menagerie: An Automated Approach to Launch Vehicle Design

Patrick D. Dees*
NASA Marshall Space Flight Center, Huntsville Al, 35808, USA

The Advanced Concepts Office (ACO) at NASA Marshall Space Flight Center (MSFC) has been performing conceptual design of launch vehicles since the Space Transportation System Program. As part of its duties ACO has been tasked with investigating the effects of known and potential threats and opportunities to a myriad of launch vehicle concepts, launch sites, and mission profiles known as the launch space. As the launch space becomes more crowded and busy with launch providers and their vehicles, the legacy method of prescribing each design discipline to a single human Subject Matter Expert (SME) has become infeasible. Rapidly changing launch, economical, and political constraints require rapid evaluation of their effects. While moving into this future of rapid response does require streamlining of data and processes, it does not however lead directly to reinventing the wheel. Flight-validated analysis tools, procedures, and lessons learned can be effectively leveraged to bring the experience from decades of government and commercial flight programs to the new paradigm. In this publication we describe the aggregation of all previous launch vehicle design efforts of ACO into an automated and integrated toolchain called *menagerie*. The toolchain utilizes a suite of NASA, Air Force Research Lab (AFRL), and ACO-developed tools and processes to evaluate launch vehicle concepts, mission profiles, and trades in a fraction of the time previously required.

I. Nomenclature

ACO = Advanced Concepts Office
$AFRL$ = Air Force Research Laboratory
CFD = Computational Fluid Dynamics
FEA = Finite Element Analysis
$GLOM$ = Gross Lift Off Mass
GRA = Ground Rules and Assumptions
$GRAM$ = Global Reference Atmosphere Model
$LaRC$ = Langley Research Center
$MDATCOM$ = Missile Data Compendium
$MSFC$ = Marshall Space Flight Center
$NASA$ = National Aeronautics and Space Administration
OML = Outer Mold Line
$POST$ = Program to Optimize Simulated Trajectories II
SAA = Space Act Agreement
SLS = Space Launch System
SME = Subject Matter Expert
SMF = Stage Mass Fraction
T/W = Thrust-to-Weight Ratio

II. Introduction

Conceptual design is a regime of rapid, critical changes. While the cost committed is low and design freedom is high it is imperative to investigate the impact of as many aspects of a design as possible. Previous studies have shown that up to 75% of the total Life Cycle Cost of a system is committed during Conceptual Design [1], resulting from downstream metrics such as reliability, safety, manufacturability, and operations cost being impacted by down-selection during this phase [2–4]. A typical downfall in this phase of design is that it is also the lowest point of design knowledge,

---

[*] Integrated Space Transportation Team Lead, NASA MSFC ED04 Advanced Concepts, AIAA Member

and so conceptual designers strive to generate as much information as possible about the design in this phase to best set a foundation for any program to follow.

The Advanced Concepts Office (ACO) at Marshall Space Flight Center (MSFC) is a part of providing the conceptual design of launch vehicles within National Aeronautics and Space Administration (NASA) for Pre-Phase A and Phase A concept studies. ACO operates as rapid turnaround analysis group that can cover broad design spaces of vehicles [5]. Several analysis tools used by *menagerie* have historically been a part of the ACO design process but were partitioned to different analysts within ACO [6]. The launch vehicle design process in this form required manual data transfer between analysts for a single design cycle. Multiple design cycles are typically recommended and warranted, with each transfer of data involving the risk of human error in transcription and communication. As we move beyond point designs and into sensitivity and trade space analysis the legacy procedure of ACO becomes infeasible. Modern trade studies embrace that the inter connectivity of complex systems require exploration of the multivariate space to effectively capture the integrated system response to a myriad of design factor changes [7–9]. These studies require large amounts of data to properly evaluate the space and resolve borders of feasibility.

Over the last several years ACO has combined a set of in-house, NASA, and Air Force Research Lab (AFRL) analysis tools to develop an integrated, automated, and generic launch vehicle sizing environment called *menagerie*. The *menagerie* toolchain streamlines the process of executing trade studies by removing the human requirement of manually moving data from one spot to another, coordinating with other analysts to ensure all necessary data has been communicated, error checking inputs and outputs, and integrating the data from all disciplines of design. Instead, our analysts are able to focus on the creation of highly flexible analysis templates which are generally capable, and in customizing them for novel technologies and concepts. The full design team can spend more time on defining the problem, trade space, and metrics to track as automation saves time running the numerous data points necessary to resolve interdependencies in the trade space. In tandem, the design team and customer can far more rapidly execute, analyze, and iterate to learn from each previous study and maximize the information gathered in a far shortened amount of time. Finally, the customer knows that the maximum throughput is being applied by handing the evaluation of the trade space to an automated system which will work day in, day out, week in, and weekend.

Moving to an automated approach does not immediately lead to leaving behind the vast stores of knowledge in our Subject Matter Expert (SME) analysts. These SMEs are centrally involved in the creation of template files for use in automation and exploration of analysis within their discipline to find more effective methods and expand our capabilities. The *menagerie* toolchain also employs heuristics derived from lessons learned across decades of experience in launch vehicle design and flight programs which cannot be sourced from anywhere else. As new lessons learned emerge in the quickly shifting launch, cis-lunar, and beyond operations we are continually updating and upgrading to position ourselves as a leading resource in complex launch and space architecture studies.

III. Tools

Analytical tools in use by the *menagerie* toolchain fall into two general categories: those developed by ACO and those developed elsewhere. In the case of ACO-External tools, Python wrappers have been developed to aid in integration with the other modules, handle common errors, are responsible for execution of the external tool, and ensure the output is translated to the appropriate format.

A. ACO-External

1. Global Reference Atmosphere Model (GRAM)

GRAM is a collection of empirical data and analytical models for the prediction of atmospheric conditions. It is developed and maintained by the Natural Environments Branch of NASA MSFC and the Atmospheric Flight and Entry Systems Branch of NASA Langley Research Center (LaRC). Since its first version in 1974 it has been continually updated and optimized and currently can deliver data on Earth, Mars, Venus, Jupiter, Titan, Neptune, and Uranus. Outputs of a GRAM execution include atmospheric conditions as a function of altitude such as pressure, density, temperature, speed of sound, and species concentrations. For Earth simulations there are also integrated datasets of recorded atmospheric data for a multitude of spacecraft launch sites. One of its unique features is the ability to inject spatial and temporal perturbations to the atmospheric and winds data, allowing for statistical investigation into best, worst, and expected performance of trans-atmospheric vehicles [10]. A current application of GRAM-generated data is to provide seasonal atmospheric predictions for use by ACO to predict performance variations of the Space Launch System (SLS).

2. Missile Data Compendium (MDATCOM)

MDATCOM is a conceptual design tool for estimating the aerodynamic coefficients of an axisymmetric body in a wide variety of configurations. It has been developed and maintained by the U.S. Army Aviation and Missile Research, Development, and Engineering Center (AMRDEC) at the Redstone Arsenal and the Aerospace Vehicles Technical Assessment and Simulation Branch of AFRL at Wright-Patterson Air Force Base. Since its inception in 1984 it has been used in a multitude of conceptual design studies. Inputs to MDATCOM are the outer mold line (OML) of the vehicle and the attitudes and atmospheric conditions at which the aerodynamic coefficients should be evaluated. Outputs of an MDATCOM run include coefficients of lift, drag, axial side and normal force, pitch yaw and roll moment, and derivatives with respect to angle of attack [11]. A current application of MDATCOM-generated data is to provide first-cut aerodynamic data of axisymmetric launch vehicles in the commercial space for ACO internal use and external Space Act Agreement (SAA) customers.

3. Program to Optimize Simulated Trajectories II (POST)

POST is an event-drive, point-mass trajectory simulation software with discrete parameter targeting and optimization capability developed by the Atmospheric Flight and Entry Systems Branch of NASA LaRC [12]. It has been continuously updated and in use across NASA and industry since its inception in the 1970s. In its current form it is capable of 3- and 6-Degree Of Freedom (DOF) analysis both in and out of atmospheric conditions and with branching trajectories for modeling Return To Landing Site (RTLS) and other reoverable stage activities. Inputs to POST are a vehicle definition according to their input file specifications, selections of atmospheric, gravity, heating and other models included in the POST source, settings for the optimizer to include indepenedent and dependent variables and an optimization target, and finally a sequence of events for the vehicle to perform. POST is currently in use by ACO [13] and LaRC for launch vehicle ascent modeling and optimization [14].

B. ACO-Developed

1. dyreqt

Dyreqt is a general capability developed by ACO for the modular build-up of multiple space elements partaking in a complex branching mission with optimization under constraints [15]. It has been in development for over a decade and in continual use by ACO and academia [16] for complex space architecture studies. In its current form each space element is defined as a collection of subelements representing propulsion systems, propellant tank(s), and other subsystems such as avionics, thermal protection, and secondary structures. Resource paths such as those between propulsive subelements and their associated tanks, power generation subelements such as solar panels, reactors, and batteries to computational subelements, mechanicsms and actuators, and heat generating subelements to heat disposal elements are supported. Boil-off of cryogenic subelements are supported as a function of their environmental conditions. Aggregation and separation of elements to and from branches composing a complex mission trajectory are supported. The tool is developed in Python utilizing the openMDAO [17] framework, with an API layer for constructing the design problem to evaluate. Inputs for dyreqt are Python dictionary definitions of the space elements involved in the architecture and a series of trajectory branches also defined as Python dictionaries, and where along the trajectory each space element first enters the architecture. Additionally, some setup of the openMDAO problem is also required to fine-tune optimization. Outputs of dyreqt are a multitude, typically those tracked for *menagerie* problems are the propellant loadings of each launch vehicle element stage and the gross mass of a vehicle stack.

2. zephyr

Zephyr is a wrapper for POST and a suite of tools for running trajectory trade studies which has been in development since 2015 in ACO. The POST wrapper portion of the tool takes in the location of a template input file and a Python dictionary of data for filling the template. The POST wrapper handles error checking the inputs against the template file, supplying the template file with default settings not provided in the inputs, running the POST executable, and parsing the output to detect errors. The parsed output is then used with a set of heuristics modeling the manual process by which a human trajectory analyst works on a trajectory problem to determine what next to do. Options are whether the vehicle and/or mission is infeasible, another run of the optimizer is advisable, or if some small tweaks to the inputs would result in a more optimal ascent [18]. The POST wrapper core is orchestrated by other portions of the tool which are aimed at efficient trade-space exploration by strategically spending more time on cases [19, 20] within the tradespace expected to have more leverage so that their converged information can be spread to their nearest neighbors [21]. Overall inputs to zephyr are a set of cases to optimize, the template trajectory file location, and output metrics to extract from optimized trajectories. Outputs are pulled from the time history of optimized trajectories, typical examples of which are ignition Thrust-to-Weight (T/W) and Gross Lift Off Mass (GLOM), payload delivered to a target orbit, maximum dynamic pressure and acceleration, and the times and altitudes of any separation events.

3. airheads

Airheads is a wrapper for GRAM developed in ACO starting in 2022. Its configuration file sets locations of the resources it requires, such as the particular GRAM executable and location of SPICE kernels. Its inputs are the location of the template input file to fill and a Python dictionary dictating the planetary body, launch and/or landing site, and date for atmospheric modeling. It handles error checking the inputs against the template file, supplying the template file with default settings not provided in the inputs, running the appropriate GRAM executable, and parsing the output to detect errors in operation. The outputs of GRAM are then converted into the tabular format used by POST for interpolation. and placed in the appropriate directory.

4. aerogen

Aerogen is a wrapper for MDATCOM developed in ACO starting in 2022. Its configuration file sets locations of the resources it requires, such as the default location of the MDATCOM executable. Its inputs are the location of the template input file to fill and a python dictionary dictating the vehicle OML and the attitude and atmospheric conditions to evaluate. It handles error checking the inputs against the template file, supplying the template file with default settings not provided in the inputs, copying the MDATCOM executable to the directory where the filled input file resides, executing MDATCOM, and parsing the output to detect erros in operation. The outputs of MDATCOM are then converted into the tabular format used by POST for interpolation, and placed into the appropriate directory.

IV. Toolchain

The *menagerie* toolchain is composed of a series of analytical tools and subroutines orchestrated by Python code. The process relies on several template files for the major tools which are used by automation to represent the different configurations under study. ACO has developed generic template files for single- and two-stage vehicle templates for all tools which cover a majority of cases. However, it is expected that each study will have some bespoke element which necessitates customization. The general steps for the environment are shown in Figure 1 below, with each step in the process detailed in the following subsections.

A. Ground Rules and Assumptions (GRA)

A first step in any study is to come to agreement on modeling methods, defaults, and appropriate levels of fidelity. In this phase there are multiple conversations between the ACO team and the customer, with ACO developing baseline models of the launch vehicle technology under consideration and iterating until the level of detail is satisfactory. The full set of default ACO GRA are also reviewed to ensure that there is no assumption or default being used by ACO the customer disagrees with. Additionally, expectations on data products and output metrics are set so that the study data can be used to answer as many questions as possible. To streamline this process, ACO retains and updates a document similar to this publication holding detailed descriptions of the tools in use by *menagerie*, the types of templates already available for the tools, defaults for each analysis, and optional model upgrades which are already available. ACO also retains and updates a GRA document tailored specifically to the study which over-rules any data in the base defaults document.

The GRA are then fed forward in each study in the form of default values, constraints, requirements, and sequences of events.

B. Atmosphere Modeling

The first step in each study is to generate any atmospheric data required. ACO employs the Global Reference Atmosphere Model (GRAM) which is capable of outputting a mixture of measured and calculated data for Earth, Mars, and other bodies in our solar system [10]. Inputs for this step are the planetary bodies involved in the study, date ranges for the mission(s) under consideration, and selected launch and/or landing sites. These inputs are relayed to airheads

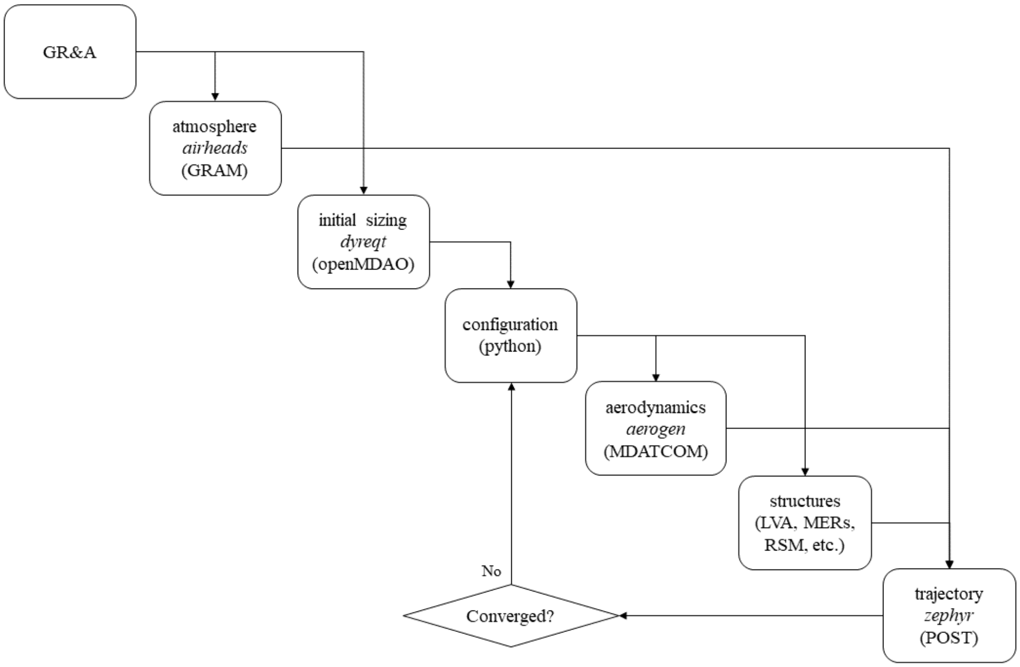


Fig. 1 N^2 **Diagram of the *menagerie* toolchain.**

which handles their usage, and results in atmospheric conditions such as pressure, temperature, density, speed of sound, and species concentrations. These outputs are then translated into the tabular format accepted by POST to be used as included data.

C. Initial Sizing

A first-pass approximation of the vehicle weights according to a candidate mission is performed by dyreqt. It takes in ranges of vehicle parameters and mission profiles from the GRA such as desired payload, number of stages, Stage Mass Fraction (SMF), Thrust and Specific Impulse of propulsion systems, propellant fluid selections, planetary body, launch site, target orbit, and others. A generic openMDAO problem is posed to optimize a delta-V (dV) split among the stages present according to their individual configuration and the dV required to perform the mission. Dyreqt is not currently capable of representing trans-atmospheric flight so an additional percentage is typically added to the calculated mission profile. Outputs of the generic dyreqt problem are the optimized dV and propellant loadings of each stage.

D. Configuration

In each design iteration the configuration of the vehicle is calculated. This module is typically unique to each study. It takes in data related to the propellant components of the vehicle such as propellant fluid(s) selection and their state (temperature, pressure) to size tanks, skirts, and intertank sections. Payload system geometry selection (Von-Karman, Tangent Ogive, etc.), payload amount, and minimum payload density are used to size a payload system with nosecone and cylindrical fairing. Engine size is assumed or calculated and used to size engine compartments, thrust structures, and interstage elements as appropriate. The final vehicle configuration is amalgamated from these parts according to the customer's specifications and can be customized for multiple general shapes (e.g. vertical vs. horizontal). The configuration module outputs the essential dimensions of the vehicle such as total length, maximum width, etc. as well as dimensions of individual components such as tank dome and barrel section height. In addition, the configuration data is then used to calculate the OML.

E. Aerodynamics

After calculation of the vehicle configuration the OML is used to estimate the aerodynamic properties of the vehicle. OML and atmospheric domain information is passed to aerogen which returns the data in the form of a file formatted for use by POST.

F. Structures

There are multiple avenues for evaluating the dry masses of the vehicle under consideration at different levels of fidelity [22]. A level-0 treatment defines a vehicle stage using SMF sizing. This is appropriate for studies on vehicles where only a subset of the subsystems which will ultimately comprise the vehicle have definition, such as applications of a new engine technology. A level-1 treatment uses the configuration definition along with material properties, maximum allowable load selection (max dynamic pressure, acceleration, etc.), and vehicle state at specific flight conditions (ignition, burnout, max dynamic pressure, etc.) to calculate required thicknesses and weights of primary structural components. A level-2 treatment uses the same inputs as level-1, however a full Finite Element Analysis (FEA) execution is typically too costly time-wise to be feasible in this workflow. In the case where this level of accuracy is desired, a sub-study varying portions of the vehicle configuration and vehicle state at various flight conditions is executed. The data on structural weights of the major components of the vehicle is then captured as a surrogate model [23] which allows utilization of the higher-fidelity analysis in-line with the rest of the toolchain. Outputs from this stage are the weights of the primary structures of the vehicle along with aggregate metrics such as gross weights of each stage.

G. Trajectory

The final step in a design iteration is attempting to find an optimized trajectory. The trajectory phase takes in most all the information used and/or generated by the previous phases. In the current configuration of *menagerie*, all the vehicle and mission combinations are submitted as cases to be evaluated as trajectories by zephyr. This step is typically the longest as many executions of POST are required to find optimized trajectories for each individual case. The time required for this step does reduce drastically after the first iteration however, as the optimized ascent information in terms of steering and payload delivered are stored with the case and utilized in the next iteration. Outputs for this step are typically optimized propellant loads for each stage, payload delivered, and vehicle state at specific flight conditions such as launch, maximum dynamic pressure and acceleration, staging, and at end of mission.

H. Convergence

At the end of each design iteration, each case is evaluated according to a convergence criterion to determine whether the design is within some tolerance. Since *menagerie* operates as a fixed-point iteration scheme, proper selection of the output assigned to the convergence criterion is important. Gross parameters such as GLOM are better suited since they do not tend to change by a large percentage after the first few iterations. Other parameters such as flight loads can vary by a large percentage with only a small change in the ascent trajectory. The criterion value to consider a case converged also widens as the iteration progress, as unstable designs can exhibit a multiple fixed point behavior which falls outside the tolerance. Once a vehicle has been marked as converged it does not participate in further design iterations and its final iteration's data is moved to a separate location for final analysis. If after twenty iterations a design still has not converged it is marked as complete.

I. Multi-Processing

Discussion on the *menagerie* tool up to this point has presented the steps as serial, however the actual implementation of the code utilizes multiprocessing to the extent of the compute resources' capability. The exception to this process is trajectory. The zephyr tool works best with all cases submitted simultaneously as its methods utilize nearest neighbor information to speed up optimization of cases. It has also been designed with parallelization in mind.

J. Data Products

There are several data products which are automatically produced by the *menagerie* toolchain. As design iterations progress, each case's data is saved to a dedicated directory. At the case level, a JavaScript Object Notation (JSON) formatted file contains the latest data pertaining to the case. Additional directories are present for iteration by iteration data from aerogen/MDATCOM such as the raw and POST-formatted aerodynamic data. At the study level, Comma

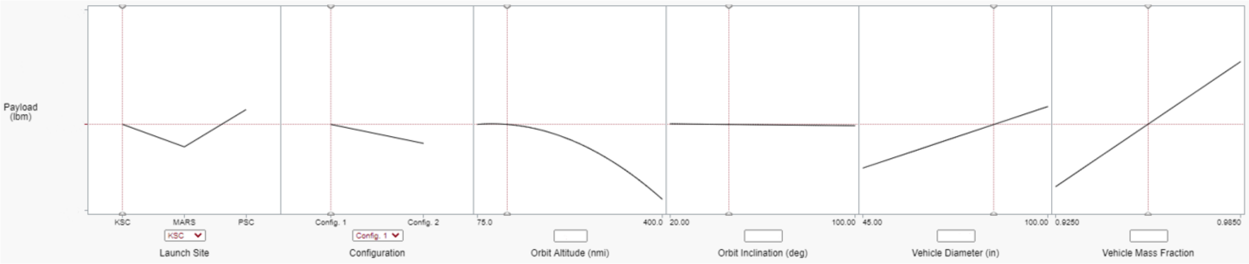


Fig. 2 Profiler of Vehicle Performance.

Separated Values (CSV) formatted data is saved iteration by iteration of the state of every case under consideration. This information is useful to understand convergence rates of the individual cases to detect unstable vehicle configurations. As cases reach convergence their final data is moved to a separate directory, and an additional CSV is kept of the final data. In addition, detailed logs at several levels of detail are saved as text files in the study data for the practitioner to debug when cases do not converge.

There are several additional data products which are not automatically generated, however are considered a standard set of outputs. First and foremost, a text document is compiled by the study lead detailing setup of the cases to include input factors, their ranges, the number of cases evaluated within those ranges, and any other defaults applied to the study. Pertinent results from the study are also presented and discussed in aggregate. These will tend to be explorations on the root causes of trends within the data, such as the effect of a maximum dynamic pressure constraint on payload delivered to a target orbit. Recommendations on specific vehicle designs which show robustness to mission profile variations are also presented when desired. The final data of the study is also used to create surrogate models for output metrics of interest such as payload delivered, GLOM, and ignition T/W. These surrogate models can then be displayed as a prediction profiler as shown in Figure 2. Each pane of the profiler displays a partial derivative of the integrated system response to each input factor and is a very useful tool for design space exploration and communication. ACO uses the program JMP [24] to perform surrogate modeling and create the profiler view. Customers without the JMP software can have the profiler delivered to them as an interactive HyperText Markup Language (HTML) document. Finally the surrogate model itself can be delivered to the customer as Python code for their own internal studies.

A semi-automated data product at the case level which is optionally available is a visual summary of the vehicle and its performance as an image colloquially known as a "baseball card". An example of this product is shown in Figure 3. The fields on either side of the vehicle depiction can be from any of the metrics collected by any step of the *menagerie* process. An additional ACO-developed tool named *bambino* takes in a definition of the text fields to surround the image and the location of target data of the concept to generate the image.

V. Future Work

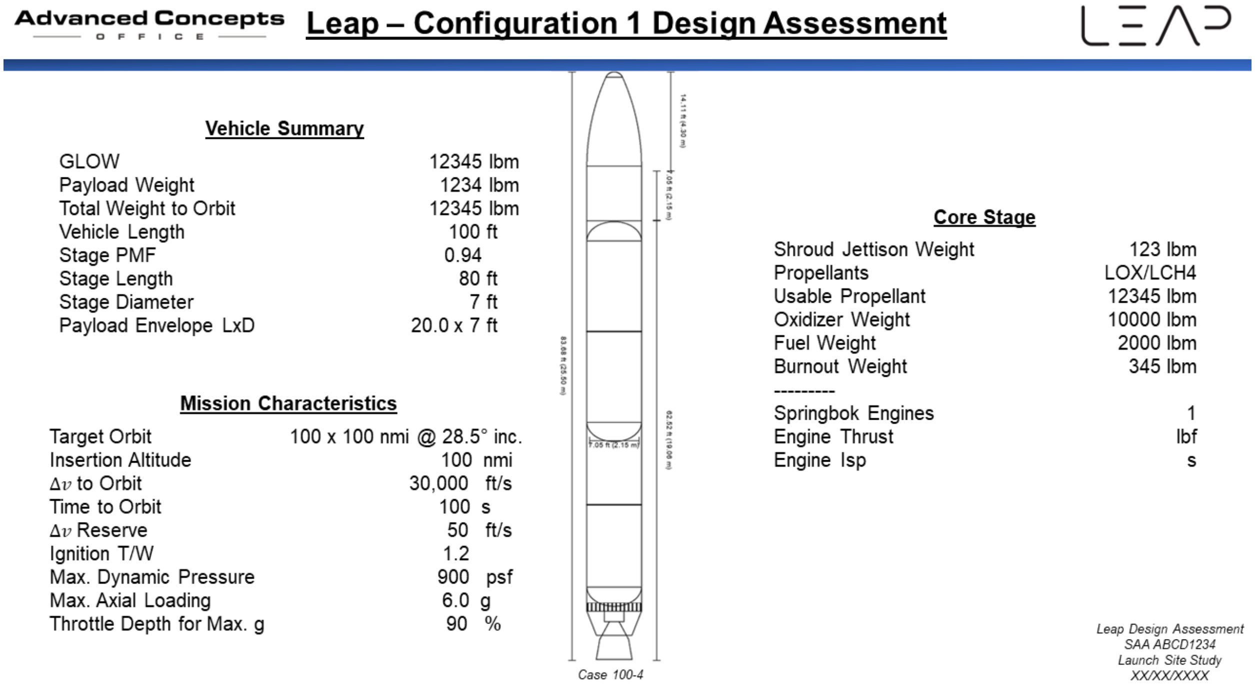
There are several research and development activities currently in work in ACO to further expand the capabilities of *menagerie*.

- openMDAO Refactoring

The current code structure of *menagerie* follows the functional paradigm. A next version of *menagerie* will refactor the modules into openMDAO Components to enable the capability of optimizing individual vehicle designs to targets. Examples of such would be maximizing performance robustness across a selection of mission profiles, minimizing the depth of throttle required of an engine to keep below stringent maximum acceleration constraints, and matching simulated performance to published values.

- Configuration Upgrades

The current configuration subroutine works only in two dimensions. Essential dimensions of lengths and widths are derived individually or from their neighbors with only rarely enforced alignment. Recent work undertaken in ACO in automated geometry generation and meshing will enable usage of tools which require a 3D data structure. Examples of this include higher-fidelity Computational Fluid Dynamics (CFD) tools which would provide an avenue for more detailed aerodynamic analysis of the vehicle concepts, and lift our current restriction of requiring axial symmetry.



Advanced Concepts OFFICE **Leap – Configuration 1 Design Assessment**

Vehicle Summary

GLOW	12345 lbm
Payload Weight	1234 lbm
Total Weight to Orbit	12345 lbm
Vehicle Length	100 ft
Stage PMF	0.94
Stage Length	80 ft
Stage Diameter	7 ft
Payload Envelope LxD	20.0 x 7 ft

Mission Characteristics

Target Orbit	100 x 100 nmi @ 28.5° inc.
Insertion Altitude	100 nmi
Δv to Orbit	30,000 ft/s
Time to Orbit	100 s
Δv Reserve	50 ft/s
Ignition T/W	1.2
Max. Dynamic Pressure	900 psf
Max. Axial Loading	6.0 g
Throttle Depth for Max. g	90 %

Core Stage

Shroud Jettison Weight	123 lbm
Propellants	LOX/LCH4
Usable Propellant	12345 lbm
Oxidizer Weight	10000 lbm
Fuel Weight	2000 lbm
Burnout Weight	345 lbm

Springbok Engines	1
Engine Thrust	lbf
Engine Isp	s

Leap Design Assessment
SAA ABCD1234
Launch Site Study
XX/XX/XXXX

Fig. 3 Baseball Card Summary of Vehicle Concept.

Another toolset enabled is FEA for determining the primary structural weights.

- Aerodynamics Upgrades

With automated mesh generation in hand the aerodynamics portion of the *menagerie* toolchain can be upgraded. Efforts are currently underway in ACO to compare CFD tools such as CBAero, Cart3D, and Fun3D. Comparisons between these tools [25] show strengths and weaknesses of each, which will be further explored as applied to complex launch vehicle configurations.

- Pre-FEA Structural Analysis

A current capability gap for *menagerie* is in a level-1 structural analysis tool appropriate for generic launch vehicles, from axially symmetric to not, small to heavy lift. A multi-year effort in collaboration with the Aerospace Systems Design Laboratory at Georgia Tech [26] is yielding first results and undergoing verification and validation iterations with several organizations within MSFC. When at a stable state this capability will provide structural weight estimates for full vehicle stacks and individual elements for either integrated or concentrated analysis.

- Trajectory Tool Updates

The developers of POST at LaRC recently completed a full refactoring of their code to ensure that it is completely thread-safe [27]. While this is a monumental achievement unto itself, it is a step on their path toward a full suite of upgrades to the trajectory tool. In particular, an upcoming release of POST will expose a Python API. The API will allow users of POST to interact with trajectories at a much deeper level, enabling new methods of optimizing trajectories, performing trade studies, and removing the need for most file i/o. When this version is released it will be on the ACO short list of capabilities to integrate.

VI. Conclusion

In the past each discipline's analysis and operation of all domain-specific tools were relegated (typically) to a single-point-of-failure SME. This process is best applied to evaluating a point design, or monovariate trades due to the human-hours involved. The *menagerie* toolchain represents a fundamental shift in the way ACO performs launch vehicle conceptual design studies. In this updated paradigm execution of the 100s to 1,000s of analyses required for a full trade study are handled by automation, freeing up SMEs for higher-level work. The SME is then free to focus on excellence in providing a flexible solution which can handle multiple design trades simultaneously, and in upgrading utilization of the domain tool to evaluate novel concepts.

Automation of trade studies has also allowed ACO to expand their pool of customers. To date, the *menagerie* toolchain has been utilized for several NASA-internal customers on multiple studies and several commercial partners through SAA contracts. Established vehicle programs such as SLS regularly pose future evolution and alternate utilization studies to ACO, which through usage of *menagerie* have become far more rapid and thorough. New entrants to the launch space, such as Leap Space have also executed studies through ACO via the SAA contracting vessel to investigate applicability of their engine design on a multitude of vehicle configurations and mission profiles. Finally, ACO-internal studies on the applicability of current, planned, or possible launch vehicles and technologies to complex architectures in cis-lunar space and beyond employ *menagerie* to develop general-use databases of parametric performance and prediction of launch states.

References

[1] Hirshorn, S. R., Voss, L. D., and Bromley, L. K., "Nasa systems engineering handbook," Tech. rep., 2017.

[2] Wertz, J. R., and Larson, W. J., *Reducing space mission cost*, Microcosm Press Torrance, CA, 1996.

[3] Blair, J., *Launch vehicle design process: characterization, technical integration, and lessons learned*, NASA, 2001.

[4] Ullah, R., Zhou, D.-Q., Zhou, P., Hussain, M., and Sohail, M. A., "An approach for space launch vehicle conceptual design and multi-attribute evaluation," *Aerospace science and technology*, Vol. 25, No. 1, 2013, pp. 65–74.

[5] Mulqueen, J., Maples, C. D., and Fabisinski III, L., "1.3. 2 Tailoring Systems Engineering Processes in a Conceptual Design Environment: a case study at NASA Marshall Spaceflight Center's ACO," *INCOSE International Symposium*, Vol. 22, Wiley Online Library, 2012, pp. 100–114.

[6] Waters, E., Garcia, J., Threet, G., and Philips, A., "NASA Advanced Concepts Office, Earth-to-Orbit Team Design Process and Tools," *AIAA SPACE 2013 Conference and Exposition*, 2013, p. 5372.

[7] Rechtin, E., *Systems architecting of organizations: Why eagles can't swim*, Routledge, 2017.

[8] Wertz, J. R., Everett, D. F., and Puschell, J. J., "Space mission engineering: the new SMAD," *(No Title)*, 2011.

[9] Larson, W. J., and Pranke, L. K., "Human spaceflight: mission analysis and design," *(No Title)*, 2000.

[10] White, P., and Hoffman, J., "Earth Global Reference Atmospheric Model (Earth-GRAM): User Guide," Tech. rep., 2021.

[11] Rosema, C., Doyle, J., and Blake, W. B., "Missile DATa COMpendium. User Manual – 2014 Version," Tech. Rep. AFRL-RQ-WP-TR-2014-0281, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, 45433-7541, 2014.

[12] Lugo, R. A., Shidner, J. D., Powell, R. W., Marsh, S. M., Hoffman, J. A., Litton, D. K., and Schmitt, T. L., "Launch vehicle ascent trajectory simulation using the Program to Optimize Simulated Trajectories II (POST2)," *AAS/AIAA Space Flight Mechanics Meeting*, 2017.

[13] Zwack, M., Dees, P., Thomas, H., Polsgrove, T., and Holt, J., "Program to Optimize Simulated Trajectories II (POST2) Surrogate Models for Mars Ascent Vehicle (MAV) Performance Assessment," Tech. rep., 2017.

[14] Lugo, R. A., Shidner, J. D., Powell, R. W., Marsh, S. M., Hoffman, J. A., Litton, D. K., and Schmitt, T. L., "Launch vehicle ascent trajectory simulation using the Program to Optimize Simulated Trajectories II (POST2)," *AAS/AIAA Space Flight Mechanics Meeting*, 2017.

[15] Edwards, S. J., Trent, D., Diaz, M. J., and Mavris, D. N., "A model-based framework for synthesis of space transportation architectures," *2018 AIAA SPACE and Astronautics Forum and Exposition*, 2018, p. 5133.

[16] Sudol, A., Edwards, S. J., and Mavris, D. N., "Technology Assessment for Launch Vehicle Concepts Using The DYnamic Rocket Equation Tool (DYREQT)," *2018 AIAA SPACE and Astronautics Forum and Exposition*, 2018, p. 5415.

[17] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, 2019, pp. 1075–1104. https://doi.org/10.1007/s00158-019-02211-z.

[18] Holt, J. B., Dees, P. D., and Diaz, M. J., "An expert system-driven method for parametric trajectory optimization during conceptual design," *AIAA SPACE 2015 Conference and Exposition*, 2015, p. 4486.

[19] Dees, P. D., and Zwack, M. R., "Automation of POST Cases via External Optimizer and "Artificial p2" Calculation," *AIAA SPACE and Astronautics Forum and Exposition*, 2017, p. 5353.

[20] Zwack, M. R., and Dees, P. D., "Improvement of automated POST case success rate using support vector machines," *AIAA SPACE and Astronautics Forum and Exposition*, 2017, p. 5129.

[21] Dees, P. D., Zwack, M. R., Edwards, S., and Steffens, M. J., "Augmenting conceptual design trajectory tradespace exporation with graph theory," *AIAA SPACE 2016*, 2016, p. 5650.

[22] Robinson, J., "An overview of NASA's integrated design and engineering analysis (IDEA) environment," *17th AIAA international space planes and hypersonic systems and technologies conference*, 2011, p. 2392.

[23] Myers, R., Montgomery, D., and Anderson-Cook, C., "Response Surface Methodology, John Wiley&Sons," *Inc., USA*, 2002.

[24] SAS Institute Inc., N., Cary, "JMP Version 17.0.0," http://jmp.com/, 1989–2023.

[25] Schwartz, H. D., Hiller, B. R., Robertson, B. E., and Mavris, D. N., "Comparison of Three Aerodynamic Analysis Software Packages Against the Army Navy Finner Projectile to Determine Fidelity Level," *2018 AIAA Atmospheric Flight Mechanics Conference*, 2018, p. 0292.

[26] Cox, A., Harris, L., Tang, J. Q., and Mavris, D. N., "Launch Vehicle Structural Analysis (LVSA): A Multifidelity, Multidisciplinary Environment for Design and Analysis of Rocket-Powered Vehicles," *AIAA SCITECH 2022 Forum*, 2022, p. 0806.

[27] Williams, R. A., Lugo, R. A., Marsh, S. M., Hoffman, J. A., Shidner, J. D., and Aguirre, J. T., "Enabling Thread Safety and Parallelism in the Program to Optimize Simulated Trajectories II," *AIAA SCITECH 2023 Forum*, 2023, p. 0148.