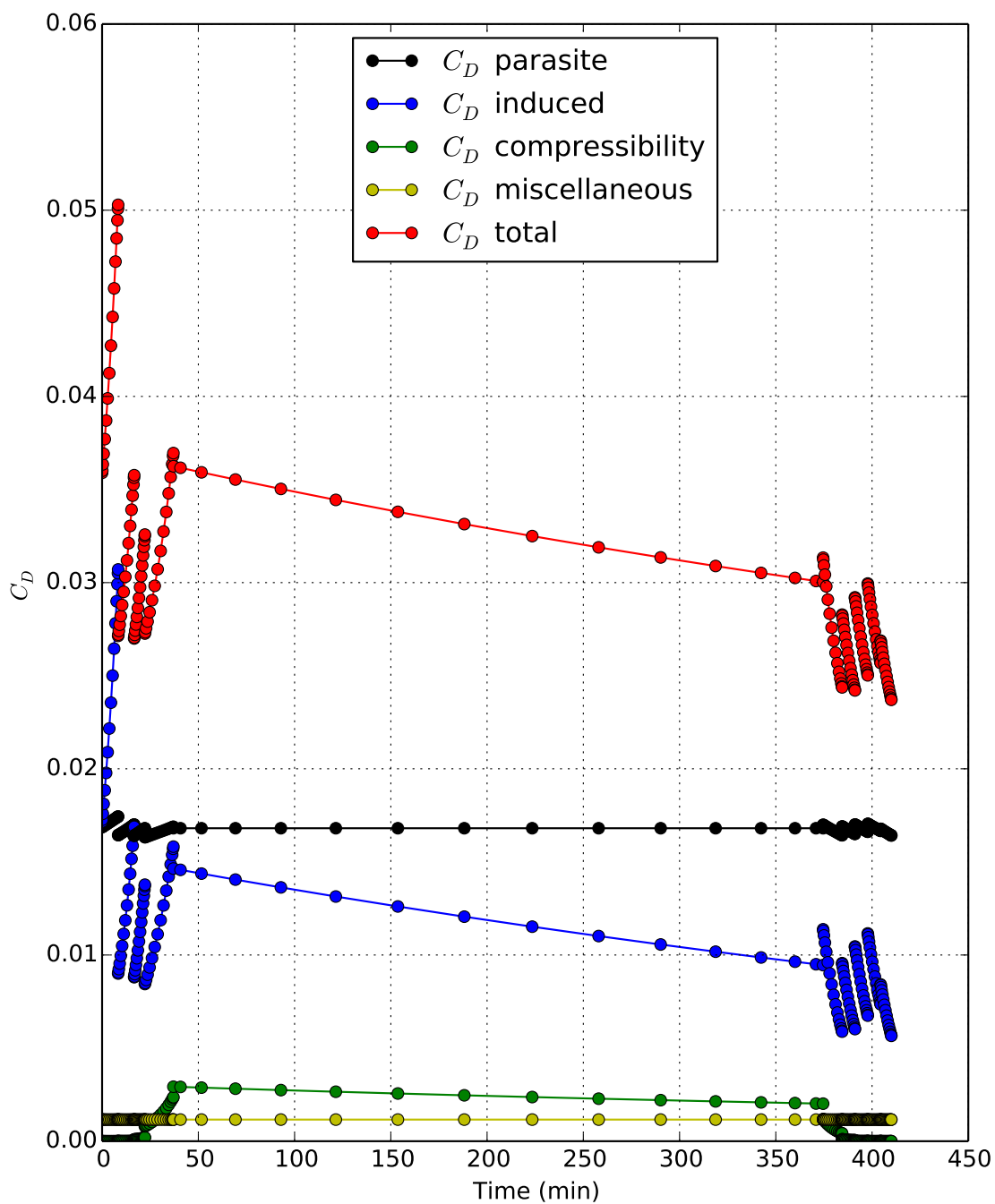
Figure 15: Force Breakdown for the B737-800

Figure 16: Force Breakdown for the B737-800

2. Regional Jet - Embraer E-190

For the evaluation of SUAVE methods applied to regional jet aircraft, an Embraer E190 type airplane, based in public data available[32, 33] was used.

All geometrical data was taken from 3-view drawings and the engine variables were based on public data of engine manufacturer.[34] All unknown variables were estimated based on similar airplane data.

Table 7 below presents a comparison of key parameters:

Table 7: EMB-190 Comparison Data

Parameter	E190 Public Data[32, 33]	SUAVE Result	Difference
Empty Weight [kg]	27837	27034	-823kg, -3%
Balanced Field Length [m]	1830	1784	-46m, -2.5%
Factored Landing Distance [m]	1244	1277	+33m, +2.6%

For the mission performance evaluation a payload-range comparison was made, presenting a very close agreement between the estimated and the public data[33] , as shown in Figure 17 below.
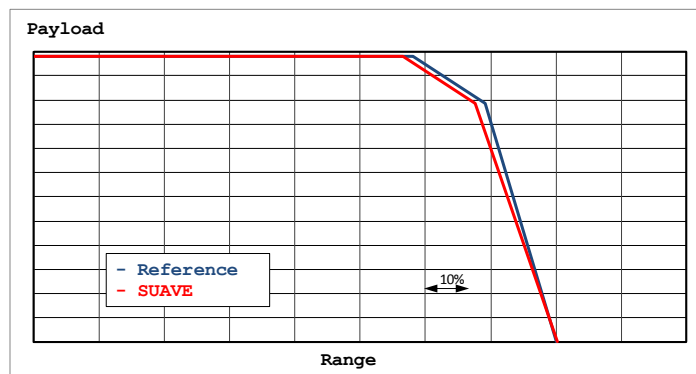

Figure 17: Validation of E190 Payload Range Diagrams

SUAVE results for a regional jet type airplane are within a $\pm 3\%$ error band, which can be considered more than adequate for conceptual design and optimization.

American Institute of Aeronautics and Astronautics

3. Supersonic Jet - Concorde

Since the supersonic modules must be validated separately, Concorde is used as an example aircraft. This demonstrates the supersonic zero-fidelity aerodynamics modules and a turbojet modified to be usable in supersonic conditions.

Concorde's mission includes a climb segment to 50,000 ft with linearly increasing Mach number, a cruise-climb to 60,000 ft at Mach 2.02, a supercruise segment at Mach 2.02, and a descent with linearly decreasing Mach number. The segments are broken up to cluster evaluation points near Mach 1.0. An increase in drag is observed here, but reliable data for Concorde at this point is not readily available. Aerodynamic values near Mach 1.0 are determined in this case by interpolating between lift and drag values at Mach numbers where data is more reliable. This is necessary because the typical methods used to determine lift and drag are generally not valid here, and several have singularities associated with linearized compressibility corrections. The precise methods used to avoid these issues are described in Section 3. The use of these coarse methods is justified with the relatively short period of the time that the aircraft will be operating in the regime very close to Mach 1.0

These results and Figure 18 are about what is expected for this level of fidelity. Many of the values used to create Concorde are also estimates due to limited available data. Supercruise takes values at the constant altitude section at Mach 2.02. High subsonic values are taken near Mach 0.93 as this is a typical Mach number used for subsonic cruise.[9] The drop in lift to drag ratio from subsonic to supersonic is expected as shown in Figure 18. This change is primarily due to the addition of wave drag. Which has been implemented with modules that only become active in the supersonic regime.

Table 8: Concorde Parameters

Variable	Value[35]
MTOW	185000 kg
Operating Empty	79000 kg
Wing Area	358 m^2
Wing Sweep	55°
Span	25.6 m

Table 9: Comparision of Performance Data for Concorde

Variable	Literature[36,37]	SUAVE
Range	3900 nmi	3800 nmi
Supercruise L/D	7.1	6.4
High Subsonic L/D	11.3	10.2

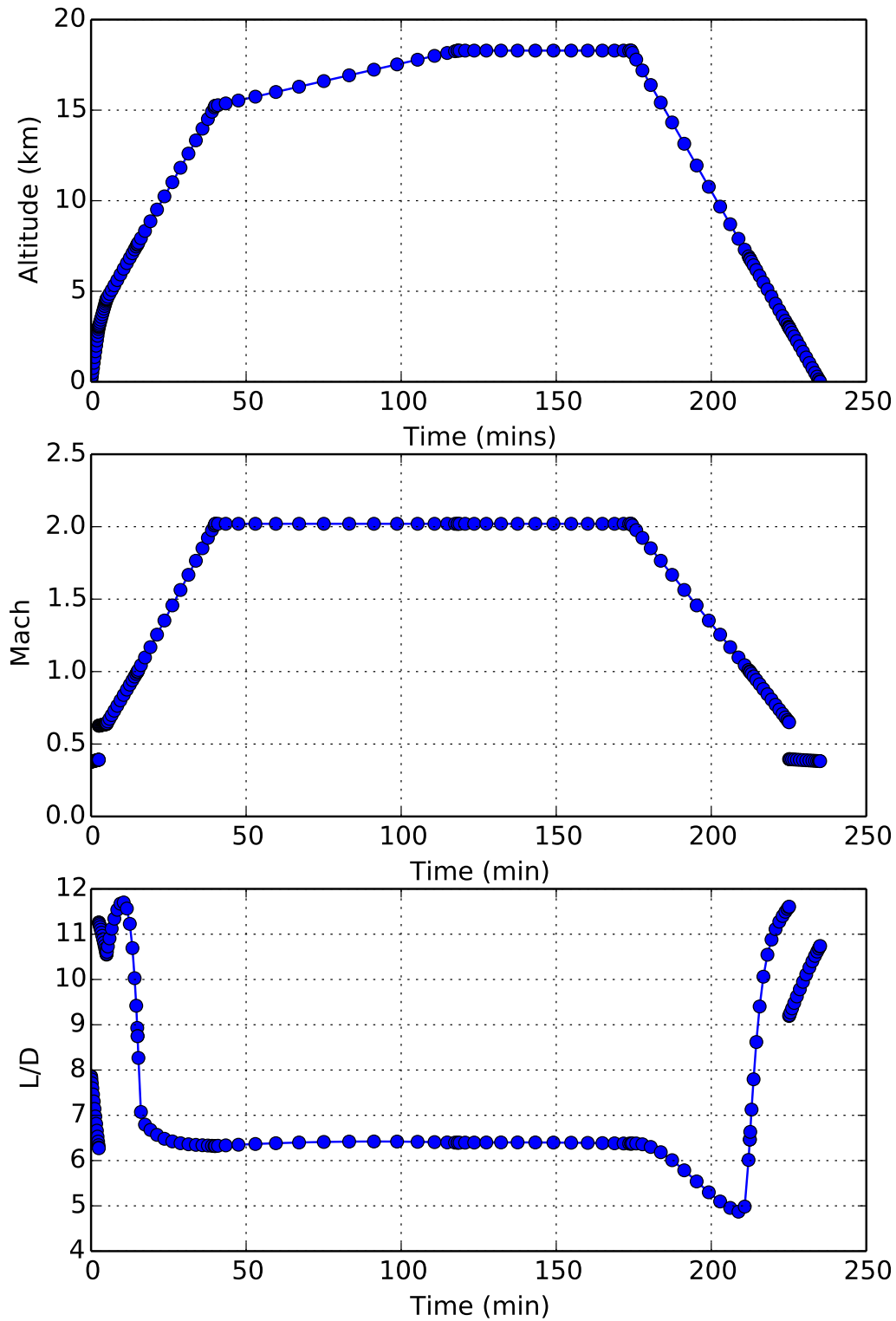American Institute of Aeronautics and Astronautics

Figure 18: Concorde Simulation Data

B. Unconventional Configurations

1. Blended Wing Body

Conceptual design of the blended wing body (BWB) is tricky given that prediction of its aerodynamics, stability, and structural weight cannot be done using the correlations and methods used for tube-and-wing aircraft. Physics-based relations are required to accurately model BWB aircraft. Using the multi-fidelity modular framework of SUAVE, conceptual design simulations have been run with improved models to try to estimate the performance of the BWB.

In the test case described below, SUAVE was used to analyze a BWB geometry representative of a concept developed as part of the NASA Subsonic Ultra Green Aircraft Research (SUGAR) study.[38] The aircraft was analyzed in SUAVE over a 2,100 nautical mile mission at a cruise altitude of 35,000 feet and a cruise Mach number of 0.74.

Table 10: SUGAR Ray Configuration and Mission Variables

Parameter	Value
Gross Wing Area	4,136 ft^2
Wingspan	161.4 ft
Gross Wing AR	6.865
Sweep Angle	27.7°
Takeoff Gross Weight	182,500 lb
Passengers	155
Cruise Altitude	35,000 ft
Cruise Mach	0.74

The mission was evaluated using two different aerodynamic models: first, the Fidelity-Zero vortex lattice method, which was also applied to the conventional cases; second, a database of three-dimensional simulation results collected using the SU2 CFD solver. A result from the CFD simulation is shown in Figure 19. Representative aerodynamic results in the cruise condition are shown in Table 11, below.

Table 11: Comparison of SUAVE aerodynamics results and published data on the SUGAR Ray[38]

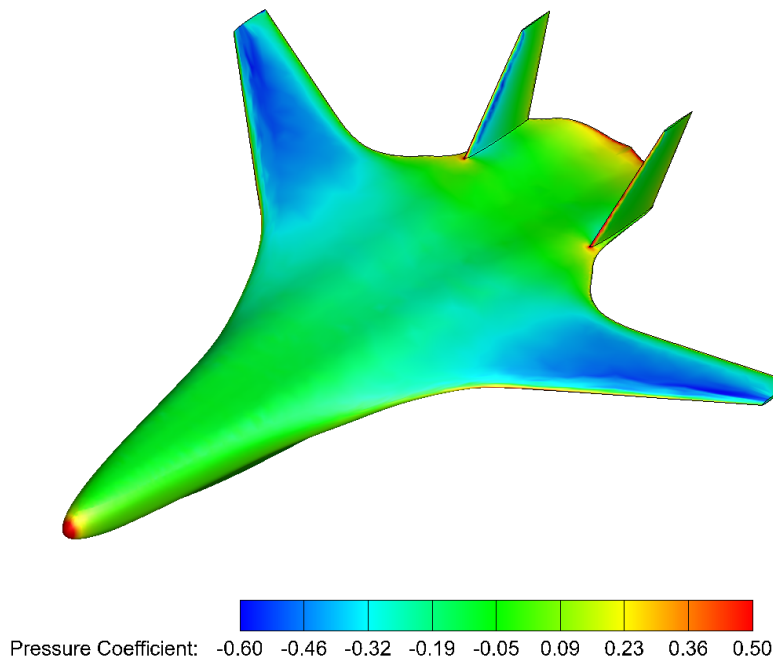| Result in cruise	Published	SUAVE
L/D - Lift-Drag Ratio	26.6	25.9
C_D - Total Drag	0.0113	0.0113
C_{D0} - Lift Independent Drag	0.0060	0.0055
C_{Di} - Induced Drag	0.0047	0.0051
C_{Dc} - Compressibility Drag	0.0006	0.0005
$C_{D,misc}$ - Miscellaneous Drag	0.0	0.0003

American Institute of Aeronautics and Astronautics

Figure 19: Pressure contour results from the CFD analysis used to generate a BWB aerodynamic model

In the Fidelity-Zero analysis, the geometry was modeled by a simple, tapered wing with a vertical tail (combining the areas of the SUGAR Ray's twin vertical tails). A dummy horizontal tail and dummy fuselage were added with zero wetted area. These dummy components made the geometry compatible with the Fidelity-Zero code, which is designed primarily for tube-and-wing aircraft. Fidelity-Zero yielded results reasonably matched to the available data on the SUGAR Ray BWB. The resulting drag results were close to the expected values, but with higher induced drag and lower compressibility drag relative to the total cruise drag. NASA's drag results involved no miscellaneous drag – a component of drag which SUAVE uses to account for control surface gaps, nacelle base drag, and fuselage upsweep. If the miscellaneous drag is included in the overall parasite drag, then the cruise parasite drag coefficient becomes 0.0058 – within 5% of the published value. Overall, SUAVE captures the lift-to-drag ratio (L/D) of the SUGAR Ray well, returning an L/D of 25.9, compared to 26.6 in the NASA report.[38]

The CFD approach used lift results from an inviscid Euler simulation run on a representative BWB at several angles of attack. These results were corrected for differences in freestream Mach number and used to create a surrogate model for lift. The CFD database does not currently contain accurate drag coefficient data, so drag was calculated using the same semi-empirical methods used for drag estimation in the Fidelity-Zero analysis. The higher fidelity CFD analysis yielded slightly lower angles of attack than were obtained from the Fidelity-Zero analysis. The Fidelity-Zero analysis reported a cruise angle of attack of between 1.8 and 2.0 degrees, while the CFD analysis returned a cruise angle of attack between 1.07 and 1.14 degrees. The CFD-based angle of attack results are expected to better represent the actual flight attitude of this BWB.

American Institute of Aeronautics and Astronautics

Table 12: Comparison of SUAVE weight prediction and published data on the SUGAR Ray[38]

Weight Component	Published [lb]	SUAVE [lb]
Operating Empty Weight	110,493	106,651
Zero Fuel Weight	146,918	141,526
Body	41,137	39,764
Tails	904	871
Outer Wing	12,500	12,432
Propulsion	15,918	15,885
Payload	36,425	36,119
Landing Gear	7,198	7,300
Flight Controls	6,015	2,743
Electrical Systems	3,346	2,009
Instruments and Avionics	4,304	1,696
Furnishings and Equipment	9,080	12,987
Pneumatics, Air Conditioning, and APU	3,553	6,086

Figure 20: SUAVE and NASA[38] estimates of the SUGAR Ray weight breakdown

The BWB weight estimation method outlined in Section 3 was used to estimate the weight of the SUGAR Ray aircraft. The SUAVE weight methods were effective for calculating structural weights of the centerbody, outer wing, tails, and landing gear of the SUGAR Ray. All these estimates were within 5% of the results set forth in the NASA report. However, the published weights for flight controls, electrical systems, and avionics were significantly higher than those found using SUAVE. Conversely, SUAVE found higher weights for furnishings and equipment, and for pneumatics, air conditioning, and the auxiliary power unit (APU). Because the NASA N+3 report does not include details of the methodology for finding system weights, the causes of discrepancy are not definitely known. Some expected reasons are as follows.

First, SUAVE and NASA may be handling the flight controls weight estimate differently. SUAVE's default approach bundles the weight of hydraulics with the pneumatics in the main wing, and includes only tail actuation systems in the flight controls. It is possible that NASA's flight controls result includes the control actuators in the outer wings, leading to higher reported flight controls weight and lower reported pneumatics weight.

Second, the SUGAR Ray is designed to make extensive use of electromechanical actuators. The high electrical system weight may be a result of this design choice.

Third, BWB aircraft like the SUGAR Ray typically have a larger number of independent control surfaces than a typical modern aircraft. This may further contribute to the higher weight of electrical systems and flight controls.

Finally, the SUGAR Ray is designed as a concept for potential use nearly 50 years in the future. As such, the weights of furnishings and equipment, air conditioning, and APU should be expected to be lower than SUAVE would calculate using data from modern aircraft.

All in all, SUAVE successfully estimated the structural weights of the SUGAR Ray blended wing body, despite its unconventional design; and SUAVE was able to find the operating empty weight to within 5% of existing results in literature.

American Institute of Aeronautics and Astronautics

2. Electric Regional Jet

SUAVE is capable of evaluating and optimizing a wide array of vehicle types, including ones with exotic energy-storage systems. One particular design of interest is an all-electric passenger jet using Lithium-air batteries as an energy storage system. This analysis assumes a specific energy of 2000 W-h/kg and specific power of 0.66 kW/kg. In comparison, Jet A has a specific energy of about 12,000 W-h/kg. Now, as Equation 96 showed, Lithium air batteries accumulate mass from the air as they discharge, and as a result, the aircraft is actually heavier in landing than in takeoff. As such, vehicle landing mass was chosen as the objective function for an optimization problem, where the mission trajectory, wing, and motor power were modified, and constraints (to ensure consistency in the mission, takeoff and landing field length, as well as feasibility of the propulsion system), were implemented via penalty functions. Motor mass was estimated from the state of the art air-cooled motor correlation shown in Figure 6. Other component weights were estimated based on traditional tube and wing sizing correlations as described in the Weights section. Table 13 shows all of the design variables for the vehicle while Table 14 depicts the flight profile design variables. A more detailed overview of the analysis and optimization can be found in previous work.[39]

Table 13: Aircraft Design Variables

Variable	Definition
α_{rc}	root chord twist
α_{tc}	tip chord twist
Λ	wing sweep
P_{motor}	motor power
S_{ref}	wing reference area

Table 14: Mission Design Variables

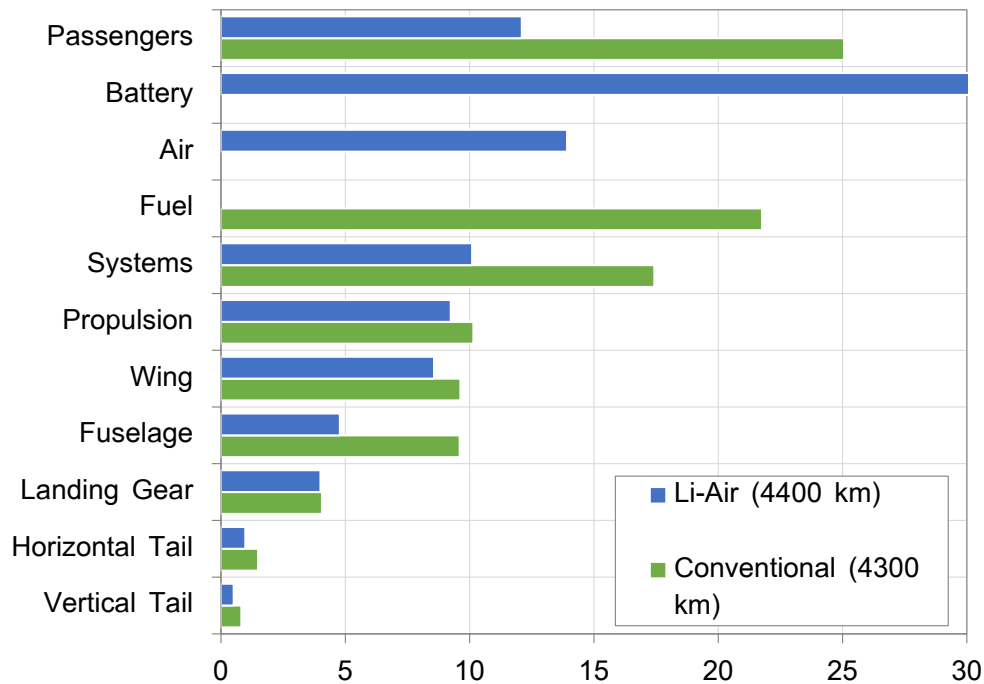| Variable	Definition
h_{climb_1}	final altitude of climb segment 1
h_{climb_2}	final altitude of climb segment 2
h_{climb_3}	final altitude of climb segment 3
h_{climb_4}	final altitude of climb segment 4
h_{climb_5}	final altitude of climb segment 5
V_{climb_1}	velocity magnitude of segment 1
V_{climb_2}	velocity magnitude of segment 2
V_{climb_3}	velocity magnitude of segment 3
V_{climb_4}	velocity magnitude of segment 4
V_{climb_5}	velocity magnitude of segment 5
$h_{descent_1}$	final altitude of descent segment 1
$h_{descent_2}$	final altitude of descent segment 2
cruise range	cruise segment range

American Institute of Aeronautics and Astronautics

Figure 21: Weight Breakdown Comparison

An overall mass breakdown can be seen in Figure 21, comparing a reference aircraft and an electric case. Note the relative magnitudes of battery and propulsion mass; the battery alone is about 36% of the overall mass of the vehicle, while the air is about 14% of the overall landing mass; the energy system mass therefore takes up about 50% of the vehicle mass vs. the conventional baseline design where fuel is only 22 % of the overall system mass. Additionally, to reduce power demands on the battery and motor, climb velocity was significantly reduced, while the vertical climb rate of the reference aircraft was maintained. Sample mission outputs from SUAVE for this design can be seen in Figure 22, with the electric aircraft model shown in Figure 23.
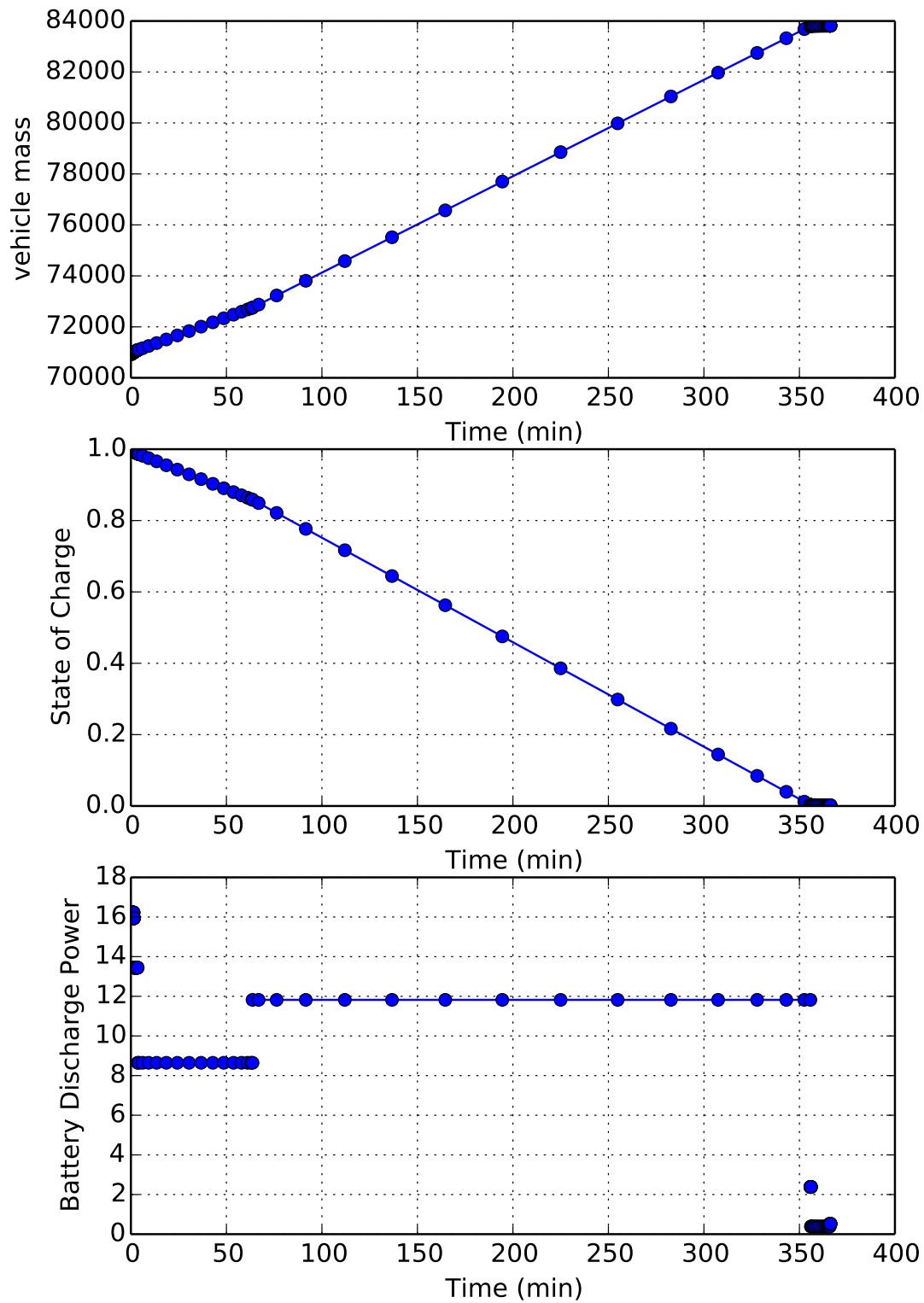
American Institute of Aeronautics and Astronautics

Figure 22: Sample Aircraft Mission Outputs
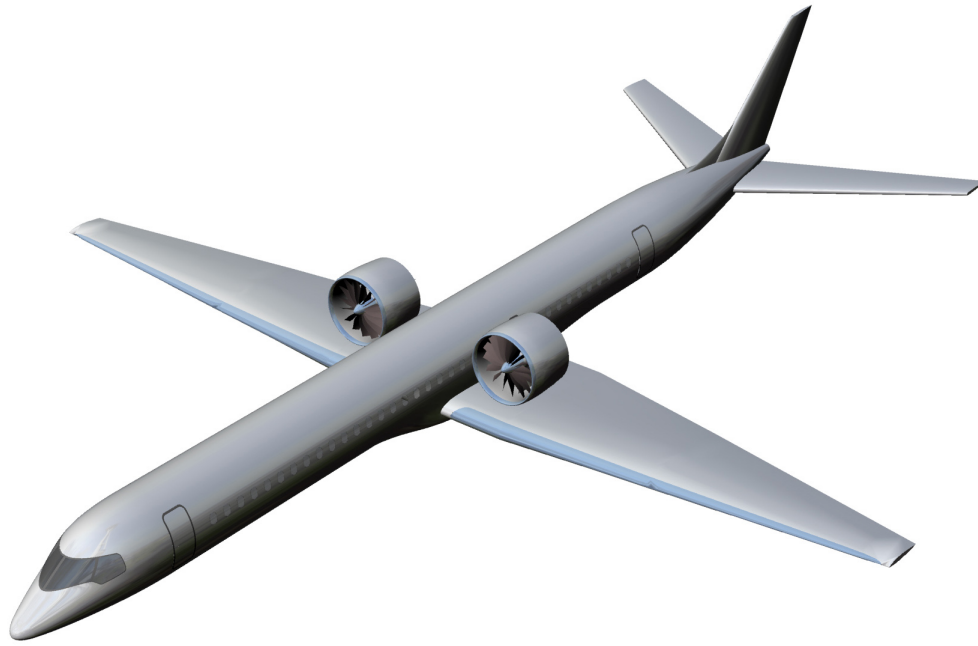
American Institute of Aeronautics and Astronautics

Figure 23: A regional jet designed to use Lithium-air batteries as a power source.

Additionally, with the large battery, storage space may become an issue. While the battery is heavier than the conventional jet's fuel weight, the battery volume (14.4 m^3) is actually smaller than the fuel volume of the reference aircraft (16.7 m^3). As the electric aircraft has a substantially larger wing than the reference aircraft, battery storage should not be an issue from the perspective of volume availability.

This analysis demonstrates SUAVE's versatility, in that it is capable of computing the performance of exotic propulsion systems, taking into account their various idiosyncrasies, such as the unusual mass change properties of metal-air batteries. Allowing for the integration of exotic subsystems within time-tested frameworks such as this enables a more rapid evaluation of these concepts, while maintaining a degree of credibility.

American Institute of Aeronautics and Astronautics

3. High Altitude Long Endurance Solar UAV

An example Solar UAV design was analyzed to demonstrate SUAVE's capabilities to analyze other exotic propulsion systems as well as other flight regimes. This example design is far from optimal, but it does shows that under the correct conditions, a truly unlimited endurance aircraft can be analyzed and designed within the SUAVE framework. This notional vehicle is similar in configuration to the original Qinetiq Zephyr as seen in Figure 24. However, the specifications of the vehicle analyzed do not represent the Zephyr.

Figure 24: Qinetiq Zephyr,[40] a motivational vehicle for this test case.

This notional design is far larger than the Zephyr at 40 meters in wing span and weighs in at 200 kg. However, it does carry double the payload of the Zephyr at 5 kg and accounts for constant payload power usage. The battery energy density is set to an optimistic estimate of 450 Watt-hours per kg. Additionally, 90 percent of the wings are covered with solar panels that have a 25 percent efficiency.

This mission exercises the methods developed for solar radiation estimation, propeller, and motor integration, as well as the human powered aircraft weight estimation. The mission setup is a constant altitude cruise at 15 km at Mach 0.12 for 24 hours. The location is over the California Bay Area during the summer solstice.

The analysis shows that the selected motor and propeller are both about 87 percent efficient during the flight. Further optimization analysis could be performed to increase the efficiency. The long day during the summer solstice allows the vehicle to collect solar power throughout a large portion of the day. The available power to charge the battery is very high during the day compared to the discharge rate at night. In fact, more power is created than can be stored. Within SUAVE, a constraint is placed that maximum battery capacities are respected. However, discharge capacities are not respected and batteries can be withdrawn less than their capacities to allow the designer to estimate the power deficit. Illustrations of these power concepts can be found in Figure 25.

The execution of this design case further demonstrates that SUAVE is capable of analyzing non-traditional vehicles for exotic missions.
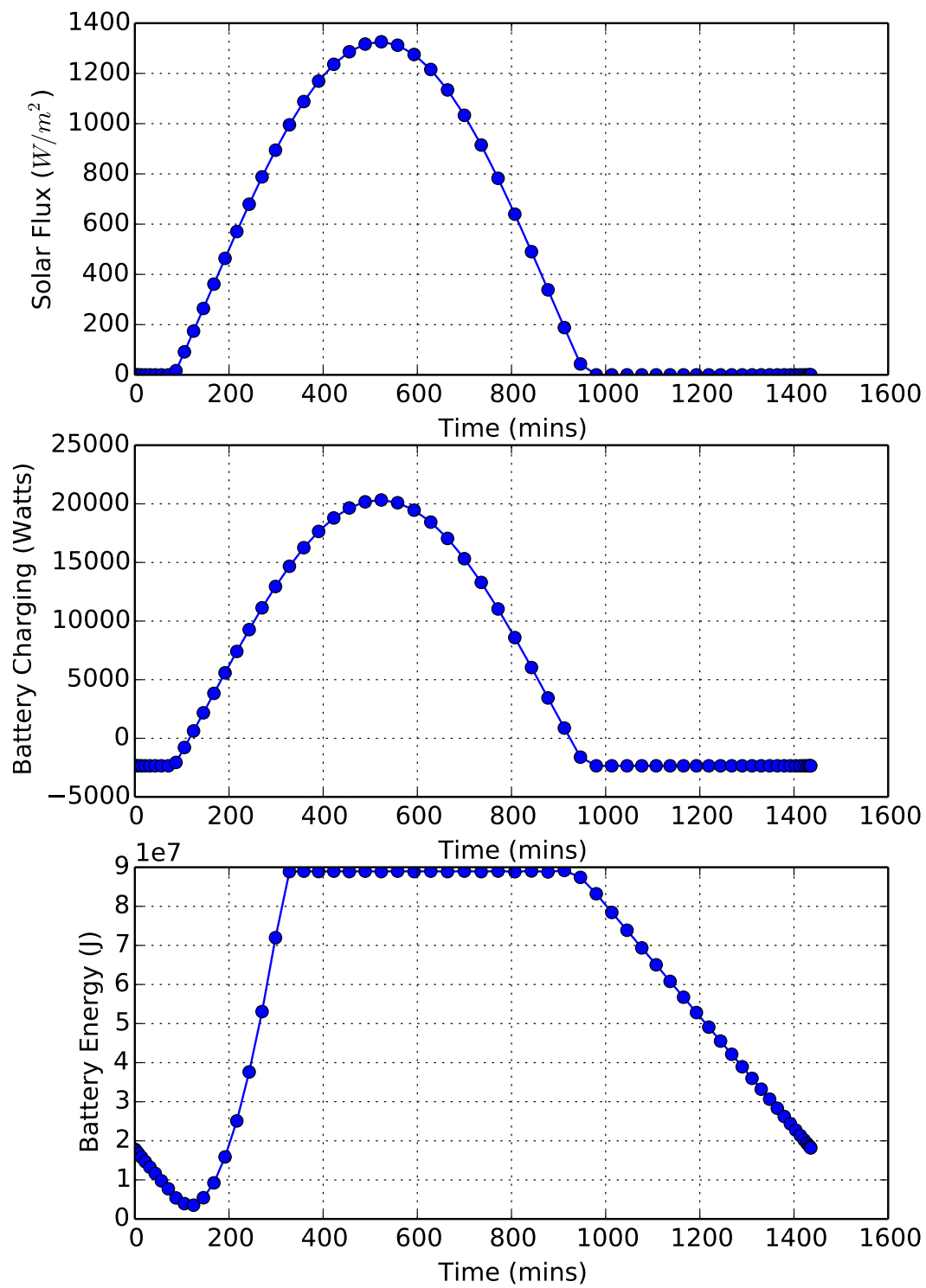
American Institute of Aeronautics and Astronautics

Figure 25: Solar Output Plots

American Institute of Aeronautics and Astronautics

V. Summary

In this work we have presented SUAVE, a conceptual level multi-fidelity design environment. It is developed from the ground up to be capable of credibly producing conceptual level design conclusions for future aircraft that incorporate advanced technologies. SUAVE achieves its goals through (a) an object-oriented architecture that enables arbitrary aircraft and propulsion system topologies, (b) an infrastructure that permits the incorporation of multi-fidelity analysis modules that can use the right level of fidelity at the right time, and (c) an energy-source-agnostic formulation that makes the integration of both traditional and carbon-free propulsion systems possible. The seamless integration of right-fidelity analyses will become increasingly useful as constantly improving computational resources allow for more physics to be included into the analysis during the conceptual design phase.

SUAVE contains a modular set of analysis tools that address a diverse array of vehicle types, both conventional and unconventional. Interfaces are standardized to enable a robust reliable combination of methods of any fidelity. Major analysis methods summarized in this paper include:

- Aerodynamics for subsonic and supersonic flight,

- Weight correlations for tube-and-wing, blended wing body, and human powered aircraft,

- Stability metrics for static and dynamic stability,

- Propulsion models for gas turbines and electric ducted fans,

- Energy networks for battery, fuel-cell, and solar based vehicles,

- Noise correlations for tube-and-wing aircraft based on the DC-10,

- Basic performance correlations such as takeoff and landing field lengths.

These modules are used in the evaluation of relevant equations of motion during a simulated mission that can be composed of arbitrary numbers of segments from a growing library of takeoff and landing, climb and descent, and cruise segments.

The software was exercised on several verification case studies. Three conventional vehicles (Boeing 737, Embraer E190, and Concorde) demonstrated results that compare well with publicly available data and existing aircraft design tools. Three unconventional vehicles (a blended wing body aircraft, an electric regional jet, and a solar unmanned aircraft) demonstrated results that integrate new analysis of unconventional energy systems and configurations.

With the software foundation in place and the basic functionality included in SUAVE, numerous areas exist to expand and improve SUAVE's capabilities for the aerospace community. Mentioned throughout this paper, SUAVE is built to work with optimization software. To that end our future work will center around completing and presenting optimization examples that use a input/output structure that easily interfaces with standard optimization packages, such as SciPy, PyOpt, OpenMDAO, and Dakota. This will enable a large degree of freedom when conducting design studies, making SUAVE useful for traditional gradient based optimization, as well as optimization under uncertainty and advanced multidisciplinary optimization decompositions.

In addition to incorporating SUAVE into existing optimization packages, our long term goals include several areas for work, and are available for collaboration through this open-sourced project. With continued development, SUAVE will include key multi-fidelity analysis modules that are useful for primary structural weight estimation, automated insertion of CFD data for flight performance, propulsion integration, and aeroelastic constraints. Part of this will involve developing and testing interfaces to simulation tools like SU2, and geometry representations like OpenVSP. To make SUAVE more suitable for efficient optimization, we will pursue an integrated gradient capability, that could rely on complex step, automatic differentiation, or analytic gradients. Additionally, we will continue to develop the energy networks to flexibly describe arbitrary topologies with radically new propulsion and energy sources. Through all development, SUAVE's functionality will be used to analyze unconventional aircraft designs and conduct new trade studies that explore new design spaces.

In the open-source, SUAVE is growing a unique connection to a global community of researchers and developers in the field of aircraft design. The release of the software under the Creative Commons NC-BY-SA 4.0 License has enabled engineers and scientists to work from a common code-base, providing worldwide

access to industry-standard analysis tools. Advances in aircraft conceptual analyses can be rapidly disseminated to a wide, knowledgeable user base in this online community. Contributions to SUAVE are actively sought and welcomed. Our goal is for SUAVE to be built by a community of experts from around the world.

Acknowledgments

References

[1]Stanford Aerospace Design Lab, "SUAVE: An Aerospace Vehicle Design Environment," `github.com/suavecode/SUAVE`, Version 0.2.0, `http://dx.doi.org/10.5281/zenodo.19734`, June 2015.

[2]Abelson, H. and Sussman, G. J., *Structure and interpretation of computer programs*, MIT Press, 1984.

[3]Szyperski, C., "Independently Extensible Systems - Software Engineering Potential and Challenges," *In Proceedings of the 19th Australasian Computer Science Conference*, 1996.

[4]Palacios, F., Alonso, J. J., Duraisamy, K., Colonno, M. R., Aranake, A. C., Campos, A., Copeland, S. R., Economon, T. D., Lonkar, A. K., Lukaczyk, T. W., and Taylor, T. W. R., "Stanford University Unstructured (SU2): An open source integrated computational environment for multiphysics simulation and design," *51st AIAA Aerospace Sciences Meeting and Exhibit*, Jan 2013.

[5]Grecco, H. E., "Pint: a Python units library," `http://pint.readthedocs.org/en/0.6/index.html`, Accessed: May 20, 2105.

[6]Owens, D. B., "Weissingers model of the nonlinear lifting-line method for aircraft design," *36th Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, December 1998.

[7]Kroo, I. and Shevell, R., "AA241A Course Notes: Aircraft Design, Synthesis and Analysis," `http://adg.stanford.edu/aa241/AircraftDesign.html`, Accessed: Nov. 11, 2014.

[8]Shevell, R. S., *Fundamentals of Flight*, Prentice-Hall, 2nd ed., 1989.

[9]Leney, D. and Macdonald, D., *Aerospatiale-bac Concorde: 1969 to 2003 (Owners' Workshop Manual)*, Haynes Publishing, 2010.

[10]Bradley, K. R., "A Sizing Methodology for the Conceptual Design of Blended-Wing-Body Transports," Tech. Rep. CR-2004-213016, NASA, 2004.

[11]Cruz, J. R., "Weight Analysis of the Daedalus Humanpowered Aircraft," *Proceedings of 21st OSTIV Conference*.

[12]Sinsay, J., Tracey, B., Alonso, J., Kontinos, D., Melton, J., and Grabbe, S., "Air Vehicle Design and Technology Considerations for an Electric VTOL Metro-Regional Public Transportation System," *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA, Indianapolis, IN, 2012.

[13]Roskam, J., *Airplane Design: Part VI, Preliminary Calculation of Aerodynamic, Thrust and Power Characteristics*, Design, Analysis and Research Corporation, Lawrence, Kansas, 2008.

[14]Gilruth, R. R. and White, M. D., "Analysis and Prediction of Longitudinal Stability of Airplanes," Tech. Rep. 711, NACA, 1941.

[15]Blakelock, J. H., *Automatic Control of Aircraft and Missiles*, John Wiley & Sons, Inc., 1991.

[16]Greitzer, E., "N+3 Aircraft Concept Designs and Trade Studies, FInal Report," Tech. rep., NASA/CR-2010-216794/VOL2, 2010.

[17]Cantwell, B. J., "AA210A Course Notes," `http://web.stanford.edu/~cantwell/AA210A_Course_Material`, Accessed: May 5, 2015.

[18]McDonald, R. A., "Electric Propulsion Modeling for Conceptual Aircraft Design," *53rd AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 2014.

[19]Larminie, J. and Dicks, A., *Fuel Cell Systems Explained*, Wiley Press, West Sussex, England, 2nd ed., 2003.

[20]Datta, A. and Johnson, W., "Requirements for a Hydrogen Powered All-Electric Manned Helicopter," *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM*, American Institute of Aeronautics and Astronautics, September 2012.

[21]Stückl, S., van Toor, J., and Lobentanzer, H., "Voltair: The All Electric Propulsion Concept Platform-a Vision for Atmospheric Friendly Flight," *28th International Congress of the Aeronautical Sciences*, ICAS, 2012.

[22]Duffie, J. A. and Beckman, W. A., *Solar Engineering of Thermal Processes*, Vol. 3, Wiley New York, 1980.

[23]Bower, G., Flanzer, T., and Kroo, I., "Conceptual Design of a Small UAV for Continuous Flight Over the Ocean," *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, American Institute of Aeronautics and Astronautics, 2011.

[24]Theodorsen, T., *Theory of Propellers*, Vol. 3, McGraw-hill New York, 1948.

[25]Adkins, C. N. and Liebeck, R. H., "Design of Optimum Propellers," *21st Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 1983.

[26]Powell, M. J. D., *A Hybrid Method for Nonlinear Equations, Numerical Methods for Nonlinear Algebraic Equations*, Gordon and Breach, 1970.

[27]L., K. L., "Subsonic Aircraft: Evolution and the Matching of size to Performance," Tech. Rep. NASA-RP-1060, L-13367, NASA, August 1980.

[28]Torenbeek, E., *Advanced Aircraft Design: Conceptual Design, Technology and Optimization of Subsonic Civil Airplanes*, Wiley Press, 2013, Equation 9.25.

[29]Obert, E., *Aerodynamic Design of Transport Airplane*, IOS Press, 2009, Figure 27.34.

[30]Askin, T. I., *Quasi-analytical modelling and optimization techniques for transport aircraft design*, Ph.D. thesis, KTH, 2002.

[31]"Boeing Airplane Characteristics for Airport Planning," `http://www.boeing.com/commercial/airports/plan\_manuals.html,`, December 2010.

[32]"Embraer E190 Weights," `http://www.embraercommercialaviation.com/AircraftPDF/E190_Weights.pdf`, June 2014, Accessed: Nov. 11, 2014.

[33]"Embraer E190 Performance," `http://www.embraercommercialaviation.com/AircraftPDF/E190_Performance.pdf`, June 2014, Accessed: Nov. 11, 2014.

[34]"GE Aviation CF34-10A White Paper," `http://www.geaviation.com/engines/docs/commercial/datasheet-CF34-10A.pdf`, February 2011, Accessed: Nov. 11, 2014.

[35]Jane's All the World's Aircraft, "Concorde," `https://janes.ihs.com/CustomPages/Janes/DisplayPage.aspx?DocType=Reference&ItemId=+++1336723&Pubabbrev=JAU_`, June 2005, Accessed: Nov. 12, 2014.

[36]"Concorde FAQ," `http://www.concordesst.com/faq.html`, Accessed: Nov. 12, 2014.

[37]Rech, J. and Leyman, C., *Case Study by Aerospatiale and British Aerospace on the Concorde*, AIAA Press, 2003.

[38]Bradley, M. K. and Droney, C. K., "Subsonic Ultra Green Aircraft Research: Phase I Final Report," Tech. Rep. CR-2011-216847, NASA, 2011.

[39]Vegh, J., Alonso, J., Orra, T., and Ilario da Silva, C., "Flight Path and Wing Optimization of Lithium-Air Battery Powered Passenger Aircraft," *53rd AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, Kissimee, FL, 2015.

[40]Defense Industry Daily, "NAVAIR Orders 7 Zephyr Ultra-Long Endurance UAVs," `http://www.defenseindustrydaily.com/NAVAIR-Orders-7-Zephyr-Ultra-Long-Endurance-UAVs-05426/`, Accessed: May 22, 2015.

American Institute of Aeronautics and Astronautics