National Aeronautics and
Space Administration

Automated Preliminary Aircraft Sizing with HyperX

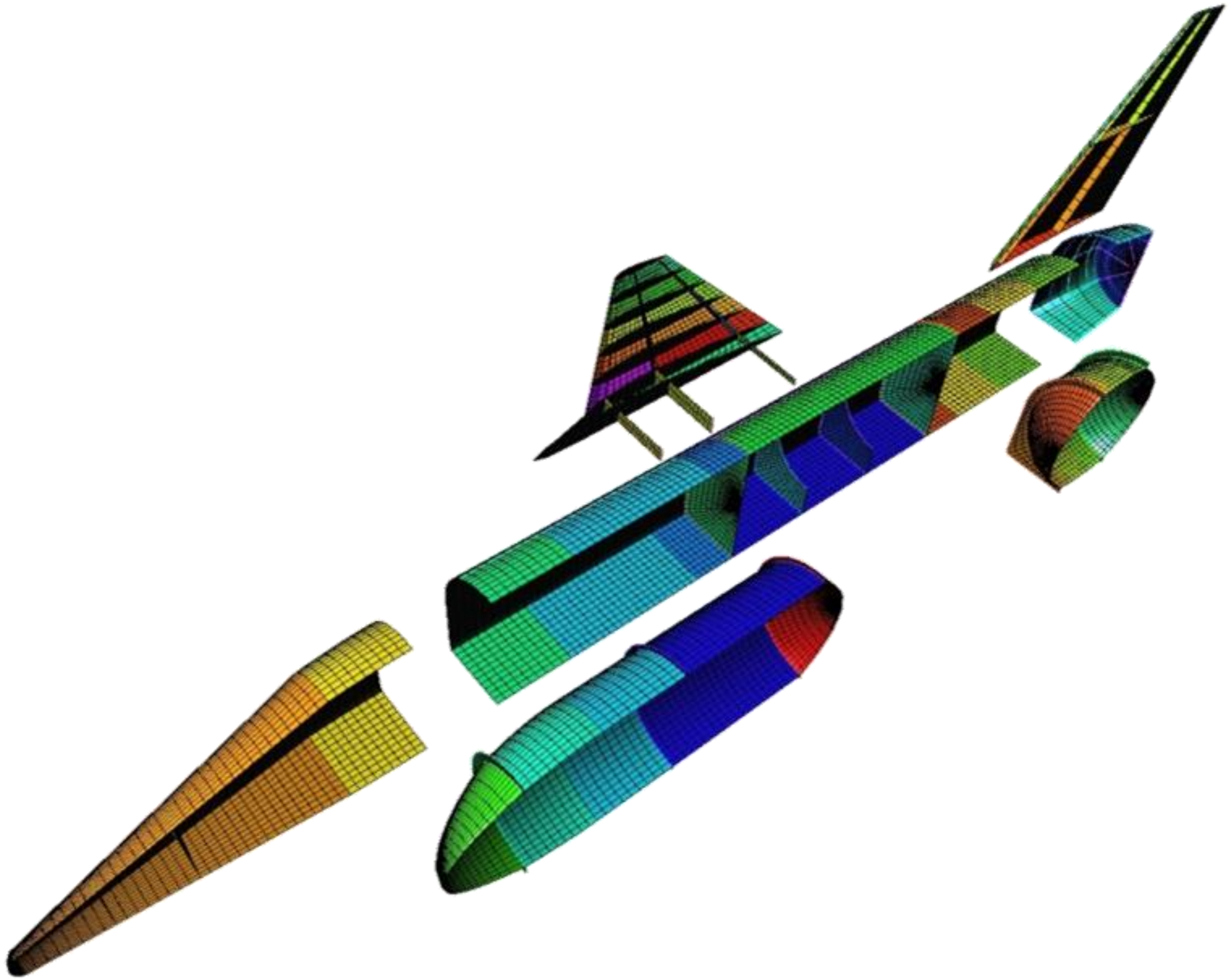
Lloyd B. Eldred, Ph.D.
Alexander M. Seifans

Vehicle Analysis Branch
NASA, Langley Research Center

SYSTEMS ANALYSIS AND CONCEPTS DIRECTORATE

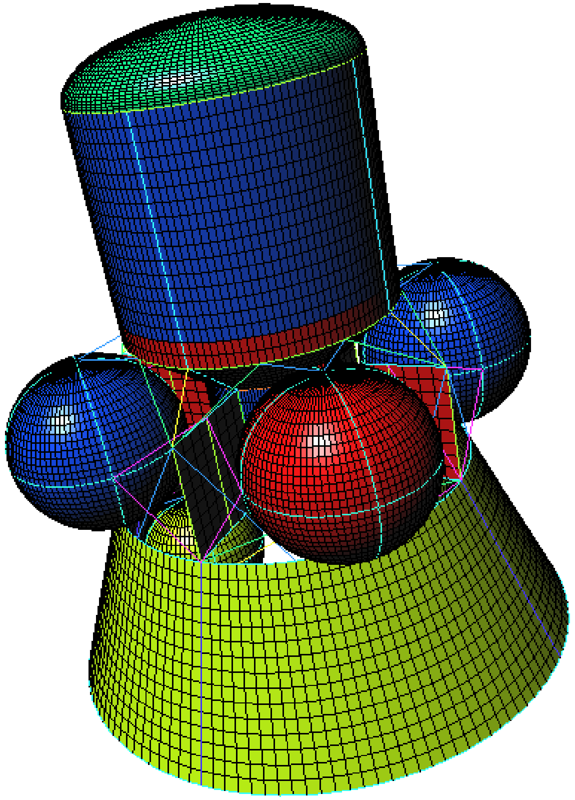
Outline

- Motivation
- SPARC overview
- HyperX Integration
- Results
- Conclusions



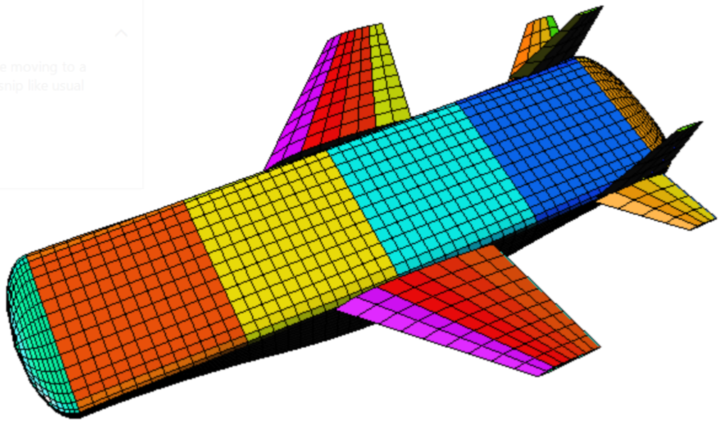
Motivation

- NASA Langley's Vehicle Analysis Branch is tasked with early conceptual design creation and maturing
- Provide informed structural design input at the earliest possible stage of vehicle concept development
  - Historically structural design has lagged during initial aero, propulsion, and trajectory studies.
- May have only broad system design parameters and requirements as input
- Existing legacy toolset for meshing, solving, and sizing
- Design aircraft, launch vehicles, landers, etc.
- Enable rapid exploration of the design spaces for components and full vehicles
- Support and refine design as system matures



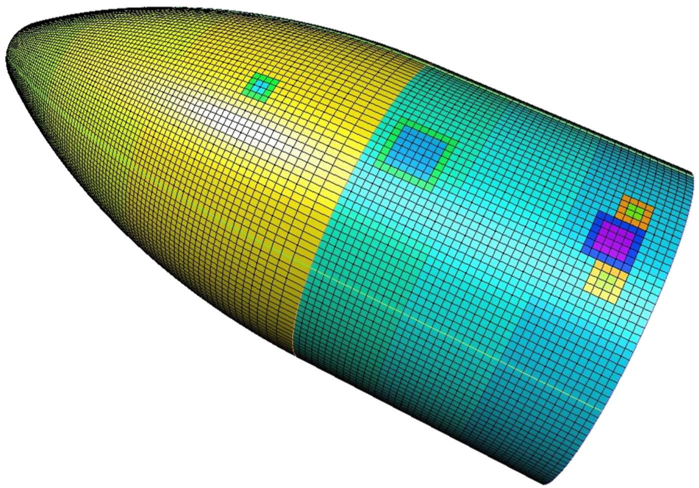
SPARC overview

- **S**tructural **P**reliminary **A**nalysis for ae**R**ospa**C**e vehicles
- User interface in JSON
- System coding in Python
- Structural & Load Modeling in *Loft*
- Load case creation in SPARC and scaling using ParseNAS
- Structural stress solution in NASTRAN
- Weight estimation and structural sizing in HyperX
- Is currently in active use for component design studies such as 50+ different wing designs for a high-speed vehicle



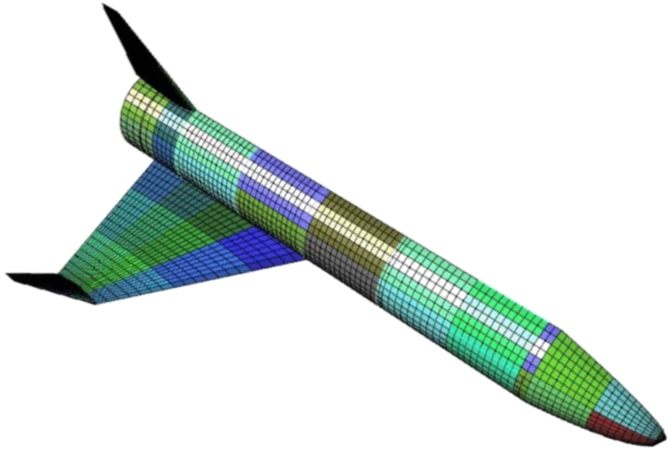
SPARC programmed in Python

- Common, free, modern coding language
- Huge variety of tool libraries available
  - JSON to Python dictionary
  - HyperX API driving
- Easily drive stand-alone tools by writing and reading text files
  - *Loft* input and output files are text
  - NASTRAN input (bdf) and output (f06) are text
  - ParseNAS reads bdf and f06 and writes text and csv



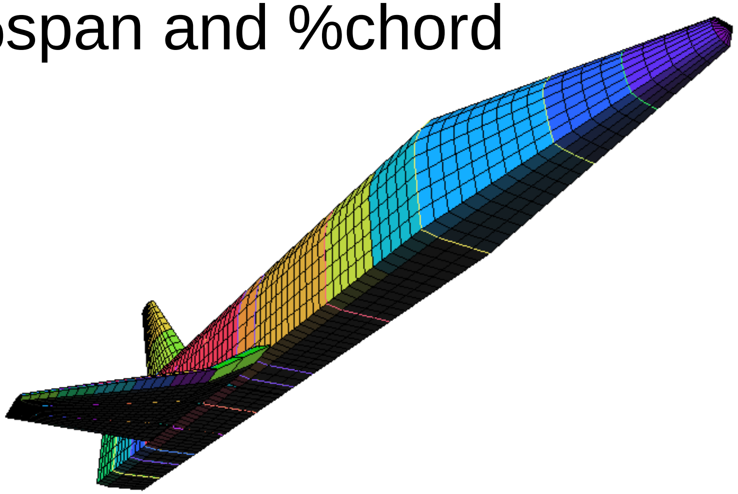
Loft structural modeling

- *Loft* is a general-purpose, parametric, conceptual level aerospace structural model creation tool
  - Free to download and use
  - AIAA SciTech 2024 presentation on “Using Loft”
  - Manual: https://ntrs.nasa.gov/citations/20230001772
  - Software: https://software.nasa.gov/software/LAR-18704-1
  - NASA TEAMS group for updated files, tutorials, and discussion



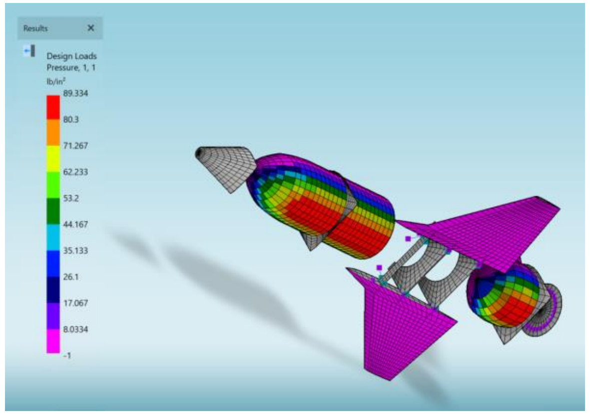
Loft use in SPARC

- Parametric templates created for models
  - SPARC can modify defined parameters
- *Loft* outputs commented NASTRAN mesh
- *Loft* is also used to create partial & surrogate models
  - OML and tank only meshes for load mapping
  - Tank meshes rotated to align with acceleration vector for hydrostatic loading
  - Wing meshes mapped to [0,1] square to compute %span and %chord



HyperX structural sizing

- SPARC uses HyperX to inform design decisions and explore a design space. HyperX is then directly used to refine the final SPARC created configuration.
- HyperX has an updated API that we are accessing via an in-house Python code called HXLoad.
- Buckling lengths automatically calculated by HyperX based on *Loft* property assignment
- Planned additions
  - HyperFEA iteration
  - Analysis documentation with HyperX dashboard

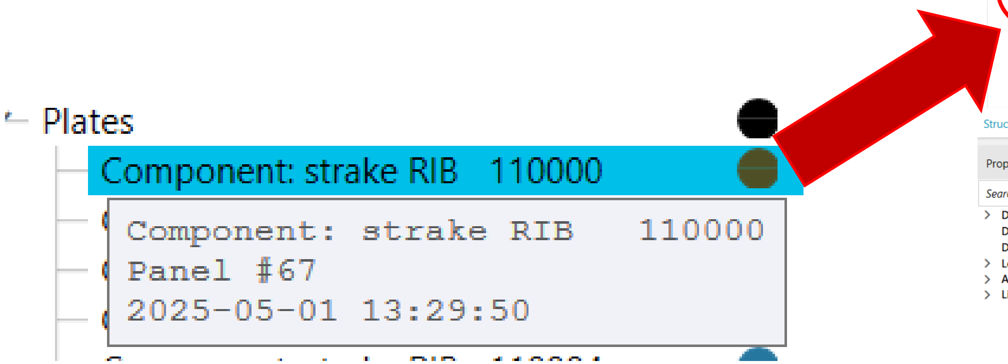


HyperX interfacing

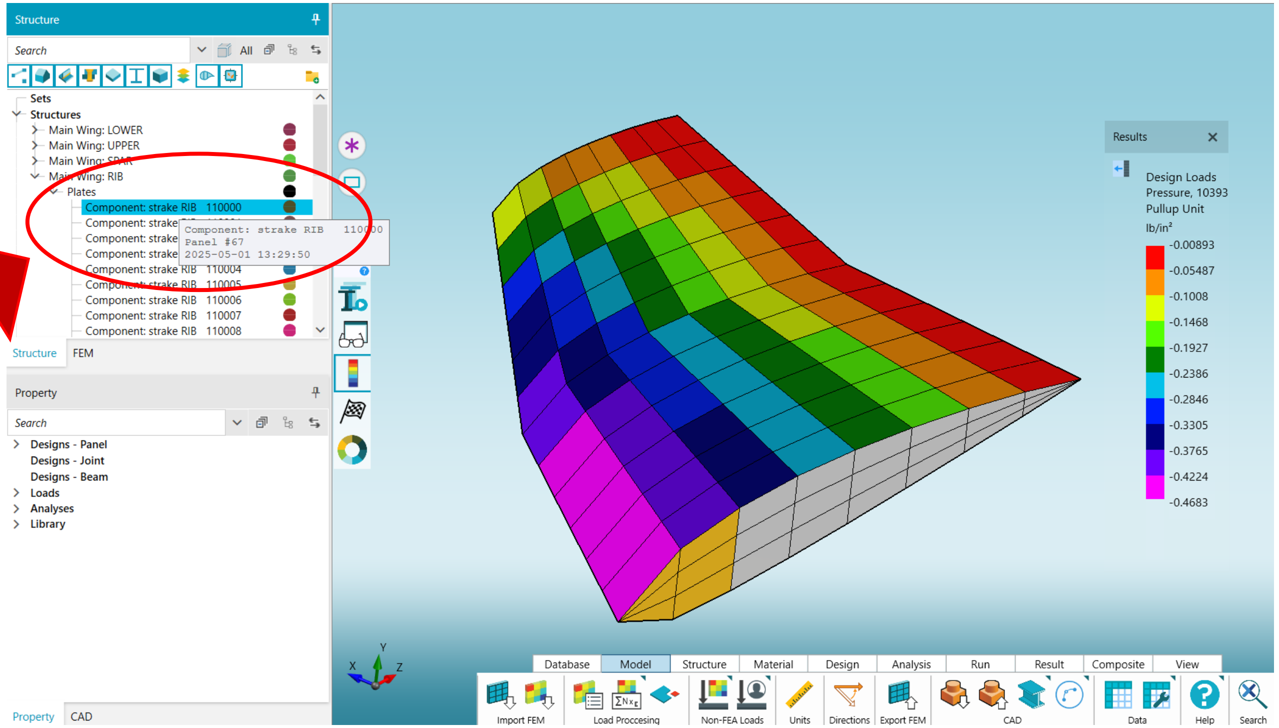
- *Loft* creates comments in NASTRAN file that enable group creation in HyperX
- HyperX analyzes the structure and can be opened to view the results mapped on the mesh

NASTRAN BDF Input lines from *Loft*

```
$HMCOMP ID                    110000  110000        4
$HMNAME PROP                  110000"strake RIB    110000" 4
$HMNAME COMP                  110000"strake RIB    110000"  110000 "strake RIB    110000" 4
$ Pset: "strake RIB" will be imported as: "pshell.110000"
PSHELL     110000  100000  0.0100  100000          100000
```

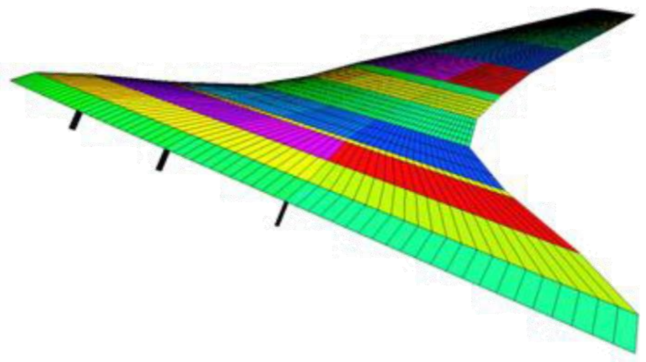


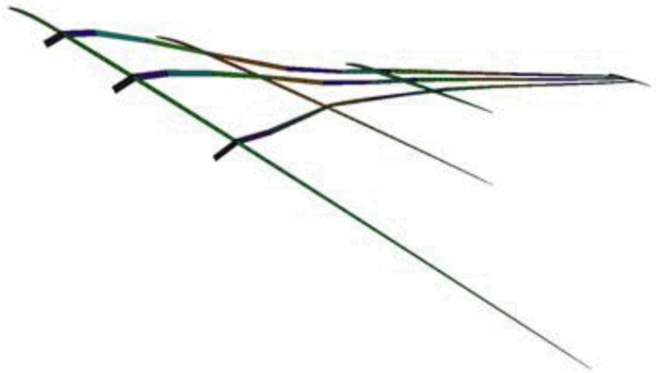
HyperX Analysis



HyperX templates

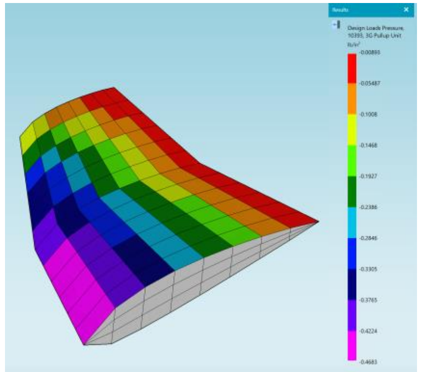
- SPARC uses a pre-defined user template to set material, panel design, load cases, failure criteria, etc. settings in HyperX
- Can have a simple metal plate for quick design-space exploration
- Can have a complex composite layup template for a more mature design
- Tune the analysis effort with design maturity



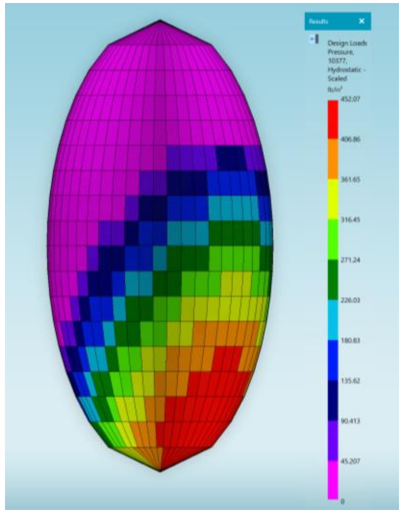


SPARC status

- System design & initial documentation complete
- HyperX API tool written and working with analysis definition, mass reporting, and margin checks
- Structural Analysis and Sizing Modules
  - Wing Module - Complete
  - Tank Module - Complete
  - Partial Vehicle Analysis – Complete
    - Vehicle loading implemented for multiple wing and tank surfaces within a full vehicle model.
  - Fuselage Module – In Development
    - Current Status: Able to load geometric information; Load application in progress
  - Full Vehicle Analysis – In Development
    - Awaiting completion of fuselage model



Pressure Distribution on a compound wing



Pressure Distribution on an oval tank with a 45 degree acceleration vector

SPARC load case generation

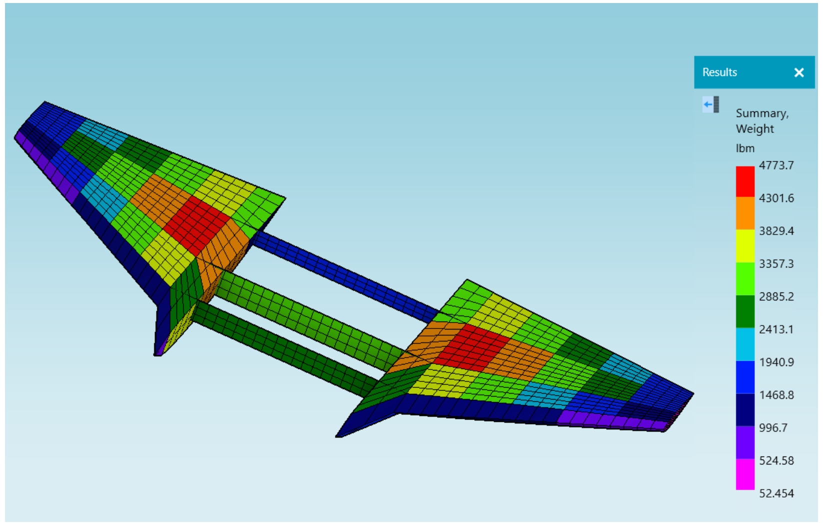
- User selects from available typical load set types and specifies loading parameters

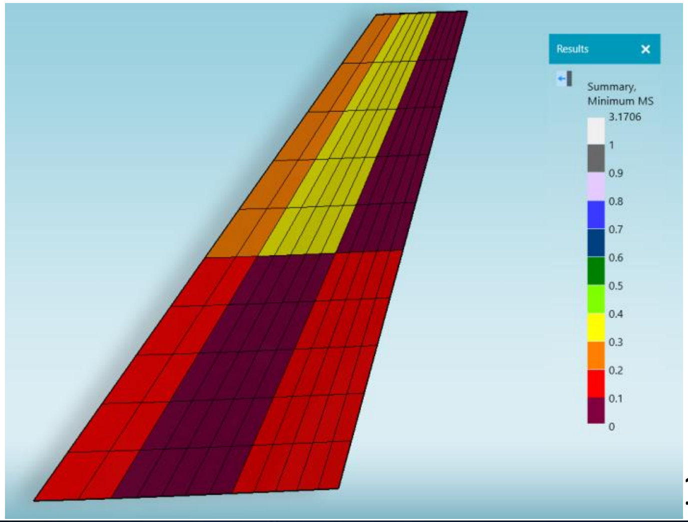
Component	Design Load	Input Variables
Wing	Max Maneuver	vehicle mass, max maneuver accel
Vert Tail	Engine Out	single engine thrust, engine/tail offsets (moment arms)
Tank	Max Hydrostatic Load	Ullage, fuel density, mass, accel vector
Fuselage	Runway bump	Major non-structural masses (fuel, payload, engines)

- *Loft* generates list of nodes and elements to apply loads to
  - Upper/lower skin element lists for lift
  - All elements above/below a specified z-coordinate for hydrostatic loading
    - (An initial rotation is applied to the model to align with the acceleration vector)
- SPARC creates NASTRAN load and case control cards for each load case

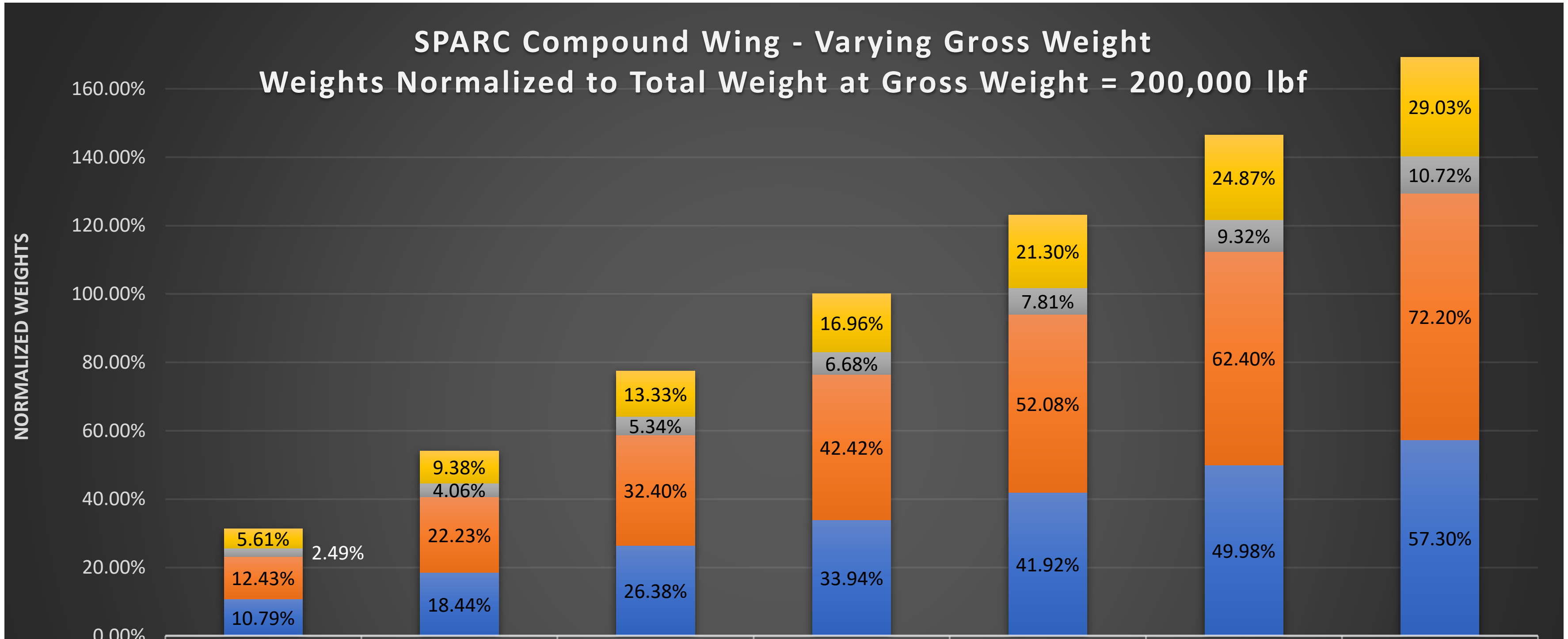
SPARC module - wing

- The user selects from available typical load set types
- *Loft* generates files for the upper skin elements, lower skin elements, and root spar nodes
- The Wing Module applies constant pressure or elliptical spanwise/triangular chordwise load cases and then performs load scaling and internal load calculation
- SPARC can also modify geometric design parameters for the wing directly

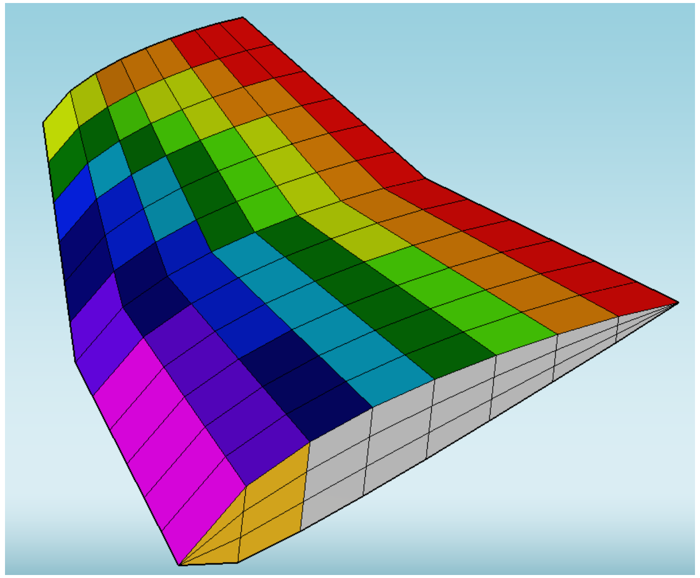




SPARC module - wing

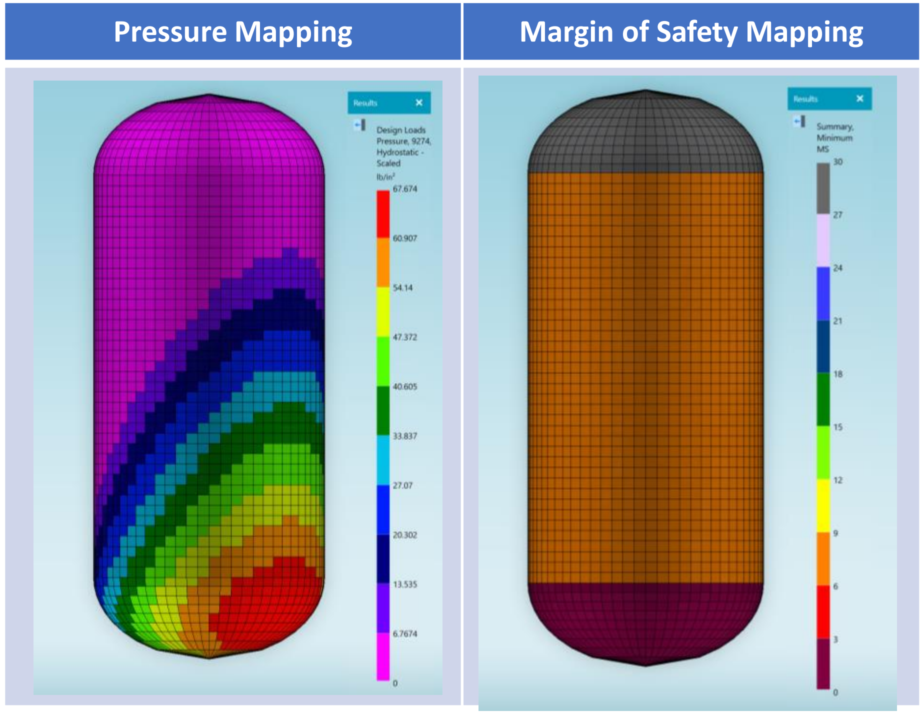


	GW = 50000	GW = 100000	GW = 150000	GW = 200000	GW = 250000	GW = 300000	GW = 350000
Total	31.32%	54.11%	77.46%	100.00%	123.11%	146.56%	169.24%
Lower Skin	5.61%	9.38%	13.33%	16.96%	21.30%	24.87%	29.03%
Upper Skin	2.49%	4.06%	5.34%	6.68%	7.81%	9.32%	10.72%
Spar	12.43%	22.23%	32.40%	42.42%	52.08%	62.40%	72.20%
Ribs	10.79%	18.44%	26.38%	33.94%	41.92%	49.98%	57.30%



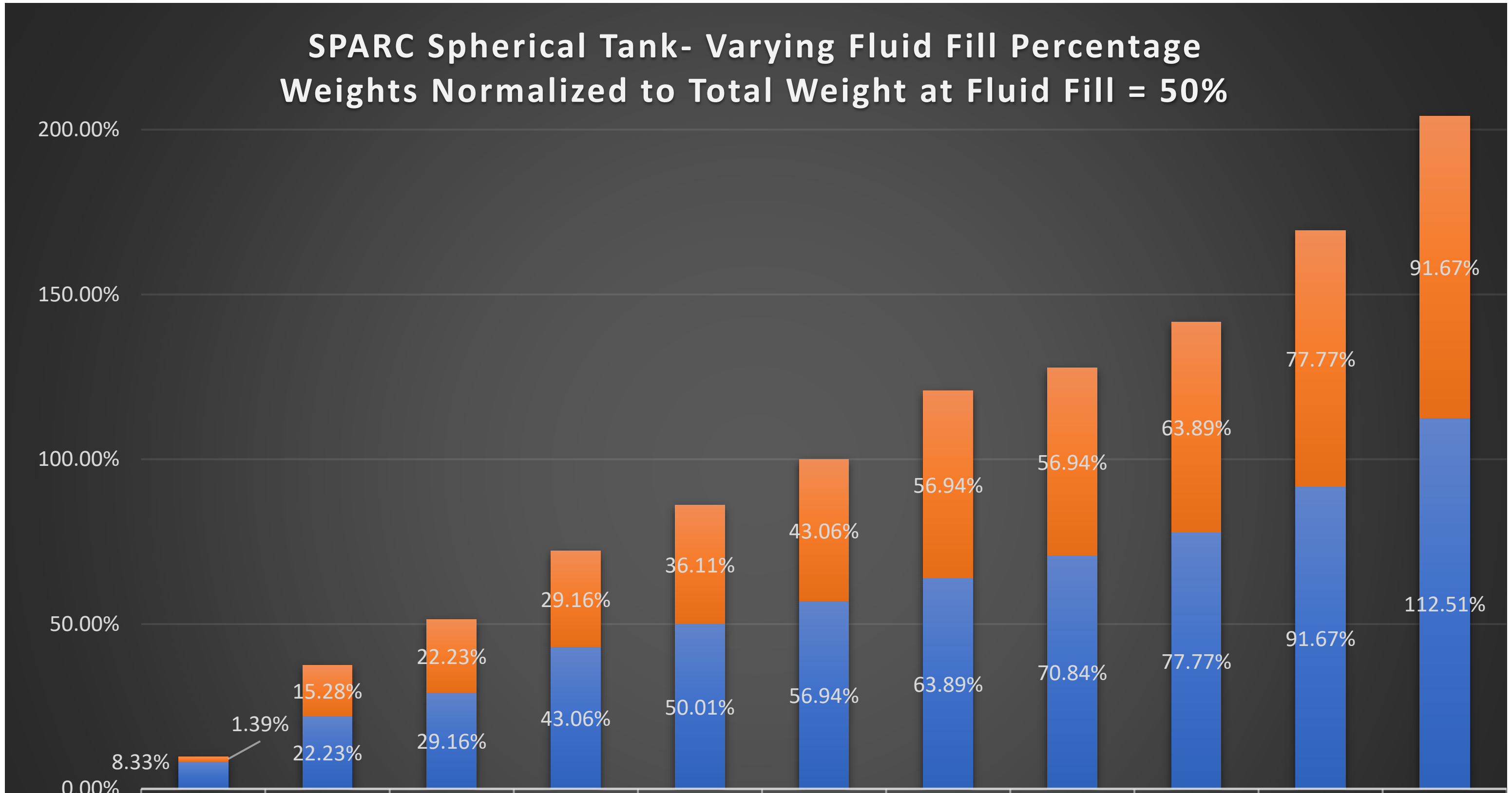
SPARC module - tank

- *Loft* generates a surrogate tank model that is rotated to align with the acceleration vector so that the fluid depth of each element can be easily computed
- The Tank Module applies ullage and/or hydrostatic load cases and then performs load scaling and internal load calculation
- The boundary conditions are chosen at the max locations along the axes
- SPARC can iterate on free surface location or adjust load scale to produce exact specified load.
- SPARC can directly modify Loft geometric parameters for design studies

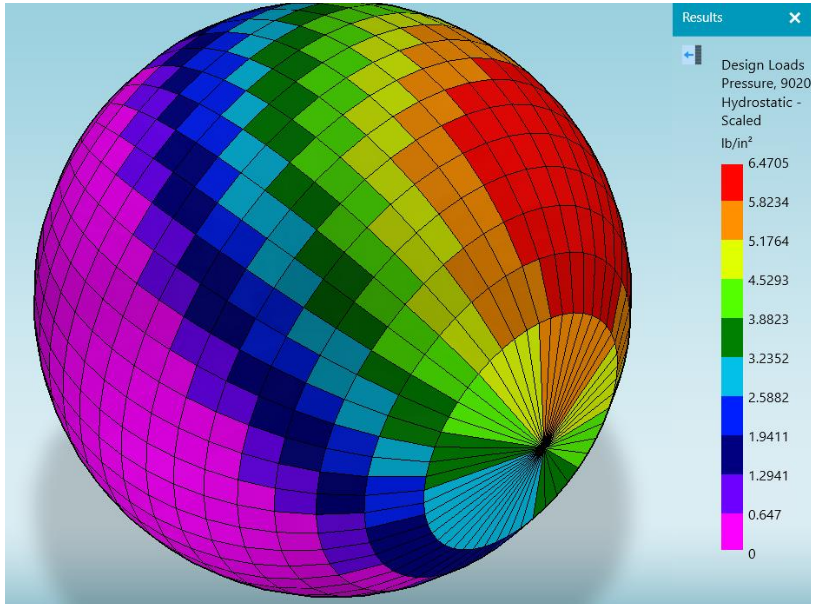


Loading and Margin for
Cylindrical Tank with 50% fill
at 45 degrees acceleration vector

SPARC module - tank

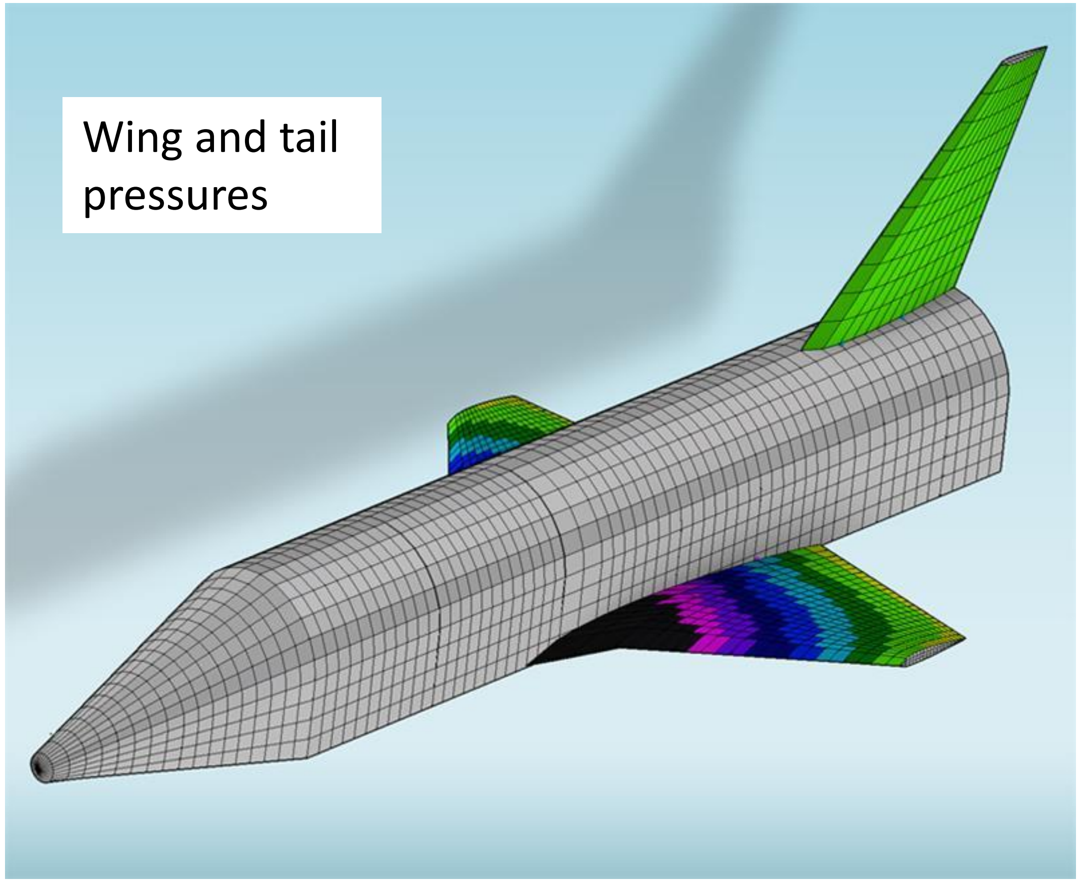


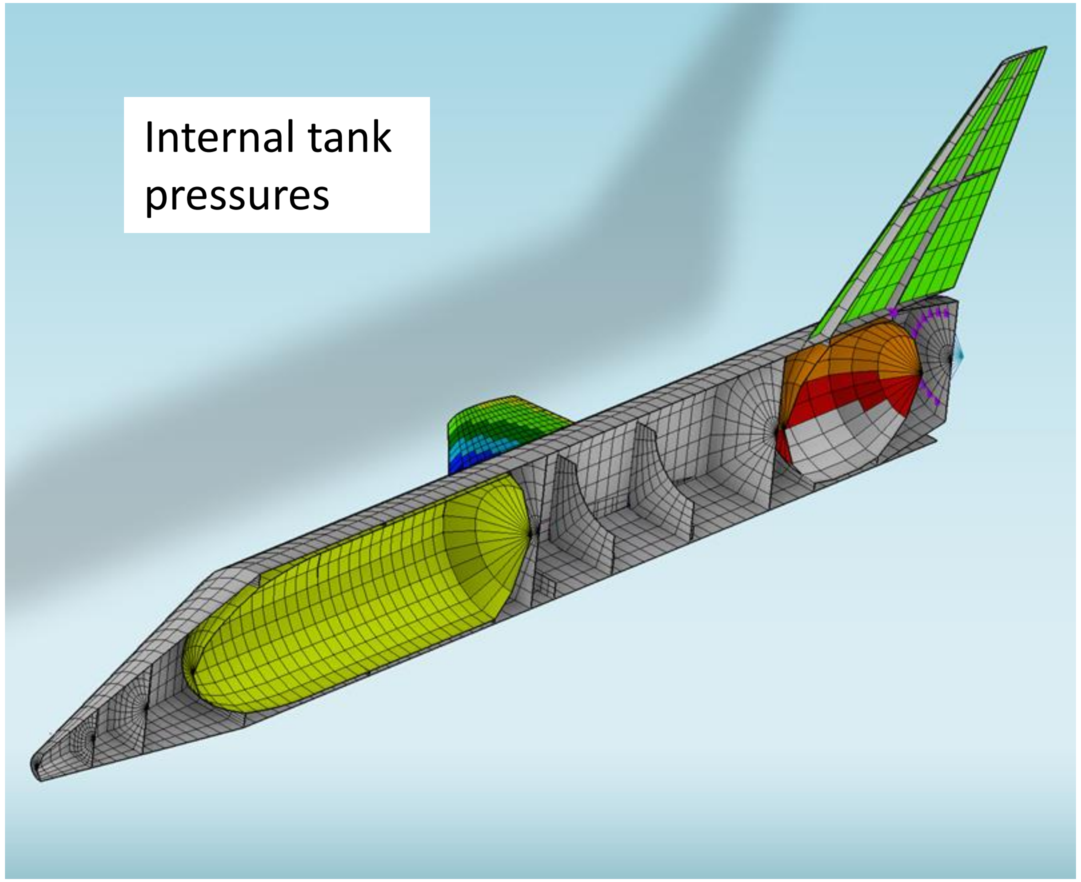
	Fill = 1%	Fill = 10%	Fill = 20%	Fill = 30%	Fill = 40%	Fill = 50%	Fill = 60%	Fill = 70%	Fill = 80%	Fill = 90%	Fill = 100%
Total	9.72%	37.50%	51.39%	72.22%	86.12%	100.00%	120.83%	127.78%	141.67%	169.45%	204.18%
Bottom	1.39%	15.28%	22.23%	29.16%	36.11%	43.06%	56.94%	56.94%	63.89%	77.77%	91.67%
Top	8.33%	22.23%	29.16%	43.06%	50.01%	56.94%	63.89%	70.84%	77.77%	91.67%	112.51%



SPARC partial vehicle loading

SPARC's full vehicle analysis and sizing is partially implemented, using the existing component modules. SPARC can apply loads to multiple wing and tank surfaces within a full vehicle model. FEA Model can be loaded into HyperX for sizing.



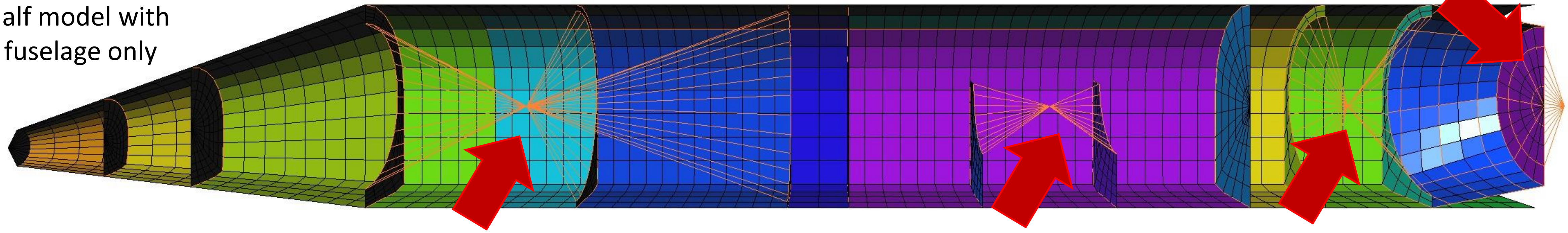


SPARC module – fuselage (in dev)

The Fuselage component sizing module is in initial development. Upon completion, basic sizing of full aircraft or launch vehicle structures will be possible. *Loft* is used to create the non-fuselage point masses and the rigid boundary element (RBE) spiders that connect them to the fuselage.

Thrust load RBE spider

Half model with fuselage only

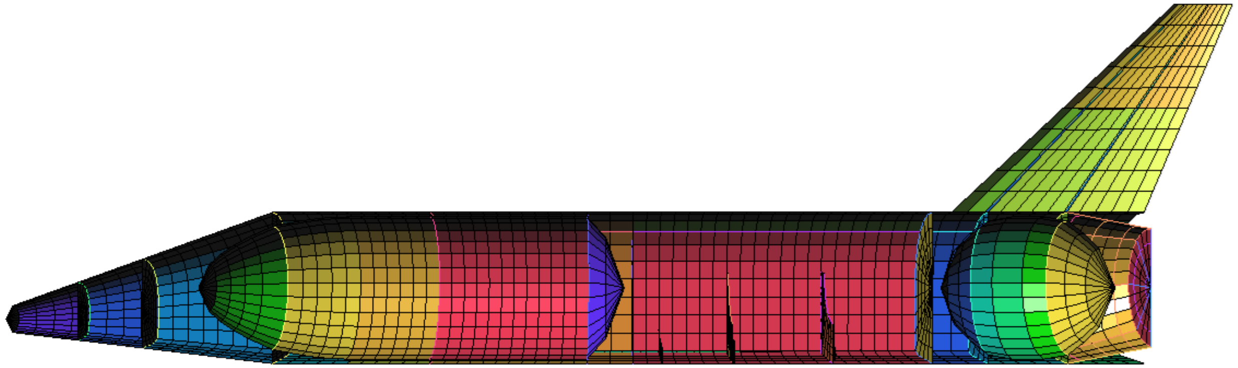


Forward Fuel tank point mass

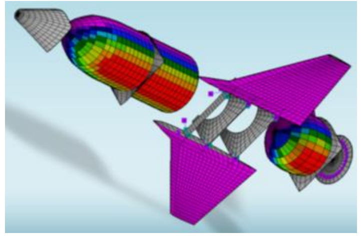
Payload point mass

Aft Fuel tank point mass

Half model with fuselage, tanks, wing, and tail.



Conclusions



- SPARC is a preliminary structural sizing environment that connects high-fidelity tools
  - *Loft* for vehicle meshing
  - Critical design load cases for each component – scaled to desired magnitudes
  - NASTRAN for finite element solution
  - ParseNAS to extract load resultants needed to compute scaling
  - HyperX for vehicle structural sizing
- HyperX's API allows it to
  - be treated as an analysis black box
  - use SPARC as a pre-processing tool to configure an analysis
  - or both!

Links to *Loft*

Backup Slides

JSON format user interface

- ISO standard text format
- Key/name followed by value
- Supported by most smart programming text editors for nice formatting, but notepad can do the job
- Python has a built-in library to convert JSON to a Python dictionary and back
  - No need to write a parser
  - Other tools can easily drive SPARC
    - change input file parameters
    - run SPARC from a command line
  - Design studies, optimization, etc. are straightforward

```
"Component Geometry":{
  "SubGeom1":{
    "Use?": "Yes",
    "Type": "Wing",
    "Name": "Main Wing",
    "LoftTemplate": "trapwing.txt",
    "LoftInputs":{
      "span": 20.0,
      "chord": 5.0,
```

Part of a SPARC input JSON file

SPARC load scaling

- SPARC is designed to treat load distribution and load magnitude as potentially separate items
- A NASTRAN run deck is created and run with each “unit” load case
- Load case reactions are extracted using ParseNAS in-house tool
- A linear scale factor is computed to produce the desired reaction load, e.g. 3 times vehicle mass for a 3G pullup case
- An updated NASTRAN deck is created with the scale factors applied and NASTRAN is run
- If exact loads are being used (e.g. CFD results) the scaling step can be skipped by the user specifying a scale factor of 1.0

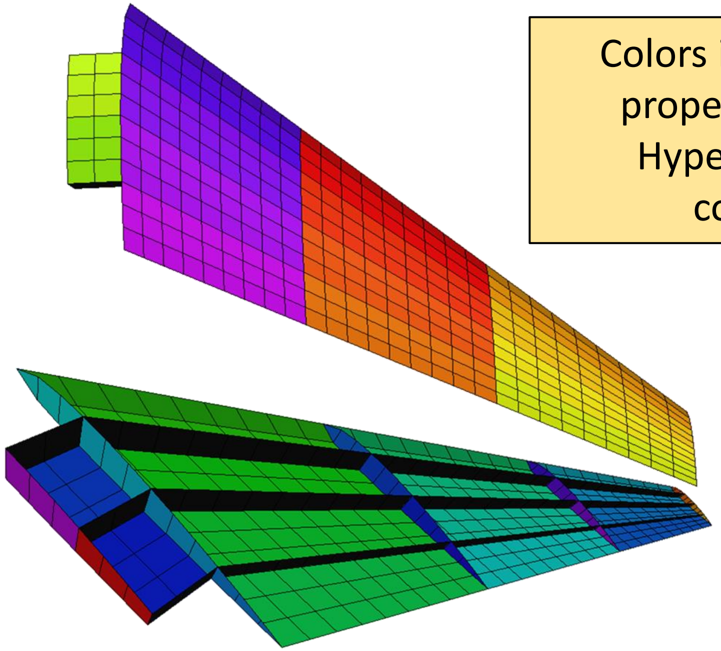
What is *Loft*?

Loft is a finite element model creation tool that takes a descriptive text file input and generates a structural mesh in NASTRAN, TecPlot, VRML, STL, etc. format.

```
object wing Main Wing
    chord 30
    span 60
    taper 0.25
    sweep 40
    wingbox 6
    nribs 4
    nspars 3
    mesh 0.5
    naca 2412
write vrml mywing.wrl
```

11-line text input file

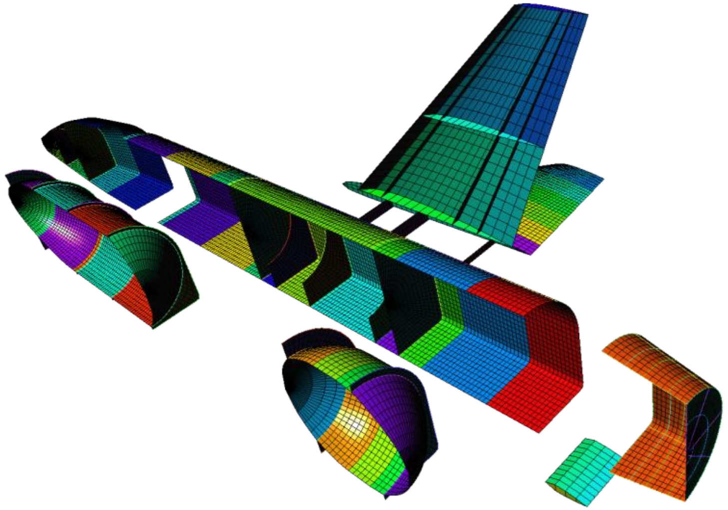
Loft



Colors indicate physical property changes for HyperSizer/HyperX components

Top skin removed to show ribs and spars

Why is *Loft* right for *SPARC*?

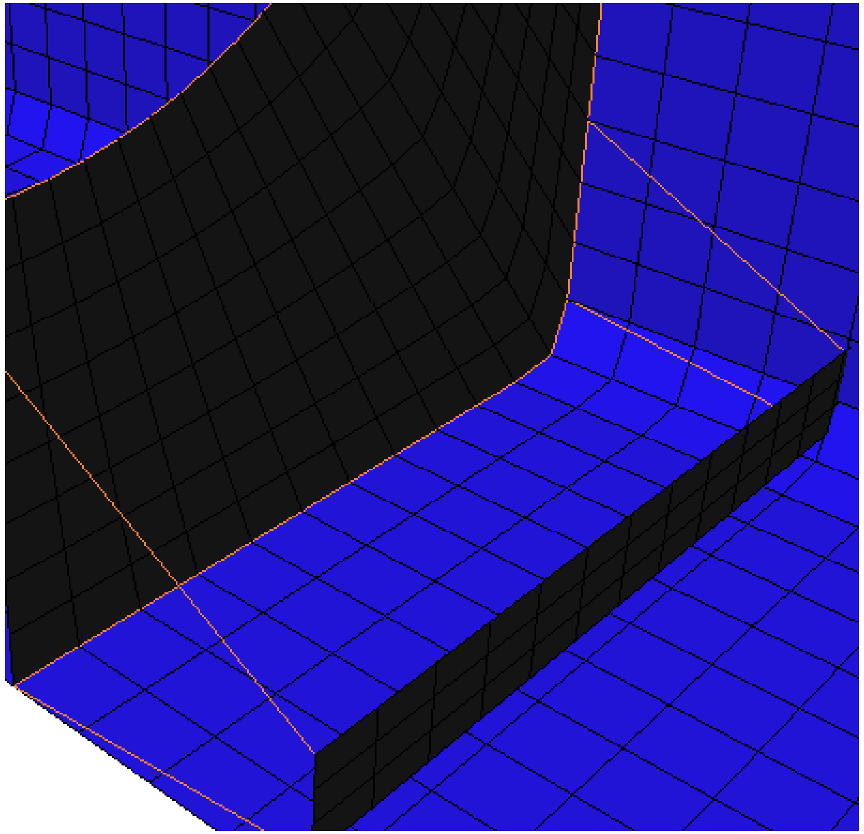


- Designed for rapid conceptual model creation
- No GUI
  - Runs from a command line
  - Reads and writes text files
- Automatic component creation in HyperSizer/HyperX
- Partial model output without changing node/element indexes
- Automatic and manual group creation of nodes and elements for application of loads and boundary conditions

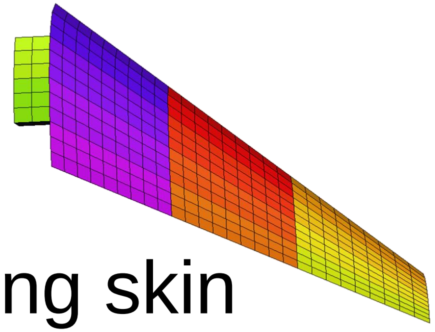
Loft is ideal for batch analysis

Loft updates for *SPARC*

- Expanded tools for marking, selecting, and reporting on partial models based on location
  - Added spatial inequalities, e.g. x>=3.5
  - Added corner node identification capability
  - Automatic surrogate model creation for situations where using an existing capability and knowledge of the model in *Loft* is more efficient than creating new code in SPARC.
- Semi-automatic creation of rigid boundary elements (RBEs) to stitch wings and tails to fuselages using spatial and corner extensions
  - Enables use of full vehicle models in an automated tool
- Support for creation of NASTRAN SPC, FORCE, PRESS, point masses and case control blocks
- Expansion of airfoil shapes to include biconvex and user defined options.
- Cloning, mirroring
- Flow control options: include/subroutines, if, goto, loops

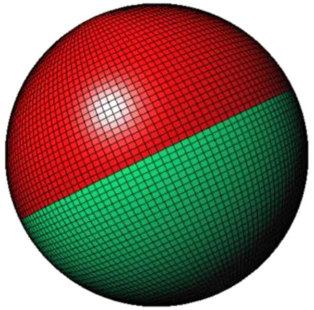


Wing load case



- A constant upward unit load pressure is applied to the wing skin
  - Outward on the upper skin, inward on the lower skin
- Nodes at the top and bottom of the leading edge and trailing edge spars are constrained
- Load reactions are computed and scaled to produce half the lift (one wing) needed for the design load case
- This is structurally conservative as it produces a higher moment at the root than a more accurate distribution would produce
- Elliptical span-wise, triangular chord-wise distribution is available as a more accurate user option

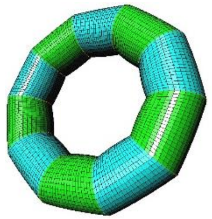
Hydrostatic tank loading

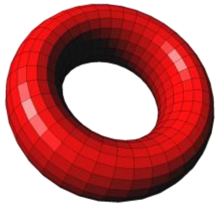


- A constant ullage pressure can be applied to all elements
- An arbitrary acceleration vector can be defined
- User specifies fuel density, volume and can guess a tank percentage fill value
- SPARC guesses free surface location based on inputs (user input or volume fraction)
- *Loft* surrogate model is created that is rotated to align with acceleration vector
- *Loft* reports all elements below guessed free surface location along with their node locations
- SPARC computes element centroids, distance from free surface (z_{free}-$z_{centroid}$), and then element hydrostatic pressure (ρ*g*h)

Hydrostatic tank loading

- NASTRAN is run, reactions read, scale factor computed and applied to produce the exact expected net load from the mass of the fuel under the given acceleration
- User can select if this scaled load is sufficient or if (planned) iteration on the free surface location is desired.





- Arbitrary tank shapes allowed
- Volume computed by either
  - projected area*distance from centroid for all elements (does not work on toroids or other shapes with voids)
  - using NASTRAN reaction force on a 100% full tank with unit density and acceleration (slower, but more accurate)

SPARC Structure

- Object oriented data structures stored in JSON format
- Interface code written in Python
- All items have a reasonable default value
- Only changes need to be specified