Aerospike Nozzle Design Using iSIGHT

Andrea Salas
NASA Langley Research Center
Multidisciplinary Optimization Branch (MDOB)
http://fmad-www.larc.nasa.gov/mdob/MDOB

2000 International iSIGHT User's Conference
October 2 - 4, 2000

Outline

- Aerospike Nozzle background
  - provide history and description of application
  - describe results of previous study
- Aerospike Nozzle iSIGHT implementation
  - describe implementation in iSIGHT
  - examine issues such as ease of use, performance, and benefits

Background

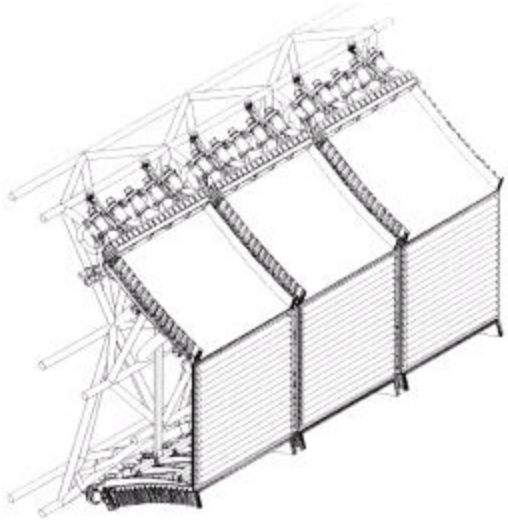
- In 1996, the Aerospike Nozzle application project was a collaborative effort between MDOB and Rocketdyne with the following goals:
  - demonstrate MDO on engine concepts
  - assess performance of various MDO approaches

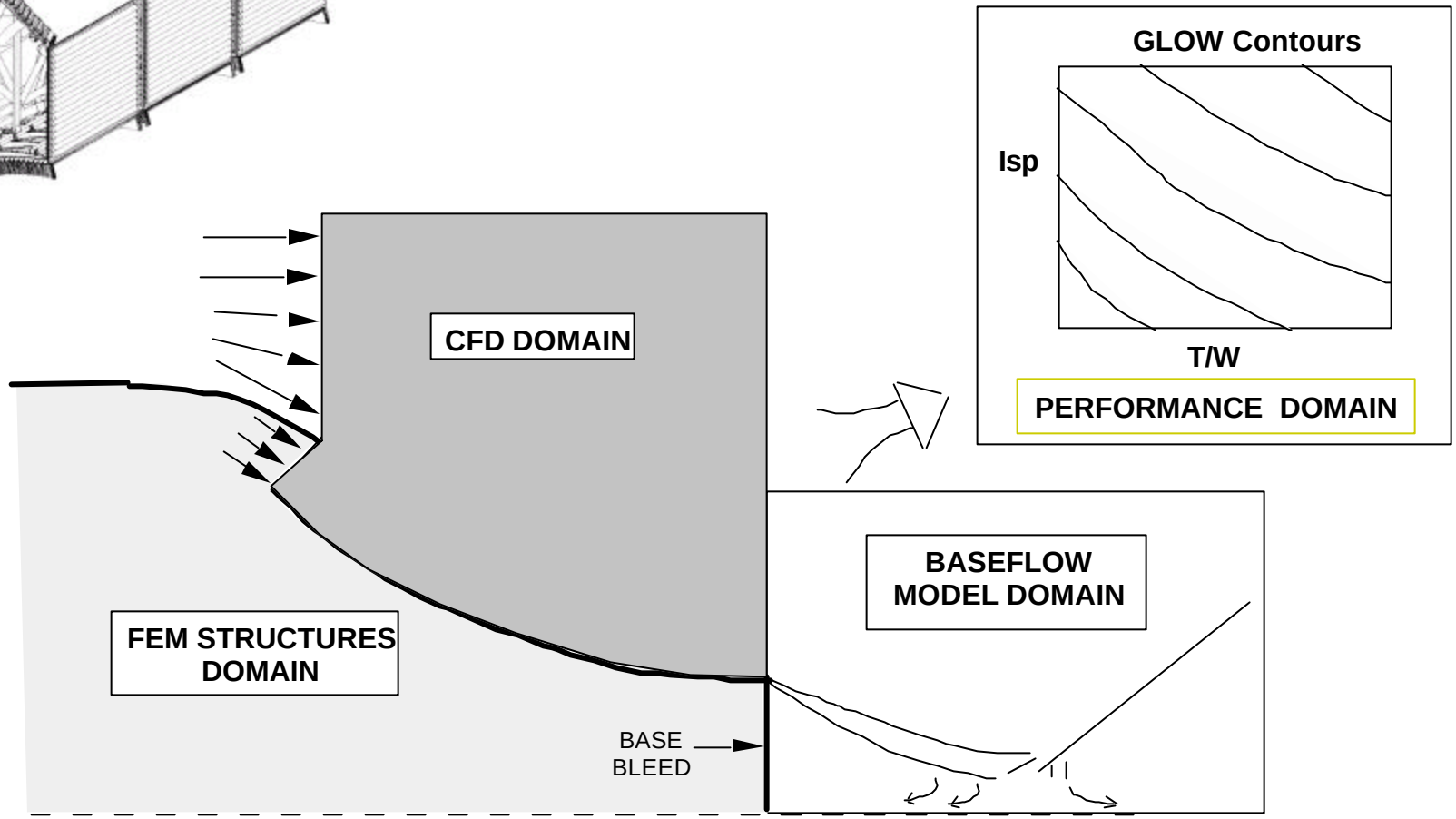
- Acknowledgements
  - John Korte (aerodynamics)
  - Jack Dunn (structures analysis)
  - Natalia Alexandrov (MDO formulations)

Aerospike Nozzle Baseline Application

- Aerodynamics
  - calculates engine thrust and static loading on the nozzle structure
  - uses nonlinear CFD and a baseflow model
- Structures
  - calculates weight and structural responses
  - 2D FEM
- Gross-Lift-Off-Weight (GLOW)
  - curve fits of GLOW for mission-averaged I_{SP} and Thrust/Weight
- MDO formulation
  - multidisciplinary feasible (MDF) strategy

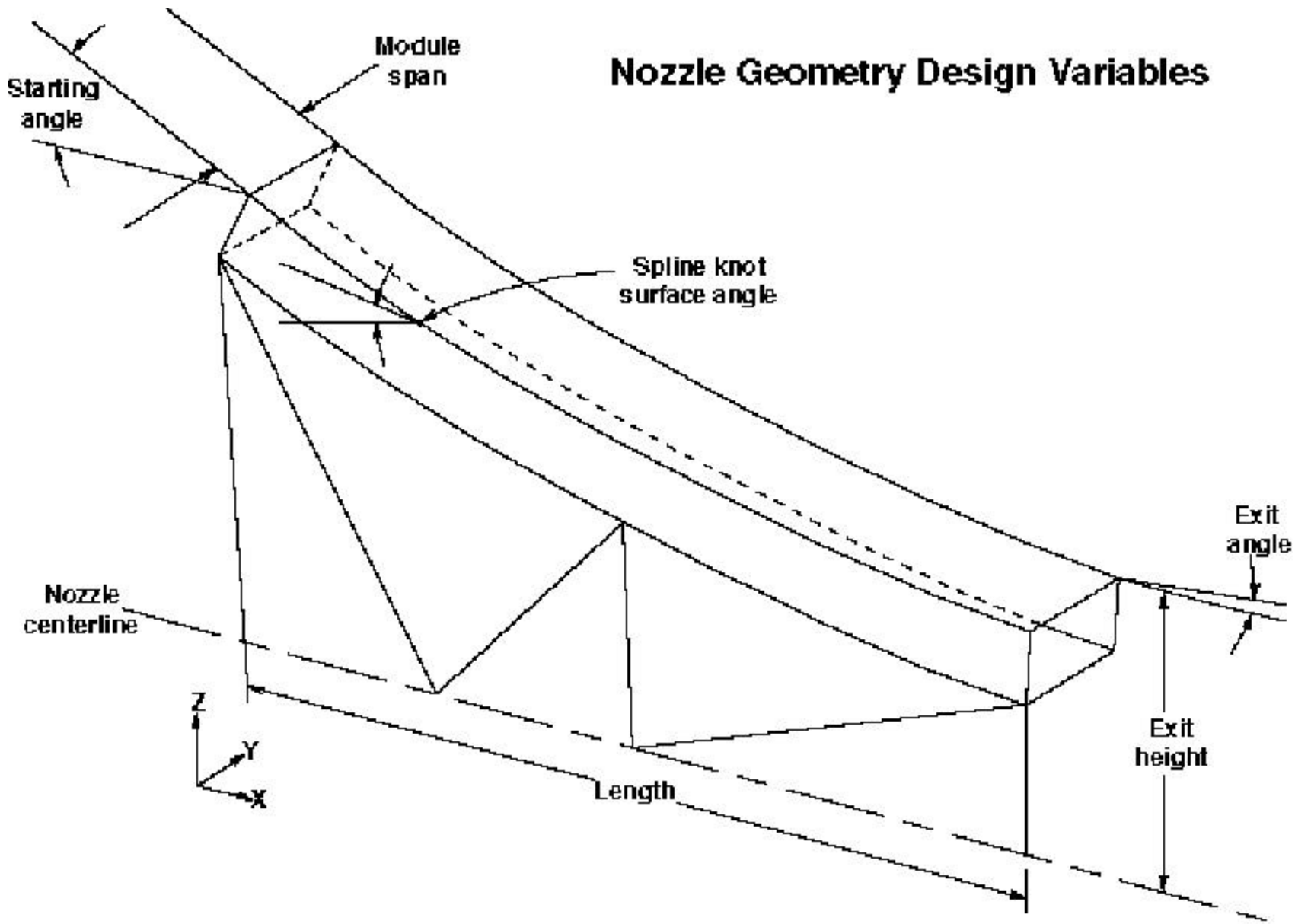
Aerospike MDO Domain Decomposition



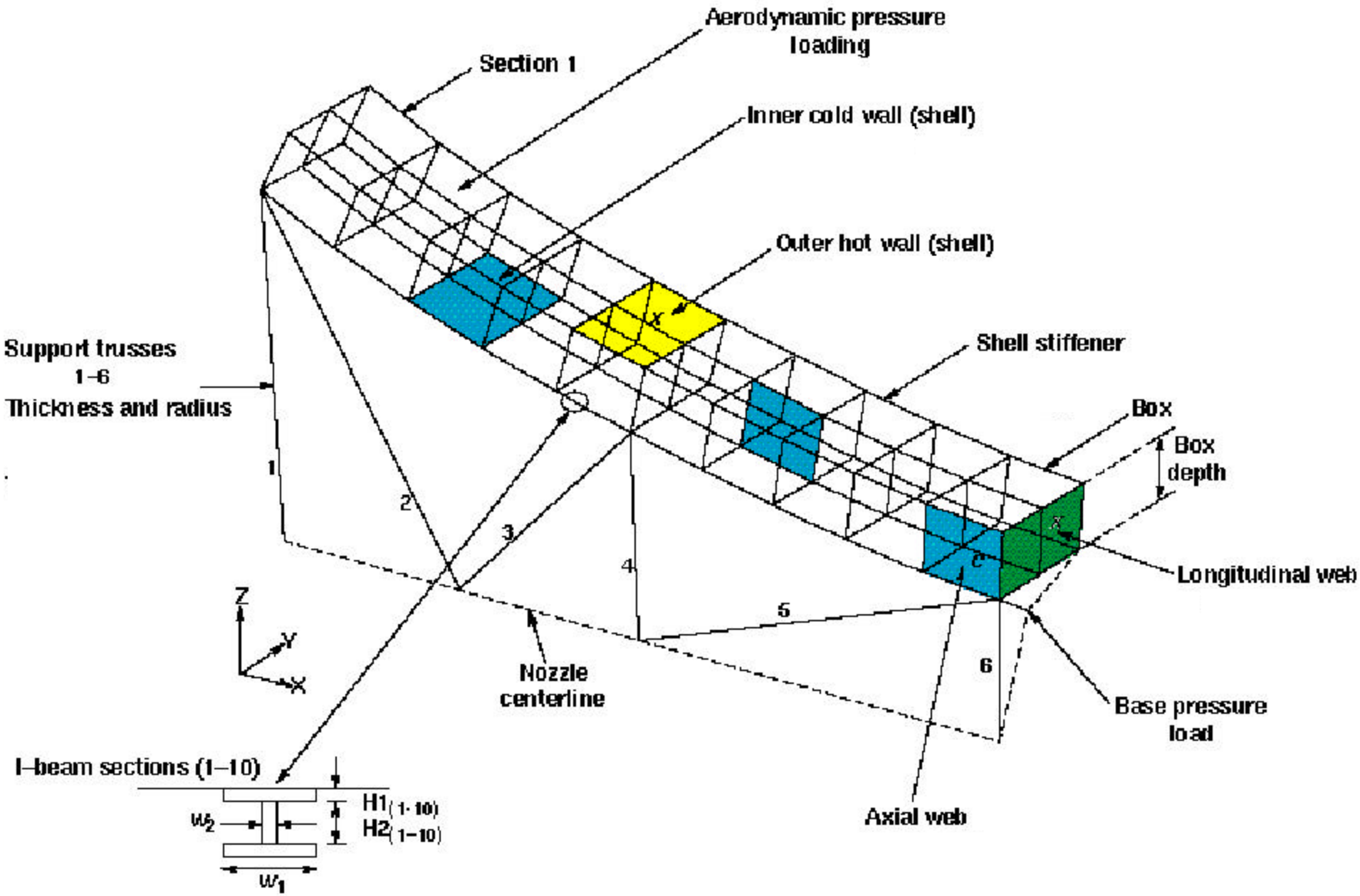


Aerospike Design Problem

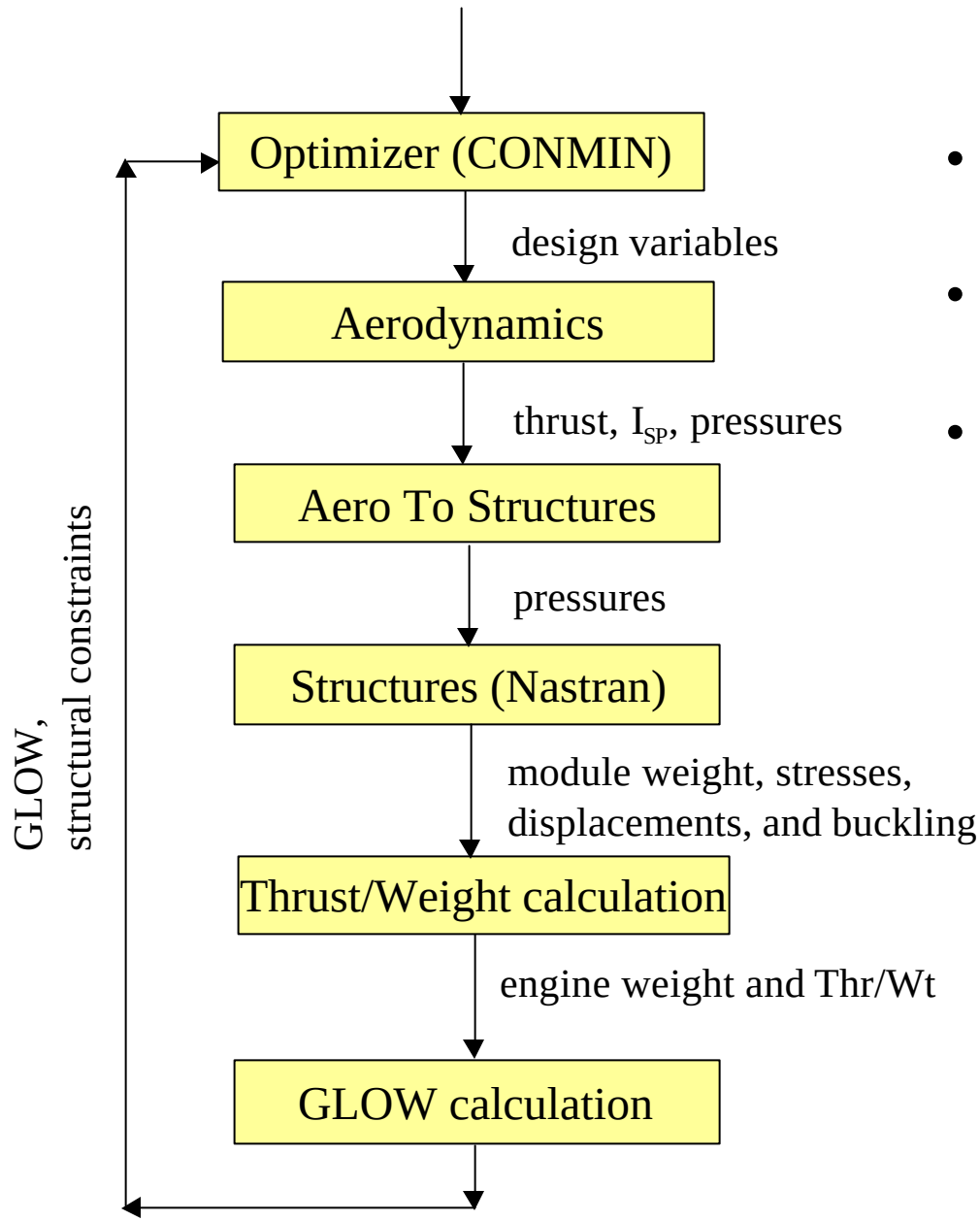
- Objective: Minimize Gross-Lift-Off-Weight
- Design parameters (18)
  - 4 geometry variables
    - Thruster angle
    - (2) Surface slopes
    - Nozzle base height
  - 14 structural variables
    - I-beam parameters (4)
    - Thicknesses (7)
      - Hot wall, cold wall, axial web, long. web, stiffeners, trusses, base plate
    - Radii (2)
      - trusses, stiffeners
    - Structural box depth
- Structural constraints (564)
  - displacement, stress, buckling


Nozzle Geometry Design Variables
Module span
Starting angle
Spline knot surface angle
Exit angle
Nozzle centerline
Z
Y
X
Length
Exit height

Structural Loading and Design Variables



Aerospike Computational Processes



- **Optimizer** computes gradients using finite differences
- **Aero to Structures** writes pressure data to Matlab file
- **Structures** generates Nastran model and runs analysis on remote machine

Aerospike Baseline Results

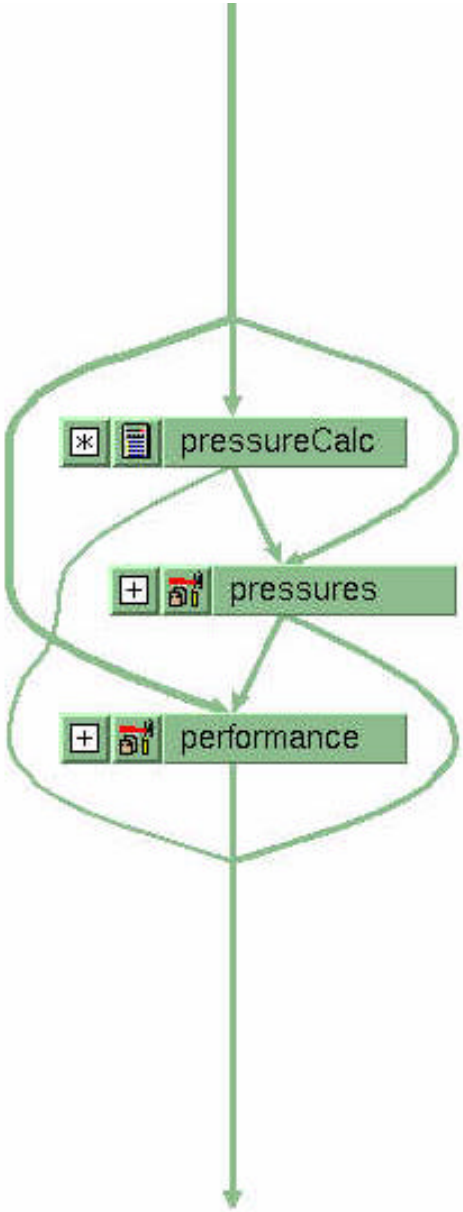
- Method 1 : Sequential Optimization
  - aerodynamics (maximize thrust)
  - structures (minimize weight)
- Method 2: Multidisciplinary Design Optimization
  - minimize GLOW
  - initial design variable values are the results from Method 1
- Significant improvement obtained using MDO approach
  - reduction in GLOW is from a slight decrease in thrust, which reduces nozzle weight
- For more information
  - Korte, J. J.; Salas, A. O.; Dunn, H. J.; Alexandrov, N. M.; Follet, W. W.; Orient, G. E.; and Hadid, A. H.: *Multidisciplinary Approach to Aerospike Nozzle Design.* NASA TM-110326, February 1997.
  - http://techreports.larc.nasa.gov/ltrs/ltrs.html

Aerospike iSIGHT implementation

- Goal: Explore usability of iSIGHT for Aerospike
  - ease of use, performance, and benefits
- Implement baseline application approach in iSIGHT
  - single discipline and multidisciplinary analyses and optimizations
- Validate iSIGHT Aerospike implementation against standalone results
- Performance timings
  - *Task Manager* loading of description file is not timed
  - MDOL *Audit* statement is set to off
  - timings taken with iSIGHT 5.5

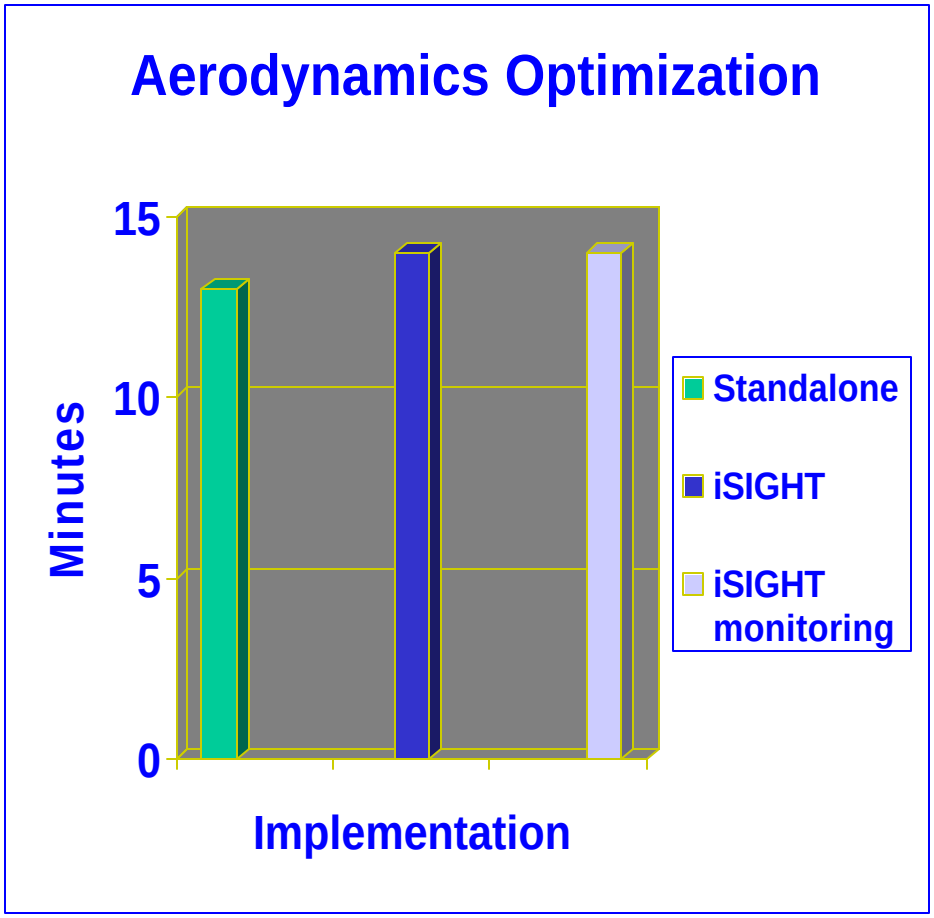
Aerodynamics Task

- iSIGHT modules
  - 1 calculator, 2 simcodes
- 19 iSIGHT parameters
  - 8 inputs
  - 4 auxiliaries
  - 7 outputs
- Files parsed
  - *pressures* (2 input, 1 output)
  - *performance* (1 input, 1 output)



Aerodynamics Optimization

- Optimization
  - CONMIN
  - 4 design variables
  - unconstrained
- Reproduced standalone results
- Negligible overhead from iSIGHT
  - monitoring included 4 design variables and 2 responses

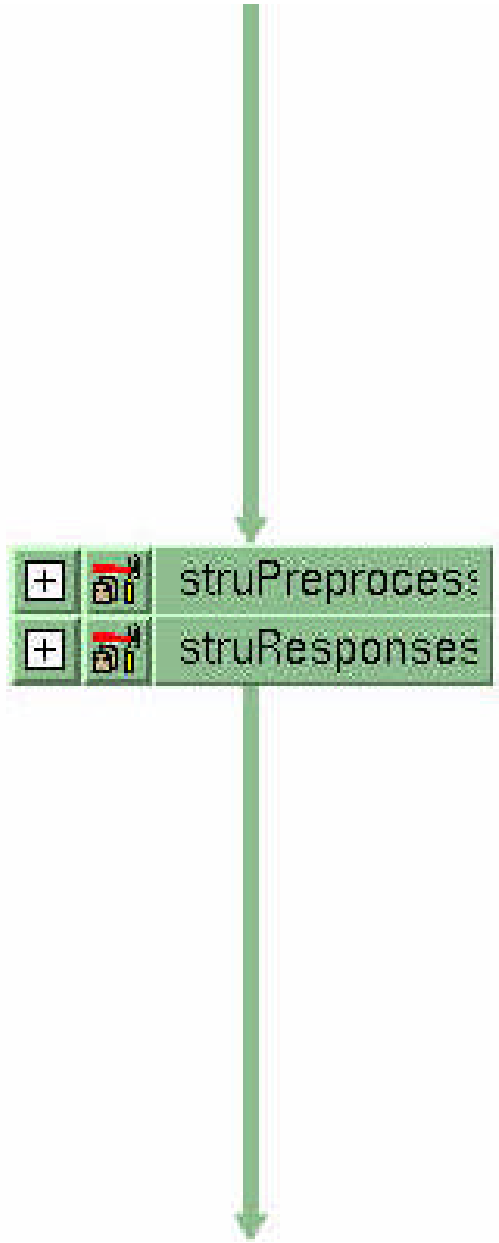


Aerodynamics Task Observations

- Simple file parsing
  - *moveto*, *replace*, *find*, *read*, *provide*
- Task built entirely with iSIGHT *Process Integration* and *Task Manager* GUIs
- Can prevent CONMIN finite difference calculations from being stored in database
- Multiple Optimization Plans in a task is convenient
- Suggestion: Have database record design at the end of each optimization iteration

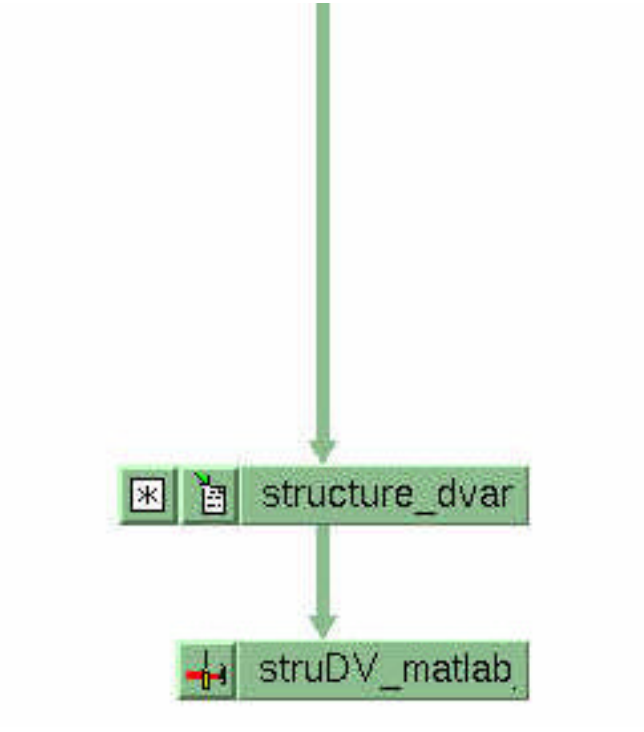
Structures Task (1)

- iSIGHT modules
  - 2 simcodes
- 435 iSIGHT parameters
  - 14 inputs
  - 1 auxiliary
  - 420 outputs (organized in arrays)
- Files parsed
  - *struPreprocess* (1 input)
  - *struResponses* (2 output)
- *struResponses*
  - shell script that runs Nastran remotely (uses “rcp” and “rsh”)
  - iSIGHT NASTRAN interface was not used

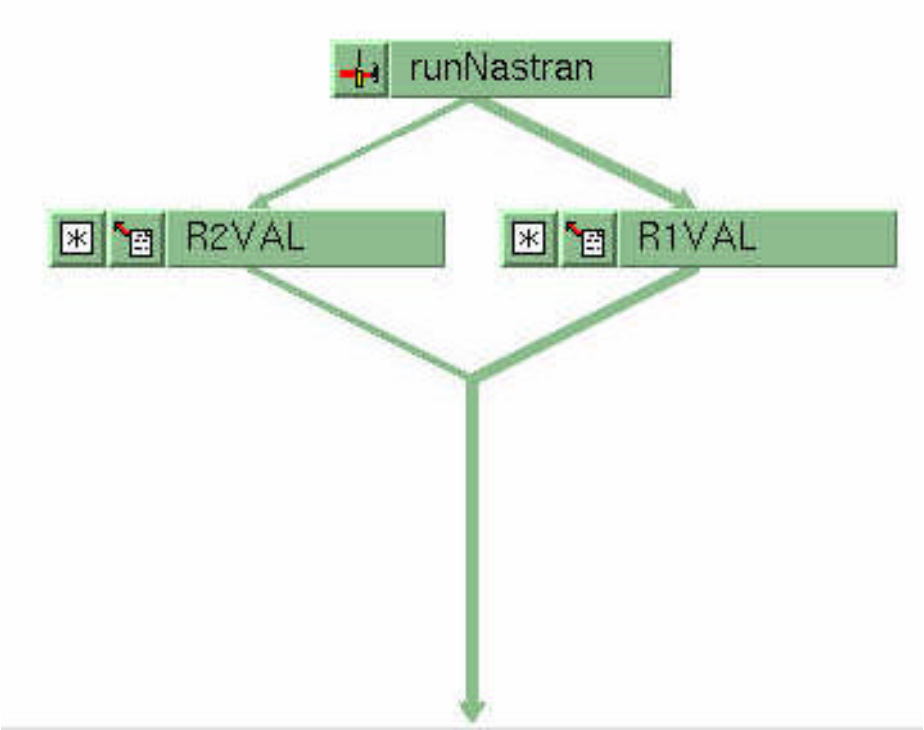


Structures Task (2)

- *struPreprocess* output is input to *struResponses*, but is not parsed
  - Use *Sequence* to order processes



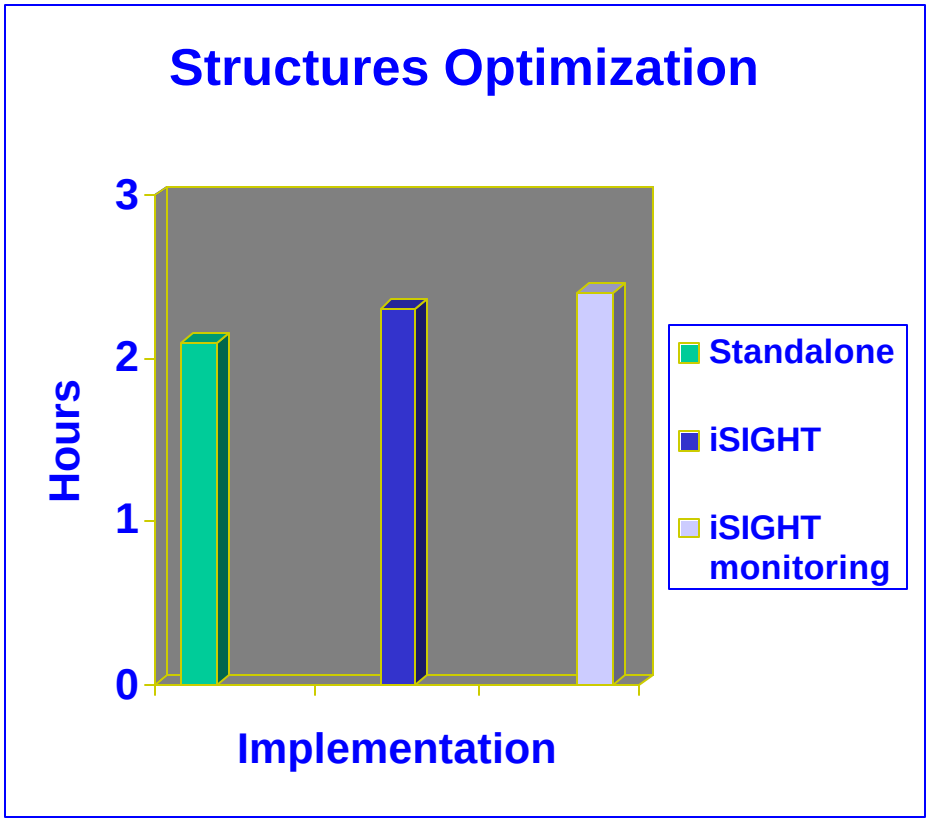
Expanded view of *struPreprocess*



Expanded view of *struResponses*

Structures Optimization

- Optimization
  - CONMIN
  - 14 design variables
  - 564 constraints
- Reproduced standalone results
- Acceptable overhead from iSIGHT
  - monitoring included 16 parameters on 2 pages

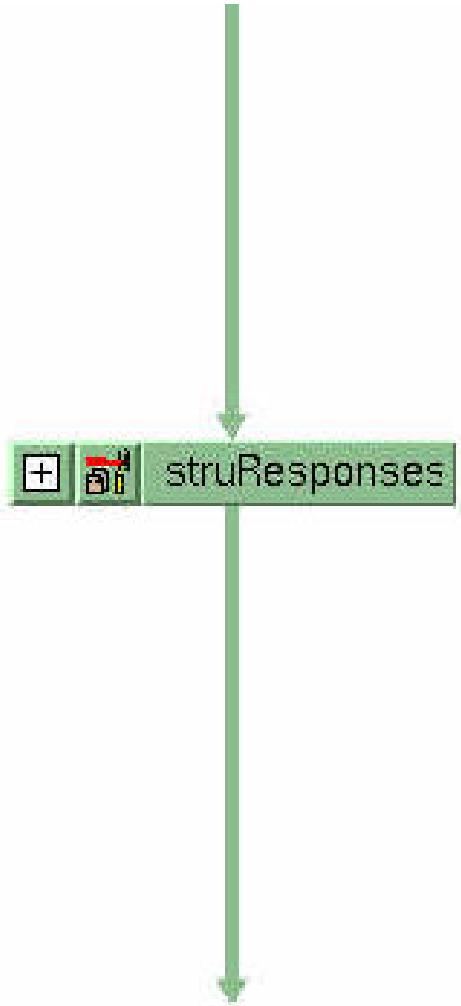


Structures Task Observations

- More complicated file parsing
  - while loops, if statements, expressions, arrays
- Parameter shortcut keys allow fast entry of constraint information
- Structures task built *almost* entirely with iSIGHT *Process Integration* and *Task Manager* GUIs
  - MDOL required api_SetDeltaForInEqualityConstraintViolation
  - later modified MDOL to use abbreviated form for constraints
- Suggestion:
  - mark responses responsible for infeasible designs
  - allow user to select parameters to be stored in database

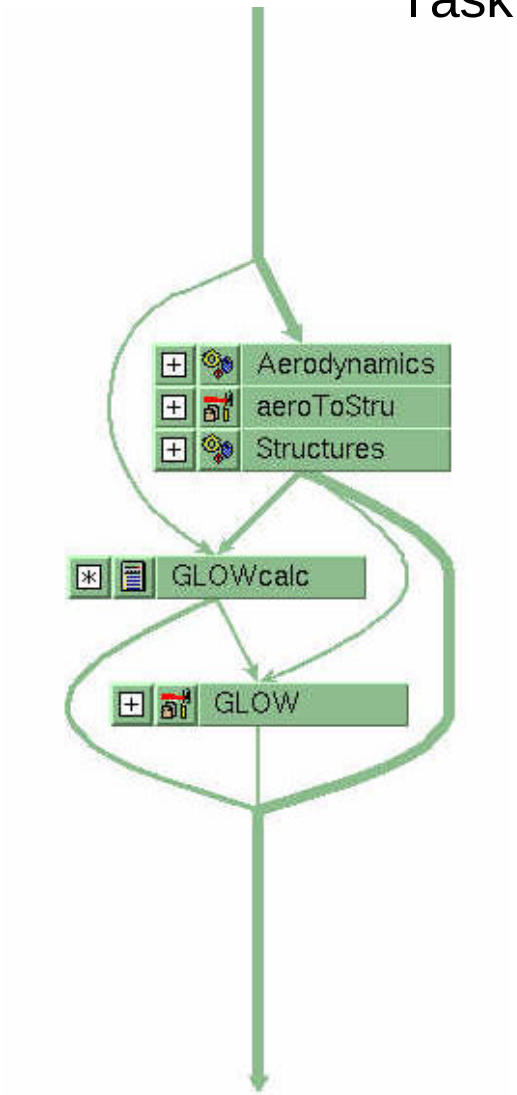
Using iSIGHT Distributed Processing

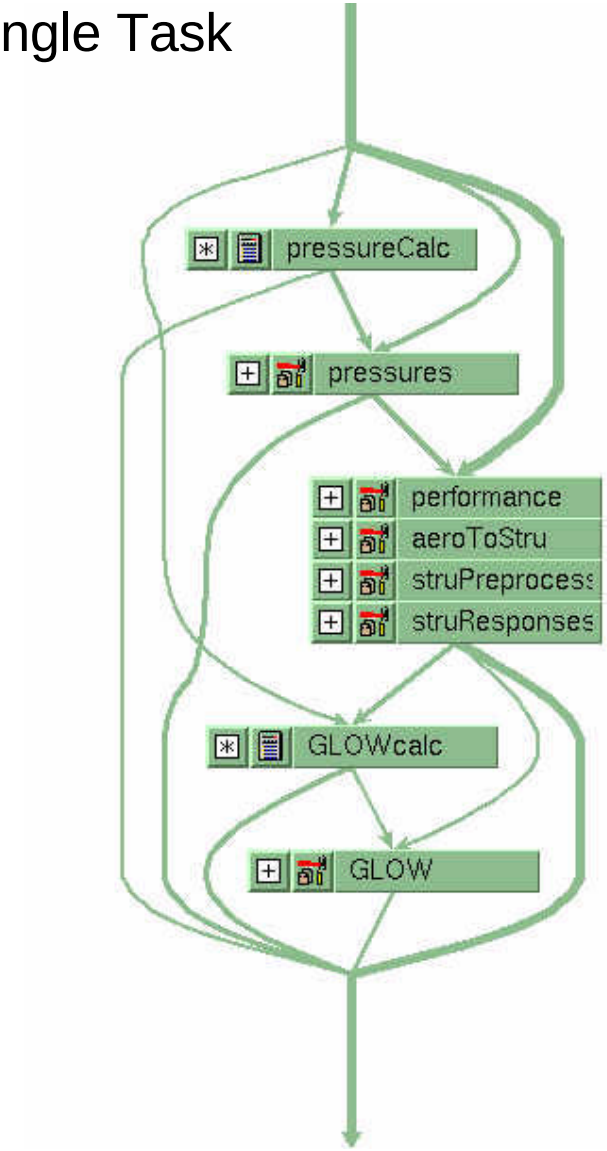
- Simcodes *struPreprocess* and *struResponses* combined into 1 simcode
  - iSIGHT must know about files to be transferred
- Files parsed
  - *struResponses* (1 input, 2 output)
- Host and topology are defined
- iSIGHT starts the remote process and transfers files



MDO Aerospike Task

Task Hierarchy versus Single Task

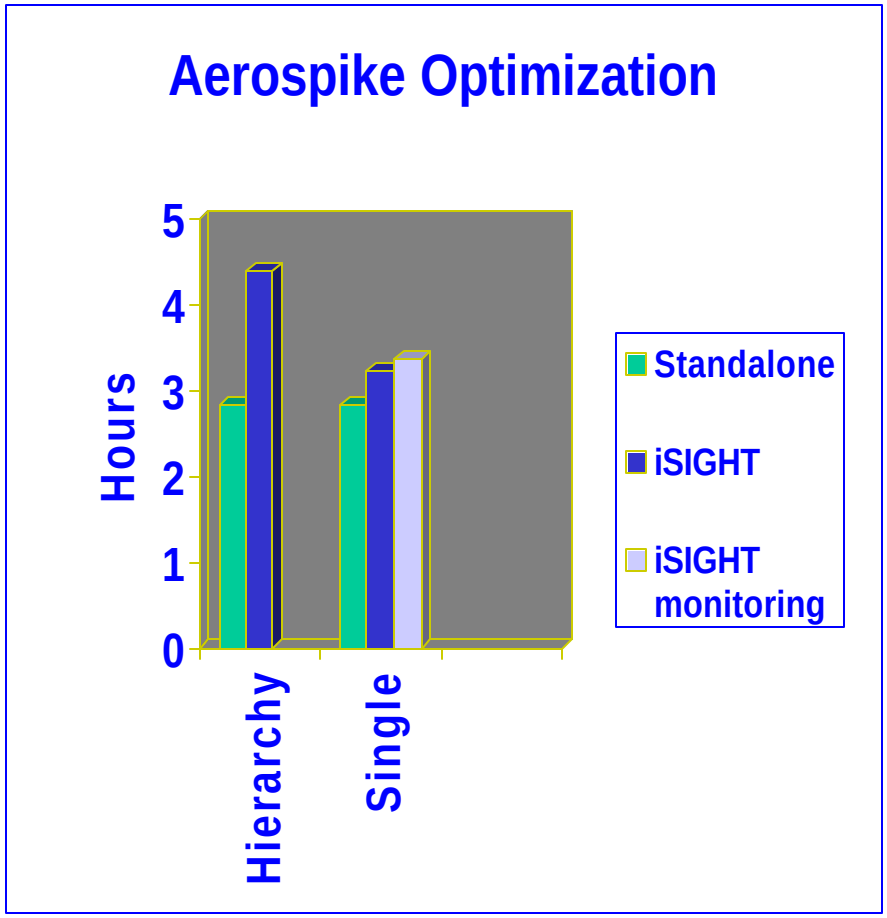




- 458 iSIGHT parameters (22 inputs, 5 auxiliaries, 431 outputs)

Aerospike Optimization

- Optimization
  - CONMIN
  - 18 design variables
  - 564 constraints
- Reproduced standalone results for 10 iterations
- Task hierarchy produces more overhead
- Monitoring for single task case
  - included 26 parameters organized on 3 pages



Aerospike Task Observations (1)

- iSIGHT distributed processing was not used
  - matlab pressure data file must be transferred, but is not parsed
- Hierarchical task organization
  - allows convenient selection of single discipline analysis/optimization and multidisciplinary analysis/optimization
  - *Component* subtasks make the process integration faster
- More MDOL editing was required for Aerospike Tasks
  - redefining the parent task parameters
  - cannot cut and paste simcodes from one part of task hierarchy to another

Aerospike Task Observations (2)

- Single task execution is preferred over the hierarchical tasks for performance reasons
- Exploring different sets of design variables is convenient
- Organizing monitored values on multiple pages is convenient
- Aerospike 10 iteration case produced a 1.4 MB database (with gradients) which was easily traversed
- Aerospike case run to completion generated a 5.7 MB database (with gradients)
  - slower, but still no problems traversing

iSIGHT Aerospike Implementation Effort

- Preliminary Effort
  - majority of project time was spent resurrecting former codes and generating standalone results
- Process Integration and Optimization Effort
  - approximately 3 weeks
  - majority of effort was with the Structures parsing and validating constrained optimization against standalone results (needed to use *ScaleFactor* to normalize constraints)
- Additional Effort
  - exploration of additional features such as multiple optimization plans, component library, distributed processing, performance timings, different task organizations, etc.

Summary

- iSIGHT has advantages for the Aerospike problem
  - not hard to implement within iSIGHT
  - performance is acceptable within iSIGHT
  - database and monitoring are helpful
  - iSIGHT reduces code and file management for the user
  - iSIGHT reduces time required to explore variety of optimization schemes and design parameters
- Some suggested areas for improvement:
  - additional database information and screening
  - GUI flexibility for making changes to task organization
  - recognition of file dependencies (not parsed by iSIGHT)